Association Rule Mining on Grid Monitoring Data to Detect Error Sources

Gerhild Maier[1,2], **Michael Schiffers**[3], **Dieter Kranzlmueller**[3], **Benjamin Gaidioz**[1]
[1] CERN, Geneva, Switzerland
[2] Johannes Kepler University, Linz, Austria
[3] Ludwig Maximilian University, Munich, Germany

E-mail: Gerhild.Maier@cern.ch

Abstract. Error handling is a crucial task in an infrastructure as complex as a grid. There are several monitoring tools put in place, which report failing grid jobs including exit codes. However, the exit codes do not always denote the actual fault, which caused the job failure. Human time and knowledge is required to manually trace back errors to the real fault underlying an error. We perform association rule mining on grid job monitoring data to automatically retrieve knowledge about the grid components' behavior by taking dependencies between grid job characteristics into account. Therewith, problematic grid components are located automatically and this information - expressed by association rules - is visualized in a web interface. This work achieves a decrease in time for fault recovery and yields an improvement of a grid's reliability.

1. Introduction

Grid computing is associated with a complex, large scale, heterogeneous and distributed environment. The combination of different grid infrastructures, middleware implementations, and job submission tools into one reliable production system is a challenging task. Given the impracticality of providing an absolutely fail-safe system, strong error reporting and handling is a crucial part of operating a grid infrastructure.

In the scope of grid computing, not only a lot of input and output data is produced, transfered, and used, but also meta information is needed to make statements about reliability, functionality, and usage of a grid. There are several monitoring tools, which generate data to help understand and control the behavior of a computing grid.

The Worldwide LHC Computing Grid (WLCG)[7] provides an infrastructure for data storage and processing designed to handle the high energy physics experiments undertaken at the Large Hadron Collider (LHC) located at CERN.

Considering the large amount of hardware resources distributed in more than 140 computing centers in 33 countries and the large number of jobs run every day (in average about 300.000), a large amount of monitoring data is produced. This data contains, among other things, information about faulty grid components which are responsible for failing grid jobs and weaken a grid's reliability.

In this paper we describe a data mining technique called association rule mining (ARM) which is performed on grid job monitoring data, to reveal hidden information about grid components in the mass of monitoring data.

The following section states the problem and introduces our approach. Section 3 presents an introduction to association rule mining in general and section 4 demonstrates the application of ARM to grid job monitoring data of one of the experiments undertaken at the LHC at CERN. We conclude our work and outline future research in section 5.

2. Problem Formulation and Approach

As rich as grid job monitoring data may be - even including exit codes of jobs - often the exit code can not denote the responsible, faulty grid component, which could be an entire site, only a computing or storage element of a site, or the application submitted by the user. Furthermore, once failures are detected, they need to be diagnosed, understood, and lastly related to the grid component, which hosts the actual fault responsible for erroneous behavior on a grid. The latter can only be accomplished by a person, who can apply knowledge and reasoning to the data [3]. In very simple cases, parts of the interpretation of monitoring data can be performed automatically. For example, on a site with 100% failing jobs, an alarm can be triggered and sent to the person in charge of the site. Thus, the person will investigate, where exactly the failures occure and will reason about their underlying fault.

Currently, the solution to trace back errors to the underlying fault is to investigate the different components' behavior using monitoring tools like GridView [9], the Experiment Dashboard [6], the Service Availability Monitoring (SAM) portal [10] or log files.

In general, the detection and recovery of faults requires an human expert's time and effort. In a production grid infrastructure it is important to provide available and reliable services. Therefore, the time to detect, diagnose, and solve a problem should be minimized and automatized as much as possible.

2.1. Approach

A lot of information is buried deep inside of the huge amount of monitoring data that can be revealed by taking dependencies between grid components into account. Alternatively, this process can be described as "the science of extracting useful information from large data sets or databases", which is according to [4] the definition of data mining.

Data mining is a subdiscipline of artificial intelligence as it emerged from fields like machine learning, pattern recognition, and statistics. One of the widely used methods of data mining is association rule mining (ARM), which generates associations between items in a database.

In our contribution we propose to apply ARM to grid job monitoring data to detect grid components which are related to frequently occurring errors.

We proceed in four steps:

(i) data preparation of previously collected grid job monitoring data,
(ii) generation of association rules,
(iii) reduction of the number of rules by removing redundancies and keeping only interesting rules,
(iv) visualization of a set of rules, which represent extraordinary behavior of grid components, in a intuitive web interface.

Steps one, two, and four are relatively simple and straight forward, but the decision about which rules to keep and which to remove in step three is tricky, time consuming, and most important.

3. Association Rule Mining

This section gives an introduction to association rule mining in general.

Association rule mining is a representatives in the pattern recognition branch of data mining. The objective is to find associations between items of a certain domain. The data is typically stored in a database; the columns are defined by the different items and the rows represent transactions, which consist of a subset of possible items. Let $I = \{i_1, \ldots, i_n\}$ be the set of items, then a row consists of 1s and 0s, denoting whether an item is or is not in the transaction.

An association rule denotes co-existence and dependence of items. A valid association rule is an implication $A \Rightarrow C$ where $A \subset I, A \neq \emptyset$ and $C \subset I, C \neq \emptyset$ are non-empty subsets of I and it applies $A \cap C = \emptyset$. The two disjoint itemsets are called the *antecedent* and *consequent*, respectively. In the literature, they are also referred to as left-hand-side (lhs) and right-hand-side (rhs) of an association rule.

A typical domain for ARM is a supermarket database, which stores all the sets of items every customer purchases. Applying ARM to this data delivers information that was possibly not known before. In this domain an association rule could look like the following:

$$\underbrace{\{bread, butter\}}_{Antecedent} \Rightarrow \underbrace{\{milk\}}_{Consequent} \underbrace{[confidence = 0.9]}_{Quality}$$

This rule states that 90 percent of the people who bought bread and butter also bought milk. This information is not exactly new, but with association rule mining the following not intuitive information has been discovered: most people, who buy diapers also buy beer. A supermarket can profit from this information and use it for example for shelf planning.

The procedure of association rule mining consist of several steps, which are depicted in figure 1. The three inputs are on the very left. Along with the data, there are two threshold numbers, the minimum support and the minimum confidence, which have to be fulfilled by the rules. The minimum support, $minSupp$, is the minimum percentage of records in the database which contain the items of A and C of a rule. A rule meets the minimum confidence, $minConf$, if $minConf\%$ of records that contain the items of A also contain the items of C.

The two main steps of association rule mining are the generation of association rules and the pruning of the large amount of found rules. For the creation of the association rules the *apriori* algorithm[8] was selected, which operates in two steps.

(i) finding frequent itemsets of the input data, taking $minSupp$ into account,
(ii) based on the frequent itemsets, generating association rules, which satisfy $minConf$.

The output of the *apriori* algorithm is a large set of association rules in the order of a thousand. It follows the most important, time consuming, and tricky step, the pruning. Several operations are undertaken to reduce the amount of association rules by removing redundant rules and uninteresting rules. Therewith, the term *interesting* needs to be defined specifically for a domain. According to the definition, there is a variety of possibilities to proceed, which is summarized in [11].

4. ARM with CMS Analysis Job Monitoring Data

In the domain of grid computing, we selected job monitoring data as the information source, which is typically stored in a database. As mentioned above, ARM discovers dependencies between items. Here, items consist of attribute-value pairs. The attributes are job characteristics reported by the monitoring tools and associate to the columns in the database. Typical attributes are user name, application to be executed, dataset, computing element (CE). Each of the attributes has a set of possible values (e.g. for the attribute user name, all different names of user submitting jobs), which form the value-sets to each attribute. A row in the database denotes to

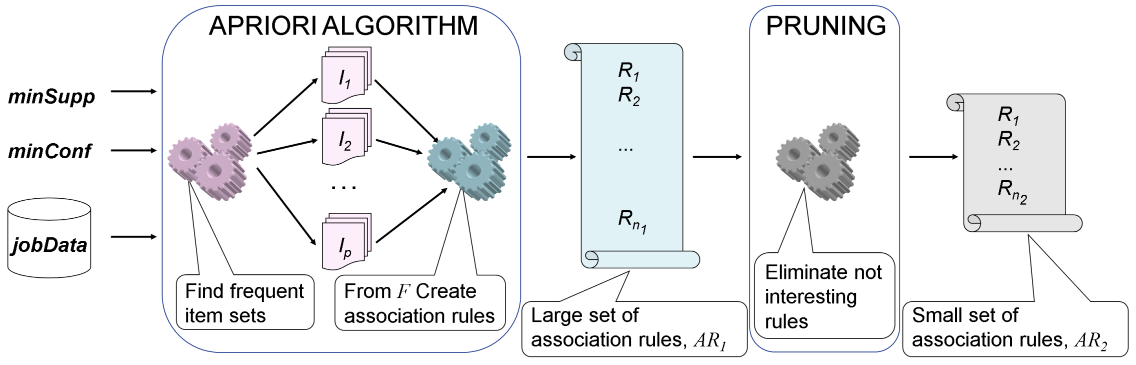
minSupp
minConf
jobData
APRIORI ALGORITHM
I_1
I_2
...
I_p
Find frequent item sets
From F Create association rules
R_1
R_2
...
R_{n_1}
Large set of association rules, AR_1
PRUNING
Eliminate not interesting rules
R_1
R_2
...
R_{n_2}
Small set of association rules, AR_2

Figure 1. Components and steps in the association rule mining process.

a job, which comprises a set of items. Let I be the sum of all items (attribute-value pairs) in a database, then each job consists of a subset of I with the same number of elements. However, for the process of association rule mining, the database structure is transformed to contain transactions. In the transformed database the columns are items and the rows transactions, containing 1s and 0s as described in section 3.

In general, all items are treated equally, but in case of grid job monitoring data with the perspective to detect faulty grid components, there is one restriction: each transaction must contain an exit code, at least denoting success or failure, preferably a more specific exit codes categorization.

As for the domain specific pruning, we undertake three steps. The first trivial and fast step is to remove all rules, which don't have an exit code greater than zero, denoting a failure, in the consequent, since we are only interested in item combinations which lead to an error. The second step consists of testing two rules at a time. Let the antecedent of R1 have n items, the antecedent of R2 be a subset of R1's antecedent, but with $n-1$ items and the consequent of both rules be the same. We perform a X^2 test on R1 with respect to R2. If the test shows no correlation, R1 is pruned, because within the data covered by R2, R1 is not significant [2]. Next, the found rules are sorted and filtered by an interestingness measurement called lift ($lift = \frac{P(AC)}{P(A)*P(C)}$), which is the ratio between the probability of items in the antecedent (A) and consequent C of a rule occurring together and the probability of items in A and items in C occurring separately.

4.1. Experimental Setting

The process depicted in figure 1 has been tested with Compact Muon Solenoid (CMS) analysis job monitoring data. CMS is one of the four central experiments undertaken at CERN in Switzerland. The monitoring data of jobs run on the WLCG is taken from the CMS Experiment Dashboard database[6]. Each record in this database is one submitted job. The columns are the details and status information of a job. Here, we only take a subset of all the available monitoring information, on which we perform association rule mining. The following eight job characteristics (columns) of terminated jobs are selected: the user, the site, the queue of the computing element, the computing element, the worker node, the input physics dataset, the application, and the exit code. As for the time range in which to withdraw jobs, we decided to use twelve hours. It is not reasonable to use a time range larger than twelve hours, because hardly ever a problem stays undetected and relevant for such a long period of time. However, to use less than 12 hours of jobs results in too few jobs with the same characteristics (transactions comprising the same items) to formulate rules, meaning to detect extraordinary behavior of certain grid components.

4.2. Results

Along with above mentioned data input, a $minSupp$ of about 1% and a $minConf$ of 90%, the first step of the ARM process generates about 3000 association rules (see AR_1 in 1). After the threefold pruning, the list of the most interesting observations in the present data can be reduced to six rules on average. This final list of rules is presented in a web interface called QAOES (Quick Analysis Of Error Sources). A possible rule in this domain could look like the following:

$$\{USER = user203\} \Rightarrow \{ERROR = 50115\}\,[support = 0.018, confidence = 0.99]$$

Taking dependencies between items into account, the most significant rule leading to exit code 50115 depends only on one item, the user user203. 99.48% of jobs submitted by user203 failed with exit code 50115. The user most probably has a problem in the program code he submitted for execution. A possible reaction to this association rule is an e-mail to the user to call attention to the faulty behavior of his jobs.

QAOES initially has been implemented to support the CMS analysis support group by detecting faulty components. The web interface has been used by members of the ASTF (Analysis Support Task Force). However, the implementation can be adapted easily to work with different job monitoring data, as long as the one restriction - an exit code among the job characteristics - is fulfilled.

5. Conclusions and Future Work

We applied association rule mining on grid job monitoring data. The process operates in two steps. Firstly, the generation of association rules, which is accomplished by the *apriori* algorithm and secondly, the pruning, which comprises a combination of methods appropriate in this domain to reduce the huge amount of association rules. Each association rule of the output list is an observation in the present data which denotes a faulty grid component. To test our approach we used CMS analysis job monitoring data and received positive feedback from experts in charge of fault recovery about the practicality of the output list.

In the future we will use the specific association rules to collect experts' interpretations and comments to formulate general rules. These rules will build a knowledge base and together with the implementation of an inference engine set up an expert system to further simplify and reduce the time for fault recovery in grids.

References

[1] Alan Ableson and Janice Glasgow, Efficient Statistical Pruning of Association Rules, Conference Proceedings on Principles and Practice of Knowledge Discovery in Databases, p 23, Springer Berlin, 2003.
[2] Bing Liu, Wynne Hsu and Yiming Ma, Pruning and Summarizing the Discovered Associations, KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, p 125, USA: ACM Press, 1999.
[3] Raissa Medeiros, Walfredo Cirne, Francisco Brasileiro and Jacques Sauve, Faults in Grids: Why are they so bad and What can be done about it?, p 18, USA: IEEE Computer Society, 2003.
[4] David J. Hand, Heikki Mannila and Padhraic Smyth, Principles of Data Mining, USA: The MIT Press, 2001.
[5] Daniel T. Larose, Discovering Knowledge in Data, An Introduction to Data Mining, Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 2005.
[6] Julia Andreeva, Benjamin Gaidioz, Juha Herrala, Gerhild Maier, Ricardo Rocha and Pablo P. Saiz, Experiment Dashboard: the monitoring system for the LHC experiments, High Performance Distributed Computing, Proceedings of the 2007 Workshop on Grid Monitoring, p 45, New York, USA: ACM, 2007.
[7] Worldwide LHC Computing Grid (WLCG): http://cern.ch/lcg.
[8] Rakesh Agrawal, Tomasz Imielinski and Arun Swami, Mining Associatioon Rules between Sets of Items in Large Databases, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, p 207, Washington, D.C., USA: ACM Press, 1993.
[9] GridView: Monitoring and Visualization Tool for LCG: http://gridview.cern.ch.

[10] Alexandre Duartea, Piotr Nyczyk, Antonio Retico and Domenico Vicinanza, Global grid monitoring: the EGEE/WLCG case, GMW '07: Proceedings of the 2007 workshop on Grid monitoring, Monterey, California, USA, pp 9-16, New York, NY, USA: ACM, 2007.

[11] Liqiang Geng and Howard J. Hamilton, Interestingness Measures for Data Mining: A Survey, ACM Comput. Surv., vol 38, number 3, p 9, New York, NY, USA: ACM, 2006.