Spark *Lightning-fast cluster computing*

Apache Spark™ is a fast and general engine for large-scale data processing.

Paul Newton (IBM)
Sr. Software Engineer
Certified Consulting IT Specialist

Can recognize a good thing

Anthony Sofia (IBM)
z System Software Design & Development
Marist College Alumini

Can explain details behind a good thing
Spark SME, Subject Matter Expert

Abstract

Spark – A Big Data Processing Framework

Spark is big and getting bigger because it runs everywhere such as z/OS and LinuxONE.

You will walk away from this presentation understanding the significance of Spark in industry and your opportunity to expose your students to this rapidly emerging Big Data Analytics tool that runs on z/OS.

Spark provides industry with an immediate opportunity to bring analytics to business critical transactions and data controlled by z/OS.

Agenda

Setting the stage

Prediction is very difficult, especially if it's about the future

Biff made presciptions with 100% accuracy, not predictions

Predictive vs Prescriptive analytic tools

Business solution architecture, the big picture

Opportunities to expose students to Apache Spark and Scala

What is Apache Spark

Setting the Stage

Big Data

Big Data Analytics

Big Data

Initially a phrase describing the exponential growth of data

- Volume massive amount
- Velocity rapid rate
- Variety structured and unstructured
- Veracity truthfulness

Rapidly changed thinking related to analytics technology and techniques

- Data Science discipline became a hot new job category
- Other new job responsibilities
  - Chief Digital Officer
  - Digital Marketing Manager

Big Data Analytics

Tools enabling competitive advantage and disruptive technologies

Many open source projects, appliances, and commercial software products

Apply analytics at time of transaction

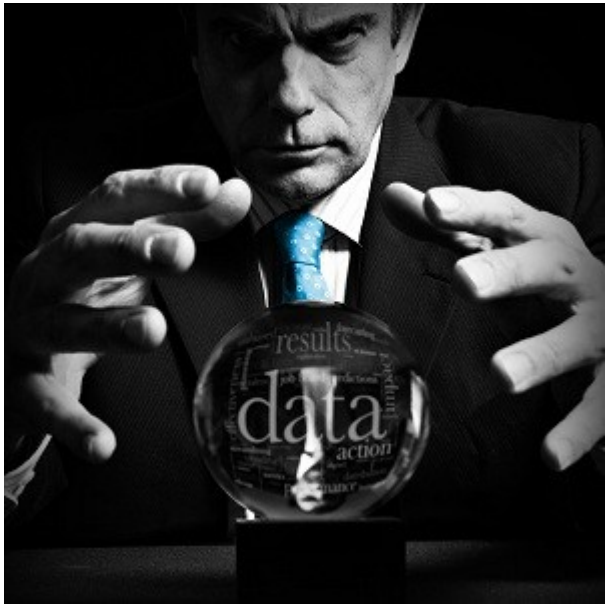
Race to get this in place and continuously innovate

Opportunity life span of unstructured (streaming) data is shrinking **

Technology is changing rapidly requiring industry to accept the need to continuously implement leapfrog strategies **

Prediction is very difficult, especially if it's about the future

Niels Bohr



An expert is a person who has made all the mistakes that can be made in a very narrow field

Niels Bohr

Opportunity life span of streaming (unstructured) data is shrinking **

Doc Brown: “Correct. They were in the time machine because Biff was in the time machine...with the sports almanac!”

Doc Brown: “Well, it's all in the past.”

Marty: “You mean the future.”

Unexpected opportunity is possible during analysis of the future

Marvin Berry: “Chuck. Chuck. It's Marvin - your cousin, Marvin BERRY.”
“You know that new sound you're looking for? Well, listen to this.”

Biff made prescriptions with 100% accuracy, not predictions

Hadoop MapReduce or Biff ?

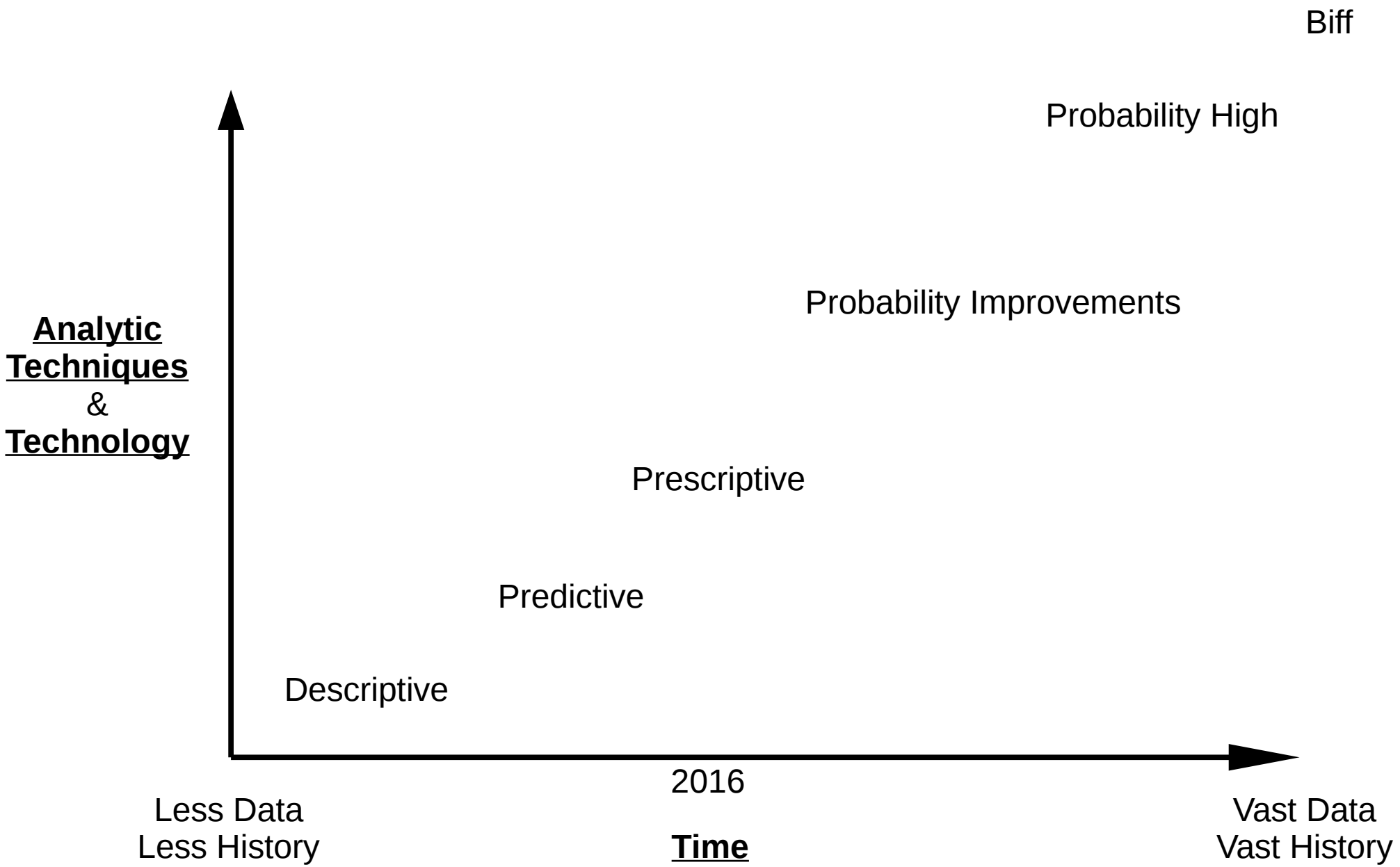
Predictive vs Prescriptive analytic tools

Hadoop MapReduce strength includes collection and analysis of a massive amount of data leading to **predictions**

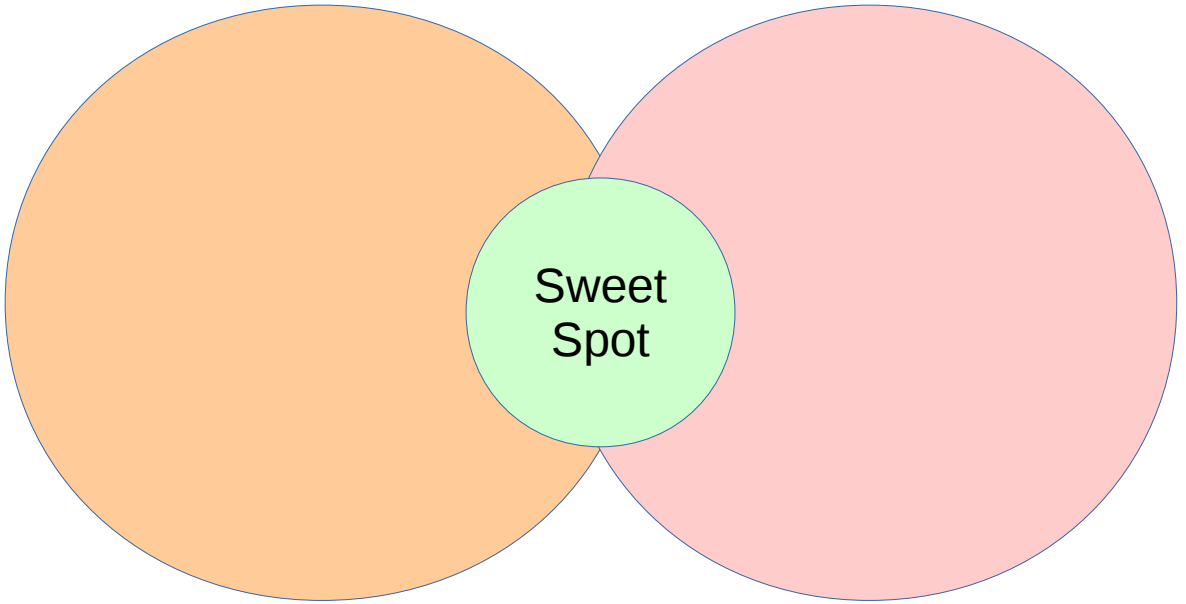
Apache Spark strength is to be more like Biff making **prescriptive** decisions

Bottom Line - Business winners will successfully apply high probability prescriptive analytics/decisions to transactions

Technology is changing rapidly requiring industry to accept the need to continuously implement leapfrog strategies **



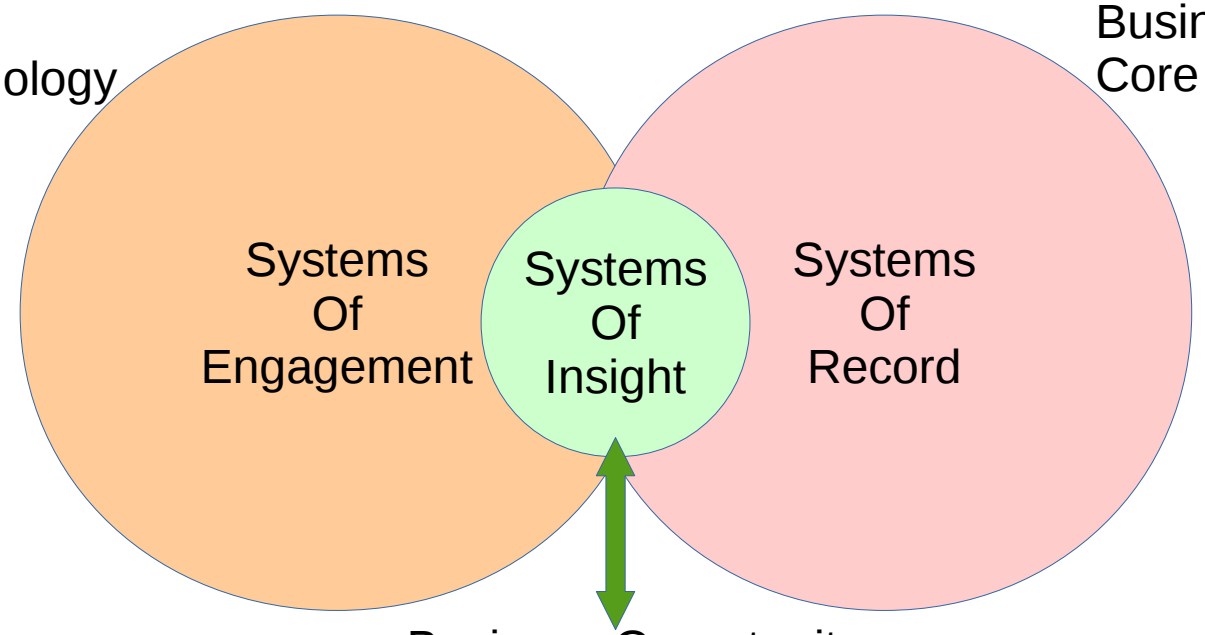
Business solution architecture, the big picture



Technology Categorization

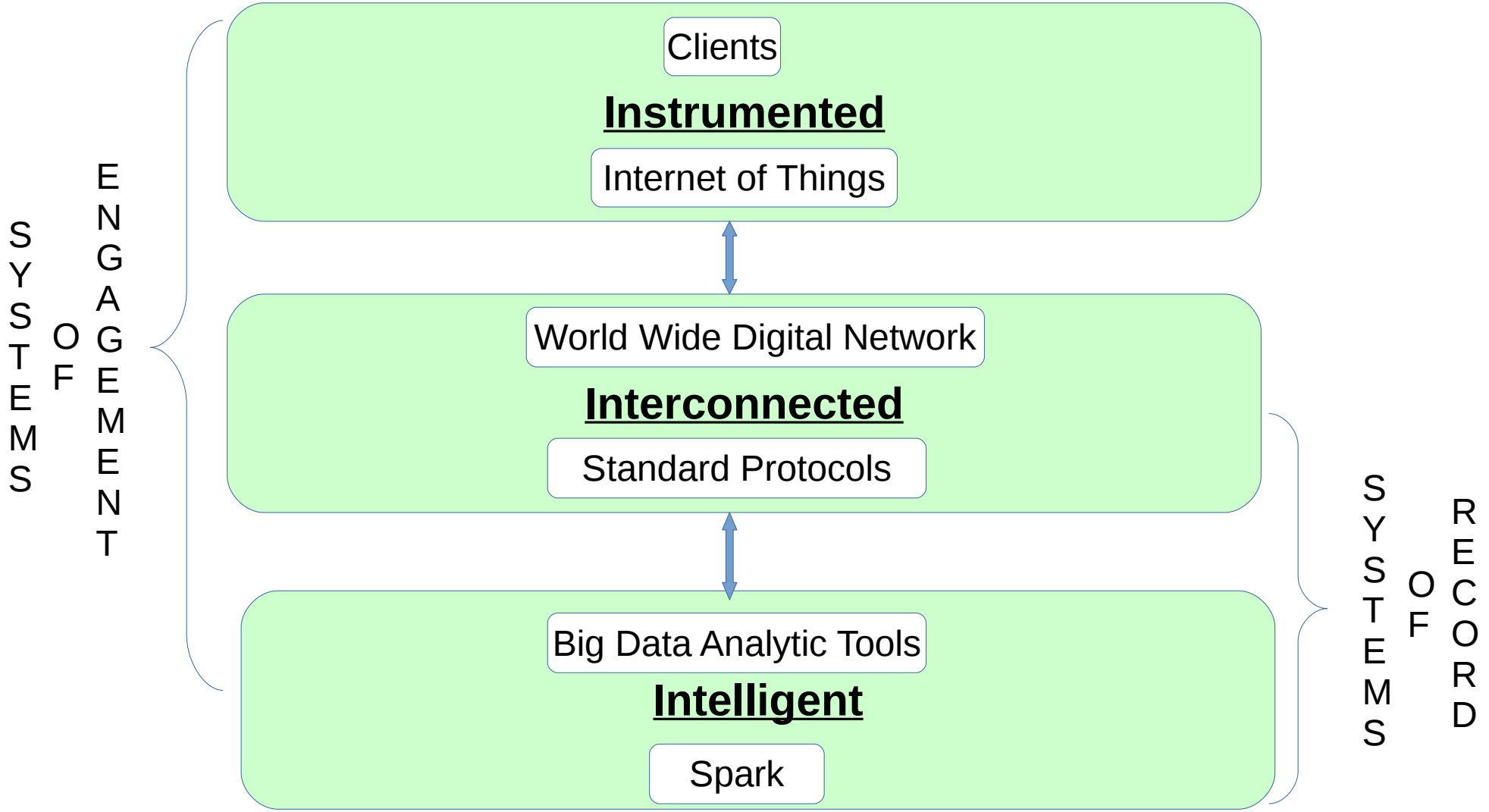
Customer Interface
Streaming Data
Disposable Data
New Frontier Technology

Business Back Office
Structured Data
Business Critical Data
Core Technology



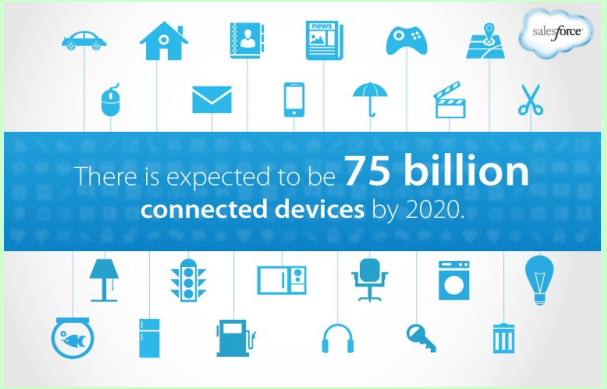
Business Opportunity
Prescriptive Analytics
Analytics at Time of Transaction

Data Gold Rush Has Begun – Think 3 I's



Think 3 I's

Instrumented



Interconnected

Message Queuing Telemetry Transport
Advanced Message Queuing Protocol
Representation State Transfer
Blockchain

Intelligent

z Systems
Mainframe

LinuxONE

Opportunities to expose students to Apache Spark and Scala

z System Academic Initiative Plans
NYIT – Pilot

A Use Case Demo

https://www.youtube.com/watch?v=yw0dQFMyxFQ

Tutorial

https://www.youtube.com/watch?v=XJyG9j1TRfw

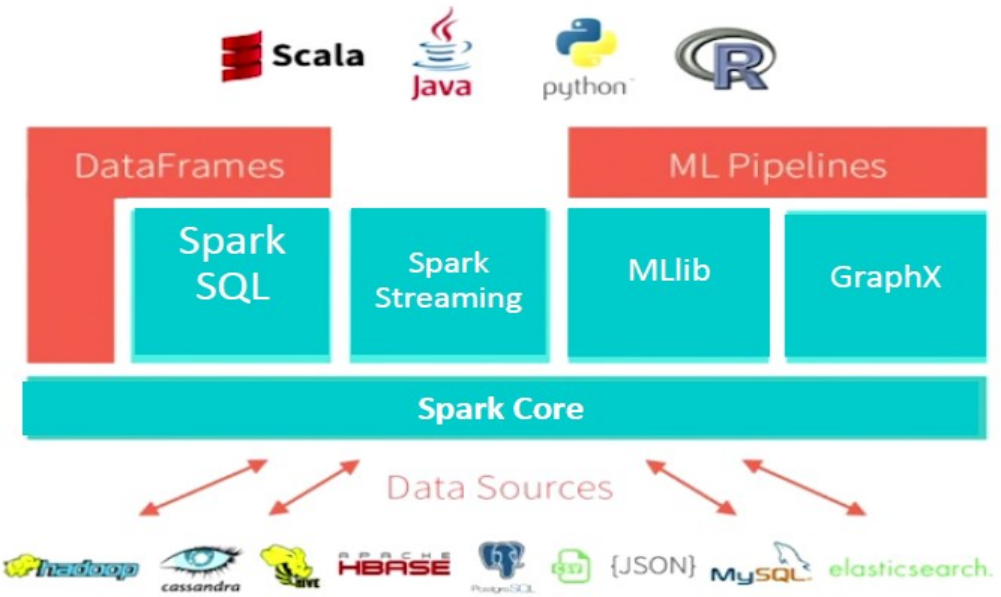
Scala

```
var jdbcDF = source match {

case "db2" => sqlContext.load("jdbc",
 Map( "url" -> "jdbc:db2://IP:PORT/DBNAME:user=USERNAME;password=PASS;",
 "dbtable" -> “DBTABLE"))


case "ims" => sqlContext.load("jdbc",
 Map( "url" -> "jdbc:ims://IP:PORT/DBNAME:user=USERNAME;password=PASS;",
  "dbtable" -> “DBTABLE"))

}
```

What is Apache Spark?

Apache Spark

Apache Spark is a fast, general purpose cluster computing platform.



- Project of the UC Berkeley AMPLab
  - Contributed to Apache in 2013
  - Now the most active project at Apache
  - 41 committers, 32 PMC members
- Databricks
  - Corporation employing principal developers
  - Forming close working relationships with a number of players
  - Offer Spark cloud service

DATABRICKS

- Spark is the natural successor to MapReduce
  - The core provides a modest set of reliable, scalable data transformation operations
  - There is a wide ecosystem built on these for SQL, Streaming, ML, and Graph processing

z/OS Platform for 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 – Will be available in March 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

Apache Spark and z/OS Data Sources

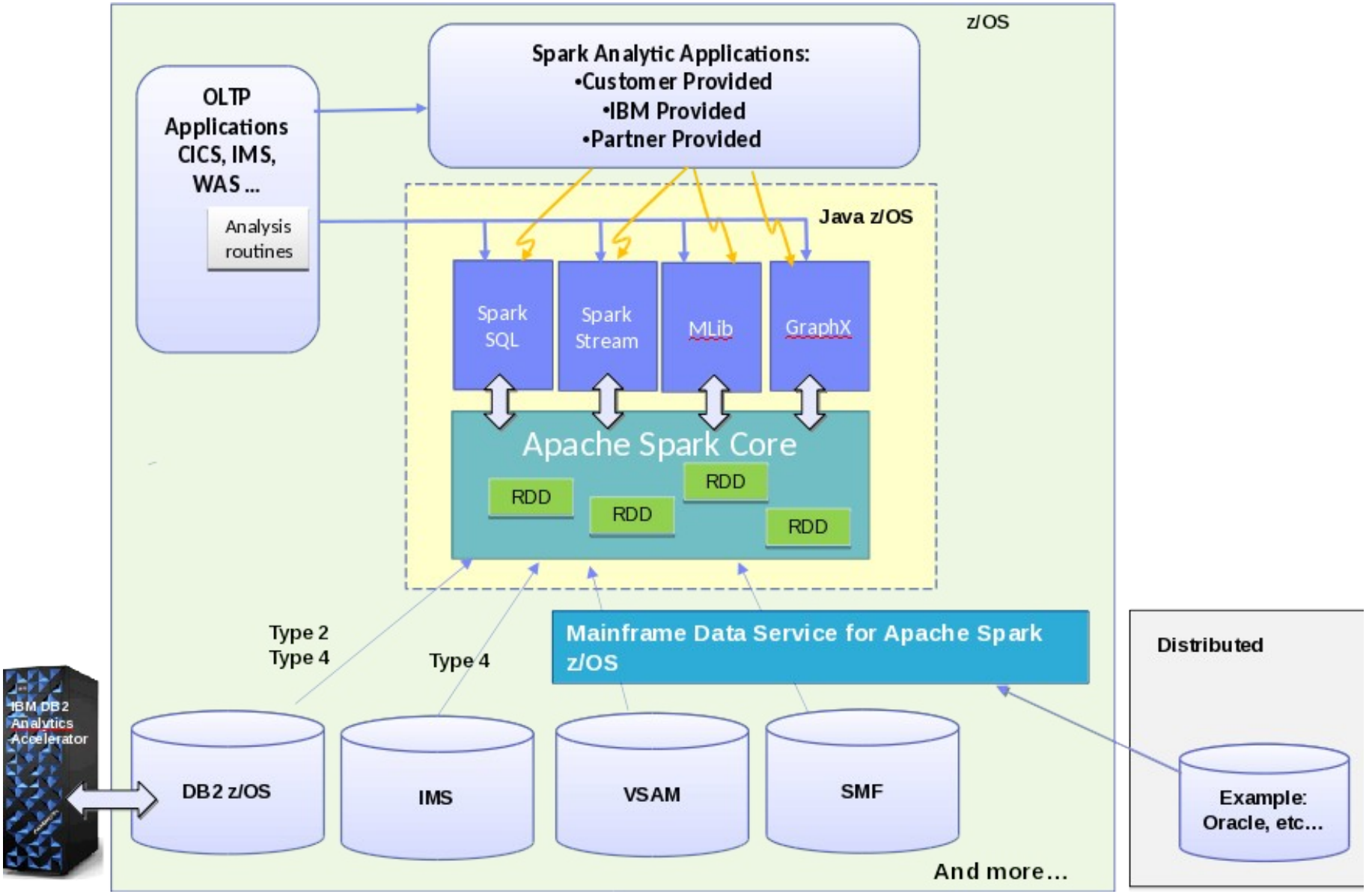
What it is not

- Not a data cache for all data in DB2, IMS, IDAA, VSAM …
- Not 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

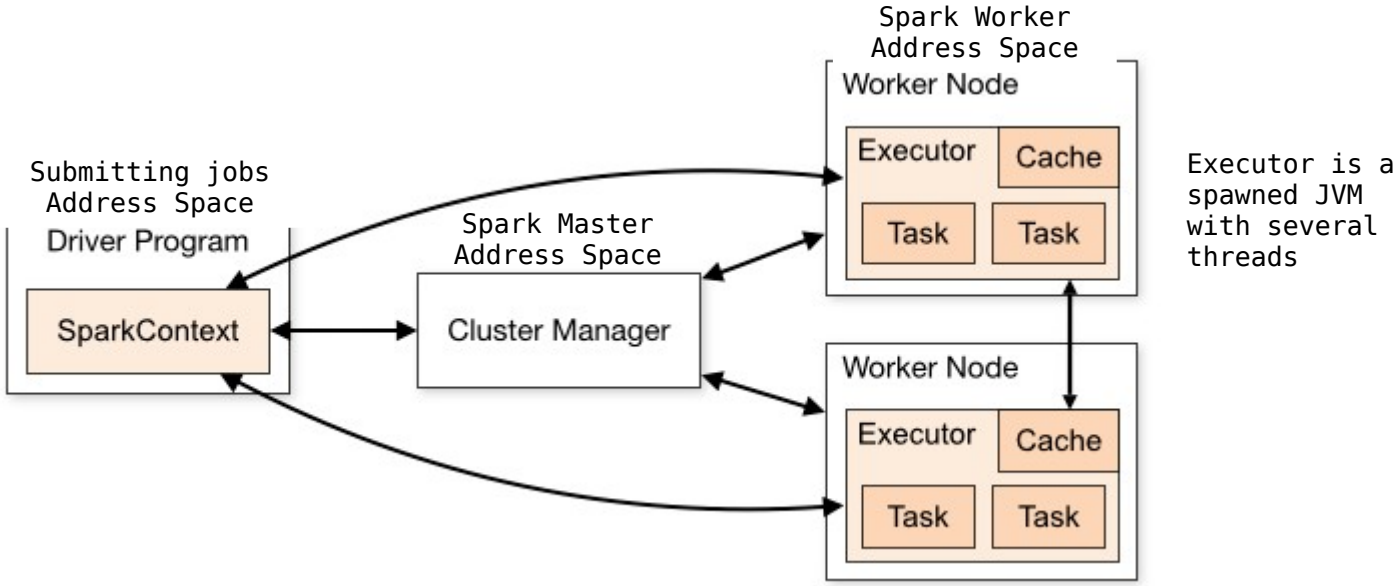
- Spark does not optimize SQL queries
- Spark is not a mechanism to store data, but rather provides interfaces to access portions of required data & most importantly to apply analytics using a unified interface
- IDAA interaction with applications is via DB2 z/OS paradigm; Spark interaction with applications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

Apache Spark and z/OS Data Sources



Apache Spark Architecture

- A *Driver Program* contains the code that will be executed, for example a Java or Scala program. This code will establish a SparkContext
- Communication is via TCP/IP between the *Driver*, *Master and Worker*
- The Cluster Manager will try to secure resources for the job to run on from the workers
- Workers will spawn Executors to run the Spark job



Apache Spark Sample Code

- This was executed in the interactive scala shell
- Mainframe Data Service used to access a z/OS Sequential Data Set
  - Mapping created for the data set in the data studio
  - Virtual Table then available to Apache Spark via SparkSQL
- Mapping and code done by a Marist Intern

```
scala> val dfReader =  sqlContext.read.format("jdbc").option("driver","com.rs.jdbc.dv.DvDriver")

scala> dfReader.option("url", "jdbc:rs:dv://xxxx.xxxx.ibm.com;DBTY=DVS;user=asofia;password=xxxxx")

scala> dfReader.option("dbtable", "SMF_03000")

scala> val df = dfReader.load

scala> df.head
```

Apache Spark Sample Code

- This was executed in the interactive scala shell with the Mainframe Data Service used to access a z/OS Sequential Data Set
  - Mapping created for the data set in the data studio
  - Virtual Table then available to Apache Spark via SparkSQL
- Code finds the number of z/OS jobs that had product usage information
  - z/OS SMF data is directly accessed to perform the query

```
scala> val df =
sqlContext.read.format("jdbc").option("driver","com.rs.jdbc.dv.DvDriver").option("url",
"jdbc:rs:dv://xxx.yyy.ibm.com;DBTY=DVS;user=asofia;password=xxxxx").option("dbtable",
"SMF_03000").load
df: org.apache.spark.sql.DataFrame = [SMF_LEN: bigint, SMF_ZERO: bigint, SMF_FLAG: string, SMF_RTY:
int, SMF_TIME: timestamp, SMF_SID: string, SMF_SSI: string, SMF_STY: int, SMF_SEQN: string, SMF30SOF:
bigint, SMF30SLN: bigint, SMF30SON: bigint, SMF30IOF: bigint, SMF30ILN: bigint, SMF30ION: bigint,
SMF30UOF: bigint, SMF30ULN: bigint, SMF30UON: bigint, SMF30TOF: bigint, SMF30TLN: bigint, SMF30TON:
bigint, SMF30COF: bigint, SMF30CLN: bigint, SMF30CON: bigint, SMF30AOF: bigint, SMF30ALN: bigint,
SMF30AON: bigint, SMF30ROF: bigint, SMF30RLN: bigint, SMF30RON: bigint, SMF30POF: bigint, SMF30PLN:
bigint, SMF30PON: bigint, SMF30OOF: bigint, SMF30OLN: bigint, SMF30OON: bigint, SMF30EOF: bigint,
SMF30ELN: bigint, SMF30EON: bigint, SMF30EOR: bigint, SMF30RVD: bigint, SMF30EOS: bigint, SMF30DRO:
b...
scala> df.filter("SMF_STY = 5").count
res30: Long = 541
scala> df.filter("SMF_STY = 5").filter("SMF30UDN > 0").count
res31: Long = 17
```

Thank You!