Realtime Apache Hadoop at Facebook

Jonathan Gray & Dhruba Borthakur
June 14, 2011 at SIGMOD, Athens
# Agenda

1. Why Apache Hadoop and HBase?
2. Quick Introduction to Apache HBase
3. Applications of HBase at Facebook
Why Hadoop and HBase?
For *Realtime* Data?
Problems with existing stack

- MySQL is stable, but...
  - Not inherently distributed
  - Table size limits
  - Inflexible schema

- Hadoop is scalable, but...
  - MapReduce is slow and difficult
  - Does not support random writes
  - Poor support for random reads
Specialized solutions

- High-throughput, persistent key-value
  - Tokyo Cabinet
- Large scale data warehousing
  - Hive/Hadoop
- Photo Store
  - Haystack
- Custom C++ servers for lots of other stuff
What do we need in a data store?

- Requirements for Facebook Messages
  - Massive datasets, with large subsets of cold data
  - Elasticity and high availability
  - Strong consistency within a datacenter
  - Fault isolation

- Some non-requirements
  - Network partitions within a single datacenter
  - Active-active serving from multiple datacenters
HBase satisfied our requirements

- In early 2010, engineers at FB compared DBs
  - Apache Cassandra, Apache HBase, Sharded MySQL

- Compared performance, scalability, and features
  - HBase gave excellent write performance, good reads
  - HBase already included many nice-to-have features
    - Atomic read-modify-write operations
    - Multiple shards per server
    - Bulk importing
    - MapReduce
HBase uses HDFS

We get the benefits of HDFS as a storage system for free

- Fault tolerance
- Scalability
- Checksums fix corruptions
- MapReduce

- Fault isolation of disks
- HDFS battle tested at petabyte scale at Facebook

Lots of existing operational experience
Apache HBase

- Originally part of Hadoop
  - HBase adds random read/write access to HDFS

- Required some Hadoop changes for FB usage
  - File appends
  - HA NameNode
  - Read optimizations

- Plus ZooKeeper!
HBase System Overview

Database Layer

- Master
- Backup Master
- Region Server
- Region Server
- Region Server

Storage Layer

- Namenode
- Secondary Namenode
- Datanode
- Datanode
- Datanode

HDFS

Coordination Service

- ZK Peer
- ZK Peer

Zookeeper Quorum
HBase in a nutshell

- Sorted and column-oriented
- High write throughput
- Horizontal scalability
- Automatic failover
- Regions sharded dynamically
Applications of HBase at Facebook
Use Case 1

Titan

(Facebook Messages)
The New Facebook Messages

- Messages
- IM/Chat
- email
- SMS
Facebook Messaging

- **High write throughput**: Every message, instant message, SMS, and e-mail

- Search indexes for all of the above

- **Denormalized schema**

- **A product at massive scale on day one**
  - 6k messages a second
  - 50k instant messages a second
  - 300TB data growth/month compressed
Typical Cell Layout

- Multiple cells for messaging
- 20 servers/rack; 5 or more racks per cluster
- Controllers (master/Zookeeper) spread across racks

<table>
<thead>
<tr>
<th>Rack #1</th>
<th>Rack #2</th>
<th>Rack #3</th>
<th>Rack #4</th>
<th>Rack #5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZooKeeper HDFS NameNode</td>
<td>ZooKeeper Backup NameNode</td>
<td>ZooKeeper Job Tracker</td>
<td>ZooKeeper HBase Master</td>
<td>ZooKeeper Backup Master</td>
</tr>
<tr>
<td>Region Server</td>
<td>Region Server</td>
<td>Region Server</td>
<td>Region Server</td>
<td>Region Server</td>
</tr>
<tr>
<td>Data Node</td>
<td>Data Node</td>
<td>Data Node</td>
<td>Data Node</td>
<td>Data Node</td>
</tr>
<tr>
<td>Task Tracker</td>
<td>Task Tracker</td>
<td>Task Tracker</td>
<td>Task Tracker</td>
<td>Task Tracker</td>
</tr>
</tbody>
</table>
Horizontal Scalability

- Region

- ... (dotted line indicating continuation)

- Two blocks of rectangles connected by lines, indicating scalability.
Automatic Failover

- HBase client
- Find new server from META
- server died

No physical data copy because data is in HDFS
Use Case 2

Puma

(Facebook Insights)
Puma

- **Realtime Data Pipeline**
  - Utilize existing log aggregation pipeline (Scribe-HDFS)
  - Extend low-latency capabilities of HDFS (Sync+PTail)
  - High-throughput writes (HBase)

- **Support for Realtime Aggregation**
  - Utilize HBase atomic increments to maintain roll-ups
  - Complex HBase schemas for unique-user calculations
  - Store checkpoint information directly in HBase
Puma as Realtime MapReduce

- Map phase with PTail
  - Divide the input log stream into N shards
  - First version only supported random bucketing
  - Now supports application-level bucketing

- Reduce phase with HBase
  - Every row+column in HBase is an output key
  - Aggregate key counts using atomic counters
  - Can also maintain per-key lists or other structures
Realtime URL/Domain Insights
- Domain owners can see deep analytics for their site
- Clicks, Likes, Shares, Comments, Impressions
- Detailed demographic breakdowns (anonymized)
- Top URLs calculated per-domain and globally

Massive Throughput
- Billions of URLs
- > 1 Million counter increments per second
<table>
<thead>
<tr>
<th>Age</th>
<th>Female (31%)</th>
<th>Male (62%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>13–17</td>
<td>7.9%</td>
<td>13%</td>
</tr>
<tr>
<td>18–24</td>
<td>8.5%</td>
<td>22%</td>
</tr>
<tr>
<td>25–34</td>
<td>6.6%</td>
<td>16%</td>
</tr>
<tr>
<td>35–44</td>
<td>3.9%</td>
<td>6.3%</td>
</tr>
<tr>
<td>45–54</td>
<td>2.3%</td>
<td>2.8%</td>
</tr>
<tr>
<td>55+</td>
<td>1.5%</td>
<td>2.3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Language</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>English (US)</td>
<td>(34%)</td>
</tr>
<tr>
<td>Spanish</td>
<td>(12%)</td>
</tr>
<tr>
<td>Arabic</td>
<td>(6.1%)</td>
</tr>
<tr>
<td>Turkish</td>
<td>(6%)</td>
</tr>
<tr>
<td>English (UK)</td>
<td>(5.3%)</td>
</tr>
<tr>
<td>French (France)</td>
<td>(5.2%)</td>
</tr>
<tr>
<td>Spanish (Spain)</td>
<td>(4.6%)</td>
</tr>
<tr>
<td>Indonesian</td>
<td>(3.9%)</td>
</tr>
<tr>
<td>Italian</td>
<td>(3.7%)</td>
</tr>
<tr>
<td>German</td>
<td>(2.7%)</td>
</tr>
<tr>
<td>Portuguese (Brazil)</td>
<td>(2.6%)</td>
</tr>
<tr>
<td>Thai</td>
<td>(1.2%)</td>
</tr>
<tr>
<td>Russian</td>
<td>(0.96%)</td>
</tr>
<tr>
<td>Polish</td>
<td>(0.91%)</td>
</tr>
<tr>
<td>More</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Country</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>(13%)</td>
</tr>
<tr>
<td>Turkey</td>
<td>(5.9%)</td>
</tr>
<tr>
<td>Philippines</td>
<td>(5%)</td>
</tr>
<tr>
<td>Indonesia</td>
<td>(4.5%)</td>
</tr>
<tr>
<td>India</td>
<td>(4.4%)</td>
</tr>
<tr>
<td>Mexico</td>
<td>(4.3%)</td>
</tr>
<tr>
<td>Egypt</td>
<td>(4.1%)</td>
</tr>
<tr>
<td>Italy</td>
<td>(3.8%)</td>
</tr>
<tr>
<td>Brazil</td>
<td>(2.7%)</td>
</tr>
<tr>
<td>Argentina</td>
<td>(2.7%)</td>
</tr>
<tr>
<td>Germany</td>
<td>(2.6%)</td>
</tr>
<tr>
<td>Malaysia</td>
<td>(2.6%)</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>(2.3%)</td>
</tr>
<tr>
<td>Spain</td>
<td>(1.8%)</td>
</tr>
</tbody>
</table>
Use Case 3

ODS

(Facebook Internal Metrics)
ODS

- **Operational Data Store**
  - System metrics (CPU, Memory, IO, Network)
  - Application metrics (Web, DB, Caches)
  - Facebook metrics (Usage, Revenue)
    - Easily graph this data over time
    - Supports complex aggregation, transformations, etc.

- **Difficult to scale with MySQL**
  - Millions of unique time-series with billions of points
  - Irregular data growth patterns
Dynamic sharding of regions

Region

server overloaded
Future of HBase at Facebook
User and Graph Data in HBase
HBase for the important stuff

- Looking at HBase to augment MySQL
  - Only single row ACID from MySQL is used
  - DBs are always fronted by an in-memory cache
  - HBase is great at storing dictionaries and lists

- Database tier size determined by IOPS
  - HBase does only sequential writes
  - Lower IOPs translate to lower cost
  - Larger tables on denser, cheaper, commodity nodes
Conclusion

- Facebook investing in Realtime Hadoop/HBase
  - Work of a large team of Facebook engineers
  - Close collaboration with open source developers

- Much more detail in Realtime Hadoop paper
  - Technical details about changes to Hadoop and HBase
  - Operational experiences in production
Questions?

jgray@fb.com

dhruba@fb.com