IE494 BIG DATA PROCESSING
Project Proposal

A Deep Dive into "Tenzing: A SQL Implementation on the MapReduce Framework"

Problem Area:

This project will focus on a thorough study and analysis of the research paper "Tenzing: A SQL Implementation on the MapReduce Framework." The goal is to explore the technical concepts, architecture, and design decisions that led to the development of Tenzing, a high-performance, scalable SQL query engine built on top of Google's MapReduce framework. The project will involve an in-depth examination of how Tenzing achieves low-latency querying, scalability to petabyte-scale data, and SQL compatibility, while leveraging distributed computing and MapReduce optimizations. Understanding the integration of SQL features with the MapReduce execution model. Exploring Tenzing's support for various storage formats and optimizations like indexing and execution techniques.

Group Members:

- Shravan Kakadiya - 202201333

Expected Outcomes:

Expected Outcomes for Phase 1:

1. **Introduction and Motivation**:
    - Understand the motivation behind Tenzing's creation, including the challenges Google faced with existing database systems like DBMS-X, and why they transitioned to MapReduce.

    - Provide a summary of the key problems Tenzing aimed to solve, such as scalability, cost-efficiency, and performance bottlenecks.
2. **Preliminary Architecture Understanding**:
    - Outline the high-level architecture of Tenzing, emphasizing how SQL queries are executed on top of the MapReduce framework.
    - Describe the types of data sources Tenzing supports, such as row stores, column stores, and Bigtable, and the methods for handling different formats.
3. **SQL Implementation in Tenzing**:
    - Initial exploration of Tenzing's SQL implementation, particularly how it integrates SQL92 and SQL99 standards while extending the framework for more advanced use cases like user-defined functions.

By the end of Phase 1, I will have established a solid understanding of the core issues that led to the development of Tenzing and the initial design and capabilities of its SQL engine on MapReduce.

Expected Outcomes for Phase 2:

1. **In-depth Performance Analysis**:
    - Conduct a detailed analysis of Tenzing's performance optimizations, including query execution plans, indexing strategies, and how the system achieves low-latency performance.
    - Compare Tenzing's performance with other SQL-on-MapReduce systems (such as Hive and Pig), highlighting the advantages in terms of scalability and efficiency.
2. **Scalability and Reliability**:
    - Explore how Tenzing achieves massive scalability (petabyte-scale data, thousands of cores) while ensuring system reliability, especially on commodity hardware.
    - Evaluate Tenzing's fault tolerance mechanisms and how it handles hardware failures effectively.
3. **SQL Extensions and Advanced Features**:
    - Provide a comprehensive understanding of the advanced SQL features supported by Tenzing, including complex user-defined functions, nested relational data,OLAP, Views, DDL, DML, Nested

Queries and Sub-Queries, Hash Joins, Aggregations and support for multiple storage formats (Bigtable, GFS, protocol buffers).
    - Analyze how Tenzing enhances SQL for large-scale distributed environments without sacrificing performance.

By the end of Phase 2, I will have a thorough understanding of how Tenzing implements SQL on MapReduce, its system optimizations, scalability features, and its impact on distributed data analysis.

Selected Readings:

- Chattopadhyay, Biswapesh, et al. "Tenzing a sql implementation on the MapReduce framework." Proceedings of the VLDB Endowment 4.12 (2011): 1318-1327.