A Centralized Architecture for Energy-Efficient Job Management in Data Centers

George Perreas, Petros Lampsas
Department of Computer Engineering
Technological Educational Institute of Central Greece
Lamia, Greece
e-mails: {georgeperreas@gmail.com, plam@teilam.gr}

***Abstract*—In this work, we present a centralized monitoring entity that attempts to reduce power consumption in Internet Data Centers (IDCs) by employing live Virtual Machine (VM) migrations between blade servers. To perform live VM migrations, usage statistics collected by servers are evaluated and the servers that may be offloaded are selected. VMs that belong to the servers that may be offloaded are scattered to other active servers provided that the user-perceived performance is sustained. Overall, jobs submitted by users should be consolidated to as few servers as possible and the servers that host no job can be put in stand-by or hibernate mode, thus achieving an overall power reduction. Data Center management authorities may take advantage of such a monitoring entity in order to decrease energy consumption attributed to computing, storage and networking elements of data centers.**

Keywords–Data Center Energy Efficiency; Energy efficient Job Management; Virtual Machine Migrations.

I. INTRODUCTION

Data Centers are facilities used to host cloud computing resources comprising computing systems and associated equipment, such as networking, storage, security and environmental control systems. These computing resources can be accessed through Internet. An IDC, usually, deploys hundreds or thousands of blade servers, densely packed to maximize space utilization. It generally includes redundant or backup power supplies, redundant data communications connections, environmental controls (e.g., air conditioning, fire suppression) and security devices. To protect these systems and their vital functions, however, data centers also employ energy-intensive air conditioning systems, fire suppression systems, redundant/backup power supplies, redundant Internet connections, and security systems.

Running services in consolidated servers in IDCs provides customers an alternative to maintaining in-house equipment and personnel that provides services. IDCs achieve economies of scale that amortize the cost of ownership and the cost of system maintenance over a large number of machines. However, with the rapid growth of IDCs in both quantity and scale over the last few years, the energy consumed by IDCs, directly related to the number of hosted servers and their workload, has been skyrocketed [1].

The most commonly used metric to determine the energy efficiency of a data center is Power Usage Effectiveness (PUE). This simple ratio is the total power entering the data center divided by the power used by the information technology equipments. However, according to an Uptime Institute survey [2], only half of the large organizations (over 2000 servers) measure PUE in a detailed fashion, while only 18% of smaller data centers (those with fewer than 500 servers) had any PUE focus. This is an indication that there is a lot of space for optimizations, as far as IDC energy consumption is concerned.

In IDCs, servers, storage and networking systems may get underutilized during daily operation, especially in cases where job population, resource utilization, arrival and completion rates vary significantly over time. This is not a problem from the scheduling algorithms point of view that distribute load by employing as many servers as possible in order to minimize e.g. job completion time. In the general case, despite the efficiency of scheduling and job placement algorithms when new jobs arrive in an IDC, job completion and/or varying resource needs during job lifetime may create the opportunity to consolidate jobs to servers. By consolidating jobs to some selected servers until the rest of them possess no more jobs, these jobless servers may be put in low power consumption mode or even turned off; thus, achieving decreased energy consumption in the DC. Server consolidation is performed up to the point that the servers selected to host the jobs are fully exploited, as far as their computing power is concerned, without violating user perceived performance and Service Level Agreements (SLAs).

Technologies that tackle energy efficiency in IDCs are network power management, chip multiprocessing energy efficiency, power capping and storage power management, to name a few. VM technology can be considered as a software alternative to the approaches that tackle energy efficiency in IDCs. VM technology (such as VMWare [4], Xen [3]) enables multiple OS environments to co-exist on the same physical machine, albeit in strong isolation from each other.

 ISBN: 978-1-61208-338-4

Despite their technical differences, both technologies support migration of virtual machines (i.e., VM transfer across physical computers). There are two types of migration: regular migration and live migration. The former stops a VM, moves a VM from one host to another and then restarts the VM, while the latter transfers the VM without ceasing to offer the service during transition.

VMs may be transferred between physical machines, without user intervention, when certain conditions apply to the physical machine that originates the migration. VM and VM migration technologies exhibit great potential to efficiently manage workload consolidation, and therefore, improve the total IDC energy efficiency.

In this work, we implemented and tested Open Data Center Manager (ODCM), a centralized mechanism that decides VM migrations (and consequently migrations of every job executed in this VM) according to a multi-criteria decision making algorithm and gathered monitoring information concerning computational load incurred in an IDC. VM migrations are decided in such a way that results in server consolidation, i.e. all the jobs submitted to a data center run to as few servers as possible, taking into account Service Level Agreement between data center managers and end users. We conducted an initial evaluation of ODCM in a, relatively small, cluster and initial results depict that ODCM may result in increased energy efficiency through server consolidation by employing live VM migrations.

The rest of the paper is structured as follows. Section II gives an overview of related work. Section III presents the system model and gives a problem formulation. The building blocks of ODCM are also described there. Section IV outlines implementation issues. Finally, Section V concludes the paper.

II. RELATED WORK

Energy efficiency in data centers, as far as computing elements are concerned has been studied in different contexts. Approaches adopted by researchers fall in two broad categories: i) solutions that attempt to minimize power consumption in the hardware elements of the IDC and ii) software solutions that manage IDCs and schedule jobs on servers taking into account not only performance but the minimization of the overall energy consumption.

The first type of solutions can be generally classified under power management approaches. These options include the Dynamic Voltage/Frequency Scaling (DVFS), turning On/Off system components, the Chip-Multiprocessor approach, etc. Orgerie et al. [6] address theoretical and experimental aspects of energy efficiency in large-scale distributed system both in the power management (study of On/Off models) as well as in the virtualization domain.

DVFS is a prominent approach to adjusting the server power states. Horvath et al. [11] have studied how to dynamically adjust server voltages to minimize the total power consumption while at the same time end-to-end delay constraints in a Multi-tier Web Service environment, are met. Barroso and Hölzle [13] studied how to use Chip Multi-Processor (CMP) to achieve energy-proportional designs. Raghavendra et al. [12] suggested coordinating individual approaches in software and hardware power management in order to efficiently manage energy in multiple levels in data center environments.

The second category of solutions involves mainly job assignment to servers as well as VM migrations to achieve energy-efficiency. Liu et al. [5] proposed an architecture that enables comprehensive online-monitoring, live virtual machine migration, and VM placement optimization, in order to reduce power consumption in Data Centres. Wood et al. [7] proposed the CloudNet architecture that builds a pool of geographically distributed data centres through efficient WAN VM migrations. This approach unifies data centre equipment and offers enterprises a seamless and secure application execution environment.

Chaisiri et al. [8] proposed Optimal Virtual Machine Placement algorithm that can be used in renting resources between cloud providers in order to reduce user costs for deploying applications in data centres. Finally, Tarighi et al. [9] adopted and deployed an approach similar to ours in the context of cluster computing which, however, does not aim at decreasing power consumption.

The approach adopted in ODCM falls in the second category. To the best of our knowledge, this is the first attempt to tackle energy efficiency in data centres by a centralized entity that consolidates applications and required data by live migrating VMs between hosts within an IDC.

III. SYSTEM MODEL

In this work, we assume Internet Data Centers that host compute and storage entities. These entities host applications and data associated to the applications. Customers receive a specific Quality of Service (QoS) as far as application execution and/or perceived response times are concerned, according to SLAs signed between the customer and the IDC service provider.

Compute entities that reside in IDCs host VMs, the execution environment for customers' applications. Both data needed for application execution and the application code are stored in the same IDC; however, replicas may be created among IDCs owned by the same service provider. VMs and hosted applications may be migrated among compute entities that reside in the same or different IDCs. We suppose that, in order for an application to run on an IDC, the associated data must reside in the same IDC. We also suppose that, SLA for a submitted job is satisfied if a certain amount of the host's computing power is assigned to the VM that hosts this job.

Applications that arrive in an IDC are placed in the most loaded available server that, after hosting the newcomer application, still operates below a certain, user-defined, threshold and meets the requirements derived from the hosted applications SLAs.

ODCM works in a periodic fashion. After a certain period of time set by the administrator, ODCM is invoked attempting to consolidate applications. The servers that will be attempted to be put in low-power mode or hibernated are selected according to a multi-criteria method, i.e., TOPSIS [10]. When the servers that will act as originators of migrating VMs are selected, a bin packing heuristic is

 ISBN: 978-1-61208-338-4

executed that produces a series of live migrations that lead to the applications being run in as few servers as possible.

The next phase comprises implementation of live migrations in order to achieve the result of bin packing heuristic. This step perhaps involves VM placement rearrangement, i.e., migrations between operating servers not selected as originators for migrations, for space creation. Each operating compute element is checked to see whether he can act as a host to a migrating VM. If there is no host that possesses the required resources to run a VM, then compute elements are checked to see whether they can offload some VMs to other operating compute elements in order to create enough free space for the migrating VMs.

A. Topsis

The TOPSIS method is a technique for order preference by similarity to ideal solution, proposed by Hwang and Yoon [10]. The ideal solution (also called the positive ideal solution) is a solution that maximizes the benefit criteria and minimizes the cost criteria, whereas the negative ideal solution (also called the anti-ideal solution) maximizes the cost criteria and minimizes the benefit criteria. The so-called benefit criteria are those for maximization, while the cost criteria are those for minimization. The best alternative is the one that is closest to the ideal solution and farthest from the negative ideal solution.

The TOPSIS procedure is divided in five steps that are described below.

Step 1. A table with the data that will be used for decision making is constructed and normalized.

$$R_{ij} = \frac{x_{ij}}{\sum x_{ij}^2} \quad for\ i = 1,...,hosts\ and\ \ j = 1,..,criteria \qquad (1)$$

The values x_{ij} are the weighted moving averages of the values reported by each server by the monitoring component (e.g., CPU usage, VM usage).

Step 2. Table R is taken as input from step 1 and is weighted using the matrix with the weights that correspond to the criteria being set.

$$V_{ij} = W_j * R_{ij} \qquad (2)$$

In our case, criteria for classifying overloaded servers (in decreasing order of significance) are CPU usage (%), CPU Speed, Free Cores, Total Cores, Total RAM, Free RAM and Total VMs Executing. For deciding the VMs to migrate from overloaded servers, a different set of criteria is introduced: Virtual CPU Usage (%), Virtual RAM Usage and Virtual Cores (used by a VM). Weights, set after experimenting with several potential values, vary from 1 to 9, depending on the significance of each criterion.

Step 3. Ideal solution is the one that is closest to the ideal solution and farthest from the negative ideal solution. The ideal solution can be calculated as follows (Eq. 3a and 3b):

$$A^w = \left\{ \left\langle \max\left(V_{ij} \middle| i = 1,2,...,m\right) \middle| \ j \in J \right\rangle, \left\langle \min\left(V_{ij} \middle| i = 1,2,...,m\right) \middle| \ j \in J' \right\rangle \right\}$$

$$\equiv \left\{ V_{wj} = 1,2,...,n \right\} \rightarrow A^w = \left\{ V_1^w, V_2^w, ..., V_j^w, ...V_n^w \right\}$$

where w is the worst ideal solution and b the best ideal solution.

The negative ideal solution is:

$$A^b = \left\{ \left\langle \min\left(V_{ij} \middle| i = 1,2,...,m\right) \middle| \ j \in J \right\rangle, \left\langle \max\left(V_{ij} \middle| i = 1,2,...,m\right) \middle| \ j \in J' \right\rangle \right\}$$

$$\equiv \left\{ V_{wj} = 1,2,...,n \right\} \rightarrow A^b = \left\{ V_1^b, V_2^b, ..., V_j^b, ...V_n^b \right\}$$

$J = \left\{ j = 1,2,...,n \middle| j \right\}$, for criteria with positive impact, and

$J' = \left\{ j = 1,2,...,n \middle| j \right\}$, for criteria with negative impact.

Step 4. Euclidean distance between every solution and the ideal and negative ideal solution respectively is calculated as follows:

$$S_i^w = \sqrt{\sum \left(V_j^w - V_{ij}\right)^2}, where\ i = 1,2,...,m \qquad (4a)$$

$$S_i^b = \sqrt{\sum \left(V_j^b - V_{ij}\right)^2}, where\ i = 1,2,...,m \qquad (4b)$$

Step 5. The following amount depicts how close a solution is to the ideal solution (the best choice is the one that is closer to 1):

$$C_i^w = \frac{S_i^b}{S_i^w + S_i^b},\ where\ 0 < C_i^w < 1 \qquad (5)$$

B. Bin Packing

Note that, in order to consolidate VMs in as few servers as possible, we actually need to implement a heuristic for a variation of the bin packing problem. In its general form, the bin packing problem (a combinatorial NP-hard problem [14]) can be stated as follows: objects of different volumes must be packed into a finite number of bins or containers each of volume V in a way that minimizes the number of bins used.

The analogy in our problem setting is to consider bins as servers (each of different VM hosting capacity) and objects as VMs that must be hosted on as few servers as possible. The bin packing heuristic is invoked when ODCM is executed and at least a server exceeds a (tunable) CPU utilization limit. All servers that exceed this CPU utilization limit are sorted by TOPSIS from the most appropriate to the least appropriate to be offloaded. These are servers that will act as originators for live VM migrations. After this step, TOPSIS is run again to produce a list of servers that can receive VMs in decreasing order of suitability to act as receivers. Finally, VMs that must be migrated are sorted by TOPSIS from the one that imposes the more load to the CPU to the one that imposes the lesser load to the CPU. Each server in the receiver list is checked and if it possesses enough free resources (CPU cores, free memory space) to host VMs that will be offloaded by the originator server, VM migration commences.

If the server that is selected to be a candidate receiver cannot host VMs that are to be migrated, an additional check is performed in order to find out if the candidate receiver can free enough resources by migrating VMs to other candidate receivers. This attempt to free resources in one server will (perhaps) trigger a series of recursive migrations originated from the servers that are selected to get checked if they can

 ISBN: 978-1-61208-338-4

receive a migrating VM. If these checks result in freeing enough resources the migration is performed; otherwise ODCM concludes that no migration can be carried out. The cost for each live migration is considered to be negligible, since migrations are performed within an IDC, over a high-speed local area network.

IV. ODCM IMPLEMENTATION

ODCM is implemented as a client-server application using the Java programming language and UDP transport (Fig. 1). Values obtained from the individual servers concerning CPU and VM virtual CPU utilization are stored in a MySQL database after being processed to obtain the weighted moving average. In this way the values stored take into account not only the last value reported by the monitoring component, but all the reported values within a time period. Older values contribute less to the computed value whereas the more recent the obtained value, the greater the contribution to the value computed and stored.

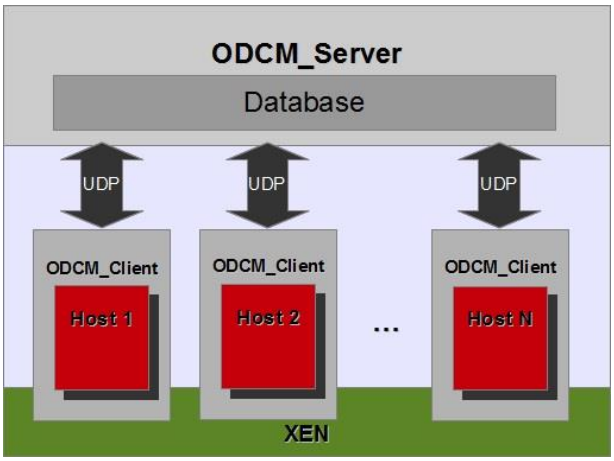

Figure 1. ODCM Architecture

Data management is performed by using the Java Persistence API (JPA). In JPA, there exist persistence entities, i.e., lightweight Java classes, whose states are typically persisted to a table in a relational database.

After the invocation of ODCM, the servers that are loaded above a user defined threshold are selected and the VMs they host are migrated. 20% of these servers (or at least one server) are placed in low power consumption mode in order to be ready to execute new jobs that cannot be hosted to any of the already operating servers. The remaining servers are hibernated, and when one of the servers that are set in low power consumption mode resumes normal operation, one of the hibernated servers is chosen randomly to join the pool of servers in low power consumption mode that are ready to undertake newcomer jobs.

V. CONCLUSION AND FUTURE WORK

ODCM is a periodically executed service that attempts to consolidate applications in as few servers as possible in order to conserve energy. Lab tests to a relatively constrained setting (consisted of 5 servers) revealed that the attempted consolidation (and thus, the resulting power consumption reduction) is achieved and VM live migrations are decided and executed in a timely manner.

ODCM execution could also be event-driven, triggered when specific simple or metrics reach a certain threshold. We plan to evaluate these two approaches, i.e., periodic execution vs. asynchronous event-driven execution and also check which of the metrics are giving best results assuming different workloads.

Since current solutions for VM migration incur service disruption because they slow-down storage I/O operations during migration, we intend to accompany scheduling algorithms with data allocation and replication techniques so that the data required for the computation be as "near" to the computation as possible.

Extensive testing of ODCM using appropriate infrastructure should take place. ODCM will be extended with data consolidation, i.e., migrating data needed for computations to as few storage servers as possible. ODCM could also be extended to minimize energy consumption by migrating tasks and the relevant data among IDCs that belong to the same owner, taking into account time zone job submission statistics. Furthermore, the source of energy provided to the IDC could be taken into account (i.e., it is preferable to migrate jobs to IDCs that are powered using renewable energy sources, as long as user-perceived performance remains within acceptable levels).

ACKNOWLEDGMENT

This work is implemented through the Operational Program "Education and Lifelong Learning" and is co-financed by the European Union (European Social Fund) and Greek national funds.

REFERENCES

[1] Department of Energy (DoE), Data Center Energy Efficiency Program, http://www1.eere.energy.gov/manufacturing/tech_assistance/pdfs/doe_data_centers_presentation.pdf, April 2009 [retrieved: April,2014].

[2] Uptime Institute Survey Results, http://uptimeinstitute.com/2012-survey-results, 2012 [retrieved: April,2014].

[3] Xen User Manual, http://bits.xensource.com/Xen/docs/user.pdf [retrieved: April,2014].

[4] http://www.vmware.com/ [retrieved: April,2014].

[5] L. Liu, H. Wang, X. Liu, X. Jin, W. Bo He, Q. Bo Wang, and Y. Chen, "GreenCloud: a new architecture for green data center," Proc. of the 6th International Conference on Autonomic Computing and Communications industry session (ICAC-INDST 09), 2009, pp. 29-38.

[6] A.-C. Orgerie, L. Lefevre, and J.-P. Gelas, "Demystifying energy consumption in Grids and Clouds," Proc. of the International Conference on Green Computing (GREENCOMP 10), IEEE Computer Society, 2010, pp. 335-342.

[7] T. Wood, K. K. Ramakrishnan, P. Shenoy, and J. van der Merwe, "CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines," Proc. of the 7th ACM SIGPLAN /SIGOPS International Conference on Virtual Execution Environments (VEE 11). ACM, 2011, pp. 121-132.

[8] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal Virtual Machine placement across multiple Cloud providers," Proc. of the Services Computing Conference (APSCC 09), IEEE Asia-Pacific, 2009, pp. 103-110.

 ISBN: 978-1-61208-338-4

[9] M.Tarighi, S.A.Motamedi, and S.Sharifian, "A new model for virtual machine migration in virtualized cluster server based on Fuzzy Decision Making," Journal of Telecommunications, vol. 1, no. 1, Feb 2010, pp. 40-51.

[10] C.L. Hwang and K. Yoon, "Multiple Attribute Decision Making: Methods and Applications," 1981, New York: Springer-Verlag.

[11] T. Horvath, T. Abdelzaher, K. Skadron, and X. Liu, "Dynamic voltage scaling in multitier Web servers with end-to-end delay control," IEEE Trans. on Computers, vol. 56, no. 4, Apr. 2007, pp. 444-458.

[12] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No power struggles: coordinated multi-level power management for the data center," Proc. 13th Intl. Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 08), ACM, 2008, pp 48-59.

[13] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," IEEE Computer, vol. 40, no. 12, Dec. 2007, pp. 33-37.

[14] Bin Packing, http://en.wikipedia.org/wiki/Bin_packing_problem [retrieved: April,2014].

 ISBN: 978-1-61208-338-4