COMP283-Lecture 8
Applied Database Management

Introduction	
High Availability	Redundancy
	Distribution

COMP283-Lecture 8
High Availability: Introduction

- High Availability requires that all components of a system from end-to-end provide an uninterrupted service.
- The components include network, storage, servers, databases, and data.
- High availability can be achieved through redundancy or very fast recovery.
- Identify any single point of failure.
  - e.g. network infrastructure
- Identify the goals for your system.
- High Availability does NOT 100% guarantee the data!

COMP283-Lecture 8
High Availability: Clustering

- Clustering uses multiple servers to serve a shared database system.
- Each server within a cluster is called a node.
- If the primary node fails, then the virtual database service will become active on one of the secondary nodes.
- low disk and network latency essential - why?

COMP283-Lecture 8
High Availability: Clustering

- Some clustering systems operate at the Operating System level – not the database level e.g. MS-SQL
- MySQL InnoDB cluster is a collection of products that work together
  - A group of MySQL servers can be configured to create a cluster using *MySQL Shell*
  - The cluster has a single master (the primary), which acts as the read-write master.
  - Multiple secondary servers are replicas of the master.
  - A minimum of three servers are required to create a high availability cluster.
  - A client application is connected to the primary via *MySQL Router.*

COMP283-Lecture 8
High Availability: Clustering: Aside: InnoDB

- InnoDB is a general-purpose storage engine that balances high reliability and high performance.
- Key advantages of InnoDB include:
  - Its DML operations follow the ACID model, with transactions featuring commit, rollback, and crash-recovery capabilities to protect user data.
  - Row-level locking and Oracle-style consistent reads increase multi-user concurrency and performance.
  - Data is arranged on disk so as to optimize queries based on primary keys.
  - To maintain data integrity, InnoDB supports FOREIGN KEY constraints. With foreign keys, inserts, updates, and deletes are checked to ensure they do not result in inconsistencies across different tables.

COMP283-Lecture 8
High Availability: Clustering

- MySQL Shell includes an AdminAPI, which enables you to create and administer an InnoDB cluster
  - Can use JavaScript or Python scripting.
- MySQL Router caches the metadata of the InnoDB cluster
  - Performs high availability routing to the MySQL Server instances which make up the cluster.
  - If the primary instance becomes unavailable, MySQL Router automatically routes client requests to a promoted secondary
- MySQL Server provides the MySQL Group Replication mechanism to allow data to be replicated from the primary to the secondaries in the cluster.

COMP283-Lecture 8
High Availability: Clustering

- Preferably use a dedicated private network
  - MS-SQL servers exchange a “heartbeat”
- Uses failover technology, with a primary server and one or more secondary servers.
- Active/Passive Failover.
- Active/Active Failover.
  - Ensure hardware can support the potential virtual services in addition to its own.
  - MS SQL Server has a Cluster Validation Tool

COMP283-Lecture 8
High Availability: Mirrored DBs

- in MS-SQL:
- Database mirroring utilises two server machines which can use differing server hardware.
- Database mirroring works at the user database level.
- Failover is quicker than in cluster failover.
- Uses failover from a principal database to a secondary database.
- Optionally use a third “witness” server.
- Three modes; High-Performance, High-Protection, and High-Availability modes.

COMP283-Lecture 8
High Availability: Mirrored DBs

- in MySQL:
- Equivalent to mirroring is Replication.
- Enables data from one MySQL database server (the master) to be copied to one or more MySQL database servers (the slaves).
- Is asynchronous by default; slaves do not need to be connected permanently to receive updates from the master.
- Advantages of replication in MySQL include:
  - Scale-out solutions - spreading the load among multiple slaves to improve performance.
  - Data security - it is possible to run backup services on the slave without corrupting the corresponding master data.
  - Analytics - live data can be created on the master, while the analysis of it can take place on the slave without affecting master's performance.
  - Long-distance data distribution - can use replication to create a local copy of data for a remote site to use, without permanent access to the master.

COMP283-Lecture 8
High Availability: DB Snapshot in MS-SQL

- A point-in-time record of a database.
- Works at the Page level.
- Can not be used to restore a corrupted or offline database.
- Snapshot is basically a read-only copy of the database.
- Useful for restoring a database quickly to the state when the snapshot was taken.

COMP283-(From Lecture 2)

DB Design: Database Snapshot files in MS-SQL

- A database snapshot provides a read-only, static view of a source database as it existed at snapshot creation, minus any uncommitted transactions.
- Database snapshots are dependent on the source database and must be on the same server instance as the database.
- If that database becomes unavailable for any reason, all of its database snapshots also become unavailable.
- Useful - Can revert a database using snapshots.
- Based on sparse files.
  - Initially an empty file allocated no space.
- Each page of the file is only "filled out" when the original data in that page of the database changes.
  - known as "Copy on Write"

COMP283-Lecture 8
Conclusions

- Introduced high-availability considerations and implementations:
  - Clustering
  - Mirroring
  - Database Snapshot