



Generating HPC Job Profiles and Expectations with Time-Series Data

Joshi Fullop
Brett Layman

7/16/2020

Disambiguation and Basis

- ▶ Job Profiling in this context:
 - ❑ Measure machine resources used by an application
 - ❑ Not an embedded library (tracking code with function calls)

- ▶ Looking at metrics taken from LDMS such as:
 - ❑ Load
 - ❑ Memory
 - ❑ Bandwidth

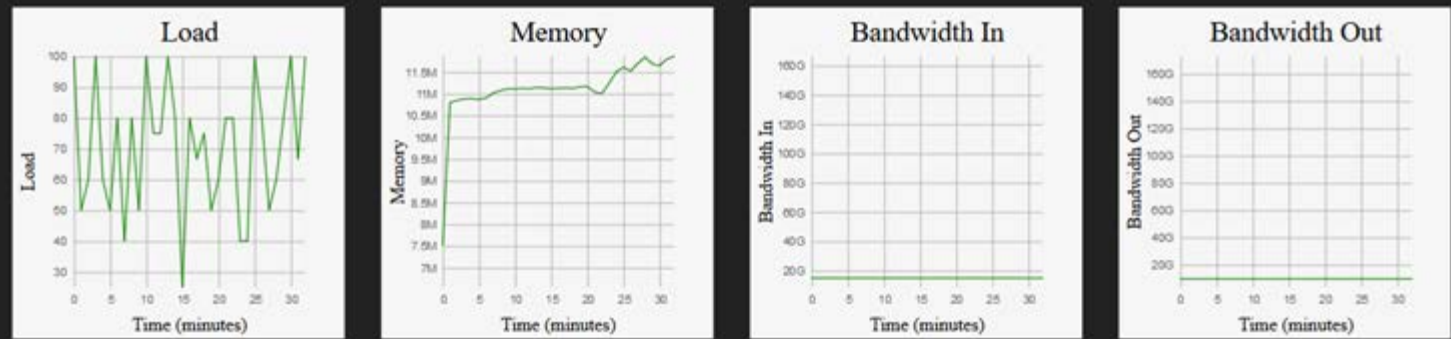
Job Profiles

- ▶ Combine time series and organizational data into JSON objects
 - ❑ Top level: Job name, start time, node count, etc.
 - ❑ Metric series: arrays
- ▶ Node series are statistically aggregated into summary series
 - ❑ Mean, min, max, standard deviation
 - ❑ Calculated as data arrives
 - ❑ Pre-summarized data enables faster analysis and instant visualization

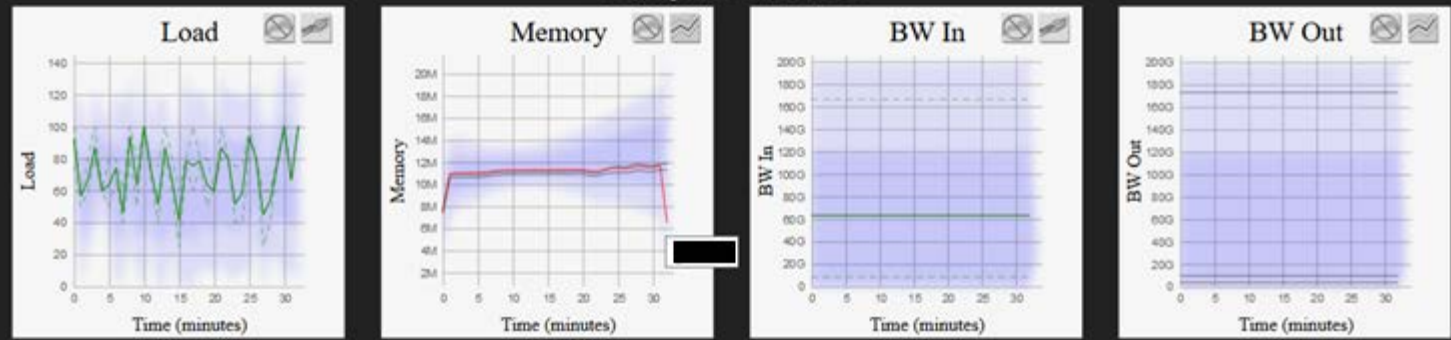
Job Profiles

Job ID [REDACTED]	Account [REDACTED]	User [REDACTED]	WC Key [REDACTED]
Submit Time [REDACTED]	Start Time [REDACTED]	End Time [REDACTED]	Run Time (sec) [REDACTED]
Node Count [REDACTED]	Node List [REDACTED]	Exit Code [REDACTED]	

Head Node

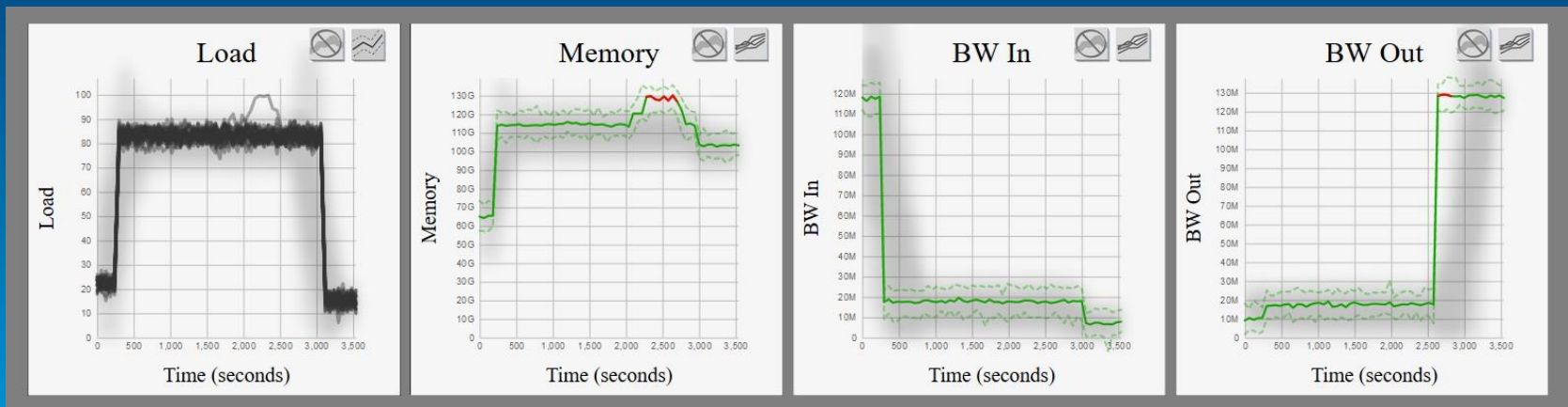


Compute Nodes

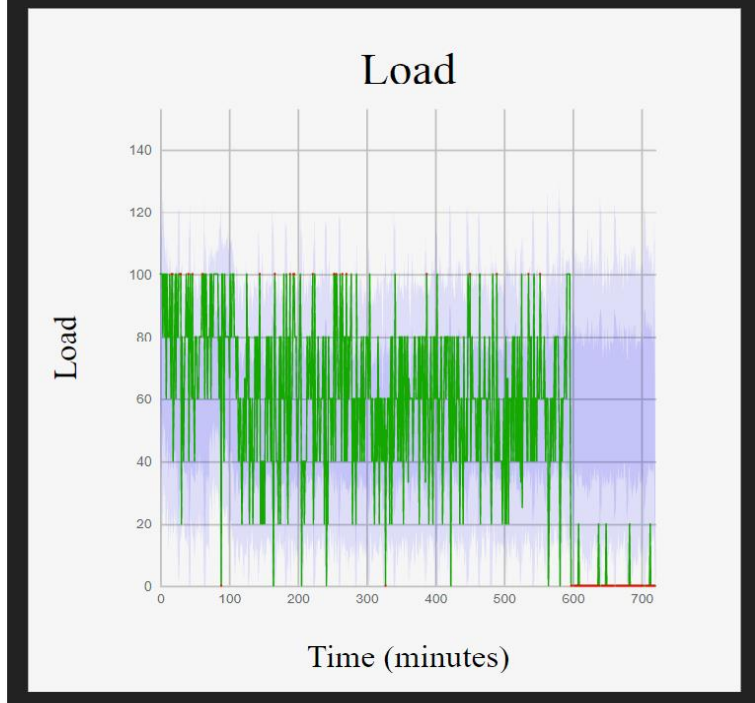
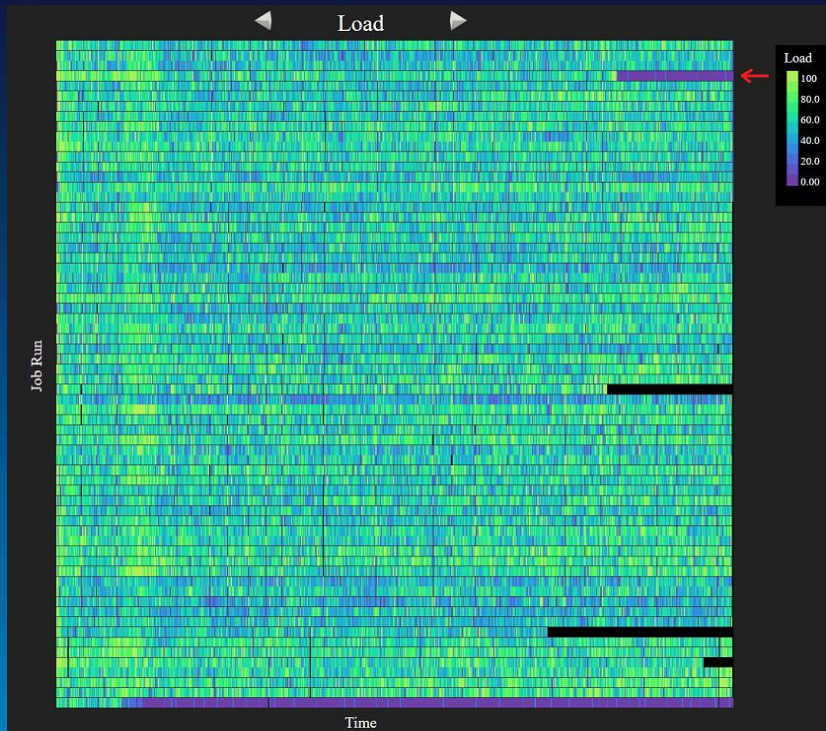


Job Expectations

- ▶ An expected performance profile based on past successful runs
 - ▣ Statistical summary: Average, Standard Deviation, etc.
 - ▣ Can be visualized as a cloud path
 - ▣ Allows for detection of deviant jobs
 - ▣ Those jobs can be canceled well before wallclock termination

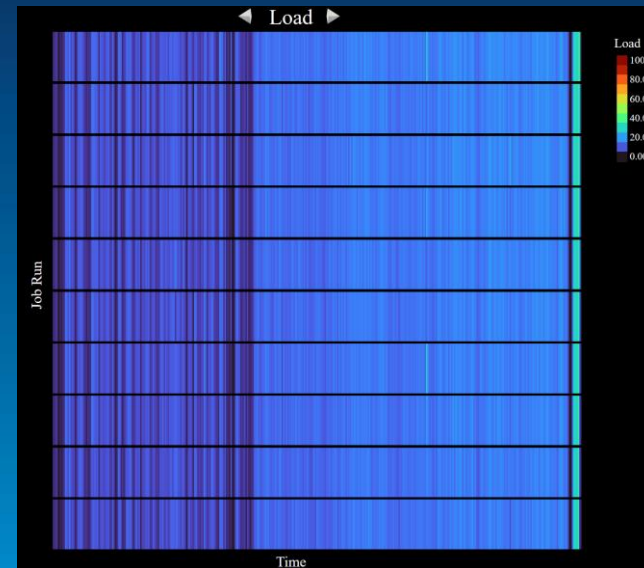
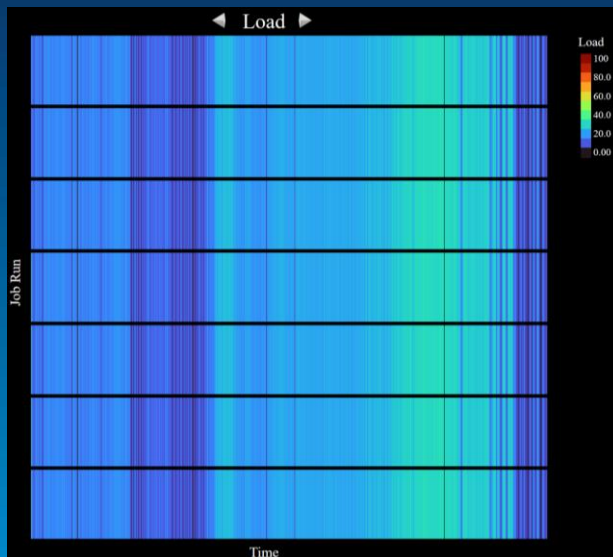


Real World Example



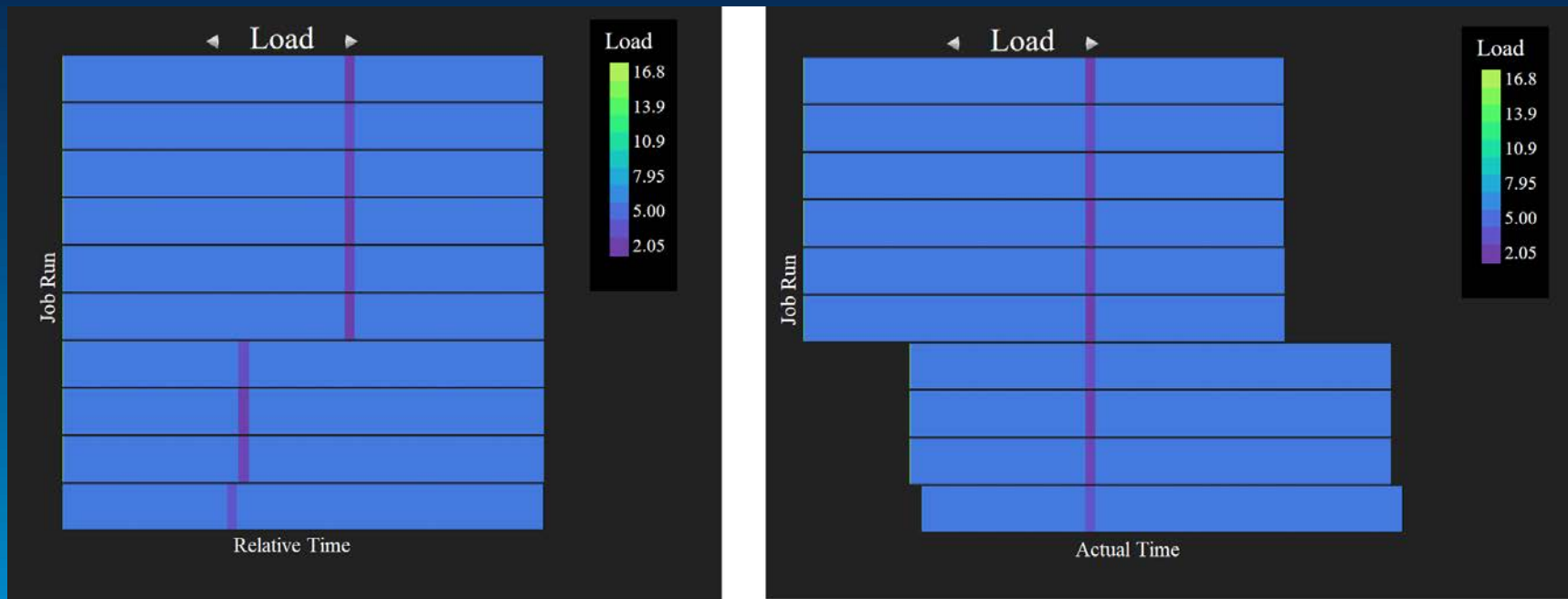
Job Consistency

- ▶ Certain jobs were highly consistent
 - Tended to be smaller jobs (single node)
 - Some more consistent than others
 - Important for utility of job profiles and expectations

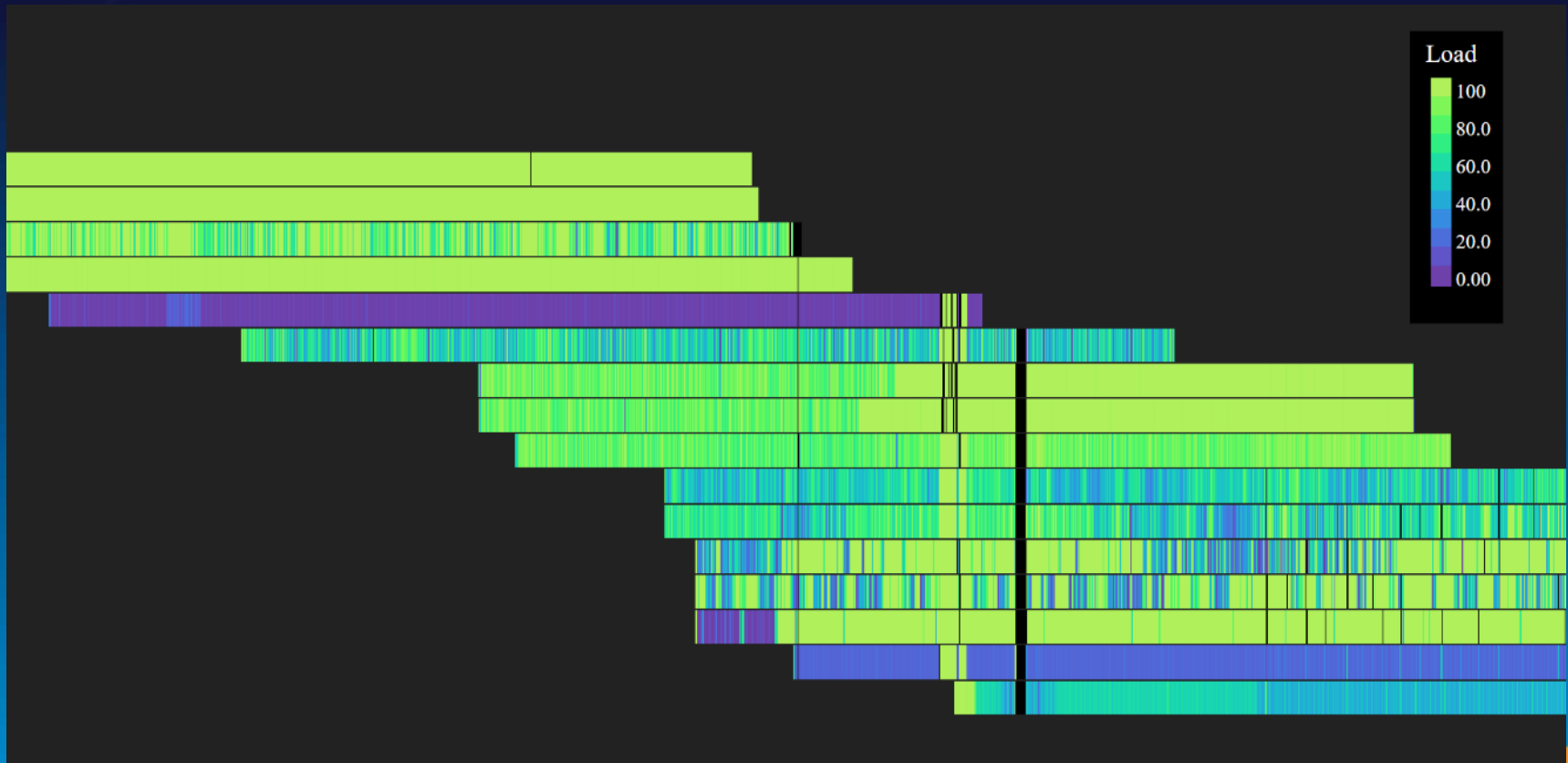


System-Wide Events

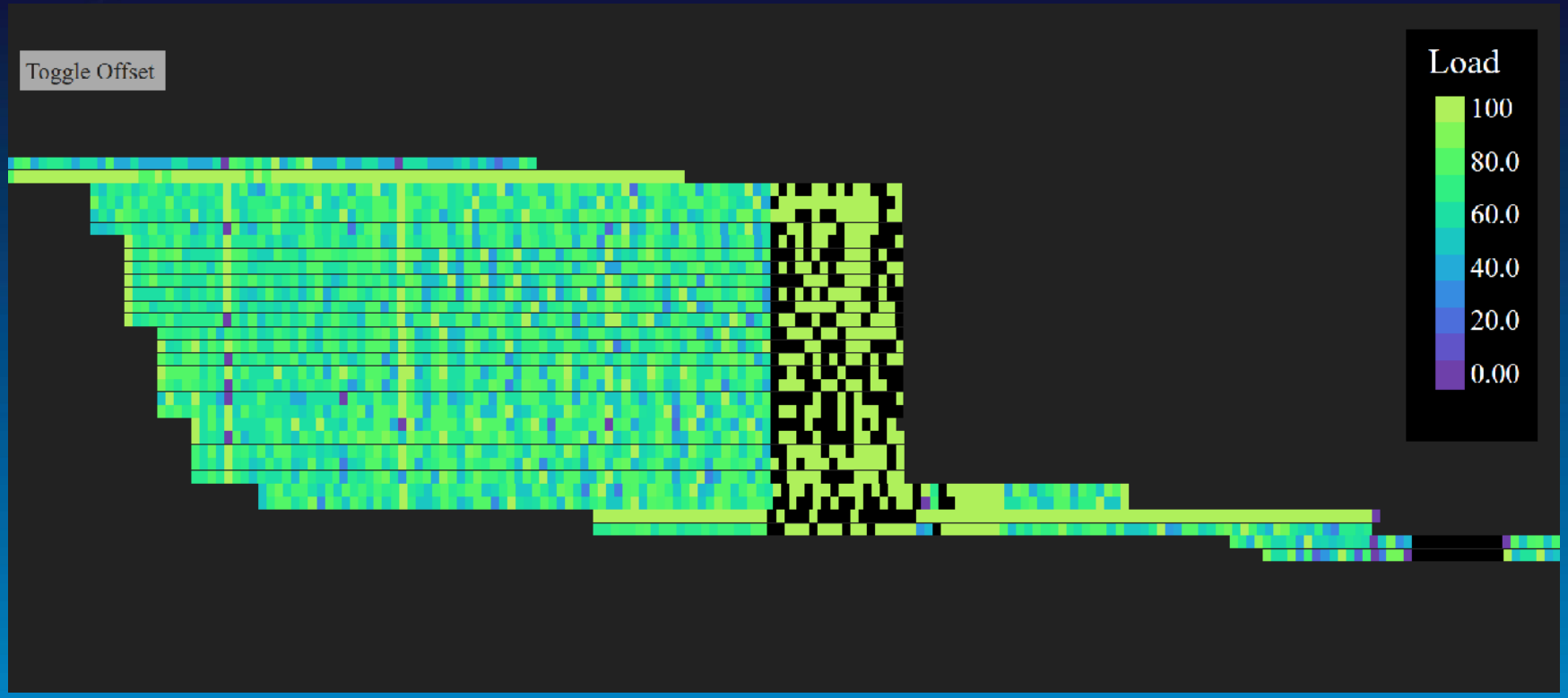
- ▶ When you shift job series by start time, you can visualize anomalies that occur across many different jobs (i.e. the system) at one point in time.



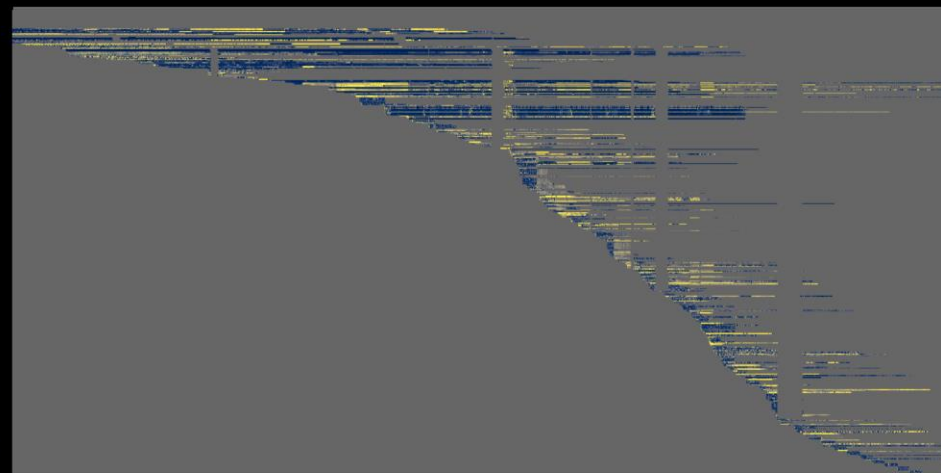
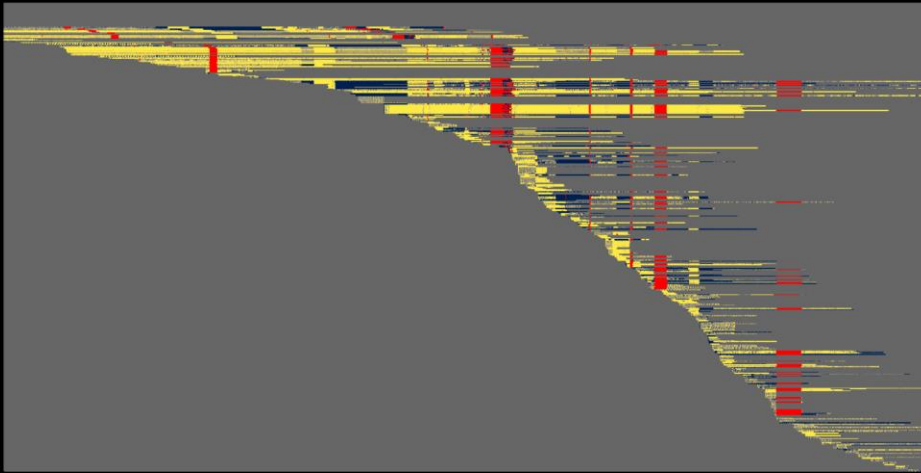
System-Wide Events



System-Wide Events



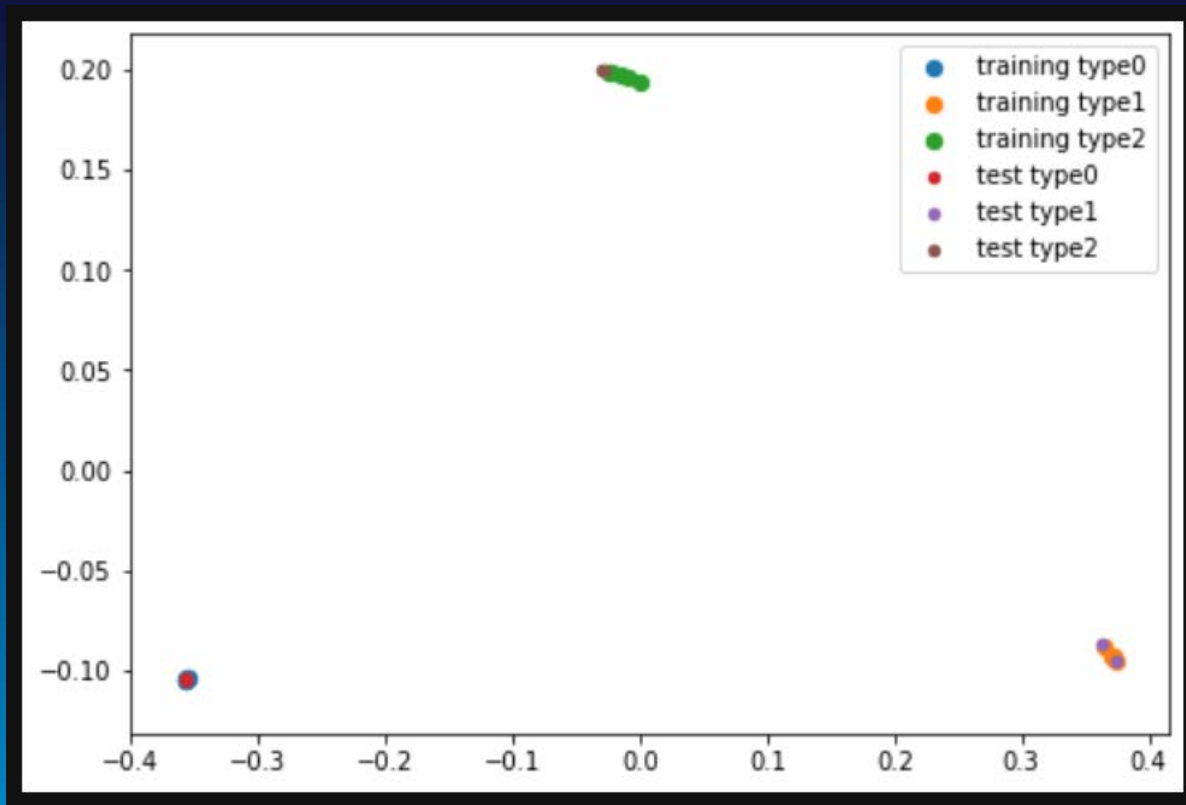
System-Wide Events



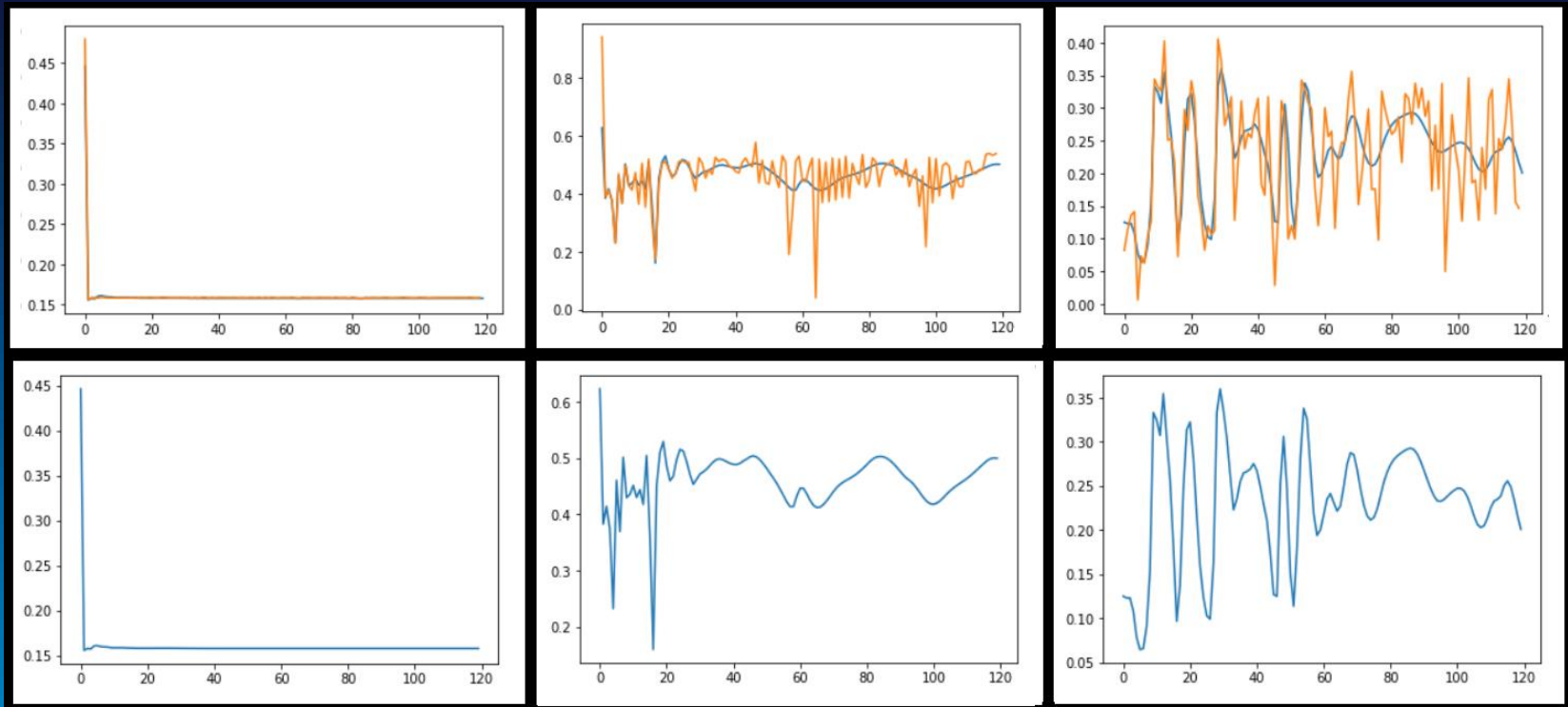
Job Classification

- ▶ Jobs aren't always grouped into workload types
 - ❑ Even when labeled, the grouping may be incorrect
 - ❑ Would be beneficial to automatically categorize jobs based on their time series
- ▶ Recurrent Neural Nets (RNNs) work well for encoding sequential information.
 - ❑ Handles offset and stretching of series nicely.
 - ❑ Long Short Term Memory (LSTM) network is especially good at encoding long sequences.

Experiment results

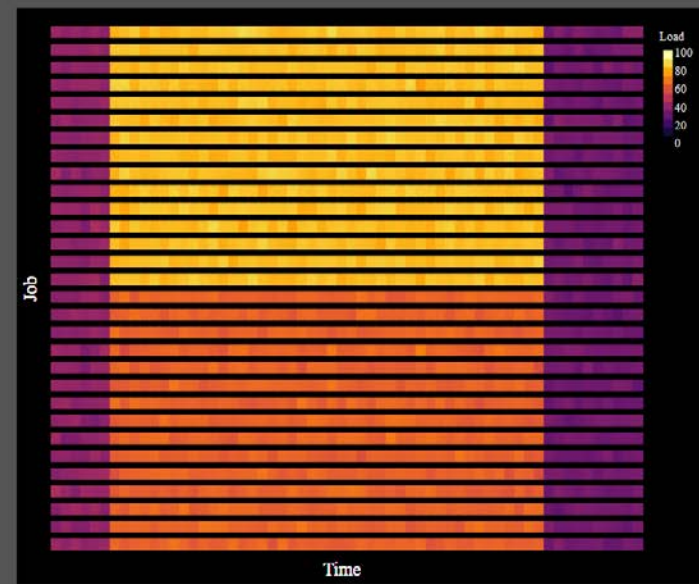
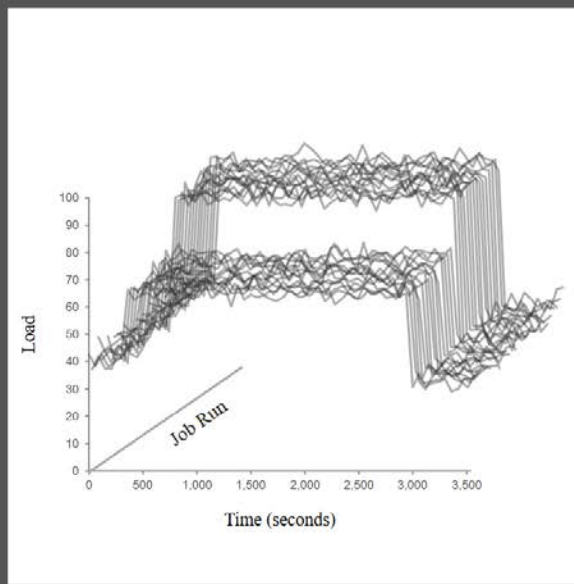


Experiment results



System Benchmarking

- ▶ Run same job periodical to pick up on system-wide changes



Resource Scheduling

- ▶ Use job expectation to estimate resource use
 - ❑ Use this estimation when scheduling jobs
 - ❑ Allows you to avoid going over budget on resources (such as power)

System Utilization and Planning

- ▶ Taking job profiles in aggregate, we can determine how a supercomputer is being utilized
 - ❑ Determine which resources are creating bottlenecks
 - ❑ Design future supercomputers based on identified needs

Log Analytics

- ▶ We can add messages to the log data when a job or node deviates out of expectation
 - ❑ Logs can be used to help explain why the job deviated at that point
 - ❑ This helps with the issue of unlabeled data in log analysis (on a time-step level)
 - ❑ Helps to validate whether a log anomaly is actually impactful

Summary

- ▶ Job Profiles and Expectations provide important insights into workloads
 - ❑ Job Profile: window into how a job is running
 - ❑ Job Expectation: Is that job behaving as expected
 - ❑ Provides us with actionable information

- ▶ Machine learning can be used to group job runs into workload types
 - ❑ Identified groups can then be used for generate expectations

- ▶ Profiles and Expectations also enable the study of:
 - ❑ System-wide events, tracking system changes
 - ❑ System resource utilization and scheduling
 - ❑ Marking log data for further investigation or failures/anomalies