THE UNIVERSITY OF MELBOURNE

COMP90050 Advanced Database Systems

Winter Semester, 2023

Lecturer: Farhana Choudhury (PhD)

Week 1 part 2

Types of Database Systems

How the data are stored –

Simple file

- As a plain text file. Each line holds one record, with fields separated by delimiters (e.g., commas or tabs)

RDBS

- As a collection of tables (relations) consisting of rows and column. A primary key is used to uniquely identify each row.

Object oriented

- Data stored in the form of 'objects' directly (like OOP)

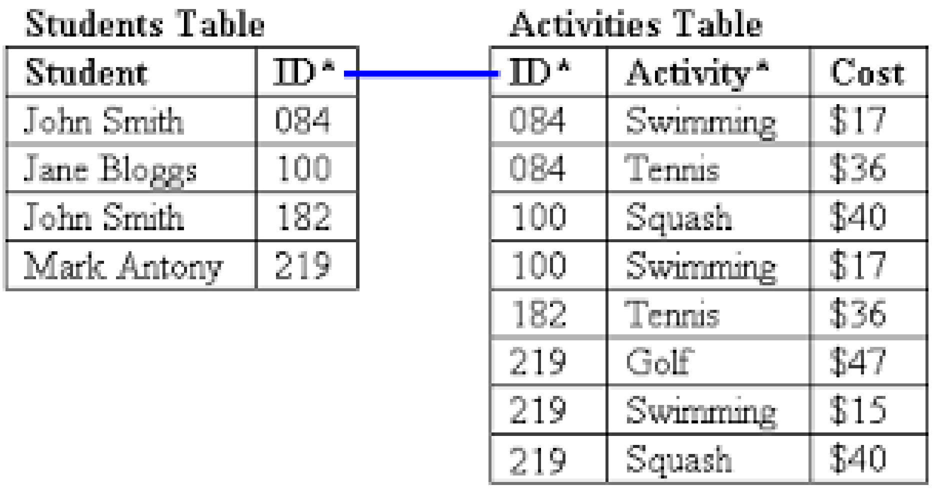
No-SQL

- Non relational – database modelled other than the tabular relations. Covers a wide range of database types.

Simple file

- Usually very fast for simple applications but can be slow for complex applications
- Can be less reliable
- Application dependent optimisation
- Very hard to maintain them (***concurrency*** problems)
- Many of the required features (that exist in relational databases) need to be incorporated - unnecessary code development and potential increase in unreliability

Relational DB systems



Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Activities Table

ID*	Activity*	Cost
084	Swimming	$17
084	Tennis	$36
100	Squash	$40
100	Swimming	$17
182	Tennis	$36
219	Golf	$47
219	Swimming	$15
219	Squash	$40

Source: http://www.databasedev.co.uk

- Very *reliable* (consistency of data – we will learn more later)
- Application independent optimisation
- Well suited to many applications, very fast due to large main memory machines and SSDs.
- Some RDB also support Object Oriented model e.g., Oracle, DB2, and XML data+queries
- Can be slow for some simple applications

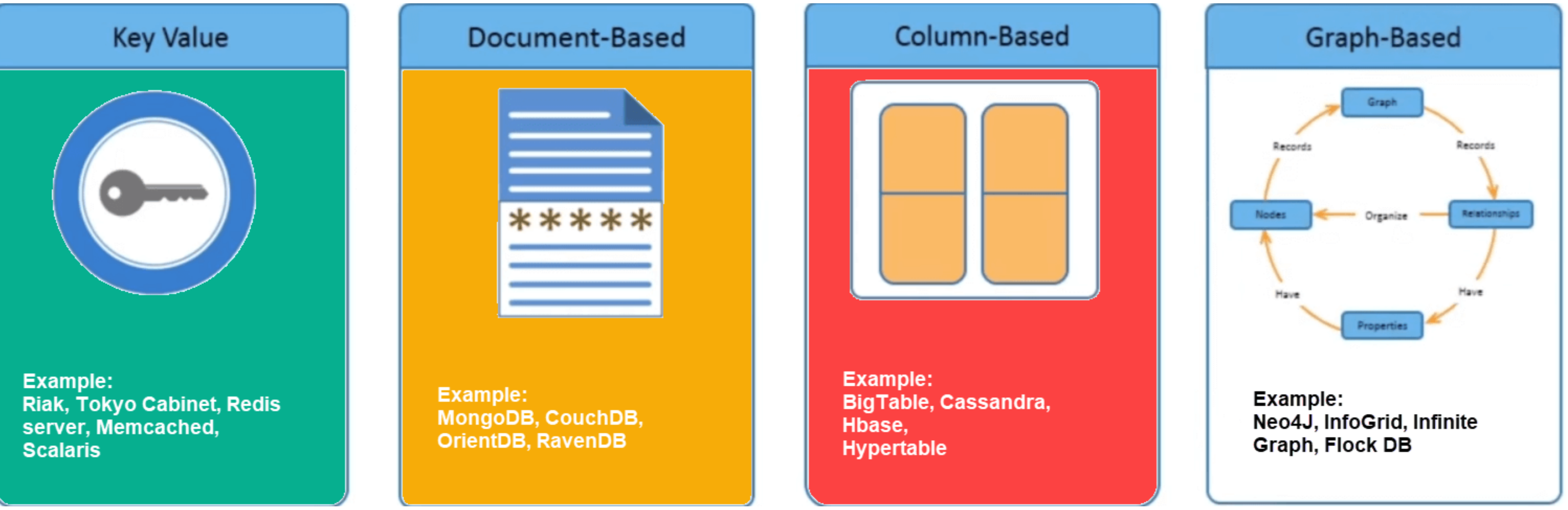
Object Oriented (OO) DB Systems

- Stores as objects directly, not tables
- May contain both data (attributes) and methods – like OOP
- Can be slow on some applications
- Reliable
- Limited application independent optimisation
- Well suited for applications requiring complex data
- Unfortunately, many commercial systems started did not survive the force of RDB technology and basically disappeared from the market.

NoSQL (also called Not Only SQL)

- Flexible/ no fixed schema (unlike RDB)
- Provides a mechanism for storage and retrieval of data modelled in means other than the tabular relations
- Simple design, should linearly scale
- NoSQL has ***compromise consistency*** and allows replications - We will discuss this more later.
- Most NoSQL databases offer "***eventual consistency***", which might result in reading data from an older version, a problem known as stale reads – we will learn more later

Types of NoSQL databases



Key Value

Example:
Riak, Tokyo Cabinet, Redis server, Memcached, Scalaris

Document-Based

Example:
MongoDB, CouchDB, OrientDB, RavenDB

Column-Based

Example:
BigTable, Cassandra, Hbase, Hypertable

Graph-Based

Example:
Neo4J, InfoGrid, Infinite Graph, Flock DB

Image source: https://jameskle.com/writes/no-sql

Key-value pair database systems

Stores data as a collection of key–value pairs, where each key is unique

Why useful - Many applications do not require the expressive functionality of transaction processing – e.g. Web search, Big Data Analytics can use MapReduce technology.

- Can be seen as a type of NoSQL database
- Used for building very fast, highly parallel processing of large data - MapReduce and Hadoop are examples
- Atomic updates at Key-value pair level (row update) only.

Some other DB systems – Deductive database systems (DDBS)

- Allows recursion
- Most of the application can be developed entirely using DDBS
- There are no commercially available systems like RDBs
- Many applications do not require the expressive power of these systems (e.g. many commerce related applications)
- Many RDBs do provide some of the functionality – e.g. supporting transitive closure operation (a form of recursion in SQL2).
- E.g.

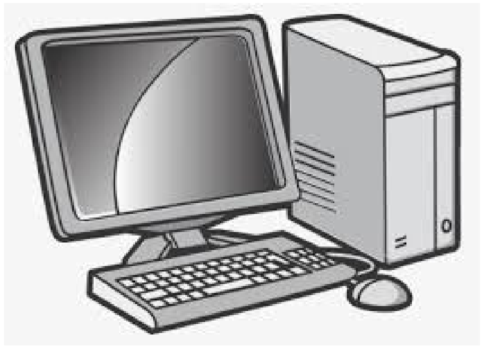
```
path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z), path(Z,Y).
```

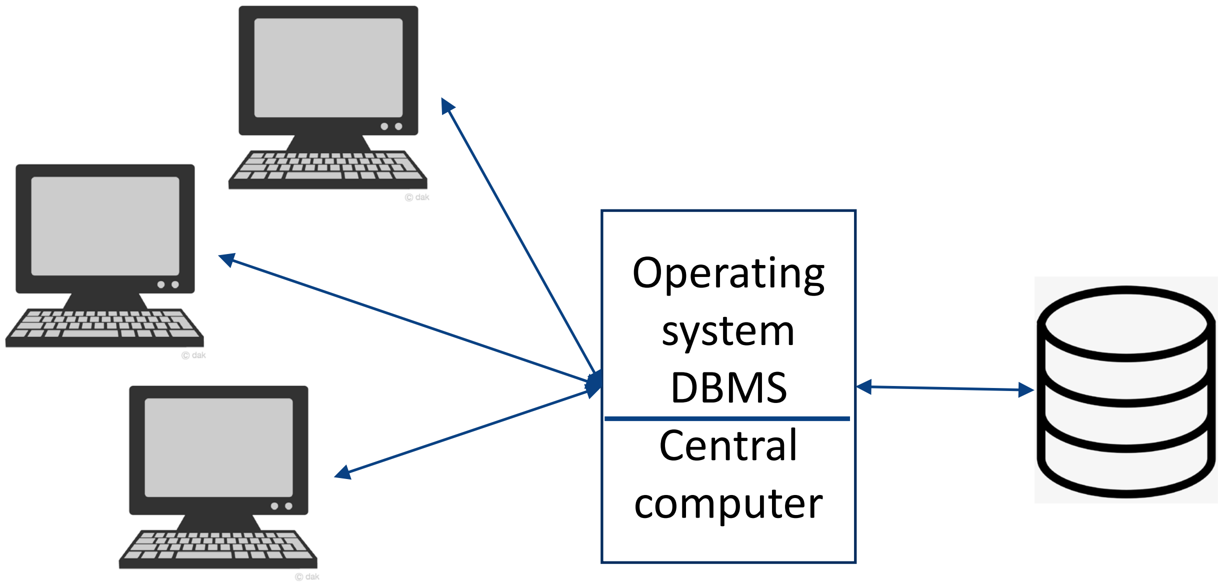
Database Architectures

Centralized	Distributed	WWW	Grid	P2P	Cloud
• Data stored in one location	• Data distributed across several nodes, can be in different locations	• Stored all over the world, several owners of the data	• Like distribu-ted, but each node manages own resource; system doesn't act as a single unit.	• Like grid, but nodes can join and leave network at will (unlike Grid)	• Generalization of grid, but resources are accessed on-demand

Database Architectures

Centralised database systems

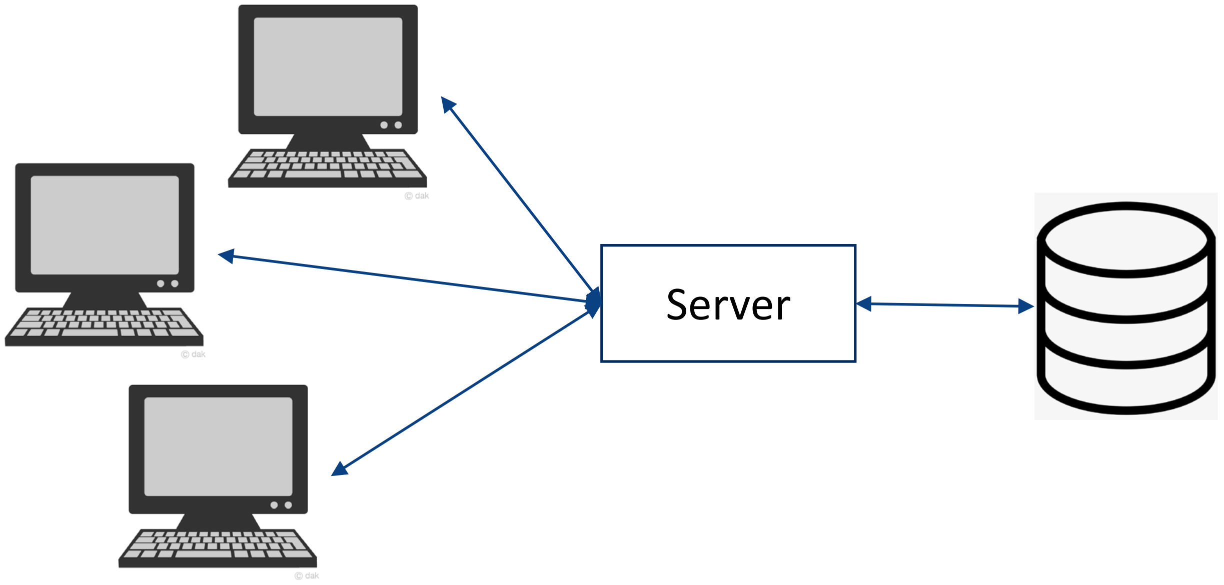




- Data in one location
- System administration is simple
- Optimisation process is generally very effective
- PC/Cluster Computing/data centres are examples of this

Database Architectures

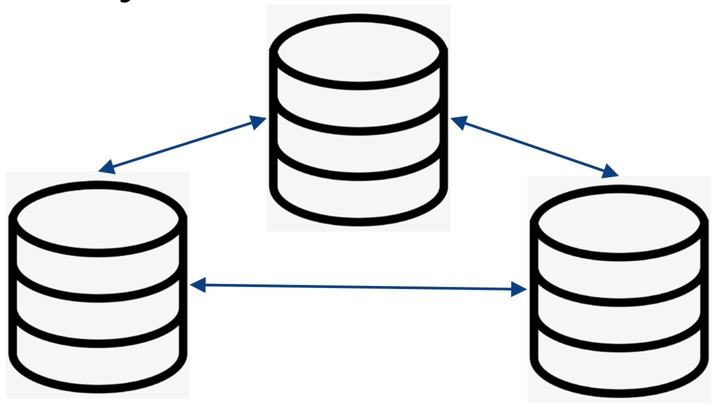
Client-server architecture that uses a centralized database server-



- A central server with a central DB in one location
  (but client and server can be in different locations)
- Client generally provides user interfaces for input and output
- Server provides all the necessary database functionality
- System administration is relatively simple
- System recovery is similar to centralised systems

Database Architectures

Distributed database systems:



- Data is distributed across several nodes in different locations
- Nodes are connected by network
- System provides concurrency, recovery and transaction processing
- System administration is very hard – usually with a single resource manager
- Crash recovery is complicated
- There are usually data replication -- potential inconsistency

Database Architectures

World Wide Web

- Data is stored in many locations
- Several owners of data - no certainty of data availability or consistency
- Optimisation process is generally very ineffective
- Evolving database technology -- no standards have been developed except in case of XML/http and some protocols for accessing data
- Security could be a potential problem
- Notions of Transactions is much more difficult to enforce

Database Architectures

Grid Computing and Databases

- Very similar to distributed database systems
- Data and processing are shared among a group of computer systems which may be geographically separated.
- Usually designed for particular purpose – e.g., a scientific application
- **Administration of such systems are done locally by each owner of the system**
- Reliability and security of such systems are not well developed or studied.
- Grid systems model is more or less outdated by Cloud Computing model

Database Architectures

P2P Databases

- Data and processing is shared among a group of computer systems which may be geographically separated as in Grid Database systems
- **Computer nodes can join and leave the network at will unlike in Grid Databases => much harder to design transaction models**
- Usually designed for particular usage – e.g. a scientific application
- Administration of such system is done by the owners of the data

Database Architectures

Cloud computing

https://www.youtube.com/watch?v=dH0yz-Osy54

Database Architectures

Cloud computing

Cloud computing offers online computing, storage and a range of new services for data and devices that are accessible through the Internet.

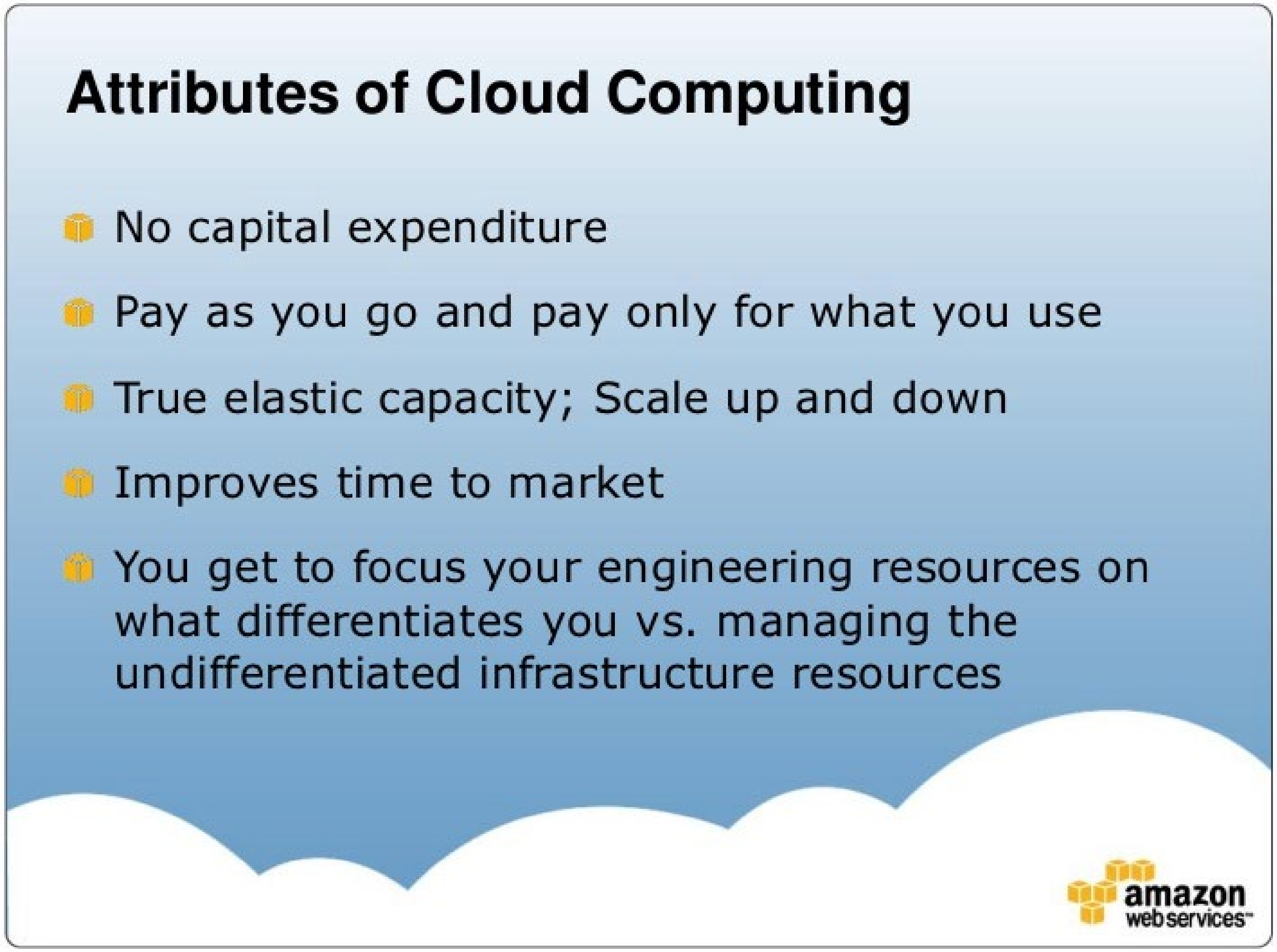
User pays for the services just like phone services, electricity, etc.

Huge potential for developing applications with minimal infrastructure costs

Cloud services offered in several forms:

- Iaas – Infrastructure as a service (provide virtual machines)
- Paas – Platform as a service (Provide environment like Linux, Windows, etc.)
- Saas – Software as a service (Specific applications like RDB, Mail, etc.)

An example: AWS: Amazon Web Services

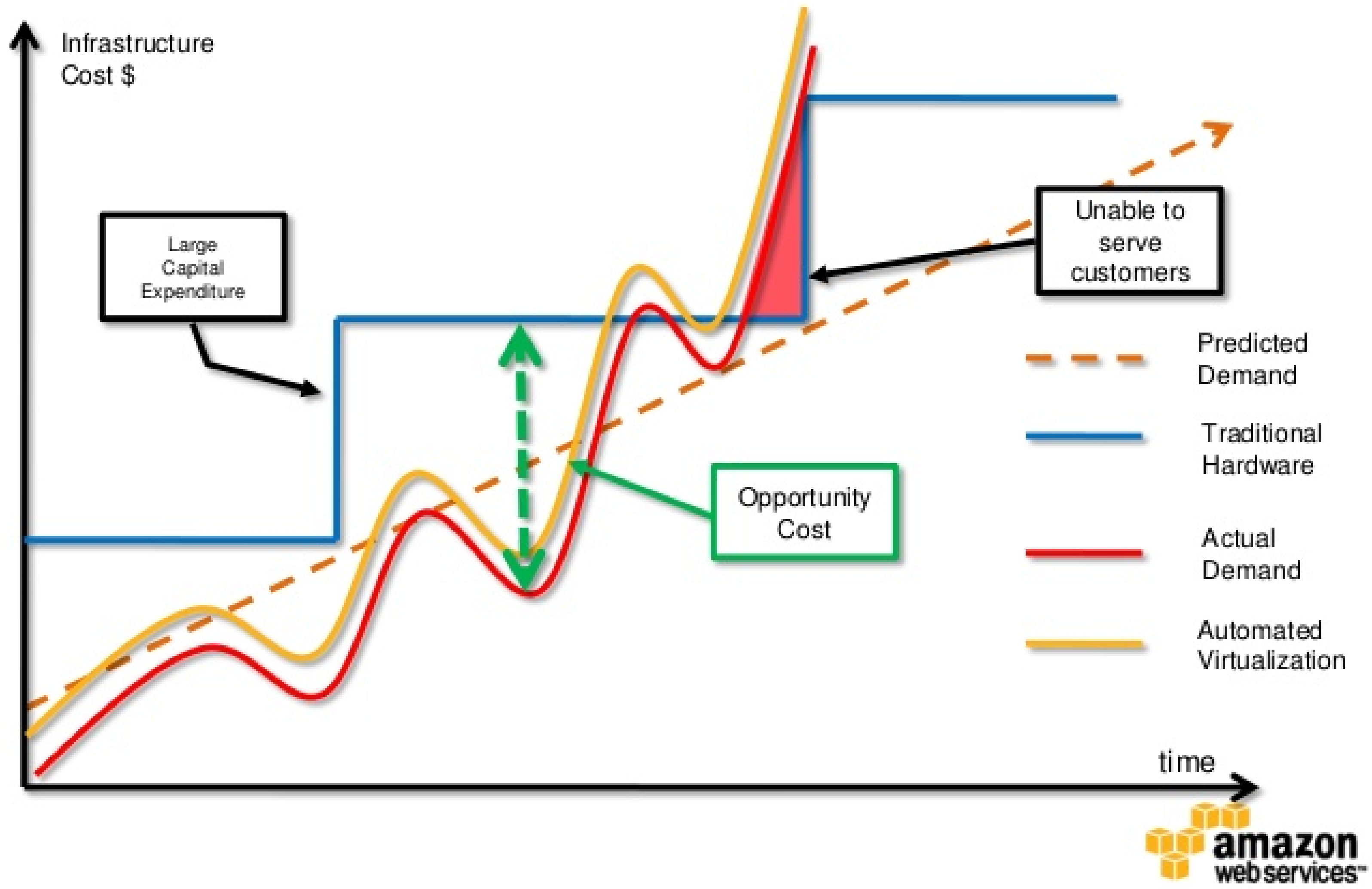


Attributes of Cloud Computing

- No capital expenditure
- Pay as you go and pay only for what you use
- True elastic capacity; Scale up and down
- Improves time to market
- You get to focus your engineering resources on what differentiates you vs. managing the undifferentiated infrastructure resources

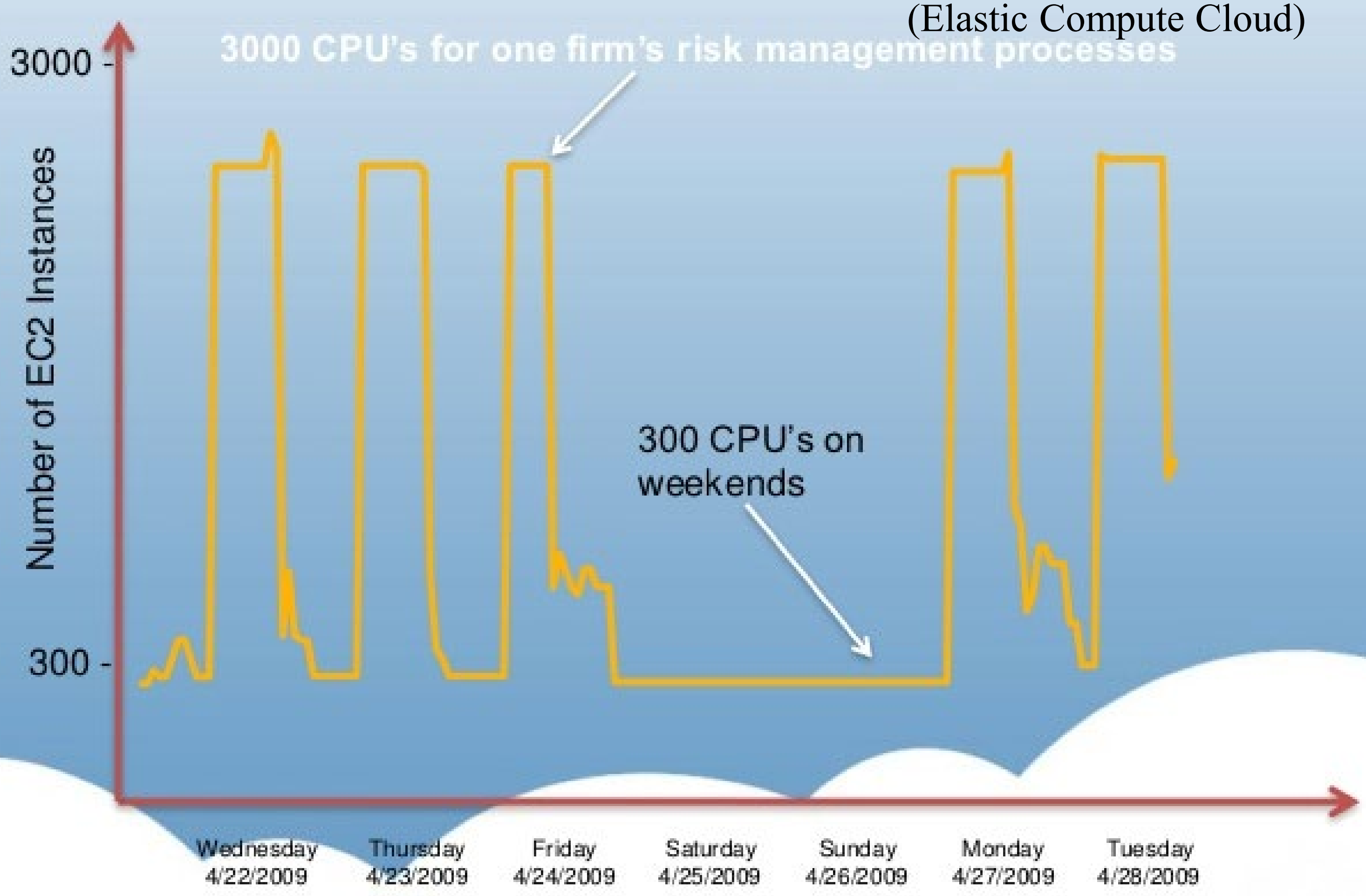
amazon
web services

Elastic and Pay-Per-Use Infrastructure



Example: Wall Street App on Amazon EC2

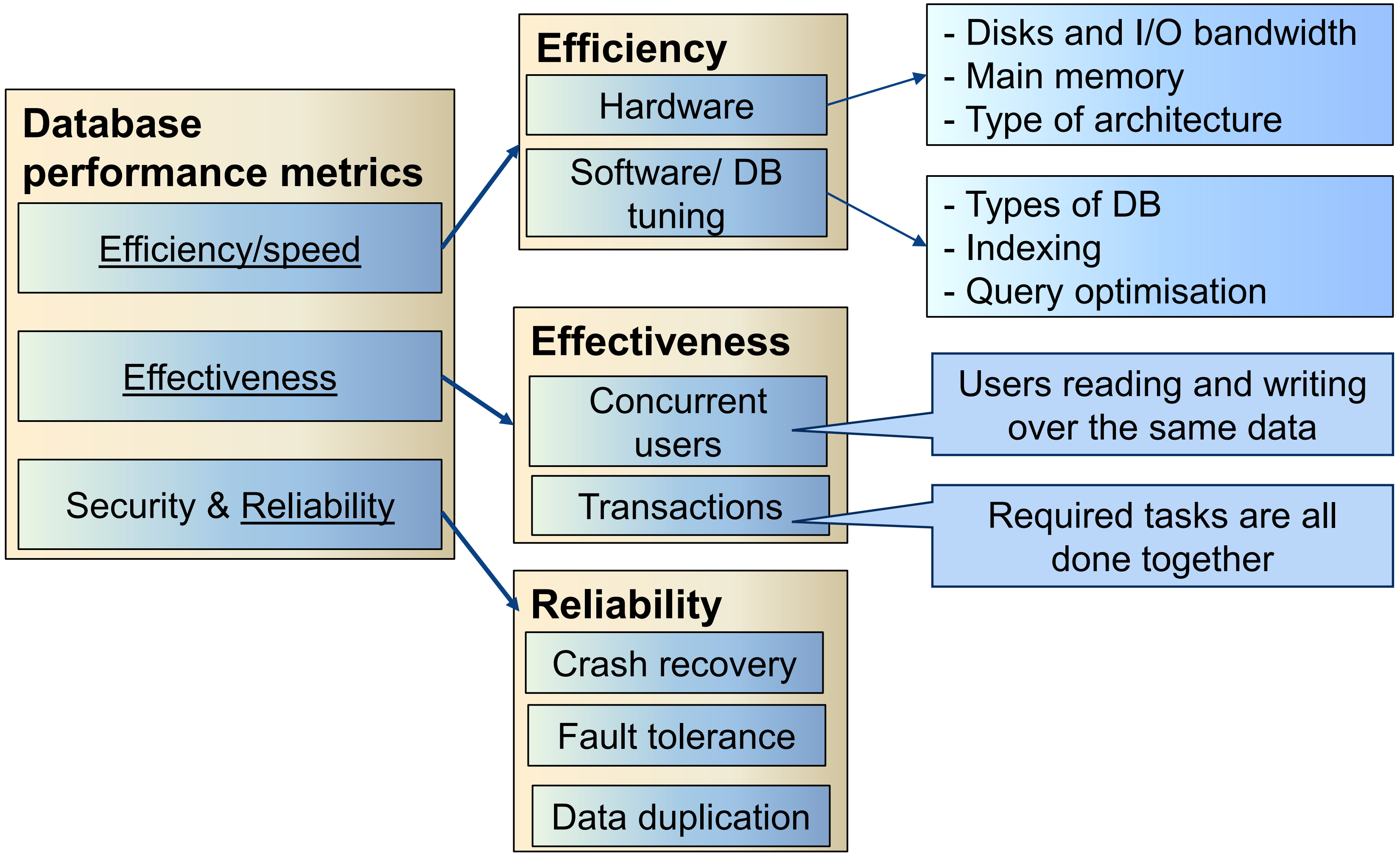
(Elastic Compute Cloud)



Database Architectures

Centralized	Distributed	WWW	Grid	P2P	Cloud
• Data stored in one location	• Data distributed across several nodes, can be in different locations	• Stored all over the world, several owners of the data	• Like distributed, but each node manages own resource; system doesn't act as a single unit.	• Like grid, but nodes can join and leave network at will (unlike Grid)	• Generalization of grid, but resources are accessed on-demand

Core Concepts of Database management system


Database performance metrics
Efficiency/speed
Effectiveness
Security & Reliability
Efficiency
Hardware
Software/ DB tuning
- Disks and I/O bandwidth
- Main memory
- Type of architecture
- Types of DB
- Indexing
- Query optimisation
Effectiveness
Concurrent users
Transactions
Users reading and writing over the same data
Required tasks are all done together
Reliability
Crash recovery
Fault tolerance
Data duplication