Indian Journal of Science and Technology, Vol 8(35), DOI: 10.17485/ijst/2015/v8i35/86643, December 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Constraint-based Decentralized Task Routing Approach to Large-Scale Scheduling in Cloud Environment

M. Lavanya[1]*, T. N. Janani[2], S. Sushmita[2], M. Sunandha[2], V. Vaithiyanathan[1] and S. Saravanan[1]

[1]School of Computing, SASTRA University, Thanjavur - 613401, Tamil Nadu, India; m_lavanyass@ict.sastra.edu, vvn@it.sastra.edu, saran@core.sastra.edu
[2] Information and Communication Technology, SASTRA University, Thanjavur - 613401, Tamil Nadu, India; natarajanjananitn@gmail.com, sushmitasuresh10@gmail.com, sunandhamadhes16@gmail.com

Abstract

This article mainly focused on cloud scheduling with constraint based decentralized task routing. The job of scheduling tasks across various nodes in a hierarchical network scenario is an exigent problem. The concept of decentralized distribution scheme proposed in base paper is time-consuming since it has to compute the availability function for each and every node. In this paper we proposed a CBDA (Constraint Based Decentralized Algorithm) which offers the expediency of being quick and "Make-span minimization policy" is implemented to reduce the completion time of the currently executing nodes. In our presumption, the submission nodes are semi centralized and it can store the availability information of the nodes or routers within its area. This paper considers the allotment of the tasks to the execution nodes which are unoccupied by other tasks. The dynamic allotment of the tasks to the nodes in the tree based approach is the major criteria for selecting the desired node. This paper proposes a trade-off between fully centralized model and the decentralized model by implementing a new constraint based decentralization scheme which saves time consumption and enhances efficiency of task scheduling.

Keywords: Cloud Computing, Constraint based Method Scheduling, Decentralized Router, Dynamic Task

1. Introduction

Cloudcomputing offers a large number of services to millions of users across the globe over the Internet. For instance as an executive of your company providing software and hardware to all the employees individually is highly impossible. But through the cloud technology we can able to accomplish this task. The supremacy of cloud computing relies on its features like virtualization, reliability, maintenance and performance. The major issue of cloud computing is scheduling of tasks over billions of nodes. Condor[1], a distributed job scheduler furnishes a queuing methodology, resource allocation modules for managing massive workloads. The distributed scheduling model[3], which depends on information available from CAN[10] (Content Addressable Network), offers a scalable routing.

CBDA provides quicker scheduling of the nodes as it implements the "Make-span minimization policy" to lessen the completion time of the nodes that are being executed currently. Our model makes use of the tree based network overlay. This produces a tradeoff between the fully centralized and decentralized models and thereby enhancing the efficiency of task routing. Scalability, fault tolerance, Versatility and time-efficiency are some of the salient features of our model.

*Author for correspondence

The CBDA proposed model contributes an extensible scheduling scheme to all scheduling platforms. The goal of condor is to enhance, implement and access the mechanism that supports high throughput computing. This being a centralized technique, needs detailed information about every node in order to produce high outcome. BOINC[2] is a software system used for grid computing that assists the researchers to designate extensive computer preprocessing power. It procures these large scales by considering even the slightest detail about all the nodes which are capable of executing the required task. It proposes a search for the extraterrestrial intelligence which analyzes radio signals, searching for a sign of extraterrestrial intelligence has been one of the many activities undertaken as part of SETI@home.

Hadoop uses centralized datasets and various scheduling algorithms to solve a number of cloud computing issues like traffic during computation phase[4,5]. Map Reduce actually is a programming model which implements processing and spawning large datasets and a wide variety of real world tasks. It has two distinct functions namely map and reduce which runs in parallel thus saving computation time. Several proposals involve the usage of grid schedulers and thus aggregating information which describes each domain.

In Rodero's paper[6] uses the domain information to rank brokers suitably. The meta-broker architecture is being described. It splits the aggregated information to each domain in order to find a suitable broker who can accomplish the intended job. A two level hierarchy has been implemented here which operates with a centralized scheduler that is capable of routing the tasks to its respective nodes. Decentralized schemes[8] are also proposed in which some makes use of the clustered domain information.

In[7] author used to construct a chord-like DHT on a distributed environment with the support of the clustered domain information for grid resource monitoring. In[9] author specifies a CAN overlay model used by Wave Grid. This approach is used to arrange the nodes systematically based on the resources which are currently unoccupied and piling up the length information of the corresponding queues. They rationalize the CAN arrangement based on its good extensible and scaling capabilities better than a tree overlay structure, but it does not use a hierarchical structure which is less likely adapted for the clustering of data.

The heterogeneity[11] of workflow while organizing cloud resources is done by discovering concurrently running tasks and then distributing them accordingly. For the allocation of the resources efficient mechanisms are proposed[12]. Scheduling of independent task with accessible resources to reduce the time of execution is adapted by PBCOPSO[13].

2. Model Overview

The proposed model consists of a tree based network overlay. The tree structure is mostly a balanced binary tree. The model overview is specified in Figure 1.

2.1 Notation

In order to ease the readability of this section and to understand the proposed algorithm some notations are

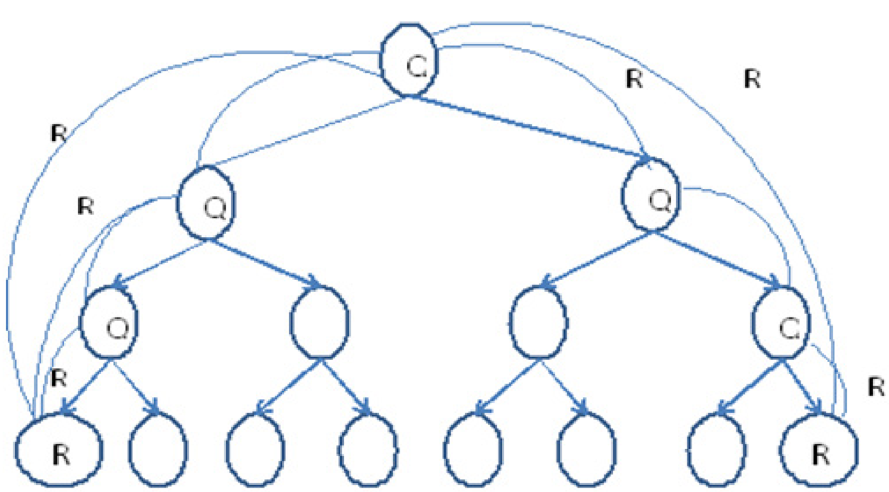

Figure 1. Ri tasks are scheduled in the tree based hierarchy.

included in the Table 1. The specific use of each of these notions will be described further in the paper.

3. Proposed Method

The semi-centralized model proposed here can be applied for various scheduling policies. Scheduling policies are the factor that determines what tasks can be scheduled. It schedules based on the Accessibility Information (AI_i) and the algorithm proposed in our paper. The accessibility information consists of framework of information. Let $AI_i = \{Nn_i, Ne_i, Nf_i, Ct_i\}$ be the information about all the nodes in the hierarchy. This information's are contained within routing nodes (Rn). Let R_i be the number of tasks that are waiting to be executed. R_i has information about the length of the tasks, the memory and disk space needed for it to be executed and also the efficiency required by the nodes to finish the task in the allotted time and also producing maximized results. This kind of information is essential in the scheduling algorithm in order to schedule the jobs in a parallel manner.

The scheduling policy determines whether the queue has free space to add the newly arrived tasks. Our assumption is derived from the idea that there is no distinct priority to the tasks. So the tasks are apportioned across the routing nodes and added in the queue. Generally FCFS or EDF algorithms are used. CBDA algorithm uses the accessibility information for routing the tasks in the desired direction. The tasks that are to be routed are grouped in to three new types of tasks which will be going through the left, right and the parent node. The algorithm does not have any documentation for the tasks which are already routed. This enhances scalability and fault tolerance than the centralized models.

Table 1. Notation in texts and in algorithms

Rn	– Routing Nodes.
R_i	– Number of tasks to be scheduled.
Nn_i	– Number of nodes in the scheduled.
Ne_i	– Number of tasks executing in each nodes.
Nf_i	– Number of tasks that can be allocated in the nodes (free space).
Ct_i	– Computing efficiency of nodes in the networking hierarchy.
Nt_i	– Number of tasks allocated in the nodes from the waiting tasks.

4. Algorithm

- If (Rn== rightchild)

(i) Checks the accessibility information ($AI_{i(R)}$) and finds suitable nodes.

(ii) If number of tasks to be allocated < the number of free space in the nodes

  (i) Allocate all the tasks in that node.

(iii) Else

  (i) Allocate the tasks sufficing the free space.

  (ii) Repeat step2 for the remaining tasks.

- Else if (Rn==leftchild)

(i) Checks the accessibility information ($AI_{i(L)}$) and finds suitable nodes.

(ii) Repeat the steps from 3 to 7.

- Else

(i) Checks the accessibility information ($AI_{i(P)}$) and find suitable nodes.

(ii) Repeat the steps from 3 to 7.

- End.

5. Experimental Analysis

Our proposed model overcomes all the challenges that are proposed in the previous models. The model should be extensible, scalable, and accurate and fault tolerant. By observing the results our model is found to produce better performance than the centralized and decentralized schemes.

The features of condor, decentralized and constraint based decentralized approach are compared in the Figure 2. The outcome of the comparison is found and the results are tabulated in Table 2. The time consumption stumbling block can be controlled by the CBDA algorithm.

6. Conclusion and Future work

In our proposed algorithm, the availability function need not be calculated for each node. Instead the task will be routed to the node where the computation efficiency is fast. Considering many nodes with the available space (Nf_i) the CBDA will choose the most suitable node for executing the particular task. Due to this advantage, time consumption is reduced when compared to other scheduling techniques.

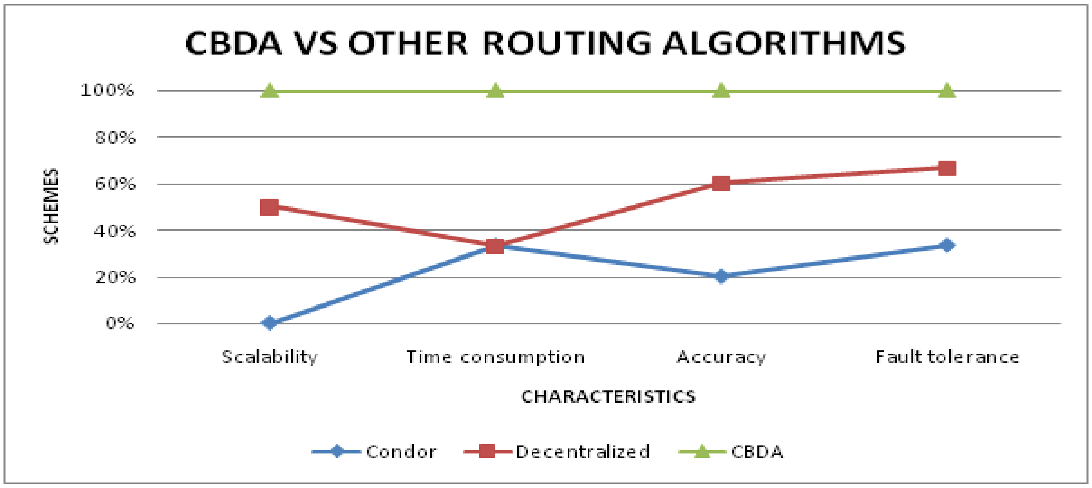

Figure 2. Comparison chart for CBDA Vs other routing algorithms.

Table 2. Comparison of schemes

Characteristics	Condor	Decentralized	CBDA (Proposed Algorithm)
Scalability	Poor	Good	Good
Time-consumption	Average	Poor	Good
Accuracy	Average	Good	Good
Fault tolerance	Good	Good	Good

7. References

1. Tannenbaum T, Wright D, Miller K, Livny M. Condor: A distributed job scheduler. Beowulf Cluster Computing with Linux; 2002. p. 307–50.
2. Anderson D. A system for public-resource computing and storage. Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, IEEE Computer Society; 2004. p. 4–10.
3. Celaya J, Arronategui U. A task routing approach to large-scale scheduling. Future Generation Computer Systems. 2013; 29(5):1097–111.
4. Abramson D, Bethwaite B, Enticott C, Garic S, Peachey T. Parameter exploration in science and engineering using many-task computing. IEEE Transactions on Parallel and Distributed Systems. 2011; 22(6):960–73.
5. Dean J, Ghemawat S. Map reduce: Simplified data processing on large clusters. Communications of the ACM. 2008; 51(1):1–13.
6. Rodero I, Guim F, Corbalan J, Fong L, Sadjadi SM. Grid broker selection strategies using aggregated resource information. Future Generation Computer Systems. 2010; 26(1):72–86.
7. Cai M, Hwang K. Distributed aggregation algorithms with load-balancing for scalable grid resource monitoring. IEEE International Parallel and Distributed Processing Symposium, IPDPS'07; 2007. p. 1–10.
8. Stoica I, Morris R, Liben-Nowell D. A scalable peer-to-peer lookup protocol for Internet applications. IEEE/ACM Transactions on Networking. 2003; 11(1):17–32.
9. Kim J-S, Nam B, Keleher P, Marsh M, Bhattacharjee B, Sussman A. Tradeoffs in matching jobs and balancing load for distributed desktop grids. Future Generation Computer Systems. 2008; 24(5):415–24.
10. Ratnasamy S, Francis P, Handley M, Karp R, Schenker S. A scalable content addressable network. Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM'01; 2006. p. 161–72.
11. Bagheri R, Jahanshahi M. Scheduling workflow applications on the heterogeneous cloud resources. Indian Journal of Science and Technology. 2015; 8(12):1–8.
12. Shyamala K, Rani ST. An analysis on efficient resource allocation mechanisms in cloud computing. Indian Journal of Science and Technology. 2015; 8(9):814–21.
13. Priyadarsini RJ, Arockiam L. PBCOPSO: A parallel optimization algorithm for task scheduling in cloud environment. Indian Journal of Science and Technology. 2015; 8(16):1–5.