DBMS Lab

Project Ideas

Guidelines

- Project groups will be the same as that of the mini-project groups.
- Final report submission and demonstration will be done by April 15th, 2024. There will be intermediate evaluations.
- Project will be evaluated based on the volume of work, and completeness.
- Some broad project ideas are given below. You are free to choose other ideas too. Multiple groups may take up the same project idea.
- You have to submit a concrete project proposal within a week and get it approved by us.

Deliverables

- Report on the project and a demonstration. The report should be about 10 pages long.
- The content would be:

1. Title and team members
2. Objective
3. Methodology
4. Results/Screenshots
5. References

Project Ideas: Specialized Databases

- Spatial Databases - Disaster Management:
- Design and populate a database of road networks, population, water level etc is provided from multiple data sources. The goal of the project is to provide an information system for decision support in disaster management tasks like evacuation, relief. It is part of a larger project for National Spatial Data Infrastructure specific to coastal disaster management system in Eastern coast of India. The database should be supported by a web interface. Data may be pulled from the Google Earth Engine. Similar spatial database project ideas are listed in: https://sites.google.com/view/summerofearthengine/projects

Text Search Engine

- Objective: The project will use the Apache Solr framework running on ZooKeeper framework to develop scalable text search engines.
- Methodology: Develop keyword based search engine. Do some benchmarking, to see how search performance scales in a very high query per second (QPS) setting. QPS can be obtained using either something like a multithreaded HTTP client using a ThreadPool and ConnectionPool, or by using a tool like Siege, that can simulate multiple concurrent HTTP requests on a server.
- Solr is an open source enterprise search platform from the Apache Lucene project. Its major features include full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document (e.g., Word, PDF) handling. Providing distributed search and index replication, Solr is highly scalable.
- Siege is an http load testing and benchmarking utility. It was designed to let web developers measure their code under duress, to see how it will stand up to load on the internet. Siege supports basic authentication, cookies, HTTP and HTTPS protocols. It lets its user hit a web server with a configurable number of simulated web browsers. Those browsers place the server “under siege.”
- Outcome: QPS data outcome using multithreaded http clients with the help of seige connection pool and thread pool.

Graph Databases

- The goal of the project is process large graphs in a database

i. Install any graph processing systems e.g., ApacheGraph, Pregel (GoldenOrb), Giraph, Neo4j, or Stanford GPS,

ii. Load a large graph from Stanford SNAP large graph repository

iii. Provide interface to run simple graph queries. Bonus for computing PageRank.

iv. Profile performance

Multi-media Databases

- Multimedia databases store and query music, video, image data
- They are common in various applications
- Searching over multimedia data often involves high dimensional indices like KD-trees
- Deliverable: Build a multimedia database consisting of text/structured data/music/images/video. Public data may be downloaded to populate the database. Basic queries should be supported.

Temporal Databases

- The goal of the project is to build a platform for quick query and processing of time series financial data. Checkout timescale for inspiration:
- https://docs.timescale.com/tutorials/latest/financial-tick-data/

Query Languages

- Datalog
- Datalog is a query language based on the relational calculus. Write datalog programs for a graph database to answer various graph queries.

- QBE: Query By Example
- Microsoft Access is a QBE platform. The goal of this project is to develop a QBE platform for a database application

Semi-structured Query Languages

- XML provides a framework for handling semi-structure data
- DBLP/Wikipedia are examples of large collections of XML data. They may be downloaded freely
- Deliverable: Build a XQuery interface for the Wikipedia/DBLP XML data. The interface should support structured as well as unstructured queries.

NoSQL

- The data models used by NoSQL databases are usually based on key/value pairs, document stores, or networks.
- Implementations include: MongoDB, Cassandra, Hbase
- A noSQL database is designed to support
  - Very large collections of data
  - High throughput with data from streams
  - support tree or graph models for its data
  - support heterogenious collections of data
- Deliverable: A NoSQL database for an application like BTP project report management system

Design a MLops and HITL database

- Based on VertaAI or Apache ModelDB to manage ML experiments, models and metadata, extend the same or design a new database to store and incorporate human evaluations such as labeling and reviews as part of the ML pipeline.

Database Connectivity

Client-side database

Build a Javascript library that client-side Web applications can use to access a database;
This layer should cache objects on the client side whenever possible, but be backed by a server-side database system.

As a related project, HTML5 browsers (including WebKit, used by Safari and Chrome),
include a client-side SQL API in JavaScript. This project would involve investigating how to user
such a database to improve client performance, offload work from the server, etc.

Mobile Databases

- **Data Management for Ad Hoc Mobile Workgroups** - Groups of people can form temporary electronic collaboration groups using their smartphones. Although group interaction often involves working on a shared document or database, modern smartphones give very little support for local group management, and almost none for the data products that groups create. It should be possible for physically-nearby groups to create temporary data storage areas with some standard features: reliable storage, versioning, preserved user identity, report generation, query processing of the shared data, and so on.

Sensor Network Datalogging

- IITKgp Bus Tracker
- Design an information system to track the buses within IITKgp campus. On board GPS systems output can be obtained from the smartphone of a student onboard.
- Goal is to build a smartphone interface to get updates about nearest bus, expected arrival time etc on a map to the students.

Web Mashups

- In this project you can connect a number of web services provided as APIs like google/bing map, instagram, twitter and combine them to provide interesting applications. Exact mashups may be decided by you. Examples include, geotwitter, integrating google map with twitter etc.

Database Security

- Preventing denial-of-service attacks on database systems. Databases are a vulnerable point in many Web sites, because it is often possible for attackers to make some simple request that causes the Web site to issue queries asking the database to do a lot of work. By issuing a large number of such requests, and attacker can effectively issue a denial of service attack against the Web site by disabling the database. The goal of this project would be to develop a set of techniques to counter this problem — for example, one approach might be to modify the database scheduler so that it doesn’t run the same expensive queries over and over.
- SQL injection queries may also be studied

Buffer Manager

- The of this project is to simulate a small buffer pool for simple Join/Selection queries on few small tables in the C/C++ language. Popular buffer manager strategies like LRU/MRU/CLOCK/Pinned blocks may be simulated. The strategies may be compared in terms of the number of disk i/o required.
- You may use the SQLite C Library for more realistic simulation. https://sqlite.org/index.html

Performance Analysis of DBMS

- Many modern architectures and OS are designed for database workloads. Designing systems for large distributed database workloads is an interesting problem.
- Performance monitoring is an important portion of data-center and database management. An interesting project consists in developing a monitoring interface for Postgresql, capable of monitoring multiple nodes, reporting both DBMS internal statistics, and OS-level statistics (CPU, RAM, Disk), potentially automating the detection of saturation of resources. You may use some benchmark like TPC etc

Indexing and Hashing

- High Dimensional Indexing

Implement the R-Tree/KD-Tree for high dimensional indexing. You may choose a high dimensional data (say, audio or image) and use the tree to index and search it.

- Implementing Extendable Hashing

Implement the extendable hashing algorithm. Implement the data structures for the hash table. Assume only data expansion occurs. Benchmark the access time on a suitable dataset. You may compare it with the SQLite implementation.

Indexing

- Auto-admin for Index Creation

Given a set of query workloads and some table statistics along with a index storage budget write a program to decide the best indices to create. Auto-admin tools are often available to recommend indices, etc. Design a tool in SQLite that recommends a set of indices to build given a particular workload and a set of statistics in a database.

Reference: https://www.cs.toronto.edu/~consens/tab/

- Implementing learned indexes

Train neural network models in place of traditional indexes structures such as B+Tree, implement a machine learning model index and compare it's performance to B+Tree.

Query Processing

- External Memory Join Algorithms

Implement external memory join computation algorithms and profile their performance on large data sets.

- Metric Reporting

Build a wrapper/interface which collect query processing metrics like table statistics, CPU/memory usage in run time while executing a query. You may use the inbuilt commands of Postgres.

- Rule based query rewriting

Specify some fixed rules for query optimization. Write a query rewriter for simple queries which takes as input a relational algebra query and returns its optimal version.

Simple Query Processor

- Write a query executor for in-memory data

– Parse simple SQL DML commands and generate execution plans to evaluate

– Implement plan nodes for sequential scans, sorting, grouping/aggregation, joins

– Use a heuristic-driven plan optimizer

– Experiment with different strategies, measure performance

Simple Query Optimizer

• Write a cost-based query planner/optimizer

– Take simple, unoptimized execution plans as input

• Will need an appropriate representation of query plan nodes

– Output optimized execution plans, along with associated cost measure

– Need to properly cost different plan nodes

Use (faked) table statistics to choose optimal plans

Take CPU, memory, disk requirements into account

– Integrate a mechanism for limiting search effort of optimizer

Distributed Databases

- The goal of the project is to design a large database running on a distributed map-reduce platform that can handle heterogeneous data obtained from different sources. The map-reduce distributed Hadoop platform will be used. The database will use Apache Hbase system as the table structure. It will integrate multiple data sources using the Protocol Buffer architecture. Such databases are common in processing of large and unstructured data common in search engines and online social networks.

1. Install Hadoop/Hbase on a laptop/server cluster
2. Load data to nodes
3. Write map-reduce operations
4. Pipe map-reduce outputs using Protocol Buffer
5. Run simple queries

- Technologies Involved: Hadoop, Apache Hbase, Protocol Buffers
- Data APIs (each group may select a separate data and appropriate queries):
    - Twitter, Amazon public data sets (http://aws.amazon.com/datasets)

Text Search in MapReduce

- The project will use the Apache Solr framework running on ZooKeeper framework in a MapReduce architecture.
- In this project you will develop a keyword based search engine. Then do some benchmarking, to see how search performance scales in a very high query per second (QPS) setting. QPS can be obtained using either something like a multithreaded HTTP client using a ThreadPool and ConnectionPool, or by using a tool like Siege, that can simulate multiple concurrent HTTP requests on a server. Solr is an open source enterprise search platform from the Apache Lucene project. Siege supports basic authentication, cookies, HTTP and HTTPS protocols. It lets its user hit a web server with a configurable number of simulated web browsers.
- Outcome is a high QPS data search benchmarking results using multithreaded http clients with the help of seige connection pool and thread pool.

Data Warehouses

- The goal of the project is to run the Hive data warehouse system on Hadoop. And run aggregate/ reporting queries on large data sets in a map-reduce framework.
- Steps:
- 1. Install Hive on Hadoop running on a laptop/server cluster
- 2. Load data to Hive
- 3. Write and execute queries in HiveQL

Databases and ML

- Modern databases often have integrated machine learning functionalities. This includes LLM support. The goal of this project is to build a DBMS+ML application for a suitable problem.
- The project involves integrating SQL queries with machine learning on databases. The postgresML platform may be explored.

LLM SQL

- The goal of this project is to build a LLM for generating SQL queries from natural language questions. User asks questions in a natural language and the system generates answers by converting those questions to an SQL query and then executing that query on PostgreSQL database. Any open source LLM may ne used. Example: Create a database on CDC information about placements. A student may ask questions such as,
- How many ML companies will visit campus in March?
- What were the programming skills of students who got offers exceeding 20LPA?

Project Proposals Due in a Week

- Group details
- Title
- Abstract
- Weekly Work Plan

Thank you!