Capítulo 3

DevOps para HPC: Como configurar um cluster para uso compartilhado

Lucas Leandro Nesi, Lucas Mello Schnorr
Universidade Federal do Rio Grande do Sul

Resumo

O compartilhamento de recursos para a execução de aplicações de alto desempenho é uma tarefa considerável. Requisitos como controle de usuários, compartilhamento de dados, assim como a reserva e isolamento de recursos são cruciais no sistema. Neste minicurso, apresentaremos um conjunto de softwares que pode ser utilizado para resolver tais desafios, sempre demonstrando as configurações possíveis para adequar o sistema a ser compartilhado para os requisitos dos usuários. A pilha de software é inspirada naquela utilizada no PCAD (Parque Computacional de Alto Desempenho) do Grupo de Processamento Paralelo e Distribuído da Universidade Federal do Rio Grande do Sul.

3.1. Introdução

O agrupamento e compartilhamento de recursos computacionais entre diversos usuários é um passo essencial para garantir a boa utilização dos mesmos, saciando as demandas dos utilizadores. Com um compartilhamento bem feito, pode-se garantir uma justa utilização dos recursos por diversos usuários sem que suas cargas de trabalho interfiram umas nas outras. A agregação de diversos nós computacionais é conhecida como *cluster*. Cargas de trabalho de Computação de Alto Desempenho (CAD) normalmente utilizam um ou mais nós para a execução de uma tarefa finita de um usuário.

Para um bom funcionamento, o ambiente de compartilhamento deve proporcionar algumas características básicas, entre elas, um sistema de autenticação e um sistema de controle de recursos. O minicurso considera **os seguintes objetivos** na construção de uma infraestrutura compartilhada para CAD. (i) A uniformidade dos usuários entre nós com um sistema de autenticação unificado. Isto garante o mesmo perfil (UID, GID, caminho da `$HOME`) em todas as máquinas. (ii) O compartilhamento da `$HOME` via um sistema de arquivos em rede para todos os nós. Desta forma, existe uma unificação virtual da infraestrutura para os usuários de forma transparente. Não sendo necessário realizar cópias de

arquivos manualmente via rede entre máquinas. (iii) Um sistema que controla a utilização dos recursos pelos usuários, como um que ofereça escalonamento de tarefas. (iv) Que a infraestrutura computacional tenha algumas características mínimas de segurança.

Para atingir tais objetivos, a organização da infraestrutura proposta neste minicurso segue uma abordagem centralizada. Ela é estruturada com a presença de um nó controlador, que sustenta a maioria dos serviços essenciais e controla os nós de computação denominados clientes. Considera-se adicionalmente o acesso dos usuários direto ao sistema operacional, sem virtualização. É responsabilidade dos usuários a instalação de bibliotecas necessárias para a compilação de suas aplicações, incluindo a própria execução destas em nível de usuário. Um sistema operacional único já atende a maioria dos usuários e facilita a manutenção, sendo um primeiro passo para o compartilhamento dos recursos.

Este minicurso é inspirado no caso real do Parque Computacional de Alto Desempenho (PCAD)[1], mantido pelo Grupo de Processamento Paralelo e Distribuído (GPPD) do Instituto de Informática da UFRGS. O PCAD está em funcionamento com configurações similares as propostas desde 2018 (Nesi et al. 2019, Nesi et al. 2020).

Este capítulo está organizado da seguinte forma. A Seção 3.2 apresenta um conjunto de softwares que podem ser utilizados para cada objetivo e as razões para estas escolhas. A Seção 3.3 discute os passos iniciais da instalação, estrutura e configuração. As próximas seções discutem a instalação e configuração de softwares específicos no controlador: Seção 3.4 para segurança, Seção 3.5 para autenticação unificada, Seção 3.6 para o sistema de arquivos distribuído, Seção 3.7 para compartilhar os recursos. A Seção 3.8 discute a instalação e configuração de um nó computacional. A Seção 3.9 apresenta outros softwares úteis que podem ser configurados. Finalmente, a Seção 3.10 conclui este documento.

3.2. Softwares para a infraestrutura

Esta seção discute os vários programas e serviços que podem ser utilizados para atingir os objetivos mencionados na introdução. Primeiramente, utiliza-se o sistema operacional Linux com a distribuição Debian 12, por ser *open-source*, possuir uma forte comunidade, e contar com várias aplicações em seu gerenciador padrão de pacotes. Considera-se que vários usuários estão familiarizados com sistemas baseados no Debian.

3.2.1. Sistema de autenticação

Os programas existentes para oferecer um sistema de autenticação unificado possuem diferentes objetivos e sua disponibilidade e estabilidade em diversos sistemas é variada. Uma das soluções mais comuns é o protocolo LDAP para o compartilhamento de informações de diretório. Neste caso, ele pode ser utilizado para armazenar informações de usuários, senhas e chaves ssh. Um cliente LDAP se comunica com um controlador LDAP para recuperar estas informações. Por ser concebido na década de 1990 e amplamente utilizado, diversas implementações do protocolo LDAP existem para diferentes sistemas operacionais e distribuições. No caso do Debian, uma opção é o OpenLDAP, uma implementação aberta. Como alternativa mais recente, pode-se citar FreeIPA, mantida pela Red Hat para seus sistemas. Ela possui um suporte experimental no Debian atualmente.

[1]<https://gppd-hpc.inf.ufrgs.br/>

3.2.2. Sistema de arquivos distribuído

Um dos sistemas de arquivos distribuídos mais simples e utilizado é o *Network File System* (NFS). Ele possui um controlador que exporta um diretório do sistema de arquivos local para clientes remotos de forma transparente. Atualmente o NFS faz parte do kernel do Linux, condição que melhora seu desempenho, disponibilidade e facilidade de uso comparada com outras alternativas. Entretanto, em ambientes CAD é comum a utilização de sistemas de arquivos paralelos para maximizar o desempenho, sendo o Lustre um exemplo. Apesar disto, seus requisitos mínimos (nós dedicados para armazenamento e controle de requisições) e sua complexidade de instalação maior que o NFS são desvantagens em *clusters* de menor porte. Na necessidade destes sistemas de arquivos paralelos, os passos que descrevem a instalação do NFS neste minicurso podem ser evitadas.

3.2.3. Sistema de compartilhamento de recursos

No ambiente de CAD, o gerenciador de recursos e workloads SLURM é bastante utilizado, sendo presente em alguns dos maiores supercomputadores do mundo. Com ele, usuários podem submeter tarefas, que serão escalonadas e isoladas nos recursos pelo SLURM. Outras alternativas existem, como o OAR. Umas das principais diferenças do OAR em relação ao SLURM é a possibilidade da reserva de recursos por usuários e seu possível forte acoplamento com o Kadeploy, permitindo o boot de imagens customizadas. Considerando a grande utilização do SLURM, sua fácil instalação, e para proporcionar aos estudantes do grupo experiência na utilização do mesmo, o PCAD escolheu utilizar o SLURM, estendendo tais razões para este minicurso.

3.3. Prelúdio da Instalação

Esta Seção introduz os passos para a instalação da infraestrutura. Consulte a versão do guia atualizado em: <https://gitlab.com/lnesi/mc-hpc-share>.

3.3.1. Estrutura do sistema de arquivos

O Controlador possui uma `/home` exportada para todas as máquinas via NFS. Cada máquina possui um diretório `/scratch` com um subdiretório por usuário, normalmente `/scratch` é um volume em um disco diferente do `/`, e local à máquina.

3.3.2. Instalação do Debian

O Debian utilizado neste minicurso é o 12. Atualmente o Debian 12 encontra-se em *release candidate*[2]. Durante sua instalação pode-se seguir a estrutura de arquivos sugerida e padrão. Atente-se que o usuário criado durante a instalação (UID 1000) deve ser o mesmo entre controlador e clientes. Assumiremos que trata-se do usuário USER.

3.3.3. Redes do Controlador e dos Clientes

Na configuração do PCAD, as máquinas possuem duas redes. A primeira interface está conectada na rede do Instituto de Informática da UFRGS (com DHCP), permitindo conexão direta com qualquer máquina da instituição e com a internet. A segunda in-

[2]<https://www.debian.org/devel/debian-installer/>

terface de cada máquina é a rede interna da infraestrutura, com IP fixo nos endereços 192.168.30.01/24. Os serviços relacionados a infraestrutura (NFS, LDAP, SLURM) estão somente habilidades na rede interna. Ainda, para as máquinas com IPMI a rede é configurada nos endereços 192.168.10.01/24 quando possível.

A configuração de rede no Debian é feita modificando o arquivo: `</etc/network/interfaces>`. Para habilitar a segunda interface com IP estático é necessário adicionar as seguintes linhas, sendo INTERFACE o nome da segunda interface (verificar com `ip address`) e FINAL o identificar escolhido da máquina.

Adição em `/etc/network/interfaces`

```
auto INTERFACE
iface INTERFACE inet static
  address 192.168.30.FINAL
  network 192.168.30.0
  netmask 255.255.255.0
  broadcast 192.168.30.255
```

É possível ainda uniformizar os domínios na rede, e forçar o uso dos *hostnames* via a segunda interface. É necessário configurar com a rede interna o arquivo `</etc/hosts>`:

Adição em `/etc/hosts`

```
192.168.30.2 controlador
192.168.30.3 client1
```

3.3.4. Pacotes e Configuração Básica

Com a instalação base do Debian é necessário instalar o `sudo` via superusuário (root) e adicionar o usuário não `root` ao `sudoers`. Assumimos que trata-se do USER criado na instalação com UID 1000.

Comandos no terminal do controlador como superusuário (root)

```
apt install sudo git nano
/sbin/adduser USER sudo
```

De volta ao usuário USER, já parte do grupo `sudo`, configuramos o emprego do comando `sudo` sem pedir sua senha, por agilidade, e geramos as chaves internas deste usuário.

Comando no terminal do controlador (usuário normal)

```
echo "%sudo   ALL=(ALL:ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers
ssh-keygen -t rsa -f ~/.ssh/id_rsa -q -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

Aconselha-se copiar a sua chave pública ssh para o usuário USER criado na instalação do Debian no controlador. Execute o próximo comando em sua máquina pessoal.

Comando no terminal de uma máquina pessoal

```
ssh-copy-id USER@CONTROLADOR
```

3.3.5. Nosso repositório com configurações e scripts

Com os pacotes básicos instalados utilizando o superusário, é possível clonar o repositório que acompanha este minicurso utilizando o usuário USER, criado durante a instalação do Debian no controlador. Para clonar o repositório, usa-se este comando no controlador:

Comando no terminal do controlador (usuário normal)

```
git clone https://gitlab.com/lnesi/mc-hpc-share
```

3.3.6. Variáveis utilizas durante a instalação

Algumas variáveis serão referenciadas durante a instalação e utilizadas no script automático da próxima subseção. Considere as seguintes variáveis: `NOME_INFRA` é nome dado para a infraestrutura, `CONTROLADOR_IP` e `CONTROLADOR_DNS` são o IP e o *hostname* do controlador, `CONTROLADOR_LDAP` é o *hostname* do controlador no formato para o LDAP (adicionando *dc=* para cada ponto), `ARQUIVO_SENHA` é o caminho para o arquivo de senha, `REDE_INTERNA` é a rede interna, `SCRIPTPATH` é o caminho absoluto para a pasta script do repositório, `ADMIN_USER` é o usuário USER criado durante a instalação do Debian (UID 1000).

Comando no terminal do controlador (usuário normal)

```
export NOME_INFRA="gppd-hpc"
export CONTROLADOR_IP="192.168.30.2"
export CONTROLADOR_DNS="controlador"
export CONTROLADOR_LDAP="dc=${CONTROLADOR_DNS}"
export ARQUIVO_SENHA="${SCRIPTPATH}/confs/pass"
export REDE_INTERNA="192.168.30.2/24"
export SCRIPTPATH="${SCRIPTPATH:-$HOME/mc-hpc-share/}"
export ADMIN_USER="parque"
```

3.3.7. Instalação via script

Disponibilizamos o script `<scripts/make_controler.sh>` para realizar a instalação automática do controlador seguindo este tutorial. Basta abrir o script e alterar as variáveis discutidas na seção anterior e o arquivo de senha `<scripts/confs/pass>`. Este arquivo não deve conter o caractere *newline* (utilize a opção -L no nano, `:set binary` no vim). Para executar o script entre no repositório git recém clonado e utilize o bash no arquivo `<scripts/make_controler.sh>`:

Comando no terminal do controlador (usuário normal)

```
cd mc-hpc-share
bash scripts/make_controler.sh
```

Após a conclusão deste script será necessário um reboot manual. Feita a reinicialização manual da máquina, continue a instalação com o segundo script:

Comando no terminal do controlador (usuário normal)

```
bash scripts/make_controler_2.sh
```

3.4. Segurança básica do controlador

Esta Seção apresenta três configurações para melhorar a segurança do controlador.

3.4.1. Firewall

Como possivelmente o controlador estará exposto à internet, é recomendado controlar quais serviços estão realmente expostos. Pode-se instalar o `ufw` para gerenciar o `iptables`, adicionar o comportamento padrão de negar qualquer conexão, adicionar a regra permitindo acesso à porta 22 (ssh) e 80 (http), e liberar tudo para a rede interna.

Comando no terminal do controlador (usuário normal)

```
sudo apt install -y ufw
sudo ufw default deny incoming
sudo ufw allow 22
sudo ufw allow 80
sudo ufw allow from $REDE_INTERNA to any
sudo ufw --force enable
```

3.4.2. Fail2Ban

O `fail2ban` é um software que monitora os arquivos de logs de serviços comuns (ssh, apache) e bane conexões que tentaram e falharam em se autenticar nestes serviços. Um usuário pode personalizar suas configurações criado o arquivo `</etc/fail2ban/jail.local>`. Por exemplo, considere um banimento de 240 horas, se dentro de 6 horas um mesmo IP tentou e falhou três vezes para se conectar em um serviço.

Arquivo `/etc/fail2ban/jail.local`

```
[DEFAULT]
bantime  = 240h
findtime  = 6h
maxretry = 3
```

Os comandos em seguida podem ser utilizados para instalação e configuração.

Comando no terminal do controlador (usuário normal)

```
sudo apt install -y rsyslog fail2ban
sudo cp $SCRIPTPATH/confs/jail.local /etc/fail2ban/jail.local
sudo chown root:root /etc/fail2ban/jail.local
sudo chmod 644 /etc/fail2ban/jail.local
sudo service fail2ban restart
```

3.4.3. Audit

O *Audit* é o sistema de auditoria presente no kernel do Linux. Ele permite o registro de eventos baseados nos arquivos de configuração. Podemos remover alguns eventos muito recorrentes e adicionar todos os comandos executados por um usuário nos registros alterando o arquivo `</etc/audit/rules.d/audit.rules>`. Considere a configuração proposta em `<scripts/confs/audit.rules>`.

Para instalar o serviço de acesso à auditoria, desativar que os registros apareçam no *journal* (para não sobrecarregar de mensagens, para acessá-los utilizar o comando *ausearch*):

Comando no terminal do controlador/cliente (usuário normal)

```
sudo -E apt install -y auditd
cat $SCRIPTPATH/confs/audit.rules | \
   sudo tee -a /etc/audit/rules.d/audit.rules
sudo systemctl restart auditd
sudo systemctl stop systemd-journald-audit.socket
sudo systemctl disable systemd-journald-audit.socket
sudo systemctl mask systemd-journald-audit.socket
```

Para buscar os comandos executados pelos usuários hoje:

Exemplo de comando para controlador/cliente (usuário normal)

```
sudo ausearch -ts today
```

3.5. Sistema unificado de autenticação

Esta Seção descreve a instalação dos pacotes selecionados e discutidos em um Debian 12.

3.5.1. Controlador: Instalação do LDAP - OpenLDAP

O LDAP já se encontra nos repositórios oficiais do Debian. Comando de instalação:

Comando no terminal do controlador (usuário normal)

```
sudo apt install -y slapd ldap-utils
```

Para configurar o controlador do LDAP:

Comando no terminal do controlador (usuário normal)

```
sudo dpkg-reconfigure slapd
```

Utilize as seguintes configurações:

Configuração	Sugestão de Valor
Omit OpenLDAP server configuration?	No
DNS domain name: provavelmente:	**$CONTROLADOR_DNS**
Organization name:	**$NOME_INFRA**
Administrator password:	**$SENHA_SERVICOS**
Database backend?	MDB
Remove the database when slapd is purged?	No
Move old database?	Yes

Considerando algumas questões de segurança[3] é necessário aplicar os comandos LDAP

[3]<https://www.openldap.org/doc/admin23/security.html>

do arquivo <scripts/confs/ldap_disable_bind_anon.ldif> para desativar acesso anônimo e requerer o uso de autenticação:

Comando no terminal do controlador (usuário normal)

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f scripts/confs/ldap_anon.ldif
```

3.5.2. Controlador: Informação de chave ssh nos usuários

Primeiramente é necessário adicionar ao LDAP scheme a variável que guardará a chave ssh. Aplicando o arquivo <scripts/confs/openssh-lpk.ldif> utilizando:

Comando no terminal do controlador (usuário normal)

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f scripts/confs/openssh-lpk.ldif
```

3.5.3. Controlador: Gerenciamento auxiliar (web) do LDAP - phpLDAPadmin

O phpLDAPadmin é uma interface web para a manipulação dos dados armazenados no LDAP. Para a instalação do pacote:

Comando no terminal do controlador (usuário normal)

```
sudo apt install -y phpldapadmin
```

As próximas configurações são para melhorar a qualidade de vida do administrador no uso do phpLDAPadmin. Conside o patch <scripts/confs/diff_phpldapadmin_config> para ser aplicado no arquivo </etc/phpldapadmin/config.php>. Alterando as variáveis necessárias.

Comando no terminal do controlador (usuário normal)

```
sudo patch -l /etc/phpldapadmin/config.php \
          $SCRIPTPATH/confs/diff_pla_config
sudo sed -i "s/NOME_INFRA/${NOME_INFRA}/g" /etc/phpldapadmin/config.php
sudo sed -i "s/CONTROLADOR_LDAP/${CONTROLADOR_LDAP}/g" \
          /etc/phpldapadmin/config.php
```

Aplicar os seguintes patchs. Primeiro no arquivo </etc/phpldapadmin/templates/creation/posixAccount.xml> para adicionar os campos e-mail, chave ssh, e forçar o uso do bash. Então no arquivo </usr/share/phpldapadmin/htdocs/create_confirm.php> para melhorar a confirmação de adição de usuários. No arquivo </usr/share/phpldapadmin/lib/ds_ldap_pla.php> para mudar o padrão de senha. Por último no arquivo </usr/share/phpldapadmin/lib/PageRender.php> para automaticamente gerar uma senha.

Comando no terminal do controlador (usuário normal)

```
sudo patch -l /etc/phpldapadmin/templates/creation/posixAccount.xml \
           $SCRIPTPATH/confs/diff_pla_posix
sudo patch -l /usr/share/phpldapadmin/htdocs/create_confirm.php \
           $SCRIPTPATH/confs/diff_pla_create
```

```
sudo patch -l /usr/share/phpldapadmin/lib/ds_ldap_pla.php \
           $SCRIPTPATH/confs/diff_pla_pla
sudo patch -l /usr/share/phpldapadmin/lib/PageRender.php \
           $SCRIPTPATH/confs/diff_pla_page
```

3.5.4. Controlador: Primeiro uso do LDAP e do phpLDAPadmin

Com estas configurações, o phpldapadmin está em <http://controlador/phpldapadmin/>.

O login é definido por "cn=admin,CONTROLADOR_LDAP"(deve estar autopreenchido) e a senha do LDAP. É necessário criar as primeiras entidades que vão organizar os diretórios do LDAP. Após logado criar uma entrada “Generic: Organisational Unit”, que será a organização com a qual as contas estarão associadas. Pode-se utilizar o nome da infraestrutura. Então, crie uma *child entry* na organização e adicione um entrada “Generic: Posix Group”, chamado “members”. Este será o grupo principal de membros da infraestrutura. Pode-se agora cadastrar um usuário criando uma “/child entry/ Generic: User Account” no grupo “members”. O “User ID” será o identificador Linux do usuário. Um primeiro usuario a ser criado é o usuário com o “User ID” “slurm”, sem chave ssh.

3.5.5. Controlador: Login de novos usuários: autocriação da HOME e banner

Edite o arquivo `</etc/pam.d/common-session>` para configurar a autocriação da `home`. Adicionando na última linha:

Adição em `/etc/pam.d/common-session`

```
session required    pam_mkhomedir.so skel=/etc/skel umask=0077
```

Desta forma, a pasta `/etc/skel` será copiada para todo novo usuário que entrar no sistema. Por exemplo, podemos alterar `</etc/skel/.bashrc>` para gerar uma chave que será utilizada internamente na infraestrutura:

Comando no terminal do controlador (usuário normal)

```
sudo patch -l /etc/skel/.bashrc $SCRIPTPATH/confs/diff_skel_bashrc
ssh-keygen -t rsa -f ~/.ssh/id_rsa -q -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
```

Para adicionar um Banner de entrada, adicione as seguintes linhas em `</etc/pam.d/common-session>` e criar o arquivo `</etc/motd>` com a mensagem desejada.

Adição em `/etc/pam.d/common-session`

```
session [default=1 success=ignore] pam_succeed_if.so quiet user ingroup
   → members
session optional pam_motd.so motd=/etc/motd
```

3.5.6. Controlador e Cliente: Login de usuários via LDAP e chaves ssh

Esta configuração se aplica ao controlador e aos clientes. Pode-se utilizar o `PAM` (Linux Pluggable Authentication Modules) para consultar os usuários no LDAP. Para isso, utilizamos um sistema de caches dos nomes com o *nscd* e do LDAP *nslcd*. Para instalação:

Comando no terminal do controlador/cliente (usuário normal)

```
sudo apt install libnss-ldapd libpam-ldapd nscd nslcd nslcd-utils
```

Para fazer todas as configurações é necessário utilizar:

Comando no terminal do controlador/cliente (usuário normal)

```
sudo dpkg-reconfigure nslcd
```

Considere as seguintes configurações:

Configuração	Sugestão de Valor
LDAP server Uniform Resource Identifier:	ldap://**$CONTROLADOR_IP/**
Distinguished name of the search base:	**$CONTROLADOR_LDAP**
Name services to configure:	passwd, group and shadow
LDAP authentication to use:	Simple
LDAP database user	cn=admin,**$CONTROLADOR_LDAP**
LDAP root account password:	**$SENHA_SERVICOS**
Use StartTLS	No[4]

Estas configurações habilitam uma base de usuários unificada. Para realizar login com chaves ssh armazenadas no LDAP é necessário criar um script que consulte elas e informe ao serviço do sshd. Primeiramente copiando a senha para um arquivo local:

Comando no terminal do controlador/cliente (usuário normal)

```
sudo cp $ARQUIVO_SENHA /etc/infra_access
sudo chown root:root /etc/infra_access
sudo chmod 700 /etc/infra_access
```

Agora, definindo o script `</usr/bin/authp>` com o script de exemplo em `<scripts/authp>`. Modificando as variáveis corretamente e a permissão do script.

Comando no terminal do controlador/cliente (usuário normal)

```
sudo cp $SCRIPTPATH/authp /usr/bin/authp
sudo -E sed -i "s/CONTROLADOR_IP/${CONTROLADOR_IP}/g" /usr/bin/authp
sudo -E sed -i "s/CONTROLADOR_LDAP/${CONTROLADOR_LDAP}/g" /usr/bin/
   → authp
sudo chown root:root /usr/bin/authp
sudo chmod 700 /usr/bin/authp
```

Modificar para o sshd consultar o script, adicionar o arquivo `<scripts/confs/custom_sshd.conf>` em `</etc/ssh/sshd_config.d>`. Consulte este arquivo para verificar outras configurações do ssh.

[4]Para simplificação do tutorial apenas, em instalações complexas deve-se considerar sua configuração.

Comando no terminal do controlador/cliente (usuário normal)

```
sudo cp $SCRIPTPATH/confs/custom_sshd.conf /etc/ssh/sshd_config.d/
sudo systemctl restart sshd
```

Uma configuração no </etc/pam.d/su> para permitir que somente o usuário base possa utiliza o comando su.

Comando no terminal do controlador/cliente (usuário normal)

```
sudo -E sed -i  "s/# auth       required   pam_wheel.so/auth
   → required   pam_wheel.so group=$ADMIN_USER/" /etc/pam.d/su
```

3.5.7. Checkpoint de progresso

Foram configurados a gestão unificada de usuários. É importante um reboot do sistema:

Comando no terminal do controlador/cliente (usuário normal)

```
sudo reboot
```

3.6. Sistema de arquivos distribuído

Esta seção explica a instalação do controlador NFS.

3.6.1. Controlador: Gerenciador NFS

Para instalar o NFS:

```
sudo apt-get install -y nfs-kernel-server
```

Para compartilhar um diretório é necessário declará-lo para cada máquina cliente no arquivo </etc/exports>. Um exemplo se encontra em seguida. Uma linha define uma exportação, onde o primeiro item, /home, é o diretório local para compartilhar, REDE_INTERNA é o IP ou hostname da máquina que faz o compartilhamento, neste caso sendo permitido o compartilhando para todas as máquinas desta rede. A opção rw é para permitir escrita e leitura; sync para responder a requisições somente quando as alterações forem confirmadas em disco; no_root_squash para manter o UID de root nas requisições; no_subtree_check para desabilitar o check na subtree, opção recomendada para /home e para compartilhamentos que iniciem na raiz do volume. Outras opções estão descritas no manual do exports[5].

Adição em /etc/exports

```
/home       $REDE_INTERNA(rw,sync,no_root_squash,no_subtree_check)
```

Após cara alteração é necessário reiniciar o gerenciador do NFS:

Comando no terminal do controlador (usuário normal)

```
sudo systemctl restart nfs-kernel-server
```

[5]<https://linux.die.net/man/5/exports>

3.6.2. Cliente: Configurando o Cliente NFS

Primeiramente no cliente é necessário instalar o seguinte pacote:

Comando no terminal do Cliente (usuário normal)

```
sudo apt-get install nfs-common
```

Para adicionar o diretório no cliente é necessário adicionar em `/etc/fstab` a entrada do volume. Por exemplo, a seguinte linha, onde CONTROLADOR_DNS é o IP ou hostname do gerenciador, `:/home` é o caminho do gerenciador a ser compartilhado (o mesmo definido em `/etc/exports`) e o segundo `/home` é o caminho de montagem no cliente.

Adição em `/etc/fstab`

```
CONTROLADOR_DNS:/home       /home      nfs auto,nofail,noatime 0 0
```

Para validar a configuração sem reiniciar a máquina pode-se utilizar:

Comando no terminal do Cliente (usuário normal)

```
sudo mount -a
```

3.7. Compartilhamento de Recursos

Esta Seção descreve a instalação do gerenciador de trabalhos SLURM no controlador.

3.7.1. Controlador: Instalação do SLURM

Instalação de dependências no controlador:

Comando no terminal do controlador (usuário normal)

```
sudo apt install -y build-essential pkg-config libpam0g-dev \
     libmunge-dev munge mariadb-server libopenmpi-dev libmariadb-dev \
     libnuma-dev libjson-c-dev libyaml-dev hwloc libhwloc-dev \
     libcurl4-openssl-dev libdbus-1-dev liblz4-dev \
     libhttp-parser-dev libreadline-dev
```

O Slurm utiliza o munge para autenticações. Para isso, é necessário criar uma chave compartilhada entre todas as máquinas. A geração é feita somente uma vez no gerenciador e a chave será copiada para os clientes:

Comando no terminal do controlador (usuário normal)

```
sudo sh -c 'dd if=/dev/urandom bs=1 count=1024 > /etc/munge/munge.key'
sudo chown munge:munge /etc/munge/munge.key
sudo chmod 600 /etc/munge/munge.key
sudo cp /etc/munge/munge.key /home/munge.key
sudo chown root:root /home/munge.key
sudo chmod 600 /home/munge.key
```

Iniciando o serviço do munge:

Comando no terminal do controlador (usuário normal)

```
sudo systemctl restart munge
```

Baixando a última versão do Slurm e descompactando:

Comando no terminal do controlador (usuário normal)

```
wget https://download.schedmd.com/slurm/slurm-23.02.0.tar.bz2
mkdir slurm
tar --bzip -x -f slurm*tar.bz2 -C ./slurm --strip-components=1
```

Realizando a compilação e instalação:

Comando no terminal do controlador (usuário normal)

```
cd slurm
./configure
make -j
sudo make install
```

É necessário **criar um usuário slurm no LDAP** sem chave ssh para este usuário ter o mesmo UID em todas as máquinas. Consulte o procedimento na Seção 3.5.4.

Para configurar o controlador do Slurm, o primeiro passo é configurar o banco de dados do mesmo. Crie o arquivo `</usr/local/etc/slurmdbd.conf>` com o seguinte conteúdo[6], modificando SENHA para uma senha utilizada em seguida no gerenciador do banco de dados.

Arquivo `/usr/local/etc/slurmdbd.conf`

```
AuthType=auth/munge
DbdHost=localhost
DbdPort=6819
LogFile=/usr/local/etc/dbd.log
SlurmUser=slurm
StorageHost=localhost
StorageLoc=slurm_db
StoragePass=SENHA
StoragePort=3306
StorageType=accounting_storage/mysql
StorageUser=slurm
DebugLevel=info
```

É necessário configurar as permissões do arquivo:

Comando no terminal do controlador (usuário normal)

```
sudo chown slurm:root /usr/local/etc/
sudo chown slurm:root /usr/local/etc/slurmdbd.conf
sudo chmod 600 /usr/local/etc/slurmdbd.conf
```

Agora é necessário criar o usuário Slurm no gerenciador do banco, o banco de dados e dar as permissões corretas no Mariadb (*fork* do MySQL). Abra a interface de gerenciamento:

[6]Explicação de configurações extras disponível em <https://slurm.schedmd.com/slurmdbd.conf.html>

Comando no terminal do controlador (usuário normal)

```
sudo mysql -u root
```

E execute os seguintes comandos SQL alterando SENHA para a mesma senha acima:

Comando no terminal do MYSQL no controlador

```
CREATE DATABASE slurm_db;
CREATE USER 'slurm'@localhost IDENTIFIED BY 'SENHA';
GRANT ALL PRIVILEGES ON slurm_db.* TO 'slurm'@localhost;
```

Ainda é necessário alterar algumas variáveis no banco, alterando o arquivo `</etc/mysql/conf.d/mysql.cnf>` e reiniciando o serviço.

Comando no terminal do controlador (usuário normal)

```
echo "[mysqld]
innodb_buffer_pool_size=4096M
innodb_log_file_size=64M
innodb_lock_wait_timeout=900" | sudo tee -a /etc/mysql/conf.d/mysql.cnf
sudo systemctl restart mariadb.service
```

Copiando o serviço do systemd e ligando-o:

Comando no terminal do controlador (usuário normal)

```
sudo cp etc/slurmdbd.service /etc/systemd/system/
sudo systemctl start slurmdbd
sudo systemctl enable slurmdbd
sudo systemctl status slurmdbd
```

O próximo passo é a configuração do gerenciador do Slurm, pelo arquivo `</home/slurm.conf>` que terá um link simbólico com `</usr/local/etc/slurm.conf>`. Um exemplo de configuração básica se encontra em seguida. Algumas explicações de variáveis: `SlurmctldHost` e `ClusterName` são o hostname, ip do controlador e o nome da infraestrutura. As configurações de `Slurmctld` e `Slurmd` são para a execução dos *daemons* dos serviços no controlador e nos clientes. Para a contabilidade de usuários, as variáveis de `AccountingStorage` controlam tanto as permissões de utilização quanto que dados armazenar. Mais ao final do arquivo, existe a entrada `NodeName`, com uma linha descrevendo um nó de computação cliente. Os nós de computação são organizados em partições, explicitamente configurada nas linhas com `PartitionName`. As configurações completas estão disponíveis na documentação do Slurm[7].

Arquivo `/home/slurm.conf`

```
SlurmctldHost=CONTROLADOR_DNS(CONTROLADOR_IP)
ClusterName=NOME_INFRA
MaxJobCount=10000
SlurmUser=slurm
```

[7]<https://slurm.schedmd.com/slurm.conf.html>

```
AuthType=auth/munge
CryptoType=crypto/munge

ProctrackType=proctrack/cgroup

SlurmctldPidFile=/run/slurmctld.pid
SlurmctldPort=6817
SlurmctldDebug=info
SlurmctldLogFile=/usr/local/etc/log.txt
StateSaveLocation=/var/spool/slurm.state

SlurmdPidFile=/run/slurmd.pid
SlurmdPort=6818
SlurmdDebug=info
SlurmdLogFile=/usr/local/etc/node_log.txt
SlurmdSpoolDir=/var/spool/slurmd

TaskPlugin=task/cgroup
TaskPluginParam=None
TrackWCKey=no
SrunPortRange=60001-63000

AccountingStorageEnforce=associations,limits,qos,safe
AccountingStorageHost=localhost
AccountingStoragePort=6819
AccountingStorageType=accounting_storage/slurmdbd
AccountingStoreFlags=job_comment

MinJobAge=2

SelectType=select/cons_res
SelectTypeParameters=CR_CPU
SchedulerType=sched/backfill
SchedulerParameters=sbatch_wait_nodes,salloc_wait_nodes,bf_continue,
    → bf_hetjob_immediate

JobCompType=jobcomp/none

GresTypes=gpu
ReturnToService=1
PrologFlags=contain

#Nodes
NodeName=client1 NodeAddr=192.168.30.3 Sockets=1 CoresPerSocket=1
    → ThreadsPerCore=1 CPUs=1 MemSpecLimit=512 RealMemory=3900 State=
    → UNKNOWN

PartitionName=shared Nodes=client1 Default=YES MaxTime=24:00:00
    → DefaultTime=10:00 State=UP
```

Ainda é necessário definir uma configuração de como o Linux `cgroups` deve ser comportar em cada nó. Definindo o arquivo `</home/cgroup.conf>`.

Arquivo `/home/cgroup.conf`

```
CgroupAutomount=yes
ConstrainCores=yes
ConstrainRAMSpace=yes
```

O próximo passo é mudar as permissões destes arquivos, criar o link simbólico (todas as máquinas vão compartilhar o mesmo arquivo via NFS), copiar o serviço do systemctl do slurmctld, e criar a pasta de spool do daemon do slurm.

Comando no terminal do controlador (usuário normal)

```
sudo chown slurm:root /home/slurm.conf
sudo chmod 644 /home/slurm.conf
sudo ln -s /home/slurm.conf /usr/local/etc/slurm.conf
sudo chown slurm:root /home/cgroup.conf
sudo chmod 644 /home/cgroup.conf
sudo ln -s /home/cgroup.conf /usr/local/etc/cgroup.conf
sudo cp etc/slurmctld.service /etc/systemd/system/
sudo mkdir /var/spool/slurm.state
sudo chown slurm /var/spool/slurm.state
```

Para garantir que o Slurm inicie apenas depois do LDAP e do NFS é necessário adicionar esta configuração no systemd. Criando arquivo `</etc/systemd/system/slurmctld.service.d/override.conf>` com o conteúdo:

Arquivo `/etc/systemd/system/slurmctld.service.d/override.conf`

```
[Unit]
After=network-online.target munge.service remote-fs.target nslcd.
   → service
```

Ou copiando o arquivo do repositório:

Comando no terminal do controlador (usuário normal)

```
sudo mkdir /etc/systemd/system/slurmctld.service.d/
sudo cp $SCRIPTPATH/confs/slurmctld_override.conf /etc/systemd/system/
   → slurmctld.service.d/override.conf
sudo systemctl daemon-reload
```

Após, é necessário inicializar o daemon do controlador do slurm:

Comando no terminal do controlador (usuário normal)

```
sudo systemctl start slurmctld
sudo systemctl enable slurmctld
sudo systemctl restart slurmctld
```

3.7.2. Controlador - Recomendado: Prólogo e Epílogo de trabalhos Slurm

Uma das configurações possíveis no Slurm é a definição de scripts que executaram antes ou depois das tarefas. Uma descrição completa está disponível na página no Slurm[8].

[8]<https://slurm.schedmd.com/prolog_epilog.html>

É recomendado fortemente adicionar as variáveis propostas no módulo PAM do Slurm[9] nos prólogos e epílogos referenciados. Outra possível utilização é a configuração dos nós para o governor "powersave"ou "performance", ou a configuração de algum pacote (como `docker`). A adição de scripts de prólogo e epilogo é feito no `<slurm.conf>` com a adição das seguintes linhas. Lembrando de incluir as permissões de execução.

Adição no arquivo `/home/slurm.conf`

```
Prolog=/usr/local/etc/prolog.sh
Epilog=/usr/local/etc/epilog.sh
```

3.7.3. Controlador - Recomendado: Prioridade dos Jobs

Um dos aspectos fundamentais do Slurm é o escalonamento de tarefas seguindo prioridades. Uma extensiva documentação está disponível numa página do Slurm dedicada[10]. Também incluímos no repositório o arquivo `<scripts/confs/priority.conf>` com os exemplos de configuração de prioridades adotados no PCAD. Estas configurações devem ser adicionadas no `<slurm.conf>`.

3.7.4. Controlador - Opcional: Sistema de e-mail para o Slurm

O Slurm pode alertar os usuários via e-mail em alguns eventos relacionados às tarefas. Para isto, é necessário configurar um script que recebe como segundo parâmetro o endereço de destino (do usuário) e como terceiro parâmetro a mensagem do e-mail. Um exemplo é o script abaixo, configurado em `</usr/bin/smail>` no controlador.

Arquivo `/usr/bin/smail`

```
#!/bin/bash
mail -s "$2" --append="FROM:NOME <infra@instruicao.com>" "$3"
```

É necessária a configuração do `MailProg` no `</home/slurm.conf>`.

3.8. Configurando um nó computacional (Cliente)

A instalação pode ser executada via o script `scripts/make_client.sh`. Atentando-se que este script é um exemplo tendo em vista que no PCAD empregamos Ansible para fazer a instalação obedecendo condições mais complexas. O repositório tem mais informações sobre este assunto. Primeiramente devemos seguir os passos da Seção 3.5.6 para o LDAP, e depois da Seção 3.6.2 para o NFS.

3.8.1. Configurando opções do /scratch

Uma das configurações do PCAD é ter um diretório local onde diretórios reservados para cada usuário são criados. Os usuários recebem a recomendação de usar estes diretórios locais para seus experimentos, evitando o uso do NFS por questões de desempenho. Este é um passo opcional. Para criação e configuração do `</scratch>`:

[9]<https://slurm.schedmd.com/pam_slurm_adopt.html>
[10]<https://slurm.schedmd.com/priority_multifactor.html>

Comando no terminal do Cliente (usuário normal)

```
sudo mkdir -p /scratch
sudo chmod 755 /scratch
sudo mkdir /scratch/$ADMIN_USER
sudo chown $ADMIN_USER:$ADMIN_USER /scratch/$ADMIN_USER
sudo chmod 700 /scratch/$ADMIN_USER
echo 'SCRATCH=/scratch/$(whoami)/' | sudo tee -a /etc/profile
echo 'export SCRATCH' | sudo tee -a /etc/profile
```

Criar o script </usr/bin/scratch.sh> que será executado pelo PAM toda vez que o usuário entrar no nó computacional. Este script criará a pasta do usuário no </scratch> se ela não existir. Uma versão se encontra no repositório em <scripts/scratch.sh>. Em seguida, configurar as permissões e a chamada no PAM, modificando o arquivo </etc/pam.d/common-session>:

Comando no terminal do Cliente (usuário normal)

```
sudo cp  $SCRIPTPATH/scratch.sh /usr/bin/scratch.sh
sudo chmod 755 /usr/bin/scratch.sh
echo "session required    pam_exec.so /usr/bin/scratch.sh" | \
     sudo tee -a /etc/pam.d/common-session
```

3.8.2. Instalação do Slurm

Caso a máquina tenha aceleradores, como GPUs NVIDIA, seus drivers devem ser instalados antes do Slurm, para que este possa usá-los (exemplo CUDA) em sua compilação. Instalação de dependências e procedimentos de instalação do Slurm no cliente:

Comando no terminal do Cliente (usuário normal)

```
sudo apt install -y build-essential pkg-config libpam0g-dev \
    libmunge-dev munge libopenmpi-dev libnuma-dev libjson-c-dev \
    libyaml-dev hwloc libhwloc-dev libcurl4-openssl-dev libdbus-1-dev \
    liblz4-dev libhttp-parser-dev libreadline-dev
cd /scratch/$ADMIN_USER/
wget https://download.schedmd.com/slurm/slurm-23.02.0.tar.bz2
mkdir slurm
tar --bzip -x -f slurm*tar.bz2 -C ./slurm --strip-components=1
cd slurm
./configure
make -j
sudo make install
sudo cp /home/munge.key /etc/munge/munge.key
sudo chown munge:munge /etc/munge/munge.key
sudo chmod 600 /etc/munge/munge.key
sudo systemctl restart munge
pushd contribs/pam_slurm_adopt/
make -j
sudo make install
popd
echo "/usr/local/lib/" | sudo tee -a /etc/ld.so.conf.d/slurm.conf
sudo ldconfig
```

É necessário realizar os links de configuração do slurm com os arquivos do NFS. Ainda é necessário criar o arquivo `</usr/local/etc/gres.conf>` para configurar aceleradores. Caso a máquina não tenha aceleradores pode-se utilizar o exemplo do repositório em `<scripts/confs/gres.conf>`. Caso contrário, verificar a documentação[11].

Comando no terminal do Cliente (usuário normal)

```
sudo ln -s /home/slurm.conf /usr/local/etc/slurm.conf
sudo ln -s /home/cgroup.conf /usr/local/etc/cgroup.conf
sudo cp $SCRIPTPATH/confs/gres.conf /usr/local/etc/gres.conf
sudo chown slurm:root /usr/local/etc/gres.conf
sudo chmod 644 /usr/local/etc/gres.conf
```

Finalmente é necessário copiar o serviço do slurmd, criar os diretórios de spool, e alterar o serviço para esperar o LDAP e o NFS. Assim pode-se ligar o serviço do slurm:

Comando no terminal do Cliente (usuário normal)

```
sudo cp etc/slurmd.service /etc/systemd/system/
sudo mkdir /var/spool/slurmd
sudo chown slurm /var/spool/slurmd
sudo mkdir /etc/systemd/system/slurmd.service.d/
sudo cp $SCRIPTPATH/confs/slurmd_override.conf /etc/systemd/system/
   → slurmd.service.d/override.conf
sudo systemctl daemon-reload
sudo systemctl start slurmd
sudo systemctl enable slurmd
```

3.8.3. Configurações de autenticação para o Cliente

Algumas configurações são necessárias no arquivo `</etc/pam.d/sshd>` para configurar o PAM quando utilizado ssh nos clientes. As modificações se resumem ao patch disponível no repositório em `<scripts/confs/diff_pam_sshd_perm>`. Primeiramente ele adiciona o módulo do slurm para controlar acessos somente com quem tem a máquina reservada, mas adiciona uma exceção para os usuários configurados no `</etc/security/access.conf>`. Depois, ele desabilita o motd para usuários comuns, e a checagem de mail para todos. Para aplicar o patch e adicionar a exceção:

Comando no terminal do Cliente (usuário normal)

```
sudo patch /etc/pam.d/sshd $SCRIPTPATH/confs/diff_pam_sshd_perm
sudo sed -i "s/ADMIN_USER/${ADMIN_USER}/g" /etc/pam.d/sshd
echo -e "+ : $ADMIN_USER : ALL\n- : ALL : ALL" | sudo tee -a /etc/
   → security/access.conf
sudo systemctl restart sshd
```

3.8.4. Aplicação de teste

Para o usuário utilizar a infraestrutura é necessário associá-lo ao um grupo. Pode-se criar um grupo geral com o nome da infraestrutura e adicionar o usuário padrão. O Comando `add user` deverá ser realiza para todos os novos usuários.

[11]<https://slurm.schedmd.com/gres.conf.html>

Comando no terminal do controlador (usuário normal)

```
sudo sacctmgr -i create account $NOME_INFRA
sudo sacctmgr -i add user $ADMIN_USER account=$NOME_INFRA
```

Finalmente, pode-se submeter a primeira tarefa de teste:

Arquivo `job_teste.sbatch`

```
#!/bin/bash
#SBATCH -p shared
#SBATCH -o job_%j.out

echo "Estou executando em:"$HOSTNAME
sleep 60
```

Submeter o job e acompanhar a fila:

Comando no terminal do controlador (usuário normal)

```
sbatch $SCRIPTPATH/job_teste.sbatch
squeue
```

3.9. Outras configurações opcionais

Outras configurações são interessantes mas que este minicurso apenas aponta como próximos passos. Por exemplo, recomendamos (1) o uso de sistemas de arquivos como BTRFS com compressão nativa para a `</home>`, (2) a utilização de softwares de monitoramento de recursos como o Ganglia ou o Netdata, (3) a configuração de sincronia de relógios das máquinas com o NTP e (4) a utilização do ansible para uniformizar as configurações e softwares entre os nós.

3.10. Conclusão

No final deste minicurso, esperamos que seus recursos computacionais estejam aptos para serem compartilhados entre seus usuários. Diversas outras configurações são possíveis e talvez mais adequadas para cada caso.

Referências

Nesi et al. 2019 Nesi, L. L., Serpa, M. S., Schnorr, L. M., and Navaux, P. O. A. (2019). Hpc resources management infraestruture description and 10-month statistics. In *Proceedings do XVIII Workshop de Processamento Paralelo e Distribuído (WSPPD)*. GPPD, Porto Alegre.

Nesi et al. 2020 Nesi, L. L., Serpa, M. S., Schnorr, L. M., and Navaux, P. O. A. (2020). Advances in gppd–pcad management with 12-months analysis and perspectives. In *Proceedings do XVIII Workshop de Processamento Paralelo e Distribuído (WSPPD)*. GPPD, Porto Alegre.