Supercharged Analytics for Prometheus Metrics with Spark, Presto, & Superset

Rob Skillington and Gibbs Cullen, Chronosphere

November 19, 2020

Who are we?

Rob Skillington

CTO / Co-Founder at Chronosphere
M3DB Creator
OpenMetrics Contributor
Twitter: @robskillington

Gibbs Cullen

Developer Advocate at Chronosphere
Previously Product Manager at AWS
Twitter: @gibbscullen

Preview

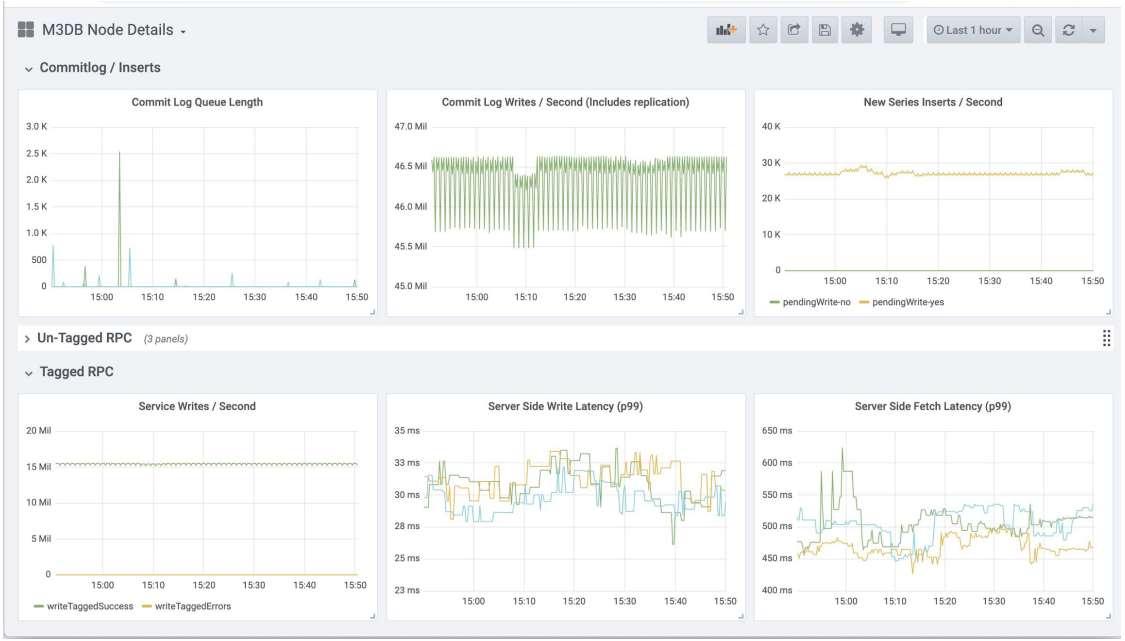
Agenda

- Metrics & Analytics
- Problem
- Solution
- Demo
- Q&A

Metrics

Metrics

- Essential for **monitoring** and **alerting**

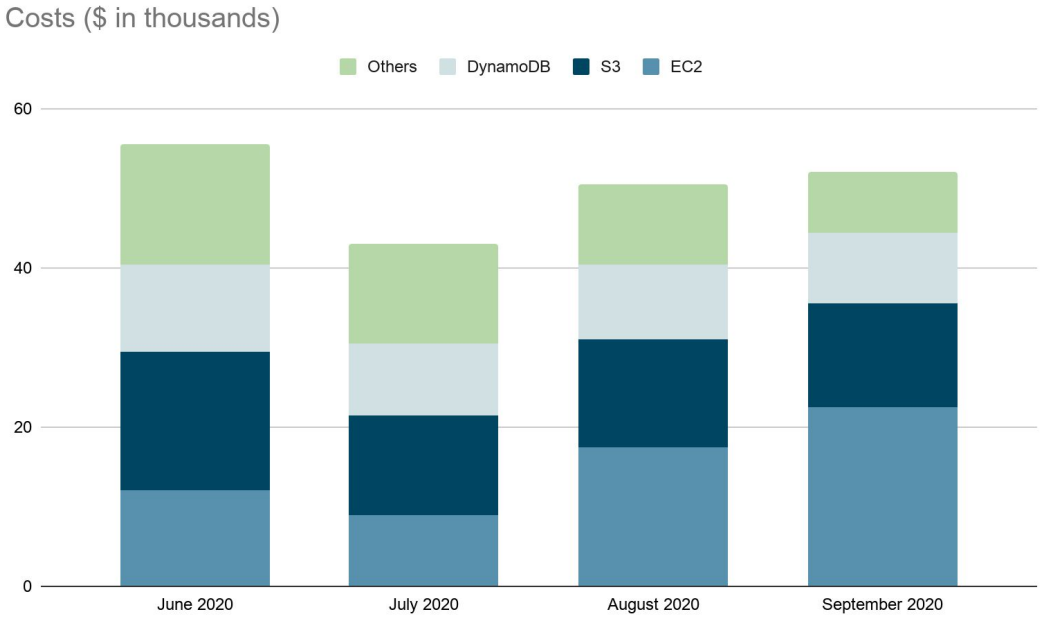


- But also has historical data that would be valuable for **analytic** use-cases

Analytics

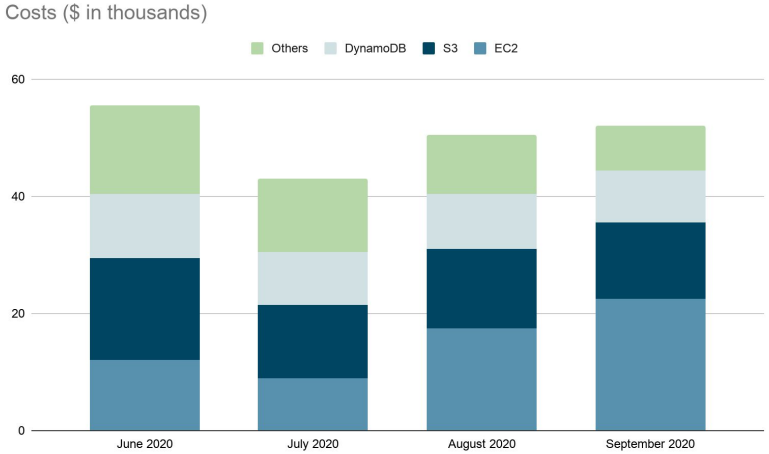
Example: Cost Monitoring

- Imagine your infrastructure team sees the following cost report from your cloud provider (e.g. AWS)



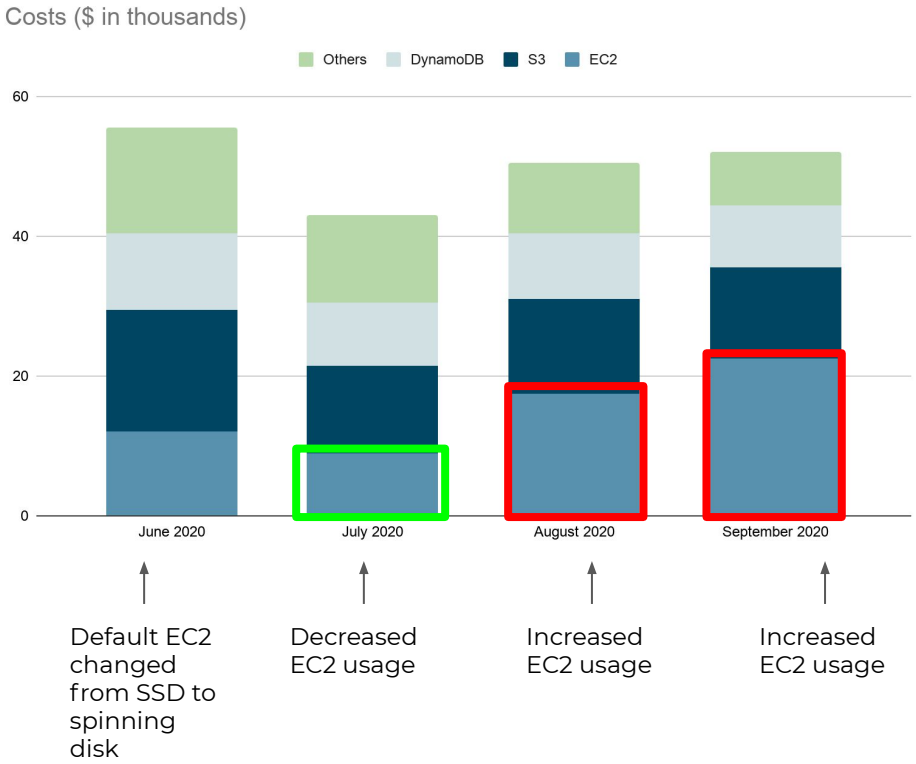
Example: Cost Monitoring

- Looks like the costs for **EC2 are increasing over time**, so we want to understand why



	EC2	S3	DynamoDB	Others
June 2020	**12**	17.5	11	15
July 2020	**9**	12.5	9	12.5
August 2020	**17.5**	13.5	9.5	10
September 2020	**22.5**	13	9	7.5

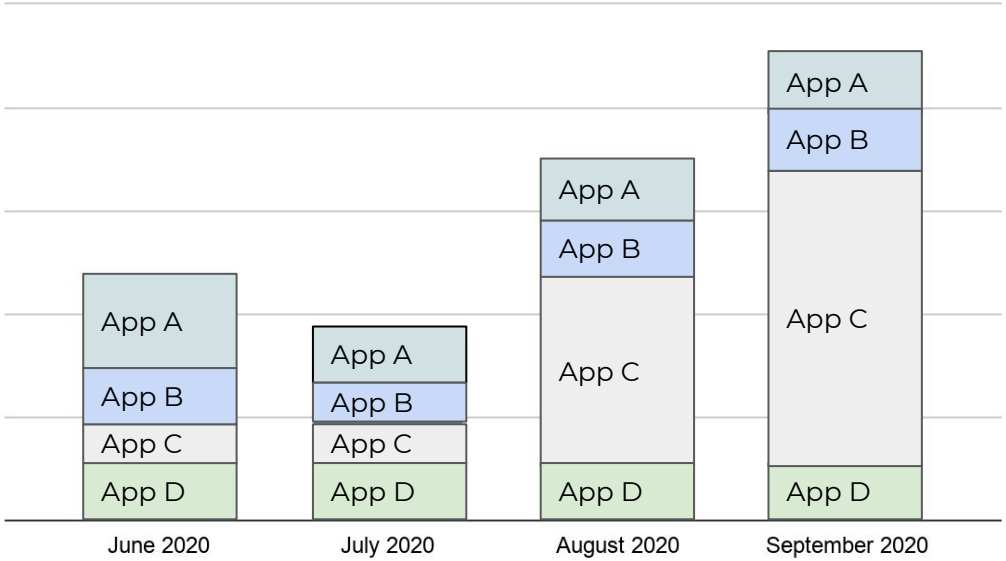
Example: Cost Monitoring (Cloud Bill)



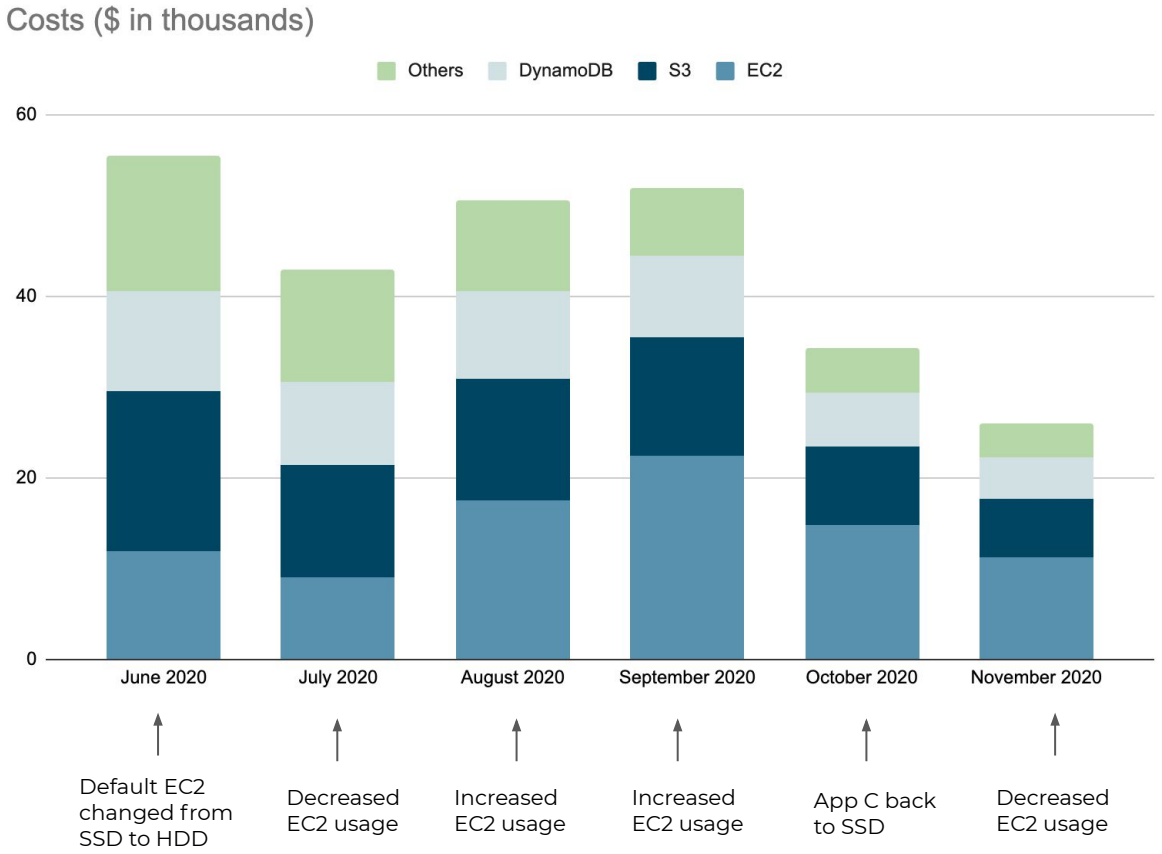
What is causing this increase over time??

Example: Cost Monitoring (App + Infra Metrics)

- With **more granular analytics on metrics**, we can breakdown the EC2 allocations into specific applications across all clusters over long time windows

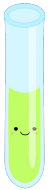


Example: Cost Monitoring

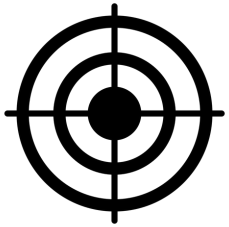


Other analytic use-cases?

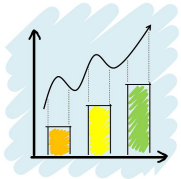
- Data science



- Optimizing



- Forecasting



Example: Data Science

- Leverage historical metric data to **extract outcome probabilities and back-test algorithms** that might have been more optimal

- E.g. imagine running a high-volume real-time ad-bidding platform

Example: Data Science

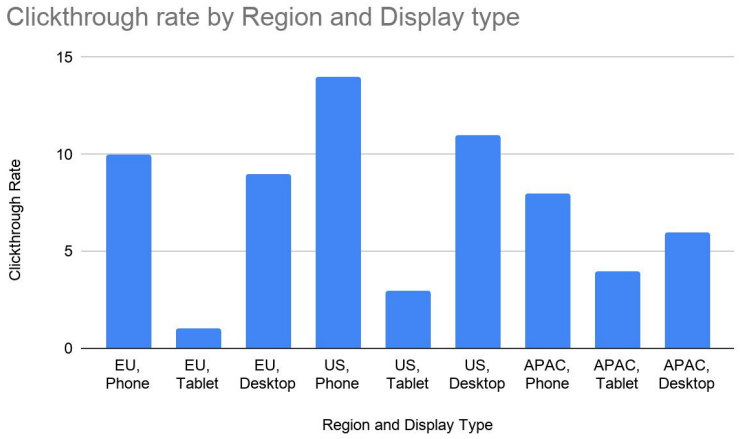
- Leverage historical metric data to extract outcome probabilities and back-test algorithms that might have been more optimal

- E.g. imagine running a high-volume real-time ad-bidding platform
  - Metrics for ad bidding
    - Number of ads purchased
    - Number of ads with clickthroughs
    - Latency to respond to ad request
  - Metric tag values for type of ad and where it was displayed
    - Display ad type (e.g. mobile/desktop)
    - Regions (e.g. Europe, US, etc)
    - Machine learning model used
    - Machine learning model config Parameters

Example: Data Science

- Leverage historical metric data to extract outcome probabilities and back-test algorithms that might have been more optimal

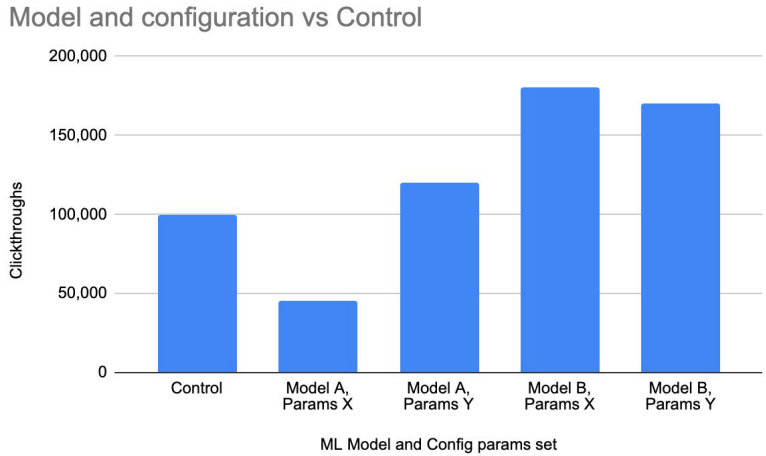
- E.g. imagine running a high-volume real-time ad-bidding platform
  - Metrics for ad bidding
    - Number of ads purchased
    - Number of ads with clickthroughs
    - Latency to respond to ad request
  - Metric tag values for type of ad and where it was displayed
    - Display ad type (e.g. mobile/desktop)
    - Regions (e.g. Europe, US, etc)
    - Machine learning model used
    - Machine learning model config Parameters
  - **Compute expected clickthrough** broken down by display type and regions



Example: Data Science

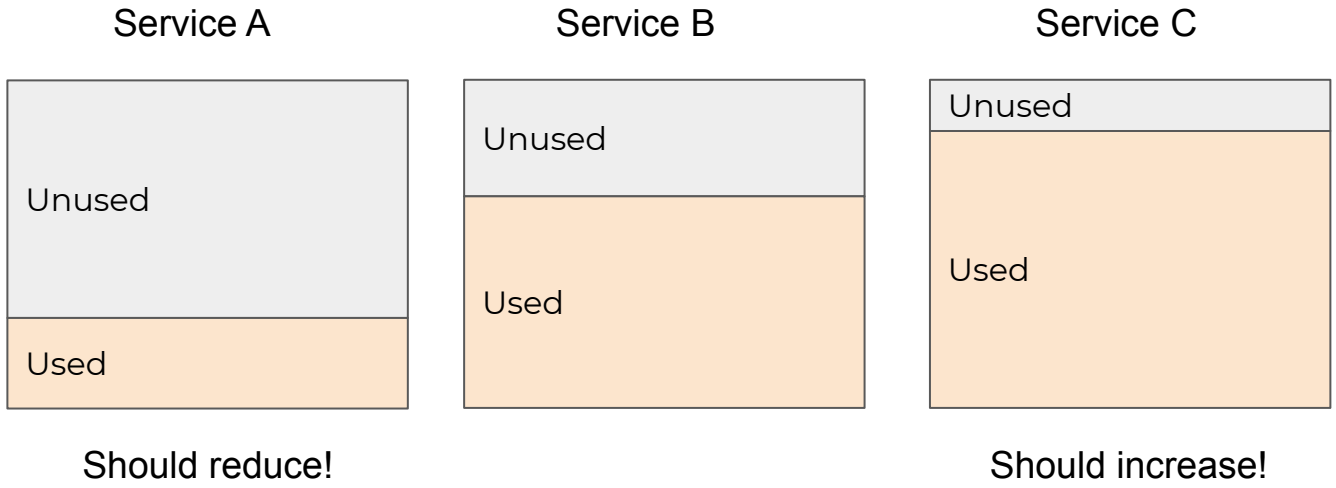
- Leverage historical metric data to extract outcome probabilities and back-test algorithms that might have been more optimal

- E.g. imagine running a high-volume real-time ad-bidding platform
  - Metrics for ad bidding
    - Number of ads purchased
    - Number of ads with clickthroughs
    - Latency to respond to ad request
  - Metric tag values for type of ad and where it was displayed
    - Display ad type (e.g. mobile/desktop)
    - Regions (e.g. Europe, US, etc)
    - Machine learning model used
    - Machine learning model config Parameters
  - **Now dissect by different machine learning models and config params**



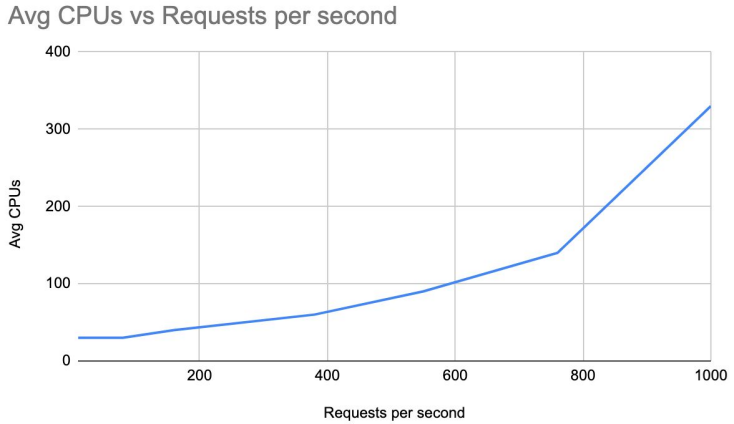
Example: Optimizing

- Identity highest yield optimizations areas
- E.g. services reliably *under* or *over* utilizing CPU and/or memory
- Aggregate (e.g. p95 or avg) across long time period, grouped by service



Example: Forecasting

- Identify correlation between resource metrics (e.g. memory) and usage metrics (e.g. user requests)
- E.g. organization forecasting end of year with 30% growth of user request volume, how does that translate to % growth of application CPU / memory
- Forecasting necessary additional resources likely not as simple as +30%... so look at historical trend of user count relative to memory usage



Analytic power!

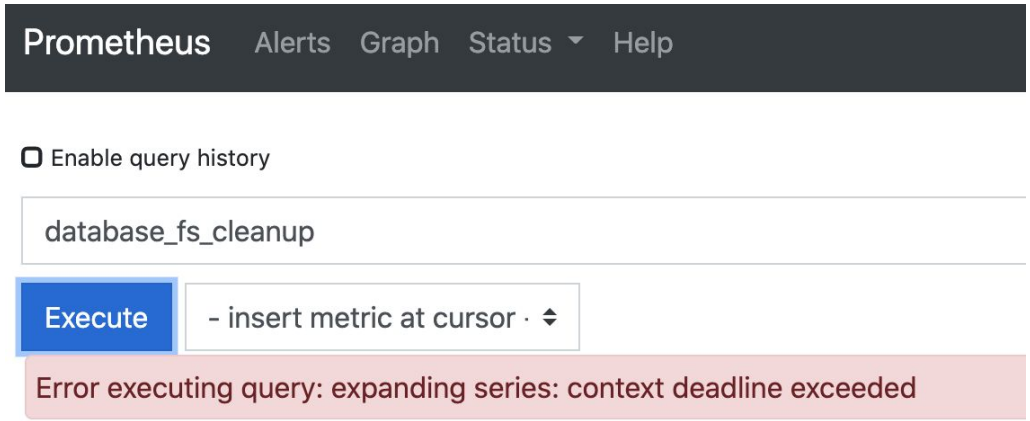
In summary... metrics can unlock a lot of analytic value!

> “In one case, we found a service that was wasting enough RAM to pay my salary for a decade.”
>
> - Dan Luu @ https://danluu.com/metrics-analytics/

The Problem

Why is metrics analytics hard?

- Existing monitoring solutions don't perform well or timeout when running long-term queries that also span a high cardinality set of metrics
  - Prometheus has a **max sample limit** as well as a default **timeout** of 2 minutes
  - Results are typically computed **in-memory**



What a Real Solution needs

- Big data query engine built to execute queries of arbitrary size
- Both Spark and Presto will spill intermediate query results to disk

Does my data need to fit in memory to use Spark?

No. **Spark's operators spill data to disk if it does not fit in memory**, allowing it to run well on any sized data.

Spill to disk > Overview

In the case of memory intensive operations, **Presto allows offloading intermediate operation results to disk**. The goal of this mechanism is to enable execution of queries that require amounts of memory exceeding per query or per node limits.

What a Real Solution needs

- Way to utilize and gather insights from the data, such as Spark and/or machine learning tooling
  - e.g. MLlib, a popular machine learning library, natively integrates with Spark but not Prometheus

Ease of Use

Usable in Java, Scala, Python, and R.

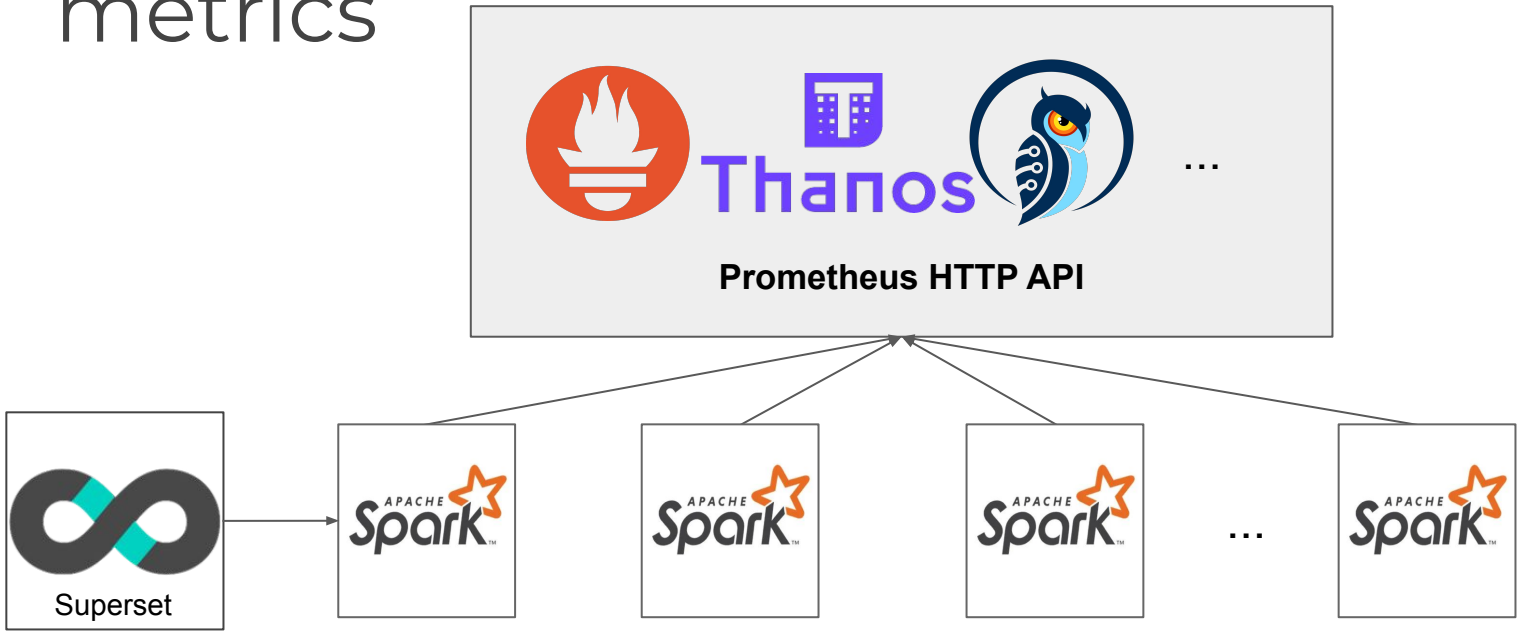
MLlib fits into Spark's APIs and interoperates with NumPy in Python (as of Spark 0.9) and R libraries (as of Spark 1.5).

What a Real Solution needs

- Ability to join metrics with other data, as not all monitoring data lives in metrics store
  - e.g. Kubernetes cluster node information in ElasticSearch or MySQL

One real solution

Spark or Presto with Prometheus metrics



Let’s analyze something

Which Kubernetes deployments account for the majority of resources over time?

- This type of query lends itself more to SQL than a time-series specific query language such as PromQL
  - Many group-bys
  - Sub-aggregations

Demo

https://github.com/chronosphereio/demo-metrics-analytics

Thank You
Q&A