Harnessing Hardware Acceleration with RISC-V and the EU Processor

Harnessing Hardware Acceleration with RISC-V and the EU Processor

Juan Fumero[1], Athanasios Stratikopoulos[1], Mehdi Goli[2], Ruymán Reyes[2], Konstantinos Nikas[3], Dionisios Pnevmatikatos[3], Nectarios Koziris[3] and Christos Kotselidis[1*]

[1]Department of Computer Science, The University of Manchester
[2]Codeplay Ltd, UK
[3]ICCS, National Technical University of Athens, Greece

Abstract

In this talk, we will present the newly EU-funded project AERO (Accelerated EU Cloud) whose mission is to bring up and optimize the software stack of cloud deployments on top of the EU processor. After providing an overview of the AERO project, we will expand on two main components of the software stack to enable seamless acceleration of various programming languages on RISC-V architectures; namely, ComputeAorta which enables the generation of RISC-V vector instructions from SPIR-V binary modules, and TornadoVM which enables transparent hardware acceleration of managed applications. Finally, we will describe how the ongoing integration of ComputeAorta and TornadoVM will enable a plethora of applications from managed languages to harness RISC-V auto-vectorization completely transparently to developers.

Introduction

The European Commission (EC) via its recent Chips Act [1], is investing significant resources in achieving sovereignty in chip manufacturing and production. To that end, the European Processor Initiative (EPI) [2] project has been working towards creating chips targeting various segments such as HPC, IoT/Edge, and others. The EPI designs are mainly based on ARM and RISC-V architecture with a common theme running amongst them being the presence of heterogeneity in the form of accelerators. Various EU companies and research institutions are working towards the commercialization of the EPI-derived designs while actively participating in the RISC-V International via the various SIG groups.

In order to provide a complete compute stack, besides the EU hardware designs, the EC is heavily investing in the upbringing of the software infrastructure too. To that end, the AERO project [3] has been funded and recently started with the main focus of optimizing key software components for the future heterogeneous EU cloud.

AERO

The Accelerated European Cloud (AERO) project is a three-year EU project with the clear mission of **enabling the future heterogeneous EU cloud infrastructure**. Towards this, it will develop - to a high TRL level - all components necessary to achieve out-of-the-box heterogeneous execution of the cloud ecosystem on the EU processor. The outcome will be a set of **compilers, runtime systems, operating systems, system software, and applications** that can leverage the underlying capabilities of the future EU cloud equipped with GPUs, FPGAs, and other accelerators.

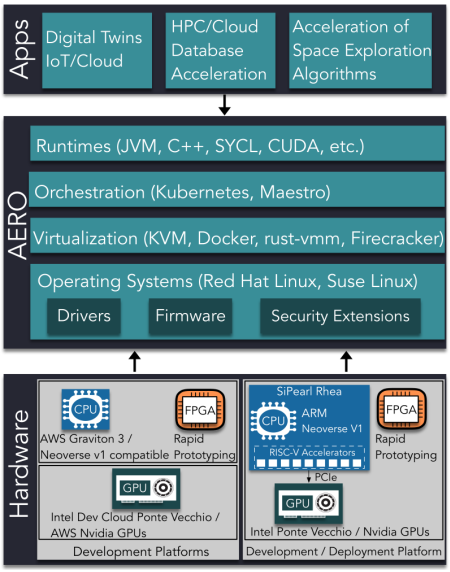

Figure 1: *AERO Hardware/Software compute architecture for the EU Processor and RISC-V Accelerator Platform.*

Figure 1 provides an overview of the AERO project hardware/software stack. The AERO project targets a plethora of key hardware and software components of typical cloud deployments. A key layer of the software stack is the runtime system layer in which significant work will be conducted by two partners of the AERO project: Codeplay Ltd, and The University of Manchester. Codeplay will work towards optimizing the RISC-V scalar to vector LLVM compiler modules, while The University of Manchester will work towards optimizing TornadoVM on the EU processor by ac-

*Corresponding author: christos.kotselidis@manchester.ac.uk

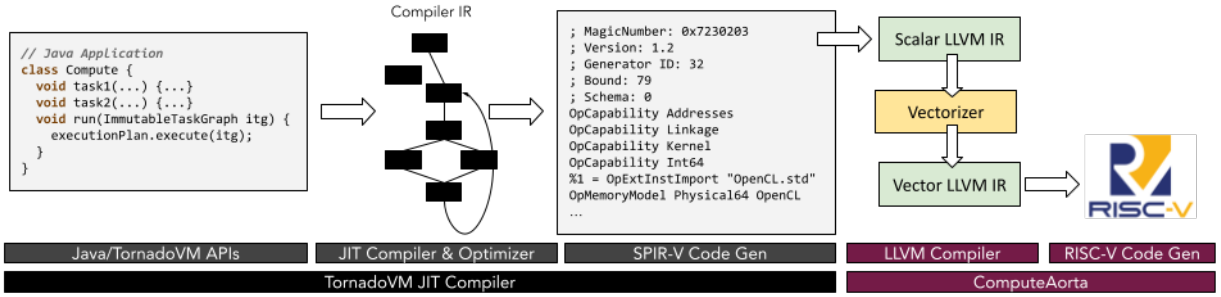

Figure 2: *Compilation pipeline to convert Java (and JVM-based) programs to RISC-V vectorized code using the TornadoVM JIT Compiler front-end, and ComputeAorta backends.*

celerating managed applications on GPUs and other hardware accelerators.

Enabling Language Frontends TornadoVM [4] is a JVM plugin that allows the transparent hardware acceleration of managed applications on devices such as GPUs and FPGAs. This is achieved by automatically compiling Java bytecodes to one of three supported backends: OpenCL, PTX/CUDA or SPIRV. Depending on the characteristics of the target accelerator, developers can select which backend to utilize, and TornadoVM will apply specialized optimizations for the target architecture. Besides, since TornadoVM compiles Java bytecode, any programing language implemented on top of Java can benefit from heterogeneous execution.

Enabling RISC-V Backends ComputeAorta [5] is a framework to provide implementations of open standards, such as OpenCL and Vulkan, for a wide range of heterogeneous hardware, enabling large complex codebases to be offloaded to accelerators. ComputeAorta includes all components required to convert high-level representations of programs in the form of OpenCL/Vulkan with SPIR-V or OpenCL-C into low-level ISA and hardware driver API calls.

Thus, ComputeAorta enables high-level programming models emitting SPIR-V to be offloaded to wide ranges of hardware accelerators, including RISC-V accelerators cores that can support the RVV Vector Extension.

Enabling Vectorization for RISC-V The combination of TornadoVM and ComputeAorta creates new opportunities in harnessing RISC-V vector units from managed runtime programming languages completely transparently to developers. Figure 2 depicts how Compute AORTA could be used independently or in combination with TornadoVM to enable vector acceleration of Java and various programming languages for RISC-V architectures.

In this talk we will use Compute Aorta to offload the SPIR-V generated code on simulation of running two RISC-V cores with the RISC-V Vector Extension for the accelerator cores, targeting the RISC-V SPIKE simulator as a demonstration platform. By leveraging existing ComputeAorta vectorizer (VECZ), existing OpenCL and SPIR-V programs will automatically generate the correct instructions for RVV. Please note that since this is work-in-progress the final integration points might vary in their final form.

Conclusions

In this talk, we will introduce the AERO project and its contributions to the software ecosystem of the EU processor. We will focus primarily on the RISC-V aspect of the project by explaining two key components of AERO: ComputeAorta RISC-V compiler, and the TornadoVM Java Framework. Through the ongoing integration activities, we showcase the potential benefits that developers may have by harnessing RISC-V accelerators transparently via their chosen programming languages.

Acknowledgement

This work is partially supported by the European Union's Horizon Europe programme under grant agreement No 101092850 (AERO). In addition, this work is funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee for grant numbers 10048318 and 10048915.

References

[1] European Chips Act. *https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/europe-fit-digital-age/european-chips-act_en.*

[2] European Processor Initiative. *https://www.european-processor-initiative.eu.*

[3] *AERO: Accelerated ErRopan clOud.* Last access: March 2023. URL: `https://aero-project.eu/`.

[4] Juan Fumero et al. "Dynamic Application Reconfiguration on Heterogeneous Hardware". In: *Proceedings of the 15th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*. VEE 2019. Association for Computing Machinery, 2019. DOI: `10.1145/3313808.3313819`.

[5] Alastair Murray and Ewan Crawford. "Compute Aorta: A Toolkit for Implementing Heterogeneous Programming Models". In: *Proceedings of the International Workshop on OpenCL*. IWOCL '20. Association for Computing Machinery, 2020. DOI: `10.1145/3388333.3388652`.