WQD7007 BIG DATA MANAGEMENT

OCCURRENCE 3

SEMESTER 2 SESSION 2024/2025

TUTORIAL 2

NAME:

BARKAVI A/P P CHEVEN (23093149)

LECTURER'S NAME:

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

Question 1: Explain the importance of YARN and HDFS in Hadoop architecture.

YARN (Yet Another Resource Negotiator) and HDFS (Hadoop Distributed File System) form the core foundation of Hadoop architecture, each serving critical functions that enable big data processing. YARN revolutionized Hadoop by separating resource management from processing logic, allowing multiple data processing engines to run simultaneously on the same cluster. This separation enables dynamic allocation of cluster resources, significantly improving utilization efficiency and supporting diverse workloads including batch, interactive, and real-time processing. YARN's ResourceManager acts as the ultimate authority for resource arbitration, while NodeManagers monitor individual machines and ApplicationMasters negotiate resources for specific applications. Meanwhile, HDFS provides the distributed storage layer essential for handling massive datasets. It implements a fault-tolerant system by splitting files into large blocks (typically 128MB) and replicating them across multiple nodes, ensuring data reliability even when individual nodes fail. The NameNode manages the file system metadata and coordinates access, while DataNodes store the actual data blocks. Together, these technologies create a robust ecosystem that enables scalable, fault-tolerant distributed computing and storage, making Hadoop capable of processing petabytes of data across commodity hardware clusters.

Question 2: There are 5 main pillars in Hadoop. Give one example tool for each pillar and explain the functionality of the tool.

The Hadoop ecosystem is structured around five fundamental pillars, each supported by specialized tools. For Data Management, HDFS (Hadoop Distributed File System) provides the distributed storage foundation, splitting files into blocks and replicating them across nodes to ensure fault tolerance and high availability. In the Data Access pillar, Apache Hive offers a data warehouse infrastructure that enables SQL-like querying through HiveQL, allowing analysts to interact with massive datasets using familiar syntax while abstracting the underlying MapReduce operations. For Data Governance and Integration, Apache Sqoop efficiently transfers bulk data between Hadoop and structured datastores like relational databases, supporting both import and export operations with parallel processing capabilities. In the Security pillar, Apache Ranger delivers comprehensive security framework with centralized administration for authentication, authorization, and auditing across the Hadoop ecosystem, enabling fine-grained access control policies. Finally, for Operations, Apache Ambari provides an intuitive web interface for provisioning, managing, and monitoring Hadoop clusters, simplifying administration tasks through visualization tools and REST APIs that streamline cluster maintenance and optimization.

Question 3: By taking a real-life big data example, explain how the data pipeline for the particular big data solution is designed.

Netflix's data pipeline exemplifies a sophisticated big data solution designed to handle massive volumes of diverse data. The pipeline begins with data ingestion, where multiple event streams including video viewing behaviors, UI interactions, error logs, and device information are collected through technologies like Apache Kafka and Chukwa. These raw data streams then enter the processing layer, where both batch processing (using Hadoop MapReduce for historical analysis) and real-time processing (using Apache Spark Streaming for immediate insights) occur simultaneously. The processed data moves to the storage layer, utilizing HDFS for long-term storage of large datasets and NoSQL databases like Cassandra for data requiring quick access. An analytics layer built on technologies like Elasticsearch and Druid enables complex queries and aggregations, while machine learning models implemented through frameworks like TensorFlow analyze viewing patterns to generate personalized recommendations. Finally, the presentation layer delivers insights through dashboards for internal business intelligence and personalized content recommendations for users. This multi-layered pipeline architecture enables Netflix to process petabytes of data daily, transforming raw user interactions into actionable insights that drive content creation decisions and personalized user experiences, demonstrating how carefully designed data pipelines can create significant business value from massive datasets.

Question 4: By taking a real-life example, explain how machine translation works, and compare it with human translation, in terms of efficiency and accuracy.

Machine translation, exemplified by Google Translate, works through a sophisticated neural machine translation (NMT) process that has evolved from earlier statistical methods. The system processes text through several stages: first extracting sentences, then parsing words into grammatical components, analyzing the sentence structure, and finally translating and reassembling content according to the target language's grammatical rules. Modern NMT systems use encoder-decoder architectures with attention mechanisms to capture contextual relationships between words, enabling more natural translations. When comparing with human translation, machine translation excels in efficiency, processing millions of words per second at minimal cost, making it ideal for high-volume, time-sensitive content like news articles or social media posts. However, human translators maintain superior accuracy, especially for nuanced content requiring cultural context, idiomatic expressions, or specialized terminology. For example, when translating legal documents or literary works, human translators can preserve subtle meanings and cultural nuances that machines often miss. The accuracy gap narrows for technical documents with standardized terminology and straightforward sentence structures. In practice, the most effective approach often combines both methods—using machine translation for initial drafts and high-volume content, with human translators reviewing, editing, and handling sensitive or complex materials, thus leveraging the efficiency of machines while maintaining the quality assurance that only human linguistic and cultural expertise can provide.

Question 5: Discuss the advantages and disadvantages to use autocoding method in big data resources.

Autocoding methods in big data resources offer significant advantages, primarily through their ability to process massive volumes of unstructured text at speeds impossible for human coders. This efficiency enables the analysis of entire document collections rather than samples, potentially revealing patterns that might otherwise remain hidden. Autocoding also ensures consistency by applying the same rules uniformly across all documents, eliminating the inter-coder reliability issues that plague manual coding efforts. Cost-effectiveness represents another major advantage, as automated systems can process millions of documents at a fraction of the cost of human coding teams. However, autocoding also presents several disadvantages. Accuracy limitations remain significant, particularly for ambiguous terms, contextual meanings, and domain-specific jargon that require nuanced understanding. Most systems struggle with semantic complexity, including sarcasm, metaphors, and cultural references that humans interpret intuitively. Domain specificity presents another challenge, as autocoding systems often require extensive customization for different subject areas, necessitating specialized vocabularies and rule sets. Additionally, these systems demand substantial computational resources for large datasets and require regular maintenance to update vocabularies and rules as language evolves. The optimal approach typically combines automated coding for initial processing of large volumes with human oversight for quality control, review of edge cases, and continuous system improvement, leveraging the strengths of both methods while mitigating their respective limitations.

Question 6: Given a big data resource that contains text about cognitive science, explain how term extraction is carried out to create index automatically for the resource.

Term extraction for cognitive science texts involves a sophisticated multi-stage process that combines linguistic analysis with statistical methods to identify domain-relevant terminology. The process begins with text preprocessing, where documents are tokenized into individual words, normalized to standard forms, and filtered to remove common stop words that carry little semantic value. Next, linguistic processing applies techniques such as part-of-speech tagging to identify potential terms (primarily nouns and noun phrases), lemmatization to reduce words to their base forms, and named entity recognition to identify specialized terminology. Statistical analysis then evaluates candidate terms using metrics like term frequency and TF-IDF (Term Frequency-Inverse Document Frequency) to measure importance, while co-occurrence analysis identifies terms that frequently appear together, suggesting conceptual relationships. Domain-specific processing applies cognitive science ontologies to recognize field-relevant terminology, identifies multi-word terms common in the discipline (such as "working memory" or "neural correlates"), and maps synonymous terms to canonical forms. The system then validates and ranks extracted terms based on statistical significance and relevance to cognitive science, clustering related terms into conceptual groups. Finally, the index generation phase establishes hierarchical relationships between terms, creates cross-references between related concepts, and produces a searchable structure with term definitions and document references. This automated approach enables the creation of comprehensive, navigable indexes for large cognitive science text collections, facilitating efficient information retrieval and knowledge discovery while maintaining the semantic richness necessary for specialized academic domains.