Where is Hadoop Going Next?

Owen O’Malley
owen@hortonworks.com
@owen_omalley

November 2014
Who am I?

- Worked at Yahoo Search
  - Webmap in a Week
  - Dreadnaught to Juggernaut to …

- Hadoop
  - MapReduce
  - Security

- Hive

- Apache/Open Source Champion

- PhD in Software Engr from UC Irvine
Topics

• **Hadoop History**
  “A beginning is the time for taking the most delicate care that the balances are correct.”
  - Herbert

• **Themes**
  – Storage
  – Computation
  – Security
What was the Problem?

• **Yahoo needed to build WebMaps faster**
  – Whole web analysis for Yahoo Search
  – WebMap in a Week

• **WebMap used Dreadnaught**
  – Roughly like MapReduce and HDFS
  – Scaled to 800 machines
  – Assigned nodes in backup pairs
  – Single application cluster

• **Started on C++ DFS & MapReduce**
What did Hadoop Do Right?

• **Focus on a few customers**
  – Helped Yahoo Search analytics team
  – Terasort benchmarks

• **Expected Failures**
  – Storage corrects automatically
    – Healthy in minutes instead of hours
  – Nodes are automatically assigned

• **No chokepoints**
  – Data never travels through singleton

• **RAM isn’t large enough**
What did Hadoop Do Right?

• **Simplified FileSystem abstraction**
  – No random writes

• **Apache**
  – Many companies working together
  – Open governance

• **Open Source**
  – Many hands and eyes
  – “Use the source, Luke”

• **Open platform**
Storage

“The more storage you have, the more stuff you accumulate.”

- Stewart
HDFS

• Phases
  – Single HDFS NameNode
  – Cross cluster references
  – Federated HDFS NameNodes

• Need HDFS Block Storage factored out
  – Wider variety of applications

• Need co-location of files
  – Not entire table, but sections of the table
  – ACID (and HBase) base and delta files
  – Correlated tables
File Formats

- **Phases**
  - Text and Sequence File
  - RCFile
  - Avro
  - ORC and Parquet

- **Columnar formats**

- **Type specific encoding**

- **Self describing metadata at end**
ORC

• Light-weight indexes
  – Predicate pushdown
  – Answers from metadata

• Seeking within file

• Available from Hive, Pig, & MapReduce

• C++ reader/writer coming
Computation

“A process cannot be understood by stopping it. Understanding must move with the flow of the process, must join it and flow with it.”

- Herbert
Why does Hadoop Need ACID?

• Hadoop and Hive have always…
  – Worked without ACID
  – Perceived as tradeoff for performance
  – Add or replace entire partitions

• But, your data isn’t static
  – It changes daily, hourly, or faster
  – Managing change makes the user’s life better

• Need consistent views of changing data!
Use Cases

• Updating a Dimension Table
  – Changing a customer’s address

• Delete Old Records
  – Remove records for compliance

• Update/Restate Large Fact Tables
  – Fix problems after they are in the warehouse

• Streaming Data Ingest
  – A continual stream of data coming in
Longer Term Use Cases

- **Multiple statement transactions**
  - Group statements that need to work together

- **Query tables as they appeared in past**
  - Configurable length of history

- **Row-level lineage**
  - Track users and queries that updated each row
Design

• HDFS Does Not Allow Arbitrary Writes
  – Store changes as delta files
  – Stitched together by client on read

• Writes get a Transaction ID
  – Sequentially assigned by Metastore

• Reads get Committed Transactions
  – Provides snapshot consistency
  – No locks required
  – Provide a snapshot of data from start of query
Vectorization

• **MapReduce’s RecordReader**
  – boolean next(K key, V value);

• **Better to process 1000 rows at a time**
  – Amortizes the cost of method calls
  – Use primitive arrays for tight inner loops
    – No access methods
  – Extremely important for operator trees
    – Branches (including virtual dispatch) kill pipelining

• **Can run at 100m rows/second**
Tez

- Replacing MapReduce as basis for
  - Hive, Pig, Cascading
- Executes entire DAG of tasks
- More options for shuffle
- Scales up and down dynamically
- Queries scheduled as one application instead of waves of jobs.
Hive Cost Based Optimizer

• Current optimizer is a mess of rules
  – Rule interactions are complex

• Optiq provides a framework
  – YACC for optimizers

• Make better choices
  – Huge impact on performance

• Obsoletes lots of old advice
LLAP

• **Live Long and Process**  
  – Persistent Hive execution engine

• **JVM startup costs are huge**  
  – JIT cost alone is staggering

• **Hot Table Data Caching**  
  – Keep hot columns and partitions in memory

• **Sub-second answers**
Security

“There is no such thing as perfect security, only varying levels of insecurity.”

- Rushdie
Audit and Authorization

- Three A’s of security
  - Authentication, Authorization, and Audit

- Phases
  - No users
  - Users, but no authentication
  - Authorization

- Next centralized authorization and audit

- Encryption
Encryption

• Underlying file system
  – Thief breaks into data center…

• HDFS encryption
  – Parallels HDFS authorization
  – Prevents AFN attacks

• Column encryption
  – Encrypt just PII columns, rolling keys

• Value encryption
  – No salt ➔ weak sauce so joins work
Thank You!

Questions & Answers