RM-Replay for Cluster Scheduling Project Report 2/14-2/28

Zhen Huang, Blake Ehrenbeck

Overview

In the conclusion of our last report, we wrote that we were still experiencing issues getting RM-Replay to replay our job. We have since been able to fix all of these issues. Many of the issues stem from the lack of portability of RM-Replay, which contains many components seemingly written specifically for the Daint supercomputer at the Swiss National Supercomputing Centre. In addition to now being able to replay a simple, one node job, we can now also replay an MPI job running across three nodes.

Progress

Our first step was to fix the issue connecting to the Slurm database daemon. After several hours of debugging, we found the issue boiled down to a simple race condition.

```
eval "$SLURM_REPLAY_LIB slurmdbd $VERBOSE"
sleep 1
echo "done."
```

The first line above in *start_slurmdbd.sh* starts the Slurm database daemon, sleeps for 1 second, and then prints "done". We found out the race condition was the issue by modifying the command in the first line to run as: *slurmdbd -Dvvv* so we could easily see the output. It turns out that 1 second was not long enough for the daemon to start, so we edited the script to sleep for 15 seconds instead. In the same file, *Daint*, was hardcoded into the script so we simply replaced all instances of *Daint* with our cluster name, *jarvis.*

We again tried to feed the trace of our simple and multi-node jobs to RM-Replay. We were met with a different error, this time an issue detecting the nodes enumerated in the *slurm.conf* we supplied. We found that our configuration file was missing a line configuring the FrontEnd mode:

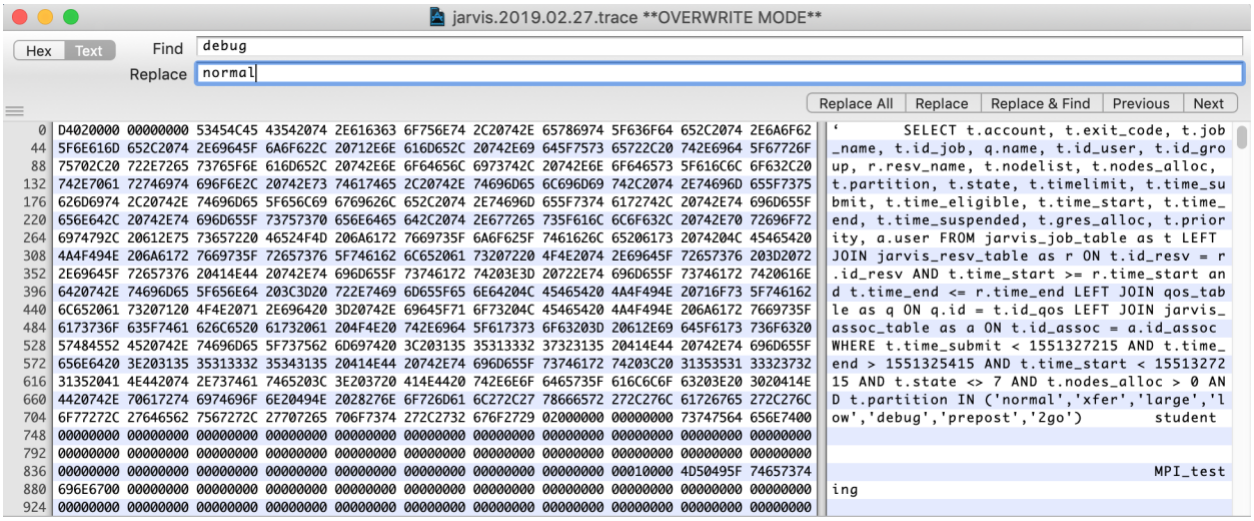
```
# COMPUTE NODES

FrontendName=localhost FrontendAddr=localhost Port=7000
```

This line didn't end up getting written to our Slurm configuration file because when RM-Replay ran a find and replace command on our file through *configure_slurm.sh*, it didn't find an exact match and therefore didn't add this line.

Again, we tried to feed our trace files. Another error: "Invalid partition name specified". After some digging we found for some reason RM_Replay only accepts reservations of partitions

named “normal” or “xfer”. Our partition is named debug. We decided to edit the trace binaries using a hex editor to replace our partition name of debug with normal.



This worked. But we had two errors left to fix:

1. We now couldn’t connect to the Slurm Controller
2. The clock’s start time was being incorrectly set in RM-Replay

1. The first fix again came to there not being an exact match for the find and replace mechanism in our Slurm configuration file. Some of the authentication methods and *cgroup* settings should have been set differently by *configure_slurm.sh* so we ended up writing them by hand into our configuration file.
2. RM-Replay was reading the start date incorrectly, so we hardcoded the start date our workload into the *start_replay.sh* script. To do this was located the start date of our workload and used the Linux *date* program to convert the date to a number readable by RM-Replay.

Fixing the above allowed us to be able to replay our jobs with RM-Replay. The multi-node job we ran was a gaussian elimination program written in C. We ran it across three nodes we this batch script:

```
#!/bin/bash

#

#SBATCH --job-name=MPI_testing

#SBATCH --output=res.txt

#SBATCH --reservation=behrenbe_4

#

#SBATCH --ntasks=3

#SBATCH --time=10:00

#SBATCH --nodes=3
```

After submitting the trace of this job to RM-Replay, it successfully generated the log files and some output to the console. Here is an example of what we saw in stdout:

```
Slurm is configured and ready:

PARTITION AVAIL  TIMELIMIT   NODES(A/I/O/T)  NODELIST
normal*      up    1:00:00        0/12/0/12  jarvis[12,14,22,31-32,41,44-45,81-82,84-85]

Start submitter and node controller... Submitter using no special option ...done.
Replay tentative ending time is Thu Feb 28 10:48:10 CET 2019
Clock: njobs=2 start='2019-02-27 06:55:00', end='2019-02-28 05:11:07', duration=80167[s], rate=0.00010[s] for 1 replayed second
Schedule not finished - current 2019-02-28 05:11:07 - hard end 2019-02-28 06:11:07 - njobs 1
Schedule not finished - current 2019-02-28 05:55:00 - hard end 2019-02-28 06:11:07 - njobs 2
Hard end time reached at 2019-02-28 06:11:07
1551330667 -- 2019-02-28 06:11:07  Schedule is over
*******************************************************
sdiag output at Thu Feb 28 06:11:07 2019 (1551330667)
Data since      Thu Feb 28 03:23:01 2019 (1551320581)
*******************************************************
Server thread count:  3
Agent queue size:     0
Agent count:          0
DBD Agent queue size: 0

Jobs submitted: 2
Jobs started:   2
Jobs completed: 2
Jobs canceled:  0
Jobs failed:    0

Job states ts:  Thu Feb 28 06:11:07 2019 (1551330667)
Jobs pending:   0
Jobs running:   0

Main schedule statistics (microseconds):
        Last cycle:   0
        Max cycle:    0
        Total cycles: 4
        Mean cycle:   0
        Mean depth cycle:  0
        Cycles per minute: 0
        Last queue length: 0

Backfilling stats
        Total backfilled jobs (since last slurm start): 2
        Total backfilled jobs (since last stats cycle start): 2
        Total backfilled heterogeneous job components: 0
        Total cycles: 2
        Last cycle when: Thu Feb 28 05:55:00 2019 (1551329700)
        Last cycle: 1000000
        Max cycle:  1000000
        Mean cycle: 500000
        Last depth cycle: 1
        Last depth cycle (try sched): 1
        Depth Mean: 1
        Depth Mean (try depth): 1
        Last queue length: 1
        Queue length mean: 1
```