Thermal-aware Workload Distribution for Data Centers with Demand Variations

Somayye Rostami, Douglas G. Down, and George Karakostas
Department of Computing and Software
McMaster University
Hamilton, ON, Canada
Email: {rostas1,downd,karakos}@mcmaster.ca

***Abstract*—Thermal-aware workload distribution is a common approach in the literature for power consumption optimization in data centers. However, data centers also have other operational costs such as the cost of equipment maintenance and replacement. It has been shown that server reliability depends on frequency of their temperature variations, arising from workload transitions due to dynamic demands. In this work, we formulate a nonlinear optimization problem that considers the cost of workload transitions in addition to IT and cooling power consumption. To approximate the solution, we first linearize the problem; the result is a mixed integer linear programming problem. A modified heuristic is then proposed to approximate the solution of the linear problem. Finally, a Model Predictive Control (MPC) approach is integrated with the proposed heuristics for automatic workload reconfiguration when future demand is not known exactly, but predictions are available. Numerical results show that the proposed schemes are attractive in different settings.**

***Index Terms*—switching cost, model predictive control, thermal-aware workload distribution, data center**

I. Introduction

Apart from considerable energy consumption of data centers, they also have other operational costs such as the cost of equipment maintenance and replacement. However, when the workload distribution is changed due to dynamic demands, the cost of varying the workload on a server (which we will call switching costs) has seen little attention in the thermal-aware workload distribution literature. The varying workload leads to temperature variations that can impact the reliability of servers [1]. There are a few works that consider switching costs in the workload distribution policy. Most of the literature addresses thermal-aware workload distribution for a constant demand (steady- state). In [2], switching costs are considered but cooling power consumption is not considered. In [3], the costs of input fluctuations are considered in the thermal-aware workload distribution policy, where a transient thermal model is used. In this work, we formulate a thermal-aware workload distribution problem in discrete time that considers switching costs in addition to IT and cooling power consumption.

The proposed problem is a generalized form of the problem introduced in [4][5]. In [5] a general power optimization problem with nonlinear cooling power consumption and steady-state thermal model is proposed. Our contributions can be listed as follows:

- Generalization of the constant demand problem to a discrete-time, time-varying problem which also considers switching costs
- Generalization of the proposed approach in [5] for the resulting mixed integer programming problem and demonstrating its applicability for the proposed problem
- Integration of an MPC approach with demand predictions for the proposed heuristic
- Evaluation of the proposed schemes that suggest the potential for significant cost reductions, e.g. when compared to separating the problem into independent instances at each time step

II. System Model

We consider a discrete time demand model in which there are K time slots and the demand at time slot k is denoted by D_k, the number of required servers at time slot k. The system considered in this paper consists of n servers and one or more cooling facilities, where the control of the cooling facilities is performed through m parameters (setpoints, fan speeds, etc.). The decision variables are the cooling parameters and the server utilizations at time slot $k, k = 1, ..., K$, denoted by the vectors $v^{(k)}_{m\times1}$ and $\rho^{(k)}_{n\times1}$, respectively. As a power reduction scenario, two red-line temperatures are considered corresponding to idle or fully-utilized servers, so the server utilizations are 0 or 1. The servers are assumed to be identical. The cost function is the summation of cooling and IT power consumption along with the cost of workload migration and switching the servers between idle or fully-utilized (or on and off states in the case of server consolidation) in consecutive time slots. Thus, the problem that we wish to solve is problem (1), $F(v^{(k)})$ is the cooling power consumption corresponding to the cooling variable vector $v^{(k)}_{m\times1}$ at time slot k, $\rho^{(k)}_{n\times1}$ is the vector of workload distribution at time slot k, and $M(v^{(k)}, \rho^{(k)})$ is the function corresponding to the thermal model. Within each time slot, a steady-state thermal model is considered. The first constraint is a performance constraint with the target demand D_k, and the second constraint limits the inlet temperatures to be less than the corresponding red-line temperatures, T_{idle} and T_{busy} (according to [7], $T_{idle} > T_{busy}$). The cost of switching (and migration) per server for the kth time slot is denoted by w_k. The computing (IT) power consumption of server i in the kth time slot is denoted by $P(\rho_i^{(k)})$. The vectors of lower bounds and upper bounds for the cooling variables are V_{LB} and V_{UB}, respectively.

$$
\begin{array}{lll}
\min & \sum_{k=1}^{K} F(v^{(k)}) + \sum_{k=1}^{K} w_k \sum_{i=1}^{n} |\rho_i^{(k)} - \rho_i^{(k-1)}| + \sum_{k=1}^{K}\sum_{i=1}^{n} P(\rho_i^{(k)}) & \\
\text{s.t.} & \sum_{i=1}^{n} \rho_i^{(k)} \geq D_k & \forall k = 1,...,K \\
 & M(v^{(k)}, \rho^{(k)}) \leq T_{idle} 1_{n\times 1} - (T_{idle} - T_{busy})\rho^{(k)} & \forall k = 1,...,K \\
 & v^{(k)} \geq V_{LB} & \forall k = 1,...,K \\
 & v^{(k)} \leq V_{UB} & \forall k = 1,...,K \\
 & \rho_i^{(k)} \in \{0,1\} \quad \forall i = 1,...,n & \forall k = 1,...,K
\end{array}
\tag{1}
$$

The model we use for IT power consumption is $P(\rho_i^{(k)}) = c + d\rho_i^{(k)}$, where c and d are constants, but we assume that there is server consolidation, so that idle servers are turned off and $P(\rho_i^{(k)}) = 0$ when $\rho_i^{(k)} = 0$. Server consolidation requires an extra step of linearizing the IT power consumption.

We first linearize problem (1) and then generalize the heuristic proposed in [5] to approximate the solution of the linear problem. Linearizing the switching cost is straightforward and leads to introducing the new variables $s_{k,i}$. When the server utilizations are 0 or 1, linearizing the IT power consumption is also straightforward. In this case $P(\rho_i^{(k)}) = (c+d)\rho_i^{(k)}$. So, the integer linear programming problem is:

$$
\begin{array}{lll}
\min & \sum_{k=1}^{K}\sum_{j=1}^{m} v_j^{(k)} + \sum_{k=1}^{K} w_k \sum_{i=1}^{n} s_{k,i} + (c+d) \sum_{k=1}^{K}\sum_{i=1}^{n} \rho_i^{(k)} & \\
\text{s.t.} & \sum_{i=1}^{n} \rho_i^{(k)} \geq D_k & \\
 & -\sum_{j=1}^{m} A_{l,j} v_{k,j} + \sum_{i=1}^{n} B_{l,i}\rho_i^{(k)} + a\rho_l^{(k)} \leq b - E_l & \forall l = 1,...,n \\
 & s_{k,i} - \rho_i^{(k)} + \rho_i^{(k-1)} \geq 0 & \forall i = 1,...,n \\
 & s_{k,i} + \rho_i^{(k)} - \rho_i^{(k-1)} \geq 0 & \forall i = 1,...,n \\
 & v_j^{(k)} \geq V_{LB}^{(j)} & \forall j = 1,...,m \\
 & v_j^{(k)} \leq V_{UB}^{(j)} & \forall j = 1,...,m \\
 & \rho_i^{(k)} \in \{0,1\} & \forall i = 1,...,n
\end{array}
\tag{2}
$$

where all constraints are $\forall k = 1,...,K$, $a = T_{idle} - T_{busy} > 0$, $b = T_{idle}$, $A_{n\times m}$, $B_{n\times n}$ and $E_{n\times 1}$ are the cooling matrix, the heat-recirculation matrix and the constant part, respectively. In addition, $A_{i,j}, B_{i,j} \geq 0$ (nonnegative entries).

However, with the relaxation of server utilizations that is needed for the approximation algorithm, more work is needed to linearize the IT power consumption in problem (1). According to the IT power consumption model, in the case of consolidation there is a jump in $P(\rho_i^{(k)})$ when $\rho_i^{(k)} = 0$. We approximate $P(\rho_i^{(k)})$ with a piecewise linear function. For a small value ϵ, if $\rho_i^{(k)} \leq \epsilon$, then the IT power consumption is approximated as $P(\rho_i^{(k)}) = \frac{P(\epsilon)}{\epsilon}\rho_i^{(k)}$, and if $\rho_i^{(k)} > \epsilon$, then $P(\rho_i^{(k)}) = c\rho_i^{(k)} + d$. More details are provided in [6]. This problem is called relaxation of problem (2) in the next sections.

III. Approximation Algorithm

Our aim is to approximate the solution of problem (2) and use it for the original problem (1). We generalize the H2 heuristic in [5] to approximate the solution of problem (2). Let us denote the solution for the relaxation of problem (2) by $(v^{*(k)}, \rho^{*(k)}), \forall k = 1,...,K$. For our problem, the proposed heuristic, called DCVS (Dominant Cooling Variable with Switching cost), is similarly based on gradual rounding of the fractional server utilizations. However, instead of one problem, K problems are approximated. The values of $\hat{\rho}^{(k)}$ are computed consecutively, as the greatest correlation between demands will typically be between consecutive time slots. The problem for time slot k is problem (3), where $B' = B + I_{n\times n}$ (I is the identity matrix), there are R dominant cooling variables (the variables with the largest corresponding coefficient for at least one row of A), S_r is the set of servers with corresponding dominant cooling variable r, z_l is the corresponding coefficient of the cooling variable r (in the lth row of A) for the server $l \in S_r$, and $D_k^* = \lfloor \sum_{i=1}^{n} \rho_i^{*(k)} \rfloor$ ($\lfloor\rfloor$ is the floor function). The cost function for (3) is an approximation of the component of the cost function of problem (2) that is affected by the value of $\hat{\rho}^{(k)}$.

Similarly to H2, DCVS is greedy and includes three phases. The first (main) phase is modified to approximate the solution of problem (1) in terms of server utilizations. The algorithm is presented in [6]. The other two phases can also be found in [5].

IV. MPC Approach

We now consider the scenario where demand predictions are available. In problem (2), it may not be efficient or sufficiently precise to solve the problem for the whole time interval of size K. This is both due to the size of the problem and the fact that distant demand predictions may not be sufficiently accurate. One possibility to address these issues is using an MPC approach.

Algorithm 1 Calculation of $\hat{\rho}^{(s)}, \hat{v}^{(s)}$ using MPC approach with window size W

```
update the (predicted) demand values
solve problem (2) for $k = s, ..., s + W - 1$ and call the solution
    $(v'^{(k)}, \rho'^{(k)}), \forall k = s, ..., s + W - 1$
$\hat{\rho}^{(s)} = \rho'^{(s)}, \hat{v}^{(s)} = v'^{(s)}$
return $\hat{\rho}^{(s)}$ and $\hat{v}^{(s)}$
```

We use the MPC scheme which is described in Algorithm 1. Each time, a problem of size W is solved and the solution for the first time slot is kept and used as the initial workload distribution for the next round.

V. Evaluation

The system we use for evaluation comes from an experimental data center at McMaster University that is modeled in [8]. The data center has 25 servers located in 5 racks and two cooling facilities. Additional details are provided in [5]. According to [5], the matrices A, B and E in problem (2)

$$
\begin{aligned}
\min \quad & \sum_{r=1}^{R} \max_{l \in S_r} \frac{[\sum_{i=1}^{n} B'_{l,i}\rho_i^{(k)} - (\sum_{j=1}^{m} A_{l,j} v_j^{*(k)} + b - E_l)]^+}{z_l} + w_k \sum_{i=1}^{n} |\rho_i^{(k)} - \hat{\rho}_i{}^{(k-1)}| + w_{k+1} \sum_{i=1}^{n} |\rho_i^{*(k+1)} - \rho_i^{(k)}| \\
\text{s.t.} \quad & \sum_{i=1}^{n} \rho_i^{(k)} = D_k^* \\
& \rho_i^{(k)} \in \{0,1\} \qquad \forall i = 1, ..., n
\end{aligned} \tag{3}
$$

TABLE I: Performance of the Algorithms for Different Values of w

	OPTi	Sep	SR	DCVS
w	wrc	wrc	wrc	wrc
1	1.02	1.10	1.05	1.05
4	1.02	1.10	1.04	1.07
16	1.02	1.08	1.05	1.06
64	1.02	1.13	1.04	1.05
512	1.01	1.24	1.04	1.04
2048	1.01	1.87	1.16	1.02
8192	1.00	2.63	1.16	1.02

are known. We perform a (small) random perturbation of the matrices, each time that the algorithms are run.

We use simple rounding (SR) as the baseline algorithm. In simple rounding for each time slot k, the D_k^* largest values in $\rho^{*(k)}$ are rounded to one. We also solve the single time slot problem for each of the K time slots (without switching cost) using the *intlinprog* function in MATLAB and calculate the cost of the solution for the multiple time slot problem (with switching cost). This scheme is called Sep in the results. We present the average and the worst case ratios (avg and wrc columns in the results), where the solution is compared with the solution for the relaxation of problem (2).

The first results correspond to sensitivity to w. The number of intervals K is equal to 3. The pair of demands (D_1, D_3) covers all possible combinations, where the values for the demand are chosen from $D = \{1, 2, ..., 24\}$. For each combination, D_2 is randomly chosen from D. The results are reported in Table I, with an extra column OPTi corresponding to solving the problem using the *intlinprog* function in MATLAB. Although OPTi has the best performance, in [5] and [6] we showed that the running time does not scale well for larger problem sizes. The results show that the performance of DCVS is more resilient to changes in w and for larger values of w, SR has poor performance with respect to the worst case ratio. The results also show the performance of the Sep scheme is not as good as the others, specially for larger values of w.

The second results correspond to the integrated MPC approach. The number of time slots is $K = 50$ and the size of the planning window W is varied between 1 and 10. To calculate the solution over $K = 50$ time slots, the MPC approach uses a total of $K + W - 1$ demand values. So with $W = 10$, the length of the required demand sequence is $50 + 10 - 1 = 59$. We consider six scenarios for generation of demand sequences. There are three cases for the range of demand values, where the range for the next demand D_{k+1} is randomly chosen from $\{1, ..., 24\}$, $\{max(D_k - 5, 1), ..., min(24, D_k + 5)\}$, $\{max(D_k - 2, 1), ..., min(D_k + 2, 24)\}$, respectively, and p is the probability of changing the demand for the next time slot. We assume that we have a noisy version of demands coming from demand predictions. The value of noise for the time slot $s + k - 1, k = 2, ...W$, is randomly chosen from the interval $[-\eta \times k, \eta \times k]$, where η is the basic noise value. More details are provided in [6]. We consider three cases of $\eta = 0, \eta = 1, \eta = 3$, where $\eta = 0$ corresponds to the actual values without noise.

The results for $w = 1000$ are shown in Table II. The results for $\eta = 0$, show that for smaller window sizes (in particular $W = 1$), the performance is poor for all scenarios, with good performance achieved when $W = 4$. In general, the long term and short term solutions may be different. It can be inferred that as long as the window size is not too short, the MPC approach is beneficial as is shown for the case of $W = 3$ or $W = 4$.

TABLE II: Performance of the Integrated MPC Approach with DCVS

<table>
<tr><th colspan="7">Case 1 with $p = 0.2$</th></tr>
<tr><th></th><th colspan="2">η=0</th><th colspan="2">η=1</th><th colspan="2">η=3</th></tr>
<tr><th>W</th><th>avg</th><th>wrc</th><th>avg</th><th>wrc</th><th>avg</th><th>wrc</th></tr>
<tr><td>1</td><td>1.27</td><td>1.60</td><td>1.27</td><td>1.60</td><td>1.27</td><td>1.60</td></tr>
<tr><td>3</td><td>1.00</td><td>1.02</td><td>1.01</td><td>1.02</td><td>1.02</td><td>1.07</td></tr>
<tr><td>4</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.02</td><td>1.03</td><td>1.07</td></tr>
<tr><td>5</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.03</td><td>1.04</td><td>1.11</td></tr>
<tr><td>10</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.03</td><td>1.04</td><td>1.11</td></tr>
<tr><td colspan="7">Case 1 with $p = 0.8$</td></tr>
<tr><td>1</td><td>1.10</td><td>1.19</td><td>1.10</td><td>1.19</td><td>1.10</td><td>1.19</td></tr>
<tr><td>3</td><td>1.04</td><td>1.07</td><td>1.05</td><td>1.09</td><td>1.08</td><td>1.14</td></tr>
<tr><td>4</td><td>1.02</td><td>1.04</td><td>1.04</td><td>1.06</td><td>1.06</td><td>1.09</td></tr>
<tr><td>5</td><td>1.02</td><td>1.03</td><td>1.03</td><td>1.06</td><td>1.05</td><td>1.08</td></tr>
<tr><td>10</td><td>1.02</td><td>1.03</td><td>1.03</td><td>1.04</td><td>1.05</td><td>1.08</td></tr>
<tr><td colspan="7">Case 2 with $p = 0.2$</td></tr>
<tr><td>1</td><td>1.51</td><td>2.45</td><td>1.51</td><td>2.45</td><td>1.51</td><td>2.45</td></tr>
<tr><td>3</td><td>1.01</td><td>1.02</td><td>1.01</td><td>1.02</td><td>1.02</td><td>1.05</td></tr>
<tr><td>4</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.02</td><td>1.02</td><td>1.11</td></tr>
<tr><td>5</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.03</td><td>1.03</td><td>1.12</td></tr>
<tr><td>10</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.04</td><td>1.04</td><td>1.26</td></tr>
<tr><td colspan="7">Case 2 with $p = 0.8$</td></tr>
<tr><td>1</td><td>1.25</td><td>1.69</td><td>1.25</td><td>1.69</td><td>1.25</td><td>1.69</td></tr>
<tr><td>3</td><td>1.02</td><td>1.04</td><td>1.03</td><td>1.06</td><td>1.05</td><td>1.09</td></tr>
<tr><td>4</td><td>1.02</td><td>1.03</td><td>1.03</td><td>1.04</td><td>1.04</td><td>1.07</td></tr>
<tr><td>5</td><td>1.02</td><td>1.04</td><td>1.03</td><td>1.05</td><td>1.04</td><td>1.08</td></tr>
<tr><td>10</td><td>1.02</td><td>1.04</td><td>1.03</td><td>1.05</td><td>1.04</td><td>1.07</td></tr>
<tr><td colspan="7">Case 3 with $p = 0.2$</td></tr>
<tr><td>1</td><td>1.41</td><td>1.98</td><td>1.41</td><td>1.98</td><td>1.41</td><td>1.98</td></tr>
<tr><td>3</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.02</td><td>1.01</td><td>1.04</td></tr>
<tr><td>4</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.03</td><td>1.02</td><td>1.07</td></tr>
<tr><td>5</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.02</td><td>1.02</td><td>1.08</td></tr>
<tr><td>10</td><td>1.00</td><td>1.01</td><td>1.01</td><td>1.02</td><td>1.03</td><td>1.12</td></tr>
<tr><td colspan="7">Case 3 with $p = 0.8$</td></tr>
<tr><td>1</td><td>1.31</td><td>1.52</td><td>1.31</td><td>1.52</td><td>1.31</td><td>1.52</td></tr>
<tr><td>3</td><td>1.00</td><td>1.02</td><td>1.03</td><td>1.06</td><td>1.04</td><td>1.08</td></tr>
<tr><td>4</td><td>1.01</td><td>1.03</td><td>1.03</td><td>1.06</td><td>1.04</td><td>1.08</td></tr>
<tr><td>5</td><td>1.01</td><td>1.03</td><td>1.02</td><td>1.05</td><td>1.04</td><td>1.09</td></tr>
<tr><td>10</td><td>1.01</td><td>1.03</td><td>1.03</td><td>1.05</td><td>1.04</td><td>1.06</td></tr>
</table>

References

[1] N. EI-Sayed, I. A. Stefanovici, G. Amvrosiadis, and A. A. Hwang, “Temperature Management in Data Centers: Why Some (Might) Like It Hot,” *SIGMETRICS ’12: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 163-174, Jun. 2012.

[2] Z. Xiong, M. Zhao, Z. Yuan, J. Xu, and L. Cai, “Energy-saving Optimization of Application Server Clusters Based on Mixed Integer Linear Programming,” in *Journal of Parallel and Distributed Computing*, vol. 171, pp. 111-129, Jan. 2023.

[3] A. D. Carnerero, D. R. Ramirez, T. Alamo and D. Limon, ”Probabilistically Certified Management of Data Centers Using Predictive Control,” in *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2849-2861, Oct. 2022.

[4] S. Mirhoseininejad, G. Badawy, and D. G. Down, “A Data-driven, Multiset Point Model Predictive Thermal Control System for Data Centers,” in *Journal of Networks and Systems Management*, vol. 29, no. 7, 2021.

[5] S. Rostami, D. G. Down, and G. Karakostas, “Linearized Data Center Workload and Cooling Management,” *arXiv:2304.04731 [eess.SY]*.

[6]

[7] S. MirhoseiniNejad, G. Badawy, and D. G. Down, “EAWA: Energy-aware Workload Assignment in Data Centers,” *2018 International Conference on High Performance Computing & Simulation (HPCS)*, Orleans, France, 2018, pp. 260–267.

[8] R. Gupta, S. Asgari, H. Moazamigoodarzi, D. G. Down, and I. K. Puri, “Energy, Exergy and Computing Efficiency Based Data Center Workload and Cooling Management,” in *Applied Energy*, vol. 299, 117050, Oct. 2021.