International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Analysis of Frequent Item set Mining of Electronic Evidence using ISPO Tree based on Map/Reduce

Pranav Kumar Bhadane[1], R. V. Patil[2]

[1, 2]Department of Computer Engineering, SSVPS"s BS Deore College of Engineering, Dhule, 424005, India

Abstract: ***Association rules can mine the relevant evidence of computer crime from the massive data and association rules among data item set, and further mine crime trends and connections among different crimes. They can help detect and leads case policies and prevent crime with given criterions. Frequent item set mining (FIM) plays a fundamental Associations, correlations and electronic evidence analysis area like many real-world data mining areas. FP-growth pattern of constant search is the most famous FIM algorithm. Incrementing data, time and space costs FP-growth will be mining algorithms bottleneck. Information and communication technologies in the world, with rapid advancements in the crimes committed are becoming technologically intensive. Use digital devices when crime, forensic examiners and practical frameworks which can pose as evidence to recover the data for analyzing the methods to adopt. Data Generation, Data Warehousing and Data Mining, are the three essential features involved in the investigation process. So that we proposed a novel parallelized algorithm called PISPO based on the cloud-computing framework MapReduce, which is widely used to cope with large scale data and captures both the content and state to be distributed to the changed and original of the transactions dataset to SPO tree.***

Keywords: Data mining, ISPO tree algorithm, Minimum support, Threshold value, Electronic evidence, Map/Reduce approach, Frequent itemset mining, Association rules

1. Introduction

The digital world has penetrated every aspect of today"s generation, both in the space of human life and mind, not even sparing the criminal sphere of the world. According to Jim Christy, Director of Cyber Crime Institute, forensic science is the application of science to legal process and therefore against crime. In Science and technology, and in fact or in a Court of law of evidence relating to the use of in the process. When crime is aided by or digital device (s), including forensic investigations using digital or cyber forensic categorized under. If only one computer or digital storage medium is digital tools, as we check in computer forensics. Computer forensic (a.k.a. digital forensic)forensic science, whose goal is to explain the current state of digital artifact is a branch. Digital Forensic Research Workshop (DFRWS) [13] has defined Digital Forensic Science as "the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations". Digital forensic science covers computer forensics, network forensics, disk, firewall forensics, forensics, database device, mobile device forensics, software forensics, live system forensics, etc. [12]
The key contribution of this research is proposing and developing a novel tree structure for maintaining frequent patterns about electronic evidence in an incremental dataset.

We offer algorithm ISPO-tree (single pass ordered tree)based on both content and an innovative approach for parallelizing Map Reduce FP-FP-growth algorithm for a tree change also have captured the State of transactions in the dataset that intelligently on a large scale mining operations and functions in computational free shards MapReduce jobs map. These computer failures with the ability to start from the tree can achieve near linear speedup. Although the improved Single Pass Ordered tree is that it builds associate in nursing economical single pass tree structure for FP-tree primarily based incremental mining. The algorithmic rule supported the character of electronic proof, that not solely improves the potency of the algorithmic rule however even be convenient toupdate association rules in real time as proof been inserted, deleted or updated [4].

2. Literature Survey

In this section we discussed about literature survey on Frequent Item Set mining.

[1] L. Zhou, Z. Y. Zhong, and J. Chang, et al, Frequent itemset mining (FIM) plays an essential role in mining associations, correlations and many other important data mining tasks. Unfortunately, as the volume of dataset gets larger day by day, most of the FIM algorithms in literature become ineffective due to either too huge resource requirement or too much communication cost. In this paper, they propose a balanced parallel FP-Growth algorithm BPFP, based on the PFP algorithm, which parallelizes FP-Growth in the MapReduce approach. BPFP adds into PFP load balance feature, which improves parallelization and thereby improves performance. Through empirical study, BPFP outperformed the PFP which uses some simple grouping strategy.

[2]S. K. Tanbeer, C. F. Ahmed, and B. S. Jeong, et al, FP-growth algorithm using FP-tree has been widely studied for frequent pattern mining because it can give a great performance improvement compared to the candidate generation-and-test paradigm of Apriori algorithm. However, it still requires two database scans which are not applicable to processing data streams. In this paper, we present a novel tree structure, called CP-tree (Compact

Pattern tree), that captures database information with one scan (*Insertion phase*) and provides the same mining performance as the FP-growth method (*Restructuring phase*) by dynamic tree restructuring process. Moreover, CP-tree can give full functionalities for interactive and incremental mining. Extensive experimental results show that the CP-tree is efficient for frequent pattern mining, interactive, and incremental mining with single database scan.

[3] C. K. S. Leung, Q. I. Khan, and Z. Li, et al, since its introduction, frequent-pattern mining has been the subjectof numerous studies, including incremental updating. Many existing incremental mining algorithms are Apriori-based, which are not easily adoptable to FP-tree-based frequent-pattern mining. In this paper, we propose a novel tree structure, called Can-Tree(canonical-order tree), that captures the content of the transaction database and orders tree nodes according to some canonical order. By exploiting its nice properties, the Can-Tree can be easily maintained when database transactions are inserted, deleted, and/or modified. For example, the Can-Tree does not require adjustment, merging, and/or splitting of tree nodes during maintenance. No rescan of the entire updated database or reconstruction of a new tree is needed for incremental updating. Experimental results show the effectiveness of our Can-Tree in the incremental mining of frequent patterns. Moreover, the applicability of Can-Trees is not confined to incremental mining; Can-Trees can also be applicable to other frequent-pattern mining tasks including constrained mining and interactive mining.

[4] Y. S. Koh, and G. Dobbie, et al, since the introduction of FP-growth using FP-tree there has been a lot of research into extending its usage to data stream or incremental mining. Most incremental mining adapts the Apriori algorithm. However, we believe that using a tree based approach would increase performance as compared to the candidate generation and testing mechanism used in Apriori algorithm. Despite this FP-tree still requires two scans through a dataset. In this paper we present a novel tree structure called Single Pass Ordered Tree SPO-Tree that captures information with a single scan for incremental mining. All items in a transaction are inserted or sorted based on their frequency. The tree is reorganized dynamically when necessary. SPO-Tree allows for easy maintenance in an incremental or data stream environment.

3. Proposed Approach Framework and Design

3.1 Problem Definition

One of the existing incremental frequent pattern mining algorithms called SPO-tree can perform incremental mining by a single scan for incremental mining. But how to apply this algorithm to the analysis of electronic evidence more effectively will become the focus of this paper. In the past research, little people take care of the itemmined to the frequent item needing to update or inserted a little data. The past algorithms are not suit for this problem especially in forensic area. So, in this paper, we propose a novel parallelized algorithm called PISPO based on the cloud-computing framework Map/Reduce, which is widely used to cope with large-scale data and captures both the content and state to be distributed to the changed and original of the transactions dataset to SPO-tree.

Aims and Objective

In this project we have main aim is to present the extended scalable technique to analyze and find frequent item set of electronic evidence.

Apart from this below are the main objectives of this project:
- To present literature review for finding frequent item set.
- To present the practical simulation of proposed algorithms and evaluate its performances.
- To present the comparative analysis of existing and proposed algorithms in order to claim the efficiency of our proposed system.

3.2. Methodology

3.2.1 Edge Detection Algorithm

We developed an efficient Content Based Image Retrieval (CBIR) system using Sobel's edge detection algorithm. Content Based Image Retrieval (CBIR) is a process to retrieve a stored image from database by supplying an image as query instead of text. This can be done by proper feature extraction and querying process. A universal content based image retrieval system uses color, texture and shape based feature extraction techniques for better matched images from the database. In proposed CBIR system, shape features are used. Edge detection is a fundamental tool in image processing and computer vision. To do analysis of the shape of image there are different techniques one way is that first upon finding out edges of respective image and then matching the shape of identified images. We use the Prompt edge detection method to detect edge points, these edge points are detected using the Sobel edge detection algorithm. These features are then compared to the features of the images which are already stored in our imagedatabase and most similar images are retrieved [15].

What are Edges in an image?
- Edges are significant local changes of intensity in an image.
- Edges typically occur on the boundary between two different regions in an image.
- Using edges we can produce a line drawing of a scene from an image of that scene.
- Important features such as (corners, lines, curves) can be extracted from the edges of an image.
- These features are used by higher vision computer algorithms for further processing.

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking

for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image to find edges. Edge detection isa necessary preprocessing step in most of computer vision and image understanding systems. The accuracy and reliability of edge detection is critical to the overall performance of these systems. Earlier researchers paid a lot of attention to edge detection, but up to now, edge detection is still highly challenging. There exist several methods for edge detection such as Sobel, Robert, Prewitt, Canny, etc. These methods have been proposed for detecting transition in images. In this project we are going to retrieve images basedon the features of the images. These features can be color, edge or texture of the image. As edge being a prominent feature of an image, we are extracting edges as the features of the image using Sobel's edge detection algorithm. The Sobel operator performs a 2-D spatial gradient measurement on images. It uses a pair of horizontal and vertical gradient matrices whose dimensions are 3×3 for edge detection operations. Standard Sobel operators, for a 3 × 3neighborhood, each simple central gradient estimate is vector sum of a pair of orthogonal vectors. Each orthogonal vector is a directional derivative estimate multiplied by a unit vector specifying the derivative‟s direction. The vector sum of these simple gradient estimates amounts to a vector sum of the 8 directional derivative vectors. The directional derivative estimate vector G was defined such as density difference/distance to neighbor. This vector is determined such that the direction of G will be given by the unit vector to the approximate neighbor. Note that, theneighbors group into antipodal pairs:
(a,i), (b,h), (c,g), (f,d). The vector sum for this gradient estimate:
G=((c - g)/R.[1,1]/R) + ((a – i)/R.[-1,1]/R) + (b – h).[0,1]
+ (f - d).[1,0]
Where, R = 2. This vector is obtained asG = [(c − g − a + i) / 2 + f − d, (c − g + a − i) / 2 + b − h]

Here, this vector is multiplied by 2 because of replacing thedivide by 2. The resultant formula is given as follows:
G ′ = 2.G = [(c − g − a + i) + 2.(f − d), (c − g + a − i) + 2.(b − h)]

The following weighting functions for x and y components were obtained by using the above vector. This sobel operator consists of a pair of 3×3 convolution kernels as shown infigure.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

One kernel is simply the other rotated by 90°.These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation(call these Gx and Gy). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{G x2 + Gy2}$$

Typically, an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

This is much faster to compute. The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\Theta = \arctan (Gy / Gx)$$

Content Based Image Retrieval is designed to retrieve the images by giving a image as an input instead of any text as in any web browsers. The aim behind it is to retrieve the perfect images which are actually similar to the required query image. In the proposed system, software is divided into two parts-user and admin. Admin part will input various images along with their details in the database. The features of these images are extracted and stored in the database. On other hand, when the user gives input as a query image, the features of that image are extracted and stored in the feature vector. Then thecomparison is done between the feature vector and the feature database. Later indexing is applied on the images and the most similar images are displayed in the field of retrieved images.

3.2.2ISPO_Tree (DB, Ɛ, Htable) Algorithm

Define and clear Htable:H[];
1. **foreach** Transaction Ii in DB **do**
2. **if** flag==0 and D_Trans==0 in Ii **then do**
3. **foreach** Item aj in Ti **do**
4. H[ai].count++;
5. **else if** flag==0 and D_Trans!=0 in li **then do**
6. **foreach** Item aj in D_Trans **do**
7. **if** „+“ in D_Trans then **do**
8. H[ai].count++;
9. **else if** „-“ in D_Trans **then do**
10. H[ai].count--; Define and dear the root of ISPOaree ; r;
11. **Call** ConstructTree(ai, r)
12. **end**
13. **else if** flag in Ii==1 then **return**;
14. Make Ii ordered according to H; H[ai]=d; the next step(I1, I2,…,Ii-1);
15. **If** the next step(Ii)>=k then **do**
16. Resorting ReconstructTree(Ii-1, r);
17. **end**
18. **foreach** item ai in Ii **do**
19. **Call** Growth(r,ai);
20. **End**

3.2.3 Map/Reduce Approach:
MapReduce approach following three main steps
1. Map function
2. Shuffle function
3. Reduce function

Map function
Map Function is the first step in MapReduce Approach. It takes input tasks (say DataSets. I have given only one DataSet in below diagram) and divides them into smaller

sub-tasks. Then perform required computation on each sub-task in parallel.
This step performs the following two sub-steps:
1. Splitting
2. Mapping

Splitting step takes input DataSet from Source and divides into smaller sub DataSets.

Mapping step takes those smaller sub DataSets and performs required action or computation on each sub DataSet.The output of this Map Function is a set of key and value pairs as <Key, Value>.

Shuffle Function

It is the second step in MapReduce Algorithm. Shuffle Function is also known as "Combine Function". It performs the following two sub-steps:
1) Merging
2) Sorting
It takes a list of outputs coming from "Map Function" and performs these two sub-steps on each and every key-value pair. Merging step combines all key-value pairs which have same keys (that is grouping key-value pairs by comparing "Key"). This step returns <Key, List<Value>>.

Sorting step takes input from merging step and sorts all key-value pairs by using Keys. This step also returns <Key, List<Value>> output but with sorted key-value pairs.

Finally, Shuffle Function returns a list of <Key, List<Value>> sorted pairs to next step.

Reduce function

It is the final step in MapReduce Algorithm. It performs only one step: Reduce step.

It takes list of <Key, List<Value>> sorted pairs from Shuffle Function and perform reduce operation.
Final step output looks like first step output. However final step <Key, Value> pairs are different than first step <Key, Value> pairs. Final step <Key, Value> pairs are computed and sorted pairs.

We can observe the difference between first step output and final step output with some simple example.

3.3 Proposed Work

1) We have to select dataset which is in .sql form to upload.
2) Dataset containDetail information about the criminal with his name,id,address,images etc. Thatinformation is considered as electronic evidence of criminal.
3) After uploading dataset in system, that dataset is stored in system in from of tree structure using association rule.
4) When user have some evidence of crime and they want to find out suspected criminal of the crime then user will upload that evidence in system for search query.
5) Then that evidence is match with the available records using various techniques.
6) Actual searching of the dataset is done with
7) MapReduce approach.
8) If any match found then that particular record is given as result of the search query.
9) Record contains detail profile of the criminal and that criminal may be suspected criminal for that particular crime.
10) Stop.

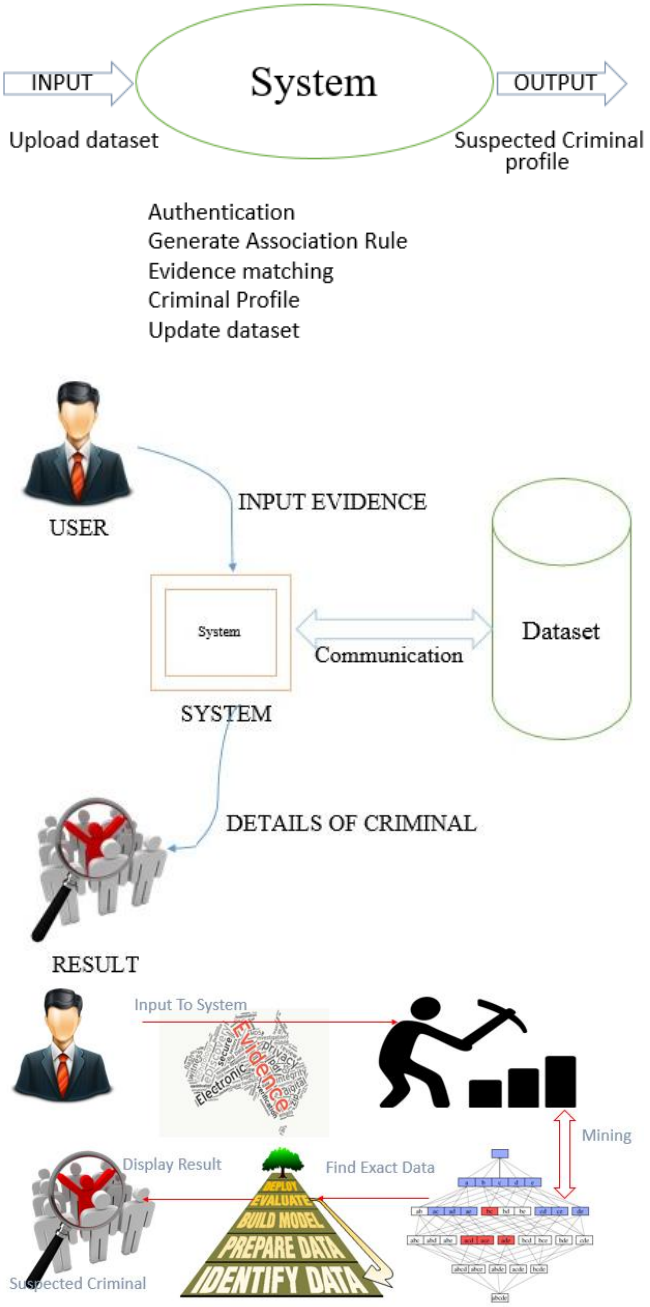

Figure 1: Architecture Diagram

The above figure shows the general model of the Content-based Image Retrieval system. The above system has been divided into two phases. In the first phase image is fired as the query image which is to be searched and in the other phase the collection of images is maintained in the database from which the query image or the input image is to be compared. In input module, the feature vector is extracted from each input image and stored into the image database with its input image. When query image is entered into the query module, the feature vector of the query image is

extracted. In retrieval module, the extracted feature vector of query image is compared with the images stored in the database. Similar images are retrieved according to their similarity with the query image. Finally the target image will be obtained from the retrieved images. On the second level of the system the features extraction of images is carried out. The features are generally color, texture and edges of the images. Mostly color and e features are edge features are used for image retrieval. In our case we are going to use edge features of an image for comparison. These edge features are extracted using Sobel's edge detection method.

Semantic analysis Approach for Analysis, Frequent Password, Keywords Search(Textual Data analysis):

WordNet is a lexicon of the English language that also captures the semantic relationships between words. It also contains information about different senses of words, combines synonyms into structures called synsets, and facilitates the processing of these features via an API. WordNet is made freely available for processing and analysis from its developers / maintainers at Princeton University.

Of particular importance to us are the hypernym / hyponym relationships captured by WordNet [14]. We read query text file and segment the data in file by removing delimiters in data. Then each word is extracted from data. Each word is sent to JAWS system. The Java API for WordNet Searching (JAWS) is an API that provides Java applications with the ability to retrieve data from the WordNet database [14].

Procedure:

1) JAWS system gets two inputs in one word from query file and one word from a file in database.
2) JAWS return 2 arrays of sysnets each for one word(Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms).
3) We iterate through both array to find common word.
4) If common word found a counter in incremented(a counter is set for each file in database).
5) Then for same word in query file and next word of file from database is selected and 1 and 2 is repeated.
6) When all words from database files are over we select next word from query file and 1,2,3,4,5 repeated
7) At the end we find percentage word count matched
8) Procedure is repeated for each file in database.
9) This way matched file from database is retrieved.

3.4. Mathematical Model

Input: Frequent Item set

Let $I = \{I_1, I_2,\ldots,I_m\}$, be a set of data items, the element of the set is known as item. L set $= \{I_j,\ldots,I_k\} \subseteq I$ where $\leq kj$ and $1 \leq j$, $\leq nk$ is called an item-set. L transaction is $= _{id} BtT$),(where t_{id} is the transaction id and B is an item-set. If $\subseteq BA$ is an item-set, then A occurs in T. L transactional dataset D over I is a set of transactions and D is the number of transactions in the dataset. The support of an item-set A is theportion of transaction in the dataset that contains A.
So:

$$\sup(A) = \frac{count\,(A \subseteq D)}{|D|}$$

And

$$\sup(A \rightarrow B) = \frac{|\{A \cup B \subseteq T, T \subseteq D\}|}{|D|}$$

An item-set is frequent if its support is no less than a support threshold given by user called min-sup. An association rule isan implication of the form, that is:

$$confidence(A \rightarrow B) = \frac{|\{A \cup B \subseteq T, T \subseteq D\}|}{|\{A \subseteq T, T \subseteq D\}|}$$

4. Work Done

In this section we are discussing the practical environment, scenarios, performance metrics used etc.

Dataset

Dataset contain criminal record i.e. (criminal profile contain criminal name, criminal id, face image, finger image, address, some other evidence). As well as crime evidence in digital form. Crime evidence contain crime id, related information, suspected evidence etc.

4.1 Input

Input for our practical experiment is frequent item-set.

4.2 Hardware and Software Configuration

Hardware Requirements:
Processor :Intel i3
Ram : 4GB DD RAM
Hard Disk : 20 GB

Software Requirements:
Front End : Java
Tools Used : NetBeans, XAMMP Server
Operating System : Windows 7/8

XAMMP Server

We have used XAMMP server our project. XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server is extremely easy as well [11].

4.3 Results

The results compared here are time graph between existing and proposed system is shown in below figure.

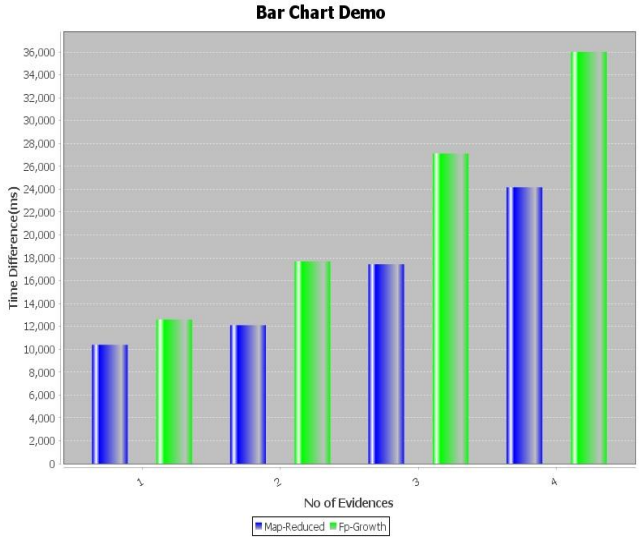

Figure 2: System Parameters Charts

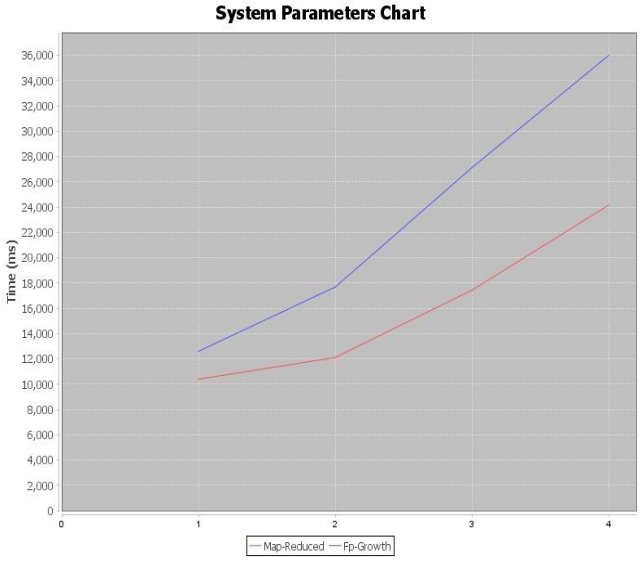

Figure 3: System Parameters Charts

5. Conclusion & Future Work

The improved algorithmic program named as PISPO supported the character of proof record, that required to updatea trifle someday, that not solely cut back the range of treebranches however additionally update the tree in real time. Once adding new listing or some record has been modified,flag ought to distribute the item for a prefix tree updates. Things and flags during a dealings are inserted into the tree supported a descendant order of frequency. The tree isreconstructed once the proportion of the edit distance of thingswithin the sorted order changes on top of a precise threshold.This algorithmic program relies on a completely unique knowledge and computation distribution theme whicheliminates communication among computers just about andmakes it attainable for United States with the MapReducemodel. We tend to incontestable that the algorithmic program is effective once on large knowledge scene ought to be mining. Our future work can extract and correlate the proof keep by the jpg, rmvb, doc, wvm so on supported non-relational information like MongoDB. For future work, we will try to implement parallel execution of this algorithm i.e. PISPO (Parallel Improved Single Pass Ordered Tree) tree structure algorithm

References

[1] L. Zhou, Z. Y. Zhong, and J. Chang, et al, "Balanced parallel FPGrowthwith MapReduce" IEEE Youth Conference on InformationComputing and Telecommunications, November, 2010, Beijing,China, pp. 243-246.
[2] S. K. Tanbeer, C. F. Ahmed, and B. S. Jeong, et al, "CP-Tree: a treestructure for single-pass frequent pattern mining," Advances inKnowledge Discovery and Data Mining, Springer, LNCS, Vol. 5012,2008, pp. 1022-1027.
[3] C. K. S. Leung, Q. I. Khan, and Z. Li, et al, "CanTree: a canonical-ordertree for incremental frequent-pattern mining," KnowledgeandInformation Systems, Vol. 11, No. 3, April 2007, pp. 287-311.
[4] Y. S. Koh, and G. Dobbie. "SPO-tree: efficient single pass orderedincremental pattern mining," Data Warehousing and KnowledgeDiscovery (DaWaK 2011), Springer, LNCS 6862, 2011, pp. 265-276.
[5] J. Dean, and S. Ghemawat, "MapReduce: simplified data processingon large clusters," Communications of the ACM (CACM), Vol. 51,No. 1, January 2008, pp. 107-113.
[6] S. L. Garfinkel, "Digital forensics research: the next 10 years,"Digital Investigation (the Proceedings of DFRWS '10), Vol. 7,Supplement, August 2010, pp. s64-s73.
[7] J. W. Han, J. Pei, and Y. W. Yin, "Mining frequent patterns withoutcandidate generation," Proceedings of the 2000 ACM SIGMOD, June, 2000, Dallas, TX USA, pp. 1-12.
[8] S. G. Totad, G. RB, and P. P. Reddy, "Batch processing for incremental FP-tree construction," International Journal of ComputerApplications, Vol. 5, No. 5, August 2010, pp. 28-32.
[9] Sankalp Mitra1, Suchit Bande2, Shreyas Kudale3, Advait Kulkarni4,Asst. Prof. Leena A. Deshpande, "Efficient FP Growth usingHadoop - (Improved Parallel FP-Growth)", International Journal ofScientific and Research Publications, Volume 4, Issue 7, July 2014 1ISSN 2250-3153.
[10] H. Y. Li, Y. Wang, and D. Zhang, et al, "Pfp: parallel fp-growth forquery recommendation," Proceedings of the 2008 ACM conferenceon Recommender systems (RecSys '08), October 23-25, 2008,Lausanne, Switzerland, pp. 107-114.
[11] https://en.wikipedia.org/wiki/XAMPP.
[12] Ajay, Rakesh Kumar, Dr. P.K. Jakhar, Dr. Anuj Kumar, "Use of Data Mining Techniques in Data Analysis for Digital Applications", (IJRST) 2012, Vol. No. 1, Issue No. IV, Jan-Mar.
[13] http://www.forensicswiki.org/wiki/Digital_Forensic_Research_Workshop.
[14] https://en.wikipedia.org/wiki/WordNet.
[15] O.R Vincent ,O. Folorunso A Descriptive Algorithm for Sobel Image Edge Detection 2009.