KestrelHPC

Jon Ander Hernández

EHUko Software Librea sustapenerako taldea

7 de Abril de 2011

Introducción

¿Qué es KestrelHPC?

KestrelHPC es un conjunto de herramientas que facilitan la gestión de un cluster:

- Instalacción y configuración de los servicios básicos en el frontend.
- Configuración del sistema operativo de los nodos.
- Apagado y arranque de los nodos.
- Instalación de software.

¿Qué es un cluster?

"A computer cluster is a *group* of *linked* computers, *working* together closely thus in many respects forming a single computer."

"The components of a cluster are commonly, but *not always*, connected to each other through fast local area networks."

"Clusters are usually deployed to *improve performance and availability* over that of a single computer, while typically being much more *cost-effective* than single computers of comparable speed or availability."


NASA

Clasificación de clusters

La clasificación se realiza según la función que desempeña :

High Availability (HA)

"They operate by having redundant computers or nodes which are then used to provide service when system components fail."

Load balancing

"is a computer networking methodology to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload."

High-performance computing (HPC)

"uses supercomputers and computer clusters to solve advanced computation problems"

HPC y MPI

- is an *API specification* that allows processes to communicate with one another by **sending and receiving messages**.
- is de **facto standard** for communication among processes that model a parallel program running on a **distributed memory system**.
- MPI's goals are high performance, scalability, and portability. MPI remains the dominant model used in high-performance computing today.
- *Provide* :
  - essential virtual topology
  - synchronization
  - communication functionality between a set of processes
  - in a language-independent way

¿Qué es un cluster Beowulf?

Es un cluster montado por *hierros* :

¿Cómo funciona un cluster Linux?

¿Cómo funciona un cluster Linux live? / Ingredientes

- Compartir el sistema de archivos raíz mediante NFS 3.

  El 4 es más complejo y mucho más eficiente aunque nosotros sólo lo soportamos en Debian
  Requiere configurar el fichero */etc/exports* y el fichero */etc/fstab*[1].

- Compartir el directorio home :
  - Necesitamos compartir las propias aplicaciones entre los nodos, datos, resultados, etc...
  - Normalmente instalamos las bibliotecas que usan nuestras aplicaciones también en el home.
  - Cada usuario tiene sus cosas separadas.

  Es interesante usar un sistema de archivos distribuido frente a uno centralizado :

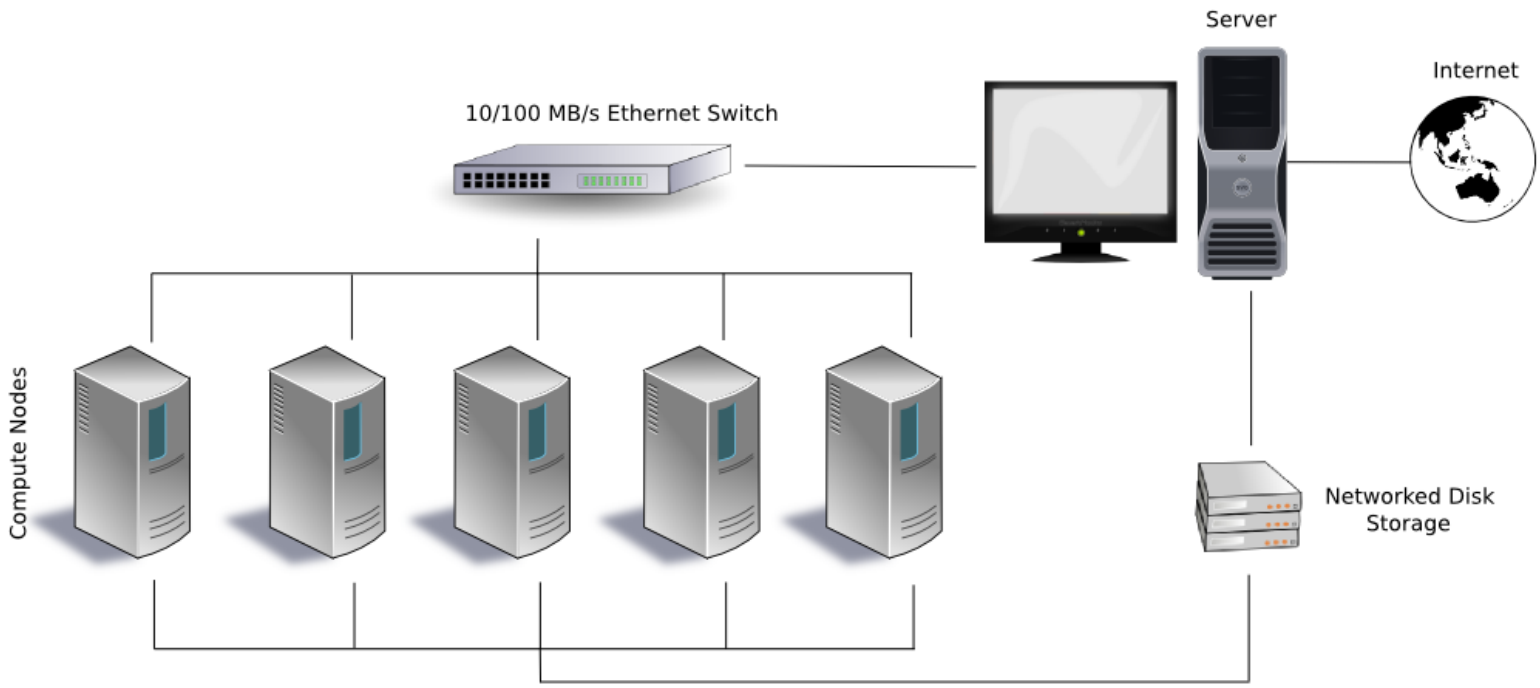
  - Red Hat Global File System (GFS)
  - Oracle Cluster File System (OCFS)
  - ...

[1] NFS4 requiere hacer un mount bind al directorio */etc/exports*

- SSH. Una aplicación MPI se invoca usando SSH.

1. Necesitamos crear certificados digitales para realizar las conexiones (el acceso por contraseña no es una opción con varias maquinas... ;-).

2. Creamos claves para cada usuario y las añadimos como claves autorizadas para el propio usuario.



- Arranque por red : Preboot Execution Environment
  - Elementos necesarios :
    1. Servidor de DHCP → dhcpd vs dnsmasq
    2. Servidor de TFTP → tftpd-hpa vs dnsmasq
    3. Bootloader PXE → grub, grub2 e pxelinux (forma parte de syslinux)
  - Funcionamiento :
    1. La BIOS durante tras el POST delega el arranque en la tarjeta de red usando una PXE ROM.
    2. PXE firmware broadcasts a DHCPDISCOVER packet extended with PXE-specific options.
    3. El servidor de dhcp responde indicando un path y una ip de donde descargar el NBP usando TFTP[2].
    4. El bootloader emplea las APIs del firmware PXE para localizar el kernel y el initramfs a través de la red (Pre-boot, UDP, TFTP, Universal Network Device Interface (UNDI)).
  - Posibilidades :
    - PXE permite configurar el arranque según la mac de la máquina (el kernel, el initramfs y los parámetros de arranque).

[2] Una especie de FTP que usa udp

¿Cómo funciona el modo live?

- La raíz del sistema en los nodos es de sólo lectura. Si fuera un sólo nodo se podría plantear de escritura, pero no si está compartido. Un sistema funcional necesita poder escribir /var/log, /var/run, /var/lib, ...
- Solución : UnionFS. Permite unificar capas de sistemas de archivos[3].
- Unificamos un tmpfs, un sistema de archivos *RAM*, en modo RW encima del sistema de archivos NFS en modo *lectura*.
- Toda esta magia es necesaria que este preparada antes de arrancar el primer programa, el init, quien iniciará el arranque del sistema. El núcleo tan sólo se limita a abstraer el hardware y a gestionar los recursos, y por tanto es necesario un primer sistema para poder arrancar, y este es el llamada initramfs. Un sistema de archivos muy pequeño que se carga en memoria con las mínimas herramientas, y hace el mínimo número de pasos para montar el / y arrancar el init real.

Demo : casper vs dracut-aufs

[3]Reglas : Los ficheros presentes en los sistemas de archivo de arriba reemplazan a los inferiores, las modificaciones se registran en el sistema de archivos de escritura superior, los fichero no se eliminan tan sólo se marcan como eliminados mediante un fichero oculto (un "whiteout" .wh.<nombre fichero>)

KestrelHPC 1.0

Objetivos :

- Nodos live. Imprescindible no instalar nada en los nodos.
- Facilitar la ejecución de aplicaciones MPI manteniendo una lista de los nodos.
- Que sea instalable en una maquina.

Incentivos :

- Basado en una distribución Debian.

Lo que lo llevó a convertirse en un fork de PelicanHPC. Un live cd para montar clusters live.

PelicanHPC

Features

- /pelican_config file to allow for persistence, customization and headless boot.
- autodetection of persistent frontend home
- autodetection of frontend and node local scratch space
- ability to run local scripts post boot and setup
- node beep after boot
- firewall
- automated node booting using wake-on-lan
- configuration of slots and optional frontend inclusion for mpi
- ganglia
- static IP assignment configurable using MAC addresses.
- node startup/shutdown script
- possibility to serve DHCP to machines that are not compute nodes

KestrelHPC 2.0

Objetivos :

- Un cluster es algo sencillo. Podemos reunir todas las recetas necesarias en batería y facilitar así su configuración.
- Modular. Un cluster no tiene sólo porque ser HPC con MPI
  - ¿Y jugar al quake en varios ordenadores?
  - ¿Añadir planificadores como Slurm o Torque?
  - ¿Soporte para clusters SSI[4] como Kerrighed o OpenMosix?
  - ¿Un cluster de nodos VoIP?
- Soporte para grupos de ordenadores.
- Múltiples imágenes para los nodos.
  Cada departamento puede crear su propio sistema instalando el software que necesitan directamente desde Debian
- Facilitar el registro de nodos. Arrancamos los ordenadores y los registramos automáticamente.

[4] Single System Image, el cluster se comporta a nivel de aplicación como un único ordenador

- Soporte WakeOnLan[5].

- Cambiar la configuración debe de ser sencillo : Cambiamos el fichero de configuración y aplicamos al sistema los cambios mediante kestrel-reconfigure. Este regenerará ficheros de configuración de servicios, eliminará o creará entradas en /etc/fstab, montará y desmontará directorios del nfs, reiniciará servicios, etc...

- Extensible. No sólo a nivel de módulos, ¿Y si queremos añadir nuestros propios scripts o reemplazar uno existente? Basta con añadir los scripts en */etc/kestrel/{node|frontend}/{configure.d|install.d}*... Y si el script tiene el mismo nombre que el de Kestrel será ejecutado en su lugar.

[5] Las tarjetas de red convencionales pierden el modo WakeOnLan cada vez que arrancan y el sistema operativo debe de volver a restaurarlo. Si usamos otro sistema operativo en los nodos, deberemos configurarlo para que active el modo WOL

¿Cómo usamos KestrelHPC?

`http://kestrelhpc.sourceforge.net/documentation.html`

Instalación :

- Instalamos los paquetes en una Ubuntu o Debian.
- Configuramos el sistema (como mínimo la IP)[6].
- Creamos una imagen para los nodos.
- Añadimos los usuarios al cluster.
- Registramos los nodos de nuestro cluster con tan sólo con arrancarlos en modo red.

Comandos :

- kestrel-users
- kestrel-images
- kestrel-nodes
- kestrel-apt
- kestrel-reconfigure

[6] Idealmente la primera vez podemos consultar la configuración usando dialog/whiptail/zenity

Diseño general

Simple (filosofía KISS):

- Los nodos conectados se guardan como entradas en el fichero */etc/hosts*.
  Cuando se produce un evento de encendido o apagado se añaden o se quitan las entradas.
- La información de los nodos se guarda como las entradas del servidor DHCP
  kestrel-nodes –register asocia a la mac un hostname, pero podemos editarlo manualmente y asociar al nodo también una ip
- La configuración tan sólo son *scripts* que se ejecutan en fases.
- Todo script se puede reemplazar creando una versión con el mismo nombre en /etc/kestrel.
- Facilitamos el registro de nodos usando el fichero *leases* del dhcp para obtener datos como la mac.

Imágenes de los nodos

1. Creamos una **imagen mínima** con *debootstrap* (sin entorno gráfico, ni cualquier servicio no básico)

2. Fase *install* : Configuramos la imagen mínima e instalamos algunas cosas.
   Ejemplos : desabilitamos el dhcp, añadimos el beep de arranque, añadimos la rpc de los nodos, etc...

3. Fase *packages* : Instalamos los paquetes de cada módulo.

4. Fase *post-install* : Configuramos los paquetes recién instalados.

5. Fase *configuración* : Configuramos la imagen del nodo.
   Ejemplos : La melodía del beep, Si cambiamos la IP del servidor tenemos que cambiar la configuración de la RPC, o si cambiamos la ubicación de los exports tenemos que reconfigurar el /etc/fstab

La diferencia entre las fases de instalación y las fases de configuración, es que las de instalación sólo se realizan una vez cuando se crea la imagen.

Seguridad

- Trabajamos con imágenes y el home compartidos por nfs, lo cual implica que nada de lo exportado debe comprometer el frontend.
- Accedemos a los nodos sin clave, hay que generar estas claves y evitar que su uso permita el acceso al frontend.
- El nfs es abierto por defecto[7], y controlamos el acceso cerrando el NFS mediante un firewall y abriéndolo para un nodo con una determinada mac.
- Hay que comprobar que el nodo que ha arrancado es realmente un nodo, y no otro equipo sospechoso o el mismo equipo arrancado con otros sistema operativo.

  Solución si se puede abrir una sesión ssh a ese ordenador entonces es un nodo.
- Hay que notificar eventos al frontend de manera segura.
  - Polling con pings (PelicanHPC hace un polling bajo petición del usuario).
  - Los nodos ejecutan un comando en el frontend.
  - Usar una RPC simple.

Es realmente difícil garantizar la seguridad en un escenario así. La mejor solución es aislar la red del cluster.

[7] Usar el sistema de control de acceso de NFS implicaría editar el fichero exports y reiniciar el servicio nfs cada vez que añadimos o quitamos un nodo

kestrel-reconfigure –all –force

- **frontend**
  Instala las herramientas y configura los servicios
- **Reinicia el demonio**
- **nodo**
  Instala las herramientas, instala paquetes de cada módulo, y configura el nodo.

```
$ kestrel-reconfigure --all --force
Executing install scripts

    Executing : dnsmasq_config
    Executing : nfs4_root-export

Reconfiguring the frontend

    Executing script : dnsmasq_config
    Executing script : dnsmasq_tftpboot
    Executing script : ganglia_monitor
    Executing script : ganglia_webfrontend
    Executing script : nfs4_root-export
    Executing script : openmpi_set-default-hostfile
    Executing script : sshd_users-keygen
    Executing script : ufw_kestrel-rpc

Restarting KestrelHPC Daemon
```

```
Reconfiguring the image "image1"

    Executing install scripts

        Executing : host_nfs_export-image
        Executing : host_pxe_add-to-imagelist
        Executing : image_core_add-kestrel-users
        Executing : image_core_beep-on-startup_chroot
        Executing : image_core_dhclient-hook-set-hostname_chroot
        Executing : image_core_dhclient-wol-on-startup_chroot
        Executing : image_core_disable-dhclient-script_chroot
        Executing : image_core_disable-dpkg-upstart_chroot
        Executing : image_core_num-of-cpus_chroot
        Executing : image_core_set-hostname_chroot

    Installing extra software

        Installing kestrel extra packages : basic_packages
        Installing kestrel extra packages : ganglia
        Installing kestrel extra packages : openmpi

    Executing post-install scripts

        Executing : image_core_delete-sshd-motd_chroot
        Executing : image_nfs_configure_chroot
        Executing : image_openmpi_fix-oldlinks_chroot
```

```
Executing configure scripts

    Executing : host_nfs_export-image
    Executing : host_pxe_image-menu
    Executing : image_core_beep-on-startup_chroot
    Executing : image_core_kestrel-rpc_chroot
    Executing : image_ganglia_monitor_chroot
    Executing : image_nfs_home-fstab_chroot
    Executing : image_sshd_authkeys_chroot
    Executing : image_user-home
```

Demo Time

¿Preguntas?

¡Gracias a todos por venir! ;-)

Referencias

- I ❤ http://en.wikipedia.org 0:-)

- "KestrelHPC"

  http://kestrelhpc.sourceforge.net/