Mini-HPC and Hybrid HPC-Big Data Clusters

Abstract

This project aims to provide students with a comprehensive understanding of distributed computing systems through a hands-on approach. By building a mini-HPC (High-Performance Computing) cluster and integrating it with Big Data tools, learners will explore real-world computing challenges and technologies. This report outlines the steps taken, technologies used, and insights gained during the implementation of a hybrid computing environment using virtual machines, MPI, and Big Data platforms like Hadoop and Spark.

1. Introduction

High-Performance Computing is essential for solving complex computational problems quickly and efficiently. Traditionally, such systems require powerful hardware and specialized networks. However, with virtualization and open-source tools, it's possible to simulate such environments on standard personal computers. This project takes advantage of this concept by using VirtualBox and Ubuntu Server to create a cluster of virtual machines that mimic a real-world HPC system, followed by extending it with Big Data technologies to demonstrate hybrid computation.

2. System Requirements

To simulate a cluster environment, the following requirements were used:

Software Tools:
- VirtualBox: for virtualization and creating isolated VMs.
- Ubuntu Server 20.04 LTS: a lightweight and efficient Linux distribution.
- OpenSSH Server: for secure remote access and communication between nodes.
- OpenMPI: an open-source implementation of the Message Passing Interface protocol.
- Hadoop & Spark: to handle Big Data tasks in a distributed fashion.

Hardware Configuration (for each VM):
- 4 GB RAM
- 2 CPUs
- 20 GB Storage (dynamically allocated)

These resources provide a balance between performance and resource consumption, suitable for simulation on a personal computer.

3. Virtual Machine Setup

The first step involved installing VirtualBox and downloading the Ubuntu Server ISO image. Three virtual machines were created, each representing a node in the cluster. These were named 'master', 'worker1', and 'worker2'. Each VM was allocated sufficient memory and CPU to support basic clustering tasks. Ubuntu was installed using the ISO, and initial configurations were made to prepare the system for networking and MPI installation.

4. Network Configuration

To enable communication between the VMs, a host-only or internal network adapter was configured. This ensures that the VMs can communicate among themselves while being isolated from the external internet for security. Static IP addresses were manually assigned to each VM to make communication predictable. The '/etc/hosts' file on each VM was updated to map hostnames to IPs, allowing the use of readable names instead of numeric IPs.

5. SSH and MPI Setup

SSH was installed and configured to allow secure, passwordless login between the machines. This was achieved using SSH key pairs. MPI was then installed to enable parallel computation across nodes. MPI allows the master node to launch and manage processes on remote nodes as if they were local, thus simulating real parallel execution. A configuration file (hostfile) was created to specify participating nodes.

6. Testing the HPC Cluster

To verify the cluster, a simple MPI command was executed:

```
  mpirun -np 3 --hostfile hostfile hostname
```

This command runs a process on each specified node that simply returns the hostname. Successful output from all nodes confirms that SSH, MPI, and network configurations are functioning correctly. This test serves as a foundation for running more complex parallel tasks.

7. Big Data Integration

To create a hybrid computing environment, Big Data tools like Hadoop and Spark were installed. The 'master' VM was configured as the central controller (NameNode for Hadoop, Spark Master), and the two worker VMs acted as DataNodes or Spark Workers. Configuration involved setting environment variables, editing XML configuration files, and formatting the Hadoop filesystem. Sample programs such as word count (Hadoop) and Pi approximation (Spark) were run to verify distributed data processing functionality.

8. Results and Observations

The cluster successfully executed both MPI and Big Data tasks. Key observations include:
- Communication between nodes was seamless after proper SSH setup.
- MPI-based tasks scaled well with the number of nodes.
- Hadoop/Spark provided real-time insights into resource usage.
- Logs and web interfaces confirmed proper job distribution and execution.

9. Challenges

Several challenges were encountered:
- Initial network misconfigurations caused SSH failures.
- Ensuring all nodes had the same software versions and configurations was time-consuming.
- Debugging MPI errors without sufficient logs required deeper exploration of configurations.

These were addressed by careful step-by-step setup, verifying each component before proceeding, and documenting changes.

10. Conclusion

This project provided a rich, hands-on experience with distributed and hybrid computing. By starting with a minimal HPC setup and extending it with Big Data tools, students developed a clear understanding of parallel processing, system configuration, and data distribution. The skills gained are directly applicable in cloud computing, data science, and systems engineering roles.