Near Real-Time Recommendations - Spark Streaming

Elliot Chow, Netflix
Nitin Sharma, Netflix

#ML7SAIS

Agenda

- Recommendations @ Netflix
- The Need for Near Real Time
- Use Cases
- Common Infrastructure
- Scaling Challenges

Recommendations at Netflix

- Personalize the Netflix experience for each member
  - **Goal**: Quickly help members find content they'd like to watch
  - **Risk**: Member may lose interest and abandon the service
  - **Challenge**: Recommending at scale

Scale @ Netflix

- 125M+ active members
- 190 countries
- 450B+ unique events/day
- 700+ Kafka topics

Typical Data Pipelines @ Netflix

- Data stored in Hive/S3
- Batch ETLs using Spark/Hive
- Table partitioning by day or hour
- Job scheduling by both CRON or data availability
- Latency often is on the order of days

The Need for Near Real Time (NRT)

- Dynamic catalog
- Growing member base
- Time sensitivity
  - Content popularity changes
  - Member interests evolve

The Need for Near Real Time (NRT)

- Increasing amount of data
    - Process data as soon as possible to keep latencies low
    - Minimize amount of data to reprocess in case of failure
- Multi-Armed Bandits Adoption

Use Cases

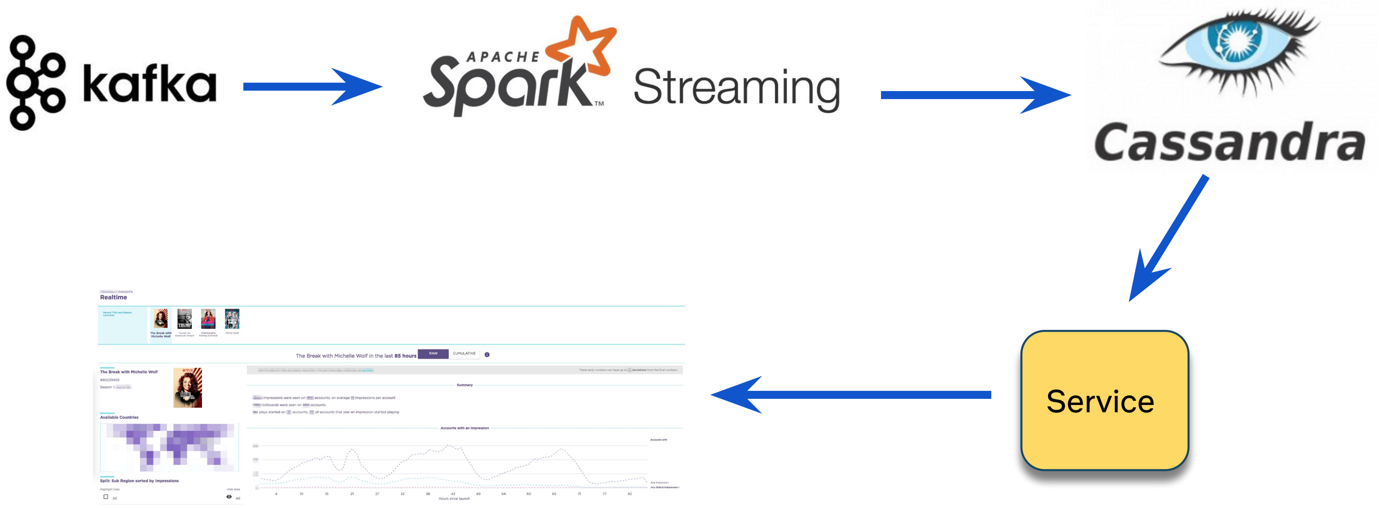
- Video Insights
- ML Pipelines for Recommendations

NRT for Video Insights

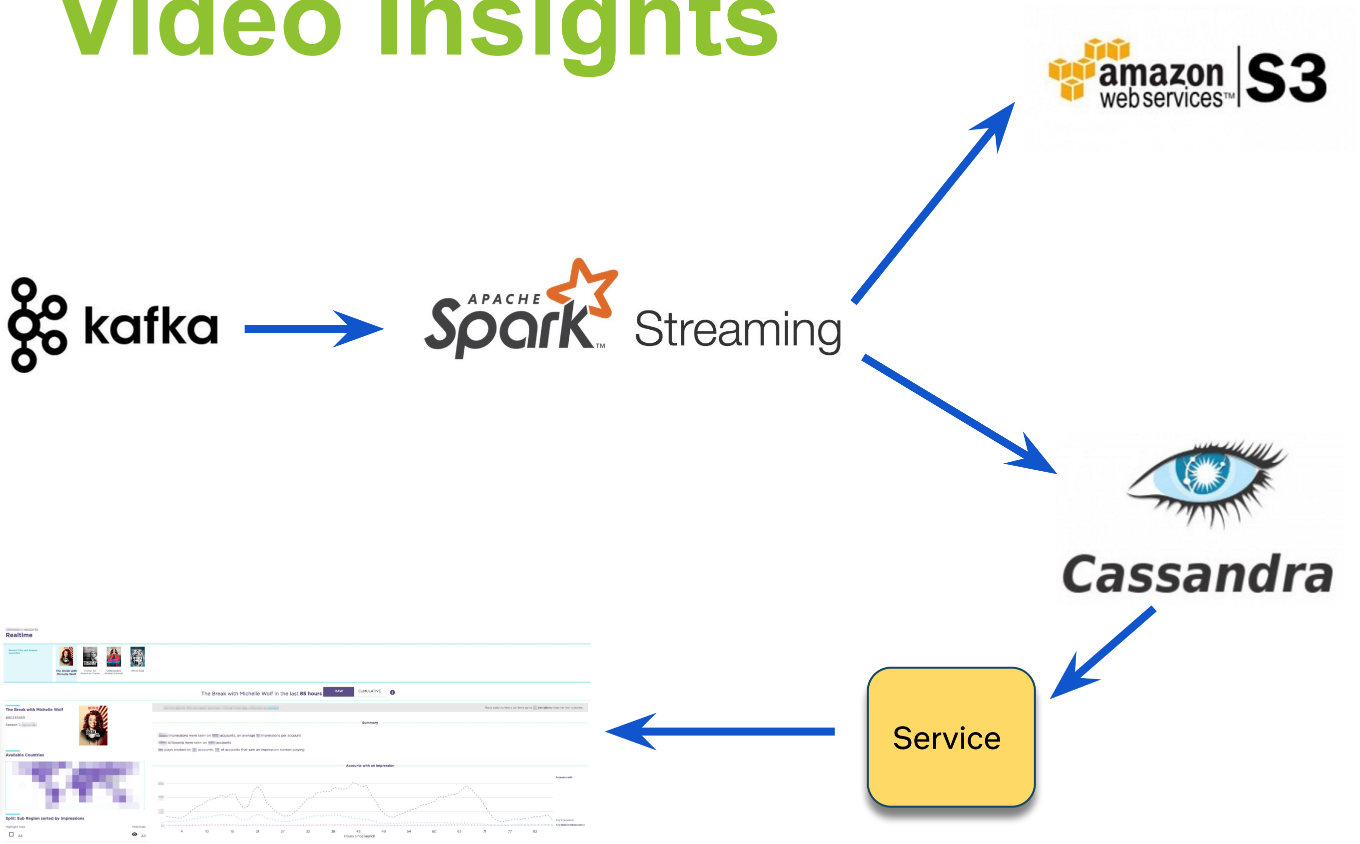
Video Insights

- New title launches
- Early reads on title performance
- Slice by various dimensions

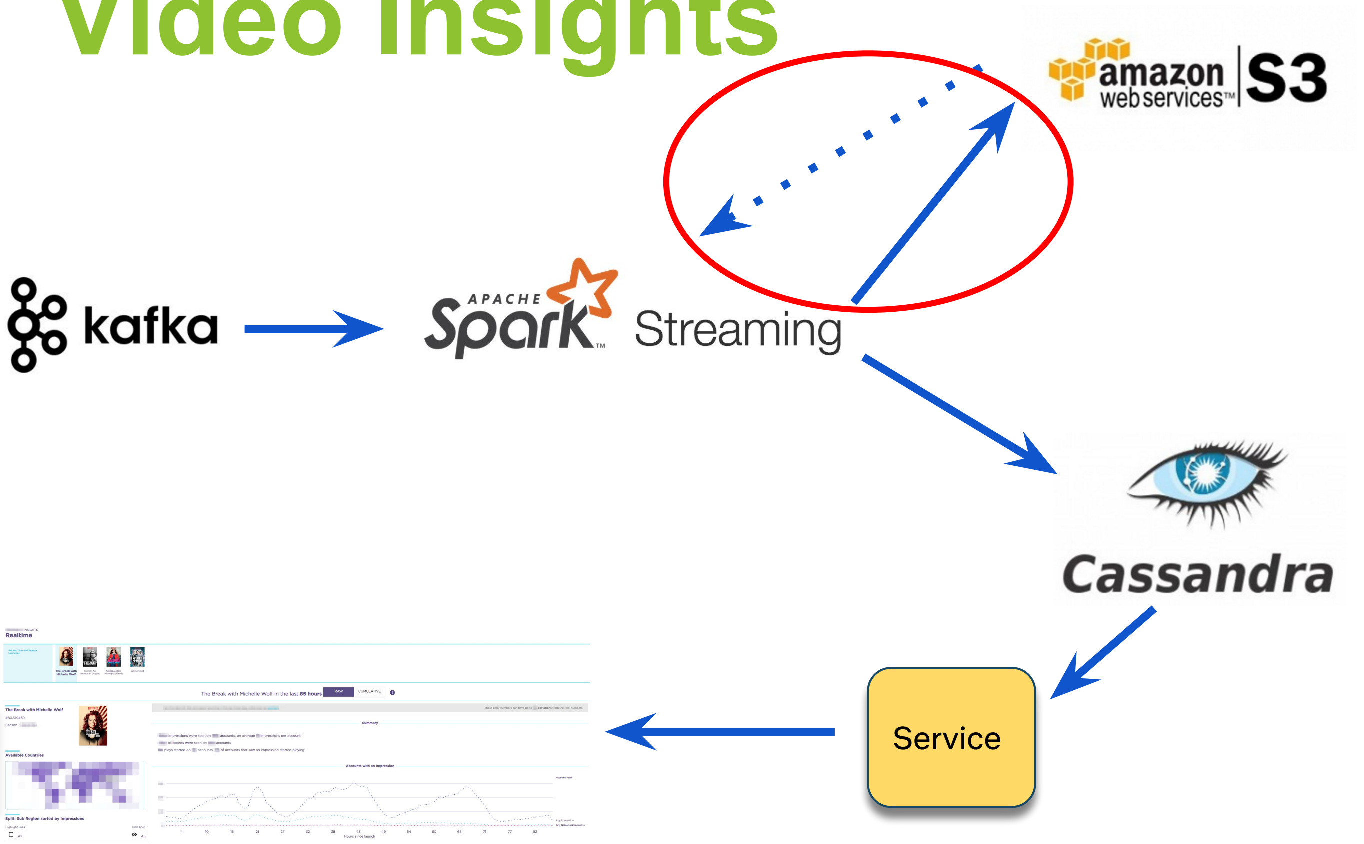
Video Insights



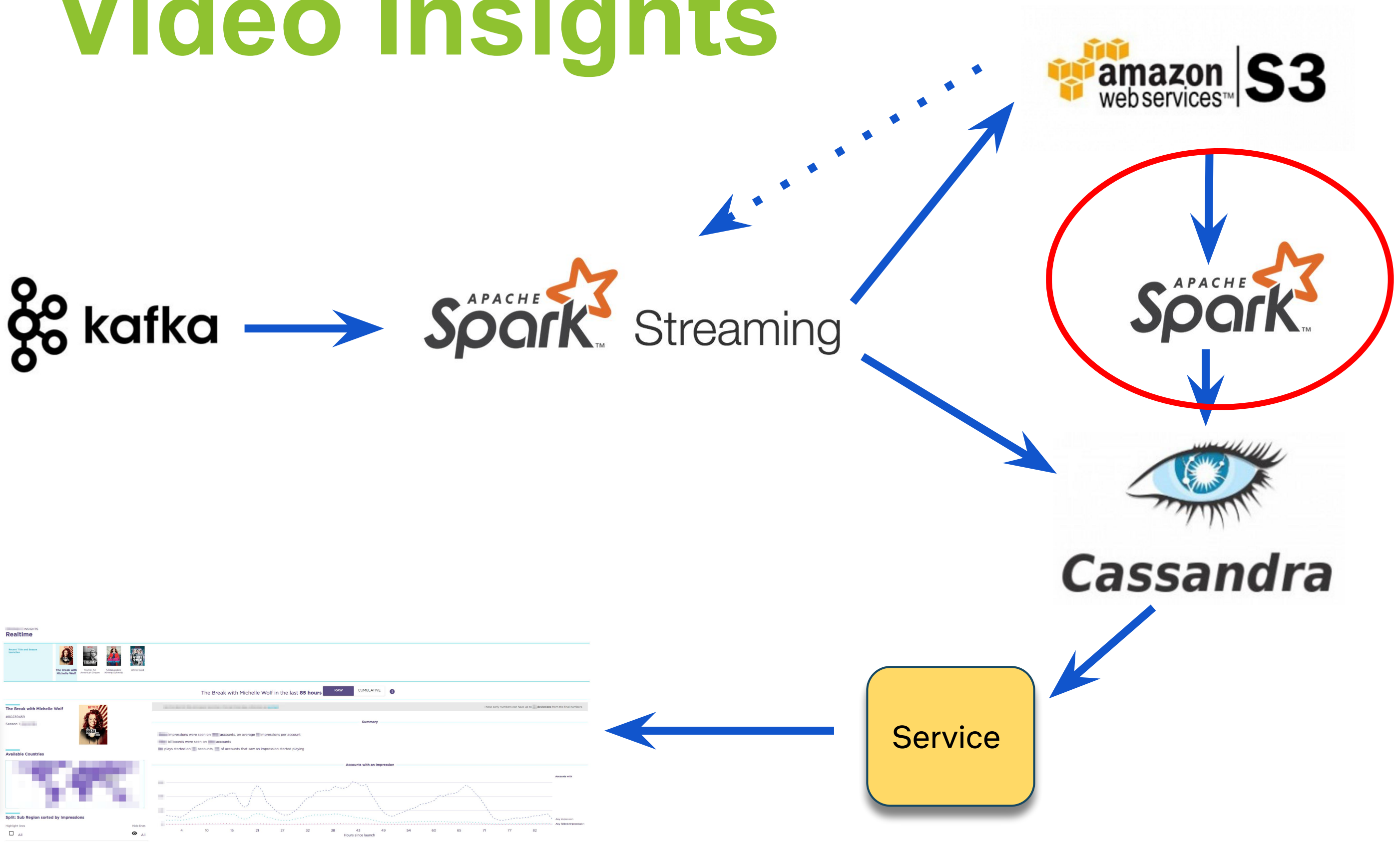
Video Insights



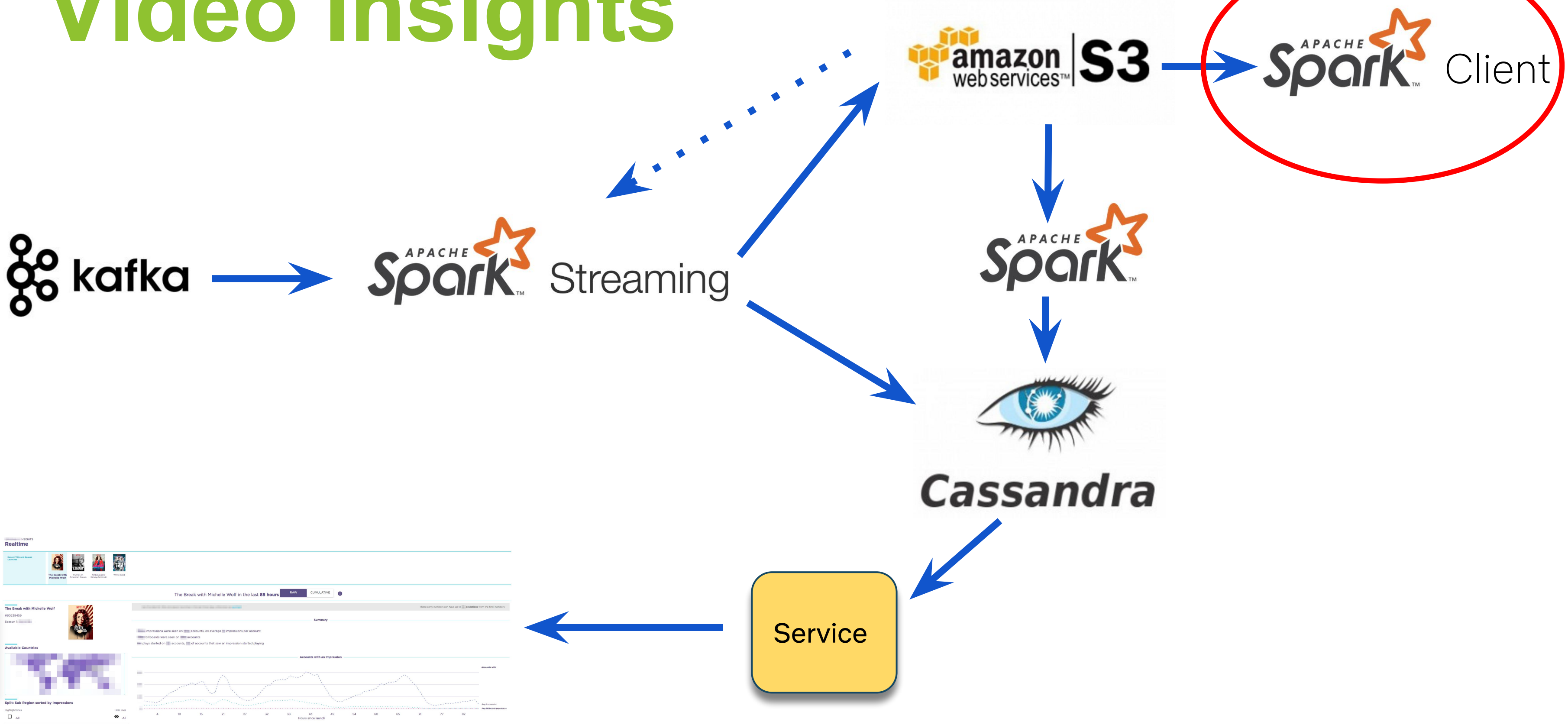
Video Insights



Video Insights



Video Insights



Video Insights - State

- Counts maintained in Spark
- Custom state management
    - Based on `mapWithState` implementation

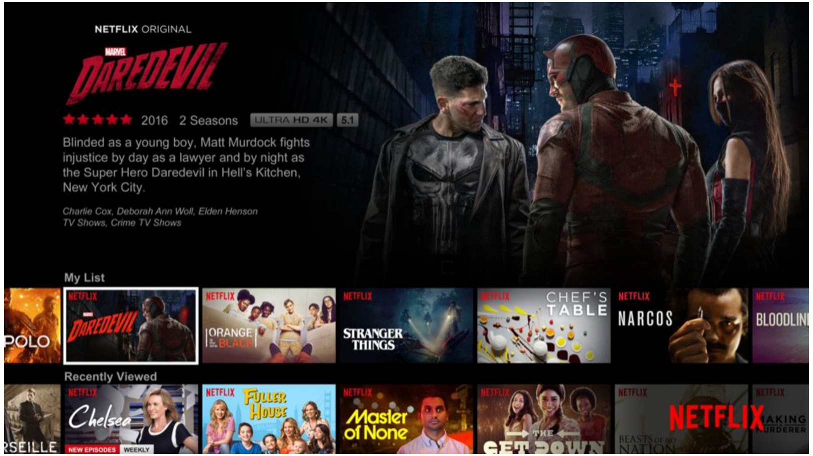
```
input.scan(initRDD)((currentRDD, batchRDD) => f(currentRDD, batchRDD))
```

    - Easier to re-use the function `f` in batch mode
    - Lower-level control over state management
    - `scanByKey` alternative for keyed state

NRT for Recommendations

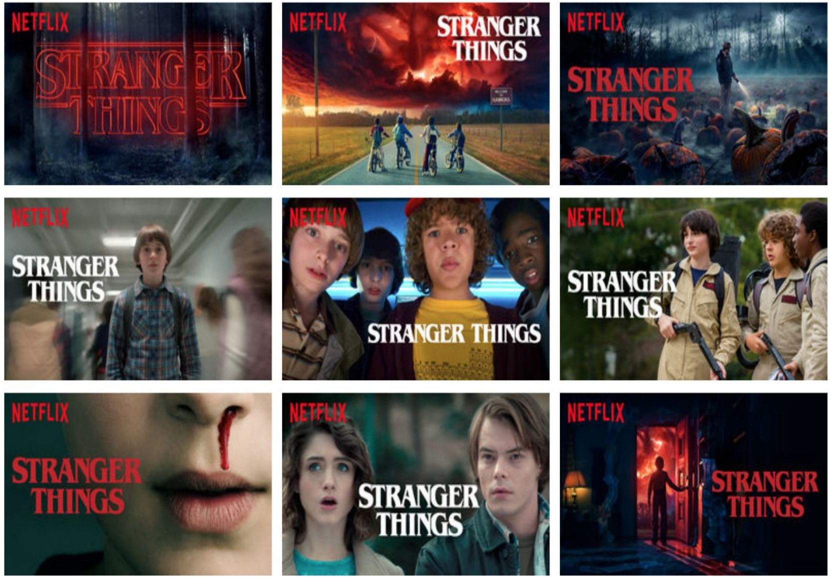
Billboard Recommendations

- Recommend a relevant title to each member
- Right time
- Respond quickly to member feedback

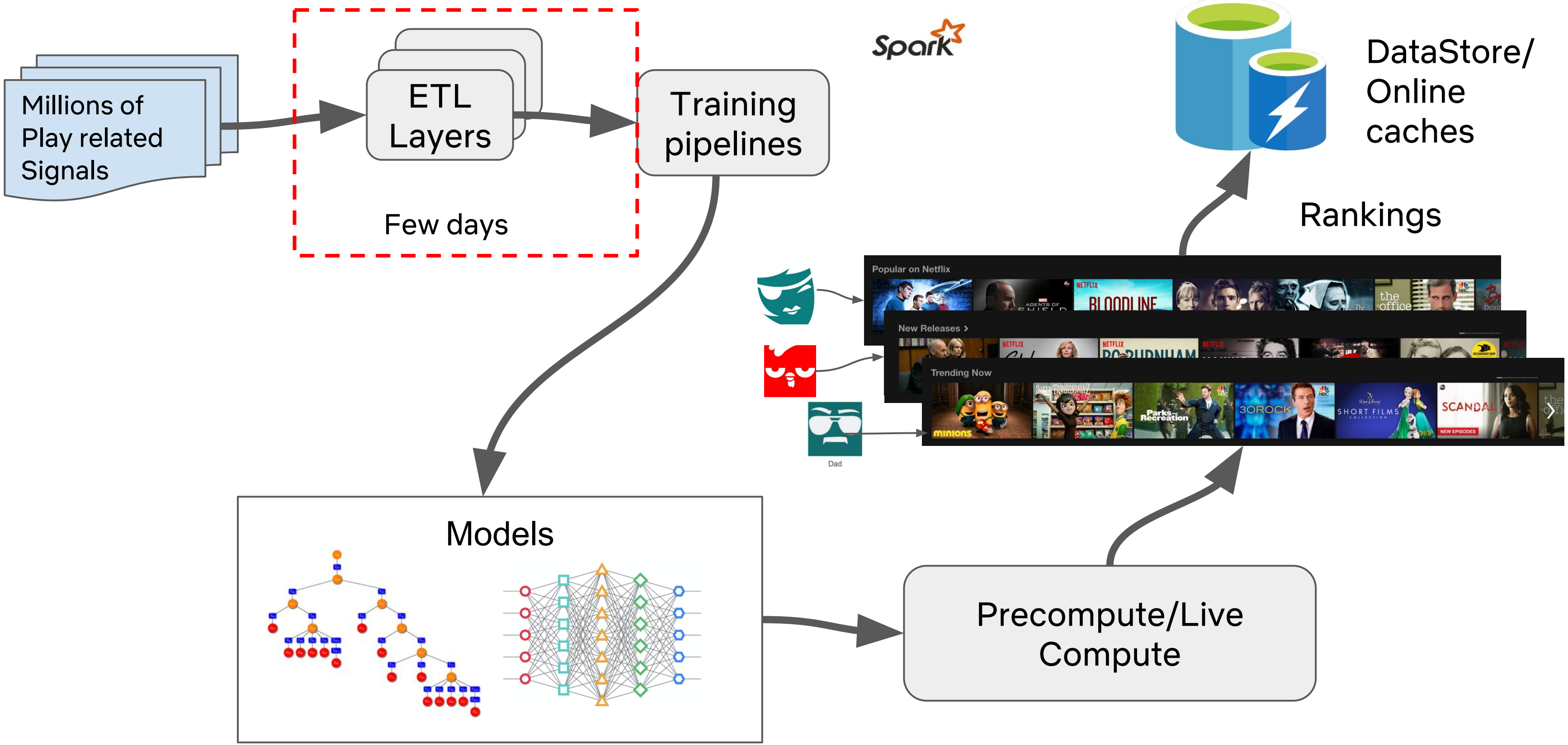


Artwork Personalization

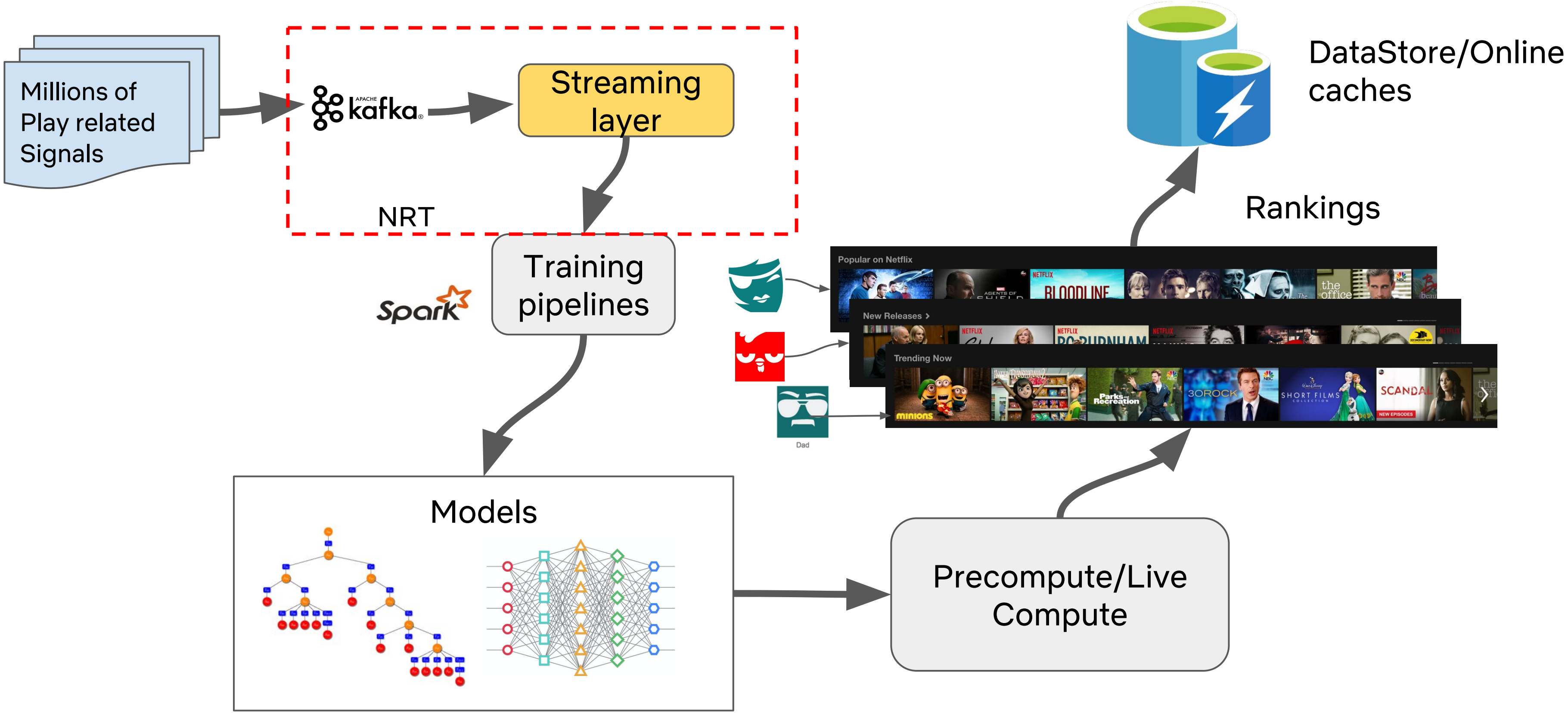
- Personalized Image
- Visual Evidence
- Quickly adapting - Title launches, member tastes
- Rapid learning - Cold start



Traditional Recommendations

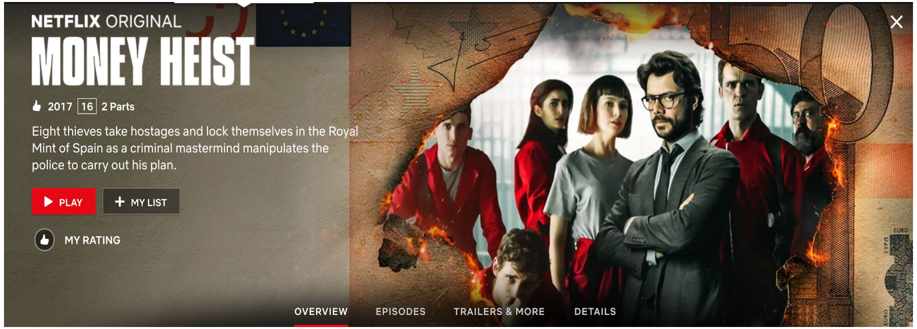


NRT Recommendations

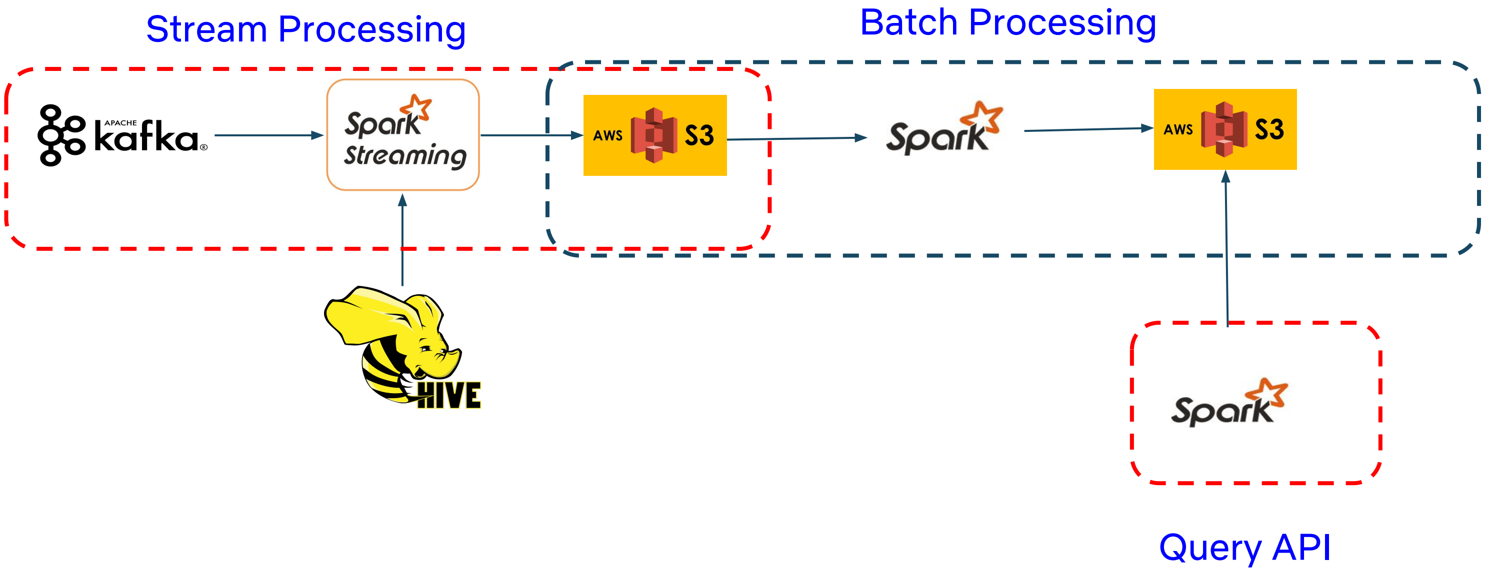


Required Data

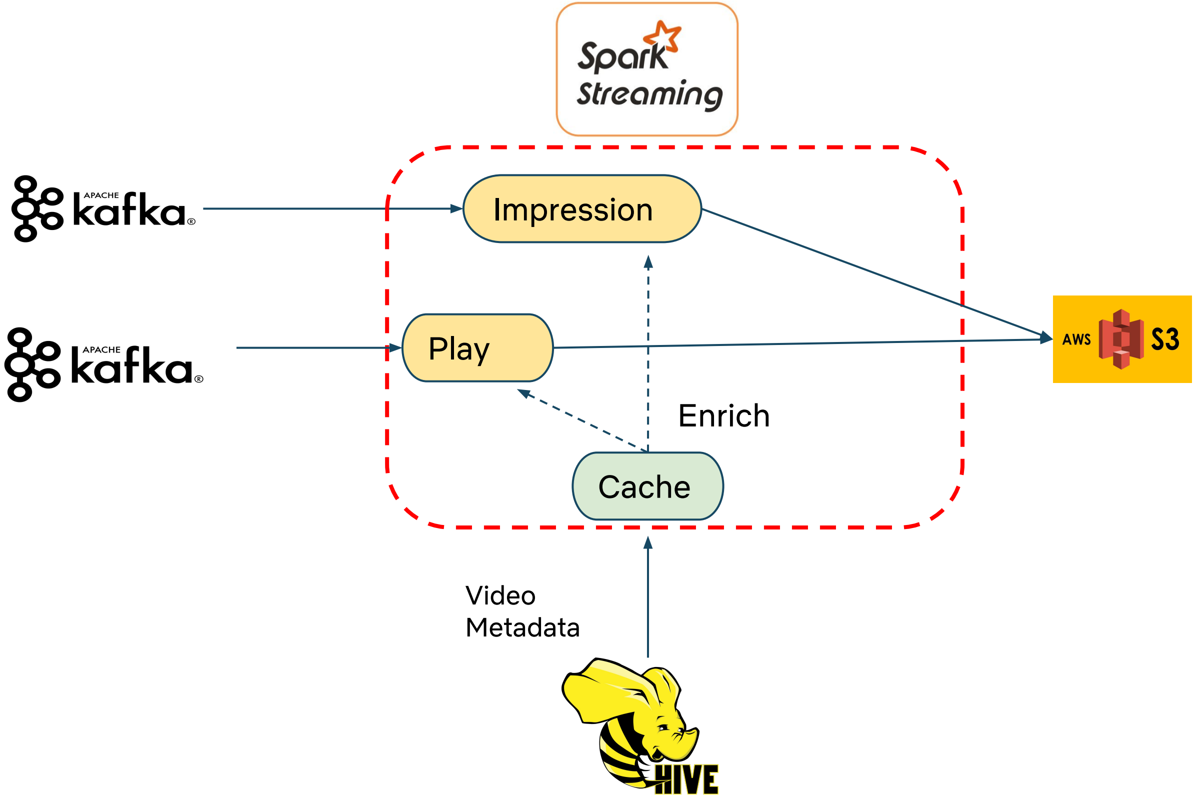
- Impressions, Plays, etc.
- Attribution
- Explore/Exploit Metadata



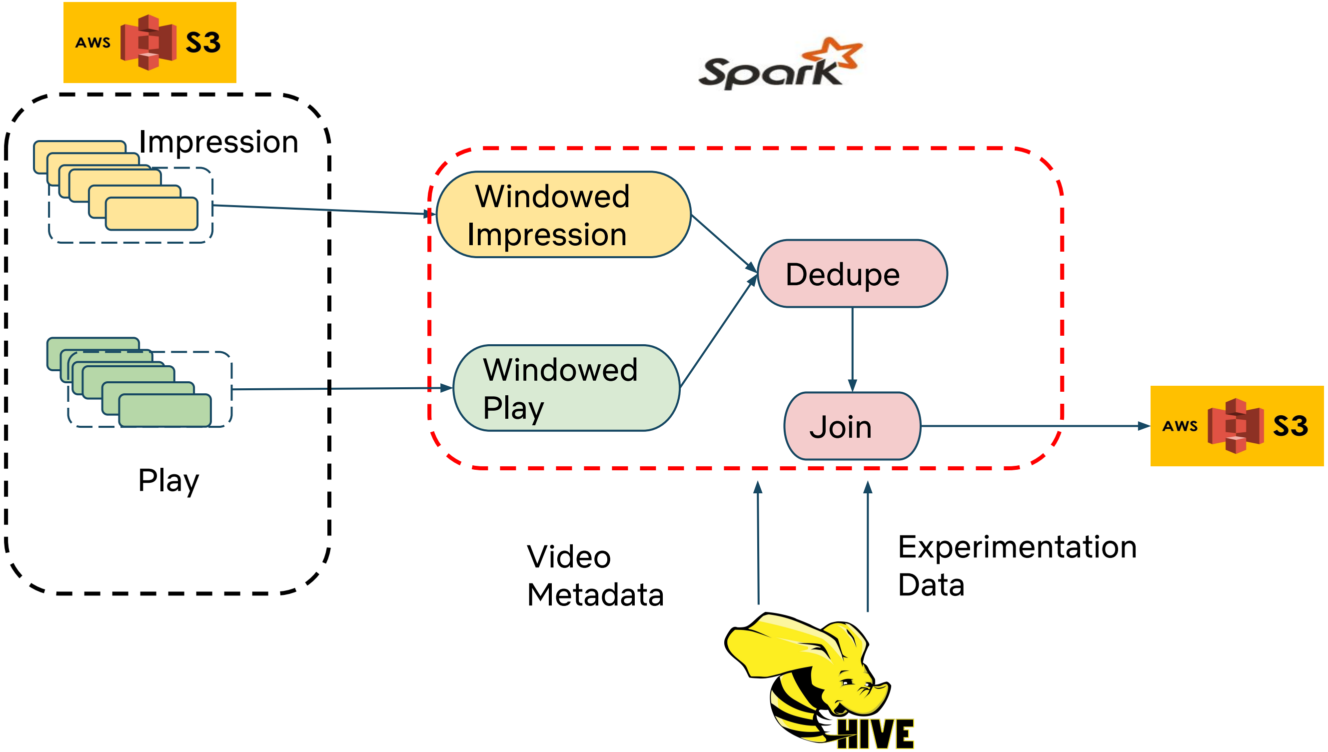
Attribution Infrastructure



Stream Processing - Zoomed in



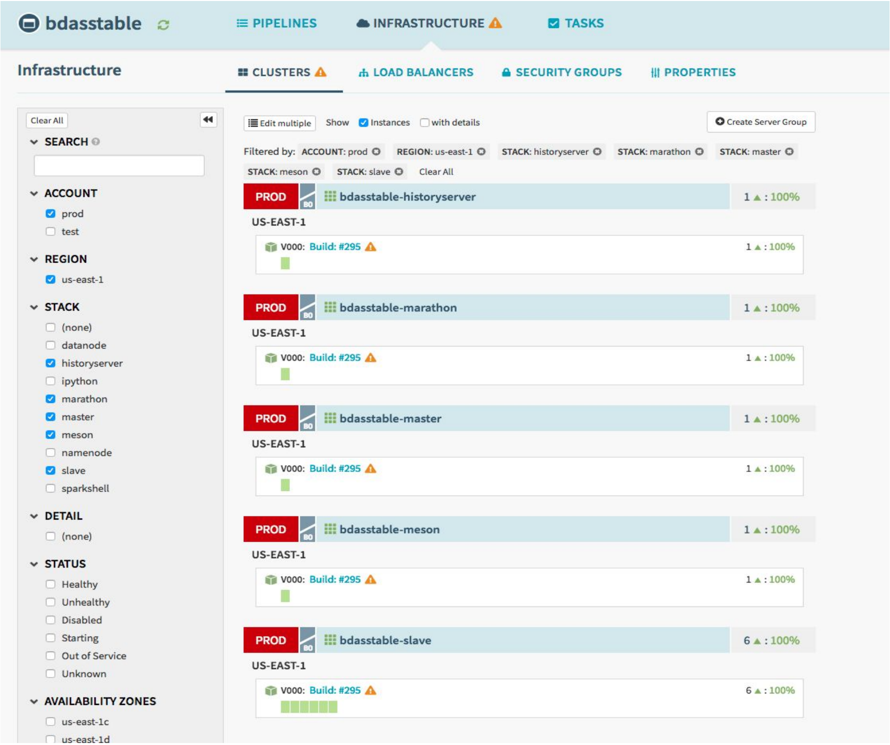
Batch Processing - Zoomed in



Common Infrastructure

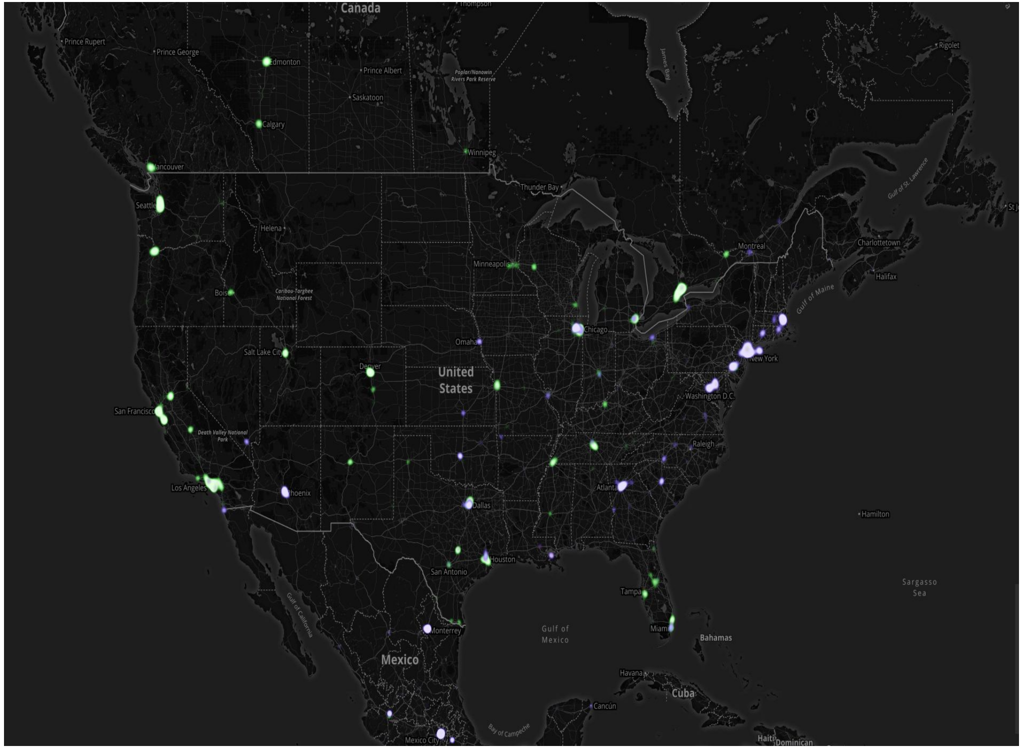
Netflix Spark Stack

- Jenkins
- Spinnaker
- ASG
- **Runners:** Marathon, Meson
- **Resource Manager:** Mesos
- **Storage:** HDFS, S3, EFS
- Multi-Region



Multi Region Challenges

- Geo routing
- Impression in one region; play in another
- Streaming - Multi Region
- Batch Reduce/Merge - One region



Can it withstand Chaos?

- Chaos is a design principle
- Instance Failovers => Region Failovers
- Transparent to the consumers
- Over provisioned
- Long term - Autoscale

When Things Go South

- What if something doesn’t look right?
  - Stream Processing is stuck
  - Driver/Executor failures
  - Intermittent issues with external dependencies

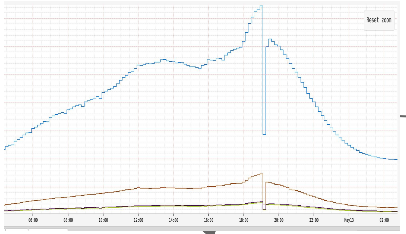
- Metrics - Spark metrics to Atlas (similar to RRDTool + Graphite)

- Getting paged at 2 am - Not fun :) !

- Need for auto-remediation - less operational overhead

Auto Remediation Infrastructure

Triggers



Metadata

SQS

De-queue

Auto Remediation

Spark

Streaming Challenges

- Scalability Performance tuning
  - Micro batch interval
  - Memory Tuning
  - Parallelism/Shuffle tradeoff
- Data quality issues
  - Low latency vs data quality