Enhanced Power Utilization for Grid Resource Providers

Tariq Alwada'n[1,*], Thair Khdour [2], Abdulsalam Alarabeyyat[3], Ali Rodan[4,5]

[1]*School of Computing, Engineering & Digital Technologies, Teesside University, Middleborough, UK*

[2]*Computer Information Systems Department, Al-Balqa' Applied University, Amman, Jordan*

[3]*Software Engineering Department, Al-Balqa' Applied University, Amman, Jordan*

[4]*Computer Information System Department, the Higher Colleges of Technology, AL Ain, UAE*

[5]*King Abdullah II School for Information Technology, the University of Jordan, Amman, Jordan*

A R T I C L E   I N F O

A B S T R A C T

A grid is a system that can manage and organize services and resources that spread amongst different control domains, employ interfaces and protocols, and offer a high quality of services. The integration of Multi-Agent Systems (MAS) with a grid environment significantly affects grid performance. MAS is considered a suitable solution for open systems that modify frequently. Grid offers a wide range of resources for its users, and some of these resources might not be used or utilized for some time before any new jobs come to the grid for processing. Usually, these resources are accumulated in massive data centers to cater to the grid users' growing demand. This accumulated will lead to consuming a considerable amount of electricity for their operation. The ever-increasing usage of grid computing has led to an increase in electrical energy usage by massive servers in their data centers. In this paper, we have proposed an automated system composed of modern agents that can be used the power wastage resulted from inactive servers inside the data centers for a specific amount of time. The proposed technique depends on switching inactive virtual and physical machines to lower power positions (Sleep/Wakeup or switched off) while still preserving customers' performance requirements. The Automated system has been tested and evaluated using the Jade tools. The results show that the newly proposed method can reduce power wastage for inactive grid resources.

1. Introduction

This paper is an extension of work presented initially in the ITT2019 conference [1]. Grid computing is an improved technology that involves a pool of linked machines that be owned by numerous organizations in diverse locations to construct a distributed system. The grid system can be accustomed to working with complicated scientific and business problems. It is created to assist in the sharing of distributed and diverse resources and to simplify resolving a considerable volume of computing issues [2]. The essential purpose of grid computing is to improve reliability, cut the cost of computing, and enhance flexibility by changing computers from an object that we purchase and operate privately to an object managed by a third organization [3]. However, grid computing is encountering various challenges, such as finding the

right resources and decreasing the number of discarded jobs [4], [5]. Figure 1 displays an example of a grid computing environment [6], where the grid may contain different types of resources which need a resource management system in order to coordinates them.

The grid structural design shown in Figure 2 contains the following components [7]: a grid portal, which is also named a grid interface. It is used by the authorized grid clients to access the grid providers (resources). The portal should hide the complexity of the grid from clients by employing an uncomplicated interface. In this manner, it enables the sorting of grid job requisites.

The second element is the grid broker, where this element is described as the core of grid computing. It performs critical jobs in creating an effective grid system by connecting the user demands to the grid providers to fulfill the clients' tasks. The broker is also responsible for reducing transmission delays and increasing the

resource assigning jobs through resources by avoiding the depending on a certain one. The grid broker's primary role is to locate and reserve the most appropriate support for the users' tasks. The broker receives the requests from the clients and sends the job's description files to the clients' requirements is responsible for monitoring the running tasks and hands over the outputs to their related users. To process the grid users' requests, the broker asks the Replica Catalogue Information Service to locate the available resources and data in the grid.
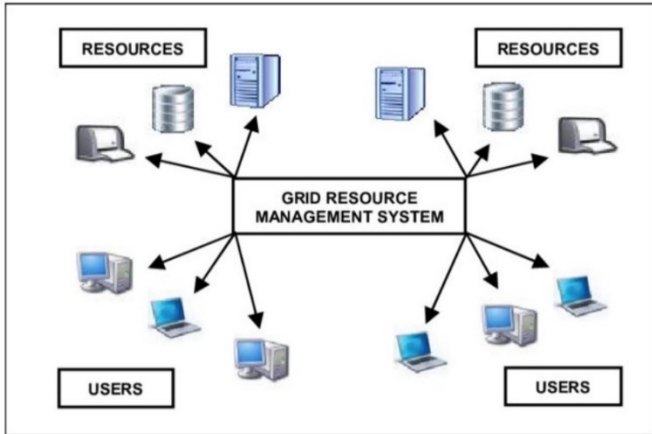
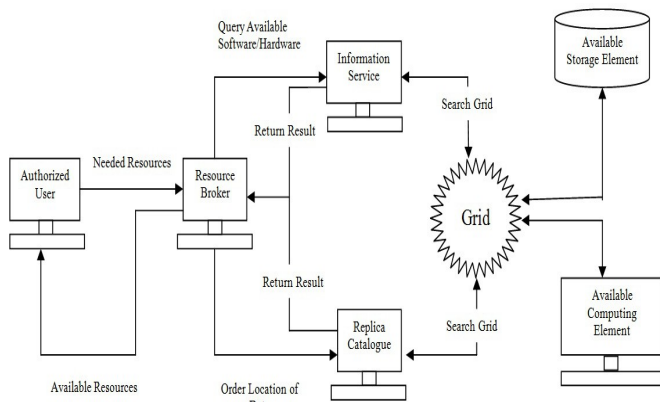

Figure 1: Grid Computing Environment

Figure 2: Grid Architecture. [8]

The third element is the Information Service. It is the directory service of the grid, where the grid keeps the information about all grid resources (nodes) and all the running jobs on those nodes. The grid resources have to register and update their information with the Information Services and their available resources. The grid broker retrieves all the information about the available and free resources from the Information Services. The information stored in the Information Services can be dynamic or static. The static information is employed to specify the operating system specifications and hardware requirements. In contrast, the dynamic information is associated with the current job running on the grid resources, the type of software specification, the available time in resources, disk space, and policies.

Replica Catalogue (RC) is the fourth element. RC is the directory that presents information, which helps in locating the saved data in the grid. To simplify employing the grid's data, the grid broker requests the RC for information about locations of the

data in the grid and the way to identify the access mechanism needed to utilize this data.

To locate the best resources for the grid clients' requests, the grid broker gets the job specifications from the grid portal and looks for the suitable resources that can fit these specifications. For this purpose, the broker inquires for resources information and the data stored in the grid from the Information Service and the Replica Catalogue. Later it chooses the resource or resources that able to fit the job specifications.

Processing and modeling jobs have grown to be more complicated as the system scopes increased. Accordingly, it is difficult to carry out tasks with such an extensive system using centralized methods. Multi-Agent System (MAS) presents an excellent manner for delivering control. It also can be used to explore and change the local communications amongst entities [9]. Furthermore, artificial intelligence mechanisms can be combined with it. Intelligent agents are incorporated to operate autonomously in a changing environment. The intelligent agent also has the following main features: [10].

- Autonomy: the power to behave dynamically to a particular point in support of the clients and programs and adjust how agents accomplish their jobs.

- Cooperation and Communication: the power to cooperate and communicate with other agents to swap information, receives orders, and respond.

- Learning: the power to enhance the performance steadily while behaving with the external world.

- Re-activity: the ability to react to exterior calls and moderate the agents behave concerning these calls.

- Pro-activity: the ability to create choices as a consequence of internal evaluations.

In this introduction, we have introduced some background about Grid and MAS as a suitable solution for open systems, such as Grid, that modify frequently. This paper is organized as follows. Section 2 introduces related works to this research. Section 3 gives the system model. The experiment and results are presented in Section 4. Lastly, the conclusion of this research is given in Section 5.

2. Related Works

In recent times, combining the MAS with grid computing has taken the interest of lots of researchers. Most topics that have been concerned in the area of grid computing are how to achieve better use of grid data storage, service distribution, infrastructures, cost, and energy efficiency. Nevertheless, the worry in the agent environment's area is on the intelligent side of agents and the procedures that can be exploited to solve complicated problems.

The authors in [11] suggested a multi agent system that can support the growth of the IDAPS micro-grid where this proposed system includes a DER agent, database agent, and control and a user agent. The Agents transmit their requirements using a TCP/IP protocol embedded in the IEEE FIPA standard to assure the system capability. In addition, authors in [12] suggested a model for agent-based grid computing (AGEGC). Based on their model, AGEGC

was created by using the MAGE (Multi-Agent Environment) platform. This research included various levels, such as; information tier, knowledge tier, service-oriented tier, and operating system tier. AGEGC focuses on service-oriented tier provided that current processing system.

In recent times, there are many paradigms concerning the growth of agent systems, such as; MOLE [13], DMARS [14], RETSINA [15]. Lots of companies are contributing to building characteristics for standard agent models; for example, OMG [16] KSE [17] and FIPA [18].

Garimella in [19] employs a new preserve server that works in assistance with the Dynamic Soft Real-Time system [20]. The purpose of the Garimella system is to reserve the resource of the CPU beforehand. By using this system, the user request for some quality of service conditions. E.g. of that the percentage of CPU needed and the beginning time and the duration. Once the request for the preservation is begun, the reserved resources are ready to be reachable for the users. One more proposed model is the Resource Broker (RB) presented in [21] merged with a new preserve server shown in [19]. Both system have continual and quick response times and numerous negotiation probability for the users. Nevertheless, the volume of re-negotiation introduces an extra-large cost to the system, in the event of an unpredicted lack of resources.

The authors in [22] presented several algorithms for keeping an improved preservation approach for supercomputer scheduling systems. Those algorithms improve traditional organizing techniques by combing the reservation demand mechanism with scheduling regular tasks mechanism. Using this way of reservations, let clients ask for more than one resource in parallel from the scheduling systems at the same time. However, the mechanisms used in [22] preserve the whole time slot, which means the resources are not carried on in a distributed way by multiple users for the same amount of time.

3. System Model

Our suggested architecture is adopted from the grid architecture in [23], as shown in Figure 3. It employs Client/Server architecture since it is the most famous architecture in the heterogeneous system [24]. Our proposed architecture uses multi-tier architecture, considering that there is a necessity for the parted database to save the grid resources' location. Our grid architecture includes the following entities; the grid portal, the grid broker, and the grid resource providers.

A grid portal is an interface for an authorized client to send their jobs to the grid. A portal should hide the complexity of the grid from clients by employing an uncomplicated interface. In this manner, it enables the sorting of grid job requisites. The grid broker can connect with agents that operate, control, and support both the grid clients and grid resource providers. The broker is an agent handler who organizes and monitors the running jobs and maintains the grid clients updated with their running jobs.

The grid broker in this paper is moved from operating as a database warehouse that saves the available resources and conditions and matches the clients' jobs with these resources into an agent manager with the ability to control event creations and create and eliminate several agents in the system. The grid broker

in our proposed architecture, is a MAS that can interconnect with the grid system on account of grid users to find the information regarding grid resource provider whom they can fit the users' requirements and choose the best one among them. Furthermore, our advanced system can provide and monitor the implementation of the jobs and support the grid users with live feedback concerning the job processes.
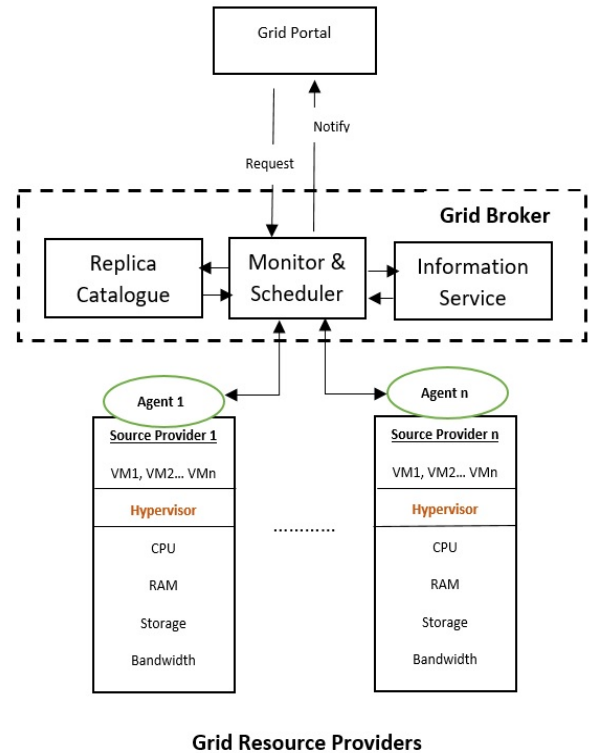

Figure 3: System Architecture

The grid broker is split into three primary entities. The first one is the monitor & scheduler. This entity is in charge of getting the grid clients' tasks and deciding the best available resource providers. The scheduler requests all data about the available resources from the Information Service entity and the data information saved in the Replica Catalogue (RC) entity. The monitor has the authority to monitor and provision the currently running jobs in the grid resource providers. The grid broker might have the improvements and ability to split clients' tasks between multiple resource providers to cut the cost and speed up the operations.

After the Monitor & Scheduler transmit the jobs to the available resource(s) for processing, it receives frequent feedbacks from the agents at the resource providers to verify the standing of the processing jobs and the percentage of the job completed to the grid broker. This point helps the grid broker to provision the running jobs' accomplishment, and by its turn, it will help determine an alternate resource provider(s) in case of failure at any point. The local agent at the resource providers is similarly in charge of updating the grid broker as the job is finished.

The resource provider can include several physicals and virtual machines. The Hypervisor in every machine is in charge of organizing and generating virtual machines on the upper of every physical machine. To react to the fast development of clients'

requests for data processing ability and storage, grid resource providers create a massive amount of data centers throughout the world. The grid datacenters typically include a large number of interconnected resources from physical machines and virtual machines, which will consume a considerable amount of electricity for their operations.

The proposed system depends on switching inactive virtual and physical machines to lower power positions (Sleep/Wakeup or switched off) while still preserving customers' performance requirements. The local agents at the resource providers are responsible for this operation. Each virtual machine (VM) has a local agent that can report the VM utilization to the data center's primary agent. The utilization here means the CPU utilization. Suppose the utilization for a specific VM has a zero amount for 300 seconds. In that case, the data center's primary agent will automatically shut down this VM to reduce the power consumption of the physical machine containing that turned off VM.

4. Implementation and Testing

The Jade platform was chosen to execute our proposed topology. Jade stands for the Java Agent Development framework, developed to help build out and enhance agent software as ruled by the FIPA conditions for smart MAS [25].

We introduced a simulation that deals with 200 physical hosts. The specifications of physical and virtual machines are shown in Table 1 and Table 2, respectively.

Table 1: Specification of Physical Machines

Type	Name	MIPS	RAM (MB)	Bandwidth (Gbps)	Number of Hosts
1	HP ProLiant ML110 G4 servers	2000	4096	2	100
2	HP ProLiant ML110 G5 servers	3000	4096	2	100

Table 2: Specification of Virtual Machines

Type	MIPS	RAM (MB)	Bandwidth (Gbps)	Number of VM
1	2000	1024	100	200
2	1500	2048	100	200
3	500	512	100	200

To create real power-consuming data, we took advantage of the actual data of power consumption offered by the SPECpower benchmark [26] and [27], as shown in Table 3. The power consumption by physical machines can be precisely expressed by a linear relationship between CPU utilization and power consumption. It can be noticed from Table 3 that even at small utilization, the machine exhausts a large volume of power. Therefore it is necessary to shut down such machines or reduce the number of VMs in it when not in use to reduce the power consumption of the physical machine. Figure 4 shows part of our simulation.

Table 3: Power Consumption (Watts per hour) at different load levels.

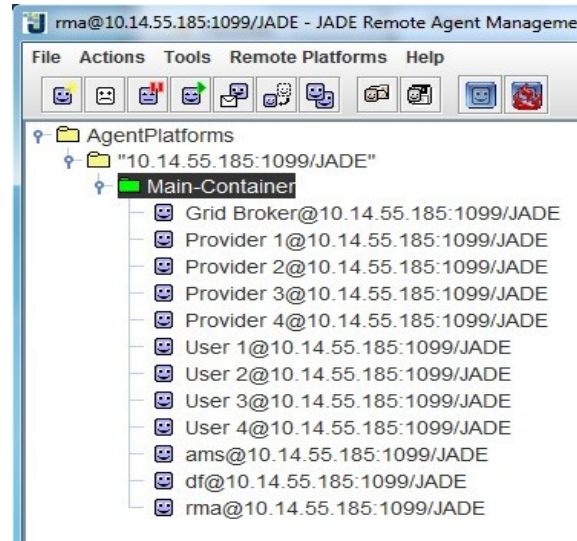| Machine type	0%	20%	50%	70%	100%
1	86	92.6	102	108	117
2	93.7	101	116	125	135

Figure 4: Part of the Summation Stage

The following stages explain the main steps of our simulation:

- Grid users state their tasks by choosing the required hardware and software specification to run their jobs through the grid portal.

- The grid broker receives the grid users' jobs and looks for the best provider(s) for those jobs. This step is done by helping the Information Services and the Replica Catalogue.

- Later, the grid broker sends the jobs to the suitable resource provider(s) and connects to the primary agent attached to that provider.

- The grid broker can divide the job between multiple resource providers to cut down the cost and speed up the operations.

- As the job is running at the resource provider, the provider's agent (primary agent) sends periodic feedbacks about the percentage of completion of the running job to the grid broker. This stage helps the grid broker to supervise the completion of the running job, and that will help find an alternative resource provider in case of failure.

- At the same time, the primary agent at the resource provider periodically checks the CPU utilization of its VMs at the physical machines.

- If the utilization reaches the zero point for 300 seconds, the primary agent will automatically shut down that VM.

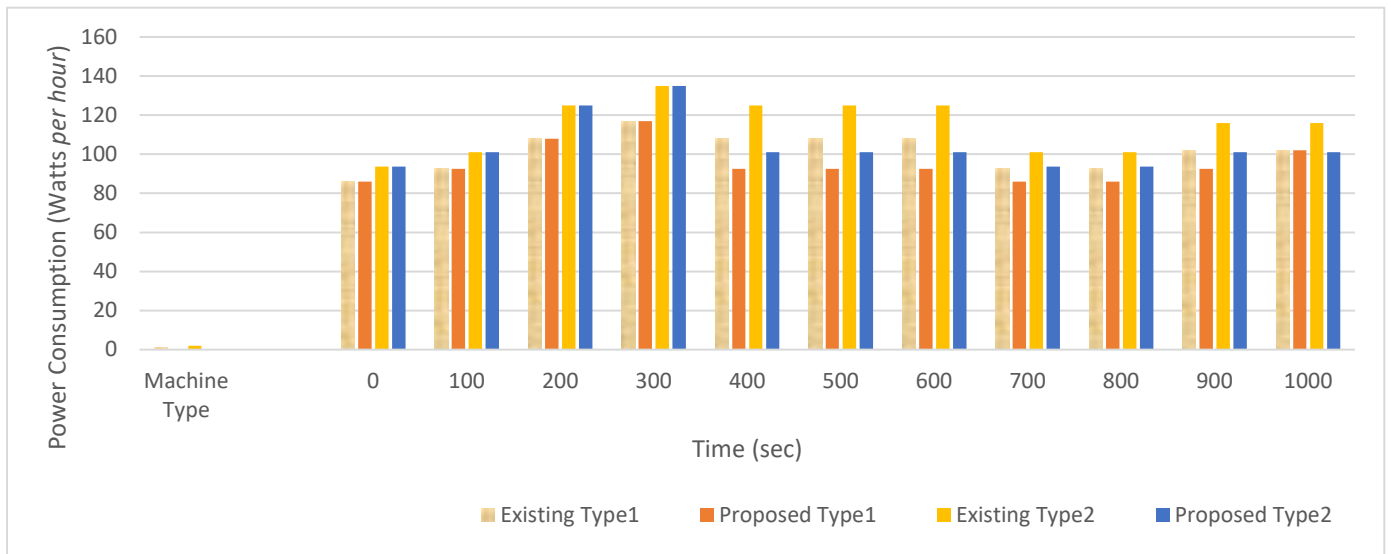- Finally, the grid broker sends the job outputs to grid users.

Figure 5: Simulation Results

From the simulation results in Figure 5. It can be seen that by using our proposed system, the power consumption has been slightly reduced for the two types of physical hosts (HP ProLiant ML110 G4 servers and HP ProLiant ML110 G5 servers), especially after the first 300s. This reduction is because the utilization of the physical machines has been reduced as the primary agents in both physical machines automatically shut down that VMs if the its utilization reaches the zero point for 300 seconds. As a result, the power consumption has been reduced in total for machines.

5. Conclusion

The purpose of the grid is to produce a broad scale, and heterogeneous systems intend to solve industrial or scientific problems. The integration of the Multi-Agent Systems (MAS) with the grid environment has a significant impact on grid performance. Grid offers a wide range of resources for its users, and some of these resources might not be used or utilized for some time before any new jobs coming to the grid for processing. As a result, the grid servers will consume a considerable amount of electricity for their operation. The growing usage of grid computing has led to an increase in electrical energy usage by the massive amount of servers in their data centers. In this paper, we have proposed an automated system composed of modern agents that can be used to reduce the power wastage resulted from inactive servers inside the data centers if the utilization of the VM reaches the zero point for 300 seconds, In this case the primary agent in the physical machine will automatically shut down that VM. The Automated system has been tested and evaluated, and the results show that the newly proposed method can reduce the amount of power wastage for the inactive grid resources.

References

[1] T. Alwada'n, A. Alarabeyyat, T. Khdour, and A. Rodan, "Utilizing Multi-Agent Systems in Grid Environments," Sixth Intentional Conference on HCT Information Technology Trends (ITT), Ras Al Khaimah, United Arab Emirates, 2019, doi: 10.1109/ITT48889.2019.9075099.

[2] T. Alwada'n, H. Aldabbas, H. Janicke, T. Khdour, and O. Aldabba. Dynamic "policy management in mobile grid environments", International Journal of Computer Networks & Communications (IJCNC), **4**(2), 2012.

[3] H. Alhakami, H. Aldabbas, and T. Alwadan. "Comparison between cloud and grid computing: Review paper", International Journal on Cloud Computing: Services and Architecture (IJCCSA), 2, 2012.

[4] T. Alwadan, H. Janicke, O. Aldabbas, and M. Alfawair. "New framework for policy support for mobile grid services", The 6th International Conference on Risks and Security of Internet and Systems (CRISIS2011), Romania 2011.

[5] T. Alwadan, H. Janicke, O. Aldabbas, and H. Aldabbas. "New framework for dynamic policy management in grid environments", In Recent Trends in Wireless and Mobile Networks, Third International Conferences, Springer, 2011.

[6] Ku. Ku-Mahamud and H. Abul Nasir. "Ant Colony Algorithm for Job Scheduling in Grid Computing", Asia International Conference on Modeling and Simulation, 2010, 10.1109/AMS.

[7] T. Alwadan, H. Janicke, A. Alarabeyyat, A. Alkharabsh, and T. Khdour. "Policy-based support for mobile grid services", International Journal of Computer Science Issues (IJCSI), 10, 2013.

[8] T. alwada'n, H. Aldabbas, and H. Janicke, editors. "Mobile Grid System: Policy and Privacy", LAP LAMBERT Academic Publishing, 2014.

[9] A. S. Grimshaw, M. A. Humphrey, and A. Natrajan. "A philosophical and technical comparison of legion and globus", IBM J. Res. Dev., March 2004.

[10] D. Talia. "Cloud computing and software agents: Towards cloud intelligent services", 2011.

[11] M. Pipattanasomporn, H. Feroze, and S. Rahman. "Multi- agent systems in a distributed smart grid: Design and implementation", 2009.

[12] Z. Shi, He Huang, J. Luo, Fen Lin, and H. Zhang. "Agent-based grid computing. Applied Mathematical Modelling", 2006, 30:629640, 07.

[13] J. Baumann, F. Hohl, K. Rothermel, and M. Straßer. "Mole – concepts of a mobile agent system", World Wide Web, Sep 1998.

[14] A. S Rao and M. P George. "Bdi agents: From theory to practice", In Proceedings of the First International Conference on Multi- Agent Systems (ICMAS-95), 1995.

[15] K. Sycara, A.e Pannu, M. Williamson, D. Zeng, and K. Decker. "Distributed intelligent agents", IEEE Expert: Intelligent Systems and Their Applications, December 1996.

[16] D. Milojicic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran, and J. White. "Masif: The omg mobile agent system interoperability facility". Personal Technologies, Jun 1998.

[17] P. Patel-Schneider, D. P McKay, T. Finin Thomas, G. Ramesh Patil, R. Fikes, and R. Neches. "The DARPA Knowledge Sharing Effort: Progress Report". Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning. Morgan Kaufman, August 1992.

[18] fipa.org. "The foundation for intelligent physical agents", Website, 2019.

[19] G. Garimella, "Advance CPU Reservations with the Dynamic Soft Real-Time Scheduler", Master's Thesis, University of Illinois at Urbana-Champaign, 1999.

[20]   H. Chu and K. Nahrstedt, "A Soft Real Time Scheduling Server in UNIX Operating System," European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS '97), Sep. 1997.

[21] K. Kim and K. Nahrstedt, "A Resource Broker Model with Integrated Reservation Scheme," IEEE International Conference on Multimedia and Expo 2000 (ICME '00), Aug. 2000.

[22] W. Smith, I. Foster, and V. Taylor, "Scheduling with Advanced Reservations," International Parallel and Distributed Processing Symposium (IPDPS '00), May 2000.

[23] T. Alwada'n, S. Alghyaline., A. Alazzam. "High Priority Requests in Grid Environment", International Journal of Advanced Computer Science and Applications, No. 10, 2020.

[24] C. Jean Dollimore, and T. Kindberg. "Distributed Systems: Concepts and Design", fourth Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[25] F. Bellifemine, A Poggi, and G. Rimassa. "Jade - A FIPA- compliant agent framework", The Practical Application Company Ltd., 1999.

[26] A. Verma, P. Ahuja, A. Neogi. "pMapper: power and migration cost aware application placement in virtualized systems", Proceedings of ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, Belgium, 2008.

[27] N. Patel, H. Patel. "Energy efficient strategy for placement of virtual machines selected from underloaded servers in compute Cloud", Journal of King Saud University - Computer and Information Sciences, 6, 2020.