

High Performance Energy-Aware Cloud Computing: A Scope of Future Computing

Shailesh Saxena, Mohammad Zubair Khan, Ravendra Singh



Abstract- This publication discusses high-performance energy-aware cloud (HPEAC) computing state-of-the-art strategies to acknowledgement and categorization of systems and devices, optimization methodologies, and energy / power control techniques in particular. System types involve single machines, clusters, networks, and clouds, while CPUs, GPUs, multiprocessors, and hybrid systems are known to be device types. Objective of Optimization incorporates multiple calculation blends, such as “execution time”, “consumption of energy” & “temperature” with the consideration of limiting power/energy consumption. Control measures usually involve scheduling policies, frequency based policies (DVFS, DFS, DCT), programmatic API’s for limiting the power consumptions (such as “Intel- RAPL,” “NVIDIA- NVML”), standardization of applications, and hybrid techniques. We address energy / power management software and APIs as well as methods and conditions in modern HPEACC systems for forecasting and/or simulating power/energy consumption. Eventually, programming examples are discussed, i.e. programs & tests used in specific works. Based on our study, we point out some areas and there significant issues related to tools & technologies, important for handling energy aware computations in HPEAC computing environment.

Keywords: Energy, Temperature, Control, Policies, Cloud

I. INTRODUCTION

Understanding energy and power plays an increasingly important role in modern high-performance cloud computing systems. In near future, such exascale computing systems will develop which consume near about 20 MW of maximum power. On the basis of top performance-oriented ranking (<https://www.top500.org/lists/top500/>), and the ranking of supercomputers per watt efficiency (<https://www.top500.org/green500/>), Mass acceptance of GPUs has contributed to expand this balance for programs that can operate on such machines efficiently. Parallelization with programming in these hybrid systems must require achieving high efficiency, but there is an issue while it applies with multiple and many core computing. In place of scheduling policies, scaling policies (like DVFS, DFS & DCT) and power reducing APIs have become acceptable for both-CPU & GPU machines, servers as well as mobile lines as power and energy management approaches. Now we have power capping system like Slurm for job management in cluster, which allows idle nodes to be shut down and rebooted when required for reducing the power consumption through DVFS [29].

Revised Manuscript Received on April 30, 2020.

* Correspondence Author

Shailesh Saxena*, Research Scholar, MJP Rohilkhand University, Bareilly, India

Mohammad Zubair Khan, Department of CS, College of Computer Science and Engg., Taibah University, Medina, KSA

Ravendra Singh, Department of CS and IT, MJP Rohilkhand University, Bareilly, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

In different applications-total computation time, utilization of resources, energy consumption, and temperature surrounding the machine are define as key parameters including in different contexts and in different patterns. A constant and detailed review of possibilities, processes & techniques are needed to handle the challenges of HPEAC computing for better results, so we review the literature with the same objective and describe our study in further section of the paper.

II. EXISTING LITERATURE

First of all, in the literature [67], the question of adequate energy and performance metrics was reviewed. Several survey associated with high-performance energy-aware computing are there and based on only the study of related techniques, but the environment and tools are also have the impact on energy aware computation. Which are missing in those surveys? So we also include these missing aspects in our review.

In [79] a study of recent data center and cloud were defined which indicate a wide range of energy-aware factors that are associated with computation in cloud environments. The researchers recommended a nomenclature regarding the optimization of power/energy in computing systems including different levels of abstraction and introducing energy-related works including energy model characterizing features of software and hardware both. So we expand this survey with new approaches, tools and represent a more streamlined review of the power/energy issues of today.

A segregates energy-aware computational approach for different computing environment (i.e. servers, clusters, data centers, grid and clouds) is address in [80]. But there is a lacks of description regarding the optimization parameters, techniques such as power capping and the analysis of API’s. We therefore include in our classification study of these performance metrics for optimization of techniques regarding the energy-aware regulation and measurements.

Work in [68] reviews the strategies based on software & hardware techniques for the energy-aware performance analysis as well as technologies of monitoring the energy of computation in HPEAC systems.

However this paper does not review those strategies which evaluate and control the consumption of power/energy. It’s main objective is to accumulate the available monitoring strategies for power/energy.

However, the review of existing tools in terms of cost, portability and consumer-friendly criteria is included in the paper. Subsequently, we reviewed the controlling policies of power/energy and include it in our review.

Work in [10] comprises an analysis of software technologies from a different point of view to boost energy efficiency in parallel computing; that is, it emphasizes on increasing the efficiency of parallel computing in term of energy. In it authors addresses the components like processor, memory & network in parallel computing and elements like load sharing and balancing algorithms.

A study of strategies for enhancing energy efficiency in distributed systems was addressed in [45] concentrating more on grids and clouds. As we reviewed that it does not include the study of potential optimization policies on the level of node(s) and cluster(s) or energy-conscious simulation of HPEAC systems. We therefore provide a comprehensive list of optimization parameters used by researchers in their works and also classify strategies according to environment and type of resources.

Computational frameworks of power & energy for elevated-performance of computing systems and its implementations are thoroughly explained in [11] with proper references of latest research of others. It discusses node architecture and includes models based on “CPUs”, “GPUs”, “Intel- Xeon Phis” and “FPGAs”. There is an analysis of counter-based systems. We are more oriented on approaches and techniques as well as whole engineering systems that can use these models.

Energy efficient techniques are discussed in [66] for cluster environment, based on software and hardware level issues that might affect energy efficiency, power and resource management and dynamic scaling of voltage and frequency approaches. We also extend this study in our review through including all possible considered strategies.

A review in [30] regarding principles, strategies and architectures for the processing of energy-efficiency in ultra-scale structures was examined with hardware frameworks, power and energy consumption software mechanisms, energy-conscious planning, algorithmic energy characteristics and algorithmic modulation schemes. This review viewed as a supplementary for our research as it includes explanations of energy-conscious algorithms and algorithmic strategies that is not concentrating in our review but the positive side is we have a broader considerations of the metrics and methods define in it for energy efficiency in our review.

Current research on energy efficiency and processing of constrained energy approaches in HPEAC computing systems in form of EXASCALE computing is discussed in [18]. In particular, it takes into account the 20 MW power cap for HPEAC computing systems in future. This work also discusses various monitoring software of power such as “Watts Up? Pro”, vendor devices such as “Intel-RAPL”, “NVIDIA- NVML”, “AMD-APM”, “ IBM- EnergyScale”, and finer grain tools: “PowerPack”, “Penguin PowerInsight”, “PowerMon & PowerMon2”[86] and “HDEEM-High Definition Energy Efficiency Monitoring”. This paper also address a detailed overview of specific approaches like DVFS on primarily level and monitoring tools. So we are extending the classification of energy approaches in our review with respect to types of systems, resources and optimization metrics.

In [46] authors address how to acclimate the tools of throughput measurement in parallel applications for energy efficiency, explicitly the “libadapt library” and “OpenMP wrapper”.

A review is provided in [47] of different techniques for

energy savings with examination of their efficacy in a setting where failures take place. Reliability of energy costs is also considered in the review. A metric of energy-reliability is also suggested here that calculates the required energy to run an application in computing systems.

The research study addressed in [48] provides a structured approach for reviewing the research regarding the energy efficient computing technologies and the end-user software used for managing the components of cloud data centers like computing servers and network devices. The research also highlights current problems in cloud computing systems with the possibilities of future works in the same field. Our study is more concerned with HPEAC computing strategies, so some points will be common in both the reviews.

Ultra-scale power monitoring strategies are address in [2]. The focus in this article is about the description of those approaches used in online power measurement, including a thorough study about their current state of computing performance. It also addresses the detailed description of designated tools & techniques with their usage and the possibilities of research in future. To fulfill our main aim about the study of efficient power/energy management, we also include the review of some simulation tools along with strategies and models used for power/energy management.

III. RESEARCH MOTIVATIONS

As per the study of available current research on the aspects of energy on computation, which affected the global environment as well as the performance of HPEAC computing systems, our research can be seen as a latest and effected review of development in the same area. We include the description about the following aspects in our review -

- Analysis of available energy and power management APIs and Tools in HPEAC computing.
- Different computing environments such as single computing, multiprocessing, cluster, grid, and cloud computing are considered.
- Assessment of different types of resources, including CPU machines, GPU machines and hybrid computing resources.
- Examination about the various optimization metrics of efficiency available in literature including the effect of power/energy and temperature.
- Recognition of different power saving optimization policies like scheduling policies, frequency based policies (DVFS, DFS, DCT) but also the new CPU and GPU power capping functionality, device optimization and hybrid approaches.
- Evaluation of applications used in energy-aware works for measurement and benchmarking.
- HPEAC computing systems tools for predicting and simulating energy and power consumption.
- Reinterpretation of open field research problems based on the latest developments and outcomes.

In this study, we only emphasis on reviewing the methods and techniques for proper configuration, simulation and management of energy-aware processing based HPEAC computing systems. Here we are not address the designing aspect of new software/model/technique of power saving, we are only discussing the available APIs, tools and power saving policies that can be use by different researchers/users to develop/use the HPEAC computing applications. In this paper we also not include the study of hardware modifications or improvements in architecture to improve the power efficiency of HPEAC computing systems.

IV. POWER/ENERGY TOOLS FOR HPEACC SYSTEMS

The power/energy tools can be perceived in two category, first monitoring tools and second controlling tools. Some tools are used only to analysis the power/energy consumption in the HPEACC systems while others may allow to limiting the consumption of power/energy other than analyzing the consumption only. We also have some tools that used not only to limit the usage of power/energy, but also have a feature to alter the type of power/energy usage, such as device frequency. At last, there are many derived tools that wrap the above mentioned low-level drivers in a very smooth manner.

The paper [18] provided a comprehensive survey of existing tools for energy / power management. Here we suggest a distinctly different list selecting the most important tools that can be obtained in 2019 and filling out a few holes that the sample above lacks.

4.1. Power Monitoring Tools- our study says that researchers' already start uses some external meters like devices "Watts Up?Pro" for measuring the consumption of power/energy during computation, when they start focusing on energy efficiency with other performance parameter like execution time, resource utilizations etc. The big advantage of such external devices is that it tracks real power/energy consumption. While these devices cannot able to report the component wise (CPU, GPU and memory etc.) consumption of power/energy.

4.2. Power Controlling Tools- As study said about different implicit techniques and tools that empower us to regulate the usage of power/energy in any computing system. Dynamic voltage and frequency scaling (DVFS) quite often dealt with separately as DVS and DFS has become one strategy that enables us to minimize both voltage and frequency of the computing resources in order to mitigate power/energy usage but subsequently it also degrade the efficiency of computation. DVFS technique works in both (CPUs and GPUs) computing devices. In [60], authors address discrepancies of DVFS usage either in CPU or GPU.

Another technique address in [12] that can lead to power/energy savings is DCT i.e dynamic concurrent throttling and concurrent packing. Reducing the number of resources for any application through reducing the threads define in an application program like OpenMP, we are able to control the power consumption and performance of the application.

4.3. Hybrid Tools- Various tools and hardware are also available for manage the power efficiency in both manners; controlling the power/energy consumption as well as monitoring the usage of power in the systems. In the appendix of [18], authors describe such tools in detail. Some of them are identified as: RAPL (Intel), APM (ADM), NVML(NVIDIA) and Energy Scale. In [7] some additional tools like C-based programming library (NVML) and a command line utility nvidia-smi included in the list. These tools are supported most of the computing systems like Tesla, Quadro, Titan, and GRID lines. The summarize details of such tools along with their usage references are described in Table 1.

Table 1: Hybride Tools of Power Management in HPEACC System

Tool	Vendor	Resource	References	Description
RAPL	Intel	CPU	[21–24]	Used for performance vs maximum power measurements
APM	AMD	CPU	[25]	Developers guide describing the capabilite of the AMD TDP power cap
Energy Scale	IBM	CPU	[26]	Overview on POWER7 power management capabilities
NVML / nvidia-smi	NVIDIA	GPU	[18]	Discussion on differences of using DVFS on CPU

4.4. Extended Tools. We also have some command line interfaces used to evaluate the performance of the computing environments like Performance Application Programming Interface (PAPI) which is still being developed since its release and address first time in paper[88], and has recently been expanded by providing access to the RAPL(Intel) and NVML(NVIDIA) libraries via the PAPI interface[3], in addition to processor quality counters.

In [8] an another open source library tool designed by INTEL named Processor Counter Monitor (PCM) is address which is also a command line utility similar to PAPI interface. It also compiles output counters and allows tracking of power/energy through the interface of RAPL.

In [20] PUPil-"Performance under Power Limits" is address as a hybrid application which include both hardware & software solution to extracting benefits of reducing the consumptions of power/energy. This tries to control both DVFS and core allocation, use of sockets, use of memory and hyper threading. Quite a technique has been contrasted to raw RAPL power capping by authors, and PUPil is in pursuit of the outcomes achieved.

Score-P [21], supposed for the assessment and successive optimization of HPEAC computing applications, which also allow for the assessment of energy-aware strategies. It shows how clock frequency on the SuperMUC network, affects execution time of finite-element during operations with minimum consumption of power/energy. As result says, energy-optimal settings save 2% energy but increase the execution time by 14%, while time-optimal settings save 14% execution time on the value of 6% increment in the consumption of power/energy.

The power limiting strategies has become available in form of a cap-set [9] as a user-friendly command-line utility, since the release of Ubuntu 18.04 LTS. It is also based on RAPL(Intel), therefore it is said that its usefulness is only for Intel processors. This permits a power cap to be set for each of the required domains likes - PP0, PP1, PKG and DRAM.

V. CLASSIFICATION OF HPEAC COMPUTING OPTIMIZATIONS

In this paper we categorize existing literature in a number of aspects, including the following key parameters:

ComputingEnvironment- What type of computing environment like cluster, grid & cloud with how many computing devices like single or multiple have an impact on optimization of HPEACC systems?

Resources- Which types of computers or computing resources, especially processing units like-CPU(s), GPU(s) or hybrid (CPU+Accelerator) computing resource(s), are also considered for optimization. There are several works in the literature that represent all identified computing environments and resources (Table 2).

PerformanceMetric(s)-some parameters like energy consumption, power limitation, execution time and temperature, called performance metric(s) are also being optimized for better results (Table 3). During the review of literature we found that many research papers try to reducing the consumption of power/energy during computation but on the other hand they compromise with rest of the performance impact. It can be accomplished by applying power/energy reduction techniques on application phases. There are very few studies concerned with network and memory elements. Automatic profiling and concurrent implementations of hybrid computing systems have been missing.

Methods of Power/Energy Controlling- How the utilization of computing resources are optimized through selection of resource, scheduling of resources, controlling CPU with low level frequency, CPUs / GPUs Power limiting at application level or modification on application level (Table 4).

The literature defines the frequency, computing core [13], memory disk and network links [53] as regards system components that can be regulated in terms of power and energy. Finally, Energy-Efficient-Ethernet that is refers as EEE [57], can also convert the devices works on physical layer into the low mode devices, save the power up-to even 70%. It also describe that in many realistic situations the technological overhead is negligible. The MREEF method considered under [38] points out measures to automate the identification of system stages, process definition, classification, system estimation and the reconfiguration to minimize energy consumption.

Table 2: Types of Computing Environment & Resources

Reference	Description	Computing Environment	Computing Resource
[33]	A platform based on ARM Cortex A9, 4, 8, and 16 core architectures	Single Computing	Multicore CPU
[34]	Scheduling kernels on a GPU and frequency scaling	Single Computing	GPU/Accelerator
[35]	A chip with k cores with specific frequencies is considered, and chips with 36 cores are simulated	Single Computing	Multicore CPU
[36]	Finding best application configuration and settings on a GPU	Single Computing	GPU/Accelerator
[37]	Server-type NVIDIA Tesla K20 m/K20c GPUs	Single Computing	GPU/Accelerator
[38]	Exploration of thermal-aware scheduling for tasks to minimize peak temperature in a multicore system through selection of core speeds	Single Computing	Multicore CPU
[39]	Comparison of energy/performance trade-offs for various GPUs	Single Computing	GPU/Accelerator
[40]	Server multicore and manycore CPUs, desktop CPU, mobile CPU	Single Computing	Multicore/Manycore CPU
[41]	Single CPU under Linux kernel 2.6–11/ Single-core Pentium-M (32-bit) in a off-the-shelf laptop	Single Computing	Single CPU
[42]	Intel Xeon Phi KNL 7250 processor with 68 cores, flat memory mode	Single Computing	Manycore CPU
[43]	Exploration of execution time and energy on a multicore Intel Xeon CPU	Single Computing	Multicore CPU
[44]	Task scheduling with thermal consideration for a heterogeneous real-time multi-processor system-onchip (MPSoC) system	Multiprocessing	Multiple CPU
[30]	Presents a hybrid approach PUPIL (Performance under Power Limits)—a hybrid software/hardware power capping system based on a decision framework going through nodes and making decisions on configuration, considered for single and multiapplication scenarios (cooperative and oblivious applications)	Multiprocessing	Multiple CPUs
[45]	With notes specific to clusters	Multiprocessing	Multiple CPUs
[14]	Systems with 2 socket Westmere-EP, 2 socket Sandy Bridge-EP, and 1 socket Ivy Bridge-HE CPUs	Multiprocessing	Multiple CPUs
[46]	Dual-socket server with two Intel Xeon CPUs Scientific Programming 5	Multiprocessing	Multiple CPUs
[47]	Proposes integration of power limitation into a job scheduler and implementation in SLURM	Cluster Computing	Multiple CPUs
[48]	Proposes the enhanced power adaptive scheduling (E-PAS) algorithm with integration of power-aware approach into SLURM for limiting power consumption	Cluster Computing	Multiple CPUs
[49]	Multicore CPUs as part of a node and cluster on which an MPI application runs but focusing on states of processes running on CPUs, i.e., reducing power consumption of CPUs on which processes are idle or perform I/O operations	Cluster Computing	Multicore CPUs
[50]	Proposes DVFS-aware profiling that uses design time profiling and nonprofiling approach that performs computations at runtime	Cluster Computing	Multiple CPUs
[51]	Split compilation is used with offline and online phases, results from the offline-phase passed to runtime optimization, grey box approach to autotuning, and assumes code annotations on a cluster with Intel Xeon CPUs and MICs	Cluster Computing	Hybrid

Table 2: Types of Computing Environment & Resources (Continued...)

Reference	Description	Computing Environment	Computing Resource
[52]	Proposes a runtime library that performs poweraware optimization at runtime and searches for good configurations with DFS/DCTfor application regions Sandy Bridge and Haswell Xeon CPUs	Cluster Computing	Multicore CPU
[53]	Approaches for modeling, monitoring, and tracking HPC systems using performance counters and optimization of energy used in a cluster environment with consideration of CPU, memory, disk, and network	Cluster Computing	Multiple CPUs
[54]	Proposed an energy-saving framework for a cluster with ranking and correlating counters important for improving energy efficiency which related to runtime, system, CPU, and memory power	Cluster Computing	Single/ Multicore CPU
[55, 56]	Energy savings on a cluster with Sandy Bridge processors	Cluster Computing	Multiple CPUs
[57]	With consideration of disk and network scaling	Cluster Computing	Multiple CPUs
[58]	Possibly 2 multiprocessor systems, including disk, memory, processor, or even fans	Cluster Computing	Multiple CPUs
[24]	Analysis of performance vs power of a 32-node cluster running a NAS parallel benchmark	Cluster Computing	Multiple CPUs
[59]	A procedure for a single device (a compute node with	Cluster Computing	Single / Multi
[60]	Homogeneous multicore cluster	Cluster Computing	Multiple CPUs
[61]	Cluster with CPUs	Cluster Computing	Multiple CPUs
[62]	Computer system with several nodes each with	Cluster Computing	Multiple CPUs
[63, 64]	Cluster with several nodes each with multicore CPUs	Cluster Computing	Multicore CPUs
[65]	Cluster with several nodes with CPUs	Cluster Computing	Multiple CPUs
[66, 67]	Cluster in a data center	Cluster Computing	Multiple CPUs
[68]	Sandy Bridge cluster	Cluster Computing	Multiple CPUs
[69]	Cluster with InfiniBand	Cluster Computing	Multiple CPUs
[70]	Overprovisioned cluster which can run a certain	Cluster Computing	Multiple CPUs
[71]	Cluster with 1056 Dell PowerEdge SC1425 nodes	Cluster Computing	Multiple CPUs
[72]	A cluster with 9421 servers connected by InfiniBand	Cluster Computing	GPU/Accelerator
[73]	A cluster or collection of clusters allowed in the model and implementation	Grid Computing	Hybrid
[74]	Implementations of hierarchical genetic strategybased grid scheduler and algorithms evaluated against genetic algorithm variants	Grid Computing	Multiple CPUs
[75]	Meant for cloud storage systems/ GPUs used for generation of parity data in a RAID	Cloud Computing	GPU/Accelerator
[76]	Related to assignment of applications to virtual and	Cloud Computing	Multiple CPUs
[77]	Used as IaaS for computations	Cloud Computing	Multiple CPUs

Table 3: Performance Metric

Reference	Description	Performance Metric			
		Execution Time	Power Limit	Energy Consumption	Temperature
[73]	Minimization of application running time with an upper bound on the total power consumption of compute devices selected for computations	✓	✓		
[47]	Shows benefit of power monitoring for a resource manager and compares results for fixed frequency mode, minimum power level assigned to a job, and automatic mode with consideration of available power.	✓	✓		
[30]	Performance under power cap, timeliness, and efficiency/weighted speedups are considered	✓	✓		
[48]	Execution time vs maximum power consumption per system considered, consideration of system utilization, power consumption profiles, and cumulative distribution function of the job waiting time.	✓	✓		
[52]	Application slowdown vs power reduction and optimization of performance per Watt	✓	✓		
[24]	Analysis of performance vs power limit configurations	✓	✓		
[42]	Analysis of performance vs power limit configurations	✓	✓		
[62]	Analysis of performance/execution time vs power limit configurations	✓	✓		
[68]	Turnaround time vs cluster power limits	✓	✓		
[46]	Consideration of impact of power allocation for CPU and DRAM domains on performance when power capping	✓	✓		
[70]	Optimization of the number of nodes and power distribution between CPU and memory in an overprovisioned HPC cluster	✓	✓		
[33]	Task partitioning and scheduling, heuristic algorithm task partitioning, and scheduling TPS based on task partitioning compared to Min-min and PDTM	✓		✓	✓
[38]	Finding such core speeds that tasks complete before deadlines, and peak temperature is minimized	✓		✓	✓
[67]	Thermal aware task scheduling algorithms are proposed for reduction of temperature and power consumption in a data center, and job response times are considered	✓		✓	✓
[44]	Minimization of energy consumption of the system with consideration of task deadlines and temperature limit	✓		✓	✓
[66]	Workload placement with thermal consideration and analysis of cooling costs vs data center utilization	✓		✓	✓
[34]	Concurrent kernel scheduling on a GPU+ impact of frequency scaling on performance and energy consumption	✓	✓	✓	
[45]	Dynamic core and uncore tuning to achieve the best	✓	✓	✓	
[57]	Trade-off between performance (measured by execution time) and energy consumption (with consideration of disk and network scaling)	✓	✓	✓	
[14]	Trade-off between performance and energy consumption	✓	✓	✓	

Table 3: Performance Metric (Continued...)

Reference	Description	Execution Time	Power Limit	Energy Consumption	Temperature
[74]	Biobjective optimization task with make-span and average energy consumption	✓	✓	✓	
[50]	Joint optimization of value (utility) and energy, consideration of jobs with dependent tasks, profiling, and nonprofiling-based approaches	✓	✓	✓	
[51]	Performance and energy efficiency, focus on application autotuning, and framework	✓	✓	✓	
[58]	Keeping performance close to initial and make energy saving	✓	✓	✓	
[53]	Performance vs energy consumption, trade-off, impact of detection, and recognition thresholds on energy consumption and execution time	✓	✓	✓	
[75]	Maximization of performance and minimization of energy consumption at the same time shown for the proposed GPU-RAID compared to a regular Linux-RAID	✓	✓	✓	
[39]	Exploration of trade-offs between performance and energy consumption for various GPUs	✓	✓	✓	
[76]	Simultaneous minimization of energy and execution time	✓	✓	✓	
[77]	Minimization of energy consumption (KWh) while maintaining defined QoS (percentage of SLA violation)	✓	✓	✓	
[59]	Minimization of energy consumption (J) while keeping a minimal performance influence	✓	✓	✓	
[42]	Analysis of execution time vs energy usage for various power limit configurations, using DDR4 or MCDRAM memories	✓	✓	✓	
[60]	Pareto optimal solutions incorporating performance and energy taking into account functions such as speed of execution vs workload size and dynamic energy vs workload size, and optimal number of processors is selected as well	✓	✓	✓	
[63, 64]	Consideration of impact of DCT and combined DVFS/DCT on execution time and energy usage of hybrid MPI/OpenMP applications and controls execution of OpenMP phases with the number of threads and DVFS level based on prediction of phase execution time with event rates	✓	✓	✓	
[65]	Minimization of energy delay product of MPI applications in a transparent way through reduction of CPU performance during MPI communication phases	✓	✓	✓	
[43]	Exploration of execution time and energy of parallel OpenMP programs on a multicore Intel Xeon CPU through various strategies involving various loop scheduling ways, chunk sizes, optimization levels, and thread counts	✓	✓	✓	
[69]	Minimization of energy used at the cost of minimal performance loss and proposes energy-aware MPI (EAM) which is an application oblivious MPI runtime that observes MPI slack to maximize energy efficiency using power levers	✓	✓	✓	
[72]	Providing a default configuration for an acceptable power-performance trade-off, with additional policies implied for a specific computing	✓	✓	✓	
[49]	Energy minimization with no impact on performance			✓	
[35]	Energy minimization at the cost of increased of execution time, integer linear programming-based approach in order to find a configuration with the number of cores minimizing energy consumption			✓	

Table 3: Performance Metric (Continued...)

Reference	Description	Performance Metric			
		Execution Time	Power Limit	Energy Consumption	Temperature
[55, 56]	Energy minimization at the cost of increased execution time, achieving energy savings while running a parallel application on a cluster through DVFS and frequency minimization during periods of lower activity: intranode optimization related to inefficiencies of communication, intranode optimization related to nonoptimal data, and computation distribution among processes of an application			✓	
[54]	A what-if prediction approach to predict energy savings of possible optimizations, and the work focuses on identification of a set of performance counters for a power and performance model			✓	
[36]	Finding an optimal GPU configuration (in terms of the number of threads per block and the number of blocks)			✓	
[37]	Minimization of energy after an application has finished through frequency control			✓	
[40]	Energy minimization at the cost of increased execution time through power capping for parallel applications on modern multi and manycore processors			✓	
[41]	Energy minimization with low-performance degradation (aiming up to 5%)			✓	
[61]	Energy minimization of MPI programs through frequency scaling with constraints on execution time, linear programming approach is used, and traces from MPI application execution are collected			✓	
[36]	Finding an optimal GPU configuration (in terms of the number of threads per block and the number of blocks)	✓		✓	

Table 4: Energy/Power Controlling Methods.

Reference	Description	Energy/Power control method				
		Selection of devices	Scheduling	DVFS/DFS /DCT	Power Capping	Application Optimizations
[73]	Selection of devices in a cluster or collection of clusters such that maximum power consumption limit is followed + data partitioning and scheduling of computations	✓	✓			
[35]	Selection of cores for a configuration minimizing energy consumption	✓				
[75]	Using GPUs for optimization/generation of parity data	✓				
[39]	Selection of best GPU architectures in terms of performance /energy usage point of view	✓				
[58]	Specific scheduling and switching off unused cluster nodes		✓			
[33, 71]	Task partitioning and scheduling		✓			
[44]	A two-stage energy-efficient temperature-aware task scheduling algorithm is proposed : in the first stage, dynamic energy consumption under task deadlines and in the second stage temperature profiles of processors are improved		✓			

Table 4: Energy/Power Controlling Methods. (Continued...)

Reference	Description	Energy/Power control method				
		Selection of devices	Scheduling	DVFS/DFS/DCT	Power Capping	Application Optimizations
[76]	Application assignment to virtual and physical nodes of the cloud		√			
[66]	Workload placement in a data center		√			
[68]	Proposal of RMAP—a resource manager that minimizes average turnaround time for jobs provides an adaptive policy that supports overprovisioning and power-aware backfilling		√			
[49]	For MPI applications with the goal not to impact performance			√		
[47]	Uniform frequency power-limiting investigates results for the fixed frequency mode, minimum power level assigned to a job, and automatic mode with consideration of available power			√		
[45]	Core and uncore frequency scaling of CPUs			√		
[55, 56]	Minimization of energy usage through DVFS on particular nodes			√		
[52]	DFS, DCT			√		
[14]	DVFS, DCT			√		
[37]	Control of frequency on a GPU			√		
[41]	DVFS with dynamic detection of computation phases (memory and CPU bound)			√		
[59]	DVFS with a posteriori (using logs) detection and prioritization of computation phases (memory and CPU bound)			√		
[61]	Sysfs interface is used			√		
[63, 64]	DCT, combined DVFS/DCT			√		
[65]	Sysfs interface			√		
[72]	Setting the frequency according to the established computing center policies			√		
[24]	Using Intel RAPL for power management				√	
[40]	Using Intel RAPL for analyzing energy/performance trade-offs with power capping for parallel applications on modern multi- and manycore processors				√	
[42]	Using PAPI and Intel RAPL				√	
[62]	Using Intel RAPL				√	
[46]	Using Intel's power governor tool and Intel RAPL				√	
[54]	theoretical consideration of optimizations of an application that results in improvement of performance countervalues					√
[36]	Finding an optimal GPU configuration (in terms of the number of threads per block and the number of blocks)					√
[53, 57]	Control of CPU frequency, spinning down the disk, and network speed scaling					√
[43]	Exploration of various loop scheduling ways, chunk sizes, optimization levels, and thread counts					√
[30]	Software + RAPL, the proposed PUPIL approach combines hardware's fast reaction time with flexibility of a software approach					
[48]	Scheduling/software + resource management (including RAPL), the proposed algorithm takes into account real power and energy consumption	√	√			

Table 4: Energy/Power Controlling Methods. (Continued...)

Reference	Description	Energy/Power control method				
		Selection of devices	Scheduling	DVFS/DVS /DCT	Power Capping	Application Optimizations
[34]	Concurrent kernel execution + DVFS	✓		✓		
[50, 74]	Scheduling + DVFS		✓	✓		
[38]	Scheduling + DVFS for minimization of temperature and meeting task deadlines		✓	✓		
[51]	Scheduling jobs and management of resources and DVFS	✓	✓	✓		
[77]	Selection of the resources for a given user request, with VM migration and putting unused machines in the sleep mode	✓	✓	✓		
[60]	Workload distribution + DVFS-based multiobjective optimization		✓	✓		✓
[69]	Polling, interrupt-driven execution (relinquishing CPU and waiting on a network event), DVFS power levers		✓	✓		✓
[70]	Selection of nodes in an overprovisioned HPC clusters and Intel RAPL		✓			

In Table 5, we take three of the main factors described in Tables 2-4 and correlate them to illustrate current work in the form of measurements, methods of managing power/energy and sort of appliance. This correlating of factors is providing a strong basis of description the recent trends in research into HPEAC computation as well as to identify the transparent fields of further study.

Although most of the literature discussed focuses on efficiency and power or energy management during program execution, pre or post execution situations can also be called. In fact, after the application on a GPU has been ended the study in [35] considers the post-execution scenarios. In a case like this, energy consumption can be minimized relative to the normal scenario by means of imposed frequency controls. The tables take information into consideration.

1. Prediction Based Simulation of Power/Energy Consumption in HPEAC Systems -

Several Tools are available for predict the future need of power/energy in HPEAC computing systems and simulate it for increase the performance of computing. The comparative summary of such types of tools is present in Table 6.

Grid-S-Sim[81] is a simulator specially design to simulate the performance of scheduling policies in grid computing environment with the proper utilization of computing resources and their energy.

The DCworms [65] extension provides additional plugins in a built data center for temperature and usage of power/energy. This simulator is able to work on three types of approaches for energy saving modeling: (i) static approach with various power-level modes, (ii) dynamic approach in which energy consumption depends on the load of resources at run time, and (iii) application specific

approaches which can be used for advanced model tuning. On the comparison of actual hardware implementation with the simulations results of these approaches, we found a very high correlation in actual HPEAC computing system and its simulation for both power and thermal models[43].

Modeling Efficiency, Reliability and Power consumption of multilevel parallel HPC Systems using CPUs and GPUs (MERPSYS[17]) enables hierarchic modeling and testing of a given program for the grid, cluster or single machinery architecture. The platform includes the flexible system and task specifications for simulating power consumption and running time (Java scripts defined by using the web simulator interface). Typical SPMD and DAC (Divide and Conquer) software [26] were used to test this simulator.

CloudSim [82] is a platform devoted to simulating behavior, following an IaaS architecture, of a cloud or a server federation. The platform helps you to model all core cloud elements, including physical devices, VM distribution, cloud sector, network properties and complex workflows. The findings of the simulation help the data centre's in - allocation of services, QoS and measurement of energy consumption. CloudSim is used by scientific and industry analysts, such as HP Laboratories in the United States.

SimGrid[58] is a multi-faceted, multi-faceted and modular event modeling system for grid environments. Two styles of APIs, such as MPIs from application of real-world and directed acyclic graph (DAG) formats of work flows are provided in this method. The SimGrid extension [17] requires energy consumption for parallel HPC network implementations with multimedia device DVFS technology.

High Performance Energy-Aware Cloud Computing: A Scope of Future Computing

GENSim [78] simulator is used to simulate the data centers under the modeling of mixed task feed, both for web based activities and batch activities.

The method was used for energy consumption calculation with both brown and green energy being used to speed up the current computations of batch processing under that period of time when renewable energy predicted at peak. It is also used to calculate power consumption. The test results were tested with a real hardware test bed made up of a set of cloud servers centered on the CPU (Intel Nehalem)[76].

For HPEAC computing model, combinations of OMNet++ tools with INET [72] were used, in which energy algorithms were tested in the programming process. This took into consideration the individual cluster setup and simulated the clients request about 400

jobs. The conduct of the key server modules, including the departure of inactive nodes, was evaluated. The outcomes of the simulation were contrasted with the results found in

real environment.

A comprehensive system to test data center power consumption is given by GDCSim[83] (Green data center Simulator). This method allows for study of geometries of data centres, task features, network resources forecasting and optimization algorithms. This facilitates both the thermal analysis of the resource management system in different physical conditions (use of CFD) and the energy efficiency analyzes. The simulator has been used to estimate the scheduling and the workload in the Internet data center in a HPC context.

A cloud simulator, which offers a energy consumption model for different architectures of a data centre, has been provided by GreenCloud[77]. The architecture includes basic elements of the networks for workload: storage, connectivity, convergence and core network equipment including switches (like L2 & L3) and routers that run at different speed of networks such as an Ethernet of 1, 10 and 100 Giga byte.

Table 5: Correlation of Performance Metrics with Computing Devices & Power/ Energy Controlling Methods

Performance Metric	Device	Power/Energy Controlling Method				
		Selection of devices/scheduling	DVFS/DFS/DCT	Power capping	Application optimization	Hybrid
Performance/execution time with power limit	1x CPU		[52]	[24, 42]		
	Nx CPU	[68]	[47]	[46, 62]		[30, 48, 70]
	GPU					
	Hybrid	[73]				
Performance/execution time/energy minimization + thermal aware	1x CPU	[33]			[38]	
	Nx CPU	[44, 66, 67]				
	GPU					
	Hybrid					
Performance/execution time/value + energy optimization	1x CPU	[58]	[59]	[42]	[43]	[77]
	Nx CPU	[76]	[14, 45, 63–65, 72]		[53, 57]	[60, 69, 74]
	GPU	[39, 75]				[34]
	Hybrid					[51]
Energy minimization	1x CPU	[35]	[41, 49]	[40]	[54]	
	Nx CPU		[55, 56, 61]			
	GPU		[37]		[36]	
	Hybrid					
Product of energy and execution time	1x CPU					
	Nx CPU					
	GPU				[36]	
	Hybrid					

Table 6: Prediction or Evaluation Simulators for Power/Energy Consumption in HPEAC system.

Simulators	Computing Environment	Reference	Description
GSSim/DCworms	Grid	[80, 81]	A scheduler simulation concerning performance and energy consumption for complex grid architectures
MERPSYS	Grid/cluster	[82, 83]	Used to simulate the energy consumption of a cluster compute nodes
CloudSim	Cloud	[84]	Used for simulation of VM provisioning in a cloud environment
SimGrid	Grid	[85]	Focused on its versatility and scalability
GENSim	Cloud	[86]	Used to simulate green energy prediction
OMNet++, INET	Cluster	[58]	Used for simulation of switching off the unused nodes in a cluster
GDCSim	Data center	[87]	Used for holistic evaluation of HPC and Internet data centers
GreenCloud	Data center	[88]	Used for evaluation of cloud data centers with various infrastructure architectures
TracSim	Cluster	[89]	Used for maximizing the performance for a given power cap
ASKALON	Cloud	[90]	Used for cloud simulation with a given power cap
Energy-aware HyperSim-G	Grid	[74]	Used for assessment of energy-aware scheduling algorithms
GPU design space exploration	GP GPU	[39]	Dedicated for multiobjective GP GPU evaluation and selection
Sniper + McPAT	CPU	[35]	Used for multicore CPU energy-aware simulation

The simulator implements DVFS and DNS in simulation of power management with the numerous workload characteristics that are built into the specified model of data center. It also presents different examples that estimate the consumption of energy under 2-layer and 3-layer DC systems, including a high-speed (100 GB) Ethernet version of interconnection.

TracSim[27] is an high performance cluster simulator with a fixed capacity of power/energy usage that can't be increased because of limitation about cooling as well as the connections of electrical signals. Another theory is that some organizations need less electricity. Some can use more space for energy consumption. In order to simulate different approaches about determine the accurate level of power consumption, it applies various programming policies. Experiments have shown that this approach can give best results in a particular environment, i.e. high performance cluster computing at LANL (Los Alamos National Laboratory), and in most instances the average simulation findings are correct at 90 percent.

Authors of [28] proposed a hybrid combination of 3 methods that provide the entire device with a special power limit on the InfraS (IaaS) platform for an integrated event-based cloud Simulation. The elements of the simulator are, (i) ASKALON [99] used to describe the workflow, (ii) DISSECT-CF[84] describe the functionalities of cloud

environment, and (iii) Ground-Sim [44], simulate the experiment according to proposed models/ algorithms. For the performance evaluation, it focused on the workflows simulation with the implementation of real trace. Given the complex solution it was efficient and scalable.

In order to test genetic-based planning algorithms used within a grid setting, the energizer HyperSim-G simulator [74] was used. A simple HyperSim-G simulation package defined in [93] is used as a base for this method. It utilized the DVFS technique in simulation of power management policies and conducted different experiments to show that grid schedulers are systematically evaluated the minimization of energy and performance.

Project GP GPU space exploration is suggested in [50], which offers a method to test GP GPU systems for multi-objective uses specifically in the field of medical computations or processing of large industry based applications. The analysis is carried out the different evaluation parameters, such as Performance, Energy Consumption and Evaluation Capacity of modeled devices at real time. The main objective of this simulator (follows a distributed framework) is to supports a range of GPUs, in the setup of heterogeneous cloud elements. A realistic

streaming test was taken to verify the approach and a small error level was found (in contrast to actual systems below 4 percent) in the worst case.

Langer et al. introduced in [67] a study on minimizing energy consumption in the multi-core chips for two specific HPEAC benchmarks. They described an optimal configuration based on integer linear programming that also solved different heuristics. The simulation follows the Sniper program [85] to improve the precision of the simulations to increase efficiency. The McPAT Framework [91] has been strengthened to the simulators to provide energy-sensitive space for multicore computing. It also able to simulate dynamic changes, short-circuit and leak power modeling.

VI. SCOPE OF RESEARCH UNDER HPEAC -

At last, based on our analytical study of research in high efficiency computing with energy consciousness, we can point out the following research fields that seem crucial for further progress in the same area:

- (1) Table 1 presents the range of HPEAC tools used for power/energy management, shows the need to API's unification with vendors or supplier, so that we easily choose the effective power-aware API suitable for HPEAC computing systems (multi-core machines like CPUs/GPU/Accelerator) to support the maximum parameters related to power/energy saving for increase the computing performance.
- (2) Table 6 presents the precision and usability of the simulators used for prediction based simulation according to the specific computing environments and resources described in Table 2. and used metrics, show that further development of, possibly empirical, performance-energy models for a wide range of CPU and GPU architectures for various types of applications is required including performance (power limit) functions, availability for runtime usage as well as simulator environments.
- (3) As per Table 5, we can identified several directions of research in the field of HPEAC computing in which further development is still required:
 - (i) Power/Energy saving techniques for hybrid computing systems (i.e. CPU+Accelerator).
 - (ii) Optimization of power/energy saving strategies with the target of minimizing the consumption of energy as well as execution time
 - (iii) About the hybrid energy strategies, used for minimizing both energy consumption as well as energy-time.
- (4) Finally, the analysis shown in Table 4 about the strategies of energy controlling, gives us the following conclusions:
 - (i) It is required to develop any tool which has the ability of automatic configuration of HPEAC computing system for achieving the best computational performance with less energy consumption. While there are some strategies for some specific type of applications but there is a lack of such general tools, do adjustment in the variation of APIs. These tools can also be used the proposed model(s) to implement the "performance-energy" profile of the computation.

- (ii) Self/Automatic tuning of the HPEAC computing device for hybrid (CPU + accelerator) resources and efficiency at runtime, in which the excess of computations can be calculated by power / energy limits, not only at the time of the computation.
- (iii) During the validation of existing strategies of power/energy management, it is also essential to achieve the technical up-gradations and quality improvements. For example- validation of (ADM) TDP Power Cap tool or the (IBM) Energy Scale functionality.

VII. CONCLUSION AND FUTURE SCOPE-

In the article, we study about the APIs/Tools/Simulators used for the energy controlling and management in HPEAC computing systems, which include state of art about CPUs and GPUs, and introduced power/energy consumption modeling methods and/or simulation throughout HPC systems. We explored techniques, control methods, standards for optimization, and sample programming, as well as metrics for state of the art energy efficient computing work. In specific, approaches have been proposed for structures like workstations, servers, networks and clouds utilizing different computer hardware such as multi- and multi-core CPU and GPU. Evaluation included output duration, energy consumption, and temperature variations used. The possible solution covers scheduling and DVFS / DFS / DCT control methods, power capping and task optimization as well as alternative strategies. Finally, open areas for future research in this field have been presented and recommendations.

REFERENCES

1. Jagode, H., Vaccaro, P., Haidar, A., YarKhan, A., Tomov, S., & Dongarra, J. 2019, "Investigating Power Capping Toward Energy Efficient Scientific Applications", in *Concurrency and Computation: Practice and Experience*, vol. 31-6, e4485.
2. Assuncao, D., M., Almeida, F., & Barbosa, J., 2018, "EnergyMonitoring as an Essential Building Block towards Sustainable Ultra -scale Systems", in *Sustainable Computing: Informatics and Systems*, vol. 17, pp. 27-42.
3. Niemi, T., Khan, N., K., Hirki, M., Nurminen, K., J., & Ou, Z., 2018 "RAPL In Action: Power Measurement's Experiences in UsingRAPL", in *ACM Transactions on Modeling and PerformanceEvaluation of Computing Systems*, Vol. 3, no. 2, PP. 1-26.
4. Proficz, J., Krzywaniak, A., & Czarnul, P., 2018 "Analyzing Energy/Performance Trade-offs with Power Capping for Parallel Applications on Modern Multi and Many Core Processors", in *Proceedings of the 2018, FedCSIS*, PP. 339-346, Poland.
5. Miwa, S., Inoue, K., Miyoshi, I., & Kondo, M., 2018, "Run-timeDFS/DCT Optimization for Power-Constrained HPC Systems", in *Proceedings of the HPC Asia 2018*, Chiyoda, Japan.
6. Manumachu, R., R., & Lastovetsky, A., 2018, "Bi-Objective Optimization of Data-Parallel Applications on HomogeneousMulti-Core Clusters for Performance and Energy", in *IEEE Transactions on Computers*, vol. 67- 2, PP. 160-177.
7. NVML API reference, 2018, [https://docs.nvidia.com/deploy/nvml-api/nvml-api-reference.html.]
8. Intel PCM (processor counter monitor), 2018, [https://github.com/opcm/pcm].
9. SPEC: Standard Performance Evaluation Corporation, 2018, [http://www.spec.org/.]

10. Jin, C., Abramson, D., & de Supinski, R., B., 2017, "A Survey on Software Methods to Improve the Energy Efficiency of Parallel Computing", in e-International Journal of High Performance Computing Applications, Vol. 31-6, PP. 517–549.
11. Reddy, R., Lastovetsky, A., O'Brien, K., Pietri, I. & Sakellariou, R., 2017, "A Survey of Power & Energy Predictive Models in HPC Systems and Applications", ACM Computing Surveys, Vol. 50, no. 3, PP. 1–38.
12. Sensi, D., Kilpatrick, P., & Torquati, M., 2017, "State-Aware Concurrency Throttling", in Proceedings of the International Conference on Parallel Computing, ParCo 2017, PP. 201–210.
13. Sourouri, M., Raknes, B., E., & Reissmann, N., 2017, "Towards Fine-Grained Dynamic Tuning of HPC Applications on Modern Multicore Architectures", in Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, SC'17, PP. 41:1–41:12, ACM, Denver, USA.
14. Tafani, D., Georgiou, Y., Rajagopal, D., Glesser, D., & Ott, M., 2017, "A Novel Approach for Job Scheduling Optimizations Under Power Cap for ARM and INTEL HPC Systems", in Proceedings of 24th IEEE International Conference on High Performance Computing PP. 142–151, Portland, USA.
15. Tian, C., Y., Vasudevan, M., Tang, M., & Kozan, K., 2017, "Profile-Based Application Assignment for Greener and more Energy Efficient Data Centers", in International Journal Future Generation Computer Systems, vol. 67, PP. 94–108.
16. Czarnul, P., Kuchta, J., & Matuszek, M., 2017, "MERPSYS: An Environment for Simulation of Parallel Application Execution on Large Scale HPC Systems", in Simulation Modeling, Practice and Theory, vol. 77, pp. 124–140.
17. Heinrich, C., F., Carpen-Amarie, A., & Degomme, A., 2017, "Predicting Performance and Power Consumption of MPI Applications with SimGrid",.
18. Labasan, S., 2016, "Energy-Efficient & Power-Constrained Techniques for Exa-Scale Computing",.
19. Paradis, C., Desrochers, S., & Weaver, M., V., 2016, "A Validation of DARM-RAPL Power Measurements", in Proceedings of 2nd International Symposium on Memory Systems (MEMSYS'16), PP. 455–470.
20. Zhang, H., & Hoffmann, H., 2016, "Maximizing Performance Under a Power Cap: A Comparison of Hardware, Software and Hybrid Techniques", in ACM SIGPLAN Notices, Vol. 51, no. 4, PP. 545–559.
21. Diethelm, K., 2016, "Tools for Assessing and Optimizing the Energy Requirements of High Performance Scientific Computing Software", in PAMM, Vol. 16, no. 1, PP. 837–838.
22. Chen, M., Zhou, J., Wei, T., Hu, S., X., Ma, Y., & Yan, J., 2016, "Thermal Aware Task Scheduling for Energy Minimization in Heterogeneous Real-Time MPSOC Systems", in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 35, no. 8, PP. 1269–1282.
23. Agosta, G., Silvano, C., & Cherubin, S., 2016, "The Antares Approach to Auto-tuning and Adaptively for Energy Efficient HPC Systems", in Proceedings of ACM International Conference on Computing Frontiers, CF'16, pp. 288–293.
24. Taylor, V., Wu, X., Cook, J. & Mucci, P., J., 2016, "Using Performance-Power Modeling to Improve Energy Efficiency of HPC Applications", in MDPI Compute Journal, Vol. 49, no. 10, PP. 20–29.
25. Peraza, J., Laurenzano, M., Tiwari, A., Snively, A., & Carrington, L., 2016, "PMAC's Green Queue: A Framework for Selecting Energy Optimal DVFS Configurations in Large Scale MPI Applications", in Concurrency and Computation: Practice and Experience, Vol. 28, no. 2, PP. 211–231.
26. P. Czarnul, P. Rościszewski, J. Proficz and J. Kuchta, "Modeling of Energy Consumption for Parallel Applications", in Proceedings of Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 855–864, Gdansk, Poland, September 2016.
27. Z. Zhang, S. Pakin, S. Fu and M. Lang, "TRACSIM: Simulation and Scheduling Trapped Power Capacity to Maximize Machine Room Throughput", in Parallel Computing, vol. 57, pp. 108–124, 2016.
28. S. Ostermann, R. Prodan and G. Kecskemeti, "Multi-layered Simulations at the Heart of Workflow Enactment on Clouds", in Concurrency and Computation: Practice and Experience, vol. 28, no. 11, pp. 3180–3201, 2016.
29. Y. Georgiou, D. Trystram, M. Hautreux, and D. Glesser, "Power Adaptive Scheduling", in Proceedings of SLURM User Group, Edinburgh, UK, September 2015.
30. J. Carretero, D. Petcu & S. Distefano et al., "Energy-Efficient Algorithms for Ultra Scale Systems," in Supercomputing Frontiers and Innovations, vol. 2, no. 2, pp. 77–104, 2015.
31. D. Hackenberg, T. Ilse, R. Schöne, J. Schuchart, R. Geyer and D. Molka, "An Energy Efficiency Feature Survey of the Intel HASWELL Processor", in Proceedings of IEEE International Parallel and Distributed Processing Symposium Workshop, pp. 896–904, Orlando, FL, USA, May 2015.
32. AMD: Bios and kernel developer's guide (BKDG) for AMD family processors (15h models 00h-0fh), 2015.
33. Z. Wang, P. Mishra and S. Ranka, "Efficient Task Partitioning and Scheduling for Thermal Management in Multicore Processors", in Proceedings of the International Symposium on Quality Electronic Design, Santa Clara, CA, USA, March 2015.
34. A. Langer, U. S. Palekar, E. Toton and L. V. Kalé, "Energy Efficient Computing for HPC Workloads on Heterogeneous Manycore Chips", in Proceedings of 6th International Workshop on Programming Models and Applications for Multicores and Manycores, PMAM'15, pp. 11–19, ACM, San Francisco, CA, USA, February 2015.
35. E. D. Carreño, P. O. A. Navaux & A. S. Sarates, "A Mechanism to Reduce Energy Waste in the Post-Execution of GPU Applications", in Journal of Physics: Conference Series, vol. 649, no. 1, article 012002, 2015.
36. A. Tiwari, L. Carrington and M. Schulz, "Predicting Optimal Power Allocation for CPU and Dram domains", in Proceedings of IEEE International Parallel and Distributed Processing Symposium Workshop, pp. 951–959, Orlando, FL, USA, May 2015.
37. A. K. Singh, L. S. Indrusiak and P. Dziuranski, "Value and Energy Optimizing of Dynamic Resource allocation in many-core HPC systems", in Proceedings of IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 180–185, Vancouver, BC, Canada, November-December 2015.
38. G. L. T. Chetsa, J. M. Pierson, L. Lefevre, G. D. Costa and P. Stolf, "Application-Diagnostic Framework for Improving the Energy Efficiency of Multiple HPC Subsystems", in Proceedings of 23rd Euro-micro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 62–69, Cambridge, UK, March 2015.
39. T. Patki, A. Sasidharan and D. K. Lowenthal et al., "Practical Resource Management in Power-Constrained HPC", in Proceedings of 24th International Symposium on High-Performance Parallel & Distributed Computing, HPDC'15, pp. 121–132, ACM, Portland, USA, June 2015.
40. A. Venkatesh, K. Hamidouche and A. Vishnu et al., "A Case for Application Based Energy-Efficient MPI at Runtime", in Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–12, Austin, USA, November 2015.
41. M. Pirahandeh and D. H. Kim, "Energy-Aware GPU RAID Scheduling for Reducing Energy Consumption in Cloud Systems", in Computer Science and its Applications, pp. 705–711, Springer, Berlin, Heidelberg, Germany, 2015.
42. J. Proficz and P. Czarnul, "Performance and Power-aware Modeling of MPI Applications for Cluster Computing", in Proceedings of 11th International Conference on Parallel Processing and Applied Mathematics - PPAM 2015, vol. 9574, pp. 199–209, Springer, Krakow, Poland, September 2015.
43. W. Piątek, G. DaCosta and A. Oleksiak, "Energy and Thermal Models for Simulation of Workload and Resource Management in Computing Systems", in Simulation Modeling, Practice and Theory, vol. 58, pp. 40–54, 2015.
44. G. Kecskemeti, "Dissect-CF: A Simulator to Foster Energy-Aware Scheduling in Cloud Infrastructure", in Simulation, Modeling, Practice and Theory, vol. 58, pp. 188–218, 2015.
45. A. C. Orgerie, L. Lefevre and M. D. D. Assuncao, "A Survey on Techniques for Improving the Energy Efficiency of Large Scale Distributed Systems", in ACM Computing Surveys, vol. 46, no. 4, pp. 1–47, 2014.
46. R. Schöne and D. Molka, "Integrating Performance Analysis and Energy Efficiency Optimizations in a Unified Environment", in Computer Science—Research and Development, vol. 29, no. 3, pp. 231–239, 2014.

47. R. E. Grant, J. H. Laros, S. L. Olivier, A. K. Porterfield and R. Brightwell, "Metrics for Evaluating Energy Saving Techniques for Resilient HPC Systems", in Proceedings of IEEE International Symposium and Workshop on Parallel & Distributed Processing, pp. 790–797, Cancun, Mexico, May 2014.
48. T. Mastelic, H. Claussen, J. M. Pierson, I. Brandic, A. Oleksiak, and A. V. Vasilakos, "Cloud Computing: Survey on Energy Efficiency", in ACM Computing Surveys, vol. 47, no. 2, pp. 1–36, 2014.
49. T. Li, V. K. Narayana, and T. El-Ghazawi, "Symbolic Scheduling of Concurrent GPU Kernels for Performance and Energy Optimizations", in Proceedings of 11th ACM Conference on Computing Frontiers, CF'14, pp. 36:1–36:10, ACM, Cagliari, Italy, May 2014.
50. P. Libuschewski, D. Siedhoff, P. Marwedel and H. Muller, "Multi-Objective, Energy-Aware GP-GPU Design Exploration for Medical Industrial Applications", in Proceedings of 10th International Conference on Signal-Image Internet-Based Systems, pp. 637–644, Washington, DC, USA, November 2014.
51. A. M. Malik, A. Nandamuri, B. M. Chapman and A. Qawasmeh, "Power and Energy Footprint of OPENMP Programs using OPENMP Runtime API", in Proceedings of Energy Efficient Supercomputing Workshop, pp. 79–88, New Orleans, LO, USA, November 2014.
52. D. Bodas, M. Rajappa, J. Song, and A. Hoffman, "Simple Power-Aware Scheduler to Limit Power Consumption by HPC System with in a Budget," in Proceedings of 2nd International Workshop on Energy Efficient Supercomputing, E2SC'14, pp. 21–30, IEEE Press, USA, November 2014.
53. G. L. Tsafack Chetsa, J. M. Pierson, L. Lefevre, G. Da Costa and P. Stolf, "Exploiting Performance Counters to Predict and Improve Energy Performance of HPC Systems", in Future Generation Computer Systems, vol. 36, pp. 287–298, 2014.
54. K. Fukazawa, M. Aoyagi and M. Ueda et al., "Power Consumption Evaluation of an MHD Simulation with CPU Power Capping," in Proceedings of 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 612–617, Chicago, USA, May 2014.
55. A. Auweter, M. Brehm and A. Bode et al., "A Case Study of Energy Aware scheduling on Super MUC", in Proceedings of 29th Int. Conference on Super Computing, ISC 2014, pp. 394–409, Springer, June 2014.
56. P. Czarnul and P. Ro'sciszewski, "Optimization of Execution Time under Power Consumption Constraints in a Heterogeneous Parallel System with GPUs and CPUs", in Distributed Computing and Networking, pp. 66–80, Springer, 2014.
57. S. Aita, S. Miwa and H. Nakamura, "Performance Estimation of HPC Systems with Energy Efficient Ethernet Technology", in Computer Science–Research and Development, vol. 29, no. 3, pp. 161–169, 2014.
58. H. Casanova, A. Legrand, A. Giersch, F. Suter and M. Quinson, "Versatile, Scalable and Accurate Simulation of Distributed Applications and Platforms", in Journal of Parallel and Distributed Computing, vol. 74, no. 10, pp. 2899–2917, 2014.
59. E. D'Hollander, I. Foster, J. Dongarra, G. Joubert and L. Grandinetti, "Transition of HPC towards Exa-Scale Computing", in Advances in Parallel Computing, Vol. 24, IOS Press, 2013.
60. R. Ge, R. Vogt, A. Alam, J. Majumder, Z. Zong and M. Burtscher, "Effects of Dynamic Voltage and Frequency Scaling on a k20GPU", in Proceedings of 42nd International Conference on Parallel Processing, pp. 826–833, October 2013.
61. B. Unni, A. Kumar, N. Parveen and B. S. Bindhumadhava, "An Intelligent Energy Optimization Approach for MPI based Applications in HPC Systems," in CSI Transactions on ICT, vol. 1, no. 2, pp. 175–181, 2013.
62. D. Li, M. Schulz, B. R. de Supinski, K. W. Cameron and D. S. Nikolopoulos, "Strategies for Energy-Efficient Resource Management of Hybrid Programming Models", in IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, pp. 144–157, 2013.
63. O. Sarood, L. Kal'e, A. Langer, B. de Supinski and B. Rountree, "Optimizing Power Allocation to CPU and Memory Subsystems in over Provisioned HPC Systems", in Proceedings of IEEE International Conference on Cluster Computing (CLUSTER), pp. 1–8, September 2013.
64. J. Kolodziej, L. Wang, S. U. Khan, N. Min-Allah, A. Byrski and S. Madani, "Hierarchical Genetic-Based Grid Scheduling with Energy Optimization", in Cluster Computing, vol. 16, no. 3, pp. 591–609, 2013.
65. J. K. Kurowski, W. Piatek, A. Oleksiak, J. Weglarz, T. Piontek, and Przybyszewski, "DCworms—A Tool for Simulation of Energy Efficiency in Distributed Computing Infrastructures" in Simulation, Modeling, Practice and Theory, vol. 39, pp. 135–151, 2013.
66. A. R. Khomane, A. R. Surve and S. Cheke, "Energy Awareness in HPC: A Survey", in International Journal of Computer Science and Mobile Computing, vol. 2, no. 3, pp. 46–51, 2013.
67. C. H. Hsu, S. W. Poole and J. A. Kuehn, "Towards Efficient Supercomputing: Searching for the Right Efficiency Metric", in Proceedings of the 3rd ACM/SPEC Int. Conference on Performance Engineering, ICPE'12, pp. 157–162, ACM, March 2012.
68. S. Benedict, "Review: Energy-Aware Performance Analysis Methodologies for HPC Architectures—an Exploratory Study", in Journal of Network and Computer Applications, vol. 35, no. 6, pp. 1709–1719, 2012.
69. M. H'ahnel, B. D'obel, M. V'olp and H. H'artig, "Measuring Energy Consumption for Short Code Paths using RAPL", in ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 3, p. 13, 2012.
70. B. Rountree, B. R. de Supinski, D. H. Ahn, D. K. Lowenthal, and M. Schulz, "Beyond DVFS: A First Look at Performance under a Hardware-Enforced Power Bound", in Proceedings of 26th IEEE Int. Symposium & Workshop on Parallel and Distributed Processing & PhD Forum, pp. 947–953, IEEE, May 2012.
71. A. Tiwari, J. Peraza, M. Laurenzano, A. Snively and L. Carrington, "Green queue: customized large- scale clock frequency scaling", in Proceedings of 2nd Int. Conference on Cloud and Green Computing, pp. 260–267, Xiangtan, China, Nov- 2012.
72. O. M'ammel'a, R. Basmadjian, M. Majanen, A. Giesler, H. DeMeer and W. Homberg, "Energy-Aware Job Scheduler for High-Performance Computing", in Computer Science–Research and Development, vol. 27, no. 4, pp. 265–275, 2012.
73. L. Wang, J. Dayal and S. U. Khan, "Thermal Aware Workload Placement with Task-Temperature Profiles in a Data Center", in Journal of Super Computing, vol. 61, no. 3, pp. 780–803, 2012.
74. Z. Wang, "Thermal-aware Task Scheduling on Multicore Processors", Ph.D. thesis, AAI3569706, University of Florida, Gainesville, USA, 2012.
75. A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud computing", in Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, 2012.
76. B. Aksanli, L. Zhang, J. Venkatesh and T. Rosing, "Utilizing Green Energy Prediction to Schedule Mixed Batch and Service Jobs in Data Centers", in ACM SIGOPS Operating Systems Review, vol. 45, no. 3, p. 53, 2012.
77. D. Kliazovich, S. U. Khan and P. Bouvry, "GreenCloud: A Packet-Level Simulator of Energy-Aware Cloud Data Centers", in Journal of Supercomputing, vol. 62, no. 3, pp. 1263–1283, 2012.
78. B. Aksanli and J. Venkatesh, "Rosing Using Data Center Simulation to Evaluate Green Energy Integration", in Computer, vol. 45, no. 9, pp. 56–64, 2012.
79. A. Beloglazov, Y. C. Lee, R. Buyya and A. Zomaya, "A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems", in Advances in Computers, vol. 82, pp. 47–111, Elsevier, 2011.
80. C. Cai, S. U. Khan, L. Wang and J. Tao, "Energy-Aware High Performance Computing: A Taxonomy Study", in Proceedings of 17th Int. Conference on Parallel and Distributed Systems, pp. 953–958, Tainan, Taiwan, December 2011.
81. S. Bak, K. Kurowski, M. Krystek, A. Oleksiak, J. Weglarz and W. Piatek, "GSSIM—a Tool for Distributed Computing Experiments", in Scientific Programming, vol. 19, no. 4, pp. 231–251, 2011.
82. R. N. Calheiros, A. Beloglazov, R. Ranjan, C. A. De Rose and R. Buyya, "Cloudsim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", in Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.
83. S. K. Gupta, A. Banerjee, R. R. Gilbert, T. Mukherjee, Z. Abbasi and G. Varsamopoulos, "GDOSim: A Tool for Analyzing Green Data Center Design and Resource Management Techniques", in Proceedings of 18th Int. Conference and Workshop on Scientific Programming Green Computing, pp. 1–8, IEEE, July 2011.

84. S. Ostermann, R. Prodan and K. Plankensteiner, "Using a NewEvent-Based Simulation Framework for Investigating ResourceProvisioning in Clouds", in Scientific Programming, vol. 19, no. 2-3, pp. 161–178, 2011.
85. T. E. Carlson, L. Eeckhout, W. Heirman and S. Niper, "Exploringthe level of abstraction for scalable and accurate parallel multi-coresimulation", in Proceedings of International Conference on HighPerformance Computing, Networking, Storage and Analysis, pp. 1–12, November 2011.
86. D. Bedard, R. Fowler, M. Y. Lim and A. Porterfield, "Powermon: fine-grained and integrated power monitoring for commoditycomputer systems," in Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), pp. 479–484, March 2010.
87. M. Ware, M. Floyd and K. Rajamani et al., "Architecting for power management: the IBM@ power7TM approach", in Proceedings of the HPCA– 16th International Symposium on HPC Architecture, pp. 1–11, Bangalore, India, January 2010.
88. D. Li, M. Schulz, B. R. de Supinski, D. S. Nikolopoulos and K.Cameron "Hybrid MPI/OPENMP Power-Aware Computing", inProceedings of IEEE International Symposium on ParallelDistributed Processing (IPDPS), pp. 1–12, USA, April 2010.
89. D. Terpstra, H. You, H. Jagode and J. Dongarra, "CollectingPerformance of Data with PAPI-C," in Tools for High PerformanceComputing 2009.
90. J. J. Chen, N. Fisher, S. Wang, and L. Thiele, "Thermal-AwareGlobal Real-Time Scheduling on Multi-core Systems," inProceedings of 15th IEEE Real-Time, Embedded Technology and Applications Symposium, pp. 131–140, USA, April 2009.
91. S. Huang, W. Feng and S. Xiao, "Survey on the Energy Efficiencyof Graphics Processing Units for Scientific Computing", inProceedings of IEEE International Symposium on Parallel andDistributed Processing, pp. 1–8, Rome, Italy, May 2009.
92. S. Li, J. Ho Ahn, J. B. Brockman et al., "McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multi-core andMany-core Architectures," in Proceedings of 42nd AnnualIEEE/ACM International Symposium on Micro Architecture, pp.469–480, ACM, USA, December 2009.
93. B. Rountree, S. Funk, D. K. Lowenthal, B. R. de Supinski, V. W.Freeh and M. Schulz, "Bounding Energy Consumption in Large-Scale MPI Programs", in Proceedings of ACM/IEEE Conference onSupercomputing, SC'07, pp. 49:1–49:9, ACM, November 2007.
94. F. Xhafa, L. Barolli, J. Carretero and A. Duresi, "Requirements for an event-based simulation package for grid systems", in Journalof Interconnection Networks, vol. 8, no. 2, pp. 163–178, 2007.
95. C. Isci, M. Martonosi and G. Contreras, "Live, Rruntime phasemonitoring and prediction on real systems with application todynamic power managemen,," in Proceedings of 39th AnnualIEEE/ACM International Symposium on Micro Architecture(MICRO'06), pp. 359–370, USA, December 2006.
96. M. Y. Lim, D. K. Lowenthal and V. W. Freeh, "Adaptive,transparent frequency and voltage scaling of communication phasesin MPI programs", in Proceedings of ACM/IEEE Conference onSupercomputing, p.14, USA, Nov-2006.
97. V. W. Freeh and D. K. Lowenthal, "Using multiple Energy gears inMPI programs on a power-scalable cluster", in Proceedings of 10th ACM SIGPLAN symposium on Principles and practice of parallelprogramming, p. 164, ACM, USA, June 2005.
98. J. Moore, P. Ranganathan, J. Chase and R. Sharma, "Making Scheduling Cool: Temperature-Aware workload placement in datacenters", in Proceedings of USENIX Annual Technical Conference(ATEC), p. 5, USENIX Association, USA, April 2005.
99. M. Wiecezorek, T. Fahringer and R. Prodan, "Scheduling ofScientific Workflows in the ASKALON Grid Environment," in ACM SIGMOD Record, vol. 34, no. 3, pp. 56–62, 2005.

Dr. Mohammad Zubair Khan is presently working as Associate Professor in Deptt. of CS, College of Computer Science and Engineering, Taibah University, Medina, KSA. He has authored more than 40 research. His area of interest is parallel or distributed computing, theory of computations and computer networks. Currently he continues his research in the field of data mining, big data and machine learning.

Dr. Ravendra Singh is presently working as a Professor in Department of CS & IT, Faculty of Engineering, M.J.P. Rohilkhand University, Bareilly, India. He has more than 21 year academic experience including 17 years of research. He has authored over 90 publications and 4 books. In past he did maximum research in the field of parallel and distributed computing or computer network regarding task allocation & scheduling. Currently his research includes Artificial intelligence, Machine learning, Deep learning and its applications.

AUTHORS PROFILE



Shailesh Saxena is a Ph.D scholar in CSE at MJP Rohilkhand University, Bareilly, India. His main area of research is computing. He published several research papers on Distributed Computing, Grid Computing and Cloud Computing. Now he continues his research on Energy-Aware Computing or Green Computing.