CS-495/595
Hadoop (part 1)
Lecture #3

Dr. Chuck Cartledge

4 Feb. 2015

Table of contents I

- Updated syllabus to reflect late submission penalties
- Grading submissions
- Need to schedule tests

With a little license.

A simply stated problem: Count the number of unique words in Shakespeare's Macbeth.

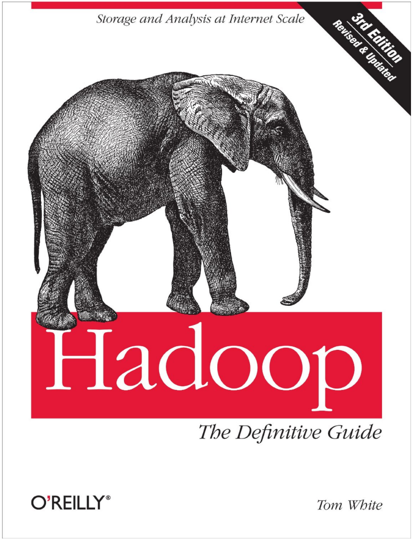
- A few Java classes
- A Hadoop environment
- Process strings from a file
- Summarize the results

Grad students have a little more to do.

Dennis Ritchie
1941-2011

Hadoop, The Definitive Guide

- Version 3 is specified in the syllabus [2]
- Version 4 came out in November 2015
- We'll use Version 3 as much as possible



Data!

"In pioneer days they used oxen for heavy pulling, and when one ox couldn't budge a log, they didn't try to grow a larger ox. We shouldn't be trying for bigger computers, but for more systems of computers." — Grace Hopper

- Lots of data from all sorts of places. Some that we've addressed before.
- Microsoft Research's My Life Bits project (`http://research.microsoft.com/en-us/projects/mylifebits/default.aspx`)
- Look at processing from a systemic point of view:
  1. Paralizable
  2. Data locality
  3. Coordination
  4. Output

These issues and problems appear time and time again.

Compared to other systems

MapReduce is batch processing (vice interactive or real-time)

- Dividing line between batch and real-time is slippery
- Other parallel programming technologies provide greater control, but require greater management (MPI)
- Some parallel computing is relatively trivial (SETI@home)

Table 1-1. RDBMS compared to MapReduce

	Traditional RDBMS	MapReduce
Data size	Gigabytes	Petabytes
Access	Interactive and batch	Batch
Updates	Read and write many times	Write once, read many times
Structure	Static schema	Dynamic schema
Integrity	High	Low
Scaling	Nonlinear	Linear

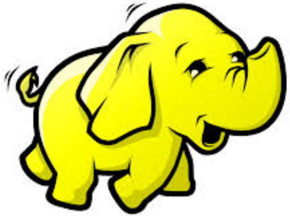
Image from [2]. .

Minimizing execution by balancing computation and data access times

Summary

"This, in a nutshell, is what Hadoop provides: a reliable shared storage and analysis system. The storage is provided by HDFS and analysis by MapReduce. There are other parts to Hadoop, but these capabilities are its kernel."

- Paralizable – number of mappers determined by the size of the input (growth in linear)
- Data locality – HDFS distributes input data to processing nodes
- Coordination – starts, monitors, stops, restarts parallel tasks
- Output – intermediate output is local, global is in HDFS



A CLI execution

Find maximum temperature

- No program to compile
- Single thread of control
- Single output

Execution time: 42 minutes

```
#!/usr/bin/env bash
for year in all/*
do
  echo -ne `basename $year .gz`"\t"
  gunzip -c $year | \
    awk '{ temp = substr($0, 88, 5) + 0;
           q = substr($0, 93, 1);
           if (temp !=9999 && q ~ /[01459]/ && temp > max) max = temp }
         END { print max }'
done
```

A MapReduce execution

- A program to compile
- Multiple threads of control
- Multiple intermediate files
- Single output
- Input and output is via HDFS

Execution time: 6 minutes

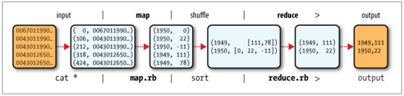


Figure 2-1. MapReduce logical data flow

Hadoop supports many languages (this is a Ruby example).

Basics of what Hadoop provides

- Standard input (from the HDFS)
- Repeated and/or multiple executions of the mapper
- Consolidation, shuffling, and sorting of mapper outputs
- Multiple executions of the reducer
- Sorting of reducer outputs
- Safely writing output to HDFS

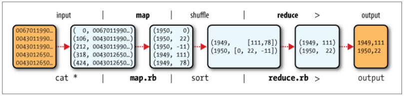


Figure 2-1. MapReduce logical data flow

Basic pipe and filter architecture.

Hadoop supports any language that supports STDIN and STDOUT.

How many mappers and reducers?

- Number of mappers = input file size / HDFS block size
- Number of reducers defaults to 1
- Optimal number of reducers is < number of nodes in system
- Key value sorting is local
- All key values are available to all reducers
- reducer outputs are separate in HDFS

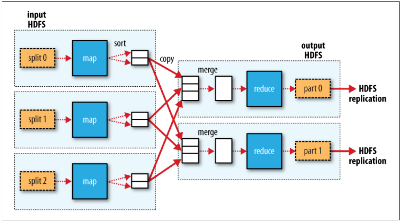
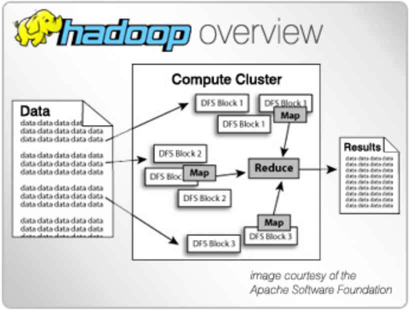


Figure 2-4. MapReduce data flow with multiple reduce tasks

The specifics will differ. (We've seen these before.)

1. Write a `Mapper`
2. Write a `Reducer`
3. Write a `main`
4. For Java: compile all the classes into a jar (including the Hadoop classes)
5. For Java: run the jar: `hadoop jar jar_file mainClass arguments`
6. Use `hadoop fs` to retrieve the output



Nothing magic about Java.

Hadoop supports any mix of languages that support STDIN and STDOUT.

- C, C++, Ruby, Python, ...
- Supports STDIN and STDOUT

Example 2-10. Map function for maximum temperature in Python

```
#!/usr/bin/env python

import re
import sys

for line in sys.stdin:
  val = line.strip()
  (year, temp, q) = (val[15:19], val[87:92], val[92:93])
  if (temp != "+9999" and re.match("[01459]", q)):
    print "%s\t%s" % (year, temp)
```

Example 2-9. Reduce function for maximum temperature in Ruby

```
#!/usr/bin/env ruby

last_key, max_val = nil, 0
STDIN.each_line do |line|
  key, val = line.split("\t")
  if last_key && last_key != key
    puts "#{last_key}\t#{max_val}"
    last_key, max_val = key, val.to_i
  else
    last_key, max_val = key, [max_val, val.to_i].max
  end
end
puts "#{last_key}\t#{max_val}" if last_key
```

How to run Hadoop with Ruby.

```
% hadoop jar $HADOOP_INSTALL/contrib/streaming/hadoop-*-streaming.jar \
  -input input/ncdc/sample.txt \
  -output output \
  -mapper ch02/src/main/ruby/max_temperature_map.rb \
  -reducer ch02/src/main/ruby/max_temperature_reduce.rb
```

Assumes that ruby is available.

Typical software development process.

- Requirements
- Development
- Unit testing with local test data
- Cluster testing with HDFS data
- Refine as necessary

Debugging a distributed application can be challenging. Unit tests should be through.

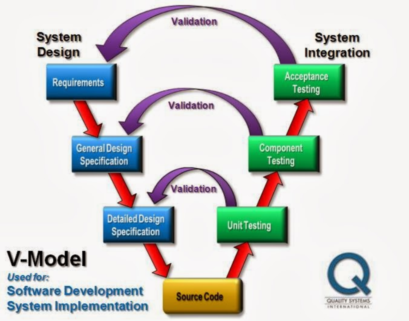
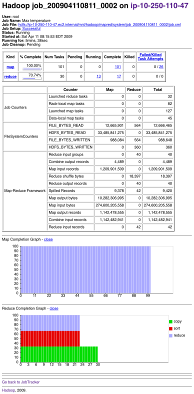


Image from: http://allabttesting.blogspot.com/2013_10_01_archive.html

Local and cluster testing controlled by Hadoop arguments, or by configuration files

Web UI might be available on some installations.



Hadoop job_200904110811_0002 on ip-10-250-110-47

User: root
Job Name: Max temperature
Job File: hdfs://ip-10-250-110-47.ec2.internal/mnt/hadoop/mapred/system/job_200904110811_0002/job.xml
Job Setup: Successful
Status: Running
Started at: Sat Apr 11 08:15:53 EDT 2009
Running for: 5mins, 38sec
Job Cleanup: Pending

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	101	0	0	101	0	0 / 26
reduce	70.74%	30	0	13	17	0	0 / 0

	Counter	Map	Reduce	Total
Job Counters	Launched reduce tasks	0	0	32
	Rack-local map tasks	0	0	82
	Launched map tasks	0	0	127
	Data-local map tasks	0	0	45
FileSystemCounters	FILE_BYTES_READ	12,665,901	564	12,666,465
	HDFS_BYTES_READ	33,485,841,275	0	33,485,841,275
	FILE_BYTES_WRITTEN	988,084	564	988,648
	HDFS_BYTES_WRITTEN	0	360	360
Map-Reduce Framework	Reduce input groups	0	40	40
	Combine output records	4,489	0	4,489
	Map input records	1,209,901,509	0	1,209,901,509
	Reduce shuffle bytes	0	18,397	18,397
	Reduce output records	0	40	40
	Spilled Records	9,378	42	9,420
	Map output bytes	10,282,306,995	0	10,282,306,995
	Map input bytes	274,600,205,558	0	274,600,205,558
	Map output records	1,142,478,555	0	1,142,478,555
	Combine input records	1,142,482,941	0	1,142,482,941
	Reduce input records	0	42	42

Book says Web UI is available at http://jobtracker-host:50030/

How many reducers should I have?

- I == input to the reducers
- O == output from the reducers
- p == number of reducers available
- q == number of inputs
- r == number of replications

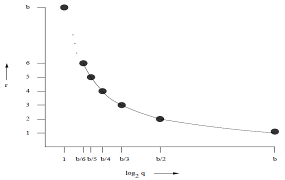


Figure 1: Known algorithms matching the lower bound on replication rate

$$r = \frac{\sum_{i=1}^{p} q_i}{|I|} \geq \frac{q|O|}{g(q)|I|}$$

Image from [1]. .

Work to compute r.

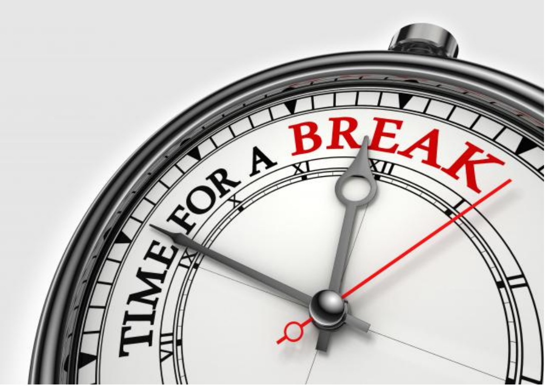
An example

Problem	$\lvert I\rvert$	$\lvert O\rvert$	$g(q)$	Lower bound on r
Hamming-Distance-1, b-bit strings	2^b	$\frac{b2^b}{2}$	$\frac{q\log_2 q}{2}$ (Section 3.1)	$\frac{b}{\log_2 q}$ (Section 3.2)
Triangle-Finding, n nodes	$\frac{n^2}{2}$	$\frac{n^3}{6}$	$\frac{\sqrt{2}}{3}q^{\frac{3}{2}}$ (Section 4.1)	$\frac{n}{\sqrt{2q}}$ (Section 4.1)
Sample Graphs (size s nodes) in Alon Class in graph of m edges, n nodes	$\binom{n}{2}$ or m	n^s	$q^{s/2}$ (Section 5.2)	$\left(\frac{n}{\sqrt{q}}\right)^{s-2}$ or $\left(\sqrt{\frac{m}{q}}\right)^{s-2}$ (Sections 5.2 and 5.3)
2-Paths in n-node graph	$\binom{n}{2}$	$\frac{n^3}{2}$	$\binom{q}{2}$ (Section 5.4.1)	$\frac{2n}{q}$ (Section 5.4.1)
Multiway Join: N bin. rels, m vars., Dom. n, parameter ρ from [6]	$N\binom{n}{2}$	$\binom{n}{m}$	q^ρ ([6])	$\frac{n^{m-2}}{q^{\rho-1}}$ (Section 5.5.1)
$n\times n$ Matrix Multiplication	$2n^2$	n^2	$\frac{q^2}{4n^2}$ (Section 6.1)	$\frac{2n^2}{q}$ (Section 6.1)

Table 1: Lower bound on replication rate r for various problems in terms of number of inputs $\lvert I\rvert$, number of outputs $\lvert O\rvert$, and maximum number of inputs per reducer q.

Understanding your problem will drive your replications.

Break time.



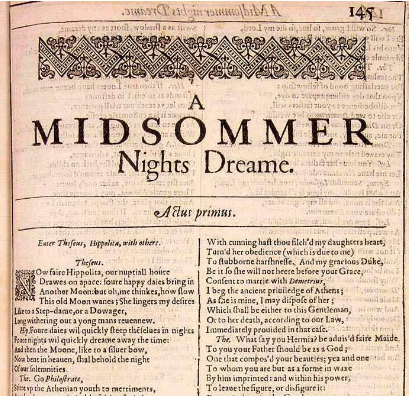
Take about 10 minutes.

Looking at where words are used.

A simply stated problem: where are certain words used?

- Undergrad students – which lines have the word "loue"
- Grad student – which lines of have the word "loue", which have the word "course", and which have both

Interested in the line numbers and the line itself.
An example: 1408: And wonne thy loue, doing thee iniuries:



A MIDSOMMER Nights Dreame.

Actus primus.

Enter Theseus, Hippolita, with others.

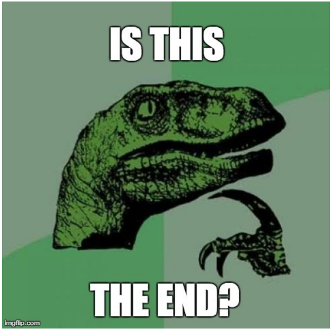
Theseus.

Now faire Hippolita, our nuptiall houre
Drawes on apace: foure happy daies bring in
Another Moon: but oh, me thinkes, how slow
This old Moon wanes; She lingers my desires
Like to a Step-dame, or a Dowager,
Long withering out a yong mans reuennew.
Hip. Foure daies wil quickly steep thēselues in nights
Foure nights wil quickly dreame away the time:
And then the Moone, like to a siluer bow,
Now bent in heauen, shal behold the night
Of our solemnities.
The. Go *Philostrate*,
Stirre vp the Athenian youth to merriments,

With cunning hast thou filch'd my daughters heart;
Turn'd her obedience (which is due to me)
To stubborne harshnesse. And my gracious Duke,
Be it so she will not heere before your Grace,
Consent to marrie with *Demetrius*,
I beg the ancient priuiledge of Athens;
As she is mine, I may dispose of her;
Which shall be either to this Gentleman,
Or to her death, according to our Law,
Immediately prouided in that case.
The. What say you Hermia? be aduis'd faire Maide,
To you your Father should be as a God;
One that compos'd your beauties; yea and one
To whom you are but as a forme in waxe
By him imprinted: and within his power,
To leaue the figure, or disfigure it:

What have we covered?

- Review the idea of lots of data
- Summarized what Hadoop brings to the table
- Overview of the Hadoop architecture
- Hadoop is almost language agnostic
- Mappers controlled by Hadoop
- Reducers can be controlled
- Assignment #2



Next lecture: Hadoop book, Chapters 3 and 4

References I

[1] Anish Das Sarma, Foto N Afrati, Semih Salihoglu, and Jeffrey D Ullman, Upper and lower bounds on the cost of a map-reduce computation, Proceedings of the VLDB Endowment, vol. 6, VLDB Endowment, 2013, pp. 277–288.

[2] Tom White, Hadoop: The definitive guide, 3rd edition, O'Reilly Media, Inc., 2012.