UNIVERSITI MALAYA

FAKULTI SAINS KOMPUTER DAN TEKNOLOGI MAKLUMAT
Faculty of Computer Science and Information Technology

WQD7005 BIG DATA MANAGEMENT
OCCURRENCE 3
SEMESTER 2 SESSION 2024/2025

TUTORIAL 5

NAME:

BARKAVI A/P P CHEVEN (23093149)

LECTURER'S NAME:

TS. DR. MOHD SHAHRUL NIZAM BIN MOHD DANURI [msnizam@um.edu.my]

Question 1: Describe the important characteristics of NoSQL database that makes it more preferable in big data solution compared to SQL database.

NoSQL databases have emerged as the preferred technological foundation for contemporary big data architectures due to several distinctive characteristics that address the limitations of traditional SQL systems. The schema-less design paradigm of NoSQL databases enables dynamic accommodation of heterogeneous and evolving data structures without requiring predefined schemas, facilitating rapid adaptation to changing business requirements and data formats. This flexibility contrasts sharply with the rigid schema constraints of relational databases that necessitate complex migration procedures for structural modifications. NoSQL's horizontal scalability architecture enables seamless distribution across commodity hardware clusters through automatic sharding and data partitioning mechanisms, allowing linear performance scaling with additional nodes—a significant advantage over SQL databases' vertical scaling limitations.

The distributed architecture inherently provides superior fault tolerance through data replication across multiple nodes, ensuring high availability and disaster recovery capabilities essential for mission-critical big data applications. NoSQL databases implement specialized data models (key-value, document, column-family, and graph) optimized for specific use cases, delivering superior performance for targeted operations compared to the one-size-fits-all approach of relational systems. The eventual consistency model employed by many NoSQL implementations prioritizes availability and partition tolerance over strict consistency, enabling exceptional performance in distributed environments while providing configurable consistency levels to balance operational requirements. These systems excel in high-throughput scenarios, particularly write-intensive workloads, through optimized storage structures like log-structured merge trees and memory-mapped files. Additionally, NoSQL databases typically offer native integration with distributed processing frameworks such as Hadoop and Spark, facilitating seamless data exchange between operational and analytical systems within comprehensive big data ecosystems.

Question 2: Describe the difference between MongoDB and HBase, in terms of the storage structure.

MongoDB and HBase represent distinct architectural approaches to non-relational data storage, with fundamental differences in their underlying storage structures that significantly impact their performance characteristics and use case suitability. MongoDB implements a document-oriented storage model where data is organized into collections of JSON-like BSON (Binary JSON) documents with dynamic schemas. Each document encapsulates related data in a self-contained structure that can vary in composition within the same collection, enabling intuitive representation of complex hierarchical relationships without requiring joins. MongoDB's storage engine (WiredTiger) employs B-tree indexing for efficient document retrieval and implements document-level concurrency control with MVCC (Multi-Version Concurrency Control) to optimize read-write operations. The storage architecture includes memory-mapped files with a journal for durability, while compression algorithms reduce storage requirements.

Conversely, HBase adopts a wide-column store model inspired by Google's Bigtable, organizing data into sparsely populated distributed tables where each cell is addressed by a combination of row key, column family, column qualifier, and timestamp. This multi-dimensional mapping enables efficient storage of semi-structured data with variable attributes. HBase's storage architecture implements a log-structured merge tree (LSM) pattern with a hierarchical storage structure comprising MemStore (in-memory write buffer), HFiles (persistent immutable storage files), and Write-Ahead Logs (WAL) for durability. Data is physically distributed across the cluster through automatic sharding based on row key ranges, with each region server managing a subset of the table data. While MongoDB optimizes for developer productivity with flexible document structures and rich query capabilities, HBase prioritizes extreme scalability and consistent performance for high-throughput, append-oriented workloads across massive, distributed datasets, reflecting their distinct design philosophies and target use cases within the NoSQL ecosystem.