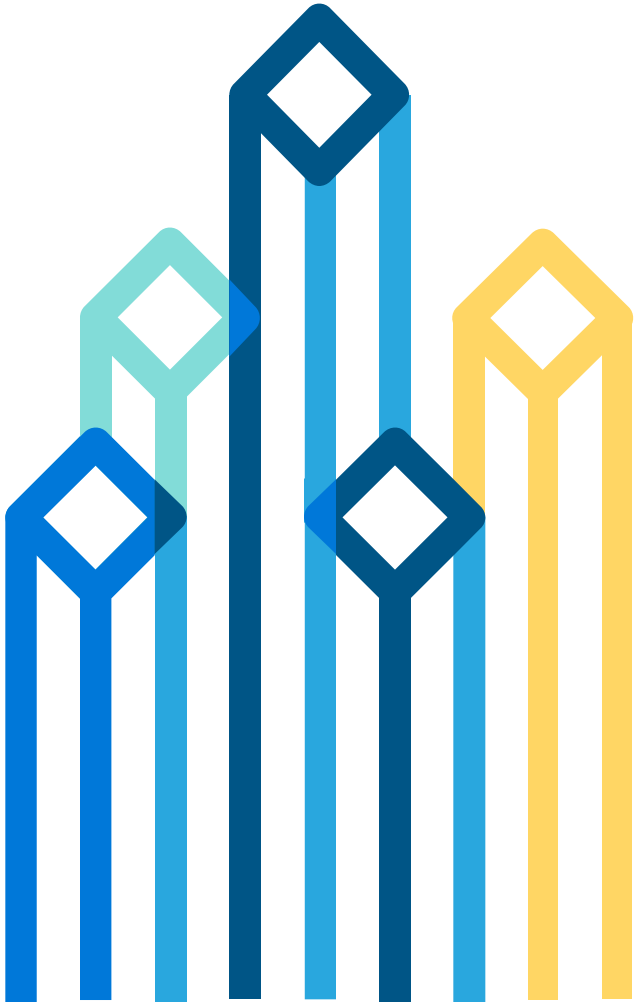
cloudera

Apache HBase: Overview, Hands-On, and Use Cases

Apekshit Sharma

Esteban Gutierrez



Who we are

Apekshit Sharma

- Distributed Software Engineer, Cloudera
  - Software Engineer, Google
- Apache HBase contributor
- Performance improvements and configuration framework

Esteban Gutierrez

- Software Engineer, Cloudera
- Apache HBase committer
- +15 years of experience on Internet scale platforms.

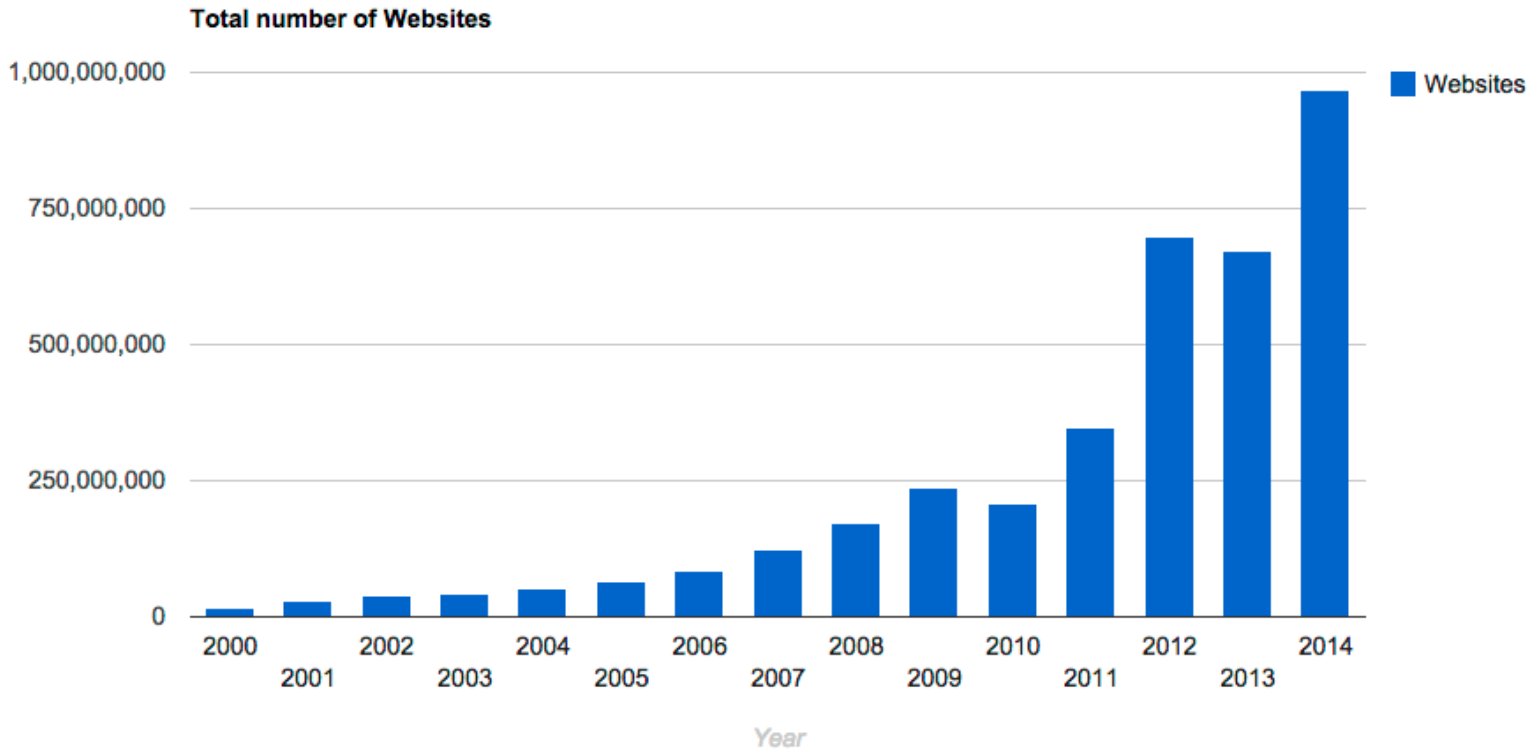
cloudera

Contents

- Motivation
- Introduction to Apache HBase
  - Data model
- Hands-On: Installation, HBase shell
- A more in-depth introduction to Apache HBase
  - Apache Hadoop HDFS
  - HBase Architecture

cloudera

Motivation



http://www.internetlivestats.com/total-number-of-websites/

cloudera

Motivation

- Indexing the internet has challenges:
  - Scale
    - Volume
    - Rate
  - Diversity of content
    - URLs
    - High-resolution images
    - Video

cloudera

Motivation

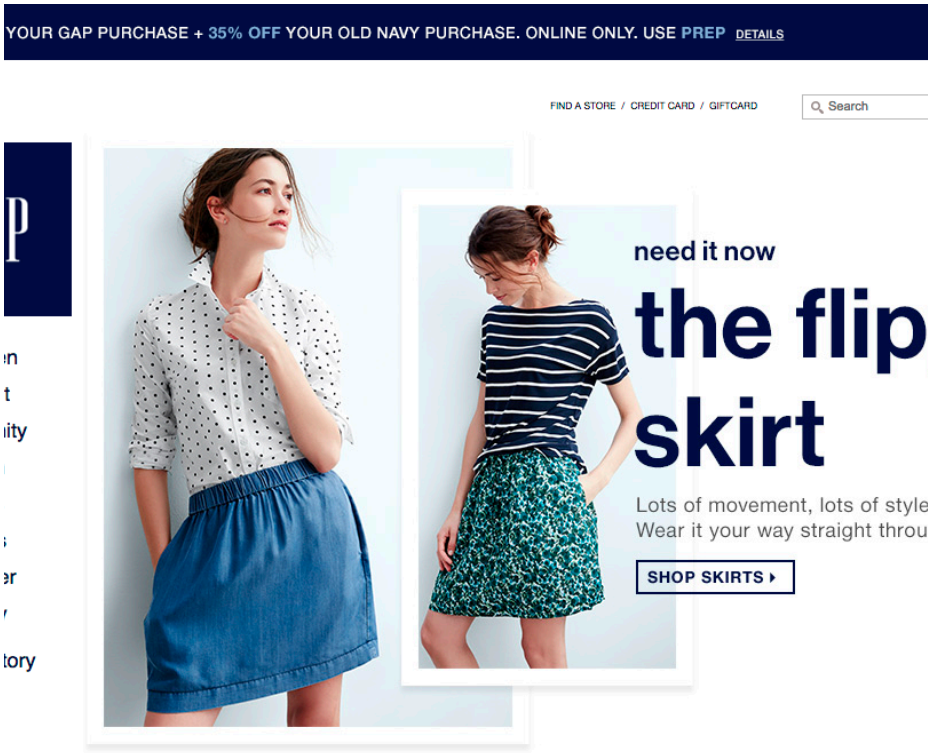
Bigtable: A Distributed Storage System for Structured Data

Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach
Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber
{fay,jeff,sanjay,wilsonh,kerr,m3b,tushar,fikes,gruber}@google.com
Google, Inc.

cloudera

Motivation

- What if you're not trying to index the internet?

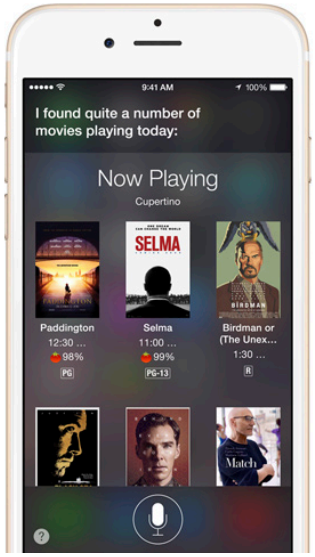


cloudera

Motivation

- Data for analytical processing
- User-facing real-time platforms

"What movies are playing today?"



cloudera

Apache HBase

- Open source
- Random access, Low latency and Consistent data store
- Horizontally scalable
- Built on top of Apache Hadoop

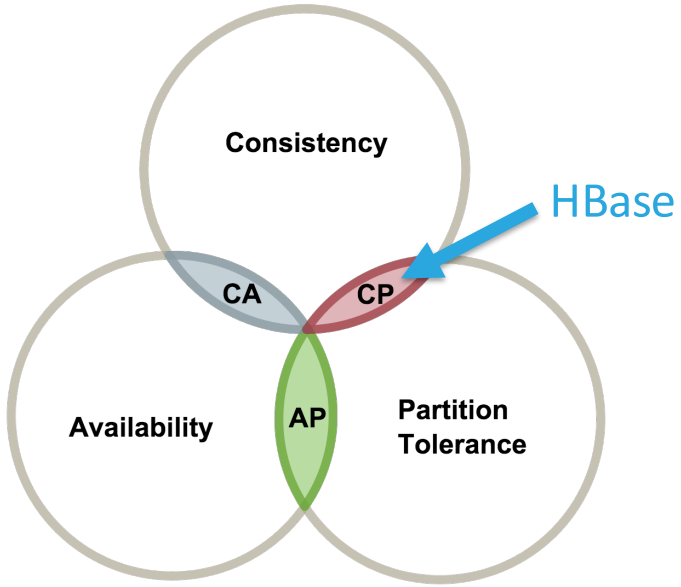
cloudera

Apache HBase is open source

- Apache 2.0 License
- A community project with committers and contributors from diverse organizations
  - Cloudera, Facebook, Salesforce.com, Huawei, TrendMicro, eBay, HortonWorks, Intel, Twitter, …
- Code license means anyone can modify and use the code.

Apache HBase is consistent

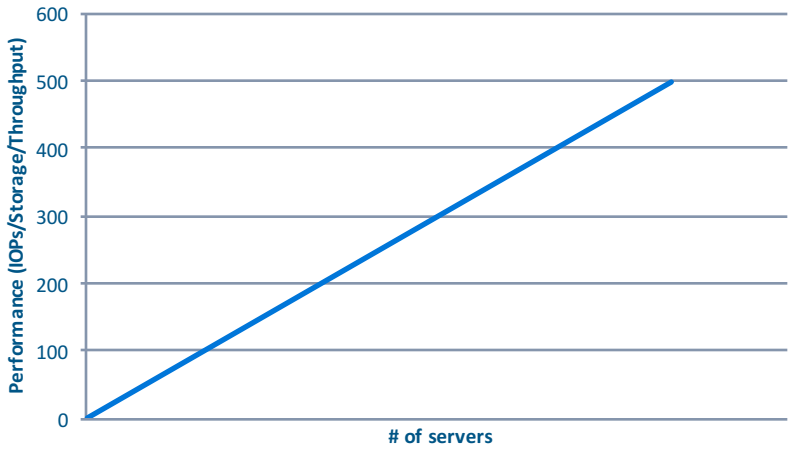
- CAP theorem
  - Consistency
  - Availability
  - Partition tolerance



cloudera

Apache HBase is horizontally scalable

- Adding more servers linearly increases performance and capacity
  - Storage capacity
  - Input/output operations
- Store and access data on commodity servers

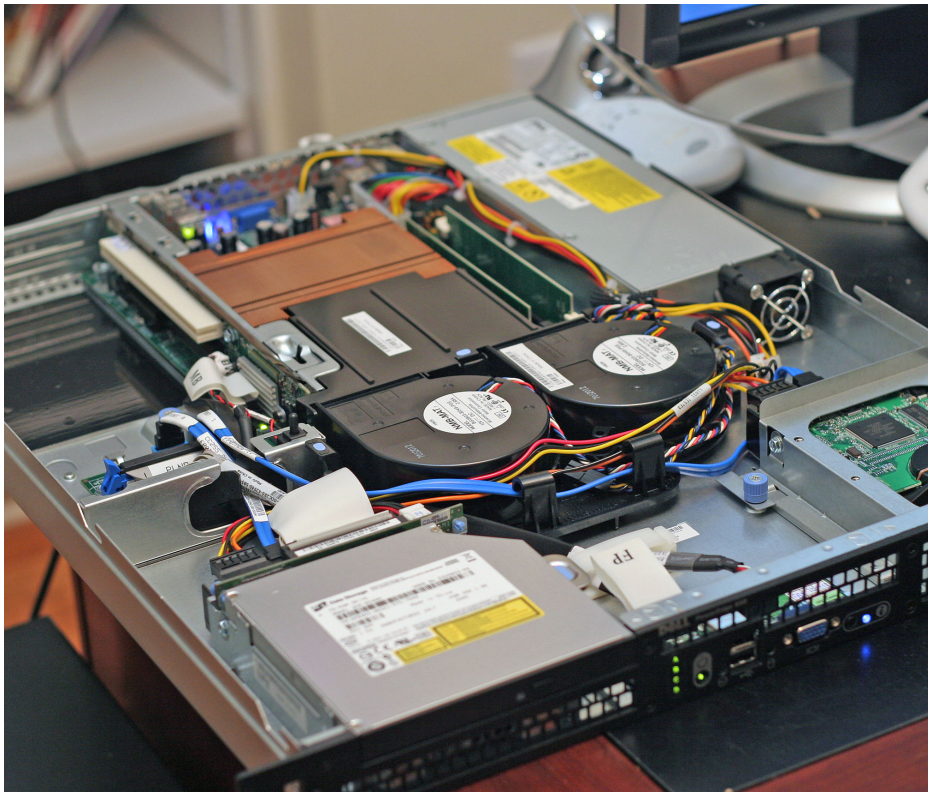


- Largest cluster: > 3000 nodes, > 100 PB
- Average cluster: 10-40 nodes, 100-400TB

cloudera

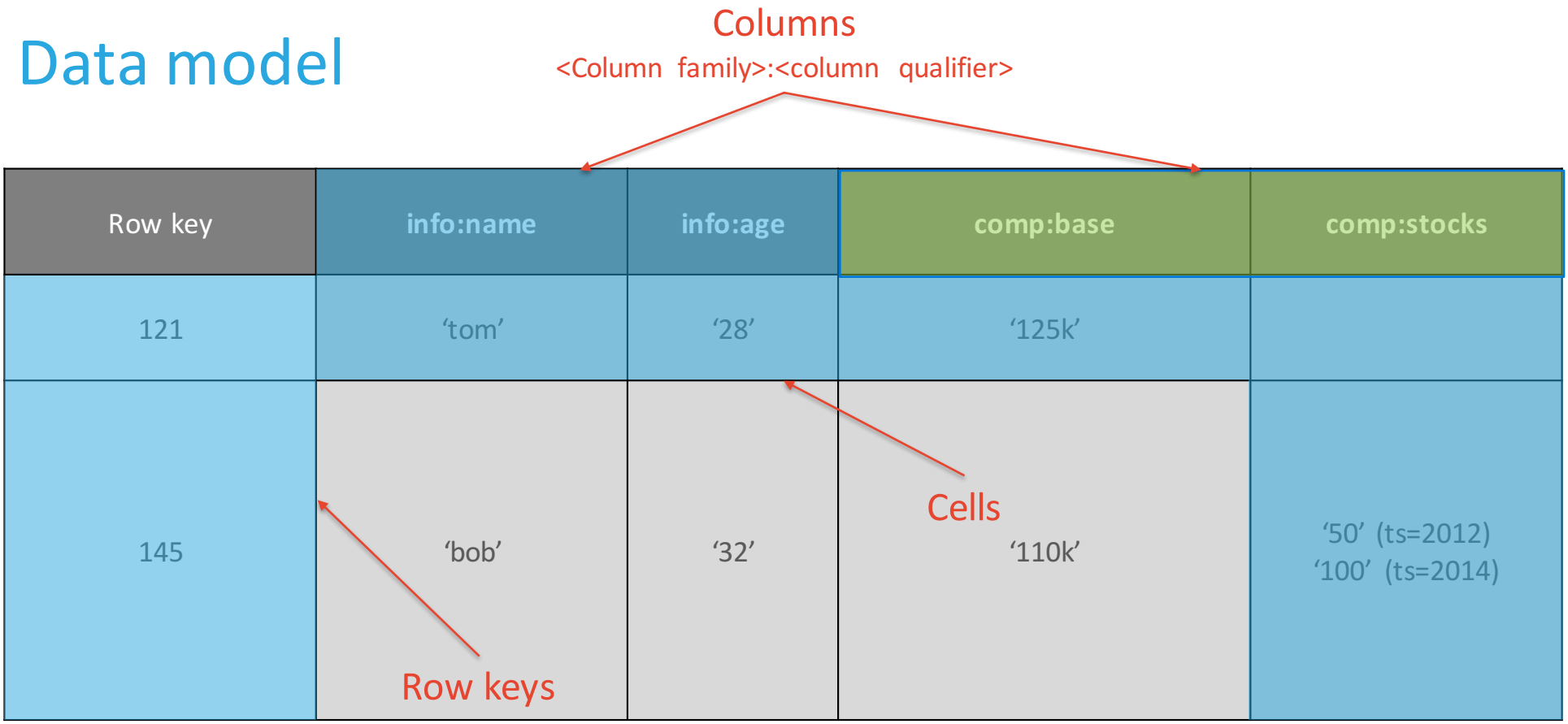
Apache HBase is horizontally scalable

- Commodity servers (circa 2015)
  - 12-24 2-4TB hard disks
  - 2 (quad/hexa/octa)-core CPUs, 2-3 GHz
  - 64 - 512 GBs of RAM
  - 10 Gbps ethernet
  - $5,000 - $10,000 / machine



cloudera

Data model



Data model

- Data is stored… in a big table
- Sorted rows with primary row keys. Supports billions of rows.
- Column : <column family>:<column qualifier>; Supports millions of columns
- Cell : intersection of row and column.
  - Can be empty. No storage/processing overheads
  - Can have different time-stamped values

cloudera

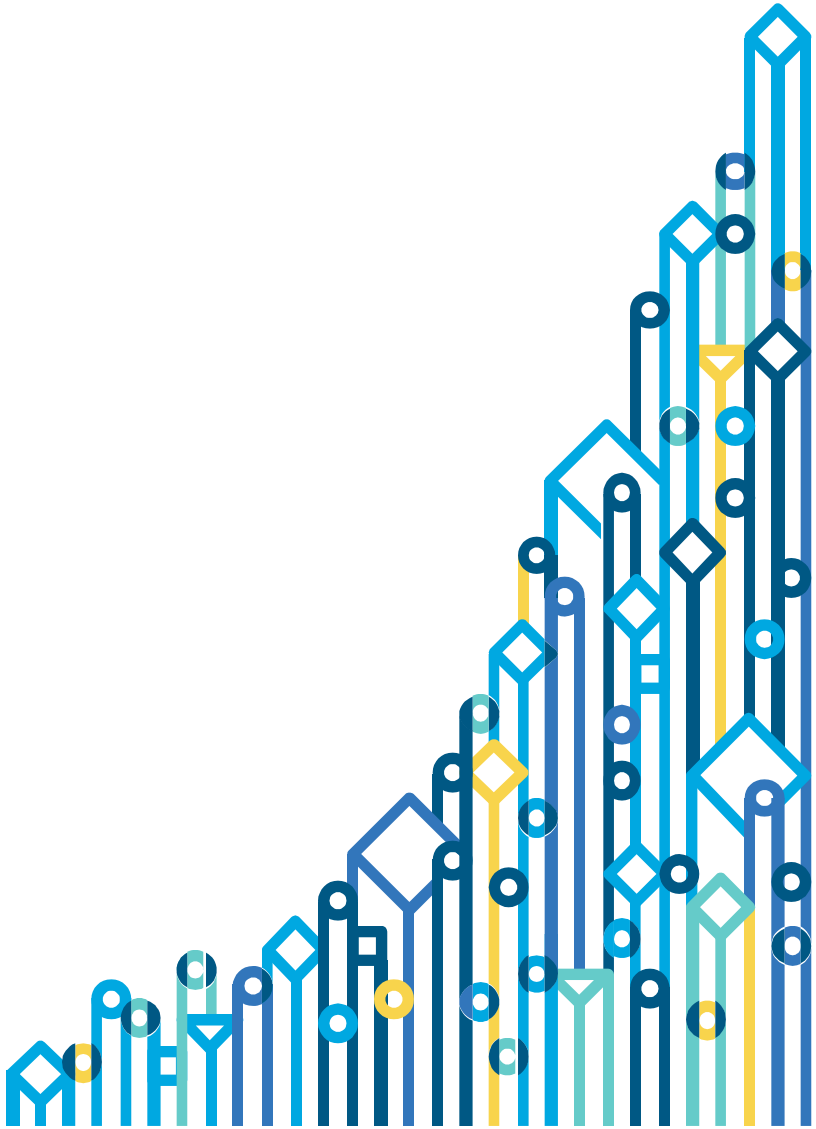
cloudera®

Hands On

Apache HBase installation

Play with HBase shell

http://pastebin.com/60EUN75f



Understanding basic architecture of

Hadoop (HDFS)

Apache Hadoop

- Open source
- Commodity servers
- Horizontally scalable
- Highly fault-tolerant
- Massive processing power

cloudera

2 Core Components

Apache Hadoop

HDFS
(Hadoop Distributed File System)

MapReduce + YARN

cloudera

History

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google*

2003

cloudera

GFS

- Distributed file system
- Commodity servers
- horizontally scalable
- Highly fault-tolerant
- **Proprietary**

cloudera

HDFS

- Distributed file system
- Commodity servers
- Horizontally scalable
- Highly fault-tolerant
- Open source

cloudera

HDFS API

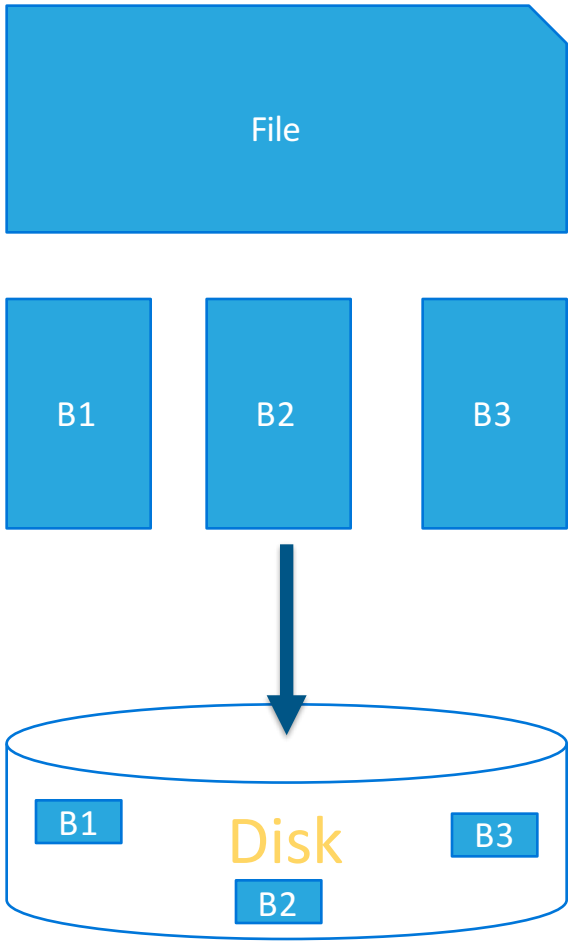
- File
  - Open, Close, Read, Write, Move, etc
- Directories
  - Create, Delete, etc
- Permissions
  - Owners, Groups, rwx permissions

cloudera

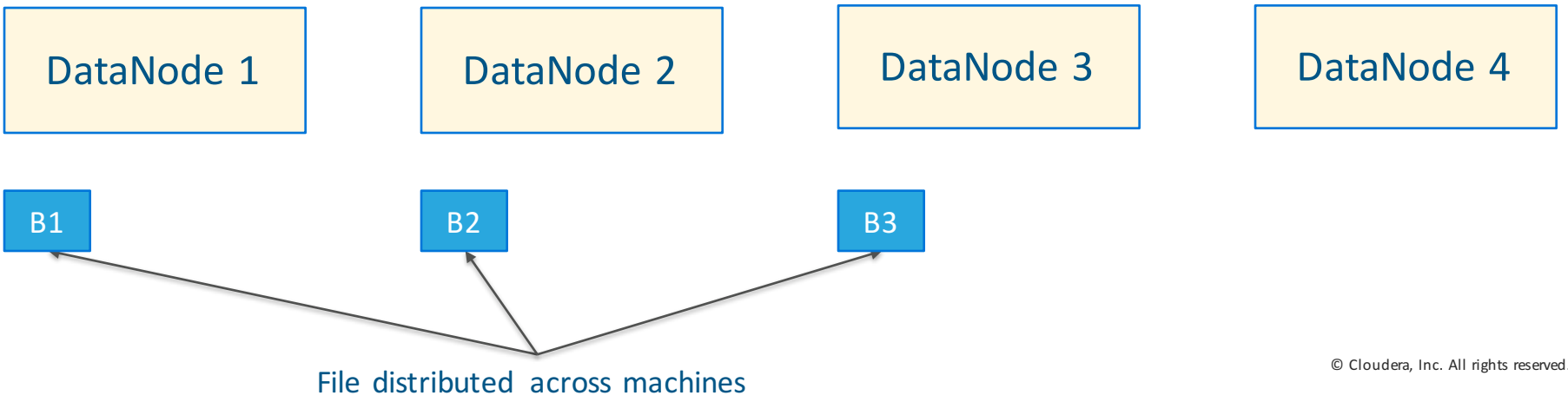
Basic Architecture of HDFS

Local file system

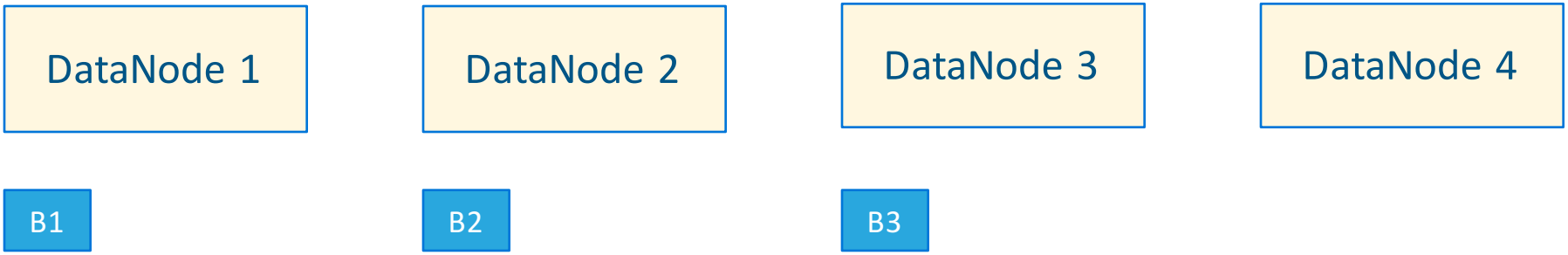
File system will split the file into blocks



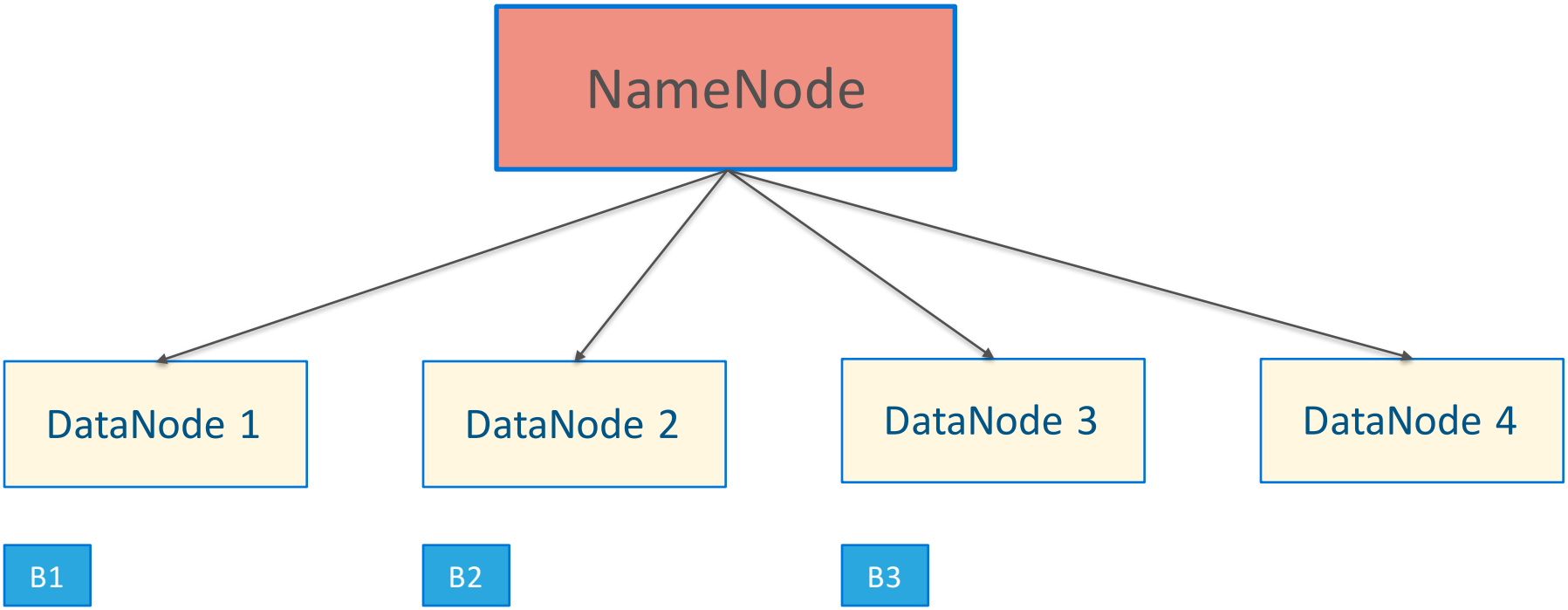
HDFS



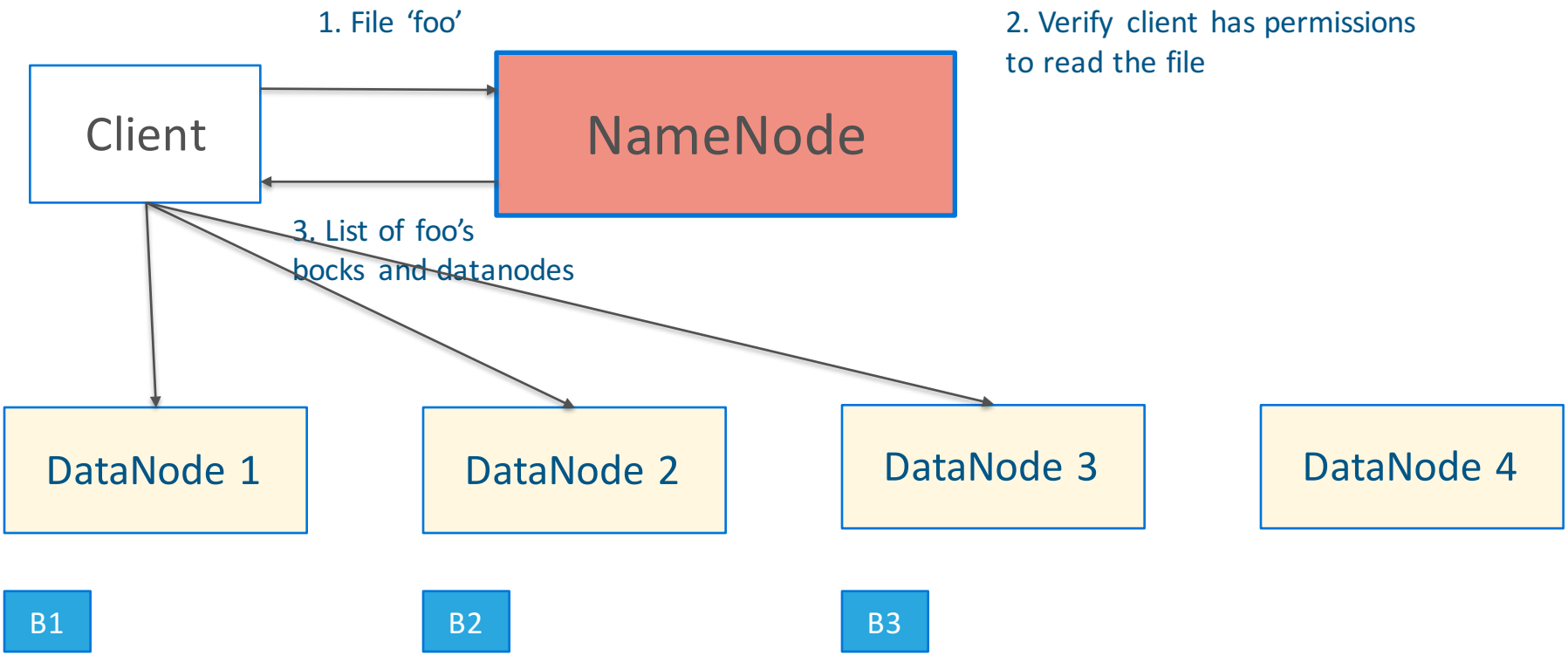
HDFS DataNode



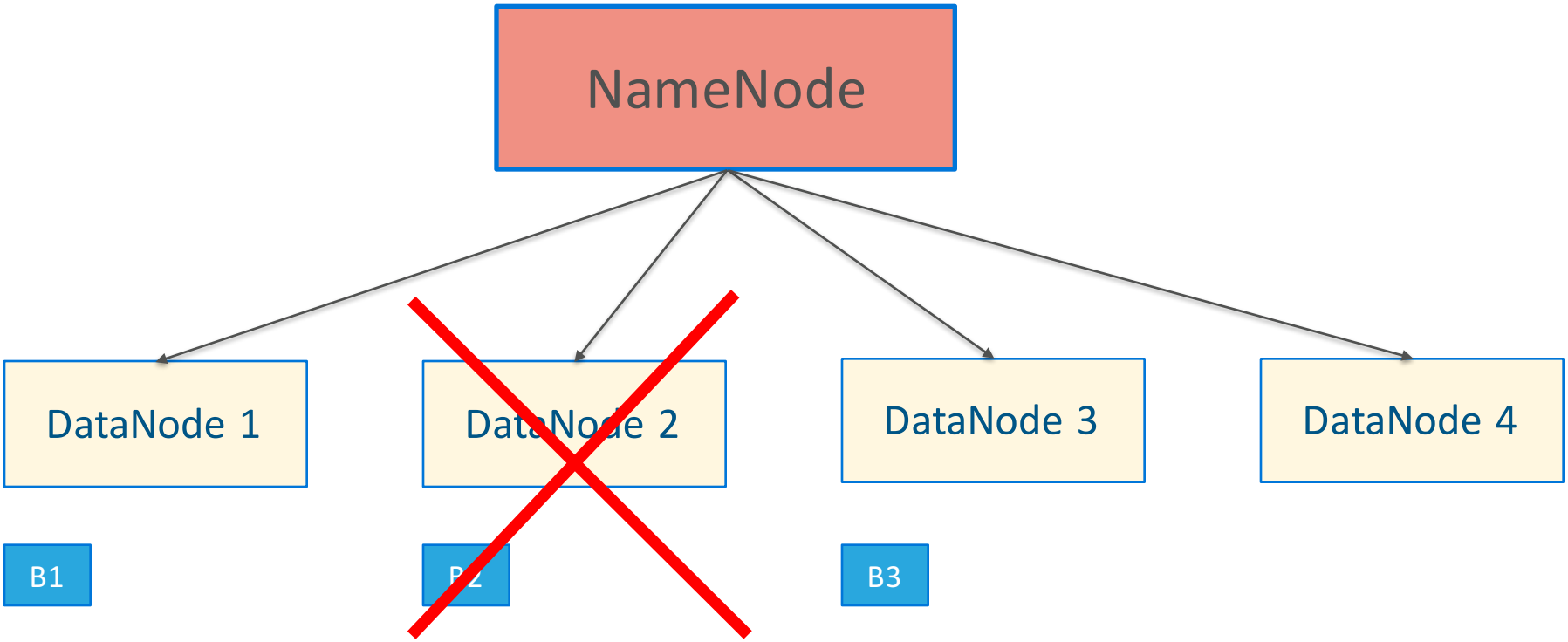
HDFS NameNode



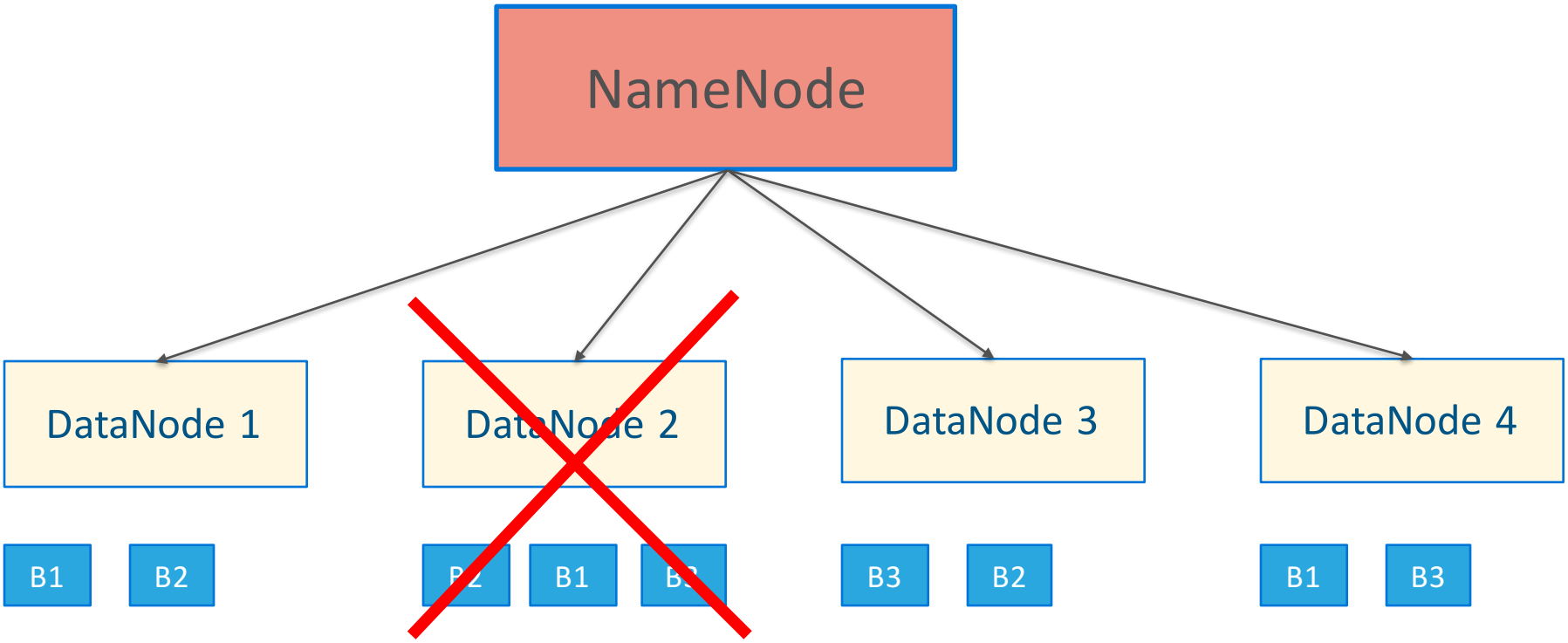
HDFS Reading a file



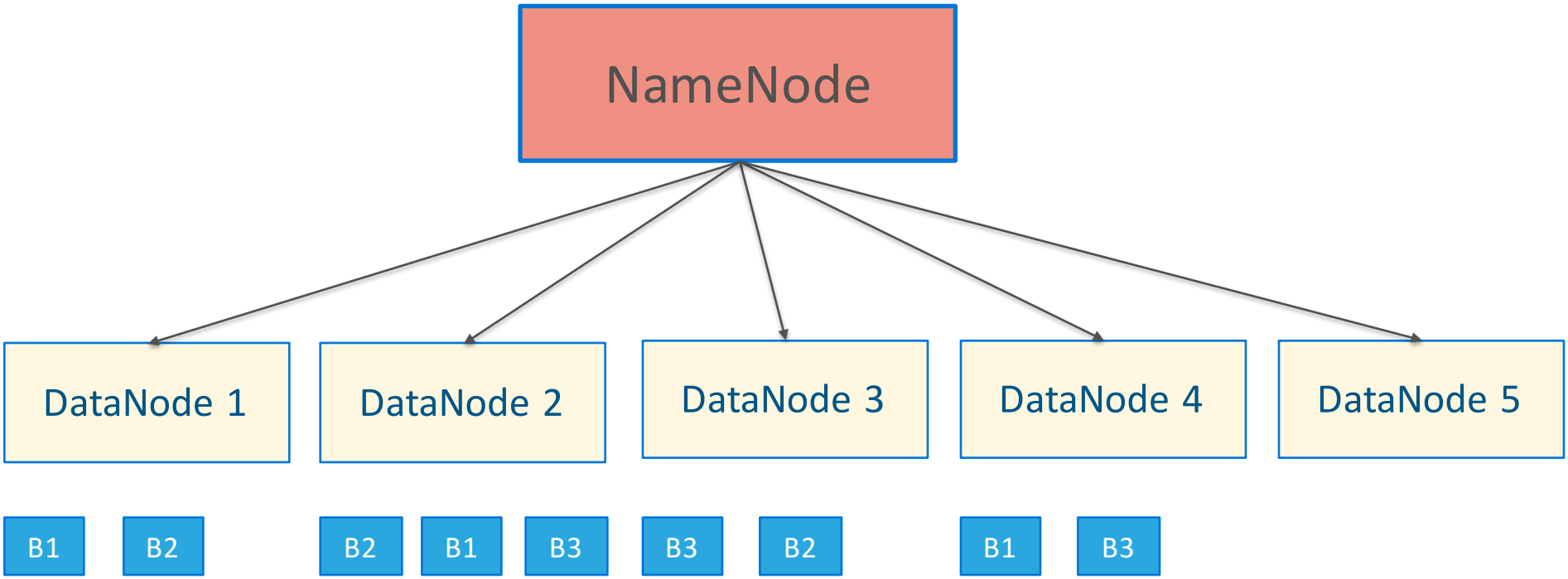
HDFS Fault tolerance



HDFS Redundancy



HDFS Horizontal Scalability



Let’s look at some existing HDFS systems…

- Yahoo! HDFS Clusters

  40k+ servers, 100k+ CPUs, 450PB data

  - Spam filtering, web indexing, personaliztion, etc.

- Facebook HDFS Cluster

  15TB new data per day

  1200+ machines, 30PB in one cluster

  - Datawarehouse, Messages, etc.

cloudera

- 1-10PB HDFS Clusters

  Production critical workloads for websites, retail, finance, telcom, government, research …

- Sub PB HDFS Clusters

  PoCs, R&D, production in early stages …

cloudera

But…. there are restrictions!

It’s not a magic wand!

cloudera

Files are **append only**

- Access Model : Write-once-read-many
- Can not change existing contents

cloudera

Not designed for small files

- HDFS block sizes are in MB (default 128MB)
  - Designed for typical GBs / TBs of file sizes
  - Normal files system have 4kb block size!
  - Name node becomes bottleneck because metadata grows too big

cloudera

Summary

HDFS is a great distributed file system!

- Store massive data
- Scalable
- High throughput
- Fault tolerance

cloudera

MapReduce

- Distributed processing framework
- Commodity machines
- Fault tolerance

cloudera

MapReduce

- HBase uses and extends MR APIs for data access

Further reading: http://hadoop.apache.org

cloudera

cloudera

HBase Architecture

Unique id	Name	price	weight	store1	store2	store3
“1000000”	snickers	$9.99	4 Oz	Yes	Yes	Yes
“3000000”	almonds	$9.99	8 Oz	Yes	No	Yes
“8000000”	coke	$9.99	16 Oz	Yes	Yes	Yes

cloudera

HBase Architecture

Unique id	Name	price	weight	store1	store2	store3
“1000000”	snickers	$9.99	4 Oz	Yes	Yes	Yes
“3000000”	almonds	$9.99	8 Oz	Yes	No	Yes
“8000000”	coke	$9.99	16 Oz	Yes	Yes	Yes
“4000000”	new	$34.63	16 Oz	No	Yes	Yes

cloudera

HBase Architecture

Unique id	Name	price	weight	store1	store2	store3
“1000000”	snickers	$9.99	4 Oz	Yes	Yes	Yes
“3000000”	almonds	$9.99	8 Oz	Yes	No	Yes
“8000000”	coke	$9.99	16 Oz	Yes	Yes	Yes
“4000000”	foo	$34.63	16 Oz	No	Yes	Yes
“5000000”	bar	$22.54	16 Oz	Yes	Yes	Yes

cloudera

HBase Architecture

Unique id	Name	price	weight	store1	store2	store3
“1000000”	snickers	$9.99	4 Oz	Yes	Yes	Yes
“3000000”	almonds	$9.99	8 Oz	Yes	No	Yes
“8000000”	coke	$9.99	16 Oz	Yes	Yes	Yes
“4000000”	foo	$34.63	16 Oz	No	Yes	Yes
“5000000”	bar	$22.54	16 Oz	Yes	Yes	Yes
“9000000”	new1	$2.5	16 Oz	Yes	Yes	Yes
“7000000”	new2	$6.4	16 Oz	Yes	Yes	Yes
“2000000”	new3	$6.4	16 Oz	Yes	Yes	Yes

cloudera

HBase Architecture | Regions

[“”, “5000000”)

Row Key	Name	brand	price	weight	store1	store2	store3
“1000000”	snickers	xxx	$9.99	4 Oz	Yes	Yes	Yes
“2000000”	new3	xxx	$6.4	16 Oz	Yes	Yes	Yes
“3000000”	almonds	xxx	$9.99	8 Oz	Yes	No	Yes
“4000000”	foo	xxx	$34.63	16 Oz	No	Yes	Yes

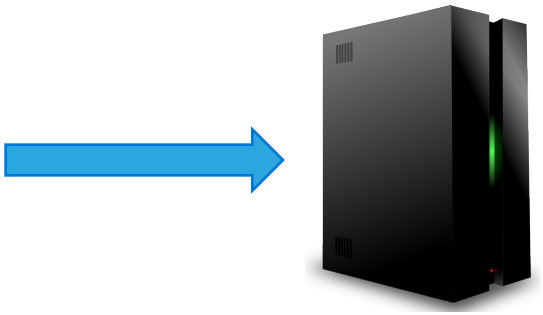
[“5000000”, “”)

Row Key	Name	brand	price	weight	store1	store2	store3
“5000000”	bar	xxx	$22.54	16 Oz	Yes	Yes	Yes
“7000000”	new2	xxx	$6.4	16 Oz	Yes	Yes	Yes
“8000000”	coke	xxx	$9.99	16 Oz	Yes	Yes	Yes
“9000000”	new1	xxx	$2.5	16 Oz	Yes	Yes	Yes

cloudera

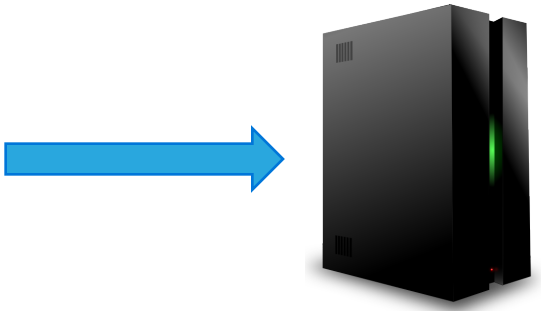
HBase Architecture | RegionServer

Row Key	Name	price	weight	….
“1000000”	snickers	$9.99	4 Oz	….
“2000000”	new3	$6.4	16 Oz	….
“3000000”	almonds	$9.99	8 Oz	….
“4000000”	foo	$34.63	16 Oz	….



Server 12

Row Key	Name	price	weight	….
“5000000”	bar	$22.54	16 Oz	….
“7000000”	new2	$6.4	16 Oz	….
“8000000”	coke	$9.99	16 Oz	….
“9000000”	new1	$2.5	16 Oz	….

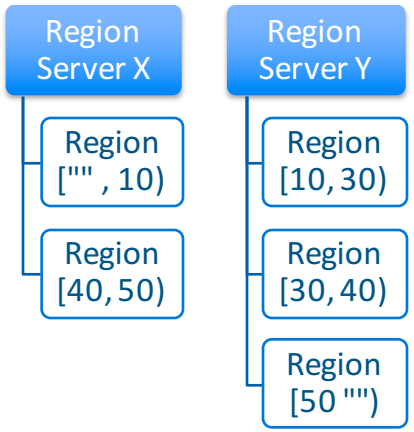


Server 7

cloudera

HBase Architecture | Regions

- **Horizontal split** of tables
- Regions are served by RegionServers
- Automatically (based on configurations). Can be done manually too.



Relationship:

- A Region is only served by a single RegionServer at a time.
- A RegionServer can serve multiple Regions.

cloudera

HBase Architecture

Row Key	Name	price	weight	store1	store2	store3
“1000000”	snickers	$9.99	4 Oz	Yes	Yes	Yes
“2000000”	new3	$6.4	16 Oz	Yes	Yes	Yes
“3000000”	almonds	$9.99	8 Oz	Yes	No	Yes
“4000000”	foo	$34.63	16 Oz	No	Yes	Yes
“5000000”	bar	$22.54	16 Oz	Yes	Yes	Yes
“7000000”	new2	$6.4	16 Oz	Yes	Yes	Yes
“8000000”	coke	$9.99	16 Oz	Yes	Yes	Yes
“9000000”	new1	$2.5	16 Oz	Yes	Yes	Yes

What if you have many more columns?

cloudera

HBase Architecture | Column family

Row Key	info: name	info: price	info: weight	availability: store1	availability: store2	availability: store3
“1000000”	snickers	$9.99	4 Oz	Yes	Yes	Yes
“2000000”	new3	$6.4	16 Oz	Yes	Yes	Yes
“3000000”	almonds	$9.99	8 Oz	Yes	No	Yes
“4000000”	foo	$34.63	16 Oz	No	Yes	Yes
“5000000”	bar	$22.54	16 Oz	Yes	Yes	Yes
“7000000”	new2	$6.4	16 Oz	Yes	Yes	Yes
“8000000”	coke	$9.99	16 Oz	Yes	Yes	Yes
“9000000”	new1	$2.5	16 Oz	Yes	Yes	Yes

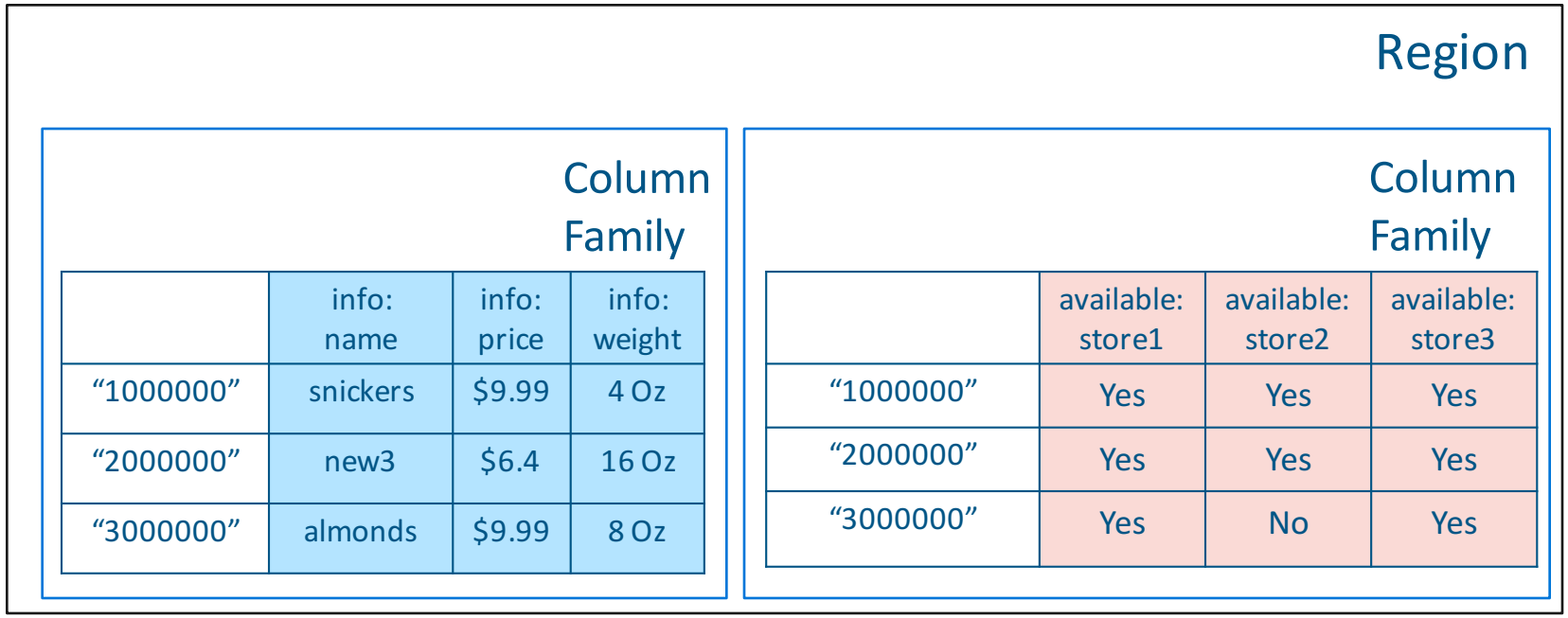
cloudera

HBase Architecture | Column family

- Set of columns that have similar access patterns
- Data stored in separate files
- Tune performance
  - In-memory
  - Compression
  - Version retention policies
  - Cache priority
- Needs to be specified by the user

cloudera

HBase Architecture



cloudera

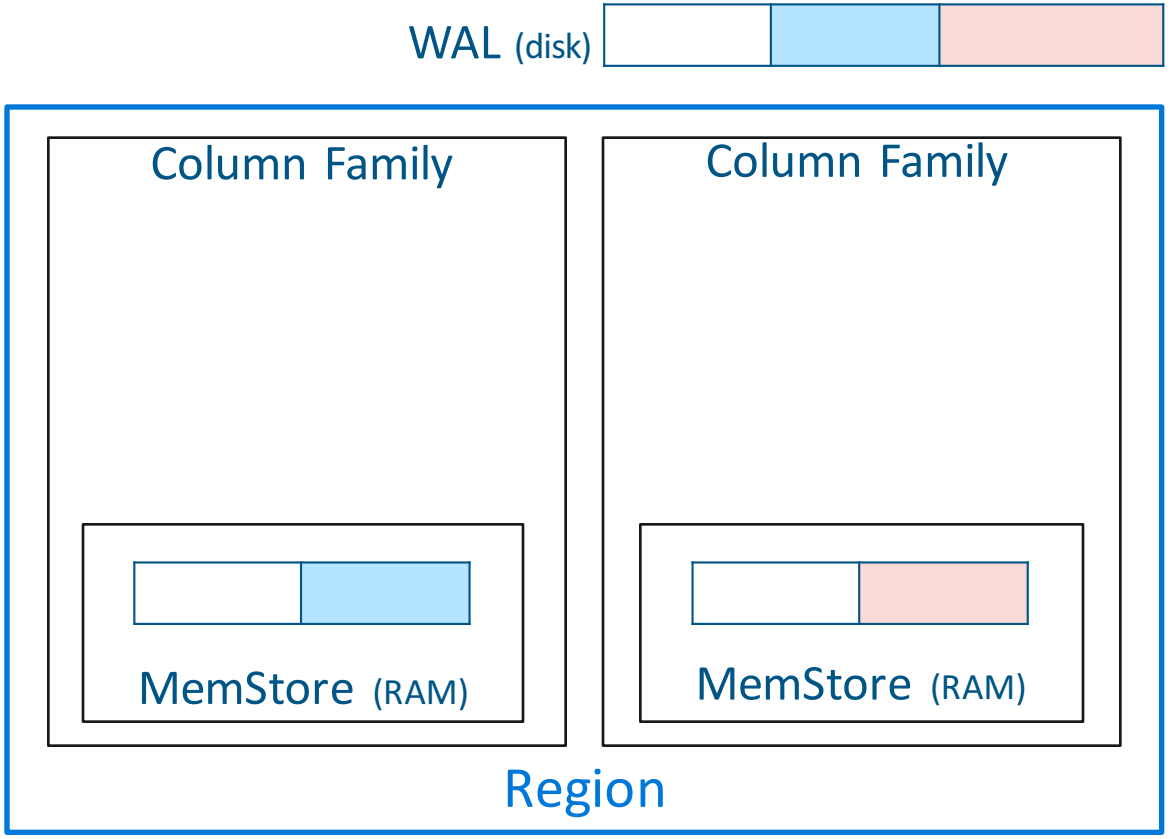
HBase Architecture | Put

1. Client creates a row to put.
2. Client checks which RegionServer hosts this row
3. Row is written into write-ahead log (WAL) on disk for persistence
4. Row is written to MemStore (in memory storage)

cloudera

HBase Architecture | Put

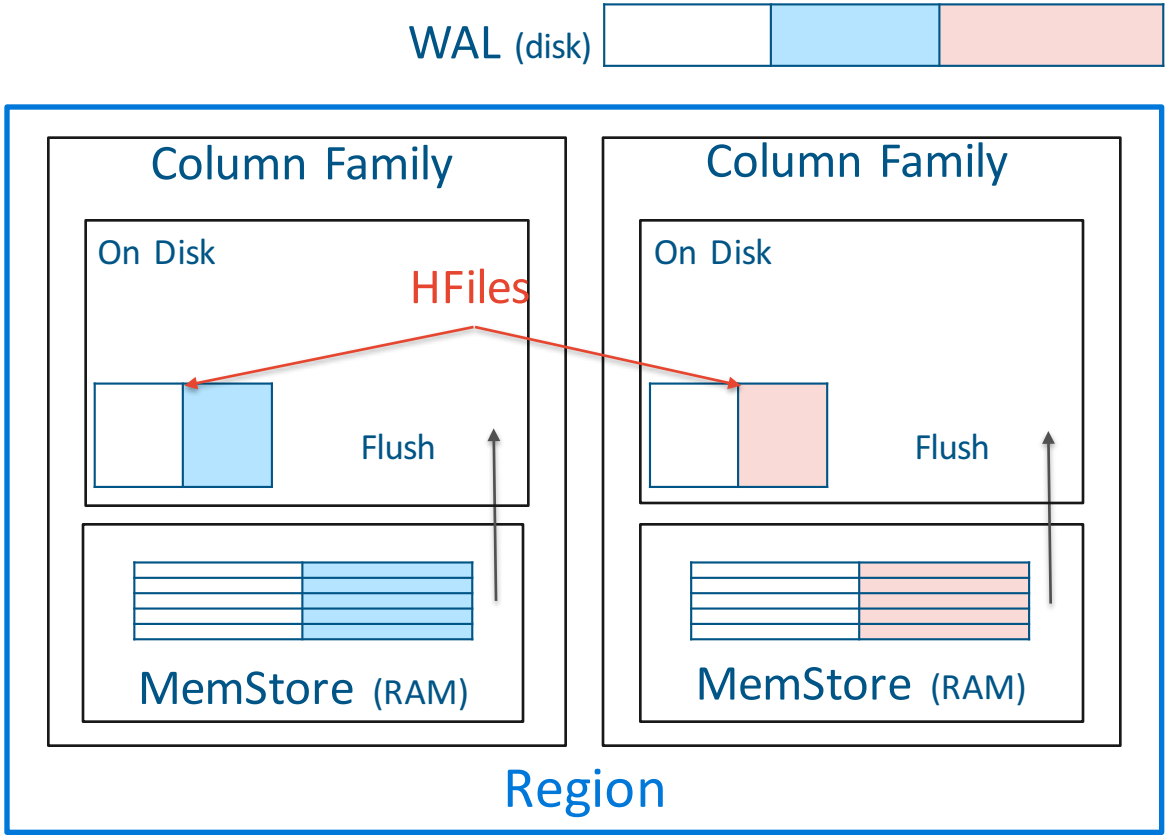
Put



cloudera

HBase Architecture | HFile

Put



RegionServer 2

cloudera

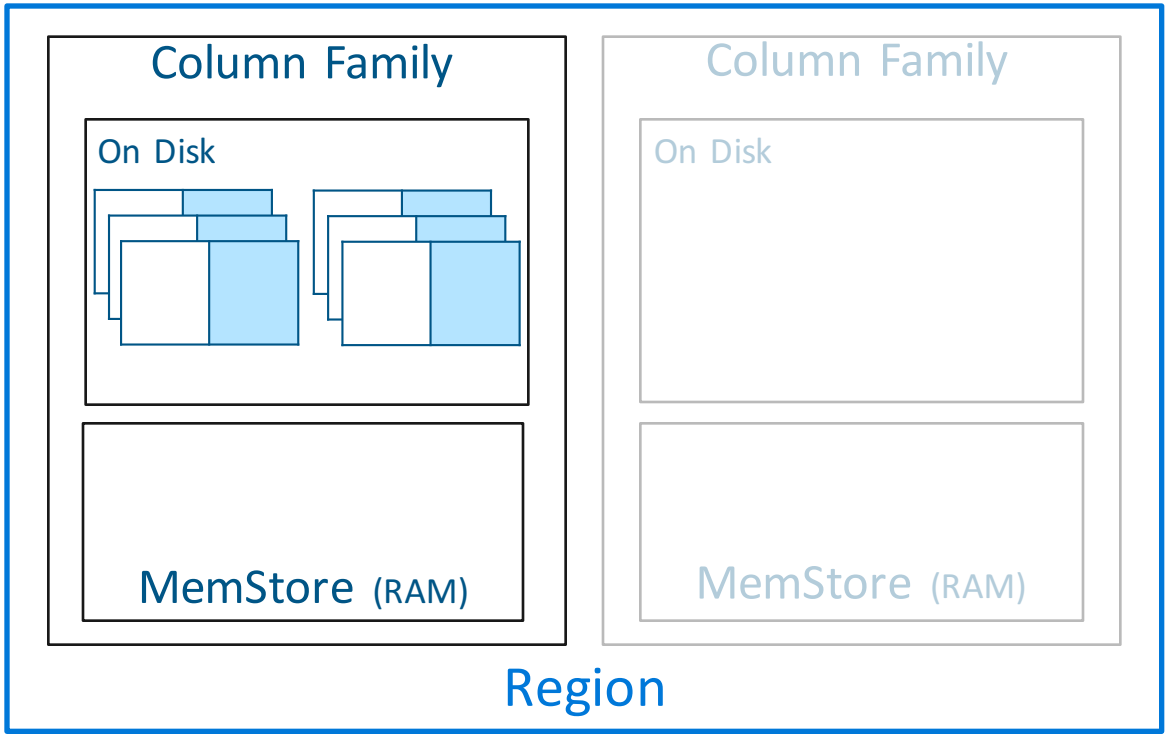
HBase Architecture | HFile

- As more rows are added to the table
  - Write to WAL
  - Write to MemStore
  - When MemStore gets full, its contents are flushed to HDFS as Hfiles

cloudera

HBase Architecture | HFile

- Side effect of more HFiles in a Region
  - More disk seeks on read
  - More disk usage (overwritten value, deletes, etc)



RegionServer 2

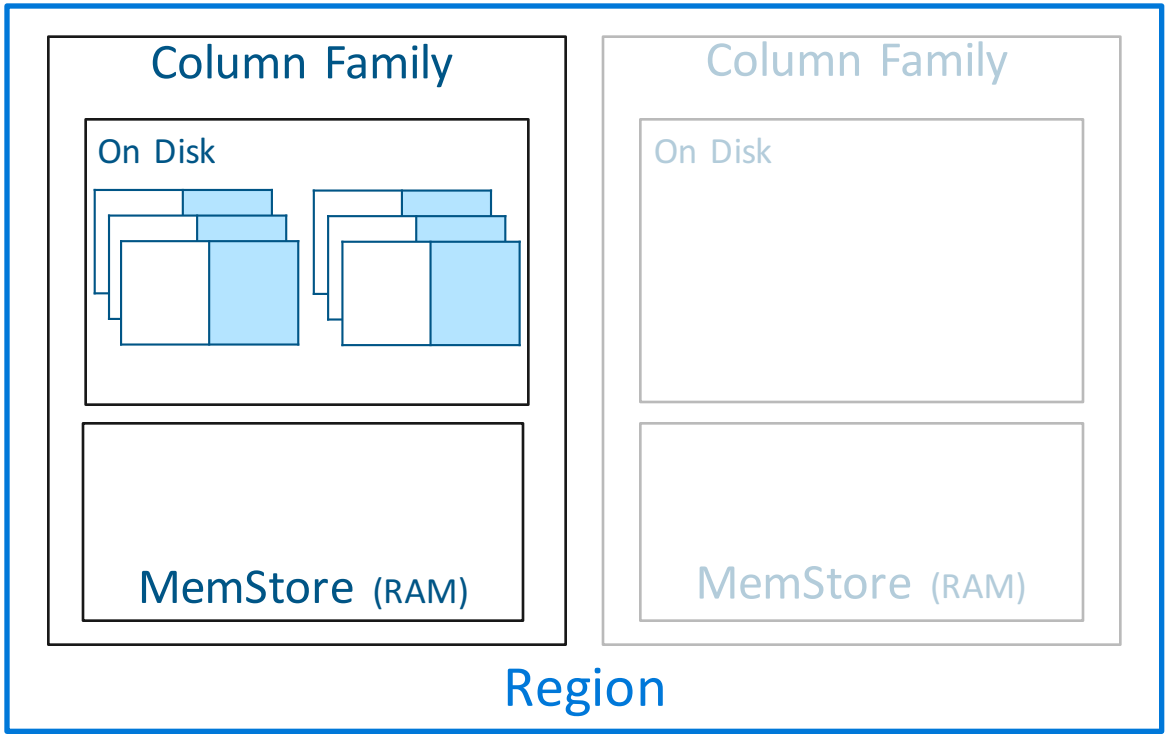
HBase Architecture | Compactions

Merging multiple Hfiles into a single Hfile.

- Minor compactions
- Major compactions

cloudera

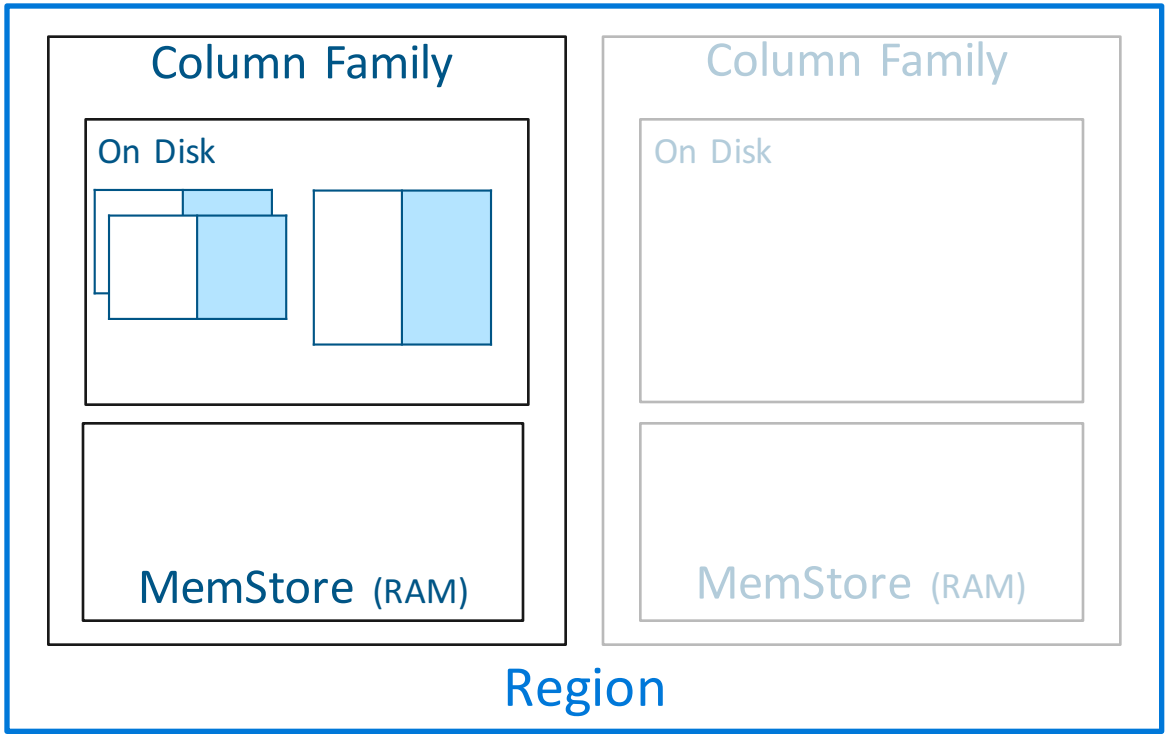
HBase Architecture



RegionServer 2

cloudera

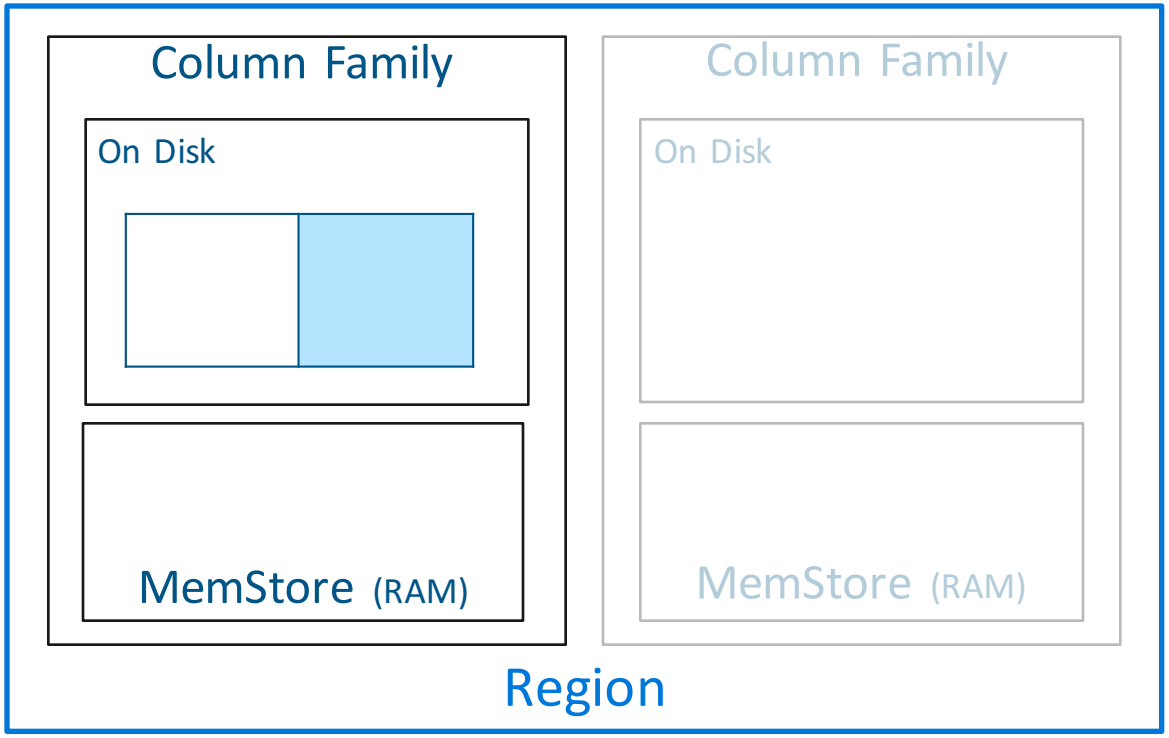
HBase Architecture



RegionServer 2

cloudera

HBase Architecture



RegionServer 2

HBase Architecture | Compactions

- Minor compactions
  - Can be finely tuned using multiple configurations
  - Controlled by policy (pluggable)
- Major compactions
  - Periodic (set from configuration) or manually triggered.
  - Tend to be run during off-peak times.

cloudera

HBase Architecture | Splits

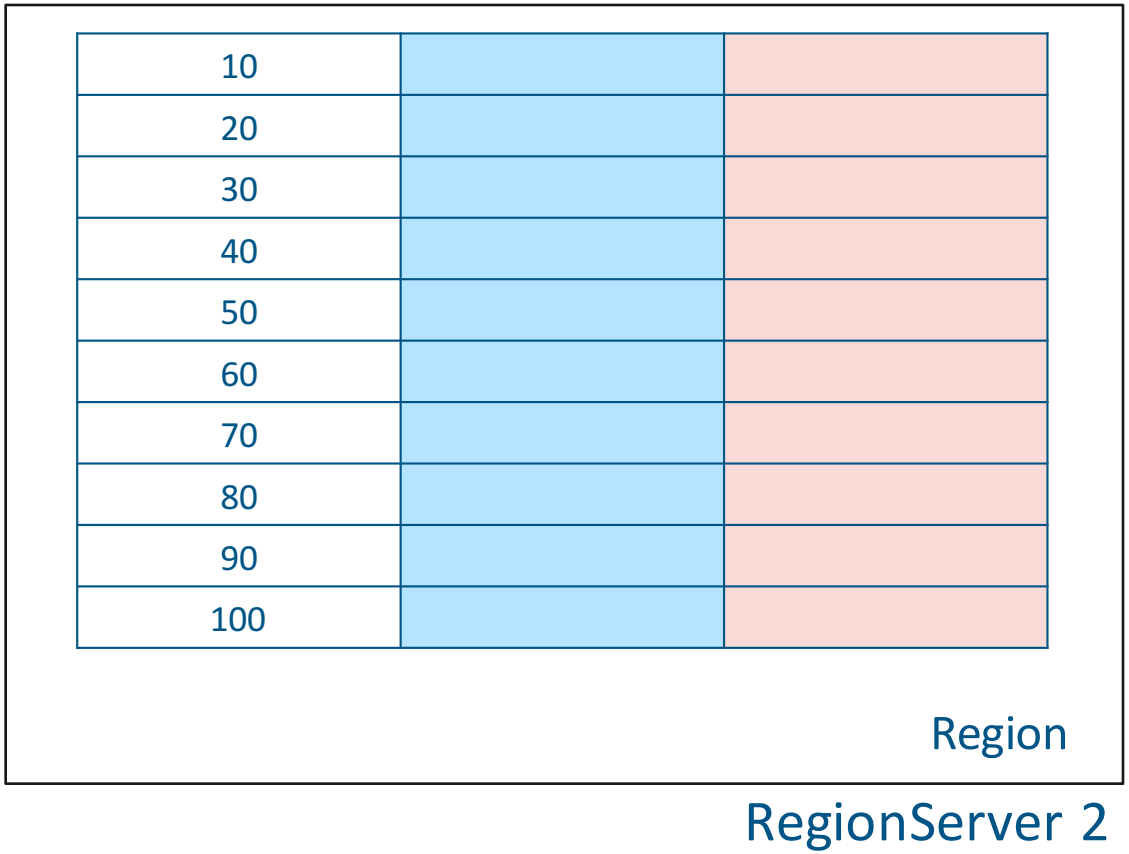
- Eventually, Regions become imbalanced.
- HBase can split a Region into two.
- Daughter Regions are can be moved to a different RegionServer, if necessary.

Gives HBase horizontal scalability!

cloudera

HBase Architecture | Splits

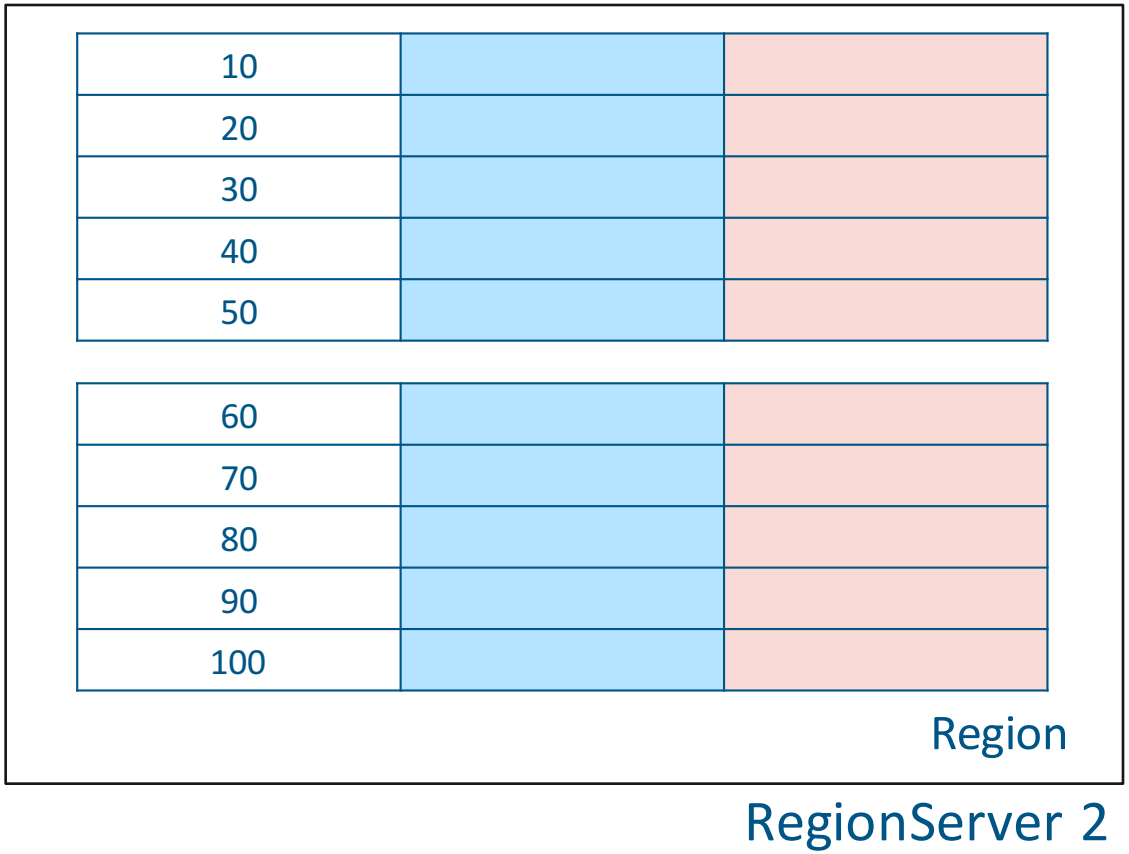
- Region has rows [0, 100)
- Splitting
  - Middle of current Region's row range
  - All column families are split



cloudera

HBase Architecture | Splits

- Splits
  - [0, 50)
  - [50, 100)



cloudera

HBase Architecture | Bulk Import

- This write path is durable, but if you're importing a lot of data
  - Every put goes into WAL, which means disk seeks. Lots of puts mean lots of disk seeks.
  - Lots of data into MemStores means lots of flushing to disk.
  - Lots of flushing to disk might mean lots of compactions.

So we have other better options like….

cloudera

HBase Architecture | Bulk Loading

- Bypass conventional write path
  - Extract data from source
  - Transform data into HFiles (done with MapReduce job) directly
  - Tell RegionServers to serve these HFiles

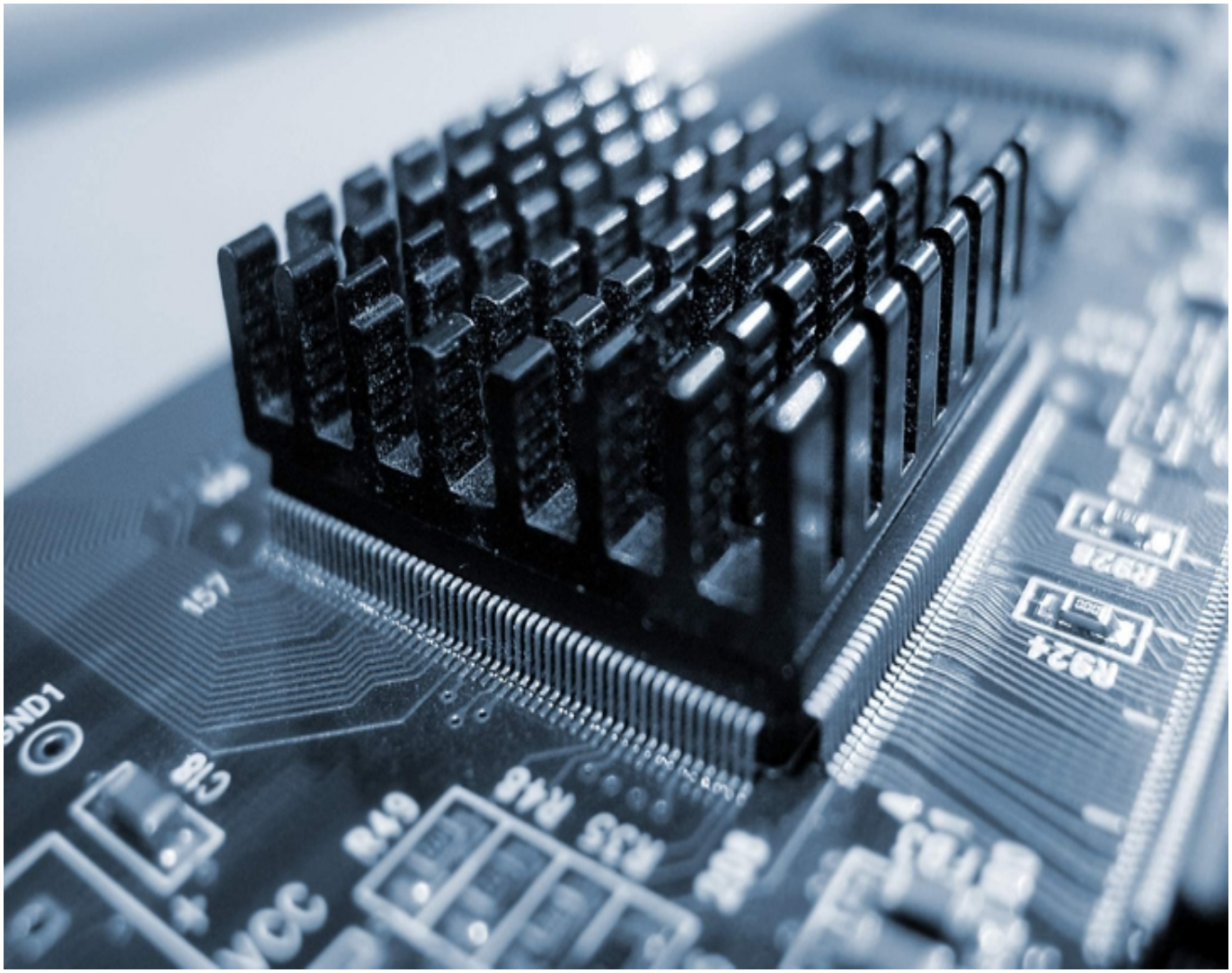
cloudera

HBase Architecture | APIs

- Java API
  - Most full-featured.
- REST API
  - Easily accessible.
- Thrift API
  - Support for many languages (e.g. C, C++, Perl, Ruby, Python).

cloudera

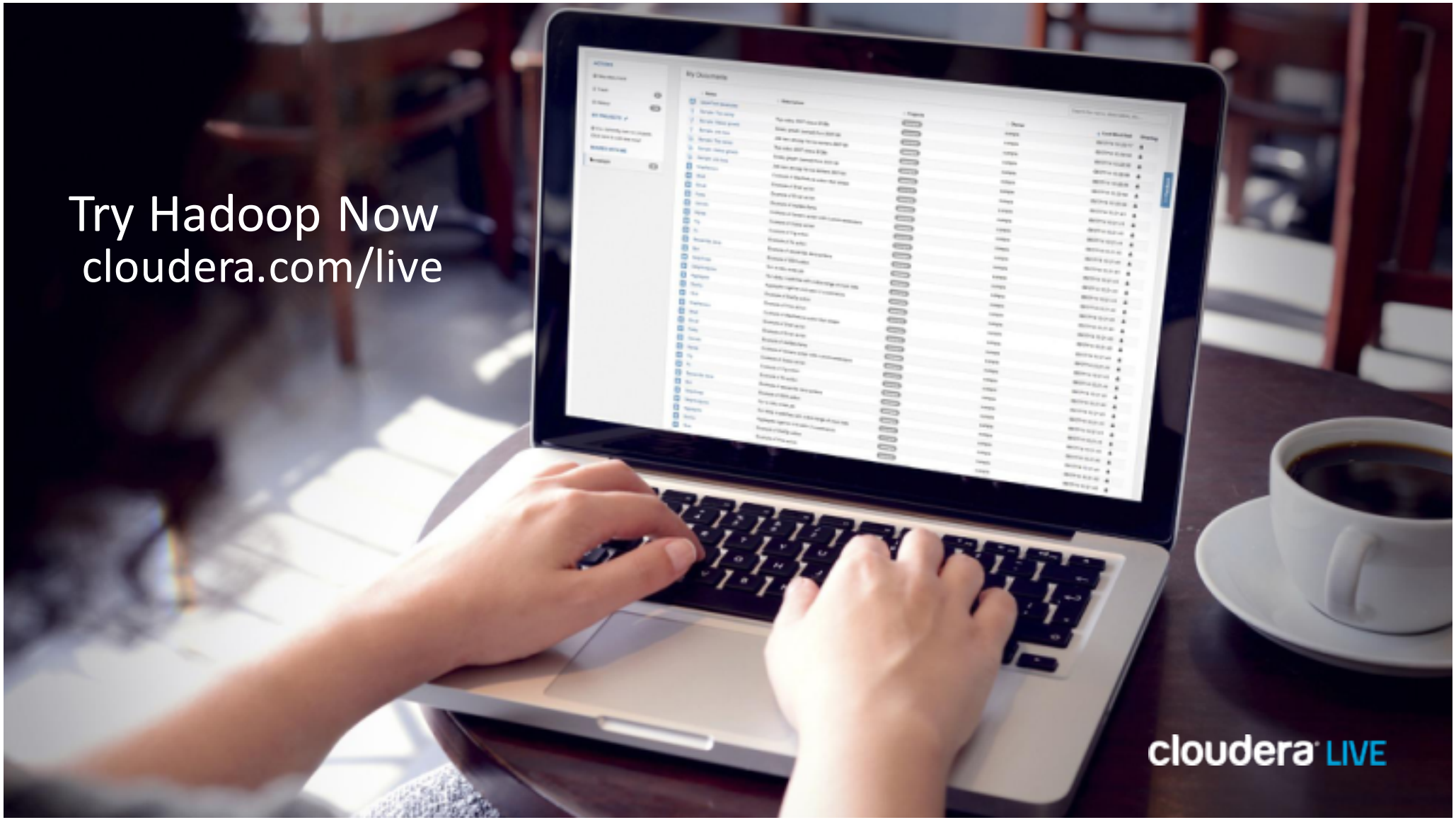
Enough of Architecture

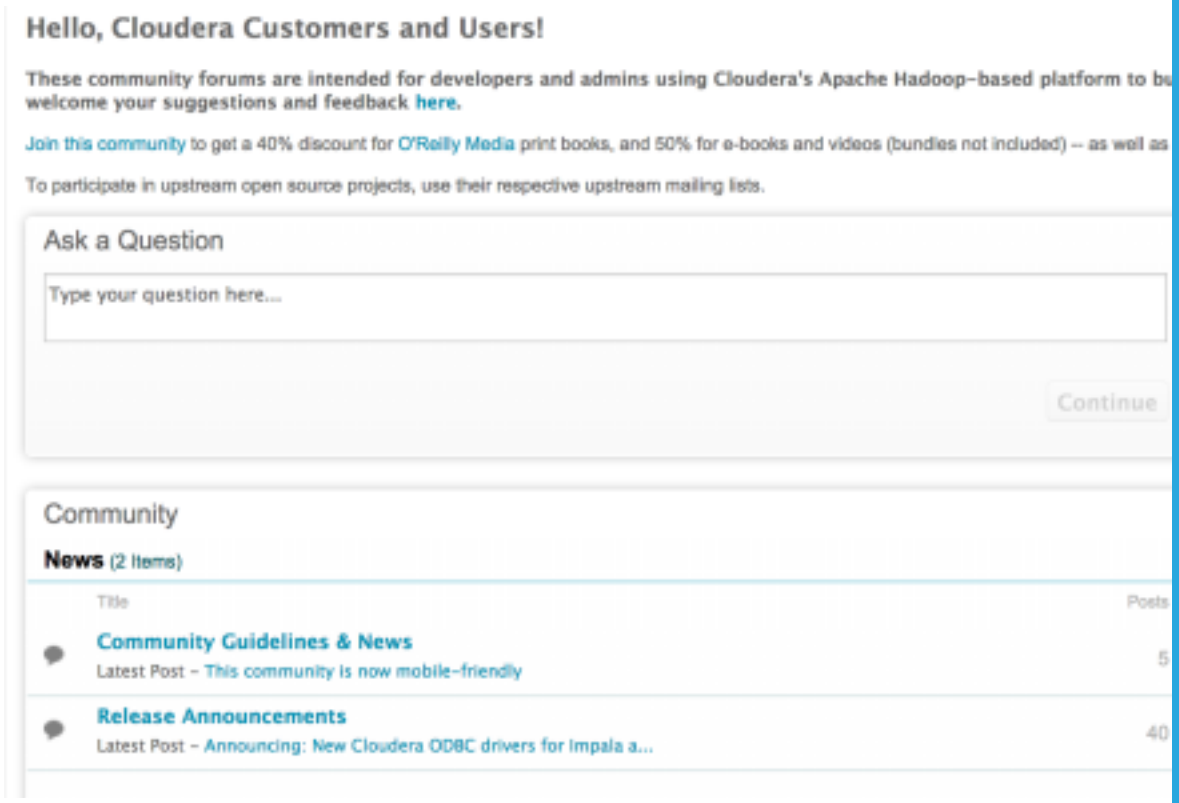


cloudera

That’s all folks!

cloudera


Try Hadoop Now
cloudera.com/live
cloudera LIVE



Get community help or provide feedback

cloudera.com/community

Sources

- A Survey of HBase Application Archetypes
  - Lars George, Jon Hsieh
  - http://www.slideshare.net/HBaseCon/case-studies-session-7
- Hadoop and HBase: Motivations, Use cases and Trade-offs
  - Jon Hsieh's slides

cloudera

cloudera

Questions ?

