International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

Thermal and Power-Aware VM Scheduling on Cloud Computing in Data Center

Abdolkhalegh Balouch
Department of Computer and Information Engineering
Amirkabir University of Technology
Tehran,Iran

Abdolvahed Bejarzahi
Department of Computer and Information Engineering
Islamic Azad University
Nikshahr, Iran,

***Abstract*— Data centers now play an important role in modern IT infrastructures. Related research shows that the energy consumption for data center cooling systems has recently increased significantly. There is also strong evidence to show that high temperatures in a data center will lead to higher hardware failure rates, and thus an increase in maintenance costs. datacenter is often also a Cloud center This paper devotes itself in the field of thermal aware workload and VM placement for Cloud data centers. The goal of this Work is to ameliorate, through scheduling, the energy-efficiency of data center. To support this work a novel VM scheduling mechanism design and implementation will be proposed. The proposed scheduling mechanism on CloudSim will be finally validated, a well-known simulator that models data centers provisioning Infrastructure as a Service. For a comparative study, we also implemented other scheduling algorithms, non power control, DVFS. The experimental results show that the proposed scheduling scheme, combining the power-aware with the thermal-aware scheduling strategies, significantly reduces the energy consumption of a given Data Center because of its thermal-aware strategy and the support of VM migration mechanisms.**

Keywords— Cloud computing , datacenter, thermal and power management, VM scheduling, load balancing

I. INTRODUCTION

High-performance computing data centers have been rapidly growing, both in number and size. Thermal management of data centers can address dominant problems associated with cooling such as the recirculation of hot air from the equipment outlets to their inlets and the appearance of hot spots. Cloud Computing is part of the data centers, a computing paradigm that can provide dynamic and scalable virtual resources through the Internet service to users on demand, and also it is further development of distributed computing, parallel computing and grid computing [1]. Its main advantage is that it can quickly reduce the hardware costs and increase computational power and storage capacity; users can access high quality of service by low cost. And it is unnecessary to purchase expensive hardware equipment, as well as upgrade and maintain frequently. Task scheduling is an important part of cloud computing, which is a mechanism that maps users' tasks to appropriate resources to execute, its efficiency will directly affect the performance of the whole cloud computing environment. Compared with grid computing, there are many unique properties and the mainly include virtualization and flexibility for cloud computing [2]. By using virtualization technology, all physical resources are virtualized and are transparent to user. All user has their own virtual machine and don't affect each other, which is created according to the user's requirement. Furthermore, one or multiple virtual machines can be run on a single host, and the utilization of resources is improved effectively, and the running independency of users' application is ensured as well as the information security of system and service availability is improved.

Various algorithms have been proposed by researchers including those focusing on Energy Consumption. In the field of Cloud Computing, Cloud providers are surely expecting a low operation overhead. Considering the importance of energy saving in the field of Cloud Computing a specific algorithm for scheduling the virtual machines on the Cloud was here developed. This algorithm first performs an initial scheduling and then inspects the change of workload and temperature on the host. In case of over-loading or over-heating, the VM is migrated to another host thereby to avoid hot spots with respect to load and temperature. We implemented this algorithm on CloudSim ,a well-known simulation platform for research work in Cloud Computing. In addition to the proposed scheduling scheme we also implemented several other scheduling policies for a comparative study. The experimental results show that the proposed approach performs better than others with the lowest energy consumption while the service quality is granted in most cases.

this paper is organized as follows: Section 2 introduces the related work . Section 3 describes cloud computing architecture and scheduling model. implementation details in Section 4. Section 5 experimental result and conclusion this paper in Section 6.

II. RELATED WORK

Task scheduling has been investigated for many years and is currently still a hot topic in various research domains. For example, Cluster Computing relies on good scheduling technologies for achieving the performance of parallel applications; Grid Computing requires efficient job scheduler to best utilize the Grid resources and thereby reduce the job waiting time. Over the last years, many research works have been performed for investigating the scheduling strategies. The scheduling of jobs to the compute resources is an NP-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

complete problem in the general form [7]. The problem is NP-complete even in two simple cases: (1) scheduling jobs with uniform weights to an arbitrary number of processors and (2) scheduling jobs with weights equal to one or two units to two processors [7]. Deel man et al. [5] have done considerable work on planning, mapping and data-reuse in the area of workflow scheduling. They have proposed Pegasus [5], which is a framework that maps complex scientific workflows onto distributed resources such as the Grid. DAG Man, together with Pegasus, schedules tasks to Condor system. Kim et al. [10] implement a guest-aware priority-based scheduling scheme, which is specifically designed to support latency-sensitive workloads. The proposed scheduling scheme prioritizes the virtual machines to be allocated by using the information about priorities and status of guest-level tasks in each VM. It preferentially selects the VMs that run latency-sensitive applications to be scheduled and in this way to reduce the response time to the I/O events of latency-sensitive workloads. task scheduling is a NP-Complete problem, Genetic Algorithm (GA) has been used for scheduling workflows [25]. However, GA may not be the best approach. Salman et al. [6] have shown that the performance of PSO algorithm is faster than GA in solving static task assignment problem for homogeneous distributed computing systems based on their test cases. Wang et al.[11] propose a novel VM scheduling algorithm for virtualized heterogonous multicore architectures. The algorithm exploits the core performance heterogeneity to optimize the overall system energy efficiency. It uses a metric termed energy-efficiency factor to characterize the power and performance behavior of the applications hosted by VMs on different cores. The VM's energy efficiency factors are calculated and based on the values virtual machines are mapped to heterogeneous cores with a final goal of maximizing the energy efficiency of the entire system. Takouna et al. [12] also address on the VM scheduling of heterogeneous multicore machines. A scheduling policy is designed to schedule each virtual machine to an appropriate processing core based on the performance sensibility to the CPU clock frequency and the performance dependency on the host.

In addition to the research work on VM schedulers of general purposes, the specific problem of VM scheduling on the Cloud has also been addressed over the last years. Our work is both power and thermal-aware scheduling policies will be combined to reduce the energy consumption. More importantly, the extension of this work to support using temperature constraints as new scheduling parameters. The evaluation on CloudSim has shown the improvement of this approach over the existing one in terms of saving energy consumption. The experimental results will be given after the description of the proposed scheduling algorithm and its implementation.

III. CLOUD COMPUTING AND SCHEDULING MODEL

Cloud Computing Architecture includes three layers, application layer, platform layer and infrastructure layer [4].The application layer is oriented to users, it implements the interaction mechanism between user and service provider with the support of platform layer. Users can submit tasks and receive the results through the application layer in the task scheduling process. The infrastructure layer is a set of virtual hardware resources and related management function, in which the implementation of internal process automation and optimization of resource management can provide optimal dynamic and flexible external infrastructure services. Furthermore, the platform layer is a set of software resources with versatility and reusability, which can provide an environment for cloud application to develop, run, manage and monitor. According to the above architecture, two levels scheduling model [3] are adopted in this paper as shown in Fig.1.

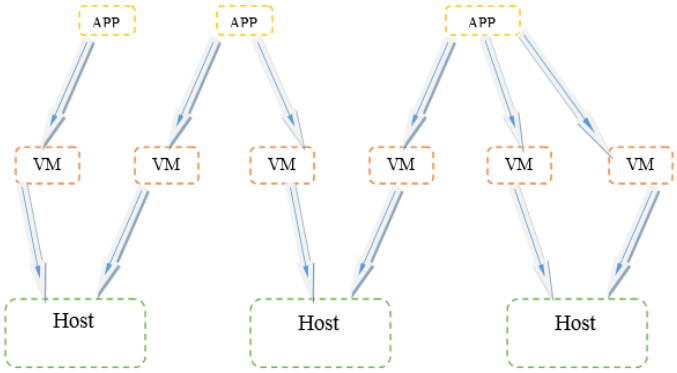

Fig.1 Two levels scheduling Model in cloud computing

As shown in Figure 1, the first level scheduling is from the users' application to the virtual machine, and the second is from the virtual machine to host resources. In this two levels scheduling model, the first scheduler create the task description of a virtual machine, including the task of computing resources, network resources, storage resources, and other configuration information, according to resource demand of tasks. Then the second scheduler find appropriate resources for the virtual machine in the host resources under certain rules, based on each task description of virtual machine. Via the two levels scheduling, the task can obtain the required resources, and it would not lead to the resource allocated to some tasks is less than requirement and increase the implemental time while others are more than their requirements and lead to the waste of resources.

IV. PROPOSED METOD AND IMPEMENTATION

A. Proposed Algorithm

The basis of our algorithm is the scheduling and migration of VM on cloud computing , and in order for our proposed algorithm to achieve the best results, it is a combination of three layers that are as follows :

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

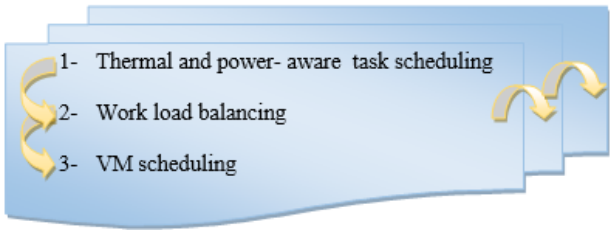

Fig.2 The layers of our algorithm

Our thermal and power-aware scheduling concept is based on several existing strategies, which are applied for the requirement of different scheduling scenarios. These strategies are individually integrated in our framework at the run-time based on the current load and temperature state. The main tasks of our scheduler, called Thermal-aware VM Scheduler (ThaVS), are the following:

Applying the DVFS technique for power management. Since the power consumption depends on the CPU usage, this metric has to be measured before each VM scheduling step in the whole execution process in order to calculate the current consumed energy by the CPU.

The main focus of our ThaVS scheme is to schedule virtual machines with respect to the temperature of the processors. Such a scheduling strategy needs a temperature model to describe the property of this parameter while applications (here virtual machines) are running. The lumped RC thermal model [13] will be used due to its simplicity. The RC model can be described with:

$$T_{amb} = RC\, dT/dt + T - RP \quad (1)$$

where C stands for the thermal capacitance of the processor, R for the thermal resistance, P for the processor dynamic energy consumption, and T_{amb} for the ambient temperature. This model is however limited to a single-core processor. Therefore, ThaVS supports now only single core machines.

Migration of Virtual Machines : Our ThaVS scheme inspects the change of load and temperature on a host machine. In case that a host is about to be over-heated or over-loaded it performs VM migration to avoid critic scenarios. Migration or live migration refers to the process of moving a running virtual machine or application from one physical machine to another. In addition to the machine image, the data storage and network connectivity of the virtual machines have also to be transferred from the source host to the destination. The proposed scheduling scheme implements this kind of VM migration and runs the entire migration process transparently to the user. The migration contains several steps, including the Push phase, the Stop phase, the Copy phase and the Pull phase. The VM migration can take a long time when a VM has a large amount of memory. In order to save the unnecessary energy consumption during the VM migration, two strategies in ThaVS are implemented :

- the Pining strategy that allows the allocation of multiple VMs on the same host to free other physical hosts;
- the energy-save-modus strategy that sets the unused hosts (or CPUs) in the idle mode.

ThaVS decides during the runtime which VM shall be allocated on which host. It works in the following way: As a starting point a VM request coming from the user is scheduled on a physical host based on the traditional round-robin scheme. In order to make a decision, ThaVS calls the thermal model and the power model to acquire all scheduling parameters including the current CPU temperature, the CPU usage for each host, the data center configuration and the application (VM) requirements. In case that a physical host approaches to the critical temperature (Temperature threshold) or the critical CPU utilization value (Utilization threshold), ThaVS looks for another host with better temperature or load criteria and migrates the running VM from the source host to the destination host. Therefore, our ThaVS schedules the virtual machines not only for minimizing the energy consumption but also for load-balancing. ThaVS is designed to be integrated in a Cloud middleware to replace its original one, usually a round-robin scheme that allocates the VMs to each host equally in a cyclic order. ThaVS implements an interface between the VM (hypervisor) and the virtual machines in a Cloud center. It replaces the conventional scheduling algorithm of a hypervisor to map a virtual machine request to a physical machine with consideration of the load and temperature on the hosts. The deployment, such as start, stop, migration, etc., of the virtual machines on the physical host remains the task of the hypervisor. In this way, our scheduler acts as an allocation decision component for the hypervisor.

B. Implementation

A Java-based simulation environment, to verify the concept and to validate the functionality of the proposed scheduling strategies for ThaVS was implemented .The prototypical implementation is based on a well-known simulator, CloudSim [9], for research work on the Cloud. CloudSim models large scale data centers with internal broker, sensors, physical host, as well as virtual machines. CloudSim also provides the functionality to configure datacenter resources. By an incoming VM request our scheduler ThaVS starts first with an initial task of allocating the VM to one of the hosts that meet the requirement described in the request. In the following, it performs the runtime tuning with respect to load and temperature. steps of our algorithm as follow:

1- The first task of ThaVS for runtime tuning is to detect the critical hosts with either higher temperature or workloads. This allows us to model cloud providers with different system scale and hardware resources.
2- In the second step, the list of critical hosts, which has been created by the scheduler in the first step, is processed again for finding the VMs running on

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

them. These virtual machines are the concrete candidates for migration.

3- The third and last step of the VM migration procedure is to find an appropriate target host for hosting the migration objects in the list created in the last step.

The candidate VMs in step 2 are then sorted by their CPU usage .The VMs with minimal CPU usage are marked with a higher priority of migration for the reason of not to bring high workload on the target host, thus to avoid possible further migrations.

We modeled a scenario of scheduling the virtual machine requests of three days with a randomly generated VM request every five minutes. A VM request contains four requirement attributes, i.e., the number of CPU cores, the CPU frequency, the RAM size and the bandwidth. The properties of the modelled VM types are described in Table 1. The bandwidth and VM size for all simulated virtual machines are set as 100 Mbit/s and 2.5 GB individually.

TABLE I. VIRTUAL MACHINE CONFIGURATION.

Number	VM Type	VM_MIPS	VM_RAM[MB]
1	1	500	643
2	2	1500	1890
3	3	2000	1890
4	4	2500	870
5	VM_Cores	VM_BW[Mbit/s]	VM_Size[GB]
6	1	100	2.5

V. EXPRIMENTAL RESULTS

As described above, we use a temperature threshold to limit the temperature of the processors. This means that in case of higher temperature than the threshold some workload on the host has to be moved to other hosts in order both to avoid hardware defect and for energy reasons. Therefore, our first three experiments were done for studying the impact of the Temperature threshold using a number of values ranging from 333 Kelvin to 360 Kelvin.

Figure 3 shows the results of the first experiment, where we measured the energy consumption of the two simulated data centers in a single day using our VM scheduler. The x-axis of the figure demonstrates the thresholds, while the y-axis presents the energy consumption in Kwh. For each temperature threshold there are two values with one for the first data center (DC-1) with 50 hosts and the second for the other one (DC-2) with 250 hosts. The data are presented in two forms, one in bars and the other in curves. Observing the curves in the figure, it can be seen that the energy consumption with different temperature thresholds can be divided into four areas. In the first area, the temperature threshold between 333 and 335, the energy consumption maintains constant with a value of 52 Kwh for DC-1 and 102 for DC-2. In the second region, 335 < temperature threshold < 340, the energy consumption goes down as the threshold getting larger. This behavior stops at threshold 340, where a new phase starts with again a constant energy consumption. After this phase, a slight reduction in energy consumption can be seen but generally the temperature threshold does not change the energy behavior in phase 4.

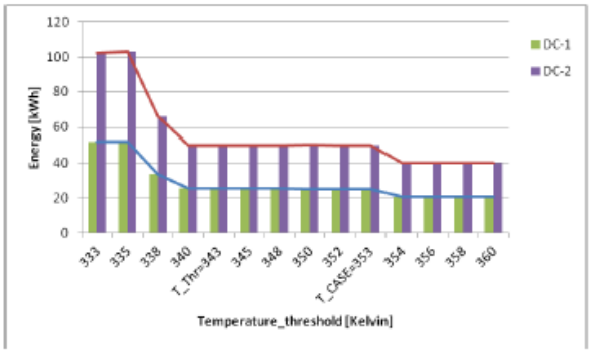

Fig.3 Energy consumption.

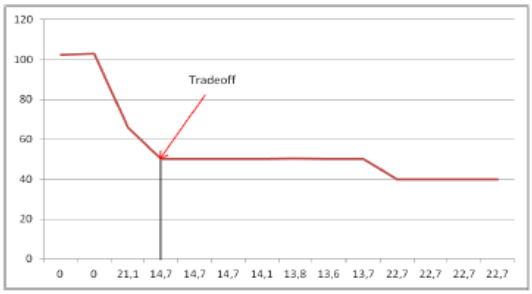

Fig.4 Optimal trade-off for the temperature threshold (x-axis: SLA violation, y-axis: energy consumption)

In order to give a clearer view about this optimal threshold of temperature, we made a diagram with the energy consumption and the SLA violation. Figure 4 depicts this graph, where the data are from the tests with the second data center. The x-axis shows the SLA violation while the y-axis presents the energy consumption. Observing the graph in the figure, it can be seen that there is a point where both the energy and the SLA violation are low. This point(threshold 343 Kelvin) is exactly the optimal threshold we are looking for. Hence, we selected 343 Kelvin as the optimum value of the Temperature threshold for our ThaVS scheme.

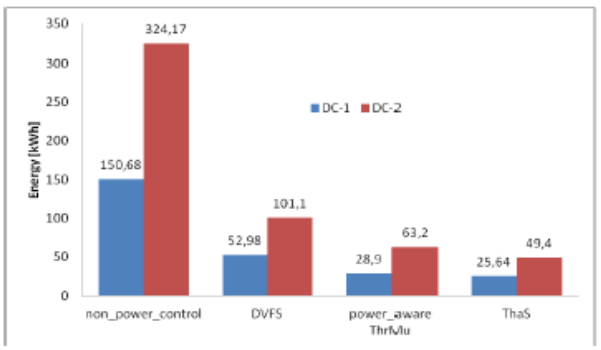

Fig.5 Comparison of ThaVS with other scheduling schemes.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

The last experiment studies the feasibility of our scheduler by comparing the energy consumption with ThaVS and with other three scheduling schemes, i.e., non power control, DVFS and power aware ThrMu. Figure 5 depicts the result of the experiment, where the energy consumption was measured during a single simulation run with all four algorithms. We measured the energy consumption of the two data centers in a single day. Comparing ThaVS with the other scheduling algorithms, it can be observed that ThaVS achieves the lowest energy consumption with a value of 25.64 Kwh with DC-1 and 49.4 Kwh with DC-2. The energy consumption with other schemes are Non power control of 150.68 and 324.17, DVFS of 52.98 and 101.1, and Power aware ThrMu of 28.9 and 63.2. For the first data center with 50 hosts we achieved a reduction of 83% to Non power control, 52% to DVFS and 11.3% to power aware ThrMu. For the other data center with 250 hosts the reduction in energy consumption by ThaVS is 85% to non power control, 52% to DVFS and 21.8% to power aware ThrMu. It can be seen that ThaVS results a higher reduction in energy consumption for larger data centers than the smaller ones, especially in comparison with the power-aware scheduling scheme ThrMu. Overall, we can conclude.

VI. CONCLUSION

High-performance cloud computing data have been rapidly growing, both in number and size. Thermal management of data centers can address dominant problems associated with cooling such as the recirculation of hot air from the equipment outlets to their inlets and the appearance of hot spots. As a result, researchers are investigating on various methods to reduce the energy requirement of a computer system. This work targets on a scheduling algorithm aiming at allocating virtual machines to physical hosts of data centers in such a way that the target host will not be overloaded or over-heated. This means that we schedule virtual machines with respect to the temperature and CPU utilization of processors. We validated the novel scheduler on a simulation environment and compared our achievement with several other scheduling schemes. The experimental results show a clear benefit with our scheduler.

REFERENCES

[1] Rimal, B.P., Choi, E, Lumb, " A Taxonomy and Survey of Cloud Computing Systems." In: Fifth International Joint Conference on INC,IMS and IDC, vol. 218, pp. 44–51 ,2009.

[2] Foster, I., Zhao, Y., Raicu, I., Lu, S.: "Cloud Computing and Grid Computing 360-Degree Compared. " In: Grid Computing Enviroments Workshop, pp. 1–10 ,2008.

[3] Sadhasivam Dr., S., Jayarani, R. " Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment." In: International Conference on Advances in Recent Technologies in Communication and Computing, vol. 148, pp. 884–886 ,2009.

[4] Armbrust, M.: " Above the Clouds: A Berkeley View of Cloud Computing. In" : EECS Department, University of California, Berkeley ,2009.

[5] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: " A framework for mapping complex scientific workflows onto distributed systems." Sci. Program., 13(3):219–237, 2005.

[6] A. Salman. "Particle swarm optimization for task assignment problem". Microprocessors and Microsystems, 26(8):363– 371, November 2002.

[7] J. D. Ullman. "Np-complete scheduling problems." J. Comput. Syst. Sci., 10(3), 1975.

[8] J. Yu, R. Buyya, and K. Ramamohanarao. "Workflow Scheduling Algorithms for Grid Computing " , volume 146, pages 173–214. Springer Heidelberg, 2008.

[9] Calheiros, R. N., Ranjan, R., Beloglazov, A., e Rose, C. A. F. D., and Buyya, R.: "CloudSim: A Toolkit for Modeling and Simulation of Cloud Comp uting Environments and Evaluation of Resource Provisioning Algorithms". Software: Practice and Experience, 41(1):23–50,2011.

[10] Kim, D., Kim, H., Jeon, M., Seo, E., and Lee, J. Guest-Aware Priority-based Virtual Machine Scheduling for Highly Consolidated Server. In Proceedings of the 14th International Conference on Parallel and Distributed Computing (Euro-Par 2008), pages 285–294, 2008.

[11] Kolodziej, J., Khan, S., Wang, L., Kisiel-Dorohinicki, M., and Madani, : "Security, Energy, and Performance-aware Resource Allocation Mechanisms for Computational Grids". Future Generation Computer Systems.2012.

[12] Takouna, I., Dawoud, W., and Meinel, C. "Efficient Virtual Machine Scheduling-policy for Virtualized heterogeneous Multicore Systems". In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications,2011.

[13] Skadron, K., Abdelzaher, T., and Stan, M. R. " Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management". In Proceedings of the 8th International Symposium on High-Performance Computer Architecture, HPCA '02, pages 17–27, Washington, DC, USA. IEEE ,Computer Society.2002.

[14] Tang, Qinghui, Sandeep Kumar S. Gupta, and Georgios Varsamopoulos. "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach." IEEE Transactions on Parallel and Distributed Systems 19, no. 11 : 1458-1472.2008.

[15] Kolodziej, J., Khan, S., Wang, L., and Zomaya, A. Energy Efficient Genetic-Based Schedulers in Computational Grids. Concurrency and Computation: Practice & Experience. 2013.