Next-Gen Big Data Analytics using the Spark stack

Jason Dai

Chief Architect of Big Data Technologies

Software and Services Group, Intel

Agenda

Overview

- Apache Spark stack
- Next-gen big data analytics
- Our vision and efforts

Highlights of our work on Spark stack

- Reliability of Spark Streaming
- SQL processing on Spark
- Spark Stream-SQL
- Tachyon hierarchical storage
- Analytics & SparkR
- …

Next-Gen Big Data Analytics

Volume

- Massive scale & exponential growth

Variety

- Multi-structured, diverse sources & inconsistent schemas

Value

- Simple (SQL) – descriptive analytics
- Complex (non-SQL) – predictive analytics

Velocity

- Interactive – the speed of thought
- Streaming/online – drinking from the firehose

The Spark stack

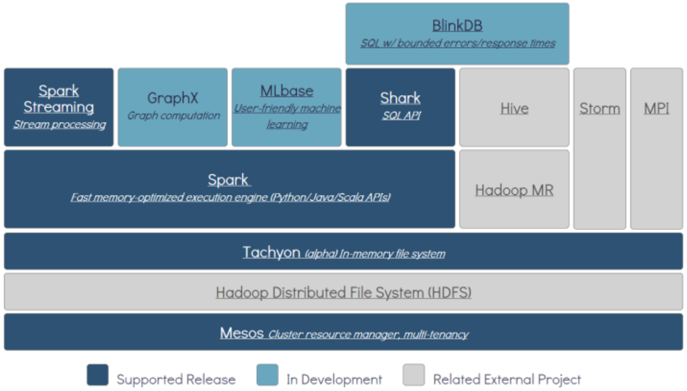
Project Overview

Research & open source projects initiated by AMPLab in UC Berkeley

Intel closely collaborating with AMPLab & community on open source development

- Started in 2012 when still a research project
- Intel among top 3 Spark contributors
  - Multiple Intel committers since the start of the project
  - Many critical contributions
    - Netty based shuffle, FairScheduler, metrics system, “yarn-client” mode, …

BDAS: Berkeley Data Analytics Stack
(Ref: https://amplab.cs.berkeley.edu/software/)


BlinkDB
SQL w/ bounded errors/response times
Spark Streaming
Stream processing
GraphX
Graph computation
MLbase
User-friendly machine learning
Shark
SQL API
Hive
Storm
MPI
Spark
Fast memory-optimized execution engine (Python/Java/Scala APIs)
Hadoop MR
Tachyon (alpha) In-memory file system
Hadoop Distributed File System (HDFS)
Mesos Cluster resource manager, multi-tenancy
Supported Release
In Development
Related External Project

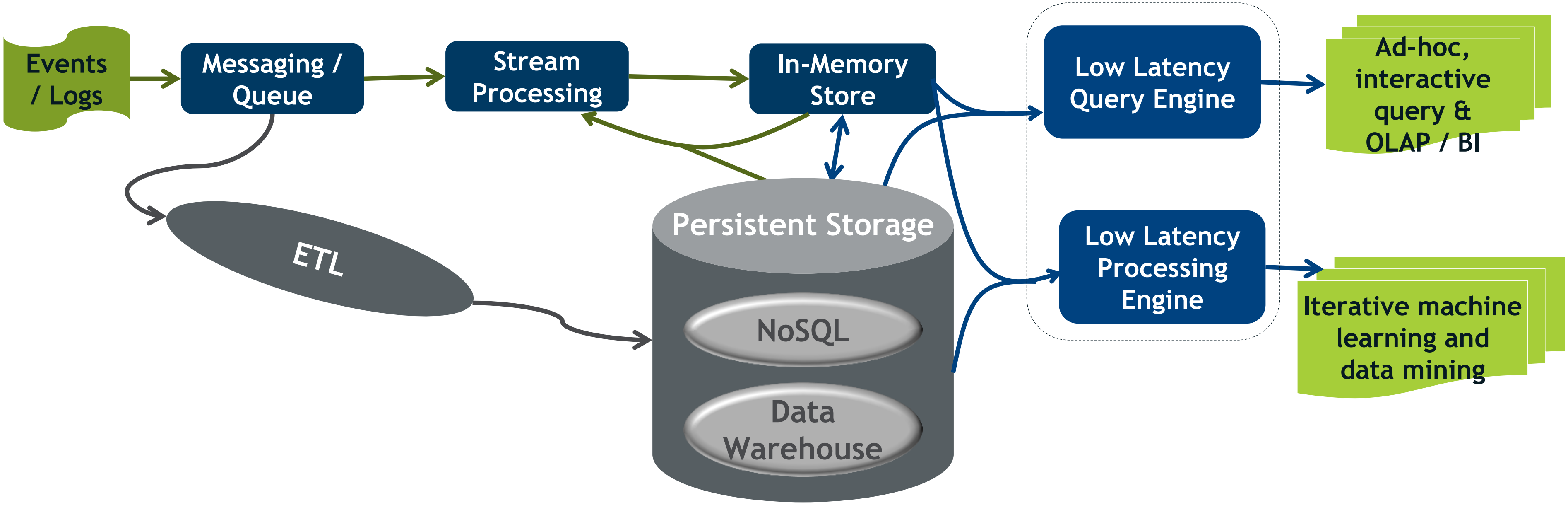
Continuous collaborations with AMPLab on new innovations in BDAS

- Major industry partners for Tachyon, GraphX & other machine learning efforts

Next Generations of Big Data Analytics

Next-gen big data analytics paradigm

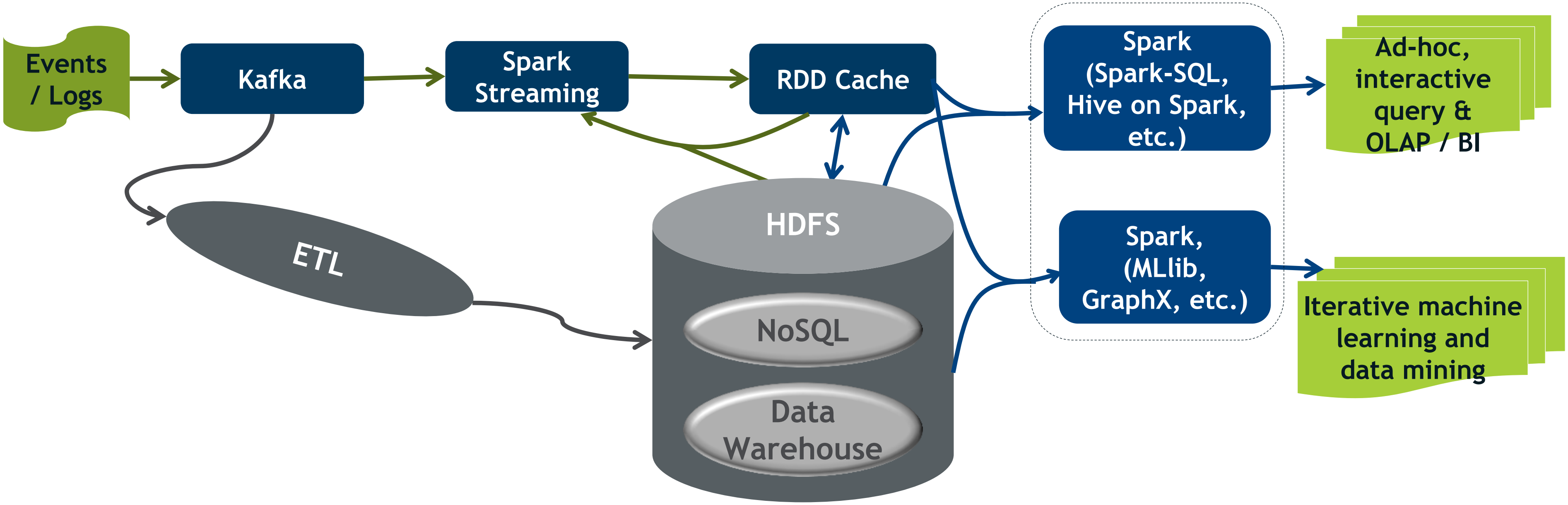
- Data captured & processed in a (semi-) real-time / streaming fashion
- Data mined using SQL queries as well as large-scale advanced analytics (e.g., machine learning, graph analysis, etc.)
- Interactive and iterative computations leveraging distributed in-memory data cache



Next-Gen Big Data Analytics using the Spark stack

Build next-gen Big Data analytics with the Spark stack

- *Analysis on streaming data*: “real-time” decisions
- *Interactive analysis*: faster decisions
- *Advanced analytics*: “better” decisions


Events / Logs
Kafka
Spark Streaming
RDD Cache
Spark (Spark-SQL, Hive on Spark, etc.)
Ad-hoc, interactive query & OLAP / BI
ETL
HDFS
NoSQL
Data Warehouse
Spark, (MLlib, GraphX, etc.)
Iterative machine learning and data mining

For more details, see our talk “*Real-Time Analytical Processing (RTAP) using Spark and Shark*” at Strata Conference + Hadoop World NYC 2013

Agenda

Overview

- Apache Spark stack
- Next-gen big data analytics
- Our vision and efforts

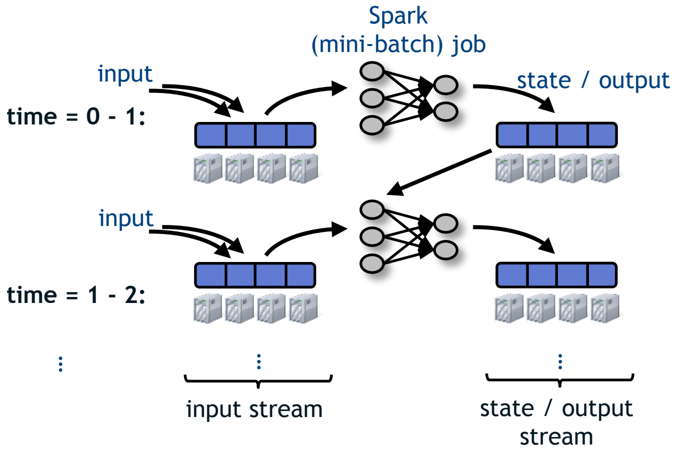
Highlights of our work on Spark stack

- Reliability of Spark Streaming
- SQL processing on Spark
- Spark Stream-SQL
- Tachyon hierarchical storage
- Analytics & SparkR
- …

Spark Streaming

Spark Streaming: discrete stream (DStream)

- Run streaming computation as a series of very small, deterministic (mini-batch) Spark jobs
  - As frequent as ~1/2 second
- Better fault tolerance, straggler handling & state consistency



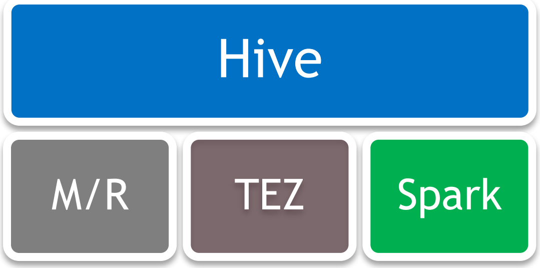
High reliability of Spark Streaming

- Collaborations between Databricks, Cloudera and Intel
- Improving Spark driver reliability through WAL
  - https://issues.apache.org/jira/browse/SPARK-3129
- Improving Kafka receiver reliability by better managing Kafka offset tacking & WAL operations
  - https://issues.apache.org/jira/browse/SPARK-4062

SQL Processing on Spark: Hive on Spark

Hive on Spark

- Spark as a new execution engine for Hive
- Combine the strength of Hive and Spark
  - Support full Hive feature set
  - Utilize Spark as the powerful execution engine, greatly boost SQL performance
- Smooth migration for existing Hive users



Community efforts

- https://issues.apache.org/jira/browse/HIVE-7292
- Collaborations among Cloudera, Intel, MapR, Databricks, IBM, etc.
- Current plan
  - Merge back to Hive trunk in February 2015
  - Community Beta release in 1H’2015

SQL Processing on Spark: Spark SQL

Spark SQL

- Structured data analysis using SQL queries on Spark
  - Hive tables, Parquet files, etc.
- Integration with analytics pipelines

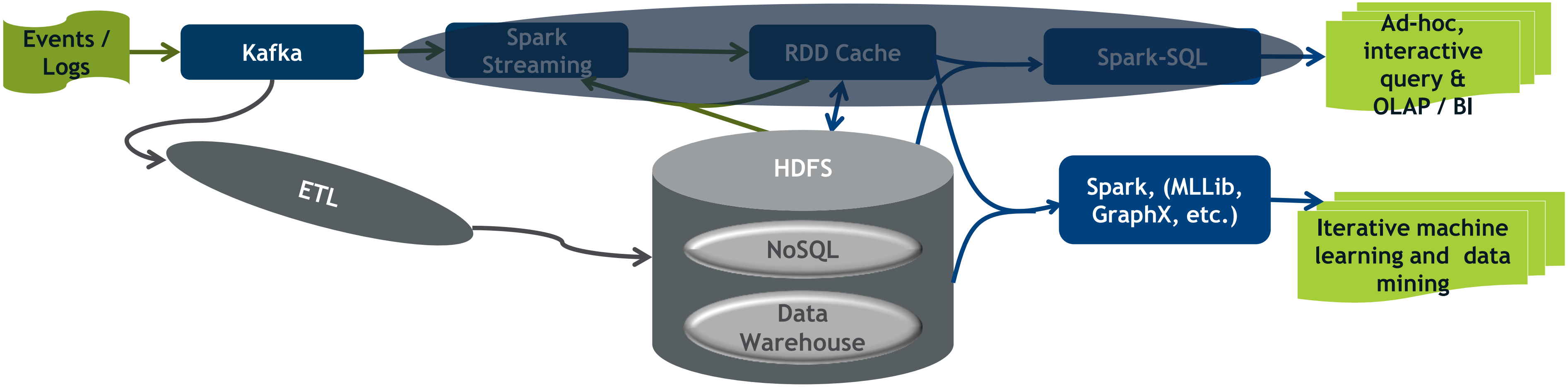
Intel’s efforts

- Functionalities (e.g., HiveQL compatibility)
  - Data types (Timestamp, Date, Union, etc.)
  - GroupingSet/ROLLUP/CUBE (https://issues.apache.org/jira/browse/SPARK-2663)
  - …
- Performance (esp. complex queries)
  - Join performance (https://issues.apache.org/jira/browse/SPARK-2211)
  - Aggregation performance (https://issues.apache.org/jira/browse/SPARK-4366)
  - Hive UDF performance (https://issues.apache.org/jira/browse/SPARK-4093)
  - …

Spark Stream-SQL: SQL + Streaming

Spark Stream-SQL

- Processing and analyzing input data stream (potentially combined with history/reference data) using SQL queries
- Built on top of Spark Streaming and Spark SQL frameworks



Events / Logs
Kafka
Spark Streaming
RDD Cache
Spark-SQL
Ad-hoc, interactive query & OLAP / BI
ETL
HDFS
NoSQL
Data Warehouse
Spark, (MLLib, GraphX, etc.)
Iterative machine learning and data mining

```
CREATE STREAM IF NOT EXISTS people_stream1 (name STRING, age INT)
STORED AS LOCATION 'kafka://…';

CREATE STREAM IF NOT EXISTS people_stream2 (name STRING, zipcode INT)
STORED AS LOCATION 'kafka://…';

SELECT zipcode, AVG(age)
FROM people_stream1 JOIN people_stream2 ON people_stream1.name = people_stream2.name
GROUPBY zipcode;
```

Spark Stream-SQL: SQL + Streaming

Open sourced under Apache 2.0 license

- https://github.com/intel-spark/stream-sql
- Developer preview (based on Spark 0.7) available

Currently under active development

- An update based on latest Spark version will be available soon
- Many more features & optimizations are being added
- Plan to contribute back to the main Spark project

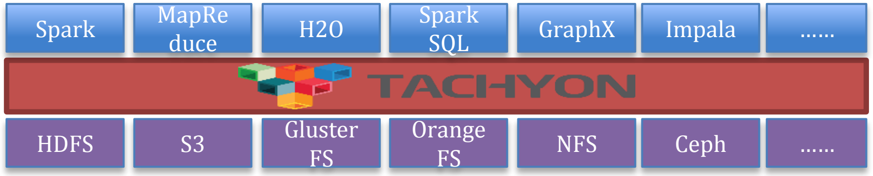
Welcome Collaboration!

Tachyon Hierarchical Storage

Tachyon

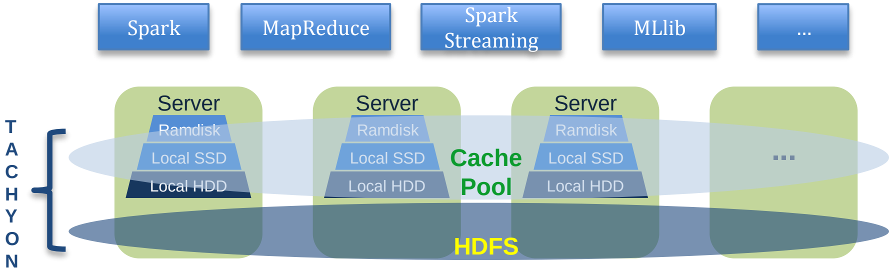
- Distributed, reliable in-memory file system unifying different underlying storage systems
  - Memory across different servers organized as a cache pool for memory-speed data sharing

(Source: http://www.slideshare.net/haoyuanli/tachyon20141121ampcamp5-41881671)



Tachyon hierarchical storage

- The cache pool manages multiple storage tiers for memory-speed data sharing
  - Efficient support for flash/SSD, cloud, and/or HPC environment
- Currently under active development
  - https://tachyon.atlassian.net/browse/TACHYON-33
- Will be available soon in Tachyon 0.6 release
  - https://github.com/amplab/tachyon



Analytics & SparkR

SparkR

- Distributed R frontend for analytics on Spark
  - RDD → Distributed list in R
- Alpha developer release from UC Berkeley
  - https://github.com/amplab-extras/SparkR-pkg

Our efforts/plans

- Complete SparkR RDD APIs
- Analytics performance improvements
- Better integrations of SparkR and Spark analytics frameworks
  - MLlib, GraphX, etc.
- Distributed DataFrame support
  - Leveraging Spark SQL (SchmeRDD, optimizer, etc.)

Summary

Driving next generations of Big Data Analytics with Spark Stack

- *Analysis on streaming data*: “real-time” decisions
- *Interactive analysis*: faster decisions
- *Advanced analytics*: “better” decisions

Partnering with open source community

- BDAS (Berkeley Data Analytics Stack)
- Apache Spark project
- Tachyon project
- SparkR project
- …

Welcome Collaboration!

Notices and Disclaimers

-
- Intel, the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.
  See Trademarks on intel.com for full list of Intel trademarks.
- All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.
- Performance tests are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.
- For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.
- Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.
- Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.
- Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.
- No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.
- You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- The products described may contain design defects or errors known as errata which may cause the product to deviate from publish.