Instalación, configuración y puesta en producción de un clúster Rocks

Eduardo Rodriguez Gutiez
Verónica Barroso García

19 de septiembre de 2014

Índice

1. Introducción

Figura 1: Imagen general del cluster en el laboratorio 1L022

El propósito del presente trabajo consiste en montar un clúster para la asignatura "Infraestructuras para el desarrollo de aplicaciones de computación de altas prestaciones" del Máster en Ingeniería Informática, con el objetivo de ejecutar aplicaciones que sean intensivas respecto al cálculo y que admitan paralelización; en este caso, esta aplicación será el simulador de dinámica atómica y molecular LAMMPS.

Para ello se dispone de los siguientes equipos, todos ubicados, a fecha de redacción, en el laboratorio 1L022 del Edificio de Tecnologías de la Información y las Telecomunicaciones:

16x Intel Pentim 4 640 (3.2 GHz), 2-4GB RAM, 1TB HDD

5x Intel Pentim D 840 (3.2 GHz, dual core), 4GB RAM, 1TB HDD, NVidia GTX650Ti-2GBDDR5

2x Intel Xeon X3210 (2.13 GHz, quad core), 4GB RAM, 160 GB HDD

1x Intel Xeon X3220 (2.4 GHz, quad core), 4GB RAM, 250 GB HDD

En esta lista se pueden crear tres tipos de ordenadores de acuerdo a sus características, agrupando los dos últimos; para empezar, los primeros, que tienen como procesador un Pentium 4, serán utilizados tanto como nodos de cálculo (los que residen en la balda superior, en la mesa), como frontend (uno de los que está en la parte inferior de la mesa), y aquellos empleados

como nodos de cálculo se denominarán *normales*. A continuación, los equipos con un Pentium D se denominarán *con gráfica* o *gráficos*, ya que tienen una tarjeta nVidia GTX650 Ti, lo cual permite que, además de ser doble núcleo, mediante la utilización de CUDA para realizar cálculos paralelos utilizando la gráfica, dispongan de una mayor potencia de cálculo, y por último, los tres ordenadores tipo Xeon, dado que están en el armario enracable y tienen unas características de procesador muy similares (8 MB de caché, 1066 MHz FSB, quad core, etc., siendo la única diferencia entre sus modelos, X3210 y X3220, la velocidad máxima de reloj; 2.13 y 2.4 GHz, respectivamente) y la misma cantidad de RAM, se denominarán *enracables*.

Como puede apreciarse, esta clasificación de los nodos de cómputo está hecha en base a las características del procesador, la capacidad de cómputo del ordenador, y el factor de forma (enracable o semitorre ATX).

2. Instalación

El proceso de instalación del sistema operativo en los equipos o *hosts* tiene dos partes bien diferenciadas: la instalación del front-end, que será el equipo desde el que se administre el clúster (en cuyo ámbito entran tareas como lanzamiento de trabajos, instalación de librerías y programas, obtención de resultados, gestión de usuarios, gestión de colas, etc.) y se acceda al resto de los host que conforman éste; y los nodos de cómputo, que se encargan de realizar los cálculos necesarios, bien únicamente a través de los núcleos de sus CPUs, bien utilizando aceleradores y GPUs.

En esta sección veremos estos dos pasos de instalación, empezando por el frontend, ya que será éste el que posteriormente detecte e instale los nodos a través de red, usando PXE y mandando las imágenes de disco de instalación necesarias.

2.1. Instalación del frontend

El ordenador elegido como frontend, de los dos que existen en la parte inferior de la mesa, es el que está más cerca del armario enracable.

En primer lugar, ha de conocerse si los procesadores que forman parte del clúster (nodos de cómputo y front-end) soportan instrucciones de 64 bits o únicamente de 32. En este caso, dado que se conocen los modelos de los microprocesadores, puede accederse a la página del fabricante para conocer este detalle[1] Para los equipos que formarán parte del clúster, estos datos ya han sido buscados y las respectivas páginas con las especificaciones están enlazadas en la lista de equipos del laboratorio de la Sección 1, y todos los procesadores soportan el conjunto de instrucciones de 64 bits.

[1]En nuestro caso, dado que los procesadores son de Intel, puede utilizarse el buscador de productos Intel ARK en `http://ark.intel.com/`.

En caso de desconocerse el modelo de procesador, puede grabarse un live DVD de alguna distrubución de Linux, por ejemplo Ubuntu, e iniciarle en modo live, ya que el propósito no será instalar Ubuntu sino conocer los detalles del procesador. Una vez iniciado el escritorio, basta con lanzar un terminal y utilizar el comando sudo lshw para conocer los detalles del hardware:

```
ubuntu@Ubuntu:~$ sudo lshw
    description: Computer
    width: 64 bits
    capabilities: smbios-2.3 dmi-2.3 vsyscall64 vsyscall32
    configuration: boot=normal uuid=59CF9F6A-DD82-11D9-97D9-00E018EB099A
  *-core
       description: Motherboard
       product: D915GUX
       vendor: Intel Corporation
       physical id: 0
       version: NNNNNNNN-NNN
       serial: LLLLNNNNNNNN
[...]
     *-cpu
          description: CPU
          product: Intel(R) Pentium(R) 4 CPU 3.20GHz
          vendor: Intel Corp.
          physical id: 4
          bus info: cpu@0
          version: Intel(R) Pentium(R) 4 processor
          serial: To Be Filled By O.E.M.
          slot: J2E1
          size: 3200MHz
          capacity: 3200MHz
          width: 64 bits
          clock: 200MHz
          capabilities: fpu fpu_exception wp vme de pse
 tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm syscall nx
 x86-64 constant_tsc pni monitor ds_cpl est cid cx16 xtpr
 cpufreq
[...]
```

Listing 1: Salida del comando list hardware (lshw) para el frontend

Otro comando que permite conocer esta información es sudo lscpu, o utilizar el software Intel Processor Diagnostic Tool, que tiene una versión ejecutable desde un live USB con Fedora[2]. No obstante, es bueno disponer, si aún no se ha montado el procesador en la placa base o hace falta cambiar la pasta térmica, de la referencia que aparece marcada a láser sobre la cubierta metálica del procesador. De esta forma, se puede saber si el chip es la variante *Extreme Edition* de alguna de las arquitecturas, detalle que será útil a la hora de configurar la biblioteca ATLAS.

[2]Ver `http://www.tcsscreening.com/files/users/IPDT_LiveUSB/index.html`.

Una vez conozcamos si el procesador soporta instrucciones de 64 bits o únicamente 32, accederemos a la página de descargas del proyecto Rocks Clusters[3] para descargar los discos de instalación. Dado que el ordenador seleccionado como frontend dispone de una unidad de DVD y que el procesador es x86_64, se ha optado por bajar y grabar la versión Jumbo (DVD) x86_64 (que resulta más cómoda que descargarse el conjunto de 4 CDs) de la versión 5.5 de Rocks. Esta versión ha sido elegida por ser la antepenúltima versión a fecha de inicio de las prácticas, lo que en el mundo de la informática suele contribuir a la existencia de mayor documentación y un menor número de errores.

Una vez iniciado el ordenador que actuará como frontend e insertado el DVD de Rocks en la unidad lectora, conviene acceder a la BIOS (generalmente pulsando la tecla F2 del teclado) para verificar que en el orden de arranque se sitúe antes la unidad lectora que el disco duro.

Al iniciar el instalador, saldrá durante un breve instante (del orden de unos pocos segundos) el *prompt* `boot:`, que a menos que se reciba una pulsación de teclado pasará a la siguiente pantalla. Para que el instalador sepa que en ese ordenador ha de configurarse el frontend ha de escribirse en este *prompt* los parámetros `build ksdevice=p2p1 asknetwork`. `build` sirve para indicar al instalador que ese ordenador hará de frontend (de lo contrario asumirá que es un nodo de cálculo) y `ksdevice=p2p1 asknetwork` se utiliza para solicitar por pantalla la configuración de la red, paso que será necesario dadas las características de la red del laboratorio. En resumen e incluyendo el *prompt*, la pantalla quedaría así:

```
boot: build ksdevice=p2p1 asknetwork
```

En caso de omitir esta línea a la hora de instalar el frontend, aparecerá el error "Failed to connect to HTTP server" (que se puede ver en la Figura 2) en un paso posterior de la instalación, ya que, como se ha comentado anteriormente, el programa de instalación habrá asumido que el nodo actual se está instalando como nodo de cálculo y estará intentando obtener la imagen desde algún ordenador de la red externa al laboratorio (pero interna al edificio), que la instalación habrá identificado como probable frontend, pero que evidentemente no lo es, en vez de buscar la imagen a instalar desde el DVD. Si eso llega a suceder, basta con reiniciar el ordenador utilizando el botón correspondiente del panel frontal, o pulsando el botón de arranque durante más de cinco segundos para forzarle a apagarse, soltar, y pulsar de nuevo el de arranque para que vuelva a iniciarse.

La página web del proyecto Rocks Clusters dispone de una guía de instalación que puede encontrarse en [6] y es la que se ha seguido en este paso, aunque con las particularidades que se comentan a continuación.

[3] `http://www.rocksclusters.org/wordpress/?page_id=80` o accediendo a `http://www.rocksclusters.org/` y pulsando con el ratón sobre la sección [Downloads >>] del menú principal.

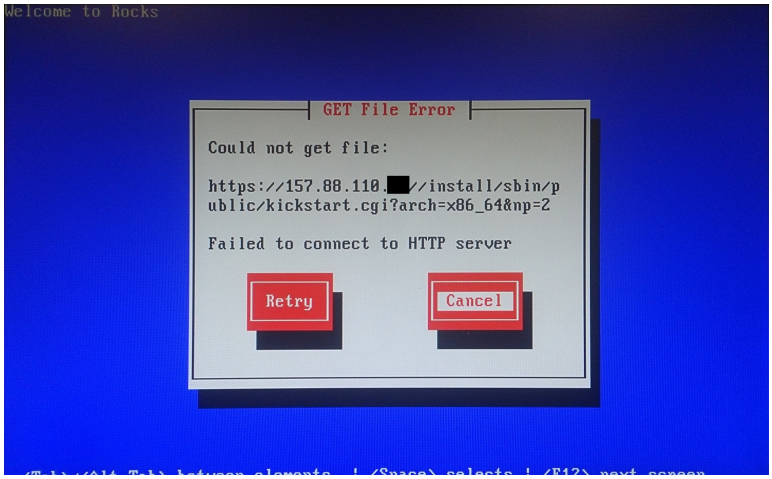

Figura 2: Pantalla que aparece al no insertar la línea de configuración en el arranque durante la instalación del frontend

Selección de interfaz de red externa Una de las primeras pantallas que aparecen pregunta sobre qué interfaz de red desea instalarse Rocks; para ser más específico pregunta por la interfaz de salida a la red externa, no al clúster. En la mayoría de los frontends suele haber dos interfaces; una que se dirige a un switch o router que interconecta todos los nodos entre sí y con el frontend, y otra que sale del frontend a la red externa (internet). Generalmente (y este es el caso para la red del clúster del laboratorio) la tarjeta con mayor ancho de banda se suele dejar para la conexión con el clúster (ya que seguramente sea necesario mover ficheros de datos voluminosos con los datos iniciales o resultados, que además estarán compartidos mediante un sistema de ficheros en red). Con lo cual Eth0, que soporta ethernet 10/100 será la de salida hacia internet y Eth1 (que soporta Gigabit Ethernet y además es PCI Express, ya que pone PCI-E) es la que lleva hacia la red interna. No obstante, utilizando los datos que aparecen en pantalla (como ya hemos dicho, Eth1 es una tarjeta PCI Express, con lo que estará en una de las ranuras de expansión, y además posee capacidad de Gigabit, lo que hace muy probable que tenga dos LEDs, uno para indicar 10/100 y otro para Gigabit, y de la otra no se dice su tipo pero se puede buscar en internet), conviene verificar hacia dónde va cada tarjeta, mirando si el latiguillo[4] conectado a Eth0 lleva al router o switch que sale a internet o al que va hacia el router, haciendo lo mismo con Eth1 y anotando estos datos, ya que nos serán solicitados más adelante de nuevo durante el proceso de instalación del frontend de Rocks.

[4]El término *latiguillo* se utiliza en informática y telecomunicaciones para referirse a los cables para transmisión de datos que acaban, en cada extremo en un conector, generalmente RJ45. En inglés se denominan *patch cord*. Ver `http://en.wikipedia.org/wiki/Patch_cable`.

Por tanto ya que ha de seleccionarse la interfaz de salida a la red externa, y para la configuración del laboratorio, elegiremos Eth0 y pulsaremos Intro, como puede verse en la Figura 3.

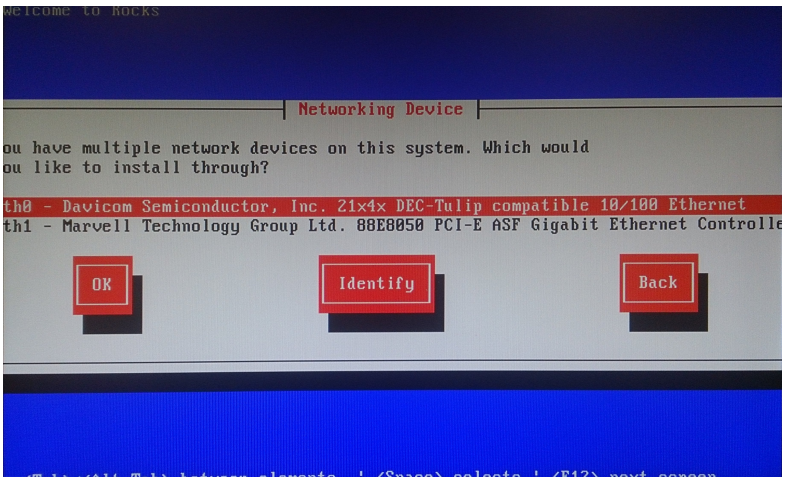

Figura 3: Pantalla de selección de la interfaz hacia la red externa

Select Your Rolls En esta pantalla pulsaremos sobre el botón *CD/DVD-based Roll* que aparece en la sección "Local Rolls". La bandeja de la unidad lectora de CD/DVD expulsará el DVD y la pantalla mostrará el mensaje *Insert the Roll CD/DVD*; basta con volver a introducir la bandeja de la unidad lectora con el disco y pulsar sobre el botón *Continue*.

El programa de instalación de Rocks leerá entonces el disco y buscará los rolls existentes en el disco, que, al tratarse del DVD son todos los disponibles. Para nuestro caso particular debemos marcar las casillas de *area51*, *base*, *ganglia*, *hpc*, *java*, *kernel*, *os*, *perl*, *python*, *sge* y *web-server* es decir, para nuestro propósito hemos de marcar todas las casillas menos *bio*, y *condor* y *xen-5.5*.

Durante la realización de este trabajo se utilizará *sge* para proporcionar control de colas de trabajos; *ganglia* permitirá la monitorización de todos los *hosts* que forman parte del clúster y obtener estadísticas como la carga de red, la carga de CPU, el porcentaje de memoria ocupada, etc.; *web-server* instalará el servidor web Apache, que resulta útil ya que, una vez instalado el frontend, pueden verse los resúmenes proporcionados por gangila accediendo desde el servidor a `http://localhost/ganglia`. El roll *hpc* instala entornos de paso de mensajes tales como OpenRTE, MPI y MPICH, además de ciertos

benchmarks como *stream*[5] o *IOzone*[6], y el software *PVM*[7], que permite utilizar un conjunto heterogéneo (tanto en software como en hardware) de máquinas físicas unidas por red como si se tratara de un ordenador paralelo.

Una vez seleccionados estos rolls para su instalación, ha de pulsarse con el ratón sobre el botón *Submit* para avanzar hacia la siguente pantalla, donde el instalador mostrará, a la izquierda, los rolls marcados en la anterior. Para aceptar la selección realizada, esta vez ha de pulsarse el botón *Next*, situado en la mitad derecha.

En caso de que haber cometido algún error a la hora de haber marcado las casillas de los rolls a instalar, basta con pulsar, en lugar de *Next*, el botón *Insert the Roll CD/DVD*, y el programa vuelve a la pantalla de selección.

Cluster Information El formulario que aparece en esta pantalla sirve para indicar al programa de instalación algunos detalles sobre el clúster. Según la ayuda que aparece al margen, el único campo obligatorio es el primero, el *Fully qualified domain name* o *FQDN*[8], que consiste en el nombre de la máquina que hará de frontend (la que estamos instalando) seguida del nombre de dominio en el que se encuentra el frontend. Por ejemplo, si la máquina se llama *moonstone*, y estamos en el dominio de electrónica (*ele.uva.es*), en este campo se debe escribir `moonstone.ele.uva.es`. En la Figura 4 este campo (que en el formulario aparece como *Fully-Qualified Host Name*) es erróneo, ya que únicamente aparece la parte del nombre de dominio. Esto provocaría que el programa de instalación asumiera que el nombre del ordenador que hace de frontend es *ele* y está en el dominio *uva.es*, lo cual, como hemos comentado anteriormente, es incorrecto. Es importante no cometer este error.

Los otros campos que se pueden rellenar son el nombre que tendrá el clúster entero (en nuestro caso se ha decidido llamar *HPCInformática*, abreviatura del nombre de la asignatura); la organización (*UVa*, abreviatura de Universidad de Valladolid); y los datos sobre la ubicación física del clúster, que son: el nombre de la ciudad (Valladolid), la provincia (*Valladolid*), las siglas del país (*ES* para España), y la latitud y longitud (en el caso del laboratorio 1L022 estos datos son *N41.662371 W04.705688*). También se puede introducir el correo electrónico del administrador del clúster en el campo *Contact* y la URL, si es que tuviera alguna. En nuestro caso, hemos preferido dejar algunos de estos campos con los valores por defecto en vez de en

[5]Ver `https://www.nersc.gov/users/computational-systems/nersc-8-system-cori/nersc-8-procurement/trinity-nersc-8-rfp/nersc-8-trinity-benchmarks/stream/`.

[6]Benchmark diseñado para medir el rendimiento de un sistema de ficheros respecto a operaciones de lectura, escritura, relectura, reescritura, y varoas funciones de C. Ver `http://www.iozone.org/`.

[7]Ver `http://www.csm.ornl.gov/pvm/`.

[8]Ver `http://es.wikipedia.org/wiki/FQDN`.

blanco.

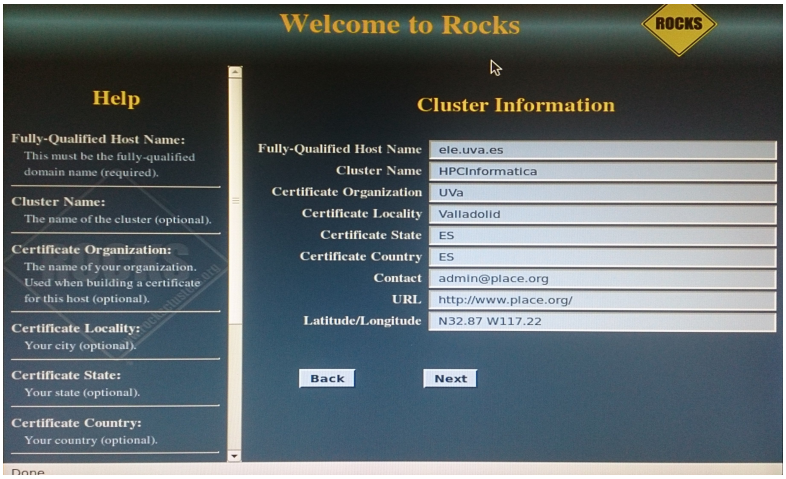

Figura 4: Formulario para entrada de información sobre el clúster. El primer campo es erróneo, debería ser `moonstone.ele.uva.es`

Ethernet Configuration for Public Network Aquí ha de seleccionarse la interfaz de red hacia internet, para lo que son necesarios los datos y la explicación del párrafo 'Selección de interfaz de red externa' en la Sección 2.1. Si en este paso, mirando la ubicación de los conectores, los datos que aparecieron en pantalla, y el router o switch al que conecta el latiguillo que une cada tarjeta, se determinó que la interfaz hacia la red externa ("internet") es `eth0` (como resulta en nuestro caso), ha de selecionarse ésta en la caja desplegable *Public Network Interface*. En el campo *IP address* ha de introducirse la dirección IP del frontend en la red externa; en el caso del ordenador del laboratorio, esta IP ha sido proporcionada por el profesor de la asignatura, como puede verse en la Figura 5. Por último, en el campo *Netmask* ha de escribirse la máscara de subred para la red externa a la que pertenece el frontend. Para el caso de la UVa, que tiene todo el rango de direcciones que comienza por 157.88.XXX.YYY (lo que en el ámbito de las redes se conoce como red de clase B, donde los dos primeros octetos son fijos, y se permite direccionar hasta 65534 equipos diferentes[9]), cada dirección XXX se asigna a un grupo, departamento o edificio diferente. Por ejemplo, hasta hace poco tiempo, el Grupo Universitario de Informática disponía de todas las IP que comenzaran por 157.88.36.YYY, el departamento de electrónica todos los que empezaran por 157.88.110.YYY, etc. (en una idea similar a la de un rango de clase C), aunque es posible que algunos departamentos dispongan varios rangos .

[9]Ver `http://es.wikipedia.org/wiki/Direccion_IP`.

Las máscaras de subred se forman, si las vemos como un número binario, con "1" en la parte que identifica la red, y "0" en la parte que identifica al host, esto es, la parte YYY de las direcciones IP escritas anteriormente. Por tanto, la máscara de subred para el departamento de Electrónica deberá ser `255.255.255.0`.

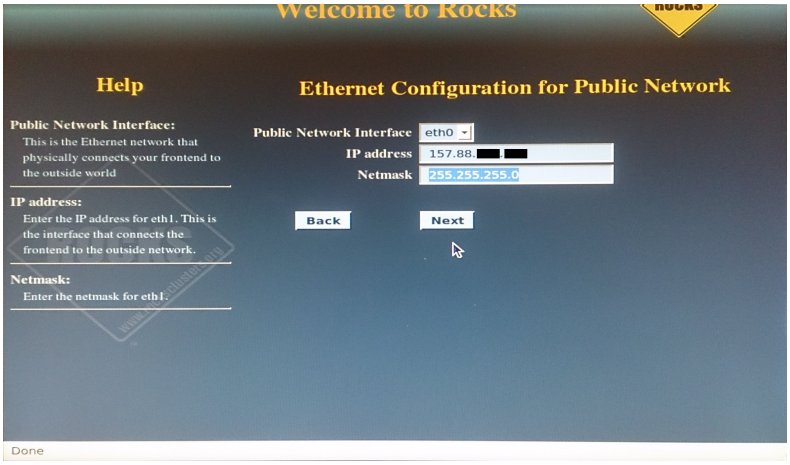

Figura 5: Formulario para introducir la configuración de la red externa

Ethernet Configuration for Private Network Al igual que en el apartado anterior sobre configuración ethernet para la red externa, en este caso se elige en la caja desplegable la interfaz cuyo latiguillo conecta con el router (o switch) que a su vez está conectado con los nodos de cálculo. Ya que en nuestro caso, únicamente existen dos tarjetas de red en el frontend y eth0 es la que va hacia la red externa, `eth1` es, por descarte la que conecta con la red de nodos del clúster y es la que debe seleccionarse. El resto de valores se han dejado tal y como vienen por defecto, es decir, `10.1.1.1` como dirección IP y `255.255.0.0`, lo que permite añadir hasta 65534 nodos al clúster. Esto puede verse en la Figura 6.

La guía de instalación oficial de Rocks para el frontend en [6] recomienda no cambiar estos valores por defecto salvo que se den circunstancias especiales que obliguen a elegir otros valores diferentes, y añade que, aunque el programa de instalación permite elegir la misma interfaz de red tanto para la red interna como para la externa, elegir la misma en ambos formularios es un error. Cuando la instalación detecta que únicamente existe una tarjeta de red crea una interfaz de red virtual del estilo a `eth0:0`.

Para continuar hacia la siguiente pantalla ha de pulsarse con el ratón sobre el botón *Next*.

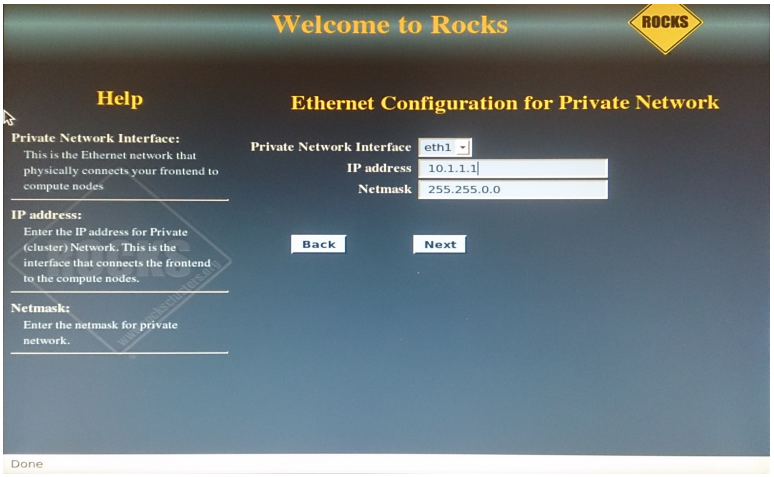

Figura 6: Formulario para introducir la configuración de la red interna del clúster

Miscellaneous Network Settings En este paso el programa de instalación solicita la dirección IP del gateway, puerta de enlace o pasarela[10] de la red externa, a la que pertenece el frontend del clúster. Tradicionalmente, los gateways se utilizan para conectar una red (en nuestro caso, la red externa a la que pertenece nuestro frontend) con otra red más exterior, generalmente Internet, traduciendo las direcciones, cuando las dos redes utilizan protocolos y/o arquitecturas diferentes (por ejemplo, Ethernet y Token Ring). En este caso la red 157.88.110.YYY a la que pertenece nuestro frontend es ya una dirección IP pública y además utiliza Ethernet, probablemente igual que la red "más exterior" (157.88.XXX.YYY), con lo cual no sería un gateway propiamente dicho sino un router o switch. En el Departamento de Electrónica, la dirección del equipo que realiza está función ha sido proporcionada por el profesor de la asignatura y ha de introducirse en el campo que aparece en la parte superior del formulario, como puede verse en la Figura 7.

En cuanto a los servidores de nombres de dominio que se deben utilizar en nuestro caso, también han sido proporcionadas por el profesor de la asignatura.

Root Password Este formulario sirve para introducir la contraseña del usuario *root*, y ha de introducirse dos veces (una en cada caja de texto) para evitar que pueda ser introducida incorrectamente, y una vez terminada la instalación, no se pueda acceder.

Como consejo cabe citar que, aunque en un principio parezca buena idea utilizar caracteres no comunes en la contraseña, tales como @, _, &, etc., su uso puede presentar el problema de que, dado que rocks se instala por defecto

[10] Ver `http://es.wikipedia.org/wiki/Puerta_de_enlace`.

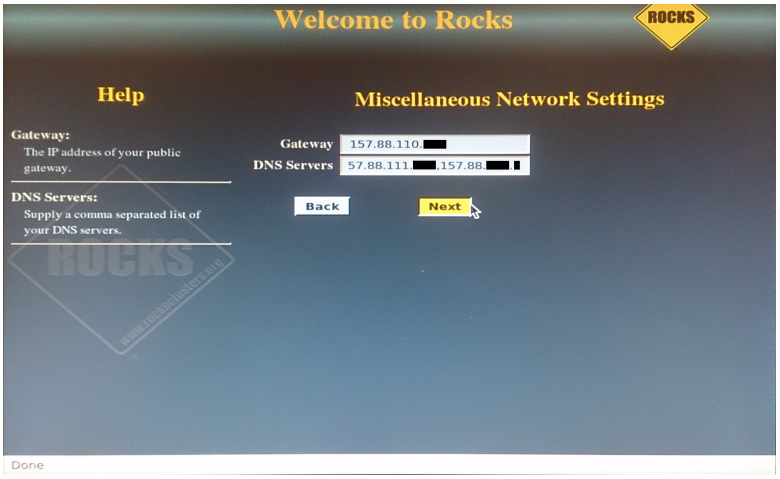

Figura 7: Formulario de configuración extra de la red

con la configuración en inglés, pueda resultar difícil introducirla posteriormente ya que las teclas del teclado no se corresponden con la configuración de éste en el sistema operativo. Lógicamente, si la tecla de un teclado español 'ñ' corresponde en el teclado inglés a la tecla ':', la primera vez que entremos en el sistema operativo no habrá problema ya que esto sucederá en las dos ocasiones, tanto al seleccionar la contraseña como al introducirla para entrar por primera vez, pero puede plantear problemas ya que lo más natural es que deseemos cambiar la configuración de CentOS para utilizar el teclado español. Por este motivo se desaconseja su uso, limitándose únicamente a caracteres del alfabeto inglés que sean mayúsculas, minúsculas y números, en la contraseña.

Time Configuration En este formulario se solicita introducir la configuración sobre la zona horaria en la que está ubicado el clúster y la dirección de un servidor NTP. La zona horaria sirve para saber la diferencia horaria entre el meridiano de Greenwich y la hora local. En este caso y a fecha de redacción, España está situada una hora por delante de la hora media de Greenwich (GMT), lo que se denota como GMT+1. Para el caso del instalador, esta diferencia se indica seleccionando en la caja desplegable *Time Zone*, el continente y la cuidad que hace de capital de provincia, comunidad autónoma, estado o país donde está ubicado el clúster, que en el caso de España es `Europe/Madrid`, como se ve en la Figura 8.

En cuanto al servidor de tiempo (*NTP Server*)[11], que sirve para que el equipo obtenga la fecha y hora de forma actualizada, podemos dejar el que viene por defecto, que es `pool.ntp.org`. Estos servidores forman parte de un proyecto que agrupa varios equipos en todo el mundo con la hora exacta,

[11] Ver `http://es.wikipedia.org/wiki/Network_Time_Protocol`.

aunque existe un problema: debido a la latencia de la red, un ordenador que solicite la hora a un servidor puede tener pequeñas diferencias que se acentúan si el servidor NTP está físicamente ubicado lejos de nuestro equipo. El protocolo NTP corrige parcialmente este problema solicitando la fecha y hora a varios servidores y haciendo una operación estadística, conocida como algoritmo de Marzullo[12] ente los valores devueltos. Aun así, el problema no se corrige completamente ya que al estar físicamente lejos, puede haber redes intermedias con mucho tráfico o con poco ancho de banda. Por ello, es mejor utilizar servidores NTP cercanos, y, aunque el que viene por defecto (`pool.ntp.org`) es buena solución, asigna servidores de forma aleatoria, para cada petición, repartidos por todo el mundo. En su lugar, puede utilizarse `es.pool.ntp.org`, que asigna las peticiones a servidores únicamente en la zona de España, o incluso `hora.roa.es`, que es el servidor NTP del Real Instituto y Observatorio de la Armada, que se encuentra en San Fernando, Cádiz. Este último (`hora.roa.es`) es el servidor NTP que mantiene la hora oficial española.

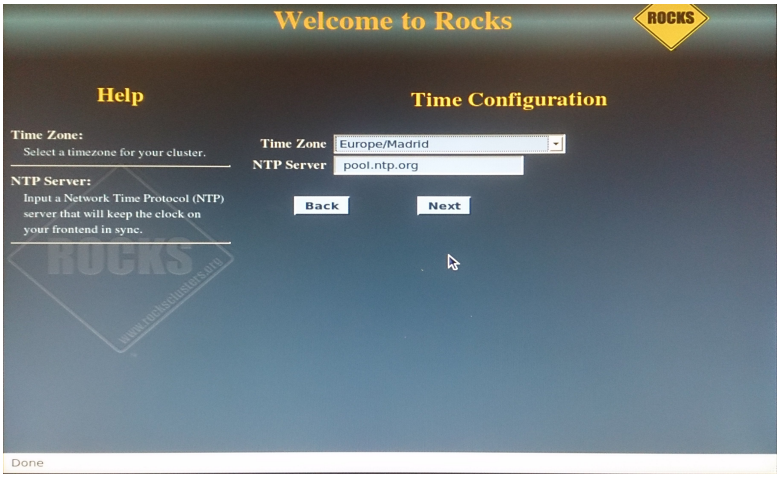

Figura 8: Formulario para introducir la configuración horaria

Disk Partitioning Esta pantalla sirve para seleccionar si se desea que el programa de instalación realice un particionado automático de los discos o, por el contrario, si se desea configurar de forma manual. De acuerdo a la guía de instalación en [6], el particionado automático reformateará todo el espacio del primer disco que encuentre de acuerdo a la siguiente tabla de particiones:

Donde la partición montada sobre `/export` está simbólicamente enlazada con `/state/partition1`. A su vez, todos el contenido que se mueva o cree dentro de la carpeta `/share` aparecerá sobre `/state/partition1`, como si

[12] Ver `http://en.wikipedia.org/wiki/Marzullo's_algorithm`.

Nombre de partición	Tamaño
/	16 GB
/var	4 GB
swap	1 GB
/export	resto del disco

Cuadro 1: Tabla de particiones para el primer disco del frontend en particionado automático

se tratara de un subconjunto dentro de esta última. La peculiaridad de esta partición es que se comparte a través de la red con todos los nodos, de forma que tanto los programas como bibliotecas a ejecutar, y los datos de entrada y salida pueden ser accedidos por el clúster para poder realizar operaciones en paralelo.

El resto de discos presentes en el ordenador quedarán sin modificación alguna, aunque esto no es problema ya que en cada ordenador existe únicamente un disco duro (reconocido como *sda*). Para nuestro caso, se ha elegido la opción de particionado automático.

Por último, la pantalla pasará a negro y aparecerá una barra de progreso que indicará los pasos de particionado y formateo del disco duro, como puede verse en la Figura 9.

Figura 9: Barra de progreso indicando el estado del paso de particionado y formateo

Una vez la barra de progreso se complete, comenzará el proceso de instalación del sistema operativo CentOS, y posteriormente se instalarán los diferentes *rolls* que hayamos seleccionado anteriormente, tal y como aparece en la Figura 10.

Una vez completada la instalación, el sistema operativo se reiniciará, y entraremos con el usuario root y la contraseña elegida.

2.2. Instalación de los nodos

En el laboratorio, los nodos están ubicados tanto en la balda superior de la mesa como en el armario enracable. Existen tres tipos de nodos: los normales; los que tienen tarjeta gráfica NVIDIA y los enracables. Tenemos que registrar todos nuestros nodos en el frontend.

Antes de empezar, debemos asegurarnos que todos los nodos están apagados, y el único ordenador encendido en la red es el frontend. Si existiera

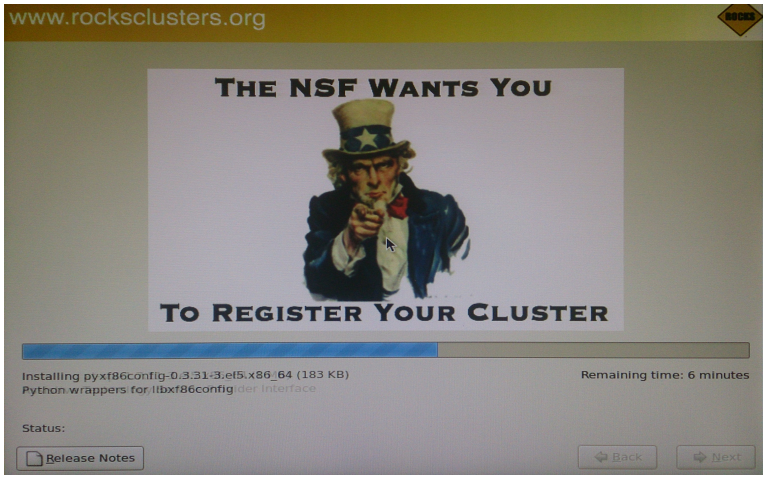

Figura 10: Instalación del sistema operativo y los *rolls*

alguno encendido, lo apagaremos. Una vez estemos listos, escribiremos en un terminal de nuestro frontend el siguiente comando:

```
[root@cluster ~]# insert-ethers
```

En el caso particular de nuestro clúster, todos los nodos serán de cálculo, con lo cual en la pantalla que nos aparece en el terminal (ver Figura 11), seleccionaremos "Compute" (viene seleccionada por defecto en Rocks 5.5) y presionaremos intro.

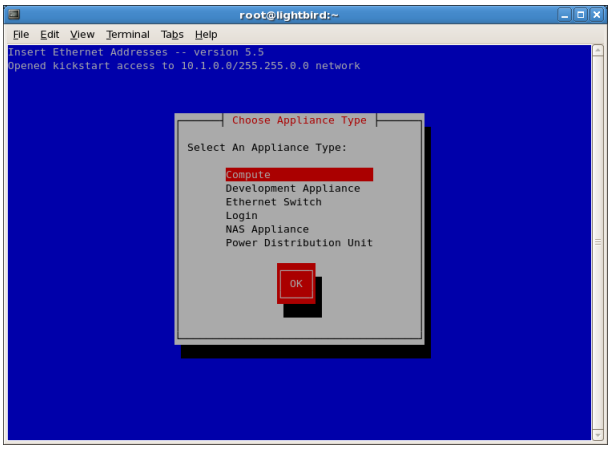

Figura 11: Selección del tipo de nodos a instalar.

A continuación, arrancaremos únicamente uno de los nodos normales, el que queramos que aparezca como nodo normal número 0, y le conectaremos una pantalla y un teclado[13]. En caso de que un nodo tenga tanto una tarjeta

[13]Parece no ser totalmente necesario para todos los nodos conectarle la pantalla y teclado

PCI, AGP, PCI express o similares y otra integrada en la placa base, lo normal es que la señal de vídeo siempre salga por las primeras y no por la integrada.

El script de registro de nodos permite realizar grupos de nodos por armarios (cabinets) o racks; nosotros lo usaremos para agruparlos por tipos (como dijimos antes, normal, con gráfica y enracables). Rocks nombra cada nodo con el formato xxxxxx-yy-zz, donde xxxxxx designa el tipo de nodo (la mayor parte de las veces será "compute"), yy es el número de armario (que es lo que nosotros usaremos para identificar el tipo de nodo; 0 serán los normales, 1 serán los que tienen tarjeta gráfica y 2 serán los enracables), y por último zz es el número de nodo en el grupo yy. Para especificar el número de armario, hay que añadir al comando insert-ethers el párametro --cabinet yy, donde yy es, siguiendo la nomenclatura anterior, el número de armario.

Al cabo de un rato, aparecerá en la pantalla la dirección MAC del nodo que acabamos de arrancar, el nombre que el script le ha dado (dado que hemos ejecutado insert-ethers sin especificar el armario, el primero va a ser *compute-0-0*) y unos paréntesis que indican si el nodo ha solicitado correctamente el fichero Kickstart[14] (si aún no lo ha hecho, aparecen dos paréntesis vacíos "( )" y cuando ya lo ha solicitado correctamente aparece un asterisco entre medias "(*)".

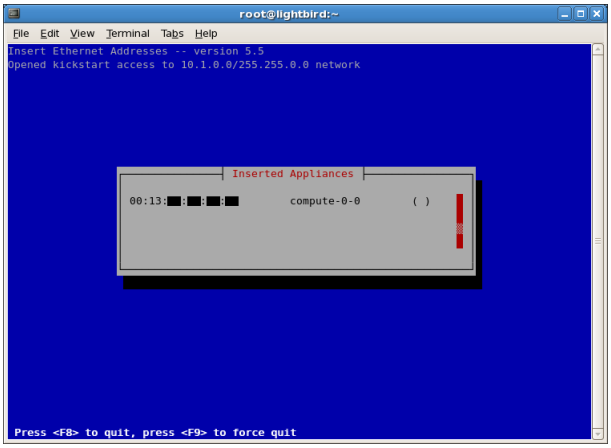

Figura 12: Detección del primer nodo de cálculo en la red, antes de solicitar el fichero Kickstart.

Una vez aparezca el primer nodo marcado con el asterisco entre paréntesis y en la pantalla conectada a éste desaparezca la barra de progreso de

al arrancarlo para añadirlo al sistema con insert-ethers, pero algunos de ellos no han llegado siquiera a solicitar la imagen a instalar por PXE si no detectan la conexión de la pantalla.

[14] Ver `http://es.wikipedia.org/wiki/Kickstart_(Linux)`

formateo de particiones[15], podemos desconectar la pantalla, conectarla al siguiente nodo que queramos instalar e iniciarle; tras un rato aparecerá en el programa insert-ethers una línea con los datos del nuevo nodo (que el programa llamará *compute-0-1*), con los paréntesis vacíos, y al rato aparecerá el asterisco entre medias.

Podemos seguir este procedimiento hasta haber añadido todos los nodos normales de nuestro cluster al programa, en cuyo caso cerraremos insert-ethers simplemente pulsando F8 en el teclado. Sin embargo, es posible, una vez que todos los nodos detectados aparezcan marcados con el asterisco, interrumpir el programa para seguir añadiendo nodos más adelante, pulsando F8. La siguiente vez que arranquemos el programa con los mismos parámetros (en caso de los nodos normales, sin parámetros o con --cabinet 0) éste continuará con la numeración.

2.2.1. Error "Reboot and Select proper Boot device"

A la hora de instalar un nodo, es posible que aparezca el bootsplash (la pantalla de la BIOS en la que aparece el logotipo del fabricante) y a continuación el mensaje que aparece en la Figura 13:

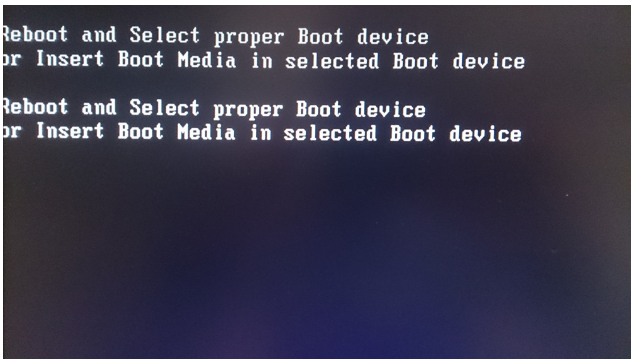

Figura 13: Error que aparece tras el bootsplash de la BIOS cuando PXE no está activado.

Esto suele deberse a que la opción PXE está desactivada en la BIOS o el orden de arranque no es el correcto. Para corregirlo, basta con pulsar la tecla que nos permita acceder a la BIOS durante el arranque (F2 para las BIOS de Intel), e ir a la pestaña "BOOT" utilizando las flechas horizontales del teclado. Allí, es muy probable que nos encontremos una imagen similar

[15]Realmente puede no ser necesario esperar tanto, siendo quizás posible desconectarle la pantalla al arrancarse el entorno X.org, o incluso nada más aparecer el asterisco entre los paréntesis en insert-ethers pero hemos preferido hacerlo al acabar el formateo para dar más margen al programa y evitar problemas. Además, el hecho de tener la pantalla conectada permite detectar y corregir errores en la instalación.

a la de la Figura 14, con la opción de PXE desactivada y "Silent Boot" activado, o lo que es lo mismo, no mostrar mensajes de arranque:

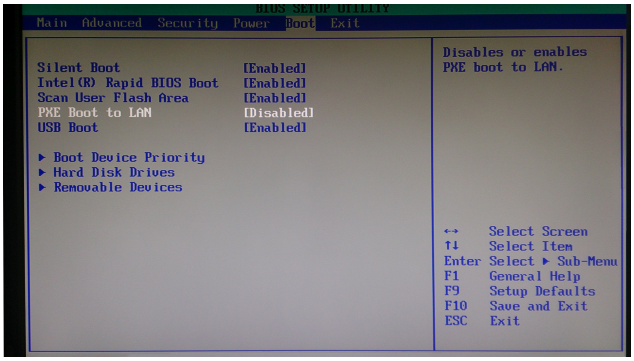

Figura 14: Configuración en la pestaña de arranque en una BIOS Intel con PXE y verbose (como opuesto a "Silent Boot") desactivadas.

Debemos cambiar la configuración de ambas, es decir, habilitar PXE y deshabilitar "Silent Boot"[16], usando las flechas verticales (arriba y abajo) del teclado para posicionarse encima de la opción, que aparecerá en blanco, pulsando Enter cuando estemos sobre la opción deseada para que aparezca una lista, volviendo a usar las flechas verticales para activar o desactivar esa opción, y pulsando Intro de nuevo para validarla. La pantalla anterior debería quedar, entonces, como se muestra en la Figura 15:

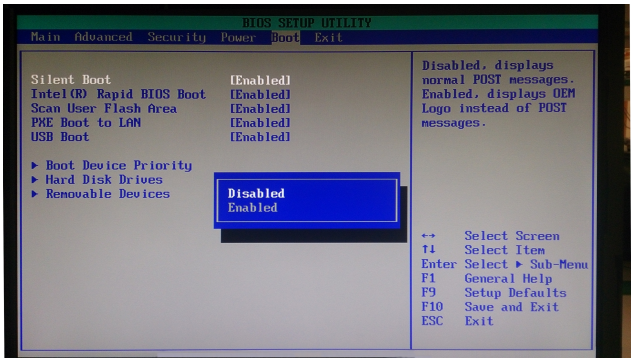

Figura 15: Configuración correcta en la pestaña de arranque en una BIOS Intel, antes de pulsar intro para dejar deshabilitada (*Disabled*) la opcion "Silent Boot".

Pulsamos F10 para salir guardando los cambios y el ordenador se reiniciará. Volveremos a entrar en la BIOS, navegando con las flechas horizontales

[16]En realidad no es obligatorio deshabilitar "Silent Boot" para corregir este problema, pero siempre es recomendable que la BIOS nos muestre mensajes sobre el estado del sistema que nos puedan servir como debug.

para navegar de nuevo hasta la pestaña “BOOT”, y allí entrar en la opción “Boot Device Priority” para modificar el orden en el que la BIOS intentará arrancar de los dispositivos de almacenamiento o red. Aquí debemos verificar y en caso negativo, modificar esta lista para que el primer dispositivo sea PXE, el segundo la disquetera o unidad de CD y por último el disco duro, como se ve en la Figura 16. Finalmente, pulsamos F10 para guardar los cambios y salir.

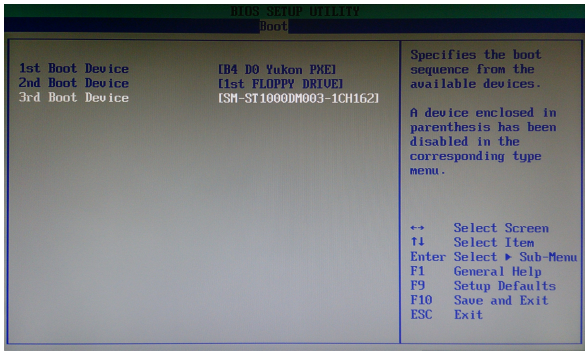

Figura 16: Lista de orden de arranque de la BIOS correcto para un nodo de cómputo del clúster.

El reinicio que hemos hecho tras habilitar PXE y guardar cambios es completamente necesario, ya que sin él, no nos aparecerá la opción PXE en la lista del orden de arranque, tal como sucede en la Figura 17:

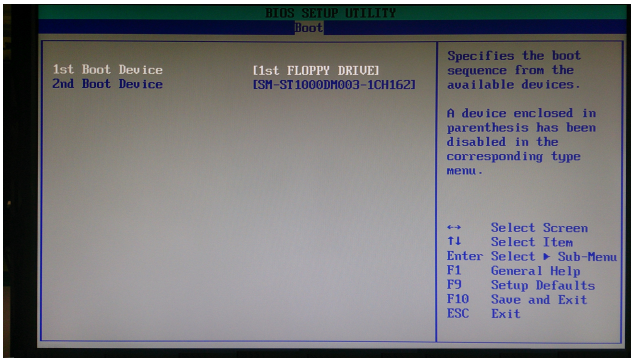

Figura 17: Apariencia de la lista de orden de arranque de la BIOS cuando no se han guardado los cambios y reiniciado tras habilitar PXE.

2.2.2. Algunos comandos útiles en Rocks

A la hora de gestionar un cluster con Rocks, existen una serie de comandos que pueden ser de mucha utilidad. La mayoría de ellos se ejecutan a través del ejecutable rocks, que en realidad se trata de un script en python.

Una lista completa de todas las opciones disponibles con rocks está disponible en [5], y se pueden encontrar otros comandos útiles en [3].

Sincronizar la configuración El sistema operativo Rocks utiliza una serie de ficheros y una base de datos para llevar cuenta de la configuración. Cuando se hacen ciertos cambios en ésta, conviene sincronizar todos estos ficheros y base de datos para mantenerlos actualizados. El comando que permite llevar a cabo esta acción es rocks sync config:

```
[root@moonstone ~]# rocks sync config
```

Sincronizar los datos de usuarios Puede ser necesario añadir nuevos usuarios al clúster, ya sea para ejecutar determinados programas o bien que se trate de usuarios reales que necesiten acceso a las capacidades de cómputo que ofrece. También es común que los datos de los usuarios sean modificados, tales como contraseñas o similares.

Para estos casos existe el comando rocks sync users, que debería ser ejecutado después de estas acciones desde el frontend. De esta forma, los cambios que se hayan hecho en los usuarios desde uno de los equipos (generalmente desde el frontend) se propagarán al resto de ordenadores en el clúster, actualizando ficheros como `/etc/passwd`, `/etc/shadow`, etc.:

```
[root@moonstone ~]# rocks sync users
```

Eliminar un nodo de la lista Cuando se están instalando los nodos de cómputo puede llegar a suceder, si no se tiene cuidado de arrancar los ordenadores individualmente, primero iniciando uno, esperando a que lo detecte insert-ethers, y una vez detectado e instalando arrancar el siguiente, que se arranquen dos ordenadores seguidos y se desconozca qué ordenador es cuál. También puede suceder que se añada uno de los nodos como perteneciente a un armario (o grupo) usando insert-ethers --cabinet C y realmente el número de armario o grupo se haya introducido incorrectamente.

Para esas situaciones resulta útil eliminar el nodo que se acaba de añadir mediante rocks remove host <nombre-host>; por ejemplo para eliminar el nodo *compute-0-4*:

```
[root@moonstone ~]# rocks remove host compute-0-4
```

Reinstalar un nodo Durante la instalación de los nodos a veces puede suceder que un nodo sea reconocido e introducido en la base de datos de nodos usando insert-ethers, y por alguna razón, la instalación falle y dicho nodo no sea capaz de iniciar correctamente el sistema operativo, o se quede esperando el fichero Kickstart y por algún motivo, el frontend no se lo envíe o no le llegue, como puede verse en [7].

Uno de los comandos de rocks da la posibilidad de marcar un nodo para que el sistema operativo sea reinstalado la siguiente vez que éste se reinicie. Por tanto, el primer comando, rocks set host boot compute-0-4 action=install marca el nodo para reinstalación al siguiente reinicio y el segundo, en caso de que se tenga acceso al terminal desde el frontend usando *ssh*, permite reiniciarlo (en los casos en los que no sea posible reiniciar a través de *ssh*, se puede intentar conectar una pantalla y teclado al nodo y escribir shutdown -r now, y si no, reiniciarle del botón):

```
[root@moonstone ~]# rocks set host boot compute-0-4 action=install
[root@moonstone ~]# ssh compute-0-4 'reboot'
```

Saber los nodos ya instalados Existen algunas ocasiones, como cuando ha de instalarse un gran numero de nodos, en las que se ha perdido la cuenta de qué nodos han sido ya instalados y cuáles falta por instalar. Un comando que puede ayudar con esto es rocks list host, que además de proporcionar los nombres de los host (tanto el frontend como los nodos de cómputo), muestra el número de CPUs, el armario o rack al que pertenecen (en este caso sirve para denotar el grupo al que pertence el nodo) y el número de equipo dentro del armario (en este caso el número de equipo dentro del grupo):

```
[root@moonstone ~]# rocks list host
HOST            MEMBERSHIP CPUS RACK RANK RUNACTION INSTALLACTION
moonstone:      Frontend   2    0    0    os        install
compute-0-0:    Compute    2    0    0    os        install
compute-0-1:    Compute    2    0    1    os        install
compute-0-2:    Compute    2    0    2    os        install
compute-0-3:    Compute    2    0    3    os        install
compute-0-4:    Compute    2    0    4    os        install
compute-1-0:    Compute    2    1    0    os        install
compute-1-1:    Compute    2    1    1    os        install
compute-1-2:    Compute    2    1    2    os        install
compute-0-5:    Compute    2    0    5    os        install
compute-0-6:    Compute    2    0    6    os        install
compute-0-7:    Compute    2    0    7    os        install
compute-0-8:    Compute    2    0    8    os        install
compute-0-9:    Compute    2    0    9    os        install
compute-0-10:   Compute    2    0    10   os        install
compute-0-11:   Compute    2    0    11   os        install
compute-0-12:   Compute    2    0    12   os        install
compute-0-13:   Compute    2    0    13   os        install
compute-2-0:    Compute    4    2    0    os        install
compute-1-3:    Compute    2    1    3    os        install
compute-2-1:    Compute    4    2    1    os        install
compute-2-2:    Compute    4    2    2    os        install
```

3. Instalación de las librerías de cálculo

La instalación de librerías un clúster Rocks se realiza siguiendo procedimientos similares a los que se requieren para instalar este tipo de software en un servidor Linux. Sin embargo, para que dichas librerías puedan estar accesibles desde todos los nodos de cómputo (es decir, para que los programas que se ejecuten en paralelo puedan encontrar los componentes necesarios tales como bibliotecas, archivos de configuración, repositorios temporales, etc.) es necesario configurar estas librerias y los sistemas de archivos de manera apropiada.

Como se comentó anteriormente en la Sección 2.1, en los clústers Rocks, existe una particion en cada ordenador que los conforma (tanto nodos de cómputo como frontend) reservada para ser montada a través de la red y está montada sobre `/export`, que a su vez está simbólicamente enlazada con `/state/partition1`. No obstante en nuestro caso utilizaremos la carpeta `/share` para trabajar con todos aquellos directorios, bibliotecas y programas que queramos exportar a todos los ordenadores del clúster, ya que cualquier fichero o carpeta que se mueva a `/share` aparecerá también en `/state/partition1`.

Es por ello que el primer paso consiste en crear una carpeta dentro de `/share` para poner las aplicaciones y bibliotecas, a la que denominaremos `/share/apps`:

```
[root@moonstone ~]# mkdir /share/apps/
[root@moonstone ~]# ls -lFd /share/apps
```

En caso de que el comando anterior retorne algún fallo, se puede intentar ejecutar el mismo procedimiento de otra manera, verificando al final que la carpeta haya sido creada:

```
[root@moonstone ~]# cd /share/
[root@moonstone ~]# mkdir apps
[root@moonstone ~]# ls -lFd /share/apps
```

El último paso (utilizando ls -lFd /share/apps es completamente obligatorio ya que es la forma que tenemos de obligar al sistema operativo de que automonte la partición compartida por red.

A lo largo de esta sección procederemos a incorporar al clúster las siguientes librerías:

1. **BLAS. -** Librería compuesta por un conjunto de rutinas para realizar operaciones vectoriales y matriciales básicas.

2. **ATLAS. -** Provee rutinas de cálculo de álgebra lineal mejorando las proporcionadas por BLAS, ya que esta librería posee optimizaciones específicas para diferentes arquitecturas de microprocesadores (Implementación eficiente y portable de la biblioteca BLAS y de ciertas funciones de la biblioteca LAPACK).

3. **LAPACK. -** Conjunto de subrutinas escritas en Fortran 90 para resolver problemas típicos de álgebra lineal: resolución de sistemas ecuaciones lineales simultáneas, ajuste por mínimos cuadrados, problemas de autovalores y valores singulares.

4. **FFTW.-** Librería de subrutinas en C para el cálculo de la transformada discreta de Fourier (DFT) en una o más dimensiones, con un tamaño de entrada arbitrario, para datos reales y complejos (así como para datos pares e impares, es decir transformadas discretas seno/-coseno).

5. **LAMMPS.-** Programa de dinámica molecular con opciones para materiales blandos (biomoléculas, polímeros), sólidos (metales, semiconductores) y de grano grueso o mesoscópicos. Puede utilizarse para modelar átomos o, en general, como un simulador de partículas paralelo a escala atómica, mesoscópica o continua.

3.1. Basic Linear Algebra Subprograms (BLAS)

BLAS es un acrónimo de *Basic Linear Algebra Subprograms* (del inglés Subprogramas de Álgebra Lineal Básica). Se trata de un conjunto de subrutinas matemáticas que permiten realizar operaciones con vectores y matrices [10]. Estas subrutinas están divididas en tres conjuntos o *niveles* de acuerdo al tipo de sus argumentos: las funciones *Level 1 BLAS* realizan operaciones escalares, sobre un vector y también vector-vector; las funciones *Level 2 BLAS* realizan operaciones matriz-vector y, Por último, las funciones *Level 3 BLAS* realizan operaciones matriz-matriz. Debido a que BLAS es eficiente, portable y se encuentra fácilmente disponible, se utiliza a menudo en el desarrollo de software de álgebra lineal, como LAPACK.

Su instalación es muy sencilla; únicamente se necesita obtener el fichero comprimido que contiene todos el código fuente de BLAS de su página oficial[17] usando `wget`, descomprimirlo, mover la carpeta obtenida de la descompresión a `/share/apps/` y por último ejecutar un `make all` para compilar el código fuente:

```
[root@moonstone ~]# wget http://www.netlib.org/blas/blas.tgz
[root@moonstone ~]# tar xvf blas.tgz
[root@moonstone ~]# mv BLAS/ /share/apps/
[root@moonstone ~]# cd /share/apps/BLAS/
[root@moonstone BLAS]# make all
```

3.2. Linear Algebra Package (LAPACK)

LAPACK es una biblioteca que permite efectuar operaciones de álgebra lineal, tales como resolución de sistemas de ecuaciones, ajuste por el método

[17] `http://www.netlib.org/blas/`

de mínimos cuadrados, problemas de autovalores, descomposición en valores singulares y factorización de matrices [12]. La biblioteca LAPACK depende de la instalación de la biblioteca BLAS, aunque el fichero que contiene el código fuente también dispone de una versión de ésta; aunque se permite el enlazado con una biblioteca ya compilada previamente, con el objetivo de evitar cualquier posible problema de compatibilidad, conviene tener por una parte la versión que instalamos anteriormente, y por otra permitir que LAPACK compile y utilice su propio BLAS, en un directorio diferente.

Los detalles sobre la instalación y el uso de LAPACK pueden leerse con mayor extensión en las referencias [1] y [2].

Para instalar LAPACK, primero es necesario descargarse el fichero comprimido que contiene el código fuente, la documentación y los ficheros de compilación para `make` de la página oficial ddel proyecto. Una vez en el equipo, ha de descomprimirse y mover la carpeta resultante al directorio `/share/apps/` para que las bibliotecas puedan ser compartidas entre todos los hosts del clúster.

LAPACK se instala utilizando ficheros `Makefile`. Dentro del directorio descomprimido puede encontrarse un `Makefile` de prueba, llamado `make.inc.example`, que debe editarse para adaptar la instalación a las características software del clúster:

```
[root@moonstone ~]# wget http://www.netlib.org/lapack/lapack-3.5.0.tgz
[root@moonstone ~]# tar xvf lapack-3.5.0.tgz
[root@moonstone ~]# mv lapack-3.5.0/ /share/apps/
[root@moonstone ~]# cd /share/apps/lapack-3.5.0/
[root@moonstone lapack-3.5.0]# cp make.inc.example make.inc
[root@moonstone lapack-3.5.0]# vim make.inc
```

Una vez editado, basta con ejecutar `make` para comenzar con la instalación:

```
[root@moonstone lapack-3.5.0]# make
```

3.2.1. Verficando que la carpeta compartida es visible

Llegados a este punto, es interesante ver si existe algún problema con la exportación de los directorios, esto es, si los nodos de cálculo están viendo correctamente el contenido de la carpeta compartida /share. Para verificarlo, simplemente nos logueamos en uno de los nodos (por ejemplo, *compute-0-0* y vemos el contenido de la carpeta en este nodo.

```
[root@moonstone ~]# ssh compute-0-0
Last login: Wed May 14 15:29:45 2014 from moonstone.local
Rocks Compute Node
Rocks 5.5 (Mamba)
Profile built 10:55 27-Mar-2014

Kickstarted 11:02 27-Mar-2014
[root@compute-0-0 ~]# cd /share/
```

```
[root@compute-0-0 share]# ls
[root@compute-0-0 share]#
```

En el listado anterior podemos ver que la carpeta /share/ del nodo está vacía. Esto se puede deber a que el frontend no está exportando correctamente esta carpeta o que los nodos no tienen montada la carpeta de red. Podemos comprobar esto mirando las carpetas que están montadas y viendo si la carpeta /share/ (que usa el sistema de ficheros NFS[18]) está montada en el nodo o no:

```
[root@compute-0-2 share]# mount
/dev/sda1 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda2 on /var type ext3 (rw)
/dev/sda5 on /state/partition1 type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

Debería aparecer al menos una entrada con tipo *nfsd*, que no está. En Rocks 5.5, podemos hacer que el sistema de ficheros automonte /share/ solicitando el contenido de un directorio que sabemos que existe, en nuestro caso sabemos (porque la hemos creado en el frontend) que dentro de /share/ existe la carpeta apps. Aunque sabemos que dentro del nodo no está, si hacemos un ls desde el nodo a un directorio dentro de /share/ que sí que esté en el frontend, el nodo automontará el disco de red.

Antes de nada, conviene sincronizar los datos de los usuarios desde el frontend antes de provocar el automontaje del directorio en cada nodo:

```
[root@moonstone ~]# rocks sync users
```

Volvemos al terminal donde tenemos una conexión por ssh al nodo *compute-0-0* y solicitamos el contenido de la carpeta /share/apps. Volvemos a hacer un ls para verificar que el directorio está correctamente montado:

```
[root@compute-0-0 ~]# ls /share/
[root@compute-0-0 ~]# ls /share/apps/
BLAS  lapack-3.5.0
[root@compute-0-0 ~]# ls /share/
apps
[root@compute-0-0 ~]#
```

Si ahora ejecutamos un mount, podemos encontrar, al final del todo, el demonio nfs (*nfsd*) que utiliza /share para ser compartida a través de la red:

```
[root@compute-0-0 ~]# mount
```

[18] Network File System, del inglés *Sistema de ficheros en red*. Ver `http://es.wikipedia.org/wiki/Network_File_System`.

```
/dev/sda1 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda2 on /var type ext3 (rw)
/dev/sda5 on /state/partition1 type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
```

3.3. Fastest Fourier Transform in the West (FFTW)

Como hemos comentado antes, FFTW es una biblioteca que permite obtener la transformada discreta de Fourier (DFT) implementando un conjunto de algoritmos que presentan una serie de optimizaciones para un cálculo más rápido, en lo que se conoce como transformada rápida de Fourier (FFT), que suelen estar basados en el uso de matrices. FFTW estima cuál de todos los algoritmos es el que obtendría los resultados en menor tiempo en unas circunstancias concretas, alcanzando una cota superior de $O(n\ log\ n)$ en tiempo en todos los casos [11].

Aunque la biblioteca se ha descargado, no se ha llegado a instalar dado que Intel Parallel Studio ya provee una interfaz para adaptar las llamadas a funciones de FFTW a sus propias librerías, como se verá posteriormente.

3.4. Automatically Tuned Linear Algebra Software (ATLAS)

El software de álgebra lineal ajustado automáticamente (ATLAS) es una importante mejora de BLAS, utilizando una serie de scripts que detectan la configuración del sistema y producen un BLAS altamente optimizado para el ordenador concreto [9]. Como en este caso los nodos de cómputo en un mismo bloque tienen las mismas características y disponemos de tres bloques, produciremos tres versiones diferentes, una para los nodos normales, otra para los que disponen de gráfica y otro para los enracables, que en este caso son Xeon.

Existe un manual de instalación en PDF dentro del fichero comprimido que contiene ATLAS, que hasta la fecha, es el más actualizado, y puede encontrarse en `ATLAS/doc/atlas_install.pdf`, al descomprimirlo, aunque se puede obtener de internet [8]. En este trabajo se seguirá este manual de instalación.

El primer paso es descargar dicho fichero comprimido, yendo a la página del proyecto[19] y accediendo a "[Software]", en el menú de la página, para

[19] `http://math-atlas.sourceforge.net/`

obtener la última versión. A fecha de redacción, ésta es la 3.10.1 y se obtiene desde SourceForge[20].

Seguidamente han de averiguarse los detalles de procesador y frecuencia de los ordenadores de cada uno de los grupos en el clúster. Para ello ha de accederse a uno de los nodos de cada grupo y ejecutar los comandos cat /proc/cpuinfo y *uname -a*:

```
[root@compute-1-0 ~]# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 15
model           : 4
model name      :                Intel(R) Pentium(R) D CPU 3.20GHz
stepping        : 4
cpu MHz         : 2800.000
cache size      : 1024 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
apicid          : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 5
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8
 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi
 mmx fxsr sse sse2 ss ht tm syscall nx lm constant_tsc
 pni monitor ds_cpl est cid cx16 xtpr
bogomips        : 6400.13
clflush size    : 64
cache_alignment : 128
address sizes   : 36 bits physical, 48 bits virtual
power management:

processor       : 1
vendor_id       : GenuineIntel
[...]
```

Listing 2: Salida de cpuinfo para el nodo compute-1-0

Se ha omitido intencionadamente casi la mitad de la salida del comando, ya que al tratarse de un Pentium4 con HyperThreading, cada núcleo aparece como si fueran dos.

Otro comando a ejecutar para conocer la capacidad del ordenador es utilizar el comando lshw, que es una abreviatura de *list hardware*:

```
[root@compute-1-0 ~]# lshw
    description: Computer
    width: 64 bits
    capabilities: smbios-2.3 dmi-2.3 vsyscall64 vsyscall32
```

[20] http://sourceforge.net/projects/math-atlas/files/

```
   configuration: boot=normal uuid=59CF9F6A-DD82-11D9-97D9-00E018EB099A
 *-core
      description: Motherboard
      product: D915GUX
      vendor: Intel Corporation
      physical id: 0
      version: NNNNNNNN-NNN
      serial: LLLLNNNNNNNN
    *-firmware
         description: BIOS
         vendor: Intel Corp.
         physical id: 0
         version: EV91510A.86A.0444.2005.0429.2108
         date: 04/29/2005
         size: 64KiB
         capacity: 448KiB
         capabilities: pci pnp apm upgrade shadowing
 cdboot bootselect edd int13floppynec int13floppytoshiba
 int13floppy360 int13floppy1200 int13floppy720
 int13floppy2880 int5printscreen int9keyboard int14serial
 int17printer int10video acpi usb agp ls120boot zipboot
 biosbootspecification netboot
    *-cpu
         description: CPU
         product: Intel(R) Pentium(R) 4 CPU 3.20GHz
         vendor: Intel Corp.
         physical id: 4
         bus info: cpu@0
         version: Intel(R) Pentium(R) 4 processor
         serial: To Be Filled By O.E.M.
         slot: J2E1
         size: 3200MHz
         capacity: 3200MHz
         width: 64 bits
         clock: 200MHz
         capabilities: fpu fpu_exception wp vme de pse
 tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
 clflush dts acpi mmx fxsr sse sse2 ss ht tm syscall nx
 x86-64 constant_tsc pni monitor ds_cpl est cid cx16 xtpr
 cpufreq
[...]
```

Listing 3: Salida de uname para el nodo compute-1-0

Del comando anterior cat /proc/cpuinfo interesa tomar nota de la frecuencia de trabajo del procesador (*cpu MHz*, en este caso son 2800), el modelo (*model name*, en este caso *Pentium D*) y los flags (o *capabilities* en lshw), tales como anchura de registros (en este caso 64 bits), o las versiones de SSE que soportan (para este procesador, sse y sse2).

```
[root@compute-1-0 ~]# uname -a
Linux compute-1-0.local 2.6.18-308.4.1.el5 #1
SMP Tue Apr 17 17:08:00 EDT 2012 x86_64 x86_64
x86_64 GNU/Linux
```

Listing 4: Salida de uname para el nodo compute-1-0

A continuación, se descomprime el fichero descargado (`atlas3.10.1.tar.bz2`. Dado que se crearán tres versiones diferentes de BLAS mediante ATLAS, el fichero `ATLAS/` obtenido se moverá a la carpeta compartida `/share/apps/`, para que pueda ser ejecutado desde cada uno de los tres tipos de máquinas, y se crearán tres directorios de compilación diferentes[21]:

```
[root@moonstone ~]# tar jvxf atlas3.10.1.tar.bz2
[root@moonstone ~]# mv ATLAS/ /share/apps/
[root@moonstone ~]# mkdir -p /share/apps/ATLAS/Linux_P464SSE2/build/
[root@moonstone ~]# mkdir -p /share/apps/ATLAS/Linux_PD64SSE2/build/
[root@moonstone ~]# mkdir -p /share/apps/ATLAS/Linux_PX322064SSE2/build/
```

En nuestras pruebas utilizamos inicialmente la carpeta `/share/src/` para guardar los códigos fuente de ATLAS y `/share/apps/` para los ficheros ejecutables, pero descubrimos que estas carpetas creadas fuera de `apps/` desaparecían en el siguiente montaje.

Con esto el directorio con el código fuente quedará compartido entre todos los nodos y el frontend, y las carpetas de compilación para cada grupo estarán creadas.

Ahora ha de ejecutarse el script de configuración en cada grupo. Como el frontend es del mismo tipo que los nodos de cómputo, empezaremos por éste. Es importante tener en cuenta que ATLAS depende de que los ciclos de CPU sean constantes para obtener temporizaciones correctas, cosa que no sucede normalmente ya que la CPU cambia la frecuencia en función de la carga, ahorrando energía. Para obligar al ordenador a mantener la misma frecuencia siempre, necesitamos ejecutar, en el nodo desde el que se vaya a realizar la configuración y posterior compilación, el comando cpufreq-selector:

```
[root@moonstone ~]# cpufreq-selector -g performance
```

A continuación lanzamos el comando configure con los parámetros que ya conocemos, que son: la frecuencia de la CPU (2800 MHz), que pondremos usando los switches -D c -DPentiumCPS=2800; el tipo de arquitectura (64 bits), que se especifica con -b 64; el path donde queremos que se instale la aplicación (--prefix=/share/apps/ATLAS/Linux_P464SSE2/); pediremos que

[21] La instalación de ATLAS usa tres directorios: donde reside el código fuente, llamado **SRCdir** en la documentación de instalación (`atlas_install.pdf`), y que en nuestro caso es `/share/apps/ATLAS`; el directorio donde se ejecuta la compilación, llamado **BLDdir** y que en nuestro caso serán `/share/apps/ATLAS/Linux_P464SSE2/build/` para los nodos de cálculo normales, `/share/apps/ATLAS/Linux_PD64SSE2/build/` para los nodos con gráfica y `/share/apps/ATLAS/Linux_PX322064SSE2/build/` para los enracables; y por último el directorio de destino, donde se instalará ATLAS, que en nuestro caso son `/share/apps/ATLAS/Linux_P464SSE2`, `/share/apps/ATLAS/Linux_PD64SSE2` y `/share/apps/ATLAS/Linux_PX322064SSE2`, respectivamente.

además de generar las bibliotecas estáticas de ATLAS (archivos con extensión *.a*) genere las dinámicas, usando --shared y por último, ATLAS incluye una opción que permite compilarlo junto con LAPACK, si se le especifica la ruta donde reside el *.tgz* de instalación de LAPACK, que nosotros descargaremos en /root/, utilizando --with-netlib-lapack-tarfile=/root/lapack-3.5.0.tgz:

```
[root@moonstone ~]# cd
[root@moonstone ~]# wget http://www.netlib.org/lapack/lapack-3.5.0.tgz
[root@moonstone ~]# cd /share/apps/ATLAS/Linux_P464SSE2/build/
[root@moonstone build]# ../../configure -b 64 -D c -DPentiumCPS=2800 \
        --shared \
        --prefix=/share/apps/ATLAS/Linux_P464SSE2/ \
        --with-netlib-lapack-tarfile=/root/lapack-3.5.0.tgz
```

A continuación se pueden realizar ajustes más finos con la configuración mirando el resto de opciones que admite configure, que se obienen con la opción --help. No obstante, algunos de los parámetros de las opciones son códigos, que han de obtenerse con el comando ./xprint_enums:

```
[root@moonstone ~]# ../../configure --help
[root@moonstone ~]# make xprint_enums ; ./xprint_enums
```

El motivo por el que se ha ejecutado el configure con los parámetros de los que se disponía en primer lugar es que de esta forma se genera el Makefile necesario para compilar xprint_enums, a su vez necesario para conocer los códigos de las opciones del configure:

```
[root@moonstone build]# ./xprint_enums
ISA extensions are combined by adding their values together (bitvector):
```

No obstante se reproducen a continuación, para mayor comodidad, los códigos admitidos para el configure de la versión 3.10.1 de ATLAS:

Una vez ejecutado correctamente el configure, bastaría con realizar una limpieza del directorio de ejecución de configure con los parámetros apropiados:

```
[root@moonstone build]# ../../configure -b 64 -O 1 -D c -DPentiumCPS=2800 \
        --shared \
        --prefix=/share/apps/ATLAS/Linux_P464SSE2/ \
        --with-netlib-lapack-tarfile=/root/lapack-3.5.0.tgz
```

Y por último proceder a la compilación de los archivos de código fuente mediante el comando make:

```
[root@moonstone build]# make
```

Una vez finalizada la compilación de la librería, conviene efectuar una verificación para asegurarse de que la biblioteca ha sido correctamente compilada:

```
[root@moonstone build]# make check
```

0	UNKNOWN	16	P5MMX	33	AMD64K10h
1	POWER3	17	PPRO	34	AMDDOZER
2	POWER4	18	PII	35	UNKNOWNx86
3	POWER5	19	PIII	36	IA64Itan
4	PPCG46	20	PM	37	IA64Itan2
5	PPCG5	21	CoreSolo	38	USI
6	POWER6	22	CoreDuo	39	USII
7	POWER7	23	Core2Solo	40	USIII
8	IBMz9	24	Core2	41	USIV
9	IBMz10	25	Corei1	42	UST2
10	IBMz196	26	Corei2	43	UnknownUS
11	x86x87	27	Atom	44	MIPSR1xK
12	x86SSE1	28	P4	45	MIPSICE9
13	x86SSE2	29	P4E	46	ARMv7
14	x86SSE3	30	Efficeon		
15	P5	32	HAMMER		

Cuadro 2: Enumeración de arquitecturas admitidas en el configure, (**MACHTYPE**)

0	UNKNOWN	5	IRIX	10	HPUX
1	Linux	6	AIX	11	FreeBSD
2	SunOS	7	Win9x	12	OSX
3	SunOS4	8	WinNT		
4	OSF1	9	Win64		

Cuadro 3: Enumeración de sistemas operativos admitidos en el configure, (**OSTYPE**)

0	ICC	2	DMC	4	DKC	6	GCC
1	SMC	3	SKC	5	XCC	7	F77

Cuadro 4: Definiciones de compiladores admitidos en el configure

1	none	16	AVXFMA4	256	SSE1
2	VSX	32	AVX	512	3DNow
4	AltiVec	64	SSE3	1024	NEON
8	AVXMAC	128	SSE2		

Cuadro 5: Códigos de extensiones ISA admitidos en el configure

También resulta buena idea realizar una serie de benchmarks sobre la biblioteca para conocer su rendimiento en términos de tiempo. El comando make time permite obtener los resultados de varios de ellos y mostrarlos organizados en función de la precisión (simple o doble precisión) y del tipo

de datos (real o complejo):

```
[root@moonstone build]# make time
```

3.5. Intel Math Kernel Library (MKL) e Intel Parallel Studio

Dado que todos los ordenadores del cluster son de arquitecturas de Intel, se puede instalar, en lugar de ATLAS o BLAS, la MKL (*Math Kernel Library*), que dispone de funciones de cálculo matemático completamente optimizadas para este tipo de procesadores, e incluye interfaces que permiten portar programas escritos para bibliotecas, como FFTW, a MKL.

3.5.1. Instalación de Intel Math Kernel Library

La licencia para esta biblioteca tiene un coste de 500 USD a fecha de redacción. Sin embargo, dado que el propósito de la práctica no es lucrativo, Intel permite la descarga de forma gratuita aceptando un acuerdo de licencia de uso no comercial del software. Para ello, ha de accederse a su página de internet [22], seleccionar el software, introducir algunos datos personales (nombre y apellido del usuario, correo electrónico, tipo de usuario (*Individual*), etc, ..), la página se redireccionará automátcamente y mostrará un número de serie con el que poder instalar el programa. Debajo del número de licencia, aparecerá un enlace para proceder a la descarga. Antes de pulsar, conviene copiar o apuntar el número de serie, aunque la página enviará un correo electrónico a la cuenta especificada con la clave generada. A fecha de escritura de este informe, la última versión del fichero es la 11.1.0.080, el nombre del fichero es `l_mkl_11.1.0.080.tgz` y ocupa 646,1 MiB[23].

No obstante, resulta más interesante la descarga del paquete Intel Parallel Studio, ya que incluye varias bibliotecas, entre las que se encuentran MKL, IPP (*Intel Performance Primitives*, que contienen conjuntos de filtros 1D, funciones estadísticas y bibliotecas para procesamiento gráfico), los compiladores de Intel de C, C++ y Fortran, y varios programas, bibliotecas y utilidades más. En nuestro caso, descargamos e instalamos la biblioteca MKL, pero posteriormente la desinstalamos e instalamos Parallel Studio, al contener el compilador de Intel y MKL.

El proceso de instalación de MKL, una vez descargado, es el siguiente:

```
[root@moonstone ~]# tar xzvf l_mkl_11.1.0.080.tgz
[root@moonstone ~]# cd l_mkl_11.1.0.080/
[root@moonstone ~]# ./install_GUI.sh
```

[22] `https://software.intel.com/es-es/non-commercial-software-development`

[23] Mebibyte, ver `http://es.wikipedia.org/wiki/Mebibyte`.

Con el último comando se iniciará el asistente gráfico que facilitará el proceso de instalación. En la primera pantalla, la de bienvenida, ha de pulsarse "Next"; la siguiente pantalla informa de que el sistema operativo no está soportado, aunque CentOS, que es en el que está basado Rocks, es en realidad un fork obtenido al compilar el código fuente de Red Hat Enterprise Linux. En la siguiente, se pregunta si el usuario acepta la licencia, a lo que ha de marcarse la opción superior para confirmarlo (*I accept the terms of the license agreement*). La siguiente pregunta si se desea activar el producto software mediante una licencia, activarla para evaluación, o mediante método alternativo; ha de elegirse mediante una licencia, que es la que viene activada por defecto e insertar en la casilla el número de serie que la página de web de Intel proporcionó anteriormente, bien directamente o a través de la dirección de correo electrónico especificada.

La siguiente pantalla comenta si se desean enviar datos de uso del producto de forma anónima; en nuestro caso concreto se ha seleccionado No. Al pulsar "Next" aparece un informe con la configuración de la instalación; por defecto el directorio para las bibliotecas es `/opt/intel` y ha de cambiarse al directorio con el que nuestro cluster comparte los ficheros a traves de la red, es decir `/share/apps`. Para mayor claridad, se elegirá que se instale en `/share/apps/intel`. Para cambiar la configuración de instalación por defecto ha de pulsarse sobre el botón "Customize installation":

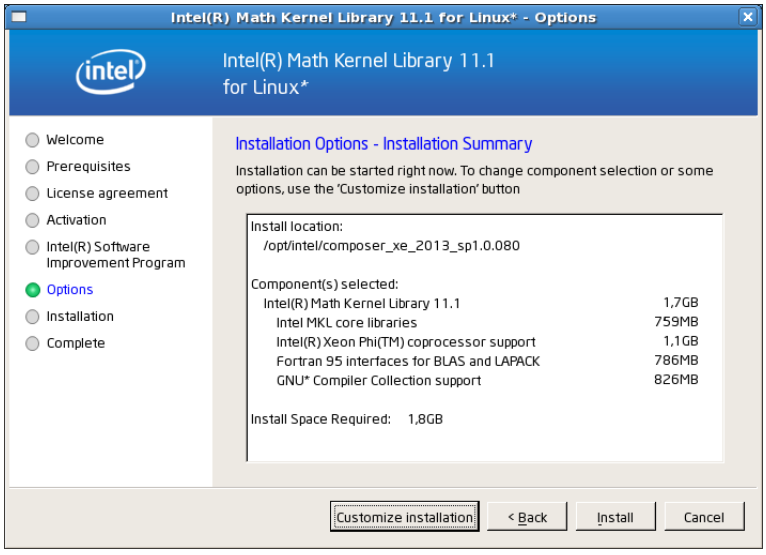

Figura 18: Informe de configuración de preinstalación (paso 6)

La pantalla que aparece a continuación muestra la ruta de instalación, donde ha de cambiarse, como se ha comentado antes, el `/opt/intel` del cuadro de texto "Destination folder" por `/share/apps/intel` y pulsar sobre el botón "Next >", como se puede ver en la Figura 19.

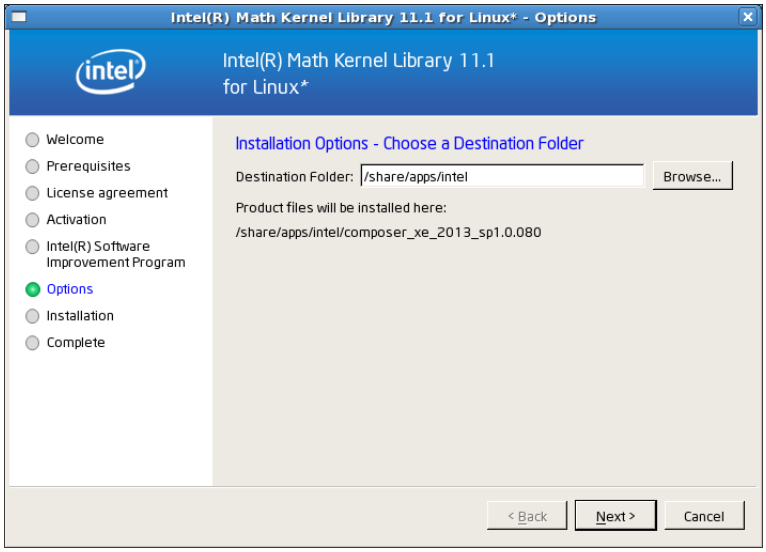

Figura 19: Configuración de la ruta de instalación de la biblioteca Intel MKL (paso 8)

Seguidamente, el programa pregunta el tipo de aplicacion que se va a generar usando la biblioteca MKL (32 bits, 64 bits o ambas). Como todas las máquinas del cluster soportan x86_64, y los procesadores de 64 bits pueden ejecutar aplicaciones pensadas para 32, se dejan las dos opciones marcadas (tal y como aparecen por defecto) y se pulsa sobre el botón "Next >" para continuar la instalación, tal y como aparece en la Figura 20.

La siguiente pantalla que aparece (mostrada en la Figura 21) sirve para seleccionar los módulos a instalar. Por defecto, el programa dejará las tres últimas sin marcar: soporte para compiladores PGI[24], que en en caso actual dejaremos desmarcada, ya que para nuestro caso particular no tenemos estos compiladores instalados; interfaz SP2DP, que permite utilizar programas escritos en Fortran al estilo Cray, con los BLAS y LAPACK de MKL, mapeando nombres de precisión simple a precisión doble, que ha de marcarse por si en un futuro se instalan y compilan aplicaciones de este tipo; y soporte para clústeres, que también se marca.

Al pulsar sobre "Next >" se pasa a la siguiente pantalla, que muestra de nuevo el resumen de la instalación, donde ha de observarse si el directorio de instalación aparece con la modificación que se hizo anteriormente (`/share/apps/intel/composer_xe_2013_sp1.1.0.080`, para la versión de MKL cuya instalación se presenta en este documento), y si entre los com-

[24]PGI son las siglas de Portland Group Inc., una compañía subsidiaria de NVIDIA, que produce compiladores de Fortran, C, C++ con y sin soporte para CUDA, diseñados para clusters. Ver `http://www.pgroup.com/` y `http://en.wikipedia.org/wiki/The_Portland_Group`

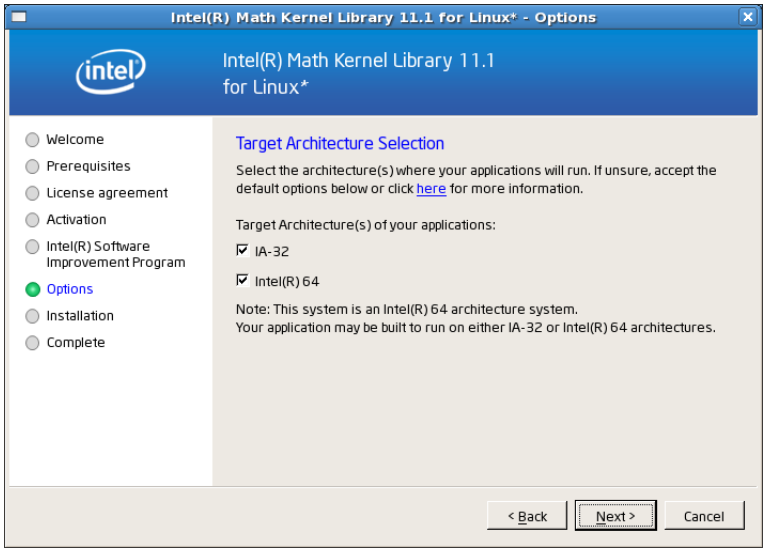

Figura 20: Selección de los tipos de aplicaciones compilables con Intel MKL (paso 9)

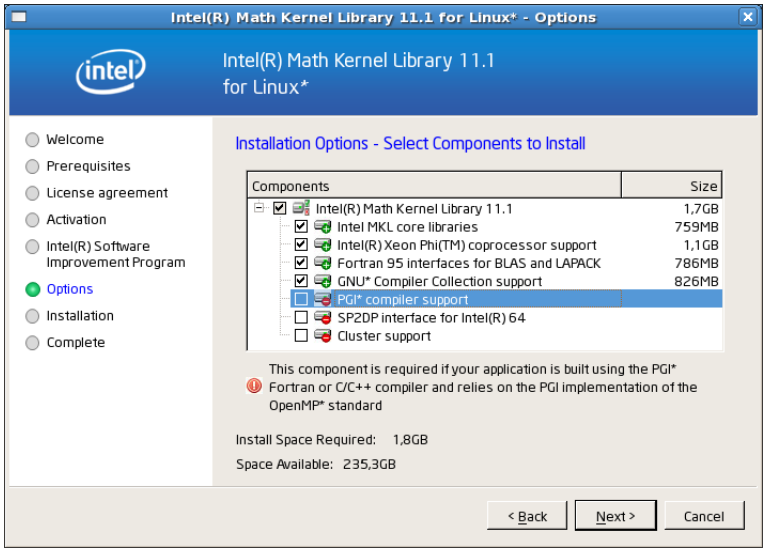

Figura 21: Pantalla de selección de componentes en el instalador gráfico de Intel MKL (paso 10)

ponentes seleccionados aparecen *Cluster support* y *SP2DP interface for Intel(R)64*, que se seleccionaron en la pantalla anterior. En este punto, se presiona sobre el botón "Instalar" y la instalación comienza.

Al cabo de un rato la instalación termina y se pulsa sobre el botón "Finish". Se abrirá una ventana en el navegador de internet con información

sobre la instalación y los componentes seleccionados.

Sin embargo, aunque la instalación haya terminado, aún hay una serie de pasos que realizar para que los componentes sean utilizables, entre los cuales se encuentran las interfaces, como luego veremos.[25].

3.5.2. Desinstalación de Intel Math Kernel Library

Como ya se ha comentado anteriormente, en vez de usar únicamente las bibliotecas MKL de Intel, la misma página de descarga que se utilizó anteriormente para obtenerlo permite descargar la suite *Intel Parallel Studio*, que además proporciona el compilador de Intel para C, C++ entre otras cosas, lo cual puede resultar útil. En caso de que se haya instalado MKL previamente, estas bibliotecas se pueden eliminar fácilmente (y utilzando interfaz gráfica, como en la instalación) utilizando el comando uninstall_GUI.sh:

```
[root@moonstone ~]# cd /share/apps/intel/composer_xe_2013_sp1.0.080
[root@moonstone composer_xe_2013_sp1.0.080]# ./uninstall_GUI.sh
```

3.5.3. Instalación de Intel Parallel Studio

En este paso se mostrará la forma de descargar e instalar Parallel Studio. A fecha de escritura de este documento, la última versión disponible (que es la que se descarga por defecto) es la 2013, Service Pack 1, actualización 2. El nombre del fichero descargado es `parallel_studio_xe_2013_sp1_update2.tgz` y tiene un tamaño de 3,2 Gibibytes. Una vez en el disco duro, se procede a descomprimirlo y lanzar la instalación como en el caso anterior:

```
[root@moonstone ~]# mv parallel_studio_xe_2013_sp1_update2.tgz /share/apps/
[root@moonstone ~]# cd /share/apps/
[root@moonstone ~]# tar xzvf parallel_studio_xe_2013_sp1_update2.tgz
[root@moonstone ~]# cd parallel_studio_xe_2013_sp1_update2
[root@moonstone parallel_studio_xe_2013_sp1_update2]# ./install_GUI.sh
```

Los pasos de instalación son análogos a los mostrados con MKL, y de la misma forma, ha de cambiarse el directorio de instalación que viene por defecto, `/opt/intel` por `/share/apps/intel` en la pantalla *Choose a destination folder*, como se pudo ver en la Figura 19. En la pantalla de selección de componentes, hemos marcado todos los componentes, salvo los últimos 3, que son *GDB Eclipse Integration on Intel(R) 64*, *Source of GNU* GDB*

[25]En este ámbito, una *interfaz* es una librería o código que se compila para permitir que una aplicación escrita para ser utilizada contra una biblioteca concreta, pueda compilarse contra otra. Un ejemplo puede ser una aplicación a la que llamaremos "APP" pensada para usar las funciones de FFTW. Si la arquitectura de los procesadores sobre los cuales va a ejecutarse la aplicación es tipo Intel y la biblioteca MKL está instalada, pueden utilizarse las funciones de esta última en su lugar. Para evitar cambios al código original de APP, se utiliza una interfaz, que muestra los nombres de las funciones de FFTW y las traduce a las de MKL. Es equivalente al patrón de diseño softare *Adaptador*.

y *Source of GDB Eclipse* Integration*, ya que en nuestro caso concreto no disponemos de Eclipse instalado, aunque pueden marcarse si el lector le ha instalado en el frontend. Con esta configuración de componentes a instalar (es decir, sin incluír los últimos tres), el tamaño total que se ocupará son 9,2 Gibibytes.

En la pantalla que permite la configuración de Intel VTune Amplifier XE pulsando el botón "Customize", nosotros hemos decidido no configurarle, ya que por defecto se instala en el directorio que se especificó y no en el que la aplicación tiene por defecto y pasar a la siguiente pantalla pulsando sobre el botón "Next >". Se volverá a mostrar la configuración de instalación, con la ruta cambiada y los módulos seleccionados, y al pulsar de nuevo "Next >", comenzará la instalación. Cuando ésta termine y al pulsar el botón "Finish", igual que en el caso del instalador de MKL, el asistente se cerrará y se abrirá una nueva pestaña del navegador de internet mostrando algunos detalles sobre los componentes instalados. El archivo html principal de la página que se abre se puede encontrar en `/share/apps/intel/parallel_studio_xe_2013/Documentation/en_US/welcomepage_studio_xe/get_started.html`.

3.5.4. Configuración de las variables de entorno

Sin embargo, aunque la instalación haya terminado correctamente, esto no significa que la configuración esté terminada. Por ejemplo, en el caso de Parallel Studio, el compilador de intel para C, que se ejecuta con el comando icc, está incluído en la instalación, pero si ejecutamos en un terminal el comando, aparecerá un error indicando que no está disponible.

```
[root@moonstone ~]# icc -help
bash: icc: command not found
```

Esto se debe a que el binario ha sido instalado en `/share/apps/intel/composer_xe_2013_sp1.2.144/bin/intel64/` y el instalador no actualiza las variables de entorno necesarias (en este caso el *PATH*). Sin embargo, existe un script de BASH llamado compilervars.sh, que se encuentra en `/share/apps/intel/bin/` que se encarga de ello. El problema con este script es que las variables de entorno sólo valen para la sesión actual con el terminal; si se abre otro ya no estarán disponibles. Para ello, ha de llamarse al script cada vez que se inicie un terminal nuevo.

Como resulta un poco molesto tener que ejecutar compilervars.sh a cada nuevo terminal, se puede configurar BASH para que, a cada sesión que se abra, se lance este comando, y esto se puede hacer para todos los usuarios o individualmente, para cada uno. Por ejemplo, ya que en el frontend se entra directamente desde el usuario root, se puede hacer que éste tenga las variables de entorno preconfiguradas a cada inicio de terminal, editando el fichero `.bashrc` que se puede encontrar en la $HOME de cada usuario:

```
[root@moonstone ~]# cd
[root@moonstone ~]# vim .bashrc
```

Al final del todo del documento, simplemente se añade la siguiente línea:

```
source /share/apps/intel/bin/compilervars.sh intel64
```

Y con eso se dispondrá de las variables de entorno perfectamente configuradas a cada inicio del terminal. Para verificarlo, se puede llamar desde BASH al compilador de intel, como antes:

```
[root@moonstone ~]# icc
icc: command line error: no files specified; for help type "icc -help"
```

3.5.5. Compilación de las bibliotecas

Como ya se ha comentado anteriormente, las bibliotecas disponen de una serie de interfaces o *wrappers* que permiten que aquellos programas que dependan de librerías tales como FFTW puedan ser utilizados con las proporcionadas por Intel Parallel Studio (en este caso las interfaces de FFTW se incluyen dentro de las Math Kernel Libraries).

Dado que posteriormente será necesario utilizar la interfaz fftw2xc para LAMMPS, es necesario compilarla. Para ello basta con acceder al directorio donde se encuentran los códigos de las diferentes interfaces (`/share/apps/intel/mkl/interfaces/`), entrar en la carpeta de la intefaz que se desea compilar (como por ejemplo `fftw2xc` o `fftw3xf`), y ejecutar make libintel64 si se desea compilar para 64 bits o make libia32 si en su lugar se prefiere la versión de 32 bits. En nuestro caso, la versión necesaria es la de 64 bits debido a las características del procesador y del sistema operativo:

```
[root@moonstone ~]# cd /share/apps/intel/mkl/interfaces/
[root@moonstone interfaces]# cd fftw2xc/
[root@moonstone fftw2xc]# make libintel64
```

Ejecutando únicamente el comando *make*, aparece un listado de todas las opciones que se pueden pasar al Makefile para la compilación:

```
[root@moonstone fftw3xc]# make
Usage: make libia32|libintel64|libmic [option...]

Options:
  compiler=gnu|pgi|intel
        Build the library using GNU gcc, PGI pgcc, or
        Intel(R) C compiler icc.
        Default value: intel

Additional macros:
  MKLROOT=<path>
        Path to MKL root directory with header files and libraries.
        Default value: ../..
```

```
INSTALL_DIR=<path>
      Install the library to the specified location.
      Default value:  . (that is, the current directory)

INSTALL_LIBNAME=<name>
      Specify the name of the library.
      Default value depends on PRECISION and compiler used:
        libfftw3xc_intel.a
```

Una vez finalizada la compilación de la interfaz, ésta se mueve como biblioteca al directorio `/share/apps/intel/mkl/lib/intel64` (ya que ha sido compilada para 64 bits):

```
[root@moonstone ~]# ls /share/apps/intel/mkl/lib/intel64
libfftw2xc_double_intel.a        libmkl_intel_lp64.so
libmkl_avx2.so                   libmkl_intel_sp2dp.a
libmkl_avx512_mic.so             libmkl_intel_sp2dp.so
libmkl_avx.so                    libmkl_intel_thread.a
libmkl_blacs_ilp64.a             libmkl_intel_thread.so
libmkl_blacs_intelmpi_ilp64.a    libmkl_lapack95_ilp64.a
libmkl_blacs_intelmpi_ilp64.so   libmkl_lapack95_lp64.a
libmkl_blacs_intelmpi_lp64.a     libmkl_mc3.so
libmkl_blacs_intelmpi_lp64.so    libmkl_mc.so
libmkl_blacs_lp64.a              libmkl_p4n.so
libmkl_blacs_openmpi_ilp64.a     libmkl_pgi_thread.a
libmkl_blacs_openmpi_lp64.a      libmkl_pgi_thread.so
libmkl_blacs_sgimpt_ilp64.a      libmkl_rt.so
libmkl_blacs_sgimpt_lp64.a       libmkl_scalapack_ilp64.a
libmkl_blas95_ilp64.a            libmkl_scalapack_ilp64.so
libmkl_blas95_lp64.a             libmkl_scalapack_lp64.a
libmkl_cdft_core.a               libmkl_scalapack_lp64.so
libmkl_cdft_core.so              libmkl_sequential.a
libmkl_core.a                    libmkl_sequential.so
libmkl_core.so                   libmkl_vml_avx2.so
libmkl_def.so                    libmkl_vml_avx512_mic.so
libmkl_gf_ilp64.a                libmkl_vml_avx.so
libmkl_gf_ilp64.so               libmkl_vml_cmpt.so
libmkl_gf_lp64.a                 libmkl_vml_def.so
libmkl_gf_lp64.so                libmkl_vml_mc2.so
libmkl_gnu_thread.a              libmkl_vml_mc3.so
libmkl_gnu_thread.so             libmkl_vml_mc.so
libmkl_intel_ilp64.a             libmkl_vml_p4n.so
libmkl_intel_ilp64.so            locale
libmkl_intel_lp64.a
```

3.6. Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)

El simulador atómico-molecular masivamente paralelo a gran escala (*Large-scale Atomic/Molecular Massively Parallel Simulator*, *LAMMPS*, en inglés)

es software de simulación de dinámica molecular, desarrollado por el laboratorio nacional estadounidense Sandia.

Para su instalación, es preciso descargar primero la última versión de LAMMPS disponible desde su página web oficial

```
[root@moonstone ~]# mkdir /share/apps/LAMMPS/
[root@moonstone ~]# tar xzvf lammps_stable.tar.gz
[root@moonstone ~]# mv lammps-1Feb14/* /share/apps/LAMMPS/
[root@moonstone ~]# cd /share/apps/LAMMPS/src/
[root@moonstone src]# make
  932  ls MAKE
  933  head MAKE/Makefile.linux
[root@moonstone src]# cd MAKE/
[root@moonstone ~]# cp Makefile.mkl Makefile.hpcInf
[root@moonstone ~]# vim Makefile.hpcInf
[root@moonstone ~]# make
```

Dado que el fichero que se utilizará (`Makefile.hpcInf`) ha sido obtenido copiando el existente que emplea la biblioteca MKL de Intel, y este, a su vez, contiene las rutas de la instalación estándar de MKL para una versión concreta, hay que editarle para cambiar dichas rutas a las que se emplean en nuestro caso. Dicho de otro modo, en el interior del fichero `Makefile.mkl` todas las bibliotecas están en `/opt/intel` en vez de en `/share/apps/intel`, que es donde han de ser instaladas para que Rocks las comparta entre todo el cluster. Además, aunque se utilizan las últimas versiones tanto de LAMMPS como de MKL, las rutas del fichero `Makefile.mkl` de LAMMPS están referidas a las de una versión anterior de MKL, con lo que ha de buscarse la ubicación real del fichero reemplazarla en el Makefile.

em64t es ahora intel64

3.6.1. Obtención del sistema MPI instalado en el sistema

El compilador mpiicc (Intel MPI) viene incluído, a fecha de redacción del presente documento, únicamente en el paquete de software Intel Cluster Studio, que no está disponible para descarga gratuita. Sin embargo, se puede ver si existe algun compilador MPI ya instalado y disponible en el sistema usando la función autocompletar de BASH. Para ello, basta con escribir en un terminal de BASH *mpi* y sin pulsar intro, pulsar la tecla tabulador del teclado:

```
[root@moonstone ~]# mpi
mpic++              mpicxx              mpif90
mpicc               mpicxx-vt           mpif90-vt
mpiCC               mpiexec             mpirun
mpicc-vt            mpiexec.hydra       mpi-selector
mpiCC-vt            mpiexec.py          mpi-selector-menu
mpicleanup          mpif77
mpic++-vt           mpif77-vt
```

Donde mpicxx indica que, probablemente, en el sistema está instalado OpenMPI. No obstante, existen una serie de pasos para verificar esto. El primero de ellos es el comando which, que muestra el directorio desde el que se ejecuta un comando:

```
[root@moonstone intel64]# which mpicxx
/opt/openmpi/bin/mpicxx
```

Lo que confirma que OpenMPI está instalado en el sistema y por tanto puede ser usado para la compilación de LAMMPS. Otros comandos que pueden ayudar a determinar el software de MPI[26] instalado son mpiexec --version, mpicc -v y ompi_info:

```
[root@moonstone intel64]# mpicc -v
Usando especificaciones internas.
Objetivo: x86_64-redhat-linux
Configurado con: ../configure --prefix=/usr
--mandir=/usr/share/man --infodir=/usr/share/info
--enable-shared --enable-threads=posix
--enable-checking=release --with-system-zlib
--enable-__cxa_atexit --disable-libunwind-exceptions
--enable-libgcj-multifile
--enable-languages=c,c++,objc,obj-c++,java,fortran,ada
--enable-java-awt=gtk --disable-dssi --disable-plugin
--with-java-home=/usr/lib/jvm/java-1.4.2-gcj-1.4.2.0/jre
--with-cpu=generic --host=x86_64-redhat-linux
Modelo de hilos: posix
gcc version 4.1.2 20080704 (Red Hat 4.1.2-52)
[root@moonstone intel64]# ompi_info
                Package: Open MPI root@centos5-64bit.localdomain
Distribution
               Open MPI: 1.4.3
  Open MPI SVN revision: r23834
  Open MPI release date: Oct 05, 2010
               Open RTE: 1.4.3
  Open RTE SVN revision: r23834
  Open RTE release date: Oct 05, 2010
                   OPAL: 1.4.3
      OPAL SVN revision: r23834
      OPAL release date: Oct 05, 2010
           Ident string: 1.4.3
                 Prefix: /opt/openmpi
[...]
```

En este caso, mpicc -v no proporciona mucha información, pero ya que gracias al which que se hizo anteriormente se sabe que OpenMPI está instalado, ompi_info permite verificarlo y además añade información sobre la versión. Respecto al comando mpiexec --version, en caso de que se haya instalado el paquete de software Intel Parallel Studio XE, mpiexec ha de

[26] MPI, (*Message Passing Interface*, del inglés Interfaz de paso de mensajes). Ver `http://es.wikipedia.org/wiki/Interfaz_de_Paso_de_Mensajes`.

ejecutarse comentando la línea que se añadió previamente al archivo .bashrc de *root* (como se vió en la sección 3.5.4), guardando los cambios en el editor de texto que se haya usado, lanzando una nueva ventana del terminal o una pestaña (en GNOME, usando [Ctrl]+[⇑]+[T]), ejecutando en ella comando mpiexec --version, viendo el resultado, descomentando la línea añadida al .bashrc y guardando los cambios en este fichero. Esto es así porque Intel MPI sólo se incluye, a fecha de redacción de este documento, con Intel Cluster Studio, y no con Parallel Studio; sin embargo, si se ejecuta el comando sin descomentar la línea, aparecerá lo siguiente:

```
[root@moonstone intel64]# mpiexec --version
Intel(R) MPI Library for Linux* OS, 64-bit applications, Version 4.1.3
Build 20131205
Copyright (C) 2003-2013 Intel Corporation.  All rights reserved.
```

Se puede verificar que la biblioteca MPI no está instalada ejecutando el comando mpiicc en un terminal nuevo tras verificar que la línea esté descomentada, para que las variables de entorno del paquete de software de intel estén aplicados:

```
[root@moonstone intel64]# mpiicc
-bash: mpiicc: command not found
```

Con lo que, con la línea descomentada e Intel Parallel Studio instalado, mpiexec --version devuelve un resultado erróneo. No obstante, con el procedimiento anterior de comentar la línea, guardar, abrir nuevo terminal, ejecutar el comando, cerrar el terminal, descomentar la línea y guardar, el resultedo en la nueva pestaña/ventana del terminal aparece correctamente:

```
[root@moonstone ~]# mpiexec --version
mpiexec (OpenRTE) 1.4.3

Report bugs to http://www.open-mpi.org/community/help/
```

3.6.2. Configuración del Makefile de LAMMPS

```
# hpcInf = Intel Cluster Tools, mpicxx, MKL MPI, MKL FFT

# Intel recommends Intel Cluster Tools Compiler Edition
# to build libfftw2xc_intel.a:
# > cd /opt/intel/mkl/10.0.011/interfaces/fftw2xc
# > become root via su
# > gmake libem64t

SHELL = /bin/sh

# ---------------------------------------------------------------------
# compiler/linker settings
# specify flags and libraries needed for your compiler
```

```
C =             mpicxx
CCFLAGS =       -O3 -unroll0
SHFLAGS =       -fPIC
DEPFLAGS =      -M

LINK =          mpicxx
LINKFLAGS =     -O -L/share/apps/intel/mkl/lib/intel64 -L/usr/lib/
gcc/x86_64-redhat-linux/4.1.1/ -I/share/apps/intel/composer_xe_2013_sp1.2.144/
mkl/lib/intel64/ -I/share/apps/intel/composer_xe_2013_sp1.2.144/mkl/include/
LIB =           -Wl,--no-as-needed -lstdc++ -lgomp -lmkl_core -lmkl_gf_ilp64
-lmkl_gnu_thread -lmkl_scalapack_ilp64 -lmkl_cdft_core -lmkl_intel_ilp64
-lmkl_blacs_intelmpi_ilp64 -ldl -lpthread -lm
SIZE =          size

ARCHIVE =       ar
ARFLAGS =       -rc
SHLIBFLAGS =    -shared

# ---------------------------------------------------------------------
# LAMMPS-specific settings
# specify settings for LAMMPS features you will use
# if you change any -D setting, do full re-compile after "make clean"
# LAMMPS ifdef settings, OPTIONAL
# see possible settings in doc/Section_start.html#2_2 (step 4)

LMP_INC =       -DLAMMPS_GZIP

# MPI library, REQUIRED
# see discussion in doc/Section_start.html#2_2 (step 5)
# can point to dummy MPI library in src/STUBS as in Makefile.serial
# INC = path for mpi.h, MPI compiler settings
# PATH = path for MPI library
# LIB = name of MPI library

MPI_INC =       -I/opt/openmpi/include/
MPI_PATH =      -L/opt/openmpi/lib
MPI_LIB =       -lmpi

# FFT library, OPTIONAL
# see discussion in doc/Section_start.html#2_2 (step 6)
# can be left blank to use provided KISS FFT library
# INC = -DFFT setting, e.g. -DFFT_FFTW, FFT compiler settings
# PATH = path for FFT library
# LIB = name of FFT library

FFT_INC =       -DFFT_FFTW -I/share/apps/intel/mkl/include/fftw/
FFT_PATH =
FFT_LIB =       /share/apps/intel/composer_xe_2013_sp1.0.080/mkl/
lib/intel64/libfftw2xc_double_gnu.a

# JPEG and/or PNG library, OPTIONAL
# see discussion in doc/Section_start.html#2_2 (step 7)
# only needed if -DLAMMPS_JPEG or -DLAMMPS_PNG listed with LMP_INC
# INC = path(s) for jpeglib.h and/or png.h
```

```
# PATH = path(s) for JPEG library and/or PNG library
# LIB = name(s) of JPEG library and/or PNG library

JPG_INC =
JPG_PATH =
JPG_LIB =

# ---------------------------------------------------------------------
# build rules and dependencies
# no need to edit this section

include Makefile.package.settings
include Makefile.package

EXTRA_INC = $(LMP_INC) $(PKG_INC) $(MPI_INC) $(FFT_INC) $(JPG_INC) $(PKG_SYSINC)
EXTRA_PATH = $(PKG_PATH) $(MPI_PATH) $(FFT_PATH) $(JPG_PATH) $(PKG_SYSPATH)
EXTRA_LIB = $(PKG_LIB) $(MPI_LIB) $(FFT_LIB) $(JPG_LIB) $(PKG_SYSLIB)

# Path to src files

vpath %.cpp ..
vpath %.h ..

# Link target

$(EXE): $(OBJ)
        $(LINK) $(LINKFLAGS) $(EXTRA_PATH) $(OBJ) $(EXTRA_LIB) $(LIB) -o $(EXE)
        $(SIZE) $(EXE)

# Library targets

lib:    $(OBJ)
        $(ARCHIVE) $(ARFLAGS) $(EXE) $(OBJ)

shlib:  $(OBJ)
        $(CC) $(CCFLAGS) $(SHFLAGS) $(SHLIBFLAGS) $(EXTRA_PATH) -o $(EXE) \
        $(OBJ) $(EXTRA_LIB) $(LIB)

# Compilation rules

%.o:%.cpp
        $(CC) $(CCFLAGS) $(SHFLAGS) $(EXTRA_INC) -c $<

%.d:%.cpp
        $(CC) $(CCFLAGS) $(EXTRA_INC) $(DEPFLAGS) $< > $@

# Individual dependencies

DEPENDS = $(OBJ:.o=.d)
sinclude $(DEPENDS)
```

3.6.3. Creación del usuario para aplicaciones

A continuación, ha de crearse uno o varios usuarios para ejecutar las aplicaciones que se lanzarán en paralelo. El motivo de crear un usuario para ejecutar las aplicaciones en el clúster es que FALTA. Para ello, en primer lugar, ha de teclearse el comando adduser, que sirve para crear un usuario; por ejemplo, se le puede llamar *usuhpc*:

```
[root@moonstone ~]# useradd usuhpc
```

Una vez hecho, debe crearse la contraseña del nuevo usuario del clúster con el comando passwd:

```
[root@moonstone ~]# passwd usuhpc
Changing password for user usuhpc.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Y por último, han de sincronizarse los usuarios entre los nodos para que se creen las cuentas de éstos en los nodos, y no únicamente en el frontend. Una vez terminado el proceso de sincronización, resulta completamente recomendable reiniciar todos los nodos para asegurarse de que los cambios en los usuarios se han exportado correctamente:

```
[root@moonstone ~]# rocks sync users
[root@moonstone ~]# rocks run host reboot
```

Lo de cerrar sesion e iniciar con el nombre del nuevo usuario para que salga aquello

en el .bashrc del usuario

3.6.4. Lanzamiento de aplicaciones en paralelo

Para la presente práctica, se lanzará una simulación del movimiento de una serie de partículas utilizando LAMMPS en cada uno de los grupos de nodos comentados inicialmente (normales, con gráfica y enracables). Dado que será necesario escribir tres ficheros de configuración, uno para lanzar la aplicación a cada grupo de nodos de cómputo, se empezará creando una carpeta para el lanzamiento:

```
[usuhpc@moonstone ~]# cd ~
[usuhpc@moonstone ~]# mkdir lammps_test_norm
[usuhpc@moonstone ~]# mkdir lammps_test_graph
[usuhpc@moonstone ~]# mkdir lammps_test_xeon
```

Luego ha de descomprimirse el fichero `hpcLAMMPSRun.zip`[27], que contiene todos los archivos necesarios para ejecutar una simulación en LAMMPS,

[27] Este fichero ha sido proporcionado por el profesor de la asignatura.

y copiar el contenido de la carpeta descomprimida a cada una de las tres carpetas anteriores:

```
[usuhpc@moonstone ~]# unzip hpcLAMMPSRun.zip
Archive:  hpcLAMMPSRun.zip
   creating: hpcLAMMPSRun/
 extracting: hpcLAMMPSRun/Si.tersoff3
 extracting: hpcLAMMPSRun/in.Si.est
 extracting: hpcLAMMPSRun/machines
 extracting: hpcLAMMPSRun/posini_lammps.c
 extracting: hpcLAMMPSRun/posini_lammps.x
 extracting: hpcLAMMPSRun/red.lammps
 extracting: hpcLAMMPSRun/runjob.sh
```

Es necesario utilizar LAMMPS para ejecutar esta simulación, con lo que existen varias opciones respecto a la ubicación de ejecución del programa. Una de ellas consiste en copiar el ejecutable binario a la carpeta resultado de la descompresión de `hpcLAMMPSRun.zip`, mientras que otra podría ser incluir el directorio de compilación de LAMMPS a la variable de entorno *PATH*. En el caso presente, se ha optado por la primera opción:

```
[usuhpc@moonstone ~]# cp /share/apps/LAMMPS/src/lmp_hpcInf hpcLAMMPSRun/
```

Y de esta forma, copiar todo el contenido de la carpeta resultante de la descompresión (que ahora contendrá además el ejecutable de LAMMPS) a cada una de las carpetas que se crearon anteriormente para realizar las simulaciones en cada uno de los grupos de nodos de cómputo:

```
[usuhpc@moonstone ~]# cp hpcLAMMPSRun/* lammps_test_norm/
[usuhpc@moonstone ~]# cp hpcLAMMPSRun/* lammps_test_graph/
[usuhpc@moonstone ~]# cp hpcLAMMPSRun/* lammps_test_xeon/
```

A continuación, en cada una de ellas existe un fichero que ha de configurarse con los nodos de cómputo en los que va a ejecutarse la simulación, la variante de MPI que se utilizará para el paso de mensajes y algunas otras más. Este fichero se llama `runjob.sh`, y debe modificarse para cada uno de los tres casos anteriores:

```
[usuhpc@moonstone ~]# cd lammps_test_norm/
[usuhpc@moonstone lammps_test_norm]# vim runjob.sh
```

El contenido del fichero `runjob.sh` tiene un aspecto similar al mostrado a continuación:

```
#!/bin/bash
#$-cwd
#$-o /home/ushpc/lammps_test/ -j y
#$-N lammps.test_x1
#$-S /bin/bash
#$-pe orte 4
#$-q all.q@compute-2-0.local
#$-M someones_mail@hotmail.com
```

```
#$-m eas
#$-V

cd /home/ushpc/lammps_test/
mpirun -np 4 ./lmp_hpcInf < in.Si.est
```

Ruta de ubicación del fichero `runjob.sh` En él hay que modificar una serie de cosas. Para empezar, debe obtenerse la ruta absoulta donde se ubica el fichero y reemplazar las líneas *#$-o /home/ushpc/lammps_test/ -j y* y *cd /home/ushpc/lammps_test/* con el path de la carpeta desde el que se va a ejecutar el fichero `runjob.sh`. Ya que se ha empezado por la carpeta `lammps_test_norm`, basta con cerrar vim pulsando la tecla [Esc] del teclado y escribiendo *:wq*, y utilizar el comando pwd de Linux para obtener el directorio actual:

```
[usuhpc@moonstone lammps_test_norm]$ pwd
/home/usuhpc/lammps_test_norm
```

Con lo cual, *#$-o /home/ushpc/lammps_test/ -j y* se convierte en *#$-o /home/usuhpc/lammps_test_norm/ -j y* y *cd /home/ushpc/lammps_test/* queda como *cd /home/usuhpc/lammps_test_norm/*.

Correo electrónico para avisos La línea *#$-M someones_mail@hotmail.com* se utiliza para especificar una dirección de correo electrónico a la que llegarán correos indicando que un trabajo ha sido completado con éxito o por el contrario ha fallado o ha sido abortado, donde la dirección *someones_mail@hotmail.com* ha de reemplazarse con alguna que se disponga (puede ser una cuenta de correo personal). Ha de verificarse la carpeta correo no deseado, ya que, para el caso de esta práctica, los correos se envían por defecto desde la dirección *root@local*, que, a primera vista, no parece una dirección de correo válida.

Un ejemplo de correo electrónico que puede llegar a la cuenta especificada al finalizar un trabajo es el siguiente:

```
Job 33 (lammps.test_x1) Complete
 User             = usuhpc
 Queue            = all.q@compute-2-0.local
 Host             = compute-2-0.local
 Start Time       = 06/11/2014 20:29:09
 End Time         = 06/11/2014 20:53:48
 User Time        = 00:39:04
 System Time      = 00:01:17
 Wallclock Time   = 00:24:39
 CPU              = 00:40:21
 Max vmem         = 769.988M
 Exit Status      = 0
```

En él se puede ver el tiempo tardado en ejecutar el trabajo, como si se tratara del comando time de Linux, la fecha y hora de inicio y fin, memoria virtual máxima, estado de finalización del programa (en este caso, ha finalizado correctamente al retornar un 0), etc.

Entorno de paso de mensajes y procesadores a utilizar La línea *#$-pe orte 4* sirve para especificar un entorno de ejecución paralela, en este caso OpenRTE (orte). Para el caso actual es el correcto, ya que en la Sección 3.6.1 se vió que, al quitar temporalmente las variables de entorno que apuntaban hacia las bibliotecas MPI de Intel (que en realidad no estaban instaladas), las que respondían al comando mpiexec eran las de OpenMPI variante OpenRTE:

```
[root@moonstone ~]# mpiexec --version
mpiexec (OpenRTE) 1.4.3

Report bugs to http://www.open-mpi.org/community/help/
```

Existe una forma de conocer los entornos de paso de mensajes instalados en el sistema utilizando los comandos de Sun Grid Engine (SGE)[28] que posteriormente serán utilizados en este trabajo para lanzar los trabajos que han de ejecutarse en paralelo. En este caso, se puede utilizar el monitor de colas *qmon*, cuyo aspecto se muestra en la Figura 22, que al pulsar sobre el icono que en ésta aparece en la esquina inferior izquierda (similar a una "valla" de color gris) da acceso a la ventana "Parallel Environment Configuration".

Figura 22: Interfaz del monitor de colas qmon de SGE

El aspecto de la ventana "Parallel Environment Configuration" se puede ver en la Figura 23. En el cuadro de la izquierda aparece la lista de los entornos de paso de mensajes disponibles (mpi, mpich y orte).

El número 4 que aparece tras *orte* en la línea (*#$-pe orte 4*), se refiere al número de procesadores (o mejor dicho, núcleos) que se encargarán de ejecutar el trabajo. Es importante tener en cuenta que en el laboratorio, los Pentium 4 disponen de tecnología HyperThreading que simula la existencia de dos procesadores, aunque realmente sólo exista uno. Debido a esto, Sun Grid Engine asignará una parte del trabajo a cada procesador virtual, lo que

[28] Ver `http://es.wikipedia.org/wiki/Sun_Grid_Engine`.

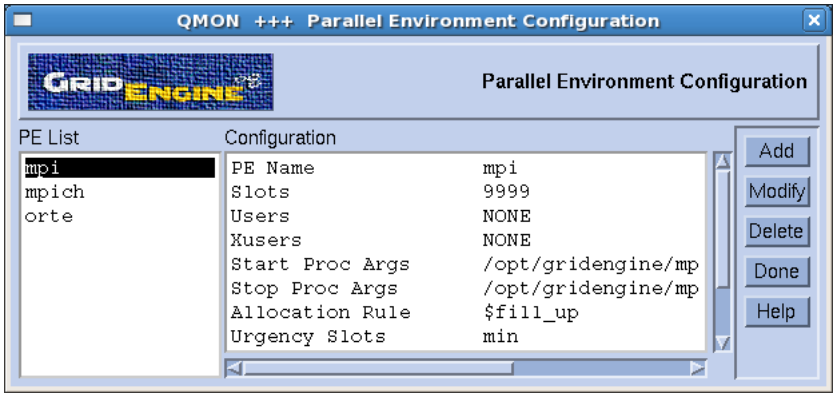

Figura 23: Ventana de Configuración del entorno paralelo en qmon, de SGE

puede conducir en algunos casos a un procesamiento más lento del trabajo que deshabilitando HyperThreading en los núcleos de cómputo.

Una explicación más detallada de cómo funciona el script `runjob.sh` puede encontrarse en [4].

ejecutar runjob

```
[usuhpc@moonstone lammps_test_xeon1]$ qsub runjob.sh
```

ver la cola de trabajos

```
[usuhpc@moonstone lammps_test_xeon1]$ qstat -f
queuename                      qtype resv/used/tot. load_avg arch          states
---------------------------------------------------------------------------------
all.q@compute-0-0.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-1.local        BIP   0/2/2          0.00     lx26-amd64
     34 0.60500 lammps.tes hpca       t     06/11/2014 20:26:09     2
---------------------------------------------------------------------------------
all.q@compute-0-10.local       BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-11.local       BIP   0/2/2          0.00     lx26-amd64
     34 0.60500 lammps.tes hpca       t     06/11/2014 20:26:09     2
---------------------------------------------------------------------------------
all.q@compute-0-12.local       BIP   0/2/2          0.00     lx26-amd64
     34 0.60500 lammps.tes hpca       t     06/11/2014 20:26:09     2
---------------------------------------------------------------------------------
all.q@compute-0-13.local       BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-2.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-3.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-4.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-5.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-6.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-7.local        BIP   0/2/2          0.00     lx26-amd64
     34 0.60500 lammps.tes hpca       t     06/11/2014 20:26:09     2
---------------------------------------------------------------------------------
all.q@compute-0-8.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-0-9.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-1-0.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-1-1.local        BIP   0/0/2          -NA-     lx26-amd64    au
---------------------------------------------------------------------------------
all.q@compute-1-2.local        BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
```

```
all.q@compute-1-3.local          BIP   0/0/2          0.00     lx26-amd64
---------------------------------------------------------------------------------
all.q@compute-2-0.local          BIP   0/4/4          -NA-     lx26-amd64     au
     32 0.50500 lammps.tes hpca      dr    06/10/2014 20:56:54     4
---------------------------------------------------------------------------------
all.q@compute-2-1.local          BIP   0/0/4          -NA-     lx26-amd64     au
---------------------------------------------------------------------------------
all.q@compute-2-2.local          BIP   0/0/4          -NA-     lx26-amd64     au

############################################################################
 - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
############################################################################
     33 0.50500 lammps.tes hpca      qw    06/10/2014 20:54:06     4
```

Listing 5: Salida del comando qstat -f para ver colas y trabajos

Referencias

[1] Department of Computer Science, University of Tennessee, Knoxville, Tennessee 37996-1301. *LAPACK Working Note 41, Installation Guide for LAPACK*, June 1999. Disponible en `www.netlib.org/lapack/lawns/lawn41.ps`, último acceso 21/05/2014.

[2] Department of Computer Science, University of Tennessee, Knoxville, Tennessee 37996-1301. *LAPACK Working Note 81, Quick Installation Guide for LAPACK on Unix Systems*, February 2007. Disponible en `www.netlib.org/lapack/lawns/lawn81.ps`, último acceso 21/05/2014.

[3] Broadband enabled Science and Technology Grid (BeSTGRID). Rocks utilities, 2010. Disponible en `http://technical.bestgrid.org/index.php/Rocks_Utilities`, último acceso 24/06/2014.

[4] National Partnership for Advanced Computational Infrastructure. Sge roll users guide: Submitting batch jobs to sge, 2009. Disponible en `http://www.rocksclusters.org/roll-documentation/sge/5.2/submitting-batch-jobs.html`, último acceso 12/06/2014.

[5] National Partnership for Advanced Computational Infrastructure. Base users guide: Command reference, 2012. Disponible en `http://www.rocksclusters.org/roll-documentation/base/5.5/c2117.html`, último acceso 24/06/2014.

[6] National Partnership for Advanced Computational Infrastructure. Base users guide: Install and configure your frontend, 2012. Disponible en `http://www.rocksclusters.org/roll-documentation/base/5.5/install-frontend.html`, último acceso 21/05/2014.

[7] National Partnership for Advanced Computational Infrastructure. [rocks-discuss] rocks reinstall on compute node, after disk rebuild, fails, 2012. Disponible en `https://lists.sdsc.edu/pipermail/`

npaci-rocks-discussion/2012-January/056284.html, último acceso 24/06/2014.

[8] R. Clint Whaley. Atlas installation guide, July 2012. Disponible en https://github.com/vtjnash/atlas-3.10.0/raw/master/doc/atlas_install.pdf, último acceso 21/05/2014.

[9] Wikipedia. Automatically tuned linear algebra software — Wikipedia, the free encyclopedia, October 2013. Disponible en http://en.wikipedia.org/wiki/Automatically_Tuned_Linear_Algebra_Software, último acceso 21/05/2014.

[10] Wikipedia. Basic linear algebra subprograms — Wikipedia, the free encyclopedia, April 2014. Disponible en http://en.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms, último acceso 21/05/2014.

[11] Wikipedia. FFTW — Wikipedia, the free encyclopedia, March 2014. Disponible en http://en.wikipedia.org/wiki/FFTW, último acceso 21/05/2014.

[12] Wikipedia. LAPACK — Wikipedia, the free encyclopedia, April 2014. Disponible en http://en.wikipedia.org/wiki/LAPACK, último acceso 21/05/2014.