

# IMPLEMENTACIÓ D'UNA INFRAESTRUCTURA WEB AMB COREOS I DOCKER

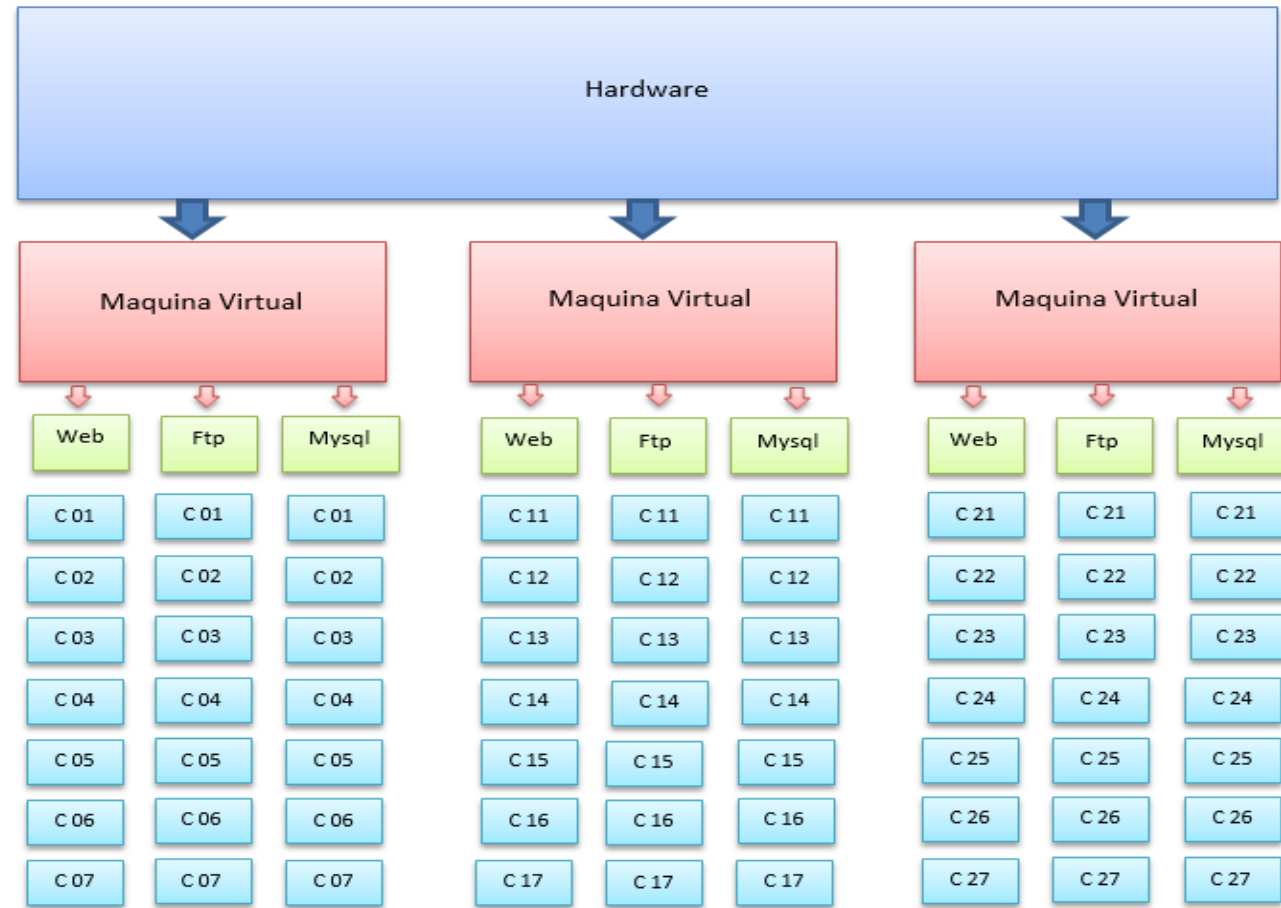
Presentació del Projecte de Final de Grau Enginyeria informàtica  
**Raul Rivero Ramos**



# Introducció

- Una empresa Hosting Web ens demana un projecte.
- Paga indemnitzacions per incompliment SLA amb els clients.
- Tenen una infraestructura basada en màquines virtuals.
- Cada màquina allotja una sèrie de clients prefixats.
- Infraestructura amb problemes de recuperar-se després d'una incidència.
- Un problema a un servidor implica molts clients sense servei.

# Introducció

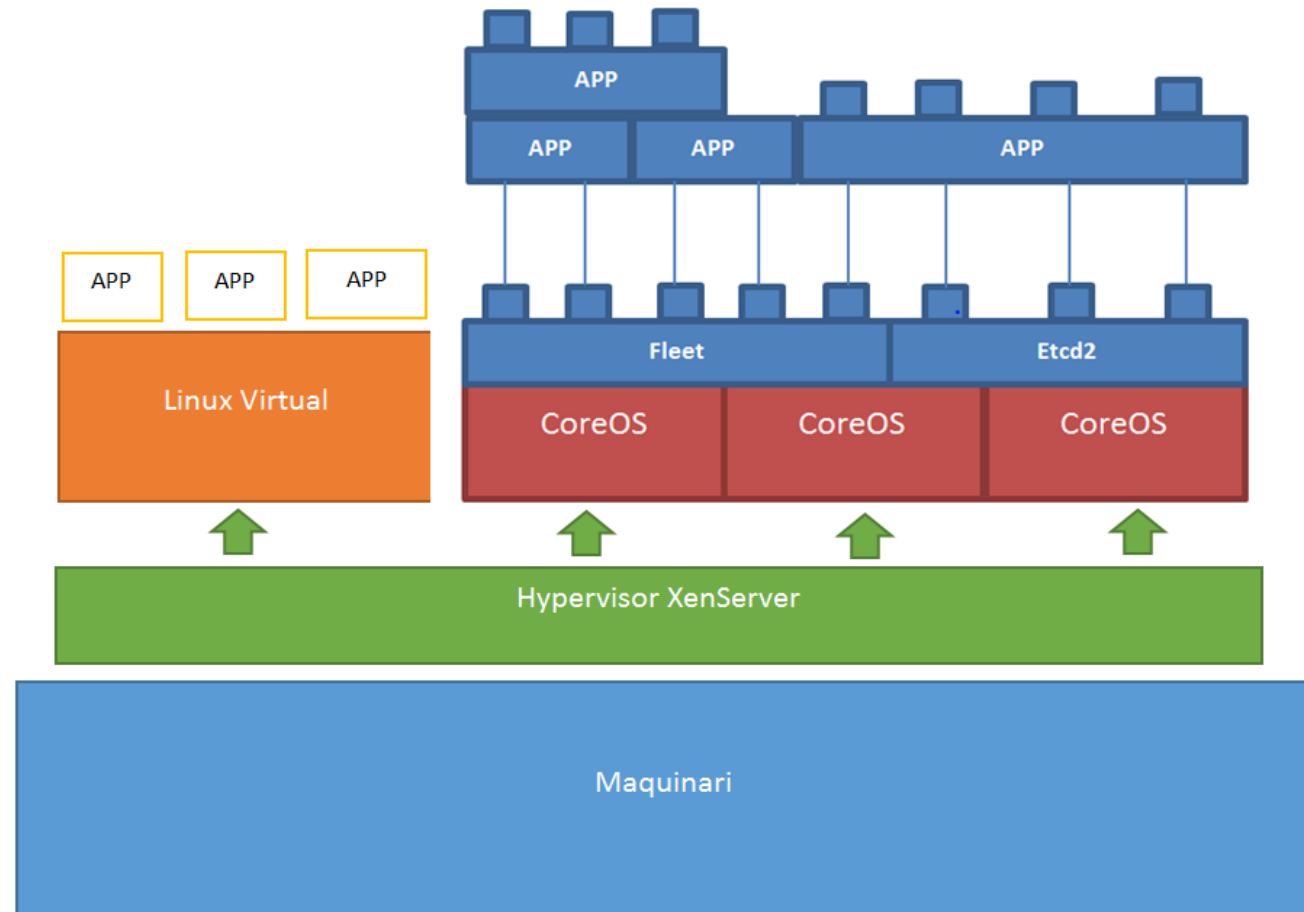


Arquitectura actual empresa Hosting, sobre un maquinari hi ha 3 màquines virtuals i a cadascuna hi ha instal·lades les aplicacions i els usuaris independents.

# Motivació

- Creació d'una infraestructura basada en contenidors Docker amb CoreOS.
- Solucionar la falta de disponibilitat de l'empresa Hosting.
- Comparar rendiments entre infraestructures.
- Demostrar la viabilitat d'aquest tipus d'infraestructures per a les empreses.
- Simular el comportament de l'empresa de hosting Instal·lant serveis web, base de dades i ftp.

# Motivació



Arquitectura proposada d'estudi on hi ha el maquinari l'hipervisor i a sobre d'ell corre l'arquitectura "tradicional" i l'arquitectura CoreOs per a poder comparar-les.

# Objectius

- Explicació teòrica del sistema operatiu CoreOs i els serveis que el complementen.
- Instal·lació d'un prototip d'un clúster de 3 CoreOs sobre un KVM, Xen Server.
- Instal·lació dels serveis Web, Mysql i Ftp sobre el clúster.
- Comparar les dues infraestructures, CoreOS i Maquina Virtual.
- Documentar la instal·lació de la nova infraestructura.
- Documentar les instal·lacions dels serveis.

# Planificació

- El TFG és una assignatura de 12 crèdits.
- La inversió d'hores ha de ser aproximadament de 225 hores.
- S'ha pre-definit treballar 3 hores diàries de dilluns a divendres.
- Els dies laborables han estat 78, que fa un total de 234 hores.
- En realitat s'han invertit més de 300 hores al projecte.

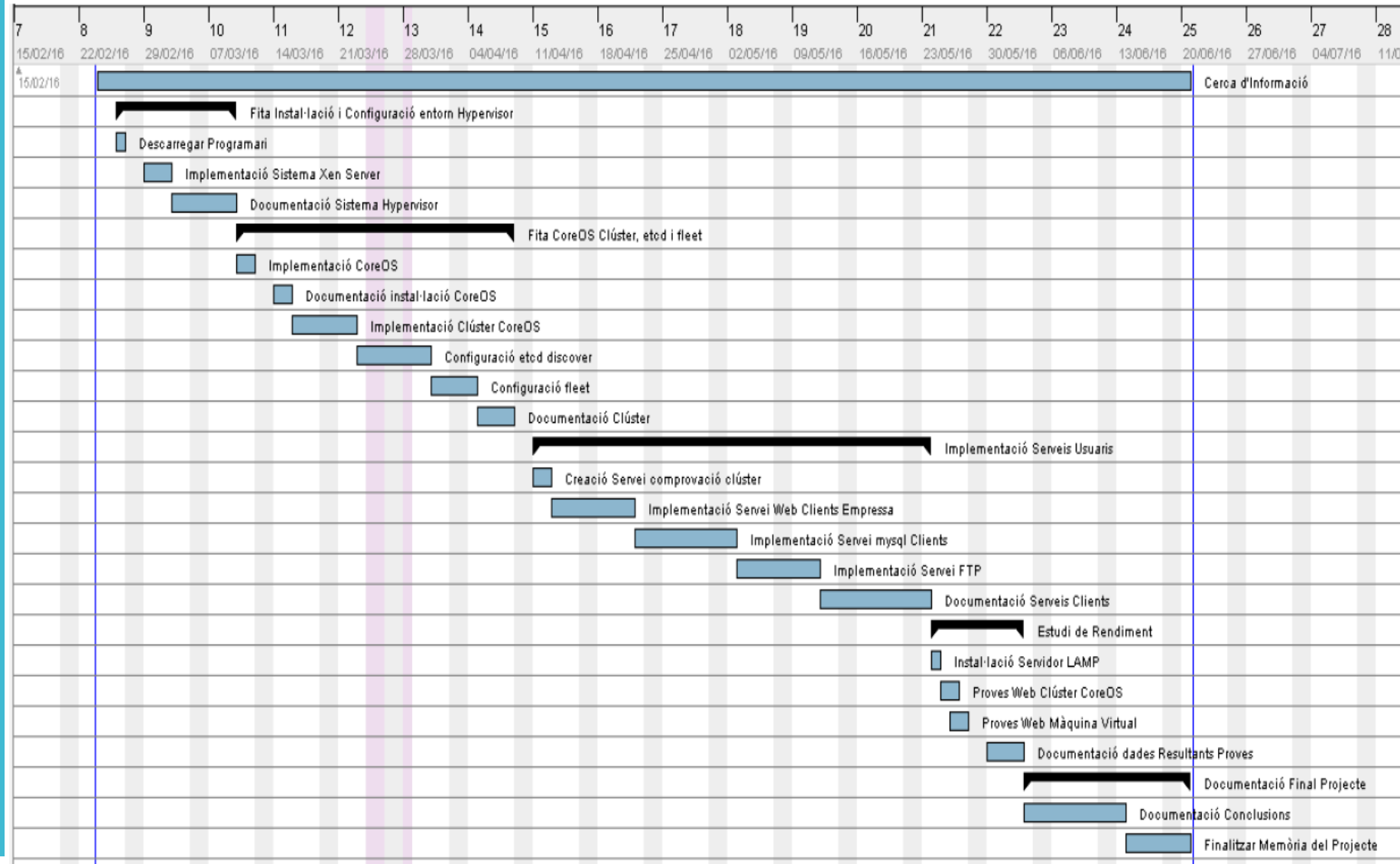
# Planificació

## (Tasques Principals)

- Cerca d'informació.
- Calcular pressupost.
- Definició de les tasques.
- Instal·lació entorn Hipervisor.
- Instal·lació del clúster CoreOS.
- Proves de disponibilitat.
- Proves d'escalabilitat.
- Proves de modularitat.
- Implementació servei Web.
- Implementació servei BBDD.
- Implementació servei FTP.
- Instal·lació servidor virtual.
- Proves rendiment entre servidors CoreOS.
- Proves de rendiment entre CoreOS i Servidor Virtual.
- Estudiar proves de rendiment.



# Planificació (Diagrama de Gant)



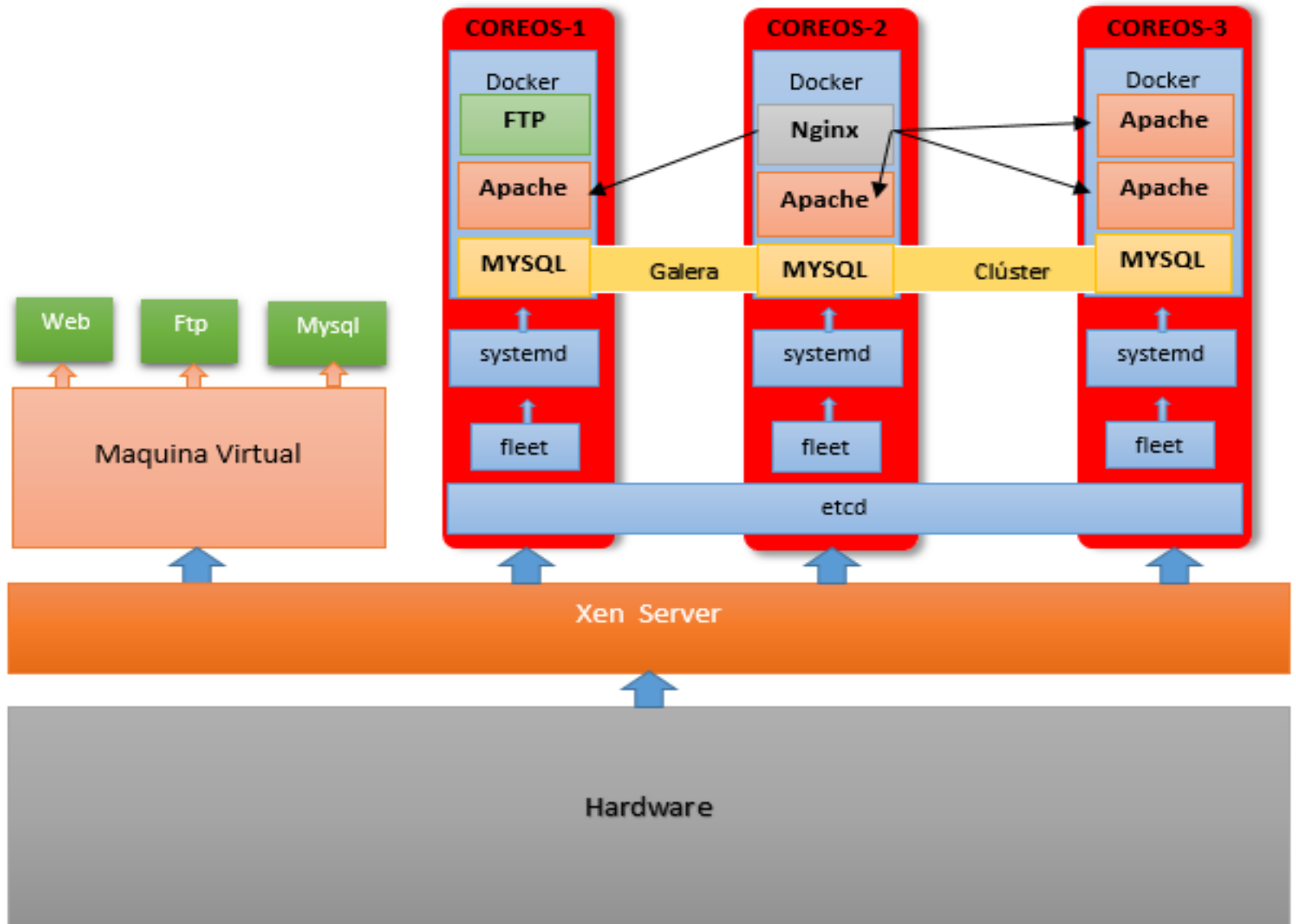
# Projecte

Infraestructura de proves

- L'objectiu és crear una infraestructura de proves amb un Hipervisor.
- S'ha estat escollit XenServer com a Hipervisor.
- Per una banda s'implementa un clúster CoreOS, de 3 nodes, amb els serveis necessaris sobre XenServer.
- Per una altra banda s'implementa un Servidor Virtual Debian amb els mateixos serveis i també sobre XenServer.

# Projecte

Infraestructura de proves



# Projecte

Solució Proposada

- S'ha proposat una solució basada en contenidors amb CoreOS, Docker, etcd i fleet.
- **CoreOS** : sistema operatiu que ocupa molt poc espai i és molt lleuger. Està dissenyat per treballar en clúster. Les aplicacions han de ser executades dins dels contenidors gràcies a Docker.
- **Docker**: permet executar una aplicació fent servir només el necessari per a que funcioni, sense necessitar un sistema operatiu per sota, totes les aplicacions de Docker comparteixen un mateix kernel de Linux.
- **Etcd**: és l'encarregat de replicar i distribuir les dades de configuració, per tots els membres del clúster CoreOS.
- **Fleet**: s'encarrega de gestionar la càrrega de treball del diferents nodes del clúster, és un systemd distribuït.

# Projecte

Solució Proposada

- CoreOS fa servir Docker per donar suport als contenidors, etcd per connectar tots els systemd dels CoreOs que formen el clúster, i per sobre tenim fleet que és l'encarregat de gestionar les unitats (contenidors) dins del clúster balancejant i donant alta disponibilitats a les aplicacions.
- Per configurar un clúster s'ha de generar un identificador únic de clúster, mitjançant un servei web que proporciona etcd, i configurar els diferents Cloud-config de cada node que vulguem afegir al clúster.

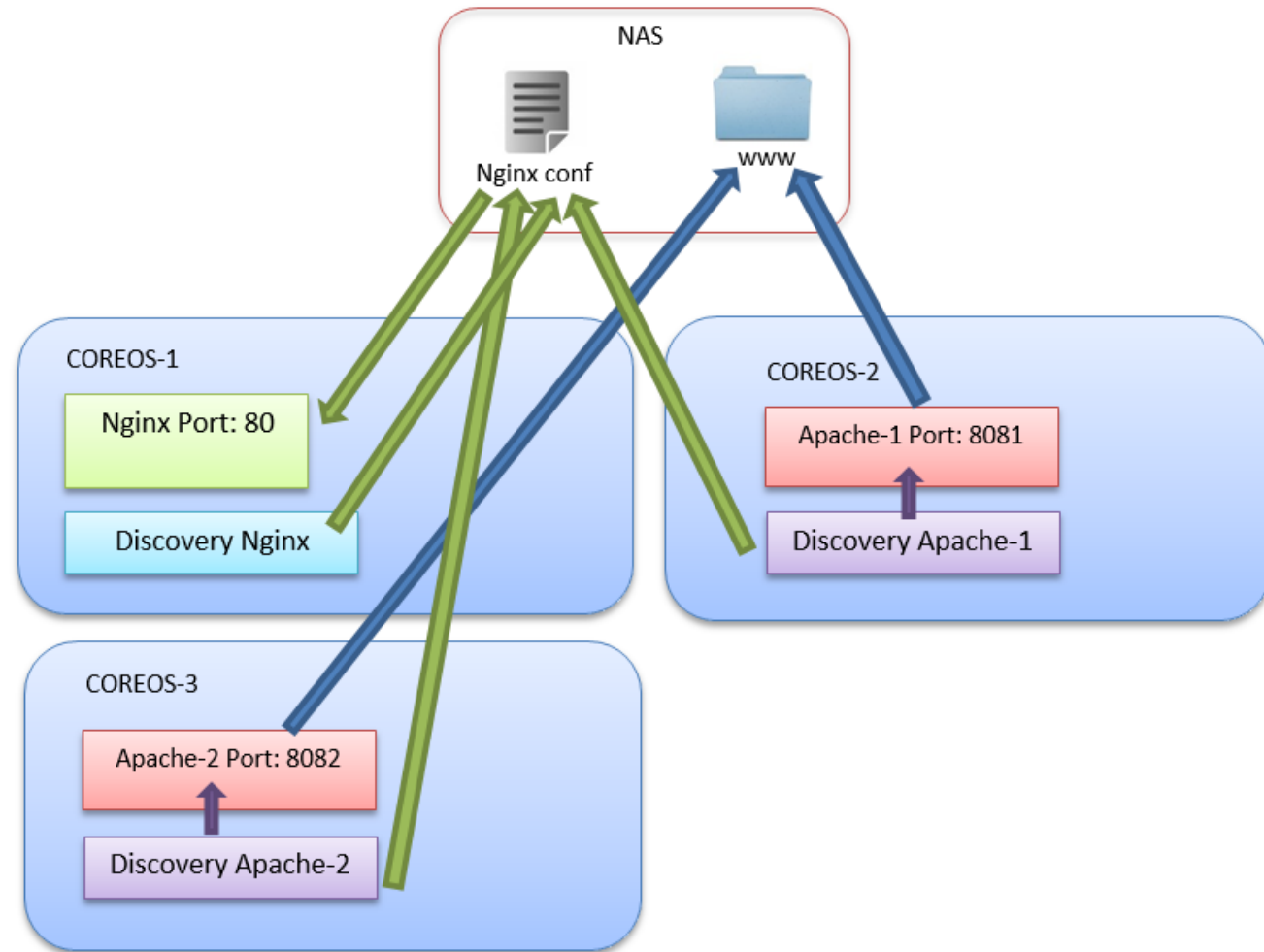
# Projecte

Instal·lació de Serveis  
(Servidor Web)

- Per implementar el servidor web s'ha escollit apache i nginx, aquest últim fa de Proxy de servidors web, els quals tenen els fitxers de configuració i d'usuaris a un directori compartit mitjançant NFS.
- Un servei de descoberta d'apache va actualitzant la llista de servidors disponibles a l'upstream del servidor nginx.
- Un servei de descoberta de nginx substitueix la ip del node on s'està executant el servei a l'script d'actualització de la configuració upstream\_script.

# Projecte

Instal·lació de Serveis  
(Servidor Web)



# Projecte

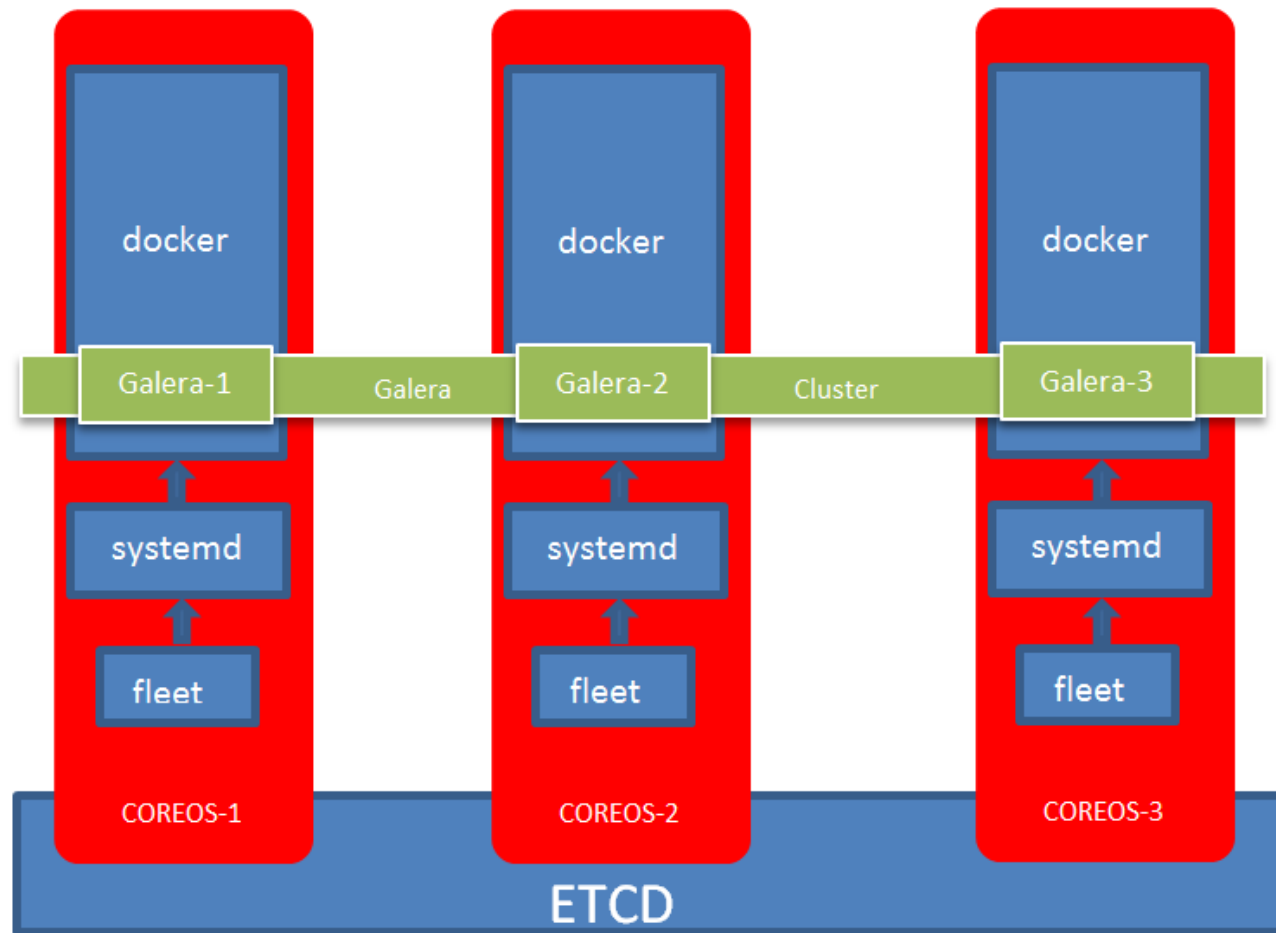
Instal·lació de Serveis  
(Servidor Mysql)

- Per implementar el servidor de BBDD s'ha escollit Mysql.
- La proposta és crear un clúster de serveis Mysql per minimitzar l'impacte de consum de memòria ram d'aquest tipus de programari.
- Al crear un clúster de serveis Mysql també garantim disponibilitat i balanceig de càrrega.
- El servei consta d'una imatge descarregada del repositori Docker i un servei de descoberta que ajuda als serveis a trobar-se i configurar-se com a clúster.



# Projecte

Instal·lació de Serveis  
(Servidor Mysql)



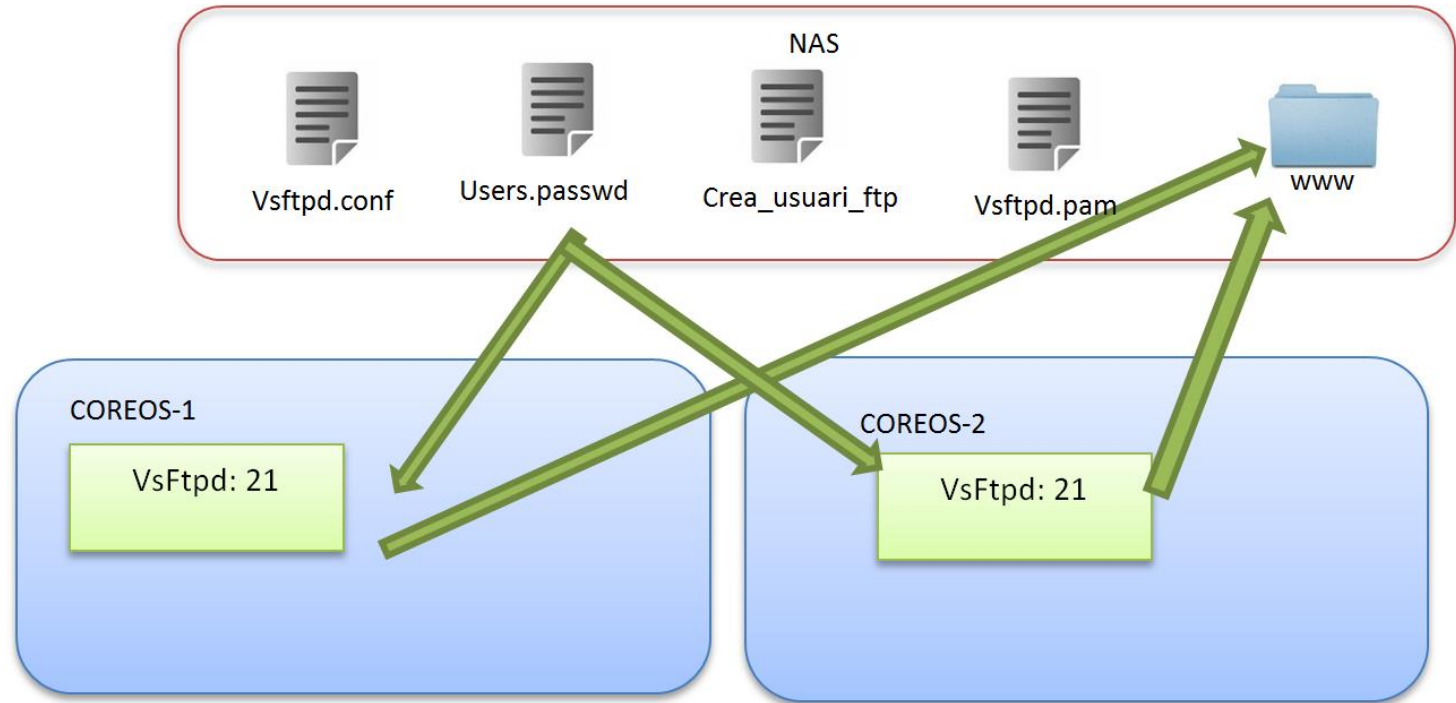
# Projecte

Instal·lació de Serveis  
(Servidor FTP)

- Per implementar el servei FTP s'ha escollit el programari vsftpd.
- Aquest programari permet connexions segures i creació d'usuaris virtuals.
- Aquest servei s'ha creat des d'un document Dockerfile, s'ha creat una imatge a un repositori propi de Docker Hub.
- Mitjançant un script es crea l'usuari nou al fitxer de configuració que tenim al directori compartit NFS.
- La configuració i els fitxers d'usuaris estan emmagatzemats al directori compartit NFS, d'aquesta manera es pot iniciar el servei a qualsevol node del clúster.

# Projecte

Instal·lació de Serveis  
(Servidor FTP)



# Projecte

Disponibilitat dels Serveis

- Els serveis estan dissenyats per tenir una alta disponibilitat.
- Els servidors formen un clúster i mitjançant fleet ens encarreguem de que els serveis allotjats als nodes estiguin balancejats i executant-se.
- Si un servei cau fleet s'encarrega d'aixecar-ho.
- Els serveis tenen els fitxers de configuració a un repositori compartit redundat. Quan engeguem un servei que ha caigut la configuració es carrega automàticament.
- Si un servidor web falla cap client es queda sense servei, nginx s'encarrega de fer servir un altre servidor actiu per mostrar els seus continguts.

# Comparativa

## Objectius

- Els objectius d'aquesta comparativa és demostrar la viabilitat d'implementar aquesta arquitectura tant a les empreses de serveis Hosting com a totes les organitzacions on la disponibilitat, l'escalabilitat, la modularitat i la facilitat de recuperació de la infraestructura en cas de desastre sigui una necessitat.
- Es comparen les arquitectures, els rendiments i l'escalabilitat de les dues arquitectures i es possa en coneixement les conclusions a les quals s'ha arribat en aquests estudis.

# Comparativa

Comparativa d'arquitectures

- Arquitectures Servidors Virtuals
- Són arquitectures que sobre un maquinari executem diferents màquines virtuals.
- Cada servidor virtual està assignat a una sèrie de clients fixes.
- Si ens donem d'alta a la plataforma de l'empresa de hosting ens assignen una màquina virtual.
- Si aquesta màquina falla s'han de reubicar els clients a altres màquines.
- Si un servidor virtual falla, tots els usuaris d'aquell servidor es queden sense servei.

# Comparativa

Comparativa d'arquitectures

- Arquitectura CoreOS
- És una arquitectura pensada per a l'escalabilitat i alta disponibilitat dels serveis.
- Poden tenir moltes instàncies d'un mateix servei corrent sobre el mateix node del clúster.
- Serveis fàcilment actualitzables.
- És una arquitectura més escalable, els nodes del clúster no necessiten un mateix hardware.
- Permet moure els serveis entre nodes o a servidors externs molt fàcilment.

# Comparativa

Comparativa de rendiments

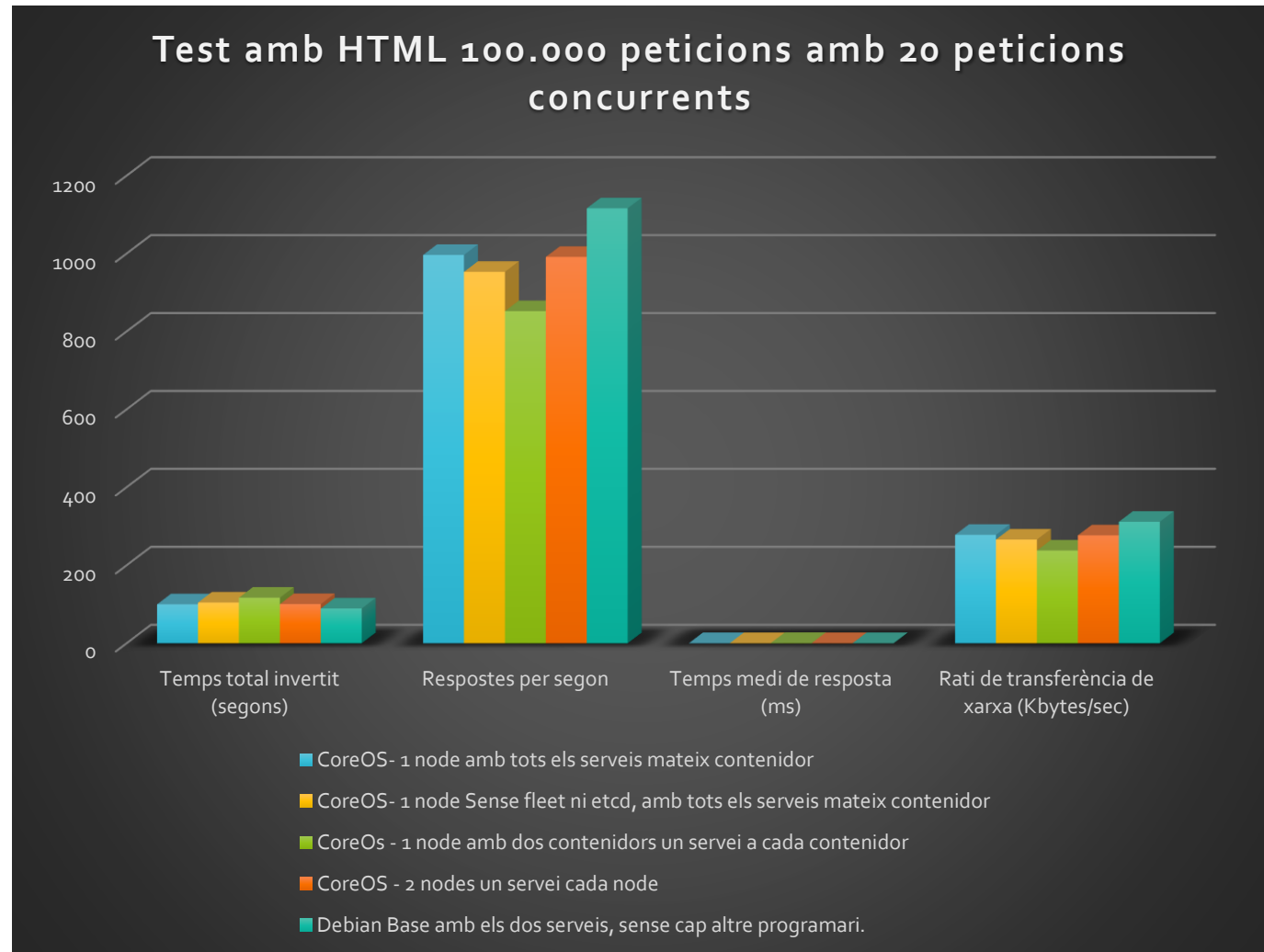
- Els servidors CoreOS fan servir Docker per gestionar els serveis i això incrementa el consum de CPU i RAM envers els servidors virtuals.
- Els servidors CoreOS necessiten un ample de banda de xarxa per comunicar-se entre serveis que els servidors virtuals no necessiten.
- S'han fet diferents estudis de rendiment, tenint en compte les diferents combinacions d'engegada de serveis, quantitat de nodes al clúster, ram i xarxa necessària per les aplicacions i quantitat de peticions concurrents.



# Comparativa

Comparativa de rendiments

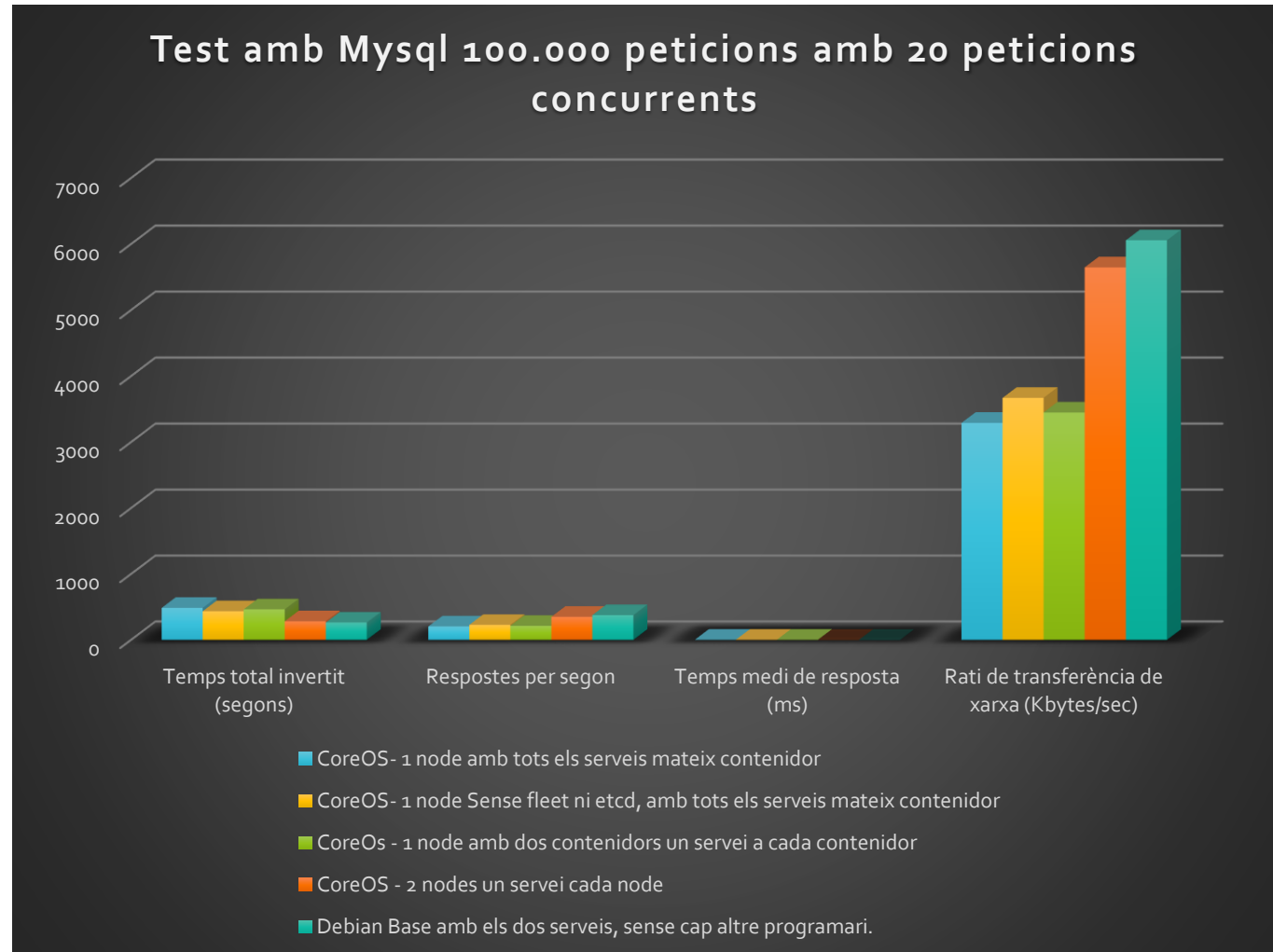
- **Estudi 1.**- Comparativa entre Debian i les diferents modalitats d'implementar els serveis a CoreOS fent 100.000 peticions HTML.



# Comparativa

Comparativa de rendiments

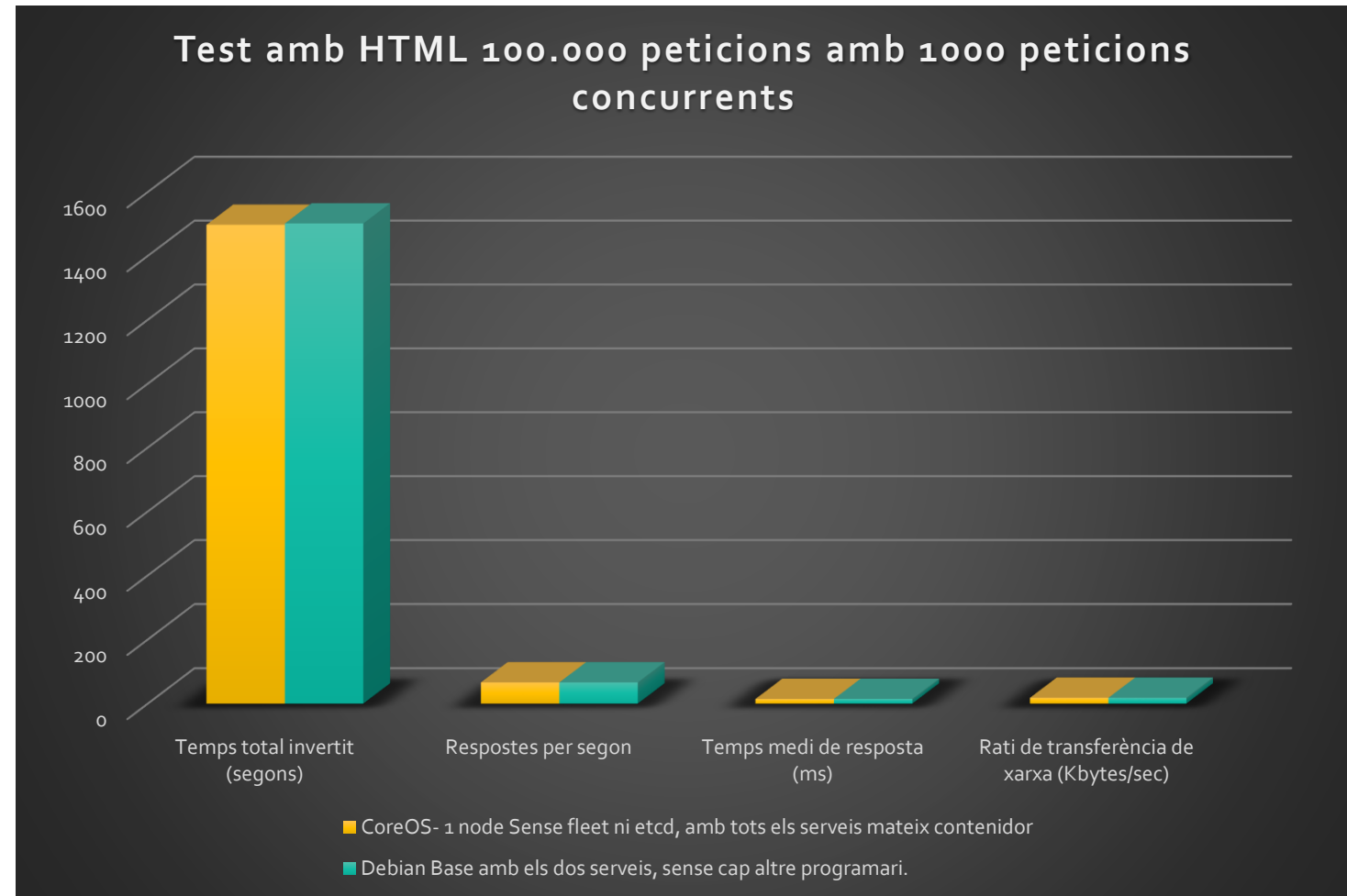
- **Estudi 2.-** Comparativa entre Debian i les diferents modalitats d'implementar els serveis a CoreOS fent 100.000 peticions Mysql.



# Comparativa

Comparativa de rendiments

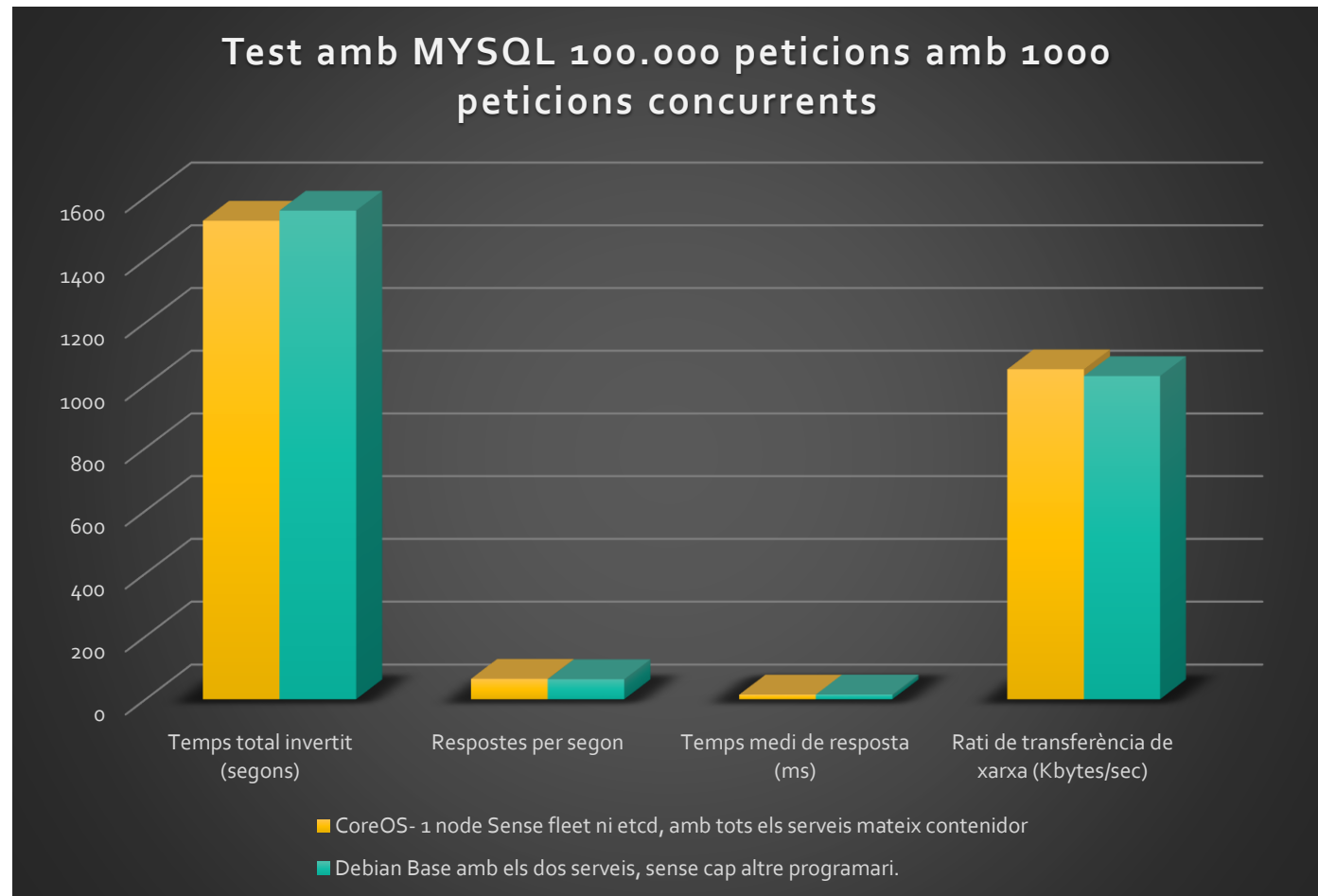
- **Estudi 3.-** Comparativa entre Debian i CoreOS amb un únic contenidor amb php, apache i mysql instal·lat fent 100.000 peticions HTML i amb 1.000 peticions concurrents.



# Comparativa

Comparativa de rendiments

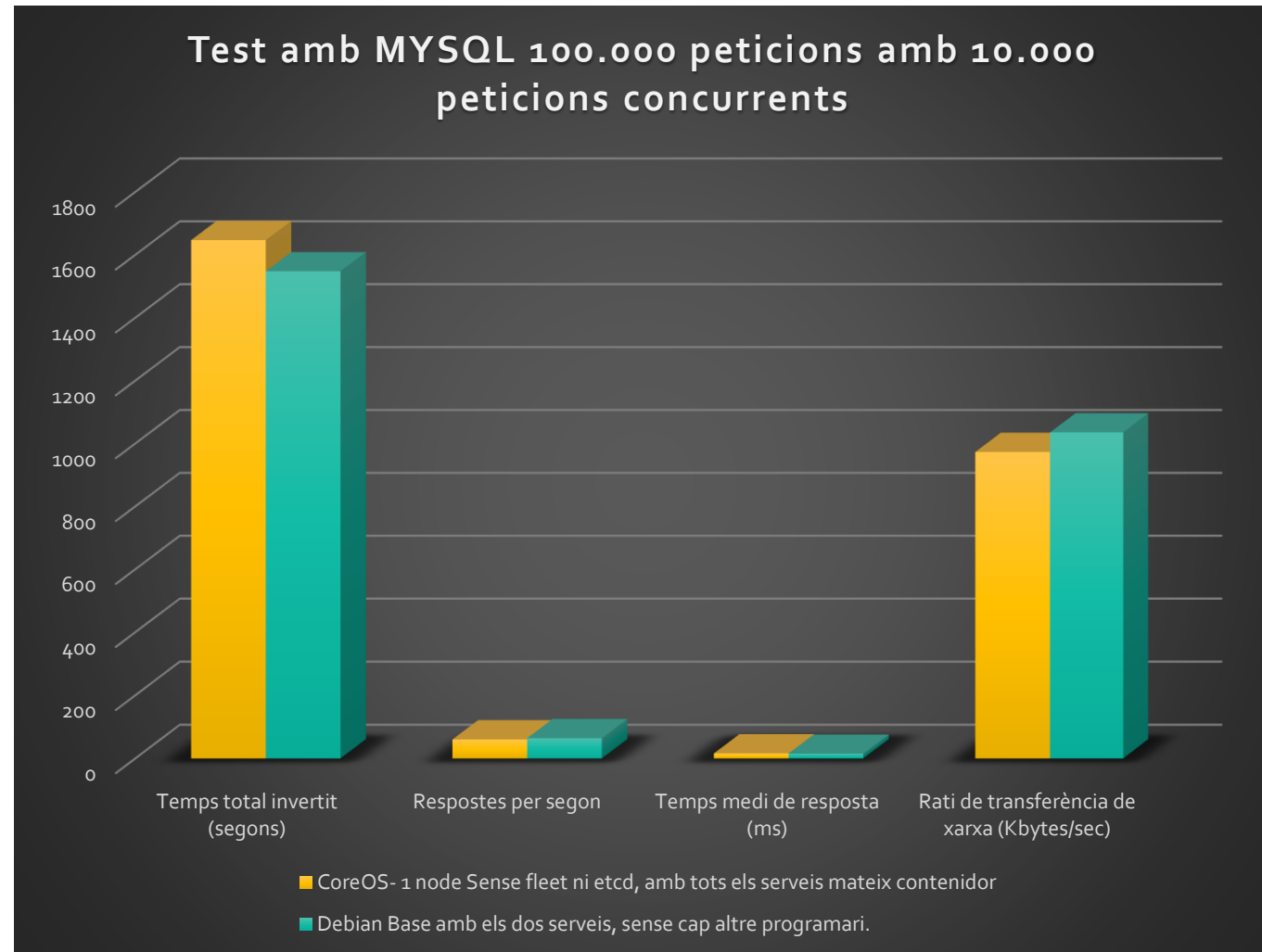
- **Estudi 4.-** Comparativa entre Debian i CoreOS amb un únic contenidor amb php, apache i mysql instal·lat fent 100.000 peticions MYSQL i amb 1.000 peticions concurrents.



# Comparativa

Comparativa de rendiments

- **Estudi 5.-** Comparativa entre Debian i CoreOS amb un únic contenidor amb php, apache i mysql instal·lat fent 100.000 peticions MYSQL i amb 10.000 peticions concurrents.



# Comparativa

Comparativa d'escalabilitat

- Els sistemes basats en contenidors, com ara CoreOS amb Docker són sistemes més escalables que les màquines virtuals.
- Quan es crea una aplicació dins d'un contenidor aquesta aplicació és independent del sistema operatiu en el que l'executem. Podem créixer amb sistemes basats en Docker amb independència de maquinari i sistema operatiu, és més portable.
- En infraestructures de màquines virtuals l'escalabilitat comporta moltes més configuracions i estructuració dels clients per poder balancejar els sistemes.

# Comparativa

Comparativa de disponibilitats

- La disponibilitat de la infraestructura proposada és una disponibilitat del 100%, ja que el clúster té una sèrie de nodes que, mitjançant fleet, es repliquen els serveis que estan executant i si un node cau, fleet redistribueix el servei a un altre node.
- Quan hem d'ampliar el clúster per motius de rendiment, podem afegir nodes sense cap tipus de tall dels serveis.
- Les actualitzacions de la infraestructura CoreOS amb Docker es fan simplement actualitzant la imatge Docker del servei, pujar-la al repositori, i anar engegant contenidors nous amb la imatge nova i aturant els contenidors amb imatges antigues. No cal aturar cap servei ni posar en manteniment cap web del clients.

# Conclusions

- A les proves de rendiment s'ha comprovat que un servidor virtual amb únicament els serveis que s'han dut a estudi té un millor rendiment que el sistema CoreOS ja que aquest últim necessita de més serveis per poder engegar i gestionar els contenidors. Però que aquestes diferències de rendiment no són tant grans com les infinites possibilitats que ens dóna CoreOS amb Docker.
- Com a conclusió personal crec que aquestes infraestructures basades en contenidors aniran implantant-se a les empreses mitjanes i petites, ja sigui amb una infraestructura física o al núvol, amb la modularitat, disponibilitat i avantatges que ens dona envers a les infraestructures "tradicionals" el futur de les empreses serà amb Dockers ha estat un projecte molt enriquidor que m'ha permès aprendre molt sobre un tema del qual no tenia cap experiència.