IBM z Systems
Introduction to Spark

Joe Bostian
jbostian@us.ibm.com

April, 2016

Acknowledgements

- Apache Spark, Spark, Apache, and the Spark logo are trademarks of The Apache Software Foundation.

Topics

- What Spark is (and is not)
- The Spark community and IBM's commitment
- Spark details
- Why use Spark on z/OS?
- The ecosystem for Spark on z/OS
- Demo video
- Discussion

What Spark is (and is not)

What Spark Is, What it Is Not

- An Apache Foundation open source project
  - *Not a product*
- An in-memory compute engine that works with data
  - *Not a data store*
- Enables highly iterative analysis on *huge* volumes of data at scale
- Unified environment for data scientists, developers and data engineers
- Radically simplifies the process of developing intelligent apps fueled by data

What Spark on z/OS is Not

What it is not

- A data cache for all data in DB2, IMS, IDAA, VSAM …
- Just a different SQL engine or query optimizer
- An effective mechanism to access a ***single*** data source for analytics

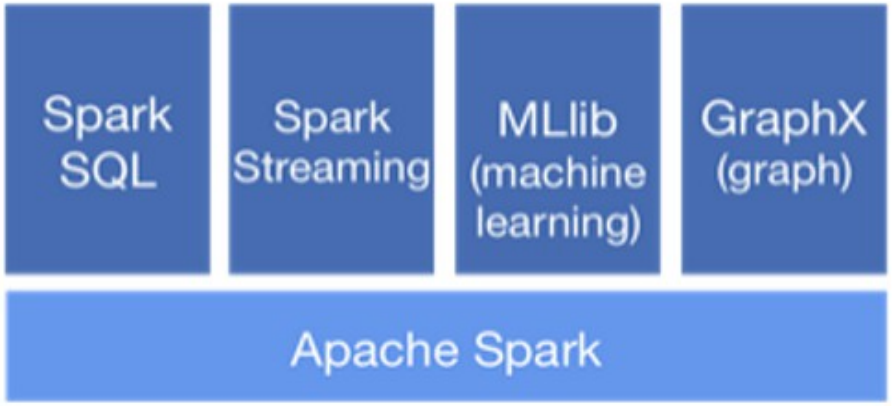
Why isn't it the same as a query acceleration / IDAA?

- Spark does not optimize SQL queries
- Spark is not a mechanism to store data, but rather provides interfaces to access portions of required data & most importantly to apply analytics using a unified interface
- IDAA interaction with applications is via the DB2 z/OS paradigm; Spark interaction with applications is via Spark interfaces (Stream, MLlib, Graphx, SQL), driven through REST or java
- Spark analytics can access data in DB2, IDAA, VSAM, IMS, off platform, etc.

Apache Spark - a Compute Engine

General Purpose

- Covers a wide range of workloads
- Provides SQL, streaming and complex analytics



From http://spark.apache.org/

Fast

- Aggressively caches data
- Distributes computing
- Uses JVM threads
- Faster than MapReduce for some workloads

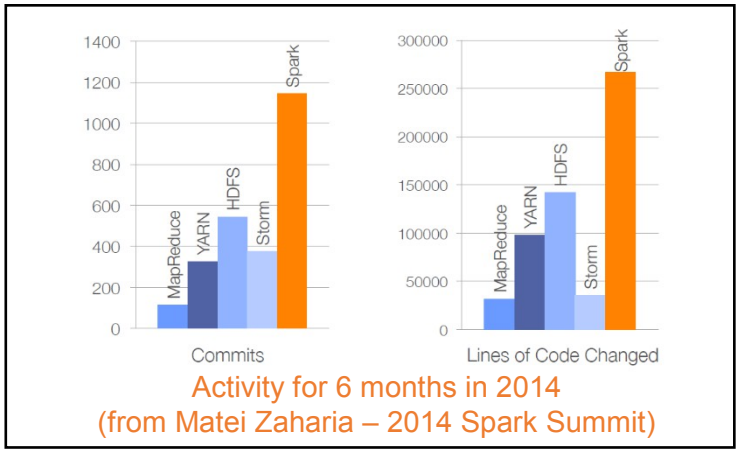
Easy to Use

- Written in Java
- Scala, Python and Java APIs
- Runs on Hadoop with Mesos, standalone, or cloud
- Scala and Python interactive shells

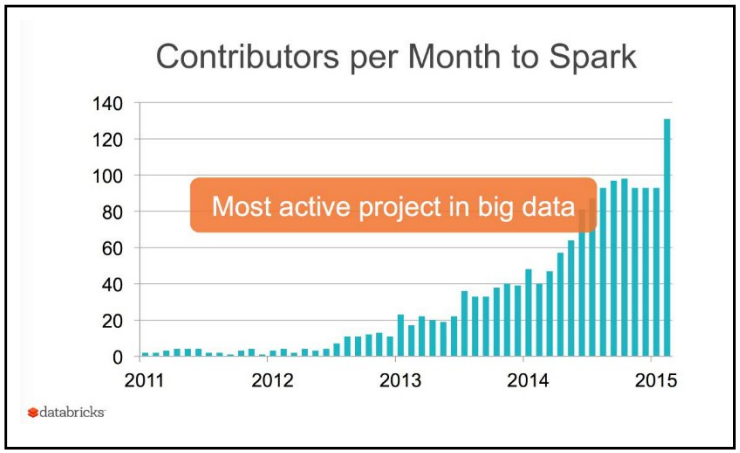
The Spark community and IBM's commitment

Brief History of Spark

2014 – 1.0.0 release in May
2014 – 1.1.0 release in September
2014 – 1.2.0 release in December
2015 – 1.3.0 release in March
2015 – 1.4.0 release in June
2015 – 1.5.0 release in September
2016 – 1.6.0 release in January
2016 – 2.0.0 release in April/May



Activity for 6 months in 2014
(from Matei Zaharia – 2014 Spark Summit)



- Most active project in Apache Software Foundation
- Databricks founded by the creators of Spark from UC Berkeley's AMPLab

IBM's Involvement and Commitment

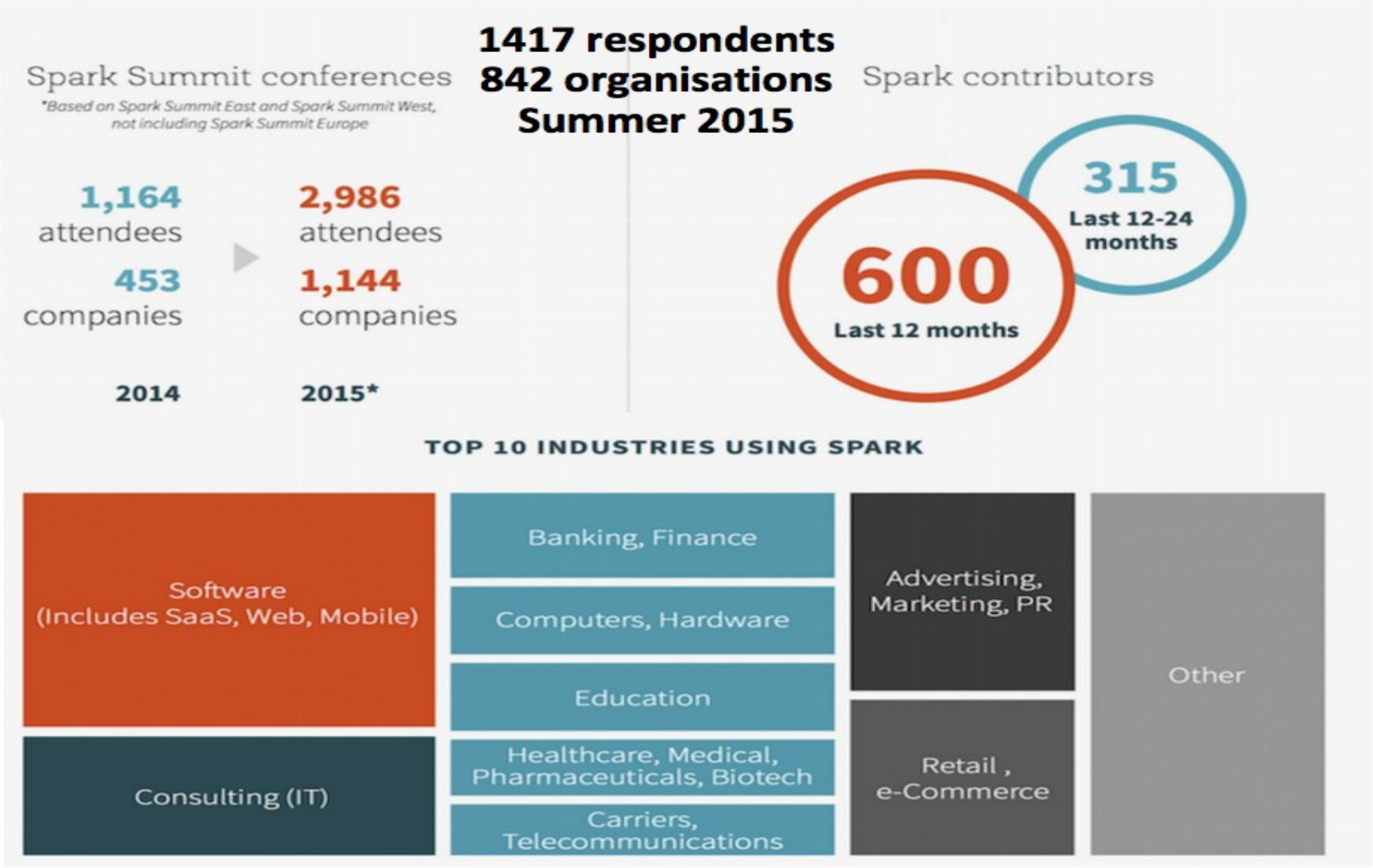
- IBM is one of the four founding members of the UC Berkeley AMPLab
  - Work closely with AMPLab research on projects of mutual interest

- June 2015, IBM announced:
  - 3,500 researchers and developers to work on Spark-related projects at IBM labs worldwide
  - IBM donated SystemML machine learning technology to the Spark open source ecosystem
  - Spark Technology Center established in San Francisco for the data science and developer community

- IBM supports the Spark community
  - Code contributions
  - Partnerships with AMPLab Galvanize and Big Data University (MOOC)
    - Education for data scientists and developers

- Visit the IBM Spark Technology Center at http://www.spark.tc

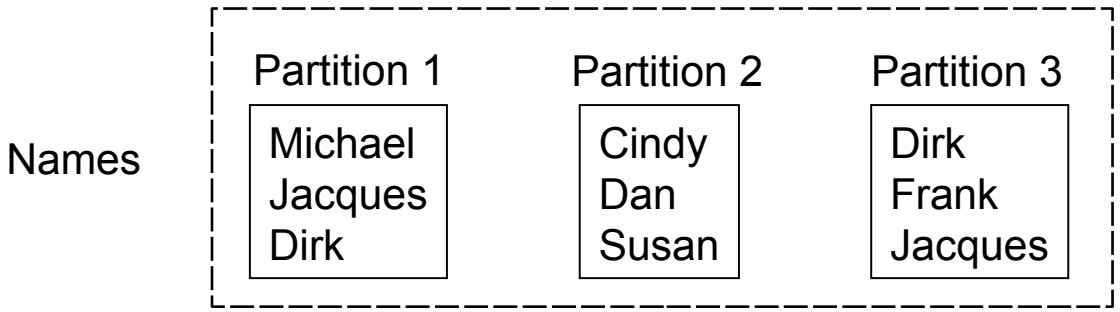
Spark Activity and Users by Industry



Spark Details

Resilient Distributed Datasets (RDDs)

- Spark's basic unit of data
- Immutable, fault tolerant collection of elements that can be operated on in parallel across a cluster
- Fault tolerance
  - If data in memory is lost it will be re-created from lineage
- Caching, persistence (memory, spilling, disk) and check-pointing
- Many database or file types can be supported
- An RDD is physically distributed across the cluster, but manipulated as one logical entity:
  - Spark will "distribute" any required processing to all partitions where the RDD exists and perform necessary redistribution and aggregations as well.
  - Example: Consider a distributed RDD "Names" made of names



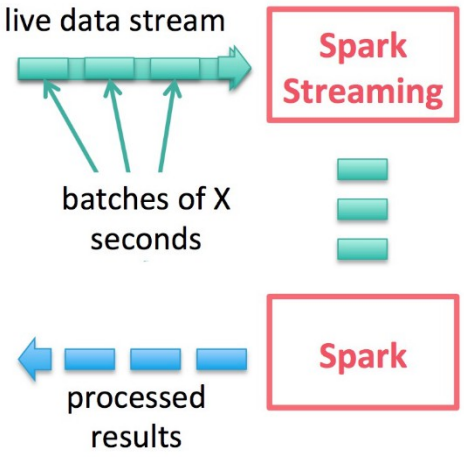
Common, Popular Methods to Access Data

- Spark SQL
  - Provide for relational queries expressed in SQL, HiveQL and Scala
  - Seamlessly mix SQL queries with Spark programs
  - Provide a single interface for efficiently working with structured data including Apache Hive, Parquet and JSON files
  - Standard connectivity through JDBC/ODBC
  - Integration of Spark z/OS with Rocket Software provides unique functionality to access data across a wide variety of environments with very high performance and flexibility

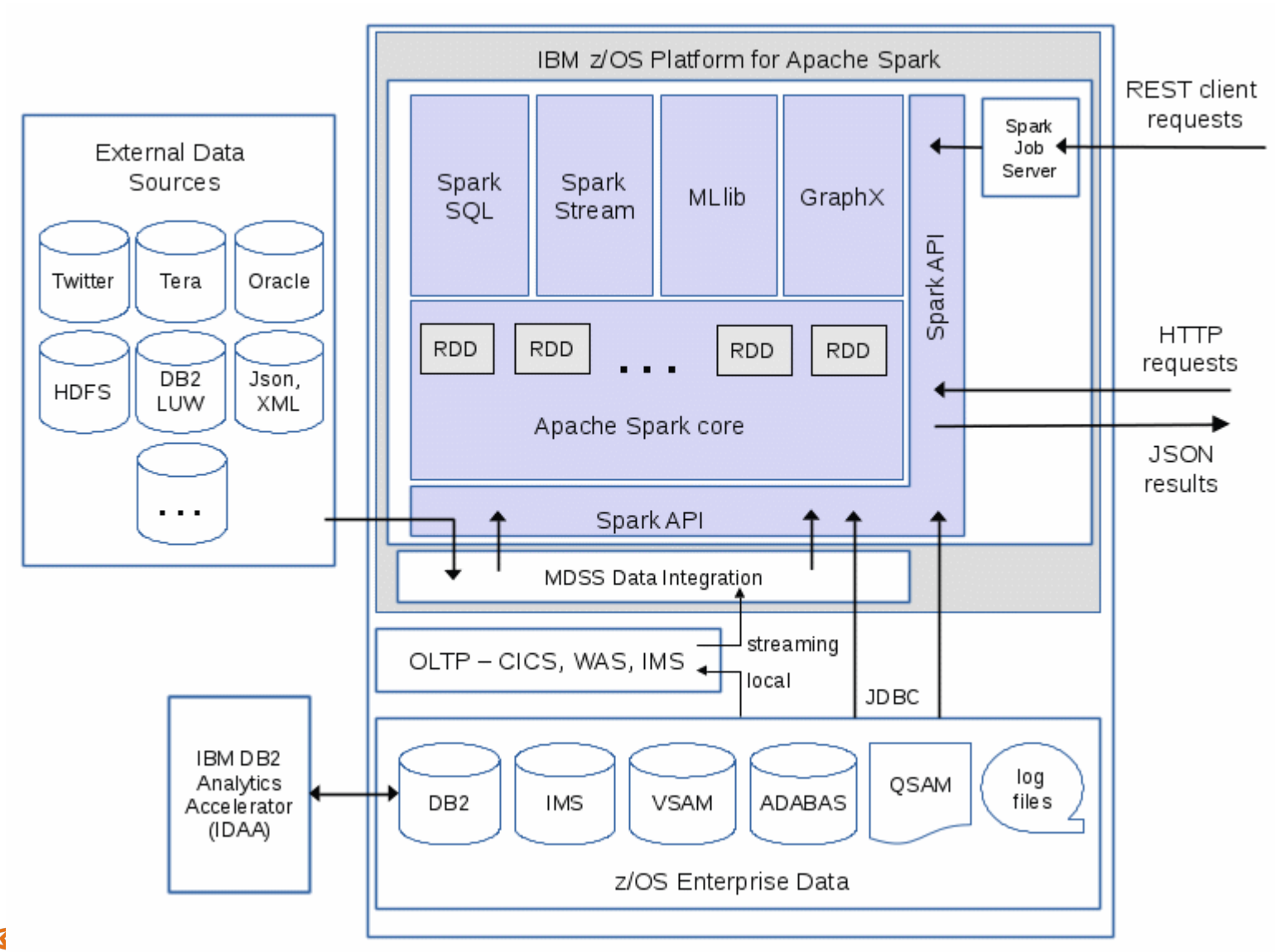
- Spark R
  - Spark R is an R package that provides a light-weight front-end to use Apache Spark from R
  - Spark R exposes the Spark API through the RDD class and allows users to interactively run jobs from the R shell on a cluster.
  - Goal is to make Spark R production ready
  - Rocket Software has announced intent to support R on z/OS

Spark Streaming

- Run a streaming computation as a series of very small, deterministic batch jobs
  - Chop up live stream into batches of X seconds
  - Spark treats each batch of data as RDDs and processes them using RDD operations
  - The process results of the RDD operations are returned in batches
- Combine live data streams with historical data
  - Generate historical data models with Spark
  - Use data models to process live data
- Combine Streaming with MLlib algorithms
  - Offline learning, online predictions
  - Actionable information



IBM z/OS Platform for Apache Spark



IBM z/OS Platform for Apache Spark ...

- Almost any data source from any location can be processed in the Spark environment on z/OS
- Mainframe Data Service for Apache Spark (MDSS) is key to providing a single, optimized view of heterogeneous data sources
  - MDSS can integrate off-platform data sources as well
  - Large majority of cycles used by MDSS are zIIP-elegible
  - Possible to use Spark on z/OS without it, but MDSS is recommended
- OnLine Transaction Processing (OLTP) is possible, but performance may be challenging
  - Spark is the high performance solution for processing big data, but was never intended to provide analysis in real-time
  - Off-platform data sources may have latency concerns
  - Is near-real-time performance good enough for your needs?
- IDAA optimization with with z/OS DB2 can be integrated with this environment

Why use Spark on z/OS?

Why Use Spark on z/OS?

The environment where Apache Spark z/OS makes sense:

- Running real-time or batch analytics over a variety of *heterogeneous* data sources
  - Efficient real-time access to current and historical transactions
- Where a majority of data is z/OS resident
- When data contains sensitive information
  - Don't scatter across several distributed nodes to be held in memory for some unknown period of time
- When implementing common analytic interfaces that are shared with users on distributed platforms

z/OS strengths are valuable in a Spark environment:

- Intra-SQL and intra-partition parallelism for optimal data access to:
  - Nearly all z/OS data environments
  - Distributed data sources

z/OS Strengths and Spark ...

- Sysplex enabled Spark clusters for world class availability
- Best-fit analytic capability for the investments made in SMF in-memory analytics
- SMT2 for added thread performance
- SIMD enhances performance on select operations
- zIIP eligible to reduce CPU cost
- z/OS's superior memory management:
  - RDMA capabilities
  - Large page support
  - Off-heap memory
  - DRAM integration with Flash for scalable elastic memory
- zEDC compression for internal data when Spark caches and shuffles
- Network acceleration for Spark clusters through RDMA SMC-R technology

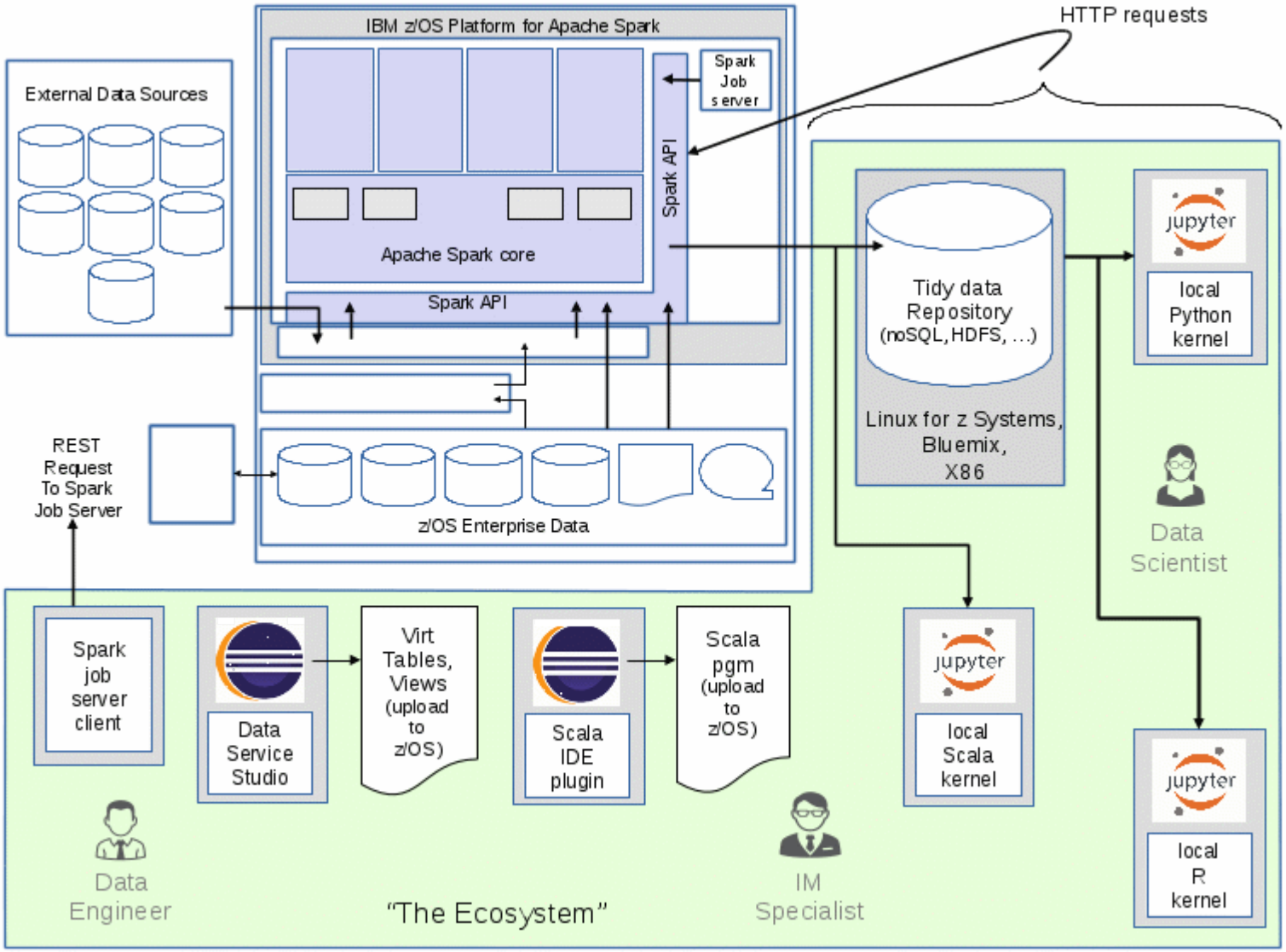
The ecosystem for Spark on z/OS

The Ecosystem for Spark on z/OS

There are several consumers / customers for the analysis performed on z/OS:

- ***The Data Scientist*** - the primary customer
  - Creates the spark application(s) that produce insights with business value
  - Probably doesn't know or care where all of the Spark resources come from
- ***The Information Management Specialist***
  - Helps the Data scientist assemble and clean the data, write applications
  - Probably better awareness about resource details, but still is primarily concerned with the problem to solve, not the platform
- ***The Data Engineer***
  - Also called the “data wrangler”
  - Close to the platform, probably a Z-based person
  - Works with the IM specialist to associate a view of the data with the actual on-platform assets

The Ecosystem



The Ecosystem for Spark on z/OS ...

- A Spark environment on z/OS can be used without an associated ecosystem, but:
  - ***The primary user (Data Scientist) lives here***
  - This is where the rich set of tools to develop Spark applications is located
- The tidy data repository catches all of the results reduced from the original data
  - Allows access to results for a large number of users without driving MIPs on the host
  - Keeps the results in a noSQL form that consumers already recognize
- Support for this ecosystem is available at our github site
  - http://zos-spark.github.io/
  - Contains information and installable code at no charge

IBM z Systems Spark Demo: A Use Case

https://youtu.be/sDmWcuO5Rk8

References

- Spark Communities
  - https://cwiki.apache.org/confluence/display/SPARK/Committers
  - https://amplab.cs.berkeley.edu/software/
- Spark SQL Programming Guide:
  - http://spark.apache.org/docs/latest/sql-programming-guide.html
- IBM SystemML
  - Open Source: June 2015, we announced to open source SystemML
    - https://developer.ibm.com/open/systemml/
  - SystemML has been accepted as an Apache Incubator project
    - http://systemml.apache.org/
    - Source code: https://github.com/apache/incubator-systemml
    - Published paper: http://researcher.watson.ibm.com/researcher/files/us-ytian/systemML.pdf
- Big Data University – Spark Fundamentals course
  - http://bigdatauniversity.com/bdu-wp/bdu-course/spark-fundamentals/
- IBM paper on Fueling Insight Economy with Apache
  - www-01.ibm.com/marketing/iwm/dre/signup?source=ibm-analytics&S_PKG=ov37158&dynform=19
- A Deeper Understanding of Spark Internals
  - https://www.youtube.com/watch?v=dmL0N3qfSc8
- IBM z/OS Platform for Apache Spark documentation
  - http://www.ibm.com/support/knowledgecenter/SSLTBW_2.2.0/com.ibm.zos.v2r2.azk/azk.htm

Backup slides

Backup

Use Cases for Apache Spark on z/OS

Analytics across OLTP & Warehouse information

- OLTP resides on z/OS
- Data warehouses on distributed platforms

- Analytics across these environments can be challenging and inconsistent

Analytics combining business-owned data and external / social data

- Clients have OLTP on z/OS
- Clients have external – public --- or social data on distributed servers
- External data delivers more value when combined with analytics from business data

Analytics of real-time transactions via streaming, combine with OLTP & social

- Combining real-time data streamlining into Spark with high performance with OLTP data and social media data --- for example, claims analytics

Custom analytics of SMF leveraging real-time as well as archived data

- SMF real-time data can add much more insight for IT across multiple systems, add in analytics over SMF data that has been archived

Use Cases for Real Time SMF Analytics

- **Detect excessive memory consumption – SMF30**
  - –Monitor high water mark for real memory usage for jobs and send alerts if usage exceeds normal consumption
- **Detect security violations in real-time – SMF 80**
  - –Monitor volume of datasets/files accessed per user within a given time period and raise alerts for above normal access rates

Discussion & Next Steps

- There are very likely ‘Spark’ projects occurring in your organizations ….they may even be using z Systems data at some level ….we’re anxious to engage to understand what these use cases are and whether there may be a fit with z/OS Spark capabilities and to help us understand requirements, integration scenarios, etc.

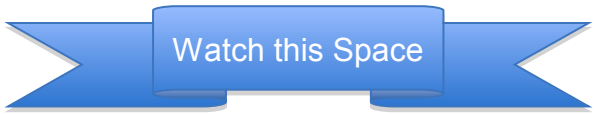
- What Operational / Integration Requirements can you envision?
  - WLM for resource management
  - TWS for Spark job scheduling
  - Application and runtime access from transactional systems
  - others?

developerWorks > Technical topics > Java technology | Big data and analytics >

IBM Packages for Apache Spark

Exploit the big data analytics capabilities of Apache Spark with this new package for IBM platforms.

- **Acquire and Install z/OS Apache Spark**
  - https://www.ibm.com/developerworks/java/jdk/spark/
  - z/OS Apache Spark is essentially a JVM that is launched and loaded with Spark class files
  - z/OS 2.1 (64bit Java 8 SDK)
- **The installation instructions for the z/OS package are available in the document**
  - Installing IBM zOS Platform for Apache Spark
- **z/OS Analytics Ecosystem Repository – Github: zos-spark (It's Coming!)**
  - Spark z/oS Scala Workbench
  - Apache Job Server
  - Python & R Workbenches
  - Sample snippets of Scala code connecting to IMS, DB2, VSAM
  - Industry specific mappings for data formats – e.g. card data



- **Technical Support**
  - Use dW Answers to ask questions and share your expertise
  - Our development team would like to hear your feedback
  - Please include the "ibmjdk" and "spark" tags to help us find your questions quickly

IBM developerWorks

Use Cases for Real Time SMF Analytics

- **Detect excessive memory consumption – SMF30**
  - –Monitor high water mark for real memory usage for jobs and send alerts if usage exceeds normal consumption
- **Detect security violations in real-time – SMF 80**
  - –Monitor volume of datasets/files accessed per user within a given time period and raise alerts for above normal access rates
- **Real time monitoring resource usage in cloud environments (CPU, Memory, Disk)**

z Systems Apache Spark

- ***Platform of choice for Apache Spark depends on use case***
  - For environments where most of the volume of data to be analyzed resides on z/OS, or where most quickly changing data resides on z/OS or most sensitive data resides on z/OS, co-locate Spark on z/OS for optimal performance, security & governance
- Spark reduces the need for clients to construct fragile, quickly out-of-date and non-agile physical "data lakes"
- Spark enables federation of analytic functions where clients can analyze data where it originates and avoid continual, costly movement
  - Available today on both z/OS and Linux on z: ***No-cost POCs available now***

Available now via developerWorks: https://www.ibm.com/developerworks/java/jdk/spark/

- ***** NEW Product – March GA with IBM support and service**
  - IBM z/OS Platform for Apache Spark
  - Optimized, native parallel access to DB2, IMS, VSAM, ADABAS, ….
  - Not limited to z/OS data – access warehouses, HDFS, etc. off platform
  - Will include special pricing for HW –zIIPs & memory
  - Analyze with Spark capabilities without spending MIPS moving data

IBM z/OS Platform for Apache Spark

ANNOUNCE DATE: Tuesday, March 22
GA DATE: Friday, March 25

LAUNCH EVENT: Strata+Hadoop World, San Jose, March 28-31

IBM z/OS Platform for Apache Spark 1.1.0

- PID: 5655-AAB
- S&S PID: 5655-AAC
- FMIDs
  - HSPK110 - Z/OS APACHE SPARK, CompID= 5655AAB01
  - HMDS110 - Z/OS MDS FOR SPARK, CompID= 5655AAB02

Spark z/OS Demo: Business Use Case

Financial Institution:
offers both retail / consumer banking as well as investment services

Business Critical Information owned by the organization:

Credit Card Info

Trade Transaction Info

Customer Info

STANDARD &POOR'S

Stock Price History – public data

Social Media Data: Twitter

GOAL

Produce right-time offers for cross sell or upsell, tailored for individual customers based on: customer profile, trade transaction history, credit card purchases and social media sentiment and information

Spark z/OS Demo: Configuration