# Manual de utilização do Cluster HPC Departamento de Computação - Cefet-MG - BH.

# Versão atualizada em: 15/10/2014

Antes de utilizar o cluster pela primeira vez, entre em contato com a equipe de Apoio Técnico do Decom para configuração de sua conta de usuário.

O cluster pode ser utilizado a partir de qualquer computador, dentro ou fora do Cefet. Após iniciar a execução de um programa, você pode sair do cluster e desligar seu computador. O cluster lhe enviará um email quando a execução terminar.

Para rodar programas no cluster, é necessário:

- 1. Realizar a transferência dos arquivos entre seu computador e o cluster. Assim que sua conta de usuário for configurada, você terá um espaço em disco de até 5GB onde poderá armazenar seus arquivos (ver detalhes na *página 2*).
- 2. Conectar ao cluster, via *ssh*, e executar os programas desejados (ver detalhes na *página* 3).

Observação:

• Sempre mantenha cópia dos dados e programas em seu computador pois não fazemos *backup*.

# 1. Transferindo arquivos entre seu computador e o cluster

A transferência de arquivos pode ser feita utilizando o programa *Filezilla* (ou qualquer outro programa que suporta *sftp*).

1.1. Instale e execute o *Filezilla* (disponível para download em https://filezilla-project.org/download.php?show all=1).

#### **1.2.** Crie uma conexão com o servidor.

- 1.2.1. Abra o Filezilla e clique em "Arquivo" -> "Gerenciador de sites".
- 1.2.2. Clique em "*Novo Site*".
- 1.2.3. Preencha os campos:
  - 1.2.3.1. Host: com cluster.decom.cefetmg.br
  - 1.2.3.2. **Porta:** com **2200**
  - 1.2.3.3. **Protocolo:** selecione **SFTP SSH File Transfer Protocol**
  - 1.2.3.4. **Tipo de logon:** selecione a opção **Interativo**
  - 1.2.3.5. Preencha o campo **Usuário** com o nome do usuário que foi criado pelo administrador do cluster.
  - 1.2.3.6. Clique em **Conectar.**
- 1.2.4. Será exibida uma janela onde você deverá digitar sua senha. Deixe marcada a opção: *"Lembrar senha para esta seção"*.

#### Observações:

- A próxima vez que utilizar o *Filezilla*, os dados da conexão já estarão salvos. Para acessá-los, basta clicar em: "Arquivos" -> "Gerenciador de sites" e escolher o site criado anteriormente.
- A janela do *Filezilla* é dividida em 2 partes. Do lado esquerdo, aparecem os arquivos que estão em seu computador e, do lado direito, os arquivos que estão no cluster. Para realizar a transferência, selecione os arquivos em um dos lados e arraste-os para o outro.
- Para facilitar a utilização do cluster, evite usar nome de arquivos com espaços em branco.

O Filezilla pode ser usado para transferir e gerenciar seus arquivos (apagar, renomear, criar pastas, remover, copiar, etc).

A execução dos programas no cluster será feita via *ssh*, conforme descrito na seguinte seção.

#### 2. Executando programas no cluster.

Passos:

- 1. conectar ao servidor via ssh (ver seção 2.1).
- 2. rodar o programa desejado (ver seção 2.2).

# 2.1. A conexão com o servidor via ssh

- 2.1.1. Faça o download do PuTTY:
  - Entre no site e faça o download

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

• ou faça o download direto clicando no link:

http://tartarus.org/~simon/putty-snapshots/x86/putty-installer.exe

- 2.1.2. Execute o PuTTY.
- 2.1.3. Preencha os campos:
  - 2.1.3.1. Host Name (or IP address): cluster.decom.cefetmg.br Alternativamente, pode-se utilizar o IP: 200.131.37.158
  - 2.1.3.2. Clique em **Open**

Reputty Configuration	-	×
Category:		
E. Session	Basic options for your PuTTY session	
	Specify the destination you want to connect to	
- Keyboard	Host Name (or IP address) 200 131 37 158	2200
Bell     Features     Window     Appearance     Behaviour     Translation     Colours     Ornection     Proxy     Telnet     Rlogin     SSH     SSH     Serial	Connection type: ◯ Raw ◯ Telnet ◯ Rlogin ⊚ SSI	H 🔘 Serial
	Load, save or delete a stored session Sav <u>e</u> d Sessions	
	Default Settings	Load Sa <u>v</u> e Delete
	Close window on exit: Always Never Only on clean exit	
About	<u>O</u> pen	<u>C</u> ancel

- 2.1.3.3. Um alerta será exibido, para prosseguir clique em **Sim.**
- 2.1.3.4. Um terminal será aberto. Entre com seu usuário e senha. Assim que sua senha for validada, você estará conectado ao servidor e poderá executar seus programas (veja a próxima seção).

O servidor possui o sistema operacional Linux instalado. Alguns comandos básicos de utilização do Linux são apresentados no Apêndice.

# 2.2. Rodando programas no cluster

- 1. Dentro do terminal aberto pelo *Putty* (descrito na seção anterior), entre na pasta onde está localizado o programa que deseja executar.
  - a. Para verificar em qual pasta você se encontra, digite: **pwd**. A saída será algo do tipo: **\home\USUARIO**, onde "**USUARIO**" é o nome de seu usuário no cluster.
  - b. Para entrar em uma pasta, digite: cd nome\_da\_pasta
  - c. Para sair da pasta, digite: cd ..
  - d. Para listar o conteúdo de uma pasta, digite: Is
  - e. Outros comandos são apresentados no Apêndice.
- 2. Com relação ao programa executável que irá rodar no cluster, na maioria dos casos você pode compilá-lo em seu próprio computador ou no cluster.
  - a. Se preferir compilá-lo em seu computador, basta transferir o executável para o cluster utilizando o *Filezilla*.
  - b. Se preferir compilá-lo no cluster, execute um dos comando abaixo, de acordo com a linguagem de programação utilizada.

Programa SEQUENCIAL em C/C++	Programa PARALELO em C/C++	
\$ g++ programa.c -o nome_do_executavel	<pre>\$ mpicc programa.c -o nome_do_executavel</pre>	
Programa SEQUENCIAL Java	Programa PARALELO em Java com threads	
\$ javac programa.java	\$ javac programa.java	

3. Dentro da pasta onde se encontra o programa executável, crie um arquivo texto (com o nome "*script.txt*") com o conteúdo aprensentado abaixo, de acordo com a linguagem de programação utilizada.

**Se o arquivo "***script.txt*" for criado no Windows, salve-o utilizando o formato UTF-8. Na sua pasta no cluster, já existe um exemplo do arquivo "script.txt". Você pode copiá-lo para seu computador (utilizando o *Filezilla*), fazer as alterações desejadas (utilizando o Notepad, por exemplo), e depois copiá-lo de volta para o cluster. Como o arquivo foi criado originalmente no padrão UFT-8, você não precisará se preocupar com o formato do arquivo.

Programa SEQUENCIAL em C/C++	Programa PARALELO em C/C++
#!/bin/bash #SBATCHntasks=1 #SBATCHmail-user= <u>seuemail@cefetmg.br</u> #SBATCHmail-type=END #SBATCHpartition=mes srun Nome_do_Programa	#!/bin/bash #SBATCHntasks=10 #SBATCHmail-user=seuemail@cefetmg.br #SBATCHmail-type=END #SBATCHpartition=mes mpiexec Nome_do_Programa
Programa SEQUENCIAL em Java	Programa "PARALELO" em Java (threads)
#!/bin/bash #SBATCHntasks=1 #SBATCHmail-user=seuemail@cefetmg.br #SBATCHmail-type=END #SBATCHpartition=mes srun java Nome_do_Programa	<pre>#!/bin/bash #SBATCHntasks=1 #SBATCHmail-user=seuemail@cefetmg.br #SBATCHmail-type=END #SBATCHcpus-per-task=16 #SBATCHpartition=mes srun java Nome_do_Programa</pre>
Programa SEQUENCIAL em Octave	
#!/bin/bash #SBATCHntasks=1 #SBATCHmail-user=seuemail@cefetmg.br #SBATCHmail-type=END #SBATCHpartition=mes srun octave Nome_do_Programa.m	

# Onde:

 --ntasks=n: representa a quantidade "n" de programas que irão rodar. Valores de <u>n>1</u> são utilizados, principalmente, em programas que rodam em <u>paralelo</u> (usando mpi, por exemplo). <u>Se seu programa for seguencial, basta manter "ntask=1".</u>

Para programas implementados em **Java com threads**, mantenha "n=1" pois as *threads* irão rodar nos núcleos alocados pela variável "--cpus-per-task" (o conteúdo desta variável será explicado abaixo).

- --mail-user: seu email. O cluster poderá lhe enviar emails no início, no fim e na ocorrência de erros durante a execução do programa. Se não quiser receber e-mails, retire esta linha do arquivo.
- --mail-type=ALL: indica que o cluster lhe enviará um e-mail nas seguintes situações: início, fim e erro do programa. Para enviar emails apenas no final, substitua o ALL por END. Se não quiser receber e-mails, retire esta linha do arquivo.
- --partition: indica a partição onde o programa irá executar. A principal

característica de cada partição é a restrição de tempo. Ao utilizar a partição *"semlimite"* o programa poderá executar sem restrição de tempo. Ao utilizar as partições *"mes"* ou *"semana"* o programa poderá executar até 30 ou 7 dias, respectivamente. Ao final deste tempo, o programa é encerrado automaticamente.

Para conhecer as partições disponíveis digite o comando \$ sinfo2

Exemplo de saída:

PARTITION	CPUS(A/I/O/T)	GROUPS	TIMELIMIT
semlimite	56/16/0/72	all	infinite
mes	121/87/0/208	all	30-00:00:0

A coluna **PARTITION** exibe o nome das partições existentes no cluster. Este nome será utilizado no momento em que o programa for submetido para execução no cluster.

A coluna **CPUS** exibe a quantidade de CPUs que estão sendo utilizadas (Allocated=56), disponíveis (Idle=16), outros (Other=0) e total (Total=72). Ao submeter um programa para executar no cluster, é importante verificar a quantidade de CPUs disponíveis. Se não houver CPUs disponíveis, o programa entrará em uma fila. A política de alocação de recursos é a FIFO (First In First Out).

A coluna **GROUPS** exibe os grupos que podem utilizar uma determinada partição.

A coluna **TIMELIMIT** exibe o tempo máximo que um programa pode rodar em uma determinada partição. Formato: dd-hh:mm:s

- --cpus-per-task: utilizado apenas para programas em Java com threads. Como as threads devem ser executadas em um único computador, este parâmetro possibilita alocar mais núcleos para uma mesma instância do programa. Como não existem computadores com mais de 24 núcleos, o valor de "n" deve ser menor ou igual a 24.
- Nome\_do\_Programa: é o nome do programa que será executado (exemplo: "a.out").
- 4. Para executar o programa digite:\$ sbatch script.txt

O "**sbatch**" irá ler os comandos e parâmetros contidos no arquivo "**script.txt**". A última linha deste arquivo contém o comando que executa seu programa (exemplo: **srun Nome\_do\_Programa** ou **mpiexec Nome\_do\_Programa**).

#### Observações:

- Imediatamente após o início da execução, a seguinte mensagem será exibida: "Submitted batch job id", onde id representa o identificador do programa no cluster. Ao término da execução, você receberá um e-mail com a seguinte mensagem "Job\_id 134 Ended" (o número 134 é um exemplo de id). Também será informado o tempo de execução do programa.
- A saída do programa (gerada pelos comandos "*printf*", por exemplo) pode ser vista no arquivo "**slurm-id.out**". Onde **id** é o identificador do programa.
- Para listar seus programas que estão rodando no cluster, digite:
   \$ squeue2
- Para listar todos os programas que estão rodando no cluster, digite:
   \$ squeue3
- Para cancelar a execução de um programa que foi submetido ao cluster, digite:
   \$ scancel id, onde id é o identificador do programa (134, por exemplo).
- Para sair do cluster digite *exit* ou feche a janela do *Putty*.

# Apêndice

Comansdo báicos para utilização do Linux (fonte: http://wiki.ubuntu-br.org/ComandosBasicos)

- **pwd** O comando **pwd** lhe permite saber em qual diretório você está no momento, onde **pwd** significa "print working directory".
  - Executando "pwd" no diretório Desktop mostrará "~/Desktop". Observe que o Terminal do Gnome também mostra esta informação na barra de títulos da janela. Veja a imagem de exemplo no topo desta página.
- cd Este comando nos permite se deslocar entre a árvore de diretórios do sistema. Quando abrimos um terminal ou seção shell, você entra direto no seu diretório pessoal. Para mover-se pelo sistema de arquivos você deve usar o cd.
  - "cd /" para ir ao diretório raiz.
  - "cd" para ir ao seu diretório pessoal.
  - **"cd .."** para acessar um diretório de nível acima do atual.
  - "cd -" para voltar ao diretório que se encontrava antes de mudar.
  - Para navegar através múltiplos níveis de diretórios em só comando, use por exemplo, "cd

/var/www", que o levará diretamente ao sub-diretório /www do diretório /var.

- **cp** Copia arquivos e diretórios.
  - **"cp file foo"** para fazer uma cópia exata do arquivo "file" dando-lhe o nome de "foo".
  - "sudo cp /etc/X11/xorg.conf /etc/X11/xorg.conf-bkp" para gerar uma cópia de segurança exata do arquivo "/etc/X11/xorg.conf" dando-lhe o nome de "/etc/X11/xorg.conf-bkp".
- **mv** Este comando move arquivos e diretórios, sendo muito usado também para renomear um determinado arquivo.
  - **"mv arquivo1 arquivo2"** para renomear o arquivo "arquivo1" localizado no diretório pessoal do usuário para "arquivo2" no mesmo local.
  - **"mv foo ~/Desktop"** moverá o arquivo "foo" para seu diretório Desktop sem alterar seu nome. Você deve especificar um novo nome se quiser renomear um arquivo.
- **ls** Comando utilizado para listar o conteúdo de um diretório. Usado com certas opções, é possível ver o tamanho dos arquivos, quando foram criados, e as permissões de cada um.
  - "ls ~" para mostrar os arquivos que estão em seu diretório pessoal.
  - "ls -hal ~" para mostrar os arquivos que estão em seu diretório pessoal, inclusive os ocultos (-a) em forma de uma listagem (-l) e com as informações de tamanho mais amigável a nós seres humanos (-h).
- rm Utilize este comando para remover (deletar) arquivos e opcionalmente diretórios. Por padrão o comando rm exibe um promptonde o usuário deve confirmar a exclusão de cada arquivo, digitando a letra "y" seguido de "Enter".
  - "rm arquivo1" para remover o arquivo chamado "arquivo1" do diretório corrente após confirmação no prompt.
  - **"rm -f arquivo1"** para remover o arquivo chamado "arquivo1" do diretório corrente sem que lhe seja exibido o prompt de confirmação.
  - "rm -R ~/temp/" para remover de forma recursiva o diretório /temp localizado em sua pasta pessoal e todo seu conteúdo, seja ele arquivos e outras arvores de sub-diretórios.
  - mkdir Comando cuja finalidade é permitir a criação de um ou mais diretórios.
    - **"mkdir musicas"** para criar um diretório chamado "musicas" dentro do diretório corrente.