On the Inevitability of Integrated HPC Systems and How they will Change HPC System Operations

Martin Schulz
Leibniz Supercomputing Centre (LRZ)
Garching, Germany
Technical University of Munich
Munich, Germany
schulzm@in.tum.de

Dieter Kranzlmüller
Leibniz Supercomputing Centre (LRZ)
Garching, Germany
Ludwig-Maximilians Universität (LMU)
Munich, Germany
dieter.kranzlmueller@lrz.de

Laura Brandon Schulz
Leibniz Supercomputing Centre (LRZ)
Garching, Germany
laura.schulz@lrz.de

Carsten Trinitis
Technical University of Munich
Munich, Germany
trinitic@in.tum.de

Josef Weidendorfer
Leibniz Supercomputing Centre (LRZ)
Garching, Germany
josef.weidendorfer@lrz.de

ABSTRACT

High-Performance Computing (HPC) is at an inflection point in its evolution. General-purpose architectures approach limits in terms of speed and power/energy, requiring the development of specialized architectures to deliver accelerated performance. Additionally, the arrival of new user communities and workloads—including machine learning, data analytics, and quantum simulation—increases the breadth of application characteristics we need to support, putting pressure on the complexity of the architectural portfolio. At the same time, data movement has been identified as a main culprit of energy waste, pushing hardware designers towards a tighter integration of the different technologies. The resulting integrated systems offer great opportunities in terms of power/performance tradeoffs, but also lead to challenges on the software side.

In this position paper, we highlight the trends leading us to integrated systems and describe their substantial advantages over simpler, single accelerated designs. Further, we highlight its impact on the corresponding software stack and its challenges and impact on the user. This introduces a different way to design, program and operate HPC systems, and ultimately the need to drop some long-held dogmas or believes in HPC systems.

KEYWORDS

HPC Architectures, Co-Design, Adaptive Systems

ACM Reference Format:
Martin Schulz, Dieter Kranzlmüller, Laura Brandon Schulz, Carsten Trinitis, and Josef Weidendorfer. 2021. On the Inevitability of Integrated HPC Systems and How they will Change HPC System Operations. In *International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies (HEART '21), June 21–23, 2021, Online, Germany*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3468044.3468046

HEART '21, June 21–23, 2021, Online, Germany

ACM ISBN 978-1-4503-8549-7/21/06.
https://doi.org/10.1145/3468044.3468046

1 INTRODUCTION

The architectural landscape of High-Performance Computing (HPC) is undergoing a seismic shift. The end of Dennard scaling in the mid-2000's—with its associated rise of multi-core platforms—and the approaching end of Moore's law requires significant changes in how we design, construct and operate major HPC platforms. How to adapt and continue to bring performance to the users of HPC systems is a central question framed by four major observations:

- Energy consumption is no longer merely a cost factor but also a hard feasibility constraint for facilities.
- Specialization is key to further increase performance despite stagnating frequencies and within limited energy bands.
- A significant portion of the energy budget is spent moving data and future architectures must be designed to minimize such data movements.
- Large-scale computing centers must provide optimal computing resources for increasingly differentiated workloads.

As a direct consequence of these observations, future architectures will have to provide a range of specialized architectures enabling a broad range of workloads, all under a strict energy cap. These architectures will have to be integrated within each node—as already seen in mobile and embedded systems—to avoid data movements across nodes or even worse, across system modules when switching between accelerator types.

This major architectural shift will also require substantial changes on the software side, both in how we program and operate these systems. Applications and runtime systems will need to be more dynamic, capable of identifying changes in workloads and phases and being able to react to those changes, e.g., by identifying the best-suited specialized architecture and directing compute along with power/energy accordingly.

In the remainder of this position paper, we first detail the four observations leading to this transformation of HPC in Section 2 and describe their consequences leading us to integrated systems in Section 3. We describe the impact on system software together with potential solutions in Section 4 followed by a discussion on more long-term changes we foresee in the operation of HPC systems in Section 5. We conclude the paper with final thoughts in Section 6.

2 TRENDS IN HPC

With the end of Dennard scaling in the mid-2000s, increasing clock speeds was no longer an option to gain further performance benefits. Instead, other options had to be found, leading to the era of multi-core and many-core parallelism. This enabled us to maintain exponential growth. With the end of Moore's law coming near, we face another critical junction that will directly impact how we design, implement, program and operate HPC systems.

2.1 The Need for Dynamic Energy Efficiency

The overarching topic of energy efficiency directly drives the development of the next generation of HPC systems. Modern compute nodes with accelerators reach the 2-3KW range, and with that a power density that is difficult to cool. Combined with the rising cost of energy and the increasing political and societal pressure towards carbon neutrality, energy consumption for HPC systems becomes not only a pure cost factor but a hard limiter for size, performance and capability of HPC systems.

This trend is further complicated by the push towards renewable energy sources, which show much higher volatility. As significant consumers of energy, HPC centers will have to react to this volatility and adjust their consumption, e.g., via frequency adjustments, intelligent scheduling or new energy storage systems, like hydrogen fuel cells, to match the available energy. Otherwise, prices may be prohibitive in peak times and endanger the stability of the overall grid due to the possible large swings in consumption. However, HPC centers may also help mitigate these issues, if they are capable of acting dynamically and adjusting loads based on the energy situation and with that can act as a buffer and stabilizer for the grid.

2.2 Cambrian Explosion of Architectures

Closely connected to energy efficiency is the current trend towards heterogeneity. Traditional opportunities for improvements achieved by riding Dennard scaling and Moore's law are coming to an end, which requires us to consider new architectural approaches. First, current general-purpose computing approaches are energy-hungry due to the need to decode and issue individual instructions as part of general von-Neumann programs. Second, the existing data formats are very wide and, in many cases, leave much of memory and compute unused; this has extreme power and performance implications. Specialization can avoid this general-purpose processing and, with that, reduce overheads.

This trend, often referred to as the Cambrian explosion of computer architectures [11], can be seen in a wide range of startups, in particular in the AI/ML space, but also the revival of dataflow technologies, compression hardware, the continued push to varying GPU and GPU-like systems and the integration of different core types within one system. Further, even within existing architectures, we see a trend towards specialized data formats, mainly covering reduced precision floating-point operations, which can be utilized for a subset of target applications.

2.3 Cost of Data Movement

In addition to improving computing in terms of performance and energy efficiency, data movement between the processing elements—general-purpose or specialized—plays an important role [6, 8]. Any data movement costs energy and the further data has to be moved, the more costly it is. Therefore, modern architectures need to strive to reduce data transmission paths and tightly couple processing elements that are used together. In particular, off-chip or even off-node communication is very costly, especially given current and foreseeable networking technologies that are not (yet) dynamically adjusting their power consumption.

2.4 New Workloads

At the same time, as we see this need for specialization and the reduction in data movements, we also observe a rise in new workloads and user communities requiring HPC services. In particular, machine learning and artificial intelligence and recently also the quantum computing communities are adding new requirements, which are often complementary to the needs of the traditional modeling and simulation HPC workloads. Further, ML, AI and High Performance Data Analytics (HPDA) have to work on large data streams and often lead to complex, tightly connected workflows, which add additional pressure to data movements and the associated costs. In particular, public computing centers, which have it in their mission to serve all these communities with their systems, will have to find the right tradeoffs—despite the specialization and the hard limiters in power and energy—between these workloads and their characteristics.

3 INTEGRATED NODE ARCHITECTURES

The need for specialized architectures, coupled with the widening breadth in workloads required to support the growing HPC community, leads to the development of heterogeneous systems where general-purpose compute cores are augmented with more specialized accelerators. We already see this trend with GPUs and, in some rare cases, FPGA-based systems, but this trend will intensify with multiple accelerators—GPUs, Tensor Cores, Inference Engines, ... all the way to quantum computers—within a single system. This will allow applications to utilize the matching architecture for their respective computation and with that complete the respective tasks both fast and in an energy efficient manner.

Heterogeneous Systems with massive heterogeneity can be built in multiple ways. The simplest way from an architecture perspective is to deploy separate compute modules for each specialized accelerator as independent clusters [16]. These clusters can be coupled via networking gateways enabling applications to compute partly on one cluster and transfer computation to a separate system when the respective architecture is more suitable.

This approach, while easy to build, maintain and grow as new accelerators become available comes with three major drawbacks: (1) it requires significant data movements between the compute modules when an application transitions between phases, limiting users to coarse-grained usage of accelerators due to the energy and performance cost of communication; (2) each accelerated compute module still has to rely on general-purpose hosts, which are typically underutilized and hence consume unnecessary energy; and (3) limits to the total problem size that can be run, as parallel processing across modules is hard to infeasible since individual application components would have to be able to be spread across vastly different architectures.

Integrated Heterogeneous Systems are a promising alternative, which integrate multiple specialized architectures on a single node while keeping the overall system architecture a homogeneous collection of mostly identical nodes. This allows applications to switch quickly between accelerator modules at a fine-grained scale, while minimizing the energy cost and performance overhead, enabling truly heterogeneous applications.

Such systems can be designed with a varying level of integration, from the use of PCIe style accelerators that connect specialized processing elements through on-node busses to integrating accelerators as part of the compute cores. The former type of system is obviously easier to develop and deploy, while the latter requires active participation of the core vendors through specialized designs. In both cases, though, the specialized computing elements are available for the application on node, typically with a single memory or with low-latency remote memory access options, allowing for easier and lower impact data management compared to having expensive cross-module transfers.

Integrated systems, however, also come with their challenges: while it is easy to run a single application across the entire system—since the same type of node is available everywhere—a single application is likely not going use all specialized compute elements at the same time, leading to idle processing elements. Therefore, the choice of the best-suited accelerator mix is an important design criterion during procurement, which can only be achieved via co-design between the computer center and its users on one side and the system vendor on the other. Further, at runtime, it will be important to dynamically schedule and power the respective compute resources. Using power overprovisioning, i.e., planning for a TDP[1] and maximal node power that is reached with a subset of dynamically chosen accelerated processing elements, this can be easily achieved, but requires novel software approaches in system and resource management, which we will discuss in Section 4.

While such highly integrated systems for HPC are currently still just emerging, the concept is already very wide-spread in other areas, especially in the mobile and embedded areas, where space and power constraints had a much earlier impact compared to HPC. Examples are the Playstation Cell Processor[2], which integrates general-purpose processing cores with special-purpose graphics and multimedia elements, and Apple's line of iPhone processors[3], which integrate a vast number of specialized units, including units for graphics, neural networks, photographic processing and matrix multiplication, as well as both fast and slow cores for high and low CPU intensive operations, respectively. On the programming side, works like by Binetto et al. [2] investigate and contrast the use of different architectures for particular problems.

The same conceptual ideas also hold for HPC: simple integrated systems with one or two specialized processing elements (e.g., with GPUs or with GPUs and tensor units) are already used in many systems. Research projects, like ExaNoDe [14], are currently investigating integration with promising results. Also, several commercial chip manufacturers are rumored to be headed in this direction. Currently and most prominently, the European Processor Initiative

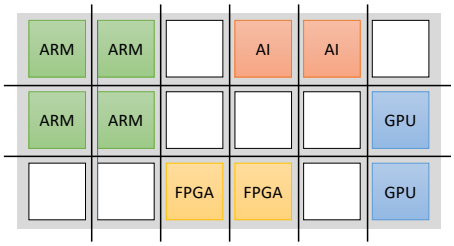

Figure 1: Heterogeneous chiplet concept combining different processing elements in one processor, based on design schematics of the EPI processor[4].

(EPI)[4] is looking at a customizable chip design combining ARM cores with different accelerator modules (Figure 1). Additionally, several groups are experimenting with clusters that GPUs and FPGAs within nodes, either for alternative workloads directed at the appropriate architecture [15] or for solving large parallel problems with algorithms mapped to both architectures [18]. Future systems are likely to push this even further, aiming at a closer integration and a larger diversity of architectures, leading to systems with more heterogeneity and flexibility in their usage.

4 IMPACT ON AND CHALLENGES FOR SYSTEM SOFTWARE

This shift towards integrated node architectures has not only implications on the hardware design but also—and perhaps significantly more so—on the software stack.

From a user's point of view, the first consideration is programmability, i.e., the needed programming environments and abstractions to exploit the different on-node accelerators. For widespread use, such support must be readily available and, in the best case, in a unified manner in one programming environment. OpenMP, with its architecture-agnostic target concept, is a good match for this. Domain-specific frameworks, as they are, e.g., common in AI, ML or HPDA (e.g., Tensorflow, Pytorch or Spark), will further help to hide this heterogeneity and help make integrated platforms accessible to a wide range of users.

4.1 Adaptive System Management

However, as discussed above, a single application will rarely be able to use all specialized units within a node simultaneously, leaving some of them idle. To compensate for this, we require a new level of adaptivity coupled with dynamic scheduling of compute and energy resources to exploit an integrated system fully. Energy will have to be directed to the most needed components for computation, while applications need to be scheduled to complementarily use the different resources.

This design is further driven by the initial observation that processing elements are no longer the limit of performance, but rather energy/power. This enables us to follow an overprovisioned design approach that assumes a maximal node power below the theoretically possible limit with all compute elements computing simultaneously. We can choose which computing elements to power or not,

[1] Thermal Design Point/Power, i.e., the maximal amount of power consumed and heat dissipated
[2] https://en.wikipedia.org/wiki/Cell_(microprocessor)
[3] https://en.wikipedia.org/wiki/Apple-designed_processors
[4] https://www.european-processor-initiative.eu/

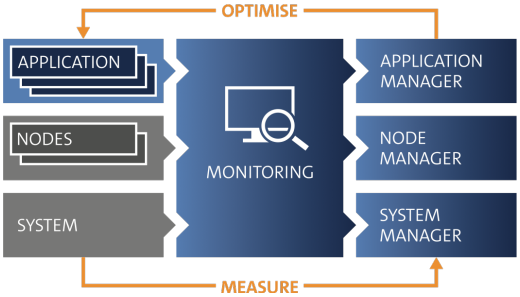

Figure 2: Adaptive Systems in the EU Project REGALE.

allowing for full utilization of the overall node in terms of power and energy. This enables us to set different tradeoffs for different applications and application phases, thereby supporting a diverse application mix on and across a single system.

The core of this adaptive management approach is a feedback loop, as depicted in Figure 2 and currently under investigation in the EU research project REGALE. REGALE uses measured information across all system layers and uses that information to adaptivity drive the entire stack:

- **Application Level** Changing application resources in terms of number and type of processing elements dynamically.
- **Node Level** Changing node settings, e.g., power/energy consumption via techniques like DVFS or power capping as well as node level partitioning of memory, caches, etc..
- **System Level** Adjusting system operation based on workloads or external inputs, e.g., energy prices or supply levels.

Coupled with a matching resource manager to control the adaptivity at these levels, systems can adjust themselves based on workloads and dynamically reconfigure applications, nodes and the system to extract maximal application performance and throughput. For this, we can build on top of a range of existing projects that target the active scheduling in heterogeneous systems [4] at the job level, adaptive task scheduling systems [7, 9] at the programming model level or using the concept of self-aware systems [1].

4.2 Holistic Monitoring

The core for any other adaptivity loop is the ability to monitor, analyze and predict system behavior. This requires systematic and holistic monitoring that captures all information about the HPC system, from application information to facility data. The DataCenter DataBase (DCDB) [12, 13], developed at the Leibniz Supercomputing Centre (LRZ) and deployed in production on LRZ's 26.9 PFlop/s production machine SuperMUC-NG, provides such capabilities. It uses a set of *P*ushers both on compute and infrastructure nodes to collect data and transmit them to *C*ollect Agents where they are stored in a federated set of Casandra databases (Figure 3). From there, it is available for analysis and use in system optimization. The concept of DCDB is extendable, allowing the integration of additional data sources, either directly from hardware sensors, from environmental facility sensors or via external measurement frameworks, like Amphere [10], which directly targets compute nodes with multiple types of acceleration.

5 BREAKING HPC DOGMAS

As discussed above, the creation of integrated systems will require a different system software stack that can dynamically monitor and adjust applications and the system alike. However, just by itself the impact of such a modified stack will be limited within current constraints on HPC systems. To fully exploit its impact, we also need to rethink several deeply entrenched dogmas in HPC. Such dogmas include that we only run a single job of a single user per node; that applications are static in their resource usage; that power and energy management are static and defined at system installation time; and that we have a clear separation between user, system and facility management.

5.1 On-Node Co-Scheduling

Current systems typically have a "one node/one user" policy, scheduling each node to a single user who utilizes the node in its entirety. This has obvious security advantages, while disadvantages are minimized due to limited node sizes and application characteristics using the entire node's resources. However, with integrated systems, we will see larger so-called "fatter" compute nodes making the scheduling granularity of whole nodes questionable. Additionally, a single application will no longer utilize a node fully due to different processing elements.

We argue that co-scheduling, which is not a new concept and has been discussed before [3, 17], is a viable way to counter these issues and exploit a node's computational capabilities fully. Using light-weight virtualization or containers, fat nodes can be split safely, without significant overhead and offered to different applications and users, thereby enabling a more targeted scheduling of the overall system resource. Assuming these applications rely on different specialized processing elements available on the same node, interference will be minimal and node resources can be used efficiently.

5.2 Dynamic Job Allocations

Current systems typically assume static job allocations, i.e., application jobs get assigned a fixed number of nodes at startup and then consume these resources until their termination. While this model is simple for the programmer and matches the typical programming model found in the dominating Message Passing Interface (MPI), it limits applications from finding their scaling sweet spot or from making use of additional resources should they become available.

We argue that enabling a more dynamic management of resources, i.e., the dynamic addition and removal of processing elements or nodes from applications, will lead to more efficient utilization of the overall system and less idle resources. This can further be combined with the idea of co-scheduling discussed above, enabling a dynamic match of applications to available processing elements that can even change over the runtime of an application.

5.3 The HPC PowerStack

Current systems typically assume a very conservative power provisioning: it is assumed that all nodes can be safely operated at their TDP without causing problems. While this a safe way to design reliable systems, it also includes significant safety margins, which

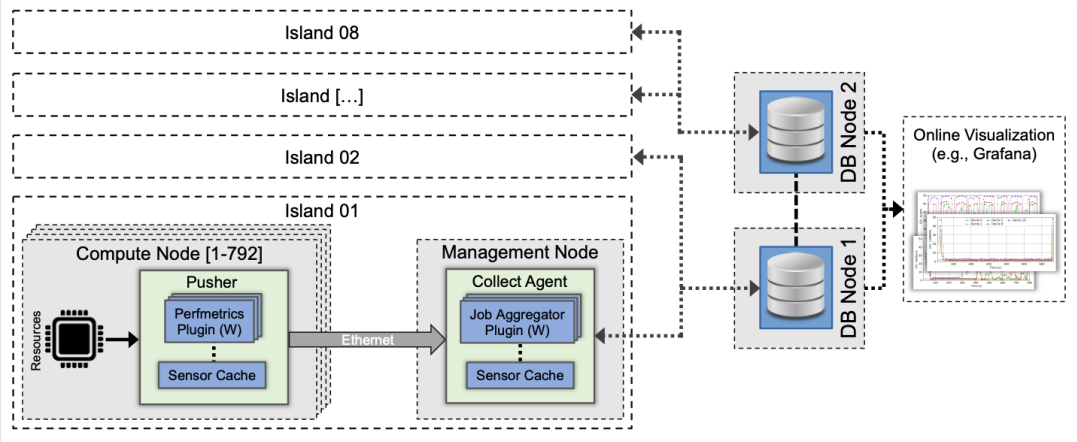

Figure 3: The DataCenter DataBase (DCDB) [12] deployed on SuperMUC-NG at LRZ https://doku.lrz.de/display/PUBLIC/SuperMUC-NG.

severely limit system performance, and for which power and energy are now hard limiters.

We argue that we need to soften this approach, leading us to the concept of overprovisioned systems. With a theoretical TDP much higher than the anticipated average or even peak load, we can use software to steer the available power and energy as needed.

We gathered experiences with this concept in a simple form on the SuperMUC system over the majority of its lifetime [5]. The system was operated at a reduced frequency by default, reducing energy consumption with a typically small impact on performance. Only applications that have shown in previous runs to significantly improve their performance—based on measurements coupled with a simple power model—were granted higher frequencies, keeping the overall energy low while boosting the performance of suitable applications only. This concept can easily be applied and extended to heterogeneous architectures with specialized processing elements, enabling access to specific components on-demand only when advantageous for performance.

In order to manage power/energy in such a system, we require a combined hardware/firmware/software solution that can steer power to where it is needed and limit overall power and energy envelopes based on site policies. This is currently being investigated in the PowerStack efforts[5], a working group consisting of key vendors, users and data centers with the goal of defining the major system components and interfaces. Its overall architecture is depicted in Figure 4, showing a hierarchical system combining node, job and system-level management driven by site-wide policies.

5.4 Application/System/Facility Integration

Current systems typically have a strong separation between applications (user-facing), the system itself (administrator-facing), and facilities (cooling, power/energy, building management facing). This arrangement enables a clear separation of concerns and highly reliable operations, but it also fosters many conservative decisions, e.g., on power supply and cooling.

[5]https://powerstack.caps.in.tum.de/

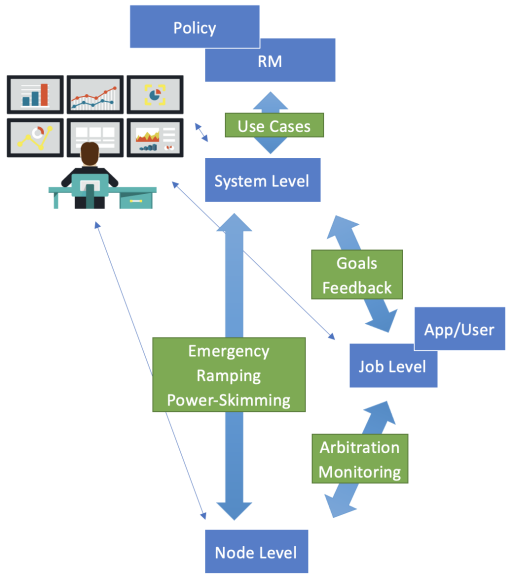

Figure 4: The HPC PowerStack concept.

We argue, with the limits shifting, that we can no longer afford such a clear separation and instead require a deeper integration between the related elements to obtain a holistic view on optimizing efficiency. E.g., recent experiments using DCDB [12] and Wintermute [13] show the benefits of controlling the cooling loop of a facility directly based on monitoring data capturing application characteristics.

6 FINAL THOUGHTS AND DISCUSSION

Changing technology trends require us to adapt and alter next-generation architectures in HPC if we want to continue to scale performance. Coupled with the fact that power and energy are shifting from desirable goals to hard limits defining the possible scale

of a system, we will see a shift from general-purpose computing to specialized computer architectures. In order to serve the broad and continuously growing user base, though, we need to integrate a wide range of such specialized processing elements into a single system, allowing us to dynamically pick the matching architecture for a particular problem or problem phase.

The type and level of this integration can vary. However, in this position paper we argue that such integration has to be on-node or even on-chip in order to: minimize and shorten expensive data transfers; enable fine-grained shifting between different processing elements running within a node; and to allow applications to utilize the entire machine for scale-out experiments rather than only individual modules or sub-clusters of a particular technology. Only this will enable us to design and deploy large-scale compute resources capable of providing a diversified portfolio of computing, at scale and at optimal energy efficiency.

The hardware approach alone, however, is insufficient. Aside from efforts in programmability and programming environments, we need to rethink our system software's design, in particular holistic monitoring, adaptive resource management and active power steering. Efforts in all these areas are already on their way:

- LRZ's DCDB enables the monitoring and analysis of system data to drive system adaptations;
- REGALE investigates the ability to implement and validate system-wide feedback loops to drive energy efficiency; and
- HPC PowerStack, together with organizations like ETP4HPC, aims to align the international community behind the common goal of active energy management, as it is needed to deploy integrated systems successfully.

At the same time, we need to ask if all preconceived notions on how we design, procure and operate HPC systems are still the right approaches for the next generation of such integrated systems. In particular, we argue that it is time to:

- re-evaluate co-scheduling capabilities to exploit heterogeneous resources fully;
- consider the ability to dynamically add or remove resources to/from applications to better react to global system events and changing constraints;
- dynamically manage power/energy in software; and
- ultimately blur the line between applications, systems and facilities, allowing for much closer integration at that level.

Combined, these efforts will enable a new generation of powerful, highly energy-efficient and widely usable HPC systems that can take us far beyond the exascale era.

ACKNOWLEDGMENTS

This work mentioned in this position paper has received funding from the DEEP-EST project under the EU H2020-FETHPC-01-2016 Programme grant agreement n. 754304, from the REGALE project under the EuroHPC Programme grant agreement n. 956560, as well as the German BMBF under grant number 16HPC039K.

REFERENCES

[1] Andreas Agne, Markus Happe, Achim Lösch, Christian Plessl, and Marco Platzner. 2014. Self-Awareness as a Model for Designing and Operating Heterogeneous Multicores. *ACM Transactions on Reconfigurable Technology and Systems* 7 (07 2014), 18. https://doi.org/10.1145/2617596

[2] Alécio Pedro Delazari Binotto, Dionisio Doering, Thorsten Stetzelberger, Patrick McVittie, Sergio Zimmermann, and Carlos Eduardo Pereira. 2013. A CPU, GPU, FPGA System for X-Ray Image Processing Using High-Speed Scientific Cameras. In *2013 25th International Symposium on Computer Architecture and High Performance Computing*. 113–119. https://doi.org/10.1109/SBAC-PAD.2013.1

[3] Jens Breitbart, Josef Weidendorfer, and Carsten Trinitis. 2015. Case Study on Co-scheduling for HPC Applications. 277–285. https://doi.org/10.1109/ICPPW.2015.38

[4] Gianluca C. Durelli, Marcello Pogliani, Antonio Miele, Christian Plessl, Heinrich Riebler, Marco D. Santambrogio, Gavin Vaz, and Cristiana Bolchini. 2014. Runtime Resource Management in Heterogeneous System Architectures: The SAVE Approach. In *2014 IEEE International Symposium on Parallel and Distributed Processing with Applications*. 142–149. https://doi.org/10.1109/ISPA.2014.27

[5] Carla Guillen, Carmen Navarrete, David Brayford, Wolfram Hesse, and Matthias Brehm. 2017. Energy Model Derivation for the DVFS Automatic Tuning Plugin: Tuning Energy and Power Related Tuning Objectives. *Computing* 99, 8 (Aug. 2017), 747–764. https://doi.org/10.1007/s00607-016-0536-3

[6] Utz-Uwe Haus. 2021. The Brave New World of Exascale Computing: Computation is Free, Data Movement is Not. Invited Talk at the TRR154/MINOA conference "Trends in Modelling, Simulation and Optimisation: Theory and Applications", https://minoa-itn.fau.de/wp-content/uploads/2021/03/TRR154-MINOA-20210303.pdf. https://doi.org/10.1109/ICPPW.2015.38

[7] Mario Kicherer, Fabian Nowak, Rainer Buchty, and Wolfgang Karl. 2012. Seamlessly Portable Applications: Managing the Diversity of Modern Heterogeneous Systems. *ACM Trans. Archit. Code Optim.* 8, 4, Article 42 (Jan. 2012), 20 pages. https://doi.org/10.1145/2086696.2086721

[8] Peter Kogge and John Shalf. 2013. Exascale Computing Trends: Adjusting to the "New Normal"' for Computer Architecture. *Computing in Science & Engineering* 15 (11 2013), 16–26. https://doi.org/10.1109/MCSE.2013.95

[9] Achim Lösch, Tobias Beisel, Tobias Kenter, Christian Plessl, and Marco Platzner. 2016. Performance-centric scheduling with task migration for a heterogeneous compute node in the data center. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*. 912–917.

[10] Achim Lösch, Alex Wiens, and Marco Platzner. 2018. *Ampehre: An Open Source Measurement Framework for Heterogeneous Compute Nodes*. 73–84. https://doi.org/10.1007/978-3-319-77610-1_6

[11] Satoshi Matsuoka. 2018. Cambrian Explosion of Computing and Big Data in the Post-Moore Era. In *Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing* (Tempe, Arizona) *(HPDC '18)*. Association for Computing Machinery, New York, NY, USA, 105. https://doi.org/10.1145/3208040.3225055

[12] Alessio Netti, Micha Müller, Axel Auweter, Carla Guillen, Michael Ott, Daniele Tafani, and Martin Schulz. 2019. From Facility to Application Sensor Data: Modular, Continuous and Holistic Monitoring with DCDB. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Denver, Colorado) *(SC '19)*. Association for Computing Machinery, New York, NY, USA, Article 64, 27 pages. https://doi.org/10.1145/3295500.3356191

[13] Alessio Netti, Micha Müller, Carla Guillen, Michael Ott, Daniele Tafani, Gence Ozer, and Martin Schulz. 2020. DCDB Wintermute: Enabling Online and Holistic Operational Data Analytics on HPC Systems. In *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing* (Stockholm, Sweden) *(HPDC '20)*. Association for Computing Machinery, New York, NY, USA, 101–112. https://doi.org/10.1145/3369583.3392674

[14] Alvise Rigo, Christian Pinto, Kevin Pouget, Daniel Raho, Denis Dutoit, Pierre-Yves Martinez, Chris Doran, Luca Benini, Iakovos Mavroidis, Manolis Marazakis, Valeria Bartsch, Guy Lonsdale, Antoniu Pop, John Goodacre, Annaik Colliot, Paul Carpenter, Petar Radojković, Dirk Pleiter, Dominique Drouin, and Benoît Dupont de Dinechin. 2017. Paving the Way Towards a Highly Energy-Efficient and Highly Integrated Compute Node for the Exascale Revolution: The ExaNoDe Approach. In *2017 Euromicro Conference on Digital System Design (DSD)*. 486–493. https://doi.org/10.1109/DSD.2017.37

[15] Michael Showerman, Jeremy Enos, Avneesh Pant, Volodymyr Kindratenko, Craig Steffen, Robert Pennington, and Wen-mei Hwu. 2009. QP: A Heterogeneous Multi-Accelerator Cluster. (01 2009).

[16] Estela Suarez, Norbert Eicker, and Thomas Lippert. 2019. *Modular Supercomputing Architecture: From Idea to Production*. 223–255. https://doi.org/10.1201/9781351036863-9

[17] Carsten Trinitis and Josef Weidendorfer (Eds.). 2018. *Proceedings of the 3rd Workshop on Co-Scheduling of HPC Applications, COSH@HiPEAC 2018, Manchester, United Kingdom, January 23, 2018*. TUM Library.

[18] Kuen Hung Tsoi and Wayne Luk. 2010. Axel: A Heterogeneous Cluster with FPGAs and GPUs. In *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays* (Monterey, California, USA) *(FPGA '10)*. Association for Computing Machinery, New York, NY, USA, 115–124. https://doi.org/10.1145/1723112.1723134