We put the SQL back in NoSQL*

@ApachePhoenix
http://phoenix.apache.org/
James Taylor (@JamesPlusPlus)

* Anybody else notice that all the NoSQL stores have SQL?
About James

• Architect at Salesforce.com
  – Part of the Big Data group
  – Lead of the Apache Phoenix project
What is Apache Phoenix?

• A relational database layer for Apache HBase
What is **Apache HBase**?

- High performance horizontally scalable byte store
- Suitable as store of record for mission critical data
What is Apache Phoenix?
What is Apache Phoenix?

• A relational database layer for Apache HBase
  – Query engine
    • Transforms SQL into native HBase API calls
    • Pushes work to cluster for parallel execution
  – Metadata repository
    • Typed access to data stored in HBase tables
    • Support multi-tenancy modeled as SQL views
  – A JDBC driver

• A top level Apache Software Foundation project
  – Originally developed at Salesforce
  – Now a top-level project at the ASF
  – A growing community with momentum
Where Does Phoenix Fit In?

- Phoenix: Query execution engine
- Pig: Data Manipulation
- Hive: Structured Query
- GraphX: Graph analysis framework
- MLLib: Data mining
- Sqoop: RDB Data Collector
- Flume: Log Data Collector
- HBase: Distributed Database
- YARN: Cluster Resource Manager / MapReduce
- Spark: Iterative In-Memory Computation
- Zookeeper: Coordination
- HDFS 2.0: Hadoop Distributed File System
- The Java Virtual Machine
- Hadoop Common JNI
Why is Phoenix fast?

- Pushes down computation to region servers
  - Start/stop key range(s)
  - Time range min/max
  - Predicates
  - Aggregation
  - Sort
  - Limit
  - TopN
- Parallelizes query from client
  - Intra-region through statistics collection
- Supports secondary indexes
  - Global & co-located
Why is Phoenix fast?

- Filter
- Aggregate
- Sort
- Limit
Skip Scans

- Push key range sets through filter
- Use SEEK_NEXT_HINT to skip data
Filters

• Pushes down WHERE clause to server
• Filters entire HFile based on time-range
What’s Next?

Big Data Landscape (Version 2.0)

You are here

© Matt Turck (@mattturck) and ShivonZilis (@shivonz) Bloomberg Ventures
Introducing Apache Calcite

• Query parser, compiler, and planner framework
  – SQL-92 compliant

• Pluggable cost-based optimizer framework
  – Sane way to model **push down** through rules

• Interop with other Calcite adaptors
  – Already used by Drill, Hive, Kylin, Samza
  – Supports any JDBC source (RDBMS - remember them 😊)
  – One cost-model to rule them all
How does Phoenix plug in?

JDBC Client

Calcite Parser & Validator

Calcite Query Optimizer

Phoenix Runtime

SQL + Phoenix specific grammar

Built-in rules + Phoenix specific rules
What’s Next?

Query

Drillbit
Calcite
Phoenix/ HBase

Drillbit
Calcite
HDFS

Drillbit
Calcite
RDBMS

Drillbit
Calcite
Samza/ Streams

?
Thank you!

Questions?

* who uses Phoenix