April 2014

An Elastic Scheduling Algorithm For Resource Co-Allocation Based on System Generated Predictions With Priority

Surendran. R

Department of Information Technology, Sathyabama University, Chennai, Tamil nadu, India,
surendran.mtech.it@gmail.com

B. Parvatha Varthini

Department of Computer Applications, St. Joseph's College of Engineering, Chennai, Tamil nadu, India,
parvathavarthini@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijcct>

Recommended Citation

R, Surendran. and Varthini, B. Parvatha (2014) "An Elastic Scheduling Algorithm For Resource Co-Allocation Based on System Generated Predictions With Priority," *International Journal of Computer and Communication Technology*: Vol. 5 : Iss. 2 , Article 1.

DOI: 10.47893/IJCCT.2014.1223

Available at: <https://www.interscience.in/ijcct/vol5/iss2/1>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

An Elastic Scheduling Algorithm For Resource Co-Allocation Based on System Generated Predictions With Priority

Surendran. R & B. Parvatha Varthini

Department of Information Technology, Sathyabama University, Chennai, Tamil nadu, India
Department of Computer Applications, St. Joseph's College of Engineering, Chennai, Tamil nadu, India
E-mail : surendran.mtech.it@gmail.com, parvathavarthini@gmail.com

Abstract - Resource Co-Allocation is basically used to execute multiple site jobs in a large scale computing environments with secure, faultless and in transparent manner. To be precise we are actually allocating multiple resources for different jobs taking into account the time parameter. Here we make use of the Scheduling queue and Resource Co-Allocation to reduce the Turn-around time with an advanced concept of System Generated Prediction based on Priority. In existing works we are scheduling the resource co-allocation request from user runtime estimation. As user runtime estimations are usually very imprecise that is not clear. In proposed work we are scheduling the resource co-allocation request based on system generated predictions through Discovery service & Priority (fairness and user experience) through topological sorting technique. The system generated predictions are better parameters than user runtime estimates for Resource co-Allocation scheduling, because System generated predictions reduce the scheduling time through proxy server based discovery service technique. The proposed work consider priorities like advanced reservation, system Generated Predictions, Negotiation, Co-scheduling, policy (SLA, Price, Trust) for resource Co-Allocation. The system generated predictions are better than user runtime estimates for Resource co-Allocation scheduling, using the experimental data's we proved this concept. End User doesn't want the grid and resource knowledge only submit job to the portal. This proposed portal will take care of all knowledge about the resource collocation automatically with fast and efficient manner.

Keywords- *Grid Computing; Resource Co-Allocation; System Generated Prediction; Scheduling; Advanced Reservation; Negotiation; Priority; policy; Co-scheduling; Prediction Algorithm; Experimental Allocation; Resource Allocation Application.*

I. INTRODUCTION

Grid technologies allow large-scale parallel computing over distributed resources which is managed by different organizations. The crucial issue for achieving high performance is Grid co-allocation of various distributed resources. Grid computing appears to be a promising trend for reasons namely: Cost effective utilization of available resources, Easy to collaborate with any organizations, Solve complex problem in quick time. Working on resource co-allocation we will face the following major problems: distributed transactions, fault tolerance, inter-site network overhead, and schedule optimization. Worldwide grid network connect the multiple resource through network wired cable in "Fig.1". The resources are available at anywhere in the world we can possible to access the all resources. Resources are normally say as Cluster, computer system, printer, scanner, webcam, lap top, server, most of hard ware devices & services. Although resource co-allocation is fundamental for Grid Computing, so far no precise solutions are found feasible on this topic. So here we describe the challenges on resource co-

allocation and in addition, the paper discusses open research issues and trends like negotiation, Co-scheduling, Priority, policy, Prediction algorithm, system generated predictions, advance reservations, and scheduling of multi-site applications.

Grid computing actually shares its resources and services in a wide-scale computing organization. Resources include Computers, Clusters, printers, scanner, webcam, Data, Storage space, services and so on. Resource Co-Allocation, Simultaneously allocates multiple resources for different applications based on the specified criteria. Scheduling queues that can be filled by scheduler both local and multi-site requests.

In System-generated prediction, consider the speed and the capacity of a resource, example CPU speed and amount of CPUs, and the load and required capacity of an activity. Herewith we must analysis the user history & resource utilization in previous time access. Advance Reservations, The user sends a booking request to each of the resource managers requesting a reservation. Negotiation, based on service level agreement negotiates

the service between provider and requester. Priority, when more than one process is waiting for run, the scheduler must decide which one to run first. Policy, based on the SLA, Price, and Trust maintain the policy from resource provider and resource requester. Co-scheduling, all of a jobs processes at the same time based on similar job group.

Fig. 1 : Worldwide Grid Network Coverage through Resources

II. PROBLEM STATEMENT

A. Existing Model

In Existing Model, scheduling the resource co-allocation request is done using user runtime estimation. This model gives inaccurate estimations provided by users for scheduling & lack of accuracy in scheduling queues.

B. Proposed Model

Here scheduling the resource co-allocation request is to be done using system generated predictions based on advance reservations cum priority. Because system generated predictions are better parameters than user runtime estimates for Resource co-Allocation scheduling. To improve the resource matching (resource search and selection), we proposed a resource matching scheme

C. Advantages of Proposed system

Quick response for Resource Co-Allocation request, High accuracy in scheduling, System generated predictions reduce the rescheduling time & number of scheduler and Low cost

III. RESOURCE ALLOCATION SYSTEM

Resources Co-Allocation are use allocate the multiple resources for different applications efficiently.

Different resources are not available in an every working place that time co-Allocation help to access the all resources from anywhere at any time. Worldwide level we are possible to connect the resources through grid network effectively

Fig. 2 : Resource co allocation from Requester to Provider

In“Fig.2” ,Authorized grid users with special grid id enter into the grid network. This system will allocate the resource automatically through GT5 (globus toolkit 5).Multiple jobs allocated to multiple resource based on co-allocation scheme.

A. Advance Reservation

The user sends a booking request to each of the resource managers requesting a reservation. The resource managers either accept or reject the booking. Time dependent co-allocation is necessary in the Grid Network system.

B. Advance Reservation by Queue Control

Initial states the reservation Queue is executed for commit or reject purpose. Each job must wait in a queue until sufficient processor resources become available to service its. Queued job & reserved job are waiting for process in the grid line separately. Based on priority jobs are process through scheduler.

C. System-generated prediction

It contains code analysis, analytic benchmarking /code profiling, and statistical prediction. Code analysis, typically limited to a specific code type or a limited class of architectures. Analytic benchmarking uses historical data to predict the job runtime, determine the performance of the machine for each code. Code profiling: It attempts to determine the composition of a task analytic benchmarking data and the code profiling data are then combined to produce an execution time estimate. Statistical prediction or execution time estimation chosen machine, the execution time is

measured & previous observations. Mainly we focus the 2 factors in this prediction process. Let as Discovery service & previous time utilization of resource.

Discovery service is a conjunctive approval reprocessing for resource co-allocation system. Discovery service automatically communicate to all process through SDP(Secondary Decision Point).It is use to collect and store the authorized all users information. In“Fig.3”, own proxy server created and use for resource allocation through discovery service. In“Fig.3”, own proxy server created and use for resource allocation through discovery service.

Fig. 3 : conjunctive approval reprocessing for resource co-allocation system

D. Scheduler or Resource Broker

It is use to access resources through resource providers, simultaneously. Meta scheduler books resources in advance based on Advance Reservation scheme. Bag of task are scheduled by local, Meta & remote scheduler.

E. Priority

Topological sorting technique use to sort the vertex in the DAG (Directed Acyclic Graph).Using this topological sorting technique we will represent the priority for the each resource requester. Find the indegree 0 vertex, then enqueue, dequeue and print(store it in the ordering) that vertex from the priority Queue. One by One all the jobs are allocated simultaneously in grid network based on topological sort. Here we are considering the grid network with resources as a Graph network with vertex and edges.

Fig. 4 : Topological sorting technique use to allocate high priority jobs first

indegree 0 vertex= Available(non allocated) resource

First we find the available resource that is in degree 0 vertex from the directed Grid worldwide network. Now again find the indegree 0 vertex with feasible price for the resource requester. Repeated do the steps finally all the jobs are allocated with priority based.

IV. ALGORITHMS

A. System Generated Predictions algorithm :

Practically all systems collect information about jobs that have run in the past that is system history or previous observation of job execution .This information can be generate the predictions about the runtimes of newly submitted jobs.

ALGORITHM 1:

PROCEDURE SYSTEM GENERATED PREDICTION ().

Step 1: Aggregate the job have executed in the past based on similarity jobs executed on grid network (user History & utilization of resource)

Step 2: Predict from technicality, usability issues, priority and Misconceptions

Step 3: Predict the code analysis, analytic benchmarking/ Code profiling, and statistical prediction.

Code analysis: typically limited to a specific code type or a limited class of architectures.

Analytic benchmarking: uses historical data to predict the job runtime. Determine the performance of the machine for each code

Code profiling: It attempts to determine the composition of a task analytic benchmarking data and the code profiling data are then combined to produce an execution time estimate.

Statistical prediction or execution time estimation

Chosen machine, the execution time is measured & previous observations.

Consider an advanced reservation based co-allocation algorithm with a aim of implementing a reservation plan that will satisfy the user requirements. The algorithm will be invoked at each reservation request entry. The steps involved are given below:

ALGORITHM 2: RESOURCE CO ALLOCATION ()

1. Proxy server basically receives a co-allocation request from the user.
2. Scheduler cum proxy server Planner will create multiple reservation plans for the obtained request.

3. Planner selects a time based resource from the available resource list. Selection based on priority, discovery service (user history), utilization of resource in previous time
4. The Planner retrieves available resource information results based on time through resource manager.
5. Using the available resource information, the Planner will determine reservation plans, based on a co-allocation

Resource provider returns co-allocation results, whether the resource co-allocation has succeeded or not, to the user. If it has failed, the user will resubmit a request with updated resource requirements.

Fig. 5 : Resource co allocation System

B. Priority Scheduling

Co-scheduling algorithm has the top priority. Because same type of task runs in the same way so run easy manner similar resources are needed for it. Co-scheduling means to scheduling all of a jobs processes at the same time based on similar job group. Topological sorting technique should be used to complete the scheduling algorithm.

ALGORITHM 3:

PROCEDURE PRORITY SCHEDULING (G,U,V)

//G-Grid network represent by graph

Step 1: Compute the indegree 0 vertex(V) [available free resource]

Step 2: Store that indegree 0 vertex from priority queue (enqueue & dequeue)

Step 3: Get the next vertex (U) from the grid network

Step 4: For all edges (U,V) update the indegree of V, and put V in the queue if the updated indegree 0.

Repeat steps 3 and 4 while the queue is not empty.

C. Scheduling algorithm

This scheduling algorithm will also consider various factors like Priority, System history, Grid Network capability, number of jobs with size, system generated prediction, Negotiation, and Co-scheduling.

V. TESTING SCENARIO AND RESULTS

Login:

If the user is existing, he can login with UserName and Password with Grid certification from grid service provider. Otherwise he cans signup by clicking NewUser button.

B. Resources List:

After successful login, the user can choose what are the resources needed, from what date & what time need. By clicking Get Availability button, the user can know the specific resources are available in the specified date. If it is available, the user has to choose the option to book and enter the comments in the command box then by clicking book button the user can book the resources.

C. Resources Allocation:

After the user booked the resources, the user has the allocation of resources table. It contains what are the resources he booked, start date and end date, status, comments he entered.

ResourceName	Start Date	End Date	Status	Comments
Cluster	15-11-2011	16-11-2011	Processing	Wait ID
Printer	22-12-2011	23-12-2011	Allocated	Allotment ID
Scanner	08-02-2012	09-02-2011	Rescheduled	Queue ID

D. Availability of Resource

The resource is not available on the user specified date. If the user likes, he can reschedule the resources.

ResourceName	AvailableDate
Printer	22-12-2011
Scanner	08-02-2012

E. Admin Process

This module is the admin process. The admin can view what are resources booked, what the dates are booked, what is the status of that.

ResourceName	Date	Status
Printer	22-12-2011	Processing
Scanner	08-02-2011	Scheduled
Node	16-03-2012	Re-Scheduled

VI. CONCLUSION

In Grid Computing, Resource allocation and Job scheduling are the main concepts to be taken care of. To manage Resource allocation or Job scheduling, we need

to keep an eye on resources and predictions. This effort aims at building a Grid system for grid resource monitoring and Prediction for Grid Task. In this paper, we present the design along with the application to allocate resources based on the System Generated Predictions based on Priority. Evaluations are performed on the provided Grid Application. This projects result has indicated that our application meets the demand of Grid System for Allocation of multiple resources to multiple clients without overlapping.

REFERENCES

- [1] K. Czajkowski, I. Foster, and C. Kesselman. Resource coallocation in computational grids. In Proc. of the 8th HPDC, pages 219–228, Redondo Beach, USA, 1999.
- [2] H. H. Mohamed and D. H. J. Epema. Experiences with the KOALA co-allocating scheduler in multiclusters. In Proc. Of the 5th CCGrid, Cardiff, UK, 2005.
- [3] D. Tsafir, Y. Etsion, and D. G. Feitelson. Backfilling using system- generated predictions rather than user runtime estimates. IEEE Trans. Parallel Distrib. Syst., 18(6):789–803, 2007.
- [4] T.H. Cormen. Introduction to algorithms. The MIT press, 2001.
- [5] Henderson, R.L.: Job scheduling under the portable batch system. In: Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing (IPPS 1995), pp. 279–294. Springer, London, UK (1995)
- [6] Kannan, S., Roberts, M., Mayes, P., Brelsford, D., Skovira, J.F.: Workload Management with LoadLeveler. IBM Press (2001)
- [7] Yom-Tov, E., Aridor, Y.: Improving resource matching through estimation of actual job requirements. In: IBM Research Report H-0244 (2006)
- [8] Zhou, S.: LSF: Load sharing in large-scale heterogeneous distributed systems. In: Proc. Workshop on Cluster Computing (1992)
- [9] Thorpe, S.R., Battestilli, L., Karmous-Edwards, G., Hutano, A., MacLaren, J., Mambretti, J., Moore, J.H., Sundar, K.S., Xin, Y., Takefusa, A., Hayashi, M., Hirano, A., Okamoto, S., Kudoh, T., Tsukishima, Y., Otani, T., Nakada, H., Tanaka, H., Taniguchi, A., Sameshima, Y., Jinno, M.: G-lambda and EnLIGHTened: Wrapped In Middleware Co-allocating Compute and Network Resource.

- [10] Marco A. S. Netto and Rajkumar Buyya: Rescheduling Co-Allocation Requests based on Flexible Advance Reservations and Processor Remapping , 9th Grid Computing Conference, 2008
- [11] Shikharesh Majumdar, The “Any-Schedulability” Criterion for Providing QoS Guarantees Through Advance Reservation Requests, 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009
- [12] Yan Ma , Bin Gong , Lida Zou, Energy-Efficient Scheduling Algorithm of Task Dependent Graph on DVS-Unable Cluster System, 10th IEEE/ACM International Conference on Grid Computing, 2009
- [13] Sofia K. Dimitriadou, Helen D. Karatza; Job Scheduling in a Distributed System Using Backfilling with Inaccurate Runtime Computations, 2010 International Conference on Complex, Intelligent and Software Intensive Systems
- [14] S. Dimitriadou and H. D. Karatza, “Multi-Site Allocation Policies on a Grid and Local Level”, to appear in the Proceedings of the Fourth International Workshop on Practical Applications of Stochastic Modelling (PASM’09), (Mascots 2009 Workshop), 24 Sept. 2009, Imperial College, London, Elsevier’s ENTCS (Electronic Notes in Theoretical Computer Science)
- [15] Qiang Wei, Student Member, IEEE, Matei Ripeanu, and Konstantin Beznosov, Member, IEEE, Cooperative Secondary Authorization Recycling, IEEE transactions on parallel and distributed systems, vol. 20, no. 2, february 2009

□□□