

RM-Replay for Cluster Scheduling

Project Report 1/14-2/3

Zhen Huang, Blake Ehrenbeck

Our task for Week 1 was to get RM-Replay configured and installed on a machine. To become familiar with RM-Replay and how it works we read

RM-Replay: A High-Fidelity Tuning, Optimization and Exploration Tool for Resource Management authored by Maxime Martinasso et al. Once we understood the motivation behind the tool and the workflow/architecture we turned to Appendix A of the paper to see the installation guide.

Our Testbed

We setup a Linux machine running CentOS 7 that we could SSH into on Digital Ocean (<https://cloud.digitalocean.com>).

Installation

The installation was straightforward using the instructions from Appendix A of the paper above:

1. Clone RM-Replay from GitHub (<https://github.com/eth-cscs/slurm-replay>)
2. Install Docker. We followed the instructions exactly as written from:
<https://docs.docker.com/install/linux/docker-ce/centos/>
3. We then built the RM-Replay Docker image (we chose Slurm 18.08.0 as that is the version of Slurm on the cluster we are targeting):

```
docker build -t rm_replay_docker --build-arg SLURM_VERSION=18.08.0
```

4. At the time of this writing we do not have the dataset from the Swiss National SuperComputing Center (CSCS) so we did a trial run of the RM_Replay container without volume binding to the dataset:

```
docker run -it --rm rm_replay_docker
```

We are able to successfully run this command and launch the container. However, to ensure that our installation is working correctly we need to try and provide RM_Replay with a dataset. We have emailed the first author of the paper to ask where we might find this data. In the meantime, we will try to run jobs on Slurm and generate a Slurm database that we can then process and feed to RM_Replay.