Final Project Documentation

Deploying and Configuring an HPC Cluster on Google Cloud Platform

Abdul Muqeet Agha

Cloud Bursting

1. Introduction

This document outlines the steps and configurations for deploying a High-Performance Computing (HPC) cluster using Slurm on Google Cloud Platform (GCP). The project aimed to integrate NYU's Greene HPC cluster with a GCP Slurm setup, implement load balancing, and set up an Apptainer for containerized MPI workloads.

2. Project Objectives

1. Deploy a scalable HPC cluster on GCP using Slurm.
2. Ensure secure file transfer mechanisms between NYU Greene and GCP.
3. Implement a load balancer for traffic management.
4. Set up Apptainer, Singularity, and MPI for containerized computational tasks.
5. Optimize performance by adjusting machine types and CPU configurations.

3. Initial Setup and Cluster Deployment

- **Google Cloud Setup**
    - **Prerequisites**: Enabled necessary APIs: Compute Engine, Filestore, Cloud Storage, Service Usage, Secret Manager, and Resource Manager.
    - **Cluster Toolkit Setup**:
        - Cloned the cluster-toolkit repository.
        - Built the gcluster binary using make.
- **Cluster Deployment Steps**

Created a cluster blueprint using hpc-slurm.yaml.

```
Used gcluster to create a deployment folder and deployed the cluster:
 ./gcluster create examples/hpc-slurm.yaml -l ERROR --vars
project_id=<PROJECT_ID>
./gcluster deploy hpc-slurm
```

- Verified deployment by logging into the Slurm login node and testing Slurm commands:

```
srun -N 3 hostname
```

- Configured Compute Engine VM machine types for optimal performance (e.g., increasing CPUs to 4 per instance).

Cluster Configuration Highlights

- Auto-scaling compute nodes configured to scale up when jobs are submitted and terminate when idle.
- Persistent disk storage ensured data integrity during node scaling and instance type changes.
- Verified job scheduling and execution using Slurm commands.

4. Load Balancer Setup

- **Steps on GCP**:
    - Created a **Network Endpoint Group (NEG)** for the cluster.
    - Configured a **health check** to monitor cluster node availability.
    - Set up a **load balancer** to distribute incoming traffic among cluster nodes.

Modified firewall rules to allow all traffic:

```
gcloud compute firewall-rules update gcp-cluster-login-fw
--source-ranges=0.0.0.0/0
```

- **Verification**:
    - Tested the load balancer by SSH-ing into the login node and ensuring traffic distribution across available nodes.

5. Apptainer, Singularity, and MPI Setup

Apptainer Installation Sample

Logged into the Slurm login node and installed dependencies for Apptainer:

```
sudo yum install -y epel-release
sudo yum install -y gcc make libuuid-devel libseccomp-devel wget
squashfs-tools cryptsetup
```

Download and compile Apptainer Sample

```
wget
https://github.com/apptainer/apptainer/releases/download/v1.1.0/apptainer-1
.1.0.tar.gz
tar -xzf apptainer-1.1.0.tar.gz
cd apptainer-1.1.0
./mconfig
make -C ./builddir
sudo make -C ./builddir install
```

Singularity and MPI Configuration Sample

- Created an MPI-enabled container image using Apptainer.
- Ran parallel jobs on the Slurm cluster with MPI inside containers:

```
mpirun -np 4 singularity exec mpi-container.sif mpi_program
```

6. Optimizing Performance

CPU and Machine Type Adjustments

- Changed the login node's machine type from n2-standard-2 to n2-standard-4 for better performance.

```
gcloud compute instances set-machine-type <INSTANCE_NAME>
--machine-type=n2-standard-4
```

- Verified CPU configurations on the nodes using:

```
lscpu
```

Cluster Resource Tuning

- Adjusted auto-scaling parameters for compute nodes to dynamically allocate resources.
- Used Slurm configurations for job prioritization and resource allocation.

7. Presentation Highlights

- **Introduction**: Overview of the project's goals and scope.
- **Deploying the Cluster**: Step-by-step guide for setting up the cluster using Slurm and gcluster.
- **File Transfer with Globus**: Securely transferring files between Greene HPC and GCP.
- **Load Balancer Configuration**: Setting up and verifying the GCP load balancer for traffic management.
- **Apptainer and MPI Setup**: Enabling containerized workloads on the Slurm cluster.
- **Conclusion**: Summary of achievements, challenges faced, and future improvements.

8. Conclusion

This project demonstrated the deployment of a robust HPC cluster on Google Cloud, leveraging Slurm for job scheduling and integrating advanced configurations such as load balancing, containerization, and secure file transfer. The setup is scalable, secure, and capable of supporting diverse computational workloads. Future work can focus on integrating GPUs, improving job scheduling policies, and automating workflows for large-scale research tasks.