

Deriving Knowledge from Data Disarray with ScrubJay

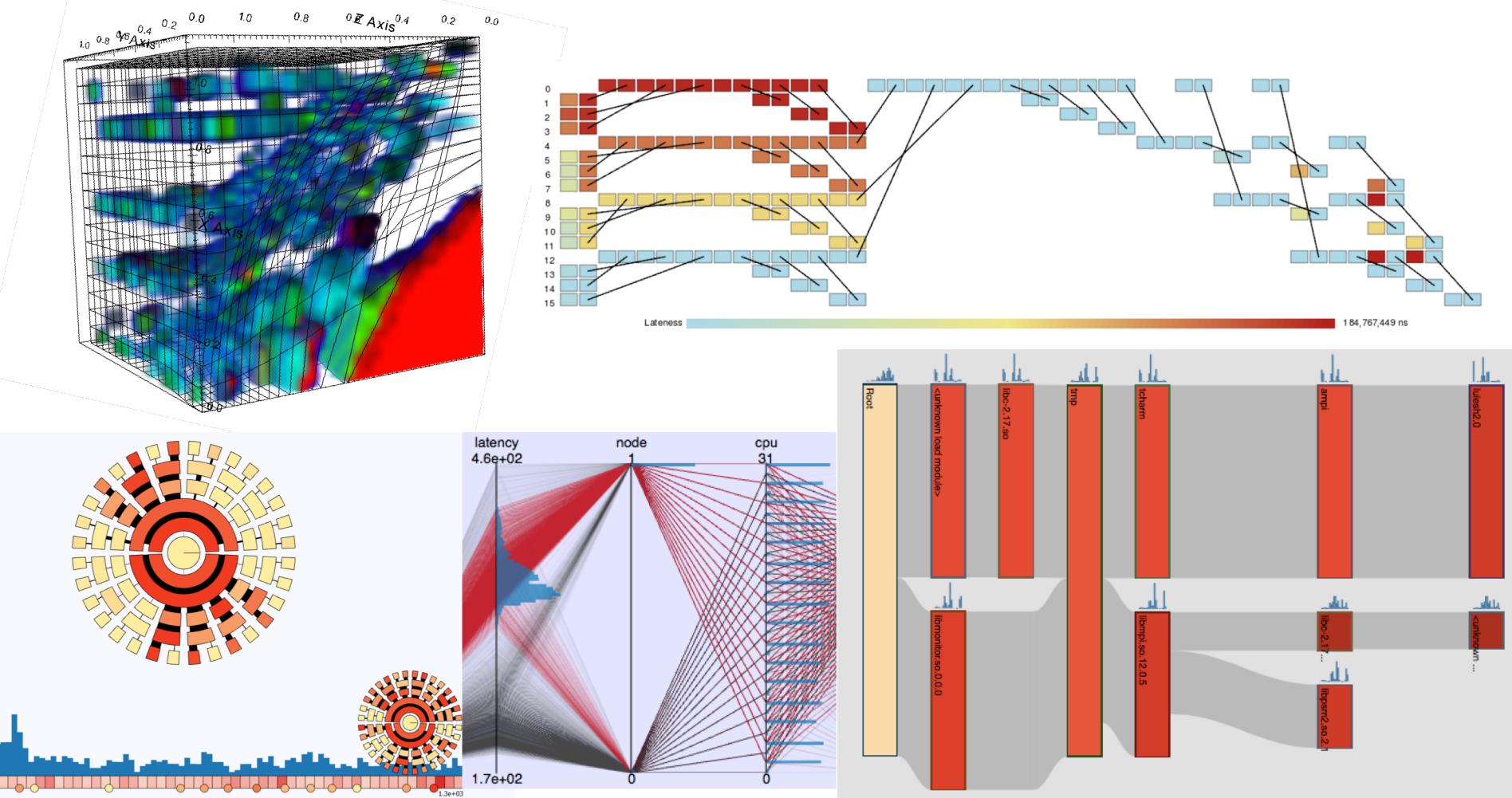
Supercomputing 2017

Alfredo Gimenez, Todd Gamblin, Abhinav Bhatele,
Chad Wood, Kathleen Shoga, Aniruddha Marathe,
Peer-Timo Bremer, Bernd Hamann, and Martin Schulz

Wednesday, Nov 15, 2017

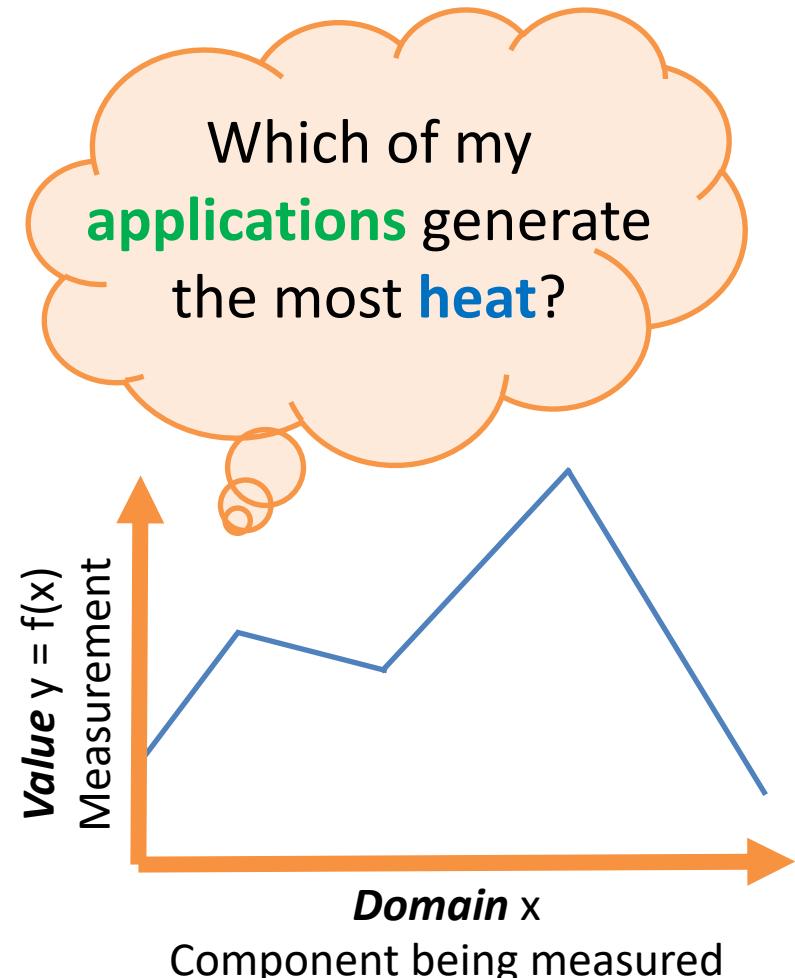
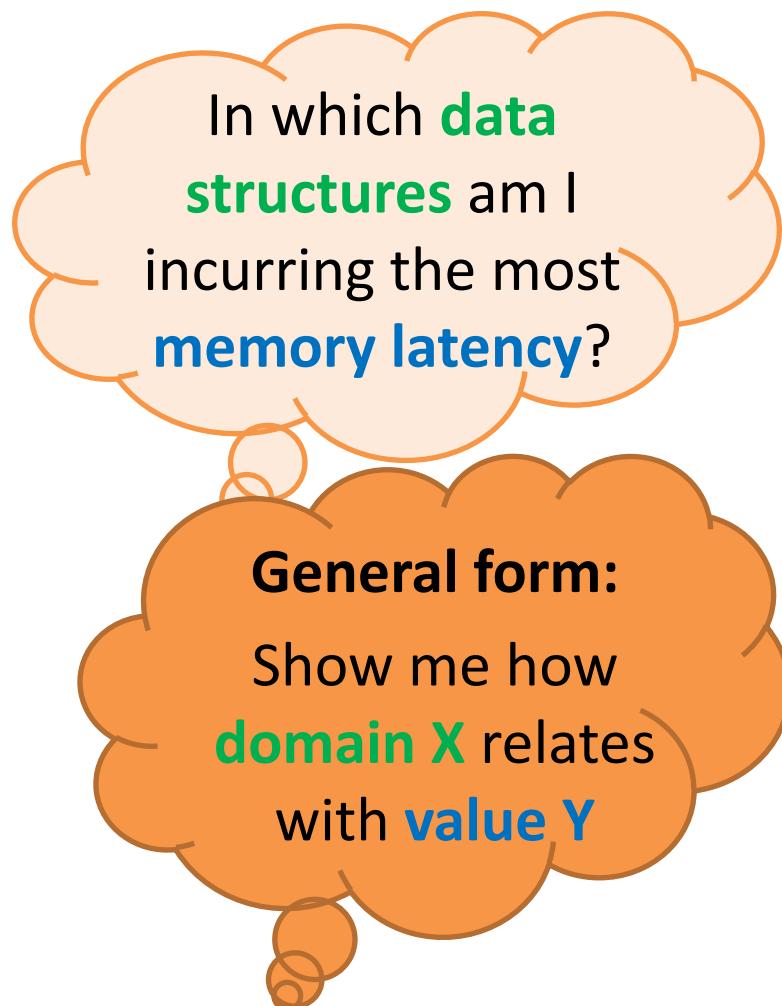


Performance Data Visualization



Full survey here: <http://hdc.cs.arizona.edu/mamba/home/~kisaacs/STAR/>

The Data Visualization/Analysis Query



The Goal:

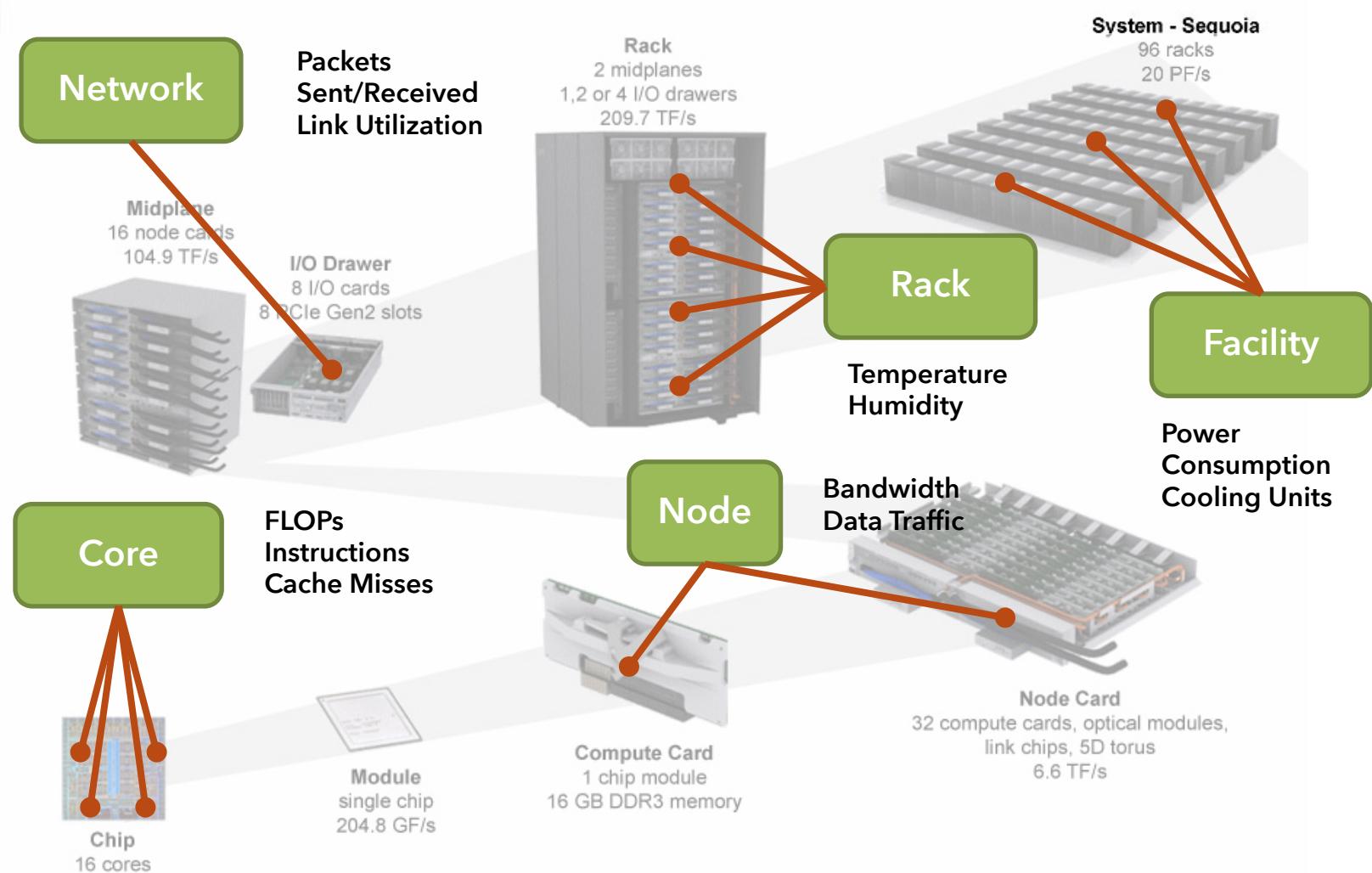
QUERY:

Values(Heat, FLOPs)
over
Domains(Job Names, Node IDs)

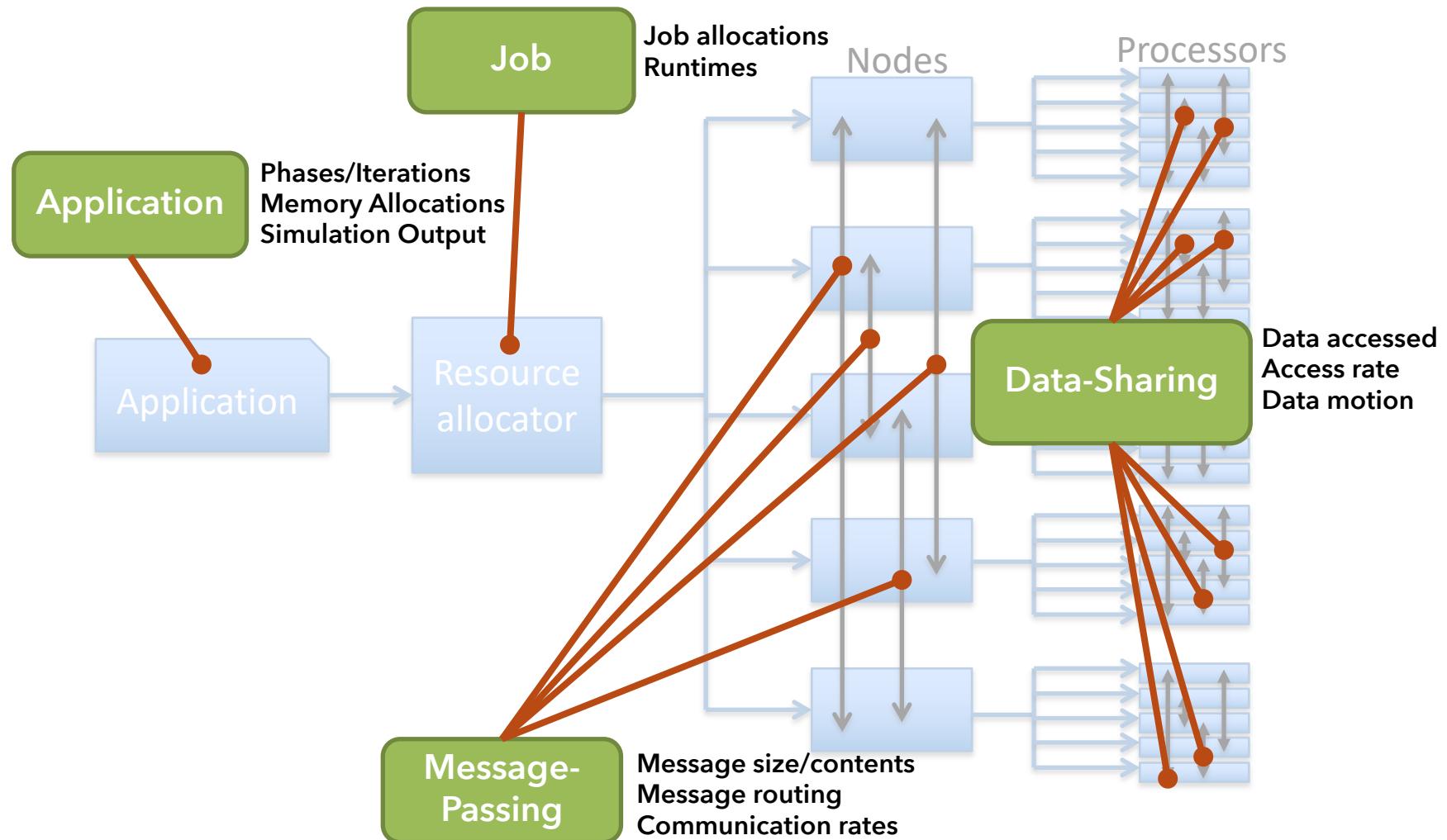
RESPONSE:

Job Name (Resource Scheduler)	Node IDs (Resource Scheduler)	Heat (Facility Sensors)	FLOPs (Core Counters)
AMG	256	76.5	8463921
miranda	64	61.2	33267600
...

Hardware Domains



Software Domains



Heterogeneous Data Disarray - Formats

- Specialized formats
 - Tightly coupled to analysis/visualization tools
- Standard text formats
 - CSV, JSON
- Databases
 - SQL, NoSQL variants
 - REST APIs
- Unstructured formats



Heterogeneous Data Disarray – Disparate Domains

Time	FLOP Counter Value
10:00	6453
10:02	786
10:04	244556



Time	Temperature
10:01	55.6
10:03	58.2
10:05	65.1

Time	Temperature	FLOP Counter Value
10:00	UNDEFINED	6453
10:01	55.6	UNDEFINED
10:02	UNDEFINED	786
10:03	58.2	UNDEFINED
10:04	UNDEFINED	244556
10:05	65.1	UNDEFINED

Heterogeneous Data Disarray – Disparate Domains

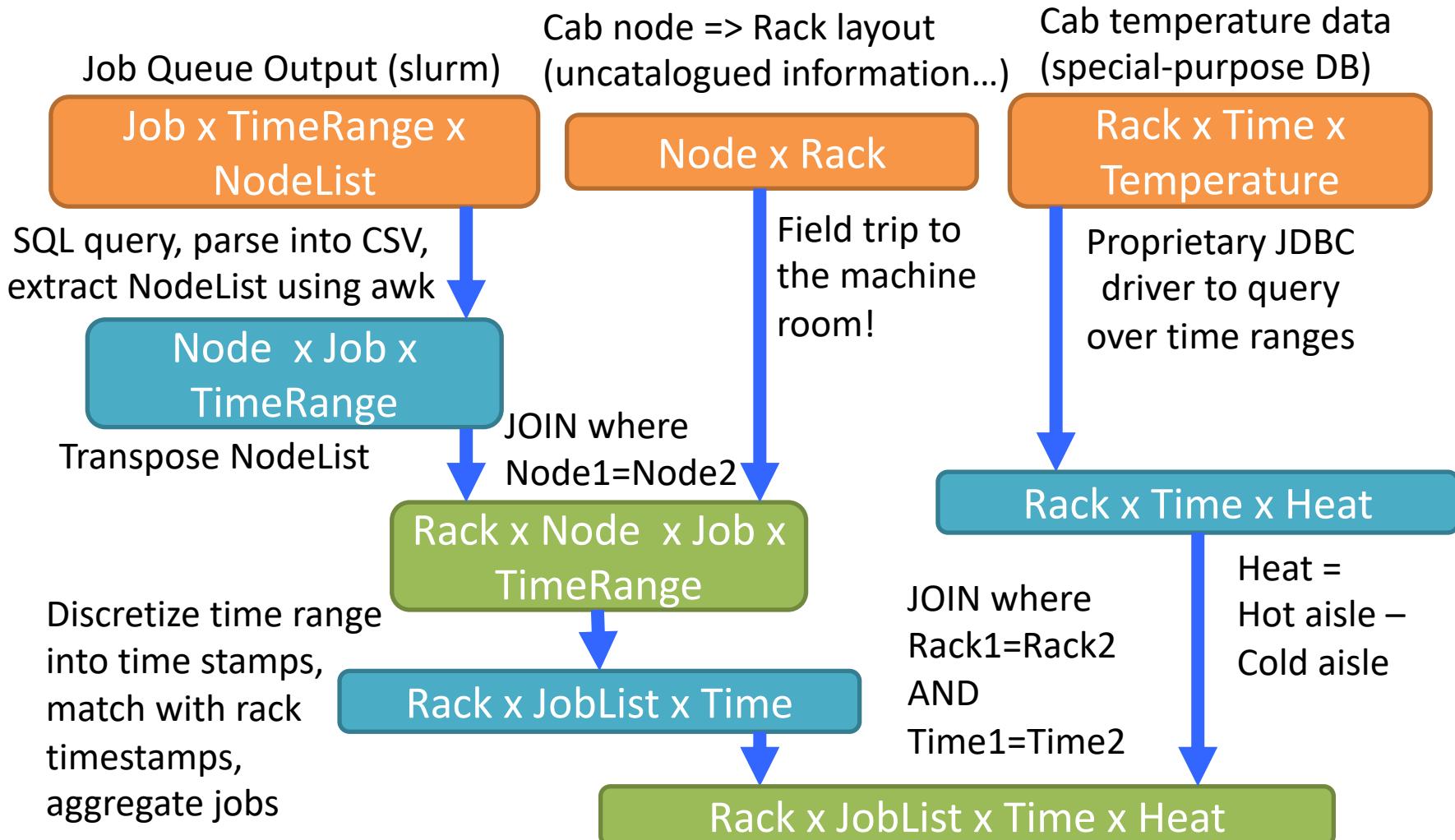
T	FLOP Counter Value
54453	6453
74453	786
84453	244556



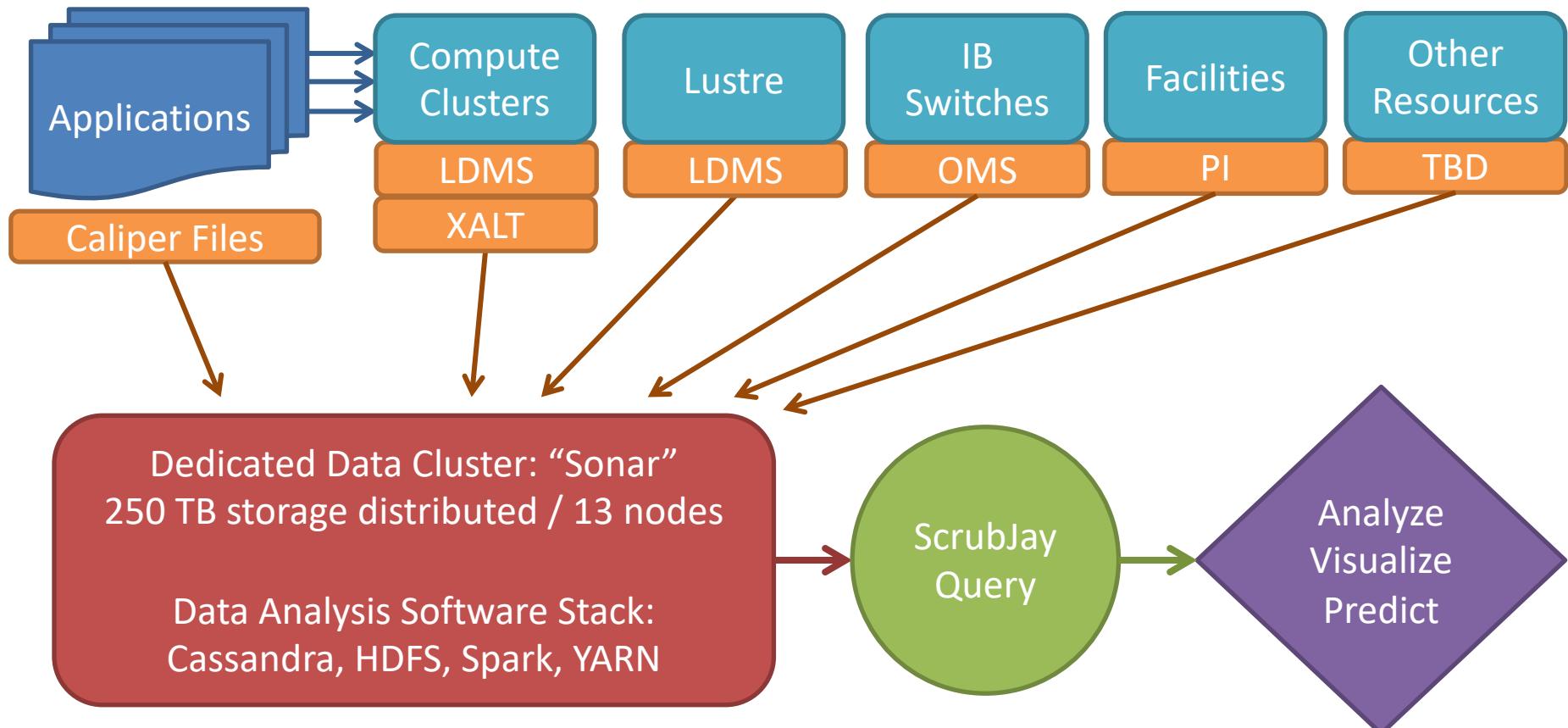
Time	Temperature
10:01	55.6
10:03	58.2
10:05	65.1

JOIN undefined!!

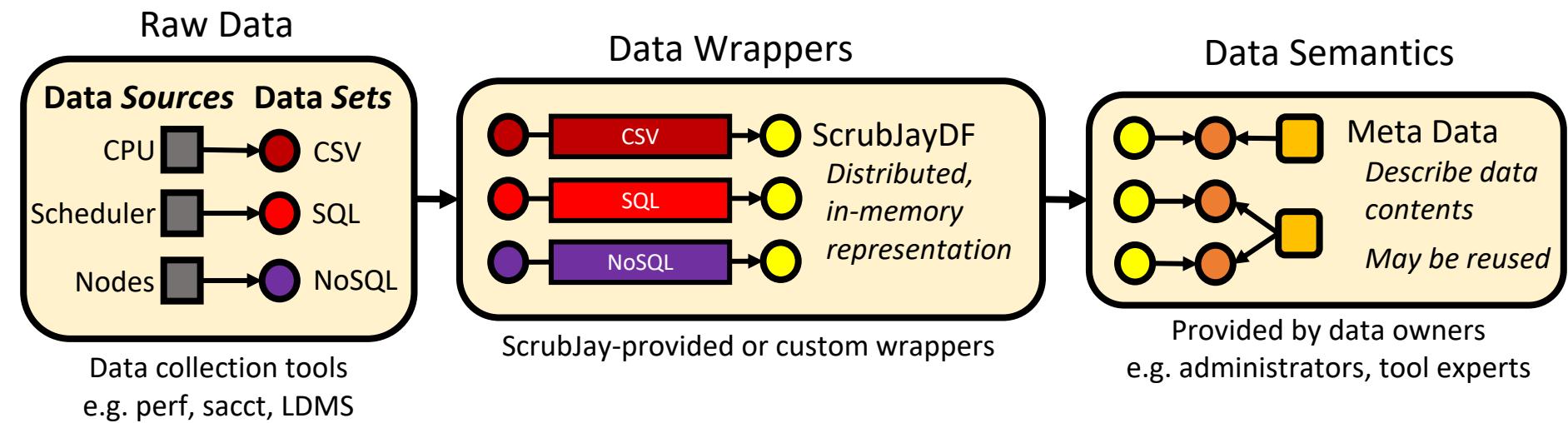
Manually Integrating Data



Sonar: Data Analysis Cluster for HPC Performance at LLNL



Parsing Heterogeneous Datasets



Data Semantics

Domains or Values

Units

Data dimensions may be:

Unordered

Ordered (discrete/continuous)

HPC Example: Node ID

Unordered (cab32 < cab65) ???

Discrete Halfway between cab32 and cab65 ???

HPC Example: Time

Ordered 10:00 < 11:00 == TRUE

Continuous Halfway between 10:00 and 11:00 == 10:30

Unordered

Tom, Bob, Alice

Ordered, Discrete



Ordered, Continuous



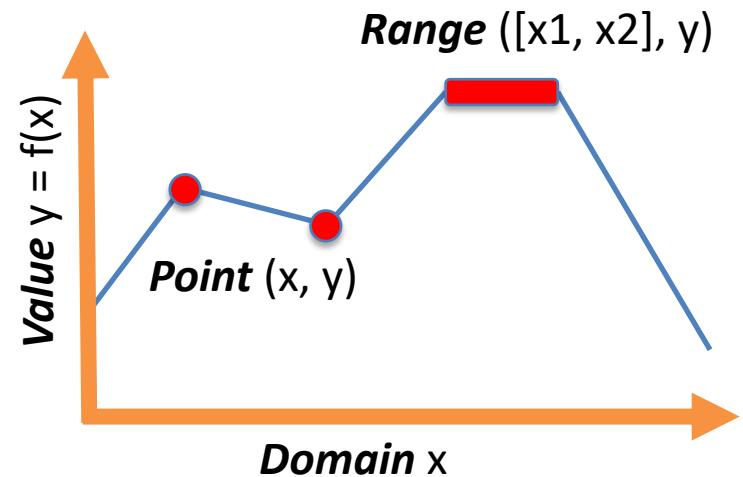
Discretization

Point

A single point of reference

Range (Ordered)

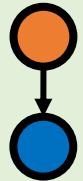
All points that lie between two points of an ordered dimension



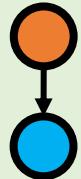
Rules for Inferring Data + Relationships

Data Derivations

Transformations

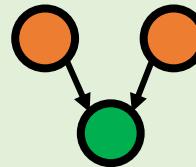


Transpose
Column

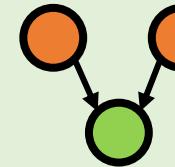


Derive
New Column

Combinations



Natural
Join

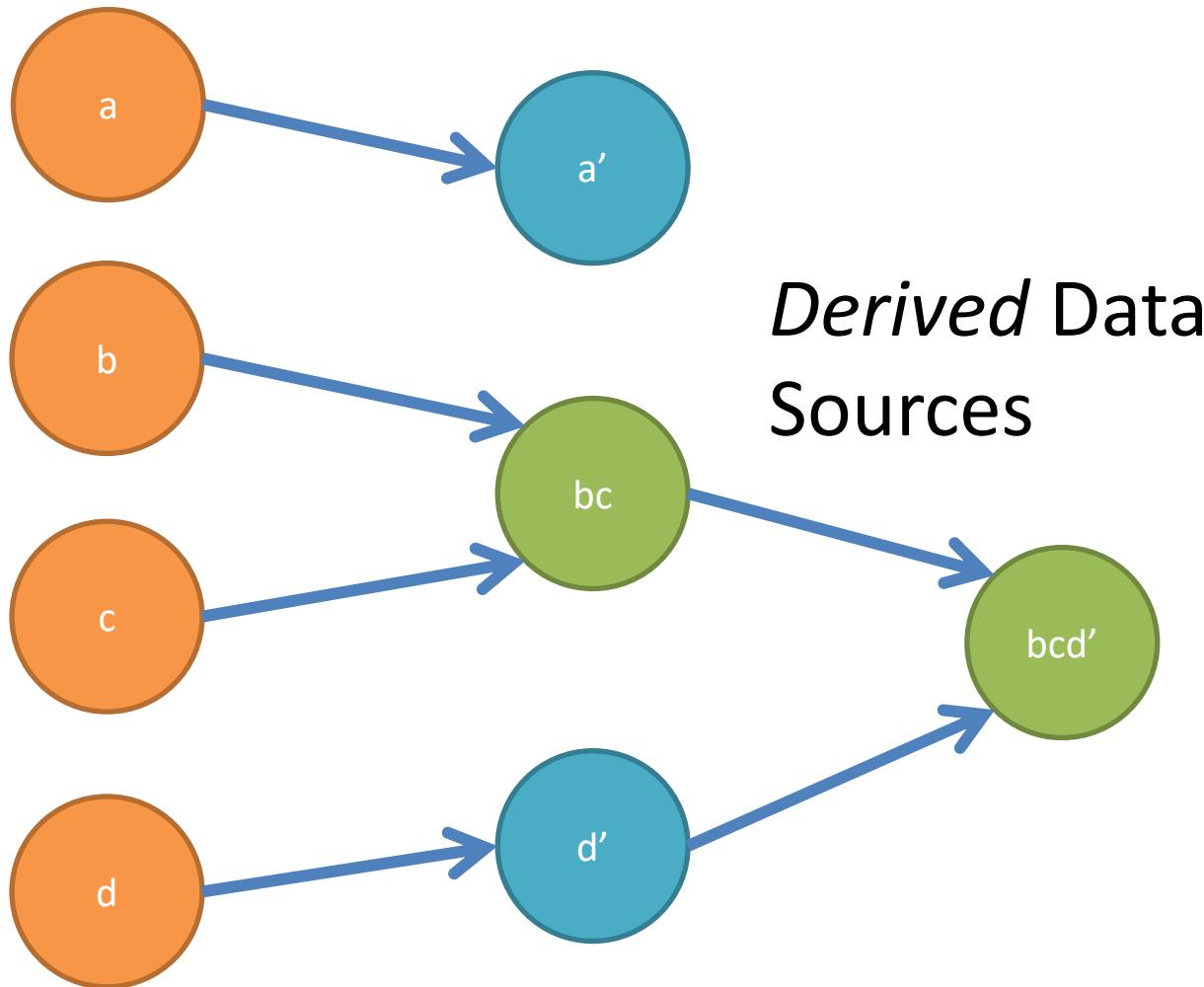


Interpolative
Join

ScrubJay-provided + domain-specific derivations provided by system experts

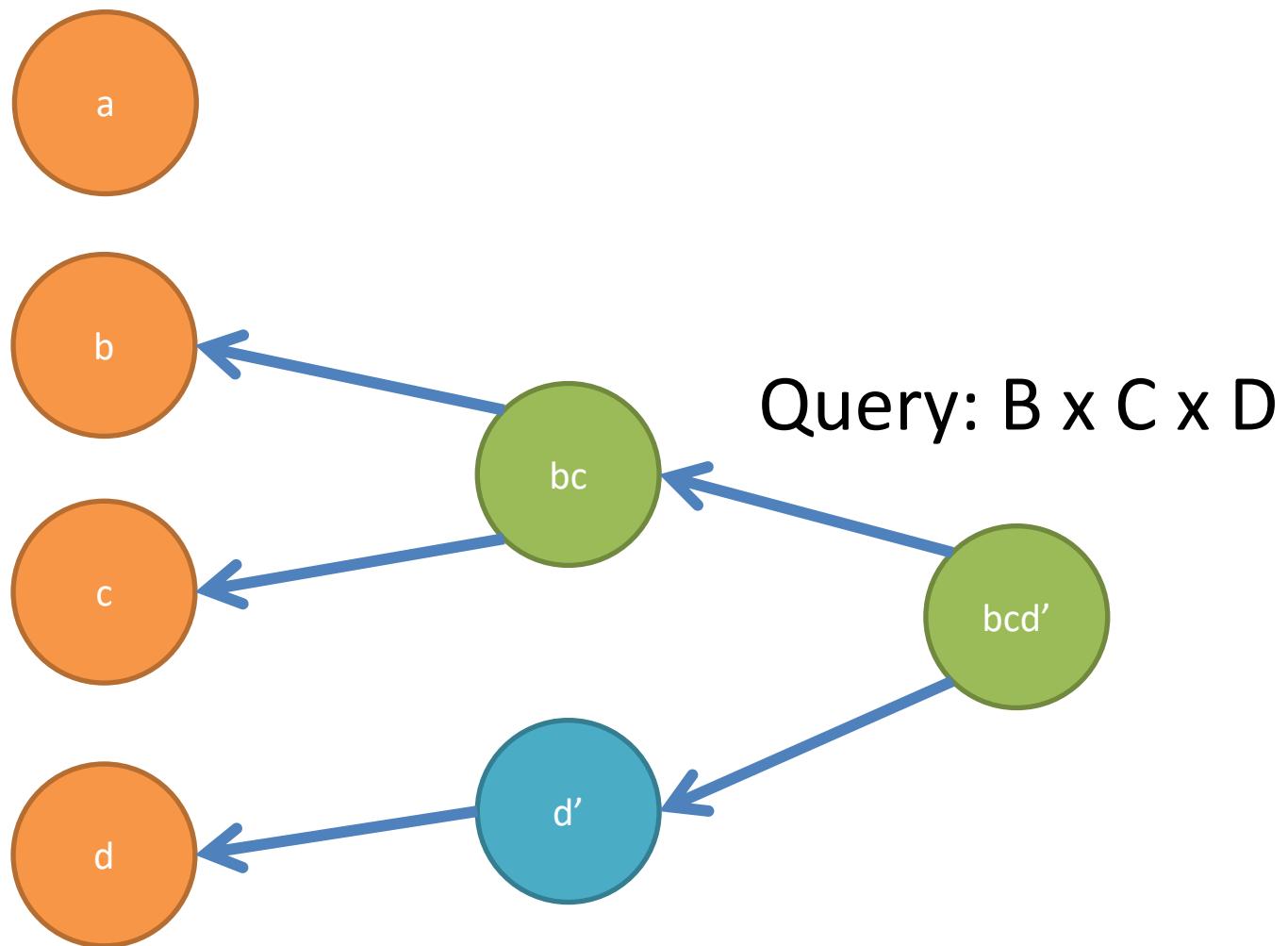
“Derivation” Engine

Original
Data
Sources

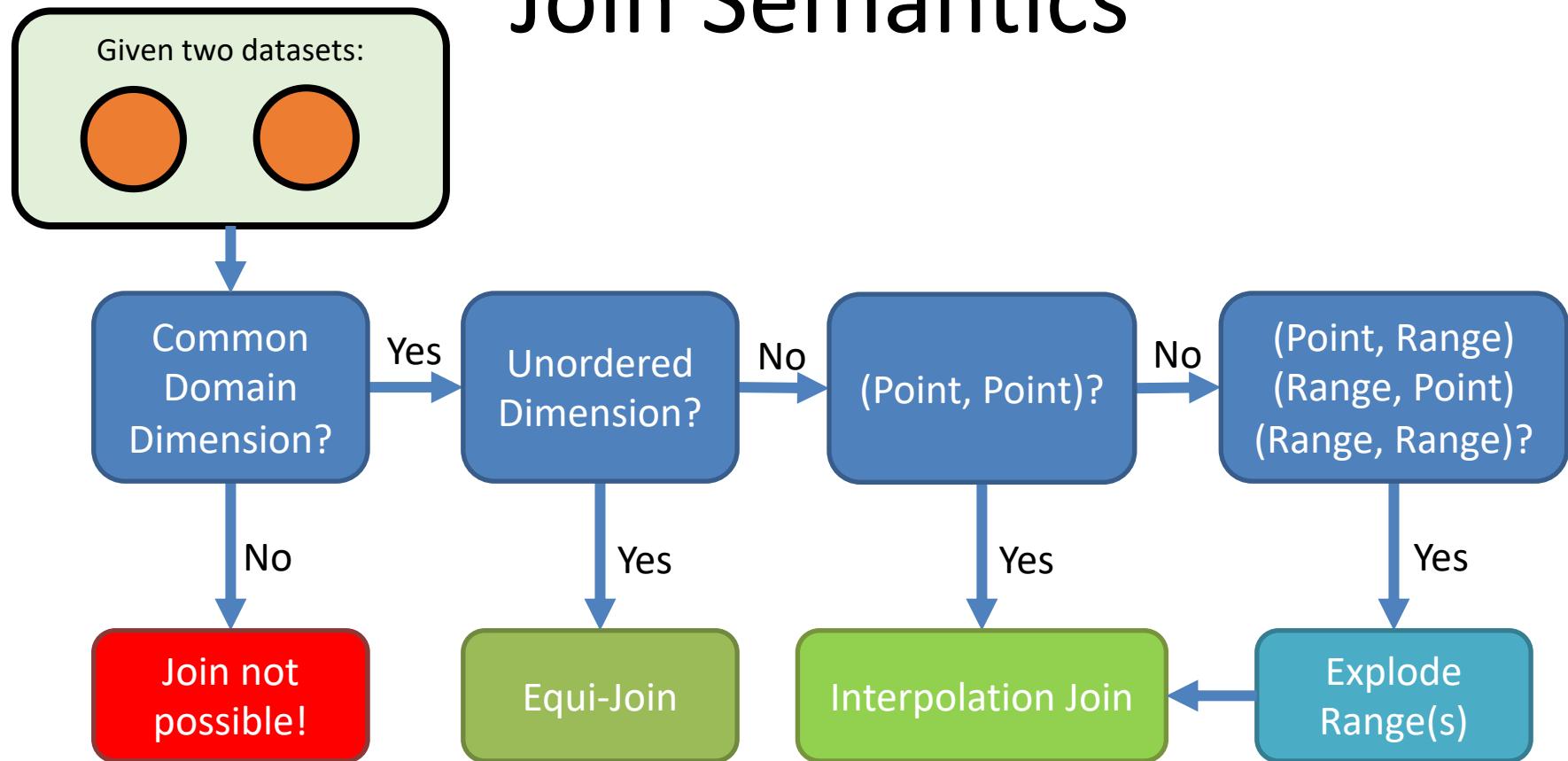


“Derivation” Engine

Original
Data
Sources



Join Semantics



Queries!

ScrubJay Query API

Query: ?

A set of data sources and measurements of interest

e.g. CPU Instructions + Rack Temperatures

Single entrypoint for performance analysts to request HPC performance data

In Action

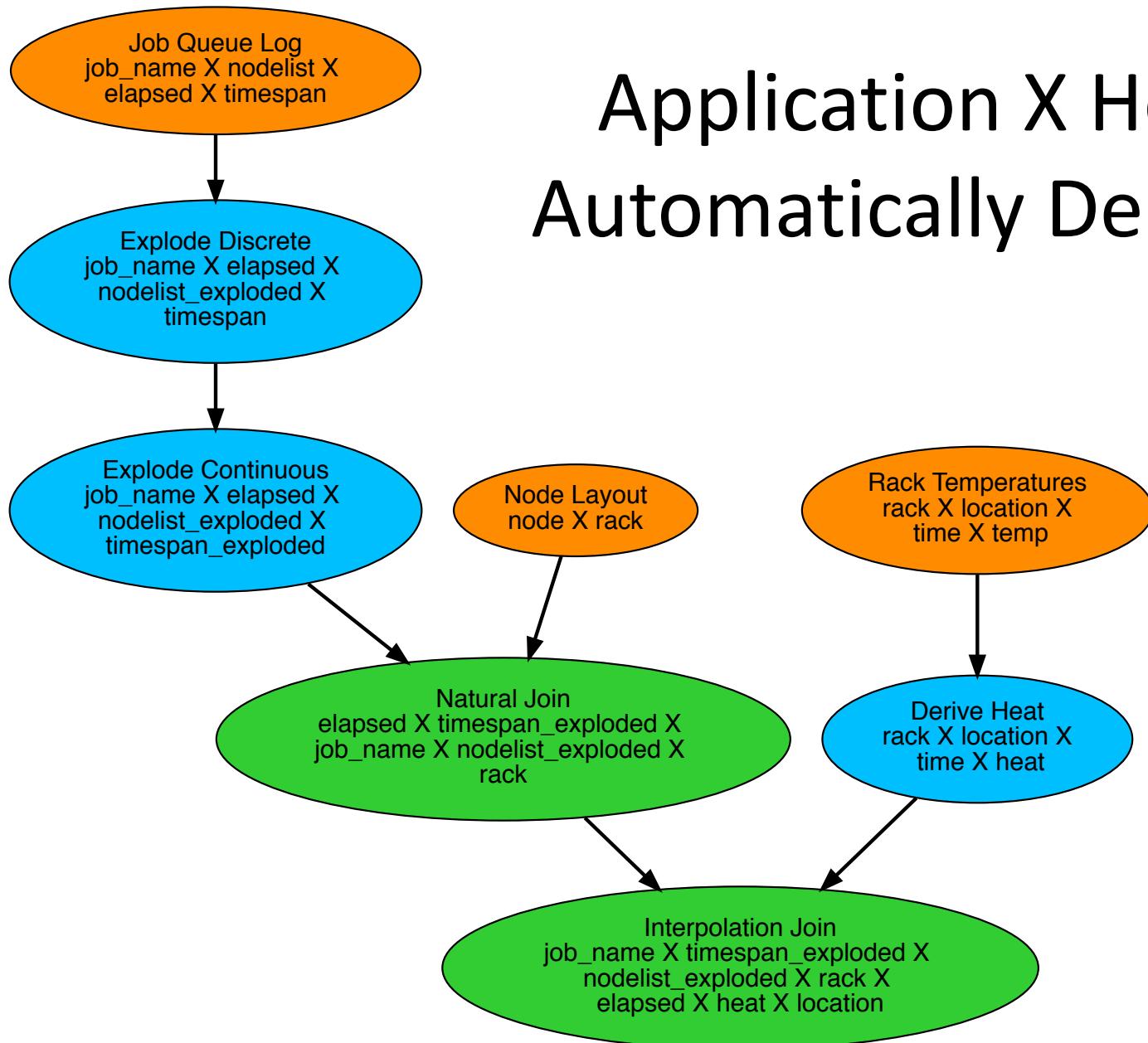
```
import scrubjay._
```

```
val dataSpace: DataSpace =  
  DataSpace.fromJsonFile(jobAnalysisDataSpaceFilename)
```

```
val queryTarget = ScrubJaySchema(Array(  
  ScrubJayField(domain = true, dimension = "job"),  
  ScrubJayField(domain = false, dimension = "heat")  
))
```

```
val query = Query(dataSpace, queryTarget)
```

```
lazy val solutions = query.solutions
```



Application X Heat

Automatically Derived

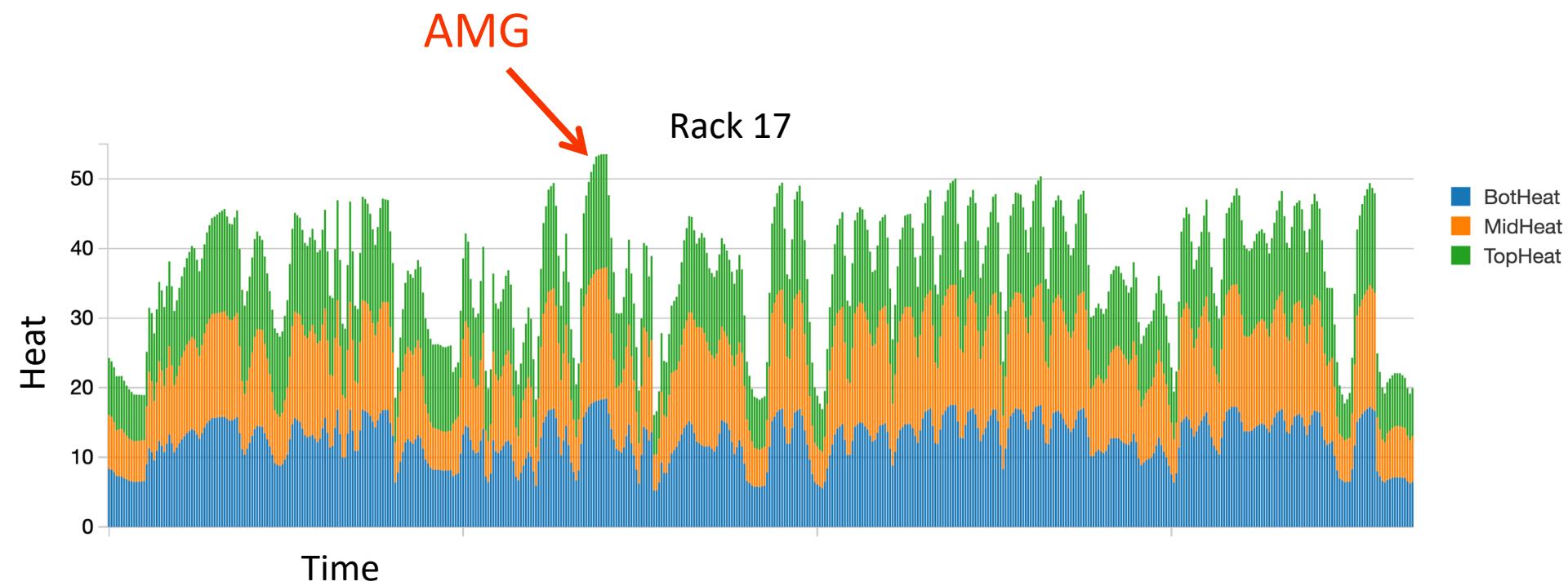
Rack X Heat Results

Rack x Joblist x Time x Heat

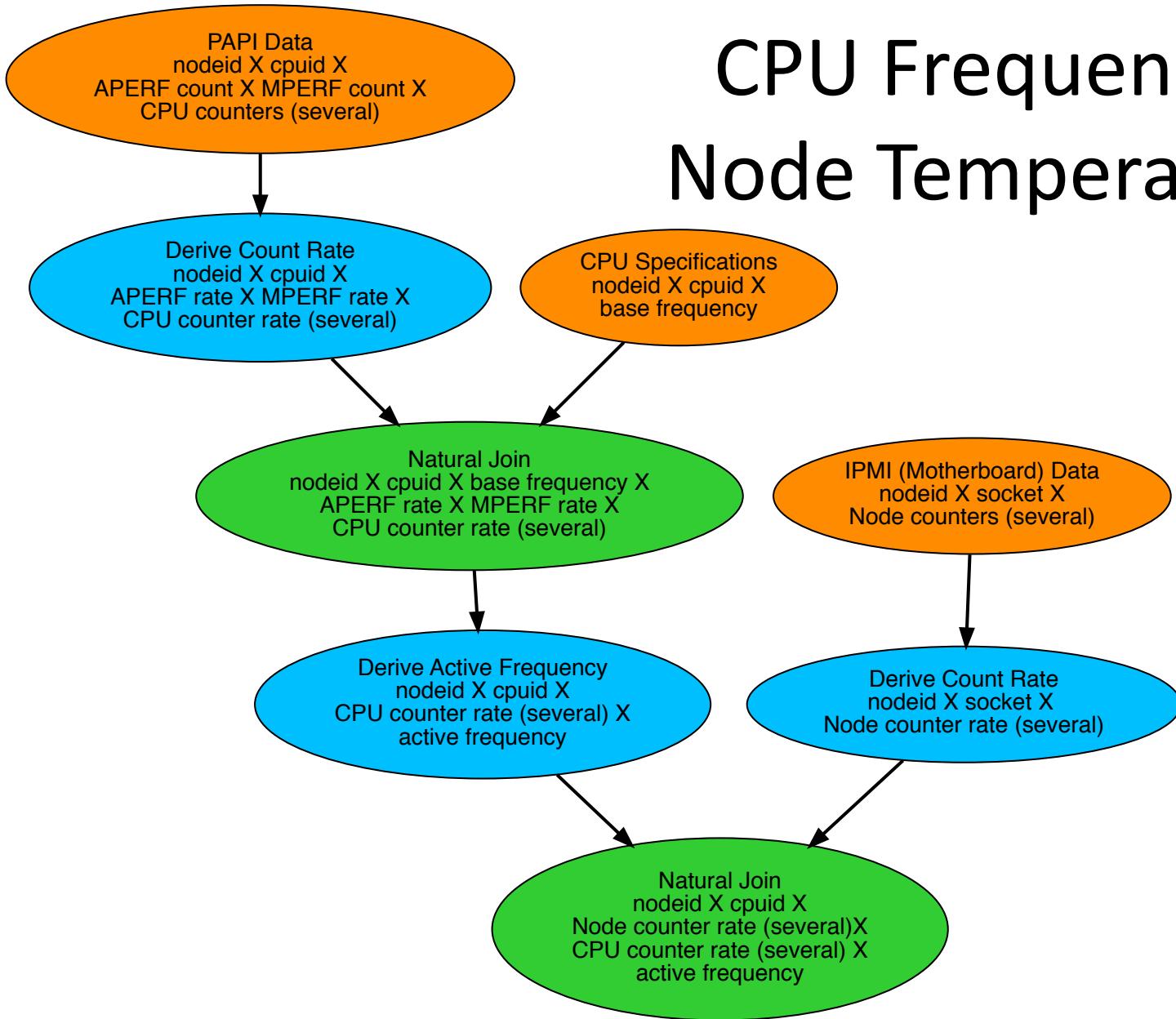
Rack	Time	TotalHeat	JobList
17	2015-08-05T22:44:00.000+0000	53.5423584	[('amg', 70)]
17	2015-08-05T22:42:00.000+0000	53.52637481999999	[('amg', 70)]
17	2015-08-05T22:40:00.000+0000	53.519630439999986	[('amg', 70)]
17	2015-08-05T22:38:00.000+0000	53.395622259999996	[('amg', 70)]
17	2015-08-05T22:36:00.000+0000	53.21107483	[('amg', 70)]
17	2015-08-05T22:34:00.000+0000	52.10317992	[('amg', 70)]
8	2015-08-05T22:42:00.000+0000	51.37504958999998	[('amg', 65)]
8	2015-08-05T22:40:00.000+0000	51.30201721	[('amg', 65)]
8	2015-08-05T22:44:00.000+0000	51.2350502	[('amg', 65)]
8	2015-08-05T22:38:00.000+0000	51.06603241	[('amg', 65)]
17	2015-08-05T22:32:00.000+0000	50.995285030000005	[('amg', 70)]
8	2015-08-05T22:36:00.000+0000	50.86448288	[('amg', 65)]
17	2015-08-06T06:46:00.000+0000	50.36058043999999	[('runjobs-c', 68)]
8	2015-08-05T22:34:00.000+0000	50.26057815	[('amg', 65)]
17	2015-08-06T12:26:00.000+0000	50.12788390000003	[('runjobs-c', 70)]

Rack 17,
70 nodes
running AMG

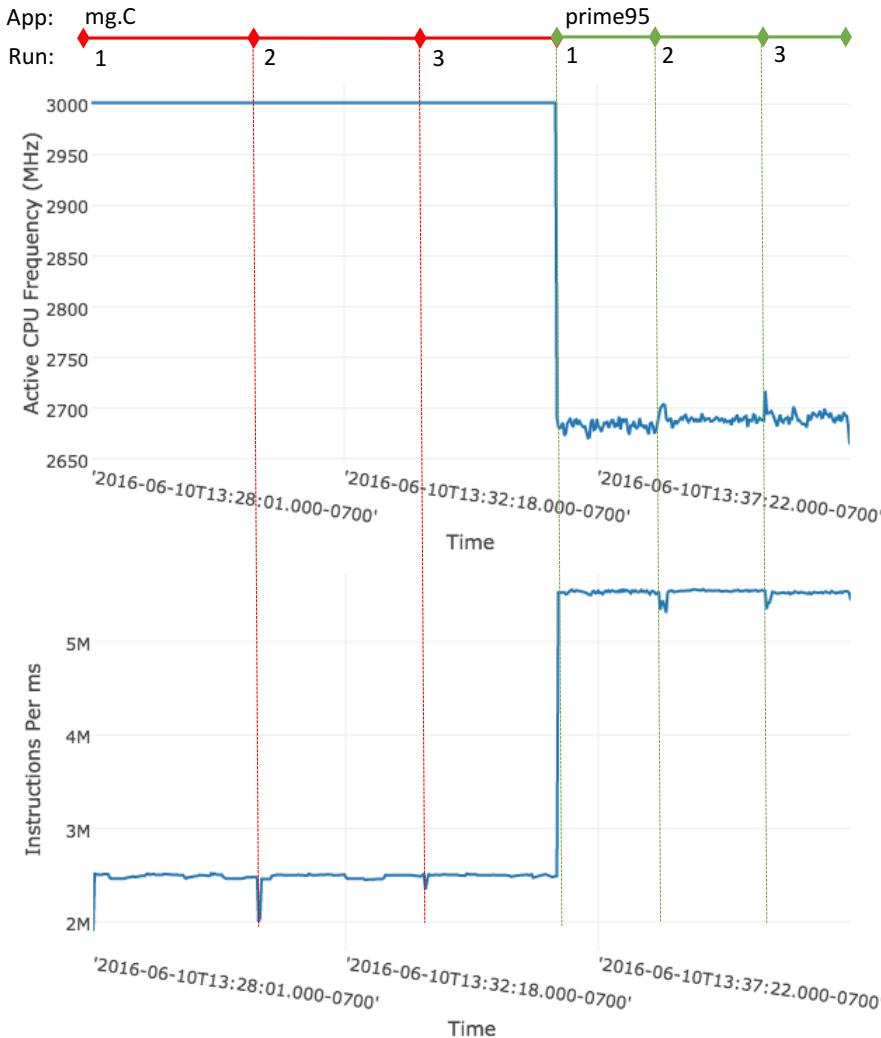
Rack X Heat Results



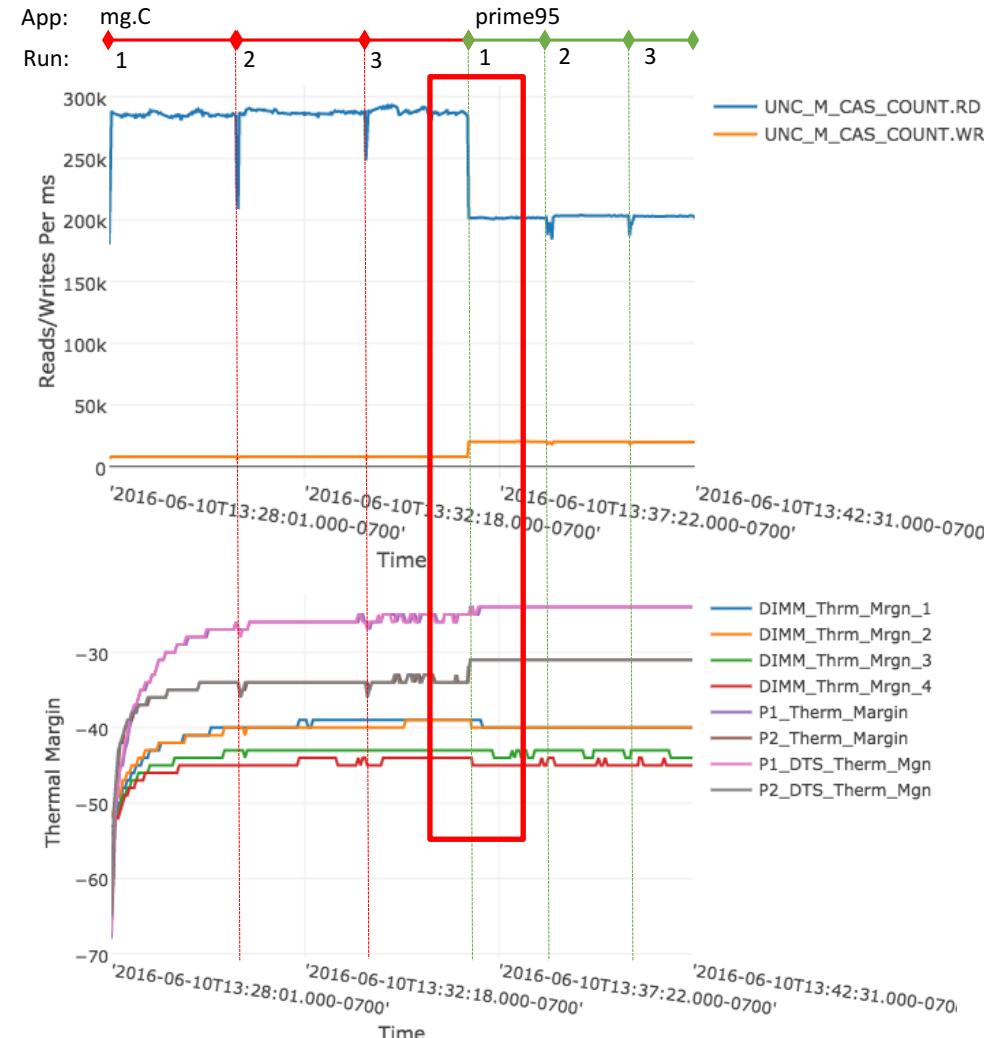
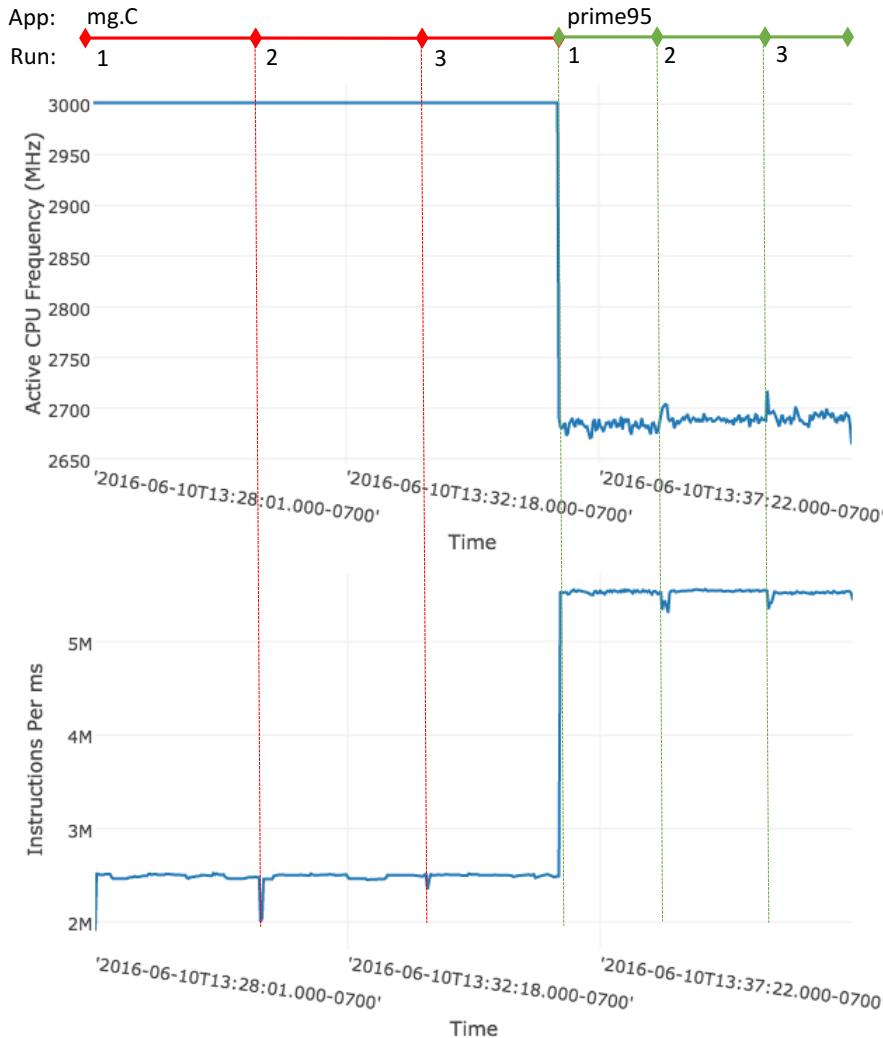
CPU Frequency X Node Temperatures



CPU Frequency X Node Metrics



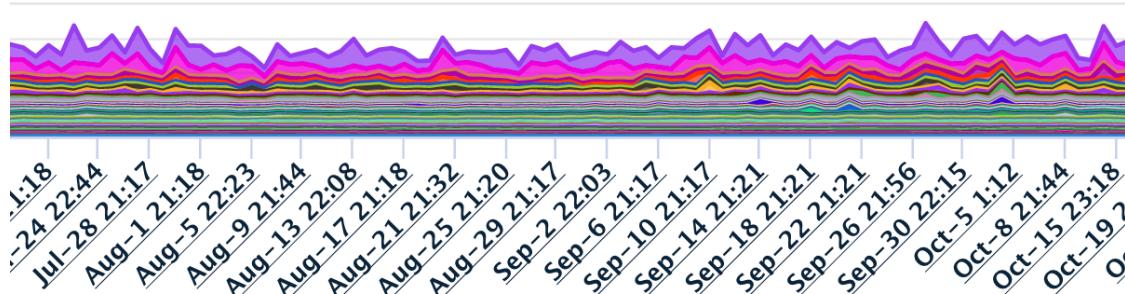
CPU Frequency X Node Metrics



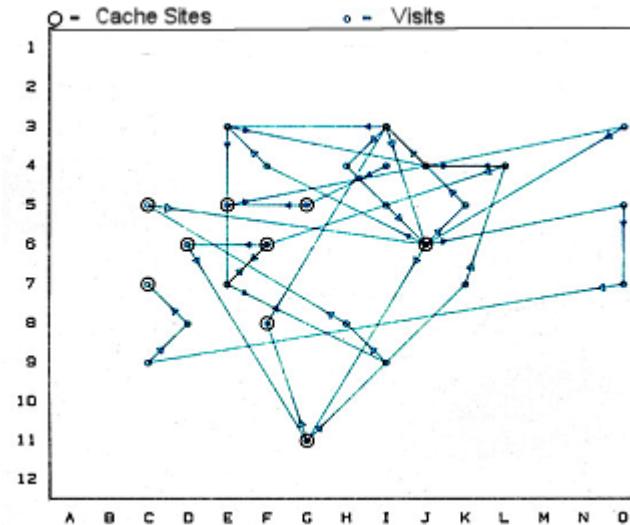
Sonar Projects that use ScrubJay

- Open source release coming soon
- Shareable performance analysis in JupyterHub
- Where should we invest on future hardware?
 - Long-term monitoring, cluster/job throughput analysis
- Application-level to facility-level analysis:
 - Application data from Caliper <https://github.com/llnl/Caliper/>
- SPOT: Tracking code performance over time

Aggregate Test Suite Performance: read restart silo



Scrub Jays



- **Left:** Scrub jay collecting peanuts for storage in food caches
- **Right:** Recorded flight path and cache collection of a scrub jay in an experiment [1]

[1] Russell P. Balda & Alan C. Kamil, “Linking Life Zones, Life History Traits, Ecology, and Spatial Cognition in Four Allopatric Southwestern Seed Caching Corvids”, <http://www.pigeon.psy.tufts.edu/asc/Balda/>

Acknowledgements

- Sonar Team
 - Alfredo Gimenez, Kathleen Shoga, Todd Gamblin, Matt Legendre, David Poliakoff, Tim Meier, Danielle Sikich, Thomas Mendoza
- Caliper
 - <https://github.com/llnl/Caliper/>
- Performance Visualization
 - http://hdc.cs.arizona.edu/mamba_home/~kisaacs/STAR/
- Workshop on Visual Performance Analysis (VPA 17)
 - <https://vpa17.github.io/>
- UC Davis and Lawrence Livermore National Laboratory

Supplemental

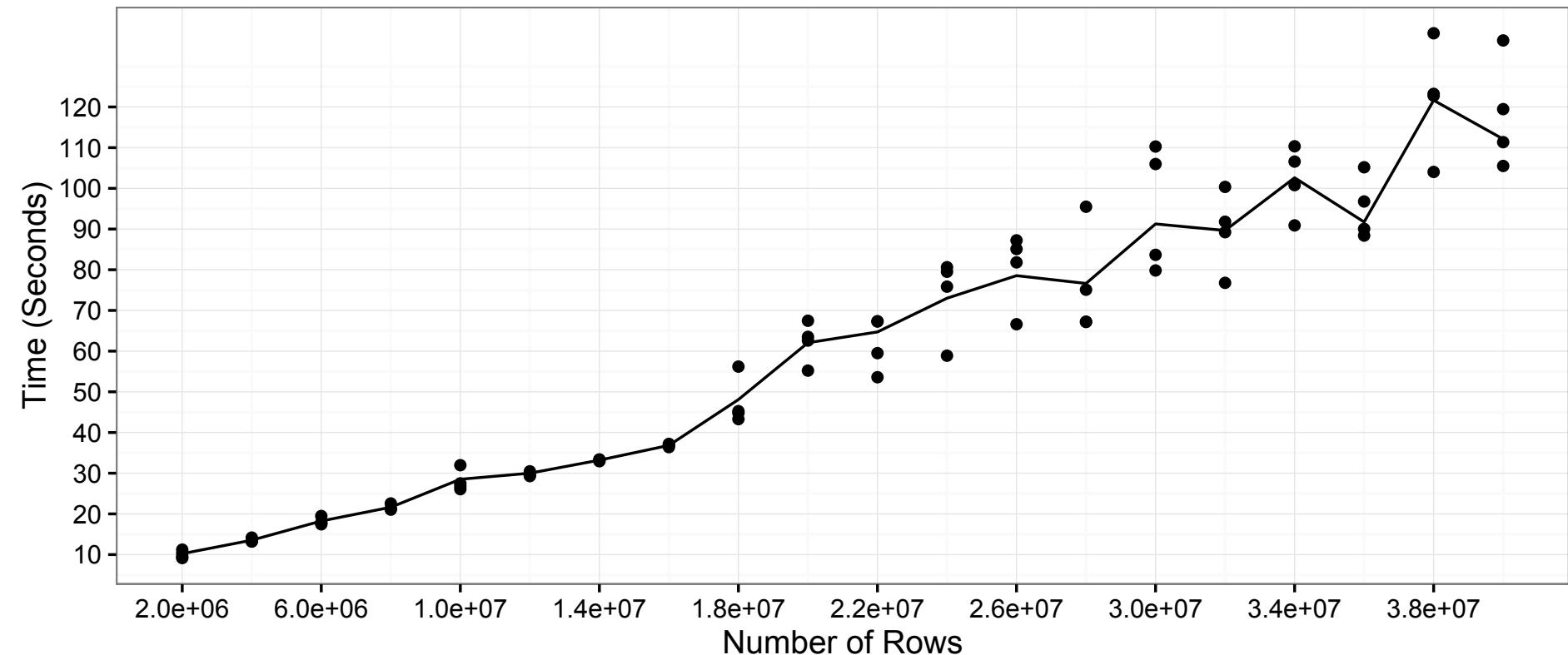
How much meta-information do I need to provide?

```
{  
  "type" : "CSVDatasetID",  
  "csvFileName" : "jobQueue.csv",  
  "options" : {  
    "header" : "true",  
    "delimiter" : "|"  
  },  
  "scrubJaySchema": {  
    "fields": [  
      {  
        "name" : "jobid",  
        "domain" : true,  
        "dimension" : "job",  
        "units" : {  
          "name" : "identifier",  
          "elementType" : "POINT"  
        }  
      },  
    ],  
  },  
}
```

```
{  
  "name" : "nodelist",  
  "dimension" : "node",  
  "domain" : true,  
  "units" : {  
    "name" : "list",  
    "elementType" : "MULTIPOINT",  
    "subUnits": {  
      "listUnits": {  
        "name" : "identifier",  
        "elementType" : "POINT"  
      }  
    },  
    ... Other fields ...  
  },  
}
```

Performance

Interpolation Join, 10 nodes, 32 cores/node



Performance

Interpolation Join, Strong Scaling, 32 cores/node, 16M rows

