e-ISSN: 2349-9745
p-ISSN: 2393-8161
Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Enhancing Security System and Fault Tolerant Scheduling for Computational Grid

P. Suresh[1], P. Keerthika[2], S. Nandhini[3]
[1]*Information Technology, Kongu Engineering College,*
[2]*Computer Science and Engineering, Kongu Engineering College,*
[3] *Information Technology, Kongu Engineering College*

Abstract—In *g*rid computing, the resources are used by an organization and it becomes increasingly difficult to guarantee that resources being used by the authorized user. Also, resources may enter and leave the grid at any time. So, fault tolerance and security is a crucial issue in grid computing. For that purpose, authentication is officially needed to each user to avoid the waste usage of the resource and to control the failure of the system. Resources are managed in the system by scheduling jobs from the authorized user. The scheduling is processed through determining the load balancing technique to balance the load among the resources with their minimal price. It provides optimum solution so that it reduces the response time, computational cost and failure of jobs occurs during the execution. Through simulations the performance of the proposed system is evaluated.
Keywords- fault tolerance; load balancing; pricing; security; scheduling; computational grid.

I. INTRODUCTION

Grid computing enables access to computing resources distributed in different locations for users. Such an environment has distinctive characteristics which distinguish them from other distributed systems and also conventional parallel processing systems [2].The main emphasis on grid is resource management and job scheduling without any failure. Resource management is an efficient and effective deployment of resources in the system when it is needed. Job scheduling is a decision process by which application components are assigned to available resources to optimize various performance metrics.

Security is considered in the grid environment to prevent unauthorized disclosure or modification of data [4]. Without Security a Grid set up is left vulnerable to unauthorized users. There are two scenarios that may occur in the environment. In Grid environment sharing of resources takes place. When the user submits the job to the resource, the resource computes them and sends the result to the user, here the resource can be malicious or the jobs can be malicious. In this paper jobs is considered to be malicious and security is provided to it.

In computational grid, the system is needed to enable the environment to continue its work when one or more resources fail [1]. In this sense, a fault-tolerant service must be included while scheduling the jobs in the resources to detect errors and recover from them and thus avoiding the failure of the grid. For scheduling load balancing strategy is considered [3]. It distributes workloads across multiple computing resources and aims to improve resource use, reduce response time, maximize throughput and avoid overload.

II. RELATED WORK

The purpose of fault tolerance is to improve the performance of the system. Job Replication is considered to avoid the failure [1]. Resource level fault tolerance works at application level in each and every one of the resources in the system. Resource level failures occur due to heterogeneous nature of the system and increases complexity in system. It is avoided by replication of jobs. With redundant copies of a job, the grid can continue to provide a service in spite of failure of some grid resources carrying out job copies without affecting the performance of the grid.

@IJMTER-2015, All rights Reserved

Fair scheduling fault tolerant algorithm focuses on execution of the job from the processor failure [5]. In which, Grid Scheduler selects the minimum load of the resource to execute the job within the Grid. Then the job is transmitted to the fault detector based on the availability of the site to detect the occurrence of the failure. It achieves high throughput and resource utilization of the system.

Fault tolerance plays an important role in order to achieve availability and reliability of a grid system [11]. The techniques like replication, job migration and check pointing are used in the grid system to manage the fault tolerant. These techniques provide better reliability and improved response time. It also sort out the advanced sharing of the resources and recovery of the faults.

While designing a grid, security checks should be performed [10]. These checks will help determine how these new changes will affect the overall security of the environment and any other areas of change. Only effective security implementation in grid would ensure the reliability on grid computing. In which solution is provided to various security risks factors of the grid environment.

Intrusion detection is an Authorization mechanism [8].It keeps track of all the activity that takes place in the network. It checks the proprieties that are possible and liable for the attack. If it detects any irregularity it immediately warns the system. But it is not very much effective on grid application.

The symmetric key encryption algorithms like DES, TRIPLE DES, AES, and Blowfish are studied in this work [6]. Symmetric Key algorithms execute and require less memory than Asymmetric Key algorithms. Additionally, the security feature of Symmetric key encryption is superior to Asymmetric key encryption. The comparison of encryption algorithms shows the authority of Blowfish algorithm over the other encryption algorithms based on the key size and security. This algorithm provides a high level of security. It also runs faster than the other algorithms.

The proposed work is different from these existing works. The objective of the proposed system is to minimize time and minimize cost. The system prefers the resources without considering their failure rate, which results in disaster of the system and user is also not satisfied. The proposed techniques provide solutions to it. Along with these, the existing work have some problems like decreasing the improvement ratio and efficiency of the system, more overhead time and high cost which are also considered in the proposed system.

III. PROBLEM FORMULATION

.The dynamic nature of grid environment introduces challenging security. The services like resources can be accessed by users dynamically. These users can be intruders. So, the security should provide by name the service with acceptable identity to the user i.e., user credentials. The failure at job site slows down the overall speed of computation. It also reduces the performance of the grid system by resulting in high cost. Therefore fault tolerance is considered.These problems can be solved by considering the following parameters,

- Resources utilization
- Makespan
- Cost

Let N number of tasks has to be scheduled with the M number of resources in the grid system. Consider set of tasks and set of resources.

$\Pi = \{T_1, T_2, \ldots, T_N\}$

$\Gamma = \{R_1, R_2, \ldots, R_M\}$

Then scheduling is defined as mapping of tasks to resources.

i.e., Π Γ $\longrightarrow$

@IJMTER-2015, All rights Reserved

IV. PROPOSED SYSTEM

The main objective of the proposed system is to improve the system performance by reducing the jobs from the failure and to avoid the unauthorized user to access the system. Therefore the system incorporates the security and fault tolerance to allocate approved jobs to the resources with minimum failure rate.

4.1. Authentication Security

Authentication is the process of verifying identity of a user to an operation or request and to avoid the attempts of the user to alter system resources or affect their operation [8]. Figure 1 shows the overview of the security in the proposed system.

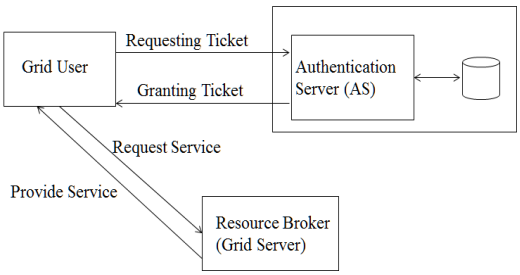

Figure 1 Overview of the Security

The steps followed in the security are [10],

- Initially grid user enters the system and requests access for using the resources. For that purpose, the user sends a message to the AS that includes the user's ID and password.
- The AS audits its database to see if the user has supplied the proper user ID and password.
- If it is correct, the AS accepts the user as authentic and creates a ticket that contains the user's ID and network address.
- This ticket is encrypted using the "Blowfish Algorithm" and the secret key shared by the AS and this server.
- Then the ticket is sent back to the user.
- With this ticket, user can request service to the grid server by sending a message, which contains user's ID and the ticket.
- Then the grid server decrypts the ticket and verifies that the user ID in the ticket is the same as the unencrypted user ID in the message.
- If it is equal, the server considers the user authenticated and grants the requested service.

The hypothetical dialogue for authentication is

- User ~~AS~~> : ID || P
- AS ~~User~~> : Ticket
- User ~~RB~~> : ID || Ticket

  Ticket = $E_{sk}(UID, AD_u)$

The algorithm used for encryption is "BLOWFISH ALGORITM" which is a symmetric key encryption [2]. It is a 64 bit block cipher with variable length key from 32 bit to 448 bits, making it ideal for securing data and also fast, free alternative to existing encryption algorithms. The algorithm has two parts- Key expansion and Data Encryption.

4.1.1. Key expansion:

The key expansion step converts 448 bit key into 4168 bytes [6]. The P-array consists of eighteen 32-bit sub keys:

P1, P2,..., P18.

There are four 32-bit S-boxes with 256 entries each:

S1,0, S1,1,..., S1,255;
S2,0, S2,1,..,, S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..,, S4,255.
Which are initialized to hexadecimal digits of π. XOR each entry in P array and S boxes with 32 bits of the key.

4.1.2 Encryption:

Blowfish has 16 rounds for data encryption. The algorithm is,
1. i/p: 64 bit (X)
2. Divide X into 32 bit halves: X_L and X_R
3. For i=1 to 16
$X_L = X_L$ XOR P_i
$X_R = F(X_L)$ XOR X_R
Swap X_L and X_R
4. Next i
Swap X_L and X_R
$X_R = X_R$ XOR P_{17}
$X_L = X_L$ XOR P_{18}
5. Combine X_L and X_R (64 bit cipher text)
For function F, Divide XL (32 Bits) into four 8-bit quarters: a, b, c, and d. Then apply to the formula,

$$F(X_L) = ((S_1,a + S_2,b \bmod 2^{32}) \text{XOR } S_3,c) + S_4,d \bmod 2^{32}$$

Decryption is the same as encryption, except the P-arrays are used in reverse.

4.2. Fault tolerance

Fault tolerance is the ability of a system to perform its function correctly even in the presence of faults [11]. In the proposed system, the grid user interacts with the Resource Broker (RB) and sends their task to computation. Then RB discovers the resources for scheduling strategies and task processing. It collects all the information fromGrid Information Service (GIS). Simultaneously the Fault Handler (FH) handles the failure of the resource by notifying and updating to the GIS. FH handles the failure by using Resource Selection Indicator (RSI) to schedule the jobs. For scheduling the jobs load balancing strategy is used. It is done by considering the current load of a resource [7]. A load is the number of jobs in the waiting queue. Current load at each resource is calculated and resources are classifiedbased on the load, whether it is under loaded, over loaded and normally loaded [9]. Thentheresourcesare selected in the under loaded list and the jobs are allocated to a resource that has minimum cost and time. For calculating load the Equation (1) is used.

$$load_i = \frac{\sum_{j=1}^{n} length}{MIPS_i \times AT_i} \quad (1)$$

Where, i-Resource

j-job

n-number of jobs that are allocated to the resource i

Then average of the load is taken by using the Equation (2).

$$Avg.load = \frac{\sum_{i=1}^{n} load}{n} \quad (2)$$

@IJMTER-2015, All rights Reserved

The cost of the resource i.e., Expected Cost (EC) is calculated using the Equation (3).

$$EC(i, j) = EET(i, j) \times \cos t_i \quad (3)$$

The Expected Completion Time (ECT) of the system is calculated using the Equation (4).

$$ECT(i, j) = \frac{\sum_{j=1}^{n} length}{MIPS_i} + EET(i, j) + DT(i, j) \quad (4)$$

where,

$$EET(i, j) = \frac{length_j}{MIPS_i} \quad (5)$$

Expected Execution Time (EET) is calculated to know how long the job will take to complete.

and

$$DT(i, j) = \frac{length_j}{baudrate_i} \quad (6)$$

Data transfer Time (DT) is calculated to know how much time will taken for the files to transfer.
Failure Rate (FR) of each resource is calculated to know the failure of resources when job is allocated to it by using Equation (7).

$$FR_i = \frac{No.\ of\ times\ a\ resource\ failed}{No.\ of\ jobs\ submitted} \quad (7)$$

After that RSI is calculated to submit the jobs to the resource which having the minimum time, cost and failure rate by using the Equation (8).

$$RSI = EC(i, j) \times ECT(i, j) \times (1 + FR_i) \quad (8)$$

At last overall execution of the system i.e., makespan is calculated by using the Equation (9)

$$makespan = \max(ECT_i) \quad (9)$$

By using these strategies the system is fully secured from the intruders and failure of the grid system is also avoided. It increases the performance of the system and the user is also satisfied.

V. A SECURED JOB SCHEDULING ALGORITHM

In which security and fault tolerance are interrelated. The distribution of the tasks is depending upon the load, time and cost of the resources [9]. Both are dynamic variables. The pseudo code for the proposed algorithm is,

STEP 1: BEGIN

STEP 2: INPUT username and password

STEP 3: CHECK the database

STEP 4: IF it is CORRECT

STEP 5: ENCRYPT the username

STEP 6: SEND the encrypted text to the user

@IJMTER-2015, All rights Reserved

STEP 7: THEN in the grid server
STEP 8: INPUT the username
STEP 9: SEND the cipher text to the RB
STEP 10: DECRYPT the cipher text
STEP 11: IF username = decrypted text
STEP 12: PRINT "Authentication Succeeded"
STEP 13: GET grid resources
STEP 14: REGISTER grid resources to GIS
STEP 15: INPUT gridlets
STEP 16: SELECT the gridlets
STEP 17: CALCULATE load at each resources
STEP 18: CLASSIFY the resources whether it is under loaded, over loaded or normally loaded.
STEP 19: SELECT the resource which is under loaded
STEP 20: CALCULATE EET and EC of the resources
STEP 21: CALCULATE FR and RSI of the resources
STEP 22: SELECT the resource that has minimum RSI
STEP 23: ALLOCATE the gridlets to those selected resources
STEP 24: CALCULATE makespan by calculating ECT
STEP 25: WRITE max (ECT)
STEP 26: IF task queue=0
STEP 27: END
STEP 28: ELSE
STEP 29: REPEAT steps 16 to 25
STEP 30: ELSE
STEP 31: PRINT "Authentication Failed"
STEP 32: ELSE
STEP 33: PRINT "Invalid username or password"
STEP 34: END

VI. RESULTS AND DISCUSSION

6.1. Parameter Evaluation

The number of gridlets are created for analyzing the parameter of the system shown in Table 1.

- Resource Utilization: The resources utilized in the system are known by calculating load to each resource and take average of it.
- Makespan: Makespan is the overall execution time of the system. It is calculated by taking the maximum value of the ECT.
- Cost: Cost is the price charged for used resources i.e., EC. It is calculated by EET and cost for each resource, based on that the processing cost of the system is calculated.

Table 1.Performance Analysis based on Parameters

@IJMTER-2015, All rights Reserved

Jobs	Resource Utilization (%)	Cost (Rs.)	Makespan (ms)
10	91	267.57	146.74
20	95	543.72	347.05
30	96	832.56	504.19

The resource utilization parameter is analyzed in percentage with the number of jobs like 10, 20 and 30. The utilization of the resource increases as the job increases because the resources are used efficiently. Figure 2 shows the resource utilization performance in the system.

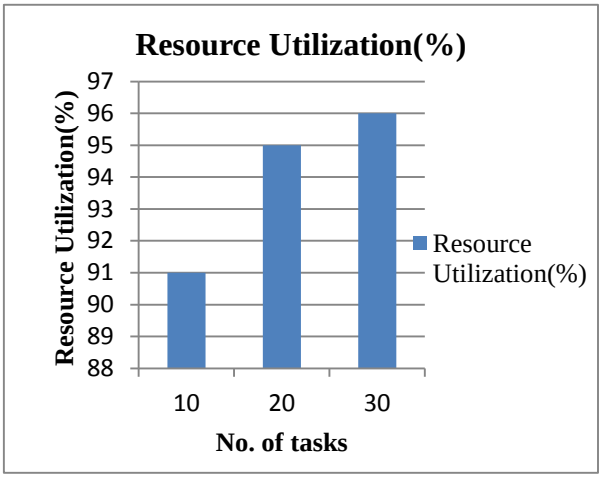

Figure 2. Performance based on Resource Utilization

The processing cost of the system is analyzed for knowing the price consumed for the usage of the resources. Here cost increases as the number of jobs increase. Figure 2 shows the cost of the system with vary amount of jobs.

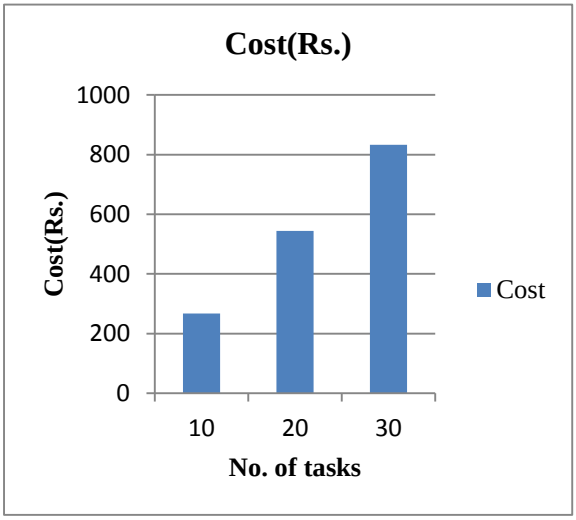

Figure 3. Performance of Processing Cost of the system

The overall execution of the system is calculated in millisecond. The makespan increases as the job increase because the time taken for all the jobs to finish processing take more time. Figure 4 shows the makespan calculation of the system.

@IJMTER-2015, All rights Reserved

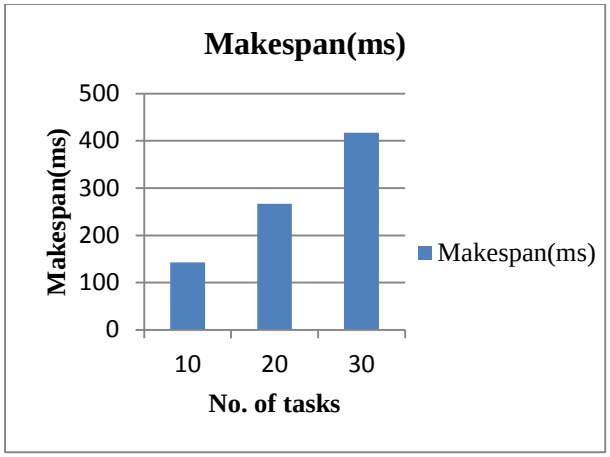

Figure 4 Performance based on Makespan

By analyzing these parameters, the system performance are increased by reduce the response time and overall cost of the system. It also maximizes the resource utilization rate and throughput with the user satisfaction.

VII. CONCLUSION

The important problem addressed in the system is masquerade i.e., unauthorized user act as an authorized one and failure of the resources leads to the system disaster. To overcome the problem, the system provides authentication security to the user and fault tolerance is considered to minimize the failure of the system. It is determined by aggregating the information of the user and failure rate of the resources. According to this information the authentic user enter the system and request service by submitting the jobs and the jobs are scheduled to the appropriate resource which is minimum failure rate. By these strategies entering of the attackers and the failure of the system is reduced. It also increases the speed of the computation time and decrease the computational cost of the system.

REFERENCES

[1] T. Altameem, 'Fault Tolerance Techniques in Grid Computing Systems', International Journal of Computer Science and Information Technologies, Vol. 4 , pp. 858-862, 2013.
[2] Ayushi, 'A Symmetric Key Cryptographic Algorithm', International Journal of Computer Applications, Volume 1 – No. 15, pp. 1-4, 2010.
[3] Dinesh Gawande, Rajesh Dharmik and ChandaPanse, 'A Load Balancing in Grid Environment', International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol-2, Issue 2, pp.445-450, 2012.
[4] Erin Cody, Raj Sharman and Raghav H. Rao, 'Security in grid computing: A review and synthesis', Science Direct, Decision Support Systems 44, pp. 749–764, 2008.
[5] S.K.KarthikumarandM.UdhayaPreethi, 'Fair Scheduling Approach For Load Balancing and Fault Tolerant in Grid Environment', IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), pp. 446-451, 2013.
[6] Monika Agrawal and Pradeep Mishra, 'A Comparative Survey on Symmetric Key Encryption Techniques', International Journal on Computer Science and Engineering (IJCSE), Vol. 4 No. 05, pp. 877-882, 2012.
[7] NeerajPandey, Shashi Kant Verma and Vivek Kumar Tamta,' Load Balancing Approaches in Grid Computing Environment', International Journal of Computer Applications (0975 - 8887), Vol-72, No.12, pp. 42-49, 2013.
[8] PreethaEvangeline.D, 'Survey on Grid Security Models and Mechanisms', International Journal on Computational Sciences & Applications (IJCSA) Vo2, No.6, pp.57-63, 2012.
[9] Qin Zheng and BharadwajVeeravalli, 'On the Design of Mutually Aware Optimal Pricing and Load Balancing Strategies for Grid Computing Systems', IEEE transactions on computers, Vol-63, No.7, pp. 407-419, 2014.
[10] Rashmi Bhatia, 'Grid Computing and Security Issues' ,International Journal of Scientific and Research Publications ISSN: 2250-3153, Vol-3, Issue 8, pp.1-5, 2013.
[11] RituGarg and Awadhesh Kumar Singh, 'Fault Tolerance in Grid Computing: State Of The Art and Open Issues', International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.1, pp. 88-97, 2011.

@IJMTER-2015, All rights Reserved