Divisible Resource Allocation to Minimize Cloud Task Length

Sneha S. Tile **Prof. Naresh Thoutam**

Abstract— Now-a-days **Cloud computing is used basically to get share access to the divisible resources .All the applications used by number of users are handled by cloud application.The user should get easy and fast access to data in the cloud. Cloud computing environment involves high cost infrastructure. The computational resources in cloud environment should maintained high scalability. These resources need to be provided (allocation and scheduling) to the end users in most efficient manner so that the lots of capabilities of cloud are utilized effectively and efficiently. But there are some constraints on cloud computing like payment budget and on demand resources. Virtual machine is widely used to divide resources to accomplish tasks . Although, there are various algorithms and methods were existing to solve the problem of resource allocation but none of these algorithms can be extended. The present study involves surveying the different task scheduling and resource allocation algorithms for cloud.**

Keywords— **ODRA Algorithm, divisible-resource allocation, convex optimization, upper bound analysis, LOAA Algorithm, web composition.**

I. Introduction

Cloud Computing uses SOA (Service Oriented Architecture) to provide IaaS (Infrastructure as a Service) , SaaS (Software as a Service) , PaaS (Platform as a Service) , DaaS (Data Storage as a Service) , CaaS (Communication as a Service) , HaaS (Hardware as a Service) to cloud users. The end users can use these resources over a network on-demand basis in pay-as-you-say manner.

Cloud computing is an attracting technology in the field of computer science. In Gartner's report, it says that the cloud will bring changes to the IT industry. Traditional task scheduling adopted in distributed systems like grids assumes discrete resource usage model. The processing ability assigned to a task cannot be customized by users elastically. Such an indivisible resource consumption model with discrete computation unit results in a non-trivial problem like binary Integer programming problem, where CPU rates may not be fully utilized. With virtual machine (VM) [11] resource isolation technology the computational resources could be partitioned and reassembled on demand, creating an avenue to improve resource utilization. There is an optimal algorithm (namely local optimal allocation algorithm (LOAA)) [12] minimizing a task's execution length, subject to a set of constraints like user's payment budget and host availability states.

Virtual[5] machine (VM) technology is widely used and being greater and fully developed, compute resources in cloud systems can be allocated on demand which contributes three technologies such as, A deadline-driven resource allocation problem is formulated based on the cloud computing environment facilitated with VM resource separation technology, and also to minimize burden on users' payment. Traditional task scheduling adopted in distributed systems like grids assumes discrete resource usage model . The processing ability assigned to a task cannot be customized by users elastically. Such an indivisible resource consumption model with discrete computation units results in a non-trivial problem like binary Integer programming problem, where CPU rates may not be fully utilized.

In general, services can be classified into two categories: a non-delay system (loss system) and a waiting system. A non-delay system allocates a spare resource immediately to the user upon the arrival of the request, and rejects the request if there is no spare capacity. A waiting system allocates a spare capacity to users in the sequence in which their requests have arrived, instead of allocating resources immediately upon the arrival of a request. This paper assumes a service that runs as non-delay. This paper also assumes static resource allocation, which is the most basic form of resource allocation, although dynamic allocation, which uses process migration and bandwidth consolidation, can increase the utilization of resources.

The paper is organize into four phases they are as follows.

1. In first step formulation of the cloud resource allocation issues a convex optimization problem aiming to minimize task length with divisible resource fractions and a set of constraints is performed

2.In second step, LOAA algorithm is used to outline the task processing procedure and .then prove that LOAA algorithm can also minimize user payments meanwhile based on tasks' final real wall-clock lengths.

3. In the third step the upper bound of task execution length considering prediction errors on task workloads and resource availability, as against to the result under the hypothetically precise prediction is derived.

4. In the fourth step, the algorithm is expanded to adapt the volatile states of the system with multiple web services deployed. The experimental results generated over a real-cluster environment.

Traditional task scheduling adopted in distributed systems like grids assumes discrete resource usage model [1], [2], [3]. The processing ability assigned to a task cannot be customized by users elastically. Such an indivisible resource consumption model with discrete computation units results in a non-trivial problem like binary Integer programming problem, where CPU rates may not be fully utilized.

With virtual machine (VM) resource isolation technology [5] the computational resources could be partitioned and reassembled on demand, creating an avenue to improve resource utilization. In our previous work [6], we proposed an optimal algorithm (namely local optimal allocation algorithm (LOAA)) minimizing a task's execution length, subject to a set of constraints like user's payment budget and host availability states.

II. Literature Survey

Many related works have been done to achieve efficient resource allocation scheme. Resource provisioning in cloud computing environment is done with the main aim of achieving load balancing. Based on various factors like spatial distribution of cloud nodes, algorithm complexity, storage/replication, point of failure etc. different techniques have evolved to provision the resources in balanced manner. The provisioning is done taking into account whether the environment is static or dynamic.

A large portion of the work in resource allocation in cloud computing mainly focused on the cost-effectiveness and easy maintenance of the systems [1].

Most of the work has been descriptive in nature. Patricia et al. [2] discusses the process of distributed cloud in which application developer considers the geographically distributed cloud. [2] Highlights and categorizes the main challenges inherent to the resource allocation process particular to distributed clouds, it offers stepwise view of this process that covers the initial modelling phase through to the optimization phase. This paper gives the evaluation of current network resource allocation strategies and their possible applicability in Cloud Computing Environment by M. Asad Arfeen [3].

Atsuo Inomata et al. [4] has proposed a dynamic resource allocation method based on the load of VMs on IaaS, abbreviated as DAIaS. This method allows users to dynamically add and/or delete one or more resources on the basis of the load. In this the conditions are specified by the user.

It has been believed that resources are virtualized and delivered to users as services a market-based resource allocation will be effective in a cloud computing environment (Fujiwara et al. [5]) and in such a market mechanism to allocate services to participants efficiently has proposed. It depends on services workflow and their current allocation and coallocation.

The processing ability and network bandwidth needed to access data in cloud are important in cloud computing to allocate resources.

Tomita et al [6] proposed a method for the congestion control method for a cloud computing environment which is used to decrease the amount of available resource for congested resource type, instead of restricting all service requests as in the existing networks.

There are some critical cases about the limitation of electric power capacity available in each area, assuming a cloud computing environment in which both processing ability and network bandwidth are allocated simultaneously so Mochizuki and Kuribayashi [7] presents cloud resource allocation guidelines for this.

Resources are allocated to nodes in cloud by dynamically[5] calculating their load balancing. For maximizing the resource usage and minimizing the cloud task length task Scheduling algorithms are used. When the number of customers requests for same resources at the same time rescheduling is used. Each and every task have different requirement of more bandwidth, response time, resource costs, and capacity of memory also differs. The load balancing of task is maintained by task scheduling algorithm . The efficiency of cloud environment is improved by using different task scheduling algorithm.

III. Proposed System

The Proposed system focuses on minimizing the task execution in cloud computing. It involves hardware and software design. It also requires divisible resources for minimizing task length.

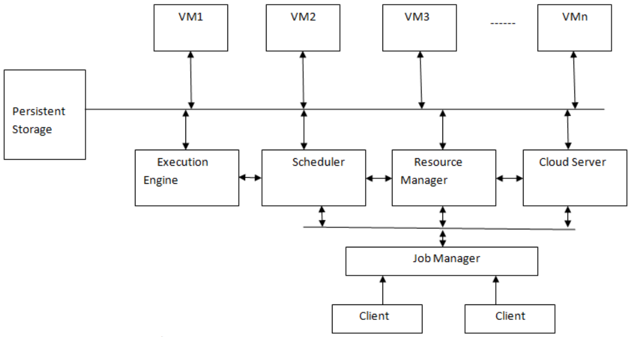


Fig: Architecture Diagram

The above diagram shows the architecture of the proposed system in which persistent storage and cloud server are used to store the data. The flow of the system goes in following way. When the job is submitted to execute the task it is first given to the execution engine. At execution engine virtual machine is created to complete multiple task, scheduler is module in the system to schedule the jobs as per their priority and requirements. Scheduler submit's the list of jobs to the resource manager. Resource manager behaves like job manager to manage all the jobs to complete the task. Finally all tasks gets completed by allocating resources on dividing them. Update scheduler is a module which is used to update the list of resources so that other jobs gets resources to accomplish the further tasks.

ADVANTAGES OF PROPOSED SYSTEM

1.Based on the elastic resource usage model, this paper aims to design a resource allocation algorithm with high prediction error tolerance ability, also minimizing users‘ payment subject to their expected deadlines.

2.The idle physical resources can be arbitrarily divides and allocated to new tasks; the divisible resource allocation is implemented on virtual machines so it is important to maintain flexibility of virtual machine.

A. Implementation of algorithm

The algorithm given shows the implementation of the proposed system.

Algorithms

- Skeleton of ODRA algorithm
- Local Optimal Allocation Algorithm
- Dynamic ODRA

Algorithm for Preemptive scheduling:

```
1. Input: Let {T1, T2,...,Tk } be the accepted tasks in the ready queue and let ei be the expected execution time of Ti. Let current time be t and let T0 be the task currently being executed. Let the expected utility density threshold be μ.
2. if a new task, i.e. Tp arrives then
3. Check if Tp should preempt the current task or not;
4. if Preemption allowed then
5. Tp preempts the current task and starts being executed;
6. End if
7. If Preemption not allowed then
8. Accept Tp if
```

$$\frac{U_p(e_0)}{e_0} \geq \mu_e$$

D is the set of Dataset.
F is the set of Functions.
Y is a set of techniques use to minimize cloud task length by using adaptive algorithms

```
9. Reject Tp if
   $\frac{U_p(e_0)}{e_0} \geq \frac{\mu e_0}{e_0}$
10. End if
11. Remove Tj in the ready queue if $\mu$
   $\frac{U_p(e_0)}{e_j} \geq \mu$
12. End if
13. If at preemption check point then
14. PREEMPTION CHECKING;
15. End if
16. If T0 is completed then
17. Choose the highest expected utility density task Ti to run.
18. Remove Tj in the ready queue if $\mu$
   $\frac{U_j(e_i)}{e_j} \geq \mu$
19. End if
20. If t = the critical time of p0 then
21. Abort p0 immediately
22. Choose the highest expected utility density task pi to run.
23. Remove pj in the ready queue
24. End if
```

IV. Abbreviations and Acronyms

A. Optimal divisible resource allocation(ODRA)

This method focuses on how to make full use of the multi-attribute resources facilitated by such a resource isolation technology to optimize the task's execution efficiency, under users' specific resource demands (such as execution payment and service level). Another challenge about the resource allocation issue is the potentially high-dimensional execution. Since task's workloads as well as computational resources are multi-attribute, the execution will be multi-dimensional in nature. Even through considering only one resource attribute (for example, the task may be computation-intensive application), a task may also be split to multiple sequential execution steps (or phases), each calling for a different resource capacity and price on demand. When user request for files in any server the processor first checks the available resources with request made by the user if it is not feasible then the resources are divided and accommodated to all the requests made by the user. So it will improve the time consumption by the resources and delay in the access to the data.

B. Implementation of Local optimal allocation algorithm (L**OAA**))

LOAA algorithm is an optimal solution to minimize task length, but it can prove that user payment is also minimized based on task's final wall-clock length.

C. Minimize the upper bound of task execution length

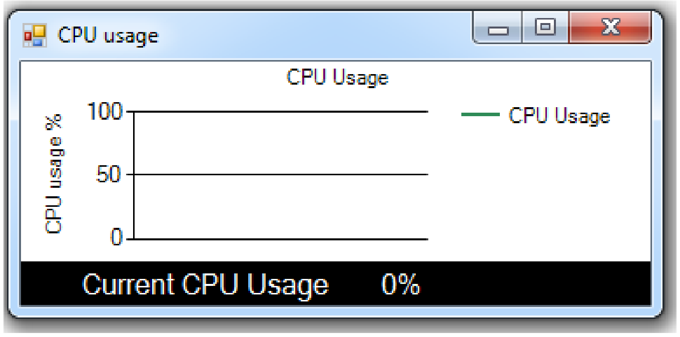
It is used when task's workload and host's availability will be predicted with errors, which is more in the line with reality. For instance, the multi-variant polynomial regression method and Bayes method have been effective in precise workload prediction and host load prediction respectively, yet they are still suffering inevitable margin of prediction errors like 10 percent. On the other hand, the flexible resource partitioning of the cloud systems may definitely result in load dynamics on resource states, and worse still, the collected states are error-prone due to the network propagation delay. The inevitable load prediction errors may significantly affect task's execution in reality. It derive the bound of task length for the LOAA algorithm[4], based on erroneous prediction of task's workload and resource availability, as compared to the theoretically optimal task length with hypothetically accurate information. This is fairly valuable/useful in that users are able to know the worst performance in advance and the resource allocation can be tuned in turn to adapt to user demand based on the bound of task execution length estimated.

D. Dynamic version for load balancing

The algorithm can further extended to a dynamic version[1],[2] to adapt to the load dynamics over time. Due to the dependency between the subtasks (or web services) of a task, the resource availability states for a particular subtask may not be forecasted upon the task's initial submission. Accordingly, we extend our algorithm to be a dynamic (or adaptive) version, which can tune the resource allocation at runtime based on task's execution progress and updated resource availability states.

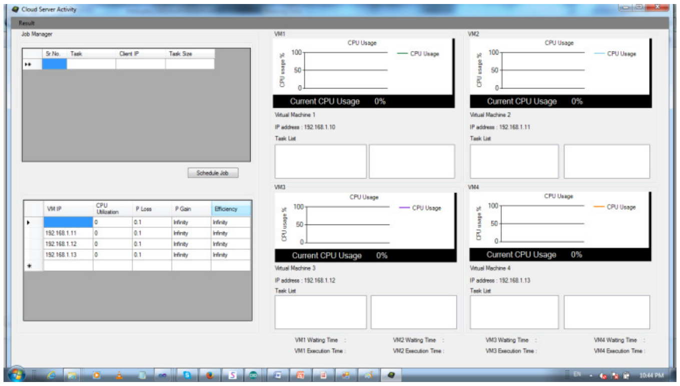
V. Performance Analysis

Following diagram shows the working of the system.



Current CPU Usage 0%

The above diagram shows the initial state of the CPU.



Sr.No.	Task	Client IP	Task Size

VM IP	CPU Utilization	P Loss	P Gain	Efficiency
	0	0.1	Infinity	Infinity
192.168.1.11	0	0.1	Infinity	Infinity
192.168.1.12	0	0.1	Infinity	Infinity
192.168.1.13	0	0.1	Infinity	Infinity

Current CPU Usage 0%
Virtual Machine 1
IP address : 192.168.1.10

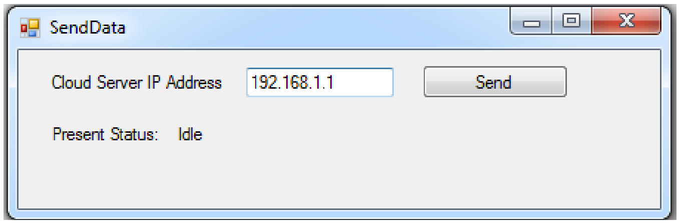
Current CPU Usage 0%
Virtual Machine 2
IP address : 192.168.1.11

Current CPU Usage 0%
Virtual Machine 3
IP address : 192.168.1.12

Current CPU Usage 0%
Virtual Machine 4
IP address : 192.168.1.13

VM1 Waiting Time :
VM1 Execution Time :
VM2 Waiting Time :
VM2 Execution Time :
VM3 Waiting Time :
VM3 Execution Time :
VM4 Waiting Time :
VM4 Execution Time :

The above diagram shows the working of job manager .When the client submits jobs to the CPU it gets divided into different task in which different virtual machines gets created.



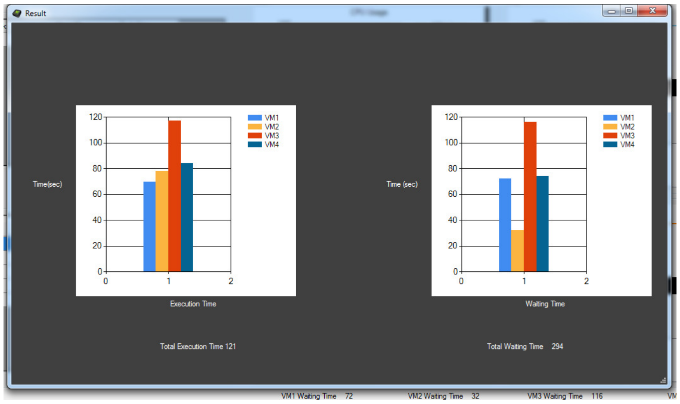
Cloud Server IP Address 192.168.1.1

Present Status: Idle

This shows the how client sends request to the server.

VI. Result

The following graph shows the result set of the above steps.



Total Execution Time 121

Total Waiting Time 294

VM1 Waiting Time 72
VM2 Waiting Time 32
VM3 Waiting Time 116

The above diagram shows the execution time and waiting time of all virtual machine how the task gets divided into different tasks.

Conclusion

This paper optimize the divisible-resource allocation with in-depth analysis on upper bound of task execution length under prediction errors. It also design a dynamic approach that adapts to the load dynamics over task execution progress. This paper evaluate the performance using a real cluster environment with composite web services. These services are of different execution patterns on multiple types of resources. Experiments show that task execution lengths with our ODRA solution are always close to their theoretically optimal results with resource capacity limitation. S

References

[1] Sheng Di, , Cho-Li Wang and Franck Cappello, IEEE "Adaptive Algorithm for Minimizing Cloud Task Length with Prediction Errors" IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1097-1106,April-June 2014

[2] D. Sheng, D. Kondo, and W. Cirne, "Host Load Prediction in a Google Compute Cloud with a Bayesian Model," Proc. IEEE/ACM 24th Int'l Conf. for High Performance Computing, Networking, Storage and Analysis (SC '12), pp. 21:1-21:11, 2013.

[3] S. Di and C.-L. Wang, "Dynamic Optimization of Multi-Attribute Resource Allocation in Self-Organizing Clouds," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 3, Mar. 2013.

[4] S. Di and C-L. Wang, "Minimization of Cloud Task Execution Length with Workload Prediction Errors," Proc. 20th High Performance Computing Conf. (HiPC '13), 2013.

[5] Gideon-II Cluster: http://i.cs.hku.hk/_clwang/Gideon-II, 2012.

[6] F. Chang, J. Ren, and R. Viswanathan, "Optimal Resource Allocation in Clouds," Proc. Third IEEE Int'l Conf. Cloud Computing (Cloud '10), pp. 418-425, 2010.

[7] C. Jiang, C. Wang, X. Liu, and Y. Zhao, "A Survey of Job Scheduling in Grids," Proc. Joint Ninth Asia-Pacific Web and Eighth Int'l Conf. Web-Age Information Management Conf. Advances in Data and Web Management (APWeb/WAIM '07), pp. 419-427, 2007.

[8] D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat, "Enforcing Performance Isolation
Across Virtual Machines in Xen," Proc. Seventh ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '06), pp. 342-362, 2007.

[9] S. Chinni and R. Hiremane, "Virtual Machine Device Queues," techical report, Virtualization Technology White Paper, 2007.

[10] D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat, "Enforcing Performance Isolation
Across Virtual Machines in Xen," Proc. Seventh ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '06), pp. 342-362, 2006.

[11] S. Chinni and R. Hiremane, "Virtual Machine Device Queues," techical report, Virtualization Technology White Paper, 2006.

[12] D. Gupta, L. Cherkasova, R. Gardner, and A. Vahdat, "Enforcing Performance Isolation
Across Virtual Machines in Xen," Proc. Seventh ACM/IFIP/USENIX Int'l Conf. Middleware (Middleware '06), pp. 342-362, 2006.

[13] J.E. Smith and R. Nair, Virtual Machines: Versatile Platforms for Systems and Processes. Morgan Kaufmann, 2005.

[14] Website http://www.net-security.org/secworld.php?id=10886, Article on "Lack of admin rights mitigates most Microsoft vulnerabilities" Posted on 12 April 2011.

[15] Patricia Takako Endo, Andre Vitor de Almeida Palhares, Nadilma Nunes Pereira, 2011. Resource Allocation for Distributed Cloud: Concepts and Research Challenges, IEEE, july 2011.

[16] Hadi Goudarzi and Massoud Pedram University of Southern California, MaximizingProfit in Cloud Computing System via Resource Allocation.

[17] M.Asad Arfeen, Krzysztof Pawlikowski, Andreas Willig .2011,
A Framework for Resource Allocation Strategies in Cloud Computing Environment, 2011 35th IEEE Annual Computer Software and Applications Conference Workshops.

[18] Atsuo Inomata, Taiki Morikawa, Minoru Ikebe. 2011, Proposal
and Evaluation of a Dynamic Resource Allocation Method based on the Load of VMs on IaaS, IEEE 2011.