SOSflow: A Scalable Observation System for Introspection and In Situ Analytics

Efficiently observing and interacting with complex scientific workflows at scale presents unique challenges. SOSflow helps meet them.

**Design and API**

- SOSflow written in C99 for high-performance w/small footprint
- Several communication backends are supported, including EVPath, MPI, sockets, and ZeroMQ
- Asynchronous design focuses on minimizing overhead and time spent in API calls within client applications
- Flexible, programmable interface
- Provides a distributed key/value store with full SQL query support
- Offers a low-latency value cache with adjustable depth
- Highly-configurable daemons
- Integrated support for UID/GID authentication (Munge)

**Results: Performance Understanding**

- 4.0% ranks of XGC on TITAN
- Data collected and aggregated online from TAU measuring ADIOS, MPI, and user code
- Python script queried SOSflow during the run and assembled VTK files with performance metrics projected over server rack and node coordinates
- SOSflow integrated performance measurements from all parts of the workflow
- Dynamic visualizations were rendered and displayed live during the run
- Any TAU-collected performance metrics could be selected for display

- S12 ranks on 32 nodes on QUARTZ and CATALYST
- SOSflow filter added to ALPINE Ascent pipeline
- TRIPKE: 3D deterministic neutron transport proxy application that implements a distributed-memory parallel sweep solver over a rectilinear mesh
- LULESH: 3D Lagrangian shock hydrodynamics proxy application that models Sedov blast test problem over a curvilinear mesh
- 4 ad hoc instrumentation needed
- Updated geometry is automatically captured during the run to observe metrics projected over a changing mesh
- Anything published to SOSflow can be projected into these online views
- SOS runtime overhead within system noise
- Enable/disable without recompilation

**SOS Model**

SOSflow functions as a hub for collecting, aggregating, and acting on a variety of information at runtime

SOSflow’s in situ (online) services work together to provide global views and online data analytics within an HPC environment

**SOS Daemons**

SOSflow daemons provide an integrated context for information from all components of a distributed workflow, for the entire duration of a job.

- In Situ Listeners
  - Runs entirely in user space
  - Minimally invasive
  - In-memory SQL database per daemon
  - Efficient push/pull data flow
  - Provides feedback/control mechanism for analysis and steering scripts
  - Integration with performance tools
  - APIs for C/C++ and Python

- Off-Node Aggregators
  - Can run on dedicated nodes
  - SQL store contains all the data captured by the listeners
  - Can launch many aggregators and run queries on them in parallel
  - Send feedback/control data to listeners
  - Online aggregation can be disabled or deferred to offline processing

**Future Work**

Apollo Performance Portability
- Next Generation of LLNL’s Apollo Project
- Intelligent RAJA policy configuration
- Caliper and SOSflow collect metrics at runtime and facilitate distributed analysis and steering
- Online machine learning adapts to changes over time
  i. Physics changes over time in a run
  ii. Code changes w/new commits and merges
  iii. System utilization changes during jobs

**Author**

Chad Wood
chad.c.wood@oregon.edu

Chad Wood is a fourth-year Computer & Information Science PhD student at the University of Oregon. His research focuses on monitoring, introspection, feedback, and control for HPC systems, emphasizing in situ operations and scalability.

**Collaborators**

- Alfredo Gimenez
  - gimenez1@llnl.gov
- Matt Larsen
  - larsen30@llnl.gov
- David Poliakoff
  - poliakoff1@llnl.gov
- Kevin Huck
  - khuck@cs.uoregon.edu

The research report was supported by a grant (DE-SC0012381) from the Department of Energy, Scientific Data Management, Analytics, and Visualization (SDMAV), for “Performance Understanding and Analysis for Exascale Data Management Workflows." Part of this work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-LTM-751122).