Self Healing in Streaming Systems
#UW Database Day
Dec 2nd, 2016

Karthik Ramasamy
Twitter

@karthikz
What is self healing?

A self healing system adapts itself as their environmental conditions change and continue to produce results.
Why?

Value of Data to Decision-Making

- **Time-critical Decisions**
- **Preventive/Predictive**
- **Actionable**
- **Reactive**
- **Historical**

**Information Half-Life In Decision-Making**

- Traditional “Batch” Business Intelligence

- Real-Time
  - Seconds
  - Minutes
  - Hours
  - Time
  - Days
  - Months

Why?

**LOSS OF REVENUE**
Impact of downtime popular event such as Super Bowl Oscars, etc

**SLA VIOLATIONS**
Impact of not honoring an SLA leading to penalty payments

**QUALITY OF LIFE**
Engineers & SRE burnt out attending to incidents

**INCREASED PRODUCTIVITY**
With reduced incidents, engineers can focus on actual development
Real-Time Budget

> 1 HOUR
high throughput

10 MS – 1 SEC
approximate

< 500 MS
latency sensitive

< 1 MS
low latency

BATCH
adhoc queries
monthly active users relevance for ads

NEAR REAL Time
ad impressions count hash tag trends

OLTP
deterministic workflows
fanout Tweets search for Tweets

REAL TIME
Financial Trading
Streaming Variants

STREAM WORKFLOWS
Continuous process of events for a deterministic workflow

STREAM PROCESSING
Analyze and act on data with continuous queries

STREAM ANALYTICS
Continuously analyze data using mathematical & statistical techniques
Heron Topology - Physical Execution

Spout 1 → Bolt 1 → Bolt 4

Spout 2 → Bolt 2 → Bolt 3 → Bolt 5
Heron

Scheduler

Topology Submission

Topology 1

Topology 2

Topology N
Heron Topology Components

Topology Master

ZK Cluster

Logical Plan, Physical Plan and Execution State

Sync Physical Plan

MASTER CONTAINER

Metrics Manager

Stream Manager

I₁ I₂ I₃ I₄

DATA CONTAINER

Stream Manager

Metrics Manager

I₁ I₂ I₃ I₄

DATA CONTAINER
Heron @Twitter

> 500 Real Time Jobs

500 Billions Events/Day PROCESSED

25-200 MS latency
Common Operational Issues

01: Slow Hosts

02: Network Issues

03: Data Skew
Slow Hosts

- Memory Parity Errors
- Impeding Disk Failures
- Lower GHZ
Stream Manager

Stream Manager

S1  B2
B3  B4

S1  B2
B3  B4

S1  B2
B3  B4

S1  B2
B3  B4

S1  B2
B3  B4

S1  B2
B3  B4

S1  B2
B3  B4

S1  B2
B3  B4

S1  B2
B3  B4
Spout Backpressure

Can we do better?
Network
Network Slowness

Delays processing

Data is accumulating

Timeliness of results is affected
Network Slowness

Can we do better?
Network Partitioning
Network Partitioning

- New Master Container
- Acquiring Mastership in ZooKeeper fails
- Master Container Dies
Network Partitioning

- Topology Master
- Stream Manager

TMaster thinks data container failed

Waits for the scheduler to reschedule new data container

Never happens
Network Partitioning

Cannot exchange data

Data accumulates

Chaos ensues!

Can we do better?
Network Partitioning

New data container spawned

TMaster realizes two data containers report as the same

Does not accept the new one and eventually it dies
Data Skew

Single Key
Single key maps into a instance and its count is high

Multiple Keys
Several keys map into single instance and their count is high
Data Skew - Multiple Keys

Spout 1 → Bolt 1 → Bolt 4
Spout 2 → Bolt 2 → Bolt 3 → Bolt 5
Data Skew - Single Key

What happens if the skew is temporary?
Rack Failures

- **Multiple Rack**: Outage of several racks simultaneously
- **Single Rack**: Outage of a single rack
Rack Local vs Rack Diversity

Rack Local:
- Advantage of reduced network latency and high p2p bandwidth

Rack Diversity:
- Higher network latency and shared bandwidth
Single Rack Failure

Rack Local

Rack failure
If job is rack local entire job is down and needs to be rescheduled

Impact on real timeliness
Depends on the application
Single Rack Failure

Rack Diversity
If job is rack diverse impact is minimal

Rack failure

Impact on real timeliness
Is affected less - depending on how many containers are running
Conclusion

Self healing is important in Streaming systems

Key operational issues - Slow hosts, network and data skew

These requirements make the streaming systems more complex
Interested in Heron?

HERON IS OPEN SOURCED

CONTRIBUTIONS ARE WELCOME!

https://github.com/twitter/heron

http://heronstreaming.io

FOLLOW US @HERONSTREAMING
Any Questions ???

WHAT  WHY  WHERE  WHEN  WHO  HOW
Thanks For Listening
@karthikz
## Detections and Resolutions

<table>
<thead>
<tr>
<th>Detections</th>
<th>Resolutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proactive</td>
<td>Affects real-time (depends on application tolerance)</td>
</tr>
<tr>
<td>Reactive</td>
<td>Full manual intervention</td>
</tr>
<tr>
<td>Lazy</td>
<td>Instant</td>
</tr>
</tbody>
</table>