Guía de uso HPC320
- Actualizado (03.08.2009)

Guía de uso
Conectando
 Uso Interactivo
 Sistemas de colas
 Sistemas de ficheiros
Compilación
 Librerias Matemáticas
 Utilizando MPI
 Utilizando OpenMP

Conectando

Unha vez obtida unha conta de usuario (aqueles usuarios que xa teñen unha conta activa nos servidores hpc4500 poseen a mesma conta neste sistema), cos datos de login e password, poderase conectar ao sistema HPC320. O nome do servidor é sc.cesga.es e o modo de conexión recomendado é ssh (unicamente a versión 2 do protocolo). Para obter clientes para as distintas plataformas véxase www.openssh.org. Un cliente para Windows pódese atopar en este enlace, así como para transferir ficheiros ao estilo ftp.

Uso Interactivo

O sistema operativo do HPC320 é UNIX (Tru64 V5.1A). Unha vez dentro do sistema, abrirase unha sesión interactiva a través dun shell que por defecto será ksh. Este shell ten impostos uns límites de tempo de CPU, memoria e disco que se poden consultar mediante o comando ulimit -a. Desta forma favorécese o uso do sistema de colas para a execución de traballos con gran demanda de recursos.

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. Este sistema é o Sun Grid Engine. O modo de enviar traballos implica coñecer de anteman os valores máximos de cantidade de tempo, número de procesadores, memoria e espazo en disco que vai requerir o cálculo. Isto permite ademais un mellor aproveitamento dos recursos do sistema por parte de todos os usuarios.

O comando para enviar traballos ao sistema de colas no HPC320 é qsub, seguido por unha lista dos recursos que necesita o traballo. Por exemplo, supoñamos que queremos enviar un traballo Gaussian que non necesita máis de 2 gigabytes de memoria e 10 gigabytes de scratch, e estimamos un tempo de execución aproximado non superior a 24 horas, utilizando un procesador. Se o ficheiro de entrada para Gaussian chámase script.com e atópase no directorio "probas", a forma de enviar este traballo á 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 executando, 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 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

1
4

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

360:00:00

s_vmem
Cantidade total de memoria RAM requerida polo traballo
Tamaño

8G

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

16G

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 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.

1. Estes valores son máximos, polo que non se poderán superar. Isto quere dicir que se cremos que o noso traballo durará unhas 23 horas, debemos por como s_rt=24:00:00 para asegurarnos de deixar unha marxe ao traballo. Despois de 24 horas o sistema finalizará o traballo, aínda que este non concluise.
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 achegados, 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 enteira [sufixo]. O sufixo actúa como un multiplicador definido na seguintes táboas:

k

Kilo (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 requerimentos 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 xenera 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 xenera 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 xenera un ficheiro de 10 gigabytes:
   qsub -l num_proc=8,s_rt=10:00:00,s_vmem=8G,h_fsize=10G traballo.sh

Se necesita utilizar valores superiores aos límites destes recursos, débese solicitar o acceso á cola especial, enviando un mail á dirección de correo .

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 o seu turno ou o momento en que estean dispoñibles os recursos solicitados, para pasar á execución, e finalmente o traballo concluirá e desaparecerá da cola.
Chequeando o estado dos traballos:
Para comprobar o estado no que se atopan os traballos, pódese utilizar o comando qstat
qstat
Obteremos unha saida como a seguinte:

Job id

prior

name

user

state

submit/start

at

Queue

master

489

0

STDIN

carlosf

r

12/29/2003

19:49:05

Cola1

MASTER

O significado dos campos é o seguinte:
Job ID: 489 é o valor do JOB-ID que lle asignou o sistema de colas PBS. O JobID é un identificador único para cada traballo e permite realizar o seguimento do mesmo.

Prior: indicar a prioridade con que se está executando 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: transfiríndose 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 polo 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 solicitaron.
Master: indica o host dende o que se enviou o traballo.

Sistemas de ficheiros

Existen distintos sistemas de ficheiros con diferentes características en función dos requerimentos de espazo e velocidade de acceso.
Directorio home
É o directorio no que estarán os datos e ficheiros habituais de traballo diario, e do cal se farán backups de modo regular. Existen cuotas (límites na súa utilización), polo que o seu uso deberá ser moderado.
Sistema de almacenamento
Todos os usuarios dispoñen dun subdirectorio SAHOME dende o cal acceden ao sistema de almacenamento masivo do CESGA. Neste espazo poderán introducir todos os ficheiros e datos que deseen conservar e de utilización menos frecuente. Tamén se realiza backup regular deste sistema. Non deberían utilizarse datos ou ficheiros dentro deste subdirectorio nos traballos que se envían á cola, polo que en caso de necesitárense nun momento dado, deberán moverse primeiramente ao directorio home de traballo habitual.
Directorio de scratch
É un espazo de almacenamento para datos temporais e que se utiliza en aplicacións como Gaussian ou Gamess que requiren dun ficheiro 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 coa variable de contorno $TMPDIR. Os datos que se atopen neste directorio desaparecerán ao finalizar o traballo. Se algún ficheiro contenido neste directorio fose necesario, é responsabilidade de cada usuario o copialo ao seu directorio home antes de que finalice o traballo.
Directorio /tmp
Neste directorio de acceso común para todos os usuarios pódense 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.

Compilación
Compiladores e opcións:
1. Os compiladores de Fortran son f77, f90 e f95
2. O compilador de C é cc
3. O compilador de C++ é cxx
4. A opción de optimación recomendada é -fast, a cal 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.
Autoparalelización cos compiladores KAP
Os preprocesadores de Kuck & Associates para Fortran e C utilízanse para obter as directivas OpenMP que se poden insertar nun código secuencial para habilitar o paralelismo SMP.
Estes optimizadores toman un código Fortran 90, Fortran 77 ou C e automáticamente o paralelizan en aqueles lugares nos que é posible e seguro. Para utilizar os compiladores e paralelizar, é necesario chamalos do seguinte modo:

```
kf90 -fkapargs=-conc prog.f
```

ou

```
kcc -ckapargs=-conc prog.c
```

Nos ficheiros prog.cmp.f ou prog.cmp.c e prog.out aparecerán as modificacións que introduciu o preprocesador no código

orixinal. Nas páxinas man do kf90 e do kcc pódese atopar máis información sobre as distintas opcións dos compiladores.
Para executar o código, é necesario primeiramente fixar a variable de entorno OMP_NUM_THREADS al número de procesadores que se van utilizar (entre 1 e 4). Por exemplo, para utilizar 4 cpus:

```
export OMP_NUM_THREADS=4
```

E a continuación poderemos executar o código como faríamos normalmente.
Para máis información sobre os compiladores KAP e as súas opcións de optimización e paralelización, diríxanse ás páxinas de documentación de Compaq.
Librarías matemáticas
Compaq Extended Math Library (CXML)
Proporcionan un conxunto de subrutinas matemáticas especialmente optimizadas para a plataforma Alpha, inclúe subrutinas nas áreas de:

Alxebra lineal básica (BLAS): inclúe BLAS1, BLAS1E, BLAS1S, BLAS2 y BLAS3
Sistemas de ecuacións lineais e autovalores (LAPACK)
Sistemas de ecuacións lineais dispersas: inclúes métodos directos e interactivos.
Procesado de sinais: inclúes FFTs, transformadas seno/coseno, convolucións, correlacións e filtros dixitais.

A maioría das rutinas inclúen versións para números reais e complexos, ademais de soportar simple e dobre precisión.
CXML tamén proporciona SCIPORT, unha implementación de Compaq Computer Corporation das CRAY SCILIB. Esta libraría proporciona rutinas de 64 bits e precisión simple para portar aplicacións de sistemas CRAY a sistemas Alpha.
Ademais, a libraría CXML inclúe unha versión paralelizada (cxmlp) co modelo de memoria compartida de todas as subrutinas (a versión paralelizada inclúe todas as subrutinas, aínda que só algunhas están paralelizadas, a lista completa de subrutinas que están paralelizadas atópase neste link.
Para compilar e enlazar un programa que contén chamadas ás rutinas CXML, debe utilizar:

f77 mi_programa.f -lcxml o cc mi_programa.c -lcxml
Para utilizar a versión paralela da libraría:

f77 mi_programa.f -lcxmlp
Para saber como executar un programa linkado coa versión paralela da libraría, véxase a sección "executando traballos OpenMP"
A documentación completa da libraría CXML atópase aquí
Compaq Portable Math Library (CPML)
Inclúe unha ampla variedade de funcións matemáticas: funcións trigonométricas, exponenciais, logaritmos, alxebraicas, complexas, etc... Pulse aquí para obter máis información.
Utilizando MPI
MPI é un inferface de programación paralela para o envío explícito de mensaxes entre procesos paralelos (para o cal é necesario ter engadido o código MPI necesario no programa). Para habilitar a utilización de MPI nos programas, é necesario incluir o ficheiro de cabeceira de MPI no código fonte e linkar coas librarías de MPI no momento de linkar.
Compilación e linkado
Para os códigos Fortran, é necesario incluir a seguinte directiva no código fonte de calquera código que utilice MPI:

```
INCLUDE 'mpif.h'
```

e compilar co seguinte comando:

```
f90 miprograma.f -o miprograma.exe -lmpi
```

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

```
#include <mpi.h>
```

e compilar cun comando similar ao seguinte:

```
cc miprograma.c -o miprograma.exe -lmpi
```

Executando códigos MPI
É necesario utilizar a orde dmpirun ao executar calquera código MPI tanto en interactivo como desde el sistema de colas.
Para máis detalles sobre como executar códigos MPI dende o sistema de colas, consúltese a guía de utilización do sistema de colas.

Utilizando OpenMP
OpenMP é un conxunto de extensións aos estándares Fortran, C e C++ para soportar a execución paralela no modelo de memoria compartida. Para a súa utilización é necesario engadir directivas al código fuente que paralelicen los bucles y que especifiquen ciertas propiedades das variables (os compiladores Kap introducen automaticamente este tipo de directivas ao utilizar a opción -conc).
Compilando e linkando
Para compilar código Fortran que inclua directivas de OpenMP:

```
f90 -fast -omp miprograma.f -o miprograma.exe
```

Para compilar código C que inclua directivas de OpenMP:

```
cc -fast -omp miprograma.c -o miprograma.exe
```

Executando traballos OpenMP
Para executar códigos OpenMP é necesario enviar o traballo a cola, dese modo fíxase convenientemente a variable de

contorno OMP_NUM_THREADS ao número de CPU que se solicitaron. Para máis información consulte a utilización do sistema de colas.

Problemas frecuentes con OpenMP

Un dos problemas máis comúns que se atopan despois de paralelizar un código con OpenMP é a xeneración de excepcións de punto flotante ou de violación de segmentos que non se producían antes da paralelización. Este adoita ser un problema debido á utilización de variables sen inicializar