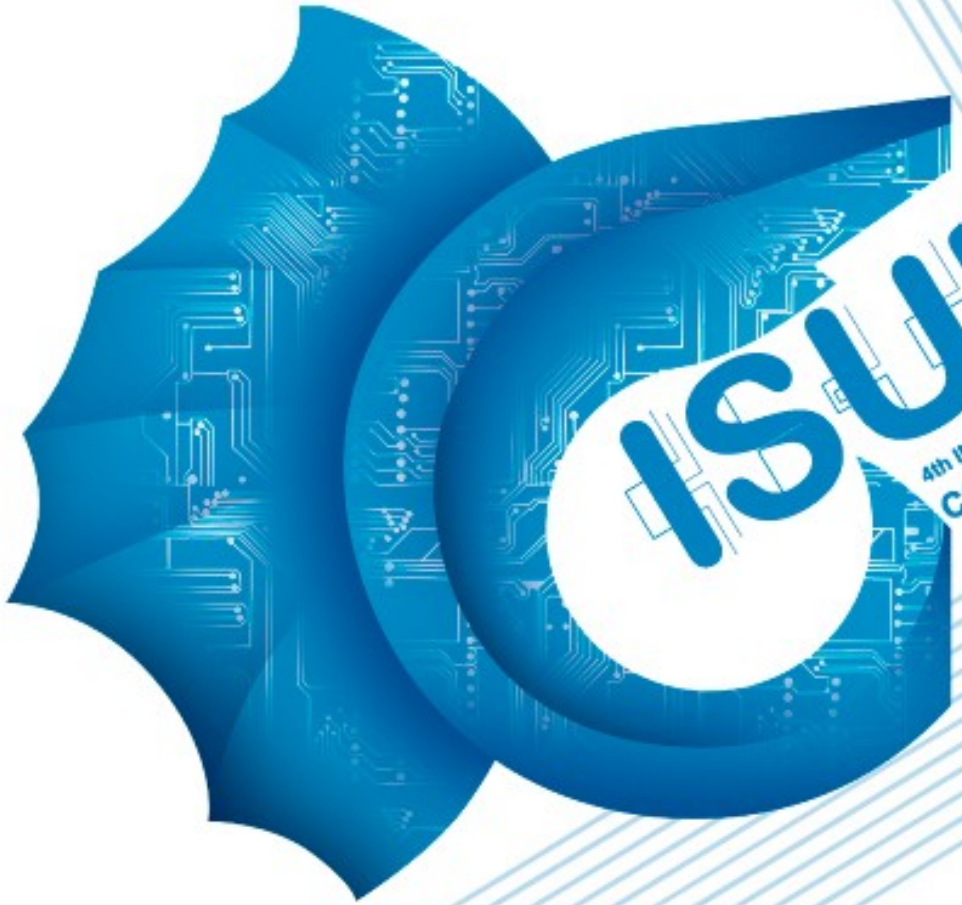



# IMPLEMENTACIÓN DE UN CLUSTER

www.isum.mx

A Navigation Through Science and Technology

DEL **5 AL 8 MARZO, MANZANILLO, COLIMA**. MÉXICO.

# TEMARIO

- Cluster?
  - Que es?
  - Para que sirve?
  - Como funciona?
  - Conceptos basicos.
  - Tipos
  - Componentes
- Adminstracion y monitoreo
  - Administrador de Recursos.
  - Planificador.
- Implementación
  - Comunicaciones
  - Almacenamiento
  - Procesamiento en paralelo
  - Torque

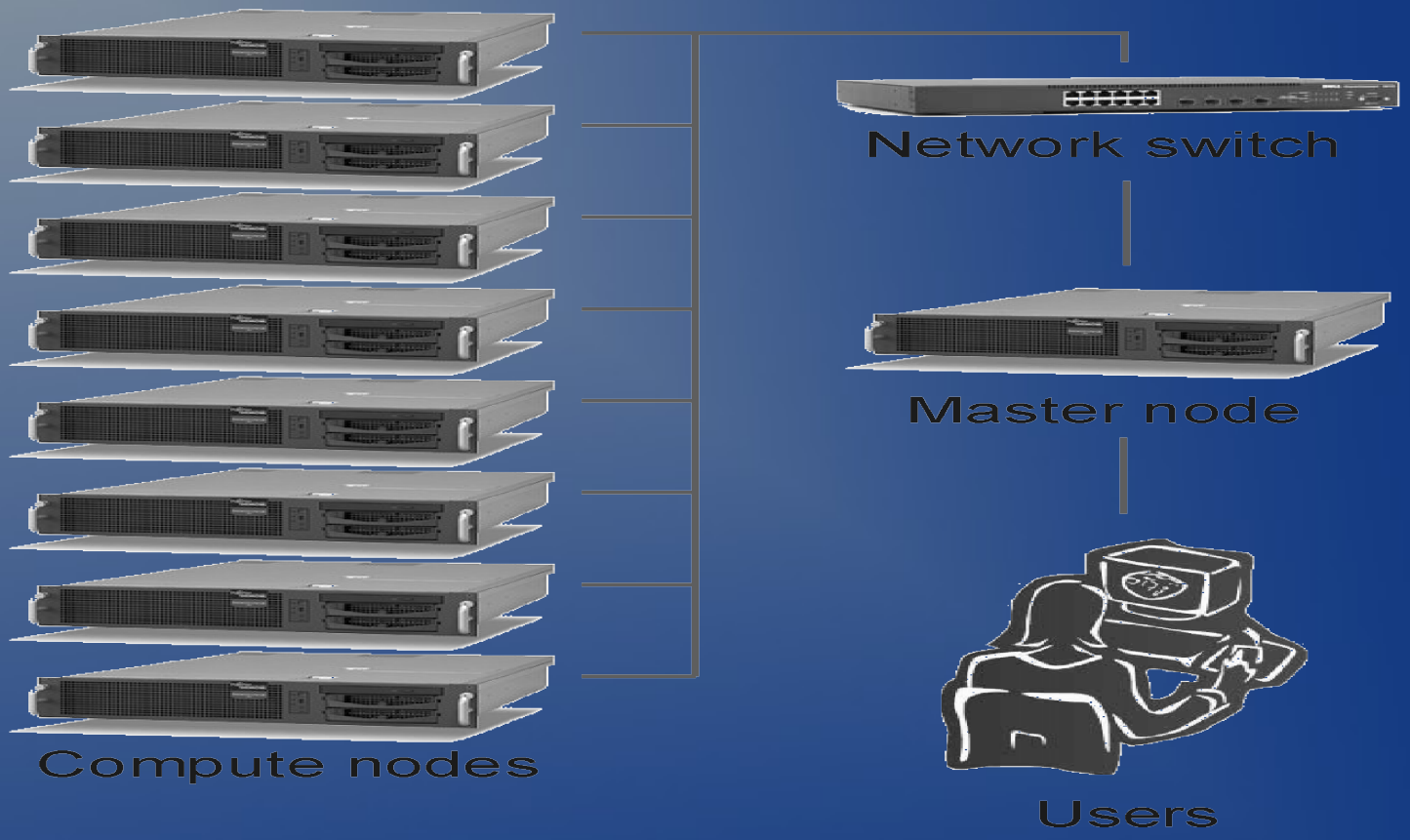
# IMPLEMENTACIÓN DE UN CLUSTER

Hoy en día los clúster en Linux para HPC (High Performance Computing) son muy populares, ya que están al alcance de todos y se consideran como una herramienta fundamental para el desarrollo de la investigación.

# QUE ES UN CLUSTER??

# PARA QUE SIRVE UN CLUSTER??

# COMO FUNCIONA??



# CONCEPTOS BASICOS

Rendimiento: Es la efectividad del desempeño de una computadora, sobre una aplicación o un benchmark en particular.

Flops: Es una medida de la velocidad del procesamiento numérico del procesador. Son operaciones de punto flotante por segundo.

Latencia: Tiempo de transferencia de mensajes de una interfaz a otra, se debe procurar que ésta sea la mínima posible.

Ancho de Banda: Capacidad de transferencia que tiene un canal de comunicaciones en una unidad de tiempo.

# TIPOS DE CLUSTERS

# POR PROPOSITO DE TRABAJO

* LB (Load Balancing).

* HA (High Availability).

* HP (High Performance).

# SEGUN LA CONFIGURACION DE LOS NODOS

Basados en SO

Heterogéneos

Homogéneos.

Dedicado

No Dedicado.

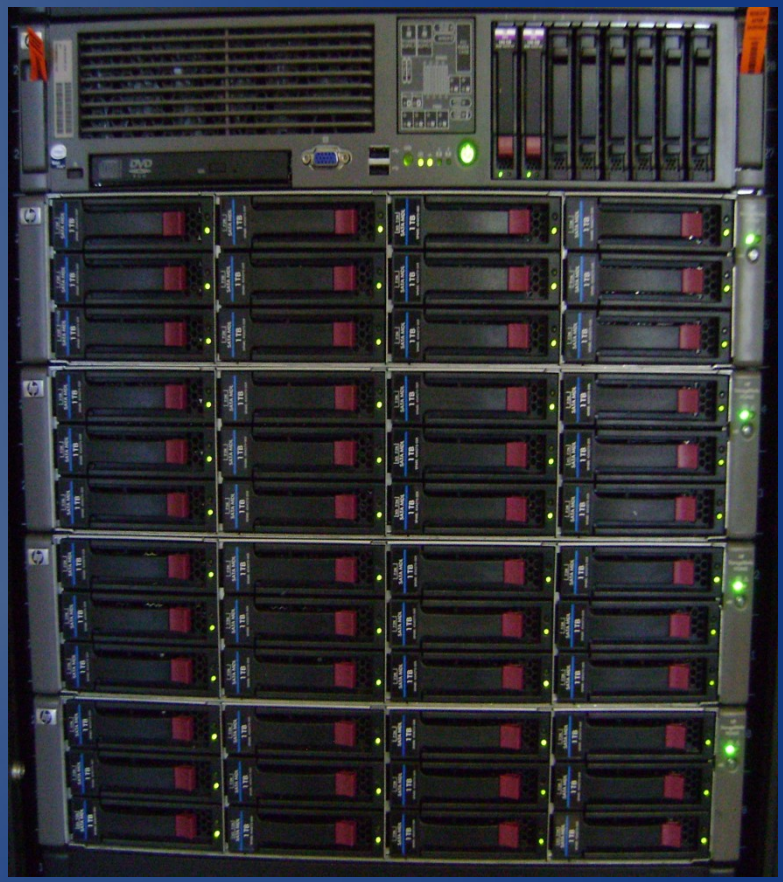
# COMPONENTES

# PROCESAMIENTO

Está integrada por los servidores que gestionan las conexiones desde el exterior, ya sea que se encargue del acceso a usuarios, proxy o balanceo de carga, y por todos y cada uno de los nodos de cómputo.

# ALMACENAMIENTO

Se compone del sistema o servidores que administran y controlan el almacenamiento de la información que se genera en los nodos de procesamiento.



# COMUNICACIONES

Se integra por los dispositivos que permiten la conectividad entre los servidores de procesamiento y almacenamiento, en este apartado se consideran dispositivos como routers, switches y cables de comunicación, para cada una de las redes ( de administración, producción y almacenamiento) que tiene un clúster.

# TIPOS DE IMPLEMENTACION

# ADMINISTRADA

La instalación de un clúster de manera administrada conlleva una larga serie de pasos que van desde el estudio de las condiciones de la infraestructura y site, como son, las condiciones del edificio y el sistema eléctrico hasta el montaje físico de los servidores, clableado eléctrico y de datos.

# NO ADMINISTRADA

CIA

El ambiente CIA (clúster Installation and Administration) fue desarrollado para simplificar en lo mas posible el manejo de un clúster y a la vez proporcionar un sistema integral para su manejo.

Los objetivos fueron flexibilidad, robustez, portabilidad y facil mantenimiento, configuración centralizada y modificable.

# NO ADMINISTRADA

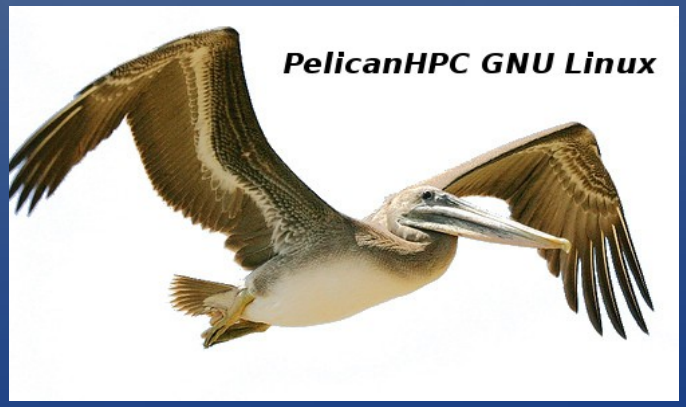
ROCKS

Rocks es un “clúster completo en un CD”, es una solución para clústers Linux Red x86 y x86_64.

# NO ADMINISTRADA

## Pelican HPC

Pelican HPC es un live CD que permite configurar un clúster HPC en minutos. PelicanHPC es una imagen basada en Debian que tiene el objetivo de hacer que sea fácil de configurar un clúster de cómputo de alto rendimiento.



# NO ADMINISTRADA

Scientific Linux

Es una recompilación Red Hat Enterprise Linux, co-desarrollado por Fermi National Accelerator Laboratory and the European Organization for Nuclear Research (CERN).

# NO ADMINISTRADA

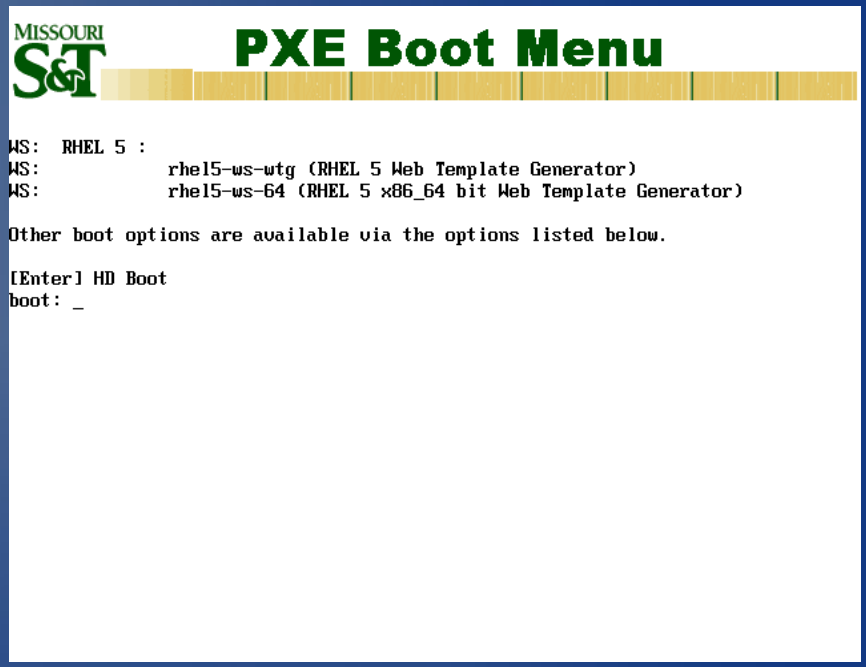
Kickstart

Es usado en muchos sistemas Linux, el cual permite una instalación desatendida y la configuración consistente de los nuevos sistemas operativos de forma automática.

# NO ADMINISTRADA

PXE

Preboot eXecution Environment (PXE) es un servicio que permite arrancar e instalar el sistema operativo GNU/Linux a través de una red



# ADMINISTRACIÓN Y MONITOREO DE UN CLUSTER

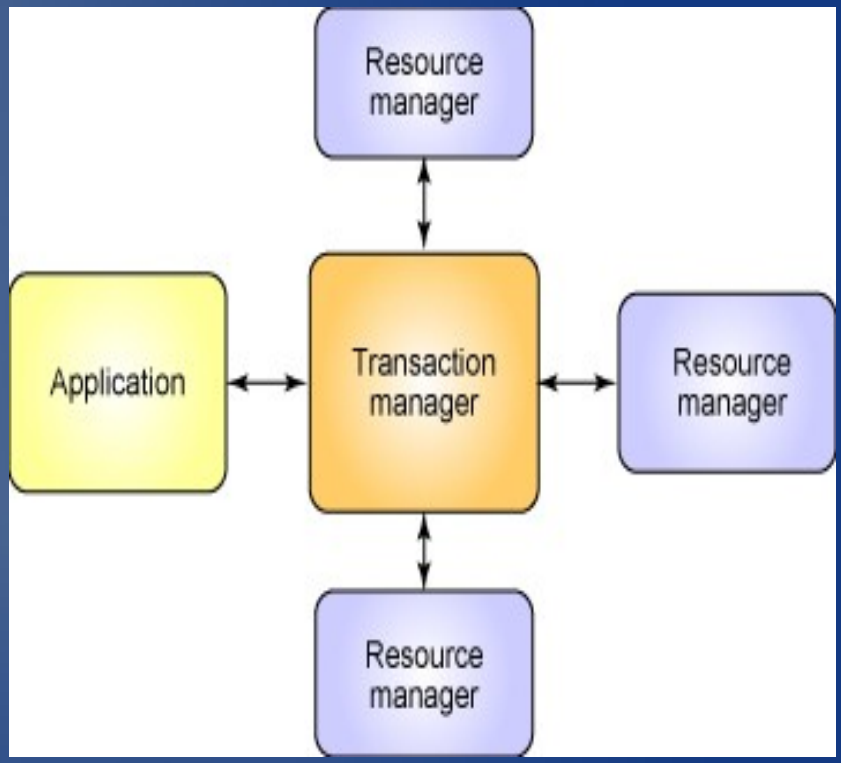
# INSTALACIÓN DE CLUSTER

Para comenzar nuestra instalación necesitamos paquetes o repositorios que el sistema GNU/LINUX nos provee en algunos casos desde nuestra instalación y en otros nosotros tenemos que instalarlos y configurarlos.

Los primero que veremos seran las comunicaciones ya que sin ellas nos podemos ver a nuestros nodos de procesamiento.

# ADMINISTRADOR DE RECURSOS

Un gestor de recursos es un sistema de administración, control y monitoreo de todos los recursos de cómputo colas de ejecución y distribución de trabajos que se envían al clúster.



# ADMINISTRADOR DE RECURSOS

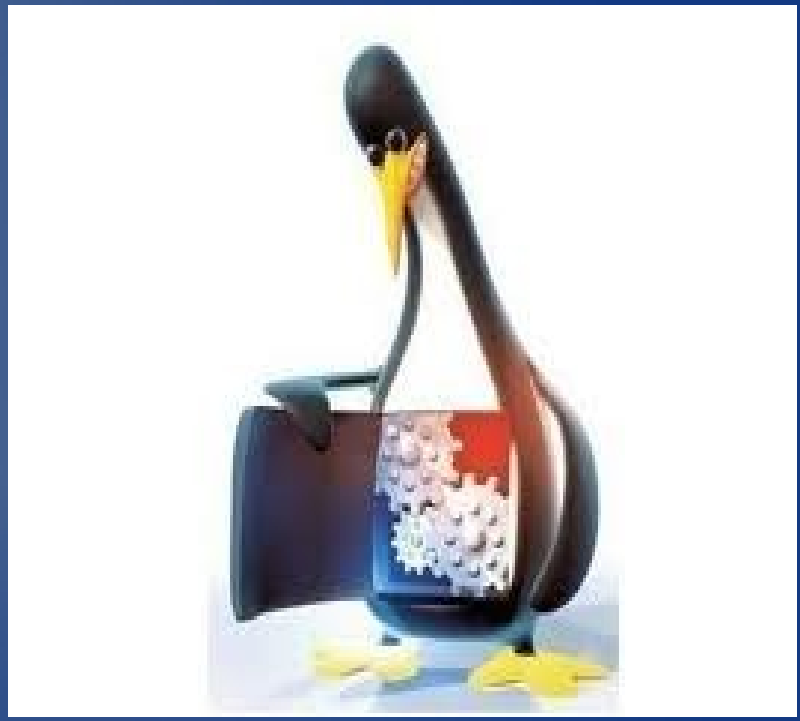
Hoy en el mercado existen diferentes administradores de recursos, tanto privativos como libres.





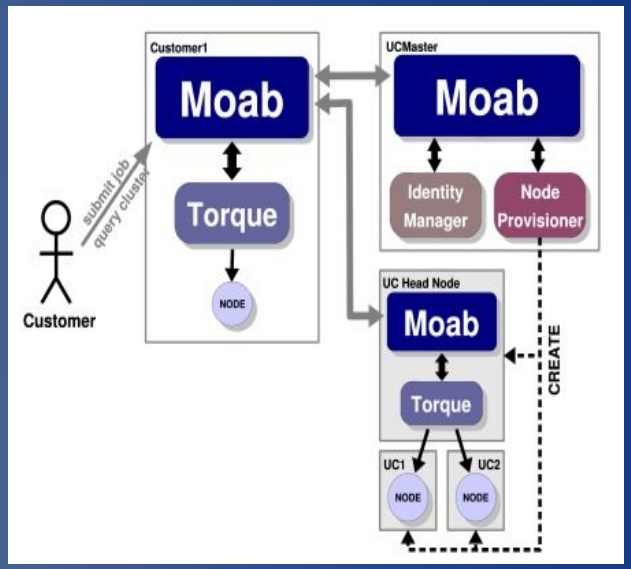
# PLANIFICADOR

Un planificador de jobs (o trabajos) se instala sobre el administrador de recursos, sustituyéndolo en dichas tareas mientras que el administrador de recursos sólo registra los recursos y distribuye los trabajos a los nodo



# PLANIFICADOR

Hoy en dia hay planificadores que se pueden integrar con diferentes administradores de recursos tanto libres como privativos.




MOAB CLUSTER SCHEDULER

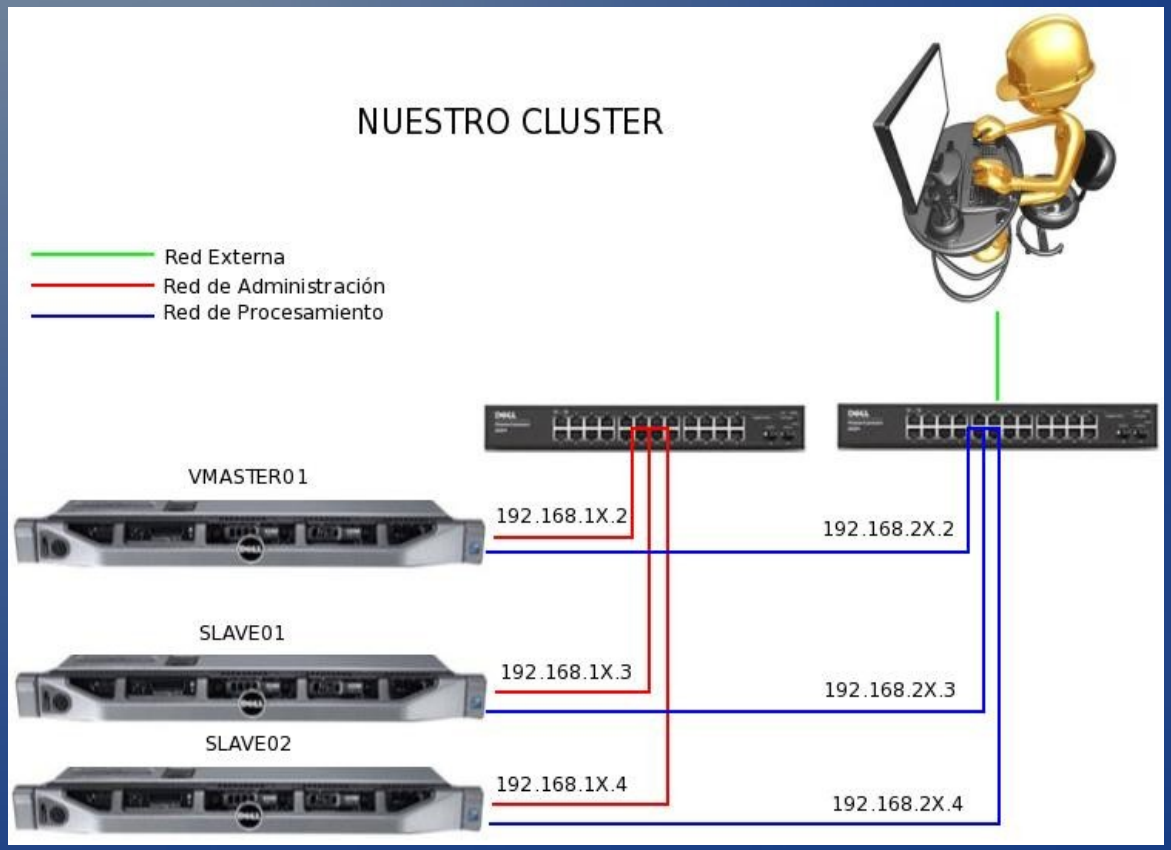
CLUSTER RESOURCES INC.

# Y AHORA??

# VAMOS A INSTALAR UN CLUSTER!!!!

# EQUIPO

Para este taller les daremos 3 maquinas instaladas con Centos 6.3, la topologia de red  de las maquinas sera la siguiente:



# COMUNICACIONES

Veremos los archivos que necesitamos para las comunicaciones.

Primero que los nodos se vean entre si mediante su hostname y no mediante una ip ya que seria incomodo hacerlo en un ambiente con mas de 50 nodos

- cat /etc/hosts

Nos permite configurar el sistema para poder ver nombres de maquinas y no solamente ip's.

# COMUNICACIONES

Los paquetes que usaremos para comunicarnos seran ssh y rsh.

Para ver si tenemos los paquetes en nuestra maquina ejecutamos lo siguiente:

- rpm -qa | grep ssh
- rpm -qa | grep rsh

Como podran observar nuestras maquinas ya los traen instalados.

# COMUNICACIONES

Los paquetes que usaremos para comunicarnos seran ssh y rsh.

Para ver si tenemos los paquetes en nuestra maquina ejecutamos lo siguiente:

- rpm -qa | grep ssh
- rpm -qa | grep rsh

Como podran observar nuestras maquinas ya los traen instalados.

# COMUNICACIONES

SSH (Secure SHell)

Es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. A diferencia de otros protocolos de comunicación remota tales como FTP o Telnet, SSH encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas.

# COMUNICACIONES

RSH

rlogin y rsh fueron diseñados sólo para la comunicación entre sistemas UNIX Berkeleys.

La principal desventaja de rsh es que no ofrece una conexión segura, pero ya que nuestro cluster vive en una red LAN aislada, no tendremos problemas y es muy util por su facil configuración, ademas de ser muy ligero en cuando al envio de paquetes.

# COMUNICACIONES

Ya que las llaves para que las maquinas se comuniquen entre si ya fueron creadas solo revisaremos los archivos necesarios para la configuracion.

Para generar una llave publica y poder conectarnos a los nodos de procesamiento hacemos lo siguiente:

- ssh-keygen -t rsa

Luego creamos el archivo para las maquinas autorizadas:

- touch /root/.ssh/authorized_keys

Y despues copiamos el archivo de la llave publica de la siguiente manera:

- cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys

# COMUNICACIONES

Ya que las llaves para que las maquinas se comuniquen entre si ya fueron creadas solo revisaremos los archivos necesarios para la configuracion.

Copiamos el directorio /etc/ssh y /root/.ssh a los nodos de procesamiento:

- scp -r /etc/ssh slave01:/etc/ssh
- scp -r /etc/ssh slave02:/etc/ssh
- scp -r /root/.ssh slave02:/root/.ssh
- scp -r /root/.ssh slave01:/root/.ssh

Con esto habremos creados las llaves para que al conectarnos a los nodos no necesitemos el password cada vez que ingresemos.

# COMUNICACIONES

Ahora procederemos a habilitar rsh.

Editamos el archivo .rhosts del siguiente modo:

- vim /root/.rhosts

y agregamos los nombres de los nodos:

- slave01
- slave02

Editamos el archivo /etc/securetty

- vim /etc/securetty

y agregamos al final lo siguiente:

- Rlogin
- Rsh
- Rexec
- Rsync

# COMUNICACIONES

Ejecutamos los siguientes comandos:

- chkconfig –level 35 rsh on
- chkconfig –level 35 rsync on
- chkconfig –level 35 rexec on
- chkconfig –level 35 rlogin on

# ALMACENAMIENTO

Existen diversos sistemas de almacenamiento tales como Lustre, NFS, Hadup, GFS, etc, todos ellos se pueden integrar con un cluster y la eleccion del sistema de almacenamiento depende de las necesidades del usuario, nosotros utilizaremos NFS por ser una opcion bastante comoda para lo que utilizaremos.

# ALMACENAMIENTO

NFS

NFS, acrónimo de Network File System, es un popular protocolo utilizado para compartir volúmenes entre máquinas dentro de una red de manera transparente, más comúnmente utilizado entre sistemas basados sobre UNIX.

# ALMACENAMIENTO

Lo que queremos en nuestro cluster es que todas las maquinas vean el home de los usuarios y no sea necesario escribir el mismo archivo 3 veces en todos los sistemas, para esto en el nodo maestro configuraremos:

- vim /etc/exports

y agregaremos las ip's de los nodos de procesamiento:

- /home/ 192.168.1x.3
- /home/ 192.168.1x.4

Iniciamos el servicio:

- /etc/init.d/nfs start

# ALMACENAMIENTO

Ejecutamos:

- exportfs -rva

Para verificar que estamos exportando el home, entramos a los nodos slave01 y slave02 y ejecutamos:

- showmount -e 192.168.1x.2

Y nos mostrara el directorio que el home esta exportando.

# PROCESAMIENTO EN PARALELO

Para el procesamiento en paralelo existen varias herramientas, pero la que nosotros utilizaremos sera OpenMPI.

Lo primero sera extraer el paquete:

- tar -xvf openmpi -C /opt

# PROCESAMIENTO EN PARALELO

Despues vamos a configurarlo para nuestra arquitectura:

- ./configure –prefix=/usr/local/openmpi

Contruimos la aplicación:

- make

Y la Instalamos:

- make install

# PROCESAMIENTO EN PARALELO

Cuando querramos ejecutar deberemos crear un archivos donde le especificamos los nodos que va a usar:

Creamos el archivo de los nodos:

- vim nodos

Agregamos:

- slave01
- slave02

Hecho esto ya podremos enviar un job.

# PROCESAMIENTO EN PARALELO

Propagamos el directorio /user/local/openmpi a nuestros nodos de procesamiento:

- scp -r /usr/local/openmpi slave01:/usr/local/openmpi
- scp -r /usr/local/openmpi slave02:/usr/local/openmpi

# PROCESAMIENTO EN PARALELO

Creamos el archivo ompi.sh en /etc/profile.d/para que cargue las variables de ambiente cuando inicie el sistema:

- echo “export PATH=$PATH:/usr/local/openmpi/bin” > /etc/profile.d/ompi.sh
- echo “export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/openmpi/lib >> /etc/profile.d/ompi.sh

Cargamos las variables de OpenMPI el nodo maestro y en los nodos de procesamiento:

- source /etc/profile.d/ompi.sh

# PROCESAMIENTO EN PARALELO

Propagamos el archivo y cargamos las variables en los nodos:

- scp /etc/profile.d/ompi.sh slave01:/etc/profile.d/.
- scp /etc/profile.d/ompi.sh slave02:/etc/profile.d/.

y cargamos las variables de ambiente en los nodos de procesamiento:

- ssh slave01 source /etc/profile.d/ompi.sh
- ssh slave02 source /etc/profile.d/ompi.sh

# PROCESAMIENTO EN PARALELO

Ejecutamos el archivo de ejemplo matrix-parallel-test.c

- mpicc matrix-parallel-test.c -o matrix
- mpirun -hostfile nodos -np 2 matrix

Ya que podemos enviar trabajos en las maquinas, ahora crearemos nuestro administrador de recursos, donde hostfile es el archivo que contiene a los nodos de procesamiento.

# ADMINISTRADOR DE RECURSOS

Torque

Torque es un poderoso administrador de recursos. El cluster consistira en un nodo cabecera y muchos nodos de computo. El usuario envia jobs o trabajos al gesto de recursos y estos son puestos en cola hasta que el sistema este listo para ejecutarlos.

# INSTALACIÓN DE TORQUE

Instalaremos torque en nuestro nodo maestro vmaster01, cada maquina en el home de root tiene el archivo torque-3.0.6.tar.gz, el cual descomprimimos en /opt usando el siguiente comando:

[root@vmaster01 ~]# tar -xvf torque-3.0.6.tar.gz -C /opt

Una vez hecho esto ingresamos al directorio:

cd /opt/torque-3.0.6

# INSTALACIÓN DE TORQUE

Ejecutemos el siguiente comando para ver los archivos que componen a torque.

- [root@vmaster01 torque-3.0.6]#  ls

Nos mostrara los archivos que acabamos de extraer de torque.

Para instalar torque debemos primero construirlo y podemos hacerlo con diferentes opciones:

- ./configure –help | less

Veremos las multiples opciones que podemos darle a nuestro torque, sin embargo no todas estan disponibles para nuestra arquitectura por lo que para configurarlo ejecutaremos:

- ./configure --prefix=/usr/local/torque

# INSTALACIÓN DE TORQUE

Una vez creada la configuración de nuestro torque procederemos a contruirlo mediante el siguiente comando:

- [root@vmaster01 torque-3.0.6]# make

Ya construido lo instalaremos en nuestro nodo maestro:

- [root@vmaster01 torque-3.0.6]# make install

Y ya habremos construido torque para nuestro nodo maestro.

# INSTALACIÓN DE TORQUE

Configuraremos el sistema para que pbs_server y pbs_sched sean servicios, desde nuestro directorio en /opt/torque-3.0.6 ejecutamos lo siguiente:

- cp contrib/init.d/pbs_sched /etc/init.d/.

Y tambien:

- cp contrib/init.d/pbs_server /etc/init.d/.

Despues lo agregamos al chkconfig mediante:

- chkconfig –add pbs_server
- chkconfig –add pbs_sched

# INSTALACIÓN DE TORQUE

Ya que al construir torque, los archivos se instalaron en el directorio /usr/local/torque tendremos que editar los archivos de servicio de pbs_server y pbs_sched, ejecutaremos:

- vim /etc/init.d/pbs_server

y editamos la siguiente linea:

- PBS_DAEMON=/usr/local/sbin/pbs_server

Por:

- PBS_DAEMON=/usr/local/torque/sbin/pbs_server

# INSTALACIÓN DE TORQUE

Haremos los mismo para el pbs_sched, que es el planificador que trae torque:

- vim /etc/init.d/pbs_sched

y editamos las siguientes lineas:

- PBS_DAEMON=/usr/local/sbin/pbs_sched

Por:

- PBS_DAEMON=/usr/local/torque/sbin/pbs_sched

# INSTALACIÓN DE TORQUE

Una vez editados los archivos de servicio, agregaremos los binarios de pbs al PATH:

export PATH=$PATH:/usr/local/torque/sbin:/usr/local/torque/bin

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/torque/lib

Ahora crearemos nuestra base de datos asi que entramos a /opt/torque-3.0.6 y ejecutamos.

- [root@vmaster01 torque-3.0.6]# ./torque.setup root

Hasta este punto no deberiamos tener mayores problemas.

# INSTALACIÓN DE TORQUE

Ya que esta creado el servidor, editamos el siguiente archivo para ingresar los nodos de procesamiento:

- vim /var/spool/torque/server_priv/nodes

Ingresamos los nodos de procesamiento con el siguiente formato:

nodo_esclavo numero_de_procesadores=n queue (slave01 np=1 DEFAULT)

En nuestro cluster, bastara con:

- slave01
- slave02

Ya que nuestros nodos tienen solo 1 procesador y pertenecen a la cola por default listo nuestro nodo maestro ya tiene configurado Torque.

# INSTALACIÓN DE TORQUE

Ahora vamos a crear el archivo pbs.sh en el /etc/profile.d/:

- touch /etc/profile.d/pbs.sh

Editamos con:

- vim /etc/profile.d/pbs.sh

Y agregamos las siguientes lineas:

- export PATH=$PATH:/usr/local/torque/sbin:/usr/local/torque/bin
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/torque/lib

Guardamos y listo.

# INSTALACIÓN DE TORQUE

PERO....

Y LOS NODOS DE PROCESAMIENTO?

# INSTALACION DE TORQUE

Primero crearemos los paquetes para nuestros nodos de computo, entramos al nodo maestro en /opt/torque-3.0.6 y ejecutamos lo siguiente:

- [root@vmaster01 torque-3.0.6]# make packages

Copiamos el paquete MOM en los nodos de computo mediante la siguiente linea:

- [root@vmaster01 torque-3.0.6]# scp torque-package-mom-linux-x86_64.sh slave01:.
- [root@vmaster01 torque-3.0.6]# scp torque-package-mom-linux-x86_64.sh slave02:.

# INSTALACIÓN DE TORQUE

Una vez hecho esto, nos conectaremos al primer nodo de procesamiento mediante:

- [root@vmaster01 ~]# ssh slave01

Le daremos permisos al paquete y lo ejecutamos con los siguientes comandos.

- chmod +x  torque-package-mom-linux-x86_64.sh
- ./torque-package-mom-linux-x86_64.sh

# INSTALACIÓN DEL TORQUE

Una vez hecho esto, nos conectaremos al segundo nodo de procesamiento mediante:

- [root@vmaster01 ~]# ssh slave02

Le daremos permisos al paquete y lo ejecutamos con los siguientes comandos.

- chmod +x  torque-package-mom-linux-x86_64.sh
- ./torque-package-mom-linux-x86_64.sh

# INSTALACIÓN DE TORQUE

Ya que hemos configurado el nodo de procesamiento, vamos a habilitar el pbs_mom como servicio.

Primero regresamos al nodo maestro:

- ssh vmaster01

entramos /opt/torque-3.0.6

- cd /opt/torque-3.0.6

ejecutamos las siguientes lineas:

- scp contrib/init.d/pbs_mom slave01:/etc/init.d/.
- scp contrib/init.d/pbs_mom slave02:/etc/init.d/.

# INSTALACIÓN DE TORQUE

Ahora al igual que hicimos en el maestro debemos editar los archivos init y colocar la ruta que contiene los binarios

- PBS_DAEMON=/usr/local/torque/sbin/pbs_mom
- PBS_HOME=/var/spool/torque

Y exportamos las rutas al PATH:

- export PATH=$PATH:/usr/local/torque/sbin
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/torque/lib

# INSTALACIÓN DE TORQUE

Ahora habilitaremos el pbs_mom como servicio.

Regresamos al nodo slave01:

- ssh slave01 y ejecutamos:
- cd /etc/init.d/
- chkconfig --add pbs_mom

Encendemos el servicio en los niveles 3 y 5:

- chkconfig –level 35 pbs_mom on

# INSTALACIÓN DE TORQUE

Ahora vamos a crear el archivo pbs_cliente.sh en ambos nodos y en la ruta /etc/profile.d/, ejecutamos lo siguiente para ambos nodos de procesamiento:

- touch /etc/profile.d/pbs_cliente.sh

Editamos con:

- vim /etc/profile.d/pbs_clliente.sh

Y agregamos las siguientes lineas:

- export PATH=$PATH:/usr/local/torque/sbin
- export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/torque/lib

Guardamos y listo.

# INSTALACIÓN DE TORQUE

Ahora que ya casi hemos terminado, vamos a encender los servicios en nuestro nodo maestro, ejecutamos:

- /etc/init.t/pbs_sched start
- /etc/init.d/pbs_server start

Revisamos que nuestro nodo maestro vea a los  nodos de procesamiento mediante:

- pbsnodes -a

# PRUEBAS

Ahora enviaremos un job a nuestro cluster, en el home del usuario taller tenemos el archivo job.pbs, lo eviamos del siguiente modo:

- qsub job.pbs

en los todos podemos ejecutar top para ver que en realidad se esta ejecutando.

Para verificar que el administrador de recursos envio el trabajo, ejecutamos:

- qstat -a

# PREGUNTAS??

# INSTRUCTORES

Jorge Elizalde Perez

Germán Edwin Reynoso Aliaga


CNS
www.isum.mx
A NavigationThrough Science and Technology
DEL 5 AL 8 MARZO, MANZANILLO, COLIMA. MÉXICO.
ISUM 2013
4th INTERNATIONAL SUPERCOMPUTING CONFERENCE IN MEXICO
IBM
intel
DELL
CNS
xyratex
CRAY
THE SUPERCOMPUTER COMPANY
hp
BROCADE
VÁNUME
Westcon
NWT
NetApp
emtec
GROUP
Qualisys
CADGRAFICS
enter
SONDA
nexthraOne
VCE
AMD