**DATABASE SYSTEMS NOTES**

**Week 8: 10-10-2023**

**Performance Tuning**

Indexing the columns

Create the index tables

**Indexing algorithm** - Hierarchical Navigable small words graph, quadtree indexing, octree indexing.

**Fine tuning -** avoid the weakest links. Optimize all the used material/resources.

**Client side fine tuning** - SQL performance tuning - football player and cs student

1. Select the right column first
2. Don’t mix the literals and calculations use something like **price > 50/1.1 ,** but not **1.1\*price > 50**

**Server side tuning** - DBMS performance tuning

Backend tuning - Binary search

Raw table scan is not performance tuning

**WEEK 9 : 10-17-2023**

**Access plan -** complied version of SQL - stored in SQL cache

**Data Cache** - where queries are executed

Easiest hash used - MD5  
we can treat the access plan as 2 column table. 1st column is hash of SQL data. 2nd column is compiled version of the SQL data.

Tables are kept in groups called as Table spaces.

Optimizer - Analyses the SQL queries and finds the most efficient way to access the data. Uses machine learning models/algorithms.

**Algorithms are based on**

1. Fastest execution time
2. Less communication costs.

**Evaluated based on**

1. Operation mode - automatic, manual
2. Timing of its optimization - static(best for embedded systems, takes place at compile time) , dynamic(MORE process over head)
3. Type of information -

symbolic reasoning(**statistics** based query optimization - 1. Dynamic - auto eval. 2.Manual - user ),

machine learning(**rule** based query optimization)

**Commands to generate the statistics:**

COMPUTE STATISTICS (Oracle)

ANALYZE TABLE(MySQL)

UPDATE STATISTICS(SQL Server)

**Query processing:**

1. Parsing
2. Execution
3. Fetching

**Query processing bottlenecks caused by:**

1. CPU - slow processor
2. RAM - shared among running process
3. HARD DISK - disk speed, transfer rates
4. NETWORK - bandwidth
5. Application Code - bad user code, poor db design

When there is low sparsity - indexing is useless.

**Data sparsity:** different value a column could have

**Data structures used to implement the indexes:**

1. **Hash indexes** - indexing based on states
2. **B-tree indexes** - indexing based on last name
3. **Bitmap indexes** - indexing based on location (NE, NW, SE, SW) {N-North S,W,E - other directions - treated as mutually exclusive} --- stored as bytes -- no of rows in original table = number of indexes in bitmap index table.

**Index Selectivity:**

Measure of likelihood that an index will be used in query processing

**Expression based index -**based on the specific SQL expression (EMP\_SALARY + EMP\_COMMISSION)

**Indexes are useful when:**

1. Indexable columns occurs in WHERE or HAVING search expressions
2. Indexable columns occurs in GROUP BY or ORDER BY clause
3. MAX or MIN is applied to an indexable column
4. High data sparsity on the indexable column

**Optimizer choices:**

1. Rule based optimizer - chooses a plan that minimizes the sum of rules.
2. Cost based optimizer - processing cost, Ram cost

**Hints used for the optimizer:**

1. ALL\_ROWS - /\*+ ALL\_ROWS\*/
2. FIRST\_ROWS - /\*+ FIRST\_ROWS\*/
3. INDEX(name) - /\*+ INDEX(P\_QON\_NDX)\*/

Here '+' indicates the compiler hint.

Without '+' sign, it is just a comment.

Equality conditions are faster than inequality conditions.

Transform conditional expressions to use literals.

Avoid the use of NOT logical operator.

**In memory database -** Redis (used for fintech, stock trading) - stores large memory of database in primary storage.

**RAID -** Redundant Array of Independent Disks

**CUBRID** relational databases used for OLTP(Online Transaction Processing)

**Business Intelligence:**

Used for converting data into models.

Formal definition: comprehensive, cohesive, integrated set of tools and processess that captures, collects, integrates, stores and analyse the data.

It is also same as Data Mining.

**ETL** - Extract , Transform, Load.

Streaming BI - Real time business Intelligence.

Datawarehouse/ OLTP/ Decision Support data - These all mean the same

**Operational data:**

1. Day to day data. --- Point of Sale Terminal
2. Mostly used for writing and rarely used for reading.

**Decision support data:**

1. Historical data
2. Only used for reading.

**Decision support data - differs from operational data in:**

1. Time span - 10, 20 ,30 years
2. Granularity - Drill down, Rollup - categorise peanuts into healthy snacks
3. Dimensionality

**Schema on read**: - Data(structured or unstructured) is stored first and then Schema is created from the data.

**Schema on write**: - Schema is created first and then data(structured) is populated. ETL(Extract , Transform, Load) is a classic "schema on write" process.

Data Marts are the subsets of Data Warehouse

**WEEK 10: 10-24-2023 - Part -1:**

Online gradient descent

DuckDB

**Business Intelligence** -

works on the existence data

Star schema, snowflake schema

In **star schema**, the centre of the dimensions is called Fact table. Each column corresponds to a Primary key in the other table. Denormalized.

**Fact table**(has the operational data) changes every second.

**Snowflake schema** - Breakdown of tables from star schema and link each other.Normalized.

(Operational data + External data) --- ETL ---> (Datawarehouse + DataMart) ---Query and Reporting -----> Data analytics -----> Data Visualization

**Advantages of Business Intelligence:**

1. Improved decision making
2. Integrating architecture
3. Common user interface
4. Common data repository
5. Improved Organizational Performance

**OLAP -** Online Analytical Processing - also called BI Data Mining

**History:**

1970 -- Centralize reporting

1980 -- spread sheets

1990 -- Enterprise reporting/ OLAP

2000 -- Dashboards/ Mobile BI

**SVM** - Support Vector Machines --- take data and split into 2

**4 different ways to organize a data warehouse:**

1. Star Schema
2. Snowflake Schema
3. Fact table can be supplanted
4. Separate fact table for each attribute

**Classification of tools:**

1. Exploratory analytics - to analyse about the past
2. Predictive analytics - to predict about the future

**ROLAP -** Relational OLAP -- uses relational database management systems.

**MOLAP -** Multidimensional OLAP -- uses Multi-dimensional database management systems -- based on the raw access of pointers

**GROUP BY MODIFIERS:**

1. **ROLLUP** - does not include the last column -- {GROUP BY ROLLUP (column1, column2)} -- all columns except the last column subtotals are returned
2. **CUBE** - includes the last column -- -- {GROUP BY CUBE (column1, column2)} -- all the columns subtotal are returned

**Data lakes --** (Modern alternative to traditional data warehouse)

Traditional data warehouse is ETL based. Traditional data warehouse is schema on write.

Data lake is not ETL based. Data Lake is schema on read. Raw, untransformed data.

**Lakehouse** = combination of ETL + DATA LAKE

**SPATIAL DATABASES:**

(Longitudes, latitudes)

Extent -- lines, polygons

JWST imagery -- astronomical spatial data

**Geographic data characteristics:**

1. Location
2. size
3. Spatial auto correlation - availability of same types of items found when value is '1'. Eg: Chat GPT, flower shops, gold shops in a single area.
4. Scale dependant -- Google maps zoom in and zoom out
5. Temporal dependant -- time dependant -- traffic

Entity view --- area filled with discrete objects.

Field view -- area filled with continuous objects

**Spatial data systems consists of: (DOI)**

1. Datatypes
2. Operators
3. Indices - Quadtree

**Examples of spatial data:**

1. CAD data
2. Agricultural data
3. Crime data
4. Pandemic data
5. Drug overdose
6. Income distribution
7. Real time traffic

**Who creates/uses spatial data?**

1. Army field commander
2. Insurance risk manager
3. Medical doctor
4. Molecular biologist
5. Astronomer

Maxar technologies rafah crossing uses the spatial data.

**Different ways to create the spatial data:**

1. CAD - user creation
2. CAD - reverse engineering
3. Maps - cartography(surveying and plotting)
4. Maps - satellite imagery
5. Maps - drone imagery
6. Maps - driving around
7. Maps - walking around

**Spatial data can be stored as:**

1. Points
2. Polylines
3. Polygons
4. Pixels/raster

**GIS -** Geographical information system -- specific application architecture built on top of SDBMS(more general purpose).

**Uses of GIS:**

1. Location analysis - buffer, corridor
2. Terrain analysis - slope, drainage
3. Flow analysis - shortest path
4. Distribution - change detection, proxiity
5. Spatial analysis - pattern, auto correlation
6. Measurements - Distance, perimeter, direction, shape
7. Search - search by region

MBR - minimum bounding rectangles -- smallest rectangle used to enclose the data

**Spatial relations :**

1. Topology based - boundary, interior, exterior -- grouped by - proximity, overlap, containment
2. Metric based - distance, angle measures
3. Direction based
4. Network based - shortest path

Eaxmples -- county

**Spatial query example:**

SELECT c1.name

FROM County c1, County c2

WHERE Touch(c1.shape,c2.shape) = 1

AND c2.name = 'Contra Costa'

**SPATIAL FUNCTIONS:**

1. Union
2. Difference
3. Intersect
4. XOR
5. Buffer
6. CenterPoint
7. Convex Hull

**Spatial operators:**

1. Equlas
2. Disjoint
3. Intersects
4. Touches
5. Crosses
6. Within
7. Contains

**WEEK 11: 10-31-2023**

**Spirograph** --- used for displaying led lights in the sky instead of fireworks

QR code display on the sky.

Triple store is a knowledge graph

SDO -- Spatial data Object

Oracle spatial library 10g -- spatial database

SDO\_GEOMETRY -- object type in oracle.

**SDO\_GTYPE -** Defines type of geometry stored in the object.

1. POINT
2. LINESTRING
3. POLYGON
4. HETEROGENEOUS COLLECTION
5. MULTIPOINT
6. MULTILINESTRING
7. MULTIPOLYGON

2D DATA -- 2001,2002….

3D DATA -- 3001,3002,…

4D DATA -- 4001, 4002…

**Operators in oracle:**

1. SDO\_FILTER
2. SDO\_RELATE
3. SDO\_WITHIN\_DISTANCE
4. SDO\_NN

**Indexing in spatial dbs:**

Aabb -- axis aligned bounding box -- normal bounding box

Oriented bunding box - in video games -- axis are tilted

Scrolling in the google maps -- spatial indexing

**R-Tree(Region Tree) Indexing** is mostly used . Others -- (KD Tree, KD btree, Quad Tree)

MBR(Minimum Bounding Rectangle) - for 2D data

MBV(Minimum Bounding Volume) - for 3D data

Foundation DB of Apple uses - Z Fractal Curve

Video games uses KD -TREE.

**KD TREE** -- alternatively split the data vertically and horizontally passing through the points . Used for mapping hospitals, gas stations, hotels etc,.

**KDB - TREE: --** alternatively split the data vertically and horizontally without passing through the points

In KDB tree , data is available only at the leaves, whereas in KD tree the data is available at leaf nodes and also non leaf nodes.

**Quadtree: --** recursively and adaptively divide the region into 4 sub regions till it is dividable.(used for 2d data)

**Octree --** do the same as quad tree, but divide into 8 cubes (used for the 3d data.

**Warnock's algorithm --** runs in the almost every printer in the world.

**Filter and refine** is the common strategy used by efficient algorithms to answer the spatial queries.

1. Filter step -- Take the region overlapping with the Minimum Bounding Rectangle(MBR).
2. Refine step - Consider the actual overlapping region.

Geocoded databases - transforms corresponding address into latitude and longitude

**3d gaussian splatting** -- NERF rendering of geo data

Walking through the streets.

**Spatial data visualizations can be done by**

1. Dot map
2. Proportional symbol map
3. Diagram map
4. Choropleth map

**KML --** key hole mark-up language

Openlayer

Map box

Map tiles

**NO SQL:--------------------- (key, value) pair**

**What is the need of nosql**

1. Support multi valued attributes
2. No need to worry about the Null values for other attributes
3. Object inside array, array inside object and can be continued infinitely.

Douglas Crockford -- JSON

**Why to use NoSQL:**

1. Flexibility
2. Efficient
3. Available
4. Scalable
5. High performance
6. Handles rapidly generated data
7. No need of object relation mapping overhead
8. Data fragmentation is easy
9. Parallel processing can be done by fragmentation

NoSQL stands for Not only SQL.

Json crack --- viewer for json

JSON-P --- Json padding -- server calling the client side java script function

JSON-LD --- Json Linked Data -- describing the structure data in specific format - food recipes.

**json data types:**

1. Number
2. String
3. Boolean
4. Array
5. Object
6. Null

**WEEK - 12: 11-07-2023**

Before Key-value databases, in olden days, we have **Memcached** with 'del', 'put' and 'get' operations.

**4TYPES OF NoSQL databases:**

1. **Key value database** - Dynamo DB, Project Voldemort, Redis
2. **Column oriented database** - Cassandra(Apache), Hbase(Apache), BigTable(Google), BigQuery(Google), SimpleDB(Amazon), RedShift(Amazon)
3. **Document database** -- MongoDB, CouchDB(Cluster Of Unreliable Commodity Hardware),
4. **Graph database** -- Neo4j, HyperGraphDB, InfineGraph, InfoGrid, NetMeshBase, FlockDB(Twitter), Giraph(Apache), GraphLab

**Additional databases:**

Time series databases - data varies according to time

Vector databases -- pinecone

**Polyglot persistence (Martin Fowler)--** use of different storage technologies to store different parts of single application.

1. User sessions - Redis DB
2. Financial Data - RDBMS
3. Shopping cart - Riak DB
4. Recommendations - Neo4j
5. Product catalog - MongoDB
6. Reporting - RDBMS
7. Analytics - Cassandra
8. User activity Logs - Cassandra

**Code refactoring --** changing the body of function/code to make it optimized. Don’t

change the apis.

Log structured merge tree -- is used for rapid querying. Deletion happens once in a while. Because it affects the indexing. Deleting occurs when system is idle.

SSS - Simple Storage Service. Everything runs on cloud. So querying can be done from the mobile itself.

Elastic cache -- contained can be assigned based on demand/requirement.

Redis DB is advanced that Dynamo DB.

Column family has group of columns. Column family can be named as Super column.

Group of super columns is called super column family. It is used in column databases.

**Inverted indexing:**

Cassandra uses the SSTables.

**Document DB querying:**

1. Simple k/v queries
2. Range queries
3. Geo spatial queries
4. Text queries
5. Aggregation
6. Map reduce queries

**Graph DB:**

It has index free adjacency.

It is linked-data.

It is used for recommendation engines, social networks.

Data can be stored on edges and nodes.

**Querying in graph DB:**

Cipher is the language for the LLMs supported by Neo4j**.**

Tinkerpop is a graph traversal language used to train LLMs.

Fine tuning

Xml types:

1. Graph Ml
2. XML

Triplet - 2 nodes(subject, object) and 1 edge(predicate).

Triplets are also stored in XML format as of now.

Subject -- what we are describing

Predicate -- a property of the subject

Object -- the value of predicate

Semantic searches ae done in GraphDB.

**TRIPLE store database implementations:**

1. AllegroGraph
2. MarkLogic
3. SparkleDB
4. Stardog

SPARQL, RQL, SeRQL -- Query languages.

The output of triple store query is called a graph.

Triple store DBs is equivalent to a form of RDF(Resource Description Framework) databases.

**Triple store DB architecture:**

1. In memory -- main memory
2. Native store -- storage provided by the DB vendors - AllegroGraph
3. Non native store -- 3rd party services -- Jena SDB uses MySQL.

Map reduce is actually Map-Combine-Shuffle-Reduce. EX: Google Search

Example: Maya DAG

**WEEK 13: 11-15-2023**

SIMD -- Single Instruction Multiple Data

Map reduce is introduced by google.

The input and outputs are of the key/value pair format

Shuffler consolidates and sends the data to the reducer.

Reducer can add the results sent by the shuffler.

Gfs -- Google File System

**Apache Hadoop Stack**

1. HDFS - Hadoop Distributed File System
2. Hadoop MapReduce - Distributed Processing Framework
3. Hbase - Distributed Table Store
4. ZooKeeper - Coordination
5. Oozie - Workflow
6. Hive - Data Warehouse
7. Pig - Data Flow
8. R - Statistics
9. Mahout - Machine Learning
10. Flume/ Chukwa - Lo Data Collector
11. Scoop - Relational Database Data Collector
12. Ambarl - Provisioning, Managing and Managing Hadoop Clusters.

Hive - Hive provide SQL like scripting Language called HQL.

Hive translates queries to map reduce jobs.

**Muskteer:**

Musketeer is an experimental approach to do the decoupling. The 3 benefits are:

1. Users write workflow once and can execute on alternative systems.
2. Multiple subcomponents of a workflow can be executed on a different backend systems
3. Existing workflows can easily be ported to new systems.

**Earlier:**

Hive, Giraph --> Hadoop

Froendends are coupled with backend systems.

**Present:**

Frontends are decoupled from the backend systems.

Intermediate system s present.

**Front-ends :** Hive,Pig, DryadLINQ, GAS DSL, GreenMarl, SQL DSL, SparkSQL, GraphX, Lindi, GraphLINQ

**Back-ends:** Hadoop, Metis, Dryad, Spark, Naiad, CIEL, Giraph, PowerGraph, GraphChi, X-Stream

**YARN -** Yet Another Resource Negotiator.

First version of MP/HADOOP was batch oriented.

Talend bi data platform

Sparks has driver and executors -- real time processing is possible in spark.

**Flink:**

Apache's Flink is based on the concepts of Map Reduce, but it generalizes in many ways.

Additional transformations for Flink are:

1. Join
2. CoGroup
3. Filter
4. Iterations

Strom:

Kafka:

Samza: -- developed at LinkedIn

Handles batch data and stream data in unified manner.

Spark, Flink and Smaza can handle batch and stream data.

Storm does stream processing.

Hadoop+ Yarn is for batch processing.

Spark,Flink and Samza for batch+stream processing.

Kafka is for handling messages.

**Stackshare.io --** important websites

**BSP --** Bulk Synchronous Parallel model is an alternative to MR.

It is mainly used for graph processing.

Giraph -- is mainly for graph algorithms

and Hama -- can do graph processing and High Performance Computing

**WEEK - 14: 11-23-2023**

**Machine Learning algorithms for Beginners:**

1. Linear Regression
2. Logistic Regression
3. CART algorithm
4. Naïve Bayes
5. KNN Algorithm
6. Apriori
7. K-Means
8. PCA
9. Random Forest Classification
10. AdaBoost

**Types of data mining algorithms:**

1. Classification - Labelling
2. Clustering - Grouping
3. Association - Relating
4. Regression - Coupling

Chat GPT is an example of unsupervised learning.