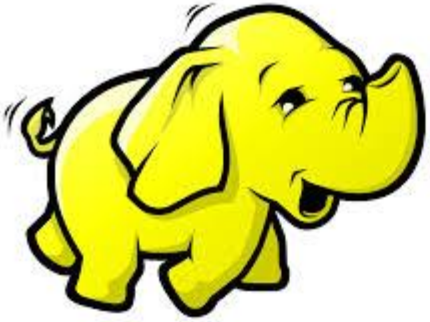


Hadoop & Pig

RMACC

Rocky Mountain Advanced Computing Consortium

Dr. Karina Hauser
Senior Lecturer
Management & Entrepreneurship

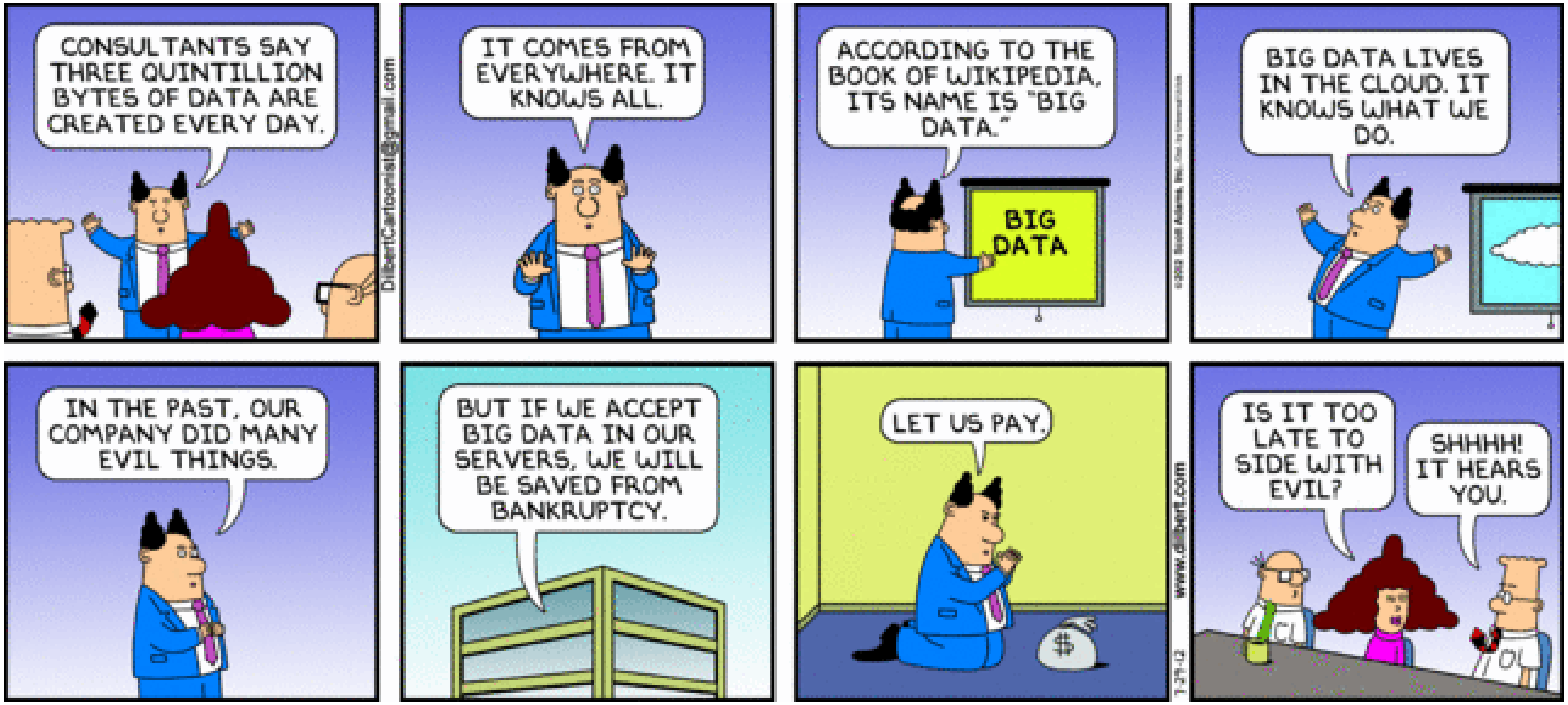
Outline

- Introduction
- (Setup)
- Hadoop, HDFS and MapReduce
- Pig

Introduction

- **What is Hadoop and where did it come from?**

Big Data


CONSULTANTS SAY THREE QUINTILLION BYTES OF DATA ARE CREATED EVERY DAY.
DilbertCartoonist@gmail.com
IT COMES FROM EVERYWHERE. IT KNOWS ALL.
ACCORDING TO THE BOOK OF WIKIPEDIA, ITS NAME IS "BIG DATA."
BIG DATA
BIG DATA LIVES IN THE CLOUD. IT KNOWS WHAT WE DO.
IN THE PAST, OUR COMPANY DID MANY EVIL THINGS.
BUT IF WE ACCEPT BIG DATA IN OUR SERVERS, WE WILL BE SAVED FROM BANKRUPTCY.
LET US PAY.
$
www.dilbert.com
7-29-12
IS IT TOO LATE TO SIDE WITH EVIL?
SHHHH! IT HEARS YOU.

Big Data Sources

- **Every day 2.5 quintillion bytes or 2.5 exabytes (10^{18}) are generated, that number is estimated to double every 40 month**
- **Astronomy**
  - Sloan Digital Sky Survey (SDSS) began collecting astronomical data in 2000; 200 GB (10^{9}) per night
  - Large Synoptic Survey Telescope (LSST) (~2020); estimated ~20 TB (10^{12}) per night
- **Business**
  - Twitter: 12 terabytes (10^{12}) of Tweets every day
  - Walmart: 2.5 petabytes (10^{15}) of data every hour from its customer transactions

Big Data - Big Business

- **IDC predicts big data technology and services will grow worldwide from $3.2 billion in 2010 to $16.9 billion in 2015. This represents a compound annual growth rate of 40 percent — about seven times that of the overall information and communications technology market.**

Infoworld Special Report: Big Data Deep Dive

Big Data in the News

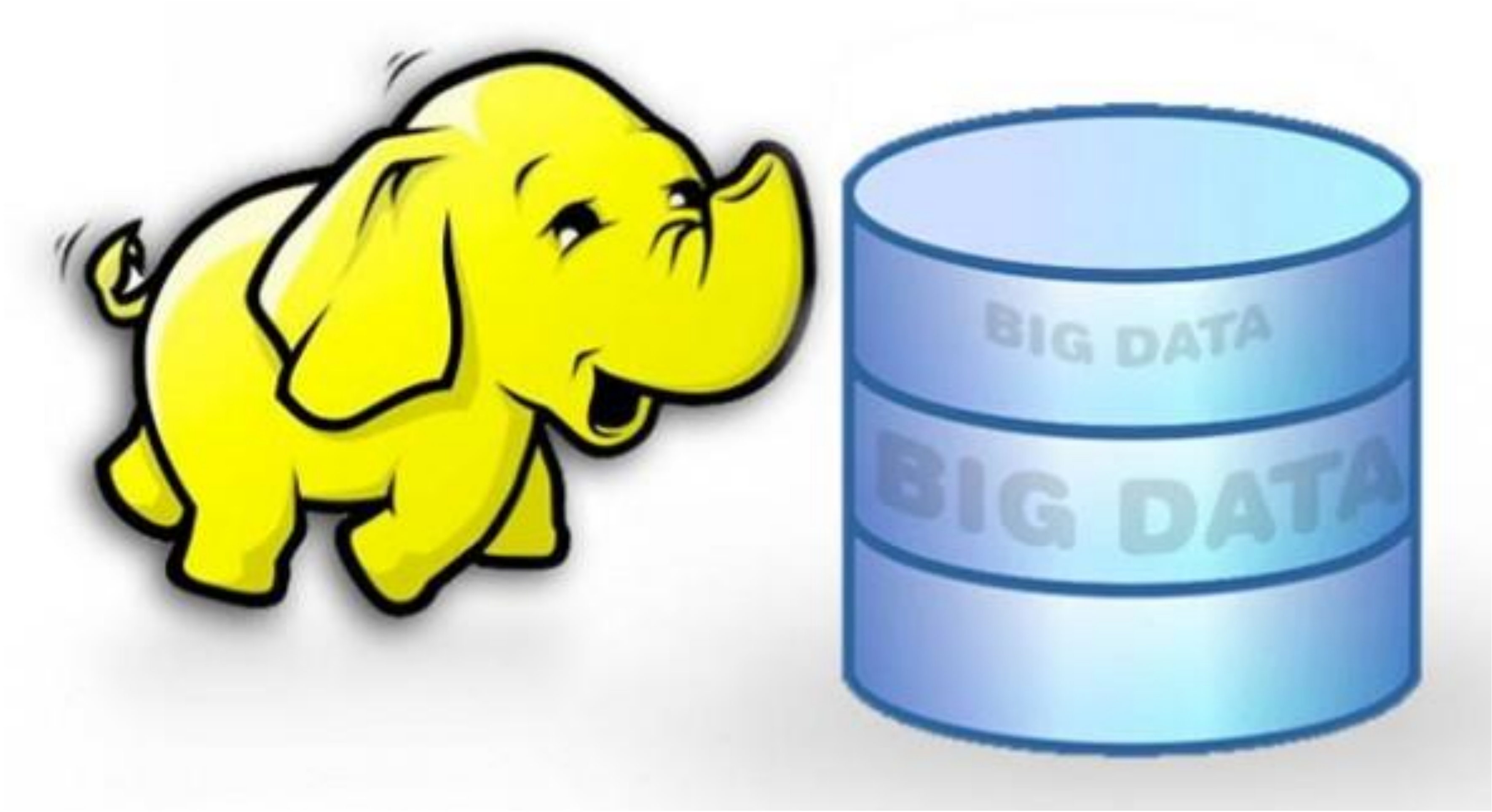
- **The Economist Intelligence Unit study showed that nine out of 10 surveyed business leaders believe data is now the fourth factor of production, as fundamental to business as land, labor and capital.**

The Economist, "Deciding Factor."

What is Big Data ?

- **“When the data itself becomes part of the problem”**
- **Three (to five) dimensions:**
  - Volume
  - Variety
  - Velocity
  - (Veracity)
  - (Value)

The Solution



Short History

- **Created by Doug Cutting, named after his son’s toy elephant**
- **2002 - Nutch, search engine, scalability problems**
- **2004 - Google papers on GFS and MapReduce**
- **2006 - Yahoo hires Doug to improve Hadoop**
- **2008 - Hadoop becomes Apache Top Level Project**

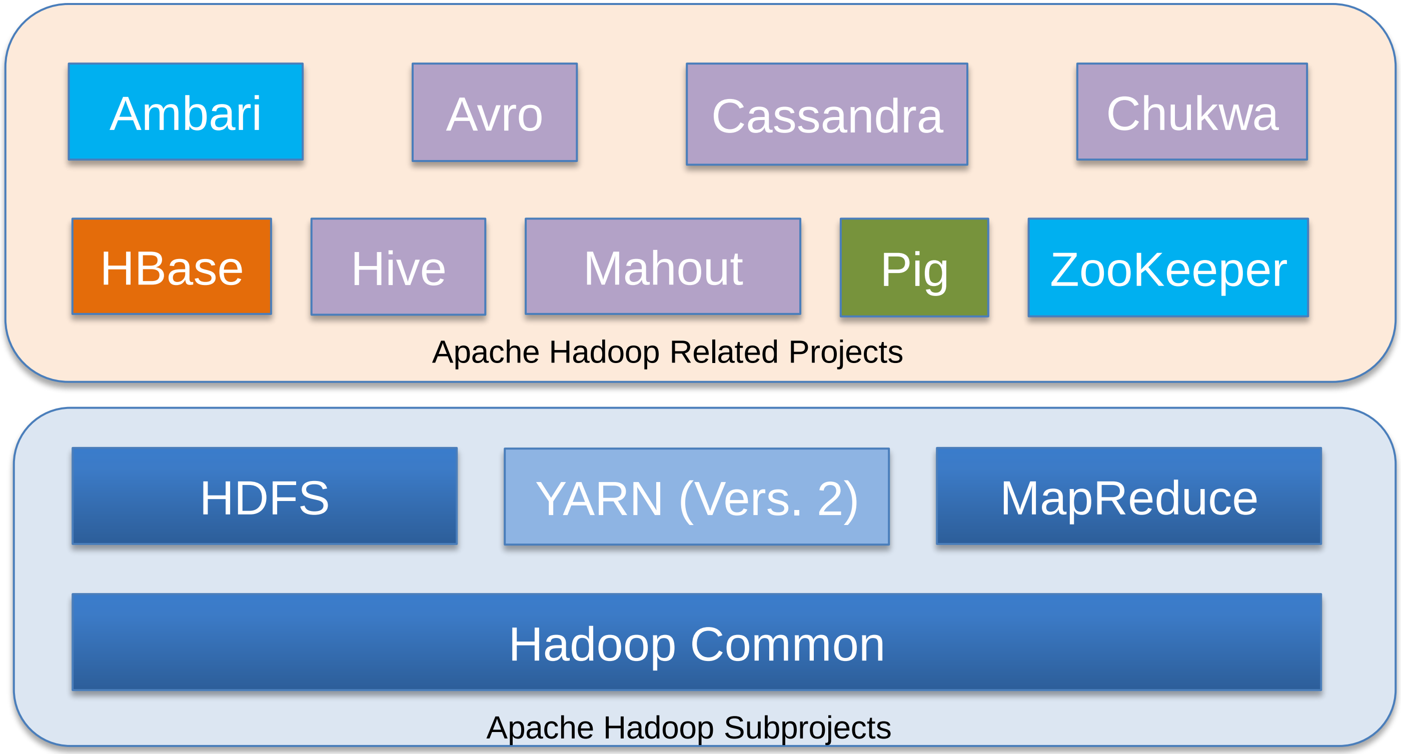
Hadoop Today

- **Moving from “Internet” companies**
  - Yahoo
  - Google
- **to business and science applications**
  - Customer relationship management
  - Bioinformatics
  - Astrophysics

Hadoop Definition

- **"Framework that allows for the**
  - – distributed processing of
  - – large data sets
  - – across clusters of computers
  - – using a simple programming model"
- **Open-source software, maintained by "The Apache Software Foundation"**
- **http://hadoop.apache.org/**

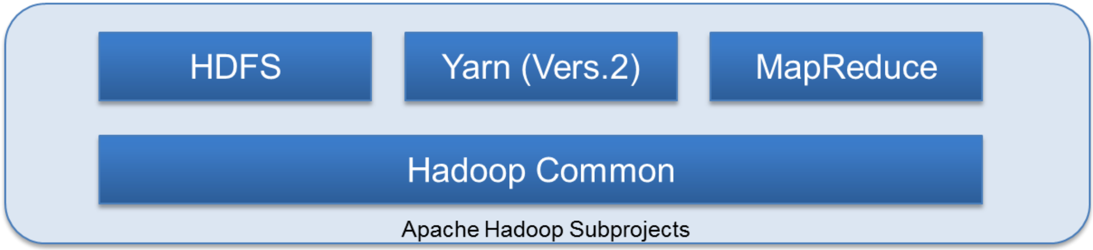
Apache Hadoop Projects



Hadoop Cluster

- **Commodity hardware**
- **Individual disk space on each node**
- **Hadoop framework handles:**
  - Data "backups" (through replication)
  - Hardware failure
  - Parallelization of code (through MapReduce paradigm)

HDFS



HDFS

Yarn (Vers.2)

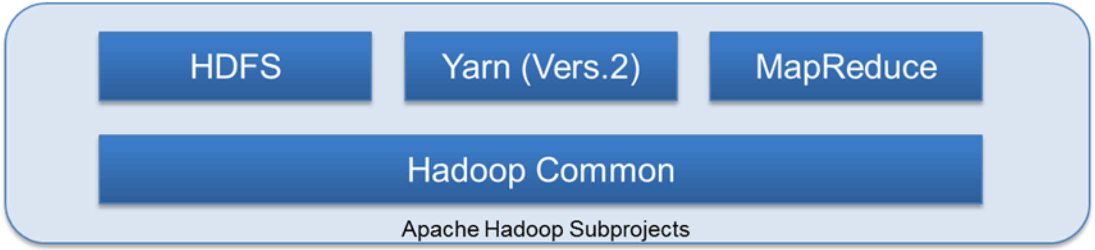
MapReduce

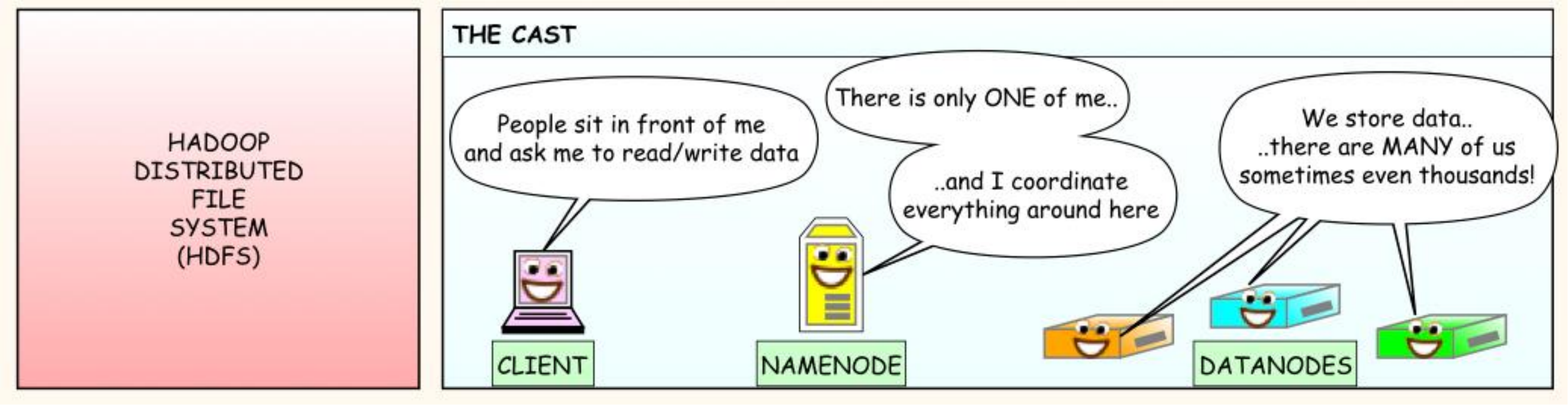
Hadoop Common

Apache Hadoop Subprojects

- **Write-once, read-many**
- **Each file is stored as sequence of same-sized blocks (default size 64MB)**
- **Blocks are replicated across different nodes**
- **Highly reliable:**
  - Redundant data storage
  - Heartbeat messages to detect connectivity problems

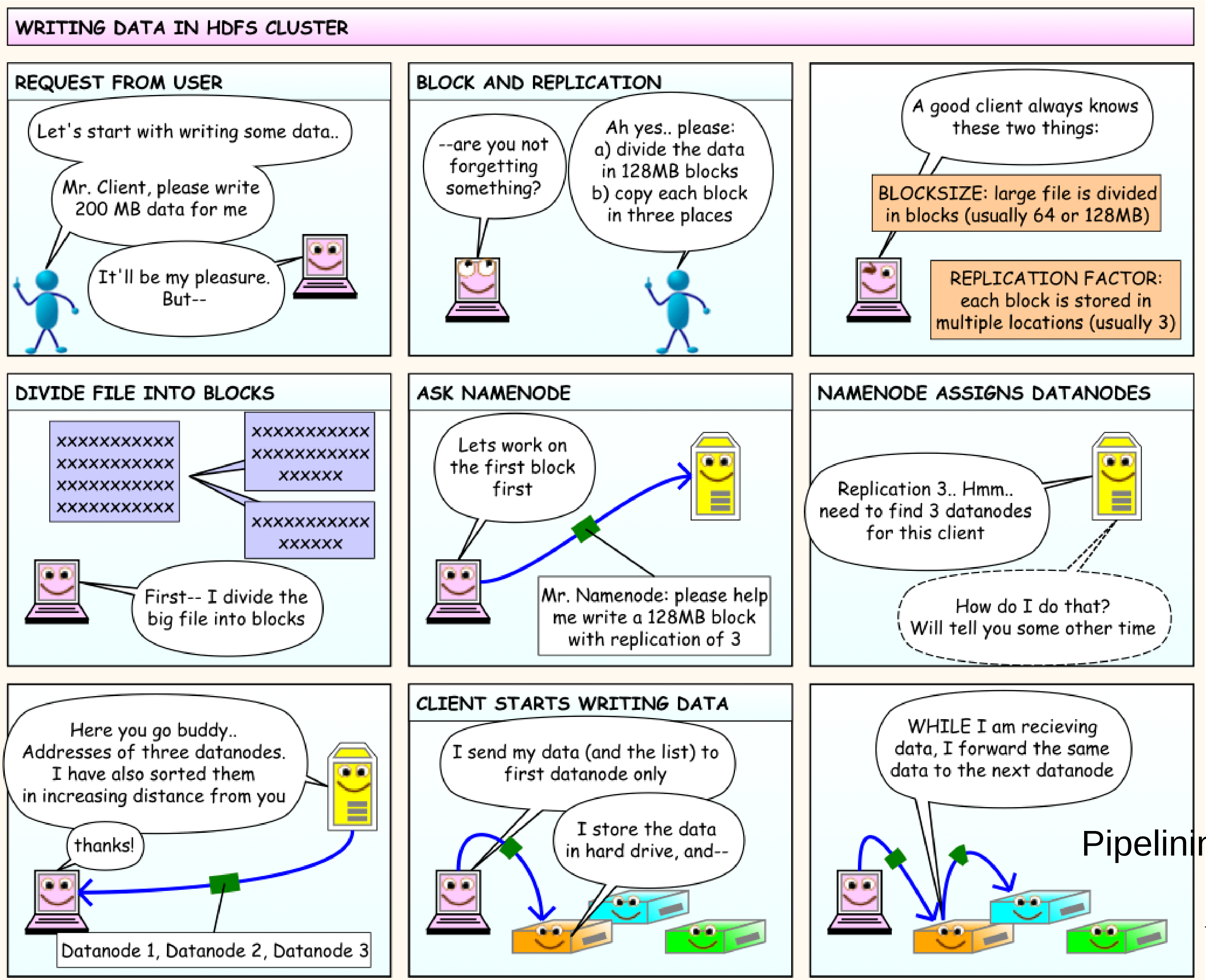
  → Automatic failover





HADOOP DISTRIBUTED FILE SYSTEM (HDFS)
THE CAST
People sit in front of me and ask me to read/write data
There is only ONE of me..
..and I coordinate everything around here
We store data.. ..there are MANY of us sometimes even thousands!
CLIENT
NAMENODE
DATANODES

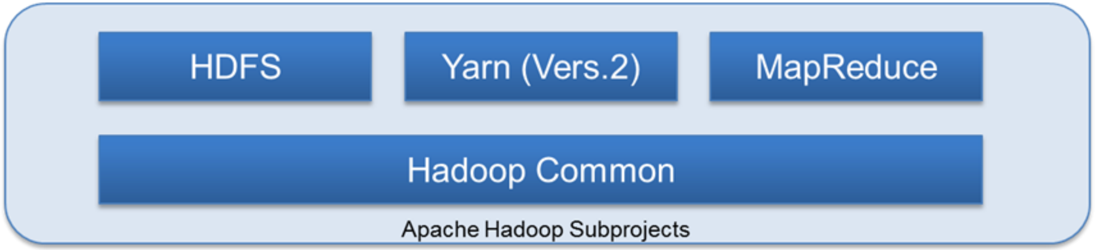
From: http://nosql.mypopescu.com/post/15561851616/hadoop-distributed-file-system-hdfs-a-cartoon-is-worth


WRITING DATA IN HDFS CLUSTER
REQUEST FROM USER
Let's start with writing some data..
Mr. Client, please write 200 MB data for me
It'll be my pleasure. But--
BLOCK AND REPLICATION
--are you not forgetting something?
Ah yes.. please: a) divide the data in 128MB blocks b) copy each block in three places
A good client always knows these two things:
BLOCKSIZE: large file is divided in blocks (usually 64 or 128MB)
REPLICATION FACTOR: each block is stored in multiple locations (usually 3)
DIVIDE FILE INTO BLOCKS
xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxxx
xxxxxxxxxxx xxxxxxxxxxx xxxxxx
xxxxxxxxxxx xxxxxx
First-- I divide the big file into blocks
ASK NAMENODE
Lets work on the first block first
Mr. Namenode: please help me write a 128MB block with replication of 3
NAMENODE ASSIGNS DATANODES
Replication 3.. Hmm.. need to find 3 datanodes for this client
How do I do that? Will tell you some other time
Here you go buddy.. Addresses of three datanodes. I have also sorted them in increasing distance from you
thanks!
Datanode 1, Datanode 2, Datanode 3
CLIENT STARTS WRITING DATA
I send my data (and the list) to first datanode only
I store the data in hard drive, and--
WHILE I am recieving data, I forward the same data to the next datanode

Pipelining

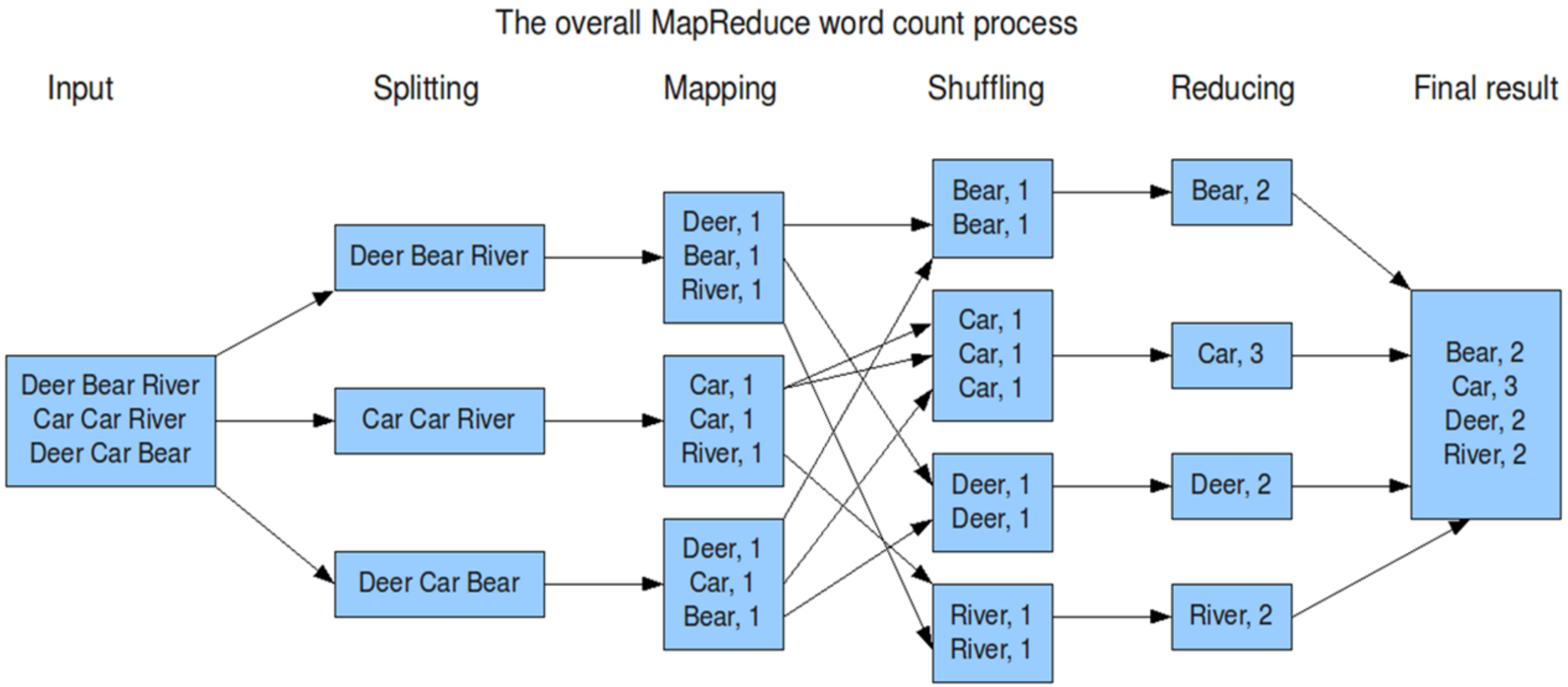
From: http://nosql.mypopescu.com/post/15561851616/hadoop-distributed-file-system-hdfs-a-cartoon-is-worth

MapReduce


HDFS
Yarn (Vers.2)
MapReduce
Hadoop Common
Apache Hadoop Subprojects

- **Programing model designed for**
  - batch processing of large volumes of data
  - in parallel
  - by dividing the work into a set of independent tasks
- **Not limited to Hadoop**

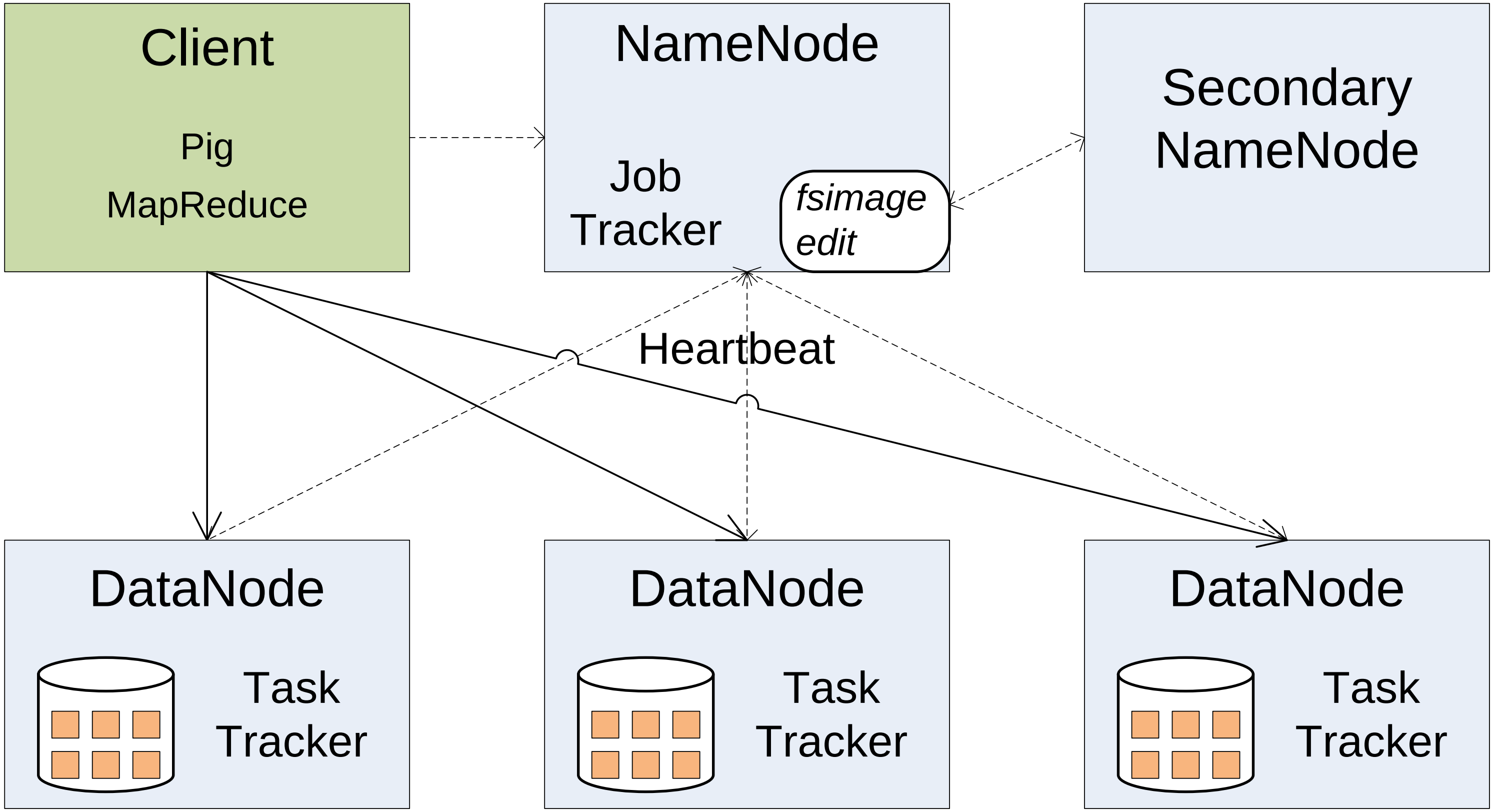
MapReduce WordCount Example



Problems suited for MapReduce

- **Iterate over a large number of records**
- **Extract something of interest from each**
- **Shuffle and sort intermediate results**
- **Aggregate intermediate results**
- **Generate final output**

Hadoop 1.x Components



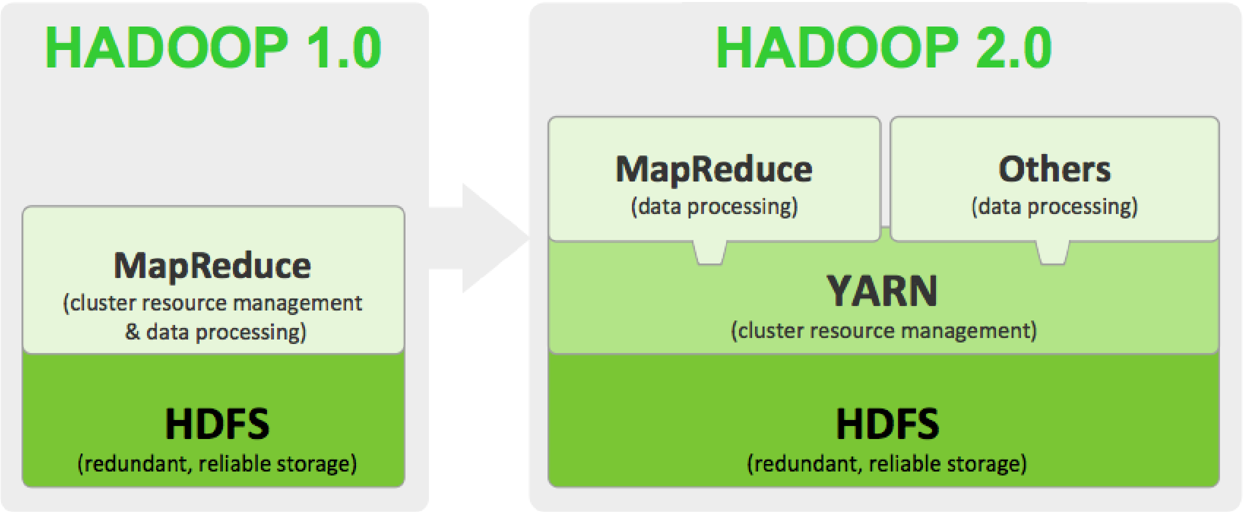
Hadoop 1.x Components

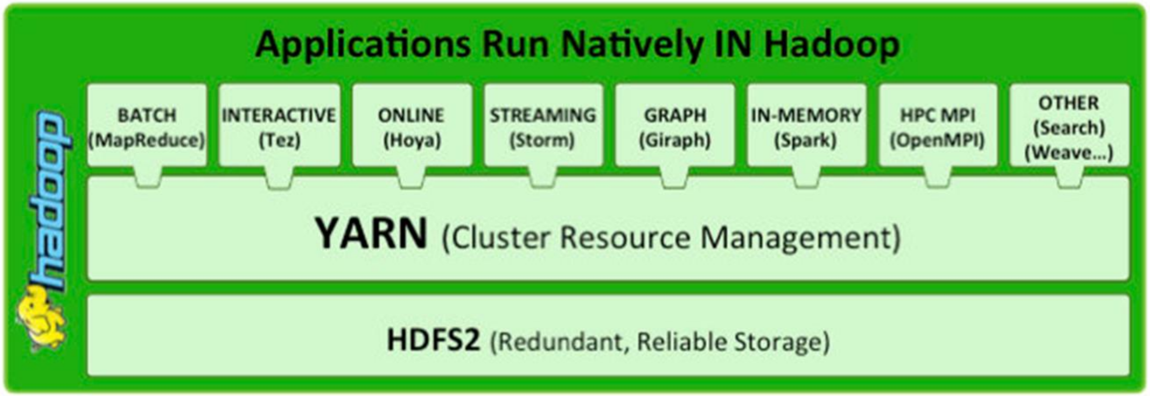
- **Name Node**
  - Stores metadata (filenames, replications factors …)
    - ***fsimage: latest checkpoint of namespace***
    - ***edits: log of changes to namespace***
  - Checks data node availability (Heartbeat)
  - If possible: Separate machine
- **Data Node**
  - Stores data
  - Replicates blocks
  - Computation

Hadoop 1.x Components

- **Secondary/Checkpoint Name Node**
  - Periodically creates checkpoints of namespace
    - ***Downloads fsimage and edits,
creates new namespace and updates Name Node***
  - Separate machine
- **Job Tracker**
  - Schedules and manages jobs
- **Task Tracker**
  - Executes MapReduce jobs on individual data node

Hadoop 2.0


HADOOP 1.0
MapReduce
(cluster resource management & data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant, reliable storage)


Applications Run Natively IN Hadoop
hadoop
BATCH
(MapReduce)
INTERACTIVE
(Tez)
ONLINE
(Hoya)
STREAMING
(Storm)
GRAPH
(Giraph)
IN-MEMORY
(Spark)
HPC MPI
(OpenMPI)
OTHER
(Search)
(Weave...)
YARN (Cluster Resource Management)
HDFS2 (Redundant, Reliable Storage)

Setup

- **Three options:**
  - Standalone (single Java process)
  - **Pseudo-Distributed (separate Java processes)**
  - Fully-Distributed
- **Prerequisites (on virtual machine):**
  - Ubuntu server 12.04
  - Ubuntu desktop (for monitoring)
  - Oracle (Sun) Java 1.7.0_25
    - ***http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html***
  - SSH

Setup Files

- **hosts (ip address)**
- **.bashrc (Java dir, home dir)**
- **hadoop configuration files in /usr/local/hadoop/conf**
  - hadoop-env.sh (Java dir)
  - hdfs-site.xml (replication factor)
  - mapred-site.xml (host/port for jobtracker)
  - master/slave (ips for multi-node cluster)

Login

- **User: rmacc**
- **Password: rmacc**
- **Start terminal (Ctrl+Alt+F1)**
- **Login as hduser**
  - User: hduser
  - Password: hduser
- **Change directory to hadoop**
  - with cd $HADOOP_PREFIX
  - or cd /usr/local/hadoop

Starting Hadoop Daemons

- **All:**

```
$ bin/start-all.sh
```

- Log output in *logs* directory

Check Daemons

- **jps** →
  - 1367 Jps
  - 8695 DataNode
  - 8609 NameNode
  - 6318 SecondaryNameNode
  - 2600 JobTracker
  - 2830 TaskTracker

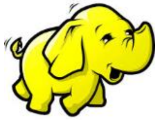
Web Interfaces

- http://localhost:50030 → Cluster status and jobs
- http://localhost:50070 → HDFS

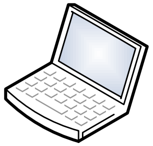
HDFS

- **Hadoop Distributed File System**

Ubuntu (Linux)



Hadoop



bin/hadoop dfs -copyFromLocal mis6110/Files/file.txt mis6110/KDD/file.txt →

← *bin/hadoop dfs -copyToLocal mis6110/Movies/file.txt mis6110/Files/file.txt*

local file system
/usr/local/hadoop/...

mis6110

Files | Scripts

(HDFS)
/user/hduser/...

mis6110

KDD | Movies

	Ubuntu (Linux)	Hadoop
Create Directory	***mkdir***	***bin/hadoop dfs -mkdir***
Show content of directory	***ls***	***bin/hadoop dfs -ls***

Create Directory	***sh mkdir***	***fs -mkdir***
Show content of directory	***sh ls***	***fs -ls***

HDFS Shell

- **bin/hadoop dfsadmin –help → all admin commands**
- **bin/hadoop dfs –help → all commands**
- **Most commands similar to unix**
  - dfs –copyFromLocal
  - dfs -ls
- Shell commands: **http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html**

Importing Data

- **KDD Example:** Legcare sales data from KDD Cup 2000
  “We wish to thank Blue Martini Software for contributing the KDD Cup 2000 data”
  - Cleaned for easier/faster use
  - Copy file KDDCupCleaned.txt to hdfs

Copying File to HDFS

- **Create new directory:**

```
$ bin/hadoop  dfs  -mkdir rmacc
```

- **Copy files:**

```
$ bin/hadoop  dfs  -copyFromLocal ../rmacc/Beowulf.txt rmacc/
$ bin/hadoop  dfs  -copyFromLocal ../rmacc/KDD* rmacc/
```

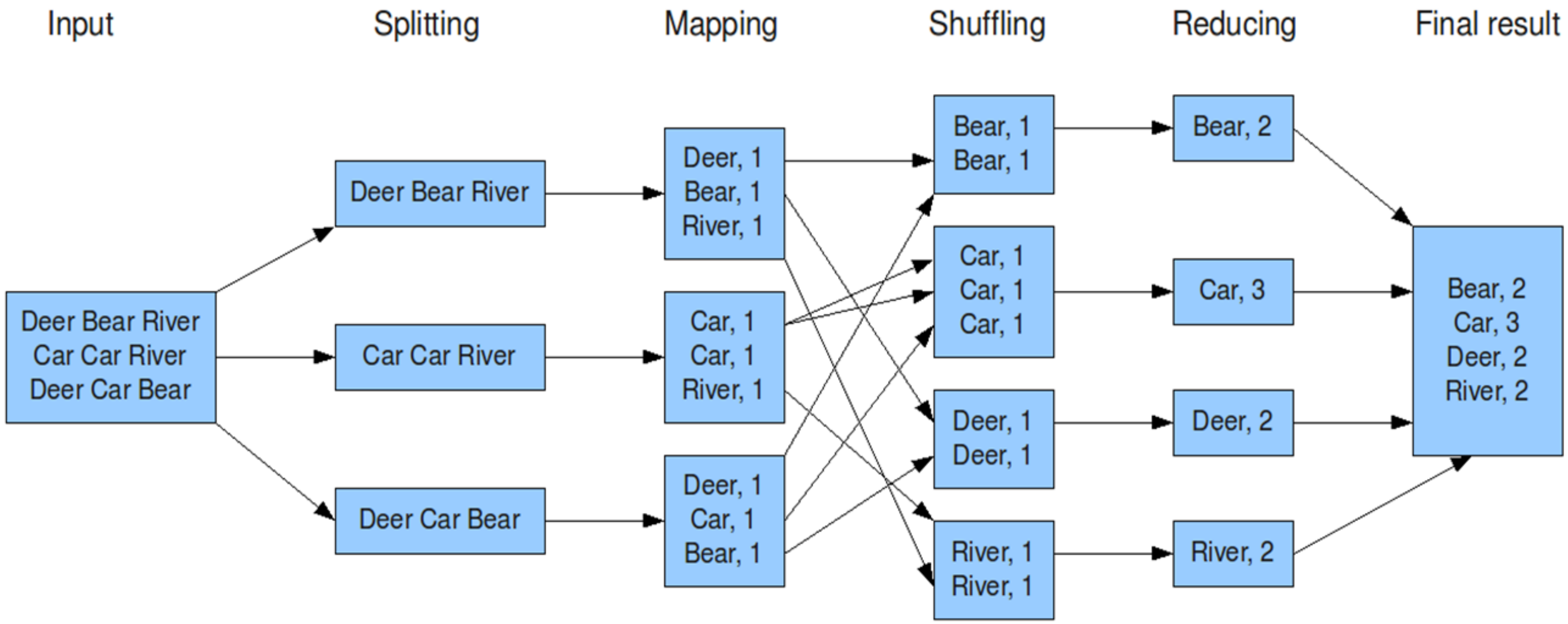
- **Check:**

```
$ bin/hadoop dfs   -lsr  rmacc
```

  - ***Localhost:50070 →Browse the filesystem***

MapReduce WordCount Example

The overall MapReduce word count process



From: http://www.rabidgremlin.com/data20/#(3)

Java Code for WordCount Example

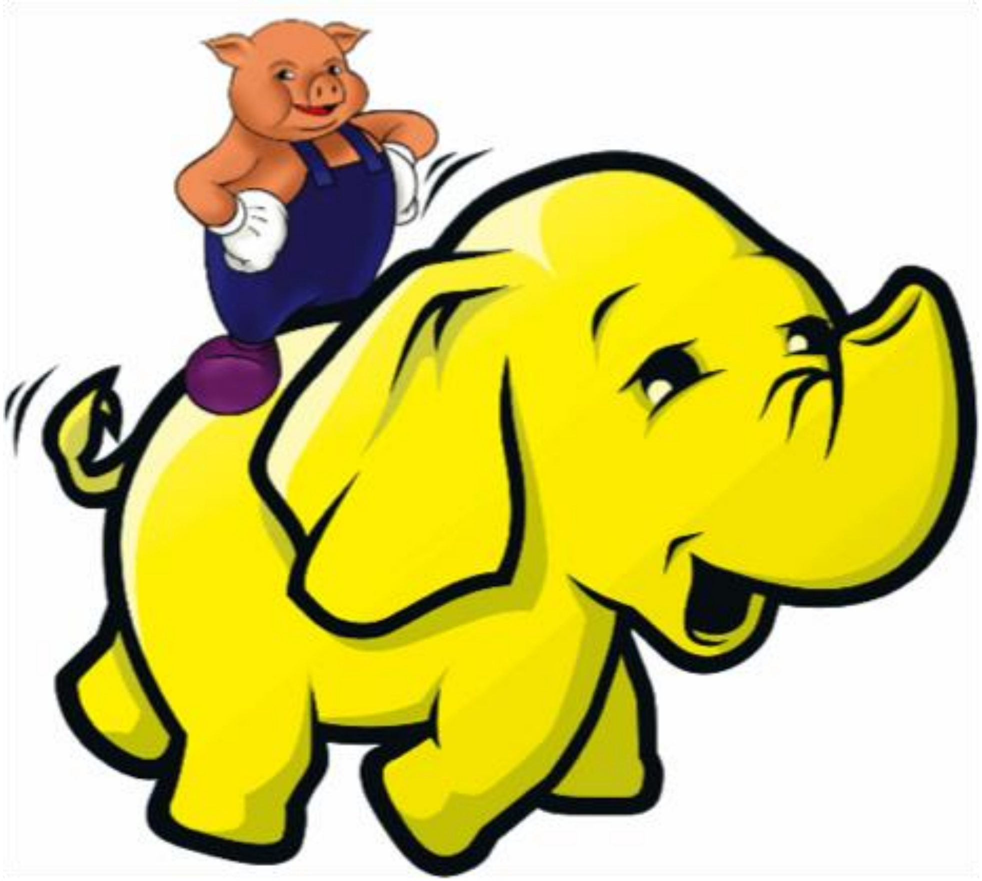
```
package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
import org.apache.hadoop.util.*;

public class WordCount {

  public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
   private final static IntWritable one = new IntWritable(1);
   private Text word = new Text();

   public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
    String line = value.toString();
    StringTokenizer tokenizer = new StringTokenizer(line);
    while (tokenizer.hasMoreTokens()) {
     word.set(tokenizer.nextToken());
     output.collect(word, one);
    }            }
  }

  public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {
```

Java Code for WordCount Example

```
    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
      int sum = 0;
      while (values.hasNext()) {
        sum += values.next().get();
      }
      output.collect(key, new IntWritable(sum));
    }
  }

  public static void main(String[] args) throws Exception {
    JobConf conf = new JobConf(WordCount.class);
    conf.setJobName("wordcount");

    conf.setOutputKeyClass(Text.class);
    conf.setOutputValueClass(IntWritable.class);

    conf.setMapperClass(Map.class);
    conf.setCombinerClass(Reduce.class);
    conf.setReducerClass(Reduce.class);

    conf.setInputFormat(TextInputFormat.class);
    conf.setOutputFormat(TextOutputFormat.class);

    FileInputFormat.setInputPaths(conf, new Path(args[0]));
    FileOutputFormat.setOutputPath(conf, new Path(args[1]));

    JobClient.runJob(conf);
  }
}
```

There Must be an Easier Way

Pig (Latin)



Pig

- **High-level data processing language (Pig Latin)**
- **Resides on user machine, not cluster**
- **Pig Latin compiled into efficient MapReduce jobs**

Test Pig Installation

- **(Hadoop has to be running)**
  - $ cd $PIG_HOME
  - $ bin/pig -help

How to Run Pig

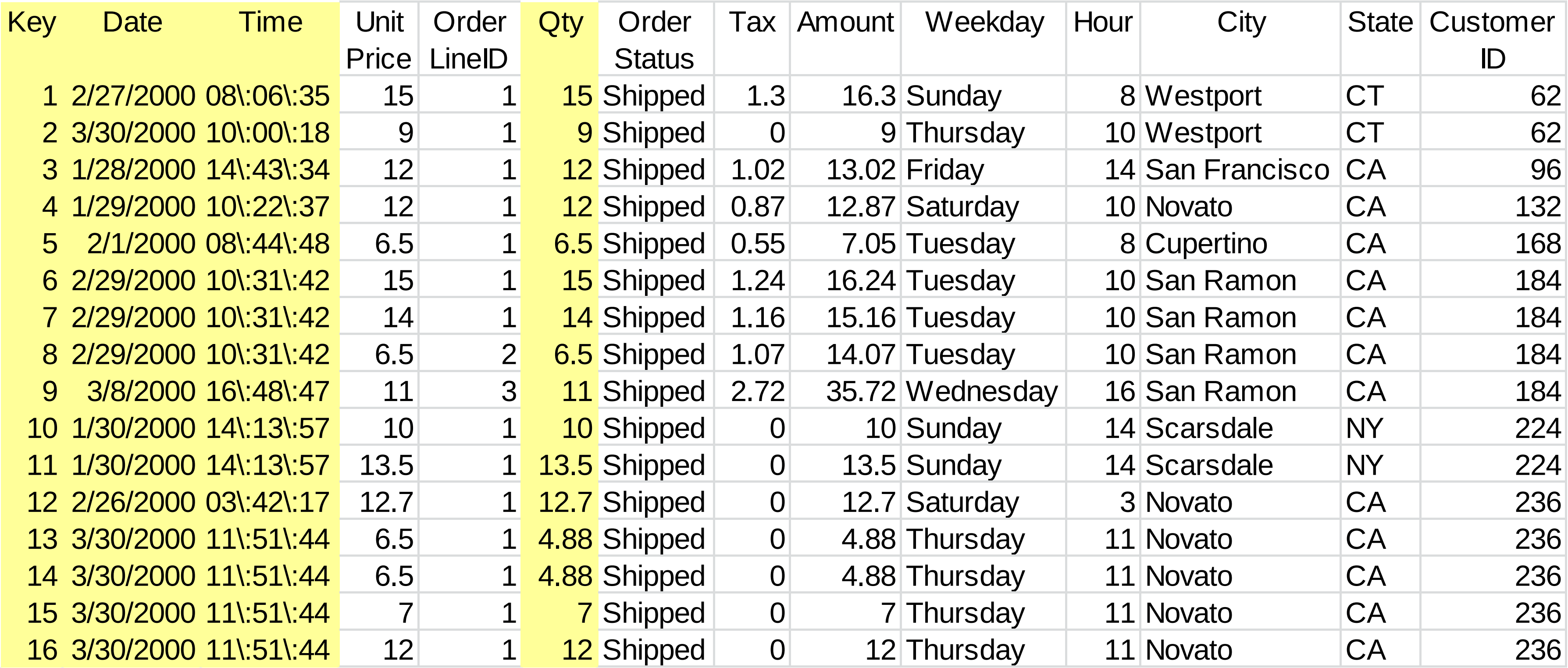
- **Grunt interactive shell**
  - Two modes:
    - ***Local, standalone (pig –x local)***
    - ***Hadoop, distributed (pig –x mapreduce)***
- **Scripts (.pig)**
- **Embedded in Java or Python**
- **PigPen, Eclipse plugin**

Pig Statements in Grunt

- **LOAD → Transform data → DUMP or STORE**
- **Example:**
  - ***grunt> A = load 'student' using PigStorage() AS (name:chararray, age:int, gpa:float);***
  - ***grunt> B = foreach A generate name;***
  - ***grunt> dump B***

– *"A " is called a "relation" or "outer bag"*

Pig Example: KDD Data



Key	Date	Time	Unit	Order	Qty	Order	Tax	Amount	Weekday	Hour	City	State	Customer
			Price	LineID		Status							ID
1	2/27/2000	08\:06\:35	15	1	15	Shipped	1.3	16.3	Sunday	8	Westport	CT	62
2	3/30/2000	10\:00\:18	9	1	9	Shipped	0	9	Thursday	10	Westport	CT	62
3	1/28/2000	14\:43\:34	12	1	12	Shipped	1.02	13.02	Friday	14	San Francisco	CA	96
4	1/29/2000	10\:22\:37	12	1	12	Shipped	0.87	12.87	Saturday	10	Novato	CA	132
5	2/1/2000	08\:44\:48	6.5	1	6.5	Shipped	0.55	7.05	Tuesday	8	Cupertino	CA	168
6	2/29/2000	10\:31\:42	15	1	15	Shipped	1.24	16.24	Tuesday	10	San Ramon	CA	184
7	2/29/2000	10\:31\:42	14	1	14	Shipped	1.16	15.16	Tuesday	10	San Ramon	CA	184
8	2/29/2000	10\:31\:42	6.5	2	6.5	Shipped	1.07	14.07	Tuesday	10	San Ramon	CA	184
9	3/8/2000	16\:48\:47	11	3	11	Shipped	2.72	35.72	Wednesday	16	San Ramon	CA	184
10	1/30/2000	14\:13\:57	10	1	10	Shipped	0	10	Sunday	14	Scarsdale	NY	224
11	1/30/2000	14\:13\:57	13.5	1	13.5	Shipped	0	13.5	Sunday	14	Scarsdale	NY	224
12	2/26/2000	03\:42\:17	12.7	1	12.7	Shipped	0	12.7	Saturday	3	Novato	CA	236
13	3/30/2000	11\:51\:44	6.5	1	4.88	Shipped	0	4.88	Thursday	11	Novato	CA	236
14	3/30/2000	11\:51\:44	6.5	1	4.88	Shipped	0	4.88	Thursday	11	Novato	CA	236
15	3/30/2000	11\:51\:44	7	1	7	Shipped	0	7	Thursday	11	Novato	CA	236
16	3/30/2000	11\:51\:44	12	1	12	Shipped	0	12	Thursday	11	Novato	CA	236

Pig LOAD Function

- **“Pigs eat anything”**
- **LOAD 'data' [USING function] [AS schema];**
- **USING**
  - PigStorage → structured text file (default)
  - TextLoader → unstructured UTF-8 data
  - Other and User Defined Functions
- **AS**
  - (Field1[:type], Field2:[type], ... FieldX[type])
  - Bytearray default type

Pig Example: Loading Data

```
>  pig  –x  mapreduce
>  a  =  LOAD ‘rmacc/KDDCupCleaned.txt‘;
```

> All Statements end with semicolon !!!

Pig Debugging Statements

Debug Operator	Description
DUMP	Display results
DESCRIBE	Display schema of relation
EXPLAIN	Display execution plan
ILLUSTRATE	Display step-by-step execution
Full list	http://pig.apache.org/docs/r0.11.1/test.html#diagnostic-ops

- **Describe and illustrate only work if schema is provided**

Pig Example: Loading Data

```
>   a = LOAD  ‘rmacc/KDDrmacc‘  , AS
    (key,date,time, qty:float);
```

- **Columns can be accessed by**
  - $0   for second column (first contains key)
  - or name   (key, date, time….)

Pig Data Types

- **Simple:**
  - int, long, float, double, chararray, bytearray, boolean
- **Complex:**
  - tuple - a set of fields (10, 5, alpha)
  - bag - a collection of tuples {(10,5,alpha) (8,2,beta)}
  - map - a set of key value pairs [key#value]

Pig Example: Reduce # of Fields

```
a = LOAD ‘rmacc/KDDCupCleaned.txt‘ AS (key:int,date,time,up,ol,qty,os,tax,amount:float,wd,hour,city,state,ci);
b = FOREACH a GENERATE key, date, time, amount;
STORE b INTO ‘rmacc/KDDCupShort’.txt USING PigStorage(‘,’) ;
```

Pig Example:Group per Date

- a = LOAD ‘rmacc/KDDrmacc.txt‘ AS (key,date,time,qty:float);
- groupday = GROUP a BY date;

- illustrate groupday →
  - *group* is new key for each bag (day)
  - tuples within data within each bag

Pig Example: Sum per Date

```
>   a = LOAD  ‘rmacc/KDDrmacc.txt‘  AS (key,date,time,qty:float);
>   groupday = GROUP  a  BY date;
>   sumday = FOREACH groupday GENERATE group, SUM(a.qty); sr
>   STORE sumday INTO ‘rmacc/sumday’;
```

Evaluation Functions

(Case Sensitive)

Function	Description
AVG	Calculates average
CONCAT	Concatenates two expressions of identical type
COUNT	Counts the number of elements in a bag
COUNT_STAR	Like count by includes NULL values in count
DIFF	Compares two fields in a tuple
IsEmpty	Checks if a bag or map is empty
MAX	Calculates maximum
MIN	Calculates minimum
SIZE	Computes the number of elements (characters)
SUM	Calculates sum
TOKENIZE	Splits a string and outputs a bag of words
List with examples	http://pig.apache.org/docs/r0.11.1/func.html#eval-functions

Other Functions

- **Math Functions**
- **String Functions**
- **Datetime Functions**
- **Tuple, Bag, Map Functions**

- **User Defined Functions**

Pig Example: Filter Purchases> $100

```
>   a = LOAD ‘rmacc/sumday‘ AS (date,sum:float);
>   bigpur = FILTER a BY sum>1000;
>   STORE  bigpur  INTO  ‘rmacc/bigpur’;
```

Relational Operators

Operators	Description
LOAD	Loads data from the file system
GROUP	Groups the data in one or more relations
FOREACH	Generates data transformations based on columns of data
FILTER	Selects tuples from a relation based on some condition
Full list with examples	http://pig.apache.org/docs/r0.11.1/basic.html#Relational+Operators

Other Operators

- **Arithmetic Operators**
- **Boolean Operators**
- **Cast Operators**
- **Comparison Operators**
- **Type Construction Operators**
- **Dereference Operators**
- **Disambiguate Operator**
- **Flatten Operator**
- **Null Operators**
- **Sign Operators**

Pig WordCount Code

```
> b = LOAD ‘rmacc/Beowulf.txt‘ AS be;
> beowords = FOREACH b GENERATE flatten(TOKENIZE(beo)) as bw;
> wg = GROUP beowords BY bw;
> wc = FOREACH wg GENERATE group, COUNT(beowords) as bc;
> sumord =ORDER wc BY bc;
> STORE sumord INTO ‘rmacc/beowulfwc’
```

Hadoop Summary

- **Accessible – runs on commodity hardware**
- **Robust – handles hardware failures**
- **Scalable – by adding more nodes**
- **Simple – allows users to quickly write efficient parallel code**
- **Pig - easy to learn**

Conclusions

- **Individual components “easy” to setup → integration more complicated**
- **Resources**
  - Apache Hadoop (download and docu)
    - *http://hadoop.apache.org*
  - Online Searches
  - Books for overview, not technical details
- **“Evolving project” → constantly changing, documentation can’t keep up with development**

Questions ?