

Submetendo e Administrado Jobs.

O RHHEAD conta com o sistema de filas instalado para submeter e administrar jobs via scripts.

Para que os recursos computacionais sejam divididos de maneira razoável entre todos os usuários, é necessário uma forma de organizar e priorizar as requisições de uso, comumente chamadas de “jobs”. Para isso, o cluster conta com o sistema Torque:

<http://www.adaptivecomputing.com/products/open-source/torque/>

e o agendador de tarefas Maui:

<http://www.adaptivecomputing.com/products/open-source/maui/>

Juntos gerenciam os Jobs que serão executados, respeitando as políticas estabelecidas. O Torque trabalha com o conceito de filas, que são estruturas para classificar e agrupar os Jobs sob critérios como: número de processadores requisitados, quantidade de nós, quantidade de memória RAM, tempo de processamento e assim por diante.

Submetendo um Job nos programas Gaussian e CPMD

No cluster não é possível submeter Jobs dos programas Gaussian e CPMD das maneiras convencionais:

```
$ g09 < input.gjf > output.out &  
$ mpi -np 4 cpmd.x < input.inp > output.out &
```

***OBS.: JOBS SUBMETIDOS POR ESTE CAMINHO SERÃO AUTOMATICAMENTE KILLADOS.**

Os Jobs devem ser submetidos no sistema Torque através de scripts '.pbs' da seguinte maneira:

```
$ qsub script.pbs
```

Sendo o arquivo 'script.pbs' o script de submissão.

Este script deve ser feito pelo usuário. Para fazer o script, o usuário deve se preocupar com 3 fatores importantes:

- 1- As diretivas PBS.
- 2- Os módulos dos programas que for utilizar.
- 3- Linha de execução do programa que for utilizar.

1. Diretivas PBS

As diretivas PBS são a primeira coisa à se colocar no script. Existem quatro delas que são fundamentais e devem ser colocadas da seguinte maneira:

```
#PBS -S /bin/bash  
#PBS -l walltime=48:00:00  
#PBS -l nodes=1:ppn=4  
#PBS -N nome
```

A primeira linha é padrão.

A segunda linha indica o tempo necessário ao job. Depois desse o tempo, o job para. Observe que no exemplo está um job de 48 horas.

***OBS: DEVEM SER UTILIZADOS 3 TEMPOS ESPECÍFICOS PARA OS CÁLCULOS: 1) 48 HORAS (2 DIAS), 2) 360 HORAS (15 DIAS) E 3) 1440 HORAS (2 MESES).**

Na terceira linha o usuário coloca a quantidade de nós e processadores que serão utilizados por job. Observe que no exemplo está sendo utilizado 1 nó e 4 processadores por job.

Na quarta linha coloca-se o “nome” referente ao job. Este nome aparecerá quando o usuário lista os Jobs no torque (será mostrado mais à frente).

***OBS: PARA MANTER A FLUIDEZ DE CÁLCULOS DEVEM SER SUBMETIDOS NO MÁXIMO 15 JOBS POR USUÁRIO. FIQUE ATENTO QUE CADA USUÁRIO TERÁ A DISPOSIÇÃO PARA OS 15 JOBS: 32 PROCESSADORES E 200 GB DE MEMÓRIA. CASO O USUÁRIO ULTRAPASSAR ESTA QUANTIDADE, INVIABILIZANDO QUE OUTRO USUÁRIO SUBMETA, O EXCEDENTE SERÁ IMEDIATAMENTE KILLADO.**

2. Módulo dos programas Gaussian e CPMD.

Ao submeter Jobs desses dois programas, é necessário carregar os módulos necessários à execução desses programas.

Para utilizar o CPMD, é necessário carregar dois módulos: o módulo do próprio CPMD e o módulo MPI.

```
module add softwares/cpmd/4.1-gnu-4.8
module add mpi/openmpi-x86_64
```

Para utilizar o programa Gaussian, é necessário apenas adicionar o módulo deste programa:

```
module load softwares/gaussian-09/d.01
```

3. Linhas de execução do Job.

Por último no script PBS coloca-se a linha de execução do Job. A linha de execução do programa CPMD é a seguinte:

```
mpirun -np N cpmd.x input.inp > output.out
```

Sendo “N” o número de processadores que foi colocado na diretiva PBS, “input.inp” o nome do input de entrada e “output.out” o nome do arquivo de saída.

A linha de execução do programa Gaussian é a seguinte:

```
g09 input.gjf
```

Observe que não é necessário colocar o nome do arquivo de saída. Este arquivo será criado automaticamente utilizando o nome do input de entrada com a extensão ‘.log’.

Exemplos de scripts no PBS

Exemplo de um script PBS para execução do programa CPMD:

```
#PBS -S /bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l walltime=30:00:00
#PBS -N cpmd_wfnop

cd $PBS_O_WORKDIR

INP=$PBS_JOBNAME".inp"
OUT=$PBS_JOBNAME".out"
EXE="cpmd.x"

echo "inicio do job: "`date`
echo "Hostname: " `hostname`
echo "PWD: "$PWD
```

```
module add mpi/openmpi-x86_64
module add softwares/cpmd/4.1-gnu-4.8

mpirun -np 4 $EXE $INP > $OUT
```

Exemplo de um script PBS para execução do programa Gaussian:

```
#!/bin/bash
##
#PBS -S /bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l walltime=99:00:00
#PBS -N Agua

echo "inicio do job: "`date`
echo "Hostname: " `hostname`

echo "PWD: "$PWD

module load softwares/gaussian-09/d.01
cd $PBS_O_WORKDIR

g09 < H2O_2.gjf > H2O_2.out
```

Ao submeter um job no programa torque é possível listar as filas de jobs utilizando o comando:

```
$ qstat -a
```

Ao listar as filas de execução do programa torque são mostradas informações relevantes como: tempo total de cálculo, tempo de execução do cálculo, nome do cálculo (colocado como uma diretiva PBS) e usuário que executou o job.

Para excluir um job da fila do torque basta utilizar o comando abaixo:

```
$ qdel N
```

Sendo “N” o número do Job ID referente ao Job que se pretende excluir da fila. Este Job ID é mostrado ao listar o Job em execução no torque.

Direcionamento de Arquivos Temporários para o Storage01

Os arquivos temporários do g09 consomem muito espaço, desta forma é necessário direcioná-los para um diretório específico. Este procedimento deve ser sempre inserido no input do g09, caso contrário o espaço total do sistema é preenchido e todos os cálculos são killados.

Exemplo:

```
%rwf=/Storage01/TempG09
%int=/Storage01/TempG09
%d2e=/Storage01/TempG09
%nosave
%chk=OHH2_0H2O_TS.chk
%mem=4GB
%nproc=4
# opt freq mp2/aug-cc-pvqz test
```

Reactant

```
0 1
O      -0.27425100  -0.10953400  -0.00007100
H      0.99680100  -0.09849300   0.00101500
H     -0.52884700   0.80918900   0.00017400
```

Os arquivos rwf, int e d2e consomem muito espaço e serão direcionados para a pasta /Storage01/TempG09. O comando %nosave deve ser usado logo após o direcionamento, indicando que os arquivos acima da indicação serão deletados ao final do cálculo.