Módulo

A

SISTEMA DE COLAS **SGE EE** EN LOS SERVIDORES DE CÁLCULO PARALELO DEL CPD **3MARES**

ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

CLUSTER HPC CALDERON

CPD 3Mares

ARQUITECTURA Y TECNOLOGÍA DE COMPUTADORES

CPD 3MARES

© Arquitectura y Tecnología de Computadores
FACULTAD DE CIENCIAS
UNIVERSIDAD DE CANTABRIA
Av. Los Castros, S/N • Santander
CANTABRIA (SPAIN)
soporte@atc.unican.es
Teléfono (942) 20 22 48 • Fax (942) 20 67 71

Tabla de contenido

CAPÍTULO 1

Contexto de trabajo

Entorno computacional 3

Esquema de acceso al CLUSTER 8

Cambio de password de usuario en el CLUSTER 10

CAPÍTULO 2

Introducción a los sistemas de gestión de trabajos por colas

Problemática existente 11

Soluciones 12

SGE EE 12

CAPÍTULO 3

Política Implementada para el CLUSTER

Sistema multicomputador de memoria distribuida 14

Nodo *frontend*, nodo de compilación y nodos de cómputo 15

Política y modo de uso 16

CAPÍTULO 4

Gestión de trabajos

Configuración del entorno 22

Envío de trabajos 22

Control de trabajos 30

Monitorización de trabajos 30

Entorno gráfico 35

CAPÍTULO 5

Recursos y PE disponibles

Recursos 41

Parallel environment 46

CAPÍTULO 6

Servicios adicionales

Sistema de copias de seguridad 49

CAPÍTULO 7

Conclusiones 54

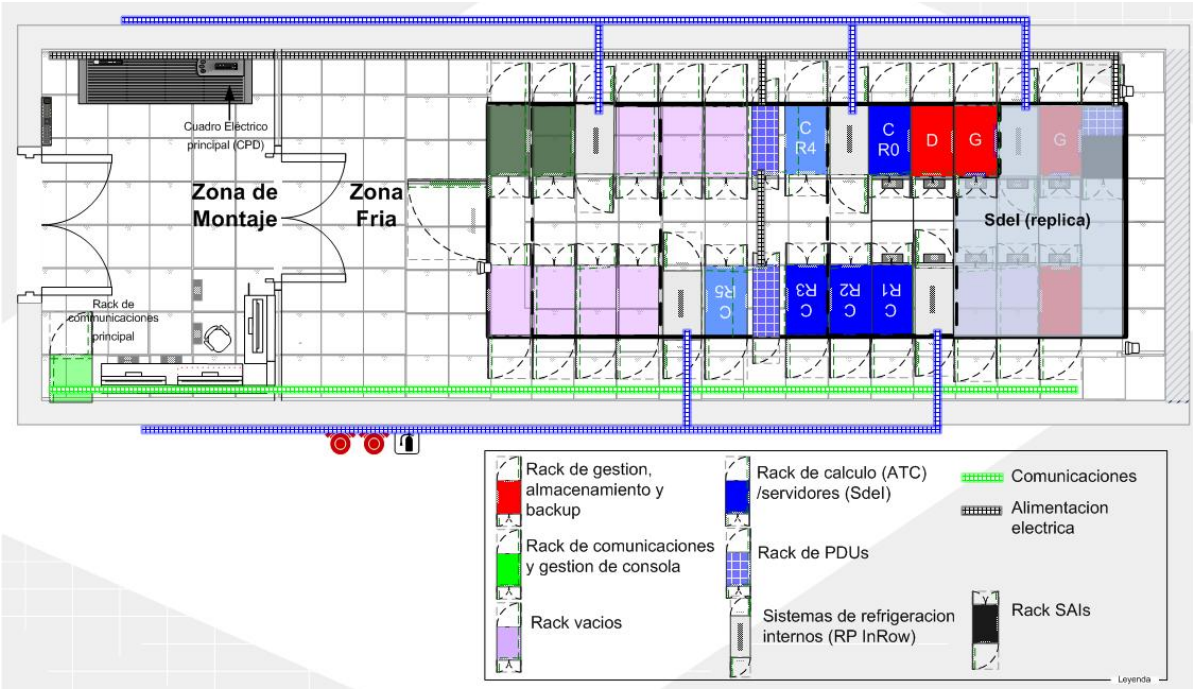
Capítulo

1

Contexto de trabajo

Entorno computacional

En nuestro grupo disponemos de algunos de los más potentes y completos entornos computacionales de cálculo paralelo de la Universidad de Cantabria. Todos ellos, instalados en nuestro Centro de procesamiento de Datos 3MARES, construido y ensamblado en el seno de la FACULTAD DE CIENCIAS de la UC y puesto en marcha en Julio de 2010. Las características mas destacados de nuestro CPD son las siguientes:



Acometida Eléctrica:

- Acometida general hasta **320 kW**
- Gestión eléctrica para *Computación Altas Prestaciones (HPC)*
    - Solución **alta escalabilidad**

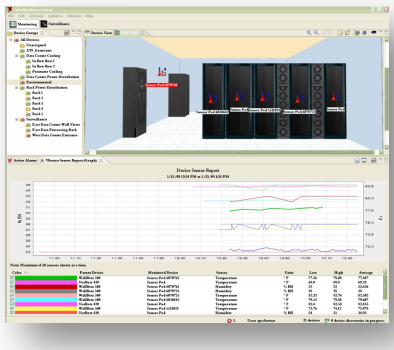
- Power Distribution Units (PDU) inteligentes
- Múltiples SAIs distribuidos **(2-6 KVAs)**

Refrigeración de Alta Eficiencia:

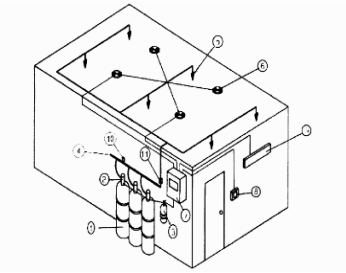
- Aislamiento de zonas calientes
  - **30%** más eficiente que soluciones basadas en impulsión por suelo técnico
- Solución **100% modular y escalable**
  - 5 Sistemas InRow con capacidad frigorífica actual de **184 kW**
  - Ampliable hasta cubrir los **320 kW** acometida eléctrica

Monitorización y gestión del CPD:

- **Monitorización Ambiental**
  - **+60 sensores** distribuidos por la solución
  - **Control de sistemas básicos**
  - Sistemas de refrigeración
  - Sistemas eléctricos
    - PDUs y SAIs "inteligentes"
    - **Control de accesos**
  - Sistema de vigilancia interno y control de accesos a 3mares
- Gestión centralizada



- Sistema de **monitorización global** (gráficos de estado, históricos de uso …)
- Gestión de **ALARMAS** en tiempo real
- "**TODO**" accesible de forma remota



Seguridad:

- Sistema **antiincendios**
  - detección automática de incendios
  - extinción automática con **agente gaseoso (FE-13)**
    - Inundación total de sala protegida

- Sistema de **control de accesos**
  - 3 niveles de control:
    - (1) principal a zona de montaje, (2) a zona "fría", (3) pasillo caliente
- Sistemas de **video-vigilancia**
  - Conectado a central de vigilancia de UC

Escalabilidad:

- El sistema es 100% escalable
  - Gracias al espacio dedicado y su diseño MODULAR, podemos escalar hasta los **32** racks de computación y **7** racks refrigeradores
  - 3mares está dividido en **módulos** compuestos por:
  - 1 enfriadora (+ intercambiador de cubierta)
  - 5 racks de cálculo
  - Disponibilidad actual (crecimiento a corto plazo)
  - 8 racks de cálculo
  - ~120KW de capacidad frigorífica

En cuanto a los sistema de cálculo, en este momento disponemos de los siguientes elementos:

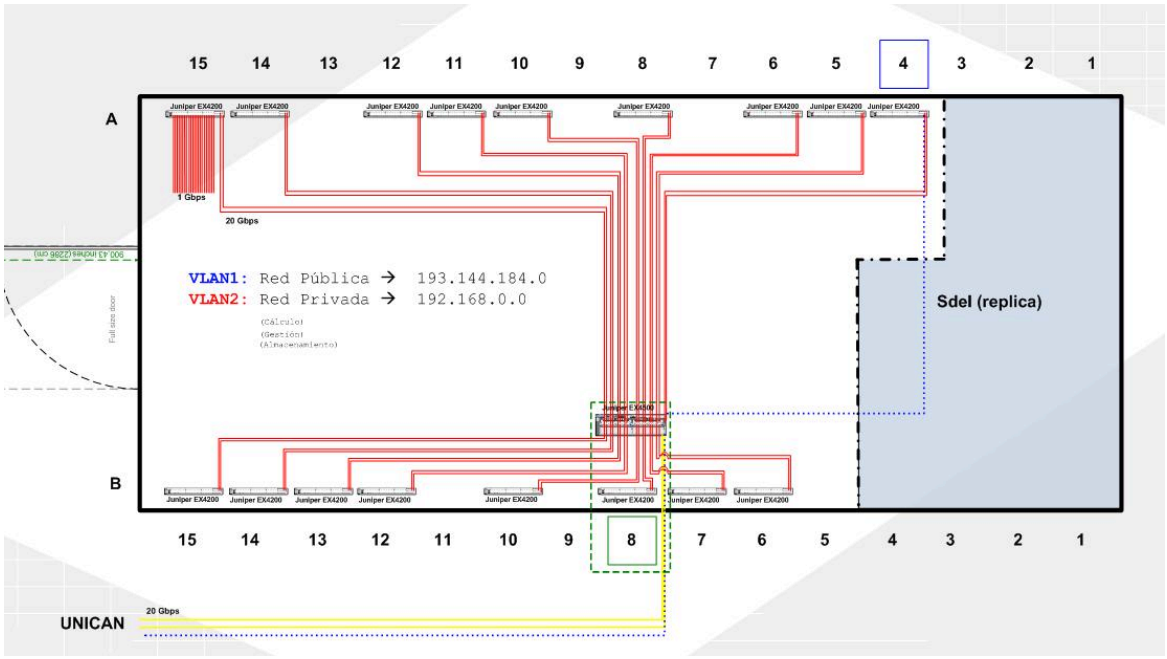
Cluster HPC

El clúster (y demás servidores departamentales de ATC) se encuentran encapsulados dentro del CPD en lo que llamamos **CUBO IntraStruXure** de APC, compuesto por 17 racks o bastidores, 4 unidades de refrigeración 2 dos *semi*-racks para gestión eléctrica.

Multicomputador clúster basado en elementos de HP/Supermicro fundamentalmente. Está formado por:

- ~~1 Front-end o servidor **HP Proliant DL 380 3º Generación** bi-procesador.~~
- ~~15 nodos **HP Proliant DL 145 1º Generación** bi-procesador con AMD Opteron 248.~~
- ~~20 nodos **HP Proliant DL 145 2º Generación** bi-procesador con AMD Opteron 265 dual core.~~
- 6 nodos **HP Proliant DL 145 2º Generación** bi-procesador con AMD Opteron 275 dual core.
- 30 nodos **HP Proliant DL 160 5ª Generación** bi-procesador con INTEL Xeon 5472 quad core.
- 1 "enclosure" **HP DL1000**: 4 nodos **HP Proliant DL 170h 6ª Generación** bi-procesador con INTEL Xeon X5550 quad core.
- 5 "enclosure" **SUPERMICRO SuperServer 6026TT-HT**: 20 nodos six-procesador INTEL Xeon X5650 six-core
- 5 "enclosure" **SUPERMICRO SuperServer 6026TT-HTRF**: 20 nodos six-procesador INTEL Xeon E5645 six-core
- 1 nodo **Supermicro SYS-6016GT-TF-TC2** con INTEL Xeon 5504 quad core y 2 tarjetas gráficas Nvidia Tesla C1060 GPU Cards
- 3 nodos **Sun UltraSPRAC T1 y T5120** (T2), con procesadores Niagará.
- ~~1 Switch **Myrinet** bi-canal con 2 Gbps por canal para comunicación paralela y 16 puertos.~~

- 1 Switch **Infiniband** Mellanox bi-canal con 10 Gbps por canal para comunicación paralela y 24 puertos.
- 1 Sistema de gestión de consolas KVM + monitor TFT.
- **Nueva RED ethernet** interna con topología en estrella de doble enlace 10Gb , implementada por 17 Switches **Juniper EX4200** con 2 enlaces 10Gb de fibra óptica y 48 puestos (10/100/1000) cobre cada uno y un switch central **Juniper EX4500** con 20 puertos 10 Gb de fibra óptica.



Se trata de una arquitectura multi-computador de memoria distribuida.

Procesadores	Memoria	Almacenamiento	Otros
2 x 15 AMD Opteron 248 a 2.2 GHz (g1) 2 (4) x 20 AMD Opteron 265 a 1.8 GHz (g2) **2 (4) x 6 AMD Opteron 275 a 2.2 GHz (g3)** **2 (8) x 30 Intel Xeon 5472 a 3 GHz (g4)** **2 (8) x 4 Intel Xeon X5550 a 3 GHz (g5)** **2 x 20 (6) x 4 Intel Xeon X5650 a 2,67 GHz (g6)** **2 x 20 (6) x 4 Intel Xeon E5645 a 2,4 GHz (g7)** **--** **2 (8) x 1 Intel Xeon 5504 a 2 GHz (ggpu)** **....** **TOTAL: ~ 824 cores**	Memoria principal: 4 x 15 GB PC2700 DDR1 SDRAM (g1) Memoria principal 8 x 20 GB PC3200 DDR1 SDRAM (g2) **Memoria Principal: 8 x 6 GB PC4700 DDR1 SDRAM (g3)** **Memoria principal: 16 x 30 GB PC2-5300 DDR2-667 SDRAM (g4/g5)** **Memoria principal: 20 x 48/24 GB DDR3 ECC SDRAM (g6)** **Memoria principal: 20 x 54 GB DDR3 ECC SDRAM (g7)** **TOTAL: ~ 2735 GB**	Discos IDE 15 x 40 GB (g1) Disco S-ATA 20 x 80 GB (g2/g3) **Disco S-ATA 32 x 160 GB (g4/g5)** **Disco S-ATA 40 x 250 GB (g6/g7)** **Disco S-ATA 12 x 500 GB (MSA20)** **Disco SAS 25 x 300 GB (MSA70)** **TOTAL: ~ 30,8 TB**	1 Unidad de DVD-ROM CD-RW ProLiant 100 Series Management Processor, supporting IPMI 1.5 x 15 1 PCI-X (64/133) spot x15 **2 PCI-X x 20** **1 PCI-Express x 24** **1 PCI-100 x 20** **1 PCI-Express x 25** **....**

El sistema operativo que usa la máquina es **Linux Debian**.

Nuestro centro de cálculo dispone de una red de comunicaciones **GigaBit** basada en una topología en "estrella" con enlaces 10 Gigabit Ethernet. de fibra óptica. Esta compuesta por 16 (+ 1) switches Juniper EX4200.

Y para que la seguridad no sea un elemento ausente en nuestro centro de cálculo, disponemos de un conjunto de medidas de seguridad para mantener los sistemas y sus servicios 24 horas al día, 365 días al año. Estas medidas se componen de los siguientes sistemas:

1. Cuatro sistemas de refrigeración que mantiene la sala de cómputo a una temperatura constante de 22º y humedad relativa entre 30-70% .

2. Sistema de UPS/SAIs distribuido por cada rack y compuesto por :

    a. 1 SAIs de 3 Kva cada uno para soportar todos los elementos de cálculo del clúster para los racks B10 de *red central (Ethernet)*.

    b. 8 SAIs gestionables de 6 Kva cada uno (48 Kva en total) para soportar todos los elementos de cálculo del clúster para el racks A8, A11, B7 y B8 y rack de *gestión (A4)*.

    c. 1 SAI de 1,125 Kvas para el rack de comunicaciones global al CPD, ubicado en "zona de montaje"

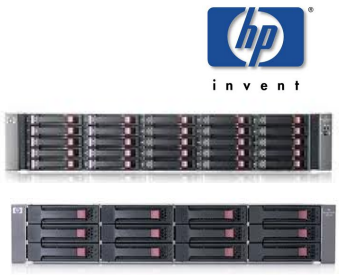
También disponemos de un completo sistema de alertas y medidas de contingencia automatizadas para las posibles incidencias que puedan tener logar en la sala (caída de frío, corriente eléctrica...) y que evitan posibles daños software y hardware en los servidores de la sala fria.

Y para el almacenamiento de datos en 2º nivel como copias de respaldo, disponemos de un

sistema de copias de seguridad para todos los datos de usuarios y sistemas del centro de cálculo, gestionados mediante un robot (HP MSL6060) de cintas magnéticas LTO Ultrium 3 con capacidad de hasta 800 GigaBytes por cinta, y con un total de 75 cintas (60 TB), gestionado mediante un sistema software de gestión de copias HP OpenView SatoreWorks DATA PROTECTOR 6.0.

Por otro lado, todos los usuarios de nuestros sistemas de cálculo paralelo poseen cuenta en los servidores actuales; calderon. Estos no accederán directamente a los servidores de cálculo, sino que utilizarán un *frontend* para ello. La pasarela de acceso será el propio calderon.atc.unican.es. Además de dar acceso así mismo y a los demás servidores de cálculo paralelo , actuará como *frontend* del clúster HPC que lleva su propio nombre.

Actualmente, la configuración de los discos de usuarios y por lo tanto, sus ficheros, posee dos variables:



- Discos de usuario **AFS**: Infraestructura formada por 3 bandeja de discos SATA/SAS:

    - HP MSA20 con 12 discos SATA de 500 GB cada uno (6 TB) y gestionados por controladora HP SmartArray 6404 UltraSCSI, implementando RAID 6.

    - HP MSA70 con 25 discos SAS de 300 GB cada uno (7,5 TB) y gestionados por controladora HP SmartArray P800 UltraSCSI, implementando también RAID 6.

    - 1 x Chasis Supermicro SC847E de 4U con 2 procesadores INTEL SANDY BRIDGE 6C E5-2620 2.0G 15M 7.2 1333MHZ , 32 GB de RAM DDR3 y 36 discos SAS2 de 1 TB cada uno (actualmente instalador 16 discos → 16 TB) y gestionados por controladora Expander supports SAS2 (6Gb/s). Ambas disponen de una tarjeta de red 10G PCI-E de fibra.

Ambas son gestionada por un servidor independiente cada una a través de un sistema de ficheros distribuido OpenAFS. Todos los usuarios inicialmente, cuando se crean sus cuentas,

poseen en dicha infraestructura un espacio de usuarios centralizado y visible (y modificable, claro) desde todos de servidores de calculo de ATC y equipos (internos y externos al departamento). De esta manera, cuando los usuarios se logean, su PATH *de usuario* estará ubicado físicamente en el clúster, y virtualmente en todas las demás (/afs/atc.unican.es/u/).

- Discos de usuario **ZFS/Lustre**: Infraestructura formada por 2 chasis de discos SAS:

    - 1 x Chasis Supermicro SC847E de 4U con 2 procesadores INTEL SANDY BRIDGE 6C E5-2620 2.0G 15M 7.2 1333MHZ , 32 GB de RAM DDR3 y 36 discos SAS2 de 1 TB cada uno (actualmente instalador 16 discos → 16 TB) y gestionados por controladora Expander supports SAS2 (6Gb/s). Ambas disponen de una tarjeta de red 10G PCI-E de fibra.

    - Servidor virtual (dominio XEN) implementado sobre contenedor XEN c4, con 2 unidades de proceso INTEL X5472 3.0G 6M, 4 GB de RAM DDR3, y 1 interfaz de red 1Gbps.

Ambos chasis SC847E, servidores autónomos, conforman el *backend* de almacenamiento del sistema ZFS/Lustre, denominados OSS u "Objet Store Server. ". El servidor virtual Xen implementa el MDS o "MetaData server".

La distribución y organización de los datos (ficheros y directorios) en lustre es similar a AFS. A partir de su directorio raíz /lustre, "cuelga" el directorio de celda atc.unican.es y tras éste, las carpetas globales:

/lustre/atc.unican.es/u →usuarios

/lustre/atc.unican.es/p → proyectos

- Discos de usuarios **Locales** a calderon (nodos de cálculo): Cada servidor dispone de una infraestructura de discos locales que usaremos también en algunos casos para ubicar los PATHs de usuarios y como sistemas de almacenamiento auxiliares. En total, se dispone de una capacidad de almacenamiento que supera los 16 TB para dichos discos. El uso de estos discos locales se gestionará de forma distribuida por cada servidor y al menos un 80% de su capacidad se destinará para datos temporales (SCRATCH).

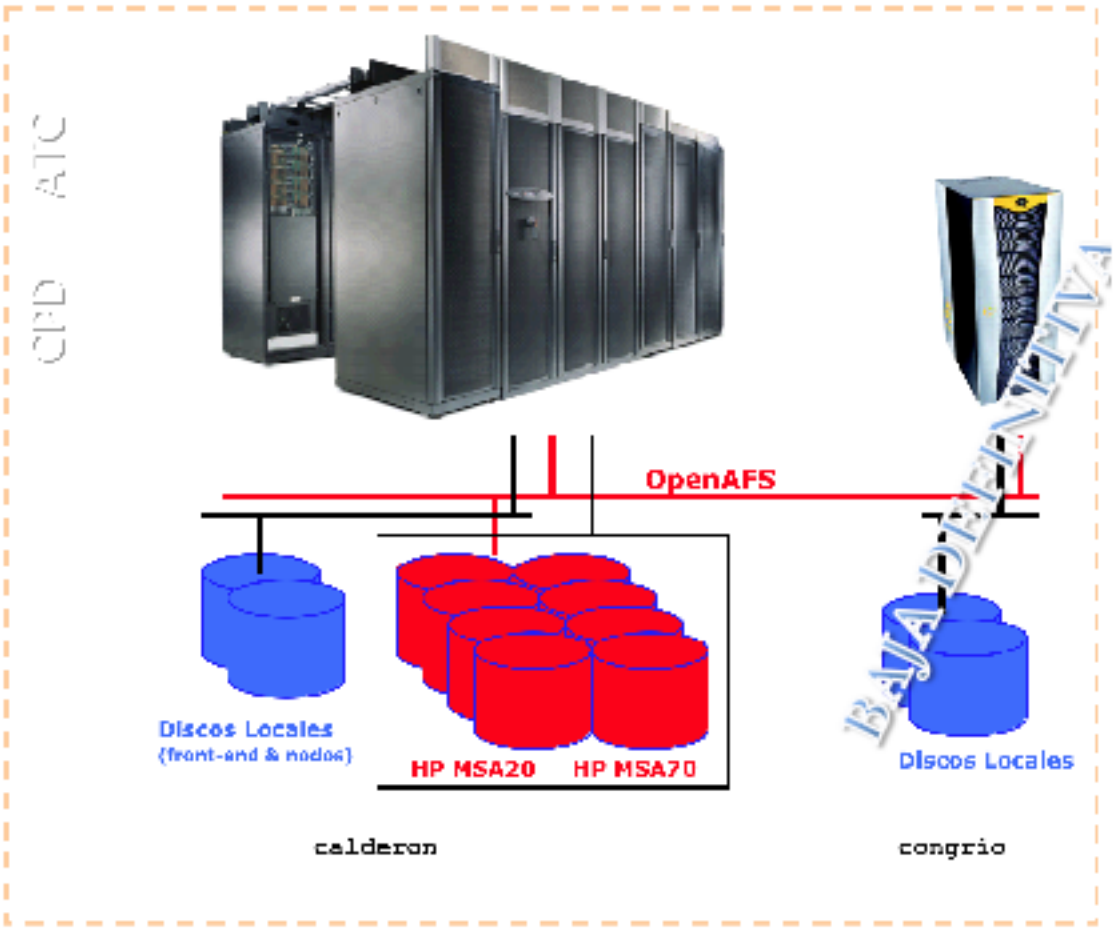


Ilustración 1 Esquema de almacenamiento (discos) definitivo de servidores de ATC

Esquema de acceso al SERVIDOR

El esquema de acceso al CLUSTER HPC calderon.atc.unican.es será el siguiente:

El primer paso es acceder al *frontend* de ATC. Antes utilizamos el servidor puersgi1.atc.unican.es (SGI ORIGIN 200) como "pasarela" de seguridad hacia los servidores de cálculo. Actualmente, nuestra infraestructura de cálculo dispone de 2 *frontend* idénticos, ambos servidores virtualizados sobre un computador físico. Usaremos dichas "máquinas" como "pasarela" hacia los demás servidores de cálculo y como *frontend* del clúster.

Para ello debemos abrir una **sesión segura SSH** contra la máquina utilizando nuestro nombre de usuario y password. Un vez allí, para acceder al resto de servidores, debemos usar únicamente sesión SSH, usando también aquí el mismo nombre usuario y password. Los servicios rlogin, telnet o similar han sido eliminados para evitar comprometer la seguridad de nuestros sistemas.

En resumen, el siguiente esquema muestra de forma gráfica la política accesos a utilizar:

Ilustración 2 Esquema de conexión a los sistemas de ATC

También es posible acceder al sistema en modo "**gráfico**" a través de un *frontend* especial, llamado fe-ts.atc.unican.es, el cual despliega tecnología GPU Nvidia Tesla (C1060 GPU Cards) combinado con el software ThinLinc (Terminal Server) para proporcionar escritorio remoto Linux a usuarios del sistema.

Cambio de password de usuaria en el CLUSTER

Para cambiar el password en el CLUSTER HPC calderon.atc.unican.es solo será necesario ejecutar el siguiente comando desde cualquiera de los frontend's disponibles:

```
Shell#:> passwd
```

Capítulo

2

Introducción a los sistemas de gestión de trabajos por colas

Un buen sistema de colas debe ser tolerante al fallo de los trabajos, adaptarse a la carga de las CPU y poder migrar procesos. Todo ello se verá complementado con un planificador (Schedule) que será el encargado de determinar cuándo, dónde, y cómo los trabajos se ejecutan para maximizar el rendimiento de la máquina. Este componente software será, en definitiva, quien aporte la inteligencia al sistema de colas...

Problemática existente

El número de usuarios de los sistemas de cálculo paralelo del grupo ATC es superior a 60 actualmente y es previsible que esta cifra siga creciendo. Bajo estas circunstancias y dado el uso que se le da a esta clase de sistemas, es habitual que existan diversos usuarios intentando ejecutar más trabajos de los que puede manejar eficientemente la máquina. En el caso del clúster HPC, hay disponibles mas de 800 procesadores con perspectivas de ampliación dada su naturaleza de alta escalabilidad. Intentar realizar más trabajos que procesadores existentes de forma simultánea es claramente contraproducente por las siguientes razones:

- Los trabajos han de abandonar el procesador cada cierto tiempo para cederle su lugar a otros trabajos que también se están ejecutando y que en ese momento se encuentran parados por no poseer un procesador disponible. Esto se denomina cambio de contexto y es extremadamente costoso en tiempo, y a la larga lo que sucede es que todos los trabajos tardan mucho más tiempo que en el caso de que esto no fuera necesario.

- El total de memoria física disponible puede no ser suficiente para todos los trabajos. Si eso ocurriese el sistema empezaría a usar memoria virtual. Esta memoria esta en el disco duro. Aunque estos discos de altas prestaciones son rápidos en comparación con discos normales, su velocidad es del orden de 3 o 4 ordenes de magnitud inferior a la memoria física. En consecuencia, los trabajos que requieran hacer uso de la memoria virtual necesitarán mucho mas tiempo que los que solo requieran memoria física.

Estos efectos son mucho más perniciosos en el caso de los programas en paralelo (MPI/OpenMP …). Si uno de los hilos de ejecución se ve afectado por trabajos de otro usuario, entonces el rendimiento de la aplicación disminuye de forma crítica, llegando a darse el caso de que la aplicación paralela tarde mucho más que la versión secuencial.

En conclusión, el uso indiscriminado de un sistema de alto rendimiento sin ningún control, implica que todos los usuarios se vean afectados, reduciendo notablemente el rendimiento global del mismo. Esto sucede con el modo de trabajo interactivo[1] que se utiliza en máquinas sin sistemas de gestión de trabajos por colas.

Soluciones

La solución a esta clase de problemas son los sistemas de gestión de trabajos en *batch,* también llamados sistema de colas. Se trata de un mecanismo encargado de gestionar de forma "racional" los recursos del sistema. En este caso siempre el sistema se mantiene en los parámetros de carga óptimos. Los usuarios se despreocupan de cómo se encuentra el sistema antes de lanzar trabajos y es el propio gestor de colas el encargado de que los niveles de carga que soporta el sistema en todo momento no superen los valores máximos admisibles. La idea, por tanto, es NO enviar los trabajos de forma directa y sin control sobre el sistema, sino a través del sistema de colas. De esta manera, el sistema de colas actúa como interface entre el usuario y el sistema.

SGE EE

El personal técnico, tras un amplio estudio de los sistemas de colas más utilizados en uso de computación de altas prestaciones[2], por **SGE EE** (*Sun GridEngine Enterprise Edition).* También la documentación relativa a este sistema se encuentra en la Web de soporte.

Es, de alguna manera, similar funcionalmente a sistemas como LSF y a los sistemas de colas en red tipo NQS, pero mucho más evolucionado y **adaptable** a las necesidades y características de un entorno computacional específico (probablemente alguno ya estará familiarizado). Implementa políticas de gestión de usuario, gestión de proyectos en base a "tickets" de consumo, PREEMTION de trabajos … Además dispone de versiones OpenSource, lo cual hace que encaje directamente dentro de nuestras políticas de bajos costes y mantenimientos. Actualmente es soportado por una extensa comunidad de programadores y administradores de sistemas (foros …) debido a su extensa utilización en entornos computacionales de carácter científico.

En lo sucesivo el trabajo en modo directo o interactivo sobre el sistema va a quedar completamente prohibido, salvo solicitud de forma expresa al administrador, debiendo utilizar siempre el sistema de colas para mandar nuestros trabajos.

[1] El trabajo interactivo es aquel que se realiza directamente desde una shell o terminal.

[2] Ver Articulo publicado en la X edición de las Jornadas de Paralelismo (MADRID 2003), Análisis y evaluación de sistemas de colas en un entorno HTC. http://www.atc.unican.es/investigacion/publicaciones/publicaciones_files/publ_22.pdf

Este documento pretende explicar, de forma resumida y particularizada en nuestro CLUSTER HPC calderon.atc.unican.es, como se maneja el sistema de colas **SGE EE,** fundamentalmente en cuanto a la gestión (envió, suspensión, borrado…) y monitorización de trabajos para su correcta ejecución.

Capítulo

3

Política implementada para el CLUSTER

En este apartado comenzaremos a explicar de forma breve y sencilla las ideas básicas de la política implementada con **SGE EE** para construir un entorno de trabajo compartido en nuestro CLUSTER HPC calderon.atc.unican.es, proporcionando así a los usuarios una visión global y única del sistema.

Sistema multicomputador de memoria distribuida

En primer lugar debemos tener en cuenta que dicha política varia sensiblemente respecto de la implementada en sistemas multiprocesador de memoria compartida. Esto se debe a la diferencia de arquitectura, en todos los sentidos, entre ambas propuestas computacionales. Los sistemas multiprocesador son una única máquina físicamente. Sin embargo, un clúster HPC es básicamente un sistema multicomputador compuesto por un elevado número de computadores completamente independientes interconectados entre sí por una red de altas prestaciones, en la que uno o varios de los computadores mantiene el "roll" de frontend y el resto de nodos de cálculo.

Los usuarios se logean al clúster a través de sus frontend (*hostname* calderon) mediante **conexión segura SSH**. Una vez ahí, las únicas tareas permitidas son el envío y control (gestión) de trabajos a través del sistema de colas, etc… Hay que recordar en este punto que, para un usuario, el clúster es el frontend (los frontend) y un conjunto de recursos hardware accesibles, en cuanto a la ejecución de trabajos, solamente mediante el sistema de colas. Esto significa que los nodos de cómputo son solo eso, nodos de cómputo, y no serán accesibles mediante *Shell* interactiva, salvo a través del sistema de colas.

Esto hace que SGE EE deba configurarse de manera especializada, atendiendo a una política de gestión de trabajos que contemple estas características, diferentes en todo caso a las de un sistema multiprocesador.

Nodos *front-end*, nodos de compilación y nodos de cómputo

Dadas las características descritas arriba, SGEE se estructura de la siguiente manera:

- **Nodos frontend**: En estos nodos (actualmente 3, dos con acceso ssh y un tercero con acceso de "escritorio remoto") se ejecutarán los procesos de gestión y control de colas (y otros elementos del sistema de colas), Schedule, y comunicación con nodos de cómputo. También se ubicará alrededor del 70 % del software del sistema de colas, como son el core o núcleo del sistema, toda la configuración del mismo (colas, *complexes, parallel environment*, calendarios, usuarios especiales…), los comando y procesos de gestión y control de trabajos (envió, detención, borrado, modificado, suspensión…) y los necesarios para su monitorización y registro (logs).

- **Nodos de compilación**: Se trata de nodos (actualmente 2 servidores físicos) exactamente iguales a cualquiera de los nodos de cálculo, solo que su dedicación es exclusiva a tareas de compilación y depuración de código. Es por ello por lo que en estos nodos tenemos, a diferencia de los demás, instalado tanto los compiladores (diferentes versiones de Porland Group Suite, PathScale Suite, Intel Compiler y GNU Compilers), debbugers y librerías específicas necesarios para dichas tareas, como la infraestructura del sistema de colas adecuada para entrar en ellos cada vez que un usuario lo requiera. El acceso a éstos se lleva a cabo de forma automática a través, como del sistema de colas, mediante un algoritmo de balanceo de carga. También es posible acceder a cada uno de ellos directamente, indicándoselo de forma explícita al comando de envió de trabajos del sistema de colas. Mas adelante se indica este procedimiento con mas detalle.

- **Nodos de cómputo**. En estos se ejecutarán los procesos de comunicación con el frontend, control de ejecución de los propios trabajos y el encargado de recopilar en todo momento información sobre el estado de los recursos del sistema local

    - **Nodos de cómputo regulares**: Son prácticamente todos los nodos del clúster, descritos en la tabla de arriba.

    - **Nodos de cómputo especiales:** Son un grupo de nodos con características hardware y arquitectónicas especiales. En este momento disponemos de 3 nodos de este tipo:

        - 2 nodos **Sun UltraSPARC SunFire T1**, con procesador **Sun Niagara** 1 GHZ y 8 GB de memoria. Sistema Operativo **SOLARIS 10.**

        - 1 nodo **Sun UltraSPARC SunFire T5120 (T2),** con procesador **Sun Niagara** 1,2 GHZ y 8 GB de memoria**. Sistema Operativo SOLARIS 10.**

      Para el acceso y utilización de éstos nodos, mientras no se indique lo contrario, se debe realizar una solicitud al administrador del sistema.

Política y modos de uso

Para este tipo de arquitectura hemos querido cambiar la idea de uso de un sistema de colas, no centrándolo en el conocimiento por parte del usuario de la configuración del propio sistema, sino en el conocimiento de los trabajos a ejecutar, es decir, de sus propios trabajos.

En sistemas multiprocesador anteriormente administrados por este personal técnico, con sistemas de colas LSF, el usuario enviaba sus trabajos a una cola de ejecución sobre la propia máquina. Para ello era necesario el conocimiento del mismo sobre las colas definidas y sus características.

Ahora se trata de enviar los trabajos directamente al sistema, sin especificar una cola concreta y con la mínima especificación del sistema y/o recursos hardware a emplear en su ejecución. Lo que haremos es indicarle a **SGE EE** como es, a grandes rasgos, nuestro trabajo.

Por lo tanto, siempre utilizaremos el sistema de colas para ejecutar trabajos en el clúster. Esta es la **regla de oro** de nuestra política de uso. Como segunda idea básica está el indicarle al sistema de colas (a través de los propios comandos de envío y gestión) y en el momento del envío del trabajo, alguna característica del mismo sin especificar la cola donde ha de ubicarse el trabajo. Esta tarea corresponde ahora al propio sistema de colas.

Para nuestro CLUTER, clasificaremos los trabajos a enviar según los siguientes criterios:

1. **Modo de ejecución:** Este criterio distinguirá los trabajos interactivos, en los que el usuarios mantiene cierto control sobre su ejecución de manera activa y de los trabajos *batch* o encolado, los cuales se ejecutan sin ninguna interacción por parte del usuario.

    a. **En modo interactivo**: Esta modalidad se refiere a la ejecución de trabajos en la que el usuario interactúa de forma directa en la ejecución del trabajo. Debemos recordar en este punto que **en ningún caso** ejecutaremos trabajos directamente en la Shell, salvo que sean en el propio frontend del clúster (host calderon) y se trate de ejecuciones de testeo y estén autorizadas por el administrador. En el resto de casos, utilizaremos el sistema de colas para ejecutar este tipo de trabajos (interactivos, compilación etc.).

    b. **En modo *batch* o encolado**: Esta segunda modalidad se refiere, como se puede intuir, a la ejecución de los trabajos a través del sistema de colas y sin interacción por parte del usuario. Este lanza el trabajo y se ejecuta en la máquina bajo los criterios definidos y "decididos" por el sistema de colas. Los resultados son devueltos al usuario en forma de *mail* y/o a fichero.

2. **Tiempo de duración del mismo:** Tendremos aquí 4 tipos de trabajos; de larga, media, corta y muy corta duración.

    a. **Larga duración:** (sin límites de tiempo)

    b. **Media duración:** hasta 72 horas

    c. **Corta duración:** hasta 3 horas

d. **Muy corta duración (interactivos):** hasta 30 minutos.

3. **Naturaleza de los trabajos:** Se distinguirán así los trabajos paralelos, secuenciales y de compilación.

   a. Trabajos Paralelos: Serán trabajos paralelos los que estén compuestos por más de 1 proceso con necesidades de algún tipo de comunicación inter-proceso. Esto son aquellos que han sido implementados mediante MPI y OpenMP fundamentalmente. Es por esto por lo que a su vez los segmentamos en dos subtipos:

      i. **MPI:** implementados con el interfaz de paso de mensajes MPI (mpich, mpich2 y mvapch) para dar soporte a la comunicación inter-proceso en sistemas multicomputador de memoria distribuida (CLUSTER)

      ii. **OpenMP:** implementados con las cláusulas de compilador o mediante variables de entorno OpenMP, para soportar comunicación inter-proceso en sistemas de memoria compartida (nodos del clúster, 2/4/8 procesadores SMP)

   También tendremos en cuenta aquí el tipo de red o hardware (conociendo las posibilidades hardware que nos ofrece el clúster) donde vamos a ejecutar el trabajo, ya sean MPI o OpenMP:

      ~~i. **Red Myrinet (myr):** Si el trabajo se compila y ejecuta en los nodos interconectados mediante la red myrinet (16 nodos).~~

      ii. **Red Infiniband (inf):** Si el trabajo se compila y ejecuta en los nodos interconectados mediante la red infiniband (24 nodos).

      iii. **Red Ethernet:** Si el trabajo se compila y ejecuta en los nodos interconectados mediante la red Ethernet. Todos los nodos del clúster están interconectados bajo esta red, usada principalmente para tareas de gestión. Para ello, solo deberemos especificar que el trabajo es paralelo, sin indicar ningún tipo de red específica.

   b. **Trabajos NO Paralelos (secuenciales):** Serán trabajos secuenciales los trabajos que posean uno o más procesos que NO requieran comunicación entre ellos. Estos a su vez los dividimos en dos subtipos:

      i. **Múltiples:** trabajos con más de un proceso secuencial idénticos lanzados en un mismo trabajo.

      ii. **Simples:** trabajos secuenciales de un solo proceso.

   Y aquí también consideraremos el hardware de la máquina. En este caso no por la red de interconexión de los nodos de cálculo, si no por los propios nodos de cálculo, ya que en el clúster se distinguen los siguientes grupos:

i. **[FUERA DE SERVICIO[3]] Grupo 1 (g1):** grupo formado por **9 nodos** con 2 procesadores (2 cores) AMD Optaron 248 a 2.2 GHz y 4 GB de memoria principal cada uno.

ii. **[FUERA DE SERVICIO[4]] Grupo 2 (g2):** grupo formado por **21 nodos** con 2 procesadores (4 cores) AMD Optaron 265/275 a 1.8/2.2 GHz y 8 GB de memoria principal cada uno.

iii. **Grupo 3 (g3):** grupo formado por **16 nodos** con 2 procesadores (8 cores) Intel Xeon X5472 a 3 GHz y 16 GB de memoria principal cada uno.

iv. **Grupo 4 (g4):** grupo formado por **9 nodos** con 2 procesadores (8 cores) Intel Xeon X5472 a 3 GHz y 16/32 GB de memoria principal cada uno.

v. **Grupo 6 (g6):** grupo formado por **20 nodos** con procesadores (12 cores) Intel Xeon X5650 a 2,67 GHz y 24/48 GB de memoria principal cada uno (habilitado HT).

vi. **Grupo 7 (g7):** grupo formado por **20 nodos** con procesadores (12 cores) Intel Xeon E5645 a 2,40 GHz y 54 GB de memoria principal cada uno.

iii. **Grupo 8 (gslow):** grupo formado por los **21 nodos mas lentos** del clúster, con procesadores (4 cores) AMD Optaron 265/275 a 1.8/2.2 con 8 GB de memoria principal cada uno (habilitado HT)..

iv. **Grupo 9 (gfast):** grupo formado por los **40 nodos mas rápidos y con mejores prestaciones** de cálculo del clúster, con procesadores (12 cores) Intel Xeon X5550, X5650 y E5645 hasta 2,67 y 3 GHz y 32/48/54 GB de memoria principal cada uno.

v. **Grupo 10 (ghuge):** grupo formado por los **20 nodos** con **mas capacidad de memoria RAM** del clúster, todos ellos con procesadores (12 cores) Intel Xeon E5645 a 2,4 GHz y 54 GB de memoria principal cada uno.

c. **De compilación:** Son los trabajos (de naturaleza interactiva) en los que el usuario realizarla las tareas propias de compilación y depuración del código de sus aplicaciones. Actualmente el clúster dispone de 3 nodos de compilación, agrupados en 2 conjuntos:

i. **Grupo 1 (comp, comp-g1, comp-pgi y comp-ps):** grupo formado por 1 nodo físico (no virtual) dedicado bajo sistema operativo Debian 6.0 con 4 procesadores AMD Optaron 275 a 2.2 GHz y 8 GB de memoria principal. El nodo integra todo tipo de herramientas y librerías para la compilación de fuentes, tanto de GNU como de PathScale[5] e Intel.

i. **Grupo 2 (comp-g2):** grupo formado por 1 nodo físico (no virtual) dedicado bajo sistema operativo Debian 7.0 con 4 procesadores AMD

[3] Dados de Baja DEFINITIVA como elementos activos de cálculo del clúster.

[4] Dados de Baja como elementos activos de cálculo del clúster. Actualmente usados para tareas de desarrollo y entornos de prueba por los administradores de sistemas del CPD 3Mares. Pueden también SER SOLICITADOS por investigadores asociados al grupo ATC para su uso "temporal" aislado en exclusiva de tareas de cálculo.

[5] Consultar versiones en lo propios nodos o preguntar al administrador.

Optaron 275 a 2.2 GHz y 8 GB de memoria principal. El nodo integra versionas **mas actuales** de las herramientas y librerías para la compilación de fuentes, fundamentalmente de GNU.

En función de todos estos criterios, distinguiremos a nuestros trabajos y los clasificaremos para su correcto envío y monitorización a través del sistema de colas. Esto será fundamental para saber que parámetros del sistema de colas hay que usar.

De manera conceptual, los parámetros necesarios para el envío de los trabajos serán los siguientes:

- Shell que ha de ejecutar el trabajo[6*]. Le indicaremos la Shell que usará el sistema de colas para ejecutar el trabajo en el servidor. Puede tratarse de cualquiera de las disponibles en el sistema. Para verlas, ejecutar el comando *more /etc/shells*.

- Nombre del ejecutable*. Este parámetro debe indicársele para el correcto funcionamiento del sistema. Se trata del nombre del binario que se ejecutará en el servidor.

- Nombre del trabajo. De carácter opcional, pero recomendable, el nombre que adoptará el trabajo a nivel de sistema de colas.

- Ubicación de los ficheros de salida de resultados y error. También le podemos indicar el/los path(s) donde de ubicarán los ficheros de salida estándar (datos resultado) y error de nuestro trabajo. Estos adoptará el siguiente formato:

  <nombre del trabajo>.<identificador de salida> identificador del trabajo.

  <identificador de salida> = { **o** (salida de datos resultado); **e** (salida de errores); **po ^ pe** (salida estándar y de errores del entorno paralelo) }

- Entorno paralelo* (si procede, solo para trabajos paralelos). En caso de tratarse de un trabajo paralelo MPI u OpenMP, debemos indicarle el *entorno paralelo* donde ha de ejecutarse. Además este entorno nos va ha servir para ubicar los procesos paralelos de forma secuencial (los procesos se van ubicando en los nodos de forma que completan (llenen) el nodo. También reconoceremos este modo como *fill-up)*, o de forma intercalada (los procesos se van ubicando uno por nodo y saltando al siguiente. También reconoceremos esta forma como *round-robin*). Actualmente, los entornos paralelos mas usados son mpich, mpich2, mvapich y mthreads y mthreads-g{1-7,gpu}.

- Duración/Naturaleza del trabajo/hardware. El sistema mantendrá **4 tipos** de trabajos según su duración (muy corta, corta, media y larga (infinita) duración) y otros **3 tipos** según su naturaleza secuencial/paralela/compilación.

[6] * Parámetros obligatorios en todos los casos

- Número de procesos del trabajo (si procede). Para trabajos OpenMP, le indicaremos el número de procesos a ejecutar. En nuestro caso 2, 4, 8 o 12.

- Numero de trabajos múltiples a ejecutar (si procede). Para lanzar trabajos NO PARALELOS de carácter múltiple, debemos indicarle el número de trabajos a ejecutar, tanto por nodo como a nivel general en el servidor.

- Definición de número de Threads (si procede). También para trabajos OpenMP, debemos definir un par de variables de entorno que le indicarán al sistema el número de *threads* a ejecutar.

Con esta información, el sistema de colas será capaz de lanzar el trabajo en el momento en que sea posible o cuando halla recursos suficientes en la máquina para poder ser ejecutado. Indicar que se pueden pasar mas parámetros de caracterización, pero con los expuestos arriba es suficiente para mantener una ejecución fluida en nuestro sistema.

Y para terminar con la política implementada con **SGE EE** en nuestro clúster, debemos indicar algunos otros aspectos también muy importantes y que el usuario debe conocer, como son las prioridades en la ejecución de los trabajos y las limitaciones impuestas por el propio sistema.

Las prioridades de ejecución[7] de trabajos serán las siguientes:

Desde el punto de vista de la duración de los trabajos, los trabajos de más duración son menos prioritarios que los de menos duración. Y desde el punto de vista del paralelismo, los trabajos paralelos, por su propia naturaleza (las interferencias pueden ser desastrosas en su rendimiento), son más prioritarios que los secuenciales. Esta prioridad quedará patente SOLAMENTE en los nodos donde se disponga de red de altas prestaciones (myrinet e infiniband), con la SUSPENSION de trabajos secuenciales en estos nodos en el caso de entrar trabajos paralelos. En el resto de nodos (donde la única red de interconexión es Ethernet), los trabajos paralelos y secuenciales mantendrán una misma prioridad, compartiendo incluso las colas de ejecución sobre los nodos.

Y en cuanto a los límites (tiempo de duración, recursos …), hay que decir que estos no podrán ser superados. En caso que sea así, el sistema **matará** los procesos que los superen, y el sistema dará por concluido el trabajo.

Estos son los propios límites impuestos por las características hardware de la máquina; No se podrán superar los dos/cuatro/ocho/doce[8] procesos por nodo (2/4/8/12 procesadores), 4/8/16/32/48/54 Gbytes de memoria física por nodo, 8/16/32 Gbytes de tamaño de swap por nodo, así como cualquier otra limitación hardware. El tiempo de duración de los trabajos también se considerará una limitación hardware, por lo tanto superar dicha limitación acarreará las mismas consecuencias ("Kill" del proceso). Ver apartado "Recursos", del capítulo 5.

También se debe tener en cuenta que se establece un límite en el **número de trabajos** que pueden estar en ejecución en un instante dado para un usuario concreto. Dicho límite está establecido actualmente en 256 trabajos. Cualquier usuario que envíe mas trabajo, solo podrá tener, como máximo, 256 trabajos en estado de ejecución. Los demás trabajos permanecerán en estado "pending" (qw). Cuantos mas trabajos se tengan en este estado, mas se penalizará

[7] En función de las prioridades, unos trabajos suspenderán a otros. Es decir, cuando un usuario lance un trabajo asociado a una prioridad mas alta, éste suspenderá a trabajos que se encuentren en ejecución asociados a una prioridad menor.

[8] Actualmente, nuestro clúster posee 7 tipos de nodos de cálculo con idéntico hardware pero con capacidades diferentes. Ver pag. 6 del presente documento.

respecto a otros usuarios con menos trabajos. Esto implica que los usuarios que envíen grandes "baterías" de trabajos, serán penalizados respecto a usuarios que envíen grupos mas pequeños de trabajos por el Schedule del sistema de colas.

Datos de *scratch*

En todos los nodos del clúster existe una partición local independiente y dedicada al uso de datos temporales o de *scratch*. Esto implica que todas las aplicaciones que se ejecuten en el clúster a través de trabajos, ya de forma interactiva o en *batch*, deberán usar exclusivamente dicha partición para tal efecto.

Se ha creado un enlace directo a la partición en todos los nodos:

`/scratch`

El contenido de dicha partición solo será ELIMINADO en dos casos:

1. Apagado/reinicio del clúster para tareas administrativas (**PARADAS**)
2. Cuando la partición se llene completamente

Queda por tanto prohibido el uso de otros directorios y/o particiones como repositorios de datos temporales o de *scratch*, como por ejemplo:

- `/tmp`
- `/var/tmp`

Capítulo 4

Gestión de trabajos

A continuación describiremos los procedimientos para el envió y control posterior de los trabajos y la monitorización de los mismos, a través de los comandos del propio sistema de colas y mediante herramientas proporcionadas por el sistema operativo. Todos los procedimientos se describirán utilizando *línea de comandos*. **SGE EE** también dispone de un potente interfaz gráfico que describiremos de forma general en la última parte de este apartado.

Configuración de inicio para los usuarios

Previamente a usar el sistema de colas para enviar trabajos al clúster (nodos de cálculo), debemos actualizar nuestro entorno de trabajo (shell). Todos los ficheros asociados al sistema de colas **SG EE** se encuentran situados bajo la jerarquía /opt/sgeee/. Para poder utilizar este sistema será necesario modificar la variable de entorno $path y definir una nueva variable propia de **SGE EE**. En /opt/sgeee/default/common/settings.csh podréis encontrar un fichero de ejemplo para las shells (csh, tcsh). Este proceso de inicialización de variables de entorno esta ya configurado para que cada vez que un usuario se conecte, sus variables de entorno ($PATH, $MANPTH…) serán actualizadas de forma automática al iniciar la sesión en el *frontend*.

Envío y control de trabajos

Para el envío de trabajos, resumimos en el siguiente esquema la clasificación hecha mas arriba:

1) **Según el modo de ejecución del trabajo:**
   a) Interactivos
   b) No interactivos

2) **Según la duración del trabajo**
   a) Muy corta duración (interactivos)

b) Corta duración

c) Media duración

d) Larga duración

3) **Según la naturaleza del trabajo:**

a) Paralelos

i) MPI

ii) OpenMP

b) Secuenciales

i) Simples

ii) Múltiples

c) De compilación

i) Normales

4) **Según el hardware requerido:**

a) Para trabajos paralelos:

i) Red myrinet (myr)

ii) Red Infiniband (inf)

iii) Red Ethernet (eth)

b) Para trabajos secuenciales:

i) Grupo 1 (g1)

ii) Grupo 2 (g2)

iii) Grupo 3 (g3)

iv) Grupo 4 (g5)

v) Grupo 5 (g5)

vi) Grupo 6 (g6)

vii) Grupo 7 (g7)

viii) Grupo 8 (gslow)

ix) Grupo 9 (gfast)

x) Grupo 10 (ghuge)

Para el envió de los trabajos al sistema de colas, definiremos un **script de lanzamiento** para cada subtipo de trabajo. Este contendrá los parámetros adecuados a cada uno de ellos y el binario (proceso) a ejecutar. Recordar que los envíos en este apartado, al igual que el resto de la gestión y monitorización de los trabajos, los realizaremos en base a comandos proporcionados por **SGE EE** desde la propia shell de sesión abierta al *logearnos* al *servidor*. (frontend o nodos de compilación)

Envío

Dicho esto, y una vez definidos los tipos y subtipos de trabajos, vamos es exponer a continuación el proceso de envió de los mismos según el subtipo de trabajo que sea. Este procedimiento se descompone en tres tareas:

1. Copiar el *script* de lanzamiento del trabajo (ejemplo) y adecuar a nuestras necesidades
2. Lanzar el trabajo ejecutado directamente el script de lanzamiento.

Y ahora, según los subtipos de trabajos definidos previamente, los siguientes pasos a seguir serán los siguientes:

- **Trabajos Paralelos MPI:**

1. Copiar, editar y adecuar el *script* de lanzamiento del trabajo a nuestras necesidades:

```
Shell#:> cp /opt/sgeee/script/run-mpi.sh <path destino>

Shell#:> vi run-mpi.sh

#!/bin/sh
# Script de lanzamiento trabajos MPI
# PARAMETROS GLOBALES

# PARAMETROS SGE-EE
shell_name="/bin/bash"
# nombre del TRABAJO
job_name="<El nombre que queráis>"
#Dirección de correo a la cual sgeee enviara el estado del trabajo
email=jherrero@atc.unican.es
# path relativo en el cual se almacenaran los ficheros de salida (-o) y error (
# e) del trabajo (desde nuestro $HOME)
output_path="<path del output>"
error_path="<path del error>"
# le indicamos al sistema si el trabajo es, en cuanto a su duración, largo,
# corto o interactivo
colas="<short-parallel | short-parallel-{myr,inf,gpu}|
          medium-parallel | medium-parallel-{myr,inf,gpu} |
        -parallel-{myr,inf,gpu} |
          inter-parallel | inter-parallel-{myr,inf,gpu}>"
# Entorno paralelo
parallel_env="<nombre del parallel environment> <numero de procesos del trabajo
                paralelo a ejecutar>"

# ENVIAMOS el TRABAJO AL SISTEMA DE COLAS
################################################################################
sleep 1
qsub << eor
#!/bin/bash
# ..................................................
#$ -S $shell_name
#$ -N $job_name

#$ -o /dev/null
#$ -e /dev/null

#$ -v output_path=$output_path
#$ -v error_path=$error_path
#$ -V

#$ -M $email
# Si quiero que el sistema me notifique por mail …

#$ -m be
# Si NO quiero que el sistema me notifique por mail …
#$ -m n
```

```
#$ -l $colas
#$ -pe $parallel_env
# ..................................................

# PERMISOS SGEEE-AFS
kinit -k -t /home/Keytab/krb5.keytab.$USER $USER
aklog -c ATC.UNICAN.ES -k ATC.UNICAN.ES

# VARIABLES DE ENTORNO PARA EL TRABAJO (las que necesites …)

# Encapsulado del trabajo
mkdir -p /scratch/\$USER/\$JOB_ID
if [ true ]; then

        #------------------------------------------------------
        mpirun[9] -np <num procesos> -nolocal -machinefile
        $HOME/.SGE/jobs_active/\$JOB_ID/machines <full_path_command_line>
        >/scratch/\$USER/\$JOB_ID/…
        #------------------------------------------------------

fi 1>\$output_path\$REQUEST.o\$JOB_ID 2>\$error_path\$REQUEST.e\$JOB_ID

cp -a /scratch/\$USER/\$JOB_ID/* \$output_path
rm -fr /scratch/\$USER/\$JOB_ID/*
##############################################################################
eof
```

2. Lanzar el trabajo usando el comando *qsub*.

```
Shell#:>./run-mpi.sh
```

- **Trabajos Paralelos OpenMP:**

1. Copiar y adecuar el *script* de lanzamiento del trabajo a nuestras necesidades:

```
Shell#:> cp /opt/sgeee/script/run-openmp.sh <path destino>

Shell#:> vi run-openmp.sh

#!/bin/sh
# Script de lanzamiento trabajos OpenMP

# PARAMETROS GLOBALES

# PARAMETROS SGE-EE
shell_name="/bin/bash"
# nombre del TRABAJO
job_name="<El nombre que queráis>"
#Dirección de correo a la cual sgeee enviara el estado del trabajo
email=jherrero@atc.unican.es
# path relativo en el cual se almacenaran los ficheros de salida (-o) y error (
# e) del trabajo (desde nuestro $HOME)
output_path="<path del output>"
error_path="<path del error>"
# le indicamos al sistema si el trabajo es, en cuanto a su duración, largo,
# corto o interactivo y el grupo que describe las características del nodo
colas="<short | short-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          medium   | medium-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          long   | long-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          inter  | inter-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu >"

# Entorno paralelo
parallel_env="mthreads/-g{3,7} <numero de procesos del trabajo paralelo a
ejecutar>"
```

[9] Dependiendo de la red y/o infraestructura MPI que usemos para compilación y ejecución de trabajos paralelos, deberemos actualizar nuestro entorno ($PATH …) para usar un *mpirun* u otro. Se RECOMIENDA usar la infraestructura de MPI de PGICDK, ubicada en /a/compiladores/pgicdk/2011/linux86-64/2011/… y /opt/mpich_myr para trabajos paralelos sobre MYRINET.

```
processors="<numero de procesos del trabajo OpenMP>"

# ENVIAMOS el TRABAJO AL SISTEMA DE COLAS
################################################################################
sleep 1
qsub << eor
#!/bin/bash
# ..................................................
#$ -S $shell_name
#$ -N $job_name

#$ -o /dev/null
#$ -e /dev/null

#$ -v output_path=$output_path
#$ -v error_path=$error_path
#$ -V

#$ -M $email
# Si quiero que el sistema me notifique por mail …
#$ -m be
# Si NO quiero que el sistema me notifique por mail …
#$ -m n

#$ -l $colas
#$ -pe $parallel_env

#$ -v OMP_NUM_THREADS=$processors
#$ -v PARALLEL=$processors
# ..................................................

# PERMISOS SGEEE-AFS
kinit -k -t /home/Keytab/krb5.keytab.$USER $USER
aklog -c ATC.UNICAN.ES -k ATC.UNICAN.ES

# VARIABLES DE ENTORNO PARA EL TRABAJO (las que necesites …)

# Encapsulado del trabajo
#
mkdir -p /scratch/\$USER/\$JOB_ID
if [ true ]; then

        #-------------------------------------------------------
        <full_path_command_line> >/scratch/\$USER/\$JOB_ID/…
        #-------------------------------------------------------

fi 1>\$output_path\$REQUEST.o\$JOB_ID 2>\$error_path\$REQUEST.e\$JOB_ID

cp -a /scratch/\$USER/\$JOB_ID/* \$output_path
rm -fr /scratch/\$USER/\$JOB_ID/*
################################################################################
eof
```

2. Lanzar el trabajo usando el comando *qsub*.

```
Shell#:>./run-openmp.sh
```

- **Trabajos NO Paralelos Múltiples:**

1. Copiar y adecuar el *script* de lanzamiento del trabajo a nuestras necesidades:

```
Shell#:> cp /opt/sgeee/script/run-multiple.sh <path destino>

Shell#:> vi run-multiple.sh

#!/bin/sh
# Script de lanzamiento trabajos Multiples

# PARAMETROS GLOBALES

# PARAMETROS SGE-EE
```

```
shell_name="/bin/bash"
# nombre del TRABAJO
job_name="<El nombre que queráis>"
#Dirección de correo a la cual sgeee enviara el estado del trabajo
email=jherrero@atc.unican.es
# path relativo en el cual se almacenaran los ficheros de salida (-o) y error (
# e) del trabajo (desde nuestro $HOME)
output_path="<path del output>"
error_path="<path del error>"
# le indicamos al sistema si el trabajo es, en cuanto a su duración, largo,
# corto o interactivo y el grupo que describe las características del nodo
colas="<short | short-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          medium   | medium-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          long   | long-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          inter  | inter-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu >"
mem="cantidad de memoria en GB necesaria para ejecutar el job"
# Entorno multiple
processors="<numero de procesos del trabajo Multiple>"

# ENVIAMOS el TRABAJO AL SISTEMA DE COLAS
################################################################################
sleep 1
qsub << eor
#!/bin/bash
# ..................................................
#$ -S $shell_name
#$ -N $job_name

#$ -o /dev/null
#$ -e /dev/null

#$ -v output_path=$output_path
#$ -v error_path=$error_path
#$ -V

#$ -M $email
# Si quiero que el sistema me notifique por mail …
#$ -m be
# Si NO quiero que el sistema me notifique por mail …
#$ -m n

#$ -l $colas
#$ -l h_vmem=$memG
#$ -t 1-$processors
# ..................................................

# PERMISOS SGEEE-AFS
kinit -k -t /home/Keytab/krb5.keytab.$USER $USER
aklog -c ATC.UNICAN.ES -k ATC.UNICAN.ES

# VARIABLES DE ENTORNO PARA EL TRABAJO (las que necesites …)

# Encapsulado del trabajo
#
mkdir -p /scratch/\$USER/\$JOB_ID
if [ true ]; then

        #-------------------------------------------------------
        <full_path_command_line> >/scratch/\$USER/\$JOB_ID/…
        #-------------------------------------------------------

fi 1>\$output_path\$REQUEST.o\$JOB_ID 2>\$error_path\$REQUEST.e\$JOB_ID

cp -a /scratch/\$USER/\$JOB_ID/* \$output_path
rm -fr /scratch/\$USER/\$JOB_ID/*
################################################################################
eof
```

2. Lanzar el trabajo usando el comando *qsub*.

```
Shell#:> ./run-Multiples.sh
```

- **Trabajos NO Paralelos Simples:**

1. Copiar y adecuar el *script* de lanzamiento del trabajo a nuestras necesidades:

```
Shell#:> cp /opt/sgeee/script/run-Simple.sh <path destino>

Shell#:> vi run-Simple.sh

#!/bin/sh
# Script de lanzamiento trabajos Simples (Secuenciales)

# PARAMETROS GLOBALES

# PARAMETROS SGE-EE
shell_name="/bin/bash"
# nombre del TRABAJO
job_name="<El nombre que queráis>"
#Direccion de correo a la cual sgeee enviara el estado del trabajo
email=jherrero@atc.unican.es
# path relativo en el cual se almacenaran los ficheros de salida (-o) y error (
# e) del trabajo (desde nuestro $HOME)
output_path="<path del output>"
error_path="<path del error>"
# le indicamos al sistema si el trabajo es, en cuanto a su duracion, largo,
# corto o interactivo y el grupo que describe las características del nodo
colas="<short | short-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          medium   | medium-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          long   | long-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
          inter  | inter-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu >"
mem="cantidad de memoria en GB necesaria para ejecutar el job"

# ENVIAMOS el TRABAJO AL SISTEMA DE COLAS
################################################################################
qsub << eor
#!/bin/bash
# .....................................................
#$ -S $shell_name
#$ -N $job_name

#$ -o /dev/null
#$ -e /dev/null

#$ -v output_path=$output_path
#$ -v error_path=$error_path
#$ -V

#$ -M $email
# Si quiero que el sistema me notifique por mail …
#$ -m be
# Si NO quiero que el sistema me notifique por mail …
#$ -m n

#$ -l $colas
#$ -l h_vmem=$memG
# .....................................................

# PERMISOS SGEEE-AFS
kinit -k -t /home/Keytab/krb5.keytab.$USER $USER
aklog -c ATC.UNICAN.ES -k ATC.UNICAN.ES

# VARIABLES DE ENTORNO PARA EL TRABAJO (las que necesites …)

# Encapsulado del trabajo
#
mkdir -p /scratch/\$USER/\$JOB_ID
if [ true ]; then

        #-------------------------------------------------------
        <full_path_command_line> >/scratch/\$USER/\$JOB_ID/…
        #-------------------------------------------------------

fi 1>\$output_path\$REQUEST.o\$JOB_ID 2>\$error_path\$REQUEST.e\$JOB_ID
```

```
cp -a /scratch/\$USER/\$JOB_ID/* \$output_path
rm -fr /scratch/\$USER/\$JOB_ID/*
##############################################################################
eof
```

2. Lanzar el trabajo usando el comando *qsub*.

```
Shell#:>./run-Simple.sh
```

Para los trabajos interactivos, no especificaremos un *script* de lanzamiento, sino que lanzaremos una sesión (shell) interactiva a uno de los nodos (seleccionado por el sistema de colas en base a las reglas de su propio *scheduller)* a través de **SGE EE**. Una vez lanzada, la shell quedará supeditada al propio sistema de colas, quien controlará su ejecución. Desde esta shell ejecutaremos el trabajo de forma interactiva.

Distinguiremos **2 tipos** de trabajos interactivos:

- Trabajos interactivos de **ejecución guiada (interactiva):**
  - Secuenciales
  - Paralelos
- Trabajos interactivos y de **compilación.**

La ejecución de trabajos interactivos no deberá durar más de 30 minutos para los trabajos interactivos de ejecución guiada. Para los trabajos de compilación, no habrá limite de tiempo. El comando para lanzar sesiones interactivas contra el clúster a través de **SGE EE** será el siguiente:

- Para trabajos secuenciales:

```
Shell#:> qrsh [-cwd] -l inter/-g1/g2/g3/g4/g5/g6/g7/gfast/ghuge/gpu
```

- Para trabajos paralelos:

```
Shell#:> qrsh [-cwd] -l inter-parallel-{myr,inf,gpu}
```

- Para trabajos de compilación:
  - Para nodos con herramientas GNU:

    ```
    Shell#:> qrsh [-cwd] -l <comp|comp-g1|comp-g2>
    ```

  - Para nodo con herramientas Pathscale:

    ```
    Shell#:> qrsh [-cwd] -l comp-ps
    ```

  - Para nodo con herramientas PGI CDK:

    ```
    Shell#:> qrsh [-cwd] -l comp-pgi
    ```

Debemos tener en cuenta que el lanzamiento de trabajos interactivos de cualquier tipo deberemos hacerlo desde nuestro $HOME, nunca desde directorio mas anidados.

Es posible especificar más parámetros para todos los modos de envío, pero eso queda a elección del propio usuario. Para más información al respecto. Ejecutar:

```
Shell#:> man qsub
```

Control del trabajo

El control de trabajos engloba la modificación de algún parámetro SGE del trabajo, el borrado y la suspensión-reanudación manual por parte del usuario.

Para la modificación de un trabajo, éste ha de estar en estado "pendiente" o "qw" y no haber pasado a ejecución. Para ello ejecutar:

```
Shell#:> qalter <id del trabajo> <parámetro a cambiar> …
```

Para más información, acceder al manual de *qalter* (man).

Para la suspensión-reanudación de un trabajo en ejecución-suspensión, ejecutaremos:

- Suspensión:

```
Shell#:> qmod –s <id del trabajo>
```

- Reanudación:

```
Shell#:> qmod –us <id del trabajo>
```

Para el *re-scheduling* del trabajo, ejecutar:

```
Shell#:> qmod –r <id del trabajo>
```

Para eliminar un trabajo del sistema (matando sus procesos), ejecutar:

```
Shell#:> qdel [-f] <id del trabajo>
```

Monitorización del trabajo

La monitorización de los estados de los trabajos que mantenemos en el sistema de colas (y por consiguiente, en el propio clúster), así como la monitorización de los recursos que consumen y el estado de la máquina son de vital importancia para mantener un seguimiento de los mismos, así como para averiguar en determinados casos por que el trabajo esta atascado, muere o ha tenido unos resultados no satisfactorios.

Para ello, en primer lugar, mostraremos un conjunto de comandos que **SGE EE** pone a disposición del usuario, así como su modo básico de uso:

Monitorización del estado de los trabajos en las colas:

```
Shell#:> qstat –u <mi nombre de usuario>
```

Esta opción muestra un resumen de los trabajos de un usuario dado que están dentro del sistema de colas, en algún estado.

```
Shell#:> qstat –f
```

Con esta otra opción vemos en que colas están cargados los trabajos para todos los usuarios.

```
Shell#:> qstat –F
```

Muestra el estado de todos los parámetros de cada cola. Esto nos indica cuales son, por ejemplo, las tasas de carga, usos de memoria, swap etc.

```
Shell#:> qstat –j <id del trabajo>
```

Esta nos muestra los datos y valores de las variables SGE para un trabajo dado.

También es posible listar los recursos "consumibles" del clúster, así como los entornos paralelos disponibles, en ambos casos, para su uso en el envío de trabajos al clúster:

```
Shell#:> qconf –sc
Shell#:> qconf –spl
```

También es posible monitorizar el estado del clúster nodo a nodo:

```
Shell#:> qhost

Shell#:> qhost -j
```

Y realizar un "accounting" del clúster a nivel de sistema de colas.

```
Shell#:> qacct –q -h
```

Este comando ofrece una abundante información de "accounting", la cual se puede formatear con mas parámetros del comando.

Se ha desarrollado un nuevo y útil comando para SGE. Este comando puede mostrar, para un usuario determinado, los procesos UNIX (PID) asociados a cada uno de sus trabajos:

```
Shell#:> qps.sh <nombre_usuario> [<nombre nodo> | "<nombre nodo1> <nombre nodo2> …"] [ext]
```

Para mas información acerca de todos estos comandos, ver el manual "on shell" de los mismos, o la documentación asociada a SGE EE, como el manual de referencia, disponibles en el siguiente directorio del clúster:

/s/documentación/manuales/sgeee

Existen otras herramientas de monitorización aparte de las que aporta el sistema de colas. Entre ellas, destacaremos las siguientes por su fácil usabilidad y manejabilidad:

- **Ganglia:**

Esta herramienta nos permitirá monitorizar todos los parámetros de carga de los recursos computacionales del clúster, tanto del *frontend*, nodos de cómputo y clúster global.

Para acceder a ella, abriremos una ventana de nuestro navegador, y accederemos a la siguiente dirección:

http://ojito.atc.unican.es

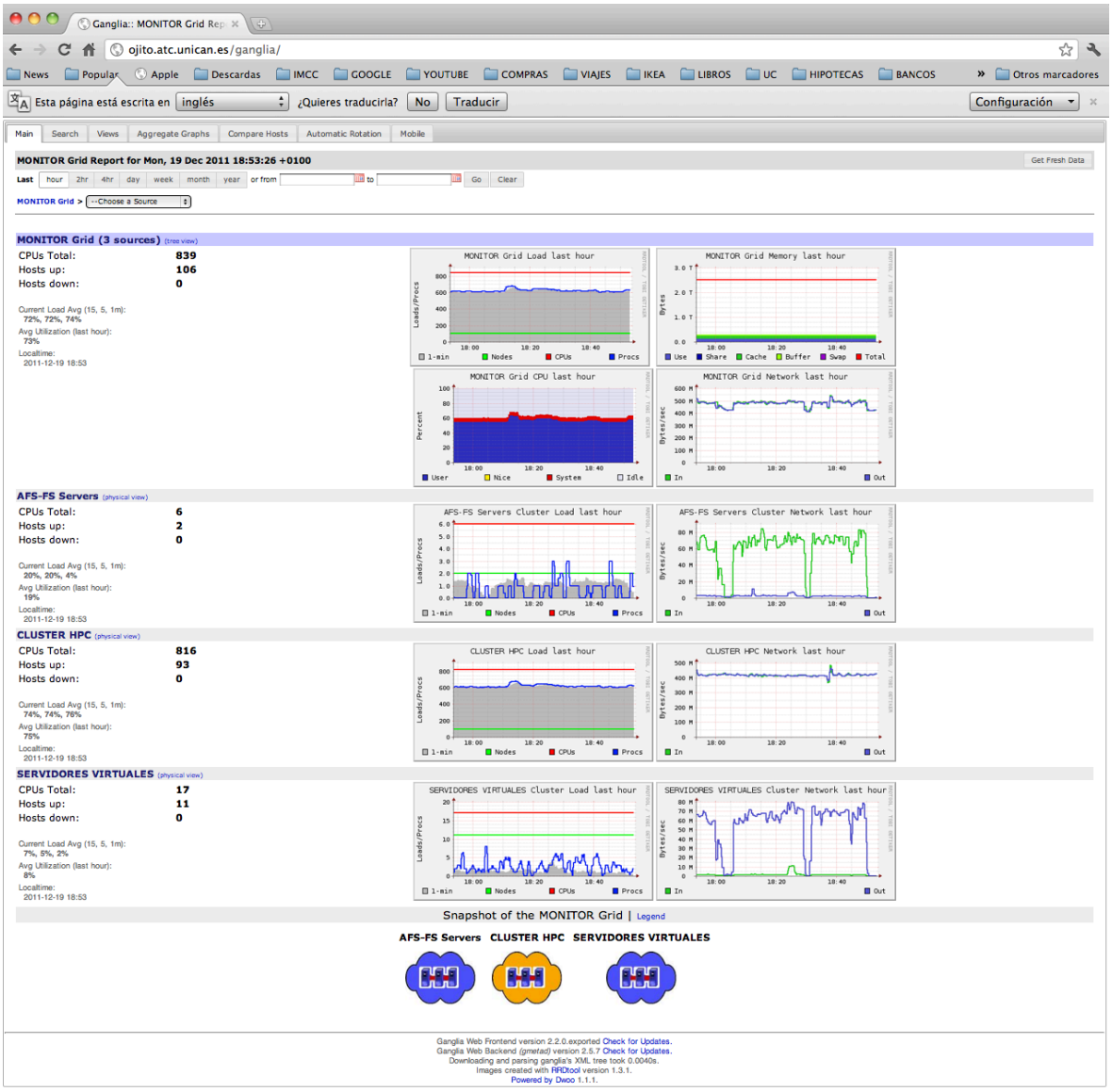


Ilustración 3: Ganglia I

Ilustración 4: Ganglia II

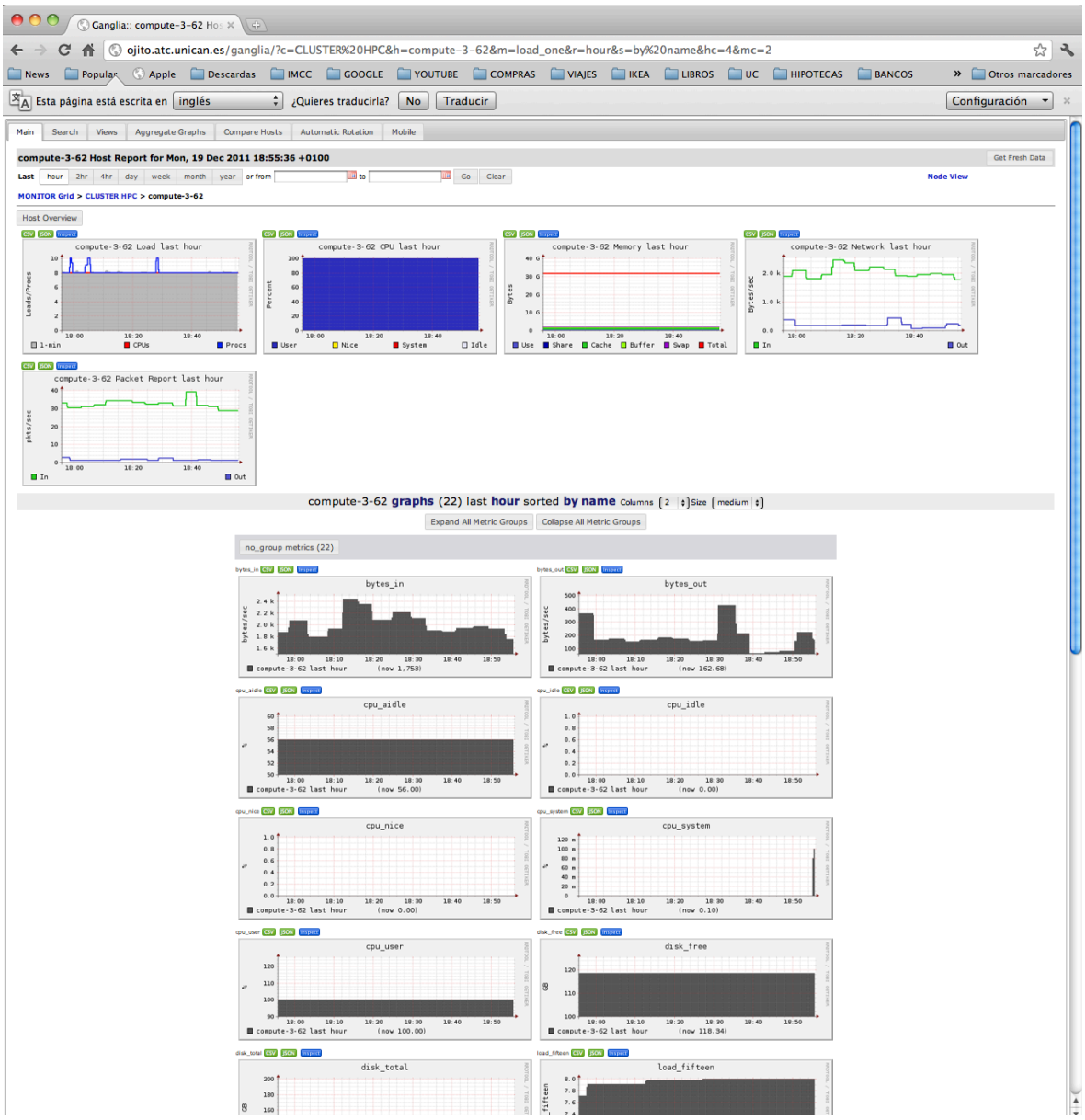


Ilustración 5: Ganglia III

- **C3-tools:**

Las C3-tools son un conjunto de comandos Unix para clúster HPC que, entre otras funcionalidades, pueden lanzar ejecuciones de comandos a todos los nodos de manera automática. Por ejemplo, podemos ver la carga de los nodos del clúster ejecutando:

```
Shell#:> /opt/c3-4/cexecs uptime
************************* local *************************
--------- compute-0-0.local---------
 17:33:47 up 34 days,  6:04,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-1.local---------
 17:26:22 up 34 days,  5:45,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-2.local---------
 17:26:26 up 34 days,  5:45,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-3.local---------
 17:26:38 up 34 days,  5:45,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-4.local---------
 17:33:11 up 34 days,  6:02,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-5.local---------
 17:26:25 up 34 days,  5:45,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-6.local---------
 17:26:04 up 34 days,  5:44,  0 users,  load average: 0.00, 0.00, 0.00
```

```
--------- compute-0-7.local---------
 17:32:42 up 34 days,  6:01,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-8.local---------
 17:33:20 up 34 days,  6:02,  0 users,  load average: 0.00, 0.00, 0.00
--------- compute-0-9.local---------
 17:33:01 up 34 days,  6:02,  0 users,  load average: 0.00, 0.00, 0.00

…
```

Existen más comandos. Están ubicados en /opt/c3-4/. Además de monitorizar, pueden servir también, por ejemplo, para matar procesos que han quedado colgados en el clúster y el sistema de colas ha perdido su control:

```
Shell#:> /opt/c3-4/cexecs "killall -9 <nombre de procesos>"
```

- **top:**

Este comando Unix es también muy utilizado para monitorizar procesos, uso de recursos de CPU, memoria, swap y cargas computacionales.

```
Shell#:> top
top - 17:26:20 up 56 days,  7:27,  7 users,  load average: 0.01, 0.02, 0.00
Tasks: 114 total,   1 running, 113 sleeping,   0 stopped,   0 zombie
Cpu0  :  0.8% us,  0.1% sy,  0.0% ni, 99.1% id,  0.0% wa,  0.0% hi,  0.0% si
Cpu1  :  0.2% us,  0.0% sy,  0.0% ni, 99.8% id,  0.0% wa,  0.0% hi,  0.0% si
Cpu2  :  0.6% us,  0.1% sy,  0.0% ni, 98.8% id,  0.4% wa,  0.0% hi,  0.1% si
Cpu3  :  1.1% us,  0.0% sy,  0.0% ni, 98.9% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:   4934924k total,  1708596k used,  3226328k free,     3524k buffers
Swap:  2056296k total,      292k used,  2056004k free,   534880k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
22348 root      16   0  5044 3612 2852 S  0.9  0.1  39:56.84 sge_schedd
    1 root      16   0  1508  552 1352 S  0.0  0.0   0:08.37 init
    2 root      RT   0     0    0    0 S  0.0  0.0   0:14.96 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.56 ksoftirqd/0
      …
```

Entorno gráfico

SGE EE Suite proporciona diferentes herramientas gráficas, no imprescindibles, pero interesantes para no tener que memorizase los comandos de envió, cambio o monitorización de trabajos. Al comenzar a utilizar el sistema de colas pueden resultar útiles, aunque a largo plazo la línea de comandos es mucho más rápida.

El comando para abrir la ventana gráfica principal de gestión de **SGE EE** es "qmon". Antes de ello, debemos habilitar las X a través de túneles ssh al logearnos al servidor y abrir las conexiones X en nuestra maquina local si es un Linux, o abrir un servidor X (Exeed) si es un Windows.

```
Shell#:> ssh -X calderón
Shell#:> qmon &
```

Este comando abre la ventana gráfica que, agrupa a su vez, el conjunto de herramientas gráficas que permiten el envío, manipulación, configuración y monitorización de trabajos, colas etc. del sistema de colas.

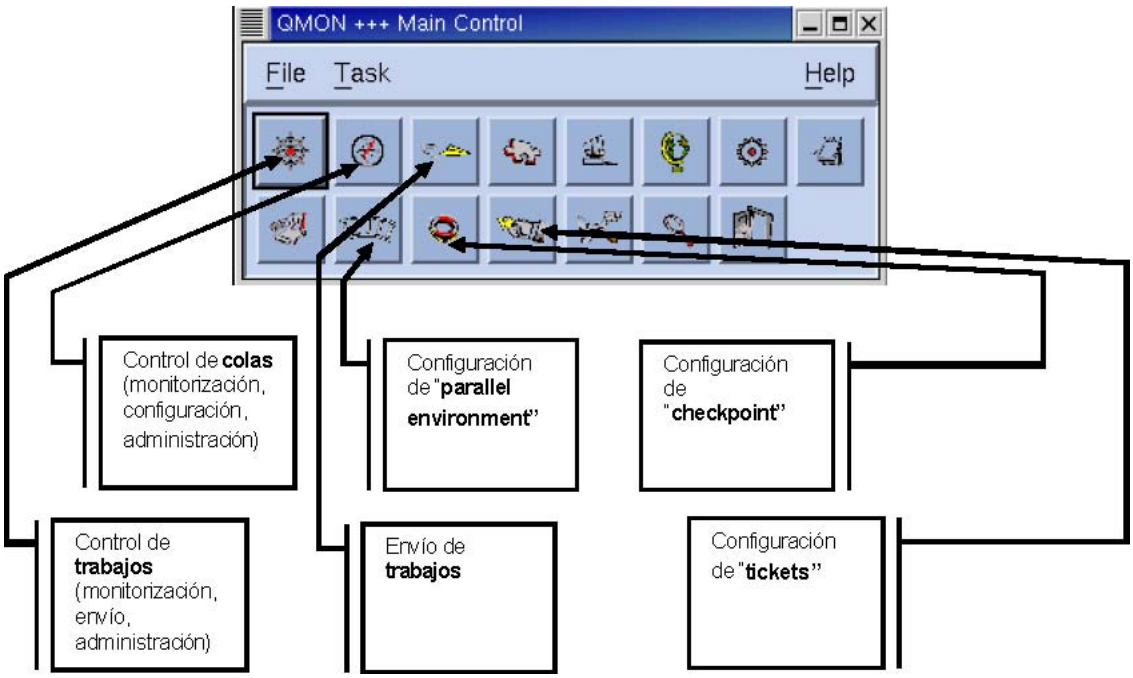


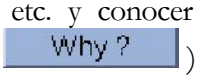
Ilustración 6 Centro de control (Gráfico)

Como se puede observar, todo queda muy centralizado. Lo más cómodo es que, si el usuario decide operar en modo gráfico, solo lance en línea de comandos `qmon`, y a partir de él, el resto de aplicaciones. También es posible acceder a la herramienta de administración gráfica, aunque aquí solo el administrador tiene permisos para realizar modificaciones.

A continuación, se presentan las aplicaciones reseñadas en el la ilustración 6 que afecta directamente al usuario.

- Pulsaremos Para acceder a la interface que controla el envío, modificación, configuración y monitorización de los trabajos. Aparecerá la ilustración 7:

Ilustración 7

Desde esta pantalla podemos fundamentalmente, monitorizar los trabajos según su estado pendiente, ejecutando y finalizados. Además, a partir de aquí se puede manipular el estado de un trabajo, (suspender, reanudar la ejecución de un trabajo suspendido, borrar y relanzar al squeduler). También se puede manipular el trabajo pendiente en todos sus parámetros (qalter), como el cambio de prioridad, salida de datos, etc. y conocer las razones de su estado suspendido/pendiente (seleccionar trabajo y pulsar 

).

Además, también es posible acceder a la pantalla de configuración y envío de trabajos pulsando el botón

.

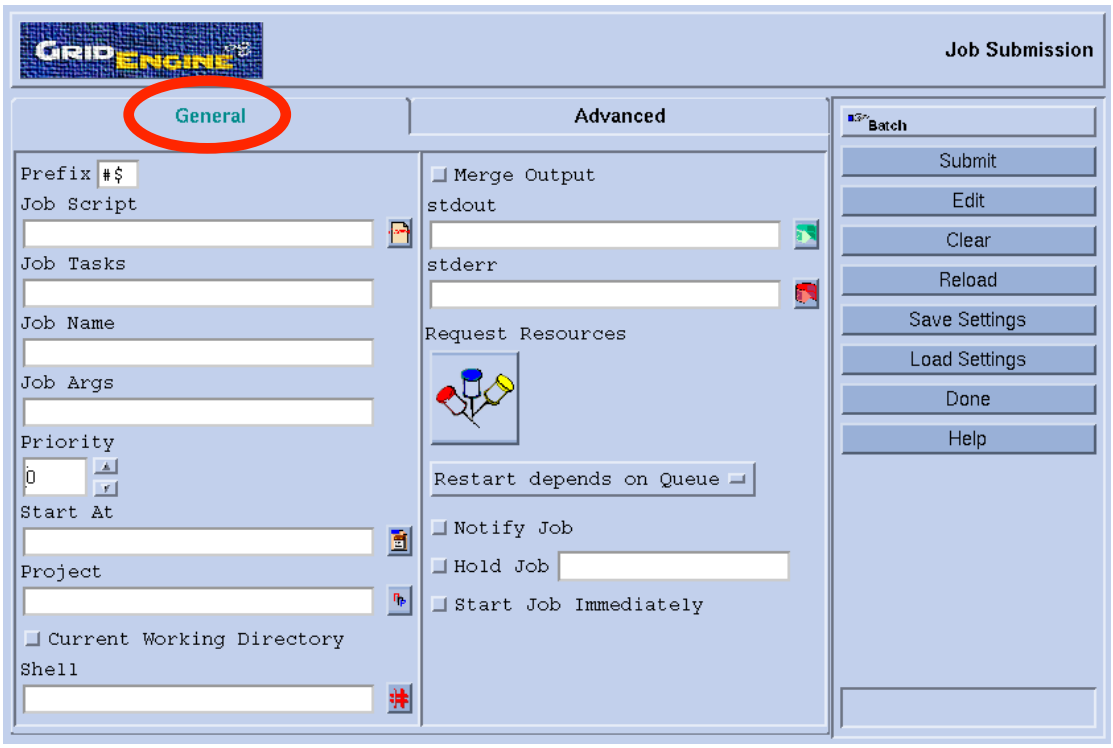


Ilustración 8 Envío de de trabajos al sistema de colas I

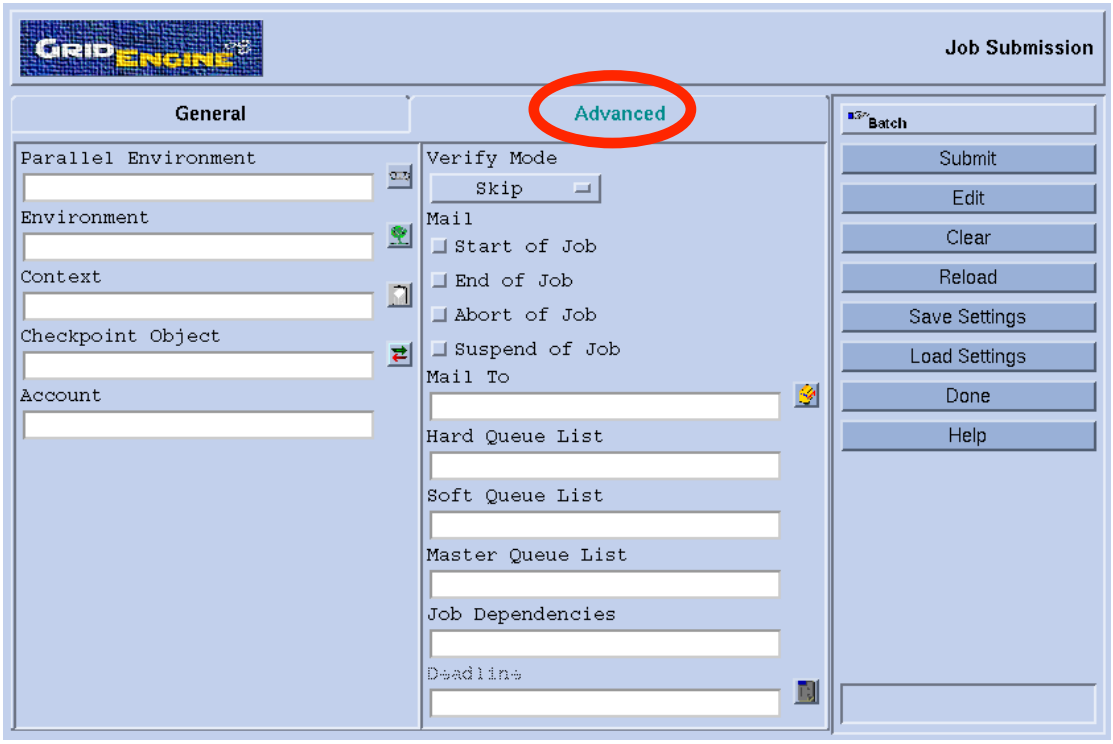
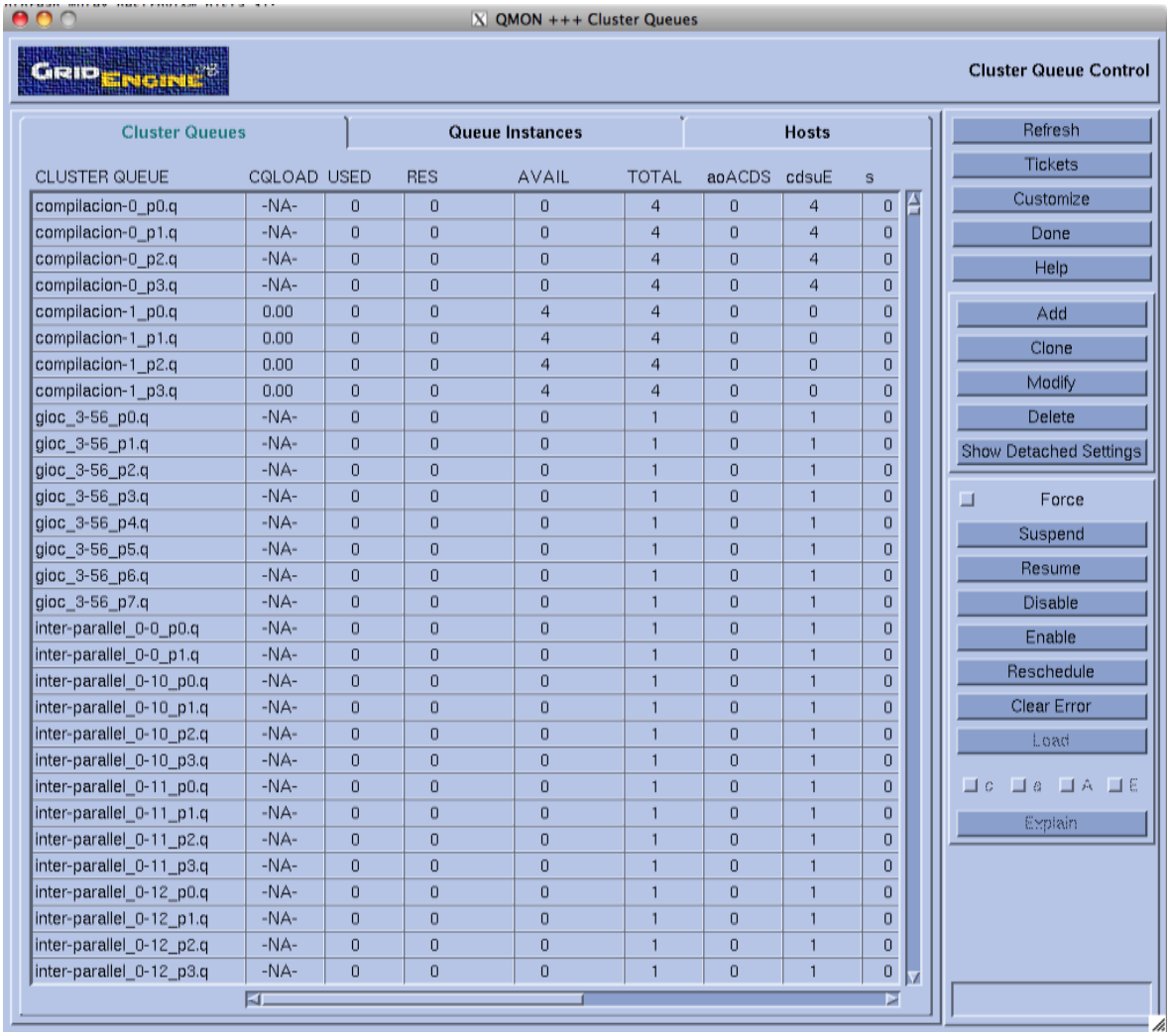


Ilustración 9 Envío de de trabajos al sistema de colas II

Es esta pantalla, el usuario podrá configurar y enviar un trabajo en todos su parámetros, como nombre del trabajo (path completo), parámetros del mismo, prioridad del trabajo (por ahora deshabilitada), shell de ejecución, recursos que el trabajo necesita etc. También aparecen las opciones de salida. Podemos volcar la salida de nuestro trabajo a fichero (opción por defecto) o mail. Estas salidas corresponden a la salida de error y la salida estándar. En el menú "Advanced" también aparece la opción de "parallel environment" para trabajos paralelos con MPI, OpenMP y PVM. Esta opción en particular sirve para especificar, en función del "parallel

environment" especificado, el entorno de ejecución de nuestro trabajo paralelo. En adelante, se configurarán varios entornos de este tipo que caracterizarán los trabajos paralelos en función de sus necesidades. El usuario, una vez seleccionado el "parallel environment" (mpich para trabajos paralelos con MPI, mthreads para trabajos paralelos con OpenMP…), deberá introducir las variables de entorno necesarias que su trabajo demande.

Para la monitorización de las colas, pulsamos y aparece la siguiente pantalla.



CLUSTER QUEUE	CQLOAD	USED	RES	AVAIL	TOTAL	aoACDS	cdsuE	s
compilacion-0_p0.q	-NA-	0	0	0	4	0	4	0
compilacion-0_p1.q	-NA-	0	0	0	4	0	4	0
compilacion-0_p2.q	-NA-	0	0	0	4	0	4	0
compilacion-0_p3.q	-NA-	0	0	0	4	0	4	0
compilacion-1_p0.q	0.00	0	0	4	4	0	0	0
compilacion-1_p1.q	0.00	0	0	4	4	0	0	0
compilacion-1_p2.q	0.00	0	0	4	4	0	0	0
compilacion-1_p3.q	0.00	0	0	4	4	0	0	0
gioc_3-56_p0.q	-NA-	0	0	0	1	0	1	0
gioc_3-56_p1.q	-NA-	0	0	0	1	0	1	0
gioc_3-56_p2.q	-NA-	0	0	0	1	0	1	0
gioc_3-56_p3.q	-NA-	0	0	0	1	0	1	0
gioc_3-56_p4.q	-NA-	0	0	0	1	0	1	0
gioc_3-56_p5.q	-NA-	0	0	0	1	0	1	0
gioc_3-56_p6.q	-NA-	0	0	0	1	0	1	0
gioc_3-56_p7.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-0_p0.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-0_p1.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-10_p0.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-10_p1.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-10_p2.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-10_p3.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-11_p0.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-11_p1.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-11_p2.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-11_p3.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-12_p0.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-12_p1.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-12_p2.q	-NA-	0	0	0	1	0	1	0
inter-parallel_0-12_p3.q	-NA-	0	0	0	1	0	1	0

Ilustración 10 Monitorización de Colas

En esta pantalla se pueden ver la colección de colas que componen es sistema de colas **SGE EE** en nuestro clúster calderon.

Desde el punto de vista del usuario, esta pantalla solo permite monitorizar el estado de las colas. El resto de opciones de control y configuración solo son accesibles (a cambios) por el administrador.

Una opción si permitida es la visualización de la configuración de las colas. Para ello, pulsando

se nos mostrara la siguiente pantalla.

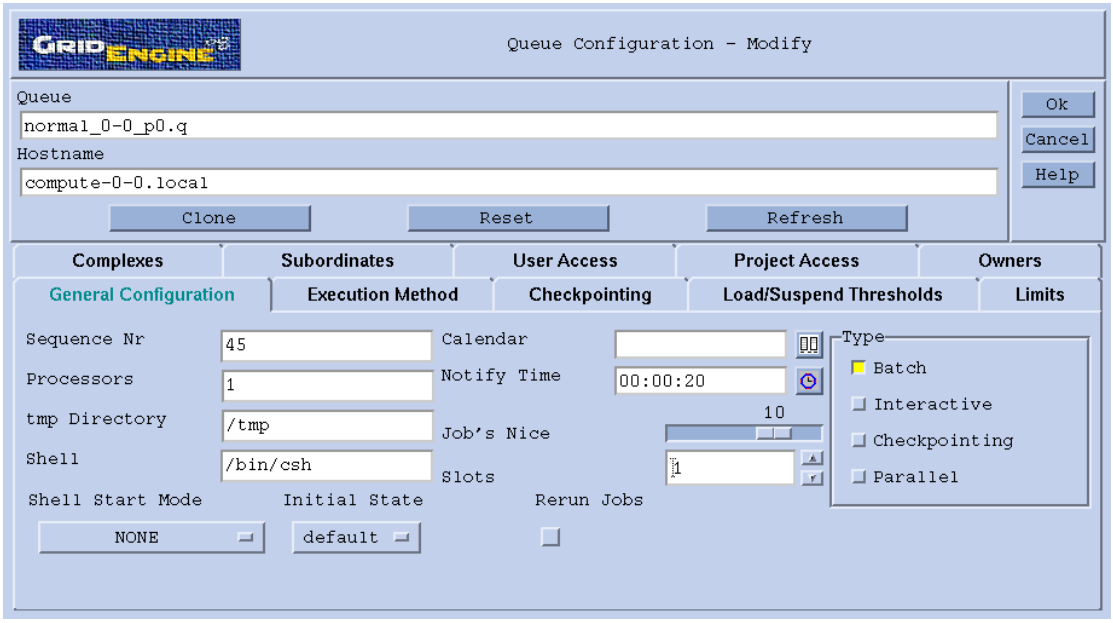


Ilustración 11 Propiedades de una cola (límites y características)

Navegando por las pestañas de la pantalla se podrán ver todos los parámetros de configuración de la cola. El conocimiento de la configuración de las colas, para el caso del clúster, es meramente informativo y carece de interés para la utilización del sistema, puesto que los trabajos YA NO se envían directamente a una cola dada.

Y estas son las opciones gráficas más interesantes de esta potente herramienta que nos proporciona **SGE EE.**

Como siempre, para mas información, consultar el manual "on shell" o la documentación "on line" recogida de **SGE EE** en https://www.ce.unican.es/resource/computing/

Capítulo

5

Recursos y PE disponibles

Ahora enumeraremos los recursos disponible y las "parallel environment" necesarios para el envío de trabajos en **SGE EE**.

Recursos

En cuanto a los recursos, los usuarios deben conocer que para el envío de trabajos de cualquier tipo, han de caracterizar sus trabajos el máximo posible, en base a recursos que van a requerir (consumir) durante su ejecución en el clúster. Existen un montón de ellos, pero para simplificar, comentaremos solamente **3** de carácter fundamental para nosotros:

- **Duración** de los trabajos (tiempo de CPU requerido):

  Este parámetro indicará la ventana de **tiempo de CPU** de que dispondremos para la ejecución de nuestro trabajo en el cluster. Para ello existen 3 tipos de ventana por cada tipo de trabajo (secuencial o paralelo) y según su duración:

  - Trabajos NO paralelos:

    - **long-**g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu: **No** posee límites de tiempo.
    - **medium-**g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu: Los trabajos no podrán exceder de **72 horas**
    - **short-**g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu: Los trabajos no podrán exceder de **3 horas**
    - **inter-**g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu Los trabajos No podrán exceder de **30 minutos.**

  ❖ Como indicábamos mas arriba, para el envío y ejecución de trabajos, podremos tener en cuenta el segmento del clúster donde queremos ejecutarlos, en función de las diferencias hardware de los nodos de cálculo que lo componen. Por eso, además de indicarle el tiempo de duración del trabajo, le indicaremos cual es dicho segmento donde lo ejecutaremos;

- **[FUERA DE SERVICIO[10]] Grupo 1 (g1):** grupo formado por **9 nodos** con 2 procesadores (2 cores) AMD Optaron 248 a 2.2 GHz y 4 GB de memoria principal cada uno.

- **[FUERA DE SERVICIO[11]] Grupo 2 (g2):** grupo formado por **21 nodos** con 2 procesadores (4 cores) AMD Optaron 265/275 a 1.8/2.2 GHz y 8 GB de memoria principal cada uno.

- **Grupo 3 (g3):** grupo formado por **16 nodos** con 2 procesadores (8 cores) Intel Xeon 5472 a 3 GHz y 16 GB de memoria principal cada uno.

- **Grupo 4 (g4):** grupo formado por **16 nodos** con 2 procesadores (8 cores) Intel Xeon 5472 a 3 GHz y 16/32 GB de memoria principal cada uno.

- **Grupo 6 (g6):** grupo formado por **20 nodos** con procesadores (12 cores) Intel Xeon 5650 a 2,67 GHz y 24/48 GB de memoria principal cada uno.

- **Grupo 7 (g7):** grupo formado por **20 nodos** con procesadores (12 cores) Intel Xeon E5645 a 2,4 GHz y 54 GB de memoria principal cada uno.

- **Grupo 8 (gslow):** grupo formado por los **21 nodos mas lentos** del clúster, con procesadores (4 cores) AMD Optaron 265/275 a 1.8/2.2 con 8 GB de memoria principal cada uno.

- **Grupo 9 (gfast):** grupo formado por los **67 nodos mas rápidos y con mejores prestaciones** de cálculo del clúster, con procesadores (8 y 12 cores) Intel Xeon X5472 a 3 GHz y 16 GB e Intel Xeon X5550, X5650 y E5645 hasta 2,67 y 3 GHz y 32/48/54 GB de memoria principal cada uno.

- **Grupo 10 (ghuge):** grupo formado por los **20 nodos** con **mas capacidad de memoria RAM** del clúster, todos ellos con procesadores (12 cores) Intel Xeon E5645 a 2,4 GHz y 54 GB de memoria principal cada uno.

En caso que esta opción sea indiferente, no indicaremos nada y será el sistema de colas quien decida en qué segmento de los tres se ejecutará el trabajo. Por ejemplo:

1. envío de trabajo de larga duración a grupo 2:

   - … -l long-g2 …

2. envío de trabajo de corta duración a grupo 1:

[10] Dados de Baja DEFINITIVAMENTE como elementos activos de cálculo del clúster.

[11] Dados de Baja como elementos activos de cálculo del clúster. Actualmente usados para tareas de desarrollo y entornos de prueba por los administradores de sistemas del CPD 3Mares. Pueden también SER SOLICITADOS por investigadores asociados al grupo ATC para su uso "temporal" aislado en exclusiva de tareas de cálculo.

- … -l short-g1 …

3. envío de trabajo interactivo (me da igual el segmento):

   - … -l inter …

- Trabajos Paralelos

  - **long-parallel**-{myr,inf,gpu}**:** No posee límites de tiempo.
  - **medium-parallel**-{myr,inf,gpu }**:** Los trabajos no podrán exceder de 72 horas
  - **short-parallel**-{myr,inf,gpu }**:** Los trabajos no podrán exceder de 3 horas
  - **inter-parallel**-{myr,inf,gpu }**:** Los trabajos No podrán exceder de 30 minutos.

❖ Como indicábamos también mas arriba, para el envío y ejecución de trabajos paralelos, debemos tener en cuenta el tipo de red de comunicaciones del clúster donde queremos ejecutarlos. Por eso, además de indicarle el tiempo de duración del trabajo, le indicaremos cual es dicha red;

- **Red Myrinet (myr):** Si el trabajo se compila y ejecuta en los nodos interconectados mediante la red myrinet (16 nodos).
- **Red InfiniBand (inf):** Si el trabajo se compila y ejecuta en los nodos interconectados mediante la red infiniband (24 nodos).
- **Red Ethernet:** Si el trabajo se compila y ejecuta en los nodos interconectados mediante la red Ethernet. Todos los nodos del clúster están interconectados bajo esta red, usada principalmente para tareas de gestión. Para ello, solo deberemos especificar que el trabajo es paralelo, sin indicar ningún tipo de red específica.

También hay que indicar, como ya de dijo más arriba, que para la ejecución de trabajos interactivos, DEBEMOS enviarlos (comando *qrsh*, no *qsub*) pasándole el parámetro "-l inter" para trabajos NO paralelos, "-l inter-parallel" para trabajos paralelos y "-l comp" para los trabajos de compilación. Esta es la única manera de ejecutar trabajos interactivos. Además debemos tener en cuenta que éstos tendrán su correspondiente límite de tiempo de 35 minutos (salvo los de compilación, para los cuales no hay límite de tiempo).

Aunque es posible ampliar este límite, y cualquier otro, siempre y cuando se lo solicitemos al Administrador de Sistemas de ATC o responsable de los servidores.

Este parámetro debemos indicárselo en nuestros envíos de la siguiente manera:

- En línea de comandos como parámetro del comando *qsub*

```
Shell#:> qsub -l <long/-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu | short/-
g1/g2/g3/g4/g5/g6/g7/gfast/ghuge/gpu | inter/-
g1/g2/g3/g4/g5/g6/g7/gfast/ghuge/gpu | long-parallel-{myr,inf,gpu}| short-
parallel-{myr,inf,gpu} | inter-parallel-{myr,inf,gpu}> …
```

```
Shell#:> qrsh -l < inter/-g1/g2/g3/g4/g5/g6/g7/gslow/gfast/ghuge/gpu |
inter-parallel-{myr,inf,gpu}| comp/-ps/> …
```

- Como parámetro del comando *qsub* en el fichero *script* de lanzamiento (como se expuso mas arriba en el envío de trabajos)

```
# le indicamos al sistema si el trabajo es, en cuanto a su duración,
#largo, corto o interactivo
#$ -l <long/-g1/g2/g3/g4/g5/g6/gslow/gfast/ghuge/gpu | short/-
g1/g2/g3/g4/g5/g6/gslow/gfast/ghuge/gpu | inter/-
g1/g2/g3/g4/g5/g6/gslow/gfast/ghuge/gpu | long-parallel-{myr,inf,gpu}|
short-parallel-{myr,inf,gpu}| inter-parallel-{myr,inf,gpu}>
#...
```

- Número de **procesos/cores** a utilizar por nodo:

Este parámetro será útil para trabajos OpenMP y trabajos de hasta 12 procesos en un mismo nodo.

Como en el caso anterior, los podemos hacer de estas dos maneras;

- En línea de comandos como parámetro del comando *qsub*

```
Shell#:> qsub -l ps=<numero de procesos> …
```

- como parámetro del comando *qsub* en el fichero *script* de lanzamiento (como se expuso mas arriba en el envío de trabajos)

```
# Máximo numero de trabajos por nodo de computo
#$ -l ps=<número de procesos>
#...
```

- Cantidad de **memoria** por trabajo:

En este caso indicaremos al sistema la cantidad de memoria estimada que nuestros trabajos van a utilizar. Debemos ser conscientes en este caso que sobrepasado este **límite**, el sistema de colas **matará** el trabajo.

Esta limitación en el uso de memoria es muy interesante para asegurarse de que un nodo de cálculo no "pete" por perdidas de memoria en nuestros trabajos.

- En línea de comandos como parámetro del comando *qsub*

```
Shell#:> qsub -l h_vmem=<cantidad de memoria> …

** por ejemplo: 2G, 102M, 15G
```

- como parámetro del comando *qsub* en el fichero *script* de lanzamiento (como se expuso mas arriba en el envío de trabajos)

```
# Cantidad máxima de memoria a usar por mi trabajo
#$ -l h_vmem=<cantidad de memoria>
#...
```

Limitaciones en el uso de Recursos

Se han establecido un conjunto de limitaciones para los usuarios en el uso de recursos computacionales del clúster con el objetivo de hacer mas equitativa la utilización del clúster. Estas limitaciones intentan garantizar que todos los usuarios activos del sistema tengan disponibilidad de un mínimo de los recursos, independiente de la demanda puntual que exista en un momento dado. Dichas limitaciones, que denominaremos **cuotas de utilización de recursos**, son:

1) Límite de trabajos en ejecución en el TODO clúster (*): 512 trabajos por usuario

2) Límite de trabajos en ejecución en el grupo "long-gfast": 128 trabajos por usuario

3) Límite de trabajos en ejecución en el grupo "médium-gfast": 64 trabajos por usuario

4) Límite de trabajos en ejecución en el grupo "short-gfast": 20 trabajos por usuario

(*) NO existe limitación en el nº de trabajos a enviar al clúster. Esta cuota solo establece un máximo de trabajos en ejecución (no en cola "qw"). Cualquier usuario puede enviar la cantidad de trabajos que estime oportuna. Aunque es aconsejable no enviar mas de 1024 trabajos.

Para evitar los desbordamientos de memoria en los nodos y mantener controlado el uso de este recurso por parte de SGE, éste establece que, cada trabajo enviado al clúster, posea un limite en el uso de memoria RAM. Dicho límite se establece por defecto en 2 GB por trabajo[12]. Sobrepasado el límite, el trabajo es inmediatamente terminado y sus procesos asociados "matados" a través de señal UNIX SIGKILL.

Dicho límite puede ser ampliado por el propio usuario a través del parámetros "h_vmem". Dicho parámetro, indicado en el script de lanzamiento como recurso (opción `#$ -l h_vmem …`) o

[12] Para trabajos simples/Múltiples. Para trabajos MPI y OpenMP, ponerse en contacto con el administrador.

directamente como parámetro de `qsub`, lleva a cabo una *solicitud* de utilización del recurso de memoria RAM mas allá del límite establecido por defecto. Indicando una cantidad mayor a 2GB, si el nodo dispone de memoria suficiente, SGE ejecutará el trabajo bajo ese nuevo límite. De la misma forma, si éste es sobrepasado. SGE eliminará el trabajo de igual forma a la antes descrita.

El personal de soporte se reserva el derecho a "ajustar" los límites indicados y/o añadir nuevas cuotas en el uso de recursos con el fin de garantizar la correcta utilización del clúster en su conjunto.

Parallel environment

Los entornos paralelos son un elementos aportado por el sistema de colas **SGE EE** para la ejecución de **trabajos paralelos**, donde los procesos mantienen una comunicación entre sí a través de uno u otro medio de comunicación (paso de mensajes, variables de memoria compartida).

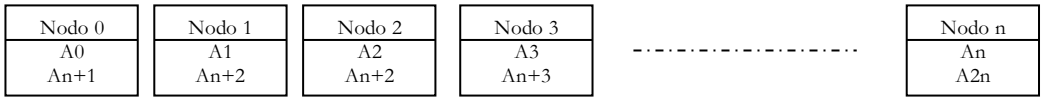
En nuestro clúster se han definido varios *pe*, de los cuales destacaremos 4:

- **mpich:**

  Este entorno paralelo se usará para trabajos MPI, implementados (compilados) con infraestructura PGI MPICH (`/a/compiladores/pgicdk/2011/linux86-64/2011/mpi/mpich`).

  Además, con este entorno paralelo, los procesos serán ubicados mediante el algoritmo clásico *round-robin* en el clúster. Esto significa que los procesos se irán ubicando, de uno en uno, en los nodos. Por ejemplo:

  Un trabajo paralelo A con A0 A1 … A2n procesos, se ubicarán de la siguiente manera:



  Este parámetro debemos indicárselo en nuestros envíos de la siguiente manera:

  - En línea de comandos como parámetro del comando *qsub*

```
Shell#:> qsub -pe mpich < numero de procesos del trabajo paralelo a ejecutar>
```

  - Como parámetro del comando *qsub* en el fichero *script* de lanzamiento (como se expuso mas arriba en el envío de trabajos)

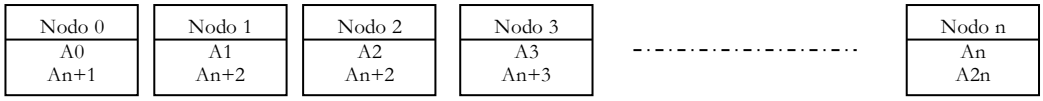
```
# Entorno paralelo
#$ -pe mpich < numero de procesos del trabajo paralelo a ejecutar>
#...
```

- **mpich2:**

  Este entorno paralelo se usará para trabajos MPI2, implementados (compilados) con infraestructura PGI MPICH2 (`/a/compiladores/pgicdk/2011/linux86-64/2011/mpi2/mpich`).

  Además, con este entorno paralelo, los procesos serán ubicados mediante el algoritmo clásico *round-robin* en el clúster. Esto significa que los procesos se irán ubicando, de uno en uno, en los nodos. Por ejemplo:

  Un trabajo paralelo A con A0 A1 … A2n procesos, se ubicarán de la siguiente manera:



Este parámetro debemos indicárselo en nuestros envíos de la siguiente manera:

- En línea de comandos como parámetro del comando *qsub*

```
Shell#:> qsub –pe mpich2 < numero de procesos del trabajo paralelo a ejecutar>
```

- Como parámetro del comando *qsub* en el fichero *script* de lanzamiento (como se expuso mas arriba en el envío de trabajos)

```
# Entorno paralelo
#$ –pe mpich2 < numero de procesos del trabajo paralelo a ejecutar>
#...
```

Es necesario también, para trabajos que usen este entorno paralelo, cargar en el script de lanzamiento estas dos variables de entorno:

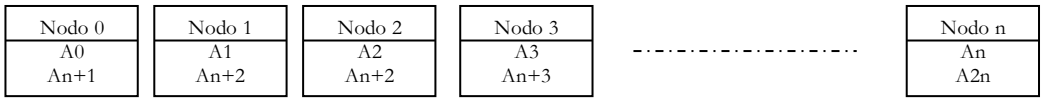
```
# VARIABLES DE ENTORNO PARA EL TRABAJO (las que necesites …
export MPICH2_ROOT="/a/compiladores/pgicdk/2011/linux86-64/11.1/mpi2/mpich"
export MPD_CON_EXT="sge_\$JOB_ID.\$SGE_TASK_ID"
```

- **mvapich:**

Este entorno paralelo se usará para trabajos MPI, implementados (compilados) con infraestructura PGI MVAPICH, que no es mas que una implementación de MPI mpich optimizada para redes **INFINIBAND**.
(`/a/compiladores/pgicdk/2011/linux86-64/2011/mpi/mvapich`).

Además, con este entorno paralelo, los procesos serán ubicados mediante el algoritmo clásico *round-robin* en el clúster. Esto significa que los procesos se irán ubicando, de uno en uno, en los nodos. Por ejemplo:

Un trabajo paralelo A con A0 A1 … A2n procesos, se ubicarán de la siguiente manera:



Este parámetro debemos indicárselo en nuestros envíos de la siguiente manera:

- En línea de comandos como parámetro del comando *qsub*

```
Shell#:> qsub –pe mvapich < numero de procesos del trabajo paralelo a ejecutar>
```

- Como parámetro del comando *qsub* en el fichero *script* de lanzamiento (como se expuso mas arriba en el envío de trabajos)

```
# Entorno paralelo
#$ –pe mvapch < numero de procesos del trabajo paralelo a ejecutar>
#...
```

Es necesario también, para trabajos que usen este entorno paralelo, cargar en el fichero de personalización de Shell del usuario (p.e, `$HOME/.bashrc`) lanzamiento esta variable de entorno:

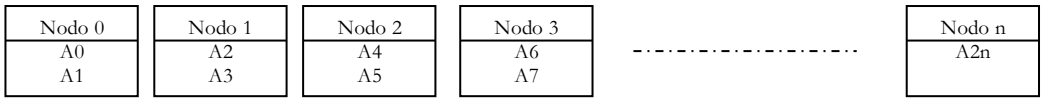
```
export VIADEV_ENABLE_AFFINITY=0
```

- **mthreads (mthreads-g{1,2,3,4,5,6,7,gpu}):**

Este entorno paralelo se usará para trabajos OpenMP.

Además, con este entorno paralelo, los procesos serán ubicados mediante el algoritmo clásico *fill-up* en el clúster. Esto significa que los procesos se irán ubicando hasta llenar cada nodo (recordar que cada nodo tiene 2/4/8/12 procesadores, es decir, capacidad para 2/4/8/12 procesos). Por ejemplo, para los nodos de 2 procesadores (cores):

Un trabajo paralelo A con A0 A1 … A2n procesos, se ubicarán de la siguiente manera:



Este parámetro debemos indicárselo en nuestros envíos de la siguiente manera:

- En línea de comandos como parámetro del comando *qsub*

```
Shell#:> qsub –pe mthreads-g{1,2,3,4,5,6,7,gpu}  <número de procesos del trabajo paralelo a ejecutar>
```

- Como parámetro del comando *qsub* en el fichero *script* de lanzamiento (como se expuso mas arriba en el envío de trabajos)

```
# Entorno paralelo
#$ –pe mthreads-g{1,2,3,4,5,6,7,gpu} <número de procesos del trabajo paralelo a ejecutar>
#...
```

Los identificadores de los grupo ({1,2,3,4,5,6,7,gpu }) mantienen la correspondencia con los grupos definidos para trabajos secuenciales. Ver págs. 16 y 17 de este manual. Es decir, si envío un trabajo indicándole por ejemplo al script de lanzamiento de trabajos:

```
#$ –pe mthreads-g6 <número de procesos del trabajo paralelo a ejecutar>
```

Lo que ocurrirá será que el trabajo "ubique" los procesos o "threads" en alguno de los nodos (lo decidirá el sistema de colas) del **grupo 6**, es decir, uno de los nodos con las siguientes características: nodos con procesadores Intel Xeon E5645 de 6 cores cada uno (12 cores en total) a 2,4 GHz y 54 GB de memoria principal cada uno.

Capítulo

6

Servicios adicionales

En este capitulo describiremos los servicios adicionales a los de cálculo propiamente dicho que ATC ofrece a sus usuarios. Se trata básicamente de dos; Un servicio de copias de seguridad de todos los servidores, tanto a nivel de sistema como de usuario, y un servicio de atención y soporte Web centralizado en un portal web. A continuación haremos una pequeña descripción de ambos .

Sistemas de copias de seguridad

En el capítulo 1, sección 2 ("entorno computacional"), además de describir y enumerar las características hardware y software de los servidores de cálculo de ATC, describíamos los elementos hardware adicionales que proporcionan las comunicaciones (switches) dentro de la sala de cómputo, los sistemas de alimentación ininterrumpidas (SAIs) que mantienen las máquinas arrancadas pese a caídas en las líneas de corriente eléctrica, los sistemas de aire acondicionado, etc … Y también hacíamos mención a un sistema de copias de respaldo (backups) compuesto por 3 elementos fundamentalmente:

1. HP StoreWorks MSL6060 Tape Library: que no es mas que una librería de cintas magnéticas compuesto por un "enclosure", el brazo móvil (robot) que maneja l a posición de las cintas, dos drivers Ultrium 960 que se encargan de la lectura y escritura de las cintas y como no, las propias cintas magnéticas Ultrium 3 con capacidad para 800 GB[13] cada una. Posee una capacidad total de 24 TB de datos. En ATC, disponemos de cintas suficientes para implementar una política de backup con capacidad para 60 TB de datos.

2. HP Proliand DL 145 G2: Servidor donde instalamos el software de gestión de backup y al cual conectamos directamente a la MSL6060.

3. HP OpenView Data Protector: que es el software de gestión de backups que es capaz de realizar backups incrementales, "scheduler" de backup etc … y que aporta la "inteligencia" al sistema.

[13] En modo compresión de datos.

Esto es lo que sería nuestro hardware para soportar las copias de seguridad de nuestros sistemas. Ahora describiremos la política implementada.

Nuestra política actual de gestión de backups en ATC (sección de cálculo) distingue dos grupos:

1. **Grupo 1:** compuesto por los servidores integrados en calderón.

2. **Grupo 2:** compuesto por sistema de discos AFS[14].

Para el grupo 1 dispondremos solo de un nivel de backup. Es decir, que los datos de usuarios de `/home` y sistema se copiarán directamente en las cintas, sin disponer de ningún elemento intermedio (disco o partición auxiliar) donde tengamos de forma temporal los datos.

El tipo de backup que se ejecuta es tanto completo (full) como incremental[15] y la periodicidad con la que se hacen dichos backups en este grupo será:

	calderon
Completo **(anual)**	(a determinar)
Incremental **(semanal)**	Sábado 3:00 horas

Para el grupo 2, dispondremos de 2 niveles de backup:

1. **Nivel 1:** directorio $HOME[16]/.backup. En este directorio tendremos una copia-imagen de "solo lectura" de nuestro directorio de usuario completo en su estado del día anterior. Es decir, todos los días se generará una especie de imagen de todo el directorio de usuario y se almacenará en el directorio oculto .backup del path de usuario. El contenido de .backup solo se podrá leer, nunca borrar o modificar. Esto nos proporcionará un nivel intermedio entre nuestros datos de usuario y los datos almacenados en las cintas. Con esto conseguimos recuperar de manera fácil y rápida cualquier cosas que teníamos en nuestro directorio de usuario 24 horas antes.

2. **Nivel 2:** Cintas magnéticas. Aquí almacenaremos todos los datos de usuarios de la infraestructura.

[14] Ver página 7.

[15] Solo lo almacenan en las cintas los ficheros y directorios nuevos o que han sido modificados desde la última copia. El sistema de Backup dispondrá por lo tanto de versiones (una por cada copia incremental) de nuestros datos.

[16] La variable $HOME indica el path de usuario en la infraestructura de discos AFS. Por ejemplo, para un usuario llamado jherrero, su path de usuario será (en todos los servidores ATC y equipos con cliente OpenAFS instalado) /afs/atc.unican.es/u/j/jherrero. También se crearán directorios .backup para los proyectos, ubicados en /afs/atc.unican.es/p/

Como en el grupo 1, también aquí realizaremos tanto backup completos (full) como incrementales.

	calderon
Completo **(anual)**	(a determinar)
Incremental **(diario)**	Diario 0:00 horas

Tanto en el grupo 1 como en el grupo 2, cuando un usuario requiera recuperar un fichero o directorio (o lo que sea necesario), solo tendrá que ponerse en contacto con el administrador que, en un plazo de tiempo razonable, realizará la operación de recuperación.

Capítulo 7

Conclusiones

Este documento trata de ser una guía fácil, resumida y útil para los usuarios de nuestro clúster HP calderon y en general, de nuestro centro de cálculo. En ella, además de querer recoger un resumen esquemático de las características básicas de nuestros sistemas de cálculo paralelo y del procedimiento y política de accesos remotos a ellos, hemos intentado justificar, caracterizar y describir el sistema de gestión de trabajos por colas usado en este clúster HPC: **SGE EE** (Sun Grid Engine Enterprise Edition), con la información justa para que los usuarios sean capaces de entender la necesidad, complejidad y usabilidad de este sistema.

Espero que resulte interesante para los usuarios. Los que estéis interesados en profundizar más podéis acceder a la documentación de **SGE EE** en la pagina web de soporte de nuestro grupo https://www.ce.unican.es/resource/computing/ . Más de 1000 páginas de documentación os pueden dar una idea de la complejidad y funcionalidades del sistema. Sin embargo, a nivel de usuario considero que este documento puede ser suficiente.