cloudera
Ask Bigger Questions

Evolution of the Big Data Stack

Jonathan Hsieh | Tech Lead / Software engineer @ Cloudera

BigDataCamp LA '14

June 14, 2014

Who Am I?

- **Cloudera since 2009**
  - **Tech Lead HBase Team**
  - **Software Engineer**
  - **Apache HBase** committer / PMC
  - **Apache Flume** founder / PMC
- **U of Washington:**
  - Research in Distributed Systems

Big Data Stack Evolution

- Inspiration
- Imitation
- Innovation

Big Data Stack Evolution

- Inspiration
- Imitation
- Innovation

cloudera
Ask Bigger Questions

Inspiration

Emergence of Big Data

cloudera
Ask Bigger Questions

> Every two days we create as much information as we did from the dawn of civilization up until 2003.
>
> -- Eric Schmidt
> Google CEO, 2010

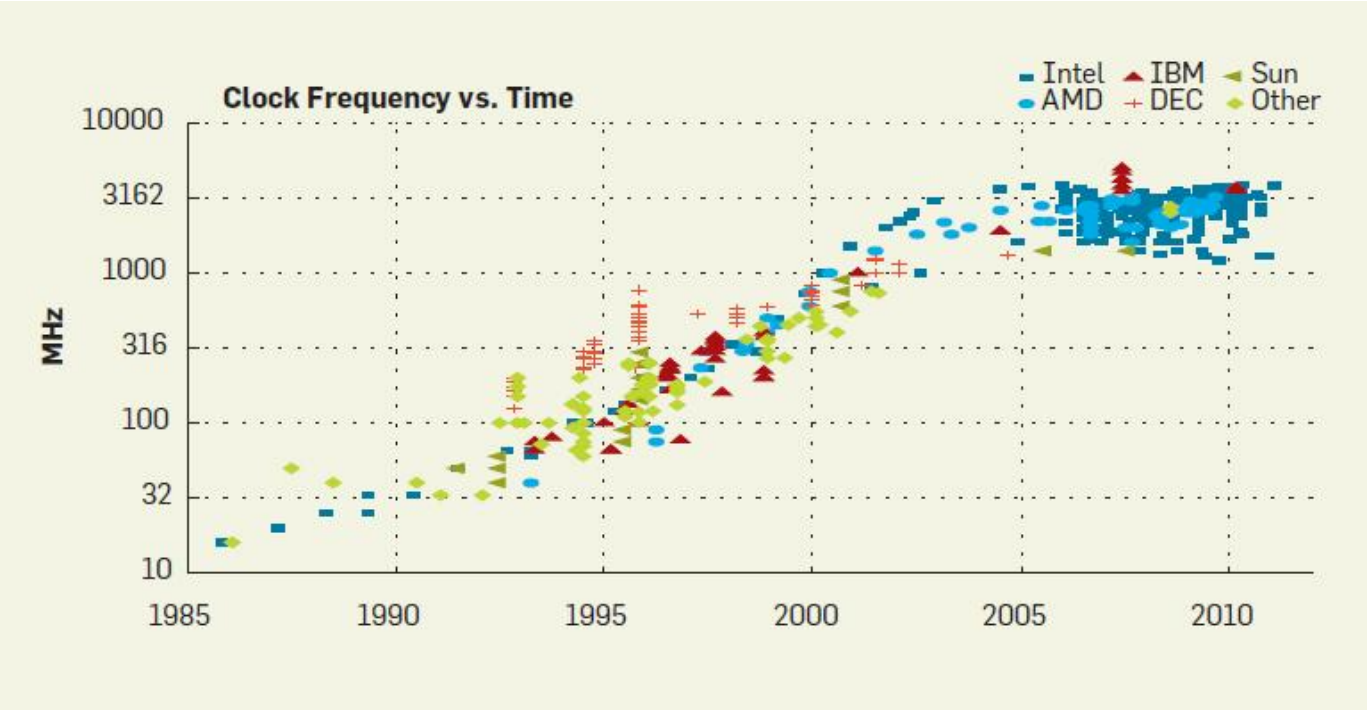
cloudera
Ask Bigger Questions

> The volume, variety and velocity of human communications make our mission more difficult each day.
>
> -- Michael V Hayden
> Director NSA (DIRNSA), 2002

The brute force solution

1. Collect **all** the data
2. Analyze **all** the data
3. Serve the results

End of free MHz coincides with Rise of Big Data



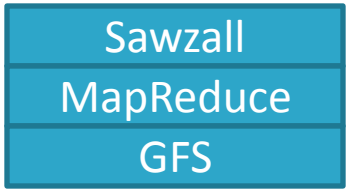
http://cacm.acm.org/magazines/2012/4/147359-cpu-db-recording-microprocessor-history/abstract

A Move towards Distributed Systems

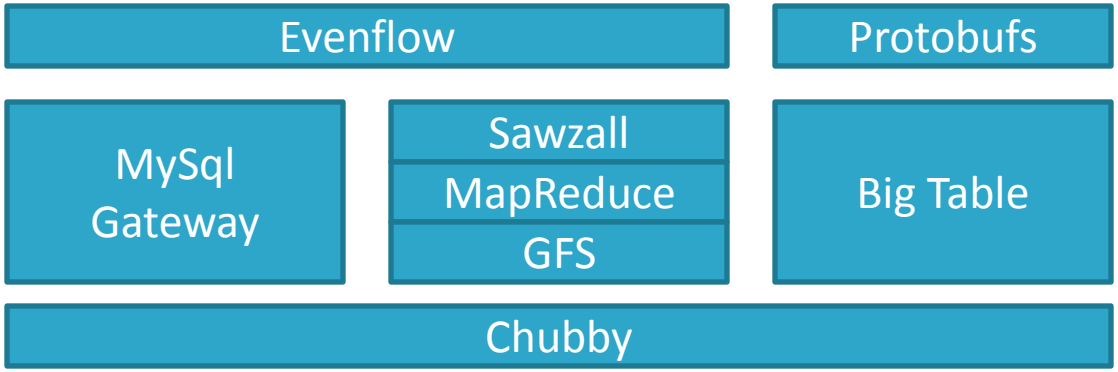
- Scaling **Horizontally** instead of Vertically

- Challenges:
  - Reliability
  - Fault tolerance
    - Atomicity / Consistency / Isolation / Durability
  - High-Availability
  - Latency Predictability

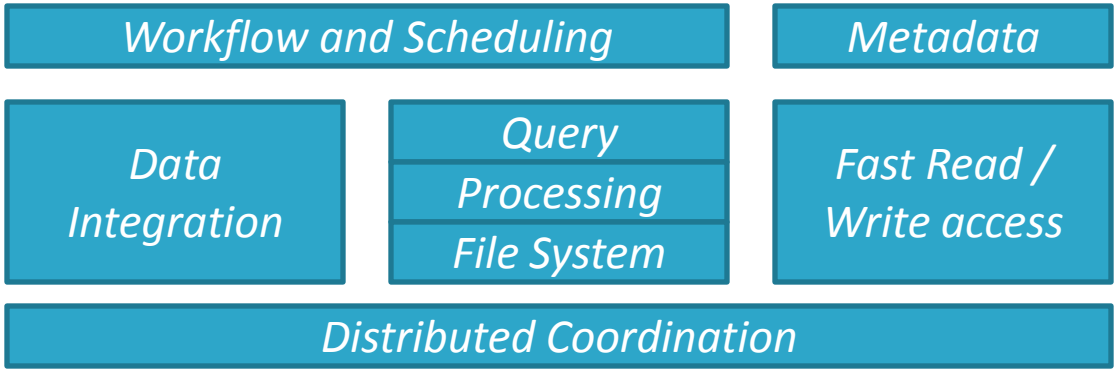
Google built a Big Data Stack



Google built a Big Data Stack



The core of a Big Data Stack



cloudera
Ask Bigger Questions

Imitation

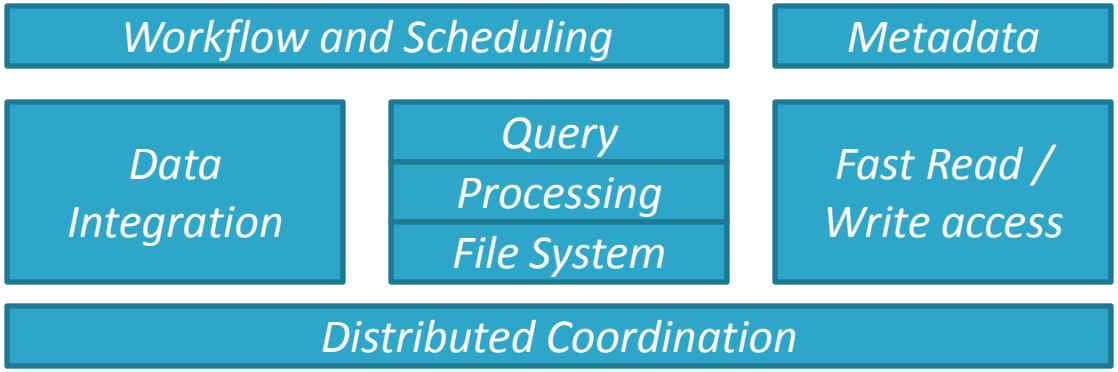
Big Data for the rest of us

cloudera
Ask Bigger Questions

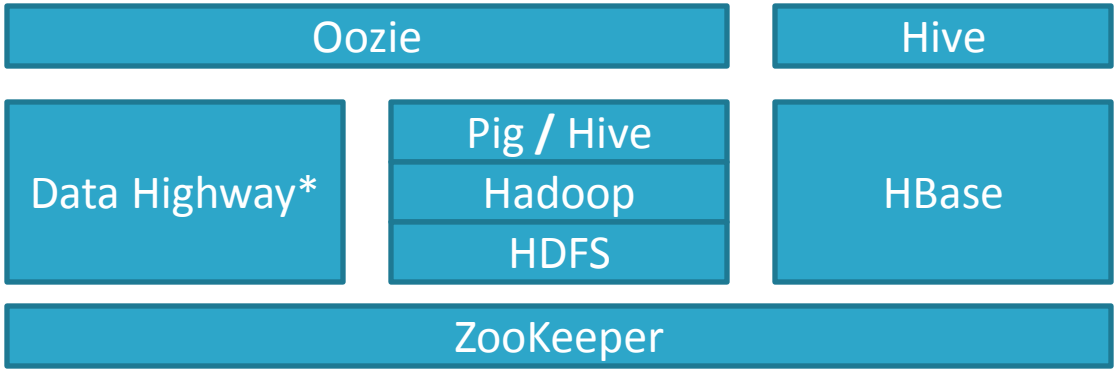
> The future is already here — it's just not very evenly distributed.
>
> -- William Gibson

.

The core of a Hadoop stack



YAHOO! built a Big Data stack



- Donated Hadoop + Friends to the Apache Software Foundation

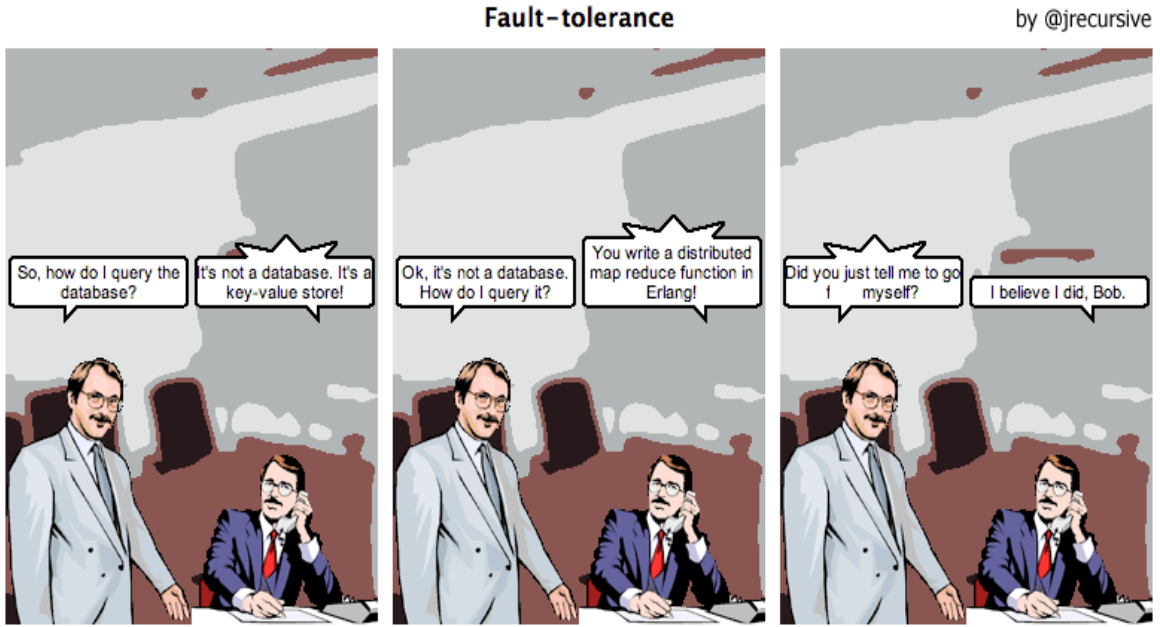
Parallel Components

Function	Google	Yahoo!	Facebook	The Rest of Us
File system	GFS => Colossus	HDFS	HDFS	HDFS
Low latency Data store (NoSQL)	BigTable => Megastore => Spanner	PNUTS => Hbase	HBase	Hbase
Batch processing	Google MapReduce	Hadoop MapReduce	Hadoop MapReduce	Hadoop MapReduce Spark
Batch query	Sawzall, Tenzing, FlumeJava	Pig	Hive	Pig, Hive, Impala, Drill, Crunch
Resource Management	Borg => Omega	=> YARN	=> Corona	YARN Mesos
Ingest	EvenFlow Custom MySQL Proxy	Custom	Scribe / Calligraphus Custom proxy	Sqoop Flume Kafka
Coordination	Chubby	Zookeeper	Zookeeper	Zookeeper
Graph Processing	Pregel		Giraph	Giraph, Golden orb Hama, Titan
Stream processing	MillWheel	S3 => Storm	Puma/PTail	Storm, Spark

Simplify and remove features to enable scaling

- Scalable and simple first
- Focus only on needed features. Exclude others.
- Re-add them later.

- Ex: NoSQL
  - No transactions
  - No Schema

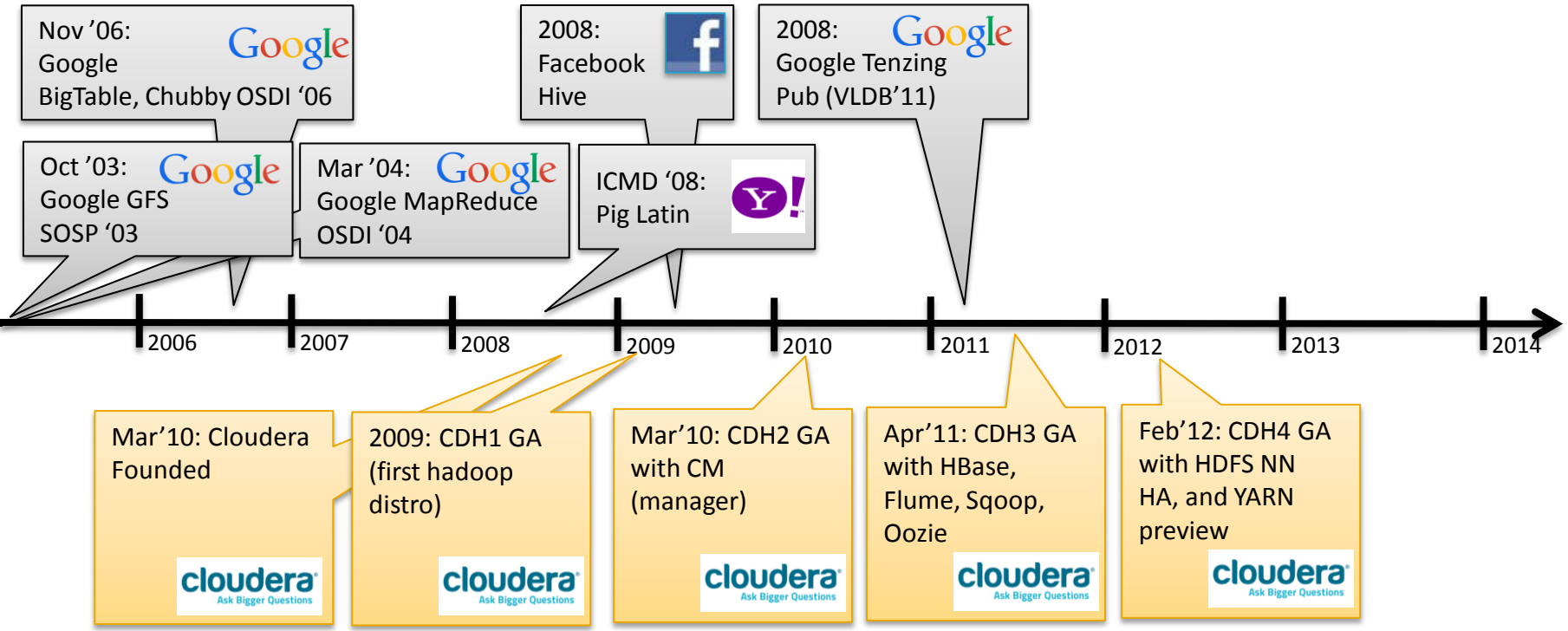


cloudera
Ask Bigger Questions

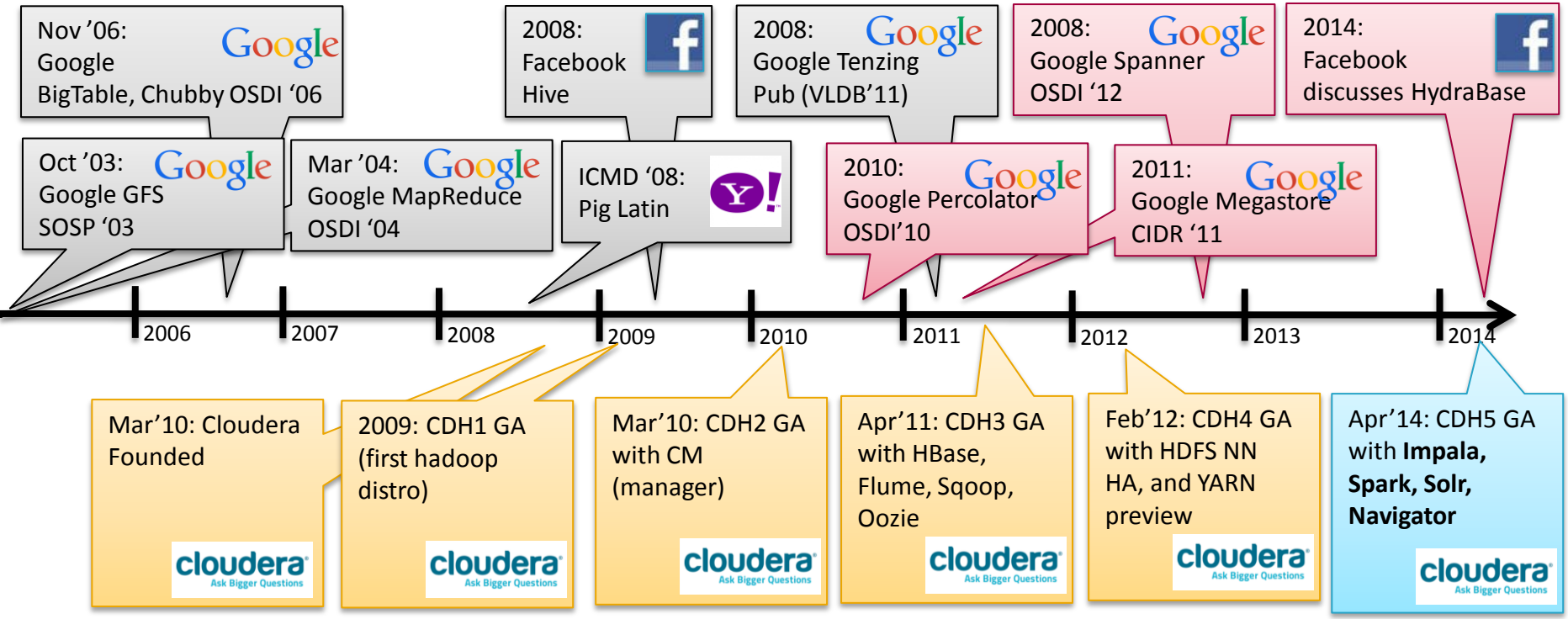
Innovation

Big Data industry steps up

Big Data Stack Timeline



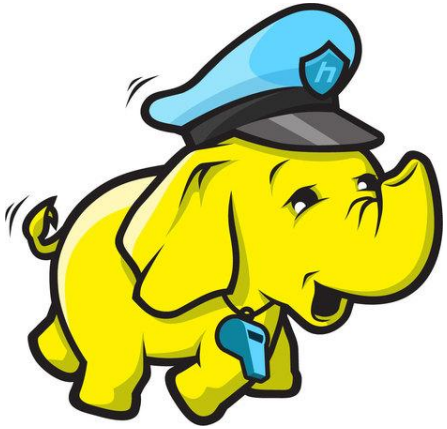
Big Data Stack Timeline



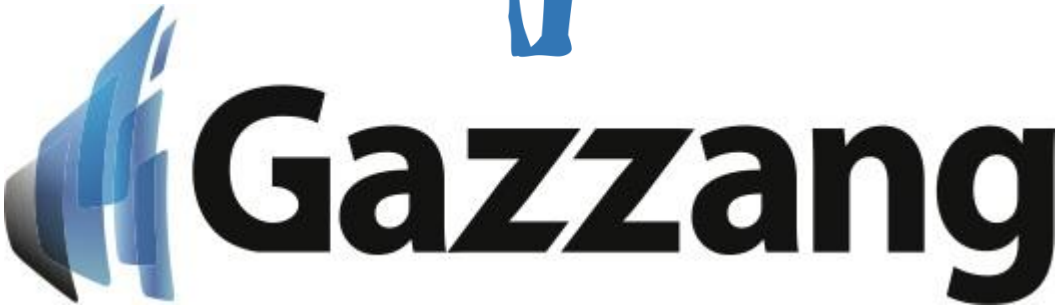
Usability


HUE
HIVE
cloudera
IMPALA
Phoenix
"We put the SQL back in NoSQL"
Trafodion
splice
MACHINE
CONTINUUITY

Security + Integration



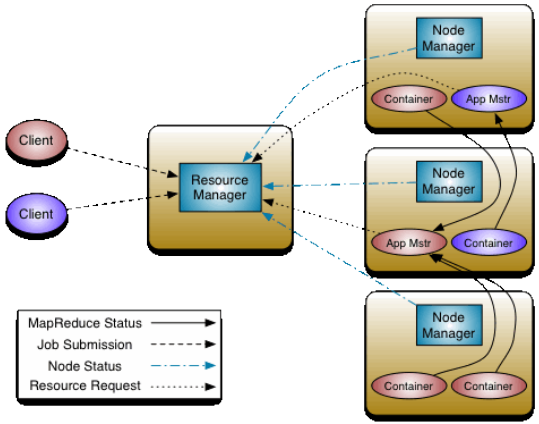
KNOX



New directions

oryx

Spark



MESOS

cloudera
Ask Bigger Questions

Thanks!
@jmhsieh