Instituto de Física - UFBA

Cluster Prometeu – Guia do usuário

1 – Características do sistema

O Cluster Prometeu consiste em um sistema HP Blade C7000, com 10 lâminas BL260c, cada uma com a seguinte configuração:

> 2 processadores Xeon E5405 (2,0 GHz, 1333 FSB) com 17 GB de memória RAM e dois discos SATA de 120 GB.

O controle do cluster e armazenamento para os usuários é efetuado por um servidor NAS DL 180 G5, com as seguintes características:

> Processador Xeon E5405 (2,0 GHz, 1333 FSB) com 4 GB de memória RAM e seis discos SATA de 250 GB em raid 5.

O sistema operacional: SUSE 11.0

Bibliotecas de paralelização: MPICH2

Gerenciamento de processos: Sun N1 Grid Engine v6.2. (SGE)

O servidor NAS é rotulado como Prometeu e as lâminas são rotuladas como: no01, no02, ...,no10.

2 – Armazenamento de dados dos usuários

Todas as contas de usuários ficam armazenadas no servidor NAS sendo o diretório *home* exportado via NFS para o restante do sistema sob o rótulo */prometeu-home/*. A área total de armazenamento é de 1,1 TB.

IMPORTANTE: O espaço de armazenamento disponibilizado destina-se a abrigar as rotinas de cálculo e os dados dos usuários, não devendo, entretanto, ser encarado como permanente. O sistema não possui suporte de backup para todos os dados armazenados no computador. Desse modo recomenda-se que o usuário descarregue para o seu computador as informações que deseje armazenar por um longo período.

Em cada lâmina existe ainda um diretório */home/work*, onde todos os usuários têm permissão rwx (leitura, escrita, execução). O espaço disponível é de ~120GB e pode ser usada como área de armazenamento temporário durante a execução de programas.

3 - Política de utilização

(1) Os usuários terão acesso aos recursos computacionais através da solicitação de conta de trabalho em Prometeu. As contas dos Docentes da instituição não terão prazo para expirar. Para os demais usuários a solicitação deverá ser feita com a anuência de um docente responsável e a conta expirará após a conclusão do projeto desenvolvido ou o afastamento do pesquisador da instituição.

(2) A unidade principal, Prometeu, é destinada apenas para o gerenciamento e submissão de tarefas. Assim não é permitido executar tarefas de cálculo nessa maquina. Qualquer tarefa desse tipo detectada será imediatamente removida.

(3) Cada usuário dispõe de 10GB (soft) de espaço no disco principal de Prometeu (/prometeu-home). Ao ultrapassar este limite o usuário receberá o aviso: "*warning, user file quota execeeded*". O limite máximo de espaço em disco, o qual não pode ser ultrapassado (hard) é de 15GB. Quando esse limite for atingido o usuário será notificado com: "*write failed, user file limit reached*".

4 - Acesso ao sistema (login / FTP)

O acesso ao sistema pode ser efetuado utilizando-se um cliente SSH a partir de uma estação Linux ou Windows.

Endereço IP do cluster Prometeu: 192.168.138.11

Em ambiente Linux:

```
ssh –l nome-usuario 192.168.138.11
```

Em ambiente windows pode-se utilizar clientes ssh como o "putty" ou o "Secure Shell Client", dentre outros.

As transferências de arquivos exigem a utilização do protocolo SFTP.

Em ambiente Linux:

```
sftp –l nome-usuario 192.168.138.11
```

Em ambiente windows pode-se utilizar por exemplo o “Secure Shell Client” ou o WS-FTP-PRO.

5 – Compiladores

Compiladores disponíveis no cluster:

Linguagem	GCC (seqüencial)	INTEL (seqüencial)	PARALELO
C	gcc	-	mpicc
C++	c++	-	mpicxx
FORTRAN 77	-	ifort	mpif77
FORTRAN 90	gfortran	ifort	mpif90

6 – A execução de programas no cluster

Os programas serão executados no cluster através de uma “Grade de Computação” (Computational Grid) envolvendo as 10 lâminas instaladas no sistema (nós), como unidades de execução.

De modo simplificado uma *grade* é uma coleção de computadores interligados através de um sistema de gerenciamento e que são capazes de realiza tarefas. Para os usuários a *grade* aparece como um único sistema que disponibiliza muitos recursos. Em geral o acesso a ela se realiza a partir de uma das maquinas do sistema, “unidade principal” (master host), a qual também pode ser atribuída a função de administração e submissão das tarefas (jobs).

Em nosso caso a unidade principal é a maquina Prometeu (192.168.138.11) que também administra o sistema e submete as tarefas.

As unidades destinadas à execução das tarefas são denominadas “unidade de execução” (execution host). Em nosso caso os nós denominados no01, no02, ..., no10.

Todas as tarefas devem ser iniciadas a partir da unidade principal (Prometeu).

As tarefas são submetidas à grade usando-se o comando **qsub**. Este recebe como argumento um “shell script” que contém os comandos que devem ser executados. Pode-se também modificar as características de execução da tarefa através de *chaves* colocadas no *script* ou como argumento do comando **qsub**. Por exemplo, para executar as tarefas em um script nomeado “calculus”, contendo a seguinte instrução

prog.exe < entrada > saida

usamos o comando:

qsub calculos

A grade então iniciará uma nova sessão de login para executar o seu *script*. Isto implica em que o diretório de trabalho para ele será ajustado para o seu diretório *home*. Se o seu script está em um diretório diferente, você deve incluir nele comandos "cd" adequados. Você pode também usar a chave *–cwd* no comando **qsub** para que ele inicie o script a partir do diretório atual, isto é

qsub –cwd calculus

Tarefas também podem ser submetidas interativamente com **qsub**.em vez de utilizar um script

qsub <ENTER>

prog.exe < entrada > saída

<Ctrl-D>

Executando programas paralelizados

A grade SGE utiliza ambientes de paralelização (PE – Parallel Environment), que é configurado pela administração do sistema, para controlar a execução de programas paralelizados. Atualmente Prometeu utiliza apenas um PE para todas as tarefas, este é denominado ***mpich2***.

Para submeter uma tarefa com paralelização devemos utilizar a chave "*-pe mpich2 n*" no comando **qsub**, onde *mpich2* especifica o PE e *n* o numero de processos que serão utilizados.

Exemplo:

Supondo que queremos executar o programa *prog.exe*, podemos criar o script *scrteste* com o seguinte conteúdo:

```
#!/bin/csh -f
# Script scrteste
set MPIR_HOME=/prometeu-home/mpich2/bin
set USER_HOME=/prometeu-home/usuario
$MPIR_HOME/mpiexec -machinefile $USER_HOME/maquinas  -np $NSLOTS $USER_HOME/prog.exe
```

Nota: a variável $NSLOTS será passada através do comando **qsub**.

Executamos então esta tarefa com o comando:

qsub -cwd -N saida -pe mpich2 8 ./scrteste

onde:

-*cwd* significa que o script será iniciado a partir do diretório atual

-*N saida* especifica o nome dos arquivos para a saída padrão (STDOUT) e de erro (STDERR)

-*pe mpich2 8* especifica qual o PE a ser utilizado e o numero de processos ($NSLOTS)

./scrteste é o script que executa a tarefa.

Estes parâmetros podem ser também especificados no script de execução. Nesse caso teríamos:

```
#!/bin/csh –f
# Script scrteste
set MPIR_HOME=/prometeu-home/mpich2/bin
set USER_HOME=/prometeu-home/usuario
#$ -N saida
#$ -pe mpich2 8
$MPIR_HOME/mpiexec -machinefile $USER_HOME/maquinas  -np $NSLOTS $USER_HOME/prog.exe
```

E o comando de execução seria simplesmente:

qsub -cwd ./scrteste

Mais sobre programas paralelizados

Para iniciar processos em paralelo o MPD deve ser iniciado em todas maquinas (nós) com o comando:

mpdboot -n N

onde N é o numero de nós para iniciar o MPD. O valor máximo para N é o numero de nós listados no arquivo *mpd.hosts* + 1.

O arquivo *mpd.hosts* lista os nós do cluster, um por linha. O numero de slots (cores) em cada nó pode ser especificado colocando-o apos o nome do nó, separado pelo caractere ":". Por exemplo

```
no01:8
no02:8
```

especifica os nós *no01* e *no02* com 8 slots respectivamente. Para iniciar o mpd com base em um arquivo com a estrutura acima poderíamos usar

mpdboot -n 3

O mpdboot deve sempre ser acionado a partir da unidade principal (Prometeu).

Para executar um programa utilizamos o comando **mpiexec**:

mpiexec -machinefile /path/maquinas -n M /path/prog.exe

-machinefile permite especificar, em /path/maquinas, o esquema de cpus/slots a serem utilizadas pelos processos. Na falta, todas as cpus carregadas com o comando **mpdboot** poderão ser utilizadas. O arquivo "maquinas" deve listar, uma por linha, a seqüência de cpus que serão utilizadas.

O numero de processos M não pode ser maior do que o numero de cpus listadas no arquivo "maquinas"

Os arquivos mpd.hosts e maquinas podem ter a estrutura descrita abaixo:

mpd.hosts ou *maquinas*	*maquinas*
no01:8 no02:8	no01 . no01 no02 . no02

NOTA: o arquivo "maquinas" é opcional e pode ter qualquer nome, já o *mpd.hosts* é obrigatório e deve estar no nível raiz do diretório do usuário. Se **mpdboot** for executado na forma

mpdboot -n N -f arquivo

arquivo poderá ter qualquer nome e devera ser da forma do mpd.hosts.

-n M Especifica o numero de processos (M) que serão iniciados

/path/prog.exe Caminho completo do executável deve ser utilizado para ser corretamente reconhecido pelos nós, uma vez que eles compartilham uma partição NFS. Em nosso caso todos os usuários estão em /prometeu-home/.

Para finalizar o MPD em todos os nós:

mpdallexit

7 – Alguns parâmetros úteis do comando qsub

A sintaxe do comando **qsub** é a seguinte:

qsub [opções] [comando | -- [argumentos do comando]]

Opções

-@ arqv_opt	Força **qsub** a usar as opções contidas no arquivo arqv_opt
-a data_hora	Define ou redefine a data e hora na qual a partir da qual a tarefa pode ser executada. Data e hora devem ser especificados no formato [[CC]YY]MMDDhhmm[.ss]
-cwd	Executa a tarefa a partir do diretório atual.
-C prefixo	Define o prefixo que declara uma diretiva de comando para a tarefa. O prefixo não é um atributo da tarefa, mas afeta o comportamento de qsub. O prefixo consiste de dois caracteres ASCII os quais, quando aparecem nas duas primeiras posições de uma linha do script, indica que o que segue é um comando do SGE. O default para esta opção é #$.
-l recurso=valor	Inicia a tarefa no SGE satisfazendo a lista de recursos solicitados. Ver no Apêndice I uma lista com alguns dos recursos que podem ser solicitados.
-N nome	O nome da tarefa. Se não estiver presente a SGE utiliza o nome do script que inicia a tarefa.
-o [[hostname]:]path,...	O caminho que será usado para a saída padrão da tarefa.
-pe ambiente n	Inicia o ambiente de processamento paralelo.
-r y[es] | n[o]	Define a capacidade da tarefa ser re-executada em caso de interrupção.
-v variável=[valor]	Define ou redefine as variáveis de ambiente que devem ser exportadas para o ambiente de execução da tarefa.

8 – Outros comando úteis

qstat	Lista o estado das tarefas submetidas
qstat –pe *nome*	Lista o estado das tarefas relacionadas ao ambiente de paralelização *nome*
qstat –u *nome*	Lista as tarefas pertencentes ao usuário *nome*
qdel *job*	Cancela a tarefa *job* esteja ela em execução ou não
quota	Mostra o espaço usado em disco e os limites da cota
quota –s	Apresenta as informações de quota em unidades mais adequadas à leitura
mpdtrace	Lista os nós do cluster onde o MPD está sendo executado

Apêndice I

Alguns Recursos Solicitáveis ao SGE

Recurso	Formato	Descrição
h_cpu	segundos, ou [[HH:]MM:]SS	(Hard) Quantidade máxima de tempo de CPU usado por todos os processos na tarefa. Se ultrapassado a tarefa é terminada.
h_vmem	tamanho*	(Hard) Quantidade máxima de memória virtual que pode ser utilizada por todos os processos na tarefa. Se ultrapassado a tarefa é abortada
hostname	string	Nome da maquina na qual a tarefa deve ser executada.
mem_free=*valor*	tamanho*	Requisita cpu com memória livre >= a *valor*
mem_total=*valor*	tamanho*	Requisita cpu com memória total maior ou igual a *valor*
s_cpu	segundos, ou [[HH:]MM:]SS	(Soft) Quantidade máxima de tempo de CPU usado por todos os processos na tarefa. Quando ultrapassado a tarefa recebe um sinal de aviso.
s_vmem	tamanho*	(Soft) Quantidade de memória virtual que pode ser utilizada por todos os processos na tarefa. Quando ultrapassado um sinal é enviado para a tarefa.
qname	string	Nome de uma fila no cluster

***tamanho**

tamanho especifica a quantidade máxima em termos de bytes ou palavras (words). Ele é expresso na forma *inteiro[sufixo]*. O sufixo é um multiplicador definido na tabela seguinte ("b" significa bytes (default) e "w" palavra)

Sufixo		Multiplicador
b	w	1
kb	kw	1024
mb	mw	1,048,576
gb	gw	1,073,741,824
tb	tw	1,099,511,627,776

Apêndice II

Glossário

core	Designa cada unidade de processamento contida em um processador multicores (core duo, core quad, etc)
cpu	"Central Processing Unit" – Aqui designará uma unidade de processamento que, conforme o caso, poderá ser um *core* ou um nó do *cluster*
nó	Computador que está conectado ao *cluster* por uma identificação de rede única.
slot	Designa a menor unidade de processamento dentro do *cluste*r. Em nosso caso cada *slot* corresponde a um *core*.