Association Rules Mining using BOINC–based Enterprise Desktop Grid*

Evgeny Ivashko and Alexander Golovin

Institute of Applied Mathematical Research,
Karelian Research Centre of Russian Academy of Sciences, Petrozavodsk, Russia,
{ivashko,golovin}@krc.karelia.ru

Abstract. The paper is devoted to association rules mining using BOINC–project based on Enterprise Desktop Grid. A high-level algorithm description is given. A BOINC–based application is developed and validated. Several experiments with the aim of performance evaluation are performed. Ways to further develop the approach are described.

Keywords: Enterprise Desktop Grid, Desktop Grid, BOINC, distributed computing, data mining, data analysis, association rules

1 Introduction

Association rules mining is one of the Data Mining methods aimed to data analysis. This method is a well-studied area, because of its importance in many problems of data analysis. Association rules express the association between records in a transactional database. The problem of discovering frequent itemsets in a transactional data set (the so called FIM problem) is the first step of association rules mining. A number of algorithms have been suggested to discover frequent itemsets: Apriori [7], FP-Growth [8], Eclat [9], and others.

The Berkeley Open Infrastructure for Network Computing (BOINC) is an open source software framework for distributed and grid computing [1]. BOINC is based on the client/server model. A central server has project's database storing information about registered users and associated hosts, applications, tasks and results of calculations as well as other information. Also there are special services on the central server:

- **Work generator** which generates new workunits and corresponding input files.
- **Feeder** is used to enhance the performance scheduler and to reduce a number of queries to the database of project.
- **Scheduler** which assigns jobs to a client taking into account its characteristics.
- **Transitioner** handles state transitions of workunits and results.

* The work is supported by grants of Russian Fund for Basic Research 15-29-07974, 13-07-00008 and 15-07-02354

- **Validator** which decides (following the special procedure) whether results are correct.
- **Assimilator** which periodically checks the completed jobs and processes results according to application-specific rules.
- **File deleter** deletes input and output files as jobs are completed.
- **Database purger** writes result and workunit records to XML-format archive files, then deletes them from the database.

Ordinary BOINC–project should be computational intensive instead of data intensive because of poor connection between BOINC server and its clients. Our research is based on Enterprise Desktop Grid to work with data-intensive applications with speed of local-area networks. The performance of Enterprise Desktop Grid also can be improved by advanced scheduling [17, 18].

The appropriate adaptation of the software is required to use an Enterprise Desktop Grid for data processing. The software uses the BOINC API to implement interaction between a BOINC server, client and running application.

This paper presents the native BOINC-based application for mining association rules in big data sets based on Enterprise Desktop Grid. The performance evaluation experiments are performed; the directions of the future work are discussed.

2 Software for association rules mining

Association rule is an implication $X \to Y$, where X and Y are (not large) data sets. Such a rule has two important characteristics: support (s) and confidence (c) of the rule. The association $X \to Y$ means that if a transaction contains data set X then it also contains data set Y; there are $s\%$ of transactions in the database containing both X and Y; there are $c\%$ of all transactions that contain X also contain Y.

A modification of Partition algorithm is used to solve the problem of finding frequent itemsets in large volumes of data. Partition itself is a parallel modification of a well-known Apriori algorithm, it has good scalability and performance. The description of the algorithm can be found in paper [4].

The algorithm requires two scans of the initial database and six steps [16]. Two steps are executed in parallel on the computing nodes of the BOINC–based Enterprise Desktop Grid:

1. **Preprocessing**: the work generator receives an input source file with the transactional database and the following parameters: the minimum support and confidence, the number which determines into how many parts is the source file divided as well as some additional BOINC–related workunit attributes.
2. **Stage I (parallel)**: the BOINC scheduler distributes jobs to clients (computing nodes of the BOINC-grid). BOINC-clients download input files (which are parts of the original transactional database) from the server. Then the clients run an application that extracts local frequent itemsets from their

parts. After that clients upload the output files to the server and report on completing the jobs.
3. **Merge stage**: the server side validation service validates the results.
4. **Intermediate stage**: completed jobs are handled by an assimilator which generates the set of all global candidates based on the received local frequent itemsets. Also this service generates new jobs.
5. **Stage II (parallel)**: the BOINC scheduler distributes the new jobs to the clients. Each BOINC-client calculates support for each global candidate itemset in its part of the transactional database.
6. **Final stage**: After receiving the canonical result the assimilator summarizes supports for each candidate and removes the ones whose support is less than the specified minimum. At the same step, the assimilator constructs the association rules.

3 Experiments

Several experiments with the aim of validation and performance evaluation of the Partition algorithm implementation were performed. We used a BOINC-based Enterprise Desktop Grid with up to 32 computing nodes connected to BOINC-server by local area network.

First of all, we validated the application by test source datasets from the Frequent Itemset Mining Dataset Repository (FIMDR) [13]. The results of the performance evaluation experiments are presented in the Fig. 1 (three datasets are used; at the figure they noted as I, II and III).

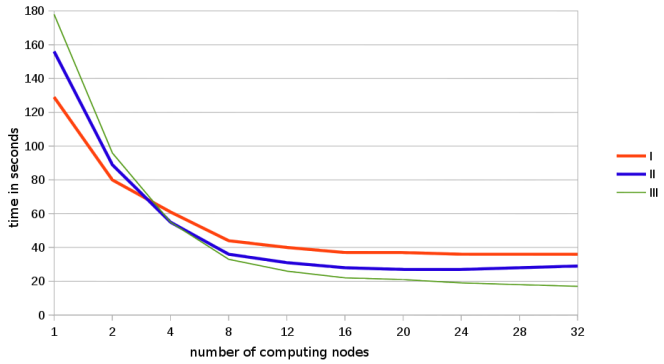

Fig. 1. The results of the experiments on the test datasets.

The figure shows that use of BOINC accelerates rules extraction up to ten times. The overall time of rules extraction depends on the minimal support and length of transactions.

However, because of data-intensive nature of the project BOINC's overhead is very large comparing with ordinary compute intensive applications (see Fig. 2 to compare overheads association rules mining and SETI@HOME projects). The main performance limitation in the performed experiments is still the net-

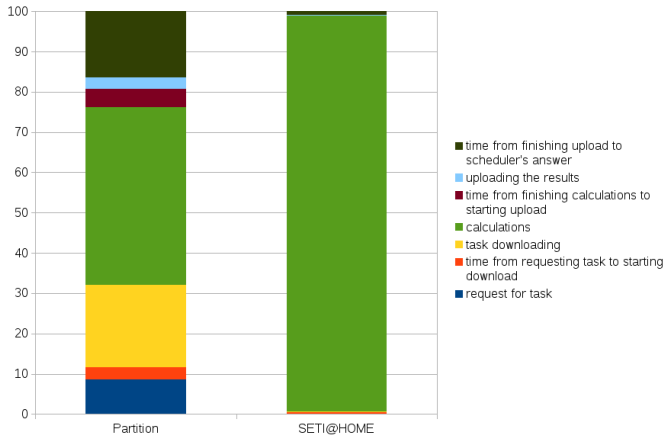

Fig. 2. BOINC server overhead.

work bandwidth. BOINC-server should distribute a database between computing nodes that becomes a very time-consuming operation. Nevertheless use of Enterprise Desktop Grid (and a local-area network) allows to improve the speed of data analysis.

4 Conclusion and discussion

BOINC is a popular tool to perform large-scale computational experiments. BOINC-based Enterprise Desktop Grid allows to small and medium companies or small scientific groups to solve their private problems using their own computing resources. One of such private problems is analysis of big datasets of an organization.

This study shows the way to extract association rules from big data sets using BOINC-based Enterprise Desktop Grid. We adapted the Partition algorithm for BOINC and performed the experiments on performance evaluation of association rules extraction. Our results show that Enterprise Desktop Grid allows to reduce expended time for data analysis.

There are also several ways to overcome a limitation of the network bandwidth. For example, the Partition algorithm can be adapted to process data at the point of gathering these data, i.e. POS-terminal of a supermarket can analyse a market basket together with other POS-terminals without gathering their

data. This removes the need to transfer a source database through a network. There are also some more workarounds to the problem of limitation the network bandwidth.

Also BOINC can be much more useful in case of searching the optimal values of minimal support. In this case one have to perform data analysis multiple times (using different values of minimal support) with the same data. So, the overhead will be reduced comparing with computations.

Further development of the study will be devoted to adapting other methods of data analysis to BOINC-environment. Also it is important to combine the developed tool with special visualization software.

References

1. D. P. Anderson, "BOINC: A system for public-resource computing and storage", in *Fifth IEEE/ACM International Workshop on Grid Computing*, pp. 4–10, 2004.
2. E. Cesario, N. De Caria, C. Mastroianni, D. Talia, "Distributed Data Mining using a Public Resource Computing Framework", in *Grids, P2P and Services Computing*, Springer US, pp. 33–44, 2010.
3. N. Schlitter, J. Laessig, S. Fischer, I. Mierswa, "Distributed Data Analytics using RapidMiner and BOINC", in *Proceedings of the 4th RapidMiner Community Meeting and Conference (RCOMM 2013)*, pp. 81–95, 2013.
4. A. Savasere, E. Omiecinski, and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases", in *Proc. 21st Int. Conf. on Very Large Data Bases*, Morgan Kaufmann, San Francisco, pp. 432–444, 1995.
5. H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang, "Pfp: parallel fp-growth for query recommendation", in *RecSys '08 Proceedings of the 2008 ACM conference on Recommender systems*, pp. 107–114, 2008.
6. D. Cheung, J. Han, V. T. Ng, A. W. Fu, Y. Fu, A. W. Yongjian, "A Fast Distributed Algorithm for Mining Association Rules", *Proc. of Int. Conf. on PDIS'96*, pp. 31–42, 1996.
7. R. Agrawal, R. Srikant, "Fast Discovery of Association Rules", in *Proc. of the 20th Int. Conf. on VLDB*, Santiago, Chile, pp. 307–328, 1994.
8. J. Han, H. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation", in *Proc. Conf. on the Management of Data*, Dallas, TX, pp. 1–12, 2000.
9. M. J. Zaki, "Scalable algorithms for association mining", *IEEE Trans. Knowledge and Data Engineering*, vol. 12, i. 3, pp. 372–390, 2000.
10. The 5th Annual Rexer Analytics Data Miner Survey [Online]. http://www.rexeranalytics.com/Data-Miner-Survey-Results-2011.html
11. Encyclopedia Britannica [Online]. http://global.britannica.com/EBchecked/topic/1056150/data-mining
12. D. Barbalace, C. Lucchese, C. Mastroianni, S. Orlando, D. Talia, "Mining@HOME: public resource computing for distributed Data Mining", *Concurrency & Computation: Practice & Experience*, Wiley, vol. 22, i. 5, pp. 658–682, 2010.
13. Frequent Itemset Mining Dataset Repository [Online]. http://fimi.ua.ac.be
14. M. K. Saad, R. M. Abed, "Distributed Data Mining On Grid Environment", *American Academic & Scholarly Research J. Spec. Iss.*, vol. 4, no. 5, pp. 240–243, 2012.
15. D. Talia, P. Trunfio, V. Verta, "Weka4WS: a WSRF–enabled Weka Toolkit for Distributed Data Mining on Grids", in *Knowledge Discovery in Databases: PKDD 2005*, pp. 309–320, 2005.

16. E. Ivashko, A. Golovin. Partition Algorithm for Association Rules Mining in BOINC–based Enterprise Desktop Grid. Parallel Computing Technologies. LNCS Vol. 9251. 2015.
17. Mazalov V. V., Nikitina N. N., Ivashko E. E. Hierarchical Two–Level Game Model for Tasks Scheduling in a Desktop Grid. Applied Problems in Theory of Probabilities and Mathematical Statistics Related to Modeling of Information Systems. 2014.
18. Ilya Chernov, Natalia Nikitina. Virtual Screening in a Desktop Grid: Replication and the Optimal Quorum. Parallel Computing Technologies. LNCS Vol. 9251. 2015. Pp. 258-267