Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018)
IEEE Xplore Compliant - Part Number:CFP18J06-ART, ISBN:978-1-5386-0807-4; DVD Part Number:CFP18J06DVD, ISBN:978-1-5386-0806-7

OPTIMIZED TASK CLUSTERING FOR MOBILE CLOUD COMPUTING USING WORKFLOWSIM

V.Meena 1, Arvind.V 2 , Vijayalakshmi.P 3, Kalpana.V 4, J.SenthilKumar 5
1,2,3,4,5 School Of Computing, SASTRA Deemed to be University,Thanjavur, India
meena@cse.sastra.edu

***ABSTRACT*---Nowadays, mobile devices are used for various applications. Although mobile can perform many things, it is a resource restriction device (i.e.) facilities and processing speed are limited. While executing high computational tasks, processing time is high in mobiles due to its sequential task processing algorithm. Computational time of an application depends upon the number of machines (Virtual) involved. Virtualization in Cloud computing is an emerging concept (field) which provides solutions for many complex problems. By combining the merits of virtualization in mobile computing, overall efficiency can be increased. Computationally intensive heavy tasks can be offloaded to rich resourceful server which meets its requirements for time efficient processing. Offloaded tasks are time efficiently processed using virtualization in cloud computing where many VMs (Virtual Machines) are used to dynamically complete the given application using suitable algorithm.The main objective of this paper is to find optimized mapping between number of tasks offloaded, and virtualization in cloud resource. The proposed approach is implemented in Workflowsim which is a cloud resource simulator, and the data obtained is statistically analysed for optimized result.**

Keywords-MobileCloud computing; Simulation; Virtualization; Offloading; Task assignment; Mapping

I. INTRODUCTION

Workflow is the series of activities are necessary to complete a heavy computational task.To facilitate evaluation of workflow algorithms, workflow generators were developed.Observations and inferences gathered from actual execution of scientific workflows along with our understanding of the processes are used by these generatorsto generate realistic workflows similar to that of,its real-world applications.

On scientific workflow, the simulation is one of the famous method of evaluation.Theexisting workflow may fail in task clustering and frame work which should be considered. So, the workflowsim is used in this paper to simulate these workflows.

The workflow can be applied in various departments like astronomy, bioinformatics, and physics due to the availability of scientific application among various programming paradigms. Thousands of tasks can be performed using single workflow. Using this we can stimulate thousands of tasks. Among the thousands of tasks one of the task to be elaborated is mobile cloud computing. Where the mobile with lower configuration are assigned to a lower task and other task are performed on the cloud. In this cloud assignment, the user can acquire virtual machine which can be considered as additional hardware.

If the workflow is needed to avoid the dead line the scheduler has to be assigned to estimate the run time of application. This may be analyses from the historical data available. But due to the term of execution and data transfer the regular performance is not maintained. If a task is delayed it may affect the succeeding task that leads to missing of dead line. There is a serious constrain if a virtual machine is fail during task execution is delayed. so, the performance failure is concern as serious issue than the resource failure.The workflowsim to avoid inaccuracies in the predicted workflow run time it ignores system overheads and failures in simulating scientific workflows. By this simulation complicated setup and efforts in the workflow execution are saved.

II. RELATED WORKS

Deelmanet al.[1] has given an overview of famous work process frameworks for logical applications. Workflowsim included different layers best of the current work process booking layer of CloudSim, which incorporate the Workflow Mapper, the Workflow motor, the Clustering Engine, the Failure Generator, the Failure screen and so forth.

Similarly, Kanittha Promsakul and Somchai Limsiroratana[2] has used the workflow simulation for automation system. Workflow Simulation can increase the efficiency of workflow management and scalable for solving heavy computational tasks.

In order to simulate an experiment, the various emerging cloud computing infrastructures and its applications, an extensible simulation framework is required.CloudSim is such a tool and a well-known system for demonstrating and re-enacting distributed computing foundations and administrations, and ithas several features.

Task clustering is a method that union's fined-grained undertakings into coarse-grained employments.

By using the workflowsim's scheduling and execution we made series of the test cases and observations on various scheduling present in the workflowsim and we arrived at the best cases of the

978-1-5386-0807-4/18/$31.00 ©2018 IEEE

scheduling. We have only changed the MIPS and number of virtual machines assumed based on workflowsim in MINMIN algorithm.

III. PROPOSED METHODOLOGY

First, we executed the workflow in Workflowsim with configuration that of a mobile device, by applying MINMIN algorithm and recorded its completion time. Montage tasks are used for purpose of simulation. Mobile devices usually have low configuration of RAM, MIPS, and number of processing elements. Second, we offloaded all the tasks in workflow to cloud resources and MINMIN algorithm is applied to schedule the tasks in workflow in different configurations of cloud resource. Then, by analysing the data results from these 2 simulations, we try to find the tasks which needed to be offloaded for MINMIN algorithm to schedule with minimum completion time and with efficient cost (i.e.) data or resources.In this analysis, we changed the VMs (1-20), MIPS. The scheduling algorithm used in this experiment is MINMIN algorithm.

IV. EXPERIMENTAL SETUP

We conducted the experiment ((i.e.) execution of the workflow) in Workflowsim (simulator) with the specifications of a mobile devicewhoseconfiguration are RAM 512 KB, HDD 16 GB, Processor Intel 1.4GHz and 1.6 GHz (i.e.) MIPS 1000 and MIPS 500, VM 1, 512 RAM. We observed that completion time is very large and power consumption is also high.

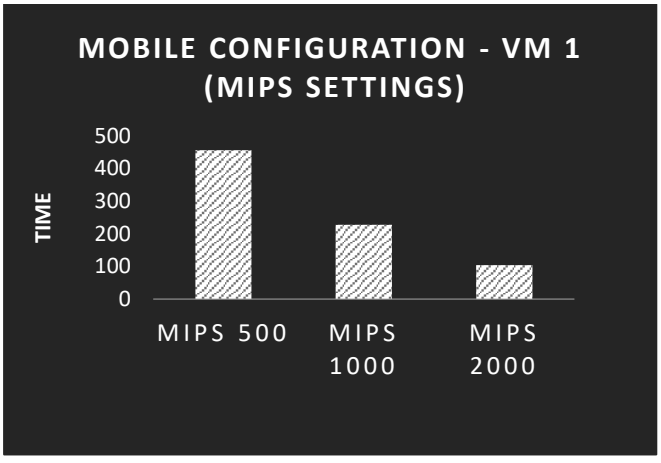

Fig 1: Plot between MIPS values and execution time

We found that, executing all the tasks in cloud doesn't provide an optimized solution. We have decided to offload specific tasks to cloud resources in order to reduce the total time of processing. Now our problem is converted to task assignment problem. Basically, task assignment problem is NP hard problem. For that we used cloudsim simulator to produce output. In that, we analyzed efficient mapping forall the tasks by varying both the number of virtual machines and the MIPS settings for MINMIN algorithm. The main objective of this paper is optimization by offloading certain tasks to cloud resources (i.e.)we determined the tasks which when offloaded to cloud resources reduce the total execution time. First, we conducted experiment on virtual machine configuration with a fixed number of VMs(only 1 VM)and thenwith maximum number of VMs for MINMIN algorithm with a MIPS value of 1000. Second, we conducted the same experiment for same MINMIN algorithm with a different fixed MIPS value of 500.

A. Fixed number of VMs (Only 1 VM) - MIPS 1000 (MINMIN)

On the normal configuration of MIPS of 1000 and adequate RAM, the completion time for the 25 tasks comprising workflow for MINMIN algorithm is recorded as 227.98 with the use of 1 VM. Upon further analysis of time taken by each task, we found that all the tasks are run one after the other in a specific order.

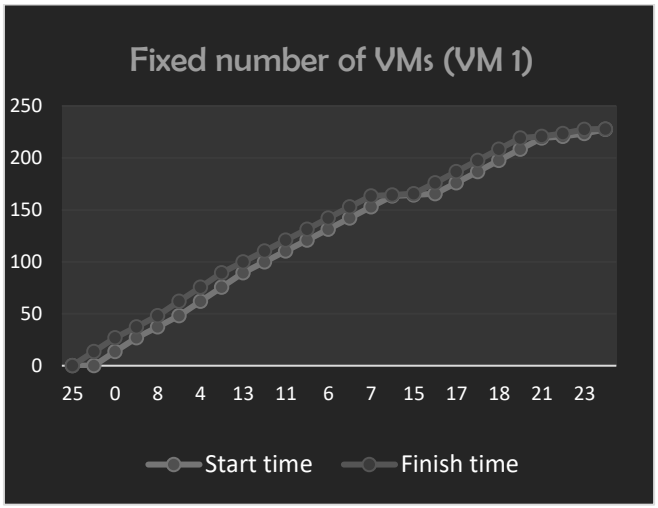

Fig 2: Plot between Time taken and tasknumber

Fig 2, shows the start time and end time of each and every task executed it in its execution order. From the graph it is found that the tasks are executed serially where the next task gets executed only after the completion of its previous task (i.e.) Task no 2 gets finished at 13.57 and the next task (Task no. 0) starts at 13.57.

B. Maximum number of VMs – MIPS 1000 (MINMIN)

In this setup, the complete execution time of the same 25 tasks comprising workflow for MINMIN algorithm is recorded as 46.77 with the use of maximum number of VMs (In this case 9).

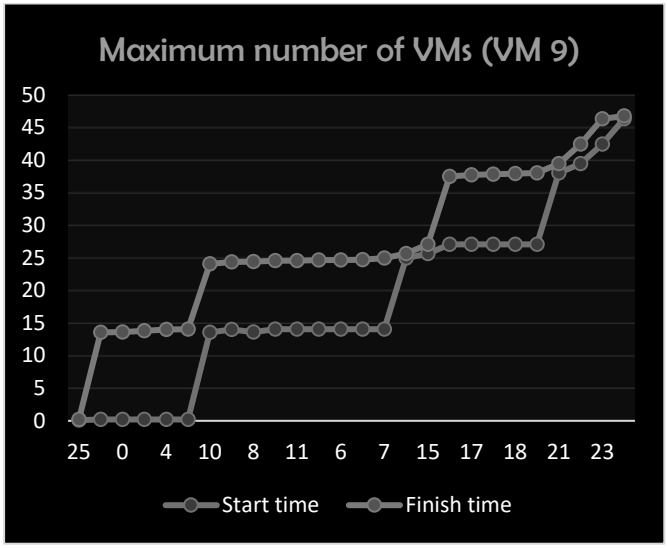

Fig 3: Plot between time and task number for max VMs (In this scenario 9)

Fig 3 shows the starting time, finishing time and the time duration of all the 25 tasks. In this scenario, all tasks are not executed one after the other. Certain tasks run parallelly with other tasks in different VMs. For example, task number 16, 17, 20, 18, 19 are executed at the same time (27.08). Tasks which doesn't require the output of the previous task run parallelly in next VM. And hence the execution time of tasks with more than 1 VM is less compared to that of 1 VM. As the number of VMs used increases, there is a decrease in the total execution time. However, no reduction in total time is observed when VM is increased beyond 9. This indicates that, only a maximum of 9 tasks can be run simultaneously in this case.

Table 1: Analyzation of Fig 3 – Correlation between tasks and their starting time

Start Time	0.21	13.6	14	27.08
Number of tasks running Parallelly	5	2	7	5

Table 1 shows the number of tasks running parallelly at a given specified time. The number of VMs used need not always be the number of tasks running parallelly at the start time. For example, at 13.6, 2 tasks (task no 8, 10) run parallelly. Similarly, at 14, 7 tasks (task no 13, 9, 11, 5, 6, 12, 7) run parallelly, but instead of 7, 9 VMs are used at that time and this is because the individual execution time of task no 8 and 10 is around 10.7. Which means, by the time the 7 tasks are running parallelly, those 2 tasks are also running at VM id 0 and 2.The contents from Table 1 are plotted, in order to get a better understanding in Fig 4.

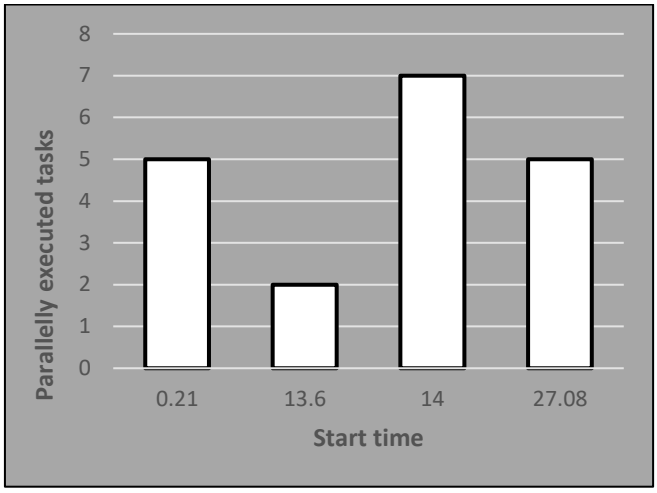

Fig 4: Correlation between the parallel execution of tasks and their respective start time

C. *Findings based on the variation of number of VMs – MIPS 1000 (MINMIN)*

Usually in a mobile with an average specs, the number of VMs it can allow is 1. But from comparing the total time taken for the execution of tasks with only 1 VM and maximum number of VMs (1 VM – 227.98, 9 VM – 46.77) in MIPS settings of 1000, it is evident that the process is more efficient and optimized when it is run with maximum number of VMs. However, From Fig 3, it is also evident that not all tasks are run on different VMs. So, instead of running the entire process in cloud, which isn't cost efficient and impractical, we could offload only those tasks which can run parallelly and thereby optimizing the execution or processing time and, this process is cost efficient and practical when compared to that of previous one.

Tasks which runs parallelly are the tasks which need to be offloaded and run in cloud. However, not all these tasks need to be run in cloud for optimization. Only tasks with very high individual execution time which when offloaded and run could drastically reduce the total time, those tasks should only be run in cloud. From Fig 3and Table 1, we could infer that tasks with task number 2, 0, 3, 4, 1, 13, 9, 11, 5, 6, 12, 7, 8, 10, 16, 17, 20, 18, 19 (19 tasks) when offloaded and run in cloud reduces the total time taken (227.28-(((13.57-0.2)*4)+((24.5-14.04)*8)+((37.84-27.08)*4))) = 47.16)which is very less and also practically feasible. Instead of uploading all 25 tasks (Total time taken is 46.77), we could upload only those specific 19tasks (Total time taken is 47.16) thereby reducing total time which also proves efficiency.

From complete analysation of Fig 5, it is clear that the time taken for running all the tasks in cloud and specific (19 in this case) tasks in cloud is more or less the same. Therefore, the efficient and optimized way is to offload only specific tasks (as the number of tasks needed to be offloaded is less) and run them in cloud.

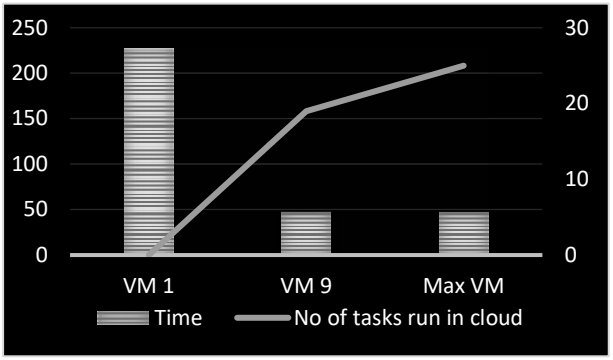

Fig 5: Analysis between the time taken andnumber of tasksoffloaded for different cases

D. Fixed number of VMs (Only 1 VM) - MIPS 500 (MINMIN)

On the normal configuration of MIPS of 500 and adequate RAM, the completion time for the 25 tasks comprising workflow for MINMIN algorithm is recorded as 455.83 with the use of 1 VM. Upon further analysis of time taken by each task, we found that all the tasks are run one after the other in a specific order.

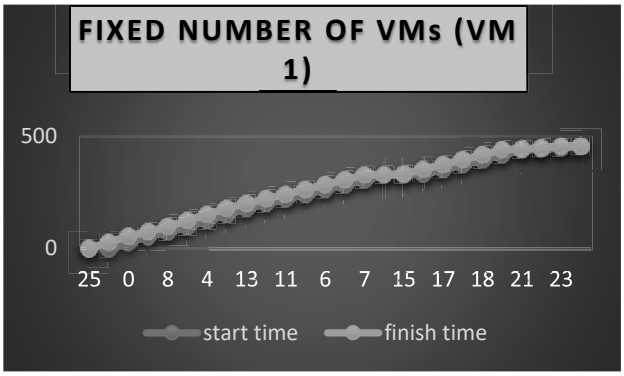

Fig-6 Plot between task and time taken

Fig 6, shows the start time and end time of each and every task executed it in its execution order. From the graph it is found that the tasks are executed serially where the next task gets executed only after the completion of its previous task (i.e.) Task no 2 gets finished at 27.04 and the next task (Task no. 0) starts at 27.04.

E. Maximum number of VMs – MIPS 500 (MINMIN)

In this setup, the complete execution time of the same 25 tasks comprising workflow for MINMIN algorithm is recorded as 93.37 with the use of maximum number of VMs (In this case 9).

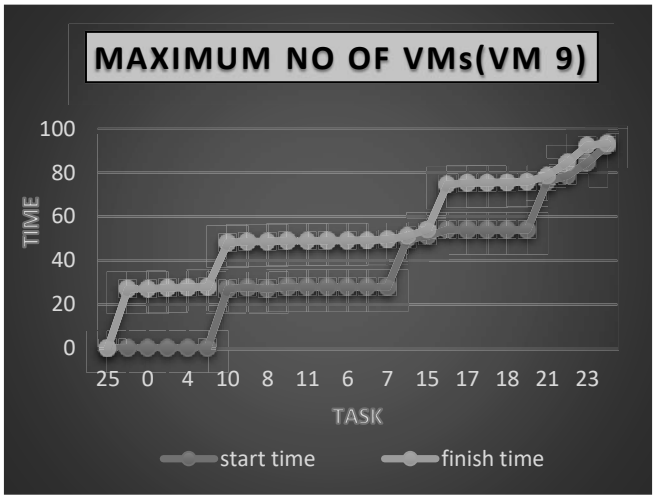

Fig 7: Plot between time and task number for max VMs (In this scenario 9)

Fig 7 shows the starting time, finishing time and the time duration of all the 25 tasks. In this scenario, all tasks are not executed one after the other. Certain tasks run parallelly with other tasks in different VMs. For example, task number 16, 17, 20, 18, 21 are executed at the same time (54.03). Tasks which doesn't require the output of the previous task run parallelly in next VM. And hence the execution time of tasks with more than 1 VM is less compared to that of 1 VM. As the number of VMs used increases, there is a decrease in the total execution time. However, no reduction in total time is observed when VM is increased beyond 9. This indicates that, only a maximum of 9 tasks can be run simultaneously in this case.

Table 2: Analyzation of Fig 3 – Correlation between tasks and their starting time

Start Time	0.32	27.04	27.98	54.03
Number of tasks running Parallelly	5	2	7	5

Table 2 shows the number of tasks running parallelly at a given specified time. The number of VMs used need not always be the number of tasks running parallelly at the start time. For example, at 27.04, 2 tasks (task no 8, 10) run parallelly. Similarly, at 27.98, 7 tasks (task no 13, 9, 11, 5, 6, 12, 7) run parallelly, but instead of 7, 9 VMs are used at that time and this is because the individual execution time of task no 8 and 10 is around21. Which means, by the time the 7 tasks are running parallelly, those 2 tasks are also running at VM id 0 and 2. The contents from Table 1 are plotted, in order to get a better understanding in Fig 8.

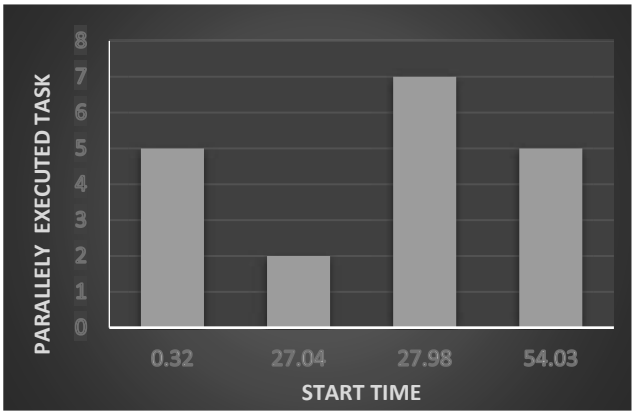

Fig 8: Correlation between the parallel execution of tasks and their respective start time

F. *Findings based on the variation of number of VMs – MIPS 500 (MINMIN)*

Usually in a mobile with an average specs, the number of VMs it can allow is 1. But from comparing the total time taken for the execution of tasks with only 1 VM and maximum number of VMs (1 VM – 455.83, 9 VM – 93.37) in MIPS settings of 500, it is evident that the process is more efficient and optimized when it is run with maximum number of VMs. However, From Fig 3, it is also evident that not all tasks are run on different VMs. So, instead of running the entire process in cloud, which isn't cost efficient and impractical, we could offload only those tasks which can run parallelly and thereby optimizing the execution or processing time and, this process is cost efficient and practical when compared to that of previous one.

Tasks which runs parallelly are the tasks which need to be offloaded and run in cloud. However, not all these tasks need to be run in cloud for optimization. Only tasks with very high individual execution time which when offloaded and run could drastically reduce the total time, those tasks should only be run in cloud. From Fig 3 and Table 1, we could infer that tasks with task number 2, 0, 3, 4, 1, 13, 9, 11, 5, 6, 12, 7, 8, 10, 16, 17, 20, 18, 19 (19 tasks) when offloaded and run in cloud reduces the total time taken (455.83 – (((27.04-0.32) *4) +((49.01-27.98) *8) +((74.81-54.03) *4))) = 97.59) which is very less and also practically feasible. Instead of uploading all 25 tasks (Total time taken is 93.37), we could upload only those specific 19 tasks (Total time taken is 97.59) thereby reducing total time which also proves efficiency.

From complete analysation of Fig 9, it is clear that the time taken for running all the tasks in cloud and specific (19 in this case) tasks in cloud is more or less the same. Therefore, the efficient and optimized way is to offload only specific tasks (as the number of tasks needed to be offloaded is less) and run them in cloud.

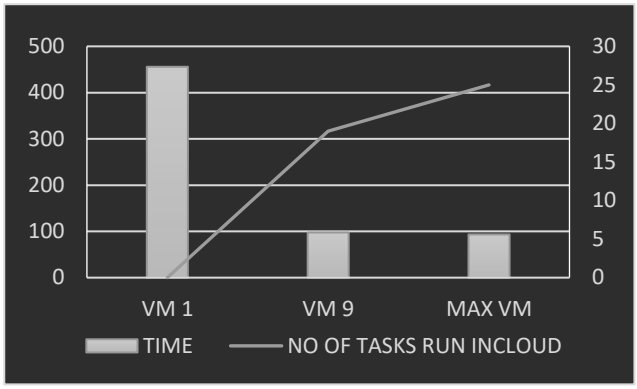

Fig 9: Analysis between the time taken and number of tasks offloaded for different cases

V. INFERENCE

Based on Fig 5, it is clear that executing 19 tasks (task number 2, 0, 3, 4, 1, 13, 9, 11, 5, 6, 12, 7, 8, 10, 16, 17, 20, 18, 19) in cloud and the remaining in mobile is the efficient way for a MIPS value of 1000. Based on Fig 9, it is clear that executing 19 tasks task number 2, 0, 3, 4, 1, 13, 9, 11, 5, 6, 12, 7, 8, 10, 16, 17, 20, 18, 19) in cloud and the remaining in mobile is the efficient way for a MIPS value of 500. Similarly, it is observed that irrespective of the MIPS setting value, those particular 19 tasks when run in cloud provides optimized output. Analysing all these results, we could draw to a conclusion and summarise it in Fig 10.

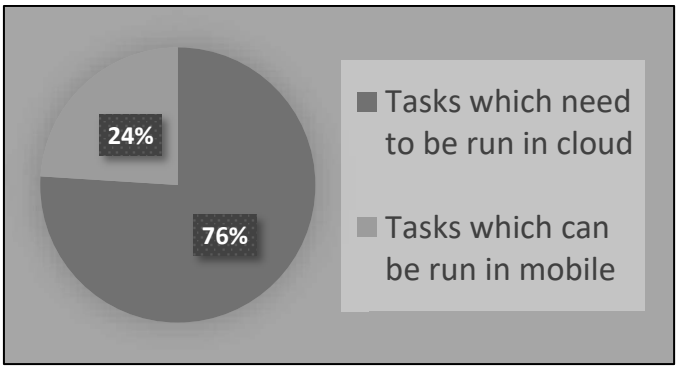

Fig 10: Optimized and efficient task configuration (irrespective of MIPS value)for mobile computing

VI. CONCLUSION

We conducted the experiment to find the efficient and optimized configuration of offloading particular tasks in cloud resource. We have done a experiment by analysing the efficient mapping for all the tasks by varying, both the number of virtual machines and the MIPS settings for a particular (MINMIN) algorithm. On complete analysation Fig 10 shows the best configuration for the montage tasks based on the completion time and number of VMs used. This experiment and its analysis has thus proved that whatever may be the value of MIPS settings, the efficient way to execute a process is to run

particular tasks which can run parallelly and takes a considerable time in cloud (19 tasks with task number 2, 0, 3, 4, 1, 13, 9, 11, 5, 6, 12, 7, 8, 10, 16, 17, 20, 18, 19) and run the remaining tasks in mobile specifications (6 tasks with task number 14, 15, 21, 22, 23, 24). This algorithm is entirely based on MIPS settings and number of VMs.

REFERENCES

[1]. Deelman, E., et al.,Workflows and e-Science: "An overview of workflow system features and capabilities," Future Generation Computer Systems, May 2009.

[2]. Limsiroratana ,Somchai and Promsakul , Kanittha. "Workflow Simulation Based On Cloud Platform For Office Automation System", Computer Science and Software Engineering(JCSSE), July 2010.

[3]. B.Schroeder, et al., "A large-scale of failures in high performance computing systems," Philadelphia, USA, Jun 2006.

[4]. G. Singh, et al., "Workflow Task Clustering for Best Effort Systems with Pegasus," Mardi Gras Conference, Baton Rouge, Jan 2008.

[5] D. Comer, "Internetworking with TCP/IP," Volume I, Principles,Protocols, and Architecture. Prentice hall Englewood Cliffs, NJ,1995, vol. 3.

[6] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu,R. Chandra, and P. Bahl, "Maui: making smartphones last longer withcode offload," in Proceedings of the 8th international conference onMobile systems, applications, and services. ACM, 2010, pp. 49–62.

[7] H. Flores and S. N. Srirama, "Mobile cloud middleware," JournalofSystemsandSoftware, http://dx.doi.org/10.1016/j.jss.2013.09.012.

[8] H. Flores and S. N. Srirama, "Adaptive code offloading for mobilecloud applications: Exploiting fuzzy sets and evidence-based learning,"in Proceedings of the Fourth ACM Workshop on Mobile CloudComputing and Services (MCS 2013), 2013.

[9]K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Canoffloading computation save energy?" Computer, vol. 43, no. 4, pp.51–56, 2010.

[10] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloadinghow much can wifi deliver?" in Proceedings of the 6th InternationalCOnference. ACM, 2010, p. 26.

[11] R.K. Sahoo, et al., Failure Data Analysis of a Large Scale Heterogeneous Server Environment, DSN 2004, Florence, Italy, Jul 2004.