BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Building distributed computation cluster

Vojtěch Bůbela

Abstract

The need to process big amount of data is more common today than ever before. One of the possible approaches to have your data processed is on a distributed computation cluster. The aim of this paper is prove that it is possible and relatively easy to build, configure and manage such cluster.
To achieve this I used slurm scheduler with number of other services such as MySQL database or MUNGE authentication daemon. I built and configured this cluster multiple times on different systems. First time on a pair of virtual machines running on one computer to get familiar with the process. Then on a set of raspberry pi computers, this time with additional functions such as accounting in a form of database or constraint of resources on computation nodes. I managed to build functional distributed computation cluster that has accounting, multiple queues and monitors resources allocated to user jobs. The resulting configuration can be applied to a set of machines to create a computing cluster and in this way save time and money for other things.

*xbubel08@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Processing big data or running a resource-heavy program on personal laptop or computer is not realistic. Instead buying compute time on distributed computing cluster is more common these days. However this can cost a lot of money so would it not be better to build and manage your own distributed computing cluster?

The main aim of my thesis is to build, configure and manage distributed computing cluster. Additionally configuring this cluster to have finer control over resources allocated to user jobs is secondary objective of my thesis. A simple distributed computing cluster has multiple nodes communicating with each other, multiple queues for user jobs and accounting database to hold information about jobs.

From already existing solutions I would like to mention computing cluster at Technical University of Ostrava. It is a robust cluster with many compute nodes. IT4Inovation is the group behind the cluster. They have great documentation outlining specifics of their cluster and also a user manual .The cluster uses PBS task scheduler for submitting jobs which is one of the schedulers i considered for my own thesis.

My own solution uses Slurm task scheduler. This task scheduler was chosen after building computation cluster with both Slurm and PBS task schedulers. As a first experiment I created several virtual machines of ubuntu server and installed task schedulers on them. As a second experiment i created cluster from three raspberry pi 3B computers and installed Slurm task scheduler on them. In addition to first experiment I also installed multiple other services normally found on computation clusters.

My thesis shows how can simple distributed computation cluster be built and managed. Configuration files and manuals created as part of my thesis can be used as a base for creating computation cluster using Slurm task scheduler. Let's say you have several computers that can function as a distributed cluster when they are not used. With this thesis configuring them to function as distributed cluster should be fairly easy.

2. Details of building and configuration of distributed cluster

I first thing I did was to get familiar with distributed computation clusters and task schedulers. On recommendation of my supervisor i studied PBS and Slurm task schedulers. From reading their documentation [1] to building very simple clusters with them.

My first experiment was to use both of these schedulers on cluster made from virtual machines ubuntu server. This was done on my personal computer with limited resources so i only created two virtual machines. The VMs were connected with virtual network and i installed both PBS and Slurm on them. Installation was done by downloading the source code from their respective repositories (both schedulers have opensource variant). After configuring and installing them I installed other software needed by these schedulers, for example MUNGE authentication daemon used by Slurm. Special attention was kept to correctly set user privileges for files and directories owned by these schedulers. If these privileges were set badly the scheduler would not work. Both schedulers were configured for the most basic use. That included defining addresses of all nodes in the cluster in the configuration file and services that should be run on each node. Afterwards I run simple job using both schedulers to prove i configured them correctly.

This experiment was very helpful with learning about clustering and task schedulers. Based on this experiment I chose Slurm task scheduler to be used further in my thesis. The reason is that Slurm has better mechanism of including plugins and felt overall easier to use. I attempted to run Slurm daemons in containers using docker, mainly to try if i could easily should result of my work to my supervisor. However this experiment did not go well because this experiment was too different from what the goal of my thesis is.

2.1 Cluster with raspberry pi computers

After deciding to use Slurm task scheduler in the second experiment I obtained three raspberry pi 3B computers from my supervisor to use as a distributed cluster. I decided to use Ansible automation when working with these computers. First step was to flash ubuntu server for raspberry pi on SD cards of the rpi computers. Then i ran ansible playbook that installed Slurm task scheduler with apt package manager. I did not install Slurm from source code this time because I was already familiar with the process and installation with apt package manager was easier. Ansible playbook also included all the set up that i previously did in experiment one.

It should be noted that raspberry pi computers were connected with Ethernet switch and connected to Internet router which assigned them IP address.

After testing that the cluster works I went on to implement my second goal for this thesis, that is finer management of clusters resources. The specifics of this goal were: having a way how to prevent user tasks from getting more resources than they were given and not allowing users to connect to compute nodes where their tasks is currently not running. I used Slurms extensive documentation In this task.

Slurm is configured using only one configuration file that needs to be present on every node on the cluster in correct directory that has correct privileges. In this configuration file `slurm.conf` administrator of the cluster can specify what services should Slurm use and what plugins will be available. First thing i configured was `consres` plugin which allows for non-exclusive allocation of nodes resource (without this or similar plugin one task will be allocated at least all of one compute node resources). I then defined hardware resources such as RAM memory or CPu count for every compute node in the cluster. When user submitted jobs they could specify the amount of specific resource their task needed.

I then used another plugin `cgroups` to monitor and limit resources used by user tasks. `Cgroups` plugin has its own configuration file where administrator can specify what resources should be limited and how much. For example when tasks requests 200 MB of RAM memory the administrator can specify that if that task should ask for more RAM it will receive Out Of Memory signal and be killed. Or maybe it can ask for up to 1.5 times the resources it asked for. Setting up this plugin was quite difficult as Slurms' documentation was not very clear.

3. Conclusions

I built and configured distributed computation cluster using multiple different sets of computing machines. Output of this effort are configuration files and manual that describe how to replicate this feat. I am confident that I met both of goals this thesis had. If i had more time I would look at user privileges on the cluster more closely.

Acknowledgements

I would like to thank my supervisor Jiří Jaroš for his help.

References

[1] SchedMD. Slurm documentation. online, 2023.