MESOS.

Administrar clústers compartint recursos de forma dinàmica.

Joan Carles Martínez Rodríguez

Treball de Final Grau en Enginyeria Informàtica

Itinerari d'Arquitectura de Computadors i Sistemes Operatius

Consultor: Francesc Guim Bernat

Lliurament: 30/12/2014

A Eugènia i Biel, per la vostra paciència, comprensió i suport aquests anys d'estudi.

A la meva família i amics, especialment a Gabriel B. P. que entre d'altres coses em va facilitar el parc de màquines per tirar endavant el projecte.

A tu Cesc, ha estat un luxe tenir-te de consultor.

Entre tots m'heu ajudat a superar els mals moments, i arribar al final.

Terrassa, 30 de desembre de 2014

Resum

El nivell actual de maduresa d'Internet ha fet aparèixer un escenari d'interacció, i accés a dades, que provoca una enorme pressió en la millora dels temps de resposta en els sistemes de tractament i servidors de dades. L'usuari no vol quedar sense servei en un entorn d'alta mobilitat com l'actual, i això significa que les aplicacions i les dades han d'acompanyar-lo en tot moment, amb independència de la seva localització geogràfica o temporal. En aquest entorn són necessaris gestors, o planificadors, de sistemes de computació molt eficients i resilients, que permetin aprofitar tots els recursos disponibles per tractar i servir dades, a qualsevol aplicació i en qualsevol moment. A més una altra màxima a complir és que els sistemes s'han d'amortitzar en termes de producció, per tant tots els recursos de computació han d'estar disponibles, i executant tasques, el màxim de temps possible. Aquesta gestió també recau sobre els gestors de recursos.

Aquest treball estudia una de les solucions actuals als requisits anteriors de gestió dels clústers de computació. MESOS és un gestor dinàmic de recursos basat en un mecanisme d'administració de dos nivells anomenat oferidor de recursos. La gestió que realitza és d'un alt nivell de granularitat, al tenir una visió del conjunt dels nodes de computació com si fos un sistema únic de recursos (*pool*). Aquest model, de forma simplificada, és com el d'un sistema operatiu que administra recursos d'un únic node, però que en realitat poden ser milers els nodes sota el domini de MESOS. Aquesta visió ajuda a les tasques d'administració de recursos, i permet oferir característiques eficients de servei, a part de garantir la pervivència dels sistemes. El treball pretén que s'observi i valori el funcionament d'un petit clúster privat, administrat per MESOS, i dissenyat per a treballs de computació paral·lela gràcies a les llibreries MPI, i l'entorn d'aplicació que incorpora MESOS per executar aquest tipus d'aplicacions.

Paraules clau: MESOS, MESOSPHERE, contenidor LXC, Docker, MPI, MPE, entorn d'aplicació, centre de dades, computació en el núvol, *big data*, recursos, *IaaS*, mètrica, planificador.

Abstract

The current level of maturity of the Internet has appeared a scene of interaction and access to data, causing enormous pressure to improve response times in data treatment systems and data servers. The user does not want to be without service in a high mobility environment like this, which means that applications and data must be accompanied at all times, regardless of geographic location or time. In this environment are required computer systems schedulers, or planners, very efficient and resilient, allowing to use all available resources to try use data to any application at any time. Also another maximum to fulfill is that systems are amortized in terms of production, so all computing resources must be available and running tasks, as long as possible. This management also falls on the resource managers.

This paper studies a current solutions to the above requirements management cluster computing. MESOS is a dynamic resource manager based on a mechanism called resource offer on two levels of government resources. The management is carried out by a high level of granularity, to have an overview of all the computing nodes as if it were a single system resources (pool). This model, in simplified form, is like an operating system that manages the resources of a single node, but may actually be thousands of nodes under the rule of MESOS. This view helps resource management tasks and features to offer efficient service, in part to ensure the survival of the systems. The study aims to assess the operation and observed a small private cluster administered by MESOS, and designed to work in parallel computing through the MPI libraries, and application environment that incorporates MESOS to run such applications.

***Keywords**: MESOS, MESOSPHERE, LXC container, Docker, MPI, MPE, framework, datacenter, cloud computing, big data, resources, IaaS, benchmark, scheduler.*

Índex de continguts

1. Introducció al projecte

En aquest apartat es tractaran les parts inicials del projecte, com són les motivacions per triar aquest camp d'estudi, l'establiment formal d'objectius a superar i la planificació temporal per resoldre'ls.

1.1 Justificació

El suggeriment temàtic del projecte ha sortit del propi consultor, que en un correu em presentava la plataforma MESOS[1] per dedicar-hi el projecte, si ho trobava interessant. Immediatament vaig començar a llegir les diferents documentacions disponibles a les web apache.mesos.org i www.mesosphere.com, versió MESOS orientada a *IaaS*[a], per tal de treure una imatge aproximada del seu significat. Una vegada analitzada la informació, encara que de forma molt inicial, i les diferents possibilitats, la tria va ser fàcil pel següent motiu:

- Cada dia és més important la presència de grans centres de computació, a causa de la immensa capacitat d'emmagatzematge i tractament de dades necessaris, que permetin extreure informació útil en un temps raonable. Aquest funcionament és el que coneixem des de relativament fa poc com *big data*, o la creació de sistemes eficients de computació paral·lela pel tractament massiu d'informació. El coneixement d'aquestes plataformes s'esdevé cabdal en organitzacions de grau mig o superiors, o en projectes informàtics relacionats amb grans tractaments de dades. Per tant de gran interès professional.

De l'estudi d'aquesta problemàtica neix la motivació d'aquest projecte: la necessitat de conèixer diferents aspectes relacionats amb aquestes grans infraestructures de gestió d'informació, i també d'estudiar formes diferents d'aplicar una solució eficaç.

1.2 Descripció general

Per poder assolir els coneixements indicats al punt anterior, la idea general del projecte és l'estudi de la plataforma MESOS, com a gestora de *datacenters*[b] o sistemes de computació de tipus *cloud computing*[c]. L'estudi no és només teòric sinó que s'espera implementar un clúster privat gestionat sobre aquesta plataforma, i format per maquinari disponible d'ús comú. D'aquesta manera s'espera conèixer amb profunditat les tècniques d'instal·lació, execució i monitorització d'activitat dels diferents components que formen el *datacenter* així configurat. També s'espera adquirir un coneixement profund sobre les problemàtiques generals de disseny, instal·lació i execució d'un *datacenter* format per milers de nodes, i s'espera comparar la solució desenvolupada a MESOS, amb aplicacions distribuïdes i *frameworks*[d], que traduirem com a entorn, amb altres tècniques com la virtualització o altres plataformes de gestió de *datacenters*.

1.3 Objectius

Si seguim la motivació per realitzar el projecte, els objectius que hem de tractar són els següents:

[a] IaaS, *Infraestructure as a Service*, servei bàsic de computació en el núvol en el que s'ofereixen servies bàsics de computació, virtualitzats generalment, a demanda del client. (http://en.wikipedia.org/wiki/Cloud_computing)
[b] Centre de dades i/o càlcul. Instal·lacions especialment dedicades a sistemes d'emmagatzematge de dades, computació i telecomunicacions. (http://en.wikipedia.org/wiki/Data_center).
[c] Computació en el núvol. Capacitat d'oferir serveis gràcies a l'agrupació de recursos i la seva compartició i accés mitjançant a Internet. (http://en.wikipedia.org/wiki/Cloud_computing).
[d] Esquema de desenvolupament o implementació (abstracció) que dóna coherència al model de dades utilitzat en una aplicació. (http://en.wikipedia.org/wiki/Software_framework).

- Conèixer les bases del disseny de clústers de computació i la problemàtica de la compartició de recursos en la millora d'eficiència.

- Estudi del gestor de clústers MESOS, en les següents funcions:

    - Creació d'un clúster totalment operatiu basat en màquines físiques i virtuals, incloent la gestió d'instal·lació de programari i configuració de xarxa.

    - Administració del clúster amb les funcions que incorpora la plataforma MESOS.

    - Execució activa del clúster per demostrar les seves funcionalitats de treball distribuït, utilitzant alguns dels entorns desenvolupats en aquest moment.

    - Projecció en el control de milers de nodes per a treball distribuït.

- Extracció de conclusions i comparació amb altres gestors actuals.

D'aquesta manera el projecte es converteix en un treball d'investigació pràctica, on s'aplica una solució real sobre un clúster per a la seva avaluació.

1.4 Enfocament i metodologia emprada en el desenvolupament

La realització del present treball seguirà la definició clàssica de gestió d'un projecte informàtic, què el podem fixar com el procés de direcció i control que ens porta a la seva definició, posada en marxa i avaluació, i tot dintre d'un control de recursos i terminis fixats per a la seva realització. En aquest cas la definició més clara és la del termini límit de lliurament, fixat per al final del present quadrimestre, i els recursos estan limitats al maquinari i programari disponible i les hores-treball que pugui dedicar al seu desenvolupament. En aquest cas em toca fixar com a cap de projecte la coordinació necessària per portar-lo a la seva fi, si és possible amb èxit, i a més la seva realització pràctica. En aquest cas la metodologia seguida ha estat fixar els següents objectius generals:

- Definir unes funcionalitats determinades

- Respectar els terminis que s'han marcat al llarg del projecte per al lliurament de cada part i el tot

- Certificar el compliment de les fites i objectius fixats del projecte

No es parla en aquest cas de respectar un pressupost determinat, perquè no era necessari fer cap inversió addicional ja que es disposa del material necessari, i les hores de treball a facturar a un client no existeixen com a tal.

En qualsevol cas es tracta d'un projecte i com a tal existeixen uns riscos inherents al seu desenvolupament:

- Fer-ho excessivament gran i no assolir els terminis de lliurament amb tots els objectius realitzats.

- Mal coneixement de la tecnologia emprada.

- Qualitat esperada i especificacions estables.

El risc directe més gran és el mal coneixement de la tecnologia emprada. En aquest cas els sistemes de gestió de clústers i quina de les possibles aplicacions possibles que s'executarà al clúster. Per aquest motiu s'ha necessitat un esforç addicional per conèixer la tecnologia i així avançar en el desenvolupament del projecte.

El projecte no està enfocat en la construcció d'una solució informàtica clàssica, sigui programa o disseny d'una infraestructura física, que doni resposta exacta a una necessitat concreta. Així les parts bàsiques de la metodologia de construcció de solucions, el que podem definir com projecte informàtic, no estan contemplades únicament al tractar-se d'un estudi sobre tecnologies de la informació i comunicació. En aquest cas el projecte està dissenyat per conèixer una aplicació, no per dissenyar-la i construir-la, tot i que al final el resultat sí és una solució viable d'infraestructura. Així podem establir paral·lelismes amb el cicle de vida d'un projecte per desenvolupar solucions, ja que de forma indirecta serà necessari construir un escenari de prova on comprovar el que s'ha entès de l'aplicació, i extreure conclusions del seu funcionament. L'enfocament es divideix en dos fases aproximadament paral·leles:

- Conèixer amb profunditat necessària les característiques tècniques del gestor MESOS, on la metodologia bàsica és la recerca de documentació i estudiar totes les característiques possibles.

- Segons les característiques de recursos que es puguin observar, dissenyar un escenari de proves. Aquest escenari serveix tant per fer funcionar l'aplicació com per verificar el seu funcionament, segons el que indiquen els manuals tècnics.

La unió de les dos fases permet crear un cert cicle de vida en cascada que consta de les següents fases:

- Estudi i anàlisi, que correspon a la primera fase de les indicades

- Disseny d'un sistema de prova, que correspon a la segona fase de les indicades

- Implementació del sistema de prova

- Execució de les proves

- Anàlisi dels resultats obtinguts

- Conclusions en relació a l'aplicació analitzada

El cicle de vida en cascada s'adapta a aquestes demandes i característiques. Permet definir les etapes clàssiques com l'estudi d'oportunitat (que fer) i l'anàlisi (quines característiques són necessàries). Per facilitar el tractament del cicle de vida s'ha fet servir un diagrama de Gantt, que s'estudia en les següents seccions, on s'especifiquen les durades de les tasques, i les precedències corresponents, que ens han d'assegurar l'èxit en la implementació del projecte.

Finalment la realització de la memòria exigeix un gran rigor documental, de forma que es pugui documentar les tasques realitzades a cada moment, els problemes trobats, la planificació i les desviacions d'aquesta així com les referències i informacions tècniques que s'han consultat. Per aquest motiu s'han disposat d'apunts, gairebé diaris, de l'avançament del projecte, i la memòria s'ha construït a partir d'aquests i la guia que disposem com a material didàctic.

L'enfocament del projecte per tant es similar al procés d'adquisició d'un sistema d'informació per a una organització, i això defineix la metodologia a utilitzar, diferent del projecte informàtic clàssic de desenvolupament ja que es tracta d'un projecte d'avaluació tecnològic.

1.5 Planificació teòrica del projecte

Els objectius del projecte presentats a la secció 1.3, ja indiquen quines seran les fites principals:

- Obtenir coneixement tangible sobre la gestió de clústers de computació.

- Conèixer el gestor de clústers MESOS.

- Implementar un clúster real com a prova de concepte, sobre el que s'executarà de forma pràctica la part teòrica anterior.

- Definir una metodologia d'avaluació d'aquest sistema, de forma que sigui una guia útil en l'estudi de sistemes distribuïts.

- Documentar a la memòria del projecte les conclusions i coneixements adquirits.

La consecució d'aquestes fites es projecta amb una divisió de tasques que ha de finalitzar com a màxim el dia de lliurament del projecte, establert pel trenta de desembre d'aquest any.

1.5.1 Projecte. Divisió en tasques.

A continuació es presenta una divisió en tasques i subtasques que han de permetre completar els objectius enunciats del projecte. De forma resumida es presenta la divisió en fases principal:

1. Recerca i estudi sobre l'estat de l'art respecte als clústers de computació.
2. Plataforma d'execució del clúster privat sobre el que es faran proves.
3. Recerca i estudi d'informació sobre MESOS.
4. Recerca i estudi d'informació sobre treball distribuït a la plataforma MESOS.
5. Recerca i estudi d'informació sobre metodologia d'estudi sobre sistemes distribuïts.
6. Recerca i estudi d'informació sobre l'entorn d'execució triat.
7. Anàlisis de la instal·lació de MESOS sobre un conjunt de milers de nodes.
8. Conclusions de l'estudi.

El treball anterior està dividit en tres grans blocs:

- Estudi general sobre clústers.

- Estudi de metodologies d'avaluació de sistemes distribuïts.

- Estudi sobre el gestor de clústers MESOS.

El primer bloc s'ha pensat per completar coneixement des de la part més bàsica, pel meu desconeixement al funcionament de clústers de computació, abans d'abordar el que significa un gestor de clústers com MESOS. D'aquesta forma s'espera obtenir un cert domini sobre el llenguatge tècnic necessari. De forma paral·lela s'ha decidit dedicar un temps inicial per depurar el maquinari del clúster, per què si no disposava d'un mínim de maquinari s'hauria d'abandonar l'objectiu de fer la implementació del clúster privat.

El segon bloc consistirà en dissenyar una metodologia d'avaluació de sistemes gestors de clústers que utilitzen l'execució distribuïda. Aquesta metodologia s'espera aplicar al bloc següent.

Una vegada superats els dos blocs anteriors entrarem al tercer bloc de treballs, el més extens, ja dedicat a la plataforma MESOS. Primer en el seu estudi bàsic d'instal·lació, funcionament i administració. El següent pas serà l'execució d'alguna aplicació, de forma que es permeti copsar l'execució distribuïda i la compartició de recursos, i per aquest motiu s'haurà de triar un entorn en funció del que es desitgi executar. A continuació s'aplicarien els coneixements d'avaluació apresos al segon bloc d'estudi, per avaluar l'execució distribuïda a la plataforma. La part final d'aquest

bloc compacte dedicat a MESOS és presentar de forma teòrica com s'implementaria un clúster real, analitzant els seus avantatges i diferències sobre el clúster privat utilitzat en el treball.

A continuació s'analitza cada tasca i subtasca de la divisió presentada a l'inici d'aquest punt, fixant la seva durada temporal.

1.5.1.1 Recerca i estudi sobre l'estat de l'art respecte als clústers de computació

En aquesta fase del projecte s'espera descriure les bases de disseny de clústers i la seva administració. En aquest sentit s'espera fer un resum històric i el punt actual, tant a nivell de maquinari com de sistemes de gestió. Tasques a desenvolupar:

- Recerca d'informació.
- Crear un document resum sobre la situació actual per aportar a la memòria del projecte.

La projecció temporal d'aquesta fase és la següent:

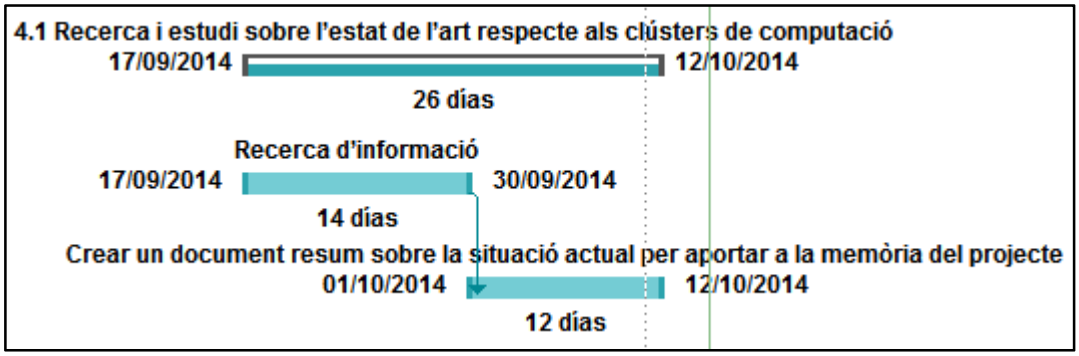


Figura 1. Planificació temporal de Recerca i estudi sobre l'estat de l'art respecte als clústers de computació.

En total la seva duració s'estima en vint-i-sis dies.

1.5.1.2 Plataforma d'execució del clúster privat sobre el que es faran proves

El clúster privat haurà de funcionar sobre un parc de màquines recuperades, una mica antigues, que s'han pogut aconseguir de forma gratuïta per executar MESOS. No es tracta de construir un clúster de màxima operativitat i rendiment, sinó una prova de concepte i de demostració que pugui ser escalable a màquines molt més potents. En aquesta fase del projecte es tracta de fer operatives almenys tres màquines (esclaus), revisar el seu funcionament i instal·lar un sistema operatiu que permeti operar al gestor. També la de configurar elements d'una xarxa dedicada al clúster. Finalment també es configurarà almenys una màquina virtual en el PC personal, que es constituirà com a màster del sistema. Com es pot preveure és una fase molt pràctica on s'hauran de solucionar problemes de maquinari i d'instal·lació de sistemes. Tasques a desenvolupar:

- Depuració de maquinari per tal d'aconseguir màquines per instal·lar el clúster.
- Definició de la xarxa local sobre la que funcionarà el clúster.
- Instal·lació del sistema operatiu necessari a tots els nodes.
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase és la següent:

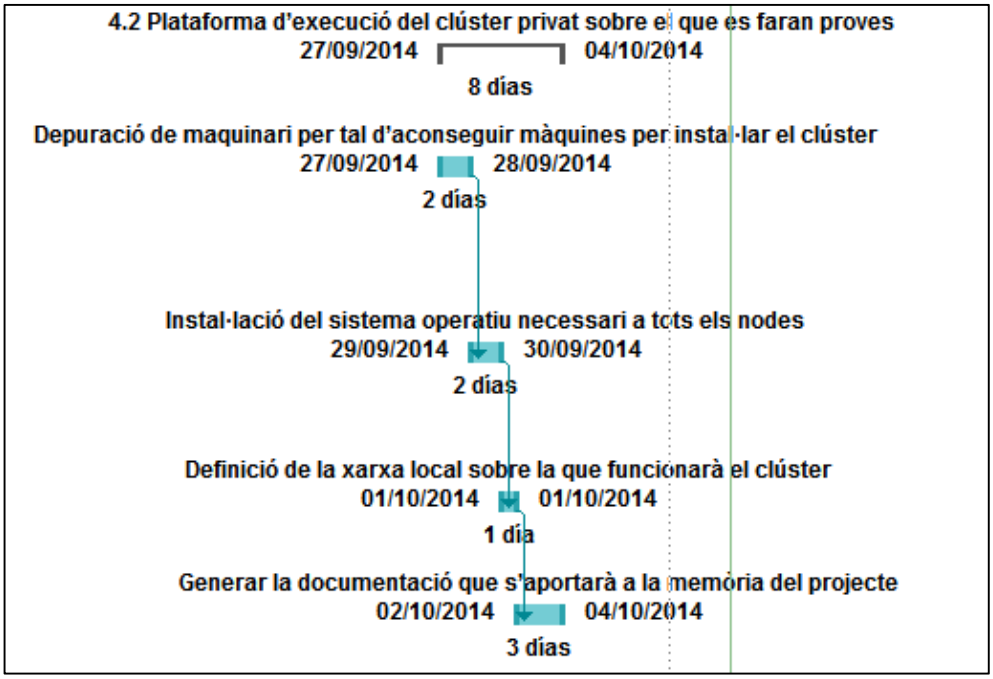


Figura 2. Planificació temporal de Plataforma d'execució del clúster privat sobre el que es faran proves.

En total la seva duració s'estima en vuit dies.

1.5.1.3 Recerca i estudi d'informació sobre MESOS

Aquesta fase es dedicarà a comprendre profundament el funcionament i necessitats operatives de la plataforma MESOS, la seva configuració i execució estable, mesures de rendiment o control, i les bases de l'administració del clúster. Es combinaran doncs una important part teòrica de coneixement del gestor, amb la implantació pràctica al clúster privat. La configuració final serà de tres nodes esclaus en màquina real i un màster en màquina virtualitzada. Les tasques a desenvolupar:

- Que és MESOS i quins components el componen?
- Instal·lació i execució dels components de la plataforma als nodes esclaus i mestre.
- Administració bàsica de la plataforma MESOS.
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase és la següent:

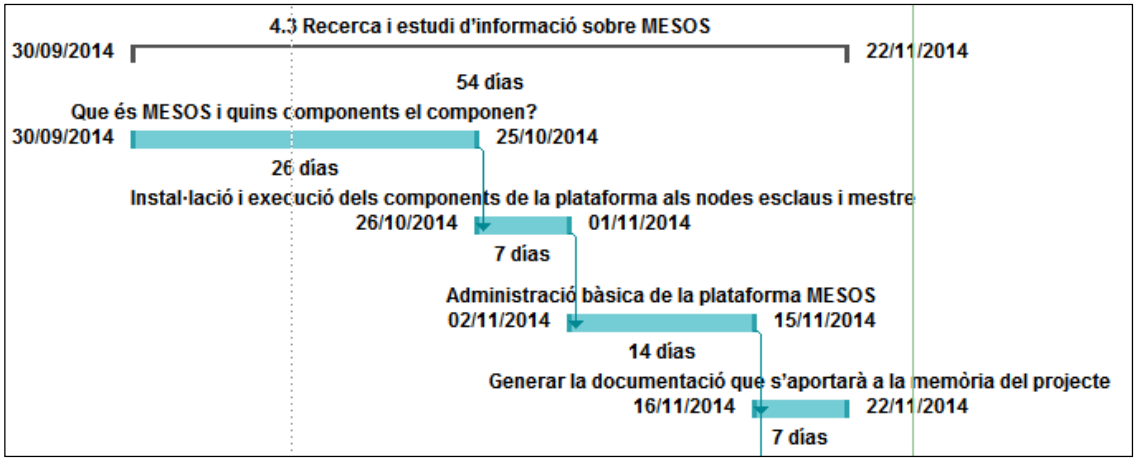


Figura 3. Planificació temporal de Recerca i estudi d'informació sobre MESOS.

En total la seva duració s'estima en cinquanta-quatre dies.

1.5.1.4 Recerca i estudi d'informació sobre treball distribuït a la plataforma MESOS

Les fases anteriors han servit per fixar una plataforma de maquinari vàlida, combinant PC, sistemes operatius i xarxa local, i també per instal·lar el gestor del clúster MESOS, objectiu principal del projecte. Arribat a aquest punt tindrem el clúster operatiu, però no podrem executar una aplicació de forma distribuïda. En aquest punt s'inicia l'estudi dels sistemes i processos que permeten l'execució, a un nou nivell de compartició de recursos, a nivell d'aplicació i de forma distribuïda. L'objectiu doncs d'aquesta fase es comprendre com MESOS pot arribar a compartir recursos entre nodes per augmentar l'eficiència global del sistema, i també analitzar les possibles limitacions. El punt final d'aquesta fase serà el triatge d'un entorn per a continuar amb una demostració de l'execució distribuïda al clúster. Les tasques a desenvolupar:

- Entorns disponibles per a la plataforma MESOSPHERE.
- Triatge d'almenys un entorn per fer proves distribuïdes.
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase és la següent:

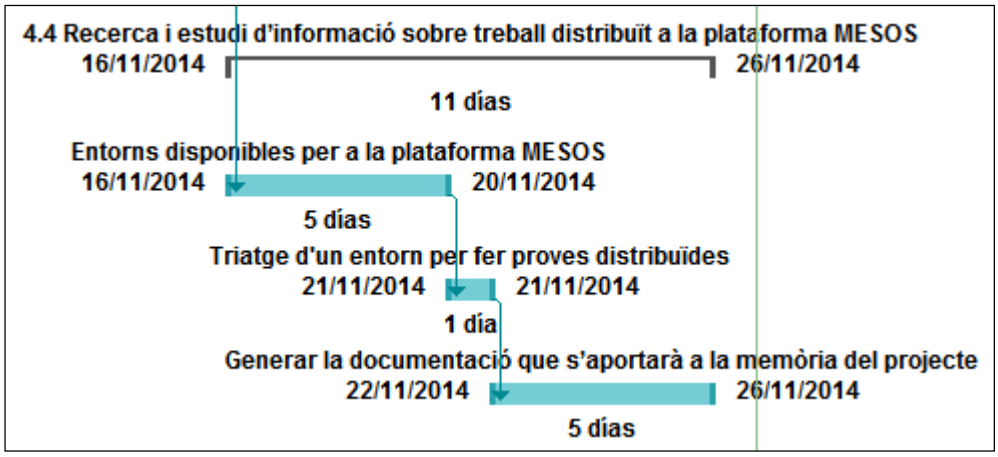


Figura 4. Planificació temporal de Recerca i estudi d'informació sobre treball distribuït a la plataforma MESOS.

En total la seva duració s'estima en onze dies.

1.5.1.5 Recerca i estudi d'informació de metodologia sobre sistemes distribuïts

En aquesta fase s'espera dissenyar una metodologia d'avaluació sobre l'eficiència de compartir recursos en clústers on l'execució és distribuïda. El resultat d'aquesta metodologia s'aplicarà en la següent fase, quan s'executi un prova d'execució o mètrica (*benchmark*[e]) i l'objectiu principal és veure les capacitats d'autogestió del clúster pel que fa als recursos de sistema disponibles (oferiment, reserva, regeneració, etcètera). Les tasques a desenvolupar:

- Recerca d'informació
- Disseny d'una metodologia d'avaluació aplicable

[e] Prova de mesura del rendiment d'un sistema.

La projecció temporal d'aquesta fase és la següent:

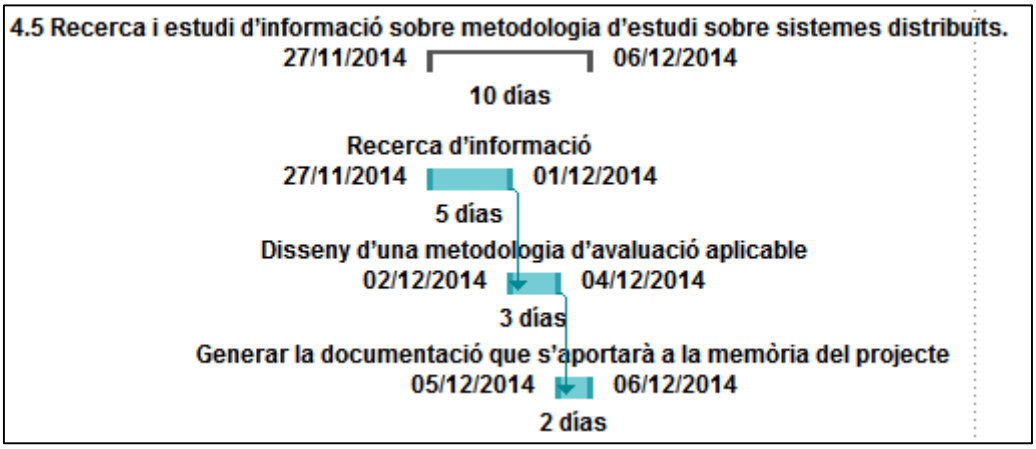


Figura 5. Planificació temporal de Recerca i estudi d'informació sobre metodologia d'estudi sobre sistemes distribuïts.

En total la seva duració s'estima en deu dies.

1.5.1.6 Estudi sobre MESOS en l'entorn triat

La fase anterior ha servit per fixar el coneixement sobre com es desenvolupa l'execució distribuïda a MESOS, també per triar un entorn per fer proves. En aquesta fase s'espera instal·lar el entorn i executar diferents proves sobre el clúster ja totalment operatiu. Les proves inclouran fallades de nodes i també s'espera mesurar l'eficiència del clúster, amb més o menys recursos compartits i disponibles. És per tant la conclusió de l'estudi sobre el clúster privat creat a l'efecte, des de la seva configuració i instal·lació inicial. Les tasques a desenvolupar:

- Instal·lar el entorn sobre la plataforma MESOS.
- Executar aplicacions de prova per demostrar les capacitats distribuïdes de MESOS aplicant la metodologia dissenyada al punt 4.5
  - Tolerància a errors
  - Eficiència i distribució dinàmica
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase és la següent:

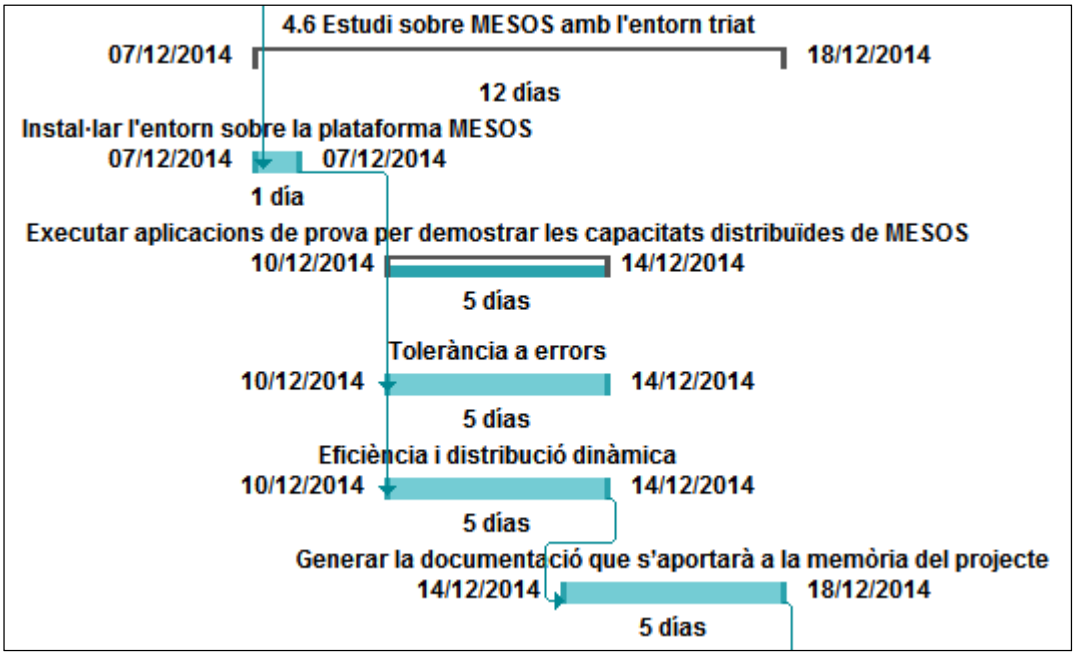


Figura 6. Planificació temporal de Estudi sobre MESOS amb l'entorn triat.

En total la seva duració s'estima en dotze dies.

1.5.1.7 Anàlisis de la instal·lació de MESOS sobre un conjunt de milers de nodes

Superada la fase pràctica del projecte, en aquest punt l'objectiu principal seria el com projectar el coneixement adquirit en una instal·lació massiva de nodes. Un clúster on per exemple factors com la fallada de màsters i nodes pot ser crítica, també l'eficiència del sistema o el tipus d'informació que el clúster ha de tractar, que serien pilars bàsics del seu disseny. Es per tant trobar les diferències bàsiques entre un clúster privat petit, com l'utilitzat, i un clúster real de producció. En aquest sentit també es vol presentar alguna solució distribuïda de proves per a enginyeria, i que estan disponibles com a servei de pagament. Les tasques a desenvolupar:

- Com s'implementaria?
- Estudi diferencial amb la solució pràctica mínima desenvolupada.
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase és la següent:

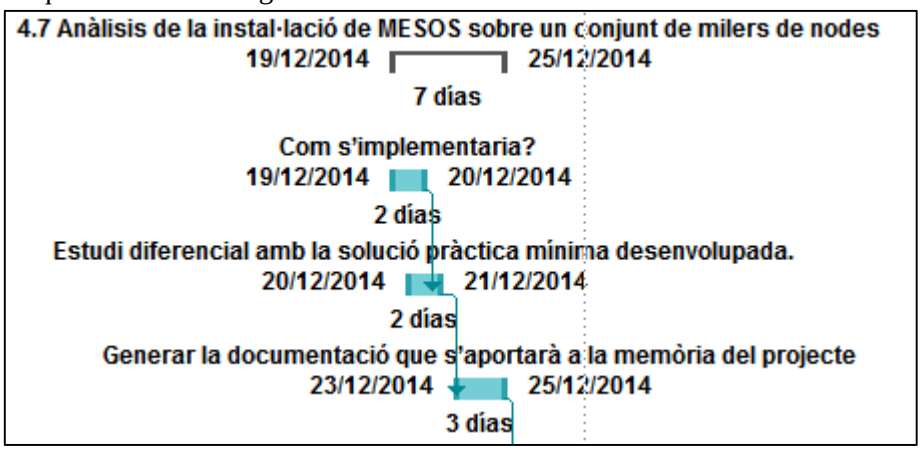


Figura 7. Anàlisi de la instal·lació de MESOS sobre un conjunt de milers de nodes.

En total la seva duració s'estima en set dies.

1.5.1.8 Conclusions de l'estudi.

La part final de qualsevol treball d'investigació finalitza amb les conclusions personals que s'han pogut extreure de tot el treball realitzat. En aquesta fase finalitza la recerca i implementació del projecte, i es completa la seva memòria.

La projecció temporal d'aquesta fase és la següent:

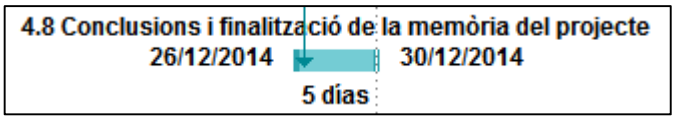


Figura 8. Conclusions i finalització de la memòria del projecte.

En total la seva duració s'estima en cinc dies.

En totes les fases del projecte es generarà documentació pròpia de forma que s'alimenti la memòria del projecte. Aquesta documentació ha de recollir el coneixement creat, les fites aconseguides i les dificultats trobades, i si aquestes han tingut o no solució. També algunes fases s'executen en paral·lel, per l'evident encavalcament de conceptes a l'estudiar la documentació de la plataforma.

El següent diagrama de Gantt, figura 9, estableix l'ordre d'execució i la planificació temporal d'aquestes tasques:

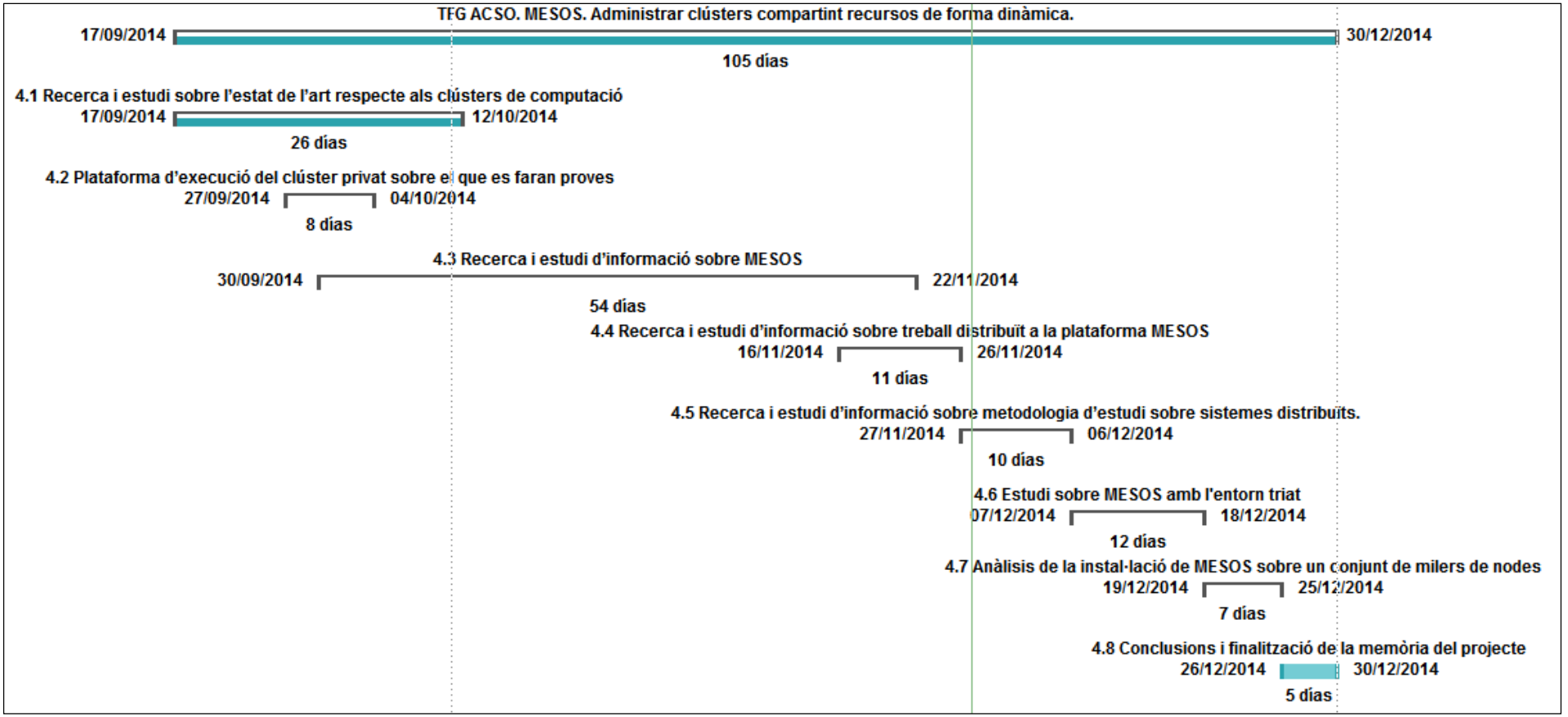


Figura 9. Planificació temporal del projecte MESOS. Administrar clústers compartint recursos de forma dinàmica..

En total la seva duració s'estima en cent cinc dies.

Taula resum temporal prevista

Tasca	Duració	Inici	Fi
TFG ACSO. MESOS. Administrar clústers compartint recursos de forma dinàmica.	105 dies	dc 17/09/14	dt 30/12/14
1.5.1.1 Recerca i estudi sobre l'estat de l'art respecte als clústers de computació	**26 dies**	**dc 17/09/14**	**dg 12/10/14**
Recerca d'informació	14 dies	dc 17/09/14	dt 30/09/14
Crear un document resum sobre la situació actual per aportar a la memòria del projecte	12 dies	dc 01/10/14	dg 12/10/14
1.5.1.2 Plataforma d'execució del clúster privat sobre el que es faran proves	**8 dies**	**ds 27/09/14**	**ds 04/10/14**
Depuració de maquinari per tal d'aconseguir màquines per instal·lar el clúster	2 dies	ds 27/09/14	dg 28/09/14
Instal·lació del sistema operatiu necessari a tots els nodes	2 dies	dl 29/09/14	dt 30/09/14
Definició de la xarxa local sobre la que funcionarà el clúster	1 dia	dc 01/10/14	dc 01/10/14
Generar la documentació que s'aportarà a la memòria del projecte	3 dies	dj 02/10/14	ds 04/10/14
1.5.1.3 Recerca i estudi d'informació sobre MESOS	**54 dies**	**dt 30/09/14**	**ds 22/11/14**
Que és MESOS i quins components el componen?	26 dies	dt 30/09/14	ds 25/10/14
Instal·lació i execució dels components de la plataforma als nodes esclaus i mestre	7 dies	dg 26/10/14	ds 01/11/14
Administració bàsica de la plataforma MESOS	14 dies	dg 02/11/14	ds 15/11/14
Generar la documentació que s'aportarà a la memòria del projecte	7 dies	dg 16/11/14	ds 22/11/14
1.5.1.4 Recerca i estudi d'informació sobre treball distribuït a la plataforma MESOS	**11 dies**	**dg 16/11/14**	**dc 26/11/14**
Entorns disponibles per a la plataforma MESOS	5 dies	dg 16/11/14	dj 20/11/14
Triatge d'un entorn per fer proves distribuïdes	1 dia	dv 21/11/14	dv 21/11/14
Generar la documentació que s'aportarà a la memòria del projecte	5 dies	ds 22/11/14	dc 26/11/14
1.5.1.5 Recerca i estudi d'informació sobre metodologia d'estudi sobre sistemes distribuïts.	**10 dies**	**dj 27/11/14**	**ds 06/12/14**
Recerca d'informació	5 dies	dj 27/11/14	dl 01/12/14
Disseny d'una metodologia d'avaluació aplicable	3 dies	dt 02/12/14	dj 04/12/14
Generar la documentació que s'aportarà a la memòria del projecte	2 dies	dv 05/12/14	ds 06/12/14
1.5.1.6 Estudi sobre MESOS amb l'entorn triat	**12 dies**	**dg 07/12/14**	**dj 18/12/14**
Instal·lar l'entorn sobre la plataforma MESOS	1 dia	dg 07/12/14	dg 07/12/14
Executar aplicacions de prova per demostrar les capacitats distribuïdes de MESOS	5 dies	dc 10/12/14	dg 14/12/14
Tolerància a errors	5 dies	dc 10/12/14	dg 14/12/14
Eficiència i distribució dinàmica	5 dies	dc 10/12/14	dg 14/12/14
Generar la documentació que s'aportarà a la memòria del projecte	5 dies	dg 14/12/14	dj 18/12/14
1.5.1.7 Anàlisis de la instal·lació de MESOS sobre un conjunt de milers de nodes	**7 dies**	**dv 19/12/14**	**dj 25/12/14**
Com s'implementaria?	2 dies	dv 19/12/14	ds 20/12/14
Estudi diferencial amb la solució pràctica mínima desenvolupada.	2 dies	ds 20/12/14	dg 21/12/14
Generar la documentació que s'aportarà a la memòria del projecte	3 dies	dt 23/12/14	dj 25/12/14
1.5.1.8 Conclusions i finalització de la memòria del projecte	**5 dies**	**dv 26/12/14**	**dt 30/12/14**

Taula 1. Resum temporal previst.

1.6 Descripció de l'estructura de desenvolupament del projecte

A partir d'ara els apartats ja estaran orientats a la solució dels objectius del projecte. De forma resumida es presenta ara el contingut de cadascun d'ells.

Apartat 2. Estat de l'art
Es tractaran les tecnologies actuals de gestió dinàmic de recursos als clústers, de forma específica la comparació de tecnologies monolítiques, de dos capes (MESOS) i d'estat compartit (OMEGA). També el que poden aportar els contenidors Docker a la virtualització.

Apartat 3. Descripció tècnica dels components del sistema
En aquest apartat es mostra un estudi descriptiu sobre MESOS i el seu funcionament. La segona part fa una introducció de la llibreria MPI, ja que l'entorn d'aplicació triat per ser executat sobre la capa MESOS és el que permet executar treballs paral·lels amb aquesta llibreria.

Apartat 4. Implementació real d'un clúster administrat per MESOS
Aquest apartat fa tot el desenvolupament necessari per instal·lar físicament un clúster privat, instal·lar l'administració MESOS i també el programari necessari per a executar aplicacions MPI. Al final d'aquest apartat tindrem un clúster MPI totalment funcional.
També es tracta de forma introductòria tres possibilitats actuals *IaaS* per executar la tecnologia MESOS.

Apartat 5. Obtenció de mètriques sobre el clúster MESOS
El producte de l'apartat anterior és un clúster funcional MPI amb administració de recursos MESOS, en aquest punt del projecte s'ha pensat estudiar el rendiment del clúster des de dos vessants:

- L'eficiència del pas de missatges MPI, quan s'introdueix l'administració MESOS (clúster MESOS) i quan aquesta administració no està present (clúster MPD).

- L'eficiència d'execució d'una aplicació MPI, quan s'introdueix l'administració MESOS (clúster MESOS) i quan aquesta administració no està present (clúster MPD).

Apartat 6. Valoració econòmica
Als projectes sempre s'ha de fer una valoració de costos, tant en esforç d'execució com de materials. Aquesta part tracta aquesta temàtica, presentant tant l'esforç calculat com la desviació real produïda.

Apartat 7. Conclusions del projecte
Aquest apartat tanca el projecte presentant conclusions sobre el treball realitzat, objectius aconseguits o no i per què, autoavaluació, problemes trobats en el desenvolupament i objectius interessants d'abordar. És un apartat resum de tot el treball realitzat i el seu anàlisi formal.

2. Estat de l'art

En aquest apartat farem una breu descripció de la situació tecnològica dels gestors de clústers i la possible evolució a curt termini. Aprofitarem per fer una comparació de MESOS amb d'altres solucions de gestió.

2.1 Gestors eficients de recursos als clústers de computació

Les necessitats actuals creixents com la mida i la rapidesa de canvi davant dels nous requeriments, fan que els clústers monolítics no siguin els més adequats i comencin ha sorgir nous plantejaments tecnològics, basats en paral·lelisme i mecanismes de concurrència. Una dada molt important ha tenir en compte és que els clúster de gran escala, formats per centenars o milers de nodes de computació, són molt cars i s'ha de maximitzar la seva utilització i eficiència. Una forma de fer-ho és ocupar amb el màxim de càrrega de treball als nodes, encara que aquesta càrrega no sigui homogènia a nivell d'ús de recursos. D'aquesta manera és possible fins i tot reduir els recursos utilitzats, ja que l'objectiu és arribar al màxim de rendiment de cada component. El problema sorgeix en que és necessari utilitzar un planificador prou eficient, que sigui capaç d'abordar l'organització de recursos i el llançament de tasques, es a dir assignar el treball a executar a cada node. A més el planificador ha de ser prou flexible per també ser escalat al mateix temps que els recursos, sinó passarà a perjudicar el rendiment del clúster. Alguns dels punts a considerar en la planificació de recursos són els següents:

- Alta utilització de recursos.

- Presa ràpida de decisions i orientades al negoci, si és possible amb diferents graus de distribució.

- Ha de configurar un sistema robust i sempre disponible.

Per complir els objectius del planificador s'han de tenir en compte alguns dels següents punts:

- Dividir la tasca a planificar entre diferents planificadors. La tasca es pot distribuir pels planificadors segons una distribució de càrrega o bé definir unes parts específiques, o la combinació de totes dues.

- Tria dels recursos a utilitzar, que poden ser tots o bé una part determinada i limitada. També poden existir recursos preferents, per exemple d'execució el màxim de local, enfront d'una tria de recursos disponibles qualsevols.

- Interferències. Es produeixen davant d'una competició pels recursos disponibles, i passa quan dos planificadors trien el mateix recurs. Hi han dos aproximacions:

    - Pessimista. S'evita el problema al limitar un recurs a un únic planificador i tasca alhora.

    - Optimista. En cas de conflicte soluciona per detecció i traspàs del recurs sol·licitat. Facilita el paral·lelisme però només si el nivell de conflicte és baix.

- Assignació de detall. Els treballs estan composats de moltes tasques, de forma que els planificadors han de triar una estratègia per poder planificar-los correctament, que pot ser atòmica llançant les tasques quan disposin de tots els recursos, o bé anar assignant recursos lliures a les tasques i executar-les en una mena d'execució incremental.

- Comportament del clúster. Els treballs a executar poden a la vegada iniciar d'altres tasques segons la importància del treball, fent que apareguin noves tasques de planificació. El comportament es pot limitar si

s'estableix un control estricte i centralitzat, per evitar per exemple l'execució de múltiples treballs de prioritat similar que poden esgotar els recursos, i per tant fer perdre quin ha de ser el comportament d'execució.

A nivell d'arquitectura es poden distingir tres tipus de planificador, segons es mostra a la figura 10:

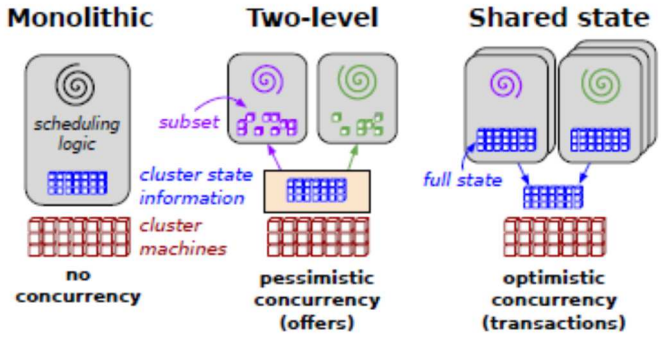


Figura 10. Arquitectures de planificadors: monolítica, de dos nivells (MESOS) i d'estat compartit (OMEGA).

- Els planificadors monolítics (*Monolithic scheduler*) utilitzen un únic algorisme de planificació per a tots els treballs. És una aproximació utilitzada als HPC[f], de forma que un planificador executa una única instància del codi de planificació i aplica el mateix algorisme a tots els treballs que rep. Tot i que es poden aplicar algunes millores de priorització com assignar una ponderació als treballs, o bé tractar de forma diferent el treball assignat segons una planificació.

- El planificadors de dos nivells (*Two-level schedulers*), utilitzen un únic gestor de recursos que ofereix recursos a múltiples entorns d'aplicació que s'executen en paral·lel. Es a dir se subministren recursos de forma dinàmica, i un coordinador central decideix quants. En aquest grup es trobaria MESOS, tecnologia que estudiem en aquest projecte, i on els recursos són distribuïts en forma d'ofertes, que són aquells realment disponibles. Els conflictes es resolen per què el coordinador només ofereix recursos als entorns de planificació un per un, de forma que el recurs està bloquejat de forma pessimista fins que es resol si l'entorn d'aplicació necessita o no el recurs. Els recursos en oferta formen part del grup de disponibles en tot el clúster, així no és possible establir una política de prioritats de recursos.

- Planificador d'estat compartit (*Shared state scheduler*), que està basat en un control de concurrència optimista i sense bloquejos, de forma que es té accés a tots els recursos del clúster de forma simultània. No existeix un coordinador central, totes les decisions respecte als recursos són preses als planificadors, que tenen accés a un registre d'estat global (*cell state*), disposa de còpia privada i local, de tots els recursos del clúster. La forma d'adquirir un recurs és reclamar-lo, sempre que tingui els permisos per fer-ho, i el podrà assignar si té més prioritat que un altre planificador en cas d'interferència. L'assignació serà una operació atòmica i quedarà reflectida al registre d'estat. Així podem tenir múltiples planificadors treballant en paral·lel per assignar recursos a tasques, que poden ser executades en bloc per solucionar un determinat treball. Per permetre un comportament del clúster també és possible definir una ponderació als treballs, per marcar precedències d'execució. Aquesta tecnologia és la que implementa OMEGA el nou gestor de clústers de Google.

Com a conclusió de les tecnologies analitzades, trobem que la tecnologia de planificador monolític no facilita afegir noves polítiques de gestió ni implementacions, tampoc escala de forma adequada amb la mida del clúster, i és difícil de mantenir sobre tot si existeixen prioritzacions. El planificador de dos nivells que ofereix MESOS millora la gestió de recursos, però té el risc de no tenir una visió global dels recursos i utilitza un mecanisme d'oferta pessimista, de forma que bloqueja recursos i el pot fer perdre eficiència. La tecnologia d'OMEGA sembla superar a les dos

[f] HPC: High Performance Computing. Computació d'altes prestacions.

aproximacions anteriors pel control de concurrència utilitzat i una visió global dels recursos amb mecanisme d'oferta optimista, vàlid si el nivell de conflicte els recursos és baix.

2.2 Tecnologia de virtualització per aïllament. Contenidors DOCKER

L'aparició de DOCKER[2] com a gestor basat en contenidors està provocant una petita revolució en l'àmbit de la virtualització[3]. La presentació a l'OSCON[g] al juliol d'aquest any ha fet que molts fabricants comencin a plantejar-se com virtualitzen (VM[h]) el seus sistemes. Els sistemes clàssics de virtualització basats en *hypervisors* (Xen, Hyper-V i KVM) funcionen emulant maquinari, i això els fa grans consumidors de recursos. Per l'altra banda la virtualització clàssica té l'avantatge de què es poden utilitzar a cada VM un sistema operatiu diferent. Així la principal diferència amb els contenidors és que aquests últims utilitzen el mateix sistema operatiu de base, el gestor de contenidors és una capa entre les aplicacions i llibreries i el sistema operatiu. D'aquesta forma es permet compartir recursos comuns del sistema operatiu i llibreries a les diferents aplicacions, que es troben en un contenidor independent i aïllat de les altres. Per tant es produeix un gran estalvi de recursos per què és comparteix el sistema operatiu base.

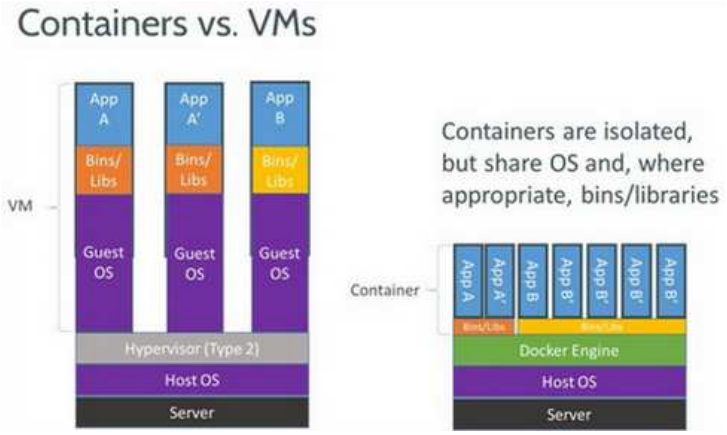


Figura 11. Arquitectura de contenidors comparada en màquines virtuals (VM).

La idea de gestió basada en contenidors no és nova, té els seus inicis a l'any 2000 i ha rebut diferents noms com *FreeBSD*[i] o *Jails* de Solaris. Google i Docker també han treballat en solucions *open source* com OpenVZ i LXC (*Linux Container*). De fet quan s'executa una aplicació Google s'està creant un contenidor de suport. En el cas de contenidors Docker aquests s'executen en LXC, i la diferència principal amb els *hypervisors* és que només es fa una abstracció del *kernel* del sistema operatiu, no de tot el sistema, però conserva totes les característiques necessàries de la virtualització (RAM, sistema de fitxers, processador, etcètera). Una aplicació directa, de menys cost de recursos en comparació amb VM, és per exemple quan es volen executar aplicacions que tenen de base el mateix sistema operatiu. Si es vol fer amb VM s'han de crear tantes com aplicacions es vulguin utilitzar, per tant el desplegament en contenidors permet extreure més dels equips al no necessitar de tants recursos de màquina.

Una altra avantatge de l'ús de contenidors LXC és que es tenim portabilitat de forma immediata, facilitant també el seu desplegament al núvol, on pot convertir-se en un gestor del seu entorn actuant com a un servidor. També proporciona l'aïllament entre aplicacions que s'executen al mateix *host*. Alguns dels usos de contenidors poden ser:

- Distribució ràpida d'aplicacions, només cal introduir el contenidor a l'entorn d'execució, el contenidor inclou l'aplicació a executar.

- Instal·lació i escalat més ràpid. Pot instal·lar-se en molts escenaris: màquines reals, virtuals o al núvol.

[g] Open Source Convention. Aquest any celebrada Portland. http://www.oscon.com/oscon2014/public/schedule/detail/33627
[h] VM: Virtual Machine
[i] https://www.freebsd.org/

- Aconseguir un major aprofitament de recursos, el contenidor és l'aplicació i llibreries de suport no tot un sistema emulat, i més càrrega de treball.

L'arquitectura del sistema és del tipus client-servidor, on un dimoni client fa les consultes a un dimoni servidor Docker, que s'encarrega de tots els treballs del contenidor (construcció, funcionament i distribució de contenidors). Les comunicacions són mitjançant *sockets* o l'API RESTful[j]. A la figura 12 podem veure els dos dimonis principals, de forma que l'usuari interactua amb el dimoni Docker client i aquest amb el dimoni servidor Docker.

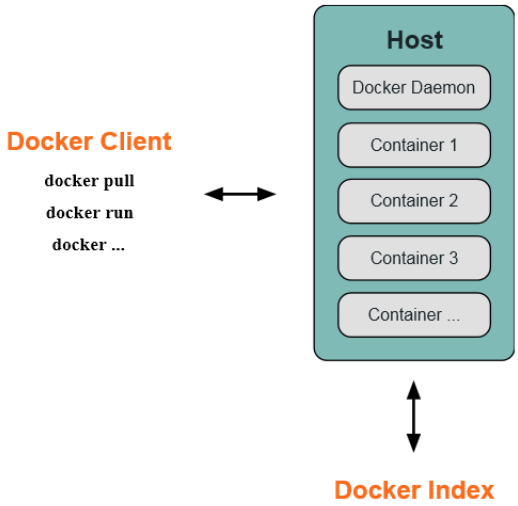


Figura 12. Interacció entre els dimonis de gestió d'un contenidor DOCKER.

L'arquitectura interna de Docker està dividida en els components següents:

- Imatges Docker. Es tracta del component que construeix el contenidor, bàsicament són plantilles de només-lectura (*read only*), que defineixen com està constituït el contenidor (per exemple: sistema operatiu i l'aplicació d'interès a executar). Internament funcionen a nivell de capes utilitzant UnionFS[k4], que fan que els canvis només afectin a nivell de capa. La construcció d'un nou contenidor comença per la tria de la imatge base, i la modificació de les seves capes per incloure una de les següents instruccions:

    - Executar una comanda.
    - Afegir un fitxer o un directori .
    - Crear una variable d'entorn.
    - Definir quin procés s'executarà quan s'activi la imatge del contenidor.

  Les instruccions es guarden en un fitxer anomenat *Dockerfile*, de forma que quant es construeix la imatge relacionada, les comandes que inclou són executades abans de tornar a la imatge resultat.

- Registres Docker. Es tracta del component de distribució, ja que conté les imatges Docker, que poden estar en repositoris públics o privats. Es pot trobar a http://hub.docker.com, i també podem publicar també les nostres pròpies imatges, només cal enviar-les al servei.

- Contenidors Docker. Aquest component és similar a un directori, conté tot el necessari per fer funcionar una aplicació, amb els avantatges d'aïllament i seguretat. Així el contenidor és el component d'execució

[j] Abstracció de l'arquitectura WWW per caracteritzar serveis Web.
http://en.wikipedia.org/wiki/Representational_state_transfer
[k] UnionFS: sistema de fitxers que permet la convivència de diferents sistemes operatius, cadascun separat en una branca (branche) i que poden estar superposades (un sistema de prioritats defineix quina té més prioritat davant d'un cnflicte per nom de fitxer), i que formen un sistema de fitxers coherent.

d'imatges Docker, que són de només lectura, però que llavors utilitzant *l'UnionFS* afegeix una capa d'escriptura per permetre l'execució.

Docker està escrit en *Go*[l] i la tecnologia que permet l'execució està basada en el *kernel* de Linux, algunes característiques rellevants:

- Espai de noms (*namespaces*). Permet la creació del contenidor, i l'espai de noms és exclusiu i limitant. D'aquesta característica s'obté l'aïllament i el no permetre cap l'execució fora d'ell.

- Grups de control (*cgroups*). Aquesta característica permet el control de l'ús dels recursos per part de cada contenidor, permetent una convivència pacífica i si és necessari limitant l'accés.

UnionFS. Aquest component ja l'hem avançat a la part de les imatges Docker, i com s'ha vist són les peces mínimes que componen els contenidors. El format de contenidor per defecte és *libcontainer*, tot i que s'accepten altres formats com LXC.

[l] Llenguatge de programació. https://golang.org/

3. Descripció tècnica dels components del sistema

En aquesta apartat aprofundirem en l'estudi teòric del gestor MESOS, també del significat d'entorn d'aplicació i triarem un dels possibles per poder fer totalment operatiu el clúster privat projectat.

3.1 MESOS. Plataforma de gestió de clústers

L'objectiu del projecte és la implementació d'un clúster privat amb recursos de baix cost, utilitzant la plataforma MESOS i realitzant un estudi d'eficiència. Aquesta plataforma ha estat desenvolupada de forma inicial des de la UC Berkeley[5], el primer document públic descriptiu és del trenta de setembre de 2010[6] i sobre ell està construïda la descripció d'aquest apartat de la memòria. En aquests moments MESOS és un projecte de col·laboració oberta[7], i a la pàgina de l'aplicació tenim les seves descripcions bàsiques:

- Programa contra el teu centre de dades com si fos un únic repositori de recursos.

- *Apache Mesos* abstrau CPU, memòria, emmagatzematge i altres recursos de computació més enllà de les màquines (físiques o virtuals, afavorint la distribució elàstica d'una forma senzilla i efectiva.

- Que és MESOS? Un sistema de *kernel* distribuït. MESOS està construït utilitzant els mateixos principis que el *kernel* de Linux, però en un nivell d'abstracció. El *kernel* de MESOS funciona en cada màquina i subministra entorns d'aplicació o *frameworks* (per exemple: Hadoop, Spark, Kafka, Elastic Search) amb API d'administració de recursos i programació de forma transversal i completa de centres de dades i aplicacions en el núvol. Capacitats:

    - Escalable fins 10.000 nodes.
    - Capacitat de recuperació d'errors, replicant màsters i esclaus, utilitzant Zookeeper.
    - Suport per contenidors DOCKER.
    - Aïllament de forma nativa entre tasques amb contenidors Linux.
    - Planificació multirecurs (memòria, CPU, disc i ports).
    - API Java, Python i C++ APIs, per desenvolupar noves aplicacions paral·leles.
    - Web UI per visualitzar l'estat del clúster.

MESOS estava implementat, en la seva primera versió, utilitzant 10.000 línies de codi C++. Funciona en sistemes LINUX, SOLARIS o MAC OS , i les aplicacions poden ser programades en C, C++, Java i Python. La llista actual d'entorns d'aplicació[8] (*frameworks*) és la següent:

Serveis de llarg durada

Es tracta bàsicament de planificadors de servei, que permeten un major control de l'execució de tasques al clúster MESOS. En aquest moment estan desenvolupats:

- **Aurora** és un planificador de servei que s'executa en la part superior de Mesos, el que li permet executar els serveis de llarga execució que s'aprofiten de l'escalabilitat MESOS, tolerància a fallades, i l'aïllament dels recursos.

- **Marathon** és un PaaS privat construït en MESOS. Maneja automàticament els errors de maquinari o programari i assegura que una aplicació estigui "sempre activa".

- **Singularity** és un programador (API HTTP i la interfície web) per a executar tasques MESOS: processos de execució de llarga durada, les tasques d'una sola execució, i els treballs programats.

- **SSSP** és una aplicació web simple que ofereix una etiqueta blanca "Megaupload" per emmagatzemar i compartir arxius en S3 (magatzem Amazon S3).

Processament Big Data

Aquests entorns permeten l'execució d'aplicacions a MESOS segons la seva especialitat. Un exemple és la computació paral·lela utilitzant Cray Chapel (llenguatge) o bé MPI (llibreries). En el cas del segon aquest projecte l'implementarà sobre llibreries MPICH2.

- **Cray Chapel** és un llenguatge de programació paral·lela productiva. El planificador Chapel Mesos li permet executar programes Chapel en MESOS.

- **Dpark** és un clon de Python de Spark, un entorn d'aplicació MapReduce escrit en Python, que s'executa en Mesos.

- **Exelixi** és un entorn d'aplicació marc distribuït per al funcionament dels algoritmes genètics a escala.

- **Hadoop** en MESOS distribueix treballs de MapReduce de manera eficient a través d'un clúster complet.

- **Hama** és un marc de computació distribuïda basada en tècniques paral·leles síncrones a granel de computació per als algoritmes de càlculs científics, per exemple matrius, gràfics i xarxes massives.

- **MPI** és un sistema de pas de missatges dissenyat per funcionar en una àmplia varietat de computadores paral·leles.

- **Spark** és un sistema informàtic ràpid i de propòsit general que facilita la escriptura de treballs paral·lels.

- **Storm** és un sistema de computació en temps real distribuït. Storm fa que sigui fàcil de processar de manera fiable els fluxos il·limitats de dades, fent per al processament en temps real el que Hadoop va fer per al processament per lots.

Planificació per lots

Igual que els serveis de llarga durada el següent són implementacions que ajuden al clúster MESOS a millorar la planificació de les tasques, tant a nivell de tolerància a fallades com a necessitats de recursos en certs moments puntuals.

- **Chronos** és un planificador de tasques distribuïda que suporta topologies de treball complexes. Pot ser utilitzat com un reemplaçament més tolerant a fallades per a Cron.

- **Jenkins** és un servidor d'integració contínua. El connector Mesos-Jenkins li permet posar en marxa de forma dinàmica els treballadors en un clúster Mesos en funció de la càrrega de treball.

- **JobServer** és un planificador de tasques distribuïda i de processador que permet als desenvolupadors construir Tasklets, de processament per lots personalitzats, utilitzant "apunta i fes clic" a la interfície d'usuari web.

- **Torque** és un gestor de recursos distribuïts que proporciona control sobre els treballs per lots i nodes de computació distribuïda.

Emmagatzematge de dades

Aplicacions de gestió de dades de gran dimensions també poden ser gestionades des de MESOS. Un exemple és Cassandra[9] que actualment està desplegat, tot i que no sobre MESOS, a Apple (75.000 nodes amb 10 PB de dades), Netflix (2.500 nodes, 420 TB), cercador xinés Easou (270 nodes, 300 TB), i eBay (100 nodes, 250 TB).

- **Cassandra** és una base de dades distribuïda eficaç i d'alta disponibilitat. La seva escalabilitat lineal i demostrada tolerància a fallades, en el maquinari de productes bàsics o infraestructura al núvol, fan que sigui la plataforma perfecta per a les dades de missió crítica.

- **ElasticSearch** és un motor de cerca distribuït. Mesos fa que sigui fàcil d'executar i escalar.

- **Hypertable** és un sistema d'emmagatzematge distribuït, d'alt rendiment, escalable, i de processament de dades estructurades i no estructurades.

Actualment MESOS està implementat en sistemes reals d'alta demanda, dos exemples de la seva utilització són:

- **TWITTER**. Instal·lat a centenars de màquines ajuda en l'execució de múltiples tasques, des de serveis fins càlculs analítics de les càrregues de treball[10].

- **Airbnb** també ha adoptat la plataforma des de juny de 2013, inicialment estava a EMR. Utilitzen els entorns d'aplicació Hadoop, Chronos i Storm[11].

3.1.1 Descripció del disseny

MESOS es presenta doncs com una plataforma que permet compartir els recursos d'un clúster a diferents aplicacions gràcies a l'ús d'entorns d'aplicació (*frameworks*), de forma que augmenta l'eficiència al no necessitar replicar les dades per a cada aplicació i també per què els recursos disponibles són utilitzats per l'aplicació que els necessita, a cada moment i de forma transparent. La forma de compartir recursos, la granularitat, és anomenada de gra fi, permetent arribar a nivell de màquina individual (lectura de dades per exemple). La planificació de les aplicacions es realitza en dos nivells, abstracció que anomena oferidors de recursos, de la següent forma:

- Nivell 1: MESOS decideix exactament quants recursos ofereix a cada entorn d'aplicació

- Nivell 2: Cada entorn d'aplicació decideix quants recursos accepta i que computacions fa sobre ells.

Per poder administrar les aplicacions que s'executen als clústers, per exemple serveis d'Internet o càlcul intensiu, són necessaris entorns d'aplicació que ajudin a simplificar la programació del clúster, ja que l'entorn d'aplicació fa d'interfície entre aplicació i MESOS en la comunicació de necessitats i recursos disponibles per resoldre-les, com es pot veure a la figura 13. Així es necessita un entorn d'aplicació per cada aplicació a executar al clúster, i es permet la convivència de nodes amb diferents tipus d'entorn d'aplicació. D'aquesta forma podem tenir un clúster heterogeni, mesclant diferents entorns d'aplicació per augmentar l'eficiència i disponibilitat dels recursos.

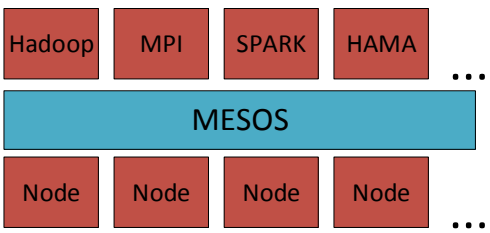


Figura 13. Abstracció MESOS desplegada en clúster.

Evolució tecnològica

La rapidesa de creixement d'aplicacions fa que no tots els entorns d'aplicació siguin òptims i a més les organitzacions volen aprofitar la inversió realitzada, de forma que un clúster no pot estar dedicat només a una aplicació, en el que s'anomena partició estàtica. És necessari permetre múltiple aplicació als clústers, de forma que augmenti la seva eficiència i millora l'accés a les dades que serien comunes a totes les aplicacions. Les solucions prèvies a MESOS tenen les següent característiques:

Servidors dedicats

Figura 14. Configuració de centre de dades amb servidors dedicats.

- Baix rati d'utilització, podem tenir un o varis servidors desbordats, pel tipus d'aplicació que executen, i altres servidors sense cap càrrega de treball.

- Es necessita molt temps per llançar nous serveis, possiblement s'ha d'estudiar les interaccions entre serveis de servidors diferents i per tant estudiar el llançament i afectacions a serveis, que no són els nous serveis a desplegar.

Virtualització

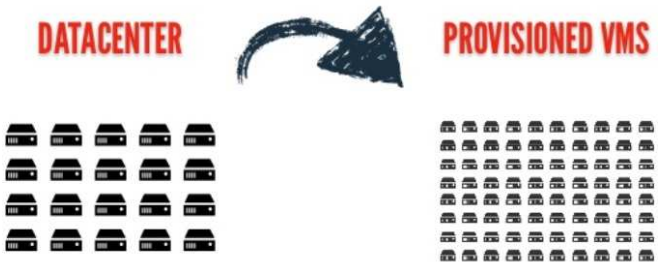


Figura 15. Configuració de centre de dades amb Virtualització (VM).

- Augmenta el total de màquines a administrar.

- L'eficiència cau a causa de la virtualització, part dels recursos s'han de dedicar a la seva gestió.

- Cost de les llicències de virtualització.

Partició estàtica

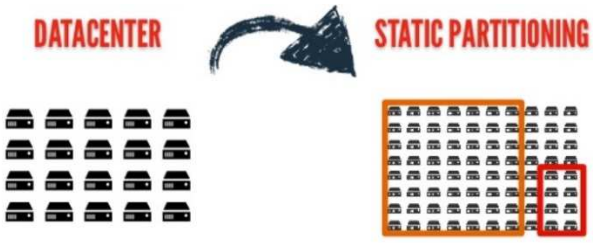


Figura 16. Configuració de centre de dades sobre partició estàtica.

- Comparteix els motius explicats per a la virtualització, i a més la complexitat de recuperació d'errors.

Així doncs algunes solucions per compartir el clúster estan basades en partició estàtica, de forma que només s'executa un entorn d'aplicació per partició, o bé s'instal·len una sèrie de màquines virtuals per cada entorn d'aplicació. El punt feble d'aquestes dos configuracions es troba en que no permeten arribar al màxim d'utilització, o compartir dades entre aplicacions de forma eficient. La dificultat sol provenir dels propis entorns d'aplicació, que tot i ser eficients en la seva execució local, solen ser independents i no permeten el traspàs de recursos entre ells. Per exemple si una de les particions està en espera, el seu entorn d'aplicació no pot traspassar els recursos que no fa servir a l'altre partició, per a què finalitzi les tasques de forma més ràpida o doni més servei. El model de MESOS és el següent:

MESOS

Figura 17. Configuració de centre de dades amb MESOS.

- El clúster es converteix en un repositori de recursos (*pool*), una abstracció que permet oblidar al programador les dificultats d'administració de recursos i llicències, de forma que només observa una capa homogènia que li facilita la tasca de programar. El centre de dades es veu com a una única màquina, enlloc de les milers que el poden formar.

Eficiència

Com es pot veure al gràfic d'ocupació de CPU d'un clúster, figura 18, la partició estàtica no pot recollir el dinamisme de les aplicacions (RAILS, MEMCACHED i HADOOP), i per tant el repartiment dinàmic de recursos entre les aplicacions de forma transparent a elles no és possible. Conseqüentment el rendiment del clúster no s'aprofita al llarg de tot el temps, en hores baixa demanda molts recursos estarien inactius.

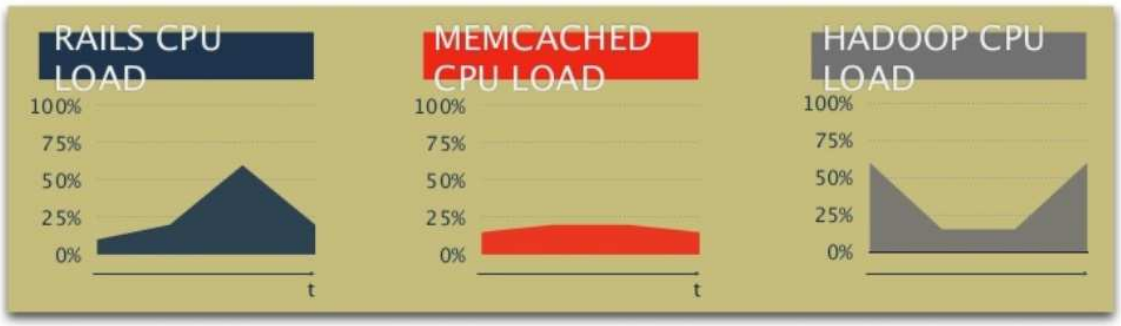


Figura 18. Possible ocupació de recursos de clúster en el temps.

L'avantatge de MESOS és que està construït com una interfície per als entorns d'aplicació, en realitat és un *kernel*, de forma que permet la compartició de gra fi de recursos entre els entorns d'aplicació, de forma que l'ocupació de CPU augmenta i per tant l'eficiència de consum de recursos. Així la part fonamental de MESOS és que es constitueix en una capa que comunica els diferents entorns entre si i d'aquesta manera la compartició de recursos és possible, els recursos tot i ser distribuïts són tractats com un sol recurs i es poden oferir per executar múltiples aplicacions de forma paral·lela com es pot veure a la figura 19.

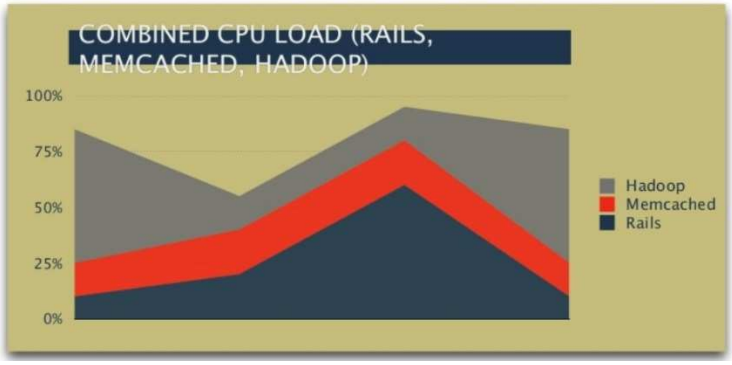


Figura 19. Ocupació de recursos de clúster en el temps amb MESOS.

L'entorn d'aplicació es converteix en un traductor entre aplicació i gestor del clúster MESOS, de forma que comunica les necessitats de recursos de l'aplicació al màster que gestiona el clúster, i aquest pot fer la distribució de recursos lliures, si els té, que satisfacin la demanda rebuda. El punt principal del disseny és per tant com trobar la solució de repartiment de recursos amb les aplicacions en execució, en aquest sentit es troben els següents punts d'interès:

- La solució necessita satisfer una amplia llista d'aplicacions, actuals i també futures.

- La solució ha de ser escalable, ja que els clústers actuals contenen desenes de milers de nodes, centenars de treballs i milions de tasques en execució.

- La planificació ha de ser resistent a errors i d'alta disponibilitat, parlem tant de la caiguda del node màster com dels nodes esclaus.

La solució que ha desenvolupat amb MESOS està basada en delegar el control de planificació sobre els propis entorns d'aplicació, amb l'abstracció que ja hem anomenat abans i que s'indica com a oferidor de recursos. Aquesta abstracció està dissenyada per encapsular els recursos d'un node per executar tasques i que són distribuïts per l'entorn d'aplicació, com es pot veure a la figura 20.

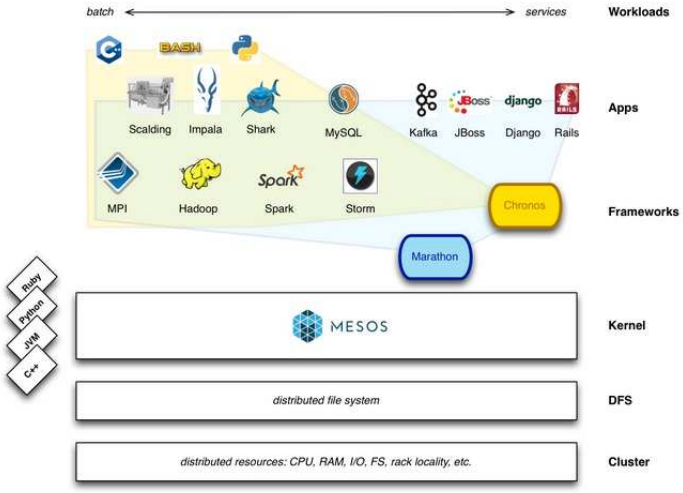


Figura 20. Capa MESOS situada entre les aplicacions i els recursos.

En tot cas MESOS decideix quants recursos ofereix als entorns d'aplicació, basat en polítiques d'organització (per exemple *fair sharing*[m]). Els entorns d'aplicació en canvi decideixen quins recursos accepten i quines tasques s'executen sobre ells. Aquesta facilitat de posar a disposició els recursos del clúster fa que sigui fàcil d'implementar i eficient, de forma que MESOS permet un clúster de fàcil escalat i robustesa als errors. Altres avantatges:

[m] Els recursos són compartits de forma igualitària entre els demandants.

- Es poden executar múltiples instàncies de l'entorn d'aplicació, o diferents versions d'aquest, en el mateix clúster.

- Permet la compartició de recursos de forma transversal entre tots els entorns d'aplicació del clúster.

- Es pot desenvolupar un entorn d'aplicació per a cada aplicació a resoldre, fins al nivell de detall que es decideixi, i aquests poden conviure al clúster.

3.1.2 Arquitectura

Com s'ha comentat MESOS subministra un nucli resilient i escalable, que permet l'execució concurrent de varis entorns d'aplicació, per compartir de forma eficient els recursos del clúster. Els propis desenvolupadors de la solució han tingut en compte en el seu disseny, la diversitat d'entorns d'aplicació i la seva ràpida evolució, i han volgut utilitzar una filosofia de disseny que impliqui una interfície mínima que permeti la compartició de recursos transversal de la forma més eficient possible, a la vegada que s'aplica un control en l'execució de tasques i entorns d'aplicació. El control s'ha derivat als entorns d'aplicació per dos raons principals:

- Permet als entorns d'aplicació la implementació diverses aproximacions a problemes en el clúster, i que s'evolucionin solucions de forma independent.

- Manté MESOS simple i minimitza els canvis al clúster, a la vegada que el manté robust i escalable.

La fora de fer-ho també passa per què MESOS resolgui els detalls tècnics de baix nivell de la interfície, i s'espera que llibreries d'alt nivell resolguin la resta de problemes, com es pot apreciar a la figura 21. D'aquesta forma es permet una evolució més ràpida del sistema i independent de la part de baix nivell, que serà la més estable.

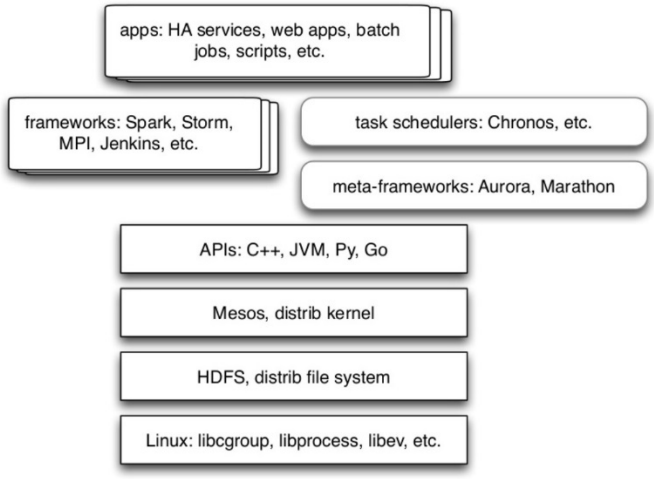


Figura 21. Arquitectura MESOS.

Els principals components de MESOS, mostrats a la figura 22, són el procés màster que administra dimonis esclau, que s'executen en cada node del clúster. Els entorns d'aplicació executen tasques sobre els nodes esclaus.

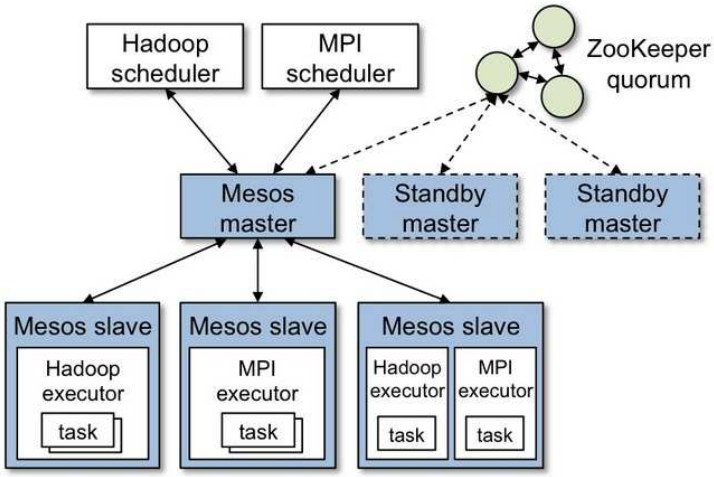


Figura 22. Components de clúster MESOS.

Funcionament del clúster:

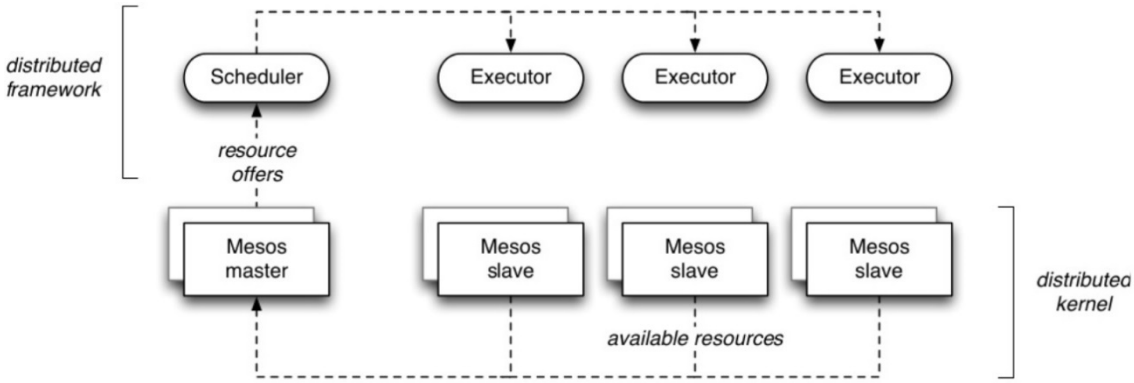


Figura 23. Funcionament de tasques i recursos al clúster MESOS.

- El node màster implementa la compartició de recursos en mode gra fi, oferint recursos als entorns d'aplicació. Cada oferiment consta d'una llista de recursos lliures, disponibles en múltiples esclaus. El node màster decideix quants recursos ofereix, d'acord a una política d'organització com *fair sharing* o *strict priority*[n].

- L'entorn d'aplicació consisteix en dos components:

    - Un planificador (*scheduler*). Es comunica amb el màster per a la gestió de recursos.

    - Un executor. Que llença les tasques de l'entorn d'aplicació als nodes esclau.

  La forma de treballar entre el màster i l'entorn d'aplicació és la següent:

    1. El màster determina el total de recursos que pot oferir a cada entorn d'aplicació.

    2. Cada entorn d'aplicació mitjançant el seu planificador, seleccioni exactament quins recursos dels oferts utilitzarà.

    3. Quan un entorn d'aplicació accepta els recursos, li passa la llista a MESOS de les tasques a executar.

[n] Existeix un ordre de prioritat, que s'executa de forma seqüencial: l'entorn d'aplicació amb més prioritat seria el triat per oferir-li els recursos fins que finalitzi les tasques, llavors es continuaria amb el següent.

4. Finalment MESOS activa les tasques que s'han demanat als nodes esclaus que formen part dels recursos oferts i seleccionats.

- Els nodes esclaus executen les tasques dels entorns d'aplicació. També informen al node màster dels recursos disponibles, i a l'executor de l'entorn de planificació dels recursos assignats per a l'execució de les tasques que li són assignades.

- Per mantenir la capacitat del sistema davant d'errors s'utilitza *Zookeper*[12]. Els entorns d'aplicació depenen que el màster estigui en funcionament, per tant s'ha de crear un mecanisme que permeti al sistema operar davant d'una caiguda del màster. MESOS utilitza dos tècniques:

  - L'estat del màster és *soft state*, de forma que es pot reconstruir completament el seu estat intern gràcies als missatges periòdics que generen els nodes esclau.

  - S'utilitza un estat d'espera està basat en tenir diverses còpies de salvaguarda del màster, de forma que quan aquest cau Zookeeper selecciona un nou node màster dels que té de salvaguarda, li traspassa les dades i redirecciona tots els nodes esclau i els planificadors dels entorns d'aplicació cap al nou màster. D'aquesta forma el nou màster pot reconstruir el seu estat intern gràcies als missatges que rep dels nodes esclaus i els planificadors dels entorns d'aplicacions.

  Apart de les dos tècniques explicades per recuperar el node màster, MESOS informa als entorns d'aplicació dels errors de tasca, esclau i execució, per a què puguin aplicar la solució que es requereixi. També es possible que un entorn d'aplicació tingui més d'un planificador d'execució, que en cas de caiguda sigui substituït per un altre i només cal notificar el canvi a MESOS.

La figura 24 mostra el funcionament d'execució de tasques en un entorn d'aplicació:

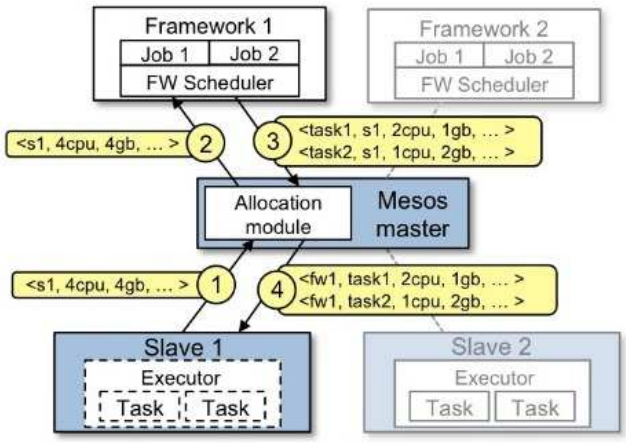


Figura 24. Execució de tasques a clúster MESOS

1. El node esclau 1 indica al node màster que té 4 CPU i 4 GB de ram disponibles. El node màster invoca al mòdul de polítiques de localització de recursos, que indica a l'entorn d'aplicació 1 la disposició d'aquests recursos.

2. El node màster envia una proposta de recursos disponibles al node esclau 1, a l'entorn d'aplicació 1.

3. El planificador de l'entorn d'aplicació 1 contesta al node màster amb informació sobre dos tasques que s'han d'executar en node esclau 1:
   a. Necessita 2 CPU i 1 GB de RAM per a la primera tasca

b. Necessita 1 CPU i 2 GB de RAM per a la segona tasca

4. El node màster envia les tasques al node esclau 1, que disposa dels recursos necessaris informant a l'executor del entorn d'aplicació 1, que en aquest cas executa les dos tasques.

En aquest moment s'estan executant les dos tasques, i a més queden disponibles 1 CPU i 1 GB de RAM, que seran oferts a l'entorn d'aplicació 2. El cicle d'oferir recursos als entorns d'aplicació continuarà segons quedin alliberats i fins que s'hagin executat totes les tasques.

Millores d'eficiència que aporta el gestor:

Aprofitar la localitat espacial de dades per millorar l'eficiència d'execució dins d'un clúster amb dades distribuïdes, té solució en un clúster MESOS. La solució passa per què l'entorn d'aplicació ha de rebutjar tots els oferiments que no inclouen els nodes esclau on la localitat espacial de les dades estigui present. Al final s'aconseguiran els nodes esclaus d'interès per a l'execució més eficient.

Com s'ha descrit MESOS aprofita recursos alliberats per ser cedits de nou, i així aprofitar millor tots els recursos del clúster i augmentar l'eficiència. Un problema pot existir quan les tasques són de llarga durada o bé els recursos no s'alliberen a la velocitat adequada, llavors el clúster pot quedar-se sense oferir recursos durant llarg temps. En aquest cas MESOS també pot eliminar una tasca, tot i que abans de fer-ho dóna un període de gràcia a l'entorn d'aplicació que ha rebut l'avís, si l'executor no respon a la crida immediatament l'elimina i amb ell a totes les tasques que hi depenen. El mecanisme de revocació de tasques consta de dos mecanismes:

- Hi han tasques que poden ser eliminades sense més problemes, però d'altres no seria possible si existeixen interdependències amb d'altres tasques. En aquest cas s'ha dissenyat un espai de garantia de no eliminació per tasques que sí tenen interdependències.

- Per decidir quan s'ha d'activar la revocació de recursos, s'utilitza la previsió de consum de recursos dels entorns d'aplicació. Es possible veure les necessitats de recursos, i si aquests poden superar el total ofert llavors és quan s'executarà el procés.

Com s'ha vist la planificació de tasques es realitza de forma distribuïda i s'inclou un procés constant de comunicació entre el màster i l'entorn d'aplicació. Per augmentar la seva eficiència s'inclouen les següents estratègies:

- Detectar els entorns d'aplicació que sempre rebutgen alguns recursos. En aquest cas formaran part d'un filtre per evitar oferir-los recursos que sempre rebutgen.

- Els entorns d'aplicació sempre ha de respondre a una oferta de recursos, aquesta resposta pot necessitar un cert temps, que MESOS compta a efectes d'eficiència. De forma que quan abans un entorn d'aplicació rebutja uns recursos, més ràpidament pot accedir als que realment li interessen.

- Si un entorn d'aplicació no dóna resposta a l'oferiment de recursos, finalment MESOS elimina la oferta i la dirigeix a d'altres entorns d'aplicació.

MESOS permet l'execució de diferents entorns d'aplicació en el mateix node esclau, i aplica l'aïllament entre els diferents entorns, com es mostra a la figura 25. Està recolzat en els mecanismes del propi sistema operatiu, coneguts con containers (LXC)[13], que són tecnologies de virtualització lleugeres més eficaces que les basades en màquines virtuals, per què no hi ha emulació de software i s'utilitza el mateix sistema operatiu que el node. Per defecte LXC crea un nom de xarxa privat per cada container, que tenen sortida externa mitjançant la targeta del node, tot i que també poden funcionar sense aquesta definició amb els mateixos drets que una altra aplicació que s'executi al node.

Les tecnologies d'aïllament permeten limitar CPU, memòria, ample de banda de xarxa i utilització d'I/O, A més se suporta la reconfiguració dinàmica dels recursos, que és part de la funcionalitat de MESOS per eliminar recursos dels executors dels entorns d'aplicació.

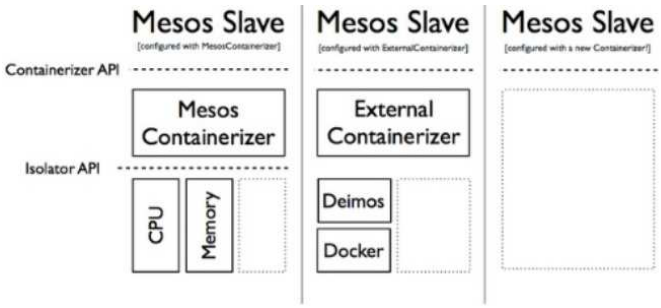


Figura 25. Aïllament entre entorns i containers MESOS

3.2 Entorn d'aplicació MESOS MPI

Com a part demostrativa d'aquest projecte s'ha presentat la possibilitat d'activar un clúster real, petit en recursos al ser aquests de baix cost, però que permeti demostrar les capacitats de gestió de MESOS. L'administració del clúster, limitada a la instal·lació de la plataforma de gestió, queda una mica limitat per conèixer les seves capacitats, ja que per si mateixa no permet cap execució real d'aplicacions fora de les que permeten fer una verificació del seu funcionament (uns petits test escrits en C, Java i Python). Per demostrar el funcionament real del clúster s'ha de desplegar una aplicació distribuïda real, i és en aquest punt on s'ha de prendre una decisió important, que és quina aplicació és vol muntar a sobre de la capa de gestió que proporciona MESOS. Aquesta decisió queda limitada pels coneixements previs i els entorns d'aplicació disponibles, que ja es van presentar a la secció 3.1, i per les limitacions tècniques de procés i emmagatzematge del clúster físic que s'ha implementat.

La solució ha passat per triar una aplicació basada en MPI (*Message-Passing Interface*), que és com el seu nom indica de forma precisa, un model de programació, o interfície, que permet l'execució de programes de forma distribuïda per pas de missatges entre les tasques en execució, tant en memòria com en múltiples processadors, i per tant ideal per provar un entorn format per computadors heterogenis i distribuïdes en xarxa MIMD de memòria distribuïda[o]. La tria compleix amb els coneixements previs necessaris per entendre el seu funcionament bàsic, i que es van adquirir en assignatures dels estudis. Les característiques més importants d'MPI són les següents:

- MPI és una interfície de biblioteca de pas de missatges, és una especificació no una aplicació ni un llenguatge, i totes les operacions estan definides com funcions, subrutines o mètodes. Es millora així la portabilitat d'aplicacions i la seva programació i ús.

- Implementa mecanismes per a comunicar-se entre processos punt a punt i també de forma col·lectiva, incloent mecanismes de sincronització. Produeix una millora en el procés de comunicació, que queda definit a baix nivell.

[o] MIMD amb memòria distribuïda. El model MIMD (Multiple Input Multiple Data) definit per Michael J. Flynn en la classificació de les màquines, segons el seu paral·lelisme, està composats per múltiples computadors independents que operen sobre diferents seqüències de dades. El model MIMD amb memòria distribuïda és una ampliació del model proposada, per Andrew S. Tanembaum, on cada processador utilitza un espai de memòria propi i independent dels altres processadors. En aquest model la comunicació dels processos es realitza per pas de missatges entre ells.
Per a referència completa: Mòdul 1, apartat 2, *Taxonomia de Flynn i altres*. Inclòs als materials de l'assignatura de l'UOC: Arquitectures de Computadors Avançades.

- De forma típica existeix un procés que s'encarrega de distribuir les dades als restants processos remots (N-1), quals els processos remots finalitzen el càlcul demanat retornen el resultat al procés iniciador. El flux d'informació entre el procés iniciador i els processos remots es realitza amb la tècnica de pas de missatges.

- Existeixen múltiples implementacions de MPI: Open MPI, MVAPICH, MPICH2, ...

- Es utilitzat en diferents llenguatges com Fortran, C, Java i Python.

Aquest model de programació va sorgir per un esforç dels diferents fabricants a partir de l'any 1992, per unificar els seus models, ja que fins aquell moment cada fabricant disposava d'un desenvolupament propi i era incompatible amb d'altres fabricants. El grup creat per al procés finalment va iniciar una versió operativa a l'any 1994 amb la versió 1.0 d'MPI[14]. Els detalls del grup de treball han continuat i en aquest moment està disponible una versió MPI-3.0 aprovada al setembre de 2012[15].

En aquest sentit MESOS disposa de dos possibilitats d'executar computació paral·lela utilitzant MPI:

- Mitjançant el seu entorn d'aplicació dissenyat específicament per a executar MPI. Es tracta d'un captura (*wrap*) de la crida original d'execució *mpiexec* de la llibreria MPI, abans de la seva execució. D'aquesta manera el gestor podrà afegir els recursos disponibles al clúster en la crida (màxim de nodes, memòria i CPU disponibles en cas necessari). Es pot estudiar el codi d'aquesta crida MPI a MESOS, anomenada *mpiexec-mesos*, a l'annex d'aquest projecte.

- Mitjançant l'entorn d'aplicació Hydra[p] a MESOSPHERE.

En aquest projecte, a l'utilitzar un clúster privat, s'optarà només per la primera opció.

[p] https://github.com/mesosphere/mesos-hydra

4. Implementació real d'un clúster administrat per MESOS

Aquest apartat té una gran vessant pràctica, ja que es tractarà tota la implementació del clúster projectat. S'iniciarà amb la configuració de xarxa i maquinari, es continuarà instal·lant el gestor MESOS i es conclourà amb les instal·lacions necessàries per a executar aplicacions MPI. El resultat final serà un clúster de computació paral·lela MPI totalment funcional.

4.1 Configuració del maquinari

Les següents seccions tractaran sobre els treballs necessaris per obtenir un clúster operatiu de computació gestionat per MESOS. Els treballs implicats comencen des de zero, amb tres màquines físiques, a les que es van configurant els elements lògics necessaris.

4.1.1 Maquinari físic utilitzat

El gestor MESOS s'utilitzarà per administrar els recursos d'un clúster privat, realitzat amb maquinari de baix cost. El maquinari i programari sobre el que funcionarà el clúster serà el següent:

- Un PC Intel (R) core (TM)2 Quad CPU Q6600@2.4 GHz, 3.25 GB DRAM.
  Programari instal·lat:
  - Sistema operatiu Microsoft Windows 7, service pack 1, amb les següents aplicacions instal·lades al principi del projecte amb relació directa:
    - Paquet ofimàtic Microsoft Office 2013
    - Microsoft Project 2013
    - Microsoft Visio 2013
    - Màquina virtual VMWare© Player, versión 6.0.3
      Programari instal·lat:
      - Sistema operatiu Ubuntu server versió 14.04
        Programari instal·lat:
        - JAVA versió 1.6
        - MESOS 0.20.0
        - MPICH2
        - NAS NPB
        - PERFTEST-1.5
      - Sistema operatiu Ubuntu Desktop versió 14.04
        Programari instal·lat:
        - JAVA versió 1.6
        - MPICH2
        - Gnuplot

- Dos PC AMD 64 3200+ 2 GHz, 2 GB DRAM.
  Programari instal·lat:
  - Sistema operatiu Ubuntu server versió 14.04
    Programari instal·lat:
    - JAVA versió 1.7
    - MESOS 0.20.0
    - MPICH2

- Un commutador (*switch*) de 8 ports a 10/100 Mbps marca SMC, model SMCFS8, sobre el que s'instal·larà la xarxa local, a 100 Mbps. Un total de quatre cables de xarxa de tipus 5 permeten connectar PC, commutador i enrutador (*router*).

- Un enrutador (*router*) ADSL2+ amb xarxa DHCP 10/100 Mbps

4.1.2 Configuració de la xarxa de comunicacions

L'esquema de xarxa és el següent:

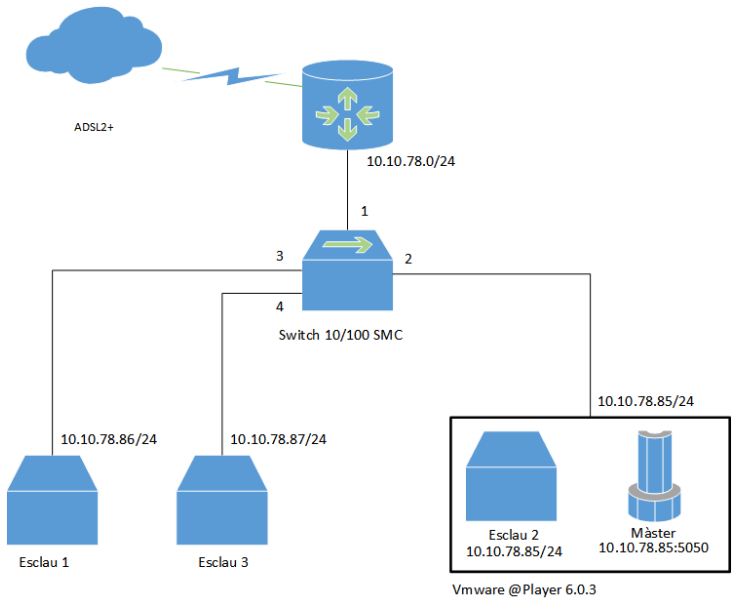


Figura 26. Esquema de xarxa del clúster sobre el que s'instal·larà MESOS.

Es pot observar la següent distribució del clúster:

- Dos màquines físiques, que conformen els nodes esclaus 1 i 3.

- Una màquina virtualitzada, que conforma el node màster del sistema, i que també permet activar un node esclau en local.

El commutador SMC crea una xarxa local de 100 Mbps entre els diferents nodes, totes les targetes de xarxa son de 100 Mbps. El commutador per si sol pot crear un segment de xarxa dedicat al clúster, i no es crea cap VLAN, però per necessitat de descàrrega de *packages*, i altres eines per a la instal·lació del clúster, és necessari disposar d'una connexió a Internet. Aquesta s'aconsegueix connectant el port 1 del commutador amb la connexió de xarxa de l'enrutador ADSL2+. Els valors de xarxa de cada element són els següents:

- Xarxa, adreça 10.10.78.0
- Gateway, adreça 10.10.78.1
- Broadcast, adreça 10.10.78.255
- Màscara de xarxa: 255.255.255.0
- Nodes:
    - Màster adreça 10.10.78.85:5050
    - Esclau 1 adreça 10.10.78.86
    - Esclau 2 adreça 10.10.78.85 (local al màster)
    - Esclau 3 adreça 10.10.78.87

La situació del node màster i esclau 3 són una mica especials, ja que es troben virtualitzats a dins de l'aplicació VMware. Aquests nodes necessiten sortir de la pròpia aplicació i és necessari fer uns canvis en la configuració de la targeta de xarxa virtual, de forma que dupliqui la interfície física real, com es mostra a la figura 27:

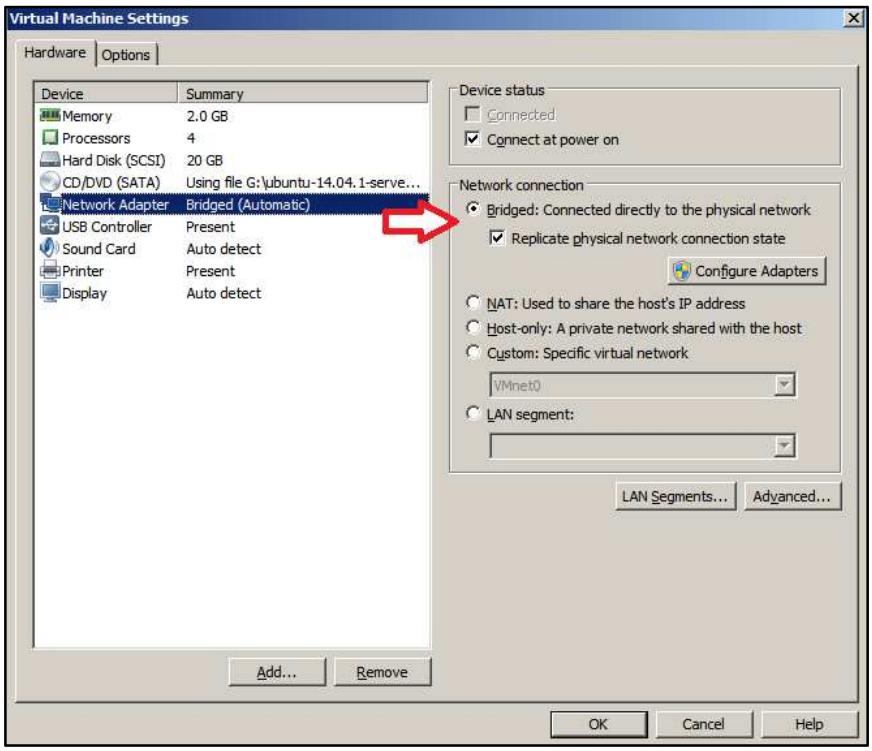


Figura 27. Configurar connexió de xarxa a VMware. Nodes virtualitzats: màster i esclau 2.

El canvi passa per replicar la interfície física de la màquina física a la virtual, que està compartida pel node màster i l'esclau 3 a l'adreça IP 10.10.78.85.

4.2 Configuració dels nodes del clúster MESOS

L'administració de clúster MESOS es pot instal·lar de forma directa sobre una màquina dedicada, de forma que pot funcionar el màster i l'esclau de forma concurrent. En aquesta configuració mínima es pot executar l'entorn d'aplicació (*framework*) mínim en C++, Java o Python per provar el seu funcionament. La instal·lació és senzilla i està perfectament indicada a la pàgina web del projecte[q]. Els passos són els següents:

1. El node, sigui màster o esclau, ha de tenir instal·lada la versió UBUNTU 12.04 almenys. En el clúster del projecte es va triar la versió UBUNTU 14.04 per problemes amb la versió 12.02 (impossible instal·lar els paquets obligatoris per enllaços perduts). Així a tots els nodes del clúster s'ha instal·lat una versió UBUNTU 14.04.1 server AMD 64[r]. La instal·lació s'ha fet des de zero, eliminant qualsevol dada anterior als discs durs de les màquines i ocupant el seu màxim espai. Com a aplicació addicional a les màquines esclau s'ha instal·lat OPENSSH per poder accedir de forma remota, al màster no ha estat necessari ja que accedirem mitjançant VMware.

2. La instal·lació de MESOS a cada node obliga a descarregar i instal·lar, de forma prèvia, els següents paquets des de la consola del sistema:

```
# Actualitzar el sistema
$ sudo apt-get update

# Instal·lar eines de construcció
$ sudo apt-get install build-essential
```

[q] http://mesos.apache.org/gettingstarted/
[r] http://releases.ubuntu.com/trusty/

```
# Instal·lar OpenJDK java.
$ sudo apt-get install openjdk-6-jdk

# Instal·lar Python.
$ sudo apt-get install python-dev python-boto

# Instal·lar libcurl
$ sudo apt-get install libcurl4-nss-dev

# Instal·lar libsasl.
$ sudo apt-get install libsasl2-dev

# Instal·lar Maven.
$ sudo apt-get install maven
```

Els dos últims paquets s'instal·len per què es farà servir la versió 0.20.0 de MESOS.

3. Una vegada s'ha completat la instal·lació es procedeix a descarregar la versió MESOS, en el cas del projecte és la 0.20.0, seguint les instruccions de la pròpia web:

```
$ wget http://www.apache.org/dist/mesos/0.20.0/mesos-0.20.0.tar.gz
$ tar -zxf mesos-0.20.0.tar.gz
```

De forma que es crea un directori mesos-0.20.0 on es descomprimeix l'arxiu descarregat.

4. En aquest moment ja tenim instal·lat sistema operatiu, paquets de suport i administrador MESOS, el pas següent és construir el nucli MESOS i de nou la pàgina web indica els passos necessaris:

```
$ cd mesos-0.20.0

# Configurar i construir sobre el directori build
$ mkdir build
$ cd build
$ ../configure
$ make

# Passar test al clúster (necessari per activar els framework de test)
$ make check
$ make install
```

El procés partint de zero té una duració temporal força elevada, tot i que amb la configuració de màquina adequada es resol perfectament seguint la guia anterior. A mode d'exemple es mostra l'abast temporal d'instal·lació en un dels nodes de tot el procés anterior:

El procés d'instal·lació comença a les 11:25h.

Hora inici	Hora fi	Tasca	Duració
11:25	11:55	Instal·lar *UBUNTU 14.04.1*	30 minuts
11:55	12:20	Instal·lar els paquets	25 minuts
12:20	12:21	Executar *./configure*	1 minut
12:22	14:34	Executar *make*	2 hores i 12 minuts
14:34	16:25	Executar *make check*	1 hora i 51 minuts
16:25	16:26	Executar *make install*	1 minut

Taula 2. Temporització de la instal·lació d'un node MESOS.

En total són necessàries **5 hores** per instal·lar el programari bàsic d'un node del clúster.

Una vegada s'han instal·lat els serveis bàsics a cada node és possible iniciar el clúster MESOS, però primer s'ha de configurar els valors de xarxa a cada element per poder permetre la comunicació entre els nodes. Inicialment la instal·lació s'ha afavorit d'una configuració de targetes amb IP dinàmica, servida pel servidor DHCP de l'enrutador, ara s'ha de canviar la configuració DHCP a fixa per afavorir les rutes d'intercanvi entre màster i esclaus. Per fer-ho a cada node s'ha d'editar el fitxer */etc/network/interfaces*. En el nostre cas s'ha utilitzat l'editor *vi* per canviar els valors i deixar els següents per node (totes les targetes de xarxa són *eth0*):

Node màster/esclau 2

iface lo inet loopback
iface eth0 inet static
address **10.10.78.85**
netmask 255.255.255.0
network 10.10.78.0
broadcast 10.10.78.255
gateway 10.10.78.1
dns-nameservers 62.36.225.150 62.37.228.20

Node esclau 1

iface lo inet loopback
iface eth0 inet static
address **10.10.78.86**
netmask 255.255.255.0
network 10.10.78.0
broadcast 10.10.78.255
gateway 10.10.78.1
dns-nameservers 62.36.225.150 62.37.228.20

Node esclau 3

iface lo inet loopback
iface eth0 inet static
address **10.10.78.87**
netmask 255.255.255.0
network 10.10.78.0
broadcast 10.10.78.255
gateway 10.10.78.1
dns-nameservers 62.36.225.150 62.37.228.20

Notes:

1. La inclusió de les adreces dels servidors *DNS* s'ha produït després que en diverses temptatives d'engegar el servei del màster, aquest no ho fes. Una vegada analitzat el problema s'ha decidit la seva inclusió, funcionant el servei sense cap problema addicional. Els valors indicats són els obtinguts de la configuració de l'enrutador ADSL2+.

2. La documentació de MESOS utilitza un exemple on només existeix una màquina, on s'instal·la tot el sistema amb un màster i un esclau. Així les adreces IP que utilitza son les de *loopback* 127.0.0.1 per comunicar els dos serveis. Aquesta configuració no serveix en el cas de muntar el clúster privat per què és necessari comunicar als esclaus 1 i 2 a quina IP s'han d'adreçar, i lògicament aquesta no pot ser de la *loopback*.

3. L'enrutador té configurat un servidor *DHCP*, però el clúster privat és més fàcil muntar-lo amb IP fixa, almenys el màster per a què sigui trobat pels esclaus. Per solucionar el problema el rang IP definit al servidor *DHCP* és 10.10.78.0/24 de forma que podem tenir tres adreces IP (10.10.78.85, 10.10.78.86 i 10.10.78.87) fixes sense provocar cap conflicte (la resta de màquines que han d'accedir a una IP del servidor *DHCP* són menys de 10).

Activació del clúster

La guia MESOS indica els següents passos per iniciar el node màster i els nodes esclaus i per tant activar el clúster de forma efectiva:

Activació del node màster 10.10.78.85

```
# S'ha d'entrar al directori build suposant que estem a /mesos.0.20.
$ cd build

# Activar el node màster
$ ./bin/mesos-master.sh --ip=10.10.78.85 --work_dir=/var/lib/mesos
```

Nota:

1. El directori /var/lib/mesos s'ha de crear de forma prèvia a aquesta crida i se li han d'assignar els permisos necessaris. En el nostre cas es va fer un *chmod 777 mesos* per donar tots els permisos a tots els usuaris.

Activació del node esclau 10.10.78.85

```
# Activar node esclau.
$ ./bin/mesos-slave.sh --ip=10.10.78.85 --master=10.10.78.85:5050 --resources:
"mem(*):1024;cpus(*):4" --no-switch_user
```

Activació del node esclau 10.10.78.86

```
# Activar node esclau.
$ ./bin/mesos-slave.sh –ip=10.10.78.86 --master=10.10.78.85:5050 --resources:
"mem(*):1024;cpus(*):1" --no-switch_user
```

Activació del node esclau 10.10.78.87

```
# Activar node esclau.
$ ./bin/mesos-slave.sh –ip=10.10.78.87 --master=10.10.78.85:5050 --resources:
"mem(*):1024;cpus(*):1" --no-switch_user
```

L'activació del clúster és força senzilla, només és necessari activar cada rol a la màquina corresponent. A més el clúster té capacitat d'autorecuperació, ja que si s'activa primer un esclau i després el màster la conformació del clúster també succeeix als pocs segons. Una altra característica és que la instal·lació de MESOS permetria activar tres màsters i tres esclaus sense problemes, ja que per defecte cada node té les dos crides, i així fer proves a cada node de forma independent. Per motius de limitació tècnica a causa del maquinari emprat en aquest projecte només es treballarà amb la configuració un màster i tres esclaus, quan fàcilment es podria establir un clúster d'alta disponibilitat amb tres màsters activant el gestor de màsters *Zookeeper* que s'encarregaria que sempre existís un màster gestor del clúster.

De forma general l'activació dels nodes del clúster s'efectua amb la següent crida

```
# Activar el node màster
$ ./bin/mesos-master.sh --ip= <ip node màster> --work_dir=/var/lib/mesos

# Activar node esclau.
$ ./bin/mesos-slave.sh –ip= <ip node esclau> --master=<ip node màster>:5050 --resources: "mem(*):<mida
memòria (MB);cpus(*):<número de CPU>" --no-switch_user
```

Les dos crides anteriors només contemplen les opcions mínimes, una llista extensa de les opcions de configuració, que es poden introduir per línia de comanda, es pot llegir si s'executa cada crida amb l'opció --help:

```
# Informació del node màster
$ ./bin/mesos-master.sh --help

# Informació del node esclau
$ ./bin/mesos-slave.sh --help
```

Algunes d'interessants:

- Canviar el port: *--port*, per defecte 5050= VALOR
- Indicar on gravar els fitxers de *log*, per defecte no els guarda: --log_dir= VALOR
- Eliminar els missatges sobre *stderr*, per defecte el valor és fals: --[no-]quiet

4.3 Administrador web del clúster (WEBUI)

La forma de comprovar que el clúster està completament desplegat és visitant la pàgina web local:

10.10.78.85:5050

On s'accedirà a un servidor web que mostra informació de l'execució del clúster. En el nostre cas el màster està virtualitzat a dins de l'aplicació VMware, de forma que comparteix IP amb la màquina local com ja s'ha indicat. Així amb el navegador de la maquina física on es troba virtualitzat el màster, i fins i tot des de qualsevol navegador en aquell segment de xarxa, accedirem al servidor web que ens mostrarà la següent informació sobre el clúster, com es comprova a les següents figures:

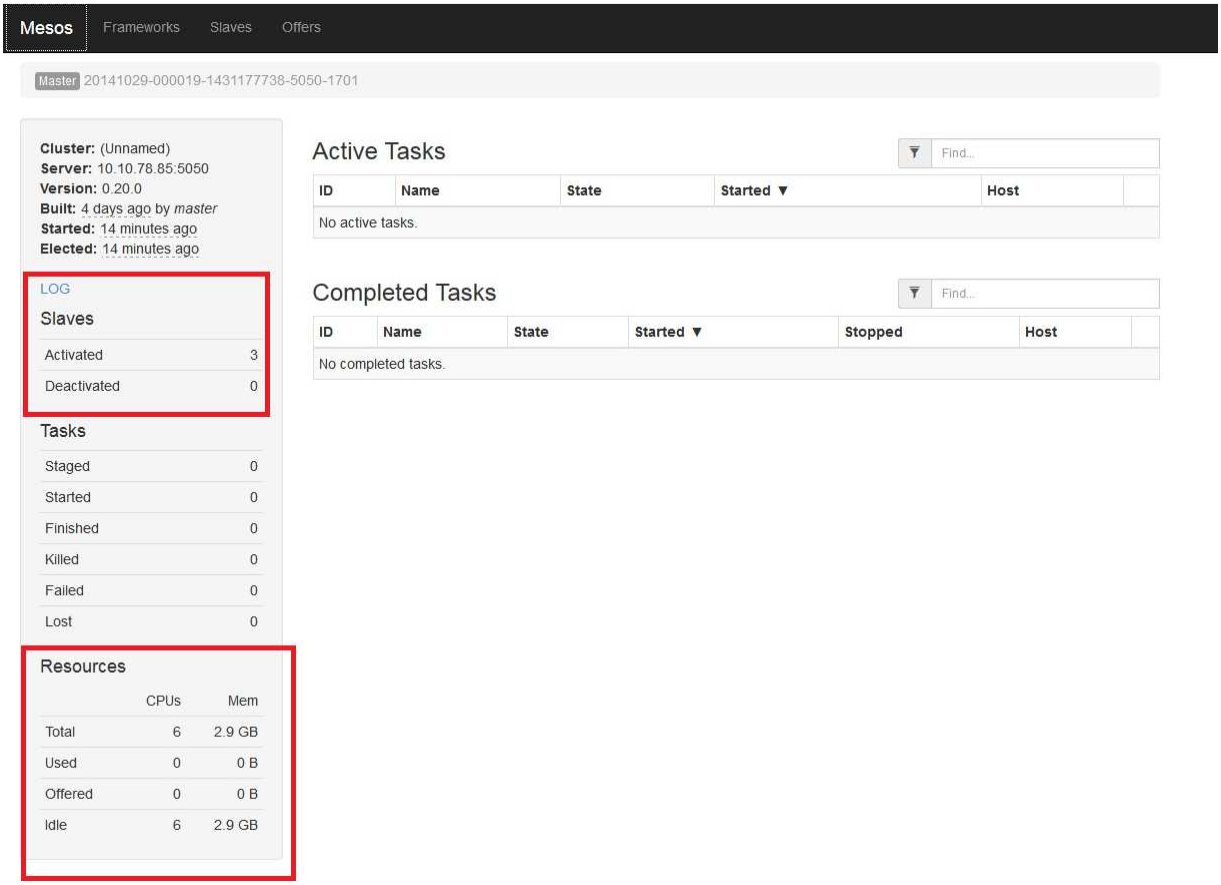


Figura 28. WebUI. Detall d'esclaus activats i recursos disponibles.

A la figura 28 es pot veure l'existència de tres esclaus activats, que posen a disposició del clúster els següents recursos:

- 6 CPU
- 2.9 GB de memòria (configuració dels esclaus amb recursos de memòria per defecte)

Si es visita la pestanya *slaves* es pot veure amb detall els tres esclaus que conformen el clúster:

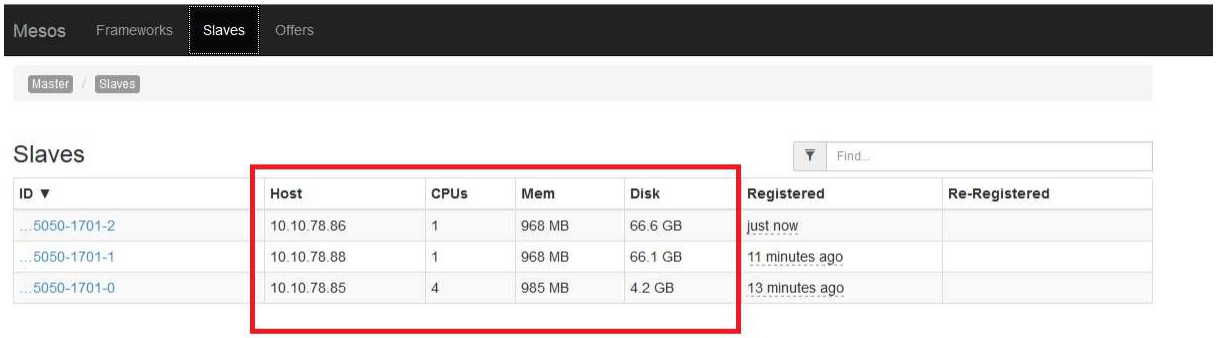


Figura 29. WebUI. Exemple d'esclaus actius i recursos que ofereixen.

La prova final que el clúster és totalment operatiu passa per l'execució d'unes tasques natives d'exemple al node màster. Per executar-les només cal fer la crida corresponent:

```
# Activar proves sobre el framework C++.
$ ./src/test-framework --master=10.10.78.85:5050

# Activar proves sobre el framework Java
$ ./src/examples/java/test-framework 10.10.78.85:5050

# Activar proves sobre el framework Python
$ ./src/examples/python/test-framework 10.10.78.85:5050
```

El màster distribueix les tasques sobre els nodes del clúster i són executades de forma distribuïda. La pàgina web de clúster mostra la informació relacionada que demostra que el clúster està totalment operatiu:

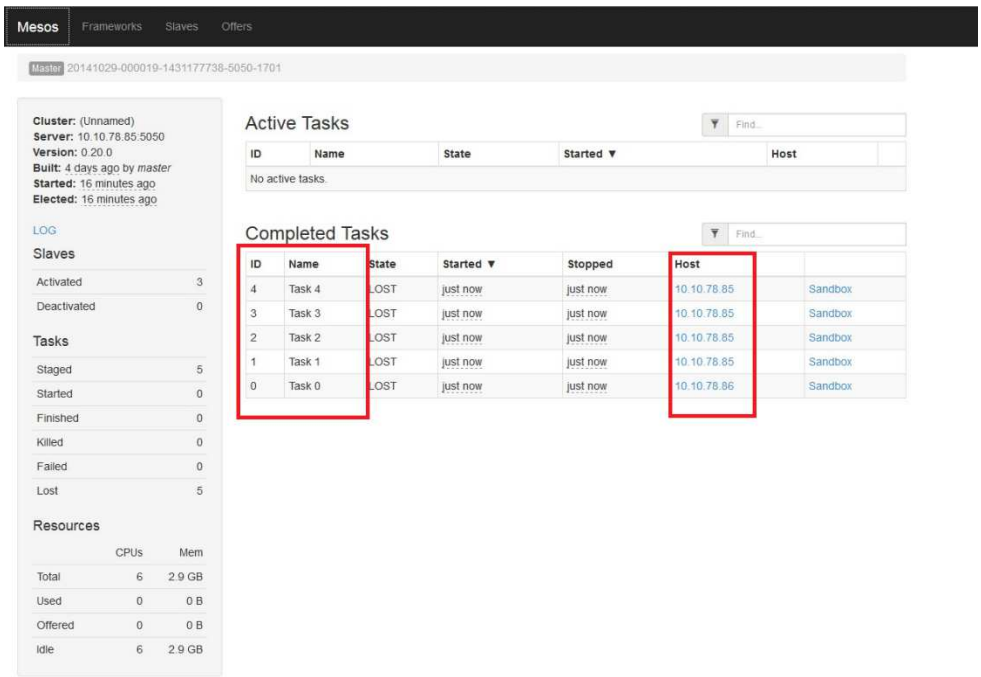


Figura 30. WebUI. Exemple d'execució finalitzada al clúster.

4.4 Instal·lació de l'entorn d'aplicació per a MPI

A la secció 3.2 s'ha indicat que, en aquest projecte, les proves d'execució real sobre el gestor MESOS es realitzaran utilitzant l'entorn d'aplicació MPI. Es a dir que crearem un clúster de computació paral·lela distribuït amb pas de missatges entre processos. L'estat actual del sistema és el d'un sistema UBUNTU 14.04 amb el clúster MESOS totalment funcional, però ara s'han d'afegir els components necessaris per a permetre el funcionament de les llibreries MPI, per així poder executar treballs de computació paral·lela als nodes del clúster. Aquest procés es recull a les seccions següents.

4.4.1 Requisits inicials

En aquest projecte utilitzarem la versió 1.2 (MPICH2), de novembre de 2009, de l'Argonne National Laboratory (ANL)[16] per recomanació del grup de desenvolupadors de MESOS[17] ja que el dimoni principal, *MPD*, no està present a les versions superiors de MPICH. La versió la descarregarem de forma gratuïta des de l'ANL[18] i seguirem les pautes d'instal·lació que indiquen els desenvolupadors de MESOS, que consisteix és instal·lar des de zero el paquet MPICH 1.2 seguint el seu manual d'instal·lació[19], però completant la informació amb d'altres fonts[20] ja que el manual és molt genèric i dóna per resoltes algunes situacions com la de disposar de sessions d'administració de sessió remota SSH[21] sense validació i també la d'un espai de fitxers comú entre tots els computadors del clúster (NFS[22]).

4.4.2 Instal·lació de les llibreries MPICH2

El procés és força ràpid, tot i que s'ha de tenir una certa cura en seguir l'ordre que s'indica, ja que la gestió d'errors del clúster MPICH2, els missatges d'error que genera, no ajuda a la seva resolució al ser poc precisos. Com es veurà la instal·lació consisteix no només en instal·lar la llibreria MPI a cada computadora del clúster, si no que s'instal·la de forma que tindrem un clúster MPI realment operatiu amb els binaris per executar aplicacions, i necessaris a l'apartat 5 de mètriques com s'explicarà.

Procés d'instal·lació:

1. Definir els computadors que formen part del clúster al fitxer /etc/hosts del computador master:

   ```
   $ vi /etc/hosts
   ```

   Llavors editar el contingut en relació al computadors del clúster:

   ```
   127.0.0.1 localhost
   10.10.78.85 master
   10.10.78.86 slave1
   10.10.78.87 slave3
   ...
   ```

2. Instal·lar NFS, per permetre tenir un directori per a fitxers compartits on deixarem els binaris a executar en tots els nodes del clúster.

   Al node màster:

   ```
   $ sudo apt-get install nfs-server
   ```

   A cada node esclau:

   ```
   $ sudo apt-get install nfs-client
   ```

3. S'ha de crear un directori compartit, que anomenarem /mirror, a cada computador del clúster:

   ```
   $ sudo mkdir /mirror
   ```

4. Per poder compartir aquest directori amb tots els nodes s'ha d'editar al node màster el fitxer /etc/exports i reactivar el servei:
   $ echo "/mirror *(rw,sync)" | sudo tee -a /etc/exports
   $ sudo service nfs-kernel-server restart

5. A cada node esclau es modificarà el fitxer /etc/fstab per indicar que es munti, de forma automàtica a cada inici del sistema, el fitxer /mirror local contra el directori /mirror al node màster:

   $ vi /etc/fstab

   I llavors afegir la línia:

   master:/mirror /mirror nfs

6. A cada node esclau es crea un usuari master amb capacitats root, ja que des del node màster serà l'usuari que mitjançant sessió SSH es comunicarà amb els nodes esclaus:

   $ sudo adduser master root

7. S'instal·larà OpenSSH a tots els nodes del clúster per permetre establir sessions segures SSH entre ells:

   $ sudo apt-get install openssh-server

8. S'han de crear les credencials segures que permetran comunicar el màster amb els esclaus, i aquestes s'ha de copiar a cada node esclau. Així la creació de credencials públiques es realitza al node màster, i llavors es copien a cada esclau:

   $ ssh-keygen –b 4096 -t rsa

   Els detalls del fitxer on es guardarà la clau i la frase de pas (passphrase) es deixen per defecte pressionant la seva validació (enter). A continuació són copiades als nodes esclau:

   $ ssh-copy-id master@10.10.78.86

   $ ssh-copy-id master@10.10.78.87

Apart d'aquestes modificacions dels usuaris és necessària la instal·lació de compiladors de C i Python, però aquests requeriments ja han quedar resolts a l'instal·lar la plataforma MESOS.

Arribat a aquest punt tenim la base adequada al clúster per permetre ara la instal·lació de MPICH2 en format clúster, i seguint les pautes del manual d'instal·lació[23], que s'hauran de repetir a cada màquina (es mostra la creació al node màster només).

1. El primer serà obtenir el paquet d'instal·lació des de la web de l'ANL, en aquest cas és el:

   mpich2-1.2.1p1.tar.gz

2. A continuació es descomprimeix:

   $ tar xfz mpich2-1.2.1p1.tar.gz

3. Es crea el directori d'instal·lació:

```
$ mkdir /home/master/mpich2-install
```

4. Es crea un directori de treball per mantenir net el directori on s'ha descomprimit el fitxer:

```
$ mkdir /home/master/mpich2-build
```

5. En aquest moment es poden triar les opcions de configuració, en el nostre cas s'han deixat les de defecte, i s'executa l'script CONFIGURE en el directori font:

```
$ cd /home/master/mpich2-build
$ /home/master/
```

6. Ara es fa un build d'MPICH2 (el fitxer m.txt serveix per revisar possibles errors del procés):

```
$ make 2>&1 | tee m.txt
```

7. S'instal·len les comandes MPICH2 (el fitxer mi.txt serveix per revisar possibles errors del procés):

```
$ make install |& tee mi.txt
```

8. En aquest moment es pot copiar el directori /home/master/mpich2-install al directori compartit per NFS (/mirror), així estarà disponible per a la resta de nodes sense necessitat d'instal·lar-lo un per un:

```
$ cp –R  /home/master/mpich2-install /mirror
```

9. S'afegeix el directori d'instal·lació al PATH del sistema, per què es puguin trobar els binaris d'execució. S'ha de fer al final de cada fitxer .bashrc dels nodes esclaus i als usuaris master creats en ells (master@slave1 i master@slave3):

```
$ vi .bashrc
```

I afegim al final del fitxer:

```
export PATH=/mirror/mpich2-install/bin:$PATH
```

10. Finalment és necessari crear un parell de fitxers, per què funcioni l'anell d'execució creat pel dimoni MPD, en cada node al directori /home. El primer fitxer es diu .mpd.conf i ha de contenir una única línea de text, a més de permisos de lectura/escriptura limitats a l'usuari propietari. Aquest fitxer :

```
$ echo "MPD_SECRETWORD=nil" | tee .mpd.conf
$ touch .mpd.conf
$ chmod 600 .mpd.conf
```

El segon fitxer es diu mpd.hosts i contindrà el nom de cada node de l'anell MPICH2. Aquest fitxer només cal crear-lo al node màster, ja que des d'ell s'iniciarà un procés automàtic, MPDBOOT, que iniciarà cada node de l'anell. Es mostra el procés de creació:

```
$ vi mpd.hosts
```

I afegim:

```
slave1
slave3
```

4.4.3 Verificació de la instal·lació correcta de les llibreries MPICH2

Una vegada finalitzat el procés d'instal·lació és necessari verificar que és possible iniciar una anell de dimonis *mpd* de forma correcta, i així estalviar verificacions addicionals quan es vulgui executar de forma conjunta amb MESOS com a entorn d'aplicació. Fer-ho d'aquesta manera estalviarà processos de depuració d'errors força més complexos. La forma de comprovar en local a cada node el funcionament de MPICH2 és utilitzant les següents eines:

- mpd: inicia un dimoni MPD.

- mpdtrace: mostra tots els dimonis MPD actius a l'anell, amb l'opció adequada dóna informació sobre el port i IP que utilitzen.

- mpdboot: inicia un anell de dimonis MPD especificats en un fitxer (el que s'ha configurat a mpd.hosts). També es pot decidir quants nodes del total es volen activar.

- mpdcheck: permet verificar si es poden activar els node local, o bé els llistats al fitxer *mpd.hosts*. També permet comprovar si la connexió SSH s'activa sense problemes.

- mpdringtest: permet verificar la latència de l'anell de dimonis MPD indicant quantes vegades s'ha de voltar l'anell (per defecte 1).

- mpdallexit: finalitza el dimoni MPD, si es fa al dimoni MPD iniciador finalitzarà tots els dimonis de l'anell.

- mpdlistjob: mostra una llista dels treballs en execució a l'anell MPD. Cada treball es mostra amb un identificador únic.

- mpdkilljob: permet eliminar un treball en execució. S'ha de passar com a argument l'identificador del treball a eliminar. Aquest identificador el torna mpdlistjob.

Procés de comprovació:

1. Es comprova que podem arribar a les comandes d'execució, en cas positiu retornarà el *path* on es troben:

   $ which mpd
   $ which mpicc
   $ which mpiexec
   $ which mpirun

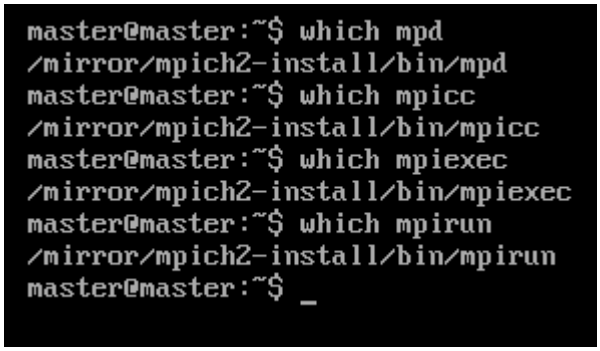


Figura 31. Comprovació dels path als binaris d'execució MPICH2.

2. Comprovar si és possible activar tots els dimonis MPD del fitxer .mpd.hosts, amb comprovació SSH:

   $ mpdcheck –v –f mpd.hosts –ssh

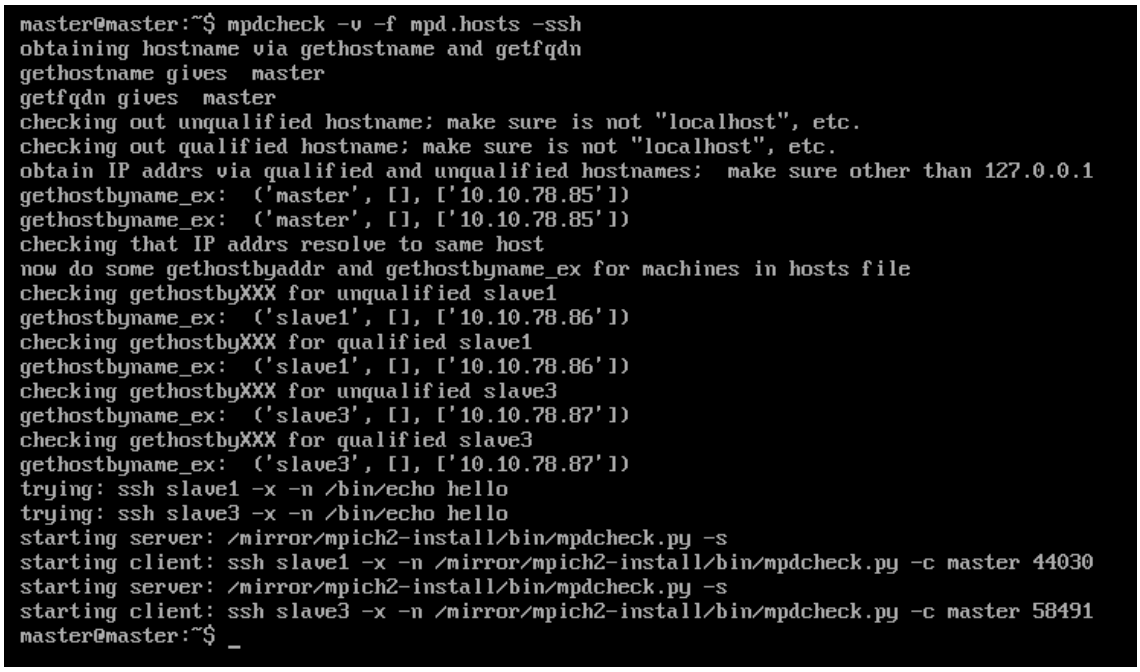


Figura 32. Comprovació de connexions a tots els dimonis MPD. Execució de comanda MPDCHECK.

3. iniciar un dimoni MPD de forma local, i comprovar que realment funciona.

   $ mpd &
   $ mpdtrace -l
   $ mpdallexit

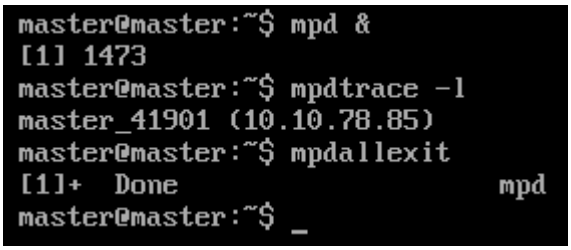


Figura 33. Comprovació d'execució local. Dimoni MPD.

4. Finalment s'activa tot l'anell de dimonis MPD des mpdboot, tres en total, i es fa una prova de verificació de l'anell i llistat de màquines associades:

   $ mpdboot –n 3 –f mpd.hosts -chkup
   $ mpdringtest 1000
   $ mpdtrace –l
   $ mpdallexit

   La figura 34 mostra la sortida del sistema:

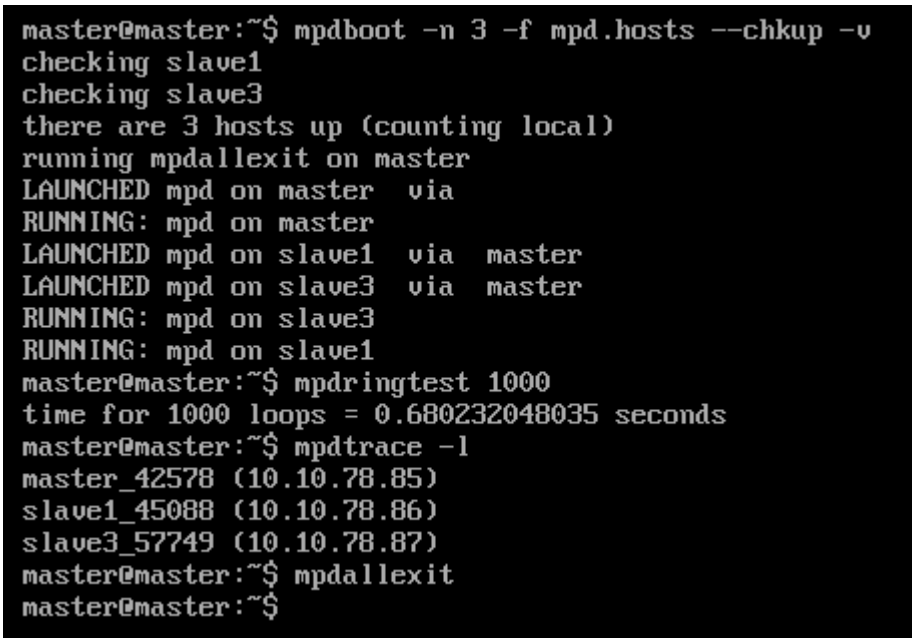


Figura 34. Comprovació d'anell MPD i execució d'un test de 1000 loops de verificació.

Una vegada comprovat que l'anell MPD funciona correctament, es pot finalitzar la verificació de funcionament llançant una aplicació a tots els nodes de l'anell i verificant que funciona correctament. En aquest cas s'ha triat l'aplicació **CPI**, forma part del paquet MPICH2 i es pot trobar al directori font mpich2-1.2.1p/examples. Aquesta aplicació torna el càlcul de π i també permet veure com funciona el pas de missatges des de cada node.

1. El primer és compilar el fitxer font de CPI utilitzant la llibreria MPI per poder executar-lo de forma paral·lela als nodes. S'utilitza el compilador MPICC[24] que també està al paquet d'instal·lació d'MPICH2:

   mpicc <nom fitxer.font> –o <nom.fitxer.sortida>
   $ mpicc –o cpi cpi.c

2. Una vegada compilat el fitxer la seva execució es redueix a executar-lo al node màster, on de forma prèvia s'inicia l'anell de dimonis MPD del clúster. L'execució es realitza amb la comanda MPIEXEC[25] que permet indicar el total de processos que s'executaran, en aquest cas s'indicarà que el valor serà 9 per provar que cada node executa tres cops el càlcul. Els processos s'executen de forma seqüencial sobre els nodes actius (1 (master) ->2 (slave1) -> 3 (slave3) ->1 (master)->2 (slave1) ->...), així tindrem una execució total de 9 processos (tres nodes x tres processos):

   $ mpdboot –n 3 –f mpd.hosts
   $ mpiexec -n 9 /mirror/cpi

En la figura següent es pot comprovar que tenim tres nodes d'execució mpd, el màster també hi participa, i que cadascú d'ells ha fet l'execució tres cops com s'havia definit a les opcions d'execució de MPICC. L'execució té l'ordre master -> slave3 -> slave1 -> master -> ...:

Procés 0 -> master
Procés 1 -> slave3
Procés 2 -> slave1
Procés 3 -> master
Procés 4 -> slave3
Procés 5 -> slave1
Procés 6 -> master
Procés 7 -> slave3

Procés 8 -> slave1

```
master@master:~$ mpdboot -n 3 -f mpd.hosts
master@master:~$ mpiexec -n 9 /mirror/cpi
Process 0 of 9 is on master
Process 6 of 9 is on master
Process 3 of 9 is on master
Process 1 of 9 is on slave3
Process 2 of 9 is on slave1
Process 4 of 9 is on slave3
Process 8 of 9 is on slave1
Process 5 of 9 is on slave1
Process 7 of 9 is on slave3
pi is approximately 3.1415926544231256, Error is 0.0000000008333325
wall clock time = 0.095264
master@master:~$
```

Figura 35. Resultat de l'execució d'aplicació CPI en 9 processos.

El fitxer binari a executar és important que estigui al directori compartit, per què així serà trobat per cada node del clúster en la seva execució.

4.5 Instal·lació MESOSPHERE com a *IaaS*

Aprofitant la tecnologia Apache MESOS diferents companyies permeten la utilització d'infraestuctures clúster com a servei (*IaaS*), en una configuració que rep el nom de Mesosphere[26]. El servei consisteix en l'oferiment d'una configuració de clúster a mida per fer proves de desenvolupament, o executar aplicacions de curt cicle de vida, i pagar pel cost exacte d'utilització. Existeixen diferents proveïdors que han adoptat el gestor de clústers MESOS, sota el nom de Mesosphere:

- Google Cloud Platform[s]
- Amazon Web Services amb el producte Elastic[t] (en fase de llançament)
- Digital Ocean[u]

La idea comuna de tots aquest serveis és el de disposar en pocs minuts d'un clúster totalment operatiu, i escalable fàcilment. Sol començar per la tria de mida del clúster que necessitem, i establir la connexió per acabar de configurar-lo amb els entorns d'aplicació, per finalment executar l'aplicació.

La idea aprofita l'existència d'un projecte que simplifica un *datacenter* en un model equivalent a convertir-lo en una mena d'abstracció d'una computadora simple, controlat per un sistema operatiu. La gestió en un futur proper la realitzarà Mesosphere DCOS, que es presentarà a principis de l'any 2015, i que farà la funció de sistema operatiu del *datacenter*. Aquest sistema operatiu està basat en Apache MESOS. Per ser totalment funcional inclou serveis de nucli com CHRONOS, MARATHON, HDFS i altres. Les característiques principals:

- Kernel distibuït amb grau de seguretat empresarial, basat en Apache MESOS i amb els serveis de nucli ja indicats.

- Executa els treballs en entorns contenidor, administrant aïllament de recursos (memòria, processador i xarxa), i basat en connexions de contenidor Linux o Docker, de forma flexible i tolerant a errors.

[s] https://google.mesosphere.com/
[t] https://elastic.mesosphere.io/
[u] https://digitalocean.mesosphere.com/

L'oportunitat d'oferir la plataforma Mesos com a *IaaS* te com a fi no haver de destinar recursos tècnics de forma permanent a necessitats temporals. Tot el que es pot fer en aquests serveis d'infraestructura també es podria fer en una configuració local amb Apache MESOS, i altres components del sistema, però de vegades pot ser una millor opció contractar el servei. Un exemple d'utilització seria el de de tenir una infraestructura temporal de prova, i si els resultats són òptims llavors sí fer el desplegament amb Apache MESOS i les aplicacions de suport necessàries. Un altre funcionalitat seria la de donar un servei de forma temporal, setmanes o mesos, i una vegada finalitzat desmuntar la infraestructura, i deixar d'assumir uns costos, seria molt ràpid. En tots dos casos una gran avantatge és que si la mida de la infraestructura s'ha d'ampliar, és ràpid fer-ho al disposar de recursos de computació a cadascun dels proveïdors, bàsicament és una qüestió de cost.

Les configuracions de servei clúster tenen els següents costos i configuracions:

Servei	Google Cloud Platform		DigitalOcean	
Configuració	Desenvolupament	Altes Prestacions	Desenvolupament	Altes Prestacions
Instàncies	4	12	4	10
vCPUS	8	24	8	20
Memòria (GB)	30	90	8	20
Màster Mesos	1	3	1	3
Marathon	1	3	1	3
Zookeper	1	3	1	3
Esclaus Mesos	3	9	3	7

Taula 3. Resum de configuracions MESOS IaaS.

5. Obtenció de mètriques sobre el clúster MESOS

Aquest apartat vol estudiar si l'execució de la capa de gestió MESOS implica un cost important del rendiment del clúster. Per fer-ho l'estudi es divideix en dos parts:

- Estudi d'eficiència MPI executant el gestor MESOS al clúster, contra una execució sense administració.

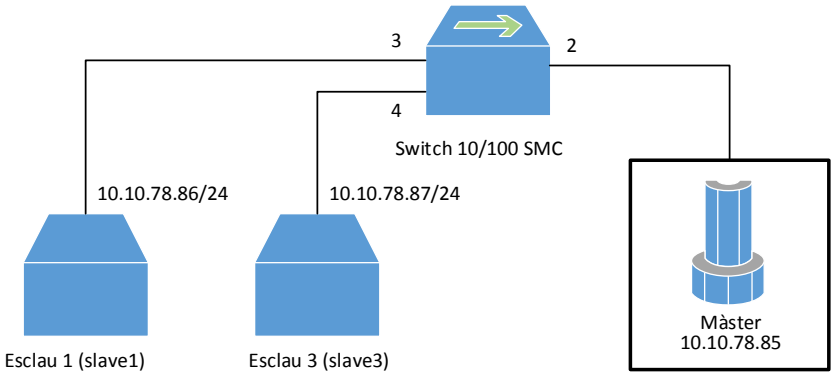
- Estudi d'eficiència d'execució d'aplicació MPI executant el gestor MESOS al clúster, contra una execució sense administració.

Per extreure conclusions sobre si el cost repercuteix de forma negativa en el rendiment general, o bé té un cost assumible davant de les avantatges de gestió dinàmica de recursos.

5.1 Disseny de la prova

En aquest punt del projecte, després de la instal·lació de MESOS i MPICH2, tenim configurat un clúster que permet llançar aplicacions de treball paral·lel sobre MPI, que a més té la següent particularitat:

- Tenim la possibilitat d'activar els computadors i l'anell de dimonis MPD per executar treballs de computació paral·lela sobre MPI amb la crida *mpiexec*, sense cap gestor de recursos, es a dir tot el treball de computació és realitza a l'anell MPD fins que aquest finalitza. A partir d'aquest moment anomenarem a aquesta forma d'execució en clúster MPD, que es mostra de forma resumida a la figura 36.



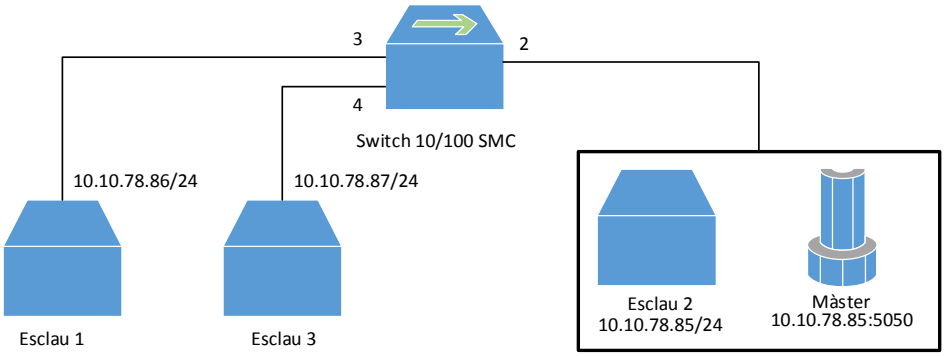

```
# Activació d'anell de dimonis MPD
$ mpdboot -n 3 -f mpd.hosts

# Contingut del fitxer mpd.hosts
slave1
slave3

# Activació d'una execució distribuïda
$ mpiexec -n 3 <binari a executar>
```

Figura 36. Configuració del clúster MPD.

- Podem també activar els computadors i el gestor de clúster MESOS, i una vegada activats llançar l'aplicació a calcular amb la crida específica *mpiexec-mesos*, que activarà l'anell de dimonis MPD fins que finalitza l'execució. A partir d'aquest moment aquesta forma d'execució l'anomenarem en clúster MESOS, que es mostra a la figura 37.




```
# Activar el node màster
$ ./bin/mesos-master.sh --ip= 10.10.78.85 --work_dir=/var/lib/mesos

# Activar node esclau.
$ ./bin/mesos-slave.sh –ip= <ip node esclau> --master=10.10.78.85:5050 --resources: "mem(*):1512;cpus(*):1" --no-switch_user

# Activar treball MPI a node màster
$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./<binari a executar>

# WEBUI
10.10.78.85:5050
```

Figura 37. Configuració del clúster MESOS.

Els objectius de l'apartat que ara s'inicia és estudiar les possibles diferències entre aquests dos clústers en el rendiment de funcionament i execució de problemes. L'estudi està dividit en dos parts:

1. Fer una prova de rendiment del clúster amb una suite de test compatible amb execucions MPI, traspàs de missatges MPI. Aquesta suite és PERFTEST-1.5[27] també de l'ANL. Com es veurà en el desenvolupament de les proves s'ha intentat trobar un escenari el més semblant en les dos execucions, per poder establir les possibles diferències d'una forma directa. El resum dels passos que s'executaran seran els següents:
   - Instal·lar la suite PERFTEST-1.5.
   - Triar les proves MPPTEST[v] d'interès que s'inclouen en la suite.
   - Executar-les al clúster MPD i al clúster MESOS.
   - Comparar de forma gràfica els resultats amb GNUPLOT[w].
   - Extreure conclusions sobre el rendiment del clúster en els dos escenaris anteriors.

2. L'objectiu d'aquesta segona part és estudiar la influència que té l'execució d'un problema paral·lel al clúster amb la participació de gestió de MESOS i sense. L'estudi és realitza a partir de compilar i enllaçar el codi de mètrica NPB[x] que s'ha triat amb les opcions de MPE[y], disponibles al compilador MPICC, i obtenir una sortida gràfica del desenvolupament de computació a cada processador gràcies a l'aplicació JUMPSHOT també disponible a MPICC. El resum de passos que s'executaran són els següents
   - Instal·lar NPB i triar la mètrica d'execució
   - Compilar la mètrica amb les opcions que permeten el seu anàlisis d'execució (llibreries MPE)
   - Executar la mètrica al clúster MPD i clúster MESOS.
   - Comparar de forma gràfica els resultats amb JUMPSHOT.
   - Extreure conclusions sobre el rendiment del clúster en els dos escenaris possibles.

Finalment es farà un resum i conclusions dels rendiments observats per analitzar quin dels dos clústers és més eficient, si és que les diferències ens porten aquesta conclusió.

[v] Programa que mesura el rendiment de les rutines bàsiques MPI en diverses situacions. Les rutines en aquest cas s'obtenen de la suite PERFTEST-1.5.
[w] Programa multiplataforma que permet gràficar dades introduïdes per línia de comanda.
[x] Nas Parallel Benchmark (NPB) de Nasa Advanced Supercomputing Division. Consisteix en una suite d'aplicacions que serveixen per avaluar el rendiment de computadors paral·lels. http://www.nas.nasa.gov/publications/npb.html
[y] Extensions que faciliten eines d'anàlisis, *debugging tools*, per a la programació MPI.

5.2 Prova de rendiment basada en pas de missatges MPI

Com s'ha comentat a la introducció, en aquesta part s'estudiarà com reacciona el clúster al pas de missatges MPI entre els nodes per avaluar la seva eficiència com a clúster MPD o clúster MESOS. S'esperen resultats molt similars per què el clúster MESOS és bàsicament un clúster MPD, executa per sota un anell MPD, però amb la possibilitat de gestionar recursos de forma dinàmica. De forma intuïtiva aquesta càrrega extra, si és important, s'hauria de veure reflectida en les proves que ara es presenten, i penalitzant al clúster MESOS.

5.2.1 Instal·lació de la suite PERFTEST-1.5

La primera gestió és la de preparar l'escenari de proves. Com s'ha comentat ja tenim un clúster que executa tasques MPI, ara es tracta d'instal·lar la suite de proves de comunicacions PERFTEST-1.5 que permet l'execució de les crides MPPTEST i GOPTEST. Aquests són programes que mesuren el rendiment de les rutines bàsiques de pas de missatges de la llibreria MPI, tot i que també pot examinar escenaris més complexos per depurar errors de rendiment (escalabilitat de problemes, mesures de missatges, ...). El seu procediment d'estudi bàsic és un test *ping-pong*, de forma que és mesura el temps que triga l'enviament d'un missatge d'un procés a un altre i el seu retorn. Per tant el paquet a instal·lar és PERFTEST-1.5[z], que conté MPPTEST i GOPTEST i és una suite de proves de rendiment que es pot executar per a qualsevol distribució MPI, no només la versió MPICH2 de l'ANL. Els passos són senzills:

1. Descarregar la versió de PERFTEST-1.5:

   ftp://ftp.mcs.anl.gov/pub/mpi/tools/perftest.tar.gz

2. Descomprimir-la en el directori compartit NFS /mirror:

   $ /mirror/tar -zxf perftest.tar.gz

   Que crea el directori amb la suite ja descomprimida:

   $ /mirror/perftest-1.5

3. Executar la instal·lació de la suite, des del directori /mirror/perftest-1.5, indicant en quin directori està instal·lat MPICH2:

   $ ./configure - - with-mpich=/mirror/mpich2-install
   $ make
   $ make install

4. Incloure el path a la suite, per que es pugui trobar el programa MPPTEST i GOPTEST, al fitxer *.bashrc* de $HOME de tots els nodes:

   vi .bashrc

   I afegim al final del fitxer:
   export PATH=/mirror/perftest-1.5:PATH$

[z] ftp://ftp.mcs.anl.gov/pub/mpi/tools/perftest.tar.gz

5.2.2 Escenaris de prova

Els escenaris que s'han de fer servir són diferents, segons si s'executa el clúster MPD o el clúster MESOS:

En el cas d'execució a clúster MPD:

1. S'ha d'iniciar l'anell MPD i verificar la seva posada en marxa (més detalls a la secció 4.4.3):

```
$ mpdboot –n <processadors> -f mpd.hosts
$ mpdtrace -l
```

2. La crida d'execució mínima és la següent, per exemple per les crides bloquejants:

```
$ mpiexec –n 3 mpptest –size 0 4096 32 –gnuplot –fname <fitxer gnuplot>
```

   Els processadors seran com a màxim 3, per ser el límit que imposa l'execució de clúster MESOS. La mida del test serà de 128 passos, amb distància entre ells de 32 bytes, fins que la mesura sobrepassa els 4096 bytes que és quan el test finalitzarà.

   L'execució generarà dos fitxers compatibles amb Gnuplot, amb mateix nom i distinta extensió, per permetre el traçat d'un gràfic en entorn X11:

   - **fitxer gnuplot**, que conté l'script d'execució que permet a Gnuplot traçar el gràfic amb les dades numèriques del fitxer .gpl.

   - **fitxer gnuplot.gpl**, que conté les dades numèriques que traspassa l'execució d'MPPTEST o GOPTEST.

En el cas d'execució a clúster MESOS:

1. Primer s'han d'iniciar el màster i tots els esclaus del clúster MESOS i verificar la seva posada en marxa a través de l'administrador WEBUI (més detalls a la secció 4.2).

2. La crida d'execució és la següent:

```
$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./<prova rendiment>
```

   Les característiques de MESOS fan que la crida *mpiexec-mesos* executa un entorn d'aplicació MPI per a la prova de rendiment que es passa com a paràmetre. El problema és que la prova de paràmetre ha d'incloure apart de la crida MPPTEST i GOPTEST la resta de dades per fer gràfiques, i la prova de la suite en concret. Aquesta crida era resolta de forma fàcil al clúster MPD, ja que s'inclou tota a la mateixa línia de comandes. En aquest cas la dificultat se soluciona creant un script executable que encapsula les crides i els paràmetres. Per exemple per a la prova per defecte MPPTEST següent:

```
$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./mpptest
```

   On mpptest és un script, ha de tenir permisos d'execució, que conté el següent codi:

```
#!/bin/bash
mpptest –size 0 4096 32 –gnuplot –fname mpptest_mesos
```

   En aquest cas els fitxers Gnuplot que es generen són mpptest_mesos i mpptest_mesos.gpl.

Per defecte l'execució sempre es farà amb el màxim de recursos del clúster[28] per facilitar la comparació de resultats:

- **Processadors**: només tres unitats, tot i que segons el gestor MESOS hi han disponibles 6 processadors. Aquest valor de processadors que indica MESOS s'obté a causa de combinar màquines de característiques diferents com són el màster, en configuració processador *multicore* amb quatre nuclis, i les màquines esclaves que només disposen d'un processador d'un nucli. D'aquesta suma s'obtenen els sis processadors que indica el WEBUI. El limitant de només tres processadors està directament relacionat amb els dimonis MPD que executaran, que és un per màquina, per tant tres processadors que és el valor per defecte. Si s'estudia la crida *mpiexec-mesos* sí és possible indicar quantes CPUs es poden utilitzar per MPD, però seria d'aplicació en clústers homogenis a nivell de processador i aquest no és el cas, ja que és combinen multicore i no multicore. Per tant els dos clústers queden limitats a utilitzar tres processadors en total per a MPI, ja que els MPD de menys recursos només disposen d'un processador. Idealment la solució passaria per què la crida fos individual per màquina, i sobre els seus recursos, que de fet s'anuncien a l'iniciar un esclau (veure secció 4.2), però mpiexec-mesos s'executa sobre el màster i des d'aquí s'inicia l'anell MPD homogeni, de forma que no podem indicar el valor host/processadors de forma individual sinó que és una crida general.

- **Memòria**: s'ha triat el valor de 1512 MB per computador, per ser el valor lliure que ha permès fer les proves de clúster MESOS sense problemes, deixant suficient memòria per a l'execució dels serveis dels sistemes operatius. Aquesta limitació només s'aplica al clúster MESOS, el clúster MPD opera sense que s'especifiqui quina quantitat de memòria podrà utilitzar, per tant és possible que disposi d'una mida més gran, però en tot cas un valor d'una quarta part lliure (75% d'ocupació) sembla un valor realista en execució. Aquesta dada s'ha de tenir en compte a l'analitzar els rendiments si les diferències són molt evidents.

5.2.3 Execució de les proves d'eficiència MPI

Finalment es presenten el total de proves de comunicació que s'han triat per avaluar els protocols de comunicació en el clúster físic, sigui en operació MPD o MESOS. Primer presentem les crides en format clúster MPD com a exemple d'execució, amb valors per defecte, i una vegada entesa, la diferència de format respecte a clúster MESOS. Després es mostren els resultats gràfics de l'execució, dades que es poden consultar de forma integral a l'Annex. De cada prova també s'ha inclòs el resultat d'execució dels entorns d'execució MPI (*framework*), tal i com ho mostra l'administrador WEBUI.

1. **Test mpptest**
   Aquesta és la prova per defecte, i es mesura l'enviament i recepció bloquejant de missatges entre dos processos que utilitzen les funcions MPI MPI_Send() i MPI_Recv(), i que no retornen dades fins que la comunicació no ha finalitzat. D'aquesta forma es poden utilitzar els recursos presents a la crida de comunicació. El test és pot interpretar com l'enviament de missatges entre nodes d'una pregunta i l'obtenció de resposta (sol parlar-se de proves *ping-pong)*. L'execució al clúster MPD és pot fer de dos formes diferents:

   ```
   $ mpiexec –n 3 mpptest –size 0 4096 32 –gnuplot –fname mpptest_mpd.mlp
   ```

   O bé:

   ```
   $ mpiexec –n 3 mpptest –sync –size 0 4096 32 –gnuplot –fname mpptest_mpd
   ```

   En execució al clúster MESOS:

   ```
   $ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./mpptest
   ```

   On mpptest és un script, ha de tenir permisos d'execució, que conté el següent codi:

   ```
   #!/bin/bash
   mpptest –size 0 4096 32 –gnuplot –fname mpptest_mesos
   ```

L'execució que reporta MESOS WEBUI al clúster:

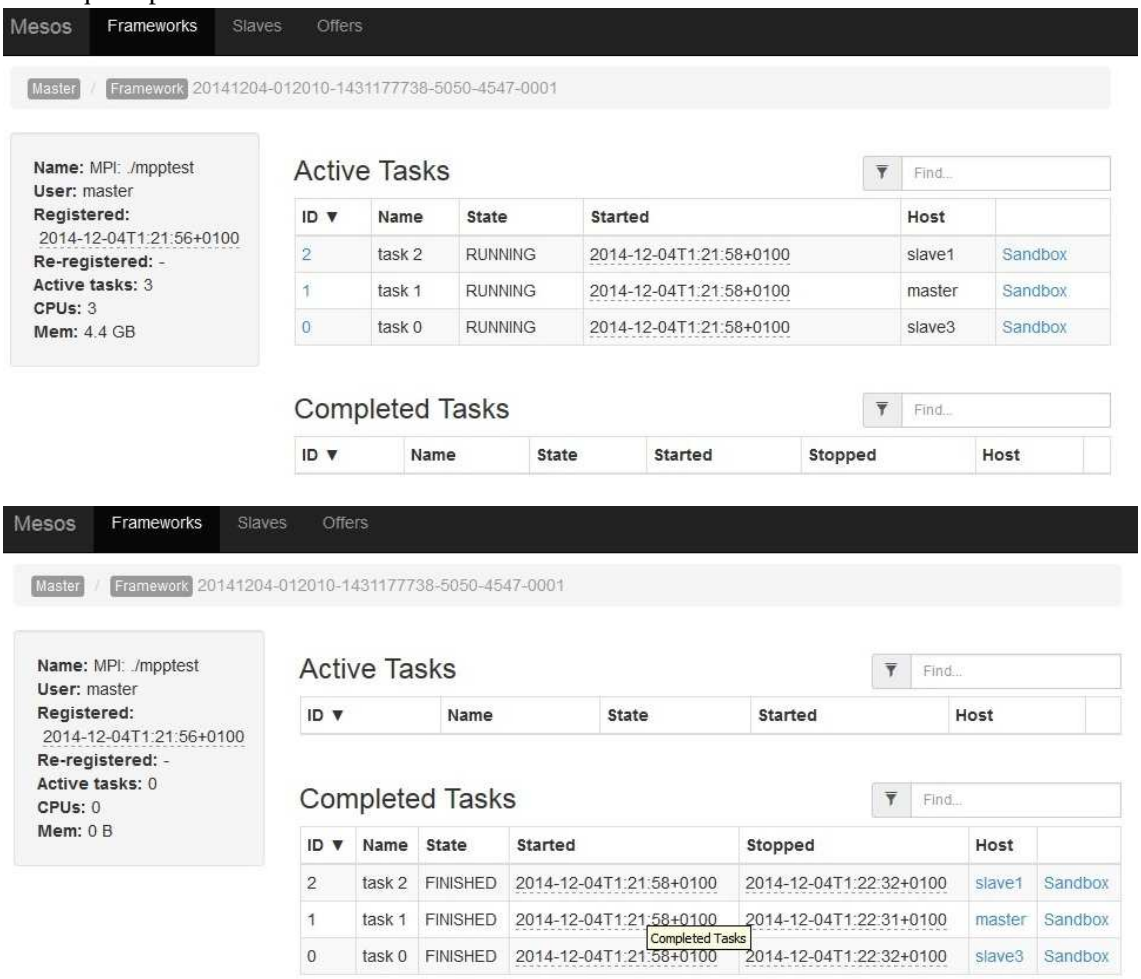


Figura 38. Execució de crida MPPTEST al clúster MESOS. WebUI mostra l'activació i posterior finalització de tasques als nodes esclaus.

El resultat comparat de l'execució als dos clústers es pot veure en la figura 39.

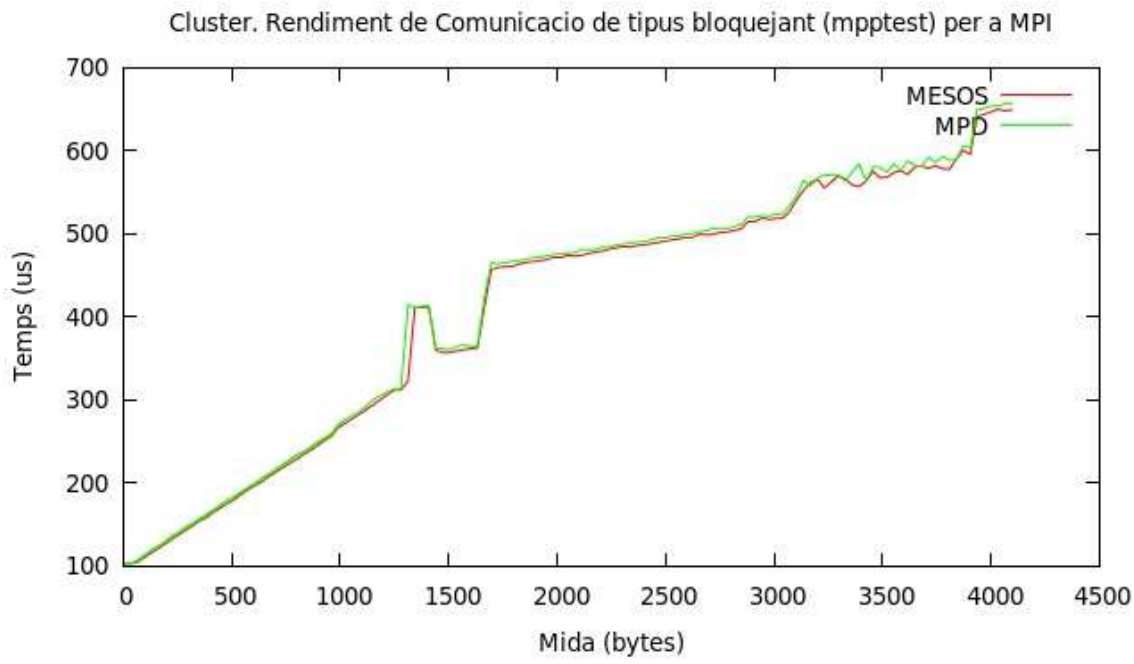


Figura 39. Resultat d'execució MPPTEST als dos clústers.

El test de missatges bloquejants entre nodes retorna un funcionament pràcticament idèntic als dos clústers.

2. Test async

En aquest prova es mesura l'enviament i recepció no bloquejant de missatges entre dos processos que utilitzen les funcions MPI MPI_Isend() i MPI_Irecv(), i que retornen dades fins i tot quan la comunicació no ha finalitzat. L'execució al clúster MPD es fa de la següent forma:

```
$ mpiexec –n 3 mpptest –async –size 0 4096 32 –gnuplot –fname async_mpd
```

L'execució al clúster MESOS:

```
$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./async
```

On async és un script, ha de tenir permisos d'execució, que conté el següent codi:

```
#!/bin/bash
mpptest –async –size 0 4096 32 –gnuplot –fname async_mesos
```

L'execució que reporta MESOS WEBUI al clúster:

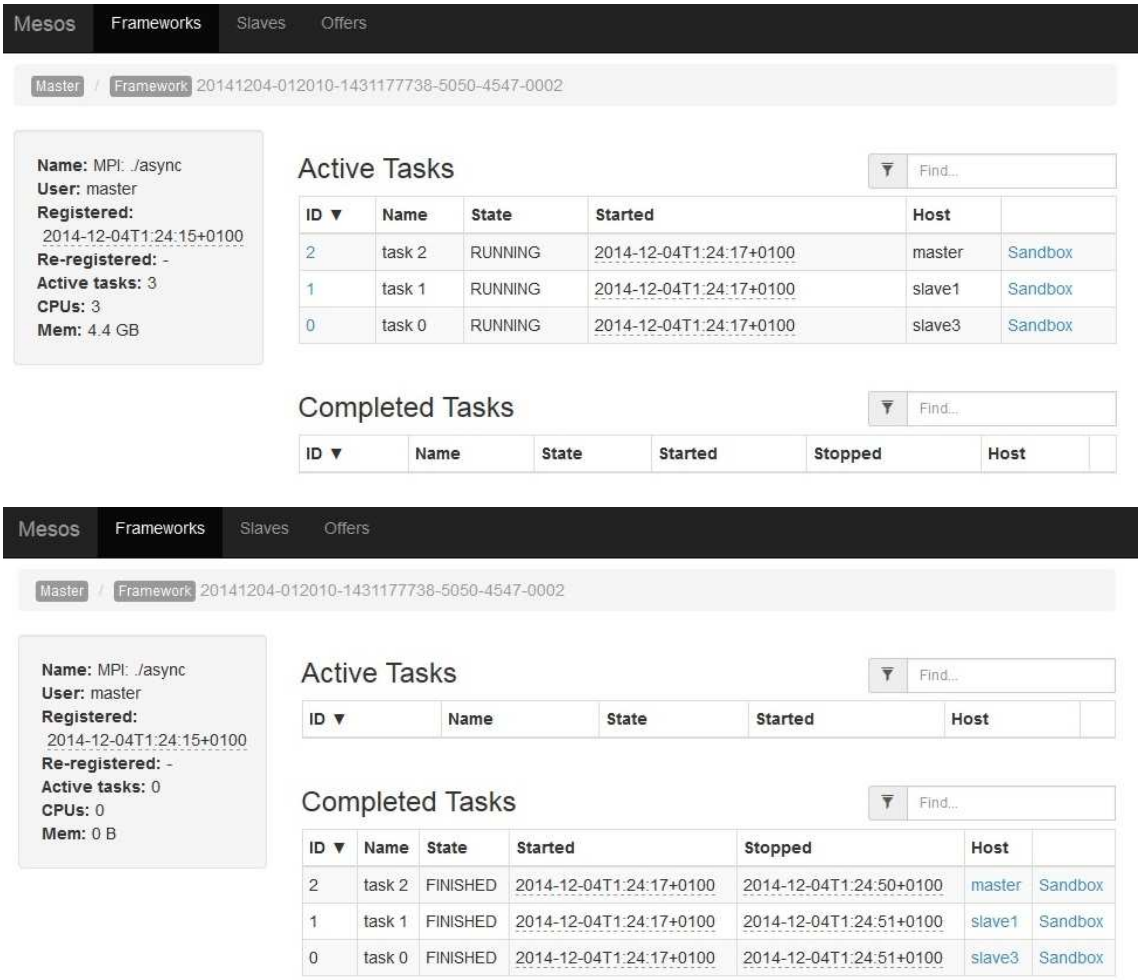


Figura 40. Execució de crida ASYNC al clúster MESOS. WebUI mostra l'activació i posterior finalització de tasques als nodes esclaus.

El resultat comparat de l'execució als dos clústers:

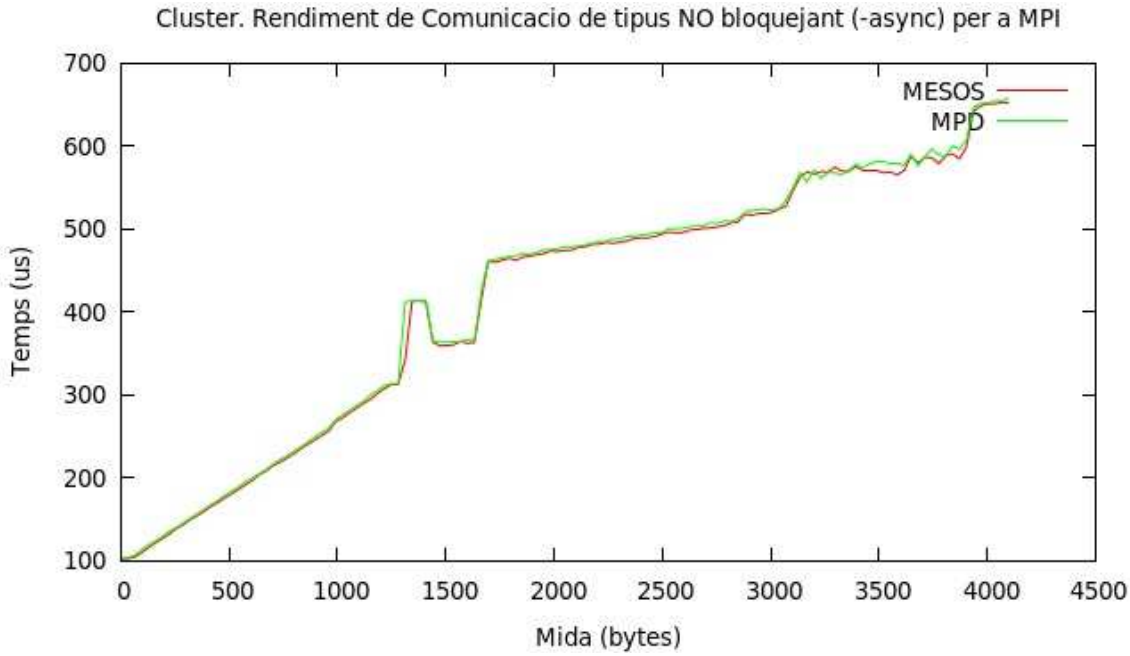


Figura 41. Resultat d'execució ASYNC als dos clústers.

El test de missatges no bloquejants entre nodes retorna un funcionament pràcticament idèntic als dos clústers, i pràcticament igual al test anterior amb missatges bloquejants.

3. Test bisect

En aquesta prova es produeix un enviament de missatges entre dos parells de processos amb l'objectiu de mesurar el *bisection bandwith*[aa] del sistema, o la capacitat de moure informació d'un extrem a l'altre del sistema. L'execució al clúster MPD es fa de la següent forma:

```
$ mpiexec –n 3 mpptest –bisect –size 0 4096 32 –gnuplot –fname bisect_mpd
```

L'execució al clúster MESOS:

```
$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./bisect
```

On bisect és un script, ha de tenir permisos d'execució, que conté el següent codi:

```
#!/bin/bash
mpptest –bisect –size 0 4096 32 –gnuplot –fname bisect_mesos
```

[aa] Es coneix com l'ample de banda màxim entre les dos parts que formen la xarxa del sistema, si aquesta és divideix en dos parts iguals. http://en.wikipedia.org/wiki/Bisection_bandwidth

L'execució que reporta MESOS WEBUI al clúster:

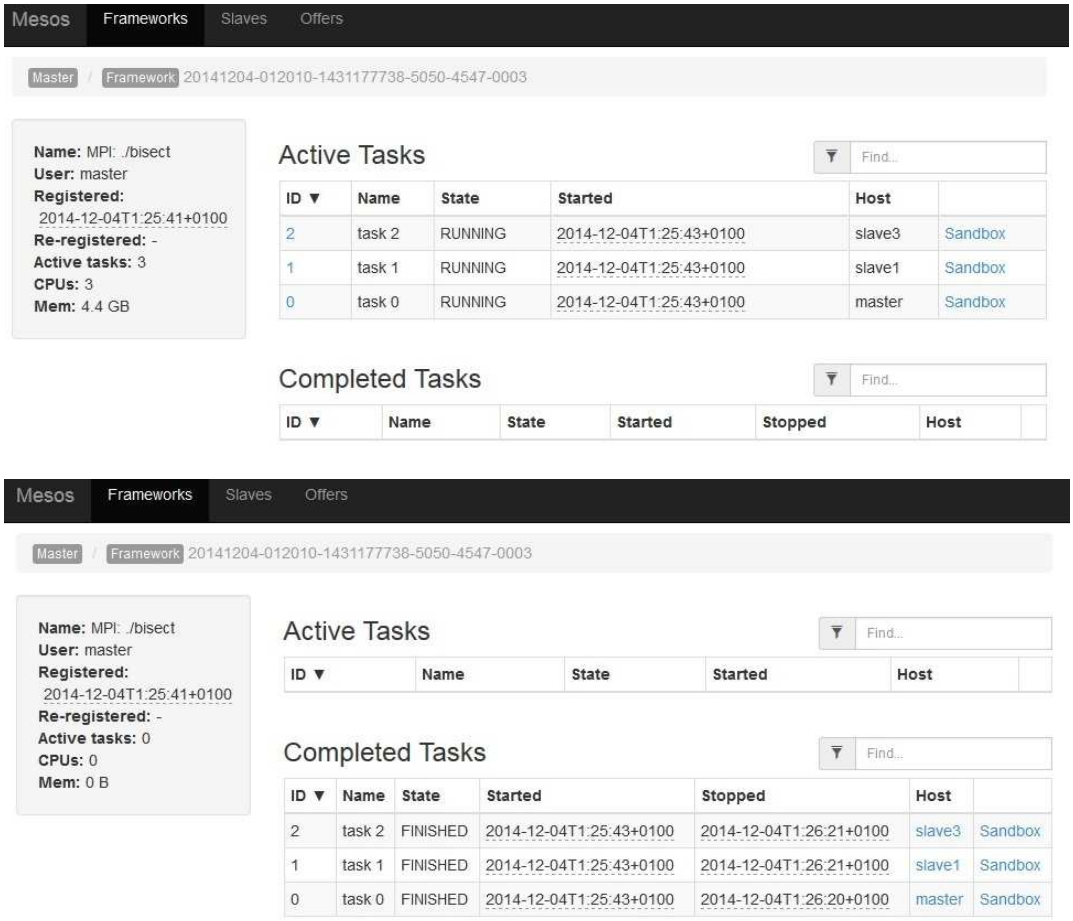


Figura 42. Execució de crida BISECT al clúster MESOS. WebUI mostra l'activació i posterior finalització de tasques als nodes esclaus.

El resultat comparat de l'execució als dos clústers:

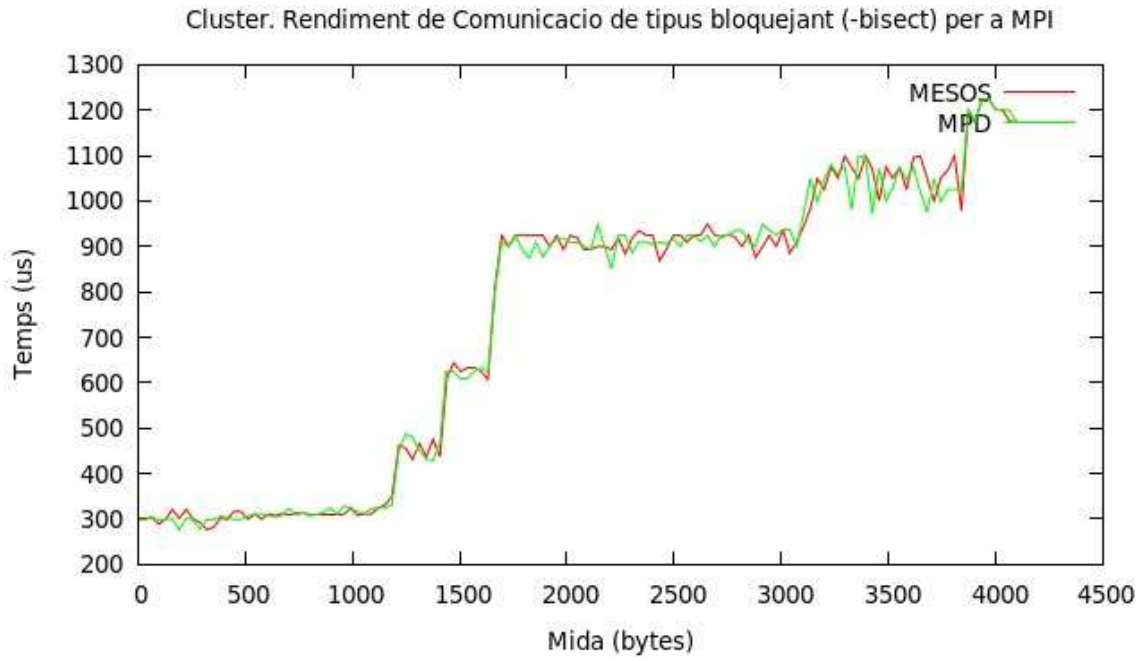


Figura 43. Resultat d'execució BISECT als dos clústers.

En aquest test els comportament del clúster MPD i MESOS són pràcticament idèntics.

4. Test overlap

En aquesta prova s'intercala tasques de computació de càrrega variable amb tasques de comunicació amb missatges de mida variable. L'execució al clúster MPD es fa de la següent forma:

$ mpiexec –n 3 mpptest –overlap –size 0 4096 32 –gnuplot –fname overlap_mpd

L'execució al clúster MESOS:

$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./overlap

On overlap és un script, ha de tenir permisos d'execució, que conté el següent codi:

```
#!/bin/bash
mpptest –overlap –size 0 4096 32 –gnuplot –fname overlap_mesos
```

L'execució que reporta MESOS WEBUI al clúster:

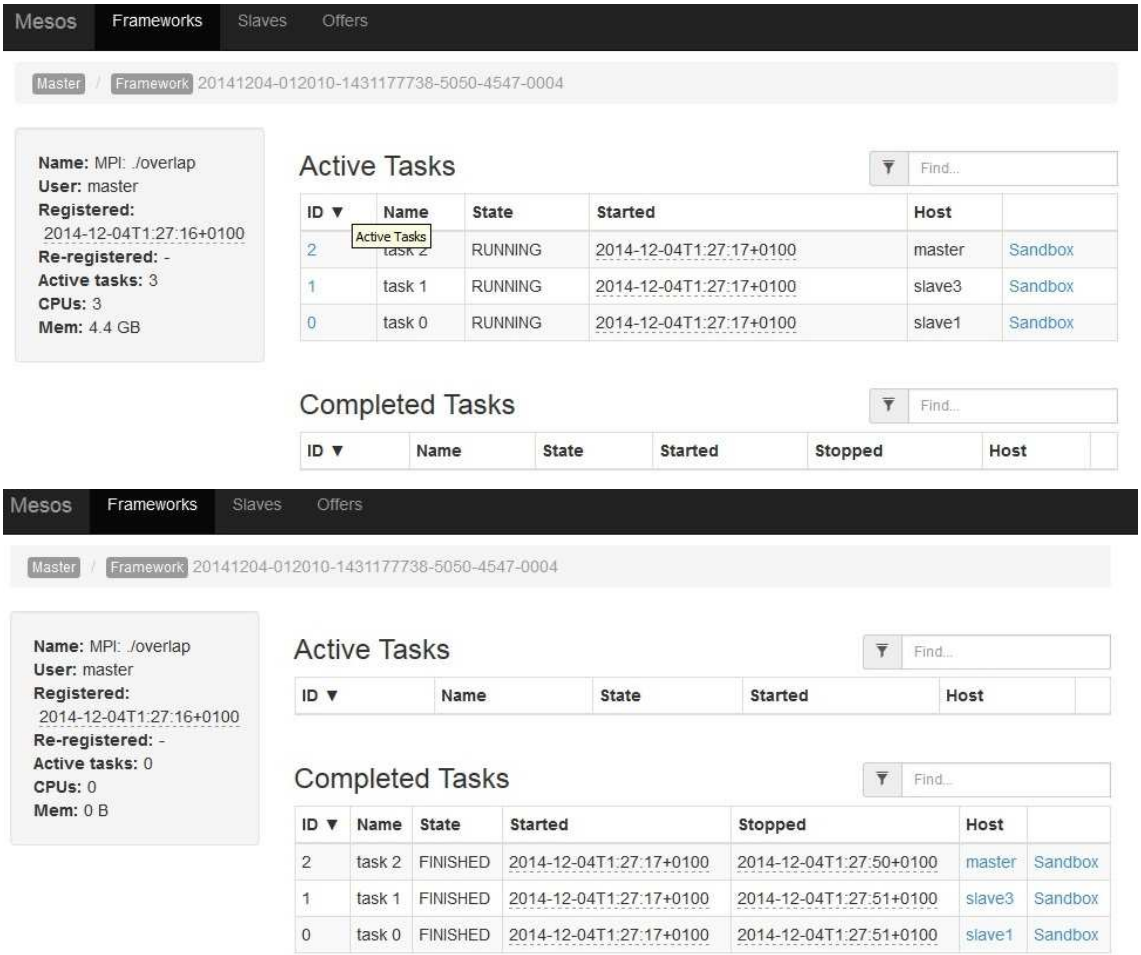


Figura 44. Execució de crida OVERLAP al clúster MESOS. WebUI mostra l'activació i posterior finalització de tasques als nodes esclaus.

El resultat comparat de l'execució als dos clústers:

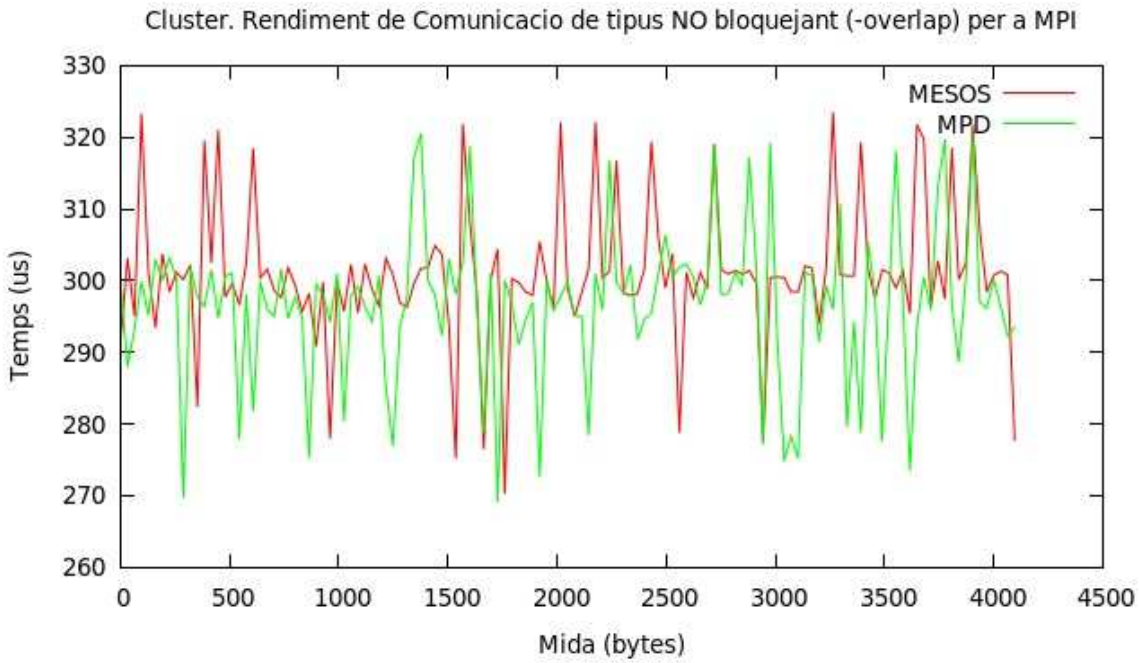


Figura 45. Resultat d'execució OVERLAP als dos clústers.

El test de computació i comunicació als clústers també indica una execució molt similar.

5. Test logscale

En aquesta prova s'utilitzen missatges de mida 2^i amb i:{0,..,17} fins 128 KB. L'execució al clúster MPD es fa de la següent forma:

```
$ mpiexec –n 3 mpptest –logscale –gnuplot –fname logscale_mpd
```

L'execució al clúster MESOS:

```
$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./logscale
```

On logscale és un script, ha de tenir permisos d'execució, que conté el següent codi:

```
#!/bin/bash
mpptest –logscale –gnuplot –fname logscale_mesos
```

L'execució que reporta MESOS WEBUI al clúster:

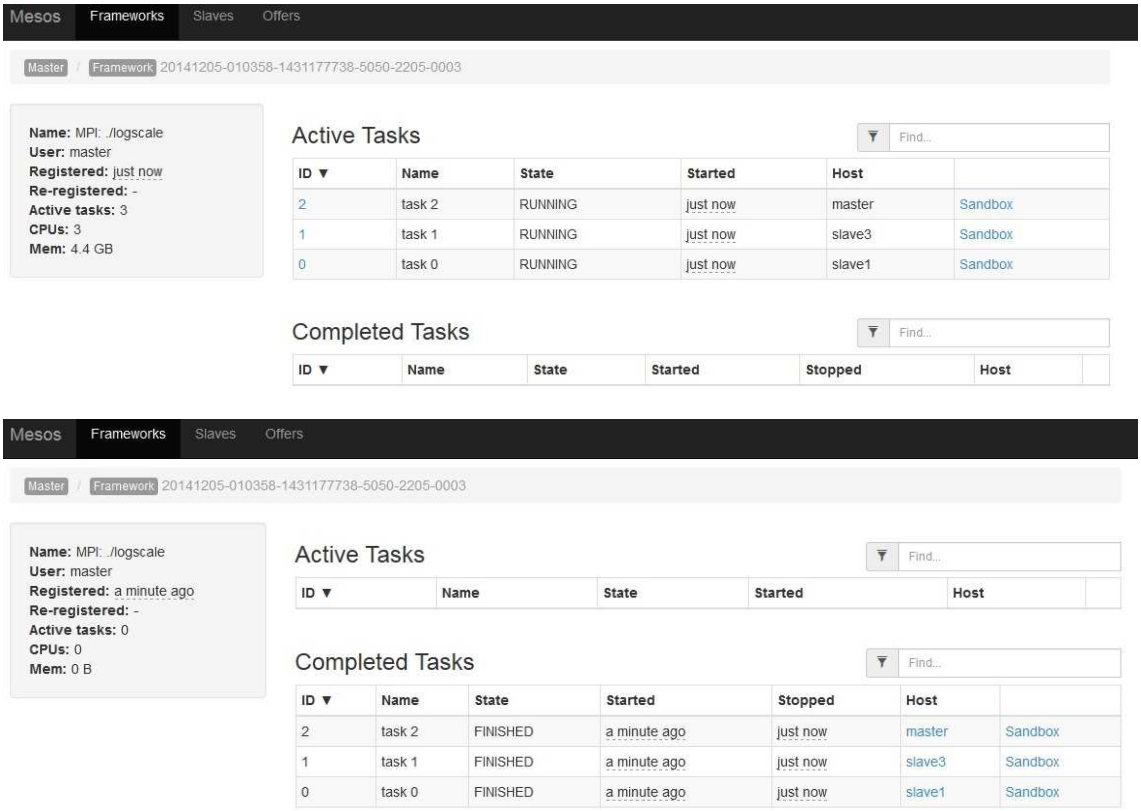


Figura 46. Execució de crida LOGSCALE al clúster MESOS. WebUI mostra l'activació i posterior finalització de tasques als nodes esclaus.

El resultat comparat de l'execució als dos clústers:

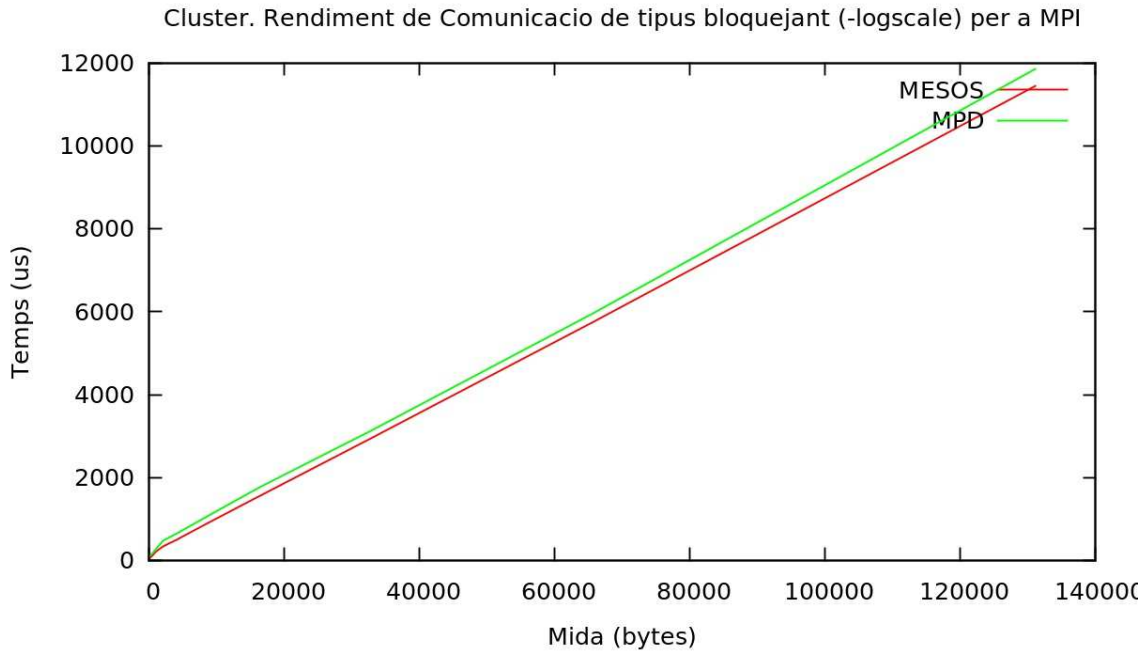


Figura 47. Resultat d'execució LOGSCALE als dos clústers.

El test de comunicació bloquejant amb paquets de mida cada cop més grans en funció de 2^i, torna un resultat pràcticament igual, una mica de comportament pitjor el clúster MPD. La diferència més gran és de $408 \cdot 10^{-6}$ S en paquets de 128 KB.

6. Test goptest

La prova mesura la comunicació col·lectiva entre els nodes, totes les anteriors són punt a punt, per aquest motiu s'utilitza un programa diferent a MPPTEST que és GOPTEST, el funcionament és similar però es defineix una llista de mides de missatge en bytes {32,256,512,1024}. L'execució s'ha de fer dos cops per analitzar la diferència d'execució amb dos i tres nodes de forma que es pugui observar l'efecte d'un clúster amb més o menys nodes. L'execució al clúster MPD es fa de la següent forma:

$ mpiexec –n 2 goptest -isum –sizelist 32,256,512,1024 –gnuplot –fname goptest2_mpd

$ mpiexec –n 3 goptest -isum –sizelist 32,256,512,1024 –gnuplot –fname goptest3_mpd

Els fitxers de dades s'hauran de modificar per incloure les dades de l'execució amb 2 nodes i 3 nodes en un únic fitxer que anomenarem goptest_mpd. A l'annex es pot veure el contingut d'aquest fitxer.

L'execució al clúster MESOS:

$ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./goptest

On logtest és un script, ha de tenir permisos d'execució, que conté el següent codi:

```
#!/bin/bash
goptest -isum –sizelist 32,256,512,1024 –gnuplot –fname goptest_mesos
```

En aquest cas es fa primer una execució amb tots els nodes i quan finalitza es canvia el nom de fitxer a goptest3_mesos.gpl, i es dona de baixa un dels nodes esclaus. Llavors s'inicia una nova execució per tenir les dades d'execució en un fitxer goptest_mesos.gpl que s'ha de modificar per tenir les dades de les dos execucions agrupades. A l'annex es pot veure el contingut d'aquest fitxer.

L'execució que reporta MESOS WEBUI al clúster:

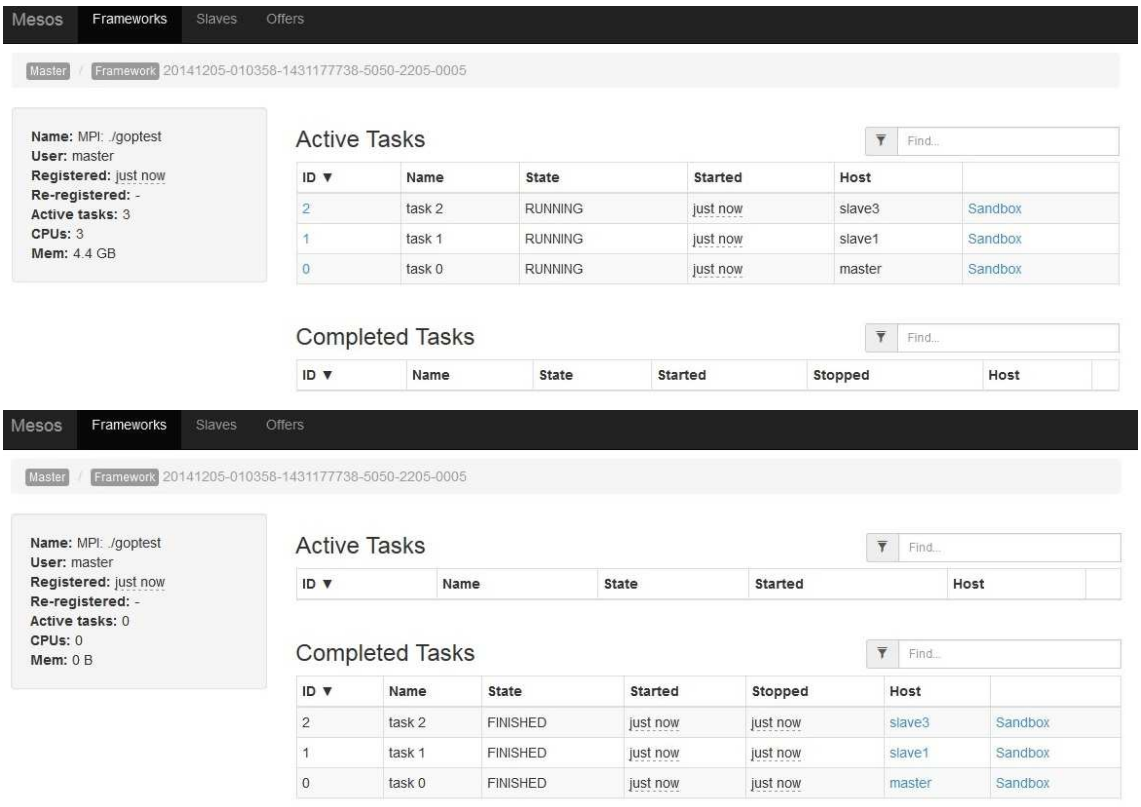


Figura 48. Execució de crida GOPTEST al clúster MESOS. WebUI mostra l'activació i posterior finalització de tasques als nodes esclaus.

El resultat comparat de l'execució als dos clústers el podem veure a continuació:

Test **goptest** en clúster MESOS

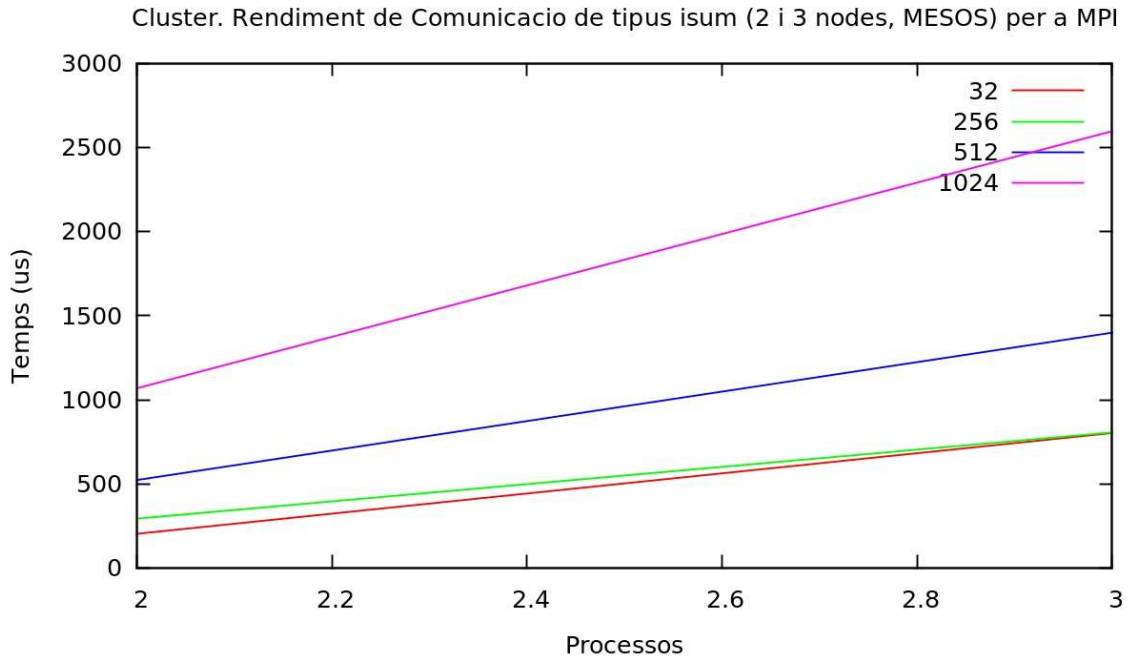


Figura 49. Resultat d'execució GOPTEST al clúster MESOS.

Test **goptest** en clúster MPD

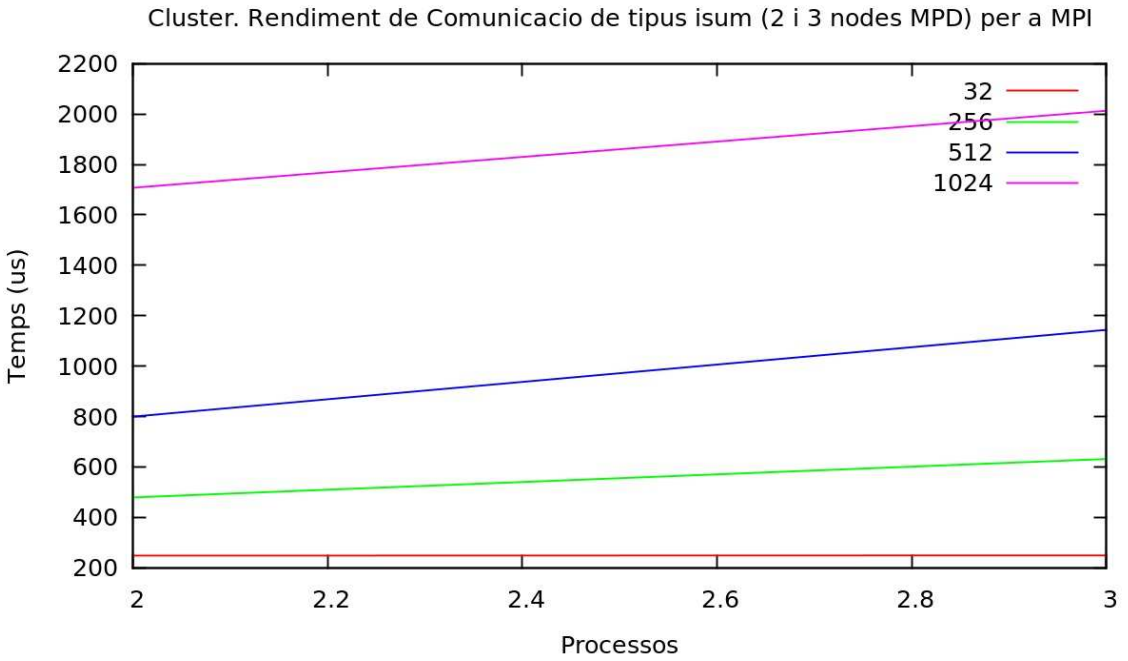


Figura 50. Resultat d'execució GOPTEST al clúster MPD.

Els resultats extrets dels fitxers goptest_mpd.gpl i goptest_mesos.gpl:

Nodes	Paquet (bytes)	Clúster MESOS Temps (uS)	Clúster MPD Temps (uS)
2	32	203,79	249,38
	256	294,05	480
	512	523,33	800,35
	1024	1.068,80	1.707,30
3	32	802,44	250,45
	256	806,19	632,51
	512	1.398,76	1.144,06
	1024	2.596,04	2.012,43

Taula 4. Resultats GOPTEST als clústers MESOS i MPD.

Amb dos nodes el clúster MESOS és més eficient per a totes les mides de paquet, la diferència més gran és de $639 \cdot 10^{-6}$ S en paquets de 1024 bytes. En canvi per a tres nodes el clúster més eficient és el MPD, amb la diferència de $584 \cdot 10^{-6}$ S en paquets de 1024 bytes.

7. Consum de recursos

Finalment es vol mostrar una captura del programa TOP, en un dels esclaus, per mostrar els recursos del sistema quan s'executa l'anell MPD o el gestor MESOS.

Execució clúster MPD:

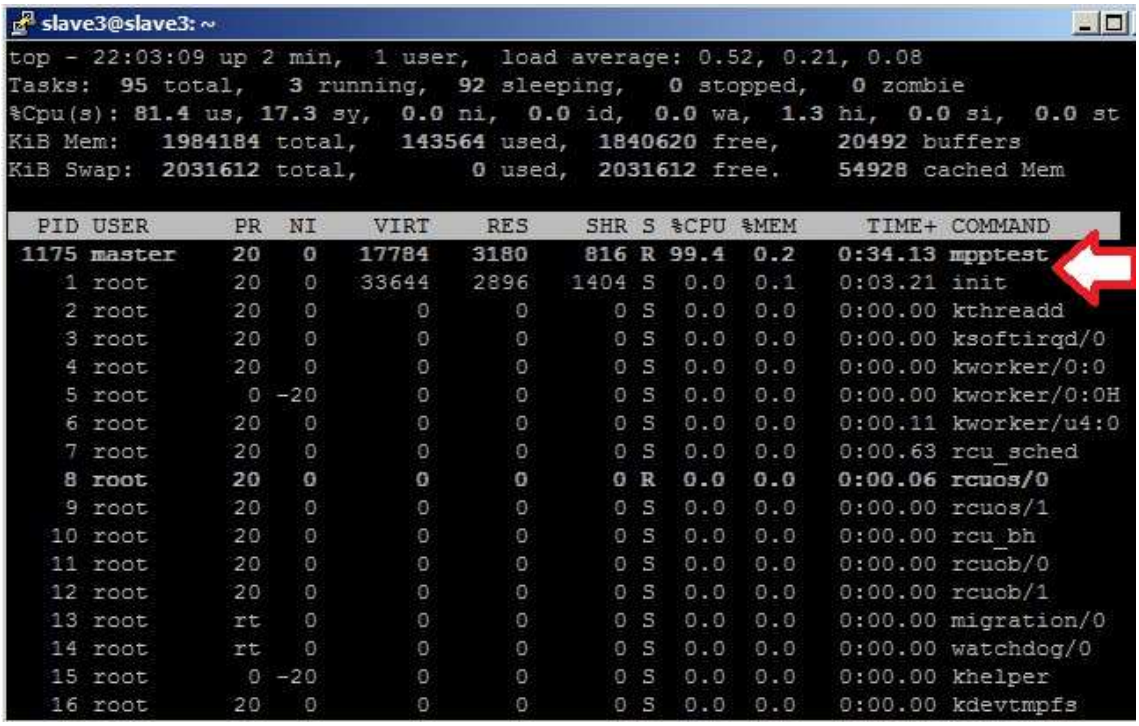


Figura 51. Consum de recursos, que informa TOP, en l'execució d'un node del clúster MPD.

Execució clúster MESOS:

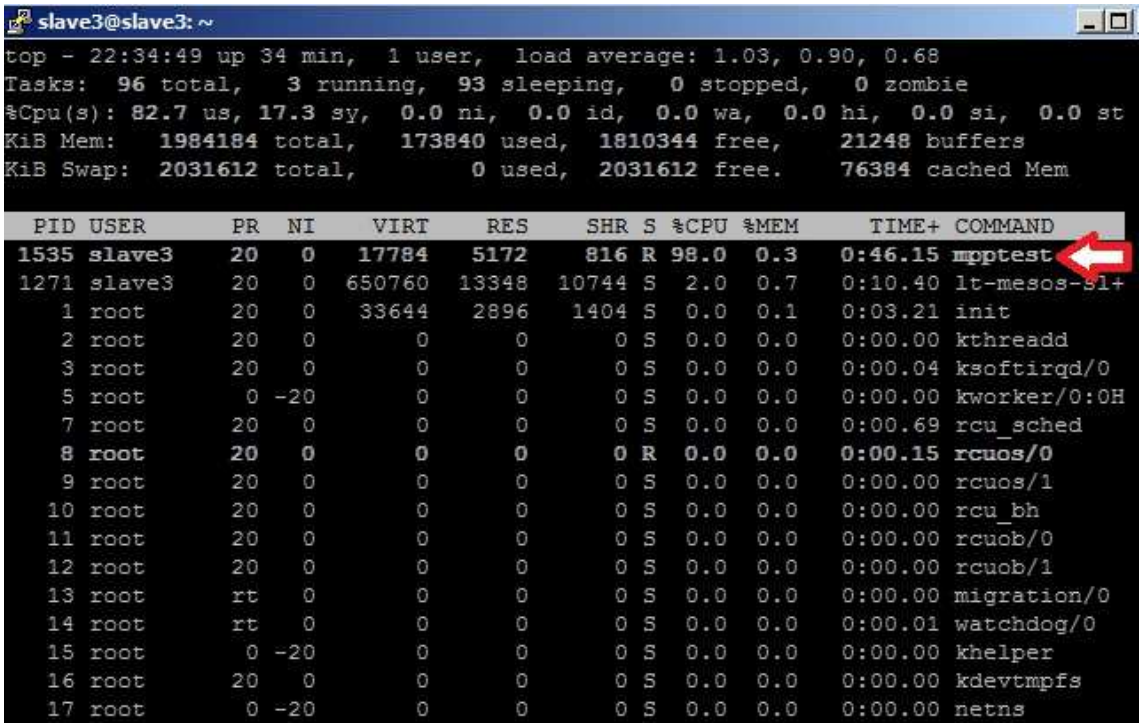


Figura 52. Consum de recursos, que informa TOP, en l'execució d'un node del clúster MESOS.

En el cas d'execució al clúster MPD el host esclau (slave3) dedica un 0,2% de memòria i 99,4% de temps de CPU a executar el programa MPPTEST. En canvi a l'execució MESOS la capa MESOS (lt-mesos-sl+) necessita 0,7% de memòria i 2% de temps de CPU, de forma que el programa MPPTEST només pot disposar d'un 98% de temps de CPU, la memòria no està penalitzada ja que pot disposar de 0,3%. Així les diferències en recursos d'execució entre els dos clústers són molt similars, i només MPD té 1,4% més de temps de processador, a nivell de memòria no hi ha cap limitant a cap dels dos processos i disposen de tanta com necessiten.

5.2.4 Conclusions de les proves d'eficiència MPI

Les mesures dels dos clúster confirmen la hipòtesi inicial, que indicava que MESOS en execució MPI havia de tornar un resultat molt similar a l'execució sobre anell MPD, ja que aprofita la crida *mpiexec* del paquet MPICH2, que és la mateixa que executa l'anell MPD[bb]. A més:

- En el cas d'execució al clúster MESOS l'aprofitament dinàmic dels recursos succeeix a l'executar-la, que es podran afegir tots els nodes que estiguin en disposició de recursos, almenys una CPU i 1 GB de memòria o la que s'indiqui en l'execució, en aquest cas era de 1512 MB. Una vegada s'ha iniciat l'execució en el clúster MESOS no es poden afegir nous recursos de forma dinàmica, s'ha d'esperar a la finalització i tornar a fer l'execució de nou, moment que sí es podran afegir. Altrament són rebutjats pel màster per tenir suficients recursos d'execució. El funcionament és molt similar al clúster MPD on s'inicia l'anell de dimonis amb tots els hosts possibles, i sobre aquest anell s'executen les aplicacions MPI, una darrera l'altra, fins que es decideix aturar l'anell i sí és el cas reconfigurar-lo. La diferència rau en que el clúster MPD a l'iniciar-se, recordem l'execució de *mpdboot*, no permetrà afegir nous nodes si no és que aturem tot l'anell MPD i el tornem a iniciar abans d'executar un nou programa MPI. En canvi el clúster MESOS a cada nova execució

[bb] Aquesta característica es pot observar analitzant el codi de mpiexec-mesos.py, que s'inclou en l'Annex o bé directament a $HOME/mesos.0.20.0/mpi/mpiexec-mesos.py.

d'un programa MPI es podran afegir automàticament tots els nodes que estiguin disponibles, ja que en el moment de finalització executa un *mpdallexit*, i en el moment previ a la crida a *mpiexec* inicia també un anell de dimonis MPD moment que es pot aprofitar per redefinir amb nous recursos quants dimonis MPD es poden executar.

Les execucions han tornat diferències com també s'apuntava al principi, ja que la gestió del clúster MESOS, tant màster com esclau, implica una càrrega extra de recursos a més d'executar un dimoni MPD per màquina. Tot i que aquesta càrrega no té un gran efecte segons es pot veure a les diferents gràfiques de rendiment i als recursos d'execució disponibles. Un apunt a tenir en compte és que a les mesures s'ha intentat tenir els mínims recursos en execució en el moment de fer les proves, per poder fer la comparació que es presenta. En el cas del clúster MESOS s'ha fixat un valor de memòria de 75% del màxim disponible, i al clúster MPD no ha estat possible fixar-ho per tant és el 100% disponible. Valors que s'han revelat més que suficients, i que no han implicat diferències de recursos d'un tipus d'execució a l'altre a causa de:

- Un valor molt lleuger dels gestors MESOS

- El suite de proves executats requereix de poques necessitats de memòria

Per tant amb les proves realitzades és pot concloure que la gestió del clúster MESOS no implica un pitjor rendiment que en un clúster MPD, i són pràcticament iguals.

La secció següent intentarà comparar l'eficiència d'execució d'un programa també sobre els dos clúster, i segons els resultats anteriors l'eficiència hauria de ser similar.

5.3 Prova de rendiment basada en aplicació MPI

A la secció anterior s'ha volgut mesurar l'eficiència del clúster en les configuracions MESOS i MPD per a comunicacions MPI, per tal de trobar una diferència significativa, bé per bona gestió de recursos o bé per disposar de menys, o bé un resultat força similars com així ha estat, i que concorden amb el fet que l'entorn d'aplicació MPI de MESOS és un *kernel* que executa al final la mateixa crida que un anell de dimonis MPD. En aquest apartat l'objectiu és executar una aplicació real que utilitza crides MPI, sobre els dos clústers i comparar els resultats obtinguts. L'aplicació que s'ha triat forma part dels NPB[cc] (*NAS Paral·lel Benchmark*), que és una suite utilitzada per mesurar els rendiments de sistemes paral·lels. Aquestes proves estan derivades de la computació de dinàmica de fluids i consisteixen en diferents *kernels* i mides de problemes. La tria d'aquesta suite és per què ja s'havia treballat en altres assignatures dels estudis d'enginyeria, i se sabia que compliria amb els requisits d'execució al clúster. En el nostre cas l'elecció havia de passar per un *kernel* que utilitzés MPI per poder ser utilitzat al clúster MESOS i al clúster MPD. Aquest *kernel* per tant s'executarà als dos clústers per estudiar els resultats obtinguts. Del resultat de la secció anterior s'espera que els temps d'execució, per exemple el temps total d'execució, han de ser el mateix o pràcticament igual en els dos entorns.

Per ajudar-nos en l'estudi de l'execució de cada *kernel* s'utilitzarà una llibreria disponible a *mpicc* que és l'anomenada MPE[29], que permet inserir al codi del programa punts de mesura, per exemple a les crides MPI, de forma automàtica. La forma d'inserir aquests punts al *kernel* es realitza en la fase de compilació de forma que tornaran un valor al fer les execucions que es podrà estudiar posteriorment amb el programa JUMPSHOT, també inclòs a la instal·lació de MPICH2.

[cc] http://www.nas.nasa.gov/publications/npb.html

5.3.1 Instal·lació de mètriques NPB

El primer pas és descarregar les mètriques d'interès, que serien les que tinguin una construcció basada en MPI. A la pàgina web de la NPB, veiem que la última versió compleix amb aquests requisits:

Versió	Mètriques incloses	Classes (mida)	Models de programació
NPB 3.3	IS, EP, CG, MG, FT, BT, BT-IO, SP, LU, UA, DC, DT	S,W,A,B,C,D,E	MPI, OpenMP, serial

Taula 5. Continguts de la versió NPB 3.3.

Per descarregar-la s'han de seguir els passos que s'indiquen a les instruccions[dd] i seguir l'enllaç de descàrrega[ee] (exigeix emplenar un formulari amb dades personals de forma prèvia a la descàrrega).
Finalment s'obté el fitxer **NPB3.3.tar.gz** que conté totes les mètriques.

La instal·lació és molt senzilla:

1. Descomprimir el fitxer

   ```
   $ tar -xzvf NPB3.3.tar.gz
   ```

2. Modificar el fitxer make.def per indicar els valors de compilador i l'opció de llibreries MPE. S'ha d'observar que la única mètrica que és MPI de la suite NPB 3.3 és la IS, i per tant serà la que s'utilitzarà en aquest apartat per mesurar eficiència d'execució. La mètrica està escrita en C i serà en aquella part on s'ha d'indicar els valors de compilador que utilitzem, en aquest cas mpicc per la instal·lació de MPICH2, i també l'opció d'enllaç a llibreria MPE per obtenir les mesures de temps i execució del programa (-mpe=mpilog).
   El fitxer make.def no existeix, però es pot utilitzar de base el fitxer make.def.template:

   ```
   $ cd $HOME/NPB-3.3/NPB-3.3-MPI/config/
   $ cp make.def.template /make.def
   ```

   Modifiquen els valors d'interès de make.def, es marquen en negreta:

   ```
   #---------------------------------------------------------------------------
   # Parallel C:
   #
   # For IS, which is in C, the following must be defined:
   #
   # MPICC      - C compiler
   # CFLAGS     - C compilation arguments
   # CMPI_INC   - any -I arguments required for compiling MPI/C
   # CLINK      - C linker
   # CLINKFLAGS - C linker flags
   # CMPI_LIB   - any -L and -l arguments required for linking MPI/C
   #
   # compilations are done with $(MPICC) $(CMPI_INC) $(CFLAGS) or
   #                            $(MPICC) $(CFLAGS)
   # linking is done with       $(CLINK) $(CMPI_LIB) $(CLINKFLAGS)
   #---------------------------------------------------------------------------

   #---------------------------------------------------------------------------
   # This is the C compiler used for MPI programs
   #---------------------------------------------------------------------------
   MPICC = /home/master/mpich2-install/bin/mpicc
   ```

[dd] http://www.nas.nasa.gov/publications/sw_instructions.html
[ee] https://www.nas.nasa.gov/cgi-bin/software/start

```
# This links MPI C programs; usually the same as ${MPICC}
CLINK    = $(MPICC)

#----------------------------------------------------------------------------
# These macros are passed to the linker to help link with MPI correctly
#----------------------------------------------------------------------------
#CMPI_LIB  = -L/home/master/mpich2-install/lib -lmpi

CMPI_LIB  = -L/home/master/mpich2-install/lib -mpe=mpilog
#----------------------------------------------------------------------------
# These macros are passed to the compiler to help find 'mpi.h'
#----------------------------------------------------------------------------
#CMPI_INC = -I/home/master/mpich2-install/include

#----------------------------------------------------------------------------
# Global *compile time* flags for C programs
#----------------------------------------------------------------------------
CFLAGS = -O

#----------------------------------------------------------------------------
# Global *link time* flags. Flags for increasing maximum executable
# size usually go here.
#----------------------------------------------------------------------------
CLINKFLAGS = -O
```

3. Finalment ja es poden crear els binaris d'interès indicant els paràmetres de mètrica, processadors i mida del problema:

   make is NPROCS=(1,2,4, .., 2^n) CLASS=(S,W,A,B)

   On

   - **is**: és la mètrica utilitzada
   - **NPROCS**: el total de processadors en execució, de valor 2^n
   - **CLASS**: la mida del problema a tractar, que pot ser: S, W, A i B.

   El directori on s'executa és *$HOME/NPB-3.3/NPB-3.3-MPI/* i els binaris són guardats a *$HOME/NPB-3.3/NPB-3.3-MPI/bin/*

Unes dades a tenir en compte en la compilació dels binaris són les següents:

- S'ha de tenir la precaució de fer *make clean* abans del *make is NPROCS...*, per què no s'agafin els valors de la última compilació.

- El número de processadors obligatòriament ha de ser de base dos, per tant no hi ha una llibertat absoluta en triar el valor, i per al nostre cas la penalització és important ja que disposarem de tres processadors com a màxim i és un valor que no es podrà triar.

- De les classes disponibles es triarà la de major mida (B) per evitar execucions massa ràpides que impedeixen tenir un valor de temps d'execució fàcil per a la comparació.

Així es faran les següents compilacions i s'obtindran els següents binaris:

$ make is NPROCS=1 CLASS=B

Binari que s'obté: **is.B.1**

$ make is NPROCS=2 CLASS=B

Binari que s'obté: **is.B.2**

5.3.2 Característiques de l'aplicació NPB - IS

Entre tots els *benchmarks* s'ha escollit l'estudi del *kernel* IS que fa una ordenació d'enters utilitzant un algorisme paral·lel. El funcionament d'aquest problema és el següent:

- Els enters, *keys*, són generats per un algorisme pseudoaleatori a partir d'una llavor (314159265) de forma seqüencial i són distribuïts de forma uniforme a la memòria. Cada clau és mapada en un paraula de memòria (no inferior a 32 bits), de la que no es mourà excepte si és requerit per procés d'ordenació. El traspàs a memòria es contempla:
    - Memòria global compartida (les claus estan a adreces contigües)
    - Memòria distribuïda (diferents unitats que emmagatzemen un cert valor de claus)
    - Jerarquia de memòria: es contempla un espai comú inicial on estaran emmagatzemades totes les claus. Aquest espai pot ser qualsevol que pugui emmagatzemar el total de claus.
- La ordenació es realitza de forma ascendent, o com diu el text explicatiu "en forma no descendent", mitjançant un procés de permuta de posicions dels enters fins que s'obté la seqüencia ordenada, que és aquella que la clau inicial té valor inferior a les següent, i així fins el final de la seqüència.
- En aquest problema no és necessari una ordenació estable, entesa aquesta com que l'ordenació ha de respectar les posicions inicials del elements. Es a dir davant de dos elements de valor igual si K_1 era anterior a K_2 en l'aparició seqüencial prèvia a l'ordenació, una vegada la seqüència s'ha ordenat K_1 ha de precedir a K_2.
- Es produeixen dos verificacions de l'ordenació, que tornen un valor cert si el valor és l'esperat:
    - Test parcial. Per cada tram ordenat es comprova que les claus estan ordenades fent la comparació amb uns valors de referència (el problema inclou una taula amb aquests valors).
    - Test total. Es comprova l'ordenació de totes les claus estan ordenades de forma ascendent.
- El procés és cronometrat en els passos d'ordenació de cada clau, càlcul de seu rang i comprovació amb test parcial. El temps per tant no inclou la generació i càrrega de valors a memòria ni el procés de verificació total.

Els valors que s'utilitzen en els problemes són els següents:

	N	B_{max}	*seed*	I_{max}
Classe S	2^{16}	2^{11}	314159265	10
Classe W	2^{20}	2^{16}	314159265	10
Classe A	2^{23}	2^{19}	314159265	10
Classe B	2^{25}	2^{21}	314159265	10

Taula 6. Taula resum de classes per al problema IS.

On:

- N: número de claus (enters) a ordenar
- B_{max}: valor màxim de clau (enter)
- *seed*: llavor d'inici del procés pseudoaleatori de generació de claus
- I_{max}: iteracions màximes del procés d'ordenació

Respecte a les Classes la referència que indica NPB és la següent:

- Classe S: petita per a test ràpids
- Class W: mida adequada per una estació de treball (anys 90)
- Classes A, B, C: problemes de test estàndard (la mida del problema augmenta en un factor ~4X al passar a una classe immediatament superior).

Una comparació de la mida del problema (claus a ordenar) entre les classes:

	is.S	is.W	is.A	is.B
is.S	1	16	128	512
is.W	0,0625	1	8	32
is.A	0,0078125	0,125	1	4
is.B	0,001953125	0,03125	0,25	1

Taula 7. Taula de diferències de mida per als problemes IS.

5.3.3 Escenaris de proves

Els escenaris de proves són diferents, segons si s'executa el clúster MPD o el clúster MESOS:

En el cas d'execució a clúster MPD:

1. S'ha d'iniciar l'anell MPD que correspongui al total de processadors necessaris, i verificar la seva posada en marxa (més detalls a la secció 4.4):

   ```
   $ mpdboot –n <processadors> -f mpd.hosts
   $ mpdtrace -l
   ```

2. La crida d'execució mínima és la següent, per exemple per les crides bloquejants (l'anell tindrà tants MPD com processadors són necessaris):

   ```
   $ mpiexec –n <processadors> ./<binari IS>
   ```

   Per exemple:

   ```
   $ mpiexec –n 1 ./is.B.1
   ```

   Els processadors seran com a màxim 3, però en execució MPD si és pot fer la tercera crida, tot i no tenir 4 processadors, ja que al finalitzar les assignacions als tres processadors físics la següent assignació es farà al primer de forma seqüencial (per exemple: màster, esclau1, escalu3, màster , ...).

   L'execució generarà un fitxer compatible amb el programa JUMPSHOT, per permetre el traçat d'un gràfic de processos en entorn X11:

   - **fitxer is.B.x.clog2**, que conté les dades numèriques que traspassa l'execució d'MPI amb la llibreria MPE. Aquest fitxer s'haurà de convertir a .slog2 al mateix programa JUMPSHOT.

En el cas d'execució a clúster MESOS:

1. Primer s'han d'iniciar el màster i tots els esclaus del clúster MESOS i verificar la seva posada en marxa a través de l'administrador WEBUI (més detalls a la secció 4.3).

2. La crida d'execució és la següent:

   $ mpiexec-mesos –n <processadors> -m 1512 10.10.78.85:5050 \ ./<binari IS>

   Per exemple:

   $ mpiexec-mesos –n 1 -m 1512 10.10.78.85:5050 ./is.B.1

   Que també tornarà un fitxer .clog2 per treballar amb JUMPSHOT.

Per defecte l'execució sempre es farà amb el màxim de recursos del clúster[28] per facilitar la comparació de resultats:

- **Processadors**: només dos unitats, tot i que segons el gestor MESOS hi han disponibles 6 processadors. Aquest valor de processadors que indica MESOS s'obté a causa de combinar màquines de característiques diferents com són el màster, en configuració processador *multicore* amb quatre nuclis, i les màquines esclaves que només disposen d'un processador d'un nucli. D'aquesta suma s'obtenen els sis processadors que indica el WEBUI. El limitant de només dos processadors està directament relacionat amb que la compilació obliga a que el seu número sigui en base dos, per tant no podrem triar un valor més gran.

- **Memòria**: s'ha triat el valor de 1512 MB per computador, per ser el valor lliure que ha permès fer les proves de clúster MESOS sense problemes, deixant suficient memòria per a l'execució dels serveis dels sistemes operatius. Aquesta limitació només s'aplica al clúster MESOS, el clúster MPD opera sense que s'especifiqui quina quantitat de memòria podrà utilitzar, per tant és possible que disposi d'una mida més gran, però en tot cas un valor d'una quarta part lliure (75% d'ocupació) sembla un valor realista en execució. Aquesta dada s'ha de tenir en compte a l'analitzar els rendiments si les diferències són molt evidents.

5.3.4 Resultats obtinguts d'execució d'aplicació MPI

Finalment es presenten el resultat de les execucions segons indica el programa JUMPSHOT per als quatre fitxer .clog2 que s'han generat, tant per al clúster MESOS com MPD. Aquest fitxers s'ha de convertir a format .slog2 per a ser tractats al programa JUMPSHOT, aquesta conversió es pot realitzar amb una utilitat del mateix programa:

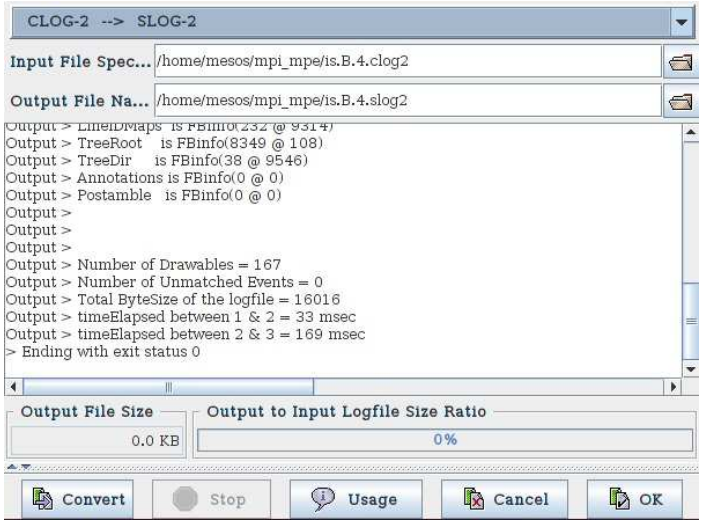


Figura 53. Conversió de format de fitxer .clog2 a .slog2 al programa JUMPSHOT.

Respecte al clúster MESOS s'indica també la informació que mostra WEBUI respecte a les execucions. Finalment es mostrarà una taula resum amb els valors de temps total i MOPs que ha registrat la pròpia aplicació per facilitar la comparació de resultats[ff] i extreure conclusions.

Aplicació executada: Binari is.B.1

L'execució al clúster MESOS per a un processador torna un temps final de càlcul de **13,95 segons**:

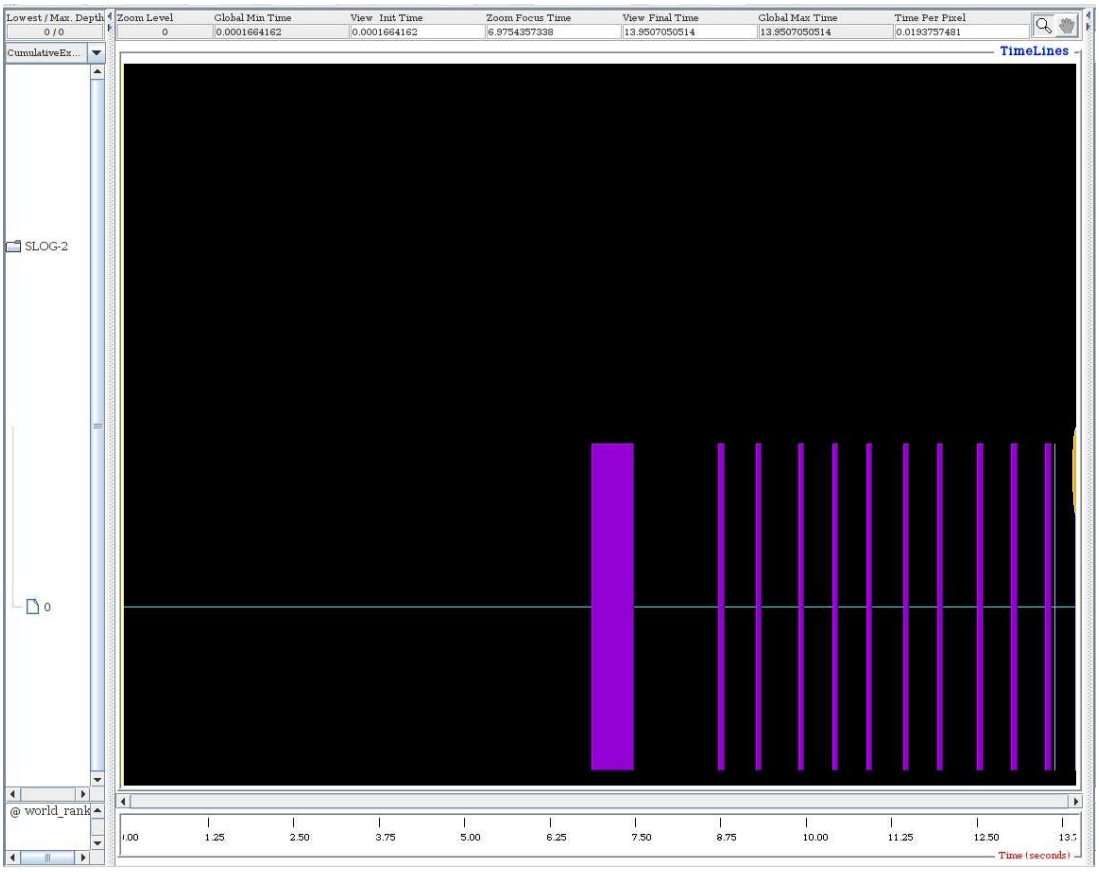


Figura 54. Execució del programa is.B.1, amb MPE, al clúster MESOS i mostrada per JUMPSHOT.

[ff] Es pot consultar el detall de cada execució a l'Annex.

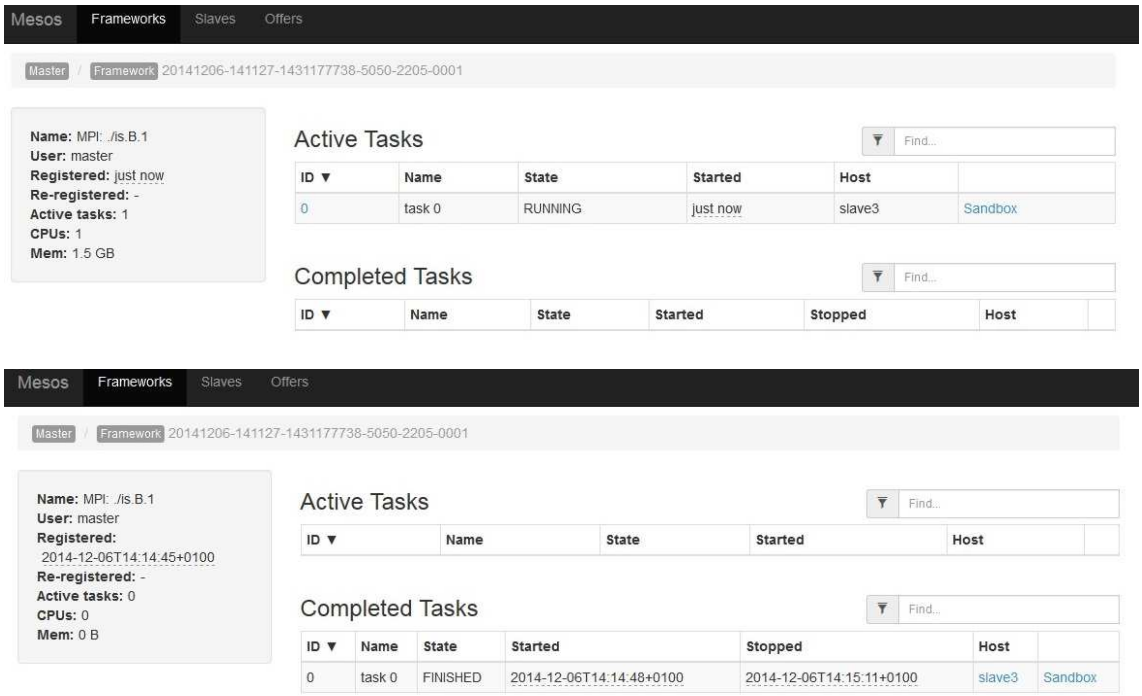


Figura 55. Execució del programa is.B.1 al clúster MESOS. WebUI mostra l'activació i finalització al node esclau 3.

L'execució al clúster MPD per a un processador, torna un temps final de càlcul de **12,33 segons**:

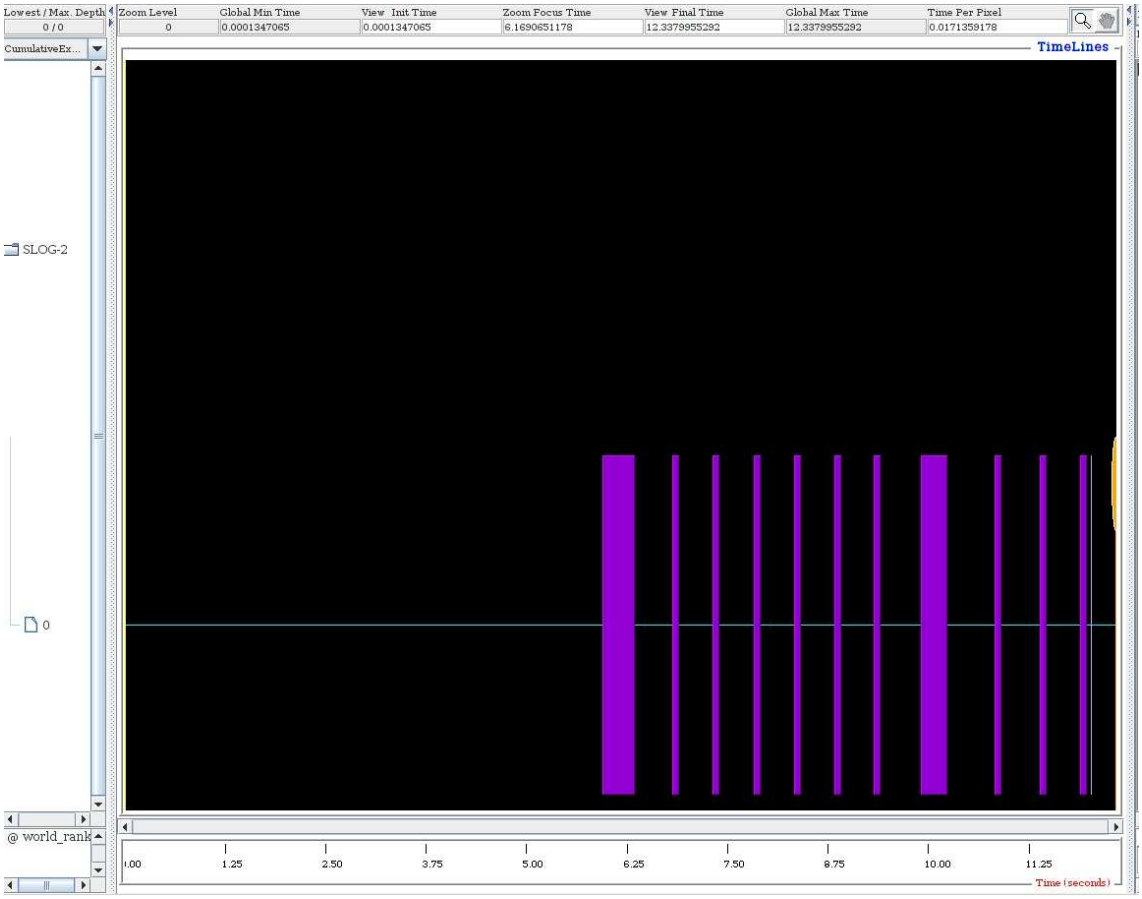


Figura 56. Execució del programa is.B.1, amb MPE, al clúster MPD i mostrada per JUMPSHOT.

Aplicació executada: Binari is.B.2

L'execució al clúster MESOS per a dos processadors, torna un temps final de càlcul de **56,58 segons**:

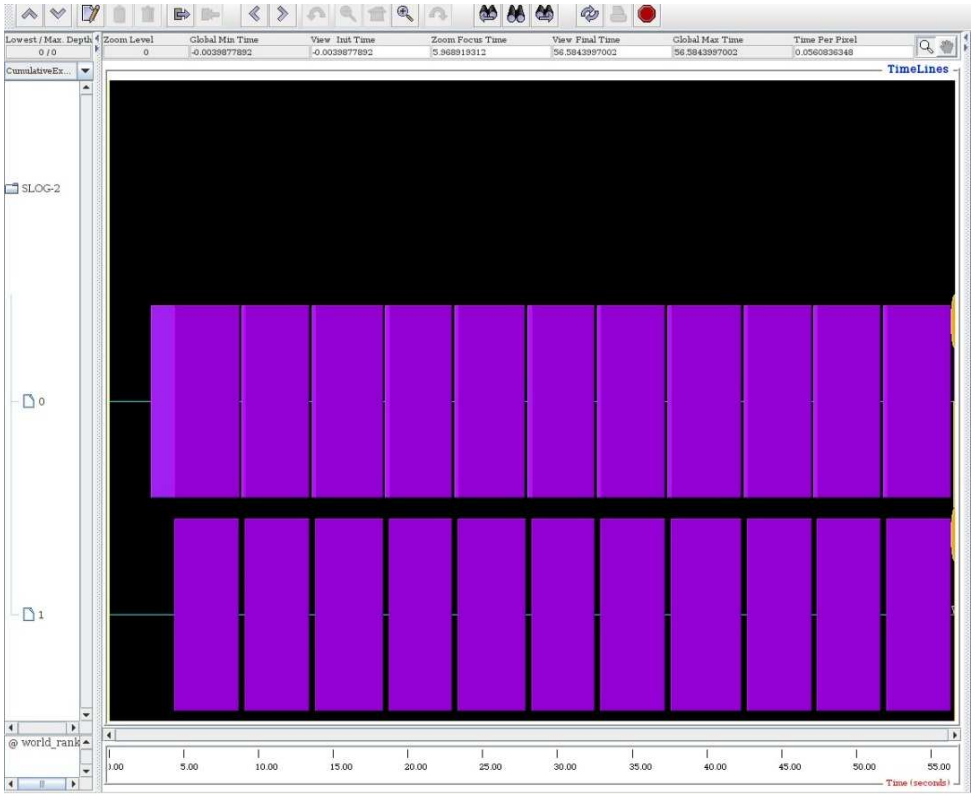


Figura 57. Execució del programa is.B.2, amb MPE, al clúster MESOS i mostrada per JUMPSHOT.

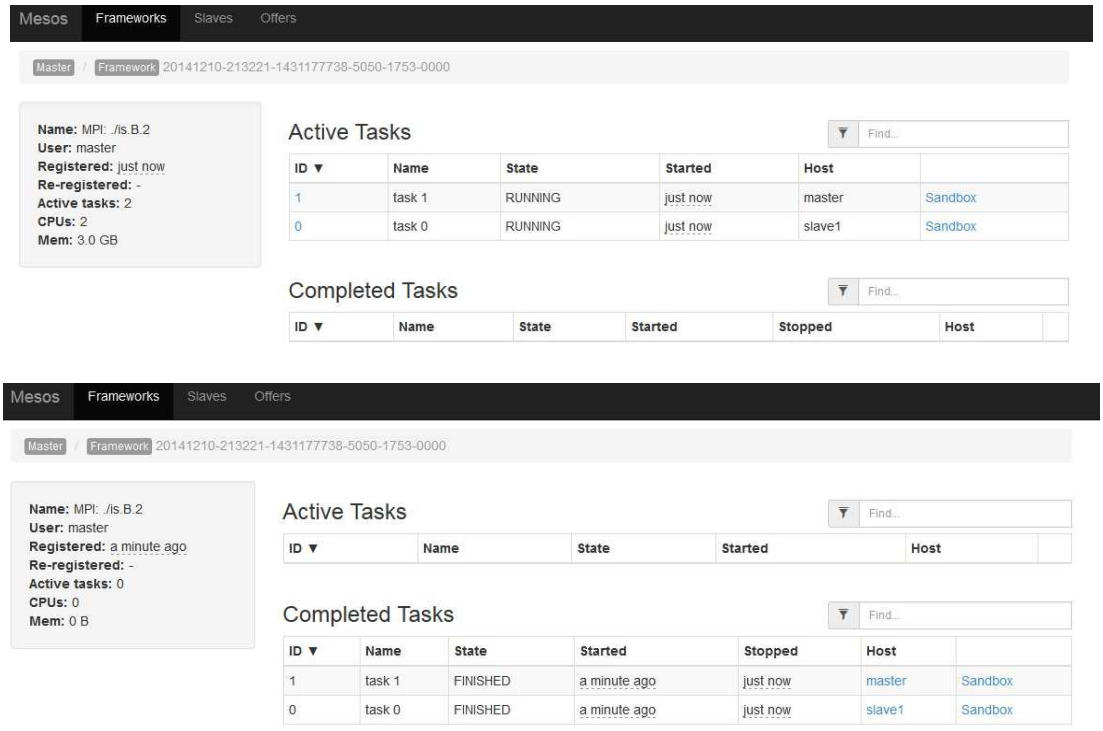


Figura 58. Execució del programa is.B.1 al clúster MESOS. WebUI mostra l'activació i finalització al node màster i esclau 1.

L'execució al clúster MPD per a dos processadors, torna un temps final de càlcul de **57,89 segons**:

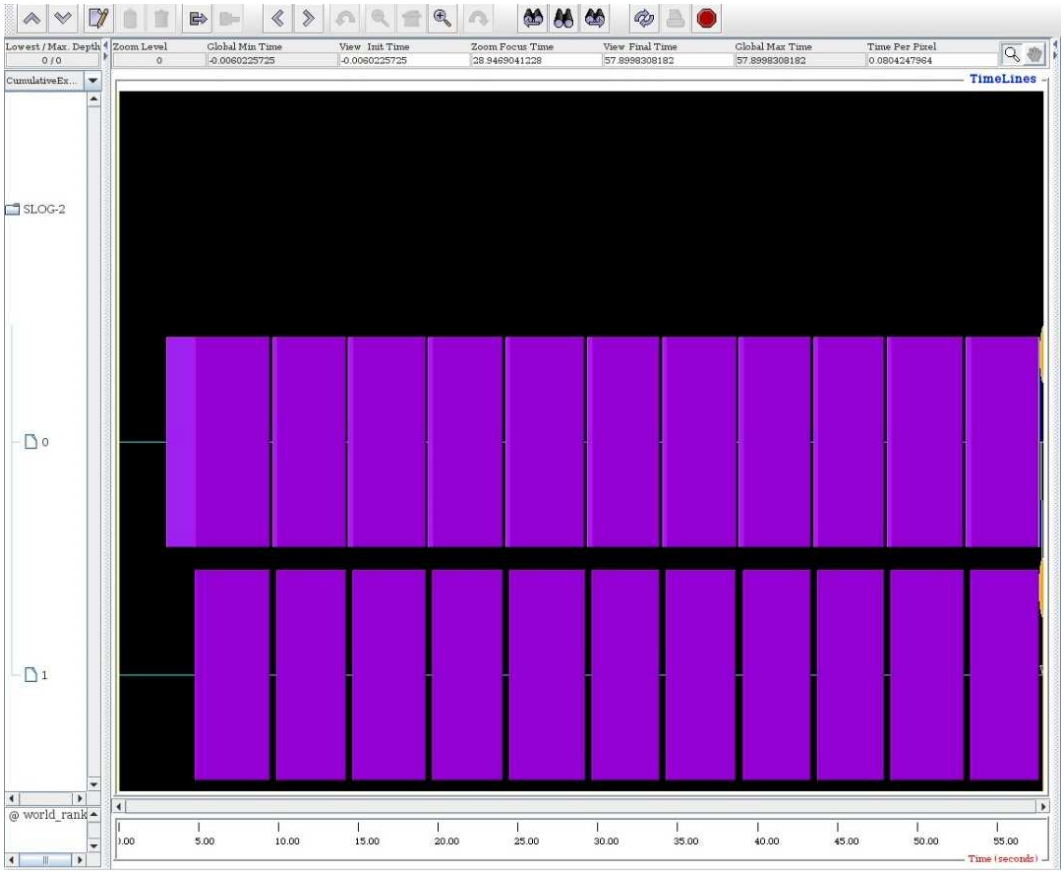


Figura 59. Execució del programa is.B.2, amb MPE, al clúster MPD i mostrada per JUMPSHOT.

Taula resum

Les execucions anteriors mostrades a JUMPSHOT, respecte a la llibreria MPE, més les que registra la pròpia aplicació, poden ser consultades a l'annex. Es resumeixen a la taula 8 per facilitar les conclusions:

	Clúster MESOS		Clúster MPD	
	1 processador	2 processadors	1 processador	2 processadors
Binari	is.B.1	is.B.2	is.B.1	is.B.2
MOPS[gg] IS	54,95	7,05	59,66	6,97
Temps IS (S)	6,11	47,60	5,62	48,15
Temps MPE (S)	13,95	56,58	12,33	57,89

Taula 8. Resum dels valors d'execució d'aplicació MPI als clústers MESOS i MPD.

5.3.5 Conclusions de l'execució d'aplicació MPI

La conclusió és l'esperada en el sentit que les diferències en les execucions són petites. Així podem confirmar la conclusió que ja s'avançava a la secció anterior: l'execució sota el gestor MESOS no penalitza el rendiment del clúster MPI.

[gg] Milion Operation per Second, Milions d'operacions per segon.

5.4 Conclusions de les execucions en clúster MESOS MPI

Les dos seccions anteriors han servit per mesurar el rendiment del clúster en execució paral·lela MPI amb i sense la gestió de MESOS. La idea era saber si la capa MESOS produïa algun efecte, i si aquest millorava o empitjorava els resultats. Els resultats indiquen que no es produeix cap limitació, i tampoc cap avantatge en rendiment que sigui remarcable. Aquesta dada és important ja que els recursos del clúster realment són molt petits, tant en memòria com en processador, i per tant els recursos operatius de gestió de MESOS no són gaire exigents, i per tant la capa MESOS no limita l'eficiència del clúster on està instal·lada.

El segon punt a destacar és que comprovat que MESOS no aporta una diferència de rendiment per la gestió dinàmica de recursos, el que es pot pensar és si realment és necessària la seva participació en un clúster paral·lel MPI, que ja té resolta la seva execució amb dimonis MPD. La resposta és sí, i pot ser una molt bona elecció en clústers reals. El raonament ha de partir d'analitzar primer el clúster de prova utilitzat. La principal limitació són els baixos recursos que disposa, que no han permès llançar múltiples execucions en paral·lel: dos màquines per als esclaus amb 2GB de memòria i un processador no *multicore*, i una màquina virtual amb 2 GB de memòria i *multicore*. Si l'escenari fos un clúster real, per exemple cent màquines amb 8 GB de memòria i processador *multicore*, en clúster CLUMPS[hh], l'execució sobre MESOS faria:

- Que fos possible llançar varies tasques MPI de forma simultània al clúster, per exemple una tasca per a 20 hosts, una segona per 10 hosts, etcètera. A cada llançament MESOS assignaria recursos disponibles a la nova tasca.

- Controlar l'execució en temps real, tant de les tasques com dels recursos utilitzats.

- El clúster no té per què estar limitat a execucions MPI, poden conviure més d'un entorn d'aplicació de forma natural, per exemple CRAY i CASSANDRA, compartint tots els recursos del clúster de forma dinàmica assignant els que estan lliures a les noves execucions, sigui quin sigui l'entorn d'aplicació.

- No cal reconfigurar el clúster una vegada s'han instal·lat els entorns d'aplicació necessaris, aquests estaran a l'espera de ser cridats per executar el treball que se li assigni en qualsevol moment i gràcies a que disposaran de recursos de forma dinàmica.

- El clúster amb MESOS pot ser molt flexible en la gestió de recursos, planificable i d'alta disponibilitat tant d'esclaus com de màsters. Només cal afegir els recursos necessaris.

En el cas del clúster MPD sí podem llançar tasques en paral·lel, però no té tota la potència i flexibilitat de MESOS per assignar recursos a d'altres models o entorns d'aplicació de forma dinàmica.

En conclusió podem dir que el gestor MESOS no penalitza les execucions, i en canvi permet la convivència pacífica de múltiples entorns d'aplicació que no afecten el rendiment, i que esperen de ser cridats en el moment que són necessaris. A més s'eliminen tasques de reconfiguració del clúster i per tant ofereix una alta disponibilitat i flexibilitat d'execució.

[hh] CLUMPS: Clúster de multiprocessadors amb memòria híbrida, que tenen pas de missatges entre nodes i memòria compartida a dins de cada node.

6. Valoració econòmica del projecte

Els projectes necessiten d'una valoració d'activitats i costos, per poder planificar el seu impacte de recursos. Aquest apartat mostra el possible cost de recursos en hores de treball, i recursos materials, que s'han necessitat per finalitzar-lo. Partirem de la projecció inicial i mostrarem la realitat final per veure la possible desviació i si ha estat motivada.

6.1 Cost de desenvolupament del sistema

La plataforma MESOSPHERE és totalment gratuïta, així com el sistema operatiu *Ubuntu Server 12.04* sobre la que s'implementa. També s'utilitzarà virtualització sobre l'aplicació VMware, en la seva versió d'ús no comercial i per tant sense cost.

Així el preu final d'implementar el clúster és zero, fora dels costos elèctrics de funcionament.
A nivell d'hores s'ha volgut establir la següent relació dia/hora:

- El projecte s'inicia el disset de setembre i finalitza el trenta de desembre.
- Tots els dies són laborables per realitzar el projecte
- Els dies feiners s'espera dedicar una mitjana de tres hores/dia
- Els dies festius s'espera dedicar una mitjana de sis hores/dia

El total de dies feiners és de setanta-quatre, el total d'hores és de dues-centes vint-i-dos (222 h).
El total de dies festius és de trenta-un, el total d'hores és de cent vuitanta-sis hores (186 h).

Així el projecte s'havia d'encaixar en un total de **quatre-centes vuit hores (408 h)**. Aquesta planificació obeeix a l'estat inicial del projecte, quan s'havia de fer una projecció del seu cost, però l**a duració real és la següent:**

- El total de dies feiners s'ha reduït a cinquanta-quatre, dels quals set dies no s'han pogut dedicar a desenvolupar el projecte. La duració real mitjana ha estat de tres hores/dia. **El total és de cent quaranta-una hores (141 h)**

- El total de dies festius s'han ampliat a cinquanta-un, aprofitant dos setmanes de vacances, dels quals cinc dies no s'han dedicat a desenvolupar el projecte. La duració real mitjana ha estat de vuit hores/dia. **El total és de tres-centes seixanta-vuit hores (368 h)**

En total s'han necessitat **cinc-centes nou hores (509 h)**, un vint-i-cinc per cent (25%) més del planificat inicialment. El motiu d'aquest desfasament important el podem justificar en el nul coneixement sobre la tecnologia estudiada, així ha estat necessari un esforç important d'adquisició de coneixements, més gran de l'esperat inicialment. També molts problemes trobats, pels pocs recursos materials disponibles, han forçat la recerca de solucions indirectes, i que han fet augmentar el temps de desenvolupament.

6.2 Cost material

El projecte està basat en la construcció d'un clúster de forma que es pugui estudiar les capacitats de la plataforma MESOS. Aquest clúster és pot construir de dos formes al meu abast:

- Utilitzant nodes llogats a partir d'una plataforma de computació com MESOSPHERE o Amazon EC2[ii].

[ii] http://aws.amazon.com/es/ec2/. *Amazon Elastic Compute Cloud (Amazon EC2)* és un servei web que proporciona capacitat informàtica amb mida modificable en el núvol. Està disseny per facilitar als desenvolupadors recursos informàtics escalables basats en web.

- Implementar un clúster petit de forma física, combinant màquines físiques i virtuals. Per a aquest supòsit disposaré d'uns PC estàndard, i una mica antics (2005), tres unitats, que m'han cedit. A més s'instal·larà una màquina virtual al meu PC, també una mica antic (2008), en principi el màster del clúster.

En aquest projecte només s'ha implementat la segona opció, sense cost econòmic, i en tot cas hi ha una secció dedicada a com s'implementaria un escenari similar en plataformes de computació com l'enunciada, però serà de forma teòrica no es un objectiu del projecte fer la implementació real d'aquesta forma.

El maquinari sobre el que s'ha instal·lat el clúster és el següent:

- Un PC Intel (R) core (TM)2 Quad CPU Q6600@2.4 GHz, 3.25 GB DRAM.
  Programari instal·lat:
  - Sistema operatiu Microsoft Windows 7, service pack 1, amb les següents aplicacions instal·lades al principi del projecte amb relació directa:
    - Paquet ofimàtic Microsoft Office 2013
    - Microsoft Project 2013
    - Microsoft Visio 2013
    - Màquina virtual VMWare© Player, versión 6.0.3
      Programari instal·lat:
      - Sistema operatiu Ubuntu server versió 14.04
        Programari instal·lat:
        - JAVA versió 1.6
        - MESOS 0.20.0
        - MPICH2
        - NAS NPB
        - PERFTEST-1.5
      - Sistema operatiu Ubuntu Desktop versió 14.04
        Programari instal·lat:
        - JAVA versió 1.6
        - MPICH2
        - Gnuplot

- Dos PC AMD 64 3200+ 2 GHz, 2 GB DRAM.
  Programari instal·lat:
  - Sistema operatiu Ubuntu server versió 14.04
    Programari instal·lat:
    - JAVA versió 1.7
    - MESOS 0.20.0
    - MPICH2

- Un commutador (*switch*) de 8 ports a 10/100 Mbps marca SMC, model SMCFS8, sobre el que s'instal·larà la xarxa local, a 100 Mbps.

Tots els materials emprats, sigui maquinari o programari, no han tingut cap cost econòmic.

7. Conclusions del projecte

Aquest apartat és el de tancament del projecte. En ell es mostraran els resultats finals, l'anàlisi de tot el treball realitzat, els problemes trobats i les seves solucions. L'objectiu és mostrar tot l'esforç i els coneixements, no tècnics necessàriament, apresos al llarg del projecte.

7.1 Llista d'objectius inicials i estat final assolit

A la secció 1.3 es va fixar la següent llista d'objectius:

- *Conèixer les bases del disseny de clústers de computació i la problemàtica de la compartició de recursos en la millora d'eficiència.*

- *Estudi del gestor de clústers MESOS, en les següents funcions:*
    - *Creació d'un clúster totalment operatiu basat en màquines físiques i virtuals, incloent la gestió d'instal·lació de programari i configuració de xarxa.*

    - *Administració del clúster amb les funcions que incorpora la plataforma MESOS.*

    - *Execució activa del clúster per demostrar les seves funcionalitats de treball distribuït, utilitzant alguns dels entorns desenvolupats en aquest moment.*

    - *Projecció en el control de milers de nodes per a treball distribuït.*

- *Extracció de conclusions i comparació amb altres gestors actuals*

En aquest punt del projecte podem copsar el compliment dels objectius fixats. A l'inici del projecte el coneixement sobre el disseny de clústers de computació i les problemàtiques associades era molt bàsic, el treball desenvolupat ha servit per mostrar les complexitats del funcionament i administració d'aquests sistemes, les diferències entre clústers monolítics i dedicats en exclusiva a una aplicació en execució, contra els clústers multi aplicació i de ràpid creixement a causa de la demanda de servei causada pels usuaris a Internet. S'ha passat doncs de clústers pensats per resoldre execucions en entorns molt específics, investigació o empresarials, per definició tancats al públic en general, a la necessitat de clústers que han de suportar grans volums d'usuaris, resilients i d'alta disponibilitat. Aquest canvi ha fet necessari pensar en gestors de recursos i la distribució dels treballs pensant en l'alta eficiència. Aquest coneixement s'ha obtingut amb la realització del projecte.

L'estudi del gestor MESOS s'iniciava en un desconeixement previ absolut, però una vegada superada la fase de coneixement i implantació, s'ha pogut demostrar el funcionament operatiu al 100% d'un petit clúster, de recursos molt limitats i per això encara és més rellevant l'eficiència del gestor. Aquest clúster de tres màquines físiques ha pogut ser administrat de forma bàsica, permetent seleccionar configuracions dels seus nodes i l'execució d'una aplicació, a partir de configurar un entorn d'aplicació, que permet l'execució en paral·lel de programes que utilitzen la llibreria MPI. Una vegada tot ha estat configurat el funcionament ha estat constant, sense errors, i per aquest motiu també podem parlar que MESOS és, a més d'eficient, és eficaç.

Al llarg del desenvolupament del treball també s'ha pogut extrapolar com es podria executar aquest gestor en un entorn de milers d'usuaris, basat en la distribució de múltiples nodes esclaus, múltiples màsters i la creació d'alta disponibilitat a partir d'altres mòduls d'administració com *Zookeeper* o les configuracions dels esclaus que permeten reviure en cas de fallada, també scripts d'administració d'inici d'execució de tots els components anteriors. Una part que també s'ha presentat és el lloguer per ús que ofereixen diverses companyies d'aquesta tecnologia com a *IaaS*, molt interessant si és vol fer estudis previs de components en un clúster model abans d'executar el servei per als usuaris.

S'han presentat també una comparació entre plataformes, sobre tot dos actuals que estan ara mateix competint com són l'estudiada i OMEGA, que és un projecte de Google.

Així els objectius inicials del projecte s'han pogut complir, tot i que la profunditat assolida en alguns dels punts ha estat iniciaria, limitada pel temps i els recursos necessaris.

7.2 Problemes trobats en el desenvolupament

Els treballs prevists al llarg del projecte han presentat diverses dificultats, la majoria causades pel meu desconeixement previ en certes matèries, que ha provocat no pocs errors i dificultats i pèrdues de temps associades. Aquestes dificultats han servit també de nova font de coneixement, ja que era necessari superar-les per finalitzar totes les tasques proposades, i això ha obligat a revisar moltes fonts d'informació, de diversos àmbits, i posar en pràctica estratègies per solucionar-les. A continuació es mostren de forma detallada els problemes i les solucions aplicades, si aquestes s'han pogut introduir.

7.2.1 Clúster físic

La idea principal del projecte és muntar un clúster de manera no totalment virtualitzada, apropant així l'execució a un model més real de màquines físiques. Gràcies a una aportació sense cost he pogut disposar de sis computadors de forma gratuïta, si bé aquests són de baix perfil respecte als models actuals, tant a nivell de processador com de memòria., i algun d'ells estava fora de servei però aprofitable com a recanvi en alguns components. Aquesta part també es pot referir com una aproximació als entorns reals de clústers actuals, on els computadors utilitzats no tenen per què ser els de major rendiment i específicament dissenyats a la tasca a resoldre. De fet la tendència és la d'utilitzar processadors de baix preu amb una bona relació preu/rendiment/cost energètic, com per exemple els utilitzats en terminals mòbils (ARM), enlloc de dedicar grans esforços a processadors especialment dissenyats[3031]. El clúster que s'ha muntat en aquest projecte no s'aproxima de forma real a aquesta tendència que sembla serà la triada en futurs supercomputadors, tampoc ho pretenia, només és una demostració que amb computadors domèstics és possible muntar un entorn que permet computació paral·lela i que està gestionat per un gestor dinàmic dels seus recursos, en aquest cas MESOS, que a més està present en clúster extremadament més grans i potents. Així és més una prova de concepte i demostració, i aplicable de forma directa a petites o mitjanes empreses, on amb poca inversió és pot fer el mateix, això si amb un maquinari no tan just com l'utilitzat en aquest projecte.

El problemes trobats en la implementació del clúster físic han estat poder obtenir el màxim de recursos de les sis màquines rebudes. Així el valor de processador, disc dur, placa base i interfície de xarxa eren impossibles de millorar, tots eren iguals, només podia tractar d'ampliar la memòria DRAM dels sistemes. Aquesta decisió va fer que de sis màquines la solució quedés reduïda a dos màquines, ja que la quantitat de memòria màxima disponible són vuit DIMM d'1 1 GB de memòria DRAM (DDR400 nonECC DIMM), i que per problemes amb la placa base A8V-VM CSM[32] de la marca ASUS ha estat impossible que siguin reconegudes en tota la seva extensió. En un principi el clúster s'havia dissenyat pensant en tres nodes esclaus i un node màster, en aquest cas virtualitzat, però els nodes esclaus només disposaven de 1,5 GB i amb aquest valor va ser impossible instal·lar MESOS per què es produïa un error per exhaunció de la memòria. Així la memòria DRAM de cada computador és de només 2 GB, i els nodes esclaus han passat de tres a dos de forma real, i el tercer s'ha virtualitzat al node que només havia de fer de màster. Les proves que s'han realitzat per augmentar aquesta capacitat, que han resultat inútils, han estat les següents:

- Actualitzar la BIOS a la última versió.
- Provar canvis de configuració a la BIOS respecte a la configuració de memòria (ECC)
- Provar de canviar les DIMM de sòcol, intercanviant-les., etcètera
- Instal·lar només un DIMM, i provar-los tots per separat.
- Utilitzar només un dels canals de memòria (Channel 0 /1).
- Instal·lar DIMMs de menor capacitat i configuració (256 MB i 512 MB).

En tots els casos el valor reconegut en la BIOS ha estat sempre una mica inferior a la meitat del valor de la memòria real instal·lada. Les cerques per Internet tampoc han aportat cap solució diferent a les provades. El fabricant de la placa base (ASUS), i el fabricant de les memòries (KINGSTON) sí indiquen que els components són compatibles, però no he trobat com solucionar aquesta incompatibilitat real que faria el clúster una mica més potent tant en memòria com en nombre de nodes.

7.2.2 Instal·lació i execució MESOS

La instal·lació de MESOS s'esperava resoldre de forma ràpida, però van ser necessaris pràcticament deu dies per tenir una versió de clúster que s'iniciés i s'aturés sense problemes. Una descripció detallada de les diverses dificultats trobades:

- La primera instal·lació es va realitzar sobre el sistema operatiu Ubuntu 12.04.5 Server segons s'indica a la guia d'instal·lació, però l'execució de l'actualització mínima prèvia a instal·lar MESOS, com instal·lar el paquet *build-essentials*, donava problemes d'enllaç amb la font (*broken dependencies*), també la resta d'instal·lacions de paquets (Java, Python i Maven). La revisió del fitxer de fonts */etc/apt/source.list* donava un resultat igual al obtingut des del repositori UBUNTU. Fins i tot el consultor va construir una versió KUBUNTU per poder provar-ho. Una altra prova va ser configurar el CD d'instal·lació com a repositori, fent la línia corresponent a */etc/apt/source.list*, i sí va funcionar però era insuficient per a la resta de paquets. La solució final ha passat per substituir la versió UBUNTU 12.04.5 Server per la versió UBUNTU 14.04 Server i desconec per què no podia arribar als repositoris UBUNTU per descarregar-los, ja que el *ping*[33] directe al repositori des del nodes arribava sense cap problema.

- Després van començar els problemes per la configuració física del clúster en la instal·lació de MESOS (versió 0.20.0), que per problemes de mida de memòria impedien la instal·lació als nodes. Aquest problema s'agreujava per la pròpia duració del procés, de vàries hores, cada cop que es volia provar alguna alternativa d'instal·lació. Al node màster es va detectar el problema a partir d'investigar un missatge d'error de procés per exhaunció de memòria, que una de les proves va indicar. Així es va ampliar la memòria del node màster a 2 GB, i el procés d'instal·lació de MESOS va finalitzar de forma positiva. El node màster disposava de 2 GB i els nodes esclaus només 1,5 GB, per tant va ser necessari desmuntar un dels nodes esclaus del clúster per aprofitar els seus recursos de memòria, reduint la mida de tres nodes esclaus a dos. Una vegada aconseguida la fita de 2 GB als nodes esclaus es va poder fer la instal·lació també sense problemes. Aquest error va arribar a fer-nos plantejar l'abandonament de la idea de crear un clúster real amb els meus recursos, i vam començar a proposar al laboratori de la UOC utilitzar màquines si fos possible. La solució del problema de memòria als nodes va finalitzar aquesta via.

- La informació de la guia d'instal·lació està referida a un escenari d'un node màster i un node esclau en execució a la mateixa màquina, així les comunicacions dels processos es produeixen pel bucle local 127.0.0.1. A més en l'escenari muntat la màquina està virtualitzada, de forma que utilitza la interfície de xarxa de la màquina real per comunicar-se amb la resta de nodes. A l'utilitzar el bucle local de la màquina virtual no podia accedir a la interfície gràfica del clúster WEBUI, registrada a la IP 127.0.0.1:5050, des de la màquina real que conté aquesta màquina virtual ja que s'apuntava al seu propi bucle local. La forma de solucionar-ho ha estat modificar la IP on es registra el servidor web i els serveis MESOS:

  ```
  $ ./bin/mesos-master.sh --ip=127.0.0.1 --work_dir=/var/lib/mesos
  ```

  Per la versió:

  ```
  $ ./bin/mesos-master.sh --ip=10.10.78.85 --work_dir=/var/lib/mesos
  ```

- La informació de la guia indica que per connectar-nos des del node esclau al node màster s'ha d'indicar de la següent forma:

```
$ ./bin/mesos-slave.sh  --master=10.10.78.85:5050
```

Però aquesta solució només serveix per connectar un esclau, ja que no es participa al node màster de la informació IP de l'esclau i només accepta un. Aquesta forma de funcionar es va detectar quan s'engegava el clúster amb el node esclau local (esclau 2), i es volia afegir un node esclau extern, per exemple l'esclau 1 o 3. El node extern no es podia registrar al clúster, i només ho podia fer quan el node esclau local era desconnectat, llavors sí que un nou node del pool es odia registrar. El registre és limitava a només un node i per tant no es permetia la concurrència que és l'objectiu del clúster. Consultant als fòrum es va trobar un problema similar[34], i la solució passa per especificar a cada node esclau la seva IP en la crida de registre al clúster:

Esclau 1

```
$ ./bin/mesos-slave.sh --ip=10.10.78.86 --master=10.10.78.85:5050
```

Esclau 2

```
$ ./bin/mesos-slave.sh --ip=10.10.78.85 --master=10.10.78.85:5050
```

Esclau 3

```
$ ./bin/mesos-slave.sh --ip=10.10.78.87 --master=10.10.78.85:5050
```

- Una vegada aconseguit un funcionament estable del gestor MESOS i els nodes esclaus, es va iniciar el procés d'instal·lar les llibreries MPI per permetre computació paral·lela en el clúster. La informació d'instal·lació indicada per la documentació disponible a MESOS respecte a clústers MPI, bàsicament indicava que primer s'havia d'instal·lar l'entorn MPI, en el cas d'aquest projecte MPICH2 d l'ANL, d'una forma estàndard. Els problemes d'aquesta instal·lació de llibreries estan recollits en el següent punt, però a nivell de MESOS van aparèixer alguns problemes que feien que només funcionés el clúster MPI de MESOS al node màster amb esclau local. Els problemes van ser els següents:

    - MESOS realitza una tècnica de *wrapper*, o adaptació, sobre la crida mpiexec de l'anell MPICH2, és a dir quan s'executa l'script mpiexec-mesos.sh s'inicia el procés d'administració de recursos demanant disponibilitats de recursos als nodes, i una vegada definits quants processadors i memòria estan disponibles llavors fa una crida parametritzada amb aquests valors a la crida mpiexec. Aquests passos es poden seguir al fitxer Python mpiexec-mesos.py. El problema que es va trobar és que quan s'iniciava la crida a l'script es tornaven valors erronis, com que no trobava les llibreries Python quan el seu *path* estava disponible des de qualsevol punt del sistema. La solució va arribar al fer una revisió del codi de l'script mpiexec-mesos.sh amb impressions per pantalla dels valors de cada línia dels *path*, per torbar l'origen. El problema es trobava que el sistema no interpretava bé la comanda d'adició +=, i la convertia en =, i l'origen d'aques problema és que l'script estava escrit per sessió *sh* i no *bash* que és el meu sistema. La solució final va ser aplicar el següent canvi a l'script:

        ```
        #! /bin/sh
        ```

      Canviar a:

        ```
        #! /bin/bash
        ```

    - Per definició un treball MPI necessita com a mínim un processador i 1 GB de memòria per a la seva execució, segons es pot veure al codi Python *mpiexec-mesos.py*[jj]. Com s'ha comentat en el punt anterior, el clúster pateix especialment de memòria i la configuració de MESOS fa que si no s'especifica en un la crida que inicia un esclau, el sistema per defecte només agafa la meitat de memòria disponible. En el nostre cas cap node esclau arribava a 1 GB. Es va arribar a aquesta

[jj] /mesos-0.20.0/src/mpi/mpiexec-mesos.py

conclusió després d'estudiar el codi del fitxer *containerizer.cpp*[kk], a la part de reserva derecursos de cpu i memòria, i al comprovar que si a l'esclau virtualitzat a VMware augmentàvem la memòria per a què el clúster es mostrés més d'1 GB, la crida a MPI (mpiexec-mesos) funcionava correctament. El problema principal és que la memòria dels nodes no virtualitzats és de 2 GB, i no es pot ampliar, per tant el clúster no podria funcionar. Es va consultar de nou la documentació disponible, i es va trobar que l'esclau disposa d'un *flag* en la crida d'activació per indicar la memòria amb la que s'iniciarà, i per tant posarà a disposició del clúster. També és possibles especificar altres recursos (*resources*) a disposició del clúster com el total de processadors. Així a partir d'ara les activacions dels esclaus es realitzen a partir de les següents crides:

Esclau 1
$ ./bin/mesos-slave.sh --ip=10.10.78.86 --master=10.10.78.85:5050 **--resources: "mem(*):1024;cpus(*):1"**

Esclau 2
$ ./bin/mesos-slave.sh --ip=10.10.78.85 --master=10.10.78.85:5050 **--resources: "mem(*):1024;cpus(*):4"**

Esclau 3
$ ./bin/mesos-slave.sh --ip=10.10.78.87 --master=10.10.78.85:5050 --resources: **"mem(*):1024;cpus(*):1"**

La modificació de valors de configuració al esclaus fa que en la següent execució s'han d'eliminar, per seguretat, configuracions anteriors. Això es realitza amb la comanda:

```
$ rm –f /tmp/mesos/meta/slaves/latest/
```

El *pool* de memòria passa ara a ser de 3 GB, tot i que podria ser fins 4,5 GB, però el procés MPI de MESOS només s'utilitzaran valors múltiples de 1024 i per tant mai arribarem a disposar dels 4,5 GB. Si el clúster es dedica a altres aplicacions sí seria possible canviar aquesta configuració per disposar de més memòria.

- Una vegada solucionats els problemes anteriors, va sorgir un nou problema a l'executar el clúster ja amb una tasca MPI a resoldre. El node esclau local al màster funcionava correctament, finalitzant les tasques MPI on només es dedica un processador. Quan es volia fer un treball afegint un esclau físic a l'anterior el procés en aquest esclau queda avortat immediatament, mostrant errors al fitxer de logs del tipus "*Failed to redirect stdout*"; "*Failed to chown*"; "*Failed to get user information for master*". En aquest punt el procés del projecte estava en dos parts problemàtiques, per un costat l'anell de dimonis MPD no s'iniciava de forma automàtica i per l'altre costat el clúster MESOS funcionava de forma correcta, aparentment, excepte a l'executar un treball MPI. Aquest primer anàlisi en va fer pensar que el problema era MPD que no estava ben instal·lat, i em vaig concentrar en solucionar els problemes fins que s'iniciés correctament segons el manual, com així vaig fer a partir d'aquell moment. El procés es pot seguir a la següent secció tot i que una de les tasques que es van fer al principi va ser modificar el fitxer mpiexec-mesos.py per a què no iniciés l'anell de dimonis MPD, l'anell era iniciat de forma manual, revisant el seu funcionament, i després s'intentava executar el treball MPI amb el clúster MESOS, fins i tot fixant un valor del port mpd màster fix per facilitar que els altres dimonis el trobessin (variable d'entorn de definició del rang de ports MPICH_PORT_RANGE:<port inicial>:<port final>). La solució no va arribar per aquesta via i llavors sí ens vam concentrar en solucionar el funcionament correcte, de manual, de l'anell de dimonis MPD abans de continuar amb més proves.

[kk] /mesos-0.20.0/src/slave/containerizer.cpp

Una vegada l'anell de dimonis MPD executava de forma correcta les tasques MPI es va provar de nou el clúster MESOS, però l'error era el mateix de l'inici, es a dir el problema no semblava tenir origen a l'anell MPD. L'estudi dels logs i la visita per fòrums de MESOS em vam portar la solució de forma indirecta[35], ja que el problema l'enunciava un usuari sobre un altre entorn d'aplicació no MESOS (Spark). El problema era que el el procés d'inici del node esclau no es feia amb privilegis de super usuari i per això no podia fer el canvis necessaris a un directori temporal on es volia executar el container amb l'aplicació MPI. Aquesta falta de privilegis feia avortar el procés al node esclau. La forma de solucionar-ho és afegir un nou *flag* a la crida (--no-switch_user) per evitar el canvi d'usuari, i que sigui l'usuari que llança la crida el que continuï la resta de processos. Així les crides als esclaus no locals al màster queden de la següent forma:

Esclau 1
$ ./bin/mesos-slave.sh --ip=10.10.78.86 --master=10.10.78.85:5050 --resources: "mem(*):1024;cpus(*):1" **--no-switch_user**

Esclau 3
$ ./bin/mesos-slave.sh --ip=10.10.78.87 --master=10.10.78.85:5050 --resources: "mem(*):1024;cpus(*):1" **--no-switch_user**

Igual ja s'ha indicat la modificació de valors de configuració al esclaus fa que en la següent execució s'han d'eliminar, per seguretat, configuracions anteriors. Això es realitza amb la comanda:

$ rm –f /tmp/mesos/meta/slaves/latest/

Una vegada posat en coneixement el problema i la solució al consultor, aquest em va expressar la seva estranyesa per què el clúster no inclogués mecanismes *setuid*, força comuns en aquests casos. Efectivament aquesta situació està recollida als scripts MESOS i les notes d'inici del clúster[36], però en el seu moment no vam saber interpretar l'abast d'aquella informació.

7.2.3 Instal·lació i execució entorn d'MPI – MPICH2

El principal problema en la instal·lació de les llibreries MPI en la seva versió MPICH2 de l'ANL, ha estat relacionat amb la meva inexperiència en aquests entorns, també en el meu baix coneixement del sistema operatiu Ubuntu utilitzat, o entorns Linux en general, i en unes descripcions tècniques d'instal·lació que si bé son perfectes en alguns punts, en altres una línia no massa descriptiva ha fet que no entengués exactament que estava passant. Aquesta falta de coneixement ha fet que trigues hores en donar-me compte de coses evidents, una vegada contemplada la solució. Un llistat d'aquestes situacions:

- Aplicar la solució MPICH2 segons el resum d'instal·lació que aporta MESOS, com s'ha indicat a la secció 4.4, feia referència a instal·lar la versió 1.2 d'aquestes llibreries. En poques línies s'indica que la instal·lació es realitzi des de zero seguint el manual, i amb poques indicacions de prerequisits. El manual d'instal·lació de MPICH2 fa una referència de només una línia a la necessitat d'establir comunicacions SSH:

  If you cannot get this to work without entering a password, you will need to configure ssh or rsh so that this can be done, or ***else use the workaround for mpdboot in the next step.***

  El cas és que el *workaround* funcionava perfectament, i pensava que tot ja estava finalitzat, i no va ser fins que vaig tenir problemes en les execucions sobre MESOS que vaig entendre que potser seria necessari aplicar SSH, sobre tot per què es busca iniciar l'anell de dimonis MPD de forma automàtica. La solució va passar

per instal·lar OpenSSH als nodes, crear claus públiques i traspassar-les als usuaris que s'havien d'accedir sense necessitat d'autenticar l'origen.

- Una altra confusió d'instal·lació ha estat en la part d'utilitzar NFS, al manual s'indica:

> *All should refer to the commands in the bin subdirectory of your install directory. It is at this point that you will need to duplicate this directory on your other machines* ***if it is not in a shared file system such as NFS.***

El problema és que la distribució de binaris s'ha de fer manualment a tots els nodes de l'anell, així que instal·lar un servei NFS és obligatori en un clúster. A més estalvia la instal·lació a mà a cada màquina de MPICH. La solució ha passat per instal·lar un servidor NFS al node màster, i els clients NFS als nodes esclaus, i crear un directori compartit (/mirror). El servidor NFS ha d'estar iniciat bans de llançar els dimonis mpd per a què l'anell funcioni correctament.

- Un error que he tingut en més d'una ocasió, i que m'ha fet quedar força astorat i sorprès, és que de sobte l'anell mpd no funcionava quan una estona abans si ho havia fet, i tots els *checks* indicaven que no existia cap problema de xarxa ni de configuració, i que per tant havia de funcionar. Què passava? Senzillament que al fer la crida de mpdboot s'ha de ser curós i pensar que permet engegar parcialment l'anell. Així aquestes crides no fan el mateix:

  - mpdboot –f mpd.hosts
  - mpdboot **–n 3** –f mpd.hosts

  Mentalment sabia que el fitxer tenia dos nodes a activar més el node iniciador, però si no s'indica explícitament mpdboot només inicia el node on es fa la crida. Així al executar la crida mpdtrace només es mostrava el node màster. El valor de n permet activar un, dos o els tres nodes del clúster segons els seu valor. Aquesta confusió m'ha fet perdre força temps en alguna ocasió i pensar que tenia alguna inestabilitat no resolta, quan no era així. La solució es ben senzilla: fer la crida de forma correcta.

- Un altre problema ha estat la configuració del fitxer de seguretat .mpd.conf que s'ha de deixar al directori $home de cada màquina. Vaig tenir un error i en una de les màquines aquest fitxer tenia més permisos dels que indica el manual, que els limita a lectura i escriptura per part de l'usuari propietari. L'execució de mpdboot només indicava un error de port en aquell node, i res sobre un problema amb la clau secreta. Un treball de comparativa de directoris, usuaris, *paths*, etcètera entre màquines va treure a la llum el problema, que tenia fàcil solució:

```
$ chmod 600 .mpd.conf
```

7.2.4 Instal·lació i execució de mètriques NPB

Aquesta part s'ha resolt sense incidències, només s'ha necessitat llegir la informació relacionada. Com a punts rellevants:

- Baixar el paquet adequat, en el nostre cas havia de ser un que tingués alguna prova de mètrica en MPI. En el nostre cas amb la versió NPB 3.3, i prova IS, aquesta necessitat quedava resolta.

- Adequar el valor de compilador al fitxer make.def, per a què reculli a *mpicc,* que l'obtenim d'instal·lar MPICH2. Resolta amb l'edició del valor corresponent.

```
#---------------------------------------------------------------------------
```

```
# This is the C compiler used for MPI programs
#--------------------------------------------------------------------------
MPICC = /home/master/mpich2-install/bin/mpicc
# This links MPI C programs; usually the same as ${MPICC}
CLINK   = $(MPICC)
```

- L'execució de *make clean* entre obtencions de binaris amb make.

7.2.5 Execució de mètriques al clúster

En aquest cas s'han trobat algunes dificultats addicionals per a poder executar en MESOS les llibreries PERFTEST-1.5, MPE sobre MPI, i també la representació gràfica, ja que fins aquell moment tots els entorns Linux eren de configuració server i no tenien entorn gràfic. Una descripció dels problemes i les solucions:

- Per obtenir un estudi d'eficiència sobre pas de missatges MPI, s'ha fet necessari instal·lar el paquet PERFTEST-1.5, que s'ha resolt directament consultant la documentació. Una vegada instal·lat és possible executar directament a un anell MPD la prova de rendiment com a paràmetre en la crida, per exemple la prova MPPTEST:

  ```
  $ mpiexec –n 3 mpptest –size 0 4096 32 –gnuplot –fname mpptest_mpd
  ```

  En el cas del clúster MESOS no és possible passar els paràmetres anteriors a la crida *mpiexec-mesos*, que posteriorment executa *mpiexec* amb els paràmetres del clúster (recursos), ja que només contempla un únic paràmetre: el de l'aplicació MPI a executar. Per resoldre aquest problema s'han creant un scripts que contenen els paràmetres de crida a la suite PERFTEST-1.5, bàsicament MPPTEST i GOPTEST amb les seves variants:

  ```
  $ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./<prova rendiment>
  ```

  Per exemple per a la prova per defecte mpptest següent:

  ```
  $ mpiexec-mesos –n 3 -m 1512 10.10.78.85:5050 ./mpptest
  ```

  On mpptest és un script, ha de tenir permisos d'execució, que conté el següent codi:

  ```
  #!/bin/bash
  mpptest –size 0 4096 32 –gnuplot –fname mpptest_mesos
  ```

- En la segona prova de rendiment es volia instrumentar l'execució de la mètrica IS, així era necessari compilar la mètrica amb les llibreries MPE que permeten obtenir aquesta informació. Per solucionar-lo s'ha indicat en els paràmetres que recull make.def de les mètriques NPB 3.3, i ha estat necessari fer de nou un *make* de la mètrica IS amb els paràmetres d'interès:

  ```
  #--------------------------------------------------------------------------
  # These macros are passed to the linker to help link with MPI correctly
  #--------------------------------------------------------------------------
  #CMPI_LIB  = -L/home/master/mpich2-install/lib -lmpi

  CMPI_LIB  = -L/home/master/mpich2-install/lib -mpe=mpilog
  #--------------------------------------------------------------------------
  ```

- La prova de rendiment és volia executar amb totes les configuracions del clúster: 1, 2 i 3 nodes, però les mètriques només permeten compilació amb valors de processador potència de 2. No hi ha solució per aquest problema.

- Tant les proves de PERTEST-1.5 com de la instrumentació tornen valors numèrics que es poden graficar amb GNUPLOT per fer més entenedors els resultats. En el cas de les llibreries MPE també se subministra un programa que permet visualitzar els resultats, apart d'eines d'anàlisi, com és JUMPSHOT. El problema és que totes les plataformes Linux utilitzades són de format server, i sense entorn gràfic X11, per exemple, que permeti treballa amb programes gràfics. Les proves per incorporar l'entorn X11 no van ser positives, i es van fer sobre una màquina virtual Ubuntu Server. Per evitar afectar alguns dels nodes es va decidir finalment instal·lar una versió Ubuntu Desktop 14.04 amb el programa GNUPLOT, i també instal·lant MPICH2 per poder treballar amb JUMPSHOT. Amb aquesta instal·lació, en forma de màquina virtual, es van resoldre els problemes de mostrar els resultats en forma gràfica.

7.3 Planificació real. Estudi de les desviacions produïdes

A la secció 1.5 es va presentar una planificació estimada del desenvolupament del projecte. Es tractava d'una aproximació realitzada des del desconeixement de la tecnologia, i per tant no es podia fer des d'un punt de vista de coneixements consolidats. El resultat natural són desviacions del pla proposat que ara presentarem de forma resumida.

7.3.1 Recerca i estudi sobre l'estat de l'art respecte als clústers de computació

Tasques a desenvolupar:

- Recerca d'informació.
- Crear un document resum sobre la situació actual per aportar a la memòria del projecte.

La projecció temporal d'aquesta fase va ser la següent:

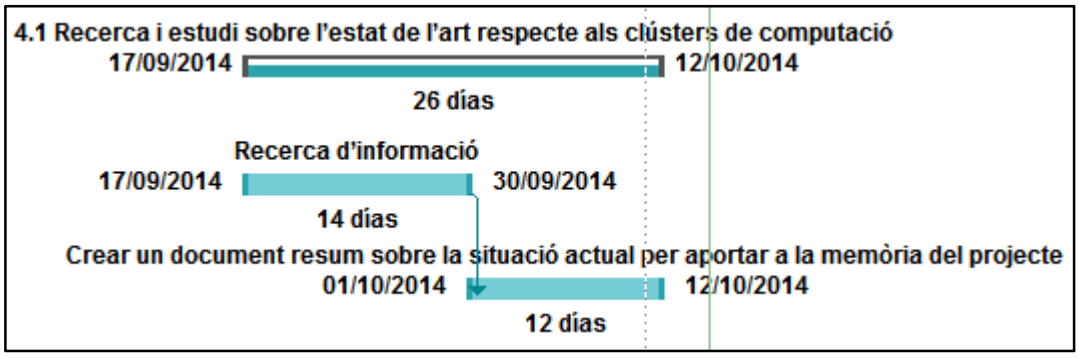


Figura 60. Planificació temporal inicial de Recerca i estudi sobre l'estat de l'art respecte als clústers de computació.

En total la seva duració es va estimar en vint-i-sis dies.

La realitat temporal d'aquests treballs:

17/09/2014 a 05/10/2014.
Treballs de documentació.
Llegir "*The Datacenter as a Computer An Introduction to the Design of Warehouse-Scale Machines*".
Visionar els videos *"Builnding and running distributed systems using Apache Mesos", "Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center" , "Run your data center like Google's with Apache Mesos", "Mesos: A Platform for Fine-Grained Resource Sharing in Datacenters", "Spark on Mesos", "Run Apache Spark on Apache Mesos"*.
Veure presentacions: *"Datacenter Computing with Apache Mesos - BigData DC", "Introduction to Apache Mesos", "Chronos: A distributed, fault-tolerant and highly available job orchestration framework for Mesos"*.

19/09/2014.
Estudi d'execució i arquitectura MESOS.

22/09/2014 a 30/09/2014.
Recerca d'informació per planificar el projecte i preparar la PAC 1. Cerca d'informació sobre clúster

08/10/2014 a 15/10/2014.
Treball per a PAC 1. Redacció de la motivació del projecte, objectius, tasques.

16/10/2014.
Refer PAC 1 amb canvis suggerits pel consultor. Lliurament el 25/10/2014.

Els treballs de documentació van durar més del previst, del 19 de setembre fins al 5 d'octubre, pel desconeixement en la matèria, també per què pràcticament tota la documentació està en anglès, i hi ha molt material en vídeo de llarga durada (més de trenta minuts la majoria). La redacció dels documents va coincidir a més amb les primeres tasques de configuració del clúster.

7.3.2 Plataforma d'execució del clúster privat sobre el que es faran proves

Tasques a desenvolupar:

- Depuració de maquinari per tal d'aconseguir màquines per instal·lar el clúster.
- Definició de la xarxa local sobre la que funcionarà el clúster.
- Instal·lació del sistema operatiu necessari a tots els nodes.
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase va ser la següent:

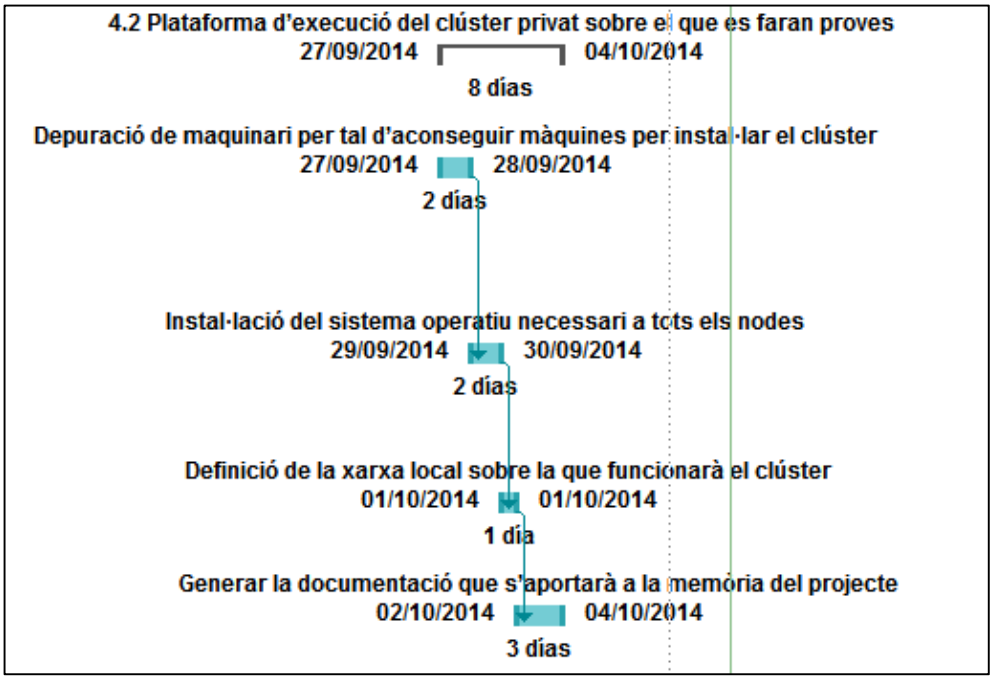


Figura 61. Planificació temporal inicial de Plataforma d'execució del clúster privat sobre el que es faran proves.

En total la seva duració es va estimar en vuit dies.

La realitat temporal d'aquests treballs:

20/09/2014 i 21/09/2014.
Es fa revisió i neteja del maquinari disponible, es configuren al màxim de recursos possibles per màquina.

17/10/2014.
Configurar la xarxa local del clúster.

04/11/2014
Avui s'ha dedicat el temps a revisar la part de xarxa, instal·lació de MESOS i execució al clúster amb les noves configuracions dels esclaus.

05/11/2014.
Finalitzar la part d'implementació física del clúster iniciada ahir.

En aquest cas el temps ha estat inferior, només cinc dies enfront dels vuit inicials, però s'han fet canvis de sistema operatiu per problemes en una etapa que en teoria el clúster ja havia d'estar consolidat.

7.3.3 Recerca i estudi d'informació sobre MESOS

Tasques a desenvolupar:

- Que és MESOS i quins components el componen?
- Instal·lació i execució dels components de la plataforma als nodes esclaus i mestre.
- Administració bàsica de la plataforma MESOS.
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase va ser la següent:

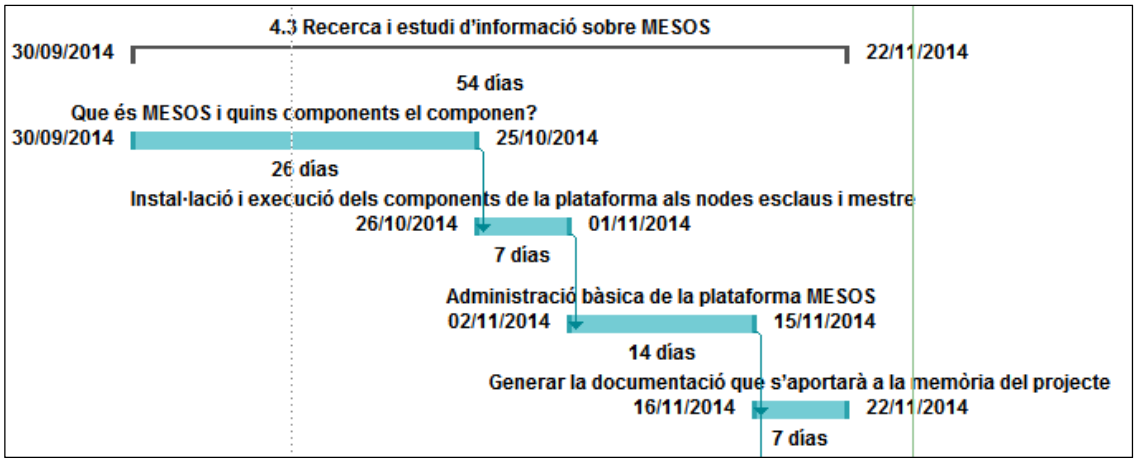


Figura 62. Planificació temporal inicial de Recerca i estudi d'informació sobre MESOS.

En total la seva duració es va estimar en cinquanta-quatre dies.

La realitat temporal d'aquests treballs:

18/10/2014.
Cercar informació sobre l'execució de tasques al clúster MESOS.

19/10/2014 a 25/10/2014.
Instal·lació de MESOS al clúster. Hi han problemes tan a nivell de maquinari com de sistema operatiu. Al llarg d'aquest dies es van solucionant: memòria és necessari almenys 2 GB per màquina, i la versió Ubuntu

Server 12.04 no funciona de forma correcta (vincles trencats). Es decideix instal·lar una superior abans d'abandonar la idea de crear un clúster privat. Es passa de tres nodes físics i un node virtualitzat, a dos nodes físics i un node virtualitzat per distribuir la memòria i arribar als 2 GB per màquina física.

26/10/2014
Descarregar imatge d'Ubuntu Linux 14.04 LTS i instal·lar a la màquina virtual. Instal·lar també MESOS 0.20.0. S'instal·len correctament.

28/10/2014
Instal·lar Ubuntu Linux 14.04 LTS i MESOS 0.20.0 als esclaus 1 i 3. S'instal·la correctament.

En aquest cas sembla que la instal·lació de MESOS i la documentació s'ha resolt de forma mol ràpida, només deu dies fins al s vint inicials. però nomes es reflecteix la fase inicial- Després amb l'execució de l'entorn d'aplicació s'ha tornat a reconfigurar el clúster, per tant no es pot considerar com una única fase, com es fa a la planificació, per aquesta interacció. La part de la memòria i documentació està inclosa en la preparació de la PAC 2.

7.3.4 Recerca i estudi d'informació sobre treball distribuït a la plataforma MESOS

Tasques a desenvolupar:

- Entorns disponibles per a la plataforma MESOSPHERE.
- Triatge d'almenys un entorn per fer proves distribuïdes.
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase va ser la següent:

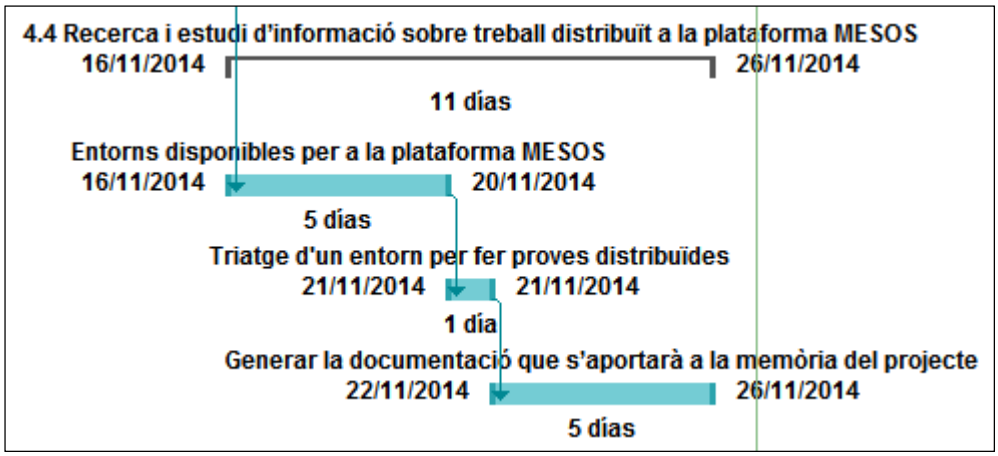


Figura 63. Planificació temporal inicial de Recerca i estudi d'informació sobre treball distribuït a la plataforma MESOS.

En total la seva duració es va estimar en onze dies.

La realitat temporal d'aquests treballs:

29/10/2014
Estudiar quin entorn d'aplicació es pot fer servir per executar aplicacions de forma distribuïda, es decideix que sigui MPI i l'aplicació serà una mètrica del NAS NPB.

7.3.5 Recerca i estudi d'informació de metodologia sobre sistemes distribuïts

Tasques a desenvolupar:

- Recerca d'informació
- Disseny d'una metodologia d'avaluació aplicable

La projecció temporal d'aquesta fase va ser la següent:

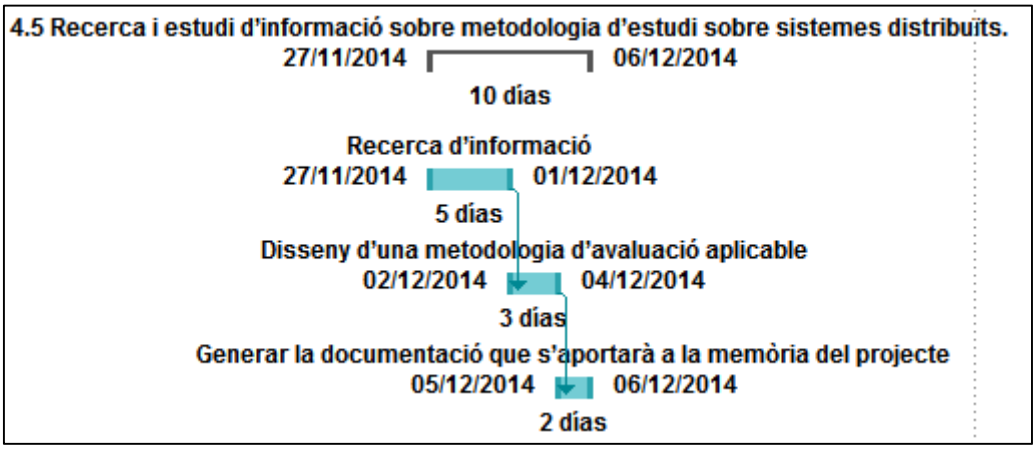


Figura 64. Planificació temporal inicial de Recerca i estudi d'informació sobre metodologia d'estudi sobre sistemes distribuïts.

En total la seva duració es va estimar en deu dies.

La realitat temporal d'aquests treballs:

29/11/2014.
Es fa un primer anàlisi de les mètriques ha utilitzar en el clúster. S'estudien les execucions de MPE, llibreria d'instrumentació, i com configurar-les, també quina configuració de nodes a nivell de memòria seria les més aproximada per a execucions anell MPD i anell MPD MESOS, que es decideix que sigui 1.512 MB.

30/11/2014.
Instal·lar X11 a un Ubuntu server virtual per provar l'execució de JUMPSHOT i fer gràfiques de les sortides MPE. Es descarta per què no és possible administrar les finestres. Es decideix que s'ha d'instal·lar un Ubuntu Desktop només per obtenir les sortida gràfica dels programa Jumpshot. Es decideixen les proves de mètriques a fer servir, que serà la única al paquet NPB 3.3 i es tracta de IS (*Integer Sort*)

01/12/2014.
S'instal·la Ubuntu Desktop 14.04 en format VM, i dóna un error per què no troba la llibreria jumpshot.jar, tot i configurar NFS i la carpeta /mirror. Es decideix que s'ha d'instal·lar també MPICH2 per facilitar la solució.

02/12/2014.
S'instal·la MPICH2 a Ubuntu Desktop 14.04 i funciona correctament Jumpshot amb dades MPE. Es troba per Internet que és possible fer un estudi d'eficiència en clúster, del traspàs de missatges MPI, utilitzant eines compatibles amb MPICH2 del propi ANL, com són MPPTEST i GOPTEST, només s'ha d'instal·lar PERFTEST-1.5. Es descarrega de l'ANL i s'instal·la sense problemes al node màster, carpeta /mirror per què els nodes esclaus també hauran de tenir accés als binaris. També s'instal·la Gnuplot a Ubuntu Desktop 14.04 per obtenir la sortida gràfica de les dades que generen els tests PERFTEST-1.5.

03/12/2014.

S'executen proves de test de PERFTEST-1.5, i funciona correctament. Es decideix que les proves d'execució seran: mpptest, async, bisect, overlap, logscale i goptest. Les proves es faran a la configuració anell MPD i anell MESOS, per analitzar si existeixen eficiències diferents.

04/12/2014
S'executen les proves de test definitives de pas de missatge MPI. Aquests test es fan sobre l'anell MPD i anell MPD MESOS.

06/12/2014.
S'executen les proves de test definitives sobre la mètrica IS en configuració clúster d'un i dos nodes, i instrumentades amb MPE MPICH2 Aquests test es fan sobre l'anell MPD i anell MPD MESOS.

09/12/2014.
Extracció de conclusions sobre les proves de mètriques al clúster, i passar dades a la memòria del projecte.

7.3.6 Estudi sobre MESOS en l'entorn triat

Tasques a desenvolupar:

- Instal·lar el entorn sobre la plataforma MESOS.
- Executar aplicacions de prova per demostrar les capacitats distribuïdes de MESOS aplicant la metodologia dissenyada al punt 4.5
  - Tolerància a errors
  - Eficiència i distribució dinàmica
- Generar la documentació que s'aportarà a la memòria del projecte.

La projecció temporal d'aquesta fase va ser la següent:

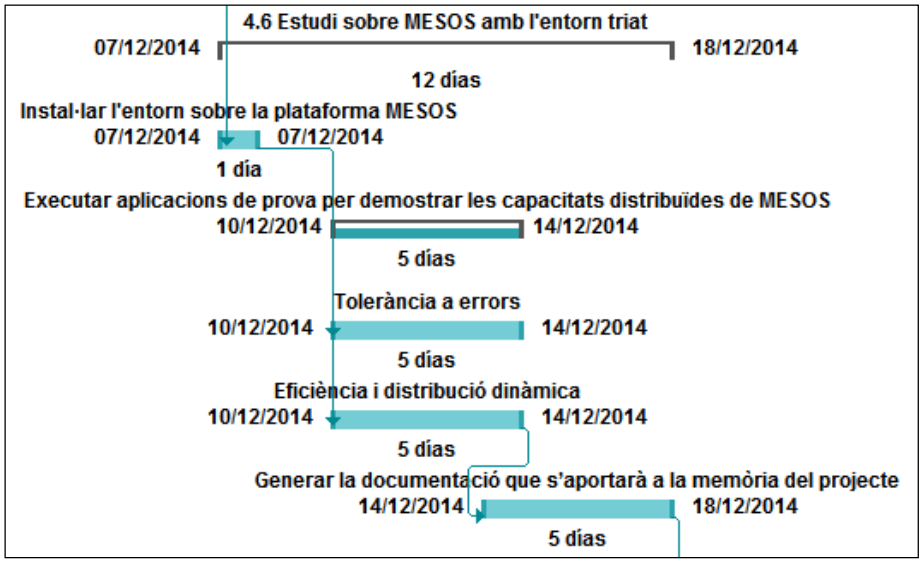


Figura 65. Planificació temporal inicial de Estudi sobre MESOS amb l'entorn triat.

En total la seva duració es va estimar en dotze dies.

La realitat temporal d'aquests treballs:

30/10/2014
Descàrrega i instal·lació de MPICH2. Descàrrega del paquet NPB 3.3. Proves d'execució amb el compilador mpicc al node màster.

02/11/2014.
Continuen els problemes d'execució MPI en MESOS. No es reconeix mpiexec-mesos dins d'un mini clúster master-slave. Dóna error amb les llibreries python. La recerca a la xarxa no dóna resultats. He provat de baixar i instal·lar MESOS-HYDRA, però està enfocat al clúster en MESOSPHERE i per tant necessitaria instal·lar també HADOOP. De moment quedarà descartat.

10/11/2014.
Instal·lar MPICH als esclaus 1 i 3. L'anell funciona correctament si es publica el node servidor i els nodes remots són vinculats de forma manual amb *mpd –h <hostname> -p <port>*, però hauria de funcionar amb *mpdboot –n <nodes> -f <fitxer de nodes>*.

14/11/2014.
Prova de l'anell MPI en local. Error d'execució per què *mpiexec-mesos* no troba les llibreries Python.

17/11/2014.
Anàlisis del fitxer script *mpiexec-mesos* per trobar l'error.

18/11/2014.
Anàlisis del fitxer script *mpiexec-mesos* executant línia a línia. Es troba que el problema és que la consola ha de ser *bash* i no *sh*, per què no fa la funció d'adició '+='. Es canvia i funciona sense problemes en local.

19/11/2014.
Es configuren les mètriques NPB 3.3, bàsicament descomprimir, al node màster. Es modifica el fitxer make.def per què reconegui el compilador *mpicc*. Es fan proves de compilació i sembla funcionar correctament.

20/11/2014.
La memòria a les màquines ha de ser superior a 1 GB lliure en les instàncies que fa MESOS, on indica els recursos disponibles. Es modifica la màquina virtual per tenir més memòria, però els nodes esclaus no arriben a 1 GB en instància MESOS, a més sembla necessari configurar sessions SSH i crear una carpeta compartida NFS, segons el manual de MPICH2.

21/11/2014.
Per anàlisi d'informació a Internet es troba una solució de configuració als esclaus per aconseguir més memòria RAM lliure, configurant els recursos que poden oferir. Es decideix que a la crida d'execució dels esclaus el valor sigui *--resources="mem(*):1024;cpus(*):1"*. Ara si que a l'executar l'entorn d'aplicació totes les instàncies dels esclaus són reconegudes, però quan s'executen als esclaus remots el contenidor es cancel·lat de forma immediata, i l'execució es queda sense finalitzar. Es prova de fixar un port fix iniciant MPD de forma manual, variant el fitxer *mpiexec-mesos.py*, però no funciona.

22/11/2014.
MESOS en execució MPI funciona però només en el node local, el que executa el màster i l'esclau MESOS. Per l'error que es mostra al cancel·lar el contenidor *'Failed to chown'* es pensa que pot ser per què s'ha de crear un usuari que permeti executar MPD als nodes esclaus via SSH. S'instal·la SSH i NFS, i es crea la carpeta /mirror compartida per tots els nodes. El problema continua.

23/11/2014.
Funciona correctament l'anell MPD, inici a partir de *mpdboot*, es confirma que feia falta instal·lar i configurar SSH i la carpeta compartida. L'execució d'anell MESOS no funciona correctament. Per Internet es troba que el problema està que l'usuari màster vol modificar una carpeta temporal als nodes esclaus, i al no tenir

permisos avorta el procés. La solució és indicar un *flag --no-switch_user* a l'execució dels nodes esclaus, a partir d'aquest moment tenim una configuració d'anell MPD i anell MPD MESOS totalment operatives.

Aquesta part ha durat més del previst pels problemes trobats en l'execució del clúster MESOS amb l'entorn d'aplicació, que inclouen la seva instal·lació per separat del clúster. Ha estat la part més problemàtica de tot el projecte, amb problemes de maquinari i de lògica de funcionament, i ha estat necessari reconfigurar i estudiar de nou el funcionament dels components de MESOS. A més segons s'observa en el desenvolupament temporal, aquesta fase s'ha executat abans del previst, la raó és que quan es va planificar a l'inici del projecte es va pensar que seria la fita final,però en realitat la fita final és la fase de mètriques on s'analitza el rendiment de l'entorn d'aplicació MPI, per tant era necessari que l'entorn funcionés abans.

7.3.7 Anàlisi de la instal·lació de MESOS sobre un conjunt de milers de nodes

Tasques a desenvolupar:

- Com s'implementaria?
- Estudi diferencial amb la solució pràctica mínima desenvolupada.
- Generar la documentació que s'aportarà a la memòria del projecte.

Superada la fase pràctica del projecte, en aquest punt l'objectiu principal seria el com projectar el coneixement adquirit en una instal·lació massiva de nodes. Un clúster on per exemple factors com la fallada de màsters i nodes pot ser crítica, també l'eficiència del sistema o el tipus d'informació que el clúster ha de tractar, que serien pilars bàsics del seu disseny. Es per tant trobar les diferències bàsiques entre un clúster privat petit, com l'utilitzat, i un clúster real de producció. En aquest sentit també es vol presentar alguna solució distribuïda de proves per a enginyeria, i que estan disponibles com a servei de pagament.

La projecció temporal d'aquesta fase va ser la següent:

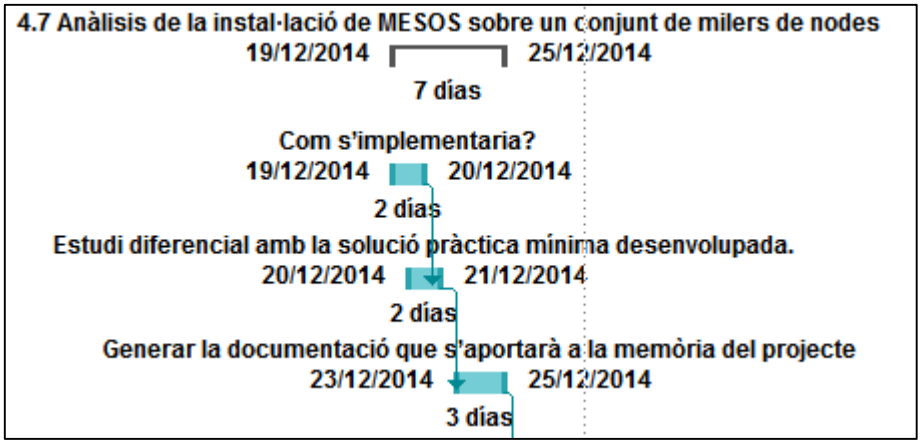


Figura 66. Planificació temporal inicial d'Anàlisi de la instal•lació de MESOS sobre un conjunt de milers de nodes.

En total la seva duració es va estimar en set dies.

Aquesta part no s'ha desenvolupat, per limitacions econòmiques i de temps, en canvi s'ha fet un estudi teòric que s'ha inclòs en la secció *4.5 Instal·lació Mesosphere com a IaaS*.

7.3.8 Conclusions de l'estudi

La part final de qualsevol treball d'investigació finalitza amb les conclusions personals que s'han pogut extreure de tot el treball realitzat. En aquesta fase finalitza la recerca i implementació del projecte, i es completa la seva memòria.

La projecció temporal d'aquesta fase va ser la següent:

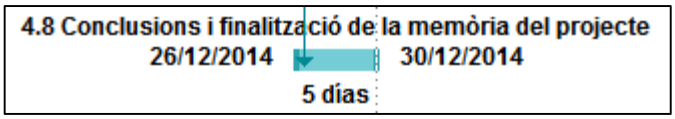


Figura 67. Planificació temporal inicial de Conclusions i finalització de la memòria del projecte.

En total la seva duració es va estimar en cinc dies.

La realitat temporal d'aquests treballs:

A partir del 09/12/2014 s'ha iniciat la fase de composició de la memòria, que inclou reflectir les conclusions de l'estudi. Tot i que en les diferents fases s'havia de construir la part documental, la realitat ha esta que s'ha dedicat el temps a les implementacions tècniques i agafar notes de cara a la memòria. Només les fites de lliurament de les PAC han fet que s'avancés el treball de redacció de la memòria. A partir del 21/12/2014 s'editarà la presentació, diapositives, del projecte.

7.3.9 Comparativa final

A continuació és pot veure de forma resumida la duració planificada del projecte, en la figura 68, i l'execució temporal real a la figura 69.

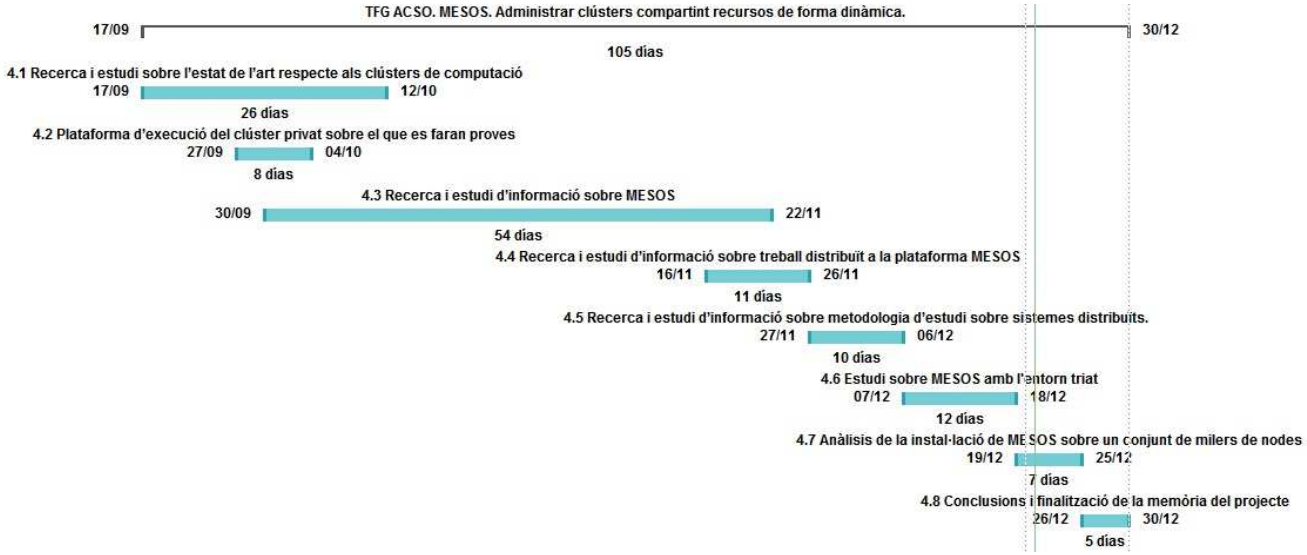


Figura 68. Planificació temporal inicial del projecte.

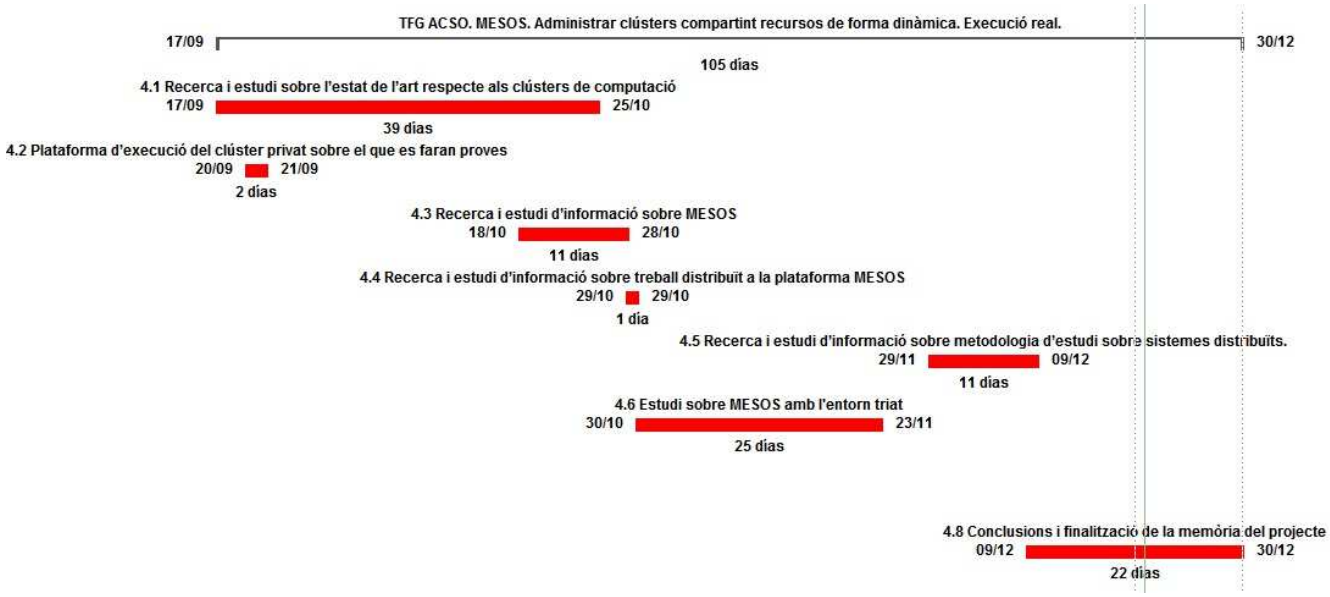


Figura 69. Planificació temporal real del projecte.

7.4 Propostes d'estudi

El projecte només ha mostrat una part de les característiques del gestor MESOS. Motius com el temps limitat, i els recursos materials disponibles, fan que no s'abordin més característiques molt interessants, o fins i tot una prova d'eficiència més propera a un entorn real d'execució. Per aquest motiu presento una llista d'objectius d'estudi que m'hauria agradat completar, i que poden donar una visió més exacta de les possibilitats del gestor MESOS:

- Desplegar el gestor MESOS en un clúster format per desenes de servidors i executar característiques com les descrites a continuació.

- Clúster MESOS d'alta disponibilitat[12]. El model provat només disposa de tres nodes esclaus i un node màster. Es tractaria d'assegurar la substitució d'un node màster caigut per un altre de reserva, i tot de forma automàtica. La tecnologia MESOS permet fer-ho de forma automàtica mitjançant Apache Zookeper[37], que és una característica que gestiona els nodes màsters, els esclaus, i l'elecció i reconfiguració i d'un nou node màster en cas necessari.

- Utilitzar altre eines de gestió, que permeten un control més exhaustiu del clúster. Per exemple:
    - CHRONOS[38]: substitut de CRON com a planificador de tasques amb tolerància a errors.
    - Anàlisi de xarxa[39]. A partir de la versió utilitzada hi ha la possibilitat d'estudiar el comportament del clúster analitzant la sortida de cada node esclau, per analitzar possibles problemes. En la última versió (0.21.0) fins i tot es pot analitzar el comportament d'un container concret per evitar que aquet pugui limitar la xarxa. Els valors que es poden monitoritzar en l'esclau són els següents:

        - *net_rx_bytes*
        - *net_rx_dropped*
        - *net_rx_errors*
        - *net_rx_packets*
        - *net_tx_bytes*
        - *net_tx_dropped*
        - *net_tx_errors*
        - *net_tx_packets*

- Inici automàtic del clúster. És possible fer-ho estudiant els scripts específics que la plataforma MESOS posa a la nostra disposició[36]. En aquest projecte, al ser de mida reduïda, no ha estat necessari per què amb sessions remotes s'ha pogut fer a partir d'uns scripts molt simples.

- Provar altres entorns d'aplicació de la llista enunciada a la secció 3.1.

- En un nivell molt més avançat es podria estudiar la realització d'un entorn d'aplicació[40] d'interès, a partir de les eines que la comunitat MESOS posa al nostre abast[41]. Aquest entorn després quedaria a disposició d'altres projectes.

Per últim indicar que seria molt interessant endinsar-se en les bases de disseny del sistema. En com s'ha definit un canvi de paradigma que sembla tan senzill, i que està inspirat en l'aïllament basat en contenidors, natius Linux o Docker, que permeten una capa de gestió alhora lleugera, però potent i flexible, i veure tot un conjunt de servidors com a un únic repositori de recursos (*pool*). Crec també que la idea anirà més enllà de considerar un *datacenter* limitat a un edifici físic, i en un futur pròxim podrem veure solucions basades en MESOS, però que permetran compartir recursos de forma distribuïda geogràficament, gràcies a les altes velocitats de transferències de xarxes. D'aquesta forma l'aprofitament de recursos i la resistència a errors, o disponibilitat de recursos, seran molt altes. També serà interessant analitzar que aportarà MESOSPHERE DCOS, ideat com a un sistema operatiu per la centre de dades, i què es té previst el seu llançament a principis del proper any.

7.5 Autoavaluació

Una de les fases més importants, o potser la més important, d'un projecte qualsevol a la vida és la aprendre dels errors propis. Aquests poden ser tant comesos per nosaltres mateixos com de la resta de companys o contractistes que intervinguin. La premissa bàsica és que mai, sobretot si és la primera vegada que ens enfrontem a un projecte de certa entitat, tot sortirà segons el previst. L'habilitat que hem de desenvolupar és la de treure profit d'aquests errors, analitzant els seus motius i les solucions aplicades, i així anar augmentant la nostra experiència personal. Només d'aquesta forma podem evitar repetir alguns aspectes desagradables, i quan es produeix algun error tenir almenys algunes possibles alternatives.

De forma resumida els punts importants d'aquest autoanàlisi són els següents:

- Com és natural és pot afirmar és que tot i haver complert els objectius marcats, sempre hi han coses a millorar o nous objectius que fins llavors havien passat desapercebuts. Queda patent en el llistat d'estudis a continuar sobre aquestes tecnologies reflectit a la secció 7.4.

- El projecte tal i com està plantejat és una tasca a abordar, organitzar i realitzar per una sola persona. Cosa que dificulta la qualitat final, ja que no és recomanable fer de director, analista, programador i provador... Aquesta "múltiple personalitat" no permet seguir fil per randa una certa planificació, ni una tria adequada de decisions per falta de coneixements, o fins i tot discussions productives. Així triar els companys de viatge, si és possible, és una de les tasques més importants per garantir l'èxit.

- Un punt important és el desconeixement de la matèria a desenvolupar, en aquest cas la plataforma MESOS. Fins a la realització del projecte tenia el recull bàsic dels conceptes teòrics adquirits als llarg dels estudis. Conceptes com treball distribuït, sistemes operatius, estudi de mètriques, computació paral·lela, arquitectura de computadors, xarxes de computadors i protocols de comunicacions. Tots ells de manera més o menys destacada amb anat sorgint en la realització del projecte: disseny i configuració de la infraestructura física, configuració del sistema operatiu i del sistema de gestió del clúster, treball distribuït de computació paral·lela entre els diferents nodes amb MPI i anàlisi de l'execució en forma de mètriques. A l'inici del projecte no tenia una visió de com de transversal ha acabat per definir-se, i com han interaccionat fins que s'han convertit en una cosa sòlida, en un sistema què, tot i molt limitat de recursos, funciona segons els previst.

El desconeixement previ ha de significar només una cosa: prudència en la tria d'objectius i en la planificació temporal. En aquest cas es podia resoldre amb pocs objectius a realitzar, que es podien ampliar si fos necessari, i un calendari amb petites toleràncies. També amb la revisió constant del punt de realització, per rectificar l'abans possible. La importància de fer-ho rau en que en un projecte real no pots avisar al teu client al final del temps previst, que allò és irrealitzable. Un anàlisi constant permet reajustar terminis, i donar explicacions a temps. Així no cal crear grans objectius, si és té una bona idea, la senzillesa d'execució serà la que garantirà el seu èxit.

- Els projectes de qualsevol tipus necessiten de comunicació constant entre els membres que formen l'equip: per fixar objectius, guiar treballs, resoldre dubtes i validar les solucions aplicades, i també motivar en els moments de major dubte en la continuïtat del treball. El projecte només comptava amb el seguiment de dos persones, però crec que aquesta part s'ha realitzat de forma excel·lent, sobretot si pensem en que la comunicació s'ha resolt sempre per correu electrònic, que afegeix una certa dificultat de temps de resposta, però pocs han estat els correus que no han tingut resposta en poques hores facilitant les solucions.

- La memòria és feble i millor tenir-ho documentat per evitar recorre de nou els mateixos camins. En el cas d'aquest projecte es va crear una base de dades dedicada a anar recollint els treballs diaris, dubtes a resoldre, etcètera. També el suport del paper en forma de diari, més flexible i ràpid d'apuntar el que es feia a cada moment.

El camí no ha estat fàcil sobretot per què el domini temporal és molt petit. L'inici dels primers estudis sobre la temàtica es van iniciar al setembre, una mica abans de l'inici del semestre, i fruït de la constant comunicació que hem mantingut entre consultor i alumne. El resultat final és una introducció als sistemes de gestió dels centres de processament de dades i càlcul, i amb tecnologies que encara estan en fase avançada de desenvolupament, i que per tant estaran encara més presents en un futur proper.

7.6 Conclusions finals

De forma molt breu, i com ja s'ha vist en la documentació precedent, el projecte s'ha dividit en diverses fases:

- Estudi de solucions actuals a la gestió de clústers de computació. De forma específica estudiar el gestor MESOS.

- Estudi de la implantació del gestor MESOS en un clúster de pocs recursos, bàsicament PC domèstics.

- Tria d'un entorn d'aplicació sobre el que llançar execucions, i la seva instal·lació i execució.

- Comparació d'eficiència entre un clúster amb gestió MESOS i sense gestió MESOS.

La informació resumides que hem pogut copsar d'aquest procés són:

- Hem pogut estudiar diferents tecnologies actuals que intenten mitigar els problemes de gestió de recursos als clúster de computació. Com a destacable la comparació entre gestors monolítics, de dos nivells i d'estat compartit.

- Hem pogut estudiar la base que permet l'aïllament d'aplicacions a MESOS, que són els contenidors. D'aquests a més veure la importància que els contenidors DOCKER ofereixen a la virtualització enlloc d'utilitzar màquines virtuals.

- Ens hem enfrontat a les dificultats d'implementar un clúster real. Aquestes han estat de tota mena: maquinari, sistemes operatius, configuracions de xarxa, llibreries, etcètera.

- Hem tingut de dissenyar un pla d'anàlisi de rendiments del clúster amb les eines disponibles, on es volia comprovar l'afectació que MESOS podia provocar al rendiment del clúster. Això ha obligat a estudiar quines mètriques i de quina forma s'havien d'executar al clúster.

- Respecte al gestor MESOS, aquest no necessita de grans recursos per funcionar, una màquina de baixos recursos com és un PC domèstic ha permès la seva instal·lació i funcionament, i una revisió dels paràmetres de recursos utilitzats ens han donat una idea de com de 'lleuger' per al sistema és aquest gestor, tant en la part màster com esclau. Així el gestor no treu grans recursos del sistema, sobretot si pensem que la plataforma d'instal·lació seran servidors amb recursos operatius molt més grans que els utilitzats en aquest projecte.

- L'entorn d'aplicació utilitzat per executar MPI bàsicament és una crida emmascarada, es a dir s'ha d'instal·lar tot el programari necessari com si no fes falta el gestor MESOS, i després la seva execució es realitza per una crida simple des de MESOS que és qui pot oferir tots els recursos disponibles en aquell moment. Es una forma simple d'execució, que a més permet que si tenim algun problema al gestor MESOS, podrem executar l'aplicació sobre la plataforma original.

- Una vegada instal·lades les dos plataformes el funcionament és estable i sense errors.

- La comparació d'eficiències amb gestió de clúster MESOS i sense, no aporta diferències destacables, a causa de les breus explicacions dels punts anteriors: baixa demanda de recursos de MESOS i crida emmascarada a la plataforma MPI.

A partir dels punts anteriors podem indicar el següent com a conclusions de l'estudi:

1. El gestor MESOS és funcional i robust, permet la seva instal·lació en entorns molt bàsics, i fins i tot en el seu desplegament mínim, ja que no hem pogut copsar la seva eficiència en entorn d'alta demanda i disponibilitat (màsters redundants, esclaus autorecuperables, etcètera). És a dir fa el que diu que ha de fer: gestió de recursos d'execució disponibles quan una aplicació així ho demana, i també informació en temps real del que està passant als diferents *hosts* del clúster.

2. L'entorn d'aplicació instal·lat permet fer l'execució en paral·lel MPI amb els recursos disponibles en el seu llançament. Aquests recursos els gestiona MESOS, i els posa a disposició de cada nova execució, de forma que amb recursos suficients seria possible fer diverses tasques de forma paral·lela, o bé tenir una cua gestionada per a la seva execució el més ràpidament possible.

3. L'eficiència del clúster no està afectada per la instal·lació del gestor MESOS, i la peculiaritat de l'entorn d'aplicació triat al projecte fa que fins i tot no sigui necessari instal·lar MESOS, llavors ens podem preguntar si és necessari fer ús. La resposta és sí pels següents motius:

    a. La baixa necessitat de recursos fa que instal·lar MESOS no impliqui sacrificar rendiment del clúster.

    b. El gestor MESOS fa que veiem el clúster de forma simplificada. Passa de ser un gran conjunt de màquines i sistemes de comunicació i gestió, a un símil de computadora de gran escala amb recursos sota una única gestió.

    c. El clúster s'ha de dissenyar per aprofitar al màxim els seus recursos, no seria una bona pràctica crear una cua de treballs pendents a l'espera d'execució tipus FIFO, si els recursos no estan esgotats al clúster. Es a dir s'han de permetre el màxim de convivències d'execució.

d. El gestor MESOS és un *kernel* de base que interactua entre els recursos d'execució i els entorns d'aplicació, que a la seva vegada són els mediadors entre aplicacions i MESOS. De forma que només és necessari un *kernel* MESOS per host, i després tants entorns d'aplicació com tecnologies diferents es vulguin executar. Les execucions de cada aplicació es realitzen des del màster, de forma que podem tenir diferents aplicacions executant-se en paral·lel, i aprofitant al màxim els recursos disponibles tot el temps. Es a dir no és una configuració monolítica en que el clúster s'ha dissenyat per a una única aplicació. L'únic límit són els clústers on el maquinari està dissenyats per a un tipus de còmput intensiu concret, però com s'ha vist a la secció 3.1 els entorns d'aplicació són múltiples pensats per a serveis distribuïts amb base Internet, per tant en general la seva instal·lació és recomanable en una gran quantitat d'escenaris.

e. La capa MESOS que s'introdueix permet la convivència d'arquitectures no homogènies, en el cas del projecte s'ha vist amb PC de diferents capacitats i recursos. De forma que pot facilitar les actualitzacions progressives de maquinaris.

f. Al projecte no hem pogut demostrar altres bondats que incorpora MESOS, i que han de reforçar la gestió, com són els planificadors de treballs, el control d'inici remot dels esclaus i la figura d'un node màster redundant per als clústers d'alta disponibilitat. És a dir peces que poden conformar un clúster molt potent, i que estan a l'abast de qualsevol empresa que ho necessiti al ser tot de llicència *open source*.

4. Hem pogut copsar les dificultats i les tecnologies que ajuden en la gestió de recursos als clústers. S'han pogut entendre les seves diferències funcionals més importants.

5. Hem millorat el coneixement general en sistemes operatius, configuració de xarxes i treball distribuït.

De forma personal crec que estem davant d'un canvi de paradigma important en la gestió de centres de computació, ja que s'apropen tecnologies que abans només estaven a l'abast de grans companyies, i pel propi model simplificat de gestió: el centre ara és un PC, tots els recursos estan a disposició de les aplicacions, i no és necessari endinsar-se en la complexitat del model d'arquitectura. La capa MESOS ho simplifica de cara al gestor. A més és un projecte relativament nou i viu, que farà que veiem cada cop més prestacions i entorns d'aplicació disponibles, així crec que qualsevol eina o entorn d'execució distribuït estarà present en poc temps i si no és així es pot col·laborar en el seu desenvolupament. També s'ha de fer seguiment a MESOSPHERE DCOS, que es presentarà a inicis de l'any 2015 i que vol simplificar encara més la gestió de recursos treballant com a sistema operatiu de clústers, i al competidor OMEGA de Google, amb una gestió de recursos que sembla més flexible i potent.

8. Glossari

benchmark: Veure mètrica.....12
Big Data: referit a sistemes que manipulen grans quantitats de dades.....25
bisection bandwith: Es coneix com l'ample de banda màxim entre les dos parts que formen la xarxa del sistema, si aquesta és divideix en dos parts iguals. http; //en.wikipedia.org/wiki/Bisection_bandwidth 62
cicle de vida en cascada: Desenvolupament de tasques seqüencial amb pas limitat a finalitzar la tasca anterior per poder continuar amb la següent. 8
cloud computing: Computació en el núvol. Capacitat d'oferir serveis gràcies a l'agrupació de recursos i la seva compartició i accés mitjançant a Internet. (http; //en.wikipedia.org/wiki/Cloud_computing) 6
clúster: Conjunt d'elements de computació i els seus serveis de suport. 7
commutador: Equip de xarxa de comunicacions que permet comunicar equips entre ells. 38
concurrent: Execució simultània. 30
contenidor: En aquest text està referit a un espai lògic aïllat de la resta de components del sistema.3
datacenters: Centre de dades i/o càlcul. Instal•lacions especialment dedicades a sistemes d'emmagatzematge de dades, computació i telecomunicacions. (http; //en.wikipedia.org/wiki/Data_center). 6
DHCP: Dynamic Host Configuration Protocol. Protocol de xarxa que permet la configuració automàtica dels equips que es connecten a ella. 41
dimoni: Programa, o servei, resident que s'executa en segon pla sense la intervenció directa de l'usuari. 22
DNS: Domain Name System. Servei d'Internet que tradueix noms de dominis en adreces IP. ..41
enrutador: Dispositiu de xarxa que permet connexió entre xarxes o de nivell 3 OSI. 38
entorn d'aplicació: Veure framework.....24
fair sharing: Els recursos són compartits de forma igualitària entre els demandants. 30
frameworks: Esquema de desenvolupament o implementació (abstracció) que dóna coherència al model de dades utilitzat en una aplicació. (http; //en.wikipedia.org/wiki/Software_framework). 6
granularitat: Nivell de detall dels recursos. 3
host: Computadora.....22
HPC: High Performance Computint. Computació d'altes prestacions. 20
hypervisors: Monitor d'execució de diferents VM en una mateixa computadora. 21
IaaS: Infraestructure as a Service, servei bàsic de computació en el núvol en el que s'ofereixen servies bàsics de computació, virtualitzats generalment, a demanda del client. (http; //en.wikipedia.org/wiki/Cloud_computing) 6
kernel: Aplicació que forma part del sistema operatiu, i que recull les peticions lògiques de l'entorn per al seu procés. 21
loopback: Interfície de xarxa virtual que permet un flux d'informació d'entrada/sortida al mateix dispositiu..... 42
mètrica: Prova per coneixer el rendiment d'un sistema o aplicació..... 12
MPI: model de programació, o interfície, que permet l'execució de programes de forma distribuïda per pas de missatges entre les tasques en execució, tant en memòria com en múltiples processadors35
multicore: Processador amb més d'un nucli d'execució. 58
NFS: Acrònim de Network File System. Es tracta d'una carpeta compartida en xarxa com si fos local. 46
nodes: En aquest text està referit a un element autònom de computació..... 7
open source: Programari realitzat i distribuït de forma lliure..... 21
packages: Fitxers i informació encapsulada que dóna solució a una aplicació concreta. 38
Programació per lots: Execució de programes o aplicacions sense la supervisió directa de l'usuari. 25
resilients: Capacitat per a afrontar amb èxit una situació desfavorable o de risc, i per a recuperar-se, adaptar-se i desenvolupar-se positivament davant les circumstàncies adverses. 3
scheduler: Planificador, organitza les execucions de tasques. 32
script: Programa format per una seqüencia de comandes, que és interpretat per la consola d'execució del sistema. 58
segment de xarxa: Part d'una xarxa de computadors, on poden comunicar-se entre ells al tenir el mateix rang d'IP i màscara..... 38
sockets: Mètode de comunicació en xarxa entre una aplicació client i una de servidora. 22
SSH: Acrònim de Secure SHell, o execució remota segura de consola d'usuari. 46
strict priority: Existeix un ordre de prioritat, que s'executa de forma seqüencial; l'entorn d'aplicació amb més prioritat seria el triat per oferir-li els

9. Índex de figures i taules.

Índex de figures

Índex de taules

9. Bibliografia i referències

1. Documentació introductòria de la plataforma MESOS
www.mesos.apache.org

2. Documentació introductòria de la plataforma MESOSPHERE
www.mesosphere.com

3. *Gestió i desenvolupament de projectes. Conceptes i suggeriments*.
Autor: Alfons Bataller Díaz
UOC. P08/19018/00444

4. The Datacenter as a Computer An Introduction to the Design of Warehouse-Scale Machines.
Autors: Luiz André Barroso and Urs Hölzle. 2009.
ISBN: 9781598295573 ebook
http://ieeexplore.ieee.org/xpl/ebooks/bookPdfWithBanner.jsp?fileName=6813235.pdf&bkn=6813234&pdfType=book

5. Datacenter Computing with Apache Mesos - BigData DC
Paco Nathan
http://www.slideshare.net/pacoid/datacenter-computing-with-apache-mesos

6. Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center
Autors: Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica
http://mesos.berkeley.edu/mesos_tech_report.pdf

7. Introduction to Apache Mesos (Slides)
Benjamin Hindman Presented August 20, 2013 at NYC Mesos Meetup
https://speakerdeck.com/benh/apache-mesos-nyc-meetup

8. Run your data center like Google's with Apache Mesos (Video + Demo)
Abhishek Parolkar Presented November 14th, 2013 at Cloud Expo Asia 2013
https://www.youtube.com/watch?v=2YWVGMuMTrg

9. Mesos: A Platform for Fine-Grained Resource Sharing in Datacenters (Video)
Matei Zaharia Presented March 2011 at UC Berkeley
http://www.youtube.com/watch?v=dB8IDu7g9Nc

10. Chronos: A distributed, fault-tolerant and highly available job orchestration framework for Mesos (Slides)
Florian Leibert Presented August 20, 2013 at NYC Mesos Meetup
https://speakerdeck.com/mesos/chronos-august-2013-nyc-meetup

11. Building and Running Distributed Systems using Apache Mesos
Benjamin Hindman , publicat el 25/04/2014 a ApacheCon North America 2014
https://www.youtube.com/watch?v=hTcZGODnyf0

12. Spark on Mesos
Paco Nathan, publicat en #Mesoscon 2014
https://www.youtube.com/watch?v=h-21KA3tZX0

13. Run Apache Spark on Apache Mesos

Paco Nathan, publicat el 15/11/2013
http://youtu.be/KVWMhIeKM_A

14. Multi-agent Cluster Scheduling for Scalability and Flexibility
Andrew Konwinski
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-273.pdf

15. Guia d'exercicis basats en la plataforma MESOS a la Berkeley - Universitat de California
http://ampcamp.berkeley.edu/3/exercises/mesos.html

16. Resolució de problemes a clúster MESOS
https://mail-archives.apache.org/mod_mbox/incubator-mesos-dev/201303.mbox/%3CCAAkWvAy7wGeaBLJSY3--KdFBJzodgMRUznUURCkDYvAU_gwBFg@mail.gmail.com%3E

17. Ubuntu 12.04 Server Guide
https://help.ubuntu.com/12.04/serverguide/serverguide.pdf

18. Ubuntu Server Guide!
https://wiki.ubuntu.com/DocumentationTeam
https://launchpad.net/~ubuntu-core-doc
https://launchpad.net/~ubuntu-server
https://help.ubuntu.com/community/
https://code.launchpad.net/serverguide
https://code.launchpad.net/ubuntu-docs

19. UNIX. Serie práctica.
Steve Moritsugu y DTR Bussiness Systems Inc.
2000. Editorial Prentice Hall. ISBN 84-205-2950-8

20. LINUX. Serie práctica.
M. Drew Streib, Michael Turner, et al.
2004. Editorial Pearson/Prentice Hall. ISBN 84-205-2951-6

21. Parallel programming in C with MPI and OpenMP
Michael J. Quinn
McGraw-Hill, ISBN 007-282256-2

22. Instal·lació de NPB 3.3 a UBUNTU
http://kenshin579.tistory.com/entry/Installation-and-Running-NPB-33-on-Ubuntu-904

23. Procesamiento Paralelo en Redes Linux Utilizando MPI
Projecte de final d'estudis. Alumne: Vicente F. Reyes Puerta
http://www.redes-linux.com/manuales/cluster/mpi-spanish.pdf

24. MPICH2 Installer's Guide
Mathematics and Computer Science Division. Argonne National Laboratory
http://www.mpich.org/static/downloads/1.2.1/mpich2-1.2.1-installguide.pdf

25. MPICH2 User's Guide
Mathematics and Computer Science Division. Argonne National Laboratory
http://www.mpich.org/static/downloads/1.2.1/mpich2-1.2.1-userguide.pdf

26. Repositori d'informació MPI a l'ANL.
http://wiki.mpich.org/mpich/index.php/Frequently_Asked_Questions

27. Tutorial on MPI: The Message Passing Interface
Mathematics and Computer Science Division. Argonne National Laboratory
http://www.idi.ntnu.no/~elster/tdt4200-f09/gropp-mpi-tutorial.pdf

28. Instal·lació de clúster MPI en Ubuntu
https://help.ubuntu.com/community/MpichCluster

29. Resolució de problema MPD amb llibreries PYTHON
http://ubuntuforums.org/archive/index.php/t-1016984.html

30. Exemple de funcionament de clúster a l'EPS Universitat d'Alacant.
http://blogs.ua.es/labseps/2014/02/20/instalacion-configuracion-y-prueba-de-mpich2-version-1-0-8-en-los-laboratorios-de-la-eps/

31. Exemples MPI del laboratori de paral·lelisme de la Universitat del País Basc.
http://www.sc.ehu.es/acwarfra/arpar/LP/LP.fitxategiak/EjclaseMPI.pdf

32. User's Guide for MPE: Extensions for MPI Programs
Anthony Chan, William Gropp, and Ewing Lusk
ftp://ftp.mcs.anl.gov/pub/mpi/mpeman.pdf

33. MPPTEST - Measuring MPI Performance
http://www.mcs.anl.gov/research/projects/mpi/mpptest/

34. Instrumenting MPI Programs with MPE
http://beige.ucs.indiana.edu/I590/node112.html

35. NAS Parallel Benchmarks
http://www.nas.nasa.gov/publications/npb.html

36. The NAS Parallel Benchmarks. Technical report.
http://www.nas.nasa.gov/assets/pdf/techreports/1994/rnr-94-007.pdf

37. Installation and Running NPB-3.3 on Ubuntu 9.04
http://kenshin579.tistory.com/entry/Installation-and-Running-NPB-33-on-Ubuntu-904

38. NPB3.3-MZ (Multi-Zone) MPI+OpenMP Versions
http://www.eneagrid.enea.it/tutorial/Scalasca/NPB3.3-MZ/NPB3.3-MZ-MPI/README.install

39. Installing and Running Gnuplot on Ubuntu
http://math65740.blogspot.com.es/2012/09/installing-and-running-gnuplot-on-ubuntu.html

40. GNUPLOT 4.2 - A Brief Manual and Tutorial
http://people.duke.edu/~hpgavin/gnuplot.html

41. Omega: flexible, scalable schedulers for large compute clusters
Malte Schwarzkopf, Andy Konwinski, Michael Abd-El-Malek, John Wilkes
http://www.e-wilkes.com/john/papers/2013-EuroSys-Omega.pdf

42. What is Docker and why is it so darn popular?
Steven J. Vaughan-Nichols
http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

43. Understanding Docker
https://docs.docker.com/introduction/understanding-docker/

Referències

[1] apache.mesos.org
[2] https://docs.docker.com/introduction/understanding-docker/
[3] http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/
[4] http://en.wikipedia.org/wiki/UnionFS
[5] http://www.berkeley.edu/index.html
[6] *Mesos: A platform for fine-grained resource sharing in the data center*. Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica.
[7] http://mesos.apache.org/community/
[8] http://mesos.apache.org/documentation/latest/mesos-frameworks/
[9] http://cassandra.apache.org/
[10] https://blog.twitter.com/2012/incubating-apache-mesos
[11] http://nerds.airbnb.com/hadoop-on-mesos/
[12] http://mesos.apache.org/documentation/latest/high-availability/
[13] https://help.ubuntu.com/lts/serverguide/lxc.html
[14] "Parallel programming in C with MPI and OpenMP", Michael J. Quinn, McGraw Hill, ISBN 007-282256-2
[15] http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf
[16] http://www.anl.gov/
[17] https://github.com/apache/mesos/tree/master/mpi
[18] Descarregar versió de MPICH2.1.2: http://www.mpich.org/static/downloads/1.2.1p1/mpich2-1.2.1p1.tar.gz
[19] Manual d'instal·lació de MPICH2.1.2: http://www.mpich.org/static/downloads/1.2.1/mpich2-1.2.1-installguide.pdf
[20] Instal·lació d'un clúster MPICH a Ubuntu: https://help.ubuntu.com/community/MpichCluster
[21] http://ca.wikipedia.org/wiki/Secure_Shell
[22] http://ca.wikipedia.org/wiki/Network_File_System
[23] http://www.mpich.org/static/downloads/1.2.1/mpich2-1.2.1-installguide.pdf
[24] http://www.mpich.org/static/docs/v3.1/www1/mpicc.html
[25] http://www.mpich.org/static/docs/v3.1/www1/mpiexec.html
[26] https://mesosphere.com/
[27] ftp://ftp.mcs.anl.gov/pub/mpi/tools/perftest.tar.gz
[28] El detall és pot estudiar al codi $HOME/mesos.0.20.0/mpi/mpiexec.mesos.py
[29] ftp://ftp.mcs.anl.gov/pub/mpi/mpeman.pdf
[30] http://www.bsc.es/sites/default/files/public/about/news/montblanc-27112013-wired.pdf
[31] http://www.montblanc-project.eu/
[32] http://support.asus.com/download.aspx?SLanguage=en&m=A8N-VM%20CSM&os=17
[33] http://manpages.ubuntu.com/manpages/lucid/es/man8/ping.8.html
[34] http://stackoverflow.com/questions/26373738/apache-mesos-slave-cannot-connect-to-master
[35] https://mail-archives.apache.org/mod_mbox/incubator-mesos-dev/201303.mbox/%3CCAAkWvAy7wGeaBLJSY3--KdFBJzodgMRUznUURCkDYvAU_gwBFg@mail.gmail.com%3E
[36] http://mesos.apache.org/documentation/latest/deploy-scripts/
[37] http://zookeeper.apache.org/
[38] https://github.com/airbnb/chronos
[39] http://mesos.apache.org/documentation/latest/network-monitoring/
[40] http://mesos.apache.org/documentation/latest/app-framework-development-guide/
[41] http://mesos.apache.org/documentation/latest/tools/

A. Annex

A.1 Fitxers rellevants MESOS MPI

A.1.1 Fitxer mpiexec-mesos

$HOME/mesos.0.20.0/build/mpi/

```
#!/bin/sh
#
# This is a wrapper script for the MPI launcher framework.
#
# This script uses MESOS_SOURCE_DIR and MESOS_BUILD_DIR which come
# from configuration substitutions.
MESOS_SOURCE_DIR=/home/master/mesos-0.20.0/build/..
MESOS_BUILD_DIR=/home/master/mesos-0.20.0/build

# Use colors for errors.
. ${MESOS_SOURCE_DIR}/support/colors.sh

# Force the use of the Python interpreter configured during building.
test ! -z "${PYTHON}" && \
  echo "${RED}Ignoring PYTHON environment variable (using /usr/bin/python)${NORMAL}"

PYTHON=/usr/bin/python

DISTRIBUTE_EGG=`echo ${MESOS_BUILD_DIR}/3rdparty/distribute-*/dist/*.egg`

test ! -e ${DISTRIBUTE_EGG} && \
  echo "${RED}Failed to find ${DISTRIBUTE_EGG} verify PYTHONPATH${NORMAL}"

PROTOBUF=${MESOS_BUILD_DIR}/3rdparty/libprocess/3rdparty/protobuf-*
PROTOBUF_EGG=`echo ${PROTOBUF}/python/dist/protobuf*.egg`

# Just warn in the case when build with --disable-bundled.
test ! -e ${PROTOBUF_EGG} && \
  echo "${RED}Failed to find ${PROTOBUF_EGG} check your PYTHONPATH${NORMAL}"

MESOS_EGGS=""
for egg in interface native; do
  base_dir="${MESOS_BUILD_DIR}/src/python/${egg}/dist/"
  egg_path="${base_dir}mesos.${egg}-0.20.0"

  if [[ ${egg} == "native" ]]; then
    egg_path+="-py2.7-linux-x86_64"
  else
    egg_path+="-py2.7"
  fi

  egg_path+=".egg"

  test ! -e ${egg_path} && \
    echo "${RED}Failed to find ${egg_path}${NORMAL}" && \
    exit 1

  MESOS_EGGS+="${egg_path}:"
done

SCRIPT=${MESOS_SOURCE_DIR}/mpi/mpiexec-mesos.py

test ! -e ${SCRIPT} && \
  echo "${RED}Failed to find ${SCRIPT}${NORMAL}" && \
  exit 1

PYTHONPATH="${DISTRIBUTE_EGG}:${PROTOBUF_EGG}:${MESOS_EGGS}" \
  exec ${PYTHON} ${SCRIPT} "${@}"
```

A.1.2 Fitxer mpiexec-mesos.py

$HOME/mesos.0.20.0/mpi/

S'han marcat en negreta les parts més importants en la crida mpixec i gestió de dimonis MPD.

```
#!/usr/bin/env python

import mesos.interface
import mesos.native
from mesos.interface import mesos_pb2
import os
import sys
import time
import re
import threading

from optparse import OptionParser
from subprocess import *


def mpiexec():
  print "We've launched all our MPDs; waiting for them to come up"

  while countMPDs() <= TOTAL_MPDS:
    print "...waiting on MPD(s)..."
    time.sleep(1)
  print "Got %d mpd(s), running mpiexec" % TOTAL_MPDS

  try:
    print "Running mpiexec"
    call([MPICH2PATH + 'mpiexec', '-1', '-n', str(TOTAL_MPDS)] + MPI_PROGRAM)


  except OSError,e:
    print >> sys.stderr, "Error executing mpiexec"
    print >> sys.stderr, e
    exit(2)

  print "mpiexec completed, calling mpdallexit %s" % MPD_PID

  # Ring/slave mpd daemons will be killed on executor's shutdown() if
  # framework scheduler fails to call 'mpdallexit'.
  call([MPICH2PATH + 'mpdallexit', MPD_PID])


class MPIScheduler(mesos.interface.Scheduler):

  def __init__(self, options, ip, port):
    self.mpdsLaunched = 0
    self.mpdsFinished = 0
    self.ip = ip
    self.port = port
    self.options = options
    self.startedExec = False

  def registered(self, driver, fid, masterInfo):
    print "Mesos MPI scheduler and mpd running at %s:%s" % (self.ip, self.port)
    print "Registered with framework ID %s" % fid.value

  def resourceOffers(self, driver, offers):
    print "Got %d resource offers" % len(offers)

    for offer in offers:
      print "Considering resource offer %s from %s" % (offer.id.value, offer.hostname)

      if self.mpdsLaunched == TOTAL_MPDS:
        print "Declining permanently because we have already launched enough tasks"
        driver.declineOffer(offer.id)
        continue

      cpus = 0
```

```
      mem = 0
      tasks = []

      for resource in offer.resources:
        if resource.name == "cpus":
          cpus = resource.scalar.value
        elif resource.name == "mem":
          mem = resource.scalar.value

      if cpus < CPUS or mem < MEM:
        print "Declining offer due to too few resources"
        driver.declineOffer(offer.id)
      else:
        tid = self.mpdsLaunched
        self.mpdsLaunched += 1

        print "Accepting offer on %s to start mpd %d" % (offer.hostname, tid)

        task = mesos_pb2.TaskInfo()
        task.task_id.value = str(tid)
        task.slave_id.value = offer.slave_id.value
        task.name = "task %d " % tid

        cpus = task.resources.add()
        cpus.name = "cpus"
        cpus.type = mesos_pb2.Value.SCALAR
        cpus.scalar.value = CPUS

        mem = task.resources.add()
        mem.name = "mem"
        mem.type = mesos_pb2.Value.SCALAR
        mem.scalar.value = MEM

        task.command.value = "%smpd --noconsole --ncpus=%d --host=%s --port=%s" % (MPICH2PATH, CPUS,
self.ip, self.port)

        tasks.append(task)

        print "Replying to offer: launching mpd %d on host %s" % (tid, offer.hostname)
        driver.launchTasks(offer.id, tasks)

    if not self.startedExec and self.mpdsLaunched == TOTAL_MPDS:
      threading.Thread(target = mpiexec).start()
      self.startedExec = True

  def statusUpdate(self, driver, update):
    print "Task %s in state %s" % (update.task_id.value, update.state)
    if (update.state == mesos_pb2.TASK_FAILED or
        update.state == mesos_pb2.TASK_KILLED or
        update.state == mesos_pb2.TASK_LOST):
      print "A task finished unexpectedly, calling mpdexit on %s" % MPD_PID
      call([MPICH2PATH + "mpdexit", MPD_PID])
      driver.stop()
    if (update.state == mesos_pb2.TASK_FINISHED):
      self.mpdsFinished += 1
      if self.mpdsFinished == TOTAL_MPDS:
        print "All tasks done, all mpd's closed, exiting"
        driver.stop()


def countMPDs():
  try:
    mpdtraceproc = Popen(MPICH2PATH + "mpdtrace -l", shell=True, stdout=PIPE)
    mpdtraceline = mpdtraceproc.communicate()[0]
    return mpdtraceline.count("\n")
  except OSError,e:
    print >>sys.stderr, "Error starting mpd or mpdtrace"
    print >>sys.stderr, e
    exit(2)
```

```
def parseIpPort(s):
  ba = re.search("([^_]*)_([0-9]*)", s)
  ip = ba.group(1)
  port = ba.group(2)
  return (ip, port)


if __name__ == "__main__":
  parser = OptionParser(usage="Usage: %prog [options] mesos_master mpi_program")
  parser.disable_interspersed_args()
  parser.add_option("-n", "--num",
                    help="number of mpd's to allocate (default 1)",
                    dest="num", type="int", default=1)
  parser.add_option("-c", "--cpus",
                    help="number of cpus per mpd (default 1)",
                    dest="cpus", type="int", default=1)
  parser.add_option("-m","--mem",
                    help="number of MB of memory per mpd (default 1GB)",
                    dest="mem", type="int", default=1024)
  parser.add_option("--name",
                    help="framework name", dest="name", type="string")
  parser.add_option("-p","--path",
                    help="path to look for MPICH2 binaries (mpd, mpiexec, etc.)",
                    dest="path", type="string", default="")
  parser.add_option("--ifhn-master",
                    help="alt. interface hostname for what mpd is running on (for scheduler)",
                    dest="ifhn_master", type="string")

  # Add options to configure cpus and mem.
  (options,args) = parser.parse_args()
  if len(args) < 2:
    print >> sys.stderr, "At least two parameters required."
    print >> sys.stderr, "Use --help to show usage."
    exit(2)

  TOTAL_MPDS = options.num
  CPUS = options.cpus
  MEM = options.mem
  MPI_PROGRAM = args[1:]


 # Give options.path a trailing '/', if it doesn't have one already.
  MPICH2PATH = os.path.join(options.path, "")

  print "Connecting to Mesos master %s" % args[0]

  try:
    mpd_cmd = MPICH2PATH + "mpd"
    mpdtrace_cmd = MPICH2PATH + "mpdtrace -l"

    if options.ifhn_master is not None:
      call([mpd_cmd, "--daemon", "--ifhn=" + options.ifhn_master])
    else:
      call([mpd_cmd, "--daemon"])

    mpdtraceproc = Popen(mpdtrace_cmd, shell=True, stdout=PIPE)
    mpdtraceout = mpdtraceproc.communicate()[0]

  except OSError,e:
    print >> sys.stderr, "Error starting mpd or mpdtrace"
    print >> sys.stderr, e
    exit(2)

  (ip,port) = parseIpPort(mpdtraceout)

  MPD_PID = mpdtraceout.split(" ")[0]
  print "MPD_PID is %s" % MPD_PID

  scheduler = MPIScheduler(options, ip, port)

  framework = mesos_pb2.FrameworkInfo()
  framework.user = ""
```

```
if options.name is not None:
  framework.name = options.name
else:
  framework.name = "MPI: %s" % MPI_PROGRAM[0]

driver = mesos.native.MesosSchedulerDriver(
  scheduler,
  framework,
  args[0])
sys.exit(0 if driver.run() == mesos_pb2.DRIVER_STOPPED else 1)
```

A.2 Fitxers de mètriques. Execució PERFTEST-1.5. Clúster MESOS

A.2.1 Fitxer mpptest_mesos.gpl

#p0	p1	dist	len	ave time (us)	rate
0	2	2	0	102.133751	0.00
0	2	2	32	102.648735	311.743e+3
0	2	2	64	104.677677	611.401e+3
0	2	2	96	110.068321	872.186e+3
0	2	2	128	115.616322	1.107e+6
0	2	2	160	121.107101	1.321e+6
0	2	2	192	126.409531	1.519e+6
0	2	2	224	131.983757	1.697e+6
0	2	2	256	137.705803	1.859e+6
0	2	2	288	143.055916	2.013e+6
0	2	2	320	148.525238	2.155e+6
0	2	2	352	154.583454	2.277e+6
0	2	2	384	158.584118	2.421e+6
0	2	2	416	164.885521	2.523e+6
0	2	2	448	170.228481	2.632e+6
0	2	2	480	175.328255	2.738e+6
0	2	2	512	180.296898	2.840e+6
0	2	2	544	186.274052	2.920e+6
0	2	2	576	192.034245	2.999e+6
0	2	2	608	197.072029	3.085e+6
0	2	2	640	201.992989	3.168e+6
0	2	2	672	207.905769	3.232e+6
0	2	2	704	213.313103	3.300e+6
0	2	2	736	218.775272	3.364e+6
0	2	2	768	223.853588	3.431e+6
0	2	2	800	228.774548	3.497e+6
0	2	2	832	234.549046	3.547e+6
0	2	2	864	239.491463	3.608e+6
0	2	2	896	245.254040	3.653e+6
0	2	2	928	251.395702	3.691e+6
0	2	2	960	256.295204	3.746e+6
0	2	2	992	267.148018	3.713e+6
0	2	2	1024	271.985531	3.765e+6
0	2	2	1056	277.743340	3.802e+6
0	2	2	1088	282.776356	3.848e+6
0	2	2	1120	288.684368	3.880e+6
0	2	2	1152	294.215679	3.915e+6
0	2	2	1184	300.228596	3.944e+6
0	2	2	1216	306.141376	3.972e+6
0	2	2	1248	311.951637	4.001e+6
0	2	2	1280	312.206745	4.100e+6
0	2	2	1312	322.468281	4.069e+6
0	2	2	1344	411.462784	3.266e+6
0	2	2	1376	411.283970	3.346e+6
0	2	2	1408	411.846638	3.419e+6
0	2	2	1440	359.942913	4.001e+6
0	2	2	1472	357.296467	4.120e+6
0	2	2	1504	357.162952	4.211e+6
0	2	2	1536	358.860493	4.280e+6
0	2	2	1568	360.510349	4.349e+6
0	2	2	1600	361.604691	4.425e+6
0	2	2	1632	362.832546	4.498e+6
0	2	2	1664	411.181450	4.047e+6
0	2	2	1696	457.055569	3.711e+6
0	2	2	1728	459.651947	3.759e+6
0	2	2	1760	460.526943	3.822e+6
0	2	2	1792	460.860729	3.888e+6
0	2	2	1824	463.833809	3.932e+6
0	2	2	1856	465.574265	3.986e+6
0	2	2	1888	466.585159	4.046e+6
0	2	2	1920	467.567444	4.106e+6
0	2	2	1952	469.279289	4.160e+6
0	2	2	1984	471.878052	4.204e+6
0	2	2	2016	471.818447	4.273e+6
0	2	2	2048	474.007130	4.321e+6
0	2	2	2080	473.289490	4.395e+6
0	2	2	2112	474.250317	4.453e+6
0	2	2	2144	476.269722	4.502e+6
0	2	2	2176	478.327274	4.549e+6
0	2	2	2208	479.280949	4.607e+6
0	2	2	2240	481.646061	4.651e+6
0	2	2	2272	483.510494	4.699e+6

```
0       2       2       2304    484.688282      4.754e+6
0       2       2       2336    484.237671      4.824e+6
0       2       2       2368    486.192703      4.870e+6
0       2       2       2400    486.857891      4.930e+6
0       2       2       2432    488.238335      4.981e+6
0       2       2       2464    489.480495      5.034e+6
0       2       2       2496    491.518974      5.078e+6
0       2       2       2528    492.606163      5.132e+6
0       2       2       2560    494.041443      5.182e+6
0       2       2       2592    495.796204      5.228e+6
0       2       2       2624    495.774746      5.293e+6
0       2       2       2656    499.770641      5.314e+6
0       2       2       2688    498.785973      5.389e+6
0       2       2       2720    499.916077      5.441e+6
0       2       2       2752    501.685143      5.486e+6
0       2       2       2784    502.278805      5.543e+6
0       2       2       2816    504.212379      5.585e+6
0       2       2       2848    506.100655      5.627e+6
0       2       2       2880    514.657497      5.596e+6
0       2       2       2912    514.738560      5.657e+6
0       2       2       2944    519.230366      5.670e+6
0       2       2       2976    517.489910      5.751e+6
0       2       2       3008    518.417358      5.802e+6
0       2       2       3040    519.053936      5.857e+6
0       2       2       3072    527.038574      5.829e+6
0       2       2       3104    541.112423      5.736e+6
0       2       2       3136    552.697182      5.674e+6
0       2       2       3168    561.769009      5.639e+6
0       2       2       3200    565.559864      5.658e+6
0       2       2       3232    555.374622      5.819e+6
0       2       2       3264    562.589169      5.802e+6
0       2       2       3296    569.932461      5.783e+6
0       2       2       3328    565.862656      5.881e+6
0       2       2       3360    558.774471      6.013e+6
0       2       2       3392    556.814671      6.092e+6
0       2       2       3424    563.945770      6.072e+6
0       2       2       3456    575.201511      6.008e+6
0       2       2       3488    567.722321      6.144e+6
0       2       2       3520    568.189621      6.195e+6
0       2       2       3552    573.728085      6.191e+6
0       2       2       3584    576.050282      6.222e+6
0       2       2       3616    571.761131      6.324e+6
0       2       2       3648    580.432415      6.285e+6
0       2       2       3680    581.519604      6.328e+6
0       2       2       3712    578.863621      6.413e+6
0       2       2       3744    582.201481      6.431e+6
0       2       2       3776    578.603745      6.526e+6
0       2       2       3808    578.045845      6.588e+6
0       2       2       3840    590.376854      6.504e+6
0       2       2       3872    600.731373      6.445e+6
0       2       2       3904    595.772266      6.553e+6
0       2       2       3936    641.362667      6.137e+6
0       2       2       3968    643.944740      6.162e+6
0       2       2       4000    646.891594      6.183e+6
0       2       2       4032    650.174618      6.201e+6
0       2       2       4064    648.100376      6.271e+6
0       2       2       4096    649.354458      6.308e+6
```

A.2.2 Fitxer async_mesos.gpl

```
#p0     p1      dist    len     ave time (us)   rate
0       2       2       0       102.241039      0.00
0       2       2       32      102.725029      311.511e+3
0       2       2       64      104.541779      612.195e+3
0       2       2       96      110.018253      872.582e+3
0       2       2       128     115.852356      1.105e+6
0       2       2       160     121.552944      1.316e+6
0       2       2       192     127.022266      1.512e+6
0       2       2       224     132.465363      1.691e+6
0       2       2       256     138.757229      1.845e+6
0       2       2       288     143.666267      2.005e+6
0       2       2       320     149.881840      2.135e+6
0       2       2       352     154.368877      2.280e+6
0       2       2       384     159.687996      2.405e+6
0       2       2       416     165.429115      2.515e+6
0       2       2       448     170.378685      2.629e+6
0       2       2       480     175.924301      2.728e+6
0       2       2       512     180.873871      2.831e+6
0       2       2       544     186.235905      2.921e+6
0       2       2       576     191.838741      3.003e+6
0       2       2       608     197.095871      3.085e+6
0       2       2       640     204.105377      3.136e+6
0       2       2       672     208.594799      3.222e+6
0       2       2       704     215.210915      3.271e+6
0       2       2       736     219.020844      3.360e+6
0       2       2       768     224.125385      3.427e+6
0       2       2       800     229.041576      3.493e+6
0       2       2       832     235.428810      3.534e+6
0       2       2       864     240.595341      3.591e+6
0       2       2       896     245.783329      3.645e+6
0       2       2       928     251.204967      3.694e+6
0       2       2       960     256.612301      3.741e+6
0       2       2       992     267.982483      3.702e+6
0       2       2       1024    272.588730      3.757e+6
0       2       2       1056    278.553963      3.791e+6
0       2       2       1088    284.199715      3.828e+6
```

0	2	2	1120	289.781094	3.865e+6
0	2	2	1152	294.694901	3.909e+6
0	2	2	1184	301.668644	3.925e+6
0	2	2	1216	307.574272	3.954e+6
0	2	2	1248	312.461853	3.994e+6
0	2	2	1280	312.566757	4.095e+6
0	2	2	1312	342.023373	3.836e+6
0	2	2	1344	412.838459	3.256e+6
0	2	2	1376	413.467884	3.328e+6
0	2	2	1408	412.247181	3.415e+6
0	2	2	1440	363.755226	3.959e+6
0	2	2	1472	359.342098	4.096e+6
0	2	2	1504	359.818935	4.180e+6
0	2	2	1536	360.496044	4.261e+6
0	2	2	1568	364.499092	4.302e+6
0	2	2	1600	362.374783	4.415e+6
0	2	2	1632	363.247395	4.493e+6
0	2	2	1664	414.988995	4.010e+6
0	2	2	1696	461.266041	3.677e+6
0	2	2	1728	460.140705	3.755e+6
0	2	2	1760	462.470055	3.806e+6
0	2	2	1792	464.894772	3.855e+6
0	2	2	1824	462.627411	3.943e+6
0	2	2	1856	466.258526	3.981e+6
0	2	2	1888	467.631817	4.037e+6
0	2	2	1920	469.040871	4.093e+6
0	2	2	1952	470.528603	4.149e+6
0	2	2	1984	473.699570	4.188e+6
0	2	2	2016	473.055840	4.262e+6
0	2	2	2048	474.207401	4.319e+6
0	2	2	2080	474.715233	4.382e+6
0	2	2	2112	478.043556	4.418e+6
0	2	2	2144	478.723049	4.479e+6
0	2	2	2176	481.615067	4.518e+6
0	2	2	2208	481.827259	4.583e+6
0	2	2	2240	484.027863	4.628e+6
0	2	2	2272	483.014584	4.704e+6
0	2	2	2304	484.323502	4.757e+6
0	2	2	2336	485.775471	4.809e+6
0	2	2	2368	488.531590	4.847e+6
0	2	2	2400	489.573479	4.902e+6
0	2	2	2432	489.506721	4.968e+6
0	2	2	2464	491.075516	5.018e+6
0	2	2	2496	493.774414	5.055e+6
0	2	2	2528	496.346951	5.093e+6
0	2	2	2560	495.617390	5.165e+6
0	2	2	2592	495.820045	5.228e+6
0	2	2	2624	498.850346	5.260e+6
0	2	2	2656	499.556065	5.317e+6
0	2	2	2688	500.767231	5.368e+6
0	2	2	2720	501.570702	5.423e+6
0	2	2	2752	502.648354	5.475e+6
0	2	2	2784	504.236221	5.521e+6
0	2	2	2816	507.504940	5.549e+6
0	2	2	2848	508.358479	5.602e+6
0	2	2	2880	517.754555	5.562e+6
0	2	2	2912	516.703129	5.636e+6
0	2	2	2944	519.478321	5.667e+6
0	2	2	2976	519.483089	5.729e+6
0	2	2	3008	520.594120	5.778e+6
0	2	2	3040	524.079800	5.801e+6
0	2	2	3072	528.323650	5.815e+6
0	2	2	3104	546.891689	5.676e+6
0	2	2	3136	562.701225	5.573e+6
0	2	2	3168	568.966866	5.568e+6
0	2	2	3200	566.391945	5.650e+6
0	2	2	3232	569.603443	5.674e+6
0	2	2	3264	568.931103	5.737e+6
0	2	2	3296	574.610233	5.736e+6
0	2	2	3328	570.168495	5.837e+6
0	2	2	3360	569.336414	5.902e+6
0	2	2	3392	575.635433	5.893e+6
0	2	2	3424	570.950508	5.997e+6
0	2	2	3456	571.725368	6.045e+6
0	2	2	3488	571.622849	6.102e+6
0	2	2	3520	567.941666	6.198e+6
0	2	2	3552	569.136143	6.241e+6
0	2	2	3584	565.462112	6.338e+6
0	2	2	3616	570.933819	6.333e+6
0	2	2	3648	587.875843	6.205e+6
0	2	2	3680	580.005646	6.345e+6
0	2	2	3712	586.497784	6.329e+6
0	2	2	3744	586.221218	6.387e+6
0	2	2	3776	579.092503	6.521e+6
0	2	2	3808	588.502884	6.471e+6
0	2	2	3840	590.913296	6.498e+6
0	2	2	3872	584.855080	6.620e+6
0	2	2	3904	599.305630	6.514e+6
0	2	2	3936	642.395020	6.127e+6
0	2	2	3968	648.455620	6.119e+6
0	2	2	4000	651.206970	6.142e+6
0	2	2	4032	651.018620	6.193e+6
0	2	2	4064	652.861595	6.225e+6
0	2	2	4096	652.587414	6.277e+6

A.2.3 Fitxer bisect_mesos.gpl

#p0	p1	dist	len	ave time (us)	rate
0	2	2	0	102.241039	0.00
0	2	2	32	102.725029	311.511e+3
0	2	2	64	104.541779	612.195e+3
0	2	2	96	110.018253	872.582e+3
0	2	2	128	115.852356	1.105e+6
0	2	2	160	121.552944	1.316e+6
0	2	2	192	127.022266	1.512e+6
0	2	2	224	132.465363	1.691e+6
0	2	2	256	138.757229	1.845e+6
0	2	2	288	143.666267	2.005e+6
0	2	2	320	149.881840	2.135e+6
0	2	2	352	154.368877	2.280e+6
0	2	2	384	159.687996	2.405e+6
0	2	2	416	165.429115	2.515e+6
0	2	2	448	170.378685	2.629e+6
0	2	2	480	175.924301	2.728e+6
0	2	2	512	180.873871	2.831e+6
0	2	2	544	186.235905	2.921e+6
0	2	2	576	191.838741	3.003e+6
0	2	2	608	197.095871	3.085e+6
0	2	2	640	204.105377	3.136e+6
0	2	2	672	208.594799	3.222e+6
0	2	2	704	215.210915	3.271e+6
0	2	2	736	219.020844	3.360e+6
0	2	2	768	224.125385	3.427e+6
0	2	2	800	229.041576	3.493e+6
0	2	2	832	235.428810	3.534e+6
0	2	2	864	240.595341	3.591e+6
0	2	2	896	245.783329	3.645e+6
0	2	2	928	251.204967	3.694e+6
0	2	2	960	256.612301	3.741e+6
0	2	2	992	267.982483	3.702e+6
0	2	2	1024	272.588730	3.757e+6
0	2	2	1056	278.553963	3.791e+6
0	2	2	1088	284.199715	3.828e+6
0	2	2	1120	289.781094	3.865e+6
0	2	2	1152	294.694901	3.909e+6
0	2	2	1184	301.668644	3.925e+6
0	2	2	1216	307.574272	3.954e+6
0	2	2	1248	312.461853	3.994e+6
0	2	2	1280	312.566757	4.095e+6
0	2	2	1312	342.023373	3.836e+6
0	2	2	1344	412.838459	3.256e+6
0	2	2	1376	413.467884	3.328e+6
0	2	2	1408	412.247181	3.415e+6
0	2	2	1440	363.755226	3.959e+6
0	2	2	1472	359.342098	4.096e+6
0	2	2	1504	359.818935	4.180e+6
0	2	2	1536	360.496044	4.261e+6
0	2	2	1568	364.499092	4.302e+6
0	2	2	1600	362.374783	4.415e+6
0	2	2	1632	363.247395	4.493e+6
0	2	2	1664	414.988995	4.010e+6
0	2	2	1696	461.266041	3.677e+6
0	2	2	1728	460.140705	3.755e+6
0	2	2	1760	462.470055	3.806e+6
0	2	2	1792	464.894772	3.855e+6
0	2	2	1824	462.627411	3.943e+6
0	2	2	1856	466.258526	3.981e+6
0	2	2	1888	467.631817	4.037e+6
0	2	2	1920	469.040871	4.093e+6
0	2	2	1952	470.528603	4.149e+6
0	2	2	1984	473.699570	4.188e+6
0	2	2	2016	473.055840	4.262e+6
0	2	2	2048	474.207401	4.319e+6
0	2	2	2080	474.715233	4.382e+6
0	2	2	2112	478.043556	4.418e+6
0	2	2	2144	478.723049	4.479e+6
0	2	2	2176	481.615067	4.518e+6
0	2	2	2208	481.827259	4.583e+6
0	2	2	2240	484.027863	4.628e+6
0	2	2	2272	483.014584	4.704e+6
0	2	2	2304	484.323502	4.757e+6
0	2	2	2336	485.775471	4.809e+6
0	2	2	2368	488.531590	4.847e+6
0	2	2	2400	489.573479	4.902e+6
0	2	2	2432	489.506721	4.968e+6
0	2	2	2464	491.075516	5.018e+6
0	2	2	2496	493.774414	5.055e+6
0	2	2	2528	496.346951	5.093e+6
0	2	2	2560	495.617390	5.165e+6
0	2	2	2592	495.820045	5.228e+6
0	2	2	2624	498.850346	5.260e+6
0	2	2	2656	499.556065	5.317e+6
0	2	2	2688	500.767231	5.368e+6
0	2	2	2720	501.570702	5.423e+6
0	2	2	2752	502.648354	5.475e+6
0	2	2	2784	504.236221	5.521e+6
0	2	2	2816	507.504940	5.549e+6
0	2	2	2848	508.358479	5.602e+6
0	2	2	2880	517.754555	5.562e+6
0	2	2	2912	516.703129	5.636e+6
0	2	2	2944	519.478321	5.667e+6
0	2	2	2976	519.483089	5.729e+6
0	2	2	3008	520.594120	5.778e+6
0	2	2	3040	524.079800	5.801e+6
0	2	2	3072	528.323650	5.815e+6
0	2	2	3104	546.891689	5.676e+6
0	2	2	3136	562.701225	5.573e+6

0	2	2	3168	568.966866	5.568e+6
0	2	2	3200	566.391945	5.650e+6
0	2	2	3232	569.603443	5.674e+6
0	2	2	3264	568.931103	5.737e+6
0	2	2	3296	574.610233	5.736e+6
0	2	2	3328	570.168495	5.837e+6
0	2	2	3360	569.336414	5.902e+6
0	2	2	3392	575.635433	5.893e+6
0	2	2	3424	570.950508	5.997e+6
0	2	2	3456	571.725368	6.045e+6
0	2	2	3488	571.622849	6.102e+6
0	2	2	3520	567.941666	6.198e+6
0	2	2	3552	569.136143	6.241e+6
0	2	2	3584	565.462112	6.338e+6
0	2	2	3616	570.933819	6.333e+6
0	2	2	3648	587.875843	6.205e+6
0	2	2	3680	580.005646	6.345e+6
0	2	2	3712	586.497784	6.329e+6
0	2	2	3744	586.221218	6.387e+6
0	2	2	3776	579.092503	6.521e+6
0	2	2	3808	588.502884	6.471e+6
0	2	2	3840	590.913296	6.498e+6
0	2	2	3872	584.855080	6.620e+6
0	2	2	3904	599.305630	6.514e+6
0	2	2	3936	642.395020	6.127e+6
0	2	2	3968	648.455620	6.119e+6
0	2	2	4000	651.206970	6.142e+6
0	2	2	4032	651.018620	6.193e+6
0	2	2	4064	652.861595	6.225e+6
0	2	2	4096	652.587414	6.277e+6

A.2.4 Fitxer overlap_mesos.gpl

#p0	p1	dist	len	ave time (us)	rate
0	2	2	0	291.540623	0.00
0	2	2	32	303.146839	105.559e+3
0	2	2	64	295.019150	216.935e+3
0	2	2	96	323.288441	296.948e+3
0	2	2	128	301.210880	424.951e+3
0	2	2	160	293.402672	545.326e+3
0	2	2	192	303.742886	632.114e+3
0	2	2	224	298.478603	750.473e+3
0	2	2	256	301.156044	850.058e+3
0	2	2	288	300.154686	959.505e+3
0	2	2	320	302.138329	1.059e+6
0	2	2	352	282.380581	1.247e+6
0	2	2	384	319.569111	1.202e+6
0	2	2	416	302.460194	1.375e+6
0	2	2	448	321.033001	1.395e+6
0	2	2	480	297.701359	1.612e+6
0	2	2	512	299.642086	1.709e+6
0	2	2	544	296.599865	1.834e+6
0	2	2	576	302.445889	1.904e+6
0	2	2	608	318.491459	1.909e+6
0	2	2	640	300.388336	2.131e+6
0	2	2	672	301.628113	2.228e+6
0	2	2	704	298.626423	2.357e+6
0	2	2	736	297.660828	2.473e+6
0	2	2	768	301.804543	2.545e+6
0	2	2	800	299.365520	2.672e+6
0	2	2	832	295.710564	2.814e+6
0	2	2	864	298.225880	2.897e+6
0	2	2	896	290.815830	3.081e+6
0	2	2	928	299.813747	3.095e+6
0	2	2	960	277.936459	3.454e+6
0	2	2	992	299.870968	3.308e+6
0	2	2	1024	295.724869	3.463e+6
0	2	2	1056	302.252769	3.494e+6
0	2	2	1088	295.526981	3.682e+6
0	2	2	1120	302.329063	3.705e+6
0	2	2	1152	298.836231	3.855e+6
0	2	2	1184	296.494961	3.993e+6
0	2	2	1216	303.108692	4.012e+6
0	2	2	1248	300.817490	4.149e+6
0	2	2	1280	296.912193	4.311e+6
0	2	2	1312	296.278000	4.428e+6
0	2	2	1344	299.546719	4.487e+6
0	2	2	1376	301.706791	4.561e+6
0	2	2	1408	301.811695	4.665e+6
0	2	2	1440	304.858685	4.724e+6
0	2	2	1472	303.666592	4.847e+6
0	2	2	1504	294.106007	5.114e+6
0	2	2	1536	275.208950	5.581e+6
0	2	2	1568	321.850777	4.872e+6
0	2	2	1600	308.225155	5.191e+6
0	2	2	1632	298.359394	5.470e+6
0	2	2	1664	276.479721	6.019e+6
0	2	2	1696	300.264359	5.648e+6
0	2	2	1728	304.403305	5.677e+6
0	2	2	1760	270.256996	6.512e+6
0	2	2	1792	300.281048	5.968e+6
0	2	2	1824	299.758911	6.085e+6
0	2	2	1856	298.395157	6.220e+6
0	2	2	1888	297.999382	6.336e+6
0	2	2	1920	305.464268	6.286e+6
0	2	2	1952	299.954414	6.508e+6
0	2	2	1984	296.127796	6.700e+6
0	2	2	2016	322.015285	6.261e+6
0	2	2	2048	299.000740	6.849e+6
0	2	2	2080	294.990540	7.051e+6

```
0       2       2       2112    298.328400          7.079e+6
0       2       2       2144    301.532745          7.110e+6
0       2       2       2176    322.122574          6.755e+6
0       2       2       2208    300.357342          7.351e+6
0       2       2       2240    301.353931          7.433e+6
0       2       2       2272    316.734314          7.173e+6
0       2       2       2304    298.299789          7.724e+6
0       2       2       2336    297.958851          7.840e+6
0       2       2       2368    298.187733          7.941e+6
0       2       2       2400    301.668644          7.956e+6
0       2       2       2432    319.399834          7.614e+6
0       2       2       2464    306.057930          8.051e+6
0       2       2       2496    298.957825          8.349e+6
0       2       2       2528    303.792953          8.321e+6
0       2       2       2560    278.761387          9.183e+6
0       2       2       2592    301.163197          8.607e+6
0       2       2       2624    297.636986          8.816e+6
0       2       2       2656    301.134586          8.820e+6
0       2       2       2688    298.874378          8.994e+6
0       2       2       2720    319.042206          8.526e+6
0       2       2       2752    301.539898          9.126e+6
0       2       2       2784    300.941467          9.251e+6
0       2       2       2816    301.403999          9.343e+6
0       2       2       2848    300.889015          9.465e+6
0       2       2       2880    301.420689          9.555e+6
0       2       2       2912    299.727917          9.715e+6
0       2       2       2944    277.554989          10.607e+6
0       2       2       2976    300.428867          9.906e+6
0       2       2       3008    300.512314          10.010e+6
0       2       2       3040    300.436020          10.119e+6
0       2       2       3072    298.457146          10.293e+6
0       2       2       3104    298.459530          10.400e+6
0       2       2       3136    302.045345          10.383e+6
0       2       2       3168    301.721096          10.500e+6
0       2       2       3200    294.034481          10.883e+6
0       2       2       3232    301.671028          10.714e+6
0       2       2       3264    323.503017          10.090e+6
0       2       2       3296    300.815105          10.957e+6
0       2       2       3328    300.695896          11.068e+6
0       2       2       3360    300.676823          11.175e+6
0       2       2       3392    319.306850          10.623e+6
0       2       2       3424    301.802158          11.345e+6
0       2       2       3456    297.517776          11.616e+6
0       2       2       3488    301.489830          11.569e+6
0       2       2       3520    301.108360          11.690e+6
0       2       2       3552    299.015045          11.879e+6
0       2       2       3584    301.392078          11.891e+6
0       2       2       3616    295.403004          12.241e+6
0       2       2       3648    321.764946          11.337e+6
0       2       2       3680    319.769382          11.508e+6
0       2       2       3712    296.459198          12.521e+6
0       2       2       3744    302.786827          12.365e+6
0       2       2       3776    297.429562          12.695e+6
0       2       2       3808    318.555832          11.954e+6
0       2       2       3840    300.135612          12.794e+6
0       2       2       3872    302.400589          12.804e+6
0       2       2       3904    321.772099          12.133e+6
0       2       2       3936    307.040215          12.819e+6
0       2       2       3968    298.411846          13.297e+6
0       2       2       4000    300.743580          13.300e+6
0       2       2       4032    301.330090          13.381e+6
0       2       2       4064    300.798416          13.511e+6
0       2       2       4096    277.788639          14.745e+6
```

A.2.5 Fitxer logscale_mesos.gpl

```
#p0     p1      dist    len     ave time (us)       rate
0       2       2       4       46.432018 86.147e+3
0       2       2       8       46.269894 172.899e+3
0       2       2       16      46.150684 346.690e+3
0       2       2       32      52.344799 611.331e+3
0       2       2       64      57.551861 1.112e+6
0       2       2       128     67.756176 1.889e+6
0       2       2       256     89.135170 2.872e+6
0       2       2       512     132.069588          3.877e+6
0       2       2       1024    221.691132          4.619e+6
0       2       2       2048    341.987610          5.989e+6
0       2       2       4096    505.821705          8.098e+6
0       2       2       8192    862.383842          9.499e+6
0       2       2       16384   1560.921669         10.496e+6
0       2       2       32768   2944.641113         11.128e+6
0       2       2       65536   5740.308762         11.417e+6
0       2       2       131072  11443.333626        11.454e+6
```

A.2.6 Fitxer goptest_mesos.gpl

```
#np time (us) for various sizes
2 203.795433 294.051170 523.328781 1068.806648
3 802.440643 806.193352 1398.763657 2596.039772
```

A.3 Fitxers de mètriques. Execució PERFTEST-1.5. Clúster MPD

A.3.1 Fitxer mpptest_mpd.gpl

#p0	p1	dist	len	ave time (us)	rate
0	2	2	0	102.751255	0.00
0	2	2	32	102.963448	310.790e+3
0	2	2	64	106.248856	602.359e+3
0	2	2	96	112.497807	853.350e+3
0	2	2	128	118.572712	1.080e+6
0	2	2	160	123.810768	1.292e+6
0	2	2	192	128.190517	1.498e+6
0	2	2	224	135.879517	1.649e+6
0	2	2	256	140.905380	1.817e+6
0	2	2	288	146.164894	1.970e+6
0	2	2	320	151.448250	2.113e+6
0	2	2	352	156.569481	2.248e+6
0	2	2	384	162.055492	2.370e+6
0	2	2	416	166.761875	2.495e+6
0	2	2	448	172.588825	2.596e+6
0	2	2	480	178.959370	2.682e+6
0	2	2	512	183.591843	2.789e+6
0	2	2	544	188.329220	2.889e+6
0	2	2	576	194.830894	2.956e+6
0	2	2	608	199.368000	3.050e+6
0	2	2	640	205.044746	3.121e+6
0	2	2	672	210.559368	3.191e+6
0	2	2	704	215.854645	3.261e+6
0	2	2	736	222.072601	3.314e+6
0	2	2	768	227.718353	3.373e+6
0	2	2	800	233.058929	3.433e+6
0	2	2	832	236.928463	3.512e+6
0	2	2	864	242.109299	3.569e+6
0	2	2	896	248.434544	3.607e+6
0	2	2	928	253.627300	3.659e+6
0	2	2	960	258.462429	3.714e+6
0	2	2	992	270.102024	3.673e+6
0	2	2	1024	276.436806	3.704e+6
0	2	2	1056	281.367302	3.753e+6
0	2	2	1088	286.109447	3.803e+6
0	2	2	1120	292.642117	3.827e+6
0	2	2	1152	299.754143	3.843e+6
0	2	2	1184	305.144787	3.880e+6
0	2	2	1216	309.500694	3.929e+6
0	2	2	1248	313.317776	3.983e+6
0	2	2	1280	313.124657	4.088e+6
0	2	2	1312	414.700508	3.164e+6
0	2	2	1344	411.379337	3.267e+6
0	2	2	1376	412.893295	3.333e+6
0	2	2	1408	413.289070	3.407e+6
0	2	2	1440	362.346172	3.974e+6
0	2	2	1472	361.897945	4.067e+6
0	2	2	1504	360.622406	4.171e+6
0	2	2	1536	363.717079	4.223e+6
0	2	2	1568	366.971493	4.273e+6
0	2	2	1600	364.069939	4.395e+6
0	2	2	1632	364.692211	4.475e+6
0	2	2	1664	423.550606	3.929e+6
0	2	2	1696	465.033054	3.647e+6
0	2	2	1728	464.046001	3.724e+6
0	2	2	1760	465.443134	3.781e+6
0	2	2	1792	466.828346	3.839e+6
0	2	2	1824	467.810631	3.899e+6
0	2	2	1856	469.560623	3.953e+6
0	2	2	1888	471.224785	4.007e+6
0	2	2	1920	471.739769	4.070e+6
0	2	2	1952	473.558903	4.122e+6
0	2	2	1984	475.192070	4.175e+6
0	2	2	2016	475.642681	4.238e+6
0	2	2	2048	476.021767	4.302e+6
0	2	2	2080	477.576256	4.355e+6
0	2	2	2112	481.343269	4.388e+6
0	2	2	2144	480.837822	4.459e+6
0	2	2	2176	481.278896	4.521e+6
0	2	2	2208	484.154224	4.561e+6
0	2	2	2240	483.829975	4.630e+6
0	2	2	2272	486.533642	4.670e+6
0	2	2	2304	487.792492	4.723e+6
0	2	2	2336	488.710403	4.780e+6
0	2	2	2368	489.981174	4.833e+6
0	2	2	2400	490.801334	4.890e+6
0	2	2	2432	492.355824	4.940e+6
0	2	2	2464	495.283604	4.975e+6
0	2	2	2496	495.505333	5.037e+6
0	2	2	2528	497.233868	5.084e+6
0	2	2	2560	497.624874	5.144e+6
0	2	2	2592	499.994755	5.184e+6
0	2	2	2624	500.421524	5.244e+6
0	2	2	2656	502.655506	5.284e+6
0	2	2	2688	503.916740	5.334e+6
0	2	2	2720	506.126881	5.374e+6
0	2	2	2752	505.526066	5.444e+6
0	2	2	2784	506.162643	5.500e+6
0	2	2	2816	509.021282	5.532e+6
0	2	2	2848	511.219501	5.571e+6
0	2	2	2880	520.265102	5.536e+6
0	2	2	2912	520.350933	5.596e+6
0	2	2	2944	522.320271	5.636e+6
0	2	2	2976	521.636009	5.705e+6
0	2	2	3008	523.436069	5.747e+6
0	2	2	3040	524.144173	5.800e+6
0	2	2	3072	533.828735	5.755e+6

0	2	2	3104	545.430183	5.691e+6
0	2	2	3136	564.389229	5.556e+6
0	2	2	3168	557.723045	5.680e+6
0	2	2	3200	567.317009	5.641e+6
0	2	2	3232	570.662022	5.664e+6
0	2	2	3264	571.095943	5.715e+6
0	2	2	3296	570.771694	5.775e+6
0	2	2	3328	563.476086	5.906e+6
0	2	2	3360	574.913025	5.844e+6
0	2	2	3392	584.046841	5.808e+6
0	2	2	3424	563.859940	6.072e+6
0	2	2	3456	580.608845	5.952e+6
0	2	2	3488	580.067635	6.013e+6
0	2	2	3520	574.457645	6.128e+6
0	2	2	3552	584.042072	6.082e+6
0	2	2	3584	576.217175	6.220e+6
0	2	2	3616	587.699413	6.153e+6
0	2	2	3648	582.566261	6.262e+6
0	2	2	3680	581.371784	6.330e+6
0	2	2	3712	592.312813	6.267e+6
0	2	2	3744	585.997105	6.389e+6
0	2	2	3776	593.023300	6.367e+6
0	2	2	3808	588.788986	6.468e+6
0	2	2	3840	589.661598	6.512e+6
0	2	2	3872	605.924129	6.390e+6
0	2	2	3904	603.451729	6.469e+6
0	2	2	3936	649.740696	6.058e+6
0	2	2	3968	651.049614	6.095e+6
0	2	2	4000	653.905869	6.117e+6
0	2	2	4032	653.760433	6.167e+6
0	2	2	4064	656.573772	6.190e+6
0	2	2	4096	657.081604	6.234e+6

A.3.2 Fitxer async_mpd.gpl

#p0	p1	dist	len	ave time (us)	rate
0	2	2	0	102.694035	0.00
0	2	2	32	103.509426	309.151e+3
0	2	2	64	106.799603	599.253e+3
0	2	2	96	113.167763	848.298e+3
0	2	2	128	119.032860	1.075e+6
0	2	2	160	123.980045	1.291e+6
0	2	2	192	129.246712	1.486e+6
0	2	2	224	135.743618	1.650e+6
0	2	2	256	139.994621	1.829e+6
0	2	2	288	145.759583	1.976e+6
0	2	2	320	151.293278	2.115e+6
0	2	2	352	156.338215	2.252e+6
0	2	2	384	162.081718	2.369e+6
0	2	2	416	167.658329	2.481e+6
0	2	2	448	172.820091	2.592e+6
0	2	2	480	178.830624	2.684e+6
0	2	2	512	184.147358	2.780e+6
0	2	2	544	189.201832	2.875e+6
0	2	2	576	195.834637	2.941e+6
0	2	2	608	199.697018	3.045e+6
0	2	2	640	204.982758	3.122e+6
0	2	2	672	210.297108	3.195e+6
0	2	2	704	216.770172	3.248e+6
0	2	2	736	221.977234	3.316e+6
0	2	2	768	226.595402	3.389e+6
0	2	2	800	232.565403	3.440e+6
0	2	2	832	237.510204	3.503e+6
0	2	2	864	243.301392	3.551e+6
0	2	2	896	249.481201	3.591e+6
0	2	2	928	254.750252	3.643e+6
0	2	2	960	259.573460	3.698e+6
0	2	2	992	270.340443	3.669e+6
0	2	2	1024	276.072025	3.709e+6
0	2	2	1056	281.229019	3.755e+6
0	2	2	1088	287.103653	3.790e+6
0	2	2	1120	292.215347	3.833e+6
0	2	2	1152	299.656391	3.844e+6
0	2	2	1184	304.732323	3.885e+6
0	2	2	1216	311.131477	3.908e+6
0	2	2	1248	313.735008	3.978e+6
0	2	2	1280	313.103199	4.088e+6
0	2	2	1312	412.077904	3.184e+6
0	2	2	1344	413.646698	3.249e+6
0	2	2	1376	413.849354	3.325e+6
0	2	2	1408	413.966179	3.401e+6
0	2	2	1440	365.912914	3.935e+6
0	2	2	1472	363.779068	4.046e+6
0	2	2	1504	363.695621	4.135e+6
0	2	2	1536	364.329815	4.216e+6
0	2	2	1568	364.744663	4.299e+6
0	2	2	1600	366.744995	4.363e+6
0	2	2	1632	365.366936	4.467e+6
0	2	2	1664	424.907207	3.916e+6
0	2	2	1696	461.654663	3.674e+6
0	2	2	1728	463.545322	3.728e+6
0	2	2	1760	465.385914	3.782e+6
0	2	2	1792	466.978550	3.837e+6
0	2	2	1824	468.173027	3.896e+6
0	2	2	1856	469.932556	3.950e+6
0	2	2	1888	469.522476	4.021e+6
0	2	2	1920	471.756458	4.070e+6
0	2	2	1952	474.660397	4.112e+6
0	2	2	1984	475.456715	4.173e+6

```
0       2       2       2016    476.369858      4.232e+6
0       2       2       2048    478.539467      4.280e+6
0       2       2       2080    478.687286      4.345e+6
0       2       2       2112    479.578972      4.404e+6
0       2       2       2144    481.131077      4.456e+6
0       2       2       2176    482.780933      4.507e+6
0       2       2       2208    485.198498      4.551e+6
0       2       2       2240    485.405922      4.615e+6
0       2       2       2272    487.775803      4.658e+6
0       2       2       2304    488.214493      4.719e+6
0       2       2       2336    490.887165      4.759e+6
0       2       2       2368    491.373539      4.819e+6
0       2       2       2400    492.722988      4.871e+6
0       2       2       2432    493.755341      4.926e+6
0       2       2       2464    495.831966      4.969e+6
0       2       2       2496    495.800972      5.034e+6
0       2       2       2528    499.968529      5.056e+6
0       2       2       2560    500.414371      5.116e+6
0       2       2       2592    500.848293      5.175e+6
0       2       2       2624    502.841473      5.218e+6
0       2       2       2656    503.430367      5.276e+6
0       2       2       2688    503.876209      5.335e+6
0       2       2       2720    507.667065      5.358e+6
0       2       2       2752    507.042408      5.428e+6
0       2       2       2784    509.762764      5.461e+6
0       2       2       2816    509.548187      5.526e+6
0       2       2       2848    511.999130      5.563e+6
0       2       2       2880    519.981384      5.539e+6
0       2       2       2912    522.398949      5.574e+6
0       2       2       2944    523.693562      5.622e+6
0       2       2       2976    524.082184      5.678e+6
0       2       2       3008    523.252487      5.749e+6
0       2       2       3040    525.028706      5.790e+6
0       2       2       3072    535.645485      5.735e+6
0       2       2       3104    550.704002      5.636e+6
0       2       2       3136    568.134785      5.520e+6
0       2       2       3168    556.797981      5.690e+6
0       2       2       3200    571.455956      5.600e+6
0       2       2       3232    561.380386      5.757e+6
0       2       2       3264    568.091869      5.746e+6
0       2       2       3296    567.953587      5.803e+6
0       2       2       3328    566.337109      5.876e+6
0       2       2       3360    570.778847      5.887e+6
0       2       2       3392    578.131676      5.867e+6
0       2       2       3424    574.333668      5.962e+6
0       2       2       3456    579.056740      5.968e+6
0       2       2       3488    581.414700      5.999e+6
0       2       2       3520    580.875874      6.060e+6
0       2       2       3552    579.414368      6.130e+6
0       2       2       3584    579.395294      6.186e+6
0       2       2       3616    577.206612      6.265e+6
0       2       2       3648    590.128899      6.182e+6
0       2       2       3680    576.376915      6.385e+6
0       2       2       3712    587.131977      6.322e+6
0       2       2       3744    596.504211      6.277e+6
0       2       2       3776    589.385033      6.407e+6
0       2       2       3808    588.214397      6.474e+6
0       2       2       3840    600.266457      6.397e+6
0       2       2       3872    596.430302      6.492e+6
0       2       2       3904    608.065128      6.420e+6
0       2       2       3936    646.510124      6.088e+6
0       2       2       3968    651.502609      6.091e+6
0       2       2       4000    652.356148      6.132e+6
0       2       2       4032    653.810501      6.167e+6
0       2       2       4064    654.416084      6.210e+6
0       2       2       4096    657.734871      6.227e+6
```

A.3.3 Fitxer bisect_mpd.gpl

```
#p0     p1      dist    len     ave time (us)   rate
0       2       2       0       299.031734      0.00
0       2       2       32      298.805237      160.640e+3
0       2       2       64      304.269791      315.509e+3
0       2       2       96      297.751427      483.625e+3
0       2       2       128     298.590660      643.021e+3
0       2       2       160     300.631523      798.319e+3
0       2       2       192     275.671482      1.045e+6
0       2       2       224     301.995277      1.113e+6
0       2       2       256     298.502445      1.286e+6
0       2       2       288     276.503563      1.562e+6
0       2       2       320     299.234390      1.604e+6
0       2       2       352     299.232006      1.765e+6
0       2       2       384     306.010246      1.882e+6
0       2       2       416     303.142071      2.058e+6
0       2       2       448     298.650265      2.250e+6
0       2       2       480     299.093723      2.407e+6
0       2       2       512     306.890011      2.503e+6
0       2       2       544     310.883522      2.625e+6
0       2       2       576     311.131477      2.777e+6
0       2       2       608     308.136940      2.960e+6
0       2       2       640     303.945541      3.158e+6
0       2       2       672     308.020115      3.273e+6
0       2       2       704     322.842598      3.271e+6
0       2       2       736     309.019089      3.573e+6
0       2       2       768     313.739777      3.672e+6
0       2       2       800     306.987762      3.909e+6
0       2       2       832     309.932232      4.027e+6
0       2       2       864     314.252377      4.124e+6
0       2       2       896     324.630737      4.140e+6
```

0	2	2	928	309.922695	4.491e+6
0	2	2	960	328.557491	4.383e+6
0	2	2	992	323.979855	4.593e+6
0	2	2	1024	315.864086	4.863e+6
0	2	2	1056	310.938358	5.094e+6
0	2	2	1088	322.616100	5.059e+6
0	2	2	1120	324.285030	5.181e+6
0	2	2	1152	324.590206	5.324e+6
0	2	2	1184	329.923630	5.383e+6
0	2	2	1216	449.602604	4.057e+6
0	2	2	1248	486.872196	3.845e+6
0	2	2	1280	481.002331	3.992e+6
0	2	2	1312	449.490547	4.378e+6
0	2	2	1344	431.921482	4.668e+6
0	2	2	1376	428.106785	4.821e+6
0	2	2	1408	467.767715	4.515e+6
0	2	2	1440	624.370575	3.459e+6
0	2	2	1472	624.303818	3.537e+6
0	2	2	1504	607.395172	3.714e+6
0	2	2	1536	607.326031	3.794e+6
0	2	2	1568	625.813007	3.758e+6
0	2	2	1600	632.259846	3.796e+6
0	2	2	1632	624.632835	3.919e+6
0	2	2	1664	800.597668	3.118e+6
0	2	2	1696	908.753872	2.799e+6
0	2	2	1728	900.928974	2.877e+6
0	2	2	1760	924.744606	2.855e+6
0	2	2	1792	892.829895	3.011e+6
0	2	2	1824	874.476433	3.129e+6
0	2	2	1856	909.020901	3.063e+6
0	2	2	1888	877.060890	3.229e+6
0	2	2	1920	899.837017	3.201e+6
0	2	2	1952	917.406082	3.192e+6
0	2	2	1984	918.371677	3.241e+6
0	2	2	2016	909.268856	3.326e+6
0	2	2	2048	909.512043	3.378e+6
0	2	2	2080	899.381638	3.469e+6
0	2	2	2112	893.185139	3.547e+6
0	2	2	2144	948.226452	3.392e+6
0	2	2	2176	899.932384	3.627e+6
0	2	2	2208	849.792957	3.897e+6
0	2	2	2240	924.878120	3.633e+6
0	2	2	2272	925.405025	3.683e+6
0	2	2	2304	886.113644	3.900e+6
0	2	2	2336	910.036564	3.850e+6
0	2	2	2368	910.842419	3.900e+6
0	2	2	2400	903.375149	3.985e+6
0	2	2	2432	909.812450	4.010e+6
0	2	2	2464	905.556679	4.081e+6
0	2	2	2496	917.994976	4.078e+6
0	2	2	2528	899.960995	4.214e+6
0	2	2	2560	924.668312	4.153e+6
0	2	2	2592	925.061703	4.203e+6
0	2	2	2624	910.780430	4.322e+6
0	2	2	2656	924.792290	4.308e+6
0	2	2	2688	900.037289	4.480e+6
0	2	2	2720	924.718380	4.412e+6
0	2	2	2752	924.608707	4.465e+6
0	2	2	2784	935.027599	4.466e+6
0	2	2	2816	935.146809	4.517e+6
0	2	2	2848	910.179615	4.694e+6
0	2	2	2880	899.980068	4.800e+6
0	2	2	2912	949.354172	4.601e+6
0	2	2	2944	936.298370	4.716e+6
0	2	2	2976	924.816132	4.827e+6
0	2	2	3008	935.325623	4.824e+6
0	2	2	3040	935.304165	4.875e+6
0	2	2	3072	899.832249	5.121e+6
0	2	2	3104	973.873138	4.781e+6
0	2	2	3136	1049.904823	4.480e+6
0	2	2	3168	997.931957	4.762e+6
0	2	2	3200	1047.124863	4.584e+6
0	2	2	3232	1080.915928	4.485e+6
0	2	2	3264	1059.551239	4.621e+6
0	2	2	3296	1075.134277	4.598e+6
0	2	2	3328	980.505943	5.091e+6
0	2	2	3360	1098.847389	4.587e+6
0	2	2	3392	1099.314690	4.628e+6
0	2	2	3424	972.316265	5.282e+6
0	2	2	3456	1072.392464	4.834e+6
0	2	2	3488	1000.051498	5.232e+6
0	2	2	3520	1030.158997	5.125e+6
0	2	2	3552	1072.847843	4.966e+6
0	2	2	3584	1047.844887	5.131e+6
0	2	2	3616	1074.743271	5.047e+6
0	2	2	3648	1021.120548	5.359e+6
0	2	2	3680	975.267887	5.660e+6
0	2	2	3712	1049.280167	5.306e+6
0	2	2	3744	998.778343	5.623e+6
0	2	2	3776	1024.963856	5.526e+6
0	2	2	3808	1025.729179	5.569e+6
0	2	2	3840	1024.999619	5.620e+6
0	2	2	3872	1199.355125	4.843e+6
0	2	2	3904	1174.988747	4.984e+6
0	2	2	3936	1224.205494	4.823e+6
0	2	2	3968	1223.545074	4.865e+6
0	2	2	4000	1199.975014	5.000e+6
0	2	2	4032	1200.098991	5.040e+6
0	2	2	4064	1199.636459	5.082e+6
0	2	2	4096	1174.771786	5.230e+6

A.3.4 Fitxer overlap_mpd.gpl

#p0	p1	dist	len	ave time (us)	rate
0	2	2	0	300.023556	0.00
0	2	2	32	287.981033	111.118e+3
0	2	2	64	293.235779	218.254e+3
0	2	2	96	299.870968	320.138e+3
0	2	2	128	295.183659	433.628e+3
0	2	2	160	302.982330	528.084e+3
0	2	2	192	300.142765	639.696e+3
0	2	2	224	303.199291	738.788e+3
0	2	2	256	300.388336	852.230e+3
0	2	2	288	269.696712	1.068e+6
0	2	2	320	301.790237	1.060e+6
0	2	2	352	297.729969	1.182e+6
0	2	2	384	296.387672	1.296e+6
0	2	2	416	301.473141	1.380e+6
0	2	2	448	294.773579	1.520e+6
0	2	2	480	300.602913	1.597e+6
0	2	2	512	301.067829	1.701e+6
0	2	2	544	277.845860	1.958e+6
0	2	2	576	298.130512	1.932e+6
0	2	2	608	281.808376	2.157e+6
0	2	2	640	299.911499	2.134e+6
0	2	2	672	296.030045	2.270e+6
0	2	2	704	295.000076	2.386e+6
0	2	2	736	301.582813	2.440e+6
0	2	2	768	294.723511	2.606e+6
0	2	2	800	297.687054	2.687e+6
0	2	2	832	294.783115	2.822e+6
0	2	2	864	275.251865	3.139e+6
0	2	2	896	299.544334	2.991e+6
0	2	2	928	298.020840	3.114e+6
0	2	2	960	294.203758	3.263e+6
0	2	2	992	301.022530	3.295e+6
0	2	2	1024	280.373096	3.652e+6
0	2	2	1056	297.915936	3.545e+6
0	2	2	1088	299.243927	3.636e+6
0	2	2	1120	296.313763	3.780e+6
0	2	2	1152	294.320583	3.914e+6
0	2	2	1184	300.686359	3.938e+6
0	2	2	1216	285.081863	4.265e+6
0	2	2	1248	276.949406	4.506e+6
0	2	2	1280	293.552876	4.360e+6
0	2	2	1312	297.694206	4.407e+6
0	2	2	1344	317.163467	4.238e+6
0	2	2	1376	320.529938	4.293e+6
0	2	2	1408	300.061703	4.692e+6
0	2	2	1440	298.032761	4.832e+6
0	2	2	1472	292.332172	5.035e+6
0	2	2	1504	303.044319	4.963e+6
0	2	2	1536	298.068523	5.153e+6
0	2	2	1568	302.550793	5.183e+6
0	2	2	1600	318.763256	5.019e+6
0	2	2	1632	296.649933	5.501e+6
0	2	2	1664	278.904438	5.966e+6
0	2	2	1696	301.043987	5.634e+6
0	2	2	1728	269.165039	6.420e+6
0	2	2	1760	300.085545	5.865e+6
0	2	2	1792	296.866894	6.036e+6
0	2	2	1824	291.070938	6.267e+6
0	2	2	1856	294.506550	6.302e+6
0	2	2	1888	297.019482	6.356e+6
0	2	2	1920	272.679329	7.041e+6
0	2	2	1952	299.937725	6.508e+6
0	2	2	1984	295.867920	6.706e+6
0	2	2	2016	297.825336	6.769e+6
0	2	2	2048	299.782753	6.832e+6
0	2	2	2080	295.059681	7.049e+6
0	2	2	2112	295.040607	7.158e+6
0	2	2	2144	278.403759	7.701e+6
0	2	2	2176	300.989151	7.229e+6
0	2	2	2208	295.920372	7.461e+6
0	2	2	2240	316.848755	7.070e+6
0	2	2	2272	299.792290	7.579e+6
0	2	2	2304	298.190117	7.727e+6
0	2	2	2336	302.166939	7.731e+6
0	2	2	2368	291.786194	8.116e+6
0	2	2	2400	294.654369	8.145e+6
0	2	2	2432	295.457840	8.231e+6
0	2	2	2464	300.948620	8.187e+6
0	2	2	2496	306.358337	8.147e+6
0	2	2	2528	300.474167	8.413e+6
0	2	2	2560	301.897526	8.480e+6
0	2	2	2592	302.336216	8.573e+6
0	2	2	2624	300.452709	8.733e+6
0	2	2	2656	296.609402	8.955e+6
0	2	2	2688	300.087929	8.957e+6
0	2	2	2720	319.092274	8.524e+6
0	2	2	2752	298.116207	9.231e+6
0	2	2	2784	298.194885	9.336e+6
0	2	2	2816	301.218033	9.349e+6
0	2	2	2848	299.286842	9.516e+6
0	2	2	2880	317.184925	9.080e+6
0	2	2	2912	302.271843	9.634e+6
0	2	2	2944	277.099609	10.624e+6
0	2	2	2976	319.201946	9.323e+6
0	2	2	3008	292.048454	10.300e+6
0	2	2	3040	274.829865	11.061e+6
0	2	2	3072	278.277397	11.039e+6
0	2	2	3104	275.204182	11.279e+6
0	2	2	3136	301.208496	10.411e+6

```
0       2       2       3168    300.681591      10.536e+6
0       2       2       3200    291.488171      10.978e+6
0       2       2       3232    299.324989      10.798e+6
0       2       2       3264    296.013355      11.027e+6
0       2       2       3296    310.800076      10.605e+6
0       2       2       3328    279.588699      11.903e+6
0       2       2       3360    294.303894      11.417e+6
0       2       2       3392    278.759003      12.168e+6
0       2       2       3424    305.488110      11.208e+6
0       2       2       3456    297.889709      11.602e+6
0       2       2       3488    277.547836      12.567e+6
0       2       2       3520    297.749043      11.822e+6
0       2       2       3552    318.179131      11.164e+6
0       2       2       3584    297.746658      12.037e+6
0       2       2       3616    273.458958      13.223e+6
0       2       2       3648    293.653011      12.423e+6
0       2       2       3680    300.509930      12.246e+6
0       2       2       3712    295.906067      12.545e+6
0       2       2       3744    312.974453      11.963e+6
0       2       2       3776    319.783688      11.808e+6
0       2       2       3808    296.568871      12.840e+6
0       2       2       3840    288.589001      13.306e+6
0       2       2       3872    300.157070      12.900e+6
0       2       2       3904    320.339203      12.187e+6
0       2       2       3936    297.014713      13.252e+6
0       2       2       3968    296.156406      13.398e+6
0       2       2       4000    300.233364      13.323e+6
0       2       2       4032    296.399593      13.603e+6
0       2       2       4064    292.279720      13.904e+6
0       2       2       4096    293.488503      13.956e+6
```

A.3.5 Fitxer logscale_mpd.gpl

```
#p0     p1      dist    len     ave time (us)   rate
0       2       2       4       102.176666      39.148e+3
0       2       2       8       102.171898      78.299e+3
0       2       2       16      102.593899      155.955e+3
0       2       2       32      102.539062      312.076e+3
0       2       2       64      104.556084      612.112e+3
0       2       2       128     117.237568      1.092e+6
0       2       2       256     139.174461      1.839e+6
0       2       2       512     182.585716      2.804e+6
0       2       2       1024    283.234119      3.615e+6
0       2       2       2048    478.868484      4.277e+6
0       2       2       4096    655.376911      6.250e+6
0       2       2       8192    1031.355858     7.943e+6
0       2       2       16384   1772.098541     9.246e+6
0       2       2       32768   3127.107620     10.479e+6
0       2       2       65536   5951.426029     11.012e+6
0       2       2       131072  11851.770878    11.059e+6
```

A.3.6 Fitxer goptest_mpd.gpl

```
#np time (us) for various sizes
2 249.381065 479.998589 800.347328 1707.305908
3 250.449181 632.510185 1144.065857 2012.434006
```

A.4 Fitxers de mètriques. Execució MPE. Clúster MESOS

A.4.1 Resultat execució is.B.1

```
NAS Parallel Benchmarks 3.3 -- IS Benchmark

 Size:  33554432  (class B)
 Iterations:   10
 Number of processes:     1

   iteration
        1
        2
        3
        4
        5
        6
        7
        8
        9
        10
```

```
 IS Benchmark Completed
 Class           =                        B
 Size            =                 33554432
 Iterations      =                       10
 Time in seconds =                     6.11
 Total processes =                        1
 Compiled procs  =                        1
 Mop/s total     =                    54.95
 Mop/s/process   =                    54.95
 Operation type  =              keys ranked
 Verification    =               SUCCESSFUL
 Version         =                      3.3
 Compile date    =              06 Dec 2014

 Compile options:
    MPICC        = /home/master/mpich2-install/bin/mpicc
    CLINK        = $(MPICC)
    CMPI_LIB     = -L/home/master/mpich2-install/lib -mpe=mpilog
    CMPI_INC     = (none)
    CFLAGS       = -O
    CLINKFLAGS   = -O


 Please send the results of this run to:

 NPB Development Team
 Internet: npb@nas.nasa.gov

 If email is not available, send this to:

 MS T27A-1
 NASA Ames Research Center
 Moffett Field, CA  94035-1000

 Fax: 650-604-3957

Connecting to Mesos master 10.10.78.85:5050
MPD_PID is master_54100
Mesos MPI scheduler and mpd running at master:54100
Registered with framework ID 20141210-204417-1431177738-5050-1460-0001
Got 1 resource offers
Considering resource offer 20141210-204417-1431177738-5050-1460-1 from master
Accepting offer on master to start mpd 0
Replying to offer: launching mpd 0 on host master
We've launched all our MPDs; waiting for them to come up
...waiting on MPD(s)...
...waiting on MPD(s)...
Task 0 in state 1
Got 1 mpd(s), running mpiexec
Running mpiexec
mpiexec completed, calling mpdallexit master_54100
Task 0 in state 2
All tasks done, all mpd's closed, exiting
```

A.4.2 Resultat execució is.B.2

```
NAS Parallel Benchmarks 3.3 -- IS Benchmark

 Size:  33554432  (class B)
 Iterations:   10
 Number of processes:     2

   iteration
        1
        2
        3
        4
        5
        6
        7
        8
```

```
        9
        10


 IS Benchmark Completed
 Class           =                        B
 Size            =                 33554432
 Iterations      =                       10
 Time in seconds =                    47.60
 Total processes =                        2
 Compiled procs  =                        2
 Mop/s total     =                     7.05
 Mop/s/process   =                     3.52
 Operation type  =              keys ranked
 Verification    =               SUCCESSFUL
 Version         =                      3.3
 Compile date    =              06 Dec 2014

 Compile options:
    MPICC        = /home/master/mpich2-install/bin/mpicc
    CLINK        = $(MPICC)
    CMPI_LIB     = -L/home/master/mpich2-install/lib -mpe=mpilog
    CMPI_INC     = (none)
    CFLAGS       = -O
    CLINKFLAGS   = -O


 Please send the results of this run to:

 NPB Development Team
 Internet: npb@nas.nasa.gov

 If email is not available, send this to:

 MS T27A-1
 NASA Ames Research Center
 Moffett Field, CA  94035-1000

 Fax: 650-604-3957

Connecting to Mesos master 10.10.78.85:5050
MPD_PID is master_48670
Mesos MPI scheduler and mpd running at master:48670
Registered with framework ID 20141206-141127-1431177738-5050-2205-0004
Got 3 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-28 from master
Accepting offer on master to start mpd 0
Replying to offer: launching mpd 0 on host master
Considering resource offer 20141206-141127-1431177738-5050-2205-29 from slave3
Accepting offer on slave3 to start mpd 1
Replying to offer: launching mpd 1 on host slave3
Considering resource offer 20141206-141127-1431177738-5050-2205-30 from slave1
Declining permanently because we have already launched enough tasks
We've launched all our MPDs; waiting for them to come up
...waiting on MPD(s)...
Task 1 in state 1
...waiting on MPD(s)...
Task 0 in state 1
Got 2 mpd(s), running mpiexec
Running mpiexec
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-31 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-32 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-33 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-34 from slave1
Declining permanently because we have already launched enough tasks
```

```
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-35 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-36 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-37 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-38 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-39 from slave1
Declining permanently because we have already launched enough tasks
Got 1 resource offers
Considering resource offer 20141206-141127-1431177738-5050-2205-40 from slave1
Declining permanently because we have already launched enough tasks
mpiexec completed, calling mpdallexit master_48670
Task 0 in state 2
Task 1 in state 2
All tasks done, all mpd's closed, exiting
```

A.5 Fitxers de mètriques. Execució MPE. Clúster MPD

A.5.1 Resultat execució is.B.1

```
NAS Parallel Benchmarks 3.3 -- IS Benchmark

 Size:  33554432  (class B)
 Iterations:   10
 Number of processes:     1

   iteration
        1
        2
        3
        4
        5
        6
        7
        8
        9
        10


 IS Benchmark Completed
 Class           =                        B
 Size            =                 33554432
 Iterations      =                       10
 Time in seconds =                     5.62
 Total processes =                        1
 Compiled procs  =                        1
 Mop/s total     =                    59.66
 Mop/s/process   =                    59.66
 Operation type  =              keys ranked
 Verification    =               SUCCESSFUL
 Version         =                      3.3
 Compile date    =              06 Dec 2014

 Compile options:
    MPICC        = /home/master/mpich2-install/bin/mpicc
    CLINK        = $(MPICC)
    CMPI_LIB     = -L/home/master/mpich2-install/lib -mpe=mpilog
    CMPI_INC     = (none)
    CFLAGS       = -O
    CLINKFLAGS   = -O
```

```
 Please send the results of this run to:

 NPB Development Team
 Internet: npb@nas.nasa.gov

 If email is not available, send this to:

 MS T27A-1
 NASA Ames Research Center
 Moffett Field, CA  94035-1000

 Fax: 650-604-3957
```

A.5.2 Resultat execució is.B.2

```
NAS Parallel Benchmarks 3.3 -- IS Benchmark

 Size:  33554432  (class B)
 Iterations:   10
 Number of processes:     2

   iteration
        1
        2
        3
        4
        5
        6
        7
        8
        9
        10


 IS Benchmark Completed
 Class           =                        B
 Size            =                 33554432
 Iterations      =                       10
 Time in seconds =                    48.15
 Total processes =                        2
 Compiled procs  =                        2
 Mop/s total     =                     6.97
 Mop/s/process   =                     3.48
 Operation type  =              keys ranked
 Verification    =               SUCCESSFUL
 Version         =                      3.3
 Compile date    =              06 Dec 2014

 Compile options:
    MPICC        = /home/master/mpich2-install/bin/mpicc
    CLINK        = $(MPICC)
    CMPI_LIB     = -L/home/master/mpich2-install/lib -mpe=mpilog
    CMPI_INC     = (none)
    CFLAGS       = -O
    CLINKFLAGS   = -O


 Please send the results of this run to:

 NPB Development Team
 Internet: npb@nas.nasa.gov

 If email is not available, send this to:

 MS T27A-1
 NASA Ames Research Center
 Moffett Field, CA  94035-1000

 Fax: 650-604-3957
```