cloudera
Ask Bigger Questions

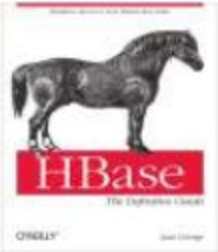
Apache HBase Application Archetypes

Lars George | @larsgeorge | Cloudera EMEA Chief Architect | HBase PMC
Jonathan Hsieh | @jmhsieh | Cloudera HBase Tech lead | HBase PMC
HBaseCon 2014
May 5th , 2014

About Lars and Jon

Lars George

- EMEA Chief Architect @Cloudera
  - Apache HBase PMC
  - O'Reilly Author of *HBase – The Definitive Guide*
- Contact
  - lars@cloudera.com
  - @larsgeorge



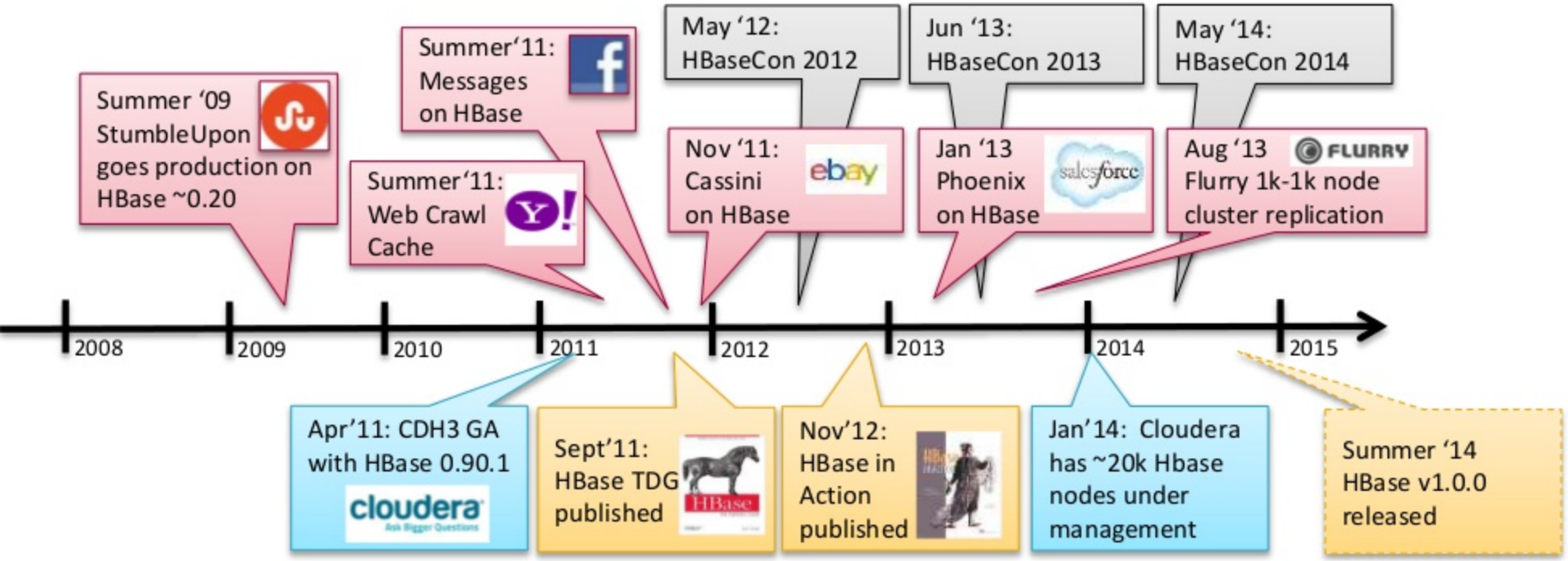
Jon Hsieh

- Tech Lead HBase Team @Cloudera
  - Apache HBase PMC
  - Apache Flume founder
- Contact:
  - jon@cloudera.com
  - @jmhsieh

About Supporting HBase at Cloudera

- Supporting Customers using HBase since 2011
  - HBase Training
  - Professional Services
- Team has experience supporting and running HBase since 2009
  - 8 committers on staff
  - 2 HBase book authors
- As of Jan 2014, ~20,000 HBase nodes (in aggregate) under management
- Information in this presentation is either aggregated customer data or from public sources.

An Apache HBase Timeline



Apache HBase "Nascar" Slide

Outline

- Definitions
- Archetypes
  - The Good
  - The Bad
  - The Maybe
- Conclusion

Definitions

A vocabulary for HBase Archetypes

Defining HBase Archetypes

- There are a lot of HBase applications
  - Some successful, some less so
  - They have common architecture patterns
  - They have common tradeoffs
- **Archetypes** are common architecture patterns
  - Common across multiple use-cases
  - Extracted to be repeatable

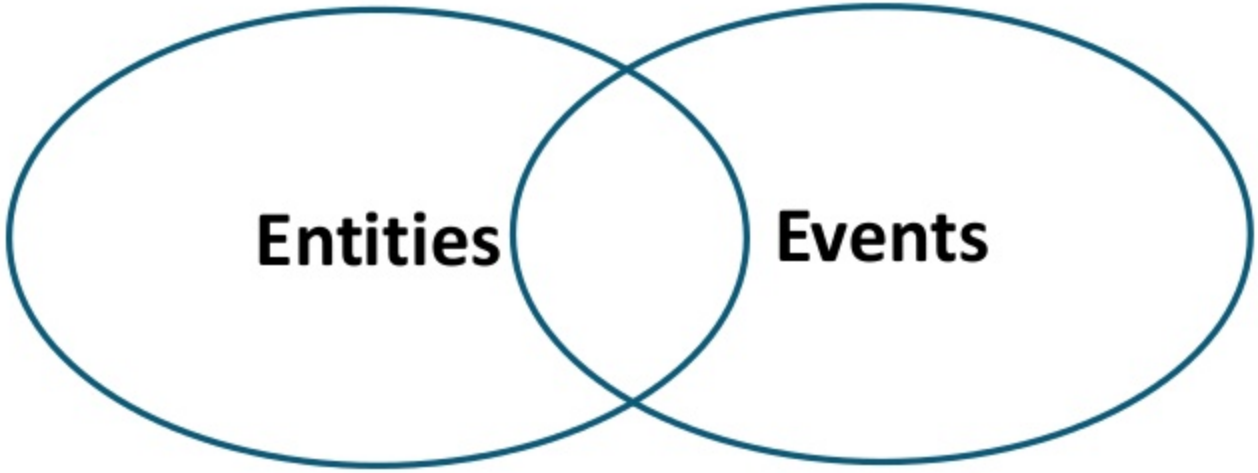
- Our Goal: Define patterns à la “Gang of Four” (Gamma, Helm, Johnson, Vlissides)

So you want to use HBase?

- What data is being stored?
  - Entity data
  - Event data
- Why is the data being stored?
  - Operational use cases
  - Analytical use cases
- How does the data get in and out?
  - Real time vs. Batch
  - Random vs. Sequential

What is being stored?

There are primarly two kinds of big data workloads. They have different storage requirements.



Entity Centric Data

- **Entity** data is information about **current state**
  - Generally real time reads and writes
- Examples:
  - Accounts
  - Users
  - Geolocation points
  - Click Counts and Metrics
  - Current Sensors Reading
- Scales up with # of Humans and # of Machines/Sensors
  - Billions of distinct entities

Event Centric Data

- **Event** centric data are time-series data points recording successive points spaced over time intervals.
  - Generally real time write, some combination of real time read or batch read
- Examples:
  - Sensor data over time
  - Historical Stock Ticker data
  - Historical Metrics
  - Clicks time-series

- Scales up due to finer grained intervals, retention policies, and the passage of time

Events about Entities

- Majority Big Data use cases are dealing with event-based data
  - **|Entities| * |Events| = Big data**

- When you ask questions, do you hone in on entity first?
- When you ask questions, do you hone in on time ranges first?

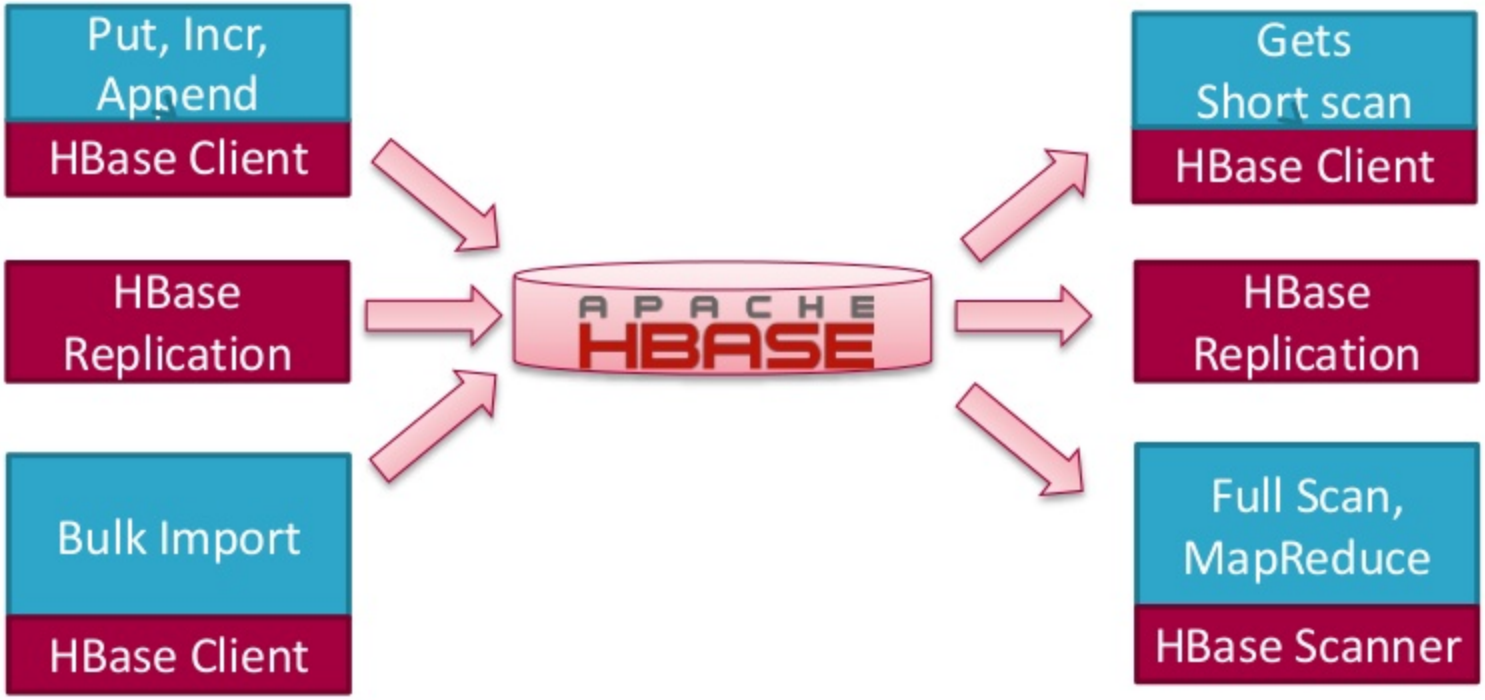
- Your answer will help you determine where and how to store your data.

Why are you storing the data?

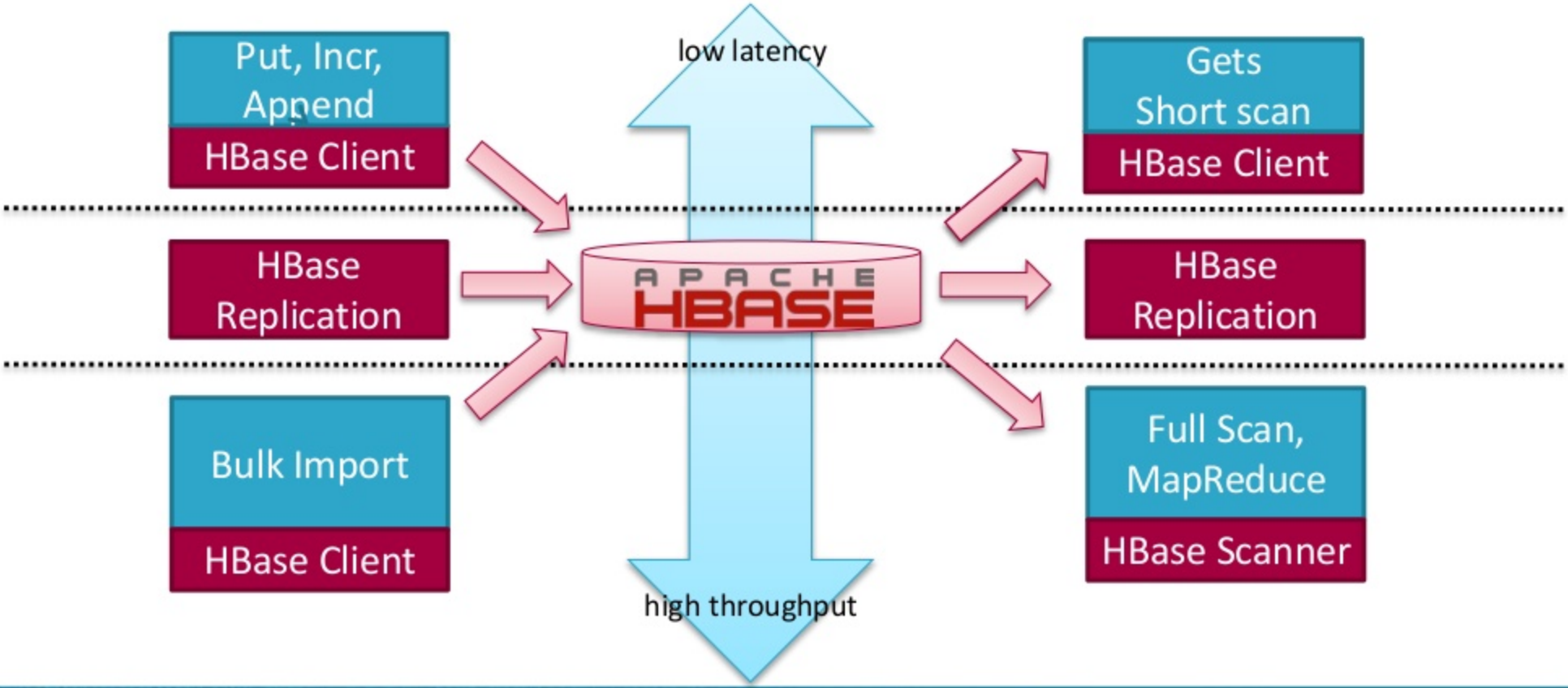
- So what kind of questions are you asking the data?

- Entity-centric questions
  - Give me everything about entity **e**
  - Give me the most recent event **v** about entity **e**
  - Give me the **n** most recent events **V** about entity **e**
  - Give me all events **V** about **e** between time **[t1,t2]**
- Event and Time-centric questions
  - Give me an aggregates on each entity between time **[t1,t2]**
  - Give me an aggregate on each time interval for entity **e**
  - Find events **V** that match some other given criteria

How does data get in and out of HBase?



How does data get in and out of HBase?



What system is most efficient?

- It is all physics
- You have a limited I/O budget
  - Use all your I/O by parallelizing access and read/write sequentially.
  - Choose the system and features that reduces I/O in general
- Pick the systems best for your workload

IOPs/s/disk

The physics of Hadoop Storage Systems

Workload	HBase	HDFS
Low latency	ms, cached	mins, MR seconds, Impala
Random Read	primary index	index?, small files problem

The physics of Hadoop Storage Systems

Workload	HBase	HDFS
Low latency	➕ ms, cached	➖ mins, MR ⚠ seconds, Impala
Random Read	➕ primary index	⚠ index?, small files problem
Short Scan	➕ sorted	⚠ partition
Full Scan	➖ live table ⚠ (MR on snapshots)	➕ MR, Hive, Impala

The physics of Hadoop Storage Systems

Workload	HBase	HDFS
Low latency	➕ ms, cached	➖ mins, MR ⚠ seconds, Impala
Random Read	➕ primary index	⚠ index?, small files problem
Short Scan	➕ sorted	⚠ partition
Full Scan	➖ live table ⚠ (MR on snapshots)	➕ MR, Hive, Impala
Random Write	➕ log structured	✖ not supported
Sequential Write	➖ HBase overhead ⚠ bulk load	➕ minimal overhead
Updates	➕ log structured	✖ not supported