HBase: column-oriented database

Software Languages Team
University of Koblenz-Landau
Ralf Lämmel and Andrei Varanovich

Motivation

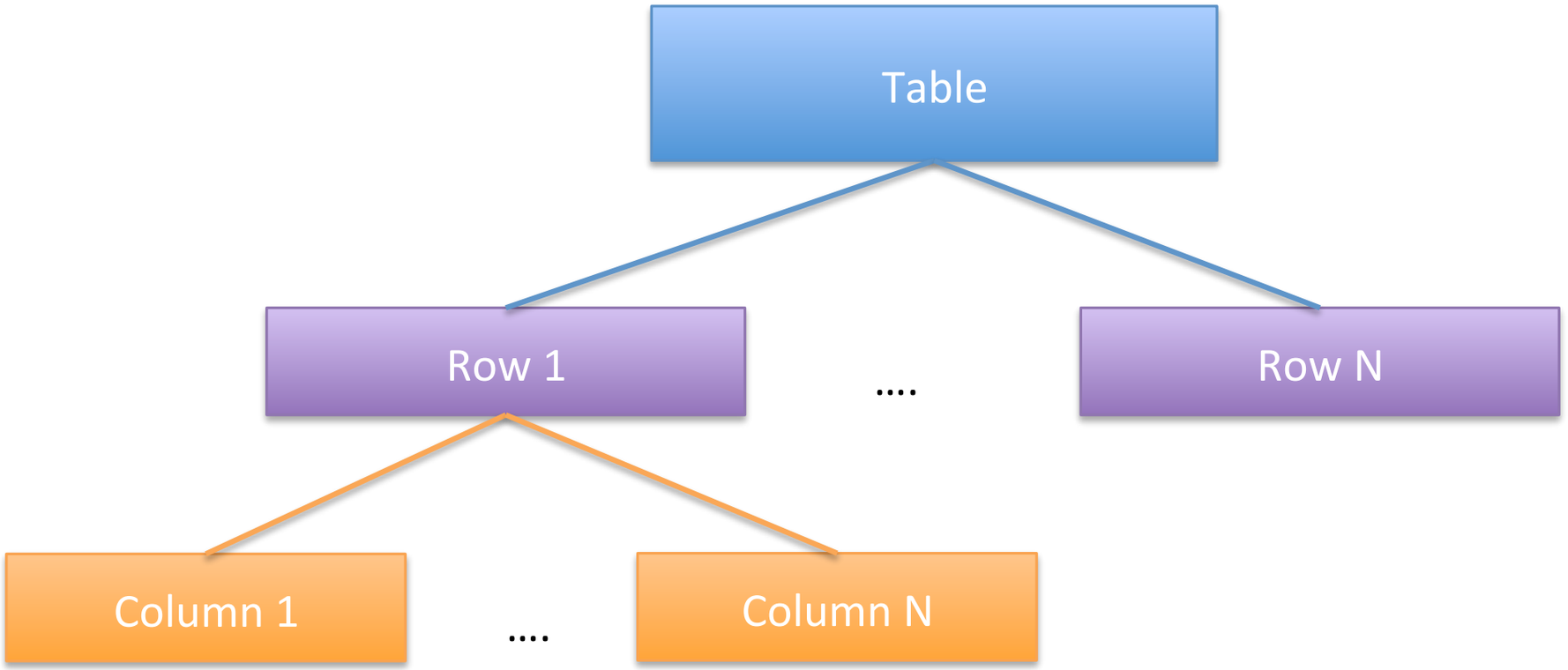
Billions of rows X millions of columns X thousands of versions = terabytes or petabytes of storage

Random, realtime read/write access

Use cases

- Messaging systems (e.g., GMail, Facebook)
- (Real-time) Data analytics (e.g., stock indicators)

Key concepts



Every row has a key

Columns can be accessed individually

Every column value (cell) has a timestamp

Cells

```
Cell = {row, column, version}
```

We get another dimension (time) for free

Cell content - uninterpreted bytes: the application logic handles types

Column family

- Columns are grouped into column families.
- All column members of a column family have the same prefix. E.g. the columns courses:history and courses:math are both members of the courses column family.

Enables: compression, in-memory storage

All columns in a column family are stored together in the same low- level storage file, called an HFile

Basic operation: create

table

column family

```
create 'employees', 'basic'
```

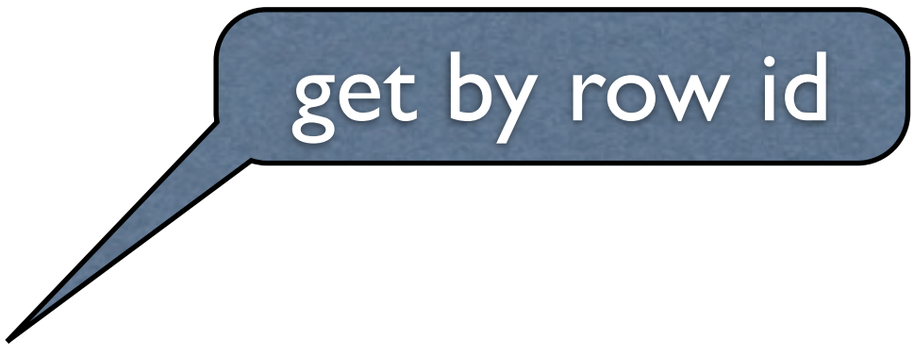
```
put 'employees', 'andrei', 'basic:salary', '12345'
```

row key

column

value

Basic operation: read



get 'employees', 'andrei'

Retunrs:

COLUMN	CELL
basic:salary	timestamp=1347551087754, value=12345

Basic operation: update

put 'employees', 'andrei', 'basic:salary', '123456'

get 'employees', 'andrei'

Returns:

COLUMN	CELL
basic:salary	timestamp=1347551251790, value=123456

Get history

per column:

get 'employees', 'andrei', **{COLUMN => 'basic:salary', VERSIONS => 3}**

Returns:

```
COLUMN                        CELL
 basic:salary                 timestamp=1347551251790, value=123456
 basic:salary                 timestamp=1347551087754, value=12345
```

Basic operation: delete

a single column:

```
delete 'employees', 'andrei', 'basic:salary'
```

complete row:

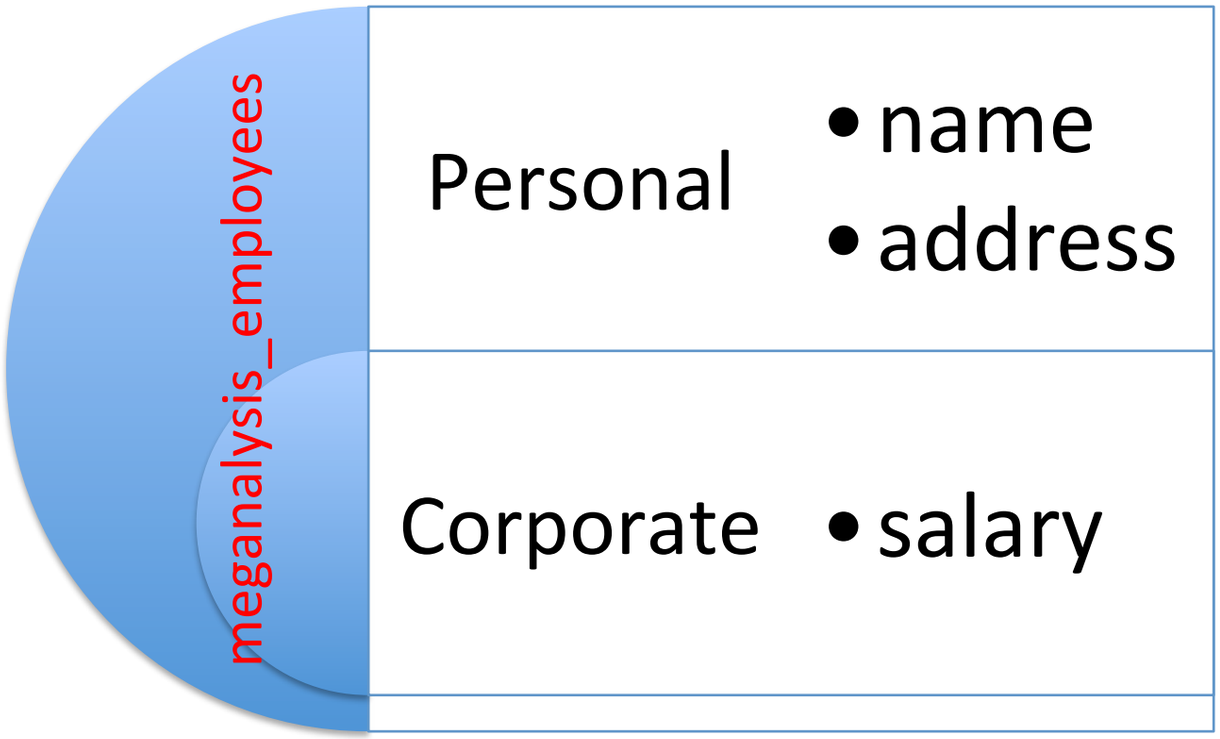
```
deleteall 'employees', 'andrei'
```

DEMO

101companies:hbase

Data schema: employees

Table: meganalysis_employees



Two column families: personal and corporate

- personal: name and address
- corporate: salary

Actual data

```
hbase(main):001:0> scan 'meganalysis_employees'
ROW                                          COLUMN+CELL
 Craig                                       column=corporate:salary,
timestamp=1347568664910, value=@\xFE$\x00\x00\x00\x00\x00
 Craig                                       column=personal:address,
timestamp=1347568664910, value=Redmond
 Craig                                       column=personal:name,
timestamp=1347568664910, value=Craig
 Erik                                        column=corporate:salary,
timestamp=1347568664919, value=@\xC8\x1C\x80\x00\x00\x00\x00
 Erik                                        column=personal:address,
timestamp=1347568664919, value=Utrecht
```

Cut

JRuby:

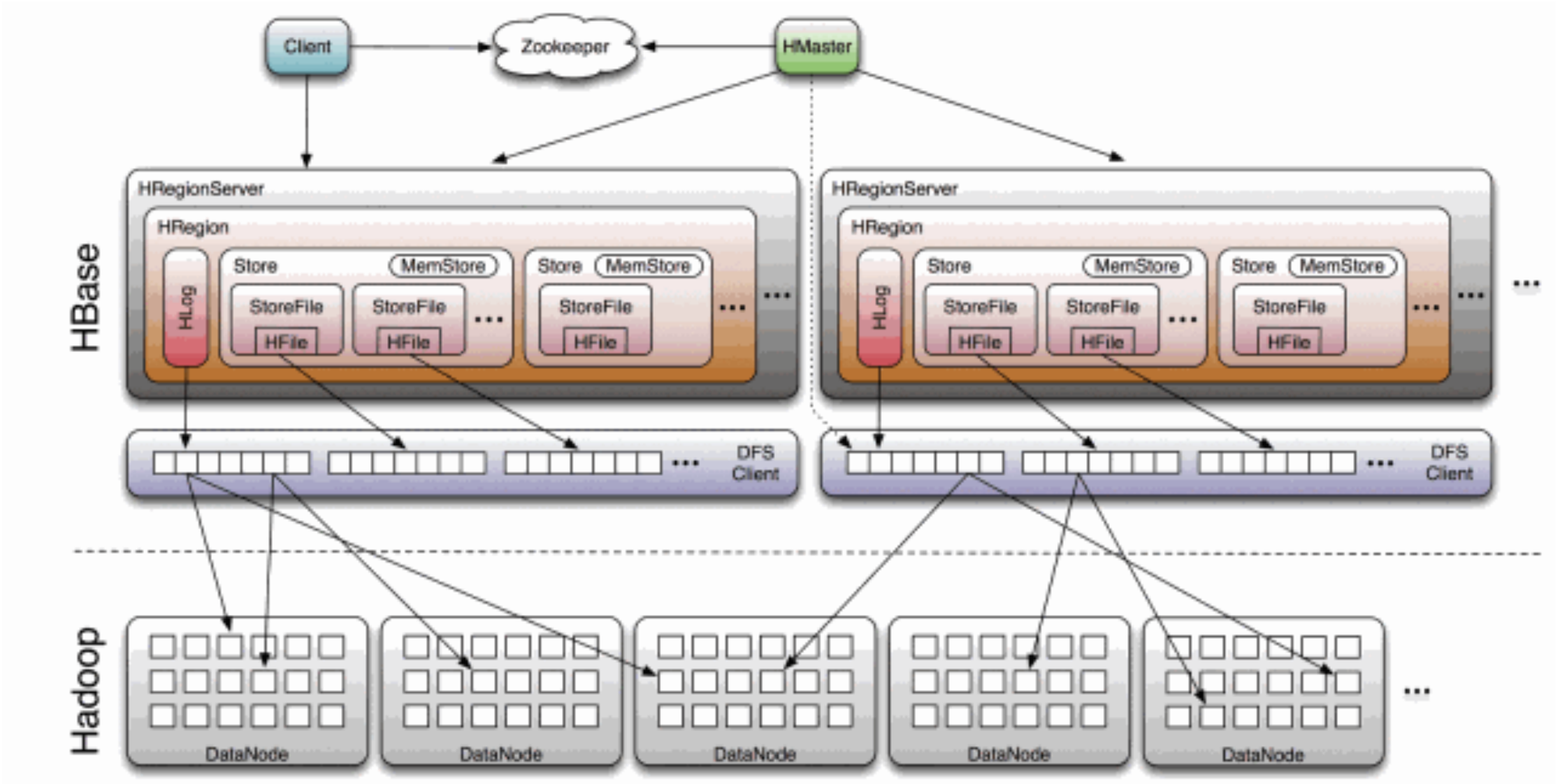
```
  employees = HTable.new(@hbase.configuration, "meganalysis_employees")
  scanner = employees.getScanner(Scan.new)

while (result = scanner.next())
  salary = Bytes.toDouble(result.getValue( *jbytes('corporate', 'salary')), 0)

  put_cut = Put.new(namebytes)
  put_cut.add( *(jbytes('corporate', 'salary') << Bytes.toBytes(salary / 2.0)))
  employees.put(put_cut) if put_cut
end
```

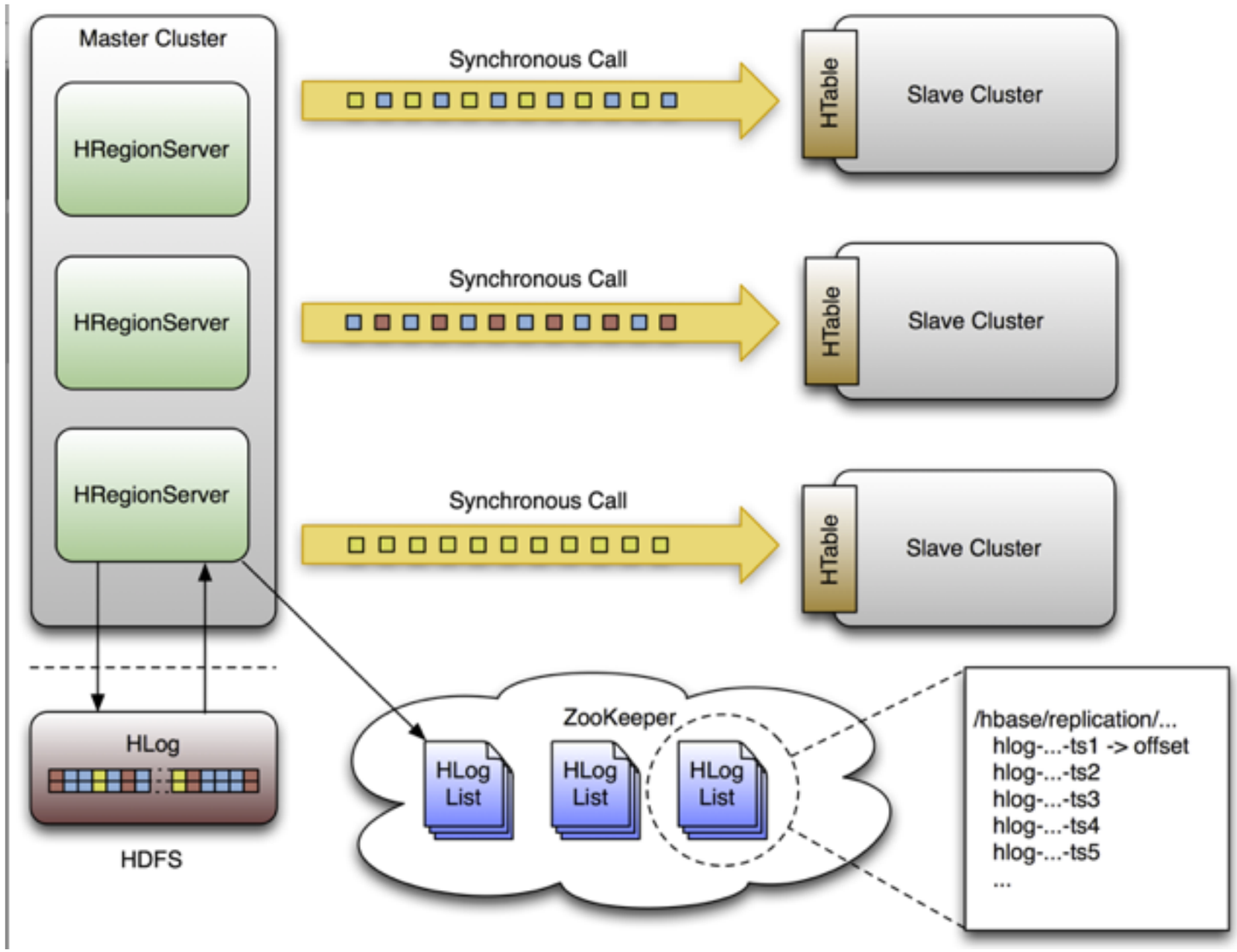
iterate over all rows

Architecture



http://lunarium.info/arc/images/Hbase.gif

Synchronization



http://hbase.apache.org/images/replication_overview.png

Summary

You learned about ...

- basic principles of column-oriented databases,
- performing CRUD operations in HBase,
- internals of HBase.

Resources

- Storage Infrastructure Behind Facebook Messages. Using HBase at Scale:

  http://sites.computer.org/debull/A12june/facebook.pdf

- HBase reference guide:

  http://hbase.apache.org/book.html