FPGA-Accelerated Analytics: From Single Nodes to Clusters
Other titles in Foundations and Trends® in Databases

*Algorithmic Aspects of Parallel Data Processing*
Paraschos Koutris, Semih Salihoglu and Dan Suciu

*Data Infrastructure for Medical Research*
Thomas Heinis and Anastasia Ailamaki

*Main Memory Database Systems*
Franz Faerber, Alfons Kemper, Per-Ake Larson, Justin Levandoski, Thomas Neumann and Andrew Pavlo

*Query Processing on Probabilistic Data: A Survey*
Guy Van den Broeck and Dan Suciu
ISBN: 978-1-68083-314-0

*Big Graph Analytics Platforms*
Da Yan, Yingyi Bu, Yuanyuan Tian and Amol Deshpande
978-1-68083-242-6
FPGA-Accelerated Analytics: From Single Nodes to Clusters

Zsolt István
IMDEA Software Institute
Spain
zsolt.istvan@imdea.org

Kaan Kara
Oracle Labs
Switzerland
kaan.kara@oracle.com

David Sidler
Microsoft Corporation
USA
david.sidler@microsoft.com
Editorial Scope

Topics

Foundations and Trends® in Databases publishes survey and tutorial articles in the following topics:

- Data Models and Query Languages
- Query Processing and Optimization
- Storage, Access Methods, and Indexing
- Transaction Management, Concurrency Control and Recovery
- Deductive Databases
- Parallel and Distributed Database Systems
- Database Design and Tuning
- Metadata Management
- Object Management
- Trigger Processing and Active Databases
- Data Mining and OLAP
- Approximate and Interactive Query Processing
- Data Warehousing
- Adaptive Query Processing
- Data Stream Management
- Search and Query Integration
- XML and Semi-Structured Data
- Web Services and Middleware
- Data Integration and Exchange
- Private and Secure Data Management
- Peer-to-Peer, Sensornet and Mobile Data Management
- Scientific and Spatial Data Management
- Data Brokering and Publish/Subscribe
- Data Cleaning and Information Extraction
- Probabilistic Data Management

Information for Librarians

Foundations and Trends® in Databases, 2020, Volume 9, 4 issues. ISSN paper version 1931-7883. ISSN online version 1931-7891. Also available as a combined paper and online subscription.
## Contents

1 Introduction .................................................. 2

2 Background .................................................. 6  
   2.1 What Are FPGAs? ................................. 6  
   2.2 FPGAs vs. ASICs ............................... 9  
   2.3 Where to Deploy Acceleration? .......... 11

3 FPGAs in Datacenters and Clouds ...................... 14  
   3.1 The Path to Adoption ....................... 14  
   3.2 Interfacing with FPGA Accelerators .... 16  
   3.3 Database Integration Case Studies ........ 23

4 Designing Accelerators with FPGAs .................... 32  
   4.1 Design Guidelines ............................ 33  
   4.2 Core SQL: Like and Regular Expressions .... 35  
   4.3 Core SQL: Group-by Aggregation ............ 42  
   4.4 Core SQL: Where Predicates and Filtering ... 52  
   4.5 Machine Learning: K-Means Clustering ....... 56  
   4.6 Machine Learning: Stochastic Gradient Descent 65  
   4.7 Distributed Joins: Data Partitioning ....... 79
5 Future Challenges and Architectures 86
   5.1 Datacenter and Cloud Architecture Trends . . . . . . . 86
   5.2 Device Trends . . . . . . . . . . . . . . . . . . . . . . 87
   5.3 Sharing FPGAs and Accelerators . . . . . . . . . . . 89
   5.4 Programming FPGAs . . . . . . . . . . . . . . . . . . . 91

6 Closing Remarks 93

References 94
FPGA-Accelerated Analytics: From Single Nodes to Clusters

Zsolt István¹, Kaan Kara² and David Sidler³

¹IMDEA Software Institute, Spain; zsolt.istvan@imdea.org
²Oracle Labs, Switzerland; kaan.kara@oracle.com
³Microsoft Corporation, USA; david.sidler@microsoft.com

ABSTRACT

In this monograph, we survey recent research on using reconfigurable hardware accelerators, namely, Field Programmable Gate Arrays (FPGAs), to accelerate analytical processing. Such accelerators are being adopted as a way of overcoming the recent stagnation in CPU performance because they can implement algorithms differently from traditional CPUs, breaking traditional trade-offs. As such, it is timely to discuss their benefits in the context of analytical processing, both as an accelerator within a single node database and as part of distributed data analytics pipelines. We present guidelines for accelerator design in both scenarios, as well as, examples of integration within full-fledged Relational Databases. We do so through the prism of recent research projects that explore how emerging compute-intensive operations in databases can benefit from FPGAs. Finally, we highlight future research challenges in programmability and integration, and cover architectural trends that are propelling the rapid adoption of accelerators in datacenters and the cloud.
Introduction

Big Data has been instrumental to our lives in the last decade and has lead to scientific insights, the boom of Machine Learning and the emergence of novel on-line services. Datacenters that are hosting such data-intensive applications are facing however an important challenge: the amount of data that needs to be stored and processed is increasing at an exponential rate whereas traditional processor performance has been stagnating for years. As Moore’s law and Dennard scaling taper off, CPU transistor counts are growing less year-on-year and, due to heat dissipation issues, it is becoming challenging to power on all parts of CPUs at high clock rates at the same time, resulting in “dark silicon”. The stagnation of CPU performance, however, presents opportunities as well: CPUs can now incorporate more heterogeneous components without having to keep them powered on at all times. In addition, for many workloads, replacing CPUs partially or entirely with specialized hardware has become the more economic choice. These devices can utilize transistors more efficiently for the tasks at hand, delivering more performance for the same energy budget.

Driven by the above described trends, and in order to keep up with growing data sizes, data processing and management applications have
been becoming increasingly distributed. Today, applications built with platforms such as Apache Spark, routinely span hundreds of server nodes. While helpful in reducing compute bottlenecks, this distribution brings new data movement bottlenecks at various levels of the software and hardware architecture. In this monograph we focus on how specialized hardware accelerators can provide an answer to the compute stagnation problem and showing how they can be also helpful in reducing data movement bottlenecks by placing them in the right location within the computer architecture.

There are many different technologies one could use for building accelerators but one particular technology stands out as a middle ground between energy efficiency and versatility: Field Programmable Gate Arrays (FPGAs). FPGAs make it possible to express algorithms in ways that are fundamentally different from CPUs or GPUs: FPGAs have no instruction sets and functionality is laid out directly as circuitry. As opposed to traditional CPUs that concentrate on-chip memory into layers of caches, FPGAs have small SRAM memories distributed throughout the device that can be configured flexibly and can be co-located with computation. These differences to CPUs result in a higher performance in the same or lower energy footprint, making FPGAs attractive both as accelerators and as energy-efficient replacements of software solutions. FPGAs are reconfigurable meaning that they can implement different functionality over time and can be reprogrammed from software. Once programmed, they act as integrated circuits (ASICs), bringing significant improvements in energy efficiency when compared to CPUs.

Today, FPGAs are available in most clouds (e.g., Amazon F1 Instances, n.d.; Firestone, 2016; Putnam, 2014; Weerasinghe et al., 2016) as accelerators. As a prominent example, the Microsoft Catapult project uses FPGAs to create programmable network-interface cards (NICs) to offload tenant network functions from the CPU. Meanwhile, programmable co-processors based on FPGAs are also becoming more common, for instance, with projects such as Intel Xeon+FPGA (Gupta, 2015). Storage devices/nodes are also increasingly more programmable,
with examples such as Samsung’s SmartSSD\textsuperscript{1} and Amazon Redshift AQUA.

Several early projects of FPGA-based database acceleration proposed PCIe-attached accelerator cards a decade ago, demonstrating that FPGAs are able to speed up projection and selection (Dennl et al., 2012; Salami et al., 2017; Sukhwani et al., 2013; Wang et al., 2016a; Woods et al., 2013), aggregation (Dennl et al., 2013; Salami et al., 2017), joins and even sorting (Casper and Olukotun, 2014; Sukhwani et al., 2013; Zhang et al., 2016), by an order of magnitude when compared to commonly used row-stores, such as MySQL and PostgreSQL. However, in many previous systems, once all integration costs were factored in, in particular the cost of data transfers over PCIe, the benefits were significantly reduced. In the meantime, however, the bandwidth of interconnects (PCIe, NVMe, etc.) and networks has been increasing at a steady pace, making FPGA acceleration once again an attractive option. Furthermore, due to increased data sizes, there are more and more distributed databases that suffer from various data movement bottlenecks, for instance, when retrieving data from distributed storage nodes or when shuffling tuples between compute nodes for a join operation. Specialized hardware can be used to move computation closer to the source, and by reducing data sizes through filtering or transformations close to storage, memory, etc., reduce data movement bottlenecks.

In this monograph we explore what is required for integrating these accelerators with software systems (focusing on database management systems) and how one should design the acceleration functionality to ensure that the overall speedups in the system are worth the additional complexity. The monograph provides a detailed look into several representative examples of FPGA-accelerated databases and the internals of accelerated SQL operators, Machine Learning operators and data shuffling operators.

\textbf{Monograph Structure.} In this section (Section 1), we outlined the reasons why FPGAs and similar specialized hardware have become not only economically feasible for database acceleration but, in many

\textsuperscript{1}https://samsungsemiconductor-us.com/smartssd/index.html.
cases, even necessary to keep up with data growth. We also provided an intuition on how, thanks to a fundamentally different execution model from CPUs and GPUs, these devices offer benefits in efficiency. The rest of the monograph is structured as follows:

- In Section 2 we present a background on FPGAs and highlight their differences with ASICs. Readers already familiar with FPGAs and their development work flow might consider skipping this section.

- Section 3 covers salient aspects of integrating FPGAs in data processing systems, in various locations of the software and hardware architecture. We describe several representative examples of how FPGAs are integrated with databases and what features the frameworks that enable this need to provide.

- In Section 4 we turn our attention to individual operators from the domain of core SQL acceleration, Machine Learning and Distributed Joins. We present both high level guidelines to help future FPGA programmers design circuits adequate to the task and deep dive into the design and performance characteristics of representative operators, implemented by the authors.

- Section 5 describes the remaining challenges related to programming and sharing FPGA accelerators more easily in datacenters and the cloud. This section also provides a peek into the future of re-programmable hardware and the opportunities this will bring for database management systems.
Achronix Vectorpath with Speedster7t FPGAs (n.d.). https://www.achronix.com/vectorpath/.
Amazon F1 Instances (n.d.). aws.amazon.com/ec2/instance-types/f1/.


Pirk, H., J. Giceva, and P. Pietzuch (2019). “Thriving in the no man’s land between compilers and databases”.


