Challenges Deploying Advanced Analytics Using Spark in a Shared Infrastructure

Nick Werstiuk
Director Strategy and Offering Management
werstiuk@ca.ibm.com

April 12, 2016
Challenges Deploying Spark in a Shared Infrastructure

Objectives

• Provide a point of view on the 4 key challenges that we hear from our clients as we work with them to establish their shared Spark environments

• Draw some conclusions about the impact of those challenges on the choices/options that clients make

• Speak to the evolution required for HPC environments to manage these workloads effectively
What is Spark?

Spark is a fast, expressive, cluster computing system

**Unified Analytics Platform**

- Spark SQL
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)

**Superior flexibility / agility**
- Unified and far simpler programming model
- 10x faster app development
- Multi-language (Scala, Java, Python)

**Superior efficiency / structure**
- 100x vs in-memory map-reduce, 10x on disk
- 2x streaming, 2x SQL vs OSS competitors
- 2.5x less code
- Better infrastructure choice

**Rich set of certifications**

- Pivotal
- Intel
- Hortonworks
- IBM
- BlueData
- SAP
- Alibaba
- DataStax
- Oracle
- Guardicore

**Aggressive community**

Most active Apache project: >300 contributors; 50 orgs, 32 committers

Quickly growing Spark Summit: 2013 (450), 2014 (>1000), …

Most active contributors: Nov 11th to now

<table>
<thead>
<tr>
<th>Org</th>
<th>Commits</th>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataBricks</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>Intel</td>
<td>272</td>
<td>8</td>
</tr>
<tr>
<td>Yahoo</td>
<td>70</td>
<td>5</td>
</tr>
<tr>
<td>Cloudera</td>
<td>240</td>
<td>3</td>
</tr>
<tr>
<td>Huawei</td>
<td>152</td>
<td>6</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>136</td>
<td>1</td>
</tr>
<tr>
<td>Tencent</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>IBM (new)</td>
<td>18</td>
<td>1</td>
</tr>
</tbody>
</table>

**Quickly expanding ecosystem**

- ADATAO
- Alpine
- Apervi
- Atigeo
- DIYOTTA
- Elasticsearch
- MicroStrategy
- lokki
- INUBE
- Pentaho
- Qlik
- talend
- tresata
- Typesafe
- QOMRA

Source: [http://spark.apache.org](http://spark.apache.org)
1. Diverse Users and Applications

- Multiple groups and lines of business with diverse range of data scientists & analyst end users
- Different application types (notebooks, streaming, batch, interactive query)

Shared Spark Infrastructure

Heterogeneous Infrastructure

- CPU/GPU
- Power
- x86
- Docker
- VM
- Cloud
2. Dynamic and Agile

- Significant workload variability, throughput and performance across the users and application types (streaming vs. interactive query vs. batch)
- ‘Bring your own’ notebooks, libraries and rapid pace of Spark evolution drive complex application life cycle management needs
3. Cost efficiency and SLA’s

- Avoid silos by user, group, applications
- Apply the right resources to the right workload
- Offer a cost effective shared service to a wide range of users – predictable throughput and performance to ensure SLA
4. Expansion to Multiple Data Sources and Frameworks

- Data Inputs and Outputs - existing Hadoop and HDFS pools but applications will also depend on a wider range of data outside of existing Hadoop
- Frameworks – ability to expand the shared infrastructure to the ecosystem of frameworks and applications beyond Spark
+ Some Future Challenges

CPU and GPU Management

- Increasing opportunity to leverage GPU and CPU to run Spark workloads
- Allocation and management of CPU and GPU tasks to get ‘best’ application outcome

Efficient use and re-use of in memory datasets

- Expand data aware scheduling from Disk aware to Memory aware
- Enable shared RDD across multiple users to improve throughput by reusing data all ready in memory
Conclusions

• Enterprises have an expectation of cloud like, multi-tenant Spark experience
  • End users are getting that type of experience from public clouds – need to have similar capabilities internally
  • Deliver flexibility of a personal/dedicated environment for users while managing as a cost effective shared service

• Performance and SLA are key due to wide range and variety of Spark workloads
  • Sophisticated resource and workload management to utilize the infrastructure in the best way across a wide variety of Spark use cases and applications
  • Ensure SLA’s while managing cost and TCO of infrastructure

• ‘Future proof’ the environment so that it can evolve and scale in multiple dimensions
  • Versions of Spark, notebooks, libraries, scale of underlying infrastructure, new data sources, resource types and application frameworks
IBM Platform Conductor for Spark

**Spark Distribution**

- Spark Core and RDDs
- Spark SQL
- Spark Streaming
- MLlib Machine Learning Library
- GraphX

**Platform Conductor Core Value-Add Technology**

- Platform Spark EGO Plug-In
- Zeppelin and plug-in
- Platform Enterprise Resource Manager 3.3
- Platform Application Service Controller
- Spark Master/Session Scheduler
- Enterprise Mgmt UI, Reporting & Security

**IBM Spectrum Scale**

- Spectrum Scale FPO

**Qualified Support for:**

- HDFS, Cassandra, OpenStack Swift, Amazon S3

**Open Source Components**

- Value Add Components

**IBM STC Spark Distribution**

- An end-to-end enterprise-grade multi-tenant, multi-version, multi-instance support - eliminates cluster sprawl.

- Increased performance and scale for multi-user, multi-tenant shared environments

- Fine grain, dynamic allocation of resources maximizes efficiency and utilization

- Enterprise workload management with guaranteed SLA, complete service/package life cycle management

- Enterprise class Management UI, Monitoring, Reporting & Security

- Distributed data store supporting POSIX, HDFS, Object interface with information life cycle management
Thank you

Nick Werstiuk
werstiuk@ca.ibm.com

ibm.com/platformcomputing