INFRAESTRUCTURA VIRTUAL

TFC – GNU/Linux

1er Semestre 2011/2012

Autor: Miquel Morey Riera

Índex

Introducció

En els darrers anys s'ha sentit a parlar molt de les tecnologies de virtualització. De fet és molt fàcil disposar d'una eina d'aquest tipus. La forta competència de companyies que volen posicionar-se en aquest sector, juntament amb l'aparició d'aquestes eines de codi obert, han propiciat que estiguin disponibles una amplia gama de sistemes hipervisors a l'abast de tothom.

Així, moltes empreses que proporcionen o requereixen de tecnologies de la informació han pogut implementar fàcilment sistemes de virtualització, ja sigui mitjançant la compra d'aquests serveis, implementant productes de codi obert o bé amb productes gratuïts.

Els CPD que àmpliament han augmentat el parc de servidors amb sistemes de virtualització tant gratuïts o com de codi obert, s'han trobat amb dificultats alhora de gestionar els recursos TI. Així, requereixen de noves eines per gestionar d'una manera més eficient la seva infraestructura d'hipervisors i de màquines virtuals.

Objectius

Realitzar un estudi en profunditat dels principals programaris GNU/Linux que proporcionen plataformes de computació en núvol, dels seus components i de les tecnologies necessàries amb la finalitat de:

- Analitzar el funcionament de les plataformes en quant a escalabilitat, rendiment i fiabilitat.
- Comparar els sistemes d'emmagatzemament que proporciona cada un d'ells i les capacitats que ofereixen.
- Analitzar la compatibilitat amb diferents hipervisors i quins d'ells son els més aptes per la virtualització.
- Comparar les interfícies gràfiques en quant a funcionalitat.
- Avaluar la possibilitat de migració des de un entorn VMware Server.
- Dissenyar i implementar una infraestructura que mitjançant un dels programaris proporcioni servei d'infraestructura virtual.
- Avaluar el grau de complexitat que requereix la instal·lació, configuració i gestió del programari.

Definició de projecte

El projecte que es desenvolupa contempla un cas real del departament de TI de l'empresa EFAKEMA dedicada al sector de serveis informàtics. Aquest departament ve utilitzant les tecnologies de virtualització, des de fa més de 5 anys, per proporcionar molts dels seus serveis.

En aquells moments es va optar per provar un dels programaris més populars del sector: VMware Server -gratuït però propietari- va esdevenir la primera plataforma de virtualització del departament. De seguida es varen apreciar les avantatges que oferia aquesta tecnologia que al llarg dels anys -i quasi de manera involuntària- ha anat consolidant-se en entorns de desenvolupament i de producció.

Així, el parc de servidors que ofereixen virtualització s'ha anat ampliant, fins al punt que la gestió de la infraestructura d'hipervisors i de les seves màquines virtuals s'ha tornat prou complexa i difícil de gestionar. VMware Server només proporciona eines d'administració a nivell d'amfitrió, és a dir, la gestió de polítiques d'accés, el control de màquines virtuals i de recursos de computació s'han de realitzar a cada un dels hipervisors.

VMware Server no garanteix la tolerància a fallades del maquinari en un entorn on hi ha sistemes que requereixen d'alta disponibilitat. El simple fet d'afegir memòria al servidor de virtualització pot suposar l'aturada d'un o varis sistemes hoste amb els corresponents serveis que aquestes proporcionen, fins i tot durant hores en el cas d'una averia greu. A més, amb VMware Server estan limitades la quantitat de CPU i memòria que es poden oferir a les màquines virtuals.

És per això que, arribats a aquest punt, cal fer un replantejament de la situació. Per una banda, es pot optar per la compra d'una solució de programari, per exemple del mateix fabricant, amb la finalitat de proporcionar una millora en la gestió de la infraestructura virtual i en les capacitats dels recursos. Per altra banda, optar per la migració a un sistema de codi obert que resolgui la situació actual.

Tant amb una opció com en l'altre es fa indispensable disposar d'un sistema d'emmagatzemament redundat i escalable per allotjar els discs virtuals, ja que és una de les solucions més viables per dotar a la infraestructura de la seguretat necessària per garantir l'alta disponibilitat.

Així, es pretén implementar una solució de programari -que proporcioni Servei d'Infraestructura Virtual- basat en GNU/Linux que resolgui el problema de la disponibilitat i les seves limitacions en quant a les capacitats de computació (CPU, RAM), que proporcioni un sistema d'emmagatzemament dinàmic i que tot plegat es pugui administrar de manera centralitzada.

Agraïments

Agrair a l’empresa EFAKEMA i al cap de servei de TI la disposició del maquinari adequat per dur a terme el projecte. Això ha permès la instal·lació i configuració de la plataforma de computació en núvol en una infraestructura de maquinari real amb les conseqüents avantatges que proporciona alhora de comprovar el rendiment de la plataforma.

També, agrair al consultor les orientacions i recomanacions proporcionades que en part han permès dur el projecte a bon terme.

Finalment, agrair a la meva dona i als meus fills la paciència i comprensió que han tingut amb el meu absentisme i poca dedicació a la família durant la realització d'aquest projecte.

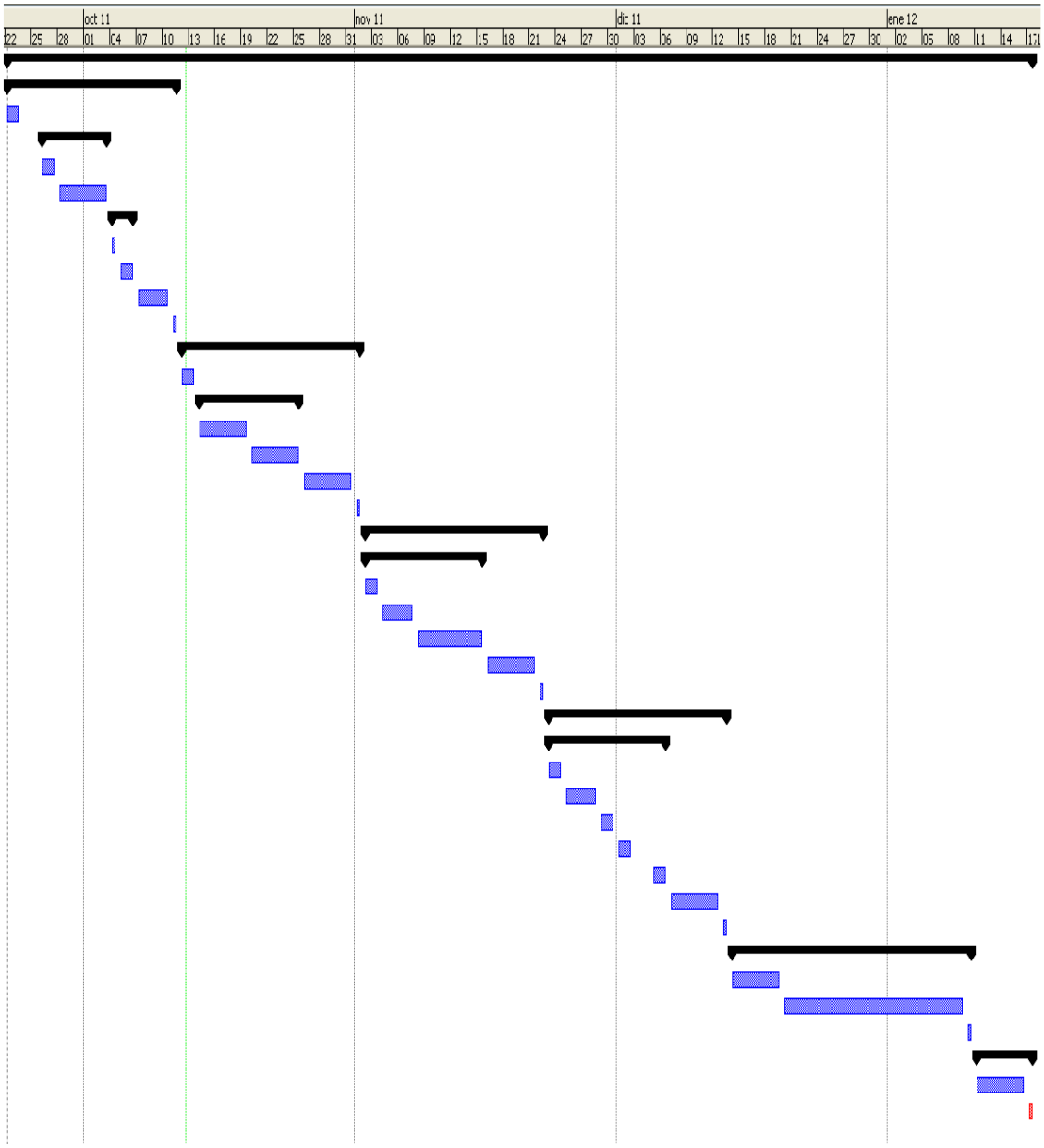
Planificació del projecte

La planificació inicial contempla implementar una plataforma de computació amb el programari OpenStack. Però en el procés d'investigació s'han descobert altres programaris que ofereixen els mateixos serveis i que també han estat objecte d'estudi. D'aquesta manera el pla de treball ha sofert alguns canvis.

La planificació inicial es pot visualitzar en el diagrama de procés (il·lustració 1) i en el diagrama de Gantt (Il·lustració 2).

		Nombre	Duración	Inicio	Terminado
1		**TFC**	84 days?	**22/09/11 8:00**	**17/01/12 17:00**
2		**PAC1: Pla de treball**	14 days?	**22/09/11 8:00**	**11/10/11 17:00**
3		Lectura pla docent i introducció al TFC	2 days?	22/09/11 8:00	23/09/11 17:00
4		**Definició del projecte**	6 days?	**26/09/11 8:00**	**3/10/11 17:00**
5		Estudi del módul "Gestió i desenvolupament de projectes"	2 days?	26/09/11 8:00	27/09/11 17:00
6		Introducció a la plataforma OpenStack	4 days?	28/09/11 8:00	3/10/11 17:00
7		**Pla de treball**	3 days?	**4/10/11 8:00**	**6/10/11 17:00**
8		Instal·lació i aprenentatge d'OpenProj	1 day?	4/10/11 8:00	4/10/11 17:00
9		Edició del pla de treball	2 days?	5/10/11 8:00	6/10/11 17:00
10		Edició Pac1	2 days?	7/10/11 8:00	10/10/11 17:00
11		Entrega Pac1	1 day?	11/10/11 8:00	11/10/11 17:00
12		**PAC2: Descripció de les tecnologies**	15 days?	**12/10/11 8:00**	**1/11/11 17:00**
13		Estudi del mòdul "Redacció de textos cientificotècnics"	2 days?	12/10/11 8:00	13/10/11 17:00
14		**Estudi de les tecnologies OpenStack**	8 days?	**14/10/11 8:00**	**25/10/11 17:00**
15		Aprofundiment en els components de la plataforma	4 days?	14/10/11 8:00	19/10/11 17:00
16		Aprofundiment en architectures d'alta disponibilitat	4 days?	20/10/11 8:00	25/10/11 17:00
17		Edició Pac2	4 days?	26/10/11 8:00	31/10/11 17:00
18		Entrega Pac2	1 day?	1/11/11 8:00	1/11/11 17:00
19		**PAC3: Disseny i implementació del projecte**	15 days?	**2/11/11 8:00**	**22/11/11 17:00**
20		**Implementació de la plataforma OpenStack**	10 days?	**2/11/11 8:00**	**15/11/11 17:00**
21		Disseny d'una architectura d'alta disponibilitat	2 days?	2/11/11 8:00	3/11/11 17:00
22		Preparació del maquinari necessari	2 days?	4/11/11 8:00	7/11/11 17:00
23		Instal·lació dels components	6 days?	8/11/11 8:00	15/11/11 17:00
24		Edició Pac3	4 days?	16/11/11 8:00	21/11/11 17:00
25		Entrega Pac3	1 day?	22/11/11 8:00	22/11/11 17:00
26		**PAC4: Resultats i anàlisis del treball**	15 days?	**23/11/11 8:00**	**13/12/11 17:00**
27		**Anàlisi de la plataforma OpenStack**	10 days?	**23/11/11 8:00**	**6/12/11 17:00**
28		Anàlisi d'escalabilitat	2 days?	23/11/11 8:00	24/11/11 17:00
29		Anàlisi de rendiment	2 days?	25/11/11 8:00	28/11/11 17:00
30		Anàlisi de fiabilitat	2 days?	29/11/11 8:00	30/11/11 17:00
31		Anàlisi de migració des de VMWare	2 days?	1/12/11 8:00	2/12/11 17:00
32		Avaluació del grau de complexitat	2 days?	5/12/11 8:00	6/12/11 17:00
33		Edició Pac4	4 days?	7/12/11 8:00	12/12/11 17:00
34		Entrega Pac4	1 day?	13/12/11 8:00	13/12/11 17:00
35		**Entregra Final: memòria**	20 days?	**14/12/11 8:00**	**10/01/12 17:00**
36		Estudi mòdul "Presentació de document i elaboració de presentacions"	4 days?	14/12/11 8:00	19/12/11 17:00
37		Acoblament i edició memòria	15 days?	20/12/11 8:00	9/01/12 17:00
38		Entrega memòria	1 day?	10/01/12 8:00	10/01/12 17:00
39		**Entrega Final: Presentació virtual**	5 days?	**11/01/12 8:00**	**17/01/12 17:00**
40		Edició presentació virtual	4 days?	11/01/12 8:00	16/01/12 17:00
41		Entrega presentació virtual	1 day?	17/01/12 8:00	17/01/12 17:00

Il·lustració 1: Diagrama de procés de la planificació inicial



Il·lustració 2: Diagrama de Gantt de la planificació inicial

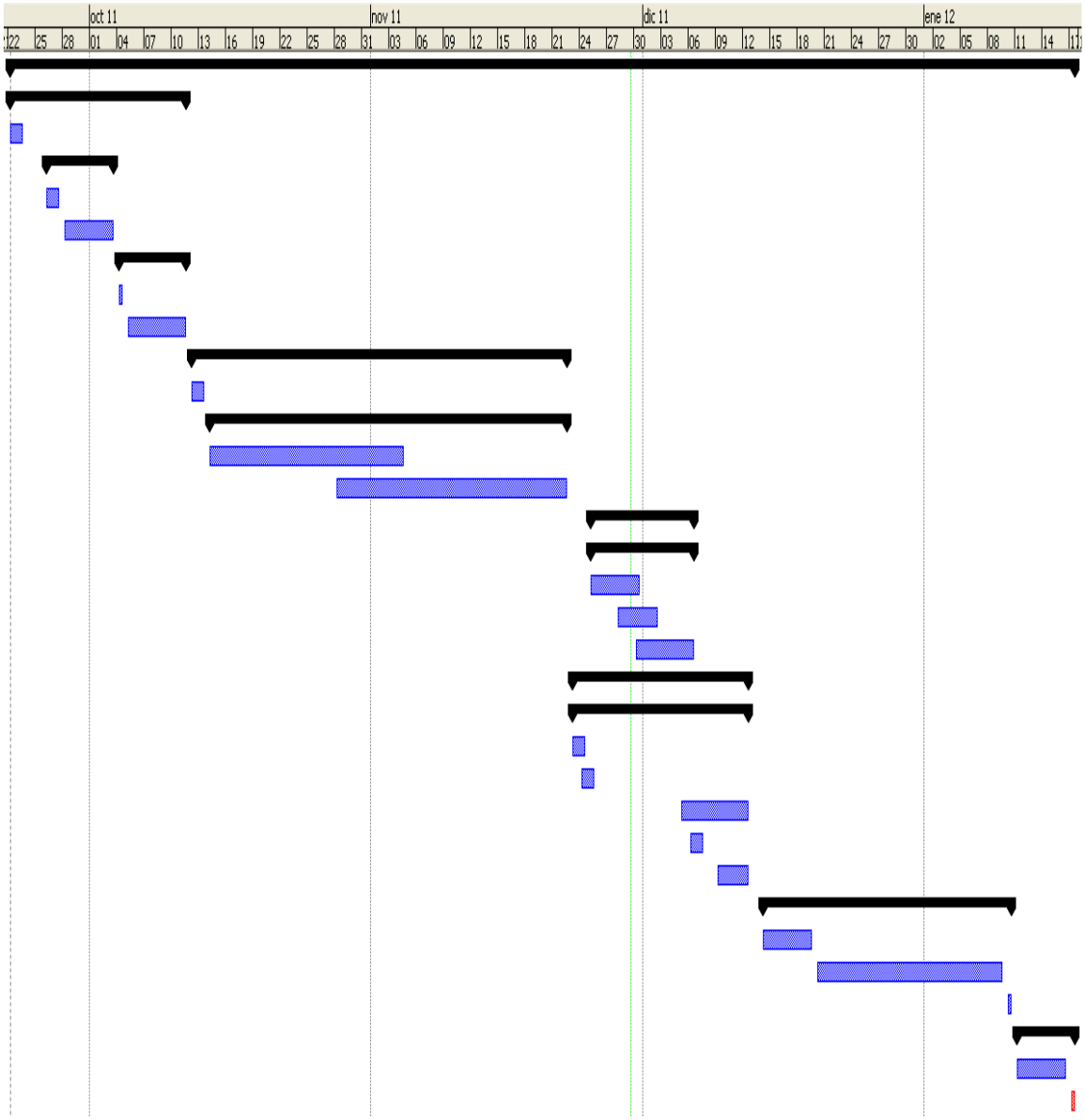
Un cop realitzat l'estudi en profunditat de la plataforma OpenStack s'ha observat que aquest programari no s'adapta completament a les necessitats del problema que pretén resoldre. Així, per tal d'aconseguir els objectius marcats, ha estat necessària l'anàlisi d'altres plataformes de codi obert que proporcionen infraestructura com a servei.

Amb la finalitat d'implementar la plataforma de computació en núvol que millor s'adapti a un entorn privat i en constant creixement, la investigació ha portat a l'estudi de dos programaris alternatius, Eucalyptus i OpenNebula. Aquest fet ha provocat de forma inevitable una re-planificació del pla de treball inicial.

La nova planificació es pot visualitzar en el diagrama de procés (il·lustració 3) i en el diagrama de Gantt (il·lustració 4).

	Nombre	Duración	Inicio	Terminado
1	**TFC**	84 days?	**22/09/11 8:00**	**17/01/12 17:00**
2	**PAC1: Pla de treball**	14 days?	**22/09/11 8:00**	**11/10/11 17:00**
3	Lectura pla docent i introducció al TFC	2 days?	22/09/11 8:00	23/09/11 17:00
4	**Definició del projecte**	6 days?	**26/09/11 8:00**	**3/10/11 17:00**
5	Estudi del módul "Gestió i desenvolupament de projectes"	2 days?	26/09/11 8:00	27/09/11 17:00
6	Introducció a la plataforma OpenStack	4 days?	28/09/11 8:00	3/10/11 17:00
7	**Pla de treball**	6 days?	**4/10/11 8:00**	**11/10/11 17:00**
8	Instal·lació i aprenentatge d'OpenProj	1 day?	4/10/11 8:00	4/10/11 17:00
9	Edició del pla de treball	5 days?	5/10/11 8:00	11/10/11 17:00
10	**PAC2: Descripció de les tecnologies**	30 days?	**12/10/11 8:00**	**22/11/11 17:00**
11	Estudi del mòdul "Redacció de textos cientificotècnics"	2 days?	12/10/11 8:00	13/10/11 17:00
12	**Estudi de les tecnologies de computació en núvol**	28 days?	**14/10/11 8:00**	**22/11/11 17:00**
13	Aprofundiment en els components de les plataformes	16 days?	14/10/11 8:00	4/11/11 17:00
14	Aprofundiment en architectures d'alta disponibilitat	18 days?	28/10/11 8:00	22/11/11 17:00
15	**PAC3: Disseny i implementació del projecte**	8 days?	**25/11/11 8:00**	**6/12/11 17:00**
16	**Implementació de la plataforma**	8 days?	**25/11/11 8:00**	**6/12/11 17:00**
17	Disseny d'una architectura d'alta disponibilitat	4 days?	25/11/11 8:00	30/11/11 17:00
18	Preparació del maquinari necessari	5 days?	28/11/11 8:00	2/12/11 17:00
19	Instal·lació dels components	5 days?	30/11/11 8:00	6/12/11 17:00
20	**PAC4: Resultats i anàlisis del treball**	14 days?	**23/11/11 8:00**	**12/12/11 17:00**
21	**Anàlisi de les plataformes**	14 days?	**23/11/11 8:00**	**12/12/11 17:00**
22	Anàlisi d'escalabilitat	2 days?	23/11/11 8:00	24/11/11 17:00
23	Anàlisi de fiabilitat	2 days?	24/11/11 8:00	25/11/11 17:00
24	Avaluació del grau de complexitat	6 days?	4/12/11 8:00	12/12/11 17:00
25	Anàlisi de migració des de VMWare	2 days?	6/12/11 8:00	7/12/11 17:00
26	Anàlisi de rendiment	2 days?	9/12/11 8:00	12/12/11 17:00
27	**Entregra Final: memòria**	20 days?	**14/12/11 8:00**	**10/01/12 17:00**
28	Estudi mòdul "Presentació de document i elaboració de presentacions"	4 days?	14/12/11 8:00	19/12/11 17:00
29	Acoblament i edició memòria	15 days?	20/12/11 8:00	9/01/12 17:00
30	Entrega memòria	1 day?	10/01/12 8:00	10/01/12 17:00
31	**Entrega Final: Presentació virtual**	5 days?	**11/01/12 8:00**	**17/01/12 17:00**
32	Edició presentació virtual	4 days?	11/01/12 8:00	16/01/12 17:00
33	Entrega presentació virtual	1 day?	17/01/12 8:00	17/01/12 17:00

Il·lustració 3: 1: Diagrama de procés de la re-planificació

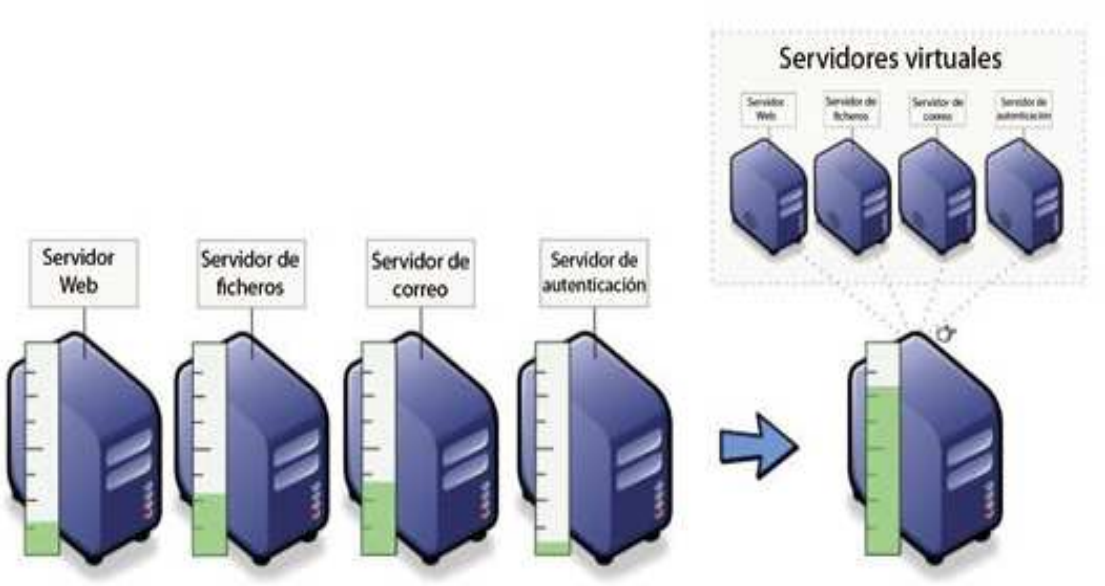


Il·lustració 4: Diagrama de Gantt de la re-planificació

1. Raó de la virtualització

En els darrers anys una de les tecnologies de computació que més ha evolucionat ha estat la virtualització de maquinari, sobretot en les plataformes x86 pel seu baix cost en comparació amb altres. La principal causa d'això és el gran potencial que ofereixen les computadores modernes (que en molts de casos no s'assoleix).

En els CPD (centres de processament de dades) és habitual disposar d'un servidor per cada servei o aplicació corporativa per tal de proporcionar la màxima continuïtat de negoci. Així, el mal funcionament d'una aplicació, reinicis per actualitzacions, possibles incompatibilitats entre aplicacions, etc. no afecten a la resta de serveis. D'aquesta manera, en un entorn clàssic (aplicació per servidor físic) moltes d'aquestes computadores estan infrautilitzades.



Il·lustració 5: Virtualització de maquinari

La tecnologia de virtualització resol aquest problema, ja que permet instal·lar i configurar múltiples computadores o servidors completament independents (coneguts com 'màquines virtuals') en un únic servidor físic. Tot i que les màquines virtuals comparteixen el mateix maquinari, cada una treballa de manera totalment independent (amb el seu propi sistema operatiu, aplicacions, configuracions, etcètera).

1.1 Avantatges i inconvenients

La virtualització té un ampli ventall d'avantatges.

- Estalvi en equipament: l'optimització dels recursos de computació comporta disposar de menys maquinari i en conseqüència de menys espai físic, menys potencia de refrigeració i menys manteniment del maquinari.

- Estalvi energètic: un aprofitament més òptim del recursos de computació esdevindrà d'igual manera en un consum elèctric més eficient.

- Independència del maquinari: la capa de virtualització proporciona independència del maquinari, és a dir, que el sistema operatiu virtualitzat només coneix els recursos de maquinari que li ofereix la màquina virtual. Així, la migració d'una màquina virtual d'un servidor físic a un altra és tan fàcil com moure un fitxer gran entre servidors, evitant les costoses tasques d'instal·lació i configuració d'algunes aplicacions.

- Capacitats d'aprovisionament: En qüestió de minuts es pot aprovisionar una nova màquina virtual o una màquina virtual preconfigurada, mantenint la contenció de costos, principalment en estudis de viabilitat o a l'inici de nous projectes.

- Augment de disponibilitat: el temps en que una màquina virtual estarà fora de servei es redueix considerablement gràcies a la independència del maquinari, a la velocitat d'aprovisionament i a l'agilitat de moure o fer còpies de seguretat de les màquines virtuals.

- Manteniment d'aplicacions heretades: les màquines virtuals poden proporcionar compatibilitat amb sistemes operatius no suportats (no disposen de controladors pel maquinari modern) i que cal mantenir a causa de la utilització d'algunes aplicacions que no s'han adaptat a noves versions de sistemes operatius.

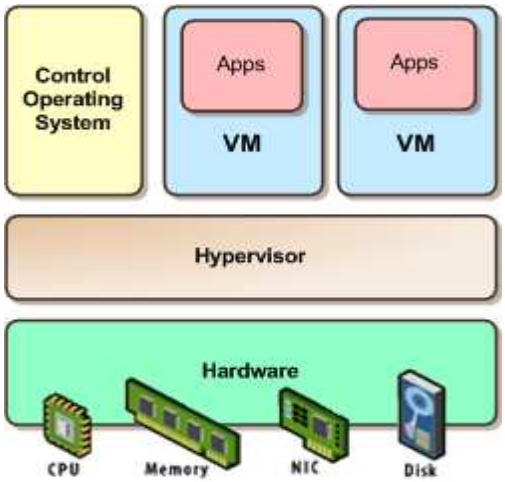
Per contra, la virtualització produeix una pèrdua de rendiment en la computació. Permetre que varis sistemes operatius funcionin a sobre la mateixa computadora requereix d'un mecanisme per aïllar-los, aquest mecanisme s'implementa mitjançant un programari que rep el nom d'hipervisor. Així, la mediació de l'hipervisor entre el maquinari real i els sistemes virtuals provoca un cost addicional a la computació.

2. Hipervisors

L'hipervisor o també VMM (Virtual Machine Monitor) no és més que una capa de software amb la capacitat d'abstreure el maquinari del servidor físic -principalment CPU, memòria RAM, xarxa i emmagatzemament- i proporcionar-lo com a interfície a cada un dels sistemes operatius virtualitzats que allotja.

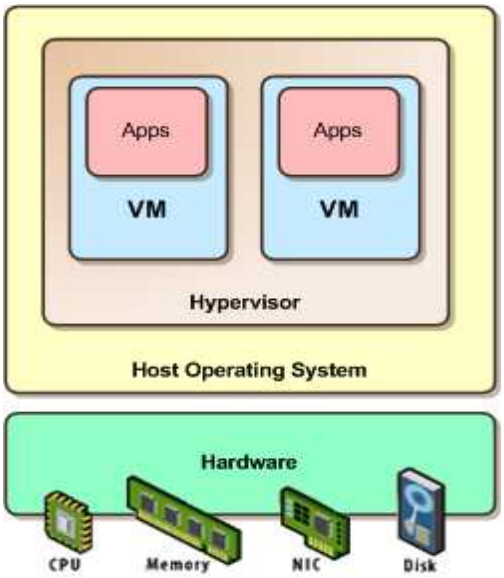
Tradicionalment els hipervisors es classifiquen en dos tipus:

- Els de tipus I (hipervisors natius o *bare-metal*) integren el nucli del sistema adaptat i optimitzat per proporcionar la capa de virtualització, és per això que es diu que no requereixen de sistema operatiu amfitrió (Il·lustració 6). Aquests, en general són més eficients i per això més aptes per entorns de producció. Alguns exemples d'aquest tipus són: VMware ESXi (gratuït), VMware ESX (de pagament), Xen (lliure), Citrix XenServer (gratuït), Microsoft Hyper-V Server (gratuït).



Il·lustració 6: Arquitectura d'un hipervisor natiu

- Els de tipus II (hipervisors hostejats) on el programari que proporciona la capa de virtualització requereix ser instal·lat a sobre d'un sistema operatiu amfitrió (Linux, Windows, MacOS, etcètera). D'aquesta manera, permeten l'execució d'aplicacions tant en els sistemes virtuals com en el físic, més aptes per l'usuari d'escriptori (Il·lustració 7). Alguns exemples d'aquests són: VirtualBox (gratuït), VirtualBox OSE (lliure), VMware Workstation (de pagament), VMware Server (gratuït), VMware Player (gratuït), QEMU (lliure), MS Virtual PC, MS Virtual Server.



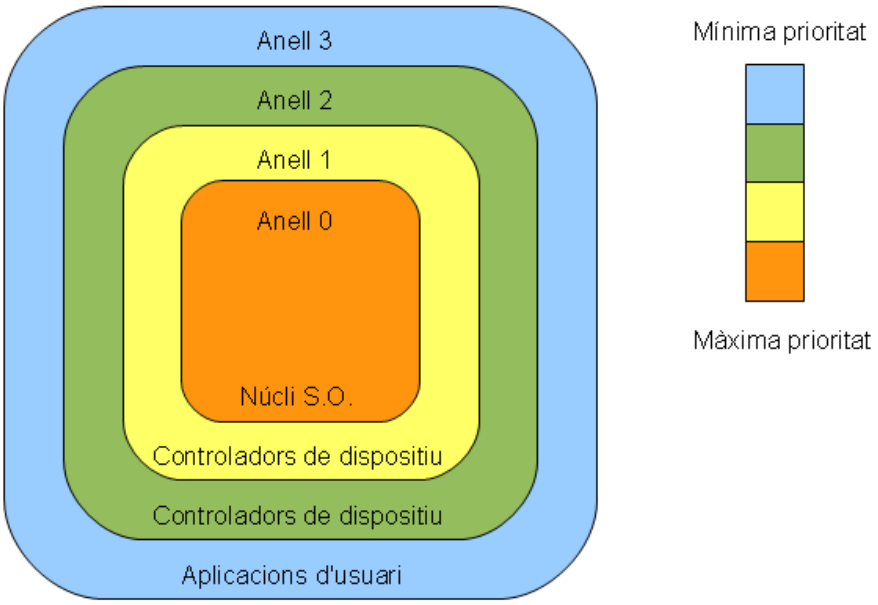
Il·lustració 7: Arquitectura d'un hipervisor hostejat

Tot i que alguns desenvolupaments es poden classificar fàcilment en un grup o en l'altre, n'hi ha d'altres que causen cert debat alhora de classificar-los. Per exemple *KVM* i *Microsoft Virtual Server* implementen la capa de virtualització en el nucli del sistema i també permeten un funcionament com un sistema operatiu convencional. D'aquesta manera, aquests hipervisors disposen dels avantatges dels dos tipus.

3. Virtualització en plataformes x86

Amb l'evolució de les tecnologies de virtualització han aparegut i evolucionat diferents tècniques per realitzar aquesta tasca. Per entendrer-les millor caldrà fer una breu introducció al mode protegit i als nivells de protecció que les CPU x86 vénen implementant des de fa més de dues dècades.

El mode protegit proporciona un rang jeràrquic de nivells de protecció o anells, enumerats del 0 al 3, en els quals es pot executar codi (il·lustració 8). L'anell 0 té el nivell de privilegi més alt, el codi que s'hi executa es diu que s'executa en l'espai del sistema, *kernel mode* o *supervisor mode*. Tota la resta de codi, tal com les aplicacions d'usuari, opera en anells de menys privilegi, típicament a l'anell 3.



Il·lustració 8: Anells de privilegi de l'arquitectura x86 tradicional i els tipus d'ús més freqüents

La majoria de nuclis dels sistemes operatius estan escrits explícitament per funcionar a l'anell 0, ja que necessiten realitzar tasques que només estan disponibles en aquest nivell, tals com l'execució d'instruccions privilegiades de la CPU i la manipulació directa de la memòria.

Així, depenent de la forma en que una màquina virtual interactua amb els recursos físics del maquinari es poden classificar diferents tipus de virtualització: emulació de maquinari, virtualització completa, paravirtualització i virtualització a nivell de sistema operatiu.

3.1 Emulació de maquinari

En l'emulació de maquinari, la màquina virtual simula completament la plataforma del maquinari, incloent l'execució de totes les instruccions a nivell de CPU. D'aquesta manera permet simular arquitectures diferents a la que proporciona el maquinari real, i així, virtualitzar sistemes operatius hoste -sense modificar- de diferents plataformes sobre el mateix amfitrió.

Aquest tipus de virtualització és més costosa i la menys eficient, ja que la simulació del comportament d'una plataforma en la capa d'aplicació implica la traducció de cada instrucció al maquinari real. Per tant, no és la més apta per entorns de producció.

Tal vegada l'exemple més destacat d'aquest tipus de programari és Qemu que emula les plataformes x86, x86-64, PowerPC, SPARC o MIPS. D'altra banda, també és molt interessant per la compatibilitat de les seves imatges, ja que es poden portar fàcilment a VirtualBox o a VMware. Altres exemples són: Bochs, MS Virtual Pc, etc.

3.2 Virtualització completa

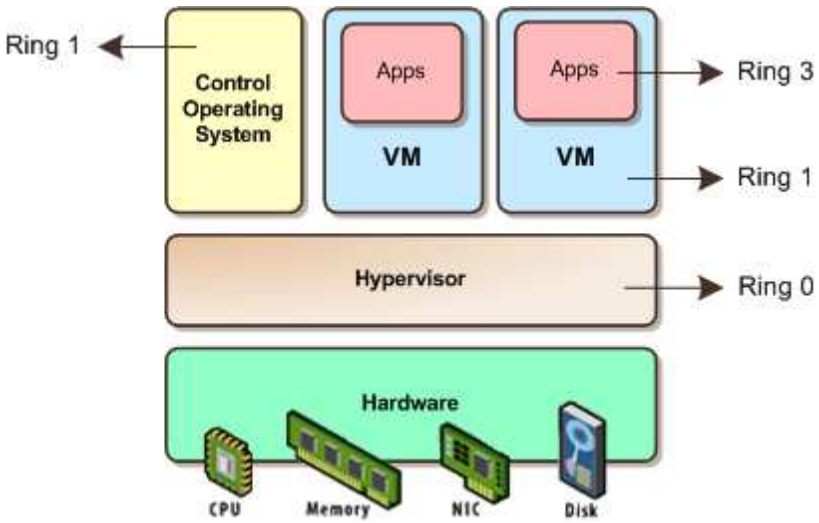
En la virtualització completa, la màquina virtual simula el maquinari suficient per permetre que un sistema operatiu hoste -sense modificar- s'executi de forma aïllada. Aconsegueix un rendiment superior en comparació a l'emulació de maquinari, fent-lo més apte per entorns de producció, tot i que cal que el sistema operatiu hoste sigui compatible amb l'arquitectura del maquinari subjacent.

Aquest tipus de virtualització és la que ha agafat major protagonisme. Permetre que sistemes operatius hoste sense modificar proporciona una plataforma de virtualització més heterogènia que en el cas de la paravirtualització, així és podem virtualitzar, tant sistemes propietaris (p.e. Windows) com lliures (p.e Linux).

3.2.1 Virtualització per traducció binaria

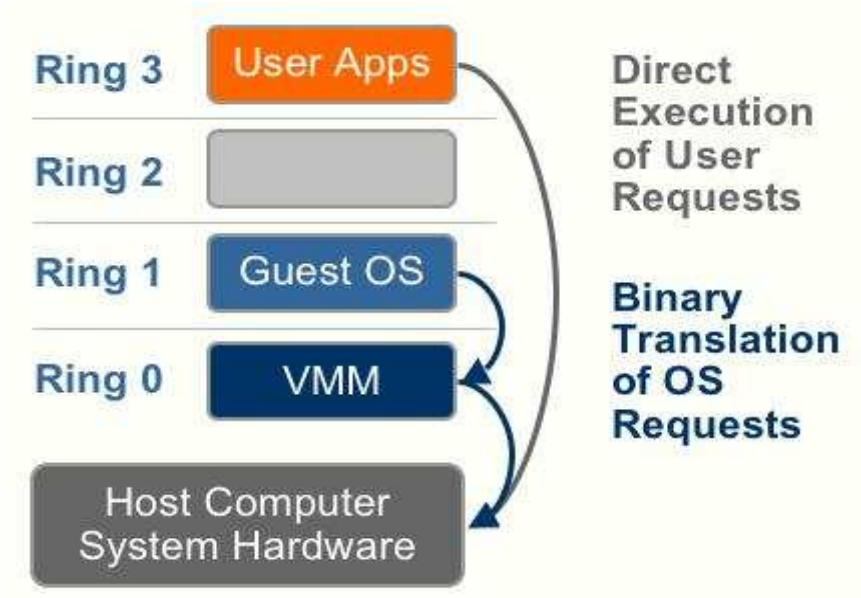
El concepte de virtualització completa, tot i estar ben definit, es va utilitzar d'una manera un poc dubtosa a l'inici de la virtualització en plataformes x86. Originalment l'arquitectura x86 presentava problemes per aquest tipus de virtualització. Tot i així, diferents companyies van desenvolupar hipervisors que, mitjançant una tècnica que combina virtualització parcial i emulació, proporcionen una aparença de virtualització completa.

Aquesta tècnica va ser inicialment desenvolupada per VMware (una de les empreses pioneres en la virtualització de maquinari en plataformes x86) i la va anomenar *binary translation*. En aquesta tècnica, l'hipervisor s'executa a l'anell de més privilegi (*Ring 0*) i els sistemes operatius hoste en un l'anell superior (*Ring 1*) tal com es pot apreciar en la il·lustració 9.



Il·lustració 9: Virtualització completa per traducció binaria

En el seu funcionament, l'hipervisor analitza dinàmicament tot el codi que executa un sistema operatiu hoste, capturant i traduint les instruccions privilegiades (que cal emular) per instruccions vàlides que funcionin al maquinari virtual, mentre que les instruccions de nivell no privilegiat s'executen directament en la CPU real. Es pot apreciar gràficament en la il·lustració 10.

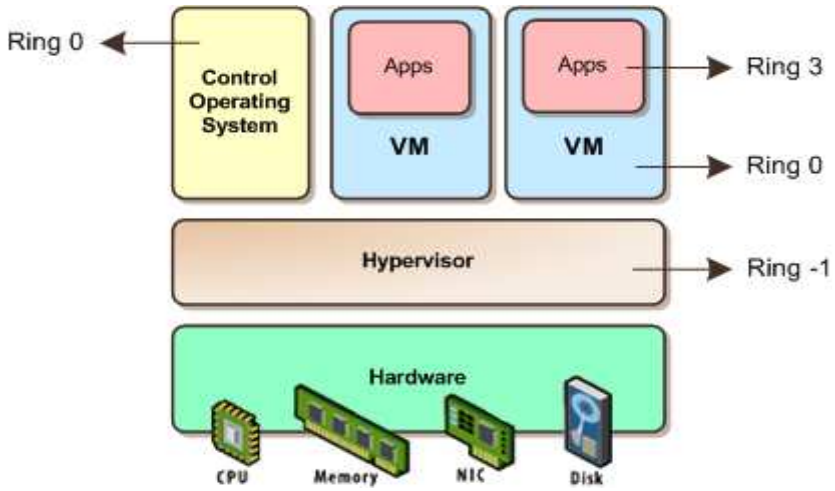


Il·lustració 10: Tècnica de la traducció binaria

Alguns exemples d'hipervisors que utilitzen aquesta tècnica són: Adeos, Mac-on-Linux, Parallels Desktop for Mac, Parallels Workstation, VMware Workstation, VMware Server, VirtualBox, Win4BSD, Win4Lin Pro, etc.

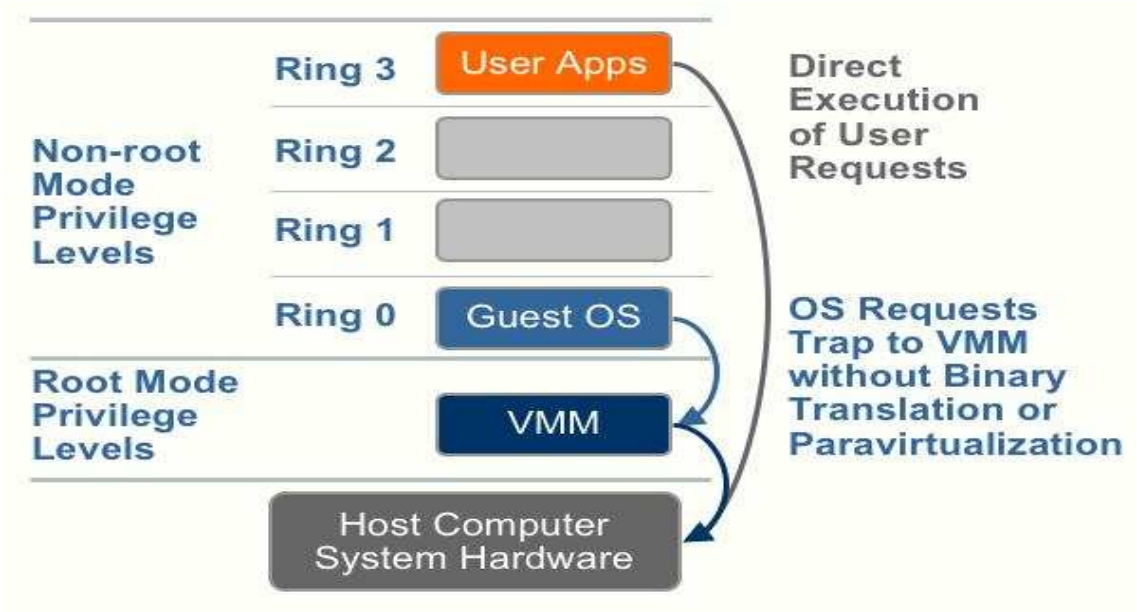
3.2.2 Virtualització completa assistida per maquinari

Posteriorment, les companyies Intel i AMD van incorporar millores en els seus microprocessadors amb les extensions Intel VT-x i AMD-V que proporcionen suport a la virtualització i permeten la virtualització completa. Aquests microprocessadors proporcionen un mode de privilegi addicional per sota de l'anell 0 (*Ring 0*), en el qual opera l'hipervisor, deixant l'anell 0 per als sistemes operatius hoste (Il·lustració 11).



Il·lustració 11: Virtualització completa assistida per maquinari

L'avantatge d'executar els sistemes operatius hoste en l'anell 0 és que ja no cal que l'hipervisor emuli les instruccions privilegiades. Per contra, els sistemes operatius hoste no tenen accés directe als dispositius reals, de manera que la CPU i l'hipervisor han de col·laborar perquè hi puguin accedir. En la il·lustració 12 es pot apreciar aquesta tècnica.



Il·lustració 12: Tècnica de la virtualització completa assistida per maquinari

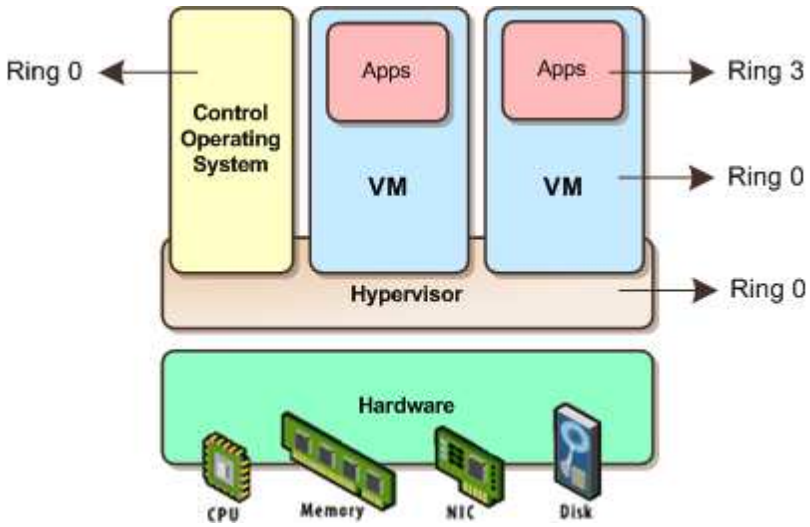
Així, els hipervisors que fan us d'aquestes extensions permeten l'execució de sistemes operatius hoste sense modificar i sense els costos addicionals que suposa la traducció binària i per tant, aconsegueixen un rendiment superior; proper al natiu i similar a la paravirtualització.

Exemples d'hipervisors que han adaptat la virtualització assistida per maquinari són: Linux KVM, VMware Workstation, VMware Server, VMware Fusion, Microsoft Hyper-V, Microsoft Virtual PC, Xen, Parallels Desktop for Mac, VirtualBox, Parallels Workstation, etc.

3.3 Paravirtualització

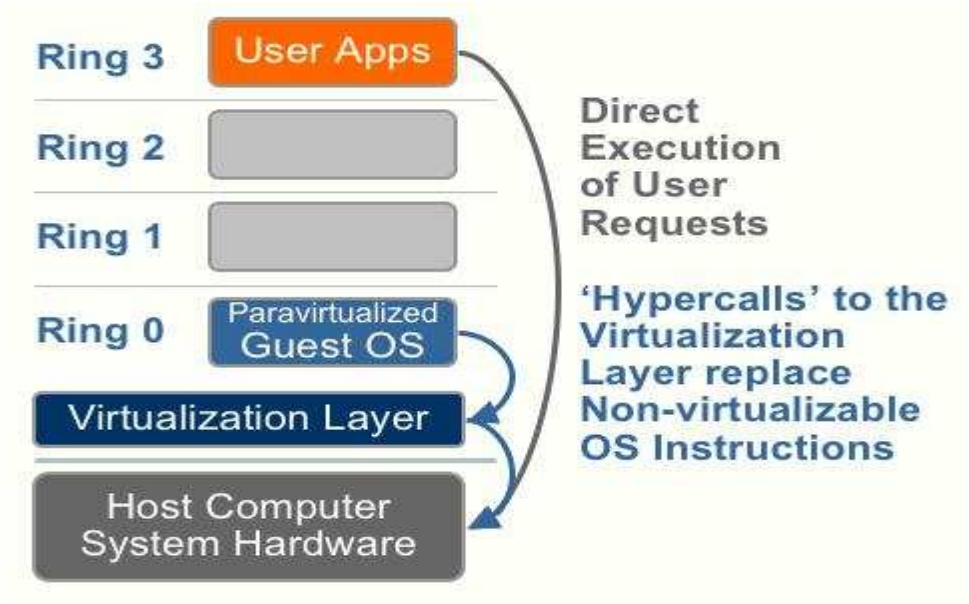
La tècnica de paravirtualització va sorgir com una forma de millorar l'eficiència de les màquines virtuals i apropar-la al rendiment natiu. Per tal de dur a terme aquesta tasca, la màquina virtual no simula el maquinari, sinó que ofereix una interfície especial que només es pot utilitzar mitjançant la modificació del nucli del sistema operatiu hoste.

La modificació del sistema operatiu hoste implica la substitució de totes les operacions privilegiades que corresponen a l'anell 0 (*Ring 0*) de la CPU amb crides a l'hipervisor (conegudes com *hypercalls*). Així, és l'hipervisor qui realitza les tasques en nom del sistema operatiu hoste.



Il·lustració 13: Paravirtualització

D'aquesta manera, no cal que l'hipervisor monitoritzi totes les instruccions que provenen de la màquina virtual, ja que és aquesta qui l'avisa quant ho requereix. Pel que fa a les aplicacions d'usuari (instruccions no privilegiades), s'executen directament sense la mediació de l'hipervisor. La tècnica de la paravirtualització es representa en la il·lustració 14.



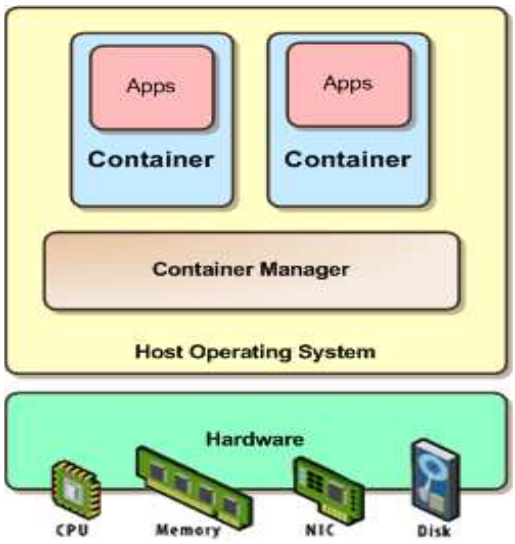
Il·lustració 14: Tècnica de paravirtualització

La principal característica de la paravirtualització és que obté un rendiment proper al natiu en plataformes que no proporcionen suport a la virtualització. Per contra, el requeriment de modificació del sistema operatiu hoste no permet virtualitzar sistemes propietaris que no proporcionen suport a aquesta tècnica (p.e. Windows). *Xen* és l'hipervisor que va popularitzar aquesta tècnica.

3.4 Virtualització a nivell de Sistema Operatiu

La virtualització a nivell de SO proporciona màquines virtuals aïllades mitjançant la creació de particions denominades entorns virtuals (*VEs*) o contenidors, en les quals el sistema hoste comparteix el mateix sistema operatiu que el sistema amfitrió, és a dir, el mateix nucli és utilitzat per implementar l'entorn de l'hoste.

Per dur a terme aquesta tasca, el sistema operatiu ofereix els dispositius físics que controla a cada entorn virtual. D'aquesta manera, les aplicacions veuen l'entorn virtual com un sistema totalment independent.



Il·lustració 15: Virtualització a nivell de S.O.

Els avantatges són un model de virtualització més econòmic, fàcil de gestionar i amb un rendiment quasi natiu, ja que no requereix de suport de maquinari ni tampoc l'emulació d'instruccions de nivell privilegiat. L'únic inconvenient és que només permet l'execució d'entorns virtuals per la mateixa CPU i el mateix sistema operatiu, ja que en realitat només utilitza un nucli.

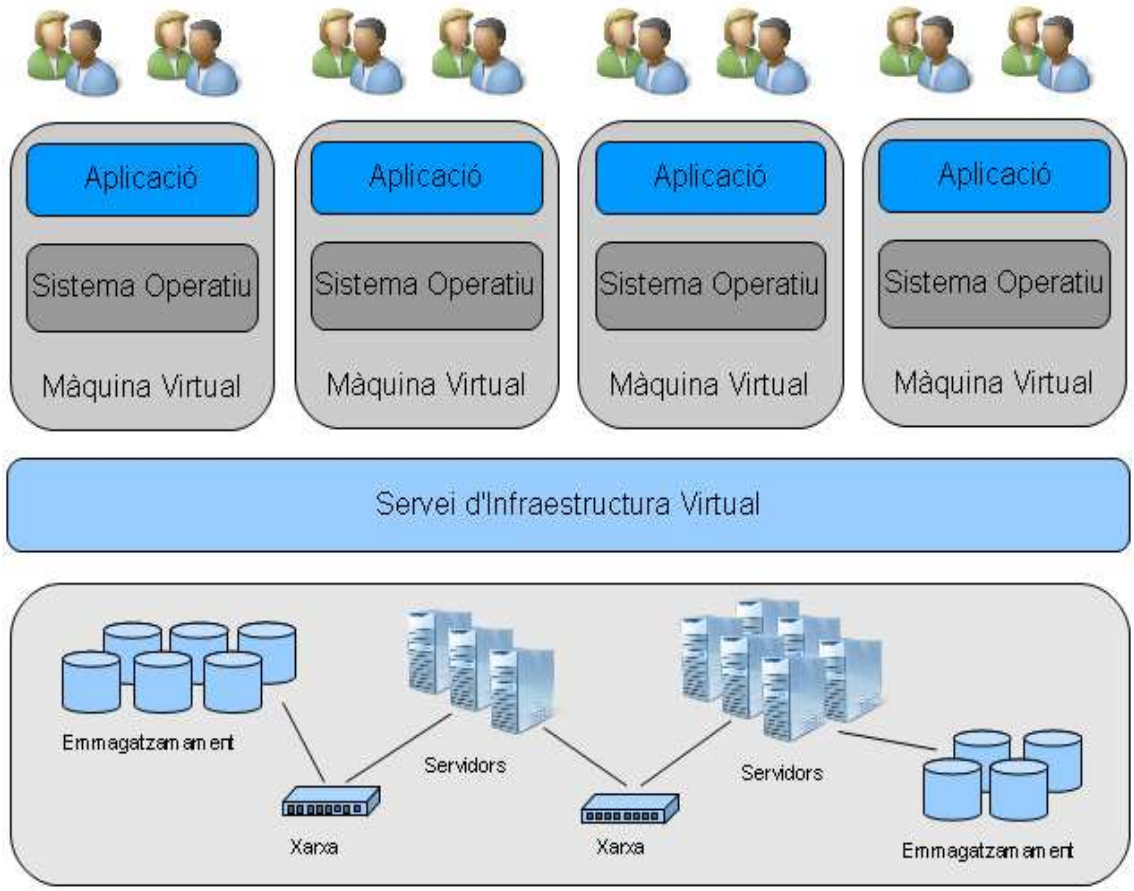
Alguns exemples d'aquest tipus són: Linux-VServer, Virtuozzo, OpenVZ, UML (user mode linux), Solaris Containers i FreeBSD Jails.

4. Virtualització de la infraestructura TI

Els hipervisors proporcionen bàsicament eines per una gestió de manera local. Amb aquestes eines, els usuaris poden administrar els recursos que proporciona cada un dels hipervisors de manera independent. Així, el control i polítiques d'accés, les xarxes virtuals i l'emmagatzemament virtual queden aïllades a nivell d'amfitrió.

L'evolució d'aquestes tecnologies no s'ha limitat únicament a la virtualització d'una computadora. Han aparegut nous programaris que, fent ús dels hipervisors, eleven a una capa superior el nivell d'abstracció i permeten la gestió d'una infraestructura TI com un *pool* de recursos.

Aquesta capa de programari proporciona el *Servei d'Infraestructura Virtual* (*IaaS* o Infraestructura com a Servei). Aquests programaris tenen la capacitat d'abstreure tots els recursos físics d'una infraestructura TI (il·lustració 16) que engloba els servidors de computació, servidors d'emmagatzemament i dispositius de xarxa i així, orquestrar-los i proporcionar-los als usuaris perquè puguin instanciar les seves màquines virtuals.



Il·lustració 16: Servei d'Infraestructura Virtual

Molt sovint, quan es parla de *Infraestructura com a Servei* s'utilitza el terme *Cloud Computing* o Computació en Núvol. Aquest terme, de caràcter més comercial, abstreu encara més el concepte d'infraestructura virtual. De fet, a un usuari d'aquests serveis li interessa la disponibilitat a demanada, l'aprovisionament ràpid, l'escalabilitat dinàmica i pràcticament il·limitada i no li cal conèixer com s'organitza la infraestructura TI.

Els terme de *núvols* és utilitzat per classificar tres tipus d'implementació.

- Núvol públic: es refereix a un model estàndard de computació, en el qual el proveïdor dels serveis disposa els seus recursos, tals com aplicacions i emmagatzemament, al públic en general a través d'Internet. Aquest serveis poden ser lliures o de pagament per ús.

- Núvol Privat: es refereix a una infraestructura TI propietària que utilitza tecnologies de computació en núvol, tals com la virtualització. Es caracteritzen per ser administrades per la pròpia organització a la que donen els serveis.

- Núvol híbrid: és una combinació de les dues anteriors, és a dir, que un núvol privat utilitza serveis d'un núvol públic.

5. Plataformes de Computació en Núvol

En aquest capítol es fa una introducció a tres dels principals programaris basats en GNU/Linux que proporcionen *servei de infraestructura virtual*. Les plataformes que han estat objecte d'estudi són OpenStack, Eucalyptus i OpenNebula.

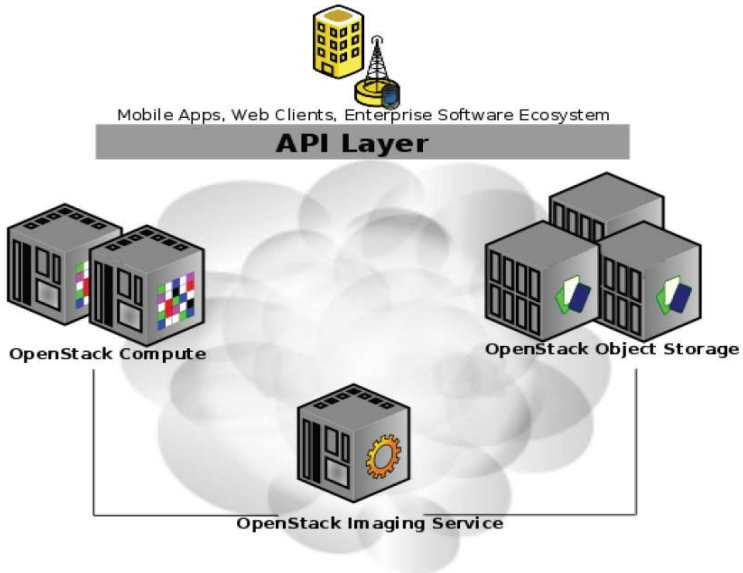
A continuació es fa una introducció a cada una d'aquestes plataformes en la que es descriuen els principals components i característiques. L'objectiu és conèixer les funcionalitats que ofereix cada una de cara a realitzar un estudi comparatiu.

5.1 OpenStack (Diablo)

Consta d'una col·lecció de tecnologies de codi obert que proporcionen un programari altament escalable de computació i emmagatzematge en núvol. OpenStack dona nom a una comunitat, a un projecte i a un programari que proveeix una plataforma operativa formada per un conjunt d'eines amb la finalitat d'orquestrar tant núvols públics com privats.

OpenStack desenvolupa dos projectes relacionats. Per una banda, *OpenStack Compute* ofereix una plataforma de computació i administració de xarxa, per l'altra banda, *OpenStack Object Storage* és un programari que proporciona emmagatzematge d'objectes.

Estretament relacionant amb *OpenStack Compute* es troba el projecte *Image Service* que s'utilitza per el descobriment, enregistrament i serveis d'entrega de les imatges dels sistemes operatius virtuals. En la il·lustració 17 es mostren les relacions bàsiques entre els projectes i com s'interrelacionen.



Il·lustració 17: Projectes OpenStack

5.1.1 OpenStack Compute (Nova)

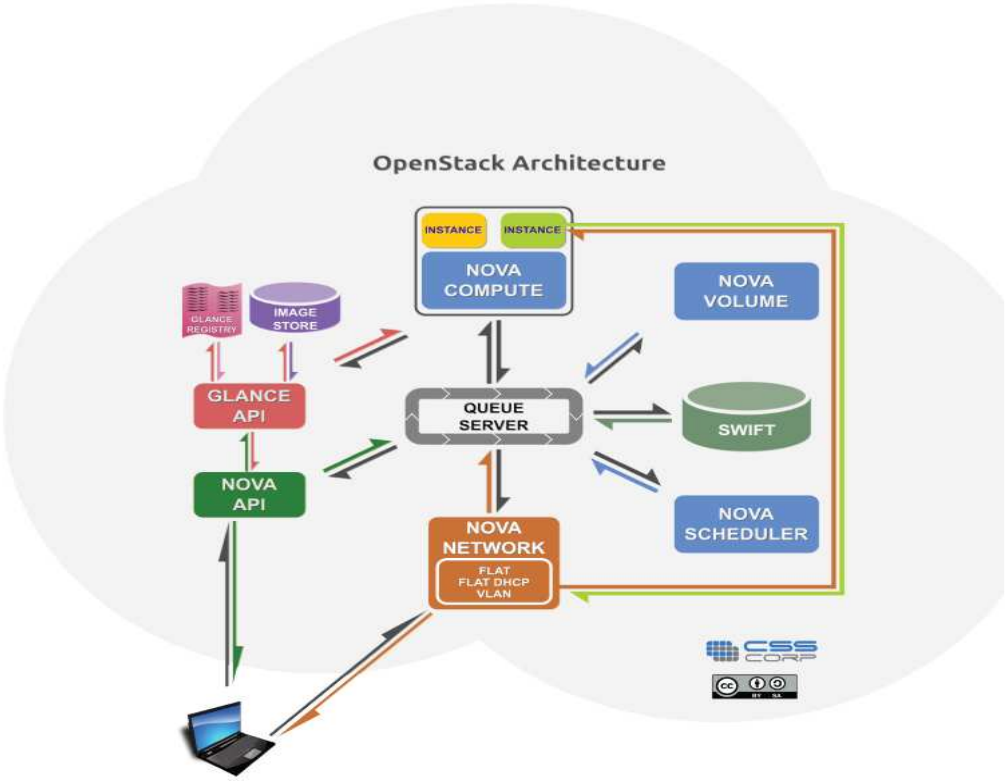
Nova és el principal servei per gestionar la infraestructura de computació i l'encarregat de controlar el cicle de vida de les màquines virtuals que hi ha instanciades. Aquest controla els recursos de computació, la xarxa, l'autorització i les necessitats d'escalabilitat del núvol.

Cal remarcar que *Nova* és una plataforma de gestió i no proporciona cap capacitat de virtualització en sí mateixa, si no que utilitza l'API 'libvirt' per interactuar amb hipervisors compatibles. També proporciona una API de serveis web que és compatible amb la de EC2 de *Amazon Web Services*.

Funcionalitats i característiques:

- Administració del cicle de vida de les instancies
- Administració dels recursos de computació
- Creació de xarxes i autorització
- API basada en REST
- Comunicació asíncrona
- Independència de l'hipervisor

L'arquitectura d'OpenStack es pot observar en la il·lustració 18. En aquesta es mostren com es relacionen tots els seus serveis.



Il·lustració 18: Arquitectura OpenStack

Els principals components (serveis) de *Nova* són:

- Servidor API (*nova-api*): implementa la interfície -i l'únic component- per poder interactuar amb la infraestructura del núvol.

- Controlador del núvol (*rabbit-mq server*): proporciona la comunicació entre els altres components de *Nova* tals com el planificador, el controlador de xarxa i el controlador d'emmagatzemament mitjançant AMQP (Advanced Message Queue Protocol).

- Nodes de computació (*nova-compute*): són els encarregats del cicle de vida de les instàncies de les màquines virtuals. Per tal de dur a terme aquesta tasca, els nodes reben sol·licituds del controlador del núvol. En un entorn de producció el més habitual és disposar de varis nodes de computació, on les instàncies són desplegades en algun dels nodes disponibles mitjançant un algoritme del planificador.

- Controlador de xarxa (*nova-network*): gestiona la configuració de la xarxa en els nodes físics fent les operacions d'assignació d'adreces IP, la configuració de les VLAN dels projectes, l'aplicació dels grups de seguretat i la configuració de xarxes entre els nodes de computació.

- Nodes d'aprovisionament de volums (*nova-volume*): s'utilitzen per l'administració de volums LVM amb les principals funcions de creació, eliminació, connexió i desconnexió d'espai d'emmagatzemament a les instàncies de màquines virtuals, ja que el disc principal, inherent a les instàncies, no és persistent i un cop aquesta és finalitzada no es manté cap canvi. D'aquesta manera, *nova-volume* permet que les dades importants generades durant el cicle de vida d'una instància s'emmagatzemin de forma persistent en un volum, els quals poden ser proporcionats per proveïdors tals com ISCSI o AoE.

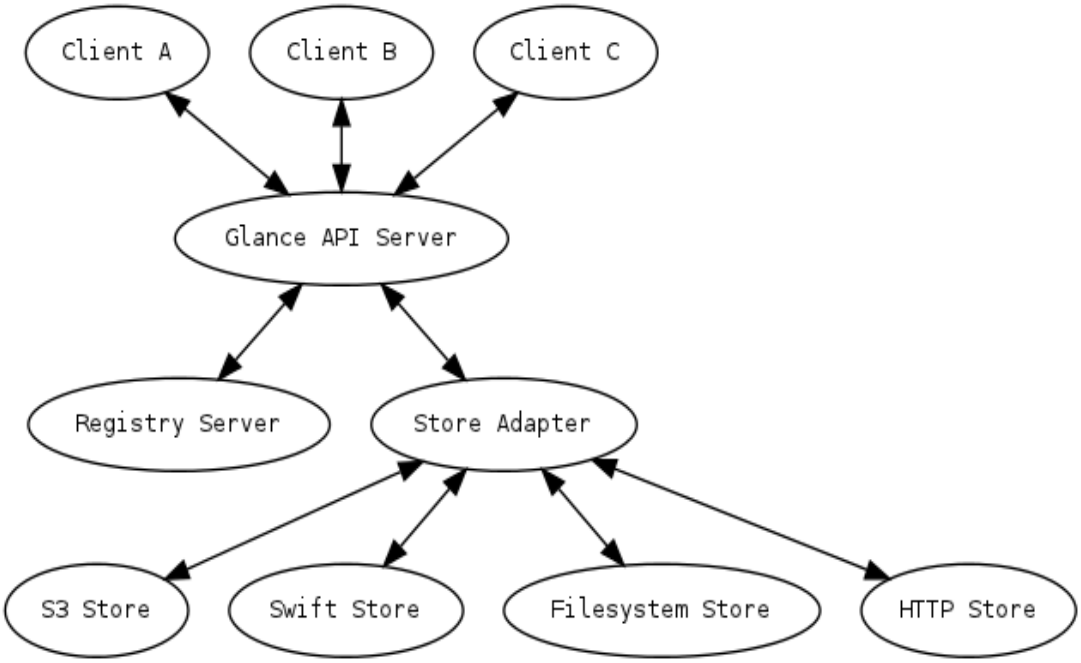
- Planificador (*nova-scheduler*): s'encarrega de determinar en quin node de computació s'executarà una instància de màquina virtual. Això ho fa mitjançant uns algoritmes que basen la seva decisió en varis factors tals com la càrrega, la memòria, localització, arquitectura de la CPU, etc.

5.1.2 OpenStack Image Service (Glance)

OpenStack Image Service és un sistema que proveeix serveis de descobriment, enregistrament i entrega per imatges de màquines virtuals. Per dur a terme aquesta tasca implementa una API que permet enregistrar noves imatges, consultar les que hi ha enregistrades i utilitzar la llibreria del client per transmetre-les.

Addicionalment aquest servei pot ser configurat per utilitzar diversos sistemes d'emmagatzemament tals com *S3 Storage*, *OpensSack Object Storage*, etc. En el

següent diagrama es mostra la relació que hi ha entre els seus components i els sistemes d'emmagatzematge que s'hi poden connectar.



Il·lustració 19: Components i sistemes d'emmagatzemament de Image Service

5.1.3 OpenStack Storage (Swift)

Object Storage és un programari de codi obert per la creació d'un emmagatzemament d'objectes que està pensat per formar *clusters* de servidors standard i arribar a allotjar fins a *petabytes* d'informació. A més, al ser un sistema distribuït i no tenir un punt de control principal, proporciona un alt nivell d'escalabilitat i redundància.

Tot i així, aquest no és un sistema de fitxers ni un sistema d'emmagatzemament en temps real, si no més bé, un sistema a llarg termini per dades majoritàriament estàtiques i que només s'actualitzen eventualment. Així, els principals exemples de dades que millor s'hi adapten són les imatges de màquines virtuals, la fotografia digital, l'emmagatzemament de correu electrònic i còpies de seguretat.

Els seus components són:

- *Swift Proxy Server* és l'encarregat de rebre les peticions de l'exterior, del control d'accés, de la gestió de la memòria cau del sistema i del control de la disponibilitat de cada node d'emmagatzemament. A part d'això, té coneixement de la localització de les entitats apropiades, i així, re-adreçar les peticions dels clients al node corresponent.

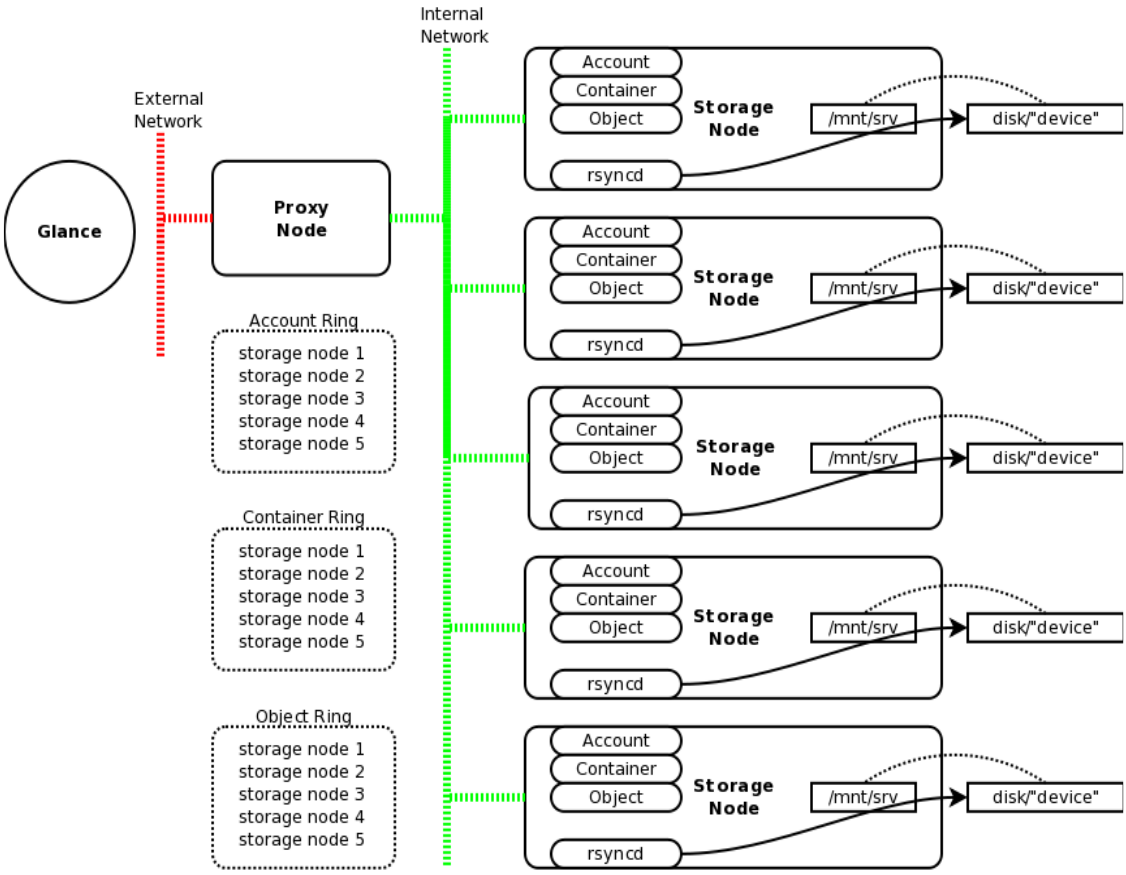
- *Swift Object Server* és el responsable de manejar l'emmagatzemament local, és a dir, recuperar i eliminar objectes del sistema d'arxius que utilitza el node. Aquets objectes són típicament fitxers binaris que contenen metadades com atributs estesos d'arxiu (xattr).

- *Swift Container Server* enumera els objectes en un contenidor i també realitza un seguiment estadístic, com el nombre d'objectes que conté i l'espai ocupat pel contenidor. Per tal de mantenir aquesta informació utilitza fitxers SQLite.

- *Swift Account Server* enumera els contenidors de la mateixa manera que el servidor de contenidors enumera els objectes.

- El *Ring* és l'anell lògic que conté la informació referent a la ubicació física dels objectes emmagatzemats a l'interior de Swift. El seu funcionament és similar a un servei d'indexació que utilitza diversos processos per cercar i localitzar la ubicació física real de les entitats (comptes, contenidors i objectes) dins del cluster.

En la il·lustració 20 es mostra , a tall d'exemple, com es distribueixen a nivell físic i lògic tots els components de Swift en una implementació amb cinc nodes.



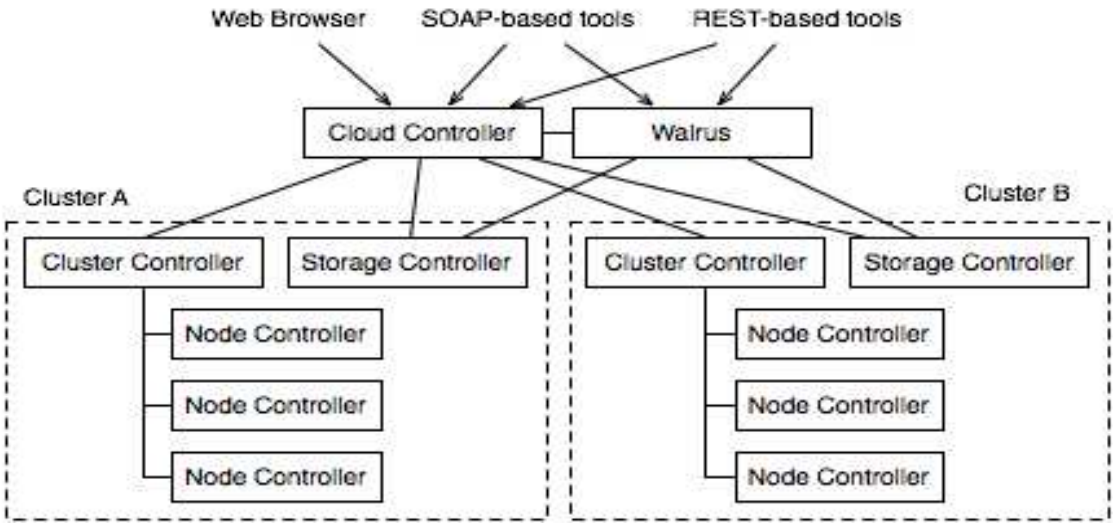
Il·lustració 20: Exemple del sistema Swift amb 5 nodes d'emmagatzemament

5.2 Eucalyptus 2.0

Eucalyptus és un programari lliure basat en GNU/Linux. Aquest implementa una arquitectura de programari escalable que millora l'eficiència de núvols privats i híbrids dins la infraestructura de TI d'una organització. Eucalyptus proporciona infraestructura com a servei (*IaaS*).

Una configuració del núvol d'Eucalyptus es compon de cinc tipus de components. Per una banda, el controlador del núvol (*Cloud Controller*) i el controlador d'emmagatzemament (*Walrus)* són els components frontals i en calen un de cada classe en una instal·lació en núvol. Per l'altra banda, *Cluster controller*, *Storage Controller* i *Node Controler.*

Per tal de conèixer com s'interrelacionen els components d'Eucalyptus (il·lustració 21) i quines son les funcionalitats que proporciona cada un d'ells, es descriuen de forma més detallada.



Il·lustració 21: Disposició en dues zones de la plataforma Eucalyptus

Cluster Controller (CC): CC gestiona un o més controladors de node (NC) i és l'encarregat d'instanciar i controlar les instàncies de màquines virtuals que hi ha en ells. També gestiona la xarxa de les instàncies que s'executen en els nodes . Per una banda el CC es comunica amb el controlador del núvol (CLC), per l'altra banda amb els controladors de node (NC). Les seves funcions:

- Rebre les sol·licituds del CLC per desplegar instàncies
- Decidir en quin NC es desplegarà la instància
- Controlar la xarxa virtual a disposició de les instàncies
- Recopilar informació sobre els NC i informar-ne al CLC

Node Controller (NC): s'executa en cada node i controla el cicle de vida de les instàncies que s'executen en ell. Aquest interactua amb el sistema operatiu i l'hipervisor que té al seu costat. Així, és com un agent que rep les ordes del controlador del *clúster* (CC). Les seves funcions:

- La recopilació de dades relacionades amb la disponibilitat i utilització de recursos en el node i la notificació de les dades al CC.
- Gestió del cicle de vida de les instàncies.

Cloud Controller (CLC): El controlador del núvol (CLC) és l'extrem davanter que integra la infraestructura de núvol. Per una banda, ofereix -a part de les eines de línia de comandes- dues interfícies web, una compatible amb els serveis EC2/S3 i una altra per la gestió de certs aspectes de la infraestructura local. Per l'altra banda interactua amb la resta dels components de la infraestructura d'Eucalyptus. Les seves funcions:

- Monitoritzar la disponibilitat de recursos en els diferents components de la infraestructura en el núvol
- Arbitratge de recursos: decideix sobre quins *clusters* seran utilitzats per a l'aprovisionament de les instàncies
- Monitoritzar les instàncies en execució

Walrus Storage Controller (WS3): WS3 ofereix un servei d'emmagatzematge persistent utilitzant APIs REST i SOAP compatible amb l'API de S3. WS3 ha de ser considerat com un sistema d'emmagatzematge d'arxius simple. Les seves funcions:

- Emmagatzemar imatges de les màquines virtuals
- Emmagatzemar instantànies
- Emmagatzemar i servir fitxers utilitzant l'API S3

Storage Controller (SC): SC ofereix emmagatzematge de blocs persistent per proporcionar a les instàncies. Aquest és similar a *Elastic Block Storage* (EBS) servit per AWS. Les seves funcions:

- Creació de volums persistents
- Proporcionar els volums a les instàncies (amb el protocols AoE o iSCSI)
- Possibilitar la creació de instantànies dels volums

5.3 OpenNebula 3.0

OpenNebula és un programari que proporciona Infraestructura virtual (Iaas), és una solució completa per la gestió de núvols de computació de centres de dades privats, públics i híbrids. OpenNebula s'integra aprofitant els actius TI dels centres de dades.

OpenNebula orquestra emmagatzematge, xarxes, virtualització, monitoratge i tecnologies de seguretat per permetre la col·laboració dinàmica dels serveis que

proporciona sobre les infraestructures de distribució, combinant els recursos del centre de dades i recursos de núvols remots.

OpenNebula proporciona una plataforma escalable i segura de computació en núvol per un ràpid i dinàmic aprovisionament de recursos virtuals.

- El sistema *Image Repository* permet configurar i compartir imatges, que tant poden contenir sistemes operatius com dades, per ser utilitzades fàcilment per les màquines virtuals.

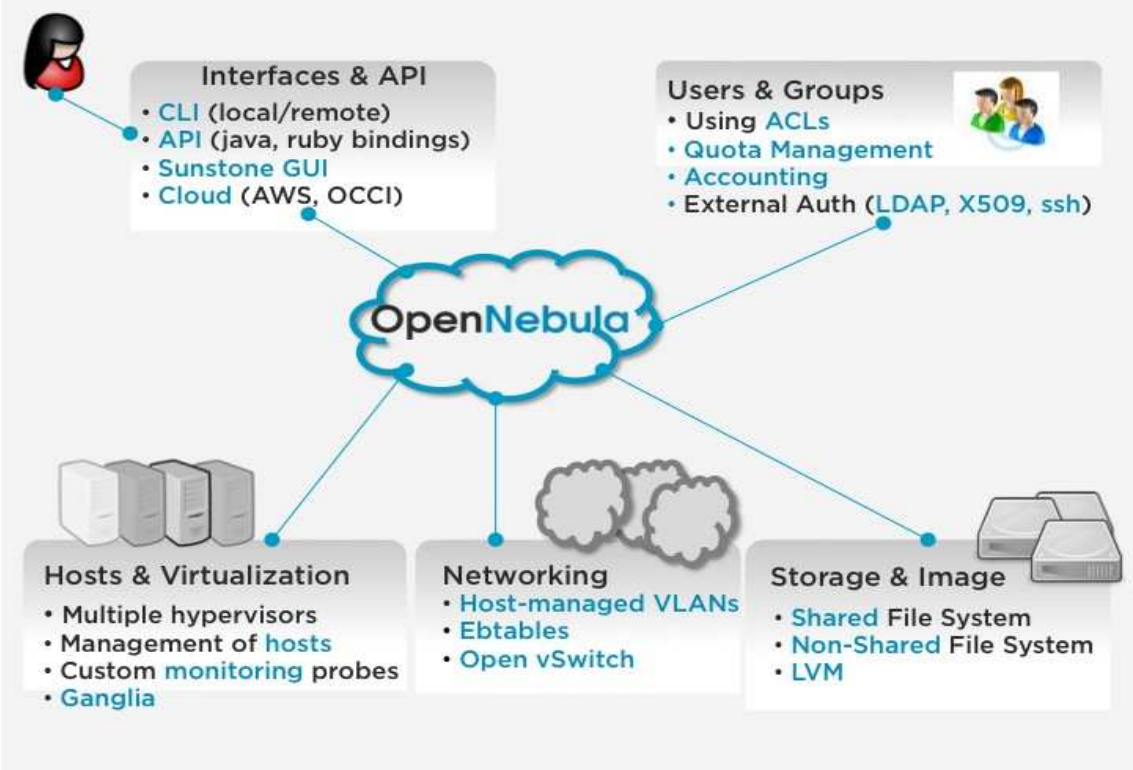
- El sistema *Template Repository* permet enregistrar definicions de Màquines Virtuals que posteriorment podran ser instanciades.

- *Virtual Networking* gestiona la interconnexió de xarxa de les Màquines Virtuals.

- La interfície de línia de comandes d'usuari CLI i consola web *SunStone* proporcionen les eines per controlar el cicle de vida de les instàncies de les Màquines virtuals.

5.3.1 Components Principals

En aquest apartat, es descriuen els principals components d'OpenNebula que es poden veure a la il·lustració 22.



Il·lustració 22: Components d'OpenNebula

Interfícies i API : OpenNebula ofereix moltes interfícies diferents que es poden utilitzar per administrar els recursos físics i virtuals. Així, proporciona dues interfícies principals, una mitjançant la línia de comandes i l'altre mitjançant una interfície web anomenada *SunStone* . També pot utilitzar les interfícies OCCI i EC2 per crear núvols públics. L'API d'OpenNebula ofereix funcions d'integració per permetre el fàcil desenvolupament de nous components (p.e. nous controladors per donar suport a hipervisors).

Usuaris i grups : OpenNebula suporta comptes d'usuaris i grups , així com varis mecanismes d'autenticació i autorització. Aquesta característica es pot utilitzar per crear compartiments estancs dins del mateix núvol, implementant el multi-lloguer . D'altra banda , el mecanisme de control d'accés permet assignar als usuaris diferents rols d'administració.

Xarxes : Un subsistema de xarxa fàcilment adaptable i personalitzable és present en OpenNebula per tal d'integrar millor als requisits específics de la xarxa de centres de dades existents . També ofereix suport per VLANs i Open vSwitch.

Servidors i Virtualització : diversos hipervisors són compatibles amb l'administrador de virtualització amb la capacitat de controlar el cicle de vida de les màquines virtuals així com el seu seguiment. Aquesta monitorització s'aplica també als servidors físics.

Emmagatzematge i imatges : OpenNebula pretén ser prou flexible per a oferir tantes configuracions diferents d'emmagatzemament d'imatges com sigui possible. El subsistema d'emmagatzematge pot ser configurat per suportar sistemes d'arxius no compartits i compartits, així com una àmplia gamma de diferents disposicions dels servidors d'imatges.

Aquest subsistema es pot implementar mitjançant un gestor de transferències específic que integra el sistema d'arxius MosseFS per tal de proporcionar emmagatzemament compartit a la plataforma. A l'apartat *Sistema d'arxius MooseFS 1.6.20* es fa una introducció a aquest sistema.

5.3.2 Components Avançats

Múltiples zones o VDC: El component OpenNebula zones (oZones) permet la gestió centralitzada de múltiples instàncies de OpenNebula, anomenades Zones. A la vegada, aquestes zones poden ser dominis administratius diferents. A més, una zona pot ser més compartimentada mitjançant l'agrupació de computadores físiques en centres de dades virtuals (VDC).

Híbrids : OpenNebula dóna suport per construir un núvol híbrid. Això consta d'una extensió del núvol privat per tal de combinar els recursos locals amb recursos dels proveïdors de núvols a distància. Tot el núvol públic d'un proveïdor pot ser encapsulat com un recurs local i així, poder utilitzar les capacitats de còmput addicionals per satisfer pics de demanda.

5.3.3 Sistema d'arxius MooseFS 1.6.20

MooseFS és un sistema d'arxius distribuït amb tolerància a fallades i altament escalable. Aquest sistema dispersa les dades que gestiona a través de varis servidors físics, tot i que els usuaris el veuen com un únic recurs. Així, la interfície de MooseFS actua com un sistema d'arxius *Unix*:

- Disposa d'una arquitectura jeràrquica (arbres de directoris)
- Gestiona atributs de fitxers (permisos, data de modificació, darrer accés)
- Suporta fitxers especials (dispositius de bloc i caràcters, etc)
- Suporta enllaços simbòlics i enllaços fixos
- Controla l'accés tant per adreça IP com per contrasenya

Pel que fa a la funcionalitat:

- Garanteix una alta disponibilitat, ja que el sistema és capaç de mantenir varies còpies de les dades en diferents servidors.
- També és altament escalable, es poden afegir en calent tant servidors com espai d'emmagatzemament per augmentar de forma dinàmica la capacitat del sistema.
- Disposa d'un sistema de 'paperera de reciclatge', en concret, el sistema permet configurar un període de temps en que els fitxers eliminats seran retinguts.
- Permet crear instantànies (*snapshots*) coherents dels fitxers, tot i que el fitxer estigui en ús.

L'arquitectura de MooseFS (veure il·lustració 23) disposa dels següents components.

Servidor principal (*Master Server*): és el responsable de gestionar el sistema de fitxers i de mantenir la integritat. Aquest emmagatzema les metadades per cada fitxer (mida, atributs i localització, incloent tota la informació de fitxers no regulars com: directoris, *sockets*, *pipes* i dispositius). Tots els altres components del clúster es comuniquen amb aquest servidor a través dels ports de xarxa especificats.

MooseFS no suporta redundància activa per el servidor principal i és aquí on rau el seu punt crític, ja que el sistema manté tota la informació de metadades en memòria (RAM) per tal de dotar al sistema de la màxima velocitat. Així , una caiguda d'aquest servidor comportaria deixar fora de servei tot el sistema de fitxers. Per aquest motiu, és altament recomanable disposar d'un servidor de backup del servidor principal.

D'aquesta manera, restaurar el servidor principal a partir del servidor de backup seria ràpid i fàcil. Tot i així, és aconsellable que el servidor que gestiona el sistema de fitxers es munti en un equip amb nivells de fiabilitat més elevats (tals com RAID-1 o superior, memòria ECC, doble font d'alimentació, etc) per tal de garantir la màxima disponibilitat.

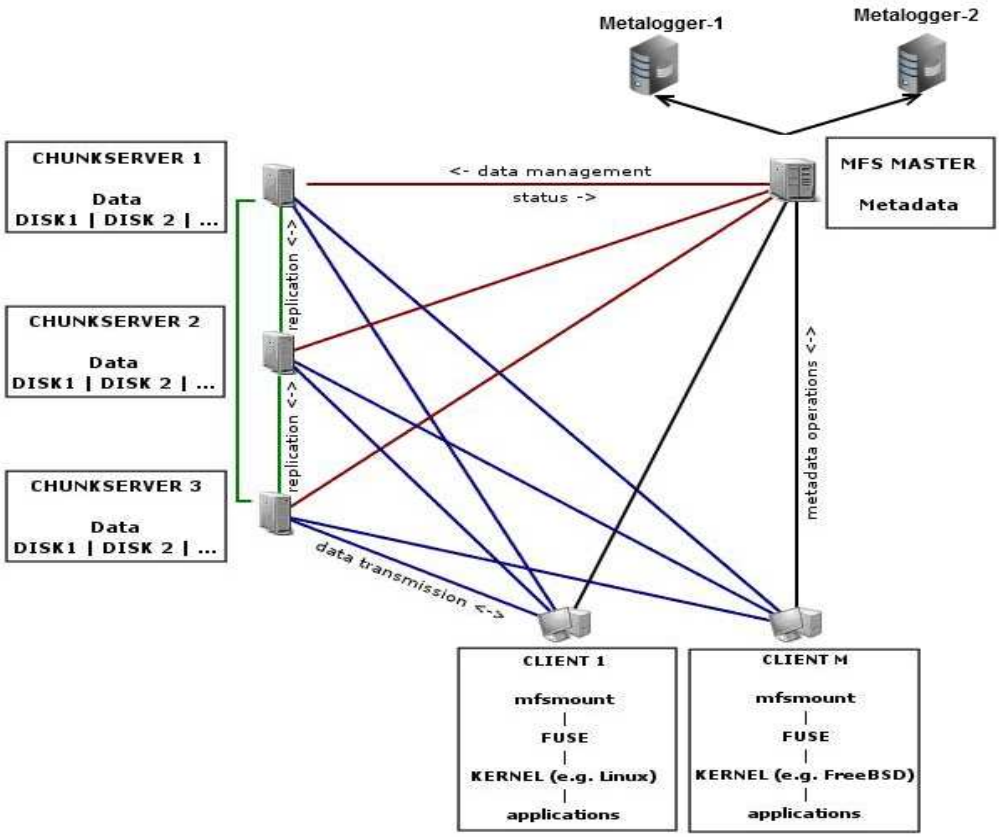
Servidor de backup de metadades (*Metalogger Server*): són opcionals però altament recomanables per la recuperació davant una catàstrofe. Aquests servidors importen cada cert temps i de forma passiva els fitxers de metadades des de el servidor principal.

D'aquesta manera es pot programar que un servidor de backup, agafi el control i es restauri com a servidor principal en el cas que aquest deixi de funcionar.

Servidors de dades (*Chunk Servers*): són els que emmagatzemen les dades del sistema d'arxius. Aquests servidors poden ser afegits i eliminats del clúster en qualsevol moment. Així, afegint nous servidors de dades, es pot escalar horitzontalment tant en capacitat d'emmagatzematge com en fiabilitat. També, si s'han configurat suficients rèpliques (anomenades *goals*), un servidor de dades es pot eliminar, reiniciar o aturar sense perdre la disponibilitat del sistema de fitxers.

Equips Client (*msfmount*): que accedeixen al sistema de fitxers - mitjançant el driver FUSE. Així, MooseFS esta disponible per cada Sistema Operatiu que utilitza aquest driver (Linux, FreeBSD, MacOS X, etc). Qualsevol nombre de clients poden utilitzar *mfsmount* per connectar-se amb el servidor principal (operacions d'intercanvi de metadades) i amb el servidors de dades (operacions d'intercanvi de dades).

Servidor d'estat CGI: que proporciona una interfície gràfica basada en web per tal de visualitzar l'estat del sistema de fitxers i dels seus components. Aquest component (opcional) pot ser instal·lat de dues maneres diferents: en un equip qualsevol, ja que proporciona un servidor *cgi httpd* bàsic; o també en un servidor web *Apache*. D'aquesta manera, un únic servidor d'estat CGI és suficient per obtenir la informació d'un o varis servidors principals per mitjà de diferents arguments URL .



Il·lustració 23: Arquitectura de MooseFS

Les operacions de fitxers en un client que utilitza MooseFS són exactament les mateixes que per altres sistemes de fitxers. El nucli del sistema operatiu transfereix les operacions al mòdul FUSE, el qual es comunica amb el procés *mfsmount*, que a la vegada es comunica per xarxa amb el servidor principal i amb els servidors de dades. Tot aquest procés és completament transparent per l'usuari.

Els equips client, per una banda, consulten el servidor principal cada vegada que realitzen les operacions de creació i eliminació de fitxers, en l'inici de lectures o escriptures de dades, lectura de directoris, lectura i canvi d'atributs, etc. Per l'altra banda, es connecten directament amb cada servidor de dades que conté o contindrà la informació (porció de dades) rellevant que el client requereix.

A més, els servidors de dades es comuniquen entre ells per tal de mantenir les rèpliques de fitxers o directoris que s'ha establert, ja que és aquí on es garanteix l'alta disponibilitat. Quan un node deixa d'estar disponible, el client és re-adreçat a un altre node que conté una rèplica de la informació.

6. Comparativa de les plataformes

En aquest apartat es comparen amb detall els aspectes més fonamentals de les tres plataformes posant especial èmfasi en la disponibilitat i escalabilitat de la infraestructura virtual per aplicar en un entorn de núvol privat.

6.1 Computació bàsica

Les plataformes de computació en núvol analitzades segueixen la mateixa política pel que fa a la forma en que s'instancien les màquines virtuals. En el seu funcionament per defecte quan una imatge és instanciada es còpia del repositori d'imatges al node de computació on s'executarà, i un cop finalitza el seu cicle de vida, la instància s'elimina del node.

Aquest funcionament té les següents avantatges. Per una banda, permet que una imatge pugui ser instanciada tantes vagades com es vulgui, ja que la imatge original del repositori no es modifica. Per l'altra banda, només utilitza elevats recursos de xarxa en el moment de ser instanciada, és a dir, en el moment en que la imatge es transfereix del repositori al node de computació. D'aquesta manera les instàncies utilitzen els recursos de disc locals del node de computació, i per tant, no calen sistemes addicionals NAS o SAN d'alt rendiment d'I/O per allotjar totes les instàncies que hi ha en execució en la plataforma.

Així, quan un node de computació queda fora de servei, les instàncies de màquines virtuals que aquest tenia en producció deixen d'estar disponibles, però es poden tornar a instanciar en un altre node a partir del repositori. D'aquesta manera es garanteix l'alta disponibilitat en la computació, tot i que calen mecanismes addicionals si es volen conservar les dades generades per les instàncies.

6.2 Connexió a volums (Persistència)

El funcionament anterior, no garanteix la persistència de les dades generades per la instància, ja que, quan s'acaba el seu cicle de vida és eliminada del node de computació. A més, davant la caiguda d'un node es poden donar els següents casos:

- El node de computació es pot recuperar: En tal cas, les instàncies que tenia en execució també es podran recuperar i no es perdran els canvis que s'hi hagin fet. Tot i així, per restaurar-les caldrà una intervenció manual o programada que comportarà un determinat temps (depenent del tipus d'averia del node) en que les instàncies estaran fora de servei.

- El node no es pot recuperar: En aquest cas, les instàncies que estaven en execució tampoc es podran recuperar i es perdran els canvis que s'hi hagin fet.

Aquest fet comporta disposar de mecanismes alternatius perquè les dades importants generades durant el cicle de vida d'una instància s'emmagatzemin de forma persistent. Tant OpenStack com Eucalyptus implementen els seus propis components, *nova-volume* i *Block Storage* respectivament. OpenNebula no l'implementa.

Tots dos components intenten resoldre el problema mitjançant la connexió de volums LVM basats en Linux . Així, les instàncies que executen el sistema operatiu Linux permeten connectar aquests volums que poden ser proporcionats per diferents proveïdors de SAN tals com ISCSI o AoE.

Pel que fa a les instàncies que executen sistemes Windows no els poden utilitzar. Aquests per disposar de volums addicionals ho hauran de fer a nivell d'instància i connectar directament amb el proveïdor de SAN. Tanmateix, aquest mecanisme proporciona persistència per les dades que hi ha en els emmagatzemades en els volums però no per les instàncies en si.

Les instancies amb sistemes operatius Linux, davant la caiguda d'un node de computació, es podran restaurar en un altre node i s'hi podran tornar al connectar el volum o volums persistents de que disposava.

Cal remarcar que OpenNebula integra un mecanisme que permet implementar la persistència, és a dir, té la capacitat de configurar, mitjançant un atribut de la instància, si una imatge de màquina virtual ha de conservar els canvis o no. Així es com ho venen fet típicament els hipervisors.

6.3 Alta disponibilitat (migració d'instàncies)

La utilització d'un sistema d'arxius compartit entre els nodes de computació permet dotar a la plataforma d'una tolerància a fallades més elevada, ja que les imatges instanciades no s'allotgen en el propi node, si no que estan allotjades en el sistema d'arxius que comparteixen entre ells.

Cal remarcar, que el sistema arxius a utilitzar, també ha de proporcionar prou fiabilitat. De no ser així, s'afegeix un nou punt crític a la infraestructura i és preferible no implementar-lo.

Amb l'ús d'un sistema d'arxius distribuït, tots els nodes de computació tenen les instàncies al mateix lloc. Així, davant la pèrdua d'un node, es poden recuperar les seves instàncies tant per restaurar-la en el mateix node com en un altre. Aquest sistema també es pot utilitzar com a repositori de les imatges de les màquines virtuals.

D'aquesta manera, no tan sols es millora la tolerància a fallades, tant OpenStack com OpenNebula permeten fer migració en calent, és a dir, poden migrar una instància d'un node de computació a un altre sense que s'aturi de donar servei. Això afegeix fiabilitat a la plataforma, ja que permet deixar lliure un node de computació per poder-li realitzar qualsevol tasca de manteniment, ampliació o per retirar-lo del servei.

En aquest apartat, es pot destacar OpenNebula que implementa un gestor de transferències per el sistema d'arxius MooseFS per aconseguir emmagatzematge compartit per als seus amfitrions. Tant OpenStack com Eucaliptus no integren cap ecosistema d'aquest tipus.

6.4 Múltiples zones

Fins aquí, s'ha vist com aquestes plataformes proporcionen mecanismes per garantir l'alta disponibilitat davant l'eventual pèrdua d'algun node de computació, però es pot donar el cas que es perdi tot el centre de dades on hi havia els nodes de computació, ja sigui per incendi, inundació, tal elèctric, etc.

Les plataformes estudiades permeten la distribució dels seus serveis entre diferents zones. Així, davant la pèrdua d'alguna d'aquestes, les màquines virtuals que han quedat fora de servei podran ser instanciades de nou en els nodes d'una altra zona. Tot i així, caldrà tenir en compte les següents consideracions:

- En el cas del funcionament bàsic, totes les zones han de tenir accés al repositori de les imatges de les màquines virtuals o bé a una rèplica.

- Tant si s'utilitzen volums per la persistència, com un sistema d'arxius compartit, o ambdós, també caldrà que les zones disposin d'accés als servidors físics que els proporcionen o bé a una rèplica.

6.5 Escalabilitat

Pel que fa a la escalabilitat en la computació, les tres plataformes la garanteixen de dues formes: mitjançant l'addició de nous nodes a una zona; i també, mitjançant l'addició de noves zones. Cal destacar que OpenNebula i Eucalyptus disposen de components específics, *Zones* i *Cloud Controler* respectivament, per unificar la gestió de zones aïllades.

Les tres plataformes estan preparades per proporcionar emmagatzemament escalable a la infraestructura virtual. Tot i així, l'escalabilitat en la infraestructura física d'emmagatzemament dependrà del proveïdor que s'utilitzi (fàbriques SAN, sistemes d'arxius compartits, etcètera).

6.6 Compatibilitat amb hipervisors

Un altre aspecte important d'aquestes plataformes és la capacitat de poder orquestrar diferents hipervisors. Cada una d'elles proporciona compatibilitat amb els següents hipervisors.

- **OpenStack**: Hyper-V 2008, KVM - Kernel-based Virtual Machine, LXC - Linux Containers (through libvirt), QEMU - Quick EMUlator, UML - User Mode Linux, VMWare ESX/ESXi 4.1 update 1, Xen - XenServer 5.5/Xen Cloud Platform (XCP)

- **Eucalyptus**: Xen i KVM

- **OpenNebula**: Xen i KVM, Vmware ESX, VMware Server

6.7 Interfície gràfica

La interfície gràfica es una part fonamental de cara a tenir una visió general tota la infraestructura virtual. Les tres plataformes proporcionen una interfície d'aquest tipus basada en web, enfocades principalment a l'usuari final dels recursos.

Aquestes interfícies, de manera general, permeten el control del cicle de vida de les instàncies i control d'accés als usuaris. Tot i així, OpenNebula va una passa més enllà en comparació amb les altres i proporciona una funcionalitat més elevada per l'administració del núvol:

- Permet l'addició de nous nodes de computació
- Mostra les característiques de maquinari i l'hipervisor de cada node
- Monitoritza l'ús de CPU, memòria i xarxa de cada node de computació
- Permet l'adició de noves imatges a la plataforma
- Permet la creació de plantilles de màquines virtuals
- Monitoritza l'ús de CPU, memòria i xarxa de cada instància

7. Implementació del núvol privat

Un cop s'han comparat les tres plataformes caldrà veure quina s'adapta millor a un entorn de computació en núvol privat. Així, en aquest capítol es descriuen les principals raons per l'elecció del programari que proporcionarà el servei d'infraestructura virtual a la futura implementació.

7.1 Justificació del programari.

Després d'estudiar en profunditat els hipervisors i les diferents plataformes de computació en núvol, s'ha decidit implementar-lo amb el programari OpenNebula. Aquest farà ús de l'hipervisor KVM per la virtualització dels nodes de computació i del el subsistema *TM MosseFS* per proporcionar emmagatzemament compartit a la infraestructura. Principalment per les següents raons:

- OpenNebula proporciona una plataforma d'assimilació més simple. Tot i que disposa de menys components que les altres, els subsistemes que integra permeten múltiples configuracions.

- OpenNebula permet una infraestructura mixta d'hipervisors. Així, durant el procés de migració, permetrà gestionar de manera centralitzada *VMware Server* i *KVM*.

- OpenNebula permet la persistència de la imatge (tal com ho fa VMware per defecte).

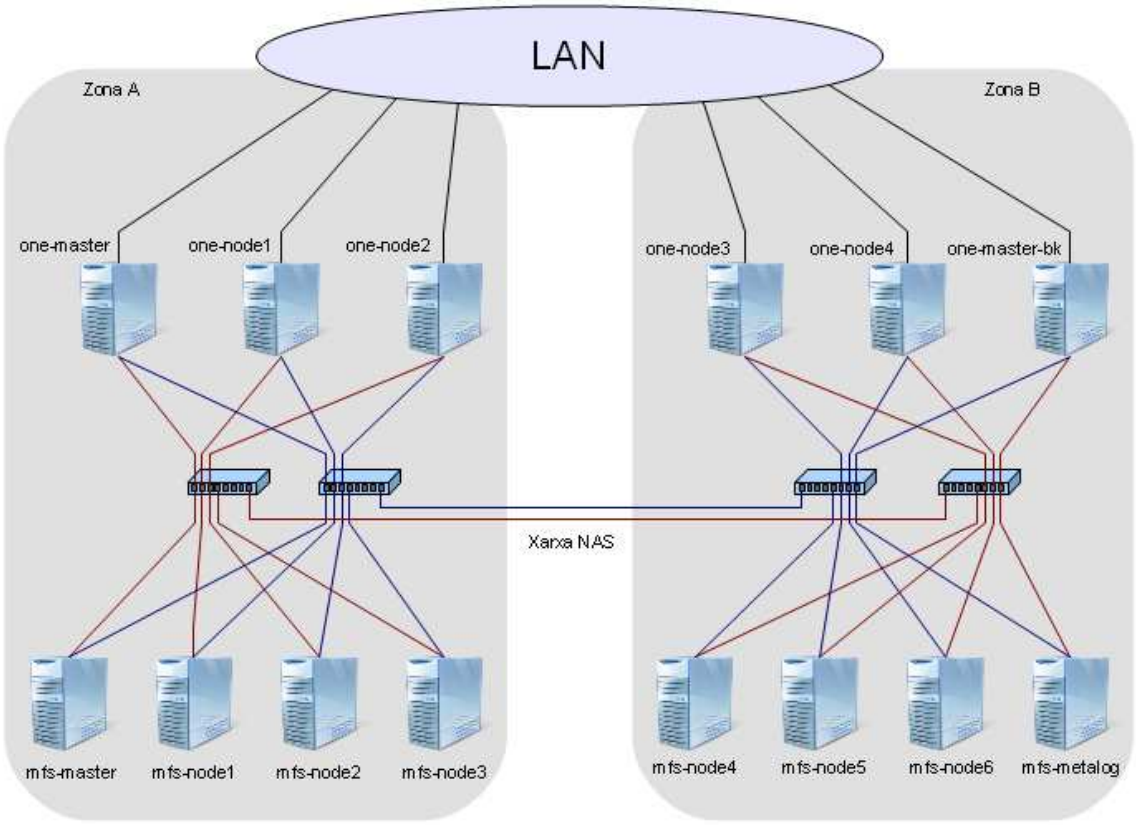
- La Interfície gràfica proporciona més funcionalitats per la gestió de la infraestructura.

- KVM és un hipervisor de codi obert que ve integrat en el nucli dels sistemes operatius Linux. Proporciona les avantatges de la virtualització completa, així permet virtualitzar la gran majoria de sistemes operatius de l'arquitectura x86.

- El sistema de fitxers MooseFS proporciona un rendiment superior a altres que fan ús de *proxy*, ja que els clients aprofiten l'amplada de banda de tots els seus nodes. Així, a mesura que s'afegeixen nodes, també augmenta l'amplada de banda i en conseqüència el rendiment del sistema.

7.2 Disseny de la infraestructura

El disseny de la infraestructura, tot i ser per un entorn de proves, persegueix garantir l'alta disponibilitat i l'escalabilitat en els recursos de computació i emmagatzemament. Per tal d'aconseguir-ho cal disposar de maquinari adequat, distribuir-lo i interconnectar-lo tal com es representa en la il·lustració 24.



Il·lustració 24: Disseny de la Infraestructura

D'aquesta manera, la infraestructura es distribueix en dues zones, la Z*ona A* i la Z*ona B,* que s'ubicaran de la manera més aïllada possible. Cada zona disposa dels seus propis controladors i nodes, de computació i d'emmagatzemament respectivament. Així, si una de les zones queda fora de servei, l'altre garantirà -com a mínim- la continuïtat de negoci dels serveis més crítics.

Cada zona implementa tres nodes de computació (identificats amb *one-*) que proporcionen serveis de virtualització mitjançant l'hipervisor KVM. A més, el servidor *one-master*, és el controlador de computació d'ambdues zones i el servidor *one-master-bk* és el backup del controlador de computació; aquest agafarà el control en el cas que falli el principal.

Pel que fa al sistema d'emmagatzemament, es disposen quatre nodes per zona. En la zona *A* hi ha el servidor principal *mfs-master* que és el controlador del sistema de d'arxius d'ambdues zones i els servidors de dades (*mfs-node1*, *mfs-node2* i *mfs-node3).* De la mateixa manera, en la zona *B* hi ha els servidors de dades (*mfs-node4*, *mfs-node5* i *mfs-node6)* i *mfs-metalogger* que és el servidor de backup del controlador d'emmagatzemament.

La interconnexió entre el sistema de computació i el sistema d'emmagatzemament es fa mitjançant una xarxa NAS aïllada i interconnectada entre les dues zones. Aquesta per tal

de ser redundant disposa de dos *switchs* per zona. Cada un d'ells connecta amb tots els nodes de la seva zona i amb un switch de l'altra zona. Així, en el cas que un deixi de funcionar, l'altre seguirà interconnectant tots els nodes.

D'aquesta manera, cada una de les computadores ha de disposar de dues interfícies de xarxa a 1Gb dedicades a NAS. Aquestes es configuren amb la tècnica de *bonding* (permet que varies interfícies actuïn com una sola), concretament amb el mode 0 o *round-robin,* que a més de permetre tolerància a fallades, proporciona balanceig de carrega i pràcticament s'aconsegueix doblar l'amplada de banda.

Per tal de que no es formin *colls de botella* entre les dues zones, el *switchs* permeten la interconnexió a 10 Gb entre ells. Aquesta amplada de banda és més adequada, ja que els nodes de computació utilitzen indistintament l'emmagatzemament de les dues zones. A més, la replicació de dades del sistema *MosseFS* també es fa entre tots els nodes d'emmagatzemament.

La instal·lació dels components d'OpenNebula juntament amb el sistema d'arxius MosseFS es descriuen de forma detallada en l'annex “*Instal·lació de la plataforma”*.

Resum

Les tecnologies de la virtualització inclouen grans avantatges en les infraestructures TI. Aquestes poden optimitzar els recursos computacionals que, gràcies als hipervisors, permeten la l'abstracció del maquinari i la disposició de varies màquines virtuals en un mateix amfitrió.

Hi ha molts hipervisors. Aquests poden utilitzar diferents tècniques de virtualització, cada tècnica té els seus avantatges i inconvenients. La virtualització completa és la més popular, ja que permet -per definició- virtualitzar tots els sistemes operatius compatibles de l'arquitectura subjacent i proporcionar un rendiment proper al natiu.

L'ús massiu de màquines virtuals en una infraestructura que no té les eines adequades acaba sent una tasca complicada. Els programaris que proporcionen computació en núvol eleven el nivell d'abstracció a una capa superior i virtualitzen tot el conjunt de maquinari d'una infraestructura TI.

Les programaris OpenStack, Eucalyptus i OpenNebula proporcionen unes plataformes de computació en núvol molt dinàmiques en quant a configuració, altament escalables i amb tolerància a fallades. OpenNebula resulta ser la que millor s'adapta a un entorn privat.

La plataforma implementada aprofita la funcionalitat d'OpenNebula per la gestió de la infraestructura virtual. Aquest programari permet orquestrar hipervisors KVM i VMware Server de manera conjunta, i aportar emmagatzemament mitjançant el sistema d'arxius MosseFS. Tot plegat, correctament distribuït i configurat, proporciona una infraestructura virtual altament escalable i fiable.

Conclusions

En principi, es pretenia implementar el projecte amb la plataforma OpenStack (molt ben acceptada per fabricants de maquinari), però un cop analitzada, no s'adaptava completament als requeriments inicials. Això ha provocat l'estudi d'altres programaris similars com Eucalyptus i OpenNebula amb la conseqüent re-planificació del pla de treball inicial.

OpenStack i Eucalyptus són plataformes que estan més orientades a núvols de computació públics. Aquestes, per aconseguir la persistència i l'alta disponibilitat de les instancies han d'utilitzar components de tercers com fàbriques SAN i sistemes d'arxius compartits.

OpenNebula, que al principi semblava la menys completa, ha resultat ser la que millor s'adapta a un entorn de computació en núvol privat, principalment per el subsistema d'emmagatzemament *TM MosseFS* i la funcionalitat de la interfície gràfica. L'ecosistema d'OpenNebula amb sistema d'arxius MosseFS proporciona persistència i altra disponibilitat de les instàncies.

L'estudi de les plataformes que proporcionen Infraestructura com a servei ha provocat el coneixement de molts hipervisors. KVM ha semblat una bona opció per la implementació de la plataforma, ja que proporciona virtualització completa i suporta la majoria de sistemes operatius x86. A més, ve integrat en el nucli dels sistemes Linux.

La instal·lació del programari no ha estat una tasca senzilla. Tant d'OpenNebula com de MosseFS, proporcionen la informació necessària per la instal·lació i configuració del programari en la seva pàgina Web oficial. La implementació de MosseFS s'ha fet sense complicacions, però en el cas OpenNebula s'han de repetir passos i cal modificar un dels *scripts*.

Annex. Instal·lació de la plataforma

En aquest capítol es descriu el procés d'instal·lació, configuració i posada en marxa de tots els components de la Zona A (il·lustració 24). Els components d'aquesta zona són suficients per posar de manifest quasi tots els aspectes relatius a les capacitats de computació i emmagatzemament.

Per començar, tots els servidors porten el sistema operatiu Ubuntu 10.04_adm64 Server i addicionalment el servidor SSH.

1. Instal·lació del sistema d'arxius MosseFS

En primer lloc s'instal·la el sistema de fitxers MooseFS, ja que aquest serà necessari per la correcta configuració de la plataforma de computació.

Per la instal·lació del sistema d'arxius tots els membres hi dediquen dues interfícies de xarxa configurades amb *bonding* mode 0 (*round-robin*). A la següent taula es mostren els participants del sistema d'arxius i la configuració de xarxa.

Tipus	Nom	IP (xarxa NAS)	Màscara
Àlias	mfsmaster	192.168.201.1	255.255.252.0
Servidor	mfs-master	192.168.201.1	255.255.252.0
Servidor	mfs-node1	192.168.201.10	255.255.252.0
Servidor	mfs-node2	192.168.201.11	255.255.252.0
Servidor	mfs-node3	192.168.201.12	255.255.252.0
Client	one-master	192.168.202.1	255.255.252.0
Client	one-node1	192.168.202.2	255.255.252.0
Client	one-node2	192.168.202.3	255.255.252.0

Taula 1. Participants i configuració de xarxa del membres del sistema d'arxius.

1.1 Controlador del sistema d'arxius

Caldrà realitzar les següents passes per la correcta instal·lació i configuració del servidor principal del sistema d'arxius (que hem anomenat *mfs-master*).

En primer lloc s'instal·len les dependències necessàries

```
# sudo apt-get install gcc pkg-config
```

A continuació es configuren els components i s'instal·len

```
# sudo groupadd mfs
# sudo useradd -g mfs mfs
# cd /usr/src
```

```
# sudo wget
http://pro.hit.gemius.pl/hitredir/id=0sWa0S8ft4sTAHF1bGAAEZPcP3ziyq7f9SdhoQf7oeT.c7/url=moosefs.org/tl_files/mfscode/mfs-1.6.20-2.tar.gz
# sudo tar -zxvf mfs-1.6.20.tar.gz
# cd mfs-1.6.20
# ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var/lib --with-default-user=mfs --with-default-group=mfs --disable-mfschunkserver --disable-mfsmount
# sudo make
# sudo make install
```

Els fitxers de configuració d'exemple s'han creat dins "/etc" amb l'extensió *.dist* , així es poden utilitzar aquest per deixar la configuració per defecte

```
# cd /etc
# sudo cp mfsmaster.cfg.dist mfsmaster.cfg
# sudo cp mfsmetalogger.cfg.dist mfsmetalogger.cfg
# sudo cp mfsexports.cfg.dist mfsexports.cfg
```

Per tal de restringir l'accés a altres possibles equips de la NAS, en el fitxer "mfsexports.cfg" es modifica l'asterisc (*) de la primera línia no comentada per el rang d'IPs que corresponen als clients del sistema d'arxius. La línia queda de la següent manera.

```
192.168.202.0/24 / rw,alldirs,maproot=0
```

S'inicialitza el fitxer de metadades a partir de la plantilla generada durant la instal·lació

```
# cd /var/lib/mfs
# sudo cp metadata.mfs.empty metadata.mfs
```

S'edita el fitxer '/etc/hosts' i es posa la IP de l'alies *mfsmaster*.

```
192.168.201.1 mfsmaster
```

S'engega el servei del controlador del sistema d'arxius i a continuació el monitor CGI

```
# sudo /usr/sbin/mfsmaster start
# sudo /usr/sbin/mfscgiserv
```

Per accedir al monitor CGI: http://192.168.201.1:9425

1.2 Nodes d'emmagatzemament

Caldrà realitzar les següents passes per la correcta instal·lació i configuració de cada un dels nodes d'emmagatzematge (*chunks servers*). Aquest són els identificats com *mfs-nodeN* -N és el número de node- en la taula 1.

En primer lloc s'instal·len les dependències necessàries

```
# sudo apt-get install gcc pkg-config
```

A continuació es configuren els components i s'instal·len

```
# sudo groupadd mfs
# sudo useradd -g mfs mfs
# cd /usr/src
# sudo wget
http://pro.hit.gemius.pl/hitredir/id=0sWa0S8ft4sTAHF1bGAAEZPcP3ziyq7f9SdhoQf7oeT.c7/url=
moosefs.org/tl_files/mfscode/mfs-1.6.20-2.tar.gz
# sudo tar -zxvf mfs-1.6.20-2.tar.gz
# cd mfs-1.6.20-2
# sudo ./configure --prefix=/usr --sysconfdir=/etc
--localstatedir=/var/lib --with-default-user=mfs
--with-default-group=mfs --disable-mfsmaster
# sudo make
# sudo make install
```

Es preparen els fitxers de configuració dels serveis per el node, així és copien els fitxers que venen per defecte

```
# cd /etc/
# sudo cp mfschunkserver.cfg.dist mfschunkserver.cfg
# sudo cp mfshdd.cfg.dist mfshdd.cfg
```

En el fitxer "mfshdd.cfg" es configuren les rutes dels discs o particions muntades i que volem cedir al sistema de fitxers. Així, en cada node s'ha muntat una partició en "/mnt/mfschunks1" i en el fitxer "mfshdd.cfg" s'ha afegit: /mnt/mfschunks1

Cal donar permisos a l'usuari *mfs* per que controli els punts de muntatge.

```
# sudo chown -R mfs:mfs /mnt/mfschunks1
```

Afegim la següent línia al fitxer "/etc/hosts"

```
192.168.201.1 mfsmaster
```

Finalment es pot arrancar el servei:

```
# sudo /usr/sbin/mfschunkserver start
```

1.3 Client del sistema d'arxius

Per tal d'instal·lar el client de MooseFS (mfsmount) és necessària la instal·lació del paquet *FUSE* (versió mínima 2.6, recomanada >= 2.7.2) a cada un dels clients del sistema d'arxius (veure la taula 1). A continuació és descriuen els passos realitzats.

```
# sudo apt-get install gcc
# cd /usr/src
# sudo wget http://downloads.sourceforge.net/project/fuse/fuse-2.X/2.8.6/fuse-2.8.6.tar.gz
# sudo tar –zxvf fuse-2.8.6.tar.gz
# cd fuse-2.8.6.tar.gz
# sudo ./configure
# sudo make
# sudo make install
```

S'instal·len les dependències necessàries per *mfsmount*

```
# sudo apt-get install zlib1g-dev pkg-config
```

A continuació s'instal·la *mfsmount* amb les següents instruccions

```
# sudo wget
http://pro.hit.gemius.pl/hitredir/id=0sWa0S8ft4sTAHF1bGAAEZPcP3ziyq7f9SdhoQf7oeT.c7/url=
moosefs.org/tl_files/mfscode/mfs-1.6.20-2.tar.gz
# sudo tar –zxvf mfs-1.6.20-2.tar.gz
# cd mfs-1.6.20-2
# sudo ./configure --prefix=/usr --sysconfdir=/etc --localstatedir=/var/lib --with-default-user=mfs --
with-default-group=mfs --disable-mfsmaster --disable-mfschunkserver
# sudo make
# sudo make install
```

Cal afegir al fitxer "/etc/hosts" la línia següent:

```
192.168.201.1   mfsmaster
```

A continuació es crea el directori en el qual muntarem el sistema de fitxers

```
# sudo mkdir /srv/vms
```

I finalment, es munta el sistema de fitxers

```
# sudo /usr/bin/mfsmount /srv/vms -H mfsmaster
```

2. Instal·lació de la plataforma OpenNebula

Per la instal·lació de tots els servidors de la plataforma de computació és disposa d'una interfície de xarxa addicional que es connectarà a la LAN. A la taula 2 es mostra la configuració de cada un dels participants.

Tipus	Nom	IP (xarxa LAN)	Màscara
Servidor	one-master	10.215.145.1	255.255.0.0
Servidor	one-node1	10.215.145.2	255.255.0.0
Servidor	one-node2	10.215.145.3	255.255.0.0

Taula 2. Participants i configuració de xarxa de la plataforma de computació.

2.1 Controlador del núvol

La instal·lació del controlador del núvol (*one-master*) requereix en aquest cas de tres elements: el sistema base, la interfície gràfica i el gestor de transferències MosseFS.

2.1.1 Sistema base

En primer lloc s'instal·la el sistema base, per això, es descarrega el paquet *opennebula_3.0.0-1_amd64.deb* des de la pàgina web oficial d'OpenNebula. Aquest correspon a distribució del sistema operatiu instal·lat.

```
# sudo dpkg -i opennebula_3.0.0-1_amd64.deb
```

Com que manquen dependències caldrà instal·lar-les

```
# sudo apt-get –f install
```

La instal·lació dona un error. Per corregir-lo cal accedir al fitxer "/var/lib/dpkg/info/opennebula.postinst" i a la línia 34 on apareix la variable ONE_HOME s'ha de canviar per ONEHOME.

A continuació executar el següent script

```
# sudo /usr/share/one/install_gems cloud sunstone
```

L'script anterior requereix de dependències, caldrà instal·lar-les totes i tornar a executar la instrucció anterior fins que acabi correctament.

Un cop finalitzada la instal·lació, s'ha cal accedir al sistema amb l'usuari *oneadmin* que s'ha creat durant la instal·lació. Això comporta establir una contrasenya per aquest usuari i afegir-lo al fitxer "/etc/sudoers" per que pugui realitzar tasques administratives.

A continuació s'exporta la variable ONE_AUTH que conte l'usuari i la contrasenya de l'usuari *oneadmin*

```
# export ONE_AUTH=/var/lib/one/auth
```

Tot seguit, es pot comprovar que el sistema base funciona

```
# onevm list
```

Si apareix una llista buida amb el següent encapçalament és que s'ha instal·lat correctament.

```
ID USER     GROUP    NAME        STAT CPU     MEM        HOSTNAME        TIME
```

2.1.2 Interfície gràfica

Amb el procés anterior la instal·lació de la interfície gràfica ha quedat a mitges. Per tal poder acabar caldrà fer les següents passes. Per fer-ho s'inicia la sessió amb l'usuari *oneadmin* i s'executen les següents línies.

```
# sudo gem install json
# sudo gem install rack
# sudo gem install sinatra
# sudo gem install thin
```

A continuació es modifica el fitxer "/etc/one/sunstone-server.conf"

```
:host: 10.215.145.1
```

En aquest punt ja es pot arrancar el servei que proporciona la interfície gràfica.

```
# sunstone-server start
```

Per accedir a la consola web cal fer-ho amb el navegador Mozilla Firefox a http://10.215.145.1:9869 amb l'usuari *oneadmin* i la contrasenya que conte el fitxer "/var/lib/one/auth".

2.1.3 Gestor de transferències TM MooseFS

Aquest component permet utilitzar els sistema d'arxius MosseFS com emmagatzematge del sistema de computació. Per tal d'instal·lar-lo cal seguir les següents passes:

En primer lloc, s'ha de muntar el sistema d'arxius compartit si encara no s'ha fet. Aquest procediment es descriu a l'apartat "Client del sistema d'arxius".

Un cop muntat el sistema d'arxius, cal instal·lar el component *TM MooseFS* que proporciona al controlador de computació la capacitat d'interactuar am el sistema d'arxius. Així, es realitzen les següents tasques:

- Es baixa el paquet des de l'adreça *https://github.com/libersoft/opennebula-tm-moosefs/tarball/master*
- Es descomprimeix la carpeta "tm_moosefs" dins "/etc/one"
- Es descomprmeix la carpeta "tm_commands/moosefs" dins "/usr/lib/one/tm_commands"
- Es modifica el fitxer "oned.conf" amb els següents paràmetres:

```
VM_DIR=/srv/vms/deploy
IMAGE_REPOSITORY_PATH = /srv/vms/images
TM_MAD = [
        name       = "tm_moosefs",
        executable = "one_tm",
        arguments  = "tm_moosefs/tm_moosefs.conf" ]
```

Per tal de configurar que el controlador utilitzi el nou gestor de transferències:

```
# onehost create host01 im_kvm vmm_kvm tm_moosefs
```

A continuació es mou el contingut del directori ‘var/lib/one’ dins el sistema de fitxers MosseFS que esta muntat al directori ‘mnt/vms’. I a continuació es crea un enllaç simbòlic.

```
# mv /var/lib/one /srv/vms/one
# ln -s /srv/vms/one /var/lib/one
```

2.2 Nodes de computació

Per la instal·lació del nodes de computació (tots els definits en la taula 2) els servidors de màquines virtuals han de disposar de processadors amb tecnologia Intel VT o AMD-V. El procediment que es descriu s’ha de realitzar en cada node de computació.

Per tal de proporcionar suport a la virtualització s’han d’habilitar aquestes capacitats des de la BIOS, ja que per defecte estan inhabilitades.

Per les proves del núvol privat els nodes de computació utilitzaran l’hipervisor KVM. De manera, que durant la instal·lació del sistema operatiu es selecciona OpenSSH server i Virtual Machine host. També es pot instal·lar a posterior amb el programa *tasksel*.

A continuació es comprova que s’han configurat correctament els *flags* del processador (‘vmx’ per Intel o ‘svm’ per AMD) que donen suport a la virtualització.

```
# grep vmx /proc/cpuinfo
```

(Correcte, es visualitza 'vmx' entre tots els *flags*)

També es comprova que el nucli ha carregat correctament el mòdul de KVM

```
# lsmod | grep kvm
```

(Correcte, es visualitzen els mòduls 'kvm' i 'kvm_intel')

A continuació, s’ha de muntar el sistema d’arxius compartit si encara no s'ha fet. Aquest procediment es descriu a l’apartat “Clients del sistema d'arxius”.

Caldrà crear l’usuari *oneadmin* amb el mateix *uid i gid* que té en el controlador del núvol. Aquest proces es descriu a continuació:

Per obtenir el *uid* i el *gid*, s’executa el en servidor *one-master*

```
# id oneadmin
```

Aquesta instrucció ens mostra:

```
uid=107(oneadmin) gid=117(oneadmin) groups=117(oneadmin)
```

Així, en el node en qüestió es crea el grup *oneadmin* amb el *gid* específic

```
# sudo groupadd --gid 117 oneadmin
```

Es crea el compte d'usuari amb el *uid* especific i se li assigna '/srv/vms/one' com a directori de treball

```
# sudo useradd --uid 107 -g oneadmin -d /srv/vms/one oneadmin
```

Per permetre que el controlador pugui controlar l'hipervisor, caldrà instal·lar el paquet *Ruby*

```
# sudo apt-get install ruby
```

Bibliografia

Davis, T.; Tarreau, W.; Gavrilov, C; Tindel, Chad N.; Girouard, J.; Vosburgh, J. (2009). Bonding, [en línia] http://www.linuxfoundation.org/collaborate/workgroups/networking/bonding, [data de consulta 28/11/2011].

Borychowski Michał (2010). *Installing MooseFS Step by Step Tutorial*, [en línia], Versió PDF disponible a: http://www.moosefs.org/tl_files/manpageszip/moosefs-step-by-step-tutorial-v.1.1.pdf, [data de consulta 10/11/2011].

Eucalyptus (2011). Introducing Eucalyptus 2.0, [en línia] http://open.eucalyptus.com/book/export/html/4260, [data de consulta 08/11/2011].

Eucalyptus (2011). Eucalyptus Administrator's Guide, (2.0) [en línia] http://open.eucalyptus.com/book/export/html/4263, [data de consulta 08/11/2011].

Jones, M. Tim (2009). La anatomía de un hipervisor Linux, [en línia] http://www.ibm.com/developerworks/ssa/library/l-hypervisor/, [data de consulta 01/12/2011].

MooseFS (2011). About MooseFS, [en línia] http://www.moosefs.org/about-mfs.html, [data de consulta 10/11/2011].

OpenNebula (2011). OpenNebula 3.0 Guides, [en línia] http://www.opennebula.org/documentation:archives:rel3.0, [data de consulta 10/11/2011].

OpenNebula (2011). Installing the Software 3.0, [en línia] http://www.opennebula.org/documentation:rel3.0:ignc, [data de consulta 10/11/2011]

OpenNebula (2011). OpenNebula TM MooseFS, [en línia] http://www.opennebula.org/software:ecosystem:moosefs, [data de consulta 02/11/2011].

OpenNebula (2011). OpenNebula Sunstone: The Cloud Operations Center 3.0, [en línia] http://www.opennebula.org/documentation:rel3.0:sunstone, [data de consulta 05/12/2011]

OpenStack (2011). *OpenStack Compute. Administration manual*, [en línia], Versió PDF disponible a: http://docs.openstack.org/diablo/openstack-compute/admin/os-compute-adminguide-trunk.pdf, [data de consulta 05/10/2011].

OpenStack (2011). *OpenStack Object Storage. Administration Guide*, [en línia], Versió PDF disponible a: http://docs.openstack.org/diablo/openstack-object-storage/admin/os-objectstorage-adminguide-trunk.pdf, [data de consulta 05/10/2011].

OpenStack (2011). *OpenStack Image Service. Administration Guide*, [en línia], Versió PDF disponible a: http://docs.openstack.org/diablo/openstack-image-service/admin/os-image-adminguide-trunk.pdf, [data de consulta 05/10/2011].

Santos Ribeiro, M (2009). Virtualization Basics. [en línia] http://itechthoughts.wordpress.com/tag/paravirtualization/, [data de consulta 02/12/2011]

Talens Sergio (2008). *Herramientas de virtualización libres para sistemas GNU/Linux*, [en línia], Versió PDF disponible a: http://www.iti.es/media/about/docs/tic/15/Articulo_1_15.pdf, [data de consulta 02/11/2011].

http://es.wikipedia.org/wiki/Hipervisor

http://en.wikipedia.org/wiki/Protected_mode#Privilege_levels

http://en.wikipedia.org/wiki/Protected_mode

http://en.wikipedia.org/wiki/Full_virtualization

http://en.wikipedia.org/wiki/Platform_virtualization

http://wiki.openstack.org/OpenStackDashboard#devstack

http://contrib.meharwal.com/home/moosefs

http://fuse.sourceforge.net/

https://help.ubuntu.com/community/KVM/Installation