Extreme Computing

Admin and Overview

Course Staff

$\frac{1}{3}$xKenneth Heafield

$\frac{2}{3}$xVolker Seeker

Currently 12 TAs/demonstrators/markers

Website

`http://www.inf.ed.ac.uk/teaching/courses/exc`

Piazza

`https://piazza.com/class/j7m5dr4ns4dta`

(Linked from website)

Mailing List

`exc-students at inf.ed.ac.uk` is populated when you enroll.

Website

`http://www.inf.ed.ac.uk/teaching/courses/exc`

Piazza

`https://piazza.com/class/j7m5dr4ns4dta`

(Linked from website)

Mailing List

`exc-students at inf.ed.ac.uk` is populated when you enroll.

$\Longrightarrow$ Check website for announcements, especially first two weeks.

Assessment

25% Assignment 1
25% Assignment 2
50% Exam in May ☹ (December ☺ for visitors)

Don't start the assignments yet; they are being updated.

Assessment

25% Assignment 1
25% Assignment 2
50% Exam in May ☹ (December ☺ for visitors)

Don't start the assignments yet; they are being updated.

Solve the assignments on your own.
Don't share code.
Exam is closed book.

Assignment Deadlines

We'll provide you with a cluster to do assignments on.
The cluster will be offline on Sunday 22 October 2017.
$\rightarrow$ Assignment 1 will probably be due before then.

Lectures Online, subject to revision.

Labs Practice on a cluster. Not marked, but in exam.

Papers Linked from the website.

Books Don't buy them. They're in the library:
Data-Intensive Text Processing with MapReduce
Hadoop: The Definitive Guide.

The exam is based on the lectures and labs.

Labs

Run 2–27 October (four weeks) at these times:
Monday 9am
Monday 10am
Tuesday 2pm
Wednesday 10am
Wednesday 2pm
Thursday 9am
Thursday 11am
Friday 11am
Friday 2pm

Lab groups will be chosen online:
https://student.inf.ed.ac.uk.

Unix Command Line

We assume you know the Unix command line (typically `bash`).

```
tar cJ . | ssh server "cd $PWD && tar xJ"
diff <(zcat a.gz) <(zcat b.gz)
```

Unix Command Line

We assume you know the Unix command line (typically `bash`).

```
tar cJ . | ssh server "cd $PWD && tar xJ"
diff <(zcat a.gz) <(zcat b.gz)
```

If you didn't understand that, work through these:
http://www.ed.ac.uk/information-services/help-consultancy/is-skills/catalogue/program-op-sys-catalogue/unix1

https://www.lynda.com/Linux-tutorials/Linux-Bash-Shell-Scripts/504429-2.html
(The university has a subscription to lynda.com)

Programming Languages

The only language we require is command line.

Examples are mostly Python and Java, with occasional C++.

Programming Languages

The only language we require is command line.

Examples are mostly Python and Java, with occasional C++.

Average submission length:

	Lines	Words	Characters
Python	45.54	140.60	1412.81
Java	57.53	153.99	1738.76

Hint: `bash` is a programming language.

Data Structures

Know and apply foundational data structures: hash tables, arrays, queues, ...

These are taught in our second year undergraduate course, Informatics 2B.

Inefficient data structure choices will lose marks.

Core Course Content

- Working with big data
- Cluster computing with 10,000 machines
- How to pass a Google interview[1]
- How clouds like Amazon Web Services work

[1] Job at Google not guaranteed.

Core Course Content

- Working with big data
- Cluster computing with 10,000 machines
- How to pass a Google interview[1]
- How clouds like Amazon Web Services work

Not Part of the Course

- How to program (expected)
- Unix command line (learn it yourself)
- Mobile phones or Internet of things
- GPUs and FPGAs

[1] Job at Google not guaranteed.

Topics

Big Data

Cloud Computing Infrastructure

MapReduce and Hadoop

Beyond MapReduce

Fault Tolerance and Replication

NoSQL

BASE vs ACID

BitTorrent

Data warehousing

Data streams

Virtualisation

What is big data?

"You can turn small data into big data by wrapping it XML."

"If things are breaking, you have big data."

What is big data?

"You can turn small data into big data by wrapping it XML."

"If things are breaking, you have big data."

Big data is relative: not the same for Google and Informatics.

What is big data?

"You can turn small data into big data by wrapping it XML."

"If things are breaking, you have big data."

Big data is relative: not the same for Google and Informatics.
Sometimes Google's big data is our small data! [Brants et al, 2007]

The Internet Archive

560,000,000,000	Unique URLs of Web Crawl
4,000,000	eBooks
3,000,000	Hours of Television
2,400,000	Audio Recordings
2,300,000	Book Archive
2,000,000	Moving Images
25,000	Software Titles

30 Petabytes total

17 Petabytes of websites (gzipped)

2-3 Petabytes/year growth

900 TB in one machine

90 hard drives, each 10 TB, in one server

General Big Data

Government	Demographics, communication
Large Hadron Collider	15 PB/year
Fraud detection	Did your debit card work?
Social media	Who to follow?
Search	Can I borrow a copy of the web?
Online advertising	Placement, tracking, pricing

Common Source: Lots of Observations

- Every web page
- Mobile phone location reports
- Twitter posts
- Every Google search

Modeling Challenges of Big Data

Hard to understand and visualize

Tools often fail: need new algorithms

Models may not scale
Models that do scale may not show gains anymore

Performance

How do we access big data efficiently?
What patterns do we use for computation?

Disk Performance

Read speed on various devices:

	Random bytes/s	Sequential bytes/s
NVMe SSD	24,732	2,774,080,000
Old SATA SSD	7,848	256,781,000
5 TB Hard drive	82	171,302,000

Disk Performance

Read speed on various devices:

	Random bytes/s	Sequential bytes/s
NVMe SSD	24,732	2,774,080,000
Old SATA SSD	7,848	256,781,000
5 TB Hard drive	82	171,302,000

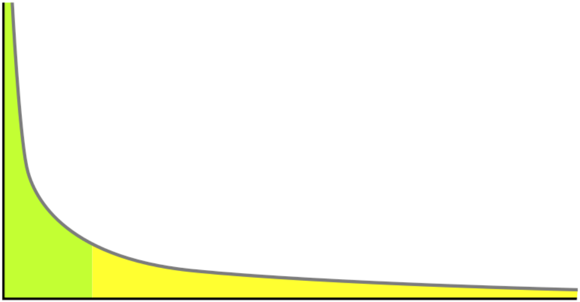
Sequential is 100,000–2 million times faster!

Sequential access impacts algorithm choice:

	Complexity	Access
Hash table	$O(n)$	Random
Merge sort	$O(n \log n)$	Sequential batches

Constant factors matter: merge sort is faster on disk.

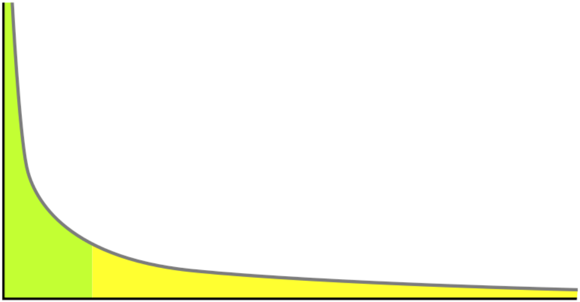
Power Law



Big data often follows a power law.

Modelling the head (e.g. common words) is easier, but unrepresentative.
Handling the tail is harder (e.g. selling all books, not just top 100).

Power Law



Big data often follows a power law.

Modelling the head (e.g. common words) is easier, but unrepresentative.
Handling the tail is harder (e.g. selling all books, not just top 100).

The machine responsible for “the” will take longer.

Challenge: Load Balancing

Distributed computing is a natural way to tackle big data.
But we need to balance work across machines:

- Head of power law goes to one or two nodes $\implies$ slow
- Tail balanced over nodes $\implies$ fast

Latency

How quickly does data move around the network?

- High-frequency trading: put machines next to the exchange
- Amazon (2007): sales decrease 1% for every 100ms increase in load time
- Google (2006): increasing page load time by 0.5 second produces a 20% drop in traffic
- Google rankings include page load time

Data centres and clusters

Supercomputers

A pile of Linux boxes in the same room, with a fast network.

Top 3 (according to top500.org):

1. Sunway TaihuLight 93,014 TFLOP/s, 10,649,600 cores
2. Tiahne-2: 33 TFLOP/s, 3,120,000 cores
3. Piz Daint: 19 TFLOP/s, 361,760 cores

Piz Daint

Economics of Servers: Own or Rent?

Many machines operate at 30% capacity.

Own

- Security
- Full control, customized hardware
- Tune for latency- or time-critical tasks
- Cheaper if machines will be used all the time

Rent

- Pay for servers, storage, and bandwidth by usage/hour
- Scale up to many servers when needed
- Compute is another commodity like electricity

Provisioning

Web traffic changes: time of day, shopping seasons, news, link from major site

High traffic $\rightarrow$ more machines
Low traffic $\rightarrow$ save cost

Target (US Retailer)

Website `target.com` is hosted on Amazon Web Services
Busiest shopping day in 2009: 28 November
Day `target.com` went offline: 28 November

Data lock-in and third-party control

Some provider hosts our data:

- But we can only access it using proprietary (non-standard) APIs
- Lock-in makes customers vulnerable to price increases and dependent upon the provider

Providers may control our data in unexpected ways:

- July 2009: Amazon remotely remove books from Kindles
- Twitter prevents exporting tweets more than 3200 posts back
- Facebook locks user-data in
- August 2010: Google drops Google Wave

Privacy and Security

Laundry list of breaches:

- Equifax
- NHS
- Ashley Madison hack
- US government HR database leaks, including security clearance
- Carphone Warehouse, Target, Health insurers

- What if your cloud provider is hacked?
- Who has access? The government? Which governments?

Need for privacy guarantees and measures.

Summary: Big Data

- Scalable algorithms
- Tools for cluster computing
- Cloud providers and how they work