Exploiting Spark for high-performance scalable data engineering and data-science on Microsoft Azure

Simon Field : Data ( Architect | Engineer | Hacker ) @ Microsoft

How to find me......

Simon Field
Data ( Architect | Engineer | Hacker)
sifield@microsoft.com
linkedin.com/in/fieldey6961
Fieldy6961
What is Spark?
# Common Spark Use-Cases

<table>
<thead>
<tr>
<th>Use-Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Big Data&quot; ETL</td>
<td>Large datasets requiring data pre-processing</td>
</tr>
<tr>
<td><strong>Flexible/Shared Data Platform</strong></td>
<td>Scalable Data Store for data in multiple formats; tabular, semi-structured, un-structured, network/graph.</td>
</tr>
<tr>
<td><strong>Machine Learning</strong></td>
<td>Scaling beyond a single core, server &amp; memory</td>
</tr>
<tr>
<td></td>
<td>Parallel Machine Learning Models</td>
</tr>
<tr>
<td></td>
<td>Pleasingly Parallel</td>
</tr>
<tr>
<td></td>
<td>- Hyper-Parameter tuning, feature-selection, Cross-Validation, Simulations etc</td>
</tr>
<tr>
<td><strong>Ad-Hoc Big Data Analytics</strong></td>
<td>Reporting / Rollup Analytics with BI Tools</td>
</tr>
<tr>
<td></td>
<td>Adhoc SQL</td>
</tr>
<tr>
<td><strong>Stream Processing</strong></td>
<td>&quot;Realtime&quot; Processing of Streaming data</td>
</tr>
<tr>
<td></td>
<td>Making data available to analyze in Seconds</td>
</tr>
<tr>
<td></td>
<td>Connection to messaging services; Event Hubs / Kafka</td>
</tr>
</tbody>
</table>
Apache Spark

An unified, open source, parallel, data processing framework for Big Data Analytics

Spark Unifies:
- Batch Processing
- Interactive SQL
- Real-time processing
- Machine Learning
- Deep Learning
- Graph Processing
- Open APIs;
  - Java/Scala, Python, R, SQL

Spark SQL
Interactive Queries

Spark MLlib
Machine Learning

Spark Streaming
Stream processing

GraphX
Graph Computation

Spark Core Engine

Yarn

Mesos

Standalone Scheduler
Spark on Azure
WHY APACHE SPARK ON AZURE?

Flexibility
Collaboration
Performance
Scale-Up
Scale-Out
Common Language, API & DSLs
Data Integration
Service Integration

Spark Core Engine
Spark SQL
Interactive Queries
Spark MLlib
Machine Learning
Spark Streaming
Stream processing
GraphX
Graph Computation

Yarn
Mesos
Standalone Scheduler

AZURE SPARK OPTIONS
HDINSIGHT

Microsoft
The R eco-system for Apache Spark

- **Apache Spark**
  - SparkR
  - RStudio
    - sparklyr
    - dplyr
    - Mleap
    - graphframes
  - H2O
    - RSparkling (H2O sparkling-water)

- **Microsoft**
  - RevoScaleR
  - MicrosoftML
  - mmlspark
  - dplyrXdf
  - azuresmr
  - Other...
    - sparkavro
    - spark.sas7bdat
...
Spark on the Azure Data Science Virtual Machine
Do you ever just need a more powerful laptop…?

**Azure Data Science Virtual Machines (DSVM)**

- Windows and Linux (Ubuntu & CentOS)
- Includes comprehensive set of data science, Azure tools/SDKs
- Linux (CentOS) instance includes Spark Standalone + HDFS
- Rstudio, Jupyter and other programming tools pre-installed
- Available on Azure Marketplace – updated regularly
- Pay for cloud hardware usage only. No software charges!

Build your first model in 30 minutes or less!
Try the DSVM for Free today...
http://aka.ms/dsvm
http://aka.ms/dsvmhandout
http://aka.ms/dsvmtenthings
How big a “laptop” do you need?! 

<table>
<thead>
<tr>
<th>INSTANCE</th>
<th>vCPU</th>
<th>RAM</th>
<th>TEMPORARY STORAGE</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2 v3</td>
<td>2</td>
<td>16.00 GB</td>
<td>32 GB</td>
<td>€0.2024/hour</td>
</tr>
<tr>
<td>E4 v3</td>
<td>4</td>
<td>32.00 GB</td>
<td>64 GB</td>
<td>€0.4048/hour</td>
</tr>
<tr>
<td>E8 v3</td>
<td>8</td>
<td>64.00 GB</td>
<td>128 GB</td>
<td>€0.8105/hour</td>
</tr>
<tr>
<td>E16 v3</td>
<td>16</td>
<td>128.00 GB</td>
<td>256 GB</td>
<td>€1.6209/hour</td>
</tr>
<tr>
<td>E32 v3</td>
<td>32</td>
<td>256.00 GB</td>
<td>512 GB</td>
<td>€3.2409/hour</td>
</tr>
<tr>
<td>E64i v3</td>
<td>64</td>
<td>432.00 GB</td>
<td>864 GB</td>
<td>€6.1106/hour</td>
</tr>
<tr>
<td>E64 v3</td>
<td>64</td>
<td>432.00 GB</td>
<td>864 GB</td>
<td>€6.1089/hour</td>
</tr>
</tbody>
</table>

Prices are exclusive of VAT. Monthly price estimates are based on 730 hours of usage.

https://docs.microsoft.com/en-us/azure/cloud-services/cloud-services-sizes-specs
AzureSMR Package – Create & start new DSVM

```bash
# CREATE RESOURCE GROUP AND DEPLOY NEW DSVM
azureCreateResourceGroup(sc, RESOURCEGROUP, REGION) # Create Resource Group
azureDeployTemplate(sc, deplname = DEPLNAME,
    templateURL=TEMPLATEURI,
    paramJSON = PARAMETERJSON,
    resourceGroup = RESOURCEGROUP, verbose = TRUE)
```

E.g. Windows DSVM “one-box”

Template Library Link
AzureSMR Package – Start an existing DSVM

Start an existing DSVM

```bash
# If the status of the VM is stopped, start the VM
if (status == "Provisioning succeeded, VM deallocated") {
    azureStartVM(azureActiveContext = sc,
                resourceGroup = RESOURCEGROUP,
                vmName = VMNAME,
                mode = "Async",
                subscriptionID = get_secret("AzKeys", vault = mdvaultDir)["SUBID"],
                verbose = FALSE) }

Start request submitted: 2017-09-12 20:13:50
[1] TRUE

# Check the status periodically to ensure it has started successfully
azureVMStatus(sc, RESOURCEGROUP, VMNAME, SUBID,
              ignore = "N", verbose = FALSE)

[1] "Updating, VM starting"

# Check the status periodically to ensure it has started successfully
azureVMStatus(sc, RESOURCEGROUP, VMNAME, SUBID,
              ignore = "N", verbose = FALSE)

[1] "Updating, VM running"
```
AzureSMR Package – Connect to remote DSVM

Multiple Ways to Connect and Execute R on the DSVM

```r
### MULTIPLE WAYS OF ACCESS THE DSVM TO WORK
# 1. Remote Desktop
# 2. Jupyter Notebook Service
# 3. R Server mrsdeploy package for remote access
# 4. VisualStudio + R Tool : Workspace (remote)
#   requires some additional configuration

# 1. Start a Remote Desktop onto DSVM
rdsCmd <- sprintf("mstsc /v:%s.%s.cloudbapp.azure.com /f", VMNAME, REGION)
shell(rdsCmd, wait = FALSE) # This works for Windows O.S. only. Use alternative
# RDP software for Linux/Mac clients

# 2. Open the Jupyter Notebook service running on DSVM from the local Browser
dsvm_jupyter_url <- sprintf("https://%s.%s.cloudbapp.azure.com:9999/", VMNAME, REGION)
browseURL(dsvm_jupyter_url,
    browser = getOption("browser"),
    encodeIfNeeded = FALSE)

# 3. R Server mrsdeploy package
library("mrsdeploy") # Load the mrsdeploy package
# Login to a R server using mrsdeploy package
# Can log into either Linux, Windows, SQL Server or Hadoop based R Server instances
dsvm_mrsdeploy_url <- sprintf("http://%s.%s.cloudbapp.azure.com:12800", VMNAME, REGION)
remoteLogin(dsvm_mrsdeploy_url, prompt = "Remote> ",
    session = TRUE, diff = FALSE, commandline = TRUE,
    username = get_secret("smrsfrg2", vault = mdvaultDir)["MRSDEPLOYADMINUSER"],
    password = get_secret("smrsfrg2", vault = mdvaultDir)["MRSDEPLOYADMINPASS"] )
```
AzureSMR Package – Stop DSVM

Stop a DSVM

```r
> # STOP VM
> (status <- azureVMStatus(sc, RESOURCEGROUP, VMNAME, SUBID,
+     ignore = "N", verbose = FALSE))

[1] "Provisioning succeeded, VM running"
> if(status == "Provisioning succeeded, VM running") {
+ azureStopVM(azureActiveContext = sc,
+     resourceGroup = RESOURCEGROUP,
+     vmName = VMNAME,
+     mode = "Sync",
+     subscriptionID = key_get("azureSubsc","azureSubsc"),
+     verbose = FALSE)
+ }

Request Submitted: 2017-09-10 11:04:52
Kev: R - running. (..) - deallocating. D - deallocated, + - starting, S - stopped
R++++++++++++++++++++++++++++++++++++++++++++D
> azureVMStatus(sc, RESOURCEGROUP, VMNAME, SUBID,
+     ignore = "N", verbose = FALSE)
+ 
[1] "Provisioning succeeded, VM deallocated"
```
Running **Spark** on the Azure Databricks Platform

**Azure Databricks**

**Collaborative Workspace**
- DATA ENGINEER
- DATA SCIENTIST
- BUSINESS ANALYST

**Deploy Production Jobs & Workflows**
- MULTI-STAGE PIPELINES
- JOB SCHEDULER
- NOTIFICATION & LOGS

**Optimized Databricks Runtime Engine**
- DATABRICKS I/O
- APACHE SPARK
- SERVERLESS
- Rest APIs

*Enhance Productivity*  
*Build on secure & trusted cloud*  
*Scale without limits*
Azure Databricks – Workspace

Split data into training/test sets

We randomly split our data into a training set (using 70% of the total data) and a validation set (30%). We also persist the training set to the cluster’s memory, since this will speed up the ML training.

Note that `randomSplit` creates a list of size 2 where the first element is the training set and the second element is the test set.

Running an ML Model

We run a logistic regression model below where:
- `isTipped` is the response variable
- `payment_type`, `passenger_count`, and `timeOverDistance` are the predictor variables

It should be noted that there are other models that can be run on Spark like GBM, Decision forests, Neural nets, etc.
Why Spark on Azure Databricks

A fast, easy and collaborative Apache® Spark™ based analytics platform optimized for Azure

- Designed in collaboration with the founders of Apache Spark
- One-click set up; streamlined workflows
- Interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.
- Native integration with Azure services (Power BI, SQL DW, Cosmos DB, Blob Storage)
- Enterprise-grade Azure security (Active Directory integration, compliance, enterprise-grade SLAs)
NYC Taxi Trip Dataset Demo
- Comparing DSVM and Databricks
DSVM - Spark Taxi “Big Data” processing example

Joining the [NYC Taxi Journey Dataset](#) on a DS13v2 - 8 Core – 56GB memory

- Trip (173,179,759 rows in 12 csv files, 28.8GB) \( \leftarrow \) JOIN \( \rightarrow \) Fare (173,179,759 rows in 12 csvs files, 19.7GB total) = \( \sim \)49GB RESULT

```r
# Using SparkR syntax to join NYC Taxi Trips to Fares
dataSetsdf <- drop( join(taxifare, taxitrip, 
taxifare$tf_medallion==taxitrip$tt_medallion & 
taxifare$tf_hack_license==taxitrip$tt_hack_license & 
taxifare$tf_pickup_datetime==taxitrip$tt_pickup_datetime), 
c("tt_medallion", "tt_hack_license", "tt_pickup_datetime"))
tic()
head(df)

<< FULL OUTPUT TRUNCATED >>
```

Note: Completed but could not cache whole Spark dataframe in memory

\~ 9 minutes elapsed
- Can we scale this any better on a 4 node Databricks cluster, using the same VM Size?
DSVM - Spark Taxi “Big Data” processing example

2.6 Minutes to run Join on 4 nodes
- 7.5 mins on DSVM / 2.6 mins on Databricks = ~3x speedup with 4x more resource
HDInsight - Spark
Running Spark on Azure HDInsight

- **HDInsight Cluster Types**
  - Spark: Spark, ML Server
  - Plus: Hadoop, Hbase, Storm, Kafka, Hive LLAP
- **IDE**
  - Rstudio Server – R Server
  - Jupyter Notebook (PySpark & Scala kernels)
- **Operationalization via Web-Services**
- **Spark Shell on Edge Node(s).**
- **Programmatic access via Spark APIs.**
- **Typically used for persistent processing/cluster needs**
- **Easily install/customize**
  - runs on all nodes via custom-script post cluster start
  - SparkR, Microsoft packages pre-installed
Azure Distributed Data Engineering Toolkit (AZTK)
Running **Spark** On Azure Batch With AZTK

- **aztk:** Command-line interface to provision Spark clusters on Azure Batch service
  - [www.github.com/azure/aztk](http://www.github.com/azure/aztk)
  - Choice of VM instance types
  - Use provided Docker instances - includes SparkR, SparklyR and Rstudio (or Bring Your Own)

- **Provision a Spark cluster in ~5 minutes**
  ```
  aztk spark cluster create --id fieldyspark4 --size 4
  ```

- **Easily connect to RStudio Server and Spark DSLs from browser**
  ```
  aztk spark cluster ssh --id fieldyspark4
  http://localhost:8787  # Connect to RStudio
  ```

- **Tools to connect and manage persistent storage**

- **Pay for VM usage only**
  - Per minute charging model - optionally, choose low-priority nodes to save costs
### Spark on Azure - When to use what?

<table>
<thead>
<tr>
<th></th>
<th>DSVM</th>
<th>AZTK</th>
<th>HDInsight</th>
<th>Databricks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strengths:</strong></td>
<td>Easy / Single User</td>
<td>Operationalization</td>
<td>Full Hadoop eco-system</td>
<td>Easy / Collaboration / Security / Performance</td>
</tr>
<tr>
<td><strong>Workload Type:</strong></td>
<td>Interactive + Jobs</td>
<td>Interactive + Jobs</td>
<td>Interactive + Jobs</td>
<td>Single + Jobs + Team interaction</td>
</tr>
<tr>
<td><strong>Costs:</strong></td>
<td>$</td>
<td>$$</td>
<td>$$$</td>
<td>$$$</td>
</tr>
<tr>
<td><strong>Spark Experience:</strong></td>
<td>Pre-Configured/ Customise</td>
<td>Vanilla – Customise Docker</td>
<td>Pre-Configured</td>
<td>Prescriptive</td>
</tr>
<tr>
<td><strong>Azure Service Integration:</strong></td>
<td>Somewhat</td>
<td>Somewhat</td>
<td>Very</td>
<td>Very</td>
</tr>
<tr>
<td><strong>Infrastructure:</strong></td>
<td>Cores + GPUs</td>
<td>Cores + GPUs</td>
<td>Cores + GPUs Multi-Node</td>
<td>Cores Multi-Node Serverless</td>
</tr>
</tbody>
</table>
R on Azure – Other Options......

**“Tooling & Processing”**
- Data Science Virtual Machine for Linux (Ubuntu)
- Data Science Virtual Machine for Windows
- Machine Learning Workspace
- Machine Learning Web Service

**“Big Processing”**
- HDInsight
- Data Lake Analytics
- Batch Service
- Azure Databricks

**“Data Storage”**
- Storage account - blob, file, table, queue
- Data Lake Store
- SQL Database
- SQL Data Warehouse
- Azure Cosmos DB
- SQL Server
Create your free Azure account today

Get $200 free credit
Start free with $200 in credit, and keep going with free options.

Try any Azure services
Explore our cloud by trying out any combination of Azure services for 30 days.

Pay nothing at the end
We use your credit card information for identity verification, but you’ll never be charged unless you choose to subscribe.

https://azure.microsoft.com/en-us/free/
Getting Started / Questions

Data Science Virtual Machine
- aka.ms/dsvm
- aka.ms/dsvmhandout
- aka.ms/dsvmtenthings
- Provisioning DSVM

azureSMR
- AzureSMR Github site
- Azure R Quickstart Templates
- Azure Quickstart Templates (All)

Blogs
- Azure Spark Toolkit
- Analysing NYC Taxi using Spark and Microsoft R

This Presentation
- Link to Presentation

Azure Account - $200 free credit
https://azure.microsoft.com/en-us/free/

How to find me.......
THANK YOU!