

# Modern methods of energy consumption optimization in FPGA-based heterogeneous HPC systems

Oleksandr V. Hryshchuk, Sergiy P. Zagorodnyuk

Taras Shevchenko National University of Kyiv, 64/13 Volodymyrska Str., Kyiv, 01601, Ukraine

## Abstract

High-Performance Computing (HPC) systems play a pivotal role in addressing complex computational challenges across various domains, but their escalating energy consumption has raised concerns regarding sustainability and operational costs. This paper presents a comprehensive investigation into the parametrization and modeling of energy consumption in heterogeneous HPC systems, aiming to provide valuable insights for optimizing energy efficiency while preserving performance. We begin by characterizing the heterogeneity within modern HPC environments, which encompass diverse hardware components, such as CPUs, GPUs, FPGAs, and accelerators. Our research delves into modeling techniques, leveraging heuristics methods and statistical approaches to construct accurate predictive models for energy consumption. Furthermore, we explore the integration of dynamic power management strategies, such as DVFS (Dynamic Voltage and Frequency Scaling) and task scheduling, to optimize energy usage without compromising performance. This paper provides a vital foundation for sustainable HPC practices, enabling researchers and practitioners to make informed decisions for achieving enhanced energy efficiency without sacrificing computational performance.

## Keywords

high-performance computing (HPC), FPGA, power modeling, power analysis, heterogeneous computing, power saving, task scheduling,

## 1. Introduction


Today's large-scale computing systems, such as data centers and high-performance computing clusters (HPCs), are severely limited by power and cooling costs for extremely large-scale (or exascale) problems. The steady increase in electricity consumption is a growing concern for several reasons, such as cost, reliability, scalability, and environmental impact. Nowadays data centers use 200 TWh per year and contribute near 0.3% of whole carbon emissions in the world, when entire complex of ICT (Information and computing technology) devices produce up to 2% of it [1]. Best case scenario model predicts that in 2030 ICT will share 8% of whole electricity consumption in the world [2], while worst case scenario anticipate 51% of global electricity usage. This potential increase in power consumption and, sequentially, cost of computing operations leads researcher and engineers to investigate and develop new techniques and approaches to optimize power management in HPC systems and in ICD domain in general.


---

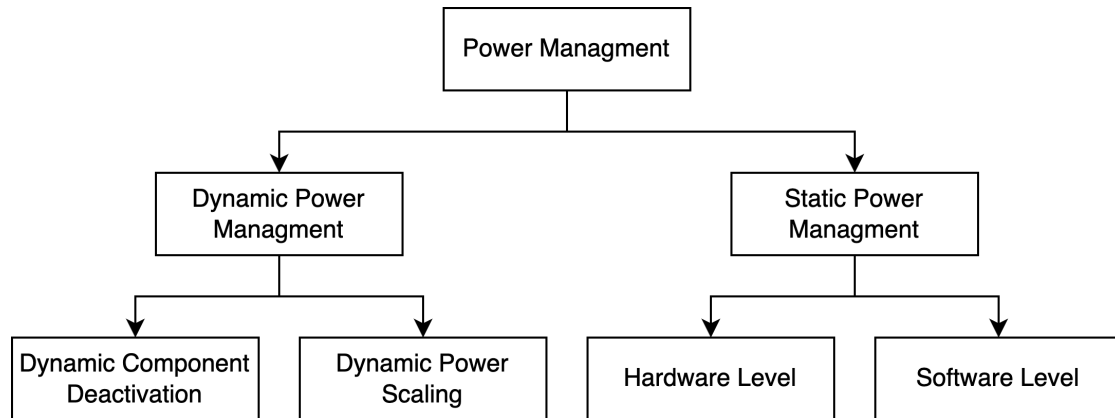
CS&SE@SW 2023: 6th Workshop for Young Scientists in Computer Science & Software Engineering, February 2, 2024, Kryvyi Rih, Ukraine

✉ oleksandr\_hryshchuk@knu.ua (O. V. Hryshchuk); szagorodniuk@gmail.com (S. P. Zagorodnyuk)

🆔 0009-0007-9926-4231 (O. V. Hryshchuk); 0000-0003-3415-7746 (S. P. Zagorodnyuk)

 © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)



**Figure 1:** General classification of power management methods in computer systems.

Present-day there are set of methods and approaches to resolve this energy optimization issue, mainly only for homogeneous CPU-based HPC systems. General taxonomy of this techniques, suggested in [3] and depicted on figure 1 and can be divided into two main groups SPM (static power management) and DPM (dynamic power management). SPM methods, divided in two separate groups (for hardware and software level management) usually defined during design time and cannot be changed in runtime. Hardware SPM techniques can be detailed and split into three separate groups [3]:

1. Circuit level
2. Logic level
3. Architecture level

DPM methods widely used in HPC [4] systems can be divided into two main groups – DCD (Dynamic component Deactivation), based on predictive and heuristic approaches, and DPS (Dynamic Power Scaling), like resource throttling and DVFS (Dynamic Voltage Frequency Scaling). This techniques can be a foundation for more complicated optimization methods, in example, task scheduling based on DVFS [5] or DCD heuristics applications [4].

Methods described before can be used on different hardware platforms, both homogeneous (well-studied nowadays) and heterogeneous (with GPU, TPU, FPGA and CGRA), which became popular in HPC according to a survey on Deep Learning hardware accelerators for heterogeneous HPC Platforms [6]. At the same time number of scientific papers on energy-aware optimization for HPC systems with FPGA controllers are extremely low (1-3 per year), compared to all researches about “FPGA heterogeneous computing” (see figure 2 with data obtained from app.dimensions.ai) which indicates a limited number of solutions in this domain, so this work will be focused on heterogeneous applications of energy-aware optimizations in HPC systems.

## 2. Energy optimization theory

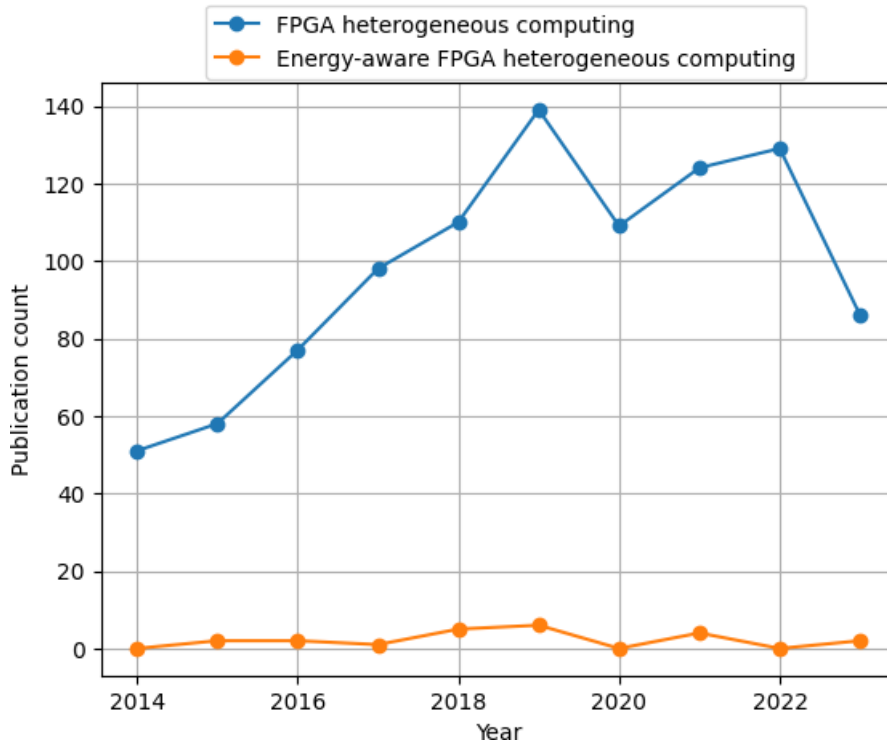
### 2.1. Optimization problem definition for task scheduling

In introduction section was mentioned that optimization techniques can be divided into hardware and software types, first of them are case-specific for different variations of hardware like CPU, memory chips, NIC, etc., while software-defined approaches can be generalized and provide a solution for disparate equipment with same characteristics/types, in example, homogeneous or heterogeneous GPU and TPU-based HPC clusters [7]. Such software solutions are often leads to energy-efficient task-scheduling methods, optimization problem for which can be defined in a way that described next.

For a finite set of jobs(task)  $J$  and a finite set of resources  $R$ ,  $time(j, r)$  is a function, that returns time of execution of job  $j \in J$  on resource  $r \in R$  [4]. Then scheduling can be described as task of finding a set of start times  $\{s_1, s_2, \dots, s_{|J|}\}$  for jobs, allocated to resources  $\{a_1, a_2, \dots, a_{|J|}\}$  in conditions where:

$$\forall s_x : \nexists s_y : s_x \leq s_y + time(y, A_y) \wedge s_y \leq s_x + time(x, A_x) \wedge a_x = a_y, \forall a_x : x \in R \quad (1)$$

Additional optimization conditions (see equation 2) can be applied to provided scheduling, where optimization criteria can be finding maximum or minimum, depending on formulation of a function which involves simple metrics such as execution time, consumed energy, etc. [4].



**Figure 2:** Count of scientific publications per year on topic “FPGA heterogeneous computing” and “Energy-aware FPGA heterogeneous computing” from 2014 to 2023.

$$\min / \max \left( \text{OptimizationCriteria} \left( \{s_1, s_2, \dots, s_{|J|}\}, \{a_1, a_2, \dots, a_{|J|}\} \right) \right) \quad (2)$$

This model is extremely simplified and does not suitable for real applications due to several reasons – it assumes that one resource can take only one task at the time, number of available resources always equal or higher than number of jobs to complete and does not include impact of communication between tasks on nodes or computing elements. To resolve these problems and adapt model to real world upgraded model was suggested [4] – for two tasks  $x$  and  $y$  from set of jobs pairs  $D$ ,  $P_j$  is set of devices, which can be assigned for job  $j \in J$ , time of communication between jobs obtained from function  $comm(x, y, a_x, a_y)$ , then solution is a set of assignments  $A_j$  and start times  $\{s_1, s_2, \dots, s_{|J|}\}$  for each job, like it described in equations 3-6:

$$\forall x \in A_j : x \in P_j \quad (3)$$

$$\forall s_x : \nexists s_y : s_x \leq s_y + \text{time}(y, A_y) \wedge s_y \leq s_x + \text{time}(x, A_x) \wedge A_x \cap A_y = \emptyset \quad (4)$$

$$\forall \{x, y\} \in D : s_x + \text{time}(x, A_x) + \text{comm}(x, y, A_x, A_y) \leq s_y \quad (5)$$

With optimization condition:

$$\min / \max \left( \text{OptimizationCriteria} \left( \{s_1, s_2, \dots, s_{|J|}\}, A_1, \dots, A_{|J|}, D \right) \right) \quad (6)$$

This method involves enumeration of all jobs for all available resources, which leads to idea that solution can not be found in polynomial time, and it was proved that problem of energy-efficient active time [8] scheduling is NP-Complete [5], so to be able use this model there can be a two possible ways – use predefined constraints and precalculated configurations or use heuristic methods, in example genetic algorithms [9], to find solution during runtime.

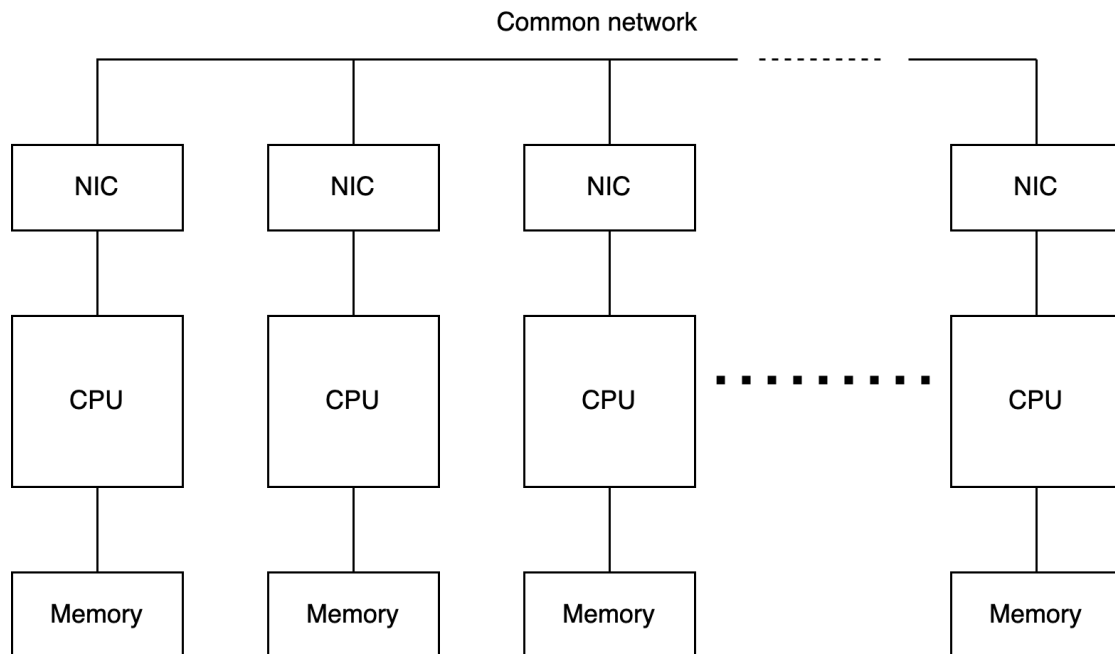
## 2.2. Optimization criteria

General optimization problem was described in previous section, and to be used in real HPC systems in requires properly defined optimization criteria. Existing solutions in this domain based on energy consumption metric (EC), or can take under consideration other properties, in example, execution time, etc. [4]. Power consumption can be described via energy itself (in joules or watts), or can be represented with more complicated models like instruction per joule or power per watt [10]. This approach used in Green500 rating as FLOPS per Watt metric [11].

More sophisticated can use combination of following metrics such as EC (energy consumption), ExecT (execution time), utilization, average weighted time, wait time, power, Pareto front, AST, AFT, clock frequency, work(job) per energy, reliability, electricity cost, temperature, EDP, EDF, Number of cores, Probability of execution, branch transition rate, cache efficiency, issue width [4]. In example new algorithm was proposed for reformed scheduling method with energy consumption constraint (RSMECC), based on AST, AFT and energy consumption metrics [12]. This algorithm can make it possible to more efficiently solve a wide range of computing tasks, including in the field of neural networks, complex 3D modeling and artificial intelligence.

### 3. Cluster architecture

Nowadays HPC clusters widespread around the world in different forms and variations, but generally main part of them are based on homogeneous massive parallel processor architecture (MPP), which inherited from older NUMA (non-uniform memory access) architecture [13]. This approach looks similar to shared-memory technology, but in this case each processor in cluster is connected to it's own part of memory and create entity of single independent node, which connected with other nodes via network interface card and common network (see figure 3). Absence of shared memory between nodes (not including common NAS) simplifies design and reduces inefficient components therefor improving scalability and stability of HPC system [13]. At the same time due to lack of shared memory, a processor core in one group must employ a different method to exchange data and coordinate with cores of other processor groups [14]. This issue become more visible for heterogeneous systems, based on CPUs form different series or types, or even for GRID computing systems [15].

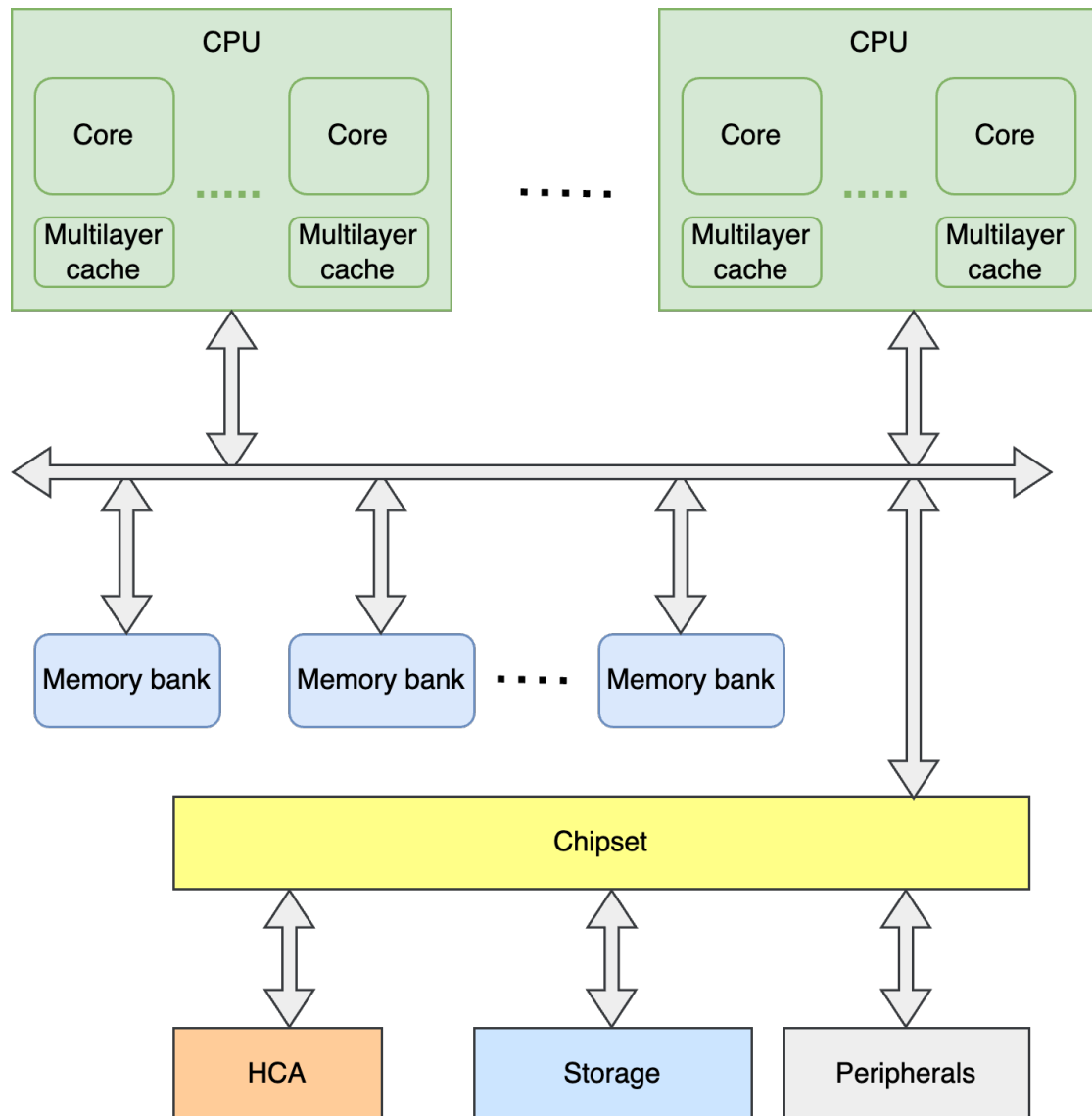


**Figure 3:** MPP HPC cluster architecture.

Another popular approach for building HPC systems is usage of symmetric multi-processors (SMP). It embodies a category of parallel architectures that harness the power of multiple processor cores to enhance performance by leveraging parallel processing, all the while upholding a unified memory structure that spans the entirety of the parallel computing system [13].

An SMP defines a self-contained and self-sustaining computer system equipped with all the subsystems and components essential for fulfilling the demands and facilitating the execution of various applications. It can operate independently to support user applications designed as shared-memory multi-threaded programs, serve as one among several equivalent subsystems

in a scalable MPP systems or commodity cluster, and work as a throughput computer for the simultaneous execution of independent concurrent tasks [14]. General architecture of SMP system depicted on figure 4.



**Figure 4:** Internal architecture of SMP HPC system.

### 3.1. Heterogeneous cluster architecture comparison

Heterogeneous computing in HPC refers to the utilization of diverse hardware accelerators, like general purpose graphic processing unit (GPGPU), field programmable gate array (FPGA), coarse-grained reconfigurable array (CGRA) [15] and specialized coprocessors, alongside traditional

CPU. This approach harnesses the strengths of different computing components to optimize performance and energy efficiency, making it particularly well-suited for workloads that can benefit from parallel processing. Most common heterogeneous clusters involve usage of coupled CPU and GPGPU as single node, therefore nowadays exists energy efficient solutions for this kind of HPC system, which was analyzed in [4].

But FPGA in same time in HPC is a new type of accelerators and less studied as it was shown in Introduction section of this paper. But nowadays there are existing works on this topic, in example the technique of cooperative CPU, GPU and FPGA task execution, based on EngineCL framework was suggested in [16]. Also, new approach, called Cooperative Heterogeneous Acceleration with Reconfigurable Multi-devices (CHARM) was proposed for multi hybrid accelerated cluster with GPU and FPGA coupling, which was implemented in “Albireo-nodes” in Cygnus cluster, based on CPU Intel Xeon Gold, GPU NVIDIA Tesla V100 x4 and FPGA Nallatech 520N with Intel Stratix10 [17]. Architecture of this nodes shown of figure 5.

Characteristic comparison for Cygnus supercomputer node and heterogeneous system from EngineCL test setup shown on table 1. At the same time, for EngineCL was shown that performance improvement from heterogeneity was obtained for all benchmark tasks (“Matrix multiplication”, “Mersenne Twister”, “Watermarking”, “Sobel Filter”, “Nearest Neighbor”, “AES Decrypt”), but energy consumption improvement was detected only for “Sobel Filter” [16], which leaves a research gap for searching energy-optimization methods for this kind of system.

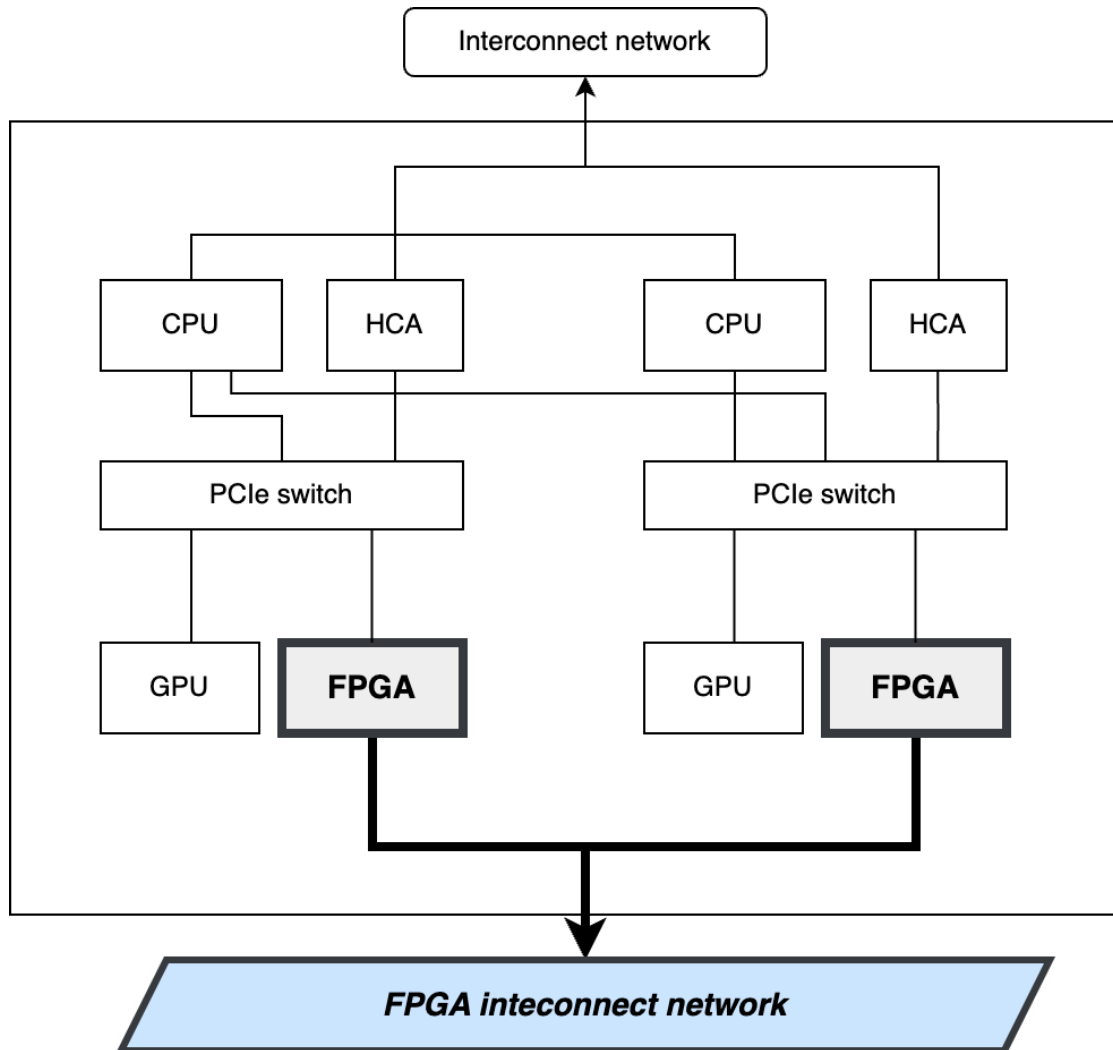
**Table 1**  
Comparison of Cygnus and EngineCL setup node specifications.

Characteristic	Cygnus	EngineCL test setup
CPU	Intel Xeon Gold x2	Intel Core i7-G700k
GPU	Nvidia Tesla V100x4 (32 Gb x4)	Nvidia GeForce GTX Titan X (12 Gb)
FPGA	Intel Stratix 10x2	Altera DE5NET Stratix V
RAM	192GB	64GB
Number of nodes	32 GPU+FPGA, 46 CPU-only	1
Energy-efficiency	N/A	1 of 6 benchmark tasks

Consequently, this two works have a lack of energy consumption optimization for described systems, and despite existing methods of power management and optimization described in survey of FPGA optimization methods for data center energy efficiency [18]. Finding “general” solution for FPGA-kind of system is complicated due to the necessity of reconfiguring of hardware for each specific task (job), but nevertheless, energy optimization constraints with proper criteria, described in “Energy optimization theory” section of this paper can be applied to multi-hybrid hardware FPGA systems to optimize power consumption.

## 4. Conclusions

This paper shows modern theories and approaches for power consumption planning and optimizations for heterogeneous HPC systems, including optimization model for MPP system, described in third section of this paper. As this problem in NP-complete, heuristics approaches



**Figure 5:** Internal architecture of Albireo-node from Cygnus cluster.

for finding solutions was mentioned. Results from mentioned solutions can be implemented on hardware or software level via DPM technologies. At the same time mentioned solutions is well suited to only CPU-GPU coupled systems, but not for CPU-GPU-FPGA coupled systems. For last one there is existing power management techniques, like easy-to use in FPGA DCD, but the is a lack of schedulers and general approaches for implementing solution from theoretical optimal model. Therefore, future work involves further search ways of amplification methods, including heuristic solutions of power consumption planning in FPGA-coupled HPC systems.



## References

- [1] N. Jones, How to stop data centres from gobbling up the world's electricity, *Nature* 561 (2018) 163–166. doi:10.1038/d41586-018-06610-y.
- [2] A. S. G. Andrae, T. Edler, On Global Electricity Usage of Communication Technology: Trends to 2030, *Challenges* 6 (2015) 117–157. doi:10.3390/challe6010117.
- [3] J. Haj-Yahya, A. Mendelson, Y. B. Asher, A. Chattopadhyay, *Energy Efficient High Performance Processors: Recent Approaches for Designing Green High Performance Computing*, Springer, 2018.
- [4] B. Kocot, P. Czarnul, J. Proficz, Energy-Aware Scheduling for High-Performance Computing Systems: A Survey, *Energies* 16 (2023). doi:10.3390/en16020890.
- [5] S. Saha, M. Purohit, NP-completeness of the Active Time Scheduling Problem, 2021. URL: <http://arxiv.org/abs/2112.03255>.
- [6] C. Silvano, D. Ielmini, F. Ferrandi, L. Fiorin, S. Curzel, L. Benini, F. Conti, A. Garofalo, C. Zambelli, E. Calore, et. al., A Survey on Deep Learning Hardware Accelerators for Heterogeneous HPC Platforms, 2023. doi:10.48550/arXiv.2306.15552.
- [7] V. Raca, S. Umboh, E. Mehofer, B. Scholz, Runtime and energy constrained work scheduling for heterogeneous systems, *Journal of Supercomputing* 78 (2022) 17150–17177. doi:10.1007/s11227-022-04556-7.
- [8] J. Chang, H. N. Gabow, S. Khuller, A Model for Minimizing Active Processor Time, in: L. Epstein, P. Ferragina (Eds.), *Algorithms – ESA 2012*, Lecture Notes in Computer Science, Springer, 2012, pp. 289–300. doi:10.1007/978-3-642-33090-2\_26.
- [9] A. Cocaña-Fernández, J. Ranilla, L. Sánchez, Energy-efficient allocation of computing node slots in HPC clusters through parameter learning and hybrid genetic fuzzy system modeling, *Journal of Supercomputing* 71 (2015) 1163–1174. doi:10.1007/s11227-014-1320-9.
- [10] M. Safari, R. Khorsand, Energy-aware scheduling algorithm for time-constrained workflow tasks in DVFS-enabled cloud environment, *Simulation Modelling Practice and Theory* 87 (2018) 311–326. doi:10.1016/j.simpat.2018.07.006.
- [11] T. Scogland, J. Azose, D. Rohr, S. Rivoire, N. Bates, D. Hackenberg, Node variability in large-scale power measurements: perspectives from the Green500, Top500 and EEHPCWG, in: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '15*, Association for Computing Machinery, New York, NY, USA, 2015. doi:10.1145/2807591.2807653.
- [12] Y. Hu, J. Li, L. He, A reformed task scheduling algorithm for heterogeneous distributed systems with energy consumption constraints, *Neural Computing and Applications* 32 (2020). doi:10.1007/s00521-019-04415-2.
- [13] T. Sterling, M. Brodowicz, M. Anderson, *High Performance Computing: Modern Systems and Practices*, Morgan Kaufmann, 2017.
- [14] S. Ramos, T. Hoefler, Modeling communication in cache-coherent SMP systems: a case-study with Xeon Phi, in: *Proceedings of the 22nd international symposium on High-performance parallel and distributed computing, HPDC '13*, Association for Computing Machinery, 2018, pp. 97–108. doi:10.1145/2462902.2462916.
- [15] P. S. Käsgen, M. Weinhardt, C. Hochberger, A Coarse-Grained Reconfigurable Array for High-Performance Computing Applications, in: *2018 International Conference on Re-*

- ConFigurable Computing and FPGAs (ReConFig), 2018, pp. 1–4. doi:10.1109/RECONFIG.2018.8641720.
- [16] M. Dávila, R. Nozal, R. Gran Tejero, M. Villarroya, D. Suárez Gracia, J. Bosque, Cooperative CPU, GPU, and FPGA heterogeneous execution with EngineCL, *The Journal of Supercomputing* 75 (2019). doi:10.1007/s11227-019-02768-y.
- [17] T. Boku, N. Fujita, R. Kobayashi, O. Tatebe, Cygnus - World First Multihybrid Accelerated Cluster with GPU and FPGA Coupling, in: *Workshop Proceedings of the 51st International Conference on Parallel Processing, ICPP Workshops '22*, Association for Computing Machinery, 2023, pp. 1–8. doi:10.1145/3547276.3548629.
- [18] M. Tibaldi, C. Pilato, A Survey of FPGA Optimization Methods for Data Center Energy Efficiency, *IEEE Transactions on Sustainable Computing* (2023) 343–362. doi:10.1109/TSUSC.2023.3273852.