An Open Reconfigurable Research Platform as Stepping Stone to Exascale High-Performance Computing

Dirk Stroobandt*, Cătălin Bogdan Ciobanu¶, Marco D. Santambrogio§, Gabriël Figueiredo‡,
Andreas Brokalakis††, Dionisios Pnevmatikatos†, Michael Huebner‖, Tobias Becker**, 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 and power requirements of future exascale-class applications, High Performance Computing (HPC) systems need ultra-efficient heterogeneous compute nodes and 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. 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. Our project proposes an open research platform that 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 enable groundbreaking research towards new exascale computing platforms.**

I. Introduction

As power and energy consumption of HPC systems skyrocket, each compute task has to be executed with the best energy efficiency. Powering a data center 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 are being introduced in datacenters. 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. One representative example of such systems is the Maxeler system. Very efficient implementations [3] have been obtained on Maxeler hardware while accelerating streaming applications. Micron (formerly 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 in their project called Catapult. Intel bought FPGA device company Altera and is also expected to produce datacenter chips combining Xeon CPUs and Altera FPGAs [8].

Both Xilinx and Altera (now Intel) see a large market potential for their FPGAs in datacenters. 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 par-

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

tial run-time reconfiguration in their current devices (Stratix-V [9] and Virtex-7 [10]). In this context, the EXTRA project 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.

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 prevents them from being used for large-scale deployment;
- The run-time reconfigurable systems 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.

To address all these problems, further research is needed on improved tools that inherently exploit the FPGA's reconfigurability, on applications that can truly benefit from the reconfigurability, and also on new reconfigurable architectures that can implement reconfiguration more efficiently. As investigations in one domain can not be tested and evaluated without using the results in the other domains, we need a platform that allows research in one domain without having to also implement the other ones. In the European project EXTRA (Exploiting eXascale Technology with Reconfigurable Architectures), we are developing such 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.

We have identified three Key Objectives (KO) for the success of the EXTRA project.

- 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,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.

The remainder of this paper is organized as follows: Section II presents our main approach and the applications we are targeting. In Section III we describe the reconfigurable platform that we are developing with a focus on the tools platform. We conclude in Section IV.

II. 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, the EXTRA project will devote significant effort to minimizing the reconfiguration overhead.

The overall approach of the EXTRA project is visually explained in Figure 1. Our first challenge is to thoroughly investigate run-time reconfiguration requirements in exascale HPC applications and the specification of system requirements for maximally exploiting the benefits of run-time reconfiguration. This includes (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. 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

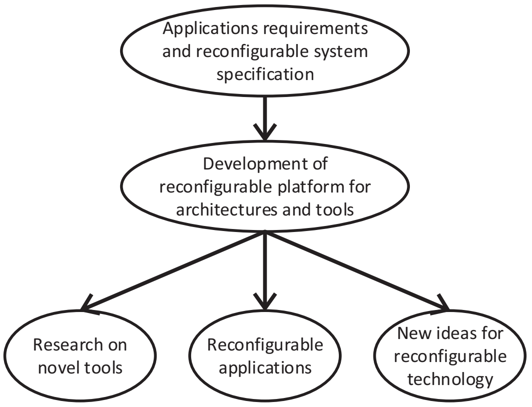
Applications requirements and reconfigurable system specification
Development of reconfigurable platform for architectures and tools
Research on novel tools
Reconfigurable applications
New ideas for reconfigurable technology

Fig. 1. Conceptual overview of the EXTRA project

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 not focus on these project parts in this paper but refer to the project website (extrahpc.org) for further information. The focus of this paper is on the open research platform which is further described in detail in Section III.

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. 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. [11], [12], [13] for Quantum Monte Carlo methods), the algorithms need to be reconsidered in order to fully exploit the power of reconfigurable architectures.

III. Reconfiguration Platform

In this section we describe the EXTRA Open Research Platform. First, we describe the high-level aspects of the platform including its main components, and then focus on the low-level platform details and reconfiguration aspects.

A. High-Level Platform Overview

Figure 2 presents a schematic of the envisioned EXTRA Open Research Platform. The main components of the platform are numbered from (1) to (6), which we explain next:

(1) Platform Inputs. The EXTRA platform requires two types of inputs:

i. A hardware platform specification and requirements, including the size, types and properties of compute, storage and network resources available in the target platform.
ii. The application source-code with user annotations and input datasets. Code generation for the full application is beyond the scope of the EXTRA project. Instead, we focus

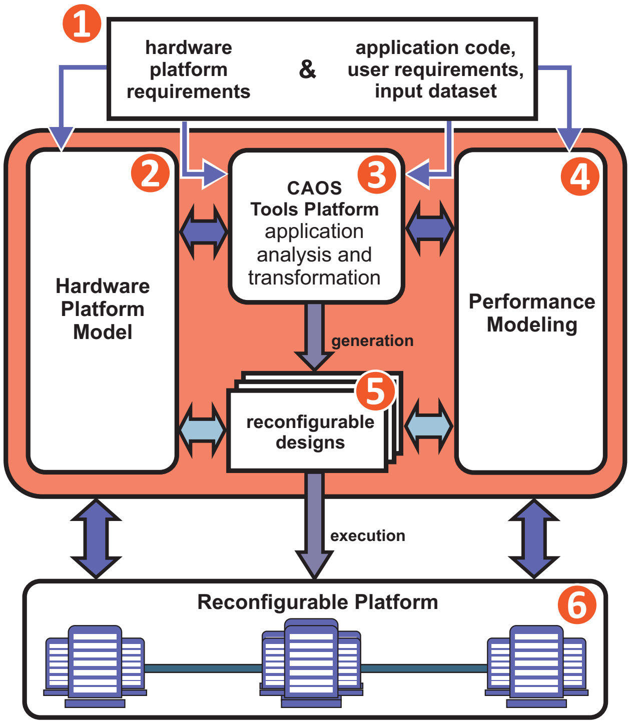
1
hardware platform requirements
&
application code, user requirements, input dataset
2
Hardware Platform Model
3
CAOS Tools Platform application analysis and transformation
4
Performance Modeling
generation
5
reconfigurable designs
execution
Reconfigurable Platform
6

Fig. 2. The EXTRA Open Research Platform Architecture

on identifying and 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. Additional application specification in the form of code annotations, i.e., code pragmas, will be used to further improve the effectiveness of the EXTRA platform, indicating potential acceleration candidates, multiple code versions for different types of devices, user requirements such as desired performance, and/or DAG-like specifications of the code. Another important input is the (collection of) input datasets or generative models that can be used to obtain realistic datasets.

In this project, we focus mainly on three application classes to validate the EXTRA approach: 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) Hardware Platform Model. This component of the platform provides a high-level specification of the target platform used for execution to guide the design and optimization processes. For example, "1 single node, with 1 CPU and 1 FPGA accelerator is a platform model with a very different set of requirements than "a HPC cluster of N nodes, each accelerated with M accelerators. This model is also used as an input to the CAOS Tools Platform (3), which takes offloading decisions based on available resources, leading to a modular application model that fits the hardware model at a conceptual level. The low-level details of the platform are described in Subsection III-B. For the hardware platform, we envision a selection between actual hardware machines and accurate simulators. Since our target is an open source platform that can explore the opportunities that reconfigurable computing can offer for large scale computational intensive applications, we cannot be limited to a single hardware alternative. The EXTRA platform prototype will include 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 by the EXTRA community or interested parties (application owners, performance engineers, M.Sc. students and visiting scholars).

(3) CAOS Tools Platform. The CAOS (CAD as an Adaptive Open platform Service) framework produces a reconfigurable design through a set of analyses and transformations by combining the knowledge of application characteristics, the application requirements, the input dataset and the hardware platform model. We develop a two-stage analysis toolchain: the application analyzer and the accelerated application design. The CAOS toolchain is explained in more detail in Subsection III-C. We 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.

(4) Performance Modeling. The goal of the EXTRA platform is not only to identify opportunities for hardware acceleration, but also to focus on scalability aspects. In particular, targeting exascale computing requires complex performance modeling techniques to derive reliable scalability estimates for the applications and the hardware. The performance modeling component has three stages:

A. In the first stage, we derive a generic high-level performance model by analyzing the DAG of the application and produce a high-level performance profile. Using Amdahl's law and the Roofline model, this model indicates the potential in terms of scalability and performance for the accelerated application.
B. Once the hardware resources are known 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 produced in the first stage, providing more information on the limitation that the platform imposes (if any) on the scalability of the application.
C. The final stage produces a full performance model. This model is calibrated to the actual hardware that is being used, and thus provides a more accurate prediction of the performance to be obtained. Based on the current state

of the art, we assume that this model will be based on statistical models, but analytical elements might also be included for certain hardware platforms.

We believe that performance modeling in EXTRA will play a double role: it can be used as a performance indicator for the end-user and tools (*CAOS Tools Platform*), or to perform scalability analysis for determining, with reasonable accuracy, the impact of exascale computing (*Performance Modeling*).

(5) Reconfigurable Designs. The Open Research Platform generates one or more reconfigurable designs, selecting the most efficient design based on the performance and hardware models used. We expect each reconfigurable design to support a collection of kernels derived using different optimization techniques (effectively different versions of multiple kernels to be benchmarked and profiled) and an application driver, which implements the DAG structure.

(6) Reconfigurable Platform. This component corresponds to the actual hardware machines and simulators used for design execution, and its runtime system, possibly based on the FASTER runtime system [14].

Reconfigurable platforms can complicate the problems of verifying that a design is performing correctly, or even debugging when designs fail. To address these problems, we will extend our work to support 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 [15] 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 having it as a separate entity. This approach 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 optimal reconfiguration infrastructure, according to the target platform architecture and device. We therefore 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 are 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 [16]. 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 [17]. 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.

C. CAOS: CAD as an Adaptive Open platform Service

The CAD as an Adaptive Open Platform Service (CAOS) framework provides a fully integrated environment for automating and supporting all the steps involved in the acceleration of applications on HPRC systems.

As shown in Figure 3, the platform targets two different types of users: application designers and tools developers.

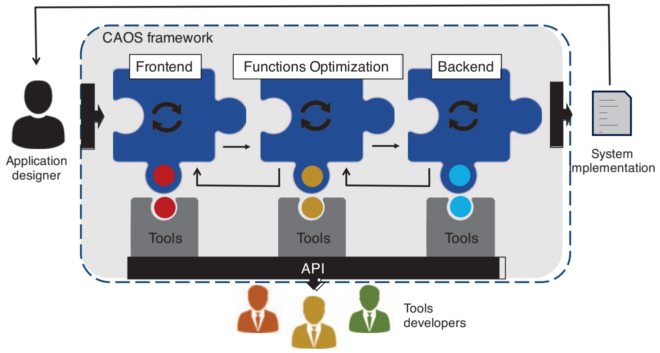

Fig. 3. High level overview of the CAOS framework in terms of the framework main components, the flow followed by the application designer in order to generate the final system implementation, and the interaction of the tool developers with the platform APIs.

From the application designer perspective, the platform has been designed with usability and interactivity in mind. One of the objective of the platform is indeed to allow users with low expertise on HPRC systems to be able to quickly optimize their application, analyze the potential performance gain and deploy the final design on the target architecture. Nevertheless, the flow defined by the framework is highly interactive, providing suggestion and error reports at each stage of the optimization process and allowing the user to specify or modify the solutions proposed by the framework. From the developers perspective, the platform is composed of a set of modular components that interact by means of well defined API. This allows tool developers and researchers to integrate their algorithms within the platform with a twofold objective: (1) improve the overall quality and experience for the application designers using the framework and (2) test and compare novel approaches with respect to already available solutions.

The framework flow is organized in three main components: the frontend, the function optimization component and the backend. The frontend requires the application designer to specify the application code, a description of the system in terms of the available HW nodes and their interconnection topology together with a profiling dataset. Within this stage, the user code is profiled and, according to the application and the HW description, the framework provides suggestions on the architectural templates to use for the configuration of the reconfigurable hardware available in the system. The result of the frontend consists of a list of application functions that are optimized within the CAOS function optimization stage. The optimized functions together with the HW description of the system are then exploited within the backend stage in order to produce the final bitstreams for the FPGA's configuration as well as the application runtime.

IV. 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. We are currently seeking collaborations with other researchers and projects that have a possible interest in using our platform for their own research. They can follow our progress by taking part in a research advisory board. Industrial partners with interest in our work can also join our industry advisory board.

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

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

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

[16] 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.

[17] 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.