Asian Journal of Information Technology 15 (24): 5102-5105, 2016
ISSN: 1682-3915
© Medwell Journals, 2016

SRR: Secured Resource Reservation in Grid Computing

[1]S. Nirmala Devi and [2]A. Pethalakshmi
[1]Department of Computer Science, Government Arts and Science College,
Kangeyam, Tamil Nadu, India
[2]Department of Computer Science, MVM Government Arts College for Women,
Dindigul, Tamil Nadu, India

Abstract: Since, the advent of networking, the computers are vulnerable to the attack of the intruders. To tackle this, trustworthy computing was adopted in the networking environment. The grid computing environment works on the agreed virtual organizations which necessitates the secured reservation schemes. There were various reservation schemes available in grid as FCFS, priority based reservation, reservation based on negotiation, TARR, optimal resource reservation, etc. in this grid environment, there is the possibility for an intruder to reserve the resource. The intruder may send the request for the resource and reserve it to block the actual process to access the resource. This study deals with the application of challenge response authentication for resource reservation in grid computing.

Key words: Grid computing, resource reservation, security, trustworthy computing

INTRODUCTION

Grid computing works on the agreed virtual organizations to solve the problem of any nature as computing intensive, storage intensive or input-output intensive [1]. The grid computing environment shares resources. The problems of different perspective require varied resources for its earlier execution. Efficient resource management makes the problem solving in a time-efficient. Resource management includes resource discovery, resource allocation, resource scheduling and resource reservation. Among these advance resource reservation guarantees the availability of the resource for the process when required.

There were various reservation schemes available as FCFS, priority based reservation, negotiation based reservation, time slice based reservation, optimal resource reservation etc. Even the gravitational search algorithms are used in advance resource reservation (Tavakkolai *et al*., 2015). All these reservations focus on ensuring the availability of resources when required with its own pros and cons.

In this digital era, the systems are vulnerable to the attack of the intruders if proper security measures are not taken. The grid is not an exception to this. Proper security measures are required to be taken to handle the intruders.

Literature review: In the model (Viera *et al*., 2017), Mathews *et al*. proposes authentication of users rather than the devices those participate in the grid environment. The authentication is also based on the user centric approach.

In Rajesh *et al*. proposes to select the secured reliable resource by finding the RF (Reliability Factor) value. Only the resource with high RF value is selected, hence even if this ensures reliable resources, it does not handle the resource hackers.

In Chong-Yen Lee *et al*. (2009) proposes a model for secured grid environment by employing a supervisor node and a back-up supervisor node. The supervisor node verifies the authentication of the execution node or the participating node. In Abbadi proposes a secured cloud model in scheduler. Both the user requirements and the infrastructure properties are considered.

Though there were various reservation schemes available, there is no measure available when a process tries to hack a resource. This study concentrates on preventing process level hacking. Activating security at lower level creates a hack-free environment.

Secured resource reservation: Resource hackers in the grid environment intrude the environment and lock the resources. This makes the resources left unutilized hence decreases the performance. Proper mechanisms are

Corresponding Author: S. Nirmala Devi, Department of Computer Science, Government Arts and Science College, Kangeyam, TamilNadu, India

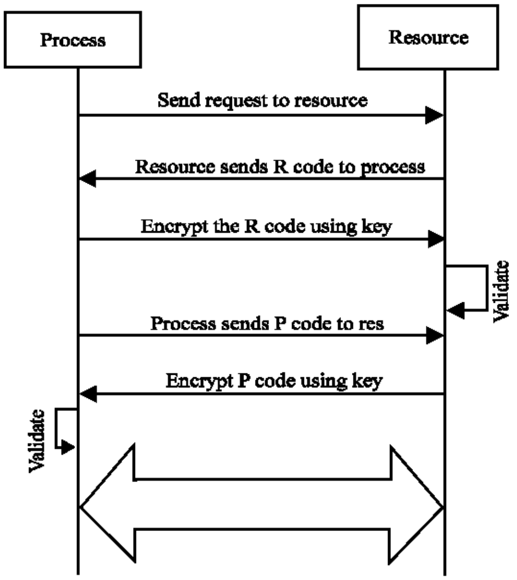
Process
Resource
Send request to resource
Resource sends R code to process
Encrypt the R code using key
Validate
Process sends P code to res
Encrypt P code using key
Validate

Fig. 1: Protocol in secured resource reservation

required to handle the intruders in the grid environment. In the proposed secure resource reservation scheme, the intruder is identified by applying challenge response authentication protocol.

Challenge response authentication protocol: In the challenge response authentication, the requesting process needs to authenticate. This process is given in the Fig 1. In the authentication protocol, the process requesting the resource sends a message to the resource it requires. The resource in turn sends a R code to the process. The processes encrypt the R code using the agreed key and send the resource where the validation takes place. Inturn the process also sends the P code to the resource which is encrypted using key and send back to the process. After validation at the process, the reservation is done.

Reservation requesting process: The secured resource reservation algorithm works on the challenge response authentication protocol as given in Fig. 1. In the algorithm Process-SRR, the process requests for resources. The receive(Rcode), receives the code from the resources. The Rcode is encrypted using the agreed key at the process and send to the resources for validation. The send(Pcode) process, sends the Pcode to the resources. The encrypted-Pcode is validated at the process.

```
Algorithm Process-SRR
Begin
•   Process $p_i$ requests for the resource $r_j$
•   Receive(Rcode)
•   Send(encrypt(Rcode,key))
•   Send (Pcode)
•   Receive(encrypted_Pcode)
•   Validate
end
```

After the validation process, the current process can reserve the resource.

Validating process at resource: The resource would wait for the process for reservation. Once if a process receive($P_{request}$) is received from the process. The resource call send(Rcode), through which the Rcode can be passed to the requesting process. Then the resource would call the receive(encrypted_Rcode), through which the encrypted code is received. It is validated. Similarly, the resources are also validated. The receive(Pcode), receives the Pcode from process. After that the encrypted_Pcode is send back to the process where the validation takes place.

Algorithm Resource-SRR:

```
Begin
•   wait()
•   receive($P_{request}$)
•   send(Rcode)
•   receive(encrypted_Rcode)
•   validate
•   receive(Pcode)
•   send(encrypted_Pcode)
•   if validated then reserve
end
```

Performance metrics: Various performance metrics are considered while incorporating the security features in the grid environment. Hacking of resources is prohibited by introducing challenge-response authentication protocol. Hence the following metrics are considered and shown improvement.

Average waiting time: The waiting time (WT) of the reservations are computed. Sometimes the resources are not available at the time of reservation requirement. But the resources can be reserved within the deferred time. At that the difference between the expected start time and the actual reserved start time is the waiting time.

$$\text{Waiting Time (WT)} = \text{Start}_{\text{reserve -}}\ \text{Start}_{\text{new}}$$

The Total Waiting Time (TWT) is computed as the sum of all the waiting time at a specific point of time.

$$\text{TotalWaitingTime(AWT)} = \sum_{i=1}^{size} \text{WT}$$

Where size refers to the length of the reservation list at a specific point of time. Then:

$$\text{AverageWaitingTime(AWT)} = \frac{\text{TotalWaiting Time}}{\text{No.of Reservations}}$$

Now because of incorporating the secured reservation the illegal resource requests are denied. Hence the average waiting time of the process will decrease.

Resource utilization time: Whenever the resources are hacked, they may not be used or they may be misused. The benefits of resource utilization cannot be enjoyed by the grid environment. If the resources are hacked then those resources may be idle even when the reservation requests are available. Many reservation policies such as TARR and ORR handle these issues. The idle time is computed based on the idle time of resources when the request is available and unable to be assigned due to illicit hold by the hacker.

$$\text{RIT} = \text{Finish}_{\text{previous}} - \text{start}_{\text{current}}$$

When there exists a reservation request with a conflict. The total resource idle time is computed by the following equation

$$\text{TotalRIT(TRIT)} = \sum_{i=1}^{\text{size}} \text{RIT}$$

The resource idle time gets decreased because of the secured reservation.

Hit ratio: Hit Ratio refers to the number of reservations accepted. At the time of requesting resources certain resources may not be available for reservation, hence the request miss would happen. The hit ratio can be computed by the formula

$$\text{Hit Ratio (HR)} = \text{Number of Hit Reservation} : \text{Total Number of Reservations}$$

Comparative analysis: The two scenarios with ten sample jobs are considered in Table 1. The JID is the Job id, ST is the Start Time of the reservation request, FT is the Finish Time and DT is the Defer Time in the Table 1. Defer Time(DT) refers to the time until which the reservation can be postponed.

In scenario 1, the Job id 7, requesting the resources as in the following Table 2. If these requests are from a hacked process, then that would take up the reservation. When the resources are allocated without authentication using FCFS, TARR etc., then the average waiting time and resource idle time decreases.

Table 1: Scenarios requesting resources

Scenario 1				Scenario 2			
JID	ST	FT	DT	JID	ST	FT	DT
J1	3	7	20	J1	1	4	10
J2	9	12	20	J2	4	6	12
J3	6	9	25	J3	8	10	20
J4	15	19	25	J4	5	8	16
J5	22	24	30	J2	12	15	25
J6	27	30	35	J5	18	19	29
J7	32	33	40	J2	20	23	33
J8	20	23	30	J6	21	23	35
J7	35	40	45	J7	25	28	38
J7	42	44	50	J8	29	31	41

Table 2: Job 7 requesting the resource reservation

JID	ST	FT	DT
J7	32	33	40
J7	35	40	45
J7	42	44	50

Table 3: Scenario 2 reservation using FCFS.

Allot	ST	ET	UT
J1	1	4	4
J2	4	6	2
J3	8	10	2
J4	Denied		
J2	12	15	3
J5	18	19	1
J2	20	23	3
J6	Denied		
J7	25	28	3
J8	29	31	2

Table 4: Scenario 2 reservation using TARR.

Allot	ST	ET	UT	WT
J1	1	4	3	0
J2	4	6	2	0
J3	8	10	2	0
J4	6	8	2	1
J2	12	15	3	0
J5	18	19	1	0
J2	20	23	3	0
J6	23	25	2	2
J7	25	28	3	0
J8	29	31	2	0

Table 5: Scenario 2 using secured reservation

Allot	ST	ET	UT	WT	RIT
J1	1	4	3	-	-
J3	8	10	2	-	-
J4	5	8	3	-	-
J5	18	19	1	-	-
J6	21	23	3	-	-
J7	25	28	3	-	-
J8	29	31	2	-	-

For the scenario 2, the reservation using FCFS is shown in the Table 3. Due to security violation J2 reserved thrice. Hence, 7 utilization time is misused. The J6 is denied because of the non-availability of resources which leads to decrease in hit ratio. The resource reservations are denied in FCFS. When using TARR, the reservations are not denied in scenario 2, rather there is waiting time. The total waiting time is 3 Table 4 and 5.

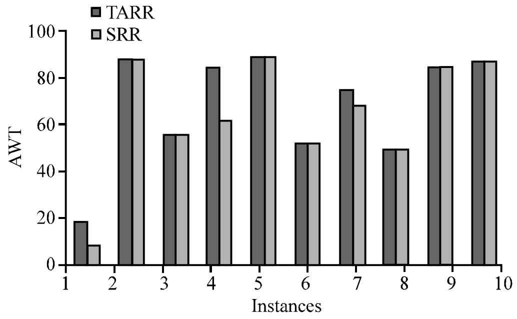

Fig. 2: Average waiting time in TARR and SRR approach

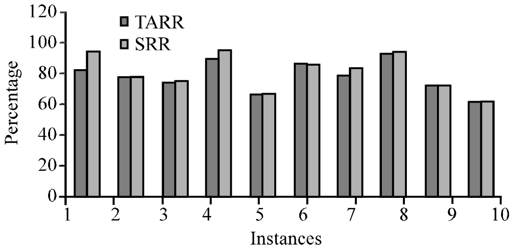

Fig. 3: Hit ratio in TARR and SRR approach

In the proposed secured resource reservation, the resource denial as in FCFS is avoided. The waiting time is also minimized when compared with the TARR. The total utilization time of the resources are also reduced in secured reservation. After a series of execution it has been arrived that there is decrease in average waiting time and increase in hit ratio. In Fig. 2, the average waiting time at various instances using TARR and SRR is shown. It is observed that the AWT in SRR is low in some instances. Similarly the hit ratio has also been increased in secured resource reservation which is shown in Fig. 3. When the reservations are denied or postponed in other approaches, SRR provides a better avenue by handling the hackers efficiently.

CONCLUSION

Thus by applying the security feature at process level, the hacking of resources is prohibited. There were various reservation schemes shown but those reservations do not handle the security issue. Even the existing security algorithms, tries to handle the hackers at the device level or user level. This paper enforces security at process level, hence the security is preserved. By implementing this secured reservation scheme the Average Waiting Time (AWT) and the resource idle time (RIT) were reduced. The Hit ratio of the resource reservation has also increased.

REFERENCES

Abbadi, I.M. and A. Ruan, 2013. Towards trustworthy resource scheduling in clouds. IEEE. Trans. Inf. Forensics Secur., 8: 973-984.

Lee, C.Y., T.Y. Lee, H. Wu, H.D. Tsui and H.S. Chen, 2009. Secure site authentication and message transmission based on grid environment. Proceedings of the 5th International Joint Conference on INC, IMS and IDC, August 25-27, 2009, Seoul, South Korea, pp: 332-337.

Tavakkolai, H., A.A.R. Hosseinabadi, M. Yadollahi and T. Mohammadpour, 2015. Using gravitational search algorithm for in advance reservation of resources in solving the scheduling problem of works inworkflow workshop environment. Indian J. Sci.Technol., Vol.8, 10.17485/ijst/2015/v8i11/71761.

Viera, M.A., C.C. Rocha, M.A. Bauer, M. Capretz and M.A.R. Dantas, 2017. Toward advance resource reservation in mobile grid configurations based on user-centric authentication. Proceedings of the VIII Workshop on Clouds Computing, June 27, 2017, ScienceCloud, Washington DC, USA., pp: 101-114.