Author/s:
Ilager, S;Ramamohanarao, K;Buyya, R

Title:
ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation

Date:
2019-09-10

Citation:
Ilager, S., Ramamohanarao, K. & Buyya, R. (2019). ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation. Concurrency and Computation: Practice and Experience, 31 (17), https://doi.org/10.1002/cpe.5221.

Persistent Link:
https://hdl.handle.net/11343/285678

Received <8> <June>, <2018>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

RESEARCH ARTICLE

ETAS: Energy and Thermal-Aware Dynamic Virtual Machine Consolidation in Cloud Data Center with Proactive Hotspot Mitigation

Shashikant Ilager | Kotagiri Ramamohanarao | Rajkumar Buyya

[1] **Clou**d Computing and **D**istributed **S**ystems (CLOUDS) Laboratory, School of Computing and Information Systems
The University of Melbourne, Australia, Victoria, Australia

Correspondence
Shashikant Ilager, **Clou**d Computing and **D**istributed **S**ystems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne
Australia.
Email: silager@student.unimelb.edu.au

Abstract

Data centers consume an enormous amount of energy to meet the ever-increasing demand for cloud resources. Computing and Cooling are the two main subsystems that largely contribute to energy consumption in a data center. Dynamic Virtual Machine (VM) consolidation is a widely adopted technique to reduce the energy consumption of computing systems. However, aggressive consolidation leads to the creation of local hotspots that has adverse effects on energy consumption and reliability of the system. These issues can be addressed through efficient and thermal-aware consolidation methods. We propose an Energy and Thermal-Aware Scheduling (ETAS) algorithm that dynamically consolidates VMs to minimize the overall energy consumption while proactively preventing hotspots. ETAS is designed to address the trade-off between time and the cost savings and it can be tuned based on the requirement. We perform extensive experiments by using the real world traces with precise power and thermal models. The experimental results and empirical studies demonstrate that ETAS outperforms other state-of-the-art algorithms by reducing overall energy without any hotspot creation.

KEYWORDS:
Cloud computing, Data center cooling, Energy efficiency in a data center, Hotspots VM consolidation

1 | INTRODUCTION

Cloud computing is a massive paradigm shift from how the computing capabilities are acquired in past from traditional ownership model to current subscription model[1]. Cloud offers on-demand access to elastic resources as services with pay as you go model based on the actual usage of resources. Cloud data centers are the backbone infrastructure to cloud services. To adapt to the increasing demand for massive scale cloud services, data centers house thousands of servers to fulfill their computing needs. However, they are power hungry and consume a huge amount of energy to provide cloud services in a reliable manner. According to the USA energy department report[2], data centers in the USA itself consume about 2% (70 billion kWh) of the total energy production. Not only do data centers consume huge power; they significantly contribute to the greenhouse gas emissions resulting in high carbon footprints. To be precise, they generate 43 million tons of CO_2 per year and continues to grow at an annual rate of 11%[3]. If the necessary steps are taken, data center power consumption can be reduced from the predicted worst case of 8000 TWh to 1200 TWh by the year 2030[4]. Therefore, improving the energy efficiency of the cloud data center is quintessential for sustainable and cost-effective cloud computing.

A significant part of cloud data centers' energy consumption emanates from computing and cooling systems. In particular, the contribution of cooling system power is almost equal to the computing system [5]. In this context, a data center resource management system should holistically contemplate computing and cooling power together to achieve overall energy efficiency.

In pursuance of reducing the computing energy, workloads are consolidated on the fewest hosts as possible, and remaining hosts are shut down or turned to low power mode [6,7,8,9]. However, such aggressive consolidation leads to localized hotspots. The effect of hotspots is manifold. It has a catastrophic effect on the entire system by affecting the reliability of the data center [10]. In addition, the temperature beyond the threshold at host damages the silicon components of CPU leading to the failure of the host itself. Moreover, to prevent further complications, the cooling system is enforced to pass more cold air which further increases the cost of cooling. This entails for thermal management through optimal workload distribution to avoid the hotspots and simultaneously reduce the overall data center energy.

The temperature variations in a data center are caused by many factors. Firstly, the power consumed by a host is dissipated as heat to the ambient environment [11], this power consumption is directly proportional to the utilization level of resources. Secondly, the supplied cold air from Computer Room Air Condition (CRAC) itself carries certain temperature along with it which is known as cold air supply temperature. Finally, existing studies have shown that the inlet temperature of hosts exhibits the spatio-temporal phenomenon [12]. The dissipating heat from one host affects the temperature of other hosts, this heat recirculation within a data center happens due to the thermodynamic feature of hot air. The air that has passed through or over hosts does not completely reach the return-air plenum but instead remains in the space to pass over the hosts again. In this aspect, it is important to address this spatio-temporal aspect to optimize energy usage.

. On the other hand, estimating the temperature in a data center is a non-trivial problem. There are three approaches to predict the thermal status of the data center. First, CFD (Computational Fluid Dynamics) models [13], which are accurate in predictions; however, the inherent complexity in rendering makes it computationally expensive and thus infeasible for real-time online scheduling. The second approach is to use predictive models with techniques like machine learning [11], it largely depends on prediction models and the quality and quantity of the data. The last approach is analytical modeling [12], which is based on the thermodynamic features of heat and physical properties of a data center. It is computationally inexpensive and efficient compared to other two approaches. Therefore, it is requisite to use an analytical model to design online schedulers which are computationally inexpensive than others.

Dynamic VM Consolidation has proven to be a prominent approach for data center energy savings [7,14]. These consolidation algorithms are not aware of the physical layout and the location of the physical machine. Furthermore, due to the skewed temperature distribution in the data center, consolidating the workload on the fewest hosts may not always save the energy as it may increase the cooling cost and create hotspots [11]. However, there exists a restricted amount of work to address this aspect. Power and thermal-aware workload allocation for the heterogeneous data center are proposed in [15]. Similarly, dynamic voltage frequency scaling (DVFS) coupled spatio-temporal aware job scheduling is discussed in [16]. These solutions either cannot be applied directly to the virtualized cloud data centers or their solutions are application specific.

In this paper, we propose a new online scheduling algorithm for dynamic consolidation of VMs that uses analytical models for thermal status estimation. blueThe randomized online algorithms commonly perform better than deterministic algorithms designed for the same problems in real-time decision making systems [17]. Therefore, this algorithm for dynamic consolidation is based on Greedy Random Adaptive Search Procedure (GRASP) [18] meta-heuristic which is fast, adaptive and suitable for online decision systems. We analyze the proposed algorithm with the extensive simulation-based experiments using CloudSim [19] with real-time workload traces from PlanetLab systems. The proposed algorithm reduces the significant amount of overall energy consumption by preventing hotspot creation with the small amount of performance overhead in terms of Service Level Agreement (SLA) violations.

The key **contributions** of our work can be summarised as follows:

- We propose policies for efficient distribution of workloads (VMs) to optimize the computing and cooling energy holistically and proactively prevent the hotspots.
- We design an online scheduling algorithm based on GRASP meta-heuristic which is used for dynamic VM consolidation.
- We implement the proposed algorithm and validate its efficiency with extensive experiments using real workload traces through simulation and demonstrate its superiority by comparing to the several baseline algorithms.

The rest of the paper is organized as follows: We introduce the system model in Section 2 and provide problem formulation and an overview of our algorithm in Section 3. The experiments and results are discussed in Section 4, and related work is described in Section 5. Finally, we draw the conclusion and future directions in Section 6.

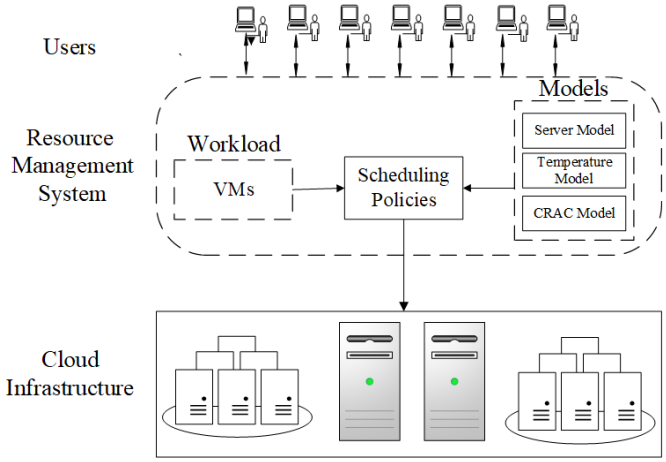


FIGURE 1 System Model

2 | SYSTEM MODEL

A model of a cloud computing system is shown in FIGURE 1 . It consists of three elements- infrastructure, resource management system (RMS) and users. An RMS receives the request for resources from users and it allocates requested resources from the data center infrastructure. We assume all the requests are submitted as virtual machines which are hosted on some physical machines. Table 1 shows the definition of all the symbols that are used in this section and the rest of the paper. A discussion on key elements of RMS, i.e., server model, temperature model, CRAC model, and workload model is presented below.

2.1 | Server Model

The data center consists of heterogeneous hosts with different processing capability. The power consumed by a host is predominantly determined by its utilization level, we adopt this power model[20] which has a linear relationship with the utilization of the CPU.

$$P_i(t) = \begin{cases} P_i^{idle} + \sum_{j=1}^{M_i} u(VM_{i,j}(t)) \times P_i^{dynamic} & M_i > 0 \\ 0 & M_i = 0 \end{cases} \quad (1)$$

The power ($P_i(t)$) consumption of a host is the summation of idle and dynamic power. In Eq. 1, P_i^{idle} is power consumption of a host in its idle state which is constant, and $P_i^{dynamic}$ is dynamic power consumption of host which has linear realtionship with CPU utilization. The $u(VM_{i,j})$ is utilization of j^{th} VM on $host_i$, and M_i is number of VMs running on $host_i$. We consider the host is active if its utilization is more than 0 and inactive if the utilization is 0.

2.2 | Temperature Model

The temperature at the host is dynamic and it depends on several factors such as its power consumption, CRAC settings and physical location of the host itself due to the heat recirculation effect[12].

The focus of this work is not to devise new metrics for these, instead, we use the existing approaches to model and incorporate it into our temperature model. The inlet temperature ($T_i^{in}(t)$) of a host is defined as a linear combination of supplied cold air temperature (T_{sup}) from CRAC and temperature increase due to heat circulation.

$$T_i^{in}(t) = T_{sup} + \sum_{k=1}^{N} d_{i,k} \times P_k(t). \quad (2)$$

Considering the heat recirculation effect exist within particular zones of data center based on its current physical layout, this recirculation effect can be quantified as a heat distribution matrix D where each entry $d_{i,k}$ in the matrix D indicates the factor by which inlet temperature of $host_i$ is affected by the $host_k$ and this factor is magnitude of power consumption ($P_k(t)$) of $host_k$. In the Eq. 2, $k \in 1, N$, is the number of hosts in

TABLE 1 Definition of Symbols

Symbol	Definition
N	Number of hosts
P_i	Power of $Host_i$
P_i^{idle}	Idle power of $Host_i$
$P_i^{dynamic}$	Dynamic power of $Host_i$
t	Time interval t
T	Total scheduling interval
U_{max}	Maximum CPU utilizayion threshhold of $host_i$
$u(VM_{i,j})$	utilization of VM_j on $host_i$
P_C	Computing system power
P_{CRAC}	Cooling system power
P_{total}	Total data center power
T_{sup}	Cold air supply temperature
$T_i^{in}(t)$	Inlet temperature at $host_i$ on time t
$T_i(t)$	Temperature at $host_i$ at time t
T_{red}	Maximum threshold of CPU temperature
R	Thermal resistance of Host
C	Heat capacity of host
$d_{i,k}$	Effect of heat recirculation to $host_k$ from i
α	Parameter to decide size of RCL in GRASP
ϵ	Iterations controller parameter in algorithm

recirculation zone. In abstract, it can be noted based on Eq. 2, though the CRAC passes similar cold air supply temperature (T_{sup}) across all the hosts in a data center, the inlet temperature varies at each host based on its physical location and heat recirculation effect.

The CPU temperature at the $host_i$ is dominated by dissipating heat by its CPU, hence, the temperature at time t can be defined by adopting widely used RC model[21] as follows:

$$T_i(t) = PR + T_i^{in} + (T_{initial} - PR - T_i^{in}) \times e^{-\frac{t}{RC}} \tag{3}$$

where P is the dynamic power of host, R and C are thermal resistance (k/w) and heat capacity (j/k) of the host respectively and $T_{initial}$ is the initial temperature of the CPU. Here, $T_i(t)$ refers to the dissipated CPU temperature (CPU temperature dissipated by $host_i$ at time t). Based on Eq. 3, it can be noted that, CPU temperature of host is not only governed by amount of power it is consuming(though it has a major effect and it is proportional to the CPU speed or workload level), it is also governed by hardware specific constants like R and C along with the inlet temperature (T_i^{in}) . Though we adopted the RC model to estimate the CPU temperature, our proposed work is independent of the temperature model and it can be applied to other models[12]. Eq. 3 captures the dynamic behavior of host temperature including heat recirculation effect.

2.3 | CRAC Model

The data center thermal management is done by Computer Room Air Condition (CRAC) system. In a modern data center, racks are arranged as cold aisle and hot aisle as shown in FIGURE 2 . The cold air flows through vented tiles from the bottom of the rack to the top of the rack in the cold aisle. The exhausted hot air is passed through hot aisle i.e, from the rear of the racks and it is collected through the ceiling and supplied back to CRAC[22]. Each data center consists of multiple CRAC units, $CRAC = \{CRAC_1, CRAC_2, ..., CRAC_n\}$. We consider CRACs are the only cooling facility available in the data center. The efficiency of such a cooling system is measured by the metric Coefficient of Performance (CoP). The CoP is a function of cold air supply temperature[11] (T_{sup}) and it is defined as the ratio of total power consumed by the computing system to the total power consumed by the cooling system to extract the dissipated heat.

$$CoP(T_{sup}) = \frac{P_C}{P_{CRAC}} \tag{4}$$

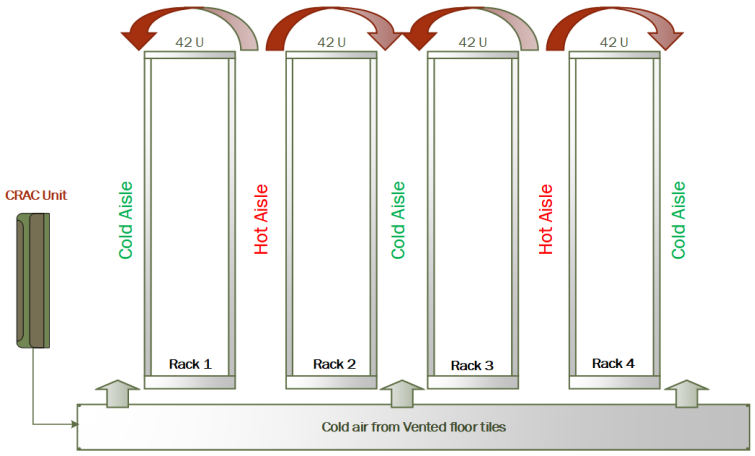


FIGURE 2 Data Center Rack Layout

In the Eq. 4, P_C and P_{CRAC} represents computing and cooling system power. The CoP of data center varies for various settings in a different data center. It depends on the physical layout and thermodynamic feature of the data center. It can be modeled using the regression techniques with multiple experiments using the different workloads and supply air temperature. In this work, we use the following model from HP lab [11] data center to estimate the CoP.

$$CoP(T_{sup}) = 0.0068T_{sup}^2 + 0.0008T_{sup} + 0.458 \tag{5}$$

Eq. 5 indicates that by increasing the value of T_{sup} we can increase the efficiency of the cooling system and reduce the cooling power.

2.4 | Workload Model

The requests from users are considered as tasks or cloudlets. Suppose n is the total number of cloudlets submitted, we consider the same number of VMs are required to execute these tasks which are represented as $VM = \{VM_1, VM_2, VM_3 ....., VM_n\}$. We model the workload in terms of the virtual machine, hence, our solution is independent of the workload type. We consider each VM executes a single cloudlet and the VM is terminated after the cloudlet is executed. Each cloudlet has CPU requirement R_{cpu}, memory requirement R_{mem} and task length l, hence each cloudlet has triplet attributes $\{R_{cpu}, R_{mem}, l\}$. Since we are addressing the dynamic consolidation, we submit all cloudlets at the beginning of the experiment. The VMs are consolidated dynamically based on certain scheduling policies at every interval.

3 | ENERGY AND THERMAL AWARE SCHEDULING

3.1 | Problem Formulation

The larger part of the data center energy consumption is contributed by the computing and cooling systems. The computing system consists of hosts and its energy consumption can be defined as follows:

$$P_C = \sum_{t=0}^{T} \sum_{i=1}^{N} x_j P_i \tag{6}$$

According to Eq. 6, computing system energy (P_C) is a summation of energy consumed by all hosts. The binary variable x_j holds value 1 if the host i is active at timestep t and 0 otherwise. It is imperative that computing systems energy is governed by the number of active hosts. Therefore, keeping an optimal number of active hosts at each scheduling interval is important.

The cooling system (CRAC) power consumption is defined as the ratio between the thermal load and CoP of the data center [11]. Considering the fact that energy consumed by a computing system is almost dissipated as heat to an ambient environment of the data center [12], thermal load can be represented as P_C. Accordingly, cooling system energy consumption can be defined as follows:

$$P_{CRAC} = \frac{ThermalLoad}{CoP(T_{sup})} = \frac{P_C}{CoP(T_{sup})} \tag{7}$$

Based on Eq. 7, it can be inferred that cooling system energy (P_{CRAC}) can be reduced either by increasing the CRAC cold air supply temperature (T_{sup}) or by decreasing the thermal load. Therefore, by using consolidation technique, we aim to decrease the thermal load of the data center and simultaneously avoid hotspots with a proactive approach for a given static cold air supply temperature (T_{sup}). Thus, the total energy consumption of the data center can be given as:

$$P_{total} = P_C + P_{CRAC} = (1 + \frac{1}{CoP(T_{sup})})P_C \tag{8}$$

The VM placement and consolidation algorithm must be aware of the orthogonal tradeoff between computing and cooling systems, where higher concentrated consolidation leads to hotspots and a highly sparsed distribution increases the energy consumption.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & P_{total} = \sum_{t=0}^{T}\sum_{i=1}^{N} x_j(1 + \frac{1}{CoP(T_{sup})})P_i \\ \text{subject to} \quad & u(h_i) \leq U_{max} \\ & T_i(t) < T_{red} \\ & \sum_{j=0}^{m} VM_{j,i}(R_{cpu}, R_{mem}) \leq h_i(R_{cpu}, R_{mem}) \\ & x_j \in \{0, 1\} \end{aligned} \tag{9}$$

The objective function in Eq. 9 takes care of the holistic minimization of energy. The constraints ensures the potential thermal violation and CPU threshold violation does not occur due to the added workload on the host. The constraints also satisfy the capacity constraints, if the host has enough resource (R_{cpu}, R_{mem}) for an accommodating VM, then the host is considered suitable for the VM placement. x_j is a binary variable whose value is 1 if the VM is allocated to $host_i$, and 0 otherwise. The above optimization function in Eq. 9 should be executed at each scheduling interval to decide the target host for the VMs. Considering that scheduling is an NP-hard problem and scale of cloud data center where a single data center hosts thousands of physical hosts, solving this optimization function in Eq. 9 is time-consuming and infeasible for real-time systems. Consequently, in the next section, we propose an online scheduling algorithm with reduced time complexity based on GRASP metaheuristic which finds the near-optimal solution in a reasonable amount of time.

3.2 | The Scheduling Algorithm

3.2.1 | Overview

In this subsection, we propose a scheduling algorithm based on GRASP metaheuristic. GRASP is simple to implement and easy to adapt based on the problem specific domain[18]. It is an iterative randomized optimization technique. Each iteration has two phases: 1) Greedy construction phase- where the solution list is constructed based on the greedy function by random sampling from the solution space. 2) Local search phase- a neighborhood search to find the current best solution from the previously constructed solution list. The iteration continues until certain stopping criteria is reached which can be chosen based on problem-specific constraints.

Its adaptive nature which provides an opportunity to dynamically update the greedy value of the objects and the simple probabilistic nature which selects the solution by random sampling is viable to achieve the near optimal solution. Moreover, its inherent capabilities like the flexibility to select the size of solution space, and to tune the stopping criteria is useful to adjust the amount of greediness and computational complexity.

Dynamic consolidation framework has mainly 3 steps[7].

1. Detect overloaded and underloaded hosts
2. Select the VMs to migrate from the hosts selected in step 1
3. Place the selected VMs into new target hosts

In this work, our primary focus will be on step 3, i,e, placement of VMs to new hosts based on thermal and energy status. These 3 optimization steps are applied to each scheduling interval to reduce the number of active hosts and keep remaining hosts in a low power mode or complete power off state to reduce the energy consumption.

Algorithm 1 ETAS: Energy and Thermal Aware Scheduling

Input: VMList, hostList

Output: Energy consumed, Number of hotspots, SLA violation percentage

```
Initialize T_red, α, ϵ
for t ← 0 to T do
    VMList ← getVMsFromOverAndUnderUtilizedHosts()
    for each vm in VMList do
        allocatedHost = ∅
        isSolutionNotDone ← true
        while isSolutionNotDone do
            SolutionList ← ConstructGreedySolution(VM, hostList)
            newHost ← LocalSearch(SolutionList)
            δ = allocatedHost.τ − newHost.τ
            if δ > ϵ then
                allocatedHost ← newHost
            else
                isSolutionNotDone ← false
            end if
        end while
        if allocatedHost == ∅ then
            allocatedHost = getNewHostFromInactiveHostList()
        end if
    end for
end for
```

Algorithm 2 Construct Greedy Solution

Input: VM, hostList

Output: SolutionList

```
SolutionList ← ∅
RCL ← makeRCLFromActiveHostList
for each s in RCL do
    if s is suitable for VM then
        s.τ ← (1 + 1/CoP(T_sup))P_i
    end if
    SolutionList ← ∪ s
end for
return SolutionList
```

3.2.2 | Algorithm

For the first two steps of dynamic consolidation, we use the following procedure. To detect overloaded hosts, we use the static CPU threshold (U_{max}) and the maximum threshold of CPU temperature (T_{red}) together as threshold parameters. To detect the underloaded hosts, we use the same approach used by Beloglazov et al.[7] where all the active hosts that are not overloaded are iterated and if all the VMs from such particular host can migrate to other active hosts then that host is considered as underloaded. For the second step of consolidation, we select VMs from overloaded hosts to migrate in an iterative manner until the host condition is not overloaded. The VM which has minimum migration time (mmt) is selected to reduce the migration bottleneck in the system (which has minimum RAM usage and takes less time to migrate with the available bandwidth).

We assume that, prior to optimization, the data center has reached to a steady state, i,e., all the requested VMs are placed into hosts and thermal status of the data center has reached to steady state. Algorithm 1 runs at the beginning of each scheduling interval (five minutes in this case), and identifies the VMs that are needed to be migrated (based on VM selection policies described above) from overloaded and underloaded hosts and migrates them to the destination hosts.

Algorithm 3 Local Search

Input: SolutionList

Output: Host with local optima

```
1: LocalOptimalHost ← ∅
2: for each s in SolutionList do
3:     if s.τ < LocalOptimalHost.τ then
4:         LocalOptimalHost = s
5:     end if
6: end for
7: return LocalOptimalHost
```

In the first step of the algorithm, all the essential parameters like T_{red} (redline temperature), α which decides the size of Restricted Candidate List (RCL) in GRASP technique and ϵ (expected amount of improvement over the previous iteration) are initialized, definitions of these are given in Table 1 . At each interval, all the VMs from over and underutilized hosts are identified (line 3) based on previously discussed policies, and for each VM to be migrated, the best possible host is allocated.

For each VM to be migrated from the migrate list, the process starts with initializing the allocated host to null initially (line 5). The line number 7-16 in Algorithm 1 shows the generic schema of the GRASP technique. At this stage, each iteration has two main steps: 1) constructing a feasible solution list from the search space, in this case, it is a list of possible hosts that can accommodate the current VM, 2) performing the local search to find a local best candidate, a sub-optimal solution. To reach the global best solution, at each iteration, the solution i,e., allocated host is updated based on the greedy value (τ) computed during the construction phase (line no 12).

If the difference (δ) between the current allocated host and newly allocated host's greedy value (τ) is greater than the predefined parameter ϵ, then the iteration process is continued. Otherwise, iteration is stopped and the current allocated host is returned as a result (line no 11-15). Here, ϵ acts as the parameter to decide the expected amount of improvement over the previous solution. If the new solution at current iteration does not give improvement greater than ϵ to the previous iteration, the process is terminated. If this process is failed to find the suitable host for VM then the new host is initiated from the inactive host's list in the data center resource pool (line no 17-18).

Algorithm 2 refers to the greedy construction solution phase. It takes VM and host list as input and returns the feasible solution list of hosts for a current VM upon each call to this procedure. The first step of this procedure is constructing the RCL which represents the finite solution search space to construct the solution list. The RCL is formed to limit the number of search in the solution space and thus reduce the time complexity. To that end, we completely exclude the inactive hosts and include a α percentage of hosts from the active number of hosts, this ensures the search space is vastly reduced. In addition, selecting the α fraction of active hosts into the RCL is done through random sampling which depicts the probabilistic part of GRASP. The cost for each host in the RCL represented as τ is calculated based on Eq. 9 for current time interval (i.e. t=1, i=1). It is important to note that, there can be a repetitive selection of the same sample in different iterations, however, with the sufficiently large number of iterations, it is assumed that the random sampling provides enough distinct distributions from the solution space.

The Algorithm 3 shows the local search applied to find the local best for each iteration. Based on the calculated greedy value τ, the best local candidate is returned as a solution.

This algorithm not only reduces the energy consumption, but it also circumvents the potential thermal violation along with satisfying capacity constraints like CPU, memory, and bandwidth, this is evidenced in line 4 of the Algorithm 2 that satisfies all the constraints that are listed in Eq. 9. Moreover, the parameters α and ϵ together act as tuning parameters to adjust the amount of greediness and decision time. If accuracy is most crucial in a system, these parameters can be set to a higher value which increases the time complexity to find the solution. Consequently, if finding a quick solution is crucial, these values can be set to lower which may compromise the quality of the solution.

4 | PERFORMANCE EVALUATION

We evaluated the feasibility and performance of our proposed algorithm ETAS and compared it to other baseline algorithms. We created a simulation environment using CloudSim[19] as it allows to model and simulates cloud computing environments that resemble real-world infrastructure elements. As the default CloudSim toolkit does not include thermal aspects of a data center, we extend the base classes to incorporate all the thermal parameters into it.

TABLE 2 Host and VM Configuration

Name	Core	CPU MIPS	RAM	Bandwidth
Intel Xeon_X5670	2	1860	4 GB	1 Gbit/s
Intel Xeon_X5675	2	2660	4 GB	1 Gbit/s
VM1 (Extra Large)	1	2500	870 MB	100 Mbit/s
VM2 (Large)	1	2000	1740 MB	100 Mbit/s
VM3 (Micro)	1	1000	1740 MB	100 Mbit/s
Vm4 (Nano)	1	500	613 MB	100 Mbit/s

TABLE 3 Host Power Consumption at Different Utilization level in Watts

Servers	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100
IBM x3550 M3 (Interl Xeon X5670 CPU)	66	107	120	131	143	156	173	191	211	229	247
IBM x3550 M3 (Intel Xeon X5675 CPU)	58.4	98	109	118	128	140	153	170	189	205	222

4.1 | Simulation Setup

In our setup, data center infrastructure comprises 1000 heterogeneous hosts. The capacity of these hosts are configured based on the IBM x3550 M3 machine with Intel Xeon X5670 and X5675 processor, configuration of these machines are shown in Table 2 . The reason for adopting CPUs with less number of cores is to demonstrate the efficiency of dynamic consolidation with a large number of VM migrations. Oppositely, hosts with a large number of cores can accommodate more number of VMs granting less opportunity for VMs migration. Nevertheless, the proposed policies do not affect this factor and also considering the fact that cloud data centers with massive heterogenous workloads induce enough triggers that generate a large number of VM migrations. The power usage of these systems is adopted from SPECpower benchmark [23], which provides power usage in watts for the respective machines at different CPU utilization level, the power usage of the two hosts that are used in this work can be seen in the Table 3 .

VMs are modeled according to the AWS[1] offerings as shown in Table 2 . The experiments are conducted on a desktop system with 64 bit Ubuntu operating system which is equipped with the Intel(R) Core(TM) i7-6700 processor, 16 GB of primary memory and 1 TB of secondary memory.

We assume that the hosts/servers are arranged in rack layout, and the racks are arranged in zones. Each zone consists of 10 racks that are laid in 5×2 rows, and each rack has 10 servers, this setup is inspired by the experimentally validated setup in [12]. We assume heat recirculation effects exist within each zone and is negligible across the zones, therefore we do not consider recirculation effect across zones. The heat distribution matrix that represents the recirculation effect within the zone is adopted based on the matrix that was used in [12].

We derive the workload from realistic traces from PlanetLab systems [24], this workload has several months of utilization history record of more than a thousand VMs that are geographically distributed. The data is recorded at an interval of 5 minutes. We use the one-day traces from this to generate the workloads for the VMs.

4.2 | Baseline Algorithms

In order to compare the performance and efficiency of our proposed algorithm, we consider the following baseline algorithms.

- **Random:** In this algorithm, all the VMs are placed on randomly selected hosts. This is a most intuitive method which does not consider either thermal or power status of the host.
- **Round Robin (RR):** In this algorithm, all the VMs are placed in a round robin fashion. This method tries to equally distribute the workloads among active hosts.
- **PABFD:** Power-aware Modified Best Fit Decreasing algorithm is proposed Beloglazov et al. [7]. This energy efficient policy only considers CPU utilization for consolidation while ignoring the thermal aspects.

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

- **GRANITE:** Greedy VM scheduling algorithm to minimize holistic energy in cloud data center proposed in [25]. This policy dynamically migrates VM to balance workload based on a certain temperature threshold.
- **TAS:** Thermal aware scheduling selects the lowest temperature host as the target host. The protective nature towards the thermal status of the host tries to avoid hotspot creation.

For all the aforementioned algorithms, similar policy for the initial two steps (over and underload host detection and VM selection) of dynamic consolidation is used as described in Section 3.2. However, they differ in VM placement strategy.

4.3 | Parameter Selection

The parameters that occur in different equations are set as follows. We set thermal resistance and the heat capacity in Eq. 3 as 0.34 K/w and 340 J/K respectively and the initial CPU temperature is set to 318 K [21]. According to the recommendation from American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) [26], the cold air supply temperature T_{sup} from CRAC is set to 25 °C.

The maximum allowable temperature of the host is between 85 and 100 °C [27, 16], we set it to 95 °C. It is important to note that, the temperature at the host is not only a factor of dissipated CPU temperature, it also includes the temperature that is associated with CRAC supply air (inlet temperature if we exclude recirculation effect). Excluding this supply air temperature (T_{sup} set to 25 °C) which is regarded as static, the CPU dynamic maximum threshold temperature (T_{red}) is 70 °C. This means, a host CPU is allowed to dissipate the maximum of 70 °C. In other words, we can say that the maximum temperature threshold is 95 °C, and after exclusion of the static part (T_{sup}) dynamic temperature threshold is 70 °C.

The CPU static utilization threshold (U_{max}) is set to 0.9. The hyperparameters α and ϵ are set to 0.4 and 10^{-1} respectively, the choice of selection and effects of these parameters are discussed in Section 4.5.

4.4 | Results and Analysis

The experiments were run for 5 times and the average results are reported. We ran the simulation for periods of 24 hours and scheduling algorithm was executed after each 5-minute interval to consolidate VMs dynamically. For the PABFD algorithm, there are many combinations based on different VM selection and allocation policies. We use local regression (LR), minimum migration time (MMT) as overload detection and VM selection policy respectively, which has shown to be the most efficient, we vary safety parameter of this algorithm from 1.0 to 1.4 with the increasing step value of 0.1 as described in their work [7].

4.4.1 | Metrics

In order to analyze the effectiveness of our proposed solution, we evaluate the results with the following metrics:

Energy: This metric indicates energy consumption of each approach in Kilowatts(kW).

SLA violation: This metric captures performance overhead caused due to dynamic consolidation. This overhead can be captured by the SLA violation metric [28] ($SLA_{violation}$) as shown in Eq. 12. Due to the oversubscription policy, hosts may reach its full utilization level (100%), in such case, the VMs on such host experiences the less performance level, this can be described as SLA violation Time per Active Host (SLA_{TAH}), and it is defined as in Eq. 10. Furthermore, the consolidation of VMs comes with performance overhead that has caused due to live VM migration [29], this Performance Degradation due to Migration (PDM) is defined as in Eq. 11.

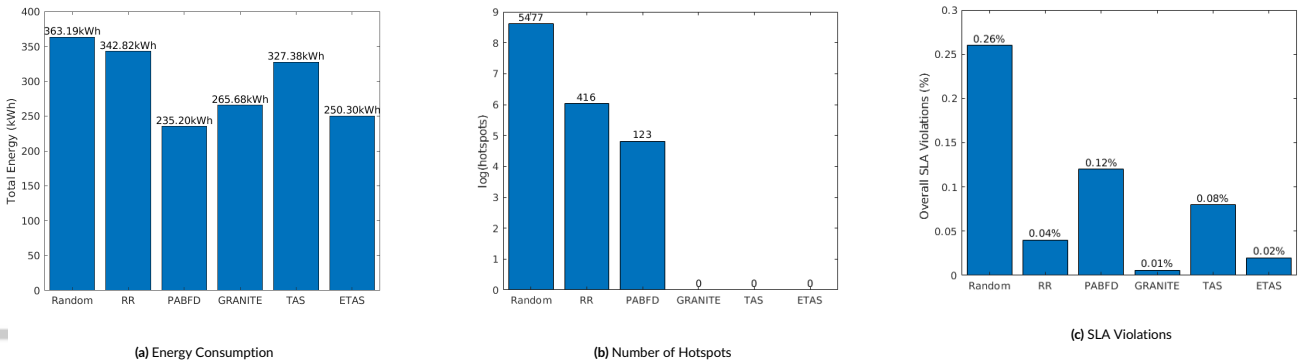
$$SLA_{TAH} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{max}}{T_{active}} \tag{10}$$

$$PDM = \frac{1}{M}\sum_{j=1}^{M}\frac{pdm_j}{C_{demand_j}} \tag{11}$$

$$SLA_{violation} = SLA_{TAH} \times PDM \tag{12}$$

Here,N is total number of hosts, T_{max} is amount of time $Host_i$ has expereienced 100% of utilization and T_{active} is total active time of $Host_i$. M is total number of VMs, pdm_j is performnace degradation due to live migration of VM_j, in our experiment, it is set to 10%, this value is similar to the one used by Beloglazov et al. [7] . The C_{demand_j} is total amount of CPU resource (MIPS) requested by VM_j in its lifetime. The overall SLA violation of cloud infrastructure ($SLA_{violation}$) can be captured by combining both of these SLA_{TAH} and PDM as shown in Eq. 12.

Hotspots: This metric indicates the number of hosts that have exceeded the redline temperature.



(a) Energy Consumption (b) Number of Hotspots (c) SLA Violations

FIGURE 3 Evaluation of Energy, Hotspots and SLA

Active hosts: This metric shows the number of active hosts present during the experimented period.

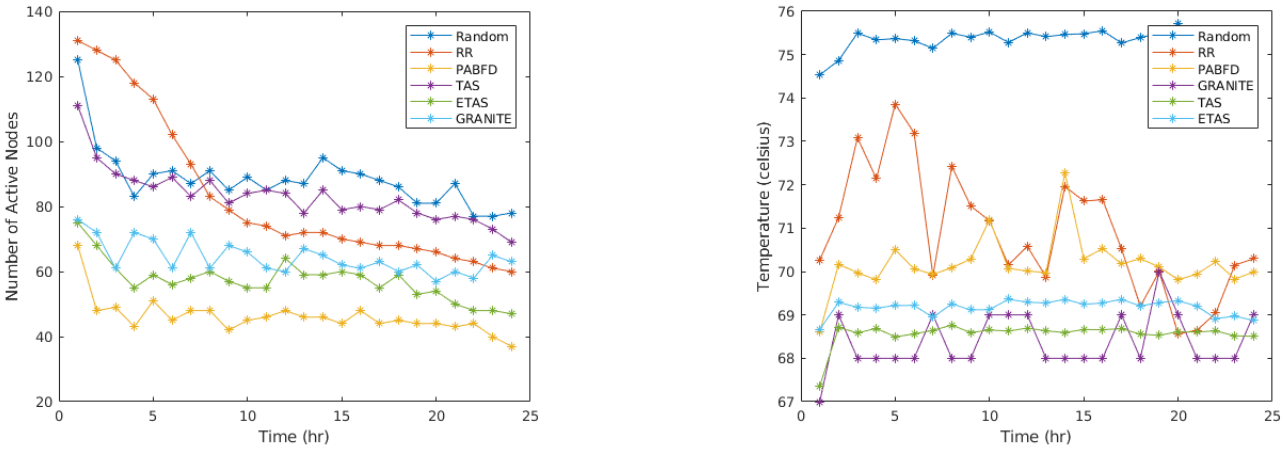
Peak Temperature: This metric indicates maximum temperature attained by any host during a scheduling interval. s

4.4.2 | Evaluating Energy, Hotspots, and SLA

The energy consumption from each of the policies is shown in FIGURE 3 a. The random policy has the highest energy usage of 363.19kWh. RR, PABFD, GRANITE, and TAS have 342.82kWh, 235.2kWh, 265. 68kWh and 327.38kWh, respectively, while ETAS has 250.30 kWh of energy consumption with 95% confidence interval (CI): (247.7, 252.5). In other words, ETAS consumes 31.1%, 27%, 5%, and 23.5% less than Random, RR, GRANITE, and TAS, respectively. Compared to PABFD, ETAS has a slight increase of 6.4%, PABFD consumes less energy due to the fact that it consolidates VMs on extremely less number of hosts compared to ETAS. This is due to PABFD is aggressive towards the consolidation and accounts for only optimizing computing energy while ignoring the potential thermal constraints which might have unfavorable effects on the system.

GRANITE though integrates both the thermal and energy aspect, it solely considers temperature as threshold parameter to balance the workload. Moreover, it sets the temperature threshold as the lowest temperature of a host among top 10% of high-temperature hosts in the data center and migrates VMs from those 10% hosts to balance the workload. This particular method is highly correlated with the workload type data center processes. For example, In the case where not all top 10% servers are exhibiting overload condition, it causes overhead due to excessive VM migrations, oppositely, if more than 10% servers are experiencing overload, GRANITE doesn't account this case too. In addition, identifying % of servers that are experiencing overload is unexplored in this approach (set to 10 % by default). Moreover, it is important to note that, default GRANITE algorithm balances the workload for only overloaded hosts, here we have applied the underload host management techniques similar to our approach with the GRANITE algorithm to balance the workload. In the process of balancing workload, there will be multiple underloaded hosts over a time, for which GRANITE doesn't have proposed policy. Applying GRANITE without underloaded hosts management resulted in a high amount of energy consumption compared to the other approaches.

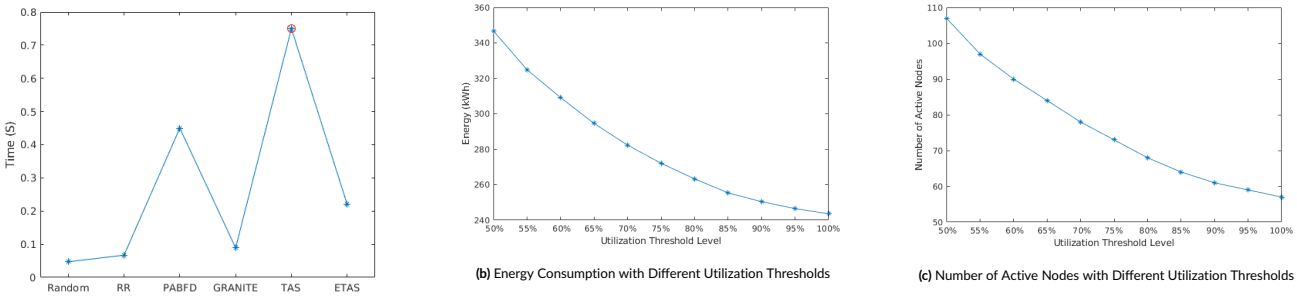
Though PABFD consumes slightly less energy compared to our ETAS, it creates a significant number of hotspots. This is evidenced in the bar chart FIGURE 3 b, where the Y axis in the figure has a logarithmic scale to respond to the skewness of large values. The consequence of randomness in Random policy has a high impact on both energy consumption and hotspots, this correlation can be observed in the result. Particularly, the Random policy has resulted in 5477 thermal violations in the experimented period. The fair policy distribution of RR performs better than Random policy and it accounts for 416 hotspots, nevertheless, its obliviousness towards thermal and energy parameters cause hotspots and more energy consumption. PABFD has resulted in a total number of 123 hotspots during the experimented period whereas ETAS resulted in 0 hotspots. It may seem like ETAS consumes slightly higher amount of energy (250.30 kWh) than PABFD (235.20 kWh), however, occurrence of 123 hotspots in case of PABFD has multiple effects, such as 1) there may be high potential of server failure due to overheating 2) whenever hotspots appear, in a reactive action to this, the data center administrators enforced to set the cooling temperature to much lower degree °C which further increases the energy consumption, this can be evidenced based on Eq.7 where cooling system energy is a function of cold air supply temperature (T_{sup}). Therefore, PABFD energy consumption will surpass when the reactive approach is enforced. This indicates that, in order to evade from hotspot creation, ETAS distributes VMs slightly spread out than PABFD and less than Random, RR, and TAS. In FIGURE 3 b, it can also be observed that GRANITE and TAS also do not account for any thermal violation due to their thermal-aware scheduling policies. The Conservative approach from TAS towards thermal status alone results in increased power consumption as its power agnostic nature spreads workloads too sparsely on more number of hosts. Consequently, thermal proactiveness and energy-aware placement of VMs from ETAS avoids hotspots and saves the energy.



(a) Average Number of Active Nodes per Hour

(b) Peak Temperature of a Host

FIGURE 4 Runtime Evaluation



(a) Running Time Analysis of Different Policies

(b) Energy Consumption with Different Utilization Thresholds

(c) Number of Active Nodes with Different Utilization Thresholds

FIGURE 5 Evaluation of Running Time and Effect of Utilization Level

FIGURE 3 c compares the percentage of SLA violation between all the policies. Random, PABFD, and TAS are least efficient while RR, GRANITE, and ETAS do not account for higher violation of SLA. Even though RR does not consider the SLA requirements while scheduling decision, its inherent characteristics that equally distributes the workload among hosts resulted in reduced SLA violations compared to Random, PABFD, and TAS. However, SLA violation percentage from ETAS (0.02%) may not seem like to completely outperform the RR (0.04%), but this SLA obtained by ETAS is while simultaneously optimizing total energy consumption which has an orthogonal tradeoff with SLA [7], whereas RR does not consider any aspect of energy optimization. Hence, ETAS is capable of minimizing SLA violations with better performance due to its performance and SLA aware scheduling policies.

4.4.3 | Runtime Evaluation

To completely understand the performance during runtime, we collect and report runtime data of following metrics: (1) number of active nodes; (2) maximum peak temperature attained by any of the hosts at each scheduling interval (5 min); (3) running time of each of the policies; (4) effect of different CPU threshold on energy and number of active nodes.

The number of active nodes in the experimented duration can be observed in FIGURE 4 a. For the sake of understanding and clear visibility of plots, we take the average value for each hour (12 intervals average represent a 1-hour data) for the results. PABFD results in less number of active nodes, while ETAS has a modest increase in a number of active nodes compared to PABFD. GRANITE has a small increase in a number of active nodes than ETAS, the reflection of this is evidenced by energy increase based on Eq. 8 and can be observed in FIGURE 3 a. The Random policy has the highest number of active nodes among all. TAS has a minimal increase in the number of active nodes while less in the case of RR. Moreover, a correlation between the number of active nodes, hotspots, and energy can be derived as the policies with less number of active nodes are prone

TABLE 4 Parameter and Values

Parameter	Values				
α	0.1	0.2	0.3	0.4	0.5
ϵ	10^{1}	10^{0}	10^{-1}	10^{-2}	10^{-3}

to the occurrence of hotspots. However, the Random policy is exceptional due to its arbitrary decisions. The difference in a number of the active nodes is not huge among all the policies due to the reason that in our consolidation process, we use same policy to detect over and underloaded hosts along with VM selection policy which contributes largely to this factor. The observed difference exists because of different VM placement decisions by each algorithm. Regardless, it can be inferred that ETAS has a modest increase in a number of active hosts compared to PABFD which is necessary to avoid high concentration of workload and prevent the potential hotspot creation.

FIGURE 4 b illustrates the comparison of the peak temperature of a host by each of the policies. The proposed ETAS never exceeds the redline temperature due to its thermal-aware placement of VMs and it operates near to redline which increases the resource utilization and reduces the cooling cost. TAS always operates at a much lower level temperature and PABFD almost operates around redline temperature (70 °C) and exceeds the red line in multiple instances. The peak temperature of a host from GRANITE policy is lowest among all as it considers temperature as threshold parameter alone and migrates VMs from high-temperature hosts to balance the workload. Though RR equally distributes workload, some of the hosts exceed the redline temperature due to its thermal unawareness. Note that, the represented results are not an average temperature of all the hosts, instead, the values represent the temperature of the hottest machine during each scheduling interval.

To analyze the computation overhead of different policies, we report the empirical values of the running time of each of these policies. This time indicates mean VM allocation time, which includes time taken by an algorithm to migrate a VM to a destination host, here major variant complexity is from deciding a target host for all VMs in this process. The FIGURE 5 a illustrates the running time of each of the policies. It can be observed that, due to the arbitrary selection of hosts by Random policy, it has the lowest running time compared to all. Since RR just need to select the next suitable host in a queue for allocation, it also accounts for less runtime. GRANITE has slightly more runtime compared to Random and RR. TAS and PABFD have high runtime overhead as they have to perform the maximum number iterations to find possible hosts for VM allocation. ETAS has minimal runtime compared to TAS and PABFD as it doesn't search complete solution space, more importantly, we can tune the runtime of ETAS with the energy trade-offs which is discussed in next section.

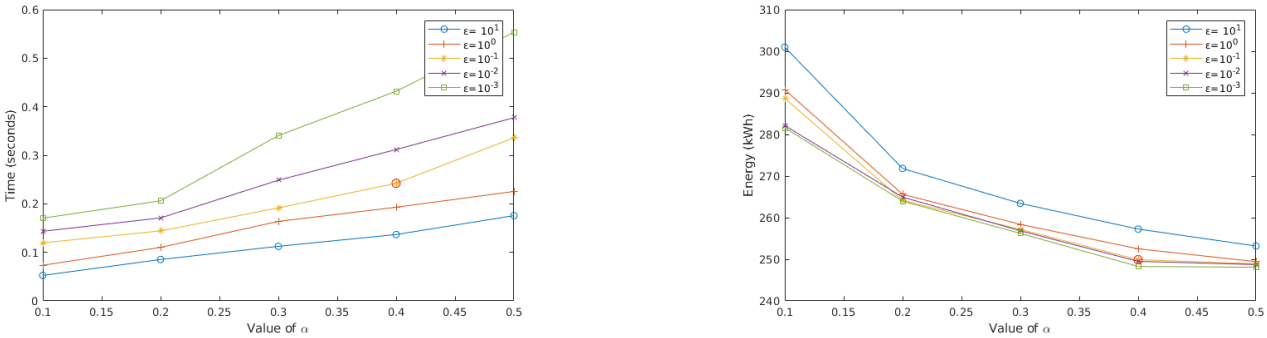
Performance of ETAS with different CPU utilization threshold values (U_{max}) can be observed in FIGURE 5 b and FIGURE 5 c which shows energy consumption and number of active nodes with different U_{max} values, respectively. For a lower utilization threshold, the energy consumption is higher as more number of hosts in the data center will be active to accommodate the given workloads. If we set the threshold to a lower value, the utilization of data center decreases and energy consumption increases. Hence, to achieve energy efficiency, the threshold should be high enough to utilize the data center resources efficiently. Furthermore, it can be observed that, after the threshold value of 0.9 (90%), the amount of reduction in energy consumption is less. Consequently, the extremely high value of U_{max} will result in a high number of QoS/ SLA violations.

In conclusion, our proposed energy and thermal-aware algorithm reduces the overall energy consumption of a data center while circumventing the hotspots by operating within the redline temperature. It also has minimal impact on the SLA violation. ETAS increases the global utilization of resources while ensuring the thermal constraints. In addition, the variant of GRASP heuristic is fast, lightweight and can be used in an online system.

4.5 | Sensitivity Analysis

The performance of our proposed algorithm is highly influenced by the parameters α and ϵ. To analyze this, we carried a sensitivity analysis and identified the best settings for these parameters. The values for α and ϵ that are considered as listed in Table 4 . These two parameters form 25 different combinations altogether. We evaluated the effect of these parameters on time and energy. The time represents a mean VM allocation time, i.e., decision time to find the target host for a VM to be migrated.

The effect of hyperparameters, α and ϵ on time and energy with all the 25 combinations can be observed in FIGURE 6 . The higher ϵ and lower α value results in higher energy consumption with the less time. However, after $\alpha = 0.4$, the energy saving is almost linear. Similarly, the lower ϵ yields better energy saving but it increases the time exponentially. The ideal setting for α and ϵ are 0.4 and 10^{-1}, respectively which can be observed from FIGURE 6 b and FIGURE 6 a. Therefore, these parameters can be tuned to manage the trade-off between time and energy.



(a) Effect on Time

(b) Effect on Energy

FIGURE 6 Sensitivity Analysis of Hyperparameters α and ϵ

5 | RELATED WORK

Energy management in a cloud data center has been the topic of interest for many researchers in recent years. The high energy consumption in data center incurs a huge operational cost and diminishes the profit margin of cloud service providers. The literature on energy management in the cloud data center itself is vast and we identify some of the relevant works in this section. Techniques like hardware optimization and Dynamic Voltage Frequency Scaling (DVFS) have been practically examined in the literature [30] to manage the energy of a host by adaptively varying the frequency of processor based on its utilization. However, such solutions are restricted to a single node level. For the data center level, techniques like VM Consolidation and load balancing [7 8] are extensively used to increase the resource utilization and reduce the energy consumption of computing nodes.

Utilization-based consolidation has been studied by Beloglazov et al. in [7], where they consolidate the VMs on the fewest server as possible based on current CPU utilization. The Modified Best Fit Decreasing (BFD) algorithm is used to place the VM into a target host. They also proposed different heuristics to detect overloaded and underloaded hosts and to select a set of VMs from those hosts and migrate to new hosts. Kansal et al. [31] have proposed Energy-aware VM consolidation algorithm using firefly meta-heuristic optimization technique. The target physical machine for VM to be migrated is selected based on a certain distance metric. The results have shown that 44% energy can be saved compared to other baseline algorithms. Li et al. [32] studied about VM migration and consolidation algorithms and proposed multi-resource energy-efficient models for the same. To avoid local optima and to reach global optima, particle swarm optimization based policies have been proposed. Verma et al. [8] proposed server consolidation using the workload analysis. They have stressed identifying the correlation between applications to consolidate on a server. These consolidation techniques are better at saving the computer system energy, however, they completely ignore the effect of consolidation on thermal status of a data center.

Thermal management is an important task for a data center resource management system. At first, Moore et al. identified the effect of workload on CPU temperature and heat recirculation effect in the data center. [11]. The authors have proposed workload placement strategies to reduce the heat recirculation effect. Similarly, Tang et al. investigated [12] thermal-aware task scheduling for homogeneous HPC data center. The scheduling policy is derived to minimize peak inlet temperature through task assignment(MPIT-TA). Moreover, they quantified the heat recirculation effect into a heat distribution matrix that was initially identified in [11]. In a similar way, DVFS coupled, thermal-aware HPC job scheduling has been investigated by Sun et al. [16] where the primary focus of the work is to reduce the makespan. Lee et al. [33] have proposed proactive thermal-aware resource management policies for virtualized HPC clouds. The authors have formulated heat imbalance model based on heat generation and heat extraction metrics. In addition, virtual machine allocation policies VMAP and VMAP+ are proposed to consolidate the workload.

All these solutions addresses either thermal-aware static job placement or mostly confined to HPC workloads. Such workload specific solutions cannot be directly applied to a cloud data centers where applications are deployed within virtualized resources and service providers usually don't have knowledge of the application characteristics running inside. The IaaS cloud services require application agnostic resource management techniques with a high abstraction of input data.

Thermal management specific to cloud data center is studied in many of the works in literature. Al-Qawasmeh et al. [15] presented power and thermal-aware workload allocation in the heterogeneous cloud. They have developed optimization techniques to assign the performance state of the CPU core at data center level. Li et al. [34] have investigated the failure and energy-aware scheduling. In this work, they have extracted failure model from the workload and developed failure and energy-aware static task assignment problem. However these approaches are static in their

TABLE 5 Related Work

Research Works	Thermal-aware	Heat Recirculation -aware	Consolidation	Online	Dynamic
Beloglazov et al.[7]	N	N	Y	Y	Y
Verma et al.[8]	N	N	Y	Y	Y
Ferreto et al.[36]	N	N	Y	Y	Y
Kansal et al.[31]	N	N	Y	Y	Y
Li et al.[32]	N	N	Y	Y	Y
Moore et al.[11]	Y	Y	N	Y	N
Qinghui et al.[12]	Y	Y	N	Y	Y
Sun et al.[16]	Y	Y	N	Y	N
Al-Qawasmeh et al.[15]	Y	Y	N	Y	N
Li et al.[34]	Y	Y	N	Y	N
Lee et al.[33]	Y	N	Y	Y	Y
Li et al.[25]	Y	Y	Y	Y	Y
Mhedheb et al.[35]	Y	N	Y	Y	Y
Our Work (ETAS)	Y	Y	Y	Y	Y

nature and do not consider runtime variation in utilization and consolidate accordingly. Moreover, cloud workloads typically run into few days to many years and they need to be dynamically consolidated at regular intervals which is the focus of this paper.

Mhedheb et al.[35] proposed heuristic algorithms with the goal of reducing energy by balancing load and temperature inside the cloud data center. The evaluation results through CloudSim has resulted that thermal-aware scheduling outperforms compared to power-aware only algorithms. The thermal models considered in this work are incomplete and excludes heat recirculation effect.

In a similar context, Ferreto et al.[36] designed consolidation algorithms with migration control based on linear programming and heuristic techniques. The results have evaluated based on a number of migrations and active physical machines as primary factors. The proposed consolidation tries to optimize only computing system of energy while completely ignoring thermal and cooling aspects.

In a similar way, Lee et al.[25] have proposed scheduling algorithm for holistic energy minimization of computing and cooling system in cloud data centers. The authors have proposed a greedy heuristic scheduling algorithm GRANITE that balances workload after fixed scheduling interval. However, the algorithm balances the workload only from the overloaded hosts that are decided based on a temperature threshold. The threshold is set by ranking hosts based on their temperature and selecting lowest temperature among top 10% of those hosts. The policy does not discuss managing underloaded hosts and setting an optimal percentage of servers that are to be considered as overloaded.

In a consolidation enabled cloud data centers, managing overloaded hosts for unknown nonstationary workloads poses challenging work for resource management systems. In this regard, Beloglazov et al.[37] proposed a solution to predict overloaded hosts and manage resources efficiently with explicitly set QoS. They have used a multi-size sliding window estimation model and Markov chain model to solve this problem. However, this work solely focuses on overload detection with regard to CPU resources. In a dynamic environment of cloud data center, the overload detection algorithm should integrate both thermal and computing resource aspects together.

The comparison of the related work can be found in Table 5 . Here, *Dynamic* means the ability to consolidate VMs in a regular interval after the initial placement based on optimization criteria.

The complexity of thermal behaviour and uncertain workload makes scheduling a complex problem. To address the dynamic consolidation with both thermal and energy awareness, in this work, we optimize the computing and cooling energy together with the aim of reducing the overall energy consumption of a data center while proactively mitigating the effect of hotspots.

6 | CONCLUSIONS AND FUTURE WORK

Cloud data centers are increasing in both number and size due to the rapid adoption of cloud computing in many spectrum of IT. Minimizing the energy consumption to increase the profit for cloud service providers without affecting the performance of user applications is a paramount need.

In this work, we proposed a dynamic consolidation framework for holistic management of cloud resources by optimizing both computing and cooling systems together. Through our proposed ETAS algorithm, we have managed the tradeoff between aggressive consolidation and sparse distribution of VMs which has an effect on energy and hotspots. Moreover, based on the system requirement, ETAS algorithm can be adjusted to manage the computational time and quality of the solution. The experiments are conducted with traces from real system and results have

demonstrated that the proposed ETAS algorithm saves 23.5% and 5% more energy as compared to the thermal-aware algorithms. Compared to the Energy-aware algorithm, ETAS is capable of avoiding hotspots with a modest increase in energy consumption.

In the future, we plan to extend our work for varying CRAC supply air temperature. We have assumed that CRAC cold air supply temperature is set to static, to vary the cold air supply temperature based on an overall status of a data center and save more energy, scheduling schemes need to be changed and we plan to do this in our upcoming work. We also intend to study the effects of using renewable energy and free air economizer usage in the data center resource management system.

References

1. Buyya Rajkumar, Yeo Chee Shin, Venugopal Srikumar, Broberg James, Brandic Ivona. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*. 2009;25(6):599–616.
2. Shehabi Arman, Smith Sarah, Sartor Dale, et al. United States data center energy usage report. 2016;.
3. Koomey Jonathan. Growth in data center electricity use 2005 to 2010. *A report by Analytical Press, completed at the request of The New York Times*. 2011;9.
4. Andrae Anders SG, Edler Tomas. On global electricity usage of communication technology: trends to 2030. *Challenges*. 2015;6(1):117–157.
5. Patel Chandrakant D, Bash Cullen E, Beitelmal Abdlmonem H. *Smart cooling of data centers*. US Patent 6,574,104; 2003.
6. Ahmad Raja Wasim, Gani Abdullah, Hamid Siti Hafizah Ab, Shiraz Muhammad, Yousafzai Abdullah, Xia Feng. A survey on virtual machine migration and server consolidation frameworks for cloud data centers.. *Journal of Network and Computer Applications*. 2015;52:11-25.
7. Beloglazov Anton, Abawajy Jemal, Buyya Rajkumar. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*. 2012;28(5):755–768.
8. Verma Akshat, Dasgupta Gargi, Nayak Tapan Kumar, De Pradipta, Kothari Ravi. Server workload analysis for power minimization using consolidation. In: :28–28USENIX Association; 2009.
9. Bobroff Norman, Kochut Andrzej, Beaty Kirk. Dynamic placement of virtual machines for managing sla violations. In: :119–128IEEE; 2007.
10. Zhou Rongliang, Wang Zhikui, Bash Cullen E, McReynolds Alan. Data center cooling management and analysis-a model based approach. In: :98–103IEEE; 2012.
11. Moore Justin D, Chase Jeffrey S, Ranganathan Parthasarathy, Sharma Ratnesh K. Making Scheduling "Cool": Temperature-Aware Workload Placement in Data Centers.. In: :61–75USENIX Association; 2005.
12. Tang Qinghui, Gupta Sandeep Kumar S, Varsamopoulos Georgios. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. *IEEE Transactions on Parallel and Distributed Systems*. 2008;19(11):1458–1472.
13. Choi Jeonghwan, Kim Youngjae, Sivasubramaniam Anand, Srebric Jelena, Wang Qian, Lee Joonwon. A CFD-based tool for studying temperature in rack-mounted servers. *IEEE Transaction on Computers*. 2008;57(8):1129–1142.
14. Dhiman Gaurav, Marchetti Giacomo, Rosing Tajana. vGreen: a system for energy efficient computing in virtualized environments. In: :243–248ACM; 2009.
15. Al-Qawasmeh Abdulla, Pasricha Sudeep, Maciejewski Anthony A., Siegel Howard Jay. Power and Thermal-Aware Workload Allocation in Heterogeneous Data Centers.. *IEEE Transaction on Computers*. 2015;64(2):477-491.
16. Sun Hongyang, Stolf Patricia, Pierson Jean-Marc. Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters. *Future Generation Computer Systems*. 2017;71:157–170.
17. Ben-David Shai, Borodin Allan, Karp Richard, Tardos Gabor, Wigderson Avi. On the power of randomization in on-line algorithms. *Algorithmica*. 1994;11(1):2–14.
18. Feo Thomas A., Resende Mauricio G. C.. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*. 1995;6(2):109-133.

19. Calheiros Rodrigo N, Ranjan Rajiv, Beloglazov Anton, De Rose César AF, Buyya Rajkumar. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*. 2011;41(1):23–50.

20. Zheng Kuangyu, Wang Xiaodong, Li Li, Wang Xiaorui. Joint power optimization of data center network and servers with correlation analysis. In: :2598–2606IEEE; 2014.

21. Zhang S, Chatha K S. Approximation Algorithm for the Temperature Aware Scheduling Problem. *Proceedings of International Conference on Computer-Aided Design*. 2007;:281–288.

22. Rasmussen Neil. Avoidable Mistakes that Compromise Cooling Performance in Data Centers and Network Rooms. *White paper*. 2003;49:2003–0.

23. SPEC . *Standard performance evaluation corporation*. 2008.

24. Park KyoungSoo, Pai Vivek S.. CoMon: a mostly-scalable monitoring system for PlanetLab.. *Operating Systems Review*. 2006;40(1):65-74.

25. Li Xiang, Garraghan Peter, JIANG Xiaohong, Wu Zhaohui, Xu Jie. Holistic Virtual Machine Scheduling in Cloud Datacenters towards Minimizing Total Energy. *IEEE Transactions on Parallel and Distributed Systems*. 2017;29(6):1–1.

26. ASHRAE . *American Society of Heating, Refrigerating and Air-Conditioning Engineers*. URL. http://tc0909.ashraetcs.org/; 2018.

27. Ebrahimi Khosrow, Jones Gerard F., Fleischer Amy S.. A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities. *Renewable and Sustainable Energy Reviews*. 2014;31:622–638.

28. Beloglazov Anton, Buyya Rajkumar. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurrency Computation Practice and Experience*. 2012;24(13):1397–1420.

29. Voorsluys William, Broberg James, Venugopal Srikumar, Buyya Rajkumar. Cost of virtual machine live migration in clouds: A performance evaluation. In: :254–265Springer; 2009.

30. Kim Wonyoung, Gupta Meeta S, Wei Gu-Yeon, Brooks David. System level analysis of fast, per-core DVFS using on-chip switching regulators. In: :123–134IEEE; 2008.

31. Kansal Nidhi Jain, Chana Inderveer. Energy-aware Virtual Machine Migration for Cloud Computing - A Firefly Optimization Approach. *Journal of Grid Computing*. 2016;14(2):327–345.

32. Li Hongjian, Zhu Guofeng, Cui Chengyuan, Tang Hong, Dou Yusheng, He Chen. Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing. *Computing*. 2016;98(3):303–317.

33. Lee Eun Kyung, Viswanathan Hariharasudhan, Pompili Dario. Proactive thermal-aware resource management in virtualized HPC cloud datacenters. *IEEE Transactions on Cloud Computing*. 2017;5(2):234–248.

34. Li Xiang, Jiang Xiaohong, Garraghan Peter, Wu Zhaohui. Holistic energy and failure aware workload scheduling in Cloud datacenters.. *Future Generation Computer Systems*. 2018;78:887-900.

35. Mhedheb Yousri, Jrad Foued, Tao Jie, Zhao Jiaqi, Kołodziej Joanna, Streit Achim. Load and thermal-aware VM scheduling on the cloud. In: :101–114Springer; 2013.

36. Ferreto Tiago C, Netto Marco AS, Calheiros Rodrigo N, De Rose César AF. Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*. 2011;27(8):1027–1034.

37. Beloglazov Anton, Buyya Rajkumar. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Transactions on Parallel and Distributed Systems*. 2013;24(7):1366–1379.

How to cite this article: Shashikant Ilager, Kotagiri Ramamohanarao, Rajkumar Buyya, (2018), ETAS: Energy and Thermal-Aware Dynamic Virtual Machine Consolidation in Cloud Data Center with Proactive Hotspot Mitigation, *Concurrency and Computation:Practice and Experience*, *xxxx;x:x–x*.