**Week 1**

* Storing data has become incredibly cheap over recent years
* Shift in Data & Analytics: People used to look back into the past with their data. But nowadays data is used in a predictive forward-facing function – Data is being stored in the cloud and the end users can directly access and analyse data
* The four V’s of Big Data:
  + Volume:
  + Velocity
  + Variety
  + Veracity

**Parallelism & Scalability**

**Parallelism**

* Execute many independent tasks at once – for example grid search
  + **Data parallelism**: Execute the same task in parallel on different slices of the data
  + Example: Query processing in modern cloud databases which store partitions of the data on different machines
  + **Pipeline parallelism**: Break task into a sequence of processing stages – each stage takes results from previous stage as input, with results being passed downstream immediately

**Scalability**

Ability of a system to handle a growing amount of wark by adding resources to the system

* Scale-up: Replace machine with better machine (easier – especially in the cloud)
* Scale-out: Add more machine of the same type

Desired goal in practice:

* **Linear scalability with number of machines / cores** in scale out settings
* Elastic scaling in cloud environments

Scalability != Performance

* A common misperception is that scalable systems are automatically performant
* Scalability often comes with increased overheads

Week 2 – SQL

Why do we distribute data?

* Performance: Amount of data is growing exotically
* Elasticity: It can be easily scale up or down depending on the demand
* Fault-Tolerance: Running on more nodes than one provides better protection against hardware failures

How do we classify distributed databases?

* Scalability: Scale up vs Scale out
* Implementation: Parallel vs Distributed
  + Parallel: Runs on tighly-coupled nodes, main goal is achieved peak performance 🡪 Typically scale-up architecture
  + Distributed: Runs on loosely-coupled nodes, main goal is usually to achieve scalability, fault-tolerance or elasticity 🡪 Typically a scale-out architecture
* Application: Analytical vs Operational
  + Online Analytical Processing (OLAP): Few, complex, low-running analytical queries
  + A picture containing diagram

    Description automatically generatedOnline Transactional Processing (OLTP): Focus on multiple concurrent, simple, short-running transactional queries
* Architecture: Shared Memory vs Shared Disk vs Shared Nothing
  + Shared Memory: All nodes have shared access to both memory & disk, typical architecture found in scale-up, parallel databases, can achieve high performance
  + A picture containing diagram

    Description automatically generatedShared Disk: Nodes have their own CPU & memory, but share same disk, most found in traditional enterprise-grade RDBMs systems
  + A picture containing diagram

    Description automatically generatedShared nothing: Data is spread across independent nodes that only communicate via the network, typical architecture found in web-scale, scale out systems. Robust architecture that offers availability & scalability

**Distributed Query Processing (very important for exam)**

Text, letter

Description automatically generated

Diagram

Description automatically generated Diagram

Description automatically generated

Diagram

Description automatically generated

Possible exam question:

Graphical user interface

Description automatically generated

Diagram

Description automatically generated

Box and whisker chart

Description automatically generated

Graphical user interface

Description automatically generated with medium confidenceDiagram

Description automatically generatedA picture containing chart

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generated