A Survey On Parallelization Of Data Mining Techniques

Shrikant Gond, Akshay Patil And V. B. Nikam
Department of Computer Engineering and Information Technology, VJTI, Mumbai

Abstract

This paper contains the overview of various parallelization techniques to improve the performance of existing data mining algorithms and make the capable of handling large amount of data. There are variety of techniques to achieve the parallelization in data mining field, in this paper a brief introduction to few of the popular techniques is presented. The second part of this paper contains information regarding various data algorithms that are proposed by various authors based on these techniques. In Introduction various results corresponding to a survey are provided.

Keywords: *OpenCL, CUDA, Cluster, Cloud, Grid, Hadoop, Association Rule Mining, GPU.*

I. INTRODUCTION

.

II. DATA MININER SURVEY

This section provides various statistical finding from a survey conducted by a Rexer Analytics [1] on their 5th Annual data miner survey report. This report contains various aspects about data miners like work area, algorithms that they are using, computing environment, data mining tools and their satisfaction, and the key findings out of this survey.

In this survey 10,000+ invitations emailed, plus promoted by newsgroups, vendors, and bloggers, 52 questions were asked to each survey participant, in all there are 1,319 data miners from over 60 countries who had participated in this survey.

The key findings in this survey are as follows:

• **FIELDS & GOALS:** Data miners work in a diverse set of fields. CRM / Marketing has been the #1 field in each of the past five years. Fittingly, “improving the understanding of customers,” “retaining customers,” and other CRM goals continue to be the goals identified by the most data miners.

• **ALGORITHMS:** Decision trees, regression, and cluster analysis continue to form a triad of core algorithms for most data miners. However, a wide variety of algorithms are being used. A third of data miners currently use text mining and another third plan to in the future. Text mining is most often used to analyze customer surveys and blogs/social media.

• **TOOLS**: R continued its rise this year and is now being used by close to half of all data miners (47%). R users report preferring it for being free, open source, and having a wide variety of algorithms. Many people also cited R's flexibility and the strength of the user community. STATISTICA is selected as the primary data mining tool by the most data miners (17%). Data miners report using an average of 4 software tools overall. STATISTICA, KNIME, Rapid Miner, and Salford Systems received the strongest satisfaction ratings in 2011.

• **TECHNOLOGY**: Data Mining most often occurs on a desktop or laptop computer, and frequently the data is stored locally. Model scoring typically happens using the same software used to develop models.

• **VISUALIZATION**: Data miners frequently use data visualization techniques. More than four in five use them to explain results to others. MS Office is the most often used tool for data visualization. Extensive use of data visualization is less prevalent in the Asia-Pacific region than other parts of the world.

• **ANALYTIC CAPABILITY AND SUCCESS**: Only 12% of corporate respondents rate their company as having very high analytic sophistication. However, companies with better analytic capabilities are outperforming their peers. Respondents report analyzing analytic success via Return on Investment (ROI), and analyzing the predictive validity or accuracy of their models. Challenges to measuring analytic success include client or user cooperation and data availability/quality.

The various data mining fields according to the survey are as follows:

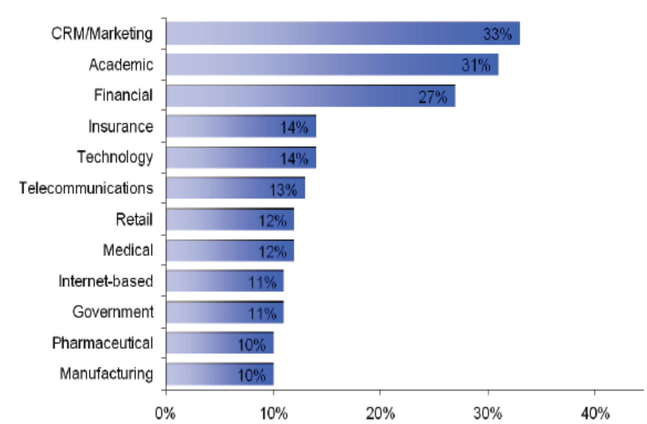

Figure 1. Data miners percentage according to sectors

Similarly various algorithms that are commonly used by these miners are as follows:

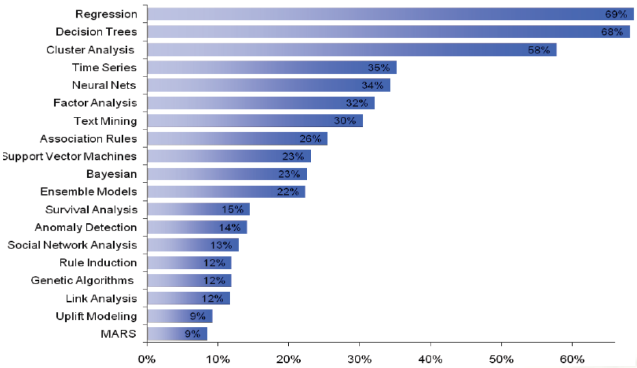

Figure 2. Various data mining algorithms used by data miners

This survey mentioned about computing environment of various data miners, it is mostly a desktop or laptop computer.

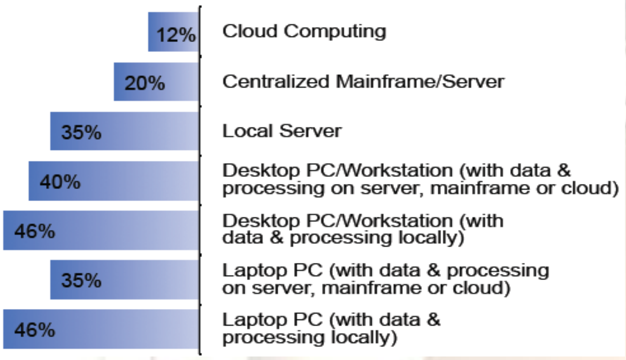

Figure 3. Various computing envirnments

This survey also provides a very important aspect of our concern i.e. various data mining tools used by these miners and overall frequency of use of each tool. The details are given in following figure.

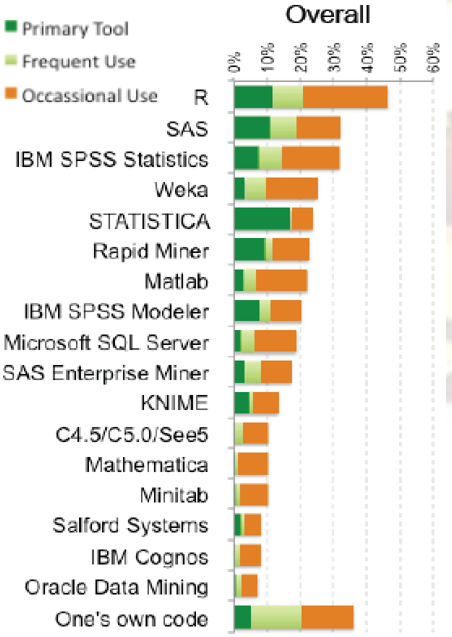

Figure 4. Various tools used by data miners

From this survey we get a clear idea of various data mining techniques and tools in current market, now we focus on various techniques for parallelization of data mining algorithms.

III. Parallelization Techniques

There are various techniques by which we can achieve parallelization in data mining, like Grid, Clusters, Shared memory parallelization (SMP), Message Passing Interface (MPI), OpenMP, Cloud, GPU.

We will now elaborate each of them in brief as follows:

A. GRID

The idea behind grid computing is that one can plug one's computer into the wall and have access to computational and data resources without knowing where they are or who owns these resources. The term grid is stemming from the field of electricity network which provides a power grid one can use by plugging a power cable into the wall socket, this way getting the electricity that is needed without knowing where it comes from. Grid computing, simple stated, is taking distributed computing to the next level. So first a short definition of distributed computing followed by the definition of grid computing. Distributed computing means dividing tasks among multiple computer systems instead of doing the tasks on one centralized computer system. Distributed computing is a subset of grid computing, grid computing encompasses much more. Grid computing provides coordinated sharing of geographically distributed hardware, software and information resources, this sharing is highly controlled defining clearly what is shared, who is sharing and the conditions of the sharing, it provides a service oriented infrastructure and uses standardized protocols to accomplish this sharing.[2]

1) Architecture:

The architecture of grids is often described in terms of layers. The lower layers being the computers and the networks and the higher layers being more focused on the user and the applications.

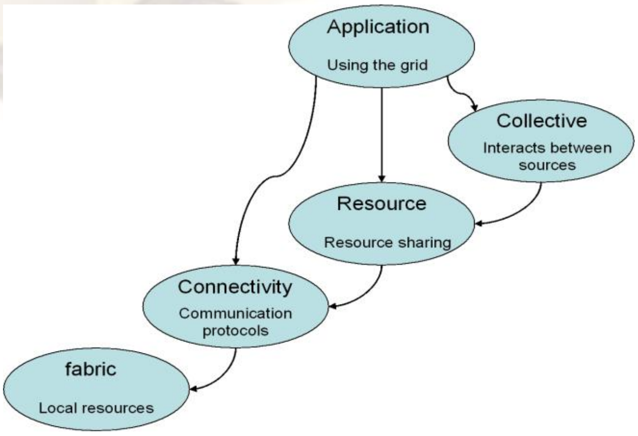

Figure 5. Architecture of a Grid

The fabric layer provides the local resource specific operations on resources that are shared on the grid, e.g. computational, network, catalogues and storage resources.

The connectivity layer provides the core communication and authentication protocols. It enables save and reliable data exchange between fabric layer resources.

The resource layer enables resource sharing, it builds on the connectivity layer to control and access resources. It uses information protocols to obtain information about resources, and it uses management protocols to negotiate access to shared resources.

The collective layer coordinates interactions between multiple sources, it ties multiple sources together. E.g. it can combine data sources from multiple sites into one virtual data source, it can perform computations on multiple sites and return the results back to the user.

The application layer holds the application of the user, this layer uses the connectivity layer, the resource layer and the collective layer to perform grid operations in virtual organizations.

The connectivity, resource and collective layer are often called the middleware layer and are implemented by middleware software. Middleware is software that connects software components or applications, it is often used for complex, distributed applications and can be seen as the intelligence that brings all the elements together. [2]

2) Techniques:

*a) **OGSA** The Open Grid Services Architecture defines grid services as an extension of web services for a standard model to use grid resources. Every resource is represented as a grid service: a Web Service that conforms to standard conventions and supports standard interfaces. A Web service is a software system designed to support interoperable machine-to-machine interaction over a network, focusing on simple internet based standards such as the Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL). OGSA provides a well defined set of basic interfaces for the development of interoperable grid systems. OGSA is an implementation of the service oriented architecture (SOA) model within the grid context. SOA is a programming model to build flexible, modular and interoperable applications. The emphasis of grid computing shifted from intensive computing tasks to data intensive tasks. That is one of the reasons that OGSA-DAI was created, it is the database access and integration service. It allows data sources to be accessed via web services and it makes it possible to integrate data from various sources.*

*b) **WSRF**, Web Service Resource Framework The WSRF defines a standard specification to merge grid and web technology, this way building a bridge between the grid and the web. OGSI was the predecessor of WSRF, it was accepted by the grid community but not by theWeb Service community because it does not work well with existing Web Services, and therefore a new standard was developed. WSRF defines specifications to access and managing stateful resources using web services; a stateful resource is a resource that can keep track of its state for multiple clients. The Globus Toolkit 4 contains Java and C implementations of WSRF, while GT3 contained OGSI implementations.*

3) Applications

a) GridMiner

The GridMiner [3] application is made for the development and runtime execution of data mining processes and data mining preprocessing on grids. It is a Service oriented grid application that integrates all aspects of the data mining process: data cleaning, data integration, data transformation, data mining, pattern evaluation, knowledge presentation and visualization. Goal of the GridMiner application: an easy to use tool for an expert data miner to ease the process of data mining on a grid system.

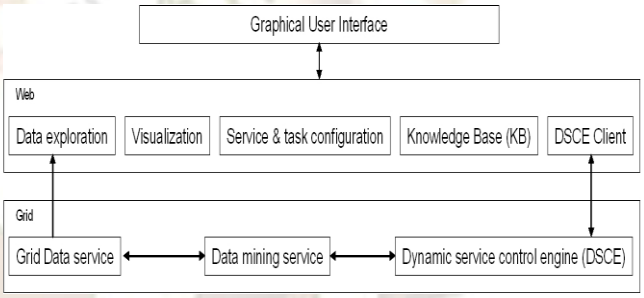

Figure 6. Architecture of the GridMiner application

This figure shows the architecture of the application divided in three layers: the user interface, the web and the grid layer. The most important and interesting layer is the grid layer.

Grid Layer

The grid layer takes care of the execution of the data mining algorithms, the data preprocessing and the OLAP services. The execution is directed by the workflow engine and is supported by services such as the mediation service, security service and the file and database access service.

b) Weka4WS

Weka4WS [2][3] is an application that extendsWeka to perform data mining tasks on WSRF enabled grids. The first prototype of Weka4WS has been developed using the Java WSRF library provided by GT4.

The goal of Weka4WS is to support remote execution of data mining algorithms in such a way that distributed data mining tasks can be concurrently executed on decentralized nodes on the grid, exploiting data distribution and improving performance. Each tasks is managed by a single thread and therefor a user can start multiple tasks in parallel, taking full advantage of the grid environment.

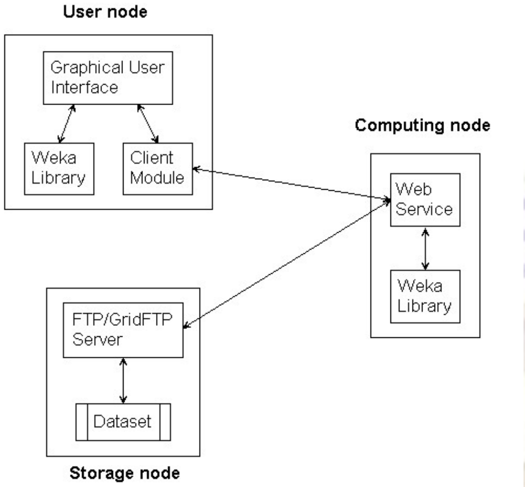

Figure 7. Weka4WS architecture

There are three types of nodes in Weke4WS: local nodes (or user nodes) with Weka4WS client software, computing nodes that provide theWeka4WSWeb Services and storage nodes which provide access to data to be mined. Data can be located on the local, the computing and the storage nodes as well as on third party nodes. If the data that is to be mined is not on a computational node it can be uploaded using the GT4 data management services (GridFTP). The other steps in the process, data-preprocessing and visualization, are still executed locally.

B. Cluster

A computer cluster consists of a set of loosely connected computers that work together so that in many respects they can be viewed as a single system. The components of a cluster are usually connected to each other through fast local area networks, each node (computer used as a server) running its own instance of an operating system.[4]
PC running its own general-purpose operating system. A general-purpose internode network, such as the Ethernet, connects these nodes together. Data communication among the PCs is controlled by application layer software rather than by lower-level system software or hardware. Thus, the latency of data communications is usually longer than that of parallel computers and supercomputers that contain specialized hardware to implement communication networks. For PC clusters, it is more difficult to exploit low-level or fine-grain parallelism potentially existing in programs. It is more appropriate to adopt coarse- or medium-grain programming models for PC clusters.

1) Architecture:

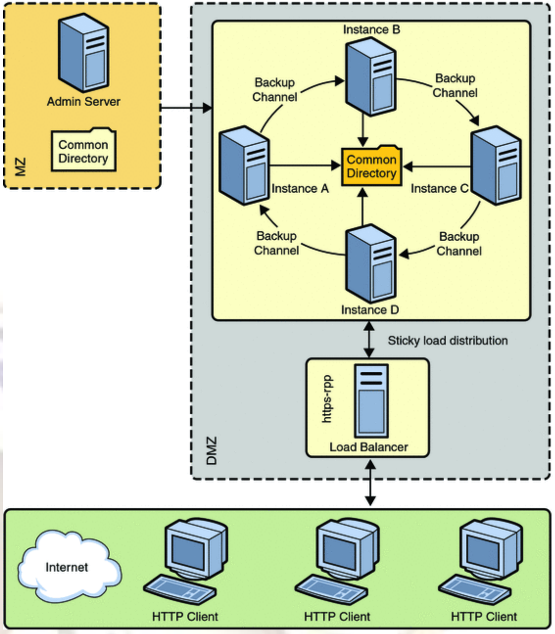

Figure 8. Architecture of a Simple Cluster[5]

2) Techniques:

a) Super Programming Model

To effectively balance the workload and also support application portability, we introduce here the super-programming model (SPM). In SPM, the parallel system is modeled as a single virtual machine. SPM decomposes applications in a workload-oriented manner. It suggests that an effective instruction set architecture (ISA) be developed for each application domain. The frequently used operations in that domain should belong to this ISA. Their operand sizes are assumed limited by predefined thresholds. Application programs are modeled as superprograms (SPs) coded with SIs.
In SPM, the PC cluster is modeled as a single virtual machine (VM) with a single super-processor; the latter includes one instruction fetch/dispatch unit (IDU) and multiple instruction execution units (IEUs).

SUPER-DATA BLOCKS (SDBS)

The superprocessor can handle a set of “build-in” data types and a set of basic abstract operators that can be applied to the former. These data types are stored in SDBs. The operations are performed by super-instructions (SIs). SDBs are high-level abstract data types. On each PC, they are expressed in their local format. Implementers are free to adopt any data structure supported by the languages used to implement VM. As long as the SDB formats have been set up along with the SDB exchange protocols, nodes with different architectures can freely exchange SDBs. This feature makes it very easy to work with heterogeneous clusters.

SUPER INSTRUCTIONS (SI$_S$)

SIs are high-level abstract operations applied to SDBs. They collectively constitute the basic operations of VM. Similarly to instructions for microprocessors: 1. SIs are subject to data dependencies. There is no communication logic embedded in their body. Dependencies are handled only at the beginning and end of an SI's execution. Once all operands are locally available, an SI can be executed without any interruption. 2. SIs are atomic. Each SI can only be assigned to and be executed on a single IEU. 3. The workload of each SI has a quite accurate upper bound for a given computer. The operands, of course, have limited size.

SUPER-FUNCTIONS (SFs)

To facilitate ease of application development, programs are usually developed using high-level structures. The latter combine many simple low-level operations to form high level abstract operations, such as reusable subprograms or functions. Entire programs consist of lists of such operations; they describe how a computer system should perform computations to solve the respective problem. In SPM, these functions are called SFs. SFs are "binary" executable procedures for VM that can be executed on the superprocessor. Application programs are modeled as SPs which are implemented as collections of SFs. When an SP runs on a PC cluster, its SFs are called as they are encountered sequentially. The IDU fetches SIs in SFs and dispatches them to the IEUs.

C. Cloud

1) Architecture:

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

End users access cloud-based applications through a web browser or a light-weight desktop or mobile app while the business software and user's data are stored on servers at a remote location. Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of infrastructure.[1] Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand.

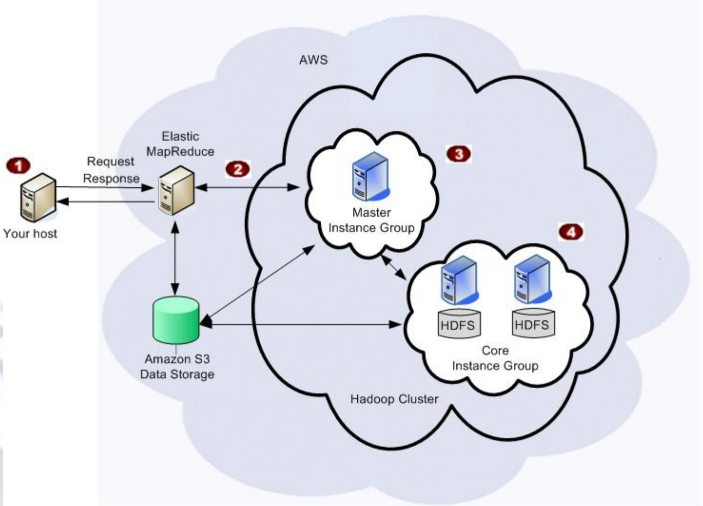

Figure 9. EC2 cloud for Hadoop

Hadoop job flow on the EC2 cloud is shown in Figure 4. To start the job, a request is sent to the EMR model with parameters, such as path to S3. The Hadoop cluster with master and slave instances is created. The Hadoop cluster works on the job and finishes the job. The temporary files created during the execution of the job can be stored either on HDFS or on S3. Storing files on S3 would not be wise for our work because it adds communication overhead. The final output is stored in S3. Only the error or fatal messages are written on the screen during the entire execution of a job. Once the job is completed, a message is sent to the user indicating the completion of the job.[6]

2) Techniques:

a) Association rule mining on Hadoop

In this technique Apriori Algorithm is applied to Hadoop cloud. The frequent item set were generated following the Apriori algorithm. As the input data and number of distinct items in the data set is large, lots of space and memory is required, so Hadoop was used, as Hadoop provide parallel, scalable, robust framework in the distributed environment.

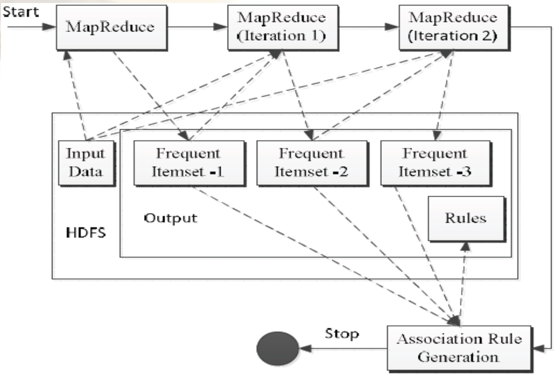

Figure 10. Data flow diagram showing two iterations.

The single node Hadoop was run on the virtual machine to mimic the Hadoop environment. The time function used for measuring was system time, it returned time in millisecs. Fig 10 shows the flow data for two iterations in the project. The Fig 11 shows the Map data is written in temporary files in HDFS which is taken in by combine and processed by it. Later the output of the Combine to be required by reduce is written in temporary files and which is further processed by the Reduce. Reduce also saves the data in a temporary file while it is processing the data. Once all the data is processed by the Reduce for that stage the resultant temporary file is converted into a permanent file and is stored in the specified output path. The details of passing the correct path of temporary files to Combine and Reduce are taken care by the Hadoop framework.

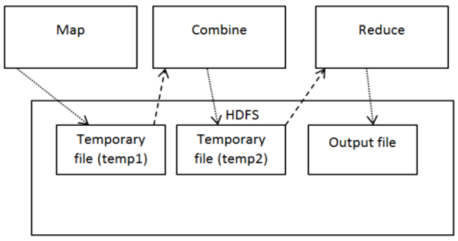

Figure 11. data transfer among the methods Map, Combine and Reduce in distributed system.

D. GPU

1) *Architecture:*

The architecture of the GPU consist of number of processing elements which forms a compute device these processing elements share local memory which limited in terms of size and bandwidth. These compute devices are connected with the a global memory which is accessible to all the processing elements inside the GPU and which is also accessible to CPU, the bandwidth and the size of this memory high as compared to local memory.

When we want to perform some operation on GPU we need to transfer the data from CPU memory (RAM) to GPU Global memory (Device Memory) the flow of data from various memory paths is shown in figure 11.

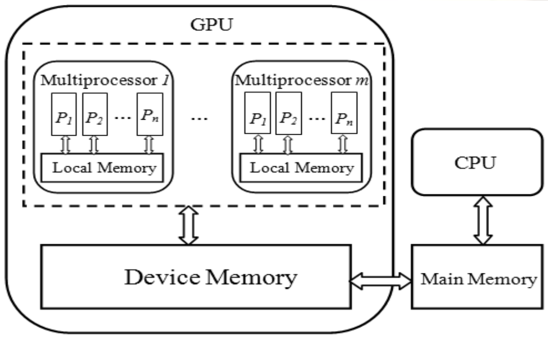

Figure 12. Architeture of a GPU[15].

There are various uses of GPU ranging from Graphics processing to General purpose computing, depending on that There are mainly two kinds of GPU programming languages: graphics APIs such as DirectX and OpenGL, and GPGPU languages such as CUDA and OpenCL.

2) *Techniques:*

a) *GPUMiner*

As an integrated data mining system, GPUMiner has the following features.

High performance: The data mining algorithms in GPUMiner are designed and implemented as parallel ones exploiting the parallelism of the entire machine, including the co-processing parallelism between the CPU and the GPU, and the on-chip parallelism within each processor. In particular, these parallel algorithms are scalable to hundreds of processors on the GPU.

I/O handling infrastructure: GPUMiner provides a flexible and efficient I/O handling infrastructure for analyzing large amounts of data.

Online visualization: Data mining is often a long-running and interactive process. Visualization helps people mine large dataset more efficiently. GPUMiner provides online visualization for the user to observe and interact with the mining process.

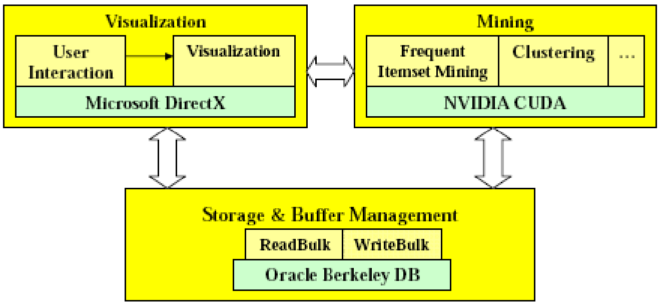

Figure 13. EC2 cloud for Hadoop[15]

GPUMiner utilizes Berkeley DB (Bdb) as the backend for storing the data persistently. Compared with the raw I/O APIs accessing data in plain text files or structured files, Bdb transparently provides the efficient buffer management between the disk and the main memory, together with convenient file I/O operations including in-place data update. Since current version of GPUMiner supports bulk reads and writes only, we store a bulk of data as a record in Berkeley DB with a unique key. Thus, a data chunk can be fetched or stored by the key. Based on the buffer management from Bdb, GPUMiner provides a lightweight I/O library consisting of two APIs, namely ReadBulk andWriteBulk.

ReadBulk reads a chunk of data from the disk and transfers them to the GPU memory, whereas WriteBulk outputs a chunk of data from the GPU memory to the disk. With these two APIs, developers can handle large data sets without considering explicit data allocation and data transfer among the GPU memory, the main memory and the disk.

IV. Applications of DM in Parallel Domain

Various applications of data mining in parallel domain includes very large data set mining such as in the field of social networking like Facebook, image and video data mining in medical science for dignosis. Astronomical analysis etc.

These fields contain data of huge volume that is required to be processed in finite amount of time. Using parallization techniqes for data mining we can perform mining operations on this data in less amount of time which will be very helpful in these areas.

V. Conclusion

This paper surveys various techniques for paralleization data minng tasks , which includes GRID, Cluster, Cloud, GPU. There are advantages and disadvanteges of each of the techniques, so depending on the application, data size and budget one can select the best suited technigque for parallelization.

There are few applications of parallelization technique in this paper provided for a better understanding of that technique, each technique varies in terms of performace, cost of implemetation, and development time. The selection of a technique must be done by concider all these factors in mind.

References

[1] "5th Annual Data Miner Survey – 2011 Survey Summary Report", Rexer Analytics, Karl Rexer, PhD krexer@RexerAnalytics.com www.RexerAnalytics.com

[2] "Data mining on grids" Maarten Altorf, maltorf@yahoo.com, Universiteit Leiden - August 2007.

[3] D. Talia, P. Trunfio, O. Verta. Weka4WS: a WSRF-enabled Weka Toolkit for Distributed Data Mining on Grids. Proc. of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005), Porto, Portugal, October 2005, LNAI vol. 3721, pp. 309-320, Springer- Verlag, 2005.

[4] http://en.wikipedia.org/wiki/Computer_cluster

[5] http://docs.oracle.com/cd/E19146-01/821-1828/gczop/index.html

[6] Juan Li, Pallavi Roy, Samee U. Khan, Lizhe Wang, Yan Bai "Data Mining Using Clouds: An Experimental Implementation of Apriori over MapReduce".

[7] http://en.wikipedia.org/wiki/Cloud_computing

[8] "Mining Association Rules In Cloud" By Pallavi Roy

[9] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[10] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[11] Electronic Publication: Digital Object Identifiers (DOIs):

[12] D. Kornack and P. Rakic, "Cell Proliferation without Neurogenesis in Adult Primate Neocortex," Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[13] H. Goto, Y. Hasegawa, and M. Tanaka, "Efficient Scheduling Focusing on the Duality of MPL Representatives," Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.

[14] "Data Mining Using Clouds: An Experimental Implementation of Apriori over MapReduce" Juan Li, Pallavi Roy, Samee U. Khan, Lizhe Wang, Yan Bai,North Dakota State University, Fargo, USA

[15] Wenbin Fang, Ka Keung Lau, Mian Lu, Xiangye Xiao, Chi Kit Lam, Philip Yang Yang, "Parallel Data Mining on Graphics Processors", Technical Report HKUSTCS0807,Oct 2008.