Slurm Simulator Development: Balancing Speed, Accuracy, and Maintainability

Nikolay A. Simakov
Robert L. DeLeon
nikolays@buffalo.edu
rldeleon@buffalo.edu
Center for Computational Research, University at Buffalo, State University of New York
Buffalo, NY, USA

1 EXTENDED ABSTRACT

Slurm is an open-source job scheduling system widely used in many high-performance computing (HPC) resources. A Slurm simulator facilitates parameter tuning to optimize throughput or meet specific workload objectives. In the previous simulator version (v2) [2], the priorities were to minimize the changes to core Slurm and have a high simulation accuracy. This resulted in speed-dependent accuracy and a simulation speed only 20-40 times faster than real-time (for a midsized system). This is not a very practical simulation speed, and it is more beneficial to trade some accuracy for increased speed. The expectation was that with diminished absolute accuracy, we could still make reasonable relative predictions. To achieve the desired speed-up goal, we use the same strategy as in our original Slurm simulator (v1) [1, 3], namely: serialize the code and call all Slurm functions from a single thread in an event-driven fashion. Our simulator's resulting version (v3) has more than 500 times acceleration over real-time, allowing simulation of a month-long workload in 90 minutes.

The simulator was tested on a Mid-sized System containing 216 heterogeneous nodes containing a mixture of resources (two types of regular compute nodes, large memory nodes, and GPU nodes). The workload (also known as job traces) was based on the historical workload at our center and consisted of almost 30,000 jobs. It requires more than 29 actual days to be executed. The reference data was obtained using our Virtual Cluster, where each cluster node is represented with its own container and has a normal Slurm installed on it (see [2] for more details). To estimate the ability to predict relative values rather than absolute ones, we also vary the priority factor of several QoS groups (priority and supporters) while keeping the general QoS the same.

The mean wait times for Mid-sized System grouped by QoS is shown in Figure 1. The absolute value of mean wait time differs between Virtual Clusters and Slurm Simulator. However, the trend is very similar, especially on higher values of mean wait time. The scheduling in Slurm is a stochastic process [1, 3], which has a particularly high manifestation on highly utilized systems. Therefore, it is crucial to have a sufficient number of independent runs. It is easy to obtain multiple runs with the Slurm simulator as it is several hundred times faster than real-time; however, Virtual Cluster goes only as fast as real-time, and it takes a lot of time to get through 29 days of the test workload. So far, we have done four independent runs for each configuration, and we plan to double it at least to have more conclusive results.

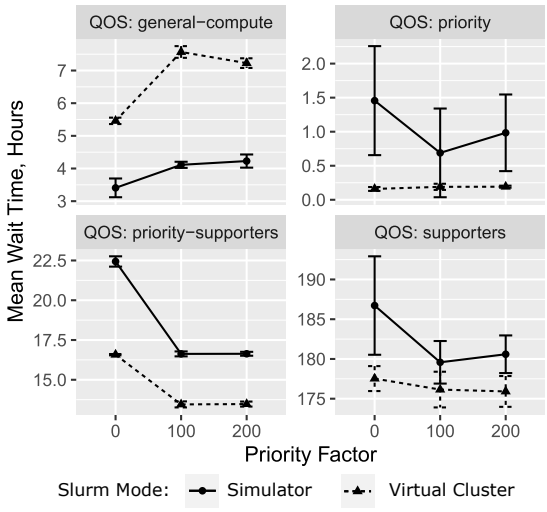

Figure 1: Change in mean wait time in different Slurm QoS upon increasing Priority Factor of several groups. The x-axis shows the value of the Priority Factor for priority and supporters QoS; priority-supporters double that number, and general-compute always has zero Priority. Y-axis shows the mean wait time (averaged over all jobs in workload). The symbol and error bar represent the mean and standard deviation from four independent runs.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under the OAC 2004954 award. This work used compute resources at UB CCR. This work also used Jetstream 2 at Indiana University through allocation CIS230285 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services Support (ACCESS) program, which is supported by National Science Foundation grants 2138259, 2138286, 2138307, 2137603, and 2138296.

REFERENCES

[1] Simakov, DeLeon, Innus, Jones, White, Gallo, Patra, and Furlani. 2018. Slurm Simulator: Improving Slurm Scheduler Performance on Large HPC Systems by Utilization of Multiple Controllers and Node Sharing. In *PEARC '18.* https://doi.org/10.1145/3219104.3219111

[2] Simakov, Deleon, Lin, Hoffmann, and Mathias. 2022. Developing Accurate Slurm Simulator. In *PEARC '22.* https://doi.org/10.1145/3491418.3535178

[3] Simakov, Innus, Jones, DeLeon, White, Gallo, Patra, and Furlani. 2017. A slurm simulator: Implementation and parametric analysis. In *PMBS17 at SC17.* 197–217. https://doi.org/10.1007/978-3-319-72971-8_10