SAND2021-13446C

Sandia National Laboratories

Lightweight Combined Application and System Performance Monitoring

Matt Mosby

LDMS User's Group Conference 2021

October 26 - 28, 2021

Virtual

U.S. DEPARTMENT OF ENERGY | NNSA

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.

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.

Acknowledgements

My involvement in creating/using the system I will present is extremely limited, I just got excited when I learned about it and asked a bunch of questions. The people coming up with the answers are:

James Brandt
Jeanine Cook
James Elliott
Ann Gentile
Si Hammond
Tony Nguyen
David Poliakoff
Benjamin Schwaller
Vanessa Surjadidjaja

Vision: Continuous Monitoring of Application/System Performance is Archived and Easily Accessible/Analyzed for Developer/HPC Admin Response

Challenge: Maintaining application performance on HPC systems is difficult

How can we inexpensively and automatically provide developers detailed performance data when they run tests?

Challenge: HPC system state can strongly influence application performance

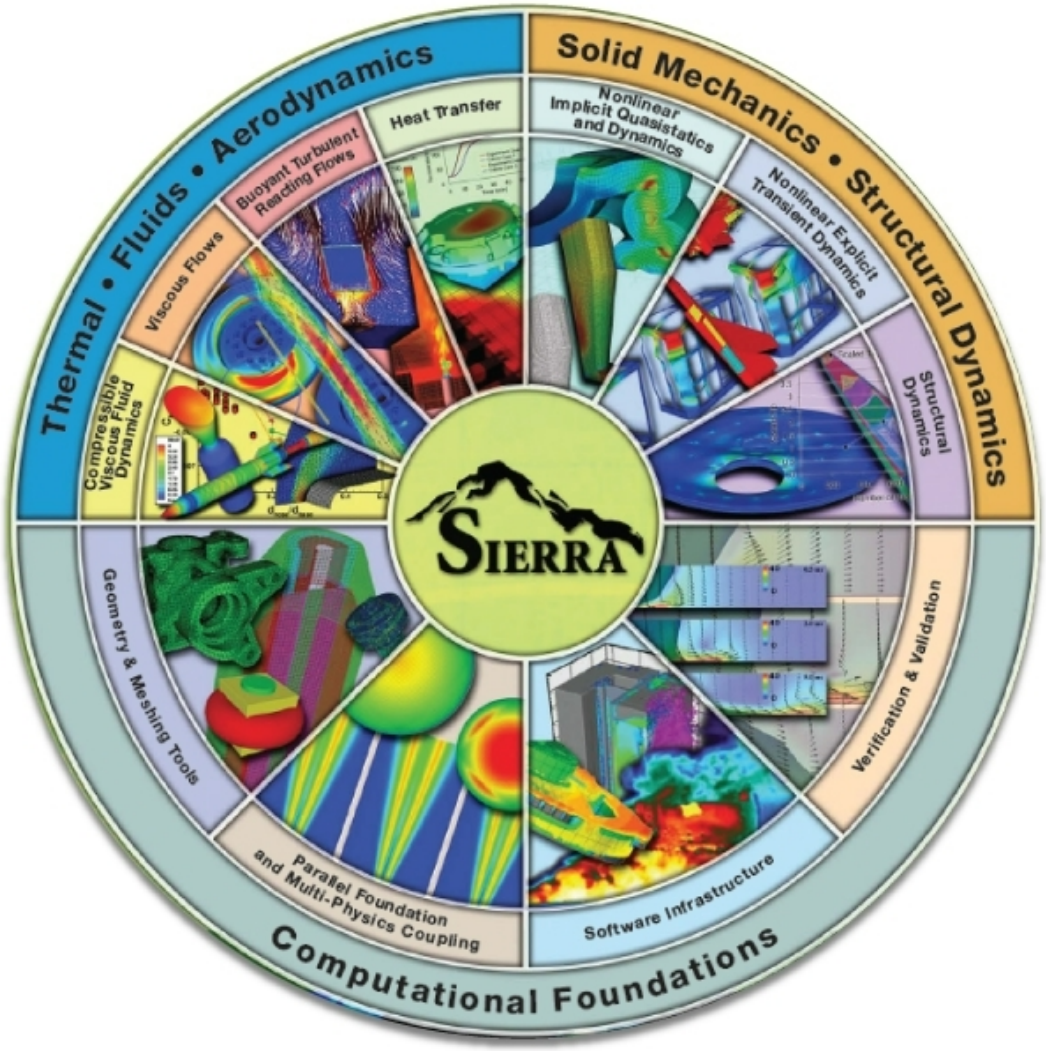
How can we help developers track performance over time and understand variations in run-to-run performance given the state of the HPC system?

How can we help the developer/user/admin community identify system issues?

Challenge: Resolving application performance issues “in the wild” after deployment is expensive

How can we enable developers to identify performance issues “in the wild” and respond proactively?

Use Case: SIERRA High Performance Multi-physics Engineering Analysis Apps



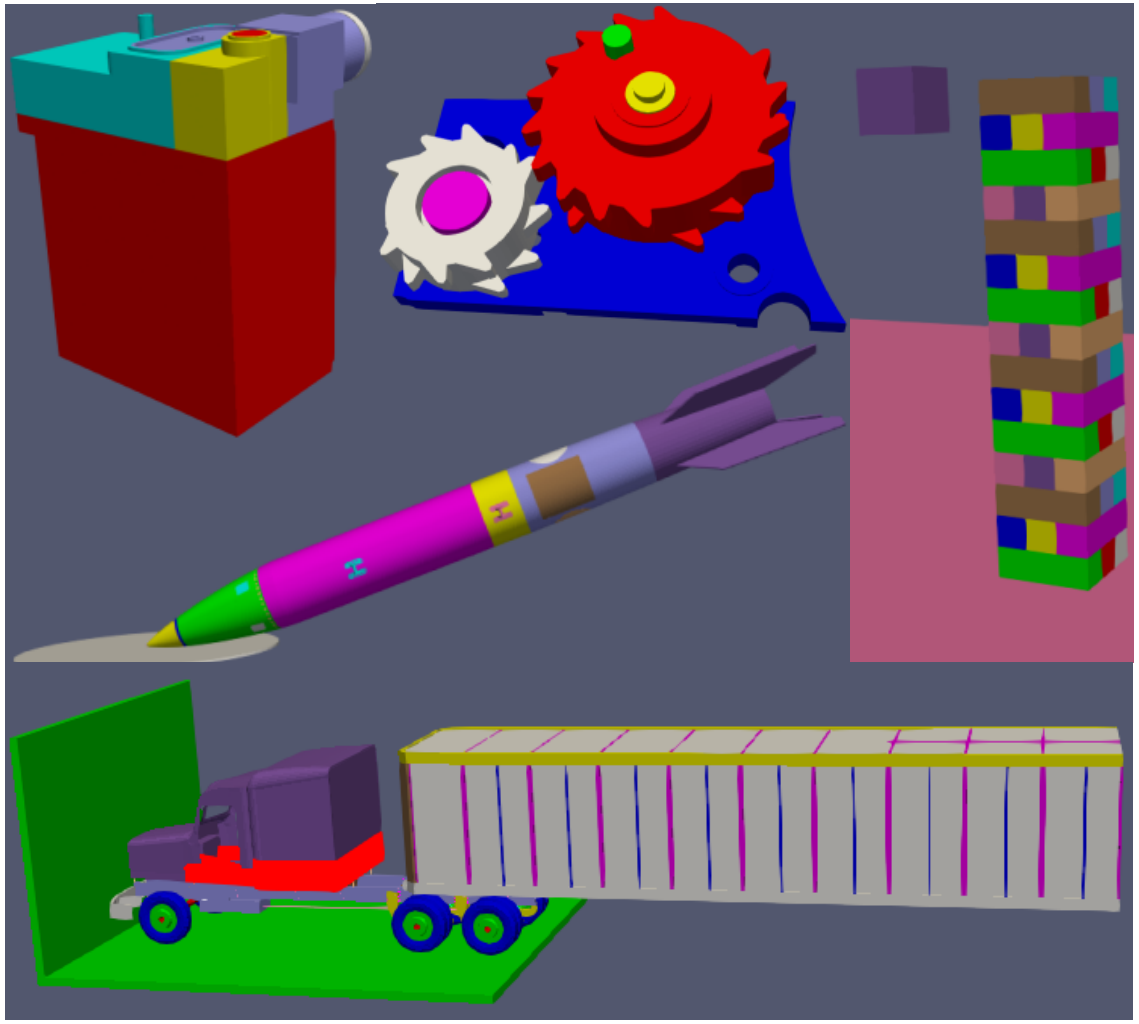
General purpose analysis software used by many government agencies to answer engineering questions in the national security problem space

One of the most-executed applications on SNL HPC resources

Often used as part of acceptance suites for HPC hardware, tools, etc.

Defining feature is high performance on variety of commodity and advanced platforms, e.g., Trinity, Sierra, Crossroads & El Capitan (future)

How does SIERRA maintain cross-platform performance?



Maintain large performance test suite (~450 tests)

Run performance tests on important HPC platforms nightly

Pass/fail on rudimentary criteria

- Total runtime
- Memory high-water
- Output/results comparison

How do developers track variability and/or trends in the performance tests?

How does SIERRA maintain cross-performance?

Scrape log files and store limited data for historical/longitudinal analysis

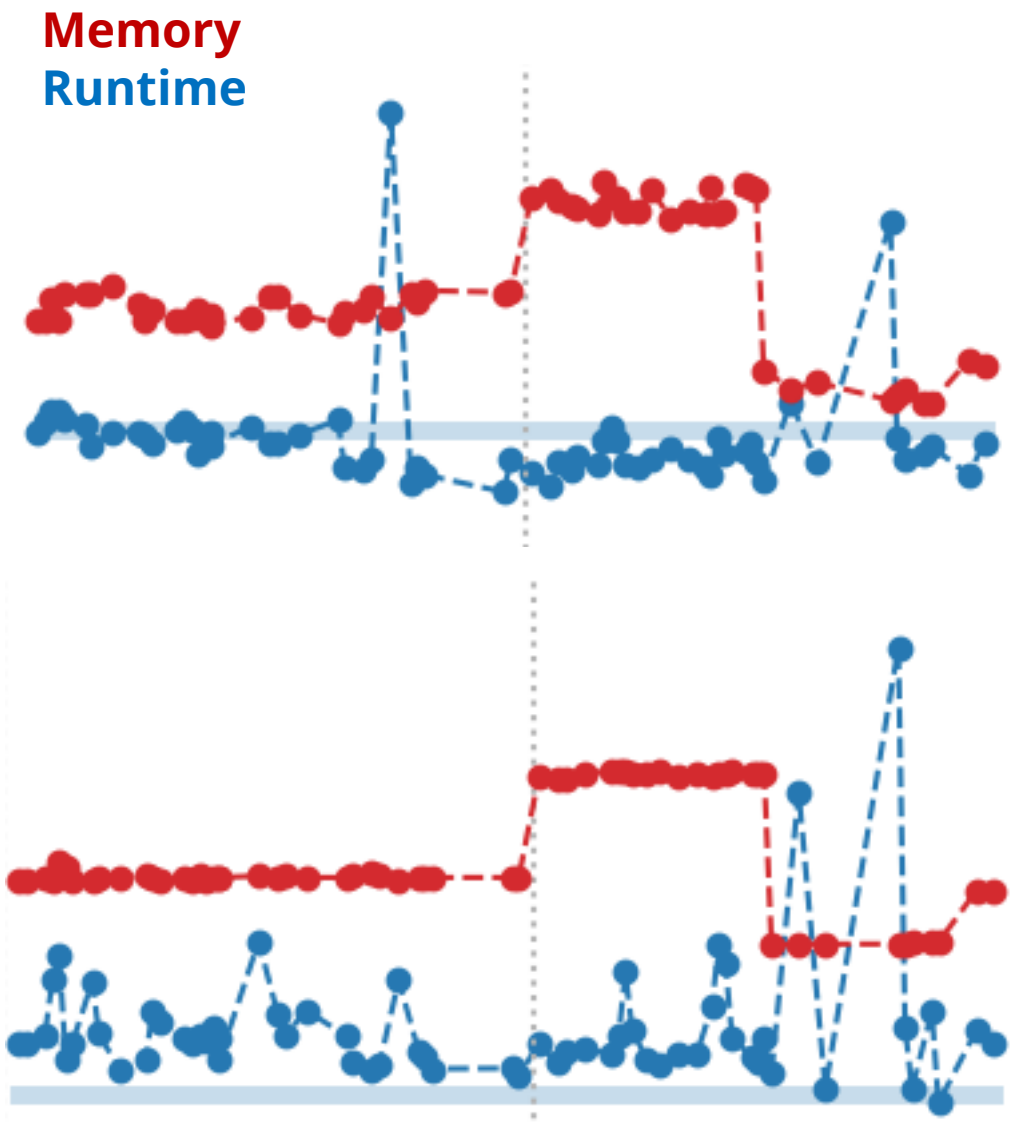
Historical data can help identify issues

- Correlation with key events (e.g., TPL/system updates)
- Provide bounds for root-cause analysis

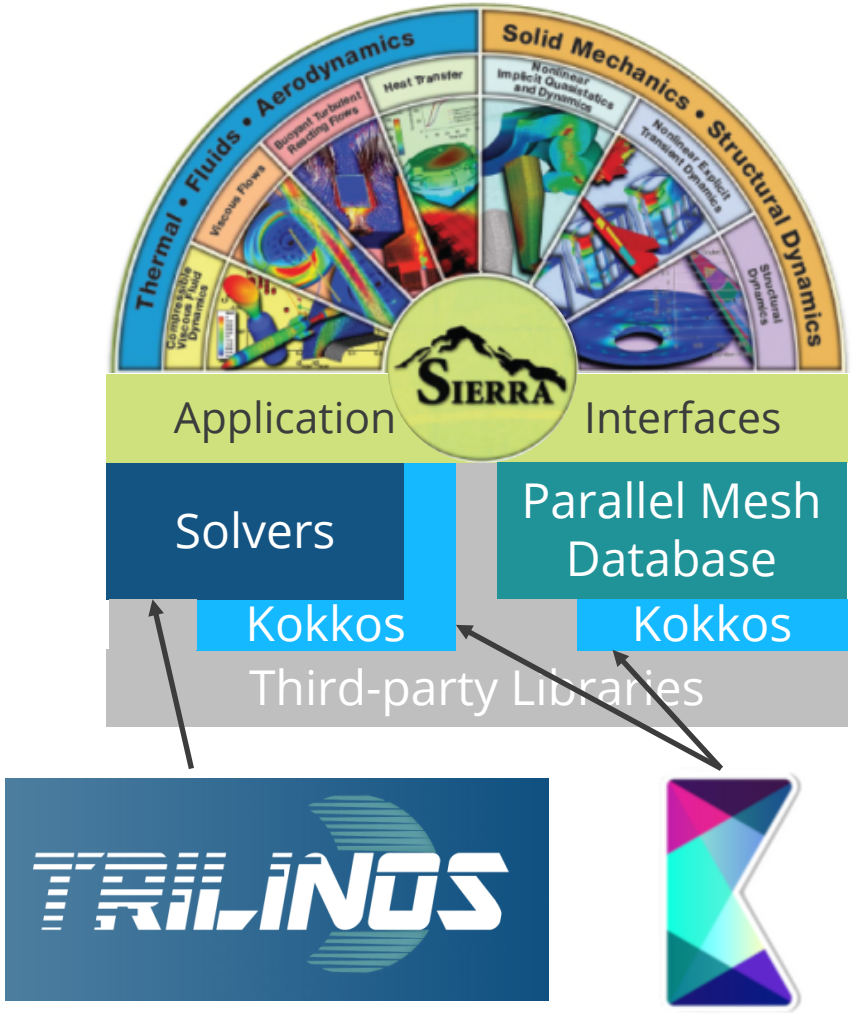
Historical data like this **cannot** identify root causes of performance issues

How can we help developers easily identify root causes of performance loss?

How can we help developers/HPC admins identify system issues?



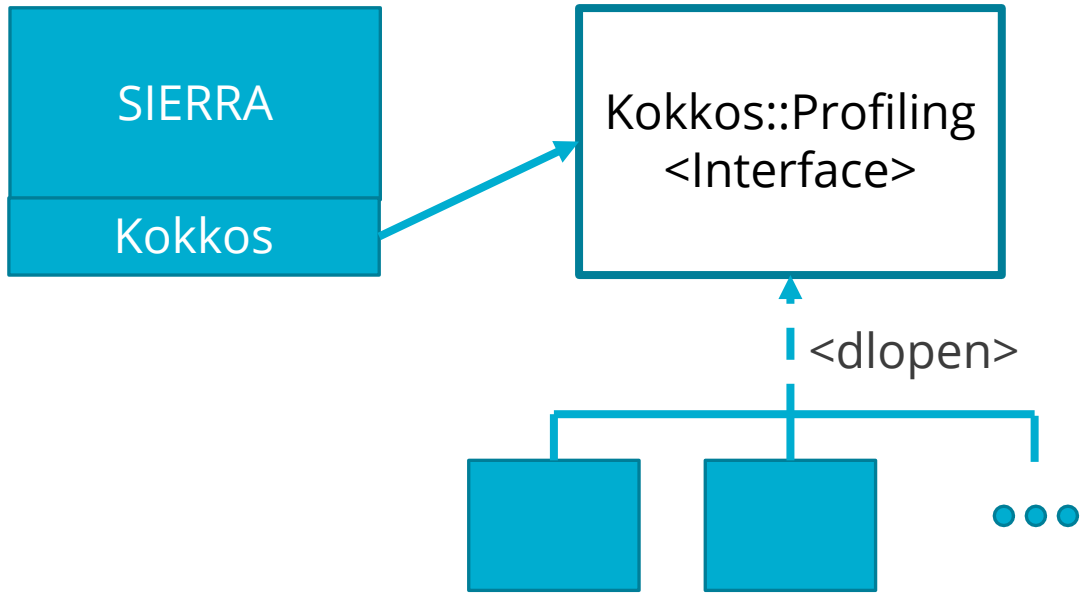
SIERRA Architecture Provides Common Entry Point for Performance Data Gathering



SIERRA has standardized on Kokkos as a performance portability abstraction

Kokkos provides a profiling interface that enables runtime link of an arbitrary profiling library

Track events: e.g., regions, kernels, copies



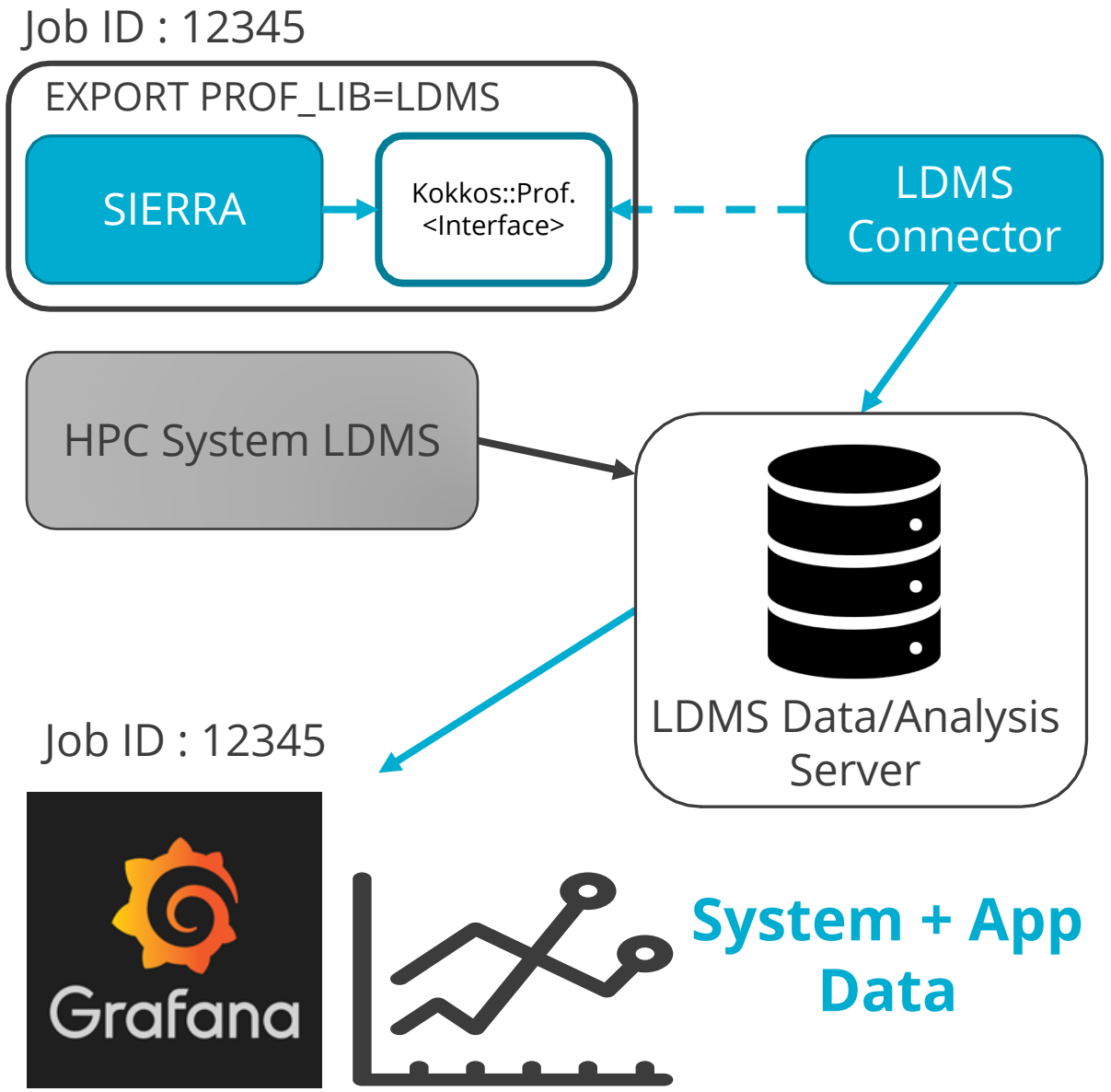
Prototype: LDMS for System and Application Performance

Prototype capability uses LDMS streams to transmit/store timestamped application performance data

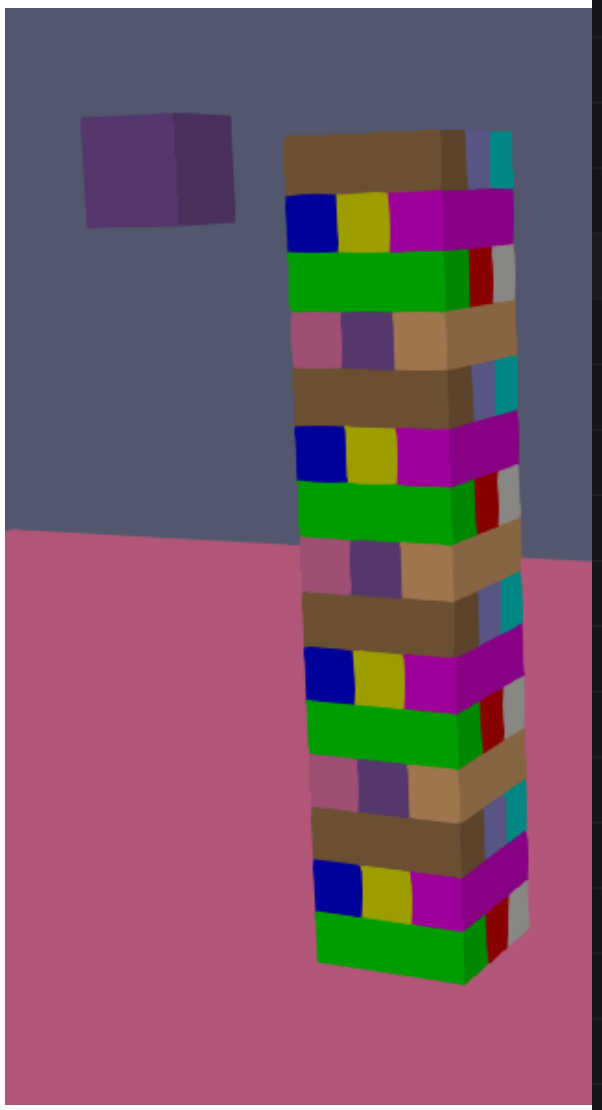
Co-location of system data enables overlay of system state with application performance

Graphana used to dynamically query/generate visualizations of data

Big step towards answering some of the challenges faced by app devs



Prototype: LDMS for System and Application Performance

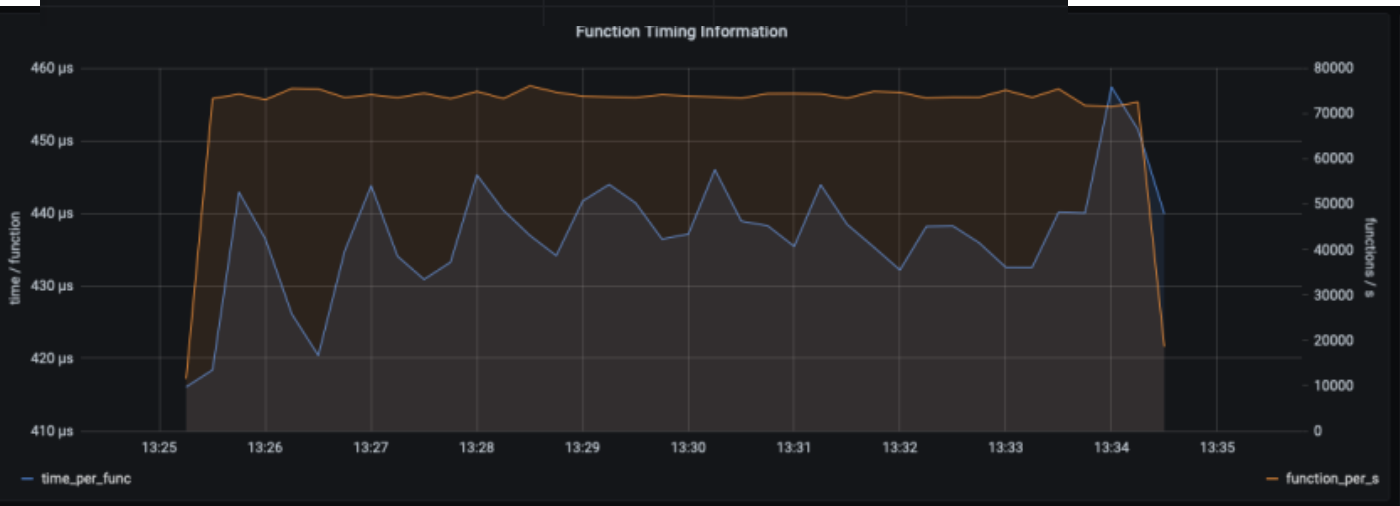


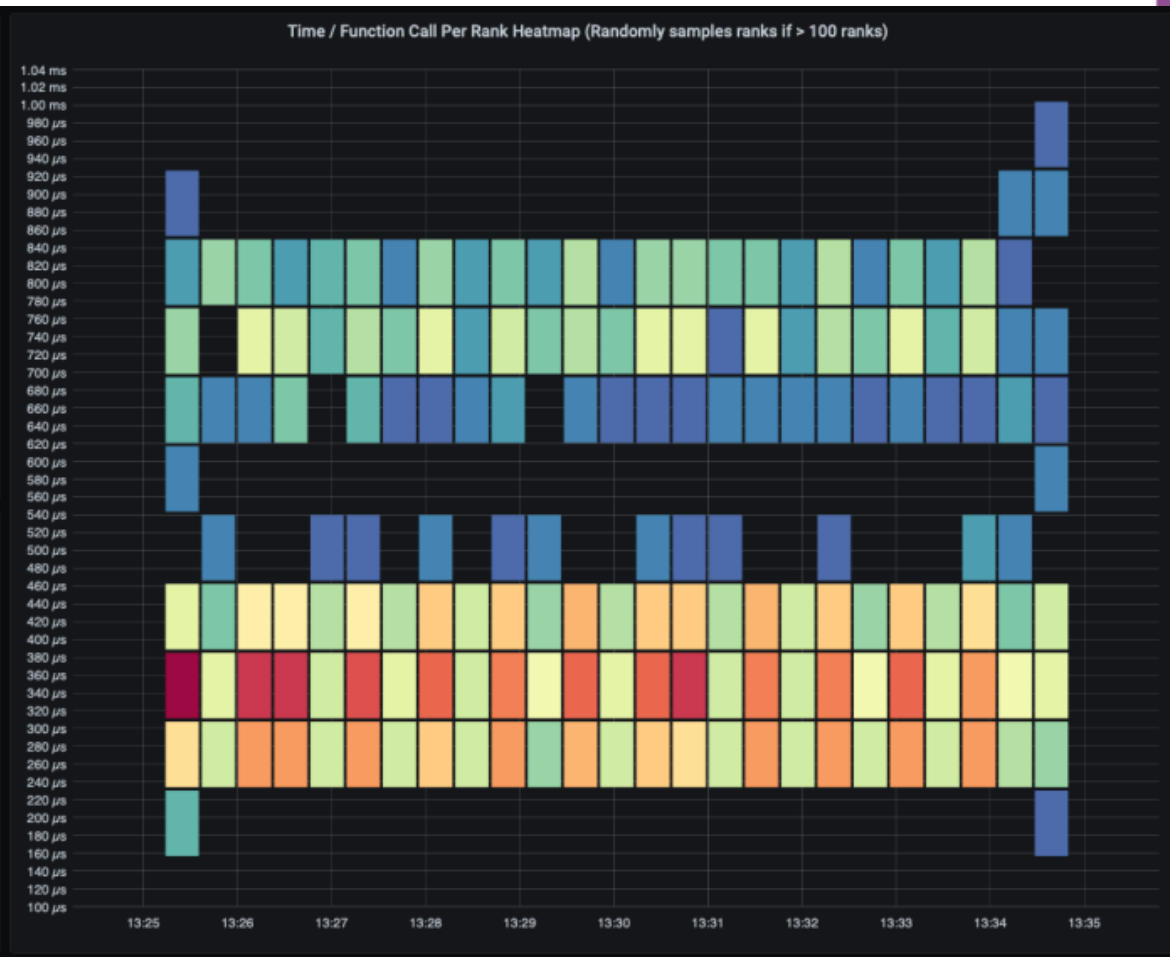
Kernel name	Aggregate c	Aggregate tim	Average time
default compute internal force	25063756	30.3 min	72.6 µs
define face-face interactions	10084547	1.2259 hour	438 µs
define lofted face-face interactions	10084042		
Kokkos::View::initialization [DualView::...	6401077		
Kokkos::View::initialization []	6398855		
LocalSumInteractionMassToNodes	6387442		
ZN6sierra4Cont23update_predicted_co...	2667511		
compute force from interactions	2666501		
Kokkos::View::destruction []	2135241		
zero net contact force	532472		
EnforceExplicitContact – update contac...	531462		
compute energy globals	531058		
process central difference operator	530957		
EnforceExplicitContact – update orig pr...	530250		
process nodal acceleration	529947		
ComputeMassScale	528937	44 s	82.4 µs
ZN3mtk26ConcatenateThreadLocalDat...	528735	293 ms	554 ns
DashExplicitEnforcement::add_nodal_a...	527624	2.1 min	244 µs

Prototype: LDMS for System and Application Performance

Kernel name	Aggregate c	Aggregate tim	Average time
default compute internal force	25063756	30.3 min	72.6 µs
define face-face interactions	10084547	1.2259 hour	438 µs
define lofted face-face interactions	10084042	3.1 s	305 ns

Detailed view of execution across processors and time, side-by-side with system information





Wrap-up, what's next?

Prototype demonstrates ability to inexpensively gather application performance data and overlay with system state

- Lightweight profiling
- Can point to system issues with appropriate context

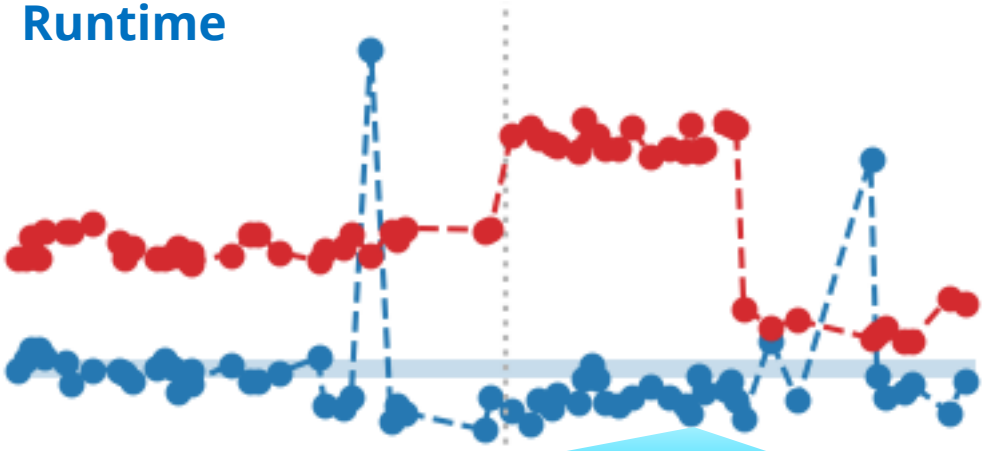
Can we associate data to enable longitudinal monitoring/analysis?

Can we engage app users to enable monitoring for proactive dev response to performance issues "in the wild?"

What are some other uses for this kind of application monitoring?

What might be possible if we can easily…

Memory
Runtime



THANK YOU