

# Resource Allocation Method for Scientific Workflow Executions in Cloud Environment

Kaliki Prasanna Kumar<sup>1</sup>, Anusooya G<sup>2</sup>

<sup>1,2</sup>School of Computer Science Engineering, VIT University, Chennai, India-600127

**Abstract:** *Cloud computing is the latest trend in large scale distributed computing. It provides diverse services on demand to distributive resources like servers, softwares, databases, etc. Energy efficiency is the main contributor in future cloud data centers. There is need to growing implement energy efficient resource allocation as green computing is becoming vital reason for todays organization. One of the challenging problem is to deploy scientific workflow execution with less energy consumption. For single scientific workflow executions some algorithms are implemented. The objective of this paper is implement an application to allocate resources for scientific workflow executions in cloud environment. It includes partitioning, migration techniques also. We done dynamic deployment of scientific workflows in cloud environment.*

**Keywords:** *Scientific workflows, energy efficient, resource allocation.*

## I. INTRODUCTION

Cloud computing has emerged as one of the premier technologies that has revolutionized the way services are made available to users using cloud data centers. Cloud computing has driven the creation of data centers made up of more than thousand servers that are capable of supporting large number of services. The cost of maintaining this cloud data centers is extraordinarily high. A recent study has shown that power and cooling costs of data centers have increased immensely by 800 percent since 1992. Hence, there is a need to introduce resource allocation techniques that efficiently utilize the servers and in turn help in energy efficiency. Data centers are one of the main backbones of cloud computing. Clouds spans over the data centers and works with the notion of location independence. Cloud computing achieves its two property of dependableness and redundancy using multiple data centers. Another important supporting feature for cloud computing is its network infrastructure. With proper networking strategies a way of green computing can be achieved. Even though cloud does not care about a particular locality, it gives importance to relative network locality and topology.

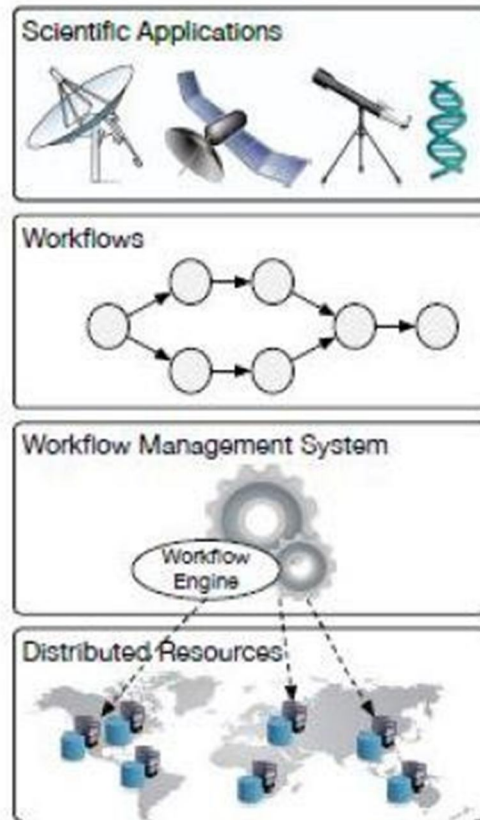
Cloud resource allocation is used in the cloud computing environment to distribute resources among different computing resources. Resource allocation is having greater importance in cloud data centers as it dynamically allocates the resources among the data centers and thereby meets the increased service demands. Resource allocation is one of the main problems of cloud computing. Resource allocation is a mechanism for distributing the resources equally across all the virtual servers inside the cloud setup. It helps in continuous service provision by distributing the requests among the instances without failing. Resource allocation will increase the overall performance of the system together with its resource utilization property. It will additionally avoid the case where some nodes are heavily loaded whereas some other nodes are doing little work. This feature of resource allocation helps to attain high system improvement and user satisfaction. It additionally helps in reducing energy usage because the distribution is managed efficiently among the server instances.

## II. BACKGROUND WORK

### A. Scientific Workflow

Scientific workflow contains so many tasks. It requires so many resources to complete these tasks. Each task in scientific workflow is depend on other tasks and communicate with other tasks. These scientific workflows consume more energy.

Examples for scientific workflows are DAG and kepler etc. Scientific workflows are split into many tasks and these tasks are processed and finally merged. Scientific workflow management system are designed to automatically spread data for large applications. Infrastructure-as-a-service provides virtual infrastructure on pay-as-use basics. This feature is useful to deploy scientific workflows. The best example for scientific workflows in cloud is amazon EC2. All workflows are executed using Pegasus. There are some workflow engines also there to manage process of business.

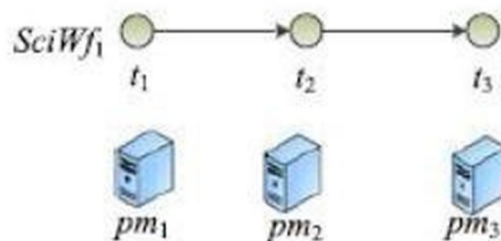


### B. Single Scientific Workflow

Firstly we take a single scientific workflow *sciwf1*. This scientific workflow contains three tasks  $t_1, t_2, t_3$ . These tasks are executed sequentially. For execution of these three tasks there are three primary machines as resources. We allocate these three tasks in two different ways.

$T_1$  to  $pm_1, t_2$  to  $pm_2, t_3$  to  $pm_3$

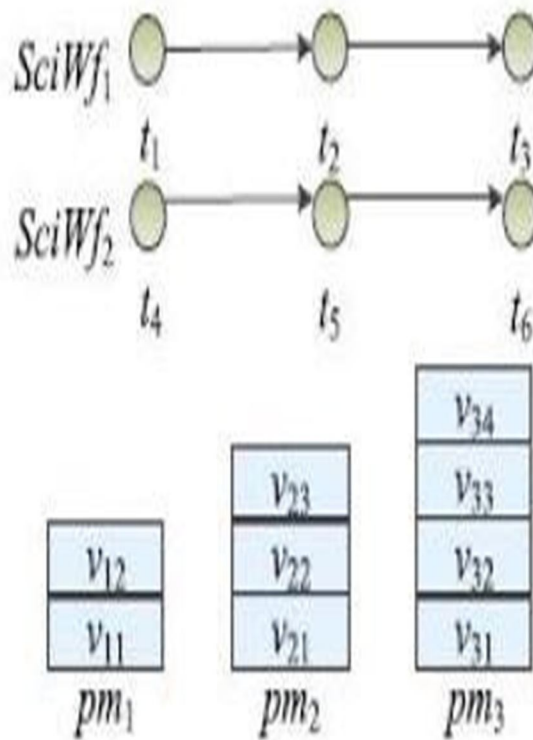
$T_1, t_2, t_3$  to  $pm_1$



Take energy consumptions of three resources are 200w, 250w, 300w. According to first resource allocation policy, the energy consumption is  $(200+250+300)*1=750w$ . For second resource allocation policy, the energy consumption is  $200*(1+1+1)=600$ . First resource allocation policy is better.

### C. Multiple Scientific Workflow

Let us consider, multiple scientific workflows *sciwf1, sciwf2*. These two are should be executed simultaneously. Virtualization concept is used here. With virtualization, we can execute simultaneously. Virtualization is one of the tremendous concept in cloud. We take 2 VMs for  $pm_1, 3$  VMs for  $pm_2, 4$  VMs for  $pm_3$ . Here is a problem that to find energy efficient resource allocation policy.



#### D. Migration Types

There are so many types of migration. The type of migration depends on whether VM has state or not. If the VM has no state then its easy to migrate. State means whether VM contains any information about particular session, is that VM contains any database or any other servers like mail etc. If VM doesn't has state, then we terminate the existing server and provide new server. This process is easy while compare to other migration techniques. If the VM has state then first turn off that VM and copy entire data then allocate new server. In our paper we use live migration technique. It doesn't depend on application is running or not. While migrating the VM it doesn't affect the running application. All hypervisors support live migration technique. Pacer is migration technique. It rectifies all problems which are occurred in other migration techniques mentioned above. It tells how much time taken for migration. If application accessed by several data centers then pacer can migrate easily. Pacer can be used for any type of application.

### III. PROPOSED SYSTEM

#### A. Start Time Partition

In this part, we consider a set of tasks  $WT = wt_1, wt_2, \dots, wt_n$  contained in the scientific workflows. These workflows are deployed in clouds dynamically. For executing of these tasks require certain amount of resources. The required resources are specified by number of VMs. Each PM consists some VMs with configurations. For executing these tasks, requested start time and requested duration time also required. Every task request is defined by tuple of five attributes.

$R_n = f_{widn}, tidn, stn, dtn, nmng$

Widn = workow sequence number

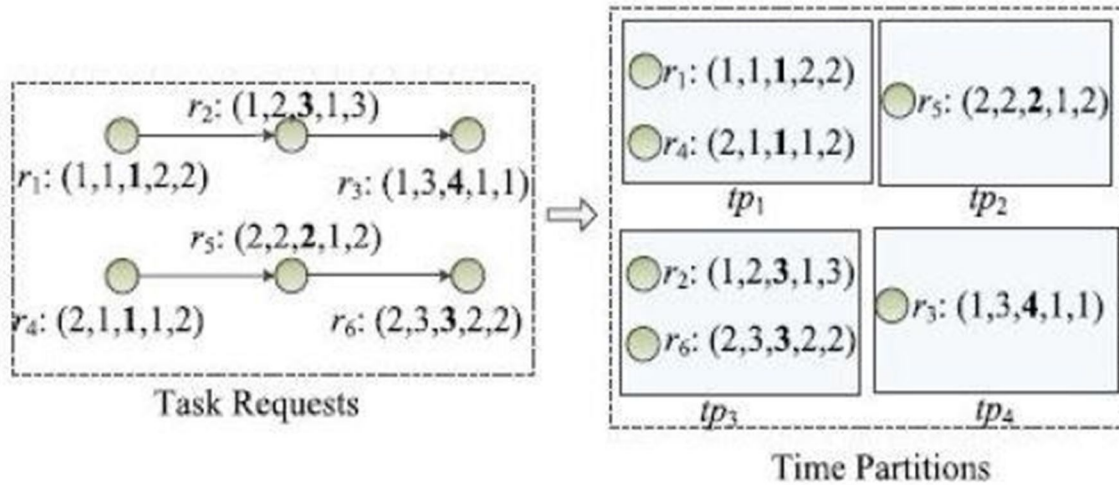
Tidn = task sequence number

Stn = requested start time

Dtn = requested duration time

Nmn= required number of VMs

For arrival of task requests, the allocation order is decided by requested start time. Task requests with minimum start time are assigned firstly. Task requests are sorted in increasing order for better response. The following figure explains how the partition happening.



There are six task requests  $r_1, r_2, r_3, r_4, r_5, r_6$ . Each request has different requested start time. We separate the task requests with same requested start time. Finally  $r_1, r_4$  belongs to one partition  $tp_1$ ;  $r_5$  belongs to  $tp_2$ ;  $r_2, r_6$  belongs to  $tp_3$ ;  $r_3$  belongs to  $tp_4$ .

The task request divided into set of partitions.

$TP = \{tp_1, tp_2, tp_3, tp_4\}$

### B. PM Resource Monitoring

After allocating resources for task requests, its better to check unused VMs on PMs. For detecting unused VMs resource monitoring is compulsory. For resource monitoring, we have to create Resource Allocation Table(RAT). Using RAT we can keep track of resources. RAT contains several attributes.

$Rarm, i = (r_{trm}, i, r_{sm}, i, r_{stm}, i, o_{tm}, i)$

$Rarm, i$  =  $i$ th allocation record

$R_{trm}, i$  = respective task request

$R_{sm}, i$  = resources set

$R_{stm}, i$  = requested start time

$O_{tm}, I$  = occupied time

The resource allocation table is dynamically updated. Through resource allocation table we can count number of idle VMs. These idle VMs are used to next task requests. Every time resource allocation table should be updated.

### C. Migration based Resource Allocation

Migration purely independent on application. A mechanism to move a running application from one PM to other PM. Migration cannot create any disturbance to running application. In this section we can migrate the task requests to idle VMs. For each allocation record contained three tuples.

$Kn, i = \{epn, i, esn, i, eon, I\}$

$Epn, i$  = Employed PM

$Esn, i$  = Start time

$Eon, i$  = Occupied time

The energy consumption for CPU is not linear to the workload. For example the energy consumption at 100 percent CPU load can be much higher than twice of the energy consumption at 50 percent CPU load. Base line energy consumption of a server is the highest part in total energy consumption. We keeping one server in a running state as opposed to keeping two servers in running mode. In this first we allocate VMs for time partitions based on requested time. In this we can migrate VMs from one PM to another PM. If there are two or more PMs are available to satisfy required number of VMs to task request then we allocate task request to the PM which has low energy consumption rate. We continuously checking and allocating the idle VMs.

#### D. Global Resource Allocation with Low Energy

In this we present a web application to allocate task requests with low energy consumption. In this migration also used. Some idle VMs are put in rest mode. Firstly we take the finishing times of time partitions for each task request. At every finish time we are supposed to find how many VMs are to be idle and remaining workload. We apply migration concept here to occupy idle VMs. We calculate energy consumption of VM at each time. If it less than average rate then put that VM into rest mode.

#### IV. CONCLUSION

In this paper, an Energy-mindful Resource Allocation strategy, named En Real, has been proposed, around virtual machines' dynamic sending for logical work process executions. Solidly, a vitality utilization demonstrate has been displayed for the application crosswise over distributed computing stages. Moreover, a vitality mindful asset designation strategy has been intended for virtual machine distribution to bolster logical work process executions. Finally, broad examination assessment utilizing the Cloudsim toolbox has been led, and the outcomes have shown the legitimacy of our technique.

In view of the work done in this paper, we plan to coordinate our vitality mindful planning of logical work process executions into true cloud stages later on. This present reality logical work processes may have their individual attributes, and the proposed strategy ought to be enhanced for meeting their specific needs.

#### REFERENCES

- [1] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1107–1117, Jun. 2013.
- [2] S. Ibrahim, B. He, and H. Jin, "Towards pay-as-you-consume cloud computing," in *Proc. 8th IEEE Int. Conf. Service Comput.*, Jul. 2011, pp. 370–377.
- [3] X. Zhu, L. Yang, H. Chen, J. Wang, S. Yin, and X. Liu, "Real-time tasks oriented energy-aware scheduling in virtualized clouds," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 168–180, Jun. 2014.
- [4] X. Liu, Y. Yang, Y. Jiang, and J. Chen, "Preventing temporal violations in scientific workflows: Where and how," *IEEE Trans. Softw. Eng.*, vol. 37, no. 6, pp. 805–825, Nov. 2011.
- [5] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," *Future Gener. Comput. Syst.*, vol. 29, no. 3, pp. 682–692, Mar. 2013.
- [6] R. N. Calheiros and R. Buyya, "Meeting deadlines of scientific workflows in public clouds with tasks replication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1787–1796, Jul. 2014.