

# RM-Replay for Cluster Scheduling Project Report 2/8-2/13

Zhen Huang, Blake Ehrenbeck

After the discussion from last Thursday, our plan is to make sure RM-Replay works by running with a simple test script. Once we make the test script work, we will write some parallel jobs using *MPI* and *pthread* to test it. Finally we will figure out how to use the large log file dataset we will be given, provided by Professor Lan's PhD student, Yuping.

## Progress This Week

We are trying to run the RM-Replay with the trace file we generated from the test script. The more familiar we became with the code, we found it was less portable than we expected. So we spent a lot of time debugging and try to make it run successfully on Lightning. We fixed many bugs which we will detail further in the report, but we still have some errors left in the code to fix and we will keep working on that next week.

Also, in order to run the Docker Images file we built in the last two weeks, we needed to volume bind the *group* file and the *passwd* file, (something we didn't do before) like so:

```
docker run --rm -it --volume /home/zhuang38/data:/slurm/data --volume /home/zhuang38/data/jarvis_etc_passwd:/etc/passwd --volume /home/zhuang38/data/jarvis_etc_group:/etc/group rm_replay_3
```

(We volume bind another file not seen above, but will mention in the section below)

## Modification

1. Originally we gave the Replay username as: *replayuser* in the Dockerfile, but we found that it didn't create the slurm user that we needed to run the slurm database.

We fixed this by passing the user name *slurm* into the Dockerfile, and then rebuilt it.

2. Several *start\_\*.sh* files in the Slurm-Replay folder have *Daint* hard coded into them. *Daint* is the cluster used by the author. So we replaced all the references with our cluster *jarvis* instead. Though RM-Replay is running on Lightning, the trace-file was generated on Jarvis.
3. The bug that took most of our time to fix was an issue with an SQL socket:

```
[replayuser@c09e221d9759 slurm-replay]$ ^C
[replayuser@c09e221d9759 slurm-replay]$ ls ../data
jarvis.trace jarvis_etc_group jarvis_etc_passwd slurmcfg slurmdb_tbl_slurm-18.08.0.sql
[replayuser@c09e221d9759 slurm-replay]$ ./start_replay.sh -w ../data/jarvis.trace -r 1 -n SR1
Preset=1 - Preset jobs (priority=1, hostlist=0, reservation=1, node_controller=1)
current date: Wed Feb 13 00:55:18 CET 2019
Set time to 2019-02-05 01:23:16
cp: cannot stat '../data/slurmcfg/topology.conf': No such file or directory
Starting mysql... done.
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/run/mysqld/mysqld.sock' (2)
Starting slurmdb... done.
Populating database... sacctmgr: error: Invalid user for SlurmUser slurm, ignored
sacctmgr: fatal: Unable to process configuration file
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/run/mysqld/mysqld.sock' (2)
^C
[replayuser@c09e221d9759 slurm-replay]$
```

It turns out the MySQL configuration file, *my.cnf* wasn't having the path of the socket written to it when the Dockerfile was used to build the image. We alleviated the issue by writing a new *my.cnf* file and volume binding it to */etc/mysql/my.cnf* . This fixed the issue.

## Problems Remaining

We are now facing the issue with slurm *sacctmgr*:

```
Starting slurmdb... done.
Populating database... sacctmgr: Accounting storage SLURMDBD plugin loaded
sacctmgr: error: slurm_persist_conn_open_without_init: failed to open persistent connection to localhost:6819:
onnection refused
sacctmgr: error: slurmdbd: Sending PersistInit msg: Connection refused
sacctmgr: error: Problem talking to the database: Connection refused
done.
```

We have tried a couple things including changing the SQL configure file, but none of them prove to work. We will continue working on this bug. As soon as we run this test trace file successfully, we will move on to the parallel job.