EXTRA: Towards the Exploitation of eXascale Technology for Reconfigurable Architectures

Dirk Stroobandt*, Ana Lucia Varbanescu¶, Cătălin Bogdan Ciobanu¶, Muhammed Al Kadi‖,
Andreas Brokalakis††, George Charitopoulos†, Tim Todman‡, Xinyu Niu‡, Dionisios Pnevmatikatos†,
Amit Kulkarni*, Elias Vansteenkiste*, Wayne Luk‡, Marco D. Santambrogio§, Donatella Sciuto§,
Michael Huebner‖, Tobias Becker**, Georgi Gaydadjiev**, Antonis Nikitakis††, Alex J. W. Thom‡‡

*Ghent University, Belgium, †Telecommunications Systems Institute, Greece
‡Imperial College London, UK, §Politecnico di Milano, Italy
¶UvA, the Netherlands, ‖Ruhr-Universität Bochum, Germany
**Maxeler, UK, ††Synelixis, Greece, ‡‡University of Cambridge, UK

Abstract*—To handle the stringent performance requirements of future exascale-class applications, High Performance Computing (HPC) systems need ultra-efficient heterogeneous compute nodes. To reduce power and increase performance, such compute nodes will require hardware accelerators with a high degree of specialization. Ideally, dynamic reconfiguration will be an intrinsic feature, so that specific HPC application features can be optimally accelerated, even if they regularly change over time. In the EXTRA project, we create a new and flexible exploration platform for developing reconfigurable architectures, design tools and HPC applications with run-time reconfiguration built-in as a core fundamental feature instead of an add-on. EXTRA covers the entire stack from architecture up to the application, focusing on the fundamental building blocks for run-time reconfigurable exascale HPC systems: new chip architectures with very low reconfiguration overhead, new tools that truly take reconfiguration as a central design concept, and applications that are tuned to maximally benefit from the proposed run-time reconfiguration techniques. Ultimately, this open platform will improve Europe's competitive advantage and leadership in the field.

I. Introduction

As power and energy consumption of HPC systems skyrockets, it becomes fundamental to execute each compute task with the best energy efficiency. Powering a data center has become problematic and managing the heat dissipated by modern high-performance systems is challenging while it contributes to more than 50% of the energy budget spent [1]. To achieve optimum energy efficiency, heterogeneous systems that combine standard high-performance general-purpose processors with customized application-specific accelerators have to be considered and are being introduced in datacenters. However, although it is known that an optimized hardware implementation of a certain compute task can achieve the highest performance at the lowest energy consumption, it is both technically and economically infeasible to include non-programmable accelerators for all possible applications (or computational kernels) that run on a typical HPC system.

As such, programmable solutions are typically considered, among which GPUs are the most popular, as it has been shown that they can deliver increased performance and power efficiency compared to CPUs [2]. The overall GPU power consumption is still very high, though, reaching 300W per card and thus limiting their deployment in large-scale HPC systems. Reconfigurable devices such as Field Programmable Gate Arrays (FPGAs) are a valid alternative, since they can provide hardware-level performance and energy efficiency by creating customized datapaths, while retaining the flexibility of a programmable device, whose functionality can be changed post-deployment. The study in [2] shows systematic performance (up to 1.55X) and energy benefits (2.9X to 3.9X) for FPGA implementations for Barrier Option Pricing, Particle Filter, and Reverse-Time Migration when compared to GPUs.

This combination of flexibility and high computational efficiency per watt is gaining momentum in the industry with multiple research and commercial systems being deployed and significant results published. One representative example of such systems is represented by the Maxeler systems. Very efficient implementations [3] have been obtained on Maxeler hardware while accelerating streaming applications. Micron (former Convey) HC-1 and HC-2 combine Intel Xeon CPUs with FPGAs [4]. Particle physics experiments at CERN deal with the high throughput requirements of real-time sensor data and rely almost exclusively on FPGAs for their speed, density, computational power, flexibility, and intrinsic radiation tolerance [5]. IBM has recently announced [6] its strategy for FPGA-enabled acceleration within its POWER8 and OpenPower initiatives. Microsoft has also recently adopted the dataflow computing approach, programming their own FPGAs [7] to accelerate various BING search engine algorithms. Intel is also expected to produce data center chips combining Xeon CPUs and Altera FPGAs in the near future [8].

There are though several obstacles that prevent those run-time reconfigurable systems from becoming mainstream. The following are identified as the most significant ones:

- The tools required for programming such run-time reconfigurable systems still face substantial reconfiguration overheads, which prevent them from being used for large-scale deployment;
- The run-time reconfigurable systems have to use existing FPGA architectures, which are not specifically built with

run-time reconfiguration in mind, and therefore lack in efficiency for maximally exploiting possible run-time reconfiguration benefits;
- For newly proposed reconfigurable architectures, the optimal granularity of the reconfiguration infrastructure is still undecided. A low-level reconfiguration infrastructure (such as in current FPGAs) has higher flexibility but larger reconfiguration time, compared to a coarser granularity;
- HPC applications are not optimized for exploiting the available reconfigurability. This is partly because current tool chains do not maximize programmability and designer productivity.

In the European project EXTRA (Exploiting eXascale Technology with Reconfigurable Architectures), we are developing an integrated environment for developing and programming reconfigurable architectures with built-in run-time reconfiguration. The idea of this new and flexible exploration platform is to enable the joint optimization of architecture, tools, applications, and reconfiguration technology in order to prepare for the necessary HPC hardware nodes of the future.

The remainder of this paper is organized as follows: the research objectives are described in Section II. Section III presents our approach and Section IV describes the reconfigurable platform that we have in mind. In Section V we focus on the main challenges and we conclude in Section VI.

II. Research Objectives

The main objective of the EXTRA project is to develop an open source research platform for continued research on reconfiguration architectures and tools. The goal is to find architectures and tools that match the next-generation HPC application requirements within a virtual tool environment. Versatile Place and Route (VPR, now part of the VTR framework [9]) is a common platform for the exploration and improvement of design tools for FPGA implementation on typical FPGA architectures. We want to provide a similar platform for run-time reconfiguration that will enable increased research efforts on run-time reconfiguration in Europe.

Because the exploitation of system reconfigurability is relatively new, more research is needed on the optimal HPC architectures that can maximally benefit from reconfiguration, on improvements in the tools to exploit reconfiguration while designing high performance and power-efficient implementations, and on the application optimizations. Therefore, we identify three Key Objectives (KO) for the success of EXTRA.

- KO1 We target the development and promotion of an open reconfigurable technology exploration platform that combines a reconfigurable architecture description with reconfigurable design tools and thus allows to evaluate and optimize reconfigurable applications.
- KO2 We aim to make significant contributions to the development of reconfigurable architectures, reconfigurable tools, and the optimization of reconfigurable HPC applications.
- KO3 We will validate both the platform and our proposed improvements using the EXTRA ecosystem to implement three HPC applications, with the aim to improve performance, area and power efficiency.

To achieve these key objectives, we identified six major technical objectives that must be achieved.

1: Enable a co-design approach for developing reconfigurable HPC architectures, tools and applications.
The co-design approach will be intrinsic to our open research platform to allow individual contributions to be tested using the complete chain from device up to the application. This is the first time that such a holistic approach is proposed and it requires that all parts of the platform allow co-design, from the architecture description over tools to the application implementation. The HW/SW partitioning may be automatic or semi-automatic, where the tools guide the user by providing profiling data.

2: Include reconfigurability as an explicit design concept in the design of future HPC systems.
Although VPR offers the ability to describe a typical FPGA architecture and offers an open source tool framework for placement and routing tools, it does not include reconfiguration as a specific design option and the architecture descriptions are limited to classical FPGA architectures. We intend to build a conceptually similar infrastructure, but with a lot more flexibility in the architecture choices and a focus on reconfigurability as a design concept.

Both Xilinx and Altera have recently announced that they see a large market potential for their FPGAs in data centers. This enforces our belief that reconfigurable architectures will be essential to the success of future exascale systems. But as reconfigurable devices get larger and more complex, reconfiguring the entire device takes longer and requires more energy; so partial reconfigurability is becoming increasingly important. Both Altera and Xilinx therefore have support for designs with partial run-time reconfiguration in their current devices (Stratix-V [10] and Virtex-7 [11]). In this context, EXTRA will explore (partial) reconfigurability as a specific design feature in future HPC systems, aiming to enable it fully in new reconfigurable architectures, new design tools, and re-engineered applications, optimized for reconfigurability.

3: Speed up the reconfiguration process through novel reconfiguration approaches for processing, BRAMs, special blocks and interconnections in a coarse-grain reconfiguration architecture.
The duration of the reconfiguration process is one of the important bottlenecks in the current reconfigurable systems. By reducing the time needed to reconfigure the hardware, more tasks can be accelerated, significantly reducing the overhead of programming new kernels into the hardware. In order to achieve this goal, EXTRA will consider all the relevant components: processing, BRAMs, specialized blocks and interconnect. Furthermore, the development of optimized hardware structures for HPC workloads allow EXTRA to

reduce the need to reconfigure, as a certain level of adaptability is built-in at design-time.

4: Provide Just-In-Time (JIT) synthesis methods for reconfiguration on the fly, based on application requirements. For specific applications, optimized configurations are depending on the input values and cannot be calculated beforehand and stored until they are needed. For such applications, on the fly configuration generation support is desirable, enhancing run-time flexibility. Thus, an adequate configuration scheme and configuration infrastructure for the EXTRA hardware has to be developed to meet the reconfiguration performance (or frequency) requirements. In order to generate this bitstream, a tool set - including synthesis, mapping, placement and routing under hard timing constraint - must be developed.

5: Improve the HPC applications under consideration. To determine the impact of reconfiguration on HPC applications, we will introduce runtime reconfiguration to three selected applications with high computational requirements.

The first application is a Financial Option Pricing application which computes Value at Risk (VaR) for a portfolio of Asian options under a collection of possible future scenarios. At its center, it repeatedly performs option pricing according to Curran's approximation model. Value at Risk is a quantitative measure of the risk of investments. Each scenario involves the pre-computation of volatility surfaces and discount curves through interpolation. These computations are only performed once per scenario and currently do not contribute a significant amount of compute time. If, however, the option pricing is significantly sped up then these pre-computations may become a relevant component in the computation time of each scenario. It could therefore become relevant to accelerate these pre-computations as well. When accelerating pre-computations, one could develop a specific stand-alone accelerator for the pre-computations which is executed once per scenario loop. This approach might involve reconfiguration as the pre-computation accelerator could be loaded just to perform the pre-computation for one loop, and it is subsequently unloaded to perform option pricing within the loop. In this scenario we have to consider that the reconfiguration overhead does not outweigh the benefit of using an accelerator.

Two other applications will be considered for testing the impact of reconfiguration on HPC workloads: a highly parallel Retinal Image Segmentation Application (vessel segmentation in large datasets) and Quantum Monte Carlo (QMC) methods (both Variational and Diffusion). The retinal vessel segmentation application refers to the extraction of the vessel structure from the background in fundus images. Vessel segmentation enables the extraction of morphological attributes of retinal blood vessels, such as length, width and branching pattern. Run-time reconfiguration is useful as the algorithm features two different types of filter modules: the matched filter and the texture filter one. In this respect an accelerator could be possible to utilize a single type at any given time, and using reconfiguration to load the other one. Only a precise design space exploration could reveal if the reconfiguration approach is more efficient than a fixed core approach.

The Diffusion Monte Carlo application seeks the electronic energy of a system built up of atomic nuclei and electrons, and it calculates this from the dynamics of objects called walkers (essentially a specific configuration specifying the position of each electron in the system), which are propagated (by changing - with a guided random walk - the positions of the electrons) such that the true quantum mechanical energy of the system can be calculated.

We note that although parallel and/or hybrid implementations (using CPUs and GPUs) do exist for the aforementioned applications (e.g. [12], [13], [14] for Quantum Monte Carlo methods), the algorithms need to be reconsidered in order to fully exploit the power of reconfigurable architectures.

6: Suggest new reconfiguration features for future technologies.
We will investigate and suggest practical improvements and necessary features to improve the technology constraints in future reconfigurable systems. We will focus on possible improvements in reconfiguration infrastructure, achieving tighter coupling with the compute cores, and providing hardware support for monitoring and emergency situation management. These architectural changes will be investigated and simulated to estimate their impact on actual implementations. However, the actual physical hardware implementation of such suggestions of architectural improvements will be left for future projects.

III. Main Approach

The main assumption in the EXTRA project is that system reconfigurability will be a key concept in future HPC systems. In order to develop reconfigurable hardware HPC systems, we need (i) to design completely new system architectures that are inherently reconfigurable, (ii) to develop new tools that enable efficient reconfiguration, and (iii) to optimize applications to maximally exploit this novel concept of reconfigurability.

The EXTRA project tackles all three issues and proposes initial architectures, tools and applications that benefit from reconfigurability. We focus on building the necessary infrastructure for enabling continued research towards reconfigurable HPC systems for exascale applications while, at the same time, presenting initial solutions that prove that our reconfigurability concept enables more efficient systems and application implementations. It is important to note that reconfiguration can only bring the necessary power efficiency to HPC systems, without excessive resource requirements, if the reconfiguration can be done while the application is running (run-time reconfiguration) and that the reconfiguration overhead should be significantly smaller than what current systems can offer. Hence, EXTRA will devote significant effort to minimizing the reconfiguration overhead.

The overall approach of the EXTRA project is visually explained in Figure 1. We are currently investigating how run-time reconfiguration can benefit exascale HPC applications. Based on the application requirements, we specified the main

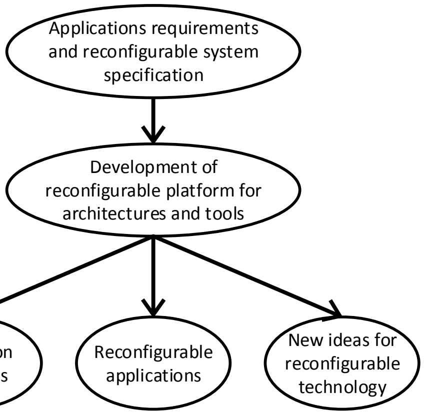
Applications requirements and reconfigurable system specification
Development of reconfigurable platform for architectures and tools
Research on novel tools
Reconfigurable applications
New ideas for reconfigurable technology

Fig. 1. Conceptual overview of the EXTRA project

system requirements for maximally exploiting the benefits of run-time reconfiguration. This is the basis for the further work in the project.

The main focus of the EXTRA project is the development of an open source exploration platform that allows the joint investigation of reconfigurable architectures, tools, and applications. The concept is that this open platform will enable many researchers to explore novel reconfigurable architectures independently from current commercial vendor solutions. At the same time, the platform provides several hooks within the tool flow to enable tool developers to investigate new tool metrics and propose new tools for designing HPC applications on chosen reconfigurable architectures. These tools will also inherently have reconfigurability included, which is not the case today. Finally, the combination of available reconfigurable architecture descriptions and tools to develop implementations on these architectures provides application developers with an easy to use platform for optimizing their applications. Again, run-time reconfiguration is available everywhere and allows application developers to optimize their applications for it and to evaluate the benefits for their applications using the platform. The open exploration platform for architectures, tools and applications will also allow the EXTRA consortium partners to make significant contributions in JIT synthesis tools for reconfigurable architectures, to efficiently optimize applications for maximally exploiting reconfiguration and evaluating their performance, and to suggest novel reconfiguration technology concepts to improve the efficiency of the reconfiguration within the architectures (bottom part of Figure 1).

We will demonstrate our open source exploration platform and make it available to other researchers in order to create a strong momentum towards research in reconfigurable HPC systems, architectures, tools and applications. Also, the benefits of reconfiguration will be demonstrated by actual implementations of the three EXTRA applications in finite elements, medical imaging, and scientific computing applications on modern commercial reconfigurable devices.

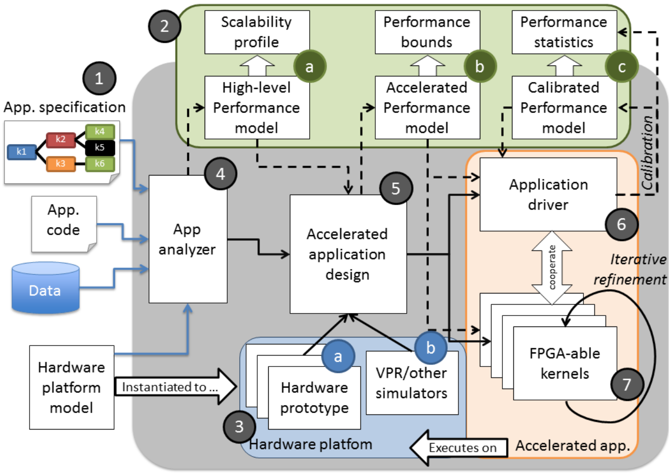

Fig. 2. Description of the EXTRA project platform

IV. Reconfiguration Platform

In this section we describe the open research platform we have in mind. Crucial is that it includes reconfiguration possibilities in all its aspects. First, we describe the high-level platform and then we focus on the low-level platform details and reconfiguration aspects.

A. High-Level Platform Overview

Figure 2 presents a schematic of the envisioned EXTRA platform. The main components of the platform are numbered from 1 to 7 and are further described in the following subsections.

1) The input data (1): This component is meant to set the whole system in motion. Specifically, we expect the following three types of input:

- Most importantly, the application itself. Typically, applications will be provided by means of code. Code generation for the full application is beyond the scope of EXTRA. Instead, we focus on extracting the application parts that are promising for performance enhancement, and build the right infrastructure to support a productive, high performance implementation and understand its performance bounds. Thus, additional application specification - i.e., code pragmas to indicate potential acceleration candidates, multiple code versions for different types of devices, performance requirements, and/or DAG-like specifications of the code - are useful and can be used to further improve the effectiveness of the EXTRA approach.
- The input data needs to be specified, either as a (collection of) datasets, or as a generative model that can be used to obtain realistic datasets.
- Finally a hardware platform model is required. At this stage, this model is assumed to be a high-level specification of the platform that will be used for the execution.

> For example, ”1 single node, with 1 CPU and 1 FPGA accelerator” is a platform model with very different requirements than ”HPC cluster of N nodes, each accelerated with M accelerators”. This model is also used as an input to the tool chain, which will take offloading decisions based on the available resources, leading to a modular application model that fits the hardware model at conceptual level. The low-level details of the platform are described in the next subsection.

In this project, we focus mainly on three application classes as proof-of-concepts: a Financial Option Pricing application, a Retinal Image Segmentation problem (vessel segmentation in large datasets), and Quantum Monte Carlo (QMC) methods (both Variational and Diffusion). Our research will identify the most suitable combination of input parameters to allow the platform to function at its best.

2) The Tool chain (4+5): Hidden behind only two boxes, the tool chain is the critical path, and the most technically challenging part of EXTRA. We picture it here as a two-stage analysis chain: the application analyzer and the accelerated application design. In these two stages, we must combine the knowledge about the application characteristics, the platform model, and the input data and obtain an annotated DAG-like description of the application which includes: data dependencies, scheduling dependencies, profiling information, performance requirements, and hardware matching for each module in the DAG. Understanding how to measure all these features is a challenge in itself. This is why we will limit ourselves to the applications at hand - the three benchmarks in the project - and leave the generalization of this analysis for the future. However, we believe that the in-depth analysis of the benchmark applications will provide us insights on certain features that could be used as indicators for the successful acceleration of similar applications.

3) The Performance Modeling (2): An important component of EXTRA is its focus not only on high performance, but also on scalability. Targeting exascale computing is ambitious and it will require performance modeling techniques to derive reliable scalability estimates for the applications and the hardware. The performance modeling component has three stages: (a) a generic high-level performance model, where the DAG of the application is analyzed and a high-level performance profile is built. Based on high-level tools such as Amdahl's law and the Roofline model, this model will indicate the potential for scalability and performance of the accelerated application. Next, in stage (b), once the hardware resources are clarified and the application is split into a driver and its kernels, a better understanding of the performance bounds can be obtained. This model is a refined version of the one in stage (a), where the added information will provide more information on the limitation that the platform imposes (if any) on the scalability of the application. Finally, (c) is the stage where the full performance model is obtained. This model is calibrated to the actual hardware that is being used, and it provides an accurate prediction of the performance to be obtained. Based on the current state of the art, we assume this model would be based on statistical models, but analytical elements might also be included for certain hardware platforms.

Thus, performance modeling in EXTRA plays a double role: it can be used as performance indication for the end-user (*The Tool chain*), or as scalability analysis for determining, with reasonable accuracy, when the use of exascale machines is actually beneficial (*The Performance Modeling*).

4) The Hardware platform (3): For the hardware platform, we envision a selection between actual hardware machines and accurate simulators (see also Section IV-B). Since our target is an open source platform, that can explore the opportunities that reconfigurable computing can offer for large scale, performance greedy applications, we cannot be limited to a single hardware alternative. Our EXTRA project will include in its prototype at least one alternative from each space - i.e. a simulator and a real hardware platform. Further exploration of other alternatives is left for future plans or additional development of EXTRA-followers or interested parties (application owners, performance engineers, M.Sc. students and visiting scholars).

5) The accelerated application (6+7): Finally, the prototype of the accelerated application will be available. We expect this to be a collection of kernels with different optimization techniques (effectively, different versions of multiple kernels to be benchmarked and profiled) and an application driver, which will implement the DAG structure. One important challenge here is to isolate the driver from the application itself, preferably in the form of a rudimentary run-time system that will build on the FASTER runtime system [15]. While this solution seems theoretically feasible, the technical challenges cannot be ignored.

6) Verification and debugging: Reconfigurable platforms can complicate the problems of verifying that a design is performing correctly and debugging when designs fail. To address these problems, we will extend our work on in-circuit assertions, which check if circuit properties meet design expectations. In-circuit assertions can run at the same rate as the design under test and can check not just Boolean conditions but also statistics of internal signals. We will build on our work on adding post-hoc assertions [16] to existing designs, allowing assertions and other monitoring and debugging circuits to be added after implementation. Such circuits could also check non-functional properties such as power consumption. At the same time, we will also implement a new hardware debugging technique, implemented by using run-time reconfiguration. This technique introduces debugging within the design environment, instead of as a separate entity. it will deliver full observability of internal hardware signals without the need to redesign the circuit, thus bringing hardware debugging flexibility to the level of software debugging flexibility.

B. Low-Level Platform Details

Current FPGAs are still offering partial reconfiguration as an extra feature but their architectures have not been optimized for this purpose. The traditional tool chain implies fixing the system modules at design time and mapping them offline to the FPGA logic. EXTRA has to introduce its own platform and tool chain to meet the targeted challenges. Duration and flexibility of partial modifications have to be placed next to other traditional metrics like performance or power consumption when designing the EXTRA platform. A different paradigm that includes JIT synthesis, online bitstream generation and partial reconfiguration should be introduced. In contrast to traditional FPGAs, future architectures have to relax many geometrical limitations on the reconfigurable areas. The realization of new system modules will be done at runtime.

A first requirement for this runtime reconfiguration is that the reconfiguration infrastructure of FPGAs should be improved. Our goal is to investigate the most optimal reconfiguration infrastructure, according to the target platform architecture and device. We will investigate reconfiguration memory models and trade-off size versus speed using SPICE simulation models. We search for an alternative to the frame-based reconfiguration model that is current practice in Xilinx components. Issues under research will be the granularity of the reconfiguration infrastructure (bit-level or function-level), the amount of parallel paths available to the reconfiguration infrastructure, etc. Our goal is to quickly generate new configurations within the device, while avoiding the time overhead of traditional synthesis and the area overhead of conventional multi-context FPGAs.

In order to improve the support of dynamic data access operations, which widely exist in applications and depend on runtime values to define operations to execute, we aim to explore new architectures and tools based on the EURECA (Effective Utilities for Run-timE Configuration Adaptation) technique [17]. This allows hardware circuits to be reconfigured within one nanosecond, in contrast to the one microsecond minimum reconfiguration time in latest devices. Such rapid reconfiguration enables hardware designs to identify operations to execute, implement customized circuits, and execute operations, all within a clock cycle.

Not only the reconfiguration process, but also synthesis, placement, routing and bitstream generation of new components have to be accomplished within time limits. The new tool chain for the low-level reconfigurable hardware platform will be based on VTR, the state of the art academic tool for realizing RTL designs on modern island-style FPGA architectures [9]. To extend the exploration space, a heterogeneous platform consisting of traditional FPGA logic next to Virtual CGRAs (Coarse Grained Reconfigurable Arrays) will be investigated. This enables the platform to meet extreme requirements that can not be intrinsically fulfilled by one of the two components. The running applications will be mapped partially or completely to one of the platform parts. The FPGA architecture will be modified to relax the execution time of JIT synthesis and make it an online task. The used CAD algorithms, like simulated annealing for placement and PathFinder for routing, have to be modified and intensively parallelized to meet an acceptable compromise between runtime and quality of results. In addition, the platform will enable exploring the effects of different granularities for basic FPGA building blocks on the time needed for reconfiguration and JIT synthesis.

V. Challenges

The overall EXTRA approach consists of three main parts.

1) A thorough investigation on run-time reconfiguration requirements in exascale HPC applications and the specification of system requirements for maximally exploiting the benefits of run-time reconfiguration. *The main research challenges* here are (1) to analyze and characterize the workloads of three HPC application domains, (2) to specify metrics and validation strategies, and finally, (3) to integrate and demonstrate the results, showing that the initial requirements are met.
2) The development of an open source reconfiguration platform that allows the joint investigation of reconfigurable architectures, tools, and applications. We focus on the employment of FPGAs as custom hardware accelerators to speed up the 'hot' portion of target applications. Productivity is guaranteed due to the adoption of broadly known programming languages such as C. The open platform development is done in two concurrent directions with tight interaction between them: the *reconfigurable architecture* itself and *the design tools*. *The main challenges for designing* the platform are (1) the definition of optimal granularity, based on the analysis of state-of-the-art reconfigurable architectures, (2) the design and development of the tools and (micro)architectural support for optimal interaction between the CPU and the FPGA accelerators, (3) the design space exploration for reconfigurable HPC applications, and (4) the design of the architectural and circuit models for evaluating the feasibility and potential of the proposed platform. The framework for the design tools that use the reconfigurability of the architecture for implementing applications must keep pace with the platform development as well. Finally, the actual integration of the reconfigurable processing units in an exascale system will pose significant challenges.
3) Significant contributions in JIT synthesis tools for reconfigurable architectures, in efficient optimization of HPC applications, and guidelines for future reconfiguration technology. *The challenges* here include (1) the development of tools and methods to enable the JIT synthesis of configuration for the reconfigurable hardware, (2) systematic analysis, selection, and optimization of applications' functions and structures that can be optimized

through reconfiguration, and (3) the impact evaluation of the new optimization techniques. Finally, we aim to develop techniques and guidelines that improve the potential of future reconfigurable technology by learning from the past mistakes, i.e., based on all the feedback obtained along the project. The main challenges here are (1) to collect this feedback coherently and comprehensively, and (2) to transform these issues into actionable points with potential impact on the future of reconfigurable HPC.

VI. Conclusion

In conclusion, this project focuses on the fundamental building blocks for run-time reconfigurable exascale HPC systems: new reconfigurable architectures with very low reconfiguration overhead, new tools that truly take reconfiguration as a design concept, and applications that are tuned to maximally exploit run-time reconfiguration techniques. The developed exploration platform ensures a smooth and efficient co-design of architecture, tools and applications.

Acknowledgments

The EXTRA project runs from September 2015 till August 2018 and receives funding from the EU Horizon 2020 research and innovation programme under grant No 671653.

References

[1] J. Liu *et al.*, “Project Genome: Wireless Sensor Network for Data Center Cooling,” *The Architecture Journal*, December 2008. [Online]. Available: research.microsoft.com/apps/pubs/default.aspx?id=78813

[2] X. Niu *et al.*, “Automating elimination of idle functions by run-time reconfiguration,” in *FCCM 2013*, April 2013, pp. 97–104.

[3] C. Tomas *et al.*, “Acceleration of the Anisotropic PSPI Imaging Algorithm with Dataflow Engines,” in *82nd Annual Meeting and Int. Exposition of the Society of Exploration Geophysics-SEG*, 2012. [Online]. Available: publications.crs4.it/pubdocs/2012/TCOPTSB12

[4] T. Brewer, “Instruction Set Innovations for the Convey HC-1 Computer,” *IEEE Micro*, vol. 30, no. 2, pp. 70–79, 2010.

[5] L. Musa, “FPGAS in high energy physics experiments at CERN,” in *FPL 2008*, Sept 2008, pp. 2–2.

[6] T. P. Morgan, “IBM Forging Bigger Power8 Systems, Adding FPGA Acceleration,” Jul. 2014. [Online]. Available: www.enterprisetech.com/2014/07/28/ibm-forging-bigger-power8-systems-adding-fpga-acceleration/

[7] J. Clark, “Microsoft ’Catapults’ geriatric Moore’s Law from CERTAIN DEATH: FPGAs DOUBLE data center throughput despite puny power pump-up, we’re told,” Jun. 2014. [Online]. Available: www.theregister.co.uk/2014/06/16/microsoft_catapult_fpgas/

[8] T. P. Morgan, “How Intel is Hedging on the Future of Compute with Altera Buy,” Jun. 2014. [Online]. Available: www.theplatform.net/2015/06/01/how-intel-is-hedging-on-the-future-of-compute-with-altera-buy/

[9] J. Luu *et al.*, “VTR 7.0: Next Generation Architecture and CAD System for FPGAs,” *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 2, pp. 6:1–6:30, Jul. 2014.

[10] “Increasing Design Functionality with Partial and Dynamic Reconfiguration in 28-nm FPGAs, Altera White Paper, WP-01137-1.0,” Jul. 2010.

[11] B. Przybus, “Xilinx Redefines Power, Performance, and Design Productivity with Three New 28 nm FPGA Families: Virtex-7, Kintex-7, and Artix-7 Devices, Xilinx White Paper WP373 (v1.0),” Jun. 2010.

[12] A. G. Anderson, W. A. Goddard, III, and P. Schröder, “Quantum Monte Carlo on graphical processing units ,” *Comput. Phys. Comm.*, vol. 177, no. 3, pp. 298 – 306, 2007.

[13] K. Esler, J. Kim, D. Ceperley, and L. Shulenburger, “Accelerating Quantum Monte Carlo Simulations of Real Materials on GPU Clusters,” *Computing in Science Engineering*, vol. 14, no. 1, pp. 40–51, Jan 2012.

[14] Y. Lutsyshyn, “Fast quantum Monte Carlo on a GPU,” *Comput. Phys. Comm.*, vol. 187, pp. 162 – 174, 2015.

[15] D. Pnevmatikatos, K. Papadimitriou, T. Becker, P. Bhm, A. Brokalakis, K. Bruneel, C. Ciobanu, T. Davidson, G. Gaydadjiev, K. Heyse, W. Luk, X. Niu, I. Papaefstathiou, D. Pau, O. Pell, C. Pilato, M. Santambrogio, D. Sciuto, D. Stroobandt, T. Todman, and E. Vansteenkiste, “Faster : facilitating analysis and synthesis technologies for effective reconfiguration,” *MICROPROCESSORS AND MICROSYSTEMS*, vol. 39, no. 4-5, pp. 321–338, 2014.

[16] E. Hung, T. Todman, and W. Luk, “Transparent insertion of latency-oblivious logic onto FPGAs,” in *FPL 2014*. IEEE, 2014, pp. 1–8.

[17] X. Niu, W. Luk, and Y. Wang, “EURECA: on-chip configuration generation for effective dynamic data access,” in *Proceedings of FPGA 2015 ACM/SIGDA*, 2015, pp. 74–83.