THERMAL-AWARE COOLING CONTROL AND WORKLOAD ASSIGNMENT FOR HETEROGENEOUS DATA CENTERS

THERMAL-AWARE COOLING CONTROL AND WORKLOAD ASSIGNMENT FOR HETEROGENEOUS DATA CENTERS

BY

Seyedmorteza Mirhoseninejad, M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF DEPARTMENT OF COMPUTING AND SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Doctor of Philosophy (2020) McMaster University

(Department of computing and software) Hamilton, Ontario, Canada

TITLE:	Thermal-aware cooling control and workload assignment for heterogeneous data centers
AUTHOR:	Seyedmorteza Mirhoseninejad M.Sc. (Computer Engineering), Iran University of Science and Technology, Tehran, Iran
SUPERVISORS:	Dr. Douglas G. Down and Dr. Ghada Badawy
NUMBER OF PAGES:	xvi, 181

Lay Abstract

The power savings opportunities available by exploiting the tight correlation between computing equipment and cooling unit behavior are explored in this thesis. Recognizing different aspects of thermal heterogeneity in data centers using thermal models is the key to our work. We first design a workload assignment algorithm that leverages differences in the thermal behavior of servers. The promising results of this approach inspire us to scrutinize the power savings opportunities that exist from the cooling unit point of view, in the sense that different locations in the data center differ in their susceptibility to cooling (cooling heterogeneity). This initial work relies on the analytic development of physical models to describe thermal behavior. Due to the difficulties in generating such models, we then develop a data-driven thermal model for temperature predictions in data centers. The accuracy and low complexity of this model are conducive to deployment in practice. In the next phase, cost-saving opportunities arising from considering cooling and server heterogeneity together are shown. Finally, a holistic infrastructure control system leveraging the thermal model is implemented on a real data center.

Abstract

Data centers are struggling with inefficient power usage, which is one of their crucial challenges. Information technology (IT) and cooling infrastructure are the major contributors to power consumption in data centers. Server over-cooling, inefficient power management of cooling units, and thermal-oblivious assignment of server workload are significant contributors to the considerable power wastage in data centers. These issues can be addressed by recognizing cooling units' ability to dissipate heat from different locations inside a data center (cooling heterogeneity) and thermal properties of individual servers (server heterogeneity). This problem has not been studied thoroughly in the literature.

This dissertation consists of five phases aiming to exploit the correlation between IT and cooling units in data centers for the efficient use of power. The study begins with exploiting thermal differences between servers and ends with implementing a complete holistic thermal-aware control system. The first phase identifies server differences due to their cooling requirements and power consumption. Hence, the problem of distributing workload to minimize power consumption while respecting the thermal differences between servers (server heterogeneity) is considered. The resulting optimization problem is addressed using an effective heuristic. This heuristic distributes workload among servers in a way that minimizes their cooling requirements

and sets the cooling set-point accordingly.

The second phase investigates data center cooling heterogeneity, exploring the thermal differences among servers and between server locations that need to be cooled by cooling units. A physics-based thermal model is used to calculate the inlet temperatures of servers based on cooling and IT settings. It is shown that both the assignment of workload and the adjustment of cooling parameters affect the cooling cost, revealing a possible trade-off that can be optimized. Potential power-savings obtained by optimal assignment of workload and choices of cooling unit operational variables are explored.

Due to their complexity (both operationally and in their development), exploring synergies between IT and cooling units using physics-based thermal models is challenging. So, during the third phase of this work, an adaptive data-driven thermal model using time series prediction methods is developed. This thermal model, to a great extent, solves the problem of temperature predictions in data centers. This learning-based thermal model is fast, adapts to thermal changes in a data center, and does not require prior knowledge of heat transfer rules.

Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning is the subject of the next phase, which considers the problem of workload assignment and cooling control, combining the main aspects of thermal heterogeneity in data centers. It assumes thermal differences between servers and between locations in a data center using the thermal model constructed in the previous phase. The results show a potential to save a considerable amount of power as a result of leveraging synergies between the workload scheduler and control of the cooling unit.

Finally, a real-time control system is implemented which jointly controls cooling units and workload assignment in a data center. The controller in this system considers the thermal differences of servers to generate the expected thermal requirements corresponding to servers using a temperature requirement map. The capability of the cooling unit and also the thermal effects of servers are accounted for in this map. The system determines the operational variables of the cooling units using model predictive control to minimize the cooling power while satisfying the required temperatures given by the temperature map.

To my wife Fatemeh,

for her constant support.

Acknowledgements

First of all, I wish to express my deepest gratitude to Professor Douglas Down, my Ph.D. supervisor, for all his support towards the accomplishment of my research. I genuinely appreciate his patience and understanding. He taught me meaningful research and his attention to the details, editorial vigilance, and supportive nature made a deep-felt experience into my life. During this research, I have felt fully supported and encouraged by him to conduct my research and collaborate with other researchers. I would also like to show my gratitude to Dr. Ghada Badawy for sharing her pearls of wisdom with me during this time. She was always available for brainstorming meetings and helped me to conduct experiments and provide technical supports.

I want to thank my advisory committee, Dr. Rong Zheng and Dr. Mark Lawford, for their excellent constructive insights and invaluable feedback that greatly helped to clarify issues and to improve the quality of the thesis.

I would like to thank Dr. Suvojit Ghosh and Dr. Souvik Pal, for their guides to conduct my research and sharing their valuable insights into my work. I would like to express my thanks to Dr. Hosein Moazamigoodarzi, Dr. Masoud Kheradmandi, Mehdi Jafari, and Fernando Martínez García for their help and comments to improve my work. In addition, the physical and technical contribution of the Computing

Infrastructure Research Center (Hamilton), is genuinely appreciated.

I wish to acknowledge the support of my family. I immensely grateful to my wife Fatemeh for her endless support and to my parents Zahra and Ali for being the best teachers of my life.

Contents

Lay Abstract **iii**

Abstract **iv**

Acknowledgements **viii**

List of Figures **xiii**

List of Tables **xvi**

1 Introduction **1**

2 EAWA: Energy-Aware Workload Assignment in Data Centers **7**

2.1 Introduction 8

2.2 Background 12

2.3 Energy-aware workload Assignment 13

2.4 Discussion 26

2.5 Conclusion 31

Bibliography 32

3 Joint Data Center Cooling and Workload Management: A Thermal-Aware Approach **36**

3.1 Introduction 37

3.2 Literature review 40

3.3 Thermal-aware workload scheduling and cooling control 47

3.4 Formulating the optimization problem and comparing the results . . . 61

3.5 Thermal effects of server consolidation 66

3.6 Conclusion 71

3.7 Acknowledgment 72

Bibliography 72

4 ALTM: Adaptive learning-based thermal model for temperature predictions in data centers **79**

4.1 Introduction 80

4.2 Literature review 83

4.3 Thermal model 85

4.4 Results 93

4.5 Conclusion 96

Bibliography 97

5 Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning **101**

5.1 Introduction 103

5.2 Literature review 105

5.3 System architecture and models 111

5.4 Thermal-aware cooling control and workload assignment 120
5.5 Results and comparison 125
5.6 Conclusion 132
Bibliography 133

6 IT-aware cooling control framework for data centers: A machine learning control approach 140

6.1 Introduction 142
6.2 Literature review 144
6.3 The methodology 148
6.4 Results 160
6.5 Discussion 167
6.6 Conclusion 168
Bibliography 169

7 Conclusion 175

Bibliography 179

List of Figures

1.1 The logical connections between the chapters 6

2.2 Wires blocking at the back of servers 10

2.3 Maximum required inlet temperature for two different configurations 18

2.4 Power consumption of data center for two workload assignment methods 26

2.5 Comparing power consumption of EAWA and uniform assignment . . 27

2.6 Total power consumption of data center for three workload assignment methods 28

2.7 Temperature of the cooling unit set-point for two methods 31

3.1 Schematic of the IT enclosure integrated with a single rack and an RMCU with separated cold and hot chambers. 48

3.2 Depiction of energy sources at the inlet and outlet zones of a server . 51

3.3 The temperature distribution at the front of servers for different combinations of operational parameters of the cooling unit 56

3.4 The trade-off between operational parameters of the cooling unit for different red line temperatures. 57

3.5 The trade-off between operational parameters of a cooling unit under different workload assignments. 59

3.6 Temperature distribution at the front of servers 60

3.7 Data center schema for two configurations of the data center and servers 63
3.8 Optimized workload assignment and temperature distribution for both configurations 64
3.9 Optimal utilization for different workload assignment methods 66
3.10 Temperature distribution of servers for different methods of workload assignment 67
3.11 Thermal effects of server consolidation 68
3.12 Heat recirculation inside a data center 71
4.1 Front view of an in-row cooling data center 86
4.2 The top view of the in-row cooling data center 87
4.3 In-row cooling schema 88
4.4 Temperature prediction of the neural network vs wRLS models 93
4.5 The box plot representation of the 25 measured temperatures 94
4.6 100 steps projection error for the neural networks model 95
4.7 The comparison between adaptive and non-adaptive thermal models . 96
5.1 Front view of data center with two in-row cooling units at either side and five IT racks 112
5.2 Top view of data center 112
5.3 In-row cooling unit 113
5.4 Closed-loop NARX network 115
5.5 MATLAB implementation of Closed-loop NARX network 115
5.6 Temperature of Zone 1 116
5.7 CPU temperature of a server in Zone 2 117
5.8 Thermal model example 121

5.9 Demonstration of the trade-off within cooling operational parameters 122
5.10 Thermal model intuition 124
5.11 Power consumption of the cooling using CHIC 126
5.12 The optimized operational variables of the cooling unit using CHIC . 127
5.13 Power consumption of the cooling unit using HDIC 127
5.14 The optimized operational variables of the cooling unit using HDIC . 128
5.15 Cooling power comparison 129
5.16 Coefficient of performance of CHIC vs HDIC 129
6.1 Data center front view: location of fans and arrangement of thermal zones 149
6.2 Top view of the data center 149
6.3 DS18B20 digital thermometer 150
6.4 Heat-map representation of inlet temperatures of servers based on different RTDMs 156
6.5 System outputs (first scenario) - Temperature of 25 thermal zones . . 162
6.6 System inputs (first scenario) - Fan speeds 162
6.7 Cooling power consumption (first scenario) 163
6.8 System outputs (second scenario)- Temperature of 25 thermal zones . 164
6.9 System inputs (second scenario) - Fan speeds 164
6.10 Cooling power consumption (second scenario) 164
6.11 System outputs (set-point-tracking) - Temperature of 25 thermal zones 165
6.12 System inputs (set-point-tracking) - Fan speeds 166
6.13 Comparing power consumption 166

List of Tables

2.1 Notations 14

2.2 Workload assignment of first few servers and the corresponding power consumption for both solutions 24

2.3 Coefficient of the baseline models per each type 25

2.4 Difference between power consumption of exact and average models in percentages 29

3.1 Notation 40

3.2 Summary of related work 46

5.1 Notation 106

5.2 Inlet and CPU temperature of servers corresponding to CHIC and HDIC 130

5.3 Comparing HDIC with HRM-based and Set-point Tracking approaches 132

6.1 Comparison of R_1 and R_2 156

Chapter 1

Introduction

Data centers are amongst the largest power consumers on Earth [1]. On the one hand, IT equipment is becoming more compact, resulting in higher densities of computing power. On the other hand, the computing landscape has moved towards migrating processes, applications, and services to data centers. This trend necessitates more IT infrastructure to support the increasing demand by expanding data centers [2]. Hence, this power-hungry infrastructure requires great attention with respect to its power efficiency. The dissipation of generated heat in data centers, mainly by servers, is a significant challenge. So, the efficient use of power to cool data centers has motivated many investments and studies. There is much effort (both in industry and academia) towards decreasing the energy consumed by the IT installed in data centers as well as corresponding cooling costs [3].

Addressing the power consumption of IT has been done at different levels. For example, low power transistor and IC designs [4, 5], different power states of devices and processors [6], powering off unneeded servers considering the affected performance [7], and high-performance power supplies, transformers, and power distribution units

(PDUs) [8]. Additionally, the design of cooling units has been explored at various levels from high-performance air blade, fan, and chiller designs to the design of efficient cooling architectures that couple with IT infrastructure, such as raised floor architectures [9].

There is a body of work on thermal-aware workload assignment approaches that use thermal models for temperature predictions to discover cooling heterogeneity in data centers. Tang et al. [10] develop a thermal-aware workload manager to minimize the peak server inlet temperature through an optimal assignment of workload. Their work is based on a static heat re-circulation matrix (HRM). Abbasi et al. [11] minimize the total amount of heat re-circulation to increase the cooling unit supply air temperature. Mukherjee et al. [12] extend Abbasi's work [11] by considering job deadlines as extra constraints in the power minimization process. Moreover, servers can be slowed down to throttle temperature peaks.

Fang et al. [13] propose a data center control and management framework by jointly making IT and cooling decisions to save power. The solution to the underlying optimization problem finds the optimal active server set, job assignment, and set-points of cooling units to minimize the total power consumption. Their thermal model corresponds to an HRM. Zhao et al. [14] present a control method to reduce power consumption, using a control loop to maintain inlet temperatures of servers at an appropriate set-point by dynamically adjusting operating frequencies and utilizations of servers. Their thermal model is also based on an HRM.

In general, these methods suffer from a couple of issues. An HRM is neither accurate enough for temperature predictions nor appropriate for the dynamic environment of a data center. Even extensions to a static HRM, such as the HRM-based

approach enhanced by considering airflow changes proposed by Wang et al. [15] have drawbacks with respect to implementation. Additionally, in [15], the only cooling unit parameter is the set-point, more granular controllable variables of the cooling units are not considered.

The literature lacks a method for fully IT-aware cooling control or holistic control of data centers. We show that exploiting thermal heterogeneity in a data center can allow for effective joint workload assignment and cooling control for saving power. This consideration can considerably decrease the data center power consumption.

One aspect of data center thermal heterogeneity is the thermal characteristics of servers. We show that even servers of the same type or make could have different thermal characteristics, leading to different cooling costs. Also, we show that power-saving potentials exist when considering cooling heterogeneity in data centers. However, the main issue in accurately modeling the cooling heterogeneity is a thermal model that estimates temperatures in a data center — the traditional method has been to use physics-based thermal models. We investigate learning-based methods through regression and neural network methods. We find that through an appropriate implementation of neural networks, an accurate thermal model can be constructed. This model is a promising alternative to computationally expensive and difficult to scale physics-based thermal models. This neural network thermal model is used in another work to construct a framework that jointly assigns workload and adjusts the operational parameters of the cooling units in a power-efficient manner. In our last piece of work, a model predictive control method is used to cool servers based on temperature requirements of servers. Our work is performed and described in five consecutive papers, which we now outline.

In the first paper [16], a workload assignment method is designed considering thermal differences between servers. It uses a polynomial regression method to relate the temperature of critical components inside a server to its inlet air temperature and processing load. An optimization process, considering thermal models of all servers, assigns workload in a manner that decreases the overall temperature requirements of servers while respecting temperature constraints of server components. This process also provides feedback for cooling units to adjust their supply air temperature.

Determining power savings opportunities by considering cooling heterogeneity is the matter of our next work [17]. Exploiting cooling heterogeneity requires relating the cost of cooling different locations inside a data center. We use a zonal-based thermal model for the temperature estimates. This thermal model is based on laws of physics and heat-transfer equations for calculating the temperature of thermal zones at the front of the servers. It is shown that both adjusting the cooling units' operational parameters and the assignment of workload considerably affects the cooling power. We form an optimization problem and find that an optimal choice of operational parameters and assignment of workload can result in a considerable amount of power savings.

We address the problem of constructing the thermal model in our next work [18]. The physics-based thermal models used in the previous work are design-specific, and are somewhat simplified. We desire to have a thermal model that is fast, accurate, adaptable to thermal changes in a data center, and does not require prior knowledge of heat transfer rules between data center entities. Hence, we present a high precision learning-based thermal model using neural networks that predicts the temperature of critical zones using data center operational variables. The operational variables are

controllable parameters of IT and cooling units, such as server loads and fan speeds.

In the next step of our study [19], data center thermal heterogeneity is exploited from all aspects for workload assignment and cooling control, resulting in a considerable amount of savings in the total data center power consumption. A neural network thermal model (as described in the previous paragraph) is used for the temperature predictions for the inlet air temperature of servers. A thermal model of servers provides the required inlet temperature of servers as a function of their utilization. The frameworks for obtaining thermal models are presented. The thermal models are then incorporated as the core of an optimization process for the workload assignment and cooling control. We observe considerable power saving possibilities as a result of using our methods.

The final step of this study is to construct a holistic thermal-aware workload assignment and cooling controller for data centers [20]. The fast and accurate temperature estimates of the neural network thermal model provide us with an excellent tool for implementing real-time control. We introduce a control mechanism which can cool servers based on a given map of required server inlet temperatures rather than a single set-point. This pattern of required temperatures is optimized through another process that considers cooling units' ability to cool different locations and the assigned workload of servers. Implementing the framework on a data center with in-row cooling shows the potential for considerable power savings compared to other conventional controllers.

The rest of this dissertation is as follows. To aid the reader in navigating the thesis, the logical connections between the chapters are provided in Figure 1.1. The next chapter (Chapter 2) considers thermal heterogeneity of servers, with the title *EAWA:*

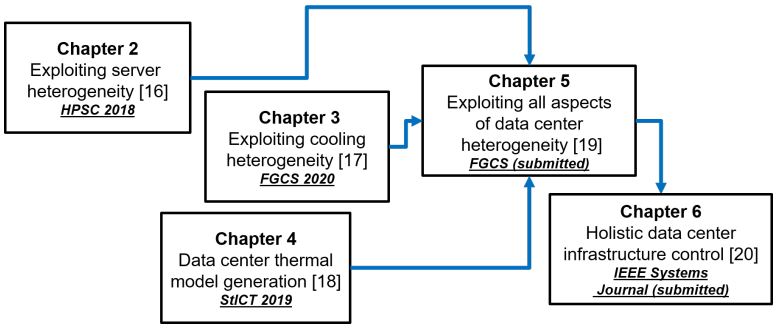


Figure 1.1: The logical connections between the chapters

Energy-aware workload assignment in data centers. Chapter 3 is *Joint data center cooling and workload management: A thermal-aware approach*, which exploits cooling heterogeneity using a physics-based thermal model. *ALTM: Adaptive learning-based thermal model for temperature predictions in data centers* is the subject of Chapter 4, which demonstrates the use of neural networks for the time-series predictions. Chapter 5 includes a complete heterogeneity-aware framework for workload assignment and cooling control in data centers. *Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning* is the title of this work. Chapter 6, *IT-aware cooling control framework for data centers: A machine learning control approach*, explains the design of a system for workload and data center control. This work describes the physical implementation of a controller that considers the transient behavior of a data center. Finally, the last chapter concludes our achievements and remarks on the highlights. It also discusses possibilities for improvements and plans for future work.

Chapter 2

EAWA: Energy-Aware Workload Assignment in Data Centers

This chapter is reproduced from "EAWA: Energy-Aware Workload Assignment in Data Centers", SeyedMorteza MirhoseiniNejad, Ghada Badawy, and Douglas G. Down, published in International Conference on High Performance Computing & Simulation (HPCS), pp. 260 - 267, IEEE, 2018.

The author of this thesis is the first author and the main contributor of this publication. His contributions to this work consist of introducing the idea of server heterogeneity, writing the manuscript, formulating the optimization problem, conducting the experiments, implementing the framework, and generating the numerical results.

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

2.1 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

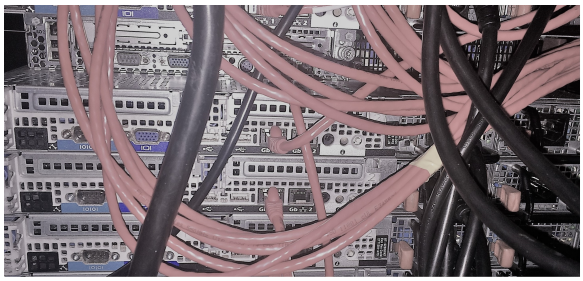


Figure 2.2: Wires blocking at the back of servers

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]. Fig. 2.2 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 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. We also note that our algorithm includes performance constraints. 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.

2.2 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.

2.3 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.

Table 2.1: 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 temp. of i^{th} server (°C)
T_{in}^{req}	Vector of maximum allowable inlet temp. of servers
T_{set}	Set-Point temperature of cooling unit
CoP	Coefficient of performance
δ_u	Workload unit which can be assigned to a server

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 2.1.

2.3.1 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 (2.3.1). The subscript i denotes the i^{th} server.

$$P_{server,i} \approx P_{cpu,i} = c_{1,i} + c_{2,i} \cdot u_i + c_{3,i} \cdot T_{cpu,i} + c_{4,i} \cdot T^2_{cpu,i}. \tag{2.3.1}$$

This model has one significant contributing factor, the CPU utilization, u_i. Although (2.3.1) shows that temperature affects CPU power consumption, it is negligible in comparison with CPU utilization. The authors in [17] simplified the model to (2.3.2) but there are also other works such as [12, 20, 21] that have also used (2.3.2). 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 (2.3.2) throughout this paper.

$$P_{server,i} = c_{1,i} + c_{2,i} \cdot u_i \tag{2.3.2}$$

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 $c_{2,i}$.

2.3.2 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.

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.

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. Equation (2.3.3) 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 (2.3.3), however with different coefficients.

$$T_{cpu,i} = \beta_{1,i} + \beta_{2,i} \cdot u_i + \beta_{3,i} \cdot T_{in,i} + \beta_{4,i} \cdot u_i^2 + \beta_{5,i} \cdot u_i \cdot T_{in,i}. \tag{2.3.3}$$

Coefficients of (2.3.3) 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 (2.3.3), 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}),

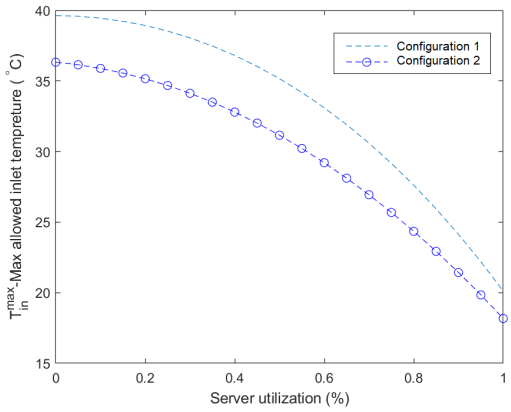


Figure 2.3: Maximum required inlet temperature for two different configurations

the maximum allowable inlet temperature can be calculated with respect to u_i.

$$T_{in,i}^{req} = \frac{T_{cpu}^{red} - (\beta_{1,i} + \beta_{2,i} \cdot u_i + \beta_{4,i} \cdot u_i^2)}{\beta_{3,i} + \beta_{5,i} \cdot u_i} \tag{2.3.4}$$

The curves in Fig. 2.3 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.

2.3.3 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.,

$$P_{total} = P_{it} + P_{cool}. \tag{2.3.5}$$

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:

$$P_{it} = \sum_{i=1}^{n} (c_{1,i} + c_{2,i} \cdot u_i). \tag{2.3.6}$$

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:

$$CoP = \frac{P_{it}}{P_{cool}} \quad or \quad P_{cool} = \frac{P_{it}}{CoP}. \tag{2.3.7}$$

Using (2.3.5) and (2.3.7), the total power consumption is thus expressed as:

$$P_{total} = (1 + \frac{1}{CoP}) \cdot P_{it}. \tag{2.3.8}$$

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

$$CoP = \gamma_1 + \gamma_2 \cdot T_{sup} + \gamma_3 \cdot T_{sup}^2. \tag{2.3.9}$$

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:

$$T_{sup} = T_{set} = min(T_{in}^{req}). \tag{2.3.10}$$

Combining (2.3.8) and (2.3.9),

$$P_{total} = (1 + \frac{1}{CoP(T_{sup})}) \cdot \sum_{i=1}^{n} (c_{0,i} + c_{1,i} \cdot u_i). \tag{2.3.11}$$

In (2.3.11), 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.

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

In the minimization problem, P_{total} is given in (2.3.5) or equivalently in (2.3.11). 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, as simply minimizing the power consumption may result in unacceptable performance. For example, one can use queuing-theoretic techniques to determine u^{max} [24]. We have chosen to constrain the performance indirectly through the utilization, but it is possible to include explicit performance constraints. This may be at the cost of a more complex optimization problem. In this problem D is supposed

to be a cap for the current status of the offered load to the system.

2.3.4 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 $getOptimalServer$

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 (2.3.4)
$CoP_{val} = CoP(min(inletTemp))$; %From (2.3.9)
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

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 (2.3.11) 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 2.2. In fact, SQP is allowed to have finer-grained utilization values which fits a little better

Table 2.2: Workload assignment of first few servers and the corresponding power consumption for both solutions

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

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.

2.3.5 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

Table 2.3: 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

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 (2.3.11) - 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 2.3 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 Fig. 2.4. 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

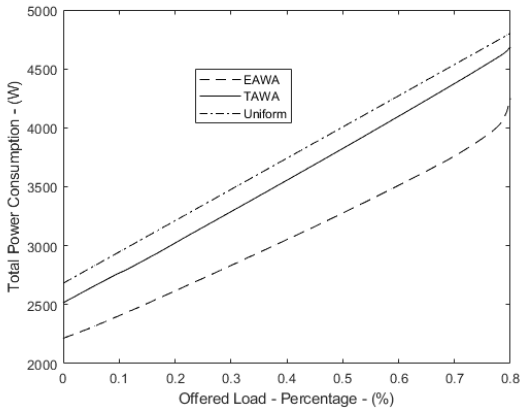


Figure 2.4: Power consumption of data center for two workload assignment methods

is notable. Significant savings come from reducing over-cooling.

2.4 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 2.3) as explained in Section 2.3.5. In this section, the data center includes 100 servers and $u^{max} = 0.8$.

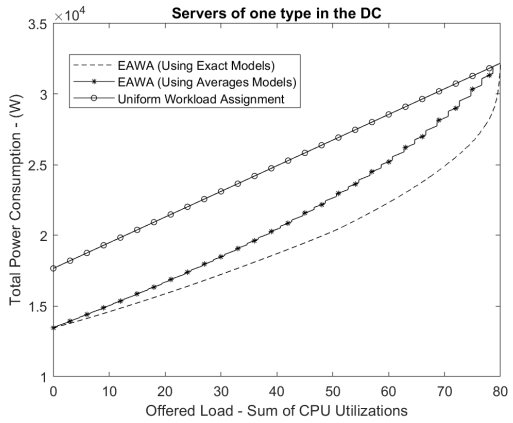


Figure 2.5: Comparing power consumption of EAWA and uniform assignment

2.4.1 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. Fig. 2.5 shows power consumption curves for three workload assignment methods based on the offered load. As expected, if EAWA uses the *exact model*, it 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.

2.4.2 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

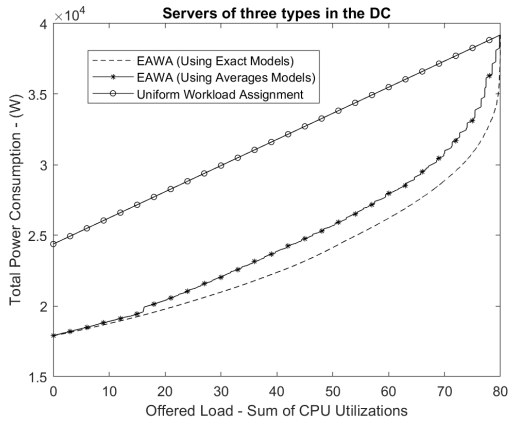


Figure 2.6: Total power consumption of data center for three workload assignment methods

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, Fig. 2.6 shows the result for three types of servers.

2.4.3 Average model versus exact model

In Fig. 2.5 and Fig. 2.6 results for both *exact models* and *average models* are presented. Using each of these models for workload assignment purposes has pros and

Table 2.4: Difference between power consumption of exact and average models in percentages

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

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.

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 (2.3.2) and (2.3.3) using a polynomial curve fitting method.

Table 2.4 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.

2.4.4 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. Fig. 2.7 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 a low amount of workload during the majority of their life because of over-provisioning that exists during their design. On the other hand, Fig. 2.7 shows that if the offered load is light, the set-point curve is flat. The set-point

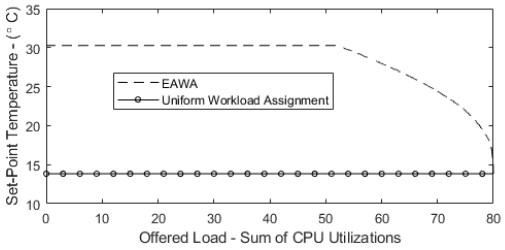


Figure 2.7: Temperature of the cooling unit set-point for two methods

could then be set to the minimum required value (that occurs at maximum load) once the offered load rises above a threshold.

2.5 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.

Bibliography

[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*, pp. 877–880, IEEE, 2012.

[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*, pp. 20–25, IEEE, 2016.

[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*, pp. 34–34, IEEE, 2005.

[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*, pp. 319–330, IEEE, 2011.

[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," tech. rep., Technical report, McMaster University, 2016.

[11] V. J. Maccio and D. G. Down, "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*, pp. 11–20, IEEE, 2016.

[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, p. 140, 2007.

[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," tech. rep., HPL-2006-55, 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*, pp. 61–75, 2005.

[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, pp. 337–350, 2008.

Chapter 3

Joint Data Center Cooling and Workload Management: A Thermal-Aware Approach

This chapter is reproduced from "Joint Data Center Cooling and Workload Management: A Thermal-Aware Approach", SeyedMorteza MirhoseiniNejad, Hosein Moazamigoodarzi, Ghada Badawy, and Douglas G. Down, published in Future Generation Computer Systems, vol. 104, pp. 174 - 186, 2020.

The author of this thesis is the first author and the main contributor of this publication. His contributions to this work consist of introducing the main idea, writing the manuscript, formulating the optimization problem, implementing the framework, and generating the numerical results.

Abstract

Information technology (IT) equipment and cooling infrastructure are key contributors to the total energy expenditure in a data center. There is typically significant power wastage due to inefficient cooling control and thermal-oblivious management of workload. Recent thermal-aware data center management techniques have not taken a unified approach in controlling IT and cooling systems. In this paper, we find that considering thermal effects of server workloads, in conjunction with control parameters of the cooling unit saves more power than optimizing each of them separately. We leverage a low complexity holistic data center model that considers thermal interactions between IT and cooling unit entities. This thermal model provides control decisions with fine-grained control variables. We propose *joint cooling and workload management (JCWM)*, which has the potential to save a considerable amount of power by exploring synergies between the workload scheduler and operational parameters of the cooling unit. In addition, we provide a significant caveat for the power efficiency of server consolidation methods when taking into account associated thermal effects.

Keywords: data center workload assignment, cooling unit control, thermal-aware scheduling, thermal model, data center power efficiency, efficient cooling

3.1 Introduction

Data centers in the United States consumed 70 billion kWh in 2014, 1.8 percent of the domestic power consumption [1]. In contrast, the power consumption of data centers in 2000 was 30 Billion kWh [2]. This increase in the power consumption is the

result of increasing public use of cloud platforms, on-line applications, and Internet services [3]. It has been estimated that from 2015 to 2020 the incoming load to data centers will double [4]. Foreseeing this increase and power usage constraints have led large data center vendors to invest more in the efficient use of power. There are many ongoing research projects on the efficient use of data center facilities, aiming to decrease the power consumption [1].

There are a number of methods and techniques to reduce power consumption at different levels of a data center. At the device level, some electronic devices support low power states to save energy, if performance of the device is not impacted [5, 6]. For example, dynamic voltage and frequency scaling (DVFS) is a method that provides different levels of power consumption and performance for processors [7, 8]. At the server level, dynamic suspension of unneeded servers, server consolidation and the ability to choose different levels of power and performance are key approaches for energy efficiency. For instance, server consolidation aims to save power by turning unneeded servers off during low workload periods [9, 10, 11]. At the facility level, power efficiency of the cooling system itself is also a significant concern [12, 13, 14].

Our goal is to decrease power wastage that is initiated from cooling and IT over-provisioning. Efficient solutions to these problems can lead to considerable cost savings and performance gains. Previously, we showed that there can be significant power savings through considering cooling unit control and workload management together [15]. In this paper, we expand our previous work by considering fine-grained cooling variables along with workload assignment.

A key component of our work is the derivation of a fast and accurate thermal model for a micro data center that provides inlet server temperature distribution

given server workloads and cooling parameters. This model can be extended to larger data centers.

Appropriately constraining the temperature at the front of servers is a factor that affects server health [16]. Thus, we investigate the variations of temperature distribution along with cooling power consumption. It is shown that there is an optimal operational point for the cooling unit parameters to satisfy thermal requirements of servers. Moreover, we show that workload assignment and cooling variables can be jointly optimized, minimizing the cooling power consumption.

We noticed a very important thermal effect when a server is turned off, that may need to be considered in server consolidation techniques. A server that is turned off results in a pathway for hot air from the back of servers to the front, which increases the inlet temperatures of adjacent servers. Compensating for this excess heat might be at the expense of increased cooling power. Using the proposed thermal model we show that this extra cooling power might be greater than the amount saved by turning off servers.

With this introduction in mind, we now list our main contributions:

- Applying low complexity physical models (zonal models) to calculate the temperature distribution within a data center
- Illustrating the trade-off that exists between cooling operational parameters
- Showing that the optimality of cooling parameters depends on the assignment of workload
- Formulating an optimization problem that jointly considers workload assignment and cooling control
- Introducing a hidden thermal challenge raised by server consolidation methods
- Investigating the optimization possibilities considering adverse thermal effects of server consolidation

Table 3.1: Notation

Variable	Definition
n	Total number of servers
d	Offered workload
u_i	CPU utilization of i^{th} server
$\bar{u}$	Utilization vector of length n (vector of u_is)
u_{max}	Maximum allowed CPU utilization
c_i	i^{th} coefficient of server power model
Q^{rmcu}_{water}	Water flow-rate (cfm)
Q^{rmcu}_{air}	Air flow-rate (cfm)
$Q^{server}_{air,i}$	Air flow-rate of server
T^{water}_{inlet}	Temperature of inlet water (°C)
$T^{server}_{inlet,i}$	Air temperature at the front of i^{th} server (°C)
$\bar{T}^{server}_{inlet}$	Vector of $T^{server}_{inlet,i}$s
T^{server}_{red}	Red line temperature of servers (°C)
P^{dc}	Total power consumption of data center (Watt)
P^{it}	Power consumption of IT (Watt)
$P^{cooling}$	Total power consumption of cooling unit (Watt)
P^{fan}	Power consumption of cooling fans (Watt)
$P^{chiller}$	Power consumption of chiller (Watt)
P^{server}	Power consumption of server (Watt)

A summary of the notation used in this paper is listed in Table 3.1.

3.2 Literature review

There is a significant body of literature that considers various methods of efficient workload assignment and cooling control. In this section, a number of previous works, related to our contributions, are reviewed: thermal-aware workload assignment, data center control, and server consolidation methods.

Assigning workload to servers while considering associated thermal effects has a significant literature. Sharma et al. [17] presented a framework for thermal load

balancing. Workload is assigned to servers inversely proportional to the exhausted air temperature. The same workload assignment decision is considered by Chaudhry et al. [18], but for a different problem statement. Moore et al. [19] introduced a number of temperature-aware methods. Their first method, which is based on [17], uses server inlet temperature and current workload of neighboring servers to assign workload to a server. Their second approach assigns workload to servers based on the recirculation of heat between servers. Although, these methods could somewhat save power, none of them necessarily minimizes the power consumption, as is shown later in this paper.

In [13], Bash and Forman presented a method that they called *cool job assignment*. They suggested to place jobs in cooling-efficient locations, ranked according to an index. 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. This approach could be efficient to minimize the power consumption if the cooling-efficient location does not change. However, air pattern changes due to fan speed variations. As is illustrated in this paper, different air patterns create different cooling-efficient locations.

Tang et al. [12] developed a thermal-aware workload manager. Peak inlet temperature is minimized through optimal assignment of workload based on a static heat recirculation matrix (HRM). The utilization of each server is assumed to be 0 or 1. They used two optimization methods to determine the optimal utilizations. The other work that is highly motivated by Tang's paper [12] is Abbasi et al. [20], who tried to minimize total power consumption of a data center while maintaining a service level agreement (SLA). Their algorithm consists of two phases, *thermal-aware*

server provisioning (TASP) and *thermal-aware workload assignment* (TAWA). TASP, a server consolidation method, considers turning a subset of servers off. An optimization problem is formulated based on the current workload to find the most energy efficient active set of servers during a given time window. In TAWA they designed an algorithm that distributes workload among servers in a manner that minimizes total power consumption. TAWA tries to minimize the total amount of heat recirculation; however, the main difference between TAWA and the approach that is taken in [12] is that finer-grained workload can be assigned to a server.

In [21], Mukherjee et al. developed a thermal-aware workload assignment algorithm that minimizes the power consumption while respecting given performance constraints (deadlines). Their approach assigns jobs to energy efficient servers and reduces heat recirculation. Their main contribution is that they allow jobs to be slowed down to throttle temperature peaks. The main drawback for Tang's [12], Abbasi's [20], and Mukherjee's [21] works is that their thermal model relies on a static HRM that may not be appropriate for the dynamic environment of a data center.

Wang et al. [22] presented a learning-based method which reflects the influence of air flow-rates on the parameters of the HRM. In this work, using an optimal air flow pattern, the hot-spot temperature is minimized. This is one of the few papers that demonstrates the inefficiency of HRMs and tries to adapt to changing air flows. However, while they do consider dynamic air flows, just one air flow is considered for each rack. From top to bottom, each rack may have a variety of air flows and assuming just one air flow for a rack endangers the model accuracy. Moreover, the air flow profile is the result of the action of multiple fans. The feasibility of providing the optimized air flow patterns by tuning fan parameters is not clear.

Wang et al. [23] considered the problem of optimal control of fan speeds. This work is interesting with respect to the thermal and heat transfer models that are used. They stated that fan speed in systems employing blade servers is typically over-provisioned. They employed a multi-input/multi-output (MIMO) control method to match fan speed to the requirements of the blades. They asserted that their control method could reduce power consumption of a blade by as much as 20%. Although this work introduced more accurate thermal models (in contrast with HRM-based models), there are opportunities to increase the resolution of fan effects on the cooling efficiency of cooling units.

Zhao et al. [24] presented a feedback controller to reduce power consumption in a data center. This work uses a control loop to maintain inlet temperatures of servers at an appropriate set point by dynamically adjusting operating frequencies and utilizations of servers. Their thermal model is based on an HRM. The idea of modelling heat generation via two factors, core frequency and utilization, and forecasting its effects in the future is a novel contribution to the literature. However, the drawbacks of their approach are the unrealistic thermal models and over-simplified cooling power models.

Fang et al. [25] presented a dynamic controller that considers both the IT and cooling decisions together to save power. The solution to the underlying optimization problem finds the optimal active server set, job assignment and set points of cooling units to minimize the total power consumption. The thermal model which is used in this work corresponds to an HRM. This work has a couple of issues. Firstly, the only cooling unit parameter is the set point, so they did not consider internal control of the cooling units. Secondly, the use of CFDs to calculate the HRM is not truly a

dynamic approach, as CFD calculations cannot be made on the same time scales as the thermal dynamics of the data center.

There are a number of works that have taken into account both energy sustainability and thermal-aware job assignment, notably the work of Zapater et al. [26] and Li et al. [4]. Zapater el al. [26] considered utilizing free cooling when the outside temperature is sufficiently low. They assumed a data center with in-row cooling (IRC) architecture. Additionally, a form of thermal-aware workload scheduler is suggested. The workload scheduler places similar jobs, in terms of their CPU and memory usage, physically close to each other. They stated that this grouping method, which they called the *power-balance* policy, balances per-rack temperature and increases cooling efficiency. The idea is that the same amount of cold air is required for servers that execute similar jobs. The control of each IRC unit avoids over-cooling or under-cooling of servers that are similar in their workloads. The ideas of using bypassing chillers, outside cold weather, and power-balancing are justifiable. However, it is not clear how practical their power balance policy is.

Li et al. [4] considered renewable energy options. Workload shifting is a mechanism that they suggest for this practice. Based on delay sensitivity of jobs and availability of renewable energy supplies, jobs are shifted toward maximizing using the renewable supplies. In order to satisfy cooling requirements, an HRM underlies the thermal model used in their optimization problem. The issues with the use of an HRM have already been discussed and we perform comparisons between the use of an HRM and our approach later in this paper.

In [15], we previously presented a thermal-aware cooling control and workload assignment method that has the potential to save a considerable amount of energy.

The method distributes workload amongst servers such that cooling requirements of servers are minimized. At the same time, the workload manager provides feedback to the cooling unit to work accordingly, which prevents over-cooling. Yao et al. [27] used an adaptive predictive control method for workload balancing in data centers. An adaptive thermal model is used to predict inlet temperatures. They formulate an optimization problem with a goal to smooth server inlet temperatures and decrease total power consumption. The cost function is formulated based on total power (cooling unit and IT racks) and tracking error of a predictive model. The controller adjusts the inputs (server workloads) to set the inlet temperatures while minimizing the total power consumption.

Later in this paper, thermal effects of server consolidation are investigated. Server consolidation methods simply turn unnecessary servers off, if there is a low offered workload for a period of time. We will show an important and interesting potential trade off for server provisioning decisions and to the best of our knowledge no work has considered the thermal effects of server consolidation. A number of works have investigated different server consolidation methods. These mostly provision servers based on forecasts of the offered workload [9, 10, 11, 20, 28]. The obvious side effect of turning servers off is performance degradation; the trade-off between data center performance and power consumption has been the subject of a number of studies [11, 29, 30]. Dabbagh et al. [31, 32] used a prediction method for the offered workload, based on Google Cluster workload traces. They used their prediction model for virtual machine placement and server provisioning decisions.

A summary of the most related literature is presented in Table 3.2.

Table 3.2: Summary of related work

Ref.	Goal	Thermal Model	Power Model	Optimiz. problem	Granularity	Validation	Approach	Cooling control
[17]	Reducing over-cooling	No	CoP-based	No	Rack or region	CFD	Assigning workload inversely proportional to exhaust air temperature of a server	No
[19]	Uniform exhaust air temperature	No	CoP-based	No	Server	CFD	Assigning workload proportional to inlet temperature of a server and state of adjacent servers	No
[18]	Reducing hot spot temperature	Inlet temperature sensitivity profile	Heat transfer equations	No	Server	Data center	Algorithm for relocating servers	No
[13]	Reducing the cooling cost	Workload placement index	No	No	Servers of cooling region	Data center	Assigning long jobs to servers of cool-efficient locations	No
[12]	Minimizing hot spot temperature	HRM-based	CoP-based	Yes	Server	CFD	Reducing heat re-circulation effects	Set-point
[20]	Minimizing hot spot temperature	HRM-based	CoP-based	Yes	Server	Power model	Selecting active set of servers and reducing heat re-circulations	Set-point
[21]	Minimizing hot spot temperature	HRM-based	CoP-based	Yes	Server chassis	CFD	CPU throttling while keeping deadlines	Set-point
[22]	Minimizing hot spot temperature	Adaptive HRM	No	Yes	Rack	CFD	Tuning air flow-rates of racks	Yes
[23]	Minimizing fan power and preventing over cooling	Heat transfer equations	Heat transfer equations	Yes	Server blade	Power model	Controlling blade fan-speed to adjust the temperature of blades	No
[24]	Balanced inlet temperature	HRM-based	CoP-based	Yes	Server	CFD	An MPC controls CPU frequencies and utilizations, and cooling set-point while maintaining system performance	Set-point
[15]	Increasing required inlet temperature	No	CoP-based	Yes	Server	Power model	Assigning workload inversely proportional to server cooling requirements	Set-point

3.3 Thermal-aware workload scheduling and cooling control

Cooling units do not provide a uniform temperature distribution within a data center. Some locations are easy to cool, while others are not. These differences stem from the physics of heat transfer and air recirculation. Taking such information into consideration when making cooling control and workload assignment decisions has the potential to yield significant power savings. One objective of this work is to optimize the operational parameters of the cooling unit and assignment of workload to minimize total power consumption. Operational factors are the control parameters of a cooling unit that affect the temperature distribution at the front of servers. So, a model is required to relate the operational parameters and the assigned workload to the temperature distribution. For this study, a model representing the power consumption of the cooling unit based on the current workload and operational parameters is also required.

3.3.1 Data center models

This section presents two important models for our study. The first model calculates the temperature at the front of each server based on assigned workload and operational parameters of the cooling unit. The second model returns the cost of tuning the operational parameters. The former is called the thermal model and the latter the power model. It is worth noting that the thermal model will be used as a monitoring tool to control servers' intake air by capturing thermal effects of each server and operational parameters of the cooling unit.

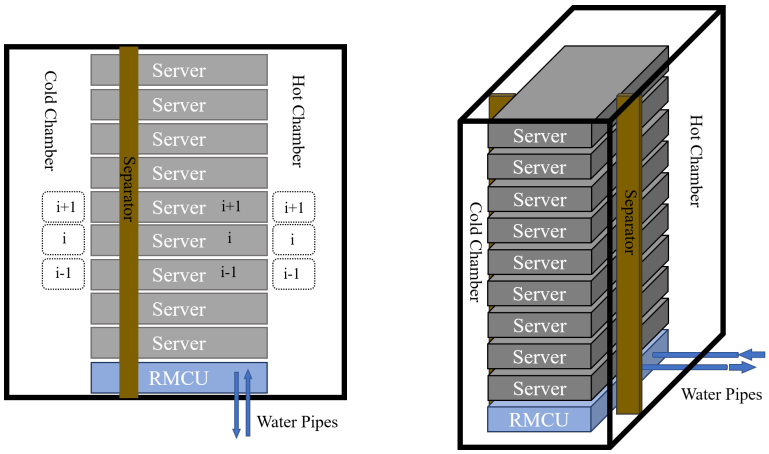


Figure 3.1: Schematic of the IT enclosure integrated with a single rack and an RMCU with separated cold and hot chambers.

Thermal model

We used a simple, low complexity zonal model to obtain the temperature distribution at the front of servers. Moazami et al. [33] developed this model for temperature prediction within a micro data center. A single rack with separated hot and cold chambers is considered in which a rack mountable cooling unit (RMCU) is installed at the bottom of the rack (Figure 3.1). The temperature prediction is based on a zonal approach that applies energy conservation to each zone. The zonal model is an intermediate method between full CFD simulations and multi-node lumped models. Physical quantities, such as temperature, are assumed uniform within a zone, simplifying spatial dependence. Here, the volume at the front of each server is considered as a zone as well as the volume at the back of each server.

Each server is considered as a heat source with specific heat transfer rate and predefined thermal capacity. Assuming no heat or mass transfer between the enclosure and the ambient, four control volumes are identified: (1) the cold chamber at the front of each server, (2) the hot chamber at the back of each server, (3) each server itself,

and (4) the RMCU. Due to the specific geometry of the enclosed rack integrated with the RMCU, the air flow-rates are predictable by applying mass conservation and characterizing the relation between air flow-rates and pressure drops for each component. After calculating all entering and exiting air flow-rates for each zone, using initial temperature values, we can apply the first law of thermodynamics (energy balance) for each zone.

To calculate air flow-rates the following steps are performed (in what follows, we will refer to a server that is on, either busy or idle, as an on server):

1. Determine the total air flow-rate for all on servers.
2. Obtain the air flow-rate of the cooling unit (a function of fan speed).
3. Determine the flow-rate mismatch between the cooling unit and all on servers.
4. Map the flow-rate mismatch to the pressure difference between the chambers.
5. Determine the flow-rates of off servers [33].
6. Determine the leakage flow-rate for each zone.
7. Determine the cold air flow input to each zone in the front chamber from the RMCU.
8. Find the output air flow-rate of the first zone in the front chamber using a mass balance equation and then use the result as the input air flow-rate of the second zone.
9. Calculate the input air flow-rate of other zones in sequence by repeating step 8.
10. Repeat steps 7 to 9 for the back chamber zones.

After calculating all input and output air flow-rates for each zone and given initial temperature values, we can apply the first law of thermodynamics (energy balance) for each zone to calculate the temperature at the next time step in a discretized version of the temperature dynamics. According to [33], the energy balance equation for a server can be written as:

$$\frac{X}{2}\left(\frac{dT_{out,i}^{server}}{dt}+\frac{dT_{inlet,i}^{server}}{dt}\right)=\rho_a c_{p,a} Q_{air,i}^{server}(T_{inlet,i}^{server}-T_{out,i}^{server})+P_i^{server} \tag{3.3.1}$$

In (3.3.1), $T_{inlet,i}^{server}$ and $T_{out,i}^{server}$ are the server inlet and outlet temperatures, ρ_a is the air density, $c_{p,a}$ is the specific heat of air, $Q_{air,i}^{server}$ is the air flow-rate of the server, X is the thermal mass of the server [34], and P_i^{server} is the total power consumption of the corresponding server. For air flow within the RMCU, based on [33], we have:

$$\begin{aligned}&\rho_a c_{p,a} V_a\left(\frac{dT_{outlet}^{rmcu}}{dt}+\frac{dT_{inlet}^{rmcu}}{dt}\right)=\\&\rho_a c_{p,a} Q_{air}^{rmcu}\left(T_{inlet}^{rmcu}-T_{outlet}^{rmcu}\right)-\frac{UA}{2}\left(T_{inlet}^{rmcu}+T_{outlet}^{rmcu}-T_{inlet}^{water}-T_{outlet}^{water}\right)\end{aligned} \tag{3.3.2}$$

and for water flow within the RMCU,

$$\begin{aligned}&\rho^{water} c^{water} V_w\left(\frac{dT_{inlet}^{water}}{dt}+\frac{dT_{outlet}^{water}}{dt}\right)=\\&\rho^{water} Q_{water}^{rmcu} c^{water}\left(T_{inlet}^{water}-T_{outlet}^{water}\right)+\\&\frac{UA}{2}\left(T_{inlet}^{rmcu}+T_{outlet}^{rmcu}-T_{inlet}^{water}-T_{outlet}^{water}\right)\end{aligned} \tag{3.3.3}$$

where T_{outlet}^{rmcu} is the air temperature at the RMCU outlet, T_{inlet}^{rmcu} is the air temperature at the RMCU inlet, T_{inlet}^{water} and T_{outlet}^{water} are the water inlet and outlet temperatures,

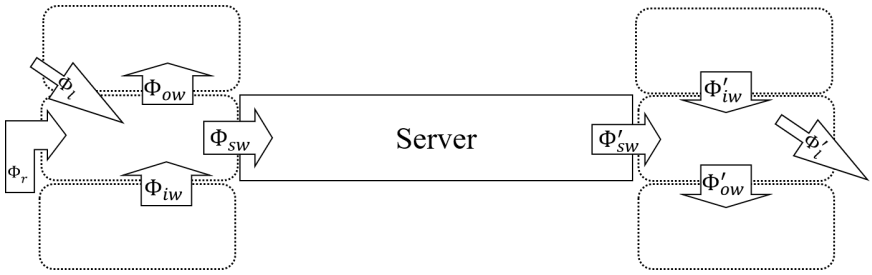


Figure 3.2: Depiction of energy sources at the inlet and outlet zones of a server

Q_{air}^{rmcu} is the air flow-rate of the cooling unit, Q_{water}^{rmcu} is the water flow-rate of the cooling unit, c^{water} is the specific heat of water, ρ^{water} is the density of water, U is the overall heat transfer coefficient inside the RMCU (as a function of Q_{air}^{rmcu} and Q_{water}^{rmcu} [34]), A is the contact area of water, and V_a and V_w are the air and water volumes inside the heat exchanger [33]. The energy balance equations for each cold and hot chamber zone can be written based on Figure 3.2. For the sake of simplicity, the energy balance equation is only shown for a zone in the cold chamber in (3.3.4); similar equations are derived for all of the cold and hot chamber zones.

$$\rho_a c_{p,a} V_c \gamma \left(\frac{dT_{inlet,i}^{server}}{dt} \right) = \Phi_r + \Phi_{iw} + \Phi_{ow} + \Phi_l + \Phi_{sw} \tag{3.3.4}$$

In (3.3.4), γ is a correction factor for the thermal masses, Φ_r corresponds to the energy that the zone receives from the RMCU, Φ_l is the leakage, Φ_{iw} is in-ward, Φ_{ow} is out-ward and Φ_{sw} is server-ward energy transfer of the zone. In Figure 3.2, the Φ and Φ' equations are the products of ρ_a, $c_{p,a}$, corresponding flow-rate, and corresponding temperature; for example, $\Phi_{sw} = -\rho_a c_{p,a} Q_{s,i} T_{inlet,i}^{server}$.

The model has been validated with extensive experiments and measurements. The geometry and zones for a rack within an enclosure that is cooled by an RMCU are shown in Figure 3.1. The RMCU is a heat removal module that transfers heat to a

chilled water loop supplied from an external chilled water system. The details of the air flow-rate calculations and energy balance equations are thoroughly explained in [33]. The proposed model captures the transient effect of server thermal mass on the temperature variation when switching servers on or off.

In this model, hot air recirculation through all possible media, in particular servers that are switched off, is considered. Turning a server off (as opposed to idling) provides a path for hot air from the hot chamber to the cold chamber. Capturing thermal effects of off servers will be used in this paper to consider the thermal effects of server consolidation methods. The thermal model can calculate the temperature profile of the cold chamber (temperature at the front of each server) as a function of water flow-rate, water inlet temperature, fan speeds or air flow-rate of the cooling unit, and power consumption of each server.

Power models

The total power consumption of a data center, P^{dc} is the sum of the power consumption of servers (which we call IT power or P^{it}) and cooling units ($P^{cooling}$):

$$P^{dc} = P^{it} + P^{cooling} \tag{3.3.5}$$

The major contributing factor of IT power (P^{it}) is the power consumption of servers. Power consumption of a server is modeled as an affine function of its utilization (u_i) [15]:

$$P_i^{server} = c_1 + c_2 \cdot u_i \tag{3.3.6}$$

In (5.3.3), c_1 is the power consumption of an idle server and $c_1 + c_2$ is the power

consumption of a fully utilized server. In this paper, we will assume that servers are homogeneous, so, c_1 and c_2 do not depend on the server (our approach is easy to modify if these values do depend on the server). Power consumption of the cooling unit is dominated by the chiller and fans:

$$P^{cooling} = P^{fan} + P^{chiller}. \tag{3.3.7}$$

A chiller provides cool water to the RMCU. The inlet water temperature (T^{water}_{inlet}) is the temperature of cool water provided to the RMCU by the chiller. The lower the value of T^{water}_{inlet} the higher the power consumption of the chiller. The model given in (5.3.6) represents the power consumption of the chiller [14]:

$$P^{chiller} = P^{heat} \cdot \left(\frac{\alpha_1 + \alpha_2 \cdot \frac{T^{chiller}_{evap}}{P^{heat}} + \alpha_3 \cdot (T^{chiller}_{cnd} - T^{chiller}_{evap})}{\frac{T^{chiller}_{evap}}{T^{chiller}_{cnd}} - \alpha_4 \cdot \frac{P^{heat}}{T^{chiller}_{cnd}}} - 1 \right). \tag{3.3.8}$$

In (5.3.6), P^{heat} is the total amount of heat that should be removed by the chiller (equal to P^{it} in our case). $T^{chiller}_{evap}$ is the evaporator temperature, which is approximately equal to T^{water}_{inlet}, and $T^{chiller}_{cnd}$ is the condenser temperature. The evaporator and condenser are the two main chiller components. Both $T^{chiller}_{evap}$ and $T^{chiller}_{cnd}$ are in kelvins and the quantities α_i are constants. While temperatures in these models are in kelvins, later in the paper temperatures will be reported in degrees Celsius.

The other contributing factor of the power consumption of the cooling unit is the power consumed by fans to provide the air flow-rate (Q^{rmcu}_{air}). A higher air flow-rate (Q^{rmcu}_{air}) requires higher fan speeds, which means greater power consumption by the

fans (P^{fan}). The power consumption of the RMCU fans is given by

$$P^{fan} = \beta_1 + \beta_2 \cdot Q_{air}^{rmcu} + \beta_3 \cdot (Q_{air}^{rmcu})^2 \tag{3.3.9}$$

where β_i, $i = 1, 2, 3$ are constants.

3.3.2 Optimal settings of cooling parameters and workload assignment

Fan speed and water inlet temperature are the contributing factors of the RMCU that determine both the server inlet temperature distribution and cooling unit power consumption. First in this section, the role of the cooling parameters in determining the temperature distribution is identified. We show that there exists a trade-off between different cooling parameters. We then investigate the effect of considering the distribution of workload as another factor. As workload assignment affects the optimal values of the operational parameters, we formulate an optimization problem that minimizes power consumption through joint control of cooling unit parameters and the distribution of workload.

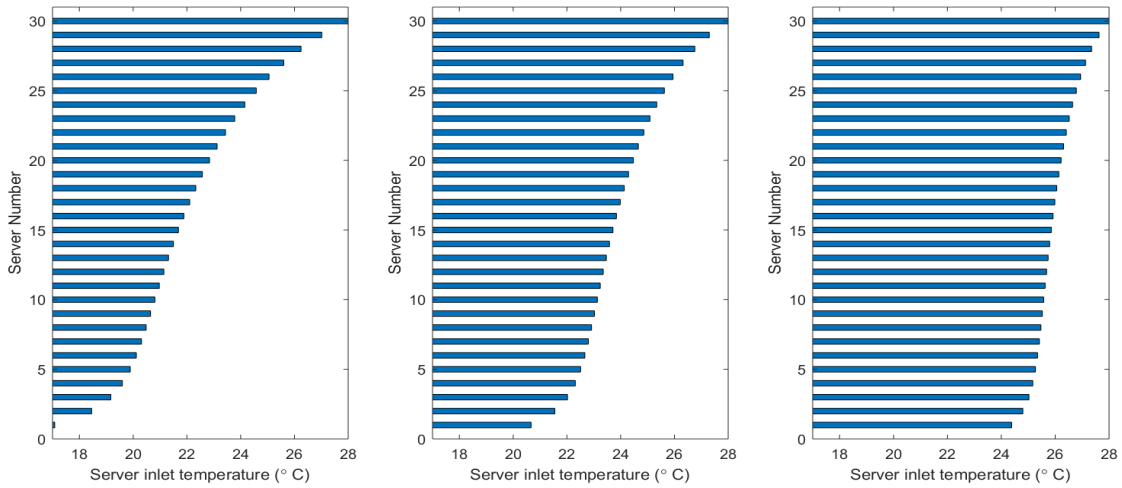
Trade-off in cooling parameters

As mentioned previously, a micro data center is our system under study. It consists of a rack that contains thirty homogeneous servers and an RMCU mounted at the bottom of the rack. Cold water provided by the chiller enters the RMCU. The RMCU is equipped with a number of fans that recirculate air in the rack. The total power consumption is given by equations (5.3.2) through (5.3.5).

Servers should be kept below a certain temperature, called the red line temperature (T_{red}^{server}). It is the maximum allowed inlet temperature that a server can tolerate (exceeding this temperature can affect a server's reliability). Using our notation, $max(\bar{T}_{inlet}^{server})$ must be less than or equal to T_{red}^{server}. The contributing operational parameters of the cooling unit are T_{inlet}^{water} and Q_{air}^{rmcu}. Adjusting these two factors can address this constraint, and the choice of parameters is not unique. For example, considering a utilization of 50% for all servers, either the tuple ($T_{inlet}^{water} = 12°C$, $Q_{air}^{rmcu} = 580cfm$) or ($T_{inlet}^{water} = 18°C$, $Q_{air}^{rmcu} = 710cfm$) can provide an appropriate temperature distribution at the server inlets. In both cases, the maximum temperature at the front of servers is set to $28°C$. Of course, a (large) number of other pairs of T_{inlet}^{water} and Q_{air}^{rmcu} can satisfy this condition. Figure 3.3 shows temperature distribution at the front of servers for different combinations of T_{inlet}^{water} and Q_{air}^{rmcu}, while servers are 50% utilized.

As shown in Figure 3.3, the cooling constraint is satisfied in all three cases. In this example, the cooling constraint is to set the maximum temperature at the front of servers (or $\bar{T}_{inlet}^{server}$) to $T_{red}^{server} = 28°C$. As air flow increases, the temperature variations at the front of servers decrease, making the average of $\bar{T}_{inlet}^{server}$ closer to T_{inlet}^{water}. Moreover, a higher T_{inlet}^{water} can be used as the air flow increases.

From the power consumption viewpoint, decreasing T_{inlet}^{water} increases chiller power consumption but requires less air circulation. On the other hand, increasing T_{inlet}^{water}, which lessens the power consumption of the chiller, must be compensated by increased Q_{air}^{rmcu} (greater P^{fan}). To study if there is a power saving opportunity in the trade-off between operational parameters of the cooling unit, the curve of power consumption of the cooling unit is drawn as a function of T_{inlet}^{water} (Figure 3.4). For each value of



(a) $T_{inlet}^{water} = 14°$C and $Q_{air}^{rmcu} = 619$cfm

(b) $T_{inlet}^{water} = 18°$C and $Q_{air}^{rmcu} = 692$cfm

(c) $T_{inlet}^{water} = 22°$C and $Q_{air}^{rmcu} = 785$cfm

Figure 3.3: The temperature distribution at the front of servers for different combinations of operational parameters of the cooling unit

T_{inlet}^{water} (which ranges from $10°C$ to $23°C$), Q_{air}^{rmcu} is adjusted in a manner such that $max(\bar{T}_{inlet}^{server})$ is set to T_{red}^{server} (the resolution of curves is $1°C$).

In Figure 3.4, three curves are plotted that correspond to three red line temperatures: $T_{red}^{server} = 26°C$, $27°C$ and $28°C$. The power consumption curves clearly show that there is an optimal setting for the operational parameters. For example, under $T_{red}^{server} = 28°C$, the power consumption of the cooling unit achieves a minimum of 3579W when $T_{inlet}^{water} = 19°C$ and $Q_{air}^{rmcu} = 890cfm$. This trade-off is due to the nonlinearity of the power consumption models for both the chiller and the fans.

To this point, we have demonstrated that the operational parameters of the cooling unit create a trade-off from the power consumption point of view and the reason for this trade-off has been discussed. However, this work is done for a particular choice of workload allocation (all servers 50% utilized). Different patterns of server

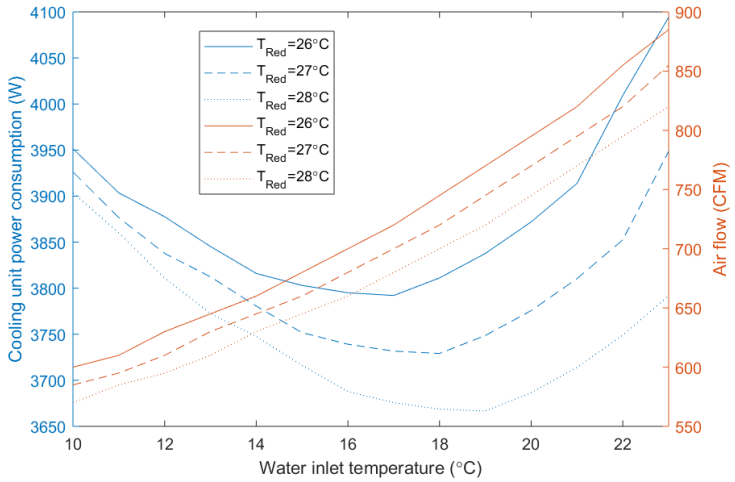


Figure 3.4: The trade-off between operational parameters of the cooling unit for different red line temperatures.

utilization result in different patterns of heat generation inside a rack and as a result the trade-off in choosing cooling parameters will be a function of the workload allocation. Therefore, in the next section, the thermal effects of heat source locations, or spatial distribution of workload and its potential incorporation with the operational parameters is studied, and an optimization problem is formulated that determines both the workload and operational parameters.

Workload distribution and optimality of operational parameters

In the previous section, the trade-off between the operational parameters of a data center cooling unit was investigated. This was performed in a scenario where the workload allocation was considered as an uncontrollable input. In this section, it is shown that assignment of workload is a contributing factor that affects the cooling unit efficiency.

As in the previous section, our data center consists of a single rack with thirty

homogeneous servers and an RMCU at the bottom of the rack (similar to Figure 3.1). Servers are numbered from 1 to 30, with the bottom server (closest to the RMCU) labeled 1 and the top server (furthest from the RMCU) labeled 30. In total, there are 30 single CPU servers.

The data center is tested under three different assignments of workload, with the constraint that half of the total computational capacity of the data center needs to be utilized, equivalent to 15 fully utilized servers. First, the workload is assigned to servers furthest from the RMCU. So, servers 1 to 15 are idle and servers 16 to 30 are fully utilized. Second, workload is distributed evenly between all servers and each server has utilization 50%. Third, workload is assigned to servers closest to the RMCU. Therefore, servers 1 to 15 are fully utilized and servers 16 to 30 are idle. These workload assignments are referred to respectively as the first, second and third assignment methods.

The power consumption of the cooling unit is calculated using T_{inlet}^{water} and the adjusted air flow-rate for each assignment of workload. The cooling air flow-rate or Q_{air}^{rmcu} is adjusted according to the given T_{inlet}^{water} to set the $max(\bar{T}_{inlet})$ to $T_{red}^{server} = 28°C$. The sum of the chiller power consumption (5.3.6) and the fans power consumption (5.3.5) is used to calculate the power consumption for the cooling unit.

Figure 3.5 shows the adjusted air flow-rate (red) as a function of T_{inlet}^{water} that satisfies temperature constraints. The power consumption of the cooling unit (blue), a function of both T_{inlet}^{water} and Q_{air}^{rmcu}, is plotted against the inlet water temperature. In this figure, T_{inlet}^{water} is constrained to take on integer values.

Three different line styles represent the three methods of assigning workload. The dotted curve represents the first, dashed represents the second, and solid represents

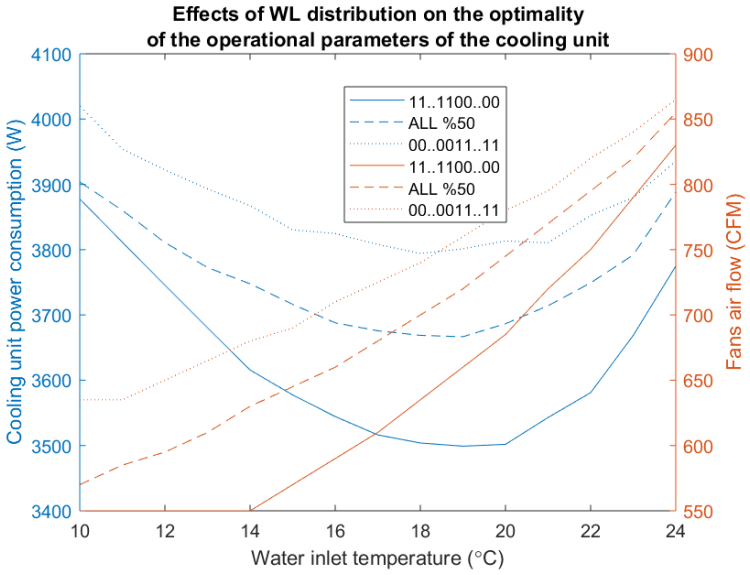
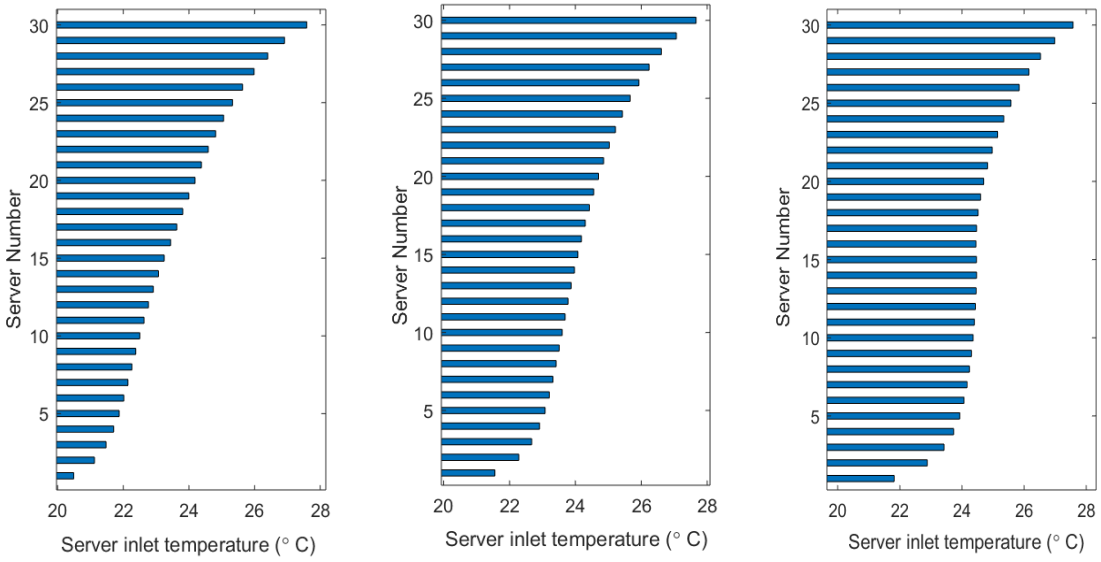


Figure 3.5: The trade-off between operational parameters of a cooling unit under different workload assignments.

the third method of workload assignment. The first insight is that the closer the workload is to the cooling unit, the less power is required to remove heat. Another insight is that the optimal values of the operational parameters are different under each assignment of the workload. For example, the minimum cooling power consumption when the first method of workload assignment is used requires $T_{inlet}^{water} = 18°C$ and $Q_{air}^{rmcu} = 740cfm$. However, the third workload assignment method requires $T_{inlet}^{water} = 19°C$ and $Q_{air}^{rmcu} = 660cfm$ to minimize the cooling unit's power consumption.

Analyzing the temperature distribution at the front of the servers helps to understand the difference in power consumption for the three workload assignment methods. Figure 3.6 illustrates the temperature distributions. In each figure, operational parameters are optimally adjusted to minimize cooling power with respect to the workload assignment method. Figure 3.6a corresponds to the first, Figure 3.6b to the



(a) First method of workload assignment

(b) Second method of workload assignment

(c) Third method of workload assignment

Figure 3.6: Temperature distribution at the front of servers. (a) Servers 1 to 15 are idle and 16 to 30 are fully utilized. $T_{inlet}^{water} = 18°C$, $Q_{air}^{rmcu} = 740cfm$, and $P^{cooling} = 3794W$. (b) All servers have utilization 0.5. $T_{inlet}^{water} = 19°C$, $Q_{air}^{rmcu} = 710cfm$, and $P^{cooling} = 3666W$. (c) Servers 1 to 15 are fully utilized and 16 to 30 are idle. $T_{inlet}^{water} = 19°C$, $Q_{air}^{rmcu} = 660cfm$, and $P^{cooling} = 3500W$.

second, and Figure 3.6c to the third workload assignment method.

In Figure 3.6, from left to right the assignment of the workload becomes closer to the cooling unit and the cooling power consumption becomes lower. Figure 3.6a shows that temperature variation is greater than in Figure 3.6b and Figure 3.6c, resulting in higher power consumption. We see that some servers are highly over-cooled. For example, the provided cool air for the first server, in Figure 3.6a, is about $20°C$ which is significantly lower than the red line temperature. As the workload becomes closer to the cooling unit, servers become less over-cooled and the result is a reduction in cooling power consumption. This figure clearly shows that the assignment of workload affects the temperature distribution and the best way of assigning workload is one

that leads to more uniform temperature distribution.

3.4 Formulating the optimization problem and comparing the results

In Section 3.3.2, the trade-off between cooling parameters was demonstrated. Next, in Section 3.3.2, the optimality of the cooling parameters as a function of the workload assignment was discussed. In general, we would like to determine the values of the cooling parameters and assignment of workload (or $\bar{u}$) to minimize the cooling power. In this section, a single optimization problem is formulated to return the optimal values of cooling variables and workload assignment. This problem is solved for two different data center configurations. We compare the results of our framework with a number of representative workload assignment methods which suggest that our method has the potential to yield significant improvements.

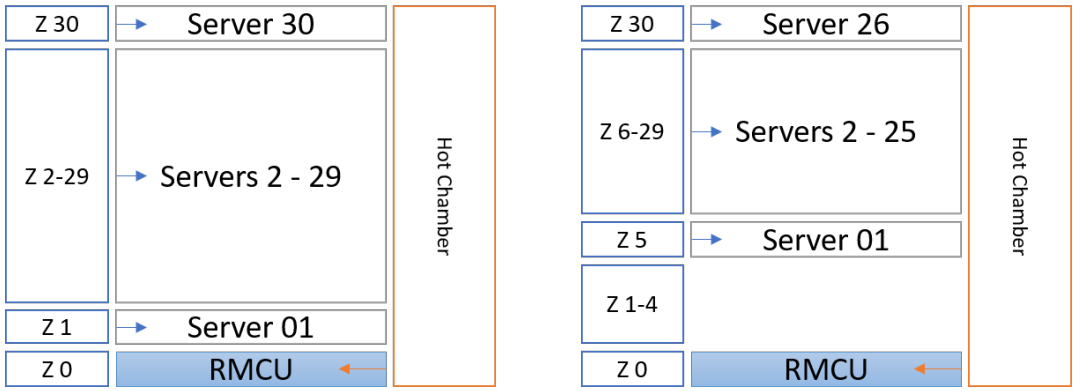
For our optimization problem, the decision variables are T_{inlet}^{water}, Q_{air}^{rmcu}, and $\bar{u}$. The goal is to find the optimal values of the variables to minimize the cooling power while maintaining the inlet temperatures of servers below the red line temperature T_{red}^{server}. The cost function is cooling power, $P^{fan} + P^{chiller}$, so the resulting optimization

problem is:

$$\begin{aligned} \underset{T_{inlet}^{water}, Q_{air}^{rmcu}, \bar{u}}{\text{minimize}} \quad & P^{fan} + P^{chiller} \\ \text{subject to} \quad & \sum_{i=1}^{n} u_i = d, \\ & 0 \leq u_i \leq u^{max},\ i = 1, \ldots, n \\ & max(\bar{T}_{inlet}^{server}) \leq T_{red}^{server} \end{aligned} \tag{3.4.1}$$

The optimization problem (3.4.1) is multidimensional with nonlinear constraints. u_i is the utilization of the i^{th} server while the sum of all u_is should be d, the offered workload. The second constraint allows for performance guarantees and the cooling constraint is enforced by the third constraint. Evaluating the third constraint requires the (nonlinear) thermal model to calculate $\bar{T}_{inlet}^{server}$. To solve the optimization problem the MATLAB function *fmincon* is used as it can support nonlinear constraints.

The optimization problem is solved for two different data center configurations. The thermal model described in Section 3.3.1 is used for both configurations. The first configuration has thirty servers stacked over each other on top of the RMCU (Figure 3.7a). However, the second configuration has 26 servers and an empty space just above the RMCU. The size of the gap is equivalent to four $1U$ servers (one U equals 1.75 inch or 44.45 mm). Moreover, it is assumed that all servers are homogeneous. The offered workload is considered to be half of the data center capacity. So, for the first and second configurations the offered workload d is 15 and 13, respectively. We assumed that utilization 100% would not hurt the performance of a single server; therefore, $u_{max} = 1$. The inlet temperature of servers should be capped by $T_{red}^{server} = 28°C$.

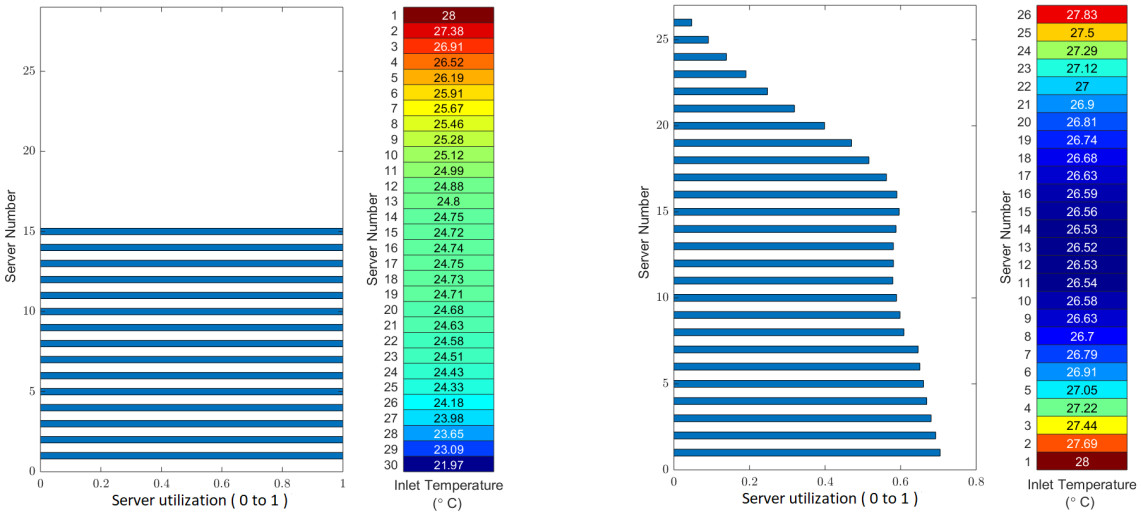


(a) Configuration 1 (b) Configuration 2

Figure 3.7: Data center schema for two configurations of the data center and servers

Solving the optimization problem (3.4.1) for each configuration returns the optimal values for the operational parameters and utilizations as shown in Figure 3.8. The figure depicts the assignment of the workload for both configurations with corresponding inlet temperatures of servers and is captioned by the optimal value of cooling parameters. For the first configuration (Figure 3.8a), servers 1 to 15 are fully utilized and servers 16 to 30 are idle. The results are as expected, servers closer to the cooling unit are easier to cool and thus assigning workload to them is more cost effective.

Although results for the first configuration suggest that the intuitive approach of locating workload as close as possible to the cooling unit is desirable, results for the second configuration are not as intuitive. This is because determining the proximity to the cooling unit is not as simple as observed in the first setting. The second configuration has more complicated air patterns and temperature distribution. This configuration suggests that simply assigning the workload to the servers that are in close proximity to the cooling unit does not always result in the optimal solution.



(a) Configuration 1 ($Q_{air}^{rmcu} = 564cfm$, $T_{inlet}^{water} = 14°C$)

(b) Configuration 2 ($Q_{air}^{rmcu} = 474cfm$, $T_{inlet}^{water} = 20°C$)

Figure 3.8: Optimized workload assignment and temperature distribution for both configurations

To better demonstrate the effectiveness of our proposed method, which we call joint cooling and workload management (JCWM), we compare it with four baseline and representative workload distribution algorithms: (1) TAWA, (2) TASA, (3) coolestInlets, (4) OnePass, and (5) Uniform distribution.

TAWA, or thermal-aware workload assignment, considers an HRM as a proxy for the thermal exchange or thermal model of a data center which is the core method of several works [4, 12, 20, 35]. Workload assignment in TAWA uses the HRM to determine the contribution of each heat source on the total accumulated heat for a thermal zone, aiming to minimize the peak inlet temperature.

TASA is a thermal-aware scheduling algorithm which allocates a new task to the server with the lowest CPU temperature [36]. *CoolestInlets* operates like TASA, but

makes decisions based on the inlet temperature of servers [19]. CoolestInlets is selected as, similar in philosophy to our approach, it considers server inlet temperatures as a key factor in satisfying thermal constraints. However, we will see that it does make substantially different decisions.

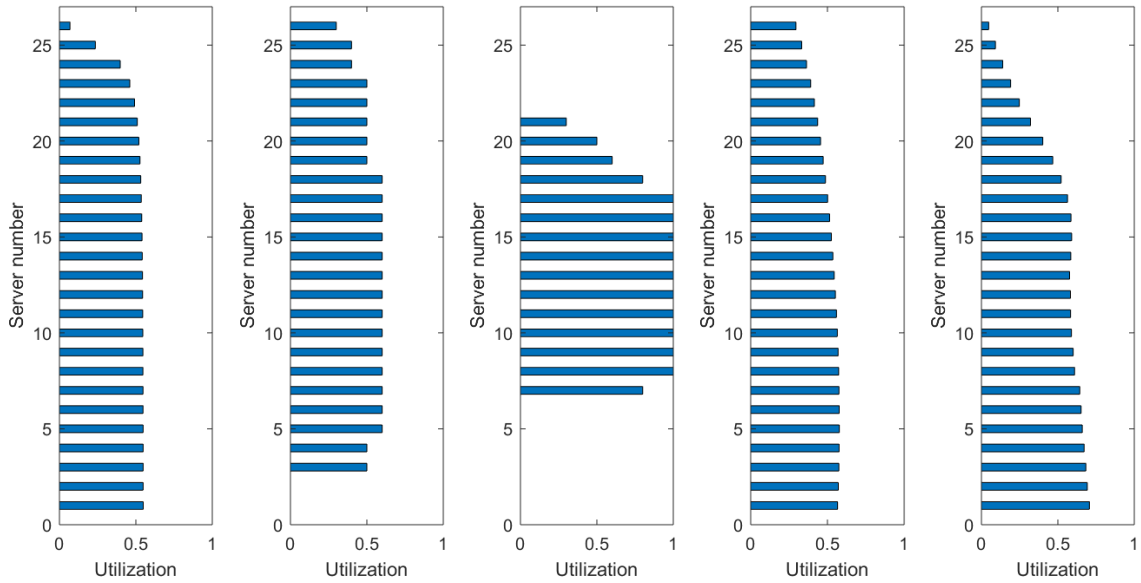
OnePass avoids generating hot spots by trying to set the exhausted air temperatures of servers to be as uniform as possible [17, 18]. Executing the algorithm requires a calibration phase where a uniform workload is assigned to all servers. This phase is for determining reference points for power (P_{ref}) and outlet temperature (T_{ref}^{out}) of a server or an average of multiple servers. Having the reference point, based on (3.4.2), the power can be determined for each server:

$$P_i = \frac{T_{ref}^{out}}{T_i^{out}} \cdot P_{ref}. \tag{3.4.2}$$

Finally, the last method for comparison to our algorithm is the Uniform workload assignment method. Uniform workload assignment is prefered in terms of response time performance [37, 38].

Figure 3.9 shows differences between various workload assignment decisions. Based on these workload assignments, we observe that JCWM makes somewhat different decisions. JCWM does not assign workload to the coolest server (TASA), or the server with the lowest inlet temperature (CoolestInlets). It also does not minimize the heat recirculation (TAWA) or variance of the outlet temperatures of servers (OnePass). Obviously, JCWM does not distribute workload evenly between all servers (Uniform).

Figure 3.10 shows the temperature distribution at the front of each server for each workload assignment method and the corresponding power consumptions. As expected, JCWM by optimal assignment of operational parameters and workload



(a) TAWA (b) TASA (c) CoolestInlets (d) OnePass (e) JCWM

Figure 3.9: Optimal utilization for different workload assignment methods

achieves the lowest cooling power amongst all of the methods. The figure clearly shows that JCWM over-cools servers less than other methods. In this specific configuration, JCWM decreased the total power consumption of the cooling unit by 11% compared to its closest competitor (Uniform). It is worth mentioning that due to the linear model of power consumption for the servers, P^{it} is independent of the workload assignment.

3.5 Thermal effects of server consolidation

In the previous section, the potential cost saving opportunities through optimizing operational parameters of the cooling unit and assignment of workload were investigated. An important assumption is that if a server does not receive any workload,

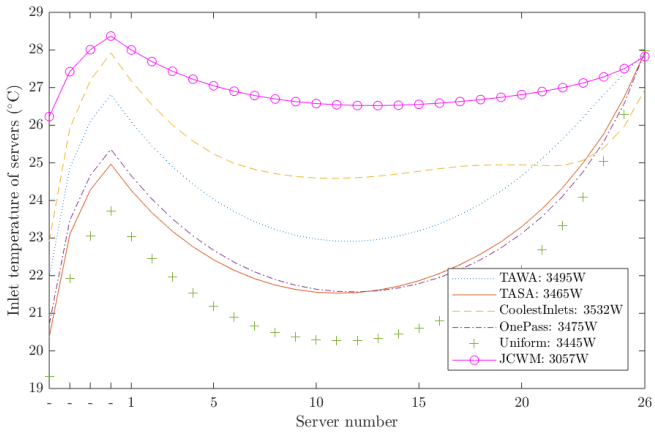
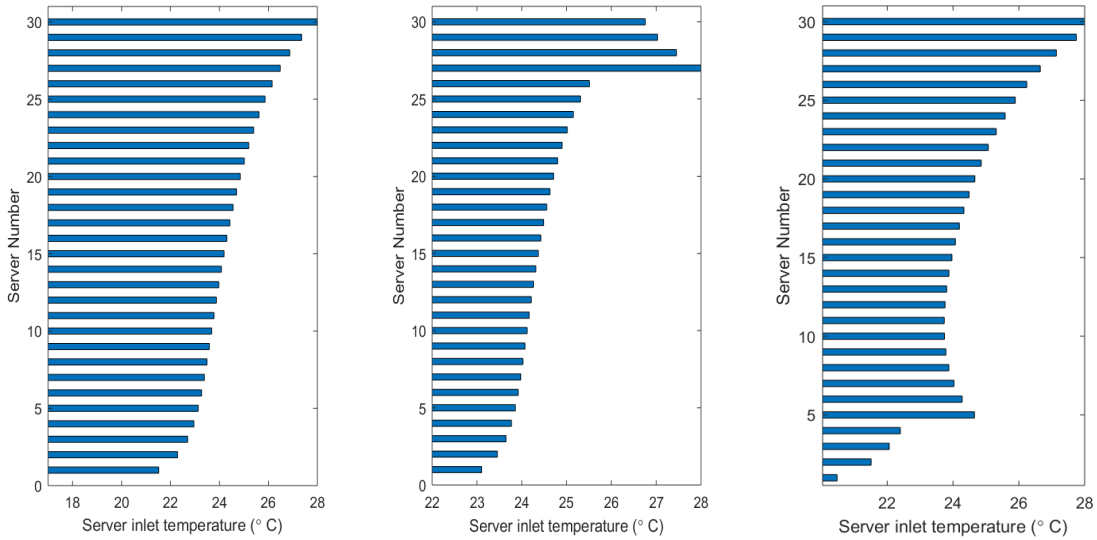


Figure 3.10: Temperature distribution of servers for different methods of workload assignment

it just remains idle and is not turned off. Server consolidation techniques turn unneeded servers off, and have been the subject of many studies [20, 29, 30, 39]. There are many debates on the power and performance balance of this method. However, there is almost no work related to this topic that investigates the thermal effects of turning servers off. We suggest that there is the potential for some adverse thermal effects, if the choice of servers to be consolidated is not made wisely. In this section, we show that neglecting the thermal effects of server consolidation could cause extra cooling efforts and consequently greater cooling power consumption. However, savings in server consolidation can be gained through turning servers off and generating less heat. So, the possibility of a trade-off when considering server consolidation is demonstrated and discussed.

Figure 3.11 shows the temperature distribution at the front of servers in three different configurations. In all configurations, the same workload is distributed between servers, with the offered workload being $d = 20$. Operational parameters are optimized to satisfy the cooling constraint ($T_{red}^{server} = 28°C$) while minimizing the cooling



(a) All server are on (b) Server 29 is off (c) Server 5 is off

Figure 3.11: Thermal effects of server consolidation, (a) $T_{inlet}^{water} = 19°C$ $Q_{air}^{rmcu} = 724cfm$ $P^{cooling} = 3698W$, (b) $T_{inlet}^{water} = 21°C$ $Q_{air}^{rmcu} = 771cfm$ $P^{cooling} = 3828W$, (c) $T_{inlet}^{water} = 18°\text{C}$ $Q_{air}^{rmcu} = 643cfm$ $P^{cooling} = 3669W$

power. In the first configuration, Figure 3.11a, no servers are allowed to be turned off and an equal amount of workload is assigned to each server, so $u_i = 0.667$ for $i = 1$ to 30. The second configuration, Figure 3.11b, the 29^{th} server is turned off; without this server, the data center is still capable of processing the offered workload. The CPU utilization of all servers, excluding server 29, is 0.69. The third configuration is the same as the second configuration, but the 5^{th} server is off.

The figure shows that turning a server off might increase the cooling power (Figure 3.11b), or decrease it (Figure 3.11c). The minimum cooling power of each configuration is obtained based on assigned workload and optimized operational parameters to satisfy the red line temperature constraint. When all servers are on, Figure 3.11a, the power consumption of the cooling unit is 3698W. The second configuration, Figure 3.11b, exhibits a power consumption which is more than the first configuration, 3828W. The third configuration, Figure 3.11c, shows that the cooling power is the lowest amongst the three configurations.

Analyzing these results reveals that server consolidation is not always reducing the power consumption of the system. The main reason that justifies this phenomenon is that if a server turns off, it allows hot air from the back zone to leak to the front zone, which in turn impacts $\bar{T}_{inlet}^{server}$. In other words, if a server turns off, hot air recirculation alters, and $max(\bar{T}_{inlet}^{server})$ might exceed T_{red}^{server}. So, the cooling unit must work harder to compensate for this increase. As mentioned, this power increase might outweigh the power savings from turning a server off. On the other hand, turning off another server might decrease $P^{cooling}$ simply due to generating less heat and the altered pattern of hot air recirculating might have minimal impact. Therefore, not only performance degradation is a concern when turning unneeded servers off, but

adverse thermal effects of server consolidation should also be considered.

3.5.1 Workload assignment under server consolidation

The optimization problem in Section 3.4 assigns workload "as close as possible" to the cooling unit. However, this adjacency is related to the physics of heat transfer and air recirculation in a data center as opposed to simply the distance from the cooling unit. The previous section showed that server consolidation disturbs air recirculation and might affect cooling efficiency through altering the recirculation of air.

We give one example here of how workload assignment and cooling unit operational parameters can be simultaneously optimized given a particular choice for server consolidation. The architecture of the data center remains the same as in the previous section. However, it is assumed that servers 11 to 13 are turned off. It is assumed that the overall workload is $d = 13$. Solving the optimization problem (3.4.1) yields $T_{inlet}^{water} = 19.1°C$ and $Q_{air}^{rmcu} = 651.1cfm$. The optimal assignment of workload is illustrated in Figure 3.12. One interesting observation is that, being close to the cooling unit is not the optimal solution in this case.

Optimizing server consolidation

Turning a server off provides a means for hot air to leak to the front zone of servers. This effect has been investigated in the previous section. Both the number and location of off servers have significant impact on air recirculation and hence the power required to cool the data center. So, in addition to the operational parameters and distribution of the workload, the server consolidation policy itself should be part of the overall optimization problem. So far, we have found no means other than exhaustive

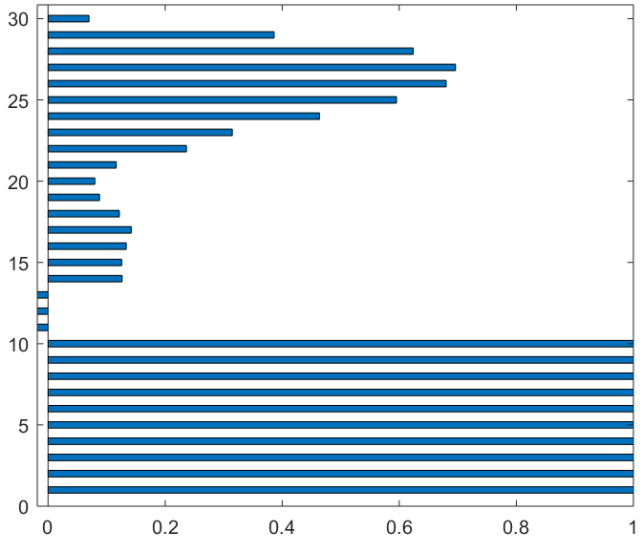


Figure 3.12: Heat recirculation inside a data center

search. We plan to perform further work on this problem, with the goal of presenting a heuristic solution that performs well for at least a large proportion of the design space.

3.6 Conclusion

We studied the detailed relation between cooling operational parameters and workload assignment and also highlighted synergies between simultaneously controlling the cooling parameters and workload assignment. Moreover, we have presented a novel approach to minimizing power consumption in data centers. The proposed approach jointly optimizes workload assignment and cooling unit operational parameters. Results have shown that the proposed joint optimization has the potential to save a

considerable amount of total cooling power when compared to other workload assignment algorithms. We have also shown that when consolidating servers care has to be taken as to which servers are being turned off as this might have an adverse affect on the power consumption due to hot air recirculation through turned off servers.

3.7 Acknowledgment

This research was supported by a Collaborative Research and Development grant CRDPI506142-16 from the Natural Science and Engineering Research Council of Canada (NSERC), and Cinnos Mission Critical Incorporated. We also acknowledge the comments of the referees, which resulted in improved exposition.

Bibliography

[1] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United states data center energy usage report," Tech. Rep. LBNL–1005775, 1372902, June 2016.

[2] R. Brown, "Report to congress on server and data center energy efficiency: Public law 109-431," Tech. Rep. LBNL-363E, Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States), Aug. 2007.

[3] H. Klemick, E. Kopits, and A. Wolverton, "Data center energy efficiency investments: Qualitative evidence from focus groups and interviews," tech. rep., National Center for Environmental Economics, US Environmental Protection Agency, 2017.

[4] Y. Li, X. Wang, P. Luo, and Q. Pan, "Thermal-aware hybrid workload management in a green datacenter towards renewable energy utilization," *Energies*, vol. 12, no. 8, p. 1494, 2019.

[5] M. Gupta and S. Singh, "Using low-power modes for energy conservation in ethernet lans.," in *INFOCOM*, vol. 7, pp. 2451–2455, 2007.

[6] N. Yadava, V. K. Mishra, and R. K. Chauhan, "Design of one-transistor SRAM cell for low power consumption," in *Proc. Int. Conf. Emerging Trends in Electrical Electronics Sustainable Energy Systems (ICETEESES)*, pp. 322–325, Mar. 2016.

[7] E. Aldahari, "Dynamic voltage and frequency scaling enhanced task scheduling technologies toward green cloud computing," in *4th Intl Conf Applied Computing and Information Technology/3rd Intl Conf Computational Science*, pp. 20–25, Dec. 2016.

[8] R. Ge, X. Feng, and K. W. Cameron, "Performance-constrained distributed DVS scheduling for scientific applications on power-aware clusters," in *Proc. ACM/IEEE SC 2005 Conf. Supercomputing*, pp. 34–45, Nov. 2005.

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

[10] 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.

[11] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. H. Katz, "NapSAC: Design and implementation of a power-proportional web cluster," in *Proceedings of the First ACM SIGCOMM Workshop on Green Networking*, Green Networking '10, (New York, NY, USA), pp. 15–22, ACM, 2011.

[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 *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, ATC'07, (Berkeley, CA, USA), pp. 29:1–29:6, USENIX Association, 2007.

[14] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[15] S. M. Mirhoseininejad, G. Badawy, and D. G. Down, "Eawa: Energy-aware workload assignment in data centers," in *2018 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 260–267, IEEE, 2018.

[16] X. Teng, H. Pham, and D. R. Jeske, "Reliability modeling of hardware and software interactions, and its applications," *IEEE Transactions on Reliability*, vol. 55, pp. 571–577, Dec 2006.

[17] 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.

[18] M. T. Chaudhry, T. Ling, S. A. Hussain, and A. Manzoor, "Minimizing thermal stress for data center servers through thermal-aware relocation," *The Scientific World Journal*, vol. 2014, 2014.

[19] 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*, pp. 61–75, 2005.

[20] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, "TACOMA: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Trans. Archit. Code Optim.*, vol. 9, pp. 11:1–11:37, June 2012.

[21] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, vol. 53, no. 17, pp. 2888–2904, 2009.

[22] Q. Wang, M. Song, Q. Fang, and J. Wang, "Thermal-aware flow field optimization for energy saving of data centers," in *2018 Annual American Control Conference (ACC)*, pp. 3744–3749, IEEE, 2018.

[23] Z. Wang, C. Bash, N. Tolia, M. Marwah, X. Zhu, and P. Ranganathan, "Optimal

fan speed control for thermal management of servers," *Proc. IPAC*, pp. 1–10, 2009.

[24] X. Zhao, Z. Xiong, L. Ding, X. Zhang, and F. Xu, "A smart coordinated temperature feedback controller for energy-efficient data centers," *Future Generation Computer Systems*, vol. 93, pp. 506–514, 2019.

[25] Q. Fang, Q. Gong, J. Wang, and Y. Wang, "Optimization based resource and cooling management for a high performance computing data center," *ISA transactions*, vol. 90, pp. 202–212, 2019.

[26] M. Zapater, A. Turk, J. M. Moya, J. L. Ayala, and A. K. Coskun, "Dynamic workload and cooling management in high-efficiency data centers," in *International Green Computing Conference and Sustainable Computing Conference (IGSC)*, pp. 1–8, IEEE, 2015.

[27] J. Yao, H. Guan, J. Luo, L. Rao, and X. Liu, "Adaptive power management through thermal aware workload balancing in internet data centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, pp. 2400–2409, Sept. 2015.

[28] 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, pp. 337–350, 2008.

[29] V. J. Maccio and D. G. Down, "Exact analysis of energy-aware multiserver queueing systems with setup times," in *2016 IEEE 24th International Symposium on*

Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 11–20, IEEE, 2016.

[30] V. J. Maccio and D. G. Down, "Asymptotic performance of energy-aware multiserver queueing systems with setup times," *The 2018 American Control Conference, ACC2018*, 2018.

[31] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "An energy-efficient VM prediction and migration framework for overcommitted clouds," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2016.

[32] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "Energy-efficient resource allocation and provisioning framework for cloud data centers," *IEEE Transactions on Network and Service Management*, vol. 12, pp. 377–391, Sept. 2015.

[33] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in it server enclosures," *International Journal of Heat and Mass Transfer*, vol. 127, pp. 890 – 900, 2018.

[34] M. Iyengar and R. Schmidt, "Analytical modeling for thermodynamic characterization of data center cooling systems," *Journal of Electronic Packaging*, vol. 131, no. 2, p. 021009, 2009.

[35] T. Van Damme, C. De Persis, and P. Tesi, "Optimized thermal-aware job scheduling and control of data centers," *IEEE Transactions on Control Systems Technology*, no. 99, pp. 1–12, 2018.

[36] L. Wang, S. U. Khan, and J. Dayal, "Thermal aware workload placement with

task-temperature profiles in a data center," *The Journal of Supercomputing*, vol. 61, no. 3, pp. 780–803, 2012.

[37] K. Chen, *Performance evaluation by simulation and analysis with applications to computer networks*. John Wiley & Sons, 2015.

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

[39] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010.

Chapter 4

ALTM: Adaptive learning-based thermal model for temperature predictions in data centers

This chapter is reproduced from "ALTM: Adaptive learning-based thermal model for temperature predictions in data centers", SeyedMorteza MirhoseiniNejad, Fernando Martínez García, Ghada Badawy, and Douglas G. Down, published in 2019 IEEE Sustainability through ICT Summit (StICT), pp. 1 - 6, IEEE, 2019.

The author of this thesis is the first author and the main contributor of this publication. His contributions to this work consist of introducing the idea of data-driven thermal modeling for data centers, writing the manuscript, constructing the algorithms, conducting the experiments, and implementing the framework.

Abstract

To design effective control schemes for energy efficiency in data centers, it is crucial to have a thermal model of the system. Constructing thermal models of data centers for temperature prediction is extremely challenging, due to inherent complexity. Computational fluid dynamics (CFD) simulations or physical heat transfer equations are conventionally used to construct such thermal models. More recent approaches combine physical heat transfer rules and data-driven methods in an effort to obtain more accurate models.

Our proposed adaptive learning-based thermal model (ALTM) is fast, adapts to thermal changes in the data center environment, and does not require prior knowledge of heat transfer rules between data center entities. Unlike other methods, ALTM is a holistic thermal model that predicts temperature of critical zones using data center operational variables as inputs. The operational variables are the controllable parameters and easily obtained measurements from IT and cooling units. A key use case for ALTM is that it can be effectively used for thermal-aware workload schedulers or cooling system controllers. Our results confirm the accuracy and adaptability of the model.

Keywords:thermal model, thermal-aware workload scheduling, data center temperature prediction, adaptive cooling control, neural network thermal model

4.1 Introduction

Air cooling systems continue to be the most common cooling systems in data centers. These can be simply building-designed coolers such as normal air conditioners (AC)

or conventional heating, ventilation, and air conditioning (HVAC) units. Many large scale data centers use computer room air conditioning (CRAC) units, in the form of a raised-floor architecture [1]. In-row cooling units and rack mountable cooling units (RMCUs) are more recent and power-efficient cooling system designs [2].

Cooling systems should provide sufficient cool air for servers. Maintaining the intake air of servers below a certain temperature ensures a safe working environment for servers and does not compromise their performance (due to automatic throttling of computing nodes [3]) or reliability [4]. The current practice of today's data centers is to keep the maximum temperature of a zone affected by a cooling unit below a certain temperature. Implementing this practice inevitably results in many servers being far below the required temperature, such servers are said to be over-cooled. Reducing over cooling of servers is an obvious opportunity for power savings [5]. The key component in achieving the minimum amount of over-cooling is to have a holistic thermal model. This model should give the distribution of air temperatures inside a data center based on the operational parameters of the cooling units and the heat generation profiles of servers [6].

A thermal model simply answers the question "what will be the temperature at the front of each server?". The answer should be in the form of a vector containing the temperature distribution, at a given future time. This can be obtained with respect to the current status of a data center, such as cooling unit configurations or the arrangement of heat sources (servers). Tracking server temperatures is crucial for operational control of cooling systems and server workload management.

There are a number of works and methods presenting thermal models of data

centers. In our previous work, we showed that using a holistic thermal model, a significant portion of the cooling power could be saved through an optimized assignment of workload and appropriate adjustment of cooling parameters [7]. Computational fluid dynamics (CFD) methods [8] can estimate the temperature of every point within a data center with high precision, however, they are very computationally intensive and are not appropriate for real-time decisions. There are a number of faster models using zonal-based methods and physical energy balance equations [6, 9]; however, these methods do not adapt with physical changes within data centers and also their accuracy deteriorates within large-scale settings. This is because determining incoming and outgoing air flows of thermal zones becomes very complicated in such chaotic environments.

In this paper, we present a means to predict the inlet temperatures of servers with high precision. This is a transient model (as opposed to steady state) that adapts to physical changes and can estimate the temperature over a time horizon. The inputs to our model are the operational parameters of the cooling unit and server workload assignment. With this in mind, we set up an infrastructure monitoring tool to provide the required data to predict future temperatures. We compared a linear regression (least squares) model and a neural network approach and concluded that an appropriate neural network can predict the temperature more accurately, further into the future. An important application of our work is as a key component for holistic system management to both schedule the incoming workload and control the cooling unit parameters efficiently in order to minimize total power consumption.

The next section provides a review of works related to temperature prediction in data centers. Next, the details of our experimental data center architecture and data

acquisition phase are explained. In Section (6.3.2), the framework to implement two model estimators is illustrated and discussed. Finally, results of implementing the framework are provided and analyzed.

4.2 Literature review

The literature lacks adaptive and/or practical solutions capturing all factors affecting air recirculation in a data center. *Computational fluid dynamics* (CFD) simulations are the predominant way of constructing thermal models for data centers. CFD simulations are based on thermodynamic laws and have heavy computational requirements. Although CFD methods have high precision and resolution, they cannot be evaluated at the time scales of data center dynamics [10].

The majority of works on thermal-aware workload assignment either simplified the effects of air recirculation using a static recirculation matrix [11, 12] or used a simple auto-regression method, simply based on IT load [13]. The drawback to these methods is that they have not considered the effects of all operational variables of a data center.

Moore et al. [14] used a neural network to compute the temperature of inlet air for all servers. The inputs of their model are pairs of power and heat profiles. Specifically, workload, cooling settings, and room layout measurements are used to train the model. However, it is a steady-state model that uses a limited number of influential parameters of the cooling unit as inputs to the model. So, the accuracy of the model can potentially be compromised by changes in parameters that have not been considered, such as a change in the air flow rates.

Zhang et al. [15, 16] developed a machine learning-based framework for temperature prediction of server cores. Several measurements of a running task are used as the features of the neural network model such as the CPU frequency, the number of instructions, floating-point operations, and cache hits or misses in different cache levels. Appropriate features are selected using a correlation feature selection (CFS) algorithm. They used this prediction model for application scheduling on different servers to reduce the maximum average core temperature.

Yao et al. [13] used a linear function that relates the outlet temperatures of IT-racks and CRACs to the inlet temperatures of IT-racks. $Y = WX$ is used as the linear model where X contains the outlet temperatures, Y contains the inlet temperatures, and W contains the weights. They used the recursive least squares (RLS) method to determine the weights (W) of the linear model.

Li et al.[17] presented an approach for energy-efficient thermal-aware workload scheduling. They used CFD methods to model the temperature distribution in a data center. Although it is asserted that the thermal model captures features of CRACs holistically, the simplified thermal model may be far from reality (for example, fan speeds are supposed to be constant, or the closest CRAC unit to a server is considered to be the only cooling unit that influences the server temperature) and the model lacks cooling unit details.

Li et al. [9] proposed ThermoCast, a thermal prediction model to predict temperatures in a data center, based on temperature and air flow measurements. This approach considers the most recent measurements of IT power consumption, temperature, and air flow rates to update the model coefficients. The main issue with this method is that the model requires a structure based on physical laws. They simplify

the air flow equations to obtain the structure. So, errors due to simplification may propagate for large scale data centers. The other issue is that the model uses air flow measurements at the front of servers, which we have found to be problematic as directly controllable (or measurable) variables.

4.3 Thermal model

The main objective of constructing a thermal model is the estimation of the temperature distribution within a data center. The model should be able to predict the inlet temperatures of servers based on the operational parameters of cooling units and the data center workload. We start with illustrating outputs and inputs of the model. The outputs are the temperatures of thermal zones. A zone is the cubical volume at the front of a number of adjacent servers. For example, the data center shown in Fig. 5.1 has 25 thermal zones. Adjacent servers typically have small differences in their inlet temperatures; as a result, we use the inlet temperature of a server and the temperature of a zone interchangeably throughout this paper. Inputs of the model are manipulatable or controllable variables which here are workload profiles and cooling profiles; the former is related to the IT facilities and the latter corresponds to the cooling facilities.

Workload profile For the sake of simplicity, this work simply considers the workload of a server to be its utilization.

Cooling profile The cooling profiles are the set of dynamic variables that can be measured and controlled and also affect the temperature distribution.

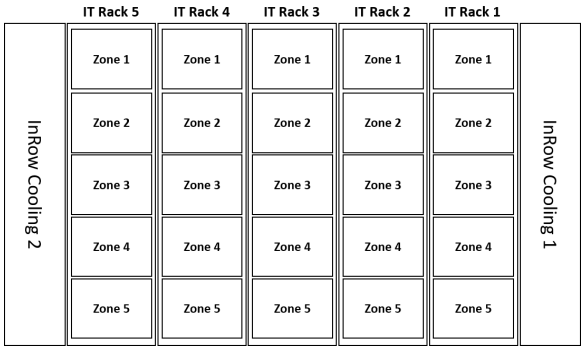


Figure 4.1: Front view of an in-row cooling data center with two cooling units at two sides and five IT racks

In this paper, we show that a reasonably accurate temperature prediction is gained using the suggested framework with the help of readily available inputs. The implementation of the framework is straightforward and there is no need for understanding the physics of the heat transfer within the data center. The determination of server inlet temperature estimates is both on-line and adaptive. To the best of our knowledge, this is the first thermal modeling approach that directly uses cooling and IT parameters for its predictions and adapts to changing thermal conditions. Changes in the thermal condition of a data center are to be expected. These changes can be initiated from component changes due to system maintenance, room alterations, device replacements, dust accumulation, modifications of the compartments and air vents, etc. An example later in the paper shows the necessity of being adaptive.

Next, the procedure for data acquisition for model estimation is described in detail. We then show the implementation of on-line model estimators and describe a framework for using them. Finally, we discuss the accuracy of using different model estimators, and illustrate the results.

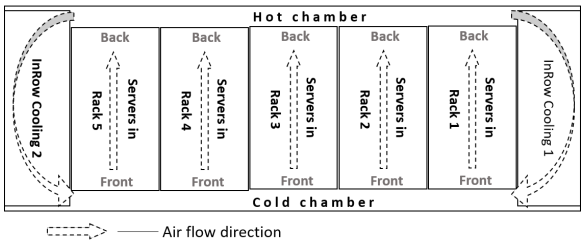


Figure 4.2: The top view of the in-row cooling data center

4.3.1 Data acquisition

An important aspect of this work was setting up equipment and reporting tools to acquire data. The setup was implemented in a data center which has two in-row cooling units at two sides and five IT racks (Fig. 5.1). We developed a data acquisition tool to both apply our desired configurations and acquire all operational variables of cooling units and server profiles. Fig. 5.2 shows the top view of the data center under study, consisting of two major parts: IT and cooling units. IT is considered to be the servers and cooling units include the facilities that provide cool air at the front of servers.

Fig. 5.3 shows the architecture of each cooling unit. As shown, each cooling unit has a number of fans that draw hot air from the hot chamber, pass the air through a heat exchanger and blow the cold air to the cold chamber. Water flow within the heat exchanger transfers the generated heat out of the facility. In other words, cold water enters the heat exchanger, and warm water exits.

Cooling unit operational parameters can be controlled and monitored using the *simple network management protocol* (SNMP); these parameters include the speed of each fan and the water flow rate inside the heat exchanger of the cooling unit. On the other hand, the IT consists of servers that process the given workload. We can apply

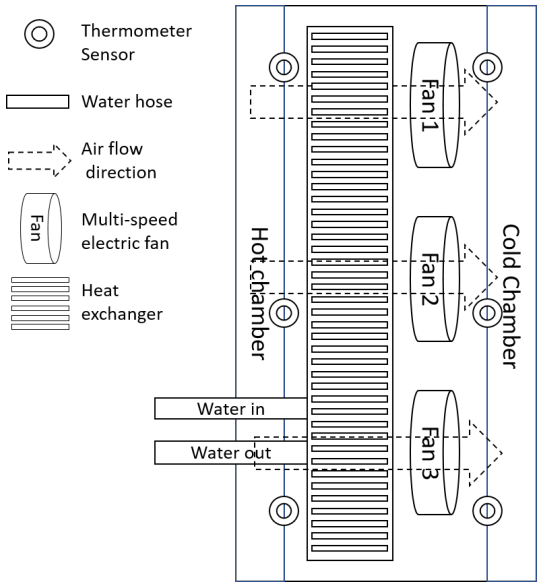


Figure 4.3: In-row cooling schema

the given workload to servers and collect real-time reports using SSH commands. Each server is able to report the current utilization and temperature of its cores. Temperatures at the front of servers obtained via thermal sensors (DS18B20 digital thermometers) which are placed in each zone. The height of each rack is divided into five equal height thermal zones.

Our designed tool connects to cooling units using SNMP, to servers by SSH, and to thermal sensors via serial ports. It takes operational scenarios as an input. A scenario is a time series of values that needs to be applied to the controllable variables of the data center at specified times. The operational scenario should be rich in parameter variation to be suitable to train the model. Upon executing a scenario workload patterns are applied to servers and patterns of operational parameters are applied to cooling units. At the same time, reported data including measurements from the thermometers (installed at 25 thermal zones), the utilization and CPU temperature of servers, and operational parameters of the two cooling units are saved in a data base.

The operational parameters of the cooling units consist of the inlet water temperature (T_{inlet}^{water}), the water flow (Q^{water}) and fan speeds (S_i^{fan}).

Gathered data is preprocessed and saved into corresponding matrices, input $\mathbf{X}$ and output $\mathbf{Y}$, to be applicable for a model estimator. Each row of $\mathbf{X}$ includes all input variables and each row of $\mathbf{Y}$ includes all output measures are obtained at the same time step. The values in $\mathbf{X}$ and $\mathbf{Y}$ are normalized to be in comparable scales. A bold capital letter, such as $\mathbf{Y}$, represents a matrix and a bold small letter, such as $\mathbf{y}_i$, denotes a vector corresponding to the i^{th} row of a matrix.

A top-level view of the thermal model can be formulated as (4.3.1). This is a transient model that predicts the inlet temperatures of servers at the next time step, shown by $\hat{\mathbf{y}}_{k+1}$. So, the model is a discrete function of the current and previous inputs and outputs. The output vector is represented by $\mathbf{y}_k$ and $\mathbf{x}_k$ is the input vector, both at the k^{th} time step. The vector $\mathbf{x}_k$ consists of the operational parameters of cooling units, $\bar{S}^{fan}$, $\bar{Q}_{water}$ and $\bar{T}_{in}^{water}$, and the workload profile given by server utilizations ($\bar{U}$). The vectors $\mathbf{x}_{k-i}$ and $\mathbf{y}_{k-i}$ are the input and output in the i^{th} previous time step.

$$\hat{\mathbf{y}}_{k+1} = f(\mathbf{x}_k, \mathbf{x}_{k-i}, \mathbf{y}_k, \mathbf{y}_{k-i}) \;\; i = 1, 2, \cdots \tag{4.3.1}$$

4.3.2 The model framework and algorithms

We explored two different approaches for temperature estimation. The first uses *weighted recursive least squares* (wRLS) for the estimation of a linear model, and the second trains a neural network model. We selected two off-the-shelf adaptive model estimators; one of them works well for a linear system and the other can better

model a non-linear system. We arranged to make a fair comparison between them by exposing them to the same input data.

Weighted recursive least squares We used wRLS for the parameter estimation of a linear thermal model [18]. Without loss of generality, the problem can be considered as a simple linear model $\mathbf{Y} = \mathbf{\Phi X}$; here $\mathbf{X}$ and $\mathbf{Y}$ are matrices of inputs and measured output values, respectively. wRLS is an on-line model estimator which is able to adapt to changes in the system being estimated. The algorithm forgets the past data using a forgetting factor λ.

wRLS has an update phase that updates the model parameters (or $\mathbf{\Phi}$) upon receiving new data. For the sake of simplicity, we do not explain the wRLS process. We just denote the update phase in the form of $\mathbf{\Phi}_{new} = parameterUpdate(\mathbf{\Phi}_{old}, \mathbf{X}, \mathbf{Y}, \lambda)$. Here, $\mathbf{\Phi}_{new}$ is the newly calculated model parameters with respect to $\mathbf{\Phi}_{old}$ and updated $\mathbf{X}$ and $\mathbf{Y}$. $\mathbf{\Phi}_{old}$ is the latest calculation of the model parameters. The matrices $\mathbf{X}$ and $\mathbf{Y}$ are updated with new data during each iteration.

The wRLS algorithm considers a number of previous data samples p, often referred to as a p^{th} order filter. So, a window of length p is updated with the most recent data samples. $\mathbf{X}$ is a p by i matrix, $\mathbf{Y}$ is a p by o matrix, and $\mathbf{\Phi}$ is an i by o matrix in which i is the number of linear terms of the input and o is the number of outputs being estimated.

Algorithm 2 gives a simple form of wRLS to estimate the linear thermal model. In this algorithm, one iteration is performed upon receiving a new vector of data $\mathbf{d}$. The function *dataGeneration()* returns the vector of new samples of inputs and the corresponding outputs. The new data $\mathbf{d}$ is used to update input and output matrices using *dataInsertion()*. Finally, *parameterUpdate*(), using the new matrices of inputs

and outputs, updates the previously obtained parameters in $\mathbf{\Phi}$.

Result: Estimation of the linear thermal model
$\mathbf{X}=[0]_{p,i}$;
$\mathbf{Y}=[0]_{p,o}$;
$\mathbf{\Phi}=[0]_{i,o}$;
λ=0.9;
while *true* **do**
  $\mathbf{d}$ = dataGeneration();
  [$\mathbf{X}$ $\mathbf{Y}$]=dataInsertion($\mathbf{X}$,$\mathbf{Y}$,$\mathbf{d}$);
  $\mathbf{\Phi}$=parameterUpdate($\mathbf{\Phi}$,$\mathbf{X}$,$\mathbf{Y}$,λ);
end

Algorithm 2: Adaptive linear thermal model

Neural Networks The second method is training an adaptive neural network for the thermal model. For the neural network model, we used a MATLAB toolkit in which the standard back-propagation method uses the Levenberg-Marquardt algorithm to train the model. Our job is to see how well an off-the-shelf neural network performs. As a result, analyzing and comparing different neural network methods is out of the scope this paper. However, it is certainly an interesting topic for future work.

As explained previously, the model should be updated as time progresses. There are a number of methods that consider updating neural networks upon system changes [19]. For example, an update can be performed upon detecting a notable mismatch between the desired and estimated data. We chose the statistical batch selection method for updating [20], as it is straightforward to implement for our scenario.

Statistical batch selection updates the neural network model upon receiving a number of new data points. Randomly generated numbers are used as indexes to select data samples from the previously saved data. The batch selection approach is

more likely to return recent data for the next iteration of the algorithm. To implement the adaptive neural network and batch selection method, we used Algorithm 3. The algorithm stores the recent data in a buffer of length i. It then selects the batch of data using the function *batchSel*(), as described. This batch is used to train the new neural network. The network uses the previous iteration weights and biases.

Result: Estimation of the neural network model
$\mathbf{X}=[0]_{p,i}$;
$\mathbf{Y}=[0]_{p,o}$;
$\mathbf{\Phi} = initialize()$;
$l = 10$;
$n = 1$;
$net = backpropagation(n)$;
while *true* **do**
  $\mathbf{D} = dataGeneration(l)$;
  $[\mathbf{X}, \mathbf{Y}] = dataInsertion(\mathbf{X}, \mathbf{Y}, \mathbf{D})$;
  $[\mathbf{X_b}, \mathbf{Y_b}] = batchSel([\mathbf{X}, \mathbf{Y}])$;
  $train(net, \mathbf{X_b}, \mathbf{Y_b}, \mathbf{\Phi})$;
  $\mathbf{\Phi} = net.weights()$;
end

Algorithm 3: Adaptive neural network thermal model

Algorithm 3 first initializes input and output data windows ($\mathbf{X}$ and $\mathbf{Y}$), and the internal neural network weights ($\mathbf{\Phi}$) using the function *initialize*(). It requires a specific number of data samples (l) at the beginning of each iteration. In our implementation, we set l to be 10. We chose one hidden layer and the back-propagation method for the neural network (n).

In the loop, after receiving a certain number of data points (l), the new data samples ($\mathbf{D}$) are inserted in the data window [$\mathbf{X}$, $\mathbf{Y}$] and the outdated data points are discarded from the window. After constructing [$\mathbf{X}$, $\mathbf{Y}$], the batch [$\mathbf{X_b}$, $\mathbf{Y_b}$] of selected data is constructed by the *batchSel()* function. The neural network is then trained

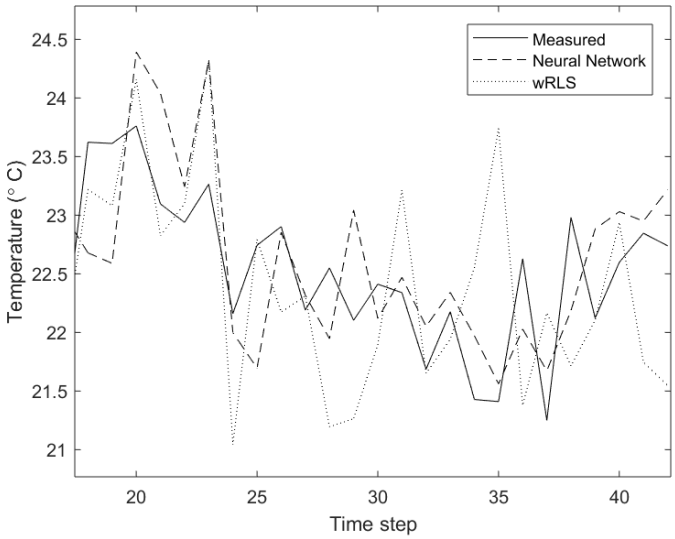


Figure 4.4: Temperature prediction of the neural network vs wRLS models

using the selected batch and the previously calculated network weights.

4.4 Results

We first compare the estimation results of the linear and neural network models. For the neural network the accuracy for the validation set is set to $0.001°C$. The termination of the neural network training happens after 9 epochs, on average. The neural network computational complexity was not limiting for our settings, however, this aspect should be be studied in the future. Fig. 4.4 depicts the estimation horizon of the neural network and a linear model. Curves represent the average temperature of the 25 temperature sensors. The solid line is the value of measurements and non-solid lines are the estimates. The figure shows that the neural network model has greater accuracy than the linear model.

To demonstrate the accuracy of the neural network model, the measured and

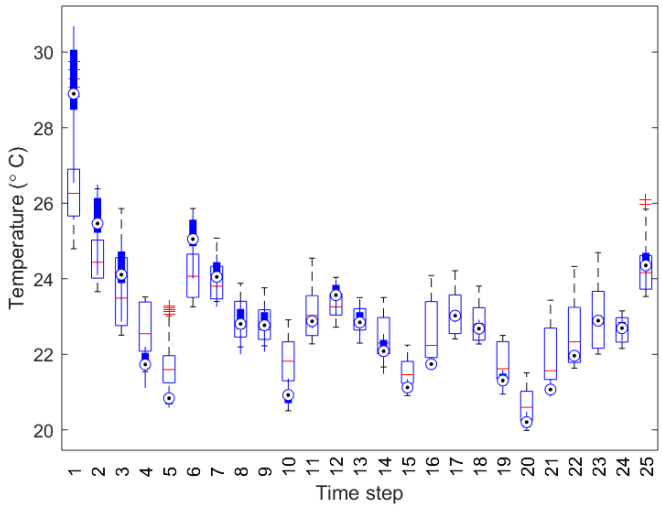


Figure 4.5: The box plot representation of the 25 measured temperatures vs their estimates (blank rectangles show the measured values and filled blue rectangles are estimates of the temperatures using neural networks)

estimated values are shown in the same plot. A box plot representation is chosen to plot 25 estimates and 25 measured values at each time step. For each box, the average is indicated by the central mark. The 75^{th} and the 25^{th} percentiles are shown by the top and the bottom edges of each box, respectively. In Fig. 4.5, the blank rectangles with red central marks show the measured values and filled blue rectangles with the central circle marks are the model estimates. The figure shows that the estimates follow the measured values accurately enough. The average estimation error for the 100 time step projection is $1.5°C$.

The neural network model is designed to be adaptive to changes that might occur in thermal conditions. We performed an experiment to demonstrate the adaptivity of the thermal model. We introduced thermal changes at the 1550^{th} time step. At that time, the front doors of the cooling units were left partially open, having been closed

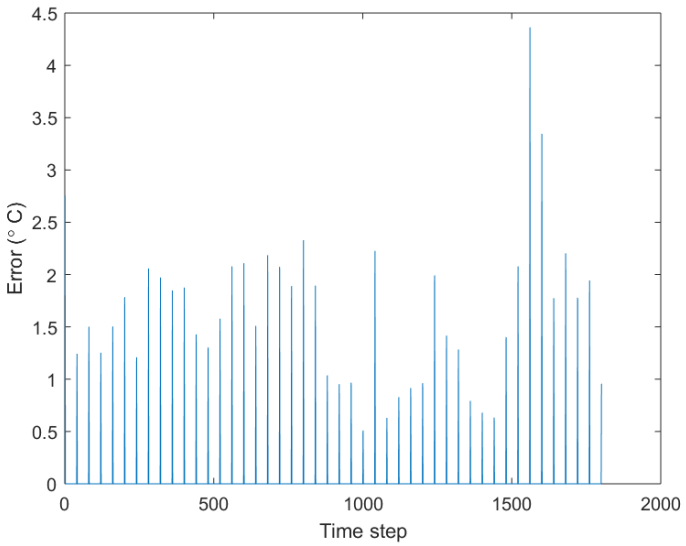


Figure 4.6: 100 steps projection error for the neural networks model - An environmental change happened at the 1550^{th} time step

before the 1550^{th} time step. Fig. 4.6 shows that at the time of the change a large error occurs in the estimates. The model then adapts to the new thermal conditions and the error decreases.

In Section 5.2, CFD models and a number of physics-based thermal models were reviewed. It was stated there that the main issue with using these is that none of them are adaptive to the thermal changes in the data center environment. Fig. 4.7 shows the behavior of an adaptive and a non-adaptive neural network model. The figure clearly demonstrates the difference between these two models. The average error for the adaptive model is $1.15°C$ and for the non-adaptive model is $2.1°C$. The errors of non-adaptive models can potentially diverge for longer prediction horizons, so these errors are exacerbated as time increases.

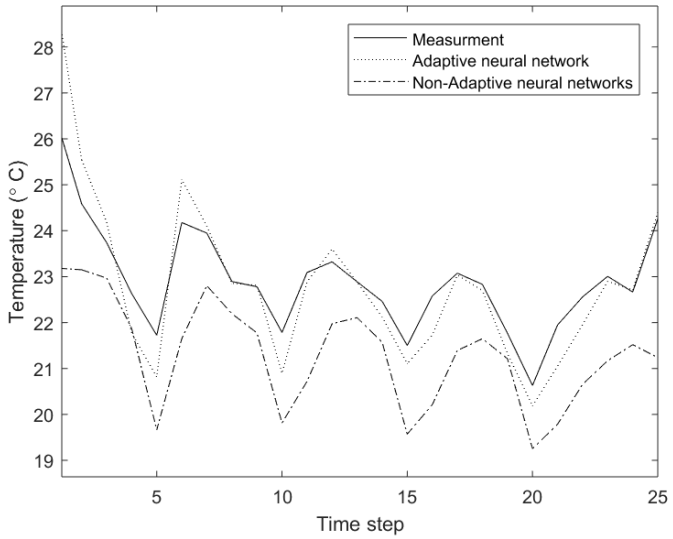


Figure 4.7: The comparison between adaptive and non-adaptive thermal models

4.5 Conclusion

We introduced a novel, low-complexity, easy to implement, and adaptive model estimator which captures the thermal dynamics of a data center. It can be applied in any data center and provides up-to-date information that could be used by a thermal-aware workload manager. The model is also attractive because it only requires readily available inputs. Other means of constructing thermal models have some deficiencies. Many of them are just fixed models that do not change with the changes within a data center, which is a serious drawback due to the dynamic nature of data centers. Some suggested adaptive thermal models do not consider the cooling infrastructure at the same level of detail as we have. Considering every operational variable of the cooling units provides the opportunity of controlling cooling together with the assignment of workload which can lead to significant power savings. Our adaptive thermal modeling approach appears to be an attractive option to incorporate into workload schedulers

or control algorithms.

Bibliography

[1] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 622–638, 2014.

[2] A. Capozzoli and G. Primiceri, "Cooling systems in data centers: state of art and emerging technologies," *Energy Procedia*, vol. 83, pp. 484–493, 2015.

[3] W. L. Bircher and L. K. John, "Complete system power estimation using processor performance events," *IEEE Transactions on Computers*, vol. 61, no. 4, pp. 563–577, 2012.

[4] X. Teng, H. Pham, and D. R. Jeske, "Reliability modeling of hardware and software interactions, and its applications," *IEEE Transactions on Reliability*, vol. 55, no. 4, pp. 571–577, 2006.

[5] S. M. M. Nejad, G. Badawy, and D. G. Down, "Eawa: Energy-aware workload assignment in data centers," in *2018 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 260–267, IEEE, 2018.

[6] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in it server enclosures," *International Journal of Heat and Mass Transfer*, vol. 127, pp. 890–900, 2018.

[7] S. M. M. Nejad, H. Moazamigoodarzi, G. Badawy, and D. G. Down, "Joint data center cooling and workload management: A thermal-aware approach," in *Future Generation Computer Systems*, FGCS, 2019.

[8] C. D. Patel, C. E. Bash, C. Belady, L. Stahl, and D. Sullivan, "Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications," in *Proceedings of IPACK*, vol. 1, pp. 8–13, 2001.

[9] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos, "Thermocast: a cyber-physical forecasting model for datacenters," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1370–1378, ACM, 2011.

[10] X. Wang, X. Wang, G. Xing, J. Chen, C.-X. Lin, and Y. Chen, "Intelligent sensor placement for hot server detection in data centers," *IEEE Transactions on parallel and distributed systems*, vol. 24, no. 8, pp. 1577–1588, 2013.

[11] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, "TACOMA: Server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Trans. Archit. Code Optim.*, vol. 9, pp. 11:1–11:37, June 2012.

[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] J. Yao, H. Guan, J. Luo, L. Rao, and X. Liu, "Adaptive power management

through thermal aware workload balancing in internet data centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 9, pp. 2400–2409, 2015.

[14] J. Moore, J. S. Chase, and P. Ranganathan, "Weatherman: Automated, online and predictive thermal mapping and management for data centers," in *2006 IEEE International Conference on Autonomic Computing*, pp. 155–164, IEEE, 2006.

[15] K. Zhang, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, and P. Beckman, "Minimizing thermal variation across system components," in *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, pp. 1139–1148, IEEE, 2015.

[16] K. Zhang, A. Guliani, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, and P. Beckman, "Machine learning-based temperature prediction for runtime thermal management across system components," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 2, pp. 405–419, 2018.

[17] X. Li, P. Garraghan, X. Jiang, Z. Wu, and J. Xu, "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1317–1331, 2018.

[18] S. Van Vaerenbergh, I. Santamaría, and M. Lázaro-Gredilla, "Estimation of the forgetting factor in kernel recursive least squares," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, IEEE, 2012.

[19] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[20] I. Loshchilov and F. Hutter, "Online batch selection for faster training of neural networks," *arXiv preprint arXiv:1511.06343*, 2015.

Chapter 5

Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning

This chapter is reproduced from "Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning", Seyed-Morteza MirhoseiniNejad, Ghada Badawy, and Douglas G. Down, submitted to Future Generation Computer Systems, 2020. The author of this thesis is the first author and the main contributor of this publication. His contributions to this work consist of introducing the idea of a holistic thermal-aware framework for data centers, writing the manuscript, formulating the optimization problems, conducting the experiments, and implementing the framework.

Abstract

Two key contributors to the energy expenditure in data centers are information technology (IT) equipment and cooling infrastructures. The standard practice of data centers lacks a tight correlation between these two entities, resulting in considerable power wastage. Considering the cooling cost of different locations inside a data center (cooling heterogeneity) and various cooling capabilities of servers (server heterogeneity) has significant potential for saving power, yet has not been studied thoroughly in the literature. There is a necessity for state-of-the-art approaches to integrate the control of IT and cooling units. Moreover, the literature still lacks an accurate and fast thermal model for temperature prediction inside a data center. In this paper, innovative approaches to create thermal models for data centers and servers are presented, which quantify data center thermal heterogeneities. Employing the models, the cost of providing cold air at the front of servers can be (indirectly) calculated, and the capability of servers to be cooled is formulated. Our approach assigns jobs to locations that are efficient to cool (from the perspectives of both servers and cooling units) and tunes cooling unit parameters. The method, called holistic data center infrastructure control (HDIC), has the potential to save a considerable amount of power by exploiting synergies between the workload scheduler and operational parameters of cooling units.

Keywords: data center workload assignment, cooling unit control, thermal-aware scheduling, neural network modeling, data center model, efficient cooling, server thermal model

5.1 Introduction

Two percent of power consumption in the United States in 2014 was due to data centers, equivalent to approximately 70 billion kWh [1]. In contrast, the power consumption of data centers in 2000 was 30 billion kWh [2]. It has been estimated that from 2015 to 2020, the incoming load to data centers will double [3]. The increasing number of online and mobile applications, public interest to access cyber entertainment, and cloud services for both personal and business users have a significant role in this jump [4]. Anticipating this increase, in addition to power usage constraints have led large data center vendors to invest more in the efficient use of power [1].

There are several methods and techniques to reduce power consumption at different levels of a data center. At the device level, some electronic devices support low power states to save energy, if the performance of the device is not impacted [5, 6]. For example, dynamic voltage and frequency scaling (DVFS) is a method that provides different levels of power consumption and performance for processors [7, 8]. At the server level, dynamic suspension of unneeded servers, server consolidation, and the ability to choose different levels of power and performance are vital approaches for energy efficiency. For instance, server consolidation aims to save power by turning unneeded servers off during low workload periods [9, 10, 11]. At the facility level, power efficiency of the cooling system itself is also a significant concern [12, 13, 14].

Different servers and locations in data centers are not cooled equally, resulting in what we call data center thermal heterogeneity. In other words, servers are different in their cooling requirements (server heterogeneity), and locations are also different in their cooling cost (cooling heterogeneity). Cooling heterogeneity refers to the fact that from a particular cooling unit, all locations in a data center do not benefit to

the same degree. Related works in the literature have either simplified or ignored heterogeneity that exists in the data center environment when studying workload assignment or cooling control. We have studied the cost-saving opportunities that exist due to server heterogeneity during workload assignments [15], and also due to cooling heterogeneity [16], however no study has considered all aspects of data center thermal heterogeneity to control cooling unit parameters and assign workload.

In this paper, a holistic data center infrastructure control (HDIC) framework is presented. HDIC is a novel method to exploit all aspects of data center thermal heterogeneity and uses them as an opportunity to save power during data center control. The proposed framework employs neural networks to construct thermal models for the data center and individual servers. Server thermal models are used to estimate the core temperature of servers, and a data center thermal model is used to predict the inlet temperatures of servers. These have the attraction of being data-driven models, as building accurate physical models for data center thermal dynamics is notoriously tricky.

The generated thermal models incorporate both cooling and server heterogeneity. These models can then be used by an optimizer to control the system in a power-efficient manner. We demonstrate that the solutions to the underlying optimization problem lead to considerable power savings while maintaining IT performance. Our contributions in this paper can be summarized as follows:

- We model the thermal differences between servers and locations in a data center using a novel thermal model.
- We incorporate low complexity data-driven thermal models to take thermal heterogeneity in data centers into account during workload assignment and

cooling control.

- We present an optimization framework that can jointly optimize the assignment of workload and the operational parameters of the cooling unit(s), while respecting the expected performance of IT equipment.

In the next section, related work is classified and reviewed. In Section 5.3, the architecture of the system under study is illustrated and the required models to formulate the problem are explained in Section 6.3.2. In Section 6.3, the methodology for cooling control and workload assignment is discussed and techniques to optimize the data center control parameters are explained. The solution of the developed optimization problems is discussed in Section 5.5, and HDIC is compared with other representative methods. Finally, concluding remarks are in Section 6.6. A summary of the notation used in this paper is listed in Table 5.1.

5.2 Literature review

There is a significant literature on this topic, studying various control methods, workload assignment frameworks, and thermal models for data centers. In this section, a number of previous works related to our contributions are reviewed: data center thermal models, thermal-aware workload assignment frameworks and thermal-aware control methods.

There are various methods of temperature prediction for data centers (data center thermal models). Computational fluid dynamics (CFD) is a traditional method for data center thermal simulations. This approach is based on fluid mechanics, using heat transfer relations and laws of physics. Having an accurate simulation requires

Table 5.1: Notation

Variable	Definition
$\bar{\square}$	Vector of a variable ($\square$ is any variable)
n_s	Total number of servers
d	Offered workload to data center (percent)
ρ	CPU utilization of server (percent)
ρ_{max}	Maximum allowed CPU utilization
$Q_{water}^{cooling}$	Cooling unit inlet water flow-rate (cfm)
$Q_{air}^{cooling}$	Air flow-rate generated by fans (cfm)
$Q_{fan}^{cooling}$	Fan speed (percent of maximum)
$T_{water}^{cooling}$	Cooling unit inlet water temperature (°C)
T_{inlet}^{server}	Inlet temperature of server (°C)
$T_{setpoint}^{cooling}$	Set-point temperature of cooling unit (°C)
T_{red}^{server}	Red line temperature of servers (°C)
T_{cpu}^{server}	Temperature of server CPU (°C)
T_{red}^{cpu}	Red line temperature of CPU (°C)
$T_{evap}^{chiller}$	Evaporator temperature of chiller (°C)
$T_{cnd}^{chiller}$	Condenser temperature of chiller (°C)
P^{dc}	Total power consumption of data center (Watt)
P^{it}	Power consumption of IT (Watt)
$P^{cooling}$	Total power consumption of cooling unit (Watt)
P^{fan}	Power consumption of cooling fans (Watt)
$P^{chiller}$	Power consumption of chiller (Watt)
P^{heat}	Total amount of generated heat (Watt)
C_i	Server power model coefficient
α_i , β_i	Chiller and fan power model coefficient, respectively
n_u , n_y	System model input and output delay, respectively
$u(t)$, $y(t)$	System model input and output at time t, respectively
$\widehat{y}(t)$	Predicted output of the system

the discretization of the volume of the simulated object and solving numerous simultaneous equations for the discretized dynamics. This method is computationally complex and needs powerful computing devices [14]. In [17], Moazamigoodarzi et al. presented a zonal-based method for temperature prediction. Their method considers heat-transfer differential equations and energy conservation laws between adjacent thermal zones to model the thermal dynamics. Due to simplifications to form heat-transfer equations, this method suffers with respect to both scalability and accuracy for a heterogeneous data center environment.

Li et al. [18] presented a learning-based temperature forecasting method for data centers. They simplified a full-fledged fluid dynamic model by combining physical laws and sensor observations. Their model constantly measures temperatures and air flow-rates surrounding each server. A learning algorithm is used to relate the sensor observations to the IT load. Although the idea of combining machine learning models and physical laws is attractive, the method needs accurate temperature and air-flow measurements from surrounding servers. There are complex air flow patterns in data centers and it is not clear that measurements from one sensor per thermal zone would result in sufficient fidelity for acceptable performance. In addition, this method requires what may be an inordinate number of sensor installations.

In addition to the above mentioned methods of temperature prediction in data centers, there are two methods which are data-driven (instead of using laws of physics and heat transfer) and resemble more closely our thermal model. Moore and Ranganathan [19] developed an approach based on neural networks for estimating temperature in data centers. In this approach the IT profile, air flow-rates, supply air temperatures, and the geometry of a data center are provided as inputs to the model.

The main issue with this approach is that it requires an extensive number of steady-state data points. Additionally, the geometry such as locations of compartments, walls, and servers, does not allow the model to be adaptive to any physical change in a data center. Wang et al. [20] presented a method for temperature prediction in data centers also based on neural networks, but with a different perspective. This method considers thermal effects on the temperature distribution caused by a server when it runs a task, which they call the *task-temperature profile*. The main issue with this method is that it does not consider the effects of other operational variables inside a data center which also impact the temperature distribution such as a change in the speed of a fan.

There are a number of works aiming to deal with cooling heterogeneity via the current status (temperature) of a data center without providing any feedback to cooling units, for example, assigning workload inversely proportional to the exhausted air temperature [21], or assigning workload based on server inlet temperature and current workloads of neighboring servers [22]. Bash and Forman [13] presented a method to rank cooling efficiency of a server location for workload assignment. Servers are ranked based on their response to a step change in the supply air temperature of cooling units. The longest jobs are then assigned to the highest ranked servers. Chaudhry et al. [23] presented a thermal-aware server relocation algorithm based on monitoring server inlet temperatures. The goal of this paper is to decrease the peak outlet temperature of servers by relocating them.

There is another group of thermal-aware workload assignment approaches that provide simple feedback to the cooling unit (usually a set-point) and use a simplified thermal model for temperature prediction. Using this simple thermal model is one

way to consider cooling heterogeneity. Tang et al. [12] developed a thermal-aware workload manager to minimize the peak server inlet temperature through optimal assignment of workload. Their work is based on a static heat recirculation matrix (HRM). The utilization of each server is assumed to be 0 or 1. Abbasi et al. [24] minimized the total amount of heat recirculation to increase the cooling unit supply air temperature. The main difference between [12] and [24] is that in [24] a finer-grained workload can be assigned to a server.

Zhao et al. [25] used a control loop to regulate inlet temperatures of servers through adjusting operating frequencies and utilizations of servers with a goal of minimizing the total power consumption. The thermal model used in this work is based on an HRM. Fang et al. [26] presented a framework for optimizing server and cooling unit settings to reduce power consumption. An optimization process, employing an HRM, is used for selecting the active set of servers, controlling the servers, and adjusting the set-points of the cooling units. They used CFDs to obtain the HRM, resulting in over-simplification. Moreover, the only parameter to control cooling units is the set-point, controlling fine-grained cooling parameters is not performed by the framework. This limitation restricts the potential to save power.

Another category of work studies the cost saving opportunities available when considering cooling heterogeneity. Mukherjee et al. [27] enhanced Abbasi et al. [24] by considering job deadlines as extra constraints in the power minimization problem. Moreover, servers can be slowed down to throttle temperature peaks. The main drawback for the work in [12, 24, 26, 27] is that their thermal models rely on a static HRM that may not be appropriate for the dynamic environment of a data center. To this end, Wang et al. [28] enhanced the HRM-based approach by making

it adaptive to air flow changes. Their method uses a learning-based method that reflects variations of air flows in the HRM. This work admits the inadequacy of a static HRM for thermal-aware workload assignment methods. However, while they do consider dynamic air flows, just one air flow is considered for each rack. From top to bottom, each rack may have a variety of air flows and assuming just one air flow for a rack endangers the model accuracy. Moreover, the air flow profile is the result of the action of multiple fans. The feasibility of providing the optimized air flow patterns by tuning fan parameters is not clear.

There is a body of work that considers finer-grained cooling unit variables than just a set-point. Wang et al. [29] studied the problem of optimal control of fan speeds using a multi-input/multi-output (MIMO) method with the aim of preventing fan speed over-provisioning in server blades. The control method is able to reduce power consumption of a blade by as much as 20%. This work is interesting with respect to the thermal and heat transfer models that are used. Although this work introduced more accurate thermal models (in contrast with HRM-based models), there are opportunities to increase the resolution of fan effects on the cooling efficiency of cooling units.

Yao et al. [30] used an adaptive predictive control method for workload balancing in data centers. An adaptive thermal model is used to predict inlet temperatures. They formulate an optimization problem with a goal to smooth server inlet temperatures and decrease total power consumption. Their cost function is formulated based on total power (cooling unit and IT racks) and tracking error of a predictive model. The controller adjusts the inputs (server workloads) to set the inlet temperatures while minimizing the total power consumption.

In [15], we developed a thermal-aware cooling control and workload assignment method which has the potential to save a considerable amount of power. The method assigns more workload to cooling-efficient servers and less to cooling-inefficient servers. In this way, a higher set-point could be set for the cooling unit while satisfying the cooling requirements of all of the servers. The higher the cooling unit set-point, the less power is consumed. This method saves power mainly by preventing server over-cooling. Additionally, we showed in [16] that the operational parameters of the cooling unit and assignment of the given workload could be configured in multiple ways with different costs while providing the same cooling and IT capacity. In particular, taking cooling cost information into account when making cooling control and workload assignment decisions has the potential to yield significant power savings. However, none of these methods considered all aspects of cooling heterogeneity. Moreover, the model is physics-based and evaluation is made via simulation. Such an approach may have issues with respect to accuracy, adaptability, and scalability.

5.3 System architecture and models

In this section, the architecture of the data center under study is provided. The steps to acquire data and then to build data center and server thermal models are explained and the power consumption is formulated.

5.3.1 System architecture

For generating the data center thermal model, we used our on-site data center, shown in Figure 5.1. The data center consists of five IT racks and two in-row cooling

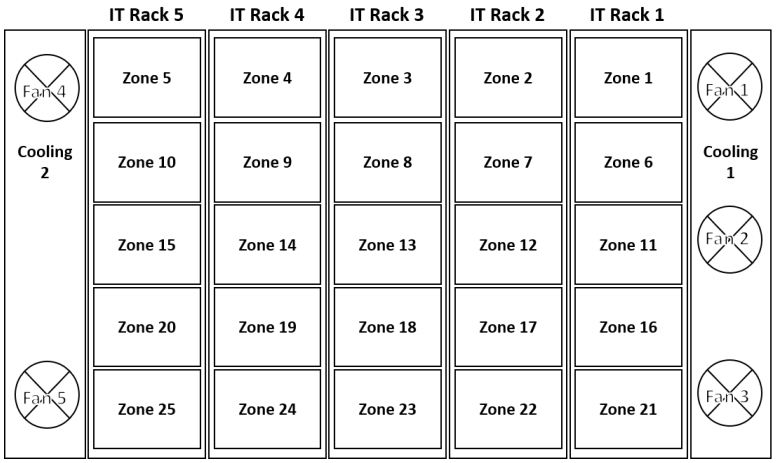


Figure 5.1: Front view of data center with two in-row cooling units at either side and five IT racks

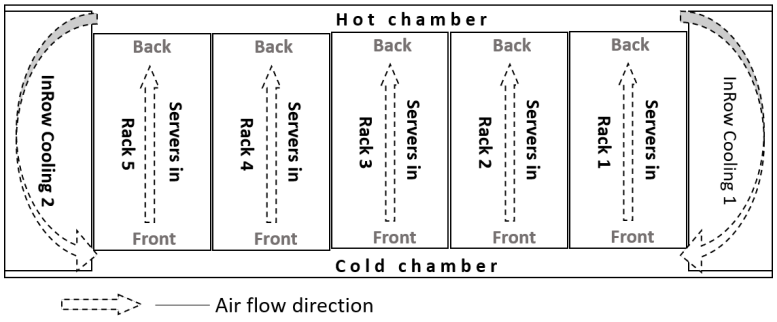


Figure 5.2: Top view of data center

units. The height of each rack is divided into five equal height thermal zones. We developed a data acquisition tool to both apply our desired configurations and acquire all operational variables of cooling units and server profiles. Figure 5.2 shows the top view of our data center.

Figure 5.3 shows the architecture of each cooling unit. As shown, each cooling unit has a number of fans that draw hot air from the hot chamber, pass the air through a heat exchanger and blow the cold air to the cold chamber. Water flow within the heat exchanger transfers the generated heat out of the facility. In other words, chilled

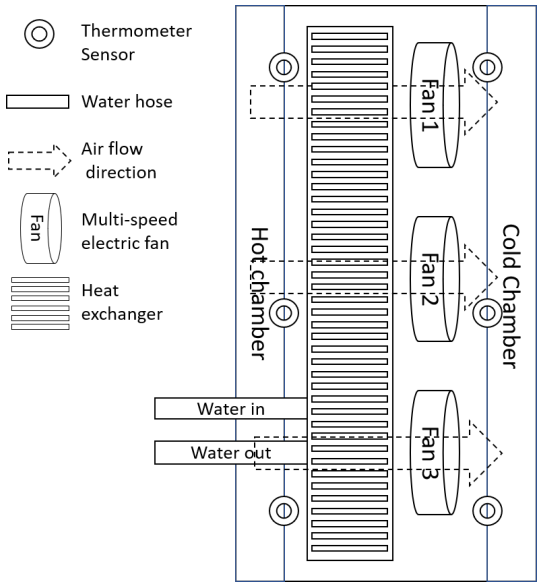


Figure 5.3: In-row cooling unit

water enters the heat exchanger, and warm water exits.

Cooling unit operational parameters can be controlled and monitored using the *simple network management protocol* (SNMP) [31]; these parameters include the speed of each fan and the water flow-rate inside the heat exchanger of the cooling unit. On the other hand, the IT consists of servers that process the given workload. We can apply the given workload to servers and collect real-time reports using SSH commands [32]. Each server is able to report the current utilization and temperature of its cores. Temperatures at the front of servers are obtained via thermal sensors (DS18B20 digital thermometers) placed in each zone.

Our tool connects to cooling units using SNMP, to servers by SSH, and to thermal sensors via serial ports. It takes operational scenarios as inputs. A scenario is a time series of values that needs to be applied to the controllable variables of the data center. The operational scenario should be rich in parameter variation to be suitable to train the model. Upon executing a scenario, workload patterns are applied to servers and

patterns of operational parameters are applied to cooling units. At the same time, reported data including measurements from the thermometers (installed at 25 thermal zones), the utilization and CPU temperature of servers, and operational parameters of the two cooling units are saved in a database. The operational parameters of the cooling units consist of the inlet water temperature ($T_{water}^{cooling}$), the water flow-rate ($Q_{water}^{cooling}$) and fan speeds ($Q_{fan}^{cooling}$).

5.3.2 Data center thermal model

An accurate prediction of server inlet temperatures is crucial for effective data center control methods, as inlet temperatures are key to safe operation of IT equipment. In [33], we provided a framework for constructing a data center thermal model using neural networks.

In this work, we use time series forecasting to model inlet temperatures of servers. It has been shown that neural networks outperform traditional time series forecasting [34]. Thus, we use a neural network model to predict the inlet temperatures of servers several time steps in the future. In particular, a nonlinear auto-regressive network with exogenous input (NARX) is chosen, where the exogenous variables are all the control variables of the data center. The choice of the NARX network is due to its ability to model nonlinear dynamic systems and capture time dependencies [35]. The NARX network has the representation

$$\widehat{y}(t+1) = f\left(y(t), \cdots, y(t-n_y), u(t), \cdots, u(t-n_u)\right), \tag{5.3.1}$$

where y and u are (a finite number of) past outputs and inputs, and $\widehat{y}$ is the predicted output. The values $n_y \geq 1$ and $n_u \geq 1$ are the orders of the delays for the

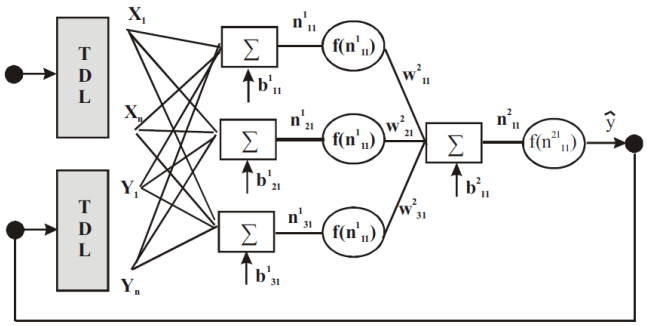


Figure 5.4: Closed-loop NARX network

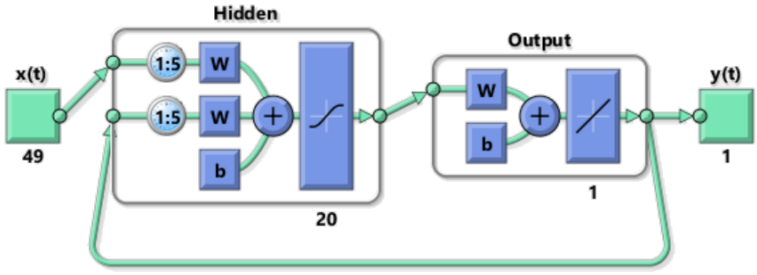


Figure 5.5: MATLAB implementation of Closed-loop NARX network

included outputs and inputs and f is a nonlinear function. Unlike recurrent neural networks, feedback to the network in Figure 5.4 is directly from the output and there is no feedback in the hidden layer. This architecture has been shown to be more computationally efficient than fully connected recurrent neural networks [35].

The model takes IT and cooling unit parameters as inputs and predicts temperature. These input parameters of the thermal model are the utilization profile of servers, inlet water temperatures, water flow-rates, and fan speeds of cooling units. A MATLAB implementation of a NARX network is shown in Figure 5.5 for $n_u = 5$ and $n_y = 5$. One hidden layer with size 20 is chosen and the model is trained using the Levenberg-Marquardt back-propagation method.

The performance of the network is demonstrated by comparing the measured and

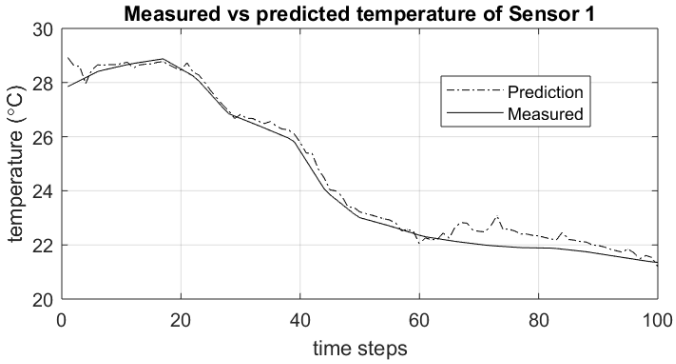


Figure 5.6: Temperature of Zone 1

estimated temperatures and errors in both the immediate time step and multiple future time steps. The average error for estimating the next time step is $0.1°C$ and for 100 steps in the future is $1.2°C$ (Figure 5.6) for a time step of 10 seconds. For this work, steady state estimation of the inlet temperatures of servers with respect to the cooling variables and workload profile is required.

5.3.3 Server thermal model

Thermal heterogeneity of servers is the result of different thermal conditions of servers which in turn cause different patterns of CPU temperature change [15]. This heterogeneity can be captured by server thermal models. Therefore, a model for the CPU temperature of each server, or server thermal model, is required to be generated.

The maximum allowable CPU temperature is called the red-line temperature (T_{red}^{cpu}). The CPU temperature of a server, T_{cpu}^{server}, depends on two contributing factors, CPU utilization (ρ) and inlet temperature (T_{inlet}^{server}) of the server [15]. Increasing each of them (T_{inlet}^{server} or ρ) increases T_{cpu}^{server}. So, a server thermal model is required to return T_{cpu}^{server} for all servers as a function of the server inlet temperatures ($\bar{T}_{inlet}^{server}$) and the CPU utilizations ($\bar{\rho}$). This model would allow an optimizer to assign server

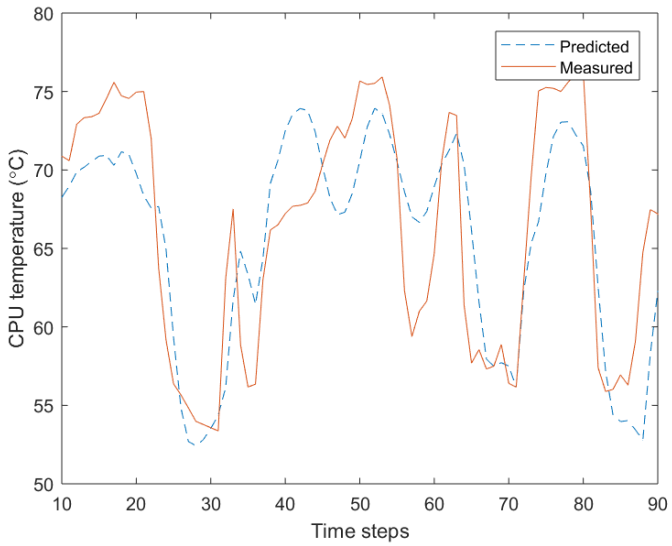


Figure 5.7: CPU temperature of a server in Zone 2

utilizations to maximize the required T_{inlet}^{server} subject to keeping CPU temperatures below T_{red}^{cpu}.

The same method as used for the data center thermal model is chosen for constructing server thermal models. However, as we are looking at individual servers, the implementation of the method is somewhat simpler. A closed-loop NARX network is chosen with T_{inlet}^{server} and ρ as the inputs of the network and the CPU temperature (T_{cpu}^{server}) as its output. Figure 5.7 shows that the NARX network is able to track temperature changes accurately.

5.3.4 Power model

The total power consumption of a data center, P^{dc} is the sum of the power consumption of servers (which we call IT power or P^{it}) and cooling units ($P^{cooling}$):

$$P^{dc} = P^{it} + P^{cooling}. \tag{5.3.2}$$

The major contributing factor for the IT power (P^{it}) is the power consumption of servers, which is modeled as an affine function of its utilization (ρ) [15]:

$$P^{it} = \sum_{i=1}^{n_s} c_{1,i} + c_{2,i} \cdot \rho_i. \tag{5.3.3}$$

In (5.3.3), $c_{1,i}$ is the power consumption of an idle server and $c_{1,i} + c_{2,i}$ is the power consumption of the i^{th} server when it is fully utilized. The other contributing factor of the data center power consumption is the cooling power. The cooling power ($P^{cooling}$) consists of the power consumption of fans (P^{fan}) and the power consumed by the chillers $P^{chiller}$:

$$P^{cooling} = P^{fan} + P^{chiller}. \tag{5.3.4}$$

P^{fan} is the power consumed by fans to circulate air inside the data center to facilitate heat transfer. Higher fan speed means greater fan power consumption (P^{fan}). The power consumption of the cooling unit fans is given by

$$P^{fan} = \beta_1 + \beta_2 \cdot Q_{fan}^{cooling} + \beta_3 \cdot (Q_{fan}^{cooling})^2 + \beta_4 \cdot (Q_{fan}^{cooling})^3, \tag{5.3.5}$$

where β_i, $i = 1, 2, 3, 4$ are constants, and $Q_{fan}^{cooling}$ is the fan speed in percentage of maximum.

A chiller provides chilled water to the cooling units. The inlet water temperature ($T_{water}^{cooling}$) is the temperature of chilled water provided to cooling units by the chiller. The lower the value of $T_{water}^{cooling}$ the higher the power consumption of the chiller. The model given in (5.3.6) represents the power consumption of the chiller [14]:

$$P^{chiller} = P^{heat} \cdot \left(\frac{\alpha_1 + \alpha_2 \cdot \frac{T_{evap}^{chiller}}{P^{heat}} + \alpha_3 \cdot (T_{cnd}^{chiller} - T_{evap}^{chiller})}{\frac{T_{evap}^{chiller}}{T_{cnd}^{chiller}} - \alpha_4 \cdot \frac{P^{heat}}{T_{cnd}^{chiller}}} - 1 \right). \quad (5.3.6)$$

In (5.3.6), P^{heat} is the total amount of heat that should be removed by the chiller (equal to P^{it} in our case). $T_{evap}^{chiller}$ is the evaporator temperature, which is approximately equal to $T_{water}^{cooling}$, and $T_{cnd}^{chiller}$ is the condenser temperature. The evaporator and condenser are the two main chiller components. Both $T_{evap}^{chiller}$ and $T_{cnd}^{chiller}$ are in kelvins and the quantities α_i are constants. While temperatures in these models are in kelvins, later in the paper temperatures will be reported in degrees Celsius.

In the next sections, only the cooling power consumption is considered and not the total power consumption of the data center. This follows from the observation that for our model the IT power is independent of the workload assignment. Note that these approaches for generating thermal and power models are not unique. The literature on power models is well established and as a result these models are appropriate for our use [14]. On the other hand, the thermal models are generated using data-driven approaches. This is mainly due to the impracticality and/or complexity of the existing thermal models in the literature.

5.4 Thermal-aware cooling control and workload assignment

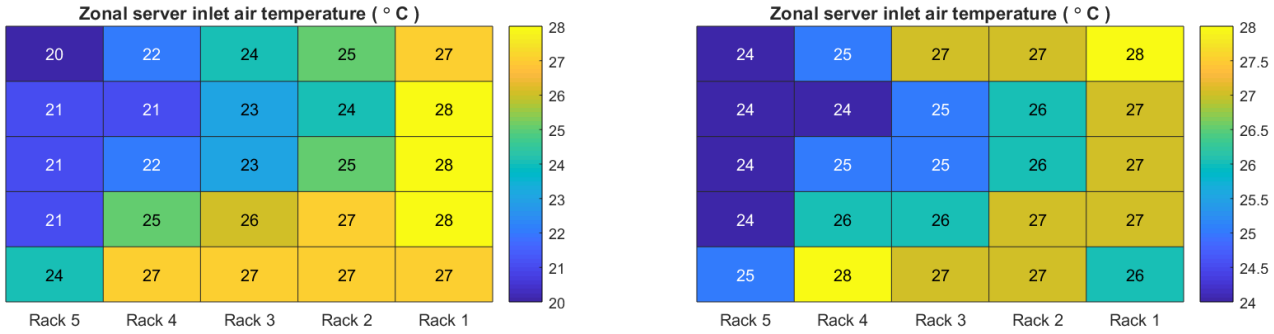
Exploring data center thermal heterogeneity is possible through thermal models. In this section, two different approaches are discussed to be compared later as a demonstration of the efficiency of HDIC. In the first approach, cooling heterogeneity is only considered via the data center thermal model. This approach is called *cooling heterogeneity-aware infrastructure control* or CHIC. The second approach is HDIC which uses both the data center and server thermal models for control decisions. An optimization problem for each method, CHIC and HDIC, is formulated and justified.

CHIC keeps the inlet temperatures of servers below a threshold temperature, T_{red}^{server}. HDIC maintains the CPU temperatures of servers below T_{red}^{cpu}. Violating the device threshold temperature dramatically increases the chance of device failure [36]. These two thresholds or red-line temperatures are chosen based on ASHRAE guidelines [37].

5.4.1 Cooling heterogeneity-aware infrastructure control (CHIC)

Cooling units do not provide a uniform temperature distribution throughout a data center. Some locations receive more cool air than others. From a cost point of view, the cooling costs of different locations in a data center are different; some locations are cooled with less cost than other locations. These differences stem mainly from the physics of heat transfer and hot air recirculation.

A cooling unit can reach a temperature target through multiple settings. The temperature target specifies the maximum allowed server inlet temperature. For example,



(a) (b)

Figure 5.8: Server inlet air heat-map (a) $T_{water}^{cooling} = 12°C$, $Q_{fan}^{cooling}$ from Fan 1 to Fan 5: 39%, 39%, 0%, 32%, 0% , $P^{cooling} = 1418W$, (b) $T_{water}^{cooling} = 18°C$, $Q_{fan}^{cooling}$ from Fan 1 to Fan 5: 62%, 57%, 34%, 51%, 51% , $P^{cooling} = 1366W$

two patterns of the temperature distribution are shown in Figure 5.8, corresponding to two different settings. T_{red}^{server} is set to $28°C$. Both settings are able to satisfy the cooling target, however with different costs. The cooling cost for the settings of Figure 5.8b is less than for the settings of Figure 5.8a.

In this paper, a data-driven neural network model is used to study the possible solutions for the power-efficient data center operations. Figure 5.9 shows the optimal adjusted values (red) and the corresponding power consumption of the cooling unit (blue) for the fan speeds versus the inlet water temperature. The target temperature is set to $24°C$. Tracing the power curve clearly explains the necessity of optimizing the operational variables of the data center. The figure shows that as $T_{water}^{cooling}$ increases fans should compensate by increasing the air flow, which is reflected in the $Q_{fan}^{cooling}$ curves. $P^{chiller}$ and P^{fan} are monotonically increasing functions of $T_{water}^{cooling}$ and $Q_{fan}^{cooling}$, respectively. Their sum, as shown in the power curve of Figure 5.9, reaches a minimum point with the optimal selection of the operational valuables.

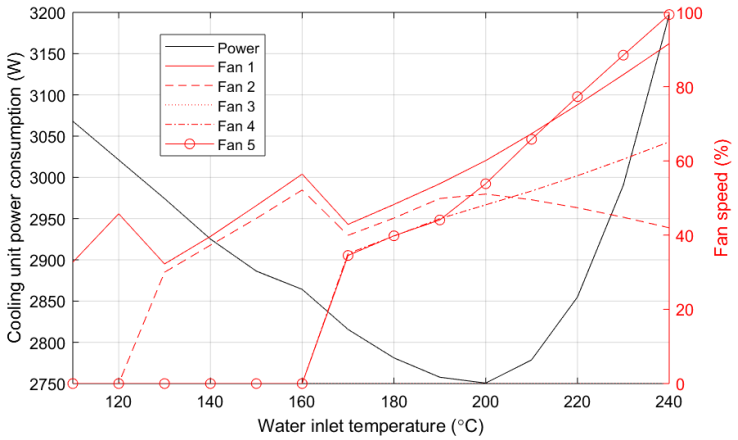


Figure 5.9: Demonstration of the trade-off within cooling operational parameters

The optimization problem (6.3.1) finds the optimal values for the controlled parameters to minimize $P^{cooling}$ while keeping the inlet temperatures of servers below the red-line (T_{red}^{server}) and respecting a number of additional constraints.

$$\min_{\bar{\rho},\bar{Q}_{fan}^{cooling},T_{water}^{cooling}} \quad P^{cooling} = P^{fan} + P^{chiller} \tag{5.4.1a}$$

$$\text{subject to:} \quad 0 \leq \rho_i \leq \rho_{max} \tag{5.4.1b}$$

$$\sum_{i=1}^{n_s} \rho_i = d \cdot n_s \tag{5.4.1c}$$

$$0\% \leq Q_{fan}^{cooling} \leq 100\% \tag{5.4.1d}$$

$$11°C \leq T_{water}^{cooling} \leq 24°C \tag{5.4.1e}$$

$$\bar{T}_{inlet}^{server} = f(\bar{\rho}, \bar{Q}_{fan}^{cooling}, T_{water}^{cooling}) \tag{5.4.1f}$$

$$\bar{T}_{inlet}^{server} \leq T_{red}^{server} \tag{5.4.1g}$$

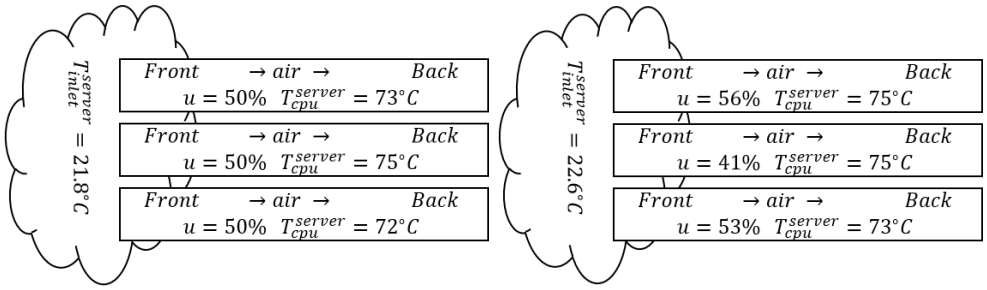
This nonlinear optimization problem (6.3.1) minimizes the power consumption of the cooling units ($P^{cooling}$). $P^{cooling}$ is the sum of the power consumption of the

fans and the chiller. P^{fan} is a function of the fan speeds ($\bar{Q}_{fan}^{cooling}$) and $P^{chiller}$ is determined by $T_{water}^{cooling}$, as addressed in Section 5.3.4. Equation (6.3.1d) constrains the assigned values to the servers between 0 and ρ_{max} (choosing ρ_{max} strictly less than one could be done to satisfy performance constraints). Equation (6.3.1e) guarantees the assignment of the total given load. The bounds for fan speeds and the inlet water temperature are given by (6.3.1f) and (6.3.1g), respectively. Equation (6.3.1h) uses the data center thermal model to generate inlet temperatures of servers, as explained in Section 6.3.2. The last constraint enforces the maximum allowed inlet temperature of a server (6.3.1i).

5.4.2 Holistic data center infrastructure control (HDIC)

Considering cooling heterogeneity during cooling control and workload assignment is able to save a considerable amount of power, as will be shown in Section 5.5. In this part, server heterogeneity along with cooling heterogeneity is considered. We use both data center and server thermal models to (1) adjust cooling unit parameters in a power-efficient way and (2) assign workload to servers that can be cooled efficiently.

Figure 5.10 gives an intuitive example of power saving capabilities of considering server heterogeneities using two settings, both with three servers. The total workload is equal for both settings. In Setting 1, the utilization of each server is 50% when $T_{inlet}^{server} = 21.8°C$ and the maximum CPU temperature is T_{red}^{cpu}. In Setting 2, to keep the CPU temperature of servers below T_{red}^{cpu} the inlet temperature of servers should be $T_{inlet}^{server} = 22.6°C$, which means lower cooling power consumption. This example shows how workload assignment considering server heterogeneity can potentially save cooling power.



(a) Setting 1 (b) Setting 2

Figure 5.10: Server heterogeneity intuition

The data center thermal model gives the server inlet temperature distribution as explained in Section 6.3.2. It uses cooling variables and the workload profile in the form of a utilization vector ($\bar{\rho}$) to return the inlet temperatures of servers ($\bar{T}_{inlet}^{server}$). In addition, server thermal models give the required inlet temperatures of servers to keep the entries in $\bar{T}_{cpu}^{server}$ below the red-line temperature (T_{red}^{cpu}).

The optimal selection of the cooling variables and workload assignment is the

solution to the optimization problem (6.3.2).

$$\min_{\bar{\rho},\bar{Q}_{fan}^{cooling},T_{water}^{cooling}} \quad P^{cooling} = P^{fan} + P^{chiller} \tag{5.4.2a}$$

$$\text{subject to: } 0 \leq \rho_i \leq \rho_{max} \tag{5.4.2b}$$

$$\sum_{i=1}^{n_s} \rho_i = d \cdot n_s \tag{5.4.2c}$$

$$0\% \leq \bar{Q}_{fan}^{cooling} \leq 100\% \tag{5.4.2d}$$

$$11°C \leq T_{water}^{cooling} \leq 24°C \tag{5.4.2e}$$

$$\bar{T}_{inlet}^{server} = f(\bar{\rho}, \bar{Q}_{fan}^{cooling}, T_{water}^{cooling}) \tag{5.4.2f}$$

$$\bar{T}_{cpu}^{server} = g(\bar{\rho}, \bar{T}_{inlet}^{server}) \tag{5.4.2g}$$

$$\bar{T}_{server}^{cpu} \leq T_{red}^{cpu} \tag{5.4.2h}$$

This nonlinear optimization problem (6.3.2) is similar to (6.3.1). The differences are the use of both the data center thermal model (5.4.2f) and the server thermal model (5.4.2g). Moreover, (5.4.2h) constrains CPU temperatures instead of inlet temperatures, as in (6.3.1i).

5.5 Results and comparison

Both optimization problems (6.3.1) and (6.3.2) should be solved by nonlinear solution methods as both the cost function and the thermal models are nonlinear. We used *interior-point* methods to solve the optimization problem. A complete description of the data center configuration is illustrated in Section 5.3.1. Briefly, for this data center configuration, the decision variables are the utilizations of 40 servers, the speed of five

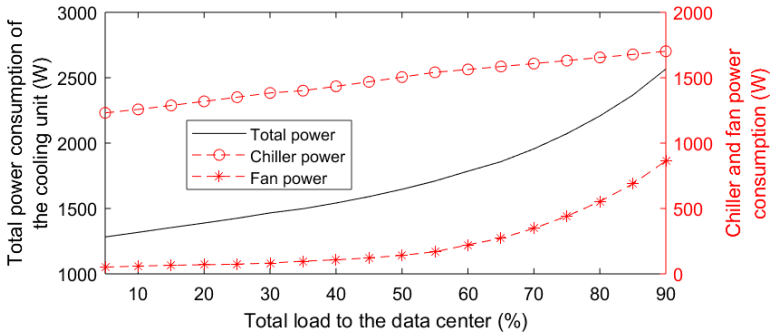


Figure 5.11: Power consumption of the cooling using CHIC

fans, and one inlet water temperature. Due to the computational simplicity of the thermal models, a solution can be obtained relatively fast. In the next subsection, the solutions to the first and the second optimization problems are discussed and compared. The comparison shows advantages of using HDIC over CHIC. In the second subsection, HDIC is compared with other representative control methods, suggesting that HDIC outperforms these other methods.

5.5.1 Results and discussion

The solutions of the optimization problems (6.3.1) and (6.3.2) are compared in this section. The minimized cooling power (in Watts) along with power consumption for the fans and chiller corresponding to (6.3.1) are shown in Figure 5.11. Figure 5.12 shows the optimized values for the cooling unit operational parameters with respect to the offered load to the system. The first optimization problem only uses the data center thermal model (cooling heterogeneity only).

As seen in Figure 5.11, the cooling power is an increasing function of the offered load to the data center. The majority of the power consumption is due to the chiller. The optimal solution sets the fan power and the chiller power such that the maximum

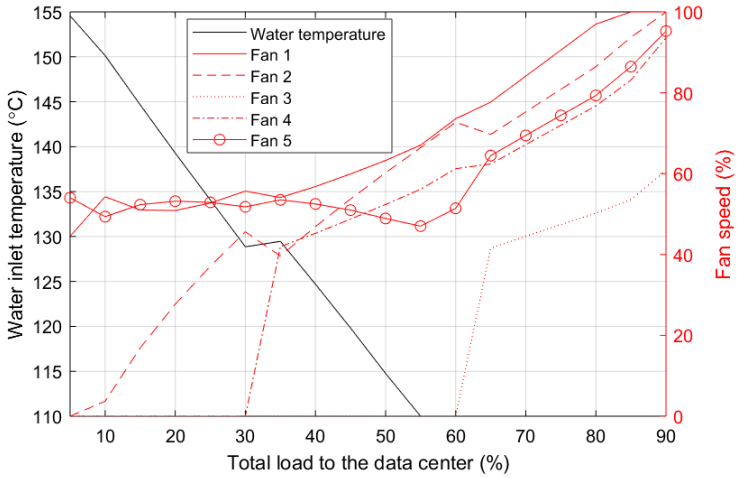


Figure 5.12: The optimized operational variables of the cooling unit using CHIC

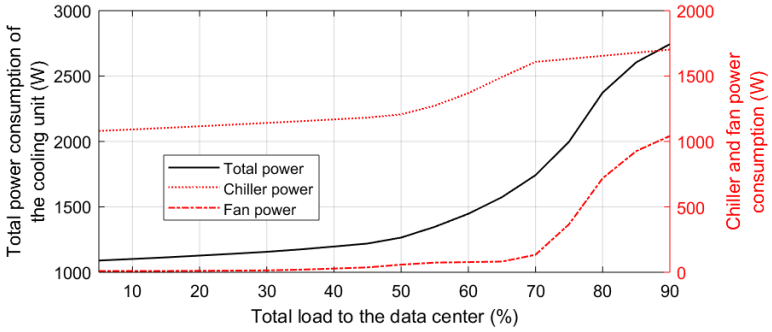


Figure 5.13: Power consumption of the cooling unit using HDIC

air temperature at the front of the servers is below T_{red}^{server} which in this case is chosen to be $24°C$.

The results for the second optimization problem (6.3.2), are shown in Figure 5.13 and Figure 5.14. Figure 5.13 shows $P^{cooling}$, $P^{chiller}$, and P^{fan}. Figure 5.14 shows the optimal values for fan speeds and inlet water temperature. The upper-bound for the CPU temperature (T_{red}^{cpu}) is set to $75°C$. This optimization problem considers both the server and data center thermal models (both cooling and server heterogeneity).

Comparing the optimal values of the cooling parameters corresponding to CHIC

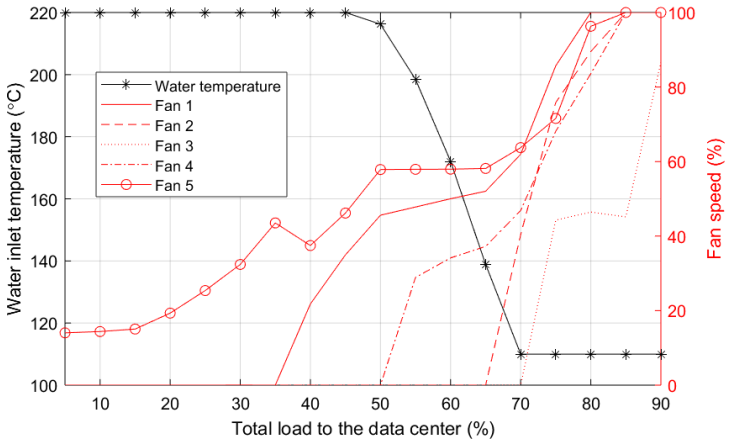


Figure 5.14: The optimized operational variables of the cooling unit using HDIC

and HDIC shows that $T_{water}^{cooling}$ decreases and fan speeds increase faster under CHIC. Figure 5.15 and Figure 5.16 compare the power consumption and the cooling coefficient of performance (CoP) for CHIC and HDIC as a function of d. The CoP is the ratio of the heat removed by the cooling system to the work required, which is usually greater than one. The higher the CoP the more efficient the cooling unit [14]. HDIC is able to save 16% more power than CHIC, between $d = 30\%$ and 70%. This saving arises mainly from the fact that the second optimization problem uses both the data center thermal model and the server thermal model. On the one hand, the server thermal model provides the CPU temperature of a server based on the inlet temperature of the server and its CPU utilization. In other words, the required inlet temperature to keep the CPU temperature below a certain threshold is controlled by the utilization of a server. On the other hand, the contribution of the generated heat by a server is accounted for by the data center thermal model. Therefore, the optimizer simultaneously provides power efficient cool air for servers by tuning the cooling variables and the assignment of workload, and adjusts cooling requirements

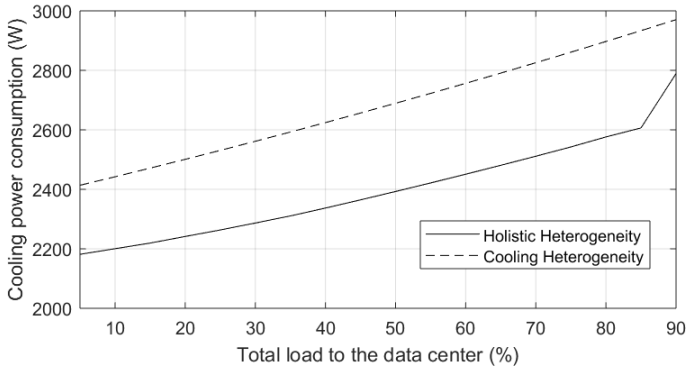


Figure 5.15: Cooling power comparison

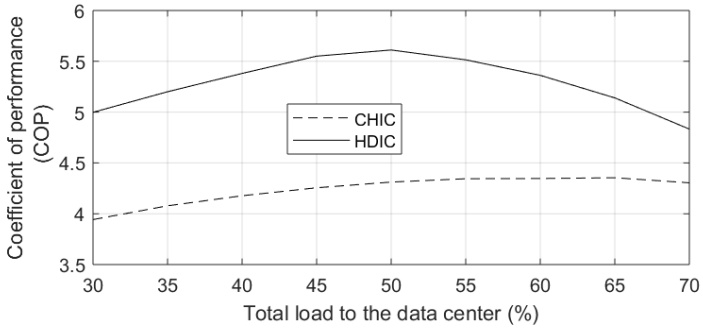


Figure 5.16: Coefficient of performance of CHIC vs HDIC

of servers.

Table 5.2 shows the CPU and inlet temperatures of servers for both HDIC and CHIC when $d = 60\%$. The cooling power consumption to satisfy temperature constraints via CHIC and HDIC methods is $1574W$ and $1209W$, respectively. There are two key insights comparing the table values. First, the inlet temperatures of servers in the HDIC method are generally at least as high as for the CHIC method. This is the main reason HDIC saves more power, due to less over-cooling. Second, the CPU temperatures for CHIC in a few cases are higher than $T_{red}^{cpu} = 75^{\circ}$ and the maximum CPU temperature of CHIC reaches $81.2^{\circ}C$. Exceeding the CPU red-line temperature can reduce server reliability (increased failure rate) and result in decreased performance

Table 5.2: Inlet and CPU temperature of servers corresponding to CHIC and HDIC

	Server No.	01	02	03	04	05	06	07	08	09	10	11	12	···	38	39	40	min	max
CHIC	$\bar{T}^{servr}_{inelt}(^{\circ}C)$	23.4	22.0	21.7	21.7	21.7	21.7	21.9	22.0	21.5	22.0	22.0	22.0	···	21.9	21.5	23.4	20.5	23.9
	$\bar{T}^{servr}_{cpu}(^{\circ}C)$	79.7	65.7	65.2	69.0	67.3	75.2	70.6	67.6	64.7	61.2	68.7	70.8	···	71.7	67.2	73.7	58.0	82.1
HDIC	$\bar{T}^{servr}_{inelt}(^{\circ}C)$	25.5	24.3	24.2	24.2	24.2	24.7	25.0	23.5	23.5	24.3	25.0	25.0	···	24.9	24.6	25.5	21.8	25.5
	$\bar{T}^{servr}_{cpu}(^{\circ}C)$	75.0	74.9	75.0	74.5	73.8	75.0	73.7	74.9	72.9	70.0	74.7	73.7	···	73.2	73.2	72.2	64.1	75.0

(for example, due to CPU throttling).

5.5.2 Comparison

We proceed with comparing HDIC with other representative workload assignment and cooling control methods. The representative methods are categorized according to two baseline approaches that are used as the core of a number of recent data center workload assignment and control methods: *HRM-based* approaches and *set-point tracking* approaches.

HRM-based approaches include works that simplify the data center thermal model via a matrix of coefficients [12, 24, 25, 27, 28]. A *heat recirculation matrix* or HRM is a cross-interference square matrix that represents the heat transfer rate between nodes. Having both the supply air temperature and the HRM an estimate of the inlet temperature of each server can be obtained using

$$\bar{T}^{server}_{inlet} = \bar{T}^{cooling}_{setpoint} + H\bar{P}. \tag{5.5.1}$$

In this equation, H is the *heat recirculation matrix*. The maximum amount of recirculated heat determines the supply air temperature ($\bar{T}^{cooling}_{setpoint}$) that should be provided by the cooling unit. The higher the supply air temperature the less power is drawn by the cooling unit. Hence, the optimal power distribution (or workload

assignment) minimizes the maximum amount of recirculated heat which in turn maximizes the required supply air temperature. Maximizing $T_{setpoint}^{cooling}$ means minimizing the cooling power. In the HRM-based approach the only feedback to the cooling units is the cooling set-point.

In order to compare this approach with HDIC H is calculated for our on-site data center. An optimization problem decides on $\bar{P}$ (equivalent to the workload distribution) for maximizing $T_{setpoint}^{cooling}$. The operational variables related to this method are obtained based on set-point tracking methods explained later in this section.

The other widely used method for cooling control in data centers is *set-point tracking*. In this approach, the cooling unit controller tries to meet the desired set-point as fast as possible with minimum undershoot and overshoot. In this method, the heat generation profile (workload distribution) is not considered. Comparing this method with HDIC, the workload is distributed uniformly between servers. The inlet water temperature is set to its minimum value, and all fans of a cooling unit are set to the same speed. The inlet temperatures of all servers are less than or equal to the set-point temperature ($T_{setpoint}^{cooling}$). An optimization problem determines the fan speed for each cooling unit.

Comparing the results of *HRM-based* and *set-point tracking* approaches with HDIC reveals that our control framework outperforms these methods. Table (5.3) presents a number of performance metrics for each of the methods. As shown in the table, the HDIC approach has the lowest power consumption and the highest CoP.

Clearly, our suggested method outperforms other representative methods due to using the thermal and power models. If one desires to only consider the problem from the view of inlet air temperature and efficiency of its distribution, without

Table 5.3: Comparing HDIC with HRM-based and Set-point Tracking approaches

Approach	Power (W)	CoP	$T_{water}^{cooling}$	$\bar{Q}_{fan}$
HDIC	1209	4.4	18	[0 0 100 29 29]
CHIC	1574	3.7	11.5	[59 54 0 53 52]
HRM-based	1621	3.5	11.0	[45 45 72 72 72]
Set-point tracking	1697	3.4	11.0	[61 61 61 88 88]

considering server differences, CHIC is also brought into the comparison in Table 5.3. The calculations were performed when $d = 50\%$. The power consumption and CoP of both CHIC and HDIC are better than the other two methods.

5.6 Conclusion

Considering all aspects of data center thermal heterogeneity for workload assignment and cooling control results in a considerable amount of savings in cooling power consumption. Data center heterogeneity can be obtained by means of data center and server thermal models. The data center thermal model predicts the temperature of different locations as a function of IT and cooling parameters. This thermal model is used to indirectly calculate the cost of providing cool air for a specific server with given cooling parameters. In addition, the server thermal model gives the required inlet temperature of a server to maintain CPU temperature constraints, based on the workload of the server. We proved that as a result of specific thermal conditions for each server, the temperature requirements of servers are different given the same workload. We presented methods to obtain these thermal models for a data center and then incorporated them in an optimization framework in order to minimize the cooling power. It is shown that our method is able to outperform other cooling control approaches and workload assignment methods.

Acknowledgment

This research was supported by a Collaborative Research and Development grant CRDPI506142-16 from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Bibliography

[1] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, W. Lintner, United States data center energy usage report, Tech. rep., Federal Energy Management Program of the U.S. Department of Energy, available: https://www.osti.gov/servlets/purl/1372902/ (2016).

[2] R. Brown, E. Masanet, B. Nordman, B. Tschudi, A. Shehabi, J. Stanley, J. Koomey, D. Sartor, P. Chan, Report to congress on server and data center energy efficiency public law 109-431, Tech. rep., US Environmental Protection Agency ENERGY STAR Program, available: https://eta.lbl.gov/sites/default/files/publications (2007).

[3] Y. Li, X. Wang, P. Luo, Q. Pan, Thermal-aware hybrid workload management in a green datacenter towards renewable energy utilization, Energies 12 (8) (2019) 1494.

[4] H. Klemick, E. Kopits, A. Wolverton, Data center energy efficiency investments: Qualitative evidence from focus groups and interviews, Tech. rep., U.S. Environmental Protection Agency National Center for Environmental Economics,

available: https://www.epa.gov/environmental-economics (2017).
URL https://www.epa.gov/environmental-economics

[5] M. Gupta, S. Singh, Using low-power modes for energy conservation in Ethernet LANs, in: INFOCOM 2007. 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 6-12 May 2007, Anchorage, Alaska, USA, IEEE, 2007, pp. 2451–2455. doi:10.1109/INFCOM.2007.299.

[6] N. Yadava, V. K. Mishra, R. K. Chauhan, Design of one-transistor SRAM cell for low power consumption, in: 2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES), IEEE, 2016, pp. 322–325.

[7] E. Aldahari, Dynamic voltage and frequency scaling enhanced task scheduling technologies toward green cloud computing, in: 4th Intl Conf on Applied Computing and Information Technology (ACIT), IEEE, 2016, pp. 20–25.

[8] R. Ge, X. Feng, K. W. Cameron, Performance-constrained distributed DVS scheduling for scientific applications on power-aware clusters, in: Proceedings of the ACM/IEEE SC2005 Conference on High Performance Networking and Computing, November 12-18, 2005, Seattle, WA, USA, CD-Rom, IEEE Computer Society, 2005, pp. 34–44. doi:10.1109/SC.2005.57.

[9] D. Meisner, C. M. Sadler, L. A. Barroso, W.-D. Weber, T. F. Wenisch, Power management of online data-intensive services, ACM SIGARCH Computer Architecture News 39 (3) (2011) 319–330.

[10] M. Lin, A. Wierman, L. L. Andrew, E. Thereska, Dynamic right-sizing for power-proportional data centers, IEEE/ACM Transactions on Networking (TON) 21 (5) (2013) 1378–1391.

[11] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, R. Katz, NapSAC: Design and implementation of a power-proportional web cluster, SIGCOMM Comput. Commun. Rev. 41 (1) (2011) 102–108. doi:10.1145/1925861.1925878. URL http://doi.acm.org/10.1145/1925861.1925878

[12] Q. Tang, S. K. S. Gupta, 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 19 (11) (2008) 1458–1472. doi:10.1109/TPDS.2008.111.

[13] C. Bash, 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] T. L. Bergman, F. P. Incropera, D. P. DeWitt, A. S. Lavine, Fundamentals of heat and mass transfer, John Wiley & Sons, 2011.

[15] S. MirhoseiniNejad, G. Badawy, D. G. Down, EAWA: Energy-aware workload assignment in data centers, in: 2018 International Conference on High Performance Computing & Simulation (HPCS), IEEE, 2018, pp. 260–267.

[16] S. MirhoseiniNejad, H. Moazamigoodarzi, G. Badawy, D. G. Down, Joint data center cooling and workload management: A thermal-aware approach, Future Generation Computer Systems 104 (2020) 174 – 186.

doi:https://doi.org/10.1016/j.future.2019.10.040.
URL http://www.sciencedirect.com/science/article/pii/S0167739X19302547

[17] H. Moazamigoodarzi, S. Pal, S. Ghosh, I. K. Puri, Real-time temperature predictions in IT server enclosures, International Journal of Heat and Mass Transfer 127 (2018) 890–900.

[18] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, C. Faloutsos, Thermocast: A cyber-physical forecasting model for datacenters, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, ACM, New York, NY, USA, 2011, pp. 1370–1378. doi:10.1145/2020408.2020611.
URL http://doi.acm.org/10.1145/2020408.2020611

[19] J. D. Moore, J. S. Chase, P. Ranganathan, Weatherman: Automated, online and predictive thermal mapping and management for data centers, in: Proceedings of the 3rd International Conference on Autonomic Computing, ICAC 2006, Dublin, Ireland, 13-16 June 2006, IEEE Computer Society, 2006, pp. 155–164.
URL http://ieeexplore.ieee.org/document/1662394/

[20] L. Wang, G. von Laszewski, F. Huang, J. Dayal, T. Frulani, G. Fox, Task scheduling with ANN-based temperature prediction in a data center: A simulation-based study, Engineering with Computers 27 (4) (2011) 381–391.

[21] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, J. S. Chase, Balance of power: Dynamic thermal management for Internet data centers, IEEE Internet Computing 9 (1) (2005) 42–49.

[22] J. D. Moore, J. S. Chase, P. Ranganathan, R. K. Sharma, Making scheduling "cool": Temperature-aware workload placement in data centers, in: Proceedings of the 2005 USENIX Annual Technical Conference, April 10-15, 2005, Anaheim, CA, USA, USENIX, 2005, pp. 61–75.
URL http://www.usenix.org/events/usenix05/tech/general/moore.html

[23] M. T. Chaudhry, T. Ling, S. A. Hussain, A. Manzoor, Minimizing thermal stress for data center servers through thermal-aware relocation, The Scientific World Journal 2014, 9 pages (2014).

[24] Z. Abbasi, G. Varsamopoulos, 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) 9 (2) (2012) 11:1–11:37.

[25] X. Zhao, Z. Xiong, L. Ding, X. Zhang, F. Xu, A smart coordinated temperature feedback controller for energy-efficient data centers, Future Generation Computer Systems 93 (2019) 506–514.

[26] Q. Fang, Q. Gong, J. Wang, Y. Wang, Optimization based resource and cooling management for a high performance computing data center, ISA transactions 90 (2019) 202–212.

[27] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. Gupta, S. Rungta, Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers, Computer Networks 53 (17) (2009) 2888–2904.

[28] Q. Wang, M. Song, Q. Fang, J. Wang, Thermal-aware flow field optimization

for energy saving of data centers, in: 2018 Annual American Control Conference (ACC), IEEE, 2018, pp. 3744–3749.

[29] Z. Wang, C. Bash, N. Tolia, M. Marwah, X. Zhu, P. Ranganathan, Optimal fan speed control for thermal management of servers, in: ASME 2009 InterPACK Conference collocated with the ASME 2009 Summer Heat Transfer Conference and the ASME 2009 3rd International Conference on Energy Sustainability, American Society of Mechanical Engineers Digital Collection, 2009, pp. 709–719.

[30] J. Yao, H. Guan, J. Luo, L. Rao, X. Liu, Adaptive power management through thermal aware workload balancing in internet data centers, IEEE Transactions on Parallel and Distributed Systems 26 (9) (2014) 2400–2409.

[31] D. Mauro, K. Schmidt, Essential SNMP: Help for System and Network Administrators, " O'Reilly Media, Inc.", 2005.

[32] J. LaCroix, Mastering Ubuntu Server: Master the art of deploying, configuring, managing, and troubleshooting Ubuntu Server 18.04, Packt Publishing Ltd, 2018.

[33] S. MirhoseiniNejad, F. M. García, G. Badawy, D. G. Down, ALTM: Adaptive learning-based thermal model for temperature predictions in data centers, in: 2019 IEEE Sustainability through ICT Summit (StICT), IEEE, 2019, pp. 1–6.

[34] A. Mellit, S. A. Kalogirou, L. Hontoria, S. Shaari, Artificial intelligence techniques for sizing photovoltaic systems: A review, Renewable and Sustainable Energy Reviews 13 (2) (2009) 406–419.

[35] A. Di Piazza, M. C. Di Piazza, G. Vitale, Solar and wind forecasting by NARX neural networks, Renewable Energy and Environmental Sustainability 1 (2016) 39.

[36] Q. Tang, T. Mukherjee, S. K. Gupta, P. Cayton, Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters, in: 2006 Fourth International Conference on Intelligent Sensing and Information Processing, IEEE, 2006, pp. 203–208.

[37] ASHRAE Technical Committee and others, Thermal guidelines for data processing environments - Expanded data center classes and usage guidance, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta, GA (2011).

Chapter 6

IT-aware cooling control framework for data centers: A machine learning control approach

This chapter is reproduced from "IT-aware cooling control framework for data centers: A machine learning control approach", SeyedMorteza MirhoseiniNejad, Ghada Badawy, and Douglas G. Down, submitted to IEEE Systems Journal, 2020.

The author of this thesis is the first author and the main contributor of this paper. His contributions to this work consist of introducing the main idea, implementing the system, writing the manuscript, formulating the control system, conducting the experiments, and constructing the algorithms.

Abstract

We present a complete system for the joint control of cooling units and workload assignment in a modular data center. The system aims to minimize power consumption while respecting temperature constraints, all in a thermally heterogeneous environment. Unlike traditional cooling controllers, our framework does not have a single set-point to satisfy. Instead, the system returns the thermal requirements of servers in the form of a temperature map and uses these thermal requirements as an input to the controller. We provide details of three phases of the system. First, a thermal model is built to predict temperatures within a data center. Second, the system assigns workloads to locations that are optimal to be cooled, considering the thermal effects of assigned workloads. Based on this assignment, a pattern for the required temperatures of servers is generated, called the required temperature distribution matrix (RTDM). The last phase uses model predictive control (MPC) to regulate the operational variables of cooling units in a power-efficient fashion to comply with the RTDM. Within each iteration of the MPC loop, an optimization problem involving the thermal model is solved, and the thermal model is updated. From an implementation on an actual modular data center, we find the potential for considerable power savings compared to other control methods.

Keywords: data center workload assignment, cooling unit control, thermal-aware scheduling, thermal model, data center power efficiency, efficient cooling, model predictive control, multi set-point control

6.1 Introduction

A massive portion of IT investment is toward data center development and expansion due to opportunities provided by the increasing use of mobile, cloud, and processing services [1]. This increase stems mainly from binding mobile applications and services to daily life, and migrating processes from end-user devices (such as laptops or cell phones for the sake of battery life, security, and integrity) to the server-side [2]. These changes have made data centers among the most power-hungry infrastructures [3].

Powering computing devices and cooling them are the two main sources of power consumption in data centers. There has been much research on the efficient use of the provided power to data centers. Power efficiency can be studied in different levels of data centers from small electronic components such as transistors and ICs to decisions on the geographical distribution of multiple data centers [4].

Cooling units present a notable opportunity to reduce power consumption since they are the second-largest power consumers (besides the servers themselves) in data centers [5, 6]. Cooling units are power inefficient, and over-cooling of servers is the primary cause of power wastage [7]. Decreasing over-cooling has been addressed extensively in the literature. In a number of studies, considering servers as heat sources is addressed through different methods of workload assignment [8, 9, 10, 11] to provide a uniform temperature distribution at the front or even back of servers. Reducing heat re-circulation to minimize the peak temperature and adjusting the cooling unit set-point accordingly are methods that involve both IT and cooling units to address the problem of over-cooling [12, 13].

In our previous work [14], we have addressed the importance of considering correlations between IT and cooling behaviors. In another work [15], we also showed

that the required inlet temperatures of servers could be determined as a function of their processing loads. Moreover, our suggested temperature estimation method [16] motivated us to put these different pieces together, in this paper, to construct a real-time holistic controller for a data center. This holistic controller, to a great extent, is capable of addressing thermal heterogeneity and the server over-cooling issue in data centers.

In this work, a system is constructed for holistic control of data center infrastructure in a power-efficient manner. The designed holistic temperature controller, unlike ordinary controllers, does not have just one reference temperature to regulate. Instead, it receives a set of required inlet temperatures of servers and tries to maintain these values using a power-efficient approach. The temperatures are calculated as the result of an optimization process. The optimization process, based on the thermal effects of servers and operational parameters of cooling units, determines the set of temperatures to minimize the cooling power consumption. The system uses a neural network model to consider the transient thermal effects of all contributing factors, including the IT and cooling equipment.

We use a *model predictive control* (MPC) algorithm and modify it to be able to satisfy the temperature requirements indicated by the set of optimized temperatures. The controller operates in real-time, and applies the optimized inputs to the system periodically, minimizing the cooling cost and satisfying temperature constraints. During the optimization, the thermal model is used for temperature predictions and evaluating the effects of future inputs. Temperature predictions are made using a time series method trained using a neural network. This model is capable of high precision predictions of the temperature of desired locations in a data center. The

accuracy and low complexity of this model are the crucial features that enable MPC to operate effectively and in real-time. The optimization process does not practically limit the duration of MPC iterations.

The system is implemented according to an algorithm that includes three phases; it starts with building the thermal model that is further used as the core of the workload assignment and control processes. In this work, the implementation process and the system under-study are described in detail. Our framework is compared with other methods of data center cooling unit control. The results are considerable power savings using our method.

In the next section, related work is reviewed. Our methodology is thoroughly explained in Section 6.3. This section describes the framework in detail, including explaining the algorithm, illustrating the system under-study, and describing the three components of the framework. Each component is studied in a subsection of Section 6.3. The results of the implementation of the framework on a data center are shown, compared with other methods, and discussed in Section 6.4 and Section 6.5. Finally, concluding remarks are provided in Section 6.6.

6.2 Literature review

An on-off thermostat based controller is the traditional way of controlling the temperature of IT equipment within data centers [17]. Controllers of this form turn the cooling on when the temperature exceeds a certain threshold and switches it off when the temperature reaches another (lower) threshold. Using this method typically results in poor performance in terms of both power consumption and the resulting output.

Proportional integral derivative or PID controllers are another simple method for adjusting the level of cooling in data centers [18]. A PID control loop employs system feedback of the current difference between the desired and the measured temperature to correct and adjust the inputs within each loop iteration. PID controllers use *proportional (P)*, *integral (I)*, and *derivative (D)* terms. P applies inputs proportional to the error, I considers the cumulative error, and D controls the current rate of change in the system [19].

Model predictive control or MPC is another approach that has also been suggested for cooling control in data centers [5, 20]. MPC relies on a dynamic model of the system; for obtaining the model, a system identification process is required. The strength of MPC comes from the fact that it determines the inputs for the current time-slot while taking into account the impact on future behavior. The future effects of the current inputs are determined from a model of the system [21]. MPC is an iterative process that calculates system inputs within each iteration. It repeatedly measures the status of the system and minimizes the cost of inputs over a time horizon. However, only the first time-step inputs are implemented, and in the next iteration, the whole process is repeated, and the prediction horizon is shifted forward. It is worth mentioning that this forecasting of future states of the system is an ability that PID controllers lack.

Several approaches use new control methods to control data center cooling units. DeepMind uses a machine learning approach to construct a model for power usage efficiency (PUE), which is, in turn, optimized by adjusting the cooling water temperature [22]. Lazic et al. [23] use the fan speeds and the water flow rate of the cooling unit to regulate the inlet temperatures of servers. They use reinforcement learning

for controlling the fans and valves. In both works, an MPC controller with a data-driven linear model is used. Although these methods employ a model for adjusting the operational variables of cooling units, they use just one set-point temperature, and the assignment of workload is not considered; in addition, the problem of server over-cooling is not addressed.

The closest work to ours is Kheradmandi et al. [20], who use MPC. In this work, controller feedback is provided by multiple sensor measurements, and each is taken into account for the next time-step control decisions (in particular, the individual measurements are all considered, not their average). The system identification process is performed via a physics-based model to generate data for constructing a linear model, and MPC is used to control the fan speeds of cooling units. This method does not consider the thermal effects of workload assignment, it uses limited control variables, and the process of thermal model construction is not scalable.

Cooling control of data centers can also involve other aspects, such as adjusting air exhaust vents. Raised floor data centers have ventilation tiles that allow cold air to blow to the front of servers from the floor. Zhou et al. [24] use MPC to control adaptive vent-floor tiles to handle the generated cold air efficiently. Different applications of cooling-related MPC with data-driven models have been studied in other types of cooling systems, such as building HVAC systems [25, 26, 27, 28].

Due to their complexity, creating thermal models for data centers has been a challenge. *Computational fluid dynamics* used to be the first choice for observing the thermal changes in data centers. However, they are computationally complex and unable to adapt to system changes; hence, using them inside MPC models is problematic [29]. Some simpler physics-based thermal models return the temperature

of critical points in data centers. For example, Moazamigoodarzi et al. [30] use a zonal-based physical model that generates the temperatures of thermal zones for a small scale data center. Their method considers heat-transfer differential equations and energy conservation laws between adjacent thermal zones to model the thermal dynamics. This thermal model is used by Kheradmandi et al. [20] (discussed earlier in this section). Although this method does not have the computational complexity of CFD methods, the model must be redeveloped for each application. However, the more significant issue is that such models are not scalable and do not adapt to changing thermal conditions within data centers.

Data-driven thermal models are the new generation of system identification for MPC controllers. ThermoCast [31] is a lighter version of the full-fledged fluid mechanics equations. It infers the IT load using a machine learning approach based on data obtained from air-flow and temperature sensors. The calculated IT load is then combined with physical laws to predict temperatures. An objection to this method is that it needs accurate temperature and air-flow measurements from surrounding servers, and this method requires what may be an excessive number of sensor installations. Moore and Ranganathan [10] use neural networks for steady-state temperature estimation based on the IT profile, air flow-rates, supply air temperatures, and geometric data of a data center. The main issue with this approach is that it requires a large number of steady-state data points. Additionally, the geometric data, such as locations of compartments, walls, and servers, does not allow the model to be adaptive to any physical changes. In our previous work [16], we provide a framework for constructing a transient data center thermal model using neural networks. This is a data-driven model that uses the workload profile and operational variables for

its temperature predictions. The model is adaptive to the thermal changes, easy to implement, and suitable to be used in MPC loops.

6.3 The methodology

We build a holistic control framework that provides the required inlet temperatures for servers in a data center. This framework incorporates workload assignment and cooling control to cool servers efficiently, based on their current cooling requirements and the existing thermal heterogeneity in the data center environment.

This framework consists of three major components: *data center thermal model generation*, *optimal zone selection*, and the *model predictive control (MPC) loop*. The first component generates the data center thermal model to predict the inlet temperatures of servers. The second component calculates the required inlet temperatures of servers. It determines the optimal locations in a data center to cool and assigns workload based on the current offered load. This component provides the temperature requirements of servers as a temperature distribution map (RTDM) to the next component. The final component controls the cooling unit to satisfy the given RTDM. These three components are embedded in an algorithm for holistic control of a data center. In this section, we describe the system under-study (Section 6.3.1) before detailing the framework (Section 6.3.2).

6.3.1 System description

Our framework is implemented in our on-site data center (Fig. 6.1). The data center consists of five IT racks and is cooled using two in-row cooling units installed at the

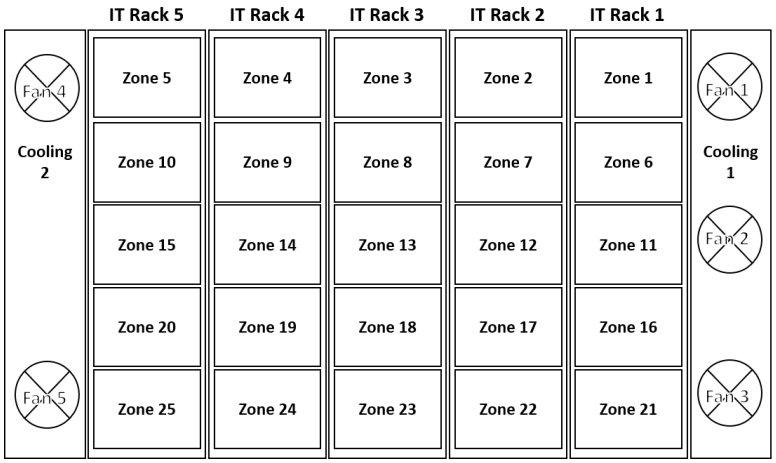


Figure 6.1: Data center front view: location of fans and arrangement of thermal zones

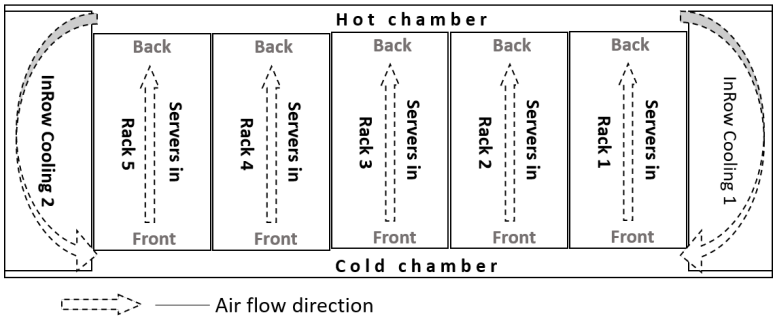


Figure 6.2: Top view of the data center

sides. The cooling units use chilled water provided by a chiller installed outside of the building. Fig. 6.2 shows the top view of the data center; cooling units receive hot air from the zone at the back of the servers (hot chamber) and provide cold air to the front zone of the servers (cold chamber).

Each cooling unit is equipped with several fans (five fans in total, see Fig. 6.1). Fans facilitate air circulation inside the data center. The cold chamber volume at the front of servers is divided into 25 thermal zones, in a five by five grid. There are five IT racks (five columns), and the height of each IT rack is considered as five equal

Figure 6.3: DS18B20 digital thermometer

height thermal zones (five rows). The zone arrangement is shown in Fig. 6.1. Due to the negligible temperature difference throughout a thermal zone, we assume that the temperature is uniform within each zone.

Control of cooling units is achieved through the Simple Network Management Protocol (SNMP). SNMP is a widely used protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change the device behavior [32]. Data center cooling units typically support SNMP. Using the Linux shell tools IPMI [33] and TOP [34], we can manage and monitor the server load via SSH scripts [35]. Twenty-five temperature sensors are installed at the centers of the thermal zones. We used DS18B20 digital thermometers, which have $\pm 0.5^{\circ}C$ accuracy from $-10^{\circ}C$ to $+85^{\circ}C$ and provide programmable resolution from 9 bits to 12 bits. Fig. 6.3 shows an installed temperature sensor. Temperature data is collected using a Raspberry Pi™.

```
INITIALIZATION
  timeStep; % Sampling and control time-step in seconds
  d; % Total offered load (percent)
  horizon; % The prediction horizon of the controller
  MODELparam = {PRDparam,TRAINparam};
THERMAL MODEL GENERATION
  dt = dataAcquisition(timeStep);
  dcThermalModel = modelGen(dt, MODELparam)
OPTIMAL ZONE SELECTION
  (inputMatrix,outputMatrix) = systemMonitor();
  RTDM = zoneOptimizer(dcThermalModel,d,inputMatrix,outputMatrix);
MPC LOOP
 time0 = clock(); % Current time of the system
 while true do
    (currentOutput,currentInput) = sensorRead();
    inputMatrix(end+1,:) = currentInput;
    outputMatrix(end+1,:) = currentOutput;
    dt = prepare(dcThermalModel,inputMatrix,outputMatrix);
    dcThermalModel = adapt(netc,dt);
    controlVars = optimizer(RTDM,horizon,dcThermalModel, inputMatrix,
   outputMatrix);
    pause(timeStep - clock() - time0);
    time0 = clock();
    applyInputs(controlVars);
end
```

Algorithm 4: Data center holistic control algorithm

6.3.2 The framework

The designed framework for the holistic control of a data center is summarized in Algorithm 4. The algorithm consists of three components, *data center thermal model generation*, *optimal zone selection*, and the *model predictive control (MPC) loop*.

Algorithm 4 first initializes a number of essential parameters. The parameter *timeStep* is the time between two consecutive control commands; it is also used as the sample time for collecting training data. The parameter d is the total server demand

or offered load to the data center in percentage of maximum, and *horizon* indicates the number of time-steps for output predictions used by the controller. Input and output feedback delay size are the important parameters of time series predictions (explained in Section 6.3.2) and are given by *PRDparam*. *NNparam* includes the size of the hidden layers, training algorithms and portions of training, validation, and test data to train the model. We choose the *Levenberg-Marquardt* back-propagation method, 20 neurons in one hidden layer, and [75%,15%,10%] as the proportions of training, validation, and testing data.

After the initialization, the algorithm generates the data center thermal model using two functions *dataAcquisition()* and *modelGen()*. The function *dataAcquisition()* generates data to train the model. It reads raw data from the sensors and returns ready-to-use data via the variable *dt*. The function *modelGen()* has two inputs, *dt* and *MODELparam*. *MODELparam* has the required parameters for both the time-series prediction process and the neural network training process.

In the next component, the algorithm decides on workload assignment and generates the required temperature distribution map (RTDM) to be used by the controller. It first monitors the system, using *systemMonitor()*, which provides the initial values for the thermal model. The function *zoneOptimizer()* returns the RTDM. The calculation of the RTDM requires the offered load d, the thermal model, and the initial values of the model in the form of two matrices. The function *prepare()* prepares the new data to be used by *adapt()* to update the model. We note that rolling back the updated model if the re-trained model is affected adversely can be internally embedded in the *adapt()* function. However, for the sake of simplicity, we do not consider this in our implementation.

The last component of the algorithm is the MPC loop, which takes the RTDM, and with the help of the thermal model, controls the system inputs through solving an optimization problem. In the beginning of the MPC loop (the while loop), *sensorRead()* reads the current status of the system and the new readings are stored in *outputMatrix* and *inputMatrix*.

The key functionality of the MPC loop is the optimization of controllable variables, performed by the function *optimizer()*. The function *optimizer()* employs the thermal model and updated inputs and outputs of the system in an optimization problem (explained in Section 6.3.2) and returns the optimal inputs to be applied to the system via *applyInputs()*. Briefly, the objectives of this optimization problem are minimizing the costs of both the inputs and input fluctuations, while being constrained by the required server inlet temperatures given by the RTDM. Just before applying the inputs, the timing for the control loop is handled. It uses the function *clock()* which returns the current system time. The function *pause()* holds the execution of the algorithm until the next time-step. The different components of the algorithm are explained in more detail as follows:

First component - Construction of the data center thermal model

The data center thermal model estimates the temperature of different locations in a data center, with a focus on the front of servers. The thermal model that is used in this work takes the operational parameters and the current thermal status and returns the temperature estimates in the next time-step. The operational parameters of the data center are the inlet water temperature, the chilled water flow rate, the speed of cooling units' fans, and workloads of servers. There are two kinds of variables in this

system, state variables and controllable variables. State variables are those that affect the air temperature but are not controllable, such as the water inlet temperature to the cooling units. The controllable variables, such as fan speeds, are adjusted by the controller. For our purposes, the data center environment is the system, the operational variables are the system inputs, and the air temperatures of the different thermal zones are the system outputs.

Specifically, the first component of Algorithm 4 is the thermal model generation. This component consists of *data acquisition* (via the dataAcquisition() method) and *model generation* (via the modelGeneration() method). For simplicity, we are not going to describe the details of the functions, but we provide a general description of their functionality. Data acquisition is performed by applying the pseudo-random bit stream (PRBS) method to the inputs of the system [36], which is a suitable method for exploring the state space of this multiple input system. This process might be different in different settings.

The method modelGen() takes raw data returned by dataAcquisition() to build the thermal model. We use a nonlinear autoregressive exogenous (NARX) model for time series prediction. The characteristics of the feedback and neural network are given as the model parameters to modelGen(). The thermal model should be able to predict the system output based on the current status of the system, the previous inputs, and the previous outputs. In our earlier work [14, 16], we showed that the use of neural networks for temperature prediction in data centers is a suitable method, with several advantages over existing methods.

Second component - Optimal temperature distribution

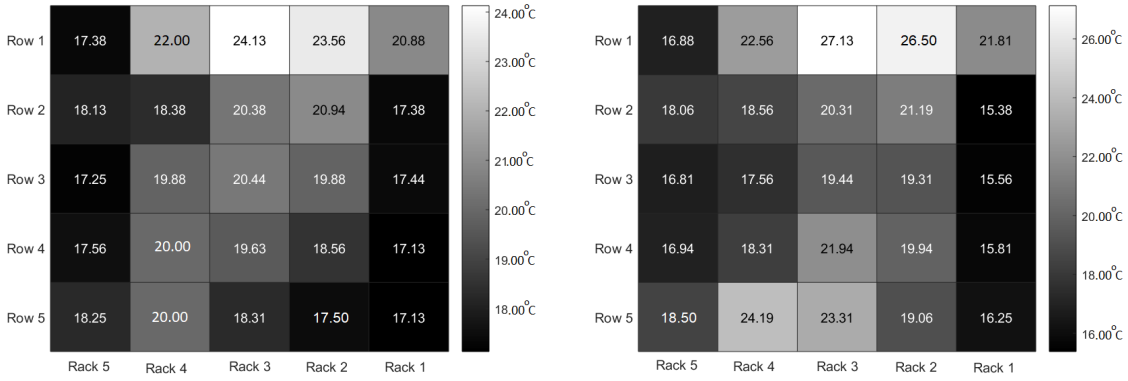
One of the important contributions of this work is that unlike traditional control methods, our algorithm does not have a single set point. Specifically, a required temperature distribution map (RTDM) must be satisfied in a power-efficient manner by the co-operation of all of the cooling units. The RTDM is obtained based on the steady-state thermal effects of workload assignment and cooling cost. Two different combinations of a set of temperatures might result in different cooling costs. For example, suppose two RTDMs, R_1 and R_2, are given by

$$R_1 = \begin{bmatrix} 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 35°C & 35°C & 35°C & 35°C & 35°C \end{bmatrix}$$

and

$$R_2 = \begin{bmatrix} 35°C & 35°C & 35°C & 35°C & 35°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \end{bmatrix}.$$

The optimal control solution to satisfy R_1 and R_2 results in the temperature distributions shown in Fig. 6.4. The figure shows that the solution to cool servers based on R_2 can be achieved in a more power-efficient manner than R_1. The difference between R_1 and R_2 is that the former allows the bottom row of thermal zones to have



(a) Inlet temperatures of servers corresponding to R_1

(b) Inlet temperatures of servers corresponding to R_2

Figure 6.4: Heat-map representation of inlet temperatures of servers based on different RTDMs

Table 6.1: Comparison of R_1 and R_2

	Fan speed in percent (%)					Power
	Fan 1	Fan 2	Fan 3	Fan 4	Fan 5	(W)
R_1	74	23	63	74	40	3813
R_2	55	27	0	81	50	3095

higher temperatures (35°), but the latter allows the top row of the thermal zone to have higher temperatures.

Table 6.1 shows the optimal parameters corresponding to each of the RTDMs. The power consumption corresponding to R_1 is $3813W$, as opposed to $3095W$ for R_2. Because of the tendency of hot air to rise, the better results for R_2 appear intuitive; however, the optimal solution is not always so obvious. This example clearly shows that there is a trade-off in the combinations of the required temperatures given by an RTDM.

The previous example suggests the feasibility of having an optimal RTDM. We

$$\underset{\bar{\rho},\bar{v}_{opr}}{\text{minimize}} \quad P_{cooling} \tag{6.3.1a}$$

$$\text{subject to:} \quad P_{cooling} = \mathcal{F}(\bar{v}_{opr}) \tag{6.3.1b}$$

$$\rho_i \in \{0,1\}, \quad for \quad i = 1,\ldots,n \tag{6.3.1c}$$

$$\sum_{i=1}^{n} \rho_i \geq d \cdot n \tag{6.3.1d}$$

$$\texttt{LowerBound} \leq \bar{v}_{opr} \leq \texttt{UpperBound} \tag{6.3.1e}$$

$$\bar{T}_{inlet} = \mathcal{M}(\bar{v}_{opr}, \bar{\rho}) \tag{6.3.1f}$$

$$\bar{T}_{inlet} \leq \texttt{RTDM} \tag{6.3.1g}$$

$$\bar{z} = \texttt{zoneMap}(\bar{\rho}) \tag{6.3.1h}$$

$$\texttt{RTDM} = T_{idle} - (T_{idle} - T_{busy}) \cdot \bar{z} \tag{6.3.1i}$$

formulate an optimization problem that minimizes the cooling power based on an optimal adjustment of operational variables of cooling units and assignment of workload. Considering the optimal workload assignment, an RTDM is generated. Entries of the RTDM are calculated based on workloads of servers of corresponding thermal zones. We have assumed that a server could only be either idle ($\rho_i = 0$) or always busy ($\rho_i = 1$). Based on our previous work [15], the required inlet temperature of servers can be determined due to their load. So, two different red-line temperatures, T_{idle} and T_{busy}, are considered for idle and busy servers, respectively. The red-line temperature is an upper bound for the inlet temperature of servers. In [15], we showed that an idle server could operate with a relatively higher red-line temperature than a busy server. Using the thermal model described in Section 6.3.2 an optimization problem respecting all of the practical constraints can be formulated as in (6.3.1):

In this minimization problem, decision variables consist of two types, the assignment of workload ($\bar{\rho}$) and operational variables of cooling units ($\bar{v}_{opr}$). The binary vector $\bar{\rho}$ contains the CPU utilization of servers whose elements (ρ_i) are either 0 (idle

server) or 1(busy server). $P_{cooling}$ is the cooling power consumption, obtained using $\mathcal{F}()$, which is calculated based on the vector of operational variables ($\bar{v}_{opr}$). In our case, the vector $\bar{v}_{opr}$ consists of the fan speeds; it can be different according to the controllable variables for various data centers. Our system uses the cooling power model given in [14].

Equation (6.3.1c) constrains the utilization of each server to be either 0 or 1 and (6.3.1d) ensures that the assigned capacity is not less than the total demand $d \cdot n$ (n is the total number of servers). The constraint (6.3.1e) confines the choices of operational variables between feasible values of lower-bounds and upper-bounds. For example, fan speeds should have a lower bound of 0% and an upper bound of 100%. In (6.3.1f), the vector $\bar{T}_{inlet}$ contains inlet air temperatures of the servers and $\mathcal{M}$ represents the thermal model that predicts inlet temperatures. The component-wise inequality in (6.3.1g) ensures that estimated temperatures are not greater than the corresponding elements in the RTDM. Without loss of generality, the RTDM is the vector version of the matrix in this optimization problem.

The auxiliary binary vector $\bar{z}$, used in (6.3.1h), encodes the active thermal zones and is used to calculate the RTDM. A thermal zone is active when at least one of its corresponding servers is busy, indicated by 1 in the corresponding element of $\bar{z}$. In this equation, the function *zoneMap()*, using the map of servers and thermal zones, returns the vector $\bar{z}$ based on $\bar{\rho}$. The vector $\bar{z}$ is used in (6.3.1i) for writing T_{idle} and T_{busy} values in the RTDM according to the respective values of 0 and 1 read from $\bar{z}$.

As discussed in Section 6.5, the assignment of workload could be performed with higher precision beyond considering servers to be idle or always busy. Consequently,

$$\underset{\substack{u(k+i|k),\Psi(i);\\ i=0,\cdots,m-1}}{\text{minimize}} \quad \sum_{j=0}^{m-1} \Big(w_1 f_1(\Psi(j)) + w_2 \|\Delta u(k+j|k)\|_2^2 + w_3 f_2(u(k+j|k)) \Big) \tag{6.3.2a}$$

$$\text{subject to: } u_{min} \leq u(k+j|k) \leq u_{max}, j = 0, \cdots, m-1 \tag{6.3.2b}$$

$$\Delta u_{min} \leq \Delta u(k+j|k) \leq \Delta u_{max}, j = 0, \cdots, m-1 \tag{6.3.2c}$$

$$y(k+j|k) = \mathcal{M}(u(k+j|k)), j = 0, \cdots, m-1 \tag{6.3.2d}$$

$$y_{min} \leq y(k+j|k) \leq \texttt{RTDM} + \Psi(j), j = 0, \cdots, m-1 \tag{6.3.2e}$$

generating the RTDM would need be modified in (6.3.1h) and (6.3.1i). This modification requires precise thermal models of servers. More accurate required inlet temperatures of servers can be calculated by including thermal models of servers; this increases the complexity of (6.3.1). The current solution to the problem, to a great extent, is capable of representing our idea, while the refined RTDM is proposed for future work.

Third component - The control loop

Once the RTDM is obtained, it is passed to the controller. We build an MPC controller that uses the thermal model, described in Section 6.3.2, for its internal calculation and optimization. This controller performs real-time optimization (6.3.2) in each time-step to obtain optimal inputs to apply to the operational variables of the cooling unit. The thermal model empowers the optimizer to estimate the output trajectory.

Zanin et al. [37] and De Souza et al. [38] showed the integration of real-time optimization into MPC. We modified this optimization to be compatible with the definition of the RTDM in (6.3.2).

In this equation, u and y are the system input and output, respectively, and m is the prediction horizon. The optimal control input $u(k+j|k)$ should be applied to the

system at time-step $k+j$; however, only $u(k+1|k)$ is actually applied to the system. The vector $y(k+j|k)$ is the output estimate at the future time-step $k+j$.

The cost function of the optimization problem is the sum of three terms, each with a non-negative weight (w_i). The cost of the slack variable $\Psi(j)$ used in (6.3.2e) is calculated by $f_1(\Psi(j))$. The reason for using the slack variable $\Psi(j)$ is to guarantee feasibility of the optimization problem. Another term in the cost function is $\|\Delta u(k+j|k)\|_2^2$ which measures the difference between two consecutive inputs, or input fluctuation, where $\Delta u(j+k|k) = u(j+k|k) - u(j+k-1|k)$. Limiting input fluctuations is of interest to extend device lifetimes [39]. The last term of the cost function is $f_2(u(k+j|k))$ which calculates the cost of inputs. Equation (6.3.2d) uses the thermal model $\mathcal{M}()$ for temperature predictions.

As shown in Algorithm 4, this optimization problem should be solved within each control loop. So, time complexity is essential. The *interior-point* algorithm is used to solve this nonlinear optimization problem. The solution of the *interior-point* method is similar to other algorithms, such as *sequential quadratic programming*, however, with fewer iterations. The number of iterations during the optimization process is limited to have a solution within the control interval (20 seconds). In our experiments, the solution is obtained more than 90% of the time without reaching the execution limit.

6.4 Results

In this section, the performance of the framework is evaluated by implementing it on a real system (the system was described in Section 6.3.1). The evaluation is performed for two scenarios. In the first scenario, an RTDM with identical entries is used by

the controller. In other words, the goal is to keep the temperature of all zones below a certain threshold. In the second scenario, an optimal RTDM is provided to the controller. This allows us to measure the potential gains from using a heterogeneous RTDM. Finally, both scenarios are compared with set-point-tracking controllers.

During the system implementation, red-line temperatures of T_{busy} and T_{idle} are set to $24°C$ and $35°C$, respectively, appropriate values for the servers in our data center. The red-line temperatures may need to be adjusted for a particular application. To perform a fair comparison, we started with the same initial conditions; the speed of all fans is set to 100% in the first few time-steps. The figures depicting the temperature of thermal zones (system output) show this effect in the initial time-steps. The controller solves an optimization problem during each time-step, and then optimal inputs are applied to the system at the beginning of the next time-step. The duration of the time-steps depends on the system and the resulting complexity of the optimization problem. For our problem, we found a time-step of 20 seconds is reasonable for the control loop calculations and the dynamics of the system.

First, the output of the system (temperature of the thermal zones) is shown when the RTDM used by the controller (the third component of Algorithm 4) is a five by five matrix with all of its elements equal to T_{busy}. This means that the controller should adjust system inputs to keep all the server inlet temperatures below $24°C$. Fig. 6.5 shows temperature variations of the 25 thermal zones over time.

After the first few time-steps, the controller is activated and adjusts the system inputs to keep the inlet temperatures of servers below the RTDM entries while minimizing the cooling cost. Fig. 6.6 shows the variation of inputs (fan speeds) over time. The maximum possible fan speeds are employed at the beginning until all outputs are

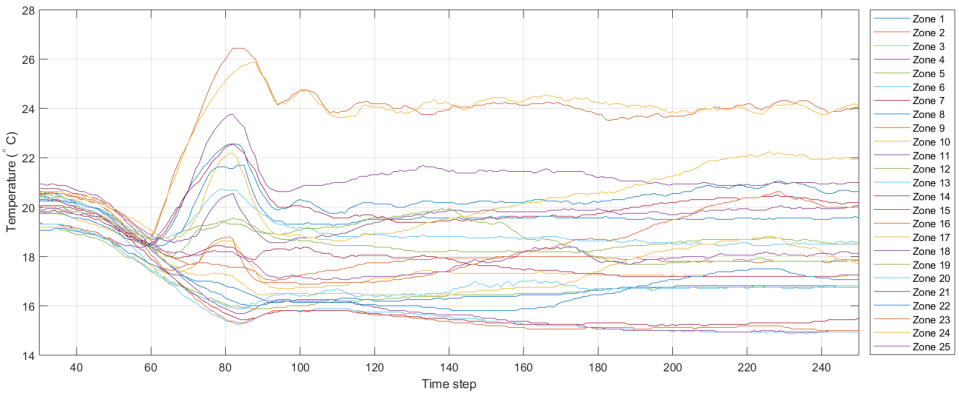


Figure 6.5: System outputs (first scenario) - Temperature of 25 thermal zones

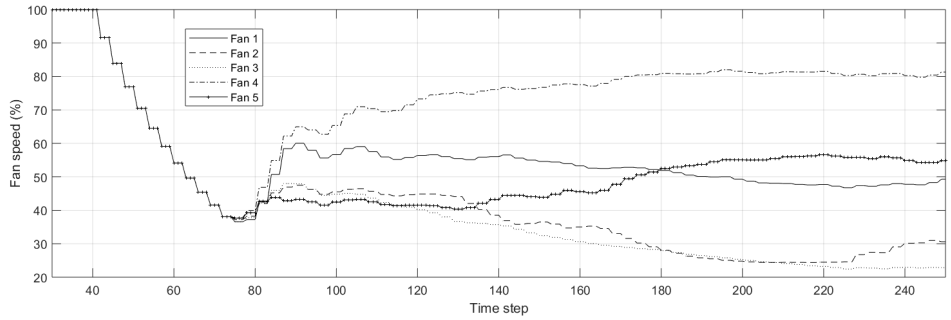


Figure 6.6: System inputs (first scenario) - Fan speeds

below $24°C$. This happens due to the dominating cost imposed by the slack variable (Ψ) in (6.3.2b). When the measured temperatures of thermal zones are above the corresponding RTDM entries, it is reflected in the slack variable in (6.3.2e), forcing the controller to decrease these temperatures. However, weights (w_i) in (6.3.2b) should be carefully tuned to prevent either slow or sudden reactions to suppress the slack variable [38]. As shown, measured temperatures become relatively steady over time, and the maximum temperature of thermal zones reaches $24°C$. Fig. 6.7 provides the cooling power consumption corresponding to the outputs shown in Fig. 6.5.

In the next experiment, an RTDM is obtained using the second component of our

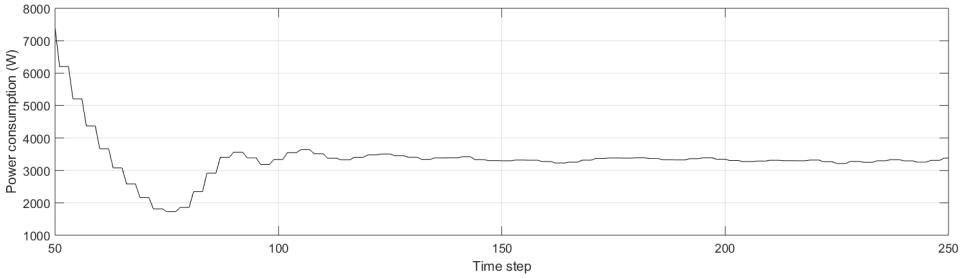


Figure 6.7: Cooling power consumption (first scenario)

framework. It uses the optimization process (6.3.1) while d is 75%. The RTDM is obtained as the following:

$$RTDM = \begin{bmatrix} 35°C & 35°C & 35°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \\ 24°C & 24°C & 24°C & 24°C & 24°C \end{bmatrix}.$$

As shown in the matrix above, zones 1, 2, and 3 (the zone numbering is shown in Fig. 6.1) are those that are set to be hotter than the rest of the thermal zones. Fig. 6.8 and Fig. 6.9 show the temperatures and the corresponding inputs using the optimized RTDM. Fig. 6.8 clearly shows that the temperatures of the three zones (zones 1, 2, and 3) become greater than T_{busy}, while the temperatures of the other zones are kept below T_{busy}, as desired. The controller adjusts inputs in a manner that respects the different temperatures required by the RTDM entries. This ability is given to the controller by embedding the thermal model of desired thermal zones in the MPC process. Fig. 6.10 depicts the corresponding power consumption of the cooling units.

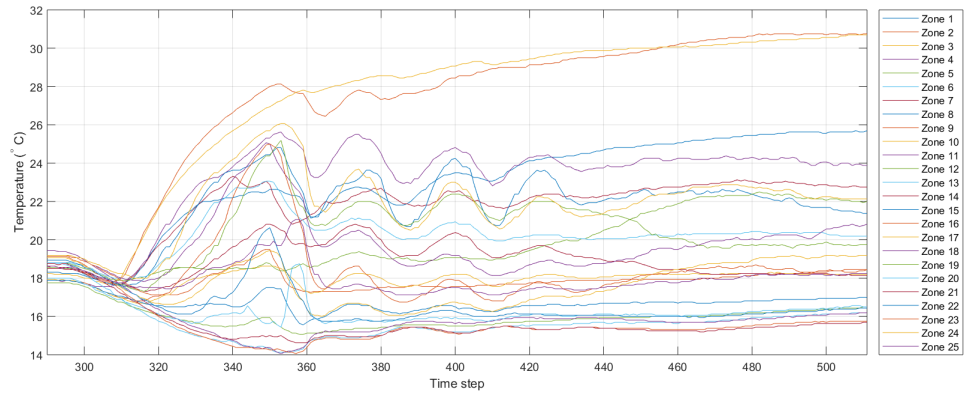


Figure 6.8: System outputs (second scenario)- Temperature of 25 thermal zones

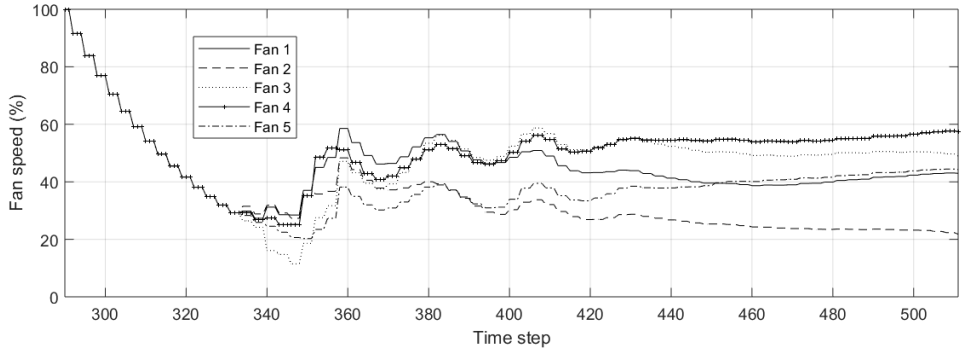


Figure 6.9: System inputs (second scenario) - Fan speeds

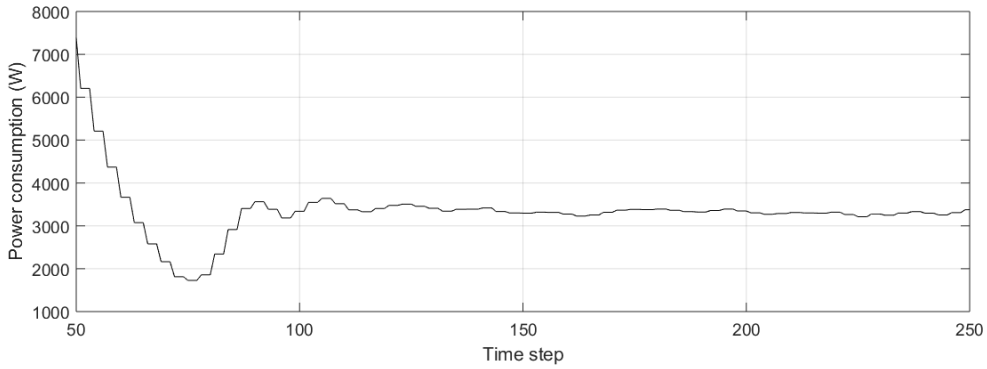


Figure 6.10: Cooling power consumption (second scenario)

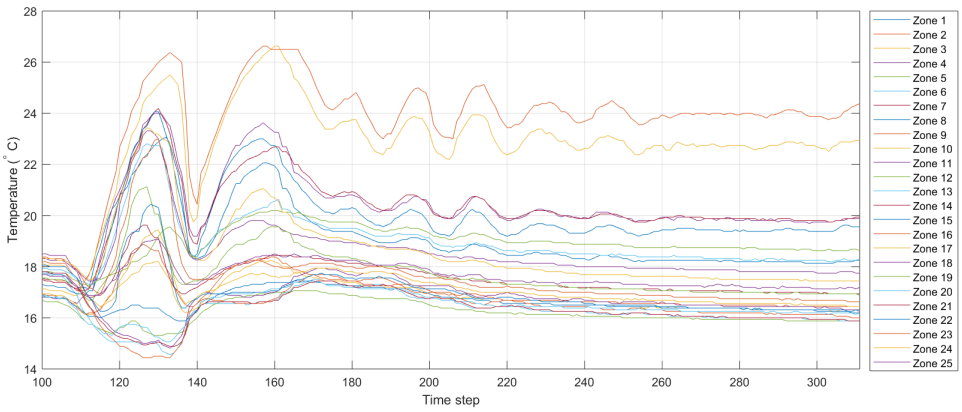


Figure 6.11: System outputs (set-point-tracking) - Temperature of 25 thermal zones

We compare our method with set-point-tracking control methods. Almost all methods for cooling unit control try to meet a set-point temperature while the deviation from that reference is minimized. We implemented a model predictive control (MPC) approach for this comparison. The MPC controller uses the temperature of a single point for the control decisions. For our experiments, this point is the middle top point in the data center, (typically) the hottest spot in the data center. The given set-point for the MPC controller is T_{busy}. We call it the *set-point-tracking* controller in our comparisons. Fig. 6.11 shows the temperature of thermal zones resulting from running the set-point-tracking controller. The corresponding inputs are shown in Fig. 6.12.

The average temperature of thermal zones in the data center is one way to measure the level of cooling required. This average corresponding to the first (Fig. 6.5) and the second (Fig. 6.8) scenarios is 18.48 and 20.02, respectively. However, the average temperature of thermal zones is 17.66 for the set-point-tracking control method (Fig. 6.11)–meaning that it has increased over-cooling of servers as compared to our methods.

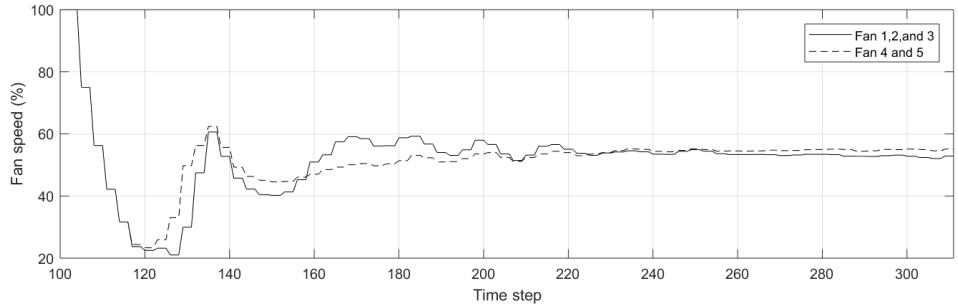


Figure 6.12: System inputs (set-point-tracking) - Fan speeds

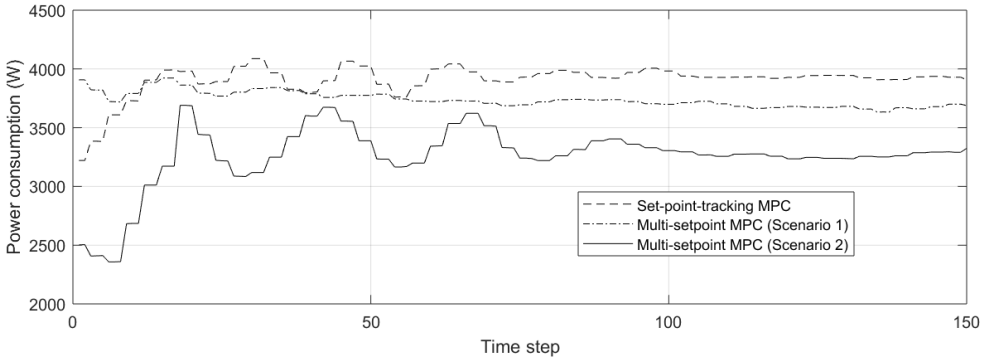


Figure 6.13: Comparing power consumption

Fig. 6.13 compares the cooling power consumption of our framework and the set-point-tracking controller. The power consumption of our framework without an optimized RTDM (first scenario) and with an optimized RTDM (second scenario) are represented using dashed-line and solid-line curves, respectively. The power consumption of the *set-point-tracking* controller is also drawn via a dash-dot-line curve in Fig. 6.13. This figure shows that our framework consumes less power than the set-point-tracking method. It also shows that using the framework with and without an optimal RTDM is preferred; however, an optimal RTDM can save a considerable amount of power.

6.5 Discussion

Our proposed algorithm for cooling the data center environment is capable of a significant reduction in power consumption by respecting IT thermal requirements and having a model for the effects of cooling units. It is observed that equipping thermal zones (as described in Section 6.3.1) with temperature sensors and controlling cooling units through a learning-based thermal model can reduce cooling power consumption significantly.

The introduced system identification methods, RTDM generator, and the controller are not the only methods of implementing the details of this framework. For example, in a large up-and-running data center, the process of data accumulation and constructing models might be different. The resolution of thermal zones can vary from server-size to rack-size zones. Depending on the application, the cost function for the control loop can be different, especially with respect to the weights and functions (f_i).

If there are considerable changes in the offered load, the RTDM calculation can be called as needed. Additionally, calculating the RTDM by using only two thresholds ($24°C$ and $35°C$) could be considerably improved. A function for calculating the exact thermal requirements of a server using its thermal model can be used to obtain the RTDM.

There are a number of suggestions for future work. The implementation of reinforcement learning can be considered for workload assignment and cooling control with respect to data center thermal heterogeneity. The assignment of workload to servers could be performed with greater granularity than considered in this work. Moreover, the thermal demands of the combination of workload type and server hardware can be studied. One promising enhancement to this work could be considering

constraints on the core temperatures of servers rather than their inlet temperatures.

6.6 Conclusion

As opposed to computationally expensive physics-based models, learning-based and data-driven thermal models can provide accurate temperature prediction tools suitable for real-time control. Their adaptability and low computational complexities empower new control methods to be applied to complicated temperature-sensitive environments such as data centers. We devised a novel control approach using these techniques and changed the notion of set-point in this context. The to-be-cooled volume is divided into several thermal zones, and an optimal temperature requirement is determined for each zone. The optimized temperature requirements of servers are provided to a controller which controls the cooling units. All the processes, from generating the thermal model to applying the control inputs to the system, are included in the framework. Implementing the framework on a data center with in-row cooling shows the potential for considerable power savings compared to other popular controllers.

Acknowledgment

This research was supported by a Collaborative Research and Development grant CRDPI506142-16 from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Bibliography

[1] A. Shehabi, S. J. Smith, E. Masanet, and J. Koomey, "Data center growth in the United States: Decoupling the demand for services from electricity use," *Environmental Research Letters*, vol. 13, no. 12, pp. 1–12, 2018.

[2] S. Umair, U. Muneer, M. N. Zahoor, and A. W. Malik, "Mobile cloud computing future trends and opportunities," in *Managing and Processing Big Data in Cloud Computing*, pp. 105–120, IGI Global, 2016.

[3] H. Klemick, E. Kopits, and A. Wolverton, "How do data centers make energy-efficiency investment decisions? Qualitative evidence from focus groups and interviews," *Energy Efficiency*, vol. 12, pp. 1359–1377, June 2019.

[4] G. Varsamopoulos, Z. Abbasi, and S. K. Gupta, "Trends and effects of energy proportionality on server provisioning in data centers," in *2010 International Conference on High Performance Computing*, pp. 1–11, IEEE, 2010.

[5] J. Dai, M. M. Ohadi, D. Das, and M. G. Pecht, *Optimum cooling of data centers*. Springer, 2016.

[6] R. Sawyer, "Calculating total power requirements for data centers, whitepaper," in *Power Conversion*, pp. 1–10, Schneider Electric's Data Center Science Center, 2004.

[7] J. Loper and S. Parr, "Energy efficiency in data centers: A new policy frontier," *Environmental Quality Management*, vol. 16, no. 4, pp. 83–97, 2007.

[8] 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.

[9] M. T. Chaudhry, T. C. Ling, S. A. Hussain, and A. Manzoor, "Minimizing thermal stress for data center servers through thermal-aware relocation," *The Scientific World Journal*, vol. 2014, pp. 1–9, Mar. 2014.

[10] J. D. Moore, J. S. Chase, and P. Ranganathan, "Weatherman: Automated, online and predictive thermal mapping and management for data centers," in *Proceedings of the 3rd International Conference on Autonomic Computing, ICAC 2006, Dublin, Ireland, 13-16 June 2006*, pp. 155–164, IEEE Computer Society, 2006.

[11] C. Bash and G. Forman, "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center," in *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, ECBS '19, (USA), pp. 19:1–19:37, USENIX Association, 2007.

[12] Z. Abbasi, G. Varsamopoulos, and S. 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, pp. 11:1–11:37, 2012.

[13] 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, pp. 1458–1472, Nov. 2008.

[14] S. MirhoseiniNejad, H. Moazamigoodarzi, G. Badawy, and D. G. Down, "Joint data center cooling and workload management: A thermal-aware approach," *Future Generation Computer Systems*, vol. 104, pp. 174 – 186, 2020.

[15] S. MirhoseiniNejad, G. Badawy, and D. G. Down, "EAWA: Energy-aware workload assignment in data centers," in *2018 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 260–267, IEEE, 2018.

[16] S. MirhoseiniNejad, F. M. García, G. Badawy, and D. G. Down, "ALTM: Adaptive learning-based thermal model for temperature predictions in data centers," in *2019 IEEE Sustainability through ICT Summit (StICT)*, pp. 1–6, IEEE, 2019.

[17] D. H. Zervos, "On-off thermostat based modulating air flow controller," Dec 1985. Google (US) Patent 4,556,169, [Online]. Available: `https://patents.google.com/patent/US4556169`.

[18] B. Durand-Estebe, C. Le Bot, J. N. Mancos, and E. Arquis, "Data center optimization using PID regulation in CFD simulations," *Energy and Buildings*, vol. 66, pp. 154–164, 2013.

[19] D. E. Rivera, M. Morari, and S. Skogestad, "Internal model control: PID controller design," *Industrial & Engineering Chemistry Process Design and Development*, vol. 25, no. 1, pp. 252–265, 1986.

[20] M. Kheradmandi, D. G. Down, and H. Moazamigoodarzi, "Energy-efficient data-based zonal control of temperature for data centers," in *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*, pp. 1–7, Oct 2019.

[21] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice: A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[22] J. Gao, "Machine learning applications for data center optimization," *Google White Paper*, pp. 1–13, 2014. [Online]. Available: `https://research.google/pubs/pub42542.pdf`.

[23] N. Lazic, C. Boutilier, T. Lu, E. Wong, B. Roy, M. Ryu, and G. Imwalle, "Data center cooling using model-predictive control," in *Advances in Neural Information Processing Systems*, pp. 3814–3823, 2018.

[24] R. Zhou, C. Bash, Z. Wang, A. McReynolds, T. Christian, and T. Cader, "Data center cooling efficiency improvement through localized and optimized cooling resources delivery," *ASME International Mechanical Engineering Congress and Exposition*, vol. Volume 7: Fluids and Heat Transfer, Parts A, B, C, and D, pp. 1789–1796, 11 2012.

[25] J. D. Feng, F. Chuang, F. Borrelli, and F. Bauman, "Model predictive control of radiant slab systems with evaporative cooling sources," *Energy and Buildings*, vol. 87, pp. 199–210, 2015.

[26] A. Kelman and F. Borrelli, "Bilinear model predictive control of a HVAC system using sequential quadratic programming," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9869–9874, 2011.

[27] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, 2011.

[28] Y. Ma, A. Kelman, A. Daly, and F. Borrelli, "Predictive control for energy efficient buildings with thermal storage: Modeling, stimulation, and experiments," *IEEE Control Systems Magazine*, vol. 32, no. 1, pp. 44–64, 2012.

[29] T. L. Bergman, F. P. Incropera, D. P. DeWitt, and A. S. Lavine, *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[30] H. Moazamigoodarzi, S. Pal, S. Ghosh, and I. K. Puri, "Real-time temperature predictions in IT server enclosures," *International Journal of Heat and Mass Transfer*, vol. 127, pp. 890–900, 2018.

[31] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos, "Thermocast: A cyber-physical forecasting model for datacenters," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, (New York, NY, USA), pp. 1370–1378, ACM, 2011.

[32] D. Mauro and K. Schmidt, *Essential SNMP: Help for System and Network Administrators*. " O'Reilly Media, Inc.", 2005.

[33] "Display IPMI sensor information." Ubuntu Manpage Repository, [Online]. Available: `http://manpages.ubuntu.com/manpages/xenial/man8/ipmi-sensors.8.html`.

[34] E. Krout, "Using top to monitor server performance." [Online]. Available: `https://www.linode.com/docs/uptime/monitoring/top-htop-iotop/`, Updated: June, 2018.

[35] J. LaCroix, *Mastering Ubuntu Server: Master the art of deploying, configuring, managing, and troubleshooting Ubuntu Server 18.04*. Packt Publishing Ltd, 2018.

[36] A. Badea, S. Halunga, and G. Luca, "Energy optimization for the low data rate iot devices by using Manchester's coded pseudo-random sequences," in *Proceedings of the 6th Conference on the Engineering of Computer Based Systems*, ECBS '19, (New York, NY, USA), pp. 19:1–19:4, ACM, 2019.

[37] A. C. Zanin, M. T. De Gouvea, and D. Odloak, "Integrating real-time optimization into the model predictive controller of the FCC system," *Control Engineering Practice*, vol. 10, no. 8, pp. 819–831, 2002.

[38] G. De Souza, D. Odloak, and A. C. Zanin, "Real time optimization (RTO) with model predictive control (MPC)," *Computers & Chemical Engineering*, vol. 34, no. 12, pp. 1999–2006, 2010.

[39] C. Edwards and S. Spurgeon, *Sliding mode control: theory and applications*. Crc Press, 1998.

Chapter 7

Conclusion

This work follows a logical path in the development of a complete system for joint workload assignment and cooling control for data centers. The following steps are performed:

- Quantify thermal differences between servers.
- Exploit the difference between different locations in a data center from the perspective of cooling units.
- Construct a thermal model for data centers that does not require knowledge of physical laws and heat-transfer equations, adapts to thermal changes, and is computationally inexpensive.
- Construct a framework that jointly considers servers and cooling thermal heterogeneity for workload assignment and cooling control.
- Implement a holistic control system for workload assignment and cooling control while considering the transient behavior of the system under control.

Exploiting data center thermal heterogeneity using thermal models is feasible, and it can potentially solve this problem. Being more specific about the stages above, the first stepping stone of this work considers the thermal differences between servers (server thermal heterogeneity). We find that assigning workload can be optimized based on these differences to reduce the cooling cost. During the next step, the consideration of the thermal heterogeneity from the perspective of the cooling units is investigated. A considerable amount of power savings is obtained by optimizing the assignment of workload and adjusting the cooling system parameters. These results are obtained without considering server heterogeneity.

Up to this point, decisions for the cooling unit are based on a physical zonal-based thermal model, which is not desirable due to a number of implementation issues for larger scale data centers. This issue triggers our next step, which is building a data-driven, adaptive, accurate temperature prediction method. We use a neural network time-series prediction method for this thermal model. In the paper *Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning*, thermal models of both a data center and servers are used in an optimization process for the infrastructure control of the data center. In this part of our work, the thermal model generated in the previous step is used in the core of the optimization process. We see remarkable results considering all aspects of thermal heterogeneity in data centers during workload assignment and cooling control.

The last piece of this work is the implementation of a system for the infrastructure control of the data center. This work, unlike other parts, considers the transient behavior of the system under-study. In this paper, the actual temperature requirements of servers (considering server heterogeneity) are provided to the cooling controller via

a required temperature distribution map. The controller uses an enhanced version of model predictive control (MPC) to satisfy the given temperature requirements of servers. The system implementation proves the effectiveness of this method through considerable power savings.

There are several aspects which could be done differently or improved in future work. These aspects are as follows:

- Considering the performance and thermal effects of the combination of a server and an application can be counted as another aspect of server heterogeneity. However, in this work, we focus on the thermal effects of affected thermal changes by CPU utilization.

- The sensitivity of the results to the size of the thermal zones could be explored, as the coarser the granularity, the simpler the control implementation, but there is an inherent trade off with accuracy/performance.

- Due to the availability of accurate cooling power models in the literature, we use a specific power model for the cooling units in our study, based on a physical model. While the physical properties of cooling units tend to be well understood, a data-driven model is another possible alternative.

- The controller system constructed in the last part of our work, uses a neural network within the MPC controller. Due to probabilistic and nondeterministic properties of neural networks, controllers may require a preventive layer of action for the purposes of fault tolerance.

- The process of acquiring data for model training and updates might be different from one system to another.

- The model predictive control method, used in the last paper, could possibly be replaced by other control methods, such as reinforcement learning methods. Having said that, it does appear that MPC is very well suited for this application.

- In the paper, *Joint data center cooling and workload management: A thermal-aware approach*, a significant caveat for the power efficiency of server consolidation methods is shown when considering associated thermal effects. Exploiting this caveat is a suggestion for future work.

Bibliography

[1] A. Shehabi, S. J. Smith, E. Masanet, and J. Koomey, "Data center growth in the United States: Decoupling the demand for services from electricity use," *Environmental Research Letters*, vol. 13, no. 12, p. 124030, 2018.

[2] S. Umair, U. Muneer, M. N. Zahoor, and A. W. Malik, "Mobile cloud computing future trends and opportunities," in *Managing and Processing Big Data in Cloud Computing*, pp. 105–120, IGI Global, 2016.

[3] H. Klemick, E. Kopits, and A. Wolverton, "How do data centers make energy-efficiency investment decisions? Qualitative evidence from focus groups and interviews," *Energy Efficiency*, vol. 12, pp. 1359–1377, June 2019.

[4] E. Yu, S. Cho, H. Shin, and B.-G. Park, "A band-engineered one-transistor DRAM with improved data retention and power efficiency," *IEEE Electron Device Letters*, vol. 40, no. 4, pp. 562–565, 2019.

[5] I. Tarasov, "Architectures of high-performance VLSI for custom computing systems," in *Journal of Physics: Conference Series*, vol. 1333, p. 022019, IOP Publishing, 2019.

[6] B. Acun, K. Chandrasekar, and L. V. Kale, "Fine-grained energy efficiency using per-core DVFS with an adaptive runtime system," in *2019 Tenth International Green and Sustainable Computing Conference (IGSC)*, pp. 1–8, IEEE, 2019.

[7] V. J. Maccio and D. G. Down, "Structural properties and exact analysis of energy-aware multiserver queueing systems with setup times," *Performance Evaluation*, vol. 121, pp. 48–66, 2018.

[8] B. Hartmann and C. Farkas, "Energy efficient data centre infrastructure—Development of a power loss model," *Energy and Buildings*, vol. 127, pp. 692–699, 2016.

[9] J. Dai, M. M. Ohadi, D. Das, and M. G. Pecht, *Optimum Cooling of Data Centers*. Springer, 2016.

[10] 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.

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

[12] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, vol. 53, no. 17, pp. 2888–2904, 2009.

[13] Q. Fang, Q. Gong, J. Wang, and Y. Wang, "Optimization based resource and cooling management for a high performance computing data center," *ISA Transactions*, vol. 90, pp. 202–212, 2019.

[14] X. Zhao, Z. Xiong, L. Ding, X. Zhang, and F. Xu, "A smart coordinated temperature feedback controller for energy-efficient data centers," *Future Generation Computer Systems*, vol. 93, pp. 506–514, 2019.

[15] Q. Wang, M. Song, Q. Fang, and J. Wang, "Thermal-aware flow field optimization for energy saving of data centers," in *2018 Annual American Control Conference (ACC)*, pp. 3744–3749, IEEE, 2018.

[16] S. MirhoseiniNejad, G. Badawy, and D. G. Down, "EAWA: Energy-aware workload assignment in data centers," in *2018 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 260–267, IEEE, 2018.

[17] S. MirhoseiniNejad, H. Moazamigoodarzi, G. Badawy, and D. G. Down, "Joint data center cooling and workload management: A thermal-aware approach," *Future Generation Computer Systems*, vol. 104, pp. 174 – 186, 2020.

[18] S. MirhoseiniNejad, F. M. García, G. Badawy, and D. G. Down, "ALTM: Adaptive learning-based thermal model for temperature predictions in data centers," in *2019 IEEE Sustainability through ICT Summit (StICT)*, pp. 1–6, IEEE, 2019.

[19] S. MirhoseiniNejad, , G. Badawy, and D. G. Down, "Holistic thermal-aware workload management and infrastructure control for heterogeneous data centers using machine learning," *Future Generation Computer Systems*, 2020. Submitted.

[20] S. MirhoseiniNejad, , G. Badawy, and D. G. Down, "It-aware cooling control framework for data centers: A machine learning control approach," *IEEE Systems Journal*, 2020. Submitted.