An Advanced Reinforcement Learning Approach for Energy-Aware Virtual Machine Consolidation in Cloud Data Centers

Conference Paper · December 2017

DOI: 10.23919/ICITST.2017.8356347

CITATIONS
27

READS
603

3 authors:

Rachael Shaw
Galway-Mayo Institute of Technology
9 PUBLICATIONS **123** CITATIONS

SEE PROFILE

Enda Howley
National University of Ireland, Galway
96 PUBLICATIONS **1,644** CITATIONS

SEE PROFILE

Enda Barrett
46 PUBLICATIONS **1,368** CITATIONS

SEE PROFILE

An Advanced Reinforcement Learning Approach for Energy-Aware Virtual Machine Consolidation in Cloud Data Centers

Rachael Shaw, Enda Howley, Enda Barrett
Department of Information Technology
National University of Ireland Galway
Galway, Ireland
r.shaw4@nuigalway.ie

Abstract— **Energy awareness presents an immense challenge for cloud computing infrastructure and the development of next generation data centers. Inefficient resource utilization is one of the greatest causes of energy consumption in data center operations. To address this problem we introduce an Advanced Reinforcement Learning Consolidation Agent (ARLCA) capable of optimizing the distribution of virtual machines across the data center for improved resource management. Determining efficient policies in dynamic environments can be a difficult task, however the proposed Reinforcement Learning (RL) approach learns optimal behaviour in the absence of complete knowledge due to its innate ability to reason under uncertainty. Using real workload data we evaluate our algorithm against a state-of-the-art heuristic, our model shows a significant improvement in energy consumption while also reducing the number of service violations.**

Keywords- energy efficiency; cloud computing; resource management; reinforcement learning

I. INTRODUCTION

Cloud computing services offered by companies such as Amazon and Google deliver on-demand virtualized resources which can be accessed over the internet and charged using a pay-as-you-go pricing model [1]. The ability to scale up or down computing resources in response to current demand has led to the tremendous growth and wider adoption of cloud computing across several domains. Despite the advancements in energy efficient computing devices and data center equipment, excessively high energy consumption and carbon dioxide emissions continue to soar in the operation of large scale data centers today. Recent studies in cloud computing have highlighted the environmental impact of data centers in terms of electricity costs and associated CO^2 emissions. In 2013, U.S data centers consumed a total of 91 billion kilowatt-hours of electricity. By 2020 the level of consumption is estimated to increase to approximately 140 billion kilowatt-hours annually, costing 13 billion per year in electricity bills and furthermore, pollution of 150 million metric tons of carbon dioxide [2]. According to GeSI SMARTer 2020 report data center emissions are projected to increase by 7% annually [3]. In addition, a study by Petty et al. highlighted that the Information Communication Technology (ICT) industry contributes to 2% of global CO^2 emission annually which is the equivalent to that produced by the aviation industry [4].

The era of on-demand computing is powered by the concept of virtualization which was pioneered by IBM in the 1960's [5]. Virtualization aids in the promotion of increased host utilization by apportioning the resources of large physical hosts into smaller independent machines each of which is known as a Virtual Machine (VM). Each VM runs in apparent isolation equipped with its own Operating System (OS) and applications which allows for the simultaneous execution of multiple tasks on a physical host. This overall results in increased efficiency and resource utilization.

However, despite the benefits gained from the advancements in virtualization technologies one of the major inefficiencies in data center deployments is caused by poorly managed and idle resources [6]. Current studies have revealed that on average hosts operate at a mere 12-15% of their full capacity resulting in the wastage of valuable resources while underutilized hosts have been proven to use up to 60% of their maximum power resulting in significant draws on energy consumption [2]. VM consolidation is one approach that can significantly improve the management of resources by strategically reallocating VMs on to a reduced number of hosts in an effort to conserve energy. Consolidating a larger number of VM instances on an already loaded host can however, cause a surge in energy consumption while also potentially causing the host to become overloaded incurring service violations and further requiring VMs to be migrated to additional hosts in the datacenter [7]. Conversely, placing a VM on an underutilized host promotes the continuation of poor resource utilization. As a result, striking a balance between both energy efficiency and performance is essential to achieving high performance while reducing overall energy consumption.

To address this issue we present a self-optimizing Reinforcement Learning (RL) VM consolidation model to optimize the allocation of VM instances while also adhering to strict Service Level Agreements (SLA). The agent continues to learn an optimal resource allocation policy depending on the current state of the system through repeated interactions with the environment. In addition, our model employs an advanced reward shaping technique known as Potential Based Reward

Shaping (PBRS). One of the more profound limitations of standard RL algorithms is the slow rate at which they converge to an optimal policy [8]. PBRS allows expert advice to be included in the learning model to assist the agent to learn more rapidly and as a result encouraging more optimal decision making in the earlier stages of learning. Our results show the RL agents ability to learn and adapt in a highly volatile cloud environment while also delivering an intelligent energy-performance tradeoff capability resulting in improved energy efficiency and performance.

The contributions of this paper are the following:

- Using a state-of-the-art RL technique we present an autonomous VM consolidation model capable of optimizing the distribution of VMs across the data center in order to achieve greater energy efficiency while also delivering the required performance.
- We apply our model to a large scale simulated data center using CloudSim. Using real workload traces we show the advantages of our approach over a state-of-the-art heuristic.

The remainder of this paper is structured as follows: Section II discusses related work, Section III introduces RL, Section IV presents the design of our proposed cloud resource management model, Section V describes our experimental setup and performance metrics, Section VI presents the results and Section VIII concludes the paper.

II. Related Work

In recent years the pervasiveness of cloud computing has urged research initiatives to tackle the challenging problems of inefficient resource management policies. In the literature two types of approaches are used, these can be classified as Heuristic/Threshold and AI based approaches.

A. Heuristic/Threshold Based Approaches

Heuristic/Threshold based approaches are the most widely used methodologies for resource management in cloud infrastructure. These types of approaches trigger static resource allocation decisions on reaching a predefined threshold. Verma et al. presented pMapper an application placement controller which aims to minimize energy consumption and migration costs while maintaining SLA [9]. The underlying architecture is composed of three distinct management entities, namely a performance manager which monitors current performance and resizes VM instances based on the SLA, A power manager which manages the power state of underlying hardware and a migration manager which estimates the cost of live migration. Their overall results showed a savings in power consumption of hosts of 25%. Cardosa et al. investigated the impact on VM resource allocation and power consumption by leveraging min, max and share parameters analogous to those used in commercial virtualization technologies [10]. These parameters define the upper and lower bounds for resource utilization for each VM while the shares parameter denotes the priority of each VM during the distribution of spare resources. The authors use such parameters to drive their VM placement and consolidation strategy which achieved a significant improvement in the overall utility of the data center. Lee et al. proposed the implementation of two task consolidation heuristics known as ECTC and MaxUtil in order to curtail energy consumption of underutilized resources [6]. One of the most important research contributions in the field of VM consolidation is accredited to the work of Beloglazov et al. as outlined in two of their more highly cited papers [7][11]. They introduced a three stage VM resource optimization approach consisting of Host overutilization/underutilization detection, VM selection and VM placement. In particular, they proposed the Lr-Mmt algorithm which manages host utilization detection and VM selection. In addition, this algorithm uses a VM placement heuristic known as Power Aware Best Fit Decreasing (PABFD). This heuristic considers the heterogeneity of cloud resources by selecting the most energy efficient hosts first in order to allocate VMs. Their experimental results concluded that the composition of Lr-Mmt in conjunction with PABFD significantly outperforms all other VM consolidation procedures resulting in a profound reduction in energy consumption and SLA violations.

B. Reinforcement Learning Based Approaches

Alternatively, research initiatives continue to explore the application of AI methodologies in order to evolve a new generation of autonomic resource management strategies. Duggan et al. introduced an RL network-aware live migration strategy, their model enables an agent to learn the optimal times to schedule VM migrations to improve the usage of limited network resources [12]. Tesauro et al. presented an RL approach to discover optimal control policies for managing Central Processing Unit (CPU) power consumption and performance in application servers [13]. Their proposed method consists of an RL based power manager which optimizes the power performance tradeoff over discrete workloads. Barrett et al. applied a parallel RL learning approach to optimize resource allocation in the cloud [14], while other work introduced an agent based learning architecture for scheduling workflow applications [15]. Rao et al. introduced VCONF an RL based VM auto configuration agent to dynamically re-configure VM resource allocations in order to respond effectively to variations in application demands [16]. VCONF operates in the control domain of virtualized software, it leverages model based RL techniques to speed up the rate of convergence in nondeterministic environments. Das et al. introduced a multiagent based approach to manage the power performance tradeoff by specifically focusing on powering down underutilized hosts [17]. Dutreil et al. also showed how RL methodologies are a promising approach for achieving autonomic resource allocation in the cloud. They proposed an RL controller for dynamically allocating and deallocating resources to applications in response to workload variations [18]. Using convergence speed up techniques, appropriate Q function initializations and model change detection mechanisms they were able to expedite the learning process. Farahnakian et al. [19] implemented RL to learn the optimal time to power on or switch off a host based on future resource demands.

Compared to related research our work is different in that:

1) Using a PBRS inspired RL technique we present Advanced Reinforcement Learning Consolidation

The primary author would like to acknowledge the ongoing financial support provided by the Irish Research Council.

Agent (ARLCA) a self-optimizing VM consolidation approach that allocates the optimal number of VMs to hosts such that each host operates at an optimized resource usage rate. We show how this approach optimizes three key performance metrics namely energy consumption, VM migrations and service violations.

2) Unlike static threshold based approaches this approach is more suitable for decision making in highly dynamic environments while also considering allocation decisions that could possibly suffer from delayed consequences. We show using real workload traces how our model outperforms a state-of-the-art heuristic approach across all performance metrics in order to deliver a more sustainable green cloud infrastructure.

III. REINFORCEMENT LEARNING

Reinforcement Learning enables an agent to learn optimal behaviour through repeated trial and error interactions with its environment at discrete time steps *t* without any prior knowledge. The agent receives a reward depending on the action selected. The objective of the agent is to discover overtime which actions yield the greatest rewards [20].

RL control problems can be intuitively modelled as Markov Decision Processes (MDP) which provide a model for sequential decision making problems faced with adverse uncertainty [14]. The learning process for an RL agent is composed of:

1) State space: A set of environment states. At the end of each time step *t* the learning agent occupies a state denoted $s_t \in S$.

2) Action space: The agent selects a possible action $a_t \in A(s_t)$ where $A(s_t)$ refers to the set of all possible actions in the current state s_t.

3) Reward signal: Once the selected action is executed it results in a state transition s_{t+1}, the agent is then allocated a positive or negative reward signal $R(s_t,a_t)$ depending on the state of the environment after an action has occurred. The overall goal of MDP is to generate a mapping of states to associated actions which maximize the accumulated reward.

Sarsa is a popular RL Temporal Difference (TD) learning algorithm which can be used to discover an optimal policy. TD methodologies implement prediction based learning by incrementally updating current estimates of state-action pairs $Q(s_t, a_t)$ based on the outcome of previous estimates. Sarsa is an acronym for state, action, reward, state, action. Its name is derived from the sequence of events that must occur in order to transition from one state to the next and update the current model estimates. The update rule is as follows:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t)+\alpha[r_{t+1} +\gamma\, Q(s_{t+1},a_{t+1})-Q(s_t,a_t)]. \quad (1)$$

where $Q(s_t,a_t)$ denotes the expected reward of selecting action a_t in state s_t. α is the learning rate, a value set close to 1 promotes continuous updates to estimates while α defined close to 0 reduces learning. γ is a discount factor which determines the degree to which an agent favors long term rewards over short term gains. A value closer to 1 results in an agent that is more forward looking and strives to maximize future rewards while a rate closer to 0 results in an agent that assigns a greater weight to short term rewards. r_{t+1} defines the reward signal allocated for selecting an action in a given state while $Q(s_{t+1}, a_{t+1})$ is the Q-value estimate of the resulting state and the action selected in the next time step *t*.

A policy π guides the agent's decision making process when selecting an appropriate action for any given state. In order to discover an optimal policy there is a tradeoff between exploration and exploitation. An agent that invariably exploits the best action fails to discover potentially more lucrative actions by choosing to explore its environment. In order to manage such a tradeoff we implement a softmax action selection strategy [21]. Softmax assigns action probabilities according to the expected utility, thus ensuring higher rewarding actions are more likely to be explored. Illustrated below is the standard Sarsa learning algorithm.

Algorithm 1*: Sarsa*

Initialize *Q(s,a)* arbitrarily

Repeat (for each episode):

  Initialize *s*

  Choose *a* from *s* using π

  Repeat (for each step of episode):

    Take action *a*, observe *r*, *s'*

    Choose *a'* from *s'* using π

    $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma\, Q(s',a') - Q(s,a)]$

    $s \leftarrow s'$; $a \leftarrow a'$;

  until *s* is terminal

One of the limitations of RL methodologies is the rate at which an agent converges to an optimum policy. In order to expedite the learning process we implement PBRS which has shown to be a powerful technique to improve the convergence rate of RL agents by using domain knowledge to assist the agent to learn more rapidly [22]. In particular, PBRS provides the learner with an additional reward through the mapping of states to associated potentials using the following function:

$$F(s,a,s') = \gamma\, \Phi(s') - \Phi(s). \quad (2)$$

where Φ is the potential function which maps states to potentials and γ is defined as the same discount factor applied in the update rule (1). The PBRS reward is concatenated to the standard reward received from the environment. Using the results of previous experiments we fine-tuned the potentials with more lucrative rewards for allocation decisions resulting in host utilization states between 40-70% as this resulted in the best performance according to our experimental analysis.

The primary author would like to acknowledge the ongoing financial support provided by the Irish Research Council.

IV. DYNAMIC VM CONSOLIDATION MODEL

In order to reach new frontiers in energy efficient cloud infrastructure we propose an Advanced RL Consolidation Agent known as ARLCA which is capable of driving both efficiency and Quality of Service (QoS) by dynamically adjusting its behaviour in response to changes in workload variability.

More specifically, ARLCA is presented with a list of VMs that require allocation to suitable hosts in the datacenter. Through repeated interactions with the environment ARLCA discovers the optimal balance in the dispersal of VMs across the data center so as to prevent hosts becoming overloaded too quickly but also ensuring that resources are operating efficiently.

A. State-Action Space

The application of RL in more complex problem domains such as the cloud requires careful definition of the state-action space in order for it to operate effectively. We design a novel state-action space by defining all possible states and actions as a percentage ranging from 0-100%. We use increments of 1% which provided the best performance according to an experimental parameter sweep. The state space S denoted below in (3) represents the global state of the environment. It can be defined as the number of active hosts a_h in the environment as a percentage of the total number of hosts t_h.

$$S=\frac{\sum_{i=1}^{n} a_h}{t_h}\times 100. \quad (3)$$

An action *A* is a combined variable composed of the utilization rate of any given host coupled with the size of the VM to be placed. As denoted below in (4) the host utilization rate h_u is calculated as the sum of the total requested resources *trr* for each VM residing on the host as a percentage of the hosts' capacity h_c. Additionally, VM utilization *vmu* is computed as the VMs requested resources *r* returned as a percentage of the total host capacity *hc*. This determines the size (CPU resource requirements) of the VM to be placed. Defining the state-action space as percentages from 0-100% significantly reduces the size of the state-action space thus preventing the agent from engaging in an exhaustive search. This type of approach allows for the deployment of a more agile and efficient agent capable of pursuing its design objectives.

$$A=\left[h_u=\frac{\sum_{j=1}^{n} trr}{h_c}\times 100+\ vmu=\frac{r}{h_c}\times 100\right]. \quad (4)$$

B. ARLCA Learning Model

The Sarsa driven learning algorithm coupled with the advanced PBRS component used to train our agent is presented above. When invoked the ARLCA learning algorithm calculates the global state of the environment using (3). The first VM to be placed is selected from the placement list and the CPU host utilization rate for each host is calculated and returned as a percent ranging between 0-100%. Next the size of the VM is computed and a list of possible actions is generated using the combined action variable (4). The agent then selects a host based on the softmax action selection strategy and the host is placed on a migration list which keep a record of allocation decisions. The global state is recalculated and both the MDP and PBRS rewards are generated. Sarsa updates Q-value estimates based on the action that will be implemented in the subsequent state, in order to do so the utilization rates of the hosts and the size of the next VM is recalculated and used to update the Q-value estimate. The result is stored in the Q-value matrix which effectively stores the mapping of states to associated actions representing the agents current knowledge. Lastly, the global state is updated and the last action selected is implemented in the subsequent iteration. ARLCA continues to execute until all VMs are mapped on to various hosts in the environment.

Algorithm 2*: ARLCA Learning Algorithm*

Input : VM Placement List
calculate *globalState*
foreach host→ *hostList* **do**
  calculate *hostUtil*
end
calculate *vmSize*
calculate *possibleActions*← *vmSize* + *hostUtil*
select *host* from *possibleActions* using π
foreach *vm*→ *vmPlacementList* **do**
  allocate *vm*
  observe *globalState+1,rewards*
  foreach *host* → *hostList* **do**
    calculate *hostUtil*
  end
  calculate *nextVmSize*
  calculate *possibleActions*←*nextVmSize* +*hostUtil*
  select *host* from *possibleActions* using π
  calculate Q(s,a) ←Q(s,a) + α[r + **F(s,s')**+ γ Q(s',a') -Q(s,a)]
  update *QValueMatrix*
  globalState ←*globalState+1*
  action← *host*
end
Output: mapping of VMs to hosts

V. EXPERIMENTAL SETUP

To evaluate the performance of the proposed energy-aware RL learning agent ARLCA we have selected the state-of-the-art PABFD heuristic as a benchmark. We develop an RL framework as an extension of the CloudSim simulator as used in the studies of Beloglazov et al. [7][11]. CloudSim supports the management of cloud resources and contains the necessary components to enable the empirical evaluation of energy-aware cloud based simulations. In order to simulate a large scale cloud

environment 800 HP ProLiant ML110 G5 hosts were configured in the data center. These hosts consisted of two cores with the capacity to process 2660 Million Instructions Per Second (MIPS). Our experiments leveraged the 10 day CPU workload traces provided by CloudSim which were generated from real hosts deployed in over 500 locations globally. In order to measure the robustness of the proposed approach we used these traces to generate a randomized 30 day workload which models more precisely the complexity and dynamic nature of the cloud environment overtime while also evaluating more thoroughly the capability of the proposed agent to learn in such an environment.

A. Performance Metrics

The key performance metrics to evaluate the effectiveness of the proposed algorithms are as follows:

1) ***Energy Consumption****:* This is defined as the total energy consumed by the data centers computational resources as a direct result of processing application workloads.

2) ***VM Migrations****:* This is the total number of VM migrations that occur during the simulation process. Each time a VM is migrated it is typically subjected to SLA violations.

3) ***SLA Violations****:* The ability of cloud providers to deliver SLA is critical and a core function of their operation and as a result we also consider the impact on SLA violations.

VI. RESULTS

We evaluate ARLCA against the Lr-Mmt policy which harnesses the state-of-the-art PABFD consolidation heuristic using the stochastic 30 day workload in order to demonstrate the benefits of a more adaptive and intelligent methodology.

Fig. 1 illustrates the behaviour of both policies in relation to energy consumption over the 30 day workload. As shown, the implementation of ARLCA resulted overall in a considerable reduction in energy by a total of 25.35% with an average energy savings of 39.7 kWh per day (Std Dev 20.49). Furthermore, it is also apparent that on day 28 ARLCA failed to outperform Lr-Mmt which resulted in a slight increase in energy of 0.5%. To analyze whether the overall energy reduction achieved is statistically significant a two tailed t-test was performed which resulted in a p-value of <0.0001 with a 96% confidence interval (32.068, 47.372). These results reveal that the energy savings achieved over the Lr-Mmt policy are extremely significant.

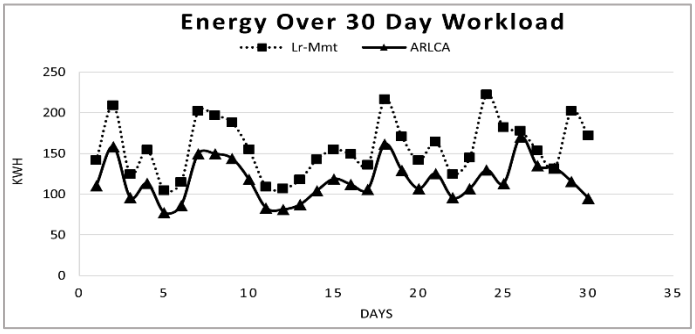

Figure 1. Energy Consumption

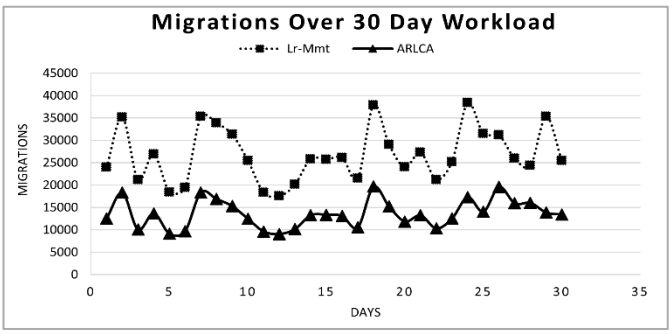

Figure 2. Migrations

Fig. 2 displays the number of migrations incurred by both policies over the 30 day workload. As illustrated, the application of ARLCA also had a positive impact on the number of migrations. ARLCA reduced migrations by 49.17% in total (394,587 migrations). In addition it reduced the mean number of migrations per day by 13,153 (Std Dev 3536.21). The results were also statistically significant with a p-value of <0.0001 with a 95% confidence interval (11832.46, 14473.34).

Fig. 3 presents the number of SLA violations incurred by both policies during the simulation. The Lr-Mmt policy resulted in a surge in the number of service violations while ARLCA showed an overall 63% decrease in the mean number of SLA violations. Again these results were also statistically significant with a p-value of 0.00001121 with a 95% confidence interval (0.21654124273, 048922395074).

An important dimension in achieving greater energy efficiency through resource optimization is managing the tradeoff between energy and performance which are inextricably linked. Notably, an interesting observation in Fig. 1 was that on day 28 the Lr-Mmt policy generated a slight improvement in energy consumption of a mere 0.5%. However, Fig. 3 confirms that this marginal improvement was achieved at the cost of increased SLA violations as indicated by the spike generated in the number of violations by the Lr-Mmt policy on day 28. This suggests that Lr-Mmt consolidated VMs more aggressively in an attempt to successfully reduce energy but failed to efficiently manage the energy-performance tradeoff resulting in a surge in the number of service violations. In contrast ARLCA strikes a more precise balance with such a tradeoff by generating a relatively similar energy rating of just .5% in the difference. However, more profoundly the agent also achieves a significant 76.2% decrease in the number of SLA violations on day 28 alone.

Overall the key points arising out of these results are that through the deployment of our advanced energy efficient learning agent ARLCA we introduce a more agile and adaptive solution to consolidate and support the movement of VMs between physical hosts in the data center. More specifically, our empirical results demonstrate the improved efficiency achieved by leveraging a more sophisticated and dynamic RL solution which has the inherent ability to efficiently adapt to a continuously changing cloud environment. As a result, we deliver a significant reduction in energy consumption of 25.35% with a decrease of up to 44.7% in energy per day over the state-of-the-art Lr-Mmt heuristic. Furthermore, we reduce

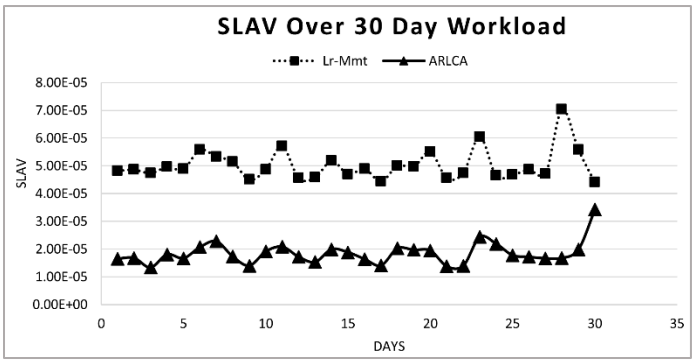

Figure 3. Service Level Agreement Violations

the number of service violations by a significant 63%. Such reductions reduce the overall data center operational costs resulting in a more competitive cloud infrastructure. This solution also has wider implications as it stands to provide a more sustainable green cloud infrastructure in support of global environmental sustainability while also promoting the greater adoption of AI techniques to achieve intelligent decision making in cloud based infrastructure.

VII. CONCLUSION

Through the innovative application of more sophisticated and advanced methodologies adopted from the field of Artificial Intelligence we developed ARLCA, an intelligent solution to optimize the distribution of VMs across the data center. ARLCA demonstrates the potential of more advanced intelligent solutions capable of reaching new frontiers in data center energy efficiency while also achieving significant improvements in the quality of the service provided.

However, there remains many open challenges that must be addressed in order to provide a more complete solution to this complex problem. One area in particular is energy efficient strategies which take into consideration the utilization of multiple systems resources. As a result, we intend on extending our proposed model to develop a solution which considers resources such as Random Access Memory (RAM) and also network bandwidth.

In addition, our results showed that even through the deployment of an intelligent agent service violations were still evident periodically. We also plan on exploring this further through the implementation of multi-objective optimization techniques where both energy and service violations are optimized simultaneously.

REFERENCES

[1] R. Shaw, E. Howley and E. Barrett, (in press) “Predicting the available bandwidth on intra cloud network links for deadline constrained workflow scheduling in public clouds”, International Conference on Service-Oriented Computing. Springer International Publishing, 2017.

[2] J. Whitney and P. Delforge, “Scaling up energy efficiency across the data center industry: evaluating key drivers and barriers”, tech. report, Natural Resources Defense Council, August, 2014.

[3] L. Neves, J. Krajewski, P. Jung and M. Bockemuehl, “GESI SMARTer 2020: The role of ICT in driving a sustainable future”, tech. report, Global e-Sustainability Initiative and The Boston Consulting Group, Inc., 2012.

[4] C. Pettey, "Gartner estimates ICT industry accounts for 2 percent of global co2 emissions", Gartner. April, 2007.

[5] P. Healy, T. Lynn, E. Barrett and J.P. Morrison, “Single system image: A survey”, Journal of Parallel and Distributed Computing, 90, 2016. pp.35-51.

[6] Y.C. Lee, and A.Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems", The Journal of Supercomputing, vol. 60, 2012, pp. 268–280.

[7] 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, vol. 24, 2012, pp. 1397-1420.

[8] M. Grześ, and D. Kudenko. "Multigrid reinforcement learning with reward shaping." Artificial Neural Networks-ICANN 2008, 2008, pp.357–366.

[9] A. Verma, P. Ahuja and A. Neogi, "pMapper: power and migration cost aware application placement in virtualized systems", Proc. 9th ACM/IFIP/USENIX International Conf. on Middleware, 2008, pp. 243–264.

[10] M. Cardosa, M.R. Korupolu and A. Singh, "Shares and utilities based power consolidation in virtualized host environments", Proc. 11th IFIP/IEEE International Conf. on Symposium on Integrated Network Management, 2008, pp. 327–334.

[11] A. Beloglazov, J. Abawajy and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", Future Generation Computer Systems, vol. 28 , 2012, pp. 755–768.

[12] M. Duggan, J. Duggan, E. Howley and E. Barrett, “A network aware approach for the scheduling of virtual machine migration during peak loads”, Cluster Computing, vol 20 ,2017, pp.1-12.

[13] G. Tesauro, R. Das, H. Chan, J. Kephart, D. Levine, F. Rawson and C. Lefurgy, "Managing power consumption and performance of computing systems using reinforcement learning", Proc. 20th International Conf. on Neural Information Processing Systems, 2007, pp.1497–1504.

[14] E. Barrett, E. Howley, and J. Duggan. "Applying reinforcement learning towards automating resource allocation and application scalability in the cloud", Concurrency and Computation: Practice and Experience, vol.25, 2013, pp.1656–1674.

[15] E. Barrett, E. Howley and J. Duggan, “A learning architecture for scheduling workflow applications in the cloud”. In Web Services (ECOWS), 2011 Ninth IEEE European Conference on, 2011, pp. 83-90. IEEE.

[16] J. Rao, X. Bu, C. Z. Xu, L. Wang and G. Yin, "VCONF: A reinforcement learning approach to virtual machines auto-configuration", Proc. 6th International Conf. on Autonomic Computing, 2009, pp. 137–146.

[17] R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D. W. Levine and H. Chan, "Autonomic multi-agent management of power and performance in data centers", Proc. 7th International Joint Conf. on Autonomous Agents and Multiagent Systems: Industrial Track (International Foundation for Autonomous Agents and Multiagent Systems), 2007, pp.107–114.

[18] X. Dutreilh, S. Kirgizov, O. Melekhova, J. Malenfant, N. Rivierre and I. Truck, "Using reinforcement learning for autonomic resource allocation in clouds: towards a fully automated workflow", Proc. 7th International Conf. on Autonomic and Autonomous Systems, 2011, pp.67–74.

[19] F. Farahnakian, P. Liljeberg, and J. Plosila, "Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning." Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on. IEEE, 2014.

[20] R.S. Sutton, and A.G. Barto. Reinforcement learning: An introduction. Cambridge: MIT press, 1998.

[21] R. Shaw. “An artificial intelligence model for autonomous resource allocation in cloud computing environments”, Masters thesis. National University of Ireland Galway, 2016.

[22] A.Y Ng, D. Harada and S. Russell, “Policy invariance under reward transformations: Theory and application to reward shaping", Proc. 16th International Conf. of Machine Learning, 1999, pp.278–287.