Temperature Balancing among Servers with Workload Distribution

THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Penghui Zhang

Graduate Program in Electrical and Computer Science

The Ohio State University

2016

Master's Examination Committee:

Dr. Xiaorui Wang Advisor

Dr. Ümit V. Çatalyürek

Abstract

Power and energy consumption of datacenters has become a serious concern in recent years, because of their contribution to cost, design and reliability constraints, and environmental concerns. According to a report from EPA, the total energy consumption from datacenters in the United States was over 100 billion kWh in 2011, and cooling part has taken the second place, which is 29% in the power usage of a typical data center. In order to decrease the power consumption of a datacenter, it is preferable to balance the temperatures of each server as close as possible. For its sake, in this paper we need to balance the temperatures of the servers in a datacenter to make sure that the power consumed by the cooling system can stay in a relatively low level. In this case, we do not turn on the cooling system until the temperatures for all the servers are higher than the threshold.

Acknowledgments

I would never have been able to finish my thesis without the guidance of my committee members, Dr. Xiaorui Wang and Dr. Ümit V. Çatalyürek, help from friends and colleagues, and support from my family.

I would like to express my gratitude to my advisor, Dr. Wang, for his guidance and patience, and providing me with an atmosphere for doing research.

I would like to thank my instructors, Dr. Yuanfang Zheng, Dr. Füsun Özgüner, and Dr. Joanne DeGroat etc., for their guidance in various details of academic.

I would like to thank my friends and colleagues who were always encouraging me and providing necessary help.

I would like to thank those authors from the references who answered my questions about their work.

Finally, I would like to than my families. They were always supporting me and cheering me up with their best wishes.

Vita

2010	Shandong Experimental High School, China
2014	B.S. Electrical Engineering, Beijing Institute of Technology, China
2016	M.S. Electrical and Computer Engineering, Ohio State University

Fields of Study

Major Field: Electrical and Computer Engineering

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Power and energy consumption of datacenters has become a serious concern in recent years, because of their contribution to cost, design and reliability constraints, and environmental concerns. Take a report form EPA (Environmental Protection Agency) as an example, the total energy consumption from datacenters in the United States was over 100 billion kWh in 2011. Server power consumption has become a first order concern for enterprise datacenters. In order to decrease the power consumption of a datacenter, it is preferable to balance the temperatures of each server as close as possible.

Traditionally, the temperatures of the servers in a datacenter are different from each other, which will result in an inefficient way to cool down these servers. For example, there are 7 servers in a datacenter, 5 of which are $35^{\circ}C$ and 2 of which are $47^{\circ}C$, and we suppose the cooling system make the temperature of the data center down linearly. Now the cooling system are required to cool down the datacenter to $30^{\circ}C$. When 5 servers are cooled down to $30^{\circ}C$, the higher ones are $42^{\circ}C$, so that the cooling system needs to keep on working until the temperatures of all the servers are below $30^{\circ}C$ where 2 servers are $30^{\circ}C$, but the other 5 are $18^{\circ}C$, which consumes much powers for cooling.

In order to balance the temperatures of CPU, some recent research has got some ideas. For instance, Essay [6] and [7] came up with CFD (Computational fluid dynamics), which is regarded as a popular and conventional approach to simulate and estimate the temperature evolution within datacenters. Also it is used to distribute the workload over selected servers and racks [8]. Besides, some previous works have focused on simplified thermal modeling of the datacenter using steady-state conditions validated with CFD analysis [9].

These techniques mentioned above are key elements of controlling the temperature of a CPU. However, the first method is very complex and takes a lot of time consuming for most practical scenarios and the shortcoming of the second one is that they use steady-state temperature models, which is not suitable because the temperatures of the servers are changing all the time.

To balance the temperature of the server more precisely, we have to face several new challenges. First, we need to import a dynamic temperature model to describe the real time temperature based on the previous steady state model. After we get the correct temperatures, the second challenge is that how we can redistribute the workload of each server by using the difference among the temperatures of the servers, which means that a controller is our meet to manipulate the allocation of workload. Here comes out our third challenge that how we can make a bridge between the workload and the temperature.

For its sake, in this paper we need to balance the temperatures of the servers in a datacenter to make sure that the power consumed by the cooling system can stay in a relatively low level. To make the whole problem simpler, we are working on the servers in one enclosure of a datacenter. Therefore, in this research, we will take advantage of the workload distribution to make the temperatures of each server into the same number. The steps of my thesis research are listed below:

- We learned the blade servers, temperature model and PID controller;
- We began to set up a problem using the optimization problem prototype;
- We designed a PID controller, which is used to control the workload distribution by detecting the variance of temperatures among the servers;
- We simulated our PID controller on MATLAB, looking for the disadvantages of our design and then fixing them.
- We analyzed the results.

Chapter 2: Related Work

Feedback-based thermal management has been applied at different levels of computer systems. At the architecture level, the authors of [1] employ feedback control to regulate the instruction fetch rate to control the temperature of the processor. Feedback control is also used to manipulate clock gating for mitigating the thermal pressure [2]. Heat-and-Run [10] balances the temperature of different cores in a multicore processor through instruction migration.

At the single-node system level, a predictive dynamic thermal management method [11] has been proposed to regulate the temperature generated by multimedia applications. At the distributed system level, Weatherman [12] adopts a data driven method to derive the heat distribution in data centers and then use this distribution to adjust workload in the data centers.

Some previous works have focused on sensor-based thermal mapping by continually monitoring temperature [14] [15]. On the other hand, workload distribution based on optimized thermal and power models has also been researched regarding temporal job scheduling and spatial task balancing in order to achieve overall energy-efficiency [16] [17] [18].

The work most closed related to this paper is Optimized Thermal-aware Workload Distribution [13]. By integrating thermal models that allow balancing cooling needs with computing needs contributing to reduce overall power consumption, the author presents a workload distribution optimization method for homogeneous server environments that minimizes total heat recirculation. Although essay [13] focuses on the minimization of total heat recirculation instead of balancing the temperature, it gives us a key to define our own problem with constraints.

Chapter 3: Background Knowledge

3.1 Blade Server

In recent years, there has been a rapid growth in the use of blade servers in data centers [4]. Commercially available blade systems include HP C-Class blades, IBM BladeCenter, and Dell PowerEdge blade servers. A survey of 166 data center operators [5] showed that 76% of operators were using blade servers [3] within their data centers, with a further 14% having plans to deploy them in the near future. Figure 1 illustrates an example of a typical blade enclosure. It has a total of sixteen blades in the front, eight on the top and eight on the bottom. The blades are cooled by up to ten fans in the back, five on the top and five on the bottom. The airflow generated by the fans is pulled through the blades towards the back of the enclosure with each fan contributing to the blade-level airflow rate. The enclosure and server architecture allows sharing of the cooling resources among the blades, and provide improved flexibility and configurability of IT resources. However, control of the fans in most cases are heuristic and zone based. Without an optimal fan control design, the over-provisioning of cooling capacity can be exacerbated in this architecture since a single hot spot can often drive the operation of a multiplicity of fans.

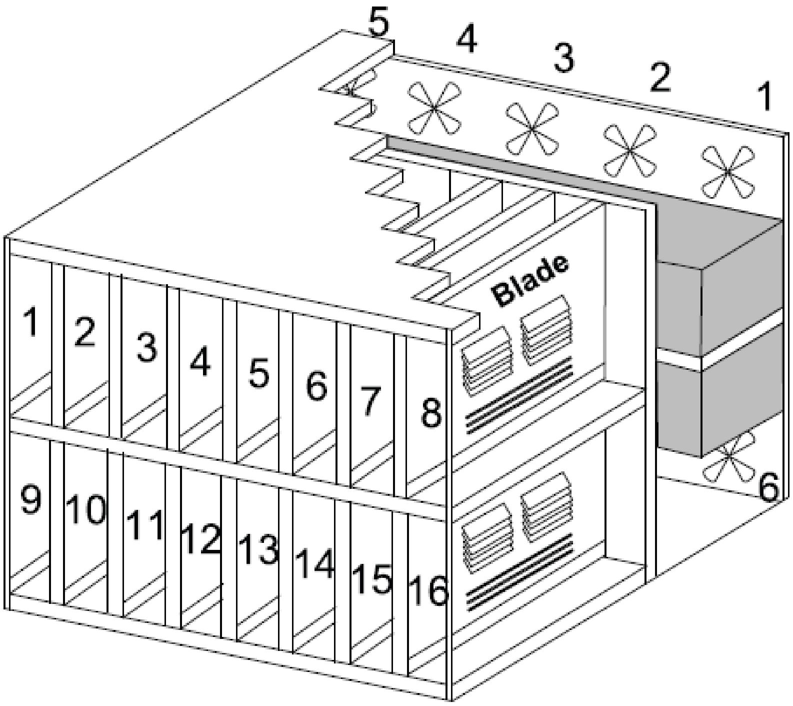

Figure 1. Enclosure

3.2 Temperature Models

3.2.1 Steady State Model

CPU temperature is related to the fan speed, the CPU utilization and the ambient temperature. Now we begin with the thermal resistance R_j, of a blade.

$$R_j = \frac{T_{CPU_j} - T_{amb_j}}{Q_j} \quad (1)$$

Where T_{CPU_j} is the temperature of CPU j, T_{amb_j} is the ambient temperature of CPU j and Q_j is the heat transferred per unit of time between the CPU and the ambient air. We can consider Q_j be the power consumed by the CPU so that we can get an equation of Q_j,

$$Q_j = g_{CPU} \bullet Util_j + P_{CPU,idle} \quad (2)$$

where g_{CPU} is a constant coefficient, $Util_j$ is the CPU utilization of CPU j and $P_{CPU,idle}$ is the power consumption of the server when CPU is idle.

In equation (1), temperature and the heat transferred play an external role of thermal resistance. Internally, R_j depends on the material properties such that we can get the internal equation of it,

$$R_j = \frac{C_3}{\dot{V}_j^{nR}} + C_4 \quad (3)$$

Combine with equation (1), (2) and (3), we can get the steady state temperature model as,

$$T_{CPU_j} = (g_{CPU} \bullet Util_j + P_{CPU,idle}) \bullet R_j + T_{amb_j} \quad \text{for any blade j. (4)}$$

3.2.2 Transient Model

We use the equation following to describe the transient model,

$$C_1 \frac{dT_{CPU_j}}{dt} = \frac{C_2}{R_j}(T_{amb_j} - T_{CPU_j}) + \mathrm{Q}_j \quad (5)$$

Where C_1 and C_2 are constants related to the fluid properties of air, CPU package geometry, and material properties. Equation (10) tells us that the transient temperature is affected by the workload, the environment and the fan speed that is through R.

Now we need discrete-time transient temperature model due the discrete-time tuned fan controller. We assume the temperature is sampled every Δt time, which is called sampling interval, during which the ambient temperature T_{amb_j} and the fan speed are never going to change. So we can get,

$$\Delta T_j\left(k+1\right)=e^{-\frac{\Delta t C_2}{C_1 R_j}}\Delta T_j\left(k\right)+\left(1-e^{-\frac{\Delta t C_2}{C_1 R_j}}\right)\frac{R_j}{C_2}Q_j \quad (6)$$

where $\Delta T_j = T_{CPU_j} - T_{amb_j}$.

3.3 PID Controller

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process through use of a manipulated variable.

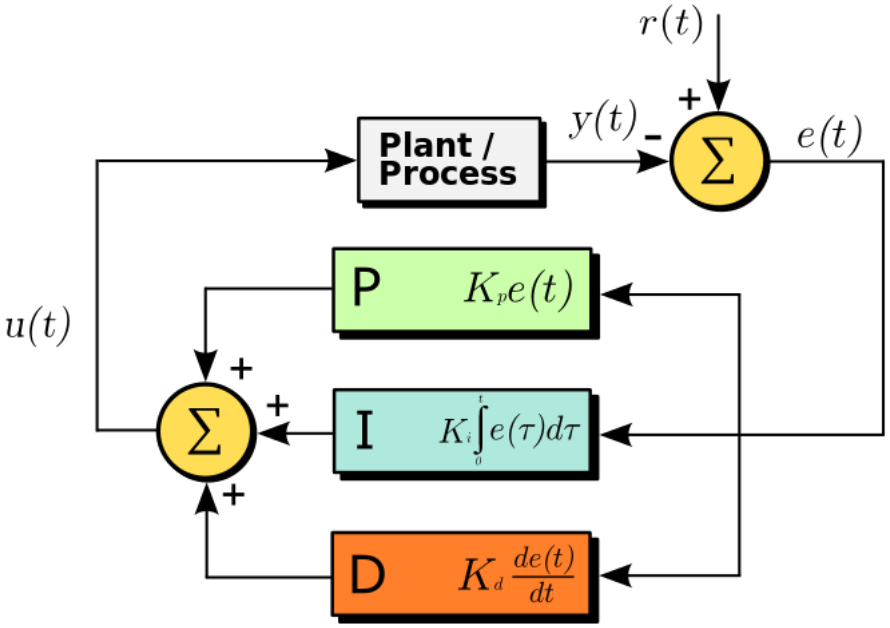

Figure 2. PID Controller

As shown in figure 2, the PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. Simply put, these values can be interpreted in terms of time: P depends on the present error, I on the accumulation of past errors, and D is a prediction of future errors, based on current rate of change. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve, a damper, or the power supplied to a heating element. For a discrete time case, the term PSD, for proportional-summation-derivative, is often used.

The PID control scheme is named after its three correcting terms, whose sum constitutes the manipulated variable (MV). The proportional, integral, and derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output, the final form of the PID algorithm is:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (7)$$

Where K_p is proportional gain, K_i is integral gain, K_d is derivative gain, $e(t)$ is the error that is $e(t) = SP - PV$.

3.4 Ziegler–Nichols method

The Ziegler–Nichols tuning method is a heuristic method of tuning a PID controller. It was developed by John G. Ziegler and Nathaniel B. Nichols. It is performed by setting the I (integral) and D (derivative) gains to zero. The "P" (proportional) gain, K_p is then increased (from zero) until it reaches the ultimate gain K_u , at which the output of the control loop oscillates with a constant amplitude. K_u and the oscillation period T_u are used to set the P, I, and D gains depending on the type of controller used.

Chapter 4: Problem Definition

Our goal of adjusting the temperature for the blade servers is to minimize the variance of all the temperatures of the servers to zero while maintaining the power consumption, server temperature and the sum of the workload below their defined threshold. Assume that the enclosure has N blade servers, then our goal is to,

$$\min\left\{\mathrm{var}\left[T\left(k\right)\right]\right\} \quad (8)$$

where $\mathrm{var}\left[T\left(k\right)\right]$ is the variance of temperatures of N servers in k^{th} control period.

Subject to,

$$P_{cooling} + \sum_{i=1}^{N} P_i \leq P_{\max} \quad (9)$$

$$T_i\left(k+1\right) \leq T_{th} \quad (10)$$

$$\sum_{i=1}^{N} W_i = W_{total} \quad (11)$$

where P_i is the working power of server i, $P_{cooling}$ is the fan power, $P_{\max}$ is the maximum power of a server; T_{th} is the threshold of temperature of a server, which is $65^{\circ}C$ in general; W_i is the workload of server i and W_{total} is the total workload.

The variance minimized function in equation (8) along with the constraints defined by equation (8), (9) and (10) formulates a well-defined optimization problem. The time-varying and sometimes unpredictable nature of application demands require that the optimization problem be solved at runtime. However, to solve the optimization problem, the controller needs models to determine the impact of workload changes on the objective function and the constraints. For instance, models are required to correlate workload distribution with the variance of temperature.

Chapter 5: Controller Design

Here we use PID controller to have workload distribution and variance of temperature related with each other. Figure (3) shows the basic concept of our PID controller.

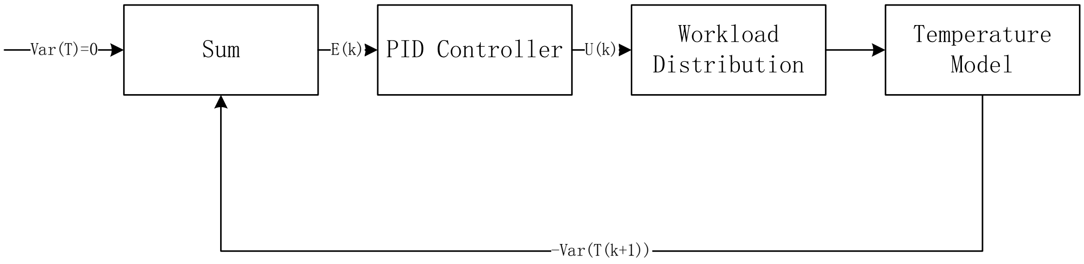

Figure 3. PID Controller Design

As we see in figure (3), the setpoint of PID controller is $\mathrm{var(T)} = 0$. After setpoint adds $-\mathrm{var(T(k+1))}$, which is variance of the temperature in the next control period, we can get the error, $e(k)$. Then the PID controller use equation (7) that can derive $u(k)$ by calculating $e(k)$ with the proportional, integral and derivative term. $u(k)$ is used to control the allocation of the workload, which we will explain in detail. After re-distributing the workload, the temperature model will calculate the temperatures of each server according to the transient temperature model, followed by the new variance of the temperature, which is utilized by the next control period. The control period will keep on working until $-\mathrm{var}(\mathrm{T}(\mathrm{k}+1)) = 0$ or the controller runs exact periods that we set before the program executes.

5.1 Explanation of the Relationship between Controller Output and Workload Allocation

Now we begin to talk about how controller output $u(k)$ affect the workload distribution. Before we start to explain it, it is necessary for us to set the default value of many parameters: the number of blade servers in an enclosure, N ; the ambient temperatures of blade servers, $T_{amb}(j)$, which are constant for each server; the utilization of each server, $Util(1, j)$, where "1" means the first control period, and "j" is server j, the sum of which will keep $50\% \times N$.

Here we need to take advantage of the temperature model to calculate the temperature of the server in every control period. When the controller start to control period 1, the temperature of each server remains steady so that we use steady state model to get $T(1, j)$.

By using equation (4) we can get,

$$T(1, j) = (g_{CPU} \bullet Util(1, j) + P_{CPU,idle}) \bullet R_j + T_{amb_j} \text{ for server j (12)}$$

After control period 1, we cannot use the steady state model because the temperature is changing all the way. In order to express the correct temperature of each server, we need to use the transient temperature model. From equation (6) we find that,

$$\Delta T(k+1, j) = e^{-\frac{\Delta t C_2}{C_1 R_j}} \Delta T(k, j) + \left(1 - e^{-\frac{\Delta t C_2}{C_1 R_j}}\right) \frac{R_j}{C_2} Q(k, j) \quad (13)$$

where $\Delta T(i, j) = T(i, j) - T_{amb}(j)$

In order to make it clear, we rewrite equation (13),

$$\Delta T(k+1, j) = A \bullet \Delta T(k, j) + B \bullet Q(k, j) \quad (14)$$

where $A = e^{-\frac{\Delta t C_2}{C_1 R_j}}$ and $B = \left(1 - e^{-\frac{\Delta t C_2}{C_1 R_j}}\right)\frac{R_j}{C_2}$.

Our goal is to make the variance of the temperature be zero. So if we make the next period temperature of each server equal to the average temperature that is called T_{avg}, then we can have the variance down to zero. To achieve this goal, if we let $T(k+1, j) = T_{avg}$, then we can get $Q(k+1, j)$, which mean the power of CPU in the next control period,

$$T_{avg} - T_{amb}(j) = A \bullet \Delta T(k, j) + B \bullet Q(k+1, j) \quad (15)$$

From equation (15), if we put $T_{amb}(j)$ to the other side, then we can find,

$$T_{avg} = A \bullet \Delta T(k, j) + B \bullet Q(k+1, j) + T_{amb}(j) \quad (16)$$

Minus $T(k, j)$ from both sides of equation (16), we can continue deducting equation (16) as following,

$$T_{avg} - T(k, j) = A \bullet \Delta T(k, j) + B \bullet Q(k+1, j) + T_{amb}(j) - T(k, j) \quad (17)$$

Because $\Delta T(k, j) = T(k, j) - T_{amb}(j)$, we can go on our deduction of equation (17) to,

$$T_{avg} - T(k, j) = (A-1) \bullet \Delta T(k, j) + B \bullet Q(k+1, j) \quad (18)$$

Let's denote $T_{avg} - T(k, j) = delta\_T$. Combine it and equation (18), we can get the equation that express $Q(k+1, j)$ as,

$$Q(k+1, j) = \frac{delta\_T - (A-1) \bullet \Delta T(k, j)}{B} \quad (19)$$

In order to get the difference between $Util(k+1, j)$ and $Util(k, j)$, which is called $\Delta Util(j)$, we need to calculate $Q(k+1, j)$ and $Q(k, j)$ like following,

$$Q(k+1, j) - Q(k, j) = g_{CPU} \bullet (Util(k+1, j) - Util(k, j)) \quad (20)$$

so that,

$$\Delta Util(j) = \frac{Q(k+1, j) - Q(k, j)}{g_{CPU}} \quad (21)$$

Since we have got $\Delta Util(j)$, we need to add the effect of $u(k)$ to it. We can find the relationship between $u(k)$ and $delta\_T$, which is $(T_{avg} - T(k, j)) \bullet \left(1 + \frac{u(k)}{U}\right)$, where U is a constant parameter and we use the sign of $delta\_T$ to control whether we need to increase the utilization or decrease it. So we update equation (19) as,

$$Q(k+1, j) = \frac{delta\_T \bullet \left(1 + \frac{u(k)}{U}\right) - (A-1) \bullet \Delta T(k, j)}{B} \quad (22)$$

From equation (21) and (22) we can get $\Delta Util(j)$ as,

$$\Delta Util(j) = \frac{\dfrac{delta\_T \bullet \left(1 + \dfrac{u(k)}{U}\right) - (A-1) \bullet \Delta T(k, j)}{B} - Q(k, j)}{g_{CPU}} \quad (23)$$

After we get $\Delta Util(j)$, it is easy for the controller to get $Util(k+1, j)$ and to distribute the workload using $Util(k+1, j)$.

Chapter 6: Simulation

From study [optimal fan.pdf], we have got the values of parameters used in our experiment because we did not simulate our PID controller in the real testbench. Instead, we simulated it in the way of MATLAB.

Now let's define some parameters in our experiment.

- C_1 and C_2 are equal to 1
- R_j is a constant number, which is equal to 0.22.
- g_{CPU} is a constant coefficient, equal to 150.
- $P_{CPU,idle}$ is equal to 60W.
- The initial value of utilization is Util(1,:)=[0.4,0.2,0.3,0.5,0.6,0.8,0.7].
- The value of the ambient temperature is T_amb=[20,21,22,23,24,25,22].

Now let's look at the MATLAB script and I will show the details about my program. In the first part of my MATLAB program, I set the default values for each parameter.

After all the initial values have been settled down, we need to calculate the steady state temperature of each server by using equation (4).

Then, it is time for us to adjust the temperature by increasing or decreasing workload for each server. Assume that we need to spend 100 control period.

After we ran this program, we can get some curves to express the results of it.

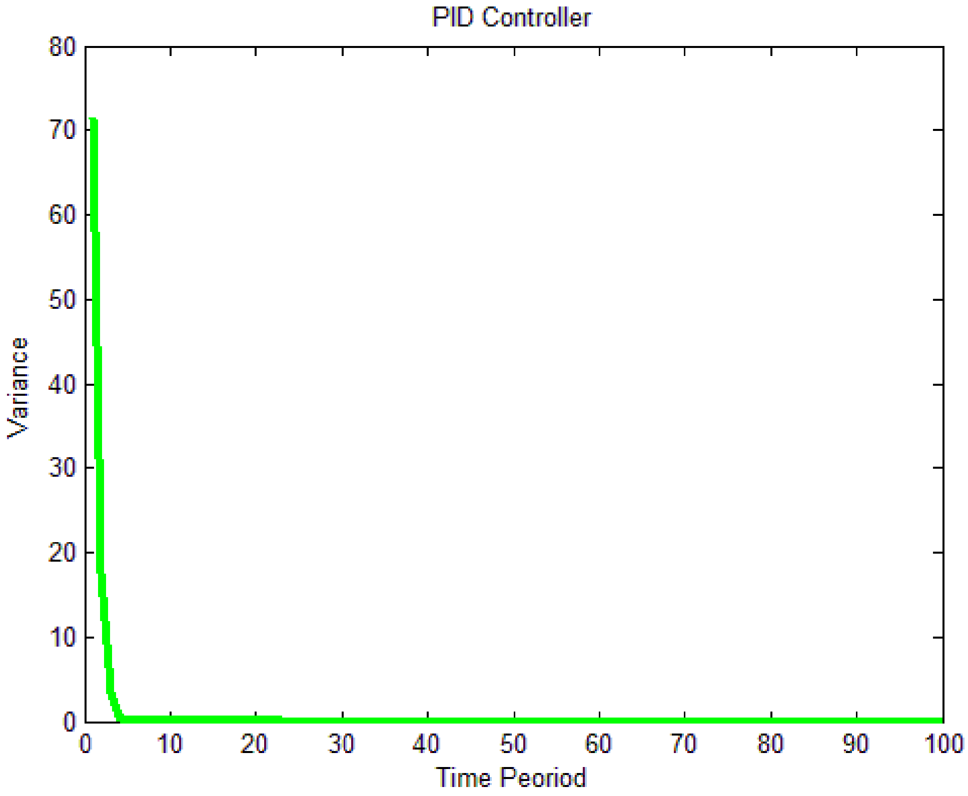

Figure 4. Result of PID Controller

Figure 4 is the curve of PID controller, which shows to us that before the 10th control period, the variance of the temperature has been close to zero, which is quite fast.

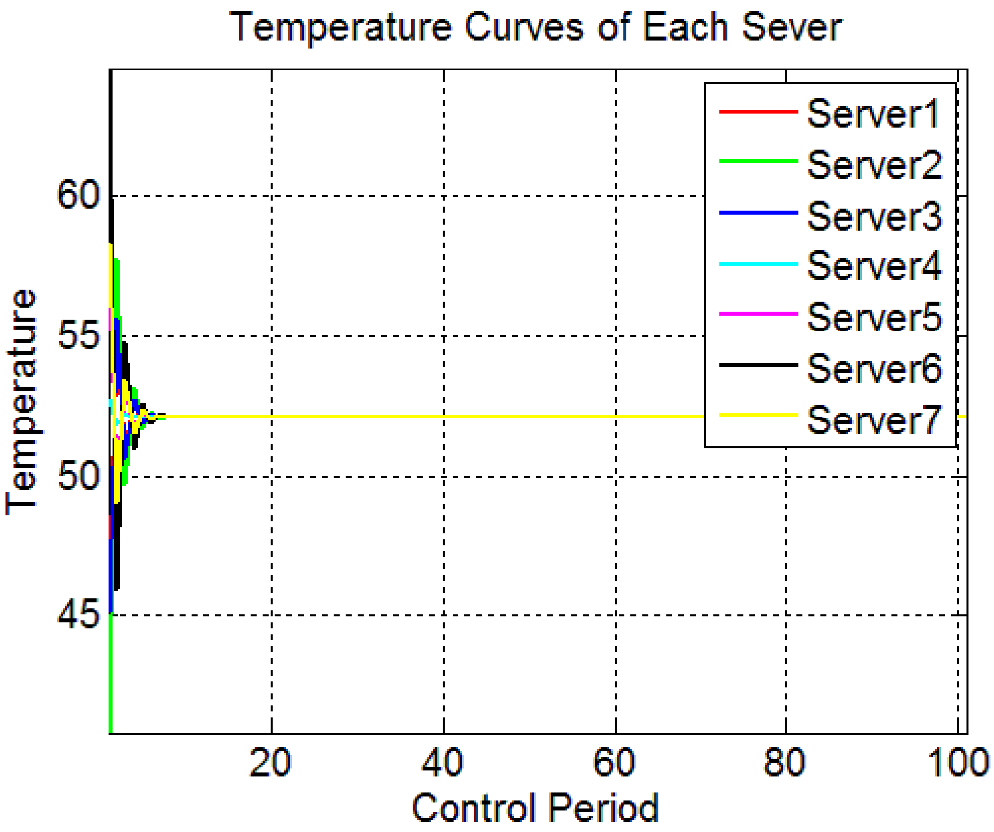

Figure 5. Temperature Curves of Each Server

Figure (5) gives us the information from each server's perspective that all the temperatures converge to one point, which is approximately $52.12857^{\circ}C$.

Server No. Period No.	1	2	3	4	5	6	7	sum of util
1	0.4	0.2	0.3	0.5	0.6	0.8	0.7	3.5
2	0.662973	0.720043	0.62265	0.473768	0.376375	0.227494	0.416698	3.5
3	0.534067	0.465125	0.464491	0.486627	0.485993	0.508128	0.555569	3.5
4	0.590802	0.577321	0.534101	0.480967	0.437747	0.384614	0.494448	3.5
5	0.566105	0.528483	0.5038	0.483431	0.458748	0.438379	0.521054	3.5
6	0.576851	0.549733	0.516984	0.482359	0.44961	0.414985	0.509477	3.5
7	0.572175	0.540487	0.511248	0.482825	0.453586	0.425164	0.514514	3.5
8	0.57421	0.54451	0.513744	0.482622	0.451856	0.420735	0.512322	3.5
9	0.573325	0.542759	0.512658	0.482711	0.452609	0.422662	0.513276	3.5
10	0.57371	0.543521	0.51313	0.482672	0.452282	0.421824	0.512861	3.5
11	0.573542	0.54319	0.512925	0.482689	0.452424	0.422189	0.513042	3.5
12	0.573615	0.543334	0.513014	0.482682	0.452362	0.42203	0.512963	3.5
13	0.573583	0.543271	0.512975	0.482685	0.452389	0.422099	0.512997	3.5
14	0.573597	0.543298	0.512992	0.482684	0.452377	0.422069	0.512983	3.5
15	0.573591	0.543286	0.512985	0.482684	0.452383	0.422082	0.512989	3.5
16	0.573594	0.543292	0.512988	0.482684	0.45238	0.422076	0.512986	3.5
17	0.573593	0.543289	0.512987	0.482684	0.452381	0.422079	0.512987	3.5
18	0.573593	0.54329	0.512987	0.482684	0.452381	0.422078	0.512987	3.5
19	0.573593	0.54329	0.512987	0.482684	0.452381	0.422078	0.512987	3.5
20	0.573593	0.54329	0.512987	0.482684	0.452381	0.422078	0.512987	3.5

Table 1. Utilization

Table (1) shows the utilization of server 1 to server 7 from period 1 to period 20.

Server No. Period No.	1	2	3	4	5	6	7
1	46.4	40.8	45.1	52.7	57	64.6	58.3
2	54.98598	57.77925	55.63441	51.84354	49.69871	45.90784	49.05027
3	50.8684	49.6365	50.58242	52.25427	53.20019	54.87205	53.48617
4	52.67705	53.21322	52.80152	52.07386	51.66216	50.9345	51.53769
5	51.88992	51.65662	51.83576	52.15238	52.33151	52.64813	52.38567
6	52.23241	52.33392	52.25598	52.11821	52.04027	51.9025	52.0167
7	52.08339	52.03922	52.07313	52.13308	52.16699	52.22694	52.17725
8	52.14823	52.16745	52.15269	52.12661	52.11185	52.08577	52.10739
9	52.12002	52.11165	52.11808	52.12942	52.13585	52.1472	52.13779
10	52.13229	52.13593	52.13314	52.1282	52.12541	52.12047	52.12456
11	52.12695	52.12537	52.12658	52.12873	52.12995	52.1321	52.13032
12	52.12928	52.12997	52.12944	52.1285	52.12797	52.12704	52.12781
13	52.12826	52.12797	52.1282	52.1286	52.12883	52.12924	52.1289
14	52.1287	52.12884	52.12874	52.12856	52.12846	52.12828	52.12843
15	52.12851	52.12846	52.1285	52.12858	52.12862	52.1287	52.12863
16	52.1286	52.12862	52.1286	52.12857	52.12855	52.12852	52.12854
17	52.12856	52.12855	52.12856	52.12857	52.12858	52.1286	52.12858
18	52.12858	52.12858	52.12858	52.12857	52.12857	52.12856	52.12857
19	52.12857	52.12857	52.12857	52.12857	52.12857	52.12858	52.12857
20	52.12857	52.12857	52.12857	52.12857	52.12857	52.12857	52.12857

Table 2. Temperature

Table (2) gives us the data of temperatures for each server from period 1 to 20.

Combine with table (1) and table (2) we can see that from period No.1 to No.2, the utilization of the server that has a lower temperature in period No.1 will increase and that of the server with a higher temperature in period No.1 will decrease, which meets our goal to balance the temperature.

As for the Error of the program, we can see from the data of “Error”, finding that in the period No.100, Error= 4.20725816117873e-29, which is so small that we can see it as zero.

Chapter 7: Conclusion

Datacenters attempt to maximize return on investment by getting down the power consumption. This program is to decrease the cooling power by balancing the servers' temperature.

In conclusion, this thesis paper has introduced a method to balance temperature for the blade servers in an enclosure. For me, this project is the best one I have ever done before. I learnt to use fundamental concepts from heat transfer theory, to develop temperature models in both steady state and transient ways and how to design a PID controller. After I have got enough knowledge, I began to design a new PID controller which is made to relate with the allocation of the workload. Then I started to write a MATLAB script to simulate my design. And the result showed that my project works well.

I believe that, this kind of research brings me not only the knowledge, but the spirit of insisting and the interest to doing research in the future as well.

References

1. K. Skadron, T. F. Abdelzaher, and M. R. Stan, “Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management,” in HPCA, 2002, pp. 17–28.

2. D. Brooks and M. Martonosi, “Dynamic thermal management for high-performance microprocessors,” in Proceedings Of The Seventh International Symposium On High-Performance Computer Architecture, Jan. 2001.

3. Zhikui Wang, Cullen Bash, Niraj Tolia, Manish Marwah, Xiaoyun Zhu†, Parthasarathy Ranganathan, “OPTIMAL FAN SPEED CONTROL FOR THERMAL MANAGEMENT OF SERVERS”, Proceedings of the ASME/Pacific Rim Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Systems, MEMS, and NEMS InterPACK ’09 July 19-23, 2009

4. IDC, 2008. Worldwide quarterly server tracker, Q4 2007, Feb.

5. Data Center Users’ Group, 2008. Spring 2008 Data center users’ group survey results. http://datacenterug.org/.

6. L. Marshall, P. Bemis, “Using CFD for Data Center Design andAnalysis ,” Applied Math Modeling White Paper, January 2011

7. C.D.Patel, R.Sharma, C.E.Bash, A.Beitelmal, “Thermal Considerations in Cooling Large Scale High Compute Density Data Centers,” in Proceedings of the Eighth Inter-Society Conference on Thermal and Thermomechaincal Phenomena in Electronic System(ITherm), San Diego, CA, June 2002.

8. R.K. Sharma, C.E.Bash, C.D.Patel, R.J.Friedrich, J.S.Chase, “Balance of Power :Dynamic Thermal Management for Internet Data Centers,” in Proceedings of IEEE Internet Computing, vol.9 issue 1, January 2005.

9. Q. Tang, T. Mukherjee, S. K. S. Gupta, and P. Cayton, “Sensor based fast thermal evaluation model for energy efficient high-performance datacenters,” in International Conf. on Intelligent Sensing Info. Proc. (ICISIP2006), December 2006.

10. M. Gomaa, M. D. Powell, and T. N. Vijaykumar, “Heat-and-run: Leveraging SMT and CMP to manage power density through the operating system,” SIGOPS Operating System Review, vol. 38, no. 5, pp. 260–270, 2004.

11. J. Srinivasan and S. V. Adve, “Predictive dynamic thermal management for multimedia applications,” in Proceedings of the Seventeenth Annual International Conference on Supercomputing (ICS’03), Jun. 2003.

12. J. Moore, J. Chase, and P. Ranganathan, “Weatherman: Automated, online, and predictive thermal mapping and management for data centers,” in the Third IEEE International Conference on Autonomic Computing, Jun. 2006.

13. Hassan Shamalizadeh, Luis Almeida, Shuai Wan, Paulo Amaral, Senbo Fu, Shashi Prabh, “Optimized thermal-aware workload distribution considering allocation constraints in data centers”, 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing.

14. H.Hamann , J. Lacey , M.O’Boyle, R.Schmidt, and M.Lengar, “Rapid Three-Dimensional Thermal Characterization of Large-Scale Computing Facilities,” in IEEE Transactions on Components and Packaging Technologies, 2009.

15. B.Weiss, H.L.Truong, W.Schott, T.Scherer, C.Lombriser, and P. Chevillat, “Wireless sensor network for continuous monitoring temperatures in data centers”, IBM Research Report, RZ 3807, June 2011

16. J.Moore, J.Chase, P. Ranganathan, and R.Sharma. “Making scheduling “cool”: temperature-aware workload placement in data center,” In ATEC, 2005.

17. T. Mukherjee, A. Banerjee, G. V., S. K. S. Gupta, and S. Rungta, “Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers,” Computer Networks, June 2009.

18. V.Petrucci, E.V.Carrera, O.Loques, J.C.B.Leite, D.Mosse, “Optimized management of power and performance for virtualized heterogeneous server clusters,” in 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid (CCGrid 11), Newport Beach, CA, USA, 2011