Introduction to Data Science

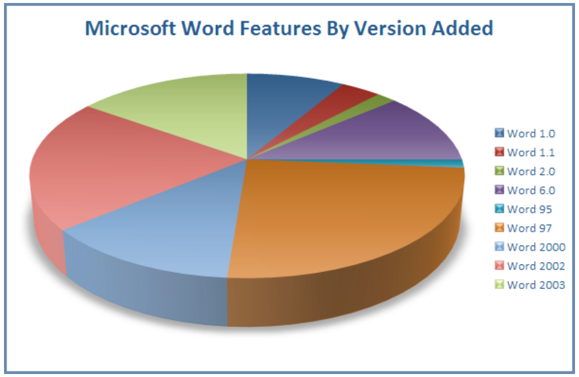
CS 5665
Utah State University
Department of Computer Science
Instructor: Prof. Kyumin Lee

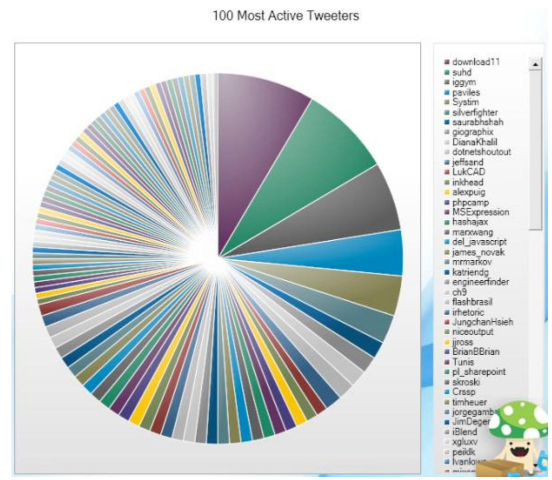
Data Visualization

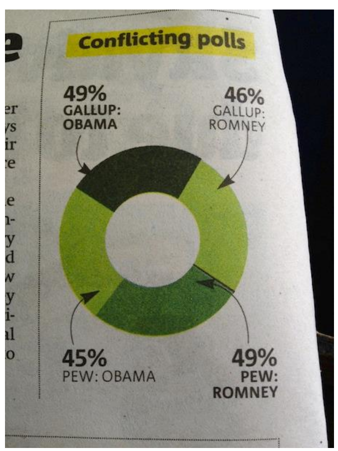
Critique

Design Critique Questions

- What is the purpose of the visualization?
- Does it serve its purpose well?
  - Does it convey the data honestly?
  - Does it show the appropriate amount of data?
- Does it address an important topic?
- Is it innovative?


Microsoft Word Features By Version Added
Word 1.0
Word 1.1
Word 2.0
Word 6.0
Word 95
Word 97
Word 2000
Word 2002
Word 2003


100 Most Active Tweeters
download11
suhd
iggym
paviles
Systim
silverfighter
saurabhshah
giographix
DianaKhalil
dotnetshoutout
jeffsand
LukCAD
inkhead
alexpuig
phpcamp
MSExpression
hashajax
merxwang
del_javascript
james_novak
mrmarkov
katriendg
engineerfinder
ch9
flashbrasil
irhetoric
JungchanHsieh
niceoutput
jjross
BrianBBrian
Tunis
pl_sharepoint
skroski
Crssp
timheuer
JimDeger
iBlend
xgluxv
peikk


Conflicting polls
49%
GALLUP:
OBAMA
46%
GALLUP:
ROMNEY
45%
PEW: OBAMA
49%
PEW:
ROMNEY

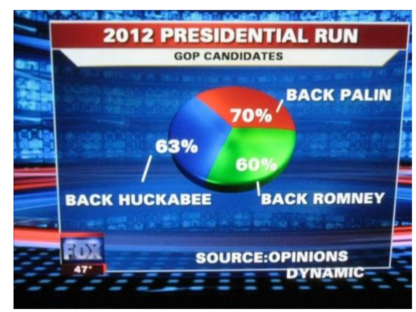
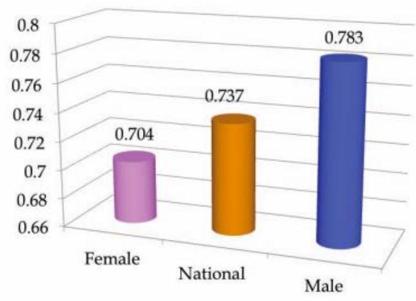
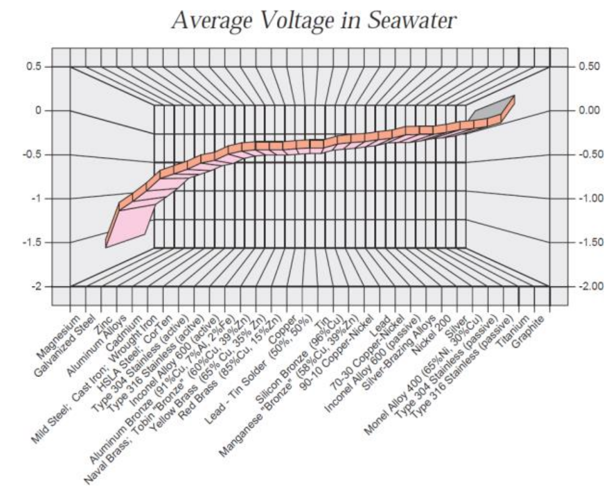
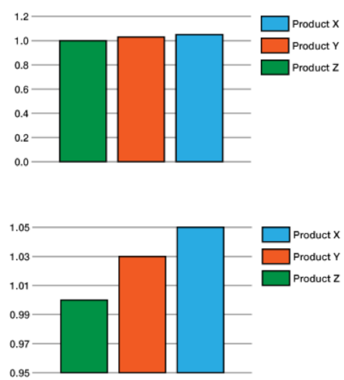

2012 PRESIDENTIAL RUN
GOP CANDIDATES
BACK PALIN
70%
63%
60%
BACK HUCKABEE
BACK ROMNEY
FOX
SOURCE:OPINIONS
DYNAMIC

Figure 11: GNH index by gender







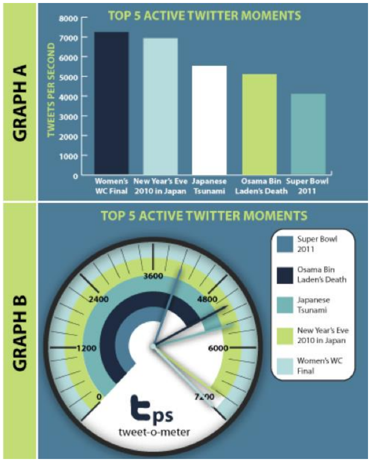
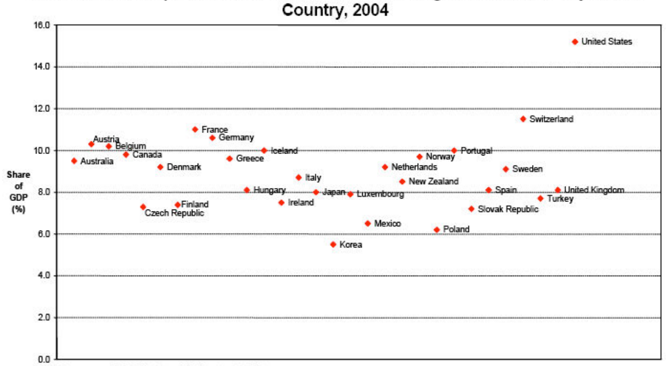
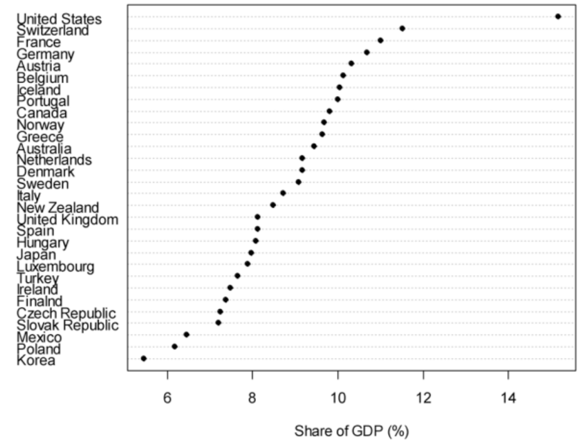


Chart 2 - Total Expenditures on Health as a Percentage Share of GDP, by OECD Country, 2004



Source: OECD Health Data 2007.
Note: For the United States the 2004 data reported here do not match the 2004 data point for the United States in Chart 1 since the OECD uses a slightly different definition of "total expenditures on health" than that used in the National Health Expenditure Accounts.

Expenditures on Health as Percentage of GDP for OECD Countries, 2004

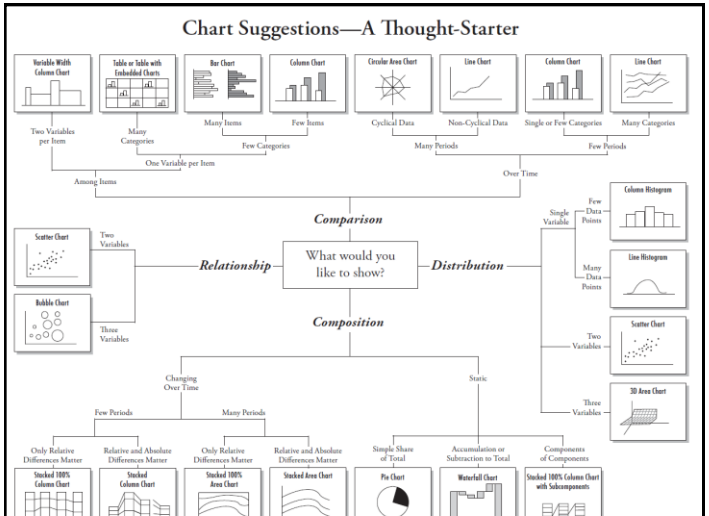


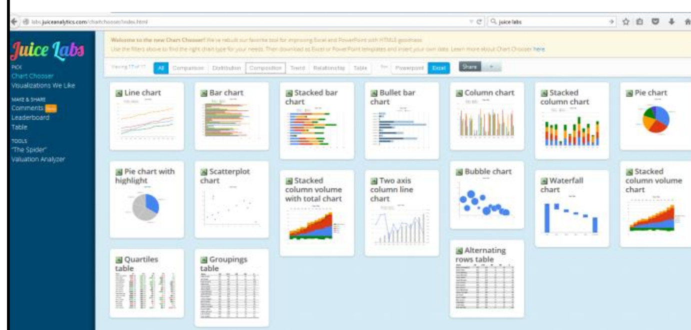
http://www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html

http://submarine-cable-map-2013.telegeography.com/

http://www.npr.org/sections/itsallpolitics/2012/11/01/163632378/a-campaign-map-morphed-by-money

Practical Tips





http://labs.juiceanalytics.com/chartchooser/index.html

http://labs.juiceanalytics.com/vizwelike/index.html

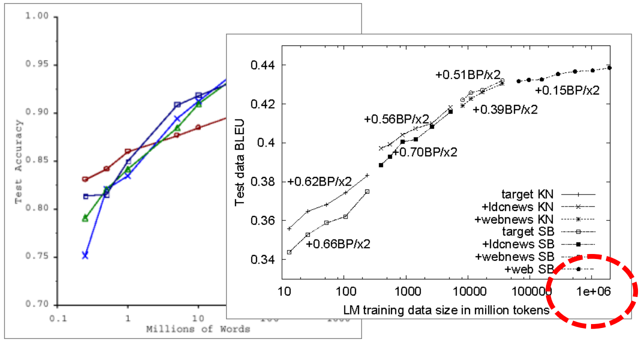
Cloud Computing + MapReduce

The Trend Today: Big Data

- Google processes 20 PB a day (2008)
- Wayback 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 LHC will generate 15 PB a year (??)

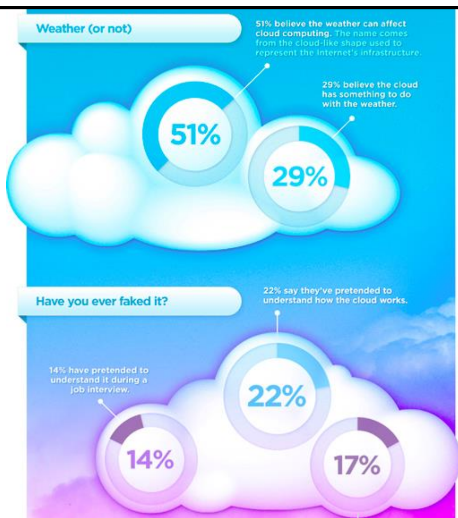
What can we do with more data?

No data like more data!



How do we get here if we're not Google?

How do we get there? Cloud computing!



Brief history of the "cloud"

- Before clouds…
  - Grids
  - Vector supercomputers
  - …
- Cloud computing means many different things:
  - **Large-data processing**
  - Rebranding of web 2.0
  - Utility computing
  - Everything as a service

Rebranding of web 2.0

- Rich, interactive web applications
  - Clouds refer to the servers that run them
  - AJAX as the de facto standard (for better or worse)
  - Examples: Facebook, YouTube, Gmail, …
- "The network is the computer": take two
  - User data is stored "in the clouds"
  - Rise of the netbook, smartphones, etc.
  - Browser *is* the OS

Source: Wikipedia (Electricity

Utility Computing

- What?
  - Computing resources as a metered service ("pay as you go")
  - Ability to dynamically provision virtual machines
- Why?
  - Cost: capital vs. operating expenses
  - Scalability: "infinite" capacity
  - Elasticity: scale up or down on demand
- Does it make sense?
  - Benefits to cloud users
  - Business case for cloud providers

Customer Success. Powered by the AWS Cloud.

Pinterest Watch the Video »	SAP Watch the Video »	Bristol-Myers Squibb Watch the Video »	Airbnb Read the Story »
GE Watch the Video »	Pfizer Read the Story »	Foursquare Read the Story »	Lionsgate Read the Story »
reddit.com Watch the Video »	CDC Biosense 2.0 Watch the Video »	Thomson Reuters Read the Story »	Mailbox Watch the Video »

Everything as a Service

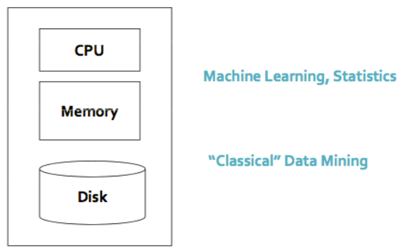
- Utility computing = Infrastructure as a Service (IaaS)
  - Why buy machines when you can rent cycles?
  - Examples: Amazon's EC2, Rackspace
- Platform as a Service (PaaS)
  - Give me nice API and take care of the maintenance, upgrades, …
  - Example: Google App Engine
- Software as a Service (SaaS)
  - Just run it for me!
  - Example: Gmail, Salesforce

Who cares?

- Ready-made large-data problems
  - Lots of user-generated content
  - Even more user behavior data
  - Examples: Facebook friend suggestions, Google ad placement
  - Business intelligence: gather everything in a data warehouse and run analytics to generate insight
- Utility computing
  - Provision Hadoop clusters on-demand in the cloud
  - Lower barrier to entry for tackling large-data problem
  - Commoditization and democratization of large-data capabilities

Data Science at Scale

Single-node architecture



Motivation (Google)

- 20+ billion web pages x 20KB = 400+ TB
- 1 computer reads 30-35 MB/sec from disk
  - ~4 months to read the web
- ~1,000 hard drives to store the web
- Even more to do something with the data

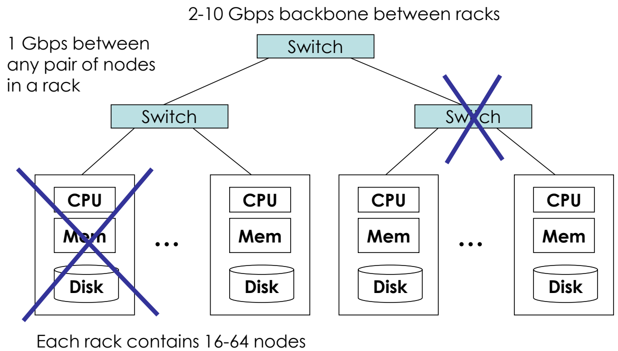
Commodity Clusters

- Web data sets can be very large
  - Tens to hundreds of terabytes
- Cannot analyze on a single server
- Standard architecture emerging:
  - Cluster of commodity Linux nodes
  - Gigabit ethernet interconnect
- How to organize computations on this architecture?
  - Issues such as hardware failure

Big computation – Big Machines

- Traditional big-iron box (circa 2003)
  - 8 2GHz Xeons
  - 64GB RAM
  - 8TB disk
  - $758,000
- Prototypical Google rack (circa 2003)
  - 176 2GHz Xeons
  - 176GB RAM
  - ~7TB disk
  - $278,000
- In Jan 2012, Google had ~1,800,000 machines

Cluster Architecture



https://www.youtube.com/watch?v=XZmGGAbHqa0

Large-scale computing

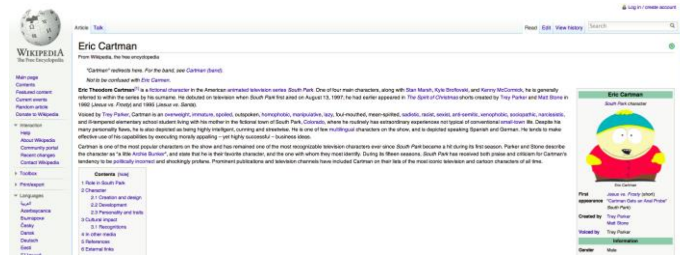
- Large-scale computing for data mining problems on commodity hardware
  - PCs connected in a network
  - Need to process huge datasets on large clusters of computers
- Challenges:
  - How do you distribute computation?
  - Distributed programming is hard
  - Machines fail

MapReduce

Class Outline

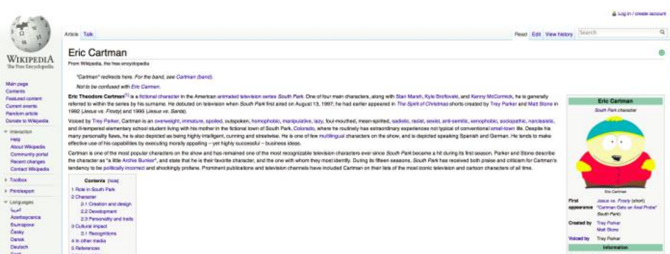
- What is MapReduce and why do we need it?
- Understand how MapReduce works
- How do we use it?
- Examples of solving large data problems using MapReduce

Word Count



- Count the number of times each distinct word appears in this document

Word Count – Case 1: Small Single File



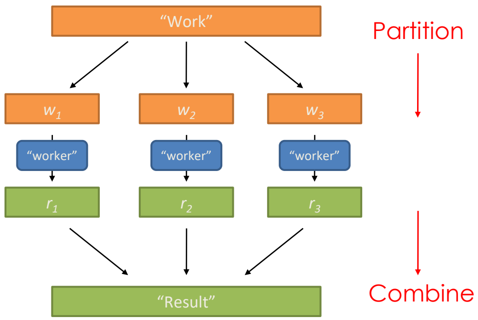
- Read file
- Use a dictionary to track the number of times a word appears

Word Count – Case 2: Large Document Corpus

WIKIPEDIA
The Free Encyclopedia

- Distributed / parallel computing

The Hope: Divide and Conquer



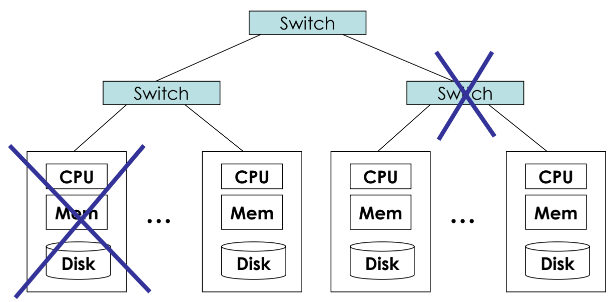
Distributed Computing Challenges - Scheduling

- How do we assign work units to workers?
- What if we have more work units than workers?

Distributed Computing Challenges - Synchronization

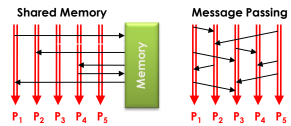
- What if workers need to share partial results?
- How do we aggregate partial results?
- How do we know all the workers have finished?

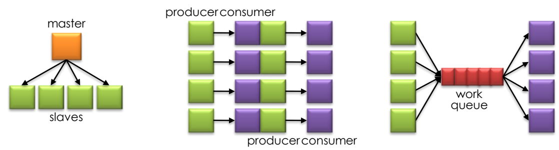
Distributed Computing Challenges - Handling Network Failure



Current Tools

- Programming models
  - Shared memory (pthreads)
  - Message passing (MPI)
- Design Patterns
  - Master-slaves
  - Producer-consumer flows
  - Shared work queues





Where the rubber meets the road

- Concurrency is difficult to reason about
- Concurrency is even more difficult to reason about
  - At the scale of datacenters (even across datacenters)
  - In the presence of failures
  - In terms of multiple interacting services
- Not to mention debugging…

MapReduce to the rescue

Introducing MapReduce

- A framework to support distributed computing on large datasets. (Wikipedia)
- Introduced by Google in 2004.
- Very popular with almost every company involved in large scale data processing.
  - Google, Twitter, Amazon, Facebook etc.

MapReduce Implementations

- Google has a proprietary implementation in C++
  - Bindings in Java, Python
- Hadoop is an open-source implementation in Java
  - Development led by Yahoo, used in production
  - Now an Apache project
  - Rapidly expanding software ecosystem, but still lots of room for improvement
- Lots of custom research implementations
  - For GPUs, cell processors, etc.

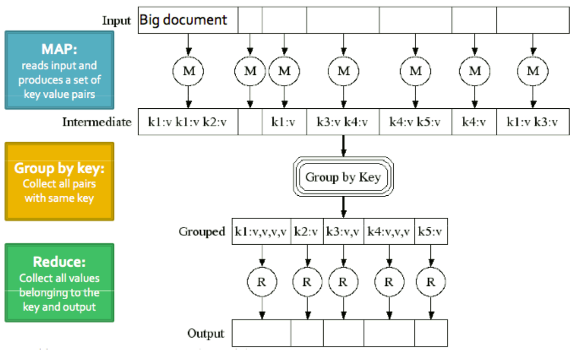
What MapReduce Does?

- Handles scheduling
  - Assigns workers to map and reduce tasks
- Handles “data distribution”
  - Moves processes to data
- Handles synchronization
  - Gathers, sorts, and shuffles intermediate data
- Handles errors and faults
  - Detects worker failures and automatically restarts
- Everything happens on top of a distributed FS
  - Ex: Google’s GFS and Hadoop’s HDFS

What do you do?

- Define two functions:
  - map (k, v) → <k’, v’>*
  - reduce (k’, <v’>) → <k’, v’’>*
- All v’ with the same k’ are reduced together and processed in v’ order

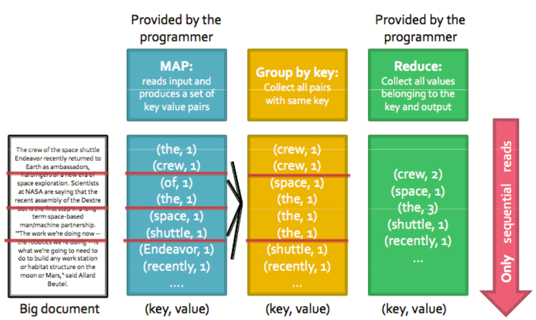
MapReduce: A diagram



MapReduce: Word counting

- Program specifies two primary methods
  - Map(k,v) → <k', v'>*
  - Reduce(k',<v'>*) → <k', v''>*
- Ex: Two documents - (d1, "the crew") and (d2, "of the")
  - Map(d1, "the crew") => [(the, 1), (crew, 1)]
  - Map(d2, "of the") => [(of, 1), (the, 1)]
  - MapReduce runs its grouper module and calls reduce for every key
  - Reduce (the, [1,1]) => (the, 2)
  - Reduce (crew, [1]) => (crew, 1)
  - Reduce (of, [1]) => (of, 1)

MapReduce: Word counting



Word Count in MapReduce

```python
def mapper(self, key, value):
    for word in value.split(): yield word, 1

def reducer(self, key, values): yield key, sum(values)
```

Word Count in MapReduce (Java)

- public void **map**(Object key, Text value, Context context) {
- StringTokenizer itr = new StringTokenizer(value.toString());
- while (itr.hasMoreTokens()) {
- word.set(itr.nextToken());
- context.write(word, *one);*
- }
- }

Word Count in MapReduce (Java)

- public void **reduce**(Text key, Iterable<IntWritable> values, Context context) {
- int sum = 0;
- for (IntWritable val : values) {
- sum += val.get();
- }
- result.set(sum);
- context.write(key, result);
- }

Data flow

- Input, final output are stored on a distributed file system
  - Scheduler tries to schedule map tasks “close” to physical storage location of input data
- Intermediate results are stored on local filesystem of map and reduce workers
- Output is often the input to another MapReduce task