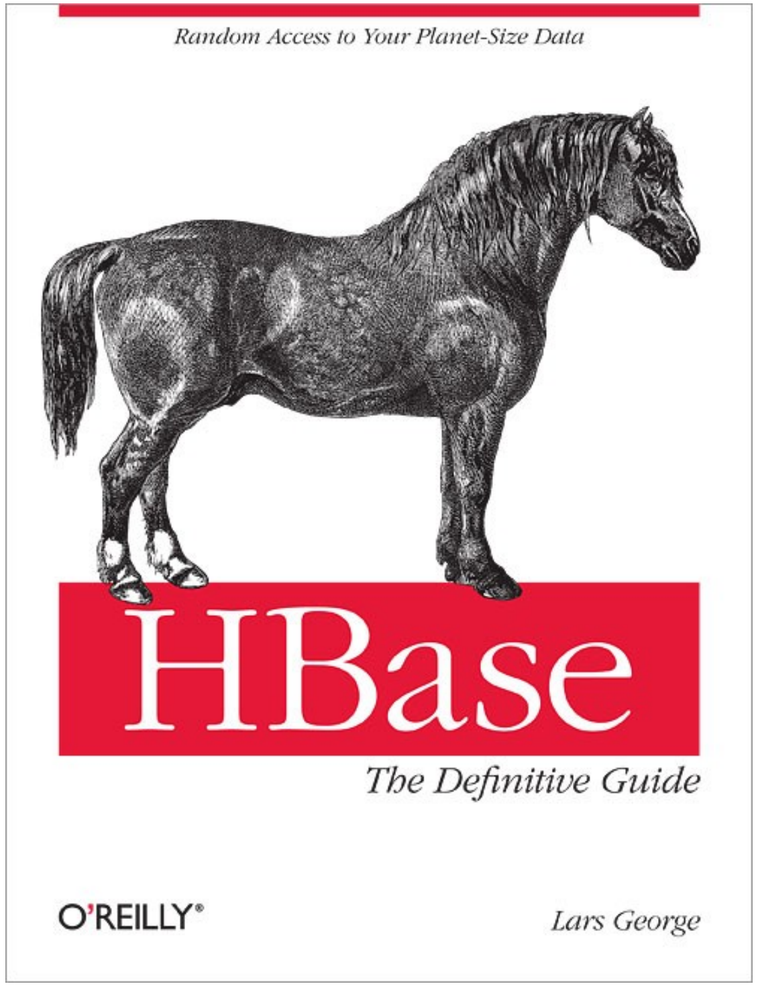
DLI Accelerated Data Science Teaching Kit

Lecture 12.1 - HBase Overview

Lesson Based on



http://shop.oreilly.com/product/0636920014348.do

APACHE HBASE

http://hbase.apache.org

Built on top of HDFS

Supports real-time read/write random access

Scale to very large datasets, many machines

Not relational, does NOT support SQL
("NoSQL" = "not only SQL") http://en.wikipedia.org/wiki/NoSQL

Supports billions of rows, millions of columns
(e.g., serving Facebook's Messaging Platform)

Written in Java; works with other APIs/languages (REST, Thrift, Scala)

http://radar.oreilly.com/2014/04/5-fun-facts-about-hbase-that-you-didnt-know.html
http://hbase.apache.org/poweredbyhbase.html

HBase is Column-Oriented

Data is stored by columns physically (instead of by rows)

- A row conceptually consists on many columns (millions!)

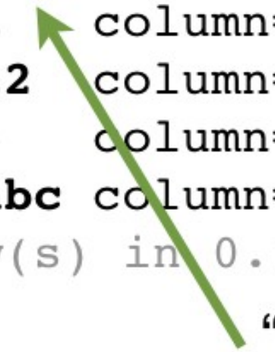
Rows form a table

- Row key uniquely locates a row, like an “index”, only one “index” per table
  - No built-in support for multiple indices; possible via extensions
- Rows sorted by row key lexicographically (~ alphabetically)

Rows Sorted Lexicographically

(=alphabetically)

```
hbase(main):001:0> scan 'table1'
ROW       COLUMN+CELL
row-1     column=cf1:, timestamp=1297073325971 ...
row-10    column=cf1:, timestamp=1297073337383 ...
row-11    column=cf1:, timestamp=1297073340493 ...
row-2     column=cf1:, timestamp=1297073329851 ...
row-22    column=cf1:, timestamp=1297073344482 ...
row-3     column=cf1:, timestamp=1297073333504 ...
row-abc   column=cf1:, timestamp=1297073349875 ...
7 row(s) in 0.1100 seconds
```



"**row-10**" comes before "**row-2**".
How to fix?

Pad "row-2" with a "0".
i.e., "row-02"

Columns grouped into column families

- Why?
  - Helps with organization, understanding, optimization, etc.
- In details...
  - Columns in the same family stored in same *file* called *HFile*
  - Apply compression on the whole family

More on column family, column

Column family

- An HBase table supports only **few** families (e.g., <10)
  - Due to limitations in implementation
- Family name must be **printable**
- Should be defined when table is created
  - Should not be changed often

Each **column** referenced as "**family:qualifier**"

- Can have **millions** of columns

Cell Value

Timestamped

- Implicitly by system
- Or set explicitly by user

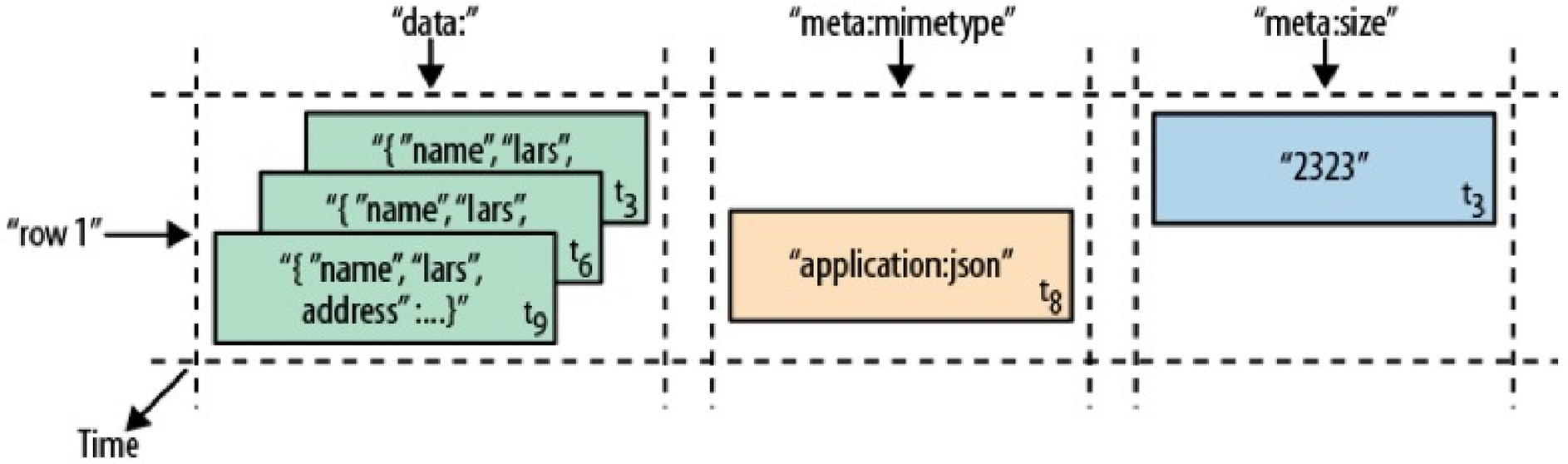
Let you store **multiple versions** of a value

- = values over time

Values stored in **decreasing** time order

- **Most recent value** can be read first

Time-oriented view of a row



Row Key	Time Stamp	Column "data:"	Column "meta:" "mimetype"	Column "meta:" "size"
"row1"	t_3	"{ "name" : "lars", "address" : ...}"		"2323"
	t_6	"{ "name" : "lars", "address" : ...}"		
	t_8		"application/json"	
	t_9	"{ "name" : "lars", "address" : ...}"		

An Exercise

How would you use HBase to create a webtable to store snapshots of every webpage on the planet, over time?

Row key	Time stamp	html	javascript	image1	image2	...
"com.cnn.www"	t8	<html>…	<javascript>…			...
						...
						...
						...

An Exercise

How would you use HBase to create a webtable to store snapshots of every webpage on the planet, over time?

DLI Accelerated Data Science Teaching Kit

Thank You