2012 IEEE Fifth International Conference on Cloud Computing

Preemption-aware Energy Management in Virtualized DataCenters

Mohsen Amini Salehi*, P. Radha Krishna†, Krishnamurty Sai Deepak† and Rajkumar Buyya*
*Cloud Computing and Distributed Systems (CLOUDS) Laboratory
Department of Computing and Information Systems,The University of Melbourne, Australia
Email: {mohsena,raj}@csse.unimelb.edu.au
† Infosys Research Labs
Email: {radhakrishna_p,KrishnamurtySai_D }@infosys.com

***Abstract*—Energy efficiency is one of the main challenges that datacenters are facing nowadays. A considerable portion of the consumed energy in these environments is wasted because of idling resources. To avoid wastage, offering services with variety of SLAs (with different prices and priorities) is a common practice. The question we investigate in this research is how the energy consumption of a datacenter that offers various SLAs can be reduced. To answer this question we propose an adaptive energy management policy that employs virtual machine (VM) preemption to adjust the energy consumption based on user performance requirements. We have implemented our proposed energy management policy in Haizea as a real scheduling platform for virtualized datacenters. Experimental results reveal 18% energy conservation (up to 4000 kWh in 30 days) comparing with other baseline policies without any major increase in SLA violation.**

***Keywords*-Energy efficiency; Virtual Machine (VM); datacenter; preemption; SLA;**

I. Introduction

Datacenters normally encounter different usage scenarios from users. For instance, running a scientific simulation, which is in the form of a batch job without a specific deadline; or hosting a corporate web site for a long period of time, which requires a guaranteed availability and low latency. In response to such diverse demands, many of the current datacenters provide services with different Service Level Agreements (SLA) that imply different priorities.

For example, Amazon EC2[1] supports reserved (availability guaranteed) and spot (best-effort) virtual machine (VM) instances. Offering a combination of advance-reservation (AR) and best-effort (BE) schemes [1], interactive and batch [2] jobs, tight-deadline and loose-deadline jobs [3] are also common practices in datacenters. In these environments, when there is a shortage of resources, commonly, lower priority requests are preempted in favour of a higher priority one [4], [5], [6], [1], [2].

Haizea [7] is a scheduler for virtualized datacenters that supports combination of advance-reservation (also termed AR) and best-effort (also termed BE) requests. In the former, the resource must be available at the requested time; whereas in the latter, requests are served as soon as possible and they are placed in a queue if necessary. In Haizea, BE requests are preempted when there is insufficient resources for a newly arriving AR request. However, preemption can potentially lead to starvation of BE requests [8]. To prevent starvation, Haizea scheduler considers a limited and predictable waiting time for BE requests (the low priority requests).

Another trend that is prominent for datacenters is the growing awareness about energy consumption within the datacenters. Thus, people from industry as well as academia are seeking for energy efficient solutions within the datacenters that are also aware of user required performance and SLA.

Efficient resource management policies, in particular, can significantly reduce the energy consumption of datacenters [9]. In many of the current datacenters Virtual Machine (VM) technology is being used by the resource management system as a unit of resource provisioning [10]. VMs offer flexibilities, such as preemption (suspend and resume), migration, and consolidation, to the resource management systems in order to implement energy efficient policies.

Taking into account the importance of supporting different SLA levels and energy efficiency in datacenters, in this research, we investigate how the energy consumption within such a datacenter can be reduced. We consider circumstances that the datacenter provides AR and BE schemes where the BE requests should not suffer from starvation. More specifically, the research question that we address is: How the energy consumption of a datacenter that supports AR and BE requests can be reduced while the BE requests do not suffer from starvation?

We answer this question by considering preemption of the lower priority (e.g. BE) requests as an approach for saving energy. In fact, we propose an energy management policy that determines whether a new arriving high priority (e.g. AR) request should be served via preempting other requests, or through reactivating switched off resources. Additionally, the policy applies VM consolidation to save energy in circumstances that do not lead to starvation for BE requests. The proposed energy management policy employs fuzzy logic in order to derive the appropriate decision.

We implement our policy in the context of Haizea [1]. For that purpose, we first add the power-awareness capability

[1] http://aws.amazon.com/ec2

978-0-7695-4755-8/12 $26.00 © 2012 IEEE
DOI 10.1109/CLOUD.2012.147

to the Haizea; then, we incorporate our proposed energy management policy into that.

In summary, contributions of this research are threefold:

- Proposing an adaptive energy management policy within a datacenter via preemption-awareness.
- Extending the Haizea scheduler with energy-awareness capabilities.
- Incorporating and evaluating the proposed energy management policy in the extended Haizea.

Extensive experiments under realistic conditions indicate that the proposed policy significantly saves energy without any major starvation for lower priority requests.

The rest of this paper is organized as follows: In Section II related research work is introduced. In Section III, three contributions of this research are explained. Performance evaluation of the proposed policy is reported in Section IV. Finally, conclusion and future works are provided in Section V.

II. Related Work

Over the last few years, energy efficient resource management has extensively been studied. Many of these studies have employed VM consolidation for energy conservation. Another well studied approach is using dynamic voltage/frequency scaling (DVFS). In this section, we review related research works in these areas and position our work in comparison with them.

Berral et al. [11] investigated a supervised machine learning approach for workload consolidation without SLA violation within a datacenter. They applied machine learning to predict the energy efficiency as well as performance of a job on a set of resources. By contrast, we investigate situation with multiple SLA levels where one level has preemptive priority over the other.

In another work based on VM consolidation [12], the authors applied limited lookahead control in order to maximize the datacenter profit via energy consumption minimization and SLA violations. The controller decides the number of physical and virtual machines to be allocated for each VM. Although we aim at the same objective in this research, we consider how preemption can affect energy consumption where requests have preemptive priority.

Petrucci et al. [13] presented an adaptive energy management policy based on DVFS that considers both user's required performance and energy consumption within a platform with multiple application services. On the contrary, we consider decisions such as switching on/off resources and preemption to save energy.

pMapper [14], provides an energy-aware application placement platform in heterogeneous datacenters that minimizes energy and migration costs while meeting performance guarantees. Our work differs from pMapper in the sense that we focus on the impact of suspend and resuming (preempting) VMs as the energy conservation tool, whereas pMapper considers the impact of live migration of VMs.

Kephart et al. [15] proposed an agent-like approach for controlling both response time and power consumption within a cluster. They apply a coordinator between two independent modules, one for optimizing response time and the other for energy consumption. The coordinator uses reinforcement learning to learn models of dependency between power consumption and response time. By contrast, we focus on the circumstances that requests are in different levels of SLA. Additionally, we consider VM-based provisioning whereas Kephart et al. [15] consider web-based requests.

III. Energy Management Scheme

In this section, we describe the proposed energy management policy in addition to the modifications in Haizea to make it energy-aware. We also describe how the proposed policy was implemented in Haizea.

A. Preemption-aware Energy Management Policy (PEMP)

The overall objective of the energy management policy is reducing energy consumption, while satisfying the users' performance demand within a datacenter.

As mentioned earlier, requests in the system are of two distinct levels of priority, namely, advance-reservation (AR) and best-effort (BE). In the former, which has the preemptive priority, the request has to be served within the requested time. In the latter, however, requests should not suffer from starvation. Nonetheless, preempting BE requests in favour of AR requests and scheduling them in a later time slot leads to increase in waiting time and can potentially end up with indefinite waiting time for BE requests (i.e. starvation) [5].

Therefore, to avoid starvation, we consider a situation where BE requests also have a predictable waiting time. For that purpose, the system administrator defines a *maximum average waiting time* (termed MAWT) for BE requests. Here, we assume MAWT to be α (also called *waiting factor*) times longer than the average duration of BE requests (i.e. $MAWT = \alpha \cdot |duration|$). For example, the administrator can choose the MAWT to be 5 times of the average duration of BE requests.

In this situation, resource acquisition for an arriving AR request should be performed in a way that the average waiting time of BE requests remain smaller than the MAWT and also minimum possible energy would be consumed. Resource acquisition can be carried out either via preemption of resources that run BE requests, or switching on more resources. Additionally, the policy can decide to perform VM consolidation to save energy.

Preemption is applied when the risk of violating MAWT for BE requests is low. By contrast, when the violation risk is high, the policy should switch on resources and offload requests to them. Finally, when energy consumption is high

and the average waiting time of BE requests is low, the policy should apply VM consolidation to save energy.

In fact, the risk of violating MAWT and energy consumption are decisive variables. These variables can be expressed using linguistic variables such as low, medium, and high. Considering the fuzzy logic power in modelling linguistic variables in a system [16], we employ that to model the variables and infer the proper decision.

The proposed fuzzy engine inputs should describe the violation risk and energy consumption. The output of the fuzzy engine is a value that drives the decision of how to allocate resources for an arriving AR request. The output broadly can be switching on resources, preemption, consolidation, or a combination of these actions. We define violation risk of MAWT as follows:

$$V = \frac{\tau}{\alpha \cdot E} \tag{1}$$

where α is the waiting factor, and E and τ are the average duration and average waiting time of BE requests, respectively. E is calculated based on Equation 2.

$$E = \frac{\sum_{i=1}^{N} n_i \cdot d_i}{\sum_{i=1}^{N} n_i} \tag{2}$$

where N is the number of BE requests waiting in the queue, n_i is the number of resources, and d_i is the duration required by BE request i.

Also, τ in Equation 1 is defined based on Equation 3.

$$\tau = \frac{\sum_{i=1}^{N} n_i \cdot w_i}{\sum_{i=1}^{N} n_i} \tag{3}$$

where w_i is the waiting time of BE request i. Values more than one for violation risk ($V > 1$) shows that BE requests are waiting for more than the MAWT.

The second input of the fuzzy engine helps in deciding about preemption or switching on/off resources. Therefore, we consider the utilization of the currently switched on resources (C) as the second input. C is defined according to the Equation 4:

$$C = \frac{L}{P \cdot T} \tag{4}$$

where P is the number of switched on resources; T is the latest completion time of the current requests, and L is the total load which is calculated based on Equation 5.

$$L = \sum_{i=1}^{N} d_i \cdot n_i \tag{5}$$

Values of C vary between $[0, 1]$. Lower values for C shows that the switched on resources are under-utilized. By contrast, values near to 1 indicate that high utilization of the currently switched on resources.

Based on the description, the fuzzy reasoning engine can be expressed as follows:

$$\begin{aligned} &V \times C \rightarrow D \\ &C = \{VL, L, M, H, VH\} \\ &V = \{VLR, LR, HR, VHR\} \\ &D = \{NP, QP, HP, 3QP, AP, LC, MC, HC\} \end{aligned} \tag{6}$$

where VL, L, M, H, VH indicate very low, low, medium, high, and very high fuzzy sets for C. VLR, LR, HR, VHR stands for very low risk, low risk, high risk, and very high risk, respectively, for V. D shows the output fuzzy sets of the fuzzy reasoning engine which can range from NP, which means no preemption and resources should be switched on, QP, which means that quartile of requested resources should be allocated through preemption and the rest has to be allocated via switching on resources. Similarly, HP and $3QP$ stand for half preemption and 3 quartile preemption. AP indicates that all resources should be allocated through preemption which implies that no additional resources should get switched on. Finally, LC, MC, and HC indicate low, medium, and high consolidation of VMs, which help in determining the number of resources that can get switched off.

Since there are 2 inputs with 4 and 5 fuzzy sets, the fuzzy rule-base has 20 rules. For instance, one rule in the fuzzy rule base is as follows:

$$\textbf{if } V \textbf{ is } M \textbf{ and } C \textbf{ is } H \textbf{ then } D \textbf{ is } HP \tag{7}$$

which means that if V is $medium$ and C is $high$, then D is HP. This means that half of the requested resources have to be allocated via preemption and the other half through switching on resources. The fuzzy rule-base is formed based on our expectation from the system behaviour. Then, these rules were fine-tuned through extensive experiments and evaluating the outcomes in different conditions. The entire rule-base is accessible in our web site[2] for interested readers.

The functionality of the fuzzy engine can be expressed via the relation in Equation 8:

$$f(x) = \frac{\sum_{r=1}^{R} \bar{y}^r \cdot \mu_C^r(x_1) \cdot \mu_V^r(x_2)}{\sum_{r=1}^{R} \mu_C^r(x_1) \cdot \mu_V^r(x_2)} \tag{8}$$

where r indicates a fuzzy rule and R is the total number of rules in the rule base (i.e. $R = 20$); x_1 and x_2 are the current

[2]http://ww2.cs.mu.oz.au/∼mohsena

values of C and V respectively, that are input values for the fuzzy engine. $\mu_C^r(x_1)$ and $\mu_V^r(x_2)$ show the membership value of the x_1 and x_2 in the membership function of r^{th} rule. Finally, $\bar{y}^r$ expresses the center of fuzzy membership function fired by rth rule from the output fuzzy set. $f(x)$ covers values more than -1.

We used triangular membership function for all inputs and output variables. Also, we implemented the fuzzy engine using a singleton fuzzifier, product inference engine, and center of gravity defuzzifier [16]. It is worth mentioning that the proposed policy is not a scheduling policy. Indeed, it is the "energy management" component of the resource management system, which works closely with the scheduler but it is not the scheduler. The proposed policy determines how resources should be allocated for a new AR request. Then, a scheduling policy, e.g. backfilling, handles the scheduling of requests on the existing resources.

B. Energy-awareness in Haizea

Haizea [1] is an open source platform that can be used as the scheduling backend of a virtual infrastructure manager, such as OpenNebula [17], within a datacenter.

Haizea is a lease-base scheduler. A lease is an agreement between resource provider and resource consumer whereby the provider agrees to allocate resources to the consumer according to the lease terms presented by the consumer [1]. In Haizea, the lease terms include the hardware, software, and availability period for that hardware and software. Haizea uses VMs to implement leases. Haizea supports AR and BE leases where AR leases have preemptive priority over the BE leases. Thus, in situation that there is not enough resources available for AR leases, BE leases have to be preempted in favour of AR leases. Haizea takes into account all the overheads of suspending and resuming the VMs and schedules them.

Haizea, by default, assumes that all resources are switched on and are ready to be utilized. To add energy-awareness to the Haizea, this assumption has to be relaxed. In fact, in the energy-aware Haizea, the assumption is that resources are switched off initially. Then, as the time passes and the demand increases, the resources are switched on. Accordingly, when there is not any scheduled request on a resource, the resource is switched off. Adding these capabilities entails significant modification in the architecture of Haizea.

As a result of these modification, Haizea lease scheduler is equipped with new functionalities that allow:

- Switching on resources in an on-demand manner. Here, on-demand refers to situation that the number of switched on resources is not adequate to serve AR requests. In our energy management policy on-demand switching on can also be carried out when there is a risk of starvation.
- Switching off the resources when they are not required. This occurs when there is not any scheduled request on a resource.
- VM consolidation which takes place when some resources are under-utilized. In this circumstances, by rescheduling and re-allocating VMs from some resources, they become idle and, therefore, switched off. In the implementation, we apply VM consolidation on resources that have smallest number of leases scheduled on them.
- The scheduler also modified in a way that just considers switched on resources at each moment. In other words, the scheduler enabled to dynamically add and remove resources from the scheduling.

Apart from the mentioned modifications, there are plenty of minor changes in the new structure. We uploaded the energy-aware version of Haizea to our web site. Interested readers should be able to understand the modifications clearly by downloading and reviewing the code and documentations. We did the implementation in a pluggable way that enables other researchers to develop their own energy management policies.

C. Incorporating the Preemption-aware Energy Management Policy into Haizea

After implementing the basic functionalities for energy-awareness in Haizea, the policy proposed in subsection III-A can be implemented and incorporated into the Haizea. The pseudo code of the implemented policy is illustrated in Algorithm 1.

The algorithm is run for each arriving AR request and decides how the resources should be allocated to the request (i.e. through switching on resources, preemption, or consolidation). Additionally, it is run periodically to avoid resource starvation or resource wastage in the datacenter. The algorithm takes the average waiting time (τ), average duration (E), and waiting factor (α) of BE leases, the request to be scheduled (req) as inputs. The result of the algorithm is the proper action that should be taken.

As we can see in the beginning of the algorithm, V and C are calculated based on Equations 1 and 4 respectively. In line 3, the fuzzy reasoning is invoked based on the values of C and V. Then, from line 4 up to the end of the algorithm, the appropriate action is performed based on the output of the fuzzy engine (f, where $f \geq -1$).

The way resources are allocated to an arriving AR request is determined based on the value of f. $0 < f < 1$ shows the situation where resources should be provided via switching on resources. As f approaches 1, fewer resources should get switched on (line 8) and more resources should be allocated via preemption. Specifically, $f = 1$ does not lead to switching on any resource and all resources should be allocated by preemption. In contrast, $f > 1$ shows the situation that the violation risk is low in a way that VM consolidation can be carried out.

Algorithm 1 Preemption-aware Energy Management Policy (PEMP).

Require: α,τ,P,E,L,req

1. $V \leftarrow \tau/(\alpha \cdot E)$
2. $C \leftarrow L/(P \cdot T)$
3. $f \leftarrow FuzzyReasoning(V, C)$
4. **if** $f \leq 1$ **then**
5. **if** $f < 0$ **then**
6. $Num \leftarrow getNumNodes(req)$
7. **else if** $f > 0$ **and** $f < 1$ **then**
8. $Num \leftarrow getNumNodes(req) * (1 - f)$
9. **end if**
10. $SwitchOnNodes(Num)$
11. **else**
12. /*Consolidation*/
13. $Num \leftarrow getNumNodes(req) * (f - 1)$
14. $minNode \leftarrow Required(req.strtTime)$
15. **if** $minNode \leq (NumSwitchOn - Num)$ **then**
16. $Consolidate(Num)$
17. **end if**
18. **end if**

For consolidation (line 12 onwards), after deciding how many of the resources can be consolidated (line 13), we must know if switching off that many resources affect currently scheduled AR leases or not. Therefore, we calculate the minimum number of resources required at that time (line 14). If switching off resources do not affect AR leases (line 15), then the consolidation is carried out (line 16). For consolidation, we use a greedy approach and consider the resources that have minimum leases scheduled on them.

IV. Performance Evaluation

In this section, we discuss different performance metrics considered, and then the scenario in which the experiments were carried out. Finally, experimental results are discussed.

A. Experimental Setup

To have a realistic evaluation, the experiments are carried out based on real traces from the Blue Horizon cluster [18] in San-Diego Supercomputer Center (SDSC). Therefore, we consider a datacenter with 512 single-CPU nodes, each having 1GB of memory, 1Gbps bandwidth between them. Aggressive backfilling [19] is used as the scheduling policy in the datacenter. We also assume that each node can run one VM. However, our proposed policy encompasses multi-core systems where multiple VMs can exist in the same node.

1) Baseline Policies: We evaluate the proposed policy against 2 other policies which are used as a benchmarks. Details of these policies are described below:

- *Greedy Energy Saver Policy (GESP):* For the sake of energy conservation, this policy switches on the minimum number of resources required for AR requests. Then, BE requests can be scheduled in the remaining available time slots (we call it scheduling fragments) of AR leases.
- *SLA-oriented Energy Saving Policy (SESP):* This policy favours the BE requests by trying to ensure that the MAWT is not violated. Therefore, whenever the violation risk is high ($V \geq 1$), the policy switches on resources based on the number of resources required by the request.

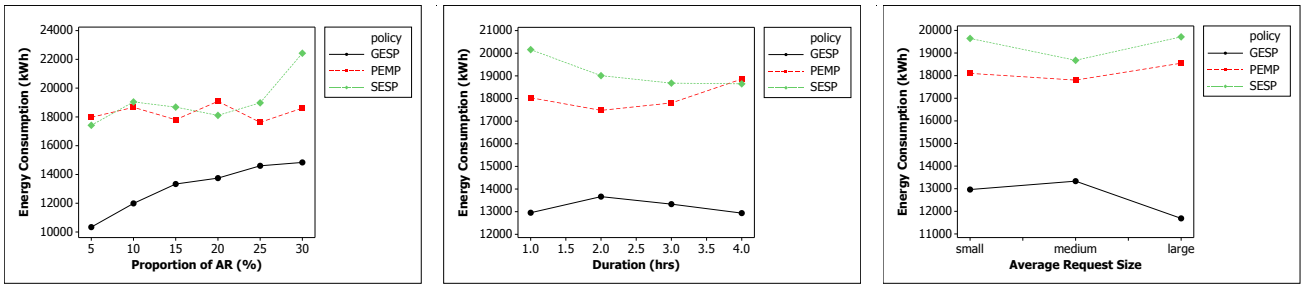
In the implementation of PEMP we have considered waiting factor as 5 ($\alpha = 5$).

2) Workload Model: To have a combination of AR and BE requests, similar to Sotomayor et al. [7], [1], we extract 30 days of job submissions from the trace (5545 submissions) and treat them as BE requests. Then, an additional set of AR requests are interleaved into the trace. We keep BE requests fixed and generate 72 different workloads by varying AR request characteristics as follows:

- ρ, the aggregate duration of all AR requests within a workload which is computed as a percentage of the total CPU hours in the whole workload. We investigate values of $\rho = 5\%, 10\%, 15\%, 20\%, 25\%, 30\%$. The reason that we do not explore larger values for ρ is that, in practice, the datacenters' utilization is between 30% to 50% [14], [20]. Considering that the trace's utilization (BE requests) is 34.8%, the overall utilization (BE and AR) is between 39.8% up to 64.8%.
- δ, the average duration of AR requests. In the experiments we explore the values of $\delta = 1, 2, 3, 4$ hours which is similar to the trace's duration. For generating the duration of AR requests, we select the duration randomly in the range of $\delta \pm 30$ minutes.
- θ, the number of nodes requested by each request. For this parameter we use 3 distinct ranges, namely, *small* (between 1 and 24). *medium* (between 25 and 48), and *large* (between 49 and 72). We choose the number of requested nodes for each request based on a uniform distribution.

Based on the above parameters we can work out how many AR requests are going to be generated. Using the number of generated AR requests in 30 days, we can find out the average arrival rate of AR requests in each day (λ). Then, the individual interval between two AR requests is randomly selected in the range of ($\lambda - 1$ hour and $\lambda + 1$ hour).

To investigate the impact of each parameter, in each experiment, we modify one of the above parameters while keeping the rest constant. When we modify ρ, we keep $\delta = 3$ hours and $\theta = medium$. When δ is changed, we keep $\rho = 15\%$ and $\theta = medium$. Finally, when θ is modified, the values of $\rho = 15\%$ and $\delta = 3$ hours. It is worth mentioning that changing δ and θ are performed in a way that the aggregate duration of all AR requests (ρ)



(a) (b) (c)

Figure 1. Power consumption of different policies. The experiment is performed by modifying: (a) the percentage of time taken by AR requests (ρ) where $\delta = 3$ hours and $\theta = medium$ (b) the average duration of AR requests (δ) changes (in hours) where $\rho = 15\%$ and $\theta = medium$ (c) the average size of AR requests (θ) where $\rho = 15\%$ and $\delta = 3$.

remains constant. This implies that increasing δ or θ lead to fewer AR requests.

The results of the experiments are studied from the practical and statistical perspectives. In statical analyses we applied T-student tests and we ensured about the normality of the underlying data.

Overheads involved in dealing with VMs such as suspend/resume time, and boot up and shut down time are also considered by Haizea and they are calculated according to Sotomayor et al. [1]. To measure the energy consumption of the cluster, we use the consumption information provided by the results of SPECpower benchmark[3]. Based on these information, the consumption of a resource with similar configuration is on average 117 watts, when it is utilized.

B. Experimental Results

1) Energy Consumption: In this experiment we measure the amount of energy consumed by each policy to run the workload trace. To measure the energy consumption, we calculate the overall time that the datacenter resources were switched on and we report the results in kWh.

Figure 1, expresses the amount of energy consumed when different policies are applied. In all subfigures of this figure we notice that GESP leads to the lowest energy consumption since it conservatively switches on resources (i.e., when they are required by AR requests).

More specifically, Figure 1(a) illustrates that PEMP remarkably consumes less energy than SESP (around 18% or 4000 kWh) when a considerable portion of requests are AR (more than 25%). However, PEMP and SESP are performing very similar when the proportion of AR requests are low (less than 25%). In fact, when the proportion of AR requests is low, preemption does not take place frequently, and, therefore, BE requests have more opportunity for running. Thus, policies that try to avoid violation do not come to the picture and result into the similar amount of consumed energy.

[3]http://www.spec.org/power-ssj2008/

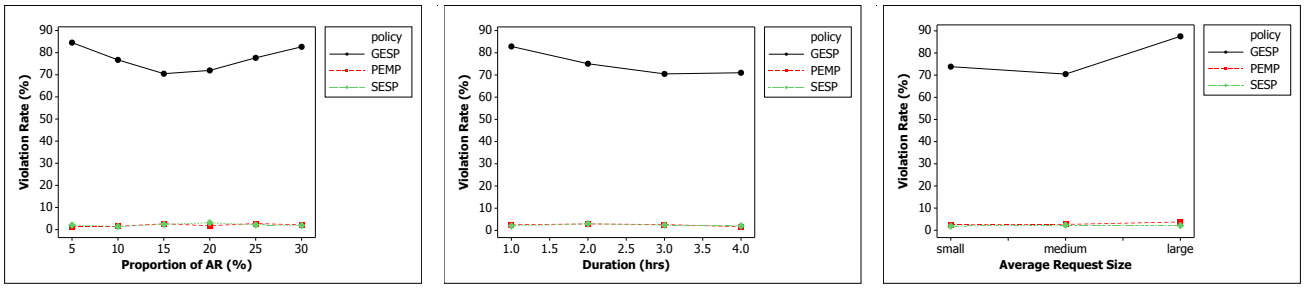
In Figure 1(b) and 1(c), we observe a decrease in energy consumption of GESP. The reason is that GESP switches on resources when there is an AR request. However, when the AR requests are long (Figure 1(b)) or their size are big (Figure 1(c)), fewer AR requests are generated, as discussed in Section IV-A2, to keep the proportion of AR requests constant. Accordingly, fewer resources are switched on and thus the energy consumption is reduced.

Additionally, in Figure 1(b) and 1(c), we observe that PEMP considerably consumes less energy than SESP. Particularly, when AR requests become shorter (Figure 1(b)), the difference becomes more significant. T-test analysis between PEMP and SESP, in Figure 1(b), for durations less than 4 hours shows that 95% confidence interval of the average difference is (193.5, 2830.1) kWh (P-value$<$0.001). Also, 95% confidence interval of the average difference between PEMP and SESP, in Figure 1(c), is (360.8, 2030) kWh (P-value=0.04). These values suggest that the difference between PEMP and SESP is statistically and practically significant. In fact, when AR requests are small or short, more gaps remain for scheduling BE requests. Thus, violation risk for BE requests is reduced which leads to more consolidation opportunity and less energy consumption.

In Figure 1(b), we notice that the energy consumption resulted from PEMP rises as the duration of AR requests increases (specially, when the duration is 4 hours). The reason is that when the AR requests are long (i.e. duration is increased), BE leases are postponed in scheduling for a long time. Therefore, the resources have to remain switched on for longer time and this implies more energy consumption.

2) Maximum Average Waiting Time Violation Rate (violation rate): In this experiment, we measure the percentage of violations from the MAWT that has taken place when different policies are applied. For this purpose, we report the percentage of BE requests that their waiting time was beyond the MAWT.

Results of the experiment, in all subfigures of Figure 2, reveal that PEMP is performing very close to SESP. However, the energy consumption resulted from these policies (see



(a) (b) (c)

Figure 2. Percentage of maximum waiting time violation resulted from different policies. The experiment is performed by modifying: (a) the percentage of time taken by AR requests (ρ) where $\delta = 3$ hours and $\theta = medium$ (b) the average duration of AR requests (δ) changes (in hours) where $\rho = 15\%$ and $\theta = medium$ (c) the average size of AR requests (θ) where $\rho = 15\%$ and $\delta = 3$.

Figure 1) show that PEMP leads to less energy consumption without increasing the violation rate. Additionally, in all of the subfigures, as expected, GESP leads to very high violation rates due to switching on few resources.

Subfigures of Figure 2, express that the violation rate of SESP and PEMP are almost unchanged as ρ, δ, and θ vary. This does not mean that the violation rate is not dependent on these parameters. In fact, in these policies the number of switched on resources are changed as ρ, δ, and θ are altered (see Figure 1) and, therefore, the violation rate does not vary significantly.

V. Conclusion

In this paper, we investigated the VM preemption as a way to reduce energy consumption in datacenters, where some requests have preemptive priority over the others. Our proposed energy management policy (PEMP) applies a fuzzy reasoning engine to determine if the resources for a request have to be allocated through switching on resources, preemption, consolidation, or a combination of these. We implemented PEMP in Haizea, as a real scheduling platform for datacenters, and evaluated under realistic conditions. Experimental results reveal that PEMP reduces the energy consumption by up to 18% (4000 kWh), over the course of 30 days, and without significant starvation of lower priority requests, compared to other baseline policies. In future we plan to consider the impact of VM migration as another possible action in preemption. Additionally, we plan to investigate the effect of consolidation policies on the proposed policy.

References

[1] B. Sotomayor, K. Keahey, and I. Foster, "Combining batch execution and leasing using virtual machines," in *Proceedings of the 17th International Symposium on High Performance Distributed Computing*, USA, 2008, pp. 87–96.

[2] J. Walters, B. Bantwal, and V. Chaudhary, "Enabling interactive jobs in virtualized data centers," *Cloud Computing and Applications*, 2008.

[3] P. Xavier, W. Cai, and B.-S. Lee, "A dynamic admission control scheme to manage contention on shared computing resources," *Concurrency and Computing: Practice and Experience*, vol. 21, pp. 133–158, February 2009.

[4] R. Kettimuthu, V. Subramani, S. Srinivasan, T. Gopalsamy, D. K. Panda, and P. Sadayappan, "Selective preemption strategies for parallel job scheduling," *International Journal of High Performance and Networking (IJHPCN)*, vol. 3, no. 2/3, pp. 122–152, 2005.

[5] M. Amini Salehi, B. Javadi, and R. Buyya, "Resource provisioning based on leases preemption in InterGrid," in *Proceeding of the 34th Australasian Computer Science Conference (ACSC 2011)*, Perth, Australia, 2011, pp. 25–34.

[6] M. A. Salehi, B. Javadi, and R. Buyya, "Qos and preemption aware scheduling in federated and virtualized grid computing environments," *Journal of Parallel and Distributed Computing (JPDC)*, vol. 72, no. 2, pp. 231–245, 2012.

[7] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Resource leasing and the art of suspending virtual machines," in *Proceedings of the 11th IEEE International Conference on High Performance Computing and Communications*, Washington, DC, USA, 2009, pp. 59–68.

[8] M. A. Salehi, B. Javadi, and R. Buyya, "Preemption-aware admission control in a virtualized grid federation," in *Proceeding of 26th International Conference on Advanced Information Networking and Applications (AINA'12)*, Japan, 2012, pp. 854–861.

[9] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, 2011.

[10] H. Lagar-Cavilla, J. Whitney, A. Scannell, P. Patchin, S. Rumble, E. De Lara, M. Brudno, and M. Satyanarayanan, "Snowflock: rapid virtual machine cloning for cloud computing," in *Proceedings of the 4th ACM European conference on Computer systems*. ACM, 2009, pp. 1–12.

[11] J. Berral, Í. Goiri, R. Nou, F. Julià, J. Guitart, R. Gavaldà, and J. Torres, "Towards energy-aware scheduling in data

centers using machine learning," in *Proceedings of the 1st International Conference on energy-Efficient Computing and Networking*. ACM, 2010, pp. 215–224.

[12] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster computing*, vol. 12, no. 1, pp. 1–15, 2009.

[13] V. Petrucci, O. Loques, B. Niteroi, and D. Mossé, "Dynamic configuration support for power-aware virtualized server clusters," *WiP Session of the 21th Euromicro Conference on Real-Time Systems. Dublin, Ireland*, 2009.

[14] A. Verma, P. Ahuja, and A. Neogi, "pmapper: power and migration cost aware application placement in virtualized systems," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. Springer-Verlag New York, Inc., 2008, pp. 243–264.

[15] J. Kephart, H. Chan, R. Das, D. Levine, G. Tesauro, F. Rawson, and C. Lefurgy, "Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs," in *Proceedings of the Fourth International Conference on Autonomic Computing*. IEEE Computer Society, 2007, p. 24.

[16] L. Wang, "Adaptive fuzzy systems and control- design and stability analysis(book)," *Englewood Cliffs, NJ: PTR Prentice Hall*, 1994.

[17] J. Fontán, T. Vázquez, L. Gonzalez, R. S. Montero, and I. M. Llorente, "OpenNebula: The open source virtual machine manager for cluster computing," in *Open Source Grid and Cluster Software Conference – Book of Abstracts*, San Francisco, USA, May 2008.

[18] "Parallel workloads archive." http://www.cs.huji.ac.il/labs/parallel/workload/.

[19] Q. Snell, M. J. Clement, and D. B. Jackson, "Preemption based backfill," in *Job Scheduling Strategies for Parallel Processing (JSSPP)*. Springer, 2002, pp. 24–37.

[20] P. Bohrer, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony, "The case for power management in web servers," *Power aware computing*, vol. 78758, 2002.