International Journal of Engineering & Technology, 7 (4.39) (2018) 535-538

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Various Task Allocation Simulator for Resource High-Performance based on Mobile Cloud Infrastructure

Hyun-Woo Kim[1], Eun-Ha Song[2]*

[1]Dept. of Multimedia Engineering, Dongguk University, Seoul, Republic of Korea
[2]Collage of Convergence & Liberal Arts, Wonkwang University, Iksan, Jeonbuk, Republic of Korea
**Corresponding author Email: ehsong@wku.ac.kr*

Abstract

Background/Objectives: We describe VTAS for efficiency of resource management, high availability in mobile cloud environment based on collaborative architecture.
Methods/Statistical analysis: VTAS sets performance normalization for task allocation according to CPU, memory, and remaining battery power of mobile resources for various task processing. Visualize task allocation information and processing time when requesting work according to performance normalization.
Findings: Research has been carried out in consideration of resource configuration, network connection status, number of tasks, and simulations for cloud resource management. There are insufficient studies to allocate various work processes based on mobile cloud. By assigning user task allocation criteria to simulation, task allocation for various task processing is possible.
Improvements/Applications: In this paper, it is possible to apply user arbitrary task allocation criterion of limited integrated resources in mobile cloud, and more work can be processed by task allocation considering static and dynamic performance and remaining battery capacity.

Keywords: *Resource High-Performance, Various Task Allocation, Mobile Cloud Computing, Task Allocation Simulation*

1. Introduction

With the advancement of ICT technology, mobile device performance is dealing with desktop level work. These mobile devices are continuously increasing their use of leisure time because of their work efficiency, convenience, and mobility due to their ability to handle the business if the Internet is connected regardless of the location. Mobility, an advantage of mobile devices, has been studied in mobile cloud infrastructures to overcome performance limitations due to limited battery power [1-2]. By integrating the storage and computing resources of nearby multi-platform mobile devices. It is the mobile cloud infrastructure allocates resources to the mobile devices that need it, making it possible for a single mobile device to perform tasks that are difficult to handle alone. The mobile cloud infrastructure is very important because its performance is determined according to the task allocation method according to the user's request [3-6]. In the past, only static and dynamic performances are considered, so there is a delay in the task allocation due to insufficient memory or CPU operation. In this paper, we propose VTAS (Various Task Allocation Simulator) to minimize work delay according to various kinds of work. The VTAS receives mobile information and job information from the user, creates a mobile node, and visualizes the process by allocating the job. This allows users to set up a baseline task for resource high availability within the mobile cloud infrastructure.

In [4], tasks are allocated in consideration of CPU, memory, and storage in a mobile cloud infrastructure consisting only of mobile devices. However, since the amount of memory required for the actual operation is not taken into consideration, a problem that the job is not allocated occurs. In [5], the mobile cloud periodically measures CPU usage and migrates work to other mobile devices when it exceeds the threshold. This allows uniform processing of tasks according to CPU performance, but it is difficult to apply them because memory is not considered.

In [6], we proposed GridSim, which is simulator using Java and can simulate of scheduling algorithm. The resource entity instance of the GridSim can represent the processor counts, processing cost, processing speed, and process policy of internal scheduling in each resource that are multi-processor. It is easy to simulate a virtual environment through GridSim. However, users have to write program source code in order to manipulate various variables needed for resource analysis, and there is a need to change the result to a chart or the like. Therefore, this paper provides GUI-based resource setting, operation process and visualization of results.

In [7] proposed ClusterSim, a Java-based parallel discrete-event simulation tool. ClusterSim supports workload and visual modeling. In addition, the simulator allows the user to select nodes for workload processing and simulate the message passing interface (MPI) and parallel job scheduling algorithm among clusters. However, it does not consider the mobile resource environment composed of heterogeneous or homogeneous, and there is a problem that the user has to manually input resources.

In [8], CloudSimis proposed to support behavior modeling such as virtual machine (VM), data center, and resource service policy in a cloud infrastructure configuration. It is also possible to allocate VMs between networks via a custom interface. However, since the result is derived from a text base, it is difficult to understand the cause and the process of the problem. Therefore, in this paper, we provide performance normalization setting and visual operation status to users in order to operate mobile resource effectively.

2. VTAS Scheme

VTAS is implemented in a mobile cloud infrastructure that operates as shown in Figure 1. In Figure 1, ①is the network node connection, ②broadcasts its own performance information in the connected network, ③broadcasts the master candidate node list to each node, ④selects the master device based on the master candidate list based on the collected master candidate lists, the selected master device is selected by the server activation and the client connection standby. After that, ⑥the user's computing work request in Figure. 1, ⑦ job distribution and allocation to the client, and ⑧the completed job is transmitted by sending the job distribution client to the user.

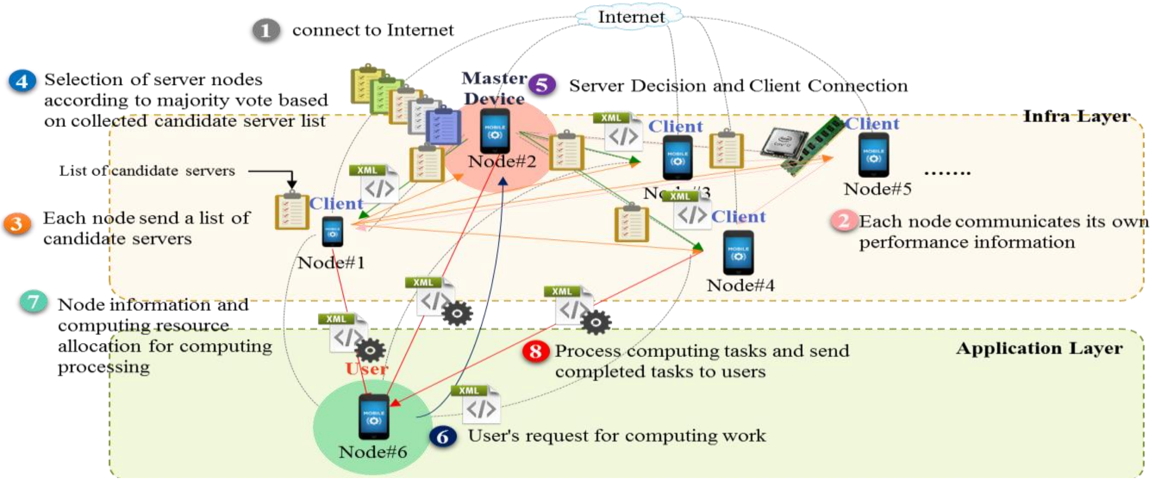

Figure 1: Task Operation and Master Device Selection in Mobile Cloud Infrastructure

In order to minimize the work delay in the mobile cloud infrastructure, VTAS requires the user to assign rules for job allocation and performance normalization of mobile resources. Performance normalization takes into consideration mobile device information such as unique ID, memory, CPU, storage, and remaining battery capacity. In addition, priority is set for job assignment. The priority of the task assignment is input by the user to the simulator. After this, the mobile device is placed. Master device select to integrate the neighboring mobile devices. Connect with the mobile device except for the master device, and add work according to the user definition. The job information includes the memory size for job allocation, and the processing time of the job. The mobile device compares the amount of memory required for the task to determine whether the task is assignable. When a job is assigned, it performs during the job processing time of the job information, and visualizes the result of the performance normalization and the job allocation method set by the user.

3. VTAS Design

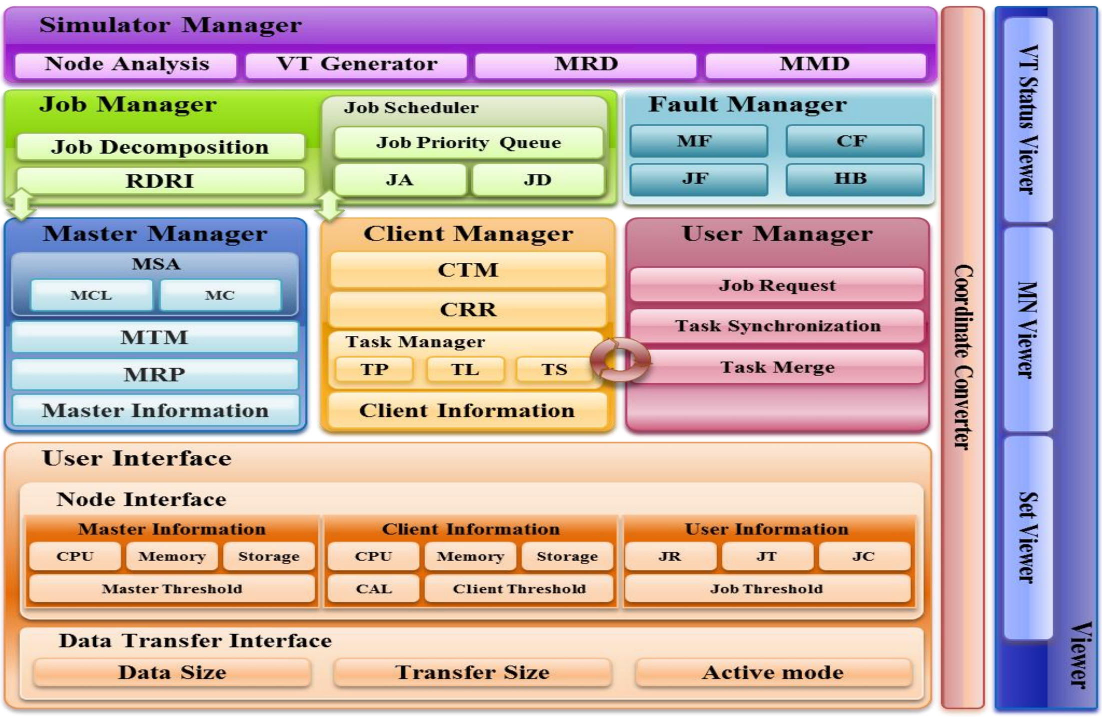

Figure 2: VTAS Architecture

The User Interface is divided into Node Interface and Data Transfer Interface in detail. The Node Interface is composed of Master Information for CPU, Memory, Storage and Master Threshold setting of master node, CPU, Memory, Storage, Client Allocation List (CAL) of client node, Client Information for setting work acceptance threshold, Request), JT (Job Table), and JC (Job Command). Data Transfer Interface consists of data size, size of data transfer between tasks, and whether or not it is active.
Master Manager consists of Master Selection List (MCL), Master Connection (MC), Master Threshold Measurement (MTM), and Mobile Resource Pooling (MRP) and Master Information in the Master Selection Algorithm (MSA) for master selection.
Client Manager is composed of Task Processing (TP), Task List (TL), Task Submission (TS) and Client Information. It is a task manager for CTM (Client Threshold Measurement) and CRR (Client Resource Release)
User Manager is composed of Job Request for job request, Task Synchronization for job division and management, and Task Merge for merging divided task results into VTAS.
Job Manager is composed of Job Decomposition for job redistribution in case of failure, Requesting Dynamic Resource Information (RDRI) for assigning job using current mobile performance information, Job Priority Queue, Job Add (JA) in Job Scheduler for job- , And JD (Job Delete).
Fault Manager manages the master fault for master node failure, CF (client fault) for fault response of client node, JF (Job Fault) for unacceptable processing, and HB Heartbeat). When a master node failure occurs, the master selection algorithm operates. When a client node fails, the client analyzes the job information that has been allocated, and transmits job reallocation and client node release notification to all the connected mobile nodes.

Simulator Manager is composed of Node Analysis, which analyzes the generated mobile node, Various Task Generator (VT), which generates various divided tasks, task, Mobile Resource definition, which defines mobile resources, and Mobile Mapping Definition, which defines rules about mobile mapping.
The Coordinate Converter plays the role of processing the data by the Viewer to visually show the process to be performed by receiving the simulation element from the user.
The Viewer consists of VT Status Viewer for visualizing the work processing status, MN Viewer for visualizing the connection status and list of the mobile node, and Set Viewer for receiving the simulation setting value from the user.

4. VTAS Implementation and Performance Evaluation

Figure 3 shows the VTAS operation screen. Figure 3 shows the process of simulating 10 jobs in a mobile cloud environment with 7 mobile nodes connected. ①describe configure the number of mobile users, mobile performance information such as Memory, CPU, battery consumption rate. ② shows the mobiles created with the information set in ①, ③ can configure the number of jobs, CPU and memory required for work in the configured mobile environment. ④ in Figure 3 visually show the progress of work performed through Run. ⑤ in Figure 3 show the configuration in which the mobile is connected. Through this, it is possible to actively cope with the actual work by visualizing the operation process according to the work set by the user.

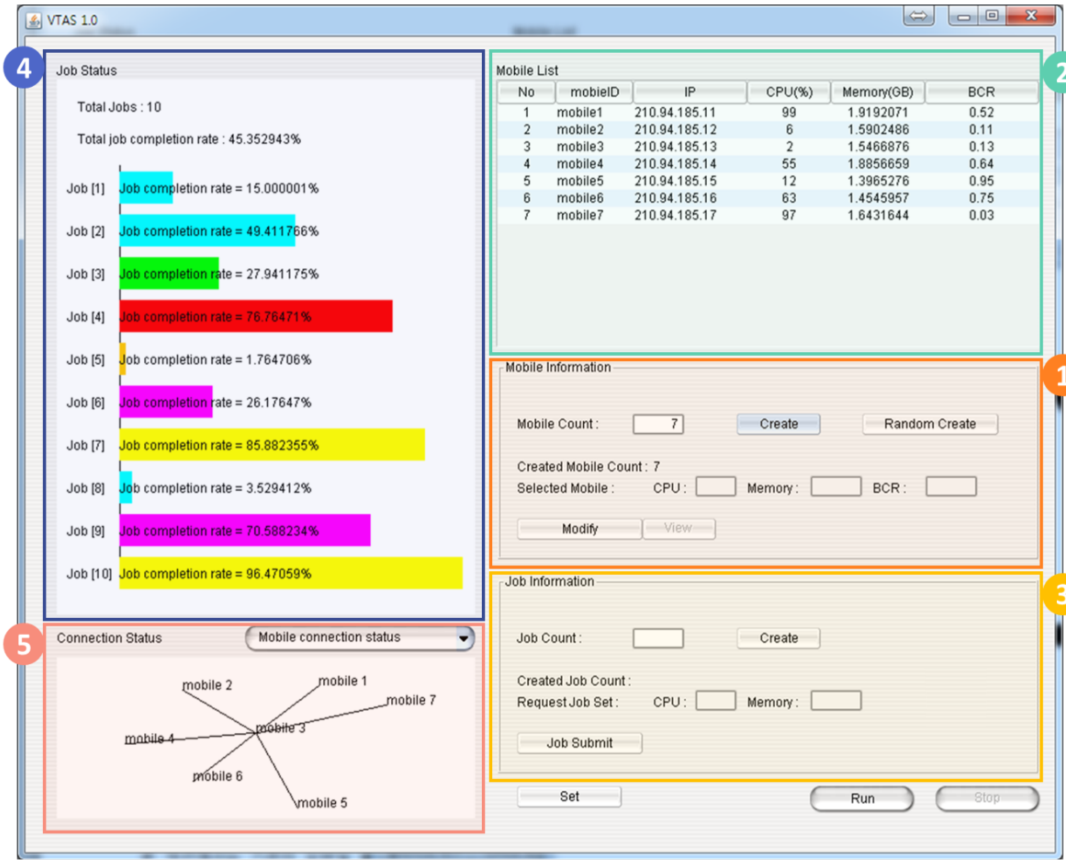

Figure 3: VTAS Operation View

In this paper, VTAS simulation and real mobile device work are compared according to the increase number of mobile devices shown in Figure 4.
The mobile increases number, 500 x 500 matrix multiplication value is an average of 50 operations shown in Figure 4.

In the case of one mobile device, the VTAS simulation and the real operation are greatly different, but the difference is slight to 0.6 seconds. 0.04 seconds for five mobile devices, 0.09 seconds for six mobile devices, and 0.03 seconds for seven mobile devices. VTAS was measured 94.31% accurately with real mobile devices.

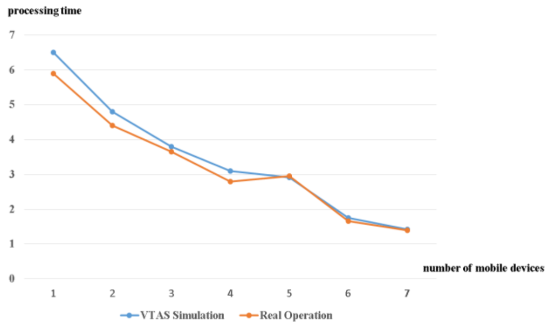

Figure 4: Comparison of VTAS simulation and execution time of real operation

5. Conclusion

In this paper, we propose VTAS which can select or set work allocation method according to the work required in mobile cloud infrastructure environment. VTAS is able to check the status of task processing by simulating the performance weights required for computing tasks in various mobile cloud infrastructures. In addition, visualization of mobile connectivity and work status allows the user to set performance criteria for various tasks. This enables high-availability of integrated mobile resources. In addition, we have confirmed that the simulation results show high accuracy over a short period of time by comparing VTAS simulations with real operation results in mobile devices.

In the future, we want to configure the system architecture so that the CPU usage, memory usage, the task allocation method are input from the user in the VTAS. In addition, we will compare and analyze the actual mobile cloud infrastructure by applying simulation results.

Acknowledgment

This paper was supported by Wonkwang University in 2018.

References

[1] Rubens, M., Jean, A., Danilo, O., Paulo, M., Kishor, T.(2015). Sensitivity analysis of a hierarchical model for mobile cloud computing. Simulation Modeling Practice and Theory, 50, 151-164.

[2] Abolfazli, S., Sanaei, Z., Sanaei, M.H., Shojafar, M., Gani, A. (2015). Mobile cloud computing: The-state-of-the-art, challenges, and future research. In Encyclopedia of Cloud Computing, Willeys and Sons: Hoboken.

[3] Lo'ai, T., Waseem, B., Houbing, S. (2016). A Mobile Cloud Computing Model Using the Cloudlet Scheme for Big Data Applications.In Proceedings of the Conference on Connected Health: Applications, Systems and Engineering Technologies, 2016 IEEE First International Conference on, IEEE, 73-77.

[4] Kim, B., Byun, H., Heo, Y.-A., Jeong, Y.-S.(2017). Adaptive Job Load Balancing Scheme on Mobile Cloud Computing with Collaborative Architecture, Symmetry, 9(5), 1-14.

[5] Yi, G., Heo, Y.-A., Byun, H., Jeong, Y.-S. (2017). MRM: mobile resource management scheme on mobile cloud computing, Journal of Ambient Intelligence and Humanized Computing, 1-13.

[6] Buyya, R., Murshed, M. (2002).GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing,Concurrency and Computation: Practice and Experience, 14(13-15), 1175-1220.

[7] Góes, L. F. W., Ramos,L. E. S., Martins, C. A. P. S. (2004). ClusterSim: A Java-Based Parallel Discrete-Event Simulation Tool for Cluster Computing. In Proceedings of the 2004 IEEE International Conference on Cluster Computing, 401-410.

[8] Calheiros,R. N., Ranjan,R., Beloglazov,A., Rose, C. A. F. D., Buyya, R. (2011).CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation or resource provisioning algorithms,Software: Practice and Experience, 41(1), 23-50.