# Challenges in Deeply Heterogeneous High Performance Systems


Giovanni Agosta
*Politecnico di Milano*
*Milano, Italy*
*agosta@acm.org*

William Fornaciari
*Politecnico di Milano*
*Milano, Italy*
*william.fornaciari@polimi.it*

David Atienza
*EPFL*
*Lausanne, Switzerland*
*david.atienza@epfl.ch*

Ramon Canal
*Barcelona Supercomputing Center*
*and UPC, Barcelona, Spain*
*ramon.canal@bsc.es*

Alessandro Cilardo
*University of Naples Federico II*
*and CeRICT, Napoli, Italy*
*acilardo@unina.it*

José Flich Cardo
*Universitat Politècnica de València*
*València, Spain*
*jflich@disca.upv.es*

Carles Hernandez Luz
*Universitat Politècnica de València*
*València, Spain*
*carherlu@upv.es*

Michal Kulczewski
*PSNC*
*Poznań, Poland*
*kulka@man.poznan.pl*

Giuseppe Massari
*Politecnico di Milano*
*Milano, Italy*
*giuseppe.massari@polimi.it*

Rafael Tornero Gavilá
*Universitat Politècnica de València*
*València, Spain*
*ratorga@disca.upv.es*

Marina Zapater Sancho
*EPFL*
*Lausanne, Switzerland*
*marina.zapater@epfl.ch*



***Abstract*—RECIPE (REliable power and time-ConstraInts-aware Predictive management of heterogeneous Exascale systems) is a recently started project funded within the H2020 FETHPC programme, which is expressly targeted at exploring new High-Performance Computing (HPC) technologies. RECIPE aims at introducing a hierarchical runtime resource management infrastructure to optimize energy efficiency and minimize the occurrence of thermal hotspots, while enforcing the time constraints imposed by the applications and ensuring reliability for both time-critical and throughput-oriented computation that run on deeply heterogeneous accelerator-based systems. This paper presents a detailed overview of RECIPE, identifying the fundamental challenges as well as the key innovations addressed by the project, which span run-time management, heterogeneous computing architectures, HPC memory/interconnection infrastructures, thermal modelling, reliability, programming models, and timing analysis. For each of these areas, the paper describes the relevant state of the art as well as the specific actions that the project will take to effectively address the identified technological challenges.**



***Keywords*-HPC; heterogeneous computing; run-time management;**


## I. Introduction and long-term objectives

High-Performance Computing (HPC) is a vital infrastructure for both industry and social actors worldwide. In addition to traditional domains, such as oil & gas, finance, or weather forecasting, HPC is increasingly important for emerging domains such as bioinformatics and medicine. Furthermore, the emergence of heterogeneous hardware architectures and the trends towards "Green HPC" have prompted the major players to re-align their research priorities in HPC. In Europe, the ETP4HPC Strategic Research Agenda (SRA) [1] highlights the need for tightly coupled approaches for application development and management of HPC resources and energy to reliably achieve the desired Quality of Service (QoS). Other key challenges include the management of thermal issues through a balanced approach involving both hardware and software support, and the capability to predict failure occurrences.

In line with the SRA's vision, RECIPE targets emerging scenarios in HPC, which are characterized by new types of requirements, like time criticality and reliability, as well as new delivery models, e.g. cloud-based solutions [2], potentially open to a large audience of HPC users. These emerging scenarios require that new HPC platforms are able to support multiple applications running concurrently, possibly with conflicting Quality of Service requirements [3], [4], [5], [6], [7]. In addition, at the technology level, RECIPE addresses deep heterogeneity, based on dedicated accelerators like GPUs and FPGAs, as an enabling factor for improved energy efficiency, building on the results collected from previous research projects [8], [9].

The main focus in RECIPE is on resource management [10], targeted at performance predictability and reliability for HPC applications. To address current limitations, RECIPE will take a hierarchical approach to develop a Run-Time Management System (RTMS) partitioned in two layers, a *global* resource manager providing proactive fault tolerance and coarse-grained resource allocation, as well as a *local* resource manager, ensuring reactive fault tolerance and fine-grained resource allocation. The fine-grained resource management and predictive strategies will support the combined optimization of reliability and timing. The interface between the runtime management system and the application will be mediated by the programming model, supporting the expression of application expectations. Furthermore, the runtime management system will rely on hardware monitors and a hardware abstraction layer, providing a uniform view

of the available accelerators, including GPU, manycore, and FPGA devices, and exposing partitioning mechanisms used to ensure scalability, system-wide reconfigurability, and dedicated resource provisioning.

The ultimate objectives of the project will include:

- *Global manager and proactive strategies for reliability, energy efficiency and QoS*:
  Increasing the system efficiency in terms of performance/watt, due to power/performance/thermal aware allocation; Reducing the mean time to failure (MTTF) due to proactive strategies; Enforcing performance guarantees for critical applications.
- *Local manager and reactive strategies for reliability, energy efficiency and QoS*:
  Improving the overall utilization of non-CPU computing devices under heavy condition of massively parallel workload; Improving the Energy-Delay product with respect to current state-of-the-art HPC infrastructures; Reducing the probability of faulty executions; Reducing the average recovery time with respect to classical checkpoint-restart approaches.

To achieve the aforementioned goals RECIPE will develop technologies focusing on:

- *Fault prediction, prevention and recovery*:
  Improving the error detection coverage.
- *Thermal modelling technique*:
  Reducing thermal gradients in the system.
- *Timing analysis and enforcement methodology*:
  Achieving zero deadline misses for critical applications under the targeted failure rates.
- *Runtime manager integration with hybrid programming models*:
  Decreasing resource management integration effort in hybrid programming models.
- *Hardware abstraction layer for resource disaggregation*:
  Increasing the effective usage of available resources under non-saturated conditions; Identifying optimal matches of application requests and available resources.

## II. Background and open issues

This section describes the current technological background which is directly relevant for RECIPE. This spans several areas including run-time management, heterogeneous computing architectures, HPC memory/interconnection infrastructures, thermal modelling, reliability, programming models, and timing analysis.

### A. Runtime resource management and reliability

The problem of unexpected hardware failures is rapidly growing, due to the increasing thermal stress sensitivity of hardware components in modern HPC systems. For some applications, reliability issues may become critical, because faults lead to silent data corruptions passing undetected by the hardware, or otherwise because mere re-execution upon a crash may lead to violating the expected execution time. In other words, for time-aware applications, the occurrence of some runtime error may lead to deadline misses (timing violations), even when appropriate error recovery mechanisms, e.g. checkpointing, are put in place. In the latest years, several *Dynamic Reliability Management* solutions have been introduced [11], [12], [13], [14]. Most operate on a *reactive* basis, by exploiting checkpoint-restore protocols, to resume corrupted data or faulty application executions. *Proactive* approaches, based on suitable predictive models, have been recently proposed. However, such models are often platform-specific and based on single-node computing systems. Furthermore, they do not usually consider the application timing requirements. The goal is typically to maximize the lifetime of the hardware components.

### B. Programming models

HPC applications are usually programmed by using *hybrid* programming models, combining a distributed memory model, often MPI, to manage the application parallelism at global level, and a shared memory model, to manage parallelism at the single-node level. In heterogeneous systems, the node-level programming model needs specialized support for heterogeneity, such as those provided by OpenCL or CUDA. In such heterogeneous programming models, the programmer is in charge of managing explicitly the required resources, which demands a large amount of boilerplate code. When resource management is available, it is generally placed at the top level (in the application dispatcher) or it operates under the assumption that an entire node is allocated to a single application. Such coarse approaches pose an inherent limitation, that will be addressed by RECIPE to enable application-driven management strategies.

### C. Heterogeneity

Current trends in HPC are increasingly moving towards heterogeneous platforms, i.e. systems made of different computational units, ranging from general-purpose processors to graphics processing units (GPUs) and even special-purpose units made of custom acceleration logic, often implemented on field-programmable gate arrays (FPGAs) [15], [16], [17], [18]. In fact, several players have recently introduced FPGA-based heterogeneous platforms used in a large range of high-performance computing applications [19], e.g. multimedia, machine learning, bioinformatics, etc. [20], [19], with speedups in the range of 10x to 100x. However, the effective exploitation of combined heterogeneous resources still poses an open issue in current HPC, and has only been demonstrated for very specific workloads.

### D. HPC interconnect

Current HPC interconnect technologies include Ethernet [21], InfiniBand [22], as well as vendor specific interconnects, particularly Intel Omni-Path technology [23].

Ethernet, a dominant standard for mainstream commercial computing requirements, has continued to evolve, reaching performance levels of 400 Gbps in 2017. InfiniBand is designed for scalability, relying on a switched fabric network topology together with remote direct memory access (RDMA) to reduce CPU overhead which enables maintaining a performance and latency edge in comparison to Ethernet in many high performance workloads. The InfiniBand roadmap [24] details bandwidths reaching 600Gb/s data rate HDR in the middle of 2018 and 1.2Tb/s data rate NDR in 2020. Introduced in 2015, Intel's end-to-end Omni-Path Architecture (OPA) claims higher messaging rates and lower latency than InfiniBand, in addition to advanced features such as traffic flow optimization, packet integrity protection and dynamic lane scaling, although it is not currently a recognized standard. In RECIPE, InfiniBand will be the HPC interconnect of choice, because of the higher performance versus Ethernet as well as its role as a de-facto standard, unlike Intel OPA. However, the project will need to address crucial issues related to the interplay with the deeply heterogeneous acceleration fabric, for which no solutions exist exposing direct access to HPC interconnect technologies, as well as the interplay between the QoS mechanisms provided at RTMS and interconnect level.

### *E. Thermal modelling*

Efficient thermal management requires accurate knowledge about the thermal profile of the chip in both steady and transient states. In conventional Multi-Processor Systems on Chip (MPSoCs), the knowledge of the chip floorplan allows the design-time identification of the hot-spot locations. However, this approach is no longer possible in heterogeneous MPSoCs equipped with a reconfigurable fabric, as the thermal distribution depends on the accelerators implemented, which are unknown during the design phase. As a result, energy efficiency comes at the cost of shifting thermal evaluation from the chip design phase to the run-time management, affecting also runtime reliability [25]. This may be particularly relevant to RECIPE, because of the adoption of FPGA-based accelerators. The project will thus aim at leveraging accurate, fast, and flexible thermal simulation to demonstrate novel efficient thermal management strategies for heterogeneous HPC platforms.

### *F. Timing analysis*

The large amount of computation and storage nodes integrated in HPC systems makes applications suffer from considerable performance variability, which will inevitably increase with the advent of exascale HPC systems. The uncertainty in the timing behavior of HPC applications is a consequence of either the internals of the application (intrinsic variability) or the interactions between different applications or with the system itself (extrinsic variability). Timing variability makes it difficult to achieve tight execution time bounds for the applications running in the computing infrastructure [26]. For applications with tight real-time requirements, like those addressed by RECIPE, standard practices of performance analysis are not enough and new methods are required to derive trustworthy and tight upperbounds to application execution times [27] [28].

## III. Key innovations in RECIPE

Based on the above technological background, we identify the key technology innovations that will be introduced by RECIPE to reach the ultimate goals set by the project.

### *A. Hierarchical runtime management*

RECIPE aims at addressing the reliability problem by operating both at the node and at the infrastructure level, thanks to a *hierarchical* resource management approach. Our objective is to integrate reactive and proactive solutions, while maximizing the lifetime of the system, as well as providing reliability guarantees to time-sensitive applications. For the latter, we need to minimize the use of reactive mechanisms and maximize the effectiveness of predictive resource management, thus supporting the combined optimization of reliability and timing.

Based on this approach, the RECIPE runtime management system will be partitioned in two layers, a *global* resource manager providing proactive fault tolerance and coarse-grained resource allocation, as well as a *local* resource manager, ensuring reactive fault tolerance and fine-grained resource allocation. In particular, RECIPE will build on the integration of BarbequeRTRM [29] and the SLURM-based Global Resource Management (GRM) framework. These will cooperate on a hierarchical basis, with GRM being in charge of dispatching the workload at the global infrastructure level, and BarbequeRTRM mapping the application's tasks and memory requests at the local node level. Each instance of BarbequeRTRM (one per node) will take into account the timing requirements of the application dispatched by the GRM layer, the current status of the node, i.e. load and power consumption when available, and the performance profile of the available (heterogeneous) computing resources.

As specific technical actions planned by RECIPE, we will extend the BarbequeRTRM capabilities by: 1) feeding the policies with data coming from suitable models, in order to implement reliability-aware resource allocation policies; 2) providing a local checkpoint-restore mechanism, to resume the execution of application's tasks by migrating them across computing resources. On the other hand, the GRM capabilities will also be extended by: 1) adding a proactive reliability module that will communicate with the 3D-ICE [30] temperature modelling module, enabling reliability modelling and prediction; 2) proposing new reliability-aware runtime global management strategies; and 3) enhancing the

capabilities of distributed computing of the GRM, enabling the application execution to span across multiple nodes.

In order to support the above predictive strategies, RECIPE will develop novel models integrated with run-time management policies, running and cooperating at both the node and the infrastructure level. Furthermore, to assess the impact of reliability and thermal issues on the overall system architecture, we will also rely on simulation, which will be possible thanks to the recent gem5-X framework developed by EPFL. In RECIPE, gem5-X [31] will be equipped with both thermal and reliability simulation to evaluate the impact of architectural choices and resource management strategies in short and long term reliability. We will investigate state-of-the-art techniques in fault prediction technologies [32]. Existing fault-prediction techniques are generally based on the existence of a model of the system that has the ability to predict the occurrence of faults when the real-time performance, power, and temperature measurements of the different system components indicate that faulty situations are likely to occur. These models can be built by using fault statistics of systems based on similar technologies, when available, or by using information provided by the component manufacturer which allows creating simulation models of the system. When both approaches are combined, synthetic models can be enriched or calibrated at runtime with information collected during the system lifetime. Given that the system temperature is a very important contributor to the component reliability, fault prediction models will also incorporate the thermal behavior. The fault/performance prediction, based on application statistics, will then drive the runtime resource manager during scheduling and task mapping in order to minimize missed deadlines and adapt the checkpoint rates to various fault recovery strategies on a given HPC system configuration.

### *B. Interplay with programming models*

As highlighted above, most RTMS-level choices will be driven by application-specific QoS requirements and behavior profiles. This is particularly important because RECIPE envisions scenarios where multiple unrelated applications will co-exist on the same nodes, thus requiring system-wide resource management in order to maximize utilization of (possibly heterogeneous) compute resources. To support resource management at the node level, the application needs to be able to interact with the resource manager in a way that is as much transparent as possible to the programmer, so as to provide the resource manager with the appropriate information about the required Quality of Service and the type of resources that the application can effectively exploit. The interface between the runtime management system and the application in RECIPE will be mediated by the programming model, supporting the expression of such application expectations. Existing experimental models, however, do not fully match the requirements of typical industry-grade implementations. In particular, there is not support for dynamic compilation, nor for any source language except C/C++. In RECIPE, we will build on the *mangolib* model and implementation developed in the MANGO FETHPC project [5] to integrate system-wide resource management within the programming model, and will extend it to reach industry-grade maturity. The current *mangolib* model allows the application to specify a different "recipe" for each kernel, as well as multiple implementations of the same kernel, targeting different accelerators. The runtime manager then automatically manages the selection of the best configuration according to the application priority level, quality of service requirements, and resource availability, while optimizing system-wide metrics of utilization and energy-efficiency. We will further extend the model to support multiple languages, depending on application requirements. Candidate extensions include improved compatibility with the OpenCL programming model and the FORTRAN programming language [33]. Furthermore, we will provide dynamic compilation capabilities to allow just-in-time compilation of kernels from the source code.

### *C. Support for deep heterogeneity*

The RECIPE hardware concept will build on the outcome of the MANGO H2020 project [5]. While the host-side part of the prototype will include commercial architectures like Intel Xeon [34], [35] as well as GPUs [36], heterogeneous acceleration will rely on the MANGO Field Programmable Gate Array (FPGA) fabric [17]. The MANGO prototype, originally aimed at manycore architecture exploration, contains two main components, the General-purpose Node (GN), which includes commercial CPUs and GPUs, and the Heterogeneous Node (HN), made of customized motherboards mounting dedicated FPGA daughterboards developed by ProDesign Gmbh. HNs offer a large degree of flexibility for exploring and instantiating customized accelerators and various heterogeneous system configurations. Among other features, the key architectural aspects of MANGO that will be inherited in RECIPE include the communication infrastructure between the GN and HN (as well as within the HN), synchronization mechanisms between the host and the hardware application kernels, QoS guarantees and HN interconnect partitionability (used for QoS in MANGO but potentially extended in RECIPE for fault tolerance), burst transfers, unified HN memory address space, and a basic HN hardware abstraction library.

An important challenge in RECIPE relates to the increased scale of the target HPC system. While MANGO focuses on the concept of an HN node made of several interconnected FPGAs, RECIPE targets a more ambitious scenario where several HN nodes coexist in the system, each defining its own set of accelerators. While the HNs are physically distributed throughout the HPC system, they will be part of a single pool of accelerators. As an example, in

MANGO an application can obtain a set of resources, like memory or accelerators, only from its corresponding HN node. In RECIPE we will extend the application capabilities to use accelerators from different, efficiently interconnected HN nodes at the same time.

In order to fully match the potential of the RTMS, the RECIPE heterogeneous accelerators will also expose hardware monitors and a suitable hardware abstraction layer. The hardware abstraction layer will provide a uniform view of the available accelerators, including GPU, manycore, and FPGA devices, exposing fine-grain partitioning mechanisms used to ensure scalability, system-wide reconfigurability, and dedicated resource provisioning.

The project will also need to address important challenges related to the integration of large-scale reconfigurable hardware in user applications. In particular, for the development of its heterogeneous abstraction layer, RECIPE will explore four different styles of use for FPGA acceleration, depending on the specific use case requirements:

- full custom HDL implementation, suitable for performance-critical, relatively simple and regular kernels;
- optimized library-based design, for well-supported kernels to be implemented in hardware, e.g. linear algebra;
- pure-hardware HLS-based design, for non-standard kernels or control-intensive parts of the application that are not performance-critical;
- software-programmed accelerators, particularly the NaplesPU vector core [37] and associated LLVM-based compiler imported from MANGO, which is suitable for control-intensive parts of the software application that do not match the restrictions of HLS and/or data-intensive kernels benefitting from custom vector-style approaches.

### D. Interconnect configurability

Similar to the multi-accelerator compute platform, the memory/interconnect architecture will also need to scale up to the system level. The MANGO solution allows some basic bandwidth allocation strategies *within* the HN, in that HN accelerators and the connected server can obtain reserved interconnect bandwidth and memory resources, although this feature is limited to the applications running on the particular server connected to the HN. These capabilities will be expanded from the node level to the system level in RECIPE, totally decoupling memory and interconnect management from the node boundary and exposing such functionality to the complete system. Indeed, one important aspect will be the interoperation of different interconnects in different layers of the system. In particular, the network within the HNs will be defined so as to match the features of the system-level interconnect, i.e. InfiniBand. In fact, we plan to augment the MANGO HN with a standard InfiniBand adapter, plugged into the host system, which in turn will allow the reconfigurable accelerator fabric within the HN to directly access other nodes' adapters. As seen from the developer's perspective, the host-side portion of the application will be exposed with a software API allowing basic communication/synchronization mechanisms. On the other hand, the FPGA-side portion of the application will see a hardware interface (called "hardware middleware" in RECIPE), acting as the FPGA counterpart of the communication/synchronization primitives. This setup will allow users to develop heterogeneous applications where "hardware kernels" are physically scattered across a multitude of nodes while acting as a single accelerator fabric. This scenario will mimic the disaggregation concept already explored in datacenters for CPUs and GPU devices, and will allow hardware/software applications to transparently rely on the features of the underlying HPC interconnect.

Unlike QoS, reliability-related mechanisms are not directly supported by the MANGO technology. However, MANGO allows multiple accelerator *replicas* to be instantiated and used on the same HN or on different HNs. This feature paves the way to new flexible mechanisms for enhancing fault-tolerance in RECIPE. In the project we will provide node-level (within the HN node) and system-level (within the Resource Manager) reliability mechanisms able to detect and isolate malfunctioning devices located in the HN node, then exploiting the underlying reconfigurability to enforce proactive and reactive fault-tolerance strategies.

### E. Proactive/reactive thermal management

To enable effective thermal management, RECIPE will develop a holistic simulation tool for a wide range of cooling techniques and settings, building on 3D-ICE and using the most recent pluggable heatsink models that enable simulation of arbitrary cooling models [38], and using the gem5-X framework [39]. To do so, RECIPE will follow a trace-based methodology, that will put together new cooling models validated using real traces obtained from an infrared high-precision microscope. The goal is to be able to accurately simulate the cooling systems available in the RECIPE prototype, that will for sure include air-cooling, but will also leverage the usage of the passive two-phase thermosyphon liquid-cooling system developed during the MANGO project by EPFL. Moreover, we will tackle the incorporation of reliability models into thermal simulation.

To allow the creation of highly-accurate transient models for heterogeneous MPSoCs under arbitrary cooling conditions in a scalable way, EPFL will leverage its expertise on the acceleration of thermal modelling in two ways:

- the development of fast neural-network and machine learning based methods able to execute at runtime, and
- the acceleration of the modelling infrastructures.

In this way, thermal modelling and control within the RECIPE project will allow proactive (prediction-based) and reactive (emergency-based) thermal management to reduce

hot-spots and maintain temperature gradients within the 5-degree limit, enhancing lifetime reliability. To achieve this goal, we will propose reliability-aware workload management techniques that come from the embedded world, scaling and porting them to the heterogeneous HPC domain.

### F. Characterization of worst-case application performance

RECIPE will develop new methodologies to improve the understanding of both intrinsic and extrinsic variability in HPC applications, as a fundamental tool driving the choices made by the RTMS. To characterize intrinsic variability, we will explore the utilization of memory layout randomization techniques. Layout randomization has been proposed as an effective method to perform statistically sound performance analysis of both real-time and high-performance applications [40], [41]. Memory layout randomization helps cover the intrinsic variability (cache, stateful resources, and other processor internals) and allows the application of statistical techniques to reason about the observed timing variability. In RECIPE we will explore for the first time how to deploy memory layout randomization techniques in an HPC system. We will also analyze to what extent memory layout randomization is useful to cover the variability related with the system itself (background network traffic, job scheduling, interrupts, etc.).

To cover the extrinsic variability, the one contributed by applications co-hosted with the application under analysis or system management tasks, we will use resource stressing kernels to develop contention models of the HPC system. The contention models will rely on both the characteristics of the application requiring timing guarantees and the estimation of the interference that co-running applications will cause to the application under analysis. Based on the application characterization developed in the RECIPE project, a detailed analysis of the mutual influence of applications will be performed. A mathematical model will be created that will simulate complex dependencies between co-running applications and hardware. This data will be used to automatically schedule the applications to the available resources, in order to maximize their performance and minimize the overall energy consumption. The deadlines of the applications will be also taken into account for ensuring timing guarantees to user applications.

For the timing analysis we will employ existing tools like Extrae [42] or VTune [43] (for Xeon architectures) to gather information on application behavior from the underlying hardware. These tools require instrumenting the application under analysis to collect statistics about the utilization of the different resources by reading predefined monitoring counters. In the performance metering process of HPC applications, apart from the end-to-end execution time measurements, we will also collect other fine-grain events to characterize how the applications utilize the different resources. The fine-grain measurements that are usually helpful to detect performance bottlenecks will be used to reason about potential unobserved behaviors of the application and determine appropriate safety margins to the maximum observed execution times.

### G. Use-case applications

RECIPE identified a number of real-world high performance applications demonstrating the emerging class of requirements addressed by the project in terms of heterogeneity, fault-tolerance, and time constraints. In particular, the RECIPE use cases include applications in the areas of weather forecasting, subsoil properties identification, and bio-medical big-data applications. The ultimate objective in RECIPE is to perform a demonstration on both industry-grade, pre-Exascale systems and on emerging deeply heterogeneous technologies, in order to measure the degree of success of the project.

## IV. Conclusions

This paper presented an overview of the recently started RECIPE H2020 FETHPC project. We first identified the technological challenges for deeply heterogeneous HPC, related to Quality of Service and proactive/reactive reliability mechanisms, that will be addressed by RECIPE. We then described the key innovations that RECIPE will bring in relevant areas, spanning run-time management, heterogeneous computing architectures, HPC memory/interconnection infrastructures, thermal modelling, reliability, programming models, and timing analysis. Directly answering crucial challenges in next-generation HPC, RECIPE will introduce important enabling technologies for the forthcoming exascale scenarios.

## Acknowledgement

The activities described in this article received funding from the European Union's Horizon 2020 research and innovation programme under the FETHPC grant agreement no. 801137 RECIPE: REliable power and time-ConstraInts-aware Predictive management of heterogeneous Exascale systems.

## References

[1] J.-P. Panziera *et al.*, "Strategic Research Agenda 2017," ETP4HPC, Tech. Rep., Nov 2017.

[2] B. Koller *et al.*, "Towards an environment to deliver high performance computing to small and medium enterprises," in *Sustained Simulation Performance 2015*. Cham: Springer International Publishing, 2015, pp. 41–50.

[3] G. Agosta, W. Fornaciari, G. Massari, A. Pupykina, F. Reghenzani, and M. Zanella, "Managing Heterogeneous Resources in HPC Systems," in *Proc. of PARMA-DITAM '18*. ACM, 2018, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/3183767.3183769

[4] J. Flich *et al.*, "Enabling HPC for QoS-sensitive applications: The MANGO approach," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 702–707.

[5] ——, "Exploring manycore architectures for next-generation HPC systems through the MANGO approach," *Microprocessors and Microsystems*, vol. 61, pp. 154 – 170, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0141933118300243

[6] G. Massari, A. Pupykina, G. Agosta, and W. Fornaciari, "Predictive resource management for next-generation high-performance computing heterogeneous platforms," in *Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS'19)*, Jul 2019.

[7] A. Pupykina and G. Agosta, "Optimizing Memory Management in Deeply Heterogeneous HPC Accelerators," in *2017 46th Int'l Conf on Parallel Processing Workshops (ICPPW)*, Aug 2017, pp. 291–300.

[8] J. Flich *et al.*, "The MANGO FET-HPC Project: An overview," in *IEEE 18th Int'l Conf on Computational Science and Engineering (CSE)*. IEEE, 2015, pp. 351–354.

[9] M. Zanella, G. Massari, A. Galimberti, and W. Fornaciari, "Back to the future: Resource management in post-cloud solutions," in *Proceedings of the Workshop on INTelligent Embedded Systems Architectures and Applications*, ser. INTESA '18. New York, NY, USA: ACM, 2018, pp. 33–38. [Online]. Available: http://doi.acm.org/10.1145/3285017.3285028

[10] W. Fornaciari, G. Agosta, D. Atienza, C. Brandolese, L. Cammoun, L. Cremona, A. Cilardo, A. Farres, J. Flich, C. Hernandez, M. Kulchewski, S. Libutti, J. M. Martínez, G. Massari, A. Oleksiak, A. Pupykina, F. Reghenzani, R. Tornero, M. Zanella, M. Zapater, and D. Zoni, "Reliable power and time-constraints-aware predictive management of heterogeneous exascale systems," in *Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation*, ser. SAMOS '18. New York, NY, USA: ACM, 2018, pp. 187–194. [Online]. Available: http://doi.acm.org/10.1145/3229631.3239368

[11] C. L. Chou and R. Marculescu, "FARM: Fault-aware resource management in NoC-based multiprocessor platforms," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.

[12] M. H. Haghbayan, A. Miele, A. M. Rahmani, P. Liljeberg, and H. Tenhunen, "A lifetime-aware runtime mapping approach for many-core systems in the dark silicon era," in *2016 Design, Automation Test in Europe Conference (DATE)*, March 2016, pp. 854–857.

[13] L. Huang and Q. Xu, "Characterizing the lifetime reliability of manycore processors with core-level redundancy," in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2010, pp. 680–685.

[14] P. Mercati, F. Paterna, A. Bartolini, L. Benini, and T. Rosing, "WARM: Workload-aware reliability management in Linux/Android," *IEEE Trans on CAD of Integrated Circuits and Systems*, 2016.

[15] "https://euroexa.eu."

[16] "https://www.altera.com/products/sip/memory/stratix-10-mx/overview.html."

[17] "http://www.mango-project.eu."

[18] A. Cilardo and L. Gallo, "Interplay of loop unrolling and multidimensional memory partitioning in HLS," in *Proceedings of the 2015 Design, Automation and Test in Europe Conference and Exhibition*, ser. DATE '15. San Jose, CA, USA: EDA Consortium, 2015, pp. 163–168.

[19] K. Paranjape, S. Hebert, and B. Masson, "Heterogeneous computing in the cloud: Crunching big data and democratizing HPC access for the life sciences," Intel Corporation, Tech. Rep., 2010.

[20] S. Sarkar, T. Majumder, A. Kalyanaraman, and P. Pande, "Hardware accelerators for biocomputing: A survey," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, May 2010, pp. 3789–3792.

[21] "IEEE standard for Ethernet - Amendment 10," *IEEE Std 802.3bs-2017*, pp. 1–372, Dec 2017.

[22] G. F. Pfister, "Aspects of the InfiniBand architecture," in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, Oct 2001, pp. 369–371.

[23] M. S. Birrittella, M. Debbage, R. Huggahalli, J. Kunz, T. Lovett, T. Rimmer, K. D. Underwood, and R. C. Zak, "Intel Omni-Path architecture: Enabling scalable, high performance fabrics," in *2015 IEEE 23rd Annual Symposium on High-Performance Interconnects*, Aug 2015, pp. 1–9.

[24] "https://www.infinibandta.org/infiniband-roadmap/."

[25] D. Zoni, L. Cremona, and W. Fornaciari, "All-digital energy-constrained controller for general-purpose accelerators and CPUs," *IEEE Embedded Systems Letters*, 2019, doi:10.1109/LES.2019.2914136.

[26] F. Reghenzani, G. Massari, W. Fornaciari, and A. Galimberti, "Probabilistic-WCET reliability: On the experimental validation of EVT hypotheses," in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, ser. COINS '19. New York, NY, USA: ACM, 2019, pp. 229–234. [Online]. Available: http://doi.acm.org/10.1145/3312614.3312660

[27] F. Reghenzani, G. Massari, and W. Fornaciari, "chronovise: Measurement-based probabilistic timing analysis framework," *J. Open Source Software*, vol. 3, no. 28, p. 711, 2018.

[28] F. Reghenzani, G. Massari, and W. Fornaciari, "The misconception of exponential tail upper-bounding in probabilistic real-time," *IEEE Embedded Systems Letters*, 2018, doi:10.1109/LES.2018.2889114.

[29] P. Bellasi, G. Massari, and W. Fornaciari, "Effective runtime resource management using linux control groups with the BarbequeRTRM framework," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 2, pp. 39:1–39:17, Mar. 2015. [Online]. Available: http://doi.acm.org/10.1145/2658990

[30] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza, "3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling," in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2010, pp. 463–470.

[31] Y. M. Qureshi, W. A. Simon, M. Zapater Sancho, K. Olcoz, and D. Atienza Alonso, "Gem5-X: A Gem5-based system level simulation framework to optimize many-core platforms," *Proceedings of the 27th High Performance Computing Symposium (HPC 2019)*, p. 12, 2019.

[32] A. Gainaru, F. Cappello, M. Snir, and W. Kramer, "Fault prediction under the microscope: A closer look into HPC systems," in *SC '12: Proc. of the Int. Conference on High Performance Computing, Networking, Storage and Analysis*, Nov 2012, pp. 1–11.

[33] G. Massari, C. Caffarri, P. Bellasi, and W. Fornaciari, "Extending a run-time resource management framework to support OpenCL and heterogeneous systems," in *Proceedings of Workshop on Parallel Programming and Run-Time Management Techniques for Many-core Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms*, ser. PARMA-DITAM '14. New York, NY, USA: ACM, 2014, pp. 21:21–21:26. [Online]. Available: http://doi.acm.org/10.1145/2556863.2556868

[34] "https://lanl.gov/projects/trinity/specifications.php."

[35] "https://www.bsc.es/marenostrum/marenostrum/technical-information."

[36] "https://www.olcf.ornl.gov/olcf-resources/compute-systems/titan/."

[37] "The Naples Processing Unit (NaplesPU)," http://www.NaplesPU.org, accessed: 2019-06-24.

[38] A. Iranfar, F. Terraneo, W. A. Simon, L. Dragic, I. Pilji, M. Zapater Sancho, W. Fornaciari, M. Kovac, and D. Atienza Alonso, "Thermal characterization of next-generation workloads on heterogeneous MPSoCs," 2017.

[39] D. Zoni and W. Fornaciari, "Modeling DVFS and power-gating actuators for cycle-accurate NoC-based simulators," *J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 3, pp. 27:1–27:24, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2751561

[40] C. Curtsinger and E. D. Berger, "STABILIZER: Statistically sound performance evaluation," *SIGARCH Comput. Archit. News*, vol. 41, no. 1, pp. 219–228, Mar. 2013. [Online]. Available: http://doi.acm.org/10.1145/2490301.2451141

[41] L. Kosmidis, C. Curtsinger, E. Quiones, J. Abella, E. Berger, and F. J. Cazorla, "Probabilistic timing analysis on conventional cache designs," in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2013, pp. 603–606.

[42] "https://tools.bsc.es/extrae."

[43] "https://software.intel.com/en-us/vtune."