

DRR: Dynamic Resource Reservation in Grid Computing

S. Nirmala Devi
Research Scholar
Manonmaniam Sundaranar University
Tirunelveli, India

A. Pethalakshmi, PhD
Associate Professor and Head
MVM Government Arts College for Women
Dindigul, India

ABSTRACT

The computationally hazardous problems necessitate deploying the complexity in the grid environment for the earlier execution. This can only be achieved by resource sharing. To ensure the availability of resources at the required time, the resources are reserved in advance. The available advance resource reservation schemes are FCFS, priority based reservation, reservation based on negotiation, time slice based advance resource reservation and optimized resource reservation. In all the reservations, it is assumed that the reservations done are utilized, but there are some situations where the reserved resources are kept idle. This paper analyzes the reservations which are unutilized and allocates the unutilized reservations to the current requirements.

Keywords

Grid Computing, Advance Resource Reservation, time-slice based reservation, Optimized resource reservation, dynamic resource reservation.

1. INTRODUCTION

In grid computing environment, the resources are shared among the virtual organizations for the earlier execution of the processes. The resources are of different types as data resource, network resource, storage resource, computing resource, web resources etc., [1]. These resources are required to be made available for execution, when required. Instead of fetching the resources when required the resources can be reserved in prior.

There were various resource reservation schemes available as FCFS (First Come First Served), reservation based on negotiation, TARR (Time Slice based Advanced Resource Reservation), ORR (Optimized Resource Reservation). In all these methods, the reservations done are not checked for utilization at any period of time. But there are chances of unutilized reservation due to network failure, termination of parent process, termination of current process etc.,

These unutilized reservations can be used for the current reservation. This paper explores the avenues available to utilize the unutilized resources in the grid environment. In this paper, the next section deals with the literature survey, section 3 deals with the dynamic reservation scheme, section 4 deals with the performance metrics, section 5 with the comparative analysis and finally the conclusion.

2. LITERATURE SURVEY

ARR in grid computing environment provides a guaranteed availability of the resources for the task to be completed. Various researches are going on in this advance resource reservation. All the reservation schemes take the Start Time (ST) and Finish Time (FT) of the resource as the input to make the reservation.

Srikumar et. al., [2] proposes the negotiation mechanism between the proposer and the responder. The resource requester is the proposer and the resource holder is the responder. The resource requester sends the request and a unique negotiation id is agreed between the proposer and responder. Then the proposal is submitted which can be accepted or rejected. All these operations are performed by the proposer and the responder, there are no time-limits or any constraints imposed in the protocol.

Rui Min et. al., [3] proposes reservation based on the priority assigned. In this approach the advance reservations are made on priority basics whenever tasks with same start time request for the resource. Always the reservation request comes with the priority. And the system tries to make the at most profit by providing the resources to the high priority request.

Sara Rezaei et al., [9] proposed the irregular cellular learning automata for resource reservation. In this approach the neighbor nodes are checked for the resource availability based on the learning automata hence the rejected requests are minimized to some extent. Mohamed et. al., [10] proposes the advance resource reservation framework in multidomain scheduling approach for data intensive application. The iterative scheduling algorithm and K-shortest paths algorithm are used for reservation. Xiao et. al., [11] proposes the adaptive redundant reservation with overlapping strategy where the free slots are identified by neglecting the start time but rather it assumes the completion of existing reservation beforehand.

J.Chen et.al., [12] makes use of the FCFS and easy backfill algorithm for resource reservation in grid environment. In [5], Eliza et.al., proposes the resource reservation for an opportunistic computing environment. When a resource reservation is made then the available nodes are checked for the free slots. If free slots are available then the reservation is done. If free slots are not available during the current reservation request then the next available free slot is reserved.

Barzehir et.al.[8] proposes an approach where the resources are managed by grouping on QoS and Semantics. Anthony Sulistio et.al.[4] proposes the advance resource reservation based on the FCFS approach. When a resource is requested for reservation then the reservation can be done on the First Come First Serve (FCFS) basis. A list is maintained at each resource for reservation. When a new request for reservation comes in then the list is checked with the start time and finish time of the existing reservations and new request. If it is possible to fit in the reservation within the empty slots then the reservation can be done otherwise the reservation is rejected. Nirmala devi et.al.[6] proposes the TARR (Time Slice based Advance Resource Reservation) which reserves the resource when time slice is free, by including defer time (DT) which is the time until the job can be completed. In ORR approach the reservation is done based on switching. As

process switching incurs more cost this scheme reduces the number of process switching.

3. PROPOSED SYSTEM

3.1 Grid Architecture

In the grid environment, there are many computational nodes. These computational nodes used to have resources associated with it. The Local Resource Manager (LRM) at the computational node, handles the details about the resources and manages the resources. The architecture diagram, in figure 1, shows the placement of computational nodes, the resources and the LRM.

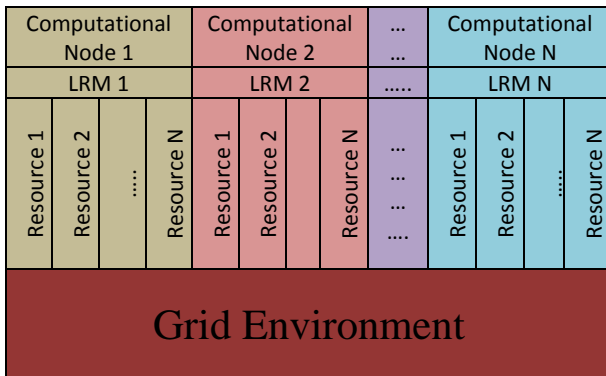


Fig 1: Grid Architecture

3.2 Resource States

Striving for efficient utilization of resources enhances the performance in grid environment. The resources can be data resource, computational resource, network resource, web resource. Whatever be resources, from the utilization point of view the resources fall into the categories as available, reserved, in-use etc., This is shown in the state transition diagram in figure 2.

Once the computational node becomes the participating node in the grid, then the resources associated with that computational node move to the available state. The available state means that the resources are available for reservation or use. When the process requires the resources immediately then it can use, hence the resource moves to the in-use state. When the process requires the resources after a period of time, then the resources can be reserved during that time slot. For that time slot alone the resource would be in reserve state.

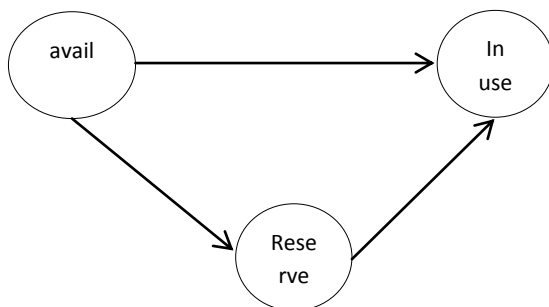


Fig 2 : Resource States

3.3 Reserve States

The resources required are reserved for future use. There are some abnormalities like a) termination of parent process may lead to the termination of child process b) network failure c) suspension of current process d) blocking of current process etc., After reserving the resources due to these abnormalities, the reservation may not be required. But these abnormalities

are unknown at the LRM and the reservations are kept as such. So, in the proposed DRR scheme the reservations are checked whenever a new reservation requested.

3.4 Dynamic Resource Reservation

In the proposed DRR scheme, the resource is checked for availability. If the resources are available then it is allotted. If the resources are already reserved then it is checked for the reservation requirement at that particular point of time. If the reservation requirement is not required then the current reserved slot is provided to the current process.

3.4.1 Slice Queue

The free time slots are maintained in the slicequeue for every resource upto a specified time. Normally, the slicequeue used to have the start time and finish time. Whenever the resource request is made then the availability is checked in this slicequeue. If the time slot is available then it is provided. If the time slot is not available then the DRR algorithm can be used to get the required slice from the slice queue depending upon the possibility

3.4.2 DRR Algorithm

The algorithm DRR shows the proposed reservation scheme. The algorithm takes in the process id, start time (ST), finish time(FT) and defer time (DT) as input.

Algorithm DRR (process_id, ST, FT, DT)

```

begin
  If the list is empty then
    No conflict found. Hence accept the reservation.
  else
    get_jobid(timeslot)
    if isalive(jobid)
      if isrequire(timeslot)
        return()
      else
        allot(current_jobid, timeslot)
      end if
    else
      allot(current_jobid, timeslot)
    end if
  end if
end
  
```

If the reservation list is empty then it means that the resource is free during the required time slot. Hence the reservation can be done. When reservation is already available during the time slot, then the job can be checked for the availability. The islive function returns whether the reserved job id is still alive or not. If the reserved job is not alive then the slot is available for the current reservation hence that can be allotted for the current reservation. If the required time slot is reserved then the isrequire function is called which determines whether the reservation made is further required or not. If it is not required then the time slot can be used for the current reservation.

4. PERFORMANCE METRICS

There were various performance metrics that are considered to evaluate the performance of the proposed DRR algorithm. The average waiting time, Hit Ratio, Resource IdleTime and makespan are taken as metrics for evaluation

4.1 Average Waiting Time(AWT)

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. In that case, 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.

$$\text{Total Waiting Time (TWT)} = \sum_{i=1}^{\text{size}} \text{WT}$$

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

$$\text{Average Waiting Time(AWT)} = \frac{\text{Total Waiting Time (TWT)}}{\text{No of Reservations}}$$

The AWT decreases while applying DRR reservation.

4.2 Hit Ratio (HR)

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

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

The HR increases in case of DRR when compared with TARR and ORR as the reservations are granted by checking on the current reservations.

4.3 Resource Idle Time (RIT)

The resources may be idle even when the reservation request available. This happens when the idle time does not fit into the allocation policy. Thus TARR provides a better allocation policy, as even the time-slices are used for reservation rather than allocating the entire request as a single unit. The RIT is computed by applying the below formula

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

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

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

The RIT also decreases incase of the TARR and DRR.

4.4 Makespan

The makespan denotes the time duration taken for the completion of execution of the job. The algorithms strive to minimize the makespan.

$$\text{Makespan} = \sum_{i=1}^n \text{TET} + \sum_{i=1}^n \text{TWT}$$

Where TET is the Total Execution Time and TWT is the Total Waiting Time, and the sum of the TET and TWT provides the makespan . The total execution time denotes the time taken by the job for its execution. The total waiting time is the time the job is made to wait for getting the resource. Thus make span is the time from which the job

5. COMPARATIVE ANALYSIS

The proposed algorithm can be analyzed by considering the scenarios under various circumstances. The table 1 depicts the reservation request at a time. Four scenarios are considered here. The defer time is also obtained from the job. The defer time is the time until which the job can be made to wait for reservation. The Utilization time (UT), Waiting Time (WT), Resource Idle Time (RIT) are computed in the FCFS, TARR and the DRR approach. Finally the Makespan is obtained.

Ten jobs are considered in all the scenarios and the resource request is made. The table 2 depicts FCFS based resource reservation with the start time, end time, utilization time, waiting time and resource idle time. Here in scenario 1, the job J3 is allowed to wait within the deferred time and the resources are reserved only in the start time of 19. And this is followed in the scenario 2 for Job id 4 and J3 in Scenario 4. Even the J8 is denied in scenario 1, as the reservation could not be made within the defer time. In scenario 2, the job J8 cannot be reserved since the defer time is 35. But the reservation can only be done until 36 which violates the defer time. Hence the hit ratio for scenario 2 in FCFS is 9:10.

The table 3 depicts the TARR approach for resource reservation. J8 which is denied in scenario 1 is able to get the slot in TARR approach. In the same scenario, J3 which is made to wait until 19 is also allowed as slices from 7 to 9 and 12 to 13. Hence the average waiting time is reduced. The resource idle time (RIT) is 4 in scenario 2. Similarly the J8 which is denied in scenario using FCFS is granted in TARR. Though the average waiting time of TARR is same as FCFS in scenario 2, the hit ratio has been increased as the J8 is granted.

Table 1. Four Scenarios Considered for Resource Reservation

SCENARIO 1				SCENARIO 2				SCENARIO 3				SCENARIO 4			
JID	ST	FT	DT	JID	ST	FT	DT	JID	ST	FT	DT	JID	ST	FT	DT
J1	3	7	20	J1	1	4	15	J1	2	5	15	J1	1	3	10
J2	9	12	20	J2	4	6	25	J2	5	9	20	J2	4	7	15
J3	6	9	25	J3	8	10	20	J3	10	15	25	J3	2	4	20
J4	13	19	25	J4	5	8	25	J4	8	10	25	J4	8	10	25

J5	22	24	30	J5	12	15	25	J5	17	19	30	J5	12	16	30
J6	27	30	35	J6	18	19	30	J6	20	25	35	J6	18	21	35
J7	32	33	40	J7	20	23	35	J7	26	28	40	J7	17	19	40
J8	20	24	30	J8	21	25	35	J8	21	24	40	J8	23	26	40
J9	35	40	45	J9	25	28	40	J9	30	33	45	J9	27	30	45
J10	42	44	50	J10	29	31	40	J10	35	37	50	J10	30	32	45

Table 2. FCFS based Resource Reservation

Scenario 1						Scenario 2						Scenario 3						Scenario 4					
All	S	E	U	W	R	All	S	E	U	W	R	All	S	E	U	W	R	All	S	E	U	W	R
Ot	T	T	T	T	IT	ot	T	T	T	T	IT	ot	T	T	T	T	IT	Ot	T	T	T	T	IT
J1	3	7	4	0	0	J1	1	4	3	0	0	J1	2	5	3	0	0	J1	1	3	2	0	0
J2	9	12	3	0	2	J2	4	6	2	0	0	J2	5	9	4	0	0	J2	4	7	3	0	1
J4	13	19	6	0	1	J3	8	10	2	0	2	J3	10	15	5	0	1	J4	8	10	2	0	1
J3	19	22	3	13	0	J5	12	15	3	0	2	J4	15	17	2	7	0	J3	10	12	2	8	0
J5	22	24	2	0	0	J4	15	18	3	10	0	J5	17	19	2	0	0	J5	12	16	4	0	0
J6	27	30	3	0	3	J6	18	19	1	0	0	J6	20	25	5	0	0	J6	18	21	3	0	1
J7	32	33	1	0	2	J7	20	23	3	0	0	J7	26	28	2	0	1	J7	21	23	2	4	0
J9	35	40	5	0	3	J9	25	28	3	0	0	J9	30	33	3	0	2	J8	23	26	3	0	0
J10	42	44	2	0	2	J10	29	31	2	0	0	J10	35	37	4	0	2	J9	27	30	3	0	0
												J8	37	40	3	16	0	J10	30	32	2	0	0

Table 3. TARR based Resource Reservation

Scenario 1						Scenario 2						Scenario 3						Scenario 4					
All	S	E	U	W	R	All	S	E	U	W	R	All	S	E	U	W	R	All	S	E	U	W	R
Ot	T	T	T	T	IT	Ot	T	T	T	T	IT	Ot	T	T	T	T	IT	Ot	T	T	T	T	IT
J1	3	7	4	0	0	J1	1	4	3	0	0	J1	2	5	3	0	0	J1	1	3	2	0	0
J3	7	9	2	1	0	J2	4	6	2	0	0	J2	5	9	4	0	0	J3	3	4	1	1	0
J2	9	12	3	0	0	J4	6	8	2	1	0	J4	9	10	1	1	0	J2	4	7	3	0	0
J3	12	13	1	3	0	J3	8	10	2	0	0	J3	10	15	5	0	0	J3	7	8	1	3	0
J4	13	19	6	0	0	J4	10	11	1	2	0	J4	15	16	1	5	0	J4	8	10	2	0	0
J8	20	22	2	0	0	J5	12	15	3	0	0	J5	17	19	2	0	0	J5	12	16	4	0	0
J5	22	24	2	0	0	J6	18	19	1	0	0	J6	20	25	5	0	0	J7	17	18	1	0	0
J8	24	26	2	2	0	J7	20	23	3	0	0	J8	25	26	1	4	0	J6	18	21	3	0	0
J6	27	30	3	0	0	J8	23	25	2	2	0	J7	26	28	2	0	0	J7	21	22	1	3	0
J7	32	33	1	0	0	J9	25	28	3	0	0	J8	28	30	2	2	0	J8	23	26	3	0	0

J9	35	40	5	0	0	J8	28	29	1	3	0	J9	30	33	3	0	0	J9	27	30	3	0	0
J10	42	44	2	0	0	J10	29	31	2	0	0	J10	35	37	2	0	0	J10	30	32	2	0	0
						J8	31	32	1	2	0												

Table 4. DRR based Resource Reservation

Scenario 1						Scenario 2						Scenario 3						Scenario 4					
All Ot	S T	E T	U T	W T	R IT	All Ot	S T	E T	U T	W T	R IT	All Ot	S T	E T	U T	W T	R IT	All Ot	S T	E T	U T	W T	R IT
J1	3	7	4	0	0	J1	1	4	3	0	0	J1	2	5	3	0	0	J1	1	3	2	0	0
J3	7	10	3	1	0	J2	4	6	2	0	0	J2	5	9	4	0	0	J3	3	4	1	1	0
J4	13	19	6	0	0	J4	6	8	2	1	0	J4	9	10	1	1	0	J2	4	7	3	0	0
J8	20	22	2	0	0	J3	8	10	2	0	0	J3	10	15	5	0	0	J3	7	8	1	4	0
J5	22	24	2	0	0	J4	10	11	1	2	0	J4	15	16	1	5	0	J4	8	10	2	0	0
J8	24	26	2	2	0	J5	12	15	3	0	0	J5	17	19	2	0	0	J5	12	16	4	0	0
J6	27	30	3	0	0	J6	18	19	1	0	0	J6	20	25	5	0	0	J7	17	19	2	0	0
J7	32	33	1	0	0	J8	21	25	4	0	0	J8	25	26	1	4	0	J8	23	26	3	0	0
J9	35	40	5	0	0	J9	25	28	3	0	0	J7	26	28	2	0	0	J9	27	30	3	0	0
J10	42	44	2	0	0	J10	29	31	2	0	0	J8	28	30	2	2	0	J10	30	32	2	0	0
												J9	30	33	3	0	0						
												J10	35	37	2	0	0						

The table 4 depicts the DRR approach for resource reservation, which provides a better approach than FCFS and TARR

In Scenario 1, when J3 requests for the resource, it is learned that J2 does not require the resource hence J3 is allotted. Hence the average waiting time and the makespan is reduced for scenario 1 in the proposed DRR. In scenario 2, J7 does not require the reserved resource further, hence the waiting time of J8 is reduced. In scenario 3, the resource reservation is required. In scenario 4, when job 7 requests for the resource it has been identified that the Job 6 does not require the resource further. Hence the reservation of job 6 is cancelled and the job 7 is provided.

Table 5 depicts the comparative analysis between the resource reservation algorithms as FCFS, TARR, and the proposed DRR. There is reduction in the AWT. The hit ratio is also optimum, as the requested resources are provided. The resource idle time is also reduced.

Figure 3 shows the graphical representation for the Average Waiting Time of the FCFS, TARR and the proposed DRR. It is evident from the chart that the average waiting time of DRR is much lower when compared with FCFS and TARR. The makespan is reduced in DRR. Thus by applying the DRR approach the makespan has been reduced.

Table 5. Comparison between the FCFS and TARR

Scenario	AWT			HR			TRIT			Makespan		
	FCFS	TARR	DRR	FCFS	TARR	DRR	FCFS	TARR	DRR	FCFS	TARR	DRR
Scenario 1	1.3	0.6	0.3	9:10	1:1	1:1	13	0	0	42	39	32
Scenario 2	1	1	0.3	9:10	1:1	1:1	7	0	0	36	36	29
Scenario 3	2.3	1.2	1.2	1:1	1:1	1:1	6	0	0	54	43	43
Scenario 4	1.2	0.7	.5	1:1	1:1	1:1	3	0	0	38	33	31

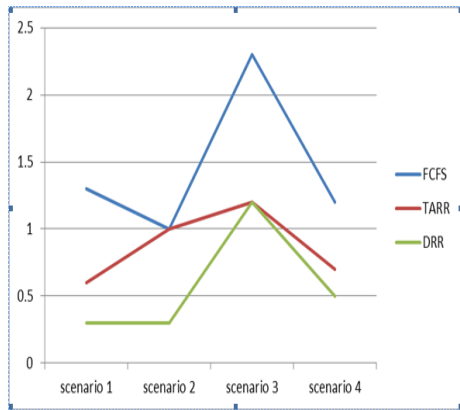


Figure 3: AWT of FCFS, TARR, DRR

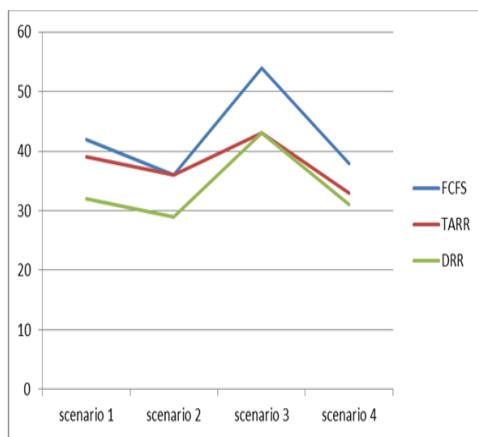


Figure 4: Chart Showing the Makespan value.

6. CONCLUSION

Resource reservation provides a way through which the resources are made readily available for the entire execution of the process. This reduces the waiting time which in turn reduces the execution time. Various parameters such as average waiting time, hit ratio, resource idle time and makespan are considered for evaluation. And this proposed algorithm provides better result when compared with other algorithms as FCFS, Time Slice based Advance Resource reservation.

7. REFERENCES

- [1] S.Nirmala Devi, A.Pethalakshmi, "Resource Discovery for Grid Computing Environment using Ant Colony Optimization by applying routing information and LRU policy", Global Trends in Computing and Communication Systems Volume 269 of the series Communications in computer and information science pp. 124 – 133.
- [2] Srikumar Venugopal, Xingchen Chu and Rajkumar Buyya, "A Negotiation Mechanism for Advance

Resource Reservation Using the Alternate Offers Protocol". In the proceedings of 2008 16th IEEE International Workshop on Quality of Service, Netherlands, 40 – 49.

- [3] Rui Min, Muthucumaru Maheswaran, "Scheduling Advance Reservations with Priorities in Grid Computing Systems". In proceedings of 2002 2nd IEEE / ACM International Symposium on Cluster Computing and the Grid, 266 – 268.
- [4] Anthony Sulistio and Rajkumar Buyya, "A Grid Simulation Infrastructure supporting Advance Reservation", proceedings of the 16th International Conference on parallel and Distributed Computing and Systems, Nov 9 – 11 2004, MIT, Cambridge, USA, pp. 1-7.
- [5] Eliza Gomes, M.A.R.Dantas, "Towards a resource reservation approach for an Opportunistic Computing Environment", Journal of Physics: Conference Series 540 (2014) 012002.
- [6] Nirmala Devi S, Dr. A. Pethalakshmi, "TARR: Time-Slice based Advance Resource Reservation in Grid Computing Environments", International Journal of Computational Intelligence and Informatics, June 2016.
- [7] Nirmala Devi S, Dr.A.Pethalakshmi "ORR: Optimal Resource Reservation in Grid computing Environments", Indian Journal of Science and Technology, Nov 2016.
- [8] Barzegar B, Esmacelzadeh H, Shirgahi H, "A New Method on Resource Management in Grid Computing Systems Based on QoS and Semantics", Indian Journal of Science and Technology. 2011 Nov, 4(11), pp. 1-4.
- [9] Sara Rezaei, Ahmad Khademzadeh and Mansour Sheikhan, "Resource Reservation in Grid Networks based on Irregular Cellular Learning Automata", International Journal of Information and Communication Technology Research, Volume 7 – Number 3 – Summer 2015 (53 -61).
- [10] Mohamed Abouelela, Mohamed EI-Darieby, "Scheduling big data application with advance resource reservation framework in optical grids", Journal of Applied Soft Computing, Elsevier publications, Volume 38, January 2016, pages 1049 – 1059.
- [11] Xiao, Peng, Peixin Qu, and Xilong Qu. "An Adaptive Redundant Reservation Strategy in Distributed High-performance Computing Environments", International Journal of Hybrid Information Technology 6.6 (2013): 51-64.
- [12] J. Chen, B. Li, E. F. Wang, "Parallel Scheduling Algorithms Investigation of Support Strict Resource Reservation from Grid", Applied Mechanics and Materials, Vols. 519-520, pp. 108-113, 2014.