Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Resource Management System for Computational Grid Building

Goran Martinovic
Faculty of Electrical Engineering
Josip Juraj Strossmayer University of Osijek
Osijek, Croatia
goran.martinovic@etfos.hr

***Abstract*— The developed DnD Grid system aims to improve building of distributed computer environments by dedicated and non-dedicated resources. The proposed approach supposes overdimensioned static performance and dynamic underloading of non-dedicated resources. Efficiency, availability and autonomy of such distributed heterogeneous environment depend on platform, application, mapping and resource owner/service user (human). The developed resource management system enables creation of a ranking list of owners/users and their resources, monitoring of parameters of the established virtual organization during application execution, as well as improvement of system performance by careful mapping resources and owners/users. The obtained experimental results show that the proposed approach and the corresponding mapping algorithm can significantly improve system performance.**

***Keywords*—computational grid, cooperation, non-dedicated resources, virtual enterprise.**

I. Introduction

Based on [12] and [17], it can be said that the following holds for a computational grid: it is a virtual pool of resources, a user can access the pool but not individual nodes, access to resources may be restricted, a user has little or no knowledge about each resource, resource spans multiple trust domains, the pool can contain a great number of dynamic and heterogeneous elements. According to grid definitions, we can say that a grid represents one form of a virtual enterprise (VE) or a virtual organization (VO).

According to [2], a virtual enterprise is a temporary alliance of companies for the lifetime of a common project, a solution for a problem, or joint production of service or a product. The virtual enterprising paradigm considers a temporary alliance of enterprises that come together to share skills and resources in order to respond better to business opportunities, and whose cooperation is supported by computer networks [19].

A computational grid provides significant support to scientific, but also industrial and business applications [1]. Computer intensive applications are still in large part executed on dedicated resources. Currently, a lot of computers connected by fast networks are working underloaded. Engagement of those non-dedicated resources provides numerous positive consequences [15], but also increases heterogeneity and dynamics of environment. This requires special attention in resource management. Availability of engaged non-dedicated resources depends mostly on human cooperation [6]. A human can be owner, user or owner/user of resources. Computational grid establishment on non-dedicated resources corresponds to computer supported cooperative work (CSCW). From this point of view, in the worst case a human can be considered a selfish owner, i.e. a greedy resource user.

By connecting the provider with a user of services, a new form of virtual organization is always formed [9]. Ownership of a considerable part of non-dedicated resources, the will and/or interest in its concession, as well as the need for high computing power are the main initiators of establishing not only enterprise grids, but also global grid organizations. Such organizations very often include research institutions working with data intensive, but often off-line applications. According to [10], the paradigm of virtual enterprising challenges the way manufacturing systems are planned and managed.

In order to make building of a distributed computer system, which also includes underloaded non-dedicated resources, easier, the DnD (Dedicated and Non-Dedicated) Grid system is developed. This system enables registration and self-description of owners/users and their resources, monitoring of static and dynamic platform parameters, application, as well as definition and adjustment of the mapping procedure. On the basis of the mentioned parameters, a ranking list of owners/users and associated resources is generated. Based on the acquired values of dynamic parameters, the DnD Grid system constantly tries to improve the established computer environment. The developed DnD Grid system matches the user execution application to engaged machines, allowing users to use grid services at the same time.

Section II describes distributed heterogeneous computer environment and parameters from the level of application, platform, resource management procedure and owner/user. In Section III, the DnD Grid system for establishing a computational grid by dedicated and non.-dedicated resources, as well as a ranking list generation procedure are presented. The proposed approach is evaluated in a simplified experimental environment in Section IV.

II. Distributed Heterogeneous Environment

According to [5], [8] and [18], today's distributed computer systems (especially computational grids, clouds, wireless and mobile environments) are typically very heterogeneous and dynamic. They can be described by parameters of platform (computer resources), user applications and resource management (mapping) procedures [16].

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

Environment parameters can be static and dynamic. By using these parameters, the mapping procedure must enable a successful interface between the application requests and available platform capacities. In this case, resource management is carried out only on the machine level. This procedure does not include the most unpredictable "component" - a human. The above mentioned heterogeneity model can be extended by some resource owner/user parameters. They can also be supervised on the machine level.

A. Human Parameters

Human is more often a direct owner of computer resources. Human can also be a user of environment resources or services. In the worst case, he/she can be a selfish owner and a greedy user. Today's distributed environments aim to engage more and more non-dedicated resources [15]. Non-dedicated resources are either idle or underloaded. According to [13], resource management systems take care of the fair treatment of users. On the other hand, they cannot prevent a human (owner) from making his/her resources unavailable to the grid, but some solutions are proposed in [15]. Namely, this problem becomes solvable when the resource owner executes computer intensive applications, i.e. when he requires services from the established environment. The problem of resource concession is almost unsolvable for owners who do not use computer system services. The solution in this case can be based on other services interesting to computer owners (Internet connection, operating system and similar). Fortunately, small potential users of grid services are at the same time the owners of smaller computer resources.

As initiated in [7] and [16], owner/user or human parameters (H) are the following:

- Ownership of resources. It can be true static ownership, but also inherited ownership of resources.
- Authorization of institution upon resources. It can be partial or full.
- Availability for on demand intervention in environment extension.
- Knowledge and experience level in distributed computer systems usage and development.
- Required ratio of given/taken resources/services.
- Infrastructure level of resource owner as precondition of successful environment extension.
- Environment performance level required by owner/user.

B. Platform Parameters

The following parameters (P) describe execution platform:

- Operating systems installed on machine.
- CPU clock frequency.
- Number of CPU's in machine and number of cores in each processor.
- Machine memory size. It can be a static parameter (installed memory) and a dynamic parameter (currently available memory).
- Network speed can also be statically and dynamically observed – as nominal and currently available network throughput.
- Hard disk capacity is also observed as nominal and currently available capacity.
- Machine support for work in computational cluster and/or grid.

C. Application Parameters

Application parameters are:

- Expected execution time of application [3] or one part of matched application on established environment. It is a static parameter.
- Makespan or maximal execution time [4] as a dynamic parameter.
- Application preemption.
- Size of execution code of application.
- Requested priority level of application execution.
- Expected application setup time.

D. Mapping Parameters

Mapping parameters (M) include a description of resource management procedure. As described in [3], mapping includes matching tasks to machines as well as scheduling tasks on the matched machine. They are:

- Mapping duration.
- Mapping preemption possibility.
- Implemented matching/scheduling algorithm.
- Remapping [15] application or part of application to other machines by system overload or unexpected events.
- Number of enabled remappings.

In addition to application, platform and mapping parameters, parameters regarding possibilities and tendencies of the resource owner/user towards distributed computer technology are also implemented in the DnD Grid system. These tendencies represent a good basis for an indirect selection of resources via the profile of their owners, as well as their cooperation in distributed computer system establishment. In that way, resource owners/service users are responsible for establishment and usage of this virtual computer organization, as in [11] and [13]. Also, they are in competitive position.

III. DnD Grid System

A. System Architecture

As described in [16], a DnD Grid system architecture allows participation of the owner/user in grid building. The owner/user makes registration on the corresponding web page and fills in a web questionnaire. On the basis of the questionnaire, willingness and capability of the resource owner for resource concession, i.e. user intention to use grid services are estimated. Upload and installation of a monitoring program (for hardware monitoring) on the owner/user machine is enabled afterwards. The hardware monitoring program checks values of static performance parameters filled in by the owner/user. In addition, this program enables monitoring, measurement and storage of dynamic parameter values, as proposed in [16]. On the basis of static and dynamic parameter values which periodically refresh the database, web service on the server generates and refreshes the ranking list. According to this list, mapping of user execution applications is carried out. Owners/users and machines with a better ranking can accept more jobs for processing, but also a greater possibility of using services of established distributed environment, as in [14].

Web pages created by using Java Server Pages and Java Bean technologies are accessible via web browser. Data filled in by the owner/user, as well as values measured by hardware monitoring application are saved in the database. This is enabled by the database manager. After saving owner/user data in the database, by using SOAP (Simple Object Access Protocol) which uses the FTP (File Transfer Protocol) server, hardware monitoring application will be uploaded on the owner/user machine. This application is developed in Java. Finally, Axis (XML Based Web Service Framework) allows creation of a web service performed by Tomcat Server Container. Technologies used so far are partially presented in [16] and in other resources. Fig. 1 shows one web page of the system.

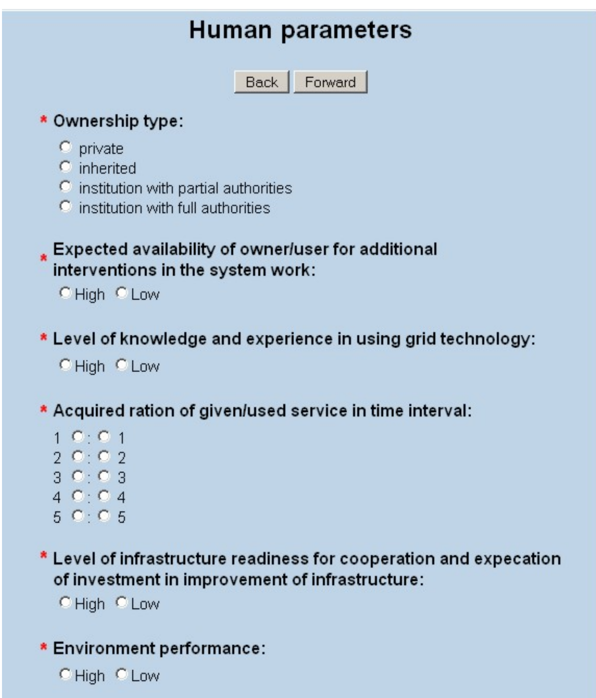

Fig. 1 DnD Grid system web page

B. Ranking List Generation

Ranking list of owners/users and machines that participate in application execution is generated according to values defined by the owner/user, as well as measured parameter values. In addition to numerical values, weights are associated to the parameters. Weights of a human profile (owner/user) are especially important. For example, the owner has the value of 0.75, owner/user 0.25 and user 0.00. Parameters from the human, platform, application and mapping level get weight depending on willingness and capability of grid extension using own resources. All level parameters usually obtain weights from the range 1 to 5 or two discrete values from this range. Fig. 2 shows steps of ranking list generation, application execution and environment improvements. These steps are implemented in the DnD Grid system.

collect *owner/user requests (from level* H *) to application* A
collect *initial requests of application* A *to platform* P*, mapping* M *and human* H
analyze *dynamic parameters of past executions from* A, H, P *and* M
select *owners/users and resources according to current parameters and parameters from past executions*
run *application or part of application* A *to established system*
collect *currently achieved parameters from* A, H, P *and* M
evaluate *application execution*
replace *weak owners/users and machines by better candidates*
repeat

Fig. 2 Ranking list generation procedure

IV. Experimental Results

A. Experimental Setup

The experiment was carried out in a simplified distributed testing environment. It included two institutions and personal computers (with corresponding processor and data resources) connected with 100Mb/s LAN and a mobile broadband with different speeds. We used 30 non-dedicated machines and 8 owners/users. One "big" owner has administrative properties on 10, and the other one on 7 machines. Other "small" owners possess one machine (13 machines together). Machines differ in processors (simple PIV and AMD, dual core and quad with different clock frequency architecture, cache, etc.), memory space, hard disk capacity and the operating system. This heterogeneous environment had a rather high level of heterogeneity.

Our methodology is evaluated on the application which is part of the image processing system in the access control system. It processes images which correspond to a quadratic matrix of dimensions 5000x5000. The nature of system behavior determines the comparison procedure as time critical. According to execution time (about 25 minutes) achieved on a single dedicated machine (dual core processor machine), we can talk about a computer intensive application. The application server implemented in the DnD Grid system divides an image into up to 40 independent and different size meta-tasks. Meta-tasks are mapped to 24 (out of 30) computers selected according to the generated ranking list and the mapping procedure. Mapping is carried out on the basis of combining static and dynamic parameters and phases described in [16]. Parameters described in Chapter II make the main selection criteria.

It is assumed that machine owners are also potential users of services of this simplified computational grid, and that they want to reach the best ratio of the given/taken services. For the purpose of a better interface with the application, the described web application and hardware monitoring program enable evaluation of machines according to both platform parameters and owners/users parameters. Prior to engagement, owners fill in a web form, which in a static sense contains model parameter values, i.e. it evaluates them statically. Dynamic evaluation is based on values of the same parameters obtained on the application server every 2 minutes from all mapping candidates.

First, mapping is carried out by the well known min-min algorithm described in [3], and then by the proposed advanced min-min algorithm (A-min-min). For each unassigned task a very successful min-min algorithm determines the time the task can be completed given the projected idle times of each machine and the estimated execution time of the task on each machine. Additionally, A-min-min is targeted to increase machine availability in non-dedicated environment. Statically, during matching it covers only the best subset of machines (by platform parameters and by owner/user characteristics). This subset is actualized during the dynamic phase of mapping. For both algorithms, each application part is matched to the machine, which has a less value of the sum of the expected time to compute this job and which is possessed by the best owner.

B. Results and Analysis

Three experiments are carried out for min-min and for A-min-min algorithm. Parameter values are shown in Table 1 and in Figures 3 and 4. Experiment 1 gives parameter values for the case of mapping based on the min-min algorithm, as well as the A-min-min algorithm. Application tasks are mapped on unintentionally selected 24 machines. Experiment 2 shows parameter values obtained by using both algorithms and resource reservation according to platform and application parameters. In other words, owner/user or human parameters are excluded from the mapping procedure and ranking list generation. Experiment 3 implies min-min and A-min-min algorithm based mapping which includes platform, application, as well as human parameters. The aim of Experiment 3 is to point out importance of owner/user parameters in environment establishment.

Results are evaluated by the following parameters described in [16]:

- Maximal application execution time ($C_{\max}$). It indicates improvement of the system throughput. It is defined as a difference between the application end time and the start time.
- Percentage of mean deviation of workload ($\sigma_{avg}W$) corresponds to the mean value expressed in percentage of all deviations of each single machine from their ideal workload. Of course, the smaller percentage of mean deviation of workload, the better load balancing.
- Ratio of communication costs by achieved processing time (com/c) indicates network workload depending on environment extension.

In comparison with min-min mapping and application execution on unintentionally selected machines (Experiment 1), resource reservation based on min-min mapping and platform parameters (Experiment 2) achieves a shorter execution time ($C_{\max}$) for about 26% and enables better load balancing between machines. It is indicated by less mean deviation of workload $\sigma_{avg}W$ for about 19%. The ratio of communication costs to the achieved processing time (com/c) is increased (for about 28%) due to the increasing number of parameters included in the mapping procedure. By including human parameters in Experiment 3, $C_{\max}$ is additionally reduced for about 15%, $\sigma_{avg}W$ is additionally improved for about 17% and com/c holds within a tolerable range. As expected, this approach can reduce the risk of owner-dependent resource availability.

TABLE I. EXPERIMENTAL RESULTS

Evaluation parameter	Exp. 1	Exp. 2	Exp. 3
min-min			
$C_{\max}$ [s]	137	101	86
$\sigma_{avg}W$ [%]	17.14	13.97	11.61
com/c	1	1.29	1.33
A-min-min			
$C_{\max}$ [s]	131	89	72
$\sigma_{avg}W$ [%]	16.69	12.85	10.28
com/c	1	1.31	1.35

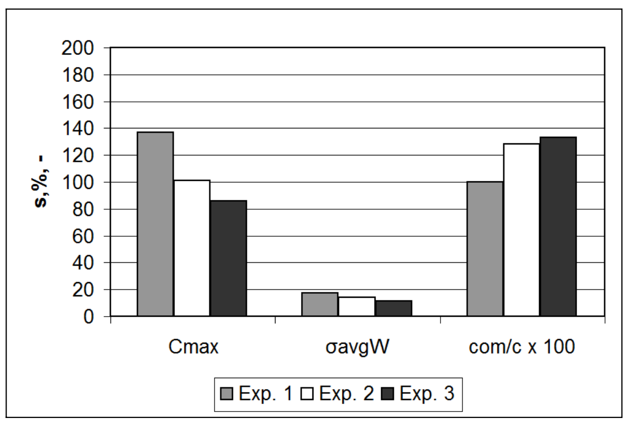

Fig. 3 Experimental evaluation for the min-min algorithm

In comparison with min-min, A-min-min based mapping improved all initial parameter values (Experiment 1), shortened execution time ($C_{\max}$) for about 6% and 4% in Experiment 2 and Experiment 3, respectively. It also improved load balancing (decreased workload deviation $\sigma_{avg}W$) for about 4% in Experiment 2 and 3% in Experiment 3. Parameter com/c holds in a similar tolerable range as for min-min mapping for all experiments.

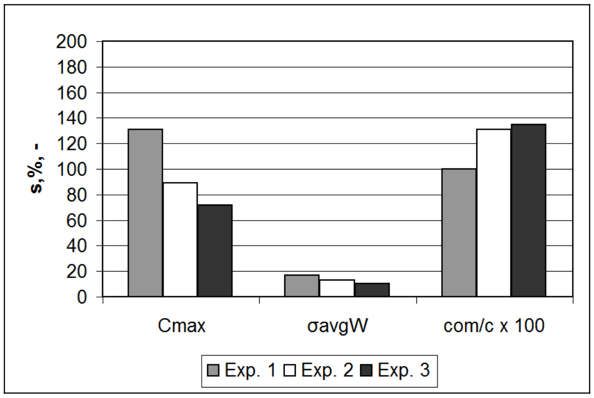

Fig. 4 Experimental evaluation for the A-min-min algorithm

V. Conclusion

Today's ubiquitous and fast wired, wireless and mobile networks enable connection of powerful desktop, mobile and multiprocessor computers, as well as their owners/users in virtual organizations. In that way, powerful, but heterogeneous distributed computer environments are created. These environments are usually established on dedicated computer resources. Simultaneously, numerous non-dedicated machines are statically overdimensioned and dynamically underloaded. Engagement of these non-dedicated resources can significantly improve computer environment efficiency, availability and autonomy. In the described approach, a human as resource owner and/or environment service user represents the most important factor. The developed DnD Grid system enables registration and self-description of the owner/user, monitoring and measurement of static and dynamic parameters from the platform, application and mapping level, generation of a ranking list of owners/users and their resources, mapping and execution of user applications on the established environment, as well as its dynamic improvement.

Experimental results are obtained in a simplified industrial environment for three execution conditions and two mapping algorithms – min-min and advanced A-min-min. A-min-min includes internal procedures for algorithm adjustment to dynamic behavior of environment. Unintentional selection and inclusion of machines in the system decrease the maximal processing time and improve load balancing insignificantly. The achieved performance is improved by monitoring the environment on the platform, application and mapping level. Consideration of owner/user parameters by resource management enables significant shortening of execution time, better load balancing and holds communication costs within a tolerable range. An advanced version of the mapping algorithm additionally improves application execution. The proposed approach seems to be adequate in time-critical applications and it will be tested, evaluated and improved for the usage in real grid or cloud computer environment.

Acknowledgment

This work was supported by research project grant No. 165-0362980-2002 from the Ministry of Science, Education and Sports of the Republic of Croatia.

References

[1] N. Andrade et al., "Automatic grid assembly by promoting collaboration in peer-to-peer grids," Journ. Parallel and Distributed Computing, vol. 67, no. 8, pp. 957-966, Aug. 2007.

[2] N.A. Assimakopoulos, A.D. Theodosi, "A Systematic approach for modeling virtual enterprise's management features", Tamkang Journ. Science and Engineering, vol.6, no.2, pp. 87-101, 2003.

[3] T. D. Braun et al., "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," Journ. Parallel Distributed Computing, vol. 61, no. 6, pp. 810-837, June 2001.

[4] P. Brucker et al., "Complexity results for parallel machine problems with a single server," Journ. Scheduling, vol. 5, no. 6, pp. 429-457, Nov. 2002.

[5] R. Buyya, C.S. Yeo, S. Venugopal, "Market-oriented cloud computing: vision, hype, and reality for delivering IT services as computing utilities," in Proc. of 10th Int. Conf. High Performance Computing and Communications, Dalian, China, September 25-27, 2008, IEEE Press, New York, NY, pp. 5-13.

[6] L.M. Camarinha-Matos, "New collaborative organizations and their research needs," Proc. of IFIP 5th Working Conf. on Virtual Enterprises (PRO-VE '03), Lugano, Switzerland, Oct. 29-31, 2003, Kluwer, Norwell, MA, 2003, pp. 3-14.

[7] P. Carayon, "Human factors of complex sociotechnical systems," Applied Ergonomics, no. 37, pp. 525-535, July 2006.

[8] D. Caromel, A. di Costanzo and C. Mathieu, "Peer-to-peer for computational grids: mixing clusters and desktop machines," Parallel Computing, vol. 33, no. 4-5, May 2007.

[9] J.S. Chase, D.E. Irwin, L.E. Grit, J.D. Moore, S.E. Sprenkle, "Dynamic virtual clusters in a grid site manager," Proc. of 12th IEEE Int. Symp. on High Performance Distributed Computing, Seattle, WA, June 22-24, 2003, IEEE Press, New York, NY, 2003, pp. 90-100.

[10] A. Corsaro, A. Di Stefano, G. Pappalardo, C. Santoro, E. Tramontana, "Emerging technologies for next-generation GRID," Proc. of 13th IEEE Int. Workshops on Enabling Technologies (WETICE '04), Modena, Italy, June 14-16, 2004, IEEE Press, New York, NY, 2004, pp. 243-246.

[11] N. Doulamis et al., "Adjusted fair scheduling and non-linear workload prediction for QoS guarantees in grid computing," Computer Communications, vol. 30, no. 3, pp. 499-515, Feb. 2007.

[12] I. Foster, C. Kesselman, S. Tuecke, "The anatomy of the grid: enabling scalable virtual organization," Int. Journal of High Performance Computer Applications, vol. 15, no.3, 2001, pp. 200-222.

[13] D. Kondo et al., "Characterizing resource availability in enterprise desktop grids", Future Generation Computer Systems, vol. 23, no. 7, pp. 888-903, Aug. 2007.

[14] M. Maheswaran and S. Ali, "A taxonomy of network computing systems," IEEE Computer, pp. 115-117, October 2004.

[15] G. Martinovic, Owner/user role in computational grid extension by non-dedicated resources," Proc. of ACM Int. Symp. on Supporting Group Work, (GROUP '03), Sanibel Island, FL, Nov. 9-12, 2003, ACM Press, New York, NY, 2004, 221-228.

[16] G. Martinovic, H. Gasparovic, D. Antonic, "DnD Grid - distributed computer environment based on non-dedicated resources," in Proc. of IEEE SMC International Conference on Distributed Human-Machine Systems, Prague, Czech Republic, 2008, pp. 281-286.

[17] Z. Nemeth, S. Sunderam, "Characterizing Grids: Attributes, Definitions, and Formalisms," Kluwer Jour. Grid Comp., vol.1, no.1, 2003, pp. 9-23.

[18] Y. Xiao, "A new wireless web access mode based on cloud computing," in Proc. of 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Applications, December 19-20, Wuhan, China, IEEE Press, New York, NY, 2008, pp. 645-649.

[19] L.T. Yang, M. Gao, "High-Performance Computing, Paradigm and Infrastructure," Hoboken, NJ, USA: Wiley-Interscience, 2005.