ECA Rule Learning for Resource Management in Grid Computing: Fuzzy Approach

F. Mahan, A. Isazadeh, and L. M. Khanli

Abstract—**Grid computing is important environment for resource sharing and distributed system integration. One discussed field in Grid computing is the resource management. In this paper, Grid-JQA architecture is used that includes active database. We focus resource management with regarding to active rule learning. Rule learning is very significant for efficient rules in active database system. Fuzzy Decision tree can use into rule update. Results shows this approach can be effectively applied for rule learning in such environments because there is attribute with continuous values and splitting at node must be dynamic.**

Index Terms—**Grid computing, rule learning, ECA rule, resource management.**

I. Introduction

Grid computing look likes as distributed and large-scale cluster computing that handle distributed parallel processing and storage and data resources [1],[2]. Resource management is very important because of the lack of central control over the hardware; there is no approach to guarantee that nodes will not secede of the network at different times.

In the grid's architecture, grid resource management system (GRMS) is the central component, which is responsible for disseminating resources information across the grid, accepting requests for resources, discovering and scheduling the suitable resources that match the requests from the global grid resources, and executing the requests on scheduled resources. For some reasons in grid such as geographically distributed, heterogeneous and autonomous in nature, the design and implementation of RMS is challenging [1], [3], [4], [5].

We used Grid-JQA (grid java based quality) an architecture including active grid information server that described in our previous research [6]-[9]. The Grid-JQA that manages by active database provides quality of service on different types of resources [7],[9]. Also, it automatically selects optimal resources using active database ECA rules and requests resource allocation via active grid information server [6], [7], [10], [11].

The active behavior of a system is specified by rules, which describe reactions of the system [12],[13]. Rules known as event-condition-action rules or ECA rules that each rule consists: a rule event, a condition and an action [14].

In this paper, we want to learn ECA rules for resource management. Decision tree learning is one of the approaches for learning [15]-[18]. We can reduce a decision tree to rule so final representation used in this research consists of a rule base created from decision trees. Some rules maybe do not necessarily contain examples of every possible value of the attribute [19] so some rules may fail in some instances. We feel in our case fuzzy logic captures better for this real time environment. In this case integration of decision trees with the knowledge component inherent in fuzzy sets is useful.

We focus on 19 ECA rules as base rules that introduced in [7],[10] for resource management in Grid-JQA. Our goal is to have an update active database system after rule learning is done independently on the new example. So the fuzzy decision tree of each event can be learning a domain with changing membership. This new changing rule set will then be used to classify unseen examples.

This paper is organized as follows: In Section II, we describe ECA rule for resource management in grid computing and Architecture of Grid-JQA. In Section III, we describe ECA rule learning process with using fuzzy decision tree. In section IV, we evaluate our approach. Finally; we conclude the paper in Section V.

II. ECA Rule for Resource Management in Grid

Our ECA resource manager rules [7],[10] select user participation from resource management and the set of optimal resources for it that must provides job execution efficiently [7]. So for improvement of performance, we need to optimize our rule dynamically and learning new rules.

A. Knowledge Model: ECA Rule

ECA rules are one way of implementing this kind of functionality [11], [14]. An ECA rule has the general syntax as follow:

DEFINE RULE *rule_name*
ON *event*
IF *condition*
DO *action*

The *rule_name* is the identifier of the rule. Every rule has a unique *rule_name*. The description of a rule commonly includes three parts:

1.Events : it describes a happening to which the rule may be able to respond.

2.*Conditions : it* specifies what condition must be checked once the rule is triggered and before it is executed. If the result of the condition evaluation is true, the condition is satisfied.

Manuscript received May 22, 2012; revised June 25, 2012.

F. Mahan is a Ph.D student in the CS department, University of Tabriz (mahan@tabrizu.ac.ir).

A. Isazadeh is a professor in the Department of Computer Science at the University of Tabriz, Tabriz, Iran (isazadeh@tabrizu.ac.ir).

L. Mohammad Khanli is currently assistant professor in the Department of Computer Science at Tabriz University, Tabriz, Iran (l-khanli@tabrizu.ac.ir).

3.Actions : The action describes the task to be carried out by the rule if the relevant event has taken place and the condition has evaluated to true.

For example in grid for resource management, one of the rules that we used is [7]:

AGIS_Advertise_Rule
ON Event AGIS_Advertise
IF Condition select agis from AGIS_UPPER
Where agis.threshold = min(*.threshold)
DO Action home.Advertise (agis);

B. Grid-JQA Architecture

Grid-JQA system architecture include grid portal, active grid information server (AGIS), fault detector, and GRAM (for more detail sees [7]). ECA rules support the semantics of the grid resource management. Here, we define active learning grid resource server that includes efficient ECA resource management rules for resource selection and job scheduling [7]-[9].

Figure 1 from [7] shows the architecture of rule engine Grid-JQA that we extend it with adding rule learning. In this system, rule learning simulates new condition and example and make new rule base of original rules by fuzzy decision tree.

C. Resource Management

Processor, network, and memory are the most important resources in grid [20] that user can assign weight for each parameter that shows the importance of the parameter regardless for memory. Since grid is real-time environment, when the resource capabilities are changed, maybe the fault detector informs the AGIS by fault event.

We introduce fuzzy decision tree for ECA rules in active database for managing environmental changes.

III. Fuzzy Decision Tree for ECA Rule Learning

A fuzzy decision tree combines fuzzy concept with symbolic decision trees. This feature provides the representative power of decision trees with the knowledge component inherent in fuzzy logic that leading to better robustness and efficient [21],[22]. Each attribute partition into fuzzy sets and assign membership degrees to original values of it regarding to membership functions. In actual, each fuzzy decision tree use assigned fuzzy membership [23] to induce an explicit fuzzy decision tree [22].

The some advantages of fuzzy decision tree are [24]:1. Better utilization of resources 2. Interpretability 3.Manaoement of uncertainties.

A *fuzzy decision tree* (see e.g. Figure 2) is a tree that every edge is annotated with a condition, and every leaf is submitted with a fuzzy set over *class (C)*. We consider only binary trees in this paper, where one of the conditions at the outgoing edges of a node is chosen, and the condition at the other outgoing edge is the negation of the test condition [13], [22]. Another property is that each condition is used at most once in each path from root to leaf [13].

Whenever an example matches the conditions the edges of a path are annotated with, we expect the example to belong to the classes regarding to the fuzzy set the leaf is annotated with. E.g., in Figure 4 along the edges labelled MAXTHRESHOLD gives us a clue, that the interest of the user in MAXTHRESHOLD is classified to positive with a degree of truth 0.82 and negative with a degree of truth 0.18. One should observe that unlike in decision trees based on classical logic, the conditions on the outgoing edges of a node are not necessarily mutually exclusive: e.g. an example can be both MAXTHRESHOLD and non– MAXTHRESHOLD to a non–zero degree of truth. So we have to combine the

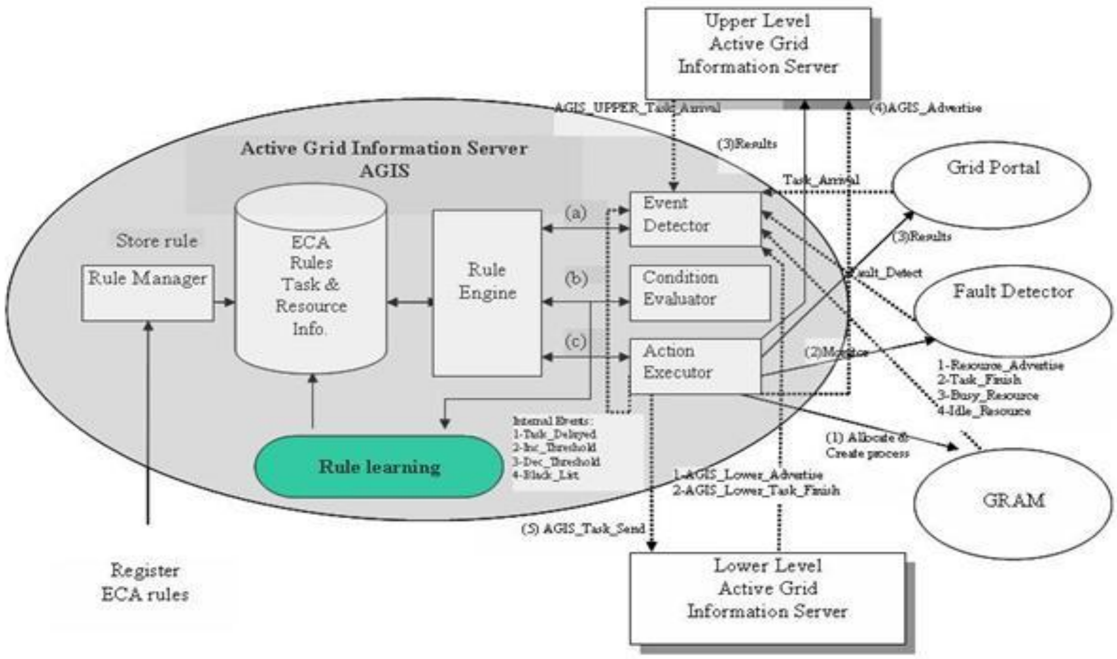

Fig. 1. Extended architecture of Grid-JQA & AGIS ECA rule engine with rule learning

clues given by all paths into one final result. In this paper, we focus rule learning that use fuzzy decision tree for rule update.

A. Rule Update

Rule learning means update and create new rule regarding to original rules and real examples. Here our rule is ECA rule. We use fuzzy decision tree for describing each original rule [25]. The root of each initial tree is the related event of each rule. Here, End of each tree are tow leafs at most because we suppose binary fuzzy tree. Fuzzy decision tree provide adaptive and more performance for updating ECA rules because attributes in rules unlike static ECA rules have range value base on membership of it.

When related event occurred, system must refer to related original fuzzy tree. If related original fuzzy tree respond to event, the rule does not need to change. But if it do not consider to event, the rule need to update or create new rule.

In actual, we need update and create new rule according to real examples in three cases:

Case 1:

For such learning, we need changing membership value of parameter that must consider. For example, in original rule, that describe in section 4, MAX_THRESHOLD is a parameter that the degree of it maybe change with membership function [23] regarding to received real example .

Case 2:

In this case some real example may have some new condition into original rule. So new fuzzy decision tree must be created base on original tree and add new condition. The using times new fuzzy decision tree determines adding it in original database.

Case 3:

We need new ECA rules that must consider some real examples because at first maybe we did not consider these rules that happened in real. So we can generate new rule from instances that the new rules must not have conflict with other rules.

IV. Case Study

We focus on 19 ECA rules as base rules that introduced in [7],[10] for resource management in Grid-JQA. Our goal is to have an update active database system after rule learning is done independently on the new example.

For evaluation our approach, we use a new training data set that happened in real environment that fuzzy decision trees learn from this training set. Let follow rule is one of original rule:

Inc_Threshold_Rule
ON Event Inc_Threshold
IF Condition (home.Threshold < _MAX_THRESHOLD)
DO Action home.IncThreshold( );
ELSE Action home.IssueEvent (AGIS_Advertise);// under loaded

In this rule, MAX_THRESHOLD is a continuous attribute that expert determines. we need to change it base of examples and new data. When value of MAX_THRESHOLD is static, maybe some examples not respond so it must be learned during the time.

Learning is done with fuzzy decision tree that described in section III. In figure 2, we presented fuzzy decision tree of above ECA rule.

When we use active ECA rules for resource management, we may be received event with different value of attribute. So we could not respond to some events because the value of threshold was static. Hence when our threshold change with fuzzy degree and used fuzzy decision tree for managing events, we could respond more event successfully.

V. Conclusion

We extended Grid-JQA system architecture. This system includes Active Grid Information Server (AGIS) with ECA resource manager rules and ECA fault manager rules. We suggested rule learning based on fuzzy decision tree in resource management in Grid computing for learning new example and data. Learning provide resource management be more efficient. As future work, it is also possible to update or invalidate the existing rules by mining the triggered rules history.

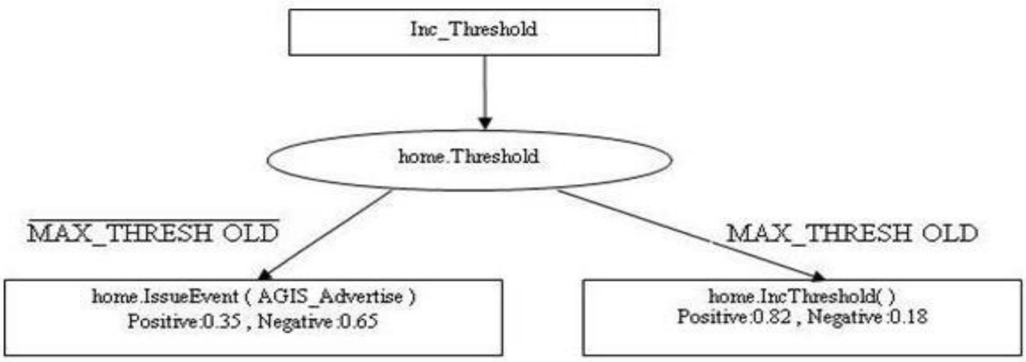

Fig. 2. Fuzzy decision tree of example ECA rule

REFERENCES

[1] H. Jin, Y. Pan, N. Xiao, and J. Sun , "An active resource management system for computational grid, " *GCC 2004*, LNCS 3251, 2004, pp. 225–232.

[2] A. Galstyan, K. Czajkowski, and K. Lerman, "Resource allocation in the grid using reinforcement learning," *AAMAS'04*, New York, USA, July 19-23, 2004.

[3] T. Abdullah, K.L.M. Bertels, and S. Vassiliadis, "Adaptive agent-based resource management for grid," in Proc. of *12th ASCI Conference*, Lommel, Belgium, June 2006, pp. 420-428.

[4] U. Schwiegelshohn and R. Yahyapour, "Resource management for future generation grids CoreGRID," technical report, Number TR-0005, May 19, 2005.

[5] A.L. M. Ching, L. Sacks, P. McKee, "Super resource-managment for grid computing, "2002.

[6] L. M. Khanli and M. Analoui, "Active grid information server for grid computing," *The Journal of Supercomputing*, 2008.

[7] L. M. Khanli and M. Analoui, "An approach to grid resource selection and fault management based on ECA rules*," Future generation computer systems*, 24(4), pp. 296-316, 2008.

[8] L. M. Khanli and M. Analoui, "Grid-JQA a New architecture for QoS-guaranteed grid computing system," *14th Euromicro conference on parallel, distributed and network-based processing*, Feb 15-17, France, 2006.

[9] L. M. Khanli and M. Analoui, "Grid-JQA : Grid Java based Quality of service management by Active database," *4th Australian symposium on grid computing and e- Research(AusGrid 2006)*, Australia, 2006.

[10] V. Kravtsov, T. Niessen ,A. Schuster, W. Dubitzky, and V. Stankovski, "Grid resource management for data mining applications," 2006.

[11] M. Zoumboulakis, G. Roussos, and A. Poulovassilis, "Active rules for sensor databases," *Proceedings of the first workshop on data management for sensor networks (DMSN 2004)*, Toronto, Canada, August 30th, 2004.

[12] J. Demsˇar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research 7*, pp. 1–30, 2006.

[13] M. Guetova and S. H ̈olldobler, H.P. St ̈orr, "Incremental fuzzy decision trees," *Advances in artificial intelligence , Computer Science*, Volume 2479, 2002.

[14] M. Dai and Y. L. Huang, "Data mining used in rule design for active database systems," in *Proceedings of the fourth international conference on fuzzy systems and knowledge discovery*, IEEE Computer Society, Washington, DC, USA, 4, 2007, pp.588-592.

[15] L. O. Hall, N. Chawla, and K. W. Bowyer, "Decision tree learning on very large data sets," *IEEE international conference on systems, man and cybernetics*, October 11-14, San Diego, CA, USA, 1998.

[16] J. J. Grefenstette, C. L. Ramsey, and A. C. Schultz, "Learning sequential decision rules using simulation models and competition," *Machine learning*, 5(4), pp.355-381, 1990.

[17] J. Mingers, "An Empirical comparison of selection methods for decision tree induction," *Machine learning*, 3 (4), pp. 319-342, 1989.

[18] E. Frank and K. Huber, "Active learning of soft rules for system modelling," *EUFIT'96*, Aachen, September 2-5, 1996.

[19] D. A. CHIANG, W. CHEN, Y. FANWANG, and L. J. HWANG, "Rules generation from the decision tree," *Journal of information science and engineering*, 17, pp.325-339 , 2001.

[20] D. C. Vanderster, N. J. Dimopoulosb, R. Parra-Hernandez, and R. J. Sobie, "Resource allocation on computational grids using a utility model and the knapsack problem," *Future generation computer systems journal*, 2008.

[21] S. K. Murthy, "Automatic construction of decision trees from data: a multi-disciplinary survey," *Data mining and knowledge discovery*, 2, pp.345–389, 1998.

[22] S. Hashemi and Y. Yang, "Flexible decision tree for data stream classification in the presence of concept change, noise and missing Values," *LLC 2009, Data Min Knowl Disc*, 19, pp.95–131, 2009.

[23] L. A. Zadeh, "Fuzzy sets ," *Information and Control*, 8, pp.407–428, 1965.

[24] X. Boyen and L. Wehenkel, "Automatic induction of fuzzy decision trees and its application to power system security assessment," *Fuzzy sets and systems journal*, 102, pp. 3-19, 1999.

[25] L. O. Hall and P. Lande, "The generation of fuzzy rules from decision trees," *Journal of advanced computational intelligence and intelligent informatics (JACIII)*, 2, pp. 128-133, 1998.

F. Mahan received her B.Sc. and M.Sc. degrees in Computer Engineering in 2002 (Iran Azad university), 2005 (Iran University of Science & Technology, Tehran, Iran) respectively. She is instructor at Department of computer Engineering, Islamic Azad University,Tabriz branch, Tabriz, Iran. Now she is Ph.D student in CS department, University of Tabriz. Her research interests are Artificial intelligent and Robotics, Grid computing and multi-agent systems.

A. Isazadeh received a B.Sc. degree in Mathematics from University of Tabriz in 1971, an M.S.Eng. degree in Electrical Engineering and Computer Science from Princeton University in 1978, and a Ph.D. degree in Computing and Information Science from Queen's University. He is a professor in the Department of Computer Science at University of Tabriz.

L. M. Khanli received her B.S. (1995) from Shahid Beheshti University Tehran, Iran, M.S. (2000) from IUST (Iran University of Science and Technology) University and a Ph.D. degree (2007) from IUST (Iran University of Science and Technology) University. All are in computer engineering. She is currently assistant professor in the Department of Computer Science at Tabriz University. Her research interests include grid computing and Quality of Service management.