Chicago Apache Flink Meetup

June 30th, 2015

Overview of Apache Flink: Next-Gen Big Data Analytics Framework

By Slim Baltagi @SlimBaltagi

Agenda

I. What is Apache Flink stack and how it fits into the Big Data ecosystem?

II. How Apache Flink integrates with Apache Hadoop and other open source tools?

III. Why Apache Flink is an alternative to Apache Hadoop MapReduce, Apache Storm and Apache Spark?

IV. Who is using Apache Flink?

V. Where to learn more about Apache Flink?

VI. What are some Key Takeaways?

I. What is Apache Flink stack and how it fits into the Big Data ecosystem?

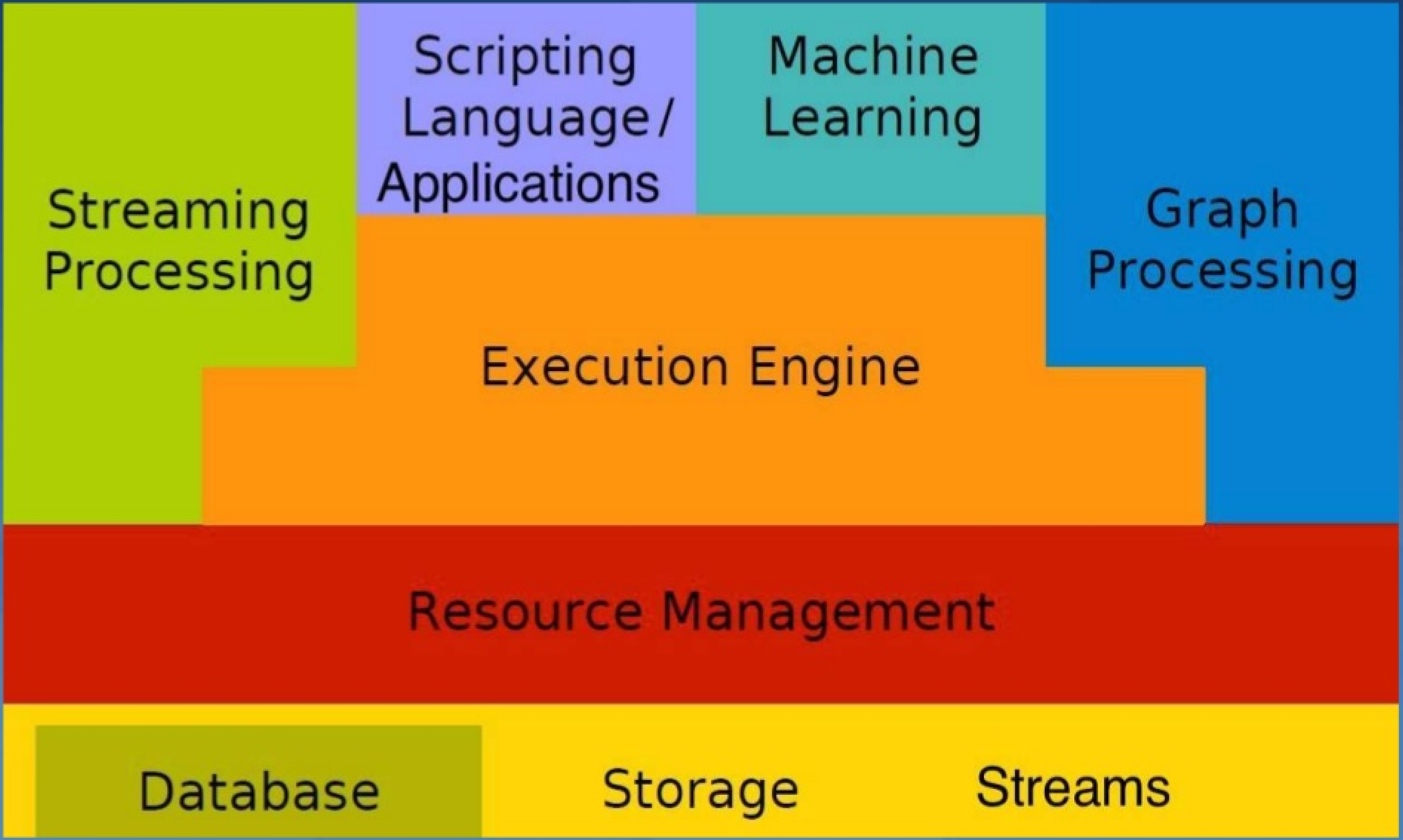
1. What is Big Data?
2. What is a typical Big Data Analytics Stack?
3. What is Apache Flink?
4. What is Flink Execution Engine?
5. What are Flink APIs?
6. What are Flink Domain Specific Libraries?
7. How Flink offers Interactive Data Analysis?
8. What is Flink Architecture?
9. What is Flink Programming Model?

1. What is Big Data?

"Big Data refers to data sets large enough and data streams fast enough, from heterogeneous data sources, that has outpaced our capability to store, process, analyze, and understand."

"THAT'S your Ark for the Big Data flood? Noah, you will need a lot more storage space!"

2. What is a typical Big Data Analytics Stack: Hadoop, Spark, Flink, …?



3. What is Apache Flink?

- Apache Flink, like Apache Hadoop and Apache Spark, is a community-driven open source framework for distributed Big Data Analytics. Apache Flink engine exploits data streaming and in-memory processing and iteration operators to improve performance.
- Apache Flink has its origins in a research project called Stratosphere of which the idea was conceived in 2008 by professor Volker Markl from the Technische Universität Berlin in Germany.
- In German, Flink means agile or swift. Flink joined the Apache incubator in April 2014 and graduated as an Apache Top Level Project (TLP) in December 2014.

3. What is Apache Flink?

The Apache Flink framework, written in Java, provides:

1. Big data processing engine: distributed and scalable streaming dataflow engine
2. Several APIs in Java/Scala/Python:
   - DataSet API – Batch processing
   - DataStream API – Real-Time streaming analytics
   - Table API - Relational Queries
3. Domain-Specific Libraries:
   - FlinkML: Machine Learning Library for Flink
   - Gelly: Graph Library for Flink
4. Shell for interactive data analysis

Key Vision of Apache Flink

Draws on concepts from MPP Database Technology

- Declarativity
- Query optimization
- Efficient parallel in-memory and out-of-core algorithms

Add

- Streaming
- Iterations
- Advanced Dataflows
- General APIs

Draws on concepts from Hadoop MapReduce Technology

- Massive scale-out
- User Defined Functions
- Complex data types
- Schema on read

What is Apache Flink stack?

APIs & LIBRARIES

Hadoop M/R | Table | Gelly | ML | Dataflow | MRQL | Cascading (WiP) | Zeppelin

Table | SAMOA | Dataflow (WiP)

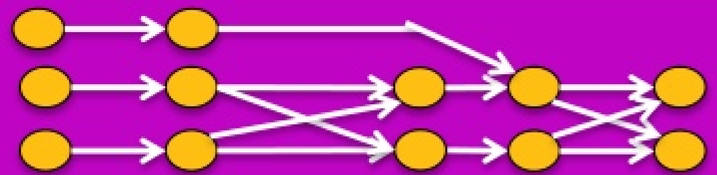
DataSet (Java/Scala/Python)
Batch Processing

DataStream (Java/Scala)
Stream Processing

SYSTEM

Batch Optimizer

Stream Builder



Runtime
Distributed Streaming
Dataflow

DEPLOY

Local
Single JVM
Embedded

Cluster
Standalone
YARN, Tez

Cloud
Google's GCE
Amazon's EC2

STORAGE

Files
Local
HDFS
S3
Tachyon

Databases
MongoDB
HBase
SQL
…

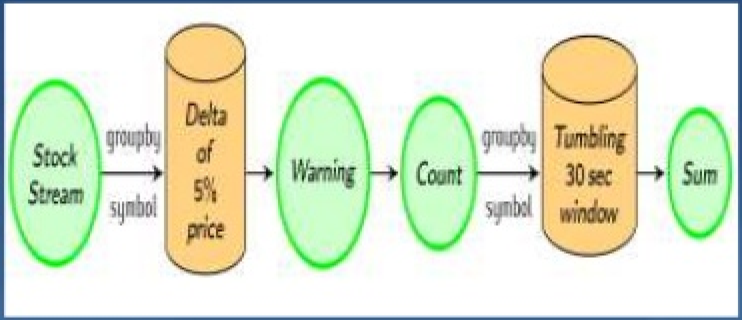
Streams
Flume
Kafka
RabbitMQ
…

4. What is Flink Execution Engine?

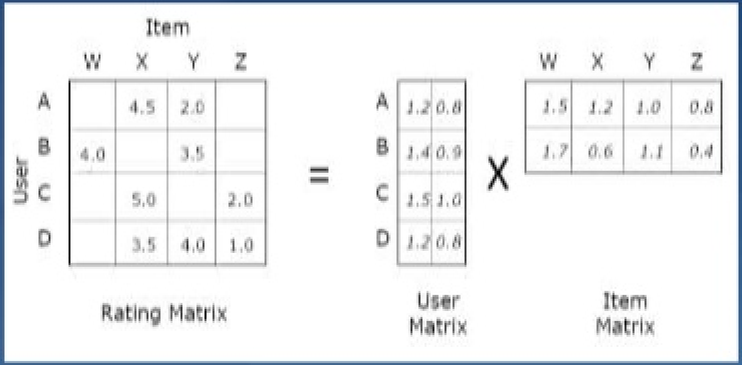
The core of Flink is a distributed and scalable streaming dataflow engine with some unique features:

1. True streaming capabilities: Execute everything as streams
2. Native iterative execution: Allow some cyclic dataflows
3. Handling of mutable state
4. Custom memory manager: Operate on managed memory
5. Cost-Based Optimizer: for both batch and stream processing

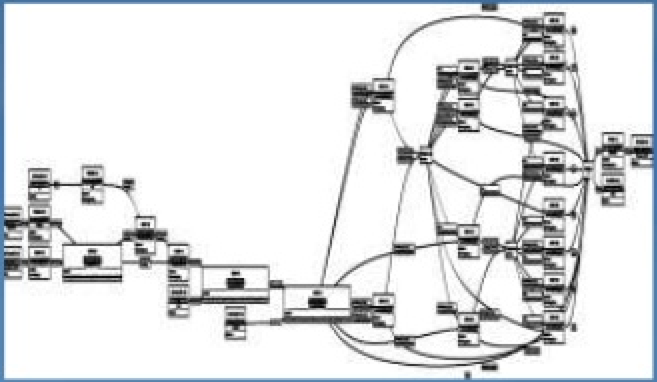
The only hybrid (Real-Time Streaming + Batch) open source distributed data processing engine supporting many use cases:



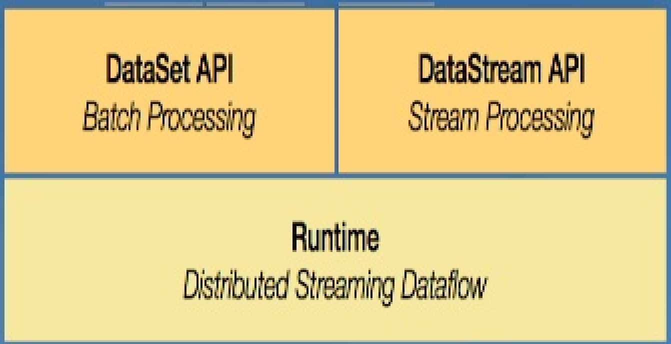
Real-Time stream processing

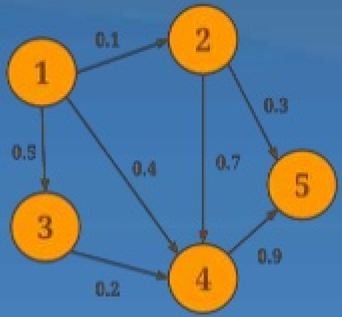


Machine Learning at scale



Batch Processing





Graph Analysis

Why Apache Flink is Next-Gen?

hadoop MapReduce	TEZ	Spark	Apache Flink
• Batch	• Batch • Interactive	• Batch • Interactive • Near-Real time Streaming • Iterative processing	• Batch • Interactive • **Real-Time Streaming** • **Native Iterative** processing
MapReduce	**D**irect **A**cyclic **G**raphs (DAG) Dataflows	RDD: **R**esilient **D**istributed **D**atasets	**Cyclic Dataflows**
1st Generation (1G)	2nd Generation (2G)	3rd Generation (3G)	4th Generation (**4G**)

5. Flink APIs

5.1 DataSet API for static data - Java, Scala, and Python

5.2 DataStream API for unbounded real-time streams - Java and Scala

5.3 Table API for relational queries - Java and Scala

5.1 DataSet API – Batch processing

```
case class Word (word: String, frequency: Int)
```

DataSet API (batch):

```
val lines: DataSet[String] = env.readTextFile(...)
lines.flatMap {line => line.split(" ")
                                  .map(word => Word(word,1))}
      .groupBy("word").sum("frequency")
      .print()
```

DataStream API (streaming):

```
val lines: DataStream[String] = env.fromSocketStream(...)

lines.flatMap {line => line.split(" ")
                                  .map(word => Word(word,1))}
      .window(Time.of(5,SECONDS)).every(Time.of(1,SECONDS))
      .groupBy("word").sum("frequency")
      .print()
```

5.2 DataStream API – Real-Time Streaming Analytics

- Many time-critical applications need to process large streams of live data and provide results in real-time. For example:
  - Fraud detection
  - Financial Stock monitoring
  - Anomaly detection
  - Traffic management applications
  - Online recommenders
- Flink Streaming provides high-throughput, low-latency stateful stream processing system with rich windowing semantics. It has built-in connectors to many data sources like Flume, Kafka, Twitter, RabbitMQ

5.2 DataStream API – Real-Time Streaming Analytics

- Still in Beta as of June 24th 2015 (Flink 0.9 release)
- Data streams can be transformed and modified using high-level functions similar to the ones provided by the batch processing API.
- Flink Streaming provides native support for iterative stream processing.
- Streaming Fault-Tolerance added in Flink 0.9 (released on June 24th , 2015) allows Exactly-once processing delivery guarantees for Flink streaming programs that analyze streaming sources persisted by Apache Kafka. See paper: 'Lightweight Asynchronous Snapshots for Distributed Dataflows' http://arxiv.org/pdf/1506.08603v1.pdf June 28, 2015

5.2 DataStream API – Real-Time Streaming Analytics

- **Data Streaming Fault Tolerance document:** http://ci.apache.org/projects/flink/flink-docs-master/internals/stream_checkpointing.html
- **Flink being based on a pipelined execution engine akin to parallel database systems allows:**
  - to integrate streaming operations with rich windowing semantics seamlessly
  - process streaming operations in a pipelined way with lower latency than micro-batch architectures and without the complexity of lambda architectures.
- **Flink Streaming web resources at the Flink Knowledge Base** http://sparkbigdata.com/component/tags/tag/49-flink-streaming

5.3 Table API – Relational Queries

Table API (queries)

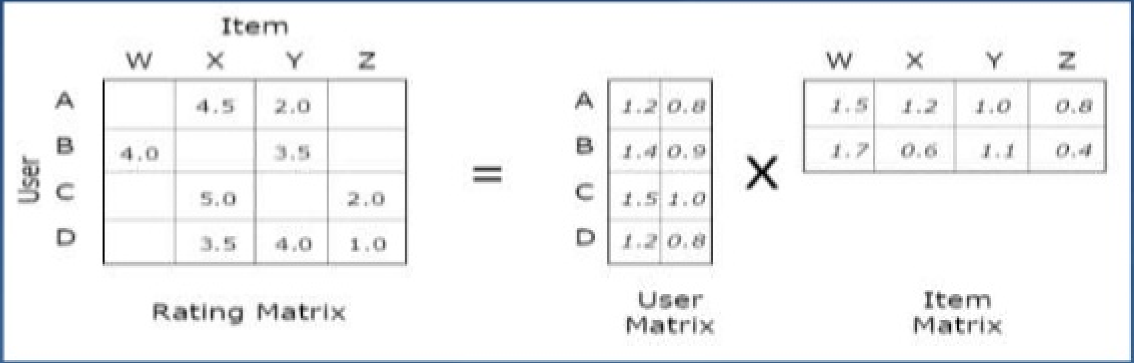
```
val customers = envreadCsvFile(…).as('id, 'mktSegment)
      .filter("mktSegment = AUTOMOBILE")

val orders = env.readCsvFile(…)
      .filter( o =>
dateFormat.parse(o.orderDate).before(date) )
      .as("orderId, custId, orderDate, shipPrio")

val items = orders
      .join(customers).where("custId = id")
      .join(lineitems).where("orderId = id")
      .select("orderId, orderDate, shipPrio,
          extdPrice * (Literal(1.0f) - discount) as
revenue")

val result = items
      .groupBy("orderId, orderDate,  shipPrio")
      .select("orderId, revenue.sum, orderDate, shipPrio")
```

5.3 Table API – Relational Queries

- Table API added in February 2015. Still in Beta as of June 24th 2015 (Flink 0.9 release)
- Flink provides Table API that allows specifying operations using SQL-like expressions instead of manipulating DataSet or DataStream.
- Table API can be used in both batch (on structured data sets) and streaming programs (on structured data streams).http://ci.apache.org/projects/flink/flink-docs-master/libs/table.html
- Flink Table web resources at the Apache Flink Knowledge Base: http://sparkbigdata.com/component/tags/tag/52-flink-table

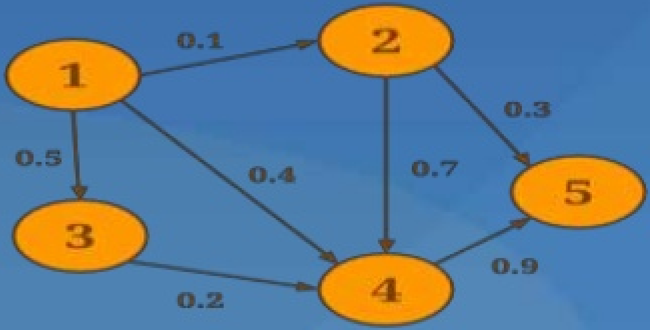
6. Flink Domain Specific Libraries

6.1 FlinkML – Machine Learning Library



$$
\begin{array}{c|cccc}
 & W & X & Y & Z \\
\hline
A & & 4.5 & 2.0 & \\
B & 4.0 & & 3.5 & \\
C & & 5.0 & & 2.0 \\
D & & 3.5 & 4.0 & 1.0
\end{array}
=
\begin{array}{c|cc}
A & 1.2 & 0.8 \\
B & 1.4 & 0.9 \\
C & 1.5 & 1.0 \\
D & 1.2 & 0.8
\end{array}
\times
\begin{array}{cccc}
W & X & Y & Z \\
\hline
1.5 & 1.2 & 1.0 & 0.8 \\
1.7 & 0.6 & 1.1 & 0.4
\end{array}
$$

6.2 Gelly – Graph Analytics for Flink



1
2
3
4
5
0.1
0.5
0.4
0.7
0.3
0.2
0.9