

# BigData Tools

Abhijit Bendale (abendale@vast.uccs.edu)

04/15/2014

# + BigData Tools



- Data Analysis and Platforms
- Business Intelligence
- Document Store
- Twitter Case Study

## Data Analysis & Platforms



## Databases / Data warehousing



## Operational



## Multivalued database



## Business Intelligence



## Data Mining



## Social



## Big Data search



## Data aggregation



## Key Value



## Document Store



## Graphs



## Multidimensional



## Project Voldemort



## Object databases



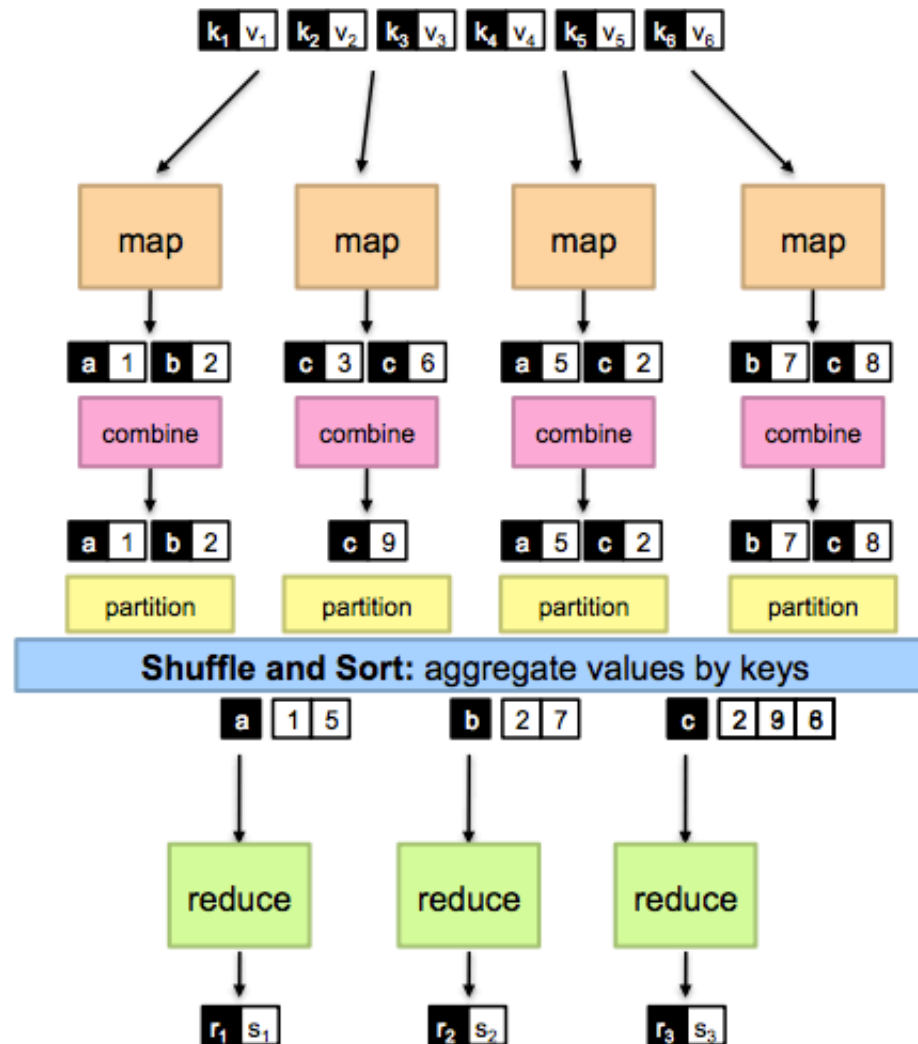
## Multimodel



## XML Databases



# MapReduce flow



# Map

- Independent record transformations
  - And deletions and replications
- $(K1, V1) \rightarrow \text{list}(K2, V2)$

Receives a key value pair  
And outputs a 0 or more  
key-value pairs

# Reduce

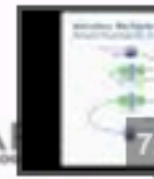
- Aggregate results from map phase
- $(K2, \text{list}(V2)) \rightarrow \text{list}(K3, V3)$

Reduce all the key-pairs with key  
K2 to a new reduced key-value pair  
K3, V3

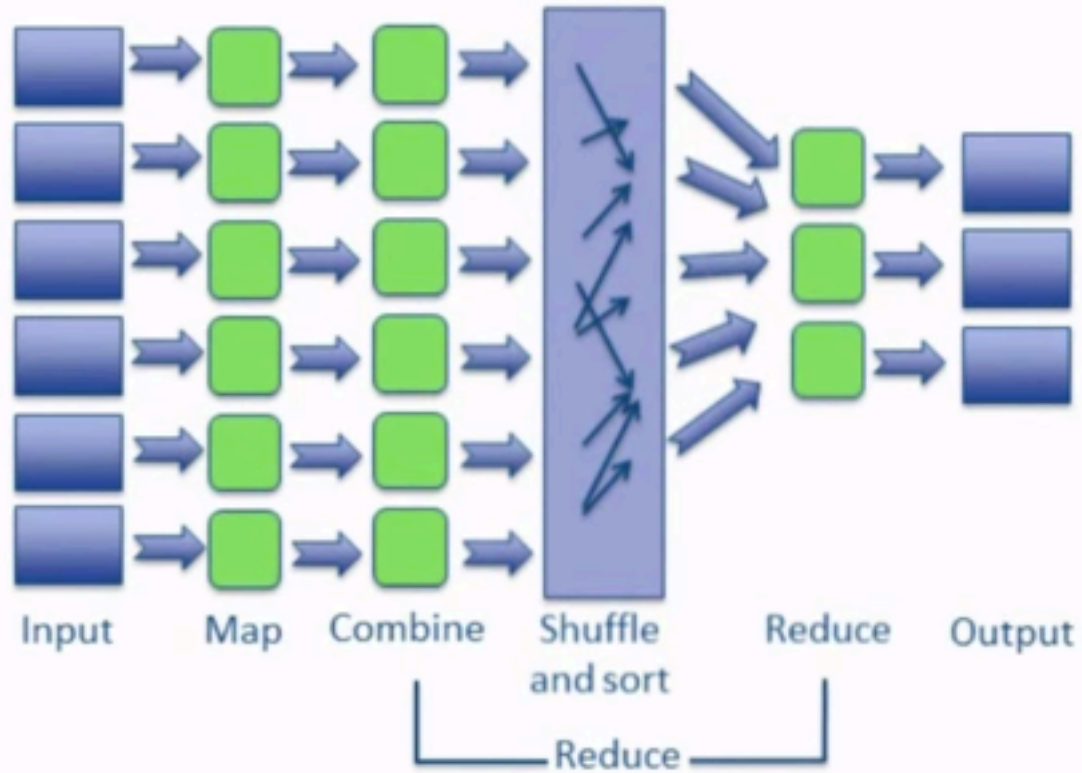
# Framework

- Schedules and re-runs tasks
- Splits the input
- Moves map outputs to reduce inputs
- Receives the results

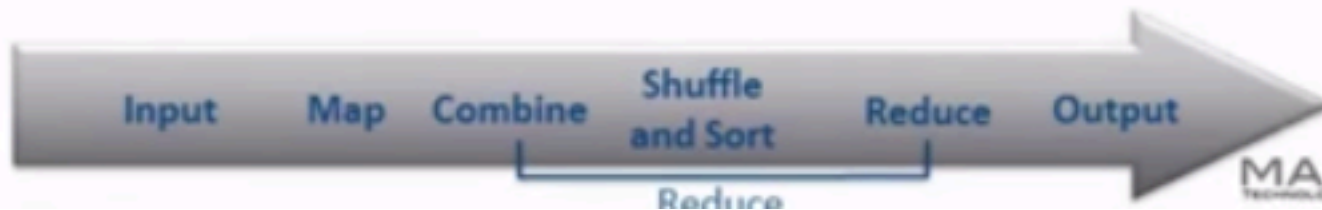
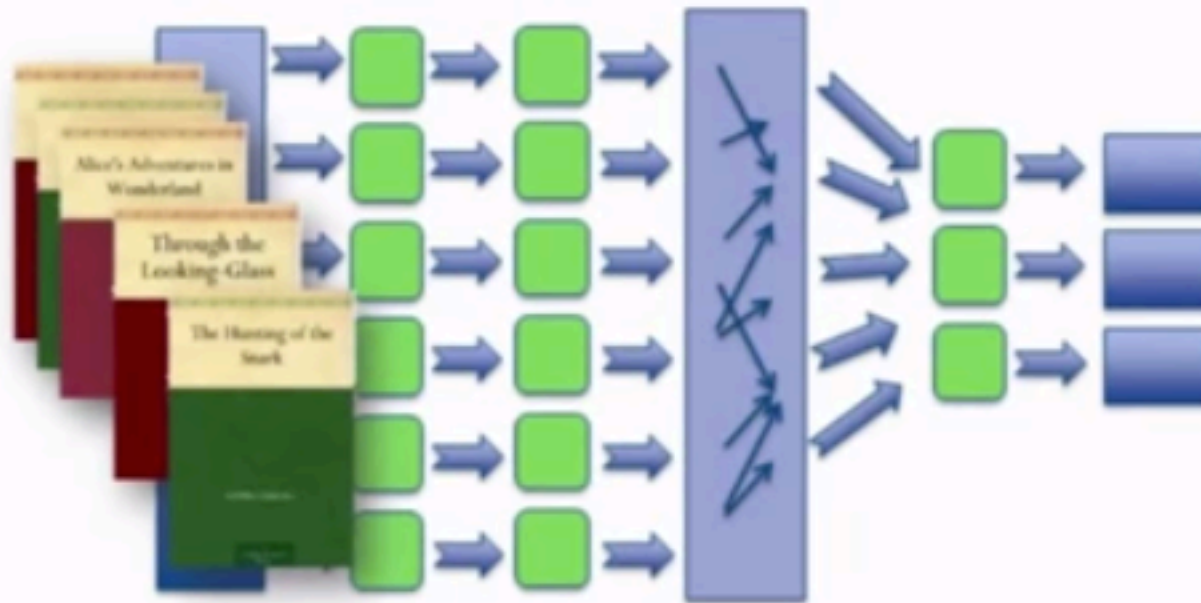
Majority of what Hadoop does..!



# MapReduce Flow



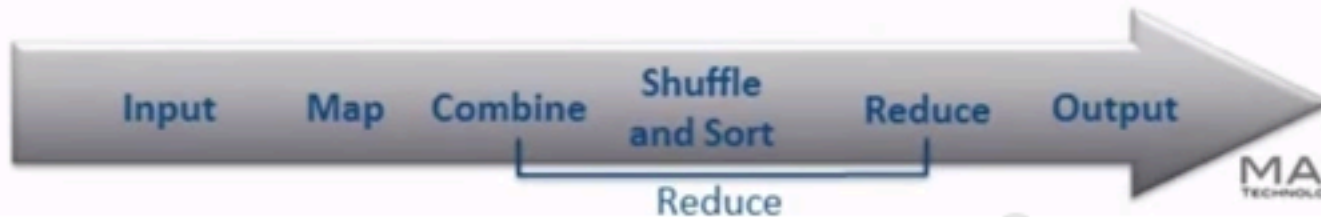
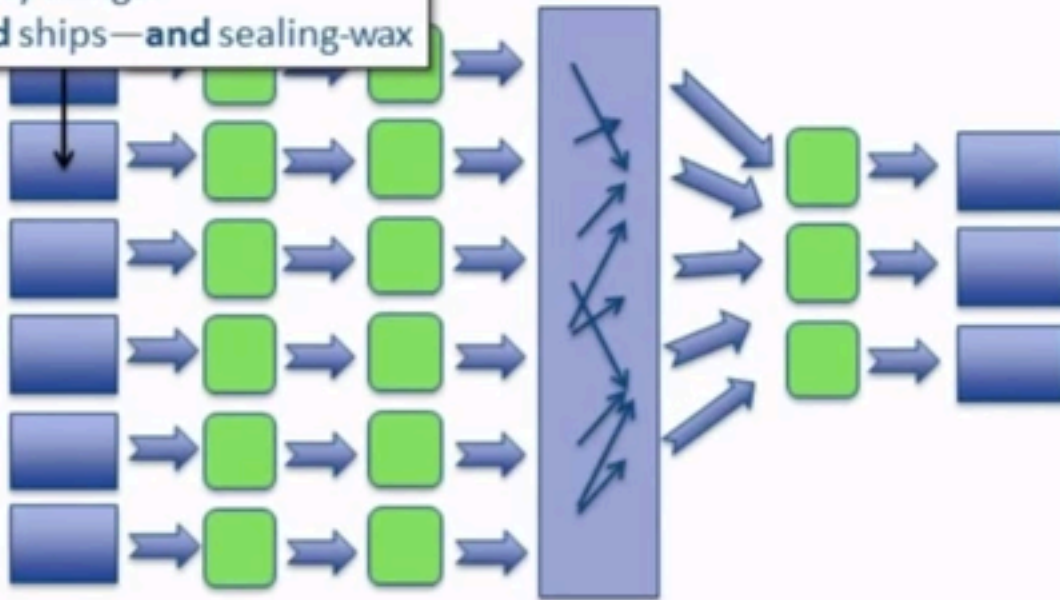
# Example: Word Count



Goal: Consider we have a list of books and we want to count occurrences of every word. Hadoop will distribute this task.

# Example: Word Count

"The time has come," the Walrus said,  
"To talk of many things:  
Of shoes—**and** ships—**and** sealing-wax



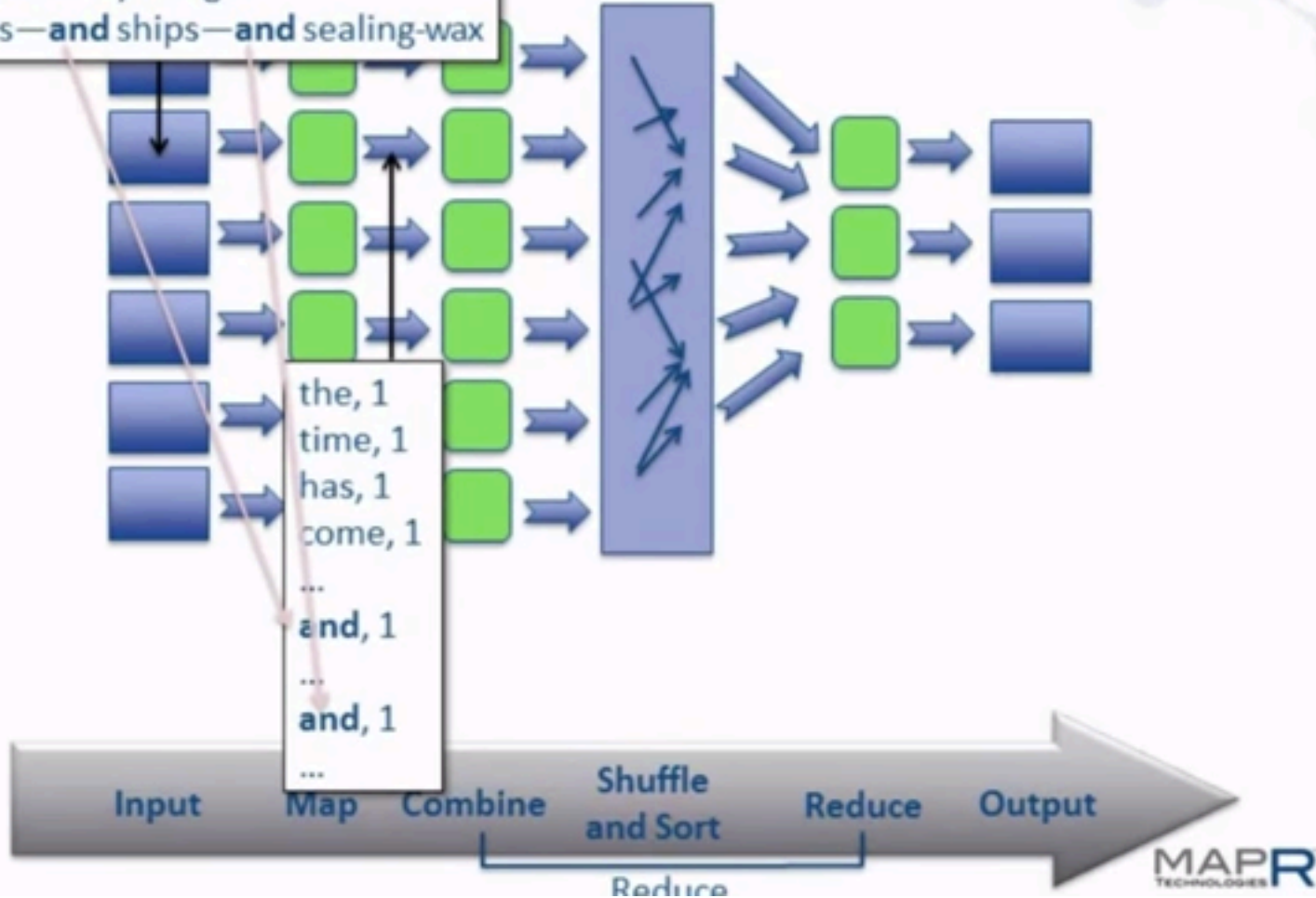
MAPR  
TECHNOLOGIES

Here Key is byte offset in the file, Value is the text.



# Example: Word Count

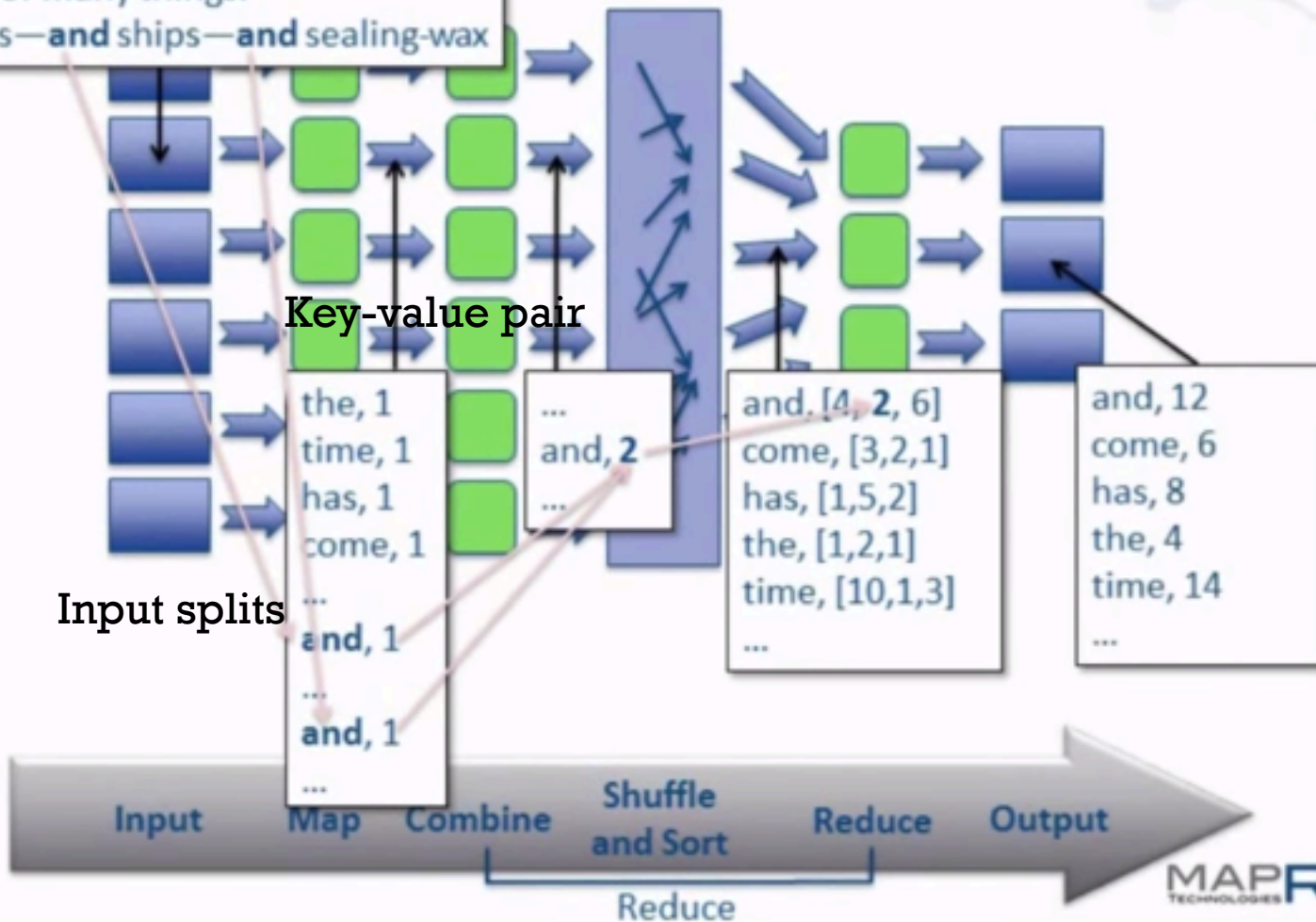
"The time has come," the Walrus said,  
"To talk of many things:  
Of shoes—**and** ships—**and** sealing-wax



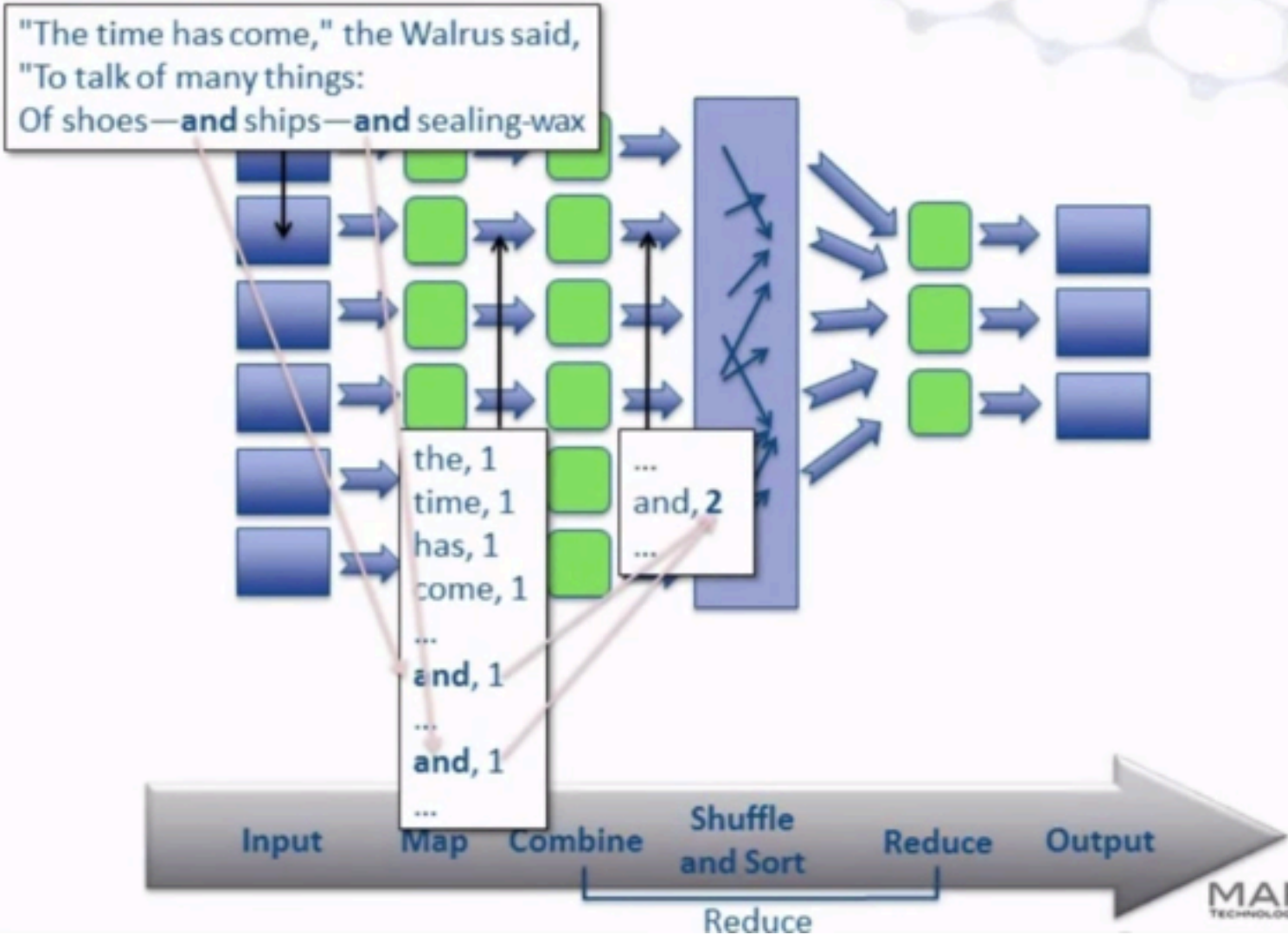
The map function tokenizes the input string and outputs key-value pair for every word. Note here “and” shows twice.

# Example: Word Count

"The time has come," the Walrus said,  
"To talk of many things:  
Of shoes—**and** ships—**and** sealing-wax



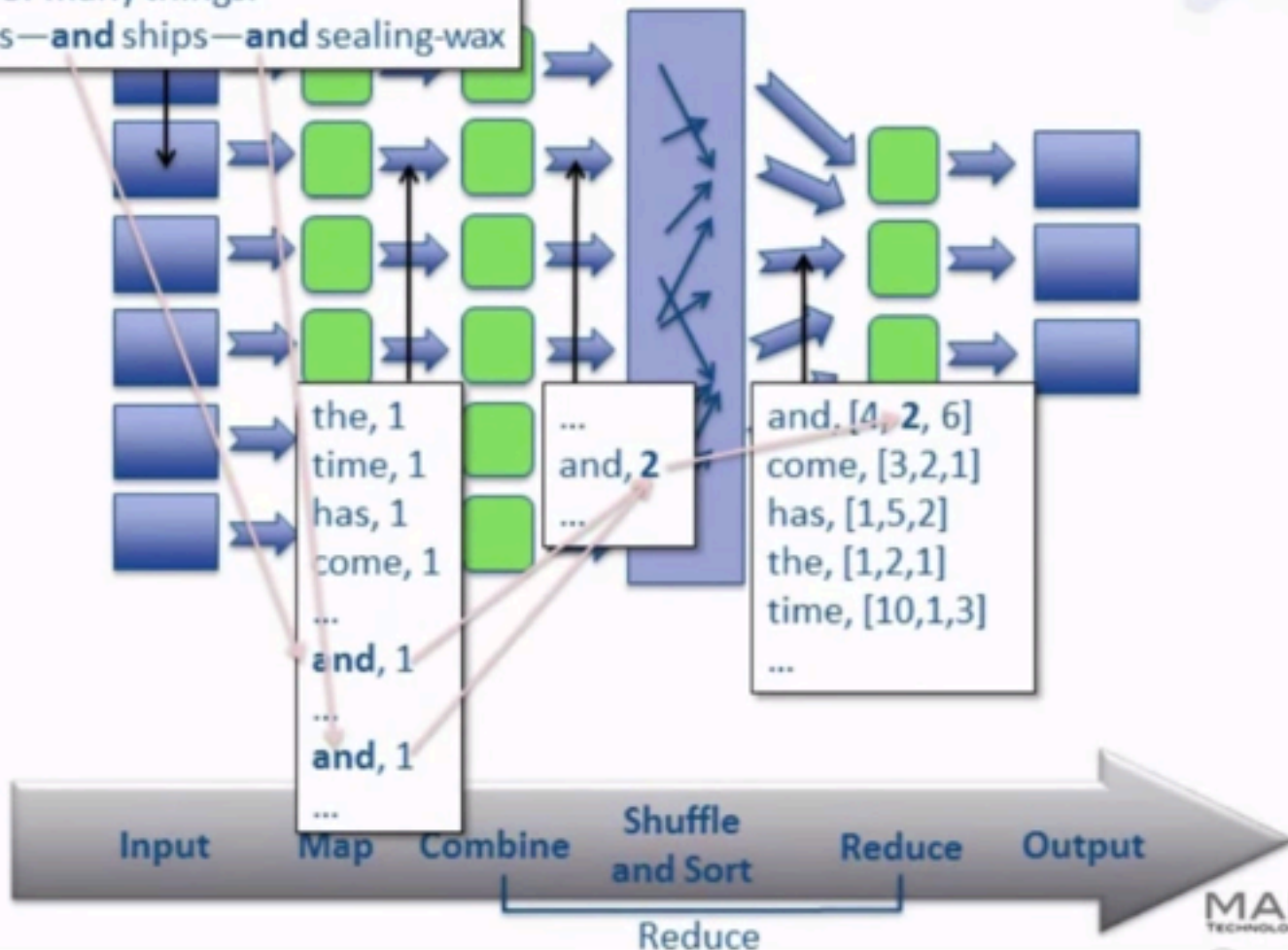
# Example: Word Count



Reduce phase will sum the values to create a reduced representation. Thus, multiple instances of same key are combined.

# Example: Word Count

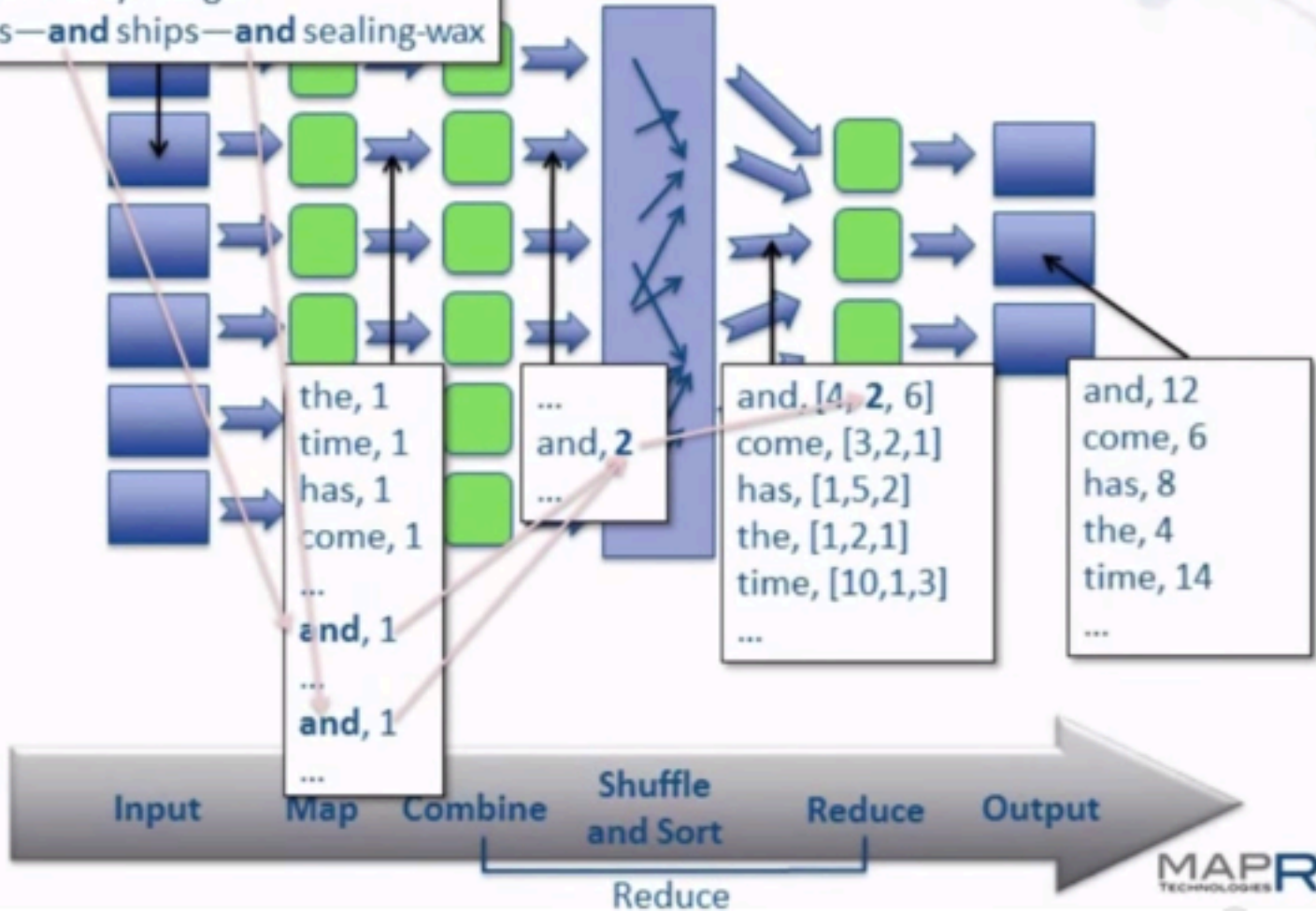
"The time has come," the Walrus said,  
"To talk of many things:  
Of shoes—and ships—and sealing-wax



Shuffle and Sort: Gather all instances of similar keys from all the tasks  
In {and, [4, 2, 6]} the other values are from other books/tasks

# Example: Word Count

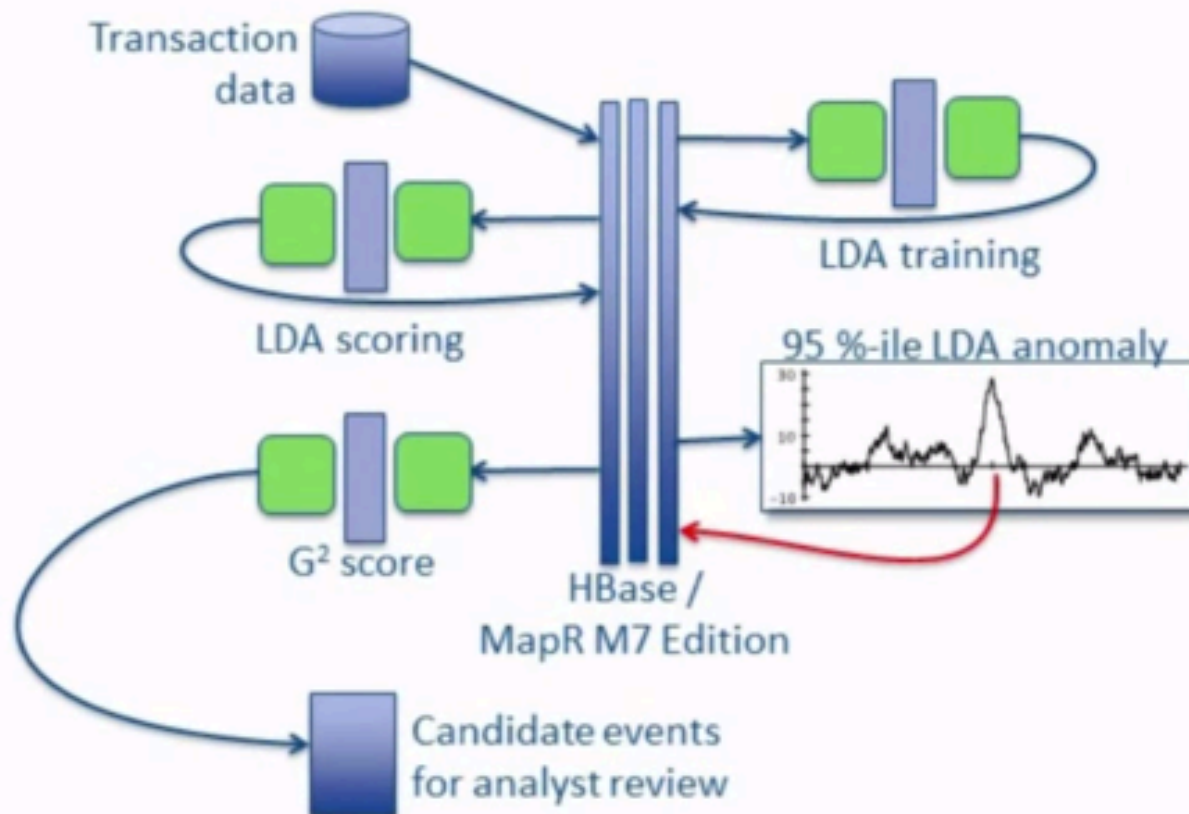
"The time has come," the Walrus said,  
"To talk of many things:  
Of shoes—**and** ships—**and** sealing-wax



Compute final result and save to disk.

# Variation: Multiple MapReduces

*Example: Fraud Detection in User Transactions*



# MapReduce - word count example

```
function map(String name, String document):  
  for each word w in document:  
    emit(w, 1)
```

```
function reduce(String word, Iterator partialCounts):  
  totalCount = 0  
  for each count in partialCounts:  
    totalCount += count  
  emit(word, totalCount)
```



# MapReduce - Java API

- **Mapper:**

```
void map(WritableComparable key,  
        Writable value,  
        OutputCollector output,  
        Reporter reporter)
```

- **Reducer:**

```
void reduce(WritableComparable key,  
           Iterator values,  
           OutputCollector output,  
           Reporter reporter)
```





# What about failed tasks?



- Tasks will fail
- JT will retry failed tasks up to N attempts
- After N failed attempts for a task, job fails
- Some tasks are slower than other
- Speculative execution is JT starting up multiple of the same task
- First one to complete wins, other is killed



# MapReduce is not good for...

- Jobs that need shared state/coordination
  - Tasks are shared-nothing
  - Shared-state requires scalable state store
- Low-latency jobs
- Jobs on small datasets
- Finding individual records





# Hadoop Distributed File System



- Designed to hold large amounts of data and provide access to this data to many clients across network
- Hadoop DFS is designed to handle data spread across multiple machines
- Data redundancy: If individual machines fail, data still should be available
- Provides fast and scalable access to this information. Can add machines in the cluster while maintaining integrity of data
- Works well with Hadoop MapReduce framework



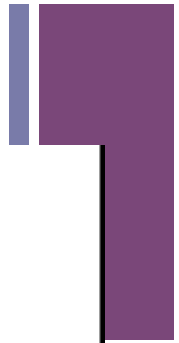
# Trade-offs of HDFS



- Applications are assumed to perform long sequential streaming reads from files
- Data will be written to the HDFS once and then read several times; updates to files after they have already been closed is not supported
- Does not provide mechanism for local caching: Just re-read the data from disk
- Designed based on Google File System
- Cannot interact with files using normal Unix tools like: ls, cp, mv. It has a separate namespace
- The management information is handled by a single machine. It has redundant information to protect it from failure of that machine.



# Hadoop Distributed File System



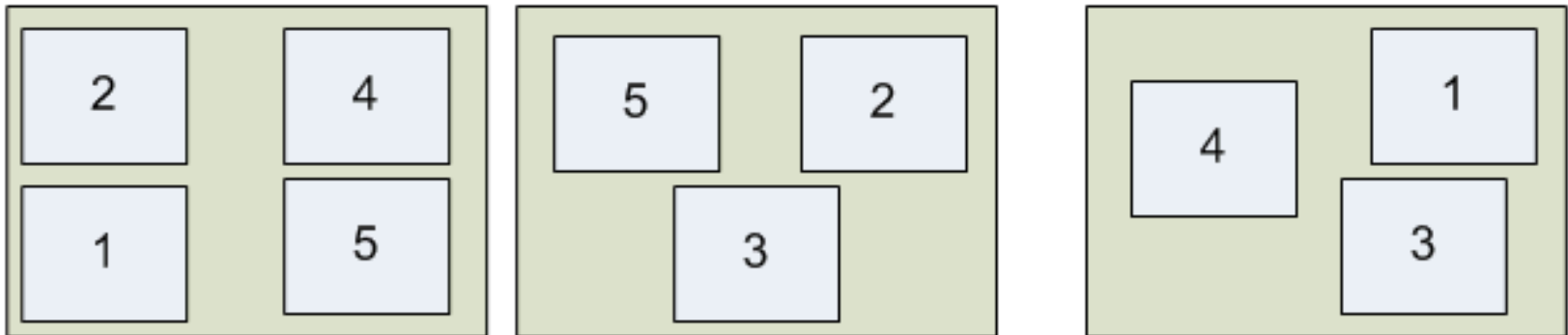
NameNode:  
Stores metadata only

METADATA:

/user/aaron/foo → 1, 2, 4

/user/aaron/bar → 3, 5

DataNodes: Store blocks from files

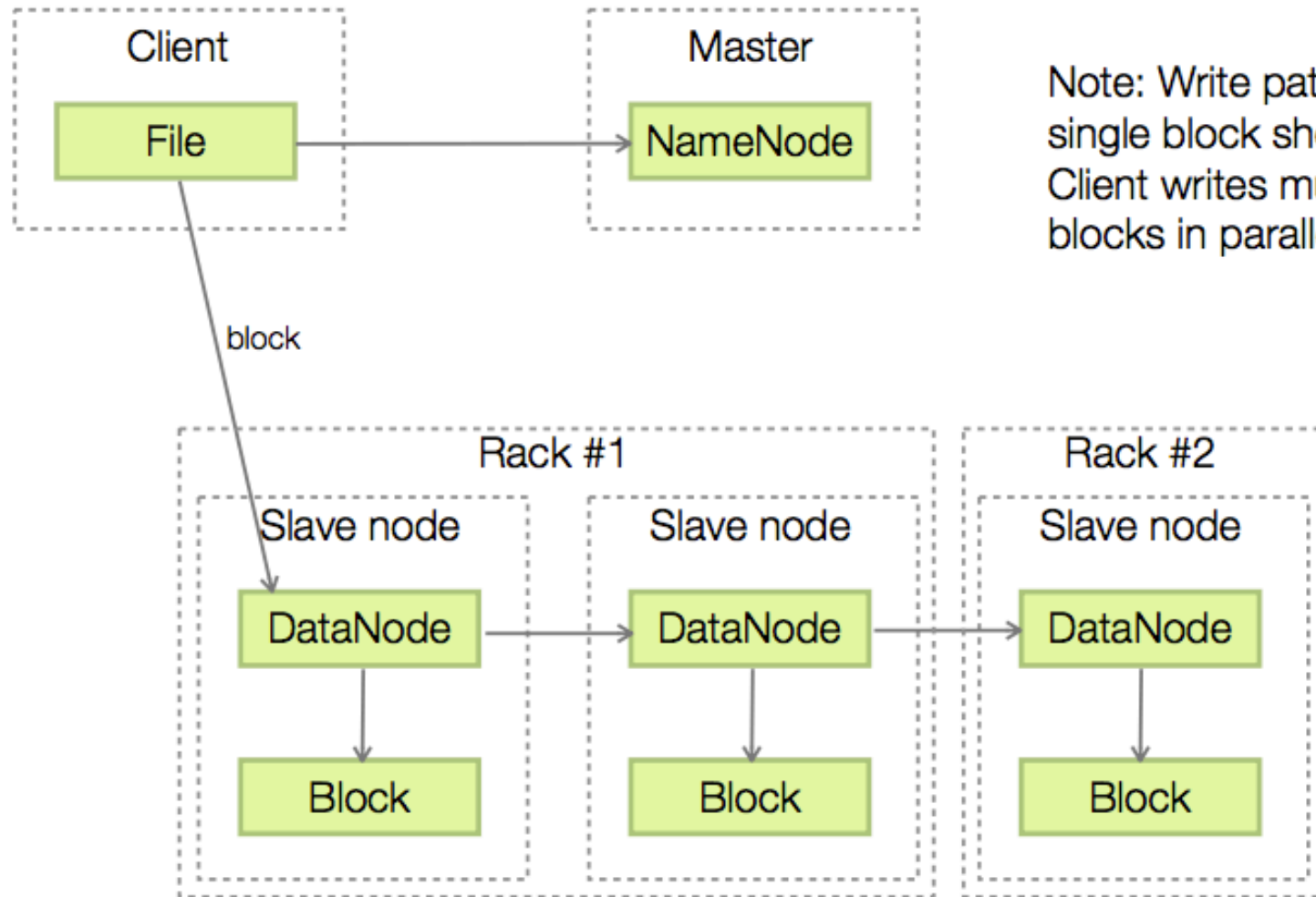


DataNodes holding blocks of multiple files with a replication factor of 2.

The NameNode maps filenames into block ids.

This redundancy in information helps when individual nodes fail

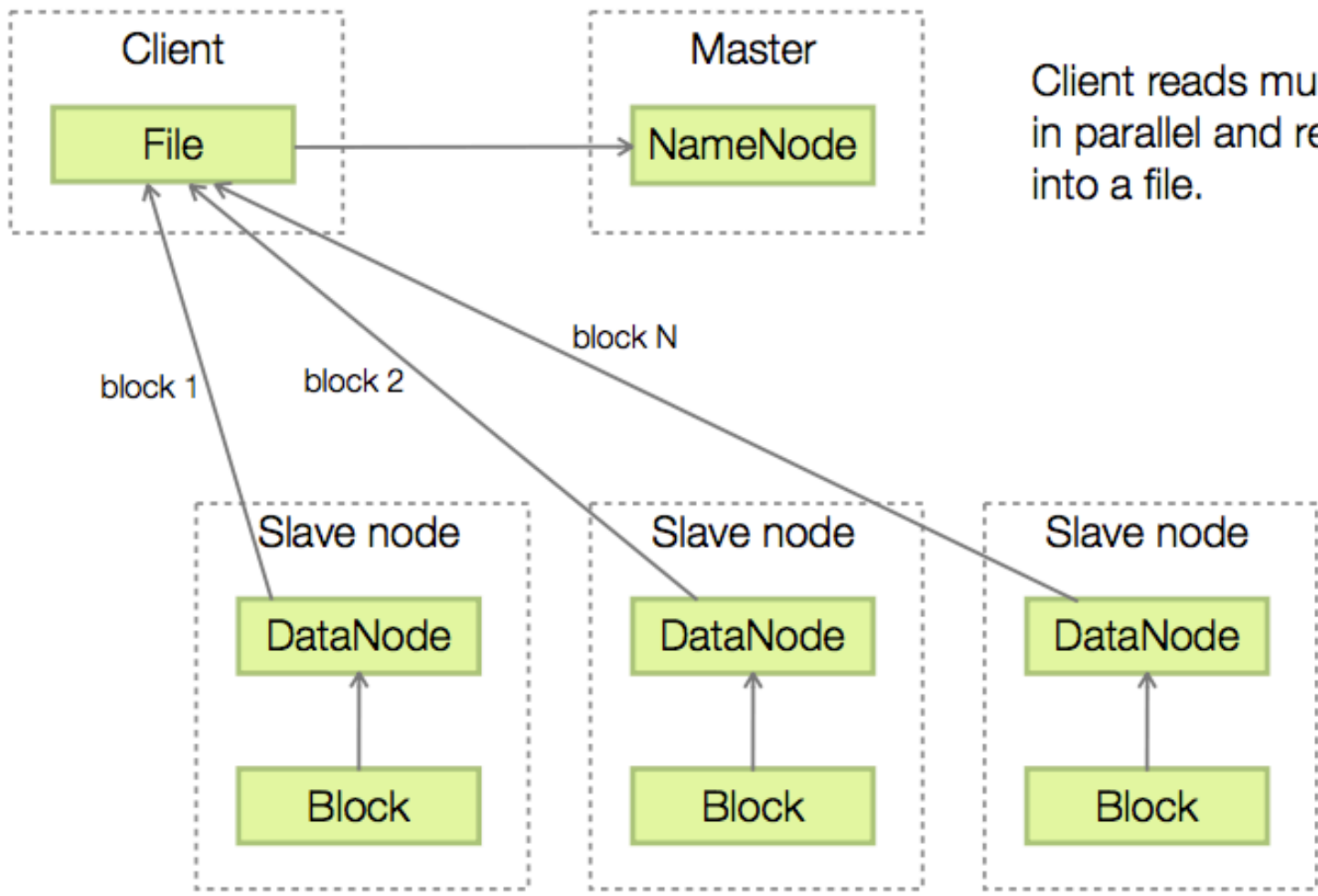
# HDFS - writes



Note: Write path for a single block shown. Client writes multiple blocks in parallel.



# HDFS - reads



Client reads multiple blocks in parallel and re-assembles into a file.



# What about DataNode failures?

- DNs check in with the NN to report health
- Upon failure NN orders DNs to replicate under-replicated blocks



Credit: <http://www.flickr.com/photos/18536761@N00/367661087/>



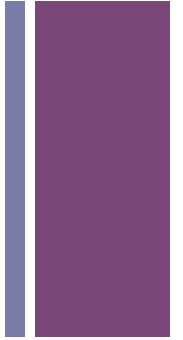
# Hadoop Map-Reduce and HDFS: Advantages

---

- Distribute data *and* computation
  - Computation local to data avoids network overload
- Tasks are independent
  - Easy to handle partial failures - entire nodes can fail and restart
  - Avoid crawling horrors of failure-tolerant synchronous distributed systems
  - Speculative execution to work around stragglers
- Linear scaling in the ideal case
  - Designed for cheap, commodity hardware
- Simple programming model
  - The “end-user” programmer only writes map-reduce tasks



# Amazon ElasticMapReduce (EMR)



- Amazon comes to your rescue again
- Super easy to use. Generate keypair and you are good to go
- Can access API in multiple languages.
- You can start with a 10-node Hadoop cluster and scale your application
- Similar web console for launching EMR

# + Pig

- Pig is a platform for analyzing large dataset
- Pig lets you specify a sequence of data transformations such as merging data sets, filtering them and applying functions to records
- Purpose of Pig is to answer queries over semi-structured data such as log files
- Pig is high-level language for writing queries over this sort of data
- Programming language used to write Pig queries is called *Pig Latin*



# What is Pig?

- Pig is a **scripting language**
  - No compiler
  - Rapid prototyping
  - Command line prompt (grunt shell)
- Pig is a **domain specific language**
  - No control flow (no if/then/else)
  - Specific to data flows
    - Not for writing ray tracers
    - For the distribution of a pre-existing ray tracer



# + Pig Latin Datatypes



- An **atom** is atomic value (e.g. “fish”). (similar to string in python)
- A **tuple** is a record of multiple values with fixed arity e.g. (“dog”, “sparky”) (similar to tuple in python)
- A **data bag** is collection of arbitrary number of values { (“dog”, “sparky”), (“fish”, “goldie”) } (similar to list in python but with differences)
- A **data map** is collection with a lookup function translating to keys and values e.g. [‘age’: 25] (similar to dictionary in python)



# Pig and MapReduce

- **MapReduce requires programmers**
  - Must think in terms of map and reduce functions
  - More than likely will require Java programmers
- **Pig provides high-level language that can be used by**
  - Analysts
  - Data Scientists
  - Statisticians
  - Etc...
- **Originally implemented at Yahoo! to allow analysts to access data**



# Pig's Features

- **Join Datasets**
- **Sort Datasets**
- **Filter**
- **Data Types**
- **Group By**
- **User Defined Functions**
- **Etc..**

# Pig's Use Cases

- **Extract Transform Load (ETL)**
  - Ex: Processing large amounts of log data
    - clean bad entries, join with other data-sets
- **Research of “raw” information**
  - Ex. User Audit Logs
  - Schema maybe unknown or inconsistent
  - Data Scientists and Analysts may like Pig's data transformation paradigm



# Pig Components

- **Pig Latin**
  - Command based language
  - Designed specifically for data transformation and flow expression
- **Execution Environment**
  - The environment in which Pig Latin commands are executed
  - Currently there is support for Local and Hadoop modes
- **Pig compiler converts Pig Latin to MapReduce**
  - Compiler strives to optimize execution
  - You automatically get optimization improvements with Pig updates

# Execution Modes



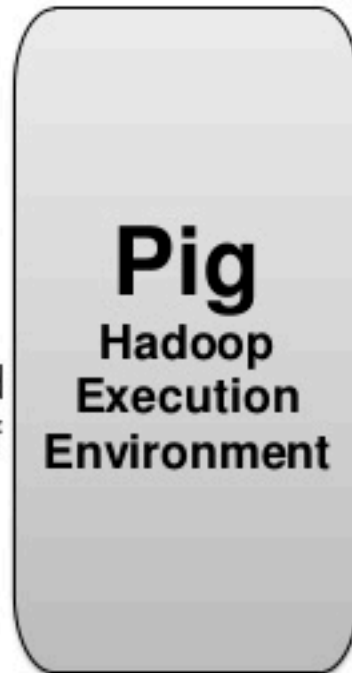
- **Local**
  - Executes in a single JVM
  - Works exclusively with local file system
  - Great for development, experimentation and prototyping
- **Hadoop Mode**
  - Also known as MapReduce mode
  - Pig renders Pig Latin into MapReduce jobs and executes them on the cluster
  - Can execute against semi-distributed or fully-distributed hadoop installation

# Hadoop Mode

```
-- 1: Load text into a bag, where a row is a line of text
lines = LOAD 'training/playArea/hamlet.txt' AS
(line:chararray);
-- 2: Tokenize the provided text
tokens = FOREACH lines GENERATE
flatten(TOKENIZE(line)) AS token:chararray;
```

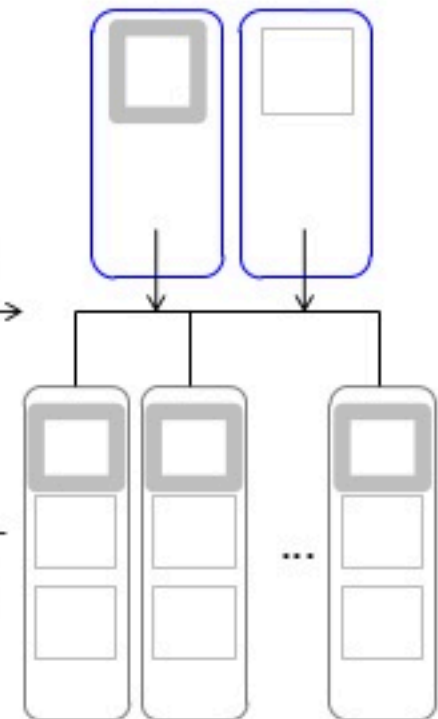
**PigLatin.pig**

Parse Pig script and  
compile into a set of  
MapReduce jobs



Execute on  
Hadoop Cluster

Monitor/Report



**Hadoop  
Cluster**

# + Loading Data in Pig

User provided parsing function

```
queries = LOAD 'query_log.txt'  
          USING myLoad()  
          AS (userId, queryString, timestamp)
```

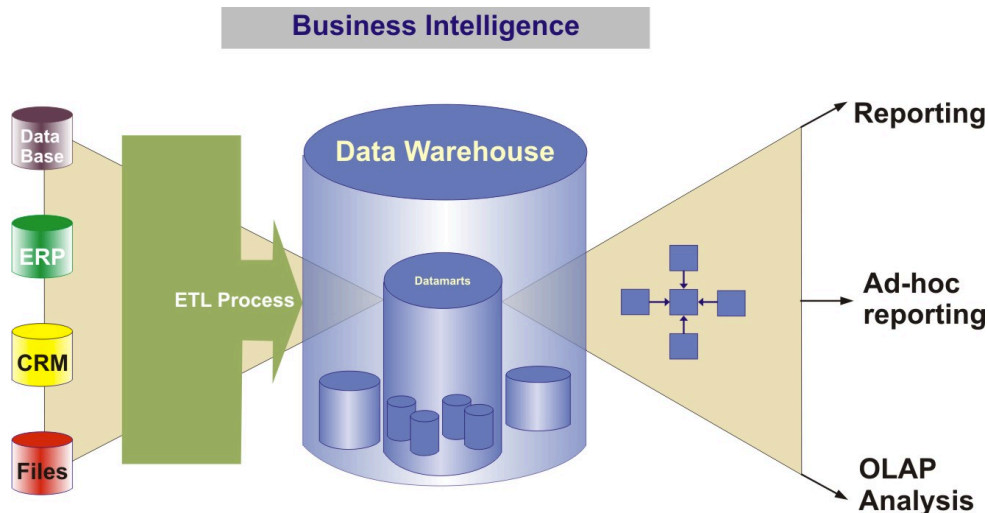
The user defined function need not be provided. A default (PigStorage() ) exists, but Pig provides you an option if you want to use it.

# Pig Latin – Diagnostic Tools

- **Display the structure of the Bag**
  - grunt> DESCRIBE <bag\_name>;
- **Display Execution Plan**
  - Produces Various reports
    - Logical Plan
    - MapReduce Plan
  - grunt> EXPLAIN <bag\_name>;
- **Illustrate how Pig engine transforms the data**
  - grunt> ILLUSTRATE <bag\_name>;

# + Business Intelligence Tools

- Lots of them
  - Jaspersoft, Excel, Talend, Penthao, RapidMiner, KNIME, etc.
- Application software designed to retrieve, analyze and report data
- Mostly visual. Geared towards enterprise applications
- Many domain specific/application specific/proprietary





# Features of Business Intelligence Tools



- Data management strategy
- Analytics, Reporting, scorecard and strategy management
- Highly advanced/specialized calculation engines, business user experience
- “What-if” analysis to develop applications that can forecast business performance
- Can operate of thousands of simultaneous users and terabytes of information
- GUI based interface

The screenshot displays the Talend Open Studio interface. The main workspace shows a Business Model diagram with the following components and relationships:

- Input:** A yellow box labeled 'input' with an arrow pointing to 'Talend Jobs'.
- Terminal:** A yellow box labeled 'terminal' with an arrow pointing to 'Talend Jobs'.
- Database:** A blue cylinder labeled 'database' with a bidirectional arrow to 'Talend Jobs'.
- Talend Jobs:** A green gear icon representing the core process.
- Partner:** A purple trapezoid with an arrow pointing from 'Talend Jobs' to it.

Annotations in the diagram include:

- A yellow callout box: "A Business Model is a non technical view of a business need in data flow management. The Business Modeler is at the core of the Top/Down approach: it allows any of the key players to take part to the project design. BMs offer a macroscopic view of the project."
- A green callout box: "Select a shape, go on the view assignment and open the associated items (double clic)."
- A yellow callout box: "1 assigned Metadata POrders (Job)".

The left sidebar shows a tree view with categories like Business Models, Job Designs, Contexts, Code, SQL Templates, Metadata, and Documentation. The right sidebar shows a Palette with various shapes like Decision, Action, Terminal, Data, Document, Input, List, Datasource, Actor, Ellipse, and Gear. The bottom section shows a table for the 'aBusinessModel 0.1' assignment:

Appearance	Type	Name	Comment
Assignment	Job	POrders	

Allows application integration,  
cloud integration





Talend Data Quality

File Edit Window Help

DQ Repository

- Data Profiling
  - Analyses(35)
    - Account\_Backup\_Comparison 0.1
    - Age\_Analysis 0.1
    - Age\_Average 0.1
    - Column\_Analysis 0.1
    - Column\_Content\_Comparison 0.1
    - DB2\_ContentAnalysis 0.1
    - Functional\_Dependencies 0.1
    - MDM\_Analysis 0.1
    - MDM\_Column\_Analysis 0.1
    - MDM\_Column\_Analysis1 0.1
    - MDM\_Column\_Analysis2 0.1
    - MDM\_Overview\_Analysis 0.1
    - Overview\_Mysql 0.1
    - Redundancy\_KeyMatching 0.1
    - SQL\_Content\_Analysis 0.1
    - Set\_of\_Columns 0.1
    - Subscription\_Date 0.1
    - account\_backup\_comparison 0.1
    - account\_number 0.1
    - birthdate\_country\_correlation 0.1
    - catalog\_Analysis 0.1
    - column\_analysis 0.1
    - column\_analysis1 0.1
    - column\_set\_analysis 0.1
    - columns\_client 0.1
    - copy of overview\_analysis 0.1
    - country\_maritalstatus\_correlation 0.1
    - customer\_data\_analysis 0.1
    - date\_pattern 0.1
    - email 0.1
    - overview2 0.1
    - overview\_analysis 0.1
    - temperature\_correlation 0.1
    - test 0.1

MDM\_Column\_Analysis 0.1

### Column Analysis

Analysis Metadata

Set the properties of analysis.

Name: MDM\_Column\_Analysis

Purpose:

Description:

Author: user@company.com

Status: development

Analyzed Columns

Connection: MDM\_Connection

Select columns to analyze

Select indicators for each column

Analyzed Columns	Datamining Type	Pattern	UDI	Operation
username (string)	Other			
Row Count				✗
Null Count				✗
Distinct Count				✗
Unique Count				✗
Duplicate Count				✗
id (string)	Other			
Row Count				✗
Null Count				✗
Distinct Count				✗
Unique Count				✗
Duplicate Count				✗
givenname (string)	Other			
Row Count				✗
Null Count				✗
Distinct Count				✗
Unique Count				✗
Duplicate Count				✗

Graphics

Refresh the graphics

Go

Column: username

Indicator	Value
Row Count	2
Null Count	0
Distinct Count	2
Unique Count	2
Duplicate Count	0

Column: id

Indicator	Value
Row Count	2
Null Count	0
Distinct Count	1
Unique Count	0
Duplicate Count	1

Column: givenname

Indicator	Value
Row Count	2

Detail View

General

Name: MDM\_Column\_Analysis

Purpose: Default

Description: Default

Type: Multiple Column Analysis

Number of analyzed elements: 3

Connection: MDM\_Connection

Technical

Data Filter

Edit the data filter:

Where

Analysis Settings | Analysis Results

Run Analysis

### CompoundLayoutTest

0% - 33% 33% - 67% 67% - 100%

Brand	LOB	LOB	Product Type	Target Quantity	Target
BizTech	Communication	Communication	Cell Phones	56,875	606,986
			Smart Phones	54,723	562,496
	Electronics	Electronics	Accessories	29,101	341,052
			Audio	72,237	751,938
FunPod	Digital	Digital	Camera	82,052	531,463
	Games	Games	Fixed	39,690	357,782
			Portable	50,620	444,068
HomeView	Services	Services	Install	7,373	68,013
			Maintenance	9,122	92,301
	TV	TV	LCD	33,456	364,720
			Plasma	37,582	379,180

### CompoundLayoutTest

#### Target, Target Quantity

- Accessories
- Audio
- Camera
- Cell Phones
- Fixed
- Install
- LCD
- Maintenance
- Plasma
- Portable
- Smart Phones

Brand	LOB	LOB	Product Type	Target Quantity	Target
BizTech	Communication	Communication	Cell Phones	56,875	606,986
			Smart Phones	54,723	562,496
	Electronics	Electronics	Accessories	29,101	341,052
			Audio	72,237	751,938
FunPod	Digital	Digital	Camera	82,052	531,463
	Games	Games	Fixed	39,690	357,782
			Portable	50,620	444,068
HomeView	Services	Services	Install	7,373	68,013
			Maintenance	9,122	92,301
	TV	TV	LCD	33,456	364,720
			Plasma	37,582	379,180

# Oracle Business Intelligence Suite

BI Analytics

BI Analytics

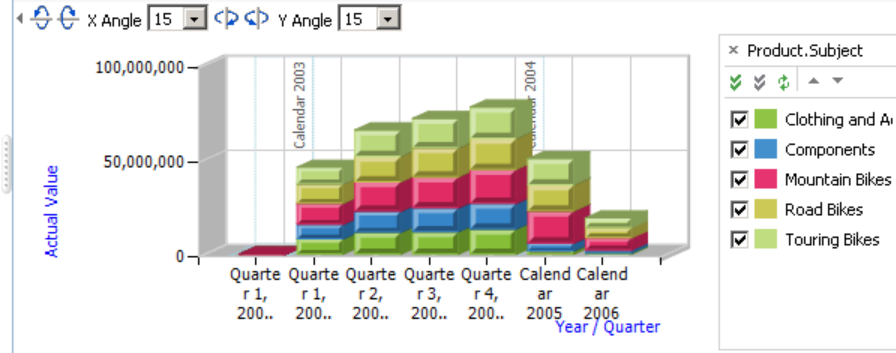
Home | Service - Resource Details | Service - Forecast Accuracy | Service - Average Time Taken | Sales Value - by Account | Quota Achievement Rate

Quota Achievement Rate

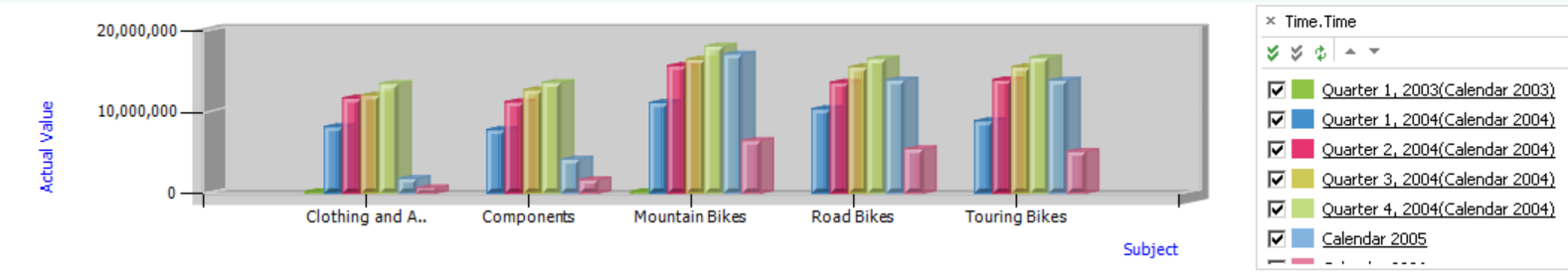
Drop a Filter Condition Here

CRM Analyzer		Subject		
Year	Quarter	Clothing and Accessories	Components	Mountain Bikes
		Actual Value	Actual Value	Actual Value
Calendar 2003	Quarter 1, 2003	\$5,825.98	-	\$5,825.98
<b>Calendar 2003 Subtotal</b>		\$5,825.98	-	\$5,825.98
Calendar 2004	Quarter 1, 2004	\$8,129,017.04	\$7,777,462.70	\$11,074,670.00
	Quarter 2, 2004	\$11,640,037.67	\$11,178,467.61	\$15,624,050.00
	Quarter 3, 2004	\$12,002,893.71	\$12,659,891.05	\$16,400,790.00
Calendar 2004	Quarter 4, 2004	\$13,409,330.02	\$13,554,302.54	\$18,012,300.00
	<b>Calendar 2004 Subtotal</b>	\$45,181,278.44	\$45,170,123.90	\$61,111,830.00
Calendar 2005		\$1,784,129.59	\$4,138,510.14	\$17,046,420.00
Calendar 2006		\$610,386.53	\$1,649,587.90	\$6,444,020.00

Quarterly Product Volume



Percentage of Achievement



Personalize Workplace ...

- Workplace
- Sales
- Marketing
- Service
- Settings
- Resource Center
- BI Analytics

Sheet1



FILE HOME

Themes Map Labels Data Shapes Add Scene Play Tour Chart Show All Legends Textbox Tour Editor Time Line Time Decorator Task Panel Add Layer Refresh Data Find Location Copy Screen

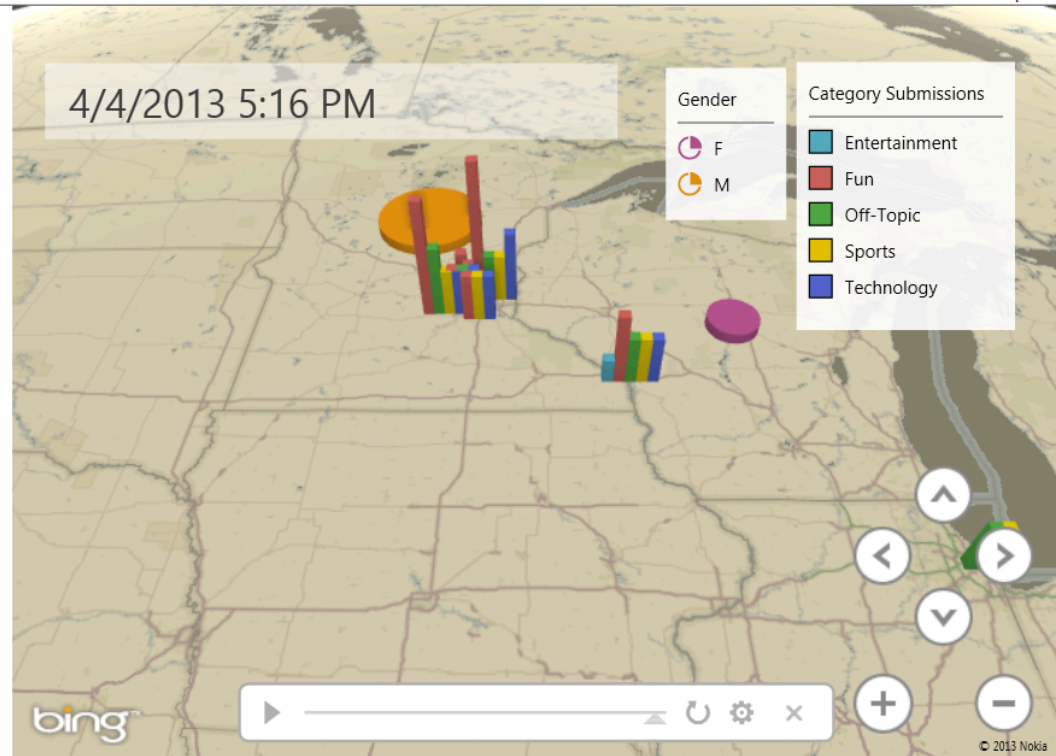
Map Tour Insert View Data Find Capture

Poll Submissions

1 4/4/2013 5:16 PM Categories and Age

2 4/4/2013 5:16 PM

3 4/4/2013 5:16 PM Age Range Responses



Layer 3

Choose the field(s) that make up the geography you would like to visualize.

Category

- CategoryID
- CategoryName

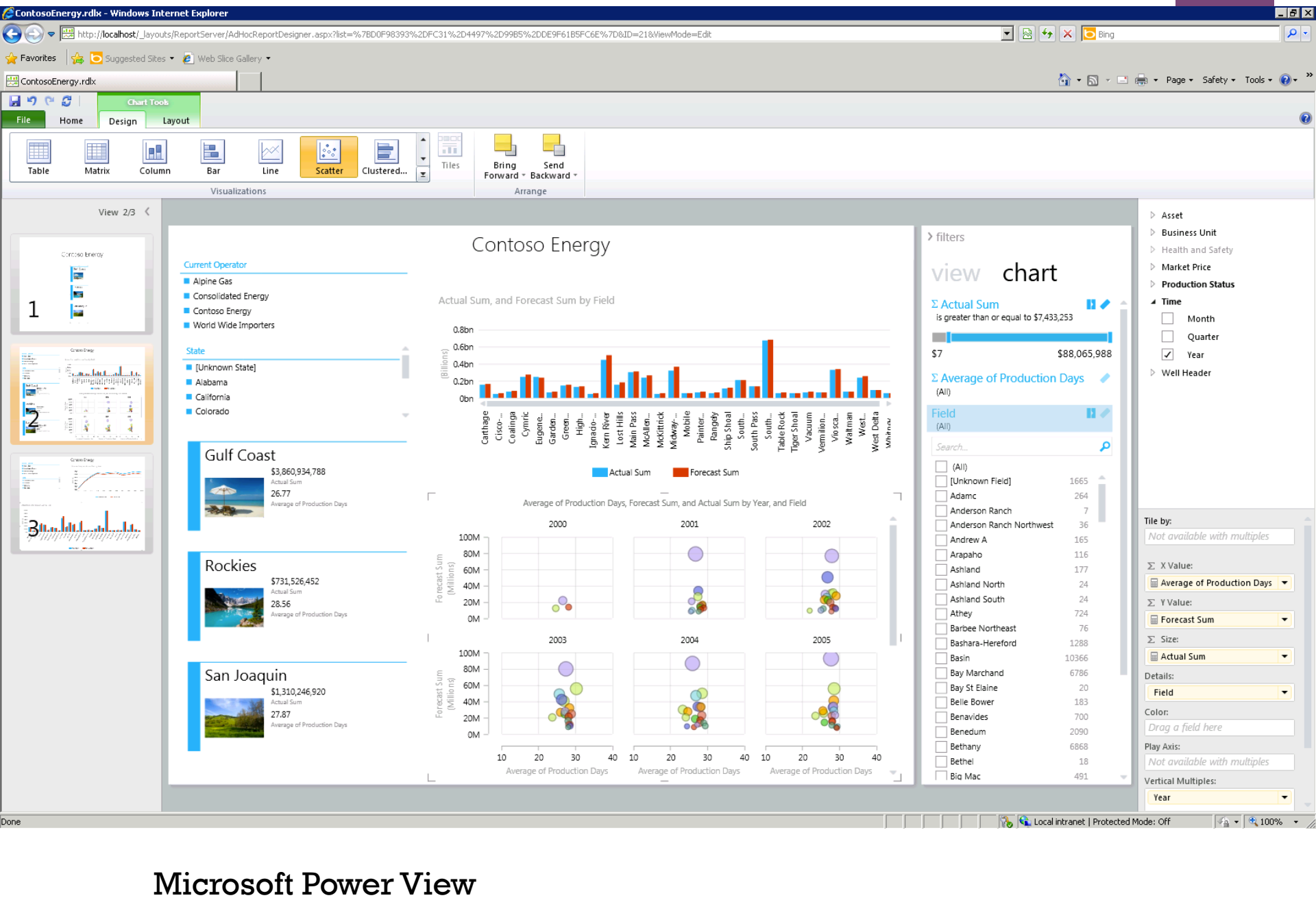
Dates

- CalendarMonth
- CalendarMonthName
- CalendarMonthNOD

GEOGRAPHY

Map It

# Microsoft Power Maps



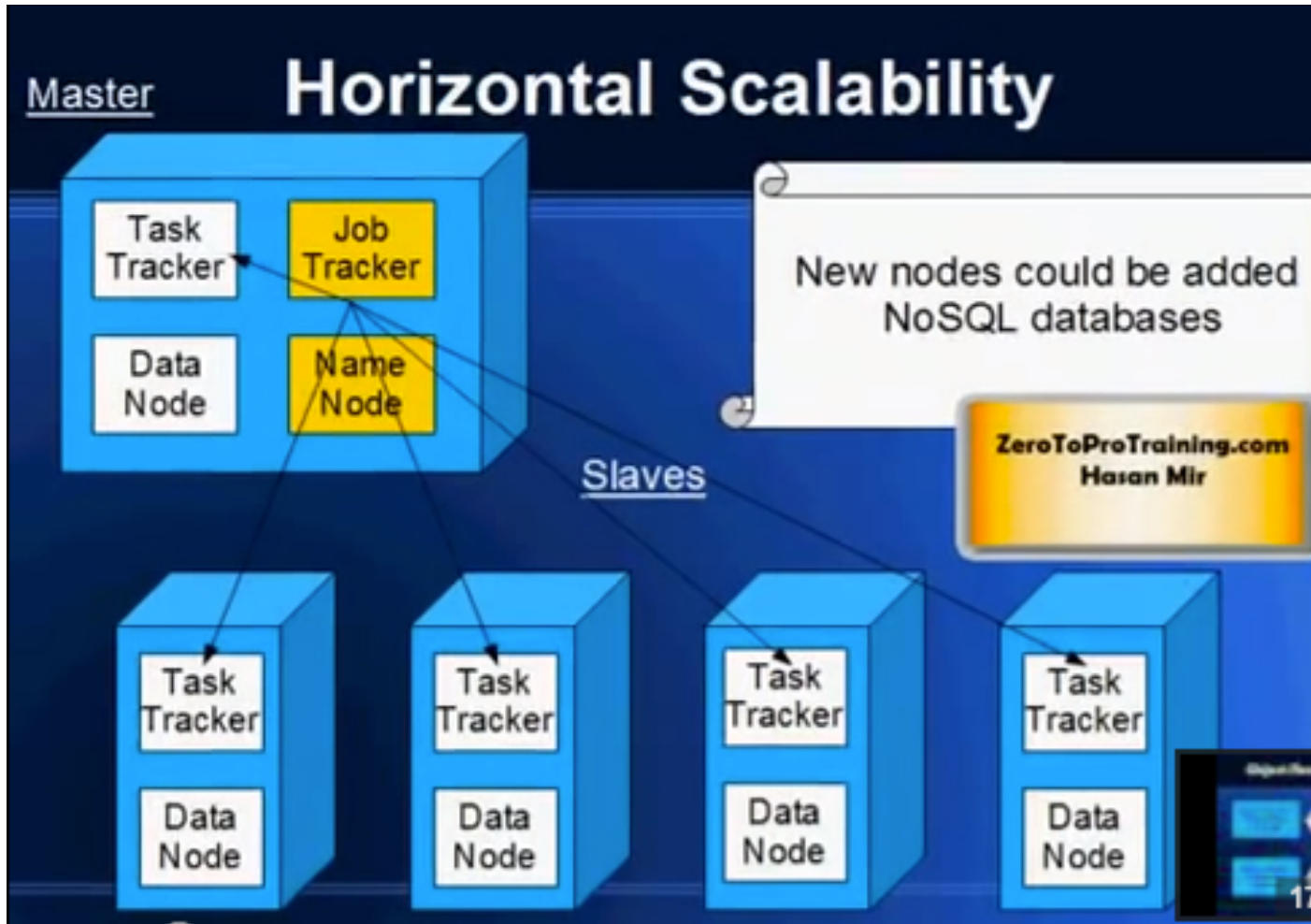
# Storing Documents ...



# + Document Store

- Document oriented database is a computer program designed for storing, retrieving and managing document oriented information
- E.g. formats: XML, YAML, JSON, BSON
- Allows structured queries and retrieval. Makes storing and retrieving data easier.
- Lots of packages available for reading such documents
- **Keys:** Documents are addressed in the database via a unique key that represents that document. Database retains an index on the key to speed up document retrieval
- **Retrieval:** Database offers an API or query language
- E.g. Cassandra, CouchDB, MongoDB, Lotus Notes etc.
- RDBMS/Flat File Systems cannot handle big data. Cannot handle horizontal scalability. Hence, NoSQL databases





Keep on adding more computers as you need more compute power i.e.  
Performance is linearly proportional to no. of computers.  
Relational Databases cannot handle horizontal scalability



# + Structured Documents

```
personal.xml x
4 <personnel>
5   <person id="Big.Boss">
6     <name>
7       <family>Boss</family>
8       <given>Big</given>
9     </name>
10    <email>chief@oxygenxml.com</email>
11    <link subordinates="one.worker"/>
12  </person>
13  <person id="one.worker">
14    <name>
15      <family>Worker</family>
16      <given>One</given>
17    </name>
18    <email>one@oxygenxml.com</email>
19    <link manager="Big.Boss"/>
20  </person>
21  <person id="two.worker">
22    <name>
```

XML

Extensible Markup Language

```
personal.json x
1 {"personnel": {"person": [
2   {
3     "id": "Big.Boss",
4     "name": {
5       "family": "Boss",
6       "given": "Big"
7     },
8     "email": "chief@oxygenxml.com",
9     "link": {"subordinates": "one.worker"}
10  },
11  {
12    "id": "one.worker",
13    "name": {
14      "family": "Worker",
15      "given": "One"
16    },
17    "email": "one@oxygenxml.com",
18    "link": {"manager": "Big.Boss"}
19  },
20  }
21  ]
22  }
```

JSON

JavaScript Object Notation



# MongoDB

- ◇ Humongous
- ◇ Document oriented database using JSON document syntax
- ◇ Features:
  - ◇ Flexibility
  - ◇ Power
  - ◇ Scaling
  - ◇ Ease of Use
  - ◇ Built-in Javascript
- ◇ Clientele: Craigslist, eBay, Foursquare, SourceForge, and The New York Times.

# + MongoDB

- A record in MongoDB is a document, which is a data structure composed of field and value pairs
- MongoDB documents are similar to JSON objects. It is a NoSQL database
- Advantages:
  - Documents (i.e. objects) correspond to native data-types in many programming languages
  - Embedded documents and arrays reduce need for expensive joins
  - Allows Map-Reduce programming model. Written in C++ and open-source. Uses replication to maintain data consistency/availability

```
{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}
```

← field: value  
← field: value  
← field: value  
← field: value



# Document

```
{
  "_id": ObjectId("2jk48d2b7d284dad101e4bc9"),
  "First Name": "Hasan",
  "Last Name": "Mir",
  "Department": "20"
},
{
  "_id": ObjectId("2jk48d2b7d284dad101e4bc7"),
  "First Name": "Bill",
  "Last Name": "Gates",
}
{
  "_id": ObjectId("2jk48d2b7d284dad101e8912")
  "First Name": "Larry",
  "Last Name": "Ellison",
  "Department": "20"
  "Date Joined": "01-01-2013"
}
```

Document/  
Object

ZeroToProTraining.com  
Hasan Mir

# Object Record Conversion

ZeroToProTraining.com  
Hasan Mir

Object Oriented  
Programming  
Language

Covert records  
into objects

RDBMS

Object Oriented  
Programming  
Language

No conversion  
required

MongoDB

Works with Java, JavaScript, Python, Ruby, C#, PHP, C++ etc.

# + MongoDB Operations



## About this Cheat Sheet

The idea behind this is to have all (well, most) information from the above mentioned Tutorial immediately available in a very compact format. All commands can be used on a small data basis created in the insert-section. All information in this sheet comes **without the slightest warranty for correctness**. Use at your own risk. Have fun ☺!

## Basic Information

Download MongoDB	<a href="http://www.mongodb.org/downloads">http://www.mongodb.org/downloads</a>
JSON Specification	<a href="http://www.json.org/">http://www.json.org/</a>
BSON Specification	<a href="http://bsonspec.org/">http://bsonspec.org/</a>
Java Tutorial	<a href="http://www.mongodb.org/display/DOCS/Java+Tutorial">http://www.mongodb.org/display/DOCS/Java+Tutorial</a>

## Inserting Documents

```
db.ships.insert({name:'USS Enterprise-D',operator:'Starfleet',type:'Explorer',class:'Galaxy',crew:750,codes:[10,11,12]})
db.ships.insert({name:'USS Prometheus',operator:'Starfleet',class:'Prometheus',crew:4,codes:[1,14,17]})
db.ships.insert({name:'USS Defiant',operator:'Starfleet',class:'Defiant',crew:50,codes:[10,17,19]})
db.ships.insert({name:'IKS Buruk',operator:'Klingon Empire',class:'Warship',crew:40,codes:[100,110,120]})
db.ships.insert({name:'IKS Somraw',operator:'Klingon Empire',class:'Raptor',crew:50,codes:[101,111,120]})
db.ships.insert({name:'Scimitar',operator:'Romulan Star Empire',type:'Warbird',class:'Warbird',crew:25,codes:[201,211,220]})
db.ships.insert({name:'Narada',operator:'Romulan Star Empire',type:'Warbird',class:'Warbird',crew:65,codes:[251,251,220]})
```

## Finding Documents

<code>db.ships.findOne()</code>	Finds one arbitrary document
<code>db.ships.find().prettyPrint()</code>	Finds all documents and using nice formatting
<code>db.ships.find({}, {name:true, id:false})</code>	Shows only the names of the ships
<code>db.ships.findOne({'name':'USS Defiant'})</code>	Finds one document by attribute

## Basic Concepts & Shell Commands

<code>db.ships.&lt;command&gt;</code>	db – implicit handle to the used database ships – name of the used collection
<code>use &lt;database&gt;</code>	Switch to another database
<code>show collections</code>	Lists the available collections
<code>help</code>	Prints available commands and help

# + MongoDB Operations

## Updating Documents

<code>db.ships.update({name : 'USS Prometheus'}, {name : 'USS Something'})</code>	Replaces the whole document
<code>db.ships.update({name : 'USS Something'}, { \$set : {operator : 'Starfleet', class : 'Prometheus'}})</code>	sets / changes certain attributes of a given document
<code>db.ships.update({name : 'USS Something'}, { \$unset : {operator : 1}})</code>	removes an attribute from a given document

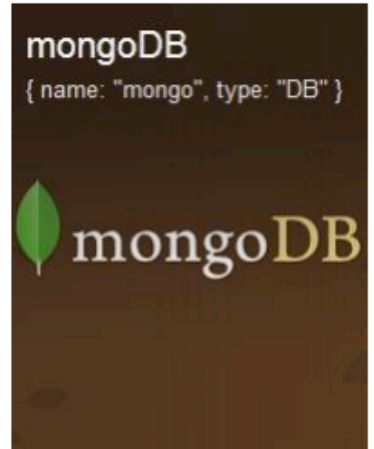
## Removing Documents

<code>db.ships.remove({name : 'USS Prometheus'})</code>	removes the document
<code>db.ships.remove({name : {\$regex : '^USS\\sE'}})</code>	removes using operator

*Each individual document removal is atomic with respect to a concurrent reader or writer. No client will see a document half removed.*

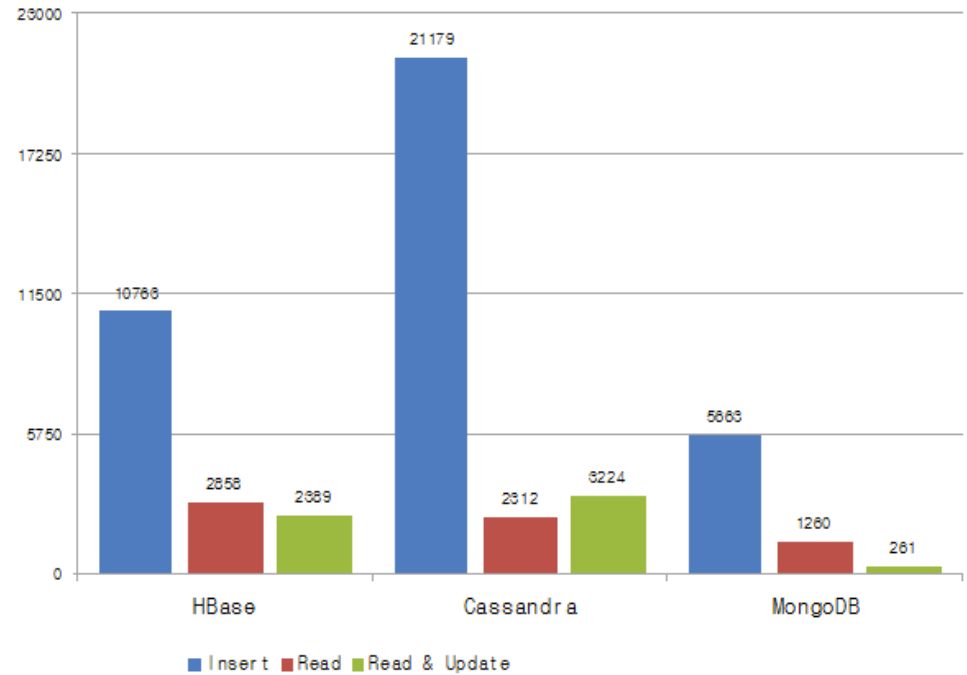
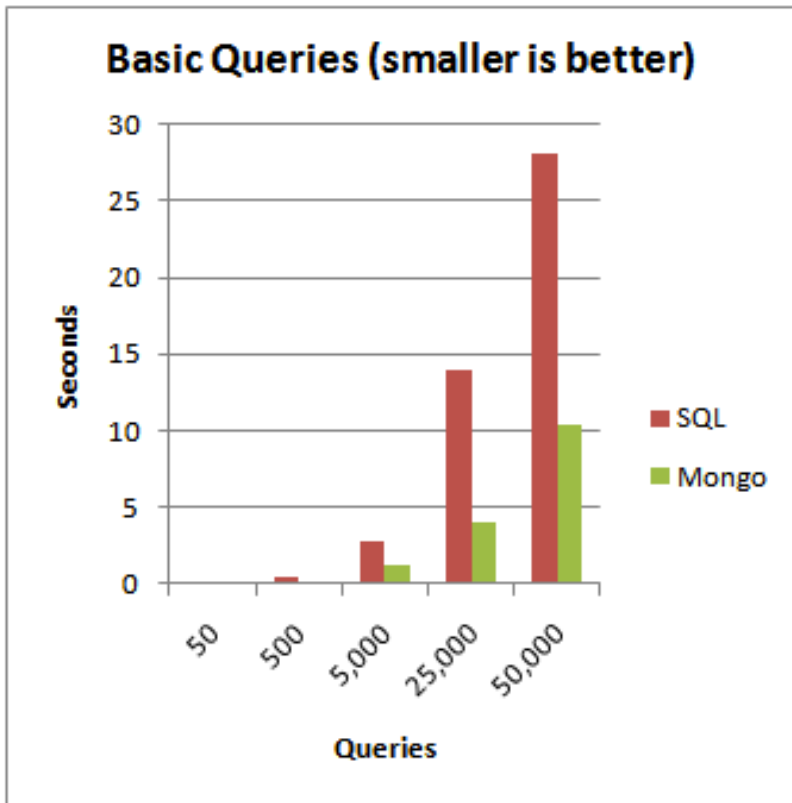
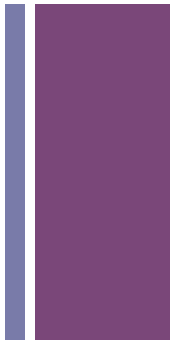
## Working with Indexes

Creating an index	<code>db.ships.ensureIndex({name : 1})</code>
Dropping an index	<code>db.ships.dropIndex({name : 1})</code>
Creating a compound index	<code>db.ships.ensureIndex({name : 1, operator : 1, class : 0})</code>
Dropping a compound index	<code>db.ships.dropIndex({name : 1, operator : 1, class : 0})</code>
Creating a unique compound index	<code>db.ships.ensureIndex({name : 1, operator : 1, class : 0}, {unique : true})</code>



G+ Community Page:  
<https://plus.google.com/u/0/communities/115421122548465808444>

# + Its is Fast..!





# Twitter Analytics and Hadoop





**Sarah Silverman** @SarahKSilverman

20 Sep

When ur relatives drive you crazy just close your eyes  
& pretend it's dialogue in a woody allen movie

[Details](#)



**mia farrow**

@MiaFarrow

Follow



[@SarahKSilverman](#) tried that. Didn't work  
RT When ur relatives drive u crazy just close  
yr eyes & pretend its dialogue in a woody  
allen movie

[Reply](#) [Retweet](#) [Favorite](#)

50+  
RETWEETS

50+  
FAVORITES



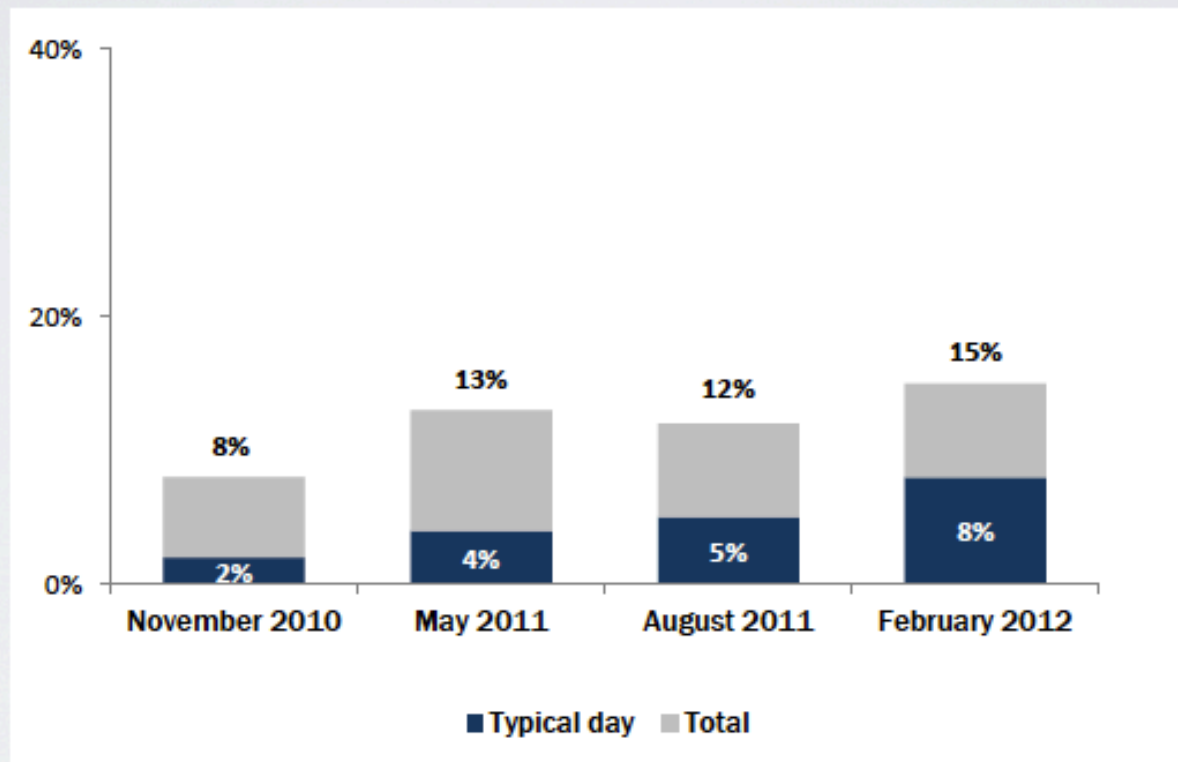
1:12 PM - 20 Sep 11 via Twitter for iPhone · [Embed this Tweet](#)

# WHAT IS TWITTER?

# TWITTER STATS

## Twitter usage over time

*% of internet users who use Twitter*



**Source:** Pew Research Center's Internet & American Life Project Winter 2012 Tracking Survey, January 20-February 19, 2012. N=2,253 adults age 18 and older, including 901 cell phone interviews. Interviews conducted in English and Spanish. Margin of error is +/-2.7 percentage points for internet users (n=1,729).

# About Twitter

- The fastest, simplest way to communicate
- More than 140M active users
  - Majority (also) mobile; 60% out of U.S.
- More than 400M twitter.com visitors
- More than 400M tweets/day (peak: 25K/sec)
- 1,000 employees (majority in San Francisco)
  - 50% engineers





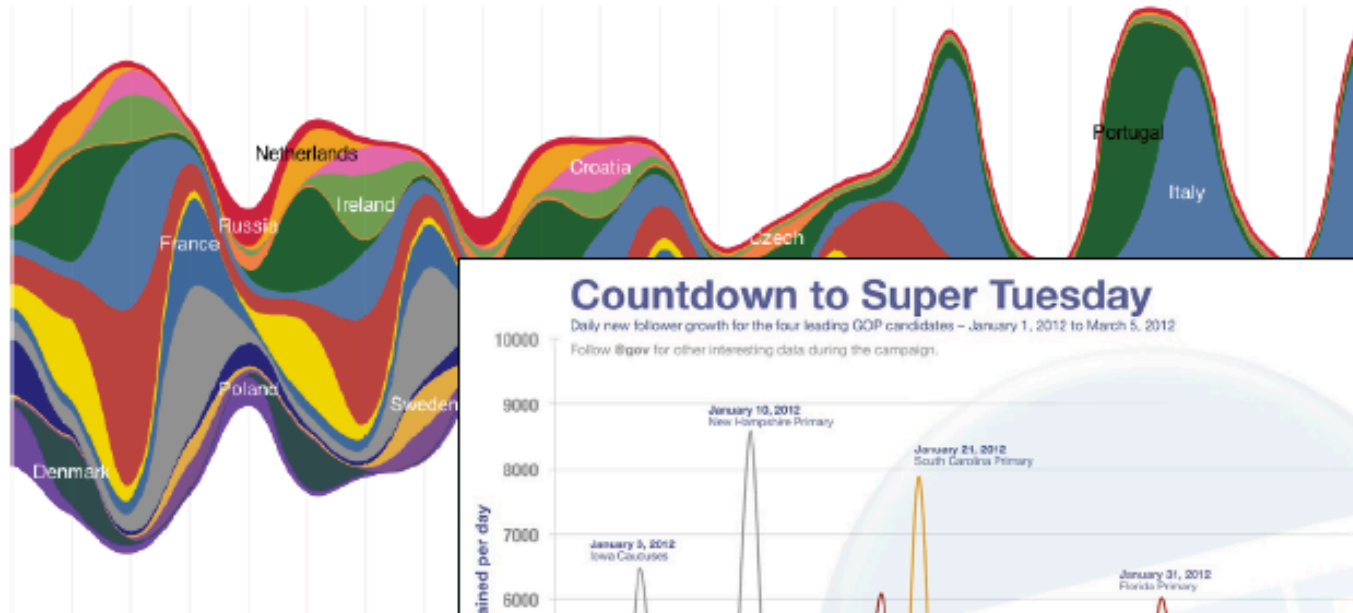
# Twitter data: time series



## #Euro2012

A summary for the action on Twitter during the European football tournament. [Tweet](#) 1,762

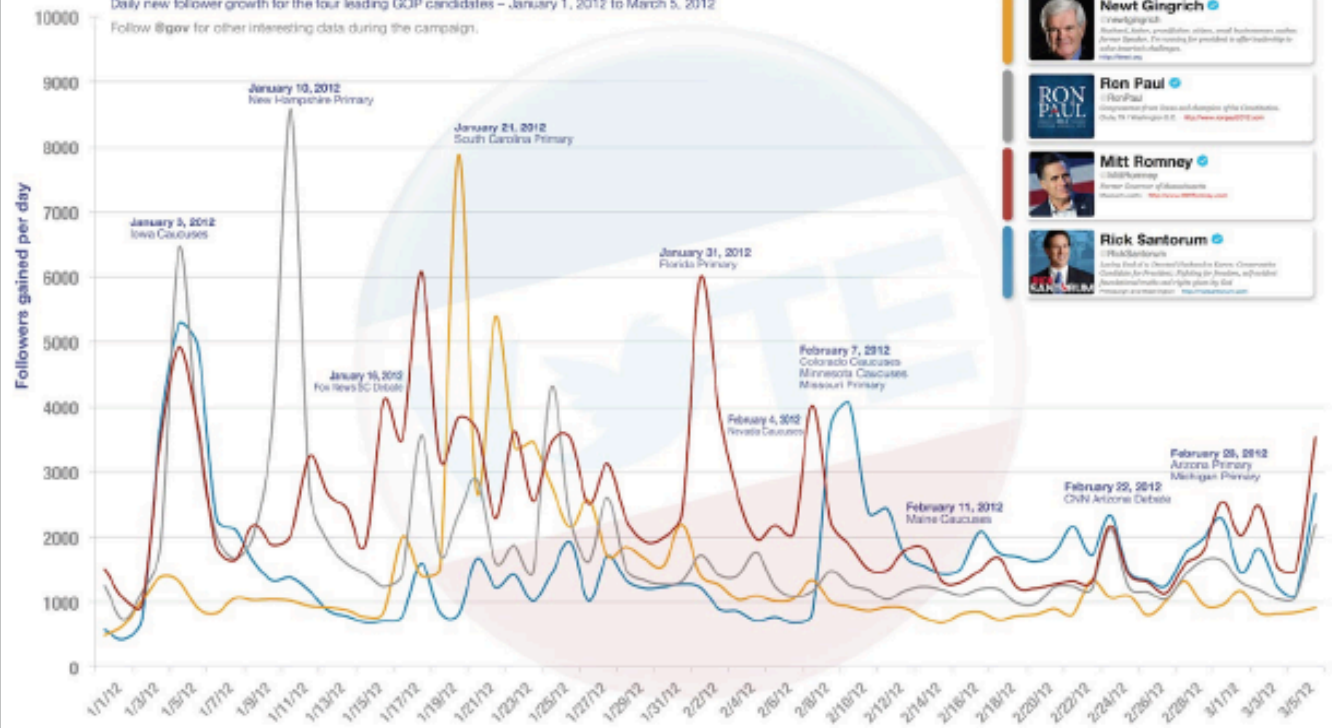
The Streamgraph below shows volume of Tweets during the #Euro2012 period. [Click on a team's name to see details.](#)



## Countdown to Super Tuesday

Daily new follower growth for the four leading GOP candidates - January 1, 2012 to March 5, 2012

Follow @gop for other interesting data during the campaign.





# + Examples of Analytic Tasks

## Search

Search results for 'SFGiants'. The interface shows a sidebar with navigation options like 'Home', 'Connect', and 'Discover'. The main content area displays 'Results for SFGiants' with a 'Top people' section featuring the San Francisco Giants (@SFGiants) and a 'Tweets' section with a tweet from 'America's Cup' (@americascup) and another from 'MLB Fan Cave' (@MLBFanCave).

## Ads

Search results for 'fidelity'. The 'Top people' section includes 'Fidelity Investments' (@Fidelity) with the ad text 'At Fidelity, we offer a full range of products to...'. The 'Tweets' section features a tweet from 'FXCM' (@FXCM) advertising 'All #forex trading strategies, including scalping FXCM's MT4 platform' with a link to 'bit.ly/zVx0Kq'. A blue arrow points from the 'FXCM' tweet towards the 'yieldthought' tweet in the adjacent image.

A close-up of a tweet from 'yieldthought' (@yieldthought) stating 'So I swapped my MacBook Pro for an iPad+Linode for a month...' with a link to 'tumblr.co/ZOd49yBPX9zo'. The tweet is circled in red, and a blue arrow points to it from the 'FXCM' tweet in the previous image.

## Recommendations

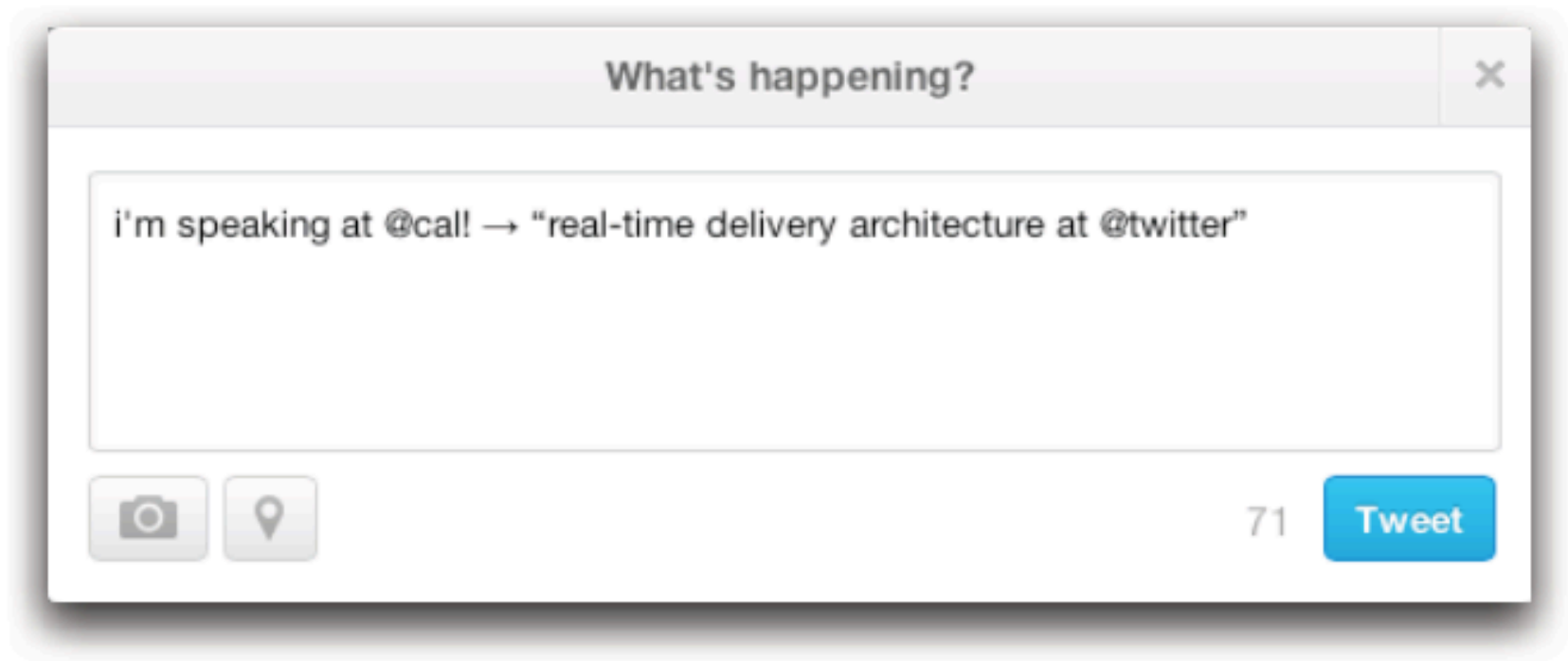
Recommendations section showing 'Who to follow' with profiles for 'Longreads', 'Adam Sharp', and 'George Takei'. Below this is a tweet from 'The Economist' (@TheEconomist) with the headline 'Indiana has seen a quiet whirlwind of education reform' and a link to 'econ.st/O679tS'. Another tweet from 'Curiosity Rover' (@MarsCuriosity) is visible at the bottom.

A promoted tweet from 'OppenheimerFunds' (@OppFunds) with the text 'Global is more than an asset class. It's a perspective. http://globalizeyourthinking.com'. The tweet includes a 'Follow' button and a 'Promoted' label.

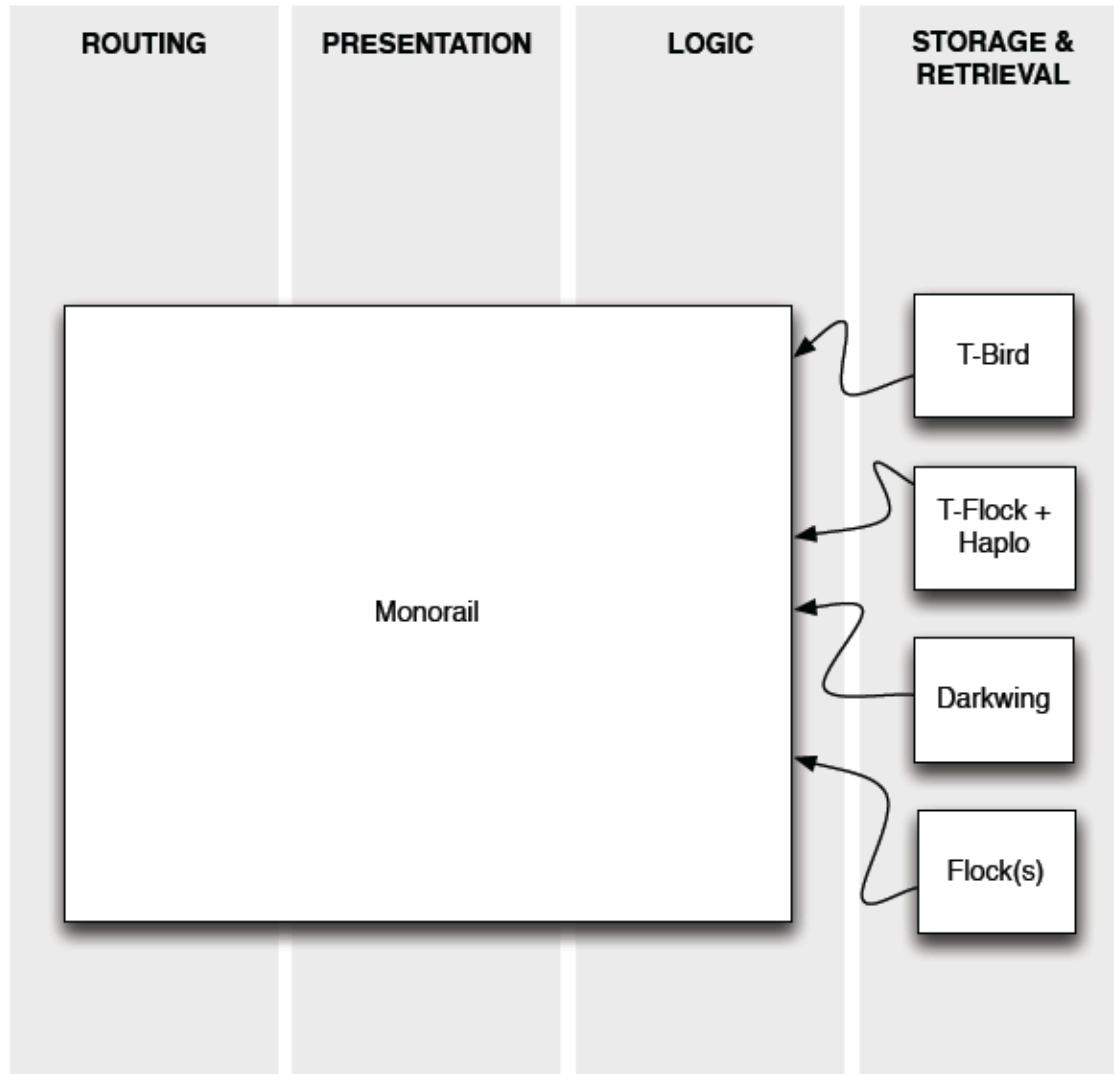
## Anti-Spam

A tweet from 'Jan Jones' (@JanJones1727) advertising 'get free sample viagra - EXTRA LOW PRICES - viagra > Save Your Money bit.ly/OUaEzM'. A context menu is open over the tweet, showing options: 'Tweet to @JanJones1727', 'Add or remove from lists...', 'Block @JanJones1727', and 'Report @JanJones1727 for spam'.





What happens when you “Tweet” this message?



2009

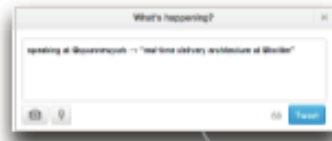
### Older Twitter Model:

Based on Ruby on Rails. Everything was being written on a big monolithic stack. Doesn't scale, 400 engineers work on same code base, no independence to team, too much time spent in co-ordination

# what are the goals?

- evolve from being solely a web stack
- isolate responsibilities and concerns
- site speed and reliability
- developer innovation speed

Everything has to happen in Real-Time.  
Event driven programming model to understand when a Tweet was posted, when someone replied etc.



This is needed to Push tweets in user's timeline. The timeline is replicated three times.

Use Write API to Write tweet in DB

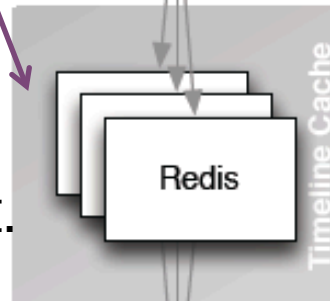
Write API

Fan out is delivering the Tweet to every single person who is following that person

Fanout

Not saved to disk..!  
Stored in RAM, Allows fast recovery: 45 ms  
Only active users in past 28 days (LRU) stored in RAM. Rest goes on Disk.

Redis Cluster: user-id = key, Tweet = value...!  
(Map part of) Map Reduce Programming



800 tweets per home timeline. Rest is stored on Data-centers

Timeline Service figures out where person's Timeline lies in Redis cluster (Reduce part of Map-Reduce)

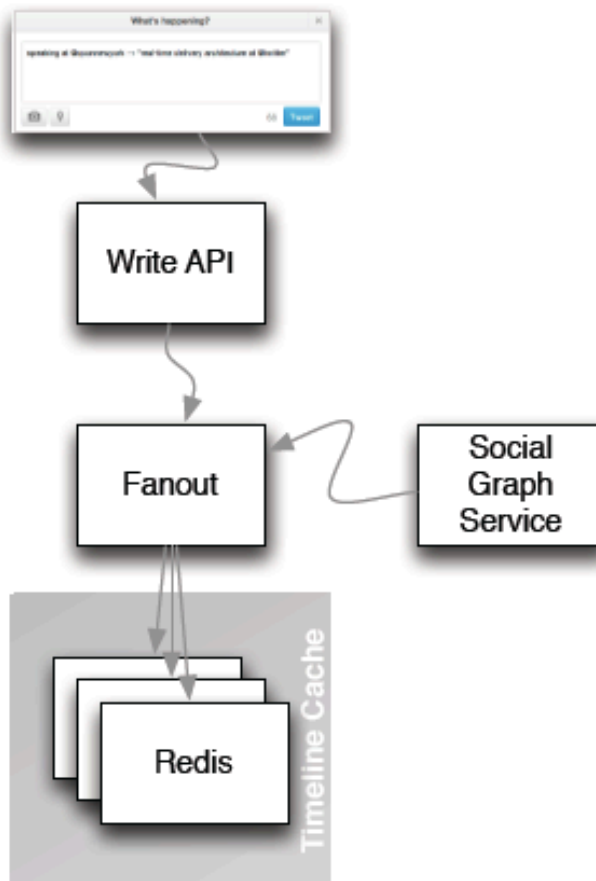
Gizmoduck

Timeline Service

TweetyPie

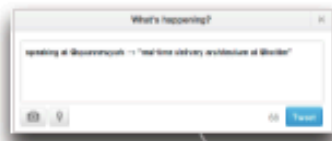
The goal is avoid hitting the disk as much as Possible..!





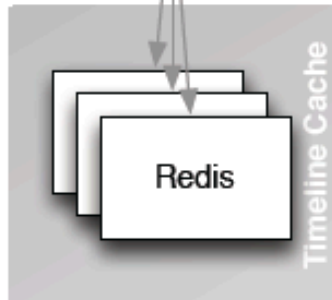
# insert

- keyed off “recipient”
- pipelined 4k “destinations” at a time
- replicated



Write API

Fanout



using redis

→ native list structure

→ RPU SHX to only add to cached timelines

Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	Tweet ID
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	Tweet ID
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	Tweet ID
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	

# + Anatomy of a User

```
id: 6253282,
id_str: "6253282",
name: "Twitter API",
screen_name: "twitterapi",
location: "San Francisco, CA",
url: "http://dev.twitter.com",
description: "The Real Twitter API. I tweet about API changes, service issues and happily answer questions about Twitter",
protected: false,
followers_count: 1217031,
friends_count: 31,
listed_count: 10784,
created_at: "Wed May 23 06:01:13 +0000 2007",
favourites_count: 25,
utc_offset: -28800,
time_zone: "Pacific Time (US & Canada)",
geo_enabled: true,
verified: true,
statuses_count: 3336,
lang: "en",
status: {
  created_at: "Thu Sep 06 17:55:54 +0000 2012".
contributors_enabled: true,
is_translator: false,
profile_background_color: "CODEED",
profile_background_image_url: "http://a0.twimg.com/images/themes/theme1/bg.png",
profile_background_image_url_https: "https://s10.twimg.com/images/themes/theme1/bg.png",
profile_background_tile: false,
profile_image_url: "http://a0.twimg.com/profile_images/2284174872/7df3h38zabcvjvlnyfe3_normal.png",
profile_image_url_https: "https://s10.twimg.com/profile_images/2284174872/7df3h38zabcvjvlnyfe3_normal.png",
profile_banner_url: "https://s10.twimg.com/profile_banners/6253282/1347053495",
profile_link_color: "0084B4",
profile_sidebar_border_color: "CODEED",
profile_sidebar_fill_color: "DDEEF6",
profile_text_color: "333333",
profile_use_background_image: true,
show_all_inline_media: false,
default_profile: true,
default_profile_image: false,
following: null,
follow_request_sent: null,
notifications: null
```



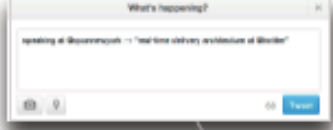
The screenshot shows the Twitter profile for 'twitterapi'. The profile picture is a blue gear with a white bird. The name is 'twitterapi' with a verified badge. The bio reads: 'The Real Twitter API. I tweet about API changes, service issues and happily answer questions about Twitter and our API. Don't get an answer? It's on my website.' The website link is 'http://dev.twitter.com'. The profile shows 988,982 followers and 33 people being followed. At the bottom, there are icons for 'GET /jobs' and several social media icons.

**twitterapi** Twitter API ✓  
Following

The Real Twitter API. I tweet about API changes, service issues and happily answer questions about Twitter and our API. Don't get an answer? It's on my website.  
<http://dev.twitter.com>

Followers **988,982** Following **33**

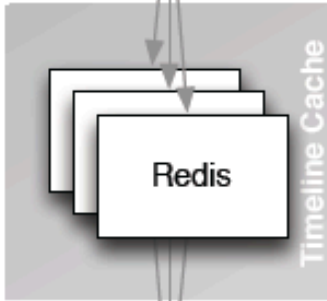
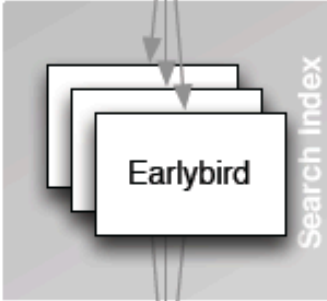
GET /jobs



Write API

Ingester

Fanout



Blender

Timeline Service



This architecture  
Allows to blend in  
copied/followed tweets  
into respective  
users timeline

200-700ms



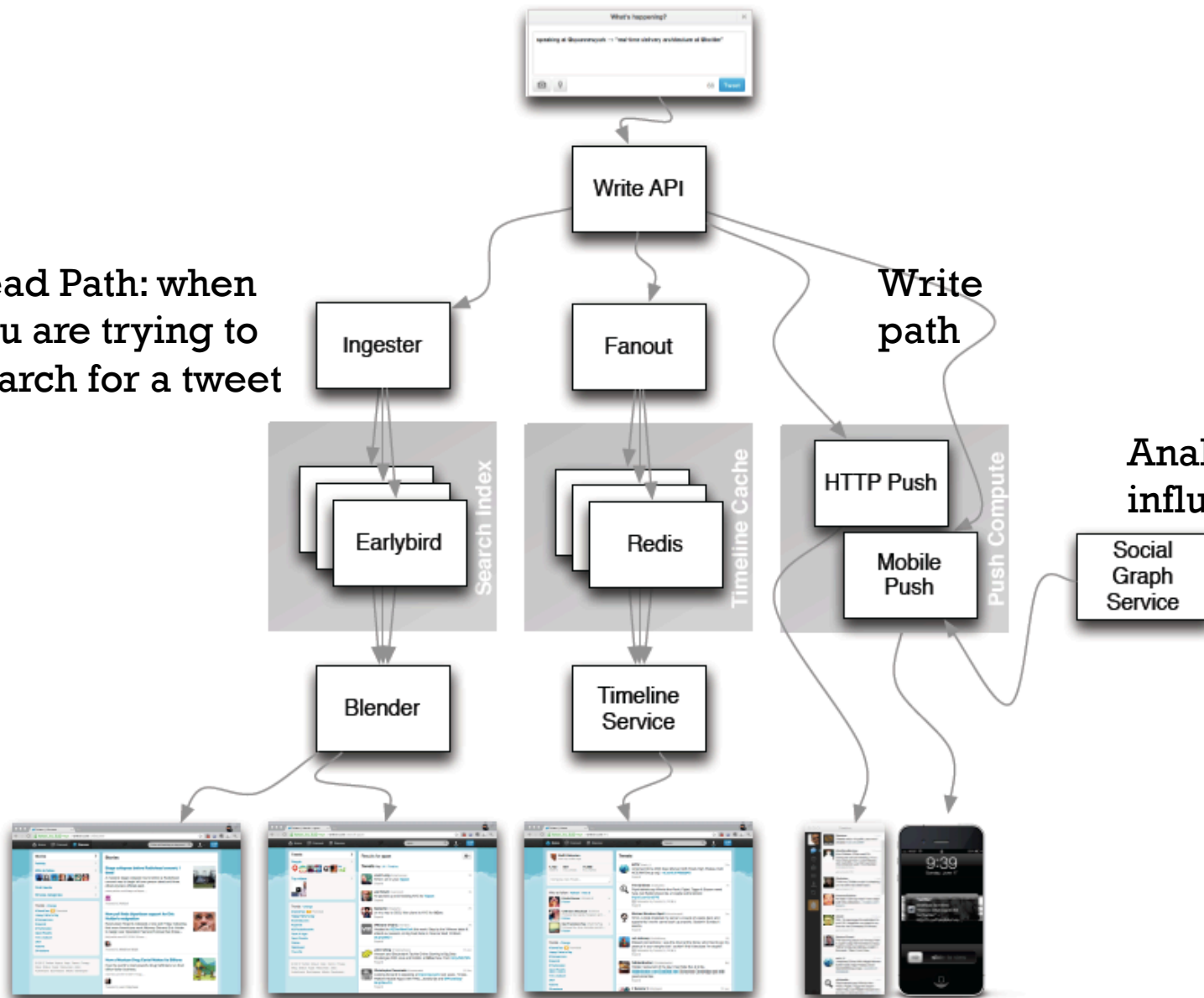




Read Path: when you are trying to search for a tweet

Write path

Analyze influence



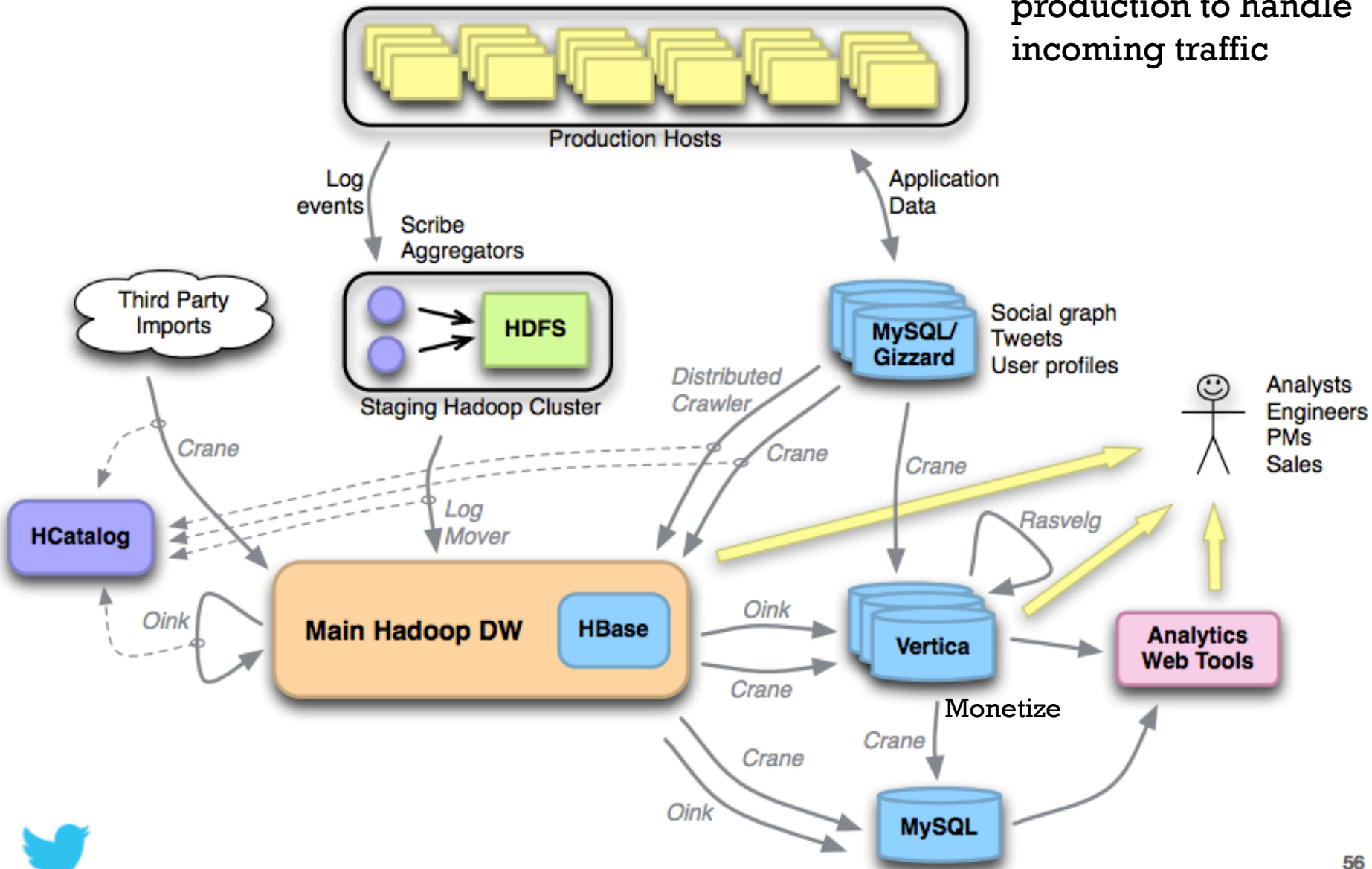
Search Operation

Regular tweet

Access via web/mobile

# Twitter Analytics data flow

Servers in production to handle incoming traffic





# Analyzing Machine Generated Data



- Searching, monitoring and analyzing machine generated big data via web interface
- Allows real-time response model when servers/clusters fail
- Allows trend detection/understanding unpredicted events
- Widely used in web-analytics