Getting stuff done with Big Data

Lecture One: Big Data, Economics and Obstacle

Miles Osborne

School of Informatics
University of Edinburgh
miles@inf.ed.ac.uk

February 10, 2012

Overview

- Lecture One:
  - Cloud Computing, Big Data
  - Background material: implications for working at scale, economics
- Lecture Two:
  - Hadoop: how to process Big Data using rubbish machines
  - Programming model well suited to Big Data
- Lecture Three:
  - Randomised algorithms: how to process Big Data when we don't have enough resources
  - Examples from machine translation and finding events in Twitter

The Petabyte Age

Wired article (June 2008):

> *Sixty years ago, digital computers made information readable. Twenty years ago, the Internet made it reachable. Ten years ago, the first search engine crawlers made it a single database. Now Google and like-minded companies are sifting through the most measured age in history, treating this massive corpus as a laboratory of the human condition. They are the children of the Petabyte Age.*

What does a Petabyte look like?

A Petabyte is a lot of data:

- 1PB = 1024 TB; 1TB = 1024GB
- 1PB: 13 years of HD Video
- 1.5PB: 10 billion photos on Facebook
- 20PB: Amount of data processed by Google *each day*

Source:
http://mozy.com/blog/misc/how-much-is-a-petabyte/

The Petabyte Age: Advertising

Optimising advertising is a multi-billion dollar business

- Advert placement on pages, pricing
- Fraud detection

Analysing query logs, web-page clicks etc –and quickly– is vital for success

The Petabyte Age: Predicting 'Flu

There is a belief that we are due for a 'Flu Pandemic

- People tend to *search* for flu-related terms when they have it:
- Google mined query logs between 2003 – 2007
- Simple machine learning techniques to predict Flu levels in the US

Results were often 1 - 2 weeks *ahead* of traditional monitoring
http://www.google.org/flutrends/

The Petabyte Age: Tackling the Real Problem

Many language tasks are small scale:

- Anything published in ACL prior to 2001

But some fields involve massive amounts of data:

- Machine translation
- Social Media
- IR

The Petabyte Age: Tackling the Real Problem

Controversial points:

- Using lots of data can be a more reliable way to improve results than hoping for some magical insight using small amounts of data
- Results we obtain across a range of training set sizes are more compelling than those we obtain with just small amounts of data
- At scale, simple techniques work

But nothing is free and this brings its own set of problems ...

Big Data

Big Data is a relative term

- If things are breaking, you have Big Data
- Big Data is not always Petabytes in size
  - Big Data for You may not the same as for Google

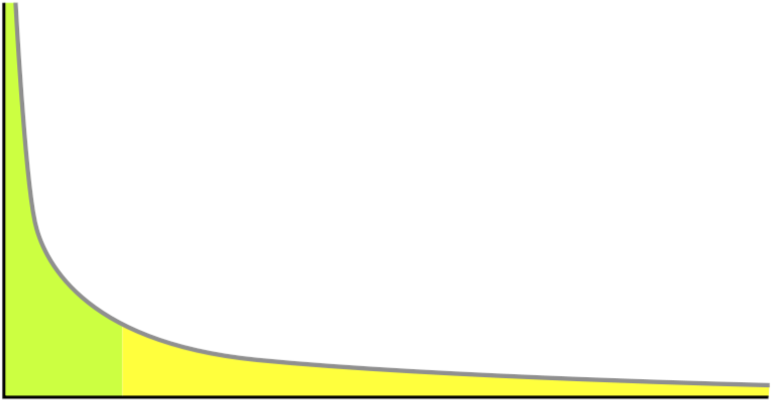
Big Data is often hard to understand

- A model explaining it might be as complicated as the data itself
  - In machine translation, our models may be bigger than the data

This has implications for Science

Big Data: Power Laws

Big Data typically obeys a power-law:



Source: Wikipedia

Comments

Modelling the head is easy, but may not be representative of the full population

- The real challenge involves dealing with the tail
- How can you learn from an example that only occurs once? (This is not sparse)

Comments

Challenges:

- Storing it is not really a problem
  - Disk space is cheap
- Efficiently accessing it and deriving results can be hard
  - Reports should be produced *now*, not decades later
- Visualising it can be next to impossible
  - How can you comprehend 1 trillion Web pages?

Problems: Repeated Observations

What makes Big Data big are repeated observations:

- Mobile phones report their locations every 15 seconds
- People post on Twitter $>$ 100 million posts a day
- The Web changes every day

Potentially we need *unbounded resources*

Problems: Access

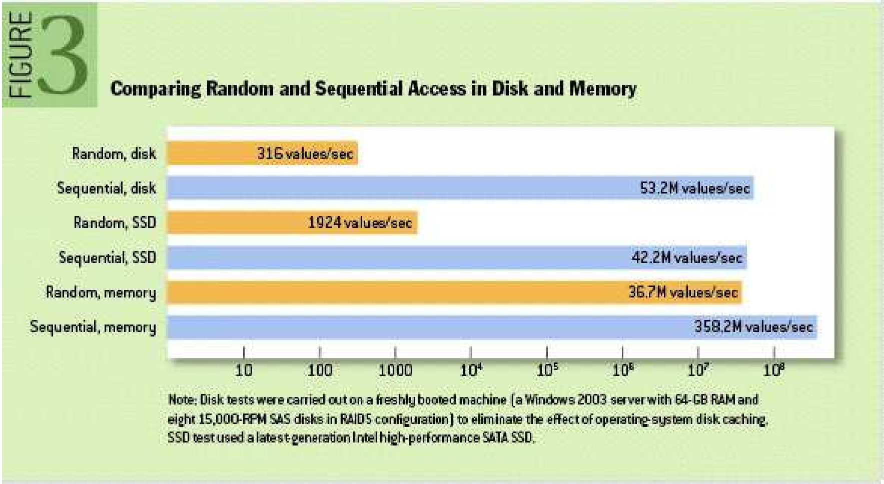
Often we want random access to data

- Find the interests of my friend on Facebook
- Tell me the probability of some sentence

But what if the Data is too big to fit into memory?

- We can start using disk etc, but poor decisions can make processing too slow

Problems: Access



Source: Jacobs, *The Pathologies of Big Data*

Problems: Denormalising

Arranging our data so we can use sequential access is great

- But not all decisions can be made locally
  - Finding the interest of my friend on Facebook is easy
  - But what if we want to do this for another person who shares the same friend?
- Using random access, we would lookup that friend.
- Using sequential access, we need to localise friend information
- Localising information means duplicating it

Denormalising data can greatly increase the size of it

Problems: Non-uniform Allocation

Distributed computation is a natural way to tackle Big Data

- Map-Reduce encourages sequential, disk-based, localised processing of data
- Map-reduce operates over a cluster of machines

One consequence of Power Laws is uneven allocation of data to nodes:

- The head might go to one or two nodes
- The tail would spread over all other nodes
- All workers on the tail would finish quickly.
- The head workers would be a lot slower

Power Laws can turn parallel algorithms into sequential algorithms

Problems: Curation

Big Data can be the basis of Science:

- Experiments can happen *in silico*
- Discoveries can be made over large, aggregated data sets

Data needs to be managed (curated):

- How can we ensure that experiments are reproducible?
- Whoever owns the data controls it
- How can we guarantee that the data will survive?
- What about access?

Growing interest in *Open Data*

Midway Summary

- Introduced notion of Big Data
- Looked at various problems
- Motivated some of the later techniques

Pay-as-you-go

Tackling Big Data means using lots of machines

- We can do it ourselves
- Or we can rent it: computing in the cloud

Pay-as-you-go

A major argument for Cloud Computing is pricing:

- We could own our machines
  - . . . and pay for electricity, cooling, operators.
  - . . . and allocate enough capacity to deal with peak demand

Since machines rarely operate at more than 30% capacity, we are paying for wasted resources

Pay-as-you-go

Pay-as-you-go rental model:

- Rent machine instances by the hour
- Pay for storage by space/month
- Pay for bandwidth by space/hour
- No other costs

This makes computing a *commodity* (sewage, electricity etc)

Pay-as-you-go: Renting a Super Computer

How much would it cost to rent a Super Computer for an hour?

- Amazon Web Services charged \$1.60 per hour for a *large instance*
  - an 880 large instance cluster would cost \$1,408
- Data costed \$0.15 per GB to upload
  - Assume we want to upload 1TB
  - This would cost \$153
- The resulting setup would be #146 in the world's top-500 machines

Total cost: \$1,561

search for (first hit): **LINPACK 880 server**

Pay-as-you-go: Renting a Super Computer

Update: 2011:

- AWS
- 30,472 cores
- 27TB RAM
- 2PB of disk space

cost: $1,279 per hour

http://arstechnica.com/business/news/2011/09/30000-core-cluster-built-on-amazon-ec2-cloud.ars

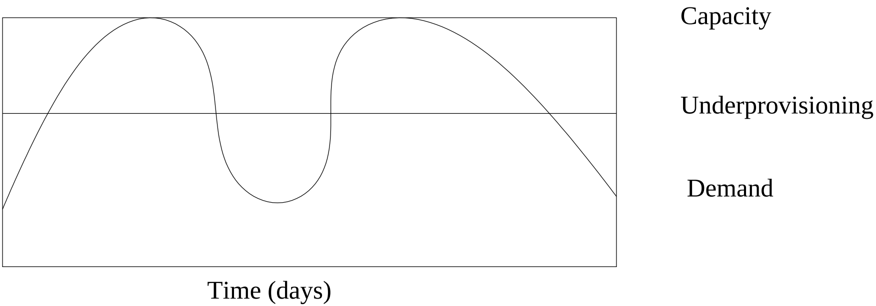
Pay-as-you-go: Provisioning

We can quickly buy resources as demand dictates

- Demand might surge, in which case we spin-up more instances
- Demand might drop, in which case we drop instances

Elastic provisioning is crucial

Provisioning



Provisioning

Example

Target (US retailer) uses Amazon Web Services (AWS) to host **target.com**

- During massive spikes (November 28 2009 –"Black Friday") **target.com** is available.
- Other retailers experience severe performance problems
  - **sears.com** had site crashes November 28 2008

http://www.webhostingunleashed.com/features/server-meltdowns-millions-020309/

Obstacles

Cloud Computing is more than just pricing:

- Can availability be guaranteed?
- What about data lock-in (and third-party control)?
- Latency?
- Privacy and Security?

This is the *total cost of ownership*

Obstacles: Availability

Organisations may demand 99.999% availability:

- 5.26 minutes downtime per year

Few bespoke enterprises are this good

- Multiple / redundant resources can boost uptime
- But what if the provider goes bust?
- Denial-of-service attacks can threaten availability
  - DDoS is itself a cloud computing notion

All of these possibilities can be tackled by spending more money (eg replicating service on another provider)

Obstacles: Data Lock-in and Third-Party Control

A serious concern is *lock-in*:

- 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

Obstacles: Data Lock-in and Third-Party Control

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*
- Anti-terror laws mean that providers have to grant access to governments
  - . . . and this privilege can be over-used

Obstacles: Data Lock-in and Third-Party Control

Government Requests to Google

Country	Requests
Brazil	3663
US	3580
UK	1166
India	1061

Requests to Google and YouTube (July 1 2009 – Dec 31 2009)
http://www.google.com/governmentrequests/

Obstacles: Latency

High Performance Computing often demands *low latency*:

- How quickly data moves around the network
- Note: total system latency is a complex function of memory, cpu, disk and network speeds
  - Often the CPU speed is only a minor aspect

Examples:

- Algorithmic Trading (put the data-centre near the Exchange); whoever can execute a trade the fastest wins
- Simulations of physical systems
- Search results
- Real-time Machine translation

Obstacles: Latency

Business Latency Examples

- Google 2006: increasing page load time by 0.5 seconds produces a 20% drop in traffic
- Amazon 2007: for every 100ms increase in load time, sales decrease by 1%
- Google's web search algorithm now rewards pages that load quickly

source: http://net.tutsplus.com/articles/general/supercharge-website-performance-with-aws-s3-and-cloudfront/

Obstacles: Latency

Low latency can be problematic in a pay-as-you-go model:

- Jobs might share resources and contend for it
- Fast networking is expensive

Hosting clusters near the client reduces latency

- Faster networking helps

Obstacles: Privacy and Security

People will not use Cloud Computing if *trust* is eroded:

- Who can access it?
  - Governments?
  - Other people?
- Privacy guarantees needs to be clearly stated and kept-to

Obstacles: Privacy and Security

Privacy Breaches

- Numerous examples of Web mail accounts hacked
- Many many cases of (UK) governmental data loss
- TJX Companies Inc (2007): 45 million credit and debit card numbers stolen

Summary

- Cloud Computing adaptation is driven by economics.
- The risks and obstacles behind it are complex
- Computing as a commodity is likely to increase over time