Configuração de Cluster com Slurm no Ubuntu LTS

Guia sobre como configurar o cluster gpu no Ubuntu 22.04 usando slurm (com *cgroups*).

0. Suposições

- masternode 100.xx.100.xx
- workernode 200.xx.200.xx
- masternode FQDN = masternode.master.local
- workernode FQDN = workernode.worker.local

Obs. 1 : Pode ser interessante instalar o pacote `iptutils-ping` para ter ferramentas de gestão de rede através do comando `sudo apt install iputils-ping`

Obs. 2: Pode ser interessante instalar o pacote `vim-nox` para ter uma opção de editor de texto através do comando `sudo apt install vim-nox`

0.1 No masternode

1. Para definir o nome do host temporariamente (até a próxima reinicialização), utilize o comando `hostname` seguido do novo nome desejado:

```
sudo hostnamectl set-hostname masternode
```

2. Para definir o nome do host permanentemente, você precisa editar dois arquivos:

```
sudo nano /etc/hostname
```

Edite o conteúdo do arquivo para o novo nome do host desejado. Por exemplo, para definir como "masternode" [o valor deve está correto pelo passo anterior, senão ajuste]:

```
masternode
```

Salve o arquivo e feche o editor.

3. /etc/hosts`: Este arquivo armazena o mapeamento entre nomes de host e endereços IP. Abra-o em um editor de texto:

```
sudo nano /etc/hosts
```

Edite o conteúdo do arquivo para o novo nome do host desejado. Por exemplo, para definir como "masternode":

```
127.0.0.1 localhost
127.0.1.1 masternode
100.xx.100.xx masternode masternode.master.local  # Substituir xx por valores reais.
200.xx.200.xx workernode workernode.worker.local  # Substituir xx por valores reais.
```

Salve o arquivo e feche o editor.

4. **Atualize o nome do host:**

Para que as alterações permanentes entrem em vigor, você precisa atualizar o nome do host na memória:

```
sudo reboot
```

0.2 No workernode

1. Para definir o nome do host temporariamente (até a próxima reinicialização), utilize o comando `hostname` seguido do novo nome desejado:

```
sudo hostnamectl set-hostname  workernode
```

2. Para definir o nome do host permanentemente, você precisa editar dois arquivos:

```
sudo nano /etc/hostname
```

Edite o conteúdo do arquivo para o novo nome do host desejado. Por exemplo, para definir como "workernode" [o valor deve está correto pelo passo anterior, senão ajuste]:

```
workernode
```

Salve o arquivo e feche o editor.

3. /etc/hosts`: Este arquivo armazena o mapeamento entre nomes de host e endereços IP. Abra-o em um editor de texto:

```
sudo nano /etc/hosts
```

Edite o conteúdo do arquivo para o novo nome do host desejado. Por exemplo, para definir como "masternode":

```
127.0.0.1 localhost
127.0.1.1 workernode
100.xx.100.xx masternode masternode.master.local  # Substituir xx por valores reais.
200.xx.200.xx workernode workernode.worker.local  # Substituir xx por valores reais.
```

Salve o arquivo e feche o editor.

4. **Atualize o nome do host:**

Para que as alterações permanentes entrem em vigor, você precisa atualizar o nome do host na memória:

```
sudo reboot
```

1. Instale drivers nvidia em todos os nodes

Atualize o sistema operacional

```
sudo apt update && apt upgrade -y
```

Remova a instalação anterior da NVIDIA

```
sudo apt autoremove nvidia* --purge
```

Execute o seguinte comando:

```
ubuntu-drivers devices
```

- Este comando listará os drivers NVIDIA disponíveis para sua placa de vídeo.
- Identifique o driver mais recente na lista.
- Execute o seguinte comando, substituindo `X` pelo número do driver (por exemplo, `nvidia-driver-550`):

```
sudo apt install nvidia-driver-X
```

- Siga as instruções na tela para concluir a instalação.

Reinicie o SO e verifique a instalação:

```
nvidia-smi
```

Instale o CUDA toolkit

```
sudo apt install nvidia-cuda-toolkit
```

Verifique a instalação do CUDA

```
nvcc --version
```

1.1 Alternativamente para instalar o Cuda 12.5 com cuda drivers

Atualize o sistema operacional

```
sudo apt update && apt upgrade -y
```

Remova a instalação anterior da NVIDIA

```
sudo apt autoremove nvidia* --purge
```

Monte o seguinte comando:

```
wget
https://developer.download.nvidia.com/compute/cuda/<release>/local_installers/cuda_<release>_<version>_linux.run
sh cuda_<release>_<version>_linux.run -m=kernel-open
```

> **Note**
>
> Verifique a versão do Cuda mais recente em https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu

Por exemplo:

```
wget
https://developer.download.nvidia.com/compute/cuda/12.5.0/local_installers/cuda_12.5.0_555.42.02_linux.run
sudo sh cuda_12.5.0_555.42.02_linux.run -m=kernel-open
```

Reinicie o SO e verifique a instalação:

```
nvidia-smi
nvcc --version
```

2. Configure o ssh sem senha

1. **Instalação do servidor SSH e Firewall**

   No `masternode` e no `workernode`:

```
sudo apt install openssh-server ufw
sudo ufw enable
sudo ufw allow ssh
```

2. **Gerar as chaves SSH no `masternode`**

```
ssh-keygen -t rsa -b 4096
```

Quando solicitado a fornecer um caminho para salvar a chave, pressione **Enter** para aceitar o local padrão (`~/.ssh/id_rsa`).

Quando solicitado a fornecer uma senha (passphrase), simplesmente pressione **Enter** para deixá-la vazia (sem senha).

3. **Copiar a chave pública para a(s) `workernode`(s)**

```
ssh-copy-id usuario@200.xx.200.xx
```

- **`usuario`**: Substitua pelo nome de usuário na máquina slave.
- **`200.xx.200.xx`**: Substitua pelo endereço IP ou hostname do `workernode`.

Se solicitado, insira a senha do usuário na máquina slave para completar a cópia da chave.

4. **Testar a configuração SSH sem senha**

   Ainda na máquina master, tente conectar-se à máquina slave sem fornecer uma senha:

   ```
   ssh usuario@200.xx.200.xx
   ```

   Você deve ser capaz de acessar a máquina slave sem ser solicitado a fornecer uma senha.

5. **Configuração adicional (opcional)**

   Para garantir que a conexão SSH funcione corretamente e para facilitar o uso futuro, você pode adicionar um bloco de configuração ao arquivo `~/.ssh/config` na máquina master. Edite o arquivo ou crie-o se não existir:

   ```
   nano ~/.ssh/config
   ```

   Adicione a seguinte configuração, substituindo os valores apropriados:

   ```
   Host workernode
       HostName 200.xx.200.xx
       User usuario
       IdentityFile ~/.ssh/id_rsa
   ```

   Isso permite que você se conecte à `workernode` simplesmente usando o comando:

   ```
   ssh workernode
   ```

6. **Verificação de permissões**

   Certifique-se de que as permissões dos arquivos de chave SSH e diretórios são corretas:

   - Assegure-se de que o diretório `~/.ssh` tenha permissão 700 na `masternode` e `workernode`:

     ```
     chmod 700 ~/.ssh
     ```

   - Assegure-se de que os arquivos de chave privada `~/.ssh/id_rsa` tenham permissão 600 no `masternode`:

     ```
     chmod 600 ~/.ssh/id_rsa
     ```

   - Assegure-se de que os arquivos de chave pública `~/.ssh/id_rsa.pub` e `~/.ssh/authorized_keys` (na `workernode`) tenham permissão 644:

     ```
     chmod 644 ~/.ssh/id_rsa.pub
     chmod 644 ~/.ssh/authorized_keys
     ```

3. Sincronize GID/UIDs

3.1 Crie usuários munge e slurm:

Nos nós `masternode` e `workernode`:

```
sudo adduser -u 1111 munge --disabled-password --gecos ""
sudo adduser -u 1121 slurm --disabled-password --gecos ""
```

E atualize as permissões de acordo com suas preferências.

4. Sincronize a hora

Em **todos** os nós (`masternode` e `workernode`), instale o NTP e defina o fuso horário para UTC.

```
sudo apt-get install -y ntp
sudo dpkg-reconfigure tzdata
Selecione America/Sao_Paulo
```

4.1 NTP Server

Escolha um nó como servidor de hora principal (**sugestão**: `masternode`) e configure-o para funcionar corretamente mesmo que não esteja ligado à Internet.

```
sudo nano /etc/ntp.conf
```

Adicione o seguinte para fornecer a sua hora local atual como predefinição, caso perca temporariamente (ou permanentemente) a ligação à Internet:

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

Reinicie o NTP

```
sudo /etc/init.d/ntp restart
```

4.2 NTP Client

Em todos os restantes nós do seu cluster, configure-os para sincronizarem os relógios com o nó que foi designado como o servidor de hora principal do cluster.

```
sudo nano /etc/ntp.conf
```

Adicione o seguinte :

```
 server <main time server> iburst
```

em que `<main time server>` é o endereço IP do nó designado como o servidor de hora principal.

Remova as seguintes linhas do arquivo (as que existirem) comentando-as com um carácter `#` no início da linha:

```
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
#server 0.ubuntu.pool.ntp.org
#server 1.ubuntu.pool.ntp.org
#server 2.ubuntu.pool.ntp.org
#server 3.ubuntu.pool.ntp.org
#pool 0.ubuntu.pool.ntp.org iburst
#pool 1.ubuntu.pool.ntp.org iburst
#pool 2.ubuntu.pool.ntp.org iburst
#pool 3.ubuntu.pool.ntp.org iburst

# Use Ubuntu's ntp server as a fallback.
#server ntp.ubuntu.com
#pool ntp.ubuntu.com
```

Salve e feche o arquivo `/etc/ntp.conf` e reinicie o NTP

```
sudo /etc/init.d/ntp restart
```

Verificar a conetividade com o servidor de hora principal:

```
ntpq -c lpeer
```

5. Configure o NFS

Para que o SLURM funcione corretamente, deve haver um local de armazenamento presente em todos os computadores do cluster com os mesmos arquivos usados para os trabalhos. Todos os computadores do cluster devem ser capazes de ler e gravar nesse diretório. Uma maneira de fazer isso é com o NFS.

NFS, ou Network File System, é um protocolo de sistema de arquivos distribuído que permite montar diretórios remotos no seu servidor. Isso permite gerenciar o espaço de armazenamento em um local diferente e gravar nesse espaço a partir de vários clientes. O NFS fornece uma forma relativamente rápida e fácil de aceder a sistemas remotos através de uma rede e funciona bem em situações em que os recursos partilhados serão acedidos regularmente.

5.1 `masternode`:

```
1  sudo apt update
2  sudo apt install -y nfs-kernel-server quota
3  sudo mkdir /storage -p
4  sudo chown usuario:grupo /storage/
5  echo "/storage 200.xx.200.xx(rw,sync,no_root_squash,no_subtree_check)" | sudo tee -a
   /etc/exports
6  sudo systemctl restart nfs-kernel-server
7  sudo ufw allow from 200.xx.200.xx to any port nfs
```

Substitua `200.xx.200.xx` pelo IP do workenode.

Substitua `usuario:grupo` por um usuário e grupo existente no `masternode` (**sugestão**: `$USER:$USER$`) .

> **Important**
>
> Note que a `linha 5` exemplifica apenas um **único nó**. Devem existir uma linha para cada nó que acessará o volume, exceto para o nó hospedeiro do volume.

5.2 `workernode`:

```
1  sudo apt update
2  sudo apt install -y nfs-common
3  sudo mkdir -p /storage
4  sudo mount 100.xx.100.xx:/storage /storage
5  echo 100.xx.100.xx:/storage /storage nfs auto,timeo=14,intr 0 0 | sudo tee -a
   /etc/fstab
6  sudo chown usuario:grupo /storage/
```

Substitua `100.xx.100.xx` pelo IP do masternode.

Substitua `usuario:grupo` por um usuário e grupo existente no workenode.

6. Configure o MUNGE

MUNGE (MUNGE Uid 'N' Gid Emporium) é uma ferramenta de autenticação que cria e valida credenciais de segurança para processos distribuídos.

MUNGE é utilizado para autenticar mensagens entre processos em clusters de computadores, garantindo que a comunicação entre diferentes nós do cluster seja segura e confiável. Ele é especialmente útil em ambientes de computação de alto desempenho (HPC) onde a segurança e a integridade da comunicação entre nós são críticas.

6.1 `masternode`:

```
sudo apt install -y libmunge-dev libmunge2 munge
sudo systemctl enable munge
sudo systemctl start munge
munge -n | unmunge | grep STATUS
sudo cp /etc/munge/munge.key /storage/
sudo chown munge /storage/munge.key
sudo chmod 400 /storage/munge.key
```

> **Tip**
>
> Você deverá obter `STATUS:  Success (0)` ao executar a `instrução 4`.

6.2 `workernode`:

```
sudo apt install -y libmunge-dev libmunge2 munge
sudo cp /storage/munge.key /etc/munge/munge.key
sudo systemctl enable munge
sudo systemctl start munge
munge -n | unmunge | grep STATUS
```

> **Tip**
>
> Você deverá obter `STATUS:  Success (0)` ao executar a `instrução 5`.

7. Configure o Slurm

SLURM (Simple Linux Utility for Resource Management) é um sistema de gerenciamento de recursos e fila de jobs (workload manager) utilizado em ambientes de computação de alto desempenho (HPC).

SLURM gerencia e aloca recursos de computação em clusters, permitindo que os usuários enviem, agendem e monitorarem jobs (tarefas de computação) de forma eficiente. Ele cuida do balanceamento de carga, agendamento de jobs, controle de acesso, e monitoramento de recursos, facilitando o uso otimizado do cluster e garantindo que os recursos sejam utilizados de forma justa e eficiente.

7.1 Configure o DB para o Slurm

Os procedimentos deverão ser realizados no `masternode`.

1. Instale o cliente para o `git`:

```
sudo apt install -y git
```

2. Clone este repositório com ficheiros de configuração e de serviço:

```
cd /storage
git clone https://github.com/marluciobarbosa/slurm.git
```

3. Instalar pré-requisitos para BD:

```
sudo apt install -y python3 gcc make openssl ruby ruby-dev libpam0g-dev libmariadb-dev
mariadb-server build-essential libssl-dev numactl hwloc libmunge-dev man2html lua5.4
sudo gem install fpm
sudo systemctl enable mysql
sudo systemctl start mysql
sudo systemctl status mysql  | grep Status:
```

Você deverá ver `Status: "Taking your SQL requests now..."`.

4. Criando o usuário e o BD para o Slurm:

```
sudo mysql -u root
```

```
create database slurm_acct_db;
create user 'slurm'@'localhost';
set password for 'slurm'@'localhost' = password('senha_para_usuario_slurm');
grant usage on *.* to 'slurm'@'localhost';
grant all privileges on slurm_acct_db.* to 'slurm'@'localhost';
flush privileges;
exit
```

> **Caution**
>
> Modifique a `senha_para_usuario_slurm` para a senha desejada para o usuário `slurm`.

7.2 Configure o Slurm

7.2.1 Configuração do Slurm no `masternode`

7.2.1.1 Criar arquivo de instalação

Você deve verificar a página de download do slurm e instalar a versão mais recente.

```
cd /storage
mkdir -p /storage/config
wget https://download.schedmd.com/slurm/slurm-24.05.0.tar.bz2
tar xvjf slurm-24.05.0.tar.bz2
cd slurm-24.05.0/
./configure --prefix=/tmp/slurm-build --sysconfdir=/etc/slurm --enable-pam --with-pam_dir=/lib/x86_64-linux-gnu/security/ --without-shared-libslurm
make
make contrib
make install
cd ..
```

> **Tip**
>
> Em algumas distribuições, o `pam_dir` pode ser `/lib/security/` ou `/usr/lib/security/`. Para localizar o diretório correto, execute `find /lib -name '*pam*.so'` ou `find /usr/lib -name '*pam*.so'`. O diretório desejado conterá as bibliotecas `pam_debug.so`, `pam_selinux.so` e `pam_group.so`. O diretório pode ter uma estrutura dada por `/usr/lib/` **ARQUITETURA**-linux-gnu `/security/`.`

7.2.1.2 Instalar o Slurm

1. **Criando e instalando o pacote do Slurm**

```
sudo mkdir -p /storage/bin
sudo fpm -s dir -t deb -v 1.0 -n slurm-24.05.0 --prefix=/usr -C /tmp/slurm-build/ -p /storage/bin/slurm.deb
sudo dpkg -i /storage/bin/slurm.deb
```

> **Note**
>
> Observe a saída do comando da linha `2`. Uma mensagem como `Created package {:path=>"slurm.deb"}` será apresentada. O `path` apresentado na mensagem que deve ser utilizado na linha `3` e na instalação no `workernode`.

2. **Criando os diretórios:**

```
sudo mkdir -p /etc/slurm /etc/slurm/prolog.d /etc/slurm/epilog.d /var/spool/slurm/ctld /var/spool/slurm/d /var/log/slurm
sudo chown slurm /var/spool/slurm/ctld /var/spool/slurm/d /var/log/slurm
sudo chmod 755 /var/log/slurm
```

3. **Copiar serviços slurm:**

```
sudo cp /storage/slurm/configs_services/slurmdbd.service /etc/systemd/system/
sudo cp /storage/slurm/configs_services/slurmctld.service /etc/systemd/system/
```

4. **Copiar a configuração do slurm DB:**

```
sudo cp /storage/slurm/configs_services/slurmdbd.conf /etc/slurm/
sudo chmod 600 /etc/slurm/slurmdbd.conf
sudo chown slurm /etc/slurm/slurmdbd.conf
```

Edite o arquivo `/etc/slurm/slurmdbd.conf` para definir a senha do BD do usuário `slurm`:

```
StoragePass=senha_para_usuario_slurm
```

5. **Portas abertas para comunicação com o slurm:**

```
sudo ufw allow from any to any port 6817
sudo ufw allow from any to any port 6818
```

6. **Inicie os serviços slurm:**

```
sudo systemctl daemon-reload
sudo systemctl enable slurmdbd
sudo systemctl start slurmdbd
sudo systemctl enable slurmctld
sudo systemctl start slurmctld
```

> ⚠ **Warning**
>
> O serviço ainda não está configurado, então `sudo systemctl status slurmctld` retornará **falha**.
> Caso deseje testar no `masternode`, execute o passo seguinte ou termine de configurar o `workernode`.

7. **Se o `masternode` for um nó de computação (`worker`) [aconselho a incluí-lo, por facilidade, com recursos limitados]:**

```
sudo cp /storage/slurm/configs_services/slurmd.service /etc/systemd/system/
sudo systemctl enable slurmd
sudo systemctl start slurmd
```

7.2.1.3 Solução de contorno

> **Important**
>
> Não resolva problema inexistente. Utilize essa solução apenas na existência do problema.

O serviço `slurmctld` pode não subir corretamente durante o boot, devido a dependências de outros serviços. Uma solução de contorno é garantir que o serviço reinicie automaticamente após a inicialização de todos os serviços da máquina.

Para isso, execute os seguintes passos:

1. **Copia o arquivo de serviço personalizado:**

```
sudo cp /storage/slurm/configs_services/restart-slurmctld.service
/etc/systemd/system/
```

2. **Recarregar as configurações do `systemd`:**

```
sudo systemctl daemon-reload
```

3. **Habilitar o novo serviço:**

```
sudo systemctl enable restart-slurmctld.service
```

4. **Reiniciar o sistema:**

   Reinicie o sistema para verificar se o serviço `slurmctld` é reiniciado automaticamente após a inicialização do sistema.

```
sudo reboot
```

5. **Verificar o status do serviço:**

Após a reinicialização, verifique o status do serviço `slurmctld` para garantir que ele foi reiniciado corretamente [isso é válido após o término da configuração].

```
sudo systemctl status slurmctld
```

7.**Verifique também o status do serviço `restart-slurmctld`:**

```
sudo systemctl status restart-slurmctld
```

7.2.2 Configuração do Slurm no `workernode`

7.2.2.1 Instalação do Slurm

Você deve verificar a página de download do slurm e instalar a versão mais recente.

```
cd /storage
sudo dpkg -i /storage/bin/slurm.deb
sudo cp /storage/slurm/configs_services/slurmd.service /etc/systemd/system
```

7.2.2.2 Portas abertas para comunicação com o slurm:

```
sudo ufw allow from any to any port 6817
sudo ufw allow from any to any port 6818
```

```
sudo systemctl enable slurmd
sudo systemctl start slurmd
```

7.2.2.3 Configurar o Slurm:

Copia o arquivo `slurm.conf` para personalizá-lo:

```
sudo cp /storage/slurm/configs_services/slurm.conf /storage/config/
```

Se tiver GPU(s), copie também o arquivo `gres.conf` para personalizá-lo:

```
sudo cp /storage/slurm/configs_services/gres.conf /storage/config/
```

Em `/storage/config/slurm.conf`, altere:

```
ControlMachine=masternode.master.local # use seu FQDN
ControlAddr=100.xx.100.xx # use o IP do masternode
```

Use `sudo slurmd -C` para imprimir as especificações da máquina. Você deve copiar as especificações de todas as máquinas no arquivo slurm.conf e modificá-lo.

Exemplo de como deve ficar em seu arquivo de configuração:

```
NodeName=workernode NodeAddr=100.xx.100.xx CPUs=2 Boards=1 SocketsPerBoard=2
CoresPerSocket=1 ThreadsPerCore=1 RealMemory=1967
```

> **Explicação dos Parâmetros:**
>
> 1. **NodeName=masternode**:
>   - Define o nome lógico do nó no cluster SLURM. Neste caso, o nome do nó é "masternode".
> 2. **NodeAddr=100.xx.100.xx**:
>   - Define o endereço IP do nó. Este é o endereço pelo qual o nó é acessado na rede. O formato é típico de um endereço IPv4.
> 3. **Gres=gpu:1**:
>   - Especifica os recursos genéricos (Generic Resources, GRES) disponíveis no nó. No caso, o nó possui 1 GPU (Graphics Processing Unit) disponível para tarefas que requerem aceleração gráfica.
> 4. **CPUs=16**:
>   - Indica o número total de CPUs (ou unidades de processamento lógico) disponíveis no nó. Este valor é calculado como:
> $$\text{CPUs} = \text{Boards} \times \text{SocketsPerBoard} \times \text{CoresPerSocket} \times \text{ThreadsPerCore}$$
>   No caso específico: $1 \times 1 \times 8 \times 2 = 16$.
> 5. **Boards=1**:
>   - Define o número de placas-mãe (motherboards) no nó. Neste caso, o nó tem uma única placa-mãe.
> 6. **SocketsPerBoard=1**:
>   - Define o número de soquetes de CPU por placa-mãe. Aqui, há 1 soquete de CPU na placa-mãe.
> 7. **CoresPerSocket=8**:

    - Define o número de núcleos de CPU por soquete. Neste nó, cada soquete de CPU tem 8 núcleos.
8. **ThreadsPerCore=2**:
    - Define o número de threads (ou hiper-threads) por núcleo de CPU. Aqui, cada núcleo suporta 2 threads, que é típico de CPUs com hyper-threading habilitado.
9. **RealMemory=63502**:
    - Define a quantidade de memória RAM disponível no nó, medida em megabytes (MB). Neste caso, o nó possui 63,502 MB (aproximadamente 62 GB) de memória RAM.

Interpretação Completa:

A linha de configuração especifica que há um nó no cluster chamado "masternode" com o endereço IP **100.xx.100.xx**. Este nó possui **1** GPU, **16** CPUs que são organizados em **1** placa-mãe, com **1** soquete de CPU por placa-mãe, **8** núcleos por soquete, e **2** threads por núcleo. Além disso, o nó tem **63,502 MB** de memória RAM disponível.

Exemplo Visual:

Para visualizar a organização interna do nó:

- 1 placa-mãe (Board)
    - 1 soquete de CPU (Socket)
        - 8 núcleos (Cores)
            - 2 threads por núcleo (Threads).

> **Caution**
>
> O comando `sudo slurmd -C` pode não apresentar o `NodeAddr`. Nesse caso, o `NodeAddr` deve ser acrescentado para cada `workernode` com o `IP` correspodente.

> **Note**
>
> O comando `sudo slurmd -C` retorna o `UpTime`. Esse parâmetro é útil para monitoramento e gestão do cluster, indicando quanto tempo o nó está disponível para o agendador de tarefas. A sua inserção no arquivo `slurm.conf` é **opcional** e a sua adoção é uma escolha do administrador do cluster, dependendo das necessidades específicas de monitoramento e gestão.

Edite o arquivo `/storage/config/gres.conf`.

```
NodeName=masternode Name=gpu File=/dev/nvidia0
NodeName=workernode Name=gpu File=/dev/nvidia0
```

Você pode usar o `nvidia-smi para` descobrir o número que deve ser usado em vez de `0` em `nvidia0`. Você o encontrará à esquerda do nome da GPU.

Caso não tenha GPUs, comente a linha `GresTypes=gpu` no arquivo `/storage/config/slurm.conf`.

No `workernode`, crie o diretório slurm: `sudo mkdir /etc/slurm/`

Copie os arquivos .conf (exceto slurmdbd.conf) em **todas as máquinas (workers e master node)**.

```
sudo cp /storage/slurm/configs_services/cgroup* /etc/slurm/
sudo cp /storage/config/slurm.conf /etc/slurm/
sudo cp /storage/config/gres.conf /etc/slurm/
```

Esse diretório também deve ser criado nos `workers`:

```
sudo mkdir -p /var/spool/slurm/d /var/log/slurm
sudo chown slurm /var/spool/slurm/d /var/log/slurm
sudo chmod 755 /var/log/slurm
```

7.2.2.4 Configurar cgroups

```
sudo nano /etc/default/grub
```

Para implementar as limitações de memória dos trabalhos e usuários do SLURM. Defina cgroups de memória em **todos os workers** com a adição da linha abaixo no arquivo `/etc/default/grub`:

```
GRUB_CMDLINE_LINUX_DEFAULT="cgroup_enable=memory systemd.unified_cgroup_hierarchy=0"
```

Então

```
sudo update-grub
```

Nota Importante: Se `masternode` for também um nó de trabalho (`workernode`), as alterações no grub devem ser executadas também no `masternode`.

7.2.3 Configurando o envio de e-mails

Para configurar o `sSMTP` para enviar emails através do servidor SMTP do Gmail usando uma senha de aplicativo (senha de app), siga os passos abaixo. O Gmail requer autenticação usando SSL/TLS para enviar emails, e a senha de aplicativo é necessária para aplicativos de terceiros como o `sSMTP`.

Você pode ajustar os passos para utilizar o seu SMTP de preferência.

Os procedimentos a seguir devem ser executados no `masternode`.

7.2.3.1 Gerar Senha de Aplicativo no Gmail

Para permitir que o `sSMTP` envie emails através do Gmail, você precisa gerar uma senha de aplicativo. Siga estes passos:

- **Passo 1:** Faça login na sua conta do Gmail.
- **Passo 2:** Acesse a página de Gerenciamento de Conta Google em https://myaccount.google.com/apppasswords
- **Passo 7:** Digite um nome para identificar o `sSMTP`, por exemplo, "ssmtp".
- **Passo 8:** Clique em "Criar".

Anote a senha de aplicativo gerada. Esta será a senha que você deve usar no campo `AuthPass` no arquivo `/etc/ssmtp/ssmtp.conf`.

7.2.3.2 Configuração inicial

1. Instale o pacote ssmtp:

```
sudo apt-get install -y ssmtp
```

2. Configure o arquivo de configuração `/etc/ssmtp/ssmtp.conf` com suas credenciais de e-mail:

```
root=seu_email@gmail.com
mailhub=smtp.gmail.com:587
hostname=seu_hostname
AuthUser=seu_email@gmail.com
AuthPass=sua_senha_do_gmail
UseTLS=YES
UseSTARTTLS=YES
```

Substitua `seu_email@gmail.com`, `sua_senha_do_gmail` e `seu_hostname` pelos valores apropriados.

3. Certifique-se de que o Gmail está configurado para permitir "Aplicativos menos seguros" ou "Acesso a app não seguro" nas configurações de segurança da sua conta do Google.

7.2.3.3 Configuração do script

Copie o script `/storage/slurm/scripts/slurm-email.sh` para `/storage/scripts/slurm-email.sh`:

```
sudo mkdir -p /storage/scripts
sudo cp /storage/slurm/scripts/slurm-email.sh /storage/scripts/slurm-email.sh
```

Modifique a permissão do arquivo para permitir execução

```
sudo chmod +x /storage/scripts/slurm-email.sh
```

Este script extrai informações relevantes do corpo da mensagem enviada pelo Slurm, como o ID do trabalho, nome e status. Em seguida, ele obtém os caminhos dos arquivos de saída e erro do Slurm, lê seu conteúdo (se existirem) e cria um corpo de e-mail em formato HTML contendo essas informações.

Edite o arquivo `/storage/scripts/slurm-email.sh` com as suas preferências.

7.2.3.4 Considerações de segurança

Pressupõe que o usuário `slurm` tem permissão para executar o comando `sudo cat` sem senha para ler os arquivos de saída e erro do Slurm. Isso é feito adicionando uma linha ao arquivo `/etc/sudoers`:

1. Abra o arquivo `/etc/sudoers` para edição usando o comando `visudo`:

```
sudo visudo
```

2. Adicione a seguinte linha no final do arquivo `/etc/sudoers`:

```
slurm ALL=(root) NOPASSWD: /bin/cat /storage/home/*
```

> **Caution**
>
> Lembre-se de que é importante revisar cuidadosamente as permissões concedidas e garantir que elas sejam as mínimas necessárias para a funcionalidade desejada, seguindo os princípios de menor privilégio e necessidade de conhecer.

7.2.3.5 Exemplo de uso [só funcionará após a completa configuração do Slurm]

A configuracão de e-mail faz uso dos seguintes parâmetros: `--job-name`, `--output`, `--error`

1. Abra o arquivo `/etc/sudoers` para edição usando o comando `visudo`:

```
#!/bin/bash
#SBATCH --job-name=meu_job
#SBATCH --output=meu_job.out
#SBATCH --error=meu_job.err
#SBATCH --partition=fila1
#SBATCH --nodes=1
#SBATCH --ntasks=2
#SBATCH --cpus-per-task=1
#SBATCH --mem=1G
#SBATCH --time=01:00:00
#SBATCH --mail-type=ALL
#SBATCH --mail-user=email@exemplo.com

date
echo "Job iniciado no nó: $(hostname)"
echo "Usando $(nproc) núcleos de CPU"
lsb_release -a
echo "Job finalizado"
```

7.3 Iniciar Slurm

Reinicie as máquinas antes de prosseguir (`masternode` e `workenode`).

```
sudo reboot
```

7.3.1 Em `masternode`

```
sudo systemctl restart slurmctld
sudo systemctl restart slurmdbd
```

Se o `masternode` for um nó de computação:

```
sudo systemctl restart slurmd
```

7.3.2 Em worknode:

```
sudo systemctl restart slurmd
```

7.3.2 Em todos os nós:

```
sudo apt update
sudo apt upgrade
sudo apt autoremove
```

8. Logs

Se algo não funcionar, você poderá encontrar os registros de `slurmctld`, `slurmdbd` e `slurmd` em `/var/log/slurm/`.

9. Scripts

Em `/storage/slurm/scripts/`, tem-se scripts utéis para a gestão cotidiana dos usuários da máquina, além de um script de teste da solução (`script_slurm_hostname.sh`).

O script de teste serve para verificar se o `slurm` funciona. O script executa `srun hostname`, que basicamente imprimiria o nó no qual o trabalho foi iniciado.

Você precisará mover o arquivo para o diretório `/storage/home`.

Dentro do script, altere:

`partition`,

`nodelist` (escolha em qual nó será executado),

Em seguida, você pode executar o script com:

```
sudo mkdir -p /storage/home
cd /storage/home
sudo cp /storage/slurm/scripts/script_slurm_hostname.sh .
sbatch script_slurm_hostname.sh
```

9.1 Conteúdo do arquivo script_slurm_hostname.sh

```
#!/bin/bash
#SBATCH --job-name=script_slurm_hostname
#SBATCH --partition=debug
#SBATCH --nodelist=workernode
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=2
#SBATCH --gres=gpu:1
#SBATCH --time=00:00:30
srun hostname
```

10. Gerenciando usuários

Nessa seção são apresentadas rotinas comuns de gestão de usuários

10.1 Adicionando usuários

Utilize o script `add_user.sh` para adicionar usuários novos:

```
#!/bin/bash

# Verifica se o script foi executado com privilégios de superusuário
if [ "$(id -u)" -ne 0 ]; then
    echo "Este script precisa ser executado como root."
    exit 1
fi

# Verifica se o nome do usuário foi fornecido como argumento
if [ -z "$1" ]; then
    echo "Uso: $0 nome_do_usuario [data_de_suspensao] [grupo]"
    echo "     data_de_suspensao deve ser fornecida no formato AAAA-MM-DD"
    echo "     grupo é opcional, o padrão é 'default'"
    exit 1
fi

# Atribui o nome do usuário à variável USERNAME
USERNAME=$1

# Verifica se a data de suspensão foi fornecida como argumento
SUSPENSION_DATE=$2

# Atribui o nome do grupo à variável GROUP ou usa 'default' como padrão
GROUP=${3:-default}

# Gera uma senha aleatória de 12 caracteres
PASSWORD=$(openssl rand -base64 12)

# Verifica se o grupo especificado existe e cria se necessário
if ! getent group "$GROUP" > /dev/null; then
    groupadd "$GROUP"
    echo "Grupo '$GROUP' criado."
fi

# Criando a pasta home para os usuarios
sudo mkdir -p /storage/home/

# Cria o usuário com o diretório home especificado e adiciona ao grupo especificado
useradd -m -d /storage/home/"$USERNAME" -s /bin/bash -g "$GROUP" "$USERNAME"

# Define a senha para o usuário
```

```
echo "$USERNAME:$PASSWORD" | chpasswd

# Exige que o usuário mude a senha no primeiro login
chage -d 0 "$USERNAME"

# Se uma data de suspensão foi fornecida, define a data de expiração da conta
if [ -n "$SUSPENSION_DATE" ]; then
    chage -E "$SUSPENSION_DATE" "$USERNAME"
    echo "A conta do usuário '$USERNAME' será suspensa em $SUSPENSION_DATE."
fi

# Exibe a senha gerada
echo "Usuário '$USERNAME' foi criado com sucesso."
echo "Senha temporária: $PASSWORD"
echo "Grupo: $GROUP"
```

Como Executar o Script

Salve o script em um arquivo chamado `add_user.sh`, dê permissão de execução e execute-o com os argumentos apropriados:

```
chmod +x add_user.sh
sudo ./add_user.sh nome_do_usuario [data_de_suspensao] [grupo]
```

Exemplos:

1. Para criar um usuário chamado `joao` que será suspenso em `2024-12-31` e adicionar ao grupo `alunos-hpc-2024-4`:

```
sudo ./add_user.sh joao 2024-12-31 alunos-hpc-2024-4
```

1. Para criar um usuário chamado `maria` sem data de suspensão e usar o grupo padrão `default`:

```
sudo ./add_user.sh maria
```

1. Para criar um usuário chamado `ana` sem data de suspensão e adicionar ao grupo `alunos`:

```
sudo ./add_user.sh ana "" alunos
```

10.2 Removendo usuários com contas expiradas

Utilize o script `delete_expired_users.sh` para adicionar usuários novos:

```
#!/bin/bash

# Verifica se o script foi executado com privilégios de superusuário
if [ "$(id -u)" -ne 0 ]; then
    echo "Este script precisa ser executado como root."
```

```
    exit 1
fi

# Verifica se o número de dias foi fornecido como argumento
if [ -z "$1" ]; then
    echo "Uso: $0 numero_de_dias"
    exit 1
fi

# Atribui o número de dias à variável DAYS
DAYS=$1

# Data atual em segundos desde Epoch
CURRENT_DATE=$(date +%s)

# Obtém a lista de todos os usuários e verifica se estão no grupo default
USERS=$(getent passwd | awk -F: '{print $1}')

for USER in $USERS; do
    if id -nG "$USER" | grep -qw "default"; then
        # Obtém a data de expiração da conta do usuário
        EXPIRATION_DATE=$(chage -l -i $USER | grep "Account expires" | awk -F: '{print $2}' | xargs)

        # Se a conta não tem data de expiração, pula para o próximo usuário
        if [ "$EXPIRATION_DATE" == "never" ]; then
            continue
        fi

        # Converte a data de expiração para segundos desde Epoch
        if ! EXPIRATION_DATE_EPOCH=$(date -d "$EXPIRATION_DATE" +%s 2>/dev/null); then
            echo "Data de expiração inválida para o usuário $USER: $EXPIRATION_DATE"
            continue
        fi

        # Calcula a diferença em dias entre a data atual e a data de expiração
        DIFF_DAYS=$(( (CURRENT_DATE - EXPIRATION_DATE_EPOCH) / (60*60*24) ))

        # Se a diferença em dias for maior que o número de dias fornecido, apaga o usuário
        if [ "$DIFF_DAYS" -gt "$DAYS" ]; then
            echo "Apagando usuário $USER (expirado há $DIFF_DAYS dias)"
            userdel -r -f $USER
        fi
    fi
done
```

Como Executar o Script

Salve o script em um arquivo chamado `delete_expired_users.sh`, dê permissão de execução e execute-o com o número de dias como argumento:

```
chmod +x delete_expired_users.sh
sudo ./delete_expired_users.sh numero_de_dias
```

Exemplo:

Para apagar todos os usuários do grupo `default` cujas contas estejam expiradas há mais de 30 dias:

```
sudo ./delete_expired_users.sh 30
```

10.3 Removendo usuários de um grupo

Utilize o script `delete_group_users.sh` para adicionar usuários novos:

```
#!/bin/bash

# Verifica se o script foi executado com privilégios de superusuário
if [ "$(id -u)" -ne 0 ]; then
    echo "Este script precisa ser executado como root."
    exit 1
fi

# Verifica se o nome do grupo foi fornecido como argumento
if [ -z "$1" ]; then
    echo "Uso: $0 nome_do_grupo"
    exit 1
fi

# Atribui o nome do grupo à variável GROUP
GROUP=$1

# Verifica se o grupo existe
if ! getent group "$GROUP" > /dev/null; then
    echo "Grupo '$GROUP' não existe."
    exit 1
fi

# Obtém a lista de todos os usuários do sistema
USERS=$(getent passwd | awk -F: '{print $1}')

# Remove cada usuário do grupo especificado
for USER in $USERS; do
    # Verifica se o usuário pertence ao grupo especificado
    if id -nG "$USER" | grep -qw "$GROUP"; then
        echo "Apagando usuário $USER do grupo $GROUP"
        userdel -r -f "$USER"
    fi
done

```

```
echo "Todos os usuários do grupo '$GROUP' foram removidos."
```

Como Executar o Script

Salve o script em um arquivo chamado `delete_group_users.sh`, dê permissão de execução e execute-o com o nome do grupo como argumento:

```
chmod +x delete_group_users.sh
sudo ./delete_group_users.sh nome_do_grupo
```

Exemplo:

Para excluir todos os usuários do grupo `default`:

```
sudo ./delete_group_users.sh alunos-hpc-2024-4
```

10.4 Adicionando usuários em lote

Utilize o script `add_users.sh` para adicionar usuários novos:

```
#!/bin/bash

# Verifica se o script foi executado com privilégios de superusuário
if [ "$(id -u)" -ne 0 ]; then
    echo "Este script precisa ser executado como root."
    exit 1
fi

# Verifica se o nome do arquivo CSV foi fornecido como argumento
if [ -z "$1" ]; then
    echo "Uso: $0 caminho_do_arquivo_csv [nome_do_grupo]"
    exit 1
fi

CSV_FILE=$1
GROUP=${2:-default}

# Verifica se o arquivo CSV existe
if [ ! -f "$CSV_FILE" ]; then
    echo "Arquivo CSV não encontrado: $CSV_FILE"
    exit 1
fi

# Função para criar o usuário e definir configurações
create_user() {
    local NAME=$1
    local ID=$2
    local EMAIL=$3
```

```
    local USERNAME=$4
    local MAXJOBS=$5
    local MAXSUBMITJOBS=$6
    local MAXWALL=$7
    local STORAGEQUOTA=$8
    local QUOTA=$9

    USERNAME=$(echo "$USERNAME" | tr '[:upper:]' '[:lower:]')

    # Gera uma senha aleatória de 12 caracteres
    PASSWORD=$(openssl rand -base64 12)

    # Verifica se o grupo especificado existe e cria se necessário
    if ! getent group "$GROUP" > /dev/null; then
        groupadd "$GROUP"
        echo "Grupo '$GROUP' criado."
    fi

    # Criando a pasta home para os usuários
    sudo mkdir -p /storage/home/

    # Cria o usuário com o diretório home especificado e adiciona ao grupo especificado
    useradd -m -d /storage/home/"$USERNAME" -s /bin/bash -g "$GROUP" "$USERNAME"

    # Define a senha para o usuário
    echo "$USERNAME:$PASSWORD" | chpasswd

    # Exige que o usuário mude a senha no primeiro login
    chage -d 0 "$USERNAME"

    # Define cotas na unidade de armazenamento
    sudo setquota -u $USERNAME $STORAGEQUOTA $STORAGEQUOTA 0 0 /storage
    sudo setquota -u $USERNAME $QUOTA $QUOTA 0 0 /

    # Configurações do SLURM para o usuário
    sacctmgr add user name=$USERNAME DefaultAccount=$GROUP MaxJobs=$MAXJOBS MaxSubmitJobs=$MAXSUBMITJOBS MaxWall=$MAXWALL

    # Exibe a senha gerada
    echo "Usuário '$USERNAME' foi criado com sucesso."
    echo "Senha temporária: $PASSWORD"
    echo "Grupo: $GROUP"
}

# Lê o arquivo CSV e cria usuários
while IFS=, read -r NAME ID EMAIL USERNAME MAXJOBS MAXSUBMITJOBS MAXWALL STORAGEQUOTA QUOTA; do
    # Ignora linhas de comentário
    if [[ "$NAME" == \#* ]]; then
        continue
    fi
```

```
    create_user "$NAME" "$ID" "$EMAIL" "$USERNAME" "$MAXJOBS" "$MAXSUBMITJOBS" "$MAXWALL" "$STORAGEQUOTA" "$QUOTA"
done < "$CSV_FILE"
```

Como Executar o Script

Salve o script em um arquivo, por exemplo, `add_users.sh`, e torne-o executável:

```
chmod +x add_users.sh
```

Exemplo:

Considere o seguinte arquivo `usuarios.csv` de exemplo:

```
# Nome,ID,Email,Username,MaxJobs,MaxSubmitJobs,MaxWall,StorageQuota,Quota
João Silva,001,joao.silva@exemplo.com,joaosilva,10,20,60,150G,5G
Joana Souza,002,joana.souza@exemplo.com,joanasouza,15,30,120,200G,10G
Alice Ferreira,003,alice.ferreira@exemplo.com,aliceferreira,5,10,30,100G,2G
```

Execute o script passando o caminho para o arquivo CSV e opcionalmente o nome do grupo:

```
# Usando o grupo padrão 'default'
sudo ./add_users.sh usuarios.csv

# Usando um grupo específico
sudo ./add_users.sh usuarios.csv nome_do_grupo
```

Considerações Finais

Esse documento apresenta uma configuração funcional e os comandos que devem permitir que os usuários utilizem o Slurm de maneira eficiente em seu ambiente HPC. Dependendo das necessidades específicas e das políticas do ambiente, você pode/deve ajustar e expandir essa configuração.

Caso não tenha mais nó de computação para ser configurado, o diretório `/storage/slurm` pode ser removido. Caso deseje mantê-lo, defina uma permissão mais restrita, por segurança.