ISSN (Online): 2456-5717

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3 Issue.5 May 2017

SHRR: Starvation Handling Resource Reservation in Grid Computing

S.Nirmala Devi[1], Dr. A.Pethalakshmi[2]
Assistant Professor and Head, Computer Science, Government Arts and Science College, Kangeyam, India[1]
Associate Professor and Head, Computer Science, MVM Government Arts College for Women, Dindigul, India[2]

***Abstract* — The grid computing allows resource sharing for earlier accomplishment of assigned tasks. The resource sharing induces efficient resource management approaches as resource discovery, resource allocation and resource reservation. Resource reservation ensures a guaranteed availability of resources, when required. There various resource reservation schemes available as FCFS, FCFS with Defer Time, Reservation based on negotiation, priority based reservation, time-slice based resource reservation, optimal resource reservation, dynamic resource reservation etc., All these reservation schemes strives to reserve for resources, but when the reservation is denied, there is no proper mechanism to handle the starving jobs. This paper handles the problem of starvation handling in resource reservation.**

***Index Terms* — Grid Computing, Resource Management, Resource Reservation, Starvation, Reservation Counter.**

I. INTRODUCTION

The geographically dispersed grid environment achieves earlier accomplishment of assigned tasks, by means of agreed polices framed in the virtual organizations. Buyya and Venugopal [1] define grid as "a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements". The grid environment consists of heterogeneous and dynamic resources. The resources in the grid environment may enter or leave the system at any period of time.

The resource management approaches are required in grid environment for prompt usage of resources. The resource management activities are broadly classified as resource discovery, resource allocation and resource reservation. Resource discovery deals with the fetching of required resources from the environment. There were various approaches available for resource discovery as centralized approach, distributed approach, ontology based approach, peer-to-peer approach, routing approach, agent based approach and hybrid approaches. The resource allocation policy handles the selecting of the required resource from the available resource. It also handles allocation of process for a particular resource.

Resource reservation guarantees the availability of resources when it is required for a particular job. There were various mechanisms available for resource reservation FCFS, FCFS with Defer Time, priority based reservation, reservation based on negotiation, time-slice based advance resource reservation, optimal resource reservation, dynamic resource reservation, secured resource reservation etc., The resource reservation concentrates on optimally reserving for the resource, but it does not focus on the resource denied jobs. Some jobs in the

environment may strive for the resources. Since there were no mechanisms available to handle the number reservation request by the same job for the same resource, a job may continuously be denied of resources. This paper focuses on this issue and provides mechanisms for rescuing starving jobs. The next section deals with the literature survey. The third section deals with the proposed work followed by performance analysis. Final section deals with the conclusion.

II. LITERATURE SURVEY

There were various resource reservation mechanisms available. These are discussed below.

In FCFS based resource reservation the start time and finish time of the reservation request are considered as input. The reservation first requested for is provided at first. Because of this approach, there were more resource denial and also the resource idle time is more.

In the FCFS with DT approach, in addition to start time and finish time, the defer time is also taken as input for resource reservation. The defer time is the time until which the resource reservation can be postponed or delayed. Because of this approach, the resources are reserved even after the finish time. This reduces the resource denial to some extent.

In the TARR (Time-Slice based Advance Resource Reservation) approach, the time slot required by the job for reservation are provided in time slices, instead of providing them in a single slot. The freely available time chunks are utilized for reservation. Though it even utilizes the time chunks of resources, it does not handle the continuously denied job.

The ORR (Optimal Resource Reservation) approach, handles the process switching. If small chunks of time are allotted, then it leads to more process switching. The overhead on process switching is highly unpredictable and it has to be reduced. Hence by optimal approach, the more convenient slot of reservation is made to the job. In the reservation based on negotiation approach, the negotiation mechanism is made between the job and resource and reservation is made.

III. PROPOSED METHOD

a) Starvation Handling Resource Reservation

In SHRR (Starvation Handling Resource Reservation), the starving process is offered with reservation, even by preempting the existing process. For this the reservation counter is maintained with each reservation request. When the reservation counter reaches a threshold value the existing reservation is revoked and the new reservation is granted.

b) Reservation Counter

While sending the resource reservation request, initially the job needs to set the reservation counter to one. When the reservation counter is one, then it means that the resource reservation request has arrived for the first time. Whenever the resource denial is made by the same resource for the same job then the resource counter is incremented by one. On reaching a threshold value, the existing resource reservation in the reservation list will be preempted and

the current request will be processed. The increment in reservation counter value is shown in the sequence diagram depicted in Fig. 1.

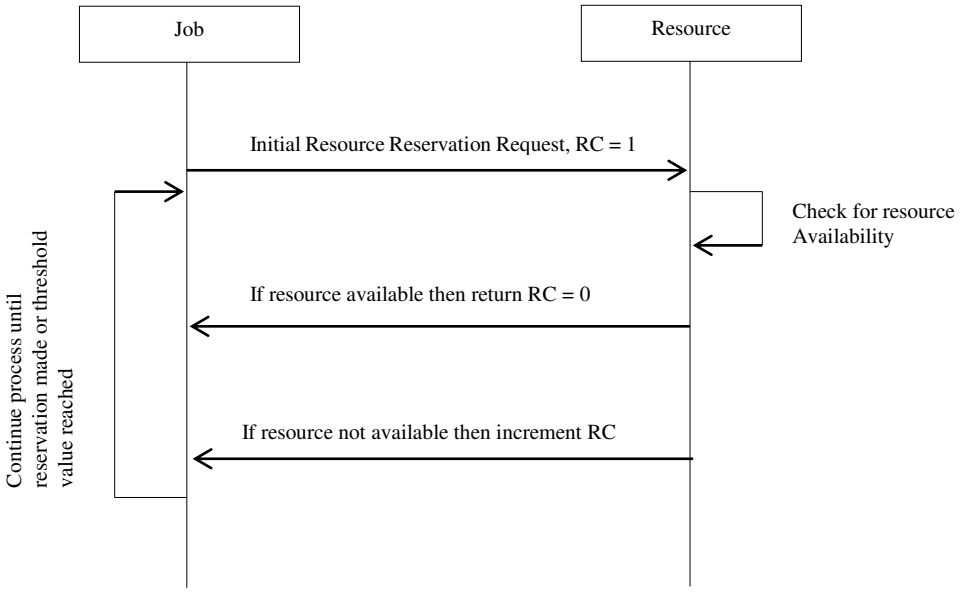

Fig. 1 Sequence Diagram showing Reservation Count

c) Fixing up the threshold value

The threshold value needs to be maintained by the LRM. The highest value of the reservation

Counter can be set as the threshold value as it entirely depends on the virtual organizations.

d) Reservation Requisition

The job always used to send the jobid, start time and finish time of resource reservation. In addition to this, the reservation counter is sent initially assigned to one value.

It needs to wait for the response. If the reservation is possible then it can be provided. If reservation is not possible then the job has to resend the reservation request to the resource.

In the res_req algorithm rc refers to the reservation counter. It is defined as static and initialized to one.

The ST and FT refers to the start and finish time of the resource reservation. The response is a Boolean variable which returns true when reservation is provided otherwise returns false.

Algorithm res_req:

```
begin
initialize  static variable rc to 1
send_request(jid, ST, FT, rc)
wait(response)
if response
return
else
if require(resource) then
resend_request(jid, ST, FT, rc)
end if
end if
Repeat steps 5 to 9 until reservation obtained.
end
```

e) Reservation Handling

At the resource, the reservation list is checked for free slots. If free slots are available then they are provided

Algorithm res_grant:

```
begin
receive(jid, ST, FT, rc)
if reservation available then
grant reservation
else
rc = rc + 1
if rc >= threshold then
revoke(curr_reserve)
assign(new_reserve)
else
Response(jid, ST, FT, rc)
end if
end if
```

IV. AN EXAMPLE SCENARIO

The typical scenario seeking resource reservation with defer time is given in Table I.

TABLE I: RESOURCE RESERVATION REQUISITION

JID	ST	FT	DT
J1	3	7	-
J2	9	12	-
J3	6	8	9
J4	15	19	-
J5	22	24	-
J6	27	30	-
J7	32	33	-
J8	20	26	39
J9	37	40	-
J10	42	44	-

It has been assumed that the job J8 is not being accepted with the defer time of 39 for resource reservation. But when it is not accepted then the J8's reservation counter would be incremented whenever it comes with the same request. The reservation counter is added with the table and presented in Table II.

TABLE II: RESOURCE RESERVATION REQUEST WITH RESERVATION COUNTER

JID	ST	FT		DT	Reservation Counter
J1	3	7		-	1
J2	9	12		-	1
J3	6	8		9	1
J4	15	19		-	1
J5	22	24		-	1
J6	27	30		-	1
J7	32	33		-	1
J8	20	26		39	1
J8	20	26		39	2
J8	20	26		39	3
J9	37	40		-	1
J10	42	44		-	1

On negotiation, the job J8 has refused with the defer time of 39. But it is in need of the resource reservation so it continuously resends the request. On reaching the threshold value of 3 for the resource counter, the resource reservation of job id J5 is revoked and the reservation is provided for J8.

ISSN (Online): 2456-5717

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3 Issue.5 May 2017

TABLE III: RESOURCE RESERVATION USING SHRR

Job – id	Start Time	Finish Time	Utilization Time	Waiting Time
J1	3	7	4	0
J3	7	9	2	1
J2	9	12	3	0
J4	15	19	4	0
J8	20	26	2	0
J6	27	30	3	0
J7	32	33	1	0
J9	37	40	3	0
J10	42	44	2	0
Total			24	1

The following Gantt chart in Fig. 2 shows the reservation scheme by applying the SHRR

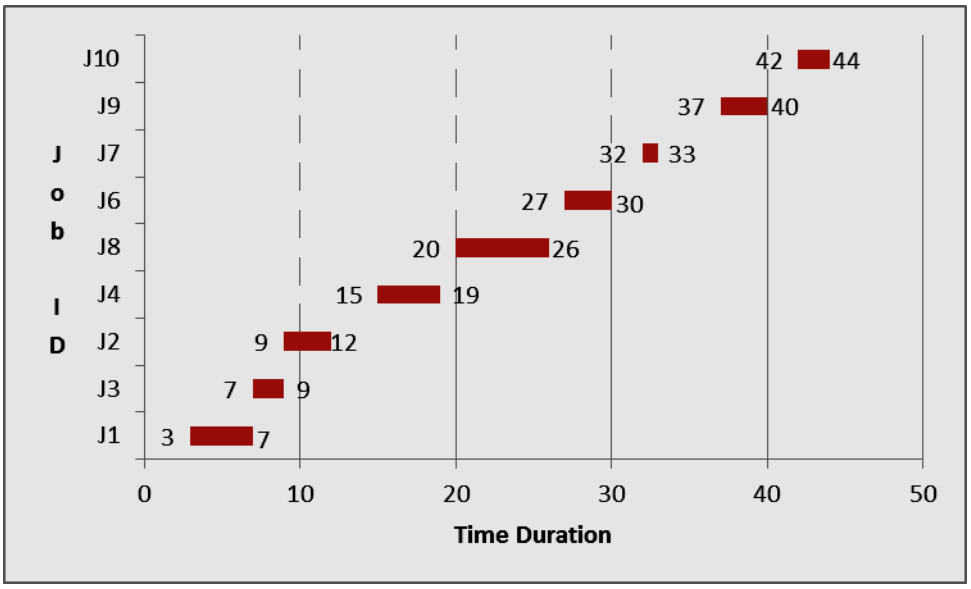

Fig. 2 Resource Reservation Using SHRR

V. EXPERIMENTAL RESULTS

The parameters as hit ratio, resource idle time, average waiting time and makespan are considered while applying SHRR. Table IV represents the comparative analysis of the approaches discussed so far.

TABLE IV: COMPARATIVE ANALYSIS OF VARIOUS APPROACHES

S. No	Approach	No of jobs	Total Execution time	Total Waiting Time	Makespan
1	FCFS with DT*	9	24	1	25
2	TARR	10	30	6	36
3	DRR	9	27	3	30
4	SHRR	9	24	1	25

*The TUT and TWT are low in FCFS with DT because this approach has denied one reservation. The makespan has been reduced in this SHRR approach and also the resources are reserved for striving resources. The comparison is depicted in Fig. 3.

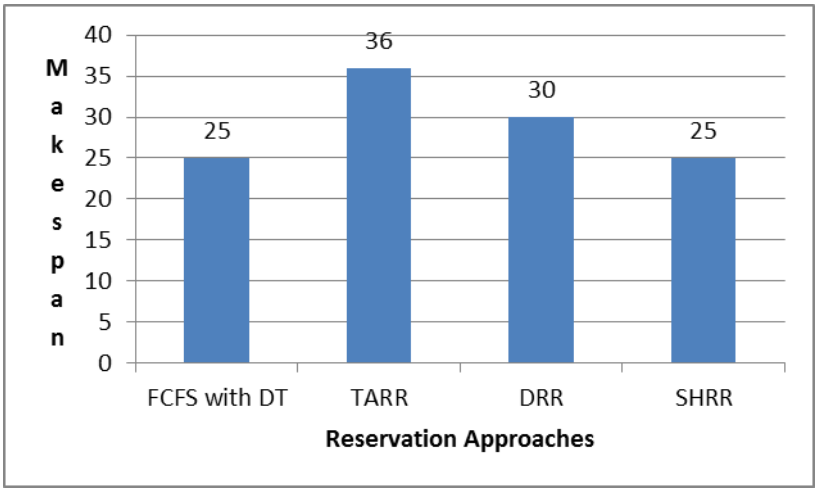

Fig. 3 Comparison of Makespan for various Approaches

VI. CONCLUSION

This reservation scheme ensures the resource reservation for starving processes. The reservation counter is used to monitor the starving processes. When the reservation counter reaches the threshold value the process seeking the reservation is provided by revoking the existing reservation. This reduces the waiting time of the reservations, which in turn reduces the makespan.

REFERENCES

[1] Buyya, R.; Venugopal, S. "A Gentle Introduction to Grid Computing and technologies". CSI Communications 2005, 29(1), 9–19. [Google Scholar]

[2] C.S.R. Prabhu "Grid and Cluster computing", PHI Learning Pvt. Ltd, 2010 pp.31 – 32.

[3] Joseph, Joshy, Mark Ernest, and Craig Fellenstein. "Evolution of grid computing architecture and grid adoption models." IBM Systems Journal 43, no. 4 (2004): 624-645.

[4] Foster, Ian, and Carl Kesselman, eds. The Grid 2: Blueprint for a new computing infrastructure. Elsevier, 2003.

[5] S.Nirmala Devi and Dr.A.Pethalakshmi, "DRR: Dynamic Resource Reservation in Grid Computing", International Journal of Computer Applications (Accepted for Publication).

[6] S.Nirmala Devi and Dr.A.Pethalakshmi, "Enhancing Resource Reservation by Computing and Applying Defer Time in FCFS Scheme for Grid Computing Environments".

[7] S.Nirmala Devi and Dr.A.Pethalakshmi, "ORR: Optimal Resource Reservation in Grid Computing Environments", Indian Journal of Science and Technology, Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/102300, December 2016

[8] S.Nirmala Devi and Dr.A.Pethalakshmi, "TARR: Time Slice based Advanced Resource Reservation in Grid Computing Environments", International Journal of Computational Intelligence and Informatics, Vol 6: No 1, June 2016, pp.1-7.

[9] Sulistio, Anthony, and Rajkumar Buyya. "A grid simulation infrastructure supporting advance reservation." In 16th International Conference on Parallel and Distributed Computing and Systems (PDCS 2004), pp. 9-11. 2004.