Streaming in the Extreme

Jim Scott – Director, Enterprise Strategy & Architecture

@kingmesal #StrataHadoop
What is Extreme?
Quantification of “Extreme”

- 1+ Trillion Events  
  - per day

- Millions of Producers  
  - Billions of events per second

- Multiple Consumers  
  - Potentially for every event

- Multiple Data Centers  
  - Plan for success  
  - Plan for drastic failure

- Completely Secure
Logical Breakdown

• Messaging Persistence / Storage
  – HDFS, MapR-DB, etc…

• Messaging Platform
  – Kafka, Flume, ZeroMQ, etc…

• Streaming Engine
  – Apache (all TLP): Flink, Samza, Spark Streaming, Storm
  – There are others out there like Akka – Reactive Streaming

Note: Spark is not real-time, it is micro-batch and not event-based
Logical Dataflow

Messaging

Stream Processors
Analytics
Consumers
Putting Goals in Perspective

• This is an “as-it-happens” world

• On-demand TV, Taxi, Everything
  – Same thing expected in business

• If you don’t start moving in this direction you will be left behind

• IoT will bury you
Messaging Platforms
Messaging Platforms

• The weakest link in a chain is what causes failures

• This is the most important facet of a real-time platform
  – Including the underlying storage (cannot forget about that)

• A messaging platform can be leveraged by any code
  – More than just a source for a Stream Processing Engine
Message Delivery Semantics

• There are three general categories of *delivery patterns*:
  – *At-most-once*: messages may be lost. This is usually the least desirable outcome.
  – *At-least-once*: messages may be redelivered (no loss, but duplicates). This is good enough for many use cases.
  – *Exactly-once*: each message is delivered once and only once (no loss, no duplicates). This is a desirable feature although difficult to guarantee in all cases.

**Note**: These apply to Messaging Platforms and Streaming Engines

**Advice**: Make idempotent part of your vocabulary
Platform Options

- Apache Qpid, ZeroMQ or RabbitMQ

- Flume
  - NIGHTMARE to administer at serious scale

- Apache Kafka
  - Publish/subscribe model and currently about the best option
  - Decent API to work with
• Disaster Recovery
  – Can’t worry about losing an entire data center

• Distributed Queues
  – Not just queue to queue within a data center, but globally distributed
  – Seamless disconnection / reconnection for long hauls

• Security
  – Authentication, Wire-level encryption, Granular user access controls

• Administration
  – Needs to be simpler
  – Coexistence of use cases on a single cluster
Scenario

What if…

your web application used a ODBC / JDBC connection to Apache Drill to query the top thousand events from the messaging platform for graphing?

Think data center monitoring, financial services, etc…

Real-time access to data

* Not just for the stream processing engine
Scenario

What if…

you need to add 30% more capacity to your application stack to support future expected growth?

Think services, micro or otherwise…

Scaling without the pain

Decoupled communication is easier to scale
System metrics / logs go through the messaging platform
Stream Processing Engines
Stream Processing Engines

• This category seems to be the “super model” of real-time enablement

• Patterned framework enabling operations on events as they happen

• Very different architectures amongst them
The Landscape

• Streaming is Fundamentally Simple
  – Inputs -> Outputs
  – But it is WAY more complicated than this…

• Optimization can be complicated
  – i.e. Lots of configuration options

• This space is STILL very confused
  – Performance of different options is dependent upon source

• Lots of misinformation
  – e.g. performance comparisons that are not apples-to-apples
Apache Flink
Apache Flink

- Developed by dataArtisans
- Streaming First, Batch on Streaming

http://flink.apache.org/
Process Model and Component Stack

Client
Compiler/Optimizer
Submit Job

JobManager
Scheduling, Resource Management

TaskManager
Task Execution, Data Exchange

TaskManager
Task Execution, Data Exchange

Deploy Task
Send Status

Exchange Intermediate Results (shuffle / broadcast)

Operator DAG (type agnostic)
JobGraphs (generic data stream program)

Java Program
Scala Program

Flink Common API / Optimizer

Flink Runtime

Cluster Manager
Direct YARN EC2

Storage
Local Files HDFS S3 JDBC ...

© 2015 MapR Technologies
Flink Resources

- http://data-artisans.com/blog/
- http://flink.apache.org/blog/
- https://youtu.be/XgC6c4Wiqvs
Apache Samza
Apache Samza

- Developed by LinkedIn
- Distributed stream processing framework

http://samza.apache.org/
Samza Architecture

• Processing layer → Samza API

• Pluggable execution layer
  (default: YARN)

• Pluggable streaming layer
  (default: Kafka)

samza.apache.org/learn/documentation/latest/introduction/architecture.html
Samza Resources

- http://www.infoq.com/articles/linkedin-samza
Apache Spark Streaming
Apache Spark

• Developed at UC Berkeley’s AMP Lab
• Fast and general engine for large-scale data processing

http://spark.apache.org/
Operating Model

Standalone

YARN
Spark Resources

- [https://www.mapr.com/spark](https://www.mapr.com/spark)
- [https://sparkhub.databricks.com/](https://sparkhub.databricks.com/)
- [http://databricks.com/spark-training-resources](http://databricks.com/spark-training-resources)
- [http://oreilly.com/go/sparkcert](http://oreilly.com/go/sparkcert)
Apache Storm
Apache Storm

• Developed by Nathan Marz at Backtype/Twitter
• Distributed, real time computation system

http://storm.apache.org/
Operating Model – Storm & Trident

- High-level abstraction processing library on top of Storm
- Rich API with joins, aggregations, grouping, etc.
- Provides stateful, exactly-once processing primitives

storm.apache.org/documentation/Trident-tutorial.html
Storm Resources

- https://www.udacity.com/course/ud381
- http://www.manning.com/sallen/
- https://github.com/tdunning/storm-counts
Streaming Engine Performance Comparisons
“The truth is out there.”

– Spock
“Don’t believe everything you read on the Internet.”

– Abraham Lincoln
O’Reilly Office Hours!

Today

(starting in less than 2 hours)

5:25pm until 6:05pm
Free on-demand Hadoop training leading to certification

Start becoming an expert now
mapr.com/training
MapR already offers FREE, interactive on-demand training...

Now a FREE, interactive Spark ebook—with real code, running in real-time.

Book signing from 10:50a to 11:30a

Download now ebooks.mapr.com/spark
Ted Dunning & Ellen Friedman
Newest O’Reilly Book
“Sharing Big Data Safely”

Available at MapR Booth 519
Complimentary books & author signings
Announcing
The First In-Hadoop Document Database

Visit Booth 519 for a Live Demo
Q&A
Engage with us!

@kingmesal  maprtech

MapR Technologies  +MapRTechnologies

jscott@mapr.com  maprtech