O Cluster NEWTON: Guia do Usuário

Lindolfo Meira
Centro Nacional de Supercomputação

22 de março de 2019

1 Panorama do Sistema

O cluster NEWTON opera com o sistema OpenSUSE Leap, versão 42.3, e conta, basicamente, com uma unidade de conexão e 6 unidades de processamento. Cada qual com 128 GB de RAM e 8 processadores quadcore AMD Opteron 8356, de 2.3 GHz de frequência, totalizando 192 núcleos de processamento e um desempenho teórico de, aproximadamente, 2 TFlops (precisão simples). Cada uma das 6 unidades de processamento possui 4 discos locais (num arranjo RAID-0) dedicados a *scratch*, montados no endereço `/mnt/scratch` (pastas `/mnt/scratch/$USER` são criadas automaticamente pelo sistema). O padrão de conexão da rede de produção no NEWTON é Ethernet, e opera à taxa de 1 Gbps. Apesar de as submissões ao sistema de filas precisarem ser feitas sempre a partir da pasta `$DATA` de cada usuário (ver Seção 4), dados sensíveis devem, depois de processados, ser transferidos ao `$HOME`. A partição que armazena as pastas `$DATA` tem caráter temporário e, portanto, nunca é submetida a rotinas de backup.

2 Acesso Remoto

Para acesso remoto, a partir de uma estação com sistema operacional baseado em UNIX, utiliza-se o programa `ssh`, usando-se a seguinte instrução, digitada diretamente na linha de comando:

```
$ ssh newton.cesup.ufrgs.br -l username
```

Sendo que `username` (argumento da opção `-l`) deve ser substituído pelo nome de usuário associado à conta disponibilizada pelo CESUP.

De modo similar, a transferência de arquivos desde a estação do usuário até o cluster pode ser feita pelo comando `scp`, da seguinte maneira:

```
$ scp arquivo username@newton.cesup.ufrgs.br:
```

Isto copiará o arquivo `arquivo` à pasta `$HOME` do usuário `username` no cluster. Ao copiar-se uma pasta inteira, a instrução `scp` deve ser substituída por `scp -r`.

Usuários de sistemas Windows devem instalar em suas estações o cliente SSH/SCP de sua preferência. Uma consulta a qualquer mecanismo de busca na Internet retornará inúmeras opções de clientes para este sistema.

3 Ambiente Computacional

Dadas as características do hardware, o NEWTON é preferencialmente empregado no processamento de códigos que operam em regime de memória compartilhada. Ambas as suítes de compiladores disponibilizadas, GNU e PGI, têm suporte à tecnologia (via OpenMP). A compilação de códigos OpenMP é habilitada pela inserção de um parâmetro específico à linha de comando. Nos compiladores da suíte GNU (`gcc, g++, gfortran`), o parâmetro é `-fopenmp`. No caso dos compiladores da suíte PGI (`pgcc, pgc++, pgfortran`), o parâmetro é `-mp`.

No que diz respeito ao processamento em memória distribuída, somente a implementação OpenMPI (não confundir com OpenMP) é disponibilizada, mas em duas distribuições distintas: uma compilada inteiramente com a suíte GNU (*default* do ambiente), outra com a suíte PGI. Usuários que necessitarem ou preferirem a distribuição PGI podem alterar seu respectivo ambiente através do comando `mpi-selector-menu`. A compilação de códigos MPI é feita pelos *wrappers* `mpicc/mpic++/mpif90/mpifort`, sem a necessidade de quaisquer parâmetros adicionais à linha de comando.

3.1 Softwares

Uma ampla variedade de softwares voltados ao processamento científico de alto desempenho estão disponíveis aos usuários do NEWTON. Entre eles destacam-se a suíte ANSYS (módulos CFD); o GAMESS, *General Atomic and Molecular Electronic Structure System*; o SIESTA, *Spanish Initiative for Electronic Simulations with Thousands of Atoms*, compilado com suporte a cálculos de transporte balístico eletrônico (TranSIESTA); e o QE (*Quantum Espresso*), uma suíte voltada ao cálculo de estruturas eletrônicas e modelagem de materiais em nanoescala, cujas particularidades acerca do modo de operação fazem com que seja necessário o emprego de técnicas instrumentais bastante específicas. Usuários deste software, após a leitura integral deste manual, devem contatar a equipe de suporte (`suporte@cesup.ufrgs.br`), solicitando informações acerca dos procedimentos adequados à sua utilização.

3.2 Bibliotecas

Entre as bibliotecas de otimização, destacam-se a suíte AMD Core Math Library (ACML), que implementa, entre outros, o Linear Algebra Package (LAPACK), o Basic Linear Algebra Subprograms (BLAS, níveis 1, 2 e 3) e um Random Number Generator (RNG), além de uma interface Fast Fourier Transforms (FFTs), que contém um subconjunto das funções originalmente implementadas pela FFTW (Fastest Fourier Transform in the West), a qual também é disponibilizada. A ACML é equivalente à Math Kernel Library (MKL), mas otimizada para hardware AMD. O Centro disponibiliza ainda o Automatically Tuned Linear Algebra Software (ATLAS), a GNU Scientific Library (GSL), a GNU Multiple Precision Arithmetic Library (GMP), o Basic Linear Algebra Communication Subprograms (BLACS), a Scalable LAPACK (ScaLAPACK), o Network Common Data Form (NetCDF) e o Hierarchical Data Format (HDF5), entre outros. Informações adicionais acerca do modo de utilização destas bibliotecas são fornecidas somente por meio da documentação original, específica a cada uma.

3.3 Cotas

É importante observar que as partições que abrigam as pastas `$HOME` e `$DATA` possuem cotas, e cada usuário é limitado à utilização de 512 GB em cada uma delas. O monitoramento das estatísticas de ocupação das partições deve ser feito com o comando `quota -s`, e é de inteira responsabilidade do usuário, que deve providenciar de antemão a transferência de seus arquivos a mídias externas, de modo a evitar que o volume de seus dados atinja o valor da cota.

4 Sistema de Filas

Todas as tarefas computacionais executadas no NEWTON, que opera com a versão 18.1.3 do Altair PBS-Pro, devem ser submetidas ao processamento somente via sistema de filas (com exceção de compilações e afins). Os comandos de submissão e monitoramento dos jobs são, respectivamente, o `qsub` e o `qstat` (para maiores informações, acesse os manuais online digitando, após a conexão, `man` seguido do nome do comando de interesse).

Para submeter um job, após transferir-se à sua respectiva pasta na partição de dados (`cd $DATA`), o usuário deve editar o modelo de script lá existente (`script.sh`) adaptando-o a seu respectivo caso. Feito isso, basta invocar o comando `qsub`, informando o nome do arquivo recém editado como argumento. Após a submissão, o usuário tem seu *prompt* de comando liberado, podendo desconectar-se do cluster sem comprometer seu trabalho. Abaixo uma versão comentada do arquivo `script.sh`:

```
#!/bin/sh
# A linha acima, incluindo o símbolo #, define o interpretador e,
# portanto, a sintaxe adotada no script, e deve ser sempre a primei-
# ra linha do mesmo, não podendo ser precedida nem mesmo por espa-
# ços ou linhas em branco.
#
# Linhas que começam com "#PBS" são interpretadas como comandos
# internos pelo PBS. Linhas que começam somente com # são tratadas
# como comentários (exceto no caso da primeira linha do script), e
# linhas em branco são totalmente ignoradas pelo sistema.
#
# Definição do interpretador utilizado internamente pelo PBS e
# do nome que se queira dar ao job (não deve conter espaços).
#PBS -S /bin/sh
#PBS -N job_exemplo
#
# Requisição de alocação de 8 slots para um código com suporte a
# processamento em memória compartilhada e concatenação do erro
# padrão na saída padrão (-j).
#PBS -l select=1:ncpus=8
#PBS -j oe
#
# Digamos que seu executável esteja em /dados/$USER/tmp e que
# seu script é submetido a partir desta pasta. Mesmo adotando-a
# como pasta de trabalho, o PBS não transfere a ela a execução
# do script, senão quando pela invocação ipsis litteris do co-
# mando abaixo.
cd $PBS_O_WORKDIR
#
# Linha de disparo de um programa com suporte a processamento em
# memória compartilhada (note que se o símbolo ./ não for prefi-
# xado ao nome do seu executável, é necessário então fornecer o
# caminho absoluto do arquivo).
./programa_openmp
```

Note-se que as linhas sem caracteres especiais no início são comandos normais do sistema operacional (obedecendo a sintaxe do interpretador definido na primeira linha do script) e são absolutamente transparentes ao PBS.

Aplicações que operam em regime de memória compartilhada necessitam de um esquema de alocação do tipo `FILL_UP`. Logo, o argumento do parâmetro `select`, na instrução `-l`, assume sempre o valor 1 no caso de jobs OpenMP, enquanto o argumento do parâmetro `ncpus` pode variar até o número máximo de núcleos de processamento nos nós do sistema. O PBS encarrega-se de estabelecer o valor de `OMP_NUM_THREADS` de acordo com o

valor de `ncpus`.

No caso do MPI (processamento em memória distribuída), o emprego de uma regra de alocação do tipo `ROUND_ROBIN` pode ajudar a minimizar o tempo de espera pela disponibilidade do recurso. Nestes casos, o valor atribuído à `select` deve refletir o número de núcleos de processamento desejado e o parâmetro `ncpus` pode ser omitido, pois assume automaticamente o valor 1. Convém mencionar que programas MPI precisam ser invocados pelo *wrapper* `mpiexec/mpirun`.

O monitoramento do *status* dos jobs no sistema de filas é feito pelo comando `qstat`. Quando disparado sem argumentos, o sistema retorna o status de todos os jobs em processamento no momento do disparo. Empregando-se o parâmetro `-u user`, o sistema apresenta somente o status do(s) job(s) do usuário `user`. O parâmetro `-f JOB_ID` fornece informações detalhadas acerca do job `JOB_ID`. Para cancelar um job em espera ou execução utiliza-se `qdel JOB_ID`, sendo `JOB_ID` o número de referência do job, assim como retornado pelo `qstat`. Uma série de parâmetros adicionais podem ser utilizados com ambos os comandos aqui mencionados. Para informações mais detalhadas, sugere-se a leitura dos manuais online (i.e., uma vez conectado digite, na linha de comando do terminal, `man comando_de_interesse`).