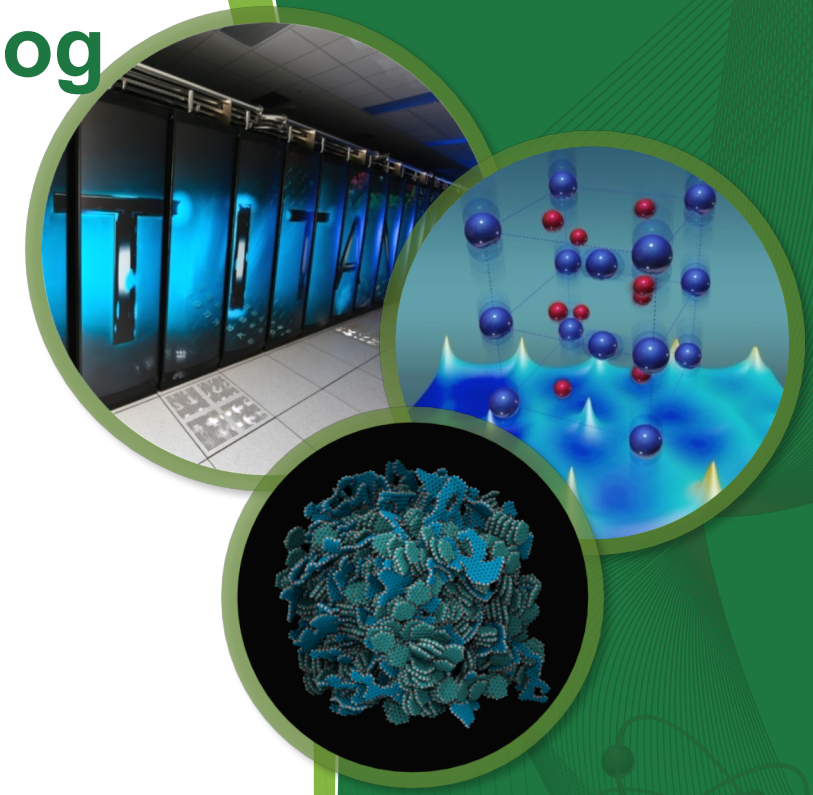
A Big Data Analytics Framework for HPC Log Data: Three Case Studies Using the Titan Supercomputer Log



Byung Hoon Park, Yawei Hui
Swen Boehm, Rizwan Ashraf,
Christian Engelmann

Oak Ridge National Laboratory

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory

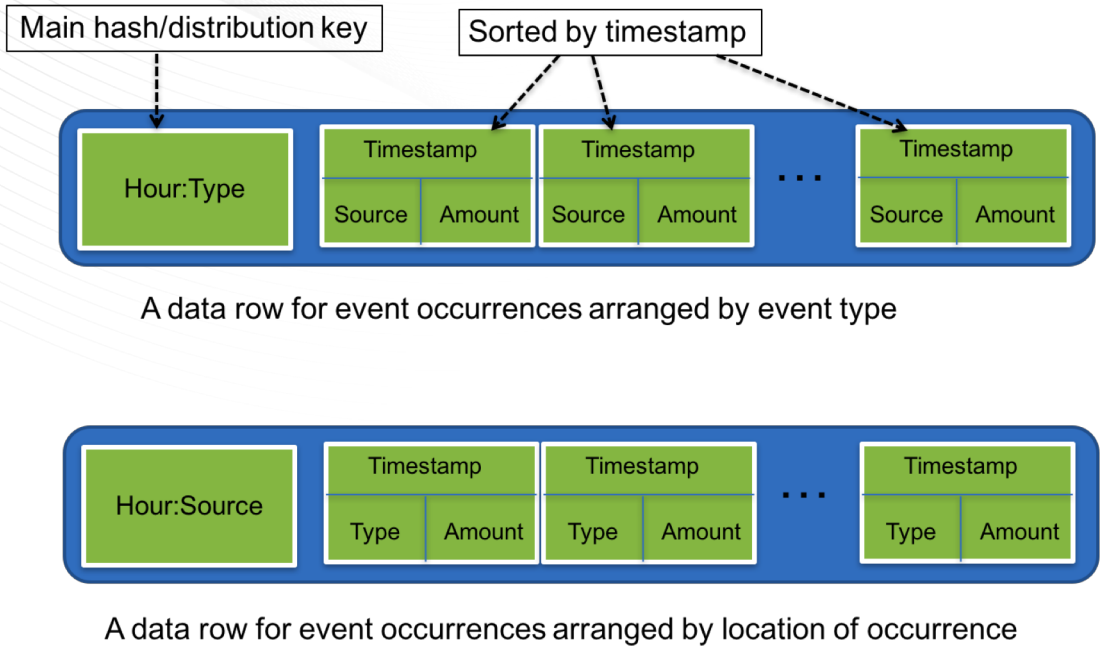
Background

- HPC Reliability, Availability, and Serviceability (RAS) Logs are valuable resources to obtain clues for system status, failures (or system anomalies), performance degradation, etc.
- Many researchers want access to logs, However …
  - Data are often heterogeneous, unstructured, huge in volume, ant not stored in database for retrieval.
- ORNL's *Log processing by Spark and Cassandra-based ANalytics* (LogSCAN) is designed to allow numerous researchers to perform their log analytics
- This presentation introduces LogSCAN and three example use cases.

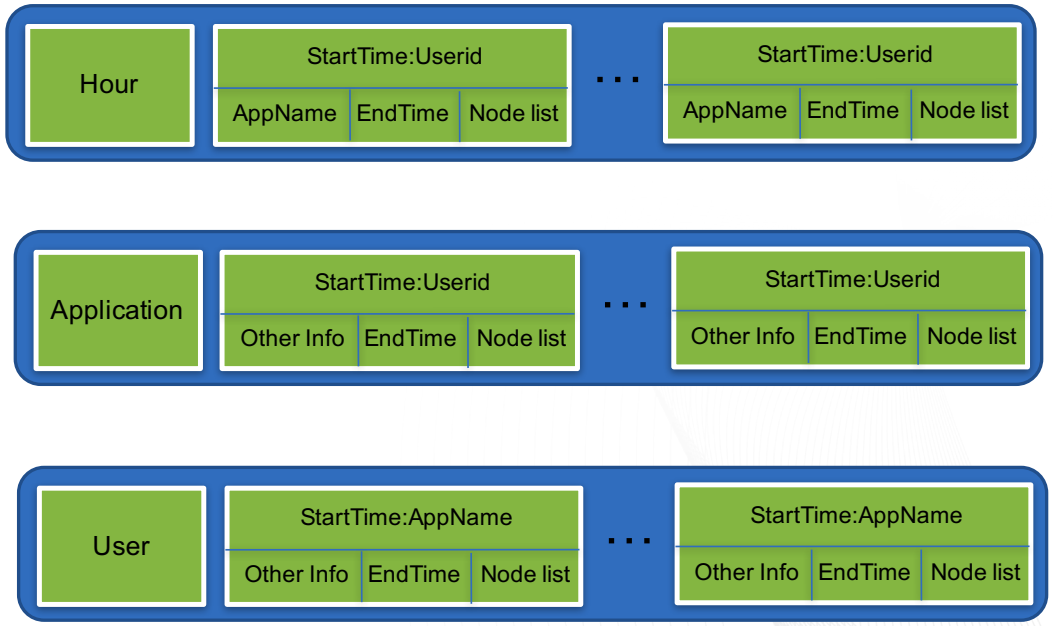
LogSCAN: Design Principle

- Scalability
  - Database should be capable of accommodate ever-increasing volume of data
- Low latency and High availability
  - Should serve interactive analytics and near real-time query requests.
  - Should survive failure.
- Flexibility
  - Should represent various types of events
- Time series friendly
  - Time series analytics over an arbitrary time interval

Data Model with Redundancy for Efficient Retrieval



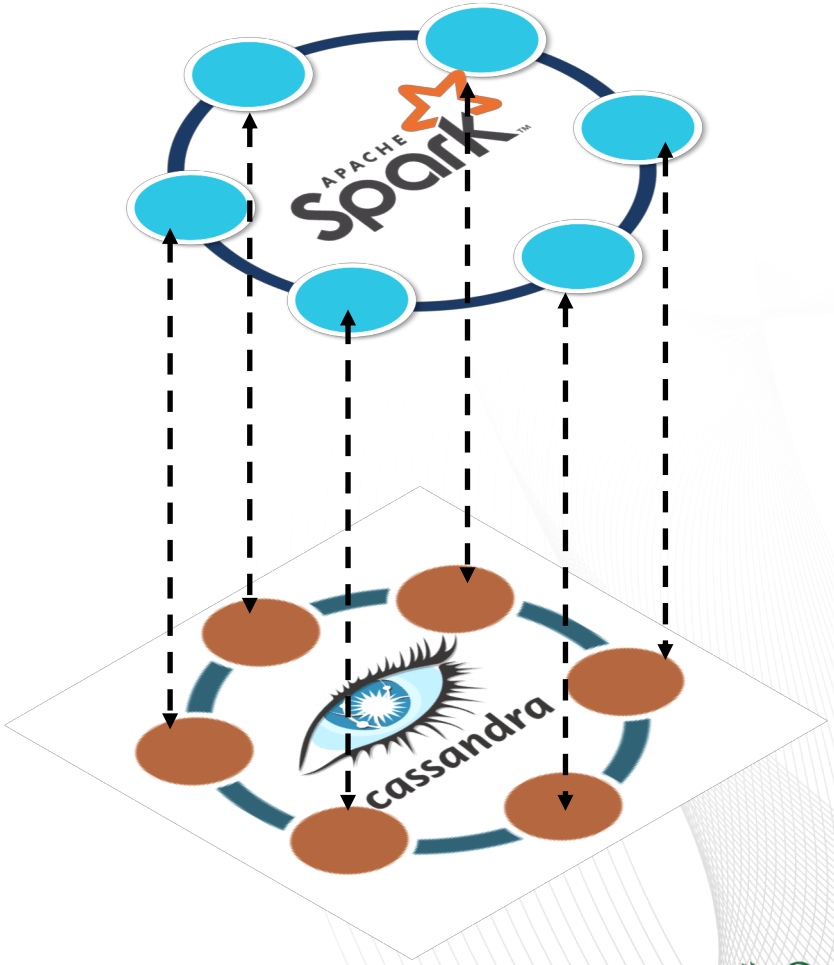
Event Data Model



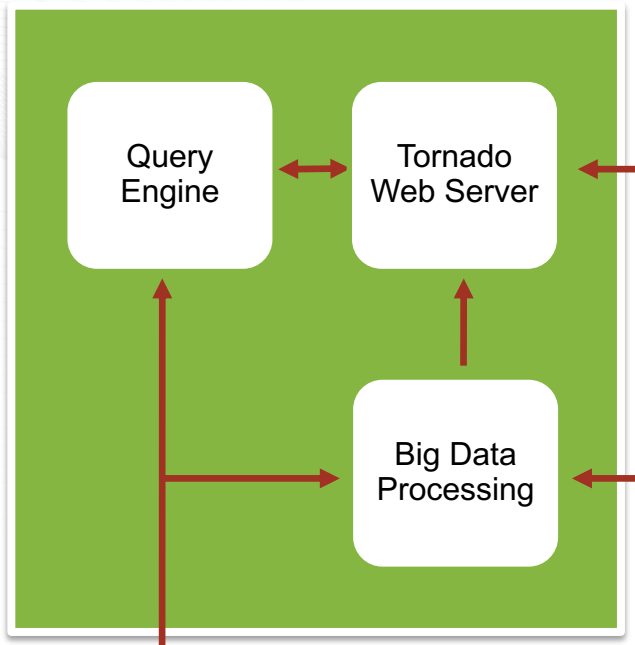
Application Run Data Model

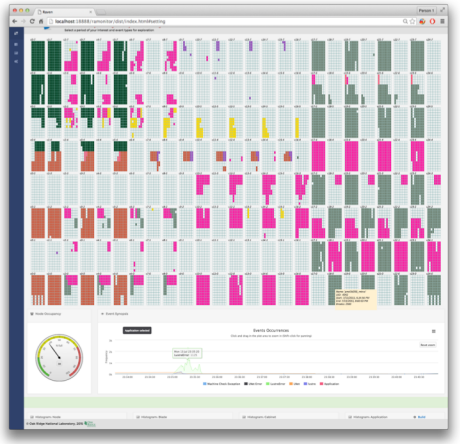
Spark: Computation Model for Cassandra

- To maximize locality, a spark node is installed in each Cassandra node.



LogSCAN Architecture





Frontend

Spark Job Server

REST

Java

Scala

Python

Three Analytic Example Use Cases

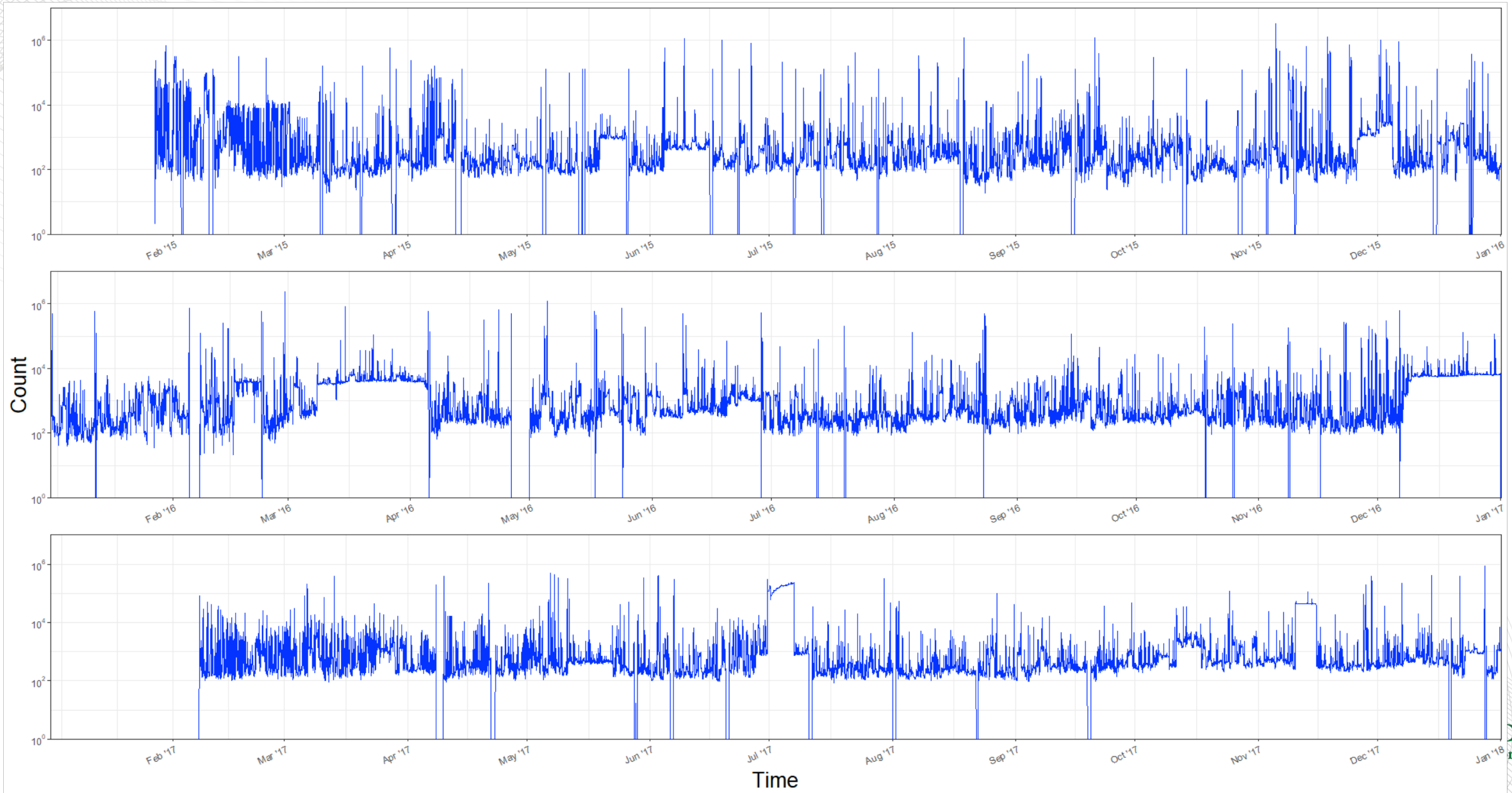
- System Information Entropy (SIE)
  - To represent system status by a simple metric
- Event correlation and frequent pattern
  - To identify correlated event types in their occurrences
- Characterizing application runs in terms of events
  - To study any malignant correlation between events and application runs
- All accessed LogSCAN using
  - R, python, and Jupyter Notebook

Data: Titan Logs between January 2015 and March 2018.

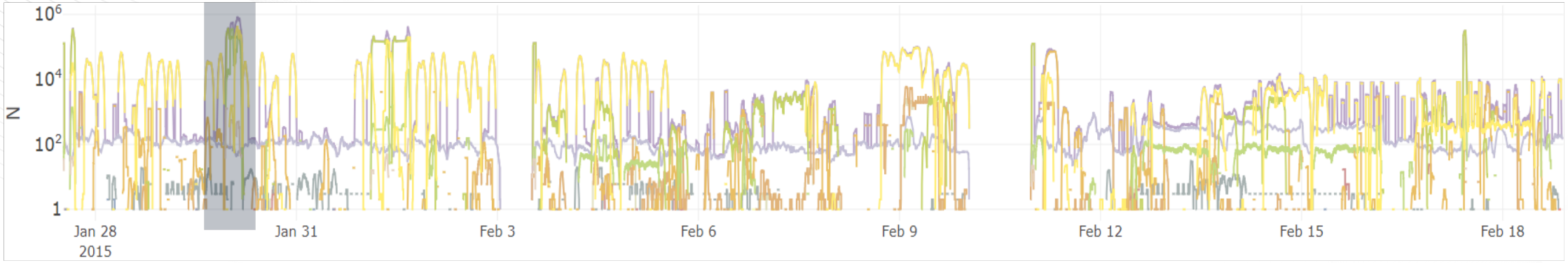
Counts	Percentage	ID	Description
10	0	1	DVS Confusion
2,998,492	2	2	NVRM Xid
5,671,348	4	3	Machine Check Exception (MCE)
1,229	0	4	NVRM DBE
49	0	5	Unknown GPU Error (UGE)
302,969	0.2	6	Graphics Engine Error (GEE)
5,732	0	7	Kernel Panic
782,337	0.5	8	Out of Memory (OOM)
16,938,194	11.6	9	HWERR
1,215,780	0.8	10	Seg. Fault
43,268,141	29.5	11	Lustre
31,498,746	21.5	12	LNet
992,997	0.7	13	LNet Error
42,809,426	29.2	22	Lustre Error
146,485,450	100		

Summary of Event Types and Their Occurrences.

Event Occurrences: Bird's Eye View



How Do We Represent Event Occurrences by a Simple Metric?

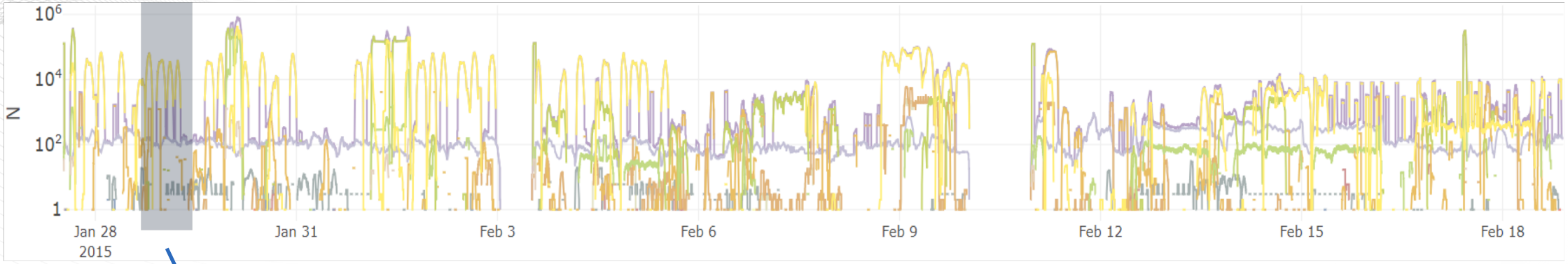


Each event has the source, i.e. where it occurs.
This spatial information should also be considered.

- Amount of event occurrences
- Types of events that occur
- Location of events

→ system information entropy

System Information Entropy: SIE



	Feature 1	Feature 2	...	...	Feature N
Record 1					
Record 2					
...					
...					
...					
...					
Record M					

Principal Components in Feature Space

SVD $\Longrightarrow \sigma_i$

σ_i: *i*-th variance out of k eigenvalues of the SVD decomposition

Shannon Entropy

$$H = -\sum_1^k \xi_i log_b(\xi_i)$$

System Information Entropy (SIE)

$$W(t) = 10^{H(t)}$$

Approximate the number of dominant features

Composition of Events into two Tables

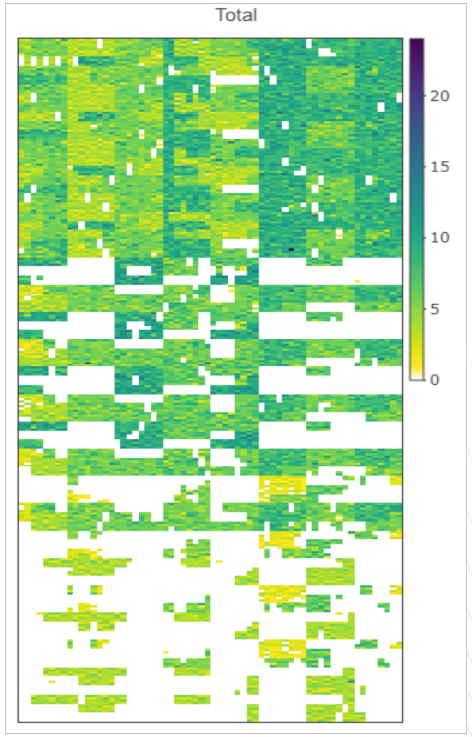
Source vs Type

Source	Machine Check Exception	Out of Memory	Seg. Fault	Lustre
c0-2c2s1n2	1	0	0	0
c10-4c0s0n0	0	0	0	4
c10-4c0s3n0	0	1	0	3
c10-4c1s6n1	0	0	3	1

Example table where each row is frequencies of event types occurred at a compute node, and each column is event type.

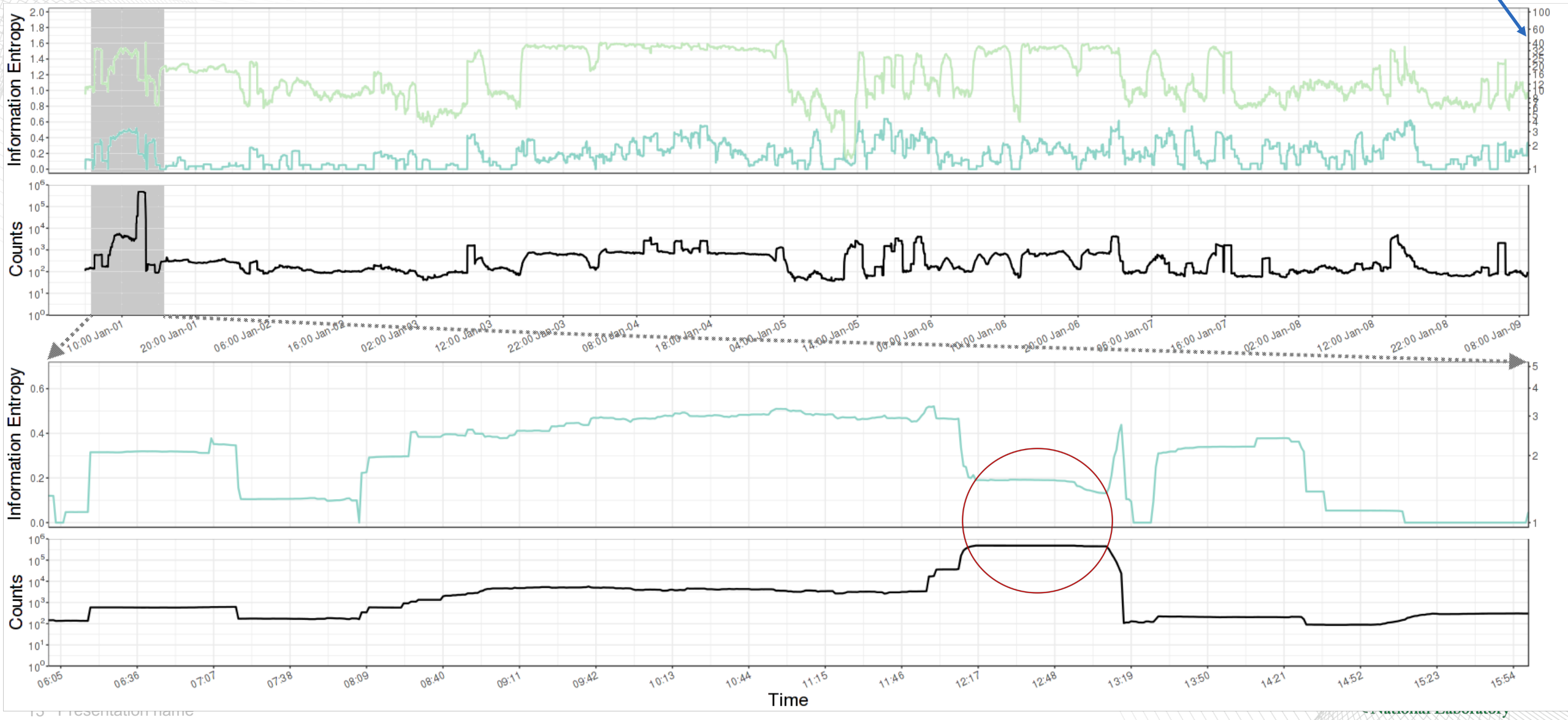
Nodal Map

$$X = 12 \times Cabinet_{Row} + 4 \times Chassis + Node$$
$$Y = 8 \times Cabinet_{Column} + Slot$$

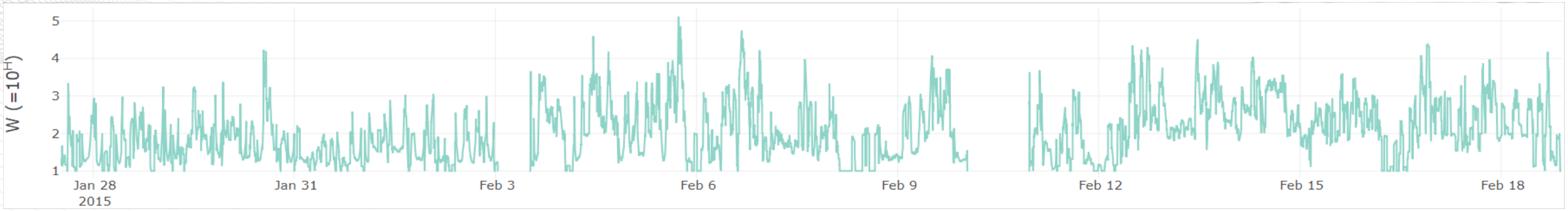


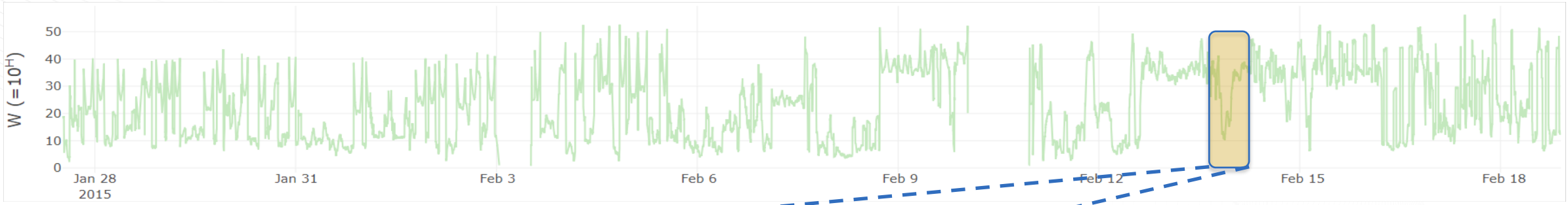
The nodal map for total events. The layout has dimensions of [300, 64] in pixel and each pixel represents a unique Titan node with its coordinates [X, Y] translated from the equation.

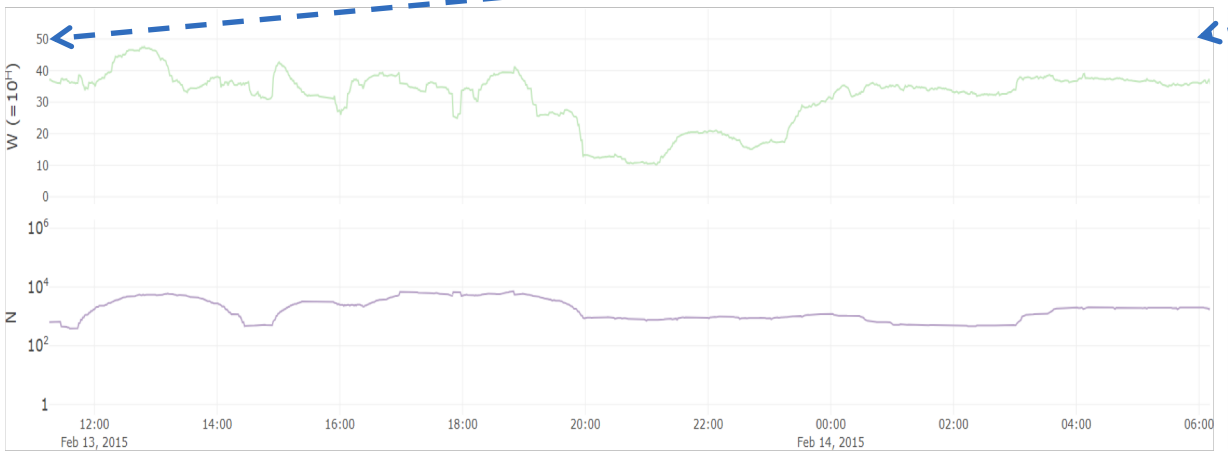
Source Type SIE Zoomed View

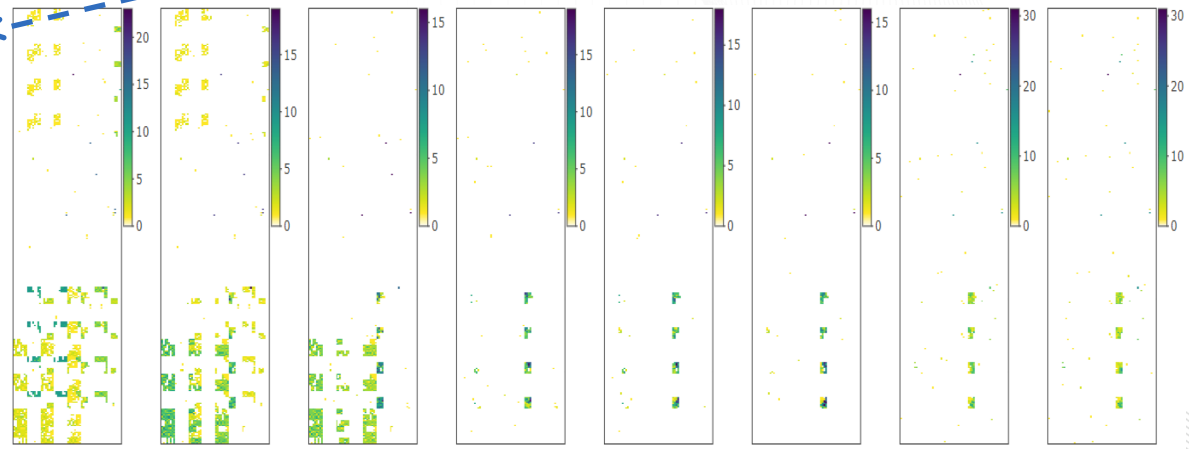


Nodal Map SIE Zoomed View









Titan nodal layouts between 18:30 and 22:00 at interval of 30 mins

Frequent Item set Mining

items	frequency
12, 11	56741
3, 12, 11	3813
22, 12, 11	18481
10, 11	2859
9, 12	2693
2, 9	8501

Frequent events for June 2016 (support = 0.01)

ID	Description
1	DVS Confusion
2	NVRM Xid
3	Machine Check Exception (MCE)
4	NVRM DBE
5	Unknown GPU Error (UGE)
6	Graphics Engine Error (GEE)
7	Kernel Panic
8	Out of Memory (OOM)
9	HWERR
10	Seg. Fault
11	Lustre
12	LNet
13	LNet Error
22	Lustre Error

Event Correlation

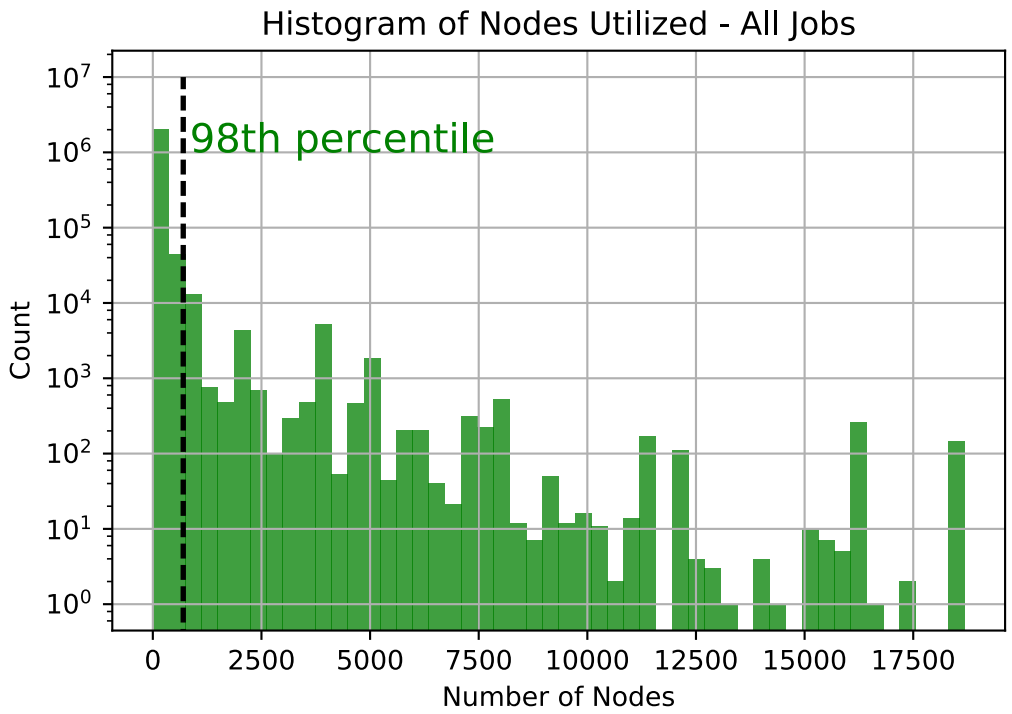
id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	22
1	1.0	-0.0	-0.0	-0.1	-0.0	-0.0	-0.0	-0.0	-0.0	0.0	-0.0	0.0	0.1	-0.0	-0.0
2	-0.0	1.0	0.1	-0.0	-0.0	0.1	-0.0	-0.0	0.0	-0.0	0.0	0.1	0.0	-0.0	-0.0
3	-0.0	0.1	1.0	-0.0	0.1	0.0	-0.1	-0.0	0.0	0.0	-0.1	-0.0	-0.0	-0.0	-0.1
4	-0.1	-0.0	-0.0	1.0	-0.0	0.0	0.0	0.1	-0.0	-0.0	0.1	0.1	0.0	-0.1	0.1
5	-0.0	-0.0	0.1	-0.0	1.0	-0.0	-0.0	-0.1	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
6	-0.	0.1	0.0	0.0	-0.0	1.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.1	-0.1	-0.0	-0.0
7	-0.0	-0.0	-0.1	0.0	-0.0	-0.0	1.0	-0.0	-0.0	-0.0	0.4	0.0	0.1	-0.0	0.7
8	-0.0	-0.0	-0.0	0.1	-0.1	-0.0	-0.0	1.0	-0.0	-0.0	-0.1	-0.0	0.0	-0.1	-0.0
9	-0.0	0.0	0.0	-0.0	-0.0	-0.0	-0.0	-0.0	1.0	-0.0	0.1	0.1	0.0	-0.0	0.0
10	0.0	-0.0	0.0	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0	1.0	-0.0	-0.0	0.0	-0.0	0.1
11	-0.0	0.0	-0.1	0.1	-0.0	-0.0	0.4	-0.1	0.1	-0.0	1.0	0.5	0.1	0.0	0.4
12	0.0	0.1	-0.0	0.1	-0.0	-0.1	0.0	-0.0	0.1	-0.0	0.5	1.0	0.3	-0.0	0.1
13	0.1	0.0	-0.0	0.0	-0.0	-0.1	0.1	0.0	0.0	0.0	0.1	0.3	1.0	-0.1	0.1
14	-0.0	-0.0	-0.0	-0.1	-0.0	-0.0	-0.0	-0.1	-0.0	-0.0	0.0	-0.0	-0.1	1.0	-0.0
22	-0.0	-0.0	-0.1	0.1	-0.0	-0.0	0.7	-0.0	0.0	0.1	0.4	0.1	0.1	-0.0	1.0

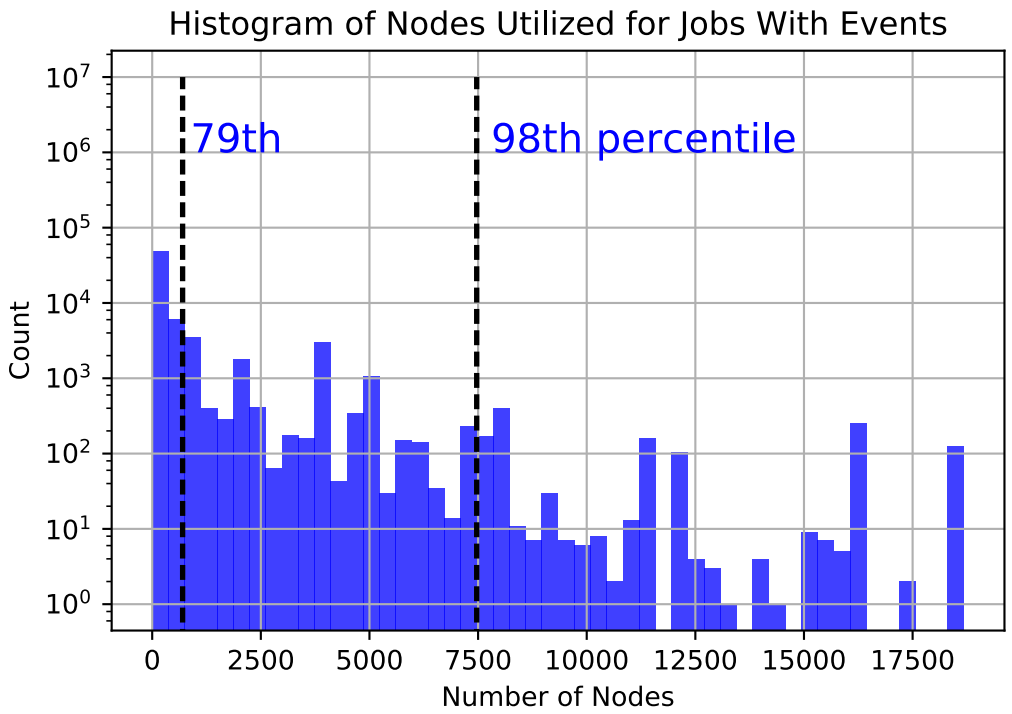
ID	Description
1	DVS Confusion
2	NVRM Xid
3	Machine Check Exception (MCE)
4	NVRM DBE
5	Unknown GPU Error (UGE)
6	Graphics Engine Error (GEE)
7	Kernel Panic
8	Out of Memory (OOM)
9	HWERR
10	Seg. Fault
11	Lustre
12	LNet
13	LNet Error
22	Lustre Error

Events and Application Runs

- Application run data of LogSCAN includes
  - Application name of each executed job,
  - The start/end date/time of each job,
  - The nodes on which each job was scheduled.
- In this case study, we analyze the characteristics of applications by correlating this information with events recorded on the compute nodes during their executions.
- This is done by conditional join of events and application data.

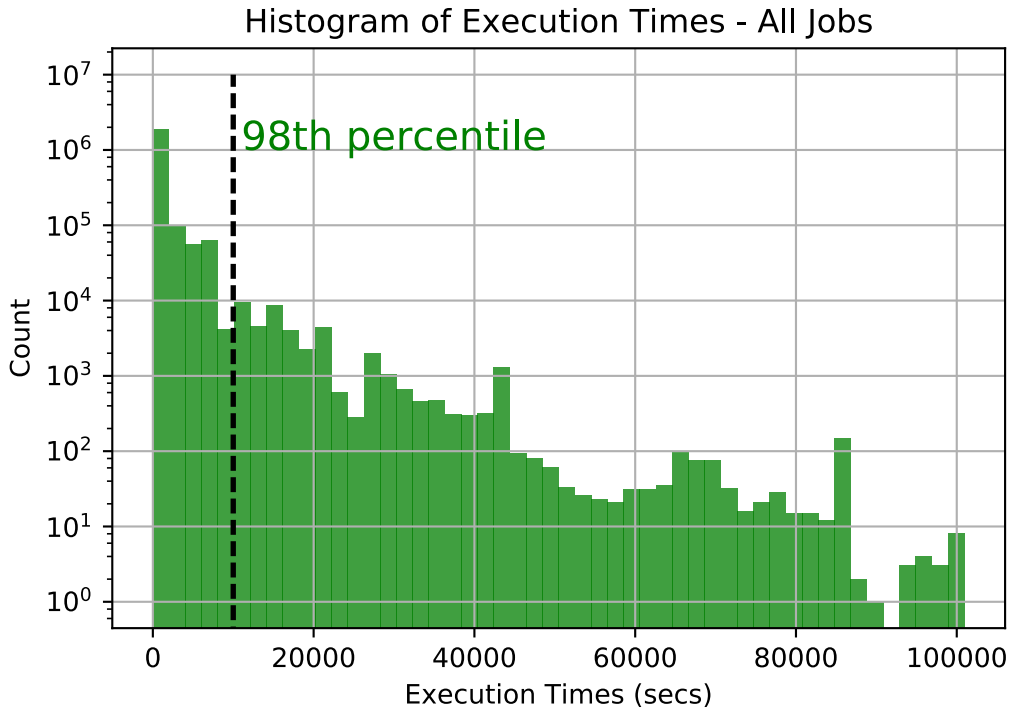
The histograms of number of nodes used (with logarithmic y-axis)

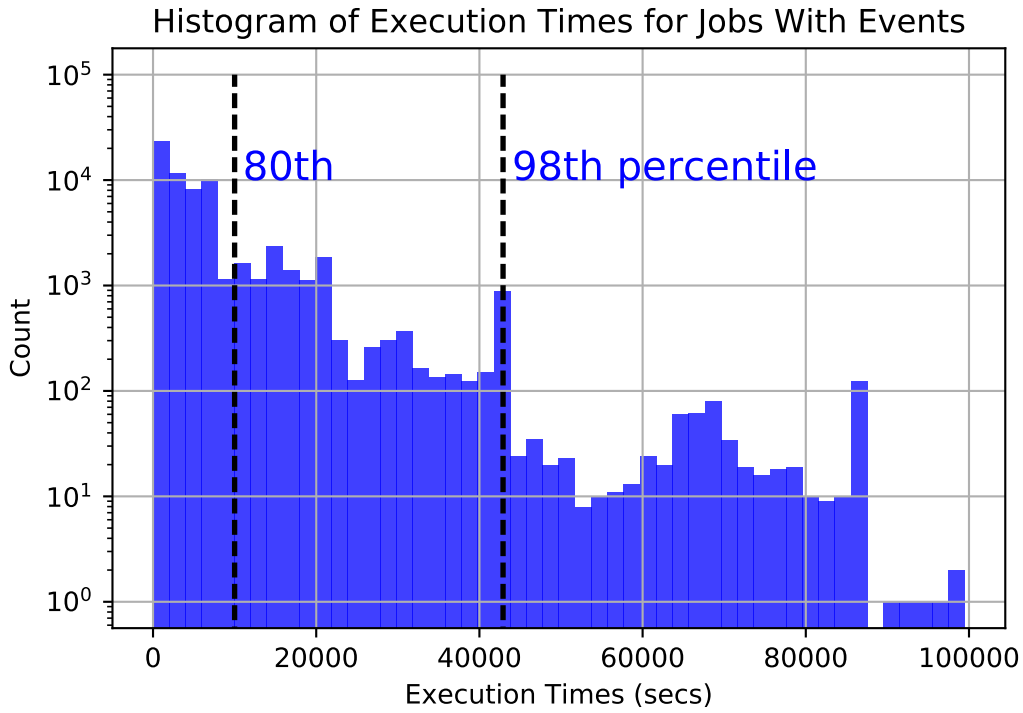




The histograms of number of nodes used (with a logarithmic y-axis) by all jobs and jobs which only had events. The comparison of the two plots demonstrates an increased likelihood of event occurrence for larger sized jobs.

Histograms of Execution Times (with Logarithmic Y-Axis)





The histograms of execution times (with a logarithmic y-axis) of all jobs and jobs which only had events. The comparison of the two plots demonstrates an increased likelihood of event occurrence for lengthier job executions.

Correlation Study per Event Group

1. All Events
2. Lustre related events (IDs 1, 11, 13 and 22) excluding Lustre network status event (ID 12),
3. Machine check exceptions (ID 3)
4. Hardware and software related events on processors excluding MCEs (IDs 6, 7 and 9),
5. Pure software issues on processors including segmentation faults and out of memory errors (IDs 8 and 10).
6. GPU events (IDs 2, 4 and 5),

Event Class	Execution Time	Job Size	Core-hours
All Events	0.249	0.129	0.310
Lustre Events	0.236	0.285	0.354
MCE Events	0.297	0.258	0.480
Processor HW/SW	0.440	0.523	0.599
Pure SW Events	0.272	0.548	0.570
GPU Events	0.476	0.526	0.614

Conclusion

- Three use cases of performing analytics using LogSCAN are presented demonstrating:
  - All examples cases utilize scalable data model for efficient retrieval from various perspectives.
  - Analytic results are also interesting, but require further study for verification and interpretation.
  - Two use cases have been submitted as individual manuscripts.
- As exascale system coming available, LogSCAN will play an important role in understanding system health, status, and utilization.

Acknowledgement

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, program manager Lucy Nowell, under contract number DE-AC05-00OR22725. This work was supported by the Compute and Data Environment for Science (CADES) facility and the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is managed by UT Battelle, LLC for the U.S. DOE (under the contract No. DE-AC05-00OR22725).

Questions?

Byung Hoon Park
parkbh@ornl.gov

OAK RIDGE National Laboratory