



An Approach to Optimized Resource Scheduling using Task Grouping in Cloud

Jignesh Lakhani

Asst. Prof. CE Dept
MEFGI, Rajkot, Gujarat, India

Hitesh A. Bheda

Asst. Prof. CE/IT Dept
SOE, RK University, Rajkot, India

Abstract— Cloud computing refers to Internet based development and utilization of computer technology, and hence, cloud computing can be described as a model of Internet-based computing. Scheduling is a critical problem in Cloud computing, because a cloud provider has to serve many users in Cloud computing system. So scheduling is the major issue in establishing Cloud computing systems. The main goal of scheduling is to maximize the resource utilization and minimize processing time of the tasks. In this thesis, an efficient task-grouping based approach has been proposed for task scheduling in computational cloud. Proposed work is grouping the tasks before resource allocation according to resource capacity to reduce the communication overhead. Cloud Resources are heterogeneous in nature, owned and managed by different organizations with different allocation policies. In our scheduling algorithm tasks are scheduled based on resources computational and communication capabilities. Here tasks are grouped together based on the chosen resources characteristics, to maximize resource utilization and minimize processing time and cost. Task scheduling is a decision process by which tasks are assigned to available resources to optimize various performance metrics. Hence in this thesis, we have specifically focused on improving computational cloud performance in terms of total processing time and total processing cost and reduce communication overhead. A simulation of proposed approach using CloudSim toolkit is conducted. Experimental results show proposed algorithm performs efficiently in computational cloud environment.

Keywords— Cloud Computing, Scheduling, task grouping, communication overhead, computation overhead, CloudSim

I. INTRODUCTION

As the IT technologies are growing day by day, the need of computing and storage resources are rapidly increasing. To invest more and more equipments is not an economic method for an organization to satisfy the even growing computational and storage need. So Cloud Computing has become a widely accepted paradigm for high performance computing, because in Cloud Computing all type of IT facilities are provided to the users as a service. Cloud computing is a category of sophisticated on-demand computing services initially offered by commercial providers, such as Amazon, Google, and Microsoft.

In Cloud Computing the term Cloud is used for the service provider, which holds all types of resources for storage, computing etc. Mainly three types of services are provided by the cloud. First is Infrastructure as a Service (IaaS), which provides cloud users the infrastructure for various purposes like the storage system and computation resources. Second is Platform as a Service (PaaS), which provides the platform to the clients so that they can make their applications on this platform. Third is Software as a Service (SaaS), which provides the software to the users; so users don't need to install the software on their own machines and they can use the software directly from the cloud. Due to the wide range of facilities provided by the cloud computing, the Cloud Computing is becoming the need of the IT industries. The services of the Cloud are provided through the Internet. The devices that want to access the services of the Cloud should have the Internet accessing capability. Devices need to have very less memory, a very light operating system and browser. Cloud Computing provides many benefits: it results in cost savings because there is no need of initial installation of much resource; it provides scalability and flexibility, the users can increase or decrease the number of services as per requirement; maintenance cost is very less because all the resources are managed by the Cloud providers [1].

A. Motivation

To realize the full potential of cloud computing, cloud middleware needs to support various services such as security, uniform access, resource management, task Scheduling, application composition, and economic computation. Though, a range of essential services are to be integrated to accomplish a real cloud environment, among them scheduler is one of the most critical service component of the cloud middleware. Since, it is responsible for selecting best suitable virtual machines or computing resources with a goal of maximizing resource utilization and scheduling tasks, in a manner that meets user and task requirements, in terms of overall processing time, processing cost or any other constraints imposed upon by the user. Various scientific and business organizations tend to have increased number of applications with large number of independent tasks, scheduling of these tasks onto the cloud is significantly more difficult and complicated than

scheduling applications in traditional supercomputer because of the heterogeneous, dynamic and diverse nature of the resources. Therefore, optimal scheduling of various tasks onto cloud is not easy to attain, since optimal scheduling of heterogeneous tasks in heterogeneous environments is known to be NP-Complete problem [2]. In order to ensure the efficiency and better performance of task scheduling, an effective and near optimal scheduling mechanism has to be developed and implemented to cater the needs of the cloud users. In traditional distributed computing system, the communication cost is considered to be insignificant as homogeneous computing nodes are interconnected. Therefore, the communication cost has become a decisive factor of performance measurement and must be taken into consideration while scheduling tasks onto the cloud. When a task is submitted to a cloud resource for execution, it is transmitted over the networks incurring problem of communication cost [2]. An application with large number of tasks when submitted individually to the cloud resources over the networks incurs a communication overhead that is more than the total computation time of each task at the resource. Moreover, this also leads to more communication cost problem, not get to high performance and not better utilization of the resources. Therefore, tasks can be grouped at the scheduling level according to the processing. Thus, here must take these properties of the cloud environment into account to make cloud services and cloud-oriented applications efficient. By efficient, we mean appropriate resources are allocated at a right time to a right task, so that task can utilize the resources effectively. In other words, I want to reduce communication overhead problem and better utilization of the resources in cloud computing.

B. Problem Definition

In cloud computing efficient task scheduling will help better utilization of the resources and achieve high performance. Task grouping strategy is also effective technique for solving task scheduling problem in cloud computing. Task scheduling problem can be solve by using task grouping strategy, Prioritization approach, using some heuristic based approach and static and dynamic scheduling techniques help to solve task scheduling problem. task grouping strategy is very effective technique to reduce communication overhead problem in task scheduling and better utilization of the resources on cloud.

C. Scheduling Model in Cloud

In cloud computing, applications are submitted for use of cloud resources by users from their terminals. The resources include computing power, communication power and storage. An application consists of number of tasks; users want to execute these tasks in an efficient manner. There are two possibilities of submission of tasks/data on resources; in one of them, task is submitted on the resources where the input data is available and in the other, on the basis of specific criteria, resource is selected on which both task and input data are transferred and where the task is submitted on a scheduler and data on a resource identified by the scheduler [3,4].

Show the figure 2-6 general step of scheduling a task in cloud computing. In this figure the main component is scheduler, information repository, resources. In cloud computing a user submits tasks to schedulers. Scheduler is connecting to the information repository it's call cloud Information services. Which have all information about to the cloud resources and it also have some virtual privet network for any other don't access to database. Scheduler is receiving tasks from user then scheduler is arranging the tasks according to criteria of tasks [5].

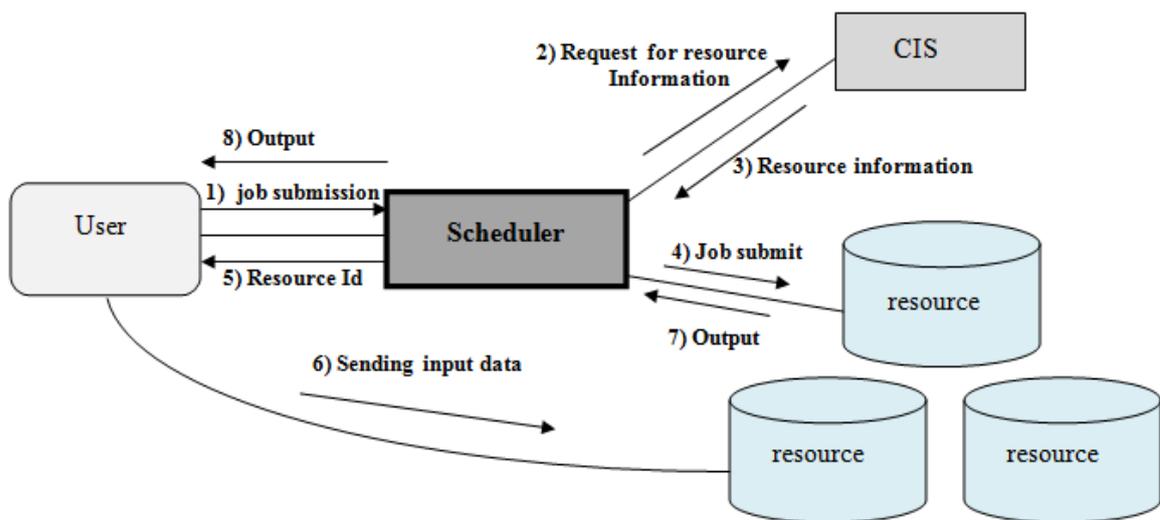


Fig. 1 Scheduling Process in Cloud

Scheduler connects to CIS and gets information about the available resources and then scheduler compute to the resources. After compute the resources, tasks are submit on that resource which have high processing capabilities. Then resource id is generated by CIS and id sent to user. So the resource is communicated to the user. The user then submits the input data or tasks directly on that resource and finally user gets the output from the resource trough the scheduler. This is general process of scheduling [6, 7, 8].

II. PROPOSED SYSTEM

In cloud computing environments, there are two players: cloud providers and cloud users. On one hand, providers hold massive computing resources in their large datacenters and rent resources out to users on a per-usage basis. On the other hand, there are users who have applications with fluctuating loads and lease resources from providers to run their applications. First, a user sends a request for resources to a provider. When the provider receives the request, it looks for resources to satisfy the request and assigns the resources to the requesting user, typically as a form of virtual machines (VMs). Then the user uses the assigned resources to run applications and pays for the resources that are used. When the user is done with the resources, they are returned to the provider. Proper scheduling is needed to meet user's requirements and satisfy the quality of services. So in this use grouping based scheduling tasks in cloud computing. Grouping means collection of components on the basis of certain behavior or attribute. By task grouping in cloud it is meant that tasks of similar type can be grouped together and then scheduled collectively.

Grouping strategy is very effective technique to solve the task scheduling problem and also better utilization of the resources. you can also apply grouping strategy on the resources means resources grouping strategy to solve scheduling problem. But here apply Tasks grouping strategy for solving scheduling problem. Grouping strategy is based on processing capability, bandwidth, and memory size those characteristics of resources and also applies grouping strategies: Grouping by deadline, location, participant, and role. Here describes grouping based Task Scheduling model, Scheduler and Task Grouping algorithm and its description.

A. Proposed Task Scheduling Model

The four basic building blocks of Cloud model are user, scheduler, Cloud Information System (CIS) and resources. User tasks submitted to the scheduler for scheduling to the resources with an objective of minimizing the processing time and utilizing the resources effectively. The scheduling framework illustrated in Figure.4-1 the design of the scheduler and its interactions with other entities. The scheduler is a service that resides in a user machine. When the user creates a list of tasks in the user machine, these tasks are sent to the scheduler for scheduling. The scheduler obtains information of available resources from the Cloud Information Service (CIS). Based on this information, the task scheduling algorithm is used to grouping the tasks and then resource selection for grouped tasks. When all the tasks are put into groups with selected resources, the grouped tasks are dispatched to their corresponding resources for computation by the dispatcher.

The Cloud Information Service (CIS) provides information about all the registered resources in a cloud. This service keeps track of all of the resources characteristics in the cloud. CIS collects resource characteristic information like operating system, system architecture, processing capability, network bandwidth and processing cost. It also provides users the availability information of the resources. The information collector collects information from the Cloud Information Service (CIS). It assembles the resource availability and processing capability to the resource information table. It also gathers information of the network bandwidth and processing cost of each listed resource provided by the CIS. The information collector is used by the grouping and resource selection service to gather necessary information to perform task grouping.

The grouping and resource selection service is responsible for grouping of task based on information collected by the information collector from CIS. In the task grouping process, user submitted tasks are collected by scheduler and tasks are grouped based on the selected available resource characteristics. The process iteratively performed until all the tasks are grouped according to corresponding resources.

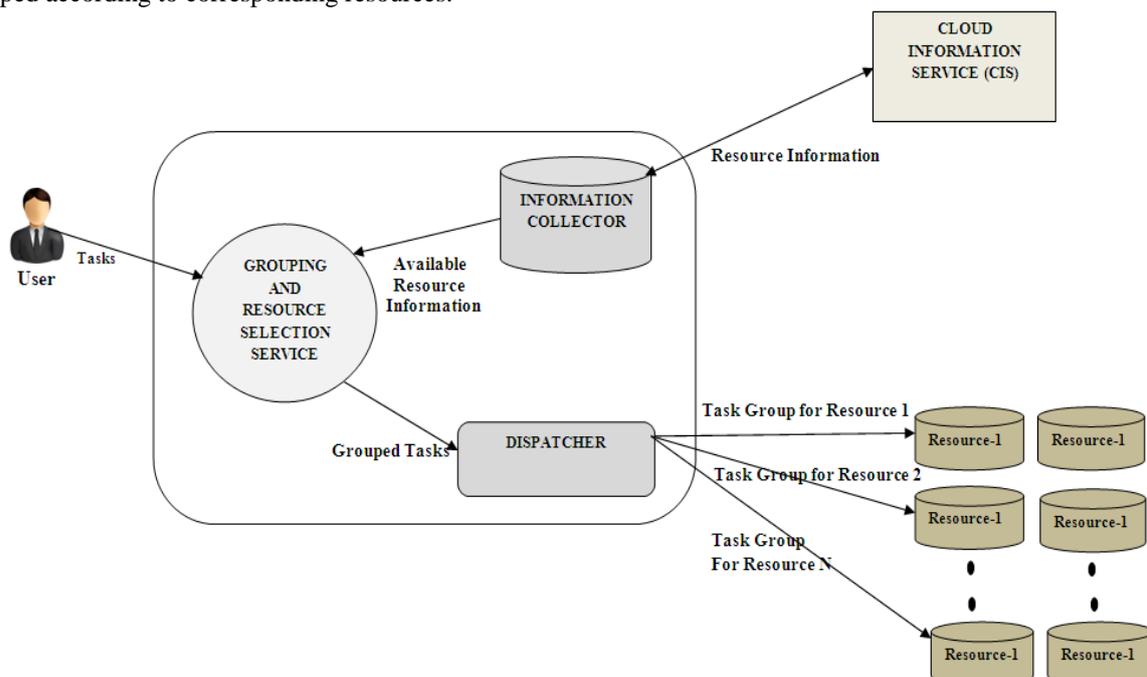


Fig. 2 Scheduling Model for Task Grouping.

The dispatcher acts as a sender that sends grouped tasks to their respective resources. The dispatcher forwards the grouped tasks based on the schedule made by the grouping and resource selection service. The dispatcher also collects the results of the completed tasks from the resources.

B. Architecture of Scheduler

The architecture of the scheduler system is described in Fig 4-2. The system accepts tasks from the users specified by their TASK ID, TASK LENGTH (in Million Instructions (MI)), TASK INPUT FILE SIZE (in Mb) and total number of tasks submitted by the user. After gathering details of user tasks, scheduler collects all the available computational resources information specified by their RESOURCE ID, RESOURCE MIPS (computational power of the resource in Million Instructions per Second), RESOURCE BANDWIDTH (in Mb/sec.), and RESOURCE COST (in cost/sec).

After gathering the details of user tasks and the available resources, the scheduler will select a resource and multiplies the resource MIPS with the given granularity time, which is the time within which a task is processed at the resource. The value of this calculation produces the total Million Instructions (MI) for that particular resource to process within a particular granularity time. The system selects tasks in first-come first-serve (FCFS) order, and then tasks are grouped based on the resulting total MI of resource and bandwidth. New IDs are assigned to grouped tasks and scheduler submits the task groups to their respective resources for computation. After executing the group task, results goes to back to the corresponding users and the resource is again available to scheduler system.

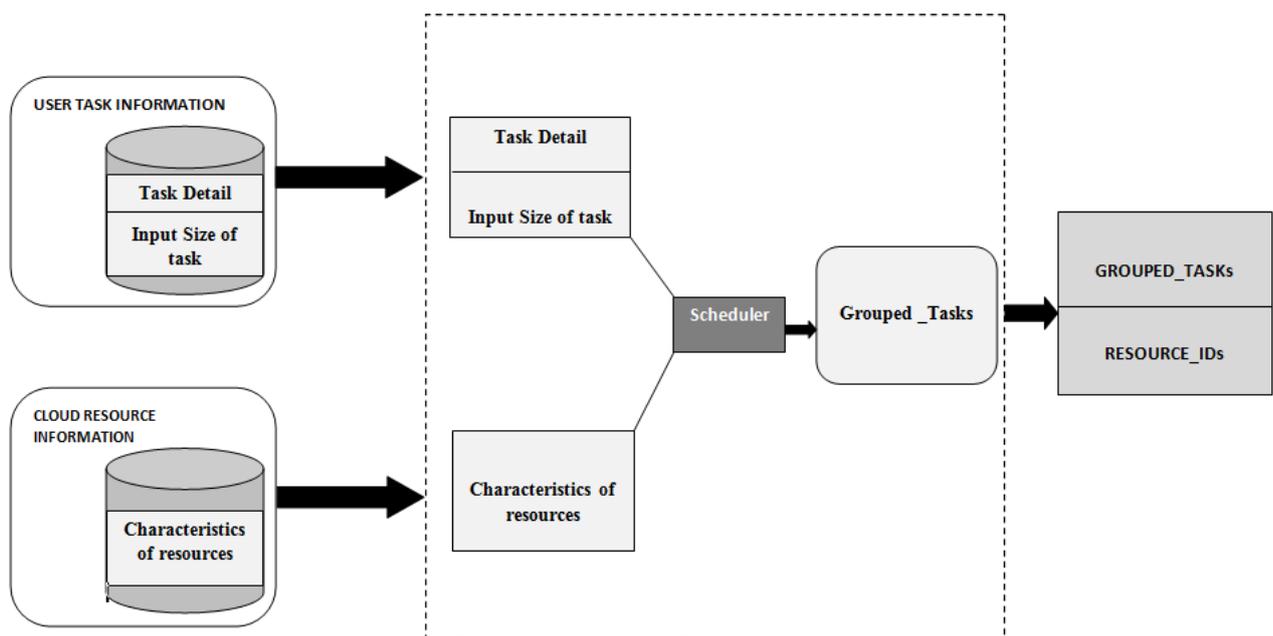


Fig.3 Architecture of Scheduler

C. Proposed Algorithm

The scheduler accepts number of tasks, average MI of tasks, deviation percentage of MI granularity size and processing overhead of all the tasks. Resources are selected. Now job grouping algorithm is applied to the FCFS order to allocate the task groups to different available resources.

- n : Total number of task
- m : Total number of Resources available
- MIPS : Million instructions per second or processing capabilities of a resource
- MI : Million instructions or processing requirements of a user task
- Tot-length : Total processing requirements (MI) of a task group (in MI)
- Tot-GMI : Total length of all tasks

Algorithm: Task grouping and scheduling algorithm

- Step 1: The scheduler receives Number of tasks 'n' to be scheduled and Number of available Resources 'm'
- Step 2: Scheduler receives the Resource-list R[]
- Step 3: The tasks are submitted to the scheduler
- Step 4: Set (Sum of the length of all the tasks to zero
- Step 5: Set the resource ID j to 1 and the index i to 1
- Step 6: Get the MIPS of resource j
- Step 7: Multiply the MIPS of jth resource with granularity size specified by the user

- Step 8: Get the length (MI) of the task from the list
Step 9: If resource MIPS is less than task length
 9.1 : The task cannot be allocated to the resource
 9.2 : Get the MIPS of the next resource
 9.3 : go to step 7
Step 10: If resource MIPS is greater than task length
Step 11: Execute steps 11.1 to 12 while Total length of all tasks is less than or equal to resource MIPS and there exists ungrouped tasks in the list
 11.1: Add previous total length and current task length and assign to current total length
 11.2: Get the length of the next task
Step 12: If the total length is greater than resource MIPS.
 12.1: subtract length of the last task from Tot-length
Step 13: If Tot-length is not zero repeat steps 13.1 to 13.4
 13.1: Create a new task-group of length equal to Tot-length
 13.2: Assign a unique ID to the newly created task-group
 13.3: Insert the task-group into a new task group list
 13.4: Insert the allocated resource ID into the Target resource list of each grouped job
Step 14: Set Tot-GMI to zero
Step 15: get the MIPS of the next resource
Step 16: Multiply the MIPS of resource with granularity size specified by the user
Step 17: Get the length (MI) of the task from the list
Step 18: go to step 9
Step 19: repeat the above until all the tasks in the list are grouped into task-groups
Step 20: When all the tasks are grouped and assigned to a resource, send all the task groups to their corresponding resources list of Grouped task
Step 21: After the execution of the task-groups by the assigned resources send them back to the Target resource list.

III. CONCLUSION

Scheduling is a critical problem in Cloud computing, because a cloud provider has to serve many users in Cloud computing system. So scheduling is the major issue in establishing Cloud computing systems. In this paper we have discussed about the problem of task scheduling in computational Cloud, where user submits tasks (requires small processing requirement) and we have tried to find a solution for that problem. We have proposed an efficient task-Grouping Based Scheduling algorithm. it is clear that proposed approaches reduces the total processing time and processing cost and also reduce communication overhead.

It is clear that the total time highly depends number of Cloudlets, Cloudlet length (MI), resource MIPS, and Cloudlet overhead processing time. Time for Large number of Cloudlets (or large the Cloudlet length) is larger. The execution time can be decreased by performing Cloudlet grouping method where large number of Cloudlets are grouped into a few Cloudlet groups, and sent to the Cloud resources. Cloudlet grouping reduces the transition time of each Cloudlet to the resource, and the overhead processing time of each Cloudlet at the resource. In addition, Cloudlet grouping method will fully utilize the processing capabilities of the resources since the Cloudlet grouping is done based on the processing capability of each resource. Another way to reduce the simulation time is to increase the number of resources used to execute the Cloudlets Granularity time is used in the experiments to test the simulation effort in a slightly different way. The purpose is to test how many Cloudlets can be processed by a group of resources within a particular time. Higher the granularity time, higher the number of Cloudlets that can be completed within that granularity time. In short, execution time of a number of Cloudlets can be reduced with higher granularity time, higher resource MIPS, and Cloudlet grouping method.

REFERENCES

- [1] Bhaskar Prasad , Eunmi Choi and Ian Lumb, "A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems", Springer International Conference on Computational Intelligence and Computing Research, ISBN: 978-1-84996-240-7, 2010, pp: 21-46.
- [2] Vishnu Soni,Raksha Sharma,Manoj Mishra, "Grouping-Based Job Scheduling Model In Grid Computing ", World Academy of Science,Engineering and Technology Publishing, Vol. 4(1),issue 5 , 2010, pp: 781-784.
- [3] N.Muthuvelu, J.Liu, N.Soe, S.Venugopal, A.Sulistio and R.Buyya, "A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids", in Proc of Australasian Workshop on Grid Computing and e-Research (AusGrid2005), Vol. 44,2005, pp: 41-48.
- [4] Monika Choudhary and Sateesh Kumar Peddoju, "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment", International Journal of Engineering Research and Applications (IJERA), ISSN: 2248-9622, Vol. 2,Issue 3, May-Jun 2012, pp: 2564-2568.
- [5] O. M. Elzeki, M. Z. Rashad and M. A. Elsoud, "Overview of Scheduling Tasks in Distributed Computing Systems", International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Vol.-2, Issue-3, July 2012, pp: 470-475

- [6] Hai Zhong, Kun Tao and Xuejie Zhanand, “An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems”, “,IEEE in Fifth Annual China Grid Conference , ISSN: 978-0-7695-4106-8, Vol.-5, 2010, pp: 124-129.
- [7] Pardeep Kumar and Amandeep Verma,” Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm”, International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Vol 2, Issue 5, May 2012,pp:111-114.
- [8] Suraj Pandey, LinlinWu and Siddeswara Mayura Guru, “A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments”, 24th IEEE International Conference on Advanced Information Networking and Applications, ISSN: 1550-445X, Vol.-3, 2010, pp: 400-407.