Guia do usuário

Particularidades do HPCC-Crab

O diferencial de um cluster HPC é a possibilidade de executar tarefas de computação distribuída (ou paralela). Há implementações específicas de programas de bioinformática que fazem uso desta tecnologia e têm o melhor aproveitamento do equipamento disponível.

Acesso remoto

O acesso ao HPCC-Crab é realizado obrigatoriamente via protocolo **SSH**. O endereço do HPCC-Crab na rede do INCA é 10.250.0.166. O acesso externo direto não está habilitado por questão de segurança.

A partir de sistemas GNU/Linux e MacOS

Para acesso remoto, a partir de uma estação com sistema operacional *UNIX-Like*, utiliza-se o programa *SSH* diretamente no interpretador de comandos (terminal, console, tela preta) conectado à rede do INCA, digitando a seguinte instrução no *prompt* (*$*):

```
$ ssh username@crab
```

Sendo *username* o seu nome de usuário associado à conta do HPCC-Crab. Use a opção -Y para ter suporte a gráficos (X11).

A maioria dos sistemas *UNIX-Like* já oferece um cliente *SSH* (openssh) na instalação padrão. Caso use uma distribuição GNU/Linux que não tenha o *SSH* instalado, ele poderá ser instalado com um dos seguintes comandos:

```
$ apt-get install openssh-client   ### (Debian/Ubuntu)

$ yum install openssh-clients      ### (Fedora/RHEL)

$ slapt-get install openssh        ### (Slackware)
```

Pode-se ainda baixar o código fonte (http://www.openssh.com) e compilá-lo.

A partir de sistemas Windows

O Windows não dispõe de um cliente SSH por padrão. Recomenda-se o uso do programa **putty** para acessar o *cluster*. O **putty** pode ser encontrado no site oficial:

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

Ao abrir o **putty** basta inserir o IP do *cluster* (10.250.0.165) no campo *Host Name* e iniciar a sessão. Na tela de login insira seu nome de usuário e senha quando requisitado.

Primeiro acesso

O usuário receberá seu nome de acesso (*username*) e uma senha provisória por e-mail. No primeiro acesso ao HPCC-Crab, altere a senha utilizando o comando *passwd*. Senhas fracas serão rejeitadas automaticamente. Evite usar palavras presentes em dicionários (em qualquer idioma), nomes de

pessoas, sequências numéricas, números de telefone, data de aniversário, número de matrícula, ou qualquer informação pessoal. Inclua na senha letras, números e caracteres especiais.

Diretórios de usuários

Cada usuário terá uma pasta exclusiva no diretório **/home**, com o endereço **/home/usuario**. Esta pasta é destinada às configurações pessoais do usuário. A pasta do usuário tem quota de 500MB e não pode ser utilizada para armazenamento de dados de projetos, que têm pasta dedicada com alocação adequada de espaço.

Diretórios de projetos

Cada projeto cadastrado no HPCC-Crab tem um diretório exclusivo, localizado em uma das unidades de armazenamento dedicadas a dados. Cada diretório recebe automaticamente sete subpastas: **bin**, **data**, **doc**, **lib**, **share, results** e **Backup**. Estas pastas não podem ser excluídas, porém membros do projeto tem permissão para escrever, ler e executar arquivos, bem como criar pastas, dentro delas. Os membros do projeto podem, também, criar novas pastas de acordo com sua necessidade. O uso destas pastas é sugerido e não obrigatório.

A pasta **bin** é destinada a receber os executáveis e scripts utilizados no projeto, a pasta **lib** armazena bibliotecas específicas, **doc** é reservado para a documentação do projeto, **data** recebe as pastas com dados e análises, e **share** é configurada para permitir o compartilhamento de documentos com não-membros do grupo. A pasta **Backup** é destinada ao armazenamento de dados (máximo total de 20GB) que serão incluídos na rotina de *backup*. Todas as pastas, com exceção de **share**, só podem ser acessadas pelos membros do projeto.

Especialmente a pasta <project>/lib/Rpackages é destinada para servir de path local de instalação de pacotes R e Bioconductor utilizados no projeto. Por favor, inclua o comando 'export R_LIBS_USER=~/<project>/lib/Rpackages' no seu script ou shell antes de rodar o R para utilizar/instalar pacotes R locais. Outra alternativa é incluir '.libPaths("~/<project>/lib/Rpackages")' ao início do seu R script.

Diretório de ferramentas

Um diretório compartilhado nomeado **/tools** foi criado para a instalação de programas para uso comum de todos os projetos. Para cada novo programa ou versão alternativa de programas já instalados, será criada uma pasta dentro de **/tools**. Estas ferramentas podem também ser acessadas via sistema de módulos descrito logo abaixo.

Módulos

Para facilitar a utilização de programas de uso comum no *cluster* estão disponíveis **módulos** que, quando carregados, preparam o ambiente com as dependências necessárias para o funcionamento correto do programa. Para saber quais módulos estão disponíveis, utilize o comando

```
$ module avail
```

e para carregar o módulo de interesse, utilize o comando

```
$ module load <nome do módulo>
```

Para mais informações, acesse este link.

Instalação de programas pelos usuários

Na instalação de programa a partir do código fonte, pode ser utilizada a opção **--prefix=~/DESTINO** no comando de configuração para compilação. Por exemplo:

```
$ ./configure  --prefix=~/SeuProjeto/bin/PROGRAMA-VERSAO

$ make

$ make install
```

Arquivos executáveis podem ser alocados em ~/*SeuProjeto*/bin/ (insira este endereço no seu $PATH, dentro do seu script).

```
export PATH=~/SeuProjeto/bin/:$PATH
```

R

O projeto R é um sistema para computação estatística e gráfica, que compreende uma linguagem de programação e um ambiente de desenvolvimento integrado. No Crab, o pacote R (versão R-3.4.4) é carregado no ambiente utilizando

```
$ module load R
```

CPAN

Configure o CPAN para instalar bibliotecas localmente (sugerimos que use o diretório *lib* do seu projeto) indicando o endereço nas variáveis de ambiente do Perl.

Exemplo:

criar diretorio para instalar pacotes

```
$ PALL=~/<SeuProjeto>/lib/perl5all ;

$ mkdir -p $PALL

$ export PERL_MB_OPT="--install_base ${PALL}"

$ export PERL_MM_OPT="INSTALL_BASE=${PALL}"

$ export PERL5LIB=${PALL}/lib/perl5

$ export PATH=${PALL}/bin:${PATH}

$ export PERL_LOCAL_LIB_ROOT="${PALL}"
```

Para instalar bibliotecas adicionais localmente execute:

```
$ cpan -i Foo::Bar
```

onde *Foo::Bar* deve ser substituído pelo módulo a ser instalado.

Quanto precisar utilizar estas bibliotecas, inclua em seu script de submissão:

```
PALL=~/<SeuProjeto>/lib/perl5all ;

export PERL5LIB=${PALL}/lib/perl5

export PATH=${PALL}/bin:${PATH}
```

Utilização do cluster

Todas as tarefas computacionais devem ser invocadas pelo sistema de gerenciamento de filas. No HPCC-Crab o sistema instalado é o **SLURM Workload Manager**. Segue um **FAQ** (em inglês) que sana várias dúvidas.

Sistema de gerenciamento de filas

Submissões ao sistema de filas devem ser feitas sempre a partir das pastas de projeto. Crie um arquivo novo ou copie o arquivo de exemplo *sleep.slurm* armazenado na pasta bin do seu projeto e altere as informações necessárias.

sleep.slurm - Exemplo de script para ser executado com o slurm do Crab.

```
#!/usr/bin/env bash

#SBATCH --job-name=sleeper

#SBATCH --ntasks=1

#SBATCH --cpus-per-task=1

#SBATCH --time=00:01:00

### comentario: print date and hostname, do nothing for 30 seconds and print date again

date; hostname;

sleep 30

date; echo;

exit;
```

A primeira linha (#!/usr/bin/env bash), chamada de *shebang*, define o interpretador que será usado para a execução deste *script*.

As linhas que começam com **#SBATCH** são interpretadas como comandos internos do slurm:

--job-name Indica o nome do job submetido. Este será o nome visível na fila.

--ntasks Especifica o número de taks por node que o job requer.

--cpus-per-task Número de processadores (ou *threads*) usados por task.

--time Define o tempo máximo que o job levará para ser completado.

Outras opções podem ser utilizadas. Um resumo das opções disponíveis pode ser encontrado aqui e uma versão mais explicativa aqui.

Linhas que começam apenas com cerquilha espaço (#) são tratadas como comentários. **Atenção** pois, se não houver um espaço após a cerquilha, a linha será lida como um comando. As demais linhas contém os comandos a serem executados pelo sistema operacional. Para executar o exemplo acima, digite na linha de comando:

```
$ sbatch ./sleep.slurm
```

Para monitorar sua execução na fila:

```
$ squeue
```

Lista todos os jobs em execução

```
$ scontrol show job <job ID>
```

Mostra detalhes do job em questão

Para melhor entendimento sobre a submissão de tarefas para o Slurm recomenda-se a leitura deste link.

Scratch

Todos os nós de computação contém um disco dedicado para gravação de dados em processamento. O uso deste disco é recomendado e proporciona aumento substancial na performance. Este disco é montado no diretório /scr de cada nó. Há um exemplo de uso deste diretório no script **usescratch.slurm** no diretório *bin* do seu projeto.

Informações Adicionais

Manuais Externos

- Comandos Linux:

http://linuxcommand.org/learning_the_shell.php

http://faculty.ucr.edu/~tgirke/Documents/UNIX/linux_manual.html

OpenMPI: http://www.open-mpi.org

Slurm Workload Manager: https://slurm.schedmd.com/documentation.html

- Guias de usuários de outros clusters (não garantimos a equivalência do sistema)

An Introduction to the HPC Computing Facility (ou Guia mal-humorado do administrador do sistema):

http://hpc.oit.uci.edu/HPC_USER_HOWTO.html

Running Jobs on The HPC Cluster: http://hpc.oit.uci.edu/running-jobs

Job Scheduling by SLURM (High Performance Computing at iCER)

Job Management by SLURM (High Performance Computing at iCER)

Apoio ao Usuário

Em caso de dúvidas ou solicitações, envie um e-mail para *contatohpc@inca.gov.br* ou ligue para o ramal 1329 (*help-desk* do INCA).