ASCF: Apriori Algorithm on Spark Based on Cuckoo Filter Structure

By

Bana Ahmad Alrahwan

A Thesis Submitted to the
Faculty of Engineering at Cairo University
In Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
In
Computer Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2018

ASCF: Apriori Algorithm on Spark Based on Cuckoo Filter Structure

By

Bana Ahmad Alrahwan

A Thesis Submitted to the
Faculty of Engineering at Cairo University
In Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
In
Computer Engineering

Under the Supervision of

Prof. Dr. Elsayed Hemayed

Professor
Computer Engineering
Faculty of Engineering, Cairo University

Dr. Mona F. Ahmed

Assistant Professor
Computer Engineering
Faculty of Engineering, Cairo University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2018

ASCF: Apriori Algorithm on Spark Based on Cuckoo Filter Structure

By

Bana Ahmad Alrahwan

A Thesis Submitted to the
Faculty of Engineering at Cairo University
In Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
In
Computer Engineering

Approved by the
Examining Committee

_____
Prof. Dr. **Elsayed Hemayed** Thesis Main Advisor

_____
Prof. Dr. **Magda Fayek** Internal Examiner

_____
Prof. Dr. **Reda Abdelwahab** External Examiner
(Faculty of Computer and Information, IT Department)

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2018

Engineer's Name:	Bana Ahmad Alrahwan
Nationality:	Syrian
Date of Birth :	3/01/1978
E-mail:	bana.alrahwan-1@eng1.cu.edu.eg
Phone:	01092799738
Address:	29 Fahmy Street, Almanial
Registration Date:	1/03/2014
Awarding Date:	…./…./2018
Degree:	Master of Science
Department:	Computer Engineering

Supervisors:

Prof. Dr. Elsayed Hemayed
Dr. Mona F. Ahmed

Examiners:

Prof. Dr. Elsayed Hemayed (Thesis Main Advisor)
Prof. Dr. Magda Fayek (Internal Examiner)
Prof. Dr. Reda Abdelwahab (External Examiner)
(Faculty of Computer and Information, IT Department)

Title of Thesis:

ASCF: Apriori Algorithm on Spark Based on Cuckoo Filter Structure

Key Words:
Data mining, Big Data, Apache Spark, Apriori algorithm, Cuckoo filter.

Summary:

The Apriori algorithm is one of the most basic techniques that are used to discover frequent patterns in dataset. Apriori is iterative and works sequentially. It generates candidate sets having all possible combinations for frequent itemsets that are generated from the previous iteration and comparing each combination of items with every transaction record in every iteration. Thus, Apriori algorithm is not efficient and gets computationally more expensive as the data size is increased. The rapid growth of data necessitates running the data intensive algorithms in parallel distributed environment to achieve convenient performance. Many approaches have been proposed to solve the Apriori major drawbacks that severely degrade the performance as the datasets get larger which is a common feature in Today's data. In this thesis, Apriori Algorithm on Spark based on Cuckoo Filter structure (ASCF) is introduced. ASCF succeeds in removing the candidate generation step from Apriori algorithm to reduce computational complexity and avoid costly comparisons. The proposed algorithm is implemented on spark in-memory processing distributed environment to reduce processing time. The ASCF offers great improvement in performance over other implementation approaches of Apriori algorithm based on spark.

Acknowledgments

Today, I have completed my master thesis, I would like to take this opportunity firstly and foremost thank **Allah** for supplied me with the courage, guides me to the right path and has helped me to complete this thesis. Without Him, I would not have had the wisdom or the physical ability to do so.

I cannot forget the ideal man of the world and most respectable personality for whom **Allah** created the whole universe, **Prophet Mohammed** (Peace Be Upon Him).

This thesis would not have been possible without the help, encouragement and support of a number of wonderful individuals during the journey of preparing this thesis — So, I have to give them all thanks and appreciation.

I would like to thank my supervisor **Prof. Dr. Elsayed Hemayed** for his continuous efforts and concern. He has been always willing to help and give his best suggestions.

Thanks Dr. Elsayed

I would like to thank my gratitude to my supervisor **Dr. Mona Farouk** for suggesting the research path, her consistent guidance, advices and optimism. She has given me a confidence in my abilities, and provided me a strong supports to successfully complete my thesis. She has been very keen on my progress in this thesis. I greatly appreciate everything she has done for me.

Thanks Dr. Mona

I would like to express my deep and sincere gratitude to my family for their continuous and unparalleled love, help and support they provided me through my entire life. I dedicate my achievement to them.

My Beloved father, your absence had broken my heart and lost me the meaning of life, but your spirit still forever dwell in my heart. Your love surrounds me with every breath I take. You had taught me bravery, patience and sacrifice. You had always been my best friend, my greatest teacher. Today, I complete my career which he encouraged me to started it. I miss him so much, and I wish if he was with me. My father will remain a crown to be proud of all my life. **You were and will always be the greatest father**.

My Dear mother, I can barely find the words to express all the wisdom, love and support you've given me. Your being in my life is the secret of my success and happiness. Thanks for everything you have given me. Thank you for being my mother. **I Love You.**

My Brothers, Engineer Mustafa who is like my father in his tenderness and his sacrifice and **Dr. Mohammad** who accompanied me during my studies journey, he was my best friend. Thank you for your continuous support.

My Sisters, **Fatimah, Engineer Lina, Razan and Dr. Ghofran**, thank you for your encouraging me towards success.

Thanks **My Family** without their love, encouragement and support, I would not have been able to achieve my goals. Thank you from the bottom of my heart. **I love you all.**

I would like to thank my **all Friends** for their help and offer me advice.

Thanks for **everything** that helped me get to this day.

Thanks for **Faculty of Engineering.**

Thanks **Egypt.**

Table of Contents

List of Tables

List of Figures

List of Acronyms

Acronym	Definition
ARM	Association Rule Mining
ASCF	Apriori Algorithm on Spark based on Cuckoo Filter structure
HDFS	Hadoop Distributed File System
HFIM	a Spark-based Hybrid Frequent Itemset Mining algorithm for big data processing
KDD	Knowledge Discovery in Databases
LHS	Left Hand Side
R-Apriori	An Efficient Apriori based Algorithm on Spark
RDD	Resilient Distributed Dataset
RHS	Right Hand Side
YAFIM	(Yet Another Frequent Itemset Mining) A Parallel Frequent Itemset Mining Algorithm with Spark

Abstract

Data mining is the process that is used for extracting interesting patterns from large amount of data using a variety of techniques. One of the techniques that help to discover important relations between variables in large dataset is Association rule mining (ARM). It is used to identify strong rules discovered in databases. To build the association rules for all itemsets this requires large memory and processing resources. So, only frequent itemsets are considered to reduce the number of itemsets. There are many frequent itemset mining algorithms; the most popular algorithm is the Apriori algorithm. The Apriori algorithm is iterative and works sequentially, its normal execution is on a single machine. Nowadays, there is a great explosion of data; the rapid growth of data necessitates running the data intensive algorithms in parallel distributed environment to achieve convenient performance. The major drawbacks of the Apriori algorithm concerning its computational complexity make the algorithm inefficient to use when the data size is getting larger. In this thesis, Apriori Algorithm on Spark based on Cuckoo Filter structure (ASCF) is introduced. The ASCF algorithm solves the inherent drawbacks in the original Apriori algorithm. It succeeds in removing the candidate generation step from Apriori algorithm to reduce computational complexity and avoid costly comparisons, and uses cuckoo filter to further enhance the performance. The proposed algorithm is implemented on spark in-memory processing distributed environment to reduce processing time. It offers great improvement in performance over other previous approaches of Apriori algorithm implementation based on spark. The ASCF on a cluster of 4 nodes achieves a time of only 5.8% of the competing approach on the Retail dataset with minimum support of 0.75% , 25.6% on Chess dataset with minimum support of 85% and 37.3% on T10I4D100K with minimum support of 0.25%.

Chapter 1 : Introduction

We live in the data age, this data is being generated by everything around us at all times: from social sites, sensors, search engine, medical reports, etc. This huge amount of data is known as 'Big Data'. It is as a collection of large, diverse, complex dataset that comes at high speed. There is an urgent need to assist humans in extracting useful information (knowledge) from this data. It requires new architecture, techniques, algorithms, and analytic methods to manage it and extract value and hidden knowledge from it. Big Data mining is the process of discovering patterns from large datasets that have those mentioned properties.

1.1. Problem Statement

The problem can be formally stated as:
Let D be a set of transactions in large transactional data where each transaction T is a set of items such that $T \subseteq I$, where $I = \{i1, i2, \cdots, im\}$is a set of items. Each transaction has a unique identifier TID. Let X and Y be two different sets of items, where $X \subseteq I$, $Y \subseteq I$ are subset of I and $X \cap Y = \varnothing$. IF $X \subseteq T$ then the transaction T is said to contain X.

Each itemset appears a number of times in D so there are a percentage of transactions that contain all items in X, which is called support(X). The itemsets X and Y appear together in D if there are S percentage transactions that contain $X \cup Y$, where S is the support($X \cup Y$), we say there is association between itemsets X and Y.

The problem is discovering important association rules between variables (items) in large transactional dataset. These rules are called strong rules if they have support and confidence greater than or equal predefined minimum support and confidence thresholds.

Mining association rule task consist of two steps:

- Finding all frequent itemsets which have support greater than or equal a predefined minimum support
- Generating confident rules from the frequent itemsets discovered.

The performance of an algorithm for mining association rules is determined by the first step, while the second step is straightforward. Most of the algorithms are inefficient and take a lot of time when working with large datasets because they are designed to work with a single machine in an iterative manner with no natural way for distribution. So, new technologies are required to store and process large data sets in a distributed computing environment like Hadoop Mapreduce and Spark.

1.2. Motivation

The Apriori algorithm is iterative and works sequentially. Its normal execution is on a single machine whose speed leads to severe performance degradation when working with big data so multiple machines and a parallel algorithm are needed where issues like data replication and synchronization must be addressed. The major drawbacks of the Apriori algorithm concerning its computational complexity make the algorithm inefficient to use when the data size get larger. It generates candidate sets having all possible pairs from frequent itemsets that are generated from previous iteration and compares each pair with every transaction record in every iteration.

In this thesis, we apply Apriori Algorithm on Spark based on Cuckoo Filter structure (ASCF) to solve these problems. We propose a parallel Apriori implementation on Spark distributed environment to reduce processing time, and use cuckoo filter structure to further enhance the performance. We focus on the step that is the most time consuming in the whole algorithm, which is generating candidate sets in the second phase and solve this problem by eliminating it to reduce computational complexity and avoid costly comparisons.