Guía de uso SVG

- Actualizado (31.07.2008)

CONECTANDO

Unha vez obtida unha conta de usuario (aqueles usuarios que xa teñen unha conta activa nos servidores Superdome e HPC320 posúen a mesma conta neste sistema), cos datos de login e password, poderase conectar ao sistema SVG. O nome do servidor é svgd.cesga.es e o modo de conexión é mediante o un cliente ssh. Un cliente para Windows (Putty) pode atoparse neste enlace, no caso de utilizar Linux dispón do comando ssh. Pódense transferir ficheiros cara a e desde o SVG utilizando un cliente Windows coma WinSCP ou o comando scp de Linux. Debe terse en conta que o SVG conta cun firewall que restrinxe as conexións dende/a servidores externos.

Para visualizar a saída gráfica de programas que dispoñan desta posibilidade consulte a sección de FAQ.

USO INTERACTIVO

O sistema operativo do SVG é Linux (baseado en Red Hat Enterprise 4.0). Unha vez dentro do sistema, abrirase unha sesión interactiva a través dun shell que por defecto será bash. Este shell ten impostos uns límites de tempo de CPU, memoria e disco que se poden consultar mediante o comando ulimit -a (actualmente 0,5 horas de cpu e 512MB de memoria). Recoméndase o uso do sistema de colas para a execución de traballos así como o uso dos entornos de compilación para compilar aplicacións ou para executar traballos interactivos.

SISTEMA DE COLAS

Para traballos que requiran máis recursos (tempo de cálculo, memoria ou espazo en disco) que os limitados polo shell interactivo, deberase utilizar o sistema de colas ou os entornos de compilación. 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, arquitectura, memoria e espazo 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 SVG 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 500 megabytes 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 sería:

```
qsub -l num_proc=1,s_rt=24:00:00,s_vmem=512M,h_fsize=10G,arch=32
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 (véxase o apartado Soporte).

A forma de especificar recursos é coa opció -l, seguida dos recursos que se solicitan separados cunha ",". É imprescindible deixar un espazo 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	Numero enteiro	1	16
s_rt				

Tempo real máximo que pode durar un traballo

TEMPO

00:01:00

300:00:00

s_vmem

Cantidade total de memoria RAM requerida polo traballo

TAMAÑO

112M

4G

h_fsize

Máximo espazo requerido por un único ficheiro creado polo traballo

TAMAÑO

1M

120G

arch

Tipo de procesador no que se desexa executar o traballo

Valores posibles: 32, 64, opteron, bw

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.

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 Kilo (1024) bytes
M Mega (1,048,576) bytes
G Giga (1,073,741,824) bytes

Os nodos dispoñibles actualmente no SVG corresponden ós seguintes tipos de máquina:

Intel 32 bits

1G / 2G

Intel 64 bits

Opteron 64 bits

num_proc
1
1
4

s_vmem
1G/2G
2G
4G

s_rt
300:00:00
300:00:00
300:00:00

h_fsize
120G
120G
120G

arch
32 bits
32 bits
64 bits (opteron)

Todos os nodos anteriores dispoñen de rede gigabit e permiten o envío de traballos paralelos utilizando a citada rede. Tamén están dispoñibles nodos PIII con rede Myrynet pero estes quedan reservados para a realización de cálculos paralelos que teñan como obxectivo a realización de probas ou a optimización de algoritmos paralelos.

Debemos ter en conta que:

É moi importante deixar un espazo en branco entre a opción "-l" e o seguinte recurso, mentres que despois non deberá

haber ningún outro espazo en branco separando os recursos.

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 deixarlle unha marxe. Despois de 24 horas o sistema finalizará o traballo automaticamente, aínda que este non rematase.

Canto máis axustados sexan estes valores aos recursos que realmente consume o traballo, maior prioridade para entrar en execución terá este.

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.

Resumindo, imos poñer uns exemplos para distintos traballos:

Para un traballo que require pouco consumo de memoria e tempo de execución:

qsub -l num_proc=1,s_rt=10:00,s_vmem=200M,h_fsize=100M,arch=32 traballo.sh

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,arch=32 traballo.sh

Un traballo con grandes requirimentos de memoria (1 Gigabyte) pero pouco tempo de execución:

qsub -l num_proc=1,s_rt=30:00,s_vmem=1G,h_fsize=10M,arch=32 traballo.sh

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,arch=32 traballo.sh

Un traballo que consume 100 horas de CPU, 1 Gigabyte de memoria e xera un ficheiro de 5 Gigabytes:

qsub -l num_proc=1,s_rt=100:00:00,s_vmem=1G,h_fsize=5G,arch=32 traballo.sh

Se precisa utilizar valores superiores aos límites destes recursos debe realizar unha solicitude tal e como se explica na correspondente FAQ.

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 o momento en que estean dispoñibles os recursos solicitados, para pasar á 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. Obteremos unha saída como a seguinte:

job-ID prior name user state submit/start at queue slots ja-task-ID

1134360 2.98007 mm5-11.sh orballo r 10/26/2006 08:19:09 normal@compute-1-27.local

O significado dos campos é o seguinte:

Job-ID: 489 é o valor do JOB-ID que lle asignou o sistema de colas SGE. 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 desde 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.

s: suspendido temporalmente para executar traballos máis prioritarios.

qw: 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: normal@compute-1-27.local é o nome da cola á que se enviou o traballo. A primeira parte do nome indica o nome da cola do cluster (normal) e a segunda parte o nodo no que está correndo o traballo (compute-1-27.local). A cola destino dependerá dos recursos que se solicitaran.

slots: indica o número de nodos nos que se está executando o traballo. Habitualmente, este número é 1 para traballos secuenciais e maior de 1 para traballos paralelos. No caso de traballos paralelos pódese usar o comando qstat -t para ver o resto dos nodos nos que se está executando o traballo.

SISTEMAS DE ARQUIVOS

Existen distintos sistemas de arquivos con diferentes características en función dos requirimentos de espazo e velocidade de acceso.

Directorio home

É o directorio no que estarán os datos e arquivos habituais de traballo diario, e do que se fan backups de modo regular. Existen cotas (límites na súa utilización), polo que o seu uso deberá ser moderado.

Directorio de scratch

É un espazo de almacenamento para datos temporais e que se utiliza en aplicacións como Gaussian ou Gamess que requiren dun arquivo grande no que escriben gran cantidade de datos de modo continuado. Só é posible acceder a este directorio a través do sistema de colas e o seu nome é $TMPDIR. Os datos que se atopen neste directorio desaparecerán ao finalizar o traballo. Se algún arquivo contido neste directorio fose necesario, é responsabilidade de cada usuario copialo ao seu directorio home antes de que remate o traballo ou ben especificar a opción -v COPIA=$HOME/destino cando se lanza o traballo co qsub (véxase FAQ para máis detalles).

Directorio /tmp

Neste directorio de acceso común para tódolos usuarios pódense introducir pequenos arquivos temporais, aínda que a súa utilización está desaconsellada e o seu contido poderá ser eliminado de forma periódica.

Sistema de ficheiros paralelo

Trátase dun espazo de información común para tódolos nodos do clúster e a sua utilización e recomendable en traballos que requiran procesamento masivo de información (data-mining, etc...). Para facer uso deste sistema de ficheros paralelo, deberá seguir os pasos mencionados na guía de servicios de almacenamento.

Directorio compartido

Trátase dun espazo de almacenamento que é visible dende tódolos servidores do CESGA. A súa utilización é recomendable para compartir datos entre distintos servidores ou para cálculos que requiran de información compartida. Para facer uso deste espazo de almacenamento, deberá seguir os pasos mencionados na guía de servicios de almacenamento.

ENTORNOS DE COMPILACIÓN

Moitos dos nosos usuarios utilizan o frontal do SVG para compilar os seus programas antes de os enviar a cola. Isto pode ser un problema tanto pola conseguinte saturación do frontal como para o usuario, xa que por defecto vai compilar nunha maquina de 64bits e ao enviar o programa a cola é posible que se execute nun nodo de 32bits.

Para axudar a solucionar este problema e que o usuario non se encontre coa situación de que tras compilar unha

aplicación esta non corre ben nos nodos, ou que mesmo mostra resultados erróneos, están a disposición dos usuarios os entornos de compilación que permiten xerar un entorno adecuado ás súas necesidades de compilación.

Para iniciar unha sesión de compilación é necesario utilizar o seguinte comando:

compilar -arch <arquitectura>

onde <arquitectura> pode ser:

32: Arquitectura de 32 bits x86_32

64: Arquitectura de 64 bits x86_64

opteron: Opteron

bw: Nodos BW con rede Myrinet

Debe terse en conta que cada sesión de compilación está limitada a 30 minutos.

De calquera forma se ten algunha dúbida sobre a compilación ou o uso deste script pode pórse en contacto connosco en

Compiladores dispoñibles e opcións

Os compiladores GNU estan dispoñibles para compilar en 32 ou 64bits: gcc, g++, g77, gcc4, g++4, gfortran (dependendo do entorno de compilación seleccionado usaránse as versións de 32 ou 64 bits)

Ademais dos compiladores de GNU, están dispoñibles os compiladores de Portland Group, que incluen soporte para as instrucións SSE:

Para contorno de 32 bits: versión 6.1-3: só Fortran (pgf77, pgf90, pgf95 e pghpf)

Para contorno de 64 bits: versión 7.1-2: só Fortran (pgf77, pgf90, pgf95 e pghpf)Está dispoñible un compilador específico para Opteron (pathscale), para a súa utilización é necesario contactar con .

No caso dos compiladores de Portland de 32 bits débese ter en conta que por defecto úsanse os compiladores de Fortran da versión 6.1-3 (pgf77, pgf90, pgf95 e pghpf) pero tamén está dispoñible a versión 6.1-1.

A opción de optimización recomendada para os compiladores de Portland é -fast, que selecciona un conxunto de opcións dirixidas a optimizar o código. Como calquera outra opción de optimización, deberase prestar atención aos resultados obtidos

co código e comprobar que son correctos antes de utilizala en cálculos definitivos.

Os manuais dos compiladores de Portland poden consultarse na páxina web de Portland.

UTILIZANDO MPI

MPI é un inferface de programación paralela para o envío explícito de mensaxes entre procesos paralelos (para o que é necesario ter engadido o código MPI necesario no programa). Para habilitar a utilización de MPI nos programas, é necesario incluír o arquivo de cabeceira de MPI no código fonte e linkar coas librerías de MPI. Ademais, non é posible utilizar códigos interactivos MPI, senón que é obrigatorio utilizar o sistema de colas destes programas.

A maior parte dos nodos do cluster SVG dispón de rede gigabit e permiten o envío de traballos paralelos utilizando a citada rede. Tamén están dispoñibles nodos PIII con rede Myrynet pero estes quedan reservados para a realización de cálculos paralelos que teñan como obxectivo a realización de probas ou a optimización de algoritmos paralelos.

Guía de uso segundo necesidades

Existen distintas versións dispoñibles de mpi:

- Mpi V1 sobre rede Gigabit
- Mpi V1 sobre rede Myrinet
- OpenMpi sobre Gigabit

A continuacion explícase como utilizar cada unha.

MPICH GIGABIT

Engadir ao ~/.bashrc

```
# mpich_p4
export PATH=/opt/cesga/mpich-1.2.7p1_p4/bin:$PAT
```

O script do traballo debe especificar as seguintes opcións do mpirun:

```
#!/bin/bash
export PATH=/opt/cesga/mpich-1.2.7p1_p4/bin:$PATH
export P4_GLOBMEMSIZE=67303416
export P4_SOCKBUFSIZE=65535
```

mpirun -np $NSLOTS -machinefile $TMPDIR/machines /home/usuario/programa

Se o valor de P4_GLOBMEMSIZE é demasiado pequeno a aplicación para,
dando unha mensaxe de erro e suxerindo un valor maior.

Enviar os traballos a cola con opcións similares ás seguintes:

qsub -cwd -l num_proc=1,s_rt=00:05:00,s_vmem=128M,h_fsize=1G,arch=32 -pe gigabit 4 test.sh

OPENMPI GIGABIT

Engadir ao ~/.bashrc

OpenMPI

export PATH=/opt/cesga/openmpi/bin:$PATH

O script do traballo debe especificar as seguintes opcións do mpirun:

#!/bin/bash

Set the environment to the appropiate MPI version

openmpi

export PATH=/opt/cesga/openmpi/bin:$PATH

It is important to give the full path to the program

mpirun -np $NSLOTS -machinefile $TMPDIR/machines /home/usuario/programa

Enviar os traballos a cola con opcións similares ás seguintes:

qsub -cwd -l num_proc=1,s_rt=00:05:00,s_vmem=128M,h_fsize=1G,arch=32 -pe gigabit 4 programa.sh

MPICH MYRINET

O script do traballo debe especificar as seguintes opcións do mpirun:

#!/bin/bash

It is important to give the full path to the program

mpirun -np $NSLOTS /home/usuario/programa

Enviar os traballos a cola con opcións similares ás seguientes:

qsub -cwd -l num_proc=1,s_rt=00:05:00,s_vmem=128M,h_fsize=1G,arch=bw -pe mpi 4 programa.sh

Compilación e linkado

Para os códigos Fortran, é necesario incluír a seguinte directiva no código fonte de calquera código que utilice MPI:

INCLUDE 'mpif.h'

e compilar co seguinte comando:

mpif77 miprograma.f -o miprograma.exe

Para os códigos en C, é necesario utilizar a seguinte directiva:

#include <mpi.h>

e compilar cun comando similar ao seguinte:

mpicc miprograma.c -o miprograma.exe

Para compilar usando Pgf90:

mpif77 -fc=”pgf90 -tp p7 -Msecond_underscore /opt/cesga/mpich-1.2.7p1_p4/cesga/farg.o” miprograma.f -o miprograma.exe

Téñase en contan que só é posible utilizar pgf90 a través do entorno de compilación de 32 bits:

compilar -arch 32

No caso de exixtir algunha duda consultese a guía de utilización do sistema de colas.

UTILIZANDO MATLAB

Matlab execútase dende as colas normalmente, pero para evitar que os traballos que empreguen Matlab fallen ó entrar en execución por falla de licencias, implementamos un novo recurso para o comando qsub: "matlab"

A forma de uso será engadir a liña habitual do qsub ",matlab=1" ó final.

De esta forma estamos pedindo unha licencia libre para executar o noso traballo.

No caso de que a licencia non estea dispoñible nese momento, o traballo non entrará en execución ate que algunha se libere.

SOPORTE

Cómo solicitar axuda:

: para calquera aspecto relacionado cos servidores de cálculo, sistemas de colas, almacenamento, etc…

: en caso de consultas relacionadas co uso de aplicacións, peticións de novas aplicacións, solicitude de axuda na compilación de aplicacións, etc…

Para máis información cosulte o boletín de marzo

.