Secure and Trusted Resource Provisioning for Computational Grid Infrastructure Using GridSim

N.V. Keerthana[1], K.S. Muralikrishnan[2]

[1]Department of Computer Science and Engineering, Nandha College of Technology, Affiliated to Anna University, Erode, Tamilnadu, India, nvkeerthu@gmail.com

[2]Department Computer Science and Engineering, Nandha College of Technology, Affiliated to Anna University, Erode, Tamilnadu, India, muralikrishnan.it@gmail.com

Abstract - Secure grid computing is an apparent concept combining many heterogeneous resources from multiple domains that can be used effectively to solve computational problems securely. The significant necessity is job scheduling which plays a major role in providing efficiency of the facilities provided. Job Scheduling in grid computing, defined as assigning appropriate resources to the jobs by considering multiple constraints includes cost, makespan, availability, resource utilization and so on. Due to the effect of resource dynamicity, the users will access the resources dynamically. So, there is a numerous chance of grid data modification in the resource provisioning system. The proposed algorithm with the encryption contemplates availability of the resources and success rate of the task submitted by the user ensures the data safety and the provisioning system security. The issue is that to by considering the necessary constraints given by the users for their satisfaction. This can be solved by Enhanced Particle Swarm Optimization (EPSO) scheduling which optimizes the user specified constraints in a secured manner. EPSO plays a vital role in achieving availability and success rate by rescheduling the jobs. User authentication is technically needed to avoid unwanted resource wastage and grid data modification in the grid infrastructure. Thus, EPSO also increases the trust level of the resource provisioning system.

Keywords – Security, Scheduling, EPSO, Heterogeneous resources, Resource Provisioning

I. INTRODUCTION

Nowadays, job processing process has been increased due to the availability of huge amount of data. In order to compensate this factor the need for the secured computational grid has been increasing. Computational grid is a hybrid infrastructure that has huge capacity to solve large pool of data in an effective. The presents of huge amount of resources in the computational infrastructure help in solving the problems effectively to attain a common goal. In nature, the resources are generally dynamic and heterogeneous. Management of such huge amount heterogeneous resources became more and more arduous because the data transfer rate may cause delay in the scheduling of the jobs. This major issue can be solved in the grid computing infrastructure. Grid are of different types [1-3] based on the nature of their prominent features: computation, data, interaction, knowledge and utility [4]. Since the resources are heterogeneous in nature in the grid computing system, the allocation of the resources to the tasks is an infeasible process. To attaining the high success rate is the major goal and extended to optimize the Quality of Service (QoS) parameters specified by the user [6]. The grid resource provisioning system is highly responsible for managing the resources to attain high delivery of QoS metrics to the user. The mapping of tasks to the appropriate resources is found to be an NP-complete problem [11].

Different benchmarks can be used for assessing the efficiencycy of scheduling algorithms and the most important which are makespan and flow time [32]. The motive of this paper is to assess makespan and cost as benchmarks for the scheduling function. Cost is the amount charged by the grid system to the user for the resource usage. An optimal scheduling is to optimize makespan and cost simultaneously. However, in a grid infrastructure the scheduling algorithm focus not only the subnet of jobs within a computing environment, but also its focus on the jobs present in the multiple administration domain to the available computational power [13]. Since these benchmarks are disputing in nature, any approach carried out to optimize one benchmark may not result in an overall optimized solution with respect to other benchmark. Both makespan and cost need to be addressed simultaneously and the foremost way for this to have balance with all user specifications [14].

The scheduling problem in the grid infrastructure becomes more competitive process as it is not only a vital role to achieve a promising prospective of huge distributed resources, but also using effective scheduling algorithms. Security is the major factor to be considered in the grid environment to prevent unauthorized disclosure of data [18]. The main aim of this research is to provide an enhance PSO scheduling with security to improve the overall performance and reliability of the whole grid infrastructure. EPSO schedules the tasks submitted by the user and reschedules it for the betterment of the system. Before the scheduling process the load monitoring mechanism is executed with the assistance of the load balancer. The rest of this paper is summarized as follows: Section 2 deals with related works. In Section 3, outlines Proposed System Design. Section 4 outlines the Enhanced PSO scheduling algorithm. Section 5 presents an experimental evaluation. Section 6 concludes the paper and the future enhancements.

II. RELATED WORKS

Scheduling and resource provisioning are the two predominant factors of the grid environment in enhancing the performance and efficiency of the grid system. Due to the dynamicity feature and heterogeneous form of resources, this fundamentally has taken the formulation of optimization problems. Reviews indicate small emphasis on provisioning of the resources and security based optimized scheduling in the management of the resources as discussed in this section.

In [23], proposed a QoS based Genetic Hybrid Particle Swarm Optimization (GHPSO) for task scheduling in a cloud computing environment. Here, the objective functions of GA are embedded with PSO. Here forth, achieves, higher performance than standard PSO with the minimization of cost within a specified execution time. Various algorithms have been proposed to optimize different objectives such as cost, makespan, reliability, availability, throughput and energy consumption in the computational grid [17], [19]. In [21], the mapping of tasks to appropriate resource according to the trust level defined in the resource. This trust-driven scheduling strategy keeps away from picking malicious and vulnerable resources by the users. In [8], proposed a load balancing on the enhanced GridSim with deadline control (EGDC) for the jobs submitted by the user. Nevertheless, this mechanism has serious flaws. More care has been paid for modeling the characteristics of resource availability [18], [19], [26-28]. The Fuzzy Particle Swarm Optimization (FPSO) algorithm [23] dynamically generates an optimal schedule so has to complete the jobs within a minimal period of time as well as utilizing the resource in an effective fashion. The location and also the speed of jobs square measure pictured in fuzzy matrices. The dynamic availability of resources and the cost of information transmission are not counted.

While designing a grid infrastructure, security verifications should be performed [14]. These verifications will help to govern how these new modifications will bear on the overall security of the grid environment and any other areas of change. If it detects any irregularity it immediately notifies the system. But it is not very much effective for the grid system application. Symmetric key algorithms demand less memory than asymmetric key algorithms. Furthermore, the security feature of symmetric key encryption is supercilious to asymmetric key encryption [15]. The comparison of encryption algorithms shows the authority of the Hybrid Encryption based on RSA Small-e and Efficient RSA (HE-RSA) over the other encryption algorithms based on the key size and security [13]. This algorithm provides a high level of security. It also runs faster than the other algorithms.

The proposed system is dissimilar from existing models. The main intention of this proposed system to minimize cost and makespan simultaneously by optimizing the scheduling function. This system considers the resources with the trust level and the tasks with the security demand specified by the user in order to increase the grid system reliability.

III. PROPOSED SYSTEM DESIGN

The overview system architectural design is given in Fig. 1. The security system is added before scheduling.

- Load monitoring: Monitors the load of the resource
- Synchronization: Exchanging the load information with the Grid Information System (GIS)
- Security system: Token based authentication and Token encryption mechanism are carried out
- Scheduling of tasks: Processing of tasks with the appropriate resource matching with multiple constraints specified by the user

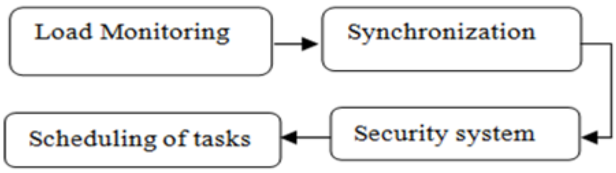

Fig. 1 Systematic Design

A. Security System Design

The security system includes two categories of process, namely as: Token based authentication mechanism and Token encryption mechanism. Token based authentication is similar to Kerberos authentication [11]. Authentication is the process of verifying whether the user is an authenticated user in order to ensure data integrity and safety of the resource provisioning system. The verifier here is the authentication server who verifies the credentials submitted by the user before granting access to the resource broker for accessing the resources from the resource provisioning system. The design of the security system is given in the Fig. 2.

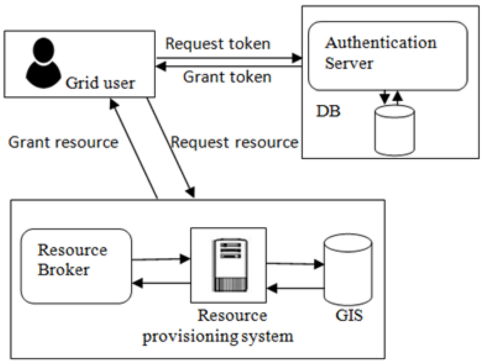

Fig. 2 Security System Design

The Security System procedure as follows:

- The grid user needs to insert a username and password for obtaining access to the resource in the grid environment
- The authentication server checks the details with the database, whether the user inserts the actual certification
- If the credentials check, the authentication server declares the user as authenticated user and generates a token that holds the username, password and network address
- HE-RSA encryption algorithm is employed to encrypt the token and the encrypted token is transmitted by the grid user to the grid server
- Then the grid user request resource to the resource broker with the avail of the encrypted token and sends a message holds username and token
- The decryption process of the token takes place along the grid server for the verification whether the username in the token and the unencrypted username in the message are same or not
- If both username matches, the grid server (resource broker) grants a resource to the authenticated grid user

HE-RSA encryption algorithm [31] is used for encryption process. It has been taken out by two successive steps, but with very same exponents, while the exponents for decrypting the cipher text have been distinct. HE-RSA is a combination of RSA small-e and the efficient RSA algorithms. RSA small-e is similar to original RSA where the public exponent is considered as smaller than φ(n) [26]. The advantage of using RSA small-e is the minimization of the encryption cost. Additionally, it reduces the process into few modular multiplications. Efficient RSA utilizes the general linear group of order k with some values that was deliberately selected from the ring of integer mod n in a random manner. HE-RSA algorithm as follows:

Key Pair Generation

1. Choose two primes p,q
2. Compute n=p.q
3. Compute $\varphi(n)=(p-1)(q-1)$
4. Compute $\alpha(n,k)=(p^k-p^0)(p^k-p^1)\dots(p^k-p^{h-1})+(q^k-q^0)(q^k-q^1)\dots(q^k-q^{k-1})$
5. Select Random Integer: r such as $1 < r < n$
6. $\gcd(r, \varphi)=1$ and $\gcd(r, \alpha) = 1$ (r should be a small integer)
7. Compute e such as $r.\, e \equiv 1 \mod \varphi(n)$ and $1 < e < \varphi(n)$
8. Compute d such as $d.\, e \equiv 1 \mod \alpha(n)$ and $1 < d < \alpha(n)$
9. Public Key: (e, n)
10. Private Key: (r, d, n)

Encryption and Decryption Process

1. Encrypt the token with the public key and transmitted to the grid server by the user
2. Encryption as: $C=((m^e \bmod n)^e \bmod n)$
3. Received token is decrypted by the authentication server
4. Decryption as: $M=((C^r \bmod n)^d \bmod n)$

B. Problem Formulation

Grid resources are heterogeneous and dynamic in nature and the concept of job scheduling is the tedious task. Thus, an efficient scheduling should be packed away. Scheduling system must optimize QoS parameters such as price, usage time, security and makespan. Choice for inspection and repairs with minimal cost and security is more complex in the gridiron. To formulate the problem, the set of n resources are considered as $\{r_1, r_2, \dots, r_n\}$ and the set of m jobs is considered as $\{j_1, j_2, \dots, j_m\}$. The jobs are considered as atomic and independent and can be executed in parallel with other available jobs. Scheduling function is specified as a mapping of tasks to resources as: S = J→R. The parameters are considered as follows: Makespan, Cost, Resource utilization. The length of the job is represented as Machine Instructions (MI). The input and output size of the job is as bytes. The resource capacity is represented as Machine Instructions per Second (MIPS). To ensure more efficiency and security, it is needed to define security demand (τ) by the grid user and the trust level (σ) by the resource provisioning system in the grid infrastructure. This security demand and trust level information provided by the GIS. The trust level determines how much the grid user can trust the resource provisioning system for executing the given tasks. The trust level determination is the summation of firewall capacity, anti-virus capacity and intrusion detection rate of the resource. It is to ensure that condition ($\tau \leq \sigma$) during the allocation time of tasks to the appropriate resource [30]. The probability of the resource failure in the resource provisioning system is determined by the difference of ($\tau - \sigma$). The failure probability is illustrated in the equation (1). There the calculative value represents the risk state of the resource. The value 0 denotes that the resource site is at high risk state. The value other than 0 denotes that the resource site is risk free.

$$\text{Probability of failure} = \begin{cases} 0, & if\ \tau \leq \sigma \\ 1 - e^{-(\tau - \sigma)}, & if\ \tau > \sigma \end{cases} \quad (1)$$

Initially the load of the resource provisioning system should be determined. According to the load capacity of the system, the tasks by the user are submitted for scheduling. The load of the system will be determined and monitored by the load manager. Additionally, the resource utilization rate is determined by the load calculation. The current load of each resource is calculated by summing all the length of the jobs submitted to the particular resource with their MIPS rating and Availability Time (AT) of the resource. The current load calculation is presented in the Equation (2).

$$\text{Load}_{r_k} = \frac{\sum_{j=1}^{n} \text{MI}_j}{\text{MIPS}_{rk} \times \text{AT}_{rk}} \quad (2)$$

Where r_k is the resource and j is the task (job) submitted by the grid user.

The resource utilization is the total amount of resources utilized by the user for executing the jobs in the system. The formula is given in the equation (3)

$$\text{Resource utilization} = \text{Load}_{r_k} \times 100 \quad (3)$$

The objective function for EPSO is the fitness function. Fitness function is the summation of cost charged for task execution in the resource provisioning system and the makespan. Make span is defined as the time completion of the latest job. Cost and makespan formula is illustrated in the equation (4) and (5) respectively. Fitness function is based on the motivation of both user and resource provider. The main motive of the resource providers is the minimization of make span and the aim of user is to execute the job with minimum cost. EPSO satisfies both user and resource provider by minimizing makespan and cost simultaneously. Fitness function is given in the equation (6)

$$\text{Cost} = \min\left(c(r_k)\right) \text{ for } 1 \leq k \leq n \quad (4)$$

$$\text{Makespan} = \max(T_j) \quad (5)$$

Where T_j is the completion time of the latest job.

$$\text{Fitness function} = \theta\text{Cost} + \delta\text{Makespan} \quad (6)$$

Where $0 \leq \theta < 1$ and $0 \leq \delta < 1$ are weights to prioritize components of the fitness function [11].

The Expected Cost (EC) of the resource and the Expected Completion Time (ECT) is defined by the grid user. Bit Rate is the rate at which the transfer of bits per second takes place. ECT is defined as the summation of load, Expected Execution Time (EET) and Data Transfer Rate (DTR). EET is defined as the expected completion time of the task submitted by the user. DTR is the rate at which the file transfer takes place. EET, EC, DTR and ECT are given in the equations (7), (8), (9) and (10) respectively.

$$\text{EET}\ (j, r_k) = \frac{\text{MI}_j}{\text{MIPS}_{r_k}} \quad (7)$$

$$\text{EC}\ (j, r_k) = \text{EET}\ (j, r_k) \times \text{Cost} \quad (8)$$

$$\text{DTR}\ (j, r_k) = \frac{\text{MI}_j}{\text{Bit Rate}_{r_k}} \quad (9)$$

$$\text{ECT}\ (j, r_k) = \text{Load}_{r_k} + \text{EET}\ (j, r_k) + \text{DTR}\ (j, r_k) \quad (10)$$

Success Rate (SR) is defined as the percentage of jobs scheduled successfully from the amount of jobs submitted for scheduling. SR is given in the equation (11).

$$SR(j) = \frac{\text{No.of Jobs Succeeded}}{\text{No.of Jobs Submitted}} \quad (11)$$

By these above objectives the secure scheduling takes place and the jobs in the queue will be scheduling completely.

C. Proposed Scheduling Algorithm

In EPSO scheduling, load monitoring and PSO algorithm are correlated. The jobs are assigned to the appropriate resource in consideration of the load level of the resource provisioning system [54]. The jobs submitted by the grid user are atomic and independent in nature. The proposed scheduling as follows:

Given n - set of resources and m - set of jobs
Initialize the jobs randomly in the waiting line (queue)
BEGIN
/*determine the Load state of the resource*/
For all resources **do**
Calculate $Load_{r_k}$ for every resource and machines
If (current resource load < machine load level)
Resource is UNDERLOADED
Else if (current resource load > resource load level)
Resource is OVERLOADED
Else
Resource is NORMALLY LOADED
End for
/* EPSO Algorithm*/
For i=1 to n **do**
For each job calculation fitness function for both planetary and local situations
Fitness function = cost + Makespan
If ($Fitness_{global\ best} > Fitness_{local\ best}$)
$Fitness_{global\ best} = Fitness_{local\ best}$
Else
Return $Fitness_{local\ best}$
Allocate the jobs to the resources with minimum Fitness and normal load
End for
Apply
While there are unscheduled jobs in the queue do
Get the jobs from the queue
Repeat the process until all jobs are allocated
END

IV. EXPERIMENTAL EVALUATION

The parameters are analyzed and presented in the Table 1. Data transfer rate increases when the jobs count increases in order to complete the within EET. The user satisfaction is determined based on the cost charged by the resource provisioning system. The cost factor is analyzed for knowing the price consumed by the resource provider based on the resource usage. The cost increases when the job count increases. The execution time is calculated in millisecond. The makespan also increases when the job count increases but it increases accordance with the cost. Thereby, the resource utilization rate is increased; overall cost of the system is decreased as per the user specification.

TABLE 1. SIMULATION RESULTS BASED ON PARAMETERS

No. of Jobs	Resource Utilization (%)	User satisfaction (%) based on Cost	Makespan (ms)	Data transfer rate (bps)
1000	92.56	93.34	133.876	240.26
2000	93.64	94.645	171.364	321.57
3000	95.21	96.253	187.56	481.69

Fig. 4, Fig. 5 and Fig. 6 shows the simulation results. Rescheduling of the tasks ensures high success rate and this scheduling algorithm schedules the unscheduled jobs present in the queue.

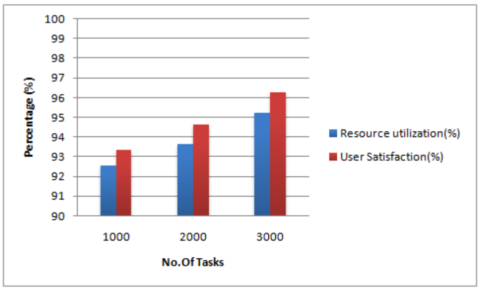

Fig. 4 Performance based on resource utilization and user satisfaction

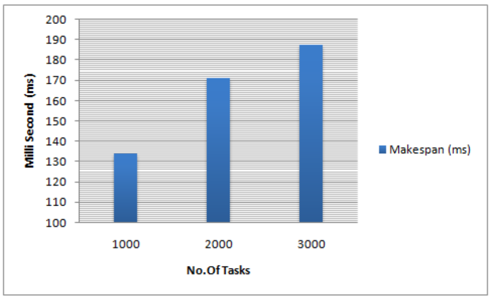

Fig.5 Performance based on makespan

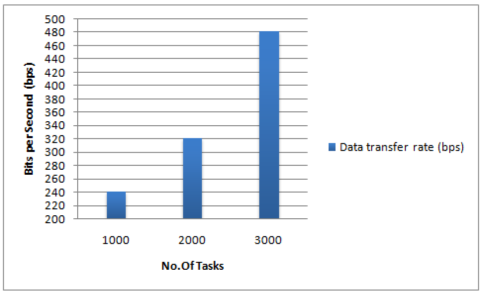

Fig.6 Performance based on data transfer rate

V. CONCLUSION AND FUTURE ENHANCEMENTS

The significant downside addressed within the system is resource management, value of the computing systems, and makespan of the tasks. To overwhelm the matter, the system integrates load equalization and best scheduling. It is settled by aggregation the data of process speeds, time, load and price of the computing resources. Agreeing to this information the scheduler submits their jobs to the resource that fulfills the anticipated price of the user. The EPSO strategy of the scheduling system improves the resource usage rate, minimizes the cost of scheduling the jobs, the finishing time of the jobs and increases the data transfer rate. The projected system not solely downplays the price, however additionally minimizes the makespan. In future, the system has the ability to function correctly even in the presence of failure of the resources. The faults may cut down the availability which makes the resource unusable and results in loss of job executions. The active nature of the grid environment introduces the challenging security. The enhancement of security will be implemented by encrypting the resource information available in the GIS. This enhancement will be the motivation for reliability increase and throughput of the grid system infrastructure.

REFERENCES

[1] Yeo C S, Buyya R, Assuncao M D, Yu J, Sulistio A, Venugopal S, Placek M, “Utility computing on global grids”, In: Bid-goli, H.(ed.) Handbook of Computer Networks, Wiley, New York, 2006.

[2] Torkestani J A, “A new approach to the job scheduling problem in computational grids”, Cluster Comput., vol. 15, no.3, pp.201-210, 2012.

[3] Pinel F, Dorronsoro B, Percero J E, Bouvry P, Khan U S, “A two-phase heuristic for the energy-efficient scheduling of independent tasks on computational grids”, Cluster Comput., vol. 16, no.3, pp.421-433, 2013.

[4] Mansouri N, “A threshold-based dynamic data replication and parallel job scheduling strategy to enhance data grid”, Cluster comput., vol. 17, no.3, pp.957-977, 2014.

[5] Asef A K, Abdullah R, Rashid N A A, “Job type approach for deciding job scheduling in grid computing system”, Journal of Computer Science, vol. 10, no. 5, pp. 745-750, 2009.

[6] Kaushik A, Vidyarthi D P, “An energy-efficient reliable grid scheduling model using NGSA-II”, Journal of Engineering and Computers, vol.15, no.9, pp. 1-22, 2015.

[7] Braun T D, Siegel H J, Beck N, Boloni L L, Maheshwaran M, Rether A I, Robertson J P, “A comparison of eleven heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems”, J Parallel Distrib Comput., vol. 61, no. 6, pp. 810-837, 2001.

[8] Raza z, Vidhyathi D P, “A scheduling model with multi- objective optimization for computational grids using NSGA-II”, International

journal of Application Evolution Computing (IJAEC), vol. 1, no.2, pp.74-94, 2010.

[9] Foster I, "The grid: a new infrastructure for 21st century science", Grid Computing: Making the Global Infrastructure a Reality, pp.51-63, 2003.

[10] Vidyarthi D P, Sarker B K, Tripathi A K, Yang L T, "Scheduling in distributed computing systems: analysis, design and models", Springer, USA, 2008.

[11] Aron R, Chana I, Abraham A, 'A hyper-heuristic approach for resource provisioning-based scheduling in grid environment', Journal on Supercomputing, vol. 71, pp.1427-1450, 2015.

[12] Kolodiej J, Khan S U, Gelenbe E, Talbi E G, "Scalable optimization in grid, cloud, and intelligent network computing", Concurr Comput., vol. 25, no. 12, pp. 1719- 1721, 2013.

[13] He X, Sun X, Laszewski G, "A QoS guided min-min heuristic for grid task scheduling", Journal on Computer Science and Technology, vol. 18, no. 4, pp. 442-451, 2003.

[14] Yu J, Kirley M, Buyya R, "Multi-objective planning for workflow execution on grids", In: Proceedings of 8th IEEE/ACM International Conference on Grid Computing, pp. 10-17, 2007.

[15] Nesmachnow S, Dorronsoro B, Pecero J E, Bouvry P, "Energy-aware scheduling on multicore heterogeneous grid computing systems", J Grid Computing, vol. 11, no. 4, pp. 653-680, 2013.

[16] Carretero J, Xhafa F, "Use of genetic algorithms for scheduling jobs in large scale Grid applications", Technol Econ Dev Econ, vol. 12, no.1, pp.11-17, 2006.

[17] Prakash S, Vidyarthi D P (2011), "Load balancing in computational grid using genetic algorithm", Adv Comput., vol. 1, no. 1, pp. 8-17, 2011.

[18] Aron R, Chan I, "Resource provisioning and scheduling in Grids: issues, challenges and future directions", In: Proceeding of IEEE International Conference on Computer and Communication Technology, MNNIT, Allahabad, 2010.

[19] Aron R, Chan I, "Resource provisioning for Grid: A policy perspective", In: Proceeding of Springer International Conference on Contemporary Computing (IC31), JP University, Noida, 2011.

[20] Gkoutioudi K Z, Karatza H D, "Multi-criteria job scheduling in Grid using an accelerated genetic algorithm", J Grid Computing, vol. 10, no.2, pp.311-323, 2012.

[21] Foster I, Roy A, Sander V, "A quality of service architecture that combines resource reservation and application adaptation", In: Proceeding of the 8th International Workshop on Quality of Service, Pittsburgh, PA, USA, pp. 181-188, June 2000.

[22] Abraham A, Izakian H, "A novel particle swarm optimization approach for grid job scheduling", Information Systems, Technology and Management, vol. 21, pp. 100-109, 2009.

[23] Garg S K, Buyya R, Siegel H J, "Time and cost trade- off management for scheduling parallel application on utility grids", Future Gener Comput Syst., vol. 26, no.8, pp. 1344-1355,2010.

[24] Sahoo R, Sivasubramaniam A, Squillante M S, Zhang Y, "Failure data analysis of a large-scale heterogeneous server environment", In: Proceedings of the International Conference on Dependable Systems and Networks (DSN), Florence, Italy, 2004.

[25] Heath T, Martin R, Nguyen T D, "Improved cluster availability using workstation validation", In: The ACM SIGMETRICS 2002, Marina Del Rey, CA, pp. 217-227, 2002.

[26] Durillo J J, Prodan R, Fard H M, "Moheft: a multi- objective list-based method for workflow scheduling", In: Proceedings of 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), pp.185-192, 2012.

[27] Song S S, Hwang K, Kwok Y K, "Trusted grid computing with security binding and trust integration", J Grid Computing, vol. 3, no. 2, pp. 53-73, 2005.

[28] Bhatia R, "Grid computing and security issues", International journal of scientific and research publications, vol. 3, no. 8, pp. 1-5, 2013.

[29] Malarvizhi N, Uthariarai V R, "Hierarchical load balancing scheme for computational intensive jobs in grid computing environment", In: Proceeding of the 1st IEEE International Conference on Advanced Computing, pp. 97-104, 2009.

[30] PreethaEvangeline D, "Survey on grid security models and mechanisms", International Journal on Computational Sciences & Applications (IJCSA), vol. 2, no. 6, pp. 57-63, 2012.

[31] Keerthana N V, Suresh P, "Security driven multi- constrained scheduling for computational grid", International Journal of Scientific & Engineering Research, vol. 7, no. 2, pp. 43-49, 2016.

[32] Nandagopal S, Karthik S, and Arunachalam VP., "Mining of meteorological data using modified apriori algorithm", European Journal of Scientific Research, Vol.47 No.2, PP.295-308, 2010.

[33] Saveetha P and Arumugam S, "Study on Improvement in RSA Algorithm and its Implementation", International Journal of Computer & Communication Technology, Vol.3 No.6,PP.78, 2012.

[34] Saveetha P, Arumugam S and Kiruthikadevi K, "Cryptography and the Optimization Heuristics Techniques", Int. Journal of Advanced Research in Computer Science and Software Engg , volume. 4, Issue.10, ISSN: 2277 128X, October 2014.

[35] Preethi, B.C. and Vijayakumar, M. " A Novel Cloud Integration Algorithm(CIA) for Energy Efficient High Performance Computing Applications in Big Data Multimedia Applications", Romanian Journal of Information Science and Technology, vol. 2, no.1, pp. 1-11, March 2018.