June 2011

# Production Hadoop systems in the enterprise

cloudera

# Agenda

**1** What Hadoop changes about data

**2** The system – past and present

**3** Living with it – your present and future

**4** Q&A

**cloudera**

# Who We Are

**Mission:** To help organizations to profit from all of their data

## How We Do It

We deliver relevant products and services.

- A distribution of Apache Hadoop that is tested, certified and supported
- Comprehensive support and professional service offerings
- A suite of management software for Hadoop operations
- Training and certification programs for developers, administrators, managers and data scientists

## Credentials

The Apache Hadoop experts.

- Number 1 distribution of Apache Hadoop in the world
- Largest contributor to the open source Hadoop ecosystem
- More committers on staff than any other company
- More than 80 customers across a wide variety of industries
- Strong growth in revenue and new accounts

## Technical Team

Unmatched knowledge and experience.

- Founders, committers and contributors to Hadoop
- A wealth of experience in the design and delivery of production software

## Leadership

Strong executive team with proven abilities.

**Mike Olson**
*CEO*

**Kirk Dunn**
*COO*

**Charles Zedlewski**
*VP, Product*

**Mary Rorabaugh**
*CFO*

**Jeff Hammerbacher**
*Chief Scientist*

**Amr Awadalla**
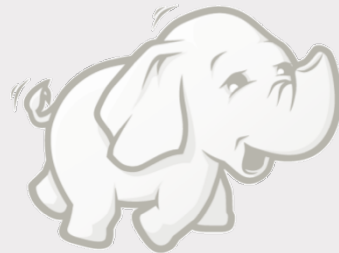*VP Engineering*

**Doug Cutting**
*Chief Architect*

**Omer Trajman**
*VP, Customer Solutions*

cloudera

# What we do: Apache Hadoop

**Hadoop** is a platform for data storage and processing that is…

- ✓ Scalable
- ✓ Fault tolerant
- ✓ Open source

## CORE HADOOP COMPONENTS

**Hadoop Distributed File System (HDFS)**

File Sharing & Data Protection Across Physical Servers

**+**

**MapReduce**

Distributed Computing Across Physical Servers

## Flexibility

- A single repository for storing processing & analyzing <u>any</u> type of data
- Not bound by a single schema

## Scalability

- Scale-out architecture divides workloads across multiple nodes
- Flexible file system eliminates ETL bottlenecks

## Low Cost

- Can be deployed on commodity hardware
- Open source platform guards against vendor lock

cloudera

# Changing the data game

- Late binding
  - Defers cost
  - Accelerates experimentation which accelerates discovery

- JGMD (just get more data)
  - Increases the probability of the novel insight
  - Lowers the data scientist time per insight

- JGMS (just get more servers)
  - Operationally way simpler and cheaper than "scale up" strategies

cloudera

# Applied to the enterprise

## Two **Core Use Cases**
### Applied Across Verticals

| Use Case | Application | Industry | Application | Use Case |
|:---:|:---:|:---:|:---:|:---:|
| **ADVANCED ANALYTICS** | Social Network Analysis | **Web** | Search index build | **DATA PROCESSING** |
| | Content Optimization | **Media** | Clickstream Sessionization | |
| | Network Analytics | **Telco** | Mediation | |
| | Loyalty & Promotions Analysis | **Retail** | Data Factory | |
| | Fraud Analysis | **Financial** | Trade Reconciliation | |
| | Entity Analysis | **Federal** | SIGINT | |
| | Sequencing Analysis | **Bioinformatics** | Genome Mapping | |

cloudera

# My point

- Hadoop introduces a deceptively large shift in how you think about data

- The impact on your end users & IT model will change

**cloudera**

# Agenda

**1**   What Hadoop changes about data

**2**   The system – past and present

**3**   Living with it – your present and future

**4**   Q&A

**cloudera**

# Where did this all start?

# What did they do?

Google

| Dremel | | |
|---|---|---|
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| | Chubby | |

cloudera
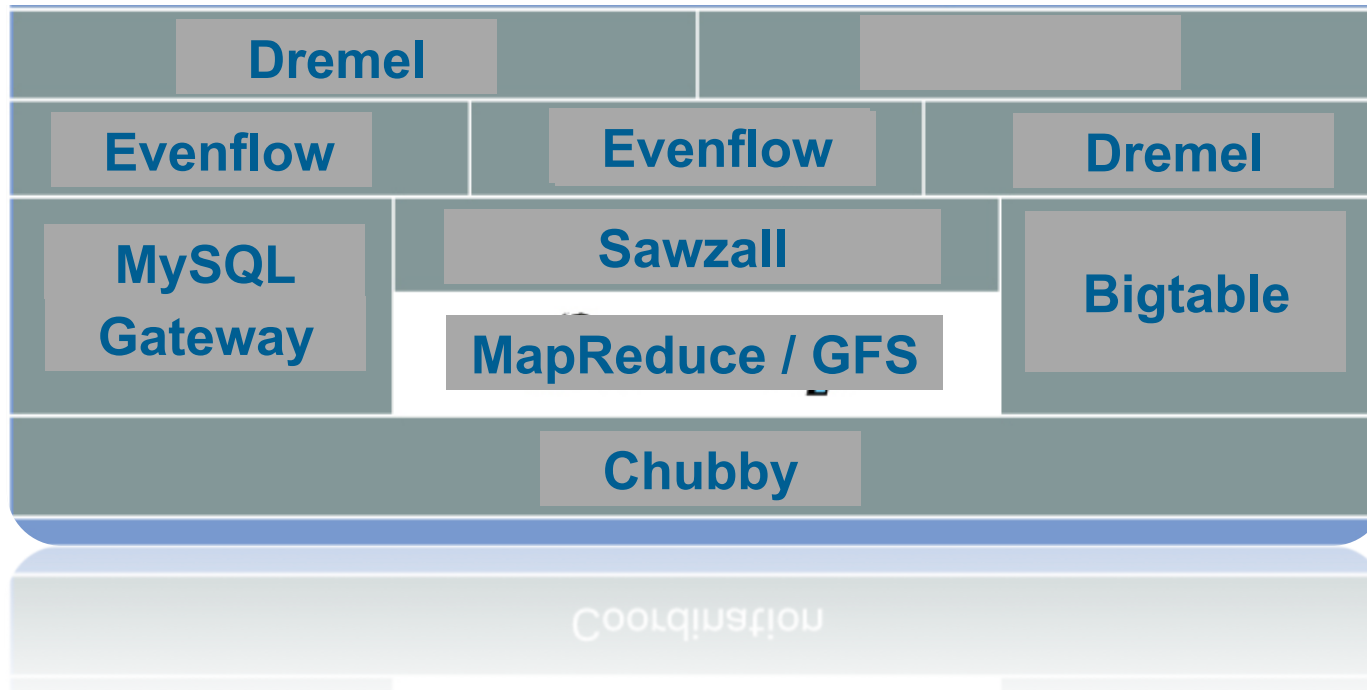
# What did Google do?

Google

**Store data**

| | | |
|---|---|---|
| Dremel | | |
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| | Chubby | |

Coordination

cloudera
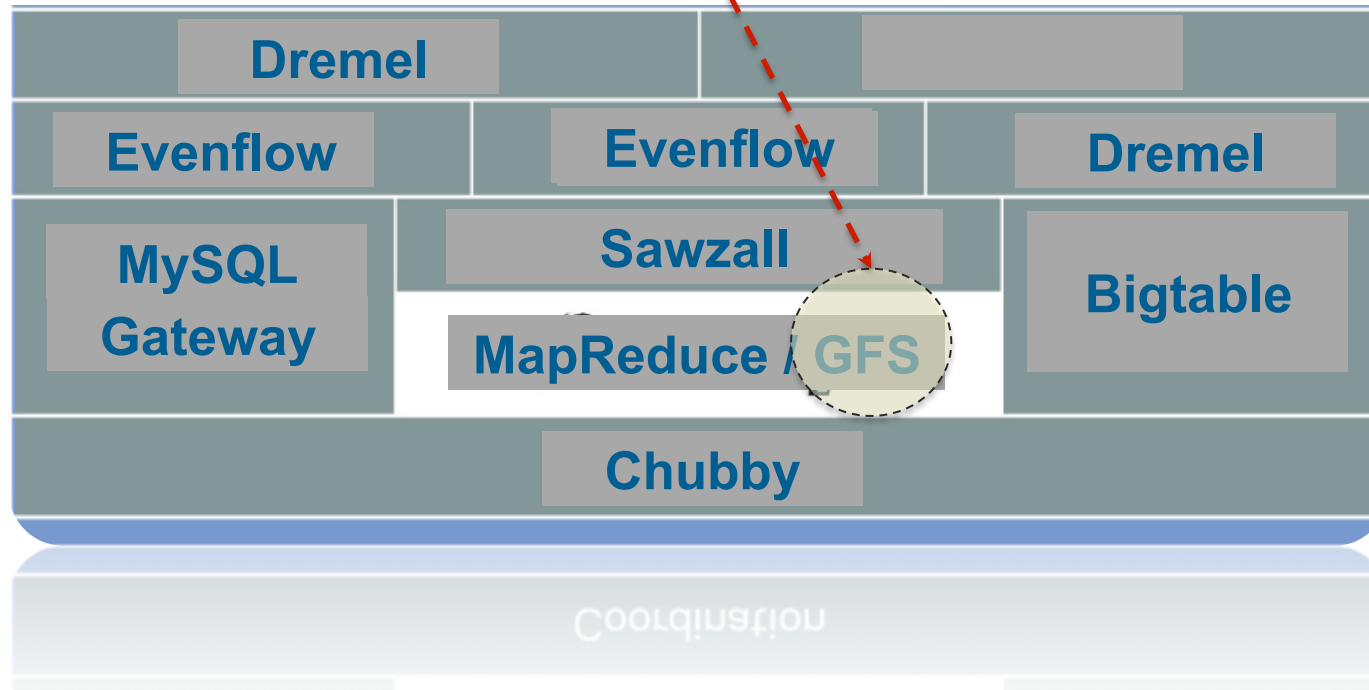
# What did Google do?

Google

Process data
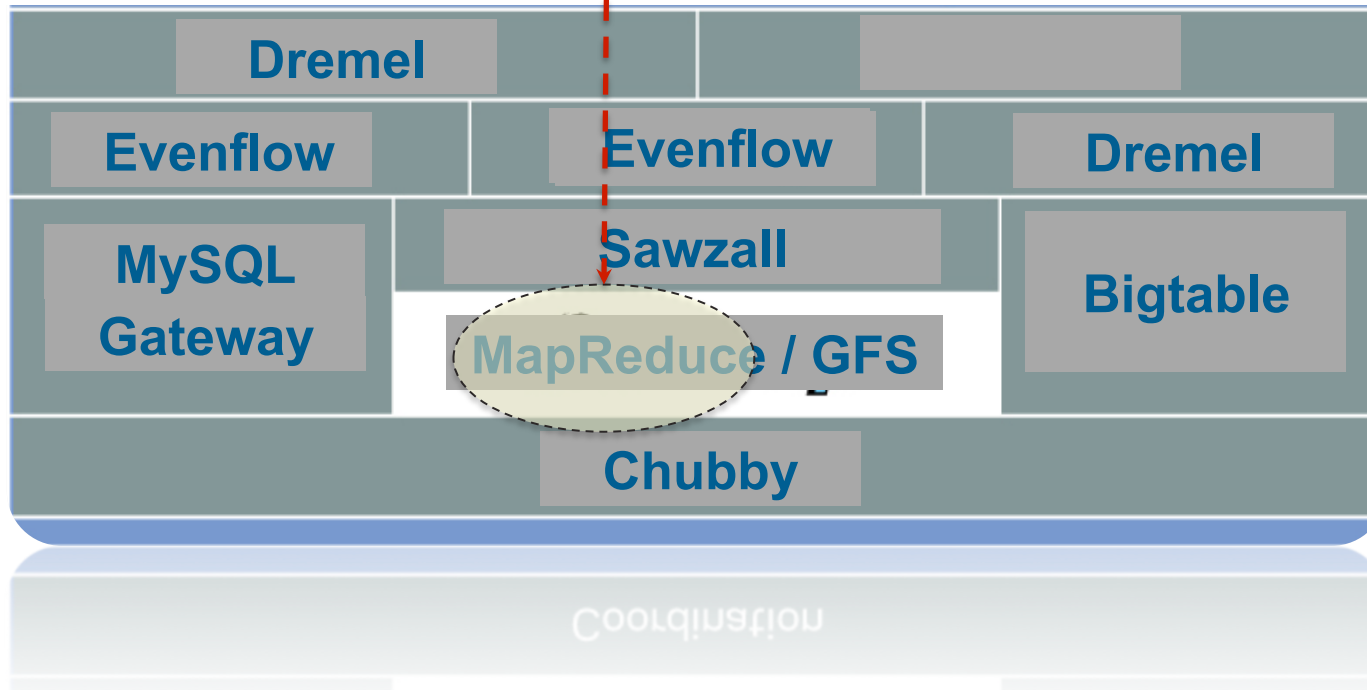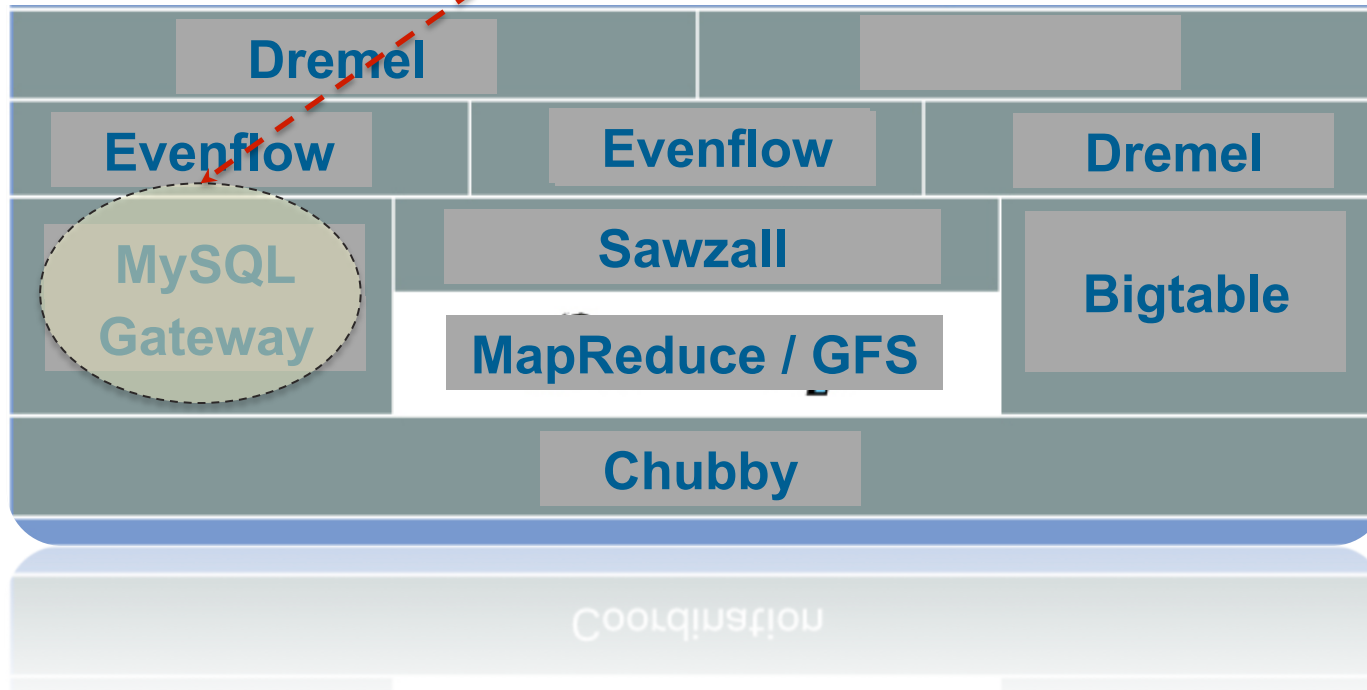
| Dremel | | |
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| | Chubby | |

cloudera

# What did Google do?

**Ingest data**

| Dremel | | |
|---|---|---|
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| | Chubby | |

cloudera

# What did Google do?

**Serve data**

| Dremel | | |
|---|---|---|
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | *Bigtable* |
| | MapReduce / GFS | |
| | Chubby | |

cloudera
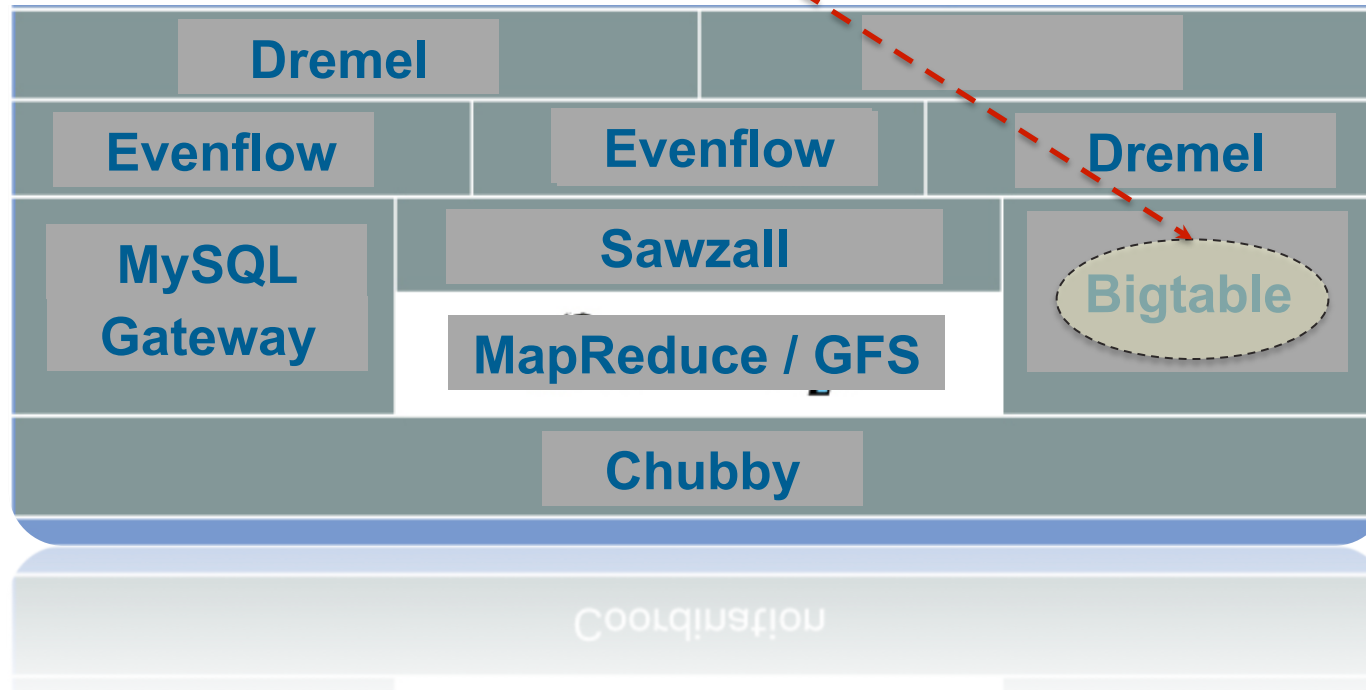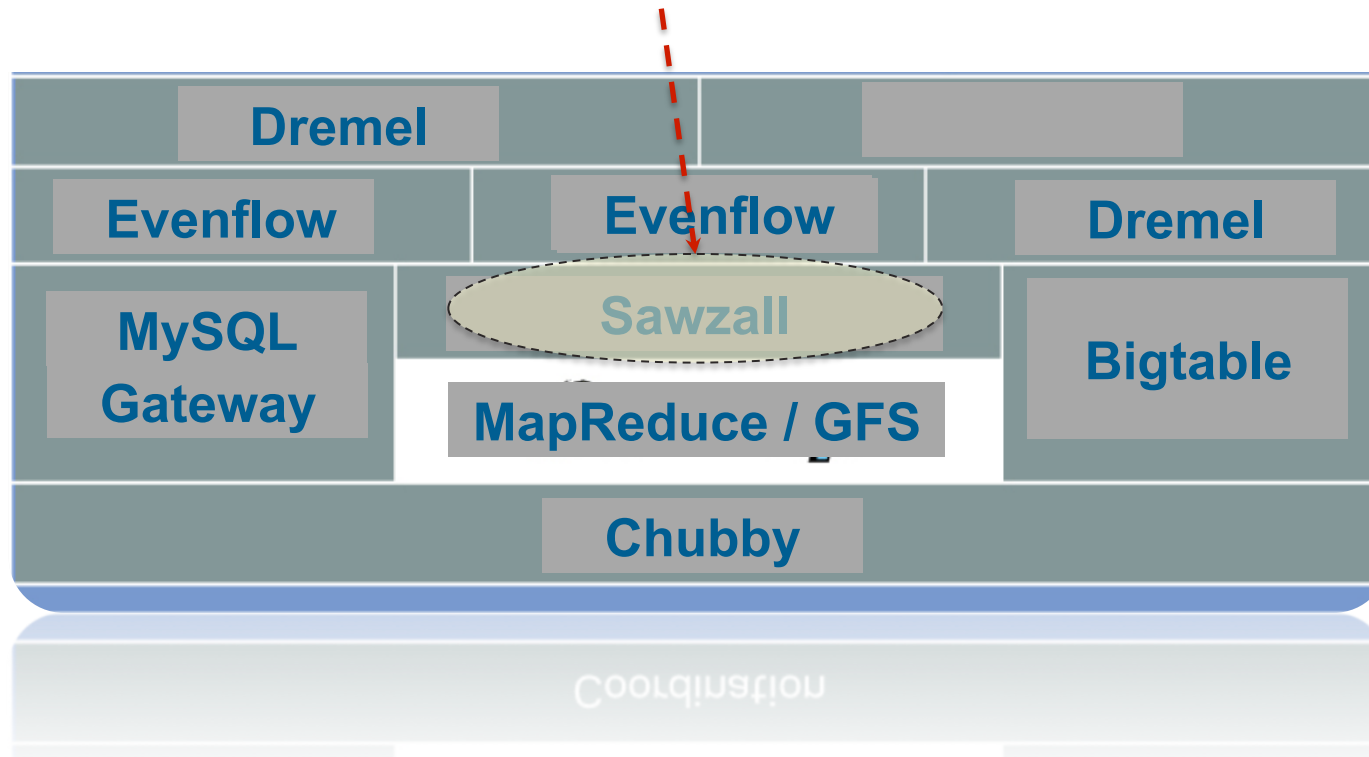
# What did Google do?

# What did Google do?

**Google**

**Chain together complex workloads**

| Dremel | | |
|---|---|---|
| *Evenflow* | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| Chubby | | |

**cloudera**
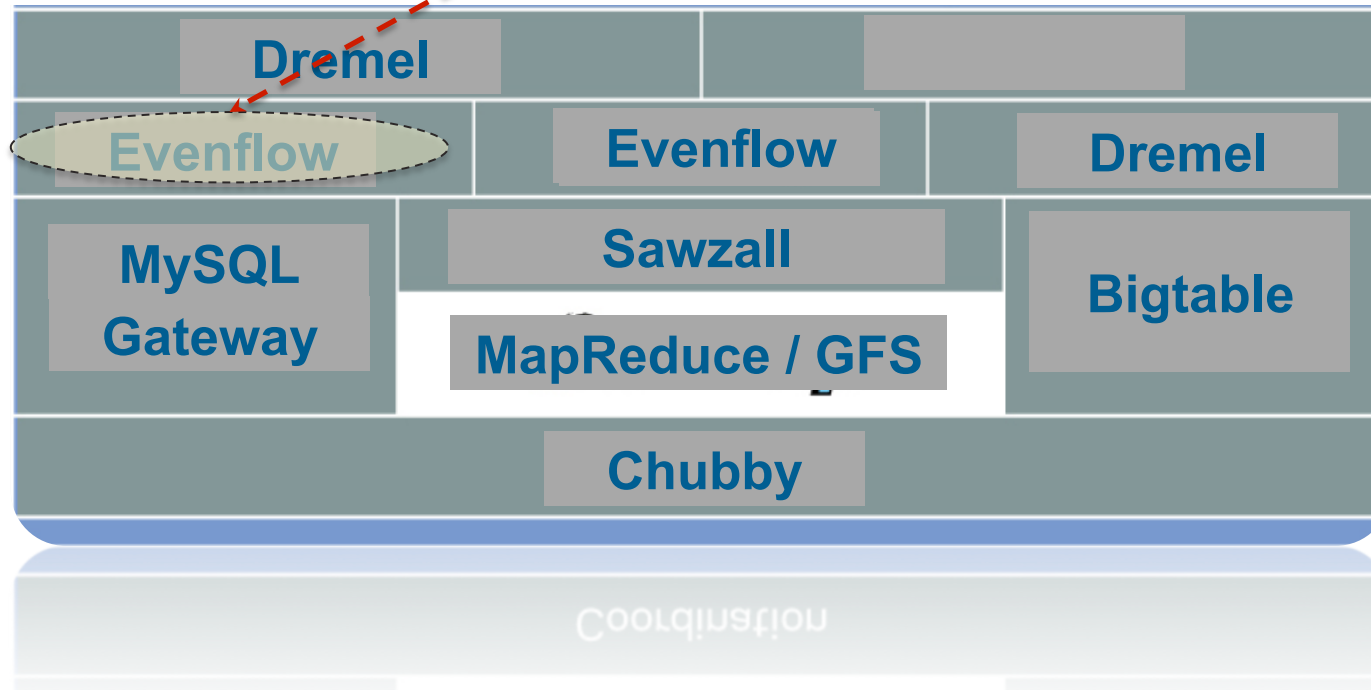
# What did Google do?

Google

**Schedule them**

| Dremel | | |
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| Chubby | | |

cloudera
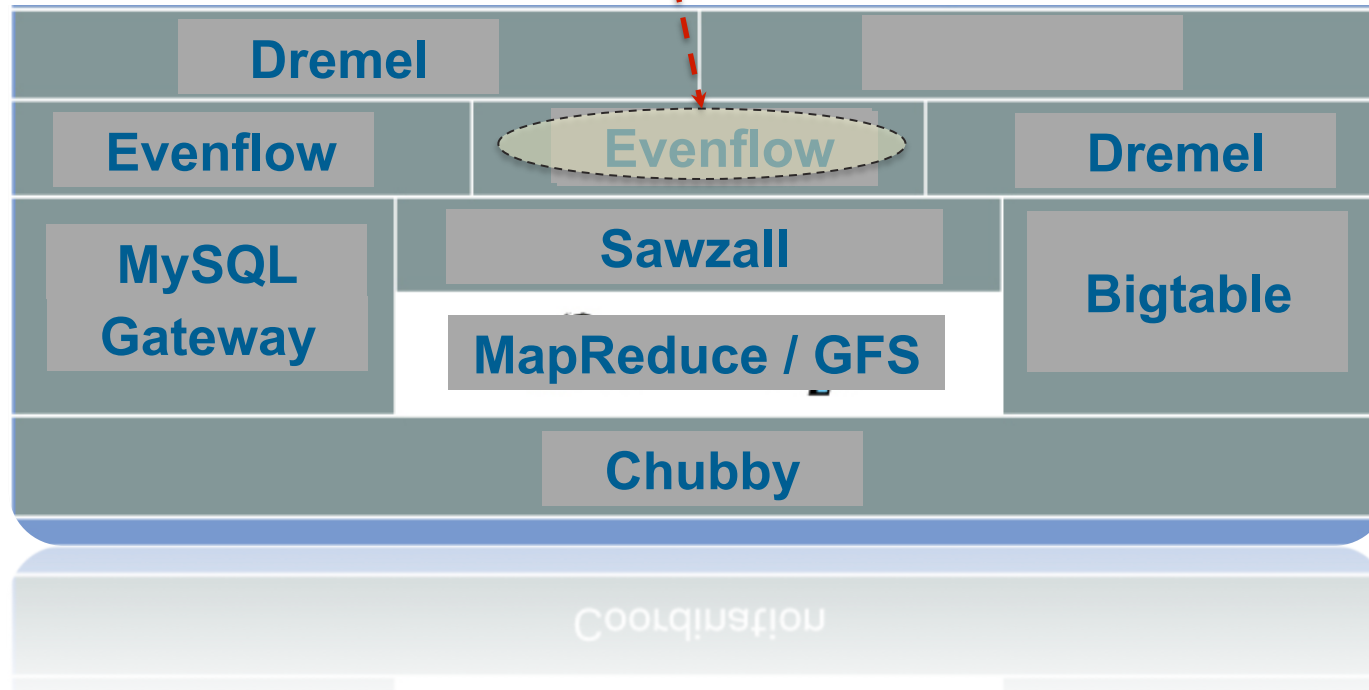
# What did Google do?

**Columnar storage + metadata**

| Dremel | | |
|---|---|---|
| Evenflow | Evenflow | *Dremel* |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| Chubby | | |

cloudera
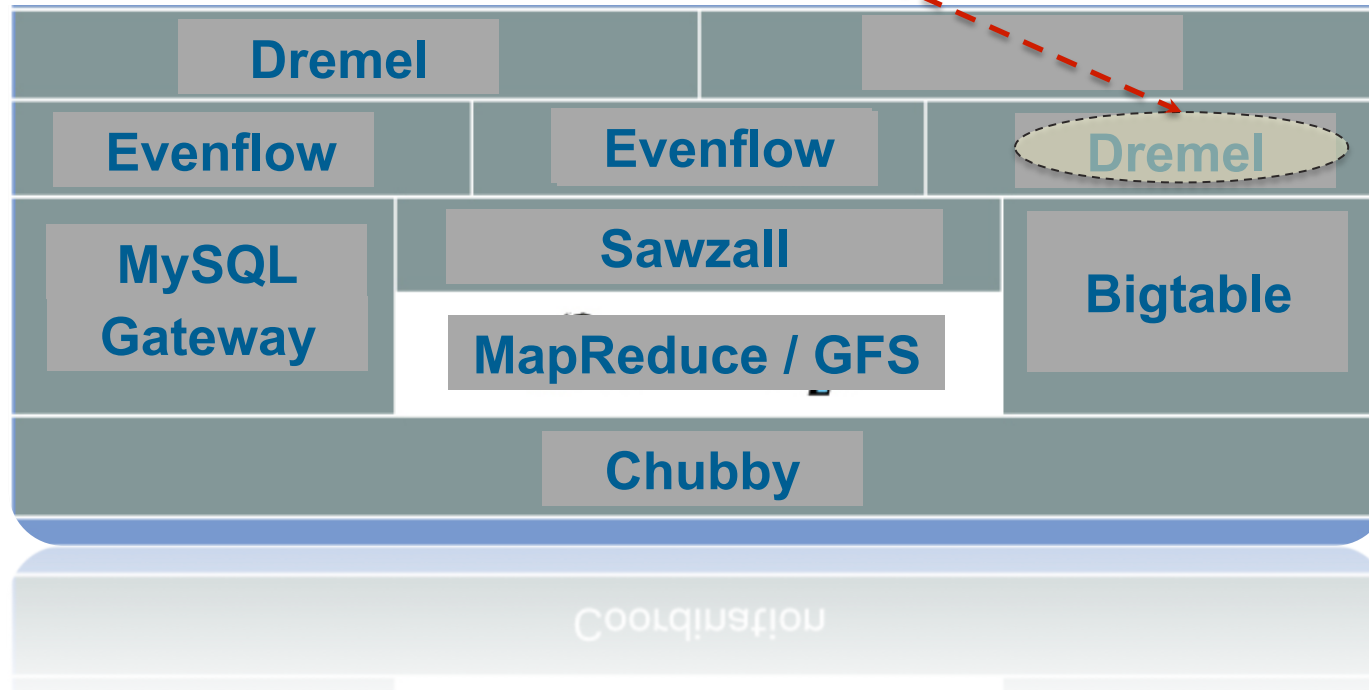
# What did Google do?

Google

**End users query data**

| Dremel | | |
|--------|---|---|
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| | Chubby | |
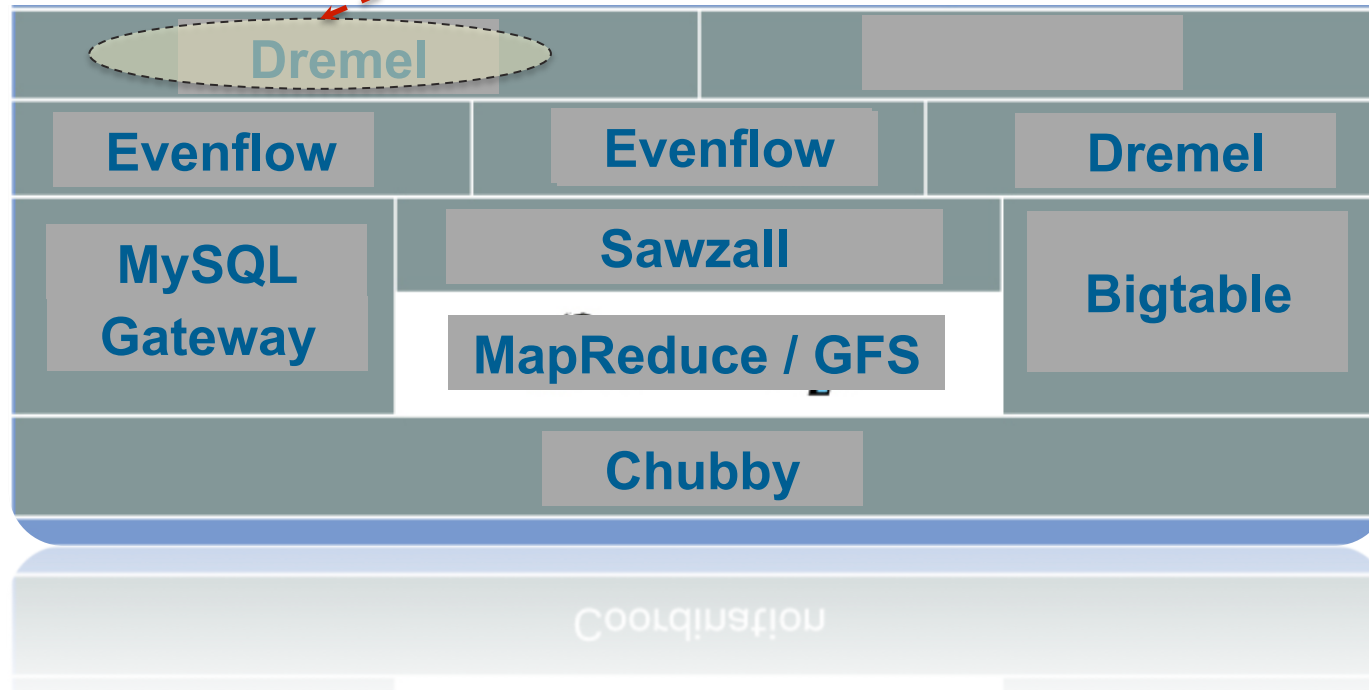
cloudera

# What did Google do?

**Coordinate within system**

Google



| Dremel | |
|--------|--------|
| Evenflow | Evenflow | Dremel |
| MySQL Gateway | Sawzall | Bigtable |
| | MapReduce / GFS | |
| | Chubby | |
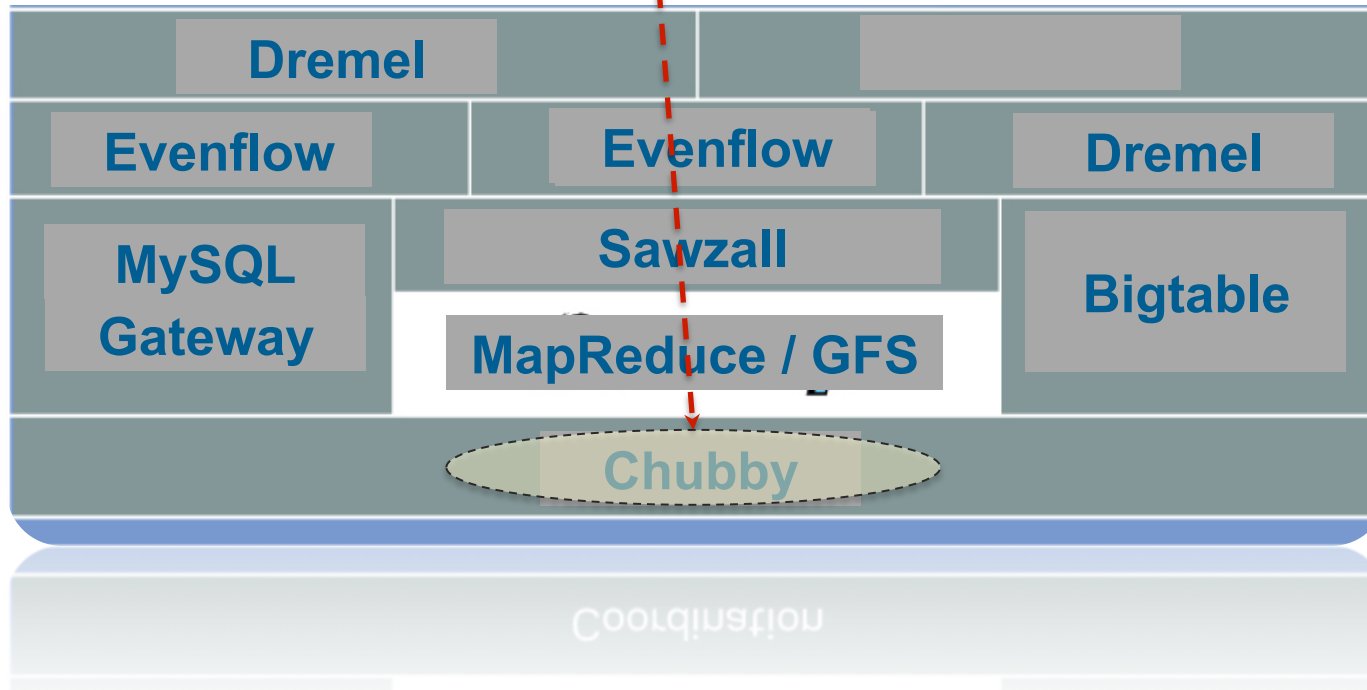
cloudera

# The pattern repeated

YAHOO!

100 employees
5 years
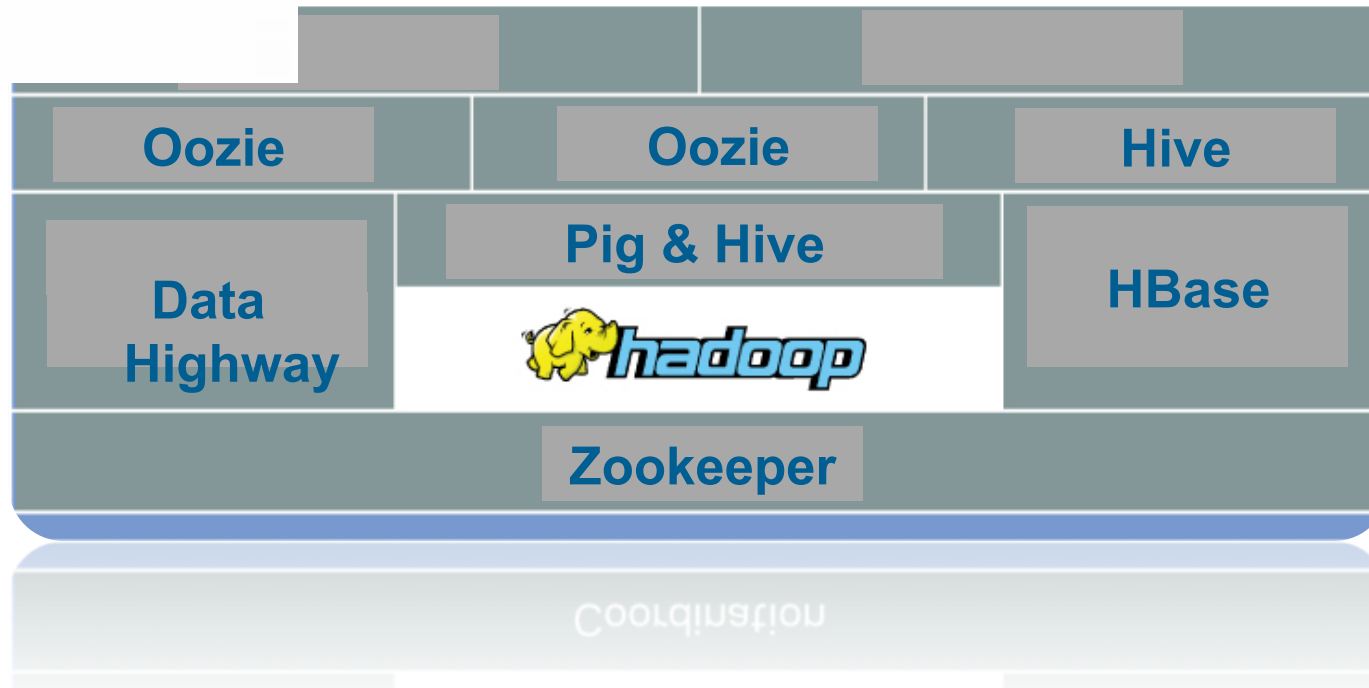
| Oozie | Oozie | Hive |
|-------|-------|------|

| Data Highway | Pig & Hive | HBase |
| | hadoop | |

Zookeeper

cloudera

# The pattern repeated

**Linked in**™

40 employees
4 years

| Azkaban | Azkaban | |
| Sqoop | Pig | Voldemort |
| Kafka | hadoop | |
| Zookeeper | | |
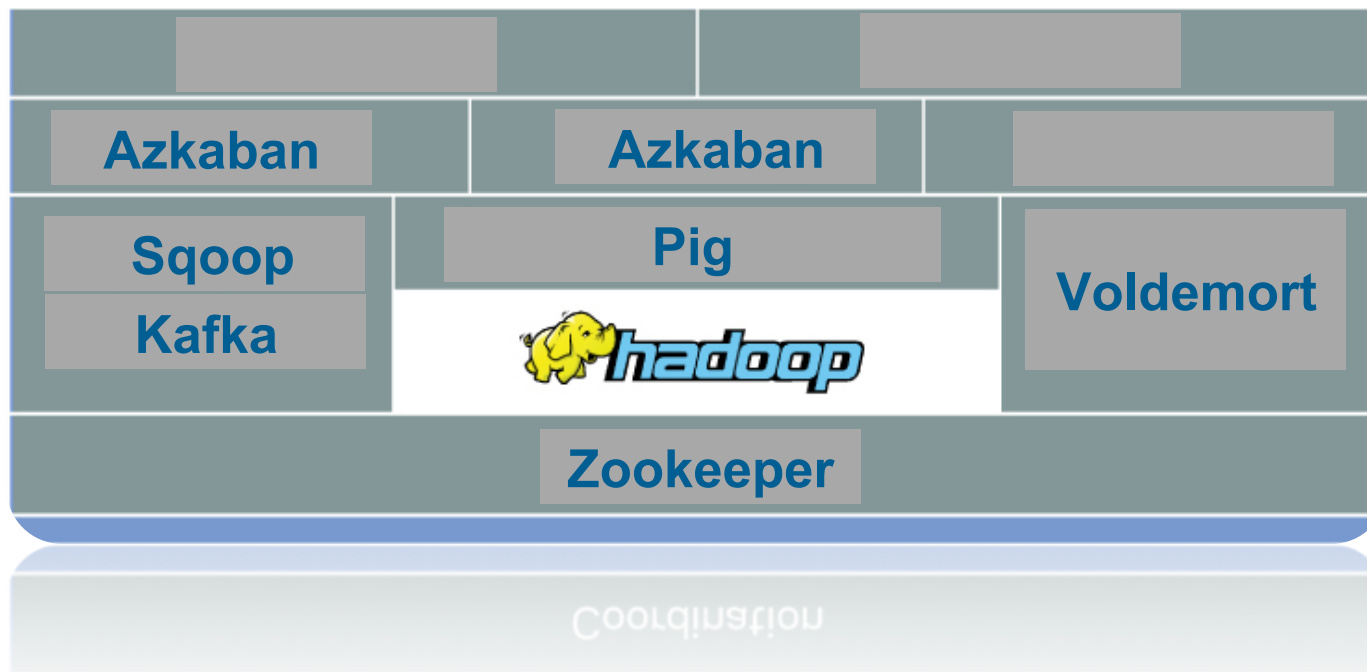
**cloudera**

# The pattern repeated
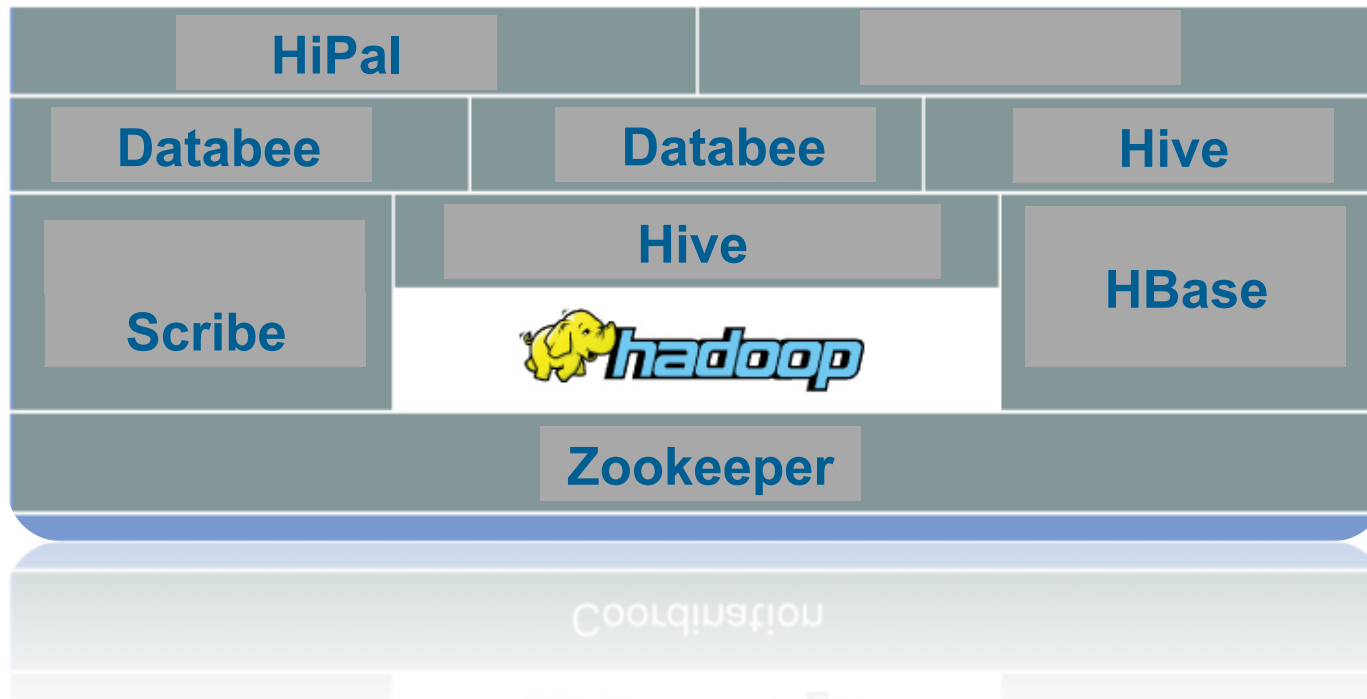
# Assembled for everyone else



cloudera

**Cloudera's Distribution Including Apache Hadoop**

| Hue | | Hue |
|---|---|---|
| Oozie | Oozie | Hive |
| Sqoop Flume | Hive / Pig | HBase |
| | Zookeeper | |

cloudera
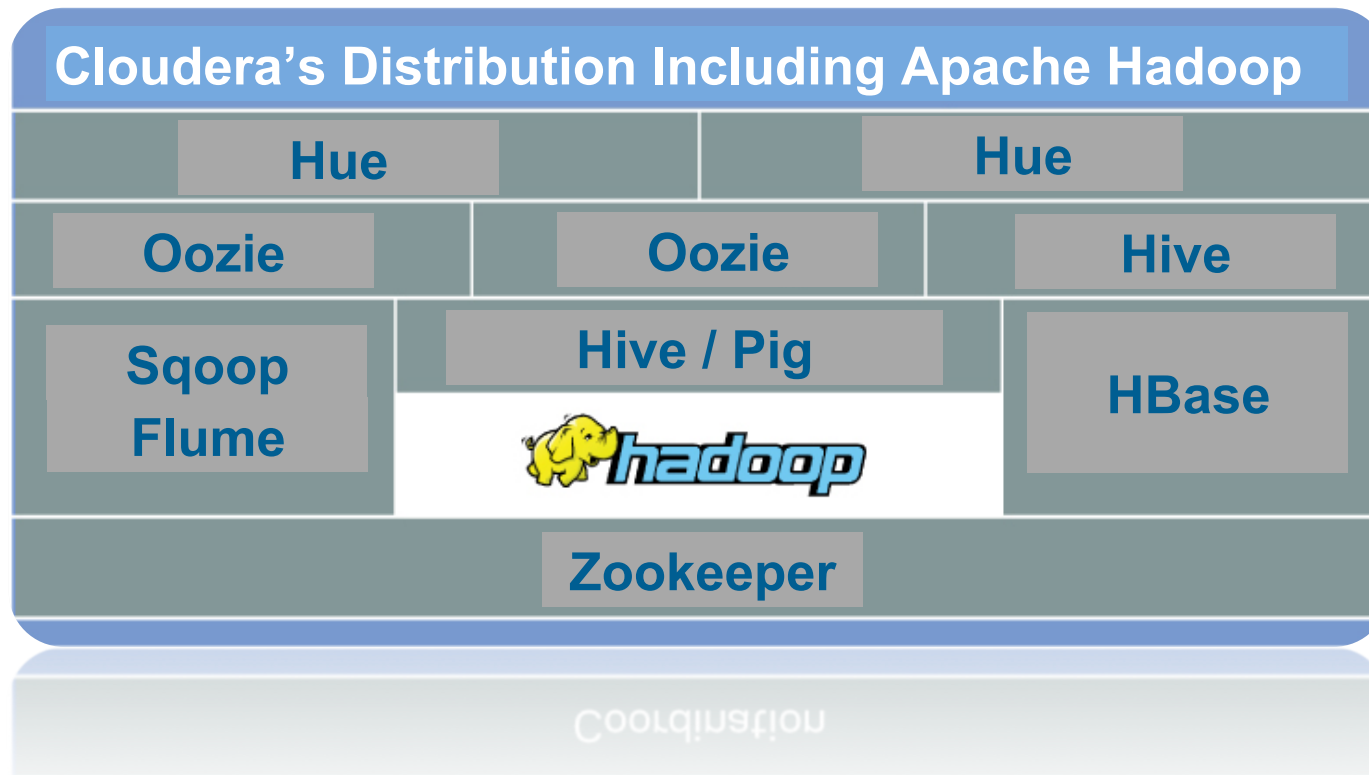
# But more coming…

- Compression
- Additional computational frameworks
- File formats
- Libraries

cloudera

# My point

- You don't need to rethink the past
- Use the current state of the Hadoop stack as your point of departure and build from there
- There are plenty of problems left to solve

cloudera

# Agenda

**1** What Hadoop changes about data

**2** The system – past and present

**3** Living with it – your present and future

**4** Q&A

cloudera

# Making Hadoop fit in the enterprise

cloudera
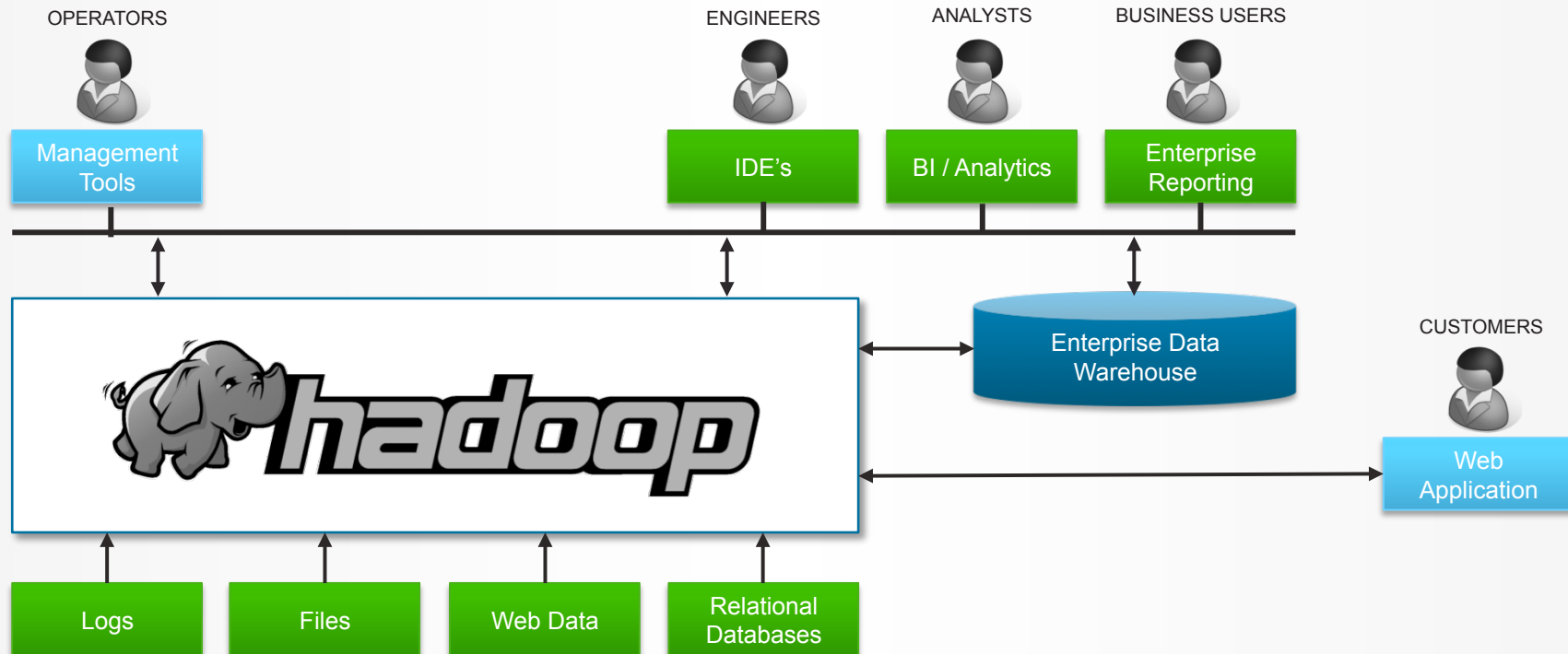
# First principles

- What are the business outcomes Hadoop is supposed to deliver?
  - New insights
  - Lower business costs
  - Lower IT costs
  - More data under management
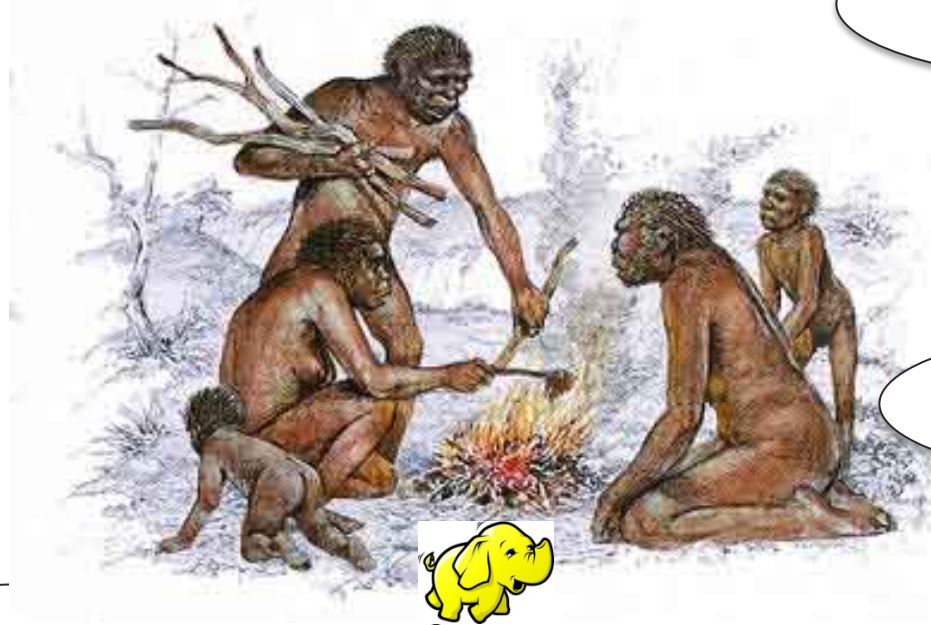  - More revenue through better targeting, conversion
  - ?

**cloudera**

# Operational outcomes?

- Performance
- Utilization
- Cost of operations
- Availability
- Quality of service
- Flexibility / elasticity
- Security
- Transparency
- ?

cloudera

# Evolution – the hunter-gatherers

cloudera
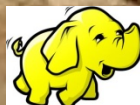
# Evolution – the tribe

# Evolution – the city



Fire is not a big deal any more.
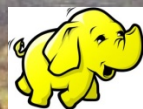
Pollution, congestion, etc a concern

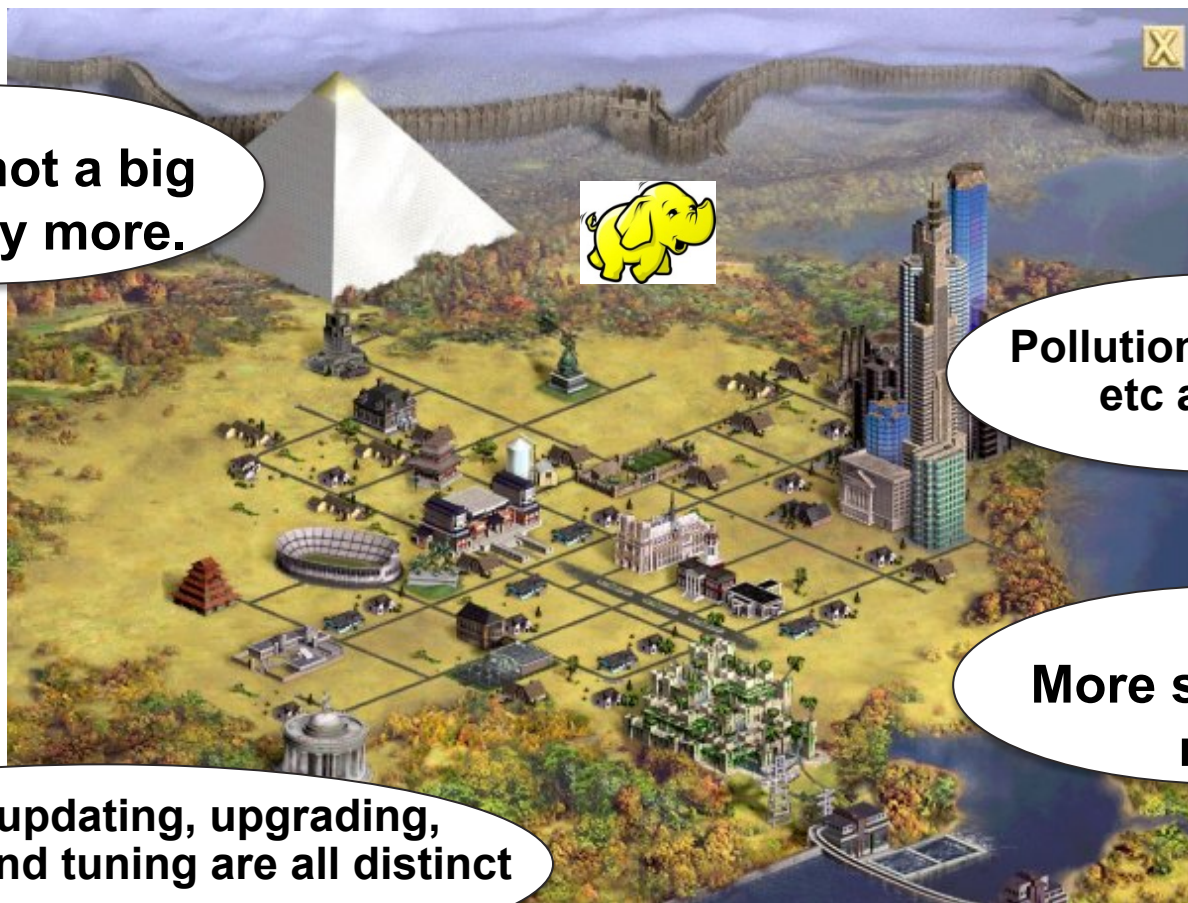More specialized roles

Patching, updating, upgrading, configuring and tuning are all distinct

cloudera

# Some basics – use the right gear

- Hadoop needs to know where it's hard drives are
  - Running on a virtualized layer is a bad idea
  - RAID is a bad idea
  - Running on remote storage the worst idea
- Servers - prioritize flexibility over bells and whistles
  - How easily will you be able to expand your cluster?
  - How easily can you evolve your core / spindle ratio?
  - How many companies support that exotic chip, card, drive, power supply, etc?
- Network – prioritize quality over bells & whistles
  - 10G on the backplane is usually unnecessary
  - Amazingly high failure rates for some gear
  - Plan how to adapt your topology as your cluster grows

cloudera

# Where it's going

cloudera

# Some basics – system should be available and never lose data

- ## Name node availability
  - HA is tricky
  - Manual recovery can be simple, fast, effective
- ## Backup Strategy
  - Name node metadata – hourly, ~2 day retention
  - User data
    - Log shipping style strategies
    - DistCp
    - "Fan out" to multiple clusters on ingestion

cloudera

# The web of scripts – how it starts

- Script to run a check
- Script to import a file
- Script to preempt a job
- Script to instrument a daemon
- Script to….

cloudera

# The web of scripts – where it ends



Nothing ever changes or improves

Garish, jerry-rigged

Time goes into maintaining scripts, not achieving the operational objectives

One and only one person loves it

cloudera

# Script avoidance – data ingestion

- Many data sources
  - Streaming data sources (log files, mostly)
  - RDBMS
  - EDW
  - Files (usually exports from 3$^{rd}$ party)
- Common place we see DIY
  - You probably shouldn't
  - Sqoop, Flume, Oozie (but I'm biased)
- No matter what - fault tolerant, performant, monitored

cloudera

# Script avoidance - ETL and Data Processing

- Non-interactive jobs
- Establish a common directory structure for processes
- Need tools to handle complex chains of jobs
- Workflow tools support
  - Job dependencies, error handling
  - Tracking
  - Invocation based on time or events
- Most common mistake: *depending on jobs always completing successfully or within a window of time.*
  - Monitor for SLA rather than pray

cloudera

# Script avoidance - monitoring

- Helps keep things running
- System monitoring
  - Duh.
  - Traditional monitoring tools: Nagios, Hyperic, Zenoss – use what you already have
  - Host checks, service checks
  - When to alert? It's tricky.

cloudera

# Script avoidance - monitoring

- Hadoop aware cluster monitoring
  - Traditional tools don't cut it; Hadoop monitoring is inherently Hadoop specific
  - Analogous to RDBMS monitoring tools
- Job level "monitoring"
  - More like analysis
  - "What resources does this job use?"
  - "How does this run compare to last run?"
  - "How can I make this run faster, more resource efficient?"
  - Two views we care about
    - Job perspective
    - Resource perspective (task slots, scheduler pool)

cloudera

# Multi-tenancy

- Definition – ability of disparate groups, users, data and workloads to operate concurrently on 1 logical Hadoop system
- Multi-tenancy helps you get more of what you really want
  - Better performance
  - Better cost of operations
  - New insights
  - Greater availability
- Multi-tenancy has some additional considerations

cloudera

# Multi-tenancy – auth and auth

- Authentication
  - Don't talk to strangers
  - Should integrate with existing IT infrastructure
  - Authentication (Kerberos) patches now part of CDH3
- Authorization
  - Not everyone can access everything
    - Ex. Production data sets are read-only to quants / analysts. Analysts have home or group directories for derived data sets.
  - Mostly enforced via HDFS permissions; directory structure and organization is critical
  - Not as fine grained as column level access in EDW, RDBMS (but this is coming)

cloudera

# Multi-tenancy - resources

- "Stop hogging the cluster!"
- Tracking & establishing policies around cluster resources
  - Files, bytes and quotas thereof
  - Tasks, memory, IO, CPU, network and scheduling thereof
  - By now you've almost certainly graduated to a fancy scheduler
- Policies to prevent bad behavior (e.g. auto-kill)
- Monitor and track resource utilization across all groups

cloudera

# My point

- Hadoop likes commodity hardware – just go with it

- Trending toward fewer bigger clusters with new planning required

- Operationalizing Hadoop usage looks very different in the enterprise
  - No committers wearing pagers
  - More things you already own to stitch into
  - Compliance matters more
  - Backup, DR, BCP matters more

cloudera

# Agenda

**1**  What Hadoop changes about data

**2**  The system – past and present

**3**  Living with it – your present and future

**4**  Q&A

cloudera