Open APIs for Open Minds

Big Data introduction

Francisco Romero Bueno

Technological Specialist. FIWARE data engineer

francisco.romerobueno@telefonica.com

FIWARE

Session outline:

- **Big Data**: What is it and how much data is there?
- **Two (three) approaches for dealing with Big Data**:

Batch and stream processing (and Lambda architectures)

- **Batch processing - Distributed storage**: The Hadoop reference (HDFS)
- **Batch processing - Distributed computing**: The Hadoop reference (MapReduce)
- **Batch processing – Simplifying the analysis**: Querying tools
- **Stream processing**: The Storm reference

Big Data:

What is it and how much data is there

What is big data?

What is big data?

http://commons.wikimedia.org/wiki/File:Interior_view_of_Stockholm_Public_Library.jpg

Not a matter of thresholds

*If both the data used by your app and the processing capabilities your app logic needs fit the available infrastructure, then **you are not dealing with a big data problem***

*If either the data used by your app either the processing capabilities your app logic needs don't fit the available infrastructure, **then you are facing a big data problem**, and you need specialized services*

How much data is there?

WIRED MAGAZINE: ISSUE 16.07

SCIENCE : DISCOVERIES

The Petabyte Age: Because More Isn't Just More — More Is Different

Products & Services Support How to Buy

Visual Networ

The Ze

WIRED.CO.UK

US World Politics Business Tech Science Health Investigations Entertainment Sports Travel Nightly News Meet the Press Dateline TODAY

NBCNEWS TECHNOLOGY

TOPICS Gadgets Security Internet Innovation More

BIG-DATA
There's so much data that we're running out of words to describe it

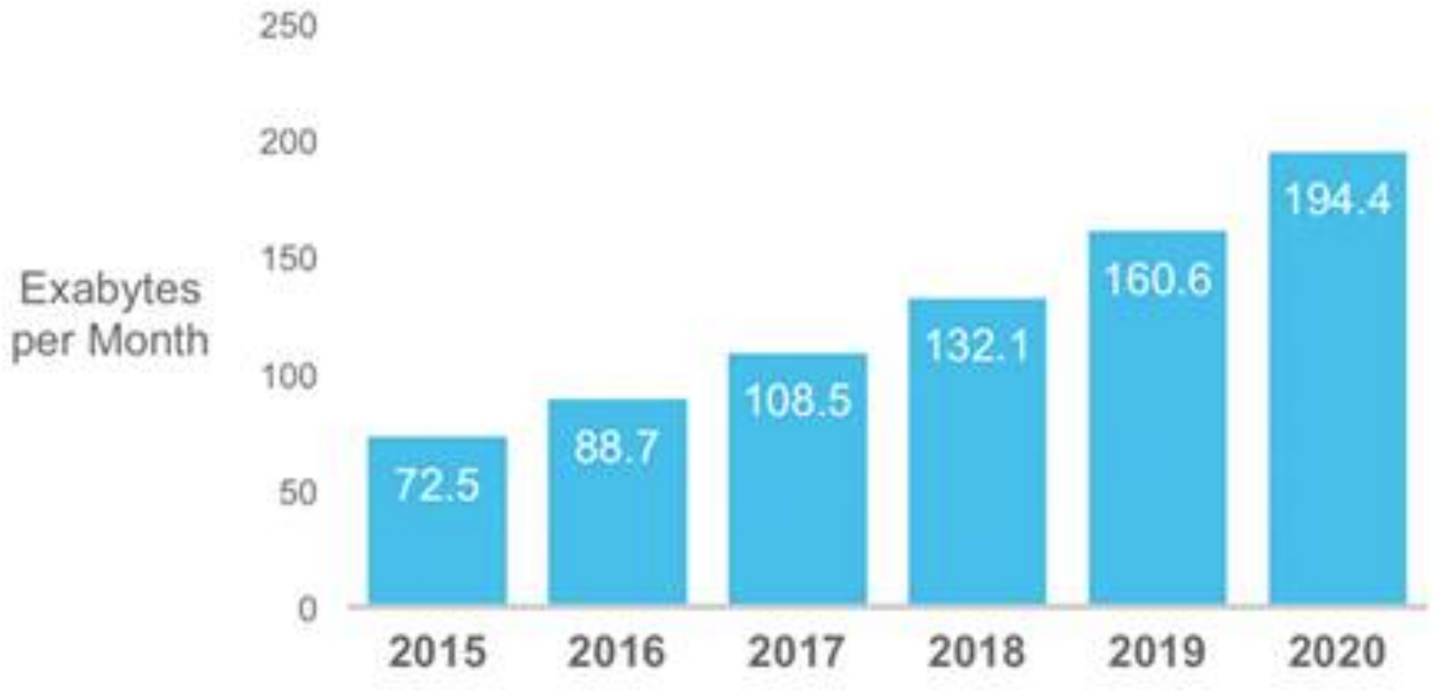
BING

There's so much data that we're running out of words to describe it

Devin Coldewey, NBC News

Dec. 11, 2012 at 6:02 PM ET

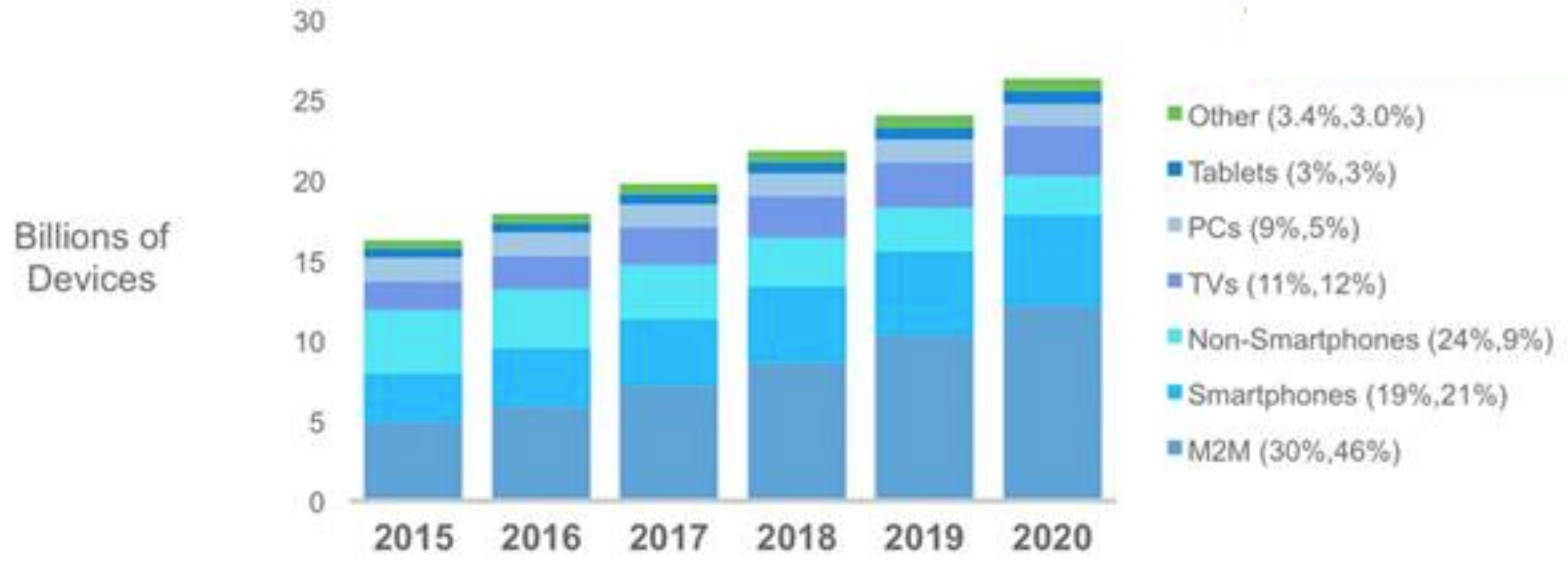
Data growing forecast



Source: Cisco VNI Global IP Traffic Forecast, 2015–2020

http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

Data growing forecast

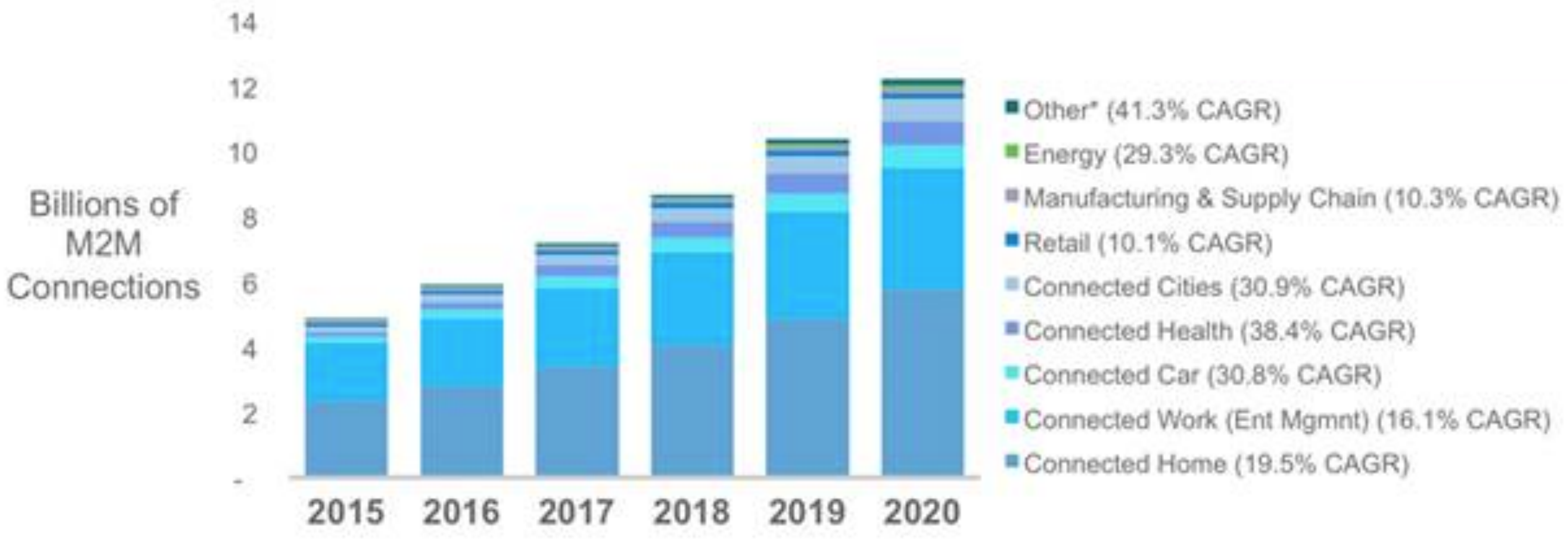


Figures (n) refer to 2015, 2020 device share.

Source: Cisco VNI Global IP Traffic Forecast, 2015–2020

http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

Data growing forecast



* Other includes Agriculture, Construction, and Emergency Services.

Source: Cisco VNI Global IP Traffic Forecast, 2015–2020

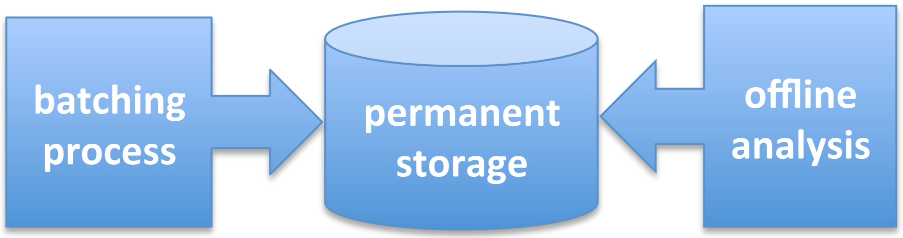
http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

<u>Two (three) approaches for dealing with Big Data:</u>

Batch and stream processing
(and Lambda architectures)

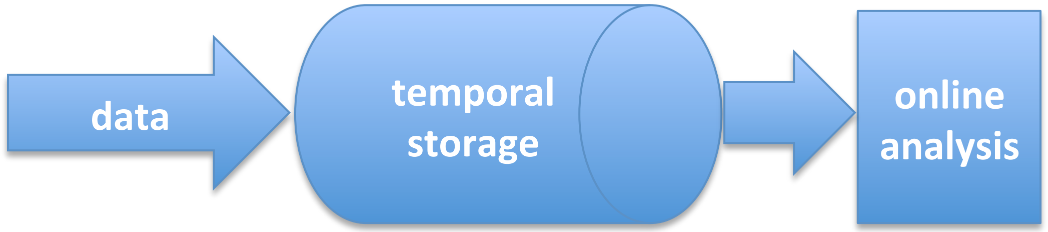
Batch processing

- It is about joining a lot of data (**batching**)
  - A *lot* may mean Terabytes or more…
  - Most probably, **data cannot be stored in a single server**
- Once joined, it is **analyzed**
  - Most probably, **data cannot be analyzed using a single process**
- **Time is not a problem**
  - **Batching can last** for days or even months
  - **Processing can last** for hours or even days
- Analysis can be **reproduced**



Stream processing

- It is about **not storing** the data and **analyzing it on the fly**
  - Most probably, **data cannot be analyzed by a single process**
- **Time is important**
  - Since the data is not stored, **it must be analyzed as it is received**
  - The results are expected to be available **in near real-time**
- Analysis **cannot be reproduced**



FIWARE

Lambda architectures

- A Big Data architecture is Lambda compliant if it produces near-real time data insights based on the last data only, while large batches are accumulated and processed for robust insights
  - Data must **feed both** batch-based and stream-based sub-systems
  - Real-time insights are **cached**
  - Batch insights are **cached**
  - Queries to the whole system **combine both kinds of cached insights**
  - Conceptually, both stream and batch **analysis must be the same** (despite stream and batch APIs may be totally different).
- http://lambda-architecture.net/

FIWARE

Batch processing

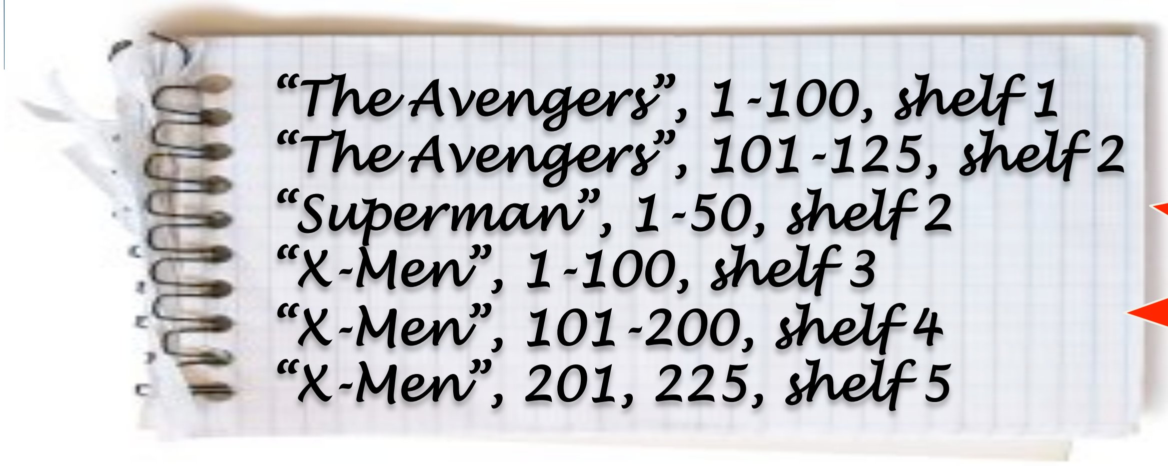
Distributed storage:

The Hadoop reference (HDFS)

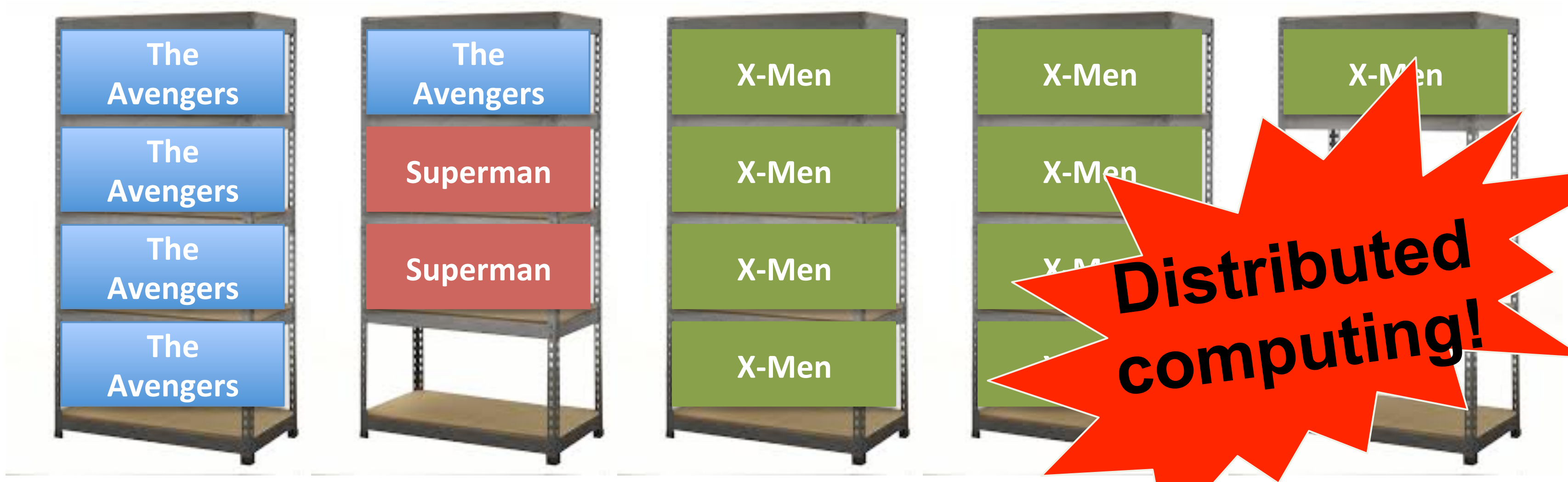
What happens if one shelving is not enough?

You buy more shelves…

… then you create an index


"The Avengers", 1-100, shelf 1
"The Avengers", 101-125, shelf 2
"Superman", 1-50, shelf 2
"X-Men", 1-100, shelf 3
"X-Men", 101-200, shelf 4
"X-Men", 201, 225, shelf 5

Distributed storage!



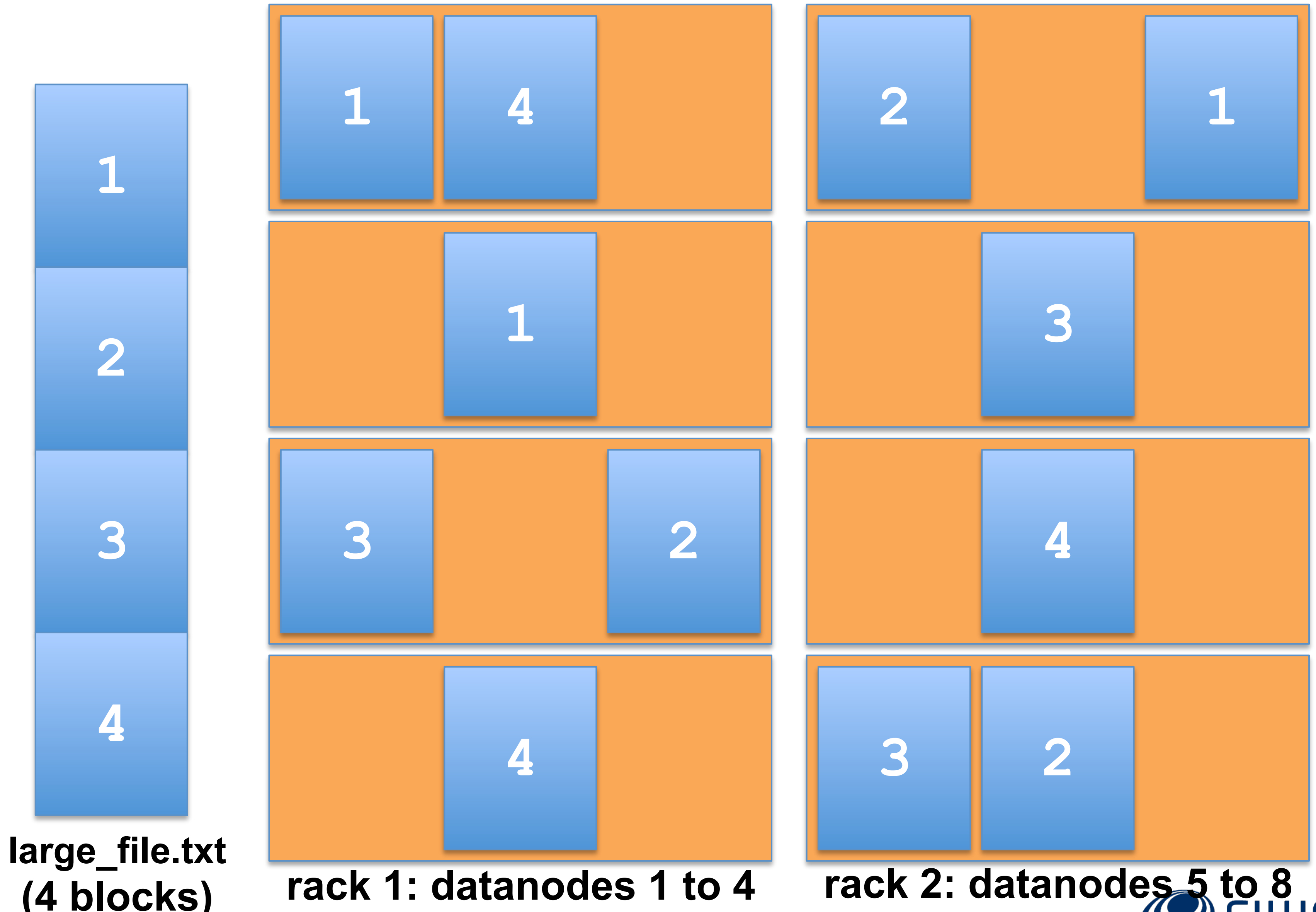
Distributed computing!

… and you call your friends to read everything!

Hadoop Distributed File System (HDFS)

- Based on **Google File System**
- Large files are stored across multiple machines (**Datanodes**) by spliting them into blocks that are distributed
- Metadata is managed by the **Namenode**
- Scalable by simply adding more Datanodes
- Fault-tolerant since HDFS replicates each block (default to 3)
- Security based on authentication (Kerberos) and authorization (permissions, HACLs)
- It is managed like a Unix-like file system

Spliting, replication and distribution



Namenode metadata

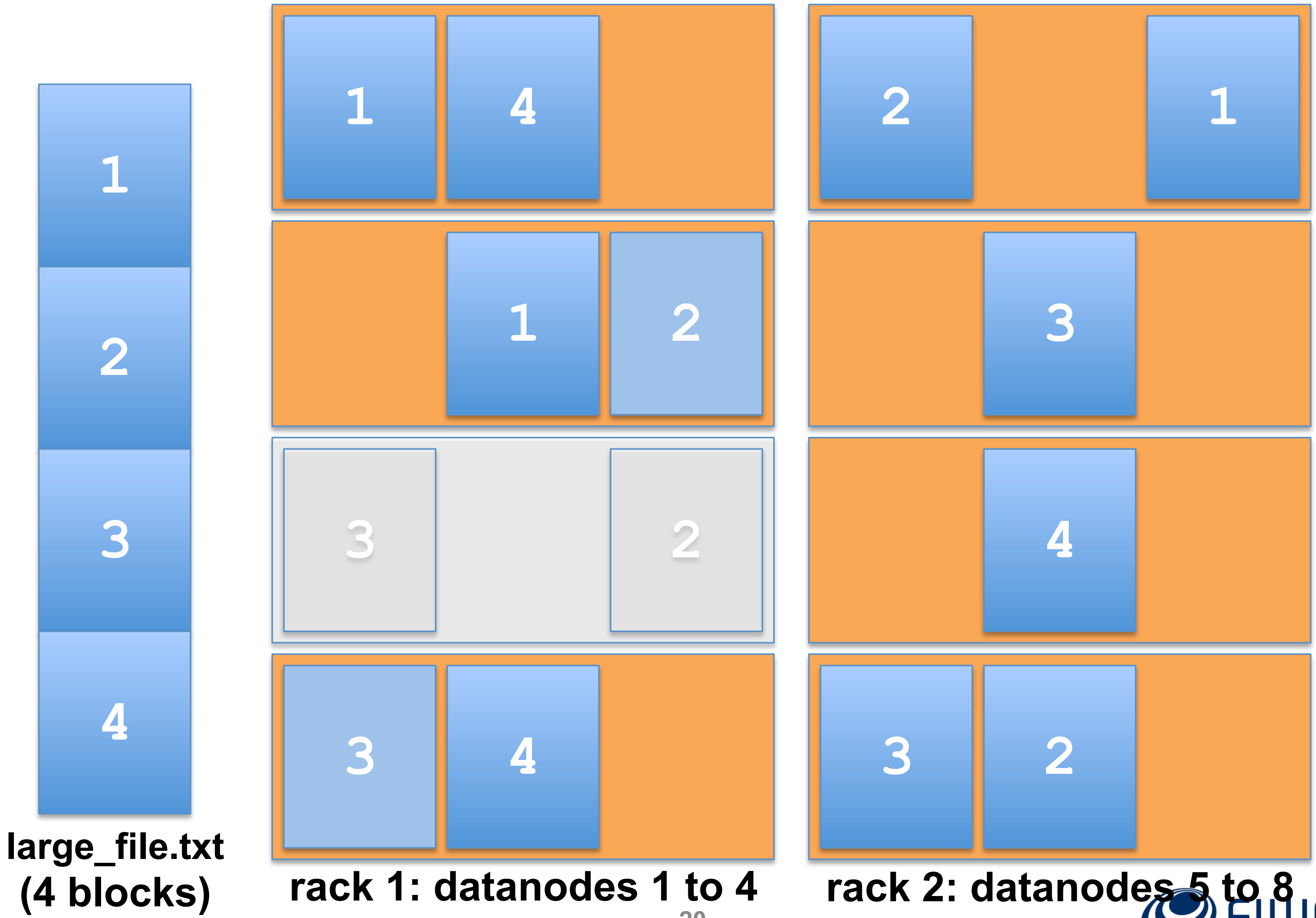
Path	Replicas	Block IDs

/user/user1/data/large_file.txt	3	1 → {dn1,dn2,dn5}
		2 → {dn3,dn5,dn8}
		3 → {dn3,dn6,dn8}
		4 → {dn1,dn4,dn7}

/user/user1/data/other_file.txt	2	5 → {…}
		6 → {…}
		7 → {…}

…	…	…

Datanodes failure recovering



Namenode failure recovering

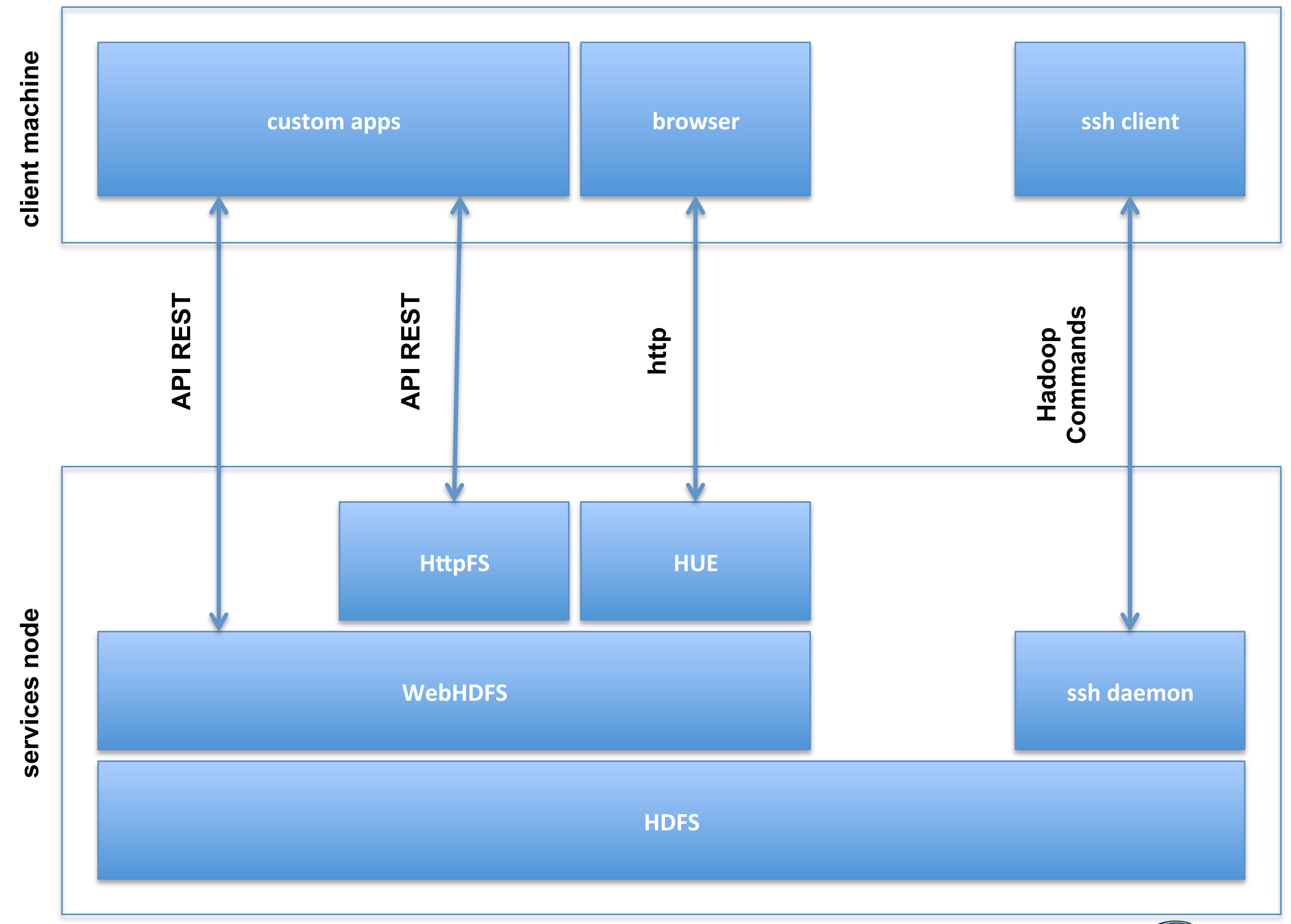
Path	Replicas	Block IDs

/user/user1/data/large_file.txt	3	1 → {dn1,dn2,dn5}
		2 → {dn2,dn5,dn8}
		3 → {dn4,dn6,dn8}
		4 → {dn1,dn4,dn7}

/user/user1/data/other_file.txt	2	5 → {…}
		6 → {…}
		7 → {…}

…	…	…

Managing HDFS



Managing HDFS: Hadoop Commands

- Hadoop Commands
  - Hadoop Common Commands
    - User Commands
      - File System Shell
      - jar
      - …
    - Administrative Commands
  - Hadoop HDFS Commands
    - User Commands
      - dfs
      - fsck
      - …
    - Administrative Commands
      - dfsadmin
      - …
    - Debug Commands

- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/CommandsManual.html#Hadoop_Common_Commands
- https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html

Managing HDFS: File System Shell

- It includes various **shell-like comands** that directly interact with HDFS
- The FS shell is invoked by any of these scripts:

```
$ bin/hadoop fs
$ bin/hdfs dfs
```

- All FS Shell commans take **URI** paths as arguments:
  - scheme://authority/path
  - Available schemas: file (local FS), hdfs (HDFS)
  - If nothing is especified, hdfs is considered
- It is necessary to connect to the cluster via SSH
- Full commands reference
  - http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

Managing HDFS: File System Shell

```
$ ssh frb@cosmos.lab.fiware.org “hadoop fs -cat webinar/
abriefhistoryoftime_page1”
CHAPTER 1
OUR PICTURE OF THE UNIVERSE
A well-known scientist (some say it was Bertrand Russell) once gave
a public lecture on astronomy. He described how the earth orbits
around the sun and how the sun, in turn, orbits around the center of
a vast
$ ssh frb@cosmos.lab.fiware.org “hadoop fs -mkdir webinar/afolder”
$ ssh frb@cosmos.lab.fiware.org “hadoop fs -ls webinar”
Found 4 items
-rw-r--r--   3 frb cosmos       3431 2014-12-10 14:00 /user/frb/
webinar/abriefhistoryoftime_page1
-rw-r--r--   3 frb cosmos       1604 2014-12-10 14:00 /user/frb/
webinar/abriefhistoryoftime_page2
-rw-r--r--   3 frb cosmos       5257 2014-12-10 14:00 /user/frb/
webinar/abriefhistoryoftime_page3
drwxr-xr-x   - frb cosmos          0 2015-03-10 11:09 /user/frb/
webinar/afolder
$ ssh frb@cosmos.lab.fiware.org “hadoop fs -rmr webinar/afolder”
Deleted hdfs://cosmosmaster-gi/user/frb/webinar/afolder
```

Managing HDFS: Other useful commands

```
$ ssh root@cosmos.lab.fiware.org "hadoop dfsadmin -report"
root@cosmos.lab.fiware.org's password:
Configured Capacity: 913243410432 (850.52 GB)
Present Capacity: 791656620865 (737.29 GB)
DFS Remaining: 38707150848 (36.05 GB)
DFS Used: 752949470017 (701.24 GB)
DFS Used%: 95.11%
Under replicated blocks: 191
Blocks with corrupt replicas: 204
Missing blocks: 45

-------------------------------------------------
Datanodes available: 9 (9 total, 0 dead)

Name: 192.168.189.10:50010
Decommission Status : Normal
Configured Capacity: 101471490048 (94.5 GB)
DFS Used: 80987041283 (75.43 GB)
Non DFS Used: 15017435645 (13.99 GB)
DFS Remaining: 5467013120(5.09 GB)
DFS Used%: 79.81%
DFS Remaining%: 5.39%
Last contact: Tue Nov 17 15:06:07 CET 2015
...
```

Managing HDFS: Other useful commands

```
$ ssh root@cosmos.lab.fiware.org "hadoop fsck /user/frb/webinar/
abriefhistoryoftime_page1 -files -blocks"
root@cosmos.lab.fiware.org's password:
FSCK started by root (auth:SIMPLE) from /192.168.189.1 for path /user/frb/webinar/
abriefhistoryoftime_page1 at Tue Nov 17 15:12:11 CET 2015
/user/frb/webinar/abriefhistoryoftime_page1 3431 bytes, 1 block(s):  OK
0. blk_-5949851128377283808_25352631 len=3431 repl=3

Status: HEALTHY
 Total size:	3431 B
 Total dirs:	0
 Total files:	1
 Total blocks (validated):	1 (avg. block size 3431 B)
 Minimally replicated blocks:	1 (100.0 %)
 Over-replicated blocks:	0 (0.0 %)
 Under-replicated blocks:	0 (0.0 %)
 Mis-replicated blocks:		0 (0.0 %)
 Default replication factor:	3
 Average block replication:	3.0
 Corrupt blocks:		0
 Missing replicas:		0 (0.0 %)
 Number of data-nodes:		9
 Number of racks:		1
FSCK ended at Tue Nov 17 15:12:11 CET 2015 in 1 milliseconds


The filesystem under path '/user/frb/webinar/abriefhistoryoftime_page1' is HEALTHY
```

Managing HDFS: HTTP REST API

- The HTTP REST API supports the **complete File System** interface for HDFS
  - Other Hadoop commands are not available through a REST API
- It relies on the `webhdfs` schema for URIs

```
webhdfs://<HOST>:<HTTP_PORT>/<PATH>
```

- **HTTP URLs** are built as:

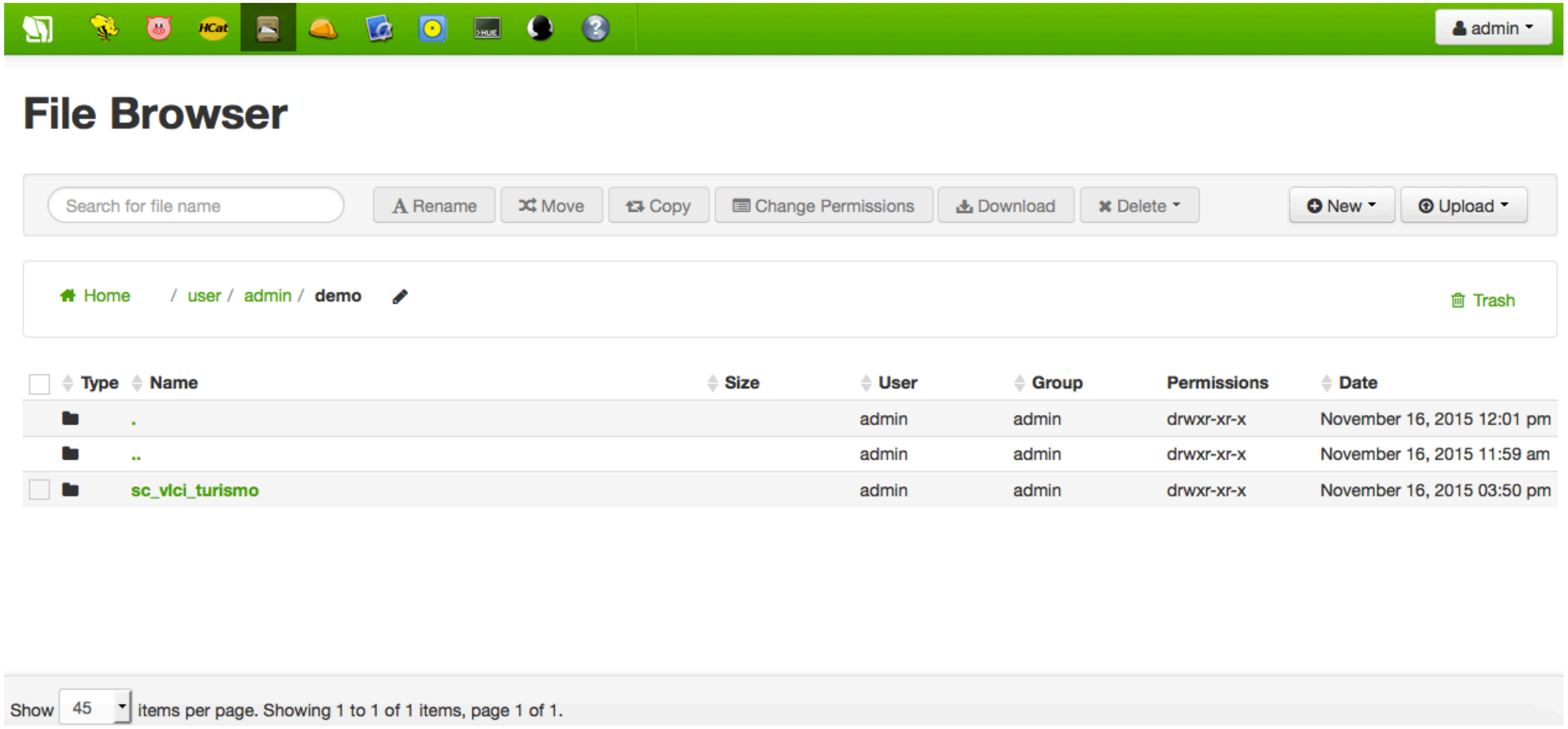
```
http://<HOST>:<HTTP_PORT>/webhdfs/v1/<PATH>?op=…
```

- Full API specification
  - http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html

Managing HDFS: HTTP REST API

```
$ curl -X GET "http://cosmos.lab.fi-ware.org:14000/webhdfs/v1/user/frb/webinar/
abriefhistoryoftime_page1?op=open&user.name=frb"
CHAPTER 1
OUR PICTURE OF THE UNIVERSE
A well-known scientist (some say it was Bertrand Russell) once gave a public lecture on
astronomy. He described how the earth orbits around the sun and how the sun, in turn,
orbits around the center of a vast
$ curl -X PUT "http://cosmos.lab.fi-ware.org:14000/webhdfs/v1/user/frb/webinar/afolder?
op=mkdirs&user.name=frb"
{"boolean":true}
$ curl -X GET "http://cosmos.lab.fi-ware.org:14000/webhdfs/v1/user/frb/webinar?
op=liststatus&user.name=frb"
{"FileStatuses":{"FileStatus":
[{"pathSuffix":"abriefhistoryoftime_page1","type":"FILE","length":
3431,"owner":"frb","group":"cosmos","permission":"644","accessTime":
1425995831489,"modificationTime":1418216412441,"blockSize":67108864,"replication":3},
{"pathSuffix":"abriefhistoryoftime_page2","type":"FILE","length":
1604,"owner":"frb","group":"cosmos","permission":"644","accessTime":
1418216412460,"modificationTime":1418216412500,"blockSize":67108864,"replication":3},
{"pathSuffix":"abriefhistoryoftime_page3","type":"FILE","length":
5257,"owner":"frb","group":"cosmos","permission":"644","accessTime":
1418216412515,"modificationTime":1418216412551,"blockSize":67108864,"replication":3},
{"pathSuffix":"afolder","type":"DIRECTORY","length":
0,"owner":"frb","group":"cosmos","permission":"755","accessTime":0,"modificationTime":
1425995941361,"blockSize":0,"replication":0}]}}
$ curl -X DELETE "http://cosmos.lab.fi-ware.org:14000/webhdfs/v1/user/frb/webinar/
afolder?op=delete&user.name=frb"
{"boolean":true}
```

Managing HDFS: Hue File Browser



Type	Name	Size	User	Group	Permissions	Date
	.		admin	admin	drwxr-xr-x	November 16, 2015 12:01 pm
	..		admin	admin	drwxr-xr-x	November 16, 2015 11:59 am
	sc_vlci_turismo		admin	admin	drwxr-xr-x	November 16, 2015 03:50 pm

Show 45 items per page. Showing 1 to 1 of 1 items, page 1 of 1.

Batch processing

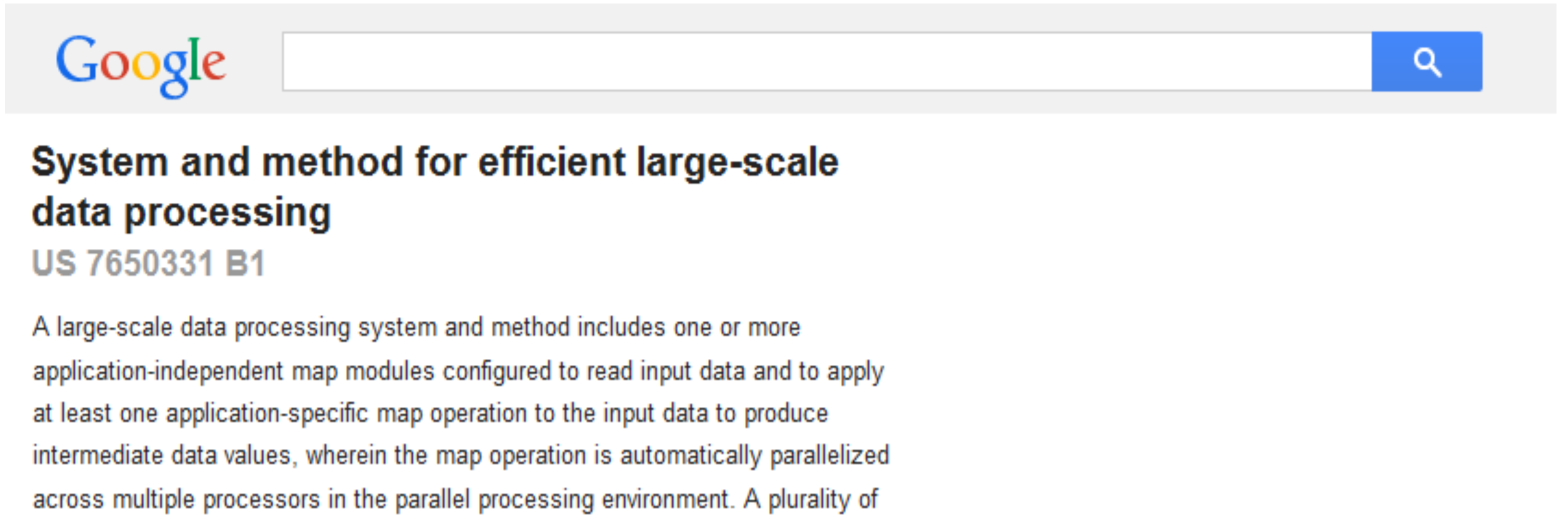
Distributed computing:

The Hadoop reference

(MapReduce)

Hadoop was created by Doug Cutting at Yahoo!...

… based on the MapReduce patent by Google

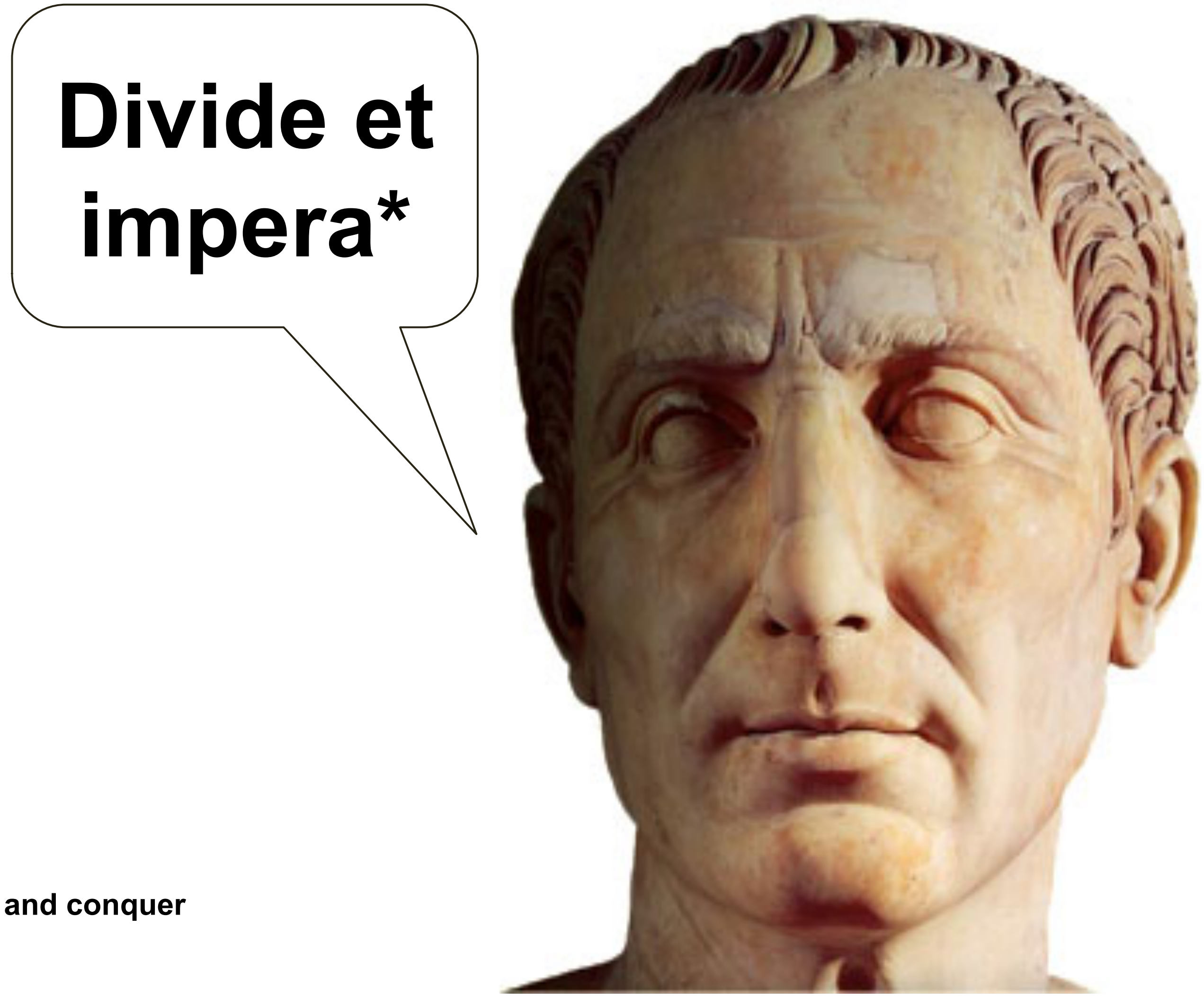


System and method for efficient large-scale data processing

US 7650331 B1

A large-scale data processing system and method includes one or more application-independent map modules configured to read input data and to apply at least one application-specific map operation to the input data to produce intermediate data values, wherein the map operation is automatically parallelized across multiple processors in the parallel processing environment. A plurality of

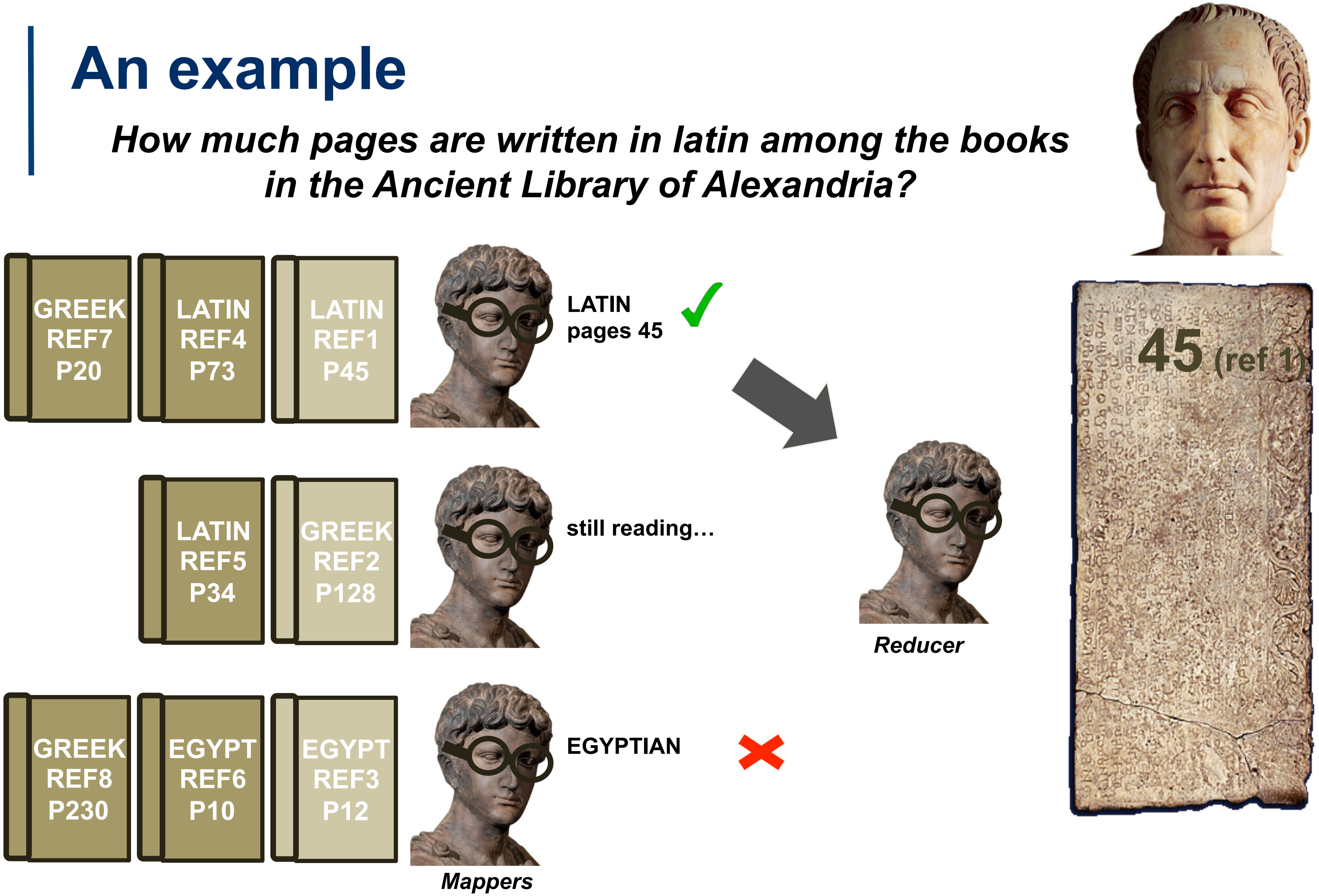
Well, MapReduce was really invented by Julius Caesar



* Divide and conquer

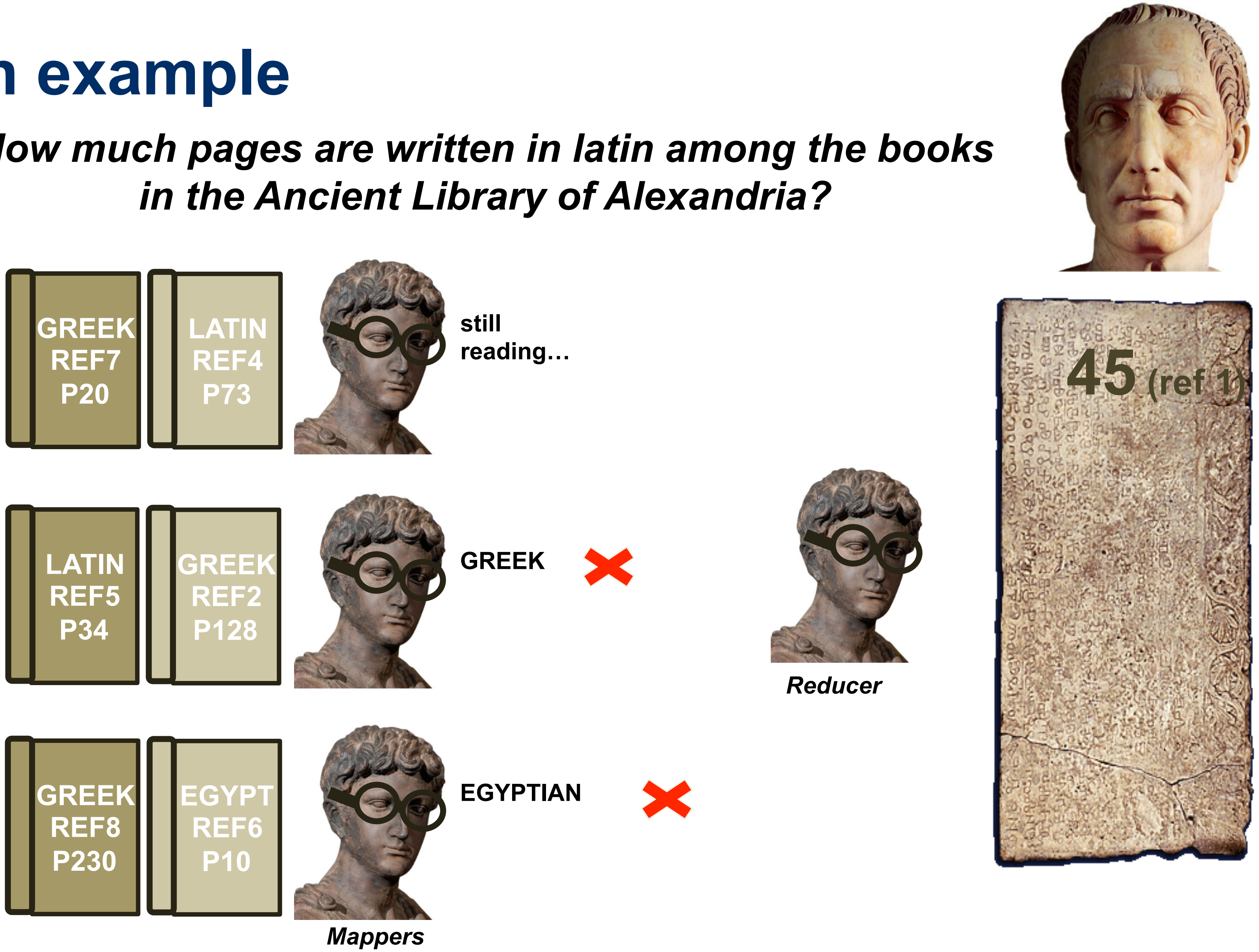
An example

How much pages are written in latin among the books in the Ancient Library of Alexandria?



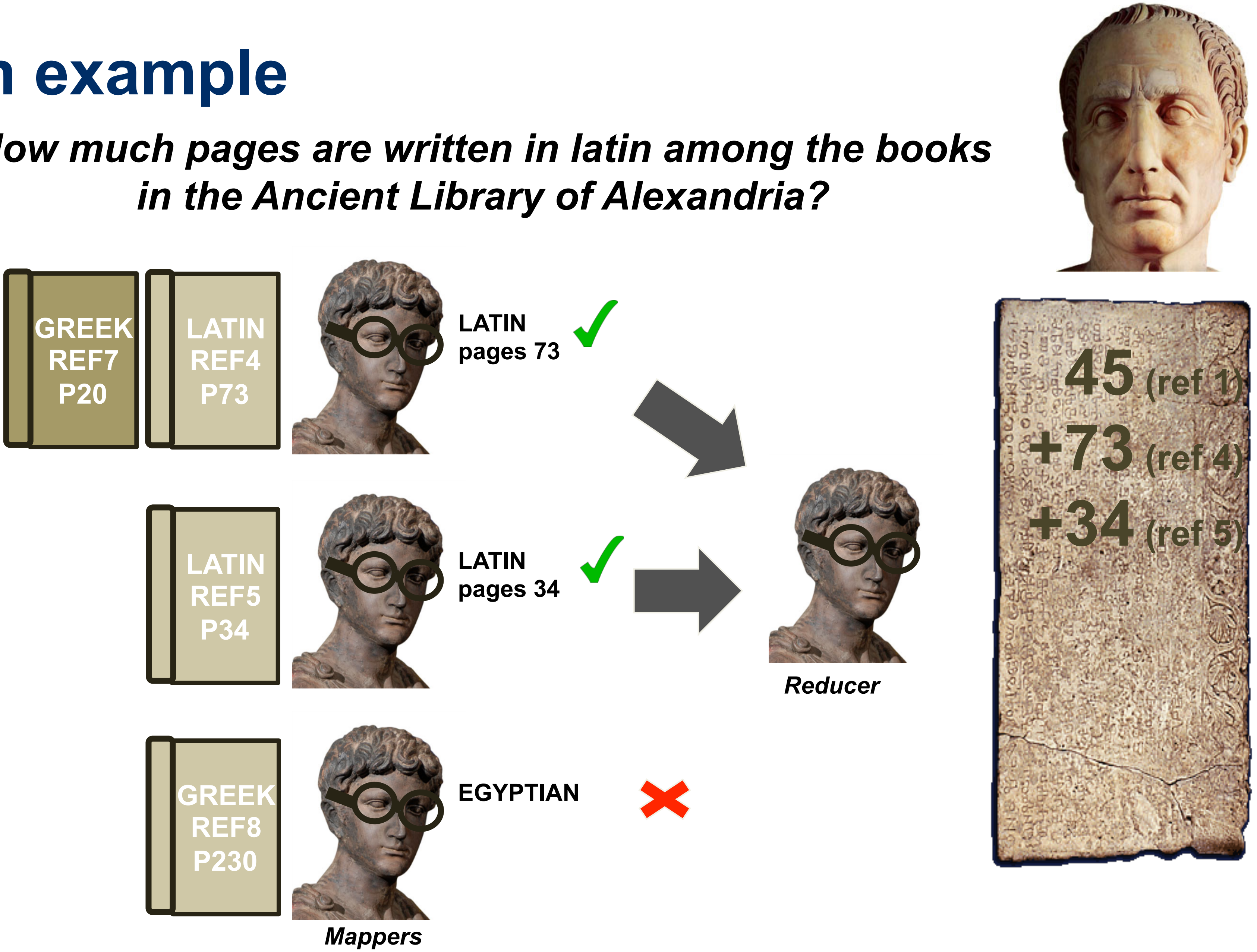
An example

How much pages are written in latin among the books in the Ancient Library of Alexandria?



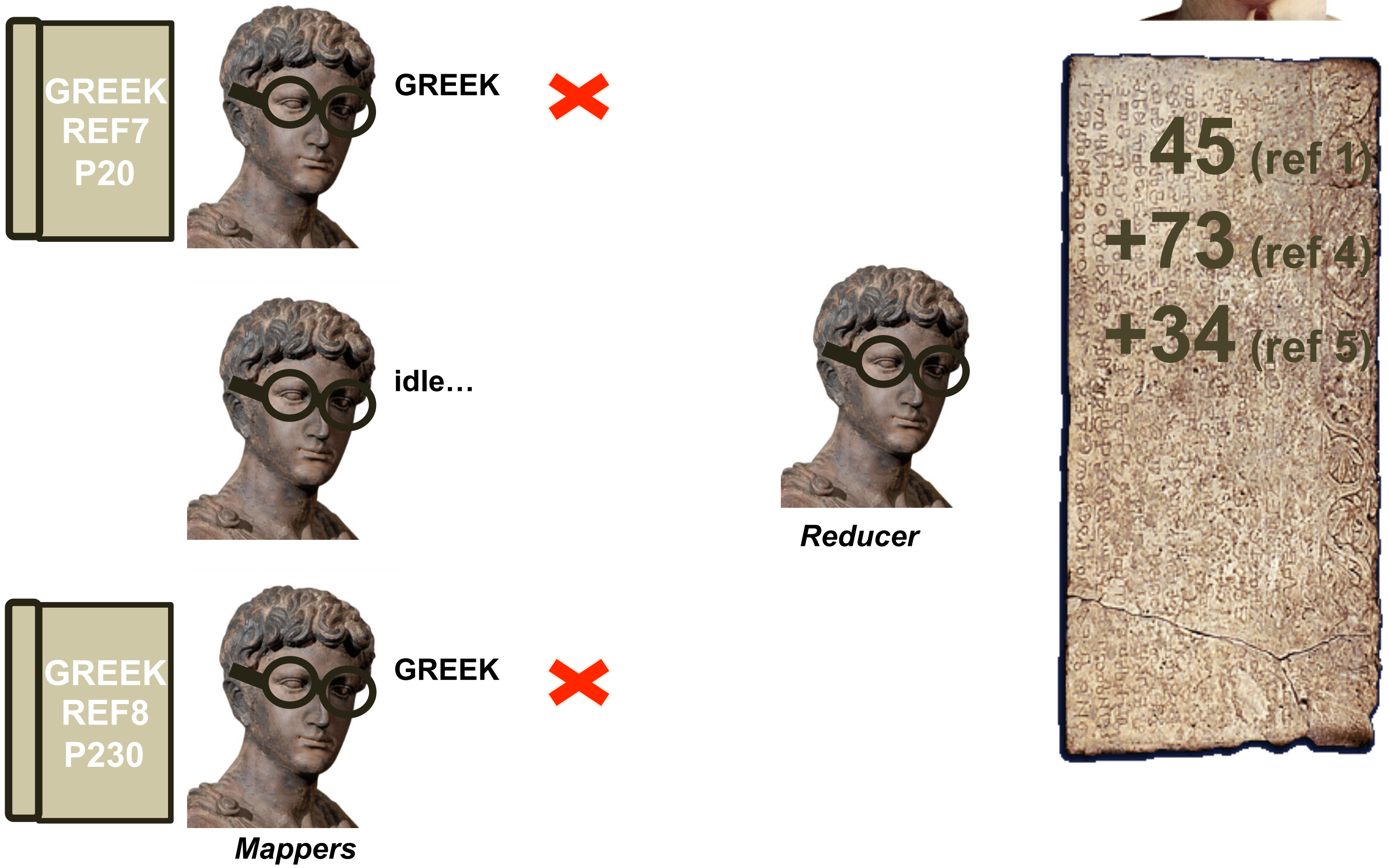
An example

How much pages are written in latin among the books in the Ancient Library of Alexandria?



An example

How much pages are written in latin among the books in the Ancient Library of Alexandria?

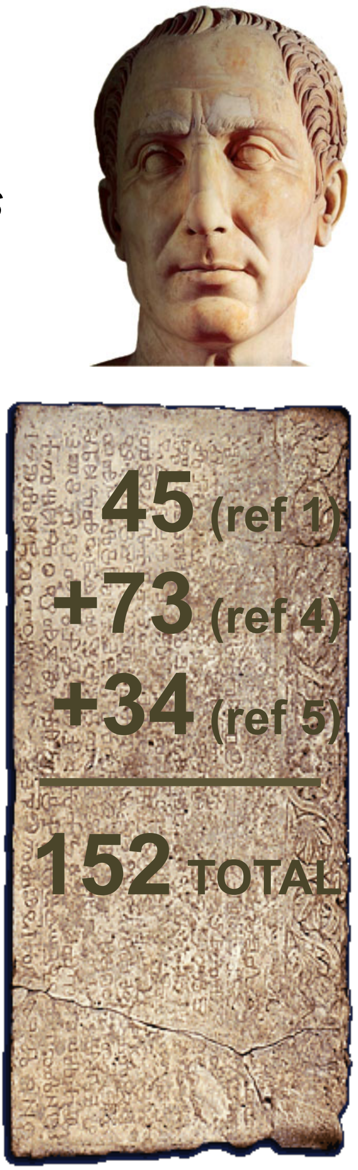


An example

How much pages are written in latin among the books in the Ancient Library of Alexandria?

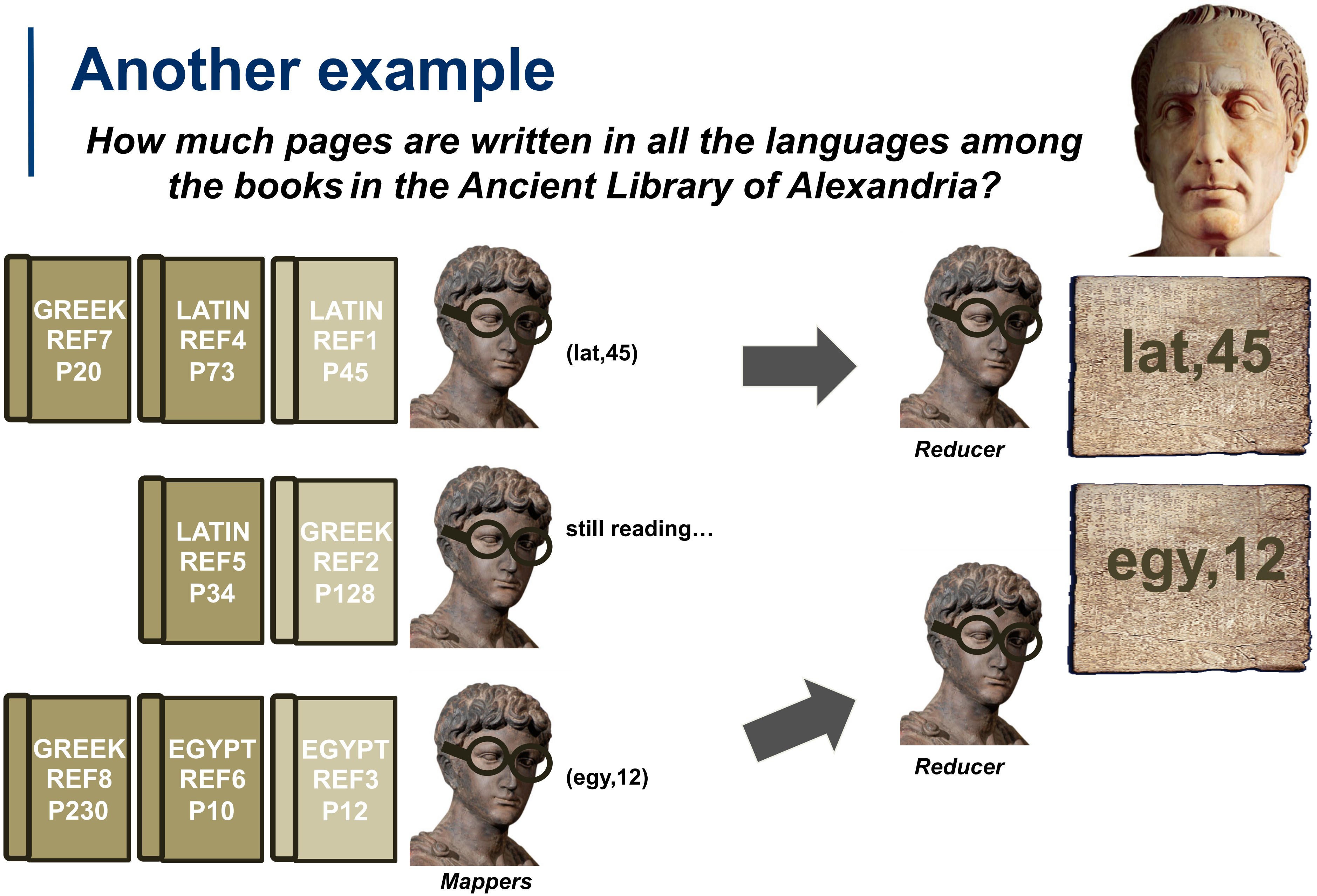
Mappers

Reducer



Another example

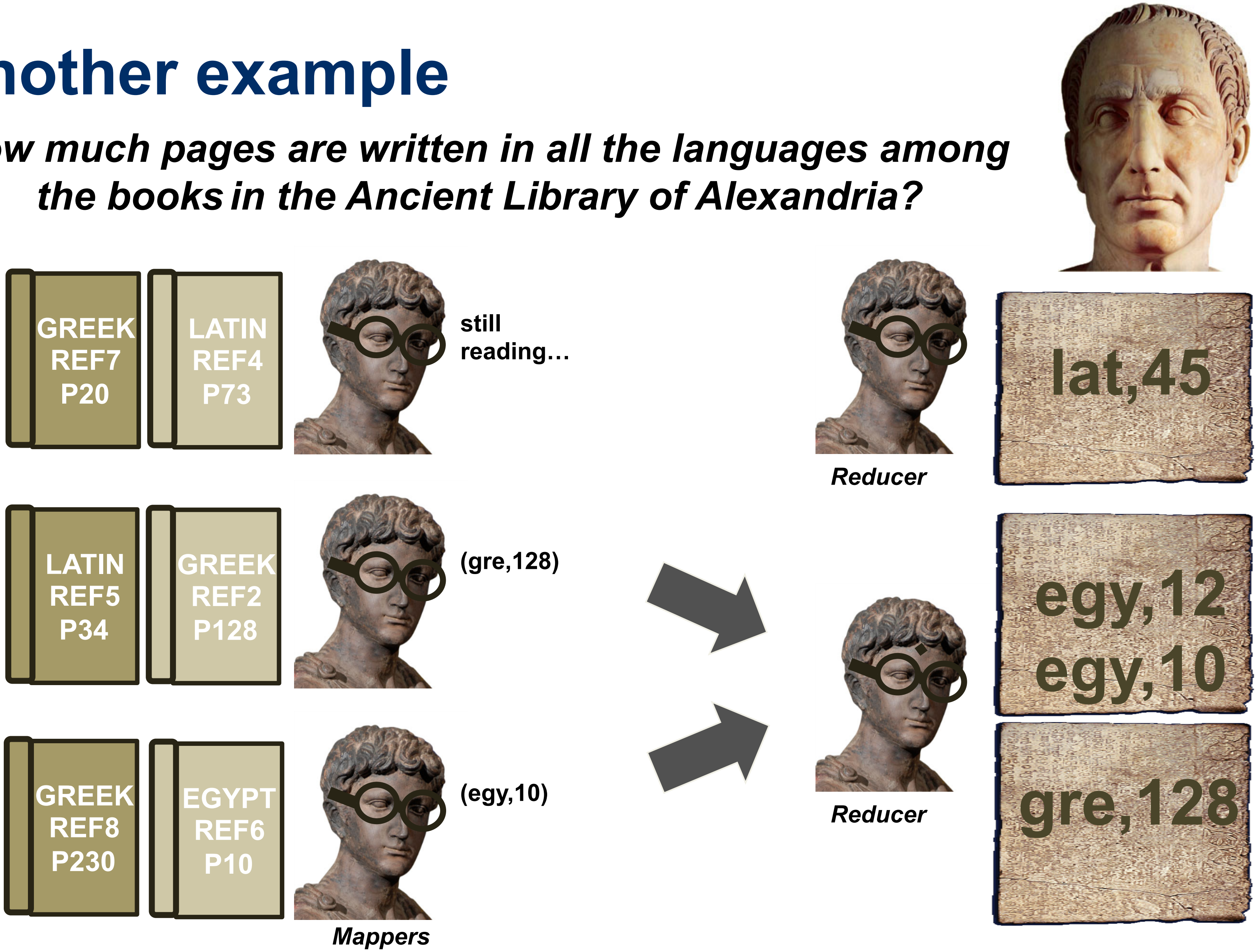
How much pages are written in all the languages among the books in the Ancient Library of Alexandria?



FIWARE

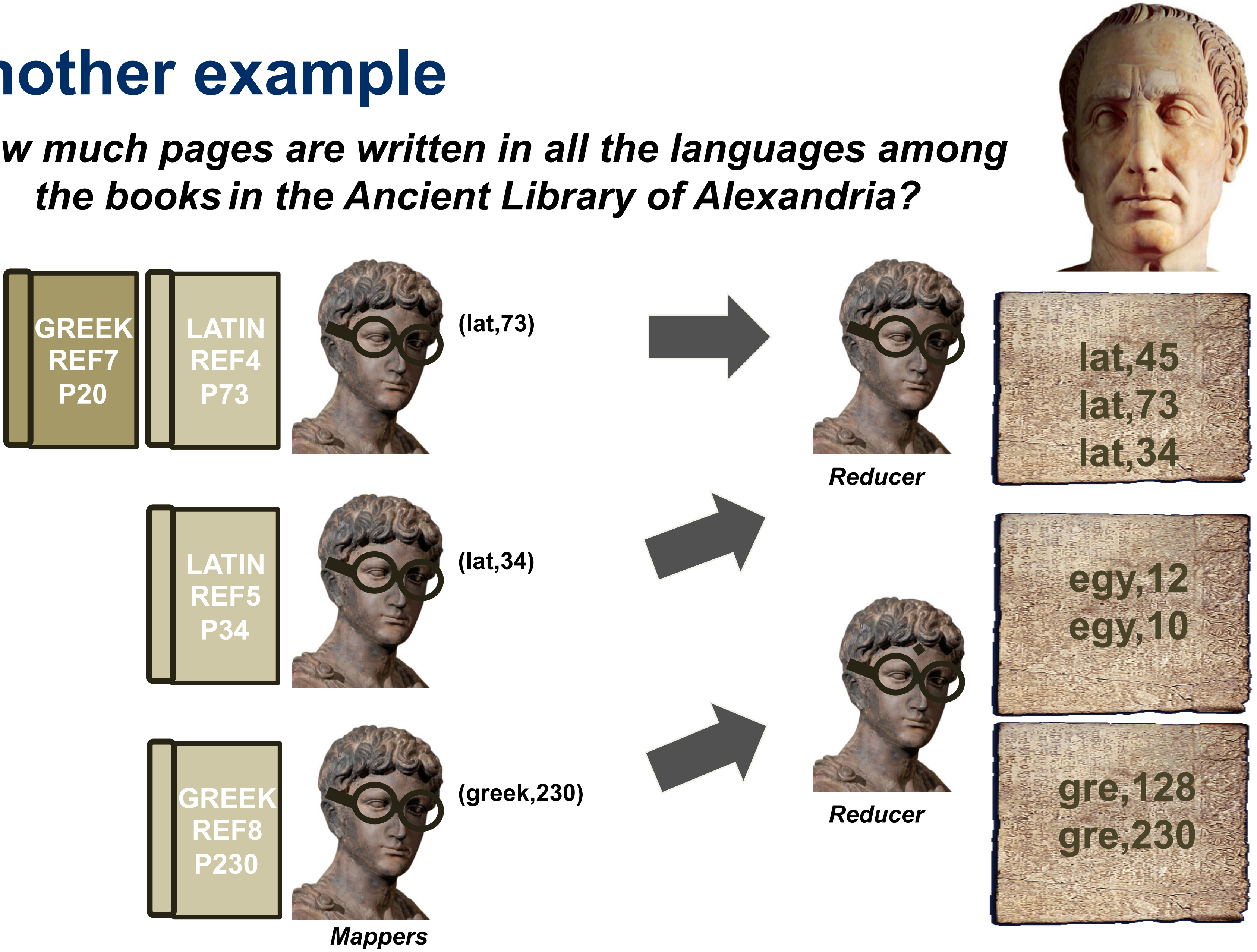
Another example

How much pages are written in all the languages among the books in the Ancient Library of Alexandria?



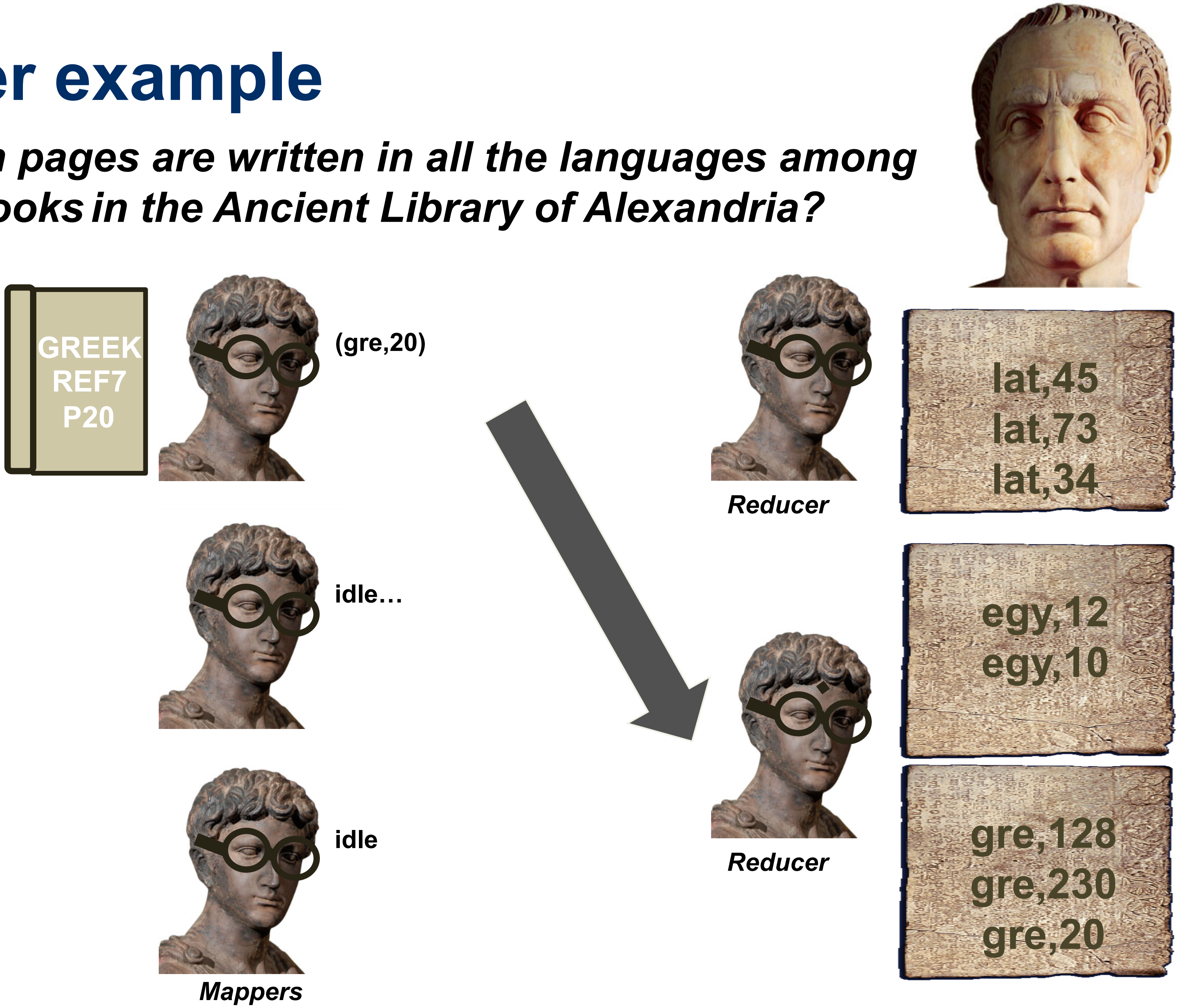
Another example

How much pages are written in all the languages among the books in the Ancient Library of Alexandria?



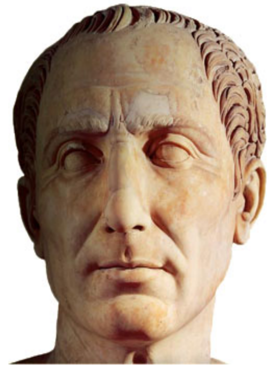
Another example

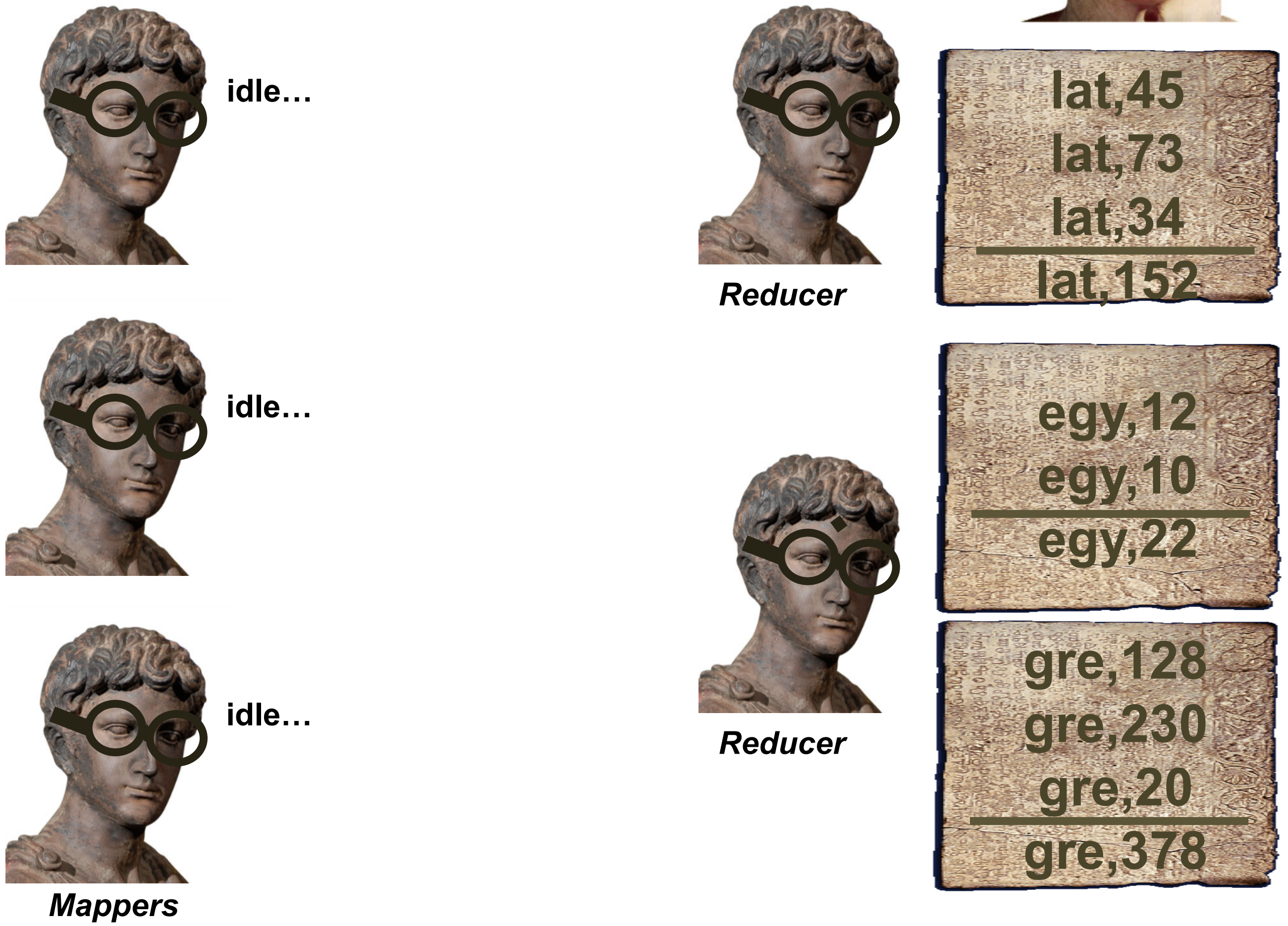
How much pages are written in all the languages among the books in the Ancient Library of Alexandria?



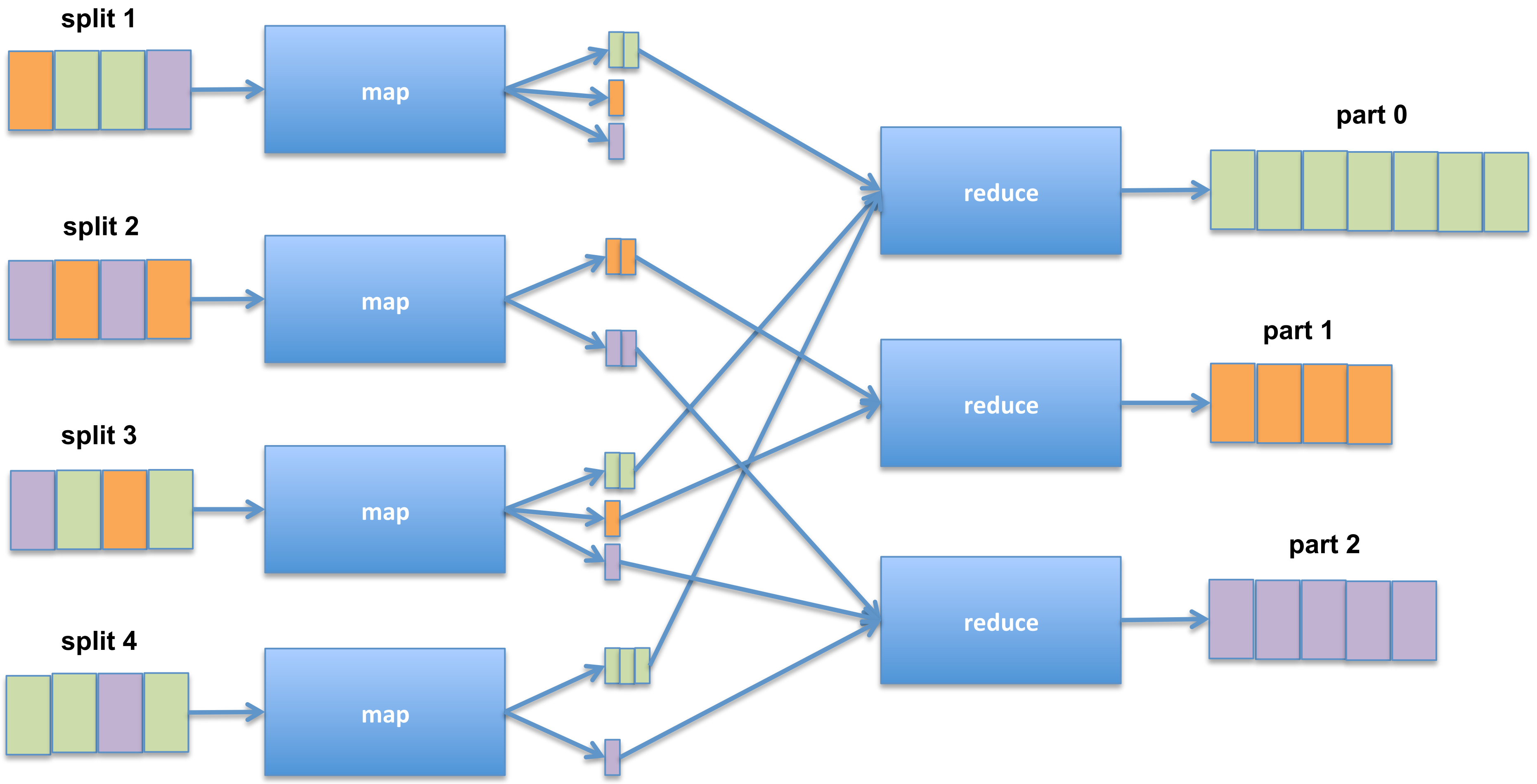
Another example

How much pages are written in all the languages among the books in the Ancient Library of Alexandria?





Example conceptualized



Writing MapReduce applications

- MapReduce applications are commonly **written in Java**
  - Can be written in other languages through **Hadoop Streaming**
- They are executed in the **command line**

```
$ hadoop jar <jar-file> <main-class> <input-dir>
<output-dir>
```

- A MapReduce job consists of:
  - A **driver**, a piece of software where to define inputs, outputs, formats, etc. and the entry point for launching the job
  - A set of **Mappers**, given by a piece of software defining its behaviour
  - A set of **Reducers**, given by a piece of software defining its behaviour
- https://hadoop.apache.org/docs/current/api/ (MapReduce section)

Implementing the example

- The input will be a single big file containing:

```
symbolae botanicae,latin,230
mathematica,greek,95
physica,greek,109
ptolomaics,egyptian,120
terra,latin,541
iustitia est vincit,latin,134
```

- The mappers will receive pieces of the above file, which will be read line by line
  - Each line will be represented by a (key,value) pair, i.e. the offset on the file and the real data within the line, respectively
  - For each input pair a (key,value) pair will be output, i.e. a common “num_pages” key and the third field in the line
- The reducers will receive arrays of pairs produced by the mappers, all having the same key (“num_pages”)
  - For each array of pairs, the sum of the values will be output as a (key,value) pair, in this case a “total_pages” key and the sum as value

Implementing the example: JCMapper.class

```
public static class JCMapper extends
Mapper<Object, Text, Text, IntWritable> {

   private final Text globalKey = new Text(”num_pages");
   private final IntWritable bookPages = new IntWritable();

   @Override
   public void map(Object key, Text value, Context context)
   throws Exception {
      String[] fields = value.toString().split(“,”);
      system.out.println(“Processing “ + fields[0]);

      if (fields[1].equals(“latin”)) {
         bookPages.set(fields[2]);
         context.write(globalKey, bookPages);
      } // if
   } // map

} // JCMapper
```

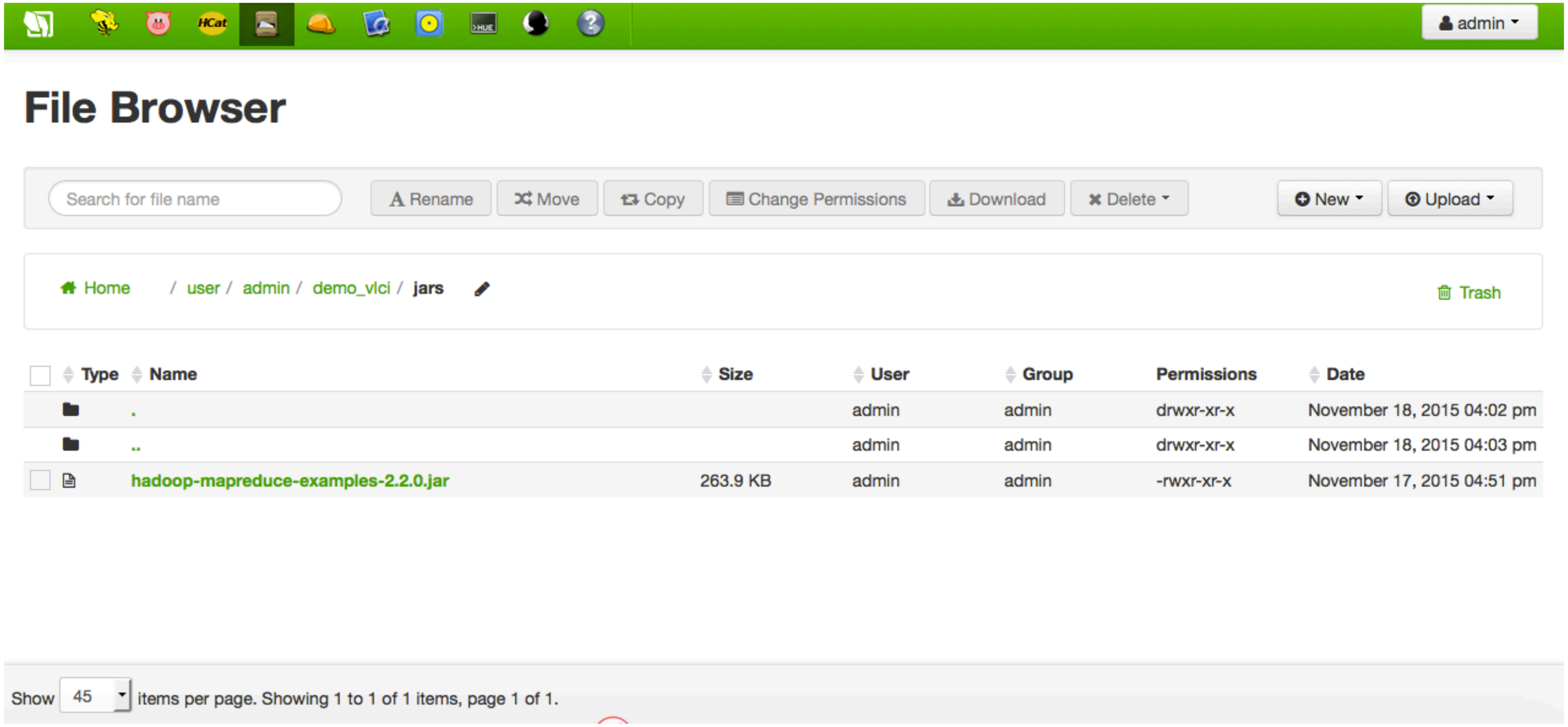
Implementing the example: JCReducer.class

```
public static class JCReducer extends
Reducer<Text, IntWritable, Text, IntWritable> {

   private final IntWritable totalPages= new IntWritable();

   @Override
   public void reduce(Text globalKey, Iterable<IntWritable>
   bookPages, Context context) throws Exception {
      int sum = 0;

      for (IntWritable val: bookPages) {
         sum += val.get();
      } // for

      totalPages.set(sum);
      context.write(globalKey, totalPages);
   } // reduce

} // JCReducer
```

Implementing the example: JC.class

```
public static void main(String[] args) throws Exception {
   int res = ToolRunner.run(
      new Configuration(), new CKANMapReduceExample(), args);
   System.exit(res);
} // main

@Override
public int run(String[] args) throws Exception {
   Configuration conf = this.getConf();
   Job job = Job.getInstance(conf, ”julius caesar");
   job.setJarByClass(JC.class);
   job.setMapperClass(JCMapper.class);
   job.setCombinerClass(JCReducer.class);
   job.setReducerClass(JCReducer.class);
   job.setOutputKeyClass(Text.class);
   job.setOutputValueClass(IntWritable.class);
   FileInputFormat.addInputPath(job, new Path(args[0]));
   FileOutputFormat.setOutputPath(job, new Path(args[1]));
   return job.waitForCompletion(true) ? 0 : 1;
} // run
```

MapReduce through Hue (jar upload)

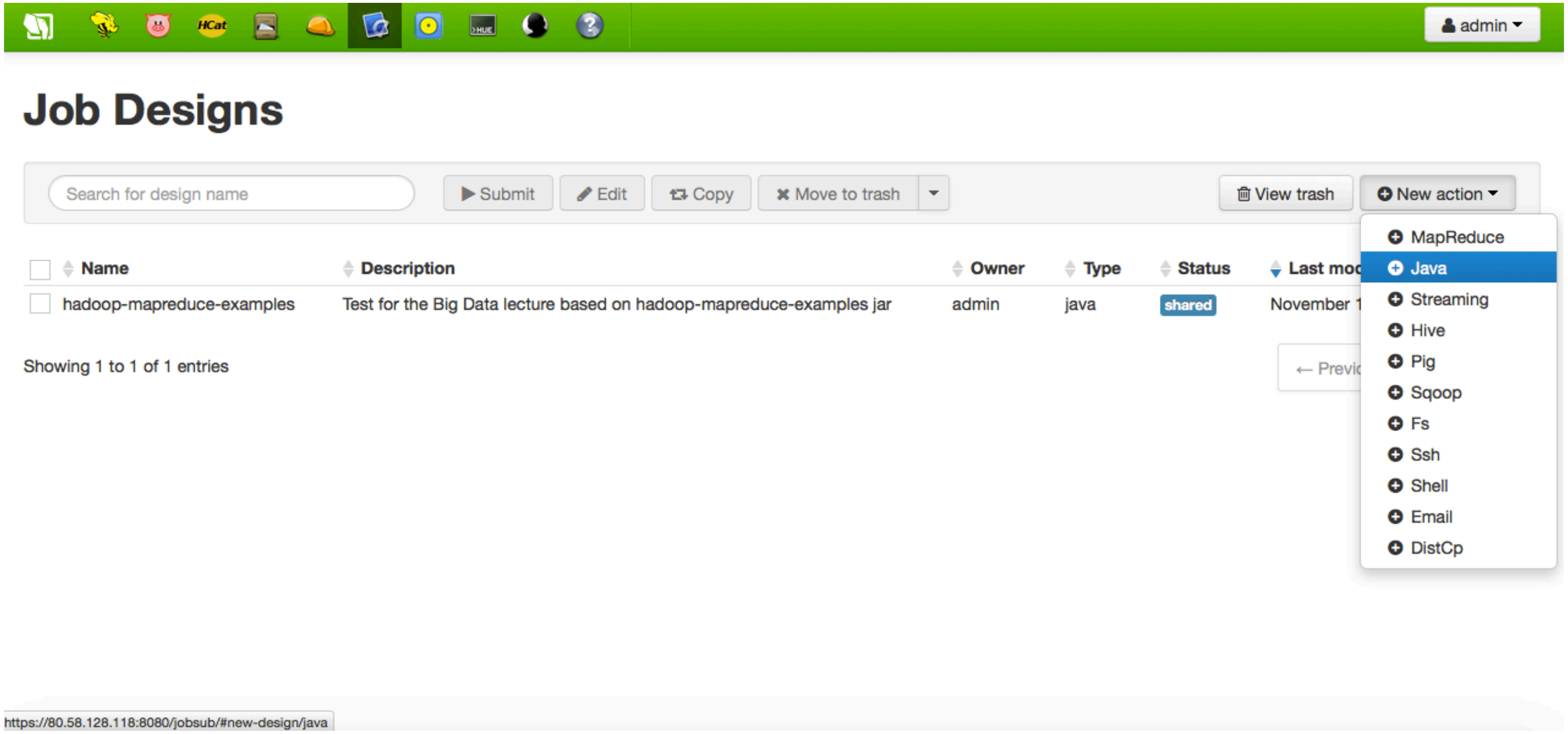


Type	Name	Size	User	Group	Permissions	Date
	.		admin	admin	drwxr-xr-x	November 18, 2015 04:02 pm
	..		admin	admin	drwxr-xr-x	November 18, 2015 04:03 pm
	hadoop-mapreduce-examples-2.2.0.jar	263.9 KB	admin	admin	-rwxr-xr-x	November 17, 2015 04:51 pm

Show 45 items per page. Showing 1 to 1 of 1 items, page 1 of 1.

http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-mapreduce-examples/2.2.0

MapReduce through Hue (MapReduce action)

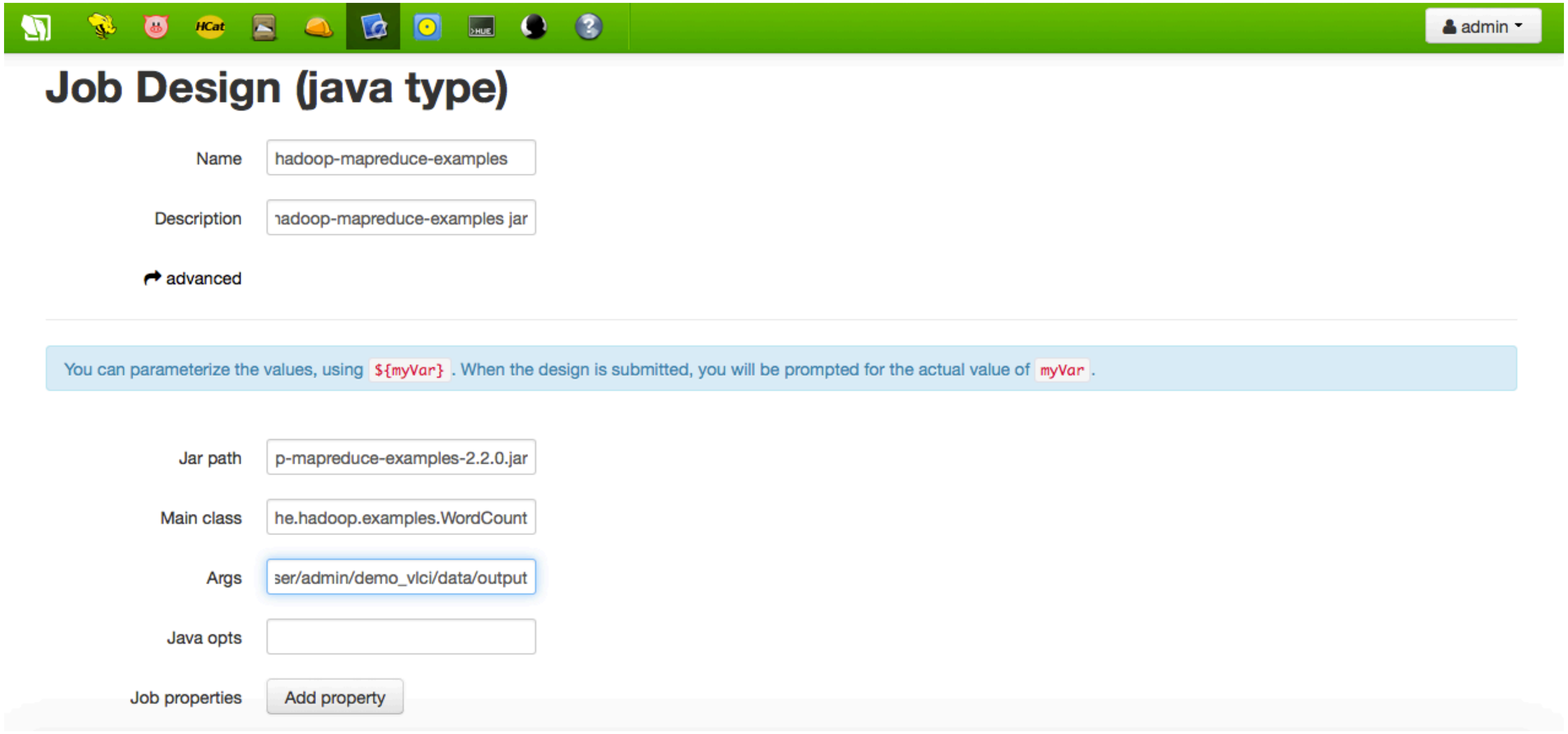

Job Designs

	Name	Description	Owner	Type	Status	Last mod
	hadoop-mapreduce-examples	Test for the Big Data lecture based on hadoop-mapreduce-examples jar	admin	java	shared	November 1

Showing 1 to 1 of 1 entries

https://80.58.128.118:8080/jobsub/#new-design/java

MapReduce through Hue (parameterization)



Job Design (java type)

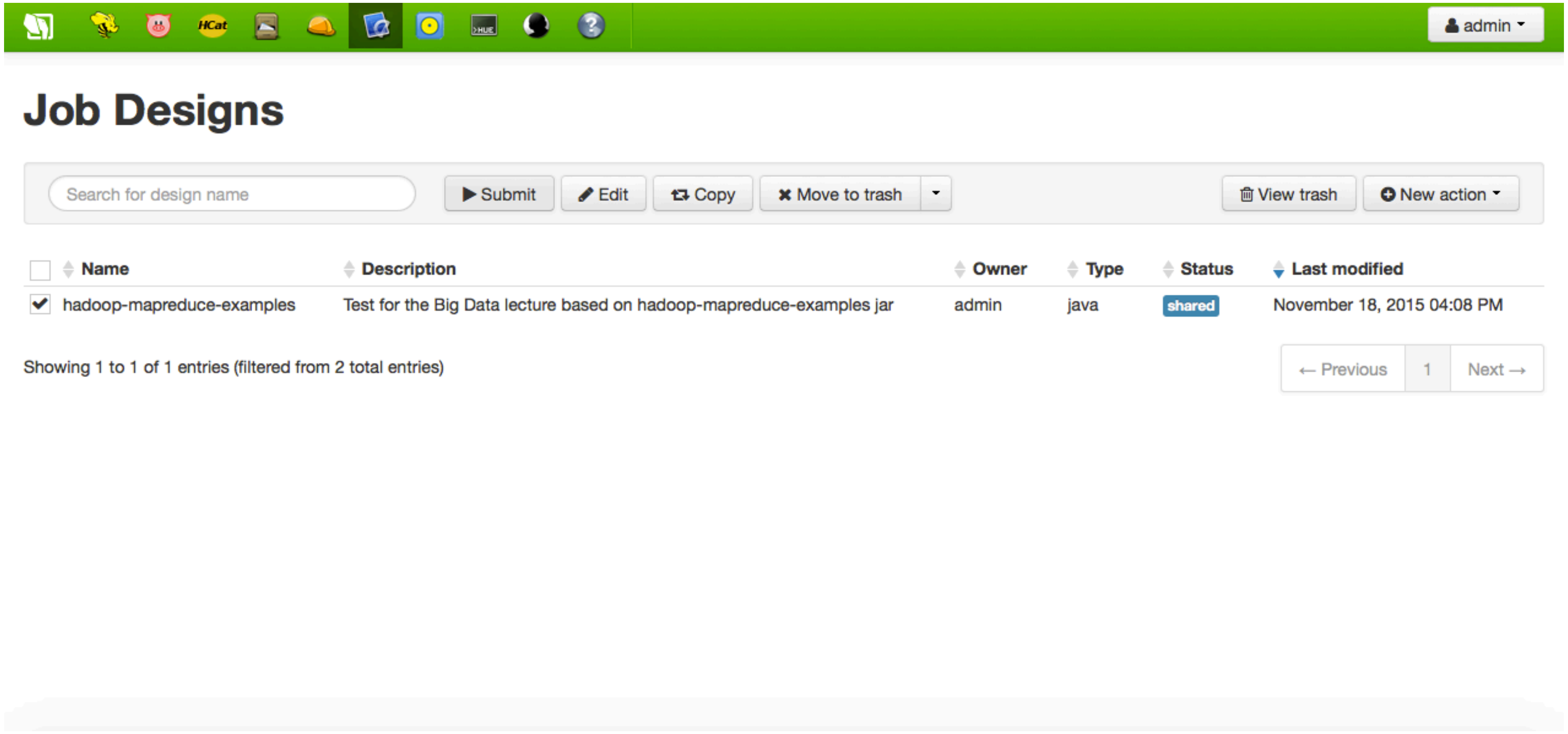
Name	hadoop-mapreduce-examples
Description	ıadoop-mapreduce-examples jar

advanced

You can parameterize the values, using `${myVar}`. When the design is submitted, you will be prompted for the actual value of `myVar`.

Jar path	p-mapreduce-examples-2.2.0.jar
Main class	he.hadoop.examples.WordCount
Args	ser/admin/demo_vlci/data/output
Java opts	
Job properties	Add property

MapReduce through Hue (submit)



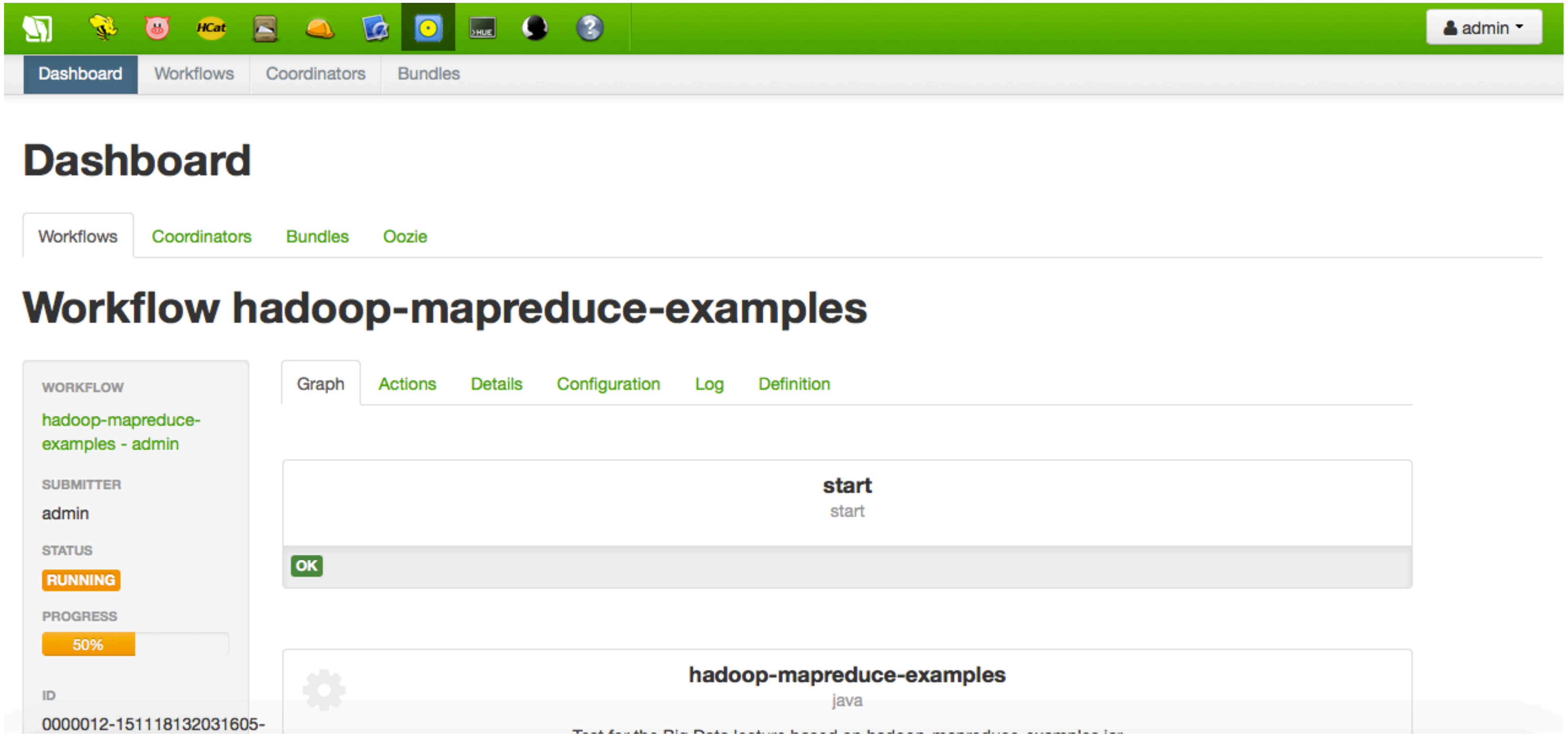
Job Designs

	Name	Description	Owner	Type	Status	Last modified
✓	hadoop-mapreduce-examples	Test for the Big Data lecture based on hadoop-mapreduce-examples jar	admin	java	shared	November 18, 2015 04:08 PM

Showing 1 to 1 of 1 entries (filtered from 2 total entries)

← Previous 1 Next →

MapReduce through Hue (run)



Dashboard

Workflow hadoop-mapreduce-examples

WORKFLOW
hadoop-mapreduce-examples - admin

SUBMITTER
admin

STATUS
RUNNING

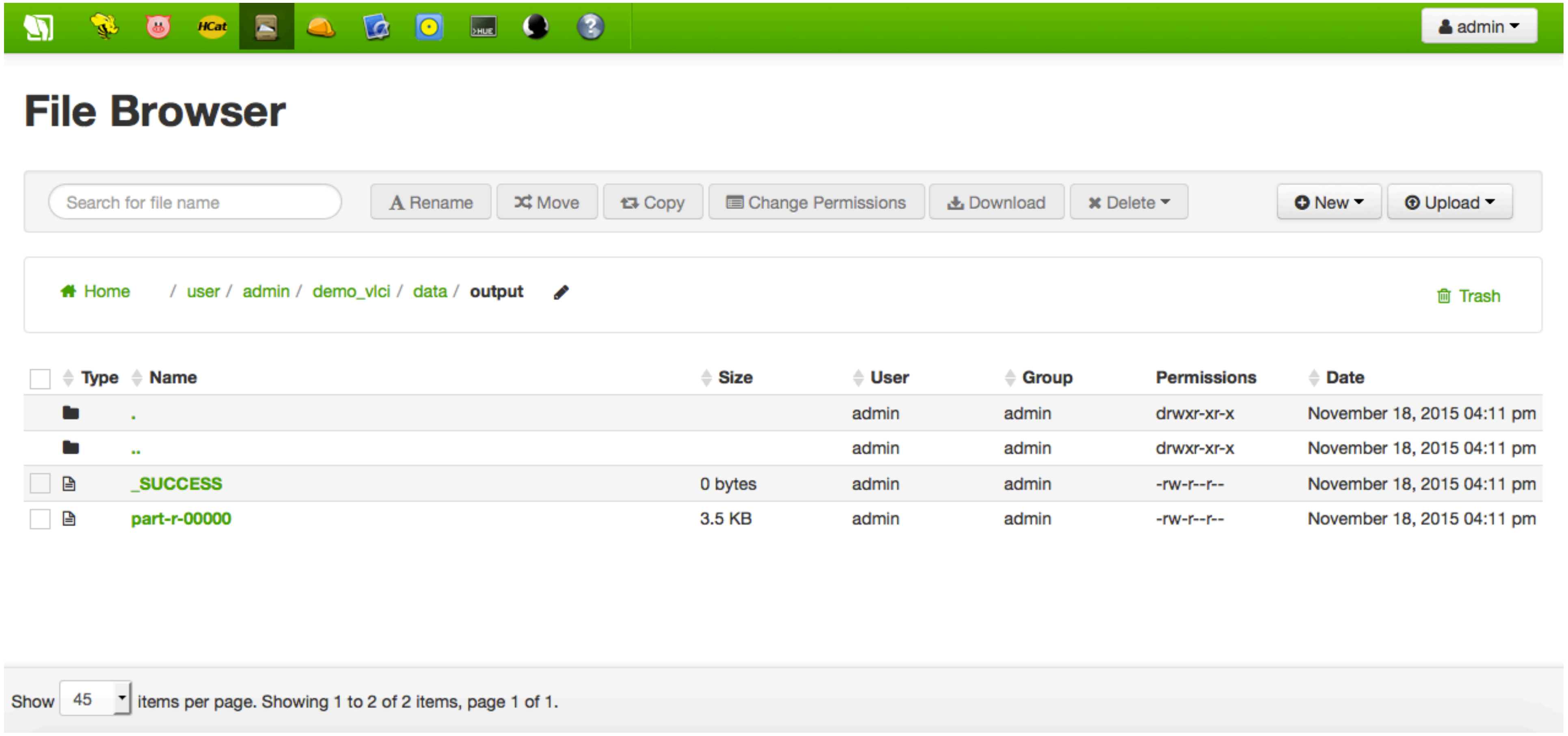
PROGRESS
50%

ID
0000012-151118132031605-

start
start
OK

hadoop-mapreduce-examples
java
Text for the Big Data lecture based on hadoop-mapreduce-examples.jar

MapReduce through Hue (result)



Type	Name	Size	User	Group	Permissions	Date
	.		admin	admin	drwxr-xr-x	November 18, 2015 04:11 pm
	..		admin	admin	drwxr-xr-x	November 18, 2015 04:11 pm
	_SUCCESS	0 bytes	admin	admin	-rw-r--r--	November 18, 2015 04:11 pm
	part-r-00000	3.5 KB	admin	admin	-rw-r--r--	November 18, 2015 04:11 pm

Show 45 items per page. Showing 1 to 2 of 2 items, page 1 of 1.

<u>Batch processing</u>

<u>Simplifying the analysis</u>

Querying tools

Querying tools

- MapReduce paradigm may be hard to understand and, the worst, to use
- Indeed, many data analyzers just need to **query** for the data
  - If possible, by using already well-known languages
- Regarding that, some **querying tools** appeared in the Hadoop ecosystem
  - **Hive** and its **HiveQL** language → quite similar to SQL
  - **Pig** and its **Pig Latin** language → a new language

Hive and HiveQL

- HiveQL reference
  - https://cwiki.apache.org/confluence/display/Hive/LanguageManual
- All the **data is loaded** into Hive tables
  - Not real tables (they don't contain the real data) but metadata pointing to the real data at HDFS
- The best thing is Hive uses **pre-defined MapReduce jobs** behind the scenes!
  - Column selection
  - Fields grouping
  - Table joining
  - Values filtering
  - …
- **Important remark**: since MapReduce is used by Hive, the queries make take some time to produce a result

Hive CLI

```
$ hive
hive history file=/tmp/myuser/
hive_job_log_opendata_XXX_XXX.txt
hive>select column1,column2,otherColumns from mytable where
column1='whatever' and columns2 like '%whatever%';
Total MapReduce jobs = 1
Launching Job 1 out of 1
Starting Job = job_201308280930_0953, Tracking URL = http://
cosmosmaster-gi:50030/jobdetails.jsp?
jobid=job_201308280930_0953
Kill Command = /usr/lib/hadoop/bin/hadoop job  -
Dmapred.job.tracker=cosmosmaster-gi:8021 -kill
job_201308280930_0953
2013-10-03 09:15:34,519 Stage-1 map = 0%,  reduce = 0%
2013-10-03 09:15:36,545 Stage-1 map = 67%,  reduce = 0%
2013-10-03 09:15:37,554 Stage-1 map = 100%,  reduce = 0%
2013-10-03 09:15:44,609 Stage-1 map = 100%,  reduce = 33%
```

Hive through Hue (show table)

- Beeswax can be installed in Hue
  - **Web-based CLI**

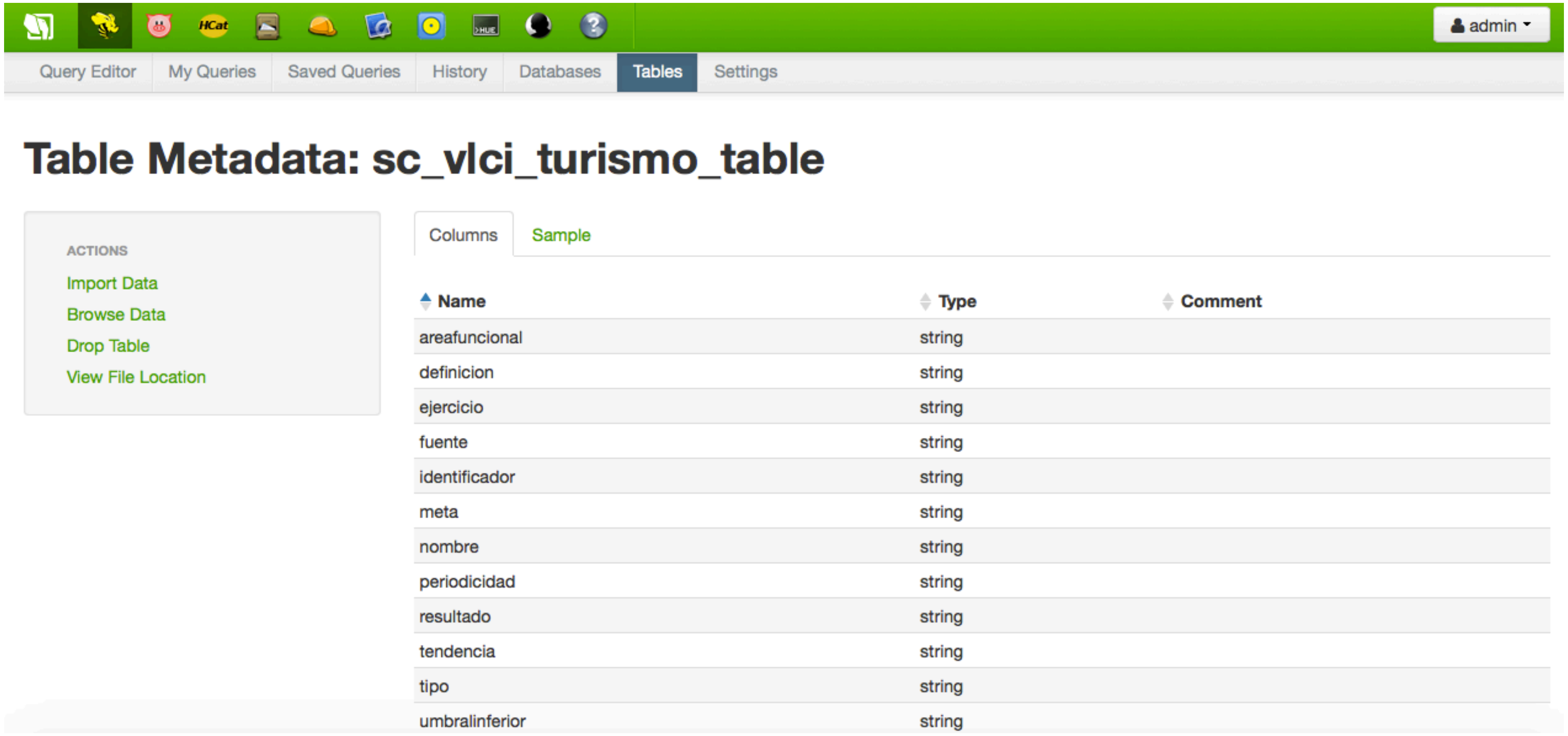
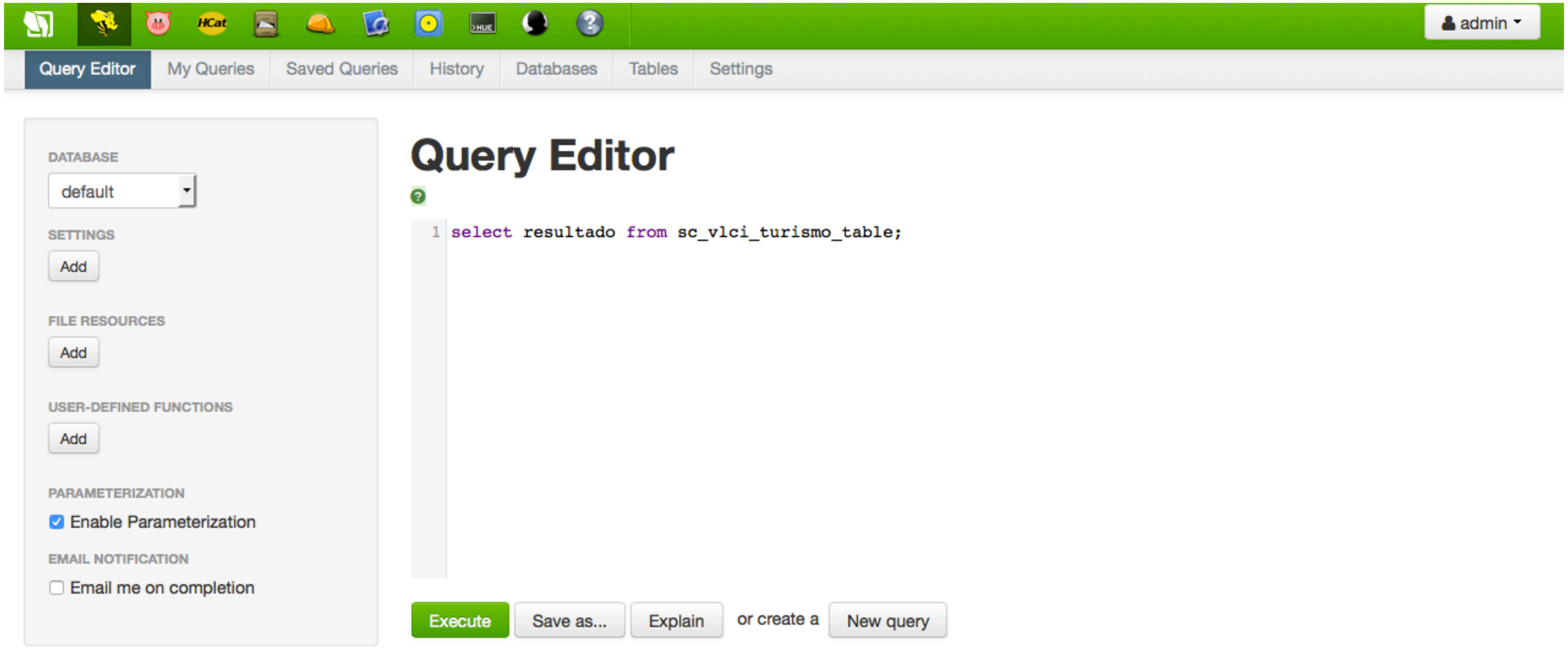


Table Metadata: sc_vlci_turismo_table

Name	Type	Comment
areafuncional	string	
definicion	string	
ejercicio	string	
fuente	string	
identificador	string	
meta	string	
nombre	string	
periodicidad	string	
resultado	string	
tendencia	string	
tipo	string	
umbralinferior	string	

Hive through Hue (query editor)



Query Editor

DATABASE
default

SETTINGS
Add

FILE RESOURCES
Add

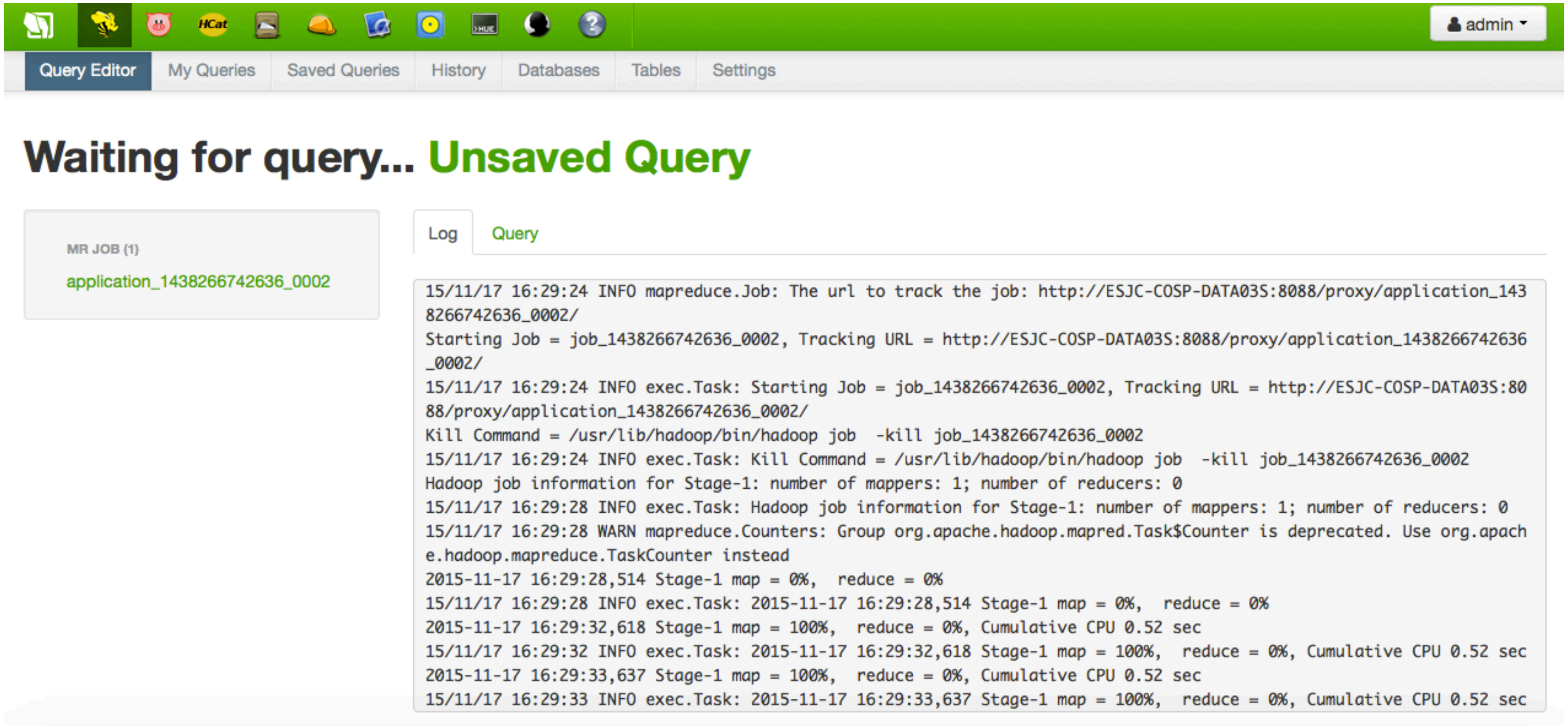
USER-DEFINED FUNCTIONS
Add

PARAMETERIZATION
☑ Enable Parameterization

EMAIL NOTIFICATION
☐ Email me on completion

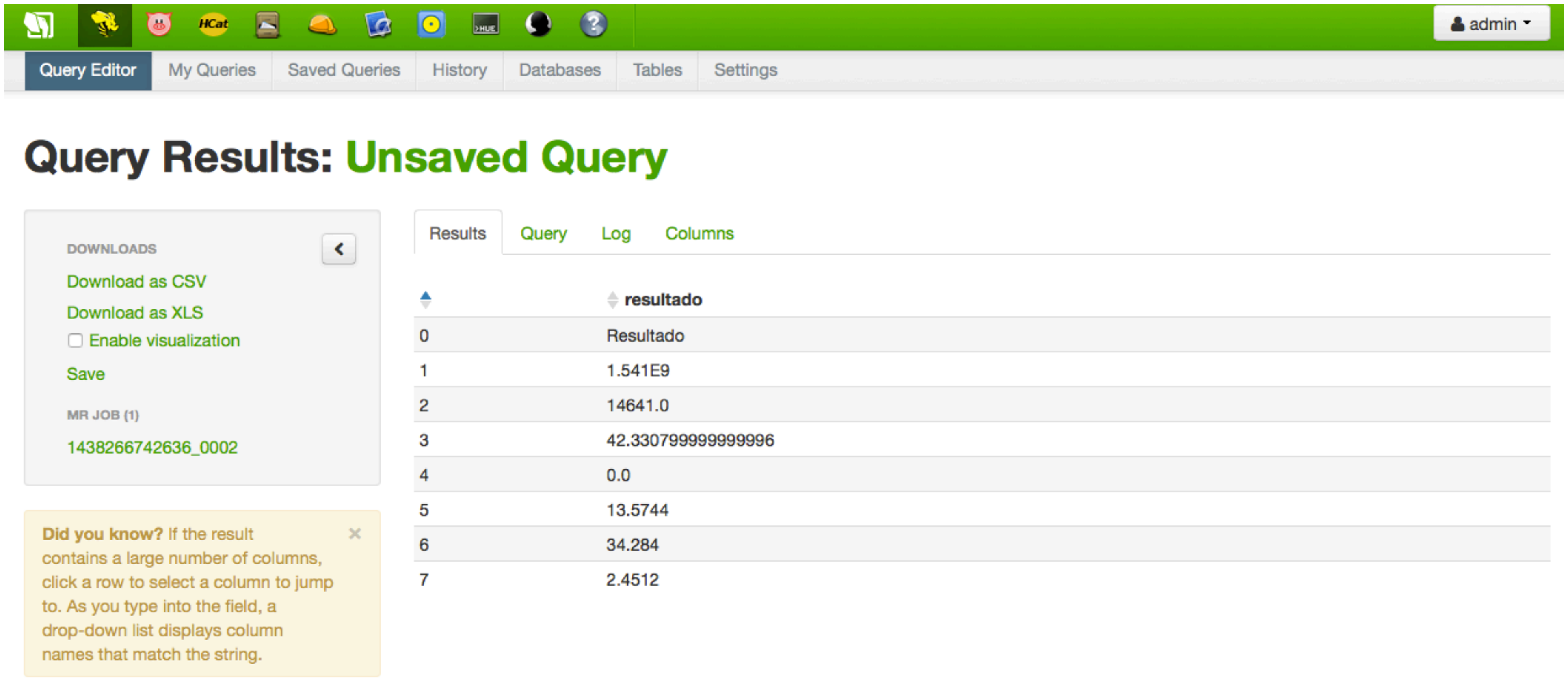
```
select resultado from sc_vlci_turismo_table;
```

Hive through Hue (execute query)



```
15/11/17 16:29:24 INFO mapreduce.Job: The url to track the job: http://ESJC-COSP-DATA03S:8088/proxy/application_143
8266742636_0002/
Starting Job = job_1438266742636_0002, Tracking URL = http://ESJC-COSP-DATA03S:8088/proxy/application_1438266742636
_0002/
15/11/17 16:29:24 INFO exec.Task: Starting Job = job_1438266742636_0002, Tracking URL = http://ESJC-COSP-DATA03S:80
88/proxy/application_1438266742636_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1438266742636_0002
15/11/17 16:29:24 INFO exec.Task: Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1438266742636_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
15/11/17 16:29:28 INFO exec.Task: Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
15/11/17 16:29:28 WARN mapreduce.Counters: Group org.apache.hadoop.mapred.Task$Counter is deprecated. Use org.apach
e.hadoop.mapreduce.TaskCounter instead
2015-11-17 16:29:28,514 Stage-1 map = 0%,  reduce = 0%
15/11/17 16:29:28 INFO exec.Task: 2015-11-17 16:29:28,514 Stage-1 map = 0%,  reduce = 0%
2015-11-17 16:29:32,618 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.52 sec
15/11/17 16:29:32 INFO exec.Task: 2015-11-17 16:29:32,618 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.52 sec
2015-11-17 16:29:33,637 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.52 sec
15/11/17 16:29:33 INFO exec.Task: 2015-11-17 16:29:33,637 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.52 sec
```

Hive through Hue (result)



Query Results: Unsaved Query

	resultado
0	Resultado
1	1.541E9
2	14641.0
3	42.330799999999996
4	0.0
5	13.5744
6	34.284
7	2.4512

Did you know? If the result contains a large number of columns, click a row to select a column to jump to. As you type into the field, a drop-down list displays column names that match the string.

Hive Java API

- Hive CLI and Hue are OK for human-driven testing purposes
  - But it **is not usable by remote applications**
- Hive has **no REST API**
- Hive has several **drivers and libraries**
  - JDBC for Java
  - Python
  - PHP
  - ODBC for C/C++
  - Thrift for Java and C++
  - https://cwiki.apache.org/confluence/display/Hive/HiveClient
- A remote Hive client usually performs:
  - A **connection** to the Hive server (TCP/10000)
  - The **query execution**

Hive Java API: get a connection

```
private static Connection getConnection(String ip, String port,
      String user, String password) {
   try {
      Class.forName("org.apache.hadoop.hive.jdbc.HiveDriver");
   } catch (ClassNotFoundException e) {
      System.out.println(e.getMessage());
      return null;
   } // try catch

   try {
      return DriverManager.getConnection("jdbc:hive://" + ip
         + ":" + port + "/default?user=" + user + "&password="
         + password);
   } catch (SQLException e) {
      System.out.println(e.getMessage());
      return null;
   } // try catch
} // getConnection
```

Hive Java API: do the query

```
private static void doQuery() {
   try {
      Statement stmt = con.createStatement();
      ResultSet res = stmt.executeQuery(
         "select column1,column2,"
         + "otherColumns from mytable where “
         + “column1='whatever' and “
         + "columns2 like '%whatever%'");

      while (res.next()) {
         String column1 = res.getString(1);
         Integer column2 = res.getInteger(2);
      } // while

      res.close(); stmt.close(); con.close();
   } catch (SQLException e) {
      System.exit(0);
   } // try catch
} // doQuery
```

FIWARE

Hive tables creation

- Both locally using the CLI, or remotely using the Java API, use this command:

```
create [external] table...
```

- **CSV**-like HDFS files

```
create external table <table_name> (<field1_name>
<field1_type>, ..., <fieldN_name> <fieldN_type>) row format
delimited fields terminated by '<separator>' location '/user/
<username>/<path>/<to>/<the>/<data>';
```

- **Json**-like HDFS files

```
create external table <table_name> (<field1_name>
<field1_type>, ..., <fieldN_name> <fieldN_type>) row format
serde 'org.openx.data.jsonserde.JsonSerDe' location '/user/
<username>/<path>/<to>/<the>/<data>';
```

Alternatives to Hadoop

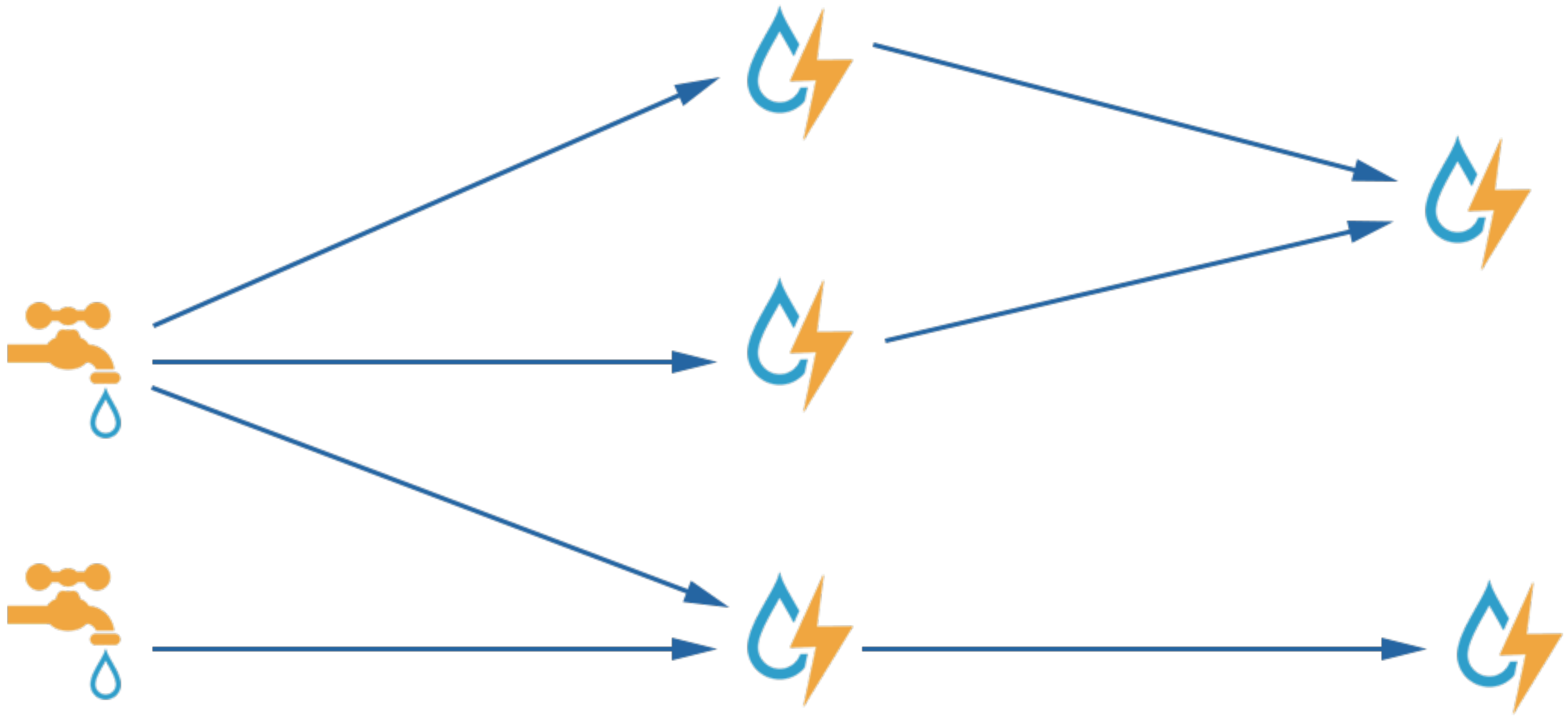
- **Spark**
  - Batches of data fitting into memory
    - Faster than Hadoop (if using the same data file)
  - Can read files from HDFS
  - Mesos or YARN (from Hadoop ecosystem) as resource manager
  - Scalable and fault tolerant
  - API with pre-defined functions
  - http://spark.apache.org
- **Flink**
  - Natively designed for stream analysis
    - Supports for batch processing as well
  - YARN (from Hadoop ecosystem) as resource manager
  - Can be deployed in Amazon Elastic Cloud Computing or Google Compute engine
  - Scalable and fault tolerant
  - API with pre-defined functions
  - https://flink.apache.org

Stream processing:

The Storm reference

Storm project

- Created by Natham Marz at BackType/Twitter
- Distributed realtime computation system



Storm project

https://commons.wikimedia.org/wiki/File:Ford_assembly_line_-_1913.jpg

> Pipeline at Ford industries

Storm basics

- Based on processing building blocks that can be composed in a topology
  - **Spouts**: blocks in charge of polling for data streams, producing data tuples
  - **Bolts**: blocks in charge of processing data tuples, performing basic operations
    - 1:1 operations: arithmetics, transformations…
    - N:1 operations: filtering, joining…
    - 1:N operations: spliting, replication…
- It is **scalable** and **fault-tolerant**
  - A basic operation can be replicated many times in a layer of bolts
  - If a bolt fails, there are serveral other bolts performing the same basic operation in the layer
- Guarantees the data will be processed
  - Storm perform an **ACK** mechanism for data tuples

Alternatives to Storm

- **Spark streaming**
  - Originally designed for batches of data fitting into memory
    - Stream processing uses mini-batches
  - Can read files from Flume, Kafka, AWS Kinesis, Twitter…
  - Mesos or YARN (from Hadoop ecosystem) as resource manager
  - Scalable and fault tolerant
  - API with pre-defined functions
  - http://spark.apache.org
- **Samza**
  - Highly tight to Kafka
  - YARN (from Hadoop ecosystem) as resource manager
  - Scalable and fault tolerant
  - Basic API designed for user-defined functions (as Hadoop does)
  - http://samza.apache.org
- **Flink**
  - Natively designed for stream analysis
  - YARN (from Hadoop ecosystem) as resource manager
  - Can be deployed in Amazon Elastic Cloud Computing or Google Compute engine
  - Scalable and fault tolerant
  - API with pre-defined functions
  - https://flink.apache.org

Thank you!

http://fiware.org
Follow @FIWARE on Twitter

FIWARE