Journal of Positive School Psychology http://journalppw.com
2022, Vol. 6, No. 3, 10242-10247

A Load And Energy Aware Virtual Machine Scheduling Approach For Data Centres

Minakshi Kamboj [1] and Sanjeev Rana [2]

[1] *Research Scholar, CSE Department MMEC, M.M. (Deemed to be University), Mullana, Ambala, Haryana, India,Email id: kamboj.meena2@gmail.com*
[2] *Associate Professor, MMEC, M. M. (Deemed to be University), Mullana Ambala, Haryana, India Email id: dr.sanjeevrana@mmumullana.org*

Abstract

A data centre frequently doubles as a cloud computing hub, offering services utilising its computing and storage resources. The host machines in data centres are typically virtualized, which introduces a tough research problem, namely, how to schedule the virtual machines (VM) on the hosts for energy efficiency. This enables on-demand resource provision with elasticity and high dependability. The purpose of this work is to improve data centres' energy efficiency through scheduling. The design and implementation of a novel VM scheduling method will be suggested to support this task. With the ultimate goal of lowering a data centre's energy consumption, this technique handles both load-balancing and temperature awareness.

Keywords: Data Center, Information Technology, Virtual Machine, Service Level Agreement, First-in-First out

1. Introduction

Data canters are buildings made up of a number of computers that operate continuously, consuming a lot of power. Within the next three to four years, the computational capacity of data centres will double; as a result, data centres will need to increase their energy efficiency to control energy growth [2]. The computing (or IT) power is the main source of data processing power [3]. By using an energy-aware task scheduling approach to arrange the tasks in the servers efficiently, energy-efficient strategies, including resource management, are employed to lower the IT power consumption. The primary goal of our plan is to schedule virtual machines in consideration of processor temperature. In order to characterise the behaviour of this parameter while applications (in this case, virtual machines) are operating, such a scheduling strategy needs a temperature model. Due to its simplicity, the lumped RC thermal model (Skadron et al., 2002) will be utilised. The temperature threshold and the utilisation threshold are two thresholds that the implemented scheduler uses to make migration decisions.

2. Related Work

A dedicated monitoring mechanism is used by Sirbu et al. [3] to anticipate the power usage of. A performance model for lengthy scientific applications is presented by Sadjadi et al. in [4]. Without utilising intrusive approaches like instrumentation or code inspection, a comprehensive profile of the application's execution is performed in order to create the performance model. According to Wong et al. [5,] the performance of parallel applications is predicted using the application signature. By running the entire application on platform A, the application signature may be extracted. A alternative platform B's performance is then predicted using the application signature. The performance is predicted by Yang et al. [6] using a portion of the original application's execution. They fully run the application in order to forecast how it will function on a new platform using the same methodology. To help supercomputers save more energy, Auweter et al. [7] describe an energy-aware job scheduler. They present a prediction model that predicts the effectiveness and capability of large-scale applications. An energy-aware scheduler that may be used with HPC data canters is demonstrated by Mämemelä et al. [8]. They employed FIFO (First In First Out) and backfilling schedulers in energy-conscious versions. A Mixed Integer Linear Programming task scheduling strategy for simultaneous independent jobs is presented by Goldman et al. [9]. A task scheduling strategy using sequential Linear Programming approximations is demonstrated by Chretien et al. in [10]. In order to determine the ideal makespan, they used an iterative linear programming technique. In comparison to Integer Linear Programming scheduling approaches, metaheuristic approaches can find nearly optimal solutions in a lot less computation time. For green data centres, Lei et al. [11] developed a scheduling strategy based on a co-evolutionary algorithm. A task scheduling technique based on Simulated Annealing is presented by Kashani et al. [12] to reduce the makespan in distributed systems. An energy-aware method to allocate virtual machines effectively in a cloud-oriented scenario is proposed by Beloglazov et al. in paper number 13 [13]. For long-running applications, a cluster scheduler is shown by Garefalakis et al. [14]. The allocation of energy-efficient resources (CPU, memory, storage, etc.) and resource utilisation are the two most difficult problems in a big cloud data centre, according to Minakshi Kamboj et al. [15]. Numerous options for evaluating the energy efficacy of data centres were presented by Minakshi Kamboj et al. [16].

3. Proposed Work

The temperature threshold and the utilisation threshold are two thresholds that the implemented scheduler uses to make migration decisions. Utilizing the modelling environment CloudSim, which simulates massive data centres providing computing infrastructures as services, the proposed scheduler is assessed.

Algorithm I

```
//Detecting Critical Hosts
Input: Host_List, VMs_List
Output: List_VMs_Migrating_From_Hosts
For each selected_host in Host_List do
If (selected_host temperature
>Threshold_Temperature) then
Add selected_host to
Over_Threshold_Temp_Hosts_List
Else
If (selected_host utilization
>Threshold_Utilization) then
Add selected_host to
Over_Utilized_Hosts_List
End if
End if
Over_Hosts =
Over_Threshold_Temp_Hosts_List +
Over_Utilized_Hosts_List
List_VMs_Migrating_From_Hosts
End for
```

Algorithm II

```
//Detecting Migratable VMs
Input: List_VMs_Migrating_From_Hosts
Output: VMs_To_Migrate_List
For each selected_host in
List_VMs_Migrating_From_Hosts do
While true do
Vm get_Vm_To_Migrate (selected_host)
If (Vm == NULL) then
Break
End if
VMs_To_Migrate_List  add Vm
Selected_ host Deallocate_Vm
 If Not ((selected_host temperature
>Threshold_Temperature) && (selected_host
utilization >Threshold_Utilization)) then
Break
End if
End while
End for
```

Algorithm III

```
// Detecting Target Host.
Input: List_VMs_Migrating_From_Hosts,
VMs_To_Migrate_List
Output: Migration_Map
Set Migration_Map = NULL
VMs_To_Migrate_List
//Arrange_Decreasing_CPU_Consumption
For each VM in VMs_To_Migrate_List do
Allocated_Host = NULL
Set Min_Power = Max
For each selected_host not in
List_VMs_Migrating_From_Hosts do
If (selected_host has sufficient assets for vm)
then
```

Power calculate_Power (selected_host, VM)
End if
If (Power < Min_Power) then
Allocated_Host = selected_host
Min_Power = Power
End if
14: if (Allocated_Host <> NULL) then
Migrate vm to Allocated_Host
End if
End for
Update Migration_Map (VM, Allocated_Host)
End for

Table 3.1: Thermal Constants

Thermal Parameters	Range/Unit
CPU Temperature (Temp_Int)	320 Kelvin
Temperature Threshold (Temp_Thr)	345 Kelvin
Ambient temperature (Temp_Amb)	310 Kelvin

Table 3.1 depicts the used thermal coefficients for the experimentations.

Table 3.2: Simulated Hosts

Configuration	Host-1	Host-2
Mips	1920	2760
Cores	1	1
RAM (MB)	2048	4096
Bandwidth (GB/s)	1.5	1.5
Storage (TB)	2	2

Table 3.2 shows the setup constraints of the physical machines simulated for the analysis,

Table 3.3: VM Configuration

VM_No.	MIPS	RAM	Cores	Bandwidth (MB/s)	Storage (GB)
1	600	512	1	100	3
2	1200	1024	1	100	3
3	1500	1024	1	100	3
4	2000	768	1	100	3

Table 3.3 describes the assets of the demonstrated VM.

4. Results and Discussion

The processors' temperature is restricted using a temperature threshold. This means that in order to prevent hardware defects and for energy efficiency, some workload on the host must be transferred to other hosts if the temperature rises over the threshold.

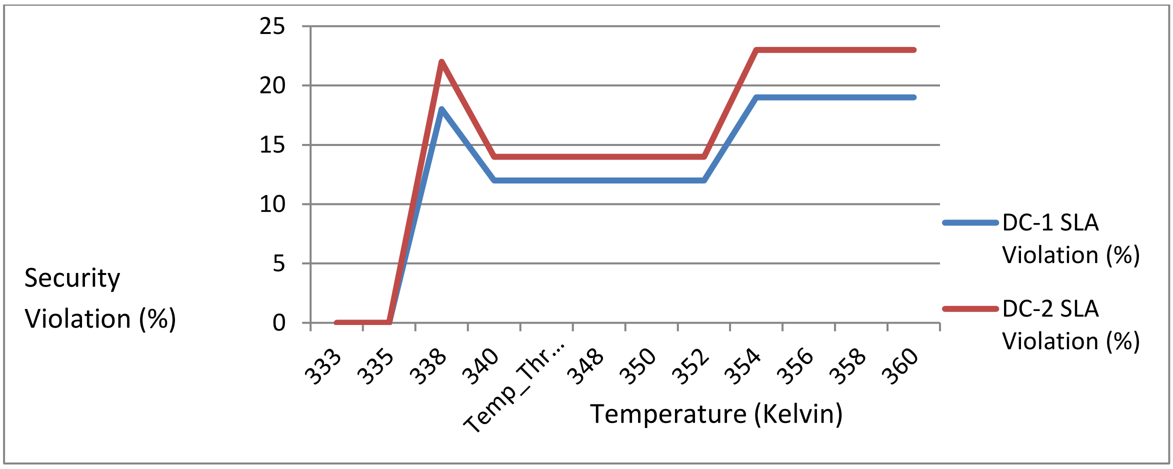

Figure 4.1: SLA Violation

As shown in figure 4.1 there is no SLA violation in starting due to zero migrations. Further SLA violation increases with number of VM migrations.

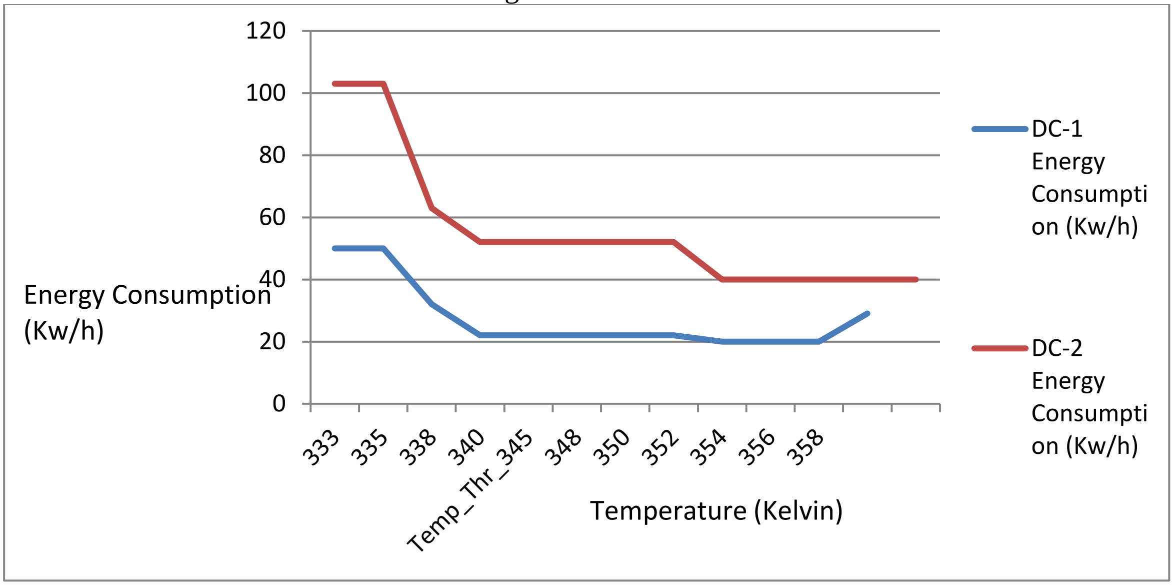

Figure 4.2: Consumed Energy

Figure 4.2 shows energy consumption of DC-1 and DC-2. As shown in figure after threshold temperature reduction in energy consumption is noticed.

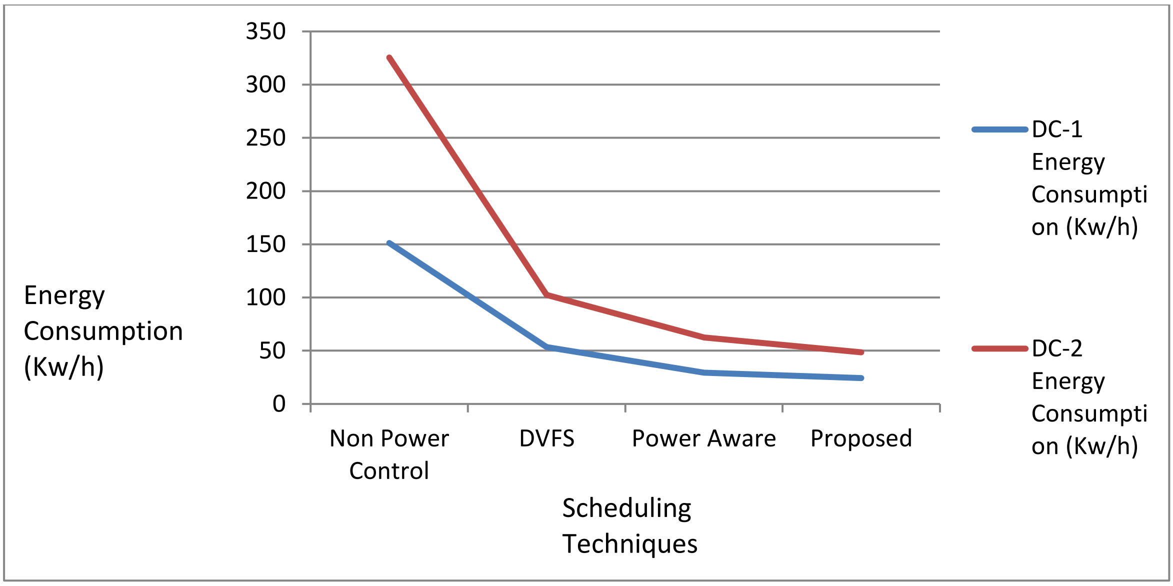

Figure 4.3: Proposed vs. Existing Energy Consumption

As shown in figure 4.3, energy consumption of the proposed scheme with two data centres is compared with three existing schedulers. As shown in figure proposed scheme achieves DC-1 24.3 Kwh and DC-2 48.5 Kwh consumed energy which is lowest compared with existing schedulers.

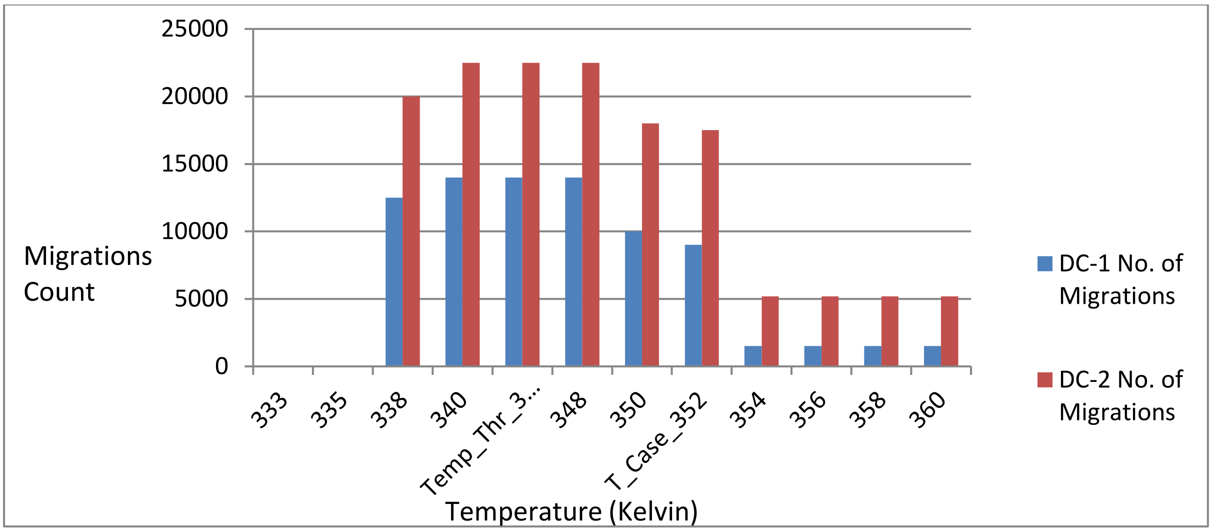

Figure 4.4: Proposed vs. Existing

As shown in figure initially proposed scheduler did not find target host to which VMs can be migrated. As we increase the threshold temperature it starts finding hosts having temperature less than threshold temperature and starts VMs migration. The proposed scheduler after the temperature 352 kelvin sets all the hosts in sleep mode to conserve the energy therefore low VMs migration.

Conclusion and Future Scope

Finding the critical hosts with either higher temperature or workloads is the suggested scheduler's first responsibility. The discovered hosts are then added to the list "List VMs Migrating From Hosts" and designated as potential candidates for VM migration. The scheduler goes through the list of essential hosts it established in the first phase once more to look for any VMs that might be present there. The actual candidates for migration are these virtual machines. The potential VMs are then arranged according to how much CPU they use. The VMs with the lowest CPU consumption are designated as having a higher migration priority. The third step is to locate a suitable target host to house the migration items on. The temperature on the target host and the workload requirements are observed. The observed host is chosen as the target host if the temperature is below the threshold value (Temperature threshold) and the VM's requirements are met.

References

[1] Haidar A, Jagode H, Vaccaro P, YarKhan A, Tomov S, Dongarra J. Investigating power capping toward energy-efficient scientific applications. Concurr Comput: Pract Exper 2019;31(6):e4485.

[2] Salinas-Hilburg JC, Zapater M, Moya JM, Ayala JL. Fast energy estimation framework for long-running applications. Future Gener Comput Syst 2021;115:20–33.

[3] Sîrbu A, Babaoglu O. Power consumption modeling and prediction in a hybrid CPU-GPU-MIC supercomputer. In: Euro-par 2016: Parallel processing. Cham: Springer International Publishing; 2016, p. 117–30.

[4] Sadjadi SM, Shimizu S, Figueroa J, Rangaswami R, Delgado J, Duran H, Collazo-Mojica XJ. A modeling approach for estimating execution time of long-running scientific applications. In: 2008 IEEE international symposium on parallel and distributed processing. 2008, p. 1–8.

[5] Wong A, Rexachs D, Luque E. Parallel application signature for performance analysis and prediction. IEEE Trans Parallel Distrib Syst 2015;26(7):2009–19.

[6] Yang LT, Ma X, Mueller F. Cross-platform performance prediction of parallel applications using partial execution. In: SC'05: Proceedings of the 2005
ACM/IEEE conference on supercomputing. IEEE; 2005, p. 40.

[7] Auweter A, Bode A, Brehm M, Brochard L, Hammer N, Huber H, Panda R, Thomas F, Wilde T. A case study of energy aware scheduling on supermuc. In: International supercomputing conference. Springer; 2014, p. 394–409.

[8] Mämmelä O, Majanen M, Basmadjian R, De Meer H, Giesler A, Homberg W. Energy-aware job scheduler for high-performance computing. Comput Sci Res Dev 2012;27(4):265–75.
[9] Goldman A, Ngoko Y. A MILP approach to schedule parallel independent tasks. In: 2008 international symposium on parallel and distributed computing. IEEE; 2008, p. 115–22.
[10] Chrétien S, Nicod J-M, Philippe L, Rehn-Sonigo V, Toch L. Job scheduling using successive linear programming approximations of a sparse model. In:European conference on parallel processing. Springer; 2012, p. 116–27.
[11] Lei H, Wang R, Zhang T, Liu Y, Zha Y. A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center. Comput OperRes 2016;75:103–17.
[12] Kashani M, Jahanshahi M. Using simulated annealing for task scheduling in distributed systems. In: International conference on computational intelligence, modelling and simulation. 2009, p. 265–9.
[13] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst 2012;28(5):755–68.
[14] Garefalakis P, Karanasos K, Pietzuch P, Suresh A, Rao S. Medea: scheduling of long running applications in shared production clusters. In: Proceedings of the thirteenth EuroSys conference. 2018; p. 1–13.
[15] Minakshi Kamboj, Sanjeev Rana, "Cloud Security and Energy Efficiency", Advances in Computational Sciences and Technology, Volume 10, Number 5 (2017) pp. 1245-1255
[16] Minakshi Kamboj, Dr. Sanjeev Rana, "Cloud Storage System Architecture and Energy Efficiency Parameters in Data Centres", International Journal on Future Revolution in Computer Science & Communication Engineering, Volume: 4 Issue: 3 (2018)