AutoTuning and Adaptivity appRoach for Energy efficient eXascale HPC systems: the ANTAREX Approach*

Cristina Silvano*, Giovanni Agosta*, Andrea Bartolini†, Andrea R. Beccari‡, Luca Benini†, João Bispo§, João M. P. Cardoso§, Carlo Cavazzoni¶, Jan Martinovič‖, Gianluca Palermo*, Martin Palkovič‖, Pedro Pinto§, Erven Rohou**, Nico Sanna¶, Kateřina Slaninová‖

*DEIB – Politecnico di Milano, †IIS – Eidgenössische Technische Hochschule Zürich, ‡Dompé Farmaceutici SpA, §FEUP – Universidade do Porto, ¶CINECA, ‖IT4Innovation National Supercomputing Center, **INRIA Rennes

Email: *name.surname@polimi.it †{barandre,lbenini}@iis.ee.ethz.ch, ‡andrea.beccari@dompe.it, §{jbispo,pmsp,jmpc}@fe.up.pt, ¶n.surname@cineca.it, ‖name.surname@vsb.cz, **erven.rohou@inria.fr

***Abstract*—The main goal of the ANTAREX project is to express by a Domain Specific Language (DSL) the application self-adaptivity and to runtime manage and autotune applications for green and heterogeneous High Performance Computing (HPC) systems up to the Exascale level. Key innovations of the project include the introduction of a separation of concerns between self-adaptivity strategies and application functionalities. The DSL approach will allow the definition of energy-efficiency, performance, and adaptivity strategies as well as their enforcement at runtime through application autotuning and resource and power management.**

***Keywords*—High Performance Computing, Autotuning, Adaptivity**

I. Introduction

High Performance Computing (HPC) has been traditionally the domain of grand scientific challenges and a few industrial domains such as oil & gas or finance, where investments are large enough to support massive computing infrastructures. Nowadays HPC is recognized as a powerful technology to increase the competitiveness of nations and their industrial sector, including small scale but high-tech businesses – *to compete, you must compute* has become an ubiquitous slogan [1]. The current road-maps for HPC systems aim at reaching exascale levels (10^{18} FLOPS) within 2020 – a $\times 1000$ improvement over petascale, which was reached in 2009, and a $\times 100$ improvement over current systems. Reaching exascale poses the additional challenge of significantly limiting the energy envelope while providing massive increases in computational capabilities – the target power envelope for future exascale system ranges between 20 and 30 megawatts. Thus, "Green" HPC systems, are being designed aiming at maximizing a FLOPS/Watt metric, rather than just FLOPS, and employing increasingly heterogeneous architectures with GPGPU or MIC accelerators. On average, the efficiency of heterogeneous systems is more almost three times that of homogeneous systems (i.e., $7,032$ MFLOPS/Watt vs 2304 MFLOPS/Watt[1]). This level of efficiency is still two orders of magnitude lower than that needed for supporting exascale systems at the target power envelope of 20 megawatts. To this end, European efforts have been focused towards building supercomputers out of the less power-hungry ARM cores and GPGPUs [2]. On the other hand, the wide margin provided by modern chip manufacturing techniques, combined with the inability to exploit this space to produce faster, more complex cores due to the breakdown of Dennard scaling [3], has given rise to a pervasive diffusion of a number of parallel computing architectures, up to the point where embedded systems are also characterized by multicore processors. The large design effort has led to a variety of approaches in terms of core interconnection and data management. Thus, the ability to port applications designed for current platforms, based on GPGPUs like the NVIDIA Kepler or Tesla families, to heterogeneous systems such as those currently designed for embedded systems is critical to provide software support for future HPC.

Designing and implementing HPC applications is a difficult and complex task, which requires mastering many specialized languages and tools for performance tuning. This is incompatible with the current drive to opening HPC infrastructures to a much wider range of users. The current model of having the HPC center staff directly supporting the development of the application will become unsustainable in the long term. Thus, the availability of effective standard programming languages and APIs is critical to provide migration paths towards novel heterogeneous HPC platforms as well as to guarantee the ability of developers to work effectively on these platforms. To fulfil the 20MWatt target, energy-efficient heterogeneous supercomputers need to be coupled with a radically new software stack capable of exploiting the benefits offered by heterogeneity at all the different levels (supercomputer, job, node).

ANTAREX will address these challenging problems through a holistic approach spanning all the decision layers composing the supercomputer software stack and exploiting effectively the full system capabilities (including heterogeneity and energy management). The main goal of the ANTAREX project is to express by a DSL the application self-adaptivity and to runtime manage and autotune applications for green heterogeneous HPC systems up to Exascale.

One key innovation of the proposed approach consists of introducing a separation of concerns (where self-adaptivity and energy efficient strategies are specified aside to application functionalities) promoted by the definition of a DSL inspired by aspect-oriented programming concepts for heterogeneous systems. The new DSL will be introduced for expressing at compile time the adaptivity/energy/performance strategies and to enforce at runtime application autotuning and resource and power management. The goal is to support the parallelism, scalability and adaptivity of a dynamic workload by exploiting the full system capabilities (including energy management) for emerging large-scale and extreme-scale systems, while reducing the Total Cost of Ownership (TCO) for companies and public organizations. ANTAREX approach will be based on: (1) introducing a new DSL for expressing adaptivity and autotuning strategies; (2) enabling the performance/energy control capabilities by introducing software knobs (including application parameters, code transformations and code variants); (3) designing scalable and hierarchical optimal control-loops capable of dynamically leveraging them together with performance/energy control knobs at different time scale (compile-, deploy- and run-time) to always operate the supercomputer and each application at the maximum energy-efficient and thermally-safe point. This can be done by monitoring the evolution of the supercomputer as well as the application status

*ANTAREX is supported by the EU H2020 FET-HPC program under grant 671623

[1] www.green500.org, June 2015

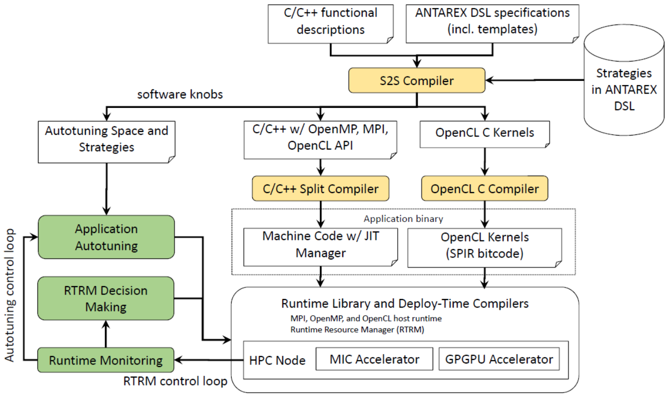

Fig. 1. The ANTAREX Tool Flow

and requirements and bringing this information to the ANTAREX energy/performance-aware software stack.

The ANTAREX project is driven by two use cases from highly relevant HPC application scenarios: (1) a biopharmaceutical HPC application for drug discovery deployed on the 1.2 PetaFlops heterogeneous NeXtScale Tier-1 Intel-based IBM system based at CINECA and (2) a self-adaptive navigation system for smart cities deployed on the server-side on the 1.457 PetaFlops heterogeneous Intel® Xeon Phi™ based system provided by IT4Innovations National Supercomputing Center. All the key ANTAREX software innovations will be designed and engineered since the beginning to be scaled-up to the Exascale level. Performance metrics extracted from the two use cases will be modelled to extrapolate these results towards Exascale systems expected by the end of 2020.

The ANTAREX Consortium comprises a wealth of expertise in all pertinent domains. Four top-ranked academic and research partners (Politecnico di Milano, ETHZ Zurich, University of Porto and INRIA) with extensive experience in the salient research topics to be explored will be complemented by the Italian Tier-0 Supercomputing Center (CINECA), the the Tier-1 Czech National Supercomputing Center (IT4Innovations) and two industrial application providers, one of the leading biopharmaceutical companies in Europe (Dompé) and the top European navigation software company (Sygic).

II. The ANTAREX Approach

The ANTAREX approach and related tool flow, shown in Figure 1, operates both at design-time and runtime. The application functionality is expressed through C/C++ code (possibly including legacy code), whereas the non-functional aspects of the application, including parallelisation, mapping, and adaptivity strategies are expressed through the DSL developed in the project. One of the benefits consists of facilitating the reuse of legacy code. In the definition of these strategies, the application developer or system integrator can leverage DSL templates that encapsulate specific mechanisms, including how to generate code for OpenCL or OpenMP parallelisation, and how to interact with the runtime resource manager. The DSL weaver and refactoring tool will then enhance the C/C++ functional specification with the desired adaptivity strategies, generating a version of the code that includes the necessary libraries as well as the partitioning between the code for the GPPs and the code for the accelerators (such as GPGPUs and MIC accelerators [4]). A mix of off-the-shelf and custom compilers will be used to generated code, balancing between development effort and optimization level.

Thus, the ANTAREX compilation flow leverages a runtime phase with compilation steps, through the use of split-compilation techniques. The application autotuning is delayed to the runtime phase, where the software knobs (application parameters, code transformations and code variants) are configured according to the runtime information coming from the execution environment. Finally the runtime resource and power manager are used to control the resource usage for the underlying computing infrastructure given the changing conditions. At runtime, the application control code, thanks to the design-time phase, now contains also runtime monitoring and adaptivity strategies code derived from the DSL extra-functional specification. Thus, the application is continuously monitored to guarantee the required Service Level Agreement (SLA), while communication with the runtime resource-manager takes place to control the amount of processing resources needed by the application. The application monitoring and auto-tuning will be supported by a runtime layer implementing an application level collect-analyse-decide-act loop.

III. The ANTAREX DSL

HPC applications may profit by adapting to operational and situational conditions, such as changes in contextual information (e.g., workloads), in requirements (e.g., deadlines), and in resources availability (e.g., energy, connectivity, number of processor nodes available). A simplistic approach to both adaptation specification and implementation (see, e.g., [5]) employs hard coding of, e.g., conditional expressions and parameterizations. In our approach the specification of runtime adaptability strategies will rely on a DSL implementing key concepts from Aspect-Oriented Programming (AOP) [6]. AOP is based on the idea that the non-functional requirements of a system (e.g., target-dependent optimizations) should be specified separately from the source code that defines the functionality of the program. An extra compilation step, performed by a *weaver*, merges the original source code and the aspects into the intended program [7]. Using aspects to separate secondary concerns from the core objective of the program can result in cleaner programs and increased productivity (e.g., higher reusability of target-dependent strategies).

A. LARA for Runtime Adaptivity and Compiler Optimization

LARA [8, 9] is an AOP approach that enables developers to capture non-functional requirements and concerns in the form of strategies, which are decoupled from the functional description of the application. Compared to other approaches that usually focus on code injection (e.g., [10] and [11]), LARA provides access to other types of actions, e.g., function renaming and inclusion of additional information. Additional types of actions may be defined in the language specification of the weaver. Examples of such actions in previous weavers include software/hardware partitioning [12] or specifying compiler optimization sequences [13].

Figure 2 shows an example of a LARA aspect that profiles function calls in order to gather information about argument values and their frequency. It injects code for an external C library that records the name of the function being called, its location and the value of the arguments. A concern intended to be applied over the target application is expressed as one or more aspect definitions, or *aspectdef*, the basic modular unit of LARA. Aspects, similarly to functions, can receive inputs, and return outputs.

An aspect is comprised of three main steps, which will carry out the intended concern. First, one captures the points of interest in the code using a *select* statement, which in the example selects calls to functions. Then, using the *apply* statement, one acts over the selected points. In this case, it will insert code before the function call. We can then define a *condition* statement to constrain the execution of the *apply* (i.e., only the function with the provided name is selected).

```
aspectdef ProfileArguments
  input funcName end
  select fCall end
  apply
    insert before %{profile_args('[[funcName]]',
                  [[$fCall.location]],
                  [[$fCall.argList]]);
    }%;
  end
  condition $fCall.name == funcName end
end
```

Fig. 2. Example of LARA aspect for profiling.

To allow the definition of more sophisticated concerns, it is possible to embed JavaScript code inside aspects. LARA promotes modularity and aspect reuse, as we can, from within an aspect, calls others and pass arguments as if they were functions.

Figure 3 presents an example of a LARA aspect that unrolls innermost `FOR` loops whose iteration count is less than the given threshold (it uses an action supported by the weaver - keyword `do`). In this example this aspect will be called in a dynamic context after we specialize a function for a given argument value.

```
aspectdef UnrollInnermostLoops
  input $func, threshold end
  select $func.loop{type=='for'} end
  apply
    do LoopUnroll('full');
  end
  condition
    $loop.isInnermost && $loop.numIter <= threshold
  end
end
```

Fig. 3. LARA aspect that unrolls loops.

Figure 4 has an example of a possible aspect that could be written if dynamic weaving was supported. This aspect will monitor function calls to the function `kernel` in runtime, and specialize it if the runtime value of parameter `size` is between a range defined by `lowT` and `highT`. First, it statically prepares the function call to support several versions of the function, according to the parameter `size` (the keyword `call` calls an aspect). Then, dynamically, it specializes the function call for the current value of the parameter `size`, and unrolls the loops of the newly developed specialized function. Finally, it adds the specialized version as one of the possible that can be called.

B. ANTAREX DSL Concepts

We consider that the current LARA infrastructure will serve as a solid foundation to build a sophisticated DSL that will enable us to specify runtime adaptability strategies.

The ANTAREX DSL approach aims at reaching a higher abstraction level, to separate and express data communication and computation parallelism, and to augment the capabilities of existing programming models by passing hints and metadata to the compilers for further optimization. We aim at improving performance portability with respect to current programming models, such as OpenCL, where fine-tuning of performance (which is very sensitive to even minimal variation in the architectural parameters [14, 15]) is left entirely to the programmer, by exploiting the capabilities of the DSL to automatically explore configuration space for the parallel code.

To this end, iterative compilation [16] techniques are attractive, since they allow to identify the best compiler optimizations for a

```
aspectdef SpecializeKernel
  input lowT, highT end

  call spCall: PrepareSpecialize('kernel','size');

  select fCall{'kernel'}.arg{'size'} end
  apply dynamic
    call spOut : Specialize($fCall, $arg.name,
      $arg.runtimeValue);
    call UnrollInnermostLoops(spOut.$func,
      $arg.runtimeValue);
    call AddVersion(spCall, spOut.$func,
      $arg.runtimeValue);
  end
  condition
    $arg.runtimeValue >= lowT &&
    $arg.runtimeValue <= highT
  end
end
```

Fig. 4. Example of possible aspect with apply at runtime.

given program/code fragment considering possible trade-offs. Given the diversity of heterogeneous multiprocessors and the potential for optimizations provided by runtime information, runtime optimization is also desirable. To combine the two approach, a *split compilation* will be used. The key idea is to split the compilation process in two steps - offline, and online - and to offload as much of the complexity as possible to the offline step, conveying the result to runtime optimizers [17]. We will encode code generation strategies to drive a dynamic code generator in response to particular hardware features as well as dynamic information. This combination of iterative- and split-compilation will have a significant impact on the performance of applications, but also on the productivity of programmes as it will relieve programmers from the burden of repeatedly optimizing, tuning, compiling and testing.

IV. Self-Adaptivity & Autotuning

The management of system adaptivity and autotuning is a key issue in HPC systems, where the computing infrastructure can easily evolve and the system needs to react promptly to changing workloads and events, without impacting too much its extra-functional characteristics, such as energy and thermal features [18]. The motivation behind this trend can be easily explained by the requirement to meet the maximum performance/power ratio across all the possible deployments of the applications. This is especially important when considering the rapid grow of computing infrastructures that continue to evolve on one hand by increasing computing nodes, while on other hand by increasing the performance exploiting heterogeneity in terms of accelerators/co-processors. Thus, there is a requirement on applications to become adaptive with respect to the computing resources. In this direction, another interesting effect is that there is a growing need of guaranteeing Service Level Agreement (SLA) both at the server- and at the application-side. This need is related to the performance of the application, but also to the maximum power budget that can be allocated to a specific computation. In this context, our efforts are mainly focused on two main paths: i) the development of an autotuning framework capable to configure and adapt application-level parameters and ii) the possibility to export the concept of precision autotuning to HPC applications.

Application Autotuning. Two types of framework have been investigated so far to support application autotuning depending on the knowledge about the target domain: white-boxes and black-boxes. White-box techniques are those approaches based on autotuning libraries (thus not general) that deeply use the domain specific knowledge to fast surf the parameter space. On the other side, black-box techniques are always-applicable frameworks since they do not require any knowledge on the underlying application, however suffering of long convergence time and less custom possibilities. The proposed framework falls in the area of grey-box approaches since, starting from the idea of non-domain knowledge, it can be fed code annotations to shrink the search space by focusing the auto-tuner on a certain subspace. Moreover, the framework includes a monitoring loop to be used to monitor the application and to trigger the application adaptation. The monitoring, together with application properties/features, represents the main support to the decision-making during the application autotuning phase since it is used to perform statistical analysis related to system performance and other SLA aspects. Continuous on-line learning techniques is adopted to update the knowledge from the data collected by the monitors, giving the possibility to auto-tune the system always according to the more recent operating conditions. Machine learning techniques are also adopted in the decision-making engine to support the auto-tuning by predicting the most promising set of parameter settings.

Precision Autotuning. In the recent years, customized precision has emerged as a promising approach to achieve power/performance trade-offs. It derives from the fact that many applications can tolerate some loss of quality during computation, as in the case of media processing (audio, video and image) and data mining. These applications are increasingly common in the emerging field of real-time HPC. In ANTAREX, the benefits of customized precision HPC applications will be investigated in tight collaboration with the domain experts of the two use cases. Additionally, we also plan to apply fully automatic dynamic optimizations, based on profiling information, and data acquired at runtime, e.g. dynamic range of function parameters.

In both cases the usage of the ANTAREX DSL will be a key-point to decouple the functional specification of the application from the definition of software knobs (such as code variants or application parameters) and from the precision tuning phase.

V. Runtime Resource & Power Management

ANTAREX follows a holistic approach toward next-generation energy-efficient exascale supercomputers. While traditional design of green supercomputer relies the integration of best-in-class energy-efficient components [19], recent works [20, 21] show that as effect of this design practice supercomputers are nowadays heterogeneous system. Indeed supercomputers are not only composed by heterogeneous computing architectures (GPGPUs and CPUs), but different instances of the same nominal component execute the same application with 15% of variation in the energy-consumption. Different applications on the same resources show different performance-energy trade-offs, for which an optimal selection of operating point can save in between 18% and 50% of node energy w.r.t. the default frequency selection of the Linux OS power governor. Moreover it has been recently shown that environmental conditions, such as ambient temperature, can significantly change the overall cooling efficiency of a supercomputer, causing more than 10% *Power usage effectiveness* (PUE) loss when transitioning from winter to summer [22]. These sources of heterogeneity, coupled with current worst-case design practices, lead to significant less in energy-efficiency, and to missed opportunities for run-time resource and power management (RTRM, RTPM). ANTAREX leverages RTRM and RTPM by combining: (1) novel introspection points, application progress and dynamic requirements; (2) autotuning capabilities enabled by the DSL in the applications; and (3) information coming from the processing elements of the Exascale machine and IT infrastructure and the respective performance knobs (resource allocation, DVFS, cooling effort, room temperature). These information flows converge in a scalable multilayer resource management infrastructure. The information will be used to allocate to each application the set of resources and their operating point which maximize the entire supercomputer energy-efficiency, while respecting SLA and safe working conditions. The latter will be ensured by the resource management solution by optimal selection of the cooling effort and by a distributed optimal thermal management controller. The ANTAREX power management and resource allocation approach is based on: (1) Expanding the energy/performance control capabilities by introducing novel software control knobs (i.e. software reconfigurability and adaptability); (2) Designing scalable and hierarchical optimal control-loops capable of dynamically leveraging the control knobs together with classical performance/energy control knobs (job dispatching, resource management and Dynamic Voltage and Frequency Scaling) at different time scale (compile time, deployment time and run-time); (3) Monitoring the HW supercomputing evolution, the application status and requirements, bringing this information to the energy/performance-aware software stack. This approach will allow to always operate the supercomputer and each application at the most energy-efficient and thermally-safe point.

VI. Target Platforms & HPC Center Roadmaps

The target platforms are the CINECA's Tier-1 IBM NeXtScale hybrid Linux cluster, based on Intel TrueScale interconnect as well as Xeon Haswell processors and MIC accelerators [4], and IT4Innovations Salomon supercomputer, which is a PetaFlop class system consisting of 1008 computational nodes.

A. CINECA Roadmap

In the aim of its mission of a continued support to the competitiveness of the Italian research infrastructure and on the ground of international agreements for supercomputing in Europe (PRACE), CINECA is going to evolve its massive computing and data handling infrastructure by setting up a five year strategy aimed at bridging the gap from here to the exascale era in 2020. This process of architecture migration is by nature manifold and complex. It should take into account emerging technological and scientific challenges and must be feasible in term of economic and environment sustainability.

To this end, CINECA forecasts future supercomputing facilities to see converging traditional HPC architecture toward a new integrated infrastructure capable of approaching the new emerging paradigms for advanced computational sciences: high performance data analytics (BigData), urgent or interactive supercomputing (Human Brain Project), real time data processing, cloud computing.

In view of this multi-faceted scenario, the model of service CINECA is going to adopt is a data centric paradigm where the data, increasing exponentially in size, will not need to be moved anymore from different computing systems but integrated with them thus offering a single solution for numerical and data analysis.

In the process of architecture evolution, the upgrade of our Tier-0 (Fermi) is just a first step toward a tight integration between computing and data analysis. The new supercomputing machine (Marconi) expected to be in production on the second-half of 2016, will be set up as a scalable hybrid cluster starting at 10 PFLOPS to reach 50 PFLOPS in three years but continuously upgradable to go even beyond 100 PFLOPS for 2020 or so. This new massive computing system will be initially coupled with our High Performance and Big Data Analytics system (Pico) capable of managing at present 10 PB of running data, but itself upgradable to be able to follow in term of size and performance the Marconi supercomputing evolution.

But the path following to high-end computing and data managing has revealed during the last ten years or so, a major constrain in the exponential growth of power consumption in a way that is now commonly accepted the sustainable limit of 20MW for the first exaFLOPS supercomputing system. With this limit in mind CINECA is modelling its strategy for future HPC/DA systems in order to experiment, design and deploy energy-aware supercomputing facilities. To this end, CINECA has been involved in several projects at national and EU level to take advantage of the most promising techniques for limiting the overall requirements of energy-to-solution computing workflow [23, 24, 25, 26].

In this context, the participation of CINECA to the ANTAREX project paves the route to even a smarter idea of managing a large computing infrastructure by leveraging from hardware to application level the computing workflow for the best and most sustainable execution behaviour.

B. IT4Innovations Roadmap

IT4Innovations operates nationally unique state-of-the-art supercomputing resources and provides open access to those resources on a scientific excellence basis. IT4Innovations portfolio of services is split into the core services and added value services. The core services consist of provisioning optimized HPC environment, in particular core hours of supercomputer time, software license hours and compute-oriented storage capacity. The added value services are based on the expertise accumulated at IT4Innovations and include user and application support, code enabling and optimization, training activities, dedicated HPC research and contact point of international HPC infrastructures. IT4Innovations and CESNET form main pillars of the e-infrastructure in the Czech Republic and serve as the base layer for other research infrastructures in the Czech Republic. Thanks to IT4Innovations, the Czech Republic has access to state-of-the-art HPC infrastructure and expertise. As a member of prestigious pan-European e-infrastructure PRACE (Partnership for Advanced Computing in Europe), IT4Innovations also enables access for Czech scientists to all services provided by this e-infrastructure and opens opportunity for international collaboration. IT4Innovations has rich international network and actively contributes to the research in HPC related areas.

1) Future development: IT4Innovations operates supercomputers Anselm (Rpeak 94 TFLOPS) and Salomon (Rpeak 2000 TFLOPS) and runs state-of-the-art data centre providing ample space and power and unique hot water cooling with recuperation. In 2015, the Salomon supercomputer ranks as the largest Intel(r) Xeon Phi(tm) coprocessor-based cluster in Europe. IT4Innovations actively develops and widens its service portfolio, including its own HPC related research, to access new research areas and user groups. Further development of IT4Innovations presumes regular upgrades of the systems to continue excellent service to the users as well as extension and development of training and visualisation capacities of the centre.

2) Socio-economic impact: HPC is globally recognized as an important innovation enabler in research as well as in industry. Supercomputing simulations are often the only way to understand complex problems and to solve grand scientific challenges. The expertise and capacity of IT4Innovations supports multiple research and development areas having significant socio-economic impact, e.g. in flood prevention, crash tests, drug design, chemical catalysis and personalised medicine.

3) Study Program of Computational sciences focused on HPC: IT4Innovations has successfully opened Ph.D. Study Program of Computational sciences focused on HPC in October 2015 in order to grow the community of scientists able to deploy supercomputers for their research. New MSc. Study Program will also follow with its start planned for September 2016. Also a training in the field of HPC, parallel programming, HPC libraries and architectures is being offered to the Czech community of scientists for nearly two years. IT4Innovations has organized two seasonal schools and co-organized another two as a member of PRACE. IT4Innovations plans to continue in these type of activities because educational and training activities are an essential part of its strategy for the development of HPC in the Czech Republic.

4) Visualization and Virtual Reality Laboratory: IT4Innovations plans to build the Visualization and Virtual Reality Laboratory with an infrastructure designed for rendering and visualization of the 3D models and scenes using state-of-the-art projection and virtual reality technology. In particular the following activities could be performed: multimedia presentation, training and education activities, visualization of scientific experimental and simulation results, industrial products presentation, prototyping and many more.

VII. Application Scenarios

The ANTAREX project is driven by two industrial HPC applications chosen to address the self-adaptivity and scalability characteristics of two highly relevant scenarios towards the Exascale era.

a) Use Case 1: Computer Accelerated Drug Discovery: Computational discovery of new drugs is a compute-intensive activity that is critical to explore the huge space of chemicals with potential applicability as drugs. Typical problems include the prediction of properties of protein-ligand complexes (such as docking and affinity), and the verification of synthetic feasibility. These problems are massively parallel, but demonstrate unpredictable imbalances in the computational time, since the verification of each point in the solution space requires a widely varying time. Moreover, different tasks might be more efficient on different type of processors, especially in a heterogeneous system. Dynamic load balancing and task placement are critical for the efficient solution of such problems [27, 28].

b) Use Case 2: Self-Adaptive Navigation System: To solve the growing automotive traffic load, it is necessary to find the best utilization of an existing road network, under a variable workload. The basic idea is to provide contextual information from server side to traditional mobile navigation users and vice versa. The approach will help to overcome the major shortcomings of the currently available navigation systems exploiting synergies between server-side and client-side computation capabilities. The efficient operation of such a system depends strongly on balancing data collection, big data analysis and extreme computational power [29, 30].

Prototypes of these two use cases will be developed, integrated and validated in relevant and realistic environments to practically assess the benefits of the self-adaptive holistic approach, as well as the scalability of the proposed approach towards Exascale systems.

VIII. Conclusions

Exascale HPC systems will need the definition of new software stacks to fully exploit heterogeneity while meeting power efficiency requirements. The goal of ANTAREX is to provide a holistic system-wide adaptive approach for next generation HPC systems. Our long-term vision is to explore an innovative application programming paradigm and description methodology to decouple functional and extra-functional aspects of the application. The impact and benefits of such technology are far reaching, beyond traditional HPC domains.

References

[1] J. Curley, "HPC and Big Data," *Innovation*, vol. 12, no. 3, Jul. 2014.

[2] N. Rajovic, P. M. Carpenter, I. Gelado, N. Puzovic, A. Ramirez, and M. Valero, "Supercomputing with Commodity CPUs: Are

Mobile SoCs Ready for HPC?" in *Proc. Int'l Conf. on High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, pp. 40:1–40:12.

[3] M. Bohr, "A 30 Year Retrospective on Dennard's MOSFET Scaling Paper," *IEEE SSCS Newsletter*, vol. 12, no. 1, pp. 11–13, Winter 2007.

[4] G. Chrysos, "Intel® Xeon Phi™ Coprocessor-the Architecture," Intel Whitepaper, 2014.

[5] J. Floch, S. Hallsteinsen, E. Stav, F. Eliassen, K. Lund, and E. Gjorven, "Using architecture models for runtime adaptability," *IEEE Softw.*, vol. 23, no. 2, pp. 62–70, Mar. 2006.

[6] J. Irwin, G. Kickzales, J. Lamping, A. Mendhekar, C. Maeda, C. V. Lopes, and J.-M. Loingtier, "Aspect-oriented Programming," in *ECOOP'97 Ž014 Object-Oriented Programming*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1997, vol. 1241, pp. 220–242.

[7] T. Elrad, R. E. Filman, and A. Bader, "Aspect-oriented Programming: Introduction," *Communications of the ACM*, vol. 44, no. 10, pp. 29–32, 2001.

[8] J. M. P. Cardoso, T. Carvalho, J. G. F. Coutinho, W. Luk, R. Nobre, P. Diniz, and Z. Petrov, "LARA: An Aspect-oriented Programming Language for Embedded Systems," in *Proceedings of the 11th Annual International Conference on Aspect-oriented Software Development*. ACM, 2012, pp. 179–190.

[9] J. a. M. P. Cardoso, J. G. F. Coutinho, T. Carvalho, P. C. Diniz, Z. Petrov, W. Luk, and F. Gonćalves, "Performance-driven instrumentation and mapping strategies using the LARA aspect-oriented programming approach," *Software: Practice and Experience*, Dec. 2014. [Online]. Available: http://dx.doi.org/10.1002/spe.2301

[10] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold, "An Overview of AspectJ," in *ECOOP 2001 Ž014 Object-Oriented Programming*. Springer, 2001, pp. 327–354.

[11] O. Spinczyk, A. Gal, and W. Schröder-Preikschat, "AspectC++: An Aspect-oriented Extension to the C++ Programming Language," in *Proceedings of the Fortieth International Conference on Tools Pacific: Objects for Internet, Mobile and Embedded Applications*, 2002, pp. 53–60.

[12] J. M. Cardoso, T. Carvalho, J. G. Coutinho, R. Nobre, R. Nane, P. C. Diniz, Z. Petrov, W. Luk, and K. Bertels, "Controlling a complete hardware synthesis toolchain with LARA aspects," *Microprocessors and Microsystems*, vol. 37, no. 8, pp. 1073–1089, 2013.

[13] R. Nobre, L. G. Martins, and J. M. Cardoso, "Use of Previously Acquired Positioning of Optimizations for Phase Ordering Exploration," in *Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems*. ACM, 2015, pp. 58–67.

[14] 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.

[15] 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.

[16] F. Bodin, T. Kisuki, P. Knijnenburg, M. O'Boyle, and E. Rohou, "Iterative compilation in a non-linear optimisation space," 1998.

[17] A. Cohen and E. Rohou, "Processor virtualization and split compilation for heterogeneous multicore embedded systems," in *Proc. 47th Design Automation Conference*. ACM, 2010, pp. 102–107.

[18] D. Gadioli *et al.*, "Opencl application auto-tuning and run-time resource management for multi-core platforms," in *IEEE Int'l Symp. Parallel and Distributed Processing with Applications (ISPA)*, Aug 2014, pp. 127–133.

[19] B. Subramaniam, W. Saunders, T. Scogland, and W.-c. Feng, "Trends in Energy-Efficient Computing: A Perspective from the Green500," in *4th Int'l Green Computing Conference*, June 2013.

[20] F. Fraternali, A. Bartolini, C. Cavazzoni, G. Tecchiolli, and L. Benini, "Quantifying the Impact of Variability on the Energy Efficiency for a Next-generation Ultra-green Supercomputer," in *Proc. 2014 Int'l Symp. on Low Power Electronics and Design*. ACM, 2014, pp. 295–298.

[21] A. Auweter, A. Bode, M. Brehm, L. Brochard, N. Hammer, H. Huber, R. Panda, F. Thomas, and T. Wilde, "A Case Study of Energy Aware Scheduling on SuperMUC," in *Supercomputing*. Springer, 2014, vol. 8488, pp. 394–409.

[22] A. Borghesi, C. Conficoni, M. Lombardi, and A. Bartolini, "MS3: a Mediterranean-Stile Job Scheduler for Supercomputers - do less when it's too hot!" in *2015 Int'l Conf. on High Perf. Comp. & Simulation*. IEEE, 2015.

[23] C. Cavazzoni, "EURORA: A European Architecture Toward Exascale," in *Proceedings of the Future HPC Systems: The Challenges of Power-Constrained Performance*, ser. FutureHPC '12. New York, NY, USA: ACM, 2012, pp. 1:1–1:4. [Online]. Available: http://doi.acm.org/10.1145/2322156.2322157

[24] A. Bartolini, M. Cacciari, C. Cavazzoni, G. Tecchiolli, and L. Benini, "Unveiling eurora - thermal and power characterization of the most energy-efficient supercomputer in the world," in *Proceedings of the Conference on Design, Automation & Test in Europe*, ser. DATE '14. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2014, pp. 277:1–277:6. [Online]. Available: http://dl.acm.org/citation.cfm?id=2616606.2617013

[25] "Mont-Blanc (FP7-ICT-2011-7 European project): European scalable and power efficient HPC platform based on lowpower embedded technology," 2011.

[26] "Mont-Blanc 2 (FP7-ICT-2013-10 European project), European scalable and power efficient HPC platform based on low-power embedded technology," 2013.

[27] C. Beato, A. R. Beccari, C. Cavazzoni, S. Lorenzi, and G. Costantino, "Use of experimental design to optimize docking performance: The case of ligendock, the docking module of ligen, a new de novo design program," *J. Chem. Inf. Model.*, vol. 53, no. 6, pp. 1503–1517, 2013.

[28] A. R. Beccari, C. Cavazzoni, C. Beato, and G. Costantino, "LiGen: A High Performance Workflow for Chemistry Driven de Novo Design," *J. Chem. Inf. Model.*, vol. 53, no. 6, pp. 1518–1527, 2013.

[29] R. Tomis, J. Martinovič, K. Slaninová, L. Rapant, and I. Vondrák, "Time-dependent route planning for the highways in the czech republic," in *Proc. 14th Int'l Conf. on Computer Information Systems and Industrial Management, CISIM 2015*, 2015.

[30] D. Fedorčák, T. Kocyan, M. Hájek, D. Szturcová, and J. Martinovič, "viaRODOS: Monitoring and Visualisation of Current Traffic Situation on Highways," in *Computer Information Systems and Industrial Management*. Springer, 2014, vol. 8838, pp. 290–300.