ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

Algorithmic scheduling of Tasks and Work flow in Grid Computing System

R.Ananthi Lakshmi[1], S.Vidhya[2]

M.Phil Scholar, Computer Science, KG College of Arts and Science, Coimbatore, India[1]

Assistant Professor, Computer Science, KG College of Arts and Science, Coimbatore, India[2]

Abstract: Grid Computing enables large-scale resource sharing by integrating the distributed resources. Scheduling of application tasks to the resources is the central to the grid computing system.We proposed the schedule work flow tasks to the dynamic grid computing resources based on rescheduling methods. Nowadays scheduling tasks on grid resources all are fully belongs to a class of NP-hard problems. The dependent tasks is fully meet the performance requirements on Quality Of Service (QOS).The Work Flow Scheduling (WFS) algorithm of grid application is consist of workflow tasks.

Keywords: Work Flow, Tasks, Scheduling, Resources, QOS.

I. INTRODUCTION

Schedulers are some types of applications that are responsible for managing the jobs, and allocating the resources needed to any specific job. Partitioning the jobs to any specific jobs. Partitioning the jobs in to schedule is the parallel execution of tasks, high level, low level management, event correlation and service level management capabilities. Once the jobs are submitted the Grid Computing schedulers are identified based on their service level requirements.Tasks are arrived randomly with the help of random computation length.These tasks can be executed any computation length.

Tasks assignment generally associated with the help of resources. In Grid Computing system local tasks have more priority. These tasks are corresponding to an overall optimal performance. Tasks assignment policies can be classified in to static task properties and dynamic task properties. A static tasks property consists of operation cycle, average operation.

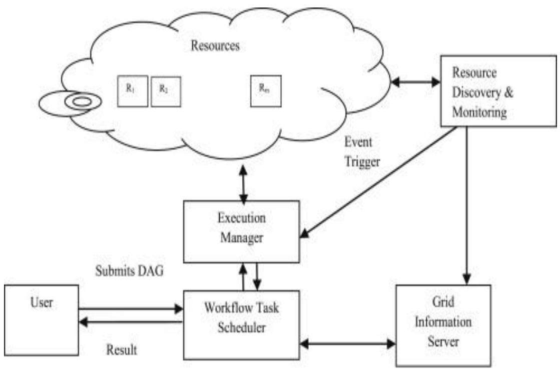
Dynamic tasks properties consist of system overloaded and other over loaded operation. Work flow[1] it is fully defines work flow activities (tasks) and controls, data dependencies. There are mainly three components in work flow .These are work flow scheduling,data movement and fault movement.Work flow is fully identified the resources and allocates the resources.

II. PROPOSED WORK

The problem in grid scheduling, work flow tasks has already been addressed in the literature. The work flow tasks assign suitable resources to achieve the high performance. It aims to reducing the communication cost in grid system. The work flow planner is to adapt the run time executor. Two types of scheduling approaches are proposed in the literature.Alogarithmic scheduling and dynamic scheduling.In dynamic scheduling ,all the tasks are scheduled at run time.In a Alogarithmic scheduling all the tasks are scheduled at the alogarithmically.

This paper introduced the Algorithmic based scheduling of tasks and work flow in grid computing system. Resources are fully involved with how the operations are controlled and services are made with the availability of other entities. Some of the resource management functions are job submission, selectionscheduling,authentication,scheduling,andjobmonitoring.Once the jobs are submitted to the resource manager[3], the resource manager takes care of the execution of the actual jobs. In grid scheduling lower level scheduling are under the control of several grid schedulars.Most complicated tasks in grid computing is the allocation of jobs and the resources. A task also includes independent and dependent tasks. The communication cost between two tasks is assigned to zero.

III. WORK FLOW SCHEDULING (WFS)



WORK FLOW SCHEDULING ARCHITECTURE

From the literature survey, we have to know the proposed work flow scheduling strategy for grid applications. It consist dependent tasks to meet the performance requirements based on QOS.The proposed Work Flow Scheduling differs from approaches in the literature by considering the dynamic availability of resources. It consists the benefits of rescheduling to availability of newly added resources. The algorithmic scheduling is helps to achieves the minimum execution time of the application. In [1],The author proposed the job scheduling algorithm to schedule the data intensive jobs.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

The figure shows the Work Flow Scheduling(WFS)Architecture[4].It consists Resource Discovery &Monitoring ,Grid Information Server,Work Flow task Scheduler,Execution Manager.Grid Information Server is used to maintain the available grid resources.Resource Discovery and Monitoring is to monitor and discovered the grid resources.Execution Manager is to rescheduling the unexecuted tasks.The Resource Discovery and Monitoring component notifies either the overload or new resources to the Execution Manager.The new schedules generated are submitted to the Execution Manager.

WORK FLOW TASK SCHEDULER

It describes fully about scheduling work flow tasks. It receives Direct Acyclic Graph (DAG) from the user. Based on fully the available information in GIS.It initiates the static scheduling phase with aim of achieving the optimal performance. For example,minimum execution time for the entire work flow and it submits to the Execution Manager.It then dispatches the ready tasks according to the execution order of the work flow tasks based on the priority.When Resource Discovery and Monitoring Component notifies either overload or new resource event to the Execution Manager.It triggers the rescheduling request for remaining unexecuted tasks to the Workflow Task Scheduler.The new schedule is then submitted to the Execution Manager.

Algorithm

Algorithm 1:Work Flow Scheduling(WFS) Algorithm

Procedure: WFS(Work Flow G(T,E) and
Resources G(R)

Input:. T is a set of tasks,E is a edges,G is a
grid network and R is a set of resources

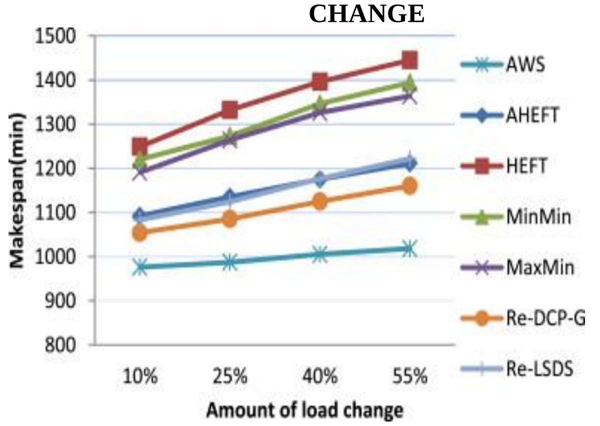
```
WFS ()
{
 Identify the tasks (T),edges(E) for each resources
   of (R).
   for each resources do
     R=WFS (G*(T*E))
     End for
}
```

ASSIGNING A PRIORITY AND TASK ORDERING

$$priority(t_i) = \begin{cases} \overline{ET(t_i)}, & \text{if } t_i \text{ is exit task} \\ \overline{ET(t_i)} + \max\limits_{t_j \in succ(t_i)} \left\{ \overline{C_{i,j}} + priority(t_j) \right\}, & \text{otherwise} \end{cases}$$

Set of parallel tasks[5] are executed predecessor tasks before the successor. To achieving this goal, we have to generate the ordered tasks for each sequence. The ordered tasks are are generated by B-level priorities. Thus the priority of tasks is defined in the equation. Where Succ(t_i) is the set of immediate successors of task t_i.ET(t_i) is the average computational tasks.The communication cost is identified in the form of C_{IJ}.The communication cost allocated to the resource known as zero.

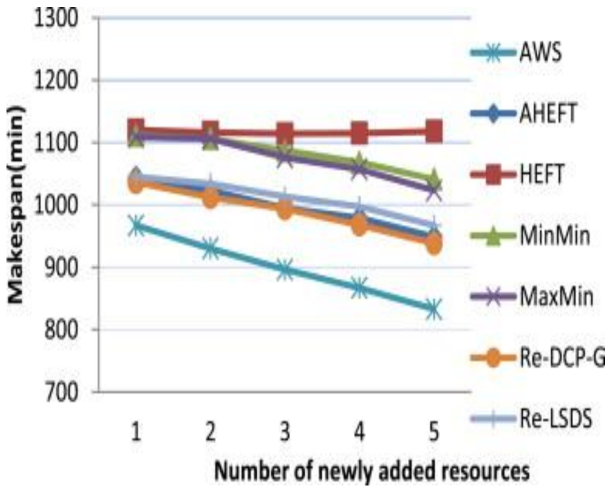
IV. THE EFFECT OF AMOUNT OF LOAD CHANGE



The figure fully illustrates the effect of amount of load change.While the heavy amount of sudden load comes to the resources. It becomes inefficient to the program.Rescheduling is the best way to this situation.In order to provide a better resources for remaining part of the applications.Thus the proposed WFS with the help of rescheduling provides a 45% better result. Over HEFT when large amount of load is dynamically change. Compared to AHERT,Min-Min,MAX-Min ,IR is to change the load.

V. EFFECT OF NEWLY ADDED RESOURCES

In this diagram we evaluated the number of newly added resources periodically to the resource pool at run time. In the figure WFS algorithm it provides better improvements, with increasing a number of newly added resources. So it provides very peak improvements to the newly added resources.



WORK FLOW MODEL

Many important grid applications in the category of work flow model.A direct Acyclic Graph(DAG) is the standard way to represent a workflow.The parallel job consists a job of tasks with constraints and it has DAG[29] and [17],G=(T,E) where

*T is the set of vertices represent n different tasks ti belongs to T,(1<i<n). That can be executed on available resources.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

*E is the set of direct edges eij=(ti,tj) belongs to E. It represents the dependencies among the tasks ti and tj indicating the task tj cannot start its execution before tj finishes and send all the required output to task tj.

The task without any predecessor is called an entity tasks.The tasks without any successor is called an exit tasks.

We assume that a DAG has one entry and exit tasks,and one entry and exit tasks is added to DAG.

VI.CONCLUSION

Our work is entirely concentrate on the identification of the best resources using algorithmic form in grid computing.The newly introduces work which helps the resources in different way.The Resources comprises with job scheduling and tasks.It keeps this as outline the proposed work will initiates more directions.In the future extension of our work will use the work flow,tasks,computation rate to predict the job schedule in a computational grid.

REFERENCES

[1] Joshy Joseph,Graig Fellenstein,"*Introduction to grid computing*",April 16,2004.Page1-3.
[2] Ahmar ABBAS "*Grid Computing A Practical Guide To Technology and Applications*",June 24,2013.
[3] A Mager et al.Workflow Expression:*In workflow in Grid Systems Work Shop* GCF-10,Berlin, March9,2004.
[4] W.M.P. van der Aalst, K.M. van Hee and G.J. Houben, "*Modelling and Analysing Workflow using a Petri-net Based Approach*", in 2nd Workshop on Computer-supported Coop-erative Work, Petri Nets Related Formalisms, pp. 31–50, 1994.
[5] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski and A.P. Barros, "*Workflow Patterns*", Technical Report, Eindhoven University of Technology, 2000.
[6] W.M.P. van der Aalst and K.M. van Hee. "*Workflow Management: Models, Methods, and Systems"*. MIT Press, Cambridge, MA, USA, 2002.
[7] J.H. Abawajy, "*Fault-Tolerant Scheduling Policy for Grid Computing Systems",* in 18th International Parallel and Distributed Processing Symposium (IPDPS'04), Santa Fe, New Mexico, IEEE Computer Society (CS), Los Alamitos, CA, USA, pp. 238–244, April 26–30, 2004.
[8] I. Altintas, A. Birnbaum, K. Baldridge, W. Sudholt, M. Miller, C. Amoreira, Y. Potier and B. Ludaescher, "*A Framework for the Design and Reuse of Grid Workflows*", in International Workshop on Scientific Applications on Grid Computing (SAG'04), LNCS 3458, Springer, Berlin, Heidelberg, New York, 2005.
[9] I. Brandic, S. Benkner, G. Engelbrecht and R. Schmidt, "*Towards Quality of Service Support for Grid Workflows, First European Grid Conference*" (EGC 2005), Amsterdam, The Netherlands, Feb 2005.
[10] R. Buyya, D., Abramson and J. Giddy, "*Nimrod/G: An Architecture of a Resource Management and Scheduling System in a Global Computational Grid*", HPC Asia 2000, Beijing, China, IEEE CS, Los Alamitos, CA, USA, pp. 283–289, May 14–17, 2000.