SQL on Hadoop

Markus Olenius

BIGDATAPUMP Ltd

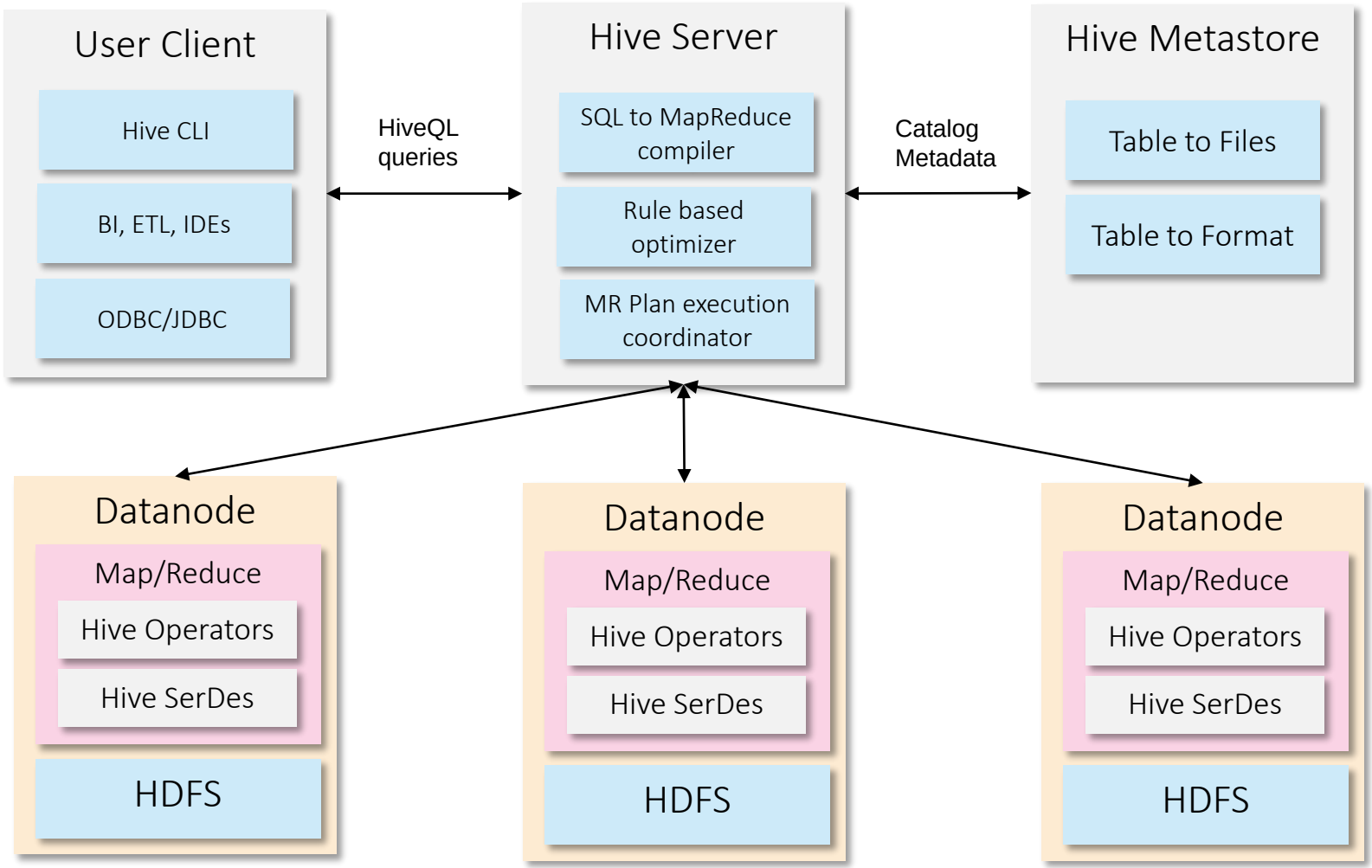
Background

- **Big Data vs. Traditional Data Warehousing**
  - Increased volumes and scale
  - Lower costs per node
  - Vendor lock-in avoidance, open source preference common
  - Agility over correctness of data
    - Schema on read vs. schema on write
- **Hadoop and MapReduce processing – support for scale, but with limitations**
  - No support for BI/ETL/IDE –tools
  - Missing features (optimizer, indexes, views)
  - Slow due to MapReduce latency
  - No schemas
  - Competence gap

What is SQL on Hadoop?

”A class of analytical application tools that combine established SQL-style querying with newer Hadoop data framework elements”

Classic Hive Architecture



Hive Classic

Hive Classic

- HiveQL is a SQL-based language, runs on top of MapReduce
- Support for JDBC/ODBC
- Familiar SQL like interface
- Economical processing for petabytes
- Logical schema – schema on read
- Missing features:
  - ANSI SQL
  - Cost based optimizer
  - Data Types
  - Security
- Hive Classic tied to MapReduce processing, leading to latency overhead

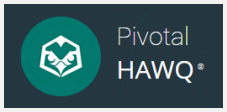
→ Slow but OK for batch SQL

SQL on Hadoop Landscape

Hive and Enhancements

MPP Query Engines

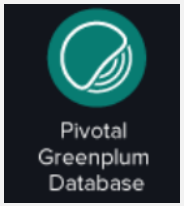
Impala



Big SQL

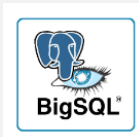
RDBMS on Hadoop

Vortex



Relying on PostgreSQL

CitusDB



Relying on HBase

SQL outside Hadoop

PolyBase

Others

Apache Kylin

Data Virtualization

Data Virtualization

Remarks on solution categories

1. MPP query engines
   - Don’t utilize MapReduce or utilize it only in some specific cases
   - Usually able to utilize Hive MetaStore
   - Typically support for ODBC and/or JDBC

2. RDBMS on Hadoop
   - RDBMS sharing nodes with Hadoop and utilizing HDFS
   - Typically proprietary file format for best performance
   - May require dedicated edge nodes

3. SQL outside Hadoop
   - RDBMS sends request to Hadoop
   - Processing work shared between Hadoop and RDBMS
   - Performance depends on network and RDBMS load

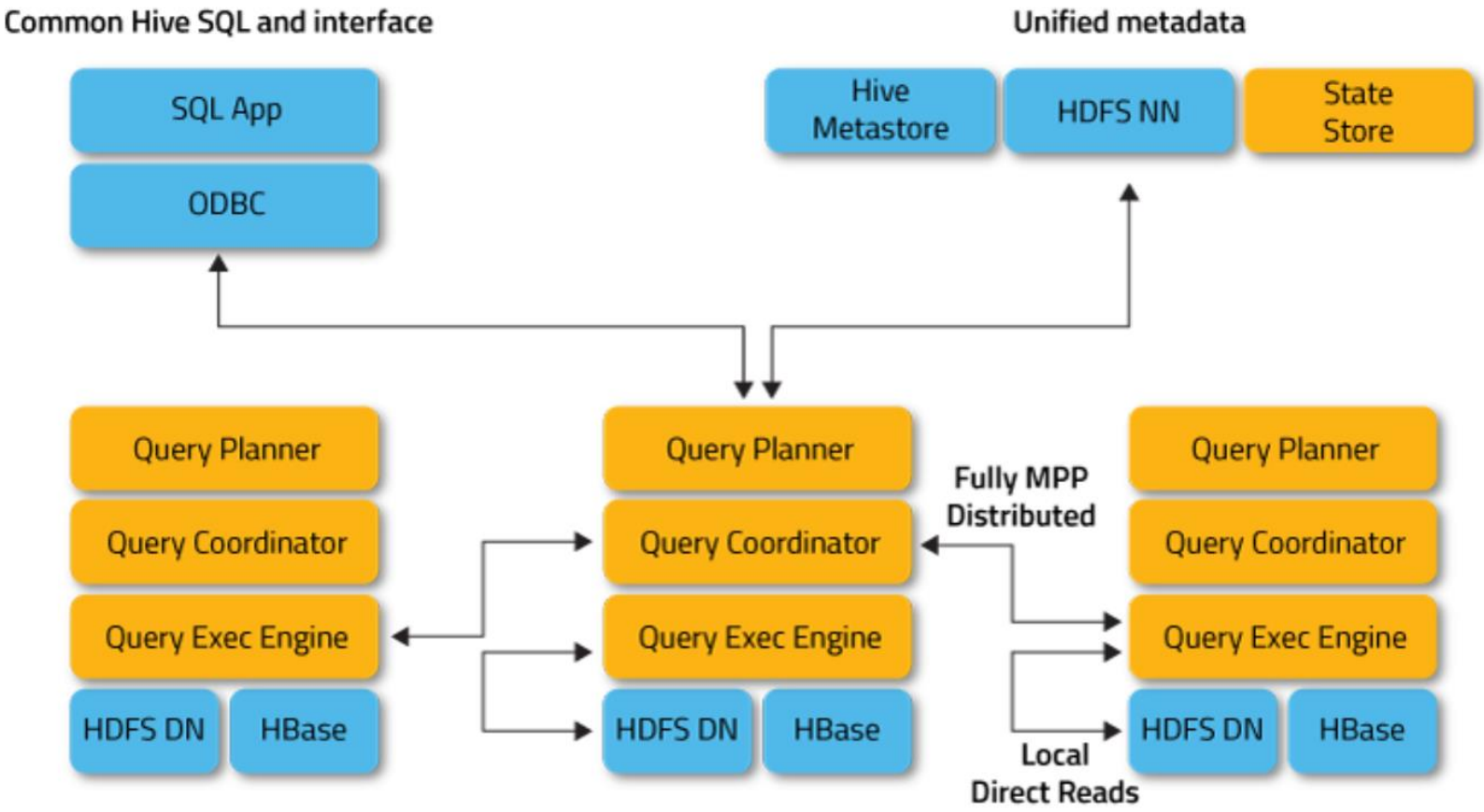
4. Data Virtualization solutions
   - Typically utilize Hive and possibly other MPP query engines
   - Similar use cases with SQL outside Hadoop

Need for Speed
- methods for getting better performance

1. Minimize I/O to get needed data
   - Partitioning and indexing
   - Using columnar file formats
   - Caching and in-memory processing

2. Query execution optimization
   - Better optimization plans, cost based optimization
   - Better query execution
   - Combining intermediate results more efficiently
     - Batch (MapReduce) vs. Streaming (Impala, HAWQ) vs. Hybrid (Spark) approaches

Example – Impala Architecture



File format selection

- Best performance with compressed and columnar data
- Trade-off - time and resources used for file format transformations
- Optimal performance and compression may require using proprietary file formats
- File format alternatives:
  - CSV, Avro, JSON
  - Columnar: ORC, Parquet, RCFile
  - Proprietary formats (e.g. Vertica, JethroData)

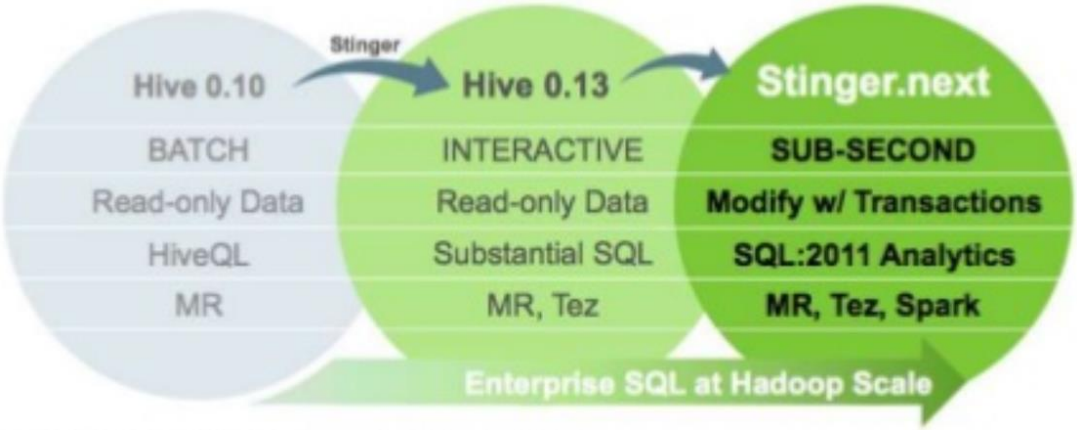
Overview of a few interesting tools

Stinger Initiative

Mainly supported by: Hortonworks

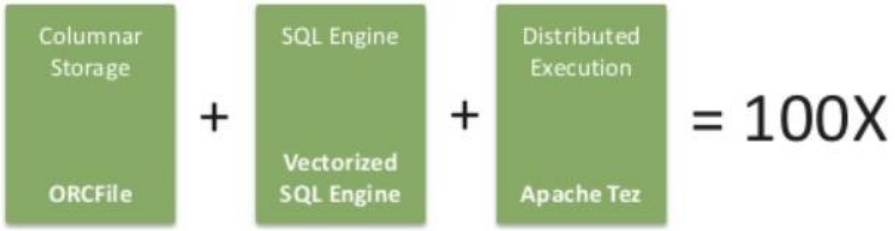
Hive 0.13

- from MapReduce to Tez
- Optimized for ORC file format
- Decimal, varchar, date
- Interactive queries



Stinger.next (Hive 0.14)

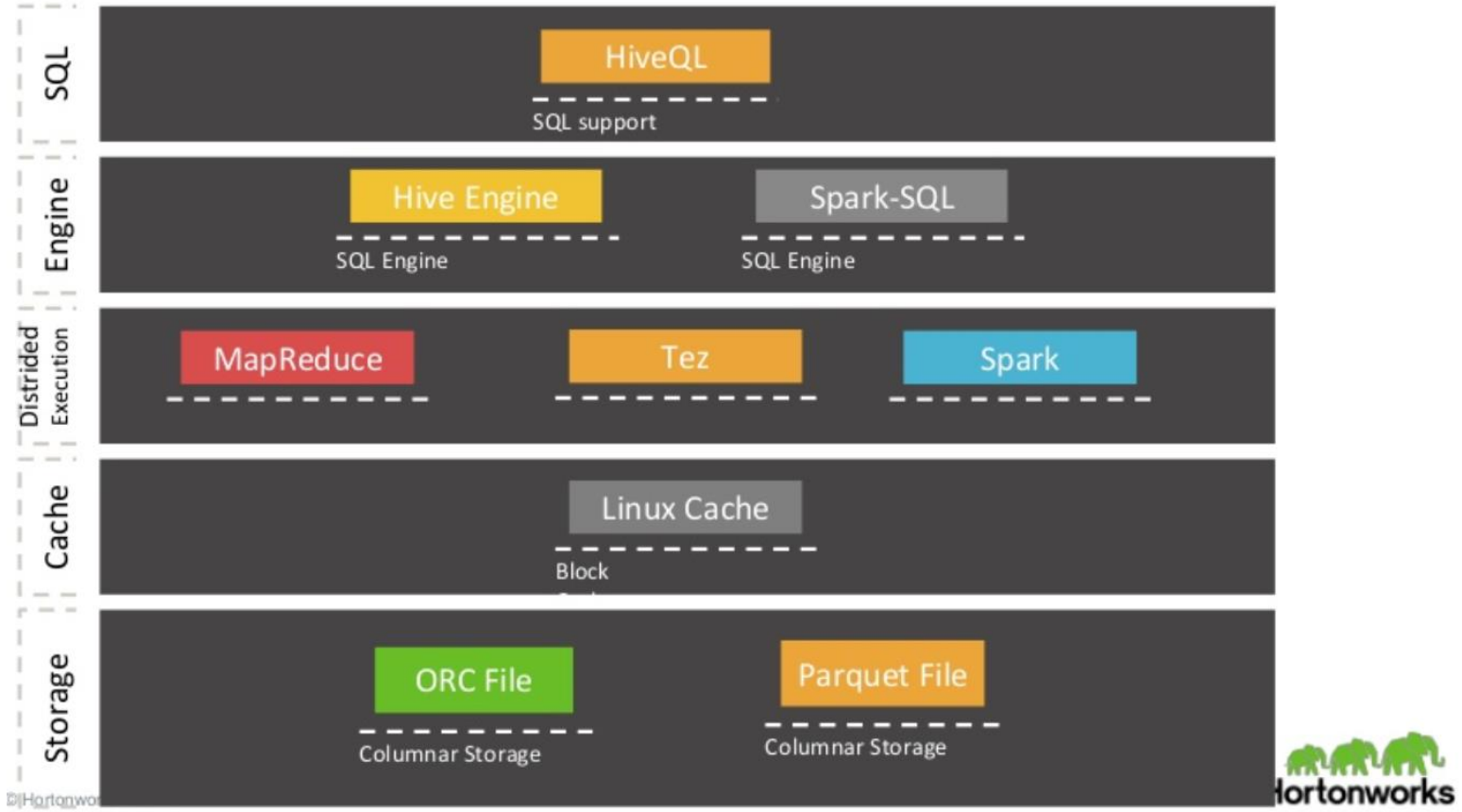
- ACID transactions
- Cost based query optimization



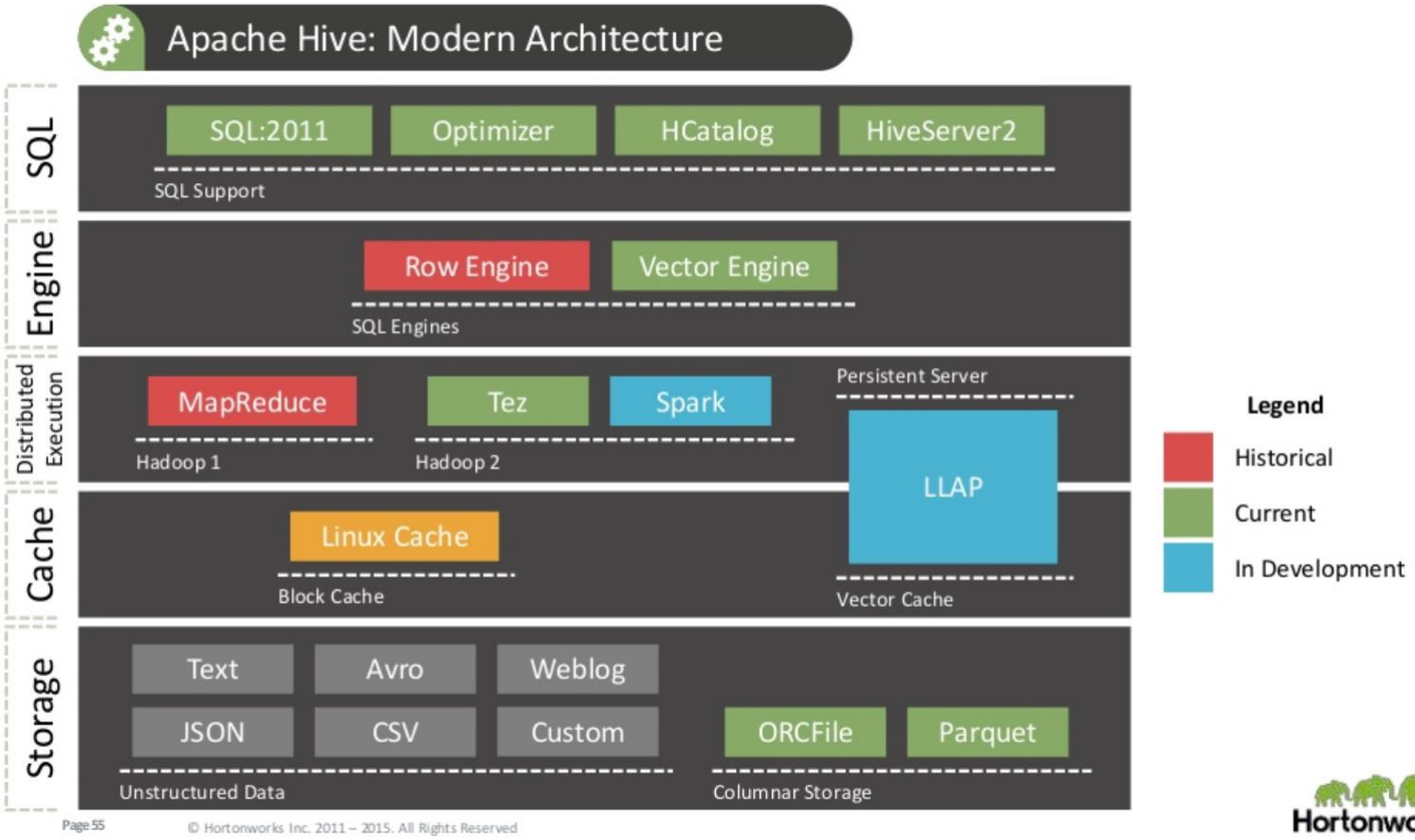
Open Source

Hortonworks

Stinger Initiative, Hive 0.13



Stinger.next, Hive 0.14



Impala

Mainly supported by: Cloudera

- An open source distributed SQL engine that works directly with HDFS
- Architected specifically to leverage the flexibility and scalability strengths of Hadoop
- Used query language is compatible with HiveQL and uses the same metadata store as Apache Hive
- Built in functions:
  - Mathematical operations
  - String manipulation
  - Type conversion
  - Date operations
- Operations that are not supported:
  - Hive user defined functions (UDFs)
  - Hive indexes
  - Deleting individual rows (overwriting data in tables and partions is possible)
- Available interfaces:
  - A command line interface
  - Hue, the Hadoop management GUI
  - ODBC
  - JDBC
- Supported file types:
  - Text files
  - Hadoop sequence files
  - Avro
  - Parquet

Open Source

Impala 2.1 and Beyond (Ships in 2015)

- Nested data – enables queries on complex nested structures including maps, structs, and arrays (early 2015)
- MERGE statement – enables merging in updates into existing tables
- Additional analytic SQL functionality – ROLLUP, CUBE, and GROUPING SET
- SQL SET operators – MINUS, INTERSECT
- Apache HBase CRUD – allows use of Impala for inserts and updates into HBase
- UDTFs (user-defined table functions) Intra-node parallelized aggregations and joins – to provide even faster joins and aggregations on on top of the performance gains of Impala
- Parquet enhancements – continued performance gains including index pages
- Amazon S3 integration

SparkSQL

- “Next generation Shark”
- Able to use existing Hive metastores, SerDes, and UDFs
- Integrated APIs in Python, Scala and Java
- JDBC and ODBC connectivity
- In-memory column store data caching
- Good support for Parquet, JSON
- Open Source

Apache Drill

Mainly supported by: MapR

- Open source, low latency SQL query for Hadoop and NoSQL
- Agile:
  - Self-service data exploration capabilities on data stored in multiple formats in files or NoSQL databases
  - Metadata definitions in a centralized store are not required
- Flexible:
  - Hierarchical columnar representation of data allows high performance queries
- Conforms to the ANSI SQL standards
- ODBC connector is used when integrated with BI tools
- Open Source

HP Vertica

Best supported by: MapR

- Enterprise-ready SQL queries on Hadoop data
  - Features included:
    - Database designer
    - Management console
    - Workload management
    - Flex tables
    - External tables
    - Backup functionality
  - Features that are not included:
    - Geospatial functions
    - Live aggregate projections
    - Time series analytics
    - Text search
    - Advanced analytics packages
- Grid-based, columnar DBMS for data warehousing
- Handles changing workloads as elastically as the cloud
- Replication, failover and recovery in the cloud is provided
- YARN not yet supported
- Data Model: Relational structured
- Transaction Model: ACID
- Commercial: Pay per term, per node

JethroData

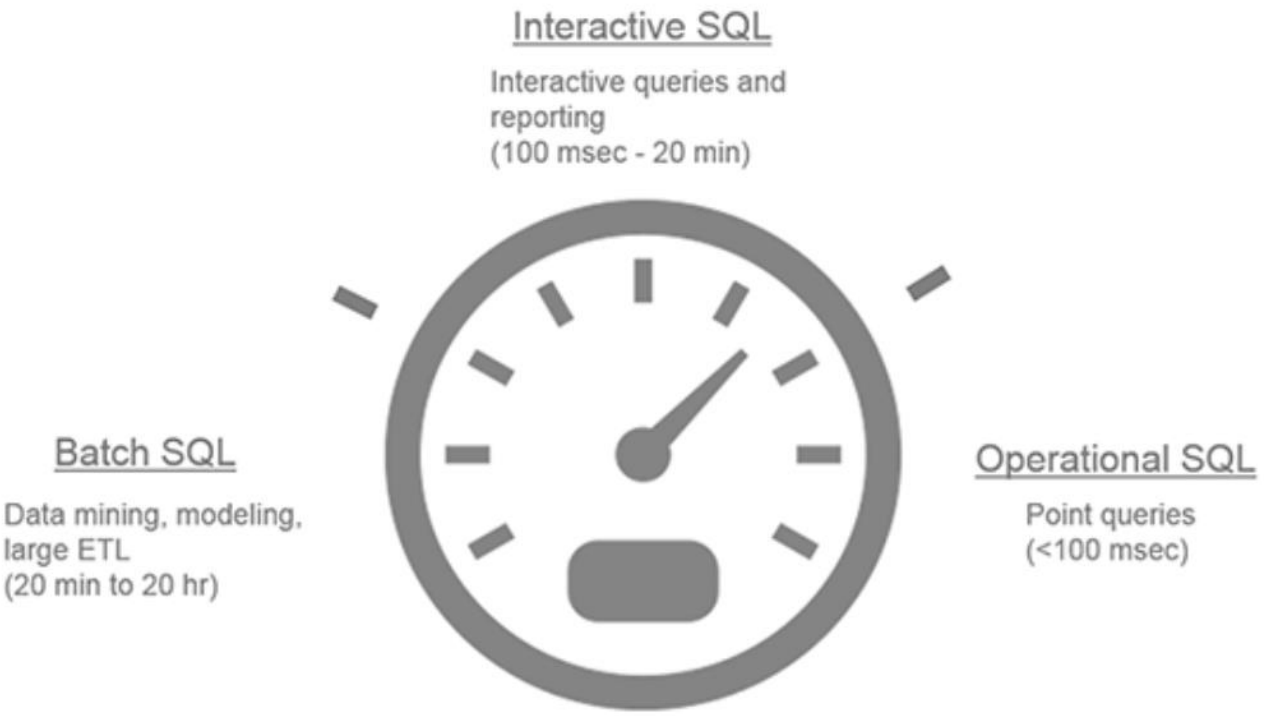
- Index based, all data is indexed
- Columnar proprietary file format
- Supports use with AWS S3 data
- Running on dedicated edge nodes besides Hadoop cluster
- Supports use with Qlik, Tableau or MicroStrategy through ODBC/JDBC
- Commercial

Demo with Qlik:

http://jethrodata.qlik.com/hub/stream/aaec8d41-5201-43ab-809f-3063750dfafd

Summary

Tool selection based on use case requirements



MAPR

https://www.mapr.com/why-hadoop/sql-hadoop/sql-hadoop-details

Tool Focus

Engine	Batch	Interactive	Point-querying
Hive	Yes	Yes (0.13->)	
Apache Tajo	Yes	Yes	
IBM BigSQL	Yes	Yes	
Hadapt		Yes	
RainStor		Yes	
Impala		Yes	limited with Hbase
Apache Drill		Yes	limited with Hbase
Presto		Yes	limited with Hbase
Apache Phoenix		Yes	Yes
JethroData		Yes	Yes
Splice Machine		Yes	Yes
HAWQ		Yes	Yes
Vertica SQL-on-Hadoop		Yes	Yes
Actian Vortex		Yes	Yes

http://www.datasalt.com/2014/04/sql-on-hadoop-state-of-the-art/
https://www.mapr.com/why-hadoop/sql-hadoop/sql-hadoop-details

How to select correct tool for your SQL-on-Hadoop use cases?

- **Long list of alternatives to choose from**
- **Many tools claim to be fastest in market (self-done evaluation tests)**
- **SQL and data type support varies**

→ It really depends on what your requirements for this capability are – at least for batch and interactive SQL use cases there are many tools that are probably “good enough”

→ Doing testing and evaluation with actual use cases is typically needed

→ Start with tools recommended or certified by selected Hadoop distribution

→ Consider also amount of concurrent use, not only single query performance

→ SQL-on-Hadoop capabilities are usually not a differentiator that should guide selection of Hadoop distribution

BIGDATAPUMP LTD

WWW.BIGDATAPUMP.COM