D02: Data center acceleration through hardware extensions

28/09/2018

INTRODUCTION

The continuous growth of the size of data is asking hardware to constantly evolve. Until now, data centers effectively addressed this issue relying on traditional multiprocessors and multicore architectures. However, with Moore's Law approaching its scaling limits, this strategy is no longer viable [4]. Also, accelerating this kind of structures is a challenging task due to the fact that they deal with objectives that are often hard to optimize at the same time: high performance, energy efficiency, flexibility and low cost. Consequently, data centers are looking for alternative approaches, often involving the use of accelerators, usually in the form of hardware extensions used as co-processors to speed up the execution of computationally intensive portions of code. In this report, we will give an overview of the available architecture designs and discuss their advantages and drawbacks in order to propose potential solutions to optimize some of the aforementioned goals.

1 ACCELERATORS

As previously stated, data centers historically exploited homogeneous chip architectures. As opposed to this, *heterogeneous* chip architectures have cores with different micro-architectures integrating GPUs, FPGAs or ASICs on a chip, so that applications can find a better match between the different components in order to improve efficiency. In this section, we survey the main kinds of accelerators and give some examples of their use in data centers.

1.1 GPGPUs

It is well known that, due to their relatively low cost, massively parallel architecture and constantly improving ease of use provided by programming environments such as the NVIDIA CUDA framework, GPUs have become extremely popular as general purpose computing devices.

General-Purpose GPUs stand out because of their high arithmetic power, a result of a very specialized architecture, conceived to extract the maximum performance on the highly parallel tasks of traditional computer graphics. This makes them very well suited to accelerate a wide range of massively parallel applications: in fact, GPUs have been used extensively for data analytics and big data applications. In particular, data mining and machine learning algorithms are accelerated in many libraries, such as Caffe [4]. Some graph applications have also been accelerated via GPUs [6]. These are the reasons why GPUs are already among the most common accelerators and IBM and NVIDIA are collaborating on their further integration in data centers [4].

Even so, GPUs are not necessarily to be considered the best option for a data center, as they are best at accelerating applications with regular computational patterns. This is due to the fact that GPUs, from an architectural point of view, according to Flynn's taxonomy, belong to the SIMD (Single Instruction Multiple Data) class, so that they are better suited for data parallelism than for task parallelism [6]. In particular, when dealing with GPUs, it is extremely important to consider which data structures are involved in the computations requiring acceleration: in general, GPUs will perform well on arrays and dense matrices, while they are less suited for operating on irregular data structures such as graphs and trees, even though recent work has shown that GPUs can also

achieve significant speedups on certain more irregular algorithms through both recent architectural improvements and software optimizations [5].

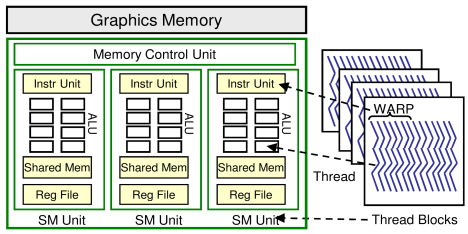

Fig. 1. GPU architecture and CUDA thread execution model.

1.2 FPGAs

Heterogeneous computing systems are gaining in popularity to meet the growing demand for energy-efficient high-performance computing. Within the various heterogeneous systems, the CPU-FPGA platform is recognized as one of the most promising systems thanks to the advantages provided by FPGAs in terms of high performance, low power consumption and reconfigurability for the realization of the various acceleration functions.

A strong point for FPGAs is their interface flexibility, allowing to connect them to any other device through any physical interface. In addition to that, the recent integration of programmable logic with CPUs has been a significant benefit for FPGAs. Moreover, FPGAs are meant to be used for concurrent fixed-point operations, with a close-to-hardware programming approach, taking full advantage of bit-wise operations. Unlike GPUs, FPGAs have a deterministic latency of the order of one nanosecond. This makes these accelerators very beneficial for applications which need to control their latency, such as audio encoding or network synchronization [9].

All these elements make FPGAs more and more popular today, and have become an integral part of clusters and data centers.

But FPGAs face many challenges in this area, first because data center servers often host multiple applications, and may require a separate accelerator for each. Second, multiple objectives can be combined over time, ranging from server-side energy efficiency to application latency constraints and workload processing. In addition, many data center applications need tighter control over their execution and several service levels, consequently requiring internal monitoring of these accelerators [1].

However, the use of heterogeneous CPU-FPGA architectures seems very promising. For example, Microsoft has deployed custom FPGA boards (called Catapult) in its data center to work with the CPU as an accelerator [2]. This has improved the efficiency of the page ranking rate of the Bing search engine by 2x with only 10% more power, reflecting the potential of CPU-FPGA platforms [3].

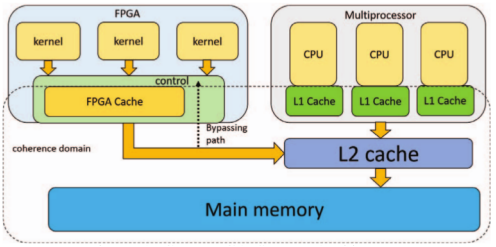

Fig. 2. CPU-FPGA system architecture [2].

1.3 ASICs

ASICs (Application Specific Integrated Circuits) are chips specifically developed for an application. They are designed for a single purpose and will operate in the same way throughout their lifetime. As these chips are fully customized, they require high development costs for their design and implementation. Unlike FPGAs, ASICs are not reprogrammable, but are much denser and can integrate several different features into a single chip. ASICs therefore provide a compact and energy efficient solution [12] [13].

1.4 Embedded accelerators

In data centers, it is also common to make use of embedded accelerators: high-performance general purpose CPUs, in fact, often include vector instruction set extensions to enable SIMD style of computation. SSE (Streaming SIMD Extensions) and the more recent AVX (Advanced Vector Extensions) are two examples of such kind of extended ISAs for Intel and AMD architecture CPUs [8] [10].

In principle, there is no extra effort required to benefit from such extended instruction sets, as standard GNU C/C++ compilers support them with optimization options. It is also possible, however, to significantly improve performance via manual SIMD programming [10].

A real-world example of AVX usage for big data acceleration is the IBM DB2 database, result of cooperation between IBM and Intel itself [4].

2 COMPARISON

FPGAs and GPUs are not straightforward to compare in terms of performance because, as previously stated, FPGAs are designed to perform well on fixed-point operations, so that measuring their performance in GFLOPS is less relevant than it is for GPUs. The performance of FPGAs is therefore usually measured in GMACS (Giga Multiply-Accumulate Operations). A comparison in terms of GFLOPS, anyway, would certainly be in favor of GPUs. As an example, we compare two high-end 2016 devices in terms of single core processing power, the Sapphire Radeon R9 Fury X GPU is more than twice as efficient then the FPGA Virtex-7 690T (7,168 GFLOPS vs 3,120 GFLOPS) [9]. As ASICs are designed for specific purposes, it is quite difficult to generalize their performance. However, for the tasks for which ASICs have been designed, they significantly outperform GPUs. For example, Google estimates that its Tensor Processing Unit (TPU), a custom ASIC designed for Machine Learning [14], is 15 to 30 times faster than contemporary GPUs [15]. On the other hand, when it comes to latency, FPGAs and ASICs provide deterministic timing, while GPUs don't.

Raw performance, anyway, is not necessarily the only parameter to consider as a guideline for technology selection: accessibility is also important. From this point of view, GPUs are certainly more convenient due to the existence of frameworks that make it possible to write portable and

backward compatible programs in high-level languages. Also, as GPUs are already commonly used, there is a wide range of algorithms directly designed for them. For their part, FPGAs and ASICs require specialized developers and engineers in order to efficiently re-design the algorithms and to make use of the device flexibility, both in terms of reconfigurability and interfaces.

Model	**Approx. price**	**Price efficiency**
Sapphire Radeon R9 Fury X (GPU)	600 €	0.08 €/GFLOPS
Virtex-7 690T (FPGA)	11'200 €	3.59 €/GFLOPS

Table 1. Cost comparison between equivalent models of FPGA and GPU [9]

Another important cost that should not be overlooked when it comes to ASICs is the design and development of the chips, which can cost up to millions of dollars [12].

	Unit cost	**NRE**
FPGAs	8 $	0 $
ASICs	4 $	1.5M $

Table 2. Comparison between FPGAs and ASICs in terms of unit and NRE (Non-Recurring Engineering) cost [13].

The cost of the engineering effort required by FPGAs and ASICs adds up to the price of the devices themselves. This appears to be a huge drawback for these kinds of chip [9].

It is worth mentioning, though, that even if FPGAs and ASICs seem economically demanding at first glance, their cost may be balanced by a drastically lower power consumption in the long term, in comparison to GPUs. To a lesser extent, FPGAs remain less energy efficient than ASICs, and require more power for the same tasks [12]. Moreover, such power efficiency allows the implementation of current FPGAs and ASICs in a compact hardware with more than reasonable thermal dissipation and cooling requirements.

Model	**Power efficiency**
Sapphire Radeon R9 Fury X (GPU)	20 GFLOPS/W
Virtex-7 690T (FPGA)	78 GFLOPS/W

Table 3. Power comparison between equivalent models of FPGA and GPU [9].

As shown in the above table, FPGAs are three to four times better than GPUs when it comes to power efficiency.

3 CONCLUSIONS

Both GPUs and FPGAs are viable options to address the problem of accelerating data centers, depending on the specific requirements and constraints.

The great popularity of GPUs as accelerators is mostly due to their remarkable cost efficiency and availability. Because of this, such devices are probably the best choice for most existing data centers in need of speeding up their applications within a reasonable time, avoiding major rearrangements both in terms of hardware and software. Even in this case, it is important to consider what kind of application has to be optimized and ensure that a SIMD style of parallelism is appropriate for the purpose.

Although FPGAs seem much more expensive than GPUs, they are a long-term cost-effective investment, especially due to their low energy consumption, which makes them a first-class choice for projects that are expected to be carried out over the long term. This solution also provides huge interface flexibility, allowing them to be used with almost any other type of device.

Despite their price and their development costs, ASICs are certainly worth considering when it comes to data centers dealing with specific tasks such as Machine Learning and Cryptomining. Although superior in terms of performance, considering the fact that most of such applications rapidly grow and change, the lack of flexibility that characterizes ASICs may lead to a preference for FPGAs [11].

In the the next few years, FPGAs are expected to become dominant due to their good balance between performance and energy efficiency, which is and will remain a crucial concern. In medium-long term, it will also be interesting to keep an eye on the ASICs industry and see what new hardware components will be released.

Nonetheless, hardware extensions are intended to be used alongside regular CPUs, so that in any case the usage of ISA extensions, which is implicit whenever a CPU supports them and comes with no additional costs, is beneficial in terms of performance, even without any dedicated manual optimization.

REFERENCES

[1] Dimitrios Mbakoyiannis, Othon Tomoutzoglou, and George Kornaros, "Energy-Performance Considerations for Data Offloading to FPGA-Based Accelerators Over PCIe", in *ACM Transactions on Architecture and Code Optimization*, Vol. 15, No. 1, Article 14 , March 2018.

[2] Liang Feng, Sharad Sinha, Wei Zhang and Yun Liang, "A Hybrid Approach to Cache Management in Heterogeneous CPU-FPGA Platforms ", in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, November 2017.

[3] Chao Wang, Xi Li and Xuehai Zhou, "SODA: Software Defined FPGA based Accelerators for Big Data", in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, March 2015.

[4] Serif Yesil, Muhammet Mustafa Ozdal , Taemin Kim, Andrey Ayupov, Steven Burns and Ozcan Ozturkfi, "Hardware Accelerator Design for Data Centers", in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 770-775, November 2015.

[5] J. Y. Kim and C. Batten, "Accelerating irregular algorithms on GPGPUs using fine-grain hardware worklists", in *Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on*, pp. 75fi?!87, Dec 2014.

[6] Sungpack Hong, Sang Kyun Kim, Tayo Oguntebi and Kunle Olukotun,"Accelerating CUDA Graph Algorithms at Maximum Warp", in *Proceedings of the 16th ACM symposium on Principles and practice of parallel programming*, pages 267-276, February 2011.

[7] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens Krger, Aaron E. Lefohn and Timothy J. Purcell, "A Survey of General-Purpose Computation on Graphics Hardware", in *Computer graphics forum*, March 2005.

[8] Chris Lomont, "Introduction to Intel Advanced Vector Extensions", Intel white paper, available: https://software.intel.com/en-us/articles/introduction-to-intel-advanced-vector-extensions, March 2011.

[9] Berten Digital Signal Processing, "GPU vs FPGA Performance Comparison", Berten white paper, available: http://www.bertendsp.com/pdf/whitepaper/BWP001_GPU_vs_FPGA_Performance_Comparison_v1.0.pdf, May 2016.

[10] Hwancheol Jeong, Weonjong Lee, "Performance of SSE and AVX Instruction Sets", *Proceedings of Science, The 30th International Symposium on Lattice Field Theory*, June 2012.

[11] Lynnette Reese, "Comparing Hardware for Artificial Intelligence: FPGAs vs. GPUs vs. ASICs", available: http://eecatalog.com/intel/2018/07/24/comparing-hardware-for-artificial-intelligence-fpgas-vs-gpus-vs-asics/

[12] Rohit Singh, "FPGA Vs ASIC: Differences Between Them And Which One To Use?", available https://numato.com/blog/differences-between-fpga-and-asics/

[13] "FPGA vs ASIC, What to Choose?", available: https://anysilicon.com/fpga-vs-asic-choose/

[14] "Google supercharges machine learning tasks with TPU custom chip", available: https://cloud.google.com/blog/products/gcp/google-supercharges-machine-learning-tasks-with-custom-chip

[15] "Quantifying the performance of the TPU, our first machine learning chip", available: https://cloud.google.com/blog/products/gcp/quantifying-the-performance-of-the-tpu-our-first-machine-learning-chip

A SUMMARY OF CHANGES

Reviewer #1. According to the first review, the survey was not entirely exhaustive with regards to ISA extensions. We addressed this issue by adding some information about how accessible this solution is in the section dedicated to embedded acceleration and by giving some hints on how to make use of such solution in the conclusions, while a proper comparison with hardware extensions did not seem appropriate, both for the nature of ISA extensions and because, as the reviewer himself observes, we did not mean to examine this solution in too much detail.

The conclusions of this updated version of the report also include a more explicit forecast, even though we also highlight the fact that the best solution may vary according to the specific sub-scenario.

Reviewer #2. The second review pointed out the fact that in the first version some references were missing, especially in the introduction, tables and section about GPGPUs. Even though the bibliography was already complete, there was in fact a lack of explicit citations, which we have added in this updated version. In particular, when it comes to Moore's law, it was not our intention to state that it has not been fairly accurate so far: we think it is not relevant in order to do a forecast due to the fact it is well known that it is *approaching* its limits. In order to clarify that, we added one more reference and opted for a clearer lexicon.

The major difference between the current version and the previous one, however, is the newly added section about ASICs. At first, we had decided not to cover that topic as it seemed too domain-specific, but on the other hand, the comparison between FPGAs and ASICs suggested by the reviewer seemed definitely on point.

In the current version, as requested, the conclusions also include a more explicit forecast.

To conclude, we try to informally answer an interesting question posed by the reviewer: *will there be a new type of accelerator in the near future?* While it's hard for us and certainly beyond the scope of the project to conceive an entirely new type of accelerator, it is likely that we will see ASICs for other purposes in the next few years.