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.

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.

**Design and API**

- SOSflow written in C99 for high-performance w/small footprint
- Several communication backends are supported, including EVPaths, 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 RegEx query-by-name
- Highly-configurable daemons
- Integrated support for UID/GID authentication (Mungs)

**“Hello, World” w/C:**

```
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

**Online query w/Python:**

```
import pySOSflow

query = "SELECT * FROM sosflow_data" 
results = pySOSflow.query(query)
```

**Results**

- 4,006 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
- XRIKE: 3D deterministic neutron transport proxy application that implements a distributed-memory parallel sweep solver over a rectilinear mesh
- LLNLH: 3D Lagrangian shock hydrodynamics proxy application that models Sedov blast test problem over a curvilinear mesh
- No 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

**Future Work**

- Apollo Performance Portability
  - Next Generation of LLNL’s Apollo Project
  - Intelligent RAIA execution policy choices
  - Use SOSflow to gather and analyze metrics at runtime
- Online machine learning adapts to changes over time
  - Physics changes over time in a run
  - Code changes w/new commits and merges
  - System utilization changes during jobs

**Author**

Chad Wood

chad@cs.uoregon.edu

**Colleagues**

- Laura Strasser
- John Snyder
- John Kime
- John Dunlap

**Collaborators**

- Lawrence Livermore National Laboratory
- University of Oregon

The research was supported by a grant (DE-SC0012661) from the Department of Energy, Scientific Data Management, Analytics, and Visualization (SDMAV), for “Performance Understanding and Analytics 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-CM-737074, LLNL-PROC-761948).