WINDOWS 10 IOT SYSTEM FOR TEMPERATURE CONTROL IN A DATACENTER

Mihai Neacşu, Valeriu Manuel Ionescu
Dept. of Communications and Computer Science,
University of Pitesti, Romania
valeriu.ionescu@upit.ro

Keywords: datacenter, cooling control, Windows 10 IoT, Raspberry Pi

Abstract: *Many IT leaders such as Microsoft, Google, Amazon or Cisco store the processing devices in datacenters. A datacenter has large power consumption and to this many active components contribute. One important factor is the cooling system. There are multiple ways to optimize the power consumption for the cooling system and one of them is by individually controlling the cooling areas. This paper presents a test system centered on the Raspberry Pi platform that individually controls two fans. The operating system is Windows 10 IoT.*

1. INTRODUCTION

A Data Center is a location used to house communications systems: servers, backup equipment, telecommunications equipment and data storage equipment. It should also include redundant internet connection, redundant power supply, environment control systems (air conditioning, fire extinguishing systems, raised floor for access and flood prevention) and security systems both physical (access control systems, surveillance video, etc) and software against cyber attacks (Firewall).

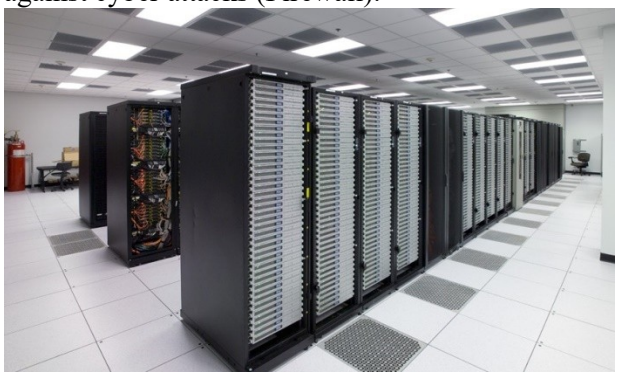

Fig. 1 A common look inside datacenter

Cooling a datacenter is an important aspect related not only to performance improvement but also to data safety and cost control as devices operating in high temperatures have a shorter lifespan and are more prone to develop hardware defects.

These cooling systems are used by large companies: Google, Amazon, Cisco, etc.

For example Google has many datacenters (currently 14) that need power all the time without interruption and keeping the temperature at a constant 26ºC for normal operation. The cooling model is presented in Fig. 2.

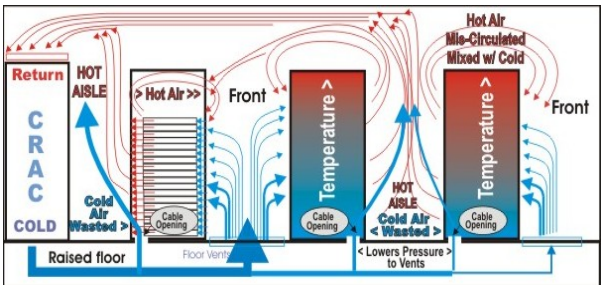

Fig. 2. Cooling model for Google datacenter

Amazon is the largest online commerce company and cloud service provider. It has 46,000 servers and the monthly expenses of the platform are presented in Fig. 3 and show a large percentage toward cooling and power distribution.

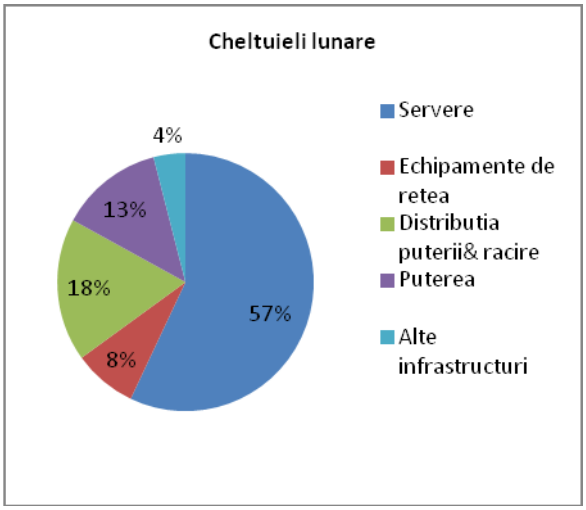

Fig. 3 Monthly expenses of the Amazon platform

Cisco is a multinational corporation has business in the design and implementation of telecommunications services. In 2000, Cisco was the most valuable company in the world with a market capitalization of more than $500 billion in the United States. Today it is still one of the most important companies in the World. It also has datacenters that it maintains. The cooling model used and recommended is presented in Fig. 4.

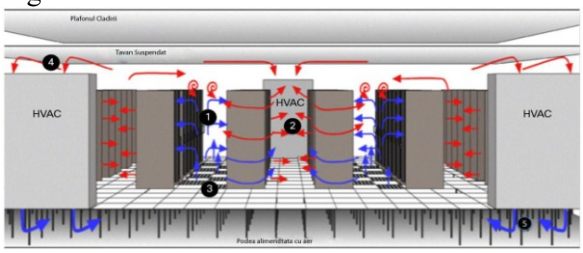

Fig. 4. Cooling model for Cisco datacenter

In most cases the cooling in a datacenter is performed by placing an air conditioning system that floods the room with low temperatures in an attempt to keep the whole location at a low temperature level. The temperature is monitored in one or more locations inside the datacenter.

To control the temperature in a data center is essential to meet maximum availability and efficiency, but must also be controlled in the right places (Fig. 5). The right places represent areas where heat faster systems requiring a rapid cooling and areas where systems are at an ambient temperature.

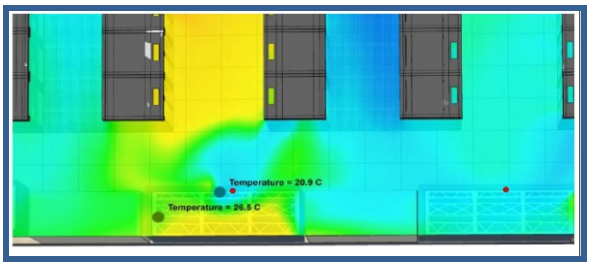

Fig. 5 Temperature distribution inside a datacenter is not uniform

A new model emerged in recent years, where the cooling is performed not in all the datacenter as a whole but in localized areas (enclosures) because the computers in a datacenter are not used similarly. Some computers are used for a while and need cooling while others can even be passively cooled for extended periods of time.

This pattern is especially visible in cloud services, for example the information of CDN usage for the Microsoft Azure [1] platform seen in Fig. 6.

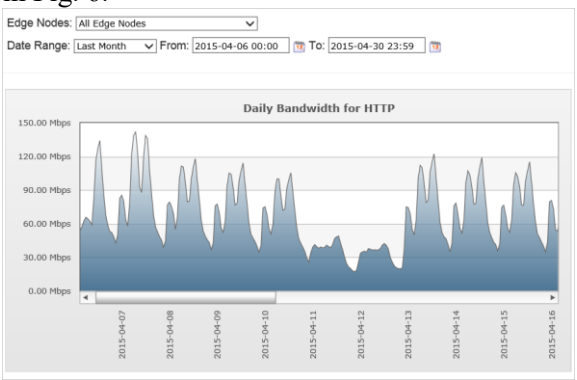

Fig. 6. Usage pattern shows that the system is not operating at peak performance all the time[1]

This new model of cooling [2] is more efficient but has a more complex control mechanism. [3][4] The number of sensors needed is larger and the hardware cooling system must be designed in order to accept localized commands (presented in Fig. 7).

In this paper we will present a prototype cooling system using this principle, based on Windows 10 IoT operating system installed on a Raspberry Pi Model V2 board. This board will run a .Net Core program developed with Visual Studio that controls the cooling of heat sources so that the system remain on a preset temperature preventing overheating and component damage.

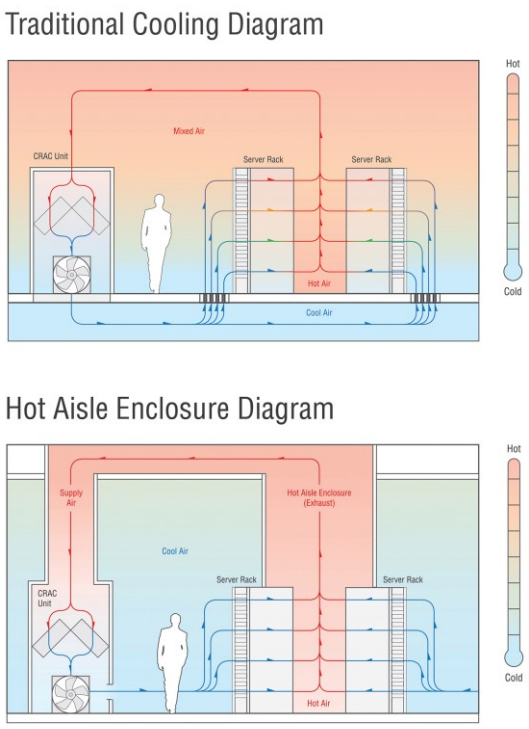

Fig. 7. Traditional cooling vs hot isle cooling [2]

We chose this project because a temperature monitoring in a data center is an important factor. The purpose of this work will be to monitor remotely the usage of temperature cooling system, improve its lifespan and reduce the power consumption.

2. SYSTEM DESIGN

To achieve this we will create a prototype consisting of a Raspberry Pi Model V2 board for control, two coolers, two control circuits, a ADC that accepts commands from the Raspberry Pi and relies them to the control system, two tubes (simulating the enclosures), sources with voltages of 12V, 5V and 3.3V, two temperature sensors, wiring, and two light bulbs which will simulate different heat sources (and load types). The information is read from the sensors and relayed by Raspberry Pi to an internet server so that a log can be monitored regarding the opening and closing of the air cooling system.

The functioning of cooling system is presented below. The converter is a MCP3008 Analog-Digital (A / D) on 10-bits which will connect the Raspberry Pi 2 with the two temperature sensors (LM 35) which will detect the temperature of the heat source. This heat source will be given by two bayonet bulbs with a power of 21 W and 5 W respectively. The choice of two different powers for the light sources will simulate different power consumption patterns (hot spots) in the data center.

The cooling fans will use the supply of a 12 V transformer and will be linked to Pi through a control circuit. The control circuit will be made of an NPN transistor (BD139) which will protect a radiator and a resistance of 10kΩ (Pi protection) and it will give the command by using GPIO pins GPIO5 and GPIO6.

Raspberry Pi is a single board system which is capable of functioning as a computer on its own, requiring only a micro SD card and a power supply. It is ARM based and is capable of running a full operating system such as Linux, or the recently introduced Windows 10 IoT. We have used the latter OS, because it uses C# and the Visual Studio Development Environment that makes debugging very easy. With its help we can communicate with the MCP and use the programming language C # to read data coming from the heat sensors and send data for starting or stopping the fans.

The MCP power will be made via pins 16.VDD and 15.VREF and the heat sensor will be connected to the pin 1(3.3V) of the Raspberry. Reading data via serial terminal will be 13.CLK, 12.DOUT, 11DIN, 10CS that will bind to the Raspberry Pi 2 by pins 23, 21, 19 and 22. For connecting the MCP heat sensors we use pins CH0 and CH1.

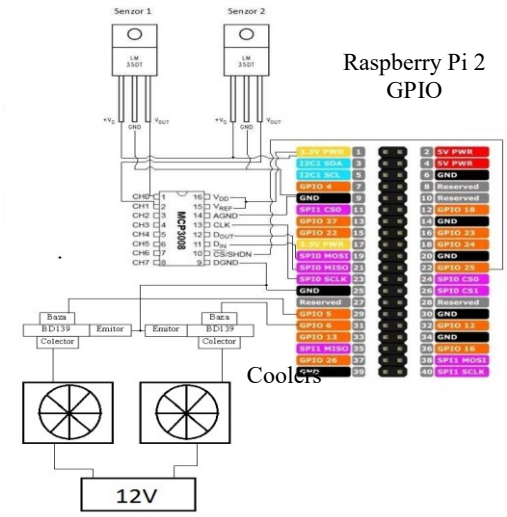

Fig.7.The general system structure

The flow of information is the following: the MCP component reads the information given by temperature sensors and converts from analog to digital. This information is then sent to the Raspberry Pi which gives the command to turn on or off the fans. The command will be given through the pins GPIO5 and GPIO6 (Fig.7.) Commands will be given for starting and stopping the two Ventilating units by pins GPIO5 and GPIO6 of the Raspberry Pi 2. Raspberry Pi GPIO pins GPIO 5 and GPIO 6 will command the two fans. Fans will turn on when a signal from temperature sensors will detect a temperature rise greater than that determined by the new program. A hysteresis was implemented for the temperature limit so that when the values are around the required one, the ventilating units are not turned on and off too frequently. The command and control system design is seen in Fig. 8.

For the software implementation we have based the application on the samples offered by Microsoft [5]

Some of the problems that had to be solved were: the material choice for the enclosure as it had to be transparent and hear resistant; the choice for the source of heat that had to produce enough heat but be safely implemented; the hardware control circuit where the first time we chose a microcontroller L293D but proved to be too expensive for what we needed and a transistor was the final choice. Images of the implemented system are presented in Fig. 9.

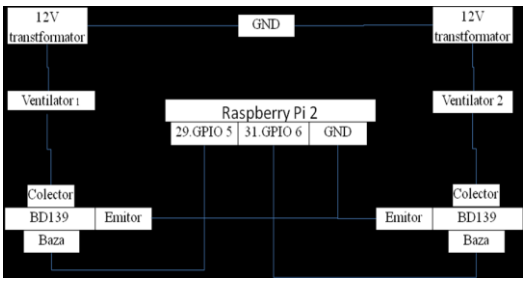

Fig. 8.Command and control system design

3. CONCLUSIONS

This paper presented the successful implementation of a prototype unit that can control independently the cooling of two enclosures. In order to use the prototype for a full server system, correct hardware scaling is necessary. Developing for the Windows IoT was a challenge because of the reduces number of sources.

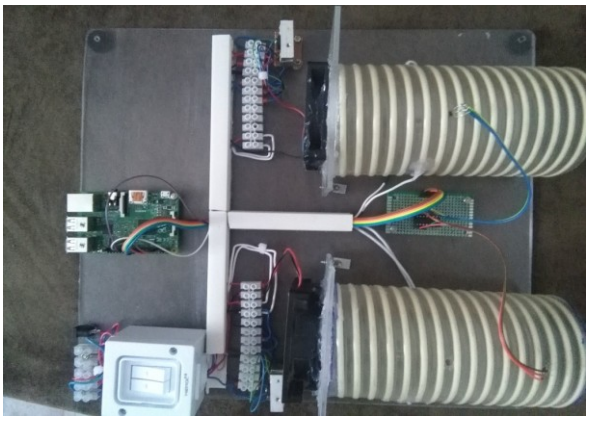

Fig. 9. Implemented prototype images

4. REFERENCES

[1]. Cam Soper, Analyze Azure CDN usage patterns,07/28/2016, Web. Available:https://azure.microsoft.com/en-us/documentation/articles/cdn-analyze-usage-patterns/

[2]. John Sasser A Look at Data Center Cooling Technologies, Available:https://journal.uptimeinstitute.com/a-look-at-data-center-cooling-technologies/

[3]. Cullen E. Bash, Chandrakant D. Patel, Ratnesh K. Sharma “Dynamic Thermal Management of Air Cooled Data Centers”, ITHERM. 2006 0-7803-9524-7/06/$20.00/©2006 IEEE, 448 pp. – 452

[4]. Shrivastava, S., Schmidt, R., Sammakia, B., Iyengar, M., "Comparative Analysis of Different Data Center Airflow Management Configurations", Proceedings of IPACK2005, San Francisco, CA, July 2005

[5]. Microsoft, Universal Windows Platform (UWP) app samples, Web. Available: https://github.com/Microsoft/Windows-universal-samples