

# HEROS: Energy-Efficient Load Balancing for Heterogeneous Data Centers

Mateusz Guzek  
University of Luxembourg  
6, rue R. Coudenhove-Kalergi  
Luxembourg, Luxembourg  
Email: mateusz.guzek@uni.lu

Dzmitry Kliazovich  
University of Luxembourg  
6, rue R. Coudenhove-Kalergi  
Luxembourg, Luxembourg  
Email: dzmitry.kliazovich@uni.lu

Pascal Bouvry  
University of Luxembourg  
6, rue R. Coudenhove-Kalergi  
Luxembourg, Luxembourg  
Email: pascal.bouvry@uni.lu

**Abstract**—Heterogeneous architectures have become more popular and widespread in the recent years with the growing popularity of general-purpose processing on graphics processing units, low-power systems on a chip, multi- and many-core architectures, asymmetric cores, coprocessors, and solid-state drives. The design and management of cloud computing data-centers must adapt to these changes while targeting objectives of improving system performance, energy efficiency and reliability. This paper presents HEROS, a novel load balancing algorithm for energy-efficient resource allocation in heterogeneous systems. HEROS takes into account the heterogeneity of a system during the decision-making process and uses a holistic representation of the system. As a result, servers that contain resources of multiple types (computing, memory, storage and networking) and have varying internal structures of their components can be utilized more efficiently.

**Keywords**—Cloud Computing; Data Center; Load Balancing; Energy Efficiency; Scheduling

## I. INTRODUCTION

Heterogeneity is a growing trend in distributed systems, including cloud computing. The increasing manufacturing capabilities combined with the need for high performance and high computational density result in growing diversification and specialization of the hardware [1]. Examples of these trends include the growing utilization of general-purpose processing on graphics processing units (GPGPUs), low-power system on a chips (SoCs), multi- and many-core architectures, asymmetric cores, coprocessors, and solid-state drives. Even standardized settings, such as data centers composed of containers are facing heterogeneity [2].

The power consumption trend of electronic hardware is one of the reasons behind the growth of the heterogeneity. Equipment specialization increases energy efficiency, which in its turn requires technologies such as Dynamic Shutdown (DNS) or dark-silicon. In essence, they aim to use the most efficient hardware (or its components) for the shortest period of time.

Further developments in the field of software, notably virtualization, enable workload consolidation and exploitation of the low-power hardware states. Cloud computing, which comprises large pools of resources accessed via common resource management framework, facilitates such

optimization by creating more opportunities for aggregation. Virtualization can add an additional dimension to the heterogeneity by the introduction of various hypervisors, and by containers<sup>1</sup> which may be encapsulated in Virtual Machines (VMs). A hypervisor has impact not only on performance, but also on energy-efficiency [3].

Modern Information Technology (IT) systems are becoming structurally complex, with elaborated software stacks. To get the most out of these systems, it is necessary to perform optimization that is aware of the underlying characteristics. In this paper, we present a highly scalable load balancer, which exploits heterogeneity in data centers and is based on a mathematical modeling of the system, which enables quick, low computationally-complex decision making.

The resulting scheduler, named Heterogeneous Energy-efficient Resource allocation Optimizing Scheduler (HEROS), is validated using the GreenCloud [4] simulator, which recent extensions [5] enable to simulate heterogeneous data centers. For the purpose of this study, we prepared additional configurations and power models [3] that provide realistic test scenarios. The obtained results show that HEROS achieves state-of-the-art results in homogeneous data centers, while in heterogeneous settings it leads to significant energy savings (up to 46.4 %) in comparison with load balancers that are not aware of heterogeneity.

The rest of the paper is organized as follows: Section II describes the state-of-the-art approaches for load balancing in data centers. Section III presents the proposed load balancer. Section IV shows the performance of HEROS in comparison with other reference algorithms. Section V summarizes the paper and presents future work directions.

## II. LOAD BALANCING IN DATA CENTERS

The challenge of load balancing in data centers is vital and already extensively presented in literature. In this section, we review a selection of the most prominent and representative works. As demonstrated by ADAPT-POLICY [6], an adaptive selection from a large set of schedulers on-the-fly leads

<sup>1</sup>e.g. <http://aws.amazon.com/containers/>, <https://cloud.google.com/container-engine/>

to better performance than any scheduler can achieve on its own. Our study contributes to this approach by presenting a specialized scheduler that excels in heterogeneous settings. In addition, it has a feature of adaptability, which allows it to be used also in homogeneous settings and to dynamically extend if new server configurations or types of resources are added to a data center.

Stratus [7] is an example of an approach where heterogeneity between data centers is exploited. Stratus does not account for data center internals. Instead it focuses on high-level characteristics. On the other hand, auxiliary factors such as carbon emissions and cooling costs are accounted. The Stratus algorithm minimizes the weighted sum of the following objectives: carbon emissions, electricity costs and response time. Garg et al. propose several greedy heuristics [8] for the multi-cloud scheduling problem. The SCMA algorithm [9] solves a similar problem of assigning workload to data centers while considering renewable energy sources and thermal storage. SCMA is based on Lyapunov optimization technique.

Load balancing in data center networks can be achieved with VM placement algorithms [10], [11]. VM migration may be the best choice of optimization for some applications and workflows, but it has considerable overheads in terms of time and bandwidth requirements, especially in case of large system reconfigurations. Our approach is focused on shaping the workload itself, which is more elastic and can quickly respond to changes, without incurring additional overheads.

Liu et al. [12] propose a Distributed Flow Scheduling (DFS) for energy-aware data center networks. Such approaches do not take into account the nature of the communication sources and sinks, nor the corresponding computation or data storage needs. Our approach is more holistic, i.e. it takes into account multiple resources that are used during the data centers operations.

Saha et al. [13] present a distributed routing which takes into account communications, computations and the heat of servers. The local decisions are based on a full power model including servers and network topology elements. The proposed load balancing requires also a discovery phase of all potential destination servers and modeling all possible future decisions, which can cause significant overheads in large-scale data centers. Our approach offers similar results with less overhead.

A simple strategy to deal with in a heterogeneous setting is to simply select the most energy-efficient server [14]. However, such greedy approach may be short-sighted, leading to uneven network load distribution.

DCEERS [15] assigns minimum subset of resources to the workload by calculation of minimum cost of a multi commodity flow using the Benders decomposition. DCEERS demonstrates a performance improvement compared to DENS [16], but it comes at a cost of an increased runtime (of approximately 10 times). DCEERS is also restricted to

homogeneous servers.

The proposed HEROS approach is inspired and based on our recent solutions DENS [16] and e-STAB [17]. DENS is used for the selection of the best fit computing resources for a job execution and specifically designed to account for the communication potential of data center components. The communication potential is defined as the amount of end-to-end bandwidth provided to individual servers or a group of servers by the data center architecture. In a three-tier data center topology, servers share uplink channels of their rack switch. The communication potential relies on the racks uplink buffer occupancy, such it reacts to the growing congestion in racks or pods rather than to transmission rate variations. The resulting function favors empty queues and penalizes fully loaded queues.

An important link between DENS, e-STAB, and DCEERS is the fact that they all benefit of a reference implementation in the GreenCloud simulator, which provides an uniform experimentation platform. However, none of these algorithms can adapt well to heterogeneous platforms. The experience shows also that one of the limitations of DENS is its strict binding with the three-tier architecture, which is commonly used, but has multiple alternatives (e.g. DCell, BCube, FiConn, DPillar). On the other hand, the e-STAB algorithm is a two step-procedure that requires online knowledge about the full data center network utilization. As a result, the load is better distributed among racks than in the case of DENS. The server selection function of these schedulers has also different shapes, resulting in opposing behaviors: while DENS promotes DNS, e-STAB favors low utilization and thus prevents consolidation.

To conclude, we propose HEROS, which combines the best features of DENS and e-STAB and contributes with a novel heterogeneity-aware decision making approach. It tolerates any standard network topology, as it operates on the rack level. Still, the network load is balanced among multiple racks similarly to e-STAB. To maximally reduce energy-consumption, server selection function promotes DNS, but prevents too high utilization levels. The decision making is based on aggregated information and is characterized by a low computational complexity.

### III. HEROS – ADVANCED HETEROGENEOUS SCHEDULER

We consider the problem of task (user request) scheduling on distributed computing infrastructures. Tasks are allocated to servers, either virtualized or physical. Servers have multiple components, which are grouped by resource type. Each component is further described by a vector of numbers, called capacities, which quantitatively represents their capabilities. An example of server in this representation is presented in Figure 1. Our previous studies present how to derive the energy-efficiency parameters for a model that

enables such specification [3], together with the implementation of the model in the GreenCloud simulator [5].

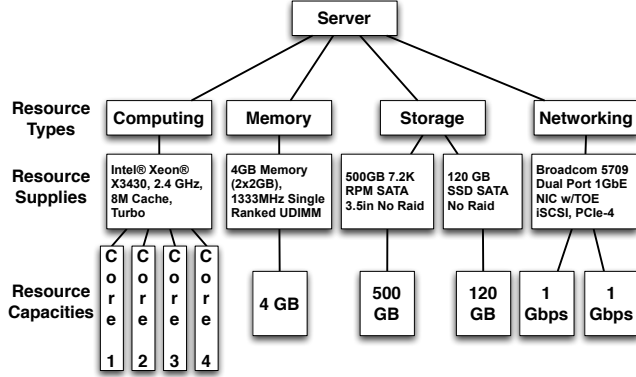


Figure 1. Example of a server with its components.

Tasks are indivisible units of work and they are described using the same model as servers, enabling composed allocations as presented in Figure 2. In practice, each task is described by the following mandatory parameters: input and output communications volumes, and the number of CPU instructions to be executed. These parameters impact utilization of networking and computing resources, respectively. It is possible that a task requires additional resources, e.g. storage on local drive or physical memory. Descriptions of tasks can be also fully heterogeneous, including several requirements for the same type of resource.

The energy-efficient scheduling has two contradictory objectives: consumed energy and mean response time (or mean *flowtime*). The minimization of the energy consumption is achieved by consolidation of the load and putting idle servers to the sleep state. The energy consumption is defined as the total energy consumed by the servers. Response time is defined as the time difference between the creation of a task and the end of its output communication. The minimization of the response time is operated by a wide distribution of the workload, which minimizes the response time. The intelligent, energy-efficient scheduling combines both of these characteristics.

The presented solution for the optimization problem is called Heterogeneous Energy-efficient Resource allocation Optimizing Scheduler (HEROS). The HEROS methodology is based on DENS [16] and e-STAB [17], and backward compatible. It indeed relies on a similar approach for establishing server selection and communication potential functions. Similarly to DENS, HEROS allocates tasks to the server with maximum *score*. The score is calculated by a decision function, which has two main components: the server selection function and the communication potential function.

The novel server selection function is based on the fact, that the range, domain, and shape of the power consumption

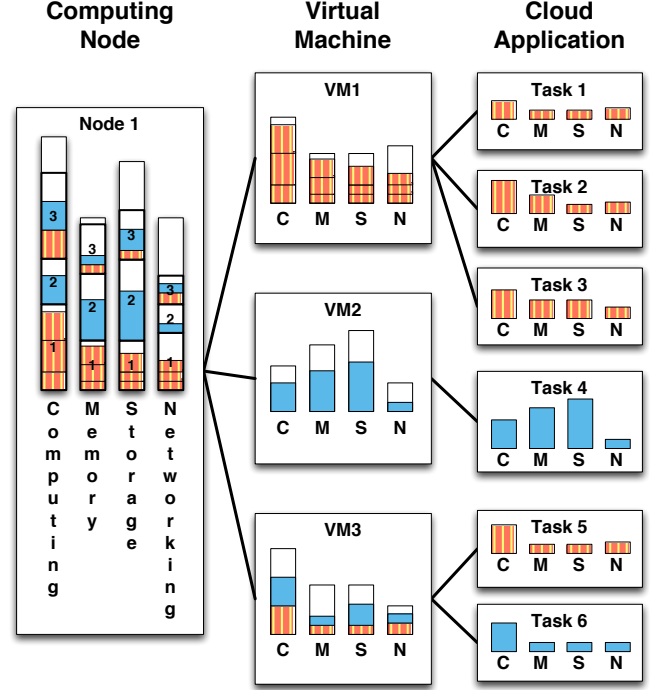


Figure 2. Resource allocation on a node with heterogeneous resources. Colors represent cloud applications and VM types, which must be compatible.

functions of heterogeneous hardware may vary significantly. Fig. 3 presents various power functions for three types of heterogeneous computing nodes: a commodity server, which is the least efficient with a concave power function, a high performance computing (HPC) server with the highest performance and a convex power function, and finally a highly efficient, yet low power micro server with a linear power function. In the literature, shape considered for the power functions is often dependent on the assumptions. Complementary Metal-Oxide Semiconductor (CMOS) technology suggest a convex relation if Dynamic Voltage Frequency Scaling (DVFS) is used [18]. The experimental studies show that the power function is in reality linear [3], [19] or even slightly concave [20]. Because of these divergences and the fact that future generations of hardware may be more energy-proportional [21], we propose a general approach, which can encompass all of these cases.

Performance per Watt (PpW) metric [20] is used to underline energy efficiency and can be directly used to select the most energy efficient server. The PpW function for server  $s$  is defined as:

$$PpW_s(l) = Perf_s(l)/P_s(l), \quad (1)$$

where  $Perf_s(l)$  is the performance function (e.g. performance in MIPS at load  $l$ ), and  $P_s(l)$  is the power consumption function.

Due to the heterogeneity, it is necessary to express  $l$  at

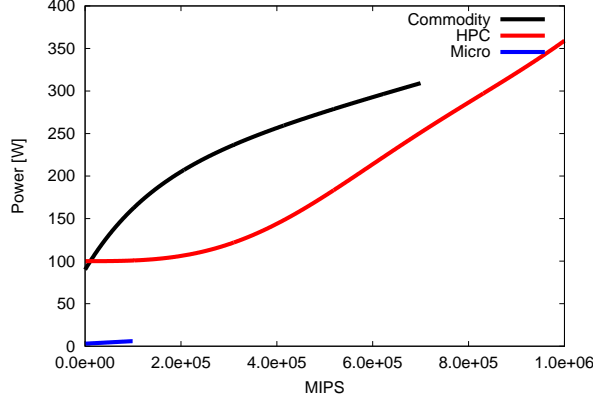


Figure 3. Power functions of heterogeneous computing servers.

the same scale for all servers, using a standard unit relevant for a scheduled application, e.g. MIPS or a number of processed requests per second. A practical drawback of the straightforward usage of PpW is the fact, that servers become the most energy-efficient when fully loaded, which in practice can easily lead to overloading and drastic reduction of performance and energy efficiency. To prevent that, the HEROS server selection function is defined as:

$$H_s(l) = PpW_s(l) \cdot \left(1 - \gamma \cdot \frac{1}{1 + e^{-\frac{\alpha}{\max l_s}(l - \beta \cdot \max l_s)}}\right), \quad (2)$$

where  $\max l_s$  is the maximum server load. The domain, or possible range of values of  $l_s$ , is defined as  $L_s := [0; \max l_s]$ . The second term of the selection function is a sigmoid scaled to the domain  $L_s$  and the range of  $PpW_s(l)$ . The sigmoid aim is to counter the impact of the PpW function for high values of load. The coefficient  $\alpha$  determines sharpness of the descending slope, while  $\beta$  is based on the maximum acceptable load of the server. In practice, these variables are set to  $\alpha = 110$ ,  $\beta = 0.9$ , and  $\gamma = 1.2$ , to assure a smooth degradation of the selection function starting from 90% of maximum load. Fig. 4 presents server selection functions for the three servers, where thinner lines present the values of  $PpW$  functions without subtraction of the sigmoid.

The communication potential  $Q(u)$  is based on the DENS communication potential, but instead of queue buffer size, it uses actual link load, and is defined as follows:

$$Q(u) = e^{-\left(\frac{2u}{U_{max}}\right)^2}, \quad (3)$$

where  $u$  is a current link load and  $U_{max}$  is the maximum link load. This function has only one component, i.e. the corresponding top-of-the-rack communication potential, which makes HEROS applicable to topologies other than three-tier. The communication potential is illustrated in Figure 5.

The final decision function is obtained by multiplication of the server selection function and the communication

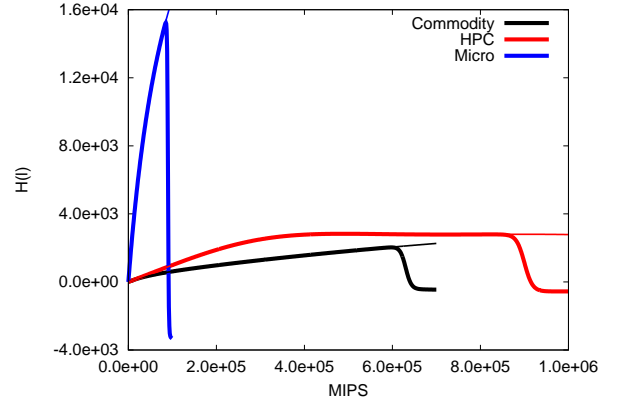


Figure 4. Selection functions of heterogeneous computing servers.

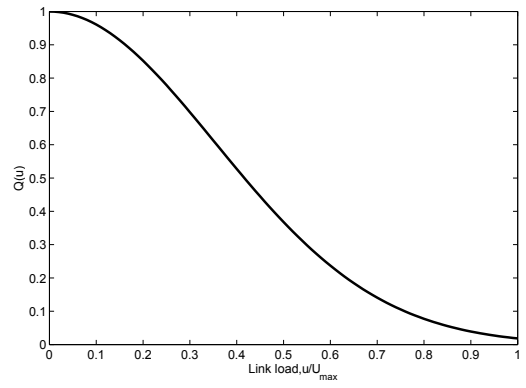


Figure 5. Communication potential.

potential function:

$$F_s(l, u) = H_s(l) \cdot Q_s(u). \quad (4)$$

Fig. 6 presents the decision function for the three presented server types.

The server chosen to execute a task is the one with the highest decision function value. In case of a tie, the server is chosen randomly among the ones featuring the best value. In case of idle servers, the maximal PpW is multiplied by the communication potential, to make a balanced choice between potential energy savings and balancing workload among racks.

Given that the data aggregation phase is performed on each computing node separately, the decision making shows little complexity. In this paper, the complexity of algorithm is  $O(n)$  in case of scanning a list of machines in order to find the best place. Sorting the list may further facilitate the selection procedure.

## IV. EXPERIMENTS

### A. GreenCloud Simulator

GreenCloud [4] is a well-known simulation tool which offers fine-grained simulation of modern cloud computing

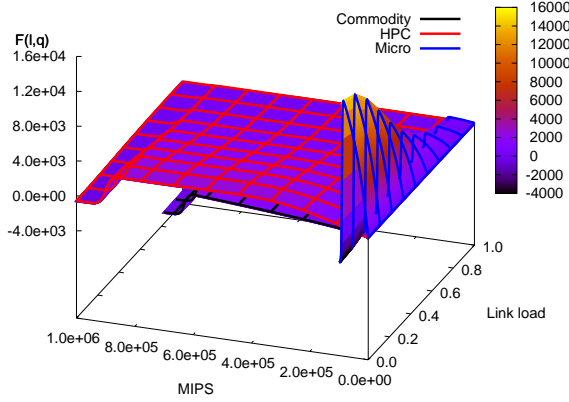


Figure 6. HEROS decision function.

environments focusing on data center communications and energy efficiency. GreenCloud is based on ns-2 [22] simulation platform. It features a detailed modeling of the energy consumed by the elements of the data center, such as computing servers, switches, and network links. It also implements a set of energy efficient metrics [23]. GreenCloud supports traditional three-tier data center architecture as well as modern data center architectures, such as DCell, BCube, FiConn, and DPillar. The three-tier architecture, used in this study, consists of the topmost core tier, the aggregation tier that is responsible for routing, and the access tier that holds the pool of computing servers arranged into racks. An important drawback of such topology is potential oversubscription. The GreenCloud simulator was extended with functionalities necessary to model heterogeneous servers [5], [3] to enable the implementation of the HEROS scheduler.

### B. Results of Simulations

The effectivity of the HEROS algorithms is validated using reference algorithms and a set of benchmarks. The first three selected algorithms: *Round Robin* (RR), *Random*, and *Green Scheduler*, are standard reference schedulers implemented in the GreenCloud simulator. The first two schedulers make uniform decisions, either cyclically allocating tasks to machines (RR) or selecting a machine from a random distribution (Random), which is uniform by default. The latter scheduler makes a greedy consolidation of the load: it looks for the first Resource Provider in the input list that can successfully finish a task. Because of that, it needs information about the current load of ResourceProviders. Another selected algorithm is the reference DENS algorithm [16], discussed in Section II.

The simulation scenarios for validating HEROS are selected to test various conditions. These benchmark scenarios are presented in Table I. *Size* is the first attribute of each scenario, while *heterogeneity* is the second attribute. Moreover, small size scenarios have larger oversubscription of the links,

including 48 servers in each of the racks, in comparison to 3 hosts per rack in the Full-scale size scenarios. The simulated specifications of servers are presented in Table II. The servers have both DVFS and DNS mechanisms enabled, and their power models are linear, defined by the minimum and maximum powers.

In each scenario, the data center load is set to 30% of the total data center power capacity. The simulation time is set to 60s and the data center is empty in the beginning. Tasks have 300,000M instructions, 8,500B of input data and 250KB of output data, and negligible needs for memory. The internal servers deadline for task execution is set to 1.2s. Because of the large number of Micro servers with low computational capacities, the heterogeneous scenarios generate less tasks than their homogeneous counterparts, composed of commodity servers.

Table I  
TESTED REFERENCE THREE-TIER CONFIGURATIONS

Configuration	Small	Full-scale	Small Hetero.	Full-scale Hetero.
Core Switches	1	8	1	8
Aggregation Switches	2	16	2	16
Access Switches	3	64	3	64
Servers in a Rack	48	3	48	3
Total Servers	144	1536	144	1536
Commodity Servers	144	1536	48	512
HPC Servers	0	0	12	128
Micro Servers	0	0	84	896
Avg. Submitted Tasks	32760	348497	21976	233783
Simulation Time	60 s			
Target System Load	30 %			

Table II  
SIMULATED SERVERS SPECIFICATIONS

Server	Commodity	HPC	Micro
Core#	4	8	4
MIPS/Core	1000100	1500150	150015
Total MIPS	4000400	12001200	600060
Max Power (W)	201	301	6
Min Power (W)	100.5	100.3	3
Hard disk	Yes	Yes	No

For each configuration, 50 independent runs with different pseudo-random number generator seeds are performed per scheduler. For each scheduler, the same set of seeds is used, to ensure identical task generation rates. The mean values serving as quality indicators are presented in Tables III–VI, while the corresponding relative values are presented in Figures 7–10. Each relative value is computed as the ratio of the value of an objective for a scheduler, to the maximum value of the objective among all schedulers in the scenario.

The main objectives for comparison of the schedulers are server energy consumption and mean response time. Both of these objectives should be minimized. Additionally, it is beneficial if the response time standard deviation is low.

Another requirement is the success rate, which is determined by the number of task failures on servers (i.e. servers detect that the tasks cannot be finished before deadline, so they drop them), and the number of unfinished tasks (i.e. tasks which do not exit data center, which is the sum of the failed tasks and tasks that did not finish sending their output communication before the end of the simulation).

All presented results were tested using statistical tests. The Shapiro-Wilk test [24] returned in many cases  $p$ -values smaller than 0.05, which means that the results were not normally distributed. To compensate this property, the Wilcoxon signed-rank test [25] was applied to each pair of the algorithms for each objective. The  $p$ -values of the tests were always smaller than 0.001, which gives stronger statistical significance than the standard threshold of 0.05.

The first considered scenario is the small, homogeneous topology. In this scenario there are more than 32 thousands tasks. As presented in Table III, most of the schedulers successfully scheduled all tasks, with a single exception. The Green scheduler results are not acceptable, as more than 10% of the tasks are unfinished. It is due to the greedy allocation policy, which locates the workload on a subset of servers that belong to the same rack. As a result, the task communication output of tasks allocated to these servers cannot leave the data center.

The relative values for this scenario are graphically presented in Fig. 7. The HEROS scheduler has the best mean response time and total energy consumption, followed by DENS. It proves that the HEROS server selection function may even outperform the one of DENS, which is designed for homogenous data centers. The Random algorithm has worse performance than RR. Both RR and Random have large energy consumption, as they prevent entering servers into sleep mode, and have also worse response time than communication-aware algorithms DENS and HEROS. The very good energy score of the Green scheduler is discredited by its large response time and a large number of unfinished tasks.

Table III  
SMALL HOMOGENEOUS TOPOLOGY – RESULTS

Scheduler	Total Energy [kWh]	Servers Energy [kWh]	Mean Response Time [s]	Response Time sd [s]	Tasks Failures #	Un-finished Tasks #
HEROS	0.302	0.149	1.22	0.0176	0	0
DENS	0.306	0.152	1.22	0.0341	0	0
Green	0.298	0.144	4.65	2.28	0	3356
RR	0.474	0.321	1.30	0.0886	0	0
Random	0.476	0.322	1.32	0.175	0	0

The results for Full-scale homogeneous scenario are presented in Table IV and Fig. 8. The Green scheduler does not leave unfinished tasks, as the oversubscription of links is smaller in this case. The behavior of the Random scheduler leads to overloading some of the servers in the large topology and consequent tasks failures. The rest of

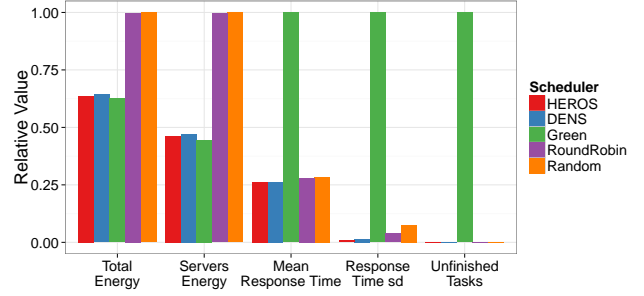


Figure 7. Small homogeneous topology – relative results

schedulers have acceptable results. The most energy-efficient algorithm is DENS, followed by Green and HEROS. The mean response time is the best for HEROS, closely followed by DENS. The good performance of HEROS is underlined by the very low standard deviation of the response time.

Table IV  
FULL-SCALE HOMOGENEOUS TOPOLOGY – RESULTS

Scheduler	Total Energy [kWh]	Servers Energy [kWh]	Mean Response Time [s]	Response Time sd [s]	Tasks Failures #	Un-finished Tasks #
HEROS	4.14	1.53	1.22	0.00878	0	0
DENS	4.10	1.49	1.22	0.0268	0	0
Green	4.13	1.52	1.27	0.0444	0	0
RR	6.03	3.42	1.30	0.0882	0	0
Random	6.05	3.43	1.32	0.173	7.64	7.64

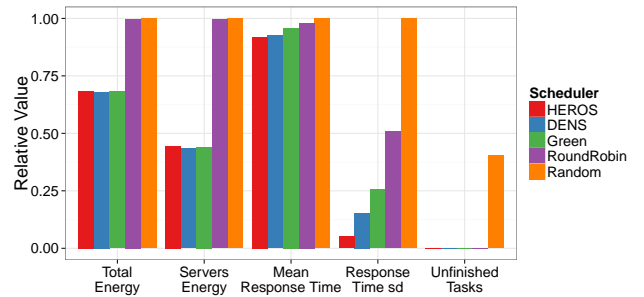


Figure 8. Full-scale homogeneous topology – relative results

The results for small heterogeneous scenario are presented in Table V and Fig. 9. Because of the heterogeneity, the schedulers that do not verify the feasibility of their allocations (RR and Random) cause failure on servers of more than a half of the tasks. In the heterogeneous scenarios data centers have fewer computational resources in comparison with the homogeneous scenarios, while the networking topology is the same, so there is no congestion in the network. Among others schedulers, Green surprisingly has the best response time and the worst energy consumption, which is explained by the fact that it sequentially chooses servers from a list that starts with the commodity servers, which are relatively



fast, but not energy-efficient. The HEROS scheduler exploits heterogeneity and its allocations consumes 39% less servers energy than in case of the Green scheduler. The small degradation of mean response time of HEROS is caused by the usage of less performant micro servers. DENS presents a behavior between Green and HEROS, however its response time standard deviation is higher than for the other two schedulers.

Table V  
SMALL HETEROGENEOUS TOPOLOGY – RESULTS

Scheduler	Total Energy [kWh]	Servers Energy [kWh]	Mean Response Time [s]	Response Time sd [s]	Tasks Failures #	Un-finished Tasks #
HEROS	0.212	0.0583	1.29	0.0369	0	0
DENS	0.231	0.0770	1.25	0.0641	0	0
Green	0.249	0.0957	1.21	0.0399	0	0
RR	0.298	0.147	1.20	7.26e-06	12802	12802
Random	0.299	0.146	1.30	0.187	12712	12712

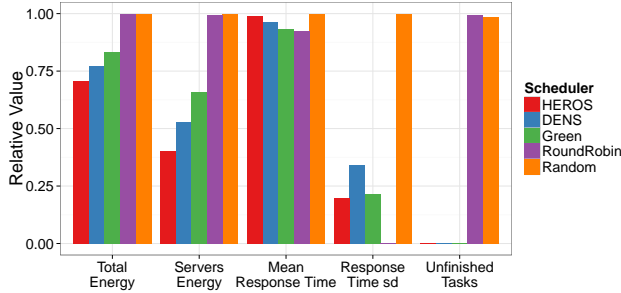


Figure 9. Small heterogeneous topology – relative results

The results for full-scale heterogeneous scenario are presented in Table VI and Fig. 10. Most of the results in this setting are similar to the small heterogenous scenario. The HEROS scheduler effectivity is better, consuming 47% less servers energy than the Green scheduler. The results of DENS are closer to the results of Green for energy consumption. Finally, the DENS scheduler has the best mean response time, while response times of Green and HEROS are the same.

Table VI  
FULL-SCALE HETEROGENEOUS TOPOLOGY – RESULTS

Scheduler	Total Energy [kWh]	Servers Energy [kWh]	Mean Response Time [s]	Response Time sd [s]	Tasks Failures #	Un-finished Tasks #
HEROS	3.15	0.536	1.25	0.0301	0	0
DENS	3.62	1.00	1.22	0.0153	0	0
Green	3.63	1.02	1.25	0.0290	0	0
RR	4.16	1.54	1.20	0.00066	136183	136183
Random	4.16	1.55	1.30	0.188	136190	136190

## V. CONCLUSION

The novel HEROS scheduler is an extension of the state-of-the-art network- and energy-aware schedulers. HEROS is

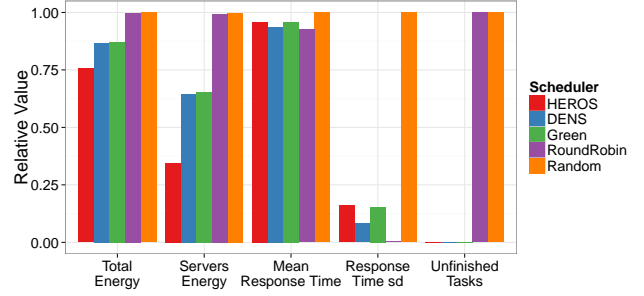


Figure 10. Full-scale heterogeneous topology – relative results

specifically designed to operate in heterogeneous systems. It bases its decisions on the aggregation of utilization and instantaneous PpW of servers with the utilization of network links. HEROS is implemented in the GreenCloud simulator, proving its effectivity in comparison with the reference scheduling approaches in homogeneous and heterogeneous systems, where it saves up to 47% of servers energy.

The decision function of HEROS effectively simplifies complex description of heterogenous servers. It also normalizes capacities and power functions of servers, making the scheduler extensible and adaptive to new settings. As a result, HEROS performs also well in homogeneous cases. Additionally, new types of servers can be dynamically added at runtime, which only requires simple calculation of their decision functions.

The exact decision-making mechanism could be further elaborated. In this paper, the complexity of algorithm is  $O(n)$ , in case of scanning all list of machines in order to find the best place. The future work will test weighted round robin algorithm approach, which would reduce complexity to  $O(1)$  in case scores are used to periodically update weights. More elaborated schemes may include a distributed organization, optimized to minimize network traffic while providing the required information.

Future directions include performing comprehensive experimentation, with non-uniform task size and task generation patterns, and simulations of more complex, virtualized, multi-tenant environments. HEROS could be improved by extension of the set of optimized objectives, integration of other data sources, and distribution of HEROS using a multi-agent framework to enable cooperation and exchange of information by schedulers in a single data center, or even between multiple cloud computing systems. HEROS could also help in solving other, related problems, e.g. energy-efficient workflow scheduling [26]. In this case, the combination of network-aware models [27] with the decision function of HEROS could enable achievement of scalable and dynamic workflow allocation in cloud systems.

## ACKNOWLEDGMENT

The authors would like to acknowledge the funding from National Research Fund (FNR), Luxembourg in the framework of ECO-CLOUD (C12/IS/3977641), Green@Cloud (INTER/CNRS/11/03), and IShOP (POL-LUX/13/IS/6466384) projects, and FNR and Tri-ICT funding under the AFR contract no. 1315254. The experiments presented in this paper were carried out using the HPC facility of the University of Luxembourg [28].

## REFERENCES

- [1] R. Nathuji, C. Isci, and E. Gorbato, "Exploiting platform heterogeneity for power efficient data centers," in *Autonomic Computing, 2007. ICAC '07. Fourth International Conference on*, June 2007, pp. 5–5.
- [2] D. Li, H. Zhao, M. Xu, and X. Fu, "Revisiting the design of mega data centers: Considering heterogeneity among containers," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1503–1515, Oct 2014.
- [3] M. Guzek, S. Varrette, V. Plugaru, J. E. Pecero, and P. Bouvry, "A holistic model of the performance and the energy efficiency of hypervisors in a high-performance computing environment," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 15, pp. 2569–2590, 2014.
- [4] D. Kliazovich, P. Bouvry, and U. Khan, Samee, "Greencloud: A packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, vol. 62, pp. 1263–1283, 2012.
- [5] M. Guzek, D. Kliazovich, and P. Bouvry, "A holistic model for resource representation in virtualized cloud computing data centers," in *2013 IEEE Int. Conf. on Cloud Computing Technology and Science (CloudCom13)*, Bristol, UK, 2013.
- [6] M. Jayasinghe, Z. Tari, P. Zeephongsekul, and A. Zomaya, "Adapt-policy: Task assignment in server farms when the service time distribution of tasks is not known a priori," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 851–861, April 2014.
- [7] J. Doyle, R. Shorten, and D. O'Mahony, "Stratus: Load balancing the cloud for carbon emissions control," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 116–128, Jan 2013.
- [8] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 732 – 749, 2011, special Issue on Cloud Computing.
- [9] Y. Guo, Y. Gong, Y. Fang, P. Khargonekar, and X. Geng, "Energy and network aware workload management for sustainable data centers with thermal storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2030–2042, Aug 2014.
- [10] D. Li, J. Wu, Z. Liu, and F. Zhang, "Joint power optimization through vm placement and flow scheduling in data centers," in *2014 IEEE International Performance Computing and Communications Conference (IPCCC)*, Dec 2014, pp. 1–8.
- [11] I. Takouna, R. Rojas-Cessa, K. Sachs, and C. Meinel, "Communication-aware and energy-efficient scheduling for parallel applications in virtualized data centers," in *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing (UCC)*, Dec 2013, pp. 251–255.
- [12] R. Liu, H. Gu, X. Yu, and X. Nian, "Distributed flow scheduling in energy-aware data center networks," *IEEE Communications Letters*, vol. 17, no. 4, pp. 801–804, 2013.
- [13] S. Saha, J. Deogun, and L. Xu, "Energy models driven green routing for data centers," in *2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 2529–2534.
- [14] N. Liu, Z. Dong, and R. Rojas-Cessa, "Task scheduling and server provisioning for energy-efficient cloud-computing data centers," in *2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, July 2013, pp. 226–231.
- [15] J. Shuja, K. Bilal, S. Madani, and S. Khan, "Data center energy efficient resource scheduling," *Cluster Computing*, vol. 17, no. 4, pp. 1265–1277, 2014.
- [16] D. Kliazovich, P. Bouvry, and S. Khan, "DENS: data center energy-efficient network-aware scheduling," *Cluster Computing*, vol. 16, no. 1, pp. 65–75, 2013.
- [17] D. Kliazovich, S. T. Arzo, F. Granelli, P. Bouvry, and S. U. Khan, "e-STAB: Energy-efficient scheduling for cloud computing applications with traffic load balancing," in *2013 IEEE International Conference on Green Computing and Communications (GreenCom)*, 2013, pp. 7–13.
- [18] Y.-C. Lee and A.-Y. Zomaya, "Energy conscious scheduling for distributed computing systems under different operating conditions," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 8, pp. 1374 –1381, aug. 2011.
- [19] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A comparison of high-level full-system power models," in *Proceedings of the 2008 conference on Power aware computing and systems*, ser. HotPower'08, Berkeley, CA, USA, 2008, pp. 3–7.
- [20] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the 34th annual int. symposium on Computer architecture*, ser. ISCA '07. New York, NY, USA: ACM, 2007, pp. 13–23.
- [21] L. A. Barroso and U. Hözlze, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, 2007.
- [22] S. McCanne and S. Floyd, "The network simulator – ns2." [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [23] C. Fiandrino, D. Kliazovich, P. Bouvry, and A. Zomaya, "Performance and energy efficiency metrics for communication systems of cloud computing data centers," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2015.
- [24] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, pp. 591–611, 1965.
- [25] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, pp. 80–83, 1945.
- [26] M. Guzek, J. E. Pecero, B. Dorransoro, and P. Bouvry, "Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems," *Applied Soft Computing*, vol. 24, pp. 432 – 446, 2014.
- [27] D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, and A. Y. Zomaya, "CA-DAG: Communication-aware directed acyclic graphs for modeling cloud computing applications," in *IEEE 6th International Conference on Cloud Computing (CLOUD)*, Santa Clara, CA, USA, 2013.
- [28] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, "Management of an academic HPC cluster: The UL experience," in *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*. Bologna, Italy: IEEE, July 2014.