Open-Source Contributions to Arbiter2

Dylan Grant Gardner

August 2020
Open-Source Contributions to Arbiter2

Dylan Grant Gardner

August 2020

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517
Open-Source Contributions to Arbiter2

Dylan Gardner – University of Utah  Mentor – Ben Nickell  C520 – High Performance Computing and Data Analytics

About

Login nodes at High-Performance Computing (HPC) sites are shared resources used by a multitude of users to compile, test and submit jobs to the batch system. Because these nodes are shared by multiple users at any time, if a minority of users are using a significant proportion of the resources of the node (CPU, memory, etc.), other users’ tasks may be negatively impacted. Arbiter2 is an open-source daemon developed by the University of Utah that aims to prevent these occurrences by dynamically limiting the resources of users depending on whether they are excessively using resources.

INL has sought to adapt and develop Arbiter2 for a potential deployment at INL and has upstreamed their changes and improvements so that other HPC sites running Arbiter2 can benefit from their work.

Contributions from INL

- Adapted Arbiter2 to work on non-CentOS 7 nodes (SLES 12)
- Optimized process collection performance
  - ~5% process memory collection speedup (heavy cost; massive improvement)
  - ~6% process metadata reading speedup (significant improvement)
  - ~2% less time overall opening files (significant improvement)
  - ~13% Total Speedup
- Notable bugfixes
  - Uncovered bug where kernel disk cached memory was counted against users (kernel caching should not be considered as user-owned memory)
  - Fixed a couple race condition bugs that can cause Arbiter2 to crash

Source Code: https://gitlab.chpc.utah.edu/arbiter2/arbiter2

How Arbiter2 Works

- Uses cgroups, a feature of the Linux kernel, to monitor and limit the CPU and memory resources of users
- Evaluates users’ cgroup usage at an interval and scores that usage based on a soft limit threshold
  - Increases the “badness score” when above soft quota threshold
  - Decreases when below (but at a slower rate)
  - Reaching the max badness score means the user is in violation of the acceptable usage policy
- Automatically emails users about their behavior and temporarily lowers resource limits upon an acceptable usage policy violation to make usage fairer for all