

Introduction to Bigdata

Venkat Reddy

Contents

- What is Bigdata
- Sources of Bigdata
- What can be done with Big data?
- Handling Bigdata
- MapReduce
- Hadoop
- Hadoop components
- Hadoop ecosystem
- Big data example
- Other bigdata use cases

How much time did it take?

- Excel : Have you ever tried a pivot table on 500 MB file?
- SAS/R : Have you ever tried a frequency table on 2 GB file?
- Access: Have you ever tried running a query on 10 GB file
- SQL: Have you ever tried running a query on 50 GB file



Can you think of...

- Can you think of running a query on 20,980,000 GB file.
 - What if we get a new data set like this, every day?
 - What if we need to execute complex queries on this data set everyday ?
 - Does anybody really deal with this type of data set?
 - Is it possible to store and analyze this data?
- Yes google deals with more than 20 PB data everyday



Yes....its true

- Google processes 20 PB a day (2008)
- Way back Machine has 3 PB + 100 TB/month (3/2009)
- Facebook has 2.5 PB of user data + 15 TB/day (4/2009)
- eBay has 6.5 PB of user data + 50 TB/day (5/2009)
- CERN's Large Hydron Collider (LHC) generates 15 PB a year



That's right

In fact, in a minute...

- **Email** users send more than 204 million messages;
- **Mobile Web** receives 217 new users;
- **Google** receives over 2 million search queries;
- **YouTube** users upload 48 hours of new video;
- **Facebook** users share 684,000 bits of content;
- **Twitter** users send more than 100,000 tweets;
- **Consumers** spend \$272,000 on Web shopping;
- **Apple** receives around 47,000 application downloads;
- **Brands** receive more than 34,000 Facebook 'likes';
- **Tumblr** blog owners publish 27,000 new posts;
- **Instagram** users share 3,600 new photos;
- **Flickr** users, on the other hand, add 3,125 new photos;
- **Foursquare** users perform 2,000 check-ins;
- **WordPress** users publish close to 350 new blog posts.

And this is one year back..... Damn!!

What is a large file?

- If you are using a 32 bit OS then 4GB is a large file
 - Traditionally, many operating systems and their underlying file system implementations used 32-bit integers to represent file sizes and positions. Consequently no file could be larger than $2^{32}-1$ bytes (4 GB).
 - In many implementations the problem was exacerbated by treating the sizes as signed numbers, which further lowered the limit to $2^{31}-1$ bytes (2 GB).
 - Files larger than this, too large for 32-bit operating systems to handle, came to be known as large files.

What the ...

Definition of Bigdata

Sorry ...There is no single standard definition...



Bigdata ...

Any data that is **difficult** to

- Capture
- Curate
- Store
- Search
- Share
- Transfer
- Analyze
- and to create visualizations

Bigdata means

- Collection of data sets so large and **complex** that it becomes **difficult to process** using on-hand database management tools or traditional data processing applications
- “Big Data” is the data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it...

BTW is it Bigdata/big data/Big data/bigdata/BigData /Big Data?

Bigdata is not just about size

- Volume
 - Data volumes are becoming unmanageable
- Variety
 - Data complexity is growing. more types of data captured than previously
- Velocity
 - Some data is arriving so rapidly that it must either be processed instantly, or lost. This is a whole subfield called “stream processing”

Types of data

- Relational Data (Tables/Transaction/Legacy Data)
- Text Data (Web)
- Semi-structured Data (XML)
- Graph Data
 - Social Network, Semantic Web (RDF), ...
- Streaming Data
 - You can only scan the data once
- Text, numerical, images, audio, video, sequences, time series, social media data, multi-dim arrays, etc...

What can be done with Bigdata?

- Social media brand value analytics
- Product sentiment analysis
- Customer buying preference predictions
- Video analytics
- Fraud detection
- Aggregation and Statistics
 - Data warehouse and OLAP
- Indexing, Searching, and Querying
 - Keyword based search
 - Pattern matching (XML/RDF)
- Knowledge discovery
 - Data Mining
 - Statistical Modeling

Ok..... Analysis on this bigdata can give us awesome insights

But, datasets are huge, complex and difficult to process

What is the solution?

Handling bigdata- Parallel computing

- Imagine a 1gb text file, all the status updates on Facebook in a day
- Now suppose that a simple counting of the number of rows takes 10 minutes.
 - `Select count(*) from fb_status`
- What do you do if you have 6 months data, a file of size 200GB, if you still want to find the results in 10 minutes?
- Parallel computing?
 - Put multiple CPUs in a machine (100?)
 - Write a code that will calculate 200 parallel counts and finally sums up
 - But you need a super computer



Handling bigdata – Is there a better way?

- Till 1985, There is no way to connect multiple computers. All systems were Centralized Systems.
 - So multi-core system or super computers were the only options for big data problems
- After 1985, We have powerful microprocessors and High Speed Computer Networks (LANs , WANs), which lead to distributed systems
- Now that we have a distributed system that ensures a collection of independent computers appears to its users as a single coherent system, can we use some cheap computers and process our bigdata quickly?

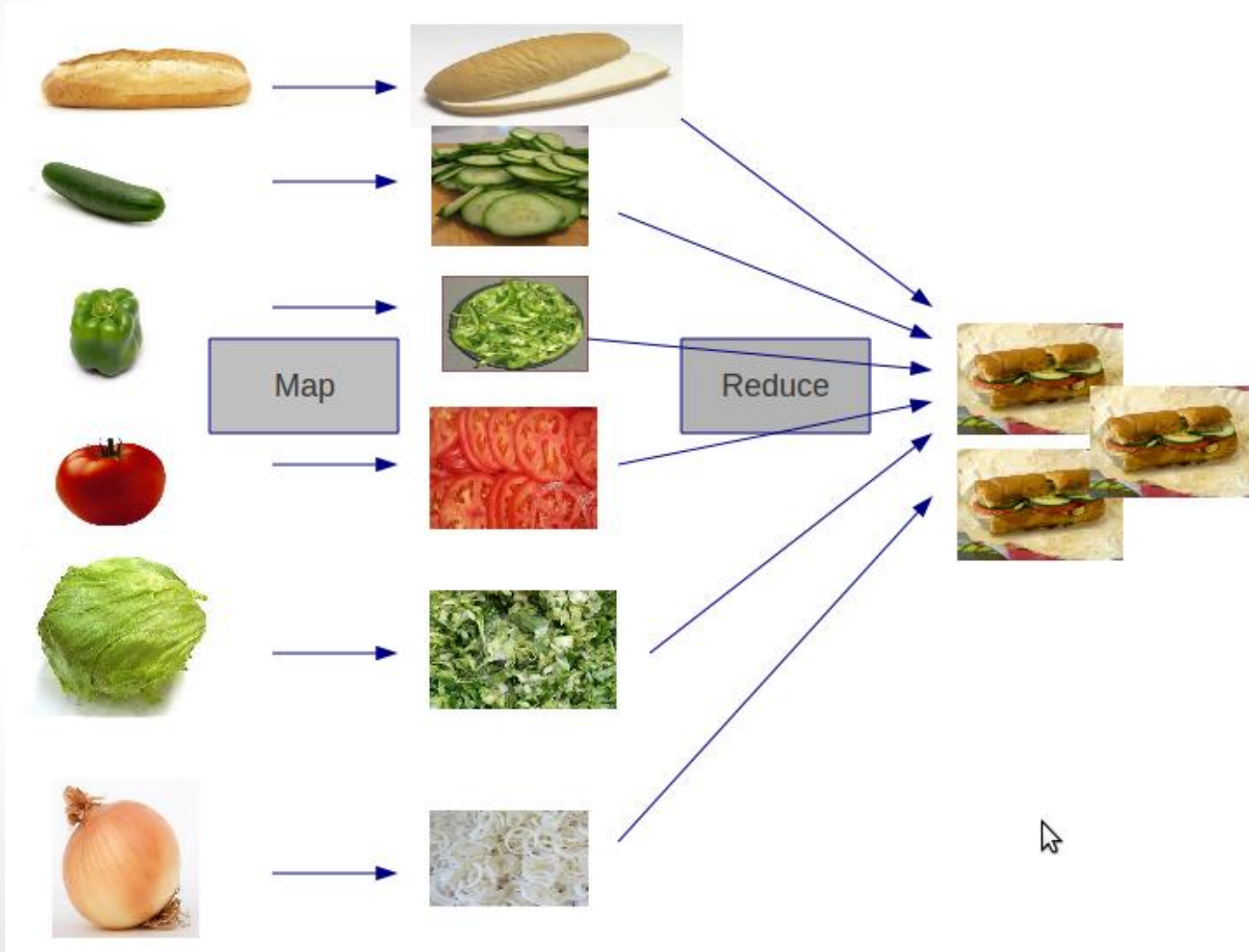
Distributed computing

- We want to cut the data into small pieces & place them on different machines
- Divide the overall problem into small tasks & run these small tasks locally
- Finally collate the results from local machines
- So, we want to process our bigdata in a parallel programming model and associated implementation.
- This is known as **MapReduce**

MapReduce.... Programming Model

- Processing data using special map() and reduce() functions
- The map() function is called on every item in the input and emits a series of intermediate key/value pairs(Local calculation)
- All values associated with a given key are grouped together
- The reduce() function is called on every unique key, and its value list, and emits a value that is added to the output(final organization)

Mummy 's MapReduce



Not just MapReduce

- Earlier `count=count+1` was sufficient but now, we need to
 1. Setup a cluster of machines, then divide the whole data set into blocks and store them in local machines
 2. Assign a master node that takes charge of all meta data, work scheduling and distribution, and job orchestration
 3. Assign worker slots to execute map or reduce functions
 4. Load Balance (What if one machine is very slow in the cluster?)
 5. Fault Tolerance (What if the intermediate data is partially read, but the machine fails before all reduce(collation) operations can complete?)
 6. Finally write the map reduce code that solves our problem

Ok..... Analysis on bigdata can give us awesome insights

But, datasets are huge, complex and difficult to process

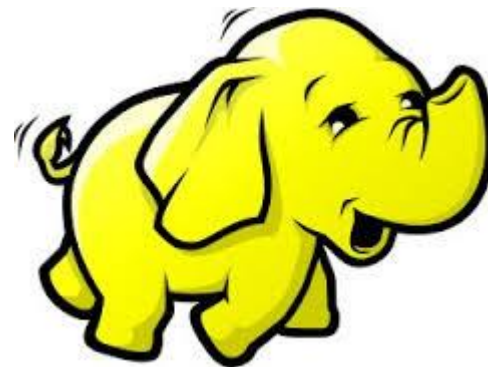
I found a solution, distributed computing or MapReduce

But looks like this data storage & parallel processing is complicated

What is the solution?

Hadoop

- Hadoop is a bunch of tools, it has many components. HDFS and MapReduce are two core components of Hadoop
 - HDFS: Hadoop Distributed File System
 - makes our job easy to store the data on commodity hardware
 - Built to expect hardware failures
 - Intended for large files & batch inserts
 - MapReduce
 - For parallel processing
- So Hadoop is a software platform that lets one easily write and run applications that process bigdata



Why Hadoop is useful

- **Scalable:** It can reliably store and process petabytes.
- **Economical:** It distributes the data and processing across clusters of commonly available computers (in thousands).
- **Efficient:** By distributing the data, it can process it in parallel on the nodes where the data is located.
- **Reliable:** It automatically maintains multiple copies of data and automatically redeploys computing tasks based on failures.
- And Hadoop is free

So what is Hadoop?

- Hadoop is not Bigdata
- Hadoop is not a database
- Hadoop is a platform/framework
 - Which allows the user to quickly write and test distributed systems
 - Which is efficient in automatically distributing the data and work across machines

Ok..... Analysis on bigdata can give us awesome insights

But, datasets are huge, complex and difficult to process

I found a solution, distributed computing or MapReduce

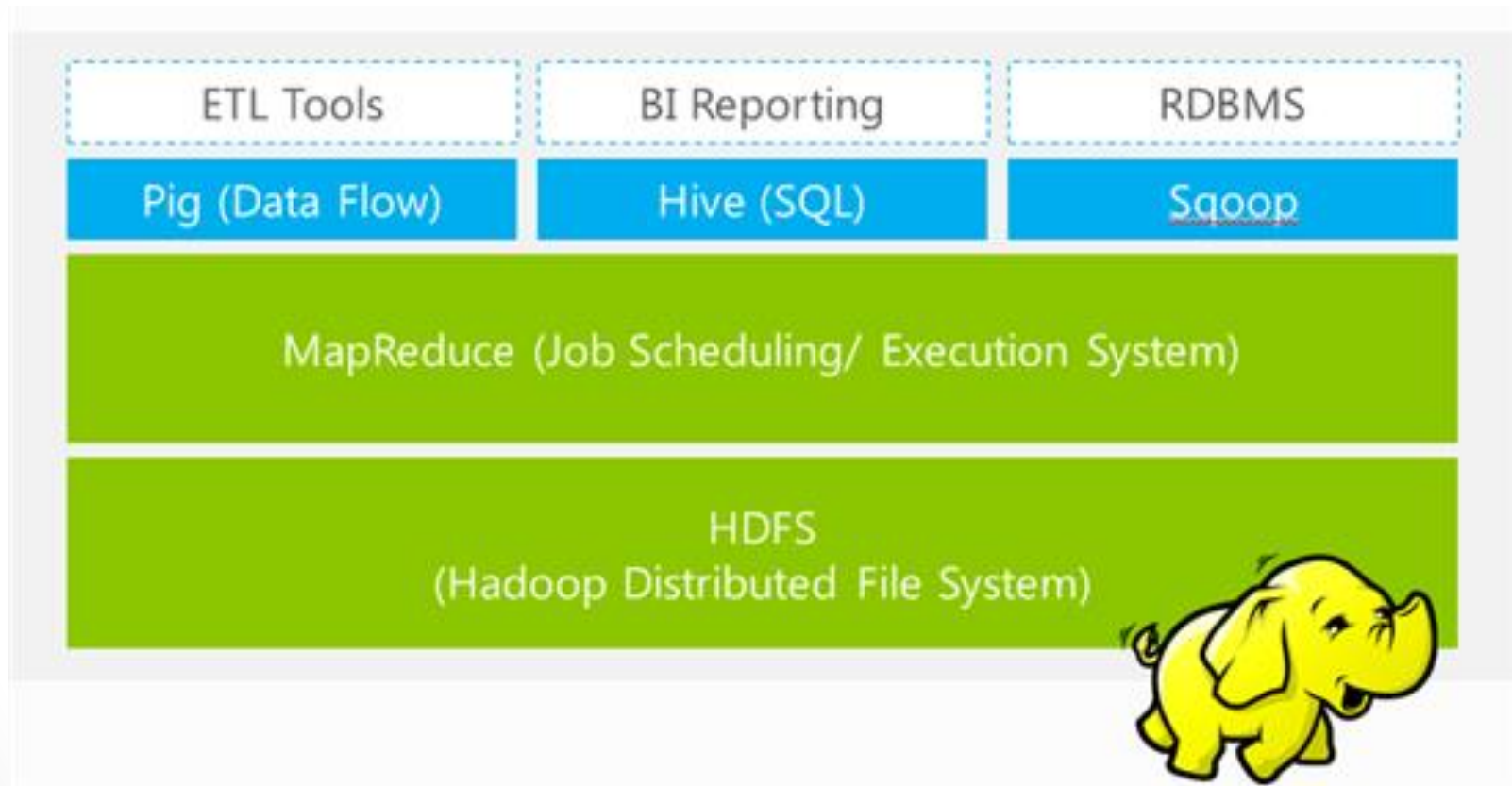
But looks like this data storage & parallel processing is complicated

Ok, I can use Hadoop framework.....I don't know Java, how do I write MapReduce programs?

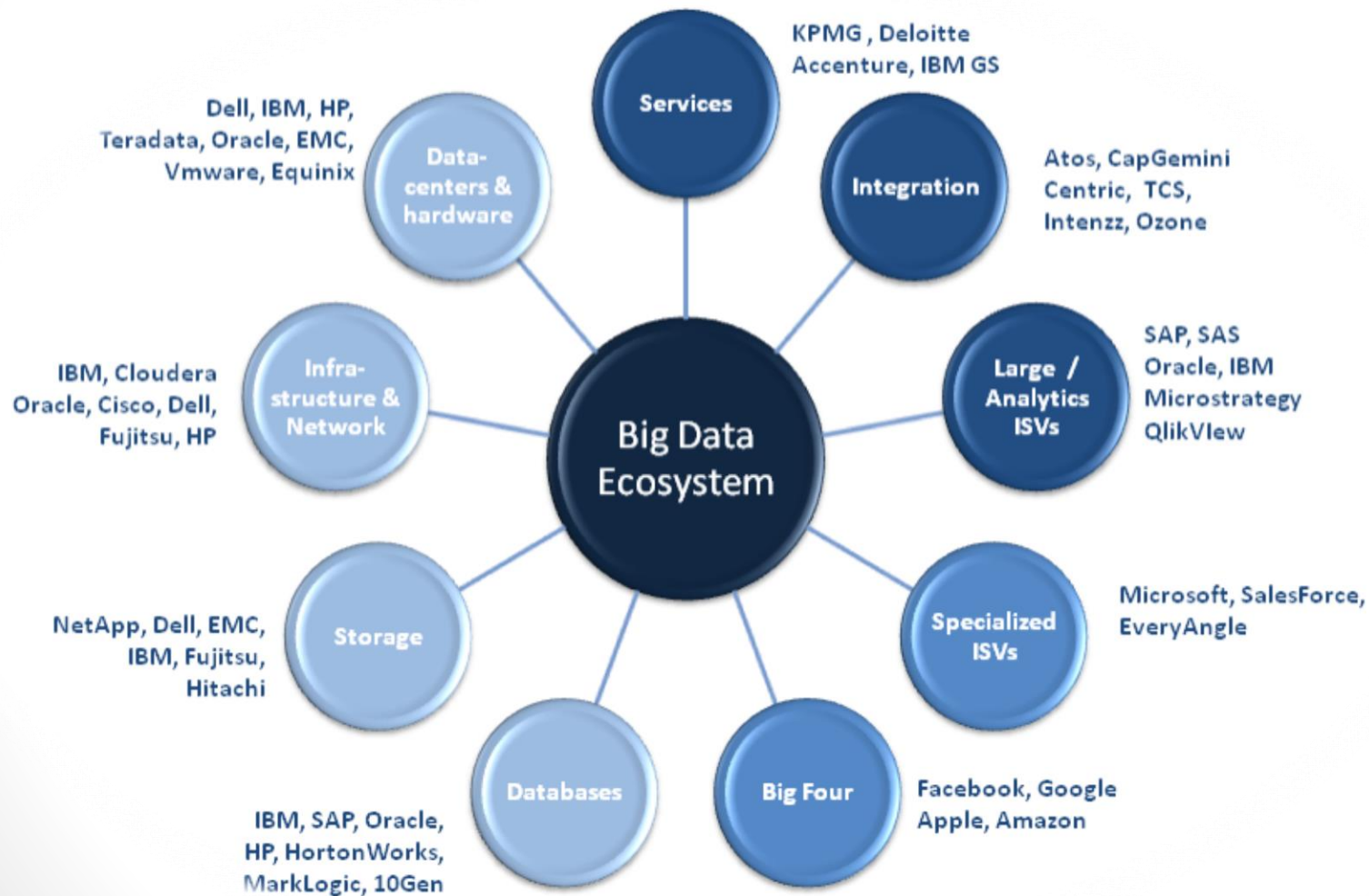
MapReduce made easy

- **Hive:**
 - Hive is for data analysts with strong SQL skills providing an SQL-like interface and a relational data model
 - Hive uses a language called HiveQL; very similar to SQL
 - Hive translates queries into a series of MapReduce jobs
- **Pig:**
 - Pig is a high-level platform for processing big data on Hadoop clusters.
 - Pig consists of a data flow language, called Pig Latin, supporting writing queries on large datasets and an execution environment running programs from a console
 - The Pig Latin programs consist of dataset transformation series converted under the covers, to a MapReduce program series
- **Mahout**
 - Mahout is an open source machine-learning library facilitating building scalable machine learning libraries

Hadoop ecosystem



Bigdata ecosystem



Bigdata example

- **The Business Problem:**

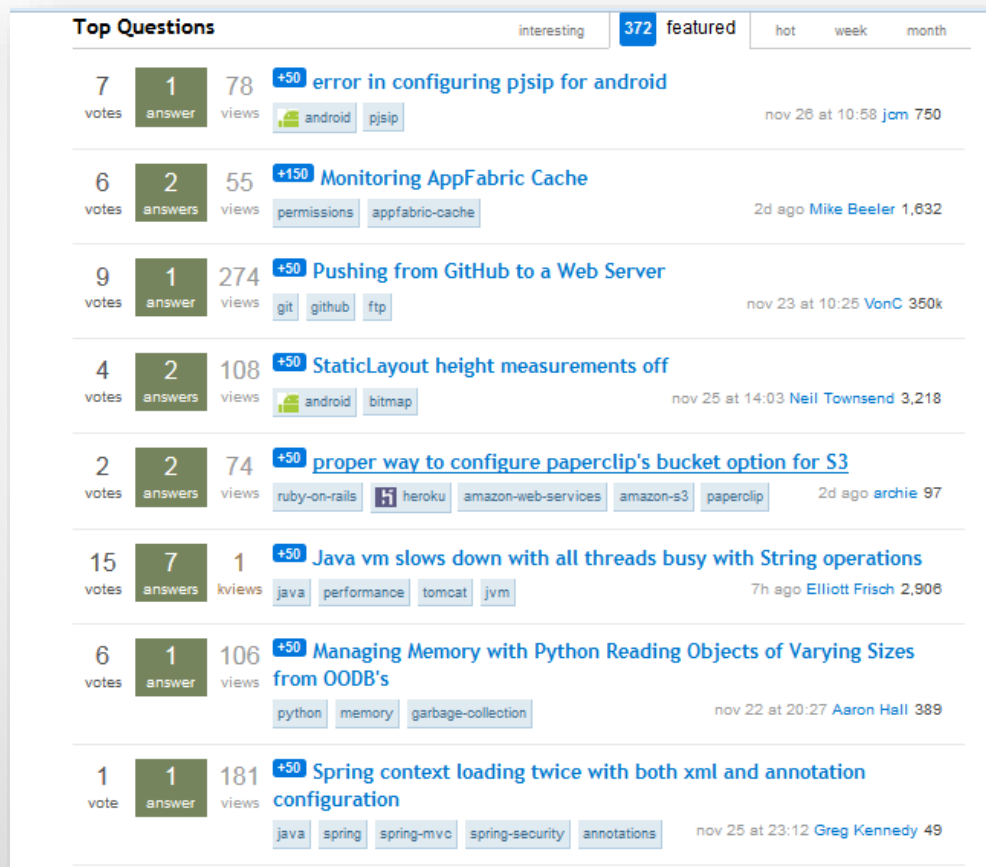
- Analyze this week's stack overflow data <http://stackoverflow.com/>
- What are the most popular topics in this week?

- **Approach:**

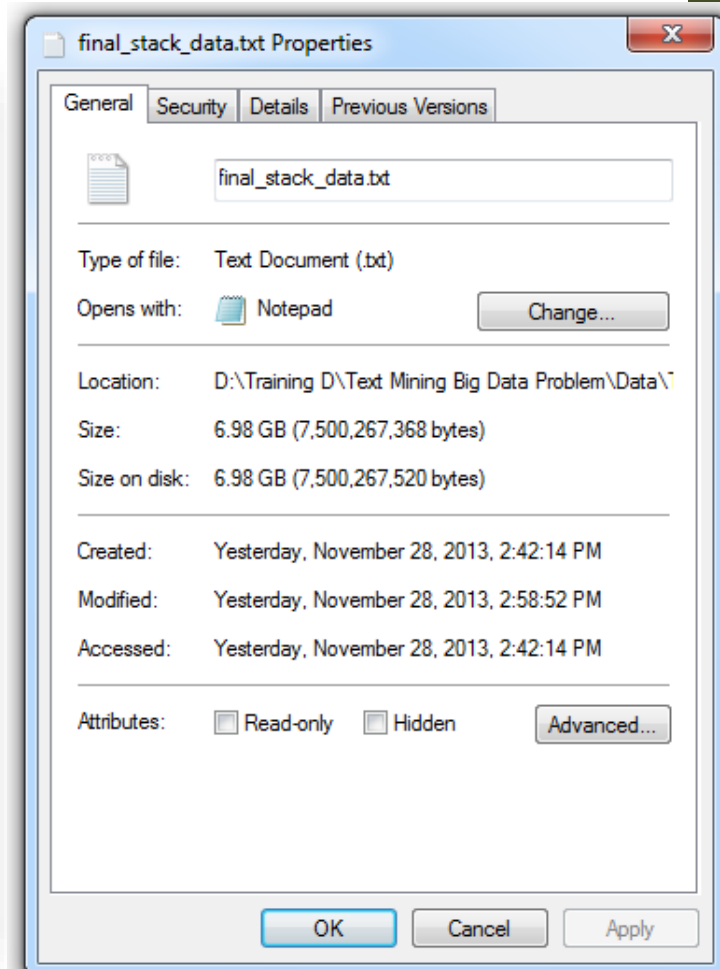
- Find out some simple descriptive statistics for each field
 - Total questions
 - Total unique tags
 - Frequency of each tag etc.,
- The 'tag' with max frequency is the most popular topic
- Lets use Hadoop to find these values, since we can't rapidly process this data with usual tools

Bigdata example: Dataset

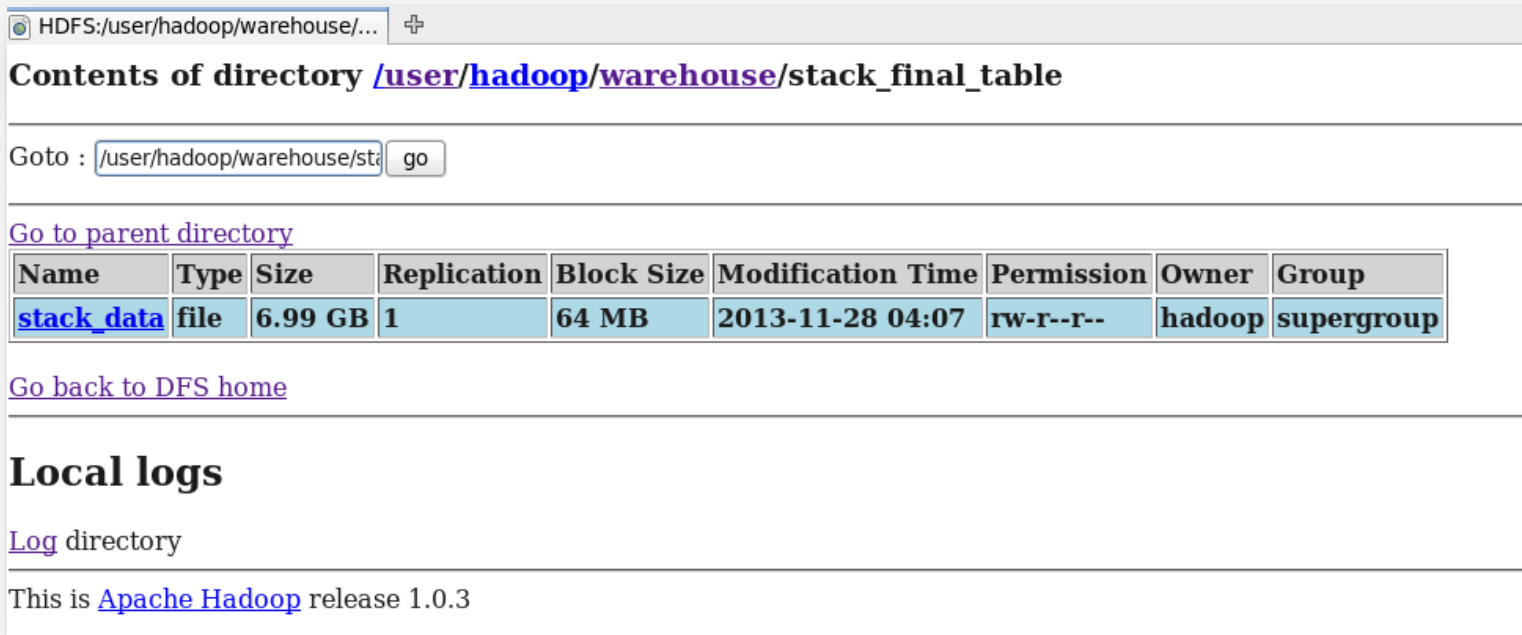
7GB text file, contains questions and respective tags



Rank	Votes	Answers	Views	Question Title	Tags	Timestamp	Author	Score
7	1	78	150	error in configuring pjsip for android	android, pjsip	nov 26 at 10:58	jcm	750
6	2	55	150	Monitoring AppFabric Cache	permissions, appfabric-cache	2d ago	Mike Beeler	1,632
9	1	274	150	Pushing from GitHub to a Web Server	git, github, ftp	nov 23 at 10:25	VonC	350k
4	2	108	150	StaticLayout height measurements off	android, bitmap	nov 25 at 14:03	Neil Townsend	3,218
2	2	74	150	proper way to configure paperclip's bucket option for S3	ruby-on-rails, heroku, amazon-web-services, amazon-s3, paperclip	2d ago	archie	97
15	7	1	150	Java vm slows down with all threads busy with String operations	java, performance, tomcat, jvm	7h ago	Elliott Frisch	2,906
6	1	106	150	Managing Memory with Python Reading Objects of Varying Sizes from OODB's	python, memory, garbage-collection	nov 22 at 20:27	Aaron Hall	389
1	1	181	150	Spring context loading twice with both xml and annotation configuration	java, spring, spring-mvc, spring-security, annotations	nov 25 at 23:12	Greg Kennedy	49



Move the dataset to HDFS



HDFS:/user/hadoop/warehouse/...

Contents of directory [/user/hadoop/warehouse/stack_final_table](#)

Goto :

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
stack_data	file	6.99 GB	1	64 MB	2013-11-28 04:07	rw-r--r--	hadoop	supergroup

[Go back to DFS home](#)

Local logs

[Log](#) directory

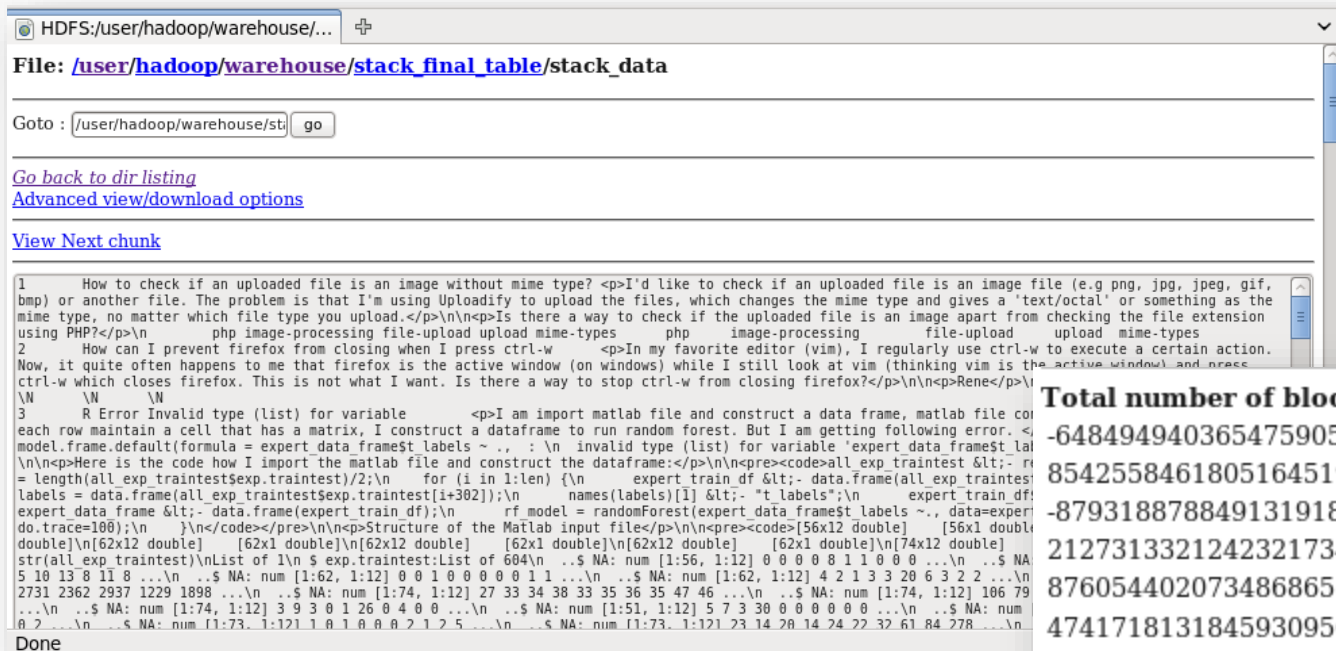
This is [Apache Hadoop](#) release 1.0.3

- The file size is 6.99GB, it has been automatically cut into several pieces/blocks, size of the each block is 64MB
 - This can be done by just using a simple command
- ```
bin/hadoop fs -copyFromLocal /home/final_stack_data stack_data
```

\*Data later copied into Hive table



# Data in HDFS: Hadoop Distributed File System



```
1 How to check if an uploaded file is an image without mime type? <p>I'd like to check if an uploaded file is an image file (e.g png, jpg, jpeg, gif,
 bmp) or another file. The problem is that I'm using Uploadify to upload the files, which changes the mime type and gives a 'text/octal' or something as the
 mime type, no matter which file type you upload.</p><n><p>Is there a way to check if the uploaded file is an image apart from checking the file extension
 using PHP?</p><n> php image-processing file-upload upload mime-types php image-processing file-upload upload mime-types
 2 How can I prevent firefox from closing when I press ctrl-w <p>In my favorite editor (vim), I regularly use ctrl-w to execute a certain action.
 Now, it quite often happens to me that firefox is the active window (on windows) while I still look at vim (thinking vim is the active window) and press
 ctrl-w which closes firefox. This is not what I want. Is there a way to stop ctrl-w from closing firefox?</p><n><p>Rene</p></pre>

```
3 R Error Invalid type (list) for variable <p>I am import matlab file and construct a data frame, matlab file co
  each row maintain a cell that has a matrix, I construct a dataframe to run random forest. But I am getting following error. <
  model.frame.default(formula = expert_data.frame$ labels ~ ., : \n invalid type (list) for variable 'expert_data.frame$ la
  \n<p>Here is the code how I import the matlab file and construct the dataframe:</p><n><pre><code>all_exp_train_test <lt;- r
  = length(all_exp_train_test$exp_train_test)/2;\n for (i in 1:len) {\n expert_train_df <lt;- data.frame(all_exp_train_test
  labels = data.frame(all_exp_train_test$exp_train_test[i+302]);\n names(labels)[1] <lt;- "t_labels";\n expert_train_df:
  expert_data.frame <lt;- data.frame(expert_train_df);\n rf_model = randomForest(expert_data.frame$ labels ~., data=exper
  do.trace=100);\n }\n</code></pre><n><p>Structure of the Matlab input files</p><n><pre><code>[56x12 double] [56x1 double
  double]\n[62x12 double] [62x1 double]\n[62x12 double] [62x1 double]\n[74x12 double]
  str(all_exp_train_test)\nList of 1\n $ exp_train_test:List of 604\n ..$ NA: num [1:56, 1:12] 0 0 0 0 8 1 1 0 0 0 ... \n ..$ NA
  5 10 13 8 11 8 ... \n ..$ NA: num [1:62, 1:12] 0 0 1 0 0 0 0 0 1 1 ... \n ..$ NA: num [1:62, 1:12] 4 2 1 3 3 20 6 3 2 2 ... \n
  2731 2362 2937 1229 1898 ... \n ..$ NA: num [1:74, 1:12] 27 33 34 38 33 35 36 35 47 46 ... \n ..$ NA: num [1:74, 1:12] 106 79
  ... \n ..$ NA: num [1:74, 1:12] 3 9 3 0 1 26 0 4 0 0 ... \n ..$ NA: num [1:51, 1:12] 5 7 3 30 0 0 0 0 0 0 ... \n ..$ NA: num
  0 2 ... \n ..$ NA: num [1:73, 1:12] 1 0 1 0 0 0 2 1 2 5 ... \n ..$ NA: num [1:73, 1:12] 23 14 20 14 24 22 32 61 84 278 ... \n
  Done
```


```

- Each block is 64MB total file size is 7GB, so total 112 blocks

## Total number of blocks: 112


```
-6484949403654759056: 127.0.0.1:50010
8542558461805164519: 127.0.0.1:50010
-8793188788491319186: 127.0.0.1:50010
2127313321242321734: 127.0.0.1:50010
8760544020734868657: 127.0.0.1:50010
4741718131845930950: 127.0.0.1:50010
1729407588028603439: 127.0.0.1:50010
-4557400295273383407: 127.0.0.1:50010
-6397591341018842243: 127.0.0.1:50010
-6464396031503874804: 127.0.0.1:50010
-4972057067362155624: 127.0.0.1:50010
-2937790813048579272: 127.0.0.1:50010
2982722182242888673: 127.0.0.1:50010
-1718996691233975717: 127.0.0.1:50010
-7891369359804788631: 127.0.0.1:50010
7818539273430730078: 127.0.0.1:50010
2196922493023086703: 127.0.0.1:50010
2622023783076943993: 127.0.0.1:50010
```



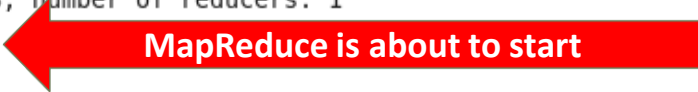
# Processing the data

- What are the total number of entries in this file?

```
File Edit View Search Terminal Help
[hadoop@localhost hadoop-1.0.3]$
[hadoop@localhost hadoop-1.0.3]$ bin/hadoop fs -copyFromLocal /home/hadoop/Desktop/Venkat/TextData/final_stack_data stack
[hadoop@localhost hadoop-1.0.3]$ hive
WARNING: org.apache.hadoop.metrics.jvm.EventCounter is deprecated. Please use org.apache.hadoop.log.metrics.EventCounter
in all the log4j.properties files.
Logging initialized using configuration in jar:file:/home/hadoop/hive-0.9.0/lib/hive-common-0.9.0.jar!/hive-log4j.properties
Hive history file=/tmp/hadoop/hive_job_log_hadoop_201311290600_249742339.txt
hive>
 > select count(*) from stack_final_table;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
 set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
 set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
 set mapred.reduce.tasks=<number>
Starting Job = job_201311290102_0001, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201311290102_0001
Kill Command = /home/hadoop/hadoop-1.0.3/libexec/../bin/hadoop job -Dmapred.job.tracker=localhost:54311 -kill job_201311
290102_0001
Hadoop job information for Stage-1: number of mappers: 28; number of reducers: 1
2013-11-29 06:05:36,089 Stage-1 map = 0%, reduce = 0%
```

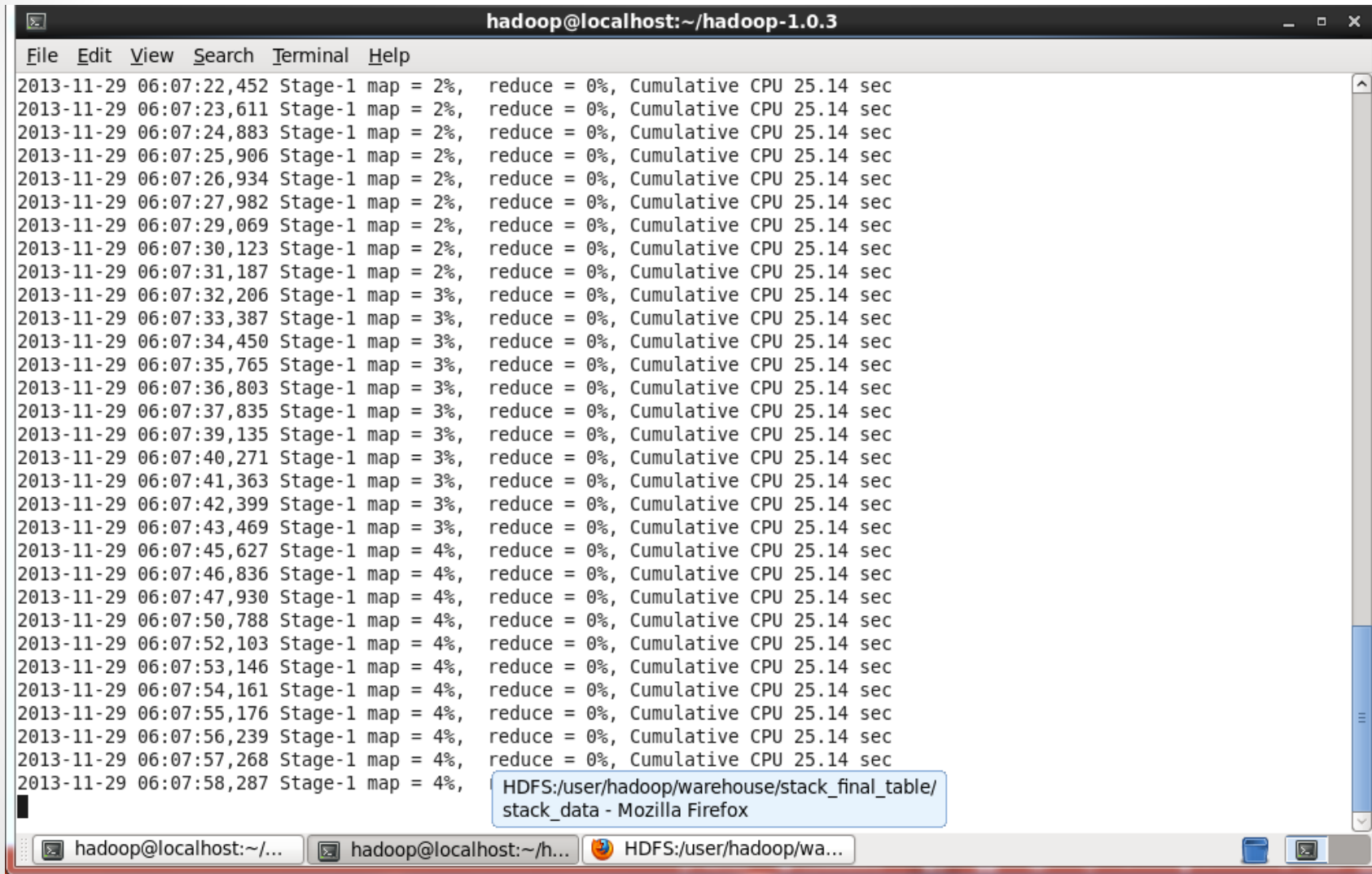


**Here is our query**



**MapReduce is about to start**

# Map reduce jobs in progress





The screenshot shows a terminal window titled "hadoop@localhost:~/hadoop-1.0.3". The window displays a list of Hadoop job progress logs. Each line represents a log entry with a timestamp, a job ID, a stage number, and the progress of map and reduce tasks. The progress is shown as a percentage. The cumulative CPU time for each job is also displayed. The log entries show that the map tasks are progressing, while the reduce tasks are not yet started. The cumulative CPU time for each job is 25.14 seconds. The log entries are as follows:

```
2013-11-29 06:07:22,452 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:23,611 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:24,883 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:25,906 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:26,934 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:27,982 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:29,069 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:30,123 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:31,187 Stage-1 map = 2%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:32,206 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:33,387 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:34,450 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:35,765 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:36,803 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:37,835 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:39,135 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:40,271 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:41,363 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:42,399 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:43,469 Stage-1 map = 3%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:45,627 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:46,836 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:47,930 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:50,788 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:52,103 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:53,146 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:54,161 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:55,176 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:56,239 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:57,268 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
2013-11-29 06:07:58,287 Stage-1 map = 4%, reduce = 0%, Cumulative CPU 25.14 sec
```

The terminal window also shows a prompt "hadoop@localhost:~/hadoop-1.0.3" and a status bar at the bottom with the text "HDFS:/user/hadoop/warehouse/stack\_final\_table/stack\_data - Mozilla Firefox".

# The execution time

```
MapReduce Total cumulative CPU time: 8 minutes 28 seconds 510 msec
Ended Job = job_201311290102_0001
MapReduce Jobs Launched:
Job 0: Map: 28 Reduce: 1 Cumulative CPU: 508.51 sec HDFS Read: 7500817971 HDFS Write: 8 SUCCESS
Total MapReduce CPU Time Spent: 8 minutes 28 seconds 510 msec
OK
6034195
```



- Note: I ran Hadoop on a very basic machine(1.5 GB RAM , i3 processor on,32bit virtual machine).
- This example is just for demo purpose, the same query will take much lesser time, if we are running on a multi node cluster setup

# Bigdata example: Results

- The query returns , means there are nearly 6 million stack overflow questions and tags
- Similarly we can run other map reduce jobs on the tags to find out most frequent topics.
  - 'C' happens to be most popular tag
- It took around 15 minutes to get these insights

# Advanced analytics...

- In the above example, we have the stack overflow questions and corresponding tags
- Can we use some supervised machine learning technique to predict the tags for the new questions?
- Can you write the map reduce code for Naïve Bayes algorithm/Random forest?
- How is Wikipedia highlighting some words in your text as hyperlinks?
- How can YouTube suggest you relevant tags after you upload a video?
- How is amazon recommending you a new product?
- How are the companies leveraging bigdata analytics?

# Bigdata use cases



- Ford collects and aggregates data from the 4 million vehicles that use in-car sensing and remote app management software
- The data allows to glean information on a range of issues, from how drivers are using their vehicles, to the driving environment that could help them improve the quality of the vehicle



- Amazon has been collecting customer information for years--not just addresses and payment information but the identity of everything that a customer had ever bought or even looked at.
- While dozens of other companies do that, too, Amazon's doing something remarkable with theirs. They're using that data to build customer relationship



- Corporations and investors want to be able to track the consumer market as closely as possible to signal trends that will inform their next product launches.
- LinkedIn is a bank of data not just about people, but how people are making their money and what industries they are working in and how they connect to each other.

# Bigdata use cases



AT&T has 300 million customers. A team of researchers is working to turn data collected through the company's cellular network into a trove of information for policymakers, urban planners and traffic engineers.

The researchers want to see how the city changes hourly by looking at calls and text messages relayed through cell towers around the region, noting that certain towers see more activity at different times

- Largest retail company in the world. Fortune 1 out of 500
- Largest sales data warehouse: Retail Link, a \$4 billion project (1991). One of the largest "civilian" data warehouse in the world: 2004: 460 terabytes, Internet half as large
- Defines data science: What do hurricanes, strawberry Pop-Tarts, and beer have in common?



- Includes financial and marketing applications, but with special focus on industrial uses of big data
- When will this gas turbine need maintenance? How can we optimize the performance of a locomotive? What is the best way to make decisions about energy finance?

# Thank you

-Venkat Reddy