Reducing Energy Consumption for Reconfiguration in Cloud Data Centers

Invited Paper

Omar Chakroun, Soumaya Cherkaoui

INTERLAB Research Laboratory, Université de Sherbrooke, Canada

{omar.chakroun, soumaya.cherkaoui}@usherbrooke.ca

***Abstract*— Mobile Cloud Computing (MCC) leverages mobile devices and infrastructure equipment to increase services accessibility. It uses increased devices computing capability to enhance services usability and ensure high availability. This growth in performances results in an increased interest for platforms use to accommodate a multitude of applications. To support such an increase in demand, new designs for resource management have to be implemented in order to reach usage optimality. In this work, we propose to design new algorithms to optimise MCC resources management techniques based on stochastic networks optimization. Our approach is focused on energy consumption optimization on the cloud data center side while ensuring resources elasticity to adapt to users' demands and insure a highly available platform. We elected an overclocking technique to enhance servers' capabilities and Lyapunov optimisation to ensure design stability and to minimise the energy cost. We perform extensive simulations under different charge conditions in order to prove the design effectiveness in ensuring the service with lower power consumption. Simulations results confirm the effectiveness of the proposed resources management design.**

***Index Terms*—MCC, resource management, energy, elasticity and high availability, Lyapunov optimization.**

I. INTRODUCTION

Mobile cloud computing (MCC) denotes the convergence between two well investigated concepts: a) cloud computing, which provides computational and storage capacity for users applications and b) mobile computing which allows portable devices to access distant resources with ad hoc or infrastructure-based wireless communication technologies. The motivation behind MCC is the need to accommodate resource-intensive applications within mobile devices with comparatively very limited capacity. The ultimate goal is to enable the execution of rich mobile applications on a set of mobiles devices by leveraging distant platforms for almost unrestricted capacity and functionality.

To ensure high service availability, cloud providers deploy huge and costly infrastructures comprising a large number of data centers. Once these infrastructures are deployed, servers' energy consumption constitutes a significant proportion of operating costs [1]. Reducing the energy cost can translate into a huge gain for cloud providers. Statistic studies conducted by cloud providers conclude that cooling and processing energy constitutes up to 50% of the total energy used in a data center [2]. Energy efficiency can be achieved with more efficient hardware and integrated thermal management [3, 4, 5]. However, designs for resource management and network reconfiguration policies can also achieve significant gains in energy savings.

Many researchers tried to reduce energy cost of such platforms by proposing new designs for resource management based on optimisation processes. These optimization processes range from simple optimization-under-constraints [6] to complex algorithms leveraging game theory and stochastic analysis [7]. Cloud resource management has to be both highly efficient and adaptive. Adaptability here means the capability to manage resources efficiently, when facing rapid demand changes in term of resource requests, while supporting heterogeneous applications.

In this work, we use a stochastic approach for resource management in order to optimize energy consumption in cloud data centers while maintaining service availability. We define resource management as the process of allocating computing, storage, and networking capabilities in order to meet both user demand and mobile cloud provider objectives. Resource management in MCC uses virtualization techniques to facilitate resource multiplexing. Virtualization combined with mobility, allows Virtual Machine (VM) migration and /or consolidation. VM Migration can be used to move resources closer to the end user so as to reduce response time. VM consolidation is usually promoted to save on energy consumption. Moving VMs strewn over multiple Physical Machines (PM) toward a smaller set of PM can reduce energy usage.

Our approach is built on the following findings: (1) VM migration consumes an important amount of physical server resources especially on the source PM. The resources needed for VM migration account for 10%-20% of the CPU and memory resources of PMs [8, 9]. Based on that observation, PMs are only run at 90% of their capacity in the most optimistic scenario. The other 10% of the resources are reserved for possible VM migrations. (2) A PM which is started and is not handling any requests, consumes an approximate energy of 45% of its maximum energy consumption at full charge. The above value is called the nominal power consumption. (3) VM consolidation constitutes a good approach to reduce overall data center power consumption, which gives cloud providers the ability to turn off unused PMs.

Based on the previously mentioned remarks, we elected an overclocking technique to accommodate VM migration requests while making an efficient use of servers' resources and reducing the overall cloud data center power consumption. We show that overclocking allows using PMs to a better capacity. We also show that at the small expense of energy cost due to overclocking, we can achieve much more significant savings on overall servers' energy consumption.

The remainder of this paper is organized as follows; Section II presents design principles for our overclocking approach. Section III introduces the system model to ensure resource allocation optimization. Section IV presents the simulation environment and an overview of the results. Finally, Section V concludes the paper.

II. Overclocking technique

Overclocking is the fact of configuring a computer to operate at a faster rate (clock frequency) than the one that was certified for by the manufacturer. The main purpose from overclocking is to gain extra performance from a given component by increasing its operating speed. Overclocking is usually applied on major components such as main processors and graphic controllers. Most components are designed with a safety margin to deal with operating conditions outside the manufacturers' control, and overclocking is the action of setting the device to run in the higher end of that margin with the understanding that temperature and voltage must be controlled as the safety margin is reduced. While most modern devices are fairly tolerant of overclocking, all devices have finite limits - generally for any given voltage most parts will have a maximum "stable" speed where they still operate correctly.

A. Overclocking extent and design considerations

In our design, we make use of overclocking in servers while staying in the safety operating range in order not to decrease the components lifetime, involve the need for extra investment in cooling systems, or increase voltage requests. Thus, we use overclocking up to a certain extent, not to exceed 15% of the processing speed, and without any modification to the processor Vcore voltage or the cooling system. We use overclocking only if a VM migration is needed and on the source PM side only. Upon completing the VM migration, the processor speed is retuned down to the maximum value specified by the manufacturer and overclocking is turned off. Authors in [10] present some results of overclocked processors performances without the need of extra investments on cooling or modifications on the hardware. They conclude that a gain in processor speed up to 26% is achievable without any changes. Of course, running a processor over its maximum speed can make the system consume more power, especially for heat dissipation. However, studies in [11] showed that the extra power consumed when overclocking is activated is relatively low and ranges from 2W - 6W per processor and per 200 MHz overclocking step up to 600 MHz. Over 600 MHz overclocking related power consumption can increase rapidly to reach up to 40W per 200 MHz step. Thus, we are considering a maximum of 15% overclocking in order to limit the power consumption overhead and we are activating the overclocking for short periods of time corresponding to VM live migration durations in order not to degrade the servers' performances and reliability.

B. Reducing the impact of reconfiguration

There exists a multitude of reconfiguration techniques, for better resources management, such as VM resizing and live migration. Modern hypervisors involve reduced overhead which allows resources entitlements to be changed on a running VM. Researchers in [12] studied the impact of VM migration on the cloud performances and particularly on the source server side. VM migration is usually useful for clustered applications and it is for high interest to data centers either to consolidate VM to reduce the number of active servers to save power or to ensure higher resources for resources intensive tasks. The need of extra resources on the source server is related to the need of more active memory usually due to cache contention between co-located VMs. In this perspective, temporarily activating the overclocking when a VM migration is needed can be beneficial in reducing the impact on coexistent VM that share the same physical memory, and can speed up the migration process. We will take such an approach on the source Data Center (DC) server side only and we will measure its impact on power consumption which we denote by the reconfiguration cost. Theoretically, the gain in processor speed translates in a gain on the number of tasks processed and can impact the memory speed. Algorithm 1 depicts the steps for computing resources allocation and the activation of overclocking when VM migration is requested.

C. Power consumption characterization

We focus on servers in our energy model. For servers, we adopt the model from [4] that characterizes the individual server power consumption function of the processing speed as an affine function as in equation Eq.(1) where P_{idle}, P_{peak} and U denotes respectively the power consumption in idle state, power consumption when the server is fully utilized and the utilization level ranging from 0 to 1.

Equation Eq.(2) presents an extension of the equation Eq.(1) when the overclocking is supported. P_{over} denotes the maximum power consumed when overclocking is at its maximum tolerable value and U_{over} denotes the rate of overclocking function of the maximum processor speed.

$$P_{Cons} = P_{idle} + (P_{peak} - P_{idle}) * U \tag{1}$$

$$P_{Cons\_over} = P_{idle} + (P_{peak} - P_{idle}) * U + (P_{over} - P_{peak}) * U_{over} \tag{2}$$

Figure 1 illustrates the power consumption with and without overclocking activation and the gain in power consumption for a 2.8GHz processor. We are taking into consideration 6W extra-power consumption for every 200MHz overclocking step up to 600 MHz and a 45W over consumption above that threshold for every step [8]. In Figure 1, "standard approach" refers to the case where no overlocking is activated and servers are working at a maximum of 90% of their capacity. "Overclocking approach" refers to the overclocking approach where servers are working at a maximum rate of 100% and usage of overclocking is limited to the case where a VM migration is needed.

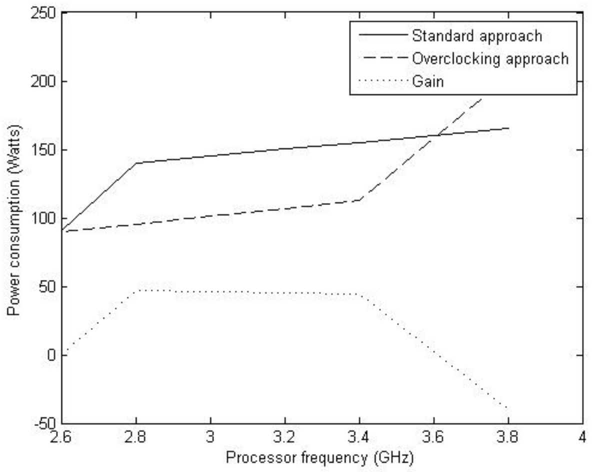

FIGURE I – OVERCLOCKING VS STANDARD APPROACH POWER CONSUMPTION

D. *Power consumption optimization*

Our design of a power-efficient data center is based on two ideas; (1) consolidating the maximum number of VM into the minimum number of PM to ensure the lowest power consumption and (2) make use of 100% of servers resources without pre-reserving resources for possible VM migrations. The resources associated with VM migration in our design will be ensured by activating the overclocking technique on the source server side when needed. Thus, we describe our approach in three scenarios: (a) new request arrival, (b) VM migration request and (c) VM consolidation to reduce the number of active PM.

(a) Assume that a new resource request is received on the admission controller side of our cloud data center. The application profiler will diagnose how much computing resources are needed to accommodate that request and depending on the servers state, will route the request to the optimal server. In case there are not enough resources on the set of active servers handling that application, the resources handler will activate a new server and instantiate the VM on it before routing the request to the newly activated server. Thus power consumption on the new activated server will be $P_{idle}+(P_{peak}-P_{idle})*U$ where U designates the utilization level needed to accommodate the demand. Otherwise, if one of the active servers is able to handle the request, its power consumption will be increased by a factor corresponding to the extra utilization needed $U+uextra$. It is worth noting that in the traditional approach, the maximum utilization does not exceed 90%. In our approach, we use the server resources at a full extent (100%).

(b) In case of a VM migration request, the traditional approach does not introduce any extra usage or resources reservation since VM migration resources account for 10% of the PM resources and these resources are always pre-reserved. On the other hand, in our approach we are making use of 100% of PM resources for request handling and an overclocking activation is needed before proceeding to the VM migration. Overclocking deactivation is needed after VM migration completion.

(c) In the DC, the set of running VMs are strewn over multiple PMs. In our approach, we aim to consolidate the maximum number of VMs on the minimum number of PM to reduce the global DC power consumption. This will help us reduce the number of active PMs which at the end reduces the overall power consumption (see algorithm 1).

Algorithm 1 VM consolidation

```
δs_i: workload of server i
δ_max: maximum workload on a server
TSU: table containing all active servers workloads ordered
increasingly except servers at δ_max
i,j: indexes
Begin:
While (j<=sizeof(TSU))
   For i=j+1 to sizeof (TSU)
      if ($\delta s_j + \delta s_i <= \delta_{max}$)
         activate overclocking on $S_i$
         allocate resources on $S_j$
         update $\delta s_j = \delta s_j + \delta s_i$
         migrate VM($S_i$) to $S_j$
         deactivate overclocking on $S_i$
         shutdown $S_i$
         remove $S_i$ from TSU and Shift table elements to the
left
         Sizeof(TSU)—
      Else
         j++
      End if
   End for
End while
```

III. PROBLEM FORMULATION AND THEORETICAL ANALYSIS

A. *Problem formulation*

We are considering a data center with S servers that hosts a set of N applications denoted by A. each server j hosts a subset of applications. It uses one VM per applications in order to ensure applications severability and an application can have multiple instances running across the data center. We define indicator function e_{ij} as equal 1 if application i is hosted on server j, 0 otherwise. We assume a time slotted system where at every timeslot new requests arrive for application i with a rate λ_i according to a random process which is independent from the amount of unfinished work and we suppose that we have no knowledge of the statistics of these arrivals.

Let $W_i(t)$ denote the router buffer containing all admitted requests after the admission controller. $R_{ij}(t)$ the requests for application i that are routed to server j in slot t and $R_i(t)$ the newly admitted requests. Thus the router dynamic can be characterized by (3).

$$W_i(t+1) = W_i(t) - \sum_j R_{ij}(t) + R_i(t) \tag{3}$$

Let $S_i(t)$ denote the set of active servers capable of handling the application i at slot t thus the routing decision must satisfy the following constraints (4) and (5) at every slot.

$$R_{ij}(t) = 0 \text{ if } e_{ij} = 0 \tag{4}$$

$$0 \le \sum_{j \in S(t)} e_{ij} R_{ij}(t) \le W_i(t) \tag{5}$$

Let us denote the set of control action available at the server j level at the instant t under any control policy by $I_j(t)$ and let $P_j(t)$ the corresponding power consumption. Then the queuing dynamics of the requests of the application i on server j follows (6) where $\mu_{ij}(I_j(t))$ denotes the service rate for the application i at server j under the control decision $I_j(t)$.

$$U_{ij}(t+1) = \max\left[U_{ij}(t) - \mu_{ij}(I_j(t)), 0\right] + R_{ij}(t) \tag{6}$$

We assume that the expected value of the service rate can be known since, as discussed in section II, we can derive the experienced power consumption based on the processor frequency assignment. Thus at every slot t, the following decisions have to be made: (1) Routing decision for the admitted requests $R_{ij}(t)$. (2) Resource allocation decision $I_j(t)$ including resources distribution among VMs and activating of the overclocking.

Let us denote the average expected rate of admitted requests for the application i under control policy η by r_i^n and the average expected power under the same conditions by p_j^n which expressions are as follows.

$$r_i^{\eta} = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\left\{R_i^{\eta}(\tau)\right\} \tag{7}$$

$$p_j^{\eta} = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E\left\{P_j^{\eta}(\tau)\right\} \tag{8}$$

Thus considering a collection of non-negative weights α_i, β, our objective is to design a control policy η that solves the following stochastic optimization problem.

$$\text{Maximize: } \sum_{i \in A} \alpha_i r_i^{\eta} - \beta \sum_{j \in S} p_j^{\eta}$$
$$\text{S.T: } \quad 0 \le r_i^{\eta} \le \lambda_i \tag{9}$$

Thus the objective problem is a general weighted linear combination of the sum throughput of the applications and the sum of the average power consumption in the data center.

B. Control Algorithm based on Lyapunov optimization

We use Lyapunov optimization in order to achieve stability of the system.

Let $V>=0$ be a control parameter that has to be chosen by the system administrator to ensure the desired tradeoff between the performance and the power consumption. Let $W_i(t)$ and $U_{ij}(t)$ be the router and the server queue backlog at timeslot t. As the backlog values evolve over time as described in (3) and (6), the algorithm adapts to the system changes and solves the problem in (9) leveraging a sequence of optimization problems over time on three distinct steps.

Request routing: let j' denote the server that is having the smallest queue backlog and belonging to the set of servers that are able to process the requests for application i. thus, a routing policy can redirect all requests for application i to such a server under the condition that $W_i(t) > U_{ij'}(t)$.

Resources allocation: at each server j, choose the resource allocation $I_j(t)$ that solves the Lyapunov optimization process where p_{max} denotes the maximum power consumption per server.

$$\text{Maximize: } \sum_{i \in A} U_{ij}(t) E\left\{\mu_{ij}(I_j(t))\right\} - Vp_j(t)$$
$$\text{ST: } \quad p_j(t) \le p_{\max} \tag{10}$$

IV. Result overview

We chose to implement our approach under Matlab since it offers good computational capability and offers a multitude of optimization frameworks. We implemented three main approaches: (1) the nominal approach which uses servers capabilities up to 100%, (2) the standard approach which uses servers up to 90% of their capacities and leaves 10% of it for VM migration purposes and (3) our overclocking approach which uses servers up to 100% for processing and overclocks servers CPU in case of VM migration up to 15%.

A. Simulation environment

We simulated a data center constituted of 100 servers using homogeneous processors that have a minimum speed of 1.4GHz and a maximum speed of 2.8GHz while power consumption ranges from 45 watts to 95 watts when server is activated. The number of request arriving to the DC is randomized and ranges between 0 and 100 requests per server per timeslot. Each request is characterised by the needed processing power expressed as a ratio from the server maximum processing capacity. Servers processing capabilities range from 5 to 10 requests per server per timeslot. The simulation duration is 4000 TS. We fixed the control parameter V to a value of 1 to ensure a fair trade-off between power consumption and performance in term of processed requests. Table I below summarizes the main simulation parameters

TABLE I – GLOBAL SIMULATION PARAMETERS

Parameter	**Value**
Number of servers	100
Number of active servers at the start	10
Number of request arriving to the DC	0-100 per TS
Servers processing frequency	1.4 – 2.8 GHz
Processing capabilities	5-10 requests per server per TS
Servers power range	45 - 95 watts
Overclocking maximum rate	15% (up to 3.2GHz)
Simulation duration	4000 TS

B. Results overview and analysis

We are mainly interested in assessing the gain in DC power consumption ensured by our designed overclocking approach while subjecting the DC to different requests admission rates. We also make use of a VM consolidation approach in order to reduce the number of active PM and consequently ensure the same processing rate while consuming less power. Figure II shows the number of admitted requests accepted by the admission controller depending on the servers processing charge over the whole simulation duration. The number of admitted requests per server per TS range from 7 to 27 with a mean value of 20 requests per server per TS. Figure III shows the mean servers charge over time. It shows that based on our model, we make use of servers capabilities to the maximum extent trying to use the power more efficiently. The mean charge per server is around 92% of the maximum capacities and most of the time servers are working at rate higher than 90%. This is reflected by the left hand side of the formula in Eq. (10), we are trying to maximize the portion $U_{ij}(t)E\{\mu_{ij}(I_j(t))\}$ which corresponds to the server processing charge where $U_{ij}(t)$ corresponds to the number of request routed to server j and $\mu_{ij}(I_j(t))$ is the server j service rate under the control decision $I_j(t)$. Also the resulted charge shows the effectiveness of proposed VM consolidation approach to reduce the number of active servers and consequently the total DC power consumption.

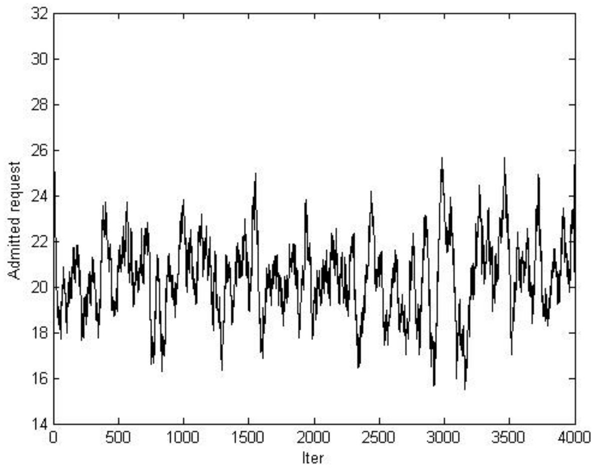

FIGURE II – ADMITTED REQUEST TO THE DATA CENTER OVER TIME

The second step of the simulation is to assess the performances of our approach compared to a nominal approach where servers are used to a full extent but without VM migration support and compared to the standard approach which uses up to 90% of the servers' capabilities and leaves 10% for VM migration purposes. Our approach, on the other hand, uses the servers' capabilities to the maximum extent and activates overclocking to a maximum of 15% when a VM migration is needed. Figure IV shows the performances evaluation between the three approaches in term of active servers to accommodate the admitted requests. We notice that the nominal approach needs to activate approximatively 43 servers per timeslot to accommodate the admitted requests shown in Figure II. The standard approach, uses approximatively 37 servers per timeslot to accommodate the same requests and finally the overclocking technique uses only 33 servers per timeslot to accommodate the same requests. The latter approach ensures a gain in the number of activated servers compared to the nominal approach of 23% and 13.5% compared to the standard approach.

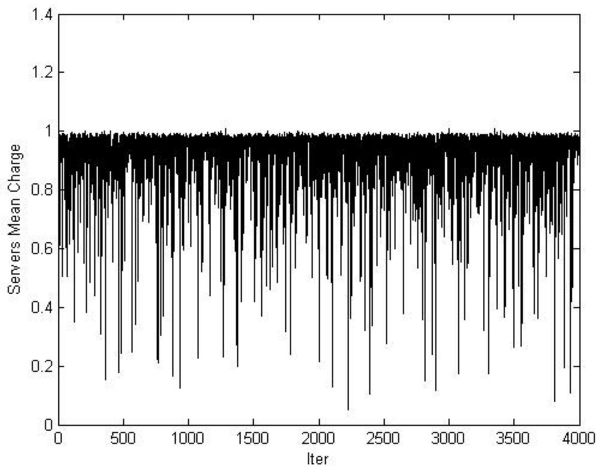

FIGURE III – MEAN SERVERS CHARGE RATE OVER TIME

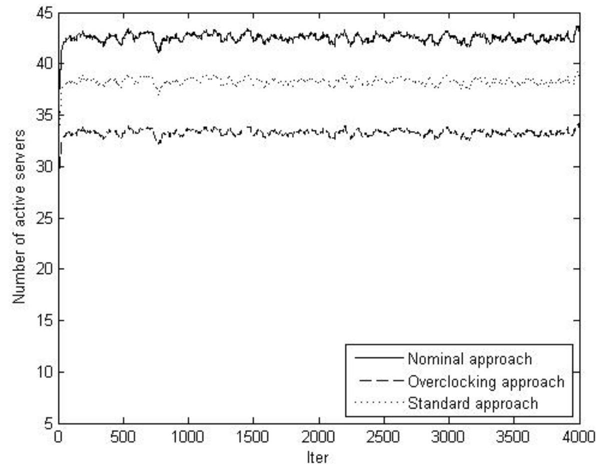

FIGURE IV – NUMBER OF ACTIVE SERVERS FOR OVERCLOCKING, NOMINAL AND STANDARD APPROACHES (SERVERS PER TS)

Since the combination of VM consolidation and overclocking ensures a gain in term of active servers compared to both other approaches,that gain should translate in gain in power consumption. Figure V shows the mean DC power consumption over time. It shows a comparison for power consumption for the simulated DC between the standard approach and the overclocking approach. The overclocking approach consumes a mean 3638 Watts per timeslot while the standard approach uses approximatively 4042 Watts per timeslot. Thus, the overclocking technique ensures a gain in power consumption of 10% in every timeslot which shows the effectiveness of the proposed scheme.

Another important remark is, since we are using Lyapunov optimization, the system behavior and performances are stabilized, which is reflected in Figures IV and V. In Figure V, the standard deviation from the mean number of servers using the nominal approach is 2.25%. Where, it is respectively 2.27% and 2.35% for the overclocking and the standard approach. In Figure V, the standard deviation for the power consumption on the DC for the standard approach is 92.18 Watts per timeslot while it is 82.96 Watts per timeslot for the overclocking approach which corresponds to 2.28% of the mean power consumption for both approaches.

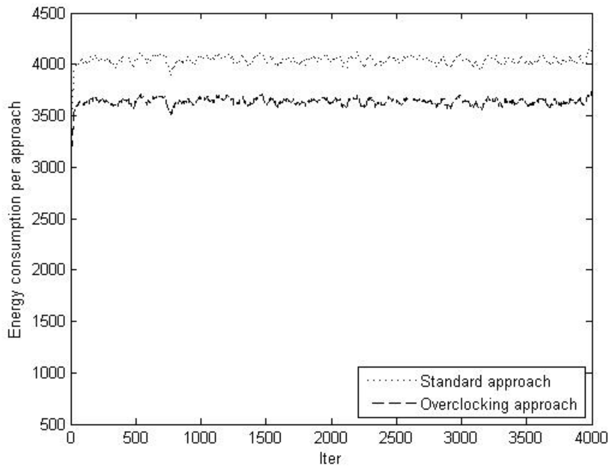

FIGURE V – DATA CENTER ENERGY CONSUMPTION FOR OVERCLOCKING AND STANDARD APPROACHES (WATTS PER TS)

Figure VI shows a comparison between the gain in term of energy and servers usage ensured by our overclocking approach (A) compared to the ADMM approach in [2] denoted by (B) and IMAPP proposed by Chen et al in [4] denoted by (C). On one hand, we notice that the overclocking approach presents relatively low gain in term of energy compared to the other two approaches (10% compared to 35% and 20% for the ADMM and IMAPP respectively), but on the other hand, it ensures a high level of servers utilization up to 90% compared to 80% and not a significant gain for ADMM and IMAPP respectively.

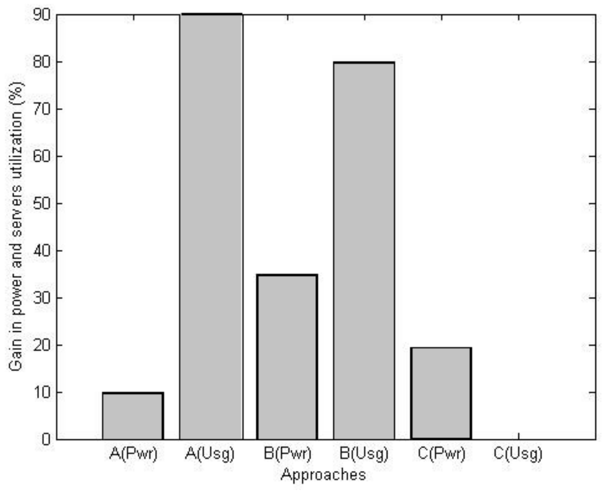

FIGURE VI – APPROACHES COMPARISON IN TERM OF POWER GAIN AND SERVERS USAGE

From the computational complexity point of view, our proposed approach ensure a complexity of O(M*N) if we are considering M servers deployed on the data center and N request per TS admitted to the system for processing. Since on every TS, the table containing all servers' states is sorted in term of computing usage, this extra computation will cost up to O(M) for the worst case. Which in total results in a worst case complexity of O(M*N)+O(M). Since the number of VMs that can be hosted on a single host is limited by the physical resources, therefore, the combined complexity is linear on the number of total hosts.

V. CONCLUSION AND FUTURE WORK

In this paper, we present an analytical and simulation based study on the impact of an overclocking technique on reducing data centers power consumption. We presented a new scheme for VM consolidation to reduce the number of active servers on the DC. The results of the study present the impact of the combination of the VM consolidation and overclocking technique on the DC performance in term of active servers and by consequence on the total power consumption. Simulations show that, even at high servers charge and high requests volume, our design allows stabilizing system behaviour while ensuring a gain up to 23% in the number of active servers and of 10% in total energy consumption.

REFERENCES

[1] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," in Proc. ACM SIGCOMM 2012 Conf. Appl., Technol., Archit, Protocols Comput. Commun. , 2012, pp. 211–222.

[2] H. Xu, C. Feng, and B. Li, " Temperature aware workload management in geo-distributed datacenters," *SIGMETRICS Perform. Eval. Rev.* 41, 1 (June 2013), 373-374. DOI=http://dx.doi.org/10.1145/2494232.2465539

[3] C. Bash and G.Forman, "Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center," In *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference* (ATC'07), Jeff Chase and Srinivasan Seshan (Eds.). USENIX Association, Berkeley, CA, USA, , Article 29 , 6 pages.

[4] Y. Chen, D. Gmach, C. Hyser,W. Zhikui, C. Bash, C. Hoover,S. Singhal, "Integrated management of application performance, power and cooling in data centers," in *Network Operations and Management Symposium (NOMS), 2010 IEEE* , vol., no., pp.615-622, 19-23 April 2010 doi: 10.1109/NOMS.2010.5488433.

[5] X. Fan, W.D. Weber, L. A. Barroso, "Power provisioning for a warehouse-sized computer ," *SIGARCH Comput. Archit. News* 35, 2 (June 2007), 13-23. DOI=http://dx.doi.org/10.1145/1273440.1250665

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternat- ing direction method of multipliers," Found. Trends Mach. Learn. , vol. 3, no. 1, pp. 1–122, 2010.

[7] X. Xu and H.Yu, "A Game Theory Approach to Fair and Efficient Resource Allocation in Cloud Computing," Mathematical Problems in Engineering, vol. 2014, Article ID 915878, 14 pages, 2014. doi:10.1155/2014/915878.

[8] I. Takouna, R. Rojas-Cessa, K. Sachs,C. Meinel, "Communication-Aware and Energy-Efficient Scheduling for Parallel Applications in Virtualized Data Centers," in Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on , vol., no., pp.251-255, 9-12 Dec. 2013 doi: 10.1109/UCC.2013.50

[9] Susmit Bagchi, Emerging Research in Cloud Distributed Computing Systems, 2015.

[10] http://www.choixpc.com/overcloc.htm

[11] http://www.xbitlabs.com/articles/cpu/display/power-consumption-overclocking_10.html#sect0

[12] A. Verma, G. Kumar,R. Koller, A. Sen, "CosMig: Modeling the Impact of Reconfiguration in a Cloud," in Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on , vol., no., pp.3-11, 25-27 July 2011, doi: 10.1109/MASCOTS.2011.37.