

RM-Replay for Cluster Scheduling

Project Report 2/1-2/7

Zhen Huang, Blake Ehrenbeck

Our task for Week 2 was to reproduce last week's work on the assigned cluster machines "Jarvis" and "Lightning" at the IIT Lab, as well as generate a testing input dataset, which includes a trace file, *passwd* file, and *group* file for RM-Replay from Slurm on Jarvis.

Progress:

We first installed RM-Replay on Lightning just like we did last week on the cloud cluster. Then we wrote a job script and submitted it to Slurm with a reservation created by the System Admin of Lightning.

Below is the batch script we wrote:

```
#!/bin/bash
#
#SBATCH --job-name=test
#SBATCH --output=res.txt
#SBATCH --reservation=behrenbe_3
#
#SBATCH --ntasks=1
#SBATCH --time=10:00
#SBATCH --mem-per-cpu=100

srun hostname
test_job.sh (END)
```

test_job.sh

Below are the reservation specifications on Jarvis:

```
ReservationName=behrenbe_3 StartTime=2019-02-05T15:41:04 EndTime=2019-02-12T15:41:04
Duration=7-00:00:00 Nodes=jarvis12 NodeCnt=1 CoreCnt=8 Features=(null)
PartitionName=(null) Flags=SPEC_NODES TRES=cpu=8 Users=behrenbe,zhuang38
Accounts=(null) Licenses=(null) State=ACTIVE BurstBuffer=(null) watts=n/a
```

We then submitted the script job with *sbatch*.

It was necessary to run a test batch job with a reservation (compared to running one without a reservation) as the trace builder bundled with RM-Replay connects to the *slurm_acct_db*, and the trace builder select data from the *jarvis_resv_table* and *jarvis_job_table*.

We compiled the trace builder, *trace_builder_mysql.c* in the RM-Replay folder with:

```
gcc -o trace_builder_mysql trace_builder_mysql.c -I/usr/include/mariadb mariadb_config --cflags -libs`
```

And ran the new trace builder binary file with:

```
./trace_builder_mysql -s '2019-02-04 18:25:00' -e '2019-02-04 19:00:00' -d slurm_acct_db -h localhost -P 6819 -p [XXXX] -u slurm -c jarvis -f jarvis.trace
```

```
Usage: mysql_trace_builder [OPTIONS]
-s, --starttime time          Start selecting jobs from this time
                               format: "yyyy-MM-DD hh:mm:ss"
-e, --endtime time           Stop selecting jobs at this time
                               format: "yyyy-MM-DD hh:mm:ss"
-d, --dbname db_name         Name of the database
-h, --host db_hostname       Name of machine hosting MySQL DB
-p, --password password      Password to connect to the db
-P, --port port              Port number of the machine hosting MySQL DB
-u, --user dbuser            Name of user with which to establish a
                               connection to the DB
-c, --cluster cluster_name   Name of the cluster used by the Slurm database to extract data
-f, --file filename          Name of the output trace file being created
-x, --dependencies filename  Name of the file containing the dependencies
-w, --where                  Do not use the where statement for the SQL query to retrieve the data
-n, --noprest                Do not preset the jobs
-?, --help                  This help message
```

Options for trace_builder_mysql

We got a trace file as one part of the dataset for RM-Replay. Then we generated the other two inputs running a python file *create_group_passwd.py* :

```
python create_group_passwd.py -passwd submitter/jarvis.trace_passwd -group submitter/jarvis.trace_group -n Jarvis
```

Next Steps: Feed the example trace into RM-Replay in “Lightning” to insure the installation working correctly. We will then generate more realistic workload trace once we make sure the testing one working correctly.