
T. Naresh¹, A. Jaya Lakshmi², Vuyyuru K. Reddy³

naresh1060@gmail.com¹, jvallabhaneni@hotmail.com², vkrishnareddy@kluniversity.in³
Department of Computer Science and Engineering¹,²,³
K. L. University Vijayawada, India¹,²,³

Abstract—Resource allocation is a plan for utilizing available resources, consider the example human resources, especially in the closer term, to achieve goals for the future. Cloud computing is internet-based effective computing in which large groups of remote servers are networked to give opportunity for the centralized storage of data, and retrieving online computer services or shared resources. Cloud computing gains business customers to scale up and down their resource usage based on requirements. Virtual machine (VM) technology being more and more grown-up, calculate resources in cloud systems can be divided in fine granularity and allocated on demand. The main aim of our approach is that effective prioritization method. Initially many users send the service request for the cloud server; the cloud server decides the priority among the different user request using the modified priority algorithm.

Keywords—Cloud Computing, Payment Minimization, Resource Allocation.

I. INTRODUCTION

Cloud system is supposed to be under a payment minimization model, in order to keep away users’ over-demand of their resources against their true needs. [1]. Distributed computing suppliers convey application through the Internet, which are gotten to from web program, while the business programming and information are put away on servers at a remote location area [4]. Applications can be used in the area of Cloud computing without any installation of software. The users can access the Internet and send messages anywhere in the world [5]. Google provides an efficient and scalable solution for working large-scale data [3]. Centralized storage, processing, memory and bandwidth are more efficient computing that are allowed in Cloud computing. The success and importance of clouds has been driven in part by the use of virtualization as their underlying technology [6].

Cloud helps consumers to outsource computation, storage and other tasks to third party cloud providers and pay for the resources used. [5]. And learn about how a cloud service provider is multiplexing its virtual resources. Thus a cloud model is expected to have a scale up and down in order to manage the load variation[2]. The grid environment consists of several components such as schedulers, load balancer, grid broker, and portals. The schedulers are the types of applications that are responsible for the management of jobs, such as allocating resources needed for any specific job, partitioning of jobs to schedule parallel execution of tasks, data management [8].

Chen He et al [7] focus the problem of decreasing data transmission in a MapReduce cluster and we implement a scheduling technique which can improve map tasks’ data locality rate. When a job is submitted on the clouds, then usually it is divided into several tasks. Following issues are need to attempt when applying parallel processing in executing these tasks:

1. In what way location of resources to tasks can be done.
2. In which order Tasks are executed in the cloud.
3. In what way to schedule overheads when VMs prepare, terminate or switch tasks.

Task scheduling and allocation of resources can solve these three issues [4]. Modern internets applications are critical software implemented on multi-tier architectures[10]. The task scheduling application is not appropriate for the use in a cloud computing domain due to the small particles. The MapReduce cluster, data are distributed to individual nodes and stored in their disks. To execute a map task on a node, it need to first have its input data available on that node [7]. Haozheng Ren et al [9] algorithm which balances the load among the resources and also increases the reliability of the grid environment. This algorithm used to reduce the Communication overhead and reply time of the individual job and increase the throughput of the entire system. [8] this algorithm has two parts. In the first part, the resources can be chosen based on the user deadline and fault tolerant factor of the resources [18-20]. In the second part, the load balancing algorithm is used to find the status of the selected resources and then the jobs are scheduled to the resources if and only if the load is balanced.
II. HISTORY & BACKGROUND

Resource allocation is the emerging topic in cloud computing and so many researchers already concentrated on this topic. Most of the famous researches are presented here.

Prasenjit Kumar Patra et al [11] have proposed that the result of evolution of on demand service in computing paradigms of wide reaching distributed computing. It was accommodate technology as it provided integration of software and resources which are dynamically scalable. Those systems were more or less prone to failure. Fault tolerance evaluates the ability of a system to respond gracefully to an unexpected software or hardware failure. In accordance to achieve robustness and dependability in cloud computing, failure should be assessed and handled effectively. Chandrashekar S.Pawar and Rajnikant B. Wagh [12] have proposed an algorithm which considered Preemptable task execution and multiple SLA parameters such as network bandwidth, memory and needed CPU time. Their experimental results showed that in a situation where resource contention is fierce our algorithm provides better utilization of resources.

Dorian Minaroli and Bernd Freisleben [13] have proposed the distributed version of the resource manager having of several ANNs in which the ANN has responsible for modeling application performance and power consumption of a single VM while exchanging information with other ANNs to coordinate resource allocation. Wang Xue-iy et al [14] have proposed the cloud environment for resource allocation. First, the system frameworks of resource allocation mechanism were constructed, and then tender descriptions were given for four kinds of common resources. Second, a support vector regression (SVR) based method was adopted to convert a combination resource demand into multiple single resource demands. Third, emotional parameter was introduced to bidding strategy and tender assessment mechanism was planned to cause an optimum tender of cloud resource provider. Goudarzi H et al [15] have proposed a resource assignment difficulties which was considered to reduce the total energy cost of cloud computing system which meets the identified client-level SLAs in a probabilistic sense. The cloud computing system paid penance for the percentage of a client’s requests which does not meet an identified upper bound on their service time. A systematic heuristic algorithm based on convex optimization and dynamic programming was extended to effectively deal the above stated resource assignment difficulties.

Doaa M. and Shawky[16] have proposed that the cloud resource were distributed and allowed on demand. That flexible qualities of resource distribution enables to the “pay as you go” concept. Thus it forms the bigger advantage of choosing cloud-based solutions. There was required to assess the performance of the resource distribution algorithm chose by cloud computing environments. Kumar A et al [17] have put forwarded a well organized framework named called EARA (Efficient Agent based Resource Allocation) for resource assignment based on agent computing on SaaS level in Cloud Computing. EARA has five discrete agents, every agent provided with functionality to gather data concerning each resource accessible in actual cloud deployment based on signed SLA agreement, and then replied to the user with appropriate allocation or response code. From the history and background we observed that the users wish to minimize their payment when guaranteeing their service level such that their job would be finished during the period of time which is before the time limit. Such a time limit fulfill resource allocation with reduced payment. More ever inevitable error in predicting task work load will definitely make the problem harder.

III. DESIGN ISSUES

Cloud computing offers business people to flake up and down their resource running based on requirements. Most of the attempts benifits in the cloud model arrive from resource multiplexing between virtualization technology. A structure that holds virtualization technology to assign data center resources dynamically formed on requisition calls and bear green computing by optimizing the number of servers in use. Nowadays cloud computing is emerging Technology. It is used to access anytime and anywhere through the internet. Hadoop is an open-source Cloud computing platform that implements the Googletm MapReduce framework. Hadoop is framework for distributed processing that deals with big datasets among huge clusters of computers. The workload of tasks in clusters mode using Hadoop. MapReduce is a programming approach in hadoop to manipulate for continuing the workload of the tasks. Depend on the task inspection statistics the future workload of the cluster is estimated for potential performance minimization by the use of genetic algorithm.

Resource scheduling algorithm for ForCES (Forwarding and Control Element Separation) networks require to arrange the programmability, flexibility and scalability of node resources. DBC (Deadline Budget Constrain) algorithm relies on users choose cost or time priority, then scheduling to assemble the needs of users. But this priority strategy of users is very poor and simple, and cannot alter to dynamic change of resources, it is inevitable to reduce the QoS. In order to improve QoS, we introduce the economic model and resource scheduling model of cloud computing, support SAL (Service Level Agreement) as pricing strategy, on the idea of DBC algorithm, propose an DABP (Deadline And Budget Priority) algorithm, the survey results show that DABP algorithm construct the job fullfilled with reduced cost during the time line, benifits to load balancing of ForCES networks[21].
IV. COMPARISON OF DIFFERENT RESOURCE ALLOCATION METHODS

Cloud computing is an attractive computing model since it allows for the provision of resources on-demand. Such a process of allocation and reallocation of resources is the key to accommodating unpredictable demands and improving the return on investment from the infrastructure supporting the Cloud. In Fig. 1 Shown Resource allocation in cloud system. It gives the procedure in processing a task which is represented as $t_i$. Assume the task’s execution times cost on computation and disk processing are considered as 4 and 3 hours, respectively. On receiving the request, the scheduler examines the precollected availability states of all candidate nodes, and approximates the minimal payment of running the job during its time line on all of them that is Step 1 the figure. The host Node p3 described in Fig. 1 that needs the reduced payment will run the job via a customized VM instance with isolated resources that is Step 2 in Fig. 1. The VM will be customized with such a CPU rate e.g., 0.4 Gflops and disk I/O rate e.g., 0.3 Gbps that the task can be completed within its deadline $D(t_i)$=1 hour in the example and the user payment can also be minimized meanwhile. Finally in Step 3, the computation results will be returned to users. Assume there are $n$ compute nodes which are denoted by $p_i$, where $1 < i < n$. As all the resources are managed centrally, then the availability state of each resource within any recent or later period can be estimated prior, for executing any given task with multiple execution dimensions.

For any particular task with $R$ execution dimensions, use $c_i$ to denote the whole set of dimensions and $c(p_i) = c_1(p_i), c_2(p_i), \ldots, c_R(p_i)$; as node $p_i$’s capacity vector on these dimensions [22]. In Fig. 1, for example, node $p_1$’s physical capacity vector is $c(p_1)=$ (CPU = 2.4 Gflops; disk IO = 1 Gbps).

Any user’s task is denoted as $t_i$, where $1 \leq i \leq m$, and $m$ refers to the total number of submitted tasks. Each task has a multidimensional workload vector, denoted by $l(t_i) = (l_1(t_i), l_2(t_i), \ldots, l_R(t_i))^T$, which needs to be completed before the task’s deadline. The resource vector allocated to $t_i$ as $r(t_i) = (r_1(t_i), r_2(t_i), \ldots, r_R(t_i))^T$, where $r_k(t_i)$ denotes the resource amount on kth execution dimension isolated by hypervisor/virtual machine monitor (VMM) for the task’s execution. Node $p_i$’s availability vector (denoted $a(p_i)$) along the multiple dimensions is calculated by $c(p_i)-l(t_i)$ running on $p_j$ $r(t)$. For example, node $p_1$ in Fig. 1 is running two VMs that are allocated with half of the total physical resources, since its availability vector $c_1(p_1)=$ (CPU = 1.2 Gflops; disk IO = 0.5 Gbps). Assume there are no workloads being executed simultaneously for a particular task, its total execution time will be the sum of the individual processing times on different dimensions. Assume the execution of the workloads overlap, however, the task’s completion time would be shorter.

On any cloud system, the resources provisioned are usually set with a price vector denoted as $b(p_i) = (b_1(p_i), b_2(p_i), \ldots, b_R(p_i))^T$ along $R$ dimensions. $b_k(p_i)(1 < k < R)$ denotes the per-time-unit price that the consumers need to pay for the consumption of the kth dimension on $p_i$. Every task $t_i$ is set with a deadline that is denoted by $D(t_i)$ for its execution and the payment is expected to be minimized.

In order to improve the gain and profit, it is intended to propose an approach in cloud computing resource allocation. The primary goal underlying this approach is that effective prioritization method. Initially many users send the service request to the cloud server; the cloud server decides the priority among the different user request using the modified priority algorithm. In few cases, Multiple requests has the same priority in that time cloud server has a problem to select the user request. To solve the problem, in proposed method group the same priority request and then find the load value of every request in that group using the modified load balancing algorithm. In proposed method, in the beginning we find the VM (Virtual Machine) list in each cloud and available resources. After the list of appropriate VMs is found, the modified load balancing algorithm decides which particular VM is allocated to service request in order to balance the load. The request with lowest load requires get highest priority between the same group. Since no VM is formed in any cloud, then the request is stored in the queue for a certain period of time. After a certain period of time, the service requests can be scheduled and deployed, then the scheduler returns a scheduling success to the cloud admin, otherwise it returns failure. If the final two priority list is ready then we apply the Improved Particle Swarm Optimization (IPSO) algorithm for selecting the best priority request in both the list.
V. CONCLUSION

In this paper, we propose a novel resource allocation strategy for cloud systems that supports the load balancing algorithm that decides which particular VM is allocated to service request in order to balance the load. The aim of the system is to minimize user’s payment on his/her task and also guarantee its execution deadline. The best priority request is send to the cloud administrator for the allocation of resources and gets the high gain and profit.

REFERENCES


