FIT1043 Lecture 10
Introduction to Data Science

Mahsa Salehi[*]

Faculty of Information Technology, Monash University

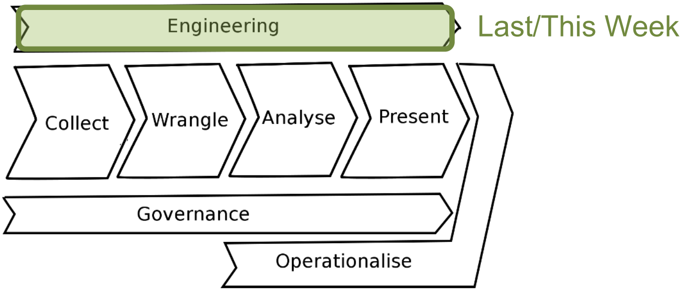
Semester 2, 2022

[*]with material from © Wray Buntine

Unit Schedule

Week	Activities	Assignments
1	Overview of data science	Weekly Quizzes
2	Introduction to Python for data science	
3	Data visualisation and descriptive statistics	
4	Data sources and data wrangling	
5	Data analysis theory	Assignment 1
6	Regression analysis	
7	Classification and clustering	
8	Introduction to R for data science	
9	Characterising data and "big" data	
10	Big data processing	Assignment 2
11	Industry guest lecture	
12	Issues in data management	Assignment 3

Our Standard Value Chain



Discussion: Unix Shell

Useful for managing and manipulating **large files**

- **without ever loading them fully into memory**
- using pipes allow us to process files as a stream
- allows us to deal with files that are too big for applications and/or don't fit into memory

Shell contains many useful commands, like

- less to view large files
- grep to search large files
- awk to process them one line at a time (and cut them down to size for visualising)

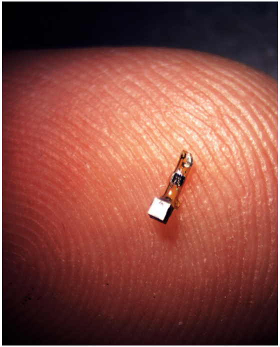
FLUX Question
New Classes of Computing

Remember Bell's law ... new classes of computing every decade.

Can you suggest some new classes of computing?

Discussion:
New Classes of Computing

mind-reading or mind-control devices



in-body devices

NB. sounds like science fiction but we know R&D exists in all these areas!

Outline

- Different databases
  - storing and accessing data
- Introduction to distributed processing
  - Map-reduce
  - Hadoop
  - Spark

Learning Outcomes (Week 10)

By the end of this week you should be able to:

- Characterize different database types
- Differentiate between SQL and NoSQL databases
- Define what distributed processing is
- Analyse the Map-Reduce framework
- Differentiate between Hadoop and Spark
- Apply R/shell commands to read/manipulate big data files

Big Data Processing

processing data at scale, especially for analysis

- databases

  storing and accessing data

- distributed processing

  breaking up computation to scale it up

Business Context

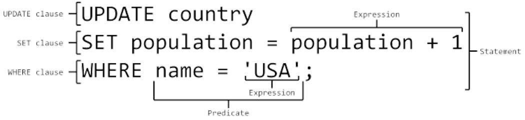
- Businesses function in a continuously changing environment:
  - Fixed formats as per RDBMS not suitable
  - Usage varies, requires complex analytical queries
- Need to reach insights faster and act on them in real time
  - Stream processing

Big Data Processing:
Databases

storing and accessing data

SQL Review

- Relational Database Management Systems (RDBMS)
- SQL ::= structured query language



- It is like a large scale set of Excel spreadsheets with better indexing and retrieval
- Transaction oriented with support for correctness, distribution, ... (ACID)

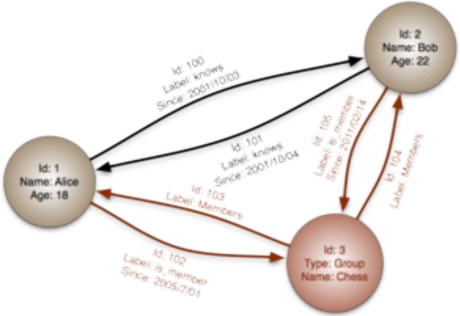
ACID: atomicity, consistency, isolation, and durability

JSON Example

```json
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

- no fixed format
- semi-structured, **key-value pairs**, hierarchical
- “friendly” alternative to XML
- self-documenting structure
- see some information *here*

Graph Database Example



- stores graph, commonly as triples, subject, verb, object
- commonly used to store Linked Open Data

Database Background Concepts

in-database analytics: the analytics is done within the DB

key-value: *value* accessible by *key*, *e.g.*, hash table

information silo: an insular information system incapable of reciprocal operation with other, related information systems

- if two big banks merge, then initially their RDBMSs will be siloed
- in a big insurance company, auto and home insurance customer RDBMSs may be siloed

Database Background Concepts

Many NoSQL and SQL DBs offer:

- large scale, distributed processing
- robustness achieved
- general query languages
- some notion of consistency
  - *e.g.* “eventually” as nodes spread updates

Beyond SQL Databases

Type	Notes
RDBMS	SQL
Object DB	navigate network
Doc. DB	JSON like, Javascript like queries
key-val cache	in-memory
key-val store	not in-memory but highly optimised
tabular key-val	relational-like, “wide column store”
graph DB	RDF, SPARQL,

SQL and Beyond SQL Databases (NoSQL)

- Use SQL database when:
  - data is structured and unchanging
- Use NoSQL database when:
  - Storing large volume of data with little to no structure
  - Data changes rapidly
- NoSQL databases offer a rich variety beyond traditional relational.

Overview: Databases

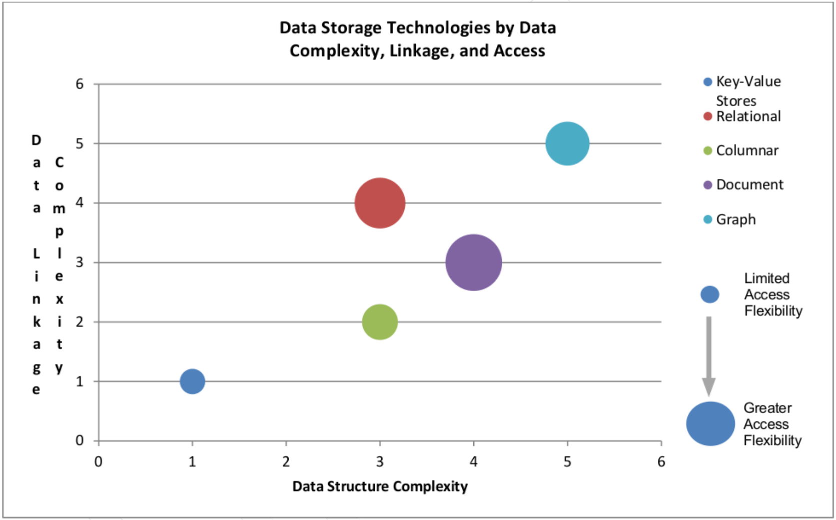


Figure 4: Data Storage Technologies

Big Data Processing:
Distributed processing

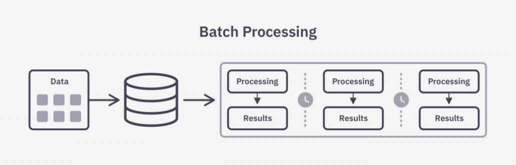
breaking up computation to scale it up

Overview: Processing

Batch: data stored and analysed in large blocks, “batches,” easier to develop and analyse

Streaming: massive data streaming through system with little storage

Interactive: bringing humans into the loop



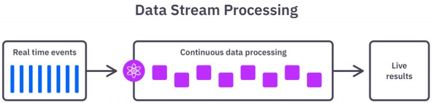


image src:quix.io/

Batch vs Stream Processing

Batch: Batch processing works well in situations where you don’t need real-time analytics results, and when it is more important to process large volumes of data to get more detailed insights than it is to get fast analytics results.

Streaming: massive data streaming through system with little storage

Sampling can be a solution to process massive datasets

Processing Background Concepts

in-memory: in RAM, *i.e.*, not going to disk

parallel processing: performing tasks in parallel

distributed computing: across multiple machines

scalability: to handle a growing amount of work; to be enlarged to accommodate growth (not just "big")

data parallel: processing can be done independently on separate chunks of data

- yes: process all documents in a collection to extract names
- no: convert a wiring diagram into a physical design (optimisation)

FLUX Question

Which one of the following tasks is very hard to make data parallel?

A. Face recognition in 1M images
B. Invert a large matrix
C. Looking for common 3-4 word phrases in a collection of documents

Distributed Analytics

- legacy systems provide powerful statistical tools on the desktop
  - SAS, R, Matlab

  but often-times without distributed or multi-processor support
- supporting distributed/multi-processor computation requires special redesign of algorithms

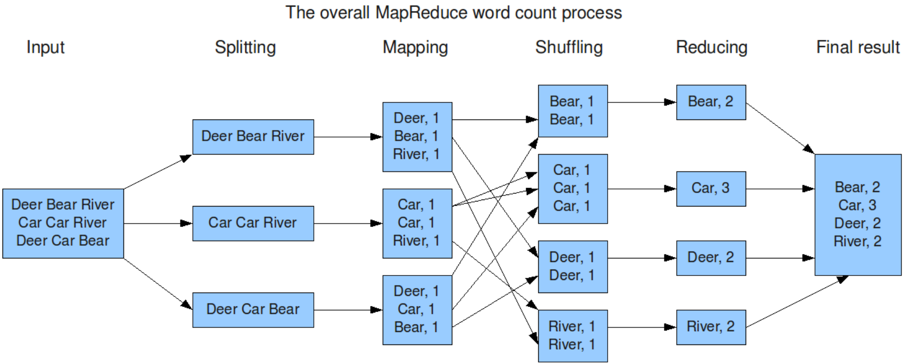
Map-Reduce

Simple distributed processing framework developed at Google

- published by Dean and Ghemawat of Google in 2004
- **intended to run on commodity hardware**; so has fault-tolerant infrastructure
- from a distributed systems perspective, is quite simple

Commodity hardware: Computer hardware that is affordable and easy to obtain. Typically it is a low-performance system that is IBM PC-compatible and is capable of running Microsoft Windows, Linux, or MS-DOS without requiring any special devices or equipment

Map-Reduce Example



for a simple word-count task: (1) divide data across machines (2) map() to key-value pairs (3) sort and merge() identical keys

Map-Reduce, cont.

- requires simple data parallelism followed by some merge ("reduce") process
- stopped using by Google probably in 2005
- Google now uses *"Cloud Dataflow"* (and *here*), available commercially, as open source

Hadoop

Open-source Java implementation of Map-Reduce

- originally developed by *Doug Cutting* while at Yahoo!
- architecture:

  Common: Java libraries and utilities
  MapReduce: core paradigm

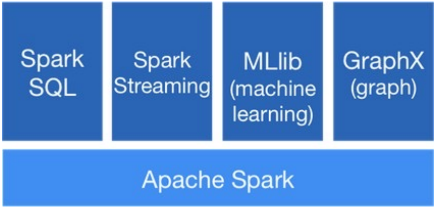
- huge tool ecosystem

- well passed the peak of the hype curve (referring to Gardner's Hype Curve)

This curve represents the maturity, adoption, and social application of specific technologies.

Spark

- another (open source) Apache top-level project at [Apache Spark]
- developed at [AMPLab] at UC Berkeley
- builds on Hadoop infrastructure
- interfaces in Java, Scala, Python, R
- provides in-memory analytics
- works with some of the Hadoop ecosystem



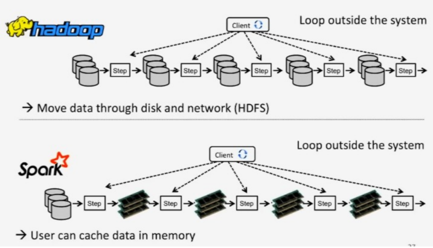
FLUX Question

Which one of the following is suitable for real-time data processing?

A. Hadoop
B. Spark

Summary: Hadoop and Spark

- Hadoop provides an inexpensive and open source platform for parallelising processing:
  - based on a simple Map-Reduce architecture
  - not suited to streaming (suitable for offline processing)
- Spark is a more recent development than Hadoop
  - includes Map-Reduce capabilities
  - provides real-time, in-memory processing
  - much faster than Hadoop

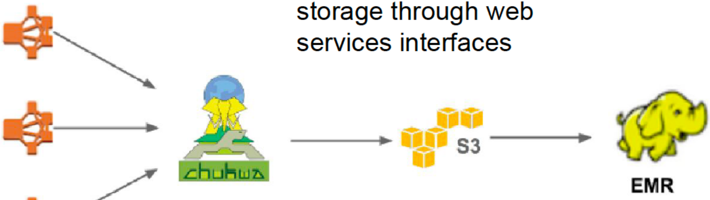


Evolution of the Netflix Data Pipeline

- Here are some statistics about Netflix data pipeline:
  - ~500 billion events and ~1.3 PB per day
  - ~8 million events and ~24 GB per second during peak hours
- There are several hundred event streams flowing through the pipeline. For example:
  - Video viewing activities
  - UI activities
  - Error logs
  - Performance events
  - Troubleshooting & diagnostic events

Netflix Data Pipeline
V1.0 Chukwa pipeline

Amazon S3 provides object storage through web services interfaces



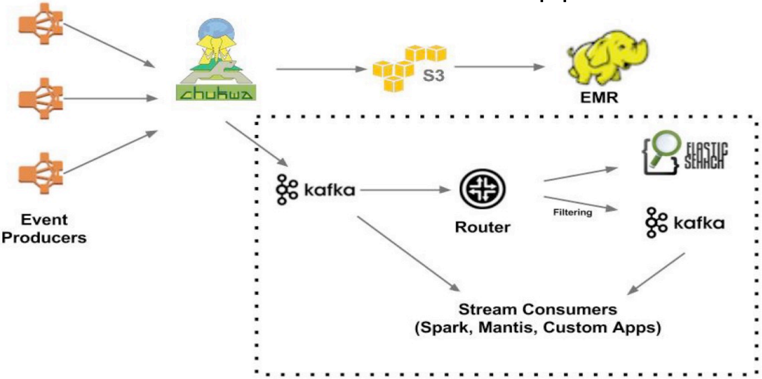
Apache Chukwa is an open source data collection system for monitoring large distributed systems

Amazon Elastic MapReduce (EMR) is a AWS tool for big data processing and analysis

V1.0: Batch jobs which usually scan data at daily or hourly frequency.

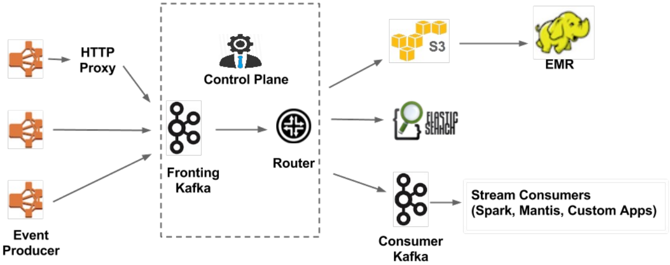
Netflix Data Pipeline: V1.5 Chukwa pipeline with real-time branch

In V1.5, approximately 30% of the events are branched to the real-time pipeline



Netflix Data Stack

Simplified view using Apache Kafka, Elastic Search, AWS S3, Apache Spark, Apache Hadoop, and EMR.



The Machine Learning Renaissance

Mike Olson (co-founded Cloudera in 2008) says without big data and a platform to manage big data, machine learning and artificial intelligence just don't work.

See *the machine learning renaissance* starting at 60 seconds.

What is Deep Learning?

- A machine learning subfield of learning representations of data.
- Deep learning algorithms attempt to learn (multiple levels of) representation by using a hierarchy of multiple layers
- If you provide the system tons of information, it begins to understand it and respond in useful ways.

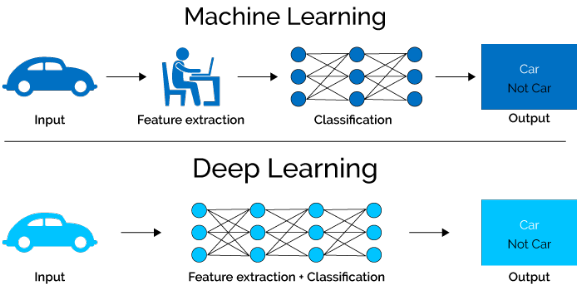


image src: quora.com

Deep Learning – People

For those who are interested, these are some of the names of people who are at the forefront of Deep Learning.

- Geoffrey Hinton
- Yann LeCun
- Andrew Ng
- Yoshua Bengio
- Fei Fei Li

Deep Learning – FIT3181

https://handbook.monash.edu/2022/units/FIT3181?year=2022

Tutorial This Week

- Manipulating large files with shell commands
- Understanding Map-Reduce: In the tutorial, we find out how to run programs in parallel using the ampersand notation: myprogram &
- Data visualization in R

Week 11

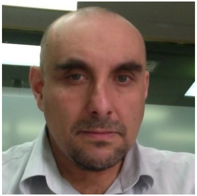
Guest lecture from Microsoft: Data and Artificial Intelligence- An Industry Perspective

Microsoft

Prashant Bhatnagar

Pursuit Lead- Data & AI

Microsoft Services



Microsoft

Stan Kotlyar

Architect, Data and Artificial Intelligence, Microsoft.