

Improving Query Processing Performance in Database Management Systems

Suren Dilanka Gamage

149210D

Faculty of Information Technology

University of Moratuwa

May 2018

Improving Query Processing Performance in Database Management Systems

Suren Dilanka Gamage

Index no: 149210D

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of the requirements of the Master of Science in Information Technology.

May 2018

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

.....
Name of Student (s)

.....
Signature of Student (s)

Date:

Supervised by

.....
Name of Supervisor(s)

.....
Signature of Supervisor(s)

Date:

Dedication

This thesis is dedicated to my wife, Mrs. U. Kumarapeli for her endless love, encouragement, and support.

Acknowledgments

First and foremost I would like to offer my sincere gratitude to my research supervisor, lecturer Mr.Chaman Wijesiriwardana, for his guidance, supervision, encouragement, and support throughout this study.

I would not know what research is and how to do research if the lecture series of thesis writing and research methodologies are not offered. So I would like to offer my sincere gratitude to the Prof. Asoka S. Karunananda for feeding the knowledge and guidance for doing researches in the proper way.

I would also like to thank my lecturer S. Premaratne and all the lecturers of Faculty of Information Technology – University of Moratuwa, for their guidance and encouragement to get maximum use of knowledge and capabilities.

I am grateful to the management and Softlogic Holdings PLC for their kind support and understanding during this work.

Finally, I would like to extend my deepest gratitude to my parents and family, for their continuous support given in every possible way to make this project successful.

Abstract

Improving Query Processing Performance in Database Management Systems has been a research challenge. This is the most important and is a real problem, this happens to be very crucial in large organizations with heterogeneous data, online system, billing systems and so on. Among other issues in the query optimization problem, faced by everyday query optimizers, get more and more complex with the server increasing complexity of user queries. During the last decade, database management systems have become important information processing system supporting business activities of geographically decentralized organizations.

The Performance monitoring has been evaluated and used by various tools. Most DBA's agreed that these tools are valuable. Our research also tried to identify how performance problems could be reduced and which methods were used in practice. Besides hardware upgrades, the following areas in tuning are known to have major impacts.

The main aim of this thesis is to produce flexible database monitoring tool and query optimization techniques that is capable of get basic idea of database server, database log, missing indexes, graphical user interface of currently running queries, optimizing large queries in a complex database. Among other issues in a database, such as deadlock, expensive query, primary key missing places, badly design queries can be simply identified.

This database monitoring tool and proposed new optimization techniques will more helpful to identify database performance issues and provide better solutions. During the evaluation, it was shown that system was successful more than 70%.

Table of Contents

| | |
|--|-----------|
| Chapter 1 –Introduction | 1 |
| 1.1 Prolegomena | 1 |
| 1.2 Background and Motivation | 1 |
| 1.3 Problem statement | 1 |
| 1.4 Hypothesis | 2 |
| 1.5 Objectives | 2 |
| 1.6 Structure of the Thesis..... | 2 |
| 1.7 Summary | 2 |
| Chapter 2 -Developments and Challenges in Improving Query Processing Performance in Database Management Systems | 3 |
| 2.1 Introduction | 3 |
| 2.1 Early developments | 3 |
| 2.3 Modern trends in Improving Query Processing | 4 |
| 2.4.1 Future challenges of Improving Query Processing | 4 |
| 2.4.2 Big Data Management System | 5 |
| 2.4.3 Big Data Service Model | 5 |
| 2.4.4 Non-structural and Semi-structured Data Storage | 6 |
| 2.4.5 Data Virtualization Platform | 6 |
| 2.4.6 Distributed Applications | 6 |
| 2.4.7 Map Reduce | 6 |
| 2.4.8 Map Reduce Optimization | 7 |
| 2.4.9 Data Transfer Bottlenecks | 7 |
| 2.4.10 Index Optimization | 8 |
| 2.4.11 Iterative Optimization | 8 |
| 2.5 Summary | 8 |
| Chapter 3 - Technology Adopted for Improving Query Processing Performance in Database Management Systems..... | 9 |
| 3.1 Introduction | 9 |
| 3.2 Technologies available | 9 |
| 3.2.1 Database Monitoring | 9 |
| 3.2.2 Query Analyzing and Optimization | 9 |
| 3.3 Technology Stack | 10 |
| 3.4 Summary | 10 |
| Chapter 4 -An approach to on Improving Query Processing Performance in Database Management Systems | 11 |
| 4.1 Introduction | 11 |
| 4.2 Hypothesis | 11 |
| 4.3 Users | 11 |
| 4.4 Input | 12 |
| 4.5 Output..... | 12 |
| 4.6 Process | 12 |
| 4.7 Features..... | 20 |

| | |
|--|-----------|
| 4.8 Summary..... | 20 |
| Chapter 5 - Design and Implementation of Database monitoring app and Database query optimizing the prototype..... | 21 |
| 5.1 Introduction | 21 |
| 5.2 Design Database Monitoring Application | 21 |
| 5.3 Implementation of Database Monitoring Application | 21 |
| 5.4 Query Optimization Techniques (Prototype) | 22 |
| 5.5 Implementation of Query Optimization Techniques..... | 29 |
| 5.6 Overall System | 33 |
| 5.7 Summary | 33 |
| Chapter 6 - Evaluation..... | 34 |
| 6.1 Introduction | 34 |
| 6.2 Setup | 34 |
| 6.3 Evaluation Methodology for Database Monitor Application | 34 |
| 6.4 Evaluation Methodology for Proposed New Optimization Techniques..... | 34 |
| 6.5 Participants | 50 |
| 6.6 Data Collection..... | 50 |
| 6.7 Discussion | 50 |
| 6.8 Summary..... | 50 |
| Chapter 7- Conclusion and Further Work..... | 51 |
| 7.1 Introduction | 51 |
| 7.2 Overall Conclusion..... | 51 |
| 7.3 Objective Wise Conclusion..... | 51 |
| 7.4 Further Work..... | 51 |
| 7.5 Summary..... | 51 |
| References | 52 |

| | |
|--|-----------|
| Appendixes..... | 54 |
| Appendix A - User Interface and Architecture Diagram of the System..... | 54 |
| Security Module – Authentication..... | 54 |
| Control Module-Server and Database Information..... | 54 |
| Server Configuration..... | 55 |
| Database Server Performance Analyzer..... | 55 |
| Database Log Information and Suggestions..... | 56 |
| Database Performance Improvement Suggestions..... | 56 |
| Database Waiting Tasks..... | 57 |
| Database Missing Index Details and Suggestions..... | 57 |
| Database IO Operations..... | 58 |
| Database Objects and Details..... | 58 |
| Database Monitoring Application Options..... | 59 |
| Appendix B – Evaluation of Database Monitoring Application..... | 60 |
| Appendix C – Evaluation of proposed optimization techniques..... | 65 |

List of Figures

| | |
|---|----|
| Figure 3.1 – Technology Stack..... | 10 |
| Figure 4.6.1 - Query cost..... | 16 |
| Figure 4.6.2 - Execution plan..... | 17 |
| Figure 4.6.2 - Execution plan..... | 24 |
| Figure 5.2 find the 20 worst performing queries..... | 25 |
| Figure 5.3 How many times execution plan is re-used..... | 26 |
| Figure 5.4 Find unnecessary indexes..... | 28 |
| Figure 5.5-Index creation process..... | 30 |
| Figure 5.6-Union operators..... | 31 |
| Figure 5.7-Group by clause..... | 32 |
| Figure 5.8-Group by clause with count..... | 33 |
| Figure 5.9-Group by clause with more column..... | 33 |
| Figure 7.4.1.1 - Database server information from newly developed Database Monitoring Application..... | 61 |
| Figure 7.4.1.2 - Database server information..... | 61 |
| Figure 7.4.1.3 - Database server statistics from newly developed database monitoring application..... | 62 |
| Figure 7.4.1.4 - Database server statistics..... | 62 |
| Figure 7.4.1.5 – Missing index suggestions from newly developed database monitoring application..... | 63 |
| Figure 7.4.1.6 – Missing index suggestions by manually..... | 63 |
| Figure 7.4.1.7 – Database memory utilization details..... | 64 |
| Figure 7.4.1.7 – Database lock..... | 64 |
| Figure 7.4.1.8 - Currently running Processors | 64 |
| Figure 6.1 - Database Configuration..... | 65 |
| Figure 6.1.2 - Complex SQL Query..... | 34 |

| | |
|--|-----------|
| Figure 6.2 - Complex Query Execution Time..... | 65 |
| Figure 6.3 - QEP Plan..... | 66 |
| Figure 6.3.1 - Proper Index..... | 36 |
| Figure 6.4 – Query Execution Time after Index..... | 66 |
| Figure 6.4.1 - Difference between before and after indexes..... | 36 |
| Figure 6.5 – SQL Profiler..... | 67 |
| Figure 6.5.1 – Take high execution query by SQL Profiler..... | 37 |
| Figure 6.6 - SQL Profiler result..... | 67 |
| Figure 6.6.1 - Difference between before and after query optimized..... | 37 |
| Figure 6.7.1 – Traditional Query..... | 38 |
| Figure 6.7 – SQL Server Execution time for Traditional query..... | 68 |
| Figure 6.8 – SQL Server Execution time for our new proposed query..... | 68 |
| Figure 6.9 - Analyze by using Sentry Plan explore with IN..... | 69 |
| Figure 6.10 - Analyze by using Sentry Plan explore without IN..... | 69 |
| Figure 6.11.1 – Query with temp table..... | 40 |
| Figure 6.11 – Query cost with temp table..... | 70 |
| Figure 6.11.2 – Query with #temp table..... | 41 |
| Figure 6.12 - Figure 6.11 – Query cost with #temp table..... | 70 |
| Figure 6.13.1 - Query with @temp table..... | 42 |
| Figure 6.13 - Query cost with @temp table..... | 71 |
| Figure 6.14 - Sentry plan with #temp table..... | 71 |
| Figure 6.15 - Sentry plan with @temp table..... | 72 |
| Figure 6.15.1 - #Table and @Table Difference..... | 43 |
| Figure 6.16 - How to find missing index..... | 72 |
| Figure 6.17.1 - Best practice for IN and Where..... | 44 |
| Figure 6.17 - Analyzed best practice IN and Where Clause..... | 73 |

| | |
|---|-----------|
| Figure 6.18.1 - Bad practice for IN and Where..... | 44 |
| Figure 6.18 - Analyzed Bad practice IN and Where Clause..... | 73 |
| Figure 6.19.1 - Bad practice for IN and Where..... | 45 |
| Figure 6.19 - Analyzed bad practice IN and Where Clause..... | 74 |
| Figure 6.20.1 - Correlated SQL subqueries..... | 46 |
| Figure 6.20 - QEP plan and Cost of Correlated SQL subqueries..... | 74 |
| Figure 6.21- QEP plan and Cost of Correlated SQL subqueries in Sentry planner..... | 75 |
| Figure 6.22.1 – Solution for Correlated SQL subqueries..... | 46 |
| Figure 6.22- Our Query QEP plan and Cost of Correlated SQL subqueries..... | 75 |
| Figure 6.23 - Our Query QEP plan and Cost of Correlated SQL subqueries in Sentry planner | 76 |
| Figure 6.24.1 – Query with Cursor..... | 47 |
| Figure 6.24 – QEP in Cusror..... | 76 |
| Figure 6.25.1 - Alternative solution for cursor..... | 48 |
| Figure 6.25 - Alternative solutinon QEP plan and query cost..... | 76 |
| Figure 6.26.1 - Using User Defined Functions..... | 49 |
| Figure 6.26 - Set no count on execution time..... | 77 |
| Figure 6.27 - Without no count execution time..... | 77 |
| Figure 6.27.1 - Difference between set no count and without no count..... | 50 |

List of Tables

| | |
|---|-----------|
| Table 7.1 – Evaluation functionality in database monitoring application..... | 60 |
| Table 6.8.1 – Differences between with IN and Remove IN..... | 39 |
| Table 6.5 – Difference between set no count and without no count..... | 49 |