

## Implementing and Evaluating Multi-Resource Scheduling Design in Slurm

Report Dated: 03/06/2019

1. Installed Docker and docker compose for running multiple containers on virtual machine of lightning server.
2. Wrote docker and docker-entry point.sh file for creating image which builds SLURM packages and generates RPM's .This is done on Centos 7 image.
3. Wrote docker scripts for creating base images which will be used by Slurm Control daemon and Compute node daemons. It has scripts which install munge, openssh and Mariadb (default).
4. Created slurs.conf file. I have done by considering a two compute nodes and a single controller.
5. Wrote scripts for configuration of Slurm controller which includes setting up rsa keys, password less workers, starting munge, copying keys to a shared folder and reading slurm.conf file.
6. Wrote scripts for configuration of Slurm compute nodes which includes starting munge using the key created from Slurm controller, waiting for slurm controller to generate rsa key so that the daemons can use the same one and running slurm.conf file.
7. Wrote docker-compose.yml to create multiple images of the base file (that is two for compute nodes and one for controller).
8. Successfully ran the network of controller and two nodes. Tested working of slurm commands and ensured the communication between the three nodes.
9. Installed CQ Sim and ran MOO code to understand working of optimization Algorithm.
10. Implemented Cross Over, Mutation and Selection in 'C' assuming that we have an input of Jobs from Slurm Scheduler.

Plan for next week:

1. Intergrating the Genetic Algorithm (cross over, mutation and selection) with slurm's first in first out scheduler (i.e in builtin.c file).
2. Have to figure out how to run Slurm in Simulator mode so as to just run this changed file builtin.c and analyse the output.