# PARALLELIZING PARM

**Akshat Bansal[1], Amitabh Tiwari[2], Jay Sheth[3], Amroz Siddiqui[4]**

[1,2,3,4] *Department of Computer Engineering, Fr.C.R.I.T, Vashi, Mumbai University (India)*

## ABSTRACT

*Data mining involves processing of large amounts of data and it involves multiple scans of data which leads to large amount of running time of algorithms. In order to reduce on running time of the algorithms, we can parallelize them. One of the ways of achieving parallelism is to setup a Linux cluster and make use of MPI library functions in the actual program code. Most of the data mining algorithms use textual data as dataset. However, PARM (Peano Count Tree Association Rule Mining Algorithm) is one of those algorithms that work on spatial dataset rather than text dataset and generates association rules for the same. An example of spatial data is a remote sensed imagery (RSI).*

***Keywords: Data Mining, Linux Clusters, Parallel Computing, PARM***.

## I. INTRODUCTION

There are many data mining activities like classification, clustering, association rule mining. While analyzing such huge data, equivalently huge amount of computing power is needed. Consider the data mining activity - association rule mining. Association rules initially used for Market Basket Analysis represent a relationship of the form X=>Y where X and Y are sets of items. X is known as the antecedent and Y is the consequence. There are two quality measures of association rules support and confidence. For a rule X=>Y the support *s*% represents the total transactions in the data set D in which contains both X and Y and confidence *c*% represents the total transactions in D which contain X also contain Y. The main aim of association rule mining is to find association rules which satisfy the minimum threshold specified by the user, minimum support and minimum confidence [1].

Apriori is one the association rule data mining algorithm. While applying the Apriori algorithm, the frequent item-sets must reside in the main memory during execution. For instance, at k-th iteration the k-th frequent item-sets must be in the main memory or some equivalent representation such as hash trees. But if the size of hash tree is bigger than what can be fit into the main memory then the hash tree must be partitioned. Even with highly efficient pruning techniques of Apriori algorithm, for a proper result on huge dataset the computing power required by this algorithm is provided only by parallel computing.

Spatial data such as Remote Sensed Imagery (RSI) is one of the promising areas of association rule mining. Such spatial data is collected from satellites on daily basis. Some of the application areas of association rules in RSI are: Disaster management, prediction of best crop yield, Air Traffic Control (ATC) for predicting traffic characteristics, traffic predictions. In this paper we would be using simple application to demonstrate the efficiency of PARM with parallel computing.

### 1.1 Problem Definition

Given multiple RSI images over time and the application domain, generate the association rules using Linux clusters in order to achieve a high speed robust mechanism.

## II. P-TREE

### 2.1 RSI Image Format

An RSI image can` be viewed as 2D array of pixels. Associated with each are various descriptive attributes called "bands" in remote sensing literature. The bands can be either visible reflectance bands or infrared reflectance bands and possibly some bands of data gathered from ground sensors. An RSI image as a relational table can be viewed as shown in Fig.1. Here we show the RSI image contents for agricultural applications.

| Pixel coordinates | Blue | Green | Red | NIR | MIR1 | MIR2 | TIR | Yield quantity | Nitrate levels |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| | | | | | | | | | |
| Primary Key | Visible reflectance bands | | | Infrared reflectance bands | | | | Data from ground sensors | |
| | All the values are scaled to the range 0-255 | | | | | | | | |

**Fig.1 RSI as a relational table**

Spatial data is generally organized in a format, called bSQ (bit sequential). A reflectance value in a band is a number in the range 0–255 and is represented as a byte. We split each band into eight separate files, one for each bit position. Each individual bit file is a bSQ file. These bSQ files are related to the "bit planes" in image processing. The various formats are shown in Fig.2.

### 2.2 P-Tree Creation

A P-tree is a quadrant-wise, Peano-order-run-length compressed, representation of each bSQ file. The ideas is to recursively divide the entire image into quadrants and record the count of 1 bits for each quadrant, thus forming a quadrant count tree.

| BAND-1 | | BAND-2 | |
|---|---|---|---|
| 254 (11111110) | 127 (01111111) | 37 (00100101) | 240 (11110000) |
| 14 (00001110) | 193 (11000001) | 200 (11001000) | 19 (00010011) |

bSQ format (16 files)

| B11 | B12 | B13 | B14 | B15 | B16 | B17 | B18 | B21 | B22 | B23 | B24 | B25 | B26 | B27 | B28 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

**Fig.2 RSI image formats [2]**

For example, given an 8 × 8 bSQ file, its P-tree is as shown in Fig.3. In this example, 39 is the number of 1s in the entire image, called root count. The root level is labelled as level 0. The numbers 16, 8, 15, and 0 at the next level (level 1) are the 1-bitcounts for the four major quadrants in raster order (upper left, upper right, lower left, lower right). Since the first and last level-1 quadrants are composed entirely of 1 bits (calledpure-1 quadrant) and 0 bits (call pure-0 quadrant), respectively, sub-trees are not needed and these branches terminate [3].

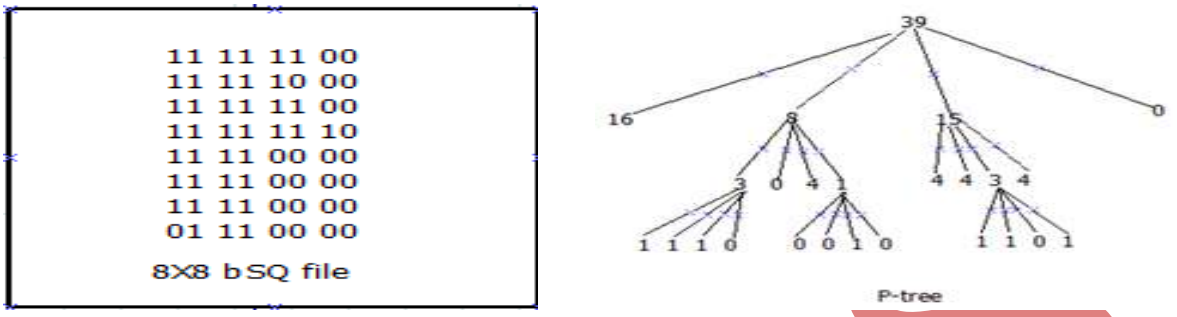


**Fig.3 Generating P-Tree**

## III. PARM ALGORITHM

The existing data mining algorithms like Apriori, Eclat, FP-growth, AprioriDP, OPUS Search algorithms lack the ability of handling multimedia data. These work very well with the textual data but when it comes to multimedia data like Remote Sensory Images i.e. the spatial data, they find it difficult to cope with the enormous quantity of data being collected on daily basis from the satellites. Here, we thus use PARM, which is capable of handling large spatial dataset.

The steps involved in PARM algorithm include the following:

**Step1:** The RSI data which is a 2D array of pixels is converted in bSQ (bit Sequential form) which represents a manner to organize spatial data.

**Step2:** The bSQ format is then converted into a P-tree which is an intermediate representation of the original spatial data.

**Step3:** Frequent item-sets called Asets are generated from the P-tree. From these sets the association rules are derived.

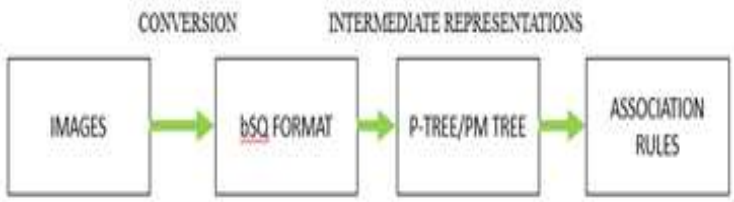


**Fig.4 steps in PARM**

The actual PARM algorithm is given as:

```
Procedure PARM
{
    Discretization;
    F1 = { c∈1-Asets | rootcount(c) >= minsup};
    For (k=2; Fk-1≠∅; k++) do begin
        Ck = p-gen(Fk-1);
        Fk = {c∈Ck | rootcount(c) >= minsup}
    end
    Answer = ∪k Fk
}
```

**Fig.5 PARM algorithm**

Here, Discretization divides the input dataset into smaller subsets while A-sets are similar to k-frequent itemsets of Apriori algorithm. Also, p-gen function works similar to the way k-frequent itemset generation function works in case of Apriori algorithm [2].

## IV. SYSTEM DESIGN

In the problem definition we have identified three problem domains and their solution sets, now let us move towards the solution set and identify how individual solution set is supposed to provide solutions to the problem domains. Before we go into details of the designing, let us define the system architecture and from there we can identify how each solution set will be embedded into the architecture. The following are the problem domains:

1. Image processing
2. Parallel computing
3. Data mining

### 4.1 Architectural Styles

Software architecture includes system decomposition, global control flow, handling of boundary conditions and inter-subsystem communication protocols. The concept of software architecture has emerged due to presence of interdisciplinary problem domains and embedding them to complete a system will lead to increasing the overall complexity of the subsystem. Here the specification of system decomposition covers all the problem domains [3].

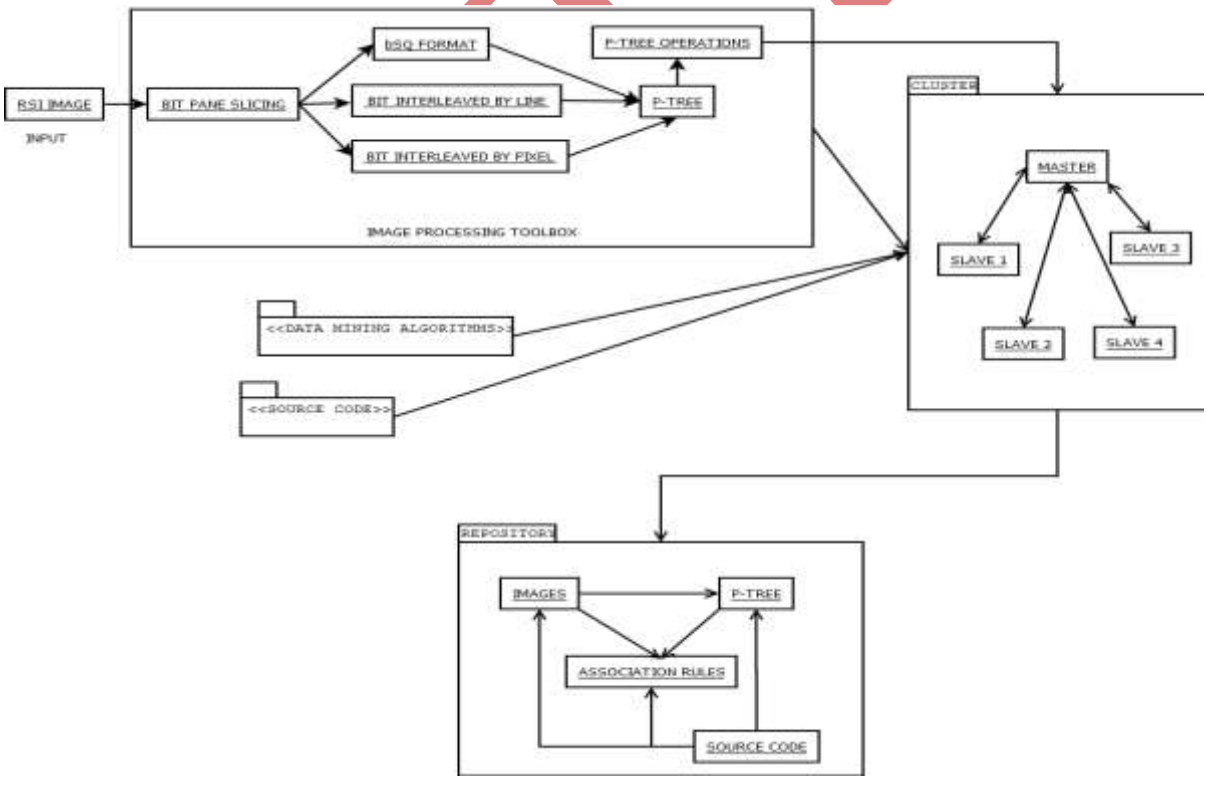


**Fig.6 System Architecture**

Here we will be using the repository architectural style, wherein the subsystems access and modify a single data structure called the central repository. Here the system architecture can be thought of as implementing global control flow through the clusters. The repository ensures that concurrent accesses are serialized. Such system are referred to as blackboard systems. The architecture consists of:

1. Standard data sources
2. Cluster
3. Repository

Here we explain each and every block in detail:

### 4.1.1 Standard Data Sources

Standard data sources include various forms in which the data is processed and stored in the repository. These deal with how the data is actually obtained rather than dealing with how these are actually stored in the repository. These are further divided into:

1. Image processing toolbox
2. Data mining algorithms
3. Source code

#### 4.1.1.1 Image Processing Toolbox

It is responsible for converting the input image into its equivalent P-tree with no loss, in other words lossless data equivalence must exist between the image and the intermediate representation developed. The various components involved in the image processing toolbox are:

1. Bit plane slicing
2. RSI image formats
3. P-tree
4. P-tree operations

Bit plane slicing is used to convert the input image into various planes wherein each plane contains values of various pixels. The bit planes obtained are then converted to any of the RSI image formats like bSQ, Bit interleaved by line or bit interleaved by pixel. From these formats the image is then converted into the data structure P-tree. On this data structure then various operations are performed. P-tree represents the input image faithfully. This intermediate representation of P-tree is lossless. Fig.7 represents the image processing toolbox.

#### 4.1.1.2 Data Mining Algorithms

These include algorithms like PARM or T-cube which are used to generate meaningful rules from the images. The input to the data algorithm is nothing but the P-tree obtained from the image processing toolbox. A general question arises why is there a need for passing the images through the clusters why not directly store them into the repository? The output of data mining algorithm are as follows:

1. Identification of attributes.
2. Generation of rules on the basis of attributes.
3. Generating relative significance of attributes.
4. Learning achieved through data mining algorithms.
5. Predictions for a new image.

The learning here deals with dynamic decisions about support and confidence in order to decide sufficient number of rules. Sufficient number of rules relates to the total rules which are not too high, making computation too costly and also not too low leading to decrease in quality of rules.

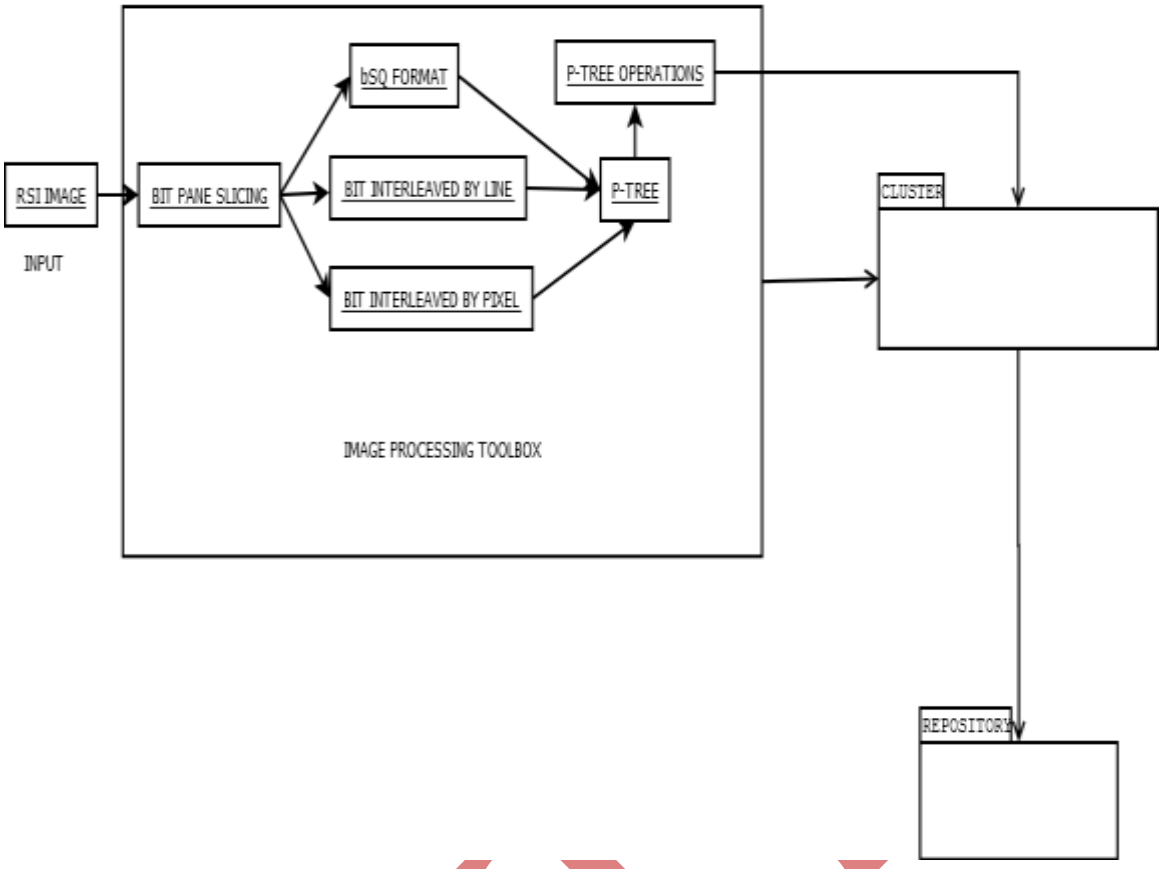


**Fig.7 Image Processing Toolbox**

#### 4.1.1.3 Source Code

This forms a part of input to the cluster. It processes all the data stored in the cluster and then assigns processing tasks to each and every node accordingly. Source code itself here generally is in parallelized form in order to decrease computing time and increase the computing power utilization provided by the cluster.

### 4.1.2 Cluster

Fig.6 represents various components of the cluster. The cluster consists of 5 off-shelf computers. One of these nodes acts as a master and all the other become slaves. The configuration and set up of the clusters was covered in detail in literature survey. The nodes are interconnected using a high speed LAN. All the data communication happens through the master and the master is responsible for distributing the parallel code among various clusters.

### 4.1.3 Repository

Repository represents a central storage area where all the data regarding the application is stored. The data stores in the repository can be partitioned into the following based on the type and source of the data.

1. Images
2. P-trees
3. Association rules.

Also various logs of data transfer between master and the slaves are stored in the repository. Now the question why repository and why not database? File I/O is much faster in case of Linux systems and thus repository is used instead of database.

Fig.8 represents the system flow chart. The input image is given to image processing toolbox which generates the p-tree. This p-tree is used for generating association rules which are stored in the repository. Then from the test image the association rules are generated using PARM. These rules are stored in a temporary file in the repository. These rules in the temporary file are compared with the rules already in the repository, if and only if the image is from the same application domain and conclusions are made based on the support and confidence needed. In case the association rules generated contain not even a single similar rule generated, then it implies that the images are from different domains and thus the input test image was from different application domain and hence invalid.

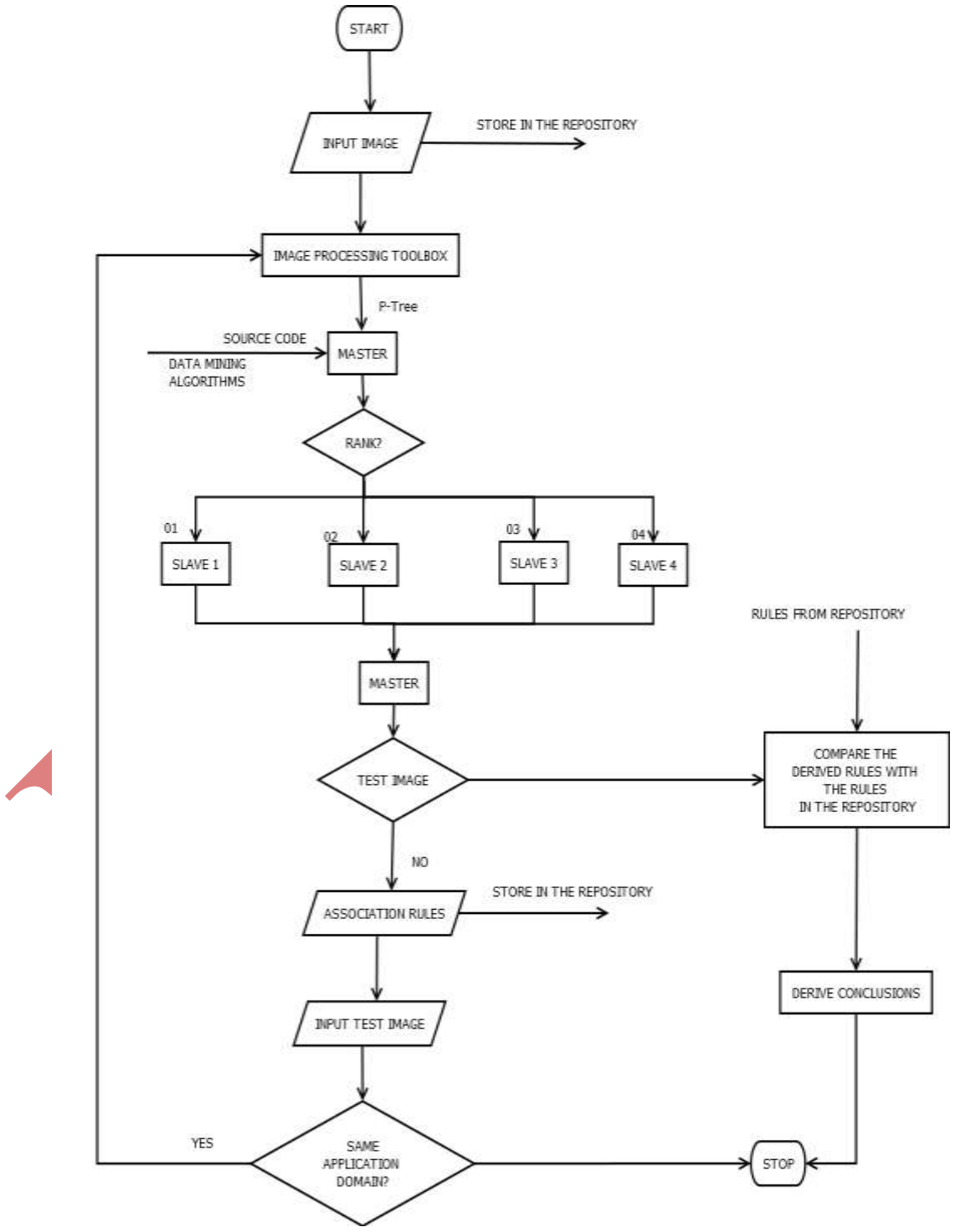


**Fig.8 System Flow Chart**

## V. APPLICATIONS

PARM algorithm finds its applications into various domains mostly containing spatial data. Here we mention some of the application areas like precision agriculture, resource location, disaster management and gene expression profiling of DNA microarray data [4]. During the parallelization of PARM we consider the RSI images pertaining to the weather in order to deduce the weather conditions in a particular region.

## VI. CONCLUSION AND FUTURE WORK

Parallelizing PARM is needed to process spatial data faster and derive conclusions from the data faster. PARM uses real time data for spatial data mining. Any abnormality in the data must be quickly detected and necessary steps must be performed. Here we propose a mechanism through which parallelization can be performed. The future work includes the implementation of this design on Linux clusters.

## REFERENCES

[1]. Margaret H. Dunham, "Association rules," in Data Mining-Introductory and advanced topics of her Published Book, 13th impression. Noida, Country India: (Dorling Kindersley) Pearson, 2013.

[2]. PARM—An Efficient Algorithm to Mine Association Rules From Spatial Data Qin Ding, Qiang Ding, and William Perrizo, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 38, NO. 6, DECEMBER 2008.

[3]. Bernd Bruegge, "System Design: Decomposing the system", in Object Oriented Software Engineering of his published book, 2nd edition, Pearson Education.

[4]. Peano count tree and rule association mining for gene expression profiling of DNA microarray data. Valdivia Granda. Proceedings of The International Conference on Bioinformatics 2002.