EAWA: Energy-Aware Workload Assignment in Data Centers

Seyed.Morteza Mirhoseini.Nejad
Department of Computing and Software
McMaster Universiy
Ontario, Canada
Email: mirhoses@mcmaster.ca

Ghada Badawy
Computing Infrastructure
Research Centre
McMaster University
Ontario, Canada
Email: badawyg@mcmaster.ca

Douglas G. Down
Department of Computing and Software
McMaster University
Ontario, Canada
Email: downd@mcmaster.ca

***Abstract*—One of the challenges that today's cloud computing infrastructures, and more specifically data centers, are struggling with is related to their energy consumption. Information technology (IT) equipment and cooling infrastructure are key parts of the total energy expenditure in a data center. A considerable amount of power is wasted due to workload management inefficiencies and the lack of coordination between cooling units and IT equipment. In this paper, server differences in terms of their cooling requirements and power consumption are taken into account for workload distribution. An optimal workload assignment problem that takes both server power consumption and thermal models into account is formulated. A simple low complexity algorithm is proposed. The algorithm not only assigns workload but it also adjusts the cooling unit set-point accordingly. Results show that the proposed algorithm can significantly reduce the total power consumed in a data center, in particular when compared to the uniform workload distribution algorithm.**

Keywords: Data Center Scheduling, Thermal Model, Workload Management, Power Efficiency, Cooling Efficiency

I. Introduction

Cloud computing infrastructures are currently drawing 3 to 5% of the world's electricity [1], [2]. These facilities are crucial in the shift from powerful personal computing devices. It has been estimated that cloud services demand will grow more rapidly in the near future [3]. These cloud services need to be run in data centers and large vendors such as Google, Microsoft, Apple and Amazon are rapidly deploying data centers throughout the world [4].

Such a shift to use cloud services and the resulting high demand for cloud applications require data centers with an increasing number of resources. There are many ongoing research projects on the efficient use of data center facilities, aiming to minimize power consumption.

A number of techniques have been considered to make data centers more energy efficient. Some IT devices support low power states to save energy if the quality of service (QoS) is not impacted. At the component level, dynamic voltage and frequency scaling (DVFS) is a method that provides different levels of power consumption and performance for processors [5], [6]. At the server level, several studies consider dynamic suspension of unneeded servers, called *server consolidation*. This saves both IT and cooling power due to the considerable length of low workload periods [7], [8], [9]. However, the trade-off between system performance and the number of On servers is a matter of debate [10], [11]. Additionally, the power efficiency of the cooling system itself is also a significant concern [12], [13].

One of the most important topics in this area is server workload management, for which there is a significant body of literature. The workload manager should distribute the offered load between servers. Inefficient distribution of workload might impose both extra cooling and computing power costs [14], [15]. Some studies that have addressed workload assignment and the resulting thermal effects are reviewed in the next section.

The work presented in this paper exploits the opportunities that arise from considering server differences. Two different servers may have different power consumption models and cooling requirements. Servers of different types/models might process a given workload with different levels of power consumption. Even for servers of the same type/model, there are some contributing factors that change thermal characteristics that in turn alter their cooling requirements.

Location, internal design, age, obstructions (at the front or back) alter the thermal characteristics of each server. Individual server characteristics result in different degrees of thermal resistance and consequently each server has its own required cooling power with respect to its thermal condition. For example, long-term operation changes the thermal characteristics of servers. If a server works continuously, dust and small particles can stick to the edges of vents, heat sinks, fins, etc. The physics of heat transfer can be used to show that the covered surface needs more power to remove the increased temperature beneath the cover [16]. Server location is also a contributing factor defining the thermal condition of a server, because different locations might have different airflows. Altered airflow of a server changes its thermal resistance [17]. Figure 1 shows an example of obstructions made by network and power cables at the back of a set of servers. Such obstructions clearly alter the air flow and unfortunately are typical features of data center environments.

We would like to quantify the opportunity (in terms of cost savings) of taking into account the individual characteristics of servers. In addition, we would like to address the fact

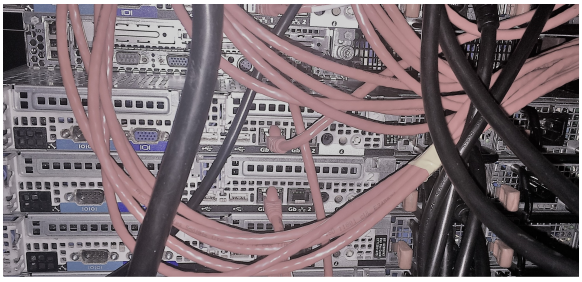

Fig. 1. Wires blocking at the back of servers

that cooling control and workload assignment are typically performed independently. If there is an estimate of the cost of assigning a job to each server, an optimal solution for workload assignment can be developed. In terms of cooling control, the current practice it to set the cooling set-point to the lowest possible value to consider the worst case scenario in a data center; this happens when all servers are heavily utilized. Such a requirement can be relaxed if an entity reports the current cooling requirement of servers to the cooling unit. The cooling unit can then increase its set-point to match the server cooling requirements, which will decrease cooling cost. Our method, energy aware workload assignment or in short EAWA, addresses these concerns. In particular, our main contributions are as follows:

- Introducing a thermal model to measure the maximum allowable inlet temperature of servers.
- Formalizing a power minimization problem to optimize the workload distribution.
- Preventing over-cooling.
- A low complexity solution for the optimization problem.

First of all, two models for the power and thermal condition of servers are introduced. Then a constrained power consumption minimization problem is proposed that uses the aforementioned models for workload distribution. A solution is presented for the optimization problem. An alternative to simplify the application of EAWA is discussed in the results.

II. Background

In [18] Mukherjee et al. developed a thermal-aware workload assignment algorithm using a model that they introduced for heat recirculation in a data center. It minimizes the power consumption with respect to given performance constraints. Tang et al. [12] also study the heat recirculation model in data centers. They tried to distribute the workload in a way that makes the temperature at the front of servers as uniform as possible.

Sharma et al. [19] presented a framework for thermal load balancing. They applied load monitoring to guide workload assignment decisions to smooth the thermal distribution in a data center. In this way, temperature is distributed uniformly and hot-spots are reduced.

In [13] Bash and Forman presented a method which they called *cool job assignment*. They suggest placing jobs in cool-efficient locations. To rank locations an index is defined. This index quantifies the response at the i^{th} rack inlet sensor to a step change in the supply temperature of the j^{th} cooling unit. The resulting algorithm for assigning workload is simple. Upon arrival of a batch of jobs, the longest job is assigned to the corresponding server of the highest ranked location and so on.

Abbasi et al. [20] presented a method to find an optimal set of *On* servers and optimal means of workload assignment; these are called *thermal aware server provisioning* (*TASP*) and *thermal aware workload assignment* (*TAWA*), respectively. The latter is related to our work. In *TAWA* they design an algorithm that distributes the workload among servers in a manner that minimizes the total power consumption in a data center. A key component of their approach is the quantification of heat recirculation effects via a heat recirculation matrix. TAWA tries to minimize this heat recirculation.

EAWA considers individual differences between servers. The differences originate from processing power and thermal requirements of servers. However, none of the previous works take these differences into account for assigning workload to servers. To the best of our knowledge, no previous study has looked at the workload assignment problem from this perspective.

III. Energy-aware workload Assignment

Inefficient workload distribution can cause extra heat production in data centers and cooling over-provisioning leads to a surplus in cool air generation. Both inefficiencies result in extra power consumption. We model servers from both perspectives of direct power consumption and thermal requirements.

The core of our idea is that workload should be assigned to servers that require less power to process the assigned workload and at the same time to those servers that impose low cooling demand. In other words, a given workload should be assigned to servers that are efficient in both processing and cooling power. In this way we not only have processing power savings from such a distribution, but additional savings can be realized from preventing over-cooling by adjusting the set-point temperature (T_{set}) of the cooling unit. Our data center model is equipped with n servers and a single cooling unit. An ideal cooling unit is assumed in this paper. Both cooling unit supply-air temperature (T_{sup}) and the temperature at the front of servers (inlet) (T_{in}) are assumed to be equal to the set-point temperature of the cooling unit. These assumptions are made for ease of presentation and the interests of space. Relaxing these assumptions is not difficult.

In this section, both thermal and power models are first presented. We then provide a means to distribute workload amongst servers in a way that minimizes total power consumption of the data center. The method determines an appropriate amount of workload to distribute to each server. In addition, the method sets the cooling set-point to the maximum possible temperature while ensuring that servers will not overheat. The notation used in this paper is listed in Table I.

TABLE I
NOTATIONS

Variable	Definition
n	Total number of servers
D	Offered Load or workload demand
u	Utilization vector of length n
u_i	CPU utilization of i^{th} server
u^{max}	Maximum allowed CPU utilization
$c_{i,j}$	i^{th} coefficient of power model of j^{th} server
$\beta_{i,j}$	i^{th} coefficient of thermal model of j^{th} server
$P_{server,i}$	i^{th} server power consumption (Watt)
P_{total}	Total power consumption (Watt)
P_{it}	Power consumption of IT units (Watt)
P_{cool}	Cooling infrastructure power (Watt)
$P_{cpu,i}$	CPU power consumption of i^{th} server (Watt)
$T_{cpu,i}$	CPU temperature of i^{th} server (°C)
T_{sup}	Supply air temperature of cooling unit (°C)
T_{cpu}^{red}	CPU red-line temperature (°C)
$T_{in,i}$	Inlet temperature of i^{th} server (°C)
$T_{in,i}^{req}$	Maximum allowable inlet temperature of i^{th} server (°C)
T_{in}^{req}	Vector of maximum allowable inlet temperatures of all servers
T_{set}	Set-Point temperature of cooling unit
CoP	Coefficient of performance
δ_u	Workload unit which can be assigned to a server

A. Power model

Ham, in [17], has developed a power consumption model for servers. He shows that the power consumption of a server ($P_{server,i}$) can be approximated by the CPU power ($P_{cpu,i}$) which is represented as a function of CPU utilization u_i and the CPU temperature ($T_{cpu,i}$) as shown in (1). The subscript i denotes the i^{th} server.

$$P_{server,i} \approx P_{cpu,i} = c_{1,i} + c_{2,i} \cdot u_i + c_{3,i} \cdot T_{cpu,i} + c_{4,i} \cdot T_{cpu,i}^2. \quad (1)$$

This model has one significant contributing factor, the CPU utilization, u_i. Although (1) shows that temperature affects CPU power consumption, it is negligible in comparison with CPU utilization. The authors in [17] simplified the model to (2) but there are also other works such as [12], [20], [21] that have also used (2). Zapater et al. [21] investigated the effect of the die or CPU temperature on the overall power consumption of servers in a thorough study, and their work confirms the imperceptibility of the impact of $T_{cpu,i}$. Therefore, we use the model (2) throughout this paper.

$$P_{server,i} = c_{1,i} + c_{2,i} \cdot u_i \quad (2)$$

Using this power model, we ran a series of experiments to find the coefficients $c_{1,i}$ and $c_{2,i}$ for several servers. We saw that the difference between two servers of the same model/type is negligible, but servers of different models or manufacturers have completely different coefficients $c_{1,i}$ and $c_{2,i}$.

B. Thermal model

The thermal model plays an important role in formulating the workload assignment problem. Our experiments on servers show that different thermal conditions considerably affect servers' CPU temperature. Using the model leverages differences in thermal conditions to assign workload. If servers have the same processing speed and power consumption, it makes more sense to send a given workload to servers that are located in favorable thermal conditions. In other words, workload should be sent to servers that are less expensive to cool.

1) Thermal model matters: To have a proper sense and understanding of the *thermal condition* and how it might affect cooling requirements of a server, an experiment was performed. We measured the cooling requirements of an HP ProLiant DL380 server under two different configurations. In *Configuration 2*, we partly blocked vents of the server. However in *Configuration 1*, the experiment was performed without the blockage of the server vents. *Configuration 1* has less restrictive airflow than *Configuration 2* and can be cooled down easily.

To determine thermal condition differences, we assigned the same workload to both configurations, installed them in the middle of a rack while other servers were working. The inlet temperature of the server was controlled by an in-row cooling unit, and all doors of the enclosure were closed during the experiment. With the same CPU utilization and the same $T_{in} = 26$°C, we found that T_{cpu} for *Configuration 1* was 65°C and T_{cpu} for *Configuration 2* was 72°C. Both temperatures are steady-state values. To compensate the increased T_{cpu}, in *Configuration 2*, and lower it back to 65°C, we had to decrease the set-point of the cooling unit to $T_{in} = 21$°C. Reducing T_{cpu} would be at the expense of increasing cooling power consumption.

Returning to workload management, the server in *Configuration 1* would be preferred to assign workload because its total power consumption -the sum of server and cooling power- is lower than the server in *Configuration 2*. This idea can be applied in general to select servers that are less expensive to cool if they are otherwise identical. We now introduce a thermal model to be used for workload distribution.

2) Thermal model: Server CPU temperature is critical and it should be kept below a certain threshold. The maximum allowable CPU temperature is called the *red-line temperature* (T_{cpu}^{red}). Our experiments show that the *CPU temperature of a server* ($T_{cpu,i}$) has two contributing factors, *CPU utilization* (u_i) and *inlet temperature* ($T_{in,i}$). We curve-fitted data measured from a series of experiments. The formula (3) provides the model, where u_i is a second order factor and $T_{in,i}$ is a first order factor. The interesting aspect of the obtained model is that all server types that we studied follow (3), however with different coefficients.

$$T_{cpu,i} = \beta_{1,i} + \beta_{2,i} \cdot u_i + \beta_{3,i} \cdot T_{in,i} + \beta_{4,i} \cdot u_i^2 + \beta_{5,i} \cdot u_i \cdot T_{in,i}. \quad (3)$$

Coefficients of (3) for the server in *Configuration 1* are: $\beta_1 = 13.4$, $\beta_2 = 10.3$, $\beta_3 = 1.5$, $\beta_4 = 26.5$ and $\beta_5 = -.25$. Using (3), we define the notion of *maximum allowable inlet temperature* ($T_{in,i}^{req}$) using T_{cpu}^{red}. For the sake of simplicity, T_{cpu}^{red} is considered to be equal for all servers. If $T_{cpu,i}$ is set to the red-line temperature (T_{cpu}^{red}), the maximum allowable inlet

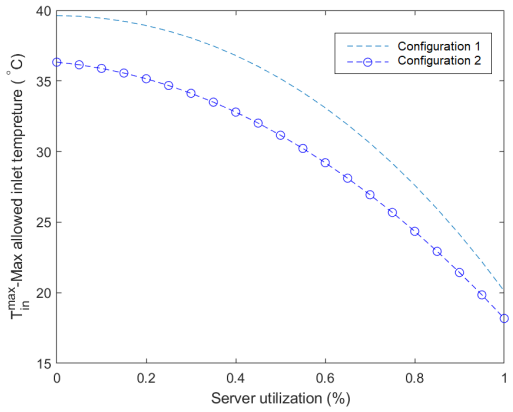

Fig. 2. Maximum required inlet temperature for two different configurations

temperature can be calculated with respect to u_i.

$$T_{in,i}^{req} = \frac{T_{cpu}^{red} - (\beta_{1,i} + \beta_{2,i} \cdot u_i + \beta_{4,i} \cdot u_i^2)}{\beta_{3,i} + \beta_{5,i} \cdot u_i} \tag{4}$$

The curves in Figure 2 show the maximum allowable inlet temperature for both settings as a function of CPU utilization. As expected, *Configuration 2* requires a lower inlet temperature. Moreover, the required inlet temperature decreases when CPU utilization increases. One thing that is important to note is that in what follows, different thermal models could easily be incorporated. The only requirement is that the maximum inlet temperature be expressible as a function of utilization.

C. Optimization problem

In our workload assignment problem, the aim is to assign the *offered load* or *demand, (D)* to a set of servers so that the total power consumption is minimized. The total power consumption (P_{total}) is the sum of server power P_{it} and cooling power P_{cool}, i.e.,

$$P_{total} = P_{it} + P_{cool}. \tag{5}$$

Obtaining P_{it} is straightforward; it is determined by adding the power consumptions ($P_{server,i}$) of all of the servers. Recalling the power model for each server allows P_{it} to be written as:

$$P_{it} = \sum_{i=1}^{n} (c_{1,i} + c_{2,i} \cdot u_i). \tag{6}$$

The *Coefficient of Performance* (CoP) of a cooling system is the ratio of useful cooling provided to work required [22]. Higher *CoP*s equate to lower cooling cost and *CoP* is usually greater than one. The *CoP* is given by:

$$CoP = \frac{P_{it}}{P_{cool}} \quad or \quad P_{cool} = \frac{P_{it}}{CoP}. \tag{7}$$

Using (5) and (7), the total power consumption is thus expressed as:

$$P_{total} = (1 + \frac{1}{CoP}) \cdot P_{it}. \tag{8}$$

CoP is typically a quadratic function of *supply air-temperature* (T_{sup}), see [23]. To be precise,

$$CoP = \gamma_1 + \gamma_2 \cdot T_{sup} + \gamma_3 \cdot T_{sup}^2. \tag{9}$$

As mentioned previously, we have assumed an ideal cooling unit. Hence $T_{set} = T_{sup}$ and both are equal to T_{in}. T_{set} should be assigned in a way that satisfies the temperature requirements of all servers. So, the set-point should be equal to the smallest required inlet temperature of all servers:

$$T_{sup} = T_{set} = min(T_{in}^{req}). \tag{10}$$

Combining (8) and (9),

$$P_{total} = (1 + \frac{1}{CoP(T_{sup})}) \cdot \sum_{i=1}^{n} (c_{0,i} + c_{1,i} \cdot u_i). \tag{11}$$

In the formula (11), the *power consumption* of the data center decreases with higher *inlet temperature*. This happens because a higher T_{in} yields a higher *CoP* in the denominator. We first formulate our optimization problem and we will then proceed to discuss it in more detail.

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & P_{total} \\ \text{subject to} \quad & \sum_{i=1}^{n} u_i = D, \\ & 0 \le u_i \le u^{max}, \ i = 1, \ldots, n \\ & T_{cpu,i} \le T_{cpu}^{red}, \ i = 1, \ldots, n \end{aligned}$$

In the minimization problem, P_{total} is given in (5) or equivalently in (11). The variable u is the utilization vector of all of the servers, where the i^{th} entry, u_i is the utilization of the i^{th} server. The value u^{max} is determined to meet performance constraints. For example, one can use queuing-theoretic techniques to determine u^{max} [24]. In this problem D is supposed to be a cap for the current status of the offered load to the system.

D. Solution to optimization problem

The optimization problem can be solved using sequential quadratic programming (SQP). However, we provide a heuristic algorithm which is attractive for implementation. We then compare the results of our algorithm with SQP to show the accuracy of our algorithm.

The heuristic solution to the problem is a greedy approach. Here, we assume that workload can be assigned in quanta δ_u and the offered load consists of an integral number of such quanta. Starting from a fully zero utilization vector, δ_u will successively be added to the currently preferred server. The sum of the assigned δ_us to a server should not exceed u^{max}, and the process is continued to the point that all of the offered workload is assigned.

In each step, the optimal server to receive δ_u is the server that increases the sum of P_{it} and P_{cool} by the smallest amount. This can be done using a linear search amongst the server set. At each step, δ_u is assigned to the server that was previously

selected, unless assigning δ_u to this server changes the minimum required inlet temperature. In other words, the algorithm tries to maximize the minimum required inlet temperature while taking into account server power consumption.

Algorithm 1 provides the details of our approach. The first **for** loop under **main**, at each iteration adds δ_u to the current load of the best server to accept this additional load. The i^{th} entry of u denotes the utilization of the i^{th} server; $optimalServer$ points to a server that executes the additional workload δ_u with the minimum total power cost. This optimal server ($optimalServer$) is returned by the $getOptimalServer$ function.

The function $getOptimalServer$ simply searches all possibilities to find the optimal server to accept δ_u. In other words, $getOptimalServer$ adds δ_u to the current load of each server and saves the power increase in another vector, δ_{pwr}; the index of the minimum value in δ_{pwr} is then returned. However, this can be written in a more efficient way. For example, $optimalServer$ can be the previously selected server, unless the given δ_u decreases $min(T_{in}^{req})$.

The $getOptimalServer$ function calls another function, $deltaPower$. $deltaPower$ returns the power increase with respect to the current load of all servers. It requires two inputs, the index of the server ($server_{index}$) and δ_u. The function adds δ_u to the current utilization of the server specified by $server_{index}$ and then returns the increased power consumption. $deltaPower$ calls another function $totalPower$ that returns the total power consumption of the data center, considering both cooling unit and server power consumption. This function uses (11) to calculate the total power. In the $totalPower$ function there is a vector *inletTemp* which stores the required inlet temperature of each server. The cooling unit should set its set-point to the minimum value stored in *inletTemp*.

The complexity of this solution is easily derived. The solution requires two main loops, the outer loop counts δ_us to assign them one by one to the optimal server, and the inner loop locates the optimal server. As each loop is of $O(n)$, the complexity of the algorithm is $O(n^2)$.

Using the proposed algorithm to solve the optimization problem is preferred. The algorithm gives almost the same results as compared to SQP, as shown in Table II. In fact, SQP is allowed to have finer-grained utilization values which fits a little better to the optimization cost function as shown in the last column of the table. While it may appear that the performance of the second solution is limited by the size of δ_u, we have varied the value of δ_u and found that as long as it is chosen to be reasonably small, the results are not very sensitive to its value. The greedy approach is the preferred solution for two main reasons. First, it exploits the problem structure and is easily implemented. It has simple steps with reasonable running time. Second, if we already have the solution for a given load D, the solution for an offered load $D+\delta_u$ is simply assigning the additional δ_u to the best server.

Result: Opt. WL assignment and set-point adjustment
void main (void):
global n=num-of-servers,D=offered-load; % Both are integers that need to be initialized
global u=zeros; % A vector of length n
global c_1, c_2; %Vectors of servers' power model coefficients (unique for each server)
global $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$; %Vectors of servers' thermal model coefficients (unique for each server)
global δ_u=delta-utilization; % The smallest fraction of the utilization that can be assigned to a server
for *index=0 :* δ_u *: D* **do**
  $optimalServer = getOptimalServer(\delta_u)$;
  $u(optimalServer) += \delta_u$;
end

Input: δ_u is the only input of this function
Output: Index of the optimal server to accept δ_u
integer getOptimalServer (float):
for *i=1 to n* **do**
  if *u(i)* $\leq u^{max}$ **then**
    $\delta_{pwr}(i) = deltaPower(i, \delta_u)$;
  end
  return index of the minimum element in δ_{pwr};
end

Input: server index ($Index_{server}$) & delta utilization (δ_u)
Output: Power increase of a server w.r.t. δ_u
float deltaPower (integer, float):
global u;
$power1 = totalPower(u)$;
$u(Index_{server}) = u(Index_{server}) + \delta_u$;
$power2 = totalPower(u)$; %Adds δ_u on top of the current utilization of a servers which is shown by $server_{index}$
$u(Index_{server}) = u(Index_{server}) - \delta_u$; %Restores the u vector
return $power2 - power1$;

Input: Vector of server utilizations (u)
Output: Total power consumption
float totalPower (vector of floats):
global c_1, c_2;
global $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$;
$inletTemp = T_{in}(u, \beta_i s)$; %From (4)
$CoP_{val} = CoP(min(inletTemp))$; %From (9)
P_{it}=$c_1 + c_2 . * u$; %Element-wise operation
$P_{it}^{total} = sum(P_{it})$;
return $P_{it}^{total} * (1 + 1/CoP_{val})$;

Algorithm 1: Optimization algorithm

TABLE II
WORKLOAD ASSIGNMENT OF FIRST FEW SERVERS AND THE CORRESPONDING POWER CONSUMPTION FOR BOTH SOLUTIONS

Server	01	02	03	04	···	Power (W)
SQP	0.4541	0.6144	0.5504	0.2625	···	2496.4
Greedy	0.4500	0.6100	0.5500	0.2600	···	2496.9

TABLE III
COEFFICIENT OF THE BASELINE MODELS PER EACH TYPE

Server	Type 1	Type 2	Type 3
c_1	110	99	103
c_2	119	102	132
β_1	13.4	12.1	14.5
β_2	10.3	11.1	9.3
β_3	1.5	1.3	1.6
β_4	26.5	23.3	25.8
β_5	-0.35	-0.23	-0.19

E. Results

The most reasonable way of demonstrating the performance of our method is comparing power consumption curves. Our method is compared with the *Thermal aware workload assignment* (TAWA) method which was presented first by Tang [12] and then improved by Abbasi [20]. Basically, TAWA minimizes hot-air recirculation within a data center. It sends a given load to a server that has less contribution to the recirculated hot air. In addition, we compared our method with the policy where workload is dispensed evenly between all servers, a policy that we call *Uniform Distribution*. Uniform workload assignment is a near optimal workload distribution policy in many studies (ignoring air-recirculation effects) see [20], [23], [25]; in addition, it is preferred in terms of response time performance [24].

To run the algorithm, we need a power consumption model and thermal model for each server. This is because obtaining the total power consumption (11) -the objective function of the minimization problem- requires c_is and β_is of all servers. Having them, we generated random values for the required coefficients using a normal random generator with the mean of the baseline model coefficients. Table III shows the coefficients for the baseline model under *type 1* to use as the means for the normal distribution. We used the variance of 20% of the mean for the normal random generator.

A system with 100 servers and u^{max}=0.8 is considered. So, the maximum offered load D cannot exceed 80. The result of comparing the power consumption of our method with the uniform workload assignment method is shown in Figure 3. The method not only saves a considerable amount of energy compared to *uniform workload assignment*, but it also leads to a simple means to control the cooling unit set-point (this is discussed in more detail later). The amount of power consumption reduction is notable. Significant savings come from reducing over-cooling.

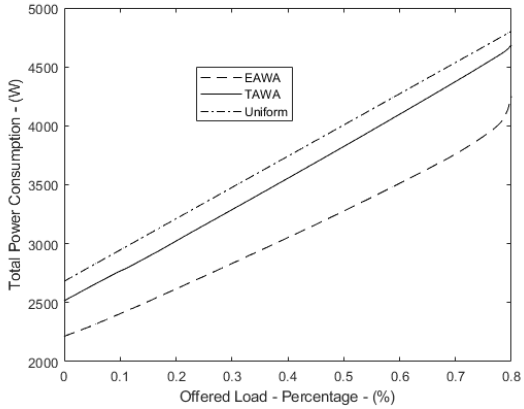

Fig. 3. Power consumption of data center for two workload assignment methods

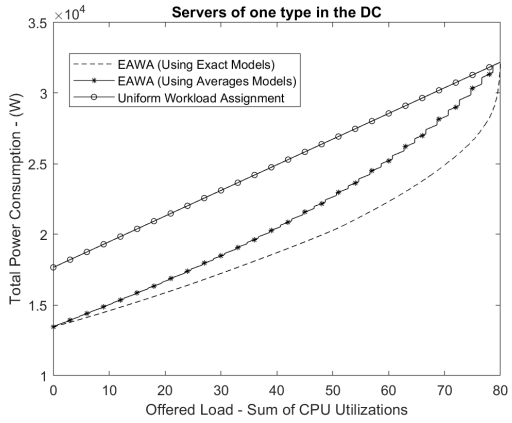

Fig. 4. Comparing power consumption of EAWA and uniform assignment

IV. DISCUSSION

In this section, we examine EAWA for different data center settings. A data center can be built up with servers of one type or servers of multiple types. Additionally, two methods for generating the energy models for servers are studied, the *exact model* and *average model*. The *exact model* considers each individual server model for workload assignment and the *average model* uses a baseline model as a representative for all servers of the same type. So, all coefficients (β_is and c_is) of servers of the same type are assumed to be equal in the *average model*. However, in the *exact model* all coefficients (β_is and c_is) of servers are specific to servers and they are drawn from a normal random generator around the type's baseline (Table III) as explained in Section III-E. In this section, the data center includes 100 servers and $u^{max} = 0.8$.

A. Servers of one type

First we consider a simple scenario for workload distribution to present the effectiveness of our method, a data center with only one type of servers. Figure 4 shows power consumption curves for three workload assignment methods based on the offered load. As expected, if EAWA uses the *exact model*, it

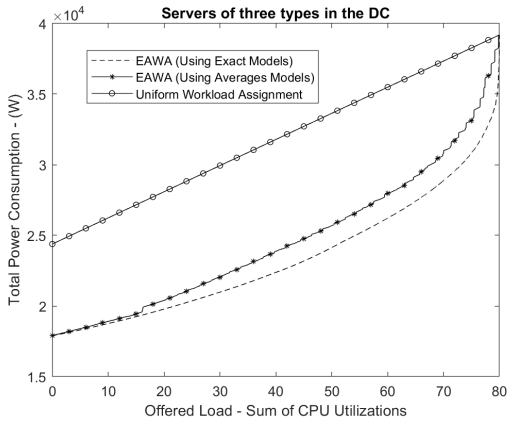

Fig. 5. Total power consumption of data center for three workload assignment methods

consumes less power compared to when it uses the *average model*. It can also be noted from the figure that when we have a light load using the *average model* works fine, however, as the load increases, workload distribution using the *exact model* becomes increasingly advantageous. A considerable decrease in power consumption is obtained using our method (EAWA) compared to uniform workload distribution.

B. Servers of different types

Usually data canters contain several types/models of servers. Each server has its own architecture and technology. If we consider a data center with different types of servers, we need to define a baseline model for each type. The means of generating *exact models* for each server of a specific type is explained at the beginning of this section. To use *average models*, it is required to assign the corresponding baseline models to servers.

Data centers with servers of multiple types have configurations for which our workload distribution method works very well and shows significant savings. Each server manufacturer employs certain hardware designs for different server types. Two different server types, even from the same manufacturer, might have different power profiles or thermal characteristics. When we consider different types of servers, there is more room for our method to exploit their differences and save power. We examined our method for the cases with multiple types of servers. For example, Figure 5 shows the result for three types of servers.

C. Average model versus exact model

In Figures 4 and 5 results for both *exact models* and *average models* are presented. Using each of these models for workload assignment purposes has pros and cons. The differences between these two models can be compared with respect to both performance and implementation concerns. All figures show that using the *exact models* outperforms using the *average models* in the power consumption aspect; however, the difference appears negligible in some scenarios. On the other hand, practically speaking, using *exact models* has more limitations than using *average models*. Exact models require scripts to run on each machine and each server is individually responsible for calculating its own model. This might hinder the acceptance of this method by some data center operators, due to security concerns, for example.

TABLE IV
DIFFERENCE BETWEEN POWER CONSUMPTION OF EXACT AND AVERAGE MODELS IN PERCENTAGES

Number of types	1	2	3	4
Percentage	12	9	6	5

The alternative is to use the *average model*. In comparison with the *exact model*, it uses more power; however, using this model is straightforward to implement. This is because there is no need for each server to compute its own model and an *average model* will be used for all. To obtain the *average model* it would be enough to test one server of the candidate model, and find the coefficients of (2) and (3) using a polynomial curve fitting method.

Table IV presents the power consumption reduction between these two models in percentages. It shows how much the *exact model* outperforms the *average model* with respect to the number of server types/models. The table shows that if the number of server types increases, the performance of these two models becomes very close to each other.

All in all, if the data center is homogeneous and there are no security or accessibility concerns, it does make sense to use the *exact model* for the proposed algorithm. On the other hand, using the *average model* is reasonable if there are a variety of servers or using the *exact model* is limited by some security or technical concerns.

D. Set-Point Adjustment and Cooling Unit Control

One of the contributions of this work is adjusting the set-point of the cooling unit, which we now discuss in more detail. Implementing a control mechanism for the cooling unit was not the initial purpose of this paper, in particular our observations here are limited to steady-state behavior, whereas a full control system design would necessarily consider transient behavior. Having said that, our experiments do yield insight on cooling control.

Plotting the set-point gives rise to an interesting observation. Figure 6 shows the set-point of the cooling unit versus the offered load. The key observation is that the curve begins as a flat line until it reaches a high offered load, at which point it drops. The reason for this is that servers are utilized up to the point that they require more cooling power. At this point, if there is any server that can serve the given workload without changing the set-point, it is preferred to send the workload to that server. A sudden decrease in the set-point happens when there is no server to accept the workload without reducing the set-point.

This suggests that a simple control mechanism may be appropriate. As mentioned, data centers usually experience

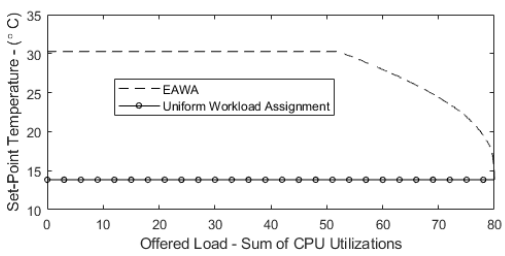

Fig. 6. Temperature of the cooling unit set-point for two methods

a low amount of workload during the majority of their life because of over-provisioning that exists during their design. On the other hand, Figure 6 shows that if the offered load is light, the set-point curve is flat. The set-point could then be set to the minimum required value (that occurs at maximum load) once the offered load rises above a threshold.

V. Conclusion

An energy-aware workload assignment method is proposed in this paper. We have leveraged the fact that the power requirements of servers can differ both in their direct power consumption and also their indirect cooling requirements. The work takes the power profiles and thermal models of servers into account to assign workloads to servers. Moreover, the cooling unit adjusts itself with the current cooling requirements of servers. An optimization problem is defined for the assignment of offered loads to servers. Two ways of modeling a server are proposed and compared; one of them is very easy to implement and provides near optimal results. The results of the paper show a way to achieve considerable amounts of savings in power consumption. It also offers the additional insight that cooling control with two set-points is near optimal.

Acknowledgment

This research was supported by Grant CRDPI 506142-16 from the Natural Science and Engineering Research Conceal of Canada.

References

[1] K. C. Armel, A. Gupta, G. Shrimali, and A. Albert, "Is disaggregation the holy grail of energy efficiency? the case of electricity," *Energy Policy*, vol. 52, pp. 213–234, 2013.

[2] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, M. Rosenberg *et al.*, "Us department of energy commercial reference building models of the national building stock," 2011.

[3] V. Cisco, "Cisco visual networking index: Forecast and methodology 2014–2019 white paper," *Cisco, Tech. Rep*, 2015.

[4] Y. Jadeja and K. Modi, "Cloud computing-concepts, architecture and challenges," in *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*. IEEE, 2012, pp. 877–880.

[5] E. Aldahari, "Dynamic voltage and frequency scaling enhanced task scheduling technologies toward green cloud computing," in *Applied Computing and Information Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science & Engineering (ACIT-CSII-BCD), 2016 4th Intl Conf on*. IEEE, 2016, pp. 20–25.

[6] R. Ge, X. Feng, and K. W. Cameron, "Performance-constrained distributed dvs scheduling for scientific applications on power-aware clusters," in *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*. IEEE, 2005, pp. 34–34.

[7] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, and T. F. Wenisch, "Power management of online data-intensive services," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*. IEEE, 2011, pp. 319–330.

[8] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

[9] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. Katz, "Napsac: Design and implementation of a power-proportional web cluster," *ACM SIGCOMM computer communication review*, vol. 41, no. 1, pp. 102–108, 2011.

[10] V. J. Maccio and D. G. Down, "Asymptotic performance of energy-aware multiserver queueing systems with setup times," Technical report, McMaster University, Tech. Rep., 2016.

[11] ——, "Exact analysis of energy-aware multiserver queueing systems with setup times," in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2016 IEEE 24th International Symposium on*. IEEE, 2016, pp. 11–20.

[12] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 11, pp. 1458–1472, 2008.

[13] C. Bash and G. Forman, "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center." in *USENIX Annual Technical Conference*, vol. 138, 2007, p. 140.

[14] S. V. Patankar, "Airflow and cooling in a data center," *Journal of Heat transfer*, vol. 132, no. 7, p. 073001, 2010.

[15] J. Cho, J. Yang, and W. Park, "Evaluation of air distribution system's airflow performance for cooling energy savings in high-density data centers," *Energy and Buildings*, vol. 68, pp. 270–279, 2014.

[16] T. Bergman, A. Lavine, F. Incropera, and D. DeWitt, "Fundamentals of heat and mass transfer, 2011," *USA: John Wiley & Sons. ISBN*, vol. 13, pp. 978–0470, 2015.

[17] S.-W. Ham, M.-H. Kim, B.-N. Choi, and J.-W. Jeong, "Simplified server model to simulate data center cooling energy consumption," *Energy and Buildings*, vol. 86, pp. 328–339, 2015.

[18] S. Mullender, Ed., *Distributed systems (2nd Ed.)*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1993.

[19] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase, "Balance of power: Dynamic thermal management for internet data centers," *IEEE Internet Computing*, vol. 9, no. 1, pp. 42–49, 2005.

[20] Z. Abbasi, G. Varsamopoulos, and S. K. Gupta, "Tacoma: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 9, no. 2, p. 11, 2012.

[21] M. Zapater, O. Tuncer, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun, "Leakage-aware cooling management for improving server energy efficiency," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 10, pp. 2764–2777, 2015.

[22] C. Patel, R. Sharma, C. Bash, and M. Beitelmal, "Energy flow in the information technology stack: coefficient of performance of the ensemble and its impact on the total cost of ownership, hp labs external technical report," HPL-2006-55, Tech. Rep., 2006.

[23] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, "Making scheduling" cool": Temperature-aware workload placement in data centers." in *USENIX annual technical conference, General Track*, 2005, pp. 61–75.

[24] M. Harchol-Balter, *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press, 2013.

[25] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services." in *NSDI*, vol. 8, 2008, pp. 337–350.