Research Journal of Applied Sciences, Engineering and Technology 4(21): 4546-4551, 2012
ISSN: 2040-7467

Submitted: June 20, 2012 Accepted: July 23, 2012 Published: November 01, 2012

Load Balancing in Local Computational Grids within Resource Allocation Process

Rouhollah Golmohammadi and Hadi Shahriar Shahhoseini
Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran

Abstract: A suitable resource allocation method in computational grids should schedule resources in a way that provides the requirements of the users and the resource providers; i.e., the maximum number of tasks should be completed in their time and budget constraints and the received load be distributed equally between resources. This is a decision-making problem, while the scheduler should select a resource from all ones. This process is a multi criteria decision-making problem; because of affect of different properties of resources on this decision. The goal of this decision-making process is balancing the load and completing the tasks in their defined constraints. The proposed algorithm is an analytic hierarchy process based Resource Allocation (ARA) method. This method estimates a value for the preference of each resource and then selects the appropriate resource based on the allocated values. The simulations show the ARA method decreases the task failure rate at least 48% and increases the balance factor more than 3.4%.

Keywords: Analytic hierarchy process, balance factor, failure rate, grid computing, multi-criteria decision making

INTRODUCTION

The concept of using distributed resources for solving large-scale problems became popular by development of computer networks. It caused forming the new environment of grid computing (Foster and Kesselman, 1999). The grid contains processing resources and the users send their tasks to grid to be processed by its resources. So, such an environment is not usable without proper resource management.

Foster and Kesselman (1999) define 3 layers for grid environment: the lowest layer or resource layer is the infrastructure of grid and consist of computational resources and communication networks. The middle layer is grid management system and its task is allocating resources to applications in a safe and stable environment. The applications that use grid environment are in highest layer or application layer.

The management entities in middle layer are responsible for resource management, service management, security management, etc. The grid resource management is the fundamental part of this layer that communicates directly with the grid infrastructure. The main subset of resource management system is grid scheduler and its role is discovering and allocating the grid resources. In grid environment, the load of some resources may decreases while the others are overloaded, for the reason that the tasks have different receiving time and properties and the computational resources are heterogeneous. This unbalanced load may cause failure of some tasks on busiest recourses, while the others remain unused. Therefore, the resource allocation and load balancing are the main issues, which are needed in resource management for utilizing the grid (Li *et al.*, 2009).

The majority of the previously developed methods for resource allocation do not support load balancing (Li and Cheng, 2009; Soon and Sim, 2010). These methods usually need a reallocation process to distribute load equally between resources (Goyal, 2011; El-Zoghdy, 2012). This load exchange process causes increment of costs and waiting time and decrement of quality of service. Therefore, it seems that it is necessary to balance the load within allocation process. It is a Multi Criteria Decision-Making (MCDM) problem, while this process bases on several qualitative and quantitative attributes.

In this study, a load-balanced method is proposed to distribute the computational load by defining an MCDM problem and selecting appropriate criteria. The Analytic Hierarchy Process (AHP) is the most important and the most popular method for solving MCDM problems (Saaty, 2008), so an AHP based Resource Allocation (ARA) method is proposed in resource selection process. The selection attributes in ARA method are tasks constraints and resources properties. The other criterion is the preference of time over budget from user perspective. This provides the user to decide on its task to be processed earlier or cheaper.

LITERATURE REVIEW

There are many works on resource allocation on traditional distributed computing systems. Almost none

Corresponding Author: Hadi Shahriar Shahhoseini, Electrical Engineering Department, Iran University of Science and Technology, Tehran, Iran. Tel: +98-2177240487, Fax: +98-2177240486

of them are applicable to the grid infrastructure, while the resources in grid are heterogeneous and geographically distributed, so researchers try to present new methods for grid system. Buyya *et al*. (2000) developed Time-Optimized and Cost-Optimized Allocation (TOA, COA) in Nimrod/G project, which are the 2 primary grid resource allocation algorithms. Min-Min and Max-Min are the other simple resource allocation algorithms (He *et al.*, 2003; Etminani and Naghibzadeh, 2007). The Min-Min algorithm assigns smallest task to fastest resource. These are the basic algorithms and many of the other works have used them for comparing with and showing the proper application of their methods, e.g., Kousalya and Balasubramanie (2008) and Kamalam and Muralibhaskaran (2010). The last two algorithms wait until more than a specific number of tasks submitted and then allocate all unmapped tasks in 1 phase. On the other hand, the formers make the decisions for each individual task as it is received to scheduler. It should be said that the proposed algorithm, ARA, schedules the tasks on-line and separately, as TOA and COA. The goal of TOA algorithm is to select a resource that reduces the completion time of tasks. In COA, the scheduler tries to find cheapest resources to apply tasks to them. TCOA (Time and Cost Optimized Allocation) is another approach based on TOA, but after selecting the fastest resources, a process of cost optimization is applied on selected ones.

Economic model, the other solution, is introduced by Buyya *et al*. (2002), which uses market mechanisms, e.g., different kinds of auction (Grosu and Das, 2004; Kant and Grosu, 2005). A problem in these methods is where the cost of resources is the only allocation criterion, while the resource cost cannot introduce it completely. Izakian *et al*. (2010) proposed an algorithm based on Continuous Double Auction (CDA) for resource allocation and load balancing. CDA is a recently developed method based on economic model and has better results than other methods of economic model (Qureshi *et al*., 2011). It has also the same propose of load balancing as ARA. Kurowski *et al*. (2006, 2007) suggested a multi-criteria resource allocation method. They consider the different possible mappings from tasks to resources and evaluate each allocation using defined functions. In these methods, if the number of tasks or resources increases, the extent of possible solutions will enlarge exponentially. Wieczorek *et al*. (2009) proposed a multi-criteria optimization to solve this problem, but even it needs to solve 2 problems of MCDM and optimization. The proposed method in this study is also a multi-criteria method, but it uses resource properties instead of allocation characteristics. Therefore, the solution extent does not grow up exponentially when the number of resources increases. There is no need for optimization in this process and so the resource selection becomes easier.

COMPUTATIONAL RESOURCE ALLOCATION

The analytic hierarchy process: The analytic hierarchy process decomposes the decision into 4 steps for solving decision-making problems (Saaty, 1980). The 1st step defines the problem and selects the needed criteria for taking best decision. The 2nd step organizes the decision hierarchy. The goal of the decision is at the top of the hierarchy. The intermediate levels contain main criteria and subcriteria. Defining the subcriteria is not necessary and depends on the main criteria. The lowest level includes a set of the alternatives. The 3rd step produces pairwise comparison matrices for the elements of each level. In these matrices, the elements should be compared with respect to the higher-level criterion. The eigenvector of this matrix represents the relative preference of elements at each level, which is named Relative Value Vector (RVV).

Finally, the last step uses the obtained preferences at each level for weighting the preferences of the next level. The final preference of each element is calculated by summing the weighted values. This process of weighting and summing should be continued to obtain the final preferences of alternatives at the lowest level.

AHP based resource allocation: The proposed method follows the 4 steps of the analytic hierarchy process for selecting the computational resources. The problem is resource allocation and load balancing; i.e., a computational resource should be selected as to execute the task in its defined time and budget constraints and the total computational load be distributed equally between resources. Therefore, the cost of resources (rc), their MIPS rating (rp), the number of processing elements (pe), average waiting time for executing the received tasks (wt) and the received load (ru) are selected as decision criteria. The last two criteria are chose for balancing the load between resources. Two subcriteria are considered for the number of processing elements, which are the total number of processing elements and the number of free processing elements. It is considered because a computational resource may have many processing elements, while the most of them or all of them are busy.

Figure 1 shows the different levels of decision hierarchy. The goal of decision is at the top level of hierarchy, as it is shown in this figure and the decision criteria and sub criteria are at the intermediate levels. The lowest level contains computational resources, which are decision alternatives. The next step is constructing the pair wise comparison matrices. The elements of intermediate levels matrices are valued manually, because they are constructed one time and will not be changed during allocation process. Each user in grid environment has different

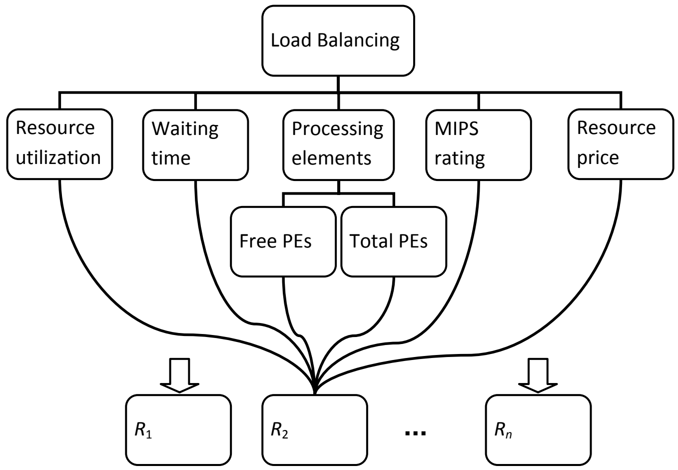

Fig. 1: The decision hierarchy structure

Table 1: Predefined relative value vectors

s_j	rc_i	rp_i	pe_i	wt_i	ru_i
1	0.200000	0.200000	0.100000	0.400000	0.100000
1/4	0.237865	0.061076	0.118933	0.463194	0.118933
4	0.061076	0.237865	0.118933	0.463194	0.118933
1/7	0.258553	0.038250	0.140639	0.421918	0.140639
7	0.038250	0.258553	0.140639	0.421918	0.140639

necessities, so a parameter (named *s*) is considered for providing their preferences. This parameter determines time to budget preference of a user. Table 1 shows the obtained relative value vectors of constructed matrices for different values of parameter*s*.

The pair wise comparison matrix of lowest level cannot be valued manually, whereas its elements are calculated based on updated information of resources. Equation (1) compares the alternatives and assigns a value to each pair:

$$K = \left[\eta \times \left| \frac{A_i - A_j}{A_{\max} - A_{\min}} \right| + 1 \right] \tag{1}$$

where, A_i and A_j are the values of A attribute for *i*-th and *j*-th resources, respectively. A_{max} and A_{min} are the maximum and minimum values of this attribute. The coefficient of η should be 8.0, whereas the value of K is a Saaty scaled number (Saaty, 1980) (a natural number in the range of [1, 9]). c_{ij}, the *i*-th row *j*-th column element of pairwise comparison matrix is obtained from (2):

$$c_{ij} = \begin{cases} K & : A_i > A_j \\ 1 & : A_i = A_j \\ K^{-1} & : A_i < A_j \end{cases} \tag{2}$$

This equation is usable just if $A_i > A_j$ means the preference of *i*-th resource over *j*-th 1; e.g., the MIPS rating and the number of processing elements. Equation (3) should be used instead of (2) for other conditions, i.e., the pairwise comparison matrix of the processing cost must be constructed by (3):

$$c_{ij} = \begin{cases} K & : A_i < A_j \\ 1 & : A_i = A_j \\ K^{-1} & : A_i > A_j \end{cases} \tag{3}$$

Each constructed matrix has a consistency ratio (Saaty, 2008) which defines the assigned values to its elements are acceptable or not. The assigned values are acceptable only if the consistency ratio is less than 0.1. The relative value vectors of these matrices are calculated by approximate method of geometric mean (Ghodsipour, 2010). The final preferences of resources are estimated using these weighted values. Then the incapable resources to provide the task constraints will be omitted. Finally, a random selection process from 2 best resources is considered to reduce the effect of negligible differences between assigned values to resources. In this process, two uniformly distributed random numbers in the range of [0, 2] and [0, 1] are selected for the first and second preferred resources, respectively. It just increases the chance of selection of the first resource. The ID number of the resource with the larger assigned random number will be sent to the user.

SIMULATION AND RESULTS

The Gridsim (Buyya and Murshed, 2002) (a popular simulator of computational grid environment) is used for evaluating the proposed method. The grid environment contains 20 users, a scheduler and 8 resources (with total number of 60 processing elements), which is acceptable for a local grid. The resource characteristics are as WWG standard (Sulistio and Buyya, 2004; Golmohammadi and Shahhoseini, 2010). Each user has 50 tasks and sends them to the grid scheduler with a poison distribution. The task length is a uniformly distributed random number in the range of (10000, 100000) MI (Million Instructions). The presented results for each value of system load are the average of 50 times of running the simulation. Two criteria of the task failure rate and the balance factor are considered for evaluate the success ratio of the proposed method. The task failure rate is the ratio of the number of the completed tasks (in their constraints) to all received tasks. The balance factor, BF, defines the load balancing level and it is calculated from (4):

$$BF = \left(1 - \frac{d}{ru}\right) \times 100\% \tag{4}$$

where, d and ru are the standard deviation and the average of resource utilization rate, respectively. Their mathematical definitions are presented in (5) and (6):

$$d = \sqrt{\frac{\sum_{i=1}^{m}\left(ru - ru_i\right)^2}{m}} \tag{5}$$

$$ru = \frac{1}{m}\sum_{i=1}^{m} ru_i \tag{6}$$

In these equations, ru_i denotes the resource utilization rate of i-th resource and m is the total number of computational resources in the network. The simulations have been done for COA, TOA, TCOA, CDA and proposed method (ARA). The 3 formers are the main methods of resource allocation in grid, which are based on cost, time and time-cost optimization, respectively. The CDA method is one of the newest methods based on continuous double auction model, which has the better results comparing with the other methods of market model. It is also has the same goal of load balancing as ARA method. The results are presented for different values of system load. The concept of system load is defined by ratio of total received load to total load that grid can execute during a specified time. Figure 2 shows the task failure rate of these methods for different system loads. The ARA method decreased the average of task failure rate 98, 79, 78 and 48% comparing with COA, TOA, TCOA and CDA methods, respectively. In low loads, the resources have enough time to process received tasks, because of significant period between task receiving times, so most of the tasks are executed successfully in their

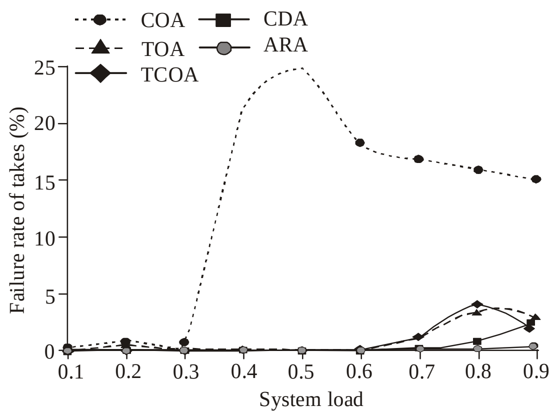

Fig. 2: Failure rate of computational tasks

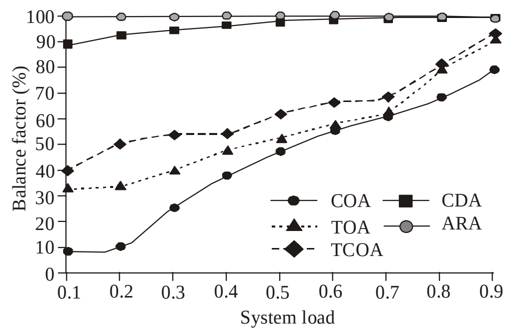

Fig. 3: The load balancing level

constraints for all methods. On the other hand, in high loads, the numerous tasks that are sent to each resource, makes some tasks to be failed.

It seems that the task failure rate should increase by increasing the system load. Figure 2 shows that it is true for ARA and CDA, but there is a maximum at load 0.5 for COA method. The reason is sending the maximum amount of possible load to cheaper computational resources, at this load. For higher loads, the scheduler sends some tasks to the faster processing resources to ensure the time constraints, which causes the failure rate to be decreased. A similar case happens at load 0.8 for TOA and TCOA. Ascendant of the failure rate curve in CDA method is for distributing the load between resources equally. There are not such variations in the failure rate curve of ARA, because the sent load to the resources is one of the important parameters in its resource allocation process and also it is not the only parameter. The load balancing level is an important factor in resource allocation problems. Figure 3 shows the balance factor of considered methods. In COA, TOA and TCOA methods, the balance factor decreases by decreasing the system load, because in lower loads there is enough time for best resources to execute more tasks. In CDA method, using auction model for resource allocation causes the users to access to high quality resources in lower loads, which makes the lower balance factor in these loads. In ARA method, the importance of the sent load to each resource in allocation process makes the balance factor to be almost fixed for all system loads. This causes ARA to have 2 advantages over other methods. First, the sudden changes in the amount of system load do not decrease the system performance. The other advantage is maintaining the utilization of all resources at each value of system load, which increases the reliability to grid from resource perspective.

The balance factor of ARA method is more than 98.8%, so it can be said that the system load is almost distributed equally between resources for all loads. The ARA method increases the load balancing level 127, 79, 57 and 3.4% in comparison with COA, TOA, TCOA and CDA methods, respectively.

CONCLUSION

Resource allocation and load balancing are the main problems in grid management unit and affect grid performance significantly. On the other hand, the resource reallocation for balancing the load increases the network overload and waiting time of tasks. Therefore, it is so important to use an algorithm, which balances the received load through allocation process. In this study, the proposed method, ARA, balanced the load within resource allocation process, by defining a multi-criteria decision making problem. The Gridsim simulator was used for evaluating the ARA method. Three main methods of resource allocation and a new method based on market model, which has the same goal of load balancing as ARA, were also simulated. The simulation results shows that the ARA decreases the task failure rate at least 48% and increases the balance factor more than 3.4%.

In future works, removing the improper resources before calculating their relative value vector can be proposed; considering that it causes the decrement in number of resources but the increment in number of criteria.

REFERENCES

Buyya, R., D. Abramson and J. Giddy, 2000. Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid. Proceedings of the 4th International Conference on High Performance Computing in Asia-Pacific Region, Beijing, May 14-17, pp: 283-289.

Buyya, R. and M. Murshed, 2002. GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. J. Concur. Comp. Prac. Exp., 14(13-15): 1175-1220.

Buyya, R., D. Abramson, J. Giddy and H. Stockinger, 2002. Economic models for resource allocation and scheduling in grid computing. J. Concur. Comp. Prac. Exp., 14: 1507-1542.

El-Zoghdy, S.F., 2012. A hierarchical load balancing policy for grid computing environment. Int. J. Comp. Net. Info. Sec., 5(1): 1-12.

Etminani, K. and M. Naghibzadeh, 2007. A min-min max-min selective algorithm for grid task scheduling. Proceedings of the 3rd IEEE/IFIP International Conference in Central Asia on Internet, Tashkent, Sep. 26-28, pp: 1-7.

Foster, I. and C. Kesselman, 1999. The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann Publishers, San Francisco.

Ghodsipour, S.H., 2010. The Analytic Hierarchy Process (AHP). 7th Edn., Amir-kabir University Publisher, Tehran, (in persian).

Golmohammadi, R. and H.S. Shahhoseini, 2010. A Grid Resource Allocation Method Based on Analytic Hierarchy Process. Proceedings of the 5th International Symposium on Telecommunications,Tehran, Dec. 4-6, pp: 187-192.

Goyal, S.K., 2011. Adaptive and dynamic load balancing methodologies for distributed environment: A review. Int. J. Eng. Sci. Tech., 3(3): 1835-1840.

Grosu, D. and A. Das, 2004. Auction-based resource allocation protocols in grids. Proceedings of the 16th International Conference on Parallel and Distributed Computing and Systems, MIT Cambridge, Nov. 9-11, pp: 20-27.

He, X., X. Sun and G. Laszewski, 2003. A Qos guided min-min heuristic for grid task scheduling. Special Issue on Grid Computing, J. Comp. Sci. Tech., 18(4): 442-451.

Izakian, H., A. Abraham and B. Tork Ladani, 2010. An auction method for resource allocation in computational grids. J. Fut. Gen. Comp. Sys., 26(2): 228-235.

Kamalam, G.K. and V. Muralibhaskaran, 2010. A new heuristic approach: Min-mean algorithm for scheduling meta-tasks on heterogeneous computing systems. Int. J. Comp. Sci. Net. Sec., 10(1): 24-31.

Kant, U. and D. Grosu, 2005. Double auction protocols for resource allocation in grids. Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, Apr. 4-6, pp: 366-371.

Kousalya, K. and P. Balasubramanie, 2008. An enhanced ant algorithm for grid scheduling problem. Int. J. Comp. Sci. Net. Sec., 8(4): 262-271.

Kurowski, K., J. Nabrzyski, A. Oleksiak and J. Weglarz, 2006. Scheduling jobs on the grid-multicriteria approach. J. Comp. Meth. Sci. Tech., 12(2): 123-138.

Kurowski, K., A. Oleksiak, J. Nabrzyski, A. Kwiecien, M. Wojtkiewicz, M. Dyvzkowski, F. Guim, J. Corbalan and J. Labarta, 2007. Multi-criteria grid resource management using performance prediction techniques. In: Integrated Research in GRID Computing. Gorlatch, S. and M. Danelutto (Eds.), Springer, pp: 215-225.

Li, Y., Y. Yanga, M. Mab and L. Zhou, 2009. A hybrid load balancing strategy of sequential tasks for grid computing environments. J. Fut. Gen. Comp. Sys., 25: 819-828.

Li, Z.J. and C.T. Cheng, 2009. An evolutionary game algorithm for grid resource allocation under bounded rationality. J. Concur. Comp. Prac. Exp., 21(9): 1205-1223.

Qureshi, K., B. Nazir and M.A.Shah, 2011. Markup based continuous double auction for resource allocation in market grid. Eng. E-Trans., 6(1): 50-54.

Saaty, T.L., 1980. The Analytic Hierarchy Process. McGraw Hill, New York.

Saaty T.L., 2008. Decision making with the analytic hierarchy process. Int. J. Serv. Sci., 1(1): 83-98.

Son, S. and K.M. Sim, 2009. Relaxed time slot negotiation for grid resource allocation. J. Grid Dist. Comp., 63: 25-32.

Sulistio, A. and R. Buyya, 2004. A grid simulation infrastructure supporting advance reservation. Proceedings of the 16th International Conference on Parallel and Distributed Computing and Systems, MIT Cambridge, Nov. 9-11, pp: 1-7.

Wieczorek, M., A. Hoheisel and R. Prodan, 2009. Towards a general model of the multi-criteria workflow scheduling on the grid. J. Fut. Gen. Comp. Sys., 25(3): 237-256.