A Novel Heuristic for Scheduling of Independent jobs on Grid Resources

Jyoti Bansal[1], Dr. Shaveta Rani[2], Dr. Paramjit Singh[3]
[1]Research Scholar , PTU, Kapurthala
erjyoti.2009@rediffmail.com
[2] GZSPTU Campus, Bathinda , Punjab Technical University
garg_shavy@yahoo.com
[3] GZSPTU Campus, Bathinda , Punjab Technical University
param2009@yahoo.com

Abstract— This Paper proposes a novel batch mode scheduling heuristic for scheduling of independent jobs on heterogeneous Grid resources. Proposed heuristic works in two stages, in the first stage Min-Min heuristic is being followed for the assignment of jobs to the resources & in the second stage jobs are being transferred from heavily loaded resource to the lightly loaded resource to achieve load balancing. The proposed heuristic and various existing heuristics were tested and compared within a grid simulator called GridSim[20] The experimental results reveal that the new heuristic performs better in throughput and flowtime value and also provide best resource utilization.

Keywords — Desktop Grid, Min-Min, Max-Min, Throughput, Flowtime.

I. INTRODUCTION

Due to the widespread growth of the Internet as well as the vast development of high speed processing devices and network technologies, leads to the possibility of using distributed computers as a single, unified computing resource, popularly known as Grid computing[1][2]

On the other hand, Desktop Grid aims to harvest a number of idle desktop computers owned by individuals on the edge of Internet [3].As in desktop grid faults may occur, and due to this execution time of the applications increases multifold. Due to these circumstances, there is a need of devising scheduling strategies that attempt to maximize application performance in face of faults occurrence. Various scheduling strategies can be categorized into two parts i.e., Knowledge-based [4][5] and knowledge-free[6] scheduling. Further these parts are subdivided into two subparts, on-line and batch-mode scheduling [7].

In the on-line mode, on arrival of a task at the scheduler it is immediately assigned to a machine & the decision remains unchanged once the computation for the same is done. On the contrary batch mode tasks are collected into a set as they arrive & are subsequently examined at prescheduled times for mapping called mapping events [7,8,9,19,20].

This paper is organized as follows: Section 2 presents the related work. Section 3 presents the various Heuristics for Grid Scheduling. In section 4, the new scheduling heuristic is proposed. Finally; section 5 concludes the paper.

II. RELATED WORK

Various heuristics [10,11,12,13,14,15,16,17] have been proposed in the literature for grid environments to solve the problem of allocating a set of tasks to a set of resources(scheduling) which are further categorized into: on-line scheduling and batch-mode scheduling.

Some instances of these categories are as follows:

In[11 & 12] Amid Khatibi et al. & Yun-Han Lee et al. Illustrated the comparison between the various online (i.e., OLB and MCT) and batch mode (i.e., Min-Min, Max-Min, LFFR-SJFR, Sufferage, and maxstd) algorithms.In [13] M. Macheswaran et al. found that on line scheduling heuristic is particularly suitable when the arrival rate is low. Consequently when the arrival rate is high & resource requirement information is used for scheduling of all tasks in the set then batch mode scheduling heuristic can get higher performance because there will be a sufficient number of tasks to keep hosts busy between the mapping events.Saeed Parsa et al.[14] proposed a new task scheduling algorithm called RASA (Resource Aware Scheduling Algorithm). In this algorithm, Min-Min & Max-Min are used alternatively to achieve better makespan. As if the number of available resources is even, then Max-min strategy is applied to assign the first task, otherwise the Min-min strategy is applied. Then either of the two strategies is used alternatively for assigning the remaining tasks to their appropriate resources. In[15] Yong Wang et al. presented a QoS based job scheduling heuristic that is able to deal with matching application requirements with available resources. Author has embedded QoS constraint in batch mode scheduling algorithm i.e., Min-Min algorithm to propose new batch mode Min-Min heuristic. Anand K Chaturvedi et al.[16] proposed a new batch mode heuristic method for scheduling of independent tasks

on grid resources comprising of two stages. While in the first stage, MET method is being used for the assignment of tasks on Resources & no consideration for the availability or load imbalance on resources in the grid is done. Then in the second stage, migration of tasks is done from heavily loaded resource to lightly loaded resource to remove the load imbalance. In[17] K.Hemant K. Reddy et al. proposed an on-line load balancing heuristic for desktop grids.Three parameters, namely CPU load, heap memory and CPU Idle time and their combination have been employed for load balancing. In [18] Zhang Jinquan et al. proposed two new heuristics for balancing the resource load i.e., WMTG-min and WMTSG-min and compare them with two batch mode heuristics i.e., Min-min and Max-Min.

III. HEURISTICS FOR GRID COMPUTING

There are several heuristics for grid scheduling. This section describes two heuristics, which are used for comparison purpose in this study.

3.1 Min-Min: In Min-Min algorithm a set of all unassigned tasks is created comprising of two phases. During first phase, tasks having minimum expected completion time on the corresponding machine are selected. In second phase, the task with the overall minimum expected completion time gets assigned to the corresponding resource. After the task which has been allocated, it will get removed from the set and the process is repeated until all jobs in the set are mapped [19].This heuristic will take O(n2m) time for n no of jobs & m no of resources.

3.2. Max-min: Max-Min is very similar to Min-Min except in phase two, where it assigns the task to the resource having maximum expected completion time [19]. This heuristic will also tae O(n2m) time for n no of jobs & m no of resources.

These heuristics estimate the Expected Execution time(Eab) & Completion time (Cb) of each of the job(Jb) on each of the Grid resources(Ra) & the two objectives throughput & flowtime which are also used in this paper are defined as:

3.3. Expected Execution Time (Eab): It can be calculated as:

$$E_{ab} = \text{Size of Job J}_b / \text{Speed of Resource R}_a$$

3.4. Completion time (Cab): It can be Computed by adding the Expected Execution Time(Eab) of Job Jb on Resources Ra & Beginning of execution of Job Ji.

$$C_{ab} = E_{ab} + J_i$$

3.5. Throughput (T): The Throughput is a metric used to measure the Output of Grid System. So the Throughput (T) of the System can be defined as per the following formula w.r.t. the Completion time (Cab) of Job (Jb).

$$\text{Throughput (T)} = \text{Max}(C_{ab}), b = 1..n$$

3.6. Flowtime: It can be defined as the sum of completion times of jobs & measures the quality of Grid system. So the Flowtime (F) of the Grid System can be defined as per the following formula w.r.t. the Completion time (Cab) of job (Jb).

$$\text{Flowtime(F)} = \sum C_{ab}, b = 1..n$$

There is a simple example of job scheduling where 6 jobs are taken to be computed on 3 resources.

Suppose that m resources Ra(a= 1..m) have to process n jobs Jb(b=1..n).

The Expected Execution Time Eab of job Jb w.r.t to size of job on Resource Rb & the speed of Resource are listed in Table 1.

TABLE 1. Expected Execution time of 6 Jobs on 3 Resources

Job(size) in MI	Resource(speed) in MIPS		
	R1(100)	R2(200)	R3(300)
J1(1000)	10	5	3.3333333333333335
J2(2000)	20	10	6.666666666666667
J3(3000)	30	15	10
J4(4000)	40	20	13.333333333333334
J5(5000)	50	25	16.666666666666668
J6(22000)	220	110	73.33333333333333

The result on applying the Min-Min & Max-Min heuristic of 6 jobs computed on 3 resources w.r.t. to the expected execution time listed in Table 1, Table 2 & Table 3 respectively and the Completion Time Cab of job Jb on Resource Ra on applying the Min-Min & Max-Min heuristics is listed in Table 4 & Table 5.

TABLE 2. The Result of Min-Min Heuristic

Resource	Speed	Jobs Assigned
R1	100	-
R2	200	J2,J5
R3	300	J1,J3,J4,J6

TABLE 3. Result of Max-Min Heuristic

Resource	Speed	Jobs Assigned
R1	100	J1,J4
R2	200	J2,J3,J5
R3	300	J6

In Min-Min heuristic jobs J1,J2,J3,J4,J5,J6 are assigned to resources R3,R2,R3,R3,R2,R3 resp. & the throughput & flowtime of Grid system on applying Min-Min is 103 & 199 respectively as per the completion time illustrated in Table 4.

TABLE 4. Completion time of Job Jb of Min-Min Heuristic

Tasks	J1	J2	J3	J4	J5	J6
Completion time(Cab)	4	11	15	29	37	103

TABLE 5. Completion time of job Jb of Max-Min Heuristic

Tasks	J6	J5	J4	J3	J2	J1
Completion time(C_{ab})	74	26	41	42	53	52

In Max-Min the jobs J6,J5,J4,J3,J2,J1 are assigned to resources R3,R2,R1,R2,R2,R1 respectively & the throughput & flowtime of Grid system on applying Max-Min is 74 & 288 respectively as per the completion time illustrated in Table 5.

IV. PROPOSED HEURISTIC

The Max-Min heuristic seems to do better than Min-Min in terms of Throughput value but the Flow time of the system will get increased when the larger jobs are assigned to fastest resources first & the shorter ones are waiting for larger to be executed or assigned to comparatively slower resources.

To resolve the difficulties of Min-Min & Max-Min, this paper proposes a novel job scheduling heuristic which works in two stages. In the first stage Min-Min is being followed for the assignment of jobs to the resources. In the second stage jobs are transferred from heavily loaded resource to the lightly loaded Resource to remove the load imbalance.

Following is the Pseudocode for the flow of working of the proposed heuristic.

1. For all jobs J_b in B_v
2. For all resources R_a
3. $C_{ab}= E_{ab}+J_i$
4. Do until all jobs in B_v are mapped
5. Identify the Resources that give the earlier Completion time for each job(J_b)
6. Identify the job J_b with minimum earliest Completion time
7. Assign the job(J_b) to the corresponding Resource(R_a)
8. Delete J_b from B_v
9. Update the values of R_a & C_{ab}
10. Find the Heavily loaded Resource R_{max} & Lightly loaded Resource R_{min}
11. Find job J_{min} having minimum execution time on maximally loaded Resource R_{max}
12. Transfer the job Jmin having minimum execution time on maximally loaded Resource R_{max} to the minimally loaded Resource R_{min}.

V. EXPERIMENTAL SETUP

The proposed heuristic is compared with other basic heuristics Min-Min & Max-Min for performance comparison by using Simulation environment known as GridSim Toolkit[20].The testing is performed by taking two scenarios:-

Scenario 1: Max no of jobs taken are small in size. Following example explains the concepts as illustrated in the proposed technique. Considering the Scenario as illustrated in table 1 when 6 numbers of jobs have to be allocated to 3 resources. In the first stage Min-Min is being followed for the assignment of jobs to the resources. The result of Stage 1 on applying the Min-Min heuristic is same as illustrated in Table 2.In the second stage, on applying the load balancing to stage 1 for the same set of 6 jobs computed on 3 resources w.r.t. to the expected execution time listed in table 1, the final result of proposed strategy is illustrated in Table 6 and the Completion Time Cb of job Jb on Resource Ra is listed in Table 7.

TABLE 6. The Result of Proposed Heuristic.

Resource	Speed	Jobs Assigned
R1	100	J1
R2	200	J2,J5
R3	300	J3,J4,J6

TABLE 7. Completion time of job J_b of proposed Heuristic.

Tasks	J1	J2	J3	J4	J5	J6
Completion time(C_{ab})	11	11	11	25	37	99

In proposed Heuristic the tasks J1,J2,J3,J4,J5,J6 are assigned to resources R1,R2,R3,R3,R2,R3 respectively & the throughput & flowtime of Grid system on applying proposed Heuristic is 99 & 194 as per the completion time illustrated in Table 7.

Scenario 2: Considering the scenario 2 when Max no of jobs taken are large in size, following example as illustrated in Table 8 calculates the Expected Execution Time Eab of job Jb w.r.t to size of job on Resource Rb & the speed of Resource.

TABLE 8. Expected Execution time of 6 Jobs on 3 Resources

Job(size)in MI	Resource(speed) in MIPS		
	R1(100)	R2(200)	R3(300)
J1(1000)	10	5	3.3333333333333335
J2(2000)	20	10	6.666666666666667
J3(30000)	300	150	100
J4(40000)	400	200	133.33333333333334
J5(50000)	500	250	166.66666666666666
J6(22000)	220	110	73.33333333333333

The result in terms of Throughput, Flowtime on applying the Min-Min, Max-Min(Existing) & Proposed strategy w.r.t. the estimated execution time illustrated in Table 8 is listed in Table 9 and the Completion Time Cb of job Jb on Resource Ra is listed in Table 10.

TABLE 9. Result comparison of Throughput & Flowtime values

Parameter	Min-Min	Max-Min	Proposed
Throughput	379	268	375
Flowtime	999	1287	993

TABLE 10. Result comparison of Completion Time of Jobs.

Heuristic	Completion Time(Jobs)					
	J1	J2	J3	J4	J5	J6
Min-Min	4	11	162	138	305	379
Max-Min	218	212	268	201	167	221
Proposed	10	11	162	134	301	375

As per the results obtained it is concluded that the throughput of the system on applying the proposed heuristic is 99, 375 (as per Scenario1&2) & flow time is 194, 993 (as per Scenario1&2), whilst Min-Min & Max-Min provide a scheduling with a throughput of 103, 379(as per Scenario1&2) & 74,268 (as per Scenario1&2) & flowtime of 199,999 (as per Scenario1&2) & 288, 1287(as per Scenario1&2) respectively as listed in figure 1,2 & figure 3,4 . It has also been observed that Completion time(Cb) also improves significantly on applying the proposed heuristic in comparison with Min-Min & Max-Min as illustrated in figure 5 & figure 6.

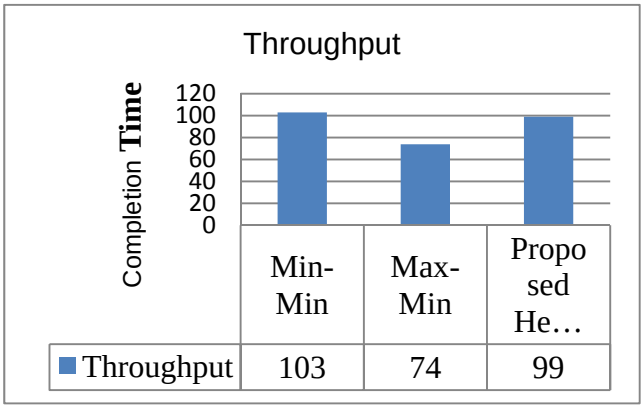


Fig 1: Comparison Analysis of throughput values as per Scenario1.

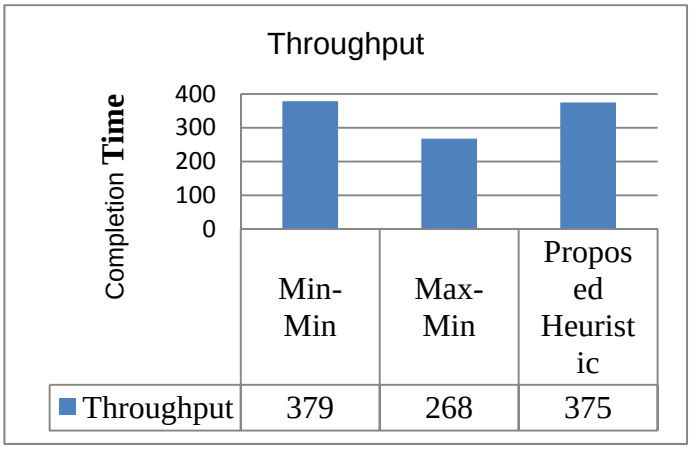


Fig 2: Comparison Analysis of throughput values as per Scenario2.

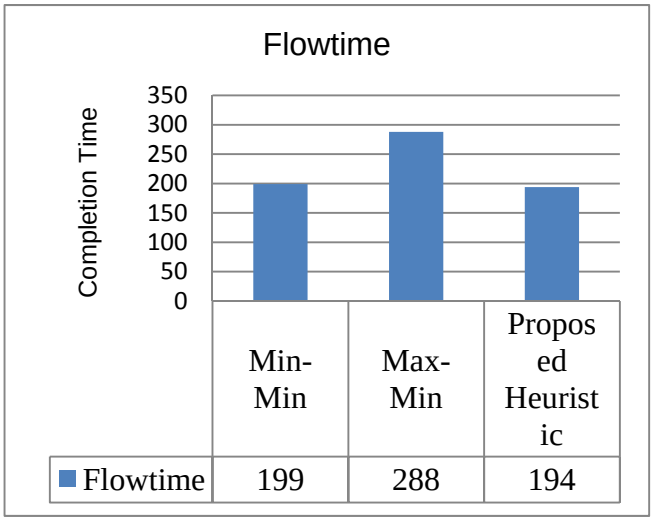


Fig 3: Comparison Analysis of Flow time values as per Scenario1.

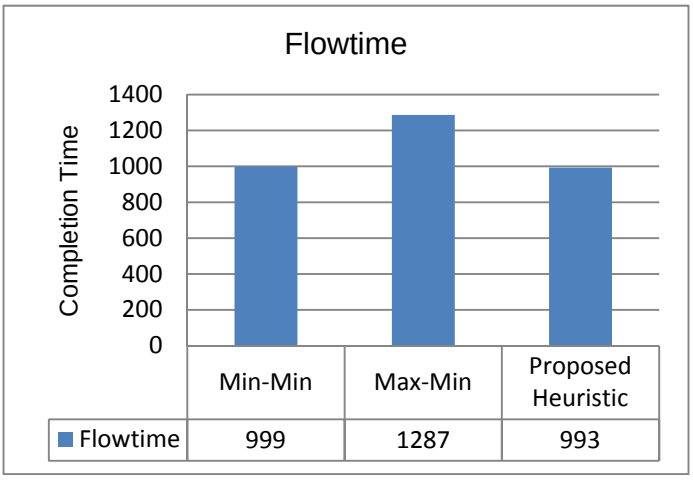
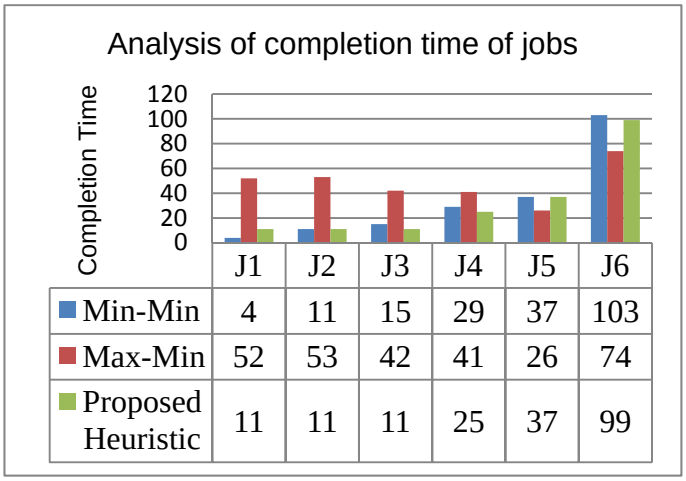
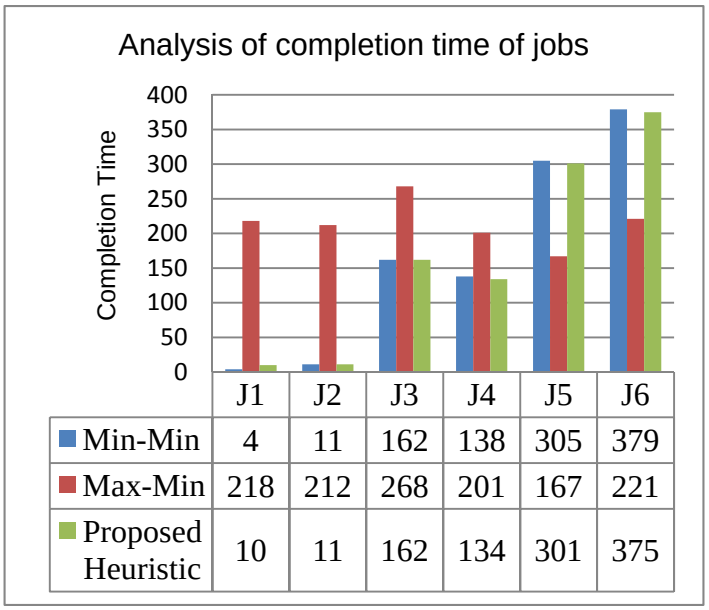


Fig 4: Comparison Analysis of Flow time values as per Scenario 2.



	J1	J2	J3	J4	J5	J6
Min-Min	4	11	15	29	37	103
Max-Min	52	53	42	41	26	74
Proposed Heuristic	11	11	11	25	37	99

Fig 5: Comparison Analysis of Completion time as per Scenario1.



	J1	J2	J3	J4	J5	J6
Min-Min	4	11	162	138	305	379
Max-Min	218	212	268	201	167	221
Proposed Heuristic	10	11	162	134	301	375

Figure 6: Comparison Analysis of Completion time as per Scenario2.

VI. CONCLUSION

In this paper a novel batch mode job scheduling heuristic has been proposed. The contribution of this paper is that load balancing has been applied along with Min-Min heuristic for improving the overall throughput & Quality of service of the Grid system. The experimental results show that the proposed heuristic gives better results in terms of Throughput & Flowtime parameter values & also provides better resource utilization.

REFERENCES

[1] F. Berman, G. C. Fox, et al."Grid Computing: Making the Global Infrastructure a Reality", Chapter 1,2,3,6,8, Wiley, 2003.
[2] M. Baker, R. Buyya, et al."Grids and Grid Technologies for wide-area distributed computing",Software: Practice and Experience (SPE), Vol. 32, Issue 15, pp. 1437-1466, Dec. 2002.
[3] D. Kondo, M. Taufer, et al."Characterizing and Evaluating Desktop Grids: An Empirical Study", Proceedings of International Parallel and Distributed Processing Symposium 2004 (IPDPS 2004), IEEE CS Press, pp. 26-35 Apr. 2004.
[4] F. Berman, R. Wolski, et al. "Adaptive computing on the grid using apples", IEEE Trans. on Parallel and Distributed Systems, vol. 14, no. 4, 2004.
[5] H. Casanova, A. Legrand, et al. "Heuristics for scheduling parameter sweep applications in grid environments", Proceedings of the 9th Heterogeneous Computing Workshop (HCW 2000), Cancun, Mexico. IEEE CS Press, USA, 2000.
[6] D. Paranhos, W. Cirne, et al. "Trading cycles for information: Using replication to schedule bag-of-tasks applications on computational grids", Proceedings of the Euro-Par 2003: Lecture Notes in Computer Science Volume 2790, pp. 169-180, 2003.
[7] T.D. Braun, H.J. Siegel, et al. "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems", Proceedings of Journal of Parallel and Distributed Computing 61, pp 810-837, 2001.
[8] Cosimo Anglano,Massimo Canonico, "Fault-Tolerant Scheduling for Bag-of-Tasks Grid Applications", Proceedings of Springer-Verlag Berlin Heidelberg in P.M.A. Sloot et al. (Eds.): EGC 2005, LNCS 3470, pp. 630–639, 2005.
[9] R. Wolski, N. T. Spring, et al. "Predicting the CPU availability of time-shared unix systems on the computational grid", Cluster Computing, vol. 3, no. 4, pp. 293–301, 2000. [Online]. Available:citeseer.ist.psu.edu/wolski98predicting.html.
[10] F. Dong , S. G. Akl, "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems", Technical Report of the Open Issues in Grid Scheduling Workshop, School of Computing, University Kingston, Ontario,vol. 504, January 2006.

[11] Amid Khatibi Bardsiri, Seyyed Mohsen Hashemi, "A Comparative Study on Seven Static Mapping Heuristics for Grid Scheduling Problem", International Journal of Software Engineering and Its Applications, Vol. 6, No. 4, pp. 247-256, 2012.
[12] Yun-Han Lee, Seiven Leu, et al. "Improving job scheduling algorithms in a grid environment", In proceedings of Future Generation Computer Systems 27, pp. 991-998, 2011.
[13] M. Macheswaran, S. Ali, et al."Dynamic mapping of a class of independent tasks onto heterogeneous computing systems", J. Parallel Distribut. Comput. 59,pp. 107–131,1999.
[14] Saeed Parsa,Reza Entezari-Maleki,"RASA: A New Task Scheduling Algorithm in Grid Environment", Proceeding of World Appl. Sci. J., 7 (Special Issue of Computer & IT): pp. 152-160, 2009.
[15] Yong Wang, Shougeng Hu, et al. "A Strategy of Resource Scheduling for Grid Computing Based On QoS", In proceeding of Information Science and Engineering (ICISE), 1st International Conference, pp. 82-85, December 2009.
[16] Anand K Chaturvedi, Rajendra Sahu, "New Heuristic for Scheduling of Independent Tasks in Computational Grid", International Journal of Grid and Distributed Computing Vol. 4, No. 3, September, 2011.
[17] K.Hemant, K. Reddy, et al. "An adaptive scheduling mechanism for computational desktop grid using gridgain." Proceeding in Procedia Technology 4, pp. 573-578, 2012.
[18] Z. Jinquan, Ni Lina, et al. "A Heuristic Scheduling Strategy for Independent Tasks on Grid",In proceedings of High-Performance Computing in Asia-Pacific Region, 2005. Proceedings. Eighth International Conference in IEEE, pp. 6-pp, 2005.
[19] F. Xhafa, L. Barolli, et al. "Batch Mode Schedulers for Grid Systems", International Journal of Web and Grid Services ,Vol. 3, No. 1, pp 19-37, 2007.
[20] R. Buyya, M. Murshed, "GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," Journal of Concurrency and Computation: Practice and Experience, pp. 1175–1220, 2002.