International Journal of Advances in Computer Science & Its Applications – IJCSIA
Volume 4: Issue 4 [ISSN 2250-3765]
Publication Date : 27 December,2014

Query Optimization Methods for Improve Query Execution Time using SQL Technologies

AHMED KHALAF ZAGER ALSAEDI

***Abstract*---Generally, the query optimizer cannot be accessed directly by users: once queries are submitted to database, and parsed by the parser, they are then passed to the query optimizer where optimization occurs. Each different way typically requires different processing time. The purpose of query optimization, which is an automated process, is to find the way to process a given query in minimum time. According to the basic operation of query such as scanning, aggregation, join. we present a four type of query a cross different size of tables to explain query performing through execution time to improve query execution time during optimization .**

***Keywords*---- Query optimization, Query execution time, SQL server 2012 technologies**

1. Introduction

The SQL Server Query Optimizer is a cost-based optimizer. It analyzes a number of candidate execution plans for a given query, estimates the cost of each of these plans, and selects the plan with the lowest cost of the choices considered [1]. Indeed, given that the Query Optimizer cannot consider every possible plan for every query, it actually has to find a balance between the optimization time and the quality of the selected plan.

However one of the challenges in query optimization is a length in query execution time therefore it is necessary to find new SQL technologies used to reduces a query execution time during the optimization according to the basic operations of query such as scanning, aggregation, join algorithms.

Once the optimizer has generated an execution plan, or retrieved one from cache, the action switches to the storage engine, which usually executes the query according to the plan [2]. In this paper we Using the database tester to measure query execution times according to very important operators query like scan, aggregation and join to improving query execution time and to enhancements the DBMS performance .

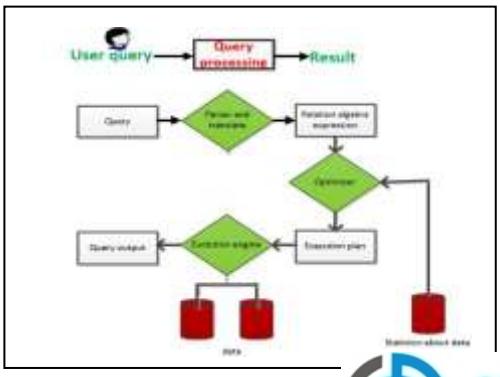
II. How the Query Optimizer Works?

At the core of the SQL Server Database Engine are two major components: the storage engine and the query processor also called the relational engine. The storage engine is responsible for reading data between the disk and memory in a manner that optimizes concurrency while maintaining data integrity. The query processor, as the name suggests, accepts all queries submitted to SQL Server, devises a plan for their optimal execution, and then executes the plan and delivers the required results.

Queries are submitted to SQL Server using the SQL language (or T-SQL, the Microsoft SQL Server extension to SQL). Since SQL is a high-level declarative language, it only defines what data to get from the database, not the steps required to retrieve that data, or any of the algorithms for processing the request. Thus, for each query it receives, the first job of the query processor is to devise a plan, as quickly

as possible, which describes the best possible way (or, at the very least, an efficient way) to execute said query. Its second job is to execute the query according to that plan [3].

Each of these tasks is delegated to a separate component within the query processor; the Query Optimizer devises the plan and then passes it along to the execution engine, which will actually execute the plan and get the results from the database. In order to arrive at what it believes to be the best plan for executing a query, the query processor performs a number of different steps; the entire query



Ahmed Khalaf Zager Alsaedi
Computer Science and Information Technology / UTHM

Figure 1: The query processing process

III. Data Access operators

A. Scanning

A Scan operation always scans the entire table, and. then filters out values to provide the desired result. The larger table, the more time the Scan will take to complete Let's start with the simplest example, by scanning a heap which, is performed by the Table Scan operator[4]. The following query on the Adventure Works database will use a table scan, as shown in Figure2

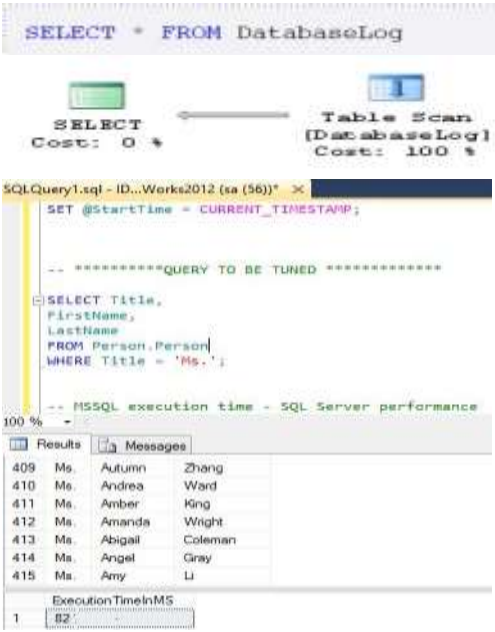
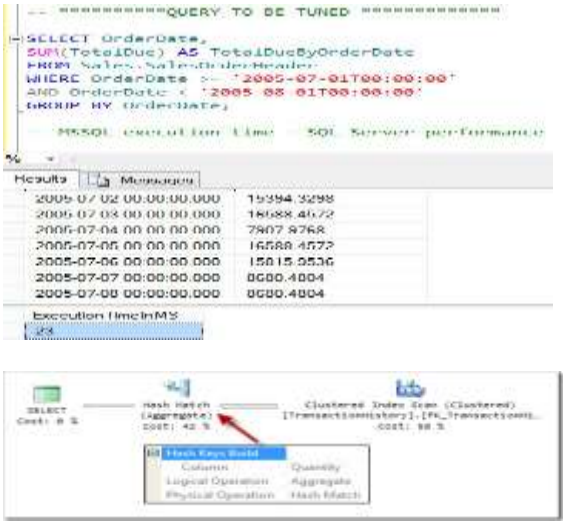


Figure 2: query plan and query execution time for scan method

B .Aggregation method

An aggregate is a collection of items that are gathered together to form a total quantity, they use to make the components of a computing system work better together. Aggregations are used in databases to summarize information about some set of data. The result can be a single value, such as the average salary for a company, or it can be a per-group value, like the average salary by department. SQL Server has two operators to implement aggregations, Stream Aggregate and Hash Aggregate, and they can be used to solve queries with aggregation functions (like SUM, AVG or MAX), the GROUP BY clause, or the DISTINCT keyword to reduce query execution time show in figure 3.



Figurer3: query plan and query execution time for aggregation method

C. Join tow table

The optimizer can select from multiple join methods. When the rows from two tables are joined, one table is designated the outer table and the other the inner table. During a join, the optimizer scans the rows in the outer and inner tables to locate the rows that match the join condition show in figure 4.

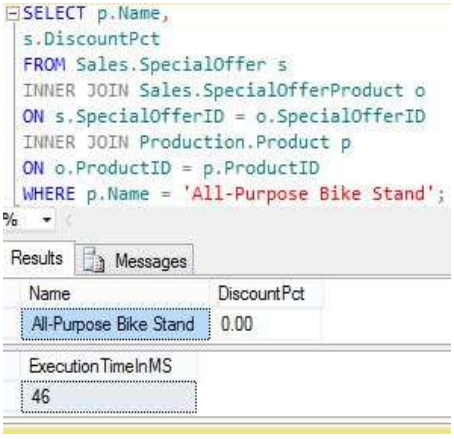


Figure 4: Query plan tow tables join method

D .Join three table method

You can use SQL JOIN statements to combine data from three or more tables. In an earlier article, we took a look at using inner joins and outer joins to combine to reduce query execution time show in figure 5.

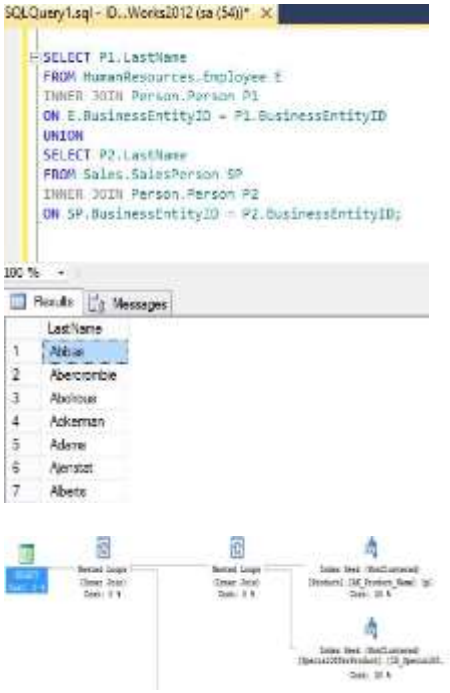


Figure 5: Query plan join three table methods

IV. Objective and propose the Query Execution Time

The purpose of this paper to execution four type of query across different size of table by using new method in SQL server to reduces the query execution time , and we focus on a model of execution time for basic data access operator (scan, aggregation, join) application. According to query execution time for each query, the application generates a sequence of SQL statements, to enhancement the DBMS performance [9].

In the context of RDBMS, previous research does not make clear the unique role that database integrity constraints can play in SQL optimization. In addition, the close relationship between SQO and the enforcement of database integrity constraints has not been established at the time of query execution, we use the basics operation query for different size tables in SQL server 2012 and suppose Q1:scanning ;Q2:aggregation ;Q3:join tow table Q4:join three tables show in figure 7.

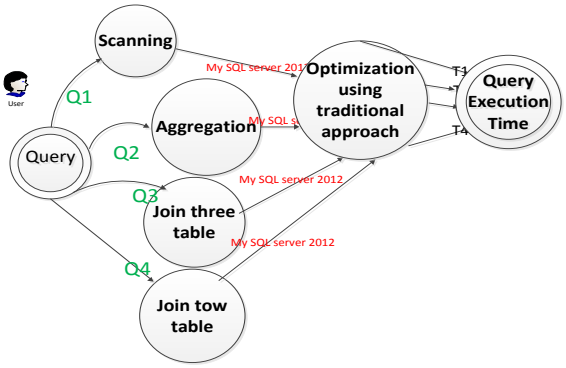


Figure 7:.framwork this paper

IIV.MEASUREMENTQUERY EXECUTION TIME

The result of this paper the query execution time is degrees in each operator of query because we use new technologies my SQL server 2012 to reduce amount of material and item passing through processing or system to get reasonable time show in fig.6 and we use this algorithm to calculate execution time for each algorithm [5,6,7].

```
USE AdventureWorks2012;
 CHECKPOINT
DBCC FREEPROCCACHE --
DBCC DROPCLEANBUFFERS
go
DECLARE @StartTime datetime,
@EndTime datetime
SET @StartTime =
CURRENT_TIMESTAMP;
SET @EndTime =
CURRENT_TIMESTAMP
SELECT ExecutionTimeInMS =
DATEDIFF(ms, @StartTime, @EndTime)
GO
```

Table 1.Summary of query execution time with set of methods

query excuted	method s	Elapesd time (query execution time)
Q1	scaning	82
Q2	Aggregation	23
Q3	join two table	46
Q$	join three table	46

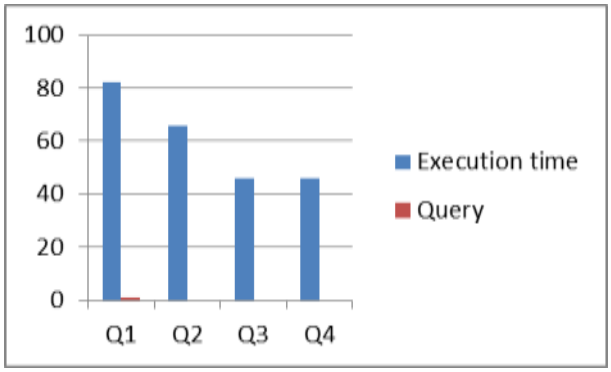


Fig6. Query methods to the execution time query

VI. Acknowledgment

I want to thank God who helped me to find a way to calculate the query equation time with group of algorithms that are used to improve times and to less effort to the materials and items crossing the system, and this methods very important to query optimization by SQL technologies and I will continue to search for other ways more useful in the development and improvement of the performance of relational databases.

V. Future work

This work can use in normal database to improve and development the performance database speciality in business and financial database. But this technologies not enough to normalization and optimization data for multi query in large database therefore we searching about new approach suitable with large database such as decision support system and data warehouse in future we propose swarm intelligent as important approach to work with optimization to improve performance of DBMS .

VI. Conclusion

Through the use of technology developed for SQL Server, These techniques include things such as scanning ,aggregation ,join this traditional technology that use to determine the best query plan (in shorter time and minimum cost in traditional DBMS, but this technology not enough to get best query plan and best query plan if we treatment with large DBMS like data warehouse or decisions support system therefore in future is necessary to obtain new approach like intelligent approach to find the best query plan .

REFERENCES

[1] Benjamin Nevarez. (2010). Inside the SQL Server Query Optimizer: First Publisher: Red Gate Books .

[2]

[3] Grant Fritchey.(October 12, 2012). SQL Server Execution Plans: 2nd Edition ***Publisher*****:** Red Gate Books.

[4] Benjamin Nevarez. (2010). Inside the SQL Server Query Optimizer: First ***published*** Book.

[5] Kalen Delaney.(2012). SQL Server Concurrency Locking, Blocking and Row Versioning : ***First published*** Book, California.

[6] http://stackoverflow.com/questions/595762/calculate-execution-time-of-a-sql-query.

[7] Jason Primal, David Dye. (2012).SQL server 2012 T_SQL Recipes

[8] Computing, database, Introduction to SQL Server 2012,using database: http://databases.about.com/od/sqlserver/a/Introduction-To-Sql-Server-2012.htm.

[9] M. Siegel, E. Sciore, and S. Salveter. A method for automatic rule derivation to support semantic

[10] Amol Deshpands ,Zachary lves, Viayshankar Roman "Adaptive Query processing :Why,How,Whene,Whatnext? 2007 ,vennaAustria .Copyright 2328.2007