The VINEYARD project: Versatile Integrated Accelerator-based Heterogeneous Data Centres

Christoforos Kachris, Dimitrios Soudris
NTUA, ICCS
Athens, GR

Georgi Gaydadjiev
Maxeler Technologies
UK

Huy-Nam Nguyen
Bull Systems
France

Dimitrios S. Nikolopoulos
Queens University of Belfast
UK

Angelos Bilas
FORTH
Greece

Neil Morgan
The Hartree Centre
UK

Christos Strydis
Neurasmus
The Netherlands

Vasilis Spatadakis
Neurocom Luxembourg
Luxembourg

Dimitris Gardelis
ATHEX
Greece

Ricardo Jimenez-Peris
LeanXcale
Spain

Alexandre Almeida
Globaz
Portugal

***Abstract*—Emerging applications like cloud computing and big data analytics have created the need for powerful centers hosting hundreds of thousands of servers. Currently, the data centers are based on general purpose processors that provide high flexibility but lacks the energy efficiency of customized accelerators. VINEYARD[1] aims to develop novel servers based on programmable hardware accelerators. Furthermore, VINEYARD will develop an integrated framework for allowing end-users to seamlessly utilize these accelerators in heterogeneous computing systems by using typical data-center programming frameworks (i.e. Spark). VINEYARD will foster the expansion of the soft-IP cores industry, currently limited in the embedded systems, to the data center market. VINEYARD plans to demonstrate the advantages of its approach in three real use-cases a) a bio-informatics application for high-accuracy brain modeling, b) two critical financial applications and c) a big-data analysis application.**

I. Introduction

Emerging web applications like cloud computing and big data analytics have increased significantly the workload on the data centers during the last years. In 2015, the total network traffic of the data centers was around 4.7 Exabytes and it is estimated that by the end of 2018 it will cross the 8.5 Exabytes mark, following a cumulative annual-growth rate (CAGR) of 33% [1]. In response to this scaling in network traffic, data-center operators have resorted to utilizing more powerful servers. Relying on Moore's law for the extra edge, CPU technologies have scaled in recent years through packing an increasing number of transistors on chip, leading to higher performance. However, on-chip clock frequencies were unable to follow this upward trend due to strict power-budget constraints. Thus, a few years ago a paradigm shift to multicore processors was adopted as an alternative solution for overcoming the problem. With multicore processors we could increase server performance without increasing their clock frequency. Unfortunately, this solution was also found not to scale well in the longer term. The performance gains achieved by adding more cores inside a CPU come at the cost of various, rapidly scaling complexities: inter-core communication, memory coherency and, most importantly, power consumption [2].

In the early technology nodes, going from one node to the next allowed for a nearly doubling of the transistor frequency, and, by reducing the voltage, power density remained nearly constant. With the end of Dennard's scaling, going from one node to the next still increases the density of transistors, but their maximum frequency is roughly the same and the voltage does not decrease accordingly. As a result, the power density increases now with every new technology node. The biggest challenge therefore now consists of reducing power consumption and energy dissipation per mm^2. The failure of Dennard's scaling, to which the shift to multicore chips is partially a response, may soon limit multicore scaling just as single-core scaling has been curtailed. This issue has been identified in the literature as the dark silicon era in which some of the areas in the chip are kept powered down in order to comply with thermal constraints [3].

A solution that can be used to overcome this problem is the use of application-specific accelerators. Specialized multicore processors with application-specific acceleration modules can leverage the underutilized die area to overcome the initial power barrier, delivering significantly higher performance for the same power envelope [4]. The main idea is to use the abundant die area by utilizing application-specific accelerators and dynamically power up only the specific accelerators designed for a given workload. The use of highly specialized units designed for specific workloads can greatly advance server processors and can also increase significantly the performance of data centers given a fixed power budget.

This paper presents an overall view of the VINEYARD H2020 project towards the development of novel servers coupled with programmable hardware accelerators. It will, also, build an integrated programming framework for allowing end-users to seamlessly utilize these accelerators in heterogeneous computing systems by using typical data-center programming frameworks (i.e. Spark).

[1]This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 687628

978-1-4673-9680-6/16/$31.00 ©2016 IEEE

II. VINEYARD OBJECTIVES

Todays data centers, which consist of homogeneous processing systems (general-purpose processors), process high volumes of data by consuming excessive amounts of power. Future heterogeneous data centers consisting of different kinds of accelerators (FPGAs, GPUs, etc.) will be able to provide higher performance under lower power consumption. However, to maintain in such heterogeneous systems the ease of programming of homogeneous ones, an integrated run-time scheduler and manager will be required to hide low-level details and relieve the user from the programming complexities involved (per different accelerator type). VINEYARD aims specifically at the automatic utilization of accelerators through developing such an integrated framework that will control the hardware accelerators, while the user will still be allowed to use typical parallel programming frameworks.

VINEYARD will develop an energy-efficient integrated platform for data centers that will consist of (1) energy-efficient servers based on customized hardware accelerators (novel programmable dataflow engines and FPGA-based servers) and a (2) programming framework that will allow users to seamlessly utilize hardware accelerators in heterogeneous computing systems by using typical scalable cloud frameworks (i.e. Spark).

The VINEYARD project will develop novel servers based on programmable dataflow accelerators that can be customized based on the data-centers application requirements. These programmable dataflow accelerators will be used not only to increase the performance of servers but also to reduce the energy consumption in data centers. Furthermore, VINEYARD will develop a programming framework that will hide the complexity of programming heterogeneous systems while at the same time providing the optimized performance of customized and heterogeneous architectures. In this suite, the user works with familiar programming frameworks (i.e. Spark) while a run-time manager selects appropriate accelerators based on application requirements such as execution time and power consumption.

Finally, VINEYARD will provide the necessary middleware that binds together servers with accelerators. Along with this task, VINEYARD will consider both physical servers and virtual machines (VMs). The middleware also handles QoS concerns that arise with the shared use of the accelerators. In this way, the VINEYARD project will develop an integrated platform for heterogeneous accelerator-based servers. The VINEYARD platform will include both the hardware components (customized accelerators) and a software framework, which consists of two novel components: (i) a programming framework that integrates familiar programming models into heterogeneous systems, and (ii) a middleware layer that supports this heterogeneity in virtualized data centers.

Figure 1 depicts the high level diagram of the VINEYARD framework. Applications that are targeting heterogeneous data centers using traditional servers or micro-servers are programmed using traditional data center frameworks, such as

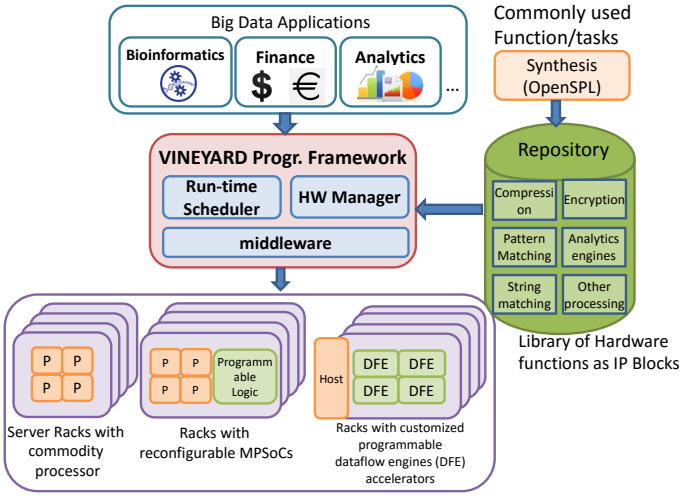

Fig. 1. High level block diagram of the VINEYARD integrated framework.

Spark, and the widely used data management technologies, such as SQL (both OLTP and OLAP for operational databases and data warehouses), NoSQL (a key value data store), and Complex Event Processing (CEP). However, some of the tasks are common across several applications such as sorting of data, key/value processing, encryption, compression, pattern matching, etc. and are extremely computationally intensive. These tasks can be implemented in hardware as customized intellectual-property (IP) accelerators that can achieve much higher performance with lower power consumption. The implementation of the hardware accelerators can be achieved using traditional hardware description languages or other high-level (C, OpenCL) or domain-specific languages (i.e. OpenSPL). These hardware accelerators can be hosted in a repository that can interface with the run-time scheduler.

III. THE VINEYARD APPROACH

A. Accelerator-based servers

In the last few years, emerging applications like cloud computing and big data has increased significantly the network traffic that the data center have to process. The increase of the traffic in the data centers has also resulted to higher power consumption. The server processors need to provide higher throughput without consuming excessive power. Currently, one of the main challenges for data centers operators is the power consumption of the servers that account for over 45% of the overall data center power consumption.

Modern server processors contain many levels of caching, forwarding and prediction logic to improve the efficiency of the traditional processor architecture; however the model is inherently sequential with performance limited by the speed at which data can move around this loop.

A dataflow computing model explicitly addresses this issue by minimizing and optimizing the flow of data. Current dataflow computing solutions utilise FPGA chip technology, which despite inherent inefficiencies have been shown to lead to orders of magnitude lower power consumption and lower

data center space needs. For example, recent journal publications ([5]) show reports from production use of dataflow computing. The delivery of a dataflow machine to JP Morgan, as part of an award-winning initiative described in the Wall Street Journal ([6]), yielded the computational power of 128 Teraflops (equivalent to over 12,000 high-end x86 control flow cores) within the space and power envelope of a single 40U rack.

Current dataflow engines implemented with FPGAs offer considerable advantages in performance and "performance per Watt". However, FPGAs are a general base technology which has significant limitations and is expensive in silicon area. For example, an FPGA is 18-35x less area efficient than an ASIC at implementing circuits, with a 3-4x higher critical path delay (i.e. decrease in clock frequency). Despite this cumulative 54-140x technology disadvantages, dataflow engines incorporating FPGAs have demonstrated high performance and energy efficiency for a broad range of applications due to the efficiency of the dataflow architecture. Given such encouraging results, we propose in VINEYARD the development of a custom dataflow servers optimized for high-performance power efficient implementations of data center applications that will maintain the capabilities of FPGAs to implement the dataflow computing paradigm while removing sources of inefficiency.

B. Programming framework

A key gap in the state of the art in this domain is the lack of a clean model to integrate the FPGA hardware-software stack with the language runtime system on the host servers. The gap exists from both a semantic and a resource management perspective. Questions on how accelerator code and state is managed by a high-level functional programming model and runtime system remain largely open. A task abstraction, presenting the accelerator as a versioned function to the programming model appears to be the most promising approach [7], but lacks transparency and breaks key desirable properties of functional parallel programming. Furthermore, scheduling, communication and synchronization in the runtime system are fundamentally influenced by the presence of accelerators. The integration of the accelerators with the data management technologies can be more natural due to the declarative nature of queries that can better exploit the data flow model to be implemented in the accelerators.

While bare-metal implementations of OpenCL and other high-level languages for FPGA accelerators have existed for some time [8], [9], [10], these implementations are localized and designed to support efficient translation to FPGA hardware rather than integration with the host software stack. The programmability of hardware accelerators (i.e. based on FPGAs) must improve if they are to be part of mainstream computing and data centers.

Combining a host-side programming model with the accelerator programming model in a hybrid solution is a challenging and rather inflexible proposition, both due to semantic conflicts (e.g. differences in memory models) and due to performance implications, notably contention between runtime systems for shared resources [11].

Furthermore, despite efforts to virtualize programmable and hardware accelerators, such as GPUs and FPGAs [12], [13], the virtualization methods deployed, notable pass-through and device drive level, introduce non-trivial performance interference within and between VMs. These are hardly traceable, let alone resolvable by programming models and runtime systems. VINEYARD aspires to address the open challenges in integrating programmable and hardware accelerators to the predominant software stacks used for data analytics in the Cloud:

(a) hide the accelerator from the programmer by presenting it as a pure library function, embeddable in query processing, data processing or aggregation tasks, and by extension to analytical libraries written on top of high-level programming models; (b) extend the runtime systems of high-level analytics languages to handle efficiently scheduling, communication, and synchronization with programmable accelerators; and (c) improve the performance robustness of analytics written in high-level languages against artifacts of virtualization, notably performance interference due to contention on shared resources and hidden noise in hypervisors and hosting VMs.

C. Virtualization and Middleware

In principle, the deployment model of accelerators in the data centers can take two basic but different forms: (a) Accelerators can be attached directly to servers and used by local workloads, or (b) accelerators can be shared over the network among many servers and their workloads.

The accelerators, whether GPUs, FPGAs or multicore CPUs, are assigned to tasks which they can perform more efficiently than general-purpose servers. The expected returns in cost, power and execution time are promising, but data movement is a strong challenge that undermines the whole proposition. Accelerators take data from the slow CPU path, process them in their customized hardware engines, and return the results either for storage in a file or direct to the memory system or for further processing by other accelerators or servers. The dominant programming frameworks in scale-out data centers have been streamlined to minimize the movement of data; thus, at the end of the day, the value of accelerator-based data centers will be weighed against the cost of the extra data copies that they introduce. With VINEYARD, we will speedup data communication through a system fabric that provides efficient communication primitives, to unify the accelerators with the servers and to reconcile them with the current computing frameworks.

Virtualization support is an additional, significant dimension of data center's infrastructures. Virtual Machines (and other similar types of technologies such as Containers) offer a mechanism for increasing consolidation of workloads on physical servers and achieving better utilization, isolating software versions and domains, and decoupling administrative domains i.e., clients from providers. Therefore, when examining the potential of accelerators in data centers, it is essential to deal

with the implications of, and to accrue the benefits from, Virtual Machines. We note that the presence of tenant VMs inside the data centers is orthogonal to accelerator virtualization [14]. VMs typically access the available hardware resources through a hypervisor, complicating the software segments of I/O stacks and increasing overheads. In addition, sharing the I/O paths among multiple VMs endangers isolation and quality-of-service. Clearly, sluggish and unreliable communication between VMs and accelerators impedes their co-existence in cloud data centers. Overall, in VINEYARD we will introduce a novel VM appliance model for provisioning of data to shared accelerators. Targeting cloud deployments, this VINEYARD effort can bring both tangible and novel results. The enhanced VINEYARD middleware augments the functionality of the orchestrator, by enabling more informed allocation of tasks to accelerators.

IV. VINEYARD Use Cases

The integrated data center that will be developed will be evaluated under three real-life workloads and industrial benchmarks for financial applications, data management, and bio-informatics. The first workload that will be evaluated will be in the domain of financial applications. For this reason the Greek Stock Exchange Market will be used as an end user demanding a) real-time analytics which are necessary for market surveillance and decision management and b) rapid computations for risk management, as an additional computation step within the trade process chain.

The second workload that will be evaluated will be in the domain of scientific computing, and more specifically in the domain of computational neuroscience which aims at better understanding the working of the human brain through use of mathematical models of biological neural networks. The particular application is a high-performance, high-accuracy simulation of the Olivocerebellar system of the brain, crucial to the understanding of cerebellar functionality. Better modeling and understanding of its function can lead to major breakthroughs in the treatment of cerebellum-related degenerative diseases (such as autism, fragile-X syndrome etc.).

The third workload will be based on data management based on TPC-C and TPC-H. TPC-C is representative of the transactional workloads run at operational databases of enterprises. It will be run on top of the LeanXcale OLTP database to represent the full stack of enterprise OLTP applications. TPC-H is representative of the analytical workloads run at data warehouses of enterprises. It will be run on top of the LeanXcale OLAP database to evaluate the efficiency improvements for analytical queries.

V. Conclusions

The main aim of the VINEYARD project is to develop a new framework for the efficient integration of accelerators into commercial data centers. The VINEYARD project will not only develop novel accelerator-based servers but will also develop all the required systems (hypervisor, middleware, APIs and libraries) that will allow the users to seamlessly utilize the accelerators as an additional cloud resource. The efficient utilization of accelerators in data centers will significantly improve the overall performance of the cloud applications and will also reduce the energy consumption in the data centers. Finally, VINEYARD aspires to foster the innovation of soft-IP accelerators in the domain of cloud computing by the promotion of a central repository for the hosting of the relevant accelerators.

References

[1] In *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 20142019 White Paper*.

[2] Hadi Esmaeilzadeh, Emily Blem, Renée St. Amant, Karthikeyan Sankaralingam, and Doug Burger. Power challenges may end the multicore era. *Commun. ACM*, 56(2):93–102, February 2013.

[3] Hadi Esmaeilzadeh, Emily Blem, Renee St. Amant, Karthikeyan Sankaralingam, and Doug Burger. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA '11, pages 365–376, New York, NY, USA, 2011. ACM.

[4] Nikos Hardavellas, Michael Ferdman, Babak Falsafi, and Anastasia Ailamaki. Toward dark silicon in servers. *IEEE Micro*, 31(4):6–15, July 2011.

[5] Olav Lindtjorn, Robert Clapp, Oliver Pell, Haohuan Fu, Michael Flynn, and Oskar Mencer. Beyond traditional microprocessors for geoscience high-performance computing applications. *IEEE Micro*, 31(2):41–49, 2011.

[6] In *Clark, D. Maxeler makes waves with dataflow design. Wall Street Journal Blog. 13 December 2011*.

[7] Javier Bueno, Xavier Martorell, Rosa M. Badia, Eduard Ayguadé, and Jesús Labarta. Implementing ompss support for regions of data in architectures with multiple address spaces. In *Proceedings of the 27th International ACM Conference on International Conference on Supercomputing*, ICS '13, pages 359–368, New York, NY, USA, 2013. ACM.

[8] Yi Shan, Bo Wang, Jing Yan, Yu Wang, Ningyi Xu, and Huazhong Yang. Fpmr: Mapreduce framework on fpga. In *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, FPGA '10, pages 93–102, New York, NY, USA, 2010. ACM.

[9] Peter Athanas, Krzysztof Kepa, and Kavya Shagrithaya. Enabling development of opencl applications on fpga platforms. In *Proceedings of the 2013 IEEE 24th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, ASAP '13, pages 26–30, Washington, DC, USA, 2013. IEEE Computer Society.

[10] Muhsen Owaida, Nikolaos Bellas, Konstantis Daloukas, and Christos D. Antonopoulos. Synthesis of platform architectures from opencl programs. In *Proceedings of the 2011 IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines*, FCCM '11, pages 186–193, Washington, DC, USA, 2011. IEEE Computer Society.

[11] Heidi Pan, Benjamin Hindman, and Krste Asanović. Composing parallel software efficiently with lithe. In *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '10, pages 376–387, New York, NY, USA, 2010. ACM.

[12] Michela Becchi, Kittisak Sajjapongse, Ian Graves, Adam Procter, Vignesh Ravi, and Srimat Chakradhar. A virtual memory based runtime to support multi-tenancy in clusters with gpus. In *Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '12, pages 97–108, New York, NY, USA, 2012. ACM.

[13] Wei Wang, Miodrag Bolic, and Jonathan Parri. pvfpga: Accessing an fpga-based hardware accelerator in a paravirtualized environment. In *Proceedings of the Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, CODES+ISSS '13, pages 10:1–10:9, Piscataway, NJ, USA, 2013. IEEE Press.

[14] Fei Chen, Yi Shan, Yu Zhang, Yu Wang, Hubertus Franke, Xiaotao Chang, and Kun Wang. Enabling fpgas in the cloud. In *Proceedings of the 11th ACM Conference on Computing Frontiers*, CF '14, pages 3:1–3:10, New York, NY, USA, 2014. ACM.