



التخصص: هندسة كهربية
القسم: هندسة الحاسبات والنظم
الدرجة: الماجستير



كلية الهندسة

الاسم: احمد فاروق عبدالعزيز ابراهيم

تاريخ الميلاد: ١٩٧٧/٢/٢٣

البكالوريوس : يوليو ٢٠٠١

جامعة: الازهر

هيئة الاشراف

أ.د. هدى قرشى محمد

د. إسلام احمد محمد المداح

لجنة الحكم

أ.د. جمال الدين علي

أ.د. ماجده بهاء الدين فايق

Frequent Itemset Mining for Big Data

Data mining has been a powerful technique in analyzing and utilizing data in today's information rich society. A fundamental problem in data mining is the process of finding frequent patterns in large datasets. Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases. With the rapid development of computers and communication networks, data mining in distributed environment is becoming a heated problem. Distributed data mining (DDM) is expected to relieve current mining methods from the sequential bottleneck, and provide the ability to scale to massive data sets and improve the response time. The aim of this work is to present a modified parallel algorithm that makes an enhancement of parallel buddy prima algorithm. The modified parallel algorithm doesn't need to load all transaction in memory. It may be inapplicable for large data sets with many distinct items. Modified algorithm prevent full scan of all transaction set each time we calculate the support count of a candidate set. The algorithm is implemented, and is compared to its predecessor algorithms. The comparisons were made on processing time and the accuracy of each algorithm. Results of applying the proposed algorithm show faster performance than other algorithms without scarifying the accuracy.

تحسين الأداء في طرق التنقيب عن البيانات بإيجاد الانماط المتكررة في قواعد البيانات الضخمة

ان التنقيب عن البيانات من اهم الطرق المستخدمة في تحليل واستخدام البيانات في العصر الحديث. وتعد المشكلة الاساسية في التنقيب عن البيانات هي عملية ايجاد الانماط المتكررة في قواعد البيانات الضخمة. وذلك لان ايجاد الانماط المتكررة تلعب دورا هاما واساسيا في الوصول الي المعلومات الهامة من قاعدة البيانات. ونظرا للتطور السريع في مجال الكمبيوتر و تكنولوجيا الاتصالات مما كان سببا اساسيا في ظهور مفهوم جديد في التنقيب عن البيانات وهو الطرق الموزعة في التنقيب عن البيانات. والذي يعمل علي الحد من المشاكل التي تواجه طرق التنقيب العادية عن البيانات مثل الاعاقة المتتالية و تقليل الوقت الزمني اللازم عند التعامل مع قواعد البيانات الضخمة. فإن الهدف من هذا العمل هو تقديم تحسين في احدي الطرق الموزعة (parallel buddy prima) في التنقيب عن البيانات والتغلب علي كافة المشاكل التي واجهت هذه الطريقة. ومن هذه المشاكل الحاجة الي تحميل كافة البيانات في الذاكرة ويعد هذا الامر غير ملائم لقواعد البيانات الضخمة. يعمل النظام المقترح علي عدم الحاجة الي البحث في كافة السجلات الموجودة بقاعدة البيانات في كل مرة يتم فيها حساب عدد مرات تكرار الانماط المختلفة. ولقد تم تنفيذ الخوارزم المقترح ومقارنته بالخوارزميات السابقة من حيث تقييم الاداء في دقة النتائج والوقت اللازم لتنفيذ الخوارزم. وتظهر النتائج ان الخوارزم المقترح يعطي نتائج اسرع دون ادني تأثير علي دقة النتائج.

ملخص

ان التنقيب عن البيانات من اهم الطرق المستخدمة في تحليل واستخدام البيانات في العصر الحديث. وتعد المشكلة الاساسية في التنقيب عن البيانات هي عملية ايجاد الانماط المتكررة في قواعد البيانات الضخمة. وذلك لان ايجاد الانماط المتكررة تلعب دورا هاما واساسيا في الوصول الي المعلومات الهامة من قاعدة البيانات. و نظرا للتطور السريع في مجال الكمبيوتر و تكنولوجيا الاتصالات مما كان سببا اساسيا في ظهور مفهوم جديد في التنقيب عن البيانات وهو الطرق الموزعة في التنقيب عن البيانات. والذي يعمل علي الحد من المشاكل التي تواجه طرق التنقيب العادية عن البيانات مثل الاعاقة المتتالية و تقليل الوقت الزمني اللازم عند التعامل مع قواعد البيانات الضخمة. فإن الهدف من هذا العمل هو تقديم تحسين في احدي الطرق الموزعة (parallel buddy prima) في التنقيب عن البيانات والتغلب علي كافة المشاكل التي واجهت هذه الطريقة. ومن هذه المشاكل الحاجة الي تحميل كافة البيانات في الذاكرة ويعد هذا الامر غير ملائم لقواعد البيانات الضخمة. يعمل النظام المقترح علي عدم الحاجة الي البحث في كافة السجلات الموجودة بقاعدة البيانات في كل مرة يتم فيها حساب عدد مرات تكرار الانماط المختلفة.

و لقد تم تنفيذ الخوارزم المقترح ومقارنته بالخوارزميات السابقة من حيث تقييم الاداء في دقة النتائج والوقت اللازم لتنفيذ الخوارزم. وتظهر النتائج ان الخوارزم المقترح يعطي نتائج اسرع دون ادني تأثير علي دقة النتائج.

Abstract

Data mining has been a powerful technique in analyzing and utilizing data in today's information rich society. A fundamental problem in data mining is the process of finding frequent patterns in large datasets. Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases. With the rapid development of computers and communication networks, data mining in distributed environment is becoming a heated problem. Distributed data mining (DDM) is expected to relieve current mining methods from the sequential bottleneck, and provide the ability to scale to massive data sets and improve the response time.

The aim of this work is to present a modified parallel algorithm that makes an enhancement of parallel buddy prima algorithm. The modified parallel algorithm doesn't need to load all transaction in memory. It may be inapplicable for large data sets with many distinct items. Modified algorithm prevent full scan of all transaction set each time we calculate the support count of a candidate set. The algorithm is implemented, and is compared to its predecessor algorithms. The comparisons were made on processing time and the accuracy of each algorithm. Results of applying the proposed algorithm show faster performance than other algorithms without scarifying the accuracy.



Ain Shams University
Faculty of Engineering
Computer & Systems Engineering Department

Frequent Itemset Mining for Big Data

Thesis

Submitted in partial fulfillment of the Requirements for the M.Sc. Degree in Electrical
Engineering
(Computer and Systems)

By

Ahmed Farouk Abd El_Aziz
B.Sc in Computer and Systems Engineering

Supervised By

Dr. Hoda Korashy Mohamed
Computer & Systems Engineering Dept.
Faculty of Engineering, Ain Shams University

Dr. Islam El_Maddah
Computer & Systems Engineering Dept.
Faculty of Engineering, Ain Shams University

Cairo 2015

ABSTRACT

Name: Ahmed Farouk Abd El_Aziz.

Thesis title: Frequent Itemset Mining for Big Data

A thesis for M.Sc. degree. Ain Shams University, Faculty of Engineering, Computer and Systems Engineering Department, 2015.

Discovering the frequent item sets requires a lot of computation power, memory and input/output values, which is better to be provided by parallel processing. The implementation of data mining ideas in high performance parallel and distributed computing environments is becoming crucial for ensuring system scalability and interactivity as data continues to grow in size and complexity.

The aim of this work is to present a modified parallel algorithm that makes an enhancement of parallel buddy prima algorithm. The modified parallel algorithm doesn't need to load all transaction in memory. It may be inapplicable for large data sets with many distinct items. Modified algorithm prevent full scan of all transaction set each time we calculate the support count of a candidate set. The algorithm is implemented, and is compared to its predecessor algorithms. The comparisons were made on processing time and the accuracy of each algorithm. Results of applying the proposed algorithm show faster performance than other algorithms without scarifying the accuracy.

Keywords:

Frequent itemset, Big data, Parallel data mining

Acknowledgments

First, all my thanks and gratitude to Allah for accepting my prayer successfully complete this work .This thesis arose in part out of years of research. I have worked with a great number of people whose contribution in assorted ways to the research and the making of the thesis deserved special mention. It is a pleasure to convey my gratitude to them all in my humble acknowledgment.

In the first place i would like to thank Prof. Hoda Korashy Mohamed for her qualified support and her availability. Her advices were precious and fundamental for the development of my work. I will be grateful to her.

I want to thank Dr. Islam El_Maddah. I appreciated his help and his continuous support through my work time.

Finally, a special acknowledgement to my father, my mother, and my wife. Their love and support through these years was very important. Words alone cannot adequately express my gratitude.

Contents

Table of Contents

Chapter 1 INTRODUCTION	1
1.1 Background	1
1.2 Motivation.....	2
1.3 Objectives.....	2
1.4 Contribution.....	2
1.5 Thesis Layout.....	3
CHAPTER 2 DATA MINING	4
2.1 Data Mining	5
2. 2 What Data Mining can and cannot do?	6
2.3 DATA MINING LIFE CYCLE.....	7
2.4 Challenges of Data Mining	9
2.4.1 Mining information from different sources of data.....	9
2.4.2 Handling of different types of data.....	9
2.4.3 Efficiency and scalability of data mining algorithms	10
2.4.4 Expression of various kinds of data mining requests and results	10
2.4.5 Interactive mining knowledge at multiple abstraction levels	10
2.4.6 Protection of privacy and data security	11
2.4.7 Usefulness, certainty, and expressiveness of data mining results.....	11
2.5 The Architecture of Data Mining System	12
2.6 Data mining tasks.....	13
2.7 Frequent itemset mining.....	15
Chapter 3 Parallel Data Mining.....	18
3.1 Parallel data mining	19
3.2 Technologies of parallelism	19
3.2.1 Parallel Computing	19
3.2.2 DISTRIBUTED COMPUTING.....	20
3.2.3 Grid Computing	20
3.3 Algorithms of parallel data mining	21
3.4 Parallel Design Space.	21
3.4.1 Distributed Memory Machines vs. Shared Memory Systems.....	22
3.4.2 Task vs. Data parallelism.....	23

3.4.3 Static vs. Dynamic Load Balancing	23
3.4.4 Horizontal vs. Vertical Data Layout.....	24
3.4.5 Complete vs. Heuristic Candidate Generation	25
3.4.6 Candidate and Data Partitioning	25
3.5 How data is distributed across sites:	26
3.6 How distribute the workload over multiple processor:	28
3.6.1 Task-parallel algorithms:	28
3.6.2 Data-parallel algorithms:	30
Chapter 4 Sequential algorithm for finding frequent itemsets.....	31
4.1 Sequential algorithm for discovering frequent itemset	32
4.2 Frequent Itemsets Generation.....	33
4.3 Finding frequent itemsets using candidate generation: The Apriori algorithm	33
4.3.1 An example	34
4.4 Mining frequent itemsets without candidate generation: The FP-Growth algorithm	36
4.4.1 An example	36
4.5 Buddy prima algorithm	37
4.5.1 Prima representation operation	37
4.5.2 Buddy prima algorithm steps	38
4.5.3 An example	39
Chapter 5 Parallel algorithm for finding frequent itemsets	42
5.1 Introduction.....	43
5.2 Parallel apriori algorithm.....	43
5.3 Parallel buddy prima	44
5.4 Steps of Algorithm	45
5.4.1 The algorithm for master node.....	45
5.4.2 The algorithm for slave node	45
5.4.3 The weaknesses of this algorithm are	46
5.5 Proposed parallel algorithm	47
5.5.1 The algorithm	47
5.5.1.1 The algorithm for master node.....	47
5.5.1.2 The algorithm for slave node.....	48
5.5.2 Pseudo code.	50

5.5.2.1 Master node.....	50
5.5.2.2 Slave node.....	50
5.6 Case study.....	52
5.7 Performance evaluation.....	55
Chapter 6: Conclusion and Future Work	69
6.1 Conclusion.....	70
6.2 Future Work.....	71
References:	73
Appendix A : Pseudo code for apriori and FP growth algorithms.....	77

List of Tables

Table 3-1 project table example which has four attribute	26
Table 3-2 how Table is partitioned horizontally.....	27
Table 3-3 how Table is partitioned horizontally.....	27
Table 3-4 how table is partitioned vertically.....	27
Table 3-5 how table is partitioned vertically.....	27
Table 4-1 transaction database example.....	32
Table 4-2 a list of data set of transaction included 5 items.....	34
Table 4-3 Mining of fp-tree.....	37
Table 4-4 each item in table 4-1 and its equivalent prime number.....	39
Table 4-5 all transaction in table 4-1 and it's representation value.....	39
Table 5-1 sample of transactions.....	52
Table 5-2 each item in table 5-1 and its prime number.....	52
Table 5-3 each transaction in table 5-1 and its representation by multiplied prime number	53
Table 5-4 performance Time of the two thread algorithm in Test 1-1.....	57
Table 5-5 performance Time of the three thread algorithm in Test 1-2.....	58
Table 5-6 performance Time of the four thread algorithm in Test 1-3.....	58
Table 5-7 performance Time of the five thread algorithm in Test 1-4.....	59
Table 5-8 performance Time of the two thread algorithm in Test 2-1.....	59
Table 5-9 performance Time of the three thread algorithm in Test 2-2.....	60
Table 5-10 performance Time of the four thread algorithm in Test 2-3.....	60

Table 5-11 performance Time of the five thread algorithm in Test 2-4.....	61
Table 5-12 performance Time of the two thread algorithm in Test 3-1.....	61
Table 5-13 performance Time of the three thread algorithm in Test 3-2.....	62
Table 5-14 performance Time of the four thread algorithm in Test 3-3.....	62
Table 5-15 performance Time of the five thread algorithm in Test 3-4.....	63
Table 5-16 performance Time of the two thread algorithm in Test 4-1.....	63
Table 5-17 performance Time of the three thread algorithm in Test 4-2.....	64
Table 5-18 performance Time of the four thread algorithm in Test 4-3.....	64
Table 5-19 performance Time of the five thread algorithm in Test 4-4.....	65
Table 5-20 performance Time of the proposed algorithm in Test 1.....	66
Table 5-21 performance Time of the proposed algorithm in Test 2.....	66
Table 5-22 performance Time of the proposed algorithm in Test 3.....	67
Table 5-23 performance Time of the proposed algorithm in Test 4.....	67

List of Figures

Figure 2-1 Data mining life cycle.....	7
Figure 2-2 Architecture of data mining system.....	12
Figure 2-3 shows a small data set transaction.....	16
Figure3-1shows The classification tree of Four major classes of parallel data mining implementations.....	28
Figure 3-2 divide-and-conquer strategy, an overall recursive program structure.....	29
Figure 3-3 A sample illustration of a task queue.....	30
Figure 4-1 Apriori example	35
Figure 4-3 FP growth example	36
Figure 4-4 Conditional FP-tree of I3.....	37
Figure 5-1 Model of transaction set at node1.....	49
Figure 5-2 Transaction set model at node 1 in case study.....	53
Figure 5-3 The performance for calculating the support count for each item and remove infrequent one	56
Figure 5-4 Performance Time of the two thread algorithm in Test1-1.....	57
Figure 5-5 Performance Time of the three thread in Test 1-2	58
Figure 5-6 Performance Time of the four algorithm in Test 1-3	58
Figure 5-7 Performance Time of the five algorithm in Test 1-4.....	59
Figure 5-8 Performance Time of the two thread algorithm in Test 2-1.....	59
Figure 5-9 Performance Time of the three thread algorithm in Test 2-2.....	60
Figure 5-10 Performance Time of the four thread algorithm in Test 2-3.....	60
Figure 5-11 performance Time of the five thread algorithm in Test 2-4	61

Figure 5-12 performance Time of the two thread algorithm in Test 3-1	61
Figure 5-13 performance Time of the three thread algorithm in Test 3-2	62
Figure 5-14 performance Time of the four thread algorithm in Test 3-3	62
Figure 5-15 Performance Time of the five thread algorithm in Test 3-4	63
Figure 5-16 performance Time of the two thread algorithm in Test 4-1.....	63
Figure 5-17 performance Time of the three thread algorithm in Test 4-2.....	64
Figure 5-18 performance Time of the four thread algorithm in Test 4-3.....	64
Figure 5-19 performance Time of the five thread algorithm in Test 4-4.....	65
Figure 5-20 performance Time of the proposed algorithm with different number of threads in Test 1	66
Figure 5-21 performance Time of the proposed algorithm with different number of threads in Test 2	67
Figure 5-22 performance Time of the proposed algorithm with different number of threads in Test 3.....	67
Figure 5-23 performance Time of the proposed algorithm with different number of threads in Test 4.....	68

Chapter 1

Chapter 1

INTRODUCTION

1.1 Background

Database has been used in business management, government administration, scientific and engineering data management and many other important applications.

Data mining is an important part of computer science; it aims on finding information from database that is not already known by users of various application areas. The ever-growing amounts of data makes mining in most of them indispensable since the data sets often contain valuable information that cannot be obtained manually.

Frequent itemset mining is a popular part in the area of data mining with the goal of finding values or items that co-occur frequently together in a data set.

In many of applications, the response time of the mining process is important because the process is often executed and the results are required as soon as possible.

In general, there exist a large number of well-designed itemset mining algorithms that are very efficient on small data sets. However, most of them cannot be applied on large data sets because they do not take advantage of the full spectrum of available parallelism provided by modern processors.