Versatile Reconfigurable Acceleration in Data Centers: a Bioinformatics Case Study

Nikolaos Alachiotis
nalachio@ics.forth.gr

Dimitris Theodoropoulos
dtheodor@ics.forth.gr

Dionisios Pnevmatikatos
pnevmati@ics.forth.gr

Computer Architecture and VLSI Systems Laboratory
Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
GR-70013 Heraklion - Crete, GREECE

ABSTRACT

Important design considerations for the cost-effective employment of hardware accelerators in next-generation data centers involve a) the type of candidate applications that a proposed solution can accelerate (generality), and b) the required development effort to successfully deploy the available accelerators for a given application (adoption overhead). To address the problem of generality, we present a versatile and dynamically reconfigurable hardware architecture that exhibits several accelerator slots and programmable interconnect to create application-specific accelerator datapaths. The proposed architecture fits in the model of disaggregated data centers, where compute, memory, and accelerators are broadly regarded as large pools of resources, and subsets of these resource pools are dynamically allocated on an as-needed basis to cooperatively boost performance of a broad range of applications. We discuss how applications can be efficiently mapped on the proposed underlying hardware using a highly compute-intensive Bioinformatics problem as a case study. Initial results for the Bioinformatics application, which deals with the detection of positive selection in populations, show a speedup of up to 6.4X in comparison with a parallel software implementation executed on a multi-core processor.

1. INTRODUCTION

With the end of Dennard scaling and the consequent interruption of processor frequency scaling due to an accompanying drastic increase of circuit power consumption, Moore's law is being continued in the form of multiple cores and wider vector registers. At the same time, the increased requirements of compute and/or memory demanding applications from various domains (e.g., bioinformatics, finance, etc) have led to the exploration of new processing approaches, e.g., on the cloud. Processing on the cloud is conducted on third-party—typically general-purpose—data centers, which

ENeSCE 2017 Stockholm, Sweden

DOI:

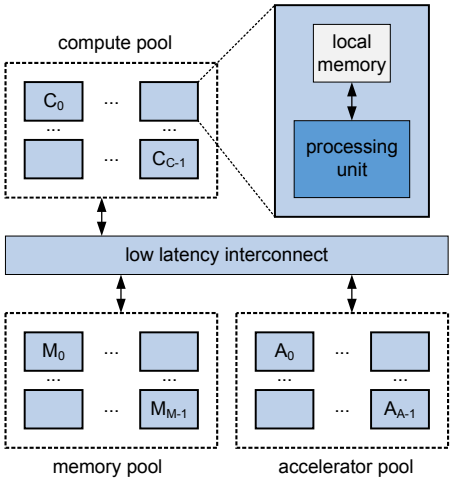

Figure 1: An abstract illustration of the concept of disaggregated computing. Resources are deployed as needed, driven by application requirements.

are traditionally organized as collections of trays with a fixed amount of compute, memory, and peripherals. Such organization can frequently cause suboptimal resource utilization at the tray level, leading to inefficient processing and increased energy consumption. To this end, an increasingly explored solution relies on the concept of data center disaggregation, where compute, memory, and acceleration resources are organized as large homogeneous pools (Fig. 1) that form specialized virtual machines (VMs) to better serve application-driven requirements.

With present computing systems being power-bound, emerging technologies like Graphics Processing Units (GPUs) and Field Programmable Gate Arrays (FPGAs) are explored as accelerators for deployment in data centers to reduce the Total Cost of Ownership (TCO) and maximize vendor profits. Acceleration-oriented solutions entail a trade-off between performance/power and generality/adoption overhead. Vector instruction extensions, for instance, such as Streaming SIMD Extensions and Advanced Vector Extensions, find application in various domains, from multimedia processing [7] and database operations [19] to cryptography [6] and scientific applications [10]. GPUs and FPGAs, on the other hand, represent more specialized solutions that potentially achieve higher performance and power savings, yet lacking generality

and requiring higher adoption effort. These factors (generality and ease of deployment) are of critical importance when such platforms are employed in data center environments.

To this end, we aim at improving generality by primarily targeting FPGAs, to benefit from their inherent reconfiguration capabilities, and by providing an architectural solution to enable a fit-for-purpose FPGA configuration, given that the required hardware accelerators exist and sufficient software support is provided. Our work builds upon a prior work [17] that addressed the problem of increasing generality of static accelerators in the logic layer of a 3D-stacked DRAM by providing support for a configurable accelerator-based datapath, as well as the required software translator for legacy codes that employ widely used library functions.

In this work, we make the following contributions:

- We present a reconfigurable architecture, which we dub dAccTiRDe (disagreggated Acceleration Tiles on Reconfigurable Device), that exhibits two levels of configuration capability: a) a placement of specialized accelerators in predefined partially reconfigurable regions (PRRs), and b) runtime-configurable crossbar switches to inexpensively interconnect accelerators.
- We adopt a master-worker scheme that facilitates disaggregation by managing application-driven resource reassignments in software, while accelerators serve as co-processors (workers) to software threads (masters). Therefore, a thread reassignment is transparently reassigning the dedicated accelerator accordingly.
- We use a compute and memory intensive Bioinformatics problem as a case study in order to describe alternative application mappings to dAccTiRDe, to demonstrate the versatility of the proposed solution. Note that, although we employ application-specific accelerators to showcase dAccTiRDe, the proposed architecture is sufficiently generic to accommodate specialized accelerators from various domains.

The choice of application as a case study is driven by the fact that genome sequencing throughput, fueled by continuous advances in DNA sequencing over the previous years, has long outpaced compute speed of modern processors. This has created an increasing gap between the amount of available genomic data and what can be efficiently analyzed (Fig. 2). Since compute performance follows Moore's law, it is only expected that this gap will continue to widen, yielding high-performance solutions a prerequisite for future large-scale analyses.

The Bioinformatics application, known as selective sweep detection, aims at finding traces of positive selection in populations through the analysis of patterns of correlation between mutations. From a computational standpoint, this heavily relies on population count operations, i.e., the enumeration of set bits in binary vectors. Current microprocessor architectures are inadequate to deliver high performance for applications that rely on bit counting due to the lack of a vectorized population counter, thus raising the need for custom-hardware-based acceleration.

The remainder of this paper is organized as follows: Section 2 revises related work and active research projects on acceleration infrastructure for data centers. Section 3 presents the proposed architecture, while Section 4 provides implementation details. Section 5 demonstrates different configurations of the proposed accelerator infrastructure for the selected Bioinformatics problem, and provides a performance evaluation. Finally, Section 6 concludes this work.

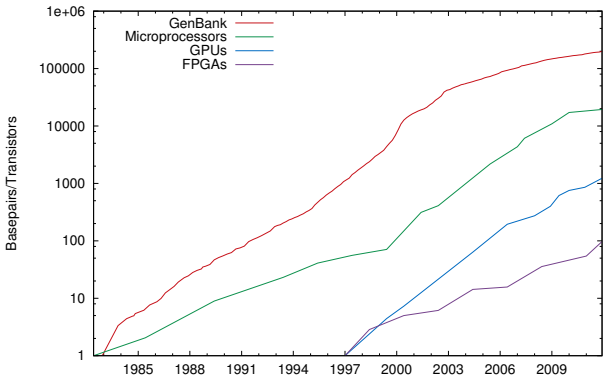

Figure 2: Growth of molecular data and transistors in processors, GPUs, and FPGAs (source: [5]).

2. RELATED WORK

Current research projects on next-generation data centers have proposed the employment of hardware accelerators to improve system performance. The Ecoscale project [11], for instance, proposes a novel architecture to automatically map and execute HPC applications to platforms that are supported by reconfigurable hardware modules. Reconfigurable resources are shared among the available processing nodes, which are mapped to a partitioned global address space.

Vineyard [8] develops an integrated platform for heterogeneous accelerator-based servers, aiming at achieving improved performance and reduced energy consumption when compared with current solutions. The project considers both low-power processors that are coupled with reconfigurable logic, as well as programmable dataflow engine accelerators that are configured by the host processor.

In the industrial domain, various companies offer HPC platforms that host powerful CPUs interconnected (loosely or tightly) with reconfigurable logic. For example, the HC series platforms from Micron [16] utilize QPI-connected x86 CPUs that are tightly coupled with an FPGA-based reconfigurable coprocessor. To invoke a hardware accelerator, the developer needs to extend the host CPU ISA by injecting assembly instructions to the original code, reducing that way the overall time-to-solution overhead.

Maxeler offers dataflow computing platforms [15] that consist of so-called DFEs (Data Flow Engines), i.e., reconfigurable accelerators that are either tightly coupled with local Intel Xeon CPUs or shared with a host CPU over Infiniband. A Java-like programming environment allows to implement custom accelerators and map them to the available DFEs.

Microsoft has already deployed its Catapult platform [2], which allows applications to utilize remote FPGAs at a data center scale. The proposed hardware plane consists of two QPI-connected x86 CPUs and an FPGA-based board, communicating over PCIe. The most recent work [1] presents a cloud-scale implementation based on Stratix V FPGAs, with applications in service (Web search ranking) and network (encryption) acceleration.

Intel has launched the HARP (Intel-Altera Heterogeneous Architecture Research Platform) project [12], which tightly

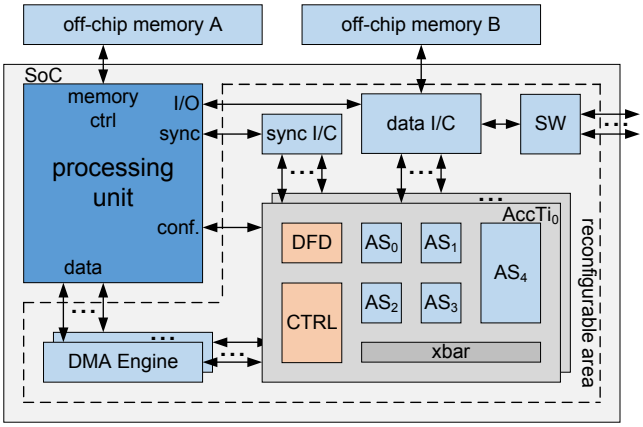

Figure 3: Overview of the dAccTiRDe MPSoC architecture.

couples Intel's Xeon CPUs with an Altera Stratix V FPGA connected over a cache-coherent QPI on a 2U standard server module. HARP provides a high-level programming framework that allows developers to automatically map OpenCL kernels to the underlying hardware, or to accelerate C/C++ applications using custom Verilog modules.

IBM proposes the CAPI architecture [18] for offloading algorithms to an FPGA. Applications mapped to the CAPI architecture are executed on a IBM Power8 CPU and can coherently exchange data with user-specific hardware accelerators implemented to an Altera Stratix V FPGA.

As already mentioned, resource disaggregation is increasingly being adopted as a mean to improve performance and reduce Total Cost of Ownership (TCO) of next-generation data centers. In Microsoft's Catapult platform, for instance, each processing node can allocate and map hardware accelerators to idle disaggregated FPGAs[1]. Another approach to resource disaggregation, the one explored in the dRedBox research project [13], focuses on a low-power data center architecture that shifts toward a flexible and software-defined block-as-a-unit paradigm. It introduces three baseline disaggregated building blocks, i.e, an MPSoC module, a high-performance memory module, and an FPGA-based module for hosting application-specific accelerators. Based on application needs, a full-fledged Type-1 hypervisor allocates the required modules and executes commodity VMs.

3. THE dAccTiRDe ARCHITECTURE

The dAccTiRDe architecture supports task-based programming environments by offloading computationally demanding tasks to application-specific hardware accelerators. Figure 3 depicts its high-level structure; dAccTiRDe considers a multi-core processing unit (PU) that is coupled with reconfigurable area (RA). Both the PU and the RA are directly connected to dedicated off-chip memories A and B, respectively. The PU runs a multi-threaded middleware that is responsible for configuring the application-specific accelerators in the RA via the "conf" port, as well as executing tasks that were not mapped onto hardware due to resource and/or performance considerations.

The RA is divided into two main parts, namely the static and the dynamic ones. The static area hosts the required hardware infrastructure for efficient communication and data exchange with the PU. It consists of two discrete interconnects (I/C), a set of DMA engines, and an array of Acceleration Tiles (AccTis) to map custom accelerators. The "sync I/C" allows the PU to initiate operations on an AccTi and monitor the progress of the accelerators. A DMA engine is connected to each AccTi in order to efficiently access off-chip memory A through the PU. The "data I/C" arbitrates access to off-chip memory among all AccTis, the PU, and a local network switch (SW). The latter is responsible for communication and data transfers between the local system and remote processing nodes.

Table 1: Resource utilization for dAccTiRDe's static area when mapped to the Zynq 7045 MPSoC.

Module	Quantity	BRAMs	LUTs
PL memory IF	1	1	6,923
DMA Engine	2	0	5,830
DMA I/C	2	0	1,540
data I/C	1	0	2,110
sync I/C	1	0	1,940
switch (duplex/4 lanes)	1	4	2,290
PS reset	1	0	30
AccTi CTRL, DFD, Xbar	2	4	3,754
Total		9	24,417
Utilization (%)		1.65%	11.17%

The dynamic part of dAccTiRDe mainly consists of Accelerator Slots (AS). Each AccTi comprises several AS slots, along with a set of configurable crossbar switches, and a Data Fetch and Dispatch mechanism (DFD) to facilitate all incoming and outgoing data transfers. The AS slots are partially reconfigurable regions (PRRs) that can be occupied by accelerators both at bootstrap time and at run time. PRRs of various sizes allow the placement of resource demanding accelerators and can facilitate the exploration of potential trade offs between performance and energy savings by dynamically switching off PRRs.

The configurable crossbar switches allow to construct a custom datapath in each AccTi by appropriately interconnecting AS slots. When an AccTi is configured, it serves as a co-processor (worker) for a software thread (master) executed on the PU. The crossbar switch configuration is initiated by the master thread, which sends a configuration instruction to the AccTi's control unit (CNTRL) by appropriately initializing a set of control registers.

The proposed MPSoC, comprising a multi-core processing unit and a series of AccTis, with various AS slots each, yields the efficient mapping of an application a challenge. As already mentioned, each AccTi serves as a coprocessor to a master thread, allowing to treat the problem of maximizing acceleration performance similarly to improving parallel performance on multi-core processors. Scalable parallel algorithms exhibit increased computation to synchronization ratios, leading to longer execution time intervals between synchronization events. This translates to longer periods of uninterrupted processing on the assigned AccTi when the coprocessor is deployed, consequently leading to improved pipeline performance and making such algorithms highly suitable for acceleration on dAccTiRDe.

4. AN EARLY EVALUATION ON ZYNQ

To evaluate our work, we mapped dAccTiRDe to a Xil-

inx ZC706 board with a Zynq 7045 MPSoC that tightly couples a dual-core ARM Cortex A9 as a Processing System (PS) and Kintex 7 Programmable Logic (PL). Our prototype comprises 2 AccTis, where each AS can be reconfigured by the PS via the PCAP port at a rate of 440 MB/sec. Table 1 provides the resources occupied by the static dAccTiRDe infrastructure. The "PL memory IF" module is implemented using the Xilinx Memory Interface Solutions; it provides an AXI-compatible interface to on-chip modules for data accesses to and from the external memory that is directly accessible from the PL, achieving an effective bandwidth of up to 6.1 GB/sec (64 bytes per clock cycle at 100 MHz). Each AccTi drives a DMA engine that is connected to one of the high-performance (HP) ports of the PS in order to access in a non-coherent way the external memory that is directly accessible from the PS. This achieves a bandwidth of up to 800 MB/sec (8 bytes per clock cycle at 100 MHz).

All internal system interconnects, i.e., data I/C, sync I/C, and DMA I/C, are implemented with the Xilinx AXI interconnect module. Finally, in our current implementation, the internal switch employs a 4-lane interconnect that is based on the Aurora link-layer protocol for high-speed serial communication with all remote processing nodes. This achieves a maximum communication speed of up to 50 Gb/sec, i.e., 12.5 Gb/sec per lane.

5. A BIOINFORMATICS APPLICATION

To evaluate dAccTiRDe, we employ a computationally intensive Bioinformatics problem, generally known as selective sweep detection. Detecting selective sweeps has practical applications, such as the detection of mutations in pathogens that are resistant to drugs (e.g., HIV[9]), and therefore can potentially lead to the design of more effective drug treatments. However, as dataset sizes increase, due to advances in DNA sequencing technologies, computational challenges arise in efficiently conducting large-scale sweep detection. In this section, we describe a selective sweep (5.1), outline a method for sweep detection (5.2), and present and evaluate alternative application mapping approaches (5.3).

5.1 Selective Sweep

A selective sweep is the elimination of variation in a genomic region as a result of positive selection acting on a population. When a beneficial mutation occurs in a population, it increases the chances of survival and reproduction of the individuals that carry the mutation. Eventually, all individuals will possess the beneficial mutation and, due to forces that drive the evolution of complex organisms, the frequency of occurrence of linked neutral mutations in the proximity of the beneficial mutation will decrease. This phenomenon leaves distinctive signatures in the genomes, which facilitate the detection of selective sweeps.

5.2 Detection Method

To detect sweeps in a population, a multiple sequence alignment (MSA) is required. An MSA is essentially a $m \times n$ data matrix of m DNA sequences with n nucleotides each, as shown below.

```
sample_0    ATGGCATACCCCT-CCAACTAGGATTCCAA
sample_1    ATGGCCTACCACTCCCAACTAGGCTTCC-A
sample_2    ATGGCATAC-C-TCCCAAGTAGGTTTTC-A
```

Only alignment columns that comprise two or more nucleotides are informative for sweep detection, and are referred to as single-nucleotide polymorphisms (SNPs, highlighted in blue and italics).

A commonly used sweep signature relies on patterns of linkage disequilibrium (LD) between SNPs, which captures the non-random associations between mutations in different locations in the genome. SNPs are typically represented as bit vectors of size N bits, with set bits denoting mutations and N being the number of sequences. An LD score between two SNPs, i and j, with bit vectors v_i and v_j, respectively, is computed using the Pearson's correlation coefficient:

$$r_{ij}^2 = \frac{[P(v_i \& v_j) - P(v_i)P(v_j)]^2}{P(v_i)P(v_j)(1 - P(v_i)(1 - P(v_j)}, \quad (1)$$

where $P(v_x)$ is the fraction of set bits in v_x or, in other words, the mutation frequency at location x in the genome.

Here, we describe the OmegaPlus [4] approach for LD-based sweep detection, which organizes large genomes into a number G of overlapping windows of variable number of SNPs, and proceeds by analyzing each window following a three-stage algorithm. Initially, all pairwise LD scores are computed and stored in a lower diagonal matrix of floating-point values (LD stage). Thereafter, a dynamic programming algorithm computes total LD scores for all possible subwindows with different numbers of SNPs (DP stage). Finally, the ω statistic [14] is employed to evaluate each window, aiming at detecting the location in the genome with the highest ω score (ω stage). Given a window of W SNPs, divided into two non-overlapping subwindows, L and R, with l and $W - l$ SNPs, respectively, the ω statistic is computed for all possible combinations of subwindows L and R with different numbers of SNPs as follows:

$$\omega = \frac{(\binom{l}{2} + \binom{W-l}{2})^{-1}(\sum_{i,j \in L} r_{ij}^2 + \sum_{i,j \in R} r_{ij}^2)}{(l(W-l))^{-1} \sum_{i \in L, j \in R} r_{ij}^2}. \quad (2)$$

Note that, prior to an analysis, all monomorphic alignment columns are discarded, and the remaining SNPs are stored in a compact representation [5].

5.3 Application Mapping

Algorithm. Parallel algorithms that exhibit an increased computation to synchronization ratio on a multi-core processor map well on dAccTiRDe. OmegaPlus exhibits alternative parallelization strategies [3] to address load imbalance issues that arise from the SNP density variance along the genomes. These algorithmic approaches have been previously evaluated for parallel performance [3], employing a range of different dataset features and combinations thereof (number of sequences, numbers of SNPs, and SNP density variance). The study revealed that the so-called generic algorithm, implemented in OmegaPlus-G, outperforms the rest of the parallelization alternatives for most dataset cases as the number of cores increases. The algorithm organizes all input data into chunks of SNPs and constructs a compute list of same-size tasks, with each task entailing the pairwise combination of SNP chunks. Thereafter, all threads consume tasks from the list, leading to increased parallel performance due to a favorable computation to synchronization ratio that is achieved because of the fixed compute load per task.

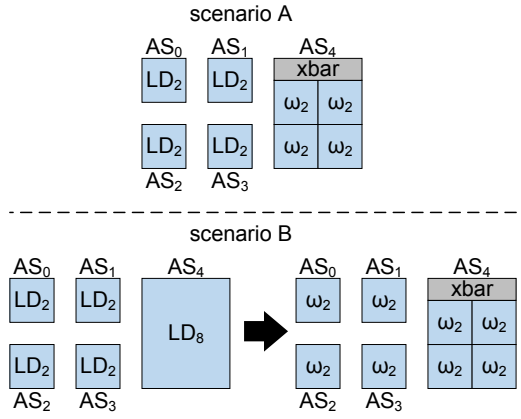

Figure 4: Alternative mapping scenarios for OmegaPlus-G on dAccTiRDe; scenario A places multiple instances of both accelerators in each AccTi, while scenario B relies on a fixed crossbar switch configuration, and dynamically replaces accelerators during execution using partial reconfiguration.

We find that the generic algorithm maps well to dAccTiRDe for the following reasons:

- The input SNP data are organized into SNP chunks of fixed size (e.g., 32 SNPs per chunk), allowing to trivially distribute genomic regions to threads, and thereby to the respective AccTis.
- Each compute task, i.e., operations on sets of SNP chunks, requires a fixed and reduced number of operations, leading to low resource requirements for the respective accelerators, thus facilitating their placement in the AS slots.

Accelerators. Computing all pairwise LD scores (LD stage) and evaluating all possible windows with the ω statistic (ω stage) dominate analysis times of OmegaPlus, with the actual contribution of each stage depending on the aforementioned dataset features (see OmegaPlus-G execution times per stage in Table 2). Therefore, we opt to employ accelerators for the LD and ω stages, while the intermediate DP stage is executed on the processing unit. The LD accelerator [5] is automatically generated by an RTL generation software (*https://github.com/alachins/fpga-ld*), while the ω accelerator, consisting of two identical floating-point pipelines, double buffers for input, and a control unit, is partially derived from the source code of OmegaPlus using Vivado HLS.

Mapping Approaches. To demonstrate the versatility of dAccTiRDe, we describe and evaluate two mapping alternatives for OmegaPlus-G. In our current implementation on the Zynq 7045 MPSoC, each AccTi hosts 5 AS; 4 occupy 3% and 1 occupies 32% of the total available resources. The first approach places multiple instances of both accelerators in each AccTi, as illustrated in Figure 4A, and relies on the rapid configuration of the crossbar switches at runtime to selectively deploy each of the two sets of accelerator instances. The second approach relies on maintaining a fixed crossbar switch configuration, and dynamically replace accelerators using the partial reconfiguration feature of dAccTiRDe.

Table 2: Execution times (in seconds), throughput performance (in parentheses, million LD scores per second for LD, and million ω scores per second for ω) per stage, and speedups for the two alternative mappings of OmegaPlus-G on dAccTiRDe in comparison with execution on a 6-core Xeon processor.

	Dataset (#sequences : #SNPs)		
	[A] 10^3:10^4	[B] 10^4:10^4	[C] 10^4:10^5
OmegaPlus-G			
LD	2.6 (14.54)	19.1 (2.63)	308.2 (2.89)
ω	32.9 (85.63)	21.5 (74.22)	7.1 (75.03)
Total	35.9	43.7	358.7
Accel. Map A			
LD	0.78 (48.39)	9.89 (5.09)	175.1 (5.08)
ω	4.41 (639.29)	2.66 (600.09)	0.81 (660.17)
Total/Speedup	5.59 / 6.42X	15.65 / 2.79X	218.8 / 1.64X
Accel. Map B			
LD	0.57 (65.76)	6.67 (7.52)	126.8 (7.03)
ω	5.84 (481.99)	3.89 (409.47)	1.37 (389.65)
Total/Speedup	6.81 / 5.27X	13.66 / 3.19X	171.57 / 2.09X

Here, all AS slots are either occupied with LD or ω cores, as shown in Figure 4B. In the figure, the subscripts (2 and 8 for LD, and 2 for ω) denote the number of floating-point pipelines that operate in parallel in every core.

Table 2 provides a performance comparison between the software benchmark (OmegaPlus-G) and the two application mappings on dAccTiRDe (Accel. Map A and Accel. Map B) for three different dataset sizes, ranging from 1,000 to 10,000 sequences, and from 10,000 to 100,000 SNPs. A workstation with an Intel Xeon E5-2630 6-core Sandy Bridge processor, running at 2.60 GHz and 32 GB of main memory, was deployed as a test platform for the software benchmark. The execution times in Table 2 refer to parallel execution of OmegaPlus-G employing 12 threads, while 2 threads (with an AccTi per thread) were launched on dAccTiRDe.

As can be observed in the table, partial reconfiguration improves performance of the larger datasets only, since the reconfiguration overhead is amortized by the longer execution times. Note however that the effect of partial reconfiguration on performance varies according to the number of partial reconfiguration events that are required throughout execution. Datasets A and B require a single reconfiguration event to complete the analysis. However, dataset C requires 5 events due to the dataset increased memory footprint and the limited amount of off-chip memory resources on the test platform.

Furthermore, filling up all AS slots with the same accelerator does not necessarily translate to better per-stage performance, as can be observed by a comparison of the ω throughput values between Accel. Map A and Accel. Map B. This is due to the fact that performance is bounded by memory rather than compute, leading to further deterioration of overall system performance for the ω stage when the memory bandwidth is distributed to more accelerator instances.

6. CONCLUSIONS

Current trends show that next-generation data centers

increasingly adopt resource disaggregation and/or tightly-coupled hardware accelerators. Our work can enhance performance and energy consumption of today's data centers; we presented a versatile reconfigurable accelerator architecture with sufficient hardware support to extend the concept of resource disaggregation to the reconfigurable fabric level. Our solution, dubbed dAccTiRDe, shifts the complexity of dynamic resource allocation to the software, while hardware resources are dedicated to increase generality and flexibility of the acceleration platform via configurable crossbar switches and partial runtime reconfiguration.

Driven by the increasing demands of large-scale biological analyses, we employed a computationally intensive Bioinformatics kernel to test dAccTiRDe and discuss alternative ways to map applications on it. Acceleration is achieved via a dedicated so-called Acceleration Tile per software thread launched on a multi-core processor of the proposed MPSoC, with each Acceleration Tile serving as a co-processor to the software thread. Increased overall performance is achieved when the parallel execution of the algorithm on the MPSoC's multi-core processing system exhibits high computation to synchronization ratio, which consequently translates to longer time intervals of uninterrupted operation for the accelerators, and thus increased pipeline performance.

Our future work includes a high-level software API that will allow developers to easily map applications to the dAccTiRDe architecture and exploit the available reconfigurable resources. Finally, we plan to analyze large sets of data from various application domains, such as network analytics and video surveillance, in order to evaluate dAccTiRDe with different data center workloads.

7. ACKNOWLEDGMENTS

This work is supported in part by EU H2020 ICT project dRedBox, contract #687632. The authors thank Gregory Chrysos, George Charitopoulos, and Iosif Koidis for constructive discussions and comments on the manuscript.

8. REFERENCES

[1] A. Caulfield, et. al. A cloud-scale acceleration architecture. In *Proceedings of the 49th Annual IEEE/ACM International Symposium on Microarchitecture*, 2016.

[2] A. Putnam, et.al. A reconfigurable fabric for accelerating large-scale datacenter services. In *Proceeding of the 41st Annual International Symposium on Computer Architecuture*, pages 13–24, 2014.

[3] N. Alachiotis and P. Pavlidis. Scalable linkage-disequilibrium-based selective sweep detection: a performance guide. *GigaScience*, 5(1):1, 2016.

[4] N. Alachiotis, A. Stamatakis, and P. Pavlidis. OmegaPlus: a scalable tool for rapid detection of selective sweeps in whole-genome datasets. *Bioinformatics*, 28(17):2274–2275, 2012.

[5] N. Alachiotis and G. Weisz. High Performance Linkage Disequilibrium: FPGAs Hold the Key. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 118–127. ACM, 2016.

[6] K. Aoki, F. Hoshino, T. Kobayashi, and H. Oguro. Elliptic curve arithmetic using SIMD. In *International Conference on Information Security*, pages 235–247. Springer, 2001.

[7] G. Bernabé, J. M. García, and J. González. Reducing 3D fast wavelet transform execution time using blocking and the streaming SIMD extensions. *Journal of VLSI signal processing systems for signal, image and video technology*, 41(2):209–223, 2005.

[8] C. Kachris, et. al. The VINEYARD approach: Versatile, Integrated, Accelerator-based, Heterogeneous Data Centres. In *International Symposium on Applied Reconfigurable Computing (ARC 2016)*, pages 3–13, 2016.

[9] N. G. De Groot and R. E. Bontrop. The HIV-1 pandemic: does the selective sweep in chimpanzees mirror humankind's future? *Retrovirology*, 10(1):1, 2013.

[10] M. Farrar. Striped Smith–Waterman speeds database searches six times over other SIMD implementations. *Bioinformatics*, 23(2):156–161, 2007.

[11] I. Mavroidis, et.al. ECOSCALE: Reconfigurable computing and runtime system for future exascale systems. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 696–701, 2016.

[12] Intel Corporation. Intel QuickAssist Technology. http://www.intel.com/content/www/us/en/embedded/technology/quickassist/overview.html. [Online; accessed 31-Oct-2016].

[13] K. Katrinis, et. al. Rack-scale disaggregated cloud data centers: The dReDBox project vision. In *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 690–695, 2016.

[14] Y. Kim and R. Nielsen. Linkage disequilibrium as a signature of selective sweeps. *Genetics*, 167(3):1513–1524, 2004.

[15] Maxeler technologies. MPC Series. https://www.maxeler.com/products. [Online; accessed 31-Oct-2016].

[16] Micron Technology, Inc. Convey HC2. https://www.micron.com/about/about-the-convey-computer-acquisition/hc-series. [Online; accessed 31-Oct-2016].

[17] Q. Guo, et. al. Enabling portable energy efficiency with memory accelerated library. In *Proceedings of the 48th International Symposium on Microarchitecture*, pages 750–761. ACM, 2015.

[18] B. Wile. Coherent Accelerator Processor Interface (CAPI) for POWER8 Systems. Technical report, 2014.

[19] J. Zhou and K. A. Ross. Implementing database operations using SIMD instructions. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 145–156. ACM, 2002.