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

The Approach to Profile the Data Logs for Energy Saving

Shoujin Wang [1, a *], Xiaotong Cheng[1, b], Song Guo[1, c]
School of Information & Control engineering, Shenyang Jianzhu University, Shenyang, China
[a] 23917240@qq.com, [b] 470022675@qq.com , [c] 79510772@qq.com

Keywords: Virtual Machines; Physical Machine; Profiling Data; First-fit-decrease

Abstract. The paper aims at developing a system to profile the data logs for virtual machines as well as virtual machine placement respect to energy consumption. A framework is designed for data profiling and virtual machine placement, each of the profile-based framework is described. It provides a clear structure of profiling, tasks classification and virtual machine placement, emphasize the improvement from original first-fit-decrease. The final step is to evaluate the performance of the approach from the feasibility and stability two aspects.

Introduction

Cloud computing, as a kind of distribution computation, has been widely applied in big data manipulation through the Internet. Data center is a common form to store data for cloud computing, which is a repository that contain multiple servers and communication equipment [1,2], commonly run by large companies or government agencies, have facilities of cables, Internet connections and power supplying, cooling systems and security systems [3].

With the network data volume increasing, energy consumption has become a critical issue for data centers. It is reported that from 2006 to 2014, the U.S. data center electricity use has increased from 62 billion kWh to 70 billion kWh and the figure is predicted to keep increasing slightly to 72 billion in 2020 [4]. Such huge amount of energy occupies roughly 2 percent of total U.S. electricity usage [5] and could support the whole New York residents for more than one year [6]. By the standard of $10-$25 per watt [7], the total cost on U.S. data centers will be over $1000 billion.

However, the utilization of data centers is not optimistic. With an only 12 to 18 percentage average server utilization and 30 percent of servers are faced of the risk of being idled [8], the low utilization has caused a huge waste in both energy consumption and money expense. As a consequence, extra expenditure has been spent on cooling device and the worldwide and 50 metrics tons of carbon dioxide pollution have been emitted annually.

Infrastructure and severs are two major sections that lead to huge energy consumption. In the Information Communication Technology Data centers, both of infrastructure and servers had consumed more than 1/3 total energy in 2011. Facebook has designed an improved conventional data center power supply chain system [9]. This improvement indicates it is feasible to make changes to the internal structure of a data center.

This paper is aiming at developing a framework to profile the data logs for virtual machines and make virtual machine placement plan. The performance of framework should be evaluated from physical machine quantity, physical machine utilization, energy saving and time cost. As the paper is depended on first-fit-decrease algorithm and have improvements on several areas, the results will be compared with original first-fit-decrease.

The Related Work for Virtual Machines Placement

Virtual machines placement is the process to map virtual machines on physical machines, selecting which virtual machines should be placed to each physical machine, considering of physical machine capacity and CPU utilization. It is one of the challenging problems for data centers in cloud infrastructure management.

In virtual machines placement, each arriving task must be set to one virtual machines and each virtual machines must be placed to one physical machine. Empty virtual machines can be destroyed or kept for later use. Empty physical machines can be switched off.

virtual machines placement is a kind of bin-packaging problem and first-fit decreasing (first-fit-decrease) algorithm is a common used algorithm in virtual machines placement. With first-fit-decrease algorithm, virtual machines and physical machines are sorted by size in descending order before the placement. When a virtual machines arrives, the system will check each physical machine and place the virtual machines on the first physical machine with enough capacity.

physical machine quantity, physical machine energy consumption, resource consumption and launching time are the key aspects that affect virtual machines placement. Combing tasks to one virtual machines is an effective method for saving resource consumption. However, large-sized virtual machines may restrict launching time and increase physical machine quantity. Hence, it is important to find a balance among these aspects.

Profiling

The first phase of the framework is profiling. It is used for optimizing allocating the request of repetitive tasks or virtual machines. In this phase, the system analyzes 18 task resource usage table files and the related task event table files to make total duration of the tasks in these files around 24 hours, samples data centers into three scales: Large scale, medium scale and small scale.

The first part of this phase is virtual machine profiling, as well as data centers sampling. Large-scale data set stores the data extracted from origin data files one line in every 200 lines. The line intervals for medium-scale data set and small-scale data set are 400 and 1000. Data attributes used for profiling include start time, end time, job ID, task ID from and CPU request (see table 1). The time span of the data is between 50700000000 and 141000000000, about 24 hours from the beginning to the end. Through virtual machine profiling, it is directly to have a view of total CPU request in each time slot. The figures can help to verify sampling correctness and predict physical machine quantity for virtual machine placement.

Table 1 Details of analyzed data features

Field	Explanations	Source table
Start time	Start time of the task, in the form of million second	Task resource usage
End time	End time of the task, in the form of million second	Task resource usage
Job ID	The ID of the job, numeric	Task resource usage
Task ID	Task index of the job, numeric	Task resource usage
CPU request	Request for CPU cores	Task events

The second part is job profiling. Job profiling is to collect job duration and task quantity of each job. Tasks can be considered as repetitive tasks when the span of end time in the previous time slot and the start time in the next time slot is within 300 seconds. If the duration of all tasks in one job covers the whole time slot, the job can be regarded as permanent jobs. Permanent job information will be stored into a file and it is used for task classification in next phase.

Tasks and Virtual Machine Classifications

The second phase is tasks & virtual machines classification. Tasks are the minimum units of the data. Different tasks have different lifetime and different CPU request. has cited a special profile-based task assignment methodology . The main purpose of this phase is similar to the ideas from the methodology, aiming at classifying tasks into different categories and assigns all the tasks to different sized virtual machines.

In the previous phase, a permanent job list is generated. Permanent jobs contain a group of tasks. As the lifetime of the job is long, these kinds of tasks are usually key services with relative high CPU capacity that should be running in the whole procedure without interruptions. They should be placed in stable virtual machines. Meanwhile, there is a certain amount of tasks with tiny CPU request and brief lifetime. It is unrealistic to create virtual machines with such tiny CPU capability. If each of the tiny tasks is assigned to a larger sized virtual machine, there will be lots space left idled that decreases CPU utilization. Thus, these tasks can be packaged together until the total CPU request reaches the smallest virtual machine CPU capacity. The rest of the tasks are normal tasks. These tasks have medium CPU request and modest lifetime.

There are several kinds virtual machines with different CPU capacities to place tasks. The minimum CPU capacity is 0.015, as same as the threshold of CPU request of tiny tasks. The highest CPU request of the tasks is 0.4026, so the maximum CPU capacity of virtual machine is set to 0.45. All kinds of virtual machines are shown in table 2. Each task in permanent tasks and normal tasks is assigned to one virtual machine based on its CPU request. Each tiny task package is assigned to one tiny virtual machine.

Table 2 virtual machines with different CPU capacity

	Huge virtual machine	Large virtual machine	Medium virtual machine	Normal virtual machine	Small virtual machine	Tiny virtual machine
CPU Capacity	0.45	0.3	0.15	0.1	0.045	0.015

The procedure of tasks and virtual machines classification is shown in figure 1:

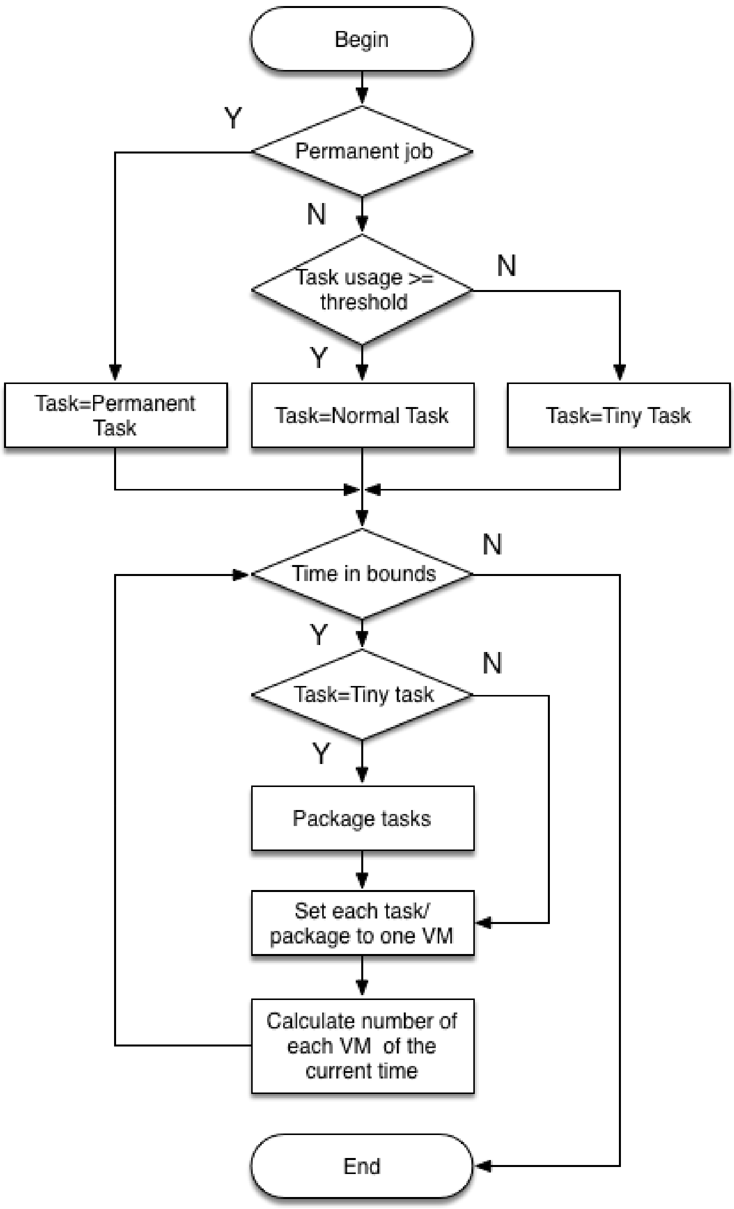

Figure 1 Tasks & virtual machines classification procedure

Acquire job ids from the permanent job file and read each line from the tasks files. Tasks with a permanent job id will be regarded as permanent tasks. For the rest of the tasks, if their CPU request is less than the threshold, 0.015, they will be classified as tiny tasks. Otherwise, the tasks are normal tasks. For the tasks in each time slot, package tiny tasks. Assign each of permanent tasks, normal tasks and packages to one virtual machine. Calculate the each virtual machine quantity at the end of each time slot.

Virtual Machine Placement

The third phase is virtual machine placement. The aim of this phase is to place the virtual machines generated from the previous phase to physical machines.

Considering of the variable sizes of CPU requests, the physical machines in data centers should have different CPU capacities. The configuration of physical machines and physical machine quantity for the three scaled data set are shown in table 3 and table 4.

Table 3 physical machine configurations

	High Performance physical machine	Medium Performance physical machine	Low Performance physical machine
CPU Capacity	1.5	1.0	0.5
Energy efficiency	High	Medium	Low
Max Power	220	250	280
Min Power	160	160	160

Table 4 physical machine quantities for three scaled data sets

	High Performance physical machine	Medium Performance physical machine	Low Performance physical machine
Large-scale	10	15	300
Medium-scale	6	9	200
Small-scale	0	2	100

CPU capacities of different physical machines are normalized as 1.5, 1.0 and 0.5. The amount of high performance physical machine and medium performance physical machine are fixed while the quantity of low performance physical machines can be any. High performance physical machine and medium performance physical machine are used for placing permanent virtual machines. The physical machine energy efficiency is co-related to the performance.

The placement period is 300 seconds. Data features, virtual machine and physical machine configurations for these two approaches are the same. In new framework, permanent tasks placement is prior to normal tasks and tiny tasks placement, while there is no task priorities in original first-fit-decrease. In original first-fit-decrease, there is no tiny task packaging. One tiny task is assign to one virtual machine.

Placing permanent tasks prior of other tasks leads to permanent tasks being placed to high performance physical machines. Meanwhile, tasks in permanent jobs usually have high CPU request, high CPU capacity physical machines

Place permanent tasks. For each time slot, order virtual machines and physical machines by size from the largest to the smallest. The size of physical machine refers to its free CPU capacity. Place virtual machines to physical machines with first-fit-decrease algorithm. For each virtual machine, check the free capacity of each physical machine and place the virtual machine to the first-fit physical machine. If there is no free capacity for all existing physical machines, create a new physical machine and place the virtual machine on the physical machine. After placing all virtual machines with permanent tasks, place the rest virtual machines with normal tasks and tiny tasks. Order virtual machines and physical machines. Place virtual machines to physical machines with the same method.

Finally, a virtual machine placement plan can be generated. According to the information of virtual machine placement, the physical machine quantity used in each time slot can be acquired directly. Utilization of each physical machine can be calculated by total usage dividing total capability. Then calculates an average value as the average physical machine utilization. These data will be used for evaluation.

Evaluation

The final step is to evaluate the performance of the approach . The evaluation standard is planned to include two aspects: feasibility and stability. Feasibility refers to whether the framework could have an obvious improvement of energy consumption saving and physical machine quantity saving, as well as a short computation time. The energy consumption is described as a sum of CPU power of each physical machine in each time slot. Computation time is not as important as the other aspects, as long as in an acceptable range.

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.

Conclusions

This paper describes the approaches of each phase of the profile-based framework, provides a clear structure of profiling, tasks classification and virtual machine placement, emphasize the improvement from original first-fit-decrease. Then describes the steps of evaluation, states several key features to be analyzed in the next section.

Acknowledgement

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

References

[1] J, Shuja., K, Bilal., S, A, Madani., & S, U, Khan. (2014). Data center energy efficient resource scheduling. 17(4). 1265-1277. doi:10.1007/s10586-014-0365-0

[2] M, D, Hill. (2009). The Datacenter as a Computer. doi: 10.2200/S00193ED1V01Y200905CAC006

[3] M, Arregoces & M, Portolani. (2003). Data Center Fundamentals. Retrieved from https://books.google. com.au/books?id=DRIryrLoxKkC

[4] A, Shehabi., S, Smith., D, Sartor., R, Brown., & M, Herrlin. (2016). United States Data Center Energy Usage Report. Retrieved from http://eta.lbl.gov/sites/all/files/lbnl-1005775_v2.pdf

[5]P, Delforge. (2016). The Power of Efficiency to Cut Data Center Energy Waste. Retrieved from https://www.nrdc.org/experts/pierre-delforge/power-efficiency-cut-data-center-energy-waste

[6] P, Delforge. (2014). New Study: America's Data Centers Consuming -- and Wasting -- Growing Amounts of Energy. Retrieved from https://www.nrdc.org/experts/pierre-delforge/new-study-americas-data-centers-consuming-and-wasting-growing-amounts-energy

[7] S, Pelley, D, Mesiner, P, Zandevakili., & T, F, Wenisch. (2010). Power Routing: Dynamic Power Provisioning in the Data Center. ACM.

[8] Data Center Efficiency Assessment. (2014). Retrieved from https://www.nrdc.org/sites/default/ files/data-center-efficiency-assessment-IB.pdf.

[9] G, Zhabelova., A, Yavarian., & V, Vyatkin. (2015). Data center energy efficiency and power quality: An alternative approach with solid state transformer. doi: 10.1109/IECON. 2015.7392279