Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
ISSN: 2788-7421

Using Ideal Time Horizon for Energy Cost Determination

Raed Abdulkareem Hasan[1,2]*, Hadeel W. Abdulwahid[3], Arwa Sahib Abdalzahra[4]

[1]Electronics Department - Alhawija Technical Institute, Noerthern Technical university, IRAQ

[2]Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, MALYSIA

[3]Department of Computer Science / Islamic Azad University-South Tehran Branch, IRAN

[4]Department of Computer Technical Engineering / Al-Salam University College, IRAQ

*Corresponding Author: Raed Abdulkareem Hasan

DOI: https://doi.org/10.52866/ijcsm.2021.02.01.002
Received September 2020; Accepted November 2020; Available online January 2021

ABSTRACT: In most optimal VM placement algorithms, the first step to determine the proper time horizon, T for the prediction of the expected maximum future load, L. However, T is dependent on the proper knowledge of the required time for servers to switch from their initial SLEEP/ACTIVE state to the next desired state. The activities implemented by this policy are (a) to relocate the VM from an encumbered server, a server that operates in an undesirably high regime with applications forecasted to rise their burdens to compute in the subsequent reallocation cycles; (b) to conduct VM migration from servers that operate within the undesirable regime to shift the server to a SLEEP mode; (c) putting an idle server to SLEEP mode and rebooting the servers from the SLEEP mode at high cluster loads. A novel mechanism for forwarding arriving client desires to the utmost suitable server is implemented; thus, in the complete system, balancing the requested load is possible.

Keywords: Energy Saving; Cloud Computing; Cost Determination.

1. INTRODECTION

The foremost purpose of this algorithm is to expand the process of achieving the principal active servers that operate within the restraints of their optimal OS. Their mechanism influences local balancing to attain global balancing. This was achieved via a periodic interface between the nodes in the system. The existing request routing (RR) techniques may be categorized into Cloud RR, train transport-layer RR, & application-layer RR. The cloud is a dynamic environment, and this allows for differences across servers & time[1]. The evolving need for the cloud to handle the workloads demands for changes in the PM-hosted services; this is driven by the need to either host new VMs or create VMs on a new PM. Empirical experiments were employed in this study to determine the stochastic model of state-change latency. This was achieved by determining the required times to switch servers from ACTIVE to SLEEP modes and vice versa via observations, this paper present Ideal Time Horizon for Energy Cost Determination while the organizing of paper is introduction as section 1, section 2 Energy – Aware Cloud Architecture, section 3 Energy Consumption in Cloud , section 4 Dynamic Energy-Aware Cloudlet-Based Model, section 5, Determining the Ideal Time Horizon and section 6 Experimental Determination of State Change Statistics

*Corresponding author: raed.isc.sa@gmail.com

2019 ©AL- Iraqia University, College of Education Publishers Office All rights reserved.

2. ENERGY-AWARE CLOUD ARCHITECTURE

The physical servers are instrumented with multi-ore Central Processing Units (CPUs). A multi-core CPU with n cores has m MIPS, whereas m MIPS is exhibited as a single-core CPU with the overall size of nm MIPS. The above is acceptable because VMs and applications are not joined with the processing cores and are implemented on an arbitrary core using a time-sharing scheduling framework. The only limitation is that the capacity of every single virtual CPU core allotted to a VM must be less or equal to the specific physical CPU core capacity because if a greater CPU size than the distinct physical core capacity is required for a virtual CPU core, a VM must be executed in parallel on the extra physical core. However, parallelization of instinctive VMs with a single virtual CPU may not be possible.

The aforementioned system has a single-tiered software layer that covers the global and local managers (Figure 1). The presence of the local managers is felt as a component of the VMM on all nodes. The duty of the local managers include continuous monitoring of the extent of CPU utilization by each node, resizing the VMs based on their resource requirements, and the determination of when and which VMs are to be migrated from the node. For the global manager, they are felt on the master node where their job is gathering information from the local managers to ensure accurate knowledge of the usage of resources. The global manager provides direction for optimal VM utilization. VMMs perform definite migration, resizing, and deviations in the node's power ways[2].

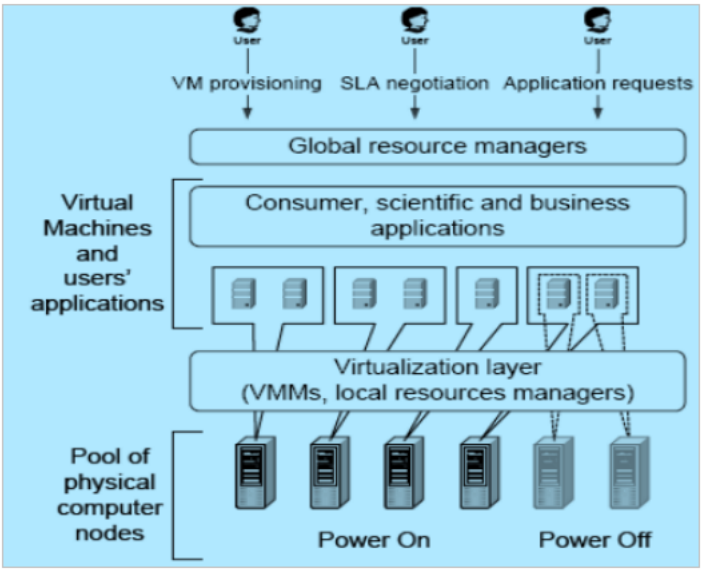


FIGURE 1. Energy-aware system architecture (Madić et al. 2013).

3. ENERGY CONSUMPTION IN CLOUD

The majority of the physical servers in the cloud utilize virtualization technology. In light of the SLA with Cloud suppliers, a set of VMs is ordered by the tenants to be set in various hosts for steady communication among them. The resource demands of each VM for sustained performance and security differs in terms of CPU, storage, and so on. Various virtual servers are run by virtualization technology on the equivalent physical machine (PM), which is necessary for better resource utilization and reduced energy consumption. Hence, cloud managers can also rely on virtualization to achieve on-demand and orderly resource deployment to ensure efficient management of resources and low-energy usage.

Two elements must be deliberated simultaneously with regard to VM position: the distribution of physical server resources such as CPU, memory, stockpiling, and so on, as well as the optimization of network resources. Consequently, the proposed scheme for VM placement can address multiple resource problems. When addressing the problems of PM resources, along with network link dimensions, the resource utilization of PMs and the network components can be augmented by the traverse optimizing VMs set on PMs and improve the device throughput by switching the idle physical device to SLEEP mode and reducing the number of network elements and active physical servers, thereby minimizing energy usage in the cloud.

Physical server optimization via VM placement is considered a bin packing problem (BPP), while network resource optimization using communication traffic and network topology is considered a quadratic assignment problem (QAP).

Because both BPP and QAP are NP-hard problems [3], [4], a need to limit network communication traffic, reduce network traffic and the number of active components in the network exists. For a typical multiple objective optimization problem, the number of PMs and the number of active network components must be reduced to minimize energy depletion in the cloud.

4. DYNAMIC ENERGY-AWARE CLOUDLET-BASED MODEL

Adopting DECM development is a motivational example of methods executed in real life [5]. The DECM model comprises three levels; Mobile Device, Cloudlet, and Cloud Computing, as elucidated in the figure below [4]. As a comparison, non-cloud services are now more efficient that customers can benefit from the service anytime, anywhere, and from any device. This model optimizes the cloud servers' usage in MCC. Figures 2 and 3 show the principle of DECM.

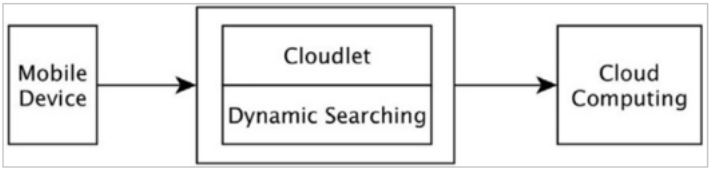


FIGURE 2. **DECM's level for mobile communications (Gai et al. 2016).**

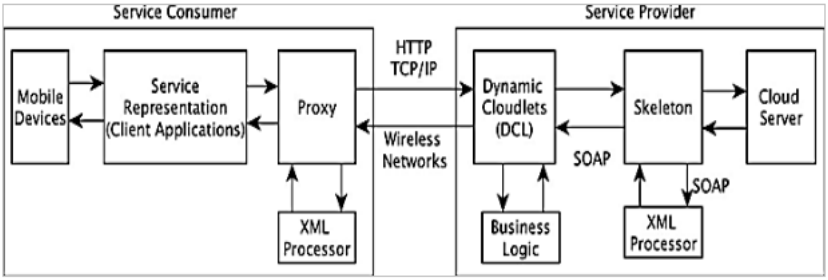


FIGURE 3. **Essential idea for DECM (Gai et al. 2016).**

The other critical factor is resource limitation in the cloud servers' allocation. An elemental requirement is signal coverage. Infrastructure-dependent mobile networks are exceedingly considered because of their high precision and steady availability where they exist. With the deployment of a range of wireless network access, it has become important to deploy & maintain the infrastructure and these are time and resource intensive tasks. 4G/LTE cellular currently provides internet coverage in urban fields despite the associated degradations in urban areas, but beyond urban areas, such coverage degrades quickly.

5. DETERMINING THE IDEAL TIME HORIZON

The first step of most optimal VM placement algorithms is to determine the proper time horizon, which is T for the prediction of the expected maximum future load, L. However, T is mainly dependent on the proper knowledge of the required time for servers to switch from their initial SLEEP/ACTIVE state to the next desired state. If a selected T is below the required time for servers to switch state, the expected number of online servers may not be achieved, and the request load may not be processed. However, if T is excessively large, there is a risk of system over-allocation because the time-varying nature of the request load is not followed during decisions on system allocation. The relationship between T. Evidently, the future allocated capacity in each T interval is determined by the maximum predicted load (the red x). The figure shows that as T moves towards zero, the predicted load also moves towards the actual load; hence, there is a need to frequently reset the allocated capacity as much as the server state-change permits. The times for changes in server state are represented as a random process, X; there is a need to determine the probability density function, fx(x) for X for effective modeling of the state change time; hence, an appropriate new T must be chosen. It is assumed that the server has static configurations, and with a small or no-load request; hence, they are expected to have small differences in the state-change times. The cloud is a dynamic environment, and this allows for differences across servers & time. The evolving need for the cloud to handle the workloads demands for changes in the PM-hosted services; this is driven by the need to either host

new VMs or create VMs on a new PM. Empirical experiments were employed in this study to determine the stochastic model of state-change latency. This was achieved by determining the required times to switch servers from ACTIVE to SLEEP modes and vice versa via observations [6].

6. EXPERIMENTAL DETERMINATION OF STATE CHANGE STATISTICS

Four Dell workstations were used for the experiments in the laboratory; three of the workstations served as the remote clients while the remaining one was the master controller. Each workstation was configured with a unique set of services. The four systems were equipped with the Ubuntu 11.10 Linux distribution system. Each computer was connected using a Bash script[7]; the script also served in issuing SLEEP and WAKE commands in a looping cycle. The recording of the time between each command state was conducted using a local file. The script was allowed to run through 1000 command cycles (Active-Sleep and vice versa) on each machine and from the data, no change was determined in the state-change time for the same system upon no configuration modification[8][9]. However, there were differences in each system in terms of the state change times. Hence, each system was configured by introducing a new service to the existing one. The procedure was repeated 10 times on each client machine and the data showed that the modification of the machine configuration has caused variations in the time required for the servers to switch from ACTIVE to SLEEP mode. The mean of each sample set was determined by plotting on a histogram. A gamma distribution was determined to be an appropriate stochastic model by examining the histogram and using Matlab® to model different distributions with the data obtained by the experiment[10][11].

The gamfit function from Matlab® was used to determine the appropriate gamma distribution modeling parameters, α and β. The results of one of the four server's experimental run is presented in Figure 4. Figure 4 (a) shows a histogram of 6,152 state changes from SLEEP to ACTIVE in the number of seconds taken to complete the state change. For this experiment, the gamfit function produced an α of 23.4543 and a β of .01741. Figure 4 (b) presented the result of the gamma pdf using the α and β coefficients determined from the experiment.

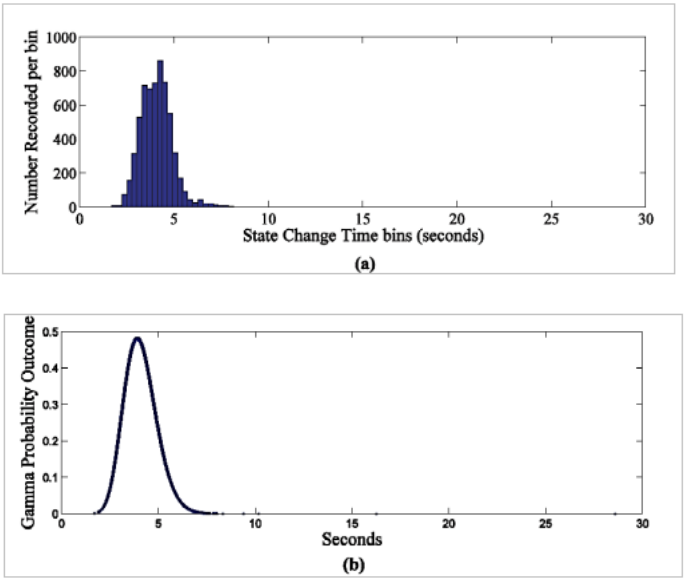


FIGURE 4. a) Histogram of experimental state change time for servers; b) the resulting gamma pdf model

7. CONCLUSION

The determination of the proper time horizon, T for the prediction of the maximum expected future load is the initial step of the optimal VM placement algorithms. A server that operates in an undesirably high regime with applications may expect an increase in the computing loads in the next reallocation cycles. A novel mechanism for forwarding arriving client desires to the utmost suitable server is implemented; thus, in the complete system, balancing the requested load is possible. However, the evaluated systems experienced variations in the state change times. Configuring the systems by introducing a new service to the existing one showed that the modification of the machines caused differences in the required times to switch the servers from ACTIVE to SLEEP modes and vice versa.

REFERENCES

[1] Hasan, R. A., Mohammed, M. A., Salih, Z. H., Ameedeen, M. A. B., pu, N., & Mohammed, M. N. (2018). HSO: A Hybrid Swarm Optimization Algorithm for Reducing Energy Consumption in the Cloudlets.*Telkomnika*,16(5), 2144-2154.

[2] R. A. Hasan and M. N. Mohammed, "A krill herd behaviour inspired load balancing of tasks in cloud computing," *Studies in Informatics and Control*, vol. 26, pp. 413-424, 2017.

[3] M. A. Mohammed, Z. H. Salih, N. pu, and R. A. K. Hasan, "Security and accountability for sharing the data stored in the cloud," in 2016 15th RoEduNet Conference: Networking in Education and Research, 2016, pp. 1-5.

[4] M. A. Mohammed and R. A. Hasan, "Particle swarm optimization for facility layout problems FLPA comprehensive study," in 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), 2017, pp. 93-99.

[5] R. A. Hasan, M. A. Mohammed, N. pu, and O. A. Hammood, "A comprehensive study: Ant Colony Optimization (ACO) for facility layout problem," in 2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet), 2017, pp. 1-8.

[6] M. A. Mohammed, R. A. Hasan, M. A. Ahmed, N. Tapus, M. A. Shanan, M. K. Khaleel, et al., "A Focal load balancer based algorithm for task assignment in cloud environment," in 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2018, pp. 1-4.

[7] M. A. Ahmed, R. A. Hasan, A. H. Ali, and M. A. Mohammed, "The classification of the modern arabic poetry using machine learning," *Telkomnika*, vol. 17, pp. 2667-2674, 2019.

[8] R. A. Hasan, I. Alhayali, A. Royida, N. D. Zaki, and A. H. Ali, "An adaptive clustering and classification algorithm for Twitter data streaming in Apache Spark," *Telkomnika*, vol. 17, 2019.

[9] R. A. Hasan, M. N. Mohammed, M. A. B. Ameedeen, and E. T. Khalaf, "Dynamic Load Balancing Model Based on Server Status (DLBS) for Green Computing," *Advanced Science Letters*, vol. 24, pp. 7777-7782, 2018.

[10] E. T. Khalaf, M. N. Mohammad, K. Moorthy, and R. A. Hasan, "Biometric Template Protection based on Hill Cipher Algorithm with Two Invertible keys," *Advanced Science Letters*, vol. 24, pp. 7643-7649, 2018.

[11] M. A. Mohammed, A. A. Kamil, R. A. Hasan, and N. Tapus, "An Effective Context Sensitive Offloading System for Mobile Cloud Environments using Support Value-based Classification," *Scalable Computing: Practice and Experience*, vol. 20, pp. 687-698, 2019.