Energy and thermal models for simulation of workload and resource management in computing systems

Wojciech Piątek [a,*], Ariel Oleksiak [a,b], Georges Da Costa [c]

[a] Poznan Supercomputing and Networking Center, Noskowskiego 10, Poznan, Poland
[b] Institute of Computing Science, Poznan University of Technology, Poland
[c] Institute of Computer Science Research, University of Toulouse, France

Keywords:
Simulations
Thermal models
Energy-efficiency
Data centers

A B S T R A C T

In the recent years, we have faced the evolution of high-performance computing (HPC) systems towards higher scale, density and heterogeneity. In particular, hardware vendors along with software providers, HPC centers, and scientists are struggling with the exascale computing challenge. As the density of both computing power and heat is growing, proper energy and thermal management becomes crucial in terms of overall system efficiency. Moreover, an accurate and relatively fast method to evaluate such large scale computing systems is needed. In this paper we present a way to model energy and thermal behavior of computing system. The proposed model can be used to effectively estimate system performance, energy consumption, and energy-efficiency metrics. We evaluate their accuracy by comparing the values calculated based on these models against the measurements obtained on real hardware. Finally, we show how the proposed models can be applied to workload scheduling and resource management in large scale computing systems by integrating them in the DCworms simulation framework.

1. Introduction

One of the main obstacles in building and cost-effective use of large scale HPC systems is their significant energy consumption. Thus, appropriate energy consumption modeling, monitoring, and optimization is essential to enable a fast growth of HPC systems computing power and to decrease their operational costs, which should pave the way for the exascale HPC systems.

Importantly, optimizing energy consumption of processors and servers is only part of the equation. Data center infrastructure (with the major contribution of cooling) may consume even the same amount of energy as IT equipment. To include cooling into optimization of the computing system, thermal aspects must be taken into consideration. These aspects influence both local energy consumption of servers and global consumption of the whole data center. Impact on local energy consumption is caused by differences in operation of fans and changes in power usage of CPUs dependent on temperature. Since the energy usage has a strong correlation with the thermal efficiency of the computing system [2], it is valuable, for its effectiveness, to take both power and temperature distribution into consideration while managing the data center equipment. On a global level the amount and distribution of heat dissipated by servers affect the overall energy consumption by the cooling system. Moreover thermal-aware management is crucial not only from the perspective of energy costs, but also for the

* Corresponding author. Tel.: +48 618582169; fax: +48 618582151.
E-mail address: piatek@man.poznan.pl (W. Piątek).

system maintenance, as excess heat can cause downtimes and shorten equipment durability. These issues lead also to additional costs of system operation and upgrades. Thus, energy consumption and heat dissipation transforms obviously into hard limits of HPC system development due to maximal power constraints as well as maintenance costs.

Often experiments on management of large scale computing systems are difficult or not possible to conduct in real environments. Especially for new planned centers, new computing architectures, and future exascale systems simulation studies are needed in advance to propose optimization of workload and resource management. Hence, simulation models and tools are needed to study architectures of new computing systems, their management, cooling and application deployment. To enable realistic simulation of such systems many challenges must be met including on one hand dealing with large scales and sizes of data, and on the other hand, taking into consideration specific aspects and technologies of new architectures such as special focus on power usage and heat dissipation, various heterogeneous and low power architectures, diversity of cooling techniques, and higher probability of failures.

In this paper we propose models and simulation environment to address the problem of simulation of large scale computing systems including energy-efficiency and thermal aspects. The main contributions of this paper include: (i) power usage and thermal models of air-cooled servers along with validation on real hardware, (ii) approach to enabling transient simulations of dynamic systems (allowing studying on-line reactions on temperature and dynamic cooling control), and (iii) demonstration of the use of models in management policies: showing effect of fan management, managing power and cooling capacity, and taking into account characteristics of temperature changes in time. All these contributions are applied to the DCworms simulator [18].

The work presented in his paper is focused on the airflow cooling approach. Proposed models should easily cover both air and heatsink-based solutions as well as any combination of them. However, nowadays there exist alternatives including variety of gaining popularity liquid-based approaches. Although the methods below cannot be directly applied to address other cooling techniques, they can provide a good basis for such extension.

The remaining part of this paper is organized as follows. In Section 2 we give a brief overview of the current state of the art concerning power and thermal models. Section 3 presents the models we propose to use in the simulation of energy consumption and temperature of computing systems. In Section 4 we assess the proposed models by comparison to results of experiments performed on a real hardware. In Section 5 we demonstrate the application of these models to a few types of resource and workload management policies in the DCworms simulator. Final conclusions and directions for future work are given in Section 6.

2. Related work

Understanding thermal phenomena and then describing it in the analytical way has been the subject of intense studies for decades. Modeling such processes has gained significant interest in the area of physics and recently also in the scope of computer science. Providing accurate models is important not only in terms of the design of IT and non-IT devices but also in terms of evaluation of the system efficiency and management policies. Moreover, temperature depends on several factors, like time-varying IT heat loads, physical room layout and performance of cooling facilities. Rapid growth of computing capabilities starting from single servers, to typical data centers and to the ultrascale systems, has make it even more challenging with respect to the precision and time complexity of the simulation tools benefiting from these models.

In order to characterize the thermal distribution, there are several approaches that are differentiated by their accuracy and required model size. Fig. 1 gives the general overview [9].

For now, Computation Fluid Dynamics (CFD) simulations are considered as the most accurate approach. However, they require lots of effort, while preparing model and even more time to obtain rewarding results, which makes it expensive to use in terms of big systems simulations. As an alternative, Potential Flow Model (PFM) has been proposed [11], that benefits from the reduction of the model. Another approach follows proper orthogonal decomposition methodology [10] and corresponds to Reduced Order Models entity in Fig. 1.

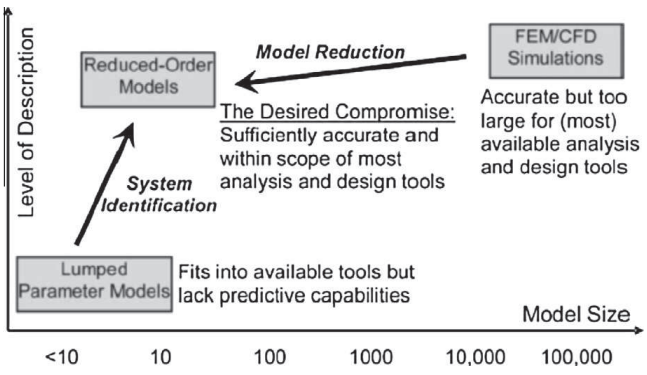


Fig. 1. Existing approaches complexity [9].

As the complexity of computing systems is increasing together with the growing importance of their energy efficiency, the processes of their evaluation has become difficult and complex to perform in real environments. Thus, simulations have been used as a mean to overcome this issue. However, to perform such studies, models reflecting the behavior of computing systems need to be provided in advance. Such models should describe the system components that are relevant from the perspective of the overall energy and thermal efficiency. The following subsections summarize some approaches proposed in the literature.

2.1. Power modeling

A review of the current state of the art shows several literature examples of power modeling for computing systems. A direction of such studies span from network systems [26] through storage systems [25] up to servers systems [23]. Additionally, researchers considered also virtualized environments [24].

As the main concern of our studies is related to the computing components, we focus on related work addressing this issue. In [12] authors propose models that present data center power usage in a comprehensive way. On the processor level they distinguish the models for a single and multi-core processor. Moreover, Basmadjian introduced also power consumption models for other system components, like hard disks, memory, power supply units and Storage Area Network.

2.2. Temperature modeling

From the perspective of thermal models, most of the approaches correspond to the Newton's law of cooling, Fourier's law of conduction and the first law of thermodynamics – namely law of energy conservation. The temperature of the processor and the temperature of outlet air (at the outlet of particular servers) are usually considered to be the most significant from the perspective of overall computing system.

2.2.1. Processor

In general, models describing processor thermal behavior benefit from the duality between thermal and electrical phenomena [1]. According to it, heat transferred through the thermal resistance can be expressed using the following formula:

$$C\frac{dT_{cpu}}{dt} + \frac{T_{cpu} - T_{amb}}{R} = P_{cpu} \tag{1}$$

where C is a thermal capacitance $\frac{dTcpu}{dt}$ describes a temperature increase of the processor, T_{cpu} specifies the temperature of the processor, T_{amb} is a temperature of ambient air, R defines thermal resistance and P_{cpu} is the processor power consumption. Solving integral equation gives the temperature of CPU in time as Eq. (2).

$$T_{cpu}(t) = P_{cpu}R + T_{amb} + (P_{cpu}R + T_{amb} - T_{cpu}(0))e^{-\frac{t}{RC}} \tag{2}$$

where $T_{cpu}(t)$ is a temperature of the processor at given time t and $T_{cpu}(0)$ is initial temperature of this processor. Equation in the given or similar form prevails in most of the encountered approaches [3–5].

2.2.2. Outlet temperature

In general outlet temperature is derived using the law of energy conservation and the basic heat transfer equation. It states that:

$$Q = \rho V C_p \Delta T \tag{3}$$

where Q is amount of heat transferred ρ is the air density, V its volume, C_p is specific heat capacity of air and ΔT defines change in temperature

The product of first three values defines the thermal absorption capacity of air:

$$K = \rho V C_p \tag{4}$$

Assuming that ΔT represents the difference in temperatures between the outlet and inlet of a server and heat transferred Q is equal to the power drawn by this server, the equation above can have the following form:

$$T_{out} = \frac{P_{server}}{K} + T_{in} \tag{5}$$

where T_{out} is an outlet air temperature P_{server} defines a power consumed by the server and T_{in} is the temperature of inlet air.

Models following this concept are the starting point of the work done by [7,6,8], where the heat recirculation idea was introduced and expanded into the concept of heat distribution matrix.

3. Modeling of thermal and energy efficiency

In the following section, we introduce the power and thermal models for the most significant data center components.

3.1. Power modeling

Our approach [29] to estimate the power consumption of computing systems is partially derived from the CoolEmAll project. We assume that data center server room is composed by a number of racks and a cooling system. Each rack consists of a set of node groups, which are then responsible for hosting a collection of nodes. Node groups can be identified with enclosure that specifies both the placement of nodes within the given node group as well as mounted fans. The main component of the node is a processor with assigned number of cores and computing capability (expressed by a clock speed). Moreover, each processor comes with its power and performance profile, described by the means of C-States and P-States defining operating states with corresponding power usage values for different utilization levels. Node definition is supplemented by a description of memory and network. Rack represents a standardized frame for carrying server and power supply modules.

The following equations show how the power usage for different resource levels is estimated.

$$P_{cpu}(t) = (P_{cpu}(idle) + load(t) * (P_{cpu}(f(t), 100) - P_{cpu}(idle))) * g(T_{cpu}) \tag{6}$$

where P_{cpu} is a power consumed by a processor operating at a given frequency f and utilized in *load* percent. $P_{cpu}(idle)$ and $P_{cpu}(f, 100)$ expresses the power drawn by an idle and fully loaded processor working at a given frequency, respectively. Finally, $g(T_{cpu})$ is a function representing the power leakage of the processor.

Power consumption of a node is given by:

$$P_{node}(t) = \sum_{i=1}^{n} P_{cpu_i}(t) + P_{mem}(t) + P_{net}(t) \tag{7}$$

where P_{node} is the node power drawn, n stands for the number of processors assigned to the node, P_{mem} is the power used by a memory, while P_{net} by a network.

The next component aggregates computing nodes within an enclosure. Its power model is defined as:

$$P_{node\_group}(t) = \sum_{i=1}^{m} P_{node_i}(t) + \sum_{j=1}^{k} P_{fan_j}(t) \tag{8}$$

where P_{node_group} is a power consumed by a group of nodes, m provides the number of nodes placed in a node group, k is the number of fans mounted within it and P_{fan_j} is a power used by particular fan j.

Finally, power usage of a rack is calculated as follows:

$$P_{rack}(t) = \left(\sum_{i=1}^{l} P_{node\_group_i}(t) \right) \Big/ \eta_{psu} \tag{9}$$

where P_{rack} is a rack power consumption, l defines the number of carried node groups and η_{psu} is efficiency of a power supply unit.

Description of cooling models for the whole data center can be found in [22].

3.2. Temperature modeling

We are considering an enclosure capable to host a number of nodes. Such node group may stand for the rackable server or a blade enclosure containing several blade servers. Each node group has a fan (or fans) attached, responsible for blowing air (described by its volume V) over its internal components. Fig. 2 shows the general idea behind the proposed model. Four spots for modeling the temperatures are considered as a significant: inlet temperature (T_{in}) represents the temperature at the inlet of the server; similarly, outlet temperature (T_{out}) indicates the corresponding value at its outlet, T_{cpu} stands for the temperature of the processor, while ambient temperature, within an enclosure, is defined by (T_{amb}). One should note, that local ambient temperature may vary depending on the processor placement within an enclosure.

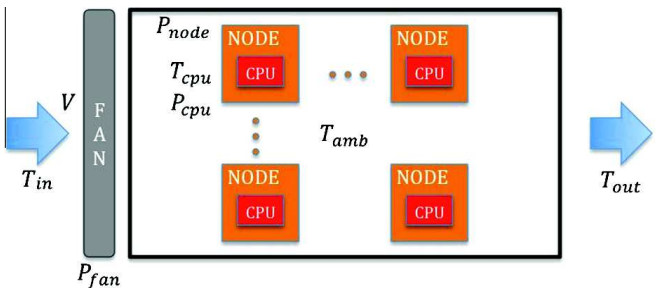


Fig. 2. Thermal model.

3.2.1. Processor

In order to model the thermal behavior of the processor we follow the duality between thermal and electrical phenomena (mentioned in the previous section) and the relation between convective resistance and airflow coming from the fan rotation [2]. According to the former model, dependency presented by Eq. (2) can be also expressed using the reference to the temperature at the previous timestamp.

$$T_{cpu}(t+\Delta t) = T^{\infty}_{cpu}(t+\Delta t) + \left(T_{cpu}(t) - T^{\infty}_{cpu}(t+\Delta t)\right)e^{-\frac{\Delta t}{R(t+\Delta t)C}} \tag{10}$$

where Δt is a time step and $T^{\infty}_{cpu}(t) = P_{cpu}(t)R(t) + T_{amb}(t)$ which is the steady temperature for the given processor dissipating a given amount of heat. Thermal resistance R is often represented by a conductive resistance and convective resistance [17,19].

$$R(t) = R_{cond} + R_{conv}(t) \tag{11}$$

with the convective part defined as:

$$R_{conv}(t) = \frac{1}{h(t)A} \propto \frac{1}{AV(t)^n} \tag{12}$$

where h is the heat transfer coefficient, A is the effective area of the resistance., V is the airflow volume and n is a resource-specific coefficient. As the heat transfer coefficient is proportional to V^n, then to decrease R_{conv}, and thus to increase the heat removal capabilities, the air velocity must increase at an even greater rate. Let us write this dependency in the following way:

$$R_{conv}(t) = \frac{1}{k_v V(t)^n} \tag{13}$$

where k_v is the parameter that needs to be determined experimentally as it's typical for a given equipment model. One should note that k_v veils the heat transfer coefficient and the effective area of the resistance.

To calculate the airflow volume V we can use the dependency:

$$V(t) = P_{fan}(t)\mu_f / dp(t) \tag{14}$$

where P_{fan} is a power consumption of the fan, μ_f defines fan efficiency and dp is a total pressure drop.

As [20]:

$$dp(t) \propto V(t)^2 \tag{15}$$

we can also write that:

$$V(t) = \sqrt[3]{k_p P_{fan}(t)} \tag{16}$$

with k_p parameter that needs to be determined experimentally for specific hardware configurations.

Substituting the above dependencies to the previous equation for the processor temperature we finally get the following relation:

$$T_{cpu}(t+\Delta t) = T^{\infty}_{cpu}(t+\Delta t) + \left(T_{cpu}(t) - T^{\infty}_{cpu}(t+\Delta t)\right)e^{-\frac{\Delta t}{R(t+\Delta t)C}} \tag{17}$$

with $T^{\infty}_{cpu}(t) = P_{cpu}(t)R(t) + T_{amb}(t), R(t) = R_{cond} + \frac{1}{k_n V(t)^n}$ and $V(t) = \sqrt[3]{k_p P_{fan}(t)}$

This form of description allows us to express processor temperature reduction, due to the higher fan rotation speed.

3.2.2. Outlet air temperature

To model changes in outlet temperature, we adopt the law of energy conservation and the basic heat transfer equation, introduced in the previous section. Combining that knowledge with our previous model for the power consumption of a node group, we specify outlet temperature for the single node enclosure as:

$$T_{out}(t) = \frac{P_{node}(t) + P_{fan}(t)}{K(t)} + T_{in}(t) \tag{18}$$

where P_{node} is the power consumed by the node and P_{fan} is the power drawn by its fan. K refers to the heat absorption capacity of air defined by $\rho V(t) C_p$. Let us write this relation as follows:

$$T_{out}(t) = \frac{P_{cpu}(t) + P_{others}(t)}{K(t)} + T_{in}(t) \tag{19}$$

where P_{cpu} is the power consumed by the processor and P_{others} is the power drawn by other components of the server (like memory, storage and fans). Then, benefiting from Eq. (1), the above formula takes the form:

$$T_{out}(t) = \frac{C\frac{dT_{cpu}}{dt} + \frac{T_{cpu}(t) - T_{amb}(t)}{R(t)} + P_{others}(t)}{K(t)} + T_{in}(t) \tag{20}$$

Solving this differential equation leads us to the following relation:

$$K(t)R(t)(T_{out}(t) - T_{in}(t)) - T_{cpu}(t) + T_{amb}(t) - P_{others}(t)K(t) = k_s e^{-\frac{t}{R(t)C}} \tag{21}$$

where k_s covers initial system condition. Isolating T_{out} we have:

$$T_{out}(t) = T_{in}(t) + \frac{P_{cpu}(t) + P_{others}(t)}{K(t)} + k_s e^{-\frac{t}{R(t)C}} \tag{22}$$

Finally, extracting the formula for the increase in outlet temperature, we receive:

$$T_{out}(t + \Delta t)) = T^{\infty}_{out}(t + \Delta t) + \left(T_{out}(t) - T^{\infty}_{out}(t + \Delta t)\right) e^{-\frac{\Delta t}{R(t+\Delta t)C}} \tag{23}$$

where $T^{\infty}_{out}(t) = \frac{P_{node}(t)+P_{fan}(t)}{K(t)} + T_{in}(t), K(t) = \rho V(t) C_p$ and $R(t) = R_{cond} + \frac{1}{k_n V(t)^n}$ as in previous section. In case of several nodes: $T^{\infty}_{out}(t) = \frac{\sum_{i=1}^{n} P_{node_i}(t)+P_{fan}(t)}{K(t)} + T_{in}(t)$.

4. Evaluation of the models

In this section we present a comparison between the temperature derived on the basis of the above models and the measured values obtained within the real environment. We start from the description of the testbed we used, together with applied methodology and conclude with the discussion of the results. Description of the power models accuracy can be found in [21,22].

4.1. Testbed configuration

We carried out our experiments on the Christmann's RECS systems (Resource Efficient Computing & Storage) [14]. We evaluated the RECS system filled with various types of nodes. The following figure (Fig. 3) shows RECS architecture for two different types of processors studied, namely Intel Core i7-3615QE and Toradex Apalis T30 with ARM processors.

As presented, RECS system contains 18 baseboards that can be equipped with 18 × 86- or up to 72 ARM CPU modules (18 baseboards × 4 CPU modules). Blue arrows indicate the direction of the airflow, while the dark-gray rectangles stand for the fans, which is 18 in total (one pair per each pair of nodes). To monitor temperature of both input and output airflows we installed temperature sensors (AKCP securityProbe 5ES) both on inlets and outlets of the RECS. Temperature sensors are also located under each of the baseboards and within the CPU unit. Moreover, in case of RECS equipped with ARM processors we are able to monitor the temperature in front and behind the node and additionally, adjust the speed of the fans in a manual way within the range 30–100%. According to the vendor, each fan can blow 0.0112 m^3/s of air drawing 6.6 W of power, while operating at full speed.

4.2. Methodology

In order to be able to extract the necessary parameters, we first ran a stress workload generator [15] on the given types of nodes. For the simplification we evaluated only nodes placed within one column (from the inlet to the outlet of the RECS). Then, we monitored each second all the relevant values like: power usage of a processor, power usage of a node, temperature

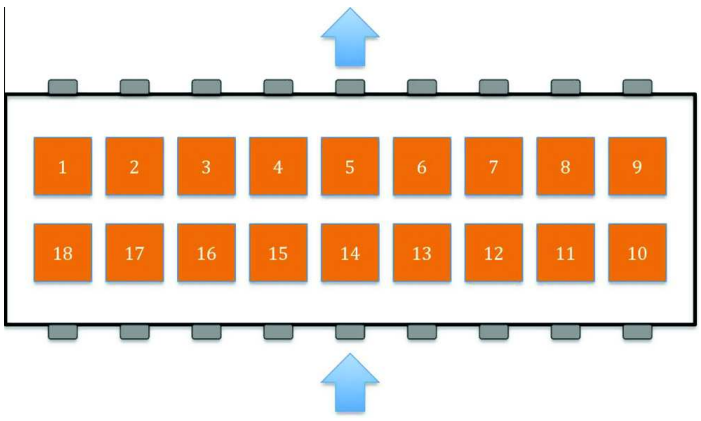


Fig. 3. Illustration of RECS architecture.

of a processor and both inlet and outlet temperature. In case of RECS with ARM processor, we could also obtain internal temperature values. Moreover, as we were able to adjust the fan speed, we repeated our tests, increasing the fan speed by 10 within the range 30–100%. During our studies we observed harmonic fluctuations in the inlet temperature with an amplitude reaching 2 °C and interval ~240 s due to characteristics of the air conditioning system applied. As we did not have means to measure the local ambient temperature (T_{amb}) as stated in [13], we had to estimated it. Based on our observations, we noticed that a good approximation of it is a weighted average of inlet and processor temperature, expressed as follows: $T_{amb} = (T_{cpu} + 2 * T_{in})/3$.

4.3. Obtained results

In this section we present the validation results obtained for two configuration of RECS system.

4.3.1. Case study 1 – RECS with Intel i7 nodes

In our first study, we ran stress tool using three different frequencies levels: 1200, 2300 and Turbo Boost mode. Based on the measurements we extracted the following values for the thermal resistance and capacitance (Table 1).

Afterwards, we ran HPL benchmark [16] and applied the required parameters into the proposed models. Then, we compared temperature values calculated according to the models and the ones derived from the measurements. The following set of pictures presents comparison of temperatures obtained for three mentioned frequencies for the processor on the inlet side. Table 2 summarizes the accuracy of the proposed model.

Obtained results suggest high accuracy of the proposed model. Both the detailed values and the evolution of temperature are reflected with satisfactory score. One should see that the precision of the model decreases with the higher frequency levels. In particular, estimation of processor temperature operating in Turbo Boost mode results the significantly higher (comparing to the higher P-states) absolute error. This may be a consequence of rapid fluctuations in processor power consumption values (caused by rapid changes in frequency) which are crucial for its temperature estimation (see Fig. 4).

Next set of pictures (Fig. 5) shows the comparison of outlet temperatures. Once again we present the results for running the HPL benchmark on the processor at the inlet side. Table 3 shows the accuracy of the proposed model.

One should see the higher relative errors of the outlet temperature model comparing to the processor one. However, the absolute error is still small and does not exceed 1 °C. Moreover, the shape of the curve of our model corresponds to the changes recorded on the testbed, except the amplitude. Higher amplitude (up to 2 °C) arises directly from the model and is consistent with the fluctuations observed for inlet temperature. One should remember that, according to the model, $T_{out}^{\infty} = \frac{P_{node}+P_{fan}}{K} + T_{in}$ and thus each change in inlet temperature affects directly outlet temperature value. Moreover, one should note that the oscillations in simulated model reflect the fluctuations of inlet temperatures with the interval ~240 s. Real data indicates similar wavelength (covering simulated model) but with lower amplitude (around 1 °C). This suppression of outlet temperature oscillations might be caused by the surroundings of outlet temperature sensor, which accumulates heat and impacts the measurements.

4.3.2. Case study 2 – RECS with ARM nodes

In the second case study, we ran stress tool once again. As said above, we ran the tests for various rotation speeds, ranging from 30% to 100%. Based on the measurements, we extracted the following values describing both conductive and convective thermal resistance as well as thermal capacitance (Table 4). Table 5 shows the detailed values of convective thermal resistance for different levels of fan speed.

In the next step we applied the required parameters into the proposed models. Fig. 6 presents comparison of temperatures obtained for three different fan speed levels, while running stress application on both nodes in one column.

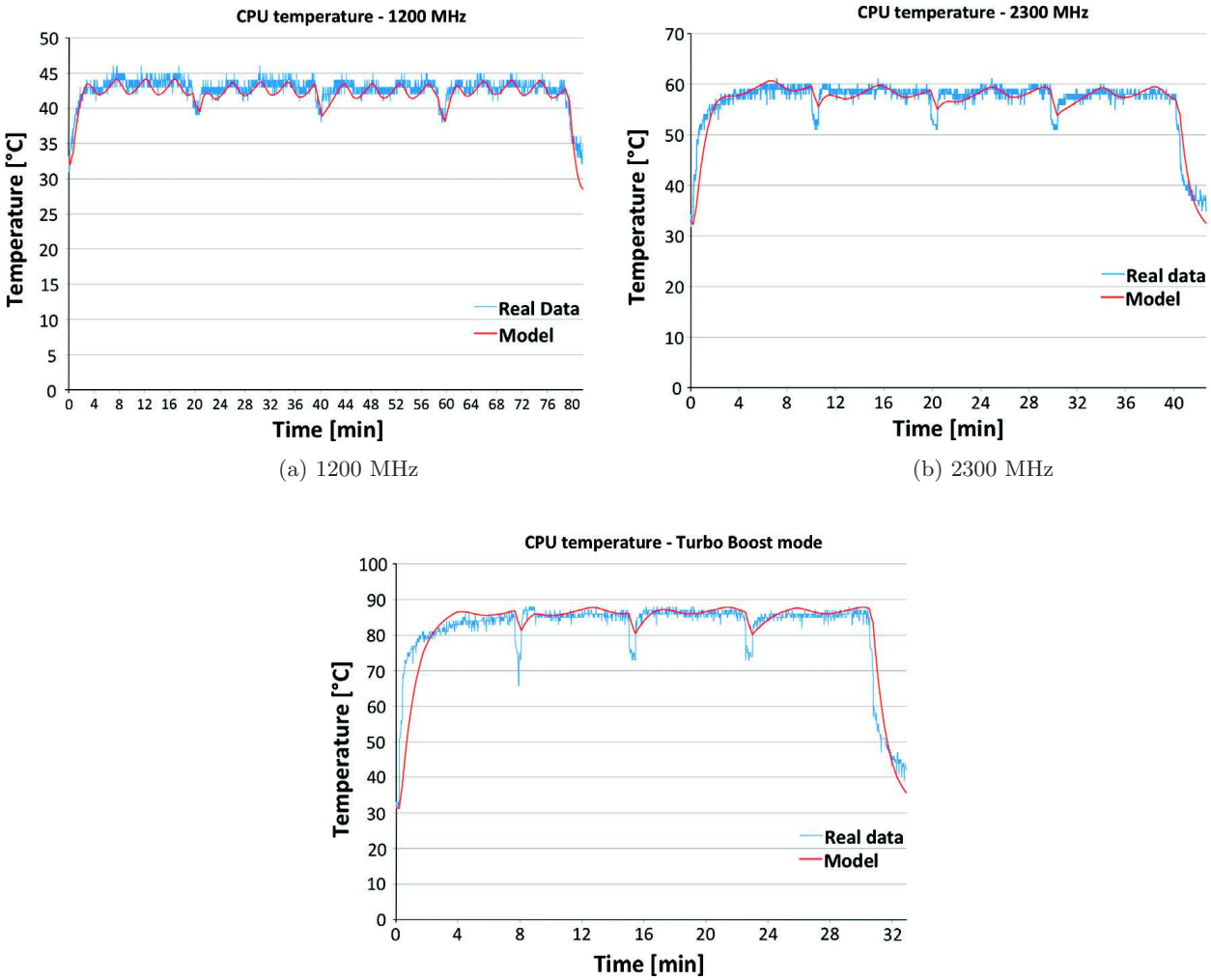
Numerical accuracy of the model is presented in Table 6.

Table 1
Thermal resistance and capacitance for Intel i7 nodes.

Parameter	Symbol	Value
Thermal resistance	R	0.99 °C/W
Thermal capacitance	C	40.1 W s/°C

Table 2
Accuracy of processor temperature model.

Frequency	Avg. temperature difference (°C)	Avg. relative error (%)
1200 MHz	0.95	2.32
2300 MHz	1.35	2.58
Turbo Boost mode	2.80	3.94



(a) 1200 MHz

(b) 2300 MHz

(c) Turbo Boost mode

Fig. 4. Comparison between measured and simulated values for processor temperature.

With respect to the results, it can be seen that taking into account the fan speed in thermal convective resistance calculations leads to around 5% of relative error. The highest temperature differences can be observed of the fan operating at 30% of its maximum speed. In this case, we noticed also the biggest dissimilarity in terms of a curve shape. For other fan speed levels the nature of the temperature evolution is reflected properly.

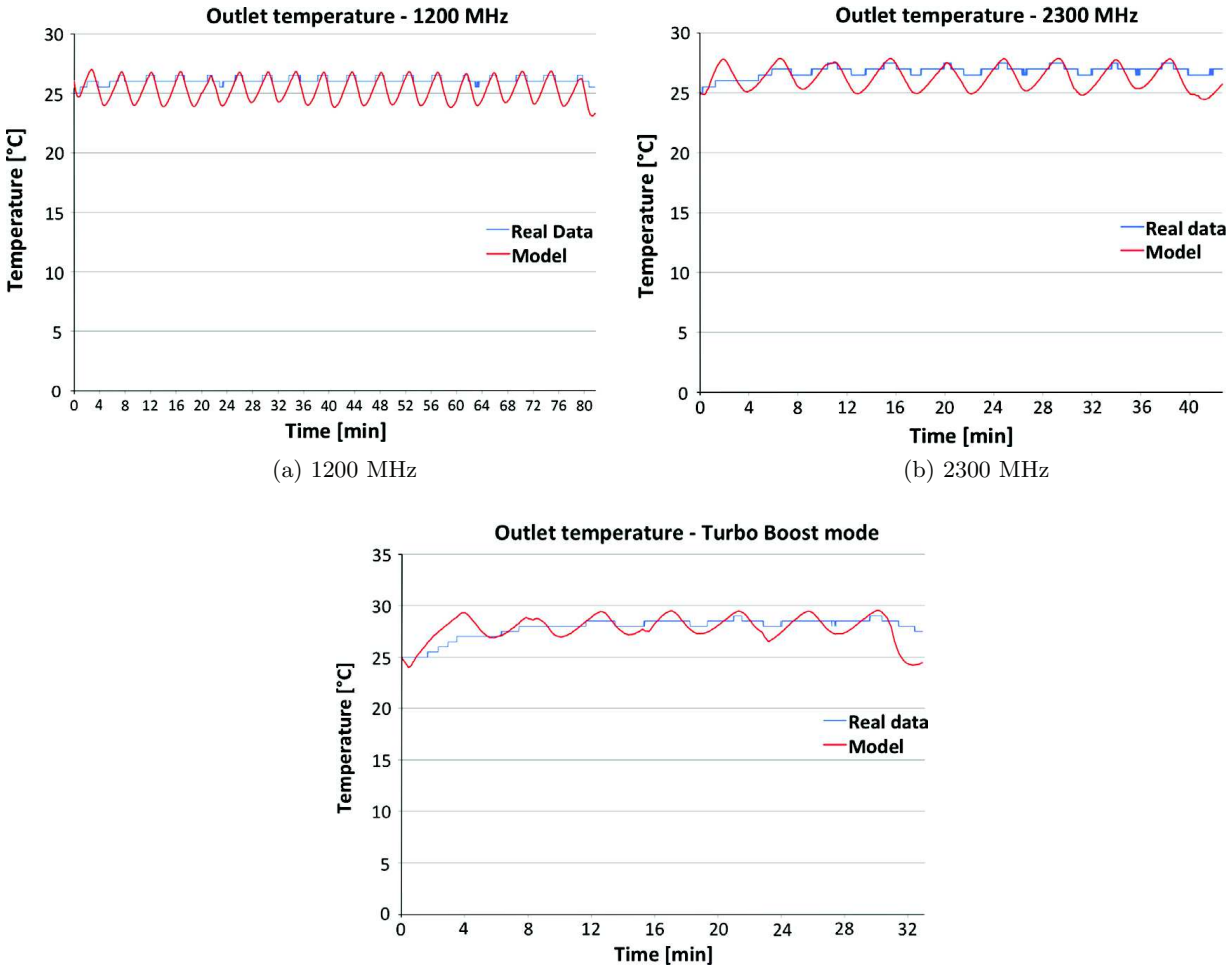
5. Simulation experiments

In this section we show, how the proposed models can be used in the simulations, in particular of the workload and resource management process. To this end we used the Data Center Workload and Resource Management Simulator (DCworms) [18] along with the power usage and thermodynamic models proposed in previous sections. We performed experiments in order to get insights into energy- and thermal-aware management policies.

5.1. Resource characteristics

In general, DCworms allows modeling various architectures, ranging from a single server up to a whole data center. For our simulation purposes, we modeled a server room with 10 racks. Each rack contains 40 1-unit enclosures, each one equipped with 4 nodes with a processor belonging to Intel Core i7-3615QE family. Such processors allow running 8 threads simultaneously. Fig. 7 shows the architecture of a single enclosure.

Below we give an overview of the power characteristic of the particular components: Processors (in Table 7), Nodes (in Table 8), Fans (in Table 9), and Cooling Facilities (in Table 10). The Power supply efficiency was assumed to be 87%. All these characteristics correspond to the power profiles created with respect to the taken measurements as well as to the vendors' guidelines. Additionally, in Table 11 we summarized environmental conditions we applied to our simulations.



(a) 1200 MHz

(b) 2300 MHz

(c) Turbo Boost mode

Fig. 5. Comparison between measured and simulated values for outlet temperature.

Table 3
Accuracy of outlet temperature model.

Frequency	Avg. temperature difference (°C)	Avg. relative error (%)
1200 MHz	0.92	3.54
2300 MHz	0.82	3.07
Turbo Boost mode	0.83	3.03

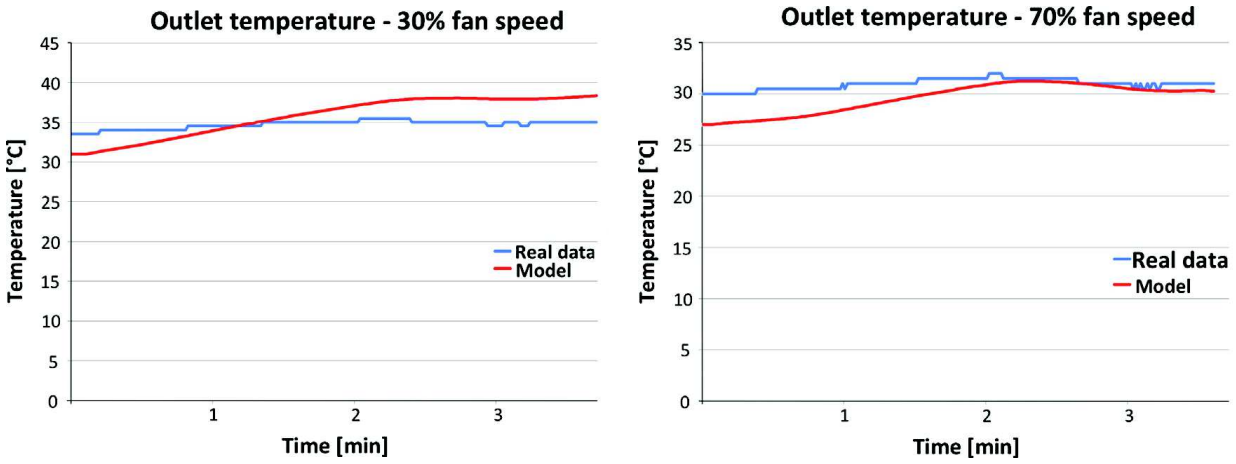
Table 4
Thermal characteristics for ARM nodes.

Parameter	Symbol	Value
Conductive thermal resistance	R_{cond}	1.69 °C/W
Convective thermal resistance	R_{conv}	0.83–1.79 °C/W
Heatsink constant	k_v	21.1 W s/°C m^3
Heatsink airflow factor	n	0.6382
Thermal capacitance	C	30 W s/°C

Detailed characteristics concerning processor power profiles can be found in [22]. As mentioned, i7-3615QE processor can operate in a Turbo Boost mode, drawing at the same time around 50 W. In our simulation experiments we disabled that P-State due to its inefficiency (the ratio between performance and power) and low predictability (frequency in Turbo Boost mode may vary up to 3300 MHz, but it cannot be anticipated). More details of i7-3615QE power behavior can be found in [22]. Thus, maximum frequency of processor is set to 2300 MHz. Hence, we also modified the node power range,

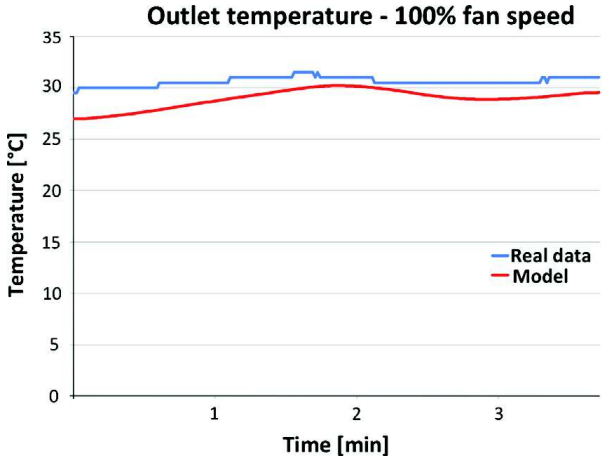
Table 5
Convective thermal resistance for different levels of fan speed.

Fan speed (%)	Convective thermal resistance (°C/W)
30	1.796
40	1.495
50	1.296
60	1.154
70	1.046
80	0.96
90	0.891
100	0.833



(a) 30% fan speed

(b) 70% fan speed



(c) 100% fan speed

Fig. 6. Comparison between measured and simulated values for outlet temperature.

Table 6
Accuracy of outlet temperature model.

Fan speed (%)	Avg. temperature difference (°C)	Avg. relative error (%)
30	2.07	5.96
70	1.47	4.46
100	1.65	5.4

decreasing maximum consumption to 50 W. Moreover, we simplified the node power consumption model from Eq. (7) by replacing network and memory part by a single value, ranging between 8 and 20 W and depending linearly on the load. Additionally, we applied a power penalty to the processor due to the increase in temperature. Such penalty corresponds to the power leakage as defined in [2]. We assumed that each °C increase in the processor temperature (over the idle state)

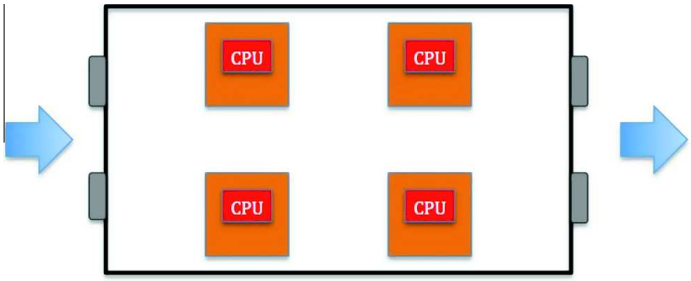


Fig. 7. Server model.

Table 7
Processor characteristics.

State	Power (W)
Idle	4
Fully loaded	30

Table 8
Node characteristics.

State	Power (W)
Idle	12
Fully loaded	50

Table 9
Single fan characteristics.

Fan speed (%)	Airflow (m^3/s)	Power (W)	Up threshold	Down threshold
30	0.00336	0.1782	–	36 °C
60	0.00672	1.4256	40 °C	44 °C
80	0.00896	3.3792	48 °C	52 °C
100	0.0112	6.6	56 °C	–

Table 10
Cooling facilities characteristics.

Parameter	Value
Cooling capacity rated	140,000 W
Energy efficiency ratio rated	3
Efficiency of cooling coil	0.95
Data center fans efficiency	0.6
Temperature difference between T_{ev} and T_{R_in}	10 °C

Table 11
Environmental conditions.

Parameter	Value
Room temperature	21 °C
Air heat capacity	1024 J/kg °C
Air pressure	1013.25 Pa
Air density	1.168 m^3/s

also leads to 2% increase of power leakage (as in [37]). Finally, we assumed that fan has 4 working states (as presented in Table 9). Last two columns indicate the fan management mechanism. Increasing the fan speed is triggered when exceeding the power threshold from the penultimate column, while decreasing occurs if the temperature falls below the value in the last column.

5.2. Workloads and application profiles

In our experiment we evaluated a workload consisting of 1280 tasks and three applications previously executed and profiled on our testbed, namely HPL, EP (Embarrassingly Parallel) [27] and Openssl [28]. We summarize their profiles in Table 12. All tasks arrived within 613 s according to the Poisson distribution.

5.3. Simulation schema

Fig. 8 shows the general steps taken by DCworms during the simulation and performed in order to perform all necessary calculations.

Table 12
Application characteristics.

Name	Duration (2300 MHz) (s)	Load (%)	Nr of threads	% in a workload
HPL	1800	98	8	50
EP	720	83	8	33
Openssl	180	12.3	1	17

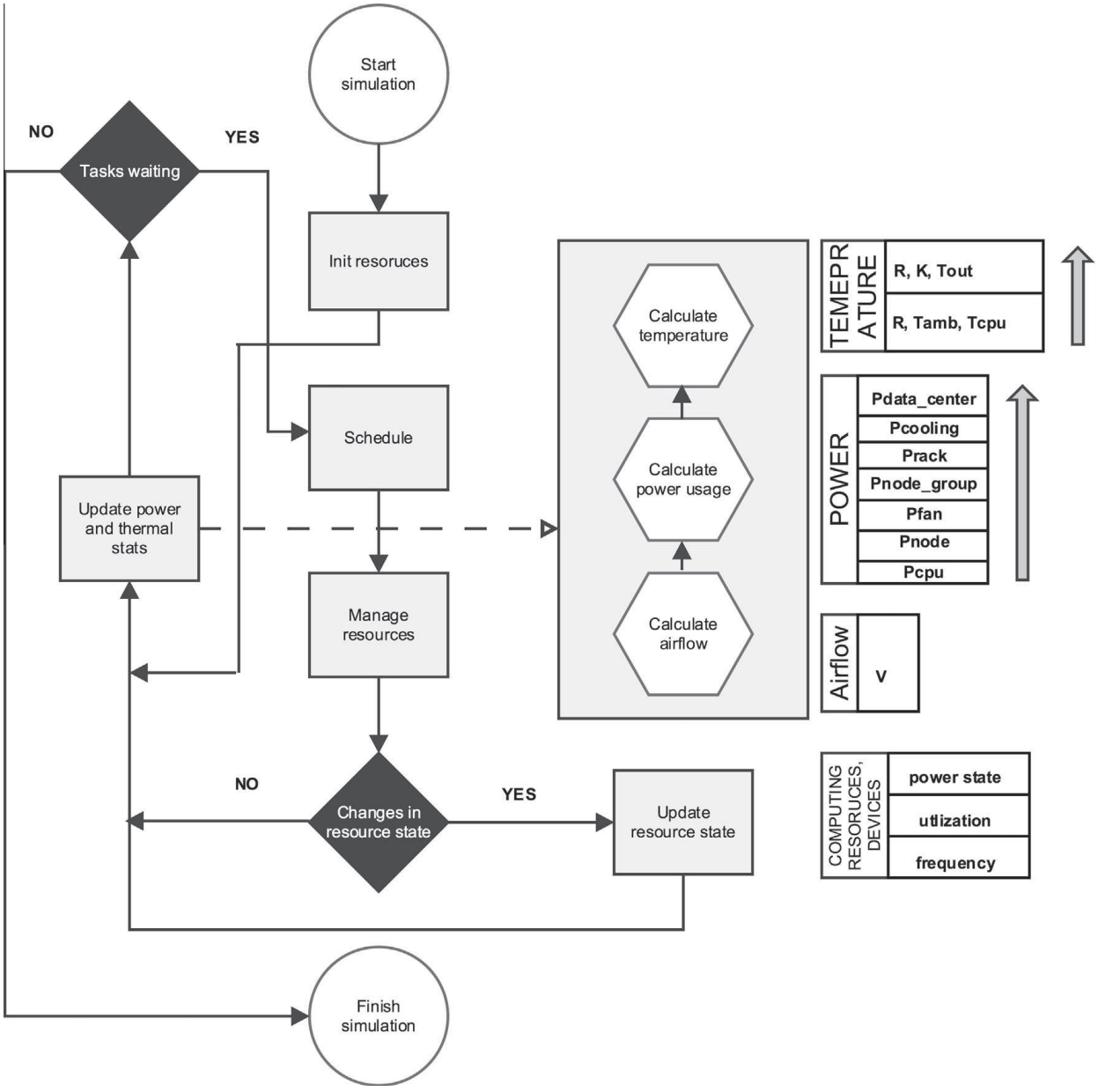


Fig. 8. Simulation schema.

5.4. Applying models to workload and resource management policies – experimental results

The ultimate goal of our work is to improve energy efficiency of computing systems. To achieve this, the proposed models should be applied by energy- and thermal-aware workload and resource management policies. To demonstrate it, we evaluated the following three different resource management strategies on resources and application set defined in previous sections.

5.4.1. Load Balancing (LB)

Tasks are assigned to nodes in order to balance the load and fans are working at full speed. The results obtained from the execution of the given workload for this policy were baseline for comparison with other policies. Results for this and other policies are presented in Table 13. Values in brackets represent energy usage until the moment represented by mean task completion time (mct) and calculated as follows:

$$energy\_mct = energy - (makespan - mct) * mean\_power$$

As in our experiments we do not switch unused nodes off, this indicator mitigates the impact of single tasks running as last. It might be especially helpful when considering strategy using power capping approach, presented as the third policy.

5.4.2. Load Balancing with Fans Management (LB + FMNG)

The same policy as in the case of Load Balancing policy but the speed of fans is adjusted according to Table 9. The goal of this experiment was to show how a proper fan management can bring energy savings, reduced power usage, and impact on a cooling system. The results demonstrate that on one hand optimized fan management is needed on a server (enclosure) level and, on the other hand global policies that lead to reduction of fan operation may bring further savings. The results are presented in Table 13. Improvements against the basic load balancing algorithm with constant fan speed are clearly visible. Reduction of energy consumption by fans is around 50%, which transformed into 10% decrease of total energy consumption by all racks. This impact of fan operation mode on overall energy consumption was also taken into account in the policy using power capping approach, presented in the next paragraph.

5.4.3. Load Balancing with Fans Management and Power Capping (LB + FMNG + PC)

This policy adds to Load Balancing with Fans Management the power capping mechanism. It is implemented as additional frequency downgrading in order to keep the power usage below the given threshold. It is based on our approach presented in [29]. We apply this technique (with additional consideration of fans power) at the enclosure level to determine the power levels forcing frequency adjustment. Thus, we got (we took into account fully loaded node with processors operating at lowest frequency):

- power cap level: $PC = \sum_{i=1}^{n} P_{CPU_i}(P_{h_i}, 100\%) + \sum_{j=1}^{m}(P_{mem_j}(100\%) + P_{net_j}(100\%)) + \sum_{k=1}^{l} P_{fan_k}$ and then: $PC = 4 * 20.5 + 4 * 20 + 4 * 6.6 = 188.4$
- power upgrade level:

$PU = PC \cdot \frac{\sum_{i=1}^{n} P_{CPU_i}(P_{h_i}, 100\%) + \sum_{j=1}^{m}(P_{mem_j}(100\%) + P_{net_j}(100\%)) + \sum_{k=1}^{l} P_{fan_k}}{\sum_{i=1}^{n} P_{CPU_i}(P_{h_i - 1}, 100\%) + \sum_{j=1}^{m}(P_{mem_j}(100\%) + P_{net_j}(100\%)) + \sum_{k=1}^{l} P_{fan_k}}$ and then: $PU = 188.44 * \frac{188.4}{4*22.5+4*20+4*6.6} = 180.7$.

Hence, we set the following limits: we downgrade the processors frequency to not exceed 190 W per enclosure and upgrade (if possible) if the power usage falls below 180 W. As shown in [29] power capping may cause even increased energy consumption by IT equipment due to longer times of execution. However, limited power usage and heat dissipation make the overall energy use smaller due to lower consumption of the cooling system (especially if power capping is introduced along with increased temperature of air supplied from the cooling system). Additionally, in long term it should result in a lower number of hardware failures and longer hardware lifetime. Another gain is potential lower investment cost if smaller chiller can be applied [29]. Table 13 shows that although total IT equipment energy consumption is slightly higher compared to both LB and LB + FMNG policy, corresponding value for energy consumption at the moment of mean completion time was

Table 13
Simulation results for three management strategies: LB – Load Balancing, LB + FMNG – Load Balancing with Fans Management, LB + FMNG + PC – Load Balancing with Fans Management and Power Capping. Values in brackets represent energy usage until the moment represented by mean task completion time.

Metrics	LB	LB + FMNG	LB + FMNG + PC
IT energy consumption [kW h]	28.72 (17.88)	29.32 (18.25)	30.79 (17.43)
Enclosure fans energy consumption [kW h]	6.47 (4.03)	3.21 (2.00)	3.39 (1.92)
Rack energy consumption [kW h]	40.45 (25.18)	37.39 (23.28)	39.29 (22.25)
Mean racks power [kW]	59.39	54.90	50.37
Max racks power [kW]	82.73	81.81	76.73
Mean task completion time [s]	1524	1524	1588
Mean task execution time[s]	1199	1199	1263

slightly decreased. Similar dependency can be observed for fan energy usage, while comparing the power capping strategy to LB + FMNG. This decrease is related to lower mean power usage (visible in the table) and relatively long periods of fan low speed operation. Mean power usage is less by around 10%. As the total energy us of IT part remains the same, total energy savings (coming from infrastructure part) can exceed 5% as pointed out in [29].

5.4.4. Workload and resource management policies – related work

In recent years, energy and thermal-oriented workload and resource management has become an investigated research area [33]. In terms of workload management most of the approaches follow load balancing or consolidation strategies. They assign and move tasks between resource in order to distribute the load to avoid the hotspot and exploit the cooling equipment more efficiently and/or to make the application of particular resource power-saving operations possible. These activities may include managing the power states of nodes, dynamic voltage and frequency scaling (DVFS) and fetch toggling.

Luo et al. [30] applied load balancing strategy on the resource allocation stage to limit energy costs, while meeting Service Level Agreement. In [32], authors propose the combination of load balancing based on task migration and DVFS to reduce cooling energy consumption and prevents hot spot formation. DVFS is used to adjust processor temperature according to the given threshold and, afterwards, tasks are moved in order to address the load imbalance between cores. Nevertheless, they do not consider fan management as a source of potential energy savings. Rodero et al. [31] made comparative studies of three different approaches, namely: virtual machine (VM) migration, processor DVFS and virtual machine monitor configuration (in particular pinning). Based on the analysis they propose proactive and reactive management strategies to minimize energy consumption and maximize the performance. However, they do not take into account cooling issues.

Consolidation policies have been studied together with resource management approaches aiming at switching unused nodes off. In [34] authors discussed the consolidation through live VM migrations that takes into account both performance and energy costs. In [21] combination of consolidation (on low power or high performance nodes) is combined with switching unused nodes off is discussed. Additionally, the impact of different task allocation policies on outlet temperature is analyzed.

In terms of fan management, Wang et al. [35] presented a fan controller that utilizes thermal models to manipulate the operation of fans. Taking into account the prediction of server temperatures, controller adjusts the speed of particular fans. In [36] authors studied the relationship between leakage and temperature of a server and provided corresponding empirical model. Based on it, they designed a controller that tunes the fan speed to minimize the energy consumption for a given workload. However, they did not expand their studies beyond a single server level to evaluate the energy-efficiency of the whole data center.

Both Moore et al. [7] and Tang et al. [6] skipped the impact of fans in their thermal considerations. Although they noticed the impact of airflow on the outlet temperature, they did not control the fan speed as a mean to improve energy-efficiency. They put their attention on reduction of heat recirculation, and thus improvement of cooling system performance by increasing the temperature of supplied air.

6. Conclusions and future work

In this paper we proposed models and the simulation environment capable to address the problem of simulation of large scale computing systems with a special focus on energy-efficiency and thermal aspects. First of all, we presented easy to calculate yet sufficiently accurate models of power usage and temperature of air-cooled servers. Comparison of models with real measurements showed mean errors lower than 4% (excluding difficult to predict Intel Turbo Boost most, even below 3%, which equals to differences around 1 °C). Proposed models enabled us to perform transient simulations of dynamic behavior of computing systems. Such information was applied to improve workload and resource management policies so they dynamically react to temperature and overall energy consumption by a computing system. In particular, we showed an effect of the fan management on energy consumption which reduced the total energy consumption by 10%. We also demonstrated how these models can be used to reduce maximum power and heat dissipation (which leads to decreased energy consumption of the cooling infrastructure: at least around 5%, see [29]) without increase in executing times (and in consequence energy consumption) of servers. Finally, we showed how characteristics of temperature changes in time, included in the models, can be taken into account in management methods. All the experiments were conducted using the DCworms simulator, enhanced with models and management policies presented in this paper.

An adequate study of future extreme scale systems requires scalable simulation methods and tools that include important new trends and possible future technologies. The important aspects of such future large scale systems are their energy-efficiency and thermal characteristics. Therefore, simulation tools addressing these issues are essential to propose new management algorithms and architectural approaches dealing with extreme scale challenges. In this paper we propose a contribution to these challenges, however there is still many research topics to investigate to meet these objectives.

First of all, further experiments with large test cases are needed to check scalability of models, simulation tools, and management algorithms. Another clear next step is proposing more advanced scheduling and resource management algorithms taking advantage of concepts presented in this paper, namely optimization of fans work, the use of power capping to limit cooling system energy consumption, and taking into account a pace of temperature changes in time to optimize workload schedules. These approaches enable application of algorithms with on-line reactions on temperature or with a dynamic cooling control. Their impact on the cooling system energy use should be studied similarly as it was done for power capping in

[29]. The topic, we also plan to study, is a simulation of the direct liquid cooling approach. Nowadays, this is an emerging technology, crucial for keeping systems of large scale and density efficient. According to our first studies, models presented in this paper can be relatively easily (apart from certain specific characteristics) extended to cover this cooling approach. Finally, we would like to apply more specific application (energy and thermal) profiles and possible future hardware architectures to simulations.

Acknowledgements

The results presented in this paper are partially funded by a grant from Polish National Science Center under Award Number 2013/08/A/ST6/00296. The work presented in this paper was also supported by the COST Action IC1305, 'Network for Sustainable Ultrascale Computing (NESUS)'.

References

[1] W. Huang, Hotspot: A Chip and Package Compact Thermal Modeling Methodology for VLSI Design PhD Dissertation, Electrical and Computer Engineering, University of Virginia, 2007.
[2] M. Patterson, The effect of data center temperature on energy efficiency, in: Proc. ITHERM, 2008, pp. 1167–1174.
[3] J. Yang, X. Zhou, M. Chrobak, Y. Zhang, L. Jin, Dynamic thermal management through task scheduling, in: ISPASS, 2008, pp 191–201.
[4] Jungsoo Kim, Mohamed M. Sabry, David Atienza, Kalyan Vaidyanathan, Kenny Gross, Global fan speed control considering non-ideal temperature measurements in enterprise servers, in: Proceedings of the Conference on Design, Automation & Test in Europe (DATE '14), 2014.
[5] Bing Shi, Ankur Srivastava, Dynamic Thermal Management Considering Accurate Temperature-Leakage Interdependency, Cooling of Microelectronic and Nanoelectronic Equipment, 2014, pp. 43–67.
[6] Q. Tang, S.K.S. Gupta, D. Stanzione, P. Cayton, Thermalaware task scheduling to minimize energy usage of blade server based, in: IEEE DASC'06, October 2006.
[7] J. Moore, J. Chase, P. Ranganathan, R. Sharma, Making scheduling cool: temperature – aware workload placement in data centers, in: Proceedings of the 2005 USENIX Annual Technical Conference, June 2005.
[8] J. Siriwardana, S.K. Halgamuge, T. Scherer, W. Schott, Minimizing the thermal impact of computing equipment upgrades in data centers, Energy Build. J. (2011) (November, Elsevier).
[9] Y. Joshi, Reduced order thermal models of multiscale microsystems, J. Heat Transfer Trans. ASME 134 (3) (2012).
[10] Rajat Ghosh, Yogendra Joshi, Rapid temperature predictions in data centers using multi-parameter proper orthogonal decomposition, Numer. Heat Transfer, Part A: Appl. 66 (2014) 41–63.
[11] http://www.electronics-cooling.com/2011/09/real-time-data-center-cooling-analysis/.
[12] Robert Basmadjian, Nasir Ali, Florian Niedermeier, Hermann De Meer, Giovanni Giuliani, A methodology to predict the power consumption of servers in data centres, in: Proc. of the ACM SIGCOMM 2nd Int'l Conf. on Energy-Efficient Computing and Networking (e-Energy 2011), ACM, 2011.
[13] http://www.intel.com/content/dam/doc/design-guide/xeon-3000-thermal-design-guide.pdf (last accessed 05.11.14).
[14] http://christmann.info/show/57 (last accessed 10.11.14).
[15] http://people.seas.harvard.edu/~apw/stress/ (last accessed 11.11.14).
[16] http://www.netlib.org/benchmark/hpl/ (last accessed 11.11.14).
[17] Christine S. Chan, Yanqin Jin, Yen-Kuan Wu, Kenny C. Gross, Kalyan Vaidyanathan, Tajana Simunic Rosing, Fan-speed-aware scheduling of data intensive jobs, ISLPED (2012) 409–414.
[18] K. Kurowski, A. Oleksiak, W. Piatek, T. Piontek, A. Przybyszewski, J. Weglarz, DCworms – a tool for simulation of energy efficiency in distributed computing infrastructures, Simulat. Model. Pract. Theory 39 (2013) 135–151. ISSN 1569-190X. http://dx.doi.org/10.1016/j.simpat.2013.08.007.
[19] R.Z. Ayoub, R. Nath, T. Rosing, JETC: joint energy thermal and cooling management for memory and CPU subsystems in servers, in: Proceedings of HPCA, 2012, pp. 299–310.
[20] R. Jorgensen, Fan Engineering; An Engineer's Handbook on Fans and Their Applications, ninth ed., Buffalo Forge Company, 1999.
[21] L. Cupertino, G. Da Costa, A. Oleksiak, W. Piatek, J.M. Pierson, J. Salom, L. Siso, P. Stolf, H. Sun, T. Zilio, Energy-efficient, thermal-aware modeling and simulation of data centers: the CoolEmAll approach and evaluation results, Ad Hoc Netw 25 (B) (2015) 535–553.
[22] D2.3.1 Update on Definition of the Hardware and Software Models – CoolEmAll Deliverable. <http://coolemall.eu>.
[23] S. Rivoire, P. Ranganathan, C. Kozyrakis, Acomparison of high-level full-system power models, in: HotPower, USENIX Association, 2008.
[24] M. Pedram, I. Hwang, Power and performance modeling in a virtualized server system, in: Proceedings of the 2010 39th International Conference on Parallel Processing Workshops, 2010, pp. 520–526.
[25] M. Allalouf, R. Kat, Y. Arbitman, K. Meth, M. Factor, D. Naor, Storage modeling for power estimation, in: SYSTOR 2009 The Israeli Experimental Systems Conference, Haifa Israel, 2009.
[26] A. Vishwanath, K. Hinton, R.W.A. Ayre, R.S. Tucker, Modeling energy consumption in high-capacity routers and switches, IEEE J. Sel. Areas Commun. 32 (8) (2014) 1524–1532 (01).
[27] http://www.phy.duke.edu/rgb/Beowulf/beowulf_book/beowulf_book/node30.html (last accessed 13.11.14).
[28] https://www.openssl.org/ (last accessed 14.11.14).
[29] G. Da Costa, A. Oleksiak, W. Piatek, J. Salom, L. Siso, Minimization of costs and energy consumption in a data center by a workload-based capacity management, in: 3rd International Workshop on Energy-Efficient Data Centres Co-located with the ACM e-Energy, 2014.
[30] Jianying Luo, Lei Rao, Xue Liu, Temporal load balancing with service delay guarantees for data center energy cost optimization, IEEE Trans. Parall. Distrib. Syst. (TPDS) 25 (3) (2014) 775–784.
[31] Ivan Rodero, Hariharasudhan Viswanathan, Eun Kyung Lee, Marc Gamell, Dario Pompili, Manish Parashar, Energy-efficient thermal-aware autonomic management of virtualized HPC cloud infrastructure, J. Grid Comput. 10 (3) (2012) 447–473.
[32] Osman Sarood, Phil Miller, Ehsan Totoni, Laxmikant V. Kale, 'Cool' load balancing for high performance computing data centers, IEEE Trans. Comput. 61 (12) (2012) 1752–1764, http://dx.doi.org/10.1109/TC.2012.143.
[33] Anne-Cecile Orgerie, Marcos Dias de Assuncao, Laurent Lefevre, A survey on techniques for improving the energy efficiency of large-scale distributed systems, ACM Comput. Surv. 46 (4) (2013) 47.
[34] Mohammad M. Hossain, Jen-Cheng Huang, Hsien-Hsin S. Lee, Migration energy-aware workload consolidation in enterprise clouds, CloudCom (2012) 405–410.
[35] Z. Wang, C. Bash, N. Tolia, M. Marwah, X. Zhu, P. Ranganathan, Optimal fan speed control for thermal management of servers, in: Proceedings of the ASME, 2009.
[36] M. Zapater, J.L. Ayala, J.M. Moya, K. Vaidyanathan, K.C. Gross, A.K. Coskun, Leakage and temperature aware server control for improving energy efficiency in data centers, in: DATA, 2013, pp. 266–269.
[37] F. Fallah, M. Pedram, Standby and active leakage current control and minimization in CMOS VLSI circuits, in: IEICE Transactions on Electronics, 2005, pp. 509–519.