

Optimization Technique For Task Scheduling In Cloud

Chandni

School of Computer Science and Engineering
Lovely Professional University, Phagwara, Punjab, India
chandnikathuria5@gmail.com

Alok Kumar Singh Kushwaha

Department of CSE, IK Gujral Punjab Technical University, Kapurthala, Punjab
dr.alokkushwaha@ptu.ac.in

Abstract:Cloud computing is an innovative way to use standardized IT capabilities such as software, services and various hardware or infrastructural resources. Organizations need to bear high upfront as well as ongoing costs for these IT resources. Idea of cloud computing is to provide these IT resources as utility where access to these resources can be offered ubiquitously through Internet technology. However there are number of challenges associated with this scalable and rapid provisioning of resources to serve multiple consumers. One of biggest challenge is to distribute workload in optimized manner to ensure balanced workload among multiple resource instances. Aim is to improve resource utilization and minimize the cost for consumer as well as provider. In this paper we proposed an optimized technique for workload balancing across different virtual machines.

Keywords:Cloud, Metaheuristic, ACO, Cloudlet, Virtual Machine

1. Introduction

The term “cloud” is metaphor for the “Internet” as it involves providing broad networked access to shared pool of IT resources on pay-per use basis. This remote access to resources may be via private/dedicated network or public network. Traditional computing basically demands all IT resources to be on the premises of consumer organization whereas cloud offers access to different resources as service; accordingly there are three service delivery models: IAAS, SAAS, PAAS. Service model specifies what services are offered by cloud provider to consumer and what needs to be managed by consumer itself. IAAS is service model where access to crucial hardware resources such as storage, network, and computing power is offered on demand. PAAS is an offering where consumers get access to ready-to-use environment/platform to design, test and deploy applications. SAAS offers consumers with access to particular software such as customer relationship management, report editing software as and when required. Deployment of cloud can also vary depending on requirements and budget of consumers. Deployment type actually defines where cloud

infrastructure resides and level of administrative power of consumer and provider. Private deployments are more secure when compared with public but costly at the same time [13,15]. Hence consumers need trade-off depending upon their business priority. Fig.1 depicts how cloud model is not quite the same as traditional computing model

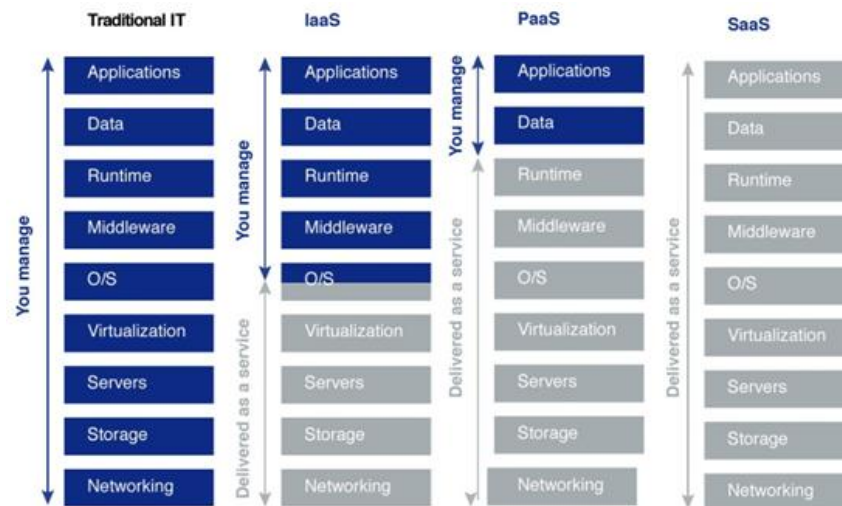


Fig.1 Traditional v/s cloud model

Virtualization is an empowering innovation for cloud computing where numerous virtual instances of the physical resource are created. These virtual instances are then assigned to different consumers for workload execution. Fig.2 portrays two ways for virtualization, type-1 where hypervisor (virtualization software) is installed directly on hardware whereas type-2 involves layer of host operating system between hardware and hypervisor

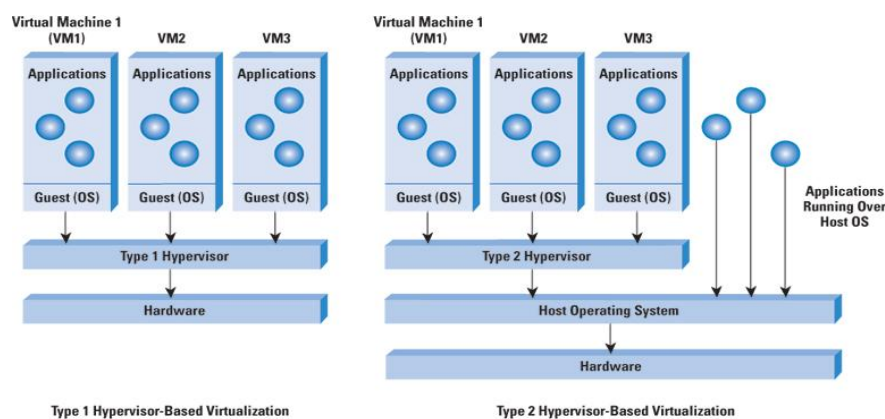


Fig.2 Types of Virtualization technology

1.1. Challenges in Cloud Computing

Despite of benefits such as flexibility, scalability, Resiliency and access to unlimited resources, there are many challenges faced by consumers as well as providers of the cloud services. Table.1 describes few challenges from both perspectives.

Table1: Challenges of Cloud Computing challenges

Challenges	Description
1. Allocation of IT resources	Aim is achieve balanced workload distribution with optimal resource usage
2. Load Balancing	It is related to download performance and Storage utilization. This problem mainly occurs in distributed nodes.
3. Availability and Scalability	Performance degradation and oversizing problems occur due to unpredictable in cloud.
4. Compatibility and Migration	Increasing dependency on particular provider results high locking cost where migration seems impossible.
5. Interoperability and Communication between Clouds	Due to lack of standardization ensuring interoperability is quite challenging.

1.2. Task Scheduling

It is characterized as the procedure which chooses which task/cloudlet ought to be executed on which VM. To accomplish this, a powerful algorithm is required that can provide optimal resource usage with balanced workload across resources i.e. VMs. Each scheduling algorithm is focused to optimize certain metrics [1] from consumer or provider perspective.

Table.2 Optimization metrics for task scheduling

Metric	Description
Makespan	Makespan indicates the finishing time of the last task. The most popular optimization criterion while scheduling tasks is minimization of Makespan as most of the users desire fastest execution of their application. [18] $Makespan = \max_{i \in tasks} \{F_i\}$ where F_j denotes the finish time of task j
Economic cost	It is measure of the cost that consumer will incur for using IT resources from provider. $Economic\ Cost = \sum_{j \in resources} \{C_j * T_j\}$ where C_j measures per unit time cost of resource j and T_j measures usage time of resource j .
Flowtime	It measures total of finishing time of all tasks.

	Flow Time = $\sum_{j \in \text{tasks}} F_j$, Where F_j is finish time of task j
Waiting time	It measures the time gap between submission and execution start Waiting Time _i = $S_i - B_i$ Where S_i and B_i are start time and submission time of task i respectively.
Turnaround time	It measures the time gap between task submission and its completion. [26] TTime _i = $W_i + E_i$ Where W_i and E_i are waiting time and execution time of task i respectively
Resource utilization	Another important criterion is maximization of resourceutilization i.e. keeping resources as busy as possible.
Throughput	Measures count of jobs completed per unit time.[18]
Fairness	It ensures that tasks are scheduled or get resources without starvation.

Rest of the paper is organized as: Section-2 summarizes pioneer methods of cloud task scheduling. Section-3 introduces proposed optimization approach for task scheduling. Implementation details and results are given in section-4 and section-5 finally concludes this work and present future directions

2.Pioneer cloud scheduling methods:

MedhatTawfeek et al. [2] presented approach that use Ant colony optimization technique and outperforms standard task scheduling methods FCFS (First Come First Serve) and RR (Round Robin). Luiz André Barroso et al. [3] proposed approach to optimize energy profile of the system. R. Raja et al. [4] proposed bio-inspired optimization approach for cloud task scheduling that helps to reduce carbon footprint of datacenters. JyotiThaman et al. [5] provides taxonomy of cloud task scheduling as shown in Fig. 4. SrinivasanSelvaraj et al. [6] applied ACO heuristic for task scheduling that outperforms fair scheduling algorithm such as FCFS.Shubham Mittal et al. [7] introduced technique that also takes scalability and distribution into account.Gurtej Singh et al. [8] presented view of different bio-inspired algorithms and has shown how they optimize overall performance metrics.

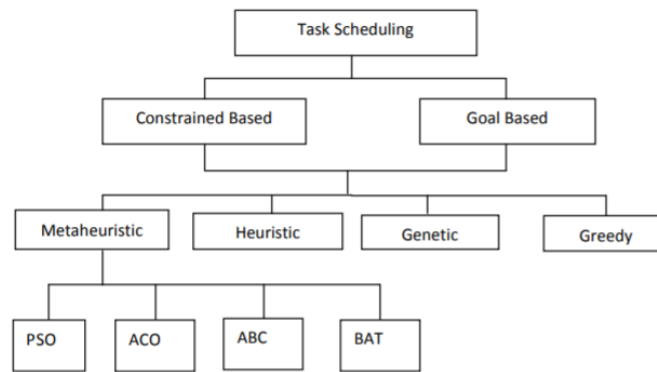


Fig. 3 Taxonomy of cloud task scheduling

Aarti Singh et al. [9] presented agent based load balancing technique that can dynamically balance load across virtual resources. M.A. Vasile et al. [10] proposed novel technique that implements hierarchical clustering of the available resources into groups during allocation.

3. Proposed Approach to scheduling

In this paper, firstly the ACO technique is used for allocation of incoming jobs/tasks to the available set of virtual machines. Workload allocation given by ACO as result then goes under migration. Basic idea is to mitigate the load imbalance by transfer of load from highly loaded machine to the lighter one. The step by step description of proposed methodology is given as Algorithm-1 and Table 3. Gives various notations used in algorithm-1.

Table 3: Notations used in Algorithm

Notation	Description
VM	<i>Virtual machine</i>
VmList	<i>List of virtual machines</i>
VmId	<i>Unique identifier for each VM in the VmList</i>
CIId	<i>Unique identifier of each cloudlet</i>
CLen	<i>Length of each cloudlet</i>
TotLoad	<i>Total length of cloudlets ie. load on particular VM</i>
AvgLoad	<i>Average load of VMs</i>

Algorithm-1:

- i. Start
- ii. $\forall VmId_i \in VmList$, keep track of $CIIds$ assigned to that $VmId_i$ as $CList_i$
- iii. Calculate $TotLoad_i$ for $VmId_i$ for $i=1$ to N , N be the count of VMs in the $VmList$ as:
$$TotLoad_i = \sum CLen_j \text{ for } j=1 \text{ to } n,$$

$$n \text{ be number of cloudlets assigned to each VM ie. } CList_i$$
- iv. Arrange $VmList$ in ascending order based on $TotLoad$ of each VM

v. CalculateAvgLoad as:

$$AvgLoad = \sum TotLoad_i / N \text{ for } i=1 \text{ to } N, N \text{ be the count of virtual machines}$$

v. Apply migration process

a. Start migration of cloudlets from from most stacked VM inVmListto Least stacked VM till the formerTotLoad is greater thanAvgLoad.

b. Delete VMs considered in stepv(a) from the VmList, i.e. least and heaviest loaded

vi. Repeat same procedure until all VM's are removed from the VmList.

vii. Finally put cloudlets for execution

viii.End

4. Experimentation and Results

CloudSim is the simulation framework designed by GRIDS laboratory, it provides programming model that simulates the cloud environment and interaction between different components.

4.1.CloudSim configuration

Proposed methodology is experimented using ten datacenters with fourtyVMs and tasks/cloudlets varying in numbers from 800-2400.The length of the task rangesfrom 20000 MIto 40000 MI(Million Instructions). Thecloudsim configuration used for experimentation is given in table 4.

Table 4: Cloudsim configuartion used to implement proposed technique

Entity Type	Parameters	Values
Task(Cloudlet)	Length of Task	20000-400000
	Total Number of Task	800-2400
Virtual Machine	Total Number of VMs	40
	MIPS	256-512
	VM Memory (RAM)	1024
	Bandwidth	500
	Cloudlet Scheduler	Time_shared and Space_shared
	Number of PEs Requirement	1-2
Data Centre	Number of Datacenter	10
	Number of Host	2
	VM Scheduler	Time_shared and Space_shared

4.2. Results

Makespan and imbalance factor is analysed with CloudSim parameter setting mentioned in Table.2. Experimentation is done by using ACO and proposed methodology for scheduling cloudlets to VMs. Fig.4clearly depicts that makespan of our approach is less as compared to ACO when experimented with fixed number of VMs and varying number of cloudlets.

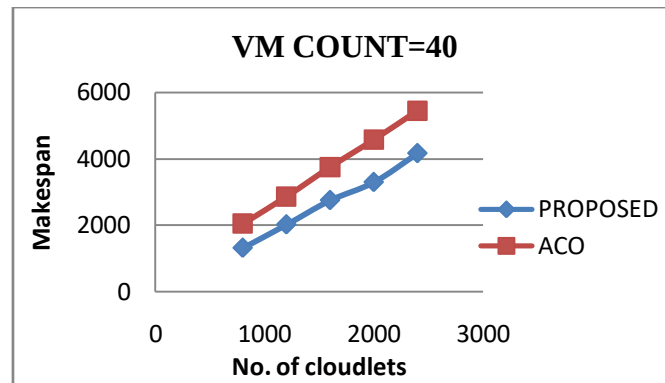


Fig.4Makespan with fixed number of Vms

Similarly Fig.5 shows that makespan of our approach also less as compared to ACO when experimented fixed number of cloudlets (1000)and varying number of Vms.

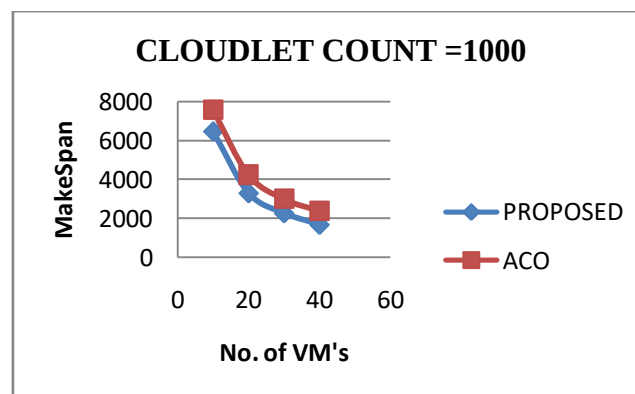


Fig. 5Makespan with fixed number of cloudlets

It can be observed that there is improvement in Makespan in both cases. Makespan refers to the finish time of last cloudlet. The proposed algorithm clearly outperforms ACO when it comes to Makespan. Imbalance factor is also analyzed for both ACO and proposed approach as depicted in Fig. 6and 7.

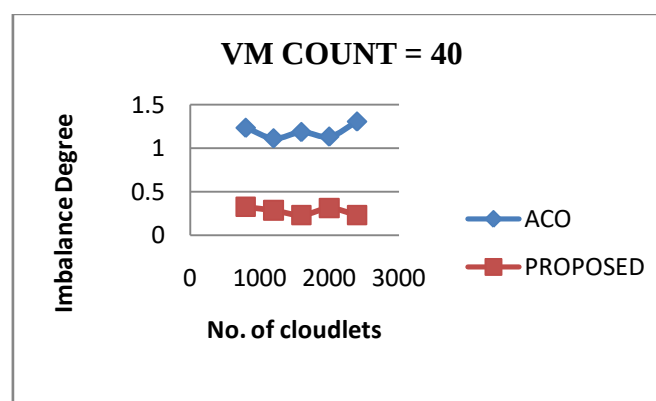


Fig.6Imbalance factor with fixed number of VMs

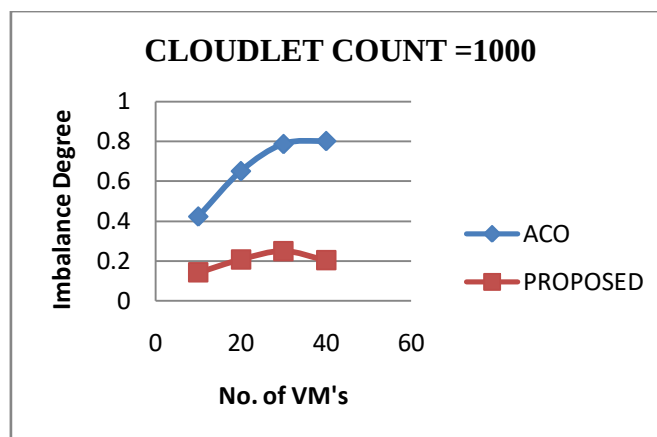


Fig.7 Imbalance Factor with fixed number of cloudlet

The idea of this paper is mainly to improve the Imbalance Degree and the results have shown that the proposed method outperforms the ACO by optimizing imbalance.

5. Conclusion and Future Scope

Proposed approach in connect with ACO heuristic improves the imbalance factor in addition to minimizing makespan. ACO algorithm basically works on random search mechanism. In this algorithm the positive feedback approach is used and follows the behavior of real ant colony in search of food and to connect. It aims to minimize makespan but ACO does very little to improve the imbalance factor. Proposed methodology improves imbalance factor but incurs additional migration overhead. Identification of tasks with least overhead and right resources for its execution is important. Idea is to minimize the no. of tasks to be migrated and also to migrate the tasks with minimum overhead cost. In future parameters like size of the cloudlet, channel band width can also be considered when migration tasks using Max to Min migration.

References

- [1] M. Kalra, S. Singh, "A review of metaheuristic scheduling technique in cloud computing", Cairo University, Egyptian Informatics Journal, vol. 16, issue 3, pp 275-295, 2015.
- [2] M. Tawfeek, A. El-Sisi, A. Keshk, F. Torkey, "Cloud Task Scheduling Based on Ant Colony Optimization", The International Arab Journal of Information Technology, vol. 12, no. 2, March 2015.
- [3] L. Andre, "The case for energy-proportional computing", IEEE Computersociety (2007), pp. 33-37.

- [4] R. Raja, J. Amudhavel, N. Kannan, M. Monisha, “A bio inspired Energy-Aware Multi objective Chiropteran Algorithm (EAMOCA) for hybrid cloud computing environment”, International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, 2014, pp. 1-5.
- [5] J. Thaman, M. Singh, “ Current Perspective in Task Scheduling Techniques in cloud Computing: A Review” , International Journal in Foundations of Computer Science & Technology (IJFCST) vol.6, no.1, January 2016.
- [6] S. Selvaraj,J. Jaquiline, “Ant Colony Optimization Algorithm for Scheduling Cloud Tasks”,International Journal of Computer Technology & Applications, vol. 7(3),491-494,May-June 2016.
- [7] S. Mittal ,A. Katal, “An Optimized Task Scheduling Algorithm in Cloud Computing”, IEEE 6th International Conference on Advanced Computing (IACC), DOI: 10.1109/IACC.2016.45, 2016
- [8] G. Singh, A. Kaur, “ Bio Inspired Algorithms: An Efficient Approach for Resource Scheduling in Cloud Computing” , International Journal of Computer Applications (0975 – 8887) vol. 116 – no. 10, April 2015.
- [9]A. Singh, D. Juneja, M. Malhotra,“Autonomous Agent Based Load Balancing Algorithm in Cloud Computing”, International Conference on Advanced Computing Technologies and Applications (ICACTA), pp. 832-841, 2015.
- [10] M.A. Vasile, F. Pop, V. Cristea, J. Kołodziej, “Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing”, Future Generation Computer systems , vol. 51, pp61-71, 2015.