Embracing Spark as the Scalable Data Analytics Platform for the Enterprise

Matthew J. Glickman

GS.com/Engineering

Spark Summit East 2015

we BUILD

Goldman Sachs

How did we get here today?

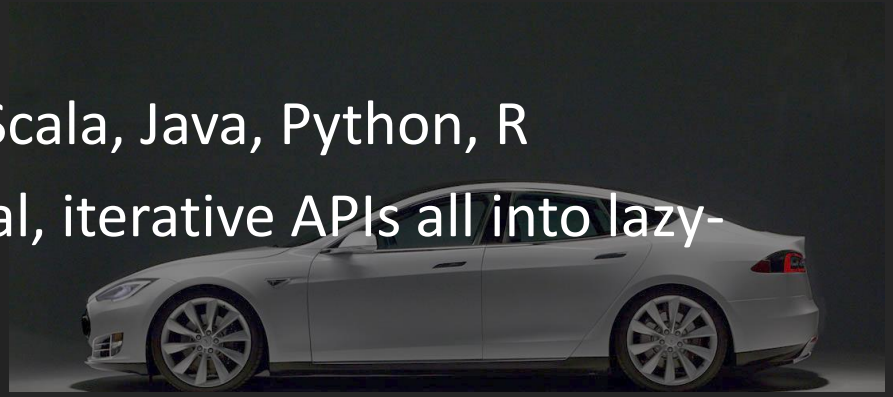
- 

- “Mixing Structured Data and Analytics with Spark SQL”
  Michael Armbrust
- “Make Simple things simple and complex things possible”
  Alan Kay (by way of Ali Ghodsi Spark Summit 2014 demo)
- Spark write-up went viral on our internal social media platform

we BUILD

Goldman Sachs

I'd Seen the Future in Apache Spark…



- Intuitive language bindings to Scala, Java, Python, R
- Combining relational, functional, iterative APIs all into lazy-evaluation data pipelines
- Storage agnostic
- Lambda closures
- Similar abstraction to GS internal platform's very successful tabular dataset framework
- Scala was already becoming a viable GS platform

we BUILD

Goldman Sachs

What makes Big Data scalable?

- Elasticity in 3 dimensions:
  - Data Storage = _____.
  - Compute = _____.
  - Users = _____.

we BUILD

Goldman Sachs

Compute Elasticity = 

- But first, how many people have…
  - Used a proprietary data analytics framework?
  - Written their own data analytics framework?
  - Wrapped their own framework around Spark?

we BUILD

Goldman Sachs

Don’t wrap Spark!

- Power of Spark is in the API abstractions (e.g., RDD, DataFrame)
- Spark is becoming “Lingua Franca” of Big Data analytics
- Contribute to open source instead of wrapping!

we BUILD

Goldman Sachs

Everyone is building Data Lakes

- Universal data acquisition makes all big data analytics and reporting easier
- Hadoop provides a scalable storage with HDFS
- How will we scale consumption and curation of all this data?

we BUILD

Goldman Sachs

There was a dream that was [Spark]…

- Embrace Spark as the elastic data consumption and curation engine to harness the power of the Data Lake
- All Data Lake datasets available as Spark RDD DataFrames
- Achieve data transformation lineage
  - Data Lake manages DAG of all datasets transformation dependencies
  - Spin up CPU-segregated Spark clusters to compute and store curated data back to Data Lake

we BUILD

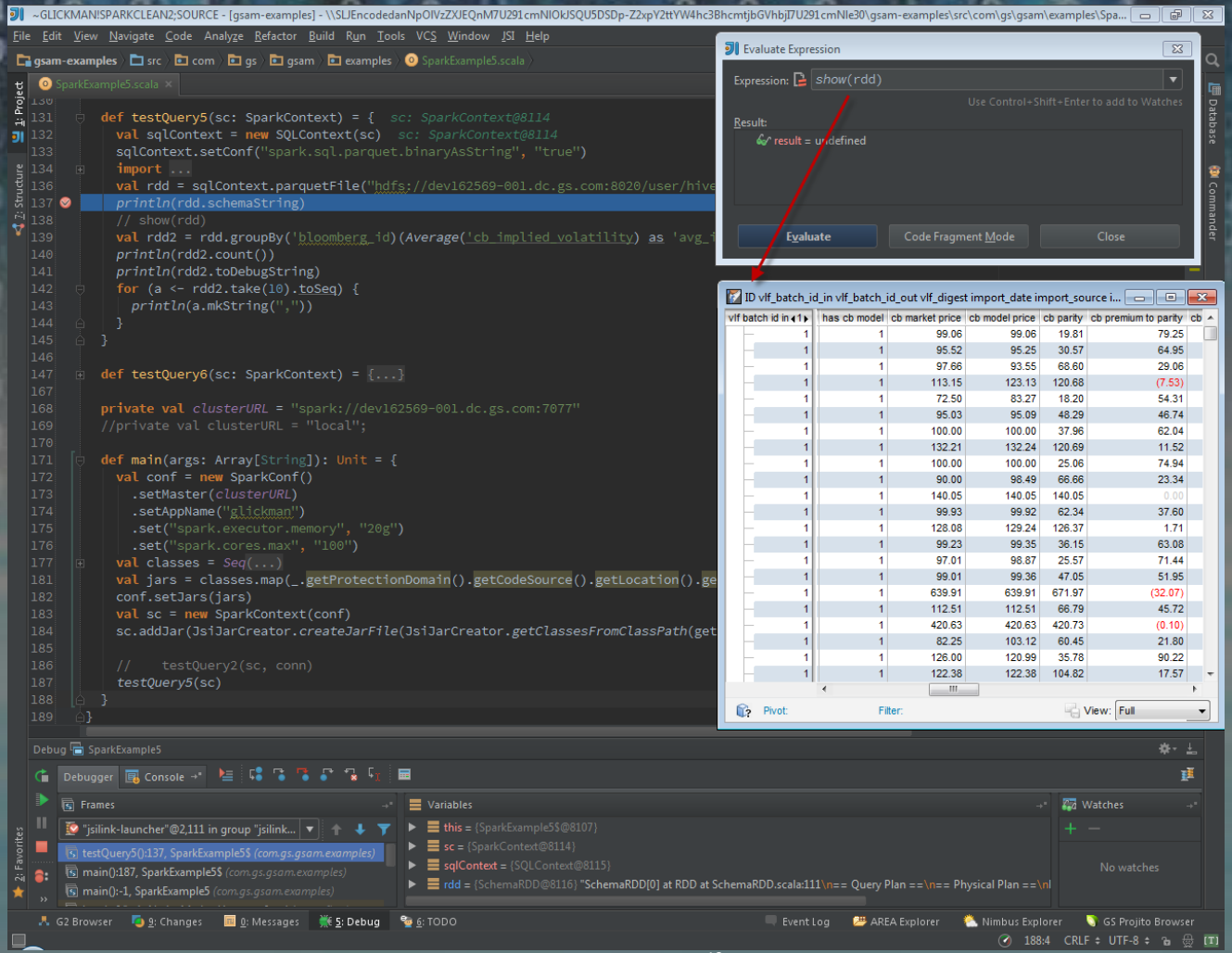
What about Integration?

- Embed Spark driver code directly inside JVM applications just like any other Scala library
- Leverage existing SDLC using existing JVM IDE environment instead of spark-submit for easier debugging
- Dynamically deploy code to cluster at run-time with lambda closures and:

```
val sc = new SparkContext(conf)
sc.addJar(JarCreator.createJarFile(JarCreator.
    getClassesFromClassPath(getClass.getPackage.getName)))
```

- Allow multiple applications with different code wavefronts to share Spark Clusters

we BUILD


~GLICKMAN!SPARKCLEAN2;SOURCE - [gsam-examples] - \\SLJEncodedanNpOIVzZXJEQnM7U291cmNlOkJSQU5DSDp-Z2xpY2ttYW4hc3BhcmtjbGVhbjI7U291cmNlNIe30\gsam-examples\src\com\gs\gsam\examples\Spa...
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window JSI Help
gsam-examples src com gs gsam examples SparkExample5.scala
SparkExample5.scala
def testQuery5(sc: SparkContext) = { sc: SparkContext@8114
val sqlContext = new SQLContext(sc) sc: SparkContext@8114
sqlContext.setConf("spark.sql.parquet.binaryAsString", "true")
import ...
val rdd = sqlContext.parquetFile("hdfs://dev162569-001.dc.gs.com:8020/user/hive
println(rdd.schemaString)
// show(rdd)
val rdd2 = rdd.groupBy('bloomberg_id)(Average('cb_implied_volatility) as 'avg_i
println(rdd2.count())
println(rdd2.toDebugString)
for (a <- rdd2.take(10).toSeq) {
println(a.mkString(","))
}
}
def testQuery6(sc: SparkContext) = {...}
private val clusterURL = "spark://dev162569-001.dc.gs.com:7077"
//private val clusterURL = "local";
def main(args: Array[String]): Unit = {
val conf = new SparkConf()
.setMaster(clusterURL)
.setAppName("glickman")
.set("spark.executor.memory", "20g")
.set("spark.cores.max", "100")
val classes = Seq(...)
val jars = classes.map(_.getProtectionDomain().getCodeSource().getLocation().ge
conf.setJars(jars)
val sc = new SparkContext(conf)
sc.addJar(JsiJarCreator.createJarFile(JsiJarCreator.getClassesFromClassPath(get
// testQuery2(sc, conf)
testQuery5(sc)
}
}
Evaluate Expression
Expression: show(rdd)
Use Control+Shift+Enter to add to Watches
Result:
result = undefined
Evaluate
Code Fragment Mode
Close
ID vlf_batch_id_in vlf_batch_id_out vlf_digest import_date import_source i...
vlf batch id in
has cb model
cb market price
cb model price
cb parity
cb premium to parity
cb
1 1 99.06 99.06 19.81 79.25
1 1 95.52 95.25 30.57 64.95
1 1 97.66 93.55 68.60 29.06
1 1 113.15 123.13 120.68 (7.53)
1 1 72.50 83.27 18.20 54.31
1 1 95.03 95.09 48.29 46.74
1 1 100.00 100.00 37.96 62.04
1 1 132.21 132.24 120.69 11.52
1 1 100.00 100.00 25.06 74.94
1 1 90.00 98.49 66.66 23.34
1 1 140.05 140.05 140.05 0.00
1 1 99.93 99.92 62.34 37.60
1 1 128.08 129.24 126.37 1.71
1 1 99.23 99.35 36.15 63.08
1 1 97.01 98.87 25.57 71.44
1 1 99.01 99.36 47.05 51.95
1 1 639.91 639.91 671.97 (32.07)
1 1 112.51 112.51 66.79 45.72
1 1 420.63 420.63 420.73 (0.10)
1 1 82.25 103.12 60.45 21.80
1 1 126.00 120.99 35.78 90.22
1 1 122.38 122.38 104.82 17.57
Pivot: Filter: View: Full
Debug SparkExample5
Debugger Console
Frames
"jsilink-launcher"@2,111 in group "jsilink...
testQuery5():137, SparkExample5$ (com.gs.gsam.examples)
main():187, SparkExample5$ (com.gs.gsam.examples)
main():-1, SparkExample5 (com.gs.gsam.examples)
Variables
this = {SparkExample5$@8107}
sc = {SparkContext@8114}
sqlContext = {SQLContext@8115}
rdd = {SchemaRDD@8116} "SchemaRDD[0] at RDD at SchemaRDD.scala:111\n== Query Plan ==\n== Physical Plan ==\n
Watches
No watches
G2 Browser 9: Changes 0: Messages 5: Debug 6: TODO Event Log AREA Explorer Nimbus Explorer GS Projito Browser
188:4 CRLF UTF-8

Goldman Sachs

What are the Integration Challenges?

- Getting machines provisioned to run Spark
- Should Spark and HDFS be run on the same cluster?
  - CPU segregation versus minimizing I/O
  - Data replication for segregation
- Library version synchronization of all open source libraries between Spark, HDFS as well as driver applications
  - Hadoop vendors only offer a partial solution

we BUILD

Is Enterprise Data going to Public Cloud?

- Managed cloud data services can provide:
  - Centralized management of clusters
  - Offer choice of Hadoop/Spark versions
  - Same SQL, Spark client APIs
  - Elastic scale and self-service automation

- Why not Public Cloud for Enterprise Data:
  - Regulation
  - Single Cloud Provider Lock-in
  - Data Encryption
  - User entitlement provisioning

we BUILD

Goldman Sachs

ETL is still the big problem

- Data needs to be ingested into scalable storage
- Each enterprise will build its own Data Lake
- Moving data is HARD!
- A lot of this data is coming from the same external vendors who are dumping their databases to file feeds for each customer
- Lots of resources at each enterprise are spent reconstructing these vendor databases in Data Lakes alongside internal enterprise data

we BUILD

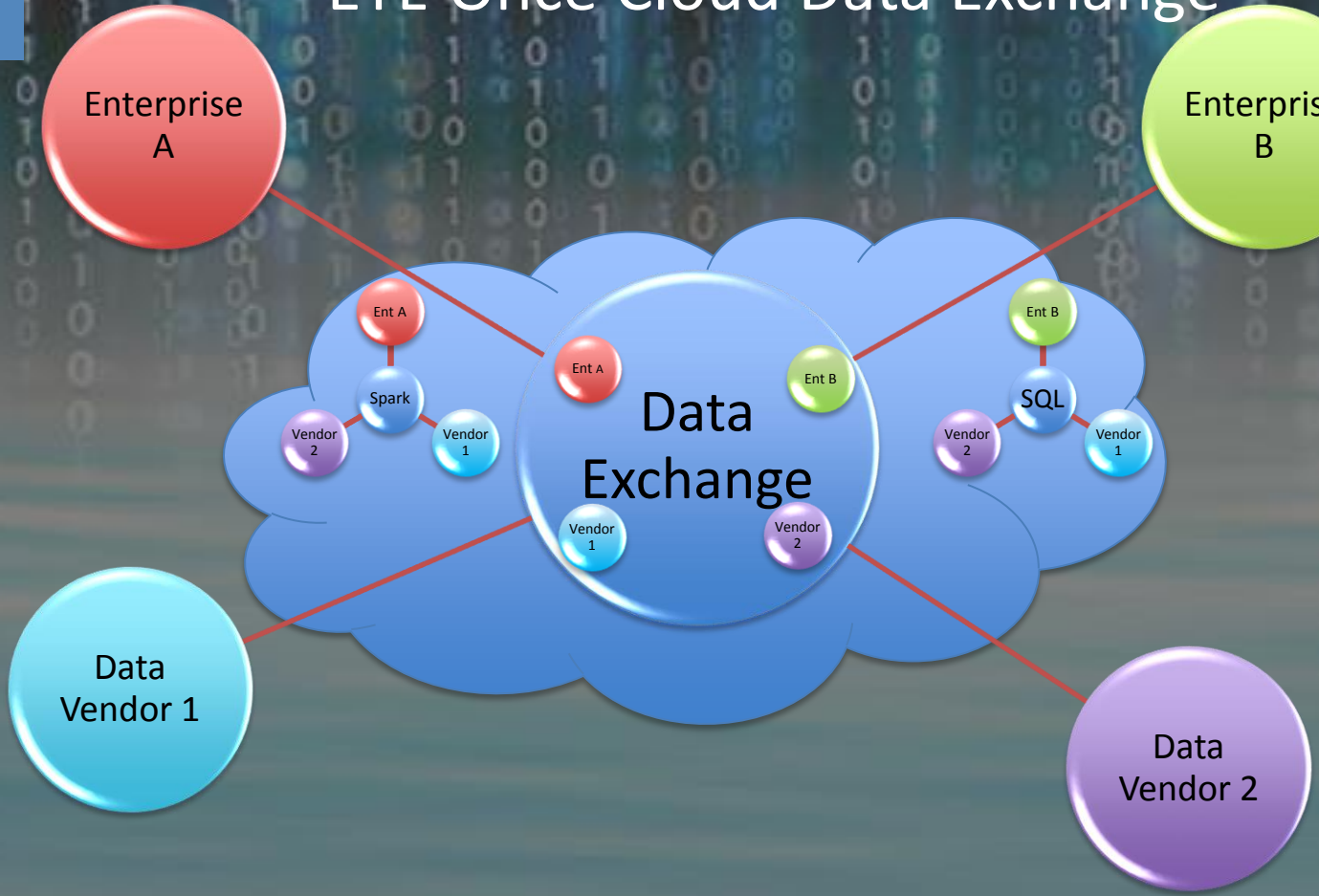
Goldman Sachs

Game Changer: Cloud Data Exchanges

- Managed cloud data services have the market disrupting potential to become Cloud Data Exchanges:
  - "ETL-once" loading of vendor data by vendors
  - Scalable compute near data with no persistent data movement
  - Vendors can provision access to their data directly to customers
  - No need for each enterprise to ETL the same vendor data
  - Enterprises could then load their data securely alongside vendor data for analytical consumption via standard APIs
- This kind of game changing managed cloud data service will be what really tips the enterprise public cloud data scale

we BUILD

ETL-Once Cloud Data Exchange



Goldman Sachs

Takeaways

- Build muscle memory for easier open source contribution
- Think of Spark Client API like ODBC/JDBC
- Embrace don’t wrap Spark APIs to prepare for accelerated move of enterprise data to Public Cloud

we BUILD

Goldman Sachs

we BUILD

Learn more at GS.com/Engineering

The term 'engineer' referenced in this section is neither a licensed engineer nor an individual offering engineering services to the general public under applicable law.

These materials ("Materials") are confidential and for discussion purposes only. The Materials are based on information that we consider reliable, but Goldman Sachs does not represent that it is accurate, complete and/or up to date, and it should not be relied on as such. The Materials do not constitute advice nor is Goldman Sachs recommending any action based upon them. Opinions expressed may not be those of Goldman Sachs unless otherwise expressly noted. As a condition to Goldman Sachs presenting the Materials to you, you agree to treat the Materials in a confidential manner and not disclose the contents thereof without the permission of Goldman Sachs. © Copyright 2014 The Goldman Sachs Group, Inc. All rights reserved.