Paper

Recent Developments in Mobile Cloud Scheduling: State-of-the-Art, Challenges and Perspectives

Katarzyna Smelcerz

Department of Computational Intelligence, Cracow University of Technology, Cracow, Poland

Abstract—Cloud computing became recently one of the most popular multi-layer distributed computational and data processing environments with various types of services, distributed data storages and resources. With rapid development of mobile technologies, computational clouds have been transformed into the systems with dynamically changing topology and flexible infrastructure through integration with the mobile devices and mobile users as the whole system nodes and actors. The aim of this paper is to provide a comprehensive study and critical comparative analysis of the recent developments in the Mobile Clouds with a new energy optimization criterion scheduling.

Keywords—energy awareness, mobile cloud computing, scheduling.

1. Introduction

Computational Cloud (CC) can be considered as a computational environment that contains many servers distributed at geographical locations [1]. The complex structure can be modeled as a hierarchical architecture composed of three main layers, namely: Application as a Service (AaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [1].

In today's cloud computing the role of the mobile devices is not limited just to the client devices for the cloud users. The mobile platforms can be also considered as data storage and access nodes of the cloud system. In mobile cloud environment, the remote server access with mobile devices is mandatory [2]. All conventional and recently developed features of traditional CC such as privacy, security, energy awareness, resource reliability, system reliance, etc., must be analyzed differently in case of Mobile Cloud (MC) computing systems [3]. A fair optimization of the energy consumption, mainly because of short battery life, seems to be a crucial for the design of modern mobile cloud schedulers. The main aim of this paper is to present comparative analysis of CC and MC systems with a focus on energy awareness in resource and cloud application management and scheduling.

The paper is divided into five sections. Section 2 discusses the generic model of MC environment, including the detailed characteristics of the system and different types of MC. Section 3 discusses wide range of scheduling problems in MC and addresses the most important scheduling criteria such as energy consumption. There is short comparative analysis of main CC and MC features. This chapter contains also some recently developed methodologies for complexity reduction of the processes implemented and executed at the mobile devices. Those methodologies are named the offloading methods and can be applied for users' applications computational time reduction, modifications of the authentication and authorization procedures. Dynamic Voltage and Frequency Scaling (DVFS) method implemented in IaaS cloud layer is also presented in this charter as a basic hardware methodology for Mobile Cloud energy consumption optimization. Section 4 presents the selected energy-efficient algorithms in MC systems, namely Scavenger [4], MobSched [5] and Scheduler of Mobile Cells [6].

2. Mobile Cloud Environment Generic Model

Similarly to traditional Cloud system Mobile Cloud (MC) environment can be modeled as multi-layer system. However in this model each cloud layer is in fact a combination of classic infrastructure with mobile devices. Cloud platforms are therefore extended by the mobile services and mobile client applications. Mobile databases must be integrated into cloud data centers, while mobile applications are usually sent to the cloud system for execution in order to offload the mobile system nodes. This last feature is very important especially due to mobile devices limited data storage and computational performance. Mobile cloud devices (e.g. smartphones, tablets, etc.) are used by mobile users who often change their location, leave and/or join the systems dynamically and may have limited access

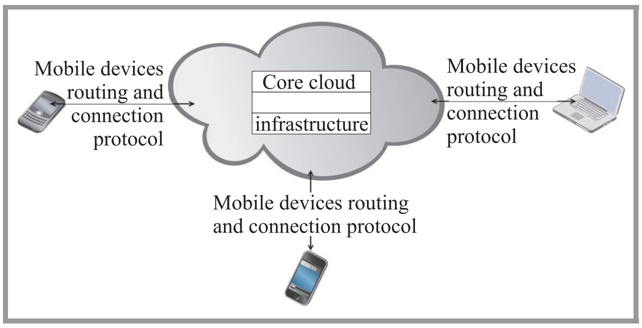

Fig. 1. Mobile Cloud system generic model.

to cloud data and services. Simplified Mobile Cloud system is shown in Fig. 1.
Brief characteristics of the system main components will be provided in the following subsections.

2.1. Mobile Cloud Core

The core of the Mobile Cloud is traditional cloud infrastructure, which is usually modeled as a 3-layer system. It consists of System as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) layers [2].

Software-as-a-Service layer can be defined as a computational environment providing online services for the Web browsers. SaaS structure delivers many different applications that can personalize cloud system according to individual user's requirements. Gmail and Amazon are examples of the SaaS [7]. SaaS layer provides specific software packages for the cloud users.

Platform-as-a-Service is development environment for cloud applications. It is independent from the hardware, so the applications can be more flexible. This platform makes data and applications portable, so teams of engineers that are in different geographical locations can concurrently work using PaaS. They can see all the changes the other did and cooperate without any distance borders. This technology is also very useful for big corporations with branch offices - employees still can work on the same project, it is easy for them to share the results of their work and to do backups. Google App Engine is the example of PaaS [7].

Infrastructure-as-a-Service is the last main layer of cloud service, which is also the core of the whole cloud structure. IaaS is based on physical networks and hosts data and databases and running applications. IaaS is a physical layer of the cloud and is composed of the physical computational unit-connecting cluster and devices. The main IaaS of the resource management problem is to provide users access to virtual machines at the same time. Another issue is effective task scheduling, in order to provide clients and tasks distribution between virtual machines to achieve the best system performance [3].

2.2. Mobile Cloud Nodes and Users

Mobile devices can be considered as integral part of the cloud network. It means that some numeric operations can be performed on those devices. The main limitation of cloud mobile nodes is limited computational capacity and data storage, as well as relatively short battery life. It means that only lightweight numerical tasks and very simple databases can be implemented there. Different mobile devices can be included in or dropped from the cloud physical layer very often. MC users' mobility management is supported by geolocation systems and communication technologies such as GSM, Wi-Fi, GPS. All such methods may increase energy consumption, which results in battery life decrease. Recently two peer based techniques, namely "Escort" [8] and "Virtual Compass" [9] have been proposed as lightweight alternative for such geolocation problems.
Three cloud's layers short comparison of the CC and MC systems main features are presented in Tables 1 and 2.

Table 1
Two cloud types system layers short comparison

<table>
<tr><th>System's layer</th><th>Computational Cloud (CC)</th><th>Mobile Cloud (MC)</th></tr>
<tr><td>SaaS</td><td>Software applications are not located on the user's devices</td><td>Applications are used by users through the vendor's website</td></tr>
<tr><td>PaaS</td><td rowspan="2">Static</td><td rowspan="2">Dynamic</td></tr>
<tr><td>IaaS</td></tr>
</table>

Table 2
Two cloud types system layers short comparison

Technology/ features	CC	MC
Security	Might be on an high level	High level is difficult to achieve because of many external users
Energy consumption	High, as computational unit is powered from AC mains	Low but mobile units are supplied from internal battery, which limits their operating time
Scheduling	Difficult to manage	Very difficult to manage because of Mobile Cloud structure

Information presented in Table 2 show that the whole resource management and scheduling issues are much more complex in MC than in classic CC.

2.3. Characteristics of the Mobile Cloud

2.3.1. Security

CC achieves high security level if data is stored in the Private Cloud. MC has dynamic structure where users can join in or drop off the cloud very often. This feature of MC can cause several issues of data security even if information is stored in the mobile Private Cloud. On the other hand it is obvious that as long as memory for data storage in mobiles is not large, many users store their personal data in the cloud. The main security issues include the following:

- privileged users access,
- data location,
- data segregation,
- recovery,
- investigative support,
- long-term viability.

One of the most interesting recent solutions for cloud systems security is model based on "weblets" technologies [10]. In this model, security framework that includes cloud manager, cloud node manager and cloud fabric interface consists of the following steps:

- check if framework is safe for the system,
- authenticate between weblets,
- perform authorization,
- establish connection and perform verification.

The safe weblets container is implemented on the mobile device and in the cloud. The authentication and authorization sessions supports access control with some specified privileges. MobiCloud is one of examples of weblets based system [10]. The architecture of MobiCloud security system is based on Mobile Adhoc Network (MANET) framework. Its main features are:

- MobiCloud is used as a link in authentication process, while cryptography is used for data access and its detachment;
- protection of the information in the system is based on VirtualTrust and Provisioning Domains that contain nodes responsible for message's security system in MobiCloud;
- MobiCloud is used to define the risk level. The risk management service finds malicious nodes that may cause some problems [10].

2.3.2. Privacy

The privacy issue in MC relates the type of cloud (hybrid, private, public, etc.) and with the type of applications that are executed on mobile devices. When the device connects to the MC, applications and user get access to the data stored in the cloud. It is important to provide way to protect data and at the same time let user to decide about privacy level. Some examples of such methods are discussed below.

In Mobile Cloud network architecture and design must be flexible and adaptive to all changes in its topology. The users often join or leave the network so some nodes are activated or deactivated by the system. Network random users are never fully trustworthy. The authentication process must be therefore adapted to the dynamic structure of the cloud. Even when the authentications requests are very frequent because of the high traffic in the cloud, the securing process needs to be completed. It can charge the system significantly, so it is important to avoid asymmetric key operations and sending large messages. The solution is the PKASSO protocol that is based on Public Key Infrastructure [11]. PKASSO offloads the mobile devices in executing complex operations. However, this method requires fast Internet connection.

Another solution for privacy in mobile cloud nodes is Onion Routing Model [12]. It provides messages multiple encryption, as every message transport chain element adds its own level of encryption. Nodes task is to remove the last layer of encoding, read routing instructions, and send data to the next router with encryption. The intermediary nodes cannot see neither sender nor recipient of the message. This solution is a type of anonymous protocol, which is sometimes unreliable as intermediary node topology can change dynamically. Anonymity is the core issue of the problem. On the other hand, it should be always considered that the higher level of privacy is implemented in the public mobile cloud, the higher are transmission and computation costs [12].

As regards the privacy issues it might be important to let users set privacy level they want to use in the public Mobile Cloud. As it was mentioned, some of the applications may use information about users of the cloud. In [13] authors discussed Privacy Rights Management for Mobile Applications system allowing users to decide what information they want to share and what privacy level they want to keep on their device.

2.3.3. Energy Utilization

Energy consumption is a crucial issue for Mobile Cloud systems. Devices connected to the cloud are battery-powered, therefore it is important to use energy saving techniques. To optimize energy consumption in the cloud, information of node performance and tasks load is mandatory.

There are two main methodologies energy consumption measurement: hardware-based and software-based. The first category includes:

- Power usage map tool, that allows programmers to modify their source code to do it more energy efficient. However it is possible only with open source operating systems, because kernel modification are required. Consumption of energy is measured by instrumentation. Optimization is manual, based on energy measurement in reference to source code [14];
- Power usage framework, where power consumption is measured by sensing power supply current via shunt resistor. To use it the test software script must be first downloaded on the mobile device from the server. The framework sends power consumption results back to application run on server, and optimization can be done automatically. After the test the results are sent back [15].

It seems that energy usage related to the tasks is not convenient and in many cases not possible as it requires hardware that usually is not available within the nodes. The software techniques based on tasks time execution let avoid this limitation. An example of the software-based methodology is PowerSpy [16] implemented for Microsoft Windows operating systems.
Some examples of energy-aware "green" mobile projects are presented in the Table 3.

Table 3
Green mobile projects examples

Name	Region	Target
Earth	Europe	Mobile network
Green IT	Japan	Commercial IT
Green Grid	Global	Data centers
OPERA-Net	Europe	Mobile networks
Green500	USA	Supercomputer

2.3.4. Cost Specification

As it was mentioned, the main issue in Mobile Clouds is the dynamics of the system topology. At any moment the architecture of the cloud can change, some devices can join or drop from the cloud. In addition the number of servers or type of executed task can change at any moment. That is why it is so difficult to optimize execution time or to set the schedulers.
The MC costs can be divided into two main groups: Total Cost of Ownership (TCO) and utilization cost [17]. There are four factors influencing cost analyze time, energy, execution time and storage time.
TCO is a cost of keeping and managing the IT infrastructure in the cloud, i.e. energy, software, servers, network support, cooling system, etc. Utilization costs are the resources used by application or user. It is important to be aware that because of the cloud architecture the number of resources (devices, servers) is dynamically changed.
Additional (surrogate) servers, used in cloud for offloading tasks from mobile devices increase system cost. The cost models, created for cloud cost optimization use resource monitoring and profiling; methods to balance the cost vs. the benefit from offloading task to surrogate servers. The cost models try to predict energy consumption, video quality, and performance are predicted for every surrogate server.

2.3.5. Main Mobile Cloud Features

As it was discussed in previous subsections MC has several problematic features. To address them the implementation of specific algorithms is required, like presented in Table 4.

2.4. Main Types of MC Systems

Similarly to CC systems, there are several types of the Mobile Clouds that can be based on the management

Table 4
Mobile Cloud implemented algorithms examples

Issue	Problem	Algorithm
Energy consumption	Video coding and data transmission	Adapts to the video and underlying network traffic to minimize the total energy consumption [18]
Video quality	The Mean Square Error between the original video frames and the decoded video frames [19]	Adopted a power rate-distortion model to capture the trade-off among the encoding rate, energy consumption of the encoder, and the video quality [20]
Power saving system	Consumption of energy during computing	Power-saving access point (PSAP) used for solar/battery powered applications [21]
Context-aware system	Different mobile devices (battery capacity, type of display), different users	Spans the application layer, Middleware layer and network layer [22]

and include the access policies to the cloud services and data [2].

Private Mobile Cloud – is a type reserved for the organizations. It is possible to completely control the privacy and security of data stored in such type of a cloud. Organizations can decide who and when can access the cloud. The high level of the security increases costs for the users.

Community Mobile Cloud – is shared by different organizations that have the same or similar security requirements. Shared data lowers security level compared to Private Mobile Cloud. It is required that all organizations that share the cloud trust each other. Because there are many partners the costs are shared among several users and can be significantly reduced.

Public Mobile Cloud – can be used by external organizations and single users. Companies are able to connect their own services to such cloud, with reduced costs and efforts. The problem is that nobody is able to control the structure, so it is not useful to store security sensitive data, e.g. medical information.

Hybrid Mobile Cloud – is a private cloud that is connected to public cloud service. This is a good solution for companies that want to have remote access to some resources. All of the company data is protected because is kept on private servers. This solution provides easier access for trusted users (in this case employees), while retaining data secured.

3. Scheduling Problems and General Scheduling Criteria

For traditional cloud systems, the following scheduling criteria can be considered:

- operational level issues (transmission protocols, scheduling algorithms, energy consumption),
- privacy, security and trust management,
- context-awareness,
- data access,
- Quality of Service.

Each of them can be used also in Mobile Cloud systems. However, due to limitation in the computational and storage capacity of mobile devices, all those criteria should be considered together with energy consumption. It means that scheduling methodologies in mobile cloud environments must be based on the idea of lightweight operations performed by mobile devices. One of the techniques which results in reduction of energy consumption is offloading operation.

3.1. Offloading Methods

In offloading methods, the computation is not executed on the mobile devices but it is distributed among the servers. Sometimes offloading can be done by sending the pointer to the file instead of sending all the data (minimizing of migration), which leads to further energy savings.
Offloading process in mobile systems can be performed in three areas: Client-Server Communication, Virtual Machine migration, Mobile agents [3].

Client-Server Communication is the transmission between offloader (mobile device) and another node. Typical communication protocols include the Remote Procedure Calls (RPC) and Remote Method Invocation (RMI) [3]. They are considered as stable protocols.

Virtual Machine migration – the memory image file of a Virtual Machine is transferred from a source server to the destination server. It is not necessary to stop application execution during migration process [23]. However, Virtual Migration method needs a long time to copy large VM data, so this method can be too slow for mobile devices.

Mobile agents – this method is used for the parallelization of mobile applications processes execution. Agents evaluate servers speed to assess the costs. The process uses special benchmarks.

Another problem is offloading procedures activation moment. Data storing, access and processing in the cloud physical and virtual networks may also increase the overall power consumption in the whole system. As it is presented in [6] offloading is only worth to be implemented in energy aware context, when there is a large number of computation data with proportionally small communication required in the system.

3.2. Dynamic Voltage and Frequency Scaling

Dynamic Voltage and Frequency Scaling (DVFS) is a hardware method used to reduce energy consumption in computing systems. Implementation of this method in cloud systems requires additional hardware circuit that can control power consumption of each node. This can be done by changing the clock speed [24] and/or supply voltage. This approach requires trade off: the best way is to increase clock and supply voltage when processor is busy, and decrease them to the lowest possible level when processor is idle. The power-frequency relation is not always linear, and is different for every platform, so optimum is difficult to find. [6]. Considering the fact that even if server is idle it still consumes energy, the sleeping modes are another solution. This can be applied to the whole system including not only servers but also all separated mobile devices.

4. Energy Efficient Scheduling Algorithms

The scheduler of Mobile Cloud Computing is implemented in PaaS layer. The data is shared by all the users connected to the Cloud at the same time, therefore it might cause some latency. The Quality of Service (QoS) is also a big problem as the number of users and the size of Mobile Cloud increase. Nowadays the solutions for data management in schedulers are well developed but still there are some issues regarding handling the data-intensive applications.
This section presents several heuristic methods implemented for scheduling problem in MC. Heuristic scheduling algorithms can be divided into two groups: heuristic and meta-heuristic. In heuristic methods every task gets priority, then is executed in decreasing order. This method is efficient but doesn't have to be optimal on global scale. Meta-heuristic methods, often called "genetic algorithms", are less efficient; however they find good solutions for complicated problems.
Dynamic structure and limited computing power of the devices in Mobile Cloud requires implementation of special schedulers.

4.1. Mobile Cells Scheduler

In this solution a model similar to mobile agents concept is proposed [25]. The whole cloud structure is build of cells, which are divided into two groups: UMSC (User Mobile Service Cell) and CMSC (Cloud Mobile Service Cell). The scheduling process is divided into three parts:

- UMSC collects requests from users;
- UMSC migrates to CMSC;
- CMSC goes to the MC and looks for the cloud's units able to handle the user requests. Then CMSC sends the information back to the mobile host.

Such methodology eliminates errors caused by turning off the Wi-Fi network or dropping off the mobile devices for Mobile Cloud users, because information is kept in the MC until it is restored. The disadvantage of this methodology appears when the mobile host is not connected while cloud cell wants to receive information. Also the cell region can lost connection with mobile host while it migrates to another cell region. The list of connection information for cloud units can solve this problem. If the wireless connection is fast the workflow is finished by cloud units, in other case the CMSC is saved until the connection is resumed.
As the model explained above, there is implemented the Genetic Algorithm (GA) based on the concept of UMSC and CMSC. Authors in [25] proposed the UMS (Universal Mobile Service) as a group of cells C with variable as arrival time, priority, state. The status of the C cell can have two values: migrated or divided.
The Mobile Cloud nodes are defined as N. The GA operates on couples like (C, N). The first step is to define initial population taking into account priority of the cells and efficiency of the computing system. Next the fitness function F minimize the computational time of the scheduling:

$$F(s) = C + D + R, \quad (1)$$

where C represents the computational time spent on executing cell C by node n, D is transmitting time of the data between cells and R is migrating time between nodes.

Next step is crossover. The representation of population is divided into two parts, for both there are solution S_1 and S_2. Next the cell P is chosen by the random function. The position of P is P_1 in S_1 and P_2 in S_2. Then S_1 and S_2 are divided into two halves and crossed like in traditional GA. Finally the last step is mutation. There are chosen two genetic representations G1 and G2 with the same cell priority. Next they are switched each other [25].

4.2. Scavenger

Scavenger provides cyber-foraging via Wi-Fi and scheduler for costs estimation. The method of estimate costs uses benchmark method to measure the speed of surrogate servers and decides if the offload process should be executed or not. The partitions, jobs, and workflows are provided by mobile code. This framework gives the possibility to offload the mobile devices to several surrogate servers. Executing the applications on a few surrogate servers as a parallel execution is more efficient. The scheduler algorithm includes:

- speed and utilization level of the surrogate server,
- latency of the surrogate and network capacity,
- complexity of the task that is described as a time needed to complete the task on the surrogate.

For every task there are created two profiles: with global task weight and for every task-device couple.
The global task weight is based on the Eq. 2 [4].

$$T_w = T_d(P_s/P_a), \quad (2)$$

where T_d = time needed to complete the task, P_s = NBench result, P_a = number of other tasks that executes at the same time on the same mobile device.

4.3. MobSched

MobSched is a scheduler providing optimization for power consumption and network capacity. It provides at the same time maximum QoS and maximum bandwidth of the computational model. The general model is presented in the Formula 3 [5].

$$\alpha(p_1x_1 + \ldots + p_Nx_N) - \beta(t_1x_1 + \ldots t_ix_i + \ldots t_Nx_N), \quad (3)$$

where α and β are the optimization parameters, p is the power consumed by every node, x is fraction of the work that was dedicated for each node and t is capacity for each node. The optimizing parameters are $0 < \alpha$ and $\beta < 1$ and $\alpha + \beta = 1$.
As the scheduler provides optimization of two (or even more) factors at the same time, there are implemented two optimization parameters (α and β). The whole system is limited by maximum tolerable error. This can be for example a number of lost packets during period.

5. Conclusions

In this work the general model of Mobile Cloud system was presented. For better explanation the MC system was compared to the traditional Computational Cloud. The paper discusses important problems related to MC, i.e. security, privacy, energy consumption and cost analysis. All of those factors contribute to the efficient scheduling. The paper also presents some examples of already implemented energy efficient schedulers, such as MobSched, Scavenger and the Mobile Cell Scheduler. As it can be noticed, the problem of energy efficient scheduling in Mobile Clouds is complex. Still the optimal solution for all of the issues in the Mobile Cloud is not found, which makes the research area wide open.

References

[1] H. Singh and D. Seehan (Sangrur) "Current trends in cloud computing a survey of cloud computing trends", *Int. J. Elec. Comp. Sci. Engin.*, vol. 1, no. 3, pp. 1214–1219, 2012.

[2] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey", *Future Gener. Comp. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[3] M. Shanklin, "MobilE CLOUD COMPUTing" [Online]. Available: http://www.cse.wustl.edu/~jain/cse574-10/ftp/cloud/index.html#sec32

[4] M. Kristensen, "Scavenger: Transparent development of efficient cyber foraging applications", in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. PerCom 2010*, Mannheim, Germany, 2010, pp. 217–226.

[5] S. Sindia *et al.*, "MobSched: Customizable scheduler for mobile cloud computing", in *Proc. 45th Southeastern Symp. Sys. Theory SSST 2013*, Waco, TX, USA, 2013, pp. 129–134.

[6] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: can offloading computation save energy?" *Computer*, vol. 43, no. 4, 2010.

[7] M. Armbrust *et al.*, "A view of cloud computing", *Mag. Commun. of the ACM*, vol. 53, no. 4, 2010.

[8] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see Bob?: human localization using mobile phones", in *Proc. 16th Ann. Int. Conf. Mob. Comput. Netw. MobiCom 2010*, Chicago, IL, USA, 2010, pp. 149–160.

[9] N. Banerjee *et al.*, "Virtual compass: relative positioning to sense mobile social interactions", in *Proc. 8th Int. Conf. Pervasive Comput. Pervasive 2010*, Helsinki, Finland, 2010. Berlin Heidelberg: Springer, 2010, pp. 1–21.

[10] D. Huang, X. Zhang, M. Kang, and J. Luo, "Mobicloud: building secure cloud framework for mobile computing and communication", in *Proc. 5th IEEE Int. Symp. Serv. Orient. Sys. Engin. SOSE 2010*, Nanjing, China, 2010, pp. 27–34.

[11] K.-W. Park, S. S. Lim, and K. Ho Park, "Computationally efficient PKI-based single sign-on protocol, PKASSO, for mobile devices", *IEEE Trans. Comp.*, vol. 57, no. 6, 2008.

[12] J. Han and Y. Liu, "Rumor riding: anonymizing unstructured peer-to-peer systems", in *Proc. 14th IEEE Int. Conf. Netw. Protoc. ICNP 2006*, Santa Barbara, CA, USA, 2006, pp. 22–31.

[13] A. Bandara *et al.*, "Privacy rights management for mobile applications", in *Proc. 4th Int. Symp. Usable Priv. Secur.*, Pitsburg, PA, USA, 2008.

[14] J. Flinn and M. Satyanarayanan, "Powerscope: a tool for profiling the energy usage of mobile applications", in *Proc. 2nd IEEE Worksh. Mob. Comput. Syst. Appl. WMCSA'99*, New Orleans, Louisiana, USA, 199, pp. 2–10.

[15] A. Rice and S. Hay, "Decomposing power measurements for mobile devices", in *Proc. 8th Int. Conf. Pervasive Comput. Pervasive 2010*, Helsinki, Finland, 2010. Berlin Heidelberg: Springer, 2010, pp. 70–78.

[16] K. Banerjee and E. Agu, "Powerspy: fine-grained software energy profiling for mobile devices", *in Proc. Int. Conf. Wirel. Netw. Commun. Mob. Comput.*, Maui, HI, USA, 2005, vol. 2, pp. 1136–1141.

[17] L. Xinhui, L. Ying, L. Tiancheng, Q. Jie, and W. Fengchun, "The method and tool of cost analysis for cloud computing", in *Proc. 2nd IEEE Int. Conf. Cloud Comput. CLOUD'09*, Bangalore, India, 2009, pp. 93–100.

[18] J. Zhang, D. Wu, S. Ci, H. Wang, and A. Katsaggelos, "Power-aware mobile multimedia: A survey", *J. Commun.*, vol., no. 9, pp. 600–613, 2009.

[19] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication dunder energy constraints", *IEEE Trans. Circ. Syst. Video Technol.*, vol. 15, no. 5, pp. 645–658, 2005.

[20] C.-H. Hsu and M. Hefeeda, "A framework for cross-layer optimization of video streaming in wireless networks", *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 7, no. 1, pp. 1–32, 2011.

[21] F. Zhang, T. D. Todd, D. Zhao, and V. Kezys, "Power saving access points for IEEE 802.11 wireless network infrastructure", *IEEE Trans. Mob. Comput.*, vol. 5, no. 2, pp. 144–156, 2006.

[22] W. He, K. Nahrstedt, and X. Liu,"End-to-end delay control of multimedia applications over multihop wireless links", *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 5, no. 2, 2008.

[23] C. Clark *et al.*, "Live migration of virtual machines", in *Proc. 2nd Symp. Netw. Syst. Design Implemen. NSDI'05*, Boston, MA, USA, 2005, vol. 2, pp. 273–286.

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

[25] Q. Liu, X. Jian, J. Hu, and H. Zhao, "An optimized solution for mobile environment using mobile cloud computing", in *Proc. 5th Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCom'09*, Beijing, China, 2009, pp. 1–5.

Katarzyna Smelcerz obtained her M.Eng. degree in Computer Science (Computational Engineering) at Cracow University of Technology, Poland in 2012. Now she is Ph.D. student at Polish Academy of Science.

E-mail: ksmelcerz@uck.pk.edu.pl
Faculty of Physics, Mathematics, and Computer Science
Department of Computational Intelligence
Cracow University of Technology
Warszawska 24 st
31-155 Cracow, Poland