AdaNexus: An Improved Nexus Algorithm

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

Master of Technology

IN

Faculty of Engineering

BY

Achint Chaudhary

Computer Science and Automation

Indian Institute of Science

Bangalore – 560 012 (INDIA)

July, 2020

Declaration of Originality

I, **Achint Chaudhary**, with SR No. **04-04-00-10-42-18-1-15879** hereby declare that the material presented in the thesis titled

AdaNexus: An Improved Nexus Algorithm

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2018-20**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: Advisor Signature

DEDICATED TO

My Parents & Close People

For those whom I have lost, wish you can see this

Acknowledgements

I am deeply grateful to Prof. Jayant R. Haritsa fir his unmatched guidance. He was and always will be an inspiration for me for his thoughts and his perception of looking at things around us, be it academia or real life. It is my good luck, that I got a chance to work with him and obtain few chunks of knowledge from him.

I am thankful to Anshuman Dutt, Srinivas Karthik and Sanket Purandare for their guidance and discussions. I would express my gratitude towards my lab mates for suggestions during my work. My sincere thanks goes to my friends Diksha and Hemant, who have patiently listened and understood many complications, and help me bring out much clear pictures to work with. Also, CSA and CDS office staff was the team of people who make academic functioning smooth with best co-ordination possible.

Finally, I would express my gratitude to my parents, who have taught me how to stay calm and keep working on what is essential, nothing I have done or will ever do will match their support that I carry within.

Abstract

Declarative query processing in database systems often leads to sub-optimal performance due to wrong selectivity estimation from those encountered during actual execution. Plan Bouquets is a technique proposed to substitute selectivity estimation by selectivity discovery at run-time, to provide worst case performance guarantees. This is done by performing multiple partial executions for same query in an incremental fashion of cost budget from a bouquet which is compiled at very first stage.

This technique is suitable for OLAP queries as high overheads of optimizing most part of selectivity error space are amortized over multiple invocation of query in OLAP scenarios. Full space exploration overheads in past are improved upon with NEXUS, which is an algorithm, that only discovers points useful for bouquet compilation.

In this work, we proposed an adaptive version of NEXUS named AdaNEXUS, which utilize geometrical properties of contours to be discovered for bouquet. This algorithm reduces overheads of compilations empirically keeping the worst case performance complexity same. Further, we provide upper bounds for maximum cost deviation possible during bouquet compilation due to use for either NEXUS or AdaNEXUS. Evaluation of proposed system is done on TPC-DS benchmark with different scales to test system. It is demonstrated that around an order of magnitude reduction is observed in compilation overheads when compared with NEXUS, also quality of plans discovered by AdaNEXUS is better than those discovered by NEXUS.

Contents

Acknowledgements **i**

Abstract **ii**

Contents **iii**

List of Figures **v**

List of Tables **vi**

1 Introduction **1**

1.1 Background 1
1.2 Motivation 2
1.3 Contributions 2
1.4 Organization 2

2 Plan Bouquets **4**

2.1 How to Write a Thesis: An Introduction 4

3 Problem Formulation **5**

3.1 How to Write a Thesis: An Introduction 5

4 Leveraging Contour Geometry **6**

4.1 How to Write a Thesis: An Introduction 6

5 Cost Deviation Bounds **7**

5.1 How to Write a Thesis: An Introduction 7

6 Experimental Evaluation **8**

6.1 How to Write a Thesis: An Introduction 8

7 Conclusion and Future Work **9**
7.1 How to Write a Thesis: An Introduction 9

Bibliography **10**

List of Figures

List of Tables

Chapter 1

Introduction

SQL query processing is declarative in nature, user only specifies what needs to be done. There are a large number of execution strategies, each called query plan. All these query plans will yield same results but have different running times. Role of database optimizer is to select optimal (in terms of running cost) query plan. During choice of optimal query plan, database optimizer makes multiple cost based decision to compare different query plans. Cost for each physical operator in query plan is a function of number of tuple, it processes, known as cardinality. Cardinality normalized in range of $[0, 1]$ is known as selectivity throughout literature. Selectivity estimations of optimizer for identifying the optimal query plan are done using statistical models and meta-data information about schema. Selectivity estimates are often sub-optimal which results in inflated query response time.

1.1 Background

There are multiple techniques proposed in literature to improve quality of selectivity estimates like better statistical models, on-the-fly re-optimization, etc., but none of them provides worst case performance guarantee bounds.

An entirely different approach based on run-time selectivity discovery is proposed called *plan bouquets*, which for the first time, provides strong theoretical bounds on worst-case performance as compared to oracular optimal performance possible from all the available plan choices.

For each given query, predicates prone to selectivity error contribute as dimension in *Error-prone Selectivity Space(ESS)*. ESS is a multi-dimensional hyper-cube. The set of optimal plans over the entire range of selectivity values in ESS is called *Parametric Optimal Set of Plans(POSP)*. POSP is generated by asking optimizer's chosen plans at various selectivity locations in ESS using *selectivity injection module*. Cost surface generated over entire ESS is called *Optimal Cost Surface(OCS)*. An *Iso-cost Surface(IC)* is a collection of all points from OCS which have same cost of optimal plan at each of these locations cost.

1.2 Motivation

Compilation of plan bouquet is process of drawing iso-cost contours, which involves getting selectivity points and corresponding optimal plans at each point for any contour we are drawing. In experimental setting ESS is discretized at some resolution 'res' where let 'd' be the dimension of ESS. Number of optimizer calls in entire ESS can reach res^d. This number is exponential in number of dimensions and results in high overheads of bouquet compilation. NEXUS is an algorithm developed in past to avoid doing optimizer calls on entire ESS, it does so by making optimizer calls only on points lying on contours. Number of optimizer calls made by NEXUS to draw 'm' contours in worst case is twice of $m * res^{(d-1)}$, which is still exponential in nature.

When 'm' is sufficiently high and 'd' is also high, for keeping things computationally feasible a moderate choice of 'res' is made. When this is the case, contours drawn by NEXUS can suffer from cost deviation from ideal desired cost value which does affect worst case performance guarantees. So it is desired to keep total overhead feasible with acceptable contour cost deviations.

1.3 Contributions

In this work, we have devised algorithms to improve contour discovery for plan bouquet, which constitutes most in compilation overhead. We have con

- **Speeding-up contour discovery:** We proposed AdaNEXUS algorithm, which is an improvement over NEXUS, that utilizes geometric properties of iso-cost contours generally observed in practice. This algorithm is designed to reduce total overheads in terms of optimizer calls and also to keep low contour cost deviations then what is there with NEXUS algorithm.

- **Contour cost deviation bounds:** We provide upper bounds on contour cost deviation values for both NEXUS and AdaNEXYS, from which it will be clear that AdaNEXUS should be preferred over NEXUS.

1.4 Organization

Chapter 2 provides a brief detail on Plan Bouquets technique. Chapter 3 discusses existing bouquet compilation techniques. Chapter 4 is about leveraging geometric properties of OCS and iso-cost contours to improve upon NEXUS and come up with AdaNEXUS algorithm. Worst case cost deviation bounds for both NEXUS and AdaNEXUS are given in Chapter 5. Experimental evaluation of our work is given in Chapter 6. At last, Chapter 7 discussed conclusion of our work and future work that can be done further.

Chapter 2

Plan Bouquets

2.1 How to Write a Thesis: An Introduction

The first chapter should have the motivation of the problem and the brief description of the solutions provided in the thesis. There should be one outline section which should give detailed description of the thesis chapters. Here is an example citation, [1].

Chapter 3

Problem Formulation

3.1 How to Write a Thesis: An Introduction

The first chapter should have the motivation of the problem and the brief description of the solutions provided in the thesis. There should be one outline section which should give detailed description of the thesis chapters. Here is an example citation, [1].

Chapter 4

Leveraging Contour Geometry

4.1 How to Write a Thesis: An Introduction

The first chapter should have the motivation of the problem and the brief description of the solutions provided in the thesis. There should be one outline section which should give detailed description of the thesis chapters. Here is an example citation, [1].

Chapter 5

Cost Deviation Bounds

5.1 How to Write a Thesis: An Introduction

The first chapter should have the motivation of the problem and the brief description of the solutions provided in the thesis. There should be one outline section which should give detailed description of the thesis chapters. Here is an example citation, [1].

Chapter 6

Experimental Evaluation

6.1 How to Write a Thesis: An Introduction

The first chapter should have the motivation of the problem and the brief description of the solutions provided in the thesis. There should be one outline section which should give detailed description of the thesis chapters. Here is an example citation, [1].

Chapter 7

Conclusion and Future Work

7.1 How to Write a Thesis: An Introduction

The first chapter should have the motivation of the problem and the brief description of the solutions provided in the thesis. There should be one outline section which should give detailed description of the thesis chapters. Here is an example citation, [1].

Bibliography

[1] Swaprava Nath. *Mechanism Design for Strategic Crowdsourcing*. PhD thesis, Indian Institute of Science, Bangalore, 2013. forthcoming. 4, 5, 6, 7, 8, 9