EDRA:
A Hardware-assisted Decoupled Access/Execute Framework on the Digital Market*

Invited Paper

Dimitris Theodoropoulos[1], Andreas Brokalakis[1], Nikolaos Alachiotis[2], and Dionisios Pnevmatikatos[3]

[1] Telecommunication Systems Institute
Technical University of Crete Campus – Akrotiri 73100, Chania, Greece
{dtheodoropoulos,abrokalakis}@isc.tuc.gr
[2] Computer Architecture for Embedded Systems
Faculty of EEMCS, University of Twente, The Netherlands
n.alachiotis@utwente.nl
[3] Institute of Communication and Computer Systems
National Technical University of Athens, Greece
pnevmati@cslab.ece.ntua.gr

Abstract. EDRA§ was an Horizon 2020 FET Launchpad project that focused on the commercialization of the Decoupled Access Execution Reconfigurable (DAER) framework – developed within the FET-HPC EXTRA project – on Amazon's Elastic Cloud (EC2) Compute FPGA-based infrastructure. The delivered framework encapsulates DAER into a EC2 virtual machine (VM), and uses a simple, directive-based, high-level application programming interface (API) to facilitate application mapping to the underlying hardware architecture. EDRA's Minimum Viable Product (MVP) is an accelerator for the Phylogenetic Likelihood Function (PLF), one of the cornerstone functions in most phylogenetic inference tools, achieving up to 8x performance improvement compared to optimized software implementations. Towards entering the market, research revealed that Europe is an extremely promising geographic region for focusing the project efforts on dissemination, MVP promotion and advertisement.

Keywords: Cloud computing · FPGAs · Decoupled Access-Execute.

1   Introduction and Concept

Over the last years, major cloud providers (Amazon, Alibaba, Nimbix) started offering services that utilize special chips called FPGAs that enable faster workload processing than conventional software-based configurations for a large range
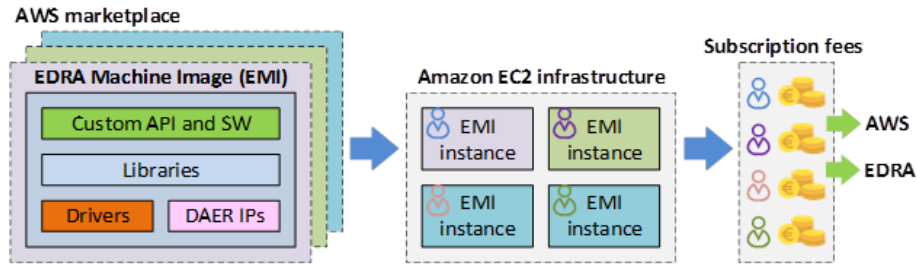
Fig. 1. Availability of the EDRA AMIs (EMIs) through the AWS marketplace to customers.

of high-performance applications. However, mapping applications onto FPGAs is a cumbersome process that requires extensive background on hardware development and specialized IT personnel.

To alleviate this hurdle, the FET project "EXTRA" (Exploiting eXascale Technology with Reconfigurable Architectures - GA 671653) [2] focused on devising efficient ways to deploy ultra-efficient heterogeneous compute nodes in order to meet the massive performance requirements of future exascale High Performance Computing (HPC) applications. A major outcome was the design and implementation of a novel framework that maps applications onto FPGAs employing a Decoupled Access Execute Reconfigurable (DAER) architecture for HPC platforms [1], originally based on the idea of Decoupled Access-Execute architectures [7].

During the EXTRA project, various algorithmic workloads were mapped to reconfigurable HPC platforms using the DAER approach, achieving significant performance improvements in spite of different memory access patterns and/or computational requirements. However, there are still two main obstacles for making the EXTRA results available and easily accessible to the market: (a) launching applications onto the EXTRA hardware is currently based on a semi-automatic tool flow, requiring developers to manually separate memory accesses from data processing tasks, and (b) FPGA-based acceleration requires the additional inherent cost of specialized hardware.

To this end, the EDRA framework tackles the aforementioned drawbacks as follows:

 – it provides a fully-automated software workflow that automatically generates DAE-compatible application executables requiring only minor code annotation;
 – it combines the EDRA workflow with Amazon's software library for taking advantage of the available reconfigurable hardware;
 – it integrates the complete stack with the EXTRA DAE architecture, wrapped within a single Amazon Machine Image (AMI) dubbed EMI (EDRA AMI).

Figure 1 illustrates the EMI exploitation strategy; EDRA plans to list the EMI instance to the AWS marketplace with EDRA software, libraries/drivers

(e.g., the Xilinx Runtime System for interfacing the FPGA), and domain-specific DAER IPs for accelerating application workloads. End-users can deploy an EMI instance via the AWS marketplace listing it Amazon's EC2 FPGA-supported machine instance, charged on a pay-as-you-go basis. The AWS marketplace platform is responsible for forwarding subscription fees paid by end-users to Amazon (for hosting EMI instances to its infrastructure) and the 3rd party seller (i.e., EDRA).

The rest of the paper is organized as follows: Section 2 presents the project achievements and impacts. Section 3 describes the EDRA framework, whereas Section 4 provides results on the developed MVP in terms of performance against other solutions. Section 5 elaborates on a market analysis tailored to the EDRA's MVP, and finally Section 6 concludes the paper.

2 Project Achievements and Impact

2.1 Project Achievements

EDRA was a FET Innovation Launchpad (GA #851631) [¶], that run from May 2019 until October 2020. Interested users can follow EDRA on Twitter[‖] and LinkedIn[**], as well as find more details on the project website[††].

EDRA achieved its goal on making the DAER technology ready for entering the market by reaching the following achievements:

1. EDRA framework: The project successfully updated the EXTRA IP, and developed a full-fledged framework that can (semi) automatically create DAE-compatible applications, capable to facilitate available hardware resources for faster workload processing.
2. MVP in the domain of phylogenetics: Having a strong scientific background in the domain of phylogenetics, EDRA decided to develop and deploy a first MVP using its in-house framework that accelerates phylogenetics analysis on Amazon's FPGA-supported machines.
3. MVP market analysis: Towards validating the decision of deploying the MVP to the market, the team conducted a thorough analysis on market size and opportunities with respect to the bioinformatics domain. Results suggest that an estimated market size directly fitting to the EDRA MVP is valued at approximately 7.9 M€, 10.8 M€ and 14.6 M€ for 2021, 2022 and 2023 respectively.
4. Business model formulation: EDRA developed a complete and sustainable business model. Its value proposition is based on solutions to customers who would like to execute applications faster compared to their current setup, as well as reduce IT costs related to deployment and maintenance. Due to the

---

[¶]https://cordis.europa.eu/project/id/851631

[‖]https://twitter.com/ProjectEdra

[**]https://www.linkedin.com/groups/8790812/

[††]https://edra-project.eu/

team expertise on hardware design and background on computational phylogenetics, EDRA plans to also provide dedicated application acceleration services on phylogenetics. Key resources required to support the EDRA value proposition are budgets related to staff support, cloud/digital resources, and facilities and IPR management. Customer segments comprise users from the HPC domain and academic institutes in the domain of Bioinformatics. As shown in Figure 1, the revenue model is based on fixed charges for deploying customer applications to Amazon's marketplace using the EDRA framework. The EDRA's MVP revenues will be based on Amazon's pay-as-you-go charging policy.

2.2   Project Impact

- Economic and business impact: The project has delivered the EDRA framework, a novel technology for rapid and (semi) automatic hardware-accelerated deployment of applications to cloud resources. The framework essentially allows the quick launch of cloud-supported services for SMEs and corporations, thus enabling faster and better services for end users, a key aspect for economic growth.
- Increased value creation from FET projects by picking up innovation opportunities: Based on the framework developed by EDRA, the team formulated a go-to-market strategy for offering hardware-acceleration services to the cloud for demanding applications. Moreover, EDRA released an MVP that enhances research on phylogenetics, an important area that can assist on the fight against the COVID-19 outbreak and other potential pandemics.
- Improved societal and market acceptance: The COVID-19 outbreak demonstrated that scientists need access to powerful computational resources with fast turnaround times of results on drug analysis and model simulation. EDRA's MVP addresses today's important need for more computational power towards faster analyses of virus evolution.
- Contributing to the competitiveness of European industry/economy: EDRA picked up on the fact that major cloud providers, such as Amazon and Alibaba, started offering services that support FPGAs; the DAE framework allows the offer of generic low-risk hardware-acceleration services on the cloud for customers that wish to remove their application back end from their premises.
- Stimulating, supporting and rewarding an open and proactive mind-set: EDRA's ability for quick deployment of hardware-accelerated applications, allow SMEs and corporations to investigate and propose new services to end-users with minimum investment risk, strengthening even more the European industry sector.
- Scientific impact: EDRA's MVP is a novel solution that allows biologists and researchers in phylogenetics to increase their productivity while reducing IT costs. The majority of the software tools used for experiments are compute-bound, hence the additional processing power that EDRA's MVP provides can further assist on understanding the origins of lethal viruses. This is a

valuable asset for constraining the spread of potential new pandemics, should anytime happen.

3   The EDRA Framework

3.1   Source-to-source translation

EDRA developed a source-to-source translator infrastructure, henceforth referred to as "EDRA-gen", to facilitate code annotation and translation. Its purpose is to reduce development time and yield a correct-by-construction design for the final accelerated system. Automated hardware generation in EDRA-gen is inspired by a generic Decoupled Access-Execute (DAE) architectural paradigm.
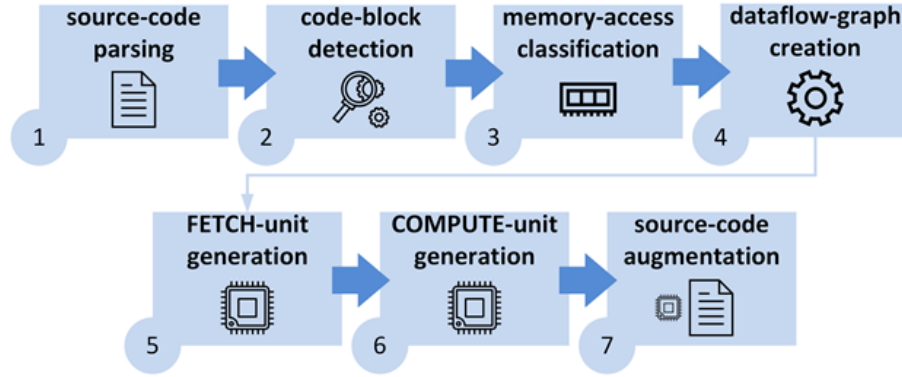

Fig. 2. Source-to-source translation stages of EDRA-gen.

EDRA-gen-based hardware generation starts with a minimally annotated user source code that indicates at least one target for-loop. The flow consists of 7 discrete steps (Figure 2) that collectively extract the code block of interest in the user's code (the target for-loop), resolve dependencies (if exist), construct an abstract syntax tree (AST), and use it to generate all the required data-fetch (ACCESS) and process (EXECUTE) units. EDRA-gen relies on LLVM to generate token lists for the source files and implements a series of algorithms directly on the token lists to extract the AST. Once the AST is created, a series of algorithms operate on the AST to generate C code for each DAE component, driven by the available Vivado HLS directives to be used.

3.2   Hardware Support

Figure 3 shows how the EDRA hardware architecture is mapped to the AWS F1 machine instance. EDRA allows hardware accelerators to exchange data with
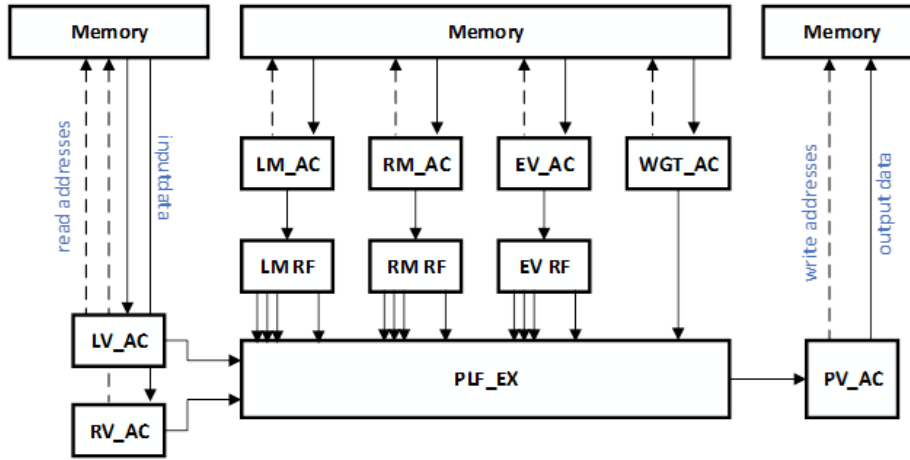
Fig. 3. DAE implementation to the AWS F1 machine instance.

the F1 host processor either via the shared DDR4 memory accessible from the DDR4-C memory controller or the APP PF memory space, accessible via the PCIe. Supporting both of the aforementioned methods allows concurrent memory access (read and write) and task offloading to accelerators either with OpenCL or the AWS FPGA PCI library, providing maximum flexibility to programmers during application development:

– Shared data stored in the DDR4 memory: an application can share data with the accelerator via the DDR4 memory, using OpenCL functions. In this case, the "read dataDDR" module reads data via the DDR4-C memory controller over an AXI4 protocol, and then forwards it over an AXI4 Stream interface to the hardware IP (HW IP) for processing. The HW IP sends results to the "write dataDDR" also over an AXI4 Stream interface, which forwards them via an interconnect (IC) module back to the DDR4-C memory controller.
– Shared data stored in the APP PF: an application can also share data with the accelerator using the AWS FPGA PCI library. In this case, the APP PF exposes a physical address space up to 127 GiB that facilitates data transfers between the host processor and the accelerator; the "read dataPCI" module reads data via the PCIES (PCIe Slave) interface (also based on the AXI4 protocol), and then forwards it via an AXI4 Stream interface to the "HW IP". When data processing is finished, the "HW IP" sends results back to the "write dataPCI" module, which forwards them to the host CPU via the PCIEM (PCIe Master) interface.

Finally, the BAR0 (Base Address Registers) AXI4 interface space exposes the BAR0 APP PF memory space that facilitates management and monitoring (CL

Fig. 4. Top-level Decoupled Access/Execute Architecture of the PLF accelerator core.

MGT) of the hardware accelerator (e.g., start/stop and sync barriers) by the application executed at the host CPU.

4 MVP Architecture and Performance

A detailed description of the EDRA MVP is provided by Malakonakis et al. [4]. Figure 4 illustrates the PLF accelerator based on the DAE approach. Overall, the PLF core has seven access units and a single execution unit. There are six input access units that fetch data from memory to the accelerator (Left and Right Vectors, Left and Right Matrices, EV vector and scaling vector (WGT)), and a single output access unit that writes the results of the computation back to memory.

An invocation of the accelerator consists of two steps. First, the Left- and Right-matrix access units retrieve the left and right probability matrices, and the EV access unit retrieves the inverted eigenvector (RAxML computes P(t) matrices based on eigenvector/eigenvalue decomposition) from memory and store them into register files. Then, the two FIFO-based access units fetch the Left and Right vectors that correspond to the left and right child nodes, and stream them through the PLF datapath. The output Parent vector is stored in memory through a FIFO-based access unit. The access units that prefetch data into register files do not contain FIFOs to lower resource utilization.

It should be noted that resource utilization coverage does not exceed 30% of the available resources in any of the FPGA hardware primitives (BRAMs, Logic Cells etc). This provides a potential for further optimization of the design by adding more PLF accelerator engines. Moreover, a double-buffering mechanism was adopted to reduce data transfer overheads.
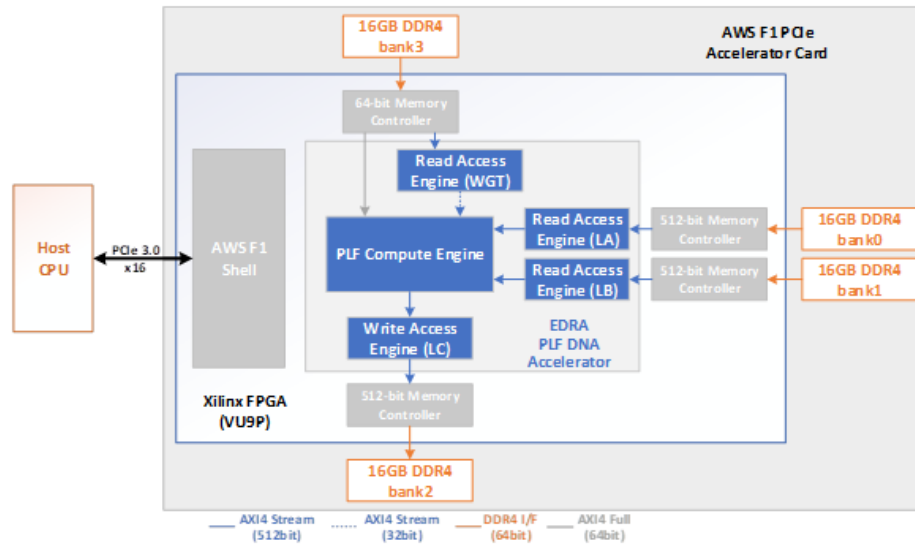
Fig. 5. PLF accelerator system architecture (AWS F1).

Figure 5 depicts the PLF integration to the AWS F1 machines. The memory controllers allow up to 512-bit-wide connections to the FPGA compute resources through an AXI stream interface. The PLF accelerator uses two such interfaces in order to transfer the two matrices from memory to the EX unit and another one to transfer the results back to memory. A fourth interface to the remaining memory channel (64bits wide) is used for the R and L matrices and EV vector as well as the scaling factors.

N	Block Size (Double Buffering)						Single Buffering	Software PLF
	4k	8k	16k	32k	64k	128k		
1M	0,098	0,052	0,042	0,042	0,053	0,071	0,055	0,18
2M	0,3	0,11	0,079	0,076	0,085	0,11	0,11	0,36
5M	1,56	0,44	0,21	0,17	0,18	0,22	0,27	0,91
10M	3,25	0,93	0,43	0,35	0,35	0,42	0,55	1,83

Fig. 6. Execution Time of the PLF function on AWS F1 instance. Software PLF is executed on the same F1 instance. N refers to the number of elements of the Left and Right Probability Vectors. All times are reported in seconds.

The PLF performance was compared against an optimized software implementation of RAxML on the same platform as the accelerated system. The underlying hardware platform is the CPU system of the AWS F1 instance, which

is based on an Intel Xeon E5-2686v4 processor (8 vCPUs available). Table 6 lists a set of experimental results that compares the software and hardware-accelerated PLF implementations. Comparing the execution time of the best accelerator case on the Amazon F1 and the time required to compute the same function in software, it can be seen that the accelerator provides 2.3x to 5.2x better performance. The performance gap widens as the input size increases. For the overall RAxML application, this translates to up to 3.2x reduction in the overall execution time for the most demanding datasets that were tested.

5 MVP Market Analysis

5.1 Market Size

EDRA's MVP accelerates the RAxML application [8], a well-established tool for phylogenetic analyses of large datasets under maximum likelihood. RAxML has more than 13K citations from its original publication (2006), and an additional 10,800 citations from researchers globally over the last 5.5 years (2014 – 2019) on its latest version. Based on accumulated data the estimated market size is valued at approximately 7.9 M€, 10.8 M€ and 14.6 M€ for 2021, 2022 and 2023 respectively. Knowing the market size is a valuable aspect, however a market strategy needs to reflect the target customer segments as well. Towards building the client profile for EDRA's MVP, the EDRA team used Google Scholar to collect information with respect to authors citing RAxML in their work. More specifically, starting on 2016, EDRA examined the first 100 publications each year that cite RAxML, and identified each article's leading author's affiliation and location. Figure 7 shows the analysis results from 2016 until Q2 2020. As observed, throughout each year leading authors affiliated with research institutes located in Europe and US represent 88% (2016), 72% (2017), 84% (2018), 82% (2019), and 64% (Q2 2020).

An interesting observation is that until Q2 2020 publications from authors affiliated with academic institutes in Asia raised from 6% in 2016 to 28%. One reason is that RAxML is widely used to study the evolution of viruses (among others), hence many researchers used it for analyzing the evolution of the COVID-19 virus in Asia, where the first recorded case occurred according to the World Health Organization records. This fact shows the importance of making available ample computing power to biologists, in order to speed up their analysis experiments.

Overall, the above study leads to the following conclusions:

- The potential market size for EDRA's MVP is valued at approximately 7.9 M€, 10.8 M€ and 14.6 M€ for 2021, 2022 and 2023, respectively.
- The primary client segment for EDRA's MVP is biologists and scientists working either for academic institutes or Contact Research Organizations.
- RAxML users can utilize additional processing power to conduct faster their experiments on COVID-19 evolution analysis.
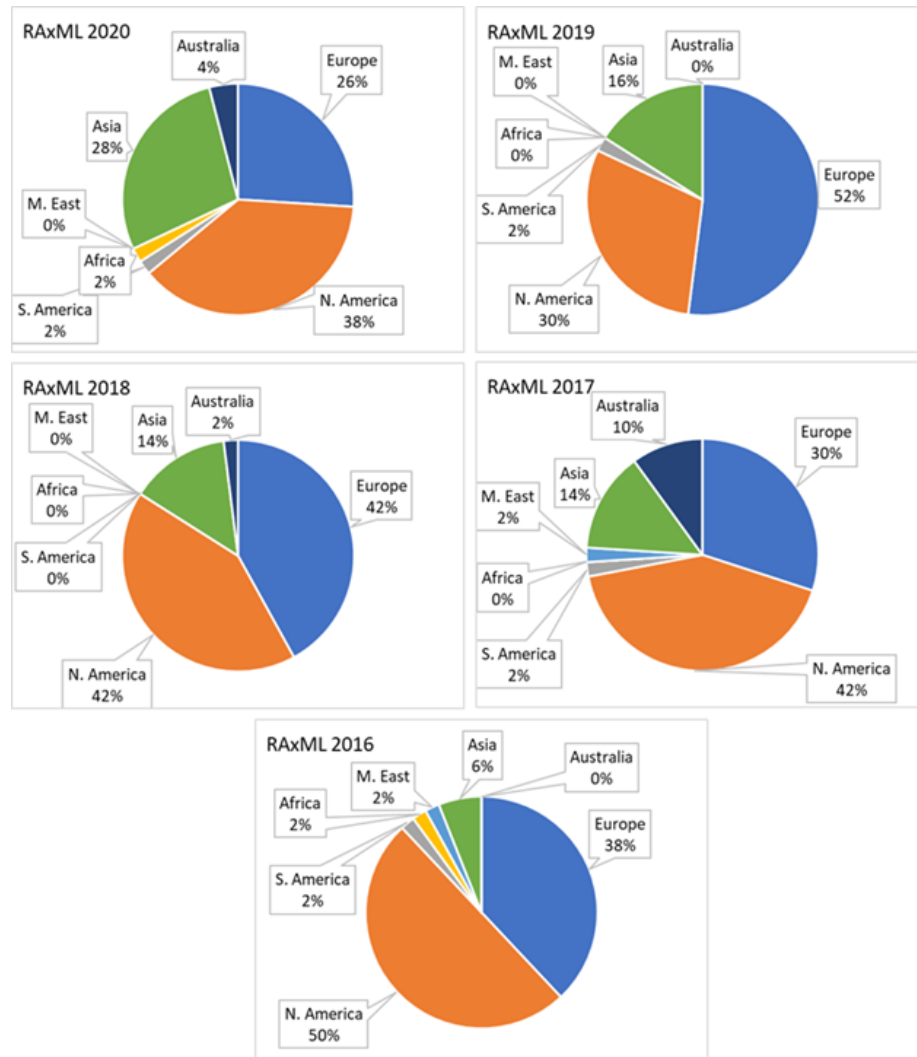
Fig. 7. Geographical breakdown of RAxML users from 2016 to Q2 2020.

5.2   Other Approaches

RAxML is made available to the community under a GNU Public License (GPL). Moreover, RAxML is already optimized for multi-threaded CPUs as well as GPUs [3, 6], hence researchers clone it either on workstations or private servers to run their experiments. This approach though requires specialized IT personnel to ensure that a machine is properly configured (e.g. installing OS updates, drivers, software development tools installed, etc.), leading to increased costs related to infrastructure acquisition (e.g. buying server-class machines that cost thousands of €) and maintenance (e.g. paying electricity bills, hosting facilities for servers, salaries for extra IT personnel, etc.).

The above ad-hoc approaches are not identically configured (e.g. different OS versions, hardware resources) and usually impose significant development challenges, resulting in reduced productivity, non-optimal utilization of the available computational resources, and excessive IT costs. To partially alleviate the issue of system software variations, Informatics LLC created a virtual machine on Amazon's marketplace, called MolBioCloud [5], that contains a large set of software tools related to molecular biology. MolBioCloud offers a fully configured and tested environment in the form of a virtual machine for biologists working on different research areas. However, MolBioCloud does not support hardware acceleration.

6   Conclusions and Next Steps

EDRA successfully delivered an end-to-end framework for mapping applications onto Amazon's FPGA-supported cloud platforms based on the DAE approach. Moreover, the project delivered a pioneering MVP that enables faster processing of workloads related to phylogenetics, as well as conducted thorough research with respect to market addressable size and the MVP average customer persona.

Towards pushing the EDRA MVP to the market, the project team has already initiated the process of launching a spin-off based on the formulated business plan. The spin-off will focus on deploying the MVP to Amazon's marketplace, where biologists and researchers on phylogenetics will be able to download it on a pay-as-you-go charging policy, and instantly conduct their experiments up to 2.5x faster compared to the currently available software-optimized configurations.

References

1. Charitopoulos, G., Vatsolakis, C., Chrysos, G., Pnevmatikatos, D.: A decoupled access-execute architecture for reconfigurable accelerators. In: IEEE 18th International Conference on Computational Science and Engineering. pp. 244–247 (05 2018)
2. Ciobanu, C.B., Varbanescu, A.L., Pnevmatikatos, D., Charitopoulos, G., Niu, X., Luk, W., Santambrogio, M.D., Sciuto, D., Kadi, M.A., Huebner, M., Becker, T., Gaydadjiev, G., Brokalakis, A., Nikitakis, A., Thom, A.J.W., Vansteenkiste, E.,

Stroobandt, D.: Extra: Towards an efficient open platform for reconfigurable high performance computing. In: 2015 IEEE 18th International Conference on Computational Science and Engineering. pp. 339–342 (2015)

3. Izquierdo-Carrasco, F., Alachiotis, N., Berger, S., Flouri, T., Pissis, S.P., Stamatakis, A.: A generic vectorization scheme and a gpu kernel for the phylogenetic likelihood library. In: 2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum. pp. 530–538 (2013)

4. Malakonakis, P., Brokalakis, A., Alachiotis, N., Sotiriades, E., Dollas, A.: Exploring modern fpga platforms for faster phylogeny reconstruction with raxml. In: 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE). pp. 97–104 (2020)

5. MolBioCloud: http://molbiocloud.com/, [Online; accessed 17-May-2021]

6. Ott, M., Stamatakis, A.: Preparing raxml for the spec mpi benchmark suite. In: High Performance Computing in Science and Engineering. pp. 757–768 (01 2010)

7. Smith, J.E.: Decoupled access/execute computer architectures. In: ACM Transactions on Computer Systems. pp. 112–119 (1984)

8. Stamatakis, A.: Raxml version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics (Oxford, England) **30** (01 2014)