Task-Scheduling in Cloud Computing using Credit Based Assignment Problem

Mousumi Paul[a] , Goutam Sanyal[b]

Department of CSE
National Institute of Technology, Durgapur,
Mahatma Gandhi Avenue, West Bengal, India - 713209

[a]Email: mousumipaul.88@gmail.com,[b]Email: nitgsanyal@gmail.com

Abstract—**This Cloud computing is a latest new computing paradigm where applications, data and IT services are provided across dynamic and geographically dispersed organization. Job scheduling system problem is a nucleus and demanding issue in Cloud Computing. How to utilize Cloud computing resources proficiently and gain the maximum profits with job scheduling system is one of the Cloud computing service providers' ultimate objectives. In this paper we have used credit based scheduling decision to evaluate the entire group of task in the task queue and find the minimal completion time of all task. Here cost matrix has been generated as the fair tendency of a task to be assigned in a resource.**

Keywords- Lexi-search, Cloud computing, ITMAP, Partial word.

I. INTRODUCTION

Many technology experts believe that cloud computing is poised to change the way we access technology — and that it may be as game-changing as the commercialization of the Internet over a decade ago. Cloud computing enables innovation. It alleviates the need of innovators to find resources to develop, test, and make their innovations available to the user community. Innovators are free to focus on the innovation rather than the logistics of finding and managing resources that enable the innovation.

So that efficient task scheduling problems and resource management are relate to the efficiency of the whole cloud computing facilities. These tasks are parallel processed on the nodes of the cluster by the policy which strives to keep the work as close to the data as possible. The scheduling algorithms in distributed systems usually have the goals of spreading the load on processors and maximizing their utilization while minimizing the total task execution time. Several heuristic algorithms has been introduced in task scheduling.

The motivation of this paper is to establish a scheduling mechanism which follows the Lexi – search approach of Shalini Arora and M.C. Puri [7] to find an optimal feasible assignment. Task scheduling has been treated as general assignment problem to find the minimal cost. Here cost matrix is generated from a probabilistic factor based on some most vital condition of efficient task scheduling such as task arrival, task waiting time and the most important task processing time in a resource. The cost for assigning a task into a resource is probabilistic result considering the above criteria.

II. CLOUD ARCHITECTURE

Cloud computing refers to the use of networked infrastructure software and the capacity to provide resources to users in an on-demand environment. Cloud computing is fully enabled by virtualization technology (hypervisors) and virtual appliances. A virtual appliance is an application that is bundled with all the components that it needs to run, along with a streamlined operating system. In a cloud computing environment, a virtual appliance can be instantly provisioned and decommissioned as needed, without complex configuration of the operating environment. This flexibility is the key advantage to cloud computing, and what distinguishes it from other forms of grid or utility computing and software as a service (SaaS).

Cloud computing system scales applications by maximizing concurrency and using computing resources more efficiently One must optimize locking duration, statelessness, sharing pooled resources such as task threads and network connections bus, cache reference data and partition large databases for scaling services to a large number of users. In Figure 1 there is a cloud computing management which comprises different managers, information manager, Transfer manager, execution manager and scheduler manager. This papers concentrates on scheduler manager.

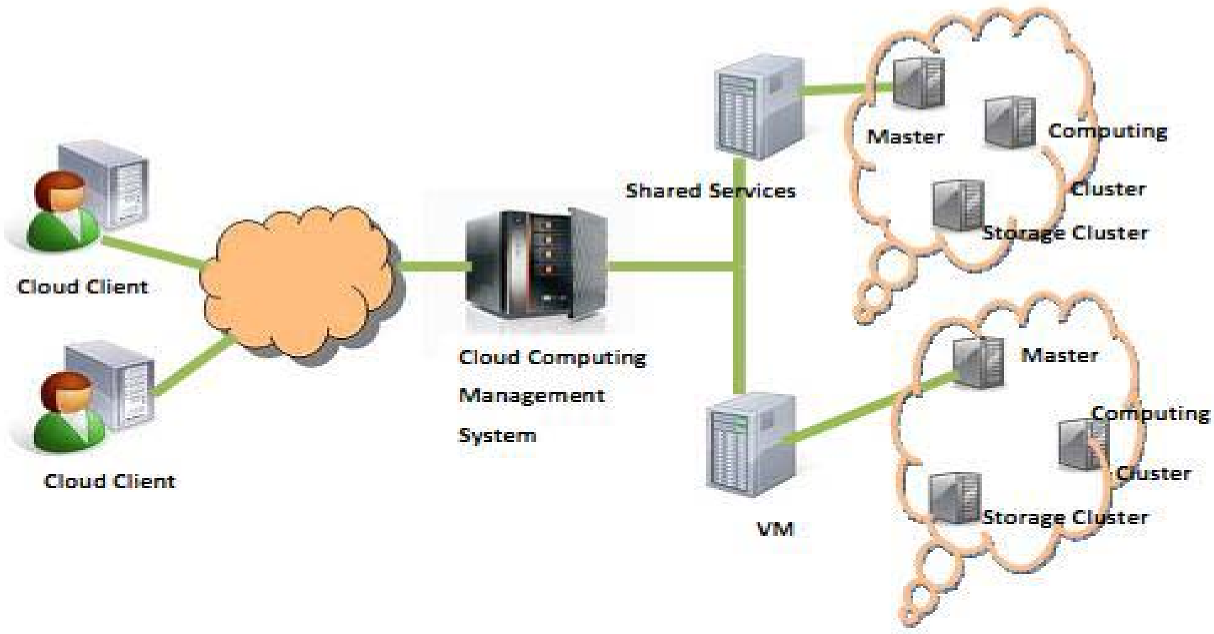

Fig: Diagram of proposed architecture

III. PROPOSED METHODOLOGY

Task scheduling is a kind of transportation problem.

There is a set *J* {1,2,3,……..,n} of **n** jobs and a set *I* {1,2,3,……..,m} of **m** available resources in which jobs will be assigned for execution. This has some restrictions that one job will be allocated to only one resource and each resource has to do only one job. All the resources start doing the jobs simultaneously but a resource doing more than one job has to do them one after the other. Let $P_{i,j}$ be the probabilistic measurement or credit of a job to be assigned to a particular resource. This $P_{i,j}$ is calculated as follows:

$$\mathbf{P}_{i,j} = \frac{\mathbf{a}_{i,j}\ \mathbf{X}\ \mathbf{e}_{i,j}}{\sum \mathbf{a}_{i,j}\ \mathbf{X}\ \sum \mathbf{e}_{i,j}}$$

$\mathbf{a}_{i,j}$ is the availability of a resource **j** to be free after executing task **i . $\mathbf{a}_{i,j}$** is also computed as the sum of arrival time of a task and the execution time of task i to be executed on j resource.

$$\mathbf{a}_{i,j} = \textbf{arrival time of task i} + \mathbf{e}_{i,j}$$

[$\mathbf{e}_{i,j}$ = execution time of task i on resource j**]**

The aim is to find that assignments of the resources to the jobs for which the corresponding time of completion of all jobs is the minimum. If the decision variable $\mathbf{x}_{i,j}$ (i,j) ϵ *I* X *J* takes the value 1 when the i th resource execute the j th job and 0 otherwise. So the mathematical formulation of the above problem is :

$$minT(X) = \max(\sum_{j=1}^{n} Q), \textbf{ where } i \in I$$

Where $Q = P_{i,j} : X_{i,j} > 0$

Subject to $\sum_{i \in I} x_{i,j} = 1 , \ j \in J$ ----------------------------- (1)

$\sum_{j \in J} x_{i,j} \geq 1 , \ i \in I$ ----------------------------- (2)

$X_{i,j} = 0$ or 1 , $(i, j) \in IXJ$ ------------------- (3)

Since number of resource is less than the number of jobs, we call this problem an *Imbalanced Time Minimizing Assignment Problem (**ITMAP**).* Clearly it always has a feasible solution.
An assignment, X = *{xij},* is one which satisfies (1) and (3), and *T(X)* is the corresponding time of completion of the jobs. An assignment is called a feasible assignment if (2) is also satisfied.
This assignment $\mathbf{x_{i,j}}$ can also be represented by a row vector as follows:

w = (il, i2,..., in), ……………………….(4)

where, all ij's are not distinct, clearly [w[= I J[= n. Thus, the assignment represented by (4) implies that the jth job is done by the ij th person, j = 1,2,..., n.
Each assignment in its vector form (4) can be thought of as a word, w, of length n, with letters ij's from the set I. Let W = {w} be the set of all feasible words of length n. Then for a feasible word, say w, given by (4), the corresponding feasible assignment $X^w = \{ \mathbf{x_{i,j}}^{w} \}$ is given by :

$$X^{w}_{i,j} = 1, \ j=1,2,\ldots\ldots\ldots\ldots\ldots,n.$$
$$X^{w}_{i,j} = 0, \ (i, j) \in IXJ - \{(i,j) : j \in J \}$$

The value of $T(X^w)$ for this feasible assignment corresponding to w is

$$T(X^w) = \max\left(\sum_{j=1}^{n} (\mathbf{P_{i,j}} : \mathbf{X^{w}_{i,j}} = 1)\right)$$

For this purpose we have defined:

A. Alphabet matrix:

It is an m × n matrix formed by the positions of the elements of the given m × n matrix *{* $\mathbf{P_{i,j}}$ *}* of credit. The jth column *of AB* consists of the positions of the entries in the jth column of the matrix { $\mathbf{P_{i,j}}$ } when they are arranged in the non-decreasing order of their values. Let *ab(y,j)* stands for the y th entry in the jth column *of AB.* Therefore, *ab(1,j)* corresponds to the smallest entry in the jth column of the matrix *{* $\mathbf{P_{i,j}}$ *}* that is,
$\min_i\{ \mathbf{P_{i,j}} \} = P_{ab}(1,j)j$. If y < z, then $P_{ab}(y,j)j \leq P_{ab}(z,j)j$.
Thus, the jth column *of AB* is [ab(1,j), *ab(2,j),..., ab(m,j)]'* where, (t) stands for the transpose. Clearly $P_{ab}(1,j)j \leq P_{ab}(2,j)j \leq P_{ab}(3,j)j \ldots\ldots\ldots \leq P_{ab}(m,j)j$

All the words in W can be systematically generated by considering the elements of the jth column of *AB* in the jth position (j -- 1,2,..., n) of a word, i.e., *ij E {ab(q,j), q = 1,2,..., m}.*

B. Partial Word (Pw):

Pw = (il, i2,..., it), $r \leq n$, represents a partial word. A partial assignment corresponding to it consists of assigning the jth job to the ijth resource, j = 1,2,..., r (jobs r + 1, r + 2 , . . . , n are still to be assigned). Pw defines a block of words each of which has first r letters as il, i2 . . . . , it. In this sense Pw is called the leader of this block of words. If a partial word is such that I/[> n - IPwl, then clearly this partial word cannot contain a feasible word, where I[[is the index set of unassigned persons. Such a partial word is called an *infeasible partial word.* On the other hand, | I | ≤ n - |Pw| then Pw is called a *feasible partial word.*
Contribution to the objective function T(.) by the partial assignment, say X^{Pw}, corresponding to Pw is given by

$$T(X^{pw}) = \max\left(\sum_{j=1}^{n} (\mathbf{P_{i,j}} : \mathbf{X^{pw}_{i,j}} = 1)\right) \ , \ i \in pw$$

Clearly, for a word 'w' whose leader is Pw , we have $\boldsymbol{T(X^{w}) \geq T(X^{pw})}$

C. N Notation :

To starting upper bound on the value of the objective function T(.).
J_s J-{j_1,j_2,......j_{s-1}} (Clearly J_1=J)

$|\bar{I}|$ index set of unsigned person.
T_u updated upper bound on the value o the objective function T(.).
Φ empty set

D. Upper bound and objective function T(.) evaluation :

For each $i \in I$, find min($P_{i,j} = P_{iji}$) (say) and set X_{iji} =1, $i \in I$,Then each of the m person is assigned to unique job in the set (j_1,j_2,....j_m). For allocation of the remaing jobs in J_{m+1} , proceed as follows : For

$$\min\left(\sum_{j \in J-Jm+k} (\mathbf{P_{i,j}} : \mathbf{X_{ij}} = \mathbf{1}) + \mathbf{min}\ \mathbf{P_{i,j}}\right) = \sum_{j \in J-Jm+k} (\mathbf{P_{im+k}}) : \mathbf{X_{im+kj}} = \mathbf{1}) + \mathbf{P_{im+k/m+k}} \quad \textbf{(Say)}$$

Then allocate job J_{m+k} to resource i_{m+k} ,k=1,2,…(n-m).

'T_0' will be given by $T_0 = \max(\sum_{j \in J} (P_{ij} : X_{ij} = 1))$

This heusistic will provide the starting upper bound on the value of T(.) quite close to its optima value.

A feasible assignment then obtained is
For $i \in I$

$X_{i,j}$ = 1 when $j=j_i$
= 0 when $j \neq j_i$

For $j \in jm+1$

$X_{i,j}$ = 1 when $i=i_j$
= 0 when $i \neq i_j$

Let the feasible word corresponding to this feasible assignment be w = (ab(y_1,1), ab(y_2,1),........ ab(y_n,1)).
Therefore the above feasible assignment can then be given as
$X_{ab(y,j)j}$ = 1, j=1,2,......,n.
X_{ij} = 0 ,
$(i, j) \in IXj$ -{(ab(yi,j);j), j=1,2,3…,n}

IV. CONCLUSIONS

With the advancement of Cloud technologies rapidly, there is a new need for tools to study and analyze the benefits of the technology and how best to apply the technology to large-scaled applications. The proposed method considers the scheduling problem as the assignment problem in mathematics where the cost matrix gives the cost of a task to be assigned into a resource. Here cost has been considered as credit or the probabilistic measurement thus only the processing time of a job is not been given importance but the other issues are considered such as the probability of a resource to be free soon after executing a task so that it will be available for other waiting job. Job which has the highest probability to get a resource as well as the resource which fits better for a job are assigned in a manner that one resource get one job at a time.

REFERENCES

[1] Shalini Arora and M.C. Puri “A variant of time minimizing assignment problem” European Journal of Operational Research 110 (1998) 314-325.

[2] Dong, J. Luo, L. Gao, and L. Ge, "A Grid Task Scheduling Algorithm Based on QoS Priority Grouping," In the Proceedings of the Fifth International Conference on Grid and Cooperative Computing (GCC’06), IEEE, 2006.

[3] Yair Amir, Baruch Awerbuch, Amnon Barak, R. Sean Borgstrom, Arie Keren, "An Opportunity Cost Approach for Job Assignment in a Scalable Computing Cluster", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No.7, July 2000.

[4] Agarwal, A., Kumar, P.: Economical Duplication Based Task Scheduling for Heterogeneous and Homogeneous Computing Systems. In: Proceedings of the Advance Computing Conference, 2009(IACC’ 09), pp. 87-93, IEEE Computer Society(2009).

[5] Remzi H. Arpaci, Andrea C. Dusseau, Amin M. Vahdat, et al, "The Interaction of Parallel and Sequential Workloads on a Network of Machines", *Proc. of ACM SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 267-278, May, 1995.

[6] H.Liu, A.Abraham, A.E.Hassanien, Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation Computer Systems (2009), doi:10,1016/j.future.2009.05.022.

[7] T. D. Braun, H. J. Siegel, N. Beck, L. L. Boloni, M. Maheswaran, A. I. Reuther, J.P. Robertson, M. D. Theys, B. Yao, D. Hensgen, and R. F. Freund, “A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems,” *Journal of Parallel and Distributed Computing*, vol. 61, issue 6, pp. 810-837, Jun. 2001.

[8] Sandeep Tayal / (IJAEST) International Journal of Advanced Engineering Sciences And Technologies Vol No. 5, Issue No. 2, 111 – 115.

[9] Linguo Gong, Xian-He Sun, and Edward F. Waston, "Performance Modeling and Prediction of Non-Dedicated Network Computing", *IEEE Trans. on Computer*, Vol. 51, No 9, September, 2002.

[10] Downey, A.B., "Using pathchar to estimate Internet link characteristics", *Proc. SIGCOMM 1999*, pp. 241-250, Cambridge, MA, Sept. 1999.

[11] T.D.Braun, H.J.Siegel, N.Beck, D.A.Hensgen, R.F.Freund, A comparison of eleven static heuristics for mapping a class of independent tasks on heterogeneous distributed systems,Journal of Parrallel and Distributed Computing,2001,pp.810- 837.

[12] E.Ullah Munir, J. Li, and Sh. Shi, 2007. QoS Sufferage Heuristic for Independent Task Scheduling in Grid. Information Technology Journal, 6 (8): 1166-1170.