SAND2022-16200C

Sandia National Laboratories

Exceptional service in the national interest

SNL HPC Monitoring and Analysis with AppSysFusion

Ben Schwaller – HPC Development

Supercomputing 2022 BoF

Operational Data Analytics – Drowning in Data

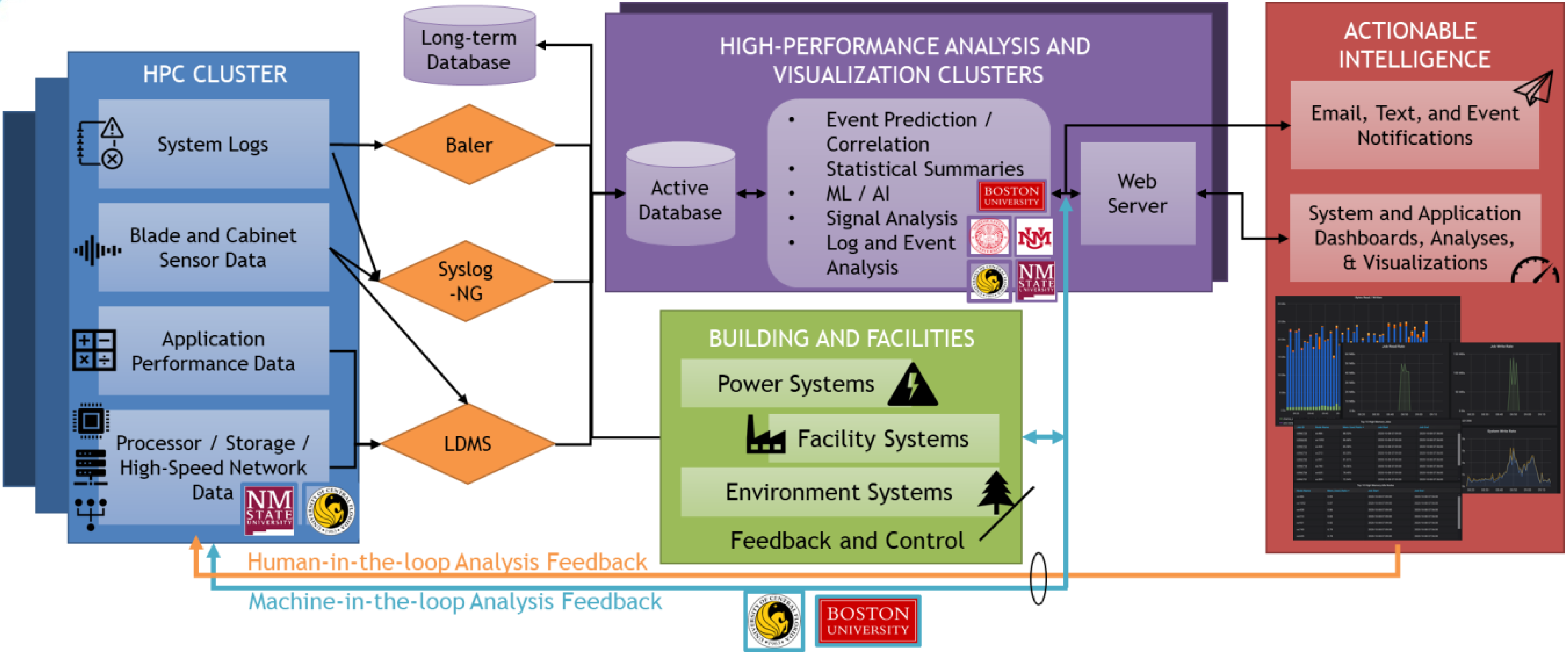
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. ation under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

U.S. DEPARTMENT OF ENERGY

NNSA

Holistic HPC Monitoring and Analysis Architectural Overview



HPC CLUSTER

- System Logs
- Blade and Cabinet Sensor Data
- Application Performance Data
- Processor / Storage / High-Speed Network Data

Long-term Database

Baler

Syslog -NG

LDMS

HIGH-PERFORMANCE ANALYSIS AND VISUALIZATION CLUSTERS

Active Database

- Event Prediction / Correlation
- Statistical Summaries
- ML / AI
- Signal Analysis
- Log and Event Analysis

Web Server

BUILDING AND FACILITIES

- Power Systems
- Facility Systems
- Environment Systems
- Feedback and Control

ACTIONABLE INTELLIGENCE

- Email, Text, and Event Notifications
- System and Application Dashboards, Analyses, & Visualizations

Human-in-the-loop Analysis Feedback

Machine-in-the-loop Analysis Feedback

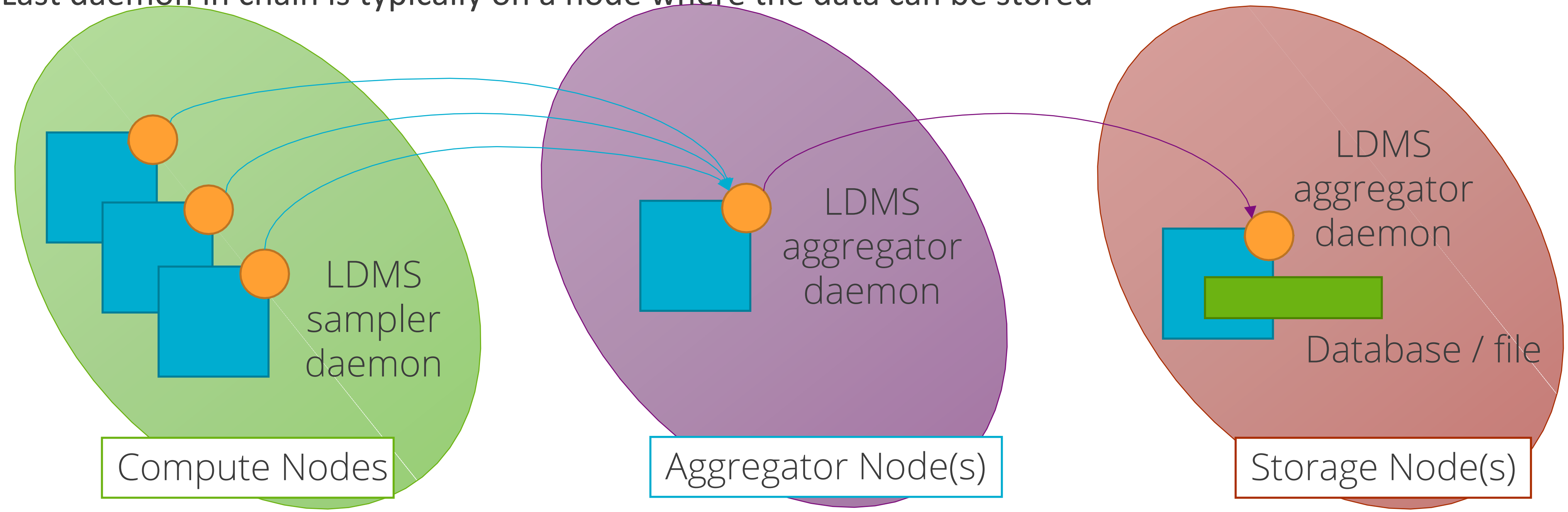
Lightweight Distributed Metric Service (LDMS)

At Sandia, we use LDMS to collect system and application data

- LDMS can collect 1000s of system metrics at sub-second intervals, typically collect at 1 Hz
- These metrics range from network performance counters to filesystem statistics to CPU and memory utilization
- Currently collecting ~10s of TB of data **each day** to custom database Distributed Scalable Object Store (DSOS)

An LDMS sampler daemon on each compute node collects information and sends it synchronously to an LDMS aggregation daemon, typically on an admin node

- Aggregator daemons can chain as many times as desired
- Last daemon in chain is typically on a node where the data can be stored

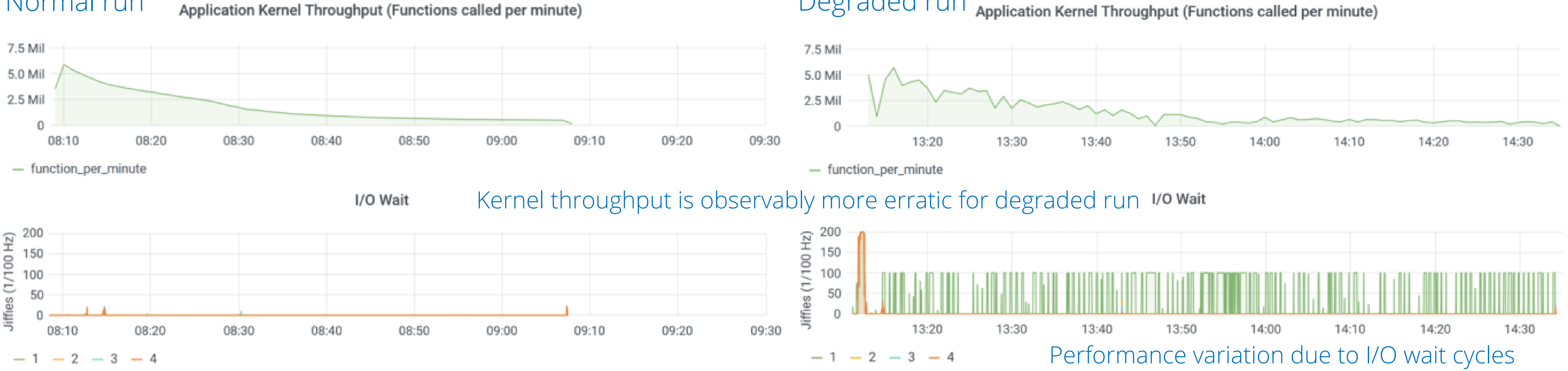


AppSysFusion: Integrating Application and System Data for Execution Time Diagnosis of Performance Variation

The **AppSysFusion** project combines kernel timing data and system metrics in an analysis and visualization framework to enable real-time insights

- ○ **Kokkos** used to sample application and timing data without recompilation
- ○ **LDMS** used to sample whole-system data and transport all collected data to separate cluster for storage and analysis

Deployed on several HPC systems at SNL and collaborating with multiple code teams



Normal run

Degraded run

Kernel throughput is observably more erratic for degraded run

Performance variation due to I/O wait cycles

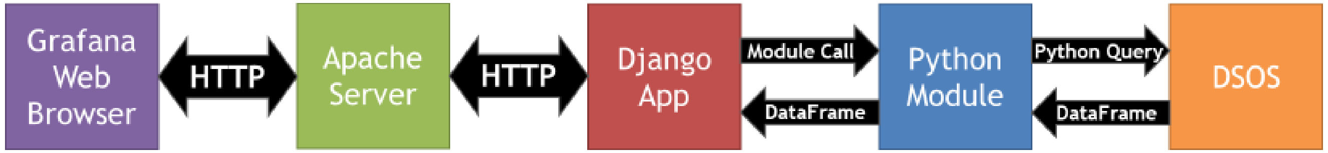
Dashboard of EMPIRE run with and without degraded performance caused by significant I/O contention

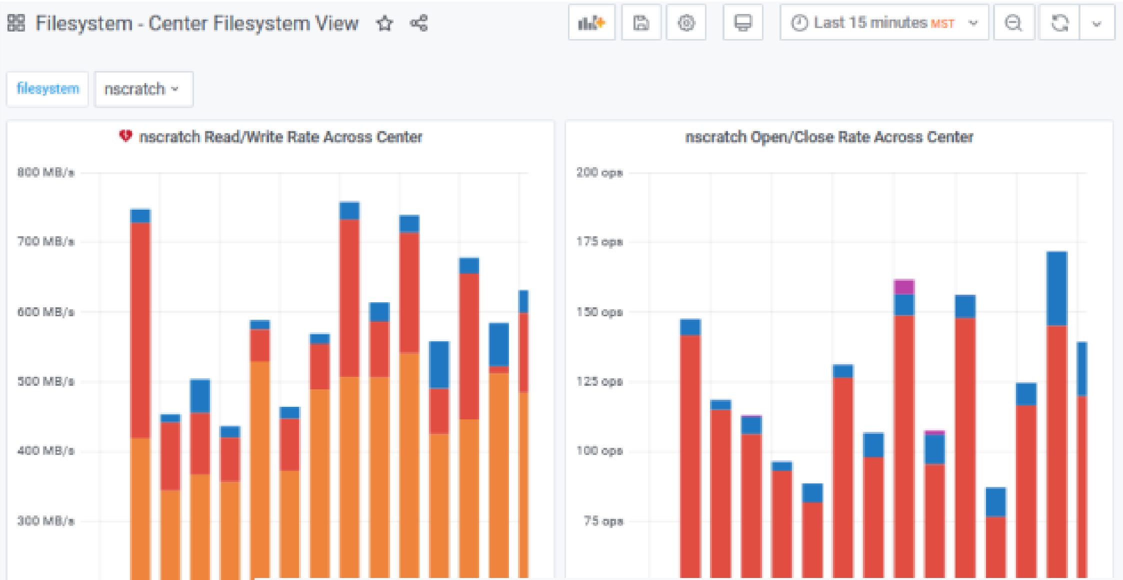
Runtime Analysis and Visualization of System and Applications

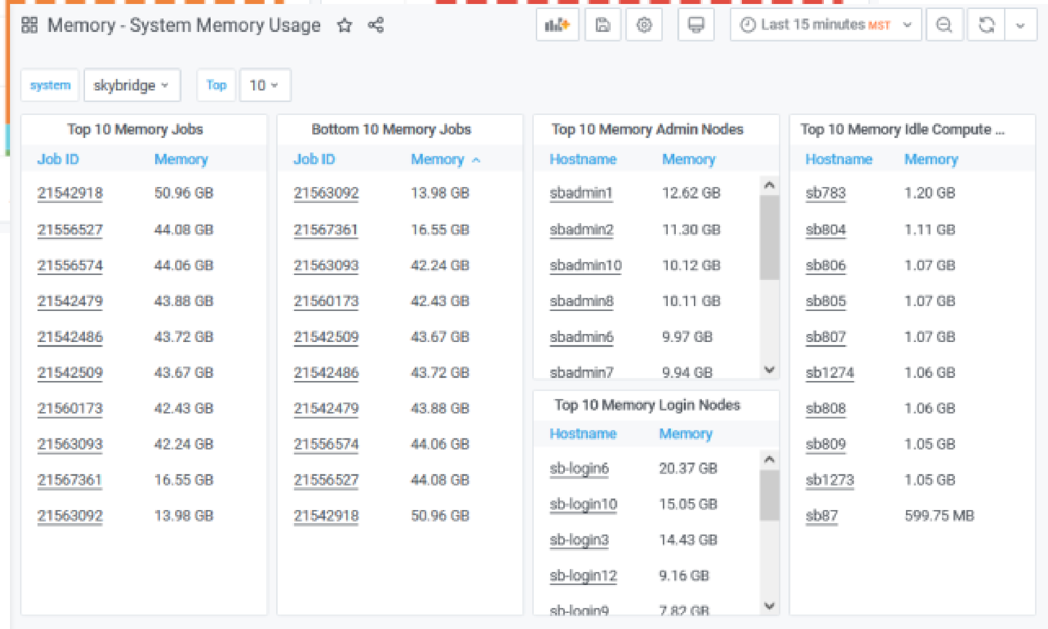
User queries from Grafana dashboards are sent through a backend python application which can call python analyses to derive metrics from raw data

- In-query analyses save significant computation time/resources for creating analysis results
  - Only data of interest is analyzed and new analyses can be created without recreation of analysis results across the database
  - Analyses can easily be changed / added to meet new challenges and decrease

Python modules can query the database and return pandas DataFrames for analysis







Top 10 Memory Jobs

Job ID	Memory
21542918	50.96 GB
21556527	44.08 GB
21556574	44.06 GB
21542479	43.88 GB
21542486	43.72 GB
21542509	43.67 GB
21560173	42.43 GB
21563093	42.24 GB
21567361	16.55 GB
21563092	13.98 GB

Bottom 10 Memory Jobs

Job ID	Memory ^
21563092	13.98 GB
21567361	16.55 GB
21563093	42.24 GB
21560173	42.43 GB
21542509	43.67 GB
21542486	43.72 GB
21542479	43.88 GB
21556574	44.06 GB
21556527	44.08 GB
21542918	50.96 GB

Top 10 Memory Admin Nodes

Hostname	Memory
sbadmin1	12.62 GB
sbadmin2	11.30 GB
sbadmin10	10.12 GB
sbadmin8	10.11 GB
sbadmin6	9.97 GB
sbadmin7	9.94 GB

Top 10 Memory Login Nodes

Hostname	Memory
sb-login6	20.37 GB
sb-login10	15.05 GB
sb-login3	14.43 GB
sb-login12	9.16 GB
sb-login9	7.82 GB

Top 10 Memory Idle Compute ...

Hostname	Memory
sb783	1.20 GB
sb804	1.11 GB
sb806	1.07 GB
sb805	1.07 GB
sb807	1.07 GB
sb1274	1.06 GB
sb808	1.06 GB
sb809	1.05 GB
sb1273	1.05 GB
sb87	599.75 MB

University Research Collaborations

SNL has partnered with six universities over the past five years to explore research topics for HPC monitoring and analysis

University collaborations allow us to explore and trial new techniques with top researchers in their field

Brief list of university topics:

○ Boston University	– Machine Learning Modelling of HPC Anomalies
○ New Mexico State University	– GPU Resource Monitoring and Application Instrumentation
○ University of Central Florida	– HPC Resource Allocation and Metric Collection Research
○ University of New Mexico	– Time-series Analysis using Dynamic Time Warping
○ University of Northeastern	– Network and DARSHAN I/O Characterization and Clustering
○ University of Urbana Champaign	– Network Contention / Modelling Analysis

SNL has been researching monitoring and analysis techniques for HPC centers since 2003

List of most publications can be found at https://github.com/ovis-hpc/ovis-publications/wiki

Questions?

Backup Slides

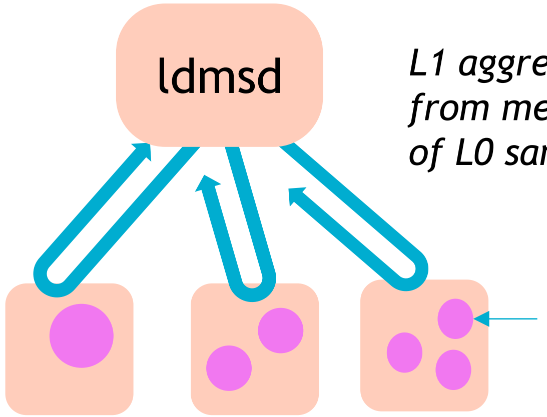
Enabling Application Data Injection via LDMS

LDMS - **low-overhead (<1% application)** data collection, transport, and storage capability designed for **continuous monitoring** supporting **run time analytics** and feedback.

- System data collection is typically **synchronous** at regular (e.g., second or less) intervals
- **Structured** data format (i.e., metric set) designed to minimize data movement
- Transport is typically **pull** based to minimize CPU interference
- Transport to multiple arbitrary consumers over both RDMA and socket

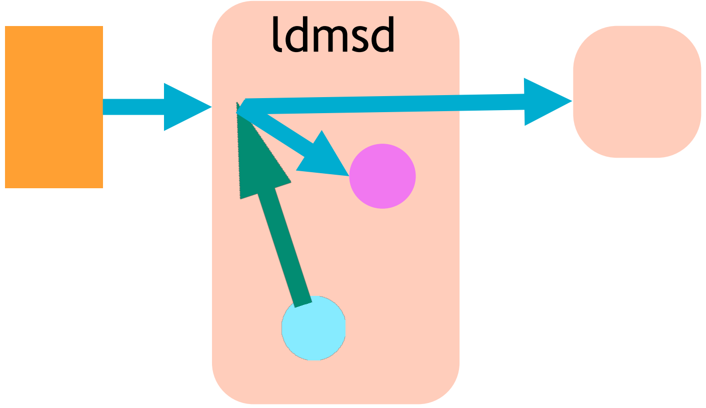
LDMS Streams – **on demand publication** of loosely formatted information to subscribers

- Transport is **push** based and supports **asynchronous** event data (e.g. scheduler and log data)
- **Unstructured** data



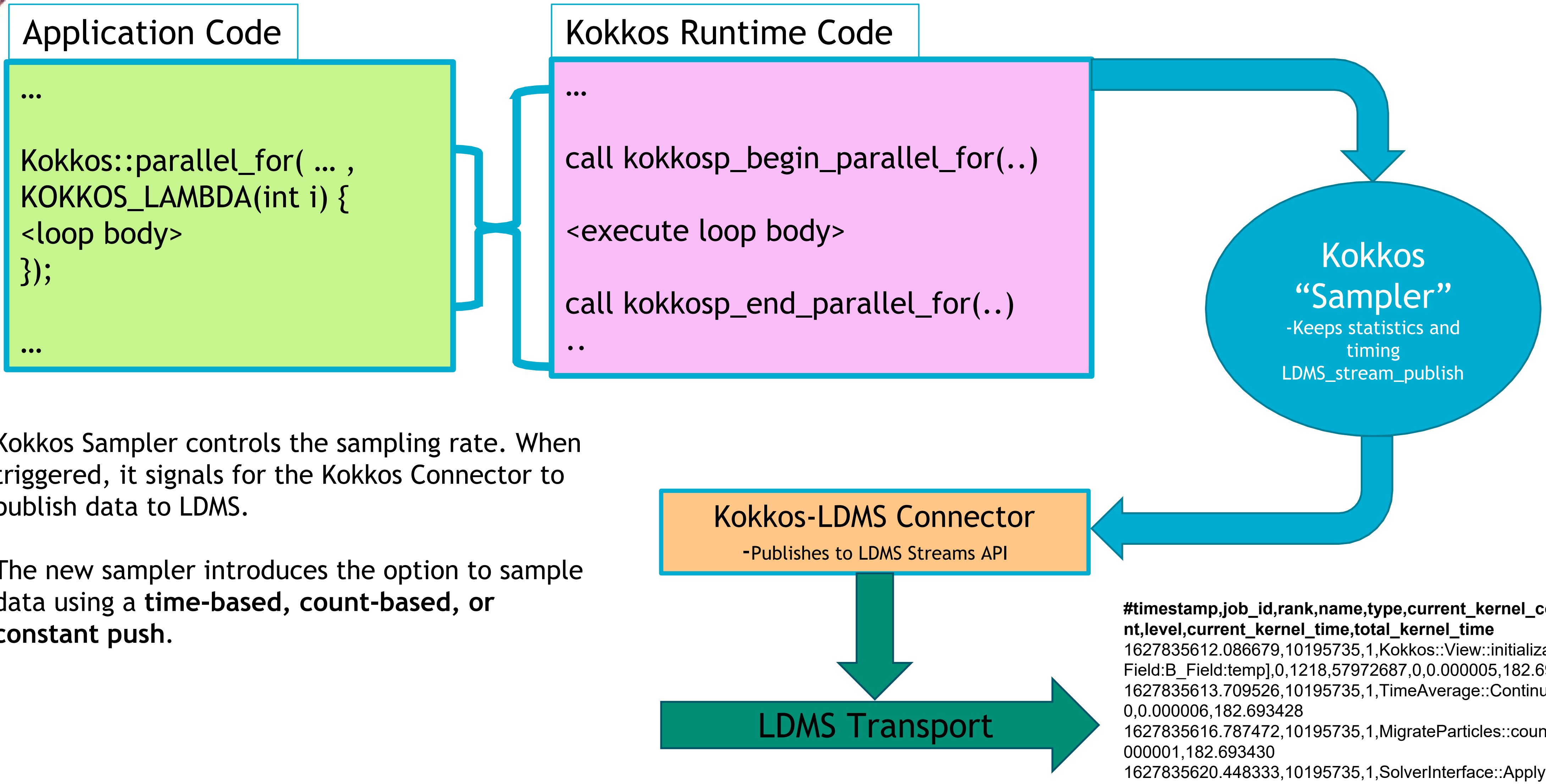
*L1 aggregator **pulls** from memory regions of L0 samplers*

Sampler plugins



*Daemon publish API called from externally or by a plugin **pushes** to ldmsd which pushes to all subscribing plugins and aggregators*

Kokkos to LDMS publish



Application Code

```
…

Kokkos::parallel_for( … ,
KOKKOS_LAMBDA(int i) {
<loop body>
});

…
```

Kokkos Runtime Code

```
…

call kokkosp_begin_parallel_for(..)

<execute loop body>

call kokkosp_end_parallel_for(..)
..
```

Kokkos "Sampler"
-Keeps statistics and timing
LDMS_stream_publish

Kokkos-LDMS Connector
-Publishes to LDMS Streams API

LDMS Transport

Kokkos Sampler controls the sampling rate. When triggered, it signals for the Kokkos Connector to publish data to LDMS.

The new sampler introduces the option to sample data using a **time-based, count-based, or constant push**.

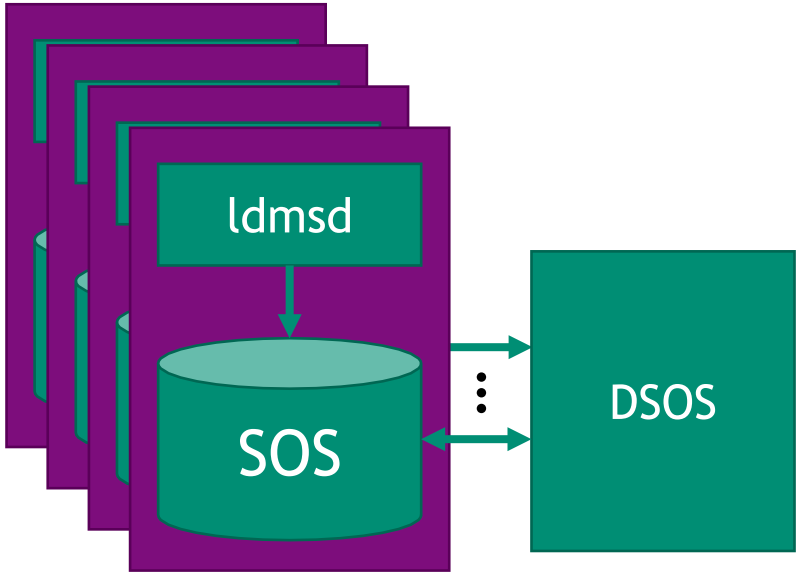
```
#timestamp,job_id,rank,name,type,current_kernel_count,total_kernel_count,level,current_kernel_time,total_kernel_time
1627835612.086679,10195735,1,Kokkos::View::initialization [diagnostic:Solver Field:B_Field:temp],0,1218,57972687,0,0.000005,182.693422
1627835613.709526,10195735,1,TimeAverage::Continuous,0,24758,57972788,0,0.000006,182.693428
1627835616.787472,10195735,1,MigrateParticles::count,1,3540,57972889,0,0.000001,182.693430
1627835620.448333,10195735,1,SolverInterface::Apply Trivial BC,0,7512,57972990,0,0.000002,182.693432
```

DSOS: Enabling Scalable Ingest and Queries for Analysis and Viz

Distributed Scalable Object Store (DSOS) is a scalable database with a variety of features which enable simultaneous large-scale data ingest and queries

- Designed specifically for large-scale HPC monitoring data ingest and query with flexibility to change and adapt as needs arise
- Coordinates databases across multiple devices and nodes to present a “single, unified” database to the end user
- High insert rate for continuous data collection
- Indices can be created or removed as needed for optimizing queries without reloading data
- Python, C, and C++ API and command line interface

Analysis Cluster


ldmsd
SOS
DSOS