ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

An Efficient Priority Based Task Management In Grid Computing-2

Bhawna(student)[#1], Asst.prof. Kompal Ahuja[*2]

Delhi Institute Of Technology And Management / DCRUST ,

Sonepat, Gannaur

Abstract - Grid computing can integrate and utilize heterogeneous computing resources which are connected through networks without the limitation of geography.In this research paper we have proposed a priority based task scheduling algorithm that can handle heterogeneous work flow. The algorithm will represent workflow in DAG then assign the priority to each level DAG. The proposed algorithm will be implemented in java.

Keywords- Grid computing, resource scheduling, DAG , Fault tolerance , Task management

I. INTRODUCTION

A. Grid Computing

The term Grid is increasingly appearing in computer literature, generally referring to some form of system framework into which hardware or software components can be plugged, and which permits easy configuration and creation of new functionality from existing components. Grids enables the sharing, selection, and aggregation of a wide variety of resources including supercomputers, storage systems, data sources, and specialized devices that are geographically distributed and owned by different organizations for solving large-scale computational and data intensive problems in science, engineering, and commerce and also in many fields.

II. GRID CHARACTERISTICS

- *Heterogeneity*: A grid hosts both software and hardware resources that can be very varied ranging from data, files, software components or programs to sensors, scientific instruments, display devices, personal digital organizers, computers, super-computers and network

- *Geographical Distribution*: Grid's resources may be located at distant places. The main advantage of Grid computing in terms of its characteristics is that we can connect different resources which are a far distinct placed from each other.

- *Resource Sharing*: Resources in a grid belong to many different organizations that allow other organizations (i.e. users) to access them. Non local resources can thus be used by applications, promoting efficiency and reducing costs. By sharing the resources we save the idle time of the resources by providing jobs automatically to the resource which are kept free from a longer time. Load balancing let the resources to be shared .Sharing of resources provide a better way to provide the good synchronization between the different resources so that the idle time of Processor should be saved.

- *Multiple Administrative Domains*: Each organization may establish different security and administrative policies under which their owned resources can be accessed and used. As a result, the already challenging network security problem is complicated even more with the need of taking into account all different policies.

- *Consistent Access*: A grid must be built with standard services, protocols and inter-faces thus hiding the heterogeneity of the resources while allowing its

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

scalability. Without such standards, application development and pervasive use would not Be possible

III. PROBLEM AREAS IN GRID COMPUTING

Security: Since the number of users within a system is increased, new security mechanisms are needed to ensure that malicious code cannot legitimate services running on the grid.

Fault Tolerance: In a heterogeneous system like the Grid, failure is inevitable. Failures of system resources have adverse effects on applications performance. Failures can make a process run slower than normal or even stop it.

Resource Scheduling: A grid federates a large number of resources contributed by individual machines into a greater total virtual resource. For applications that are grid-enabled, the grid can offer a resource balancing effect by scheduling grid jobs on machines with low utilization. This feature can prove invaluable for handling occasional peak loads of activity in parts of a larger organization.

Load Balancing: Load Balancing is crucial to computational grids. It is a mapping strategy that efficiently equilibrates the task load into multiple computational resources in the network based on the system status to improve performance.

Resource Discovery: Grid environment is an environment system in which applications are composed of the large set of hardware and software resources distributed among many locations. Hence, the problem of resource discovery can be profoundly complex due to the size and the complexity of s

IV. RESOURCE SCHEDULING

Grid is a large, dynamic, heterogeneous and collaborative environment. Grids integrate networking, communication, computation and information to provide a virtual platform for computing and data management.

V. GRID WORKFLOW MANAGEMENT

Grid workflows are an emerging research field in the Grid community. Actually there is an ongoing effort to define a standard meaning of workflow for the Grid.
A workflow has three dimensions:

1) The process: The process dimension refers to the creation and the possible modification of the process description.

2) The case: The case dimension refers to a particular instance of the workflow when the attributes required by the process enactment are bound to specific values.

3)The resource: The resource dimension refers to discovery and allocation of resources needed for the enactment of a case.

Workflow enactment is the process of carrying out the activities prescribed by the
process description for a particular case. At least four important issues can be identified in order to enable workflows for the Grid:

- User Environments or Workflow IDE (Integrated Development Environment).
- Representation and language express workflow.

VI. OBJECTIVES

1) Workflow management.

2) Workflow representation through DAG.

3) Priority based task scheduling in workflow.

4) The main advantage is that it eliminates the problem of assigning priority for workflow. The main objective is to propose a priority based task scheduling algorithm that can handle heterogeneous work flow.

5)The proposed algorithm will represent workflow in DAG then assign the priority to each level DAG.

6) The proposed algorithm will be implemented in java.

VI. PROPOSED WORK

A. Generic grid architecture

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

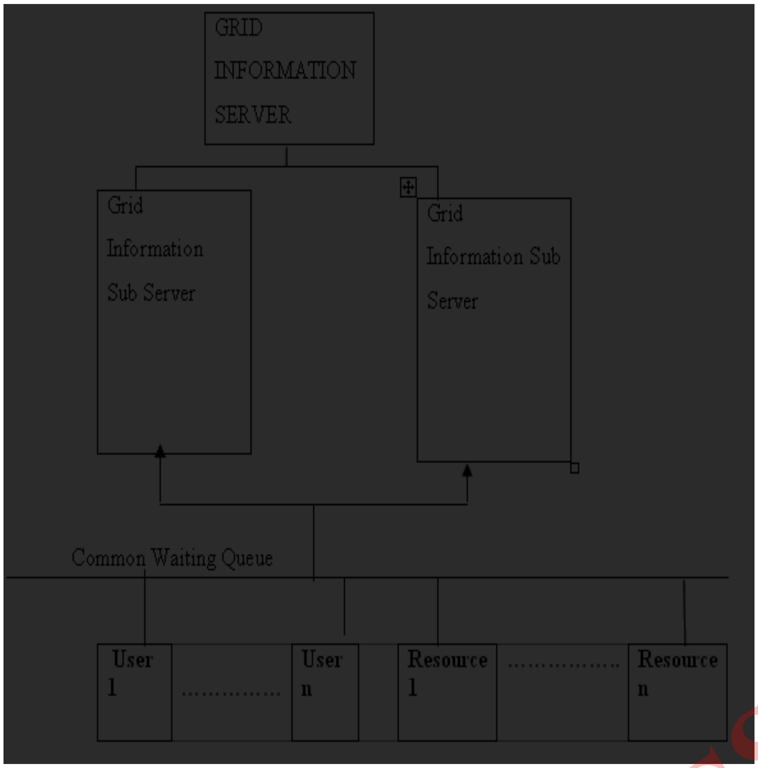


Fig.1 computational Grid model

VIII. PROPOSED ALGORITHM:

The proposed algorithm for adaptive load balancing in computational grid is as follows:

1) Number of tasks: v and the computation cost matrix of the DAG: T (v×v)

2) Amount of data to be transferred between the tasks: D (v×v)

3) Number of processors in the systems: p

4) Rate of data transfer between the processors: R (p×p)

Algorithm :

1. begin
2. read the DAG, associated attributes values, and the number of processor P;
3. level sort the given DAG;
4. for all task vk in the DAG do
5. begin
6. compute ULC, DLC and ACC values for the task vk ;
7. compute LC(vk) = max{LC(vj)} + ULC(vk) + DLC(vk), where vjÎpred(vk);
8. insert the task into the priority queue based on the LC value such that the tasks in lower level are placed in the priority queue first than the tasks in the higher level and tie if any, is broken using the ACC value;
9. end;
10. end;
11. end;
12. select the highest priority task vj from the queue for scheduling;
13. for each processor pk in the processor set P
14. begin
15. compute EFT (vj, pk) using isetion-based scheduling policy;
16. compute TEC(vj, pk);
17. if (TEC(vj, pk) < B(pk)) then
18. begin
19. compute cost = l (EFT(vj, pk)) + g (TEC(vj, pk));
20. assign the task vj to the processor pk, which minimizes the cost;
21. B(pk) = B(pk) - TEC(vj, pk);
22. TECP = TECP + TEC(vj, pk);
23. end
24. else
25. select the next processor pk+1 in the processor set P;
26. end;
27. end;
28. end.S

Algo 1 : proposed algorithm for adaptive load balancing in computational grid

ISSN: 2321-9653

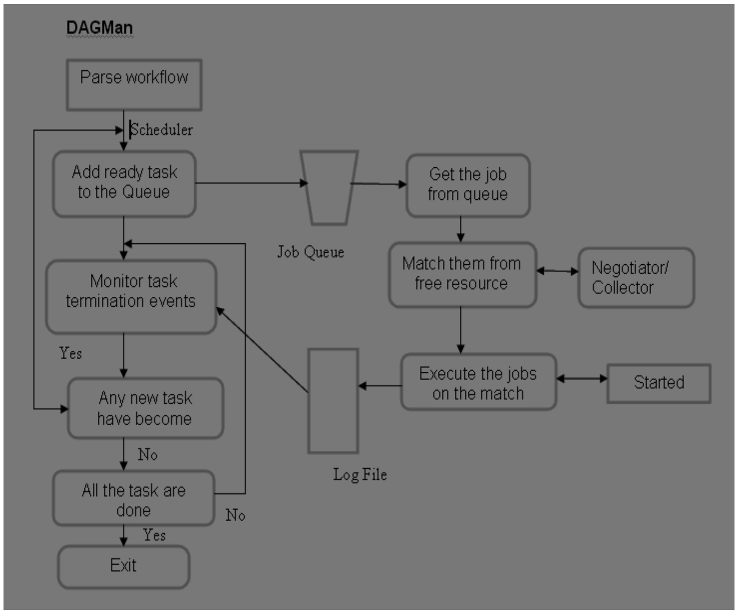


Fig.2 Sequence diagram

IX.RESULT

Following parameters are used during simulation of priority calculation algorithm:

Table 4.1: Execution Environment

Execution Environment	Java Net Beans6.7earlier

Table 4.2: Simulation Parameter

No. of Workflow	1
Workflow Division in task	10

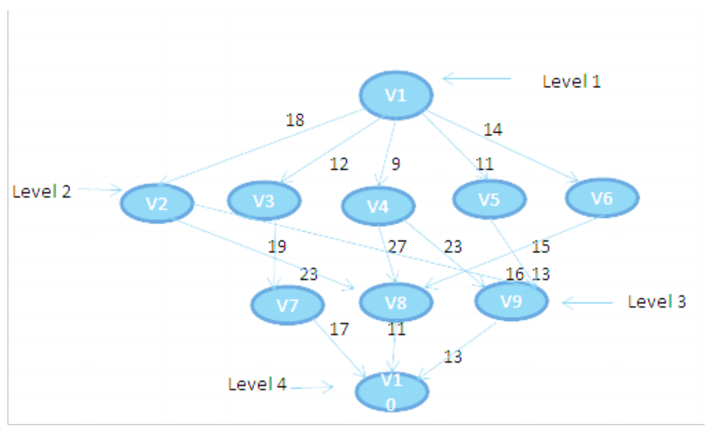
Table 4.3: Workflow Representation

V1	Entry Task
V10	Exit Task

The table 4.1 shown the execution environment is java net beans 6.7. The table 4.2 showed the simulation parameter. The table 4.3 represent the workflow. The estimated values are calculated by the algorithm and shown in table 4.4. The table 4.5 contains the task is according to its level and after apply the algorithm it will sort the task according to its priority and shown in table 4.6.

Table 4.4 Dag Representation

Vertex	Successor	Predecessor
V1	V2,V3,V4,V5,V6	None
V2	V8,V9	V1
V3	V7	V1
V4	V8	V1
V5	V9	V1
V6	V8	V1
V7	V10	V3
V8	V10	V4
V9	V10	V5
V10	None	V7,V8,V9



Workflow representation with the help of DAG

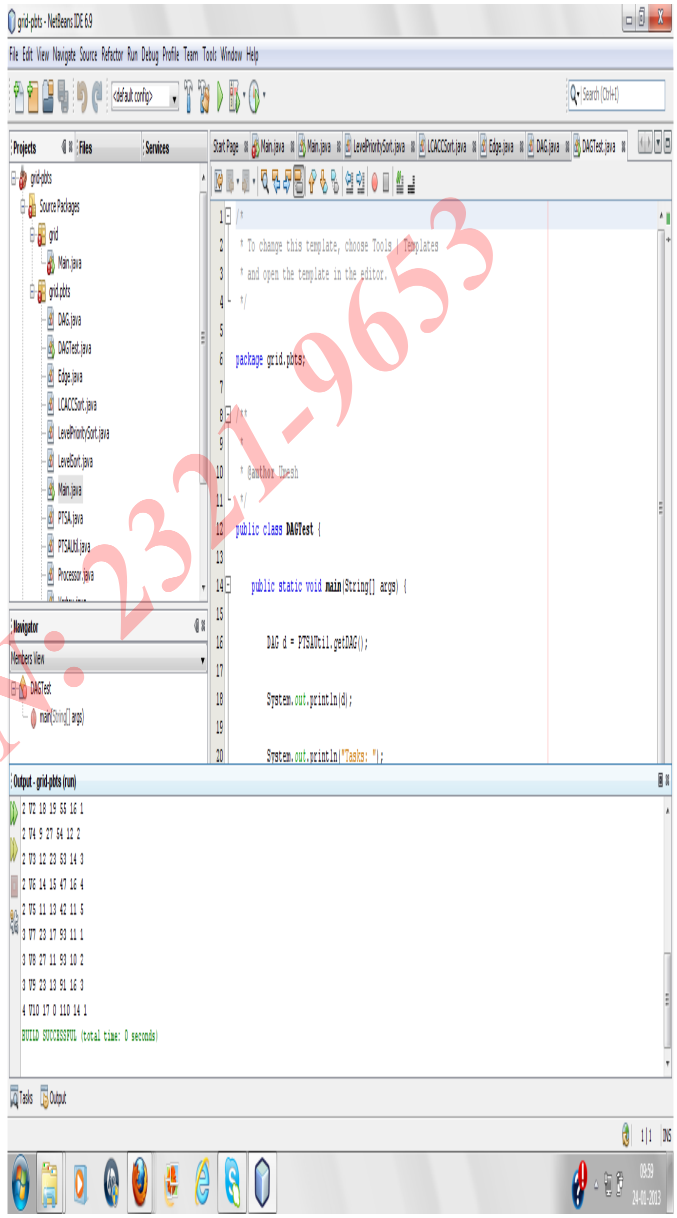
ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Level	Task	DLC	ULC	LC	ACC	PRIORITY
1	V1	0	18	18	13	1
2	V2	18	19	55	16	1
2	V3	12	23	53	14	3
2	V4	9	27	54	12	2
2	V5	11	13	42	11	5
2	V6	14	15	47	16	4
3	V7	23	17	93	11	1
3	V8	27	11	93	10	2
3	V9	23	13	91	16	3
4	V10	17	0	110	14	1

Table 4.6 Task List sorted by Level and then by Priority

Level	Task	DLC	ULC	LC	ACC	PRIORITY
1	V1	0	18	18	13	1
2	V2	18	19	55	16	1
2	V4	9	27	54	12	1
2	V3	12	23	53	14	1
2	V6	14	15	47	16	2
2	V5	11	13	42	11	2
3	V7	23	17	93	11	3
3	V8	27	11	93	10	3
3	V9	23	13	91	16	4
4	V10	17	0	110	14	5



ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

CONCLUSION

The scheduling of DAG-structured application onto the HMCS is explored in this thesis with an objective to minimize the schedule length or energy consumption or both, based on the weights given for minimizing the schedule length or energy. As the execution of an application on HMCS demands both the compile-time and runt-time support, a new two-phase task scheduling algorithm namely, HPSM has been proposed and developed, which considers either the minimization of the schedule length or the energy utilization or both, at the compile-time based on the needs and rescheduling of tasks whenever a node leaves the HMCS or the energy of the node exhausted at the run-time. The performance of the HPSM algorithm is compared with the PETSM and HEFTM, the modified versions of the PETS and HEFT algorithms respectively. Simulation experiments have been conducted to study the effectiveness of the algorithms in terms of the schedule length and energy consumption using a large set of randomly generated task graphs. The experimental results show that the HPSM algorithm significantly performs better than the PETSM and the HEFTM algorithms.

FUTURE WORK

The future work can extend to priority based distributed environment. This research work can extend priority base cloud computing. Where we have to manage large clouds and priority assignment is complicated task.

REFERENCES.

[1] Ian Foster,"What is the grid? A three point checklist", Grid Today, vol. 1, pp.6-12 2002.

[2] Ehsan Ullan Munir and Jian-Zhong Li, "Performance Analysis of Task Scheduling Heuristics in grid", Proceedings of the Sixth International Conference on Machine Learning and Cybernetics ,Hong Kong, August 2007.

[3] Ladislau L.boloni and Muthucumaru Maheswaran, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing 61, pp. 810-837, 2001.

[4] Task Scheduling for Heterogeneous Computing", IEEE Trans. on Parallel and Distributed Systems, vol. 13, No.3, Feb' 2002.

[5] M.Rahman,S.Venugopal and R.Buyya. A dynamic critical path algorithm for scheduling scientific workflow applications on global grids. Proceedings of the 3rd IEEE International Conference on e-Science and Grid Computing Bangalore,India,2007.

[6] Gizem, Aksahya & Ayese, Ozcan (2009) Coomunications & Networks, Network Books, ABC Publishers.

[7] Talukder A. K., and R. R. Yavagal, Mobile Computing: Technology, applications and servicecreation, Tata McGraw Hill, 2006.

[8] Zeng H., C. Ellis, A. Leveck and A. Vahdat. Ecosystem: Managing energy as a first class operating system resource. Proceedings of the Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2002, pp. 123-132.

[9] Weiser M., B. Welch, A. Demers and S. Shenker. Scheduling for reduced CPU energy," Proceedings of the 1st symposium on operating systems design and implementation, 1994, pp. 13-23.

[10] Mishra R., N. Rastogi, Z. Dakai, D. Mosse and R. Melhem. Energy aware scheduling for distributed real-time systems. Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03) 2003.