EE3801 Cheatsheet
Intro to Data Engineering
github.com/securespider

01.1 Intro
Data science vs engineering
- **Science** - Learn, optimise, analytics, aggregate and labelling
- **Engineering** - Cleaning, data storage, logging, sensors, pipelines

Data structure
Unstructured data
- Chaotic no order to data

Structured data
- Data stored access in the same format

Semi structured data
- Can contain both forms of data
- Some structure but not all data points follow same format

Big data
Volume, Variety, Variability

Velocity High rate of data generation

- Must create a robust and scalable pipeline

Raw Data
- Tend to have gaps

Data wrangling
Used to understand raw data

Discovery Understand what is in your data

Structure

Cleaning Dealing with gaps (nulls), outliers, formatting bugs

Enrichment Derive other data from other information/ additional data augmentation (feature selection)

Validation Verify data quality, sources

Publishing Give data scientist

Process
Extraction Retrieve raw data from unstructured pool and migrate to temp repo

Transformation Structure enrich and convert raw data

Loading Loading structured data into data warehouse

Data warehouse
Decision support system storing historical data from organisations

Data Pipeline
- Processing underlying raw data in ordered sequence of steps

01.2 Data Pipelines
Considerations
Big data
Velocity Streaming, captured and processed in real time

Volume Scalable wrt time

Variety Recognise and process diff formats

Business
- Handling streaming data?
- How much data to expect (Time horizon/how much storage consumed)
- What type/how much processing in DP
- Where is data source? Need micro-services?

Architecture
Batch-based DP
- Analysis of data that has been stored over a period of time
- N independent tasks to process with k stages
- Each stage takes max of T time process input
- Diff stage can operate concurrently
- $t(N,k) = T \times (N + k - 1)$

Streaming-based DP
- Processing as data flows through system
- Logging and persistent result storage

Lambda Architecture
- Combination of batch and streaming
- Separate processing engine for "batch" and "speed" layers combining in "service" layer
- Accounts for real-time streaming and historical batch analysis
- Encourage raw data storage and create new dst for queries
- Min errors for both layers reliably at fast speeds

Kappa Architecture
- Replay data and process both layers in same single stream processing engine
- Good for big data architecture with cheaper hardware and focus on stream

Design
1. Identify application and decide if DP needed
2. Identify DP category (architecture)
3. Understand working mechanism, parameters/variables

04. Big Data Computing Technology Platform
- Collection of interconnected stand-alone computers
- Work collectively and cooperatively as an integrated computing resource pool
  - Clusters exploit massive parallelism at job level
  - Achieves high availability through stand-alone operations
  - **Fault tolerance** - If one goes down, other can take over

Benefits
- Scalable performance, high availability, fault tolerance
- Modular growth and use of commodity components

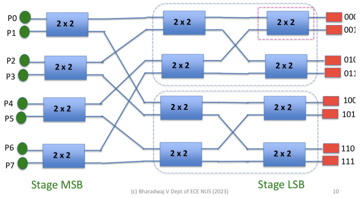
Beowulf Cluster
- Single compute job requires frequent communication among cluster nodes
- Cluster share dedicated network
- Nodes are homogeneous and coupled
- eg. Each process requires information from other process

Scalability
Limited by:
- Multicore chip technology, cluster topology, packaging method, power consumption and cooling scheme
- Memory capacity, disk IO bottlenecks, latency tolerance

Packaging
Compact Nodes closely packaged in racks where nodes are not attached to peripherals

Slack Nodes attached to peripherals connected remotely

Interconnection Medium
Considerations
1. Available link speeds
2. Message Passing Interface (MPI) latency
3. Network processor/routing mechanism/flow control
4. Differing network topologies

Self routing/Destination tag


- Every processor can be routed to every memory without external controller
- Switch should know what stage it is in to know which bit to look for
- Bit of stage defines which output interface it leaves (0-above, 1-below)

Control
Centralized Nodes owned, ctrl by central operator

- Easy to manage
- Used by compact and slack clusters

Decentralized Nodes have individual owners

- Minimize coupling and can be used w many OS
- Only slack can have

Homogeneity
Homogeneous Nodes from same platform (architecture and operating system)

Heterogeneous Nodes of different platforms and different operating systems

- Can run and share with everyone

Programmability
- Cluster Operating System (COS) must provide user friendly interface btw user, application and hardware
- Additional features like single-system image and system availability
- Ensure failure management, load balancing and tools for parallelizing computation

COS Examples
Solaris MC Prototype, distributed operating system for multi-computers

MOSIX Software package that extends the Linux1 kernel with cluster computing capabilities

- Allows any size cluster of Intel-based computers to work together like single system like SMP (Symmetrical Multi Processor)

GLUnix Global Layer Unix - Global operating system for Berkeley's Network of Workstations (NOW)

- NOW's goal was to construct a platform that would support both parallel and sequential applications on commodity hardware
- Has load balancing capability

Window-based Load Balancing technique

1. Core node C broadcast REQ msg to all nodes for status collection
2. Each node sends current load info to C
3. C computes average load and deficit/surplus for each node
4. C sends MIGRATION msg to each node to transfer excess to deficit nodes
5. After some time w, after migration, C repeats process

- Centralised controlled by core node
- w impt to decide as
  1. Communication overhead with frequent trf
  2. C may be working with outdated info

Security

- Intra-cluster communication can be **exposed/enclosed**

Exposed cluster Communication paths among the nodes exposed to outside world

- Using standard protocols eg. TCP/IP
- ICC need effort to ensure privacy and security
- Outside communications may disrupt ICC in unpredictable fashion
- May have high overhead

Enclosed Shielded from outside world

- No standard for efficient enclosed ICC

Resource sharing

- Clustering improves both availability and performance
- High Availability clusters use hardware redundancy for scalable performance
- All connects to NIC component in node

Share-nothing Each node do itself and send results together after

- Simple to configure and used in most clusters
- Nodes connected through I/O bus eg Ethernet

Shared-disk When one node fail the other take over

- Used in small-scale availability clusters in business applications
- Fault tolerance via checkpoints, rollback, failover..

Shared-memory All common data/instruction written in shared space

- Nodes connected by Scalable Coherence Interface (SCI) ring which is connected to memory bus of each node through NIC module
- Memory bus operated at higher frequency than the I/O bus

Cloud Computing Platforms

- Cluster and Grid computing leverage use of many computers in parallel to solve problems of any size
- Utility and Software as a Service provide computing resources as a service (pay per use)
- Utility/Cloud computing - High throughput computing paradigm
  - Provides services through large data center or distributed server farms
  - Leverage dynamic resources to deliver large number of services to end users
  - Enables users to share access to resources from anywhere with connected devices
  - Frees low-level tasks of setting up hardware
  - Manage system software at low cost and easy-to-use manner
  - Applies virtual platform with "elastic resources"
  - Comprise of Core Layer ↔ Aggregate Layer ↔ Access layer ↔ Leaf nodes

Layers

Access Provide network access to user

- Connect distribution layer switches to perform network foundation functions
- Routing, Quality of Service (QoS) and security

Aggregate Provides hierarchy for security and farm services

- Access Control List (ACL), Firewalls, Intrusion Prevention Systems (IPS)

Core Fabric for high-speed packet switching btw aggregation modules

- Gateway to campus core where other modules connect
- eg. Connects to extranet, WAN or internet edge

Accessibility

- Compute probability of successful access to the cloud
- Data Center (DC) comprise cloud layers + core, aggregate, access layers and leaf nodes in hierarchical system

Cloud vs Data Center

FEATURE	DC	Cloud
Scalability	Limited; depends on the capacity of the storages, servers, etc	Easily scalable – pay-as-you-go!
Security	Governed by local norms	One of the QoS factors for CSP!
Cost	High	Pay-as-you-go! Compute/Storage resources made available at cheaper costs!
Availability	Entire control on organization; their norms may dictate policies;	Largely governed by SLAs imposed by CSP; This often may provide better guarantees.

Virtualization

- Computer file/Image that behaves like an actual computer
- VM runs in a separate window like a program giving end user an identical experience as on host system
- VM is sandboxed from rest of system
  - Risky operations eg testing operating system or malwares
  - Running software on different OS/ OS backups

Implementation

- Hypervisor installed on physical hardware used to create and amange VMs
- Each VM has virtual computing resources and can run simultaneously
- Multiple OS run side by side using hypervisors to manage

4.2 Big Data Computing Technology

- Platforms and Cloud Security - On Cloud
- Data Security and Storage
  - Security and privacy in cloud platforms
  - Data security - components and issues
  - Data Integrity, Confidentiality, Availability, Privacy
  - Commonly used data encryption algorithms in Cloud
  - Distributed Data Storage

Service Models in Cloud Platforms

Software as a Service (SaaS) Software with related data deployed by Cloud Service Provider (CSP) that users can use through web browsers

Platform as a service CSP facilitates service to the users by providing certain cloud components to certain software that can solve specific tasks

Infrastructure as a Service CSP facilitates services to the users with virtual machines and storage to improve business capabilities

Cloud Categories

Public clouds Owned and operated by third-party CSP and delivered over the internet

- Low-cost and scalability (pay-as-use)
- High reliability
- No maintenance on User's side

Private Computing and Storage resources used exclusively by one specific business/organisation

- These can be physically located at organisation's on-site DC or hosted by third-party
- All equipment and resource deployment depend on local policies and norms
- Scalability (but may not be low-cost)
- Highly secured storage and access
- Custom-driven local environments can be created as per the needs of the organisation

Hybrid Combination of public/private

- Take advantage of secured on premise infrastructure while using public cloud service
- Use public cloud feature for high-volume data handling
- Commonly used w lower security needs - web based email/web-page hosting

Security and privacy in cloud platform

- Combination of data integrity, confidentiality, availability and privacy
- Prevention of unauthorized disclosure, witholding, amendment or deletion of information

Data Integrity

- Protecting data from unauthorized deletion, modification or fabrication
- Manage entity's admittance and rights to specific enterprise resources
- Ensure valuable data and services are not abused misappropriated or stolen

Data storage - Databases

- Easily achieved in a standalone system with single database
- Maintained via database constraints, transactions that follow ACID properties

  Atomicity All operations are treated as atomic/single
  - Failure of transaction will restart/rollback to earlier saved state

  Consistency Mechanism that enforce rules across all nodes storing data

  Isolation Manage concurrent access without affecting other nodes

  Durability Guarantees data is saved safely after transaction admist failures while updating
  - Data is locked until transaction is completed
  - Results are first written into local transaction logs and written into entry after work done

- Important in datalakes and DWH
- Allows users to see consistent views of data even while new data modified in real-time
- Trust stored data

How to achieve

- **RAID** - Array of stored disks
- Avoid unauthorised access
- Monitoring mechanisms to have greater transparency on any altered data

Data Confidentiality

- Facilitates storing users' private or confidential data in cloud
- Authentication and access control strategies
- No trust in CSP as cannot store sensitive data (insider attacks)
- CSP can have different subscription level for varying confidentiality

Distributed Storage

- Store data in multiple clouds or databases
  - Data divided into chunks, encrypted and stored in different databases
  - Since each segment encrypted and separately distributed, enhanced security against different attack

Data Availability

- Ensure data can be recovered and verified by techniques rather than guarantee by CSP
- Quickly and efficiently locate data
- If a node is attacked, we cannot use any direct links from a node that leads to attacked node

T-coloring problem

- Color vertices of graph st no adjacent nodes have identical colors
- Find the minimum number of colors needed

Solution

- Generate all possible coloring of nodes and backtrack by avoiding certain color possibilities

Data Privacy

- Seclude information/sensitive data to prevent adversary from inferring user behavior by visit model
- Using oblivious ram(ORAM) technology
  - Visit several copies of data to hide real visiting aims of users