Enabling HPC for QoS-sensitive applications: the MANGO approach

José Flich*, Giovanni Agosta†, Philipp Ampletzer‡, David Atienza Alonso§, Carlo Brandolese†,
Alessandro Cilardo¶, William Fornaciari†, Ynse Hoornenborg††, Mario Kovač‖, Bruno Maitre‡‡, Giuseppe Massari†,
Hrvoje Mlinarić‖, Ermis Papastefanakis‡‡, Fabrice Roudet**, Rafael Tornero*, Davide Zoni†
*Universitat Politècnica de València, †DEIB – Politecnico di Milano, ‡Pro Design GmbH,
§ESL – École Politechnique Fédérale de Lausanne (EPFL), ¶Centro Regionale Information Communication Technology SCRL,
‖University of Zagreb, **Eaton Industries SAS, ††Philips Medical Systems, ‡‡Thales Communications & Security
Email: *jflich@disca.upv.es, †name.surname@polimi.it, ‡philipp.ampletzer@prodesign-europe.com,
§david.atienza@epfl.ch, ¶acilardo@unina.it, ‖name.surname@fer.hr, **FabriceRoudet@eaton.com,
††ynse.hoornenborg@philips.com, ‡‡name.surname@thalesgroup.com

***Abstract*—In this paper, we provide an overview of the MANGO project and its goal. The MANGO project aims at addressing power, performance and predictability (the PPP space) in future High-Performance Computing systems. It starts from the fundamental intuition that effective techniques for all three goals ultimately rely on customization to adapt the computing resources to reach the desired Quality of Service (QoS). From this starting point, MANGO will explore different but interrelated mechanisms at various architectural levels, as well as at the level of the system software. In particular, to explore a new positioning across the PPP space, MANGO will investigate system-wide, holistic, proactive thermal and power management aimed at extreme-scale energy efficiency.**

***Keywords*—High Performance Computing, Customization, Energy Efficiency**

I. Introduction

High-Performance Computing (HPC) as we know it today is experiencing unprecedented changes, encompassing all levels from technology to use cases. On one hand, the escalating quest for performance/power efficiency is increasingly requiring deep application-based customization of the underlying computing architecture. On the other hand, new delivery models, such as outsourced and cloud-based HPC, are dramatically widening the amount and the type of HPC demand, posing both promising opportunities and big challenges for future HPC. In fact, cloud enables resource usage and business model flexibility, but it also deeply impacts architecture, as platforms must inherently support virtualization and be ready for large-scale capacity computing, serving many unrelated, competing applications with different workloads.

Moreover, HPC is increasingly faced with a QoS-sensitive computing demand, coming from that particular class of applications whose correctness depends on both performance and timing requirements, and the failure to meet either of them is critical. Examples of such time-critical applications include financial analytics, online video transcoding, and medical imaging. Such applications require some form of time predictability, in addition to performance and power efficiency. Time-predictability and QoS are relatively unexplored areas in HPC – traditional HPC focuses on throughput and resource usage optimization, in a trade-off with power-efficiency requirements. Extending the traditional optimization space, MANGO aims at addressing what we call the PPP space: *power*, *performance*, and *predictability*. In fact, predictability, power, and performance appear to be three inherently diverging perspectives on HPC.

In this scenario, the essential objective of MANGO is to achieve extreme resource efficiency in future QoS-sensitive HPC through ambitious cross-boundary architecture exploration. The research will investigate the architectural implications of the emerging requirements of HPC applications, aiming at the definition of new generation high-performance, power-efficient, deeply heterogeneous architectures with native mechanisms for isolation and QoS. MANGO will follow a disruptive approach challenging several basic assumptions, exploring new manycore architectures specifically targeted at HPC.

A. The MANGO Approach

The performance/power efficiency wall poses the major challenge faced nowadays by HPC. Looking straight at the heart of the problem, the hurdle to the full exploitation of today computing technologies ultimately lies in the gap between the applications' demand and the underlying computing architecture: the closer the computing system matches the structure of the application, the most efficiently the available computing power is exploited. Consequently, enabling a deeper customization of architectures to applications is the main pathway towards computation power efficiency. Theoretically, customization can enable improvements in power efficiency as high as two orders of magnitude, since it enables the computing platform to approximate the ideal intrinsic computational efficiency (ICE), defined as the energy consumption per operation achieved by purely computation circuits, e.g. FP adders.

The current uncertainty regarding on-chip HPC solutions and the essentially open nature of current architecture-level research will be regarded by MANGO as an opportunity, rather than a limitation. The fundamental intuition behind the project is that effective techniques for both performance/power efficiency and predictability ultimately share a common underlying mechanism, i.e., some form of fine-grained adaptation, or customization, used to tailor and/or reserve computing resources only driven by the application requirements. Along this path, the project will involve many different, and deeply interrelated, mechanisms at various architectural levels, from the heterogeneous computing cores, up to the memory architecture, the interconnect, the runtime resource management, power monitoring and cooling, also evaluating the implications on programming models and compilation techniques. In particular, to explore a new positioning across the PPP space, MANGO will investigate system-wide, holistic proactive thermal and power management aimed at extreme-scale energy efficiency by creating a hitherto inexistent link between hardware and software effects, which will involve all layers of an HPC system, from server to rack, to datacenter. The combined interplay of the multi-level innovative solutions brought by MANGO will result in a new positioning in the PPP space, ensuring sustainable performance as high as 100 PFLOPS for the realistic levels of power

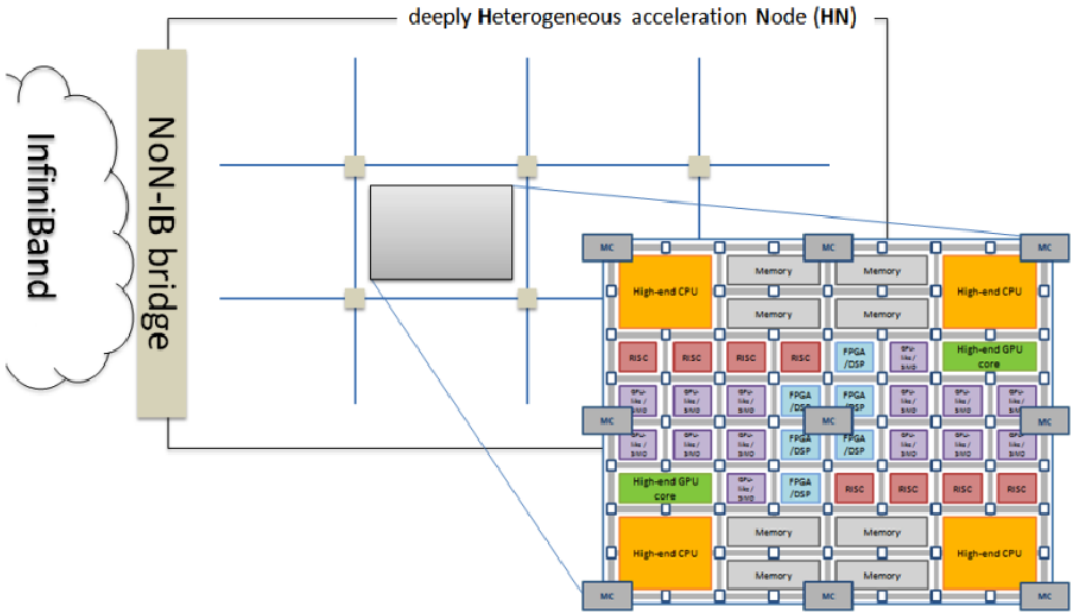

Fig. 1. MANGO Hardware Architecture

consumption ($<$ 15MWatt) delivered to QoS-sensitive applications in large-scale capacity computing scenarios. MANGO will provide essential building blocks at the architectural level enabling the full realization of the long-term objectives foreseen by the ETP4HPC strategic research agenda [1].

B. Organization of the paper

The rest of the paper is organized as follows. In Section II we describe the targeted MANGO architecture. Then, in Section III we describe the programming models and runtime management resources to be used in MANGO and in Section IV the thermal and cooling innovations proposed in the project. Section V shows the prototyping roadmap for MANGO whereas Section VI briefly describes the application scenarios. Finally, the paper finishes in Section VII with some conclusions.

II. Hardware Architecture Concept

At the architecture level, the MANGO project foresees a scenario where General-purpose compute Nodes (GNs), hosting commercial-off-the-shelf solutions (e.g. Intel Xeon Phi processors or high-end NVIDIA GPU accelerators), coexist with *Heterogeneous Nodes* (HNs), forming a common HPC infrastructure. HNs, as depicted in Figure 1, will essentially be on-node clusters of next-generation manycore chips coupled with deeply customized heterogeneous computing resources. The manycore architecture will be open, it will not rely on COTS solutions available today, but rather it will enable broad-spectrum, ground-breaking research in the area of on-/off-chip architecture. Building on recent trends in HPC research, in fact, HNs will allow borrowing solutions from the embedded/System-on-Chip domain, which is now recognized as a promising pathway to extreme-scale low-power HPC. HNs will contain a multi-chip mesh of power-efficient RISC cores augmented with custom vector resources (SIMD and lightweight GPU-like cores) as well as a dedicated memory architecture and a custom Network-on-Chip providing advanced support for partitionability and time-predictability. The cores in the multi-chip manycore architecture will be connected through a *Network-on-Node* (NoN), forming a continuum at the off-chip (on-node) level from the on-chip interconnect.

Since the first stages of the project, the architecture exploration will be extensively supported by a purposely developed emulation platform. HNs will not be prototyped in a final ASIC form, but a mixed approach will be taken. In fact, RISC processors will be instantiated as ASIC cores tightly coupled with a large-scale reconfigurable hardware fabric used to emulate in near real-time the customized acceleration units, the advanced memory management architecture and the NoN, as well as the NoN bridge to the external interconnect. The platform will support fast design space exploration and validation of the solutions at both the software- and thermal/power-level. These techniques will inherently involve multiple aspects within the system, from programming down to the architecture definition, deeply intertwined with chip- and system-wide control mechanisms of physical parameters, primarily power consumption and temperature. To gain a holistic understanding of their impact on performance/power/predictability (PPP) and quantitative information about their effectiveness, MANGO will also develop a comprehensive toolset for PPP and thermal models, which will operate in close relation with the PPP run-time information collected from the platform.

The MANGO experimental platform will include 16 GN nodes with standard high-end processors, i.e. Intel Xeon E5, as well as NVIDIA Kepler GPUs, along with 64 HN nodes. GNs and HNs will be connected through InfiniBand. HNs will contain ASIC ARM cores and a high-capacity cluster of FPGAs used to emulate the rest of the HN system. The final HN infrastructure will contain dozens of manycore chips, and thus thousands of cores. The prototypical board will enable components to be easily plugged and removed and will allow different resource mixes, e.g. nodes highly populated with ARM cores and few high-end FPGAs (e.g. 192 + 64) or vice versa (e.g. 64 + 192), plus memory modules.

III. Programming Model and Runtime Management

To reach exascale parallelism, the programming model needs to be hierarchical, much like the runtime management system. Tradi-

tionally, the programming model for homogeneous HPC systems is based on a combination of MPI and OpenMP. When heterogeneity comes into the game, the programming model needs to be extended to allow the exploitation of hardware resources. OpenCL is an open standard for the development of parallel applications on a variety of heterogeneous multi-core architectures [2]. It provides explicit management of heterogeneity, but at a significant cost in terms of tuning performance, which must be performed by the programmer, and building boilerplate code [3], [4]. In MANGO, we aim at integrating the expression of new architectural features as well as QoS concerns and parameters within the existing stack of languages and libraries for extreme-scale HPC systems, by augmenting the runtime library APIs with new functions, as well as by introducing new pragmas or keywords to the language.

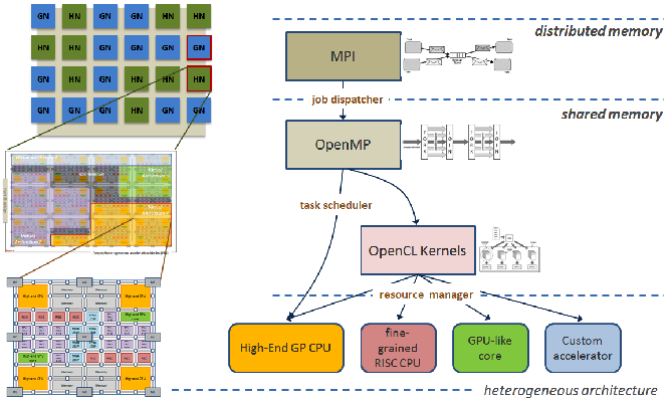

Fig. 2. MANGO Software Stack

Figure 2 shows the MANGO programming model stack and its interaction with the underlying architecture and runtime software components. The programming models employ MPI to express inter-node computation, while at the node level OpenMP will be used to allow the expression of irregular applications, and OpenCL will serve as an intermediate language behind OpenMP, allowing the construction of virtual accelerators on the fly, collecting compatible cores not already allocated. The experience of the 2PARMA project [5], [6] will help in this regard. The programming model will be integrated with the runtime management facilities, allowing job dispatcher, task manager, and virtual device manager to interact with the corresponding three levels of the programming model. Promising examples of the effectiveness of this approach have been already shown in [7].

The fine-grained configuration mechanisms exposed by the MANGO deeply heterogeneous architecture will however pose new challenges for optimizing the performance and time-predictability of accelerated kernels. Building on previous results related to custom hardware-accelerated systems in the context of the syMParallel experimental toolchain and the HtComp project [8], [9], [10], MANGO will address the optimization of statically-predictable kernels based on the polyhedral model [11]. The activity will follow two different paths: a) characterizing kernels in isolation, inferring and controlling through suitable code-level transformations the patterns within the application that are relevant to power consumption and/or time predictability (e.g. memory access patterns and related choices in terms of memory partitioning and allocation); and b) exploring innovative techniques for analysing the interference between *multiple* applications running concurrently under QoS constraints [12], [13].

The results of both the activities will be exploited as input for the runtime resource management policies. In this regard, a mix of pro-active and reactive strategies will be put in place, exploiting data coming from both the application and the hardware side. Concerning the application side, we expect to rely on a design-time characterization, to perform a priori evaluations, jointly to run-time feedback input about current performance level, to introduce dynamic adjustments during the execution. To this purpose, the introduction of suitable APIs will be taken into account. From the hardware side instead, MANGO will rely on a set of custom monitoring interfaces to investigate and implement novel resource management policies targeting heterogeneous hardware platforms.

Moreover, another feature of the MANGO software stack that is worth to remark is the combination of the aforementioned fine-grained configuration, with the exploitation of the task isolation mechanisms already provided by operating systems (e.g., Linux Control Groups) [14]. This will allow the MANGO runtime resource manager (RTRM) to enforce resource allocation constraints at multiple levels of the heterogeneous hardware architecture, from the general-purpose CPUs to the FPGA based computing devices, passing through the control of the interconnection infrastructure bandwidth. These mechanisms would also enable safe mixed workload executions, with the MANGO run-time resource manager capable of guaranteeing the required QoS to critical tasks, and maximizing the hardware resources utilization at the same time, by providing space to best-effort applications too.

Finally, taking steps from previous power-perfomance investigations considering accurate estimates for both the architecture and the actuators [15], [16], MANGO addresses in a holistic manner the concept of energy reduction considering both computing energy and cooling efficiency as primary goal, thus combining fine-grained monitoring of energy, temperature and power in servers and racks, but also optimization of the mechanical cooling part to use two-phase cooling at rack level.

First of all, MANGO proposes to extend for HPC servers the latest state-of-the-art works on semi-analytical thermal modeling approaches to enable the fast calculation of power/thermal figures of servers under dynamic workload behaviours [17]. The control/optimization policies in the MANGO resource management will be able to evaluate the QoS and non-functional requirements of the applications, in a hierarchical, system-wide multi-objective optimization going beyond electronics to mechanical aspects (e.g., liquid cooling pump control) [18]. The collected information from monitors enables the prediction of temperatures in the different parts of the servers and racks, which will be passed to the hierarchic runtime manager, structured in a hierarchical architecture exploiting both OS and hypervisor levels, which will be able to tune the system knobs (P-states, fan control, tasks assignments, etc.), to mitigate performance variability [19]. Overall, we will target to exploit the run-time thermal-power predictions to mitigate performance variability due to thermal emergencies under highly dynamic workload variations [20], as it is one of the key challenges in the highly heterogeneous MANGO computing server architecture.

IV. Thermal and Cooling Innovations in MANGO

MANGO will extend the experience acquired in the latest research on advanced compact modeling for liquid-cooling monitoring [21] to explore the time constants of thermal and energy control knobs to develop next-generation cooling technologies for HPC systems. In particular, we will explore the use of a novel passive thermosyphon (gravity-driven) cooling technology that will attempt to include multiple *parallel heat sources at multiple elevations* to eliminate

energy consumption. Thus, in MANGO we will carry out preliminary evaluations for the first time in HPC system to re-convert generated heat into electricity at chip level by exploiting the use of microfluidic fuels cells combined with the liquid cooling technology [22]. The preliminary testbeds to evaluate both thermosyphon and energy-recovery process through micro-fluidic fuel cells will be developed and measured in the facilities of the EPFL partner. The objective is to evaluate the creation of HPC servers and rack cooling technologies that can re-use part of their waste heat to generate electricity that reduces external power supply needs.

V. The MANGO Platform Roadmap

The MANGO strategy for building an effective largescale emulation platform will be articulated in three phases.

a) Phase 1 – Stand-alone single-board emulator: The research activities involving architecture exploration will initially rely on current available hardware made of a standalone emulation platform based on FPGA devices and a general purpose node. The standalone emulator will be based on a modular and scalable approach, with several FPGAs being assembled on dedicated daughter modules plugged on a common motherboard. The motherboard will give complete access to all available I/Os of the FPGA, leaving maximum freedom regarding the FPGA interconnection structure, which will allow to define the HN interconnect. The proFPGA quad V7 system provided by ProDesign as a standalone emulation platform will be used. The board is equipped with three Xilinx Virtex 7 XCV2000T FPGA modules and one Zynq module, containing a dual core ARM processor as well as reconfigurable hardware fabric to prototype external subsystems, handling up to 48 M ASIC gates alone in one board. Several proFPGA systems will be interconnected enabling the full HN infrastructure to be implemented. Due to the fact that multiple proFPGA quad or duo systems can be stacked or connected together, scalability is ensured. The highspeed boards together with the specific high speed connectors allow a maximum point to point speed of up to 1.8 Gbps over the standard FPGA I/O and up to 12.5 Gbps over the MGT of the FPGA.

b) Phase 2 – From FPGA stand-alone board to a dedicated chassis: A new board for HPC will be implemented complying with the physical constraints of HPC and datacenter racks, considering as well requirements for cooling and power supply researched within the project. The board will be extended to deliver further number of daughter boards and will provide proper connectivity through optical links to other boards. Pin-to-pin connectivity between FPGAs (either at the same board or at different boards) will allow expandability and scalability. This enables MANGO to explore future chip configurations in a predictable and accurate manner. Daughter boards will be extensible and open to new developments, particularly to new 64-bit ARM cores or even more advanced solutions like the hybrid Xeon E5+FPGA chip recently announced by Intel. In this phase, the HN interconnect will be applied to the set of HN nodes (the board) developed. It will embrace connectivity at the board level, between ARM and FPGA modules, inside the FPGA modules (within the accelerators and RISC processors implemented), and between the boards. This means a single and unified interconnect will be designed for the overall HN infrastructure (made of 64 nodes).

c) Phase 3 – Rack assembly: As a final phase, the complete rack will be implemented and populated of GNs and HNs. The system will enable a large-scale platform used to reproduce in near real-time the behavior of the MANGO manycore architecture. The full platform will consist of a rack collecting up to 16 blades equipped with high-end CPUs, e.g. Intel Xeon chips, and GPUs, mounted on the motherboard, as well as 64 HN nodes. A custom backplane will provide connectivity across the blades, both through standard bridges and using pin-to-pin connections across the FPGA chips, effectively providing a single large-scale reconfigurable hardware fabric used to emulate the fine-grained accelerator tiles envisioned in the MANGO architecture. The inter-FPGA pin-to-pin backplane interconnection will be reconfigurable on-field, providing a large degree of flexibility for the emulation of the on-chip network interconnect.

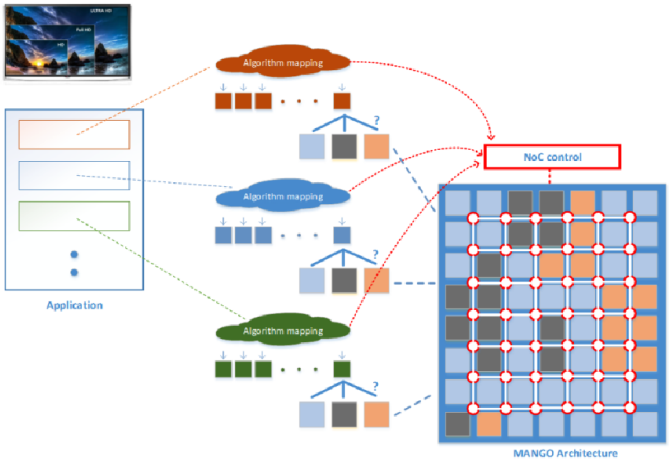

Fig. 3. Mapping Applications on the MANGO Platform

VI. MANGO Application Space

An important aspect of the MANGO architecture exploration is showcase of dynamic nature of the architecture and its capabilities to dynamically use heterogeneous processing elements in a QoS sensitive computing scenario. The integrated approach to MANGO research will be demonstrated by a set of applications that clearly demonstrate QoS aspects.

A. MANGO architecture online video transcoding application platform

Multimedia playback on different presentation devices has experienced significant growth. Some market forecasts [23] show that annual global IP traffic will pass the zettabyte (1000 exabytes) threshold by the end of 2016, and will reach 2 zettabytes per year by 2019. In the same time, globally, IP video traffic will be in the range of 80 to 90 percent of all IP traffic (both business and consumer) by 2019.

Because of the great variety of devices which are accessing the multimedia content under adverse network conditions, current video streaming systems in most cases do not provide optimal video quality and thus waste valuable resources or unnecessarily lower the Quality of Experience (QoE).

Efficient processing of video transcoding, which is extremely compute-intensive and has stringent timing requirements, provides an ideal case-study for the QoS-aware HPC solutions explored by MANGO. The heterogeneous core MANGO HPC transcoding application platform can therefore truly demonstrate how the novel MANGO HPC architecture may become dominant architecture for applications that generate more than 80% of global internet traffic. The application of same algorithms to medical domains where interoperability requirements are defined (see [24]) is also under consideration.

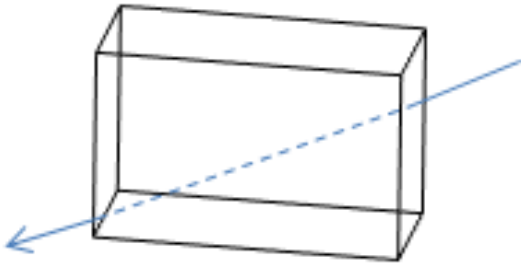

Fig. 4. Ray casting through a volume [27]

The real time video transcoding will use novel video coding algorithms such High Efficiency Video Coding HEVC/H.265. To enable efficient transcoding, significant work will be required in modeling, mapping and optimizing parts of the algorithms to different underlying MANGO architecture elements (tiles), as shown in Figure 3. Research will not only be focused to high optimization of SW implementation but also design of application specific tiles, implemented in HW (such as [25]), that will allow more efficient processing from performance, power and QoS points of view.

B. Volume rendering for medical imaging

Philips will deploy the Volume rendering technique on the MANGO prototype. Philips ships a variety of imaging equipment, which includes MRI (Magnetic Resonance Imaging), CT (Computed Tomography) and PET (Positron Emission Tomography). In such equipment, medical images are stored as sets of parallel planar images. Such a set can be stacked together, which forms a 3D rectangular space around the stacked images. We refer to such a set as a volume. Volume rendering is a visualization technique that uses the ray casting technique to visualize a volume on a 2D plane. Ray casting visualizes this volume by calculating the attenuation of rays of light [26].

In Ray Casting, the color of each resulting image pixel is determined by following a single ray of light through the volume, see Figure 4. Density values in the images of the volume are sampled along the ray. Transfer functions for the color and transparency, determine the color and transparency of these sample points along the ray. The attenuation of these colors and transparencies along a single ray will determine the color of a resulting pixel. This process is repeated for every pixel in the single visualization result image. The calculations are highly memory intensive (data sizes range from 250MB to 1GB). Fast rendering and a low latency transfer from central processing to the users workstation are crucial. This is to ensure a better hospital workflow and better diagnosis outcome. The solution must scale among many healthcare users, all operating on different patient data. Users interact with the system in real time, requesting new renderings with frequencies of up to 25 times per second. In more severe cases, imaging equipment is increasingly used during minimal invasive intervention. As the surgeon cannot see what he/she is doing, an image stream with low latency and low litter has to be presented. The MANGO solution will allow Philips to improve the offering for diagnosis equipment to the hospitals. The final product will much better serve the interventional market.

C. Datacenter Secure Traffic

The presence of secure traffic in datacenters is increasing and will continue to do so as cyber physical systems (CPS) and the internet of things (IoT) domains keep growing. Traffic from multiple sources with different levels of importance will need to be stored and

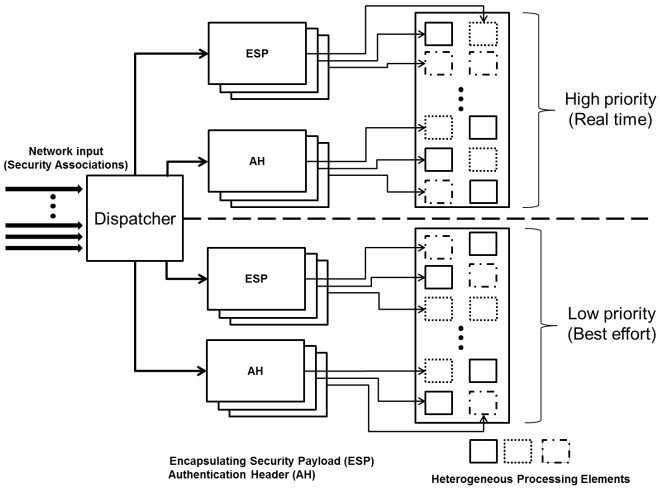

Fig. 5. Data flows of Strongswan with parallelized and heterogeneous offloading

processed. Along with such traffic comes the necessity to provide verification of the source's origin (authentication) and ensure the content has not been a subject of theft, alteration or corruption (integrity, encryption). So far such requirements have been covered at the Internet Protocol (IP) level using various open standards for authentication of origin, payload integrity, transmission confidentiality and protection against attacks. Different software implementations of these protocols exist, one being *Strongswan*, developed by University of Applied Sciences Rapperswil in Switzerland [28]. Strongswan opens secure communication tunnels between network nodes or routers which allow the transmission of information while maintaining a high level of security. The data flows of Strongswan are shown in Figure 5.

Currently computer architectures improve performance by increasing the degree of parallelism leading to manycore platforms. At the same time we notice that the multiplicity of cores is accompanied by high diversity in microarchitecture making these platforms heterogeneous. Tasks are offloaded to cores that handle a certain nature of calculations efficiently providing interesting results in terms of performance per watt. Consequently, exploiting resources to the highest degree becomes a necessity that can be achieved by adapting existing applications to this model. Strongswan is a good candidate because of its parallel nature (creating multiple tunnels) and of the heterogeneity of the algorithms it uses. Considering a thread per tunnel, offloading different parts of it to a different computation cores can potentially yield higher performance gains. Moreover, in Software Defined Networks (SDN) and Network Function Virtualization (NFV) we identify the tendency for network functions to be implemented in software, making Strongswan an adequate subject of study for both use cases. Benefits of this effort can be for example, an increased level of flexibility in infrastructure security. Managing resource usage in relation to the system's global charge can help achieve a high performance/power ratio while maintaining security and determinism constraints. Furthermore, updating supported protocols and algorithms in software will be far easier which will allow keeping platforms compliant with modern security standards.

For this architectural adaptation we need to consider limitations or challenges that can emerge. A cycle of analyzing, modifying and testing will be necessary to port any application in order to achieve maximum efficiency. Firstly we need to consider the amount of work needed to modify the source code into a structure that will

allow a modular execution with minimal dependencies. This will need to be backed by tools for analyzing the code (statically and dynamically) and a compiler that is able to support the parallelism and heterogeneity of the architecture. The objective behind it is to minimize source code modifications needed to incorporate architectural parameters. Secondly, the platforms must allow to measure and qualify the modifications in terms of performance. This will help to overcome the challenge of defining a threshold where offloading provides a higher gain as opposed to traditional execution. By testing different execution schemes we can identify the type of the cores that are more suitable for each task. Furthermore, the distance (at hardware level) between each offloaded kernel and the main program will need to be evaluated in order to obtain good performance in relation to a task's nature (data/io intensive or computation intensive). Finally, analyzing task execution will provide vital info from a real-time (latency) (critical / non-critical flows) and security (on-chip) aspects. These are indirect constraints that can be imposed depending on the the role of the infrastructure. Resource sharing policies can provide better determinism for tasks that have temporal deadlines or equally isolate data traffic for tasks that have a security profile demanding it. Overall, managing resources optimally in manycores can help fill performance gaps that might be present and maintain or ameliorate behavior in respect to timeliness or security requirements.

VII. Conclusions

The MANGO project, started in October 2015, will last for three years, with the goal of addressing power, performance and predictability in HPC. It will rely on customization to adapt the available computing resource to reach these goals.

Acknowledgements

This project has received funding from the the European Union's Horizon 2020 research and innovation programme under grant agreement No 671668.

References

[1] European Technology Platform For HPC, "ETP4HPC Strategic Research Agenda: Achieving HPC leadership in Europe," http://www.etp4hpc.eu/strategy/strategic-research-agenda/, 2013.

[2] Khronos Group, "The Open Standard for Parallel Programming of Heterogeneous Systems," https://www.khronos.org/opencl/, (retr. Jul 2015).

[3] G. Agosta, A. Barenghi, A. Di Federico, and G. Pelosi, "OpenCL Performance Portability for General-purpose Computation on Graphics Processor Units: an Exploration on Cryptographic Primitives," *Concurrency and Computation: Practice and Experience*, 2014.

[4] G. Agosta, A. Barenghi, G. Pelosi, and M. Scandale, "Towards Transparently Tackling Functionality and Performance Issues across Different OpenCL Platforms," in *2nd Int'l Symp. on Computing and Networking (CANDAR)*, Dec 2014, pp. 130–136.

[5] C. Silvano, W. Fornaciari, S. Reghizzi, G. Agosta, G. Palermo, V. Zaccaria, P. Bellasi, F. Castro, S. Corbetta, A. Di Biagio, E. Speziale, M. Tartara, D. Melpignano, J.-M. Zins, D. Siorpaes, H. Huebert, B. Stabernack, J. Brandenburg, M. Palkovic, P. Raghavan, C. Ykman-Couvreur, A. Bartzas, S. Xydis, D. Soudris, T. Kempf, G. Ascheid, R. Leupers, H. Meyr, J. Ansari, P. Mahonen, and B. Vanthournout, "2PARMA: Parallel Paradigms and Run-time Management Techniques for Many-Core Architectures," in *VLSI 2010 Annual Symposium*, ser. LNEE. Springer NL, 2011, vol. 105, pp. 65–79.

[6] C. Silvano, W. Fornaciari, S. Reghizzi, G. Agosta, G. Palermo, V. Zaccaria, P. Bellasi, F. Castro, S. Corbetta, E. Speziale, D. Melpignano, J. Zins, D. Siorpaes, H. Hubert, B. Stabernack, J. Brandenburg, M. Palkovic, P. Raghavan, C. Ykman-Couvreur, A. Bartzas, D. Soudris, T. Kempf, G. Ascheid, H. Meyr, J. Ansari, P. Mahonen, and B. Vanthournout, "Parallel paradigms and run-time management techniques for many-core architectures: The 2parma approach," in *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, July 2011, pp. 835–840.

[7] 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

[8] A. Cilardo and L. Gallo, "Improving multibank memory access parallelism with lattice-based partitioning," *ACM Trans. Archit. Code Optim.*, vol. 11, no. 4, pp. 45:1–45:25, Jan. 2015.

[9] A. Cilardo, L. Gallo, and N. Mazzocca, "Design space exploration for high-level synthesis of multi-threaded applications," *Journal of Systems Architecture*, vol. 59, no. 10, pp. 1171–1183, 2013.

[10] A. Cilardo and L. Gallo, "Interplay of loop unrolling and multidimensional memory partitioning in hls," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2015*, March 2015, pp. 163–168.

[11] M.-W. Benabderrahmane, L.-N. Pouchet, A. Cohen, and C. Bastoul, "The Polyhedral Model Is More Widely Applicable Than You Think," in *Compiler Construction*, ser. LNCS, R. Gupta, Ed. Springer Berlin Heidelberg, 2010, vol. 6011, pp. 283–303.

[12] D. Zoni, S. Corbetta, and W. Fornaciari, "Thermal/performance trade-off in network-on-chip architectures," in *System on Chip (SoC), 2012 International Symposium on*, Oct 2012, pp. 1–8.

[13] S. Libutti, G. Massari, and W. Fornaciari, "Co-scheduling tasks on multi-core heterogeneous systems: An energy-aware perspective," *IET Computers & Digital Techniques*, 2015.

[14] P. Bellasi, G. Massari, and W. Fornaciari, "Effective Runtime Resource Management Using Linux Control Groups with the BarbequeRTRM Framework," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 14, no. 2, p. 39, 2015.

[15] D. Zoni, F. Terraneo, and W. Fornaciari, "A dvfs cycle accurate simulation framework with asynchronous noc design for power-performance optimizations," *Journal of Signal Processing Systems*, pp. 1–15, 2015.

[16] D. Zoni, F. Terraneo, and W. Fornaciari, "A control-based methodology for power-performance optimization in nocs exploiting dvfs," *Journal of Systems Architecture*, vol. 61, no. 5-6, pp. 197 – 209, 2015.

[17] A. Sridhar, M. Sabry, and D. Atienza, "A Semi-Analytical Thermal Modeling Framework for Liquid-Cooled ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 8, pp. 1145–1158, 2014.

[18] A. K. Coskun, D. Atienza, M. Sabry, and J. Meng, "Attaining Single-Chip, High-Performance Computing Through 3D Systems with Active Cooling," *IEEE Micro Magazine*, vol. 31, no. 4, pp. 63–75, 2011.

[19] J. Kim, M. Sabry, D. Atienza, K. Vaidyanathan, and K. Gross, "Global fan speed control considering non-ideal temperature measurements in enterprise servers," in *Proc. IEEE/ACM Design, Automation and Test in Europe (DATE '14)*, Dresden, DE, 2014, pp. 303–308.

[20] J. Kim, M. Ruggiero, D. Atienza, and M. Ledergerber, "Correlation-Aware Virtual Machine Allocation for Energy-Efficient Datacenters," in *Proc. IEEE/ACM Design, Automation and Test in Europe (DATE '13)*, Grenoble, FR, 2013, pp. 216–221.

[21] A. Sridhar, A. Vincenzi, M. Ruggiero, and D. Atienza, "Neural network-based thermal simulation of integrated circuits on gpus," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 1, pp. 23–36, 2012.

[22] M. Sabry, A. Sridhar, P. Ruch, D. Atienza, and B. Michel, "Integrated Microfluidic Power Generation and Cooling for Bright Silicon MPSoCs," in *Proceedings of the IEEE/ACM Design, Automation and Test in Europe (DATE '13)*. Dresden, Germany: IEEE-ACM Press, 2014, pp. 10–15.

[23] CISCO, "The Zettabyte Era – Trends and Analysis," http://www.cisco.com, May 2015.

[24] M. Kovač, "E-Health Demystified: An E-Government Showcase," *Computer*, vol. 47, no. 10, pp. 34–42, Oct 2014.

[25] M. Kovač and N. Ranganathan, "Vlsi circuit structure for implementing jpeg image compression standard," Aug. 1997, US Patent 5,659,362. [Online]. Available: http://www.google.com/patents/US5659362

[26] J. Pawasauskas, "Volume Visualization With Ray Casting," http://web.cs.wpi.edu/ matt/courses/cs563/talks/powwie/p1/ray-cast.htm, Feb 1997.

[27] B. Preim and C. P. Botha, *Visual Computing for Medicine: Theory, Algorithms, and Applications*. Newnes, 2013.

[28] A. Steffen, "Advanced Features of Linux strongSwan – the OpenSource VPN Solution," in *LinuxTag*, 2005.