**02/21/2022**

The process of mapping the file or a portion of the file which is on a disk to a range of address on memory such that it can be referenced via many applications address space just like dynamic memory. It basically enables I/O operations on the file much faster, and efficient. Linux Operating system does implement with ‘mmap’ system call. It acts just like an alternative to read/write operations. Generally, when read from a file with a particular memory location on the disk, behind the scenes the mmap points to the location offset defined to the address which is similar to the strategy used by MongoDB. Common operations on files such as open, read, close, write and iseek uses the data on disk and involves fetching it, involves the mmap operations for movement of the data from disk.

Several properties include access of the same segment of the data using two different applications eventually makes it reference the same real memory page. Behind the scenes mmap distributes the shared data across different applications to utilize it and it doesn’t have any anonymous file with an exception when postgres use mmap as it is a single threaded. There is zero copy approach, and all of the accesses were aligned properly. It can load the .so files too which are divided and stored across the system. Can also restrict the data to read-only mode and cannot make the I/O operations as write to it. In order to control the memory allocation used by mmap with the help of mprotect. It can help us get read-only memory, further can help us access the memory allocation with NUMA nodes for the scientific computation.

**02/23/2022**

Understanding the stance of Modern Architecture and its ideas towards the cost of implementation. One way or the other, we can recognize it as the Distributed System where each task can be accomplished with the right communication with the distributed system over the network which connects them. Although, this implies difficulties with Synchronization and cache traffic coherency, also with the introduction of several new protocols and we can easily compare the I/O of disk with memory. Distributed Systems works efficiently as they divide the implementation into parts and try to achieve the result and combine asynchronously. Partition depends on the # of cores, delegation of work is done to workers upon finishing the assigned task tries to achieve the average performance.

Let’s try to implement some of the most used Ideas, Linear Scan- which performs the search operation based on a peculiar key and returns the respective row or columnar values. One way to implement with asynchronous access and it tries to perform the search in a random fashion assuming not multiple devices behavior. Let us consider we have several devices and using filesystem provides issues that every time only one device gets the access to the data which has small sized pages, requires many search operations. Can we have a workaround for the same? It turns out that all modern OS has the access to the filesystem and its partitions. Another approach is to use stripped Raw Disk. With Column stored approach we get medium level of performance, as it stores only the most frequently used attributes, and such is