2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing

Optimized thermal-aware workload distribution considering allocation constraints in data centers

Hassan Shamalizadeh[#], Luis Almeida[*], Shuai Wan[*], Paulo Amaral[*], Senbo Fu[+], Shashi Prabh[*]

[#] IEETA / DETI
Universidade de Aveiro
3810-193 Aveiro, Portugal
shamalizadeh@ua.pt

[*] IT / Fac. de Engenharia
Universidade do Porto
4200-465 Porto, Portugal
{lda,shuaiw}@fe.up.pt,
shashi.prabh@gmail.com

[+] Electrical and Computer Eng
Carnegie Mellon University
Pittsburgh, PA 15213, USA
senbof@andrew.cmu.edu

***Abstract*— Power management has been increasingly critical for sustainable datacenters. One particular aspect that has a strong impact on the power consumed by a data center is how the workload is distributed among its servers. This distribution can be done integrating thermal models that allow balancing cooling needs with computing needs contributing to reduce overall power consumption. In this paper, we present a workload distribution optimization method for homogeneous server environments that minimizes total heat recirculation. We use a parameter to constrain the total contribution of each node to the recirculated heat and we show that such parameter allows fine-grained control over the number of needed servers and consequently over the balance between IT computing power and cooling power needs. Additionally, we incorporate allocation constraints, representing cases where specific workloads must be allocated to a specific subset of servers only, which for example, result from Service-Level-Agreements with datacenter customers. These constraints are often found in reality but have seldom been considered in the literature. We carry out simulation experiments using measurement data provided by the Bluesim tool [20]. The results show the effectiveness of the proposed approach in controlling the active servers, thus total power, needed for a given workload while meeting allocation constraints.[1]**

Keywords- Thermal model, workload placement, optimization

I. INTRODUCTION

Over the last decade, datacenters have become the mainstream infrastructure to provide IT services to enterprises and users all over the world, such as Cloud Computing and Internet-based services [1]. These datacenters typically include hundreds of physical servers, frequently virtualized into a myriad of virtual machines, each dedicated to one particular application or customer, with added benefits of better maintainability and service flexibility [2]. Virtualization allows higher utilization of the datacenter leading to increased energy consumption, as well as higher rates of greenhouse gas emissions. Nevertheless, the electricity consumed by the datacenter servers, i.e., the IT component, typically represents only about half of the total energy used by a datacenter [3].

Therefore, it is increasingly important to run datacenters in an energy-efficient way while still meeting customers Service Level Agreements (SLAs), i.e., to use the so-called green strategies. This has motivated a substantial research effort in the last decade [19] covering a wide diversity of datacenters operational aspects [17].

In this paper we focus on homogeneous virtualized datacenters running interactive workloads, such as those typically handled by web servers, databases, etc. We further consider that such workload encompasses many applications, e.g. web services, each one executing within one virtual machine but possibly instantiated several times according to an SLA, to improve availability and reduce response times. These applications behave in a quasi-continuous mode, given the typical high rate of requests. However, despite the variability of web requests arrival, the peak rate that is considered in the SLA is typically constant or changes in fairly long intervals thus making planning actions, such as workload allocation, more persistent. In our case, given that the proposed allocation method, which is based on binary linear programming (BLP), executes in the sub-second range for a relatively small datacenter, we claim that such method is potentially adequate for online use in the referred scenario.

Our work builds upon the line pursued by Gupta *et al.* [4][8] and [17] on thermal-aware workload allocation but focusing on interactive workloads (IDC) instead of high performance computing (HPC). For the workload characterization we build upon the work of Petrucci *et al.* [13] that showed that interactive workloads also have a linear power footprint with respect to the requests rate. We test our work with simulation experiments using one example datacenter from [17] together with its heat recirculation pattern provided by the Bluesim tool [20]. Our contributions are the following:

- Optimization BLP-based thermal-aware workload allocation controlled by a specified parameter, namely a bound on the contribution of each node to the total heat recirculation, named recirculation bound, which determines the concentration/ dispersion of the workload, thus determining the number of active servers needed to run such workload and consequently, the IT power;
- A simple but flexible and effective way of incorporating allocation constraints in the allocation optimization procedure.

[1] This work was partially supported by FEDER through the COMPETE program, and by the Portuguese Government through FCT grant SENODS - CMU-PT/SAI/0045/2009;

978-0-7695-5046-6/13 $26.00 © 2013 IEEE
DOI 10.1109/GreenCom-iThings-CPSCom.2013.55

The former contribution directly addresses the criticality of controlling the inlet temperatures for the effective performance of the datacenter [14]. Note that heat recirculation is the cause for the increase of inlet temperatures over that of the cooling air.

The latter contribution allows accounting for constraints derived from the SLAs with the datacenter customers that force certain workloads to execute in a subset of the physical servers, only. For example, a customer may rent dedicated servers for his applications or even provide his own. The same technique can also be used to control which servers should be used or shut down. This way, the same optimization process can be used to generate different allocation profiles such as MCE or min-HR presented previously in the literature [16].

The allocation constraints and specified recirculation bound give the datacenter operator a simple but effective way of controlling the computing power needed to execute a given workload as well as the supply cooling air temperature needed to maintain all operating servers within their desired operational temperature ranges.

The remainder of the paper is organized as follows: the next section briefly reviews some related work, Section 3 presents the models that we consider in our work, Section 4 presents the proposed optimization framework and Section 5 shows experimental results. Finally Section 6 concludes this paper and refers to possible future work.

II. Related Work

Computational fluid dynamics (CFD) is regarded as a popular and conventional approach to simulate and estimate the temperature evolution within datacenters [5] [6]. It has also been used to distribute the workload over selected servers and racks [7]. However, it is complex and time consuming for most practical scenarios, and is unsuitable for on-line use.

Some previous works have focused on simplified thermal modeling of the datacenter using steady-state conditions validated with CFD analysis [8]. Others used sensor-based thermal mapping by continually monitoring temperature [9][10]. On the other hand, workload distribution based on optimized thermal and power models has also been researched regarding temporal job scheduling and spatial task balancing in order to achieve overall energy-efficiency [11] [12] [13].

A rather complete research tool available today is GDCSim [17], encompassing automated processing, online analysis, iterative design, thermal analysis, workload management and cyber-physical interdependencies. The same work presents an interesting survey of the state-of-the art in datacenter design and management. Concerning workload allocation, it considers three main classes namely: a) rank-based, in which the servers are ranked and the allocation follows the ranking, e.g. using FCFS [12]; b) control-based, in which the allocation is guided by a controlling parameter in a closed loop to minimize a certain metric, e.g. [18]; and c) optimization-based, in which the allocation results from solving an optimization problem, e.g. XInt [16].

Our approach falls within the optimization-based class but differs from other thermal-aware workload allocation methods by providing one single parameter that the datacenter operator can use to drive the optimizer and tune the number of needed active servers while also deciding on a thermal margin below the critical inlet temperature of the servers. The choice of servers that are kept active is automatic, following a rank-based approach inside the optimization process that uses common heuristics [12] but may include specific resource allocation constraints according to SLAs, a feature not commonly considered in the literature.

In particular, our approach resembles the MCE and Min-HR algorithms described in [16] but, beyond the differences referred above, we use a transactional IDC load model as opposed to the HPC model used therein.

III. Underlying Models and Assumptions

A datacenter is generally a warehouse with rows of racks and a series of complex computer system facilities such as servers, storages and networking equipment, management controllers, and other electrical and cooling infrastructures. Typically, most datacenters adopt the hot/cold aisle layout as shown in Figure 1, where each row is placed between a hot and a cold aisle. The cold air supplied by the Computer Room Air Conditioning (CRAC) units passes through the perforated tiles of an elevated floor and picks up the hot air heated by the servers, which is then captured by the intakes of the CRAC placed at the end of or on the ceiling of the hot aisles.

A. Datacenter configuration

As referred before, we will use the smaller case study in [17] which is a datacenter with 2 rows of 2 35U racks each, arranged in a typical hot-aisle/cold-aisle configuration, and each rack containing 5 7U chassis with 10 blade servers each, namely model IBM Series 350M2 with idle power of 100W and peak power consumption of 300W. Cold air is supplied at a flow rate of 5 m^3/s from one CRAC system through floor vent tiles and the air re-enters through ceiling vent tiles.

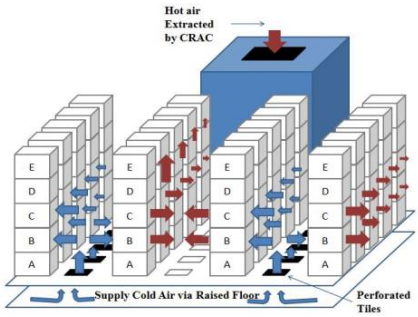


Figure 1. Hot-aisle/cold-aisle datacenter organization

B. Computing nodes and performance model

The referred data center contains 200 servers organized in 20 chassis. Allocating the workload per server is a heavy task for most allocation tools, particularly for those based on optimization processes. Thus, our workload allocation tool considers each chassis one computing unit with all its servers switched on or off together, depending on whether there is at least one application allocated to a server in the chassis or not.

The chassis will thus behave as a macro server with an idle power of 1000W and a peak power of 3000W and we refer to it in the remainder of the paper as *node*.

Concerning the electrical power variation of each node as a function of its computing load (P_i for node i), it has been previously established that such relationship can normally be considered linear with the processor utilization plus an idle power level p_{idle} [15]. This model also applies to interactive workloads with the server power varying linearly with the respective services request rate [13]. For each application there is a relationship between the request rate and power consumed by the respective node that must be determined by profiling. Here we consider that such relationship is already known and we represent our workloads by the increment in power they cause to the node where they are allocated for the request rate considered in the respective SLA. We call W_j the power footprint of application j. Note that since we consider a homogeneous datacenter, the footprints of the applications are equal independently of the node where they are allocated.

Therefore, we can model the electrical power consumed by node i (P_i) with Equation 1 where the binary variables x_{ji} and λ_i characterize the workload distribution of the n applications among the m nodes in the datacenter. If node i runs application j then x_{ji}=1 else x_{ji}=0, and if node i has some assigned workload then $\lambda_i = 1$ else it can be switched off and $\lambda_i = 0$. In a compact form, $\mathbf{X}$ is an n by m matrix representing the workload allocation and $\boldsymbol{\lambda}$ an m-length vector representing the active nodes.

$$P_i = \sum_{j=1}^{n} x_{ji} W_j + \lambda_i p_{idle}, \quad \forall_i = \{1 \dots m\} \tag{1}$$

In a matrix form, considering P the vector with the power of all m nodes and W the vector with the power footprint of all n applications in the workload, we can rewrite Equation 1 as:

$$P = \mathbf{X}^T W + p_{idle}\, \boldsymbol{\lambda}$$

C. Thermal model

A critical issue in datacenters is the heat recirculation that warms up the air at the inlet of the chassis reducing the efficiency of the air cooling system. Such recirculation is typically determined with CFD analysis or experimentally as in [8] and it is represented by a square Heat Recirculation Matrix (HRM) where the $(i,j)^{th}$ element of the matrix represents the heat contribution of node j on node i. Here, we use the matrix provided by Bluesim for this datacenter, represented by $\mathbf{D}$, where element $D_{i,j}$ represents the temperature increment at the inlet of node i per unit of electrical power consumed by node j.

Let t_{sup} be the cooling air temperature, we can compute the vector of inlet chassis temperatures T_{in} using Equation 2. This equation also means that applying a power distribution P to the m nodes, globally cooled with air at temperature t_{sup}, their inlet temperatures will converge to T_{in} in steady state.

$$T_{in} = t_{sup} + \mathbf{D}\, P \tag{2}$$

We call the term $\mathbf{D}P$ the heat recirculation vector since it contains the increments in inlet temperature of each node above t_{sup}. Then, the total heat recirculation THR is defined as the sum of all elements in DP as in Equation (3)

$$THR = \sum_{i=1}^{m} (\mathbf{D}P)_i \tag{3}$$

Moreover, for all nodes, knowing their inlet temperatures T_{in} and the power they are consuming, P, allows deriving the respective steady state outlet temperatures T_{out} (Equation 4). In this expression, $\mathbf{K}$ is a thermodynamic diagonal matrix of dimension m, representing the product of the air density ρ, air flow rate f_i and specific heat of air C_p for each server i. For simplicity and without loss of generality, we consider k_i to be constant across all nodes using the values provided in [16].

$$T_{out} = T_{in} + \mathbf{K^{-1}}\, P \tag{4}$$

Exploring the temperature model at the node inlet level enables a more efficient operation of the datacenter [14] since the CRAC can set t_{sup} such that the maximum value of T_{in} is close to but below $t_{critical}$, which is the critical value specified by the server manufacturers (also referred as the *red line temperature*). Thus, balanced inlet temperature patterns allow increasing t_{sup} while avoiding hot spots.

D. CRAC and total power models and problem statement

As seen in the previous section, t_{sup} can be used to effectively enforce inlet temperatures to be below the red line threshold. However, to achieve this with unbalanced T_{in} patterns will require a lower value of t_{sup} because the higher $\max(T_{in})$ that results from the uneven temperatures will need to be compensated for with a lower t_{sup}.

The problem of using lower values of t_{sup} is that the lower t_{sup} the higher the energy spent by the cooling system. The efficiency of the cooling system is normally characterized by the so called Coefficient Of Performance (COP) which is defined by Equation 5. Note that the heat to be removed is basically the energy spent by the computing equipment and thus the COP can also be expressed as the ratio of the total power taken by the computing equipment P_{IT} to the power consumed by the cooling system P_{AC}. Thus, we can now express the total computing and cooling power P_{total} as in Equation 6 which highlights the impact of the COP.

$$COP = \frac{Heat\ removed}{Cooling\ energy} = \frac{P_{IT}}{P_{AC}} \tag{5}$$

$$P_{total} = \left(1 + \frac{1}{COP}\right) P_{IT} \tag{6}$$

The COP is known to vary quadratically with t_{sup} and we will herein use the model of [11], expressed in Equation 7.

$$COP = 0.0068 t_{sup}^2 + 0.0008 t_{sup} + 0.458 \tag{7}$$

A fundamental aspect of our work is the impact of the concentration or dispersion of the workload in the datacenter. On one hand, if we allocate fewer loads per node but to more nodes, the dissipated power per node will be lower so as its contribution to heat recirculation. Consequently, the inlet temperatures will also be closer to t_{sup}. In other words, we will reduce the cooling power needs because we allow a higher t_{sup} but we will increase the computing power because more servers will be active and the impact of p_{idle} will be higher.

On the other hand, if we concentrate the same workload in fewer nodes, we will be reducing the computing power but some nodes will have a stronger impact on heat recirculation

and we will need a lower t_{sup} to keep T_{in} under $t_{critical}$, thus the cooling power will increase.

Thus, the main goal of our work is to provide the datacenter operator with a knob that allows controlling the concentration / dispersion of the workload so as to find the most favorable operational point that will grant a near minimal total power consumption P_{total} while considering other operational aspects such as the actual behavior of the loads and their likelihood to overrun the peak request rate specified in the SLA. We will focus on using a specified heat recirculation bound, *h*, that constrains the contribution of each node to *THR*. Lowering the *h* bound, limits the load that a node can host and thus increases load dispersion.

E. Allocation constraints

One specific aspect that we consider in our work is the incorporation of allocation constraints in our workload distribution process. In fact, it is frequent that datacenter customers rent specific nodes or even install their own nodes in the datacenter to run their application services. In this case, the load allocation cannot freely distribute computing load among the whole set of datacenter nodes.

Therefore, we model these constraints with an *n* by *m* matrix **S**, in which the element s_{ji} specifies a sufficiently large predefined penalty, represented by *iAlloc,* when the assignment of application *j* to node *i* is illegal. Moreover, we use the same matrix to specify preferences in the allocations by using penalties that are significantly lower than *iAlloc*. The lower the s_{ji} value (penalty) the higher the probability of load *j* being assigned to node *i* and vice-versa.

The allocation preferences in the **S** matrix can also be used to encode typical allocation heuristics. For example, by assigning growing penalties to the nodes from lower to higher positions in the racks, we effectively guide the allocation to favor the lower nodes, i.e., closer to the floor, which typically suffer less from heat recirculation and thus can tolerate higher t_{sup}. Moreover, the **S** matrix is also a simple way to reduce migrations between consecutive optimization steps, as the optimizer will try to allocate each load to the same node.

IV. Optimization model

Our optimization problem consists of finding an allocation of *n* loads to *m* nodes so that it uses the least number of nodes and minimizes the total heat recirculation *THR* while respecting a specified bound on the individual contribution of each node to *THR,* called the recirculation bound *h*. This problem is similar to bin packing problems and we solve it using a BLP approach.

A. General optimization constraints

Our optimization formulation is subject to a number of constraints that are expressed below.

$$\sum_{i=1}^{m} \lambda_i \leq m \tag{8}$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ji} = n \tag{9}$$

$$\sum_{i=1}^{m} x_{ji} = 1, \quad \forall_j = \{1 \dots n\} \tag{10}$$

$$\sum_{i=1}^{m} x_{ji}\, s_{ji} < iAlloc, \quad \forall_j = \{1 \dots n\} \tag{11}$$

$$\sum_{j=1}^{n} x_{ji} \leq n * \lambda_i, \quad \forall_i = \{1 \dots m\} \tag{12}$$

$$\sum_{j=1}^{n} x_{ji} \geq \lambda_i, \quad \forall_i = \{1 \dots m\} \tag{13}$$

$$P_i \leq P_{max} \quad \forall_i = \{1 \dots m\} \tag{14}$$

$$\sum_{k=1}^{m} D(k,i)\ P_i \leq h \quad \forall_i = \{1 \dots m\} \tag{15}$$

The formulation is based on two binary variables, the allocation matrix $\mathbf{X}$ and the active nodes vector $\boldsymbol{\lambda}$, as described in Section III.B, and the whole search space has $2^{m+n} * 2^m$ possible solutions. This space is further reduced by the constraints listed above. Equation 8 enforces the limitation on the total number of nodes in the datacenter. Equation 9 represents the size of the workload. Equation 10 guarantees that each load is allocated to exactly one node. Equation 11 prevents illegal allocations present in matrix **S** to be chosen. Equation 12 guarantees that only active nodes ($\lambda_i = 1$) can host loads while Equation 13 guarantees that active nodes must hold at least one load. Equation 14 constrains the maximum number of loads that a node can hold guaranteeing that the node maximum power is respected. Finally, Equation 15 constrains to *h* the total contribution of node *i* to *THR*.

B. The BLP formulation

The target of this formulation is to minimize the Total Heat Recirculation *THR*. The objective function (Equation 16) is the result of combining Equations 1 with 3, considering the allocation constraints **S**, as well as all the constraints enumerated in the previous section. We use * to represent element-wise matrix multiplication.

Minimize

$$\sum_{i=1}^{m} \left(\mathbf{D}\ ((\mathbf{S} * \mathbf{X})^T W + p_{idle}\ \boldsymbol{\lambda})\right)_i \tag{16}$$

V. Experiments

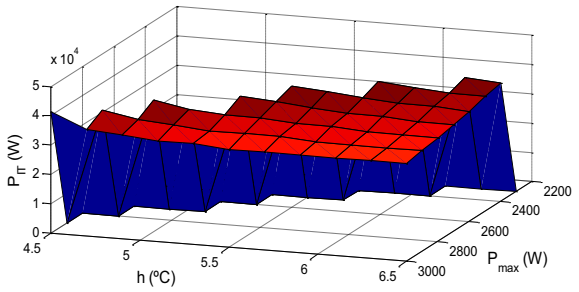
In this section we show the results of several simulation experiments to assess the effectiveness of our workload allocation method. For this purpose we generated random workloads composed of 200 applications each, with power footprints uniformly generated between 1W and 200W, i.e., up to the capacity of each server. Since there are 200 servers in the datacenter of the Bluesim dataset, our experiment configuration that it is possible to assign at least one load to each server and several workloads to each node with its 10 servers. Given the reasonable number of loads in the workload and their uniform distribution of power footprints, the aggregate power footprint imposed over the datacenter is around 10kW, which corresponds, approximately to 50% utilization of the datacenter computing capacity.

The simulations were carried out on a desktop computer with an Intel(R) Core(TM) i5-2320 CPU @ 3.00GHz, 4 Core(s), 4 Logical Processor(s), with 8GB of physical RAM. The optimization tool is Gurobi [21] running on Matlab.

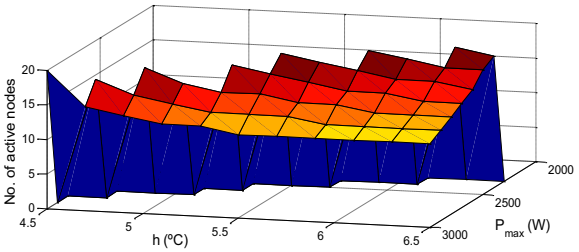
A. *Controlling the workload concentration / dispersion*

Figure 2 shows the surfaces of (a) P_{IT} and (b) number of active nodes, obtained when varying h and P_{max} independently. For each fixed value of P_{max}, i.e., the maximum power that a node is allowed to admit, particularly larger values, figure (b) confirms the effectiveness of controlling the concentration (less active nodes) or dispersion (more active nodes) of the workload when varying h, as desired. Figure 2 (a) shows the monotonically decreasing variation of P_{IT} as the workload gets more concentrated due to the growing number of nodes that get switched off.

Note that varying P_{max} has a similar effect to that of varying h. This confirms the effectiveness of power-based workload control. However, this kind of control does not consider heat recirculation and thus its potential to reduce energy is lower. Therefore, it is better to simply use the largest possible value for P_{max} (3kW in this case) and then control the workload allocation with h. In the remainder of the experiments we use P_{max}=3kW.



(a)



(b)

Figure 2. Surfaces of (a) P_{IT}, and (b) number of active nodes, as a function of h and P_{max}

B. *Minimizing joint computing and cooling power*

Figure 3 shows the variation of P_{total}, i.e., $P_{IT} + P_{AC}$, as a function of t_{sup} for three fixed different values of $t_{target} = t_{sup}+h$. In the three cases, the variation of h is the same but the absolute values of t_{sup} are different, leading to different values of P_{AC} and thus different impacts on P_{total}. Note that t_{target} is an approximation of max(Tin), since h represents the maximum heat recirculation caused by any node (not suffered).

In principle, the higher t_{target} is the most favorable, resulting in higher values of t_{sup} and thus lower values of P_{total}. However, the datacenter operator might prefer to choose a slightly lower range of t_{sup}, setting a t_{target} slightly below $t_{critical}$. This has the advantage of creating a guarding window for the case of max(Tin) possibly overrunning t_{target} of an application temporarily admitting requests above the maximum request rate specified in the respective SLA. Without such guarding window, any such situation would cause some inlet temperatures to overrun $t_{critical}$, which should be avoided, given the risk of thermal shut down of the servers. Setting the right width of this guarding window is left for the datacenter operator that must consider the possible existence of rate protection mechanisms in the virtual machines within which the applications run, and the likeliness of such overruns for each specific application.

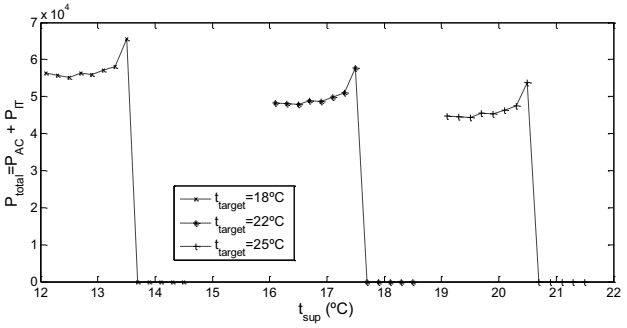


Figure 3. P_{total} as a function of t_{sup} for three different values t_{target}, considering $t_{sup} = t_{target} - h$ and similar range of h.

C. *Balancing of inlet / outlet temperatures*

A positive impact of balancing temperatures is the inherent reduction of maximum values that can create hot spots [14]. As referred before, the cooling needs are essentially determined by the maximum temperature and thus, balancing temperatures not only prevents hot spots but also reduces cooling power.

Therefore, it is important to assess how well the proposed approach addresses temperature balancing. Figure 4 shows the minimum and maximum inlet and outlet temperatures achieved with the proposed allocation method for one random workload with t_{sup}=12ºC and h varying from 3.5ºC to 6.5ºC. Note that no feasible allocations are achieved for $h \leq$4.5ºC. The constraint enforced by the recirculation bound h imposed by the allocation method (Equation 15) leads to variations in T_{in} of 2ºC to 3ºC in the range of interest. On the other hand, the outlet temperatures T_{out} accumulate the variation of T_{in} with the variations in heat resulting from the workload execution, showing thus a larger variation as expected, i.e., from 7ºC to 9ºC for the same h range.

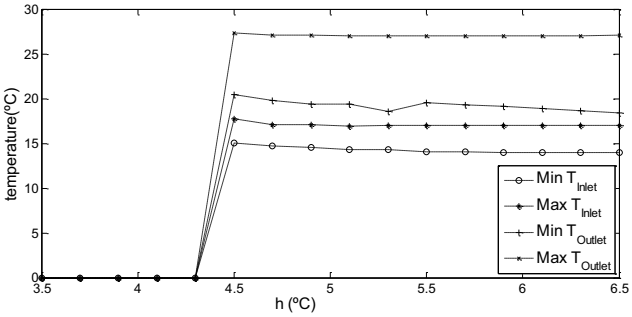


Figure 4. Balance of inlet and outlet temperatures (t_{sup}=12ºC)

D. Effectiveness of the allocation constraints

The previous subsections did not consider allocation constraints, i.e., any load could be assigned to any node. In this section we analyze the application of such constraints, thus allowing us to considering t_{sup}=17ºC and h=5ºC.

As referred in Section III.E, these constraints allow us to introduce allocation limitations in the optimization process arising from SLAs with datacenter customers and/or general preferences in the allocation of the loads.

We start by running the optimizer for a workload of 100 applications without allocation constraints, i.e., **S**=ones(n,m), as reference (Fig. 5-a). This workload uses approximately 25% of the datacenter computing capacity, a light load that facilitates the visualization of the allocation outcomes. This reference allocation also gives information about which nodes contribute less to the heat recirculation, which in this case study, is clearly rack 1 on row 1, followed by rack 3 in row 2, both on the same side as the CRAC. Note that we represent the nodes in the racks horizontally, for convenience of visualization, but they are organized vertically, from A closer to the floor, to E closer to the ceiling.

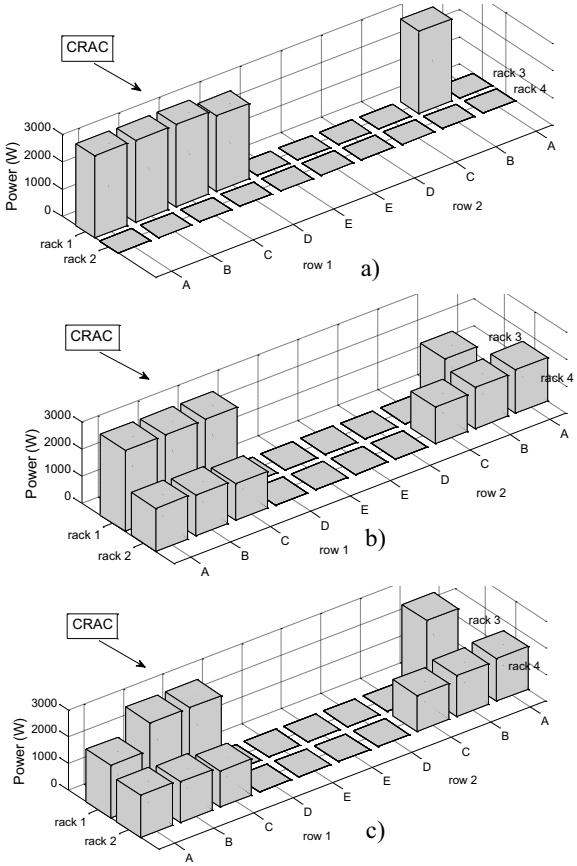


Figure 5. Controlling the allocation profile with weights in the **S** matrix; (a) no constraints, (b) forcing use of nodes with high heat recirculation and (c) incorporating fully exclusive use of nodes for certain loads as expressed in SLAs

Fig. 5-b) shows the result of using an S matrix with a weight of 10 to racks 1 and 3, and a weight of 1 to racks 2 and 4, forcing the optimizer to choose these racks that are farther from the CRAC. It is curious to see that, even with the large difference in weights, the optimizer allocated just a small part of the load on racks 2 and 4, particularly to their lower nodes, and the remainder was still allocated to racks 1 and 3, again to nodes closer to the floor. This is explained by the strong contribution of the nodes in racks 2 and 4 to the heat recirculation. Thus, their load is contained by the recirculation bound h, to avoid excessive heating (constraint 15).

Finally, we addressed the case in which the allocation has to consider a set of nodes dedicated to specific loads such as when a given customer hires or owns specific nodes in the datacenter. In particular, loads 1 through 20 are constrained to execute in any nodes of rack 3. Thus, for these loads, the S matrix includes a weight of 1 in the columns corresponding to nodes of rack 3 and all other nodes are signaled with the illegal allocation parameter *iAlloc*. For the remaining 80 loads, they can be allocated freely to racks 2 and 4 (weight of 1) and then to rack 1 (weight of 10) but rack 3 has *iAlloc*. The first 20 applications are allocated to node B of rack 3, the one that has the least recirculation coefficients. The other loads are allocated to racks 2 and 4 but only up to what the recirculation bound h allows, and then to rack 1, despite its higher weight.

An interesting remark is that our default allocation, i.e., without allocation constraints, is a combination of the MCE and the min-HR allocation algorithms, both reported in [16]. However, our proposed method can impose constraints that lead to other possible allocations, such as favoring the nodes closer to the ceiling, which suffer more from heat recirculation but contribute less to it.

These allocation constraints can also guide the optimizer to allocate multiple instances of the same application in different nodes, for availability and enhanced service. Figure 6 shows a situation in which the first 10 loads are replicated (1..10 and 10..20) and must be allocated to nodes A through D of rack 2, while the remainder 80 loads should be allocated primarily in racks 3 and 4 (weight of 1) and if needed in rack 1 (weight of 10). In order to separate the nodes in which the 10 replicas execute, we assign to the first 10 loads a weight of 1 to nodes A and C and 2 to nodes B and D, while for the other replicas (following 10 loads), we assign a weight of 1 to nodes B and D and of 2 to nodes A and C. The optimizer effectively allocated the first 10 loads to nodes A and C and the following 10 (replicas) to nodes B and D, while the remaining 80 loads were allocated to rack 1 and a part in rack 3.

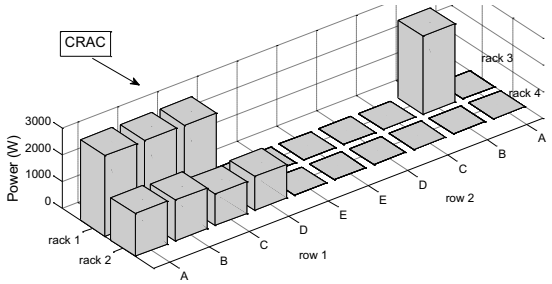


Figure 6. Automatically allocating application replicas in different nodes.

E. *Execution time of allocation process*

Finally we address the scalability of our method in terms of execution time as a function of the number of loads in the workload. We vary the number of loads from 50 to 250 in steps of 25, using h = 5ºC. Figure 7 shows the results with each point representing the average of 100 random equal size workloads. The execution time is always below 1s validating our expectation of suitability for online use according to the discussion in Section I.

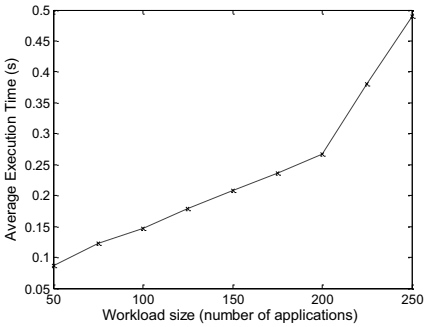


Figure 7. Execution time of the allocation optimizer for different workload sizes.

VI. CONCLUSION

Energy efficiency in datacenters is an increasingly important issue given the central role of datacenters in the IT world. In this paper, we addressed the workload distribution among servers in a datacenter. Particularly, we proposed an optimization approach to allocate the workload that minimizes the total heat recirculation among all nodes subject to a specified bound on the contribution of each node to such recirculation. We show that such bound can be used as a knob to control the number of active servers needed to execute such workload while avoiding hot spots. Acting on such knob, it is possible to find a power optimal point that combines the savings in IT power from switching off unneeded servers with the savings in cooling power from using higher cooling temperatures. This combination bears some resemblance with some previous works in the literature, such as MCE and Min-HR [16].

Nevertheless, the most innovative feature in this allocation process is the incorporation of allocation constraints in a rather simple but flexible way, using a weights matrix. We show that simply changing such weights changes the profile of the allocation. We also show how to use these constraints to reflect illegal allocations and, for example, to cater for cases in which nodes are rented or owned by datacenter customers for dedicated loads.

We are currently working on applying our approach to larger scale and heterogeneous datacenters and we also plan to analyze the use of other optimization approaches that provide an optimal balancing of the inlet temperatures, e.g., minmax as used in the Xint algorithm for HPC loads [16], to further increase the cooling temperature and reduce cooling costs.

VII. REFERENCES

[1] T. Keller and H. Hamann, "Data Center Design", Securities Industry and Financial Markets Association Technology Conference, June 2009.

[2] H. Zhou, "Cloud computing technology, application, standards and business mode," 2011

[3] DOE Data Center Energy Efficiency Program-Paul Scheihing, 2009

[4] A. Banerjee, T. Mukherjee, G. Varsamopoulos, S.K.S.Gupta, "Cooling-Aware and Thermal-Aware Workload Placement for Green HPC Data Centers," 2010

[5] L. Marshall, P. Bemis, "Using CFD for Data Center Design and Analysis ," Applied Math Modeling White Paper, January 2011

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

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

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

[9] H.Hamann , J. Lacey , M.O'Boyle, R.Schmidt, and M.Lengar, "Rapid Three-Dimensional Thermal Characterization of Large-Scale Computing Facilities," in *IEEE Transactions on Components and Packaging Technologies*, 2009.

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

[11] J.Moore, J.Chase, P. Ranganathan, and R.Sharma. "Making scheduling "cool": temperature-aware workload placement in data center," In *ATEC*, 2005.

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

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

[14] J.Fulton, "Control of Server Inlet Temperatures in Data Centers-A Long Overdue Strategy", White Paper, AFCOsystems, US, 2006.

[15] R.Buyya, A.Beloglazov, J.Abawajy,"Energy-efficient management of data center resources for cloud computing: a vision, architectural elements and open challenges", PDPTA 2010.

[16] Qinghui Tang *etal*. "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyberphysical approach". IEEE Transactions on Parallel and Distributed Systems, 19:1458–1472, 2008.

[17] Sandeep K.S. Gupta, Rose Robin Gilbert, Ayan Banerjee, Zahra Abbasi, Tridib Mukherjee, Georgios Varsamopoulos, "GDCSim - An Integrated Tool Chain for Analyzing Green Data Center Physical Design and Resource Management Techniques", IEEE IGCC 2011.

[18] Luca Parolini, Bruno Sinopoli, Bruce H. Krogh, "Reducing Data Center Energy Consumption via Coordinated Cooling and Load Management", Usenix HotPower 2008.

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

[20] http://impact.asu.edu/BlueTool/wiki/index.php/BlueSim

[21] http://www.gurobi.com