Field Programmable Gate Arrays

How FPGAs will change the hardware accelerator landscape

Abstract

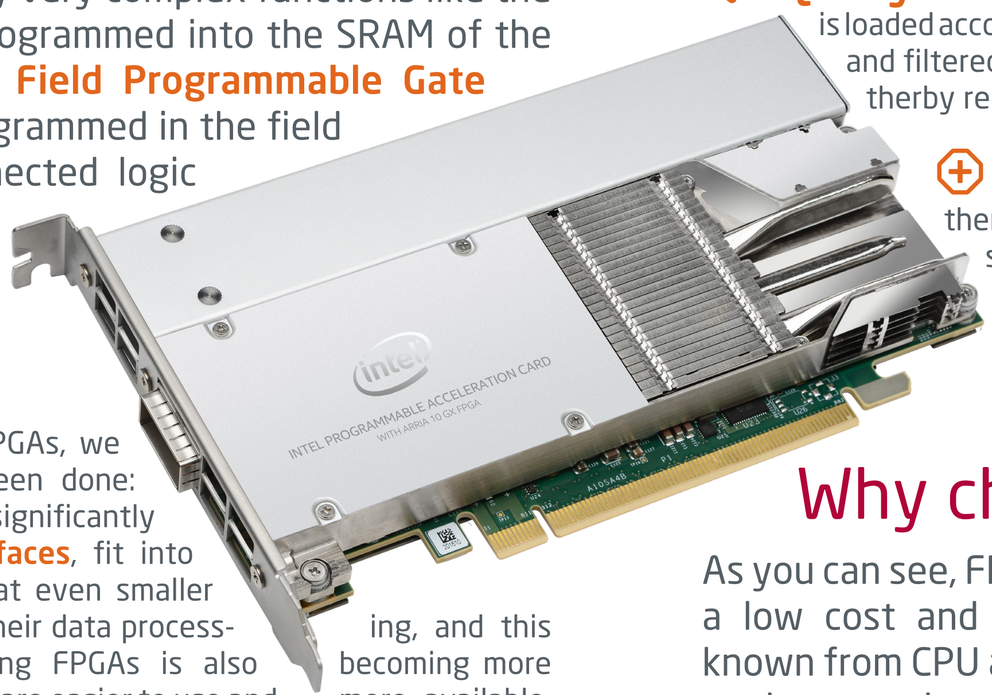
FPGAs are, in simple terms, customizable (that is, programmable) processors that can be used to **perform any task**. This includes in especially **optimizations of database** operations. However, for a long time they were too expensive for most data warehouses - this is currently changing as there are more and more manufacturers offering FPGA hardware, software and FPGA interfaces. In the following, you will find a more **detailed explanation of what FPGAs** are, how and why you might want to integrate them into your data pipeline, and the fundamental differences to **other hardware accelerators**.

What are FPGAs and hardware accelerators

First of all hardware accelerators are exactly what the word suggests, they **accelerate the operation of a computer** or server by hardware. Generally all calculating parts in a computer are hardware. accelerators; That includes CPU, GPU, ASIC (Integrated Circuits) and FPGAs. In the Table on the right you can see, that there are many good reasons to settle on a FPGA. Nevertheless, what precisely is an FPGA? An FPGA consists of **many individual cells arranged in an array**. These cells consist mainly of a table that assigns an output to each combination of inputs. Such lookup table can be used to **hard-code any logical function**. These cells then are linked together in a configurable way. That way very complex functions like the database optimizations can be programmed into the SRAM of the lookup tables. Hence the name: **Field Programmable Gate Array**, because they can be reprogrammed in the field and consist of an array of connected logic gates.

A glimpse at the future of FPGAs

If we want to estimate the future of FPGAs, we must first look at what has already been done: In recent years, FPGAs have become significantly **cheaper** and, thanks to **standard interfaces**, fit into (almost) any server rack. This means that even smaller companies and start-ups can accelerate their data processing, and this trend will certainly continue,programming FPGAs is also becoming more and more simple, as **developer tools**/API's are easier to use and more available. There are also successful approaches to translate programs from classical programming languages to FPGAs ("**high-level synthesis**"). In addition, there are more and more resources available for learning to rethink from "data movement" to "data flow" paradigms. Once a program is ready to run on an FPGA, it is easier than ever to roll it out and actually run it, since many traditional and new **Cloud Hosters** are equipping servers with FPGAs that can be rented on a per-minute basis. Another exciting prospect for the future is the **integration of FPGAs into other hardware components** such as SSDs ("Samsung SmartSSD"); this way the interface between memory and FPGA is even faster and only one hardware component needs to be installed. Furthermore, the FPGA is also increasingly spreading outside the data warehouses in **network interfaces** or in the **maker community**. These trends will not disappear in the future, and if you now consider how FPGAs also excel in other fields such as artificial intelligence, there is no doubt that FPGAs will soon be found in many more racks



How can FPGAs enhance your Data-Pipeline?

What can data accelerators do for your data pipeline - where can FPGAs support the work of the DBM system? The most important optimizations based on the example of the Swarm64 DA:

Optimized Columns: ...are used to improve data layout - especialy for handling dates, quantities, sensor data, ... - that is for columns which are often queried by range. The more optimized columns are used, the faster the query becomes and less space / IO-transfer is needed. This performance increase is further amplified by the fact that the CPU is available for other tasks - since the FPGA executes the work in parallel.

Query: The Query gets send completly to the FPGA, where the data is loaded according to the FROM statement and then scanned (SELECT) and filtered (FROM) in parallel. Only the result is send back to CPU, therby reducing the CPU processing needed.

Insert: The FPGA compresses the inserted data; thereby, instead of updating the table and indices in a row store or preparing the data and updating the columns in a column store, in such a way, that an acceleration of up to 20 times is possible

Why choose FPGAs?

As you can see, FPGAs combine the performance of ASICs with a low cost and faster development-cycles, originally only known from CPU and GPU. The only downside is the developer expierence, since the range of software tools and experienced developers is rather limited.

	FPGA	ASIC	GPU	CPU
Performance	◔	◔	◔	◔
Flexibility	🔧	-	🔧🔧	🔧🔧
Dev. Experience	⊖	⊖⊖	⊕	⊕⊕
Time-To-Market	🕒🕒	🕒🕒🕒	🕒	🕒
Cost	$	$$$	$	$$

Special thanks for the talk go to

swarm64

References

- www.intel.com/content/dam/altera-www/global/en_US/pdfs/literature/ds/ds-pac-a10.pdf
- swarm64.com
- www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html

Jannis Rosenbaum

Bachelor-Student
IT-Systems Engineering | Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3 I D-14482 Potsdam
E-Mail: jannis.rosenbaum@student.hpi.de
Web: jannisrosenbaum.de

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam