

# Formación usuarios HPC

Cluster Universidad de Navarra  
Enero 2017

**Jesús Cuenca**  
**HPC Specialist @ SIE**

# Programa

Introducción

Componentes físicos

Redes

Servicios

Acceso al cluster

Gestión de usuarios

Almacenamiento paralelo

# Programa

Almacenamiento paralelo

Ciclo de vida de datos

Slurm

Gestión del software

Fundamentos de seguridad

GPU

Kickstart

Green computing

# Introducción

## qué es un cluster LadonOS

# Introducción SIE

Sistemas Informáticos Europeos lleva trabajando ininterrumpidamente desde 1990 en el sector de la informática. Inicialmente se dedicó a las redes y comunicaciones, en 1999 monta su primer clúster en el CSIC.

A día de hoy, SIE Ladón se ha convertido en una marca de referencia en el mercado de HPC en 3 grandes grupos de soluciones hardware:

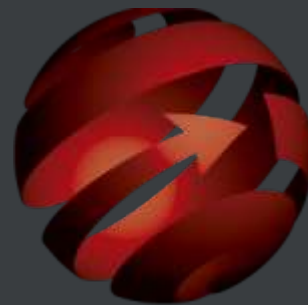
- Workstation silenciosas, ideales para los despachos de investigadores.
- Máquinas de memoria compartida, (Windows / Linux)
- Clústers basados en el sistema Ladon OS.



# Introducción HPC AdminTech

<http://www.hpcadmintech.com/>

Dirigido a administradores de sistemas HPC, investigadores que hacen cálculo científico, desarrolladores de aplicaciones HPC y personal de IT vinculado a la gestión de sistemas de cálculo intensivo.



**¡NO TE PIERDAS LA EDICIÓN DE 2018!**

# Introducción qué es un cluster LadoNOS

## CLUSTER

Ordenadores unidos entre sí (nodos) mediante conexiones de alta velocidad y que se comportan como una sola máquina.

## NODO

Ordenadores, sistemas multiprocesador o estaciones de trabajo. Realizan las operaciones de computación requeridas por los clientes del sistema (se recomienda que posean arquitecturas similares).

## CLIENTES

Realizan las peticiones contra el cluster conectándose a éste de forma remota o directa.



# Introducción qué es un cluster LadoNOS



**ALTO  
RENDIMIENTO**



**ALTA  
DISPONIBILIDAD**



**BALANCEO DE  
CARGA**



**ESCALABILIDAD**



# Introducción qué es un cluster LadonOS

Suite HPC Sistemas Informaticos Europeos  
(SIE)

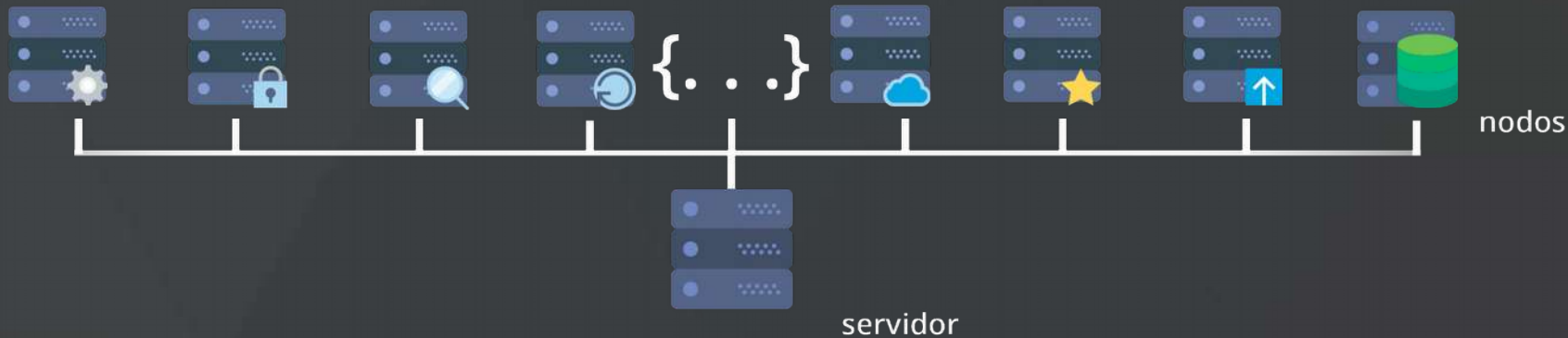
Centos (RedHat Enterprise)

Repositorios oficiales

Totalmente editable y personalizable

Preventa y postventa profesional.

¡¡No hay dos LADONoS iguales!!

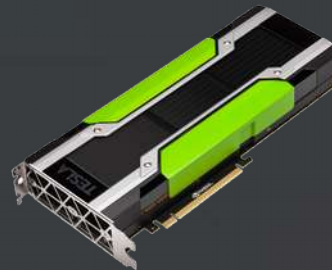


# Introducción qué es un cluster LadonOS

Perfecta armonía entre fiabilidad, seguridad y eficiencia sin dar de lado el soporte oficial.

Aplicaciones de código abierto: el código puede adaptarse a cada necesidad, haciendo que cada instalación de LadonOS sea única.

Cientos de controladores propietarios dotan a LadonOS de compatibilidad con las últimas tecnologías (Infiniband, Intel, PHI, GPUs, librerías y compiladores).



# Introducción qué es un cluster LadonOS



Desde SIE somos plenamente conscientes del coste actual de la electricidad, por este motivo nuestros sistemas HPC disponen de una herramienta de apagado y encendido automático en caso de que un nodo no se esté utilizando.



La seguridad ha sido especialmente cuidada: el servidor hace de pasarela web al resto de nodos para la correcta actualización de parches y seguridad. Dispone de servicios de firewall y entornos de seguridad a la vez que monitorización y seguimiento.



LadonOS dispone de diversos directorios exportados por NFS, dichos directorios son utilizados para la instalación y compilación de programas en el entorno HPC, de este modo el resto de nodos podrá disponer de ellos a la hora de ejecutar programas y cargar librerías. Además cuidamos de los datos más importantes asegurando su integridad.



**BeeGFS**  
Developed by BeeGFS

# Introducción algunos casos de éxito



CiC Cartuja



Universidad Pablo  
Olavide



Universidad  
Girona



Universidad  
Rovira



Universidad  
Auton Barcelona

# Instalación física componentes del cluster

# Instalación física componente del cluster

## RACK

Estructura destinada al alojamiento de todos los componentes hardware que componen el cluster. Ofrecen el soporte, la seguridad y además ayudan a mantener el flujo de refrigeración.



# Instalación física componente del cluster

## NODOS DE CÁLCULO

Ordenadores, sistemas multiprocesador o estaciones de trabajo.

Realizan las operaciones de computación requeridas por los clientes del sistema.

Habitualmente tienen una arquitecturas y configuración similares, pero pueden ser heterogéneos.





# Instalación física componente del cluster

## SWITCHES

Interconectan los nodos para que exista comunicación y se puedan realizar las labores de trabajo destinadas a los mismos.





# Redes

## interconexiones físicas y lógicas

# Redes introducción

El cluster cuenta con sus propias redes internas dedicadas, que interconectan los recursos computacionales.

El nodo principal sirve de pasarela y bastión. Encapsula los detalles del cluster en un punto de acceso, de modo que los usuarios pueden verlo como una entidad única.

# Redes infiniband

Mellanox, Intel

Cobre (hasta 10m.) / Fibra (hasta 10 Km)

Ancho de banda: SDR (2.5 Gb/s) - HDR (50 Gb/s)

Agrupamiento de enlaces:  $4 \times 14 \text{ Gb/s} = 56 \text{ Gb/s}$

Ancho de banda neto (8B/10B) =  $40 \text{ Gb/s}$

# Redes infiniband vs ethernet

| Tecnología       | Velocidad teórica     | Latencia (us) | Consumo (W/h) |
|------------------|-----------------------|---------------|---------------|
| Wifi 802.11      | 0.6 Gb/s (50MB/s)     | 150           |               |
| Gigabit Ethernet | 1 Gb/s (112 MB/s)     | 48            | 1800          |
| 10 Gb Ethernet   | 10 Gb/s (875 MB/s)    | 12            | 900           |
| Infiniband DDR   | 4 x 5 Gb/s (2.5 GB/s) | 2.5           | 600           |
| Infiniband QDR   | 4 x 10 Gb/s (4 GB/s)  | 1.3           | 600           |
| Infiniband FDR   | 4 x 14 Gb/s (7 GB/s)  | 0.7           | 600           |

# Redes ethernet + infiniband

Los nodos del sistema se comunicarán entre ellos mediante dos redes internas.

- Ethernet – Permitirá realizar el SSH a los nodos y lanzar peticiones contra ellos. Las conexiones por IPMI también se realizarán mediante Ethernet.
- Infiniband – Red de alto rendimiento que servirá para la comunicación de los nodos durante la ejecución de trabajos y la compartición de recursos

# Redes asociación de direcciones IP

| Nodo   | IPMI  | PRINCIPAL | INFINIBAN<br>D |
|--------|-------|-----------|----------------|
| nodo00 |       | 2.10      | 5.10           |
| nodo01 | 2.211 | 2.11      | 5.11           |
| nodo02 | 2.212 | 2.12      | 5.12           |
| nodo03 | 2.213 | 2.13      | 5.13           |
| nodo04 | 2.214 | 2.14      | 5.14           |
| nodo05 | 2.215 | 2.15      | 5.15           |
| nodo06 | 2.216 | 2.16      | 5.16           |
| nodo07 | 2.217 | 2.17      | 5.17           |
| ...    | 2.218 | 2.18      | 5.18           |
| nodo12 | ...   | ...       |                |

# Servidor principal estructura lógica

# Servidor principal almacenamiento

| Dispositivo  | RAID | Tamaño         | Sistema ficheros   |
|--------------|------|----------------|--|
| sdd          | 1    | 511 GB         | /<br>/usr/local<br>/opt<br>/share<br>/tmp<br>/var<br>/home |
| sda,sdb, sdc | 6    | <b>300 TiB</b> | <b>/mnt/beegfs</b>   |
| sde          | 1    | 944 GiB        | /data/beegfs/meta  |



# Servidor principal almacenamiento

Los directorios de inicio de los usuarios se encuentran en /home.

Las aplicaciones compartidas por NFS /opt

Los archivos de gestión e instalación de los nodos se guardan en /share

Todas ellas se comparten por NFS con los nodos de cálculo.

Además, se dispone del sistema de ficheros de alto rendimiento **BeeGFS**, montado en /mnt/beegfs. Es en este directorio donde se recomienda que los usuarios almacenen sus datos.



# Servidor principal redes

```
[root@nodo00 ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp5s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether ac:1f:6b:40:37:0e brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.10/24 brd 192.168.2.255 scope global enp5s0f0
        valid_lft forever preferred_lft forever
    inet6 fe80::20e1:f1c2:7350:37fd/64 scope link
        valid_lft forever preferred_lft forever
3: enp5s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether ac:1f:6b:40:37:0f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.204/24 brd 192.168.1.255 scope global enp5s0f1
        valid_lft forever preferred_lft forever
    inet6 fe80::f3a0:c5e9:3683:27a4/64 scope link
        valid_lft forever preferred_lft forever
4: enp130s0f0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000
    link/ether 90:e2:ba:ed:4d:d0 brd ff:ff:ff:ff:ff:ff
5: enp130s0f1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000
    link/ether 90:e2:ba:ed:4d:d1 brd ff:ff:ff:ff:ff:ff
6: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2044 qdisc pfifo fast state UP qlen 256
    link/infiniband 80:00:02:08:fe:80:00:00:00:00:00:00:00:24:8a:07:03:00:5d:3b:41 brd 00:ff:ff:ff:ff:ff:ff
    inet 192.168.5.10/24 brd 192.168.5.255 scope global ib0
        valid_lft forever preferred_lft forever
    inet6 fe80::af3c:522:59d0:6567/64 scope link
        valid_lft forever preferred_lft forever
```

# Servidor principal servicios

**Pasarela + Firewall**

**DHCP**

**NIS**

**NFS**

**NTP**

**Kickstart**

**SSH**

**Apache**

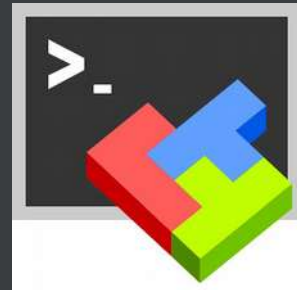
# Conexión con el cluster

## accesos y usuarios

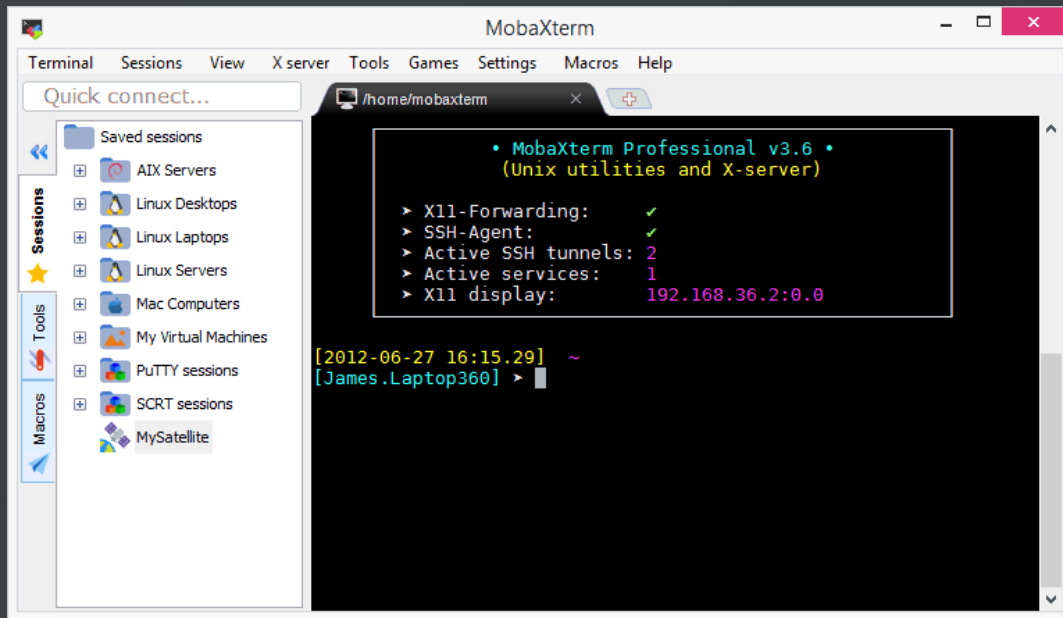
# Conexión con el cluster clientes

Conexión mediante SSH:

- Windows:
  - + Putty
  - + MobaXterm
- Mac & Linux:
  - + Terminal del sistema

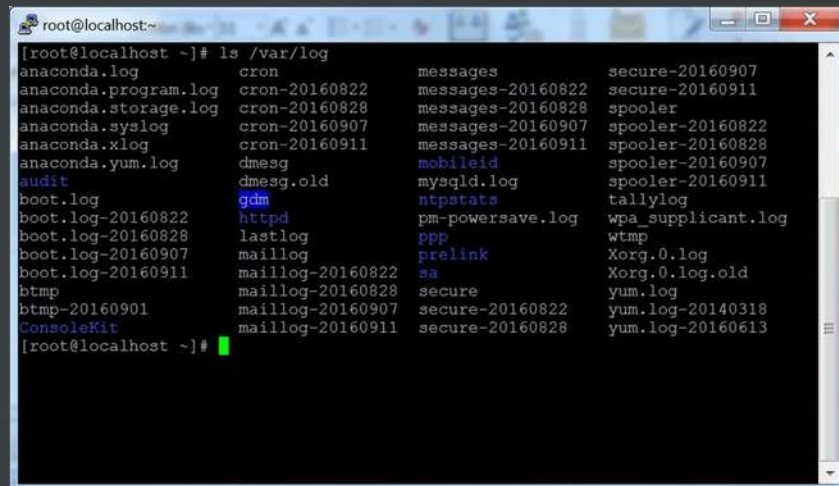
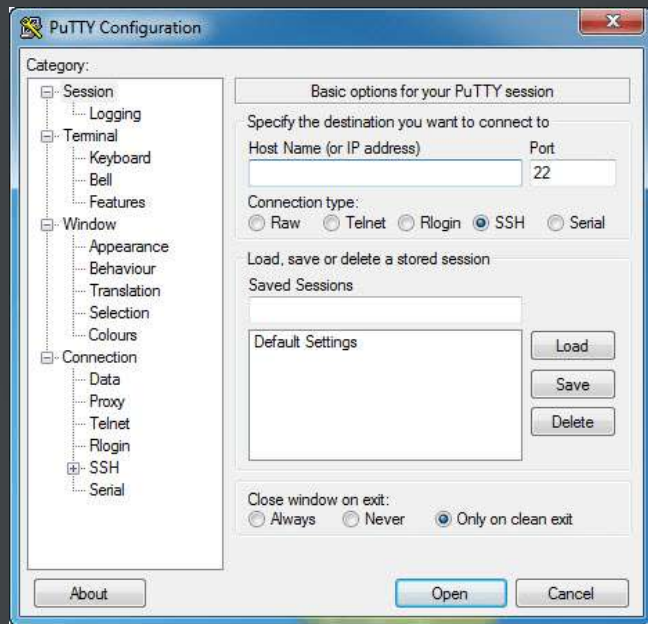


# Conexión con el cluster

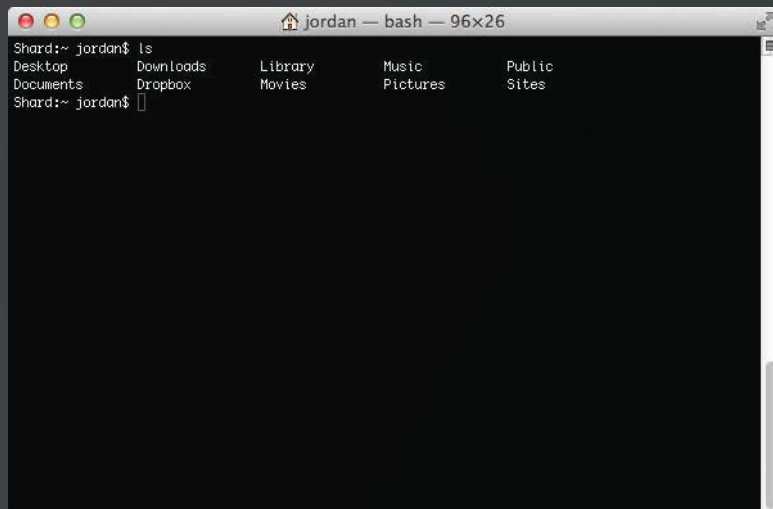


MobaXterm

# Conexión con el cluster



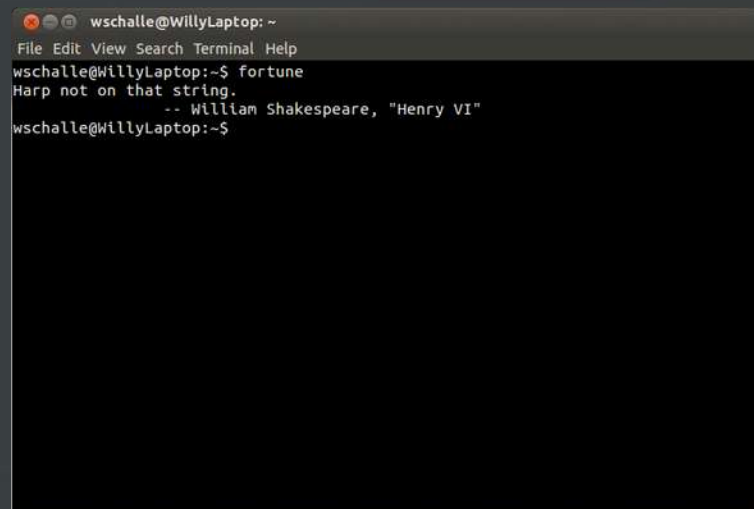
# Conexión con el cluster



A screenshot of a terminal window on a Mac. The title bar reads "jordan — bash — 96x26". The terminal shows the user "Shard" at the prompt "jordan\$". They have entered the command "ls", which displays a list of directories: Desktop, Downloads, Library, Music, and Public in the first row; Documents, Dropbox, Movies, Pictures, and Sites in the second row. The prompt returns to "Shard:~ jordan\$".

```
jordan — bash — 96x26
Shard:~ jordan$ ls
Desktop    Downloads  Library    Music      Public
Documents  Dropbox    Movies     Pictures   Sites
Shard:~ jordan$
```

MAC



A screenshot of a terminal window on a Linux machine. The title bar reads "wschalle@WillyLaptop: ~". The terminal shows the user "wschalle" at the prompt "wschalle@WillyLaptop:~\$". They have entered the command "fortune", which outputs the text "Harp not on that string." followed by a quote from William Shakespeare's "Henry VI". The prompt returns to "wschalle@WillyLaptop:~\$".

```
wschalle@WillyLaptop: ~
File Edit View Search Terminal Help
wschalle@WillyLaptop:~$ fortune
Harp not on that string.
-- William Shakespeare, "Henry VI"
wschalle@WillyLaptop:~$
```

Linux



# Conexión con el cluster ssh

```
ssh [option] <usuario>@<IP/NOMBRE> -P <PORT>  
ssh -Y demo@hpc.unav.es
```

# Conexión con el cluster transferencia datos

## Subida

```
scp datos_a_subir demo@hpc.unav.es:.  
sftp demo@hpc.unav.es  
rsync -av --progress datos_a_subir demo@hpc.unav.es
```

## Bajada

```
scp demo@hpc.unav.es:ruta/datos  
sftp demo@hpc.unav.es  
rsync -av --progress demo@hpc.unav.es:ruta/datos .
```

# NIS

## gestión de usuarios

# NIS gestión de usuarios

Network Information System

Gestión centralizada de credenciales

Base de datos cliente-servidor especializada en información de sistema:  
cuentas de usuario, claves...

El servidor, ypserv, se ejecuta en el nodo principal.

El cliente, ypbind, en todos los nodos de cálculo

## NIS gestión de usuarios

La base de datos tiene un formato propio, pero se sincroniza fácilmente con los ficheros de configuración estándar:

1) gestionar las credenciales en el nodo principal con las herramientas tradicionales: `useradd`, `groupadd`, `passwd`...

2) sincronizar con NIS:

```
make -C /var/yp/
```

# BeeGFS

## almacenamiento paralelo

# BeeGFS introducción

Sistema de ficheros paralelo

Independiente del hardware

Diseñado para entornos donde el rendimiento es crítico

Rendimiento y sencillez

Open Source + soporte comercial



# BeeGFS introducción

Desarrollado en el Fraunhofer Center for HPC

Servicios profesionales de ThinkParQ

2005: Fraunhofer File System



Fraunhofer







UNIVERSITÄT PADERBORN  
Die Universität der Informationsgesellschaft



TEXAS A&M  
UNIVERSITY



DAIMLER

GOETHE  
UNIVERSITÄT  
FRANKFURT AM MAIN



Heidelberg Institute for  
Theoretical Studies



Observatoire  
de Paris



TOYOTA



DNV GL



istituto  
italiano di  
tecnologia



NYU

جامعة نيويورك أبوظبي  
NYU | ABU DHABI



DET NORSKE



NUS  
National University  
of Singapore

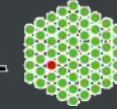


UNIVERSITY OF CAPE TOWN  
YUNIBESITHI YASERAPA - UNIVERSITEIT VAN KAAPSTAD



MAX-PLANCK-GESellschaft

EMBL



Fraunhofer

SeismicCity



Gesellschaft für wissenschaftliche  
Datenverarbeitung mbH Göttingen

Argonne  
NATIONAL LABORATORY



BNP PARIBAS



university of  
 groningen



LADONOS  
HFC ENVIRONMENT



BROAD  
INSTITUTE

Technische Universität  
ILMENAU



Bioinformatics  
Research Center  
BIRC Aarhus



Dept of  
Chemical Engineering

SEES  
LADONOS.ORG

# BeeGFS rendimiento

Optimizado para cargas de rendimiento crítico

Multihilo

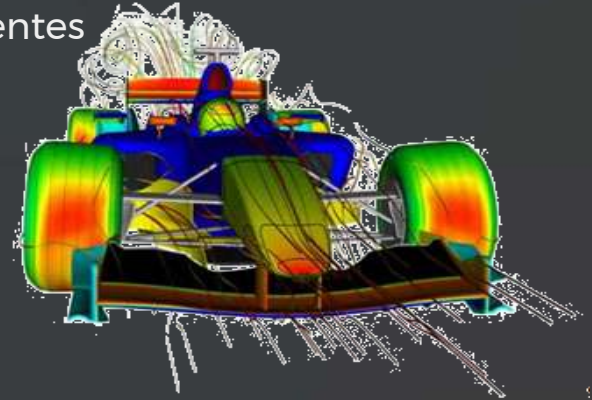
“ BeeGFS at 10GB/s on single node all-flash unit over 100Gbit network”

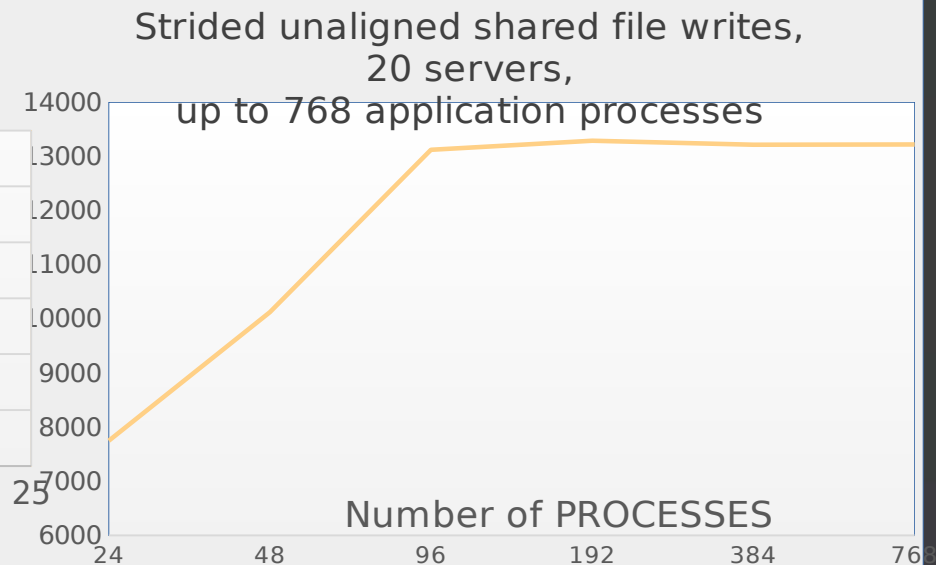
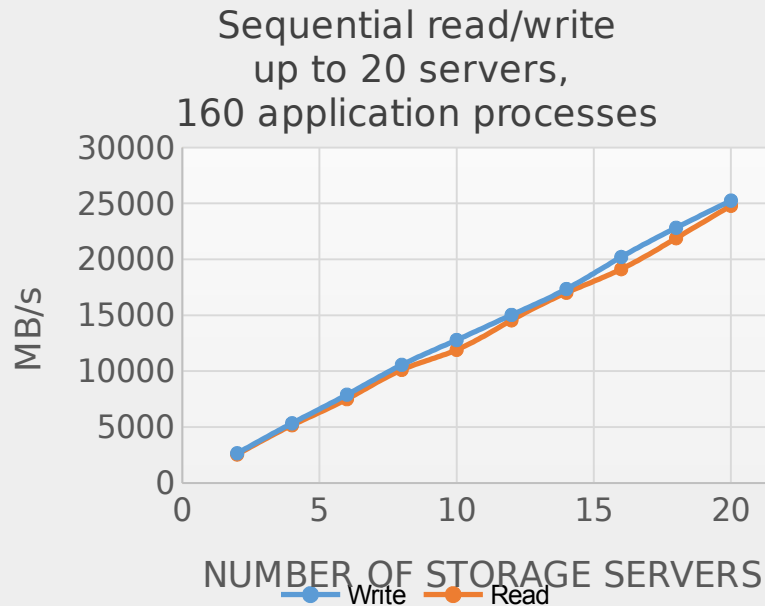
Infiniband / Ethernet

Distribuido: contenido ficheros y metadatos

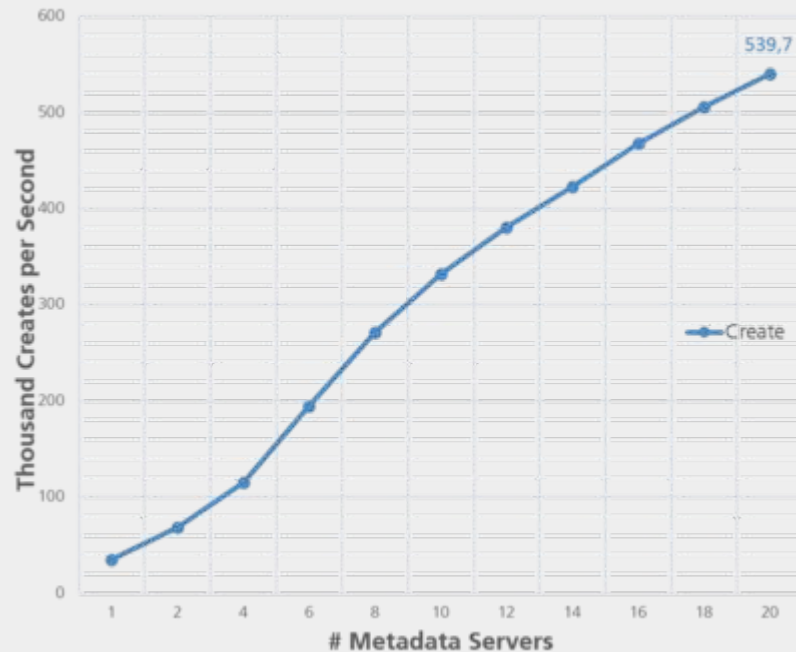
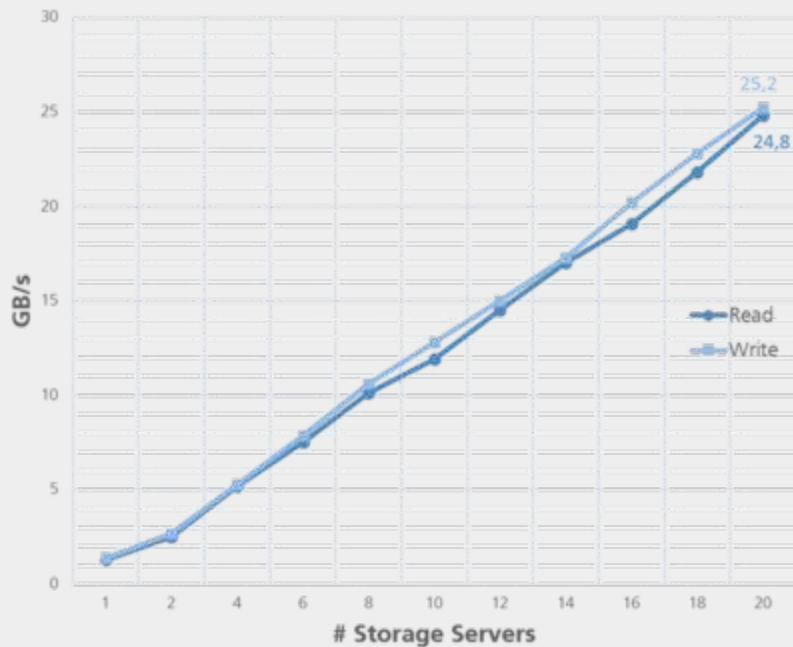
Combina la capacidad de varios servidores independientes

Operaciones sobre datos también escalan

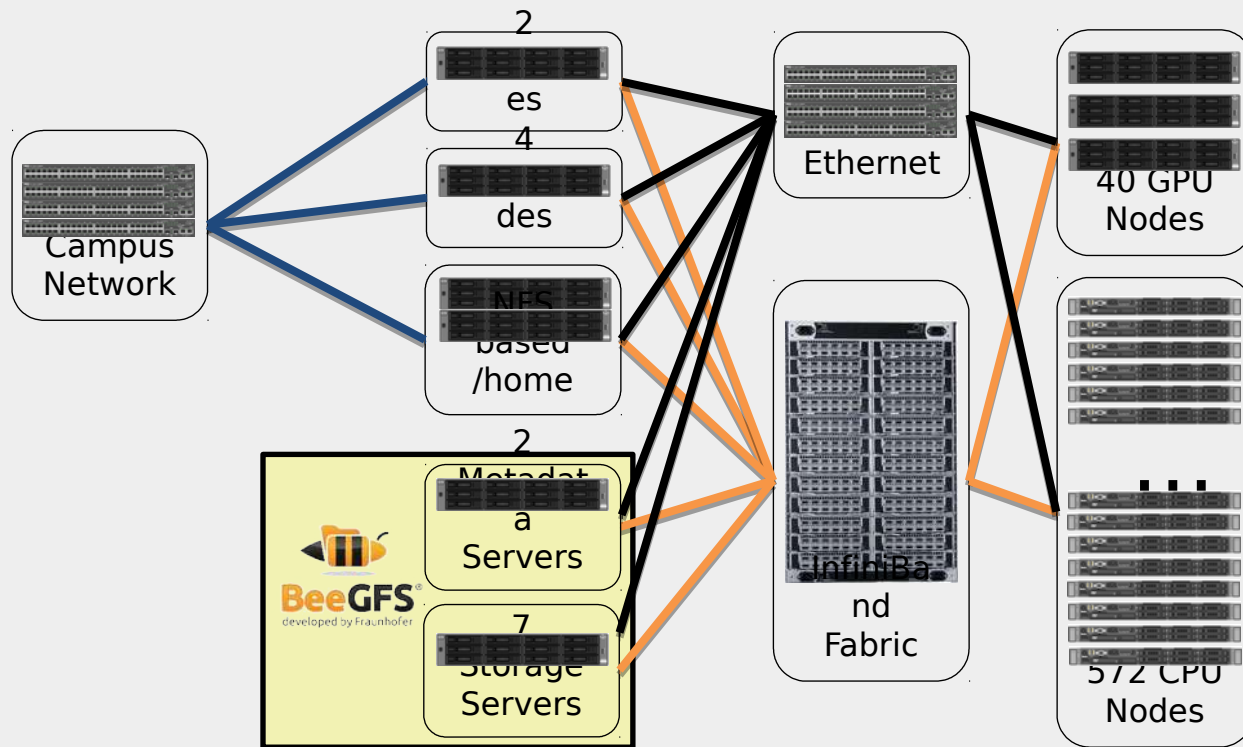




# BeeGFS escalabilidad

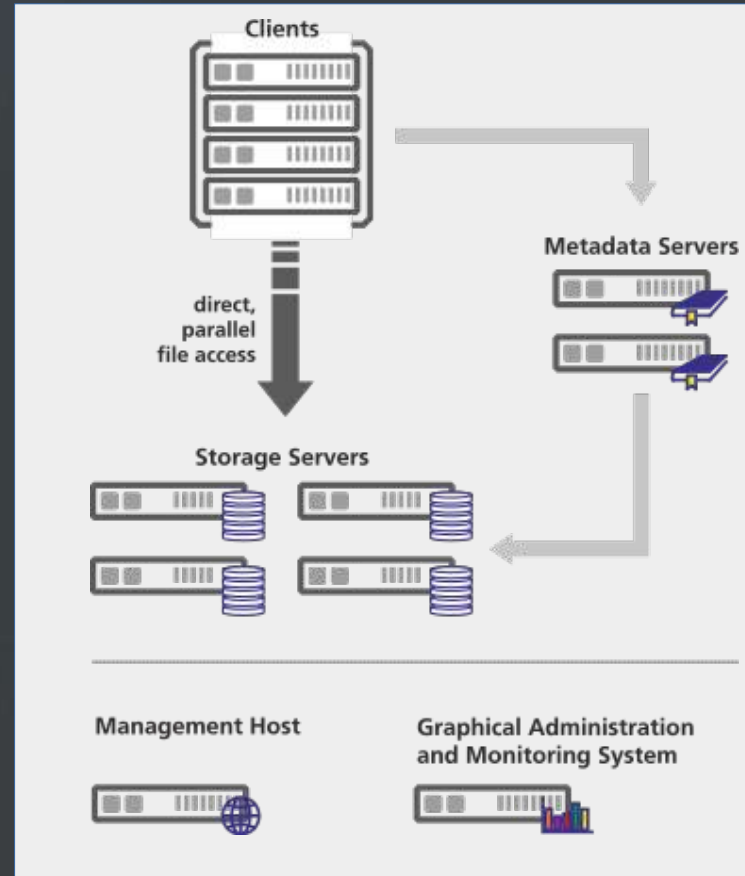


# BeeGFS escalabilidad



# BeeGFS componentes

1. Management Server (MD).
2. MetaData Server (MDS).
3. Object Storage Server (OSS).
4. File System Client.

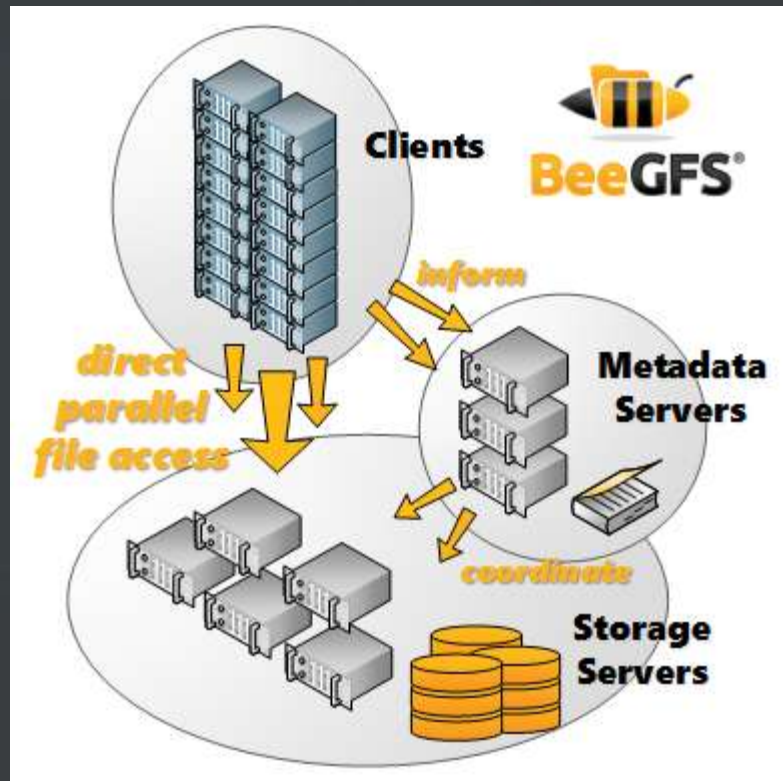


# BeeGFS metadata server

Sirve de catálogo para el sistema de ficheros.

Al igual que el almacenamiento (storage) es capaz de acomodarse a la demanda

Componente crítico: se recomienda redundancia (RAID 1 o equivalente)



# BeeGFS object storage server

Responsable del almacenamiento de bajo nivel

Cada OSS debe tener uno o varios ObjectStorageTarget (OST).

Se recomienda RAID 6 con entre 6 y 12 discos.

Se apoya en sistemas de ficheros POSIX, como ext4 o xfs.

Stripping: los ficheros a nivel cliente se distribuyen entre varios OST



# BeeGFS file system client

El único componente de BeeGFS que depende directamente del kernel es el cliente

Demonios:

- Beegfs-helperd – funciones auxiliares para beegfs-cliente.
- Beegfs-client – gestión de carga del módulo del kernel del cliente (si fuera necesario recompila).

**BeeGFS** logs

/var/log/beegfs-\*.log

# BeeGFS beegfs-ctl

```
client01:~ # beegfs-ctl --help
BeeGFS Command-Line Control Tool (http://www.beegfs.com)
```

## GENERAL USAGE:

```
$ beegfs-ctl --<modename> --help
$ beegfs-ctl --<modename> [mode_arguments] [client_arguments]
```

## MODES:

|                      |  |
|----------------------|--|
| --listnodes          | => List registered clients and servers.        |
| --listtargets        | => List metadata and storage targets.          |
| --removenode (*)     | => Remove (unregister) a node.                 |
| --removetarget (*)   | => Remove (unregister) a storage target.       |
| --getentryinfo       | => Show file system entry details.             |
| --find               | => Find files located on certain servers.      |
| --migrate            | => Migrate files to other storage servers.     |
| --serverstats        | => Show server IO statistics.                  |
| --clientstats        | => Show client IO statistics.                  |
| --userstats          | => Show user IO statistics.                    |
| --storagebench (*)   | => Run a storage targets benchmark.            |
| --getquota           | => Show quota information for users or groups. |
| --setquota (*)       | => Sets the quota limits for users or groups.  |
| --listmirrorgroups   | => List mirror buddy groups.                   |
| --addmirrorgroup (*) | => Add a mirror buddy group.                   |

# BeeGFS file system client

```
beegfs-ctl --listnodes --nodetype=storage
```

```
beegfs-ctl --listnodes --nodetype=storage -nicdetails
```

```
beegfs-df
```

```
beegfs-fsck --checkfs [--readonly | --automatic]
```

```
beegfs-ctl --setquota --uid 1002 --sizelimit=100M  
--inodelimit=unlimited
```

```
beegfs-ctl --getquota --uid 1002
```

```
beegfs-ctl --setpattern --chunksize=1m --numtargets=4  
/mnt/beegfs/test
```

# BeeGFS tolerancia a fallos

Escenario: se distribuyen los servicios entre varios servidores

Fallo en servidor de almacenamiento o metadatos

- No se pierden datos
- Los ficheros alojados en él quedan inaccesibles (stripping, timeout)

Replicas (mirrorgroups & mirrormd beegfs-ctl)

Opción de configuración en alta disponibilidad (salvo servicio de administración)

# Datos Ciclo de vida

# Datos ciclo de vida

Producción / captura

Ingesta

Procesamiento

Réplica (seguridad, copia, versiones)

Visualización

Archivado

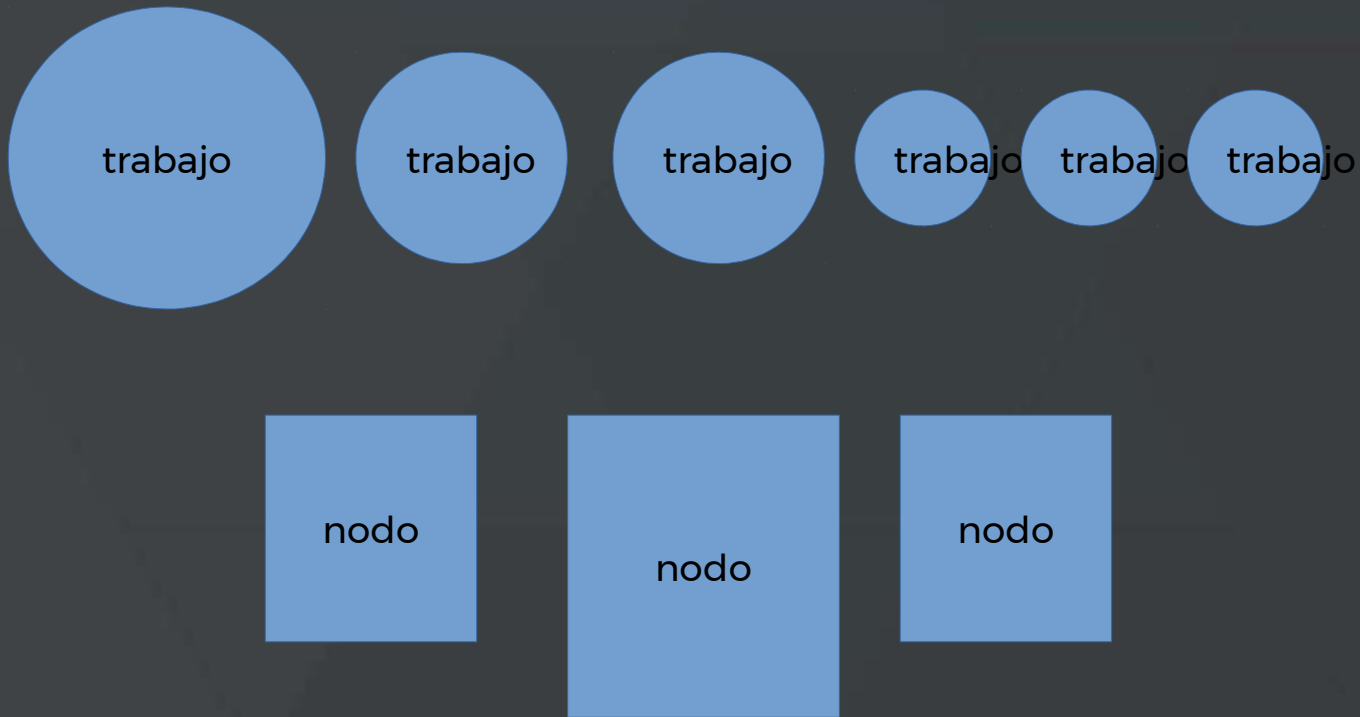
Borrado

# SLURM

## sistema de colas



# Slurm problema



# Slurm solución

trabajo

trabajo

trabajo

trabajo

trabajo

trabajo



nodo

nodo

nodo



# Slurm

Simple Linux Utility for Resource Management

<<Slurm es un sistema de gestión de clústers y planificación de trabajos escalable, tolerante a fallos y de código abierto.

No requiere cambios en el kernel y su uso es autocontenido.

Como gestor de carga, Slurm cumple tres funciones principales...

Primero, reserva el acceso a los recursos (nodos de cómputo) para los usuarios durante un tiempo, de modo que puedan realizar su trabajo.



# Slurm

Segundo, ofrece una plataforma para iniciar, ejecutar y monitorizar trabajos en un conjunto de nodos reservados.

Por último, arbitra en la competición por los recursos mediante una cola que gestiona los trabajos pendientes. Se puede recurrir a plugins opcionales para contabilidad, planificación, priorización...>>

<https://slurm.schedmd.com/overview.html>



# Slurm

Distribuido (cliente/servidor)

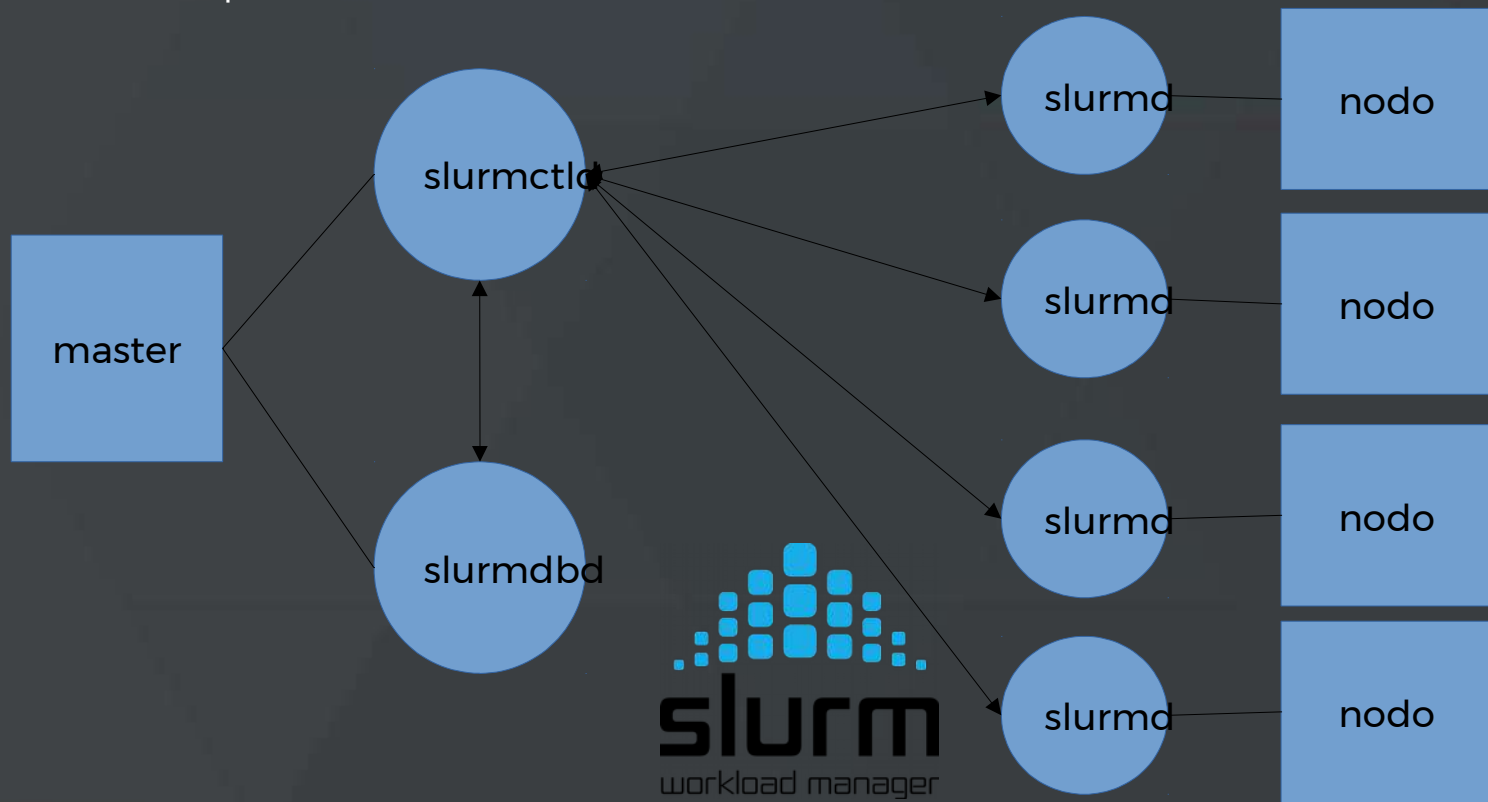
Versátil

Potente

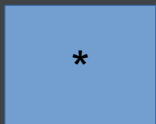
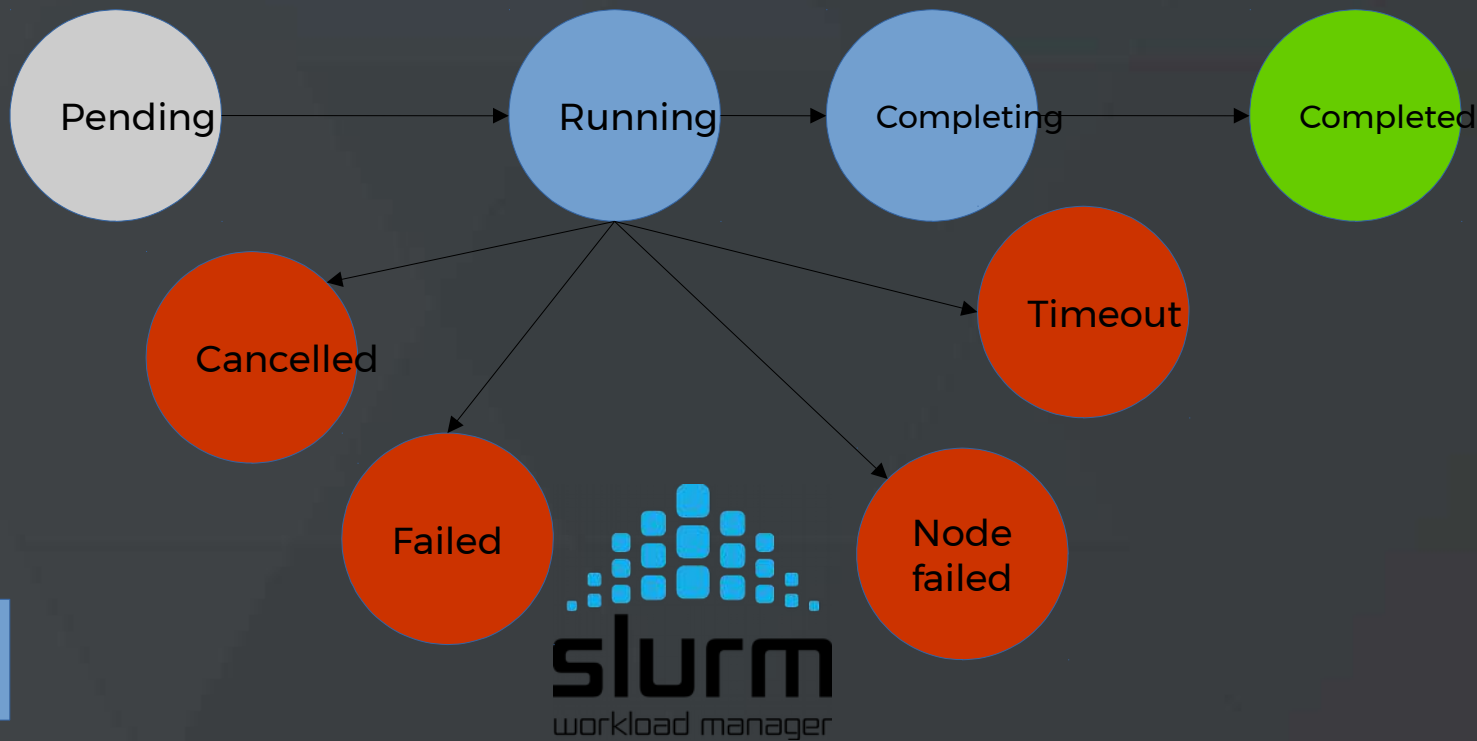
Vivo (2003-presente)



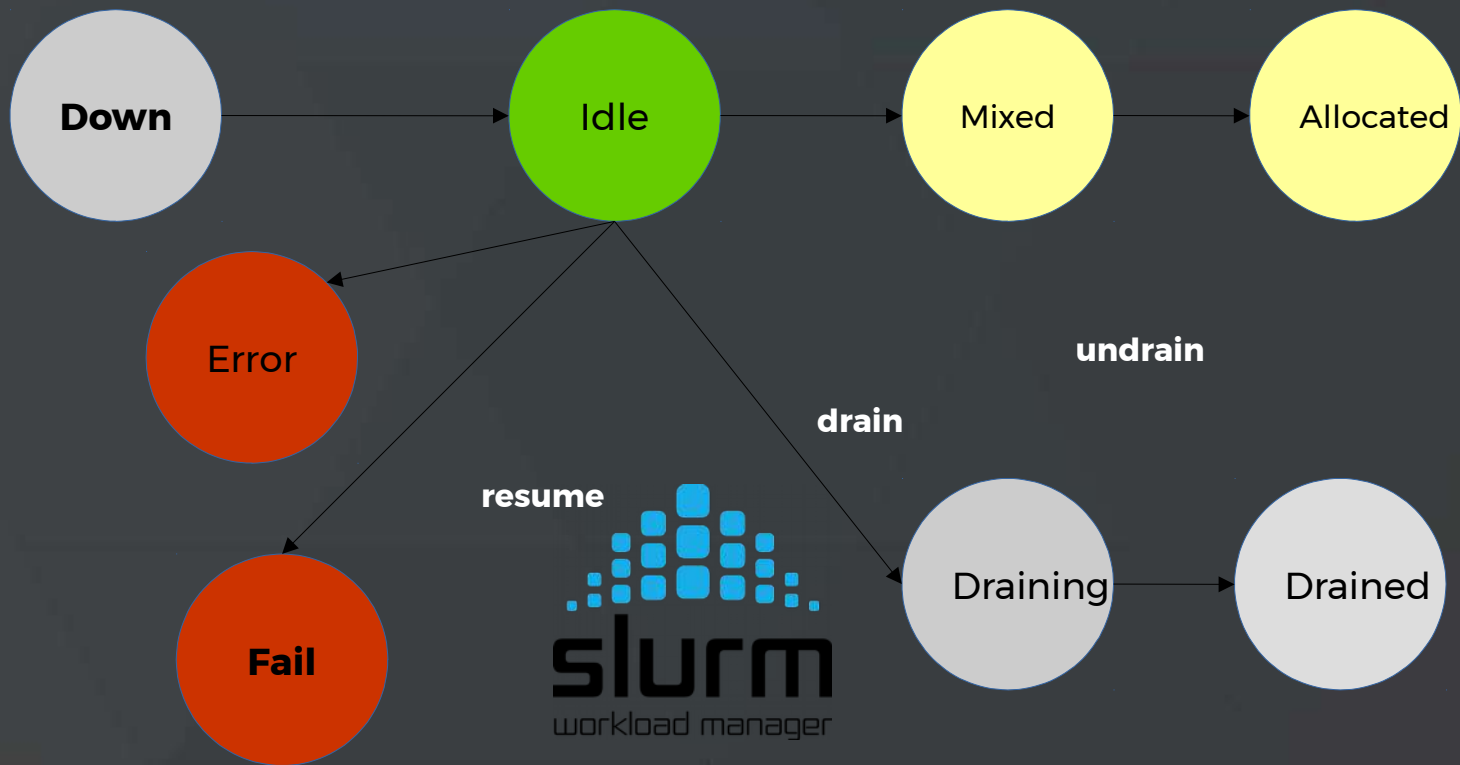
# Slurm arquitectura



# Slurm job states



# Slurm node states





# Slurm



srun / sbatch  
squeue / scancel  
scontrol / sinfo  
sacct



# Slurm configuración

/etc/slurm/slurm.conf

(idéntica en todos los nodos)

👁 <https://slurm.schedmd.com/slurm.conf.html>

## Parámetros:

ClusterName

DefMemperCPU / DefMemPerNode

MaxMemPerCPU / MaxMemPerNode

GresTypes

ReturnToService={0 ▶ manual / 2 ▶ auto}

JobRequeue={0 ▶ no / 1 ▶ yes}

# Slurm configuración

```
NodeName=nodo[01-08]  
  Procs=16  
  SocketsPerBoard=2  
  CoresPerSocket=8  
  ThreadsPerCore=1  
  RealMemory=257674  
  MemSpecLimit=4096  
  Gres=gpu:2  
  State=UP
```

# Slurm configuración

```
PartitionName=ladon  
Nodes=nodo[01-08]  
Default=YES  
MaxNodes=2  
MaxTime=7-0  
State=UP  
PreemptMode=OFF  
AllowAccounts / AllowGroups
```

# Slurm configuración

```
SlurmctldLogFile=/var/log/slurmctld.log  
SlurmctldDebug={info / verbose / debug..debug5}
```

```
SlurmdLogFile=/var/log/slurmd.log  
SlurmdDebug={info / verbose / debug..debug5}
```

# Slurm operaciones

Lanzar un trabajo

```
sbatch <nombre-script>
```

Información sobre trabajos en la cola

```
squeue
```

Eliminar un trabajo de la cola

```
scancel <JOBID>
```

De forma automática busca los UID de los procesos en cada nodo y los mata.

# Slurm descripción de trabajos

Secuencia de pasos que componen el trabajo

Parámetros globales (sbatch)

Parámetros de los pasos (srun)

Los parámetros se pueden facilitar en el fichero de descripción o en la llamada al comando (predomina en la llamada)

# Slurm parámetros sbatch

<https://slurm.schedmd.com/sbatch.html>

Duración y programación horaria:

```
--time={DD-HH / HH:MM:SS}  
--begin=YYYY-MM-DDTHH:MM:SS  
--deadline=YYYY-MM-DDTHH:MM:SS
```

Entorno

```
--workdir=  
--output=align%j.out  
--error=  
--export  
--job-name=
```



# Slurm parámetros sbatch

Cola

--partition=

Tareas

--ntasks=

Distribución de tareas

--tasks-per-node=

--odelist=nodo03,nodo06

# Slurm parámetros sbatch

## Recursos

### CPU

--cpus-per-task=

### Genéricos (GPU)

--gres=

### Memoria

--mem=<size><unit>

--mem-per-cpu=<size><unit>

# Slurm parámetros sbatch

## Aviso

```
--mail-type=end  
--mail-user=direccion_email
```

## Dependencias

```
--dependency=  
    after:JOB_ID  
    afterok:JOB_ID  
    afternotok:JOB_ID  
    singleton
```

# Slurm parámetros sbatch

## Variables informativas

```
$SLURM_JOB_ID  
$SLURM_JOB_NAME  
$SLURM_JOB_NODELIST  
$SLURM_JOB_PARTITION  
$SLURM_SUBMIT_DIR  
$SLURM_SUBMIT_HOST  
$SLURMD_NODENAME
```

# Slurm parámetros srun

## Tareas

--ntasks=1

--tasks-per-node=

--cpus-per-task=

--gres=

--mem=<size><unit>

--mem-per-cpu=<size><unit>

## Slurm trabajos/secuencial

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

```
#SBATCH -time=01:00:00
```

```
srun PAS01
```

```
srun PAS02
```

```
sbatch secuencial.sbs
```

# Slurm trabajos/paralelo

```
#!/bin/bash  
#SBATCH --time=01:00:00  
srun --exclusive -n 1 PAS01 &  
srun --exclusive -n 1 PAS02 &  
wait
```

```
sbatch --ntasks=2 paralelo.sbs
```

# Slurm trabajos/paralelismo multihilo

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

```
#SBATCH --ntasks=1
```

```
#SBATCH --cpus-per-task=4
```

```
#SBATCH -time=06:00:00
```

```
srun PROGRAMA -t 4
```



# Slurm trabajos/OpenMPI

```
#!/bin/bash
#SBATCH --tasks=20
#SBATCH --tasks-per-node=10

#SBATCH -time=06:00:00

module load mpi/openmpi-x86_64
mpirun PROGRAMA_OPEN_MPI
```

# Slurm trabajos/GPU

```
#!/bin/bash  
#SBATCH --partition gpu  
#SBATCH --gres=gpu:1  
#SBATCH --ntasks 1  
srun PROGRAMA_GPU
```

# Slurm sinfo

sinfo

sinfo -s

NODES(A/I/O/T) ► Allocated/Idle/Other/Total

sinfo -Ne1

# Slurm scontrol

Actualizar cambios configuración

reconfigure

Trabajos

show job JOB\_ID

suspend JOB\_ID / resume JOB\_ID

hold JOB\_ID / release JOB\_ID

# Gestión del software

# Gestión de software introducción

El universo Linux gira en torno al software de código abierto.

Inicialmente se recurre a la descarga y compilación.

Las distribuciones introdujeron el concepto de paquete binario.

# Gestión de sw yum

Las distribuciones de Linux ofrecen diversos gestores de paquetes: apt, yum, pacman...

Red Hat y sus derivados (como CentOS) ofrecen yum, una evolución de rpm (Red Hat Package Manager).

yum es una herramienta del sistema operativo que actúa a nivel local.

# Gestión de sw yum

| Acción                                 | Comando  |
|--|--|
| Buscar paquete                         | yum search   |
| Listar paquetes instalados             | yum list installed                                   |
| Listar paquetes a actualizar           | yum list updates                                     |
| Instalar paquete                       | yum install <b>nombre-paquete</b>                    |
| Desinstalar paquete                    | yum remove <b>nombre-paquete</b>                     |
| Instalar/desinstalar grupo de paquetes | yum groupinstall/groupremove " <b>nombre-grupo</b> " |
| Actualizar paquete / grupo             | yum update <b>nombre</b>                             |



# Gestión de sw cluster

Para poder usar transparentemente una aplicación  
en cualquier nodo del cluster...  
tiene que estar disponible en todos los nodos

¿Cómo instalar software que no está en yum?

# Gestión de sw spack

Herramienta que permite el despliegue versátil de software científico



Comandos:

`spack list PATRON`

`spack info PAQUETE`

`spack versions PAQUETE`

`spack find`

`spack load PAQUETE`

`spack unload PAQUETE`

`spack install PAQUETE`

# Gestión de sw cluster

Para poder usar transparentemente una aplicación  
en cualquier nodo del cluster...  
tiene que estar disponible en todos los nodos

Sistemas de ficheros compartidos:

`/mnt/beegfs/software`  
(paquete/versión)

`home usuario`

# Gestión de sw modules

Herramienta que permite la convivencia de varias versiones en paralelo de una misma aplicación

Comandos:

`modules available`

`module load MODULO`

`module list`

# Gestión de sw modules

## Configuración

`/etc/profile.d/modules.sh`

`/mnt/beegfs/software/modules`

# Gestión de sw modules (tcl)

```
#%Module1.0
##
## Omega 2.5.1.4

module-whatIs "Omega 2.5.1.4"

proc ModulesHelp { } {
    puts stderr "Omega 2.5.1.4"
}

prepend-path PATH "/mnt/beegfs/software/omega/2.5.1.4/openeye/arch/redhat-
RHEL6-x64/omega"
setenv OE_LICENSE
"/mnt/beegfs/software/omega/2.5.1.4/openeye/oe_license.txt"
#prepend-path LIBRARY_PATH "/mnt/beegfs/software"
#prepend-path LD_LIBRARY_PATH "/mnt/beegfs/software"
#prepend-path CPATH "/mnt/beegfs/software"
#prepend-path PKG_CONFIG_PATH "/mnt/beegfs/software"
#prepend-path CMAKE_PREFIX_PATH "/mnt/beegfs/software"
```

# Gestión de sw modules (lua)

```
help([[  
MOE 2014  
]])
```

```
whatis("Version: 2014")  
whatis("Description: MOE")
```

```
-- if not isloaded("a") then  
--     load("a")  
-- end
```

```
setenv("MOE", "/mnt/beegfs/software/moe/201409/moe2014")  
prepend_path("PATH", "/mnt/beegfs/software/moe/201409/moe2014/bin-lnx64")
```

```
-- prepend_path("LIBRARY_PATH", "/mnt/beegfs/software")  
-- prepend_path("LD_LIBRARY_PATH", "/mnt/beegfs/software")  
-- prepend_path("MANPATH", "/mnt/beegfs/software")  
-- prepend_path("CPATH", "/mnt/beegfs/software")  
-- prepend_path("PKG_CONFIG_PATH", "/mnt/beegfs/software")  
-- prepend_path("CMAKE_PREFIX_PATH", "/mnt/beegfs/software")
```

# Seguridad



# Seguridad repasando riesgos

Los riesgos a afrontar son múltiples:

- Ataques a los servicios del nodo principal
- Impersonación de usuarios legítimos
- Troyanos / botnets
- DATOS
  - Pérdida de datos (accidental o intencionada)
  - Robo / secuestro información

# Seguridad ataques

- Restricción acceso al cluster
  - Red privada (nodos cálculo)
  - Cortafuegos (nodo principal) / banning
  - Red privada (nodo principal)
- 
- Servicios públicos “seguros”
  - Actualizaciones

# Seguridad cortafuegos CentOS

Restringe el acceso desde el exterior a los servicios del nodo principal.

## Ejemplos:

```
firewall-cmd --state
```

```
firewall-cmd --list-all-zones
```

```
firewall-cmd --list-services
```

```
firewall-cmd --zone=public --add-service=http --permanent
```

```
firewall-cmd --zone=public --add-service=http
```

```
firewall-cmd --zone=trusted --add-source=192.168.2.0/24
```

# Seguridad risky software

Medidas para paliar el riesgo usando programas

- Recurrir a fuentes fiables
- Código fuente
- Ejecución restringida

Medida para paliar el riesgo de perder datos:

## Copia de seguridad automática

periódica + diferencial  
múltiple  
silo seguro  
archivado

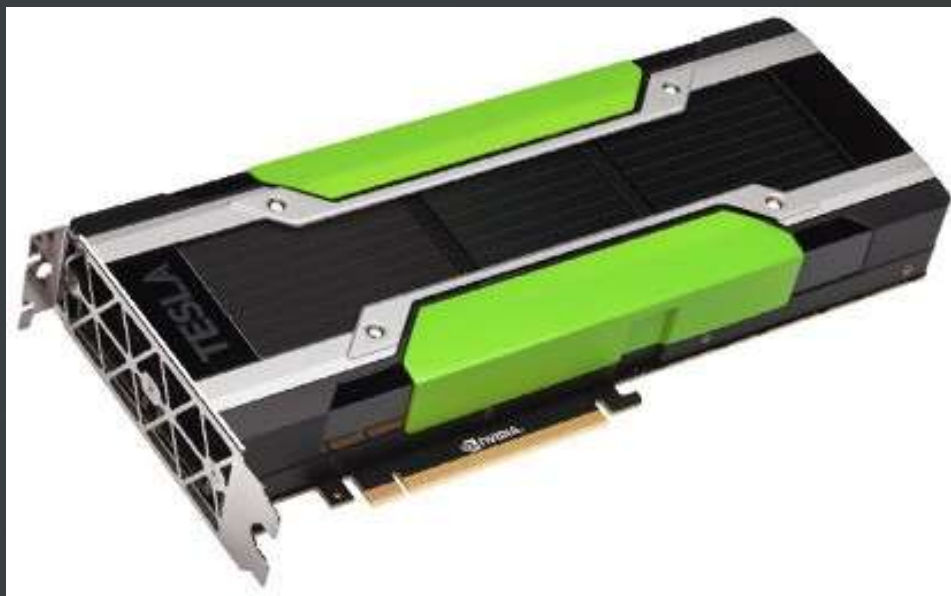
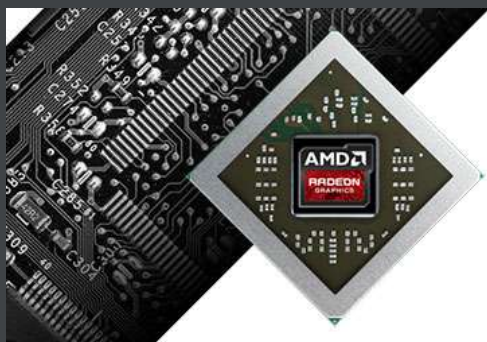
# Seguridad datos

Medidas para paliar el riesgo de robo de datos

- Autorización: permisos
- Caducidad de cuentas
- Cifrado

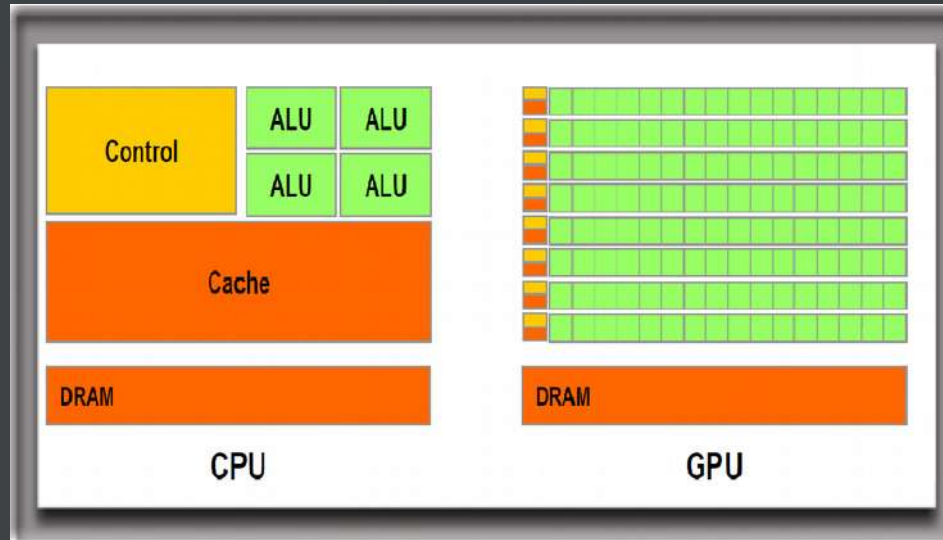
# GPU

# GPU introducción





# GPU arquitectura hardware



RAM (256 GB) vs VRAM (12 GB)

1 CPU x 20 cores vs 1 GPU x 4000 microcores

700 GB/s (vRAM) vs 25 GB/s (DDR4)

# GPU arquitectura software (CUDA)

Bajo nivel: driver Nvidia nativo (módulo kernel)

Plataforma CUDA: API (bibliotecas) + compilador + herramientas  
Rendimiento depende del nivel de optimización del código

Instalación: yum / descarga web Nvidia

/usr/local/cuda-X.Y  
/usr/local/cuda

Entorno

# Instalación nodos kickstart

# Instalación nodos kickstart

Sistema de instalación automática por red

El nodo obtiene la configuración IP y de arranque (bootp) a partir del servicio DHCP del nodo principal.

Se transfiere un cargador de sistema por TFTP

Se descarga el script de instalación por HTTP

Los paquetes base se instalan por HTTP, a partir del repositorio del nodo principal

Terminada la instalación básica, se dispone de un sistema CentOS base en el cual se realizan las tareas de post-instalación

# Instalación nodos kickstart

Normalmente la instalación se hace desatendida

Se puede hacer un seguimiento a traves de la consola IPMI

Kickstart resulta adecuado para la instalación base.

Para el mantenimiento de nodos resulta más práctico usar herramientas como clustershell o ansible

# Monitorización

# Monitorización ganglia

<http://ganglia.info>

<https://github.com/ganglia/monitor-core/wiki>

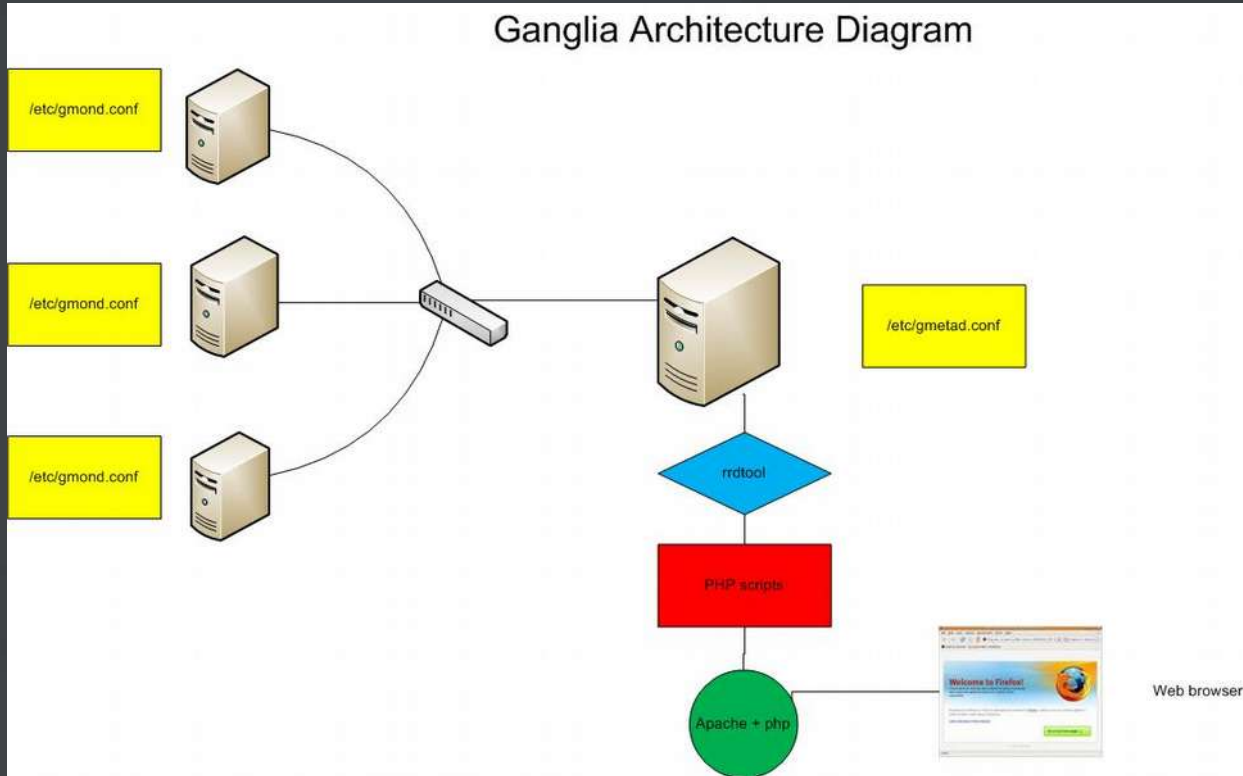
Sistema de monitorización escalable y distribuido para sistemas HPC (2001)

Interfaz web que resume factores de uso como:

- Cargas de trabajo de los nodos.
- Históricos de picos de trabajo en nodos.
- Uso de memoria.



# Monitorización ganglia





# Monitorización ganglia

[http://<IP\\_NODO\\_00>/ganglia](http://<IP_NODO_00>/ganglia)



# Monitorización ganglia

Configuración

Web access: /etc/httpd/conf.d/ganglia.conf

Meta daemon: /etc/ganglia/gmetad.conf

Monitoring daemon: /etc/ganglia/gmond.conf



# CLUES

## green computing

# CLUES green computing

Apagar los nodos de cómputo cuando no están siendo utilizados

Encenderlos de nuevo cuando son necesarios.

Para ello se integra con el middleware de gestión de recursos del cluster.



# CLUES caso de uso: odin

Cluster HPC del grupo de investigación GryCAP.

Alterna picos de uso (realización de pruebas para la publicación de algún artículo) y de infrautilización (periodos vacacionales).

| Sin CLUES |        |        |                   |
|-----------|--------|--------|-------------------|
| Estado    | Pct    | kWh    | € *               |
| Apagado   | 0%     | 0      | 0 €               |
| Ocioso    | 69,08% | 25.122 | 2.286,12 €        |
| Usado     | 30,92% | 12.805 | 1.165,27 €        |
| TOTAL     | 100%   | 37.927 | <b>3.451,39 €</b> |

| Con CLUES |        |        |                   |
|-----------|--------|--------|-------------------|
| Estado    | Pct    | kWh    | € *               |
| Apagado   | 65,67% | 5.970  | 543,27 €          |
| Ocioso    | 3,42%  | 1.242  | 113,02 €          |
| Usado     | 30,92% | 12.805 | 1.165,27 €        |
| TOTAL     | 100%   | 20.017 | <b>1.821,57 €</b> |

\* Coste: 0,091 €/kw. Dato obtenido del Ministerio de Industria, Turismo y Comercio del Gobierno de España.

# CLUES arquitectura

cluesd + cluesserver

/var/log/clues2

Conectores con middlewares de gestión (plugins)

Herramienta linea de comandos (clues)

# CLUES plugins

Ipmi – apagado / encendido nodos

Slurm – interacción con sistema de colas

# CLUES comando clues

status

enable NODE

disable NODE

poweron NODE

poweroff NODE

shownode NODE



# Laboratorio

# Laboratorio

Conexión al cluster

```
ssh demo@hpc.unav.es
```

Carpeta de trabajo

```
mkdir usuario
```

(opcional) subir / descargar ficheros

Filezilla

```
wget URL
```

# Laboratorio

## Software (Spack)

`spack list`

`spack info PAQUETE`

`spack versions PAQUETE`

`spack find`

`spack load PAQUETE`

# Laboratorio

Software (modules)

modules available

module whatis MODULO

module help MODULO

module list

module load MODULO

module list

# Laboratorio

Slurm: fichero de trabajo

```
--job-name=nombre  
--ntasks=
```

```
--cpus-per-task=
```

```
--time=01:00:00
```

```
--mem=<size><unit>
```

```
$SLURM_JOB_ID
```

```
$SLURM_JOB_NAME
```

```
$SLURM_SUBMIT_DIR
```

Entorno: spack / module

```
date ; sleep 60 ; date
```

# Laboratorio

Slurm: comandos

```
sbatch desc_trabajo.sbs
```

```
(id trabajo)
```

```
squeue
```

```
scancel
```

# Gracias

# Contacto



Sistemas Informáticos Europeos

Calle Marqués de Mondejar nº 29

913 61 10 02

[www.sie.es](http://www.sie.es)



/HPCSIE



[soporte@sie.es](mailto:soporte@sie.es)



@HPCSIE



+SistemasInformaticosEuropeosSLMadrid

