Infrastructure and Energy Conservation in Big Data Computing: A Survey

2 authors:

Some of the authors of this publication are also working on these related projects:

Project   Energy-aware control systems for high performance computing View project

Project   Development of the Reservoirs Operator Office for Flood Control View project

Infrastructure and Energy Conservation in Big Data Computing: A Survey

Ewa Niewiadomska-Szynkiewicz and Michał P. Karpowicz

Institute of Control and Computation Engineering, Warsaw University of Technology, Warsaw, Poland

Abstract—**Progress in life, physical sciences and technology depends on efficient data-mining and modern computing technologies. The rapid growth of data-intensive domains requires a continuous development of new solutions for network infrastructure, servers and storage in order to address Big Data-related problems. Development of software frameworks, include smart calculation, communication management, data decomposition and allocation algorithms is clearly one of the major technological challenges we are faced with. Reduction in energy consumption is another challenge arising in connection with the development of efficient HPC infrastructures. This paper addresses the vital problem of energy-efficient high performance distributed and parallel computing. An overview of recent technologies for Big Data processing is presented. The attention is focused on the most popular middleware and software platforms. Various energy-saving approaches are presented and discussed as well.**

Keywords—*Big Data, cloud, cluster, energy-efficient computation, grid, HPC, software platform.*

1. Introduction

A truly explosive growth in the volume, variety and speed of digital data created and collected on a daily basis may be observed from the very onset of the Internet era. Big Data computing is a critically difficult challenge for High-Performance Computing (HPC). The main goal is to develop efficient technologies for transforming massively large and often unstructured or semi-structured data - firstly into valuable information, and then into meaningful knowledge. Raw data that are gathered by numerous sources including sensors, mobile devices, open and commercial datasets and archives, social networks, etc. have to be processed, often in the on-line mode. The challenge consists in the ability to integrate, store and analyze such data while satisfying fewer software and hardware-related requirements that may apply to huge collections of data sets generated by and gathered from distributed sources. Therefore, Big Data problems require continuous improvement of processors, servers, as well as storage and network infrastructure in order to enable the efficient processing of data through remote data management applications. The main challenge is to design and develop complete frameworks for

intelligent management and communication, data filtration, aggregation, correlation and fusion. Moreover, reduction in energy consumption is another technological challenge arising with the development of computing infrastructures for Big Data-related applications. The programming abstractions and data processing techniques must therefore be designed for:

- seamless implementation of applications with efficient levels of virtualization of computing resources such as servers, storage and networks,

- effective normalization, unification and merging of various types of data into a consistent format,

- energy conservation in data centers and communication infrastructure.

Distribution transparency, reliability, scalability, information sharing, fast and secure exchange of information originating from remote sources and efficient management of energy consumption are the main requirements for HPC systems. The idea behind distribution transparency is to hide the distribution-related aspects of a system from its users and applications, i.e. to provide a single system view. Transparency is often described in terms of unification of the process, memory, distributed file systems and input/output device space. A unified process space is implemented by providing visibility of and control over all processes running across the whole computing system. Unification of memory space applies to globally addressable shared memory and process-level distributed shared memory. Distributed file systems are used to provide an aggregated hierarchy of file collections stored across the computing nodes. Unification of input/output devices provides transparency and performance. Reliability and scaling are achieved through distribution, replication and caching techniques. Distributed file systems and non-relational (NoSQL) databases, fast and secure data networks aim to improve the efficiency of information sharing and remote data exchange. Modern, energy-efficient hardware components, mechanisms and methods relied upon to ensure energy-aware management of computation and communication processes foster energy conservation.

Numerous mechanisms, algorithms, computing infrastructures, software platforms and middleware for high performance computing have been developed during the past decades. Many surveys are available on recent technologies for Big Data applications (e.g. [1], [2]). However, due to their rapid development, reviews quickly become outdated. In this paper we present and discuss the most popular infrastructures, platforms and middleware that may be used for Big Data processing, modeling and simulation.

The remainder of this paper is organized as follows. A short survey of infrastructure and middleware for Big Data-related problems is presented in Section 2. Primary attention is focused on computing clusters, grids and clouds. A brief description of job scheduling and load balancing methods and algorithms is presented in Section 3. Widely used Big Data processing, visualization and machine learning platforms are discussed in Sections 4 and 5. Some frameworks for modeling and simulating Big Data problems are described in Section 6. A general overview of approaches to energy-aware computation is presented in Section 7. Finally, conclusions are drawn in Section 8.

2. Infrastructure and Middleware for Big Data Problems

During the past three decades, software for parallel computers focused on providing powerful mechanisms for managing communication between processors and environments for parallel machines and computer networks. High Performance Fortran (HPF), OpenMP, OpenACC, Parallel Virtual Machine (PVM) and Message Passing Interface (MPI) were designed to support communications for scalable applications. Application paradigms were developed to perform calculations on shared memory machines and clusters of machines. Moreover, the architecture of computing nodes may be different (a single processor or a symmetric multiprocessor), and types of methods relied upon to access storage devices may vary as well.

On the other hand, easy access to information offered by the Internet has spawned a new idea, i.e. extending the connection between computers. Thanks to such an approach, distributed resources, including applications, computing power, storage, etc., can be accessed as easily as information on Web pages. The idea was implemented in various forms, but three computing environments have been dominating recently: computing clusters, grids and clouds.

2.1. Computing Cluster

A cluster is a group of cooperating, off-the-shelf commodity computers and resources that serves as one virtual machine [3], [4]. The efficiency of a cluster depends on the speed of processors of separate nodes and the efficiency of network technology. Each computing node may have different characteristics, i.e. may be of the single proces-

sor or the symmetric multiprocessor design, and may offer various types of storage devices. In advanced computing clusters, simple local networks are substituted by very fast communication channels – dedicated networks made up of low latency and high speed switches.

Numerous software tools supporting cluster computing have been developed. Most of them rely on the single system image (SSI) computing paradigm, where a group of computing and storage resources is aggregated and is seen by the user as a single system. SSI technology involves a broad variety of techniques, from custom hardware and hypervisors to dedicated operating systems and user-level tools. A survey of classification schemes and implementation techniques is provided in [1]. Notable hardware, hypervisor and kernel level techniques are discussed. The focus is on distributed operating systems, both dedicated operating systems and adaptations of existing operating systems. MOSIX [5] is one of the oldest and most commonly known SSI kernel patches that provides transparent process migration and automatic load balancing within the cluster. MOSIX does not provide full SSI. All processes which were launched on a given node are displayed, even if they have been moved to remote nodes. However, processes initiated on other nodes are not displayed. The unique space of process IDs is preserved in OpenSSI [6] and Kerrighed [7] systems. OpenSSI allows the migration of transparent processes and groups of threads and enables load balancing within the cluster. Kerrighed provides transparent migration of processes, as well as single threads. All kernel-level implementations mentioned above are accompanied by a complementary suite of user-level software tools.

2.2. Computing Grid

A grid [8] is a collection of loosely coupled, geographically distributed, heterogeneous computational resources and devices. Initially, the idea of a grid was to expand the parallel computing paradigm from tightly coupled clusters onto geographically distributed computing systems. However, in practice, grids are utilized more often as a platform for integration of loosely coupled applications. Nowadays, computational grids enable the sharing and the aggregation of a wide variety of geographically distributed computing resources. Moreover, they present them as a unified resource for solving large-scale computation problems.

The resources that are taken into account in these definitions include the following: computer clusters, supercomputers, databases and storage systems, visualization devices and dedicated software. To facilitate the creation and maintenance of grids, several features have been assumed. The most important of them are: scalability, adaptability, data transfer, process scheduling and allocation, use of open standard protocols, interfaces, and often processes migration. The calculation and data space are in fact heterogeneous, but virtually they are homogeneous. A broad spectrum of grid computing activities and scientific projects have been carried out, e.g. a uniform interface for comput-

ing resources Unicore [9], Globus Toolkit [10] – a software toolkit used for building grids and many others described in literature [11], [12].

2.3. Computing Cloud

In general, both grid and cloud computing offer similar functionalities and serve the same purpose. However, their implementations are different. The cloud computing model offers facilities and common resources, on an on-demand basis, over the Web [13], [14]. A typical cloud computing provider delivers common business applications online. The applications are accessed from a Web service or a Web browser, while the software and data are stored on servers. Most cloud computing infrastructures consist of services delivered through common centers and are built on servers. Cloud often appears as a single point of access for the customers' computing needs. Quality of service (QoS) and service level agreements (SLAs) are generally expected. Cloud computing provides three remarkable services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Resource virtualization is generally at the heart of cloud architectures. The concept of virtualization provides an abstract, logical overview of the physical resources and includes networks, servers and data stores. The basic idea is to join physical resources and manage them as a whole. However, it should be noted that in classic HPC applications, some performance degradation may occur while working in a virtualized environment.

2.4. GPU Computing

A Graphics Processing Unit (GPU) is a specialized massively parallel graphics processor that may be used as a general purpose computing accelerator that is a low-cost, highly accessible alternative to supercomputers [15]. GPU-enabled parallel computing has become extremely popular over the past decade. The GPU-based model for parallel computing has rapidly increased its advance into different areas of technology and is currently used to solve complex scientific and engineering problems. GPUs allow to perform massively parallel computations. Therefore, those computation tasks which may be divided into large numbers of independent parts are of special interest.

This technology may be specifically exploited for massive data processing purposes. Using CPU and GPU jointly, along with CUDA or OpenCL parallel computing platforms, many real-world applications may be easily implemented and are capable of running significantly faster than on multicore or multiprocessor systems. Nowadays, GPU clusters are one of the most progressive branches of HPC. However, problems associated with the optimization of memory management are still experienced when it is necessary to provide fast access to data chunks exceeding the size of the local GPU's memory.

3. Job Scheduling and Load Balancing

Job scheduling and load balancing are one of the most important characteristics of distributed systems. Techniques such as checkpointing, as well as placement and migration of processes allow for transparent load balancing across computing nodes. An efficient load balancing algorithm is triggered when loads of nodes are not balanced or when local resources are limited. Numerous static and dynamic load balancing techniques have been developed for cluster, grid and cloud systems. A survey on load balancing algorithms is provided in [16]–[18].

There are many software tools for supporting job scheduling and management in distributed systems. Torque [19] is a distributed resource manager that provides control over batch jobs and distributed computing resources. Torque allows also to work in an interactive mode. Torque expands the original PBS system [20] offering scalability, fault tolerance and better functionality. It may be integrated with the Moab Cloud smart workload manager that is responsible for load balancing and for optimizing application performance. Torque is customizable to match the needs of the computing system and the specific application. Slurm [21] is a fault-tolerant and highly scalable open source cluster management and job scheduling system for Linux clusters of various sizes. Slurm provides a framework for starting, executing and monitoring allocated jobs. Moreover, the Slurm system manages a queue of pending tasks. MapReduce and YARN represent two different approaches to job scheduling and managing cluster resources. MapReduce [22] is a framework that provides and implements a programming model. It simplifies the processing of massive volumes of data by using two subsequent functions that handle data computations. The MapReduce processing scheme is composed of a map method, which performs data filtering, sorting and splitting, and a reduce method, which performs a summary operation - processes intermediate output data. In fact, the idea is to design data for easy scheduling and cluster management. The YARN resource management and job scheduling technology [23] is more generic than MapReduce. YARN allows multiple data processing engines, such as batch processing, real-time streaming, interactive SQL and data science to handle data stored on a single platform. Unlike MapReduce, YARN enhances efficiency by splitting two main functionalities of the job tracker into two separate daemons responsible for allocation and management of cluster resources, and for task scheduling and monitoring, respectively.

4. Platforms for Big Data Processing

Various software platforms for supporting large scale and massive data distributed processing have been developed during the past decade. Apache Hadoop ([23], [24]), is a framework that enables distributed, scalable processing of large data sets across clusters of computers utilizing a simple programming model. It is designed to scale up from

single servers to many (thousands) machines, each offering local storage and processing capabilities. The power of the Hadoop platform is based on the Hadoop Distributed File System (HDFS), the HBase distributed and scalable non-relational database and programming models. Hadoop delivers a highly-available service on top of a cluster of computers, each of which may be prone to failures. The failures are detected and handled at the application layer. The Hadoop community has contributed to enriching its ecosystem both with open source projects and a wide range of commercial tools and solutions. Some of the best-known open source examples include:

- Pig – a framework for the generation of a high level scripting language (Pig Latin),

- Hive – a data warehouse system that is designed to simplify the use of Hadoop frameworks,

- JAQL – a declarative language designed to convert high level queries into MapReduce jobs,

- Sqoop – software that provides a command-line interface and moves relational data into HDFS,

- Oozie – a workflow scheduler system,

- Mahout – a framework for scalable machine learning, etc.

A variety of commercial tools can be used for specific Hadoop development, production, and maintenance tasks. Hadoop is designed for batch processing.

Apache Spark [25] is a unified engine for Big Data processing. Spark can run in a standalone mode or with a Hadoop cluster serving as the data source. It is both a programming and computing model. Spark provides an alternative to MapReduce that enables workloads to be executed in memory, instead of on disk, thus eliminating resource-intensive disk operations that MapReduce requires. It processes data in RAM. The implemented data model is based on the Resilient Distributed Dataset (RDD) abstraction. The Spark framework consists of components for memory management, fault recovery, data exchange, task scheduling, etc. The main Apache Spark use cases include the following: streaming data, machine learning, fog computing, etc.

Apache Storm [26] is a scalable, rapid, fault-tolerant and easy-to-use platform for distributed computing that has, unlike Hadoop, the advantage of handling real time data processing. A Storm interface may potentially support any incoming data, hence data from real time synchronous and asynchronous systems can be downloaded. It can process one million tuples per second using a simple programming model and hiding the complexity of the Big Data application. Typical use cases include real-time analytics, online machine learning, IoT, continuous computation, etc.

Apache Flink [27] is a framework for batch and stream processing, event-time processing and stateful computations. It can run in all common cluster environments, perform computations at in-memory speed and on a high scale. Simi-

larly to Storm, it may be successfully used to develop software systems for fraud and anomaly detection, monitoring, as well as real-time and discrete-event simulation.

5. Platforms for Big Data Visualization and Machine Learning

A number of tools for Big Data analysis, visualization and machine learning are available in the network. RapidMiner Studio [28], Orange [29] and Weka [30] belong to this group of solutions. Numerous novel applications have been designed and developed for browsing, interpreting, visualizing and analyzing large-scale sequencing data. Several of these, including Tablet [31] have been designed specifically for the visualization of genome sequence assemblies. Other tools, such as BamView [32] have been developed specifically to visualize mapped read alignment data in the context of the reference sequence. Artemis is a freely available integrated platform for visualization and analysis of large-scale experimental data. It is an established genome annotation tool [33] that has been used in many genome projects. It is an effective tool for visualization, analysis, interpretation and inspection of high-throughput sequence-based experimental data [34].

Plenty of platforms and packages have been developed for social network analysis and visualization. The survey of most popular ones, which find a wide range applications, including network theory, finance, biology, sociology, etc., is presented in [35].

6. Simulation Frameworks

Simulation of large scale systems is another issue requiring attention. Simulation and Big Data analytics produce the most value when used together. Methods for Big Data analytics process simulation data, extract valuable information and convert it into proper decisions or predictions of future behavior – all in a short period of time. The combination of efficient and reliable simulation software and specialized (purpose-built) hardware optimized for simulation workloads is crucial to fully exploit the value of simulating Big Data problems. The simulation power can be increased by deploying both HPC infrastructures and computing models that enable fast job execution and deliver the highest possible computing performance for the simulation workloads. Synchronous and asynchronous distributed simulation is one of the options that may improve the scalability of a simulator, both in terms of application size and execution speed, enabling large-scale systems to be simulated in real time [36], [37].

Investigations conducted in the field the development of modern simulation technologies have led to the design of general-purpose and problem-oriented software tools for Big Data systems simulations. ScalaTion [38] serves as a modeling and simulation testbed. It provides comprehensive support for discrete event simulation, and offers

an easy-to-use framework for Big Data analytics and many optimization solvers that may be successfully used to solve simulation-optimization problems. A software framework for federated simulation of WSN and mobile ad-hoc networks is presented in [39]. Paper [40] reviews several large-scale military simulations producing Big Data and describes two software frameworks for simulation and Big Data management, based on layered and service-oriented architectures.

GPU-based simulation platforms are mainly dedicated to massive data processing, e.g. high performance neural network simulators [41], [42], simulation of P systems – computational models that perform calculations using a biologically-inspired process [43], large scale volume of data visualization [44], and more.

Some software platforms have been designed to simulate large-scale distributed data centers and computer networks. Jade [45] is a heterogeneous multiprocessor design simulation environment that allows to simulate inter-chip networks, network-on-chips and intra-rack networks utilizing optical and electrical interconnects. Jade supports memory hierarchy, cache coherence and low-power technologies. SimGrid [46] can be used to simulate grids, clouds, HPC or P2P systems, as well as to evaluate heuristics or prototype applications. CloudSim [47], [13] is a Java framework for modeling and simulating cloud computing infrastructures and services. It is one of the most popular open source cloud simulators in the research and academia community. Multi2Sim [48] is a software platform for simulation of CPU and GPU technologies, used to test and validate new hardware designs before they are physically manufactured.

7. Energy-Aware Infrastructure for HPC Computing

7.1. Energy Conservation and HPC Computing

Nowadays, energy efficiency in all sectors, including HPC infrastructure, is a key part of European energy policies for the upcoming decade [49], [50]. Energy awareness is an important aspect of design and management processes involving large-scale data centers. Over the past decade, the amount of energy used by data centers has grown rapidly. This stems primarily from the growing demand for HPC services and computing clouds. Thus, progress in HPC systems depends on energy-efficient data mining and computing technologies.

Indeed, data centers, supporting both HPC applications and cloud services, consume enormous amounts of energy. Over the period of five years (2005–2010), the amount of energy consumed by data centers has increased by 56%, which accounts to 1.5% of the total electrical energy used in 2010. Growing energy consumption increases operating costs of data centers and contributes CO_2 (carbon dioxide) production. According to the analysis of current trends presented and discussed in [51], CO_2 emissions generated by the ICT industry are expected to exceed 2% of the global

emission levels [52], [53]. Data centers are very energy intensive. Typical power densities in commodity data centers equal 538–2153 W/m^2 [54], and are even higher in the case of classic HPCs, exceeding 5000 W/m^2 [55]. High energy consumption is attributed primarily to the computing and networking demands and to cooling equipment. The cooling system may use, on average, up to 40% of the energy consumed by a given data center.

Energy efficiency (FLOPS/W) of ICT systems continues to improve. However, the rate of improvement does not match the growing demand for large scale computing. Unless new energy-aware technologies are introduced, both in hardware and software domains, it seems that it will not be possible to meet DARPA's 20-MW exaflop goal (50 GFLOPS/W) by the year 2020 (50 GFLOPS/W) by the year 2020 [3]. Computational power improvements are, in fact, heavily constrained by energy budgets that are necessary for driving computing grids, clouds and data centers. Limiting power consumption and related thermal emissions has become a key problem. Based on technology development-related projections, it has been argued that the continued scaling of the available systems will eventually lead to data centers consuming more than a gigawatt of electrical power (at Exaflop level), a value that violates the economic rationale for the provision of cloud or HPC services. Optimization of energy consumption in data centers must be addressed in response to environment protection and market needs.

As an answer to the momentum that has been created, considerable research efforts devoted to energy-efficient computing and networking technologies have been undertaken both in the research domain and on the ICT market. The rapid increase in energy demand, generated by data centers and network infrastructures, may be mitigated on software and hardware levels. According to [56], the following interrelated approaches and solutions can be distinguished:

- design and development of energy-efficient hardware components (CPUs/GPUs, disks, memory units, network interface cards, etc.),

- development of energy-saving systems for controlling hardware components (servers, routers, etc.),

- introduction of energy-efficient control frameworks for task scheduling and workload balancing.

New computing components, i.e. CPUs/GPUs, disks, memory units, network line cards, have been developed to operate in multiple (performance and idle) modes and at differentiated energy consumption levels (ACPI). Mode switching and high-frequency performance monitoring functions have also been exposed by co-designed Application Program Interfaces (API) [57]. Development of APIs and management tools is essential for optimized use of computing resources. On the other hand, system-wide regulation of power consumption needs to be controlled by a centralized management framework, capable of collecting and processing energy consumption measurements,

and taking, in real time, coordinated actions across the data center infrastructure.
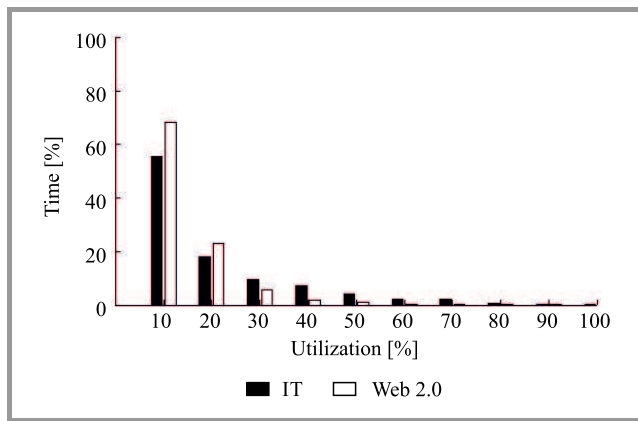

Fig. 1. Server utilization histogram [58].

According to [58], data center server utilization rate rarely approaches 100% (see Fig. 1). Most of the time the servers operate at 10–50% of their full capacity. This results from the requirements of providing sufficient quality of service (QoS) provisioning margins. The over-subscription of computing resources is applied as a sound strategy to eliminate potential breakdowns caused by traffic fluctuations or internal disruptions stemming from hardware or software faults. A fair amount of slack capacity is also required for the purpose of performing maintenance tasks. However, the strategy of resource over-provisioning is clearly a source of energy waste, i.e. the provisioned power supply is less than the sum of the possible peak power demands of all the servers [59]. This highlights the problem of distributing power throughout the entire data center. To keep the total power consumption within the available power range, servers are equipped with power (ACPI-based) budgeting mechanisms capable of limiting their power use. The challenge of energy-efficient data center or cloud control is, therefore, to design a control structure improving the utilization of computing resources and reducing energy usage in accordance with QoS constraints in a highly stochastic environment, capable of providing fast responses to fluctuations in application workloads. To reduce energy consumption, the control system is required to dynamically deactivate and reactivate (by switching between low-power modes) physical computing elements (CPU/GPU, memory, interconnect) to meet the observable resource demand.

Moreover, data intensive computing, and especially Big Data processing, requires advanced methodologies to efficiently allocate resources (CPU, memory and network capacity) to user applications. In general, the idea is to reduce the gap between the capacity provided by data centers and networking environments, and the requirements of users, especially during low workload periods. Nowadays, the main challenge is to arrange and adapt all available methods and techniques to develop energy-efficient and flexible power control systems encompassing all elements of cluster, grid and cloud infrastructures.

7.2. Energy-Saving Technologies

Improvement of energy efficiency of data centers and network infrastructures includes, as it has been already mentioned, optimization of performance of CPU units and network interfaces. For an overview of recent design trends and a detailed discussion on related technical issues concerned with the energy conservation approaches and solutions, see e.g. [56], [60]–[63]. A commonly used direction is to apply novel measurement technologies and utilize assessment of energy consumption characteristics exposing power management functionality through APIs. A detailed study of energy monitoring mechanisms for data centers can be found in [64]. A discussion of power consumption identification problems has been presented in [65], [66].
A survey of memory power management has been presented in [67], [68].
The second approach focuses on control systems and mechanisms adjusting performance of devices to their short-term workload. Two main technologies are distinguished in this context, namely, low power idle and service rate adaptation. They are discussed in [69], [66]. The first one allows a device to switch off for a short period of time whenever there is no workload to be processed, while the other one allows a device to lower its service rate – dynamic voltage frequency scaling (DVFS) mechanisms can be used whenever reduced workloads are observed. The 802.3az Ethernet standard [4] is an example of the implementation of the low power idle technology. The service rate adaptation of Ethernet links has been presented and discussed in [70]. Two mechanisms implementing low power idle and adaptive rate concepts to control the performance of CPUs are delivered by Linux kernel:

- cpuidle governor [71],

- cpufreq governor [61], [56], [72].

In [73], a design of a feedback controller for solving the problem of low utilization of servers in a data-center running I/O-intensive applications is proposed. To adjust CPU frequency, the controller relies on energy-related system-wide feedback rather than on CPU utilization levels. A technique to reduce memory bus and memory bank contention by DVFS-based control of thread execution on each core is presented in [74]. The process model identification technique applied for the purpose of designing CPU frequency control mechanisms has been presented in [61]. A feedback control design methodology that leads to stochastic minimization of performance loss is described in [75]. The optimal design of a controller is formulated as a problem of stochastic minimization of runtime performance error for selected classes of applications. A supervised learning technique is used to predict the performance state of the processor for each incoming job and to reduce the overhead of state observation (see [76]). A hardware-level implementation and performance of power management mechanisms, allowing for independent DVFS of the cores of a multi-core processor that integrates 48 cores and 4 DDR3 memory channels is given in [77].
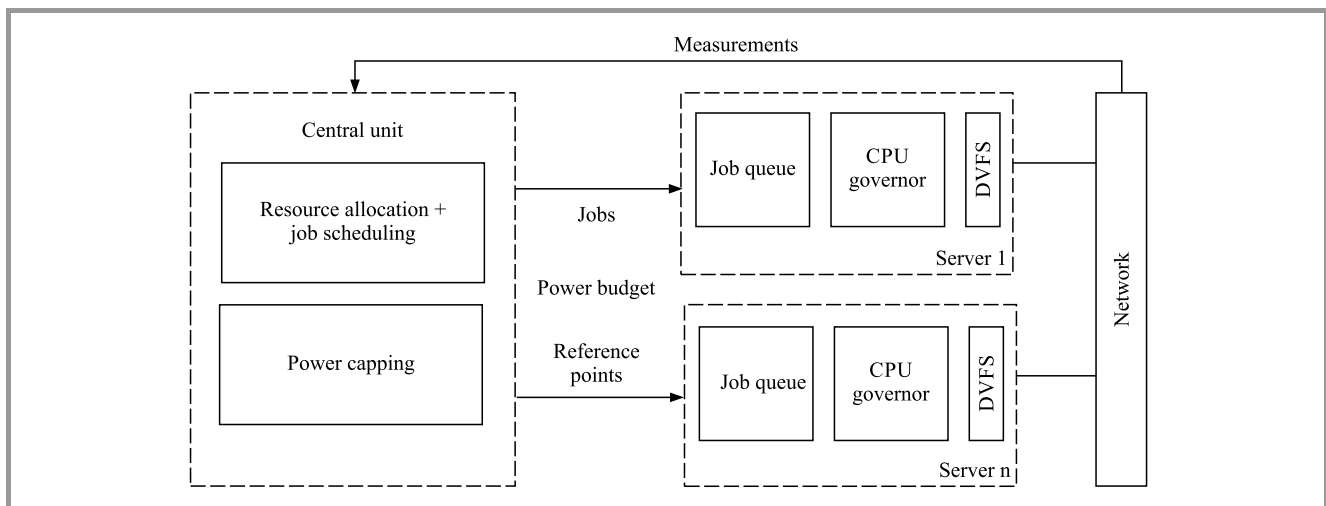
Fig. 2. General control framework architecture.

A DVFS technique that makes use of adaptive update intervals for optimal frequency and voltage scheduling has been proposed in [78]. An optimized control strategy was developed to meet the workload processing deadlines. Papers [61], [79], and [56] address issues concerned with the structure of optimized energy-aware CPU frequency scaling rules. A class of CPU frequency switching rules, exploiting DVFS, is discussed. A benchmarking methodology derived from the RFC2544 specification for identification of models of CPU workload dynamics is proposed and discussed.

Next, it is demonstrated how the proposed models can be applied in the design of customized energy-aware controllers that dynamically adjust CPU frequency to application-specific workload patterns. The numerous experiments referred to above show that customized controllers may outperform standard general-purpose governors of the Linux kernel, both in terms of server performance and power saving capabilities.

The third approach to energy conservation in HPC systems has been focusing on the development of mechanisms for energy management in data centers and networks linking computing nodes. Performance metrics for green data centers have been discussed in [80], [64]. It is believed that considerable energy savings may be achieved in this scenario if only operations of computing and network devices are coordinated and adjusted to workload and traffic patterns observed. In some scenarios, the energy consumed by clusters and network infrastructures may be minimized by switching off or idling servers, routers or line cards [81], [82]. The ability to control the activity of computing nodes is provided by various tools and platforms described in Sections 3 and 4, for instance by Slurm. Moreover, various efforts have been undertaken to develop energy-efficient task scheduling, load balancing and green routing protocols [83]–[85], [72]. However, optimized task scheduling and allocation becomes much more difficult with the classic makespan criterion with energy-efficiency and user-perceived QoS [86], [87] taken into consideration.

Figure 2 shows an overview of a general computer system for an energy-aware data center. The system introduces dynamic power and performance control technologies, based on standby and performance scaling capabilities, improving energy efficiency of computing devices. The idea is to combine the ability of setting the adequate energy states of the computing devices using CPU frequency switching rules, exploiting DVFS implemented along with a control mechanism for optimization of the allocation of resources used by the central unit. The knowledge of the expected computing workload and network traffic matrix leads to considerable and comprehensive optimization problem formulations that have to be solved by the central dispatcher.

A literature survey of green cloud computing is provided in [88]. The paper demonstrates the main achievements in energy-aware computing clouds. The recent developments are summarized, as well as future research directions and open problems related to green cloud computing are presented. Other novel solutions in the field of cluster, grid and cloud energy use optimization are surveyed in [89], [90].

8. Summary and Conclusions

HPC facilities and technologies are required in a rapidly increasing number of data-intensive domains – from life and physical sciences to socioeconomic systems. Thus, the Big Data era offers HPC striking opportunities to expand its range and to strengthen its societal and economic impact. A broad spectrum of activities concerned with the development of HPC infrastructure and middleware for solving Big Data problems have been undertaken. Much of the research has been devoted to the development of methods, algorithms and techniques for energy conservation in data centers, grids, clouds and computer networks. To meet the increasing demand for computing power, a holistic approach to energy-aware design of hardware, middleware and data processing applications is proposed. This paper

presents an overview of middleware, software platforms and simulation platforms for solving Big Data problems. Particular attention is paid to techniques developed to improve energy efficiency of HPC infrastructure. We have taken a look at power-saving algorithms utilizing low power idle and service rate adaptation mechanisms and algorithms for task scheduling and energy-efficient load balancing. However, although numerous energy conservation strategies and systems have been proposed and described in literature, development of scalable, energy-efficient infrastructures for HPC still remains a challenging task.

Acknowledgments

References

[1] P. D. Healy, T. Lynn, E. Barrett, and J. P. Morrison, "Single system image: A survey", *J. Parallel Distrib. Comput.*, vol. 90–91, pp. 35–51, 2016 (10.1016/j.jpdc.2016.01.004).

[2] A. Oussous, F. Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "Big data technologies: A survey", *J. of King Saud Univer. – Comp. and Inform. Sci.*, vol. 30, no. 4, pp. 431–448, 2018 (doi: 10.1016/j.jksuci.2017.06.001).

[3] ETP4HPC Strategic Research Agenda achieving HPC leadership in Europe [Online]. Available: www.etp4hpc.eu

[4] IEEE 802.3az-2010 – IEEE standard for information technology [Online]. Available: https://standards.ieee.org/standard/802_3az-2010.html

[5] Mosix home page [Online]. Available: www.mosix.org

[6] OpenSSI home page [Online]. Available: www.openssi.org/cgi-bin/view?page=openssi.html

[7] Kerrighed home page [Online]. Available: www.kerrighed.org

[8] F. Berman, G. Fox, and A. J. G. Hey, Eds., *Grid Computing: Making the Global Infrastructure a Reality*. Wiley, 2003 (ISBN: 978-0-470-85319-1).

[9] Unicore home page [Online]. Available: www.unicore.eu

[10] Globus toolkit home page [Online]. Available: toolkit.globus.org

[11] M. Cannataro, *Handbook of Research on Computational Grid Technologies for Life Sciences, Biomedicine, and Healthcare*. Hershey, PA, USA: IGI Global, 2009 (ISBN-13: 978-1605663746).

[12] R. J. Walters, S. Crouch, and P. Bennett, "Building computational grids using ubiquitous Web technologies", in *Collaborative Networks in the Internet of Services. 13th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2012, Bournemouth, UK, October 1-3, 2012. Proceedings*, L. M. Camarinha-Matos, L. Xu, and H. Afsarmanesh, Eds. *IFIPAICT*, vol. 380, pp. 254–261. Berlin, Heidelberg: Springer, 2012 (doi: 10.1007/978-3-642-32775-9_26).

[13] S. Chaudhary, G. Somani, and R. Buyya, Eds., *Research Advances in Cloud Computing*. Springer, 2017 (doi: 10.1007/978-981-10-5026-8).

[14] N. Sehgal and P. Ch. P. Bhatt, *Cloud Computing. Concepts and Practices*. Springer, 2018 (ISBN: 978-3-319-77839-6).

[15] W.-M. Hwu, Ed., *GPU Computing Gems Emerald Edition*. Morgan Kaufman, 2011 (ISBN: 9780123849885).

[16] A. B. Singh, J. S. Bhat, R. Raju, and R. D'Souza, "Survey on various load balancing techniques in cloud computing", *Adv. in Comput.*, vol. 7, no. 2, pp. 28–34, 2017 (doi: 10.5923/j.ac.20170702.04).

[17] A. Thakur and M. S. Goraya, "A taxonomic survey on load balancing in cloud", *J. of Netw. and Comp. Appl.*, vol. 98, pp. 43–57, 2017 (doi: 10.1016/j.jnca.2017.08.020).

[18] J. Zhang *et al.*, "Load balancing in data center networks: A survey", *IEEE Commun. Surv. Tutor.*, vol. 20, no. 3, pp. 2324–2352, 2018 (doi: 10.1109/COMST.2018.2816042).

[19] G. Staples, "TORQUE resource manager", in *Proc. of the 2006 ACM/IEEE Conf. on Supercomput. SC'06*, Tampa, FL, USA, 2006, Article no. 8 (doi: 10.1145/1188455.1188464).

[20] Portable batch system home page [Online]. Available: www.pbspro.org

[21] Slurm workload manager home page [Online]. Available: slurm.schedmd.com

[22] E. Mohamed and Z. Hong, "Hadoop-mapreduce job scheduling algorithms survey", in *Proc. 2016 7th Int. Conf. on Cloud Comput. and Big Data CCBD 2016*, Macau, China, 2016, pp. 237–242, 2016 (doi: 10.1109/CCBD.2016.054).

[23] T. White, *Hadoop: The Definitive Guide*. O'Reilly Media, 2015 (ISBN: 9781491901687).

[24] Apache Hadoop home page [Online]. Available: https://hadoop.apache.org

[25] Apache Spark home page [Online]. Available: spark.apache.org

[26] Apache Storm home page [Online]. Available: storm.apache.org

[27] Apache Flink home page [Online]. Available: https://flink.apache.org

[28] Rapidminer studio home page [Online]. Available: https://rapidminer.com/

[29] Orange home page [Online]. Available: orange.biolab.si

[30] E. Frank, M. A. Hall, and I. H. Witten, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016 (ISBN: 9780123748560).

[31] I. Milne, M. Bayer, L. Cardle, P. Shaw, G. Stephen, F. Wright, and D. Marshall, "Tablet – next generation sequence assembly visualization", *Bioinformatics*, vol. 26, no. 3, pp. 401–403, 2010 (doi: 10.1093/bioinformatics/btp666).

[32] T. Carver, T. D. Bohme, U. Otto, J. Parkhill, and M. Berriman, "Bamview: viewing mapped read alignment data in the context of the reference sequence", *Bioinformatics*, vol. 26, no. 5, pp. 676–673, 2010 (doi: 10.1093/bioinformatics/btq010).

[33] K. Rutherford *et al.*, "Artemis: sequence visualization and annotation", *Bioinformatics*, vol. 16, no. 10, pp. 944–949, 2000 (doi: 10.1093/bioinformatics/16.10.944).

[34] T. Carver, S. R. Harris, M. Berriman, J. Parkhill, and J. A. McQuillan, "Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data", *Bioinformatics*, vol. 28, no. 4, pp. 464–469, 2012 (doi: 10.1093/bioinformatics/btr703).

[35] D. Desale, Top tools for social network analysis and visualisation, 2018 [Online]. Available: https://www.kdnuggets.com/software/social-network-analysis.html

[36] A. Sikora and E. Niewiadomska-Szynkiewicz, "A federated approach to parallel and distributed simulation of complex systems. *Int. J. of Appl. Mathem. and Comp. Sci.*, vol. 17, no. 1, pp. 99–106, 2007 (doi: 10.2478/v10006-007-0009-0).

[37] A. Inostrosa-Psijas, V. Gil-Costa, M. Marin, and G. Wainer, "Semi-asynchronous approximate parallel DEVS simulation of Web search engines", *Concurr. and Comput.: Pract. and Exper.*, vol. 30, no. 7, 2018 (doi: 10.1002/cpe.4149).

[38] J. A. Miller, M. E. Cotterell, and S. J. Buckley, "Supporting a modeling continuum in scalation: from predictive analytics to simulation modeling", in *Proc. of 2013 Winter Simulations Conference WSC 2013*, Washington, DC, USA, 2013, pp. 1191–1202 (doi: 10.1109/WSC.2013.6721507).

[39] E. Niewiadomska-Szynkiewicz and A. Sikora, "A software tool for federated simulation of Wireless Sensor Networks and mobile ad hoc networks", in *Applied Parallel and Scientific Computing 10th International Conference, PARA 2010, Reykjavík, Iceland, June 6-9, 2010, Revised Selected Papers, Part I*, K. Jónasson, Ed. *LNCS*, vol. 7133, pp. 303–313. Berlin, Heidelberg: Springer, 2012 (doi: 10.1007/978-3-642-28151-8_30).

[40] X. Song, Y. Wu, Y. Ma, Y. Ciu, and G. Gong, "Military simulation big data: Background, state of the art, and challenges", *Mathem. Problems in Engin.*, vol. 2015, Article ID 298356, pp. 1–20, 2015 (doi: 10.1155/2015/298356).

[41] A. K. Fidjeland, E. B. Roesch, M. P. Shanahan, and W. Luk, "Nemo: A platform for neural modelling of spiking neurons using GPUS", in *Proc. 2009 20th IEEE Int. Conf. on Appl.-specif. Syst., Architec. and Process.*, Boston, MA, USA, 2009, pp. 137–144 (doi: 10.1109/ASAP.2009.24).

[42] P. Szynkiewicz, "A novel GPU-enabled simulator for large scale spiking neural networks", *J. of Telecommun. and Inform. Technol.*, no. 2, pp. 34–42, 2016 [Online]. Available: https://www.itl.waw.pl/czasopisma/JTIT/2016/2/34.pdf

[43] M. A. Martinez-del Amor *et al.*, "Accelerated simulation of P systems on the GPU: A survey", in *Bio-Inspired Computing – Theories and Applications. 9th International Conference, BIC-TA 2014, Wuhan, China, October 16-19, 2014. Proceedings*, L. Pan, G. Păun, M. J. Pérez-Jiménez, and T. Song, Eds. *Communications in Computer and Information Science*, vol. 472, pp. 308–312. Springer, 2014 (doi: 10.1007/978-3-662-45049-9_50).

[44] J. Beyer, M. Hadwiger, and H. Pfister, "A survey of GPU-based large-scale volume visualization" in *Proc. of the Eurograph. Conf. on Visual. Eurovis 2014*, Swansea, UK, 2014, pp. 1–19 (doi: 10.2312/eurovisstar.20141175).

[45] R. K. V. Maeda *et al.*, "JADE: a heterogeneous multiprocessor system simulation platform using recorded and statistical application models", in *Proc. of the 1st Int. Worksh. on Adv. Interconn. Solutions and Technol. for Emerg. Comput. Syst. AISTECS'16*, Prague, Czech Republic, 2016 (doi: 10.1145/2857058.2857066).

[46] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, scalable, and accurate simulation of distributed applications and platforms", *J. of Parallel and Distrib. Comput.*, vol. 74, no. 10, pp. 2899–2917, 2014 (doi: 10.1016/j.jpdc.2014.06.008).

[47] 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", *Software: Pract. and Exper. (SPE)*, vol. 41, no. 1, pp. 23–50, 2011 (doi: 10.1002/spe.995).

[48] Multi2sim workload manager home page [Online]. Available: www.multi2sim.org

[49] Energy Efficiency [Online]. Available: https://ec.europa.eu/energy/en/topics/energy-efficiency

[50] Code of Conduct for Energy Efficiency in Data Centres [Online]. Available: https://ec.europa.eu/jrc/en/energy-efficiency/code-conduct/datacentres

[51] Carbon Abatement Handbook [Online]. Available: https://gesi.org/report/detail/carbon-abatement-handbook

[52] M. Avgerinou, P. Bertoldi, and L. Castellazzi, "Trends in data centre energy consumption under the European code of conduct for data centre energy efficiency", *Energies*, vol. 10, no. 10, pp. 1–18, 2017 (doi: 10.3390/en10101470).

[53] B. Subramaniam, W. Saunders, T. Scogland, and W. Feng, "Trends in energy-efficient computing: A perspective from the Green500", in *2013 Int. Green Comput. Conf. Proc.*, Arlington, VA, USA 2013, pp. 1–8 (doi: 10.1109/IGCC.2013.6604520).

[54] D. L. Beaty, "Internal IT load profile variability", *ASHRAE J.*, vol. 55, no. 2, pp. 72–74, 2013.

[55] J. S. Vetter, *Contemporary High Performance Computing: From Petascale toward Exascale*. Chapman and Hall/CRC Computational Science series, CRC Press, 2013 (ISBN: 9781466568341).

[56] M. P. Karpowicz, P. Arabas, and E. Niewiadomska-Szynkiewicz, "Energy-aware multilevel control system for a network of Linux software routers: Design and implementation", *IEEE Systems J.*, vol. 12, no. 1, pp. 571–582, 2018 (doi: 10.1109/JSYST.2015.2489244).

[57] ETSI ES 203 237 v1.1.1 (2014-03) standard [Online]. Available: www.etsi.org

[58] L. A. Barroso and U. Holzle, "The case for energy-proportional computing", *Computer*, vol. 40, no. 12, pp. 33–37, 2007 (doi: 10.1109/MC.2007.443).

[59] H. Lim, A. Kansal, and J. Liu, "Power budgeting for virtualized data centers", in *Proc. of the 2011 USENIX Ann. Tech. Conf. USENIX ATC'11*, Portland, OR, USA, 2011 [Online]. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2011/06/VPSUsenix11.pdf

[60] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks", in *Proc. of the 2009 IEEE Int. Conf. on Commun. ICC'09*, Piscataway, NJ, USA, 2009, pp. 2298–2303 (doi: 10.1109/ICC.2009.5199404).

[61] M. Karpowicz, "Energy-efficient CPU frequency control for the Linux system", *Concurr. and Comput.: Pract. and Exper.*, vol. 28, no. 2, pp. 420–437, 2016 (doi: 10.1002/cpe.3476).

[62] R. Bolla *et al.*, "Large-scale validation and benchmarking of a network of power-conservative systems using ETSI's green abstraction layer", *Trans. on Emerg. Telecommun. Technol.*, vol. 27, no. 3, pp. 451–468, 2016 (doi: 10.1002/ett.3006).

[63] P. Arabas, "Energy aware data centers and networks: a survey", *J. of Telecommun. and Inform. Technol.*", no. 4, pp. 26–36, 2019 (doi: 10.26636/jtit.2018.129818).

[64] A. Y. Zomaya and J. Ch. Lee, Eds., *Energy-Efficient Distributed Computing Systems*. Wiley, 2012 (ISBN: 978-0-470-90875-4).

[65] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, "Evaluating the effectiveness of model-based power characterization", in in *Proc. of the 2011 USENIX Ann. Tech. Conf. USENIX ATC'11*, Portland, OR, USA, 2011 [Online]. Available: https://www.synergylabs.org/yuvraj/docs/Agarwal_USENIX11_Evaluating-Power-Models.pdf

[66] J.-M. Pierson, *Large-scale Distributed Systems and Energy Efficiency: A Holistic View*. Wiley, 2015 (ISBN: 9781118864630).

[67] S. Mittal, "A survey of architectural techniques for DRAM power management", *Int. J. of High Perform. Syst. Archit.*, vol. 4, no. 2, pp. 110–119, 2012 (doi: 10.1504/IJHPSA.2012.050990).

[68] K. Chalmers *et al.*, Eds., *Communicating Process Architectures 2015 & 2016: WoTUG-37 & WoTUG-38*. Concurrent Systems Engineering Series, vol. 69. IOS Press, 2018 (ISBN 978-1-61499-885-3).

[69] R. Bolla, R. Bruschi, A. Carrega, and F. Davoli, "Green network technologies and the art of trading-off", in *Proc. 2011 IEEE Conf. on Comp. Commun. Worksh. INFOCOM WKSHPS 2011*, Shanghai, China, 2011, pp. 301–306 (doi: 10.1109/INFCOMW.2011.5928827).

[70] V. Pallipadi, S. Li, and A. Belay, "cpuidle: Do nothing, efficiently...", in *Proc. Linux Symposium*, Ottawa, Ontario, Canada, 2007, vol. 2, pp. 119–125.

[71] V. Pallipadi and A. Starikovskiy, "The ondemand governor", in *Proc. Linux Symposium*, Ottawa, Ontario, Canada, 2006, vol. 2, pp. 215–230.

[72] M. Karpowicz, E. Niewiadomska-Szynkiewicz, P. Arabas, and A. Sikora, "Energy and power efficiency in cloud", 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. 97–127 (doi: 10.1007/978-3-319-44881-7_6).

[73] I. Manousakis, M. Marazakis, and A. Bilas, "FDIO: A feedback driven controller for minimizing energy in I/O-intensive applications", in *Proc. of the 5th USENIX Conf. on Hot Topics in Storage and File Syst. HotStorage'13*, San Jose, CA, USA 2013 [Online]. Available: https://www.usenix.org/system/files/conference/hotstorage13/hotstorage13-manousakis.pdf

[74] M. Kondo, H. Sasaki, and H. Nakamura, "Improving fairness, throughput and energy-efficiency on a chip multiprocessor through DVFs", *SIGARCH Comp. Archit. News*, vol. 35, no. 1, pp. 31–38, 2007 (doi: 10.1145/1241601.1241609).

[75] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges", *J. of Internet Serv. and Appl.*, vol. 1, no. 1, pp. 7–18,

[76] H. Jung and M. Pedram, "Supervised learning based power management for multicore processors", *IEEE Trans. on Comp.-Aided Design of Integr. Circ. and Syst.*, vol. 29, no. 9, pp. 1395–1408, 2010 (doi: 10.1109/TCAD.2010.2059270).

[77] J. Howard *et al.*, "A 48-core IA-32 processor in 45 nm CMOS using on-die message-passing and DVFs for performance and power scaling", *IEEE J. of Solid-State Circ.*, vol. 46, no. 1, pp. 173–183, 2011 (doi: 10.1109/JSSC.2010.2079450).

[78] M. E. Salehi *et al.*, "Dynamic voltage and frequency scheduling for embedded processors considering power/performance tradeoffs", *IEEE Trans. on Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 10, pp. 1931–1935, 2011 (doi: 10.1109/TVLSI.2010.2057520).

[79] M. Karpowicz, P. Arabas, and E. Niewiadomska-Szynkiewicz, "Design and implementation of energy-aware application-specific CPU frequency governors for the heterogeneous distributed computing systems", *Future Gener. Comp. Syst.*, vol. 78, pp. 302–315, 2018 (doi: 10.1016/j.future.2016.05.011).

[80] L. Wang and S. U. Khan, "Review of performance metrics for green data centers: a taxonomy study", *J. of Supercomput.*, vol. 63, no. 3, pp. 639–656, 2003 (doi: 10.1007/s11227-011-0704-3). 2010 (doi: 10.1007/s13174-010-0007-6).

[81] I. T. Cotes-Ruiz *et al.*, "Dynamic voltage frequency scaling simulator for real workflows energy-aware management in green cloud computing", *PLoS ONE*, vol. 12, no. 1, 2017 (doi: 10.1371/journal.pone.0169803).

[82] Y. Chiang, Y. Ouyang, and C. Hsu, "An efficient green control algorithm in cloud computing for cost optimization", *IEEE Trans. on Cloud Comput.*, vol. 3, no. 2, pp. 145–155, 2015 (doi: 10.1109/TCC.2014.2350492).

[83] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, M. Kamola, M. Mincer, and J. Kołodziej, "Dynamic power management in energy-aware computer networks and data intensive systems", *Future Gener. Comp. Syst.*, vol. 37, pp. 284–296, 2014 (doi: 10.1016/j.future.2013.10.002).

[84] J. Kołodziej, S. Khan, L. Wang, and A. Zomaya, "Energy efficient genetic-based schedulers in computational grids", *Concurr. and Comput.: Pract. and Exper.*, vol. 27, no. 4, pp. 809–829, 2015 (doi: 10.1002/cpe.2839).

[85] M. Kamola and P. Arabas, "Shortest path green routing and the importance of traffic matrix knowledge", in *Proc. 2013 24th Tyrrhenian Int. Worksh. on Digit. Commun. – Green ICT TIWDC 2013*, Genoa, Italy, 2013 (doi: 10.1109/TIWDC.2013.6664215).

[86] K. Govindarajan, V. S. Kumar, and T. S. Somasundaram, "A distributed cloud resource management framework for high-performance computing (HPC) applications", in *Proc. 2016 8th Int. Conf. on Adv. Comput. ICoAC 2017*, Chennai, India, 2017, pp. 1–6 (doi: 10.1109/ICoAC.2017.7951735).

[87] E. Niewiadomska-Szynkiewicz and P. Arabas, "Resource management system for HPC computing", in *Automation 2018. Advances in Automation, Robotics and Measurement Techniques*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. *Advances in Intelligent Systems and Computing*, vol. 743, pp. 52–61. Springer, 2018 (doi: 10.1007/978-3-319-77179-3_5).

[88] L.-D. Radu, "Green cloud computing: A literature survey", *Symmetry*, vol. 9, no. 12, pp. 1–20, 2017 (doi: 10.3390/sym9120295).

[89] N. Akhter and M. Othman, "Energy aware resource allocation of cloud data center: review and open issues", *Cluster Comput.*, vol. 19, no. 3, pp. 1163–1182, 2016 (doi: 10.1007/s10586-016-0579-4).

[90] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. Vasilakos, "Cloud computing: survey on energy efficiency", *ACM Comput. Surv.*, vol. 47, no. 2, Article no. 33, 2015 (doi: 10.1145/2656204).

Ewa Niewiadomska-Szynkiewicz, D.Sc., (2005), Ph.D., (1995), Professor of Control and Computer Engineering at the Warsaw University of Technology, head of the Complex Systems Group. She is also the Research Director of the Research and Academic Computer Network (NASK). Niewiadomska-Szynkiewicz is the author and co-author of over 160 journal and conference papers. Her research interests focus on complex systems modeling, optimization and control, computer simulation, parallel computation, computer networks and ad-hoc networks. She was involved in a number of research projects including EU projects, coordinated group activities and has overseen the organization of a number of national-level and international conferences.

https://orcid.org/0000-0003-4782-3816
E-mail: ens@ia.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska 15/19
00-665 Warsaw, Poland

Michał P. Karpowicz, Ph.D. (2010), Assistant Professor of Computer Science with Warsaw University of Technology, Warsaw, and NASK Research Institute. He is the author and co-author of over 50 journal and conference papers and is involved in several national and international research projects. He is also a coauthor of one book. His research interests focus on stochastic control theory, control engineering, game theory, network optimization and cybersecurity.

https://orcid.org/0000-0003-1431-3078
E-mail: m.karpowicz@elka.pw.edu.pl
Institute of Control and Computation Engineering
Warsaw University of Technology
Nowowiejska 15/19
00-665 Warsaw, Poland