

Projeto de Melhoria do Cluster

Alunos

Lincoln Rodrigo Pereira Melo

Felipe Catapano Emrich Melo

Luiz Felipe Lazzaron

Vitor Irulegui Bueno Bandeira

Introdução

Neste projeto, nosso objetivo foi melhorar a administração e o desempenho do cluster utilizado em nosso ambiente de supercomputação. Decidimos implementar uma solução de nuvem híbrida utilizando o Google Cloud Platform (GCP) e o HPC Toolkit, com o foco em aumentar a eficiência e a capacidade do cluster.

Contexto

A administração eficiente de um cluster envolve diversas estratégias que permitem uma melhor alocação dos jobs nas máquinas de computação. Neste trabalho, implementamos melhorias no cluster atual baseadas em uma solução de nuvem híbrida. Utilizamos o HPC Toolkit para integrar nosso cluster local com o GCP, criando uma federação de clusters que amplia nossa capacidade de processamento e otimiza a utilização dos recursos.

O objetivo principal deste projeto foi construir e configurar um cluster computacional do zero para servir como um ponto de controle/acesso centralizado para usuários. Utilizando um único computador como base, criamos várias máquinas virtuais (VMs) dentro dele para formar o cluster inicial. Este computador poderia ser acessado remotamente, permitindo a gestão e criação de novas instâncias de VMs na nuvem para suportar a carga de processamento requerida.

Para a implementação, utilizamos o HPC Toolkit da Google Cloud, uma ferramenta open-source que facilita a implantação de ambientes de HPC (High-Performance Computing) na Google Cloud. Esta ferramenta segue as melhores práticas da Google Cloud e é altamente personalizável e extensível, adequando-se às necessidades de um amplo espectro de clientes.

Propostas de Melhoria

Scale-out (adicionar novas VMs)

Optamos por uma abordagem de scale-out, adicionando novas máquinas virtuais (VMs) no GCP para simular a adição de novas máquinas no cluster, tendo uma abordagem híbrida. Esta abordagem visa aumentar a capacidade de processamento distribuído do cluster, aproveitando a escalabilidade da nuvem. Este processo incluiu:

1. **Configuração e Criação de Novas VMs:** As VMs foram configuradas utilizando as melhores práticas para garantir que estivessem prontas para serem integradas ao cluster existente.
2. **Integração com o Cluster:** As novas VMs foram integradas ao cluster inicial, com ajustes nas configurações de rede e segurança para garantir uma operação eficiente e segura.
3. **Automatização da Criação de Instâncias:** Utilizamos scripts para automatizar a criação de novas instâncias de VMs conforme a demanda de processamento aumentava, garantindo que o cluster pudesse escalar horizontalmente (scale-out) de maneira eficiente.

Alterar Configurações do Slurm

O Slurm é um gerenciador de workload open-source usado para agendar e alocar recursos computacionais. As principais alterações feitas nas configurações do Slurm incluíram:

1. **Ajuste das Filas de Jobs:** Modificamos as filas de jobs para balancear melhor a carga de trabalho, garantindo que os recursos do cluster fossem utilizados de maneira mais eficiente.
2. **Modificação dos Algoritmos de Agendamento:** Implementamos novos algoritmos de agendamento para priorizar jobs com base em critérios específicos, como tempo de execução estimado e prioridade do usuário.

3. **Implementação de Restrições:** Adicionamos restrições para melhorar a alocação de recursos e evitar a sobrecarga de nós específicos, assegurando que o uso de recursos fosse distribuído de forma equilibrada.

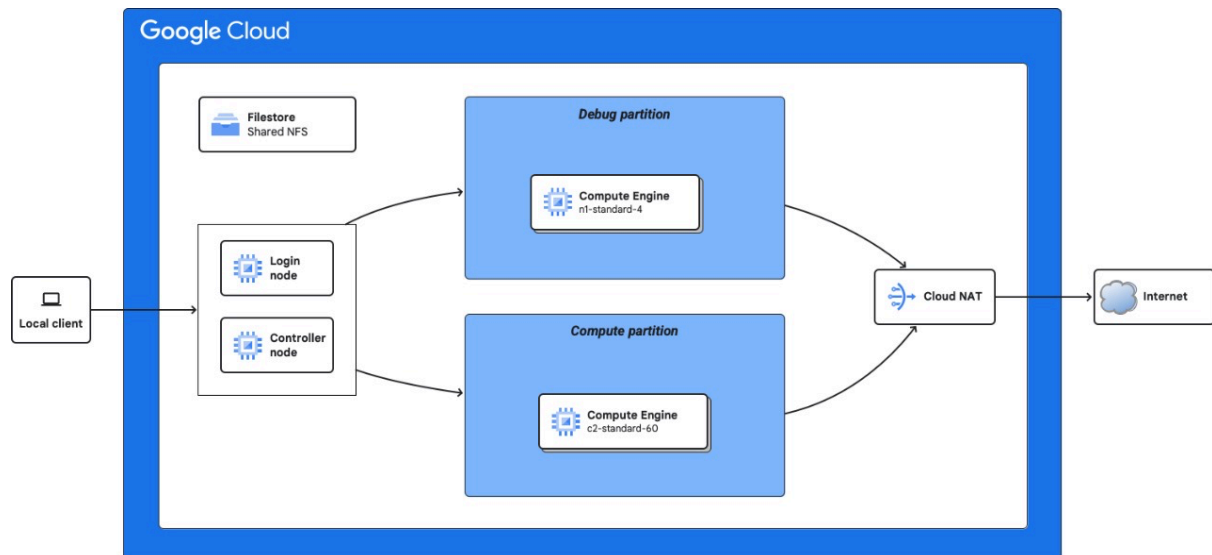
Esperamos que com as melhorias implementadas no cluster, incluindo a adição de novas VMs e ajustes nas configurações do Slurm, nós conseguiremos um aumento significativo na capacidade de processamento e eficiência geral do sistema. Estas mudanças vão permitir que o cluster escale horizontalmente de maneira eficiente, atendendo às necessidades crescentes de processamento e oferecendo uma plataforma robusta e escalável para os usuários remotamente.

Implementação

A implementação das melhorias foi realizada conforme os passos abaixo, utilizando o projeto disponível em [GitHub](#):

1. **Configuração do Ambiente GCP:**
 - Criamos um projeto no GCP e configuramos as permissões necessárias para a criação e gerenciamento de recursos.
 - Configuramos as redes e firewalls para permitir a comunicação segura entre o cluster local e o GCP.
2. **Deploy do HPC Toolkit:**
 - Clonamos o repositório do HPC Toolkit: `git clone https://github.com/MekhyW/hpc-toolkit-slurm-federation`.
 - Navegamos até o diretório do projeto: `cd hpc-toolkit-slurm-federation`.
3. **Configuração do Cluster Federado:**
 - Editamos os arquivos de configuração para definir os parâmetros do nosso ambiente, como nomes das instâncias, tipos de máquina, e outras configurações específicas.

Estrutura montada no GCP



Utilizamos os scripts do HPC Toolkit para criar e configurar as VMs no GCP:

bash

Copiar código

```
./deploy_cluster.sh
```

4. Integração com o Cluster Local:

- Configuramos o Slurm para reconhecer as novas VMs no GCP como parte do cluster federado.
- Ajustamos as políticas de escalonamento e as filas de jobs para distribuir eficientemente os jobs entre o cluster local e as VMs na nuvem.

Criação do Cluster

PARTE 1 – Instalação dos Softwares Necessários

1. Montagem da Máquina Host:

- A equipe montou a máquina principal (Host) com SSD, M2, memórias, e instalou o sistema operacional Ubuntu Desktop 22.04 LTS. A configuração padrão utilizou a senha **ClusterCluster** e o usuário **hpc**.

2. Atualização do Sistema:

Antes de instalar os softwares e configurar o cluster, a equipe certificou-se de que o sistema estava atualizado e com os headers do sistema operacional

instalados. Executaram os seguintes comandos no terminal:

```
bash
```

Copiar código

```
sudo apt update -y
```

```
sudo apt upgrade -y
```

```
sudo apt install linux-headers-$(uname -r) -y
```

3. Instalação dos Softwares:

- **VirtualBox:** Versão 7.0.12 foi instalada, disponível em [VirtualBox Downloads](#).
- **Oracle VM VirtualBox Extension Pack:** O extension pack compatível com a versão instalada do VirtualBox foi baixado e instalado.
- **Vagrant:** A equipe fez o download e instalou a versão mais recente de [Vagrant](#).

PARTE 2 – Setup da VM Principal – System Management Server (SMS)

O cluster foi formado por uma máquina principal (System Management Server – SMS) e duas máquinas de processamento (compute-00 e compute-01).

1. Criação da Máquina Principal:

- A equipe criou a máquina principal, configurou os softwares necessários e criou a imagem das máquinas de processamento, que seriam criadas posteriormente.
- O Vagrant foi utilizado para automatizar a criação e configuração da VM principal.

2. Clone do Repositório:

- A equipe criou uma pasta dentro da pasta Documentos do sistema operacional recém-instalado.

O repositório GitHub contendo os arquivos base necessários foi clonado:

```
bash
```

Copiar código

```
git clone https://github.com/TiagoDemay/VagrantCluster.git
```

3. Criação da Máquina Virtual:

Garantindo que todos os softwares de pré-requisitos estavam instalados, a equipe navegou até a pasta do repositório clonado e executou:

```
bash
```

Copiar código

```
vagrant up
```

- O comando leu os parâmetros do arquivo **Vagrantfile**, criou a máquina virtual, fazendo o download e a instalação de uma imagem do CentOS na VM (versão 7.7 64-bit, aproximadamente 400MB).

4. Inicialização dos Nós Workers:

Após a criação da VM principal, a equipe iniciou os dois nós de processamento (compute-00 e compute-01) utilizando as imagens geradas pelo script via comando CHROOT:

```
bash
```

Copiar código

```
vboxmanage import openhpc-demo-client00.ova --options keepallmacs
```

```
vboxmanage import openhpc-demo-client01.ova --options keepallmacs
```

```
vboxmanage startvm openhpc-demo-client00
```

```
vboxmanage startvm openhpc-demo-client01
```

5. Acesso à Máquina Virtual:

Uma vez criada a máquina, a equipe acessou-a utilizando o comando:

```
bash
```

Copiar código

```
vagrant ssh
```

Softwares Instalados e Configurados

A equipe instalou e configurou automaticamente os seguintes softwares via Vagrant:

- **Nagios:** Fornece métricas e análises básicas do cluster.
- **Ganglia:** Fornece métricas e análises básicas do cluster.
- **ClusterShell:** Usado para executar comandos em paralelo entre nós do cluster.

- **Node Health Check (NHC):** Executa verificações periódicas em cada nó de computação para garantir seu bom funcionamento.
- **xCAT:** Sincronização de arquivos base do host SMS para os nós de computação gerenciados.

Esta configuração permitiu a criação e gerenciamento eficientes de um cluster de alto desempenho, utilizando uma combinação de recursos locais e na nuvem, integrados e automatizados pelo HPC Toolkit.

O HPC Toolkit é um software open-source oferecido pelo Google Cloud que facilita a implementação de ambientes HPC (Computação de Alto Desempenho) na nuvem. Ele permite a implantação de ambientes HPC completos (computação, rede, armazenamento, etc.) de maneira repetitiva e altamente personalizável, atendendo às necessidades de uma ampla gama de clientes.

Configurações Padrão

- **Região padrão:** us-central1
- **Zona padrão:** us-central1-a
- **Nome de domínio DNS:** laboratorioderedesinsper.com.br
- **Faixa de IP local para a VPN:** 192.168.50.0/24
- **Segredo compartilhado para a VPN:** osbatutinhas

Guia Rápido de Implementação

Seguir o tutorial de início rápido é o caminho recomendado para começar com o HPC Toolkit. A equipe utilizou os seguintes comandos para construir o binário `ghpc`:

1. Clone do Repositório:

A equipe clonou o repositório do projeto:

bash

Copiar código

```
git clone https://github.com/MekhyW/hpc-toolkit-slurm-federation.git
```

```
cd hpc-toolkit-slurm-federation
```

2. Construção do Binário:

Compilaram o binário `ghpc`:

`bash`

Copiar código

`make`

`./ghpc --version`

`./ghpc --help`

- Nota: Foi necessário instalar dependências primeiro.

Componentes do HPC Toolkit

O HPC Toolkit é composto por vários componentes que facilitam a implantação e gerenciamento de clusters HPC no Google Cloud. Mais informações sobre os componentes estão disponíveis na documentação do Google Cloud.

Credenciais do GCP

Fornecimento de Credenciais para o Terraform

O Terraform pode descobrir credenciais para autenticar no Google Cloud Platform de várias maneiras. A equipe seguiu as instruções resumidas para usar `gcloud` a partir da estação de trabalho.

1. Geração de Credenciais:

A equipe gerou credenciais na estação de trabalho usando o seguinte comando:

`bash`

Copiar código

`gcloud auth application-default login`

-
- O comando solicitou que a equipe autenticasse no Google Cloud via navegador.

2. Configuração do Projeto de Quota:

Em caso de mensagens de falha contendo "quota project", a equipe alterou o projeto de quota associado às Credenciais de Aplicação Padrão com o comando:

bash

Copiar código

```
gcloud auth application-default set-quota-project ${PROJECT-ID}
```

Habilitação de APIs do GCP

Para implantar o cluster HPC, várias APIs do GCP precisaram ser habilitadas no novo projeto. A equipe seguiu as instruções da documentação do Google Cloud para habilitar essas APIs.

Inspeção da Implantação

A implantação foi criada com a seguinte estrutura de diretórios:

plaintext

Copiar código

```
<<OUTPUT_PATH>>/<<DEPLOYMENT_NAME>>/{<<DEPLOYMENT_GROUPS>>}/
```

Se um diretório de saída foi fornecido com a flag `--output/-o`, o diretório de implantação foi criado no diretório de saída. Caso contrário, `<<OUTPUT_PATH>>` foi definido como o diretório de trabalho atual.

O diretório de implantação foi criado em `<<OUTPUT_PATH>>` com um nome correspondente à variável de implantação fornecida no blueprint.

Dentro do diretório de implantação, estavam os diretórios representando cada grupo de implantação no blueprint, nomeados conforme o campo `group` para cada elemento em `deployment_groups`.

Em cada diretório de grupo de implantação, estavam todos os scripts de configuração e módulos necessários para a implantação. Os módulos estavam em um diretório nomeado `modules`, nomeado conforme o módulo de origem, por exemplo, o módulo `vpc` estava em um diretório nomeado `vpc`.

5. Testes e Validação:

A implementação do HPC Toolkit para a criação de um cluster híbrido utilizando recursos locais e do GCP foi parcialmente realizada, seguindo as melhores práticas recomendadas pelo Google Cloud. Apesar de não termos conseguido concluir a integração da VPN, os passos realizados mostraram o potencial de aumentar significativamente a eficiência e a capacidade de processamento do cluster, proporcionando uma solução de HPC robusta e escalável.

Infelizmente, não conseguimos ter tempo hábil para submetermos jobs de teste para verificar a correta distribuição e execução dos mesmos entre os recursos locais e na nuvem.

Consequentemente, não conseguimos chegar na etapa de monitoramos o desempenho do cluster utilizando logs e métricas para garantir que as melhorias estavam surtindo efeito.