BIG-06

Native Hadoop technologies and how to get the most out of them with Qlik

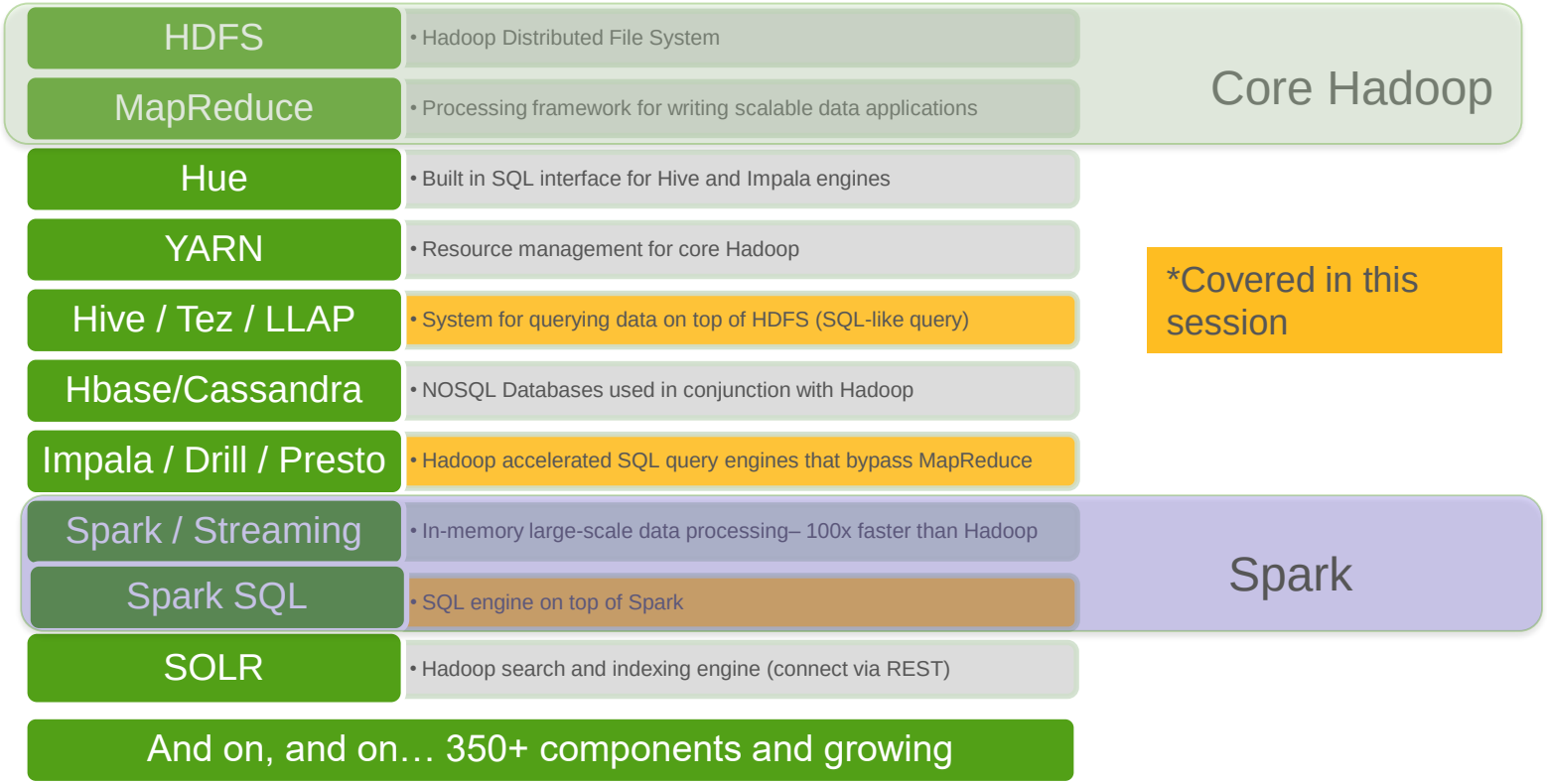
David Freriks – Technology Evangelist

February 2017

Agenda

- Review of Native SQL options for Hadoop
- Query & Application Options for Qlik
- Query Engine Review
  - Hive on MapReduce
  - Hive on Spark (Cloudera)
  - Hive on Tez/LLAP (Hortonworks)
  - Impala
  - Drill (MapR
  - SparkSQL
  - Presto

“Big Data” Refresher - Hadoop Native Interfaces



Options for Qlik apps on Big Data sources

- **In-Memory**: Traditional Qlik load scripts taking source data into memory. Limited by volumes and size, QVD's can also help w/ improving reloads.

- **Hybrid Query**: Mixing in-memory and direct query, care has to be taken to understand direct discovery and cardinality limitations.

- **Direct Query**: Good option for fast systems, will lose advanced capabilities (i.e. SET analysis), but can push supported calculations back to source engine.

Options for Qlik apps on Big Data sources (con't)

- **On Demand Options:**
  - **ODAG** (on-demand app generation): Summary Qlik app at higher level passing back selections through and extension to a detail app which runs on-demand against source data.
  - **API**: Using the API's and mashups to generate the detail app on demand against source data.

- **Server Side Extensions**: Ability to use 3rd party calculation engine functions natively in Qlik by round-tripping data through the API to target system (i.e. Spark)

Hive on MapReduce - Summary

- Hive can be used to analyze large datasets in filesystems such as HDFS or S3. The SQL language is called HiveQL, which uses schema on read to issues queries against MapReduce to process requests.
- HiveQL is somewhat ANSI-SQL92,2003,2011 compliant, but does not support the full range of SQL capabilities for such operations as subqueries, complex insert and create capabilities, or indexing as you would find in a traditional relational database.
- HiveQL does add a layer of complex unstructured data processing and transformation capabilities not possible with regular databases.

Hive on MapReduce – Qlik's Review

Review of Hive on MapReduce:								
	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity
Hive-MR	fair	good	common	slowest	fair	good	low	high

QLIK RECOMMENDATIONS for Hive on MapReduce:

PRO(s): Hive on MapReduce is a proven technology that supports most ANSI SQL compliant features and a robust extension capability to query any type of data effectively.

CON(s): Speed. MapReduce is slow and there is no caching or way to optimize, every query starts from ground zero.

USAGE: Qlik In-Memory applications with processed data refreshed daily or on a longer time window. Not suitable for real-time or even hourly updates (on large data volumes), or really any kind of exploratory analytics with live database interaction.

Qlik Recipe Guide for Hive on MapReduce						
	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
Hive-MR	Best	No	No	No	No	No

Hive on Spark (Cloudera only)

- This is a recent modification of the core Hadoop construct by replacing MapReduce with Spark as the job processing engine.

- This creates an up to 10x performance boost in job processing time, but is important to note – HiveQL is still used in this scenario, not SparkSQL and there is a cost to convert RDD's into row containers (more on that in a bit).

- This is a considerable improvement on baseline MapReduce (and very well may spell the end of that technology), however, it is still not as fast as other query technologies.

Hive on Spark – Qlik's Review

Review of Hive on Spark (*currently Cloudera Only):								
	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity
Hive-Spark	fair	good	common	fair to good	fair	good	medium	medium

QLIK RECOMMENDATIONS for Hive on Spark:

PRO(s): Hive on Spark is a slam dunk for batch jobs previously run on MapReduce if you are using a Cloudera distribution. There is very little downside or negative impact on existing queries and jobs. At the writing of this article it was Cloudera's recommendation to convert to Hive on Spark for CDH 5.7+.

CON(s): Immaturity and increased hardware cost can be possible, depending on initial cluster configuration. If you are using a distribution other than Cloudera – you will need to find another option.

USAGE: Qlik **In-Memory** and potentially well-tuned **On-Demand** applications. Still not suitable for near/real time queries, but getting closer to closing the gap to interactive analytics on Hadoop.

Qlik Recipe Guide for Hive on Spark						
	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
Hive-Spark	Best	No	No	If Tuned	If Tuned	No

Qlik

Hive on Tez/LLAP (Hortonworks only)

- A different take on Hive/MapReduce is using a technology called Tez which uses Yarn to manage and execute multiple MapReduce jobs simultaneously to boost query performance.

- Tez breaks the biggest bottleneck of MapReduce (single threaded jobs) adds optimizers to commonly used capabilities like joins, allows in-memory processing vs disk, and many other enhancements to greatly increase SQL performance.

- LLAP (Long Last and Process) was added in Hive 2.0 as an enhancement to the core execution engine. LLAP uses a persistent daemon to process certain queries directly while directing harder queries back to Yarn (via Tez) to run.

Hive on Tez/LLAP – Qlik's Review

Review of Hive on Tez with LLAP (*Hortonworks only):								
	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity (LLAP)
Hive-Tez	good	good	common	good	good	good	medium	medium

QLIK RECOMMENDATIONS for Hive on Tez with LLAP:

PRO(s): If you are using Hortonworks, you are already using Tez – LLAP is the next step in increasing the performance. The attraction of this solution is that it requires no additional work for the end users – Hive queries will just run faster, waaaay faster (Hortonworks states 25x improvement over Hive on MapReduce).

CON(s): Maturity. While Tez is a proven commodity, LLAP is a shiny new technology that was released mid-2016. Also, like Spark, you will have to have machines with significantly more memory than base Hive on MapReduce to fully utilize LLAP.

USAGE: Qlik **In-Memory, Direct Query, On Demand** applications will perform well with Tez&LLAP. **Hybrid** and **Server Side Extensions** may also be used with tuning and appropriate use cases.

Qlik Recipe Guide for Hive on Tez with LLAP						
	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
Hive-Tez	Yes	If Tuned	Yes	Yes	Yes	If Tuned

Impala

- Taking a different tact than trying to upgrade Hive, Cloudera created their own query optimization engine specifically targeting Business Intelligence users.
- Impala until recently was a proprietary engine for Cloudera, but has now been open sourced to the Apache foundation and adopted by other major distributions MapR and Amazon EMR.
- What makes Impala different is that it bypasses MapReduce entirely and uses its own MPP engine.

Impala – Qlik's Review

Review of Impala:								
	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity
Impala	high	good	common	good	good	good	high	high

QLIK RECOMMENDATIONS for Impala:

PRO(s): Impala is one of the standard benchmarks for SQL on Hadoop queries. Well documented, and well supported – it's the most proven common accelerator available if you have MapR, EMR, or Cloudera distributions.

CON(s): Cost and Capability. Impala has steep hardware requirements to achieve maximum throughput and scalability and doesn't have all the features available via Hive.

USAGE: Qlik **In-Memory, Direct Query, On Demand** applications will perform well with Impala. **Hybrid** applications may also be used with tuning and appropriate use case; however, Server Side Extension functionality does not exist as a capability with Impala.

Qlik Recipe Guide for Impala						
	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
Impala	Yes	If Tuned	Yes	Yes	Yes	No

Drill (MapR Only)

- Drill is another MPP type execution engine that queries Hadoop data with a very important distinction… It also can connect to NoSQL databases (HBase, MongoDB, etc), S3 (and other cloud storage data), and JSON.

- Drill has more flexibility to work with unstructured data than any of the other query engines discussed until now.

- The biggest detractor for Drill is primarily market penetration. MapR is the smallest of the major distributions and therefore doesn't have the install base of the other query engines for reference, support, and benchmarking.

Drill – Qlik's Review

Review of Drill (*primarily for MapR customers):								
	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity
Drill	unknown	unknown	special	good	very good	great SQL	high	medium

QLIK RECOMMENDATIONS for Impala:

PRO(s): Solid technology for MapR customers with ability to federate NoSQL, JSON, and cloud data. Fully ANSI SQL support which is very nice for business users.

CON(s): Adoption, finding benchmarks and comparisons are very tough to find compared to all other technologies in this class. The lack of market penetration/awareness will make finding people to implement Drill difficult, plus, generally only MapR customers would use the technology.

USAGE: Qlik **In-Memory, Direct Query, On Demand** applications will perform well with Drill. **Hybrid** applications may also be used with tuning and appropriate use case; however, Server Side Extension functionality does not exist as a capability with Drill.

Qlik Recipe Guide for Drill						
	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
Drill	Yes	If Tuned	Yes	Yes	Yes	No

Spark

- Apache Spark is the biggest and most powerful technology to emerge in the “Big Data” world over the last few years.
- Spark is creating a revolution in cluster computing for Hadoop data by combining speed with advanced capabilities around graph processing and machine learning capabilities built into the core engine.
- The metrics are stunning; Spark is on average 100x faster than MapReduce when running in-memory or 10x faster on disk (i.e. the reason for Hive on Spark)
- SparkSQL is relatively new and less mature than other SQL engines

SparkSQL – Qlik's Review

Review of SparkSQL:								
	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity
Spark	medium	good	special	great	good	fair	high	low-medium

QLIK RECOMMENDATIONS for Spark:

PRO(s): Spark is fast and powerful, opening up new use cases for Qlik, especially with Server Side Extensions.

CON(s): Spark is the new king of Big Data, however, the SQL capabilities lag other technologies currently available. Spark SQL does not handle concurrency as well as other technologies, so scalability may be an issue.

USAGE: Qlik In-Memory, Direct Query, On Demand, and Server Side Extension applications will perform well with Spark. Hybrid applications may also be used with tuning and appropriate use cases.

Qlik Recipe Guide for SparkSQL						
	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
SparkSQL	Yes	If Tuned	Yes	Yes	Yes	Yes

Presto

- Created initially by Facebook, Presto aims to bridge the major gap in all the tools listed above, federation and query of data not singularly contained in Hadoop and allow full ANSI SQL support with speed acceptable for business users.

- The architecture is similar to Drill and Impala (MPP DBMS), but squarely aimed at the largest of data sets (25PB @ Netflix, 300PB @ Facebook) without forcing commitment to any particular Big Data distribution.

- Much like Impala, Presto also bypasses MapReduce and employs its own custom engine to manage and issue queries, but like Impala, will require its own hardware to run effectively.

Presto – Qlik's Review

Review of Presto:								
	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity
Presto	best	medium-good	common	good	great	good	medium-high	low

QLIK RECOMMENDATIONS for Presto:

PRO(s): Presto offers powerful SQL capabilities, distribution independence, and data federation capabilities that differentiate it from all others in this class.

CON(s): Limited deployments and its relatively new to the open source community make it a bit of a risk. Consideration must be given for custom hardware requirements and you do pay a performance price to for data federation and keeping the data in source origin.

USAGE: Qlik In-Memory, Direct Query, and On Demand, applications will perform well with Presto. Hybrid applications may also be used with tuning and appropriate use cases. Server Side Extensions are not in scope for use with Presto.

Qlik Recipe Guide for Drill						
	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
Presto	Yes	If Tuned	Yes	Yes	Yes	No

Functional & Qlik Capability Summary

	Concurrency	Volumes	Skillsets	Speed	Flexibility	Functionality	Cluster Cost	Maturity
Hive-MR	fair	good	common	slowest	fair	good	low	high
Hive-Spark	fair	good	common	fair to good	fair	good	medium	medium
Hive-Tez	good	good	common	good	good	good	medium	medium
Impala	high	good	common	good	good	good	high	high
Drill	unknown	unknown	special	good	very good	great SQL	high	medium
SparkSQL	medium	good	special	great	good	fair	high	low-medium
Presto	best	medium-good	common	good	great	good	medium-high	low

	In-Memory	Hybrid Query	Direct Query	On Demand	API on Demand	Server Side Ext.
Hive-MR	Best	No	No	No	No	No
Hive-Spark	Best	No	No	If Tuned	If Tuned	No
Hive-Tez	Yes	If Tuned	Yes	Yes	Yes	If Tuned
Impala	Yes	If Tuned	Yes	Yes	Yes	No
Drill	Yes	If Tuned	Yes	Yes	Yes	No
SparkSQL	Yes	If Tuned	Yes	Yes	Yes	Yes
Presto	Yes	If Tuned	Yes	Yes	Yes	No

Summary and Resources

- All SQL engines are capable for sourcing data into Qlik, but depending on distribution – you have many options to consider.

Sources:

- Business Intelligence Benchmark Q4 2016 by AtScale
- Mastering Apache Spark 2.0 by Databricks
- Faster Batch Processing with Hive on Spark by Cloudera
- Hortonworks:
  - LLAP: Subsecond Analytical Queries in Hive
  - 100k Queries per hour with LLAP
  - Hive on Spark vs Tez
- Altoros: Hadoop Distributions: Evaluating Cloudera, Hortonwork, and MapR
- Putting Apache Drill into Production by MapR