



Shark attack on SQL-on-Hadoop

Gerd König

(Big) Data Engineer

May 27th, 2014



Agenda

- SQL-on-Hadoop
 - Why that hype?
 - Tool overview and comparison
 - File formats matters
- Shark
 - Facts & figures
 - What makes the difference?
 - SparkSQL enters the playground
- Hands-On (quick 'n dirty)
 - File formats & disk usage
 - Execution times (at a rough estimate) / Benchmarking
- Summary



SQL-on-Hadoop - Why that hype?

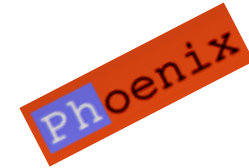
- Hadoop is widely accepted as “new technology”
- Hadoop gets more and more enterprise ready
- SQL is a well established language for many years and used by DB developers as well as Business Analysts

=> Huge demand for SQL(-like) access to Hadoop

SQL-on-Hadoop



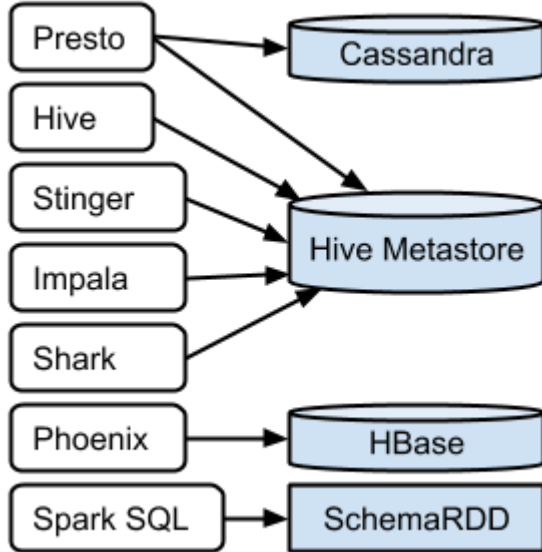
A whole bunch of tools (just an excerpt)



SQL-on-Hadoop



Clustering some tools

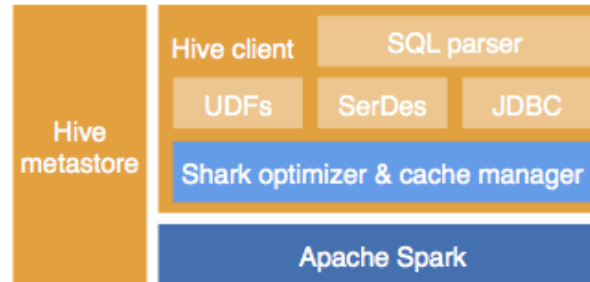


Hive	Batch SQL. Creates M/R jobs (YARN or MapReduce v1)
Stinger Phase3	Performance boosted Hive, additional functionality/data types, enables Hive on Tez, access via HTTP(S)
Impala, Presto	Low latency SQL. Distributed query execution engine, one Coordinator and many workers
Phoenix	SQL layer on Hbase, creates series of scans, delivered as client-embedded JDBC driver
Shark	Depends on Hive metadata, computes on RDDs by "extending" the Spark stack
Spark SQL	Access to several "metadata" sources via new SchemaRDD, full integration into Spark stack



Shark - Facts & figures

- ...sits on top of Apache Spark



- is tightly coupled with Hive, uses a slightly modified version
- use Hive statements, UDFs and Hive metastore (HCatalog)
- can be run in Shark-shell as well as Shark Server (connect e.g. via beeline JDBC client)



Shark / SparkSQL

- What makes the difference?
 - Performance increase due to in-memory processing ('low-latency M/R')
 - Interaction with other “Plugins” of the Spark stack, like ML-library, e.g. call ML functions directly with your SQL resultset:

```
val youngUsers = sql2rdd("SELECT * FROM users WHERE age < 20")
println(youngUsers.count)
val featureMatrix = youngUsers.map(extractFeatures(_))
kmeans(featureMatrix)
```

- SparkSQL - A new star is born?
 - no dependencies to Hive, new type of RDD “SchemaRDD”
 - fires SQL against RDDs, Parquet files, Hive (via Wrappers)



File format matters

- An appropriate file format influences
 - performance, and
 - used disk space

```
1, Smith, Joe, 40000;  
2, Jones, Mary, 50000;  
3, Johnson, Cathy, 44000;
```

Row store

```
1, 2, 3;  
Smith, Jones, Johnson;  
Joe, Mary, Cathy;  
40000, 50000, 44000;
```

Column store

- Use a columnar storage format for columnar data(bases)
 - RCFile, ORC, Parquet

Hands-On



- Part I
 - compare Parquet based table vs. flat file
- Part II
 - execute 1 query in Hive, Impala and Shark
 - get a feeling about runtime...



Further information

- Detailed Benchmarks by Berkeley AmpLab:
<https://amplab.cs.berkeley.edu/benchmark/>
- Shark
<http://shark.cs.berkeley.edu/>
- SparkSQL
<https://github.com/apache/spark/tree/master/sql>
<http://people.apache.org/~pwendell/catalyst-docs/sql-programming-guide.html>



THANKS for your attention !

Q&A

Gerd König

gerd.koenig@ymc.ch

Tel. +41 (0)71 508 24 74

 [@gerdkoenig](https://twitter.com/gerdkoenig)

 ch.linkedin.com/in/gerdkoenig