Paper

Energy Efficient Scheduling Methods for Computational Grids and Clouds

Agnieszka Jakóbik[1], Daniel Grzonka[1], Joanna Kołodziej[1],
Adriana E. Chis[2], and Horacio González-Vélez[2]

[1] Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland
[2] Cloud Competency Centre, National College of Ireland, Dublin, Ireland

Abstract—This paper presents an overview of techniques developed to improve energy efficiency of grid and cloud computing. Power consumption models and energy usage profiles are presented together with energy efficiency measuring methods. Modeling of computing dynamics is discussed from the viewpoint of system identification theory, indicating basic experiment design problems and challenges. Novel approaches to cluster and network-wide energy usage optimization are surveyed, including multi-level power and software control systems, energy-aware task scheduling, resource allocation algorithms and frameworks for backbone networks management. Software-development techniques and tools are also presented as a new promising way to reduce power consumption at the computing node level. Finally, energy-aware control mechanisms are presented. In addition, this paper introduces the example of batch scheduler based on ETC matrix approach.

***Keywords*—batch scheduling, cloud computing, energy efficient, grids, power consumption, resource allocation.**

1. Introduction

In the context of a continuous increase in the demand for computing resources, the resource allocation solutions should aim not only to allocate computing resources so that they offer satisfactory service level agreements (SLAs) but also to consume the energy in an efficient way. Therefore, efficient energy-aware scheduling and resource allocation techniques are very important.

The reduction of energy consumption is one of the major challenges arising with development of grid and cloud computing infrastructures. To meet the ever-increasing demand for computing power, recent research efforts have been taking holistic views to energy-aware design of hardware, middleware and data processing applications. Indeed, advances in hardware layer development require immediate improvements in the design of system control software. For this to be possible new power management capabilities of hardware layer, need to be exposed in the form of flexible Application Program Interfaces (APIs). Consequently, novel APIs for clouds and cluster management allow for system-wide regulation as far as energy consumption. They are capable of collecting and processing detailed performance measurements, and taking real-time coordinated actions across the infrastructure.

The paper is organized as follows. Section 2 presents techniques for power consumption measurement. Section 3 is the overview of resource allocation, tasks scheduling and load balancing methods for grid and clouds considering energy expenditure control. In Section 4 the example of Expected Time to Compute (ECT) matrix scheduling process for chosen Amazon Cloud instances and its impact on the energy consumed by this environment is described. Section 5 presents a short summary of the methods presented in the paper.

2. Power Consumption Measurement and Control

In this section we present approaches for measuring, estimating, and modeling the power consumption of computing resources. The power consumption is given by the aggregated power consumed by CPU, disk, memory, network and cooling system [1], [2].

Fan *et al*. [3] investigate the power provisioning for a datacenter, and find that the actual peak power is less than 60% of the total power budget. The research shows that the CPU and the memory are the main contributors to the peak power, followed by the disk. The authors propose a model for estimating the power usage of a server based on a linear relationship between the power consumption and CPU utilization, namely they take into account the power of busy and idle servers. The evaluation shows that the model approximates the total power usage. However, for each category of servers a calibration is needed to obtain the power usage model. In addition, two techniques are presented for saving power: Dynamic Voltage/Frequency Scaling (DVFS) and improving the efficiency of non-peak power as the idle power is never lower than 50% of the peak power.

It is worth mentioning that often the power consumed by an idle machine is high, over 50% and up to 70% of the peak power consumed [3]. Therefore, to reduce the power consumption a number of approaches relying on switching idle nodes off [4] or to sleep [5] are used.

Nathuji and Schwan introduce VirtualPower [6], a system for online power management for virtualized data centers. This is a novel approach which enables virtual machines

(VMs) to have access to "soft" power states and VM specific management policies with the aim of reducing the power consumption.

Kusik *et al*. [7] propose a dynamic resource-provisioning framework for virtualized computing environments. Their approach is formulated as a sequential optimization, which employs limited lookahead control to decide the number and characteristics of the allocated resources. The goal of the research is to maximize the revenue corresponding to the provided resources by reducing the power consumption and minimizing the number of SLA violations.

Dhiman *et al*. [8] propose a system for dynamic power prediction in virtualized environments. The authors highlight that the power consumption is different for each VM and depends on the type of workload and the different characteristics of each VM and physical machine. Based on this insight they propose a solution to predict the active power usage (i.e. the power used due to the execution of a workload) at both physical machine and VMs. The prediction uses a Gaussian mixture model based predictor to estimate the power consumption based on the architectural metrics of the physical machine and its VMs. The implementation and evaluation of the proposed solution shows that the average prediction error for the power consumption is lower than 10%.

3. Energy Efficient Task Scheduling and Load Balancing

The problem of efficient task scheduling and balancing of loads over computational nodes remains challenging in the massive, extremely dynamic, elastic, diverse and heterogeneous computational environments such as computational clouds. The main issue is to distribute workloads and perform the tasks on appropriate resources in order to optimize selected objectives.

Task scheduling and workloads balancing are strongly connected with resource allocation problem. This issue becomes even more complex when energy utilization, beyond the most common optimization criteria, is treated as additional scheduling objective.

This Section highlights the most recent research in the energy efficient task scheduling and load balancing in cloud-based environments. In addition, energy-aware resource allocation approaches are also discussed.

3.1. Energy-aware Resource Allocation Heuristics Models

Resource allocation is the key issue in every distributed virtual environment. Especially energy-aware optimization is very important. There are several approaches successfully dealing with this problem. A conceptual taxonomy on energy efficient resource allocation techniques for cloud computing systems is presented in [9]. The authors define the following instances of the problem.

Resource allocation adaptation policy. An energy-aware resource allocator is reacting and adapting to changing or uncertain cloud environment. Three categories – predictive, reactive, and hybrid – are considered.

- **Predictive resource allocation adaptation policy**. Knowledge-driven machine learning techniques are used. The aim is to dynamically anticipate and capture the relationship between users QoS targets, assumed energy efficiency objective function, and given hardware resources. The knowledge about system behavior must be recorded by the monitoring service, running continuously. Resource usage planning is done before task and jobs are performed. Several machine-learning techniques such as neural networks, genetic algorithms, or reinforcement learning [10] are used.

- **Reactive resource allocation adaptation policy**. These techniques are based on monitoring of the state of a system and detecting predefined corrective actions when the negative specified event occurs. They led to the increasing of the system energy cost. The efficiency of reactive allocation depends on the ability to detect fluctuations. This approach is computationally appealing because no extensive model of the system is necessary.

- **Hybrid resource allocation adaptation policy**. This model combines predictive with reactive allocation techniques. Predictive allocation resources is performed before the processing the work. When the system is operating the reactive allocation is switched on when the monitoring system detected abnormality.

Objective function based scheduling and resource allocation. This methodology assumes finding the mathematical expression (cost function) according to the system constraints that should be minimized by numerical methods. The value of the cost function corresponds to cost of the energy utilization.

Two main closely related characteristics of cloud system might be taken into consideration during "green" scheduler constructing:

- **power-aware methods**, aiming on reducing power dissipation, power consumption, and energy cost.

- **thermal-aware methods**, targeting on reducing the thermal effects, lowering the temperature in the location of the system hardware and increasing the energy and cost to cool down the system.

3.2. Task Scheduling and Load Balancing Problem Formulation

Task scheduling is one of the most crucial issue in cloud processing. Effective scheduling approach should guarantee users' requirements and efficient resources utilization. To ensure the last one, the balancing of task loads is used.

Load balancing helps to distribute large processing load among the computing nodes. This approach has a number of goals, i.e. [11]:

- proper resources utilization,
- fair allocation of computing resources,
- support for scalability and stability of the environment,
- avoiding network and computing bottlenecks,
- extend the life of hardware resources.

We can divide load-balancing approaches into two categories: static (divides the traffic equivalently between all nodes) and dynamic (divides the traffic depending on the current state of the environment). The dynamic balancing considers two approaches: centralized, where only one node manages and distributes the whole load, and distributed – each node independently builds its own local load vector and makes all decisions [11].

In the general case, the balancing of task loads is achieved through task scheduling. The goal of this issue is to distribute workloads and perform the tasks on appropriate machines that optimize selected objectives. The problem of task scheduling in computational clouds can be reduced to the mapping tasks on individual virtual machines. Schedule can be represented by the vectors of virtual machines or tasks labels. Two different encoding methods of schedules are defined [12]:

- direct representation:

  Definition 1: Let us denote by $\mathscr{S}$ the set of all permutations **with repetition** of the length n over the set of machine labels M_l. An element $s \in \mathscr{S}$ is termed a schedule and it is encoded by the vector:

  $$s = [i_1, \ldots, i_n]^T, \quad (1)$$

  where $i_j \in M_l$ denotes the number of machine on which the task labeled by j is executed.

- permutation-based representation:

  Definition 2: Let us denote by $\mathscr{S}_{(1)}$ the set of all permutations **without repetitions** of the length n over the set of task labels N_l. A permutation $u \in \mathscr{S}_{(1)}$ is called a permutation-based representation of a schedule in CG and can be defined by the vector:

  $$u = [u_1, \ldots, u_n]^T, \quad (2)$$

  where $u_i \in N_l$, $i = 1, \ldots, n$. The cardinality of $\mathscr{S}_{(1)}$ is $n!$.

Based on the scheduling terminology introduced in [13] and [14] researchers adopted model in the form: $A|B|C$, where A specifies the resource layer and architecture type, B specifies the processing characteristics and the constraints, and C specifies the scheduling criteria. Formally, the model can be defined as follows:

$$Rm\big[\{(batch/on\text{-}line), \ldots \quad (3)$$

$$(indep/dep/wf), (stat/dyn), (dist/centr)\}\big]\,(obj), \quad (4)$$

where:

- Rm – tasks are send into parallel resources of various computing capabilities,
- $batch/on\text{-}line$ – the task processing mode is batch mode or on-line,
- *indep/dep/wf* – independency/dependency/workflow as the task interrelation,
- $stat/dyn$ – static/dynamic mode, when given number and characteristics of VMs remains/not remains the same during scheduling process,
- $dist/centr$ – references that the scheduling objectives are optimized for multi-cloud environment, where a central meta-scheduler interacts with local cloud schedulers in order to define the optimal schedules, or the centralized mode for single cloud scheduling,
- obj – denotes the set of the considered scheduling objective functions.

Definition of the main scheduling attributes is necessary for the specification of a particular scheduling problem in clouds.

Scheduling procedure can be realized in the following six steps [15]:

1. gathering the information on available resources,
2. assembling the details of pending tasks,
3. cumulating facts about data hosts where files for tasks completion are required,
4. preparing a batch of tasks or single task and compute a schedule for that batch/single mode on available machines and data hosts,
5. allocating tasks to resources,
6. monitoring the energy spent on the process when power-aware scheduling is incorporated or thermal effects, when thermal-aware scheduling was assumed.

Due to the three level services offered by the cloud vendors, these procedures may be divided as far as the scale of optimized system is concerned. Therefore, single server, compute cluster, distributed virtualized infrastructure, data centre, and the whole cloud system [16] may be taken into consideration.

3.3. Scheduling Measures and Criteria

In the problem of task scheduling we have to find schedules that minimize chosen possible objectives. The most popular scheduling criteria, namely makespan, flowtime and maximal lateness, are defined as [17]:

- makespan – the most popular time-based objective. It indicates the finishing time of the last task from task pool. The makespan can be calculated by:

$$C_{\max} = \min_{S \in Schedules} \left\{ \max_{j \in Tasks} C_j \right\}, \tag{5}$$

  where C_j denotes the time when task j is finalized (in other words, it is the machine completion time), *Tasks* denotes the set of all tasks submitted to the cloud, and *Schedules* is the set of all possible schedules;

- flowtime – defines the sum of finalization times of all the tasks. It can be defined as:

$$F = \min_{S \in Schedules} \left\{ \sum_{j \in Tasks} C_j \right\}, \tag{6}$$

  where the variables are as above;

- maximal lateness – defines the maximum time elapsed between the finalization and assumed deadline of a task. The maximum lateness is calculated as:

$$Lat_{\max} = \max_{j \in Tasks} Lat_j, \tag{7}$$

  where Lat_j denotes the lateness for the task j and

$$Lat_j = C_j - d_j, \tag{8}$$

  where C_j denotes the time when task j is finalized, and d_j is the deadline for task j;

- total energy consumption – defines the cumulative energy consumed during task batch processing. It can be defined as:

$$E_{total} = \left\{ \sum_{i \in Machines} E_i \right\}, \tag{9}$$

  where *Tasks* denotes the set of all virtual machines, and E_i the cumulative energy utilized by the machine i for the completion of all tasks from the batch that are assigned to this machine.

When the scheduling is made according to the energy consumption one of the presented criterion – Eqs. (5)–(7) – is considered as a primary scheduling criterion. The total energy consumption by Eq. (9) is the second scheduling criterion.

3.4. ETC Matrix Model Based Energy-aware Independent Batch Scheduling

An example problem that is the subject of many studies in modern task scheduling methods is Independent Batch Scheduling (IBS) [12], [18], [19]. In this problem the tasks are gathered into batches and independently processed on assigned resources. According to the notation introduced in formula (3), the problem can be defined as:

$$Rm[\{batch, indep, (stat, dyn), centr\}](obj). \tag{10}$$

The problem of IBS can be considered under several criteria. The most popular are makespan and flowtime Eqs. (5) and (6), respectively. For estimating the execution times of tasks on machines can be the ETC matrix model adopted. The model is proposed in [20] and adapted for energy-aware independent batch scheduling in [17] and [19].
In the general case, the entries of the $ETC[j][i]$ parameters can be calculated as the ratio of the workload wl of task j and computing capacity cc of machine i:

$$ETC[j][i] = \frac{wl_j}{cc_i}. \tag{11}$$

According to [17] and [19], the average energy consumption can be considered as a complementary scheduling criterion along with the makespan – see Eq. (5) – as the primary objective. The makespan is expressed as the maximum completion time of the machines. Completion time also includes the time needed for reloading the machine i after finalizing the previously assigned tasks. The minimization of the total energy consumed in the process of tasks batch execution is considered as the second step of the suboptimal schedule selection.

3.5. Energy Efficient Task Scheduling Methods for Grids

Proposed model considers two main scheduling scenarios.

Max-Min Mode, in which each machine works at the maximal DVFS during the execution and computation of tasks, and enters into idle mode after the execution of all tasks assigned to this machine. In this scenario the completion time can be defined as:

$$completion_I[i] = ready_i + \sum_{j \in Tasks(i)} ETC[j][i], \tag{12}$$

where: $ready_i$ – the ready time of machine i and $ETC[j][i]$ – the expected completion times for task j on machine i. The makespan in this scenario is calculated as:

$$(C_{max})_I = \max_{i=1}^{m} completion_I[i]. \tag{13}$$

For Max-Min Mode the average energy consumed in the system is defined as:

$$E_I = \frac{1}{m} \cdot \sum_{i=1}^{m} \gamma \cdot (completion_I[i] \cdot f \times \times [v_{s_{max}}(i)]^2 + f_{s_{min}}(i) \cdot [v_{s_{min}}(i)]^2 \cdot Idle_I[i]). \tag{14}$$

where: m – number of machines, $\gamma = A \cdot C$ (C is the total capacitance load, A is the number of switches per clock cycle), $completion_I[i]$ – completion time of the machine i, f – frequency of the machine i, $v_{s_{max}}(i)$ – machine voltage supply, $s_{min/max}$ – minimum/maximum DVFS level, $Idle_I[i]$ – the idle time for the machine i given by:

$$Idle_I[i] = (C_{max})_I - completion_I[i]\,. \tag{15}$$

Modular Power Supply Mode, in which each machine can work at different DVFS levels during the task executions and can then enter into idle mode. In this scenario the completion time, makespan, and idle time at the level s^i take specific forms given by:

$$completion_{II}[i] = ready_i + \sum_{j \in Tasks(i)} \frac{1}{f_{s_l}(i)} \cdot ETC[j][i]\,, \tag{16}$$

$$(C_{max})_{II} = \max_{i=1}^{m} completion_{II}[i]\,, \tag{17}$$

$$Idle_{II}[i] = (C_{max})_{II} - completion_{II}[i]\,. \tag{18}$$

Whereas, the average cumulative energy is defined as:

$$E_{II} = \frac{\sum_{i=1}^{m} E_i}{m}\,, \tag{19}$$

where:

$$E_i = \gamma \cdot f \cdot \sum_{\substack{j \in T(i) \\ l \in L_i}} \left([(v_{s_l}(i))_j]^2 \cdot ETC[j][i]\right) +$$
$$+ [v_{s_{max}}(i)]^2 \cdot ready_i + f_{s_{min}}(i) \cdot [v_{s_{min}}(i)]^2 \cdot Idle[i]\,, \tag{20}$$

where: $T(i)$ – a set of tasks assigned to machine i, L_i – set of DVFS levels specified for tasks assigned to machine i, and the remaining variables as in Eqs. (12)–(14).

The objective function was assumed as minimization of E_I and E_{II}.

The above-mentioned scenarios are based on Dynamic Voltage and Frequency Scaling (DVFS) technology. This method is based on decreasing power consumption of hardware by lowering the clock frequency and/or voltage of the CPU and attached peripherals under the assumption of known computational load. DVFS optimization is taking into account only CPUs. The peripherals, i.e. interfaces, memory, and disks, are being kept at the original operating frequency [21].

For the case of control all the resources of the physical machine is used less flexible technology – Dynamic Power Management (DPM). DPM methods consist of technologies to improve power conservation capabilities of computer system during runtime by shutting down the whole servers. A scheduler used in cooperation with DPM technique have to find a minimum set of computing resources for a given jobs. This approach is more efficient because the power consumption of a each server is proportional to its CPU utilization. When server is idle it still consumes around two-thirds of its peak-load consumption. This energy is spend on keeping memory, disks, and I/O resources running and ready for next task [21].

DVFS and DPM are the most popular technologies for power management of in distributed high-performance environments.

3.6. Energy Efficient Task Scheduling Methods for Clouds

There is a significant body of research on task scheduling approaches that target an efficient energy usage [2], [22]–[27]. Many of these approaches also employ switching the idle machines to sleep mode to save further on energy consumption [5], [28], [29].

Beloglazov *et al.* [5] introduce an architectural framework and principles for energy-efficient cloud computing. The authors define policies and scheduling algorithms for energy-efficient resource allocation ensuring that take into account both the quality of service provided and the power consumption. In that research the authors use the following power model

$$P(u) = k \cdot P_{max} + (1-k) \cdot P_{max} \cdot u\,, \tag{21}$$

where P_{max} is the maximum power consumed of a fully used server, k is the ratio of the power consumed by the idle servers, i.e. 70% in that paper, and u is the CPU utilization. The authors consider P_{max} as 250 W based on results offered by SPECpower benchmark[1]. The CPU utilization is workload dependent, hence changes in time. Consequently, the total energy consumed by a physical node E can be defined as:

$$C = \int_{t_0}^{t_1} P\big(u(t)\big)\mathrm{d}t\,. \tag{22}$$

The authors evaluate the proposed heuristic using modeling and simulation, and they show that using a heuristic based on minimizing the number of VMs to be migrated and considering the performance-related SLA requirements offers good energy savings.

Follow-up work by Beloglazov and Buyya [29] introduce an optimal online deterministic algorithms and heuristics for energy- and performance-efficient dynamic VM consolidation. In the context of dynamic VM consolidation, the authors defined the cost as:

$$C = \sum_{t=t_0}^{T} \left(C_p \sum_{i=0}^{n} a_{ti} + C_v \sum_{j=0}^{n} v_{tj}\right), \tag{23}$$

where t_0 is the initial time and T is the total time. a_{ti} shows whether the host i is active at time t, and v_{ti} shows whether the host j has a SLA violation at time t, the values of a_{ti} and $v_{ti} \in 0, 1$. The cost includes both the cost of power and the cost of any violation of the SLA – in this work that is when

[1] https://www.spec.org/power_ssj2008/

the service level performance, measured as maximum allowed CPU performance, cannot be met. The authors introduce novel adaptive strategies based on historical resource usage analysis for the energy efficient dynamic consolidation of VMs that minimize the total cost C. The authors propose a power-aware VM placement algorithm where all the VMs are queued in decreasing order of their CPU utilizations, and each VM will be allocated to the host that offers the minimum increase of the power usage due to the VM allocation. The evaluation of the proposed approach uses CloudSim [30], a research cloud simulator toolkit. The experiments are conducted against a simulated data center of 800 heterogeneous physical nodes. The evaluation shows that the proposed Local Regression (LR)-based algorithm combined with the Minimum Migration Time (MMT) VM selection policy provides better results for the minimization of energy and the SLA violations because of a lower number of SLA violations and VM migrations.

Mhedheb *et al.* propose ThaS [22] a load and thermal-aware VM scheduling approach with the aim to both minimize the energy consumption and ensure a good load-balancing. ThaS has been implemented on top of CloudSim [30], a research cloud simulator toolkit. The scheduler detects all the hosts that exceed either a particular temperature threshold or a CPU threshold. Next, the scheduler determines the VMs to be migrated and the target hosts. The target hosts are chosen based on temperature first, and the resources requirements second.

3.7. Meta-heuristic Energy Efficient Task Scheduling Methods

Modern energy-aware task scheduling methods are often based on a heuristic approach. These methods are usually classified into three main categories: calculus-based (greedy algorithms and ad-hoc methods), stochastic (guided and non-guided methods) and enumerative methods (dynamic programming and branch-and-bound algorithm). According to [31], the most important and efficient scheduling methods are ad-hoc, local search-based and population-based meta-heuristics methods.

Basing on proposed taxonomy, the following exemplary methods dedicated to the problem of energy aware task scheduling can be classified as meta-heuristics methods:

- **Hierarchic Genetic Strategy Based Scheduler (HGS-Sched)** is the model proposed in [17] and [19]. HGS-Sched model in the aforementioned papers was defined as meta-heuristic scheduler for solving the problem of IBS. This scheduling problem was defined by using the ETC matrix model with estimated time needed for the completion of the task j on the machine i;
- **PATC and PALS Energy-aware parallel task schedulers** [32] presented the Power Aware Task Clustering algorithm for parallel task scheduling and the Power Aware List-based Scheduling algorithm for parallel tasks.

4. Example of Batch Scheduling for Clouds Based on ETC Matrix Approach

The example of such scheduler implementation is presented in [33], [34]. It is based on additional scheduling criteria considering security of tasks computation. From among the many cloud computing security issues, [35] the mapping the task security demand into the proper VM offering the required trust level was considered. Here, for the clarity of presentation, the case considering two chosen Amazon instances will be presented. The makespan criterion, see Eq. (5), was used for scheduling. First VM (VM_1) is based on Amazon m4.16large instance with Intel Xenon E6-2686 v4 processor. Second VM (VM_2) is m4.large instance, equipped with Xenon E6-2676 v3 processor. Computing capacities of both are: $cc_1 = 2.7$ GHz $\times$ 18 cores $\times 16 = 777.6$ GFLOPS and $cc_2 = 2.4$ GHz $\times$ 12 cores $\times 16 = 460$ GFLOPS.

The batch consisting thee tasks was considered. The workload of tasks was: $wl_1 = 2000$, $wl_2 = 4000$, $wl_3 = 10000$.

The ECT matrix for such a batch is:

$$ECT = \begin{bmatrix} 2.57 & 5.14 & 12.86 \\ 4.36 & 8.68 & 21.70 \end{bmatrix}. \tag{24}$$

The possible schedules and makespans are presented in Table 1. One can see that the proper scheduling enables to save $30.38 - 8.68 = 21.7$ s. That is to shorten the makespan of tasks by over 71%.

Table 1
Possible schedules and their makespans

Schedule no.	VM_1 tasks	VM_2	Makespan [s]
1	1	2.3	28.38
2	2	1.3	26.04
3	3	1.2	13.03
4	1.2	3	21.70
5	1.3	2	8.68
6	2.3	1	30.38

Considering two time independent states of both VMs: busy (100% computational power used for tasks calculations) and idle (70% of maximal power used for system maintaining), we may calculate the energy necessary for this tasks.

Let the t_i^1 and t_i^2 be the time when VMs are idle, and t_{busy}^1 and t_{busy}^2 be the time when they are fully loaded. Let the P_i^1 and P_i^2 be the power necessary for VMs to keep idle state, and P_{busy}^1 and P_{busy}^2 be the power of VMs when they are calculating tasks. Then:

$$E_{total} = E(VM_1) + E(VM_2) = \\ = \int_0^{completiontime} Pow_{VM_1}(t)\mathrm{d}t + \int_0^{completiontime} Pow_{VM_1}(t)\mathrm{d}t = \\ = P_i^1 \cdot t_i^1 + P_i^2 \cdot t_i^2 + P_{busy}^1 \cdot t_{busy}^1 + P_{busy}^2 \cdot t_{busy}^2 . \quad (25)$$

Following [14], the VM power is estimated as the most simple linear function of virtual CPU power consumption. According to [17], the power necessary for both VMs to keep the idle state was assumed as the 70% percent of working VM. Assuming levels of VM energy:

$$P_i^1 = 231 \text{ W}, \quad P_i^2 = 140 \text{ W}, \\ P_{busy}^1 = 330 \text{ W}, \quad P_{busy}^2 = 200 \text{ W}, \quad (26)$$

the energy consumed by each VM during processing assumed batch can be calculated, see Table 2.

Table 2
Energy and energy efficiency for possible schedules for the whole environment and particular VMs

Schedule no.	Energy	$E_{efficiency}$
1	12486.21	1.28
2	11732.10	1.36
3	6889.07	2.32
4	10412.99	1.53
5	454.37	32.21
6	14413.20	1.11

The last schedule saves $14413.2 - 454.37 = 13958.83$ W. That is over 96% comparing the worst case scheduling. Considering different energy levels for both VMs:

$$P_i^1 = 70\% \cdot P_{busy}^1, \quad P_i^2 = 70\% \cdot P_{busy}^1, \quad (27)$$

$$P_{busy}^1 \in [100, 500] \text{ W } P_{busy}^2 \in [100, 500] \text{ W} \quad (28)$$

we may find the energy dynamics necessary for this batch processing, for best (no. 5) and worst (no. 6) schedule, see Fig. 1. It shows that even for the most simple energy model, the gain from proper scheduling is significant. The energy is saved for all power configurations. Moreover, the percentage savings are bigger when energy consumption of VMs are high.

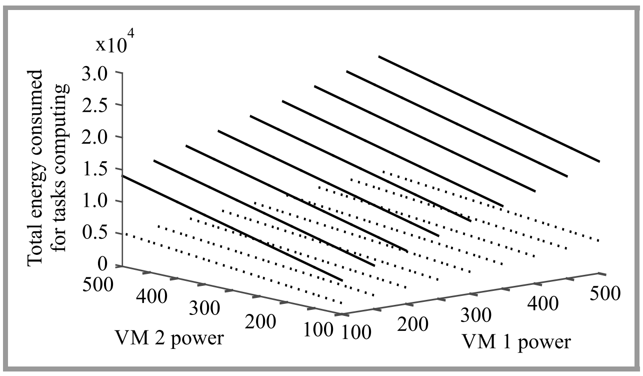

Fig. 1. Energy of batch processing for different VMs power levels for best and worst makespan schedules.

In the considered example, the power of VM was the linear function of computing capacity for particular configuration from Tables 1–2 problem of finding the schedule that minimizes the makespan is equal to the problem of finding the schedule that minimizes the total energy.
In general, the problem of finding the schedule that minimizes the makespan may be written in the form:

$$\arg \min_{s \in Schedules} \sum_{i=1,2,\, j=1,2,3} \frac{wl_j}{cc_i} \delta_{i,j} , \quad (29)$$

where $\delta_{i,j} = 0$ when the task number j is not scheduled for the machine i, $\delta_{i,j} = 1$ otherwise.
The problem of finding the schedule that minimizes the total energy may be written in the form:

$$\arg \min_{s \in Schedules} \sum_{j=1,2,3}^{\delta_{i,j}=1, i=1} P_{busy}^1 \frac{wl_j}{cc_i} \delta_{i,j} + \\ + \sum_{j=1,2,3}^{\delta_{i,j}=1, i=2} P_{busy}^2 \frac{wl_j}{cc_i} \delta_{i,j} + \\ + \sum_{j=1,2,3}^{\delta_{i,j}=0, i=1} P_i^1 \frac{wl_j}{cc_i} \delta_{i,j} + \sum_{j=1,2,3}^{\delta_{i,j}=0, i=2} P_i^2 \frac{wl_j}{cc_i} \delta_{i,j} . \quad (30)$$

One can see in this case the solution of finding the schedule that minimizes the makespan and the energy expenditure is the same. This is due to the fact that the power consumption is increasing as the computer capacity is growing, see Eq. (23).
For the schedule s and given tasks batch, the energy efficiency may be defined as the number of operations performed per energy unit (see Table 2):

$$E_{efficiency}(s) = \frac{\sum_{j=1,\dots,n} wl_j}{E_{total}(s)} . \quad (31)$$

It reflects the quality of energy aware scheduling considering given energy usage by virtual environment.

5. Summary

In this paper we addressed the problem of energy efficient task scheduling and load balancing in cloud environments. We have reviewed and discussed the methods and approaches applied for the reduction of energy consumption. The analysis shows that the problem of energy-aware task scheduling and load balancing are still very challenging.
The described model considers the multi-objective optimization problem. It focuses not only on energy consumption, but also on taking into account the time-based objectives, which are crucial in the problem of energy consumption. As a result, it considers the problem of finding the right compromise between the makespan and energy efficiency.
Additionally, we presented simple numerical example illustrating the influence of proper scheduling into energy saving.

All presented models achieved effective results in this field and are worthy of additional attention.

References

[1] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of cloud resources for real-time services", in *Proc. 7th Int. Worksh. on Middleware for Grids, Clouds and e-Science MGC'09*, Urbana Champaign, IL, USA, 2009, pp. 1:1–1:6 (doi: 10.1145/1657120.1657121).

[2] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers", in *Proc. 8th Int. Worksh. on Middleware for Grids, Clouds and e-Science MGC'10*, Bangalore, India, 2010, pp. 4:1–4:6 (doi: 10.1145/1890799.189080333).

[3] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer", in *Proc. 34th Ann. Int. Symp. on Comp. Architec. ISCA'07*, San Dieco, CA, USA, 2007, pp. 13–23.

[4] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath, "Load balancing and unbalancing for power and performancee in cluster-based systems" in *Proc. of the Worksh. on Compilers and Operat. Syst. for Low Power COLP'01*, Barcelona, Spain, 2001, pp. 182–195.

[5] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[6] R. Nathuji and K. Schwan, "Virtualpower: Coordinated power management in virtualized enterprise systems", in *Proc. 21st ACM SIGOPS Symposium on Operat. Syst. Principles SOSP'07*, Stevenson, WA, USA, 2007, pp. 265–278.

[7] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control", *Cluster Comput.*, vol. 12, no. 1, pp. 1–15, 2009.

[8] G. Dhiman, K. Mihic, and T. Rosing, "A system for online power prediction in virtualized environments using Gaussian mixture models", in *Proc. 47th Design Autom. Conf. DAC'10* Anaheim, CA, USA, 2010, pp. 807–812.

[9] A. Hameed *et al.*, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems", *Computing*, vol. 98, no. 7, pp. 751–774, 2016.

[10] S. Russell and P. Norvig, *Artificial Intelligence. A modern approach.* Englewood Cliffs: Prentice-Hall, 1995.

[11] R. Kaur and P. Luthra, "Load balancing in cloud computing", in *Proc. of Int. Conf. on Recent Trends in Inform., Telecom. and Comput. ITC '12*, Bangalore, India, 2012.

[12] D. Grzonka, J. Kołodziej, J. Tao, and S. U. Khan, "Artificial neural network support to monitoring of the evolutionary driven security aware scheduling in computational distributed environments", *Future Gener. Comput. Syst.*, vol. 51, no. C, pp. 72–86, 2015.

[13] P. Fibich, L. Matyska, and H. Rudová, "Model of grid scheduling problem", in *Proc. Worksh. on Explor. Plann. and Schedul. for Web Services, Grid and Autonomic Comput.*, Pittsburgh, PA, USA, 2005, pp. 17–24.

[14] D. Klusáček and H. Rudová, "Efficient grid scheduling through the incremental schedule-based approach", *Computat. Intell.*, vol. 27, no. 1, pp. 4–22, 2011.

[15] J. Kołodziej, S. U. Khan, L. Wang, A. Byrski, N. Min-Allah, and S. A. Madani, "Hierarchical genetic-based grid scheduling with energy optimization", *Cluster Comput.*, vol. 16, no. 3, pp. 591–609, 2013.

[16] C. Cai, L. Wang, S. U. Khan, and J. Tao, "Energy-aware high performance computing: A taxonomy study", in *Proc. 2011 IEEE 17th Int. Conf. on Parallel and Distrib. Syst. ICPADS 2011*, Tainan, Taiwan, 2011, pp. 953–958.

[17] J. Kołodziej, *Evolutionary Hierarchical Multi-Criteria Metaheuristics for Scheduling in Large-Scale Grid Systems*. *Studies in Computational Intelligence*, vol. 419. Springer, 2012.

[18] J. R. Anette, W. A. Banu, and Shriram, "A taxonomy and survey of scheduling algorithms in cloud: Based on task dependency", *Int. J. of Comp. Appl.*, vol. 82, no. 15, pp. 20–26, 2013.

[19] J. Kołodziej, S. U. Khan, and F. Xhafa, "Genetic algorithms for energy-aware scheduling in computational grids", in *Proc. Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Comput. 3PGCIC 2011*, Barcelona, Spain, 2011, pp. 17–24.

[20] S. Ali, H. J. Siegel, M. Maheswaran, D. Hensgen, and S. Ali, "Task execution time modeling for heterogeneous computing systems", in *Proc. 9th Heterogen. Comput. Worksh. HCW 2000 (Cat. No. PR00556)*, Cancun, Mexico, 2000, pp. 185–199 (doi: 10.1109/HCW.2000.843743).

[21] D. Kliazovich, P. Bouvry, and S. U. Khan, "Dens: Data center energy-efficient network-aware scheduling", in *Int. Conf. on Green Comput. & Commun. GreenCom 2010 and IEEE/ACM Int. Conf. on Cyber, Physical & Social Comput. CPSCom 2010)*, Hangzhou, China, 2010, pp. 69–75 (doi: 10.1109/GreenCom-CPSCom.2010.31).

[22] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Kołodziej, and A. Streit, "Load and thermal-aware VM scheduling on the cloud", in *Algorithms and Architectures for Parallel Processing: 13th International Conference, ICA3PP 2013, Vietri sul Mare, Italy, Dec. 18-20, 2013, Proceedings, Part II*, R. Aversa, J. Kołodziej, J. Zhang, F. Amato, and F. Giancarlo, Eds. Springer, 2013, pp. 101–114.

[23] C. Ghribi, M. Hadji, and D. Zeghlache, "Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms", in *Proc. 13th IEEE/ACM Int. Symp. on Cluster, Cloud, and Grid Comput. CCGrid 2013*, Delft, The Netherlands, 2013, pp. 671–678.

[24] C. Ghribi and D. Zeghlache, "Exact and heuristic graph-coloring for energy efficient advance cloud resource reservation", in *Proc. IEEE 7th Int. Conf. on Cloud Comput. CLOUD 2014*, Anchorage, AK, USA, 2014, pp. 112–119.

[25] A. Beloglazov, "Energy-efficient management of virtual machines in data centers for cloud computing", Ph.D. thesis, The University of Melbourne, 2013 [Online]. Available: http://beloglazov.info/thesis.pdf

[26] A. Beloglazov and R. Buyya, "OpenStack Neat: A framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds", *Concurr. Comput.: Pract. Exper.*, vol. 27, no. 5, pp. 1310–1333, 2015 (doi: 10.1002/CPC.3314).

[27] A. Iqbal, C. Pattinson, and A. L. Kor, "Managing energy efficiency in the cloud computing environment using SNMPV3: A quantitative analysis of processing and power usage", in *Proc. IEEE 14th Int. Conf. Dependable, Autonom. & Secure Comput., 14th Int. Conf. Pervasive Intell. & Comput., 2nd Int. Conf. Big Data Intell. & Comput., and Cyber Sci. and Technol. Congr. DASC/PiCom/DataCom/ CyberSciTech*, Auckland, New Zealand, 2016, pp. 239–244 (doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.60).

[28] H. Goudarzi, M. Ghasemazar, and M. Pedram, "SLA-based optimization of power and migration cost in cloud computing", in *Proc. 12th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Comput. CCGrid 2012*, Ottawa, Canada, 2012, pp. 172–179.

[29] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers", *Concurr. Comput.: Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, 2012 (doi: 10.1002/CPC.1867).

[30] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, 2011.

[31] J. Kołodziej, F. Xhafa, L. Barolli, and V. Kolici, "A taxonomy of data scheduling in data grids and data centers: Problems and intelligent resolution techniques", in *Proc. Int. Conf. on Emerging Intell. Data and Web Technol. EiIDWT 2011*, Tirana, Albania, 2011, pp. 63–71 (doi: 10.1109/EIDWT.2011.20).

[32] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C.-Z. Xu, and A. Zomaya, "Energy-aware parallel task scheduling in a cluster", *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1661–1670, 2013.

[33] A. Jakóbik, D. Grzonka, and F. Palmieri, "Non-deterministic security driven meta scheduler for distributed cloud organizations", *Simul. Modell. Practice and Theory*, Elsevier, Nov. 2016 (doi: 10.1016/j.simpat.2016.10.011).

[34] A. Jakóbik, D. Grzonka, J. Kołodziej, and H. Gonzalez-Velez, "Towards secure non-deterministic meta-scheduling for clouds", in *Proc. 30th Eur. Conf. on Modell. and Simul. ECMS 2016*, Regensburg, Germany, 2016, pp. 596–602 (doi: 10.7148/2016-0596).

[35] A. Jakóbik, "Big data security", in *Resource Management for Big Data Platforms: Algorithms, Modelling, and High-Performance Computing Techniques*, F. Pop, J. Kołodziej, and B. Di Martino, Eds. Springer, 2016, pp. 241–261.

Agnieszka Jakóbik (Krok) received her M.Sc. in the field of stochastic processes at the Jagiellonian University, Cracow, Poland and Ph.D. degree in the field of neural networks at Tadeusz Kosciuszko Cracow University of Technology, Poland, in 2003 and 2007, respectively. From 2009 she is an Assistant Professor at Faculty of Physics, Mathematics and Computer Science, Tadeusz Kosciuszko Cracow University of Technology. Her main scientific and didactic interests are focused mainly on cloud computing security, artificial neural networks, genetic algorithms, and additionally on cryptography.

E-mail: agneskrok@gmail.com
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Daniel Grzonka received his B.Sc. and M.Sc. degrees with distinctions in Computer Science at Cracow University of Technology, Poland, in 2012 and 2013, respectively. Currently, he is Research and Teaching Assistant at Cracow University of Technology and Ph.D. student at Jagiellonian University in cooperation with Polish Academy of Sciences. He is also a member of Polish Information Processing Society and IPC member of several international conferences. The main topics of his research are grid and cloud computing, multi-agent systems and high-performance computing.

E-mail: grzonka.daniel@gmail.com
Institute of Computer Science
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Joanna Kołodziej is an Associate Professor in Department of Computer Science of Cracow University of Technology. She is a vice Head of the Department for Sciences and Development. She serves also as the President of the Polish Chapter of IEEE Computational Intelligence Society. She published over 150 papers in the international journals and conference proceedings. She is also a Honorary Chair of the HIPMOS track of ECMS. The main topics of here research is artificial intelligence, grid and cloud computing, multiagent systems.

E-mail: jokolodziej@pk.edu.pl
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Adriana E. Chis received her Diploma-Engineer (Honors) title in Computer Science and Engineering from the Department of Computer and Software Engineering, Faculty of Automation and Computers, "Politehnica" University of Timisoara, Romania in 2007. She received a Ph.D. in Computer Science from University College Dublin in 2013. Currently, she is a Lecturer in the School of Computing at National College of Ireland.

E-mail: adriana.chis@ncirl.ie
Cloud Competency Centre
National College of Ireland
Dublin, Ireland

Horacio González-Vélez is currently an Associate Professor and Head of the Cloud Competency Centre at the National College of Ireland. He spent over a decade working in systems engineering and product marketing for innovation-driven companies such as Silicon Graphics and Sun Microsystems. He earned a Ph.D. in Informatics from the University of Edinburgh.

E-mail: horacio@ncirl.ie
Cloud Competency Centre
National College of Ireland
Dublin, Ireland