04/11/2022

Time is one of the factors that decides the efficiency of the solution, similarly we need to identify the means where it can be reliable with the increase in the size of the input, generally it is measured using the asymptotic analysis. Several grades of the process are used to derive the efficient solution. SQL query optimization is the process of determining the most efficient way to execute a SQL statement, and it is an important part of any program because it can improve the user experience. Even minor adjustments to SQL queries can dramatically increase speed, and I've developed a few best practices that have helped me improve query performance over time, and I hope these will be useful to anyone wanting to design performant queries.

Some of the ways we change the code faster using Inlining, Loop Unrolling etc. Inline functions are a C++ feature that improves the speed of a program's execution. Functions can be told to make themselves inline so that the compiler can use them instead of their definitions whenever they are called. Instead of referring to the definition of inline functions during runtime, the compiler changes it at build time. Creating Equivalent programs, considering only left-deeply plans would help in solving several issues. Evaluating the query plan is also important aspect. In a database setting, query processing is the process of deducing information from the database. One of the primary difficulties with query processing is how to process queries efficiently due to the vast amount of data accessible. In many circumstances, getting specific responses in the short query response time is impracticable or too expensive for users. Optimization based on costs (Physical) This is determined by the query's cost. The query can take several paths depending on indexes, restrictions, sorting algorithms, and so on. Optimization based on heuristics (Logical) Rule-based optimization is another name for this technology.

04/13/2022

In today's session we discussed on understanding the plans like ordering of table joins, plans related to left-deep, cost function analysis, total cost, etc. The cost of running a query plan is calculated by the Query Optimizer based on two key factors such as the cardinality of a query plan refers to the total number of rows handled at each level of the plan. The cost model of the algorithm is determined by the query operators. Cardinality, the first factor, is utilized as an input parameter for the cost model, the second factor. As a result, increased cardinality leads to more accurate cost estimates and, as a result, speedier execution plans.

We have discussed one issue related to Process streaming data, also we have discussed several techniques to perform the same. Assume uniform distribution, Sampling, Non-Parametric Statistical model, square error, Compression, wavelets. In a database management system, sampling can be utilized to offer low-cost estimates of aggregate query replies. Database parameters utilized by the query optimizer to choose query evaluation plans can also be estimated through sampling. Non-parametric Models are statistical models that rely on continuous data rather than discrete values and hence do not always conform to a normal distribution. Ordinal numbers, or data that does not have a fixed value like a discrete number, are frequently used in non-parametric statistics.

04/15/2022

Today's discussion on Query estimation of the cardinality selection before processing of the tables. The selectivity is much more important to fetch the results fastly. Several techniques to choose them are a direct approach for selectivity and street-smart approach. Involving the methodologies to choose the small joins resulting in less risky joins. Also, sometimes we rely on Indexing values of the data. Within Postgres, HStore is a key-value store. You can use it in the same way you would a dictionary in any other programming language, however it is specific to a database table field. H-Store was marketed as part of a new class of parallel database management systems known as NewSQL that provide the high-throughput and high-availability of NoSQL systems without sacrificing the transactional consistency of an ACID-compliant DBMS (atomicity, consistency, isolation and durability). Rather than using a single, more powerful, and expensive machine, such systems use several machines.

By omitting numerous aspects of typical relational database management systems, H-Store is able to do transaction processing at a high throughput.