



Apache HBase in the Real World

Joey Echeverria
@fwiffo

#bigdatadc

The (Shameless) Plug

We're Training!

Developer Training August 22 to 24
Admin Training August 25 to 26

<http://www.cloudera.com/training>

We're Hiring!

Solution Architects, Trainers,
Distributed Systems Engineers

<http://www.cloudera.com/careers>

1 Minute Hadoop Recap

HDFS

- Distributed file system
- Optimized for streaming reads and writes
- Block-level replication

MapReduce

- Distributed processing framework
- Reads/writes data in HDFS (typically)
- Operates over (key, value) view of data

Where does HBase come in?

Google

Google invented GFS and MapReduce
GFS optimized for streaming reads and writes

BigTable

Google's answer to random read/write workloads

HBase: BigTable-like storage (for Hadoop)



What is HBase?

Key/value column **family** store

Data stored in HDFS

ZooKeeper for coordination

Access model is get/put/del

Plus range scans and versions

Tables and Column Families

Static part of the schema

Column families also form locality groups

- One Store per family
- Multiple HFiles per Store

Tables split into regions

- Continuous range of row keys
- Unit of distribution
- Automatically split
- Pre-split for performance

Why use HBase?

Variable schema in each record

Collections of data for each key

Atomic control of per-key data

Row access to each column family

HBase Applications



“Smart Data, at Scale, made Easy”
<http://www.lilyproject.org>

OpenTSDB

“Distributed, scalable Time Series Database (TSDB)”
<http://opentsdb.net>

Real-time ad optimizations

Capturing impressions and serving ads

HBase front-end – to serve models (via memcached)

HBase back-end – to serve pixels and capture cookies

MapReduce to compute models between the two

Click stream sessionization

Key on userid and time

Seperate table for significant events (e.g. purchase)

Load data using HBase importtsv tool

Sessionization performed by simple scans

Mozilla - Socorro

When Firefox crashes, where do reports go?

The Mozilla team gathers those crashes in HBase

Crashes vary widely and change format often

Processors take each individually and parse it out

<http://crash-stats.mozilla.com>

<http://code.google.com/p/socorro>

Navteq

Location based content serving

All served out of HBase, location makes a great key

Content is variable – Maps, POI, User Data

Preprocessing is all done via MR jobs

Cloudera

Gathers data about customer clusters

Each customer node is a key with Avro values

Easy to browse, quick to find issues on Nodes

Dump to HDFS and process with Pig

cloudera