

## Agent Based Grid Resource Discovery with Negotiated Alternate Solution and Non-Functional Requirement Preferences

<sup>1</sup>P. Muthuchelvi, <sup>2</sup>G.S. Anandha Mala and <sup>3</sup>V. Ramachandran  
<sup>1</sup>Sathyabama University, Chennai, India  
<sup>2</sup>St. Joseph's College of Engineering, Chennai, India  
<sup>3</sup>Anna University, Trichy, India

---

**Abstract: Problem statement:** As grid resources are geographically distributed, efficient resource discovery and management has become one of the important requirements. Besides, Grid users are independent identities and negotiation is necessary for reconciling their diverse characteristics. Therefore special mechanism was required to negotiate and discover the required resource or similar resource as an alternative when discovery fails. Moreover the quality of the service being provided in the grid environment depends on both functional as well as the Non-Functional Requirements [NFR]. But conflicts between NFRs are not yet resolved effectively. Discover the requested resources to the requester, Provide compromised alternate resources by negotiation when resource discovery fails to increase the success rate of the agent, Provide knowledge for efficient management of resources and quality of service is to be improved by considering NFR. **Approach:** A system Agent Based Grid Resource Discovery with Negotiated Alternate Solution and Non-Functional Requirement Preferences (AGRD\_NFRP) was proposed to provide an expeditious resource and most relevant alternate resource when discovery fails. Four types of intelligent and mobile agents were proposed for judicious management of resources to the advantage of resource providers and requesters in ensuring speedy execution of processes. Resource discovery, negotiation and alternate solution were handled by these agents. In order to improve the quality of the service the non-functional requirements of the grid user request with their preferences were identified and conflicts among them were analyzed using fuzzy rules. **Results:** The results showed that the AGRD\_NFRP system proposed is producing consistently higher success rate by providing alternate solution and getting knowledge from the cognitive agent. Quality of the service was enriched by prioritizing the preferences of grid user. **Conclusion:** On numerous occasions, grid users face non availability of high-end resources for completing the task on hand. In this context, the approach outlined in this research is most appropriate, convenient and efficient. The AGRD\_NFRP system proposed herein played a crucial role in bridging the seemingly wide gape between resource requester and resource owners.

**Key words:** Grid resource discovery, agents, negotiation, alternate resource, non-functional requirements

---

### INTRODUCTION

A large scale distributed system that will provide high-end computational and storage capabilities to a set of differentiated users is a grid infrastructure. Grid computing is a hot research direction and drawing a lot of attentions from both academia and industry. It has emerged to facilitate better utilization of under utilized heterogeneous and geographically distributed resources. The main motivating factor in grid computing is resource sharing. It is provided by resource management system, which is the central component of grid computing. It mainly focuses on management and

scheduling of computations over dynamic resources scattered geographically across the internet<sup>[1]</sup>. Resource management acts not only as an interface between grid resource and grid application but also to provide reliable service to the user<sup>[1]</sup>.

As the participating parties of grid market are independent bodies, negotiation activities are required<sup>[2]</sup>. An application or client must engage in a multi-phase negotiation process with resource managers, as it discovers, reserves, acquires, configures, monitors and potentially renegotiates resource access<sup>[3]</sup>. In this context, negotiation has emerged as one of the important activities in the grid

market. However, the development of the grid speeding-up in recent times continues to be unsatisfactory. One possible reason for this is an overly complicated support for resource brokering and management provided by current grid software infrastructure<sup>[4]</sup> and economic model was not considered. But an economic model was considered in<sup>[9]</sup>. There are many issues and challenges in the Grid environment. Amongst the challenges for Grid computing, to date, there is little work that addresses the issues of requirement engineering. To the best of the author's knowledge, at present, there are only a few (preliminary) efforts on considering NFRs to provide quality software. However the grid user may not aware of the conflicts among NFRs. One of the motivation of proposed system is to identify non-functional requirements of the grid user request, resolve the conflicts among them

Agent technologies have been used for the development of distributed software system<sup>[5]</sup>. An Agent can be considered both as a resource provider and a service requester<sup>[6]</sup>. Multi-agent system is one of the promising software technologies to achieve the goal<sup>[12]</sup> due to the characteristics of the agent like autonomy, proactive, mobility and adaptability. A mobile agent toolkit was designed for resource discovery<sup>[13]</sup>. The agent system which incorporates all such agents, bridges the gap between grid users and resources in order to schedule the applications that require grid resource, in an efficient manner. Agent achieved higher success rate by slightly relaxing the bargaining terms, in intense pressure situation. However, relaxation was decided on fuzzy decision controller<sup>[7,8,14]</sup>. In the existing Agent Based Resource Management system, when the root of the hierarchy is

reached due to navigation and the available resource is still not found, the discovery terminates unsuccessfully<sup>[10]</sup>. In such a scenario, the proposed system provides alternate resource.

Considering the inadequacies of the efforts previously made, the motivation behind the research is to sets out the system of AGRD\_NFRP for devising a speedy and very efficient method of resource discovery with the help of agents. The main objective of this study is to provide most relevant resource when necessary and to increase the success rate of agent. The mechanism of providing alternate resource when discovery fails by negotiation is also discussed in this study. Agents are classified into groups according to the service they provide, for quickening the discovery process and the system is supported with cognitive agent to provide knowledge when required. As the user is the ultimate judge for deciding the appropriateness of the resource, the system gets refined input from the user at the outset itself. The quality of the service being provided in the grid environment depends on both functional as well as the Non-Functional Requirements [NFR]. But conflicts between NFRs are not yet resolved effectively. This study also presents an approach to identify the non-functional requirements of the grid user request with their preferences and analyze the conflicts among them.

## MATERIALS AND METHODS

**Architecture of the proposed system: AGRD\_NFRP** is proposed to provide an alternate solution even when resource discovery fails for time bound and cost bound processes. An overview of the AGRD\_NFRP architecture is shown in Fig. 1.

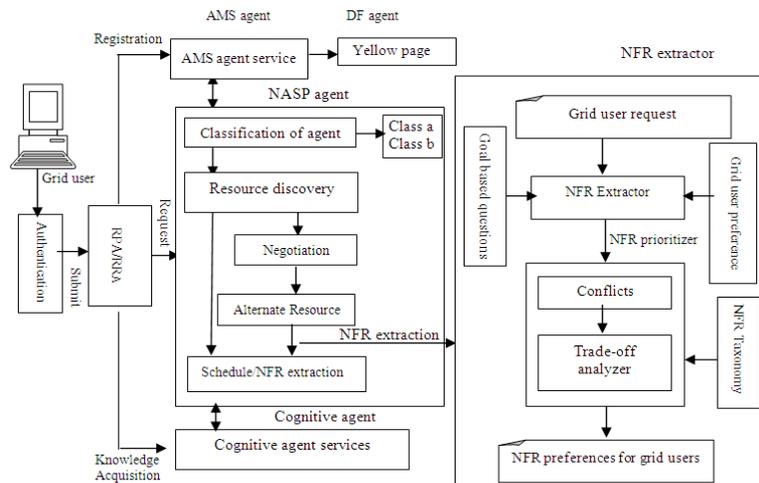


Fig. 1: Architecture of AGRD\_NFRP

The main components of this system comprise mainly (i) Grid users (ii) Agent Management System-AMS (iii) Directory facilitator-DF (iv) Negotiation and Alternate Solution Provider agent (NASP) (v) Resource Requester Agents-RRA (vi) Resource Provider Agent-RPA and (v) Cognitive agent. (vi) NFR extractor (vii) NFR Prioritizer and taxonomy (viii) Goal-based questionnaires.

**Grid users:** There are two categories of users in the grid computing environment viz (1) Resource requester and (2) Resource owner. The resource requester may seek the resource for application Second type of user is resource owner who registers his resources with the agents, with specifications and constraints.

**Agent Management System (AMS):** AMS is the agent exercising supervisory control over access of agents. Each agent must register with the AMS in order to get a valid agent identification number. It maintains a directory of agent identifiers and agent state like active, transit, wait.

**Directory Facilitator (DF):** Directory facilitator is a registry playing a vital role in providing a list of agents in the system and the services of the respective agents with identification number. Thus an agent can find the other agents in the same platform to achieve goals.

**Resource Requester Agents (RRA):** Agents, main components of the proposed AGRD\_NFRP system are independent and autonomous processes that have identities, possibly persistent, requiring communication with other agents for completing the tasks. RRA reads the resource request with specification from user who desiring to utilize the services from the Grid. To discover the resource, RRA accepts the specific request with grid user id using format shown in the Eq. 1-3:

$$\text{Guser-id } \langle \text{name of the application, resource specification} \rangle \quad (1)$$

$$\text{Guid } \langle \text{ua, utype}_i \mid i = 1 \text{ to } n \rangle \quad (2)$$

$$\text{Guid } \langle \text{Information processing service,} \\ 5000 \text{ GHz, 70 GB, 900\$, 1 h} \rangle \quad (3)$$

Where:

ua = The name of the application  
 utype<sub>i</sub> = The i<sup>th</sup> specification of the resource requested by the user

As soon as agent receives the request from the user, it contacts NASP agent for requested resource or

compromised alternate resource. It sends a request to the NASP agent as an ACL message 'CALL FOR PROPOSAL-CFP' along with: (1) Name of the resource, (2) Hours required and buying price offered with other constraints. Given its pivotal position, NASP will route the request to the corresponding group of agents. For example, request is routed to the 'super-agent' if the request is for supercomputer and 'cluster-agent' if request is for cluster. Thus the risk of referring the request to irrelevant agent can be avoided. Then it receives the transaction id from NASP agent for the requested trade and also confirms the acceptance of the proposal through an ACL message 'ACCEPT PROPOSAL'. If the resource has been bought from the relevant RPA, the NASP advises the resource requester agent through an ACL message 'INFORM' about the resource allocated.

**Resource Provider Agent (RPA):** RPA agent works on behalf of provider user who offers the resource to the grid community. When RPA approaches NASP as a ACL message 'CALL FOR PROPOSAL' along with (i) Name of the resource, (ii) Specification of the resource, (iii) Estimated sale price and other constraints for registering the resource. NASP includes it in the relevant group after due verification of service that can be provided. If the RPA does not fit into any of the existing group, NASP creates a group for the new service and this RPA will become the first agent in that group. Thus the number of groups can be increased according to the service provided. After adding the agent to the relevant group, NASP sends "REPLY ACL" with unique resource-id to the RPA. If the resource has been sold to the relevant requester agent, NASP advises the RPA through ACL message "INFORM" with the agreed price of resource sold. The RPA then waits for the 'CONFIRM PROPOSAL' from NASP for the resource to be allocated. In the grid market scenario, RPA can be expected to have mix of resource in heavy demand and resource with little or no demand. It will therefore have to continuously monitor the market demand with reference to its resource mix to propose the acceptable prices and increasing the success rate.

**Cognitive agent:** The developments of the grid in recent times have significantly speeded-up its performance and yet the position has remained inadequate. One possible reason for this could be that RRA and RPA have not had the benefit of assistance of Cognitive agent before the process was set in motion. In the proposed system, Cognitive agent helps in this regard. As resource discovery is important in dynamic grid environment, Cognitive agent is equipped to play

the crucial role. Acting as a back-end assistant for providing information when required, it acquires its knowledge from the processes happening in the system and updates itself while fulfilling its role. In the proposed system, Cognitive agent is thus customized agent and is responsible for the following services

**High/low demand resource:** In the dynamic grid environment, user may not aware of heavily/ lightly demanded resources and Cognitive agent helps in this regard. This information will come in handy, when RRA requests for a resource which is identified as heavily in demand. To acquire such a resource in intense pressure, RRA can offer a price higher than scheduled price. If the resource is identified as in less demand, the cost of the resource can be reduced by the RPA. For the purpose of identifying highly/less demanded resource, the NASP agent maintains Utilization Factor (UF) of the system. Thus NASP plays crucial role of striking a balance between demand and supply of resource guided by their respective UF.

**Instantly available resource:** As the availability of the resources is dynamic in the grid platform, the prospective user may not be aware of the availability position of resource. Cognitive agent provides this service to reduce delay overhead in waiting for instantly unavailable resource. On receiving the request, Cognitive agent provides the list of freely available resources, at any given point of time and the user may select any suitable resource on offer. If user needs to know more details of the resource before placing request, Cognitive agent provides the resource details. Cognitive agent updates its list of available and unavailable resource with the help of NASP agent. This function of Cognitive agent eliminates request for unavailable resource and minimize the failures in the discovery process.

**Transaction History:** Past history of the resource helps to improve the performance of the system. When the requested resource is not available, NASP tries to find an alternate resource. However, before going into the standard procedure to find the alternate resource, NASP checks with cognitive agent for the history of transactions where alternate resources had been provided. If past history is satisfactory for an alternate resource, NASP checks its current availability and allocates it. Thus, delay in finding an alternate resource is overcome and the process is quickened. A history of all the transactions is maintained in the system for reference. A transaction detail includes details of the

provider and requester, cost, time factor and success rate. This data can be used in both current and subsequent dealings.

**Suitable resource identification:** However, in some cases user may not have precise knowledge of the resource most suited for executing the specified application. There is no set approach to help the user in finding suitable resource for special purpose application. Cognitive agent helps the user in this respect. Even when the user knows the suitable resource, it is better to consult Cognitive agent before making request because there is always the possibility of addition or deletion of the resource in the dynamic grid environment. Sometimes, better suited resource might have been added into the grid recently without the user's knowledge. Similarly well-known resource might have been deleted from the grid. So it is always better for the user to contact the Cognitive agent to know about the grid market dynamics.

**Performance Evaluation:** The evaluation by NASP is based on the experience of RRA after utilization of the resource. Success rate of the resource is found with the help of utilization report generated by the RRA to NASP. This report shows the satisfaction level of the user. Whenever NASP gets such report, success rate is updated. The resource which has highest success rate is preferable to others. This approach helps to find the better suited resource when more number of matches for the request is available. Same procedure is followed also for finding out the best resource provider agent.

**Negotiating and Alternate Solution Provider (NASP) agent:** Negotiation is the process by which agents try to reach an agreement for the requested resource. As the grid participants are independent identities with diverse policies, objectives, it is necessary to solve the differences between providers and requesters. Negotiation is used in two scenarios. 1. To find out most suitable resource when more matches occurs and 2. To find alternate resource, when discovery fails. This agent plays a critical role and acts as a link between resource requester and resource provider agent. It performs many important functions like collecting the resource details from providers, match making, negotiating and providing alternate resources besides monitoring the quality of service in the system. NASP and Cognitive agent occupy a unique position in the system envisaged, considering the level of knowledge and experience generally vested in them. To conduct negotiation, NASP must follow a well-defined algorithm specifying the sequence of steps that can be

taken at different stages of the process. NASP agent seeks assistance from Cognitive agent to short list the most appropriate RPA before negotiation begins. During negotiation, evaluation of RRA requests and RPA proposals facilitate expeditious decision on the alternate resource.

On receiving the request from RRA agent, NASP searches for the RPA group that can provide the requested resource. When exact match is found, resource discovery succeeds and NASP sends message to the respective provider agent allocating the resource and updating the tables. It then waits for the utilization report from the requester. In case discovery fails, it tries to find alternate resource based on the type of processes (time/cost bound) after negotiation. To provide alternate resource, NASP relaxes attributes other than cost for cost bound process and time for time bound process. However for both the processes, relaxation is allowed only within the level of relaxation factor.

**Negotiation for time bound process:** Time bound processes should be completed within the specified time. When RRA approaches NASP for assistance, the latter searches for the resource in the relevant group for the exact match, if there is no exact match then it searches for the resources with the requested speed in the available list by relaxing other attributes, assuming that speed is the highest priority. On receiving multiple matches, it negotiates and finds the least cost resource for the benefit of requester. Then it sends “CONFIRM PROPOSAL” to the provider and allocates the resource. After allocation, it waits for the utilization report from the requester. If there is no resource satisfying the conditions imposed by RRA, as a last resort NASP tries to negotiate and find the similar resource for the required resource by compromising price using weight factor shown in Eq. 3-6. Normally compromised price lies between price quoted by provider and price offered by requester:

$$W_p = (Pr*Hr) / (Pr*Hr + Pp*Hp) \quad (4)$$

$$W_r = (Pp*Hp) / (Pr*Hr + Pp*Hp) \quad (5)$$

$$\text{Compromised price} = PpW_p + PrW_r \quad (6)$$

Where:

Pr = Price offered by requester

Pp = Price quoted by provider

Hr = Hours required by the requester

Hp = Hours resource can be provided by the provider

Wr = Weight of requester

Wp = Weight of provider

This is the common pricing strategy that is used by both the RPA and the RRA. When no such agent is readily available in the system, the RRA and the NASP have to wait for a suitable RPA to arrive.

**Negotiation for cost bound process:** It is the process that should be completed within the specified cost. When RRA requests NASP for the required resource, NASP searches for the resource in the corresponding group with the given cost. If multiple matches found, it negotiates and finds the least cost resource and allocates it. When the required resource is found to be engaged with some other process, NASP finds an alternate solution making a precise assessment of the de-allocation time. De-allocation time can be ascertained as:

$$DAT = EETC + EETW \quad (7)$$

Where:

DAT = De-allocation time of the requested resource

EETC = Expected execution time of current process

EETW = Expected execution time of processes waiting in queue)

NASP compares the de-allocation time with maximum waiting time and when maximum indicated waiting time is greater than de-allocation time NASP sends message to RRA informing waiting time. During negotiation, for time bound process, NASP will seeks resource with the given or greater speed, as higher speed resource may be available in some other resource name at a higher cost. If discovery again fails, the process has to continue for finding a lower speed resource within the permissible relaxation level. But, for the benefit of requester, such a resource identified by NASP should be within relaxation factor Rf:

$$\text{Allocation speed} - \text{requested speed} < Rf$$

Now in the proposed system, Rf is assumed to be within the tolerance level which may be ascertained from the requester at the input stage itself, as the requester is the ultimate judge for deciding the permissible level. Similarly in the case of cost bound process, the tolerance level can ascertain from the requester.

NASP agent executes each transaction between the RPA and the discover agent observing specific procedure. Such a procedure facilitates maximum advantage to both the RPA and discover agent

**NFR extractor:** NFR preferences for the grid user requirements are extracted with the help of goal based questionnaires and grid user preferences.

Table 1: Goal based questionnaires

Event	Preference	NFR
Type grid user-id	Invalid user-id	Security
Type grid user-id	Provide wizard or portal /guide to enter the user-id without problem	Usability
Negotiation	Terminate the negotiation when two parties not come to an agreement	performance
Press login/logout button	Provide Wizard/Guide to press logout button	Usability
Press login/logout button	Enable user to use button effectively at reliable places	Reliability
Choose resource for application	Information for choosing resource is accurate and available	Correctness
Triggers alarm	Alarm resource repaired, changed and maintained	Maintainability
Triggers alarm	Constraints for trigger only when authorized users access the resource.	Security

Table 2: NFR taxonomy

Correctness#Reliability+#Efficiency+#Accuracy+#Conciseness+#Tolerance+#Precision+ Performance#Response+#Throughput+#Timeliness+#Availability-#Reliability- Reliability#Efficiency+#Accuracy+#Latency-#Throughput-#Availability- Security#Identification+#Authorization+#Immunity+#Nonrepudiation+#Privacy+#Performance- Usability#Simplicity+#Accessibility+#Installability+#Operability+#Maintainability- Maintainability#Flexibility+#Simplicity-#Operability-#Usability-#Portability+ Availability#Reliability-#Integrity-#Precision-#Throughput+#Tolerance- Authorization#Security+#Performance-#Authentication+#Reliability+#Privacy+ Efficiency#Simplicity+#Maintainability+#Latency+#Performance-#Maintainability- Identification# Security+# Performance- Authentication# Security+# Performance-
---

Table 3: Sample fuzzy rules

If (Efficiency is low) and (Accuracy is low) then (Reliability is low)
If (Efficiency is high) and (Accuracy is high) then (Reliability is high)
If (Latency is low) and (Throughput is high) and (Availability is high) then (Reliability is high)
If (Latency is high) and (Throughput is low) and (Availability is low) then (Reliability is low)

The goal based questionnaire includes all possible questions for the activities of both resource requester user and resource provider user. Users have to give their preferences by appropriately answering for the questions provided by the user friendly portal designed for this purpose. From this portal information ‘NFR extractor’ extracts the non-functional requirements preferences for the user and redirects them to the ‘NFR prioritizer’. Sample goal based questionnaires are shown in the Table 1.

**NFR prioritizer:** ‘NFR Prioritizer’ consists of two components namely 1.Conflicts Identifier 2.Trade-off-Analyzer. First component ‘Conflicts identifier’ identifies the conflicts among the extracted NFRs with the help of NFR taxonomy. In ‘NFR Taxonomy’ all the NFRs are associated with other conflicting and dependable NFRs are stored. The entries in NFR taxonomy looks like:

Performance#Response+#Throughput+#Timeliness+#Availability-#Reliability-

It states that ‘Performance’ is directly proportional to ‘Response’, ‘Throughput’, ‘Timeliness’ and ‘Availability’ but indirectly proportional with ‘Reliability’.

The sample NFR taxonomy and fuzzy rules are shown in the Table 2 and 3. After identifying the

conflicting NFRs, the NFRs are prioritized based on the trade-off analysis. Trade-off analysis explores the cost of relaxing one NFR in order to achieve an increase in another NFR. This is implemented using fuzzy rule sets. These rules are formulated for each NFR according to the conflicting and dependable NFR. After the process of fuzzyfication and de-fuzzyfication system produces the result.

## RESULTS

Java Agent Development framework-JADE platform has been used to develop mobile agents. Grid environment is created using Globus Toolkit (GT4) in the Scientific Linux platform. However, In order to identify JADE agent classes by Globus, JADE main library files are placed in the Globus library. Whenever Globus container is started, it will identify the JADE container as shown in the Fig. 2. The agent platform has been split on several hosts provided there is no firewall among them. Agents are created in Grid environments among five systems. Agents are implemented as a java thread and Agent Communication Languages (ACL) messages are used for effective and lightweight communication between agents. User friendly portals are created in ASP. The goal based questionnaires are stored in Ms-Access database. The “AGRD-NFRP” is producing success rate of 85% with alternate solution and success rate of 76% with out alternate solution.

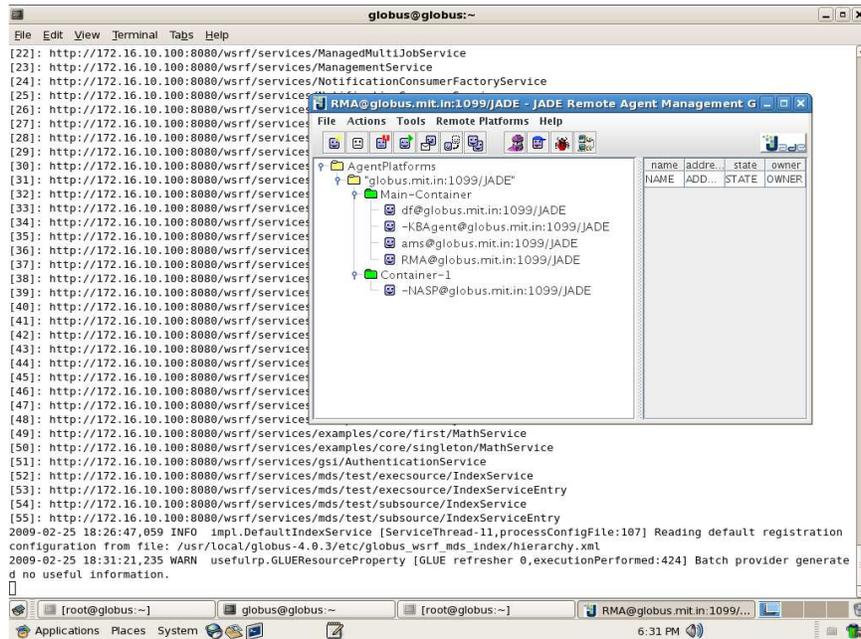


Fig. 2: Integration of agent and Grid environments

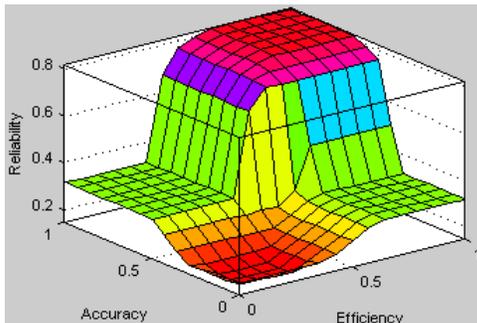


Fig. 3: Surface output for reliability

NFR are identified for the grid user request and conflicts between NFR are identified and prioritized using trade off analysis with the help of fuzzy rule sets in MATLAB fuzzy engine. The prioritization was done by adjusting the weight value of NFR. The surface output for reliability are shown in Fig. 3.

### DISCUSSION

It was observed that the success rate of the system with alternate solution was consistently higher than the success rate of the system without alternate solution. Initially, as the number of processes increased in the grid (grid load), the success rate also increases. But when the number of processes kept increasing i.e.,

when the grid was heavily loaded, there were fewer available resources in the grid and the success rate started declining for both with and without alternate solution.

But success rate was always higher for with alternate solution. It was observed that numbers of processes completed with alternate solution were more than number of processes completed without alternate solution at various stages.

### CONCLUSION

This study shows the integration of agent paradigm and grid environment. The AGRD\_NFRP system provides the environment for the execution of agent service as one of the grid service. The AGRD\_NFRP system is proposed to enhance the known methods of grid resource discovery with an innovative negotiation mechanism. It plays a vital role in bridging the seemingly wide gap between grid environment, Multi-agent system and requirement engineering. A novel approach of deploying NASP, Cognitive agent and NFR extractor and Prioritizer used is suggested for gratifying the critical functions of linking different domain. As the quality of the service being provided by the 'AGRD\_NFRP' system also depends on the non-functional requirements such as feasibility, reliability, performance. This study also focused on NFR preferences. Further, the service provided by NASP

agent can be considered for brokerage. Trade-off analysis of 'NFR\_Extractor' can be automated by introducing knowledgebase.

## REFERENCES

1. 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 and Exhibition on High Performance Computing in Asia-Pacific Region, May 14-17, IEEE Computer Science Press, USA., pp: 1-7. DOI: 10.1109/HPC.2000.846506
2. Chao, K.M., R. Anane, J.H. Chen and R. Gatward, 2002. Negotiating agents in a market-oriented grid. Proceeding of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, May 21-24, IEEE Xplore Press, USA., pp: 436-436. DOI: 10.1109/CCGRID.2002.1017183
3. Czajkowski, K., I. Foster and C. Kesselman *et al.*, 2002. SNAP: A protocol for negotiation of service level agreements and coordinated resource management in distributed system. Proceeding of the 8th Workshop on Job Scheduling Strategies for Parallel Processing, Apr. 30-30, Edinburgh Scotland, pp: 153-183.  
<http://www.mcs.anl.gov/~schopf/ggf-sched/GGF5/sched-GRAAP.3.pdf>
4. Dominiak, M., W. Kuranowski, M. Gawinecki, M. Ganzha and M. Paprzycki, 2006. Utilizing agent teams in Grid resource management-preliminary considerations. Proceedings of the IEEE JV Atanasoff Conference, (IJAC'06), IEEE Computer Science Press, Los Alamitos, CA., USA., pp: 46-51.  
[http://www.ibspan.waw.pl/~paprzyck/mp/cvr/research/agent\\_papers/JVA\\_06.pdf](http://www.ibspan.waw.pl/~paprzyck/mp/cvr/research/agent_papers/JVA_06.pdf)
5. Cao, J., S.A. Jarvis, S. Saini, D.J. Kerbyson and G.R. Nudd, 2002. ARMS: An agent-based resource management system for grid computing. Sci. Program., 10: 135-148.  
<http://www.dcs.warwick.ac.uk/~saj/papers/arms.pdf>
6. Cao, J., D.P. Spooner, J.D. Turner, S.A. Jarvis, D.J. Kerbyson, S. Saini and G.R. Nudd, 2002. Agent based resource management system for grid computing. Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, Mar. 21-24, IEEE Xplore Press, USA., pp: 350-350.  
DOI: 10.1109/CCGRID.2002.1017159
7. Sim, K.M., 2006. A survey of bargaining models for grid resource allocation. ACM SIGecom Exchanges, 5: 22-32.  
<http://doi.acm.org/10.1145/1124566.1124570>
8. Sim, K.M., 2007. Relaxed-criteria G-negotiation for grid resource co-allocation. ACM SIGecom Exchanges., 6: 37-46.  
<http://doi.acm.org/10.1145/1228621.1228625>
9. Manvi, S.S., M.N. Birje and B. Prasad, 2005. An agent-based resource allocation model for computational grids. Multi Agent Grid Syst., 1: 17-27.  
<http://portal.acm.org/citation.cfm?id=1233724>
10. Muthuchelvi, P. and V. Ramachandran, 2007. ABRMAS: Agent based resource management with alternate solution. Proceeding of the 6th International Conference on Grid and Cooperative Computing, Aug. 16-18, IEEE Computer Society Press, USA., pp: 147-153.  
DOI: 10.1109/GCC.2007.36
11. Muthuchelvi, P., G.S. Anandha Mala and V. Ramachandran, 2008. KBRMAS-knowledge based grid resource management for compromised alternate solution. Int. Trans. Syst. Sci. Appli., 3: 338-345.  
<http://siwn.org.uk/press/sai/itssa0003.htm>
12. Ouelhadj, D., J. Garibaldi, J.R. MacLaren, R. Sakellariou and K. Krishnakumar, 2005. A multi-agent infrastructure and a service level agreement negotiation protocol for robust scheduling in grid computing. Lecture Notes Comput. Sci., 3470: 651-660. DOI: 10.1007/11508380\_66
13. Rana, O.F. and B. Di Martino, 2004. Grid performance and resource management using mobile agents. Perform. Anal. Grid Comput., 251-263.  
<http://portal.acm.org/citation.cfm?id=976110>
14. Sim, K.M., 2004. Negotiation agents that make prudent compromises and are slightly flexible in reaching consensus. Comput. Intell., 20: 643-662.  
DOI: 10.1111/j.0824-7935.2004.00258