

Guia do Usuário

Titânio

titanio.ufabc.edu.br

UFABC

versatushpc.com.br

A Versatus

Com foco exclusivo em Computação de Alto Desempenho (HPC), a Versatus é uma empresa Brasileira especialista em soluções de HPC. Utilizando software e hardware de ponta combinados com uma grande gama de serviços profissionais, ajudamos nossos clientes a criar soluções de HPC de alta qualidade, eficiência e confiabilidade.

Nossa missão é permitir que pesquisadores e instituições de pesquisa possam focar exclusivamente nos seus objetivos e produção científica, fornecendo o que há de melhor em termos de equipamentos, softwares e gerenciamento completo de ambientes computacionais.

A Versatus constrói clusters de alto desempenho sob medida para seus clientes. Com uma abordagem holística que analisa desde as técnicas e algoritmos computacionais que seus clientes utilizarão até a infraestrutura elétrica e térmica do recinto em que os sistemas repousarão, a Versatus

projeta o cluster certo para resolver o seu problema computacional em pesquisa ou produção. Isto garante que você adquira um sistema tecnicamente robusto, ecologicamente responsável, com preço competitivo e um suporte de alta qualidade. Este guia tem o objetivo de oferecer um início rápido aos usuários do sistema computacional do Titânio-UFABC descrevendo brevemente o cluster e o sistema instalado, além do seu uso.

COPYRIGHT

© 2020 VersatusHPC. Todos os direitos reservados. Algumas partes deste manual podem ter direitos autorais protegidos por terceiros. O presente manual é distribuído em sua íntegra e não é permitida nenhuma alteração, cópia, distribuição ou criação de trabalhos derivados dos conteúdos deste documento, total ou parcialmente, sem a expressa autorização por escrito da VersatusHPC.

Conteúdo

Descrição do Cluster

- Hardware

 - Headnode

 - Nós Computacionais

 - Rede de Gerenciamento

- Partições

- Arquivos e Armazenamento

- Nós de Processamento

- Redes

- Formas de Acesso

Softwares Instalados

- Compiladores

 - GNU (gcc, g++, gfortran)

 - LLVM

- Bibliotecas de Transferência de Mensagem

 - OpenMPI

 - MPICH

 - MVAPICH

- Bibliotecas Matemáticas

 - FFTW

 - HDF5

 - OpenBLAS

 - PnetCDF

 - NetCDF

 - ScaLAPACK

- Softwares Científicos

 - Abinit

 - Autodock Vina

 - Clustal Ômega

 - Clustalw

 - Cp2k

 - DFTB+

 - Fenics

Gaussian'09

GROMACS

LAMMPS

MPFR

MrBayes

ORCA

OpenFOAM

Quantum ESPRESSO

ROOT

SIESTA

Wannier90

Variáveis de Ambiente

Utilizando o Cluster

 Sistema Operacional

 Alterando a Senha de Usuário

Submetendo e Gerenciando Jobs

 Diretivas e Variáveis do Job

 Filas

 Scripts de Submissão

 Tipos de Jobs

 Batch Job

 Exemplo 1

 Exemplo 2

 Job interativo

 Solicitando GPU

 Monitorando os seus Jobs

 Cancelando Jobs

 Problemas, Resoluções e Dúvidas

Obtendo Ajuda

Descrição do Cluster

Hardware

Headnode

1 (um) Nó de Gerenciamento R740xd configurado com:

- 2 (dois) processadores AMD Opteron 6344;
- 64GB de memória DDR3 ECC Reg;
- 1 (uma) Controladora RAID SAS LSI 2108;
- 4 (quatro) HDDs de 1 TB em RAID10;

Nós Computacionais

Tipo 1) 1 (um) nó computacional SGI configurado com:

- 2 (dois) processadores AMD Opteron 6344;
- 64 GB de memória DDR3 ECC Reg;

Tipo 2) 40 (quarenta) nós computacionais SGI configurados com:

- 2 (dois) processadores AMD Opteron 6376;
- 256 GB de memória DDR3 ECC Reg.

Tipo 3) 3 (três) nós computacionais SGI C2110G-RP5 configurados com:

- 2 (dois) processadores Intel Xeon E5-2660 v2;
- 64 GB de memória DDR3 ECC Reg;
- 3 (três) GPGPUs NVIDIA Tesla K20m;
- 1 (um) HDD 1 TB SATA 6 Gb/s.

Rede de Gerenciamento

1 (um) Switch Ethernet Dell Networking S4048-ON:

- 48 portas 10GbE SFP+;
- 6 portas 40GbE QSFP+;
- 18 (dezoito) cabos Dell Networking, Cable, SFP+ para SFP+ 10GbE, 1m;

Rede de Alto Desempenho

1 (um) Switch Mellanox Infiniband EDR SB7890 Gerenciável

- Não-Gerenciável;
- 36 portas QSFP28, EDR IB2;
- 1 (um) Mellanox Telescopic Rack install kit;
- 17 (dezesete) cabos Mellanox EDR VPI EDR InfiniBand QSFP passive copper, 1m;

Partições

Headnode:

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
------------	------	------	------	-------	------	------------

```

/dev/mapper/cl_headnode-root xfs          230G   53G  178G  23% /
/dev/sda1                    xfs          1014M  250M  765M  25% /boot
/dev/mapper/cl_headnode-opt xfs          1.6T   27G  1.6T   2% /opt
172.27.255.253:/scratch      nfs          512G  286G  227G  56% /scratch/global
172.27.255.253:/home        nfs          14T   6.4T  7.7T  46% /home

```

Arquivos e Armazenamento

O diretório padrão do usuário é construído da seguinte forma:

/home/<usuário>/

Ex: ***/home/vhpc/***

Esta área possui **14 TB** disponíveis que são compartilhados entre todos os usuários.

Há também outra área de trabalho destinadas ao armazenamento temporário de trabalhos, que está disponível em **'/scratch/global'**. O **'/scratch/global'** é a área de trabalho reservada no storage, com **512 GB** de espaço disponível. O scratch possui a mesma estrutura do HOME dos usuários, ou seja, **'/scratch/global/<usuário>'**.

Este diretório de scratch global é um local temporário e pode ser removido ao final do job. Arquivos importantes, devem ser copiados para o diretório HOME do usuário.

Nós de Processamento

Os nós desse cluster estão nomeados no formato "nXXX". Ex: "n003"

O cluster Titânio possui 24 nós listados abaixo (ver [Nós Computacionais](#)):

Hostname	Tipo de máquina
n[001-040]	2
gn[001-003]	3
titanio-login	1

Redes

Os nós estão conectados a 3 redes distintas que são relacionadas apenas para completude das informações:

1. 10 Gigabit Ethernet;
2. IPMI;
3. Infiniband.

A rede IPMI é utilizada pelos procedimentos administrativos do cluster via protocolo IPMI, enquanto as redes 10GbE e Infiniband são utilizadas para trocas de dados entre os nós e também mensagens MPI.

Formas de Acesso

O acesso remoto diretamente ao “titanio-login” é feito através de SSH (Secure SHell):

```
$ ssh <login>@titanio.ufabc.int.br
```

Softwares Instalados

O sistema tem um grande conjunto de softwares instalados de acordo com as necessidades dos usuários. Os softwares disponíveis em produção deste guia estão listados abaixo.

Todos os softwares, compiladores e bibliotecas pré-instalados no cluster estão disponíveis em `'/opt/ohpc/pub/'` e `'/opt/spack/opt/spack/linux-centos7-x86_64/'`.

Compiladores

A família de compiladores GNU e LLVM e PGI e suas ferramentas, estão disponíveis nas seguintes versões:

GNU (gcc, g++, gfortran)

Versão 8.3.0

Executáveis: `'/opt/ohpc/pub/compiler/gcc/8.3.0/bin/'`

Bibliotecas: `'/opt/ohpc/pub/compiler/gcc/8.3.0/lib/'`

Module: `'gnu8/8.3.0'`

LLVM

Versão: 5.0.1

Executáveis: `'/opt/ohpc/pub/compiler/llvm/5.0.1/bin/'`

Bibliotecas: `'/opt/ohpc/pub/compiler/llvm/5.0.1/lib64/'`

Module: `'llvm5/5.0.1'`

Bibliotecas de Transferência de Mensagem

As bibliotecas MPI pré-instalados na workstation são as seguintes:

OpenMPI

OpenMPI é uma biblioteca de troca de mensagens de código aberto implementada

Versão 3.1.4 compilada com GNU 8.3:

Module: `'openmpi3/3.1.4'`

Versão 3.1.5 compilada com GNU 8.3:

Module: `'openmpi-3.1.5-gcc-8.3.0-z3nft5q'`

MPICH

MPICH é outra biblioteca de troca de mensagens aberta existente

Versão 3.3.1 compilada com GNU 8.3.0

Executáveis: '/opt/ohpc/pub/mpi/mpich-gnu8-ohpc/3.3.1/bin'
Bibliotecas: '/opt/ohpc/pub/mpi/mpich-gnu8-ohpc/3.3.1/lib'
Module: '**mpich/3.3.1**'

MVAPICH

MVAPICH é uma biblioteca de troca de mensagens MPI sobre InfiniBand.

Versão 2.3.2 compilada com GNU 8.3
Executáveis: '/opt/ohpc/pub/mpi/mvapich2-gnu8/2.3.2/bin'
Bibliotecas: '/opt/ohpc/pub/mpi/mvapich2-gnu8/2.3.2/lib'
Module: '**mvapich2/2.3.2**'

Bibliotecas Matemáticas

Estão pré-instalados no cluster as seguintes bibliotecas:

FFTW

Essa biblioteca são sub-rotinas para cálculo das transformadas rápida e discreta de Fourier, além de transformadas discretas do seno e cosseno e transformadas de Hartley.

Versão 3.3.8 compilada com GNU 8.3:
Module: '**fftw/3.3.8**'

Versão 3.3.8 compilada com GNU 8.3:
Module: '**fftw-3.3.8-gcc-8.3.0-kcvqays**'

HDF5

Hierarchical Data Format (HDF, HDF4, ou HDF5) nome para um conjunto de formatos de arquivos e bibliotecas criadas para organização e armazenamento de grandes quantidades de dados numéricos.

Versão 1.10.5 compilada com GNU 8.3:
Module: '**hdf5/1.10.5**'

OpenBLAS

O BLAS (Básicos Álgebra Linear Subprograms) são rotinas que fornecem blocos de construção padrão para a realização de operações básicas vetoriais e matriciais.

Versão 0.3.7 compilada com GNU 8.3:
Module: '**openblas/0.3.7**'

Versão 0.3.7 compilada com GNU 8.3:

Module: **'openblas-0.3.7-gcc-8.3.0-ex5qeai'**

PnetCDF

PnetCDF é uma biblioteca de E/S (entrada/saída) paralelo de alta desempenho para acessar arquivos compatíveis com NetCDF.

Versão 1.12 compilada com GNU 8.3:

Module: **'pnetcdf/1.12.0'**

NetCDF

Network Common Data Form (netCDF) é um conjunto de bibliotecas de softwares criadas para melhor organização de dados científicos.

Versão 4.7.1 compilada com GNU 8.3:

Module: **'netcdf/4.7.1'**

ScaLAPACK

O Scalable LAPACK é uma biblioteca de rotinas de alto desempenho para álgebra linear baseada no LAPACK.

Versão 2.0.2 compilada com GNU 8.3:

Module: **'scalapack/2.0.2'**

Softwares Científicos

Os softwares científicos instalados são mostrados em cada tópico abaixo com seus respectivos módulos de ambiente para carregamento via comando **“module load”**:

Abinit

Versão 8.10 compilada com GNU 8.3.0:

Module: **'abinit-8.10.3-gcc-8.3.0-kj7txeh'**

Autodock Vina

Versão 1.1 compilada com GNU 8.3.0:

Module: **'autodock-vina-1_1_2-gcc-8.3.0-ji2etz2'**

Clustal Ômega

Versão 1.2 compilada com GNU 8.3.0:

Module: **'clustal-omega-1.2.4-gcc-8.3.0-ukqcjnn'**

Clustalw

Versão 2.1 compilada com GNU 8.3.0:

Module: **'clustalw-2.1-gcc-8.3.0-kxiy4wl'**

Cp2k

Versão 6.1 compilada com GNU 8.3.0:

Module: **'cp2k-6.1-gcc-8.3.0-igmikij'**

DFTB+

Versão 19.1 compilada com GNU 8.3.0 e OpenMPI 3.1:

Module: **'dftbplus-19.1-gcc-8.3.0-bt3ynap'**

Versão 19.1 compilada com GNU 8.3.0 com suporte à GPU (MAGMA):

Module: **'dftbplus-19.1-gcc-8.3.0-q6al2bt'**

Fenics

Versão 2019.1 compilada com GNU 8.3.0:

Module: **'fenics-dolfin-2019.1.0.post0-gcc-8.3.0-ephhzi2'**

Gaussian'09

Versão D.01 compilada com GNU 8.3.0:

Module: **'gaussian/09d01'**

GROMACS

Versão 2019 compilada com GNU 8.3.0:

Module: **'gromacs-2019-gcc-8.3.0-s3e6bvc'**

Versão 2019 compilada com GNU 8.3.0 com suporte à GPU (CUDA):

Module: **'gromacs-2019-gcc-8.3.0-7rtgtbe'**

LAMMPS

Versão 20190919 compilada com GNU 8.3.0:

Module: **'lammeps-20190919-gcc-8.3.0-nnlb54h'**

MPFR

Versão 4.0 compilada com GNU 8.3.0:

Module: **'mpfr-4.0.2-gcc-8.3.0-qh3hdx2'**

MrBayes

Versão 2019.1 compilada com GNU 8.3.0:

Module: **'mrbayes-3.2.7a-gcc-8.3.0-wc6cjon'**

ORCA

Versão 4.2 compilada com GNU 8.3.0:

Module: **'orca/4.2.1'**

OpenFOAM

Versão 7.0 compilada com GNU 8.3.0:

Module: **'openfoam-org-7.0-gcc-8.3.0-vn77g6b'**

Quantum ESPRESSO

Versão 6.4 compilada com GNU 8.3.0:

Module: **'quantum-esspresso-6.4.1-gcc-8.3.0-bvbbaex'**

ROOT

Versão 6.16 compilada com GNU 8.3.0:

Module: **'root-6.16.00-gcc-8.3.0-ih7ao2c'**

SIESTA

Versão 4.0 compilada com GNU 8.3.0:

Module: **'siesta-4.0.2-gcc-8.3.0-53ka4ig'**

Wannier90

Versão 2.1 compilada com GNU 8.3.0:

Module: **'wannier90-2.1.0-gcc-8.3.0-c77aqsk'**

Variáveis de Ambiente

Em ambientes de computação de alto desempenho é comum que haja diversos compiladores e bibliotecas diferentes para a mesma rotina. Por exemplo, para compilação de códigos Fortran e MPI, temos disponíveis “gfortran” e “mpifort”. Além dessa variação de softwares que desempenham a mesma função, também é comum “coleccionar” diferentes versões de compiladores, bibliotecas e softwares. Com isso, surge a necessidade de selecionar quais arquivos queremos que o sistema operacional disponibilize para o uso, de acordo com a atividade que iremos desempenhar. No cluster Titânio está disponível uma lista com todas as bibliotecas e compiladores presentes na máquina, basta o usuário selecionar quais ferramentas ele irá utilizar. Essa seleção é feita por sessão, ou seja, a cada job, as ferramentas devem ser selecionadas, e ao final do job, elas voltam ao valor padrão de “nenhuma seleção”. O comando utilizado para este gerenciamento é o “module” [<http://lmod.readthedocs.io/en/latest/>], ele pode listar, carregar, descarregar ou substituir uma ferramenta dentre as disponíveis.

Os módulos de variáveis de ambiente que você necessita podem ser carregados manualmente na linha de comando. Ainda assim, é mais conveniente incluir os módulos para serem carregados nos scripts de submissão que você utilizar (veja [Submetendo e Administrando Jobs](#) e alguns exemplos que seguem). Outra possibilidade é modificar o seu “.bash_profile” para carregar os módulos automaticamente no login (não esqueça de descarregar os módulos se há algum conflito com algum outro módulo que você precisa para um software específico).

Os módulos de variáveis atualmente disponíveis são:

```
[vhpc@titanio-login ~]$ module avail
...
```

A sintaxe do module é:

- **module av, avail** - lista todos os módulos disponíveis;
- **module list** - lista todos os módulos atualmente em uso;
- **module show <caminho_do_módulo>** - mostra o conteúdo do arquivo do módulo;
- **module add, load <caminho_do_módulo>** - carrega as configurações e prepara o ambiente de acordo com o módulo selecionado;
- **module rm, unload <caminho_do_módulo>** - remove as configurações específicas deste módulo;
- **module purge** - remove todos os módulos carregados ;
- **module help** - exibe todos os subcomandos do module.

Referência completa (em inglês): [http://lmod.readthedocs.io/en/latest/010_user.html]

Ex:

Cenário 1: Compilar um programa utilizando o GCC 8.3, para esta compilação o usuário deve carregar o módulo do compilador com o comando:

```
[vhpc@titanio-login ~]$ module add gnu8/8.3.0
```

Cenário 2: O usuário deseja remover todos os módulos já carregados:

```
[vhpc@titanio-login ~]$ module purge
```


Utilizando o Cluster

Sistema Operacional

A versão do seu sistema operacional é o CentOS Linux 7.7 x86_64, distribuição baseada no Red Hat Enterprise Linux.

Alterando a Senha de Usuário

Para alterar a senha de seu usuário no sistema, use os comandos:

```
$ kinit <login>
$ ipa user-mod --password <login>
```

Assim que você digitar este comando, o sistema irá iniciar automaticamente o processo onde você deverá:

- Digitar a nova senha do usuário;
- Digitar novamente a nova senha do usuário.

Submetendo e Gerenciando Jobs

O Titânio 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 de filas SLURM [<https://slurm.schedmd.com/documentation.html>] para gerenciar os jobs que serão executados, respeitando as políticas estabelecidas. O SLURM trabalha com o conceito de filas, que são estruturas para classificar e agrupar os jobs e nós computacionais sob critérios como: número de processadores requisitados, quantidade de nós, quantidade de memória RAM, tempo de processamento e assim por diante. Na configuração atual, estão configuradas partições com ordem de prioridade e em que apenas alguns grupos de usuários têm acesso a algumas delas (na seção [Filas](#) temos mais detalhes).

É possível exibir as filas do sistema e os seus status com o comando:

```
[vhpc@titanio-login ~]$ sinfo -a
```

O acesso via SSH aos nós é controlado pelo sistema de filas, ou seja, usuários comuns só serão permitidos acessar os nós se houver(em) job(s) em execução nestes nós.

Diretivas e Variáveis do Job

Dentro dos jobs existem algumas instruções que são dedicadas ao SLURM, entre outras coisas para instruí-lo onde, como e por quanto tempo rodar. Essas instruções são chamadas diretivas, e estão nas linhas que se iniciam com: “**#SBATCH**”. O SLURM irá considerar todas as diretivas até que encontre uma linha executável, depois deste ponto, será ignorada qualquer outra diretiva.

Diretivas comuns são:

#SBATCH --job-name=<nome> → Nome do job

#SBATCH --ntasks=<número> → Número total de núcleos

#SBATCH --time=<dias-hh:mm:ss> → Tempo de execução (dias-hh:mm:ss)

#SBATCH --partition=<fila> → Fila onde o job será alocado

Além disso, quando o job é submetido, o SLURM coloca a disposição do job algumas variáveis de ambiente que tem o intuito de facilitar algumas operações. Algumas delas, por exemplo são:

\$SLURM_JOB_NODELIST → esta variável contém a lista de nós e núcleos que foram alocadas ao job, é útil, por exemplo para ser usada como parâmetro para o “mpirun”:

```
mpirun -machinefile $SLURM_JOB_NODELIST ./meuprograma
```

\$SLURM_SUBMIT_DIR → esta variável aponta para o diretório de onde o script foi submetido.

\$SLURM_JOB_NAME → contém o nome especificado na diretiva “**#SBATCH --job-name=**”

A lista completa de diretivas, opções e variáveis pode ser encontrada em (em Inglês): [\[https://slurm.schedmd.com/sbatch.html#lbAH\]](https://slurm.schedmd.com/sbatch.html#lbAH).

Filas

As filas são o que permitem um bom desempenho, distribuição e agendamento de uso de recursos. No Titânio temos configuradas as filas abaixo:

Fila	Tempo limite	Máximo de Nós Alocados por Job	Nó(s) associado(s)	Observação
medium	4 dias	4	n[013-040]	Fila padrão
long	14 dias	2	n[001-012]	
gpu	14 dias	--	gn[001-003]	

A política das filas está configurada como FIFO (First In First Out), ou seja, os jobs em espera serão executados por ordem de chegada nas respectivas filas.

Scripts de Submissão

Exemplos de scripts de submissão de jobs estão localizados em `'/opt/versatushpc/data/jobscripts/'`. Os scripts `"template.slurm"` e `"template-gpu.slurm"` podem ser utilizados como modelos para criação de scripts personalizadas para os softwares.

Tipos de Jobs

Quanto aos jobs, existem dois tipos mais comuns:

Batch Job

Um arquivo de script que contém comandos para executar aplicações específicas, usado pelo programa `"sbatch"` para informar ao SLURM as características do seu job. Exemplo de script:

```
#!/bin/bash
#SBATCH --time=24:00:00
#SBATCH --ntasks=8
#SBATCH --job-name=teste

cd $SLURM_SUBMIT_DIR

#### Prepare o ambiente com os devidos módulos #####
module add openmpi3/3.1.4

#### Exporte as variáveis relevantes para o job #####3
export OMP_NUM_THREADS=1

#### Execução do job ####
mpirun -np $SLURM_NTASKS prime 200000 > teste.out
```


- A primeira linha informa ao SLURM que o Shell escolhido pelo usuário para rodar o job é o Bash (“/bin/bash”);
- A segunda linha indica, em horas reais, qual a duração do job;
- A terceira linha indica o número de núcleos para este job;
- A variável **\$SLURM_SUBMIT_DIR** tem por valor padrão o diretório onde o job foi submetido. Logo após seguem os comandos para a execução do job propriamente dito;
- Variáveis de ambiente necessárias à execução do job devem ser declaradas/exportadas antes do job.

Exemplo da submissão de um batch job:

```
[vhpc@headnode ~]$ sbatch script.slurm
```

Abaixo são apresentados scripts de submissão com ideias que podem ser mescladas para se construir um script ideal para cada tipo de trabalho, além dos já presentes no diretório **'/opt/versatushpc/data/jobscripts'**.

Exemplo 1

O exemplo abaixo mostra um job simples de execução de um software genérico em 20 núcleos e walltime de 2 horas:

```
#!/bin/bash
#SBATCH --job-name=TESTE
#SBATCH --ntasks=20
#SBATCH --time=2:00:00

module load openmpi3/3.1.4

cd $SLURM_SUBMIT_DIR

mpirun -np $SLURM_NTASKS ./meuprograma
```

Exemplo 2

O job abaixo é um exemplo de execução do software em 16 núcleos de 1 nó e walltime de 30 horas:

```
#!/bin/bash
##
## Copyright (C) 2009-2017 VersatusHPC, Inc.
##
## nodes = quantidade de nodes
#SBATCH --nodes=1
##
## ntasks-per-node = quantidade de processos por node
#SBATCH --ntasks-per-node=16
##
## time = quantidade de tempo
#SBATCH --time=30:00:00
```

```

##
## Nome do job . Aparece na saida do comando 'qstat' .
## E recomendado, mas nao necessario, que o nome do job
## seja o mesmo que o nome do arquivo de input
#SBATCH --job-name=<Job_Name>
##
## Consulte <https://slurm.schedmd.com/sbatch.html> para mais informacoes sobre
## as diretivas acima.

echo -e "\n## Job iniciado em $(date +%d-%m-%Y as %T') #####\n"

## Variavel com o diretorio de scratch do job
## Consulte <https://slurm.schedmd.com/sbatch.html#lbAH> para mais informacoes
## sobre as variaveis de ambiente do SLURM.
WRKDIR=$SCRATCH/$SLURM_JOB_ID

## O nome dos arquivos de input e output sao baseados no
## nome do job (linha "#SBATCH --jobname=xxx" acima).
## Observe que nao e obrigatorio esta forma de nomear os arquivos.
INP=$SLURM_JOB_NAME".inp"
OUT=$SLURM_JOB_NAME".out"

## O diretorio onde o job sera executado sera apagado, por padrao, ao
## final do mesmo.
## Se desejar que nao seja apagado, substitua Y por N.
APAGA_SCRATCH=Y

## Informacoes do job impressos no arquivo de saida.
echo -e "\n## Jobs ativos de $USER: \n"
squeue -a --user=$USER
echo -e "\n## Node de execucao do job:          $(hostname -s) \n"
echo -e "\n## Numero de tarefas para este job: $SLURM_NTASKS \n"

#####
##----- Inicio do trabalho ----- #
#####

## Configura o ambiente de execucao do software.
module load @SOFTWARE_MODULE@

## Informacoes sobre o ambiente de execucao impressos no arquivo de saida.
echo -e "\n## Diretorio de submissao do job:   $SLURM_SUBMIT_DIR \n"
echo -e "\n## Diretorio de scratch do job:       $WRKDIR \n"
echo -e "\n## Arquivo de input:                     $INP \n"

## Cria o diretorio de scratch para o job.
mkdir -p $WRKDIR

## Transfere os inputs e arquivos necessarios para o diretorio de scratch
cd $SLURM_SUBMIT_DIR

```

```

cp $INP $WRKDIR/
cd $WRKDIR

## Execucao do software

mpirun ./mysoftware < $INP > $OUT

## Copia o diretorio de scratch para o diretorio original do job.
cp -r $WRKDIR/ $SLURM_SUBMIT_DIR/

## Apaga o diretorio de scratch do job.
if [ x"$APAGA_SCRATCH" = x"Y" ]; then
    rm -rf $WRKDIR
else
    echo -e "\n0 diretorio \e[00;31m$WRKDIR\e[00m deve ser removido manualmente"
    echo -e "para evitar problemas para outros jobs e/ou usuarios. \n"
fi

echo -e "\n## Job finalizado em $(date +%d-%m-%Y as %T') #####"

```

Job interativo

É executado como um “batch job”, porém o terminal do usuário é conectado ao host de execução, similar a uma sessão de login. A partir daí o usuário envia as opções do script de job como comandos individuais, o que facilita ao usuário depurar um script com problemas.

Para submeter um job interativo basta executar o comando abaixo:

```
[vhpc@titania-login ~]$ srun --pty bash -i
```

Para encerrar o job interativo, basta sair do terminal com o comando “exit”.

Outras opções podem ser utilizadas no comando “srun”, referência completa (em inglês):

[\[https://slurm.schedmd.com/srun.html\]](https://slurm.schedmd.com/srun.html).

Solicitando GPU

A workstation está equipada com três nós com três GPGPU Tesla K20m (nove no total). Caso o software esteja habilitado para rodar em GPU e você deseja que seu cálculo utilize a GPU, você deve solicitá-las no job com a opção:

```
--gres=gpu:1
```

O exemplo abaixo mostra como solicitar uma GPU e 20 núcleos em um job:

```

#!/bin/bash
#SBATCH --partition=gpu
#SBATCH --job-name=GPU-TESTE
#SBATCH --ntasks=20
#SBATCH --gres=gpu:1

# Aqui o comando de execução de seu programa.
srun ./meuprograma

```

Monitorando os seus Jobs

Para verificar o estado do(s) seu(s) job(s), usa-se o comando “**squeue**”. Algumas das opções mais usadas são:

- **--user=<usuário>** - mostra somente as informações referentes ao usuário “usuário”
- **--jobs=<jobid>** - mostra informações completas referentes ao job com a id “jobid”
- **--name=<jobname>** - mostra informações completas referentes ao job com o nome “jobname”

Referência completa (em inglês): [<https://slurm.schedmd.com/squeue.html>]

Cancelando Jobs

O comando utilizado para apagar os jobs da fila é o “**scancel**”, a sintaxe é:

- **scancel <jobid>** - jobid é o número que o SLURM retorna quando o job é aceito e enfileirado.
- **scancel --jobname=<jobname>** - cancela o job com nome “jobname”.
- **scancel --usage** - mostra brevemente as opções do comando.

Referência completa (em inglês) [<https://slurm.schedmd.com/scancel.html>]

Problemas, Resoluções e Dúvidas

A documentação oficial do SLURM mantém uma compilação de problemas comuns e suas resoluções como também dúvidas mais frequentes sobre o sistema de filas.

Para problemas e resoluções (em inglês): [<https://slurm.schedmd.com/troubleshoot.html>].

Para dúvidas e questões frequentes (em inglês): [<https://slurm.schedmd.com/faq.html>].

Obtendo Ajuda

Em caso de contrato ativo de manutenção continuada os chamados podem ser abertos por telefone (**+55 11 3436-0664**) ou através de e-mail [**suporte@versatushpc.com.br**], fornecendo o ID do cliente e as informações abaixo:

1. O que você está tentando fazer (executar um programa, copiar um arquivo, acessar um diretório);
2. Passos que você está executando para reproduzirmos o problema;
3. Qual/quais as mensagens de erro estão sendo exibidas;
4. Quais os passos alternativos você já tentou e quais os resultados.



HAPPY
CLUSTERING!

contato@versatushpc.com.br



Rua Estela, 515, 21B

Paraíso - São Paulo - SP

CEP 04011-002

Fone: +55 11 3436-0664 | 11 2275-9733