# Hadoop and its Usage at Facebook

Dhruba Borthakur
dhruba@apache.org,
June 22rd, 2009

# Who Am I?

- **Hadoop Developer**
  - Core contributor since Hadoop's infancy
  - Focussed on Hadoop Distributed File System
- **Facebook** (Hadoop)
- **Yahoo!** (Hadoop)
- **Veritas** (San Point Direct, VxFS)
- **IBM Transarc** (Andrew File System)

# Hadoop, Why?

- **Need to process huge datasets on large clusters of computers**
- **Very expensive to build reliability into each application.**
- **Nodes fail every day**
  - Failure is expected, rather than exceptional.
  - The number of nodes in a cluster is not constant.
- **Need common infrastructure**
  - Efficient, reliable, easy to use
  - Open Source, Apache License

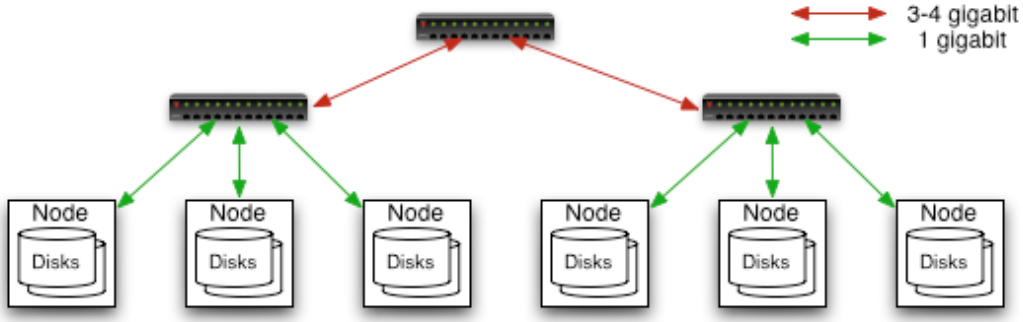
# Hadoop History

- **Dec 2004 –** Google GFS paper published
- **July 2005 –** Nutch uses MapReduce
- **Feb 2006 –** Becomes Lucene subproject
- **Apr 2007 –** Yahoo! on 1000-node cluster
- **Jan 2008 –** An Apache Top Level Project
- **Feb 2008 –** Yahoo! production search index
- **Nov 2008 –** SQL query language called Hive

# Who uses Hadoop?

- Amazon/A9
- Facebook
- Google
- IBM : Blue Cloud?
- Joost
- Last.fm
- New York Times
- PowerSet
- Veoh
- Yahoo!

# Commodity Hardware




**Typically in 2 level architecture**

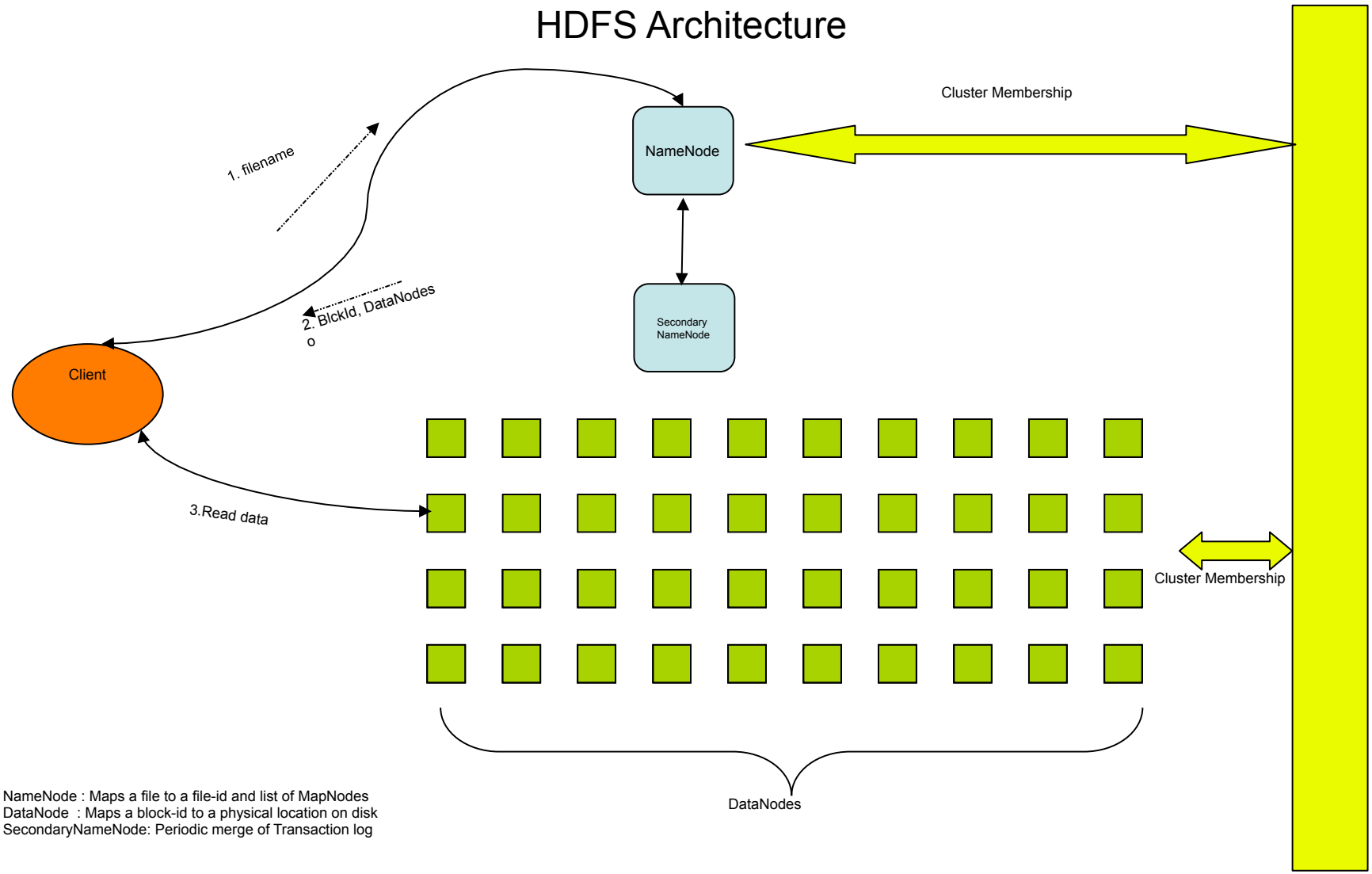
– Nodes are commodity PCs
– 30-40 nodes/rack
– Uplink from rack is 3-4 gigabit
– Rack-internal is 1 gigabit

hadoop

# Goals of HDFS

- **Very Large Distributed File System**
  – 10K nodes, 100 million files, 10 PB
- **Assumes Commodity Hardware**
  – Files are replicated to handle hardware failure
  – Detect failures and recovers from them
- **Optimized for Batch Processing**
  – Data locations exposed so that computations can move to where data resides
  – Provides very high aggregate bandwidth

# HDFS Architecture




NameNode : Maps a file to a file-id and list of MapNodes
DataNode : Maps a block-id to a physical location on disk
SecondaryNameNode: Periodic merge of Transaction log

# Distributed File System

- **Single Namespace for entire cluster**
- **Data Coherency**
  - Write-once-read-many access model
  - Client can only append to existing files
- **Files are broken up into blocks**
  - Typically 128 MB block size
  - Each block replicated on multiple DataNodes
- **Intelligent Client**
  - Client can find location of blocks
  - Client accesses data directly from DataNode

# Hadoop Map/Reduce

- **The Map-Reduce programming model**
  – Framework for distributed processing of large data sets
  – Pluggable user code runs in generic framework
- **Common design pattern in data processing**
  cat * | grep | sort | unique -c | cat > file
  input | **map** | shuffle | **reduce** | output
- **Natural for:**
  – Log processing
  – Web search indexing
  – Ad-hoc queries

# Hadoop/Hive at Facebook

- Cross functional team of 11 members
  - 5 people working in Hive development
  - 2 people on Hadoop development
  - 2 people on Data Pipelines and Oracle Data Mart
  - 1 Production Operations

# Why Hive?

- Large installed base of SQL users
- Analytics SQL queries translate well to map-reduce
- Files are insufficient data management abstractions
  - Need Tables, schemas, partitions, indices
- Scalability of Hadoop

# Why Hive?

```
hive> select key, count(1) from kv1 where key > 100
    group by key;
```

vs

```
$ cat > /tmp/reducer.sh
uniq –c | awk ‘{print $2”\t”$1}`
$ cat > /tmp/map.sh
awk –F ‘\001’ ‘{if($1 > 100) print $1}`
$ bin/hadoop jar contrib/hadoop-0.19.2-dev-streaming.jar –input /
    user/hive/warehouse/kv1 –mapper map.sh –file /tmp/reducer.sh
    –file /tm;map.sh –reducer reducer.sh –output /tmp/largekey –
    numReducerTasks 1
```

# Hive Query Language

- Basic SQL
  - From clause subquery
  - Join
  - Multi table insert
  - Multi group-by
  - Sampling
- Extensibility
  - Pluggable map-reduce scripts

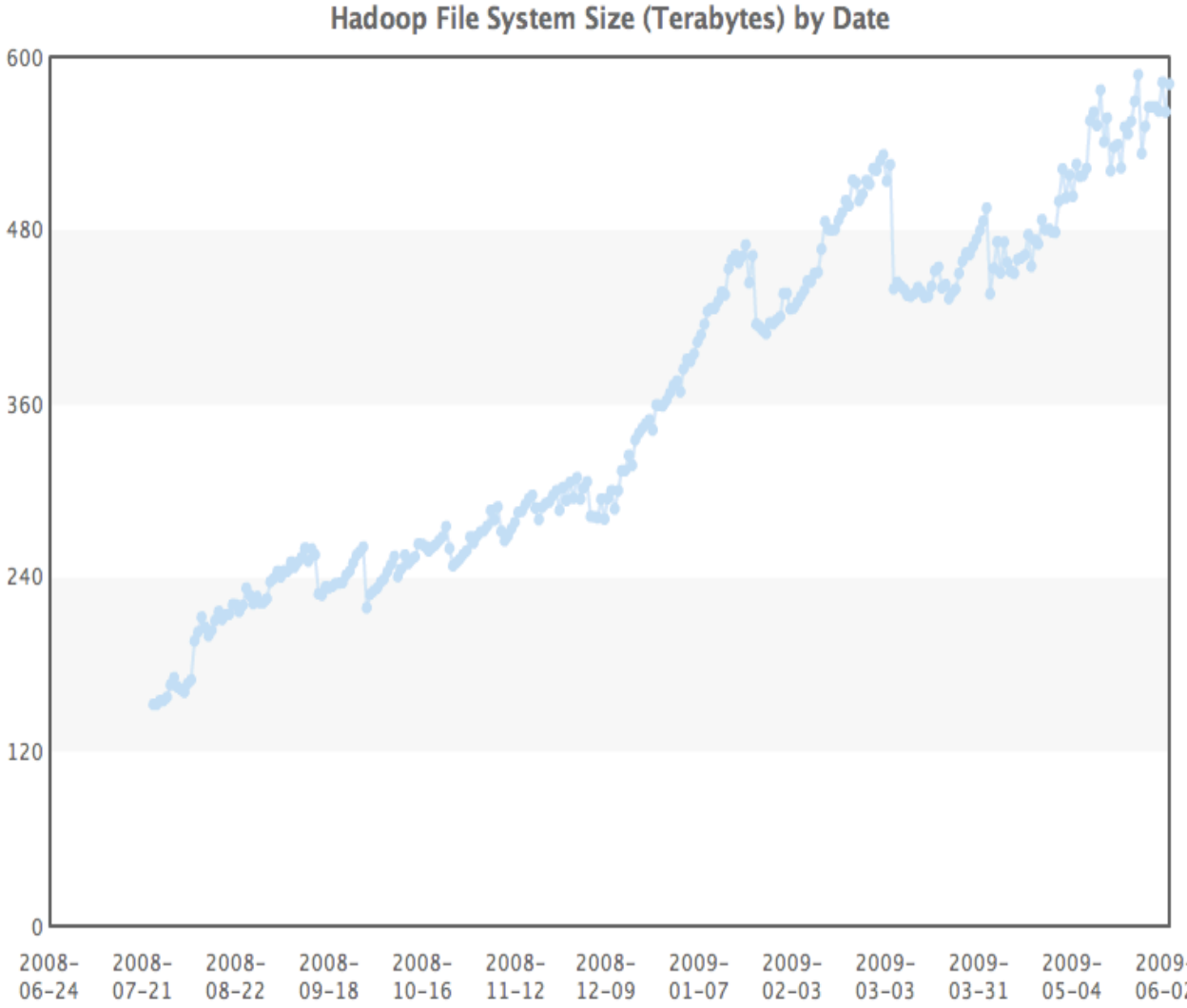
# Who generates this data?

- Lots of data is generated on Facebook
  - 200 million active users
  - 20 million users update their statuses at least once each day
  - More than 850 million photos uploaded to the site each month
  - More than 8 million videos uploaded each month
  - More than 1 billion pieces of content (web links, news stories, blog posts, notes, photos, etc.) shared each week

# Where do we store this data?

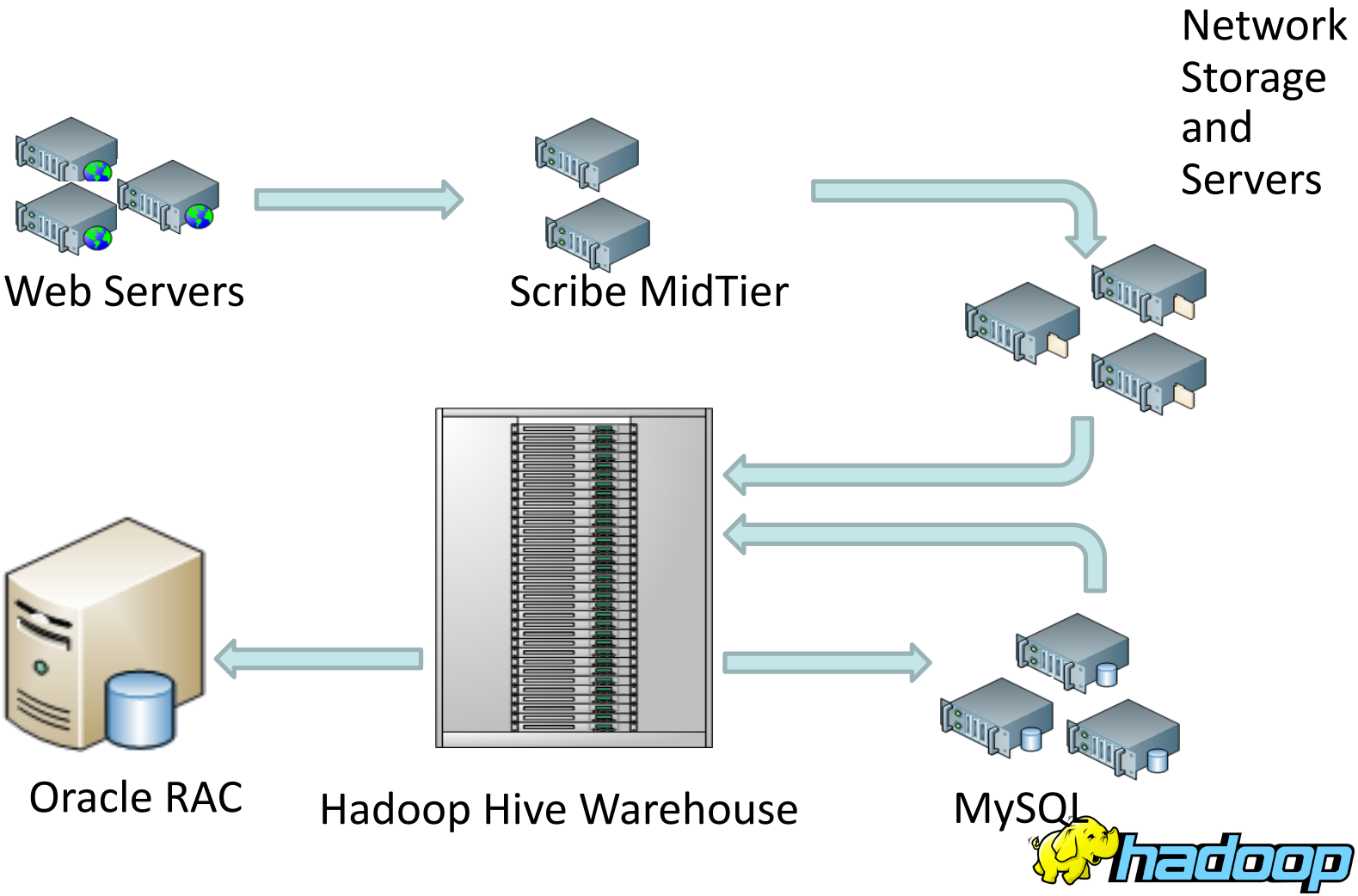
- Hadoop/Hive Warehouse
  - 4800 cores, 2 PetaBytes total size
- Hadoop Archival Store
  - 200 TB

# Rate of Data Growth




hadoop

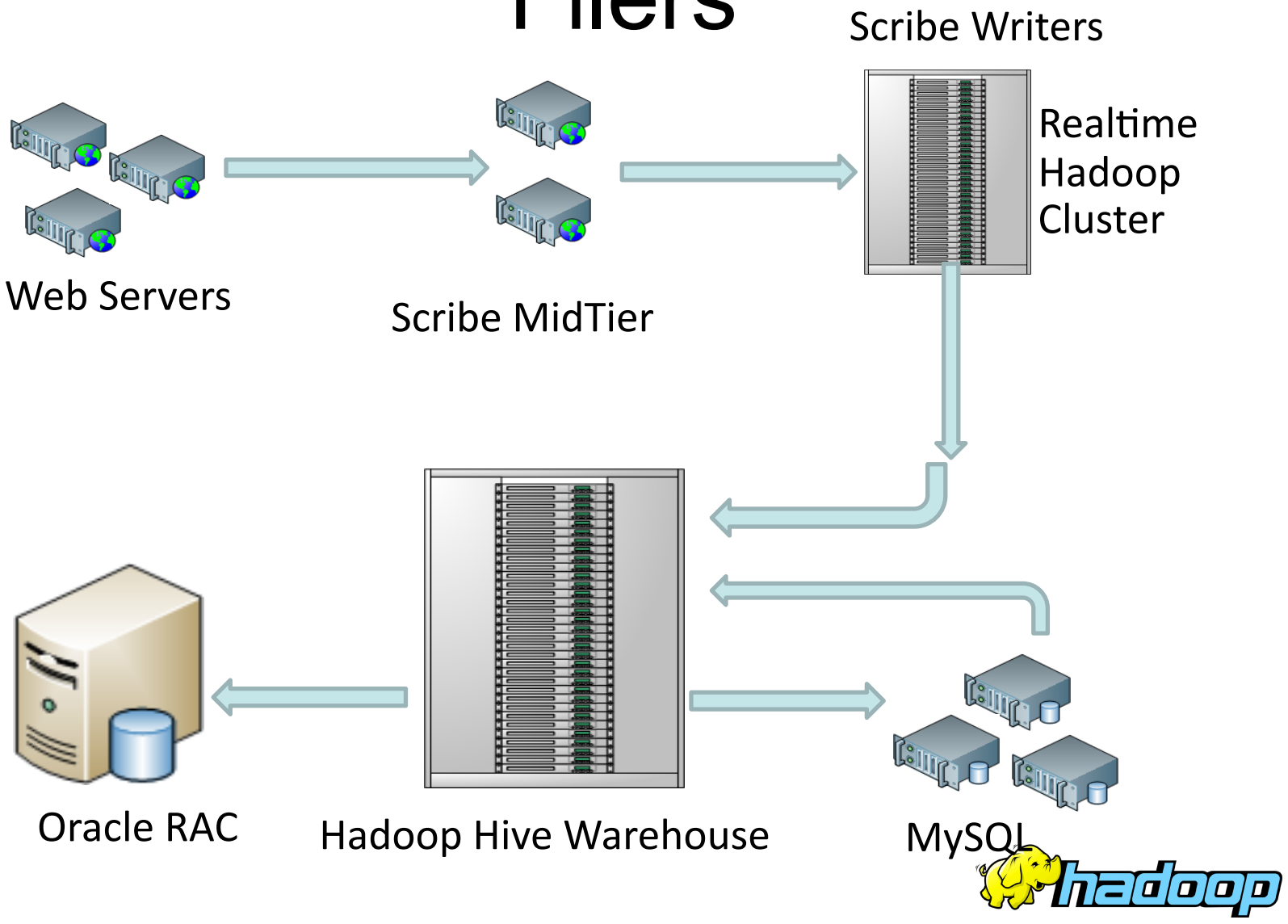
# Data Flow


Network Storage and Servers
Web Servers
Scribe MidTier
Oracle RAC
Hadoop Hive Warehouse
MySQL
hadoop

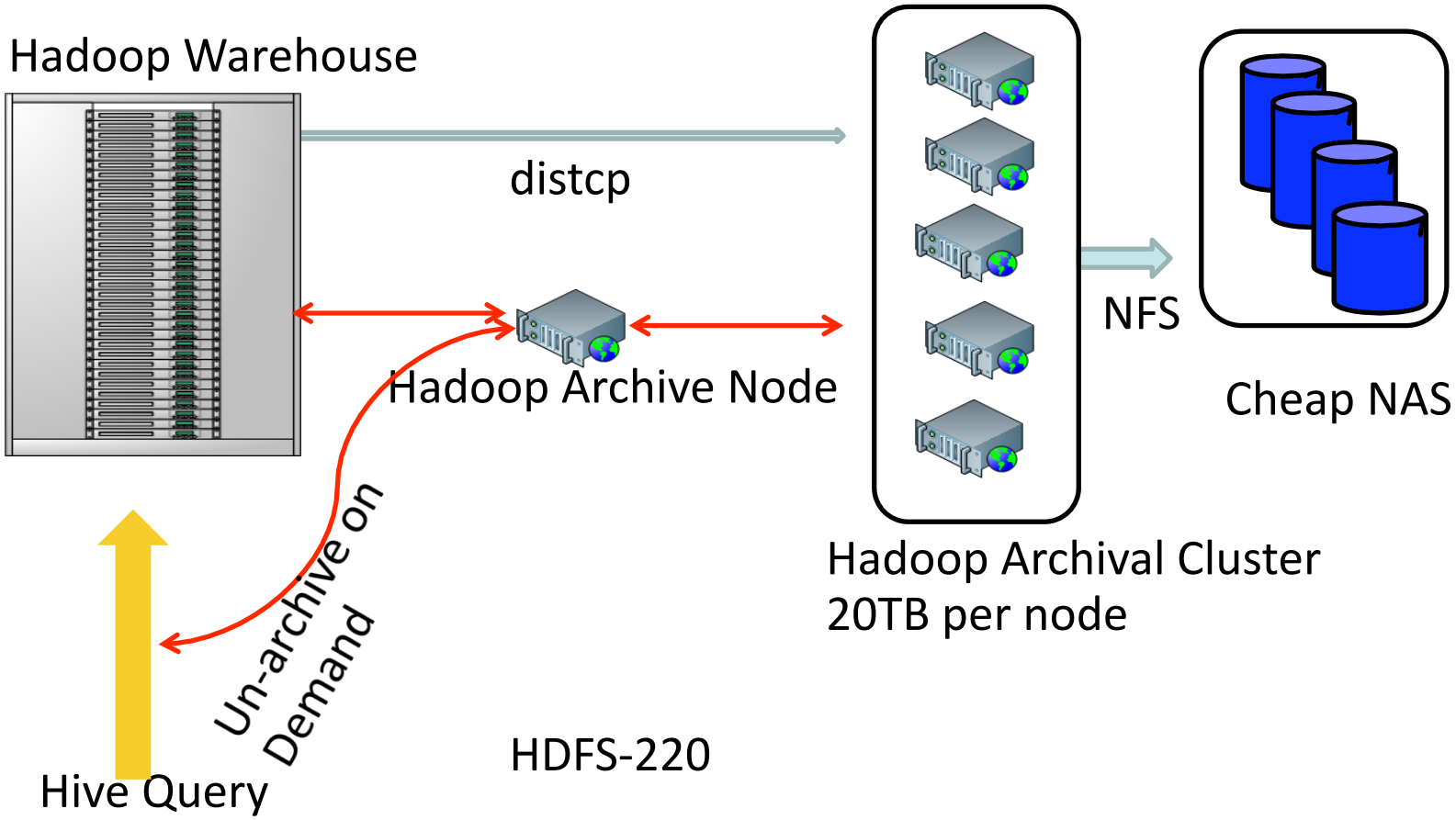
# Data Usage

- Statistics per day:
  - 4 TB of compressed new data added per day
  - 55TB of compressed data scanned per day
  - 3200+ Hive jobs on production cluster per day
  - 80M compute minutes per day
- Barrier to entry is significantly reduced:
  - New engineers go though a Hive training session
  - 140+ people run jobs on Hadoop/Hive jobs
  - Analysts (non-engineers) use Hadoop through Hive

# Hadoop Scribe: Avoid Costly Filers



# Archival: Move old data to cheap storage

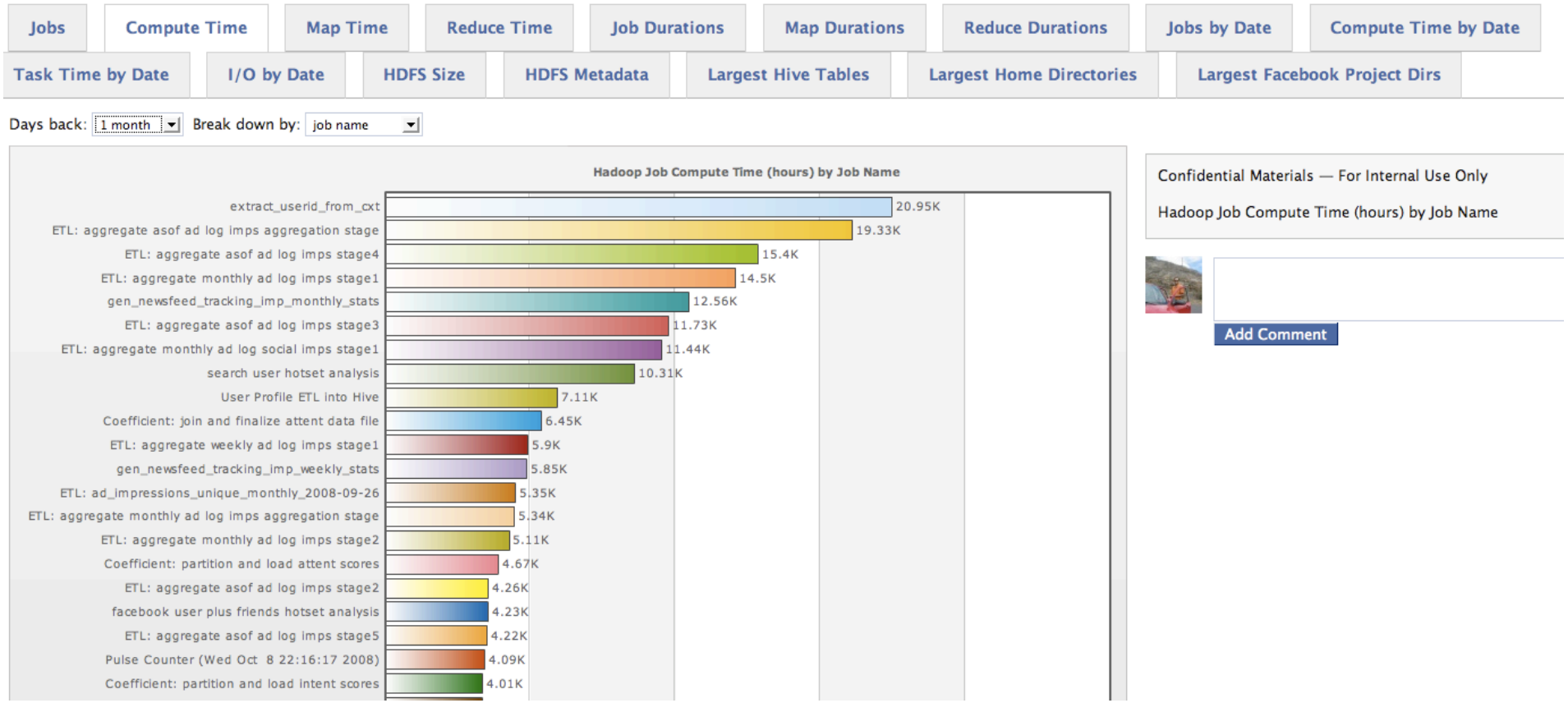



hadoop

# Cluster Usage Dashboard

- History logs are fed into a mySQL database
- A Dashboard displays cluster usage statistics from the database
- Displays cluster utilization, growth rates of cluster usage, etc
- HADOOP-3708

# Cluster Usage Dashboard




# Hive WebUI

## HiPal: an Online Tool for Querying Hive/Hadoop Data Warehouse

+ Learn More about HiPal + Why am I on dev127?

### Query

**Table**: u_full

**Start Partition** | **End Partition**

**Data Size (bytes): Cat/Export Data**: 734,184,513,313 Get 10 rows | Export the whole u_full

**Select Columns** [All] [Clear]

☑ userid ☐ base ☐ affiliations ☐ last ☑ friends ☐ ext ☐ groups ☐ fbpages ☐ whs ☐ events ☐ photo_tags ☑ schools ☐ applications ☐ regionid

+ Join Options

+ Group By Options

+ Where Options

+ Query Options

```
CREATE TABLE tmp_hipal_<QUERYID> (userid STRING, friends STRING, schools STRING);
FROM u_full TABLESAMPLE (BUCKET 1 OUT OF 1024) a
INSERT OVERWRITE TABLE tmp_hipal_<QUERYID>
SELECT a.userid, a.friends, a.schools
```

Submit Query

[Join HiPal User Mailing List][Report problems or ask questions]

### Job Status

**Show all jobs**: ☐ enable | **Sort By**: Submit Time | **In**: Descending Order

| QueryId | Submit Time | User | Query (Last Update: 2008-10-27 12:42:58 AM) | Time (sec.) | Query Progress | Latest Hadoop Job | C |
|---|---|---|---|---|---|---|---|
| 3393 | 2008-10-15 1:48:20 pm | dhruba | CREATE TABLE tmp_hipal_<QUERYID> (user STRING); FROM f_add_video TABLESAMPLE (BUCKET 1 OUT OF 1024) a INSERT OVERWRITE TABLE tmp_hipal_<QUERYID> SELECT a.user WHERE a.ds>='2007-10-27' AND a.ds<='2008-05-28' | 57 | 100% | Status | L |

# Questions?