Igniting Analytics

Apache Spark's Promise and Potential Perils

Philip Russom

Research Director for Data Management, TDWI

September 24, 2015

Sponsor

Speakers

Philip Russom
TDWI Research Director,
Data Management

Brent Wenerstrom
Data Engineer,
Think Big Analytics

Agenda

- What is Spark and why care?
  - *High performance engine*
  - *Analytic uses of Spark's libraries*
- Desirable characteristics of Spark
  - *High performance*
  - *Broad compatibility*
  - *Flexible deployment*
  - *One console for multiple functions*
  - *Native ANSI SQL*
- Real-world use cases for Spark
  - *Iterative queries for data exploration, etc.*
  - *SQL-based analytics*
  - *Multiple, diverse forms of analytics in one app*
- Conclusions

PLEASE TWEET --
@pRussom, @Teradata,
#TDWI, #Hadoop,
#Analytics, #BigData

DEFINITION

Apache Spark™

- Apache Spark is a parallel processing engine for big data that achieves high speed and low latency by leveraging in-memory computing and cyclic data flows.
- Spark today includes four libraries of functionality.
  All have direct applications in BI, DW, DI, & analytics.
  - *SQL, streaming data, machine learning, graph analytics*
- Spark's high performance and analytic functionality are distinct advantages over the current state of Hadoop MapReduce.

The Apache Spark Libraries

- Spark SQL
  - *ANSI standard; ODBC/JDBC; APIs for Python, Scala, Java*
  - *Hive's GUI & metastore can be a frontend for SQL queries*
- Spark Streaming
  - *Scalable fault-tolerant streaming apps; in Python, Scala, Java*
  - *Reuse batch jobs; join stream & historic data; query a stream state*
- MLib (machine learning)
  - *Contains many high-quality algorithms that leverage iteration*
- GraphX (API & parallel computation engine for graph analytics)
  - *Work with graphs and collections simultaneously*

Desirable Characteristics of Spark

- High performance
- Broad compatibility
- Flexible deployment
- One console for multiple functions
- Native support for standard SQL

- I'll dive into more detail for each of these.

High Performance

- Benchmarks show Spark to be up to one hundred times faster than Hadoop MapReduce with in-memory operations.
- Spark is ten times faster than MapReduce with disk-bound operations.
- Spark has the low latency required of new practices, like data exploration, discovery, and SQL-based analytics.

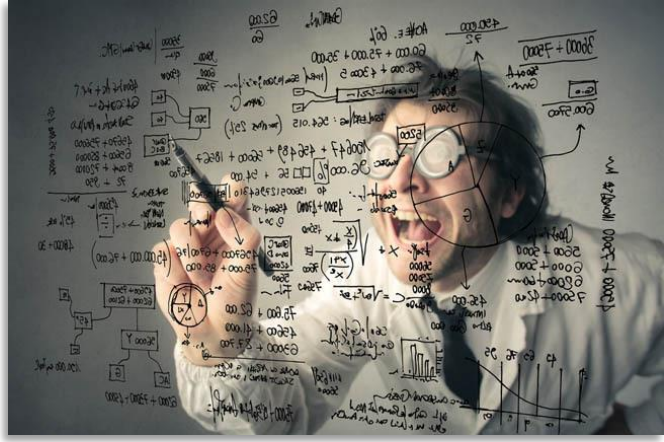
Broad Compatibility

- Spark SQL reuses the Hive front-end and metastore, to provide compatibility with existing Hive data, queries, UDFs.
- Spark SQL's server mode extends interoperability via industry-standard ODBC/JDBC.
- Spark can process data in S3, HDFS, HBase, Hive, Cassandra, and any Hadoop InputFormat.

Flexible Deployment

- Spark requires some kind of shared file system (NFS compliant), so its deployment options are diverse.
- Spark runs on its standalone cluster, Hadoop YARN, Apache Mesos, and Amazon EC2; on premises or cloud.
- Works with systems from Databricks, Paxata, IBM, etc…
- A single job, query, or stream processing can be deployed in either batch or interactive mode via Scala, Python, and R shells.

One console for multiple functions



- Apache Spark includes libraries for four high-level applications:
  - *SQL, streaming data, machine learning, and graph analytics*
- These are integrated tightly, so users can create applications that mix SQL queries and stream processing, alongside complex analytic algorithms.
- Hence, Spark fosters seamless development with diverse analytic functionality.

Native support for standard SQL

- A modern enterprise wants to leverage pre-existing SQL skills and SQL-based tools that comply with ANSI & ISO.
- Furthermore, users want fast queries on Hadoop, to enable data exploration, analytics, and other interactive, data-driven practices.
- Spark and its SQL support promise to enable these, which in turn will spark big data analytics for end users.

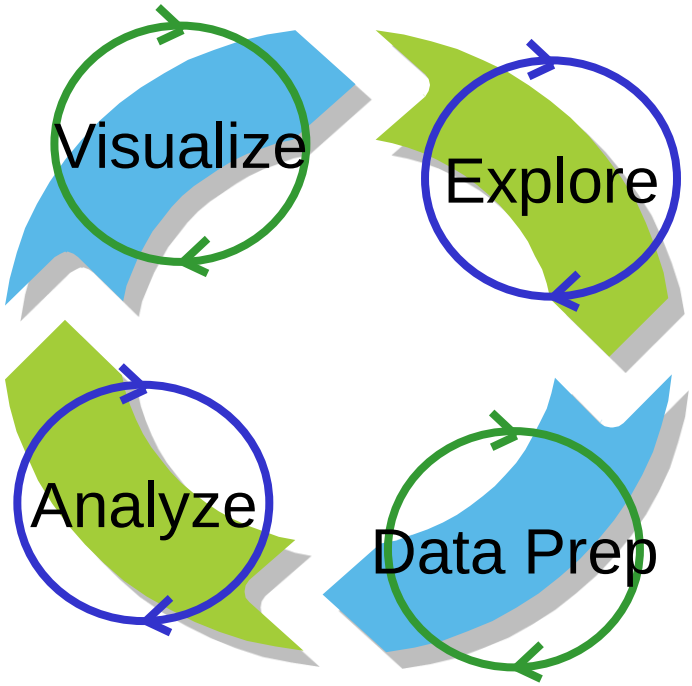
Real-World Use Cases for Spark

- Data exploration
  - *with fast iterative SQL queries*
- SQL-based analytics
  - *with more complex SQL than ever before*
- Miscellaneous analytic applications
  - *with mixtures of SQL, streams, graph, etc.*

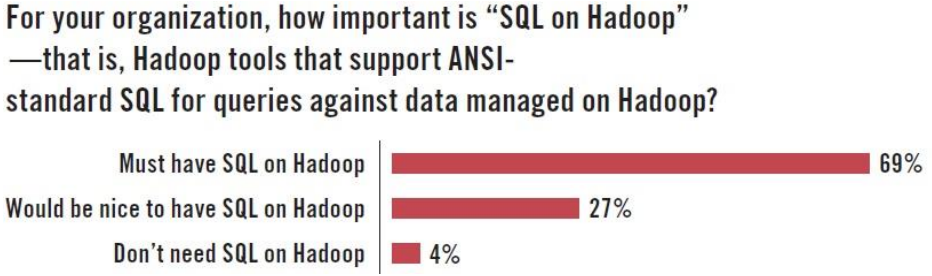
ITERATIVE, FOUR-STEP PROCESS FOR

Exploratory Analytics with Large and Diverse Data Sets

- Spark's libraries can all potentially contribute to capturing, exploring, preparing, and analyzing a wide range of data, both old and new.
- Spark promises to do that with speed, scale, and standard SQL.



SQL is More Important than Ever



- For example, consider the "SQL on Hadoop" versus "SQL off Hadoop" argument
  - *In recent TDWI survey, SQL on Hadoop is a "must have" (69%)*
    - Only 4% don't need SQL on Hadoop
    - (Source: TDWI survey run in late 2014. Based 99 respondents.)
  - *Users interviewed by TDWI want BOTH !*
    - In fact, the consensus is that BI/DW pros need solid SQL on all platforms.
    - Spark promises to help with that.

SQL-Based Analytics

- Most common form of analytics after OLAP
- Data Exploration = Ad-hoc queries on steroids
  - *Size, scope, complexity of query grows with each iteration*
- Complex SQL expresses many things
  - *Data access via many interfaces, near real time*
  - *Data models, even dimensional ones*
    - Data Prep = usually SQL-based
  - *Multi-way joins, but also complex transformations*
- KLOCs = Thousands of Lines of [SQL] Code
  - *Whether tool-generated, hand-written, or both*
- Growing number & diversity of users who need SQL
  - *Data analysts, data scientists, BI/DW pros, biz analysts*
- All the above demand a hefty tool environment
  - *SQL on Hadoop is key; Spark promises to help*

Miscellaneous Uses for Apache Spark

- SQL analytics & related set-based applications
  - *Customer-base segmentation, financial analyses, dimensional modeling and analysis, reporting*
- Stream capture and analysis
  - *Monitoring facilities (utilities, factories), tracking social sentiment, predictive machine maintenance, reroute traffic, manage mobile assets, any time-sensitive process*
- Graph analytics
  - *Anomaly detection for fraud or risk, behavioral analysis, entity clustering, patient outcome optimization*
- Mixtures of the above
  - *Trend: mix multiple analyses; each reveals different insight*
  - *Spark promises unified environ for such mixtures*

Conclusions

- Keep an eye on Apache Spark™
  - *It's too early to tell, but Spark has the potential to ignite analytics in multiple ways*
- Spark promises a lot for data management and analytics professionals
  - *Corrects a number of Hadoop weaknesses around SQL, speed, concurrency, analytic diversity, streams and real time*
  - *But Spark is not just for Hadoop*
  - *ANSI/ISO standard SQL with speed and scale*
  - *High performance for SQL and analytics*
  - *Analytic apps that mix multiple approaches*
- Spark has many real-world uses, including
  - *Low latency queries for data exploration*
  - *Analytics based on combinations of SQL, streams, machine learning, and graph*

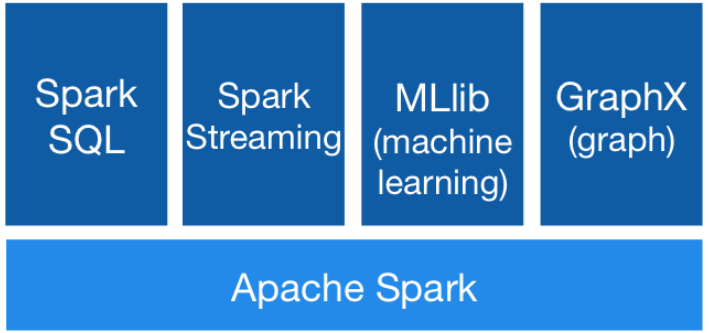
Spark: Insights From Field Engagements

Brent Wenerstrom, Sr. Data Engineer

Sept. 24, 2015

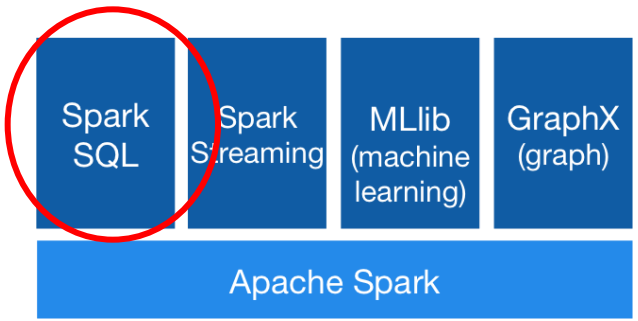
Where Does Spark Core Fit?

- Framework for parallel processing
- Hadoop competitor
  - MapReduce killer
  - Often built on the back of Hadoop
    - Cluster Resource Manager (Yarn)
    - Distributed File System (HDFS)
- Take input from some system, transform/analyze, write
  - Storage needed such as HBase, S3, Hive, database, etc.

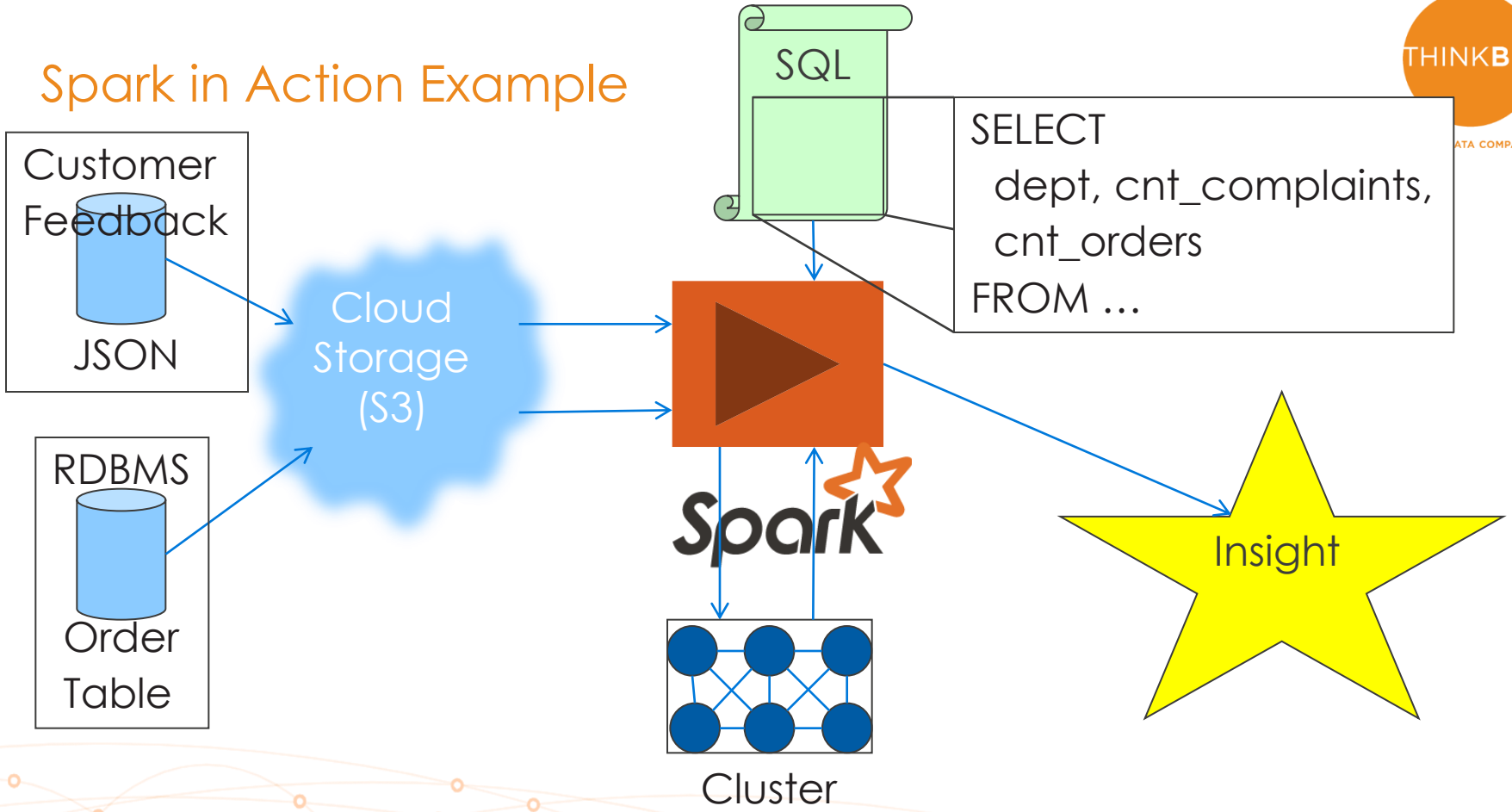


Spark SQL: What is it?

- One of a variety of languages to create parallel jobs
  - In place of Scala, Python, Java and now R
  - Replacement for Hive, Pig
- Similar to Hive
  - In fact uses the Hive library
  - Can read from and write to Hive, processing done in Spark engine
  - Hive scripts with minor modifications could in many cases be run through Spark
- Limited documentation from apache.spark.org
  - Details of supported SQL missing



Spark in Action Example

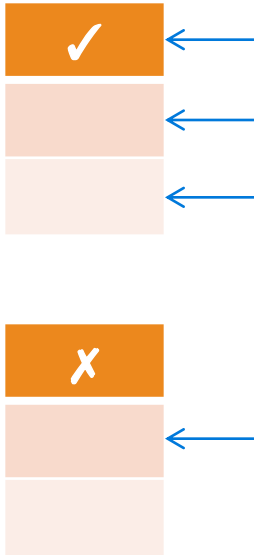


Spark Streaming

- Near real-time
  - Keep up to minute counts
  - React to data events
- Versus Storm
  - Storm meant for real-time
  - Spark is higher level
  - Spark great for windowed aggregation (sum last 10 minutes)
  - Spark can use SQL

Spark Realities

- Where Spark shines
  - Transform whole data set at once
  - Ad-hoc analysis
  - One-off find needle in haystack
- Where Spark may not be first choice
  - Data set fits in memory on single machine
    - Unnecessary overhead
  - Frequent single data point access or update
    - Each update creates new RDD
  - Storage engine
    - Not a database replacement

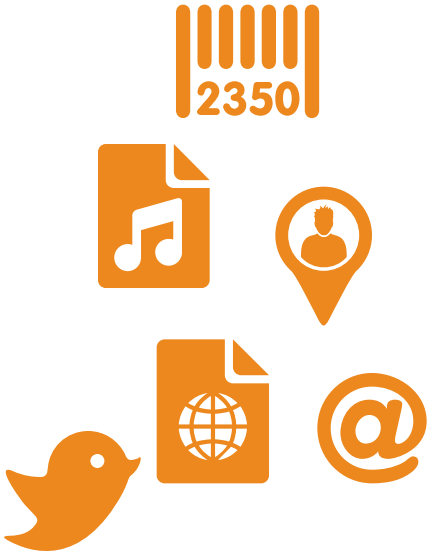


Ad Hoc Spark

- Original need for Hive and Pig
  - Hadoop code large to accomplish simple aggregations
  - Not interactive
- Spark Core
  - Many scripting options just as simple as Pig and Hive
  - Much larger support out of the box
    - Deploy any Python library to cluster
    - Parse Excel documents with Java library
  - Directly perform statistics and machine learning on the cluster

Data Sets and Spark

- Size:
  - Big enough to necessitate cluster (at least a few gigabytes)
  - Small enough to fit in memory of affordable cluster (smaller than a terabyte)
- Type:
  - Tables may or may not gain performance boost
  - Any text format
  - Custom processing with standard Java/Scala/Python library
    - JSON
    - Images
    - Graph (using GraphX module)



Production Pitfall: Spark Memory

- Estimate memory requirements
  - Need at least double main memory in cluster
- Usually first challenge of production code
  - Forget to remove previous states from memory
  - "Hydrated" version of data bigger than original bytes
  - Map and reduce operations require some open memory to work with
  - Track memory usage with GUI

Production Pitfall: Data Assumptions

- Data is too large to view
  - Impossible to review every possible scenario
- Make bad assumptions
  - Always have a create date
  - Names can be used to join tables
    - First name part of last name, etc.
  - Never have cycles in graph (loops)

Photo credit: John Drake. Courtesy of Flickr. Creative Commons.

Future of Spark

- Recent releases have emphasized Data Science
  - Integrated with R
  - Expansion of machine learning algorithms in MLLib
  - Data Frames similar to R and Pandas (commonly used in Data Science in Python)
- Spark is gaining momentum and critical mass
  - With YARN or standalone cluster
  - Cloud deployments gaining traction

Who is Think Big?

- Founded in 2010 with over 100 engagements across 70 clients
- Specialize in creating business value from big data
- Proven vendor-neutral open source expertise
- Dedicated Think Big Academy for change management and organizational development
- On-shore Solution Center and off-shore delivery model for cost effective big data deployments
- Data engineering solutions for data lake, analytics, data science

Summary

- Parallel processing framework
  - Great for combining and analyzing data
- SQL is a job creation language
- Streaming when timing is everything
- Pitfalls of production: memory and data assumptions
- If Spark is a good fit, embrace the speed of in cluster memory analytics!

Questions?

Contact Information

If you have further questions or comments:

Philip Russom, TDWI
prussom@tdwi.org

Brent Wenerstrom, Think Big
brent.wenerstrom@thinkbiganalytics.com