

INTEGRATED CONTROL OF MULTIPLE
COOLING UNITS

INTEGRATED CONTROL OF MULTIPLE
COOLING UNITS

By SHIRIN MOZAFFARI, B. Sc.

A Thesis Submitted to the School of Graduate Studies in Partial Fulfilment of
the Requirements for the Master of Science

McMaster University Master of Science (2019) Hamilton, Ontario (Computing
and Software)

TITLE: Integrated Control of Multiple Cooling Units

AUTHOR: Shirin Mozaffari, M.Sc. (McMaster University)

SUPERVISOR: Professor Douglas Down

NUMBER OF PAGES: ix, 78

To my sister Nahid,
without whom I could not continue my studies.

Contents

1	Introduction	6
1.1	Data Centre Cooling	8
2	Related Work	11
3	Initial Experiments	15
3.1	Architecture	15
3.2	Results and Findings	19
3.3	Workload Assignment	26
4	Cooling Power Consumption Optimization	30
4.1	Power Model	30
4.2	Comparing Power Consumption	33
4.3	Optimization Problem	36
4.4	Solving the Optimization Problem	38
5	Proof of Optimality	40
5.1	Convex Power Function	42
5.2	Concave Power Function	52
5.3	Neither Convex nor Concave Power Function	57
6	$T_{out,j}$ Model Estimation	65
6.1	The Necessity of New Experiments	65
6.2	Architecture	67
6.3	Results and Discussion	68
6.4	Proof of Integrity	71

7 Conclusion	73
7.1 Future Work	74

1 Introduction

Data centres are an integral part of today's technology. With the growing demand for data centers to meet computational needs, there is pressure to decrease data center-related costs. Cloud services account for 5% of electricity consumption around the world [1, 2, 3]. Cooling units are one of the key components that consume a considerable amount of power in data centers. From [4], in a data centre as large as 5000 square feet about 40% of power consumption is accounted for by cooling (see Figure 1). By reducing the amount of power needed to cool servers, the overall power consumption can be decreased. Efficient cooling of data centres involves meeting temperature constraints while minimizing power consumption. By exploring the opportunities that may be available through controlling multiple cooling units, we can avoid issues such as overcooling (some parts of the data center being cooled more than necessary) or warm air recirculation (return of exhausted hot air to inlets of servers). Currently, in data centres with more than one cooling unit, each of the cooling units is controlled independently. In other words, cooling units do not collaborate with each other. This mode of operation results in each cooling unit needing to be set for the worst case, which results in over cooling and is not energy efficient. Coordinating cooling units has the potential to decrease the power consumption of a data centre by eliminating this over cooling. Furthermore, coordinating with workload management may help mitigate cooling unit power consumption.

This research is concerned with exploring what is feasible within the options discussed above. It contains two main parts. The first part focuses on a cooling control algorithm that integrates the operation of multiple cooling units to minimize the total power consumption in data centres. Through this part we gain insight into the concept of zones in our experimental data centre and try

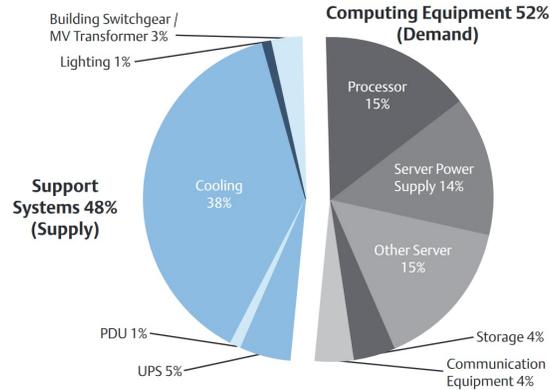


Figure 1: Analysis of the power consumption of a 5000 square foot data centre [4]

to design a control algorithm via solving an appropriate optimization problem. The results of the first part are for one particular model. In the second part, we show that our proposed solution is optimal in greater generality (not just for the particular model that we study). We prove that our proposed control algorithm works for any data centre, given that the power consumption of the cooling units satisfies appropriate properties. In particular if it is convex (as appears to be the case in most practical situations), we show that the cooling units should be operated in a homogeneous manner, i.e., there is no benefit to unbalancing workload so that some cooling units are more heavily loaded, while others are more lightly loaded. In our setting, it is reasonable to assume that the power function is continuous and increasing. We also use machine learning techniques to find a data-driven model for one component of the model, the exhaust air temperature of a server. In order to become more familiar with cooling technologies and to introduce some common technical terms in the data centre domain, we explain some of the concepts that are used throughout this thesis in the next section.

1.1 Data Centre Cooling

The traditional manner of cooling data centers uses raised floor technology, in which the floor is elevated by about 40 cm to create a void between the actual floor and the elevated floor [5]. In the space created between these two, the cold air is pressurized and delivered to servers using perforated tiles. After the cold air passes through the servers, exhaust air leaves from the back of servers and returns to the cooling unit to be cooled down. CRAC (Computer Room Air Conditioner) units are used to pressurize cold air underneath the raised floor.

To better manage airflow and save more energy (subsequently lower cost), a hot aisle/cold aisle rack configuration is used. It is a common practice in which servers are lined up in alternate rows. The front side of servers - which takes the cold air and hence is known as the cold aisle - is faced one way while the rear side - which emits exhaust hot air and hence is referred to as the hot aisle - is faced the other way. Figure 2 shows this configuration as well as the raised floor technology for cooling data centres.

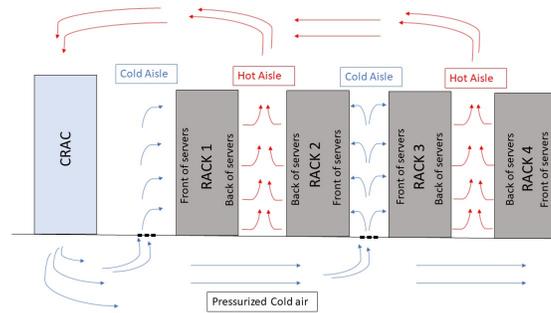


Figure 2: Raised floor technology - hot aisle-cold aisle

One of the downsides of using raised floor is that hot air can to some degree mix with cold air in front of servers, increasing the server inlet temperatures

and lowering the efficiency of the cooling system. This problem is very common in data centres and is known as *hot air recirculation*. On the other hand, in this method it is easy to deliver cold air to servers because all that is required is placing a perforated tile (instead of a solid tile) in front of the rack. Raised floor has been in use for many years. It is in fact what most people have in mind when they think of data center cooling.

The technology that we are using to cool our data centre is called in-row cooling (IRC). In-row cooling does not require any elevated floor or swapping of tiles. In this technology, cooling units are embedded in data centre cabinets and are in close proximity to racks of servers to ease the targeted cooling. This results in improved efficiency and reduced hot air recirculation. Figure 3 shows the in-row cooling technology.

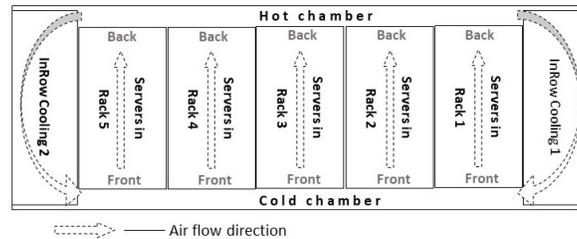


Figure 3: Top view of an in-row cooled data centre

As can be seen in Figure 3, there is one in-row cooling unit at each end and racks are placed between the two cooling units. Also, a hot chamber and cold chamber have replaced the hot aisle and cold aisle because here, there is no space between racks. Similar to the concept of a hot aisle and a cold aisle, the hot chamber refers to the side towards which backs of servers are faced and the cold chamber - which receives the cold air produced by cooling units - is the side towards which the fronts of servers are faced.

It is worth noting that cooling units return cold air with a temperature equal to the setpoint of the cooling unit. According to ASHRAE ¹ guidelines for data centres, minimum and maximum server inlet temperatures are 16°C and 28°C, respectively. This is a safe range in which no performance degradation happens. Temperatures lower than 16°C can result in condensation problems, leading to server corruption and failure. We assume that the lower temperature constraint is always met, so only the upper bound is of concern.

In the next chapter, we will discuss some previous research works that are related to our research. Throughout this chapter, we also discuss our first thoughts and ideas. Our experimental setup is like the one in Figure 3. Also, from now on, wherever the term “cooling unit” is used, it is shorthand for “in-row cooling unit”.

¹American Society of Heating, Refrigerating and Air-Conditioning Engineers

2 Related Work

Efficient cooling has a huge positive impact on data centre performance, both economically and operationally. Temperature management is a key factor in data centre environmental control. In order to achieve high performance computing goals and meet service level agreements (SLA), server circuitry should work below a temperature threshold to prevent failure. High temperatures as a result of inefficient cooling negatively affect IT equipment by reducing reliability [6]. There are some techniques currently used to reduce power consumption in different data centres. For example, to control energy consumption of servers, processors use dynamic voltage and frequency scaling (DVFS) which lets them change voltage or frequency dynamically during run time [7, 8]. This is a server-level optimization which is not in the scope of this thesis. *Server consolidation* is also another technique which tries to assign workload to a minimum number of servers and shuts down the rest [9, 10, 11]. On the other hand, *load balancing* is the technique which distributes workload uniformly among all the servers in a data centre to achieve evenly distributed power in order to reduce hot spots and decrease latency [12]. A number of previous works have examined the problem of controlling multiple cooling units in a data centre, the focus of this work. Most of these works - and our work - focus on different zones within a data centre. A zone or a region of influence of a cooling unit is an area within which the temperature of different points is relatively the same, within some tolerance. While operating, locations closer to a cooling unit have lower temperatures than distant locations, creating a region or a zone. The shape and location of a zone are not static and vary between different architectures and cooling unit placements within a data centre [13]. Physical barriers such as wires and walls also affect the shape of zones. It is worth noting that the concept of zones in

data centres and using zonal models for the aim of controlling or monitoring temperature has always been a matter of interest. This comes from the fact that using a zonal model can decompose the larger problem into smaller and potentially more tractable problems. Some research works have solely focused on methodologies to define zones in a data centre. For example Hamann, Lopez and Stepanchuk [14] present a methodology to define zones in a data centre. They have used thermal equations to find zones in a raised floor data centre. However, in most of the works the manner of defining zones or the parameters that affect the zones are not discussed. Song et al. [15] is one example of such work. They focus on a zonal model to describe airflow and temperature patterns in a data center. In their work, they regard a zonal model as a thermodynamics based intermediate approach and do not talk about the process of finding zones and defining their boundaries. Bash, Patel and Sharma in [16] outline a control scheme for dynamic thermal management of air cooled data centers. In this research, each CRAC's supply air temperature is perturbed and the subsequent change at each rack's inlet sensor is recorded. Regions of influence or zones are defined by this process and the shapes of zones are shown in Figure 4. Their data centre also uses a raised floor structure to cool the servers.

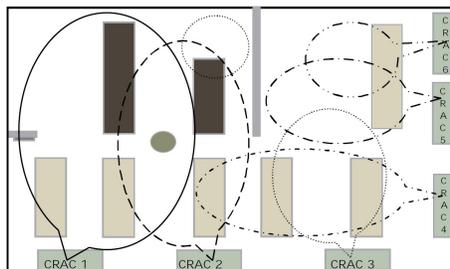


Figure 4: CRACs regions of influence presented in [16]

Cool job assignment is presented by Bash et al. in [17]. This technique uses an index to determine which locations in a data center are cooled more efficiently

by altering the supply temperature of each cooling unit and monitoring the effect on the inlet temperature of each of the servers. Sano et al. in [18] developed an “IT facility linkage system” to save power by prioritizing servers in terms of air conditioning sensitivity. This linkage system’s main goal is to reduce the power consumption in a data centre by connecting two systems; one is the server load allocation system and the other one is the air conditioning control system. The similarity between this work and our work lies in the granularity. In [18], they focus on individual servers but in the present work, we first check the validity of a zonal model in an experimental, modular data centre.

In most of the previous research in this area, cooling units and servers are physically separate, for which it seems plausible to have different zones. They make use of separate cooling units located far from servers and raised floor cooling. In contrast, our data centre is a modular system in which servers and cooling units are in an enclosure and thus are not separate from each other. We show that a zonal model is still applicable and zones can be clearly defined. In addition we explore which parameters affect the zones and to what degree. The rest of this research is structured as follows. In Chapter 3 we describe the architecture of the data centre we are using and the set of experiments that we performed. Also we discuss the initial results and the origins of the zonal model. Once we have the zonal model, it allows us to formulate a fairly straightforward optimization problem. If we did not have the zonal model, we could still formulate an optimization problem but it would both require a more complex modeling process and be potentially more difficult to solve. Chapter 4 includes the optimization problem and its solution. The structure of the optimal solution when the power consumption of the cooling units has varying forms can be found in Chapter 5. Chapter 6 includes the process of modeling the outlet temperature of servers and finally, Chapter 7 concludes the thesis, including

suggestions for future work.

3 Initial Experiments

3.1 Architecture

Experiments conducted in this study are in an enclosed system of 60 servers set up in five racks with two cooling units at either end of the racks. The number of servers was chosen to give maximum power consumption of 20kW which provided a safety range in case there was any over heating as the installed cooling capacity can handle up to 30 kW. Each rack contained 12 servers separated into three groups of four servers. Each group of servers had three temperature sensors: one at the front, one at the side of the front and one at the back. The front and back temperature sensors were aligned. In order to prevent hot air recirculation and the migration of cold air to the back of the racks, blocking panels were installed. To avoid any leakage, the racks were sealed with air tight doors and side panels. Figure 5 shows the arrangement of the temperature sensors at the front and back of a rack, respectively.

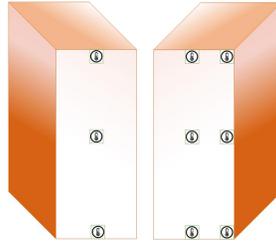


Figure 5: Location of temperature sensors in a rack. Left: sensors at the back of rack. Right: sensors at the front of rack

The cooling units' operational parameters were controlled and set to the desired value for each experiment. Temperature sensors were required to report constant readings before any experiment was started. Moreover, the experiments were left to reach steady state before temperature measurements were recorded.

Initially, one of the cooling units had three fans while the other one had only two. To maintain the symmetry, we used two fans in both cooling units. Figure 6 shows the location of fans in cooling units and the front view of our architecture.



Figure 6: Location of fans in cooling units

When data centers are overcooled, a typical cause is that the setpoints of the cooling units are set to equal values at the lowest possible magnitude. To investigate the effect of individual cooling units on the inlet temperature distribution within the racks, we conducted experiments where we incrementally changed the difference between setpoints of cooling units. One of the setpoints was fixed at 16°C , the minimum temperature based on the ASHRAE guidelines for the inlet temperature of servers, while the setpoint of the other cooling unit was increased to 26°C , which we set to be our maximum. The configuration for a sample experiment is shown in Figures 7 and 8. Figure 7 shows the temperature reported by sensors at the front and side of the racks. Figure 8 shows the temperature reported by sensors at the back. Each cell contains the temperature reported by a sensor. In Figure 7, nine cells below each rack indicate the temperature for the front and sides of the rack (side sensors are shared between racks except on the right side for rack 1 and the left side for rack 5). Among these nine cells, the three cells in the middle column are the temperatures reported by sensors at the middle front of the rack. The six remaining cells show the temperature reported by the shared sensors. As an example, Figure 9 shows the location of the sensors for rack 1 and the corresponding cells in Figure 7.

Similarly, in Figure 8, three cells below each rack indicate the temperature at the back. Here, we only need three cells for each rack because there are three sensors at the back of each rack. For the rest of this chapter, we only use the figures which depict the temperature reported by the front sensors (similar to Figure 7).

C.U.2 - 16 °C	Rack 5			Rack 4			Rack 3			Rack 2			Rack 1			C.U. 1 - 16 °C
	18.9	20.1	28.1	19.8	36.9	24.1	36.5	20.6	18.7	17.5	16.7					
	20	19.8	21.9	19.3	30.5	23.8	29.8	19.7	17.1	17.3	16.2					
	18	17.6	23.5	24.2	30.6	21.9	20.3	16.4	16	16	16.4					

Figure 7: Temperature at front and side of racks for an experiment

C.U.2 - 16 °C	Rack 5			Rack 4			Rack 3			Rack 2			Rack 1			C.U. 1 - 16 °C
		36.5		36		48.6		42.9		31.9						
		38.9		37.5		37.6		30.7		32.4						
		37		38		33.5		36.9		37.3						

Figure 8: Temperature at the back of racks for an experiment

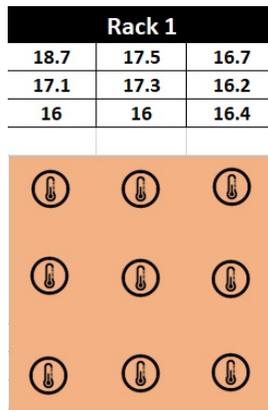


Figure 9: The location of probes in a rack and their corresponding cells in the table. Notice while each rack has six sensors at the front, it is also sharing three sensors with the racks beside it. Therefore there are nine cells per rack in Figures 7 and 8

In this particular experiment, the setpoint for both cooling units is set to 16°C. Our goal in this experiment is to see the inlet temperature of servers when cooling units have the lowest setpoint. Cooling units fan speed and workload are set at 40% and 50%, respectively. The experiment is run for one hour to reach steady state. Figures 7 and 8 show the results of one experiment after running with the mentioned configuration for one hour.

In the next section, we will conduct a set of experiments to see whether or not a zonal model is applicable for our data centre. This is important because as mentioned earlier in Chapter 2, the question of the existence of a zonal model for data centres that use in-row cooling units has not been studied before. In addition to this, we will study how each of the parameters - cooling unit setpoints, cooling unit fan speeds and workload will affect the zones, if at all. Table 1 shows the notation used in this thesis.

Parameter	Definition
N	Total number of servers
n_i	Number of servers in zone i
U	System workload (utilization)
u_j	Utilization of server j
$T_{in,j}$	Inlet temperature of server j
$T_{out,j}$	Outlet temperature of server j
$T_{setpoint,i}$	Setpoint of cooling unit i
$T_{in,i}$	Inlet temperature of cooling unit i
$T_{out,i}$	Outlet temperature of cooling unit i
F_i	Fan speed of cooling unit i
d_j	Distance of server j from cooling unit
C.U.	Cooling Unit
I.R.C.	In-Row Cooling
Π	Total power consumption of data centre
Π_{hvac}	Power consumption of HVAC
Π_{comp}	Power consumption of servers
Π_{other}	Power consumption of other parts of data centre
$Z(j)$	The zone number for server j

Table 1: Notation used in the thesis

3.2 Results and Findings

In this section, we alter cooling unit setpoints, workload and fan speed and monitor the change in the resulting temperature distribution. We observed the changes in the shape of the zones as we vary the different parameters (U , $T_{setpoint,i}$, F_i , etc.) and determined that the only thing that changes the shape of the zones is the relative value of the setpoints.

From the experimental results, we observed that a setpoint higher than 26°C would result in server inlet temperatures higher than 28°C, which is undesirable. When $T_{in,j}$ is higher than 28°C, the CPU temperature of the servers can become unacceptably high which in turn could potentially cause the internal components of the servers to fail early or the servers could throttle back their speeds (the latter mode of protecting servers is built into most operating systems).

We chose different setpoints for cooling units to be able to observe the potential zones. The temperature for the cooling unit with a higher setpoint is referred as $T_{setpoint,high}$. Similarly, $T_{setpoint,low}$ corresponds to the cooling unit with lower setpoint. As result of the setpoint difference, regions of influence were observed for each cooling unit. Zones were defined using the value

$$\alpha = \frac{T_{setpoint,high} + T_{setpoint,low}}{2}$$

When a server inlet temperature ($T_{in,j}$) is higher than α it is considered to be associated with the cooling unit with the higher setpoint; the hot zone. On the other hand if the server inlet temperature is lower than α it is considered to be associated with the cooling unit with the lower setpoint; the cold zone.

To get a better understanding of the difference in temperature, we decided to use two colors; blue and red indicating areas with lower and higher temperature, respectively. When the temperature of a cell (sensor) is higher than α , that

cell will be colored in red. Similarly, a temperature lower than α is shown with color blue. Each of the tables in Figure 10 shows the results of one experiment. Cooling unit setpoints can be seen at each end. Note that these are representative examples - we ran 60 additional experiments.

C.U 2 20 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 16 °C
	20.38	20.4	20.31	20.38	20.5	20.1	21.3	20.5	17.69	17.38	16.75
	20.56	20.31	19.69	19.81	20.19	20	19.31	18.31	17.31	16.88	17.19
	19.25	20.38	22.75	20.25	20	19.31	17.31	16.63	16.5	16.44	16.63
C.U 2 24 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 16 °C
	25.13	25.09	24.81	24.81	25	23.9	25.19	24.06	18.38	17.75	16.81
	24.94	24.56	23.5	23.31	24	23.5	21.31	19.69	17.63	17.06	17.25
	22.63	23.5	26	22.69	21.25	20.31	17.5	16.63	16.56	16.44	16.63
C.U 2 28 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 16 °C
	29.13	29.1	28.81	28.75	29.13	28.3	28.56	25.94	19.13	18.06	16.88
	28.81	28.25	27.13	26.63	27.88	27.19	23.44	21.25	17.94	17.19	17.88
	26.31	26.56	29.88	24.5	24.31	21.56	17.63	16.75	16.63	16.5	16.69

Figure 10: Initial experiments towards a zonal theory

Figure 10 shows the effect of increasing the gap between two setpoints on zones. In this figure, the gap between setpoints is initially 4°C and it increases to 8°C and 12°C in the second and third experiments. As can be seen, the pattern of the zones remained essentially unchanged.

Defining the zones based on the α definition works well when the setpoints are far apart but when the gap between the two setpoints decreases, we face issues in defining the shape of the zones. In all of the experiments in Figure 10, there is a gap of at least 4°C. We decided to decrease the gap and see the effect on the zonal structure. Figure 11 shows the results when the gap is decreased to 2°C. In this case zonal model is still valid but it is hard to verify.

IRC2 18 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		IRC1 16 °C
	18.4	18.3	18.56	18.5	18.5	18.3	18.25	17.63	17.44	17.13	16.69
	18.8	18.5	18.06	18.31	18.44	18.25	17.81	17.19	17.31	17.38	17.19
	17.9	18.69	19.56	19.31	18.44	17.81	17.63	17.19	17	16.88	17.06

Figure 11: Different possible combinations for defining zones when the gap between the two setpoints is 2°C.

As can be seen in Figure 11, when cooling unit setpoints are close to each other, it is difficult to distinguish which cooling unit has a more dominant effect on the inlet temperature and defining the zones using α is no longer meaningful. To address this issue, we use an alternative way of defining zones. Based on the results of experiments, we noticed that there is a fixed gap between the inlet temperature of servers and setpoint. We denote this gap by δ and fixed it at 2°C. In other words if server j is in the zone of cooling unit i then

$$T_{setpoint,i} \leq T_{in,j} \leq T_{setpoint,i} + \delta \quad \delta = 2^{\circ}C \quad T_{setpoint,i} \leq T_{in,j} \leq T_{setpoint,i} + 2^{\circ}C \quad (1)$$

The zonal definition in (1) relates the inlet temperature of server j and the setpoint of cooling unit i . It is consistent with the definition of α as well as the shape of the zones in Figure 10. Figures 12, 13 and 14 below compare the zones defined using each of the two approaches. The top image uses the α definition and the bottom image uses the δ definition.

IRC 2 20 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		IRC 1 16 °C
	20.4	20.4	20.3	20.38	20.5	20.8	21.3	20.5	17.7	17.38	16.8
	20.6	20.31	19.7	19.81	20.2	20	19.31	18.31	17.3	16.88	17.2
	19.3	20.38	22.8	20.25	20	19.31	17.31	16.63	16.5	16.44	16.6

IRC 2 20 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		IRC 1 16 °C
	20.4	20.4	20.3	20.38	20.5	20.8	21.3	20.5	17.7	17.38	16.8
	20.6	20.31	19.7	19.81	20.2	20	19.31	18.31	17.3	16.88	17.2
	19.3	20.38	22.8	20.25	20	19.31	17.31	16.63	16.5	16.44	16.6

Figure 12: Comparing the zones defined using α and δ . The gap between the two setpoints is 4°C.

IRC 2 24 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		IRC 1 16 °C
	25.13	25.1	24.81	24.81	25	23.9	25.19	24.06	18.38	17.75	16.81
	24.94	24.56	23.5	23.31	24	23.5	21.31	19.69	17.63	17.06	17.25
	22.63	23.5	26	22.69	21.25	20.31	17.5	16.63	16.56	16.44	16.63

IRC 2 24 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		IRC 1 16 °C
	25.13	25.1	24.81	24.81	25	23.9	25.19	24.06	18.38	17.75	16.81
	24.94	24.56	23.5	23.31	24	23.5	21.31	19.69	17.63	17.06	17.25
	22.63	23.5	26	22.69	21.25	20.31	17.5	16.63	16.56	16.44	16.63

Figure 13: Comparing the zones defined using α and δ . The gap between the two setpoints is 8°C.

IRC 2 28 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		IRC 1 16 °C
	29.13	28.93	28.81	28.75	29.13	28.3	28.56	25.94	19.13	18.06	16.88
	28.81	28.25	27.13	26.63	27.88	27.19	23.44	21.25	17.94	17.19	17.88
	28.31	26.56	29.88	24.5	24.31	21.56	17.63	16.75	16.63	16.5	16.69

IRC 2 28 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		IRC 1 16 °C
	29.13	28.93	28.81	28.75	29.13	28.3	28.56	25.94	19.13	18.06	16.88
	28.81	28.25	27.13	26.63	27.88	27.19	23.44	21.25	17.94	17.19	17.88
	28.31	26.56	29.88	24.5	24.31	21.56	17.63	16.75	16.63	16.5	16.69

Figure 14: Comparing the zones defined using α and δ . The gap between the two setpoints is 12°C.

As seen in Figures 12, 13 and 14, when the gap between the two setpoints is not small, the two approaches for defining zones have similar results. Also, using the δ definition we can see the area of overlap between the two zones. This area

is shown with white cells in Figures 13 and 14.

Further experiments were designed to consolidate this zonal hypothesis. For these experiments, we will use the α definition to define zones. The setpoints of the cooling units were switched to determine whether the pattern of the zones would stay the same. Figure 15 shows the results of three experiments which were conducted for this purpose.

C.U 2 - 16 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 - 20 °C
17.5	NR	17.56	17.63	17.94	18.2	18.88	19	20.5	20.56	20.44	
17.88	17.81	17.5	18	18.69	18.31	19.5	19.81	20.56	20.38	20.69	
17.06	20.06	23.06	21.25	21.5	19.75	20.44	20.06	19.94	19.81	19.56	
C.U 2 - 16 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 - 24 °C
17.25	NR	17.25	17.5	18.06	18.8	19.06	20.19	24.38	24.5	24.56	
17.69	17.69	17.5	18.25	19.38	18.38	21.56	22.69	24.5	24.44	24.63	
16.94	22.38	26.75	22.94	23.81	22.5	24.31	24	23.81	23.44	22.81	
C.U 2 - 16°C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 - 28 °C
17.5	17.3	17.44	18.44	19	19.63	20.75	20.75	27.81	28.44	28.63	
17.94	18	17.88	18.63	18.63	23.19	25.19	25.19	28.44	28.44	29.88	
17.13	23.38	29	24.63	24.81	28	28	27.81	27.63	27.38	26.5	

Figure 15: The effect of switching cooling unit setpoints

The pattern achieved was different but remained the same as the temperature gap increased, as shown in Figure 15. So, while the shape of the zones changed, a zonal model remains valid.

In the next step, we wanted to explore the effect, if any, of the utilization of the servers (workload) on the zones. For this purpose, we only modified servers' utilization while keeping setpoints and fan speed fixed. All the servers had the same utilization within an experiment. Three utilizations within the range of (0-100%) were chosen: 20% as a low utilization, 50% as a medium utilization and 80% as a high utilization. All three were also tested when the setpoints are switched. Figure 16 shows the zones for different server utilizations when setpoints are set to 16°C and 24°C, Figure 17 shows the results when the setpoints are switched.

Utilization 80%											
C.U 2 - 24 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 - 16 °C
25.13	NR	24.81	24.81	25	NR	25.19	24.06	18.38	17.75	16.81	
24.94	24.56	23.5	23.31	24	23.5	21.31	19.69	17.63	17.06	17.25	
22.63	23.5	26	22.69	21.25	20.31	17.5	16.63	16.56	16.44	16.63	
Utilization 50%											
C.U 2 - 24 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 - 16 °C
25.31	24.63	25.13	25.06	25	24.5	24.63	22.63	17.75	17.63	16.81	
25.19	25.06	23.88	23.44	23.19	23.44	21.44	19.5	17.19	17.06	17.06	
23	22.44	25.38	21.94	19.56	18.81	17.25	16.69	16.63	16.44	16.81	
Utilization 20%											
C.U 2 - 24 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 - 16 °C
26.06	24.38	24.88	24.81	24.63	24.31	24.19	21.88	16.94	16.75	16.06	
24.88	24.69	23.69	23.13	22.88	22.94	20.88	18.81	16.38	16.31	16.31	
22.8	21.88	24.38	21.19	19	18	16.5	16	15.88	15.75	16.06	

Figure 16: The effect of changing utilization, cooling unit 1 has the lower setpoint

Utilization 80%											
C.U 2 16 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 24 °C
16.69	16.63	16.69	16.81	17.06	17.44	17.63	20.25	23.56	23.75	24.06	
17.06	17.13	16.75	17.44	17.81	17.81	20.44	22.06	23.75	23.88	24	
16.69	22.31	26.56	21.31	22.88	22.75	23.75	23.63	23.5	23.31	22.81	
Utilization 50%											
C.U 2 16 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 24 °C
16.69	16.63	16.69	16.81	17.06	17.44	17.63	20.25	23.56	23.75	24.06	
17.06	17.13	16.75	17.44	17.81	17.81	20.44	22.06	23.75	23.88	24	
16.69	22.31	26.56	21.31	22.88	22.75	23.75	23.63	23.5	23.31	22.81	
Utilization 20%											
C.U 2 16 °C	Rack 5		Rack 4		Rack 3		Rack 2		Rack 1		C.U 1 24 °C
16.25	16.38	16.31	16.44	16.63	17.13	17.31	20.19	23.69	23.88	24.19	
16.69	16.69	16.44	17.19	17.56	17.75	20.5	22.06	23.94	24.06	24.25	
16.25	22.44	25.69	21.31	22.88	22.69	23.94	23.88	23.81	23.69	23.19	

Figure 17: The effect of changing utilization - cooling unit 2 has the lower setpoint

As shown by Figures 16 and 17 above, changing the utilization had no effect on either the shape of the zones or their boundaries. Subsequently, it was decided to keep the servers in the cold zone working at a utilization of 100%, while servers in the hot zone remain idle. This would be equivalent to having all

servers working at 40%. We performed the next experiments without switching the cooling unit setpoints. Figure 18 shows the result of an experiment when all the servers are working at 40% utilization. This is uniform workload distribution in which all the servers have the same utilization.

IRC2 24 °C	Rack 5			Rack 4		Rack 3		Rack 2		Rack 1		IRC1 16 °C
	24.88	24.13	24.63	24.56	24.56	23.88	23.81	21.56	17	16.94	16.25	
	24.63	24.56	23.56	22.81	22.63	22.44	20.75	18.81	16.63	16.5	16.5	
	22.69	22.44	21.25	21.63	19.31	18.13	16.69	16.13	16	15.88	16.13	

Figure 18: The effect of changing utilization - all the servers have a utilization of 40%

Figure 19 shows the result when servers located in the hot zone are idle while servers located in the cold zone are working at 100% utilization. In this case, there is a non-uniform workload distribution as the utilizations of the zones are different and the cold zone handles most of the workload.

IRC2 24 °C	Rack 5			Rack 4		Rack 3		Rack 2		Rack 1		IRC1 16 °C
	25.06	24.31	24.81	24.75	24.69	24.13	24	21.75	17.19	17.13	16.56	
	24.75	24.69	23.75	23.13	22.94	22.81	21	19.06	16.88	16.75	16.81	
	23.13	22.88	21.63	22	19.69	18.44	17	16.5	16.38	16.31	16.5	

Figure 19: The effect of changing utilization - All servers of Rack 1, Bottom 8 Servers of Rack 2, Bottom 4 Servers of Rack 3 are 100% . All other servers are idle

Once again, the zones remain unchanged. Consequently, the effect of one last parameter was to be tested which was the fan speed. The fan speed of the cooling unit with the higher setpoint was decreased to see if the colder air from the cooling unit with the lower setpoint could have more influence.

IRC2 24 °C	Rack 5			Rack 4		Rack 3		Rack 2		Rack 1		IRC1 16 °C
60% Fan Speed	25.06	24.5	25	25.19	25.81	25	26.31	24	17.18	17.75	16.69	80% Fan Speed
	25.06	25.06	23.88	23.13	23.13	22.06	22.31	19.69	17.19	17.13	17.13	
	23.13	22.88	21.63	20.81	19.44	17.81	17.19	16.63	16.5	16.38	16.63	

Figure 20: The effect of changing fan speed - Servers in cold zone are working at 100% and servers in hot zone are idle

As can be seen in Figure 20, the zones remained the same. From this point, the fan speed for both cooling units is set to 100%, unless specified otherwise. This is because we found that the effect of fan speed is essentially binary, either fast enough to circulate sufficient air or not.

So far, we have found that a zonal model remains invariant as different parameters change. At this point, we decided to investigate how to exploit such a model for reducing power consumption. This is the main topic of discussion of the following section.

3.3 Workload Assignment

One of the most important use cases of zones in a data centre is distributing workload between them. One approach is to use the same paradigm as divide-and-conquer. We first consider the problem when cooling unit setpoints are fixed, i.e. the workload manager is not able to also control the cooling units. This will also give us insight into whether there may be advantage into unbalancing the load to correspondingly unbalance the cooling requirements. The initial idea is shown in Figure 21. In the algorithm presented in this figure, we are trying to distribute workload between the zones such that the cold zone - which is the better cooled zone - handles more workload than the hot zone, which has higher temperature. Suppose that U is the workload assigned to the data centre. Notice that the utilization of each server is equal to U if the workload was uniformly distributed, in other words,

$$\forall i \in \{1, 2, \dots, N\} : u_i = U \quad (2)$$

The first step is to determine whether it is possible to put the entire workload on the cold zone servers and respectively increase the setpoint of the hot zone while decreasing its fan speed to the lowest possible speed. The servers in the hot zone would then be idle. The lowest possible fan speed was found to be 40%. Any further decrease would result in hot spots (in which the temperature is higher than 28°C) being generated.

If the workload completely fits in the cold zone, there is no need to use any of the servers in the hot zone. This is case 1 where there is no need to use servers in the hot zone at all and is shown on the top left of Figure 21. On the other hand, some assigned workloads are too big to be put on only the servers in the cold zone. Henceforth, servers in the hot zone will be needed to be used. This is referred to as case 2 in the flowchart. It may not be necessary to use all the servers in the hot zone. The required servers are referred to as potentially needed hot servers (servers in the hot zone that will potentially be used) in the flowchart. The number of potential servers is not necessarily equal to the number of servers in the hot zone, but regardless, we would need to raise the fan speed of the cooling unit closer to the hot zone (in this case to 60%) to ensure that enough air flow is directed to the working servers in the hot zone.

In the worst case, we need all the servers in the hot zone to work at 100%. Otherwise, only a fraction of the servers in the hot zone is enough to process the remaining workload. Moreover, a workload that requires all servers in the hot zone could be demanded at which point it would be best advised to use uniform distribution with adequate cooling settings.

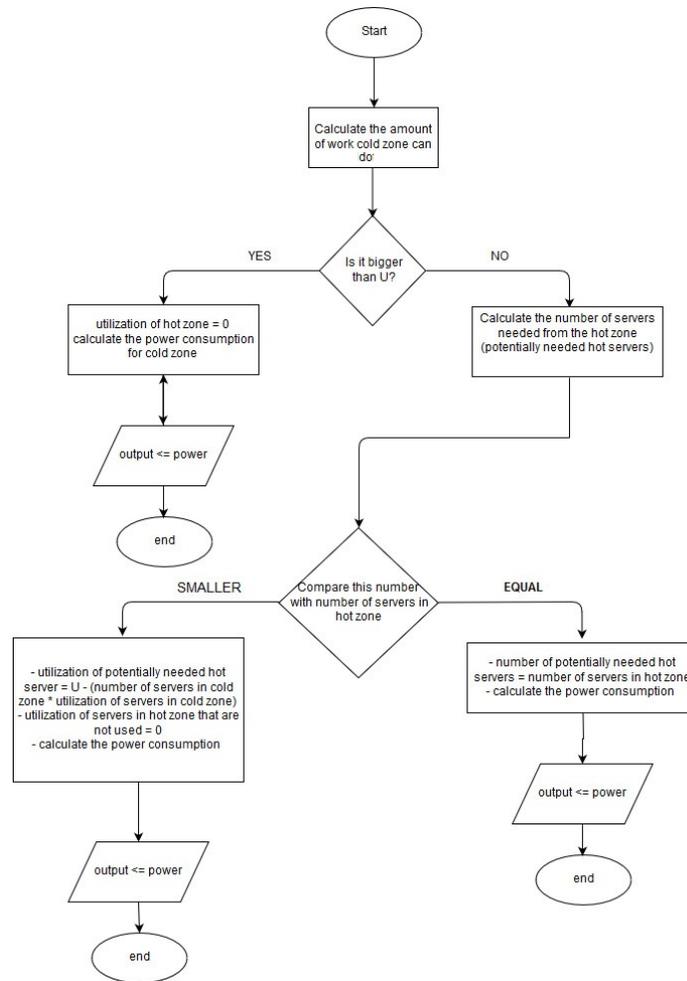


Figure 21: Zonal model workload assignment algorithm

As explained above, three scenarios are explained in Figure 21.

- U (the entire workload) is small enough to be processed by the cold zone only. In this scenario, servers located in the hot zone are idle (utilization is 0% but servers fans are still working).
- It is necessary to use servers in the hot zone as well but not all of them. Those servers in the hot zone which are being used are referred to as

potential needed servers in the hot zone.

- All the servers in the hot zone are needed as well as all the servers in the cold zone. In this scenario the difference is between the server utilization in the hot zone and the cold zone.

To compare the power consumption of each scenario, we will need a power model. This power model is discussed in the next chapter.

4 Cooling Power Consumption Optimization

A power model is needed to:

- (a) Compare the power consumption of each of the cases mentioned in the last chapter
- (b) Formulate an optimization problem in order to jointly optimize operational parameters with a goal of minimizing power consumption of cooling units

In this chapter, we are concentrating on the cooling power consumption only.

This is a reasonable thing to do because

- As mentioned in Chapter 1, a large portion of power consumption in a data centre is accounted for by the cooling units
- The IT power (the amount of power that servers use) does not change very much given a particular overall utilization

If the IT power is a direct function of the whole workload (U), then it is relatively insensitive to parameters such as where the workload is placed. In other words, IT power consumption need not be considered unless it is a function of each servers' utilization (u_j). As discussed above, in this thesis we are only focusing on the power consumption of cooling units and do not consider IT power consumption.

4.1 Power Model

There is a broad range of data centre power models in the literature. The power model used in [19] considers power consumption of the cooling unit in both idle and working states:

$$P_{CRAC} = P_{CRAC, idle} + ((1 + CoP) \cdot P_{heat}) \quad (3)$$

where

$$P_{heat} = 1.33 \times 10^{-5} \cdot \frac{P_{sf}^{max}}{\eta_{heat}} \cdot f. \quad (4)$$

The value f in (4) corresponds to the airflow rate in cubic meters per hour and P_{sf}^{max} is the maximum server load in kW . $P_{CRAC, idle}$ is the power consumption of the cooling unit in the idle mode which is normally between 10-30% of P_{sf}^{max} . Also, η_{heat} is the efficiency of the removal of heat from the system. The Coefficient of Performance or CoP is the ratio of the heat removed by the cooling unit to the amount of work done by the cooling unit. It should be noted that CoP of a cooling system is not constant and has a direct relationship with cooling unit setpoint.

$$CoP = \frac{Q}{W} \Rightarrow W = \frac{Q}{CoP} \quad (5)$$

According to (5), the necessary work to remove the heat (W) has an inverse relation with the CoP , so a higher CoP corresponds to greater efficiency. Figure 22 shows how CoP changes with the setpoint temperature of a cooling unit [20].

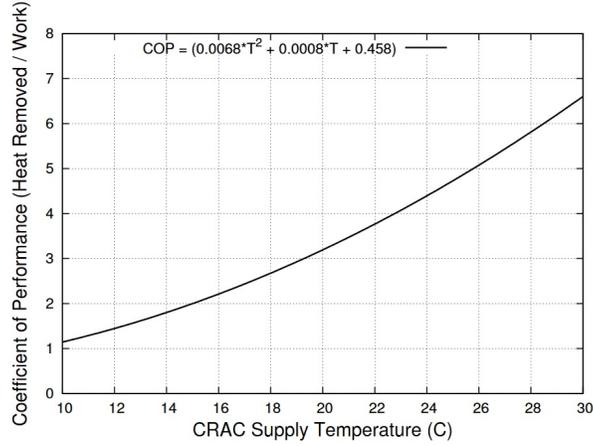


Figure 22: The relation between CoP and setpoint temperature. As the cooling unit produces air with higher temperature, CoP increases; meaning there is less work required to remove the heat.[20]

Research done by Phung, Lee and Zomaya in [21] presents a light-weight power monitoring software system. This software is application agnostic; in other words no assumption has been made in relation to typical energy use patterns of particular classes of programs. This software is mostly used to find the power consumed by individual applications or even individual threads in a virtualized environment. Furthermore, Lent [22] introduces a steady state model for total power consumed in a data center based on the power consumption of HVAC (Heating, Ventilation and Air Conditioning) equipment, computer equipment and other constant factors such as power delivery, network equipment and illumination.

$$\Pi = \Pi_{hvac} + \Pi_{comp} + \Pi_{other} \quad (6)$$

As discussed earlier, in this thesis we are only focusing on the cooling unit power consumption. The basis of the power model used in this thesis is the model presented in [23]. According to this model, the power consumed by a cooling unit is directly proportional to $T_{in,i} - T_{out,i}$ and inversely proportional to $CoP(T_{out,i})$. In order to include n_i in the power model, we also assume that the power consumption is proportional to n_i . The power consumption of the i th cooling unit is then given by the following formula:

$$Power_i = \frac{C \cdot n_i \cdot (T_{in,i} - T_{out,i})}{CoP(T_{out,i})}, \quad (7)$$

where C in (7) refers to a constant value. Without loss of generality, we can re-scale the power consumption and set $C = 1$, so the power consumption of cooling unit i is

$$Power_i = \frac{n_i \cdot (T_{in,i} - T_{out,i})}{CoP(T_{out,i})}, \quad (8)$$

To calculate the CoP , we are using the same model as [20] and [24]. In this

model, $CoP(T_{setpoint,i})$ is calculated using

$$CoP(T_{setpoint,i}) = 0.0068T_{setpoint,i}^2 + 0.0008T_{setpoint,i} + 0.458. \quad (9)$$

The relation (9) implies that by increasing $T_{setpoint,i}$, we can improve efficiency and reduce cooling cost [25].

It should be noted that, $T_{in,i}$ - inlet temperature of cooling unit i - is actually the outlet temperature of servers in the corresponding zone and $T_{out,i}$ - outlet temperature of cooling unit i - is the cold air temperature provided by cooling unit i which is the same as the setpoint for cooling unit i ($T_{setpoint,i}$). Hence, if we assume that the outlet temperatures of all servers in cooling unit i 's zone are identical, the power model can be re-written as:

$$Power_i = \frac{n_i(T_{out,j} - T_{setpoint,i})}{CoP(T_{setpoint,i})}, \quad \forall j : Z(j) = i, \quad (10)$$

where $Z(j)$ is the zone for server j . Based on this, the power consumption of each cooling unit can be calculated using the setpoint and outlet temperature of the zone corresponding to that cooling unit. Total power consumption is then the summation of the power consumption of cooling units 1 and 2. Throughout this work, indices i and j are used for cooling units and servers, respectively.

In the next section, we will compare uniform and non-uniform workload distribution using the provided power model.

4.2 Comparing Power Consumption

The best way to quantify the savings in power consumption is by looking at power consumption curves. Uniform workload distribution is the simplest way to assign workload and is preferred in terms of system performance, as it minimizes response time latency. We have considered two workloads here, 20% and 50%.

In the case of overcooling, when the setpoints of the cooling units are both set at 16°C, fan speed is set to be constant (60% of maximum) and the workload is distributed uniformly, the power consumption is calculated in Table 2.

Uniform 20% (kW)	Uniform 50% (kW)
20.18	23.78

Table 2: Power consumption for two cases of uniform workload distribution

When the workload is distributed non-uniformly based on the algorithm described in the last chapter and the cooling unit setpoints are changed accordingly, while the fan speeds remain unchanged, the power consumption, as shown by Figure 23, decreases. We assume uniform usage across time for a given server, and across servers.

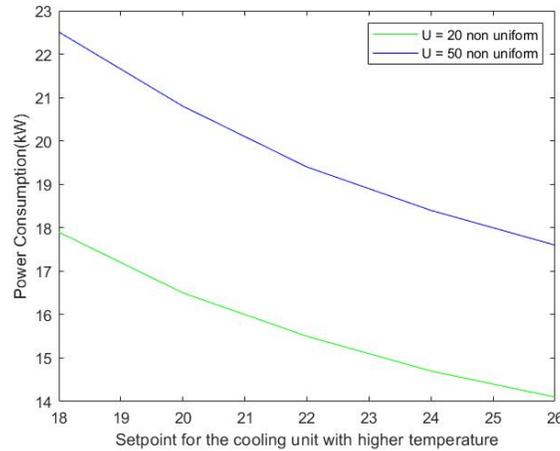


Figure 23: Comparing power consumption of non-uniform workload distribution for workloads 20% and 50% - Notice that the setpoint for one of the cooling units is fixed at 16°C

In Figure 23, the x-axis shows the temperature for the cooling unit with the higher setpoint and the y-axis shows the power consumption. For this scenario,

we have two choices of setpoint: one low and the other one high. We considered the low setpoint to be equal to 16°C and increased the second setpoint (high setpoint) gradually from 16°C to 26°C . Comparing the experiment with both setpoints at 16°C with the experiment having setpoints 16°C and 26°C at 20% utilization, we find that the power consumption decreases from 20.18 KW to 14.1 KW, a saving of 31%. Similarly, when comparing the same experimental setup but with a total workload of 50%, we find a decrease of 6.1 kW, or 25%.

However, when looking at all possible combinations of setpoints in Figure 24, we find that in some cases it is better to have uniform distribution. The following figure at 20% utilization shows that it would be less costly to have a uniform workload distribution with both setpoints at 20°C rather than having two setpoints of 18°C and 26°C .

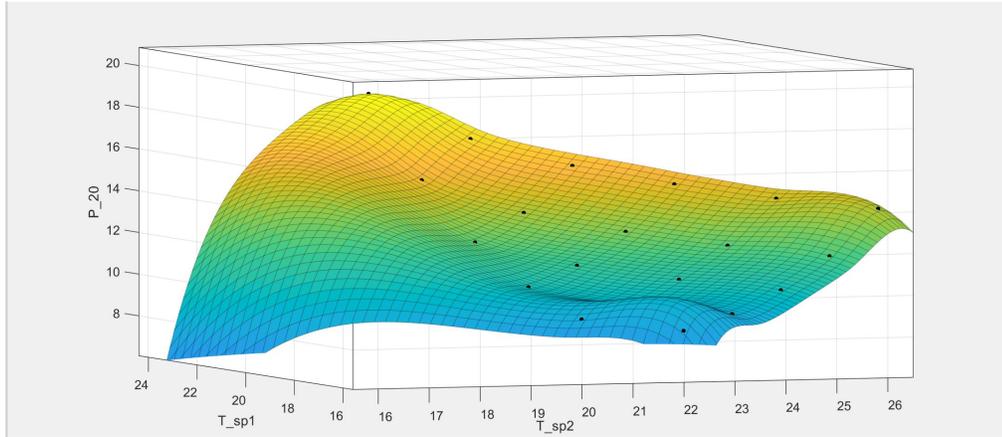


Figure 24: Power Consumption for non-uniform workload distribution when $n_1 = n_2$

In order to derive a comprehensive conclusion, we need to jointly optimize all factors (utilizations and setpoints), hence we need to formulate the problem as an appropriate optimization problem. In what follows, this optimization problem is introduced and solved.

4.3 Optimization Problem

Minimizing the power consumption of cooling units while maintaining temperature constraints can be solved using the following optimization problem. In Section 4.1 we mentioned that we are considering one inlet and one outlet temperature per zone rather than taking each servers' inlet and outlet temperature into account. In order to better clarify the notation for the optimization problem, we define a function $Z(j)$, which denotes the zone of server j (the cooling unit that is associated with the server). Because we only have two zones, $Z(j)$ only assumes two possible values, so we can write the function in the following form:

$$Z(j) = \begin{cases} 1 & \text{if server } j \text{ is in zone 1} \\ 2 & \text{if server } j \text{ is in zone 2} \end{cases} \quad (11)$$

Below is the optimization problem. Decision variables are utilization of servers (u_j), cooling unit setpoints ($T_{setpoint,i}$) and the number of servers in each zone (n_i), which indicates how big each zone is. By solving this optimization problem, we will find the optimal size of zones, setpoint temperatures and server

utilizations in order to minimize the power consumption of cooling units.

$$\text{Min}_{u_j, T_{setpoint,i}, n_i} \sum_{i=1}^2 Power_i$$

subject to

$$T_{cpu,j} = 13.4 + 26.5u_j^2 + 10.3u_j + 1.5T_{in,j} - 0.25u_jT_{in,j}, \quad j \in \{1, \dots, 60\} \quad (12)$$

$$T_{cpu,j} \leq T_{hot} = 70^\circ C, \quad j \in \{1, \dots, 60\} \quad (13)$$

$$T_{out,j} = 0.3391T_{in,j} + 16.66, \quad j \in \{1, \dots, 60\} \quad (14)$$

$$Power_i = \frac{n_i(T_{out,j} - T_{setpoint,i})}{CoP(T_{setpoint,i})}, \quad Z(j) = i, j \in \{1, \dots, 60\}, i \in \{1, 2\} \quad (15)$$

$$CoP(T_{setpoint,i}) = 0.0068T_{setpoint,i}^2 + 0.0008T_{setpoint,i} + 0.458, \quad i \in \{1, 2\} \quad (16)$$

$$T_{in,j} = T_{setpoint,i} + \delta, \quad Z(j) = i, \quad j \in \{1, \dots, 60\}, \quad i \in \{1, 2\} \quad (17)$$

$$\sum_{i=1}^2 u_i \cdot n_i = U \cdot N \quad (18)$$

This problem entails keeping CPU temperature below a pre-defined threshold. The constraint (12) can be found in [1] and was used to calculate $T_{cpu,j}$ as a function of $T_{in,j}$ and u_j . The constraint (14) was derived using experimental data through linear regression. The constraint (17) is a suggested estimate for $T_{in,j}$ in relation to $T_{setpoint,i}$. It follows the zonal model and states that a server's inlet temperature is equal to the corresponding cooling unit's setpoint plus a constant value (δ). For our purposes, we set δ equal to $2^\circ C$, but this could differ for other data centre architectures. We will see later that our structural results for the optimization problem are independent from δ . Also, F_i in (15) is set to 100%.

With the optimization problem formulated, we will solve it for u_j , $T_{setpoint,i}$ and n_i to jointly optimize all the parameters and make a holistic conclusion about workload distribution and cooling unit control.

4.4 Solving the Optimization Problem

We used Matlab's `fmincon` command to solve the optimization problem for our system. In the code, we have three parameters to optimize: servers' utilization (u_j), setpoint of each cooling unit ($T_{setpoint,i}$) and the number of servers in each zone (how big the zone is - n_i). It should be noted that U (the entire workload which is given) is not always the same as utilization of zones (u_i). Based on these parameters, we have three versions of the problem.

- **Case one:** only the utilization of each zone - u_i - is unknown. In this case we know n_i and $T_{setpoint,i}$. This is the easiest case, however it is the least practical.
- **Case two:** u_i and $T_{setpoint,i}$ unknown. For all of the instances that we considered, the proposed solution for this case is $u_1 = u_2$ and $T_{setpoint,1} = T_{setpoint,2}$. In other words workload distribution should be uniform and cooling unit setpoints should be the same. This case is the one of most practical interest and is the basis of our proof in the next chapter.
- **Case three:** u_i , $T_{setpoint,i}$ and n_i are unknown. For all of the instances that we considered, the proposed solution for this case is the same as the previous case in addition to $n_1 = n_2$. In other words, the zones are equal in size.

The reason for case two's importance is that the size of the zones are typically known beforehand (see Chapter 3) but we observed that the solution is independent of the sizes of the zones, as in case three. In conclusion, when we have a set of servers with one cooling unit at each end, distributing workload non-uniformly between the two zones does not appear to minimize power consumption. It appears that it is optimal to

1. Set the setpoint to the same temperature and as high as possible.

2. Distribute the workload uniformly between all the servers and make the zones have the same characteristics. In other words

$$\forall i, j \in \{1, 2\} \quad : \quad u_i = u_j \quad (19)$$

In the next part, we are going to prove that this solution is actually optimal and uniform workload distribution minimizes the power consumption of the cooling units. We consider different cases of power function structure and for each case, check the optimality of the solution. This is done in generality (beyond our particular architecture) and demonstrates that in general, these problems have relatively simple (and intuitive) solutions. In particular, in case of a convex power function, there is no need to unbalance the workload to try and reduce the cooling costs. Of course, this has synergy with the problem of data center performance - latency is minimized by balancing workloads across servers.

5 Proof of Optimality

In the previous chapter, we gained insight as to how the optimal solution might look like for three possible cases. In this chapter, we prove the optimality of our proposed solution for a more general version of our optimization problem. The idea to prove that our proposed solution is correct is to deviate from the proposed optimal utilization. We are going to increase the utilization of one zone by an arbitrary amount (δ) and decrease the utilization of the other zone correspondingly. If our solution is optimal, this change in utilization would cause an increase in power consumption of cooling units, proving the optimality of our solution.

As mentioned in Section 4.4, we have three possible cases of the optimization problem. Case one - in which the size of zones and setpoint temperature of cooling units are known and we only find the optimal utilization - is the least practical case and uniform workload distribution is not the unique solution for it. The other two cases are more general and using uniform workload distribution will optimize power consumption. Between these two cases, case two optimizes power consumption for any possible size of zones. In other words, case three has more degrees of freedom but it is a subset of case two because in case two, any size for zones can be specified and yet, uniform workload distribution is the unique solution for the optimization problem so we will prove that solution for case two is optimal. Proof of optimality for case three follows directly.

It is worth noting that the reason we are deviating from the utilization (and not other decision variables) is that the optimization problem is re-writable based on the utilization only; that is to say we can write the optimization problem having the utilizations u_j as the only decision variables. To do this, we need to plug the relations (12), (14), (16), (17) - the constraints of the optimization

problem - in (15), which is the power function. This gives us a function for $Power_i(u_j)$ that contains all the constraints inside it. In other words, it is no longer in form of an optimization problem with a set of explicit constraints. The constraints are embedded inside $Power_i(u_j)$. This is done in the following manner:

$$T_{cpu,j} = 70^\circ C \Rightarrow 70 = 13.4 + 26.5u_j^2 + 10.3u_j + 1.5T_{in,j} - 0.25u_jT_{in,j} \quad (20)$$

This gives us a function for $T_{in,j}$ that is solely based on u_j :

$$T_{in,j} = \frac{56.6 - 26.5u_j^2 - 10.3u_j}{1.5 - 0.25u_j} \quad (21)$$

The function (21) will be used instead of $T_{in,j}$. To re-write $T_{setpoint,i}$ based on u_j , we use (17):

$$T_{in,j} = T_{setpoint,i} + \delta \Rightarrow T_{setpoint,i} = T_{in,j} - \delta, \quad \forall j \in Z(i) \quad (22)$$

Notice that for our data centre, $\delta = 2^\circ C$. Replacing the relation (21) with $T_{in,j}$ in (22) will result in

$$T_{setpoint,i} = \frac{56.6 - 26.5u_j^2 - 10.3u_j}{1.5 - 0.25u_j} - 2, \quad \forall j \in Z(i) \quad (23)$$

In the relation (23), δ is replaced with $2^\circ C$. Thus we have a function for $T_{setpoint,i}$ based on u_j . Using the same logic for $T_{out,j}$ in (14), we can re-write it based on u_j only:

$$\begin{aligned} T_{out,j} &= 0.3391T_{in,j} + 16.66 \Rightarrow \\ T_{out,j} &= 0.3391\left(\frac{56.6 - 26.5u_j^2 - 10.3u_j}{1.5 - 0.25u_j}\right) + 16.66 \end{aligned} \quad (24)$$

Replacing (21), (23) and (24) in our power function in (15) will give the

power consumption as a function of the utilizations in zone $Z(i)$, $Power_i(u_j)$:

$$Power_i = \frac{n_i(0.3391(\frac{56.6 - 26.5u_j^2 - 10.3u_j}{1.5 - 0.25u_j}) + 16.66 - (\frac{56.6 - 26.5u_j^2 - 10.3u_j}{1.5 - 0.25u_j} - 2))}{0.0068(\frac{56.6 - 26.5u_j^2 - 10.3u_j}{1.5 - 0.25u_j} - 2)^2 + 0.0008(\frac{56.6 - 26.5u_j^2 - 10.3u_j}{1.5 - 0.25u_j} - 2) + 0.458} \quad (25)$$

As discussed earlier, (25) has all the constraints of the optimization problem inside it, except for the constraints that the utilizations must be non-negative and that the overall utilization is U . In the next sections we consider power functions based on their shape into three groups: convex power functions, concave power functions and power functions that are neither convex nor concave and study the optimality for each group independently.

5.1 Convex Power Function

As discussed earlier in this chapter, the way we will prove the optimality of our proposed solution is to start with the uniform workload distribution for both of the zones. We then increase the utilization of one zone by δ and decrease the utilization of the other zone correspondingly and compare the power consumption with the initial power consumption. If the uniform workload distribution minimizes the power consumption, then we expect the initial power consumption to have a smaller value.

Initially both zones have the same utilization. We expect to see an increase in power consumption once a deviation is made to the utilization. Figure 25 shows the change in power consumption based on the amount of change that is made in server utilization. The x-axis (δ) refers to the deviation made to the utilization and the y-axis refers to the change caused in power by δ , which is the difference between the power consumption after the utilizations have been changed and the power consumption for the uniform distribution of utilization. As can be

seen in this figure, the difference is always positive which means deviating from the initial server utilization (u_j) causes an increase in the power consumption of the cooling units, which matches our expectation.

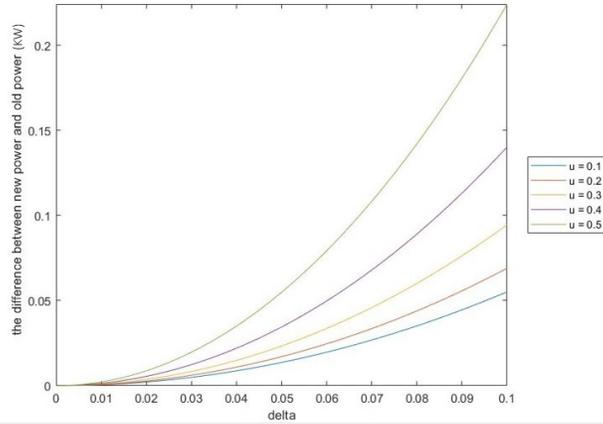


Figure 25: The difference in power consumption caused by deviation from uniform workload distribution

Theorem 1. *Let $P(u)$ be an increasing power function. Assume that the number of servers in each zone is given. If $P(u)$ is convex, then uniform workload distribution will minimize the power consumption of the cooling units.*

Proof. We know if the second derivative of a function is positive, then that function is convex. Also, if the power function is convex, it implies that power consumption is increasing as a function of utilizations of servers in the zone. The general shape of a convex power function is as in Figure 26 below.

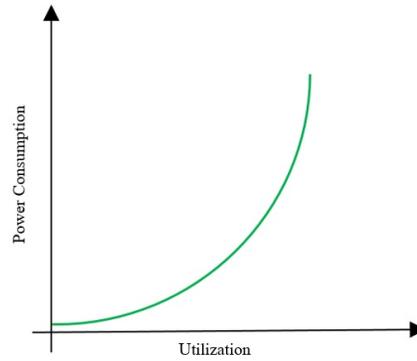


Figure 26: The general shape of power function

Figure 27 shows the change in power consumption when there is a deviation (both increasing and decreasing deviation). We define $P(u)$ as the power consumption for utilization u and $\delta > 0$ corresponds to the deviation in utilization. Here, $u + \delta$ is the increasing deviation and $u - \delta'$ is the decreasing deviation.

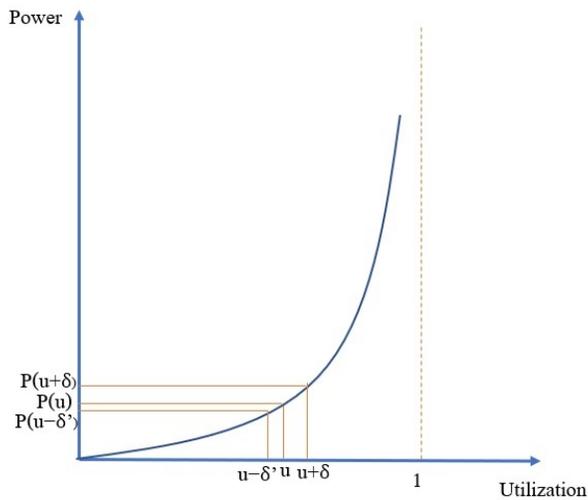


Figure 27: The change in power consumption when there is a deviation(both increasing and decreasing deviation) - case of convex power function

The deviation made in each zone is a function of the size of that zone. Assume

n_1 and δ are the number of servers in zone 1 and the deviation made in zone 1 utilization, respectively. Similarly, n_2 and δ' are the number of servers in zone 2 and the deviation in zone 2 utilization. The entire workload (U) is equal to

$$n_1 \cdot u_1 + n_2 \cdot u_2 = U \quad (26)$$

Because U is not changing, it must be held constant after deviating from the uniform workload distribution:

$$n_1 \cdot (u_1 + \delta) + n_2 \cdot (u_2 - \delta') = U \quad (27)$$

Combining (26) and (27),

$$n_1 \cdot (u_1 + \delta) + n_2 \cdot (u_2 - \delta') = n_1 \cdot u_1 + n_2 \cdot u_2 \Rightarrow \quad (28)$$

$$n_1 \cdot \delta - n_2 \cdot \delta' = 0 \Rightarrow \quad (29)$$

$$\frac{n_1}{n_2} = \frac{\delta'}{\delta} \quad (30)$$

The equation (30) shows that the amount of deviation made in each zone's utilization is proportional to the number of servers in that zone (size of the zone). We cannot assume the incremental deviation for the utilization in one zone corresponds to that in the other zone (but in the opposite direction). In other words, if we increment the utilization of one zone by δ , we cannot decrease the utilization of the other zone by the exact same amount δ . Using (30), it should be $\frac{n_1}{n_2} \cdot \delta$.

Also, we can assume that $P_i(u)$ - the power consumption of cooling unit i is

the result of multiplying n_i by $\frac{(T_{in,i}-T_{out,i})}{CoP(T_{out,i})}$. We write the power consumption model in (25) in the form

$$P_i(u) = n_i \cdot \bar{P}_i(u), \quad (31)$$

where $\bar{P}_i(u)$ does not depend on n_i . Also, we know that the total power consumption is equal to the sum of the power consumption of both cooling units:

$$\begin{aligned} Power_{total} &= \sum_{i=1}^2 P_i(u) \\ &= P_1(u) + P_2(u) \\ &= n_1 \cdot \bar{P}_1(u) + n_2 \cdot \bar{P}_2(u) \end{aligned} \quad (32)$$

Our goal is to show the power consumption after deviating from the uniform workload distribution has a larger value than the power consumption of the uniform workload distribution:

$$P_1(u) + P_2(u) \leq P_1(u - \delta) + P_2(u + \delta') \quad (33)$$

The left hand side corresponds to uniform workload distribution and the right hand side corresponds to the deviated workload distribution. Referring to the relation (30), we know that $\delta' = \delta \cdot \frac{n_1}{n_2}$, so we can re-write (33) as

$$P_1(u) + P_2(u) \leq P_1(u - \delta) + P_2(u + \delta \cdot \frac{n_1}{n_2}) \quad (34)$$

According to the relation (32):

$$P_1(u) = n_1 \bar{P}_1(u) \quad , \quad P_2(u) = n_2 \bar{P}_2(u) \quad (35)$$

By plugging the relations in (35) into (34) we will have

$$n_1 \bar{P}_1(u) + n_2 \bar{P}_2(u) \leq n_1 \bar{P}_1(u - \delta) + n_2 \bar{P}_2(u + \delta \frac{n_1}{n_2}) \quad (36)$$

Because the power function $P_i(u)$ is convex and increasing, (36) can be rearranged as

$$n_1 [\bar{P}_1(u) - \bar{P}_1(u - \delta)] \leq n_2 [\bar{P}_2(u + \delta \frac{n_1}{n_2}) - \bar{P}_2(u)] \quad (37)$$

We can write the difference on the right hand side of (37) using the following relation:

$$\bar{P}_2(u + \frac{n_1}{n_2} \delta) - \bar{P}_2(u) = \bar{P}'_2(u) \frac{n_1}{n_2} \delta + \epsilon(u) \quad (38)$$

where $\epsilon(u)$ is the error in using the derivative at u as an approximation. Because the function $\bar{P}_2(u)$ is increasing and convex, $\bar{P}'_2(u) \cdot \delta$ is an underestimate of $\bar{P}_2(u + \frac{n_1}{n_2} \delta) - \bar{P}_2(u)$, hence the error is positive as shown in Figures 28 and 29 below.

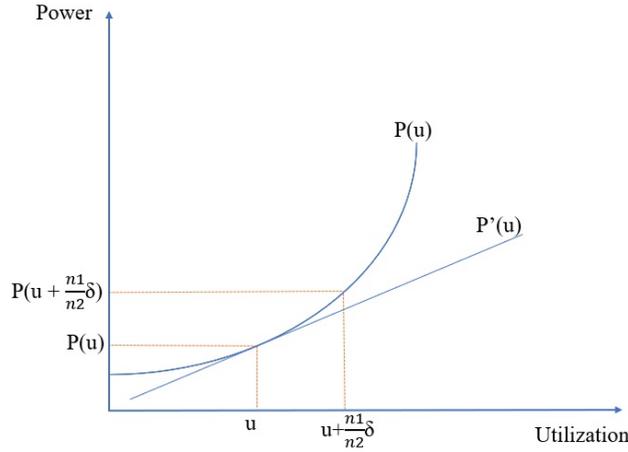


Figure 28: The derivative of the increasing and convex function $P(u)$ at point u and its positive error.

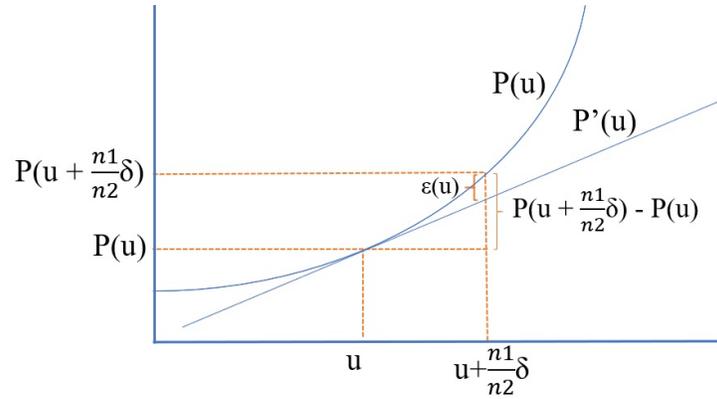


Figure 29: The derivative of the increasing and convex function $P(u)$ at point u and its positive error - a closer view

Notice in Figures 28 and 29, $P(u)$ stands for the power function. As can be seen in these figures, $P'(u)$ is underestimating the value of $P(u + \frac{n_1}{n_2}\delta) - P(u)$ because we are approaching to u from the right hand side - where the points on the horizontal axis have higher values than u . Therefore $\epsilon(u)$ should be added to compensate for the shortage. Applying the same reasoning to the left hand side of (37) yields

$$\bar{P}_1(u) - \bar{P}_1(u - \delta) = \bar{P}'_1(u) \cdot \delta - \epsilon'(u) \quad (39)$$

In this case, $\epsilon'(u)$ should be decreased from $\bar{P}'_1(u) \cdot \delta$ because $\bar{P}'_1(u - \delta) \cdot \delta$ is an overestimate for $\bar{P}_1(u) - \bar{P}_1(u - \delta)$, as shown in Figures 30 and 31.

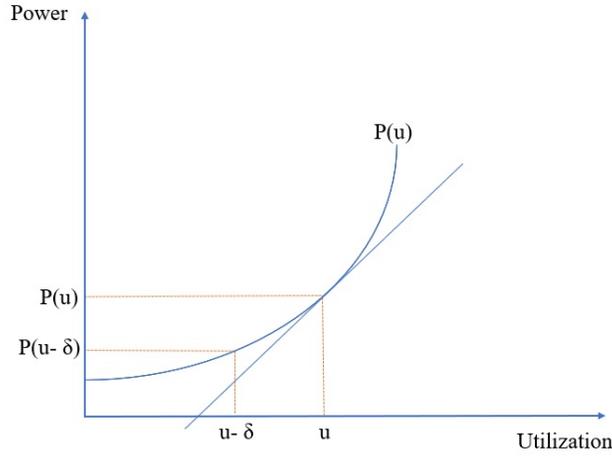


Figure 30: The derivative of the increasing and convex function $P(u)$ at point u and its negative error.

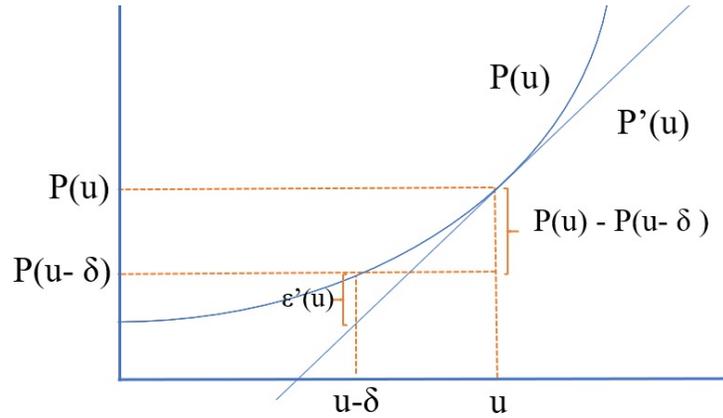


Figure 31: The derivative of the increasing and convex function $P(u)$ at point u and its negative error - a closer view

As shown in Figures 30 and 31, we are approaching to u from the left hand side. Therefore, $\bar{P}'_1(u) \cdot \delta$ overestimates the value of $\bar{P}_1(u) - \bar{P}_1(u - \delta)$. If we replace (38) and (39) with their corresponding values in (37), we will have

$$\left. \begin{aligned} n_1[\bar{P}_1(u) - \bar{P}_1(u - \delta)] &\leq n_2[\bar{P}_2(u + \frac{n_1}{n_2}\delta) - \bar{P}_2(u)] \\ \bar{P}_2(u + \frac{n_1}{n_2}\delta) - \bar{P}_2(u) &= \bar{P}'_2(u)\frac{n_1}{n_2}\delta + \epsilon(u) \\ \bar{P}_1(u) - \bar{P}_1(u - \delta) &= \bar{P}'_1(u)\delta - \epsilon'(u) \end{aligned} \right\} \Rightarrow$$

$$n_1[\bar{P}'_1(u)\delta - \epsilon'(u)] \leq n_2[\bar{P}'_2(u)\frac{n_1}{n_2}\delta + \epsilon(u)] \Rightarrow \quad (40)$$

$$n_1 \cdot \bar{P}'_1(u)\delta - n_1 \cdot \epsilon'(u) \leq n_1 \cdot \bar{P}'_2(u)\delta + n_2 \cdot \epsilon(u) \quad (41)$$

In (41), $\bar{P}'_1(u)\delta$ and $\bar{P}'_2(u)\delta$ have the same value, so (41) is equivalent to

$$-n_1\epsilon'(u) \leq n_2\epsilon(u) \quad (42)$$

In (42), both $\epsilon(u)$ and $\epsilon'(u)$ have positive values, therefore (42) is a tautology and is true for all u .

□

Corollary 1.1. *Let $P(u)$ be an increasing power function. If $P(u)$ is convex, according to Theorem 1, uniform workload distribution and equal sized zones will minimize the power consumption of servers (proof of case three).*

We proved uniform workload distribution minimizes power consumption of cooling units in the case of a convex power function. In order to apply this conclusion to our specific power model, we assume that $T_{cpu,j}$ is fixed to 70°C because the optimization problem is trying to minimize the power, so the solver sets $T_{cpu,j}$ to the maximum amount possible. Considering $T_{cpu,j}$ to be equal to 70°C and based on (12) in the optimization problem, making a change in utilization would cause a change in $T_{in,j}$. Because $T_{in,j}$ and $T_{out,j}$ of the servers are related together with (14), the change in $T_{in,j}$ will affect $T_{out,j}$. Similarly, $CoP(T_{setpoint,i})$ will also change as well. So, deviating from the proposed optimal

u_j will affect the power consumption. Figure 27 shows the change in power consumption when there is a deviation from uniform workload distribution. Note that the power function is convex in this case, as shown below.

In the next step of applying the conclusion of our specific power model, we use the characteristics of second derivative of a function. If we prove that the second derivative of the power function is always positive, the power function is convex. We calculated the second derivative of our example power function and then using the matlab command `isAlways` checked the sign of the second derivative. After confirming that the power function has a positive second derivative (thus is convex), Theorem 1 yields that uniform workload distribution minimizes the power consumption. Figure 32 shows the second derivative plot. It can be seen that the second derivative of power is always positive.

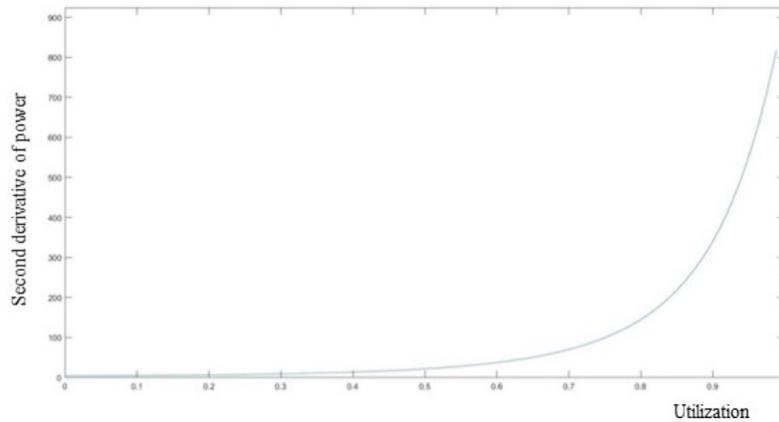


Figure 32: Second derivative plot

So, we proved that when we have a convex power function, uniform workload distribution minimizes the power consumption. We also applied this conclusion to our data centre and specific power model and proved that this conclusion still stands. In the next section, we study the case of a concave power function

and determine whether uniform or non-uniform workload distribution minimizes power consumption.

5.2 Concave Power Function

Theorem 2. *Let $P(u)$ be an increasing power function. Assume we know the number of servers in each zone (n_j). If $P(u)$ is concave, then as the difference between utilization of each zone increases, the power consumption decreases.*

Proof. We take a proof by contradiction approach. Assume Theorem 2 is not true and as for convex $P(u)$, uniform workload distribution also minimizes the power consumption for concave $P(u)$. We use the function in (31) and re-write the relations for concave $P(u)$. Although the beginning of the proof is identical to the proof for convex case, as a reminder we again repeat it here.

The total power consumption of both cooling units is equal to the sum of power consumption of each one of them:

$$\begin{aligned} Power_{total} &= \sum_{i=1}^2 P_i(u) \\ &= P_1(u) + P_2(u) \\ &= n_1 \cdot \bar{P}_1(u) + n_2 \cdot \bar{P}_2(u) \end{aligned} \tag{43}$$

For a concave power function, we wish to show that the power consumption after deviating from the uniform workload distribution has a larger value than the power consumption of the uniform workload distribution:

$$P_1(u) + P_2(u) \leq P_1(u - \delta) + P_2(u + \delta') \tag{44}$$

As for a convex power function, the left hand side corresponds to uniform workload distribution and the right hand side corresponds to the deviated

workload distribution. Applying (30) on (44) gives

$$P_1(u) + P_2(u) \leq P_1(u - \delta) + P_2(u + \delta \cdot \frac{n_1}{n_2}) \quad (45)$$

Using the relations (43) and (31), we can re-write (45) as

$$n_1 \bar{P}_1(u) + n_2 \bar{P}_2(u) \leq n_1 \bar{P}_1(u - \delta) + n_2 \bar{P}_2(u + \delta \frac{n_1}{n_2}) \quad (46)$$

Rearranging (46) results in

$$n_1[\bar{P}_1(u) - \bar{P}_1(u - \delta)] \leq n_2[\bar{P}_2(u + \delta \frac{n_1}{n_2}) - \bar{P}_2(u)] \quad (47)$$

So far, the proof for the concave case has been identical to the proof for the convex case. Applying the definition of derivative on the right hand side of (47) gives

$$\bar{P}_2(u + \frac{n_1}{n_2}\delta) - \bar{P}_2(u) = \bar{P}'_2(u) \frac{n_1}{n_2} \delta - \epsilon(u) \quad (48)$$

In (48), $\bar{P}'_2(u) \frac{n_1}{n_2} \delta$ overestimates the value of $\bar{P}_2(u + \frac{n_1}{n_2}\delta) - \bar{P}_2(u)$, therefore it is decreased by $\epsilon(u)$ as shown in Figures 33 and 34.

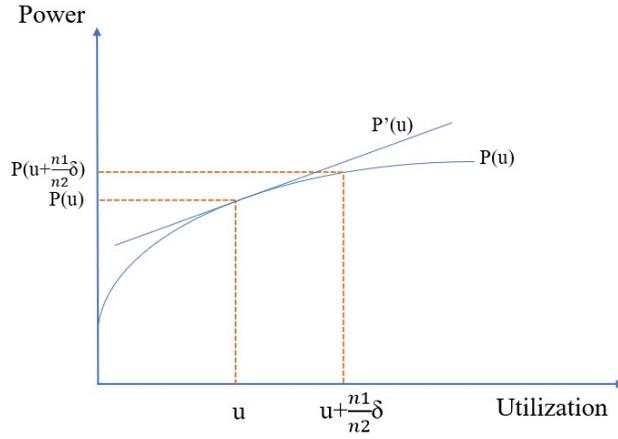


Figure 33: The derivative of the increasing and concave function $P(u)$ at point u and its negative error

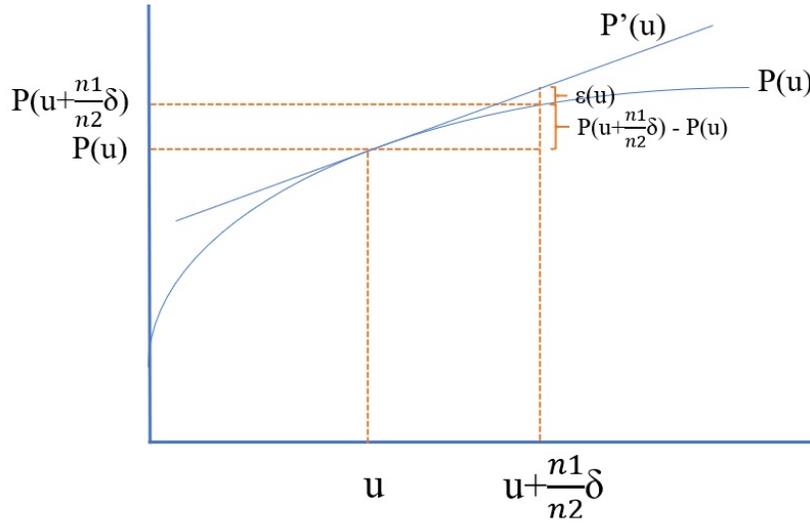


Figure 34: The derivative of the increasing and concave function $P(u)$ at point u and its negative error - a closer view

Applying the same reasoning to the left hand side of (47) will give

$$\bar{P}_1(u) - \bar{P}_1(u - \delta) = \bar{P}'_1(u)\delta + \epsilon'(u) \tag{49}$$

As shown in Figures 35 and 36, in (49) we have underestimated the value of $\bar{P}_1(u) - \bar{P}_1(u - \delta)$ so $\epsilon'(u)$ is added to the right hand side of (49).

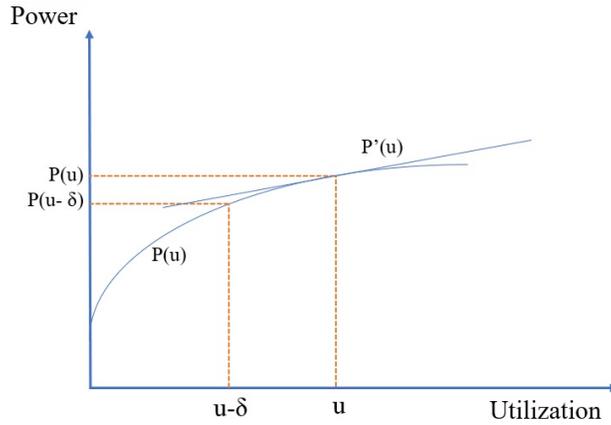


Figure 35: The derivative of the increasing and concave function $P(u)$ at point u and its positive error

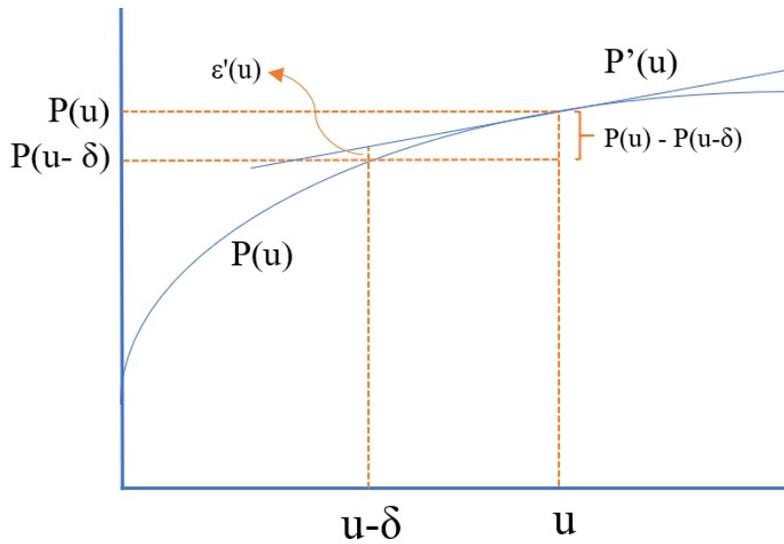


Figure 36: The derivative of the increasing and concave function $P(u)$ at point u and its positive error - a closer view

Similar to (40), we replace the values of $\bar{P}_1(u) - \bar{P}_1(u - \delta)$ and $\bar{P}_2(u + \frac{n_1}{n_2}\delta) - \bar{P}_2(u)$ in (37) with their corresponding values:

$$\left. \begin{aligned} n_1[\bar{P}_1(u) - \bar{P}_1(u - \delta)] &\leq n_2[\bar{P}_2(u + \frac{n_1}{n_2}\delta) - \bar{P}_2(u)] \\ \bar{P}_2(u + \frac{n_1}{n_2}\delta) - \bar{P}_2(u) &= \bar{P}'_2(u)\frac{n_1}{n_2}\delta - \epsilon(u) \\ \bar{P}_1(u) - \bar{P}_1(u - \delta) &= \bar{P}'_1(u)\delta + \epsilon'(u) \end{aligned} \right\} \Rightarrow$$

$$n_1[\bar{P}'_1(u)\delta + \epsilon'(u)] \leq n_2[\bar{P}'_2(u)\frac{n_1}{n_2}\delta - \epsilon(u)] \quad (50)$$

Simplifying (50) results in

$$n_1 \cdot \bar{P}'_1(u)\delta + n_1 \cdot \epsilon'(u) \leq n_1 \cdot \bar{P}'_2(u)\delta - n_2 \cdot \epsilon(u) \quad (51)$$

$n_1 \cdot \bar{P}'_2(u)\delta$ cancel out, therefore

$$n_1 \cdot \epsilon'(u) \leq -n_2 \cdot \epsilon(u) \quad (52)$$

The inequality (52) can never be satisfied because both $\epsilon(u)$ and $\epsilon'(u)$ are positive, so the theorem is proved.

□

Theorem 2 shows that in case of a concave power model, increasing the difference between utilizations reduces power consumption. In fact, it is better to have the utilization of a zone at the maximum amount possible and then assign the remaining workload to the other zone.

In conclusion, when we have a power model for cooling units and we want to know which workload distribution minimizes the power consumption, we need to check the convexity of the power function. If it is convex, uniform workload distribution minimizes the power consumption. If it is concave, then it is better to make the utilization of each zone as different as possible.

We have discussed convex and concave power models so far. In practice, it appears that power functions are typically convex but in order to consider all possible cases of different power functions, in the next section we will discuss power models that are neither convex nor concave and how power consumption can be reduced in such cases.

5.3 Neither Convex nor Concave Power Function

Functions that are neither convex nor concave can be made by stitching together one or multiple concave functions with one or multiple convex functions. The function $\sin(x)$ is an example of this type of function as shown in Figure 37.

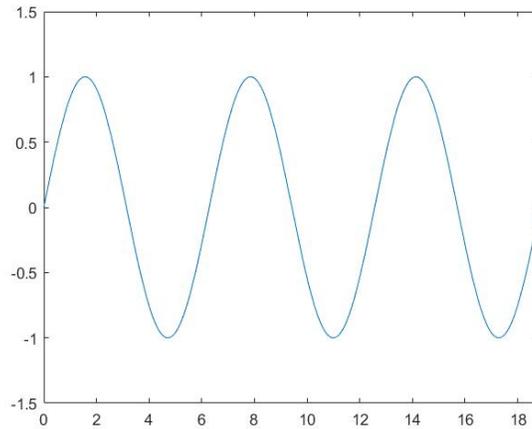


Figure 37: $\sin(x)$ is an example of a function that is neither convex nor concave and has more than one convex part and more than one concave part

In this thesis, we only consider the power functions consisting of one convex part and one concave part. These cases are shown in Figure 38. Figure 38-a shows a power function in which the first part is convex and the second part is concave (convex-concave power function). Figure 38-b shows a power function where the first part is concave and the second part is convex (concave-convex

power function).

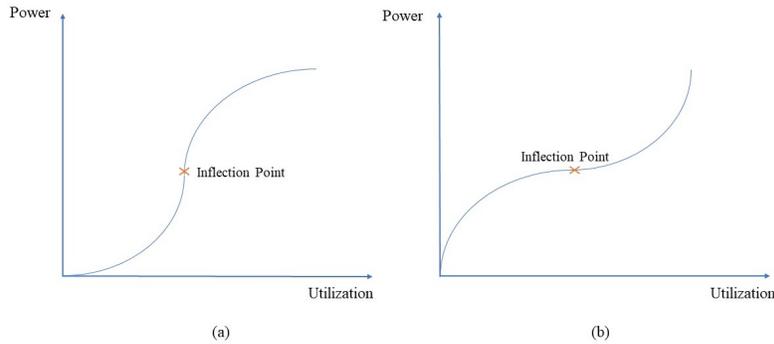


Figure 38: General shape of neither convex nor concave power function consisting of one convex part and one concave part

We concentrate here on the case depicted in Figure 38-b. We have three different possibilities for the utilization of each zone.

1. Utilizations of both zones are in the concave part of the power function, as seen in Figure 39-a
2. Utilizations of both zones are in the convex part of the power function, as seen in Figure 39-b
3. Utilization of one of the zones is in the convex part while the utilization of the other zone is in the concave part, as seen in Figure 39-c

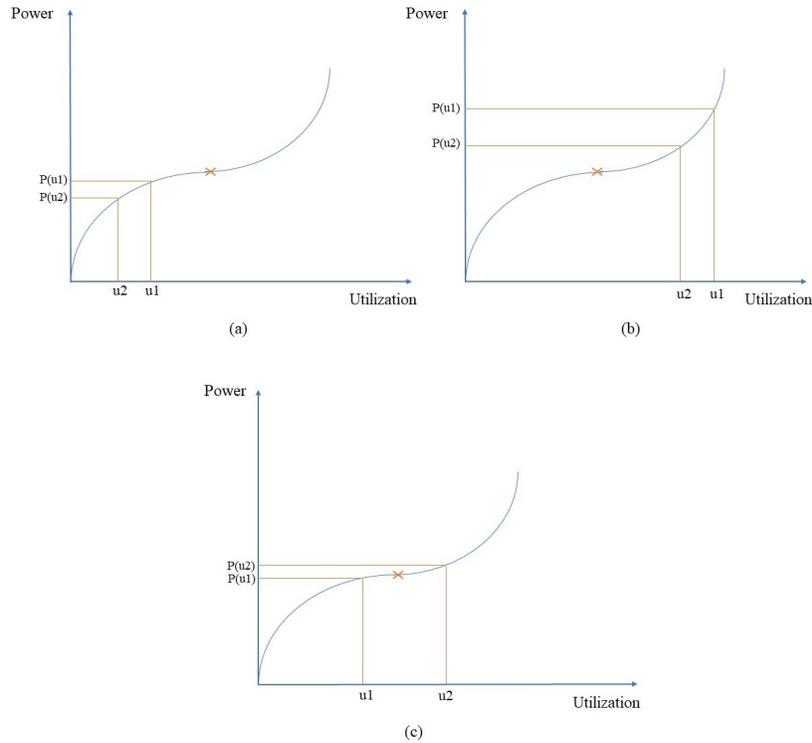


Figure 39: Three possibilities for utilization of zones when the power function is neither convex nor concave. (a) when both utilizations are in concave part (b) when both utilizations are in convex part (c) when one of them is in convex part while the other one is in concave part

Each of the cases in Figure 39-a and Figure 39-b has been discussed in the previous chapters. They are the same as concave and convex power function, respectively. The case in Figure 39-c is the topic of discussion here. It includes one utilization in the convex part of the power function while the other utilization is in the concave part and our goal here is to find the utilizations that minimize the power. Notice that in all of the cases in Figure 39, the function is symmetric with a symmetry axis of $x = \textit{inflection point}$. Studying non-symmetric neither convex nor concave functions is beyond the scope of this thesis.

When a function is neither convex nor concave - like the power function in

this section - it has at least one *inflection point*; which is the point in which the curve changes from being convex to concave or vice versa. The main idea here is to find the inflection point of the function first and start with both utilizations equal to the inflection point. Then we will deviate the utilizations from the inflection point and see the changes in the power function. We are expecting to see an increase in total power consumption, meaning when the utilizations deviate from the inflection point, it will increase the power consumption. This idea is shown in Figure 40.

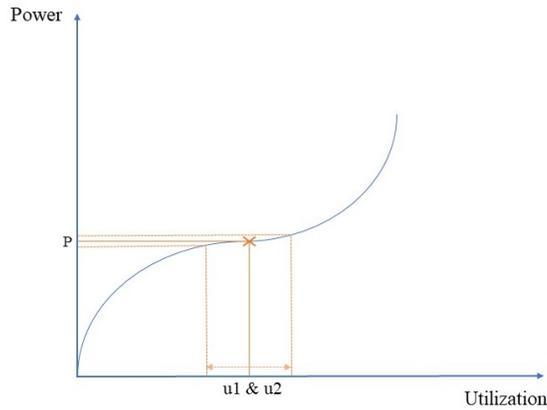


Figure 40: Initially both u_1 and u_2 are equal to inflection point and total power consumption has a value of $2P$. We are going to decrease u_1 and increase u_2 and see the changes in power

Notice that the power functions discussed in this chapter have the three following properties:

1. They all have a value equal to 0 when $u = 0$
2. The inflection point is between 0 and 1
3. They are increasing

We used matlab libraries to observe the changes in power consumption when the utilization deviates from the inflection point. At first, the utilization of both

of the zones is equal to the inflection point. In this case δ - which corresponds to the deviation - is zero. At each step we increased δ by 0.001 and then decreased the utilization in the concave part by δ . Subsequently, we added δ to the utilization in the convex part. At each step we calculated power after changing the utilization of both zones and subtracted it from the initial power consumption (when $\delta = 0$). Using the matlab `isAlways` command, we checked the sign of the difference between these two power consumptions. We expect this sign to be always positive, indicating that the total power consumption after deviating from the inflection point is more than the initial power consumption.

We repeated the above discussed procedure for all possible utilizations. Our results show that the output of the `isAlways` function is positive in all cases, meaning as we deviate from the inflection point more, we see the total power consumption going higher. Figure 41 shows the result of this procedure for a specific function with an inflection point of 0.5. As can be seen in Figure 41, as δ gets closer to 0.5, power consumption increases. Initially, when both u_1 and u_2 have the same value - which is equal to the inflection point of the power function - the power has a value of about 1.51. As u_1 and u_2 get closer to 0 and 1, respectively; the power consumption increases.

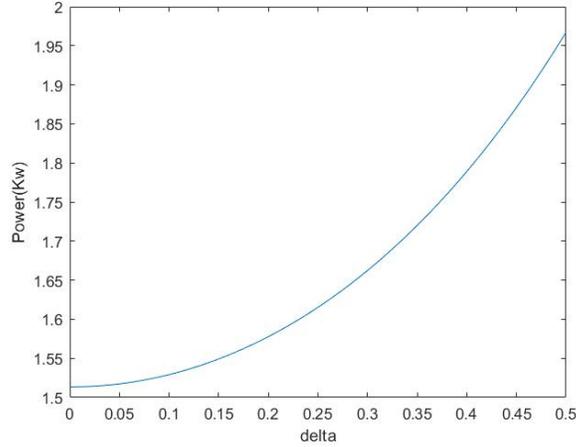


Figure 41: Changes in power when u_1 and u_2 are moved far from the inflection point. δ is the change made to the utilizations at each step. The further the utilization from the inflection point, the higher the power consumption

The conclusion in this part is that in case of a symmetric neither convex nor concave power function, we would like to keep the utilization of zones close to the inflection point in order to minimize the power consumption. The difficulty of doing so is that only one U satisfies this condition and the utilization of the zones cannot always be exactly equal to the inflection point. Also, one might keep the utilizations close to the inflection point by keeping the utilization of some servers equal to zero. For example, consider a power function with an inflection point at 0.5 and the entire workload is $U = 0.25$. Assume we have four servers. We can keep the utilization of two of the servers at 0.5 while the rest of the servers are off. Due to the combinatorial aspect of the problem, the solution for such a power functions is more complicated and is beyond the scope of this thesis.

What is shown in Figures 38, 39 and 40 only shows a specific case of power function with symmetric convex and concave parts. In practice we can have functions similar to Figure 42 with one part (either convex part or concave part)

smaller than the other.

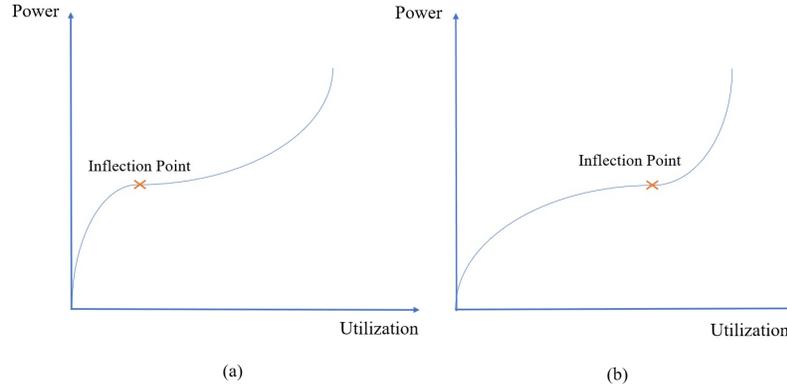


Figure 42: Examples of neither convex nor concave functions that are not symmetric

When the two sides of the inflection point are not symmetric, a deviation in utilization of one side does not result in the same deviation in reverse on the other side. For example, consider a power function of general form as given in Figure 42-a. In this case, the convex part on the right hand side of the inflection point stays flat for a long time while in the concave side power changes drastically with an arbitrary change in utilization; therefore, a deviation in utilization of this part does not affect the power that much. On the other hand, a deviation in the utilization of the concave part on the left hand side of the inflection point would cause a large change in the power. The solution for such a power function needs to take into account the trade off between the utilization of both parts. In this thesis, we do not cover this type of power function.

So far, we have found the optimal workload distribution for a data centre which minimizes power consumption based on the shape of the power function. This workload distribution works for convex and concave power functions. We also discussed neither convex-nor concave cases and the complexity of this case. The models we have used in the optimization problem are from the literature. We

would like to see if our conclusions work with different models in the optimization problem. In the next part of this research we find a model for $T_{out,j}$, which is one of the parameters in the optimization problem. Notice that we could find a new model for any of the optimization problem parameters ($Power_i$, $T_{cpu,j}$, $T_{in,j}$, etc.), but in the interest of time we chose only one of these parameters.

6 $T_{out,j}$ Model Estimation

The models that we have used for different parameters of our optimization problem are from the literature. In practice, each of these parameters need to be found based on the properties of the data centre. They can be calculated using either

- 1- Physics of the system. In this approach thermal equations are used to find the model.

or

- 2- A data driven model.

In this chapter, we explore the second approach. We focus on a model to estimate the outlet temperature of servers ($T_{out,j}$ denotes the outlet temperature of server j), which is one of the variables of the optimization problem. One way to gather the data we need for this model is to generate synthetic data using the data from the initial experiments, but the synthetic data is problematic in terms of accounting for the distance of each server from the cooling unit. Also, comparing the number of data points at the front and back of the servers, there are less of them at the back. So this option is not viable and we chose a second approach which is gathering data through running a new set of experiments. The outlet temperature of servers is also referred to as the temperature at the back of servers. In the next sections, we will discuss our approach in detail and compare our results with other models from the literature.

6.1 The Necessity of New Experiments

For the first half of this research, we conducted a series of experiments in which we changed cooling unit setpoints and server utilizations. Throughout

these experiments, we determined that there are two zones at the front of the servers which we named the hot zone and the cold zone. The initial experiments described in Chapter 3 gave us the following data:

- Temperature at the front of the servers
- Temperature at the back of servers

However, we need the following data for our model:

- Temperature at the front of the servers
- Temperature at the back of servers
- Temperature of CPU for each server
- The distance of server from the cooling unit

At the back of the racks we have 15 sensors, so there are 15 data points at the back of the racks. At the front, we have 33 sensors, so at the front of the racks we have 33 data points. Based on the data we gathered, we wanted to include $T_{in,j}$ and $T_{cpu,j}$ in our model. On the other hand, server utilization (u_j) is the same for all the servers in the same zone (uniform workload distribution). To obtain $T_{in,j}$, we used the following formula:

$$T_{in,j} = T_{setpoint,i} + \delta, \quad j \in Z(i) \quad (53)$$

Note that in (53), $T_{in,j}$ refers to all the servers in the zone corresponding to cooling unit i . Initially, we thought that we could modify our existing data by noise that follows a particular distribution (in our case uniform). Also, the formula to find $T_{cpu,j}$ depends on u_j and $T_{in,j}$. We have two key issues with the data that we generated in this manner:

- Because we use random noise to generate $T_{in,j}$, we are ignoring the effect of distance between cooling unit and server. It might be the case that a server is far from a cooling unit but the random noise generated for that particular server is smaller than for closer servers. $T_{in,j}$ for that server is going to be close to $T_{setpoint,i}$ while it potentially should be much bigger because it is a more distant server. Even if we generate biased random noise, it cannot truly reflect our data centre because the change in $T_{out,j}$ from one rack to the other is not always clear enough to be simulated by random noise.
- We have less data points at the back and more data points at the front. Using the average temperature at the back of the racks does not seem to be a suitable way to solve this inconsistency.

Therefore, conducting a new set of experiments in which we measure $T_{cpu,j}$, $T_{in,j}$ and $T_{out,j}$ directly seems necessary. In the next section, we explain the architecture of the second set of our experiments.

6.2 Architecture

In the new set of experiments, as mentioned before, we measure $T_{in,j}$, $T_{cpu,j}$ and $T_{out,j}$. Also, we change u_j randomly among 60 servers in our experimental data centre. Our desired model will accept $T_{in,j}$ and $T_{cpu,j}$ as inputs and generate $T_{out,j}$ as output. Also, because of the relation between the distance of the server from the cooling unit and its inlet temperature, we also include distance of server from cooling unit (d_j) as an input. Therefore in total we will have three input parameters and one output in our model.

To measure the above mentioned parameters, we use the same experimental data centre composed of 60 servers that we used for the first part of the research. Each rack has three sensors at the front and three sensors at the back to measure

the temperature in the cold chamber and hot chamber, respectively. All the operational parameters of the cooling unit can be controlled but we only change $T_{setpoint,i}$ for each cooling unit. Other parameters such as fan speed and water flow rate are fixed. $T_{cpu,j}$ is reported by each server. We ran 36 experiments and for each server we measured $T_{in,j}$, $T_{cpu,j}$, d_j and $T_{out,j}$. Therefore the data set we use has four columns and on the order of 2000 rows. We use 20% of this data as testing data and 80% as training data. In the next part we introduce our proposed models for $T_{out,j}$ and discuss them.

6.3 Results and Discussion

We implement different models for $T_{out,j}$ and then choose a model with reasonable trade-off between performance and complexity. The models are all polynomial and are implemented using Python's `PolynomialFeatures` library. For evaluation, we used Mean Square Error (MSE) and R-Squared Error (R^2) from the `sklearn.metrics` library. Also, as mentioned before, 20% of data is used for testing and the rest is used for training.

We used Python's `train_test_split` command from the `sklearn.model_selection` library to split the data set into training and testing sets. This command randomly chooses a specified percentage of data for testing each time the code is run. This results in different values of MSE and R^2 for each run. This is an issue for us because we need to compare the performance of different models with common testing and training sets. To address this issue, we ran the code 100 times for each model and then calculated a 95% confidence interval. Table 3 shows the resulting correlation between each of the parameters.

	$T_{in,j}$	$T_{cpu,j}$	d_j	$T_{out,j}$
$T_{in,j}$	1.00	0.18	0.59	0.92
$T_{cpu,j}$	0.18	1.00	0.10	0.22
d_j	0.59	0.10	1.00	0.56
$T_{out,j}$	0.92	0.22	0.56	1.00

Table 3: The correlation between different parameters of the model

As can be seen in Table 3, among all the parameters, $T_{in,j}$ has the highest correlation with $T_{out,j}$, so we implement two models including only $T_{in,j}$ as input. One of them has a degree of three and the other one has a degree of one. The other three models include all the parameters and cover up to third degree polynomials.

1: Linear Model with $T_{in,j}$ Only

The simplest model would be a linear model dependent only on $T_{in,j}$. Among all the other parameters we chose $T_{in,j}$ because it has the highest correlation with $T_{out,j}$.

$$T_{out,j} = 22.34 + 0.67T_{in,j} \quad (54)$$

2: Model with Degree Three Using Only $T_{in,j}$

A generalization of a linear model would be to consider higher order factors. One way to do that is to consider a higher order polynomial. For our data, the resulting polynomial of degree three is

$$T_{out,j} = 22.43 + 0.6603T_{in,j} - 0.0000354T_{in,j}^2 + 0.00000769T_{in,j}^3 \quad (55)$$

3: Model with Degree One

The following is the polynomial model of degree one using all the three

parameters $T_{in,j}$, $T_{out,j}$ and d_j .

$$T_{out,j} = 20.55 + 0.658T_{in,j} + 0.029T_{cpu,j} + 0.1247d_j \quad (56)$$

4: Model with Degree Two

The following is the polynomial model of degree two using all the three parameters $T_{in,j}$, $T_{out,j}$ and d_j .

$$T_{out,j} = 23.21 + 0.1619T_{in,j} + 0.0622T_{cpu,j} + 6.68d_j + 0.0047T_{in,j}^2 + 0.00314T_{in,j}T_{cpu,j} - 0.0235T_{in,j}d_j - 0.000694T_{cpu,j}^2 - 0.0224T_{cpu,j}d_j - 1.572d_j^2$$

5: Model with Degree Three

The following is the most general model of degree three using all the three parameters $T_{in,j}$, $T_{out,j}$ and d_j .

$$T_{out,j} = 34.73 + 0.925T_{in,j} - 0.74T_{cpu,j} + 2.23d_j - 0.0127T_{in,j}^2 - 0.00458T_{in,j}T_{cpu,j} + 0.055T_{in,j}d_j + 0.0135T_{cpu,j}^2 - 0.163T_{cpu,j}d_j + 5.67d_j^2 + 0.00026T_{in,j}^3 + 0.00027T_{in,j}^2T_{cpu,j} - 0.015T_{in,j}^2d_j - 0.000106T_{in,j}T_{cpu,j}^2 + 0.000268T_{in,j}T_{cpu,j}d_j + 0.164T_{in,j}d_j^2 - 0.000056T_{cpu,j}^3 + 0.000808T_{cpu,j}^2d_j - 0.00312T_{cpu,j}d_j^2 - 2.66d_j^3$$

Table 4 shows the performance parameters for each of the above mentioned models with a confidence level of 95%. The numbers in this table are the result of running the code 100 times. Also, lb and ub in the table stand for *lower bound* and *upper bound*, respectively.

	$MSE - lb$	$MSE - ub$	$R^2 - lb$	$R^2 - ub$
Degree 3	1.18	1.33	0.80	0.81
Degree 2	1.21	1.36	0.76	0.77
Degree 1	1.74	1.91	0.77	0.78
Degree 3 only T_{in}	1.64	1.83	0.76	0.77
Degree 1 only T_{in}	1.64	1.83	0.77	0.78

Table 4: The performance parameters for each of the models

Two factors need to be considered when choosing a suitable model

- The computation overhead (complexity) of the model
- Performance parameters of the model (lower MSE and higher R^2)

Based on Table 4, the model with degree three has a low error but it has much higher complexity compared to other models; in particular models with degree one. On the other hand, the error magnitude among the different models is not very different, so the models with degree one seem to be a wiser choice for $T_{out,j}$.

While the conclusions are that in our case choosing a simple function is appropriate, one important question is whether any of these choices of model will change the optimal workload distribution. In the next section, we show that using any of these models for $T_{out,j}$ will not affect the conclusion we made in Chapter 5, in other words it will always fall into one of the categories discussed in Chapter 5.

6.4 Proof of Integrity

In the previous section, we calculated five different models for $T_{out,j}$. In this chapter, we prove that while these models differ in complexity and performance, when plugged into the optimization problem they will not affect the convexity of the power function used. This proves that our conclusion regarding workload distribution based on the convexity of power model works for all the cases. We would expect this to hold in much greater generality. It seems that any “reasonable” fit to the data should maintain convexity, but stating a precise version (and consequently solving it) is beyond the scope of the thesis.

The main idea is to use the models in Section 6.3 one by one to replace (14) in the optimization problem and then check the convexity of the power function. We expect the power function to remain convex in all cases. After doing this

for all five models, we realized using different models for $T_{out,j}$ did not affect the power function convexity. In other words, when we plug a model other than (14) for $T_{out,j}$ in our optimization problem, the power function is still convex. This shows that as long as the power function is convex, the balanced solution minimizes the power consumption of cooling units, as proved in Theorem 1 in Section 5.1. Figure 43 shows the plot for power function when (56) in Section 6.3 is used.

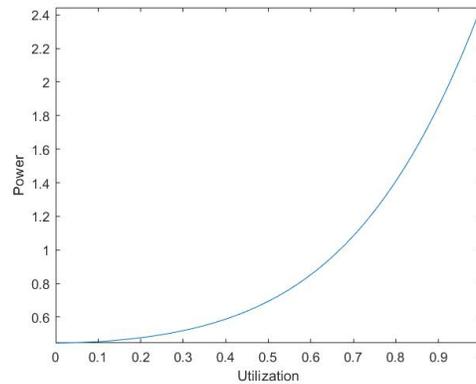


Figure 43: Changes in power when the model in Section 6.3 is used for $T_{out,j}$. As can be seen, it does not affect the convexity of power function.

7 Conclusion

Data centres consume a huge amount of power and cooling is a major contributing factor to total electricity cost. In this research an algorithm has been presented to minimize power consumption of cooling units while keeping all the server cores below a temperature threshold. We found that in a data centre consisting of racks of servers with in-row cooling units with different setpoints at each end, there exists two zones with some overlap. One of the zones has servers with higher inlet temperatures and is referred to as the hot zone and the other one is referred to as the cold zone. The size of zones shows no drastic changes with modifying server utilizations or the airflow provided by cooling units. Next, an optimization problem has been formulated and solved in order to optimize server utilizations, setpoint temperature of cooling units and size of each zone. The results show that in case of a convex power function, distributing workload non-uniformly between the zones does not seem to decrease power consumption of cooling units. It is optimal to choose the setpoints to be equal and distribute workload uniformly regardless of zones. We then found the optimal workload distribution for different groups of power models based on their shape.

In the second part of this thesis, we found a data-driven model for server outlet temperature. To do this, we gathered experimental data measuring servers' inlet and outlet temperatures as well as other operational parameters such as server utilizations and distance of each server from the cooling unit. We then applied machine learning techniques to find a model with a reasonable tradeoff between complexity and error. Finally, we demonstrated that for any of these models, our conclusions about workload distribution remained unchanged.

7.1 Future Work

This research has accomplished important goals about energy efficiency in a modular data centre, yet there are significant opportunities for future research. Extensions to this work can be made by addressing the assumptions of Chapters 1 and 3. Some possible extensions are as follows:

- In the current work we assumed servers are homogeneous. One can modify the optimization problem to include m different types of servers with different power consumptions and computational power.
- Future work can take power consumption of servers into account as well as power consumption of cooling units. It is briefly mentioned in Chapter 4 that if the power consumption of servers is defined based on the entire workload (U) then IT power consumption is invariant to the actual workload distribution. In order to be able to consider IT power as well, we need to first define the IT power based on utilization of each server (u_j). Different servers may have different power consumptions and placing jobs on servers with higher computational power while maintaining optimal power consumption is a trade off that is worth exploring.
- In addition to cooling units, servers are the other component of a data centre that consume a large amount of power. By trying to place the workload on a minimum number of servers and shutting down the rest, we can save power. This technique is referred to as *server consolidation* and is mentioned in numerous previous researches, but it is not clear how this changes the thermal properties of the system. In other words, it is not clear if shutting down some of the servers affects the zonal model.

The items above can provide a starting point for discussion and further research.

References

- [1] S.M. Mirhoseininejad, G. Badawy, and D.G. Down. EAWA: Energy-aware workload assignment in data centers. *The 2018 International Conference on High Performance Computing and Simulation*, 8(5):811–817, 2018.
- [2] C.K. Carmel, G. Shrimali, and A. Albert. Disaggregation: the holy grail of energy efficiency. *Energy Policy*, 52:213–234, 2013.
- [3] M. Deru, K. Field, D. Studer, K. Benne, B. Griffith, P. Torcellini, B. Liu, M. Halverson, D. Winiarski, and M. Rosenberg. *US Department of Energy commercial reference building models of the national building stock*, 2011.
- [4] J. Judge, J. Pouchet, A. Ekbote, and S. Dixit. Reducing data center energy consumption. *ASHRAE Journal*, 50(11):14, 2008.
- [5] A. Radmehr, R.R. Schmidt, K.C. Karki, and S.V. Patankar. Distributed leakage flow in raised-floor data centers. In *ASME 2005 Pacific Rim Technical Conference and Exhibition on Integration and Packaging of MEMS, NEMS, and Electronic Systems*, pages 401–408. American Society of Mechanical Engineers, 2005.
- [6] N. El-Sayed, I.A. Stefanovici, G. Amvrosiadis, A.A. Hwang, and B. Schroeder. Temperature management in data centers: why some (might) like it hot. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):163–174, 2012.
- [7] T. Chantem, X.S. Hu, and R.P. Dick. Online work maximization under a peak temperature constraint. In *Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design*, pages 105–110. ACM, 2009.

- [8] H. Hanson, S.W. Keckler, S Ghiasi, K Rajamani, F Rawson, and J Rubio. Thermal response to DVFS: Analysis with an Intel Pentium M. In *Proceedings of the 2007 international symposium on Low power electronics and design (ISLPED'07)*, pages 219–224. IEEE, 2007.
- [9] D. Meisner, C.M. Sadler, L. André Barroso, W.D. Weber, and T.F. Wenisch. Power management of online data-intensive services. In *ACM SIGARCH Computer Architecture News*, volume 39, pages 319–330. ACM, 2011.
- [10] M. Lin, A. Wierman, L.H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking (TON)*, 21(5):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*, pages 15–22. ACM, 2010.
- [12] 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*, 9(1):42–49, 2005.
- [13] X. Zhang, T. Lindberg, K. Svensson, V. Vyatkin, and A. Mousavi. Power consumption modeling of data center IT room with distributed air flow. *International Journal of Modeling and Optimization*, 6(1):33–38, 2016.
- [14] H.F. Hamann, V. López, and A. Stepanchuk. Thermal zones for more efficient data center energy management. In *2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, pages 1–6. IEEE, 2010.

- [15] Z. Song, B. T. Murray, and B. Sammakia. A dynamic compact thermal model for data center analysis and control using the zonal method and artificial neural networks. *62(1):48–57*, 2014.
- [16] C.B. Bash, C.D. Patel, and R.K. Sharma. Dynamic thermal management of air cooled data centers. In *Thermal and Thermomechanical Phenomena in Electronics Systems, 2006. IThERM'06. The Tenth Intersociety Conference on*, pages 8–pp. IEEE, 2006.
- [17] 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*, volume 138, page 140, 2007.
- [18] H. Sano, K. and+ Shimizu, Y. Kondo, and T. Fujimoto. Improving accuracy of temperature distribution and energy-saving technology of air conditioners in data centers. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 8(5):811–817, 2018.
- [19] R. Rahmani, I. Moser, and M. Seyedmahmoudian. A complete model for modular simulation of data centre power load. *arXiv preprint arXiv:1804.00703*, 2018.
- [20] 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*, pages 61–75, 2005.
- [21] J. Phung, Y.C. Lee, and A.Y. Zomaya. Application-agnostic power monitoring in virtualized environments. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 335–344. IEEE, 2017.

- [22] R. Lent. Evaluating the cooling and computing energy demand of a data centre with optimal server provisioning. *Future Generation Computer Systems*, 57:1–12, 2016.
- [23] M.A. Oxley, E. Jonardi, S. Pasricha, A.A. Maciejewski, H.J. Siegel, P.J. Burns, and G.A. Koenig. Rate-based thermal, power, and co-location aware resource management for heterogeneous data centers. *Journal of Parallel and Distributed Computing*, 112:126–139, 2018.
- [24] E. Pakbaznia and M. Pedram. Minimizing data center cooling and server power costs. In *Proceedings of the 2009 ACM/IEEE international symposium on Low power electronics and design*, pages 145–150. ACM, 2009.
- [25] 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*, 29(6):1317–1331, 2018.