Received: 22 April 2018 | Accepted: 22 April 2018
DOI: 10.1002/cpe.4661

WILEY

EDITORIAL

Heterogeneous and unconventional cluster architectures and applications

1 | INTRODUCTION

Recent trends in cluster computing and related topics, including processor and memory design, demonstrate a continuous growing need for more processing and memory, both in terms of capacity and performance. Cluster computing has been traditionally at the forefront of such computing systems and, usually, is one of the earliest adopters of future and emerging technologies. With this special issue, we gear to gather recent related works, hoping that the reader finds these contributions helpful for a better understanding of future directions in cluster computing. Before that, we will briefly present a short rationale on our view of cluster computing.

This rationale is based on two trends that are most important for such cluster architectures: first, the end of Dennard scaling has led to an era in which the growing amount of transistors, as described by Moore's law, cannot be simultaneously active because of an increasing power density. Second, applications continue to demand more processing and memory capacity. However, economy and technology laws imply that horizontal scaling is usually more cost effective than vertical scaling.

1.1 | Post-Dennard performance scaling

In particular, processor design heavily relies on a growing amount of transistors, which have to be continuously active. In the early days of CMOS technology, Dennard introduced a power law that describes the power consumption of a CMOS-based silicon die. According to this law, with "*a*" being the number of components, "*C*" capacity, "*V*" voltage, and "*f*" the operating frequency, the power consumption of a CMOS die is as follows:

$$P = a \cdot C \cdot V^2 \cdot f + V \cdot I_{\text{leakage}}. \tag{1}$$

Furthermore, he also described power scaling rules for CMOS silicon dies, defined by the observation that a transition to a new processing technology will decrease the feature size (ie, the size of a transistor gate) by a factor α. Based on this factor, characteristics including voltage, current, and capacity scale inversely. Formula (1) then shows that, for a given power budget, one can implement α^2 more components ”*a*” and even increase operating frequency *f* by a factor of α. Thus, Dennard scaling actually gave Moore's law its teeth by enabling constant power budgets and frequency scaling.

Unfortunately, since early 2015, this law is no longer applicable, mainly because voltage scaling is no longer possible because of saturated threshold voltages and because leakage power became a major contributor to the overall power consumption. As a result, a still growing amount of transistors, as described by Moore's law, now results in a growing power budget, which results intype a hard technical constraint. The usual escape path for post-Dennard performance scaling is two-fold: first, one can observe that frequency usually behaves linearly with regard to voltage, effectively making power consumption proportional to frequency cubed. For instance, reducing frequency by half would result in a relative power consumption of 1/8th, allowing to replicate one computational core 8 times while maintaining the power budget. Second, it is a common first-order approximation to assume that performance behaves proportional to frequency. To continue the previous example, such an 8-core design at half the initial frequency would now result in a relative performance improvement of a factor of 4. However, this is obviously only feasible if the application workloads exhibit enough parallelism. Given extreme examples of many-core processors with 1000s of vector units, heterogeneity is usually the chosen solution to also support the sequential parts of the workloads.

As a result, we are seeing a huge interest in many-core processors, which, however, only excel in performance for massively parallel workloads. Given that not all workloads fulfill this requirement, heterogeneous architectures are being deployed.

1.2 | Horizontal resource scaling

Applications continue to demand for an increasing amount of processing power and memory capacity, in particular pushed by Big Data and Machine Learning. For instance, training a recurrent deep neural network requires about 20 ExaFLOPs and still is not being trained with all data available. Similarly, in particular, deep learning required a plethora of data, leading to huge data collections for various tasks. However, the costs of resources like processors or memory do not scale linearly with capability. On the other hand, while the manufacturers are not very candid about the reasons behind, it is well known that the yield of a silicon die production is a function of the die area. While small dies are less likely to contain a manufacturing error, this probability will increase with die size. Process variation can have similar effects on operating frequency, making high-speed designs more sensitive to such variations.

2 | THEMES OF THIS SPECIAL ISSUE

With the series on Heterogeneous and Unconventional Cluster Architectures and Applications, we gear to gather most recent insights and ideas from the wide area of cluster computing. This special issue of the "International Journal of Concurrency and Computation: Practice and Experience" resembles our most recent selection.

In particular, this selection includes four interesting works.[1-4] Two of them were contributions from the last two workshop editions (HUCAA 2015 and 2016, both collocated with the International Conference on Parallel Processing - ICPP'15 and ICPP'16). Furthermore, the two other articles are contributed by the authors based on an open call for contributions of this special issue. All contributions were peer reviewed and received in between three and five reviews.

2.1 | Design considerations for GPU-aware collective communications in MPI

GPU accelerators have been established in the state-of-the-art clusters by offering high performance and energy efficiency. In GPU, such efficient communication among processes with their data residing in the GPU memory is of paramount importance to the application performance.

This paper investigates various algorithms in conjunction with the latest GPU features to improve GPU collective operations. For clusters with multi-GPU nodes, the authors of this paper propose a hierarchical framework that allows different algorithms at each hierarchy level. By studying various combination of algorithms, the authors of this article highlight the importance of choosing the right algorithm within each level. They evaluate their framework on MPI Allreduce and show promising performance results, specifically for large message sizes which are highly in-use in deep learning and big data applications.

They also show the benefit of using the Hyper-Q feature and the MPS service in jointly using different copy types to perform multiple inter-process communications. However, the authors of this paper show that efficient designs are required to further harness this potential. Accordingly, they propose Hyper-Q aware algorithms for GPU collectives. They evaluate our algorithms on MPI Allgather and MPI Allreduce operations. While their experimental results show the benefit of their algorithms, their profiling results indicate that this benefit is mainly rooted in overlapping different copy types.

2.2 | Energy-based tuning of metaheuristics for molecular docking on multi-GPUs

Virtual Screening methods (VS) simulate molecular interactions in silico to look for the best chemical compound that interacts with a given molecular target. VS are becoming increasingly popular to accelerate the drug discovery process and constitute hard optimization problems with a huge computational cost. To deal with these two challenges, the authors of this paper have created METADOCK, an application that (1) enables a wide range of metaheuristics through a parametrized schema, and (2) promotes the use of a multi-GPU environment within a heterogeneous cluster. Metaheuristics provide approximate solutions in a reasonable time frame, but given the stochastic nature of real-life procedures, the energy budget goes hand in hand with acceleration to validate the proposed solution.

This paper evaluates energy trade-offs and correlations with performance for a set of metaheuristics derived from METADOCK. The authors of this paper establish a solid inference from minimal power to maximal performance in GPUs and from there to optimal energy consumption. This way, ideal heuristics can be chosen according not only to best accuracy and performance but also to energy requirements. Their study starts with a preselection of parameterized metaheuristic functions, building blocks where we will find optimal patterns from power criteria while preserving parallelism through a GPU execution. They then establish a methodology to figure out the best instances of the parameterized kernels based on energy patterns obtained, which are analyzed from different viewpoints: performance, average power, and total energy consumed. The authors of this paper also compare the best workload distributions for optimal performance and power efficiency among Pascal and Maxwell GPUs on popular Titan models. The experimental results in this paper demonstrate that the most power efficient GPU can be overloaded in order to reduce the total amount of energy required by as much as 20%, finding unique scenarios where Maxwell does it better in execution time, but with Pascal always ahead in performance per watt, reaching peaks of up to 40%.

2.3 | SAHARA - Standard Architecture for Hardware Acceleration via Reconfigurable Arrays

Faster, lower power, and/or less expensive computation will be a software problem forever. Hardware can only make the challenge simpler or harder, and heterogeneous approaches exacerbate it. For emerging alternative computational technologies like quantum, optical, resistive (and other forms of analog computation), and/or biological computing (among others), to be successful, they must be integrated into the existing computational infrastructure (both hardware and software) if they are to realize their full potential. The increasingly main-stream options that reconfigurable logic represents (both fine and coarse grained) will also be most useful within an infrastructure that is sympathetic to legacy memory and storage models.

SAHARA is a reduction of computation into data wavefronts that, independent of the underlying technology, employs memory as the fundamental unit of computation within a simple data-flow model, essentially turning processing into a side-effect of the relevant data being made available to the logic that manipulates that data. No single aspect of SAHARA is “new”. Its foundations are more than 50 years old and started with Minsky's 1961 paper on Turing equivalence. SAHARA is an eminently useful abstraction of computation that has the potential of seamlessly integrating many disparate forms of computation behind a simple, common, architectural interface.

2.4 | Heterogeneous distributed computing based on high-level abstractions

The rise of heterogeneous systems has given place to great challenges for users, as they involve new concepts, restrictions and frameworks. Their exploitation is further complicated in the context of distributed memory systems, which require the usage of additional different programming paradigms and tools.

In this paper, the authors propose a novel approach to program heterogeneous clusters that is based on high-level abstractions such as tiles and hierarchical decomposition combined with the powerful APIs that data types and embedded languages can provide in languages such as C++. Rather than building their proposal from scratch, they have implemented it as a natural integration of the existing Hierarchically Tiled Arrays (HTA) and Heterogeneous Programming Library (HPL) projects, the first one being focused on distributed computing and the second one on heterogeneous processing. The result, called Heterogeneous Hierarchically Tiled Arrays (H^2TA), is very intuitive and easy to use thanks to the global view of the data and the single-threaded view of the execution that it provides at cluster level together with the transparency it provides with respect to the management of the heterogeneous devices. An evaluation comparing the proposal in this paper with previous MPI-based implementations shows its large programmability advantages and the reasonable overhead incurred.

3 | SUMMARY

Cluster computing is currently facing a pivotal point in time as we are hitting hard constraints about the future of CMOS processors. While, currently, most energy is still spent on computations, first research results show that an increasing fraction of overall energy is spent for data movements. Given the hard constraints on power consumption, one can imagine how influential this fundamental transition will be. Still, CMOS replacements like quantum computing, neuromorphic computing and many other candidates are either still nascent or will only be helpful for certain workloads. While it seems safe to assume that this will further increase heterogeneity in the future, we will have to find out if the currently narrow workload spectrum for these architectures can be extended or if generic computing in the future will have so solely rely on CMOS processors. In this context, we hope the readers of this special issue will find the contributions interesting and inspiring for future research.

ACKNOWLEDGMENTS

We in particular acknowledge the thorough work of our review board, which did an excellent and timely work on proving opinions and helpful feedback to the submitted articles. Furthermore, we are also thankful to the authors, which submitted their research contributions to our special edition. Last, but not least, we are also indebted to the continuous support by the editor-in-chief Geoffrey Fox, who is always available for assistance and recommendations regarding the publication procedure.

Holger Fröning[1]
Federico Silla[2]
[1] *Institute of Computer Engineering, Ruprecht-Karls University of Heidelberg, Germany*
[2] *Universitat Politècnica de València, Spain*

Correspondence
Holger Fröning, Institute of Computer Engineering, Ruprecht-Karls University of Heidelberg, Germany
Email: holger.froening@ziti.uni-heidelberg.de
Federico Silla, Universitat Politècnica de València, Spain
Email: fsilla@disca.upv.es

REFERENCES

1. Faraji I, Afsahi A. Design considerations for GPU-aware collective communications in MPI. *Concurrency Computat Pract Exper*. 2018;30:e4667.https://doi.org/10.1002/cpe.4667.
2. Viñas M, Fraguela BB, Andrade D, Doallo R. Heterogeneous distributed computing based on high level abstractions. *Concurrency Computat Pract Exper*. 2018;30:e4664.https://doi.org/10.1002/cpe.4664.
3. Pérez-Serrano J, Imbernón B, Cecilia JM, Ujaldón M. Energy-based Tuning of Metaheuristics for Molecular Docking on Multi-GPUs. *Concurrency Computat Pract Exper*. 2018;30:e4684.https://doi.org/10.1002/cpe.4684.
4. Mayhew D. SAHARA. *Concurrency Computat Pract Exper*. 2018;30:e4708.https://doi.org/10.1002/cpe.4708.