Sustainable Computing: Informatics and Systems 16 (2017) 66–75

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Non-invasive cyber-physical system for data center management

Maurizio Rossi [a,*], Luca Rizzon [b], Roberto Passerone [b], Ivan Minakov [b], Davide Sartori [a], Davide Brunelli [a]

[a] Department of Industrial Engineering (DII), University of Trento, Trento, Italy
[b] Department of Information Engineering and Computer Science (DISI), University of Trento, Trento, Italy

ARTICLE INFO

Article history:
Received 1 April 2016
Received in revised form 16 January 2017
Accepted 18 September 2017
Available online 21 September 2017

Keywords:
Internet of things
Cyber-physical systems
Thermoelectric devices
Energy harvesting
Cloud datacenter

ABSTRACT

We present a Cyber-Physical System (CPS) designed to improve the efficiency of Cloud Data Centers. The hardware part of the system consists of a number of dual functionality devices powered with scavenged thermal energy. The devices can perform two different functions: (i) act as wireless sensing nodes to monitor environmental parameters inside the server room that are important for system reliability and security; (ii) provide active cooling to the CPUs of the data center as *smart heat-sinks*. Starting from the empirical characterization of the energy harvesting module, we determine the amount of energy that can be harvested from a CPU heat dissipation while performing different tasks. We then analyze the amount of energy required to supply the low power sensors or to actuate the cooling fan. The CPS exploits a software simulator of the network of smart heat-sinks to predict the status of the devices. The simulator works in conjunction with a management algorithm used to administrate the cloud infrastructure and the network of *smart heat-sinks* for pursuing global objectives, i.e., maximizing the power efficiency of the data center, while keeping the wireless sensing network alive. Experimental data and simulation results of the framework validate the effectiveness of the proposed method.

1. Introduction

The ever growing demand for IT infrastructure virtualization, cloud services and Internet of Things (IoT) applications is causing a rapid increase in the number of data centers spread all over the world. According to analysts [1], however, this trend will eventually invert in 2017, when the internal company-owned IT infrastructure will undergo a massive migration in so-called "mega data centers" run by few very large service providers. Clients find it convenient to rent their IT facilities, as they can better manage costs and amortize the dynamism of requests, a characteristic from which both the customer and the supplier may benefit. At the same time, big market service providers are better positioned to maximize performance and profits out of their infrastructure, and can obtain hardware at a discount. Optimization also comes from the use of state of the art IT equipment—like servers based on ARM architectures [2,3]—and the reduction of waste and energy consumption. This is particularly true today, since regulatory bodies are issuing laws and guidelines that force the reduction of carbon emissions and push towards a more sustainable energy management/usage.

A well-established metric to measure the energy performance of a data center is the Power Usage Effectiveness index (PUE). The PUE is computed as the ratio between the total data center energy consumption over the IT-specific energy consumption, namely servers, storage and network:

$$PUE = \frac{Total \quad Facility \quad Energy}{IT \quad Equipment \quad Energy}. \tag{1}$$

Introduced by the Green Grid association,[1] the PUE is today the de facto standard performance assessment indicator for data centers. Today, most data centers run in the range from 1.5 to 2.0 PUE. The target PUE for the next data center generation has been envisioned to be close to one (1.0×), but low-power electronics and free cooling will not be sufficient to achieve such an ambitious goal [4].

Recently [5], managers of data centers have expressed a growing interest in monitoring their infrastructures, and in collecting data to improve security and optimization of resources, probably alarmed by the risk of failures or security breaches that can cause millions of dollars in losses [6]. In this sense, the implementation of future data centers will require a completely different design, which includes a distributed and capillary monitoring system, used

* Corresponding author.
E-mail address: maurizio.rossi@unitn.it (M. Rossi).

[1] http://www.thegreengrid.org/.

http://dx.doi.org/10.1016/j.suscom.2017.09.003
2210-5379/

on one side to map the energetic performance of the whole infrastructure and, on the other, to ensure that the required security standards are met. There are many aspects to consider in managing a data center, including the electrical energy cost, the response time (or throughput), the carbon emissions, and safety assurance [7]. Nevertheless, few companies exist in the market that propose a portfolio of data center related equipment that includes systems to extensively monitor relevant environmental parameters, to map and autonomously optimize the usage of resources, to minimize the PUE of the infrastructure, and eventually to save money [8].

To facilitate the increase of performance and security of data centers, we propose a novel framework to monitor and control the whole hardware and software IT infrastructure of cloud service providers. The framework consists of a *Cyber-Physical System* (CPS) for a combined wireless monitoring and hybrid cooling mechanism to use in data center rooms, along with a *workload management scheme* that helps to maximize the data center performance from both the Quality of Service (QoS) and the energy efficiency points of view. The cloud provides distributed control over many data center facilities that can be heterogeneous, and geographically spread. Usually, IT managers prefer policies that maximize QoS to the detriment of energy efficiency, running the hardware infrastructure in a medium performance region, distributing the workload evenly on all the available machines. Our proposed framework achieves better efficiency by monitoring the hardware infrastructure in a capillary fashion, reusing some of the energy otherwise wasted as heat (the CPS role), and exploiting both information and energy to efficiently manage the workload and increase the PUE performance of the infrastructure (the management scheme role).

The monitoring devices are *smart heat-sinks* placed inside server units, consisting of battery-less embedded devices designed to realize a maintenance-free Wireless Sensor Network (WSN) for data center monitoring. The energy necessary to operate is extracted by the thermal dissipation of the servers, the most abundant kind of wasted energy in a data center. By exploiting it, our monitoring system can provide *continuous* sampling of several environmental parameters, like temperature, humidity and light, and *enhanced security* thanks to the use of other sensors [9]. Additionally, when heat is over-abundant, it can be efficiently exploited to activate forced cooling of the underlying system, providing an "energy-free" interval for boosted computation, as already demonstrated in our previous work [10,11]. The CPS management algorithm autonomously selects the operating mode of every smart heat-sink and sends control signals to the workload management algorithm that runs the data center. To do this, the algorithm relies on the predictions of an online simulator able to estimate the charge level of the nodes, and to predict the state evolution of the monitoring/cooling systems. While the main function of the simulator is to provide the forecast data to the workload management software of the data center, it has also been used to validate the hardware and the software of the CPS, as well as to collect the results presented in this paper. This framework can run locally (to manage small private data centers) or can be implemented in the cloud along with any Virtual Machine allocation algorithm, providing more flexibility and enhanced security for distributed IT infrastructures.

In principle, any IT equipment that dissipates energy as heat can power a smart heat-sink. In this paper, we focus on the heat generated by CPUs to deliver a first proof of concept prototype which spans the entire feedback loop from energy harvesting, to monitoring, to server management, without overly complicating the control algorithms. We start from CPUs since their thermal stability is critical to the operation of the data center. However, similar strategies could be applied to the rest of the infrastructure, adapting the management algorithms, potentially coordinating with the computing subsystem. This is especially true, considering the increased adoption of virtualization also for the networking equipment. In this sense, the capillary monitoring afforded by the proposed device could be even more beneficial for improving efficiency.

The article is organized as follows. Section 2 presents the state of the art in the fields of data center management, thermoelectric harvesting to power monitoring nodes, Internet of Things, and cloud services. Section 3 illustrates the applicative scenario of the proposed framework and how the simulator operates, describing the details of the hardware and software modules of the CPS. In Section 4, we summarize the results of the framework, and we discuss resulting performance in terms of transmission rate of monitoring information and data center increase in power efficiency. Finally, Section 5 concludes the paper.

2. Related works

Traditional approaches for data center management rely on the monitoring and assessment of environmental conditions, temperature distribution, servers heat dissipation and power consumption. Several scientific and commercial sensor systems for data centers monitoring have been presented in recent years [12,13]. The available systems offer both wired and wireless monitoring solutions to be spatially distributed in a data center. The commercial wireless monitoring solution are offered by SynapSense,[2] Vigilent[3] and Spinwave.[4] All these products are based on the multi-hop mesh WSN technology comprised of multiple autonomous battery powered sensors devices. Wireless connectivity provides flexibility and ease of nodes installation virtually in any place of interest (e.g., rack shelfs, ceiling). However, system lifetime and sampling rate are limited due to the node finite energy storage. Thus, regular maintenance and battery replacement are required.

In the WSN domain, energy harvesting solutions have been proved effective to sustain or even replace primary energy sources of tiny embedded systems. Previous work in this field demonstrates that ultra low-power wireless sensor nodes can run solely on the energy procured by excessive heat [11], and could potentially be deployed in the data center facility [14]. Thermoelectric systems can convert heat energy into electricity, and are particularly useful in scenarios where the thermal energy is dissipated in large amount, such as in the automotive sector and power electronics, as well as high performance computing (HPC). To control the temperature of a microprocessors and to convert wasted heat into electricity at the same time, researchers have studied the integration of ThermoElectric Coolers (TECs) and ThermoElectric Generators (TEGs) on the CPU cooling devices [15]. The amount of scavenged energy is proportional to the heat generated by the CPU, which in turn directly depends on the kind of task the processor is running, its duration, and the environmental temperature [16]. The energy that can be recovered is relatively small compared to the energy consumed by the CPU itself, therefore it is not sufficient to supply the components of the device from which it scavenges, nor to recharge the system batteries [17]. However, this energy can be used to provide active cooling to the processor, as to increase the PUE of the system by reducing the energy used for cooling (or, conversely, allows the system to run at a higher clock frequency without causing overheat) [10].

A number of research works [18,19] present methods for dynamic servers' workload assignment to reduce heat dissipation of hot spot zones and improve overall data center power utilization. Tang et al. propose abstract models to balance computing power in a data center by minimizing peak inlet temperatures [20]. A holistic approach that manages IT, power and cooling data center infras-

[2] http://www.synapsense.com.
[3] http://www.vigilent.com.
[4] http://www.spinwavesystems.com/.

tructures by dynamically migrating servers workload and cooling is proposed by Chen et al. [21]. Experimental results for a virtual data center demonstrate a reduction in power consumption by 35% and in cooling by 15%. Parolini et al. present a control-oriented model that considers cyber and physical dynamics in data centers to study the potential impact of coordinating the IT and cooling controls [22]. Additionally, sensor data fusion and forecasts from multiple sources can improve workload optimization and scheduling algorithms. The IoT technologies, in this sense, can be suitably applied to the environmental monitoring and IT systems management in data centers as discussed by Rodriguez et al. [13]. IoT allows heterogeneous data to be seamlessly integrated into global applications [23] with the ability to aggregate and analyze multiple sensor inputs, and to control and manage complex system conditions.

Today the IoT is a rapidly growing technological field with constantly expanding application scope. A number of recently presented research works provide comprehensive surveys on the state of the art of the IoT domain. Atzori et al. introduce a brief history of the IoT, the key enabling technologies and overview a wide set of possible applications [24]. He and Xu overview applications for enterprise level IoT systems [25]. A deep analysis of the market available IoT service platforms is covered in [26]. This analysis concludes that there is no solution that fully covers all potential needs for upcoming IoT applications.

Cloud computing is a critical and enabling technology to support decision making systems of IoT-based applications [27,28]. Cloud computing as a term refers to the majority of today's web applications that provide shared resources and services for connected devices and people. This encompasses a wide set application functions, e.g., hosting, storage, security, and content distribution. In the IoT world, cloud computing is presented as a *Platform as a Service* (PaaS), which allows physical objects, devices and software to be globally interconnected within the existing Internet infrastructure. A great number of the cloud platforms for IoT and Machine to Machine (M2M) applications have emerged in the last few years providing application ecosystems with promising benefits for people and businesses.

However, due to the immaturity of the field, the existing IoT cloud market is still fragmented and scattered, thus significant research and development efforts are required [29].

The entire IoT market, and especially PaaS, is a rapidly growing field which is heading towards more complex and comprehensive solutions. However, up to date none of the available service platforms offer a full support for applications which require complex algorithms such as the application presented in this paper.

3. Framework

Fig. 1 shows the architecture of the control framework, designed to coordinate multiple data centers spread worldwide. The framework consists of a certain number of *monitoring systems* deployed in the data center, a *simulator* which predicts the state of charge of the monitoring devices, and a *management algorithm* that coordinates the whole system. The monitoring system is the combination of two electronic systems: a wireless sensor node used for monitoring [11], and an active CPU cooler [10]. Because of its features, we refer to it as the *smart heat-sink*. Each smart sink is built to perform both monitoring and cooling, but it can play only one role at a time, controlled by the manager. The monitoring device is powered with free-energy generated by a Thermoelectric Energy Recovery System (TERS) that has been proven to be self-sustainable under the conditions available inside a server room [16]. The TERS converts the heat dissipated by the data center CPU core into electrical energy that is stored into a supercapacitor and powers the monitoring device.

The simulator uses mathematical models that describe the energy harvesting system to compute and predict the evolution of the temperature of each core, the corresponding harvested power, and therefore the amount of charge in the supercapacitor over time. When the capacitor contains a sufficient amount of energy, the monitoring system is activated and performs its functions. The simulator is able to determine when the node operates, and subtract the corresponding energy the node has spent. The management algorithm exploits the prediction made by the simulator to pursue different objectives. (i) The first is to ensure a spatially distributed monitoring that guarantees that a certain amount of nodes is always configured to perform monitoring. For example, a requirement can be to sample with at least one node in each rack of the data center. (ii) The second objective is to process monitoring parameters to detect and prevent hazardous situations. (iii) The third aim is to set nodes into cooling mode, when it is preferable to cool down a CPU when it is running at high clock frequency, in what we call the High Performance Computation (HPC) state. Lastly, the management algorithm can relocate Virtual Machines to minimize the power consumption, maximize the performance of a server unit or recharge the storage of a smart heat-sink when it is particularly low. Thus, the management algorithm performs different tasks:

1. to take care of collecting and logging the monitored data;
2. to provide the data to the runtime simulator;
3. to decide the optimal policy for the coordination of the nodes (monitoring/cooling) for performance maximization, including Virtual Machines relocation, and to keep the monitoring network alive.

The cloud infrastructure collects the data captured by all the nodes together with information regarding the data centers load at runtime. All the data are logged and used by the simulator to perform system parameters forecasting.

This framework can be tuned to accommodate many different scenarios, because hardware models reside in the simulator while environment configuration (data center in our case) and the policy reside in the management algorithm. In this way, it is possible to evaluate many different scenarios by modifying the data center arrangement, the number of TERS monitoring nodes and policy, without affecting the simulator that allows the TERS node performance to be evaluated. The data center workload also is handled by the management algorithm (can be a random generated list of tasks or a trace from an actual data center), thus it is possible to evaluate many different data center configurations (web-services, HPC, map-reduce, etc.) without modifying the structure of the framework. In the same way, it is possible to keep the data center configuration constant and study just the best TERS system configuration, as we actually demonstrated in Section 4. The following sections provide details regarding each subsystem.

3.1. Smart heat-sink

The smart heat-sink is used to measure several environmental parameters whose value, when out of the ordinary, could denote that a hazard may be incurred. Different types of sensors, such as light or proximity sensor, can detect the presence of unauthorized personnel. The presence of coolant leaks, or unwanted liquids, can be revealed through moisture sensors while low-power image sensors can be used to reveal intrusions [30]. Moreover, by controlling the temperature it is possible to infer whether a fault in the cooling system occurred. Hall sensors can be used to measure power consumption, and balance in dual-supply equipment. Besides monitoring, the smart heat-sink is able to support active cooling of the CPU by activating and powering an axial fan using the collected energy.

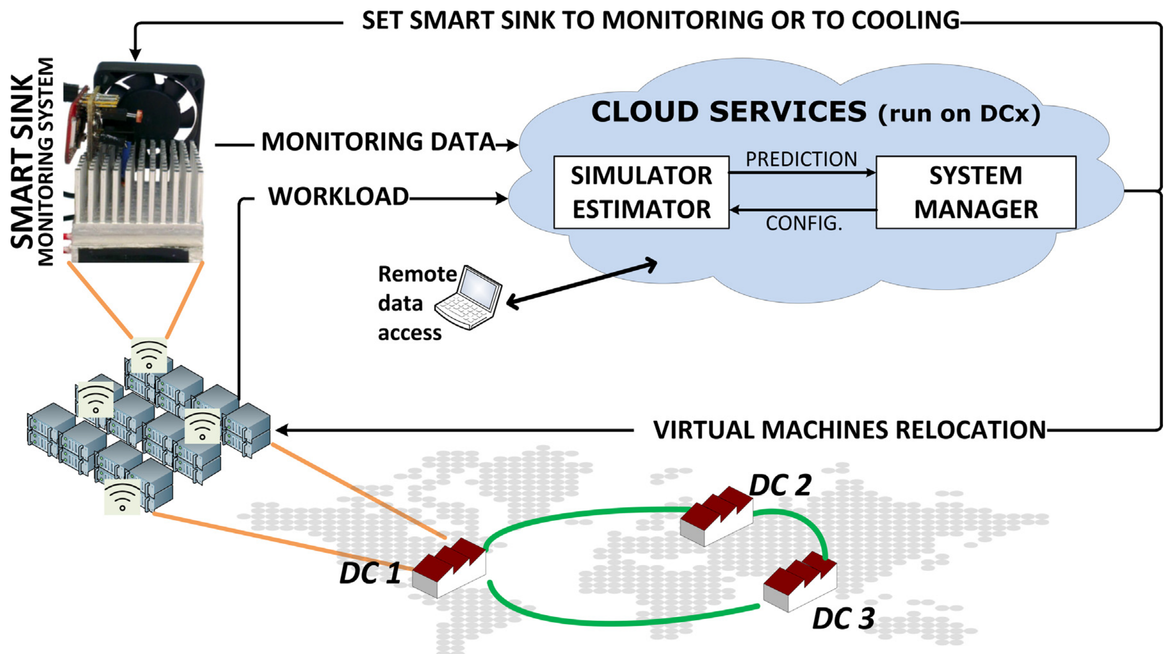

Fig. 1. Architecture of the distributed data center management framework.

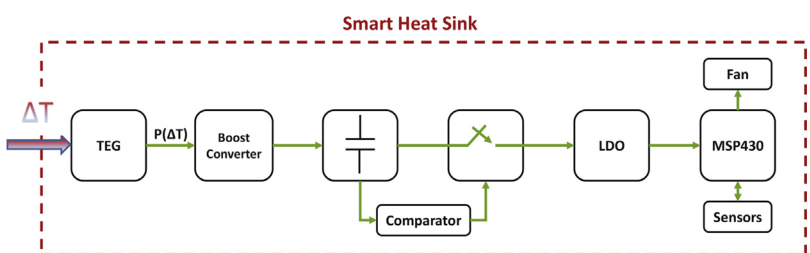

Fig. 2. Block diagram of the smart heat-sink.

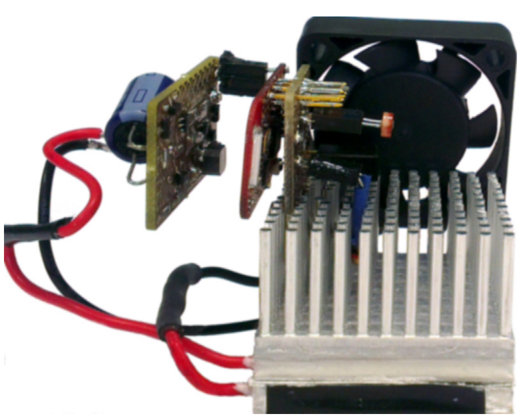

Fig. 3. Picture of the developed smart heat-sink prototype, used for data collection and performance evaluation.

The block diagram of the smart monitoring system, showing its elements and connections, is shown in Fig. 2, while the actual prototype that we have developed is depicted in Fig. 3. The device works with the free electrical energy scavenged from the heat dissipated by the server CPU, no matter what functionality the smart heat-sink is playing (WSN node or active CPU cooler). For this reason, the prototype device has been studied and characterized to maximize the power extraction from heat dissipation. First, the heat dissipated by the CPU is collected and converted into electric energy by a series of two stacked Peltier cells. On top of them, an aluminum heat-sink is able to improve the thermal exchange, and, consequently, the power extraction. These two elements compose the TERS that is as small as $4 \times 4 \times 3.4$ cm (L × W × H). The energy obtained in this way is fed to a dedicated electronic conditioning circuit that raises the weak input voltage with a step-up boost converter, storing the harvested energy in a supercapacitor, and providing supply to the monitoring board, or to the cooling unit (i.e., axial fan). The monitoring circuit board consists of several sensors connected to a system that includes a low-power microcontroller and a RF transceiver (a Texas Instruments eZ430-RF2500). The supply voltage is delivered to the transceiver by means of an LDO (a Texas Instruments TPS78001) which levels off the fluctuations due to the harvesting mechanism. The LDO is triggered automatically by a micropower comparator which enables the operation of the circuit whenever the voltage on the supercapacitor exceeds the threshold $TH_h = 3.08$ V. On the other hand, the fan is actuated only when instructed by the management algorithm by means of a P-N MOS switch controlled by the microcontroller. The power generated from the TERS is proportional to the temperature difference of its two sides (ΔT). Because of the limited amount of harvested energy guaranteed by the TERS (max. 2 mW @ $\Delta T = 31.5\,^{\circ}$C) the selection of the components and the firmware design play a key role. In our specific case, we aimed to monitor the ambient temperature, the heat-sink temperature, the ambient light, and the state of charge of the supercapacitor that feeds power to the monitoring device. The power consumption profiles of the monitoring circuit board were characterized for different dimensions of the storage capacity, and depend on the node state, which could be: *Idle* when the node is powered but doing nothing, *Monitoring mode* when it is used to perform environmental monitoring, or *Cooling mode* if the scavenged energy is used to supply the fan. When the stored energy is not sufficient, the node is placed in the *Off* state, without any power consumption.

3.2. Simulation and management

The simulator software estimates the evolution of the status of the heterogeneous nodes within the data center, and acts as a predictor for the management algorithm. The management algorithm, in turn, implements the policy that helps minimize the overall data center energy consumption and other user defined goals (e.g., maximization of the Quality of Service). The goals are configured at the very beginning of framework execution, as shown in the block diagram of Fig. 4, which illustrates the software modules implementing the framework, and the corresponding flow of control. After the initialization, the algorithm starts collecting data coming from TERS-powered embedded systems (i.e., temperature and State of Charge SoC) and CPUs (freq, load, etc.). Data are gathered periodically every 15 min (or more) and used to verify if the data center is running in compliance with the policy. The historical data are used by the simulator to compute and provide the expected evolution of the data center back to the management algorithm. Then, the management algorithm checks the expected TERS systems performance, and optimally configures the number and the load of the active CPUs, for instance by overclocking some CPUs using free energy if it is possible, or by suggesting a reallocation of the VMs to the data center manager. The models used by the simulator have been developed based on data we gathered on our experimental setup. In case the simulator needs to be customized to simulate a different CPS hardware, or a different host CPU architecture, it would be sufficient to make a measurement run in accordance with the methodology described in [31] to derive the parameters of the mathematical model implemented in the simulator before the execution of the creation of the CPU/TERS mapping step. Since market analysts highlighted that the current trend is to migrate from standard data centers architectures towards ARM based devices, we selected for our analysis an Arndale board equipped with an Exynos, 1.7 GHz dual-core ARM Cortex A15 processor and 2 GB DDR3 RAM @ 800 MHz.

The simulator calculates the amount of energy stored in each node by adding the energy supplied by the TERS, and subtracting the energy that the node has spent in carrying out its activities. The software updates its prediction according to direct analysis of data gathered from the nodes, or indirect analysis obtained from the mathematical model on the basis of the knowledge of the processes that are running on the CPU cores. The simulator calculates a prediction about the amount of energy each of the nodes has recovered and of the simulated temperature values, and provides an estimate of when the smart heat-sink performs boost, and when a device configured to perform sensing samples its sensors and sends the data. By knowing the amount of available energy, the management algorithm can move VMs to pursue its policy, and it can set the node to perform sensing (*Monitoring mode*), or perform cooling (*Cooling mode*). The simulator, in turn, takes the directives received from the management algorithm into account to adjust its predictions.

To correctly compute the energy generated by the TERS, the harvesting system must be characterized. We did this in two steps. First, we studied the correlation between the task that the CPU is performing, and the thermal gradient that builds up between the CPU package and the heat sink. Then, we deduced the function that links the thermal gradient with the TERS output power. We obtained the thermal model of the CPU cores by letting the board run a given task for the amount of time required for the CPU temperature to stabilize. To generalize our analysis, rather than referring to specific software applications, we identify a task the CPU is running as a combination of three values: the CPU clock frequency (f_{CLK}), the percentage of activity in the CPU devoted to the task execution (CPU_{load}), and the amount of time required to complete the task (I_{task}). Each software application running on the host CPU is described in the simulator as a sequence of combinations of those three parameters, which the simulator uses to calculate the thermal gradient ΔT and the generated power P. For each smart heat-sink, the amount of electrical energy available inside the storage unit is defined as E_a, initially set to 0 so that all nodes are in the *Off* state.

The harvester model takes as input the thermal gradient of the corresponding CPU ($\Delta T = T_{hot} - T_{cold}$) inside the server rack. The value of T_{hot} corresponds to the CPU package temperature T_{cpu}, and it is directly influenced by the task that the CPU is running; therefore T_{hot} is computed as a function of (f_{CLK}, CPU_{load}, I_{task}). T_{cold} depends on the cooling system, and possibly on the CPU location inside the server rack. The simulator can work with runtime data of T_{cold} measured by the monitoring nodes, or with experimental data that has been previously collected and organized in a look-up table.

The input–output relation of the harvester, which links the thermal gradient to the generated power, has been obtained by measuring the internal temperature of the host CPU (T_{hot}), the temperature of the heat-sink (T_{cold}), and the corresponding output power P on our workbench. We conducted experiments where, for each f_{CLK} (in the range 1.2, 1.3, 1.4, 1.5, 1.6, 1.7 GHz) we let the host system run with different CPU_{load} (0%,25%, 50%, 75%, 100%) to explore a broad-range of working-points and thermal-points. The results are influenced by transients, however we determined a reasonable mathematical relation using the same approach that was used to characterize a previous version of the same harvesting circuit [16]. The resulting fifth-order polynomial is:

$$P(\Delta T) = a_0 + \sum_{i=1}^{5} a_i \cdot \Delta T^i,$$

where the coefficients describing our demonstrator are:

- $a_0 = -0.6345$
- $a_1 = 0.2852$
- $a_2 = -0.0465$
- $a_3 = 0.0033$
- $a_4 = -9.5715 \times 10^{-5}$
- $a_5 = 9.9485 \times 10^{-7}$

Fig. 5 shows the cloud of dots representing measured values in green, and the fitting curve that is expressed by the polynomial in orange. The amount of generated power that can actually be saved in the storage unit of the harvesting device depends on the conditioning circuit efficiency η. To determine the circuit efficiency, we evaluated the difference between the ideal energy for the storage capacitance C when charged with a voltage V equal to the voltage available at the TERS output, and the energy actually stored in the capacitor on our workbench. The efficiency is inversely proportional to the voltage, and, in the specific case, it ranges from 20% up to 58% for an input voltage in between 240 mV and 500 mV. Thus, the value of harvested energy is: $E_h = \eta \cdot P(\Delta T)$. The energy budget E_a is updated by adding the harvested energy E_h to the amount E_p previously stored in the supercapacitor, and by subtracting the amount E_s of energy that is spent to perform monitoring, or to perform a high performance computation run. The values of E_s used by the simulator are obtained from the characterization of the proto-

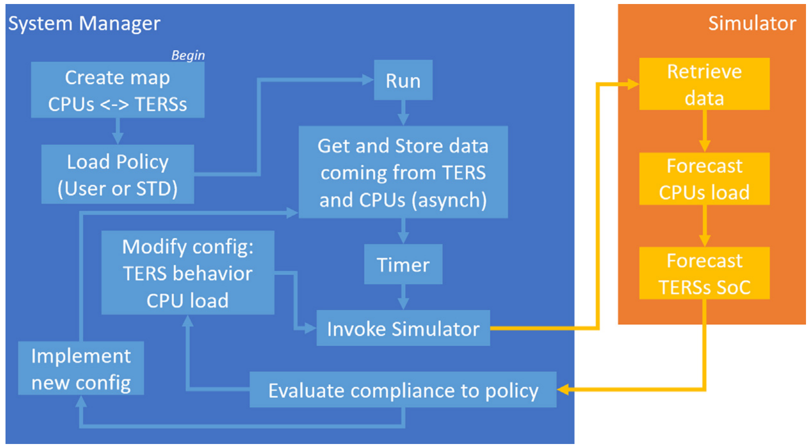

Fig. 4. Block diagram illustrating the software modules implementing the framework.

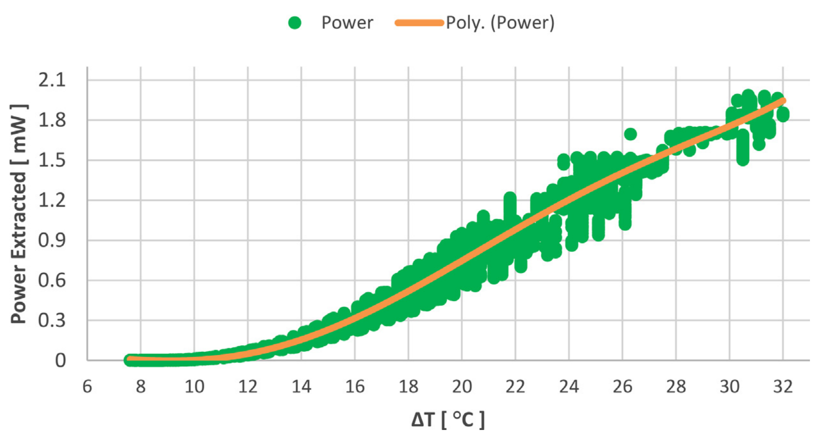

Fig. 5. Relation between thermal gradient and extracted power. Green dots represent measured values, the orange line represents the fitting polynomial used by the simulator. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

type for all its possible states (states are listed at the end of Section 3.1).

When the coordinator sets a node state into *Cooling mode*, the node does not perform monitoring in order to accumulate more energy that it can spend to supply the fan during the HPC cycle. When the available energy is enough (greater than TH_h), the core associated to the device enabled to cool can activate overclock. In the device under examination, normal activity can run at clock frequencies up to 1.5 GHz, but in overclocked mode it reaches 1.7 GHz. While overclocked, the core performs more computation in the same amount of time, or completes the same task with lower duration *I*. The software simulator computes the variation of the task duration according to the variations of f_{CLK}. During a HPC cycle, the simulator considers the CPU as a source of greater heat, thus increasing the values of T_{hot} and T_{cold} according to the look-up table filled with data obtained from our experimental setup. The CPU temperature must not increase more than a safety value, T_{max}, to prevent failures. While the CPU is in HPC, the temperature value associated to T_{hot} acts as a trigger to the activation of the cooling fan of the corresponding CPU. When the fan is activated, the simulator estimates the variation of temperature on both the hot and cold sides of the harvesting device, and decreases the amount of energy available on the storage supercapacitor by a proper amount, that essentially depends on the specifications of the fan. In our case, the smart sink supplies the fan until exhaustion of energy. Subsequently, the core continues to work overclocked until its temperature reaches again the T_{max} threshold, then f_{CLK} returns to the initial value (1.5 GHz, or even lower in case DVFS intervenes).

If the device has been set to *Sensing mode* from the coordinator, the node has to sample periodically the sensors that have been attached to it, to perform environmental monitoring, and send the data to the cloud to log it. We refer to this activity as “sense and send”. This data is saved on the cloud for visualization and is also used by the monitoring application. In this case, the energy budget E_a is updated at every clock period of the simulator. When the capacitor contains enough energy, the device starts working and performs its first “sense and send” cycle. After that, it will perform “sense and send” with a given time interval. The simulator subtracts the energy E_s used by the node only in correspondence of a “sense and send” event, by an amount of energy that strongly depends on which sensors the node is equipped with, and how many of them are used. The device can perform “sense and send” with a fixed data-rate, or, alternatively, the monitoring operation can be scheduled dynamically by the monitoring device itself according to the value of T_{hot}, since the temperature gives information on what the expected scavenged energy is. Even in this case, the simulator considers the node as active only if the available energy E_a is greater than the energy required by the node to operate. While the simulator implements models and features of the smart heat-sink, the management algorithm is responsible for the interaction with the host environment, such as the native data center management platform. This means that it must be customized according to the real platform on top of which it will be executed, to provide indications not only in accordance with the wanted policy (HPC, reduce number of active servers, etc.), but also in the expected form.

3.3. Cloud

To further improve the performance of the IT infrastructure running the CPS, the coordinating scheme (simulator and management software) should be run in the cloud to get complete control over the entire network of data centers under active management. As highlighted in several works (e.g. [32]), environmental concern and economical reasons pushed service providers to find innovative solutions to reduce and optimize the energy consumption of data centers.

Migrating the coordinator into the cloud rises issues of data security, since the data transfer may expose sensitive information and executing on third-party platforms poses concerns on the control over the data owned by the data center manager [33]. However, the cloud reduces the costs of resource upgrade and maintenance, reduces the development time, allows utilities to control geographically distributed infrastructures in a coherent manner, and ensures scalability.

Finally, the proposed framework has been designed for compatibility with state-of-art Virtual Machine allocation algorithms proposed in the last years for cloud infrastructure consolidation, both for single and geographically distributed data centers [34,35]. Considering the block diagram of Fig. 4, data from smart heat-sinks and CPUs will come from different geographical locations, and one simulation is executed for each data center to evaluate the compliance to the policy. Results from these simulations will then be used by consolidation schemes to find the best workload relocation.

The enabling idea is to provide an additional degree of freedom to Virtual Machines consolidation algorithms, allowing the allocation algorithms to boost data centers performance for free, without impacting cooling, and energy consumption. Moreover, the huge and capillary amount of data provided by the monitoring interface can be exploited to increase the performance and the level of details of the forecasting and monitoring algorithm implemented as part of the consolidation scheme.

4. Results

As a case study and performance evaluation of the proposed framework, we present an example of design space exploration targeted to the optimization of the energy storage unit based on supercapacitors. In this case study we compare several system parameters of interest by playing with the size of the energy buffer embedded in the physical system, to directly evaluate performance and the trade-offs of the smart heat-sink. Obviously, any design choice can be changed, evaluated and tuned in the framework or set as a parameter, but a thorough performance analysis of the specific heat-sink is beyond the scope of this work.

The goal in this specific case is to evaluate the trade-off between monitoring efficiency (number of sense and send events or beacon packets transmitted) and HPC sessions (total time spent at maximum CPU frequency) as a function of the capacity of the single node energy storage.

The results presented in this section concern the prototype implementation working with the simulation and management algorithm, including the entire feedback loop shown in Fig. 1, working locally instead of in the cloud. Not having an actual data center available for instrumentation, we have wrapped the whole system around an external system simulation engine based on the MetroII framework [36,37]. In the MetroII design environment, each part of the system under analysis (CPU, TERS, smart heat-sink) is described as an independent component containing its functionality and/or its mathematical model, and the simulator synchronizes the execution of the interconnected components to evaluate the whole system. This feature makes it possible to simulate different architectures by modifying only a single component, or the net of components, and reusing large parts of the existing code, making it versatile for the configuration phase of future setups. In particular, the code for the simulator and the management algorithm was embedded unchanged, and wrapped in an interface. To make the study as realistic as possible, we have used workload traces for the CPUs extracted from a real data center, and included in the simulation stimuli. Thus, the data center itself and its native CPU management software was simulated to achieve the maximum control over the framework and, of course, to simplify and speed up the analysis. The policy used for the assessment is targeted to the maximization of the HPC sessions. As a consequence, when the simulator estimates that the energy level of a node is sufficient to perform an action, the action is directly executed.

The simulator implements a conservative forecasting method, namely the persistence approach, to predict CPUs loads that are then used to estimate smart heat-sinks behavior. This forecasting method assumes that the next expected value is equal to the previous one, in a conservative fashion; obviously, the framework can handle more complex schemes tailored to the specific data center case. Tasks are randomly selected, and are not relocated between cores at run time. The only constraint is to keep alive at least one node devoted to monitoring for each layer of the data center, while all the other nodes can perform HPC as soon as enough energy is available. The results described in this section come from several simulations of the whole system and allow us to demonstrate the improvement of the *PUE* of a data center equipped with the energy neutral monitoring and cooling device presented in this paper.

According to the description provided in the previous section, the following list presents the configuration parameters and the setting used to run the simulations:

- Data Center with 100 CPUs arranged in 10 layers of 5 × 2 servers (100 nodes);
- Variable capacity range of the energy storage from 6 to 100 F;
- Beacon packet transmission interval set to 1 s;
- Switching circuit to charge capacitor in parallel (speed-up) and supply fan with capacitor in series (voltage step-up);
- Sofasco - DC Axial Fan 4.3–5.7 VDC, power consumption $P_{fan} = 0.75$ W;
- Fixed working conditions, normal $f_{clk} = 1.5$ GHz while HPC set to $f_{max} = 1.7$ GHz, CPU load fixed at 100%;
- Simulation length of 1 week (604,800 s);
- Single WSN node configuration.

We select a number of relevant output parameters to perform the design space exploration, according to the following list:

- N_{WHS} – total number of nodes in the experiment;
- $\bar{\Delta I}$ – storage recharge time, that is the time between TH_l and TH_h threshold crossing for a single node;
- $\bar{I}_{fan}$ – average fan utilization time for a single node in a single HPC session;
- $\bar{I}_{FM}$ – average fan utilization time for a single node during the whole time horizon;
- $\bar{I}_{HPCM}$ – average single HPC session length;
- $\bar{N}_{TX}$ – average beacon packet transmissions for a single node in the whole time horizon;
- $\bar{N}_{HPC}$ – average number of HPC sessions in the whole time horizon;
- $\bar{I}_{HPC}$ – average HPC session length for a single node in the whole time horizon;
- γ – percentage “gain” in terms of number of additional instructions executed taking advantage of the HPC with respect to the

baseline (running constantly @ f_{clk} = 1.5 GHz), in the whole time horizon.

The performance data is obtained by averaging the output of a number of simulation runs initialized with the same parameters (to account for the random choices). The results, summarized in Table 1, provide useful information to help designers tune their target application.

Considering the storage recharge time ($\bar{\Delta I}$), we notice that an increase in the storage capacity results in a corresponding linear increase of the recharge time, from approximately 40 min for the 6 F, up to more than 11 h for the 100 F case. Similarly, the average fan activity ($\bar{I_{fan}}$) is directly proportional to the capacity, resulting in very short active cooling tasks for small storage units (5 and 9 s in the case of 6 and 12 F) up to more than a minute for the last case; notice that losses and non-linearities caught by the model are reflected in the less than linear increase of the fan activity: we could expect to get 83 s of activity for the 100 F case since $100/6 \simeq 16.67 \cdot 5\,\mathrm{s} \simeq 83\,\mathrm{s}$, while the actual ratio we got is ≈14.

Despite the large difference in the recharge times for the various cases, the total time the fan is used ($\bar{I_{FM}}$) is quite constant in the range 850 to 900 s for all the experiments. This depends on the amount of energy that is harvested by the smart heat-sink which is almost constant and unrelated with the storage capacity, since any CPU in our scenario run with constant load/frequency while the energy depends on the working temperature. While 900 s of fan activation may seem small, this translates in a much higher duration for high performance operations ($\bar{I_{HPC}}$), which reaches 64,711.8 s in the best case. The average HPC session length ($\bar{I_{HPCM}}$) is directly influenced by the fan activity time even if, comparing the 6 and 100 F cases, we can notice only a 2-fold increase in this parameter with respect to a 16-fold increase in the capacity; once again this depends on the thermal behavior of the system more than on the storage capacity and energy efficiency. The average number of HPC sessions ($\bar{N_{HPC}}$) follows the same linearity we computed for the average fan activity ≈14.

Regarding the number of beacon packet transmissions, we notice an alignment in all the experiments above 30 thousand per node; thanks to the monitoring strategy designed, this results in more than 5 packets per second received by the gateway.

Considering the different relationships and performance parameters just discussed, we can confirm that the best tuning option is to reduce the size of the storage as much as possible to achieve better performance, as demonstrated by the “gain” factor reported in the last column. This gain takes into consideration the number of additional instructions executed by the system taking advantage of the HPC sessions with respect to the case when the CPU works with constant frequency. Notice that this gain affects neither the thermal stability of the infrastructure nor the power usage, resulting, and this is the most relevant result, in a reduction of the overall *PUE* of the data center infrastructure, as we discuss in the following.

As reported in Section 1, the *PUE* index relates the total energy consumption to the energy consumed by the IT equipment, namely the energy spent in useful computation (see Eq. (1)). To evaluate, quantitatively, the impact of the smart heat-sink control strategy on the overall data center efficiency, we initially consider power rather than energy and factor out the contribution of the IT equipment power consumption, denoted *IT_factor*, and that of the rest of the equipment, denoted *oE*, and rewrite Eq. (1) as:

$$PUE = \frac{IT\_factor + oE}{IT\_factor} = 1 + \frac{oE}{IT\_factor}. \quad (2)$$

Given that our prototype does not contribute any extra power consumption (since all extra energy is scavenged from the environment), the *oE* term remains constant after the application of the smart heat-sink. On the other hand, the power consumption *IT_factor* dedicated to providing useful computation increases, since the active cooling allows the processors to run at higher speed. Let *bf* denote the *boost factor*, or the increase in power for useful computation. Then, we may compute the new value for the *PUE* ratio as

$$PUE_{new} = 1 + \frac{oE}{IT\_factor \cdot (1 + bf)} \quad (3)$$

If we denote by PUE_{ref} the original value, then the gain, which corresponds to a reduction of the *PUE*, can be computed as:

$$gain = \frac{|PUE_{ref} - PUE_{new}|}{PUE_{ref}} = \frac{oE}{IT\_factor + oE} \cdot \frac{bf}{1 + bf} \quad (4)$$

which shows that the gain depends both on the original fraction of wasted power, and on the boost factor.

For our specific case study, we have measured the power consumption for our board in both normal working conditions and with the applied smart heat-sink. The results are shown in Fig. 6. The boost factor, or increase in power consumption due to the activation of the HPC sessions on this platform, is equal to 35.77%. We can then compute the gain by assuming an initial *PUE* value. Fig. 7 shows the reference *PUE* and the potential reduction assuming the measured increase in power of 35.77% over a range of *IT_factor* values. The results are very interesting. For instance, if PUE_{ref} = 1.5, then the IT_factor is twice as large as oE, or

$$PUE_{ref} = 1 + \frac{1}{2}.$$

Given the measured boost factor, the new value for the *PUE* (highlighted in Fig. 7) is

$$PUE_{new} = 1 + \frac{1}{2(1 + 0.3577)} = 1.37.$$

for a gain of almost 9%. This analysis was performed considering power values, which clearly illustrate the difference in performance for the reference and the improved case. To obtain the *PUE* in energy, we must integrate the power values over time, i.e., the *PUE* is given by a combination of the power values for the two cases weighted by the portion of time in which they are used. A fair comparison between the two cases must also take into account the difference in the amount of work completed in the same time. In fact, while the baseline runs at 1.5 GHz for a total time of 604,800 s in a week, the performance increase using HPC means that the system executes a larger number of instructions in the same amount of time, or, conversely, that it completes the same task in less time. Consider for example the case with the largest gain, corresponding to a storage supercapacitor C = 6 F. In this case, the host CPUs run at the HPC rate for 64,712 s. Assuming a constant instructions per cycle (IPC) rate, the number of instructions per seconds (IPS) grows linearly with the clock rate. Therefore, the server will perform the same amount of computation as the baseline in I_{tot} = 596, 172 s (which means a reduction of the execution time by 2 h 24 m), with I_{HPC} = 64, 712 spent at 1.7 GHz, and the remaining $I_{1.5\,\mathrm{GHz}}$ = 531, 460 at 1.5 GHz. The resulting *PUE* depends on the effective utilization of the resources and workload dynamics. In ideal terms, the maximum potential for the considered architecture will be:

$$PUE_{\max} = \frac{PUE_{ref} \cdot I_{1.5\mathrm{GHz}} + PUE_{new} \cdot I_{HPC}}{I_{1\mathrm{week}}} = 1.465. \quad (5)$$

In a more realistic scenario, one should also take into account the fact that the data center workload will not be constant over time, and adjust the gain earned with HPC boosting accordingly. Moreover, communication with the cloud has an impact on the PUE, since it involves network and storage systems that are supplied by the grid. However, this contribution can be neglected because, if we consider an average of 35,000 transmissions of 15 bytes each (8

Table 1
Summarized values of HPC simulations results.

N_{WHS}	$Cap(F)$	$\bar{\Delta I}$ (s)	$\bar{I}_{fan}$ (s)	$\bar{I}_{FM}$ (s)	$\bar{I}_{HPCM}$ (s)	$\bar{N}_{TX}$	$\bar{N}_{HPC}$	$\bar{I}_{HPC}$ (s)	γ (%)
100	6	2463	5.00	897.5	360.91	38,825.1	179.3	64,711.8	1.41
100	12.5	4996	9.01	895.5	385.91	31,390.0	99.1	38,244.3	0.75
100	25	10,190	17.6	888.9	429.22	37,035.9	50.4	22,138.9	0.37
100	50	21,345	32.3	881.9	543.09	30,411.4	25.6	13,914.8	0.27
100	75	30,779	51.7	884.7	650.78	31,643.7	17.1	11,128.4	0.20
100	100	41,039	68	857.9	752.2	30,345.9	12.6	9477.7	0.18

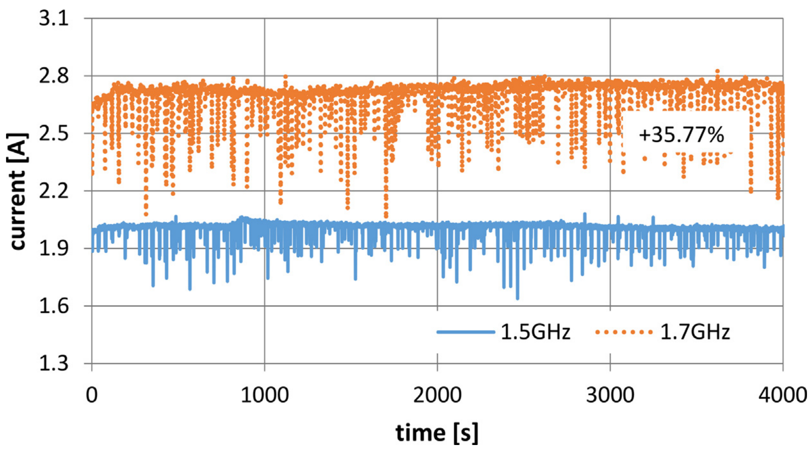

Fig. 6. Current consumption profile for the two reference modes of operation.

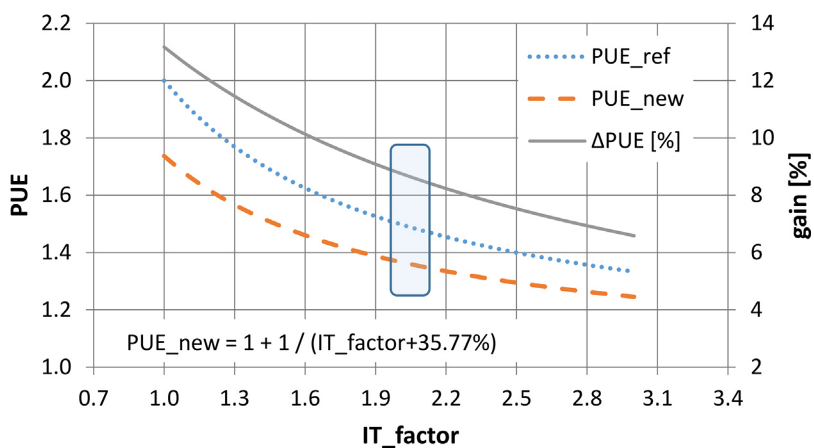

Fig. 7. *PUE* analysis.

bytes for data plus overhead) per week, as it results from the simulations, this corresponds to 525,000 bytes/week or 0.86 bytes/s that are sent through a very high-speed (optical-fiber cables with at least 1 GB/s speed) communication channel that is and must be always on to guarantee the SLA (service level agreement) of the DC. We have conducted a study over 100 CPUs and corresponding sensors. The approach can be scaled to larger deployments by an appropriate allocation of resources to the predictor. In extreme cases, if this were to become a bottleneck, it would be easy to parallelize to meet performance requirements. On the other hand, real-time performance is not a major concern, given the slow time constants involved in heat related processes. A large number of sensor may however produce a corresponding large amount of packets to be delivered to the control algorithm, resulting in communication congestion. This problem can be solved through appropriate communication protocols, which are facilitated by the delay tolerance of the system.

5. Conclusions

Monitoring and optimizing data center facilities require hardware resources and management technologies. Data Centers owners are interested in reducing the energy consumption and keeping the computational performance high to ensure optimal quality of service without affecting the costs. We described a smart heat-sink for data center CPU that provides two functionalities. It serves as a data collection node to provide remote control of health parameters under the paradigm of Internet of Things. At the same time, it can be used both as a passive and active CPU cooler. Both functions are carried out with free energy converted from the wasted heat that is abundant in the environment in which it is immersed. The system includes a simulator software used to predict the evolution of the server condition. The simulator can be used to provide data to a system coordinator that can decide the best strategy to control all the deployed smart heat-sink, to switch them to one functionality or to the other, and to provide control commands to the data center workload manager. The simulator has been also used as design tool to perform design space exploration of the CPS and determine the optimal size of the energy storage unit. Moreover, we show how the CPS allows us to estimate an increase of the energy efficiency in the device we considered as target platform.

Acknowledgement

The work presented in this paper was supported by the project *GreenDataNet*, funded by the EU 7th Framework Programme (grant no. 609000).

References

[1] Press Release, IDC finds growth, consolidation, and changing ownership patterns in worldwide datacenter forecast, in: IDC Corporate, 2014, November 10 http://www.idc.com/getdoc.jsp?containerId=prUS25237514.
[2] R. Merritt, Dell, IBM Give Thumbs Up to ARM Servers, EE Times, 2010.
[3] Z. Yeraswork, AMD Launches First ARM-based Server CPU, EE Times, 2014.
[4] NWSC, Green Technology – Raising the Bar in Data Center Efficiency. https://nwsc.ucar.edu/green (accessed 25.03.15).
[5] R. Gates, Data Center Physical Security Gets a Tougher Look, http://searchdatacenter.techtarget.com/news/4500248374/Data-center-physical-security-gets-a-tougher-look.
[6] G. Rusconi, 2011. L'incendio ad aruba.it provoca il più grande black out internet in Italia. In tilt posta e siti di molte aziende, Il Sole 24 Ore.
[7] M. Pedram, Energy-efficient datacenters, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 31 (10) (2012) 1465–1484.
[8] Aktiv Consulting, Synapsense wireless solution. http://www.aktiv-consulting.com/wireless-sensor-technology.html (accessed 15.04.15).
[9] D. Brunelli, R. Passerone, L. Rizzon, M. Rossi, D. Sartori, Self-powered WSN for distributed data center monitoring, Sensors 16 (1) (2016) 57.
[10] L. Rizzon, M. Rossi, R. Passerone, D. Brunelli, Energy neutral hybrid cooling system for high performance processors, in: Proc. of the 5th International Green Computing Conference (IGCC), Dallas, TX, USA, 2014.
[11] L. Rizzon, M. Rossi, R. Passerone, D. Brunelli, Self-powered heat-sink SoC as temperature sensors with wireless interface: design and validation, in: IEEE Sensors 2014, Valencia, Spain, 2014, pp. 1575–1578.
[12] C.-J.M. Liang, J. Liu, L. Luo, A. Terzis, F. Zhao, RACNet: a high-fidelity data center sensing network, in: Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys), ACM, Berkeley, CA, USA, 2009.
[13] M. Rodriguez, L. Ortiz Uriarte, Y. Jia, K. Yoshii, R. Ross, P. Beckman, Wireless sensor network for data-center environmental monitoring, in: Proc. of the 5th Int. Conference on Sensing Technology (ICST), Palmerston North, New Zealand, 2011, pp. 533–537.
[14] M. Rossi, L. Rizzon, M. Fait, R. Passerone, D. Brunelli, 2016. Self-powered active cooling system for high performance processors, Application in Electronic Pervading Industry Environment and Society, Volume 289 of Lecture Notes in Electrical Engineering, Editor: A. De Gloria.
[15] S. Lee, P.E. Phelan, C.-J. Wu, Hot spot cooling and harvesting CPU waste heat using thermoelectric modules, in: Proc. of ASME 2014 Int. Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers, Montreal, Canada, 2014.
[16] M. Rossi, L. Rizzon, M. Fait, R. Passerone, D. Brunelli, Energy neutral wireless sensing for server farms monitoring, IEEE J. Emerg. Sel. Top. Circuits Syst. 4 (3) (2014) 324–335.
[17] Y. Zhou, S. Paul, S. Bhunia, Harvesting wasted heat in a microprocessor using thermoelectric generators: modeling, analysis and measurement, in: Design, Automation and Test in Europe (DATE'08), Munich, Germany, 2008, pp. 98–103.
[18] C. Bash, G. Forman, Cool job allocation: measuring the power savings of placing jobs at cooling-efficient locations in the data center, in: Proc. of the USENIX Annual Technical Conference (ATC'07), USENIX Association, Santa Clara, CA, 2007, pp. 29:1–29:6.
[19] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, C. Kozyrakis, Power management of datacenter workloads using per-core power gating, Comput. Archit. Lett. 8 (2) (2009) 48–51.
[20] Q. Tang, S. Gupta, G. Varsamopoulos, Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach, IEEE Trans. Parallel Distrib. Syst. 19 (11) (2008) 1458–1472.
[21] Y. Chen, D. Gmach, C. Hyser, Z. Wang, C. Bash, C. Hoover, S. Singhal, Integrated management of application performance, power and cooling in data centers, in: Proceedings of the IEEE Network Operations and Management Symposium (NOMS), Osaka, Japan, 2010, pp. 615–622.
[22] L. Parolini, B. Sinopoli, B. Krogh, Z. Wang, A cyber-physical systems approach to data center modeling and control for energy efficiency, Proc. IEEE 100 (1) (2012) 254–268.
[23] S. Fang, L.D. Xu, Y. Zhu, J. Ahati, H. Pei, J. Yan, Z. Liu, An integrated system for regional environmental monitoring and management based on internet of things, IEEE Trans. Ind. Inform. 10 (2) (2014) 1596–1605.
[24] L. Atzori, A. Iera, G. Morabito, The internet of things: a survey, Comput. Netw.: Int. J. Comput. Telecommun. Netw. 54 (15) (2010) 2787–2805.
[25] W. He, L.D. Xu, Integration of distributed enterprise applications: a survey, IEEE Trans. Ind. Inform. 10 (1) (2014) 35–42.
[26] O. Mazhelis, P. Tyrvainen, A framework for evaluating Internet-of-Things platforms: application provider viewpoint, in: Proc. of the IEEE World Forum on Internet of Things (WF-IoT), Seoul, South Korea, 2014, pp. 147–152.
[27] C. Wang, Z. Bi, L.D. Xu, IoT and cloud computing in automation of assembly modeling systems, IEEE Trans. Ind. Inform. 10 (2) (2014) 1426–1434.
[28] L.D. Xu, W. He, S. Li, Internet of things in industries: a survey, IEEE Trans. Ind. Inform. 10 (4) (2014) 2233–2243.
[29] G. Suciu, S. Halunga, A. Vulpe, V. Suciu, Generic platform for IoT and cloud computing interoperability study, in: Proc. of the Int. Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 2013, pp. 1–4.
[30] N. Cottini, M. Gottardi, N. Massari, R. Passerone, Z. Smilansky, A 33 μW 64×64 pixel vision sensor embedding robust dynamic background subtraction for event detection and scene interpretation, IEEE J. Solid-State Circuits 48 (3) (2013) 850–863.
[31] L. Rizzon, M. Rossi, R. Passerone, D. Brunelli, Wireless sensor network for environmental monitoring powered by microprocessors heat dissipation, in: Proc. of the 1st International Workshop on Energy Neutral Sensing Systems (ENSSys), Rome, Italy, 2013.
[32] E. Pakbaznia, M. Pedram, Minimizing data center cooling and server power costs, in: Proc. of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), ACM, San Francisco, CA, USA, 2009, pp. 145–150.
[33] N. Gupte, J. Wang, Secure power grid simulation on cloud, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 34 (3) (2015) 422–432.
[34] J. Kim, M. Ruggiero, D. Atienza, M. Lederberger, Correlation-aware virtual machine allocation for energy-efficient datacenters, in: Proceedings of the Conference on Design, Automation and Test in Europe (DATE), EDA Consortium, Grenoble, France, 2013, pp. 1345–1350.
[35] M.F. Zhani, Q. Zhang, G. Simon, R. Boutaba, VDC planner: dynamic migration-aware virtual data center embedding for clouds, in: Proc. of the IFIP/IEEE International Symposium on Integrated Network Management (IM2013), IEEE, Ghent, Belgium, 2013, pp. 18–25.
[36] L. Rizzon, R. Passerone, Cyber/physical co-design in practice: case studies in METROII, in: Proceedings of the 11th IEEE International Symposium on Industrial Embedded Systems, SIES16, Krakow, Poland, 2016.
[37] A. Davare, D. Densmore, L. Guo, R. Passerone, A.L. Sangiovanni-Vincentelli, A. Simalatsar, Q. Zhu, metroII: a design environment for cyber-physical systems, ACM Trans. Embedded Comput. Syst. 12 (1s) (2013) 49:1–49:31.