Improving SaNSA: Integration with Spark and Tivan

Dakota Fulp - Coastal Carolina University
Mentors: Dr. Nathan DeBardeleben (LANL), Prof. William M. Jones (CCU)

Motivation

HPC system health is reliant on an administrator's understanding of the current state of any given machine. SaNSA aims to assist administrators and users by:

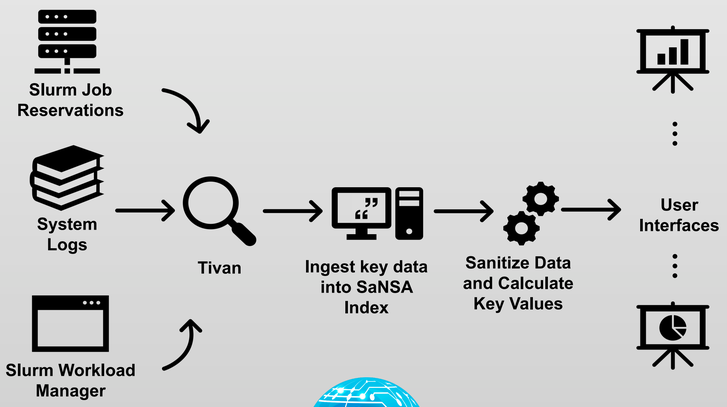
- Keeping up-to-date data on a system, including data regarding node statuses.
- Providing a user-centric graphical interface to assist in visualization and anomaly detection.

To accomplish these goals, vast amounts of data regarding each node are required. This causes issues with regards to scalability and performance and a new approach is required.

Technologies

- *Apache Spark* is a cluster-computing framework and is key to the operations of SaNSA. Through its integration, the time spent on large dataset computations is dramatically decreased.
- *Elasticsearch* is a search engine based on the Lucene library. Through its use large quantities of data can be stored and searched rapidly.
- *Tivan* is a cluster which ingests various types of logs, including job reservations, system logs, and workload manager data. SaNSA utilizes Tivan as the primary source of all signal logs.

Design Flow



Ultrascale Systems Research Center
LA-UR-19-27053

Improvements

Implementing Spark

Apache Spark leverages parallel processing to optimize calculations on large datasets. Our tests used Grizzly data involving ~1,500 nodes.

An example Spark initialization and query can be seen below:

```
# Initialize Spark
spark = start_spark()
sqlContext = SQLContext(spark)

# Get all Job Logs
query = '?q=+timestamp:["{}" TO "{}"]'.format(start_time,end_time)
df = sqlContext.read.format("org.elasticsearch.spark.sql")
.option("es.query", query).load(index_prefix + "*/slurm")
```

Previous Runtime: 8 Hours

Improved Runtime: 45 Minutes* **(966.67% Performance improvement)**

** Time includes new event messages that have since been added.*

New Event Messages

With the increased performance through using Spark,

- The addition of 14 new event messages was feasible (26 in total).
- A more complete view of the system's state was revealed.
- Performance impact of adding new event messages was negligible.

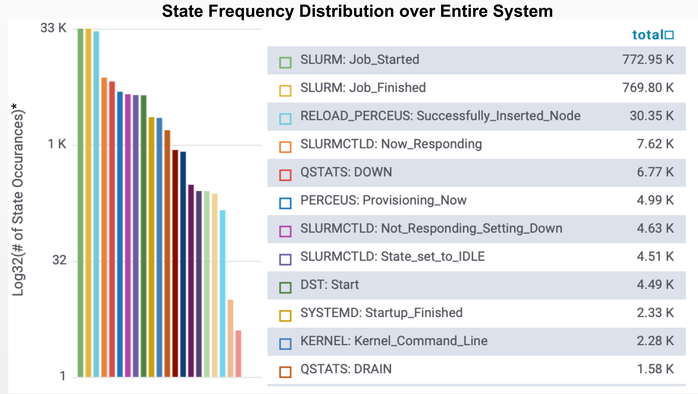
Some examples of such messages can be seen below:

State	Event Message
SLURMCTLD: State_set_to_DOWN	`update_node: node gr0426 state set to DOWN`
KERNEL: Kernel_Panic	`Kernel panic - not syncing: LBUG`
NTPD: Signal 15 Exit	`ntpd exiting on signal 15`
QSTATS: MAINT	`NODE=wc156 STATE=maint* USER=root(0) WHEN=2019-02-13T07:36:34 REASON="Epilog error"`

Grafana Dashboard

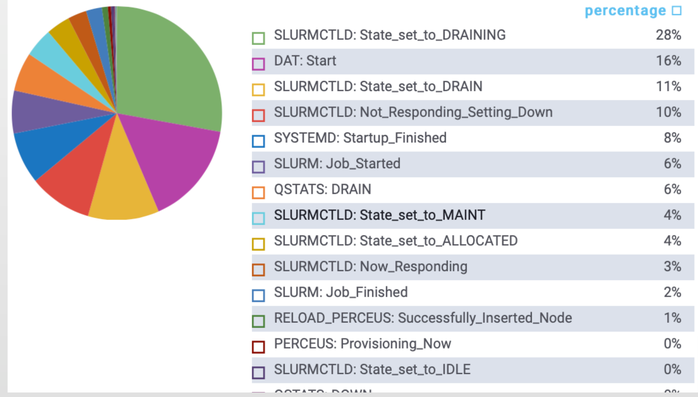
A Grafana dashboard, integrated with Tivan, was developed to assist users in visualizing the gathered data as seen below:

State Frequency Distribution over Entire System



** Note: Log Scale*

Percent of Time in State over Entire System



Future Work

- Automation of hourly data collection through Tivan.
- Separation of data on certain criteria (perspective | category).
- Addition of further signal messages.
- Generation of Markov chains to assist in detection of anomalous nodes.