03/28/2022

There have been several scenarios where we need to handle joins of large tables producing a complex and huge data to compute or perform the matching between them, and this is a hectic process and often consumes lots of resources and time. Thus, it can be handled with ease using large hash tables. It helps is performing table level operations with the help of segmentation which is often done with the help of making the entire data into segments of data of powers of 2, such that it is easy to fit and perform hashes on top of it. Also, an advantage of it is to parallelize the instructions to perform hashing. A dynamic data structure “Linear Hashing” is usually used to perform hash and to increase or reduce one bucket at an instance. The advantage of it is the file system corresponding to it has dynamic nature to respond to the changes in file size thus to extend or decrease the usage of it by a significant amount eliminating the need to for costly file arrangement.

Linear Hashing usually expands with the division of a pre-determined bucket into 2 buckets and contracts it by merging 2 pre-determined buckets into one LH* which makes this to create a scalable distributed data structure with the help of linear hashing. Each of the bucket in LH* stored on separate server, LH* is enhanced making sure to have data availability even when there is a failure with the bucket. In LH and LH*, inserts, deleted, reads and updates are performed using the keys making to utilize less time regardless of the time for the number of buckets and records.

03/30/2022

There was a discussion about the paper "UDA-GIST: An In-Database Framework to Unify Data-Parallel and State-Parallel Analytics." User-Defined Aggregate (UDA), a data-driven operator included in almost all major database management systems, may be used to execute many analytical algorithms in parallel. UDAs can't be used to develop statistical algorithms like Markov chain Monte Carlo since most of the work is done through repetitive transitions over a large state that can't easily be partitioned due to data dependence (MCMC). In most cases, pre-processing is required to construct the huge state in the first place, and post-processing is required following statistical inference. As defined in this study, GIST (General Iterative State Transition) is a one-of-a-kind database operator for concurrent iterative state transitions over large states. GIST uses a series of transitions to change a UDA-created state.

04/01/2022

With the advent of new technology and we are in a situation where we are overwhelmed with large amounts of data in size, we often face issues in processing it, a programming technique which emerged to resolve the same and is quite often used for implementation for processing gigantic sized datasets on clusters working either in a distributed fashion or in a parallelize manner known as Map Reduce. It is combination of map approach for performing filtering and sorting the data as per the requirements by reducing the overhead for the aggregation of the obtained results. With the help of marshalling distributed servers, and by running numerous jobs in parallel involves communication with each other managing each other responses and also the data transfers among the system with the intent to reduce the redundancy of doing the same task multiple times and with the implementation of fault tolerance for the soft retrieval of the data in need. All of these activities are delegated by the Map Reduce system allowing to perform distributed processing of map reduce operations. As long as the map performed don’t have any dependency with other runs can be managed with other processing accessing it to finish early using less time to finish thereby reduces the CPU resources. Likely, a large number of reductions can be performed together when all of them are based on a similar key making it to look into it all at once reducing the time

to process. Even though, it may be inefficient when compared with other linear algorithms due the running of the multiple reduction processes it has the advantage of handling huge sized datasets in. clustered or distributed environment.