Managing Data and MapReduce – Revision Notes

- **Relational Databases (RDBMS)**
  - **Default storage & retrieval** mechanism since 1980s.
  - Efficient for **transaction processing**.
  - Examples: **System R, Ingres**.
  - Uses **SQL** for querying.
  - Replaced **hierarchical & network** databases.
  - Query processing involves:
    - **Parser**: Transforms queries into memory/disk operations.
    - **Optimizer**: Reduces execution time.
    - **Disk-space manager**: Manages storage via **pages** and pre-fetching.

- **Data Storage Techniques**
  - **Row-oriented storage**: Stores entire row contiguously.
  - **Column-oriented storage**: Stores each column separately — efficient for analytics (e.g., **Vertica**).

- **Parallel Database Architectures**
  1. **Shared Memory**
     - All CPUs share a **common memory**.
     - Managed by **SMP OS** (Symmetric Multi-Processing).
  2. **Shared Disk**
     - Servers share a **common disk** (NAS/SAN).
     - Connected via Ethernet/Fiber Channel/Infiniband.
  3. **Shared Nothing**
     - **Independent servers**, each with its own memory & disk.
     - Communicate over network.

Advantages:
- Faster query execution using multiple processors.
- **Fault tolerance** via standby systems.
- SQL optimizer handles **distributed joins** and **2-phase commit**.

Examples:
- **OLTP**: Oracle, DB2, SQL Server.
- **Warehousing**: Netezza, Teradata, Vertica.

- **Cloud File Systems**
- **Google File System (GFS)**
  - Designed for large file storage across commodity servers.
  - Supports **fault tolerance**, parallel reads/writes.
  - **Client** contacts **Master** with path + offset.
  - Master returns **meta-data**, client reads from chunk server.
- **Hadoop Distributed File System (HDFS)**
  - Open-source version of GFS.
  - Available on **Amazon EC2**.

- **BigTable (Google)**
  - Built on GFS.
  - Stores data as a **sparse, multi-dimensional map**: (row key, column key, timestamp) → value.
  - Uses **column families:label** format.
  - Allows **multiple versions** of each cell (timestamp-sorted).
  - Tables split into **tablets**, managed by **tablet servers**.
  - **Meta-data table** locates tablets, organized via **root tablet**.

- **Dynamo (Amazon)**
  - Key-Value store designed for **high write concurrency**.
  - Hashing via **MD5** maps keys to a **ring of virtual nodes**.
  - Data is replicated across **N nodes**.
  - Uses **Quorum protocol** for eventual consistency.
  - Supports **conflict resolution** via versioning and timestamps.
  - Storage engines: **Berkeley DB, MySQL**, etc.

- **Datastore (Google App Engine) & SimpleDB (Amazon)**
  - Key-Value stores with **simple transaction support**.
  - Built on top of **BigTable**, but doesn't fully exploit column-based storage.

- **MapReduce**
  - Programming model for **large-scale parallel computation**.
  - Designed by **Google**, implemented as **Hadoop** (Yahoo!).
  - Works with data stored in **GFS/BigTable**.

◇ **Key Features:**

- Supports **fault tolerance**.
- Scales across **thousands of nodes**.
- Two phases:
  - **Map**: Input → intermediate key-value pairs.
  - **Reduce**: Aggregates values by key.
- Master assigns:
  - **M mappers** and **R reducers**.
- **Reducers** fetch mapper outputs, aggregate values, write results to **GFS**.

◇ **Parallel Efficiency**

- Ideal time on p processors = T/p
- Real-world efficiency < ideal due to:
  - **Synchronization**
  - **Communication overhead**
  - **Workload imbalance**
- Efficiency e = T / (p * Tp)

Key MCQ Pointers:

Concept	**Quick Recall**
SQL	Interface to RDBMS
GFS	Fault-tolerant file system by Google
BigTable	Sparse, column-based storage built on GFS
Dynamo	Amazon's Key-Value DB using virtual nodes & versioning
HDFS	Hadoop's version of GFS
SMP	Used in shared-memory parallel DBs
MapReduce	Two-phase parallel computation model
Mapper	Converts input to intermediate key-value pairs
Reducer	Aggregates key-value pairs
Quorum Protocol	Maintains consistency in Dynamo
SSTable	BigTable's storage format per column family
Tablet	Partition of BigTable managed by tablet server
Vector Timestamp	Used in Dynamo for version control