Object-Oriented Modelling and Simulation of Air Flow in Data Centres Based on a Quasi-3D Approach for Energy Optimisation

Alberto Leva*, Daniele Mastrandrea*, Marco Bonvini† and Alessandro Vittorio Papadopoulos‡
*DEIB, Politecnico di Milano, Italy, Email: alberto.leva@polimi.it
†Lawrence Berkeley National Laboratory, Berkeley, CA, USA, Email: MBonvini@lbl.gov
‡Department of Automatic Control, Lund University, Sweden, Email: alessandro.papadopoulos@control.lth.se

***Abstract*—Cloud computing has recently received considerable attention, as a promising approach for delivering ICT services by improving the utilisation of data centre resources. On the other hand, the increased usage of ICT, jointly with the increased cost of energy, make designing and managing data centres with energy efficient strategies a crucial and strategical problem. In the literature there are many studies based on Computational Fluid Dynamics (CFD) simulation models. While the obtained results are usually very reliable and accurate, the massive use of such models for control design require a large amount of time due to their computational complexity and scarce flexibility. In addition, CFD models are usually quite sensitive to any kind of change to the operating conditions, e.g. when varying the considered load balancing algorithm. In this paper, we propose a modelling and simulation framework suited for control design, that is able to simulate both heating and energy phenomena, and different load balancing strategies in a data centre. The purpose of this work is to provide a tool to perform energy-efficiency targeted studies, providing accurate and reliable results.**

***Keywords*-Data Centres; Energy Optimisation; Object-Oriented Modelling and Simulation;**

I. Introduction

The number of massive data centres is increasing with a very high rate in the last years, especially due to cloud computing. Their energy consumption however is a growing issue. Large companies such as Amazon, Google, or Microsoft have a huge amount of data centres, each housing thousands of servers. According to [1], energy costs for data centres continue to rise, already exceeding $15 billion per year. And this trend is more than positive, since the total worldwide electricity consumption in communication networks grew from 200 TWh per year in 2007 to 330 TWh per year in 2012, which corresponds to an annual growth rate of 10.4%, [2]. These data put a significant importance to the problem of efficiently managing energy in data centres.

To this end, different solutions were proposed. On one side, researchers have focused on how to adapt the number of active servers in a view to minimize energy consumption, while meeting response time Service Level Agreements (SLAs) and maximising utilisation [1, 3–5]. On the other hand, power delivery, electricity consumption, and heat management studies for data centre environments gained a lot of importance [6–10]. In this last type of problems, simulation models play a key role for the quality of the obtained results. That is the reason why Computational Fluid Dynamics (CFD) simulation is extensively used for simulate airflow and heating components in data centres.

CFD modelling offers a practical and comprehensive design approach. Computational simulations can be used for a quick setup of any proposed layout, any desired placement of conditioning units, and any imagined failure scenario. The "computational" trial-and-error process is widely preferable, since performing a simulation is much faster and more economical than building an actual layout. However, a CFD simulation can last 24 hours or even more, and its complexity hinders the possibility to use this tool for real-time simulation or control design. In addition, CFD models are not easy to integrate with others, e.g. control system models, unless co-simulation tools are used. As a result, CFD is a powerful methodology to *size* a system, but nowadays it is not possible to use it to design a control system, and even more to perform on-line forecasts, strategy evaluation, and similar tasks.

The motivation of this work is to reduce the computational complexity of modelling by trading accuracy for simplicity, while still obtaining a model that in a reasonable amount of time would be able to differentiate between energy and temperature achieved by different load balancing strategies. The aim is not to provide a novel solution for energy-efficiency in data centres, but rather a tool to compare, in terms of energy consumption, multiple control strategies based on physical models, in a reliable and efficient way.

The proposed simulation framework relies, from a technological viewpoint, on the advantage of adopting a modular Object-Oriented Modelling (OOM) approach, by using the Modelica language [11], and from a methodological viewpoint on the idea of quasi-3D subzonal models [12].

This work was partially supported by the Swedish Research Council (VR) for the projects "Cloud Control" and "Power and temperature control for large-scale computing infrastructures", and through the LCCC Linnaeus and ELLIIT Excellence Centers.

II. Related work

In the management of a data centre, there are two major control problems: managing the computational resources to best serve the received requests, and govern the ambient conditions so that no server has to shut down or to degrade performance, owing to a lack of air cooling capacity. These two problems have a relevant energy impact and are mutually dependent, since the server power consumption depends on the load management but affects the heat released to the air, while the conditioning of the air modifies the behaviour of the server cooling system. Hence, to optimise energy use, a coordination of load management and air conditioning is beneficial. However, performing experiments on real plants may be too expensive, and in many cases this possibility is hindered by the lack of an adequate sensing infrastructure [13]. These are the main reasons for the extensive use of simulation tools. On the other hand, very sophisticated models are usually useless for control design purposes, and simple ones are preferable [14–16].

A. CFD-based simulators

The typical approach is to use CFD tools, that compute thermofluid dynamics phenomena in the data centre [17–19] at a very fine-grained level. However, the devised solutions often are based on some assumptions about the internal autoscaling and load balancing algorithms, hence assuming a uniform load in the overall data centre.

In [17] only the effect of a failure of a Computer Room Air Conditioner (CRAC) is considered as a possible variation from the nominal conditions, while usually some of the mentioned assumptions are made. Patel et al. developed a CFD model in order to use inlet temperatures as the key measure of a proper data centre thermal design [9], while Karki et al. describe a CFD model for calculating airflow rates through perforated tiles in raised-floor data centres [10]. Cremonesi and Sansottera employed a simplified model – yet computationally heavy – to devise a linear model based on the assumption that the fraction of air recirculating from one server to another is constant [20] . The obtained model is fairly accurate, but the extensive use of CFD simulation during the training phase, combined with its intrinsically static nature, makes it sensitive to any kind of change.

The applicability of CFD-based approaches is thus limited, since they are less flexible to be adapted to different policies, and are not suited to include computing phenomena. Furthermore, using CFD-based approaches for control design is possible [21] but usually too computationally expensive.

B. Data centres and cloud simulation frameworks

In the context of data centres and cloud computing, different non-CFD simulation frameworks have been already proposed [22, 23]. Some of them, also take into account energy aspects, like for example CloudSim [24], GreenCloud [25], and MDCSim [26].

CloudSim [24] is modular, extensible, and open-source simulation framework. It is able to model very large scale clouds, e.g. in [24] 100000 machines have been instantiated in less than 5min, requiring only 75MB of RAM. It is based on SimJava [27], for the event based simulation engine, and on the GridSim toolkit [28] for the modelling of the cluster, including networks, traffic profiles, resources, and so forth. CloudSim extends the GridSim core functionalities by modelling storage, application services, resource provisioning between virtual machines, and data centre brokerage, and can even simulate federated clouds.

GreenCloud [25] is written in C++, and simulates a cloud as a packet network and estimates energy consumption at the servers, switches and links level. Unlike CloudSim, GreenCloud focuses specifically on the measurement of energy consumption. The power models used to estimate the energy consumption assume proportionality of the power consumption to the CPU load in servers, and the power consumption of switches to be almost constant and proportional to the transmission rate only at a very small scale.

MDCSim [26] has been designed with an emphasis on multi-tier data centres. It can analyse a cluster-based data centre with detailed implementation of each individual tier. It has been configured into three layers, including a communication layer, a kernel layer and user-level layer, for modelling the different aspects of a cloud, and can estimate the throughput, response times, and power consumption. The latter is approximated using linear functions of the server utilisation, which in turn is calculated based on the number of nodes, number of requests and average execution time of requests.

Although these simulation frameworks provide interesting solutions, they seldom include physical models, but rather some suitable approximations and heuristics. This results in a rough estimation of energy savings, with the consequence that, to date, simulation tools are "still based on assumptions and simplifications that might not fully represent an actual cloud. For this reason, it may be preferable to use real cloud testbeds" [23].

C. Remarks

The brief literature review above confirms the existence of the gap mentioned in the introduction. Physical modelling is required to evaluate a control strategy also in off-design conditions, as heuristics based on some operating points may easily not be reliable. On the other hand, studies involving even just ten simulation runs or so, become extremely lengthy with traditional CFD. We therefore devised an intermediate solution, based on a coarser CFD approach, yet providing enough accuracy at the system level, and particularly suited for control design.

III. MODELLING

To devise and assess coordinated control strategies, dynamic models of the involved system components are required. We here consider only the servers, the CRAC, the air in the data centre, and the control system components. For this purpose, we developed a proof-of-concept Modelica library which is released under the Modelica License 2[1].

A. The servers

Modelling a server is quite complicated. Commercial products are, in fact, designed to address multiple issues: besides load elaboration, which is of course the main goal, a server should provide interfaces for storage and network communication, smart power management (e.g., advanced configuration and power interface states), virtualisation and so forth. For the purpose of this work, however, there is no need to model accurately the hardware part, since a simple model, yet reliable concerning thermal issues, is enough for a system-level analysis and for control design purposes. The power consumption of the server can be represented as

$$W = W_{\text{idle}} + f \cdot l \cdot W_{\text{busy}} \tag{1}$$

where W_{idle} is the power consumed when the server is idle, W_{busy} is the power dynamically consumed according to the frequency f and to the load l. Both frequency and load are continuous quantities in the interval $[0,1]$. This assumption, jointly with Equation 1, implicitly states that the amount of work done by a server is $0 \le l \cdot f \le 1$. Notice that the product $l \cdot f$ is usually referred as utilization factor. In this work we used a Dell PowerEdge M610 blade server, which actual power consumption can be devised using one of SPEC's benchmarks[2].

Blade servers are typically enclosed in some kind of chassis organised in racks, that should be modelled as well. Here we considered the case of a Dell PowerEdge M1000e. However, a rack model does nothing more than its physical counterpart: it just holds servers and routes control signals coming from the controller. The Rack Enclosure adopted herein is a Dell PowerEdge 4220. The Modelica scheme of the overall rack is represented in Figure 1.

B. The Computer Room Air Conditioner

In accordance with the aim of the framework, there's no need to model the complex thermodynamic phenomena involved in a Heating, Ventilation and Air Conditioning (HVAC) system. The only task performed by the CRAC model is to inject an amount of power computed in order to keep its outlet temperature as close as possible to the set-point.

The modular structure of the library allows a more detailed model to be quickly integrated upon need, e.g. a model

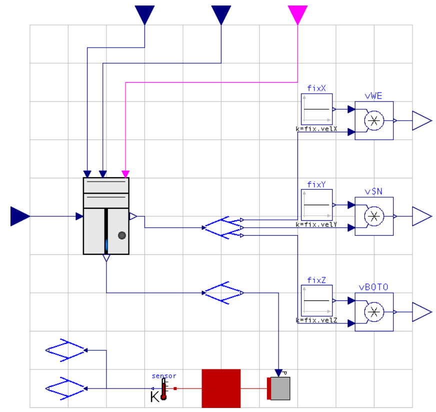

Figure 1: Modelica scheme of the rack.

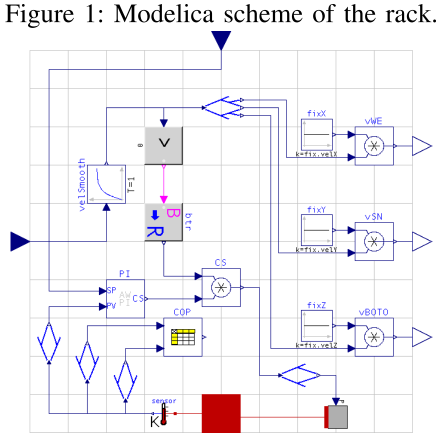

Figure 2: Modelica scheme of the CRAC.

taking into account the hydraulic part, too. The Modelica scheme of the CRAC is represented in Figure 2. The CRAC adopted as a reference here is STULZ CyberAir 2 DX 531.

C. The air in the data centre

The modelling of air volumes is the most relevant component in the overall simulation. Since the air characteristics in a data centre are not spatially uniform – while hot and cold spots are crucial to determine the operational possibilities of the installed servers – an approach named *sub-zonal modelling* [29] is here adopted to achieve a reasonably good computational efficiency.

The mass, energy balance and the Navier-Stokes (momentum) equations [30] can be written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{2}$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho \mathbf{v} h) = \nabla \cdot (k \nabla T) \tag{3}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}^{\mathrm{T}}) + \nabla p = \mathbf{f} \tag{4}$$

where the scalars p, T, e, h and ρ are respectively the fluid pressure, temperature, specific energy, specific enthalpy and density, the vectors $\mathbf{v}$ and $\mathbf{f}$ are the fluid velocity and the possible motion driving forces, and the scalar parameters k is the fluid thermal conductivity. The Modelica scheme for the data centre is shown in Figure 3.

[1] https://github.com/albertoleva/DataCenter.git

[2] http://www.spec.org/power_ssj2008/

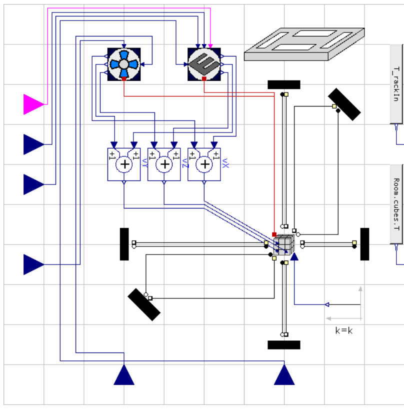

Figure 3: The data centre Modelica scheme.

Spatial discretisation is managed defining a staggered grid of points in the spatial domain of interest. Some of the nodes are attributed to control volumes, to which the mass and energy equations are referred. Some other nodes are conversely relative to the fluid motion among said volumes, thus being the locations to which velocities are attributed. In a parallelepiped-based grid the two sets of nodes form a "staggered" pattern, whence the name.

For the discretisation of the momentum equations, the velocity vector in each element is computed combining the components stored in the surrounding nodes. Pressure is calculated similarly. The partial second derivative of the velocity is dealt adopting an *ad hoc* simplification, taking a second order polynomial function as a local approximant. The final expression is obtained assuming constant viscosity.

The discretised momentum equations devised so far are valid in the volumes within a cavity (a room, a duct, a box, etc.) but apparently not for the volumes at the cavity boundaries, since velocity nodes referring to volumes at the boundary may not have one of the neighbours. A special momentum equation discretisation is thus required for boundary velocities. For further modelling details impossible to report herein, the interested reader is referred to [29].

D. Control components

Control blocks can be divided into two categories. Modulating controllers output a real-valued signal (e.g. the compressor and fan speed in an air conditioner) to make some variable (e.g. an output temperature and/or humidity) follow a reference signal, or *set point*. In typical data centre applications, modulating controls are made of simple blocks. On the other hand, data centre applications invariantly contain some logic controllers, that are typically devoted to managing the servers with the aim of properly distributing the load, turning virtual machines on and off, and so forth. Finally, and this is what makes the two control subsystems above interacting, a protection mechanism for the servers must be supplied. Servers, in fact, must work in quite a strict operating range. When temperature rises above the maximum allowed $T_{\max}$, the server is completely shut down. Additionally, since a complete shut down to start up cycle is expensive in term of components' stress (especially concerning hard disks), the controller tries to prevent its inlet temperature from surpassing the threshold by scaling the frequency and lowering the load. The jobs lost in such scenario are then pushed back to the queue.

Such a scheme is known as a *two-level* hierarchical control architecture. No matter what the load balancer determines, its decision is *overridden* by the upper level controller. At present, the library comprises three logic strategies, briefly described below. The point of these strategies is not to propose some novel and efficient control strategies, but rather that it is possible to easily include different strategies in the library and test them with a reliable description of the data centre.

1) Uniform scaling: This strategy is the most simple that can be adopted: every server is turned on and the load is equally partitioned across all the servers. Such a schedule may work well when the servers run constantly in a high load conditions, and when overheating is not an issue. In such a scenario there's not much left to be optimised. Notice that no countermeasure is taken when overheating is detected in the autoscale policy, allowing the blades to reach thermal limits for shutdown. This is an unrealistic case, however it provides a worst-case energy consumption as a baseline for the others.

2) Minimal scaling: The number of servers allocated is obtained as the ceiling of the requested load, then such load is equally partitioned across all the server currently working. Such a schedule still shows some flaws, the most important being that it does not distinguish the servers being allocated according to some metric, e.g. the temperature at inlet. An even smarter policy considers a server ranking induced by some measurable metric.

3) Smart scaling: The smartest scaling policy implemented herein relies on the ordering among the servers induced by the inlet temperatures. Servers are classified as follows.

- **Those who *can*.** If inlet temperature is $T_{\min} \leq T < T_{\text{crit}}$, where T_{crit} is a critical threshold above which some countermeasure should be taken to prevent overheating;
- **Those who *may*.** If $T_{\text{crit}} \leq T < T_{\max}$. In this case servers can still operate at reduced frequency and load (thus, power consumption), but overheating is close to happen;
- **Those who *can't*.** When the server isn't within the operating range, then it must be shut down.

The allocation policy is designed in order to scan the first two classes (those within operating range) and turn on as many servers as needed. This is the same as in III-D2, with the small difference that the servers are turned on according to their ranking. To further improve such strategy, a two-pass scanning may be adequate. The first pass, in fact, will turn on

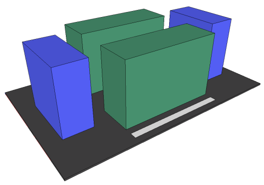

Figure 4: Configuration of the considered data centre.

just the servers that were on also in the previous allocation scheme; the second pass will deal with the remaining load, if any. This trick will reduce the number of stop-restart, thus (hopefully) extending the time-to-failure of the machines.

The allocation algorithm is not continuously applied, but it executes every T_s seconds, where T_s is assessed in order to gain a tradeoff between computation costs and timeliness. The new policy becomes operative after a T_i time frame, representing the time a server needs to be initialised. On the counterpart, load is instantly redistributed according to the current partition scheme. Higher load simply goes to the coldest server, and so forth.

IV. Case Studies

We now present three case studies of the control techniques presented in III-D. It is worth noticing that in the following no delay in the setup of a machine is considered. This is a simplifying assumption that was used in the examples, but the delay can be easily included in the models, without increasing the overall computational complexity. The geometry of the considered data centre is shown in Figure 4. The considered workload is similar to the "Dual phase" one considered in [1], i.e., for typical multi-tier applications such as the social networking site, Facebook, or e-commerce companies like Amazon. The purpose of this section is not to evaluate the real quality of the proposed control strategies, but rather to show which are type of data that one can obtain while using the proposed simulation framework, and which are the simulation performance.

The numerical results of the three policies are shown in Figure 5. Apparently the minimal and smart policies have quite similar steady-state behaviours in terms of temperatures, but when power is considered, the second one shows lower power consumption for a large amount of time. A synthetic performance and energy comparison is reported in Figure 6. As can be seen, even a quite simple improvement of the basic strategy does yield some energy benefits, that can be estimated accurately through the model. The policies considered herein have the sole purpose of evidencing this relevant fact; the presented model allows for the inclusion of more articulated ones, taken from the vast *corpus* available in the literature [1].

Finally, Table I contains some simulation performance statistics, while Figure 7 shows a snapshot of the temperature evolution as caught by the model on only 7 cutting planes, for clarity reasons. It can be observed that the accuracy in the temperature field, though clearly coarser than those

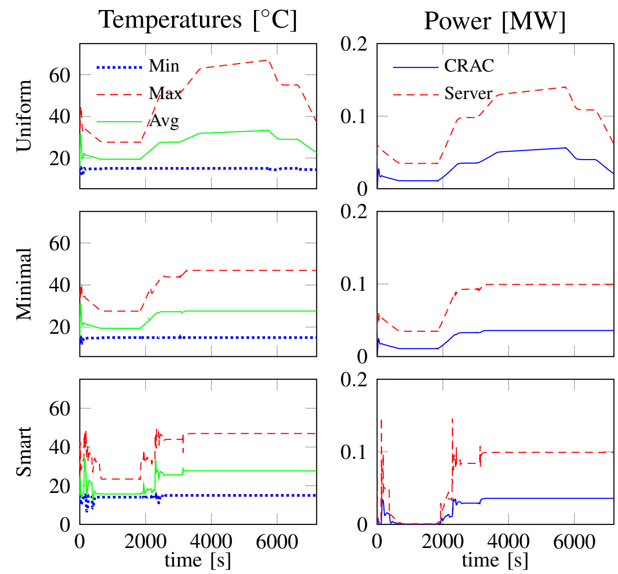

Figure 5: Simulation results for the three control strategies.

Statistic	Uniform	Minimal	Smart
Sim time	16m 50s	42m 50s	1h 32m
Sim steps	8005	8755	43657
Min stepsize	$7.52{\times}10^{-8}$s	$7.52{\times}10^{-8}$s	$7.52{\times}10^{-8}$s
Max stepsize	545s	555s	54s

Table I: Simulation statistics for a horizon of 2h.

of CFD tools, is sufficient for identifying the hot spots and then let the control actions to take the right decision. On the other hand, the simulation speed is quite high, allowing for real-time simulation—notice that the simulation horizon is 2h. Notice also that a general purpose simulation algorithm (DASSL) is here adopted. Integration methods for sparse systems can significantly improve the simulation performance.

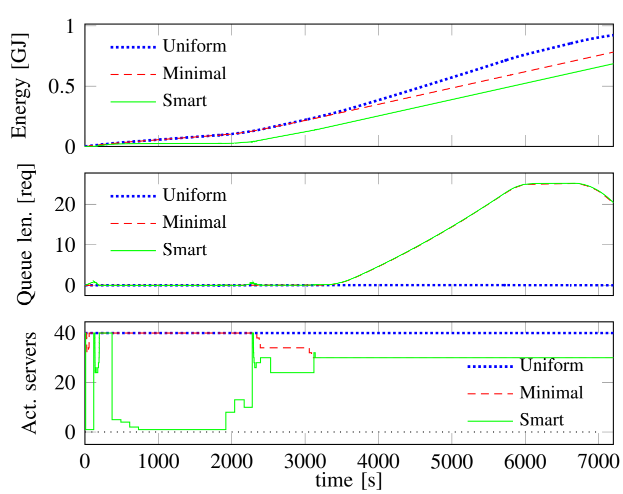

Figure 6: Comparison of synthetic performance and energy consumption.

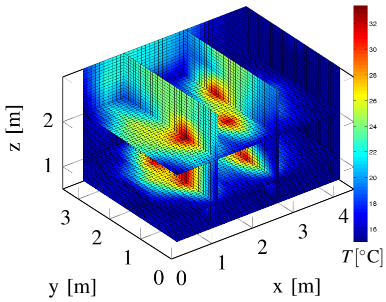

Figure 7: Snapshot of the 3D distribution of the temperature.

V. Conclusions and future work

This work proposes a modelling framework targeted for energy optimisation in data centres. Three case studies were here considered to show the type of results and of comparison that can be obtained. In particular, one can obtain synthetic data like the temperatures, the consumed energy, the number of elements in the queue, as well as more detailed ones on the 3D distribution of the temperature in the data centre. On the other hand, the simulation time is usually in the order of minutes instead of hours like in the CFD case, allowing for more convenient and inexpensive tests of different control strategies. Although simulation efficiency is already quite good, adopting sparse integration methods could help to improve the simulation efficiency. On top of that, the modelling framework is more flexible and extensible than the usual approaches, providing an interesting tool for any energy efficiency study. The considered examples are only meant to be a proof of concept. More complex load balancing policies will be implemented and analysed, and also some more complex structures of data centres considered.

References

[1] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch. “AutoScale: Dynamic, Robust Capacity Management for Multi-Tier Data Centers”. In: *ACM Trans. Comput. Syst.* 30.4 (2012), 14:1–14:26.

[2] W. V. Heddeghem et al. “Trends in worldwide ICT electricity consumption from 2007 to 2012”. In: *Computer Communications* 50.0 (2014). Green Networking, pp. 64–76.

[3] A Ali-Eldin, J. Tordsson, and E. Elmroth. “An adaptive hybrid elasticity controller for cloud infrastructures”. In: *Network Operations and Management Symposium, IEEE*. 2012, pp. 204–212.

[4] A. Ali-Eldin, M. Kihl, J. Tordsson, and E. Elmroth. “Efficient Provisioning of Bursty Scientific Workloads on the Cloud Using Adaptive Elasticity Control”. In: *Proc. of the 3rd Workshop on Scientific Cloud Computing Date*. ACM, 2012, pp. 31–40.

[5] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes. “CloudScale: Elastic Resource Scaling for Multi-tenant Cloud Systems”. In: *Proceedings of the 2Nd ACM Symposium on Cloud Computing*. SOCC ’11. Cascais, Portugal: ACM, 2011, 5:1–5:14.

[6] R. Raghavendra et al. “No “Power” Struggles: Coordinated Multi-level Power Management for the Data Center”. In: *Proc. of the 13th Int. Conf. on Architectural Support for Programming Languages and Operating Systems*. Seattle, WA, USA: ACM, 2008, pp. 48–59.

[7] X. Zhan and S. Reda. “Techniques for Energy-efficient Power Budgeting in Data Centers”. In: *Proc. of the 50th Annual Design Automation Conf.* Austin, Texas: ACM, 2013, 176:1–176:7.

[8] D. Wang et al. “Underprovisioning Backup Power Infrastructure for Datacenters”. In: *Proc. of the 19th Int. Conf. on Architectural Support for Programming Languages and Operating Systems*. Salt Lake City, Utah, USA: ACM, 2014, pp. 177–192.

[9] C. D. Patel et al. “Computational fluid dynamics modeling of high compute density data centers to assure system inlet air specifications”. In: *Proc. of IPACK*. Vol. 1. 2001, pp. 8–13.

[10] K. C. Karki, A. Radmehr, and S. V. Patankar. “Use of computational fluid dynamics for calculating flow rates through perforated tiles in raised-floor data centers”. In: *HVAC&R Research* 9.2 (2003), pp. 153–166.

[11] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 3.3: A Cyber-Physical Approach*. Wiley, 2014.

[12] M. Bonvini and A. Leva. “Object-oriented sub-zonal modelling for efficient energy-related building simulation”. In: *Mathematical and Computer Modelling of Dynamical Systems* 17.6 (2011), pp. 543–559.

[13] J. Liu and A. Terzis. “Sensing data centres for energy efficiency”. In: *Phil. Trans. R. Soc.* 370.1958 (2012), pp. 136–157.

[14] K. J. Åström and R. M. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[15] A. Leva, M. Maggio, A. V. Papadopoulos, and F. Terraneo. *Control-Based Operating System Design*. IET, 2013.

[16] A. V. Papadopoulos, M. Maggio, F. Terraneo, and A. Leva. “A dynamic modelling framework for control-based computing system design”. In: *Mathematical and Computer Modelling of Dynamical Systems* (2014), pp. 1–21.

[17] A. Beitelmal and C. Patel. “Thermo-Fluids Provisioning of a High Performance High Density Data Center”. In: *Distributed and Parallel Databases* 21.2-3 (2007), pp. 227–238.

[18] W. Abdelmaksoud et al. “Improved CFD modeling of a small data center test cell”. In: *Thermal and Thermomechanical Phenomena in Electronic Systems, 12th IEEE Intersociety Conf. on*. 2010, pp. 1–9.

[19] S. V. Patankar. “Airflow and Cooling in a Data Center”. In: *J. of Heat Transfer* 132.7 (2010), pp. 073001–1–073001–17.

[20] A. Sansottera and P. Cremonesi. “Cooling-aware workload placement with performance constraints”. In: *Performance Evaluation* 68.11 (2011). Special Issue: Performance 2011, pp. 1232–1246.

[21] Q. Meng, Y. Wang, X. Yan, and Z. Li. “CFD assisted modeling for control system design: A case study”. In: *Simulation Modelling Practice and Theory* 17.4 (2009), pp. 730–742.

[22] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Y. Zomaya. “A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems”. In: *Advances in Comp.* 82 (2011), pp. 47–111.

[23] G. Sakellari and G. Loukas. “A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing”. In: *Simulation Modelling Practice and Theory* 39.0 (2013). S.I.Energy efficiency in grids and clouds, pp. 92–103.

[24] R. N. Calheiros et al. “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”. In: *Software: Practice and Experience* 41.1 (2011), pp. 23–50.

[25] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. Khan. “GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Computing Data Centers”. In: *IEEE Global Telecom. Conf.* 2010, pp. 1–5.

[26] S.-H. Lim et al. “MDCSim: A multi-tier data center simulation, platform”. In: *Cluster Computing and Workshops. IEEE Int. Conf. on*. 2009, pp. 1–9.

[27] F. Howell and R. McNab. “simjava: A Discrete Event Simulation Library For Java”. In: *Int. Conf. on Web-Based Modeling and Simulation*. 1998, pp. 51–56.

[28] R. Buyya and M. Murshed. “GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing”. In: *Concurrency and Computation: Practice and Experience (CCPE)* 14.13 (2002), pp. 1175–1220.

[29] M. Bonvini, A. Leva, and E. Zavaglio. “Object-oriented quasi-3D sub-zonal airflow models for energy-related system-level building simulation”. In: *Simulation Modelling Practice and Theory* 22 (2012), pp. 1–12.

[30] R. Byron Bird, W. Stewart, and E. Lightfoot. *Transport Phenomena*. Wiley International edition. Wiley, 2007.