31st Jan 2022:

The process of external sort is quite useful, and it has several trade-offs in the areas of interaction and the same can be dealt using the External hash Algorithms. To implement the same, we need to ask several questions as hash involves several methods, of them which is suitable and what kind of hashing is preferred.

Hashing deals the large data by splitting them into bunches with the help of a Hash Function, which can fit in memory space. It runs by processing the data bunch by bunch based on the necessary operations which are to be done. Example: Binary Operation like Union, Join requires two bunches are processed accordingly. Also, Leveraging the In-Memory processing Algorithms helps in processing the bunches efficiently and effectively. The next question is to how many parts? As the bunches are stored in Memory, we need at most M parts of memory. This can be accomplished using two phases:

Phase1, it helps us decide the number of buckets needed for the external hash function, start by initializing the disk blocks(k) and add the tuples to them based on the hash function. When the bucket is full, we try to flush the tuples in the respective bundle and reinitialize and flush out the empty blocks such that we have disk that are small enough & can easily fit into memory. Phase 2, all the bundles in the memory are loaded for further processing.

Working of Unary operators; it is simple that it only requires a single bundle, and we use in-memory hash algorithm bundle at a time and is continued for all of it, Working of Binary Operators; we go over every bundle and initialize the in-memory hash and a serial scan is performed on the other relation and insert the has value. Finally, a lookup is performed on the tuple and retrieves the result out. Problem with this is, every bundle must be of the same size, and it is having to be controlled by the algorithm and this is not the same with the Hash as the function is not biased and only depends on the input and this can bring changes to the several variations in the statistics of the bundles when it has large size.

02nd Feb 2022:

Today’s lecture was about understanding the underling details of the open-source relational database system. Firstly, about Logical structure of Database Cluster – collection of databases managed by a PostgreSQL server. Each database is a collection of tables, indexes, other objects (view, function, sequences). Database Cluster is referred as base directory- contains subdirectories and lots of files. Executing the “initdb” creates the base directory and is set to the environment variable (PGDATA). Tablespace is an additional area apart from the base directory. Data file is divided into pages and blocks of fixed length (8KB). Hierarchy of the data stored as tuple, page, files. Each page contains information related to how it is stored and also point to the subsequent records using pd_lower and pd_upper flag. Tuples are basically read using two methods- sequential, B-tree scan.

Also, several processes cooperatively work together to manage a database cluster – typically involves server process, backend process, background process, replication associated processes, background associated processes each of the work hand-in-hand to make database cluster work. Backend process consist of 5 subsystems- parser (parses tree from SQL statement in plain text), Analyzer (carries out a semantic analysis of a parse tree and generates a query tree), rewriter (transform a query using the defined rules), planner (generates the plan tree that can effectively execute), executor (executes via accessing tables and indexes).

04th Feb 2022:

The class details the discussion on Classic & Modern Computer Architecture and the nuances involved in their designs and the functionalities. It dates to 1940-50, idea of computational unit entails the execution of instructions serially and the processing time is constant with each and every instruction every time and deterministic in nature, often considered as the fast computation and has toll on the storage makes it slow.

Memory bus combines the CPU with the memory which is a separate component. It is as fast as CPU and has the capacity of 1-4 bytes at a time. Also, I/O peripherals are way slow when compared to the CPU and memory. They simply take instructions from CPU with the help of I/O bus, generally I/O and memory are combined with all the architectures.

With the modern architectures, such as Intel 80486 – 1987; the instruction passed to the CPU are way efficient when compared with others. This includes the implementation of the floating-point execution units in CPU with 30386 and 30387. These new architectures found a way to reduce the efforts on memory for the repetitive usage of the data using cache. Although the size of cache is smaller and yet it is efficient. It includes on cache on chip L1 build and allows an external L2 cache which is 100x faster than the memory and later is 10x slower than L1. It basically acts as mediator for memory access. Intel Pentium 80586 (1995-1998) inspired from DEC Alpha- has 3 execution units. It analyzes the set of instructions ahead of time and finds an equivalent that is efficient. Super scalar execution helps us to create the instruction pipelines.