Guía de uso de SUPERDOME

- Actualizado (04.08.2009)

- Conectando
- Uso Interactivo
- Sistemas de colas
- Chequeando o estado dos traballos
- Sistemas de ficheiros
- Compilación
- Documentación adicional

Conectando

Unha vez obtida unha conta de usuario (aqueles usuarios que xa teñen conta activa nos servidores HPC4500 ou HPC320 xa dispoñen do conta neste sistema), cos datos de login e password, se poden conectar ao sistema Superdome. O nome do servidor é sd.cesga.es e para conectarse é necesario dispor dun cliente ssh. Un cliente para Windows (Putty) pode atoparse neste enlace. Utilizando este mesmo protocolo pódense transferir ficheiros cara a e dende o Superdome utilizando un cliente coma WinSCP. Para visualizar a saída gráfica de programas que dispoñan desta posibilidade, é preciso conectarse co comando ssh - X, o que fixará automaticamente a variable de entorno DISPLAY ao equipo dende o que nos esteamos conectando. Para activar esta posibilidade utilizando o cliente Putty, é preciso ir ao menú Connection -> SSH -> Tunnels e activar a opción "Enable X11 forwarding", e ademais será necesario ter instalado e funcionando un cliente X-Windows coma o X-Win32.

Uso Interactivo

O sistema operativo do Superdome é UNIX (HP - UX 11iv2). Unha vez conectados, abrirase unha sesión interactiva a través dun shell que pode ser ksh ou bash. Este shell ou intérprete ten impostos uns límites de tempo de CPU, memoria e disco que se poden consultar mediante o comando ulimit -a. Deste xeito, é necesario utilizar o sistema de colas para a execución de traballos con grandes demandas de recursos computacionais.

Sistema de colas

Para traballos que requiran máis recursos (tempo de cálculo, memoria ou espacio en disco) que os limitados polo shell interactivo, deberase utilizar o sistema de colas. Este sistema é o Sun Grid Engine. O modo de enviar traballos implica coñecer de antemán os valores máximos de cantidade de tempo, número de procesadores, memoria e espacio en disco que vai requirir o cálculo. Esto permite ademais un mellor aproveitamento dos recursos do sistema por parte de tódolos usuarios.

O comando para enviar traballos ao sistema de colas no Superdome é qsub, seguido por unha lista dos recursos que necesita o traballo. Por exemplo, supoñamos que queremos enviar un traballo Gaussian que non precisa máis de 2 gigabytes de memoria e 10 gigabytes de scratch, e estimamos un tempo de execución aproximado non superior a 24 horas, utilizando 1 procesador. Se o ficheiro de entrada para Gaussian chamase script.com e se atopa no directorio "pruebas", a forma de enviar este traballo a cola é:

```
qsub -l num_proc=1,s_rt=24:00:00,s_vmem=2G,h_fsize=10G
cd $HOME/pruebas
g98 < script.com
control+D
```

Se non se produce ningún erro, obteremos unha mensaxe deste tipo:

```
Your job 492 ("STDIN") has been submitted
```

O número 492 é o que se chama identificador de traballo ou JOBID e permítenos identificar o noso traballo dentro do sistema de colas (por exemplo, para utilizar co comando qstat e saber se se está a executar, encolado, etc.). Tamén é importante indicar este número ao facer consultas telefónicas ou por correo electrónico co persoal de sistemas.

A forma de especificar recursos é coa directiva -l, seguida dos recursos que se solicitan separados cunha ",". É imprescindible deixar un espacio en branco entre a opción "-l" e a lista de recursos. Os recursos deben ser:

Recurso

Significado

Unidades

Valor mínimo

Valor máximo

num_proc
Número de CPUs (procesadores) requeridos polo traballo

1
16

s_rt
Máximo cantidade de tempo real que pode durar un traballo
Tempo

300:00:00/num_proc
1000:00:00 (só para num_proc=1)

s_vmem
Cantidade total de memoria RAM requerida polo traballo
Tamaño
 550M
64G

h_fsize
Máximo espacio requerido por un único ficheiro creado polo traballo
Tamaño

100G

Se se precisa utilizar valores superiores aos límites destes recursos ou se precisa dunha priorización dos seus traballos consulte a páxina de Recursos Especiais, alí indicanse os pasos a seguir.
Debemos ter en conta que:
É moi importante deixar un espacio en branco entre a opción "-l" e o seguinte recurso, mentres que despois non deberá haber ningún outro espacio en branco separando os recursos.

1. Estes valores son máximos, polo que non se poderán superar. Esto quere dicir que se cremos que o noso traballo durará unhas 23 horas, debemos poñer como s_rt=24:00:00 para asegurarnos de deixar unha marxe ao traballo. Despois de 24 horas o sistema finalizará o traballo automaticamente, aínda que este non rematase.
2. Canto máis axustados sexan estes valores aos recursos que realmente consume o traballo, maior prioridade para entrar en execución terá o traballo.

Se estes recursos non son suficientes para o traballo, este abortará por falla de recursos e será necesario indicar os valores adecuados. En xeral, recomendamos solicitar recursos o máis próximos, por riba, aos valores que se estimen necesarios. O motivo é que cantos menos recursos se soliciten, con maior prioridade entrará o traballo en execución.
O formato das unidades para os recursos é o seguinte:

Tempo: especifica o período de tempo máximo durante o que se pode utilizar un recurso. O formato é o seguinte: [[horas:]minutos:]segundos, por exemplo:

30:00 son 30 minutos

100:00:00 son 100 horas

1200 son 1200 segundos (20 minutos)

Tamaño: especifica o tamaño máximo en Megabytes. Exprésase na forma enteiro[sufixo]. O sufixo actúa como un multiplicador definido na seguinte táboa:

k
Quilo (1024) bytes

m
Mega (1,048,576) bytes

g
Giga (1,073,741,824) bytes

Resumindo, imos poñer uns exemplos para distintos traballos:
1. Para un traballo que require pouco consumo de memoria e tempo de execución:

qsub -l num_proc=1,s_rt=10:00,s_vmem=100M,h_fsize=100M traballo.sh

2. Traballo que require moito tempo de execución (80 horas) e pouca memoria (é suficiente con 256mb):

qsub -l num_proc=1,s_rt=80:00:00,s_vmem=256M,h_fsize=10M traballo.sh

3. Un traballo con grandes requirimentos de memoria (4 Gigabytes) pero pouco tempo de execución:

qsub -l num_proc=1,s_rt=30:00,s_vmem=4G,h_fsize=10M traballo.sh

4. Un traballo que xera un ficheiro grande (ata 20 Gigabytes) de resultados:

qsub -l num_proc=1,s_rt=30:00:00,s_vmem=500M,h_fsize=20g traballo.sh

5. Un traballo que consume 100 horas de CPU, 2 Gigabytes de memoria e xera un ficheiro de 5 gigabytes:

qsub -l num_proc=1,s_rt=100:00:00,s_vmem=2G,h_fsize=5G traballo.sh

6. Un traballo paralelo con 8 procesadores e 10 horas de tempo de execución total, 8 Gigabytes de memoria total e xera un ficheiro de 10 gigabytes:

qsub -l num_proc=8,s_rt=10:00:00,s_vmem=8G,h_fsize=10G traballo.sh

Unha vez que executamos o comando qsub, e obtemos o identificador para o traballo, este pasa a unha cola apropiada para a súa execución. O traballo esperará a súa vez ou o momento en que estean dispoñibles os recursos solicitados, para pasar a execución, e finalmente o traballo rematará e desaparecerá da cola.

Chequeando o estado dos traballos
Para comprobar o estado en que se atopan os traballos, pódese utilizar o comando qstat

qstat

Obteremos unha saída como a seguinte:

Job id
prior
name
user
state
submit/start
at
Queue

489
0
STDIN
carlosf
r
12/29/2003
19:49:05
Cola1

O significado dos campos é o seguinte:
Job id: 489 é o valor do JOB-ID que lle asignou o sistema de colasSGE. O JobID é un identificador único para cada traballo e permite realizar o seguimento do mesmo.
Prior: Indica a prioridade coa que se está a executar o traballo
Name: STDIN é o nome do traballo que se enviou á cola. Se se enviou un traballo dende a entrada estándar (é dicir, escribindo os comandos ao enviar o traballo), aparecerá STDIN. No caso de ser un script, aparecerá o nome do script.
User: carlosf é o login do usuario que enviou o traballo á cola
State: "r" é o estado no que se atopa o traballo e indica que está en execución (running). Os outros posibles estados dun traballo son:

t: transferíndose o traballo para comezar a súa execución. R indica que o traballo está en execución.

s: suspendido temporalmente para executar traballos máis prioritarios.

w: o traballo está encolado en espera de que haxa suficientes recursos para ser executado ou debido a que se excederon os límites por usuario.

Submit/start at: Data e hora na que o traballo foi enviado á cola ou entrou en execución.
Queue: cola 1 é o nome da cola á que se enviou o traballo. A cola destino dependerá dos recursos que se solicitaran.

Execución de traballos paralelos
O Cluster Integrity Superdome está formado por 2 nodos de 64 CPUs cada un. É posible, polo tanto, utilizar modelos de paralelización de memoria compartida (do estilo OpenMP) para a comunicación dentro dun nodo, coma modelos de memoria distribuída (do tipo MPI) para a comunicación entre-nodos e tamén dentro dun nodo. Nos dous casos, o número de CPUs máximo que se pode utilizar está limitado a 16 e sempre se utilizarán dentro dun mesmo nodo
Sistemas de ficheiros

Existen distintos sistemas de ficheiros con diferentes características en función dos requirimentos de espacio e velocidade de acceso.
Directorio home
É o directorio no que estarán os datos e ficheiros habituais de traballo diario, e do que se fan backups de modo regular. A cantidade de datos por usuario no seu directorio home está limitada a 10 Gbytes. Utilizando o comando "quota -v" cada usuario pode coñecer en calquer momento a cantidade de datos que todavía pode introducir na súa conta.
Sistema de almacenamento
Tódolos usuarios que precisen maiores demandas de almacenamento que as dispoñibles no seu directorio home, deberán solicitar acceso ao sistema de almacenamento masivo, enviando un mail a precisando a cantidade de almacenamento necesario y cal vai ser o seu uso.
Directorio de scratch
É un espacio de almacenamento para datos temporais e que se utiliza en aplicacións coma Gaussian ou Gamess que requiren dun ficheiro grande no que escribe unha grande cantidade de datos de modo continuado. Só é posible acceder a este directorio a través do sistema de colas e coa variable de entorno $TMPDIR. Os datos que se atopen neste directorio desaparecerán ao remate do traballo. Se algún ficheiro contido neste directorio fose necesario, é responsabilidade de cada usuario o copialo ao seu directorio home antes de que remate o traballo.

Utilización do scratch

Para cada traballo que se envía a cola, xérase un directorio no disco scratch local dos nodos. Este disco de scratch é máis rápido que os discos que conteñen o directorio home dos usuarios, polo que os programas que requiren realizar moitas operacións de entrada-saída deberían utilizar este directorio scratch como lugar de traballo. O nome deste directorio varía en cada execución e é accesible mediante a variable de entorno $TMPDIR dentro do traballo. É preciso ter en conta que este directorio se elimina ao concluír a execución dos traballos polo que, se se escribiu nos datos que queiramos conservar, é necesario copialos ao directorio home do usuario.

Nota para os usuarios de Gaussian98:
A instalación de Gaussian98 no CESGA está configurada para utilizar automaticamente este directorio como espacio de scratch, polo que o usuario desta aplicación non necesita configurar a variable de entorno GAUSS_SCRDIR.
Directorio /tmp
Neste directorio de acceso común para tódolos usuarios se poden introducir pequenos ficheiros temporais, aínda que a súa utilización está desaconsellada e o seu contido poderá ser eliminado de forma periódica.
Editores de texto
Ademais do vi, están dispoñibles emacs, xemacs e xedit.
Máis información para a utilización do Gaussian98.
Compilación

Compilador de C/C++

Existen tres compiladores para C/C++:

-
  cc, compila código en C

  c89, compila código en ANSI C

  aCC -o CC, que compila código en C++.

Por defecto, a compilación xera código de 32 bits. Para compilar para 64 bits é necesario incluír a opción +Onivel, sendo nivel igual 0, 1 (opción por defecto), 2, 3 ou 4. A optimización -fast ou -Ofast selecciona unha combinación de opcións para unha execución óptima en velocidade (en concreto, +O2, +Onolimit, +Olibcalls, +Ofltacc=relaxed, +DSnative, +FPD, -Wl,+pi,4M, -Wl,+pd,4M y -Wl,+mergeseg).
Tódolos compiladores de C/C++ soportan OpenMP 2.0 (véxase máis abaixo como activala).
Compilador de Fortran
O compilador existente é o propio de HP. Entre as características principais están:
· Soporte de F77/F90/F95
· Soporte de OpenMP 2.0
· Autoparalelizador
· Xeración de código de 32 bits (por defecto) ou 64 bits
Para compilar un programa é necesario executar o comando

f90 fichero.F

Neste caso xerará código por defecto para 32 bits. Se se quere compilar con soporte para 64 bits, hai que incluír a opción +DD64. Se se quere producir código específico para Itanium2 (opción recomendada) existe a opción +DSItanium2 ou +Dsnative. Por exemplo, para compilar para Itanium2 con soporte para 64 bits hai que executar o comando.

f90 +DD64 +DSitanium2

Optimización
Existen catro niveis de optimización que se seleccionan con +O<nivel> ou -O<nivel>, ademais de poder controlar en cada caso as opcións de compilación coas diferentes opcións +O<opcion>. Ademais, HP Fortran inclúe unha optimización axeitada para moitos programas que se activa con +Ofast. A parte de executar as optimizacións máis apropiadas para a máquina, cambia o tamaño da páxina de datos e de programa a 4 MB, permitindo reducir as perdas de TLBs en programas que consuman memoria ou cun executable grande. Estes parámetros pódense cambiar posteriormente a través do comando chatr.

f90 +Ofast +DD64 entrada.F

Cambio de tamaño de páxina de execución
O procesador Itanium2 inclúe 128 entradas para TLB de tipo TEXT (código do programa) e 128 para DATA (datos). Para unha execución óptima do programa, é posible, en función das necesidades do mesmo, seleccionar o tamaño da páxina que se utilizará para ese programa utilizando o comando chatr coa seguinte sintaxe:
chatr +pd <tamaño páxina datos> +pi <tamaño páxina executable> executable
Os tamaños válidos son 4K, 16K, 64K, 256K, 1M, 4M, 16M, 64M, 256M, 1G, 4G, D, e L, onde D significa "por defecto" e L "a máis grande posible".
Por exemplo, supoñamos que temos que executar un programa que precise 16 Gb de memoria RAM para datos. Para distribuír eses 16GB en só 128 páxinas, precisariamos páxinas de 128MB (que non se poden seleccionar). Para mellorar o resultado do programa poderíanse utilizar páxinas de 256MB utilizando o comando:

chatr +pd 256M <executable>

Utilización de MLIB (BLAS/LAPACK/ScaLAPACK)
HP MLIB contén versións robustas e optimizadas para Itanium2 de tódalas funcións de BLAS 1,2 e 3, BLAS sparse, LAPACK, ScaLAPACCK, SuperLU_DIST, Fast Fourier transforms (FFTs) e convolucións. Inclúe ademais a funcionalidade completa da librería METIS 4.0.1 (http://www-users.cs.umn.edu/~karypis/metis/metis/).
Para compilar un programa que precise estas librerías é preciso incluír as opcións -lveclib -llapack. En función do modelo de datos seleccionado (32 bits ou 64 bits) incluiranse as referencias axeitadas en cada caso ás librerías correspondentes.
Por exemplo, o comando

f90 +DD64 entrada.F -lveclib -llapack

xerará un executable que cargará as librerías (se se utilizan no programa, claro está),

/opt/mlib/lib/hpux64/libveclib.so /opt/mlib/lib/hpux64/liblapack.so

NOTA: para 64 bits, e só cando os enteiros son de 64 bits - Integer*8 en Fortran e long long en C -, pódense tamén referenciar as librerías veclib8 y lapack8. É preferible compilar estas librerías de forma estática, xa que o rendemento é mellor.

Compilación con MPI
HP MPI está baseado nas implementacións do estándar MPICH (de Argonne National Laboratory) e LAM (da Universidade de Nôtre Dame). É totalmente compatible co estándar MMPI 1.2 e inclúe algunhas das funcionalidades de MPI 2.0 soportadas, véxase o manual HP MPI User's Guide, Seventh Edition.
Existen tres comandos que permiten compilar sinxelamente un programa para que precise MPI:

mpicc, que utiliza por defecto o compilador /opt/ansic/bin/cc

mpiCC, que utiliza /opt/aCC/bin/aCC

mpif90, que utiliza /opt/fortran90/bin/f90

Cada un deles chamará ao compilador axeitado en cada caso, coas opcións do compilador que se inclúan a maiores, e incluirá as librerías necesarias de MMPI. Ao igual que en casos anteriores, é preciso indicar se se compila para 32bits ou 64bits.
Para executar a aplicación pódense usar os comandos:

-
mpirun

mpiexec

Por exemplo,

mpirun -np 4 mpihello

Compilación e execución con OpenMP
Tanto o compilador de Fortran como de C soportan a versión 2 de OpenMP. Para compilar un programa que precise utilizar OpenMP é preciso utilizar a opción +Oopenmp. Se o código inclúe directivas para OpenMP e non se queren utilizar, débese utilizar a opción Onoopenmp. Por exemplo, para compilar o programa programa.F con OpenMP utilizarase o comando:

f90 +Oopenmp -o programa programa.F

Por defecto, o número de fíos que se arrinca é igual ao número de procesadores (é dicir, no caso do Superdome 64). Para controlar o número de fíos que se executan, utilizar a variable OMP_NUM_THREADS=<número de hilos>. Por exemplo, na shell ksh no seguinte comando limitará o número de fíos a 16:

export OMP_NUM_THREADS=16

Se se executa en batch, esa variable terá un valor igual ao número de procesadores seleccionados no qsub coa opción -l num_proc.

Documentacion Adicional
Para información máis detallada e manuais de programación, pódese consultar a seguinte páxina web http://doc.cesga.es/superdome/Programming.html, onde tamén se atopa información específica para o procesador Itanium e optimización de aplicacións para iste procesador.