ATLANTIS PRESS
Advances in Computer Science Research (ACSR), volume 76
7th International Conference on Education, Management, Information and Mechanical Engineering (EMIM 2017)

The Energy Optimization Based on Virtual Machine Placement

Shoujin Wang [1, a *], Xiaotong Cheng [1 , b], Jingang Shi [1,c]
School of Information and Control engineering, Shenyang Jianzhu University, Shenyang, China
[a] 23917240@qq.com, [b] 470022675@qq.com , [c] 8269086@qq.com

Keywords: Resource Management; Big Data; Vitalization; Energy Optimization

Abstract. In order to solve the problem of huge energy consumption in data centers, this paper delivers a new virtual machine placement framework, including profiles and task classification. The first part of this framework is profiling phase. This phase intent to build virtual machine profiles, to identify each virtual machines run-time, resource requests, and previous utilization. The profiles are based on past virtual machine logs and coming virtual machine plans. According to these profiles, the system related to this framework will make better decisions for virtual machine placements. The second part is virtual machine classification phase. Based on some typical characteristics of virtual machines in the profiles built in last phase, the system will apart virtual machines into some given sorts. Each sort of virtual machines will have differential placement methodologies. The third part is virtual machine placement phase. All information of virtual machines, including profiles and sorts, will come to this phase. Then, the system will, accordingly, conduct typical first-fit-decrease algorithm to build a virtual machine placement plan, to enhance the power efficiency.

Introduction

Problem Statement. With the lack of power-aware server management, more electricity could outflow continuously without any production . This issue has raised some far-reaching problems, such as huge operating cost in data centers , bottleneck of virtual machine performance by power delivery challenge , cooling and fireproofing trouble , and negative environment impacts of significant carbon dioxide emission and natural resource consumption . However, this problem is not only related with the infrastructure itself , but also strongly with the deployed infrastructure managing methods . As a result, some feasible measures are highly required to be taken , to improve the energy efficiency in this server-management field . Recently, data centers have already started to explore improvement in flexibility and usability by virtualization technology . Many computing service providers such as Google , Microsoft , Amazon , and IBM are rapidly implementing their data centers into highly virtualized environment , and provide cloud service through this virtualized platform , without adequate awareness on power usage ratio as well [1]. According to this urgent circumstance background, this research topic would also examine possible virtual machine management methods for energy optimization to help relieve these problems.

Research Objectives. This research topic involves examining strategic frameworks of some complex existing virtual machine scheduling for possible improvement , and implementation on one certain virtualized platform of differential data centers .How to form profiles for virtual machine placement for power-aware virtual machine placement methodology ;How to classify virtual machines into appropriate sorts to help improve power-aware virtual machine placement ;How to place virtual machines according to the results of profiling and classification results to achieve better energy efficiency and service level agreements .

According to these objectives, this research would aim to examine the power efficiency of some virtual machine scheduling strategies and frameworks for improvement. More specifically , in the future , this research would aim to develop and implement one optimized framework in one certain virtual environment hyper vising system.

This paper determines to formulate and deliver a new virtual machine placement framework using profiling methodology for energy optimization in data centers. This framework includes three stages as profiling, virtual machine classification, and virtual machine placement. Through deployment of this framework, a data center will achieve better energy efficiency with less migrations and guaranteed service level agreements.

Model and Design

Energy consumption model here is used to evaluate the power efficiency of one virtual machine placement solution. In this model, entire energy consumption in one data center, which will be affected by the methods introduced in this paper, will be calculated. Furthermore, each virtual machine placement plan should be related to one supposed energy consumption. Thus, by comparing the energy consumption of each plan, the plan combined with least energy consumption will be found and chosen as the best plan to implement. Normally, the energy consumption in one data center includes quite a number of ingredients, like PMs, administration system's, and air-conditioning system's power costs [2]. Regarding E for energy consumption, the energy consumption in one data center will be identified as:

$$E_{\mathrm{DataCenter}} = E_{PM} + E_{AdminSystem} + E_{Air-con} + \ldots \tag{1}$$

As power consumption like administration system's and air-conditioning system's are almost fixed through virtual machine placement plans, it is assumed that those ingredients other than PMs power cost will be emitted during evaluation. Moreover, for each PM, its power is consumed by different physical parts, like CPUs, memory, and disks. Which will be described as:

$$E_{PM} = E_{CPU} + E_{Memory} + E_{Disk} + \ldots \tag{2}$$

Similar as above, assuming that power consumption through parts like memory and disks are fixed, such parts will be emitted during evaluation. Nevertheless, by scaling CPU frequency and virtual machine placing mentioned in research questions, the CPU power might greatly change among different placement plans. Therefore, the energy consumption evaluation model should concentrate onto CPUs power of each PM. Consequently, the energy model for evaluation will be described as a sum of all CPU power in every PM:

$$E_{Evaluation} = \sum E_{CPU} \tag{3}$$

Commonly in Traditional physics, energy cost in a certain period of time is the integration of power via time. Using P for power, this will be written as:

$$E = \int P(t) \tag{4}$$

Therefore, for each CPU, its energy cost will be the integration of its power via its runtime:

$$E_{CPU} = \int P_{CPU}(t_{CPURuntime}) \tag{5}$$

This equation shows that during evaluation for each virtual machine placement solution, power of every CPU and its run-time are acquired. The following paragraphs will introduce methodologies applied to P_{CPU} and $t_{CPURuntime}$.

Virtual Machine Placement Model

Virtual machine placement model is used to mathematically define each placing decision, and the final unite of all single placing decisions [3].

It has given a mathematical model for virtual machines. According to this model, the unite of virtual machine is scheduling decisions will be:

$$op_i = \{p, m\} \tag{6}$$

Here, op_i is the scheduling decision for VM_i. In addition, p is for the placement of this virtual machine, and m is the migration of it. As this paper focus on the placement measures, m will be always ϕ in this paper.

Furthermore, the unite of all virtual machines decision in PM_j will be defined as:

$$OP_j = \bigcup\{op_i\} \tag{7}$$

In addition, every virtual machine always includes components as: scheduling options op, resource request r, start time t_{Start}, end time t_{End}, and the target node PM. Thus, one virtual machine will be defined as:

$$vm_i = \{op, r, t_{Start}, t_{End}, PM\} \tag{8}$$

Supposing there are m virtual machines in total, then all the virtual machines will be:

$$VM = \bigcup_{1 \le i \le m} \{vm_i\} \tag{9}$$

Energy Optimization Objective

The goal of this research topic is to find the best solution for multiple virtual machine placement of energy optimization with frequency scaling. In other words, this goal is to find a virtual machine placement plan combined with frequency scaling, to achieve the least energy consumption [4].

Profile Building

As mentioned in previous part, the profile of coming experiments will be in two parts , virtual machine part and job part. Each part will be formed in .csv file . The design of virtual machine part is displayed in the following table.

Table 1 virtual machine profile design

Content	Detail
Start Time	The start measurement time of this record
End Time	The end measurement time of this record
Job ID	The ID of related job
Task Serial	The serial number of related task
Machine ID	The ID of related local PM
Peak CPU Usage	The peak CPU usage of this task in this measurement period
CPU Request	The CPU resource request when this task is submitted

For start & end time , if this task starts before relevant measurement period , or ends after it , its Start time or End Time will be the nearest time stamp. Moreover, for CPU usage & request, Peak CPU Usage might be greater than CPU request, which means this task consumes more computation resources than requested [5].

The design of job part is displayed in the following table.

Table 2 Job profile design

Content	Detail
Start Time	The start time of this job
End Time	The end time of this job
Job ID	The ID of related job
Task CPU Request	The CPU request for each task included this job
Number of Tasks	The number of tasks included in this job

For start & end time, one job might be recorded in multiple measurement periods. As a result, this profile has to indicate whether one job repeats or runs permanently. If one job repeats, there will be two records in this profile of this job, showing different lifetime.

Experiments

This experiment of profiling uses Google Cluster-Usage Traces as original data sets. However, as this data set is too huge for proper experiments, only parts of them are used in this experiment. Firstly, these data sets show the records of one data center in one-month-time, which is too long to build a daily profile. As a result, parts 10 to 27 of 500, which indicates about 24-hours records are chosen to be used in this experiment. Secondly, the original data sets include 12500 PMs and millions of tasks, which overflows for medium data centers and this experiment. Eventually, parts 10 to 27 have been sampled. This experiment collects one every two hundred record as the final data sets.Table3 shows a sample part of the virtual machine profile.

Table 3 Segment of virtual machine profile sample

Start Time	End Time	Job ID	Task Serial Number	Machine ID	Peak CPU Usage	CPU Request
51000000000	51300000000	1106173310	41	12977617	0.008255	0.03125
51000000000	51300000000	1412625411	91	376218267	0	0.0625
51000000000	51300000000	1836173247	130	1436304298	0.001917	0.0625
51000000000	51300000000	2024770216	6	6219933940	0.04004	0.0625
51000000000	51300000000	2437916070	36	329139490	0.09436	0.03125
51000000000	51300000000	2509801309	0	1094366	0.01491	0.1875
51000000000	51300000000	2780813826	3	8053587	0.02258	0.03125
51000000000	51300000000	2902878580	149	854237645	0.09595	0.006248

Placement

Firstly, allocating long-term virtual machines saves search area and energy. According to the analysis results in last two chapters, long-term, especially permanent jobs or tasks often acquire more computing resources than those last more shortly. Therefore, arranging these long-term jobs or tasks, or relevant virtual machines into PMs with better power efficiency, will indirectly help limit power consumption. Moreover, place long-term virtual machines in advance will help reduce migrations, which usually consume more energy. By place them in advance, the searching space of PMs gets smaller accordingly for the requests are smaller than all virtual machines' requests. Therefore, those long term virtual machines will be assigned into fewer PMs, which also stands for fewer possible PMs to be switched off in the future. Thus, in a long-term view, less possible migration will occur since a more limited number of assigned PMs.

Secondly, packing multiple regular tiny tasks into one virtual machine saves total virtual machine utilization. Compared to assigning separate virtual machines to each tiny task, packing those tasks together will save possible utilization, which is the gap between task request and the size of a virtual machine. As a result, the total utilization of virtual machines will be restricted. Finally, less PMs will probably be requisitioned and energy will be saved [6].As shown in the Fig. 1 .

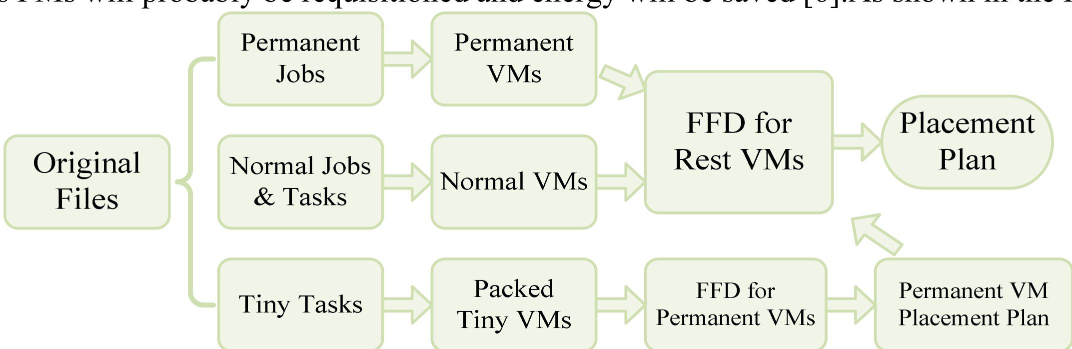

Figure 1. virtual machine placement architecture.

Stability refers to whether the system could work well with various scaled data set. More nearer the relationship between data set scale and processing time to linear, the stronger stability the framework will have.

At last, diagrams of the physical machine quantity and physical machine utilization in every time slot and power consumption will be generated. Through the comparison between the two approaches, the one with fewer physical machines, higher physical machine utilization and less time cost will be the higher effective approach.

The advanced first-fit-decrease algorithm used in this paper is based on the original first-fit-decrease algorithm. This framework determines to solve permanent, normal, and tiny tasks separately. First of all, according to the classification result, each permanent task will be assigned into one virtual machine, and be well placed through first-fit-decrease algorithm. Then, all other tasks will be placed. System keeps allocating tasks one-by-one through first-fit-decrease algorithm until all ideal computing resources in allocated PMs are available to all rest tasks. These rest tasks will be treated as tiny tasks automatically. At this stage, ideal virtual machines will be established in all allocated PMs, and their sizes just fit the ideal computing resource. Tiny tasks, whose lifetime is often even shorter than the duration of establishing and killing a virtual machine, will be assigned into these virtual machines rather than being assigned into one virtual machine for each. After all tasks has been well placed, system outputs a placement plan and stops [7].

Conclusions

This paper has introduced a new energy-aware virtual machine placement framework implementing profiling, classification, and first-fit-decrease algorithm. This new framework helps data centers in all different scales save energy through better resource allocation methodology, while guaranteeing QoS. It has combined all advantages and disadvantages of recent methodologies together, and finally formed this new framework. The result has shown that this new framework was able to save about 10% or more energy than existing first-fit-decrease algorithm. This new power-aware virtual machine placement framework will make enterprises more comfortable when establishing a new data center. It will help such data centers complete better cost control and keep greater quality of service.

Acknowledgement

This work was supported by the Liaoning Provincial Social Planning Fund (L15BGL017) and science and technology program of Liaoning (20170540767).

References

[1]Kallrath J, Pardalos P M, Rebennack S, et al. Optimization in the Energy Industry[J]. Energy Systems, 2015, 5:xx+533.

[2]Zhou L, Li J, Li F, et al. Energy consumption model and energy efficiency of machine tools: a comprehensive literature review [J]. Journal of Cleaner Production, 2015, 112:3721-3734.

[3]Li X, Qian C. Traffic and failure aware VM placement for multi-tenant cloud computing[C]// IEEE, International Symposium on Quality of Service. IEEE, 2016:41-50.

[4]Fazlollahi S, Becker G, Ashouri A, et al. Multi-objective, multi-period optimization of district energy systems: IV – A case study[J]. Energy, 2015, 84:365-381.

[5]Luo C, Bai Y, Chen T, et al. A Functional Classification Based Inter-VM Communication Mechanism with Multi-core Platform [J]. 2009, 6:332-339.

[6]Do A V, Chen J, Wang C, et al. Profiling Applications for Virtual Machine Placement in Clouds[C]// IEEE International Conference on Cloud Computing. IEEE Xplore, 2011:660-667.

[7]Chen G, Kerre E E, Vandenbulcke J. A computational algorithm for the FFD closure and a complete axiomatization of fuzzy functional dependency (FFD)[J]. International Journal of Intelligent Systems, 1994, 9(5):421-439.