CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

Experiments with a Sensing Platform for High Visibility of the Data Center

João Loureiro
Nuno Pereira
Pedro Santos
Eduardo Tovar

CISTER-TR-131206
Version:
Date: 1/1/2014

Experiments with a Sensing Platform for High Visibility of the Data Center

João Loureiro, Nuno Pereira, Pedro Santos, Eduardo Tovar

CISTER Research Unit

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail: joflo@isep.ipp.pt, nap@isep.ipp.pt, pjsol@isep.ipp.pt, emt@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

Data centers are large energy consumers and a substantial portion of this power consumption is due to the control of physical parameters, which bring the need of high efficiency environmental control systems. In this work, we de- scribe a hardware sensing platform specifically tailored to collect physical pa- rameters (temperature, pressure, humidity and power consumption) in large data centers. This platform is an important enabler to find opportunities to optimize energy consumption. We also introduce an analysis of the delay to obtain the sensing data from the sensor network. This analysis provides an insight into the time scales supported by our platform, and also allows to study the delay for different data center topologies. Finally, we exemplify some capabilities of the system with a real deployment.

Experiments With a Sensing Platform for High Visibility of the Data Center

João Loureiro[†], Nuno Pereira, Pedro Santos, Eduardo Tovar

CISTER/INESC-TEC, ISEP, Polytechnic Institute of Porto, Porto, Portugal,
joflo, nap, pjsol, ffp, emt@isep.ipp.pt

Abstract. Data centers are large energy consumers and a substantial portion of this power consumption is due to the control of physical parameters, which bring the need of high efficiency environmental control systems. In this work, we describe a hardware sensing platform specifically tailored to collect physical parameters (temperature, pressure, humidity and power consumption) in large data centers. This platform is an important enabler to find opportunities to optimize energy consumption. We also introduce an analysis of the delay to obtain the sensing data from the sensor network. This analysis provides an insight into the time scales supported by our platform, and also allows to study the delay for different data center topologies. Finally, we exemplify some capabilities of the system with a real deployment.

1 Introduction

Data center's large power consumption justifies a special attention to the design of energy efficient data centers. Power usage effectiveness (PUE) has become the metric to measure data center efficiency. It measures how much of the total energy consumed is really spent on IT work other than on facility's overhead, like lightning, cooling and power distribution, and it is given by: PUE = (IT Equipment Energy + Facility Overhead) / Energy IT Equipment Energy. It is desirable to measure it with a high spatial and temporal granularity, so that the PUE metric is as accurate as possible and to enable better understanding of the power consumption distribution in the data center. This better understanding may lead to great reductions through e.g. better load balancing, power distribution, or reduced air conditioning usage [1].

To have a full picture of the data center environment, it is important to collect air pressure, temperature, humidity and power consumption data at a high granularity (in time and space). The relevance of collecting these parameters is discussed in the next paragraphs.

In a typical data center, IT equipment is organized into rows, with a cold aisle in front, where cold air enters the equipment racks, and a hot aisle in back, where hot air is exhausted. Computer-Room Air Conditioners (CRACs) are commissioned to cycle the air, by pushing the cold air and returning the hot air to be cooled again. The CRAC systems are responsible for a big share of the facility overhead energy, and in

[†] João Loureiro is supported by the government of Brazil through CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico.

order to achieve a more uniform thermal profile, special effort must be given on airflow distribution, by preventing cold and hot air from mixing and by eliminating any hot-spots. Better understanding of the airflow can be addressed by placing pressure and temperature sensors.

By measuring the local pressure, it is possible to estimate the speed and direction of the airflow between the sensed points and possibly identify unwanted mixtures or flow bottlenecks, as shown in [2]. It can also be used for workload-balancing among servers like in [3], where the patented application describes a system that uses a load balancer to shift tasks among servers based on their particular cooling needs, which is related to air pressure drop across the server. With fine grained temperature measurement it becomes easy to localize hot-spots, and by crossing this with pressure data, a better picture of the airflow can be taken, leading to better tuned CRAC systems.

Another important environmental parameter is the local humidity. Higher relative humidity decreases the chances of static electrical discharges that can damage the IT equipment and, at the same time, increases the heat transfer from the server to the cooling airflow. But too much water particles in the air reduces the lifetime of the IT equipment and increases the chance of water condensation at the cold aisles, which is not desirable. Several entities, such as the American Society of Heating, Refrigerating & Air-Conditioning Engineers (ASHRAE), provide guidelines with allowed and recommended values of relative humidity, as well as for dry bulb temperature, maximum dew point, maximum elevation and maximum rate of temperature changes, as seen in [4].

We present a sensing platform for collection of temperature, pressure, humidity and local power consumption (at rack or even server level). The development of the platform was centered on the specific application scenario of energy optimization in large data centers, focusing on high resolution sensing: several sensing points per rack, sampled at sub-seconds time intervals. Evidently, for such system to be practical, cost is an important factor to consider.

With such deployment, our system architecture resembles to the Internet of Things (IoT) paradigm, where a common goal is pursued through the cooperation of Smart Objects (SO) [5]. According to [6] SO's are "*autonomous physical/digital objects augmented with sensing/actuating, processing, storing and networking capabilities.*" In our case, each rack (SO), provides access to processed data from its embedded sensor network, as a contribution to the overall goal of achieving high energy efficient IT rooms and data centers.

The midleware has essential role for this goal. As mentioned in [7], it provides general and specific abstractions, to allow building up complete software structures. Some generic midleware infrastructures were proposed, as in [8] for example. Ours was suited to the data-center context, more specifically to provide means for data-logging, visualization tools, alarm monitors, feedback data for the CRAC systems, or any other application that can be developed in the future. We addressed the developed midleware solution in our previous work at [9], which is not in the scope of this work.

In this paper we will detail the design of the sensor network platform and develop an analysis of the time to obtain the sensing data from the nodes. This is done in order to study the time scales supported by our platform, and also allows to study the delay for different data center topologies. We also exemplify some capabilities of the system with a real deployment.

2 Related Work

Green data centers have received considerable attention in recent research literature. Some recent approaches rely on building software models through a joint coordination of cooling and load management [10, 11], or by formulating an energy minimization problem, subject to service delay and Quality of Service (QoS) constraints. In this class it is worth to mention dynamic voltage scaling [12, 13] and on/off power management schemes [14] – [16]. The complexity of data center airflow and heat transfer is compounded by each data center facility having its own unique layout, so achieving a general model is difficult [17]. For example, in [10], authors stress that their model has several parameters that need to be determined for specific applications.

Given such models, acquiring real-time data at a fine enough spatial and temporal resolution becomes an important topic, as this data can be used to validate models and keep their inputs updated at run-time. Nevertheless, this problem poses new challenges and research issues concerning the type, number and placement of sensors [17].

Some works [18, 19] pushed in the direction of deploying wireless sensor nodes and monitor the thermal distribution, to figure out how to avoid hot-spots and overheating conditions. We differ from such approaches in the sense that we want very fine-grained (in space and time) gathering of power and environmental parameters, including physical quantities other than temperature. Using a mixed wire/wireless solution, [18] obtained a average one-round collection time of approximately 6 seconds for 50 nodes. They also deployed 694 sensor nodes in a data-center, reading every cluster of 4 at most at every 30 seconds. In this work ([18]), for every cluster there was a wireless station and nodes where powered via USB, which makes the system dependent on having a powered USB port available (this might be a problem, since the server to where the node is connected to cannot be powered off, for example). A pure wireless solution was presented in [19], where it was reported a deployment of 107 battery powered wireless nodes, taking 3 seconds to sample all of them (not considering data losses). The experiment only lasted for 35 days before the battery had to be replaced, which is not practical for large, long-lived deployments.

Our proposed system is based on a hierarchical, modular, flexible and fine-grained sensor network architecture, where data is collected from heterogeneous sensors (including power), placed in each rack. The analysis of their inter-correlations will enable closer examination and a better understanding of the flow and temperature dynamics within each data center [20]. To our knowledge, no previous work enables correlating power and environment characteristics on a per rack or per-server granularity with such temporal resolution.

Multiple long-wavelength infrared image sensors can be used to capture thermal maps of an environment [21]. While thermal cameras are an interesting approach, we find that they suffer from several practical issues: (i) the current cost of thermal cameras is substantial, and, due to field-of-view limitations (data centers are typically organized in narrow rows), a high number of them should be required to cover a data center;

(ii) mapping the view of the camera with the infrastructure being monitored is more challenging than relying on point sensors, and it is especially difficult to manage when changes are made to the layout of the data center (e.g., addition/removal of servers and racks), and (iii) by using cameras, the quantitative data analysis would need to be provided by computer vision, which is feasible, but requires a very specific tuning for each scenario and equipment. However, as claimed also by authors in [22], our system has provisions to support thermal image sensors as a smart sensor that can provide temperature field readings with a configurable resolution.

Another approach commonly used is to make measurements thorughout the data center manually, or using mobile robots to automate this task [23]. This approach does not enable practical high-resolution real-time monitoring of the data center as our system does.

3 Overview

The proposed sensor network architecture is a combination of wired and wireless technologies, designed to achieve high spatio-temporal resolution of data center rooms, keeping system's flexibility and modularity, with a low latency and low cost.

Our system is designed to cover the data center first by a short range bus that covers the communication needs inside each rack, a longer range bus that covers each row in the data center and then wireless communication is used to gather the data from the entire data center room. Four different types of devices cover each of these levels (rack, row and room): (i) *Sensing Units* sense the physical parameters (temperature, pressure, humidity, and power) in each rack, then (ii) *Sensor Nodes* collect the sensing data for the entire rack, and (iii) Wireless Base Stations (*WBS*s), collect data from several Sensor Nodes in a row, as represented in Figure 1. Finally, (iv) *Gateways* collect data from all of the *WBS*s in a data center room.

Starting at the lower level, our sensor network consists of two different types of Sensing Units: (a) a small passive sensing unit for measuring environmental quantities, with at most one temperature, one humidity and one pressure digital sensor, and (b) a power metering unit with real, active, and reactive power measurement capabilities, as presented in Figure 1 by *SU-E* and *SU-P* respectively. The environmental Sensing Units can be manufactured according to the sensing and cost needs, by having any combination of sensors on it, what is represented by the three different shapes. Both sensing units deliver data to the next level in the hierarchy, through a wired short range bus (I2C), projected to cover only one rack of servers (back and front).

At the next level, the Sensor Node is responsible to collect the data of all the Sensing Units attached to it and possibly to perform simple data aggregation and sensor fusion before delivering it to the next level in the hierarchy using a longer range wired bus (MODBUS).

WBSs are responsible for querying the Sensor Nodes within their respective cluster, and again perform data aggregation, sensor fusion and data analysis. They communicate then with devices at the next level in the hierarchy to deliver the relevant data. Gateways then provide the data gathered from the sensor network to the data distribution system in a standard format. From this point on, sensing data is published at

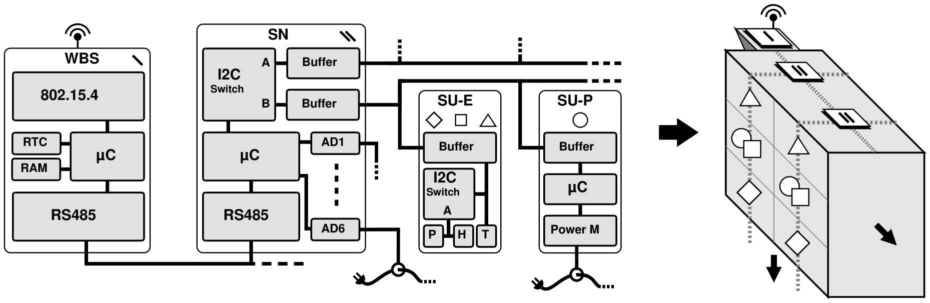
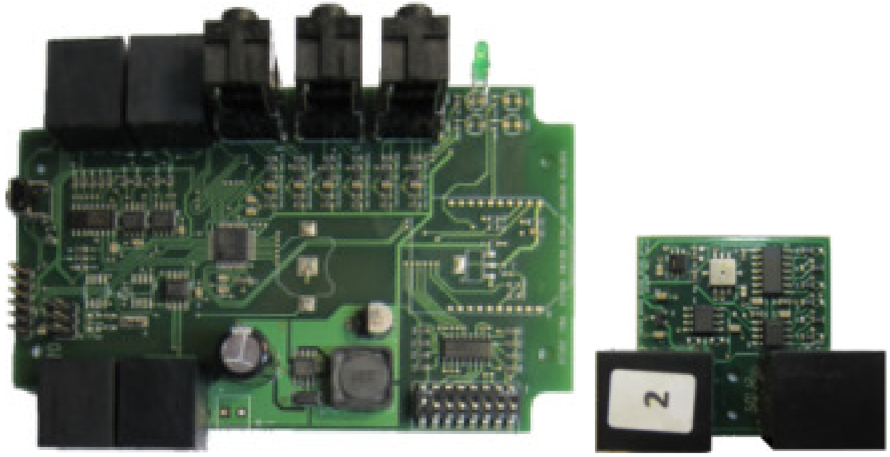


Fig. 1. Network Architecture and Layout



(a) Sensor Node (b) Sensing Unit

Fig. 2. Hardware Platforms

a publish/subscribe middleware that distributes the acquired data to different applications, where each of them will use such information with different proposes (alarms, data logging, visualization, etc).

Each Sensor Node can be connected up to 52 temperature sensors, 54 power meters, 14 pressure sensors and 14 humidity sensors. The following section describes in more detail each of the system components.

4 Platform Details

Well-known protocols, network architectures and of-the-shelf electronic components had to be chosen to compose the system, considering that the final objective was to build a fully functional, industry ready, sensor network with very low cost. Besides the architecture, the technology chosen to implement the network is described below.

4.1 Sensing Unit

With the popularization of two-wire I2C buses on motherboards, cellphones and on general embedded systems, many companies are nowadays developing sensors with digital I2C output, by embedding the micro-mechanical sensor, signal amplifiers, analogue to digital converters, memory and a I2C front end to manage with the communication on the bus. These Systems-on-Chip enable high accuracy and reliability measurements, since this decreases the probability of data corruption due to any external interference. It also prevents calibration issues found on pure-analogue sensors measurements, since digital sensors are factory calibrated and digitally compensated. Due to these reasons, I2C sensors were used to connect the several sensing units.

Some limitations of I2C buses had to be overcome to make its usage practical in this application. First, buffers had to be added as an interface between the I2C bus lines and every circuit board attached to it, in order to allow the I2C to operate over longer distances, by increasing the robustness of the logic signals of the standard I2C buses. Second, switches had to be added to every Sensing Unit on the bus in order to allow the usage of more than one sensor with non-configurable addresses, making it accessible from the main bus.

Figure 2(b) depicts one Sensing Units with temperature, humidity and pressure sensors. The temperature sensor used is a low cost and low power device with 1.5°C accuracy, maximum resolution of 0.0625°C and minimum and maximum conversion times between 27.5 and 300ms. The humidity sensor has 1.8%RH accuracy, with maximum 0.04%RH resolution and minimum and maximum conversion times between 3 and 29ms, both the temperature and humidity sensors suitable for the application, where the focus are in changes in major scales according to the ASHARE guidelines [4], which specifies a range of dew points between 5.5°C (for 60%RH) and 15°C. The pressure sensor ranges from 300 to 1100hPa, with an accuracy of +-1hPa typical and 0.03hPa of resolution with minimum and maximum conversion times between 3 and 25.5ms, also suitable for the application, where typical pressure variation values inside data center's are in greater orders of magnitude, as seen in [2].

The Power Meter Sensing Unit is composed by a dedicated chip which interfaces with the power line, and provide real, reactive, and apparent power measurements to the embedded computational unit, which is responsible for interfacing with the I2C bus as a slave, and to deliver such information to the master, at the next level.

To both Sensing Units, the power is carried into the same cable as the I2C data, and locally converted from 5 to 3.3V by a low-drop LDO converter, for more stable and lower ripple power supply for the sensors, which are sensitive to such variations.

4.2 Sensor Nodes

A Sensor Node is a communication/computation enabled device, physically linked over the I2C bus (also trough buffers) to a number of Sensing Units.The Sensor Nodes gather the data from the Sensing Units and, in turn, answer to data requests from the WBS. Figure 2(a) depicts a Sensor Node.

To keep cost and complexity low at this tier of the network architecture, the Sensor Nodes communicate with one Wireless Base Station (WBS) over a bus, e.g., using a RS485/MODBUS technology [24]. In particular, the WBS node acts as a local coordinator and master of the bus.

The Sensor Node is also composed by: (i) six analogue inputs suited for current measurement, connected to external current transducers attached to the power lines, as a cheap and simple alternative for basic current measurement; (ii) two I2C buffered ports through one switch, responsible for duplicating the bus capacity in terms of addressable devices, and enabling a better mechanical placement for cables to go to the back and front of a rack, and (iii) one RS485 port for the MODBUS.

The power supply for the Sensor Nodes is carried by a twisted pair cable, along with the MODBUS data, in another pair. At every Sensor Node, a high efficiency DC-DC step down converter, converts from 48 to 5V for the local supply. This is an important feature as it reduces the number of cables that connect to each node, facilitating installation of the devices.

4.3 Wireless Base Stations (WBSs)

The WBS is directly connected to a power source and supplies power through a twisted pair cable to all the Sensor Nodes in that bus. In all the nodes on this bus, the voltage is locally converted to lower values by a step-down switched power supply for a higher system efficiency. Wires running in the same cable form a serial data bus (MODBUS over a RS485 connection) that interconnects the Sensor Nodes.

The WBS is based on the same printed circuit board as the Sensor Node, missing the sensors interfaces, and with some extra components, like one external non-volatile ferrite random access memory (FRAM), used as a buffer and for diagnosing the system in cases of failures or power cuts (by keeping the last operational state). The WBS also includes a real-time clock used for time stamping the data packets.

The WBSs act as IEEE 802.15.4 cluster heads and are connected with each other in a mesh topology. A common Gateway is in charge of gathering measurements and sending them over long range communication technology (e.g., WiFi, Ethernet). In terms of HW platforms, the WBS node will be the same platform as a generic Sensor Node, with an on-board ZigBee radio. Thus, each Sensor Node can become a WBS with minimal modifications, i.e., just by plugging the wireless module and uploading a different firmware.

4.4 Gateways

The sensor network can have one or more Gateways. Gateways maintain representations of the data flows from the sensor network to the data distribution system. They perform the necessary adaptation of the data received from the WSN. The gateways can be deployed as one per room serving all the rows of racks in that room; more gateways can also be deployed to improve radio coverage, for load-balancing or for redundancy.

5 Delay Analysis

When performing deployments of our system, we need to answer questions related to how the network should be deployed (for example, we can choose how many sensing points should we deploy per WBS) and what is the impact of this in the performance of the network. To answer such questions we have developed an analysis of the time to transmit sensor data. This analysis also shows that our system can exhibit very low delays in the presence of a large number of sensing points.

This analysis enables us to study the communication delay as we add Sensor Nodes to the network. We consider that each Sensor Node added has N_{su-sn} Sensing Units attached to it, where each Sensing unit has three 16 bit sensors. For every N_{sn-wbs} Sensor Nodes added to the network, one WBS has to be added also. The total number of Sensor Nodes is defined as N_{sn}. Clearly, these parameters (N_{su-sn} and N_{sn-wbs}) are defined according to the topology of the deployment and of the data center room.

5.1 Calculating the Response Time

The response time R required to collect data from all the sensors is given by adding together the time to transmit all the wireless requests to all WBS (t_{req}) and also the corresponding replies (t_{rep}), as given by Equation (1).

$$R = (t_{req} + t_{rep}) \tag{1}$$

The time to transmit all requests is computed by the sum of the time required to transmit a request to each WBS (there are $\lceil \frac{N_{sn}}{N_{sn-wbs}} \rceil$ WBS:s in the network) with the worst-case blocking time, B_{mb}, is given by Equation (2).

$$t_{req} = \left\lceil \frac{N_{sn}}{N_{sn-wbs}} \right\rceil \times (t_{wtx}(S_{wreq}) + B_{mb}) \tag{2}$$

where the $t_{wtx}(S_{wreq})$ is the time to transmit a request packet in the wireless 802.15.4 network including all protocol overhead for a packet with S_{wreq} bits of payload, and will be defined later. B_{mb} is a constant given by the longest data transaction over the MODBUS, which corresponds to the largest task to be executed by the WBS in a non preemptive system.

The time to transmit all replies is given by Equation (3) as follows:

$$t_{rep} = \left(\left\lfloor (N_{su-sn} \times N_{sn}) \times \frac{S_{sd}}{S_{mwp}} \right\rfloor + 1 \right) \times t_{wtx}(S_{mwp}) \tag{3}$$

where S_{sd} is the size of the sensor data to be transmitted by each Sensor Unit and S_{mwp} is maximum wireless data payload, after accounting for all protocols headers. $t_{wtx}(S_{mwp})$ is the time to transmit a packet in the wireless IEEE 802.15.4 network with the maximum possible payload (mwp bits) and will be defined in Section 5.2.

5.2 Calculating the Wireless Transmission time

The reasoning applied to calculating the wireless transmission time ($t_{wtx}(S)$) is similar to the one found in [25, 26] when analyzing the maximum theoretical throughput of a non-beacon enabled IEEE 802.15.4. The time to send a IEEE 802.15.4 packet with payload size of S bits if given by:

$$t_{wtx}(S) = T_{ib} + t_{ppdu}(S) + T_{ack} + T_{ifs} \qquad (4)$$

where T_{ib} is the initial back-off period, which depends on the parameter $macMinBE$, and, by default, $macMinBE = 3$, resulting in $T_{ib} = 1120\ \mu s$). The time to transmit the PHY protocol data unit (ppdu) with a payload size of S bits is denoted by $t_{ppdu}(S)$. The time to transmit an acknowledgment is defined as $T_{ack} = T_{ackppdu} + T_{rxtx} = 544\ \mu s$ since it must include the time to send the acknowledgment packet ($T_{ackppdu} = 352\ \mu s$ as defined in the standard [27]) and the time for the transceiver to switch from receive to transmit ($T_{rxtx} = 192\ \mu s$ is the maximum value defined in [27], and this is the value found in the 802.15.4 transceivers employed [28]). The inter-frame spacing (IFS), T_{ifs}, is set to the value of the long IFS defined by the standard, 640 μs (actually, this is only used when the size of the MAC protocol data unit (MPDU) to be sent is above or equal to 18 bytes [27]).

The time to transmit the ppdu with a payload of size S bits, can be defined as:

$$T_{ppdu}(S) = (S_{hdr} + S_{zbee} + S + S_{ftr}) \times \tau_{bit} \qquad (5)$$

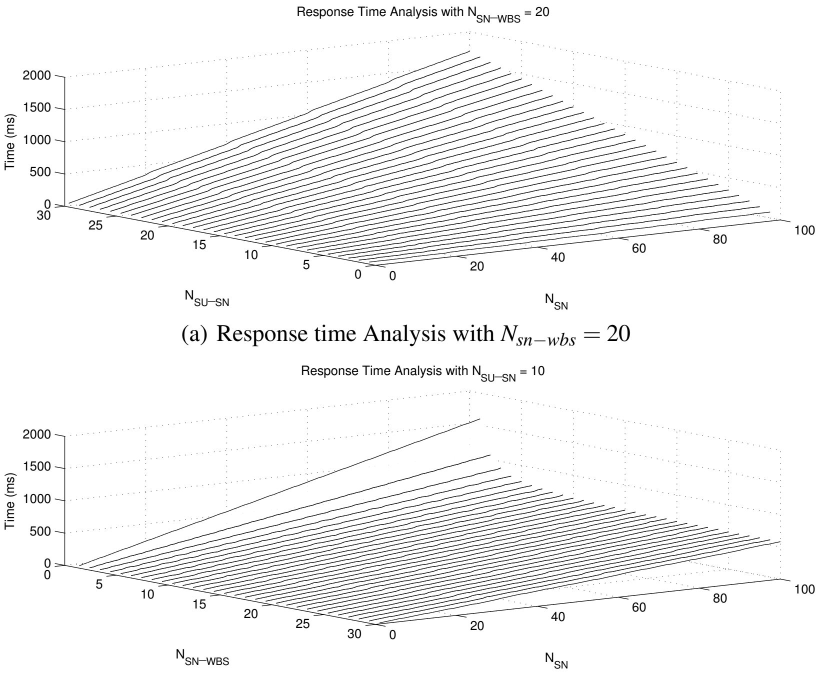
where S_{hdr} is the sum of the sizes of the synchronization header (SHR), PHY header (PHR) and MAC header (MHR; from [27]: $S_{SHR} = 40$; $S_{PHR} = 8$; $S_{MHR} = 56$ bits). The size of the ZigBee protocol headers is $S_{zbee} = 41 * 8$ bits, and the size of the MAC footer is $S_{ftr} = 16$ bits. The time to transmit one bit is $\tau_{bit} = 4\ \mu s$ (for a data rate of 250 kbps).

5.3 Delay Results

Instantiating the response time given by Equation (1) results in Figure 3(a) and 3(b). For these calculations, we have used $S_{wreq} = 16$ bits (a request with a two-byte identifier) and $S_{mwp} = 576$ bits (the maximum IEEE 802.15.4 payload minus the overhead defined in Equation (5)).

With Figure 3(a), we analyzed the impact of adding SN to the network with varying N_{su-sn}. As expected, the increase in the delay is linearly proportional to the N_{su} on the network, when keeping N_{sn-wbs} constant. The higher the N_{su-sn}, higher is the slope. This is expected because the amount data is constantly added as we added SU, however there is a more pronounced increase in response time whenever a WBS is added. In this case, at every 20 $SN's$ added, a higher step is expected due to the overhead of adding wireless links to the network.

Figure 3(b) now shows the case where N_{su-sn} is fixed, and we vary N_{sn} over N_{sn-wbs}. With smaller N_{sn-wbs}, the response time increases very pronouncedly. For example, if there is only one SN per WBS, for every SN added to the network, one more wireless link will be added, causing significant increase in the response time. By increasing N_{sn-wbs}, this effect decreases very rapidly also.



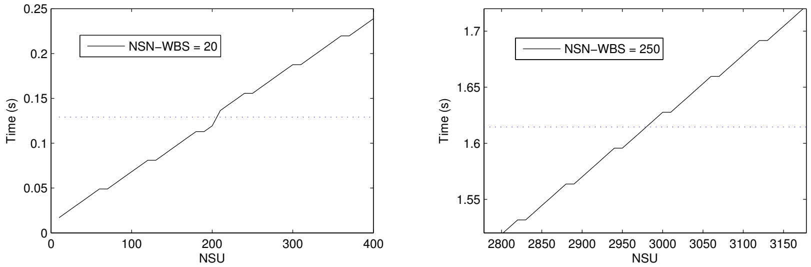
(a) Response time Analysis with $N_{sn-wbs} = 20$

(b) Response time Analysis with $N_{su-sn} = 10$

Fig. 3. Network Response Time For Different Possible Configuration Scenarios

Figures 4(a) and 4(b) present another aspect related to the network topology, which must be considered when designing the network. The horizontal line in both plots shows the time to gather the data from all Sensor Nodes attached to the WSB (20 Sensor Nodes in Figure 4(a), and 250 in Figure 4(b)). The way the network is designed, if one implements a network with N_{sn} below the intersection between the horizontal line and the response time, the wireless communication cycle of the WBS will be faster than the communication cycle on the MODBUS. Thus, the WBS would repeatedly transmit data from previous communication cycles. N_{sn-wbs} should be set such that the lines intersect at the desired N_{sn}. Something that can be easily found, given the analysis presented in this section.

In Figures 4(a) and 4(b), we can see a stepped behavior of the response time, with the growth of the N_{su}. One step happens at each $6 \times N_{su-sn}$. The reason for this step is that, as we add Sensor Nodes, there is the need for and extra packet to be sent (the length of the packet and number of packets needed depends on N_{su-sn} and also on the maximum payload *mwp*). In this scenario, the sensor data for the 7th Sensor Nodes fits in the same number of packets, and thus the delay does not increase. A bigger step is given at every N_{sn-wbs}, due to the overhead of adding one WBS more.



(a) $N_{sn-wbs} = 20$ (b) $N_{sn-wbs} = 250$

Fig. 4. Network Response Time

6 Data Center Visualization

The deployment of the described system in the data center enables interesting opportunities to have better insight into the data center conditions. In this section we will briefly provide some data from a real deployment. This data was selected for its relevance in showing different aspects of the data center conditions that are enabled by the deployment.

The deployment in this section was performed in a data center room owned by the largest telecommunications operator in Portugal. All racks were fitted with two temperature sensors in the front and two temperature sensors in the back. Per row, sensors with additional humidity and pressure sensors were deployed such that the row had three racks (at the top, end, and middle of the row) with such sensors.

Previously, data center operators add a few options to gather such pictures of the data center conditions (e.g thermal cameras or mobile robots), as discussed in the Related Work Section. We claim that our systems enables high-resolution and real-time monitoring of the data center. Something not available in practical systems to date. Our system enables real-time maps temperature, pressure and humidity. These maps are useful to have a detailed picture of the data center conditions. Because the information is collected in real-time, automated control of the data center physical conditions can be enabled.

6.1 Real-Time Thermal Profiling

To illustrate the maps enabled by our tool and to better demonstrate and exploit the capabilities and improvements that our tool can bring the data center management, we have chosen to depict the thermal map of one representative row, as shown in Figure 5.

By analysing Figure 5, it is possible to see the cold air concentration at the bottom of the racks. It happens because the cold air comes from the perforated tiles on the floor, and due to its higher density, compared to the hot air, it stays at the floor level. Enough pressure drop from the bottom to the top of the racks would be required in order to guarantee the cold air flow till the air intake of the server on the top of the racks.

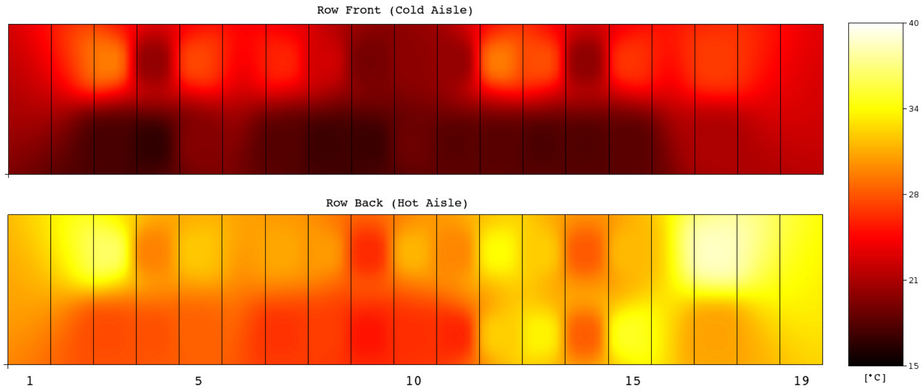


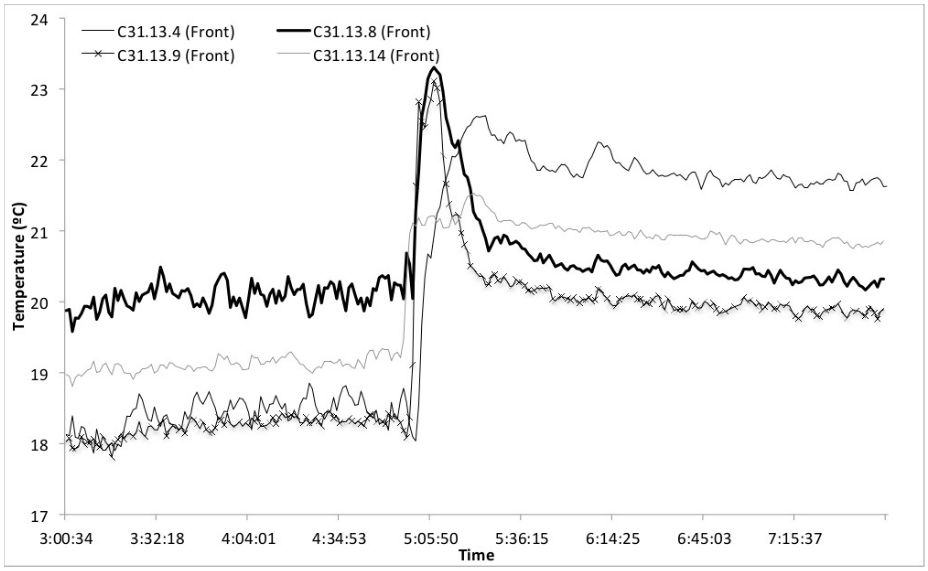
Fig. 5. Heat Map of One Data Center Row

In more detail Figure 5, shows that on the top region, the intake temperature is around 26° C, which is much higher than the cooled air temperature, which is around 17° C. This is also explained by some air recirculation of hot air from the hot aisle (with higher pressure) to the cold aisle. Therefore, the upper servers receive a mixture of hot and cold air, with an intermediate temperature between its output and the cold air temperature.

It is also possible to notice how the temperature at the bottom of the racks gradually rises on the last four racks of the row (to the right). Correlating this with the pressure data, it is possible to notice that the pressure drop between both extremes is not equally distributed, explaining why the cold air does not reach well the last four racks of the row.

Regarding the back side of the row, at the hot aisle, we can clearly see how the air output temperature is correlated with the input air temperature. The colder the air at the input, colder the output flow. Despite this, the workload can also significantly interfere on the heat transfer to the air. One example can be observed at racks 12 and 13, that, even having low temperatures at the air intake at the bottom of the rack, the output temperature raised much more, compared with the neighboring racks. This is a very common effect in heterogeneous data centers, harboring different types of machines, with different powers. Different heat outputs can also be found when workload moves between machines, for example due to workload management in a virtualized infrastructure.

Modifications and Discussion An intervention was made to the row displayed in Figure 5 in order to improve the temperature distribution. The intervention consisted in manually adjusting the perforated tiles located in the cold aisle of the row. Figure 6, presents the average temperature of some selected racks in the row, and allows us to see the evolution of the temperature during that intervention, which took place around 5PM. We can see that the adjustments momentarily caused the temperature to rise, to



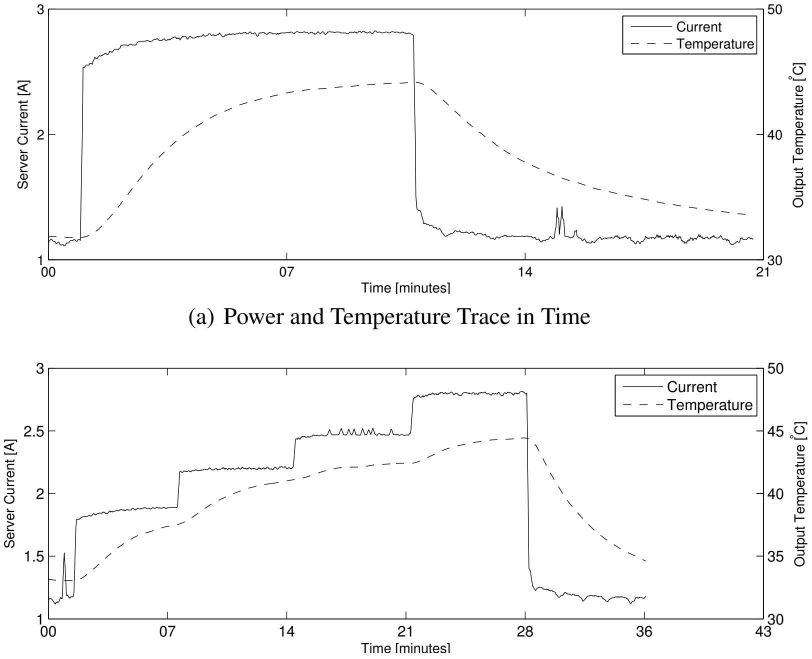
(a) Row Front (Cold Aisle)

Fig. 6. Temperature of a row during perforated tiles adjustments.

then stabilize to a value slightly higher than before the intervention. With this intervention, the temperature distribution in the row became much more homogeneous, and the set point of air conditioning system could be increased, bringing direct savings. It is important to note that the data presented in Figure 6 are provided directly by the monitoring tool developed, and facilitates the observation of how the data center conditions evolve throughout time.

All of the described issues found in this scenario are commonly found in data centers all over the world. In this case, small changes have brought a more homogeneous heat distribution. A second step, with a minor investment, could be to install curtains on the top of the racks, and on its extremity. It could prevent the mixture of cold and hot air, leading to a even better heat distribution and specially, minimizing the waste of energy due to the mixtures. The servers on the top would be directly impacted by this change, by receiving colder air on its intake, contributing to a better distribution. The PUE of the data center can be dramatically decreased with such simple actions.

Similar heat maps can be also obtained from the entire room with its view from the top, possibly presenting any of the different sensors types and position, grouped in any way. Even more, a 3D representation could be done, by using heat transfer models to estimate the surroundings of the measured point with improved accuracy. This models could include input parameters like rack or server instant power consumption, and temperature, humidity and pressure at the air input and output. Externally, this data could be



(a) Power and Temperature Trace in Time

(b) Power and Temperature as Utilization Increases

Fig. 7. Power and Temperature Traces

correlated with the instantaneous workload of every server, allowing work reallocation to minimize hot spots, for example.

With such representations in real-time, the data center manager can have a new representation of the operational condition of it. Local actuation can be done without the need of data exchanges between higher and lower levels of the network, by having a *SN* locally acting on an automated perforated tile.

These actions could be supervised by $WBS's$ due to its better overall picture of the row. It could have some influence over the individual control parameters on every *SN* for a more general control.

6.2 Real-Time Power Profiling

The system developed can also be used to collect real-time data about the power being consumed in the data center. Power data can be collected by power circuit in each rack, or even per server, when the cost is justified.

We took advantage of this feature see if the workload a typical modern server influences the physical parameters measured by our system in a significant way and if this change can happen fast (with, for example, power). This can justify automated local actuation in the data center, and thus elicit the need for real-time data collection.

We deployed sensors around a rack server used for a virtualized infrastructure, in a way similar to a normal data center deployment. We have then measured how the temperature and current consumption varied in time and with changing server workload.

Figure (7) shows the power trace when the workload of the server changes almost instantaneously from an idle state to 100% utilization. This change is reflected almost immediately in the power consumption as seen in the figure. This measurement incurs in the delay bounded by the Equation (1). While much slower, we can see that the temperature also increases significantly as a result of this workload increase. However, it takes about 11 minutes to go up to the maximum of 44°C.

Then, we have increased the utilization if steps of 25% from 0% to 100%. In Figure 7(b) it is possible to see these steps reflected in the current consumed by the server. For the current, the first step presented a rise of 40% over the background consumption, while the following steps rise 20% approximately, showing that the power consumption and temperature have significant variations even for workloads much lower than 100%.

To conclude, we verify that a physical parameter measured (power) does change very fast. The temperature, while being significantly slower still exhibits a large variation over time.

7 Conclusions

We have presented a platform for acquiring the physical parameters of a data center. This platform was developed as a mix of wired and wireless communicating nodes, such that it can enable flexible monitoring of the data center at a very high temporal and spatial resolution of the sensor measurements, while keeping the cost per sensing point very low. Compared to previous work, we enable much higher sensing resolution (several sensing points per rack, sampled at sub-second frequency), maintaining cost low and ease of installation.

We also presented an analysis of the delay of our system. This analysis enabled us to study the communication delay as we add Sensor Nodes to the network, and has shown that our system can exhibit very low delays in the presence of a large number of sensing points. This analysis also allows to try different network deployments and check the trade off between different topologies (described by parameters N_{su-sn} and N_{sn-wbs}) and the resulting delay.

Our experiments have exemplified the data that can be collected by the system and that the physical parameters measured by the system are impacted directly and in a dynamic way by the workload of the servers. Acquiring physical parameters at a very high resolution is important to find opportunities to optimize energy consumption, minimize local hot-spots, achieve more accurate predictive maintenance, perform more accurate billing, and it also enables very fast response to changes in the measured parameters, including automated actuation.

Acknowledgement

This work was supported by National Funds through the FCT-MCTES (Portuguese Foundation for Science and Technology) and by ERDF (European Regional Development Fund) through COMPETE (Operational Programme 'The- matic Factors of Competitiveness'), within projects Ref. FCOMP-01-0124-FEDER-022701 (CISTER), FCOMP-01- 0124-FEDER-012988 (SENODs) and FCOMP-01-0124-FEDER-020312 (SMARTSKIN).

References

1. Google. Google's Green Data Centers : Network POP Case Study.
2. Tom Brey, Pamela Lembke, Joe Prisco, Ken Abbott, Dominic Cortese, Kerry Hazelrigg, Jim Larson, Stan Shaffer, Travis North, and Tommy (Texas Instruments) Darby. CASE STUDY : THE ROI OF COOLING SYSTEM ENERGY EFFICIENCY UPGRADES.
3. Amir Meir Michael and Michael Paleczny. Load Balancing Tasks in a Data Center Based on Pressure Differential Needed for Cooling Servers, 2012.
4. TC ASHRAE. 2011 thermal guidelines for data processing environments expanded data center classes and usage guidance. *ASHRAE*, pages 1–45, 2011.
5. Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.
6. Giancarlo Fortino, Antonio Guerrieri, Michelangelo Lacopo, Matteo Lucia, and Wilma Russo. An agent-based middleware for cooperating smart objects. In *Highlights on Practical Applications of Agents and Multi-Agent Systems*, pages 387–398. Springer, 2013.
7. Giancarlo Fortino, Antonio Guerrieri, Wilma Russo, and Claudio Savaglio. Middlewares for smart objects and smart environments: Overview and comparison. In *Internet of Things Based on Smart Objects: Technology, Middleware and Applications*, Internet of Things. Springer, 2014.
8. Fahim Kawsar, Tatsuo Nakajima, Jong Hyuk Park, and Sang-Soo Yeo. Design and implementation of a framework for building distributed smart object systems. *The Journal of Supercomputing*, 54(1):4–28, 2010.
9. Nuno Pereira, Stefano Tennina, and Eduardo Tovar. Building a microscope for the data center. In *Wireless Algorithms, Systems, and Applications*, pages 619–630. Springer, 2012.
10. Luca Parolini, Bruno Sinopoli, and Bruce H. Krogh. Reducing data center energy consumption via coordinated cooling and load management. In *Proceedings of the 2008 conference on Power aware computing and systems*, HotPower'08, pages 14–14, Berkeley, CA, USA, 2008. USENIX Association.
11. Rongliang Zhou, Zhikui Wang, Cullen E. Bash, and Alan McReynolds. Data center cooling management and analysis – a model based approach. In *28 Annual Semiconductor Thermal Measurement, Modeling and Management Symposium (SEMI-THERM 2012)*, San Jose, California, USA, March 2012.
12. Pat Bohrer, Elmootazbellah N. Elnozahy, Tom Keller, Michael Kistler, Charles Lefurgy, Chandler McDowell, and Ram Rajamony. Power aware computing. chapter The case for power management in web servers, pages 261–289. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
13. Tibor Horvath, Tarek Abdelzaher, Kevin Skadron, and Xue Liu. Dynamic voltage scaling in multitier web servers with end-to-end delay control. *IEEE Trans. Comput.*, 56(4):444–458, April 2007.
14. Ruibin Xu, Dakai Zhu, Cosmin Rusu, Rami Melhem, and Daniel Mossé. Energy-efficient policies for embedded clusters. In *Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems*, LCTES '05, pages 1–10, New York, NY, USA, 2005. ACM.

15. David Meisner, Brian T. Gold, and Thomas F. Wenisch. Powernap: eliminating server idle power. In *Proceedings of the 14th international conference on Architectural support for programming languages and operating systems*, ASPLOS '09, pages 205–216, New York, NY, USA, 2009. ACM.
16. Shengquan Wang, Jian-Jia Chen, Jun Liu, and Xue Liu. Power saving design for servers under response time constraint. In *Proceedings of the 2010 22nd Euromicro Conference on Real-Time Systems*, ECRTS '10, pages 123–132, Washington, DC, USA, 2010. IEEE Computer Society.
17. Jeffrey Rambo and Yogendra Joshi. Modeling of data center airflow and heat transfer: State of the art and future trends. *Distrib. Parallel Databases*, 21(2-3):193–225, June 2007.
18. Chieh-Jan Mike Liang, Jie Liu, Liqian Luo, Andreas Terzis, and Feng Zhao. Racnet: a high-fidelity data center sensing network. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 15–28, New York, NY, USA, 2009. ACM.
19. Beat Weiss, Hong Linh Truong, Wolfgang Schott, Thomas Scherer, Clemens Lombriser, and Pierre Chevillat. Wireless sensor network for continuously monitoring temperatures in data centers. *IBM RZ 3807*, 2011.
20. R. R. Schmidt, E. E. Cruz, and M. Iyengar. Challenges of data center thermal management. *IBM Journal of Research and Development*, 49(4.5):709 –723, july 2005.
21. J. Fredrik Karlsson and Bahram Moshfegh. Investigation of indoor climate and power usage in a data center. *Energy and Buildings*, 37(10):1075 – 1083, 2005.
22. H. Viswanathan, Eun Kyung Lee, and D. Pompili. Self-organizing sensing infrastructure for autonomic management of green datacenters. *Network, IEEE*, 25(4):34 –40, july-august 2011.
23. Jonathan Lenchner, Canturk Isci, Jeffrey O Kephart, Christopher Mansley, Jonathan Connell, and Suzanne McIntosh. Towards data center self-diagnosis using a mobile robot. In *Proceedings of the 8th ACM international conference on Autonomic computing*, pages 81–90. ACM, 2011.
24. Modbus over serial line - specification & implementation guide - v1.0, February 2002. `http://www.modbus.org/docs/Modbus_over_serial_line_V1.pdf`.
25. B. Latré, P. De Mil, I. Moerman, B. Dhoedt, P. Demeester, and N. Van Dierdonck. Throughput and delay analysis of unslotted IEEE 802.15.4. *JNW*, 1(1):20–28, 2006.
26. *Measuring effective capacity of IEEE 802.15.4 beaconless mode*, volume 1, 2006.
27. IEEE. IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 14.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (LR-WPANs), October, 2003.
28. Chipcon. CC2420 datasheet. http://www.chipcon.com/files/CC2420_Data _Sheet_1_3.pdf.