google.com/datacenters

ANALYZING GOOGLE CLUSTER DATA

Advised by Prof. Ling Liu

Abhishek Sen
Saket Gejji
Shruthi Hiremath
Thomas George Ancheriyil

MISSION STATEMENT

Drawing Insights from Public Data

- Google Cluster Data, John Wilkes, 2011
- 12.5k machines, 29 days of trace

THE PROBLEM

‘The Good’
Cloud Computing

- High Scalability
- High Flexibility
- Cost Effectiveness

‘The Bad’
Data Centers

- Petabytes of log data
- No efficient way to analyze logs
- Difficult to draw insights from traces

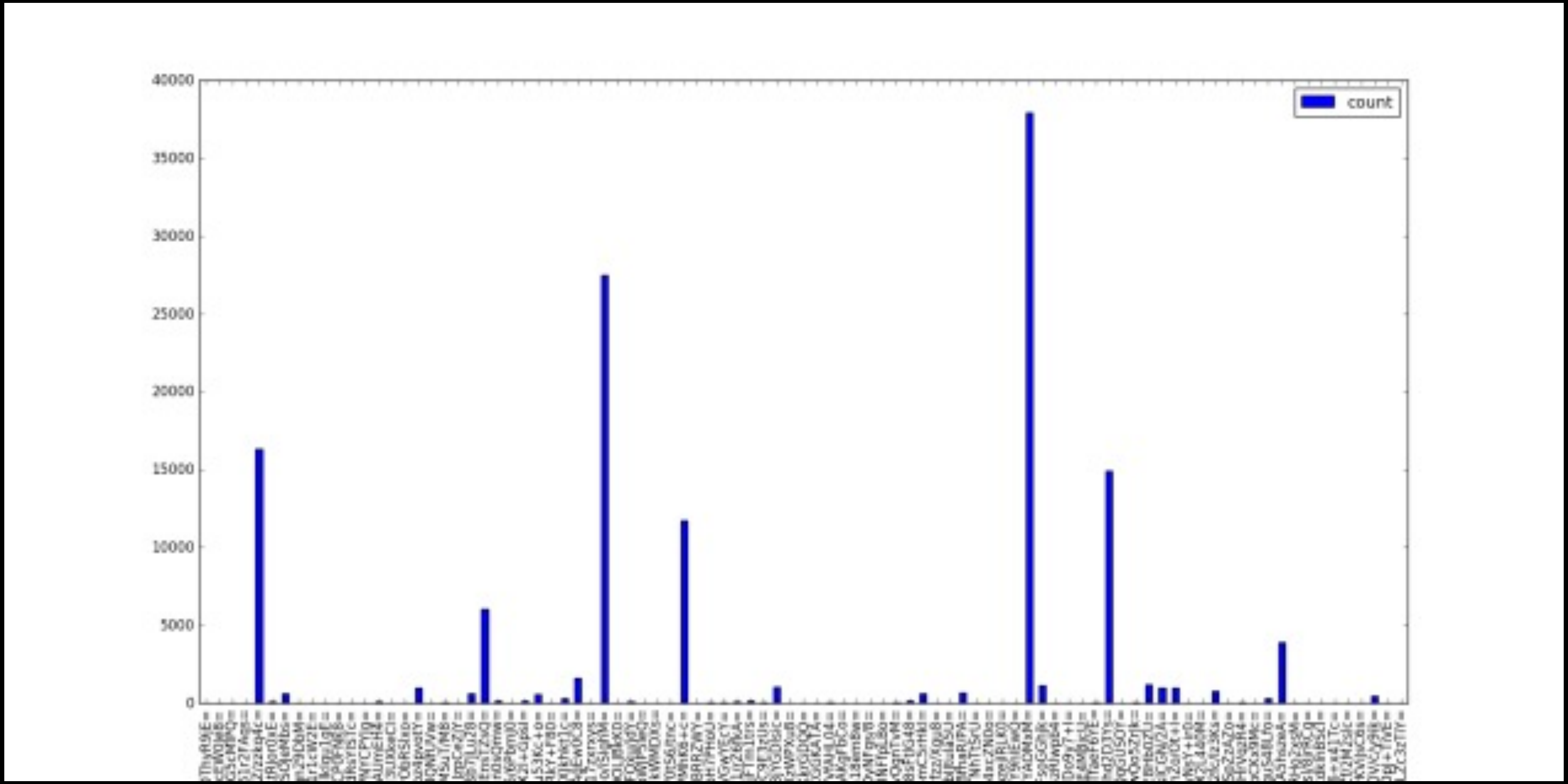
PROBLEMS OF INTEREST TO GOOGLE

- Workload characterizations
- Synthetic load generation for benchmarking
- Visualizing workloads
- Predictive modeling

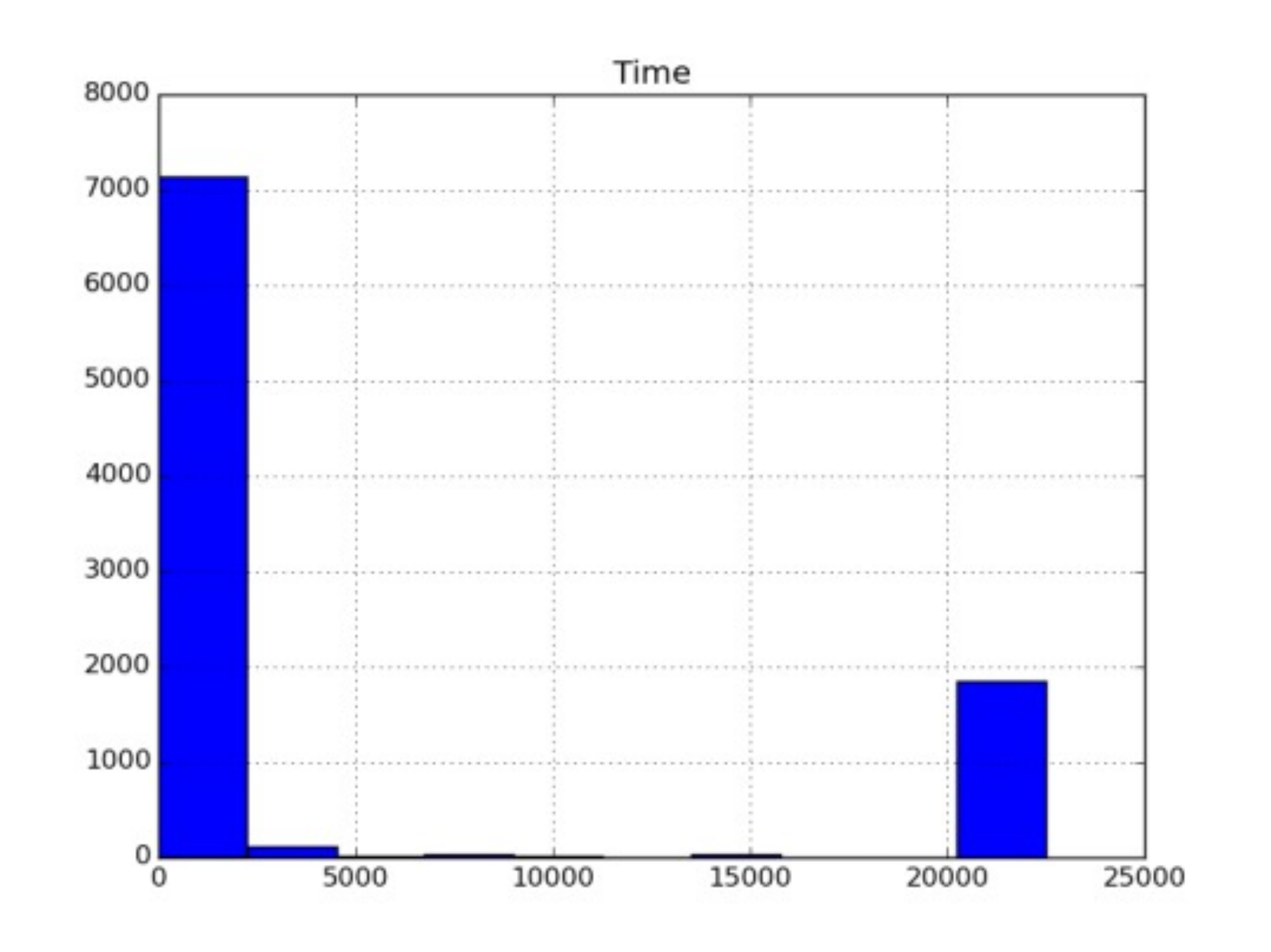
SMALL BEGINNINGS

- Smaller version of cluster trace for initial analysis
- How small? 3.5 million rows
- Provides information on normalized memory and CPU usage per user per task
- Use Python Pandas to draw initial insights

GROUPING TASKS BASED ON USER

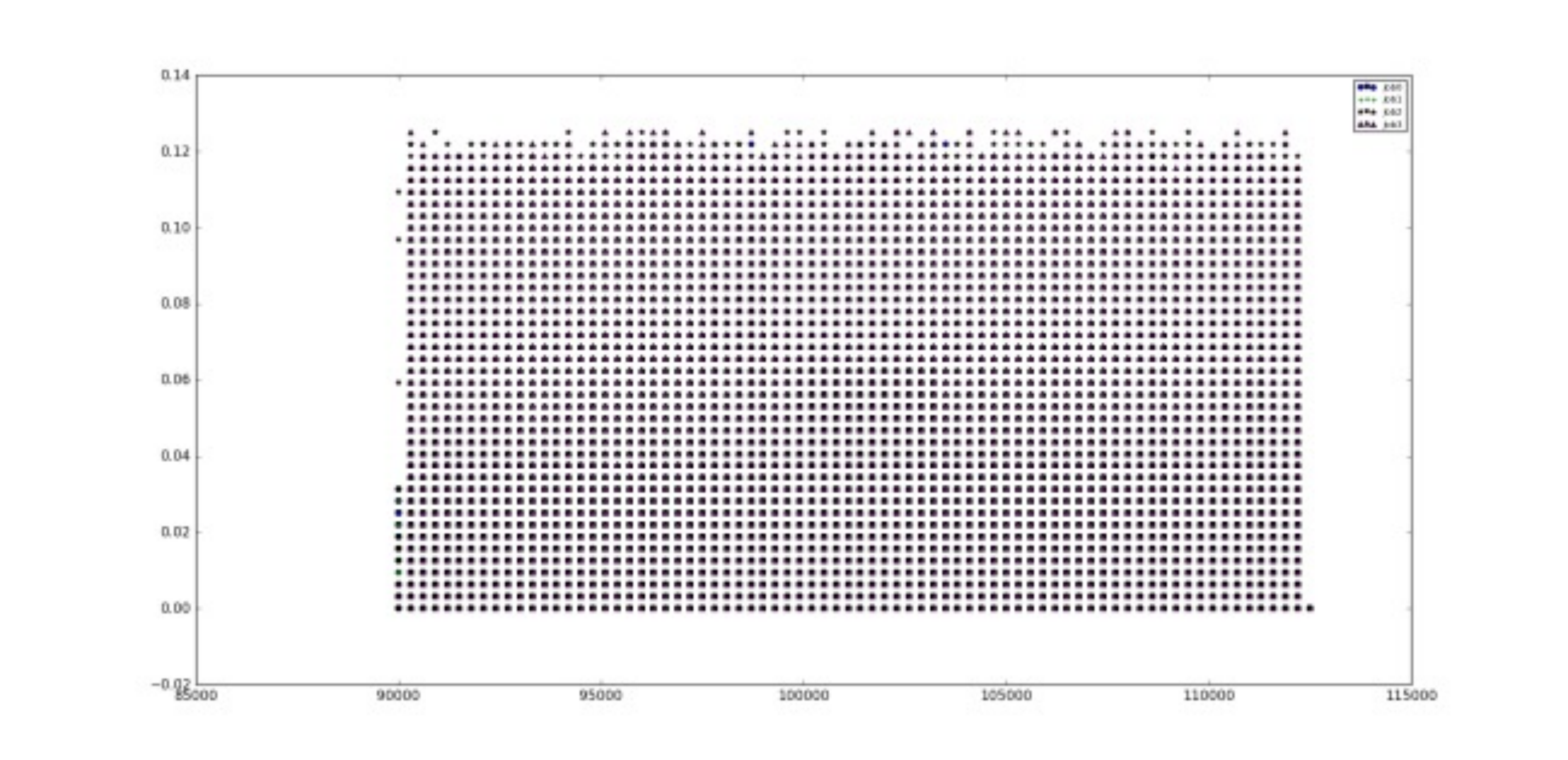


TIME SPENT BY JOBS

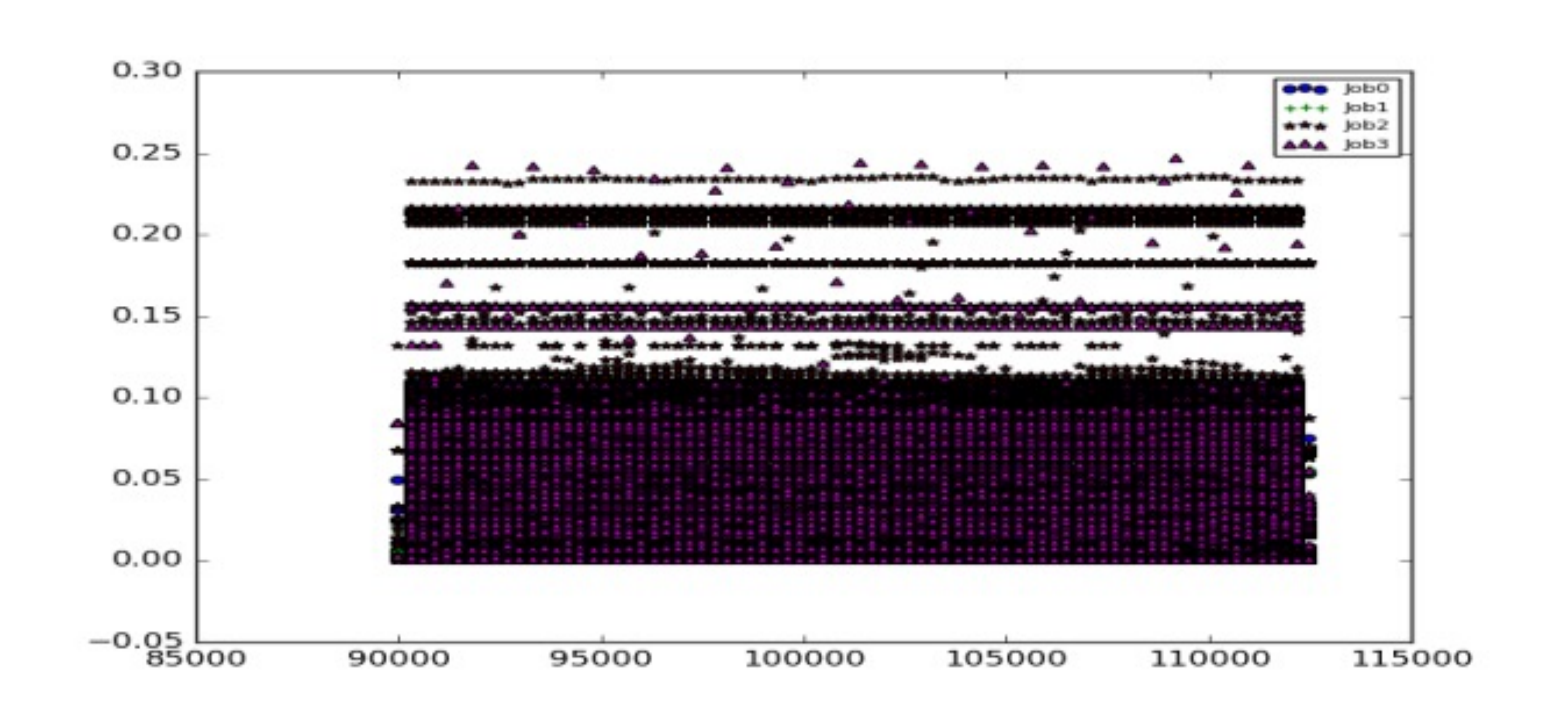


Two types of Jobs: Bursty and Service

NORMALIZED TASK CORES ASSIGNED TO EACH JOB TYPE



NORMALIZED TASK MEMORY ASSIGNED TO EACH JOB TYPE



ISSUES

Impossible to iterate through the large data set sequentially using this approach

DIVING INTO THE LARGE DATA SET

- How large? 3.2 billion rows
- Actual logs from Google Cluster containing approx. 12500 machines over 29 days
- Data set contains information on events, CPU, memory(main and cache) for each machine, task, user and job

TECHNOLOGY

- SQL vs NoSQL
- Need fast queries to columns
- Utilize the parallelisms exposed by columnar databases/data warehouses coupled with SQL based queries
- Hadoop, Google BigQuery, Amazon Redshift

WHY BIGQUERY

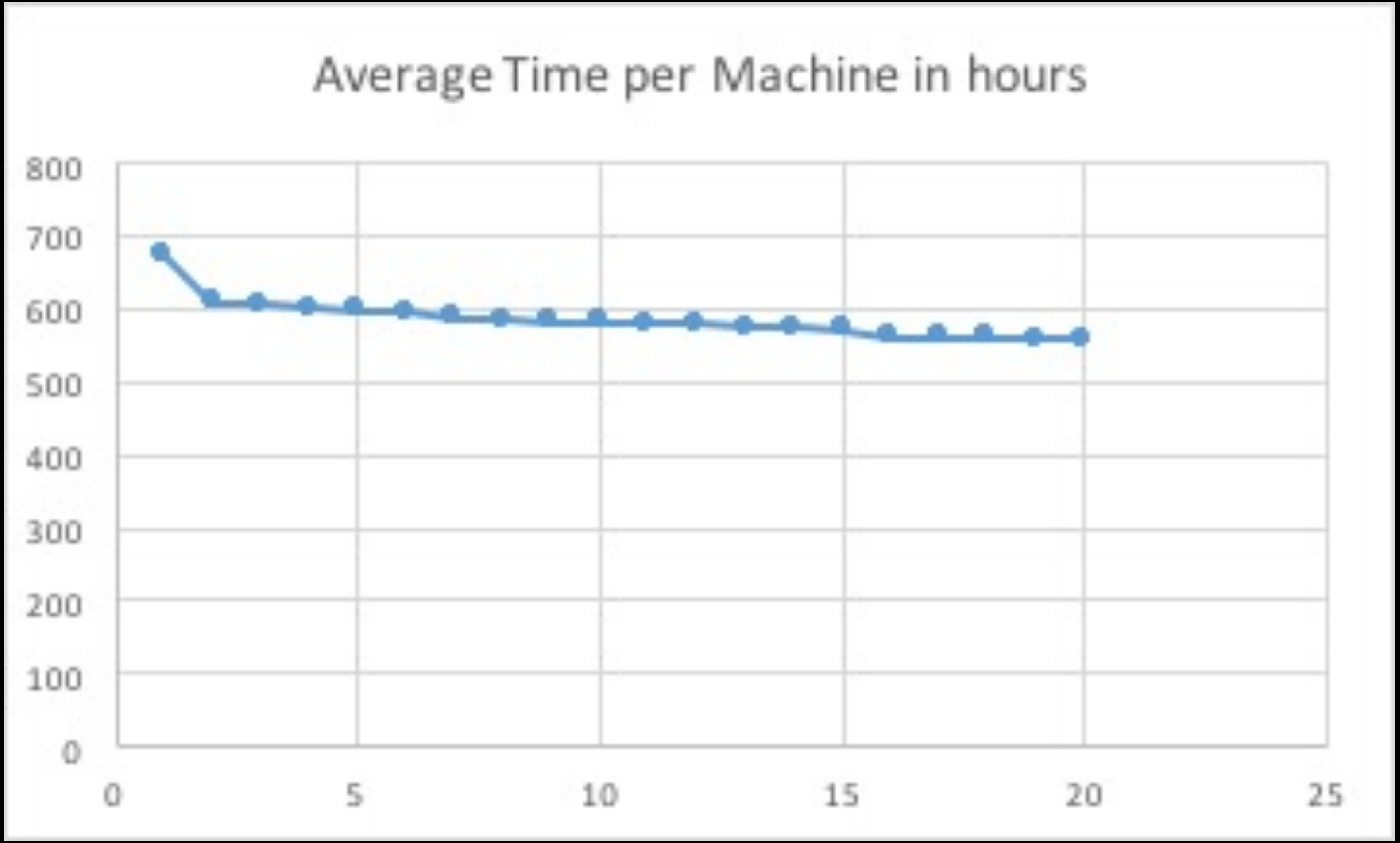
Google BigQuery

- Optimized for columnar storage
- Interacting with tables is fast
- Provides super-fast SQL-based queries against append-only tables
- Accessible using Web-UI, CLI or REST API

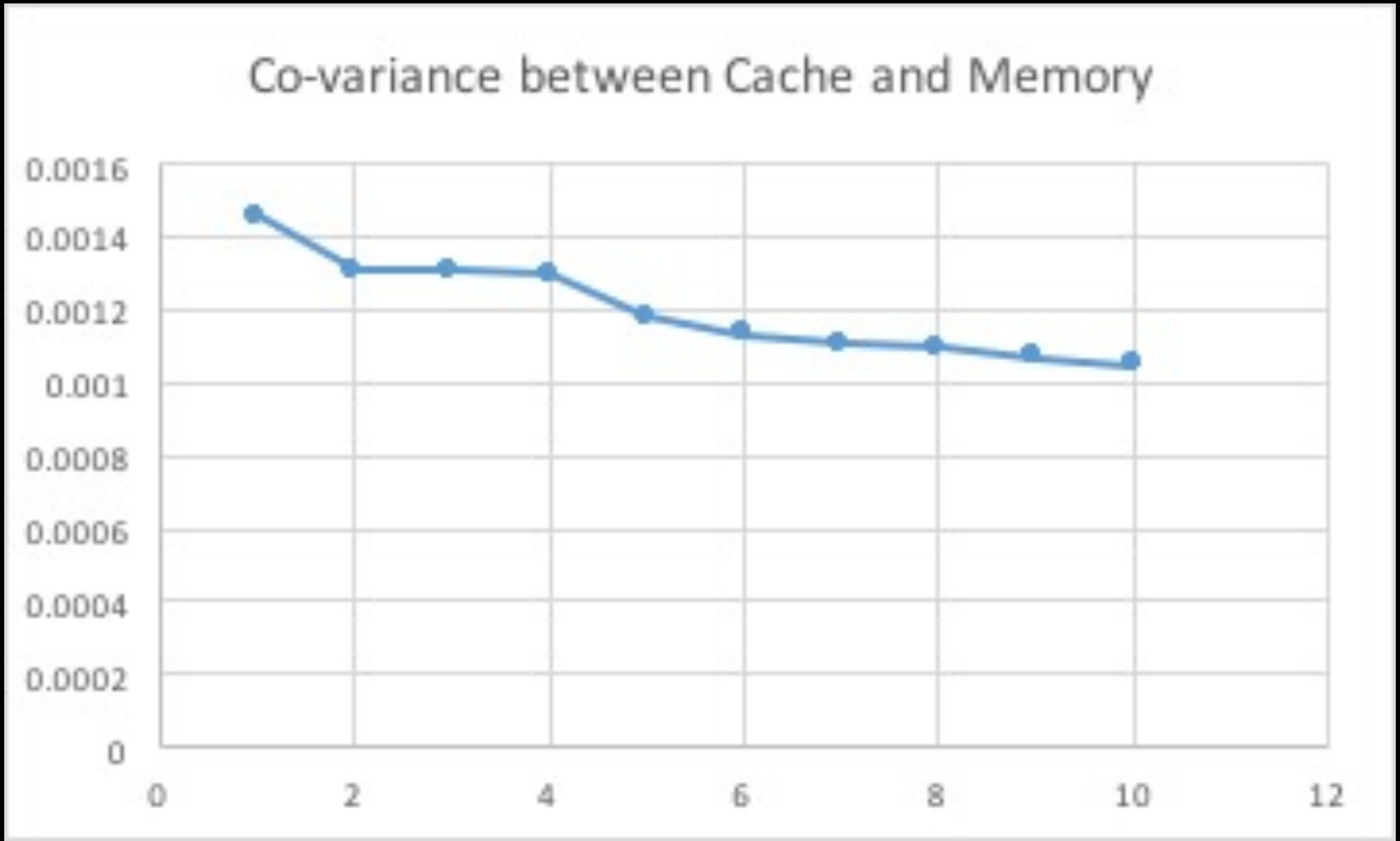
ANALYSIS

- Per machine usage
- CPU, Memory requested vs available
- Event histograms

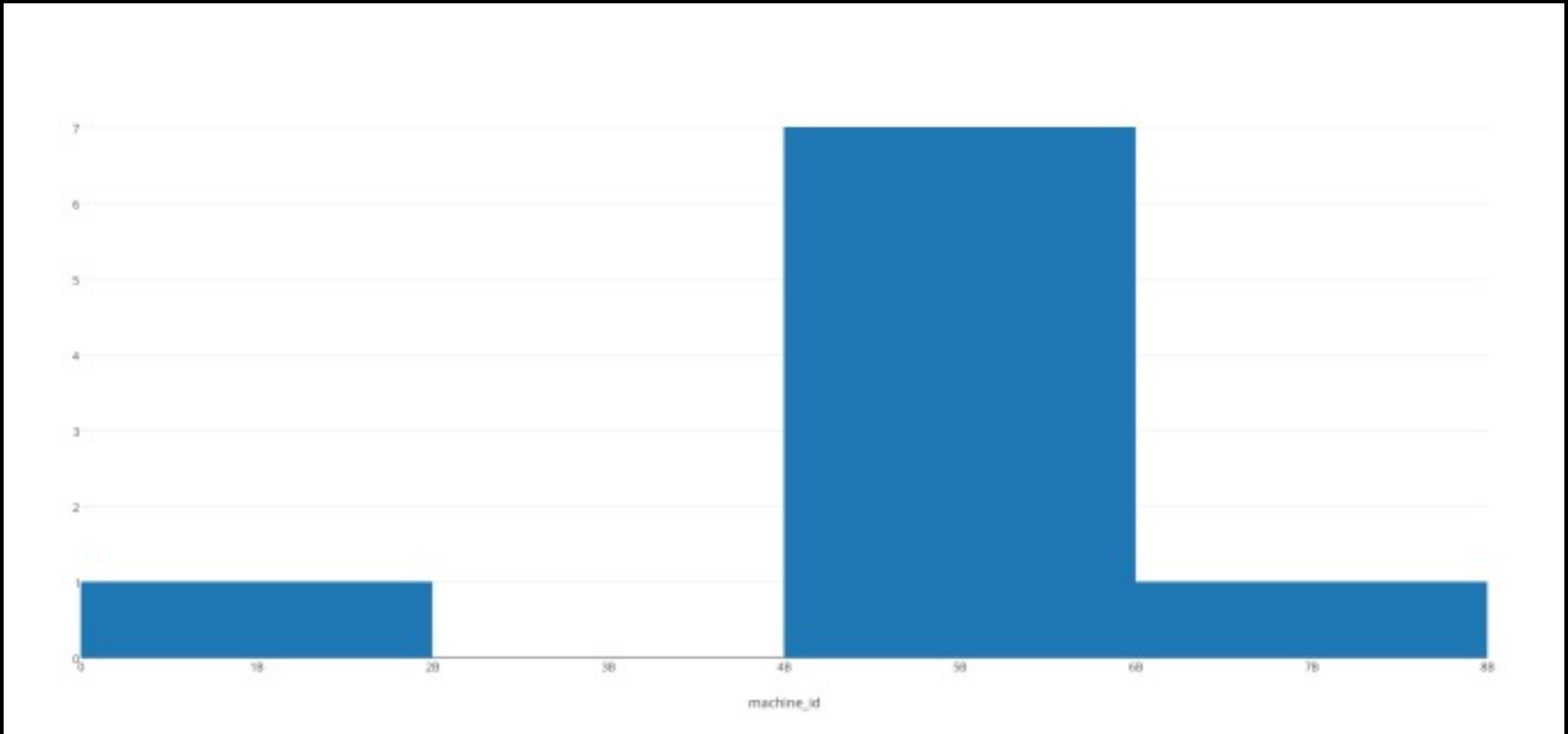
AVERAGE TIME SPENT PER MACHINE



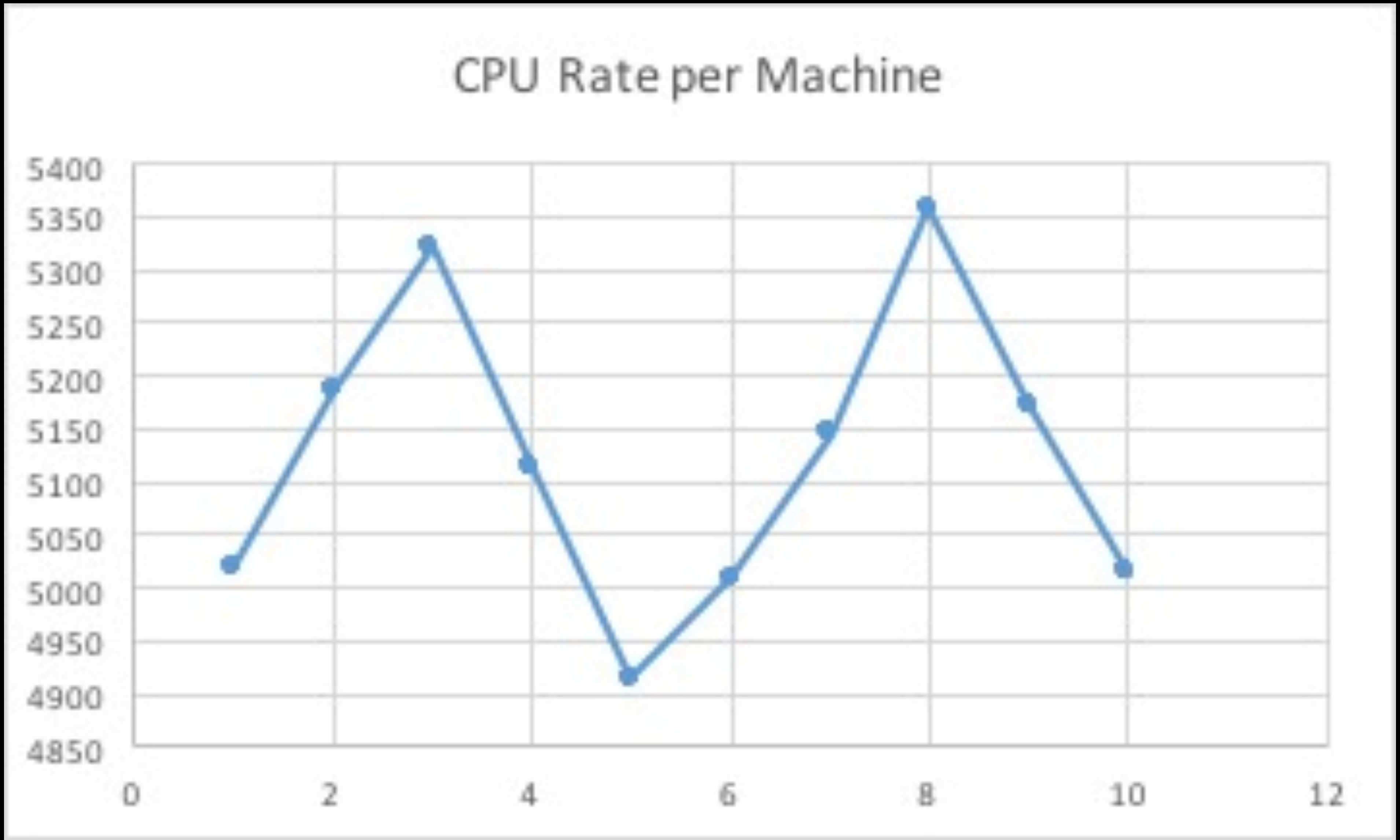
COVARIANCE BETWEEN CACHE AND MEMORY



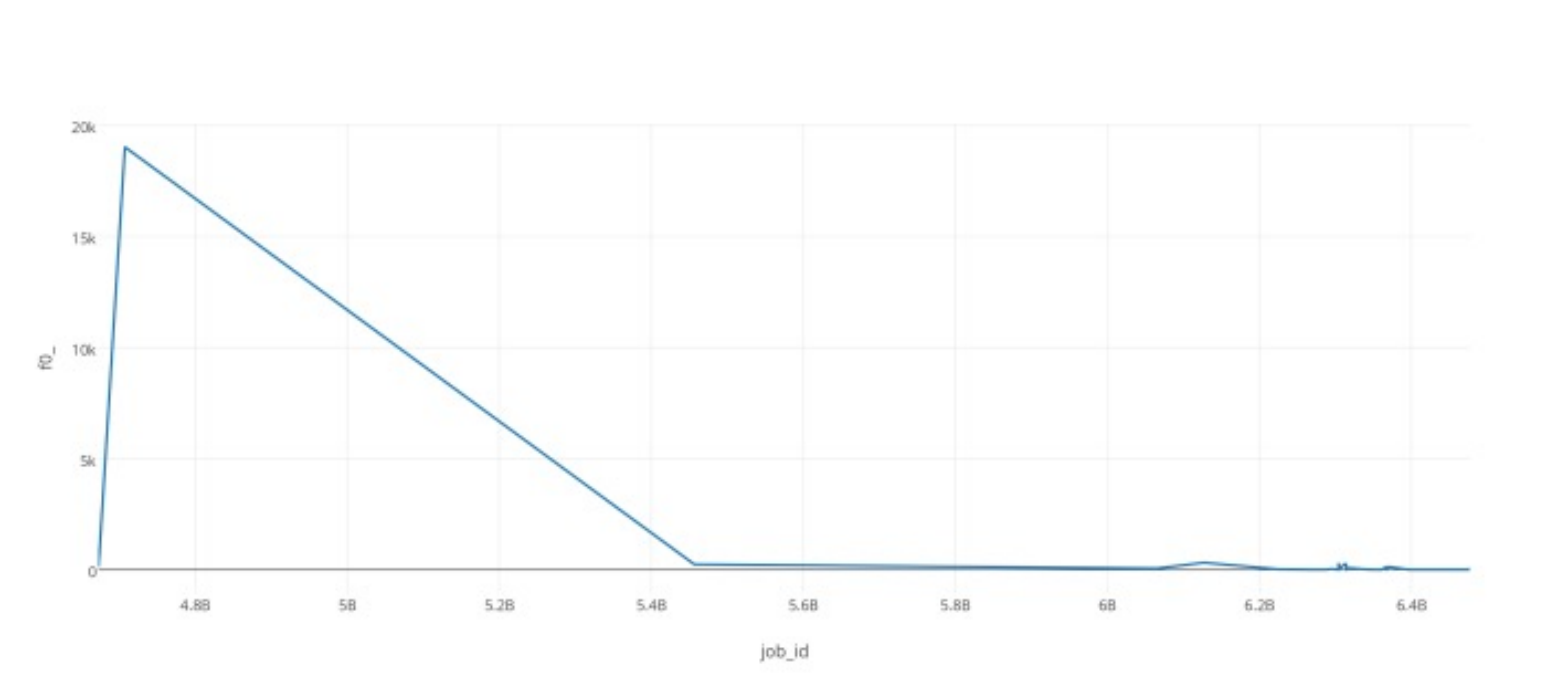
EVENTS PER MACHINE



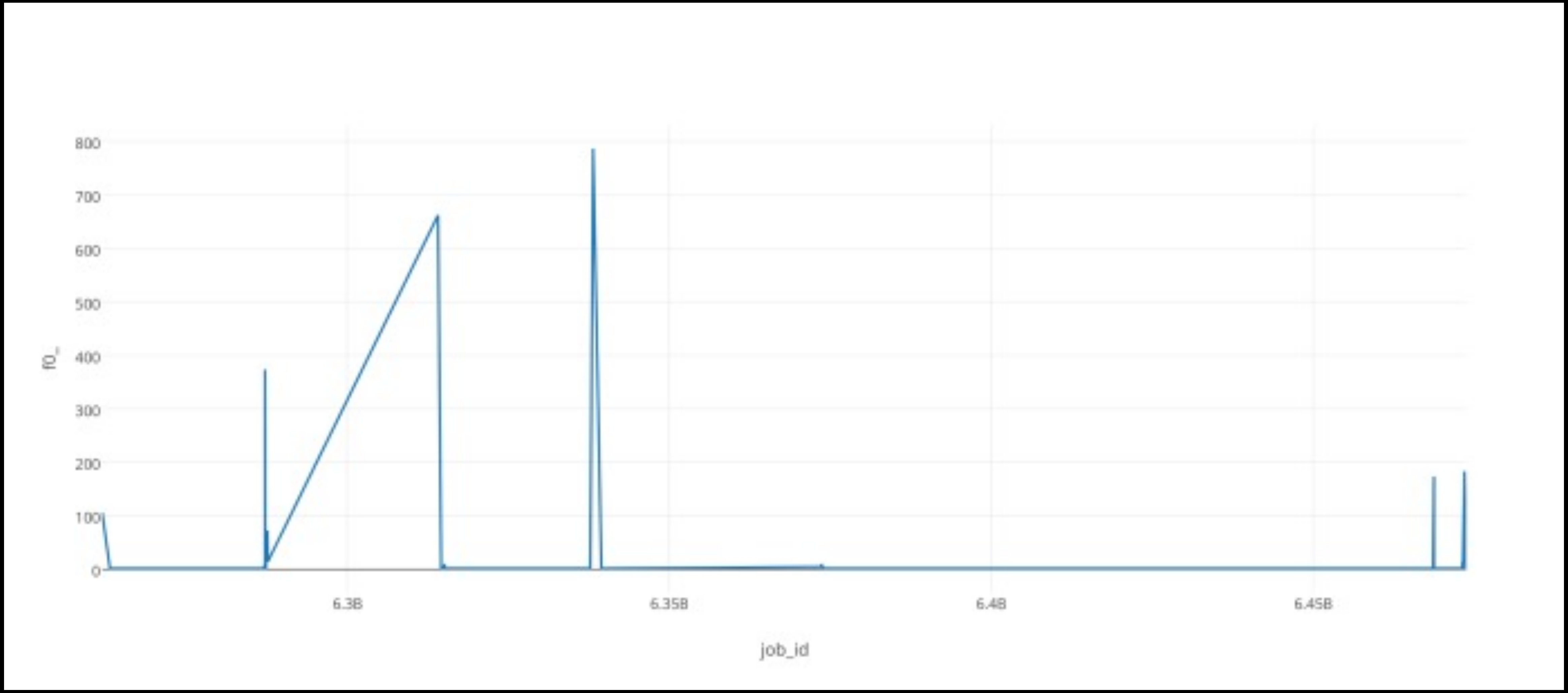
CPU RATE PER MACHINE



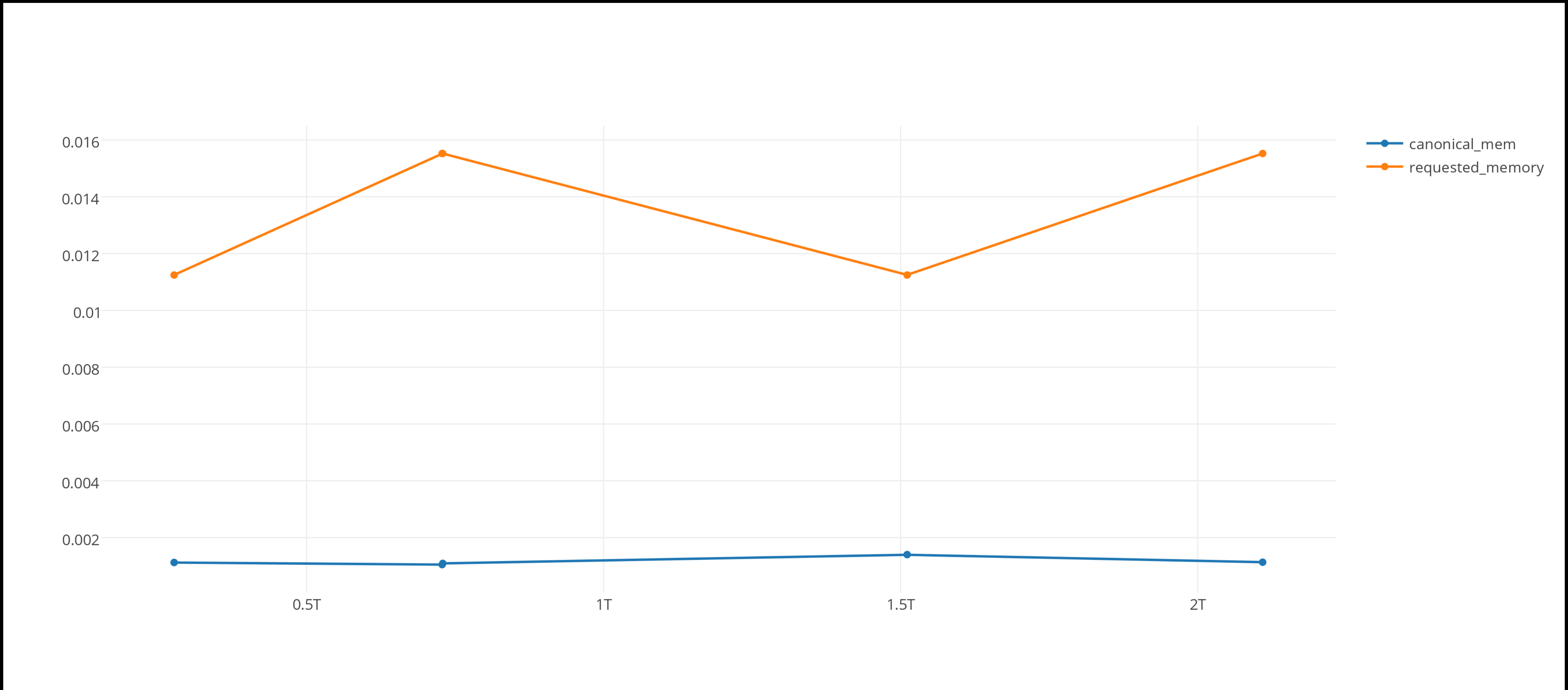
JOBS SCHEDULED AND THEN FAILED, KILLED OR LOST



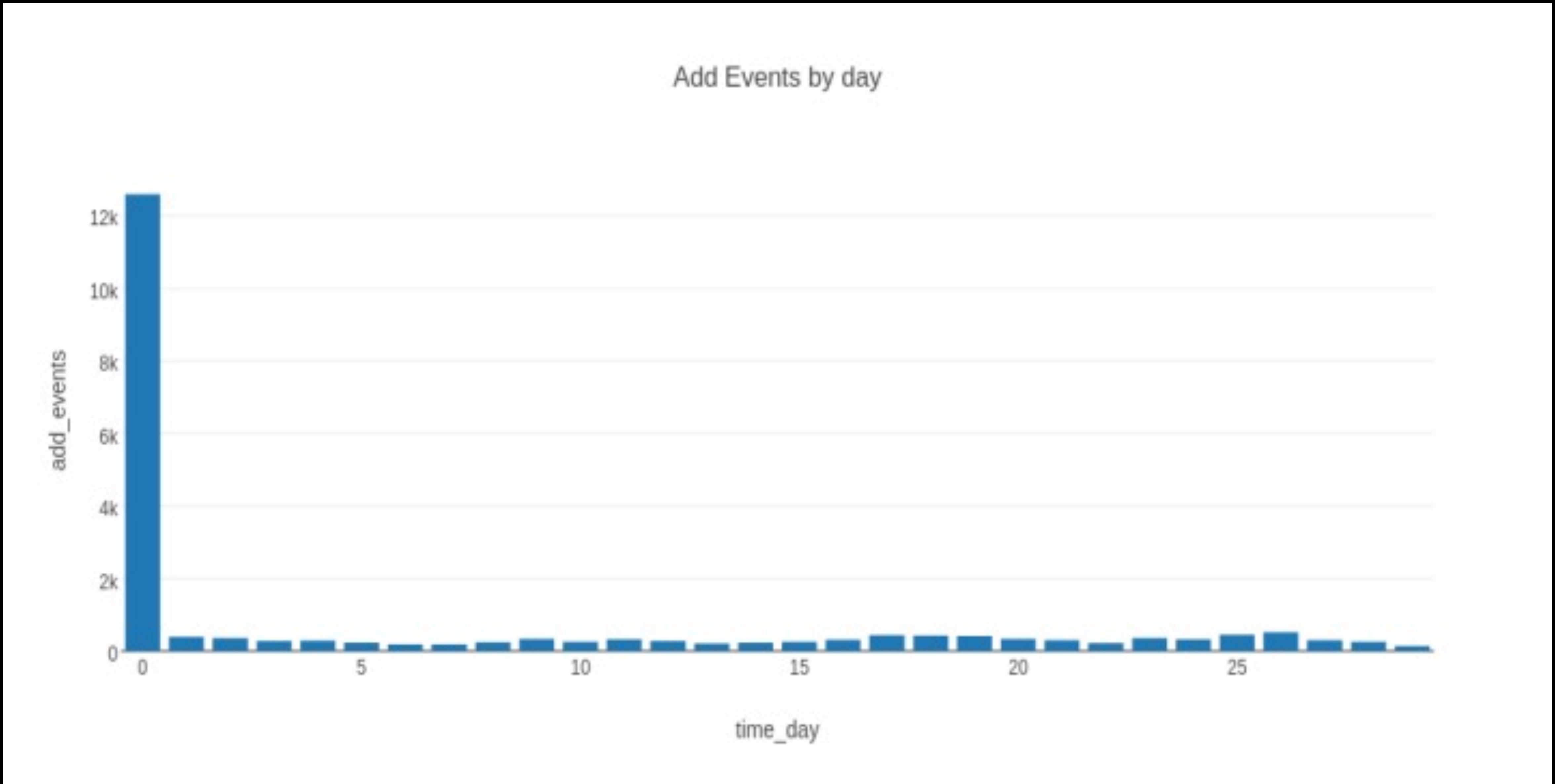
JOBS SUBMITTED AND THEN EVICTED



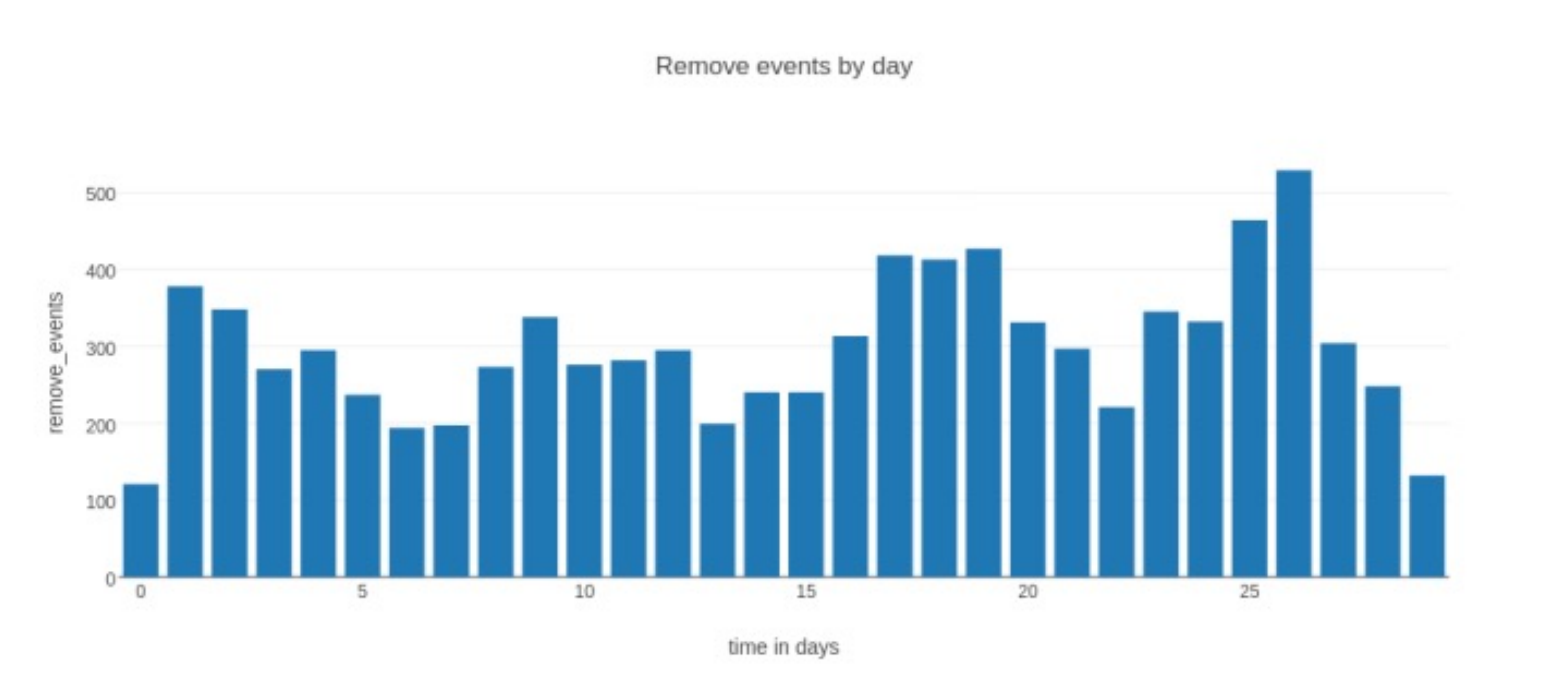
CANONICAL AND REQUESTED MEMORY



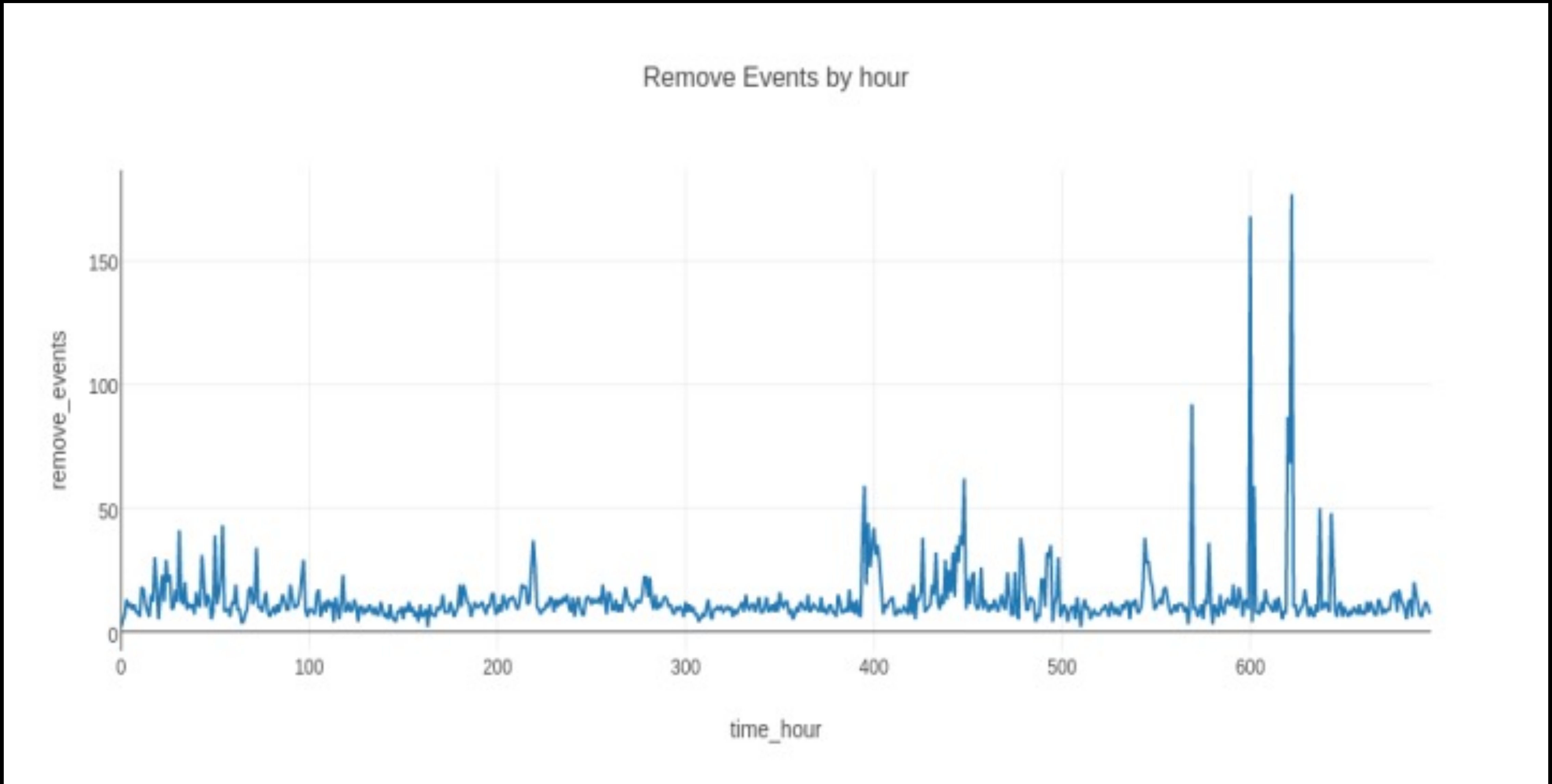
ADD EVENTS BY DAY



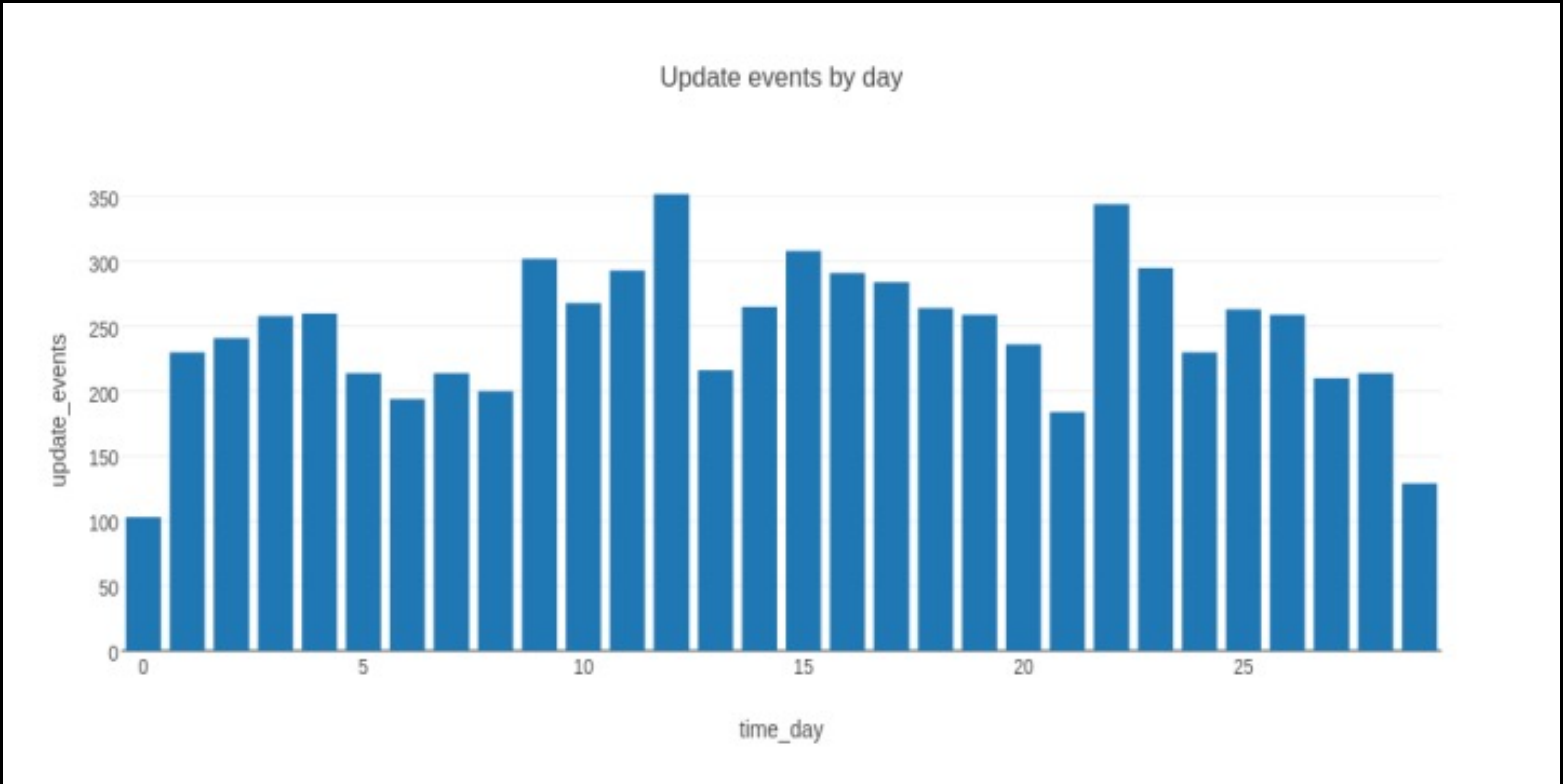
REMOVE EVENTS BY DAY



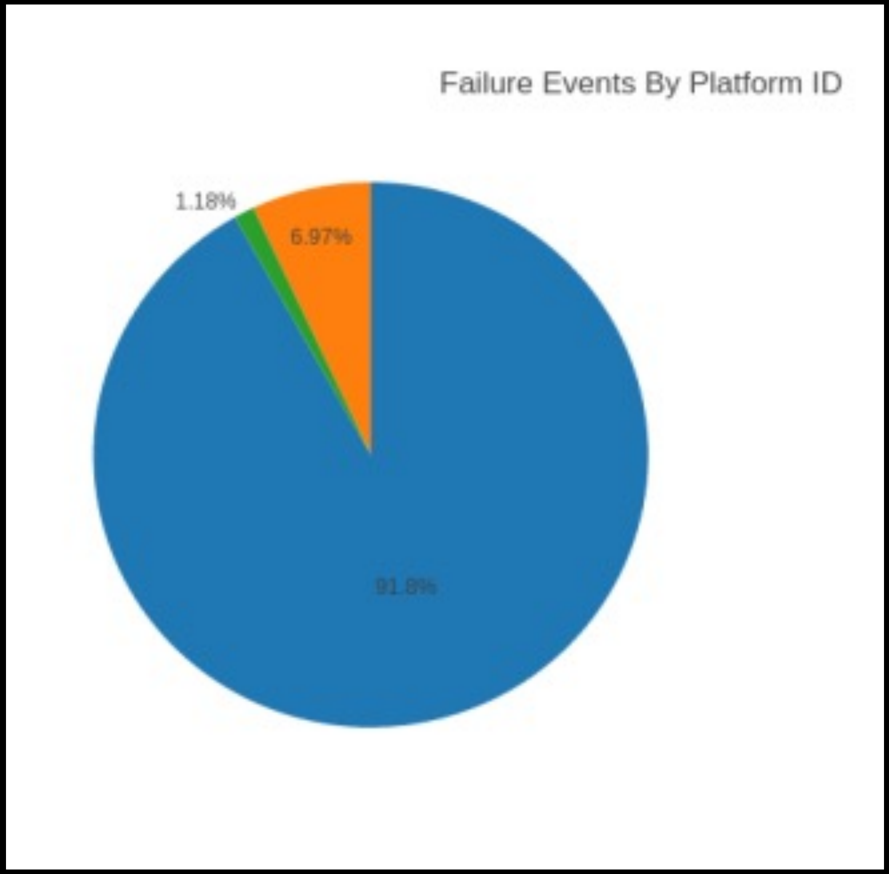
REMOVE EVENTS BY HOUR

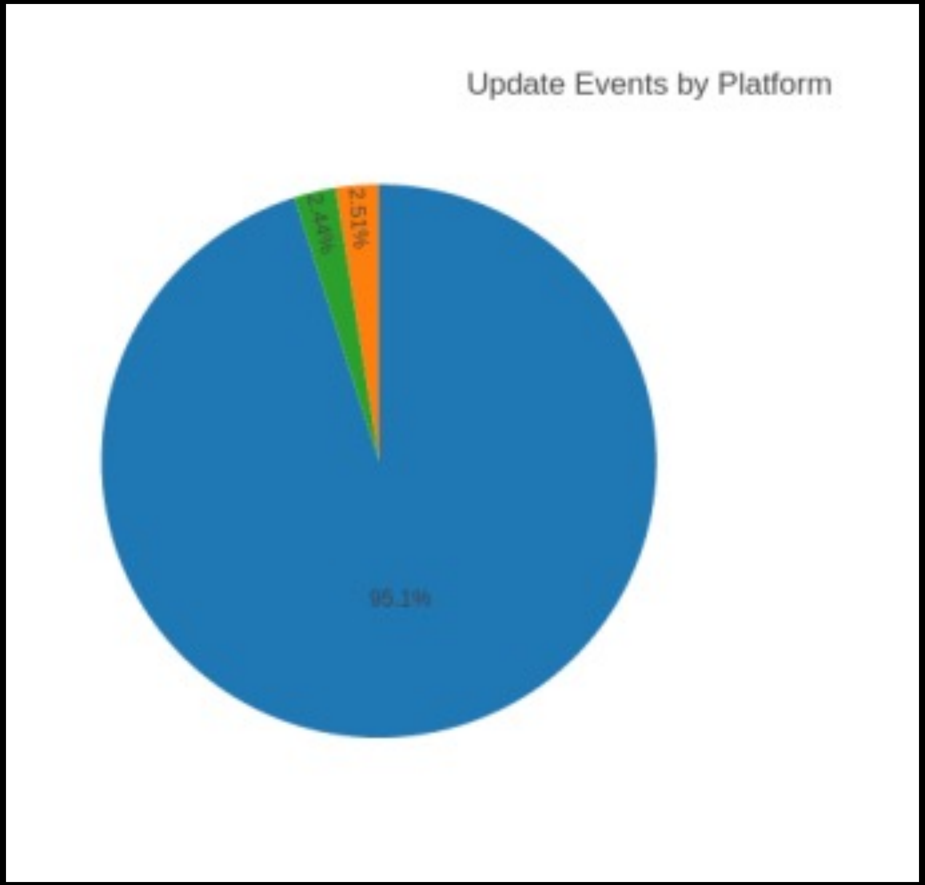


UPDATE EVENTS BY DAY



FAILURE AND UPDATE EVENTS BY PLATFORM





CHALLENGES

- Dealing with cluster logs is compute intensive and hence expensive
- Unintuitive to get insight from data with too many metrics
- Obfuscated information in the trace makes it difficult to draw conclusions

FUTURE WORK

- Use inexpensive methods to query
- Model data at a more coarse granularity
- Arrive at a solution towards autonomics

THANK YOU