

Информационные технологии

Увеличение быстродействия операций в СУБД с помощью аппаратных ускорителей FPGA

А.В. СОЛОВЬЕВ

Федеральный исследовательский центр «Информатика и управление»
Российской академии наук, г. Москва, Россия

Аннотация. В статье выполнен обзор способа ускорения операций обработки данных в СУБД с помощью FPGA. Выполнен обзор возможностей FPGA, архитектур использования FPGA, рассмотрены практические примеры использования FPGA. Определены достоинства и недостатки применения FPGA. Отмечен ряд проблем и альтернативных технологий, которые существенно загромождали применение FPGA в промышленных СУБД и системах хранения данных. Применение FPGA для ускорения обработки данных в СУБД, можно утверждать, что подход позволяет ускорять операции в БД, связанные с обработкой запросов, сжатием и шифрованием данных, параллельной обработкой данных. Однако, применение FPGA также усложняет систему в целом, не позволяет гибко и быстро перенастраивать функциональные возможности системы, увеличивает общую стоимость владения. В дальнейших исследованиях планируется рассмотреть альтернативные технологии ускорения операций обработки данных.

Ключевые слова: системы управления базами данных, СУБД, FPGA, ускорения операций обработки данных.

DOI: 10.14357/20790279250208 **EDN:** XQCEYD

Введение

В условиях цифровой трансформации экономики и общества критичными становятся практически все операции добавления новых и обработки хранящихся данных для поддержания высокой согласованности данных с одновременной высокой доступностью этих данных.

На практике это означает, что самые последние изменения в данных должны быть доступны в режиме онлайн всем пользователям, имеющим права доступа к этим данным.

В этой связи, ведущие производители СУБД стараются не только повысить производительность своих продуктов, функционирующих на стандартных компьютерах и серверах, но и по возможности ускорить ряд операций с помощью различных аппаратных ускорителей. К таким аппаратным

ускорителям относят сопроцессорные устройства, например, GPU (Graphics Processing Unit (графический процессор)) и DPU (Data Processing Unit (сoproцессор для обработки данных)), ставшими популярными за счет возможности разгрузить CPU (Central Processing Unit (центральный процессор)).

Ряд производителей пытаются выполнить ускорения ряда операций с помощью интересных, но вызывающих дискуссии об эффективности применения RISC-V (расширяемая открытая и свободная система команд и процессорная архитектура на основе концепции RISC (Reduced Instruction Set Computer – архитектура процессора, в которой быстродействие увеличивается за счёт упрощения инструкций)).

Данное исследование нацелено на обзор способа ускорения операций в СУБД с помощью



Рис. 1. Схема архитектуры «в пути»

FPGA (Field-Programmable Gate Array (программируемая пользователем вентиляционная матрица)). Перейдем непосредственно к обзору возможностей FPGA, архитектур их использования, а также рассмотрению практических примеров использования FPGA.

1. Краткий обзор источников, основных областей и архитектур применения FPGA

Исследование источников, посвященных применению FPGA, показывает высокую активность научной дискуссии и актуальность темы аппаратного ускорения операций в СУБД с помощью FPGA. Так, например, из обзора источников [1-6] можно сделать вывод о том, что FPGA в настоящее время активно используется для ускорения любых вычислительных процедур, а также для следующих областей:

- реализация алгоритмов искусственного интеллекта (ИИ);
- реализация алгоритмов машинного обучения (ML);
- реализация нейронных сетей «на кристалле»;
- реализация аппаратных модулей безопасности;
- реализация алгоритмов шифрования.

Согласно современным исследованиям (см., например, [1-6]), к достоинствам использования FPGA относят:

- возможность аппаратной реализации определенных вычислений и алгоритмов;
- возможность разгрузки CPU за счет выполнения вычислений и алгоритмов на отдельном вычислительном устройстве.

В современной литературе (см., например, [7]) выделяют три основных архитектуры применения FPGA:

- «в пути»;
- «на стороне»;
- «сопроцессор».

При реализации архитектуры «в пути», FPGA размещается между хранилищем (БД) и CPU, как показано на рис. 1.

Некоторые примеры реализации данной архитектуры:

- Netezza (машина БД на FPGA, см. [8]): FPGA применялись для эффективного аппаратного выполнения операций сжатия БД. Над упакованным сегментом БД матрица FPGA в Netezza умела выполнять проекцию и фильтрацию, а все прочие операции, включая агрегации и группировки, осуществлялись CPU;
- SmartNIC и платформа ExtraV: FPGA выполняет функцию фильтра сетевого трафика. Применяя сжатие/декомпрессию или дедупликацию, FPGA значительно повышают эффективную пропускную способность приложений связи между сетью и хранилищем (БД).

Достоинства применения архитектуры «в пути»:

- исследования показывают (см., [7, 8]), что даже без выполнения распаковки, выполняя только фильтрацию и агрегирование на FPGA, можно значительно уменьшить объем данных, отправляемых на CPU;
- эффективно использовать архитектуру для приложений с обработкой потоков данных, чувствительной к задержкам.

При реализации архитектуры «на стороне», FPGA взаимодействует с основной памятью хоста через память дополнительных устройств или через дополнительное соединение, например, с помощью PCIe, как показано на рис. 2.



Рис. 2. Схема архитектуры «на стороне»

В данной архитектуре FPGA выполняет роль ускорителя операций ввода-вывода. Примеры реализации архитектуры:

- Kickfire's MySQL Analytic Appliance. CPU соединен с FPGA через PCIe;

- Xtremedata's dbX: FPGA совместима по контактам с CPU.

К достоинствам архитектуры относят:

- возможность ускорения ключевых операций работы с данными БД: объединения, сортировки, группировки, агрегирование.

К недостаткам относят:

- требуются дополнительные копии данных (например, данных из памяти хоста в память устройства или наоборот), что приводит к увеличению времени обработки данных.

Однако, если вся БД может быть размещена в памяти (In-Memory DB, см. [7]), применение данной архитектуры может уменьшить количество копий данных для обработки.

При реализации архитектуры «сопроцессор», FPGA либо соединена с CPU через общую память (рис. 3), либо интегрирована вместе с CPU.

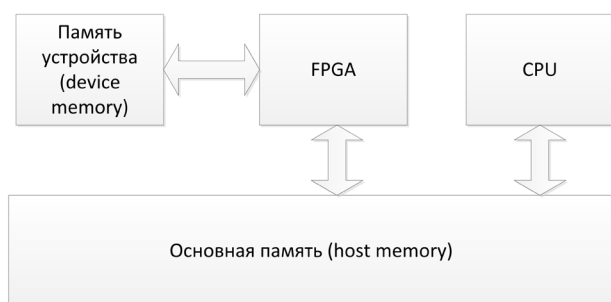


Рис. 3. Схема архитектуры «сопроцессор»

Примеры реализации архитектуры:

- DorrioDB: ускорение общих операций работы с БД (объединение, сортировка, компрессия и декомпрессия данных).

Достоинства применения данной архитектуры:

- FPGA может обращаться к памяти хоста напрямую, что не потребует дополнительных копий данных;
- связь между CPU и FPGA осуществляется через общую память, что позволяет избежать задержек при копировании данных, т.к. доступ к памяти хоста в качестве разделяемой памяти поможет избежать копирования данных из / в память устройства;
- при размещении на одном кристалле FPGA и CPU возможно существенное ускорение обработки данных за счет разделения операций.

Недостатки:

- при размещении на одном кристалле FPGA и CPU крайне трудно синхронизировать производительность устройств, т.к. FPGA значительно уступает CPU в производительности.

Экспериментальные исследования по применению FPGA в СУБД показывают, что за счет FPGA достигается:

- ускорение обработки данных In-memory БД. В частности, исследования (см., например, [7]) показывают, что теоретически можно ускорить работу потоковых операторов, сортировки и сопоставления с регулярными выражениями;
- ускорение алгоритмов сжатия данных в БД (см., например, [9]), FPGA-реализация алгоритма сжатия на основе LZMA (Lempel-Ziv-Markov chain-Algorithm (алгоритм основан на использовании последовательных повторений данных для достижения высокой степени сжатия)), позволяет сжимать 100+ Гб данных в 10-20 раз быстрее, чем без использования FPGA.

Но это теоретико-экспериментальные исследования, а как обстоят дела с применением FPGA в промышленных СУБД? Приведем краткие результаты обзора таких применений.

2. Обзор применения FPGA в промышленных СУБД

Примером достаточного универсального аппаратного ускорителя можно считать SQL2FPGA (см., например, [10]). В основе устройства лежит принцип автоматического ускорения обработки SQL-запросов с помощью использования платы Alveo U280 FPGA на базе AMD/Xilinx HBM. Тестирование показало, что использование SQL2FPGA обеспечивает в среднем ускорение производительности от 10 (для небольших баз данных размером до 1 Гб) до 14 раз (для БД размером 30 Гб) по всем 22 тестовым запросам теста производительности TPC-H (стандартный тест для тестирования производительности сложных аналитических запросов, см. подробнее <https://tpc.org/tpch/>).

Следующим примером может служить платформа Centaur, которая предназначена для совместного использования CPU-FPGA в архитектуре «сопроцессор» с целью ускорения выполнения запросов к БД. Платформа позволяет существенно ускорять построение планов запросов за счет динамического распределения операторов с помощью FPGA для планов запросов, а затем за счет конвейерной передачи операторов и гибридное (CPU-FPGA) выполнение операторов запросов (см. подробнее [11]).

Еще одним примером, хотя и находящимся на стадии разработки можно считать ускоритель выполнения запросов Sailfish (см. [12]). Принцип функционирования Sailfish предположительно позволяет преодолеть недостаток использования архитектуры «сопроцессор», а именно тот факт, что оптимизаторам запросов практически невозможно распределить согласованные вычислительные

ресурсы, чтобы обеспечить одинаковую производительность обработки на уровне CPU и FPGA. Sailfish позволяет регулировать вычислительные ресурсы времени выполнения на стороне CPU, чтобы соответствовать производительности обработки на стороне FPGA. Кроме того, в Sailfish реализован ускоритель FPGA, который поддерживает параллельную обработку хэш-соединения. Интегрируя с PostgreSQL разработанный FPGA-ускоритель Sailfish в экспериментальных тестах показывает значительное превосходство производительности по сравнению с встроенной подсистемой оптимизации запросов PostgreSQL.

FPGA используется в ускорителе Swarm64, специально разработанном для повышения производительности выполнения запросов для СУБД PostgreSQL, MySQL, MariaDB. Swarm64 встраивается между подсистемой хранения и подсистемой обработки данных, сжимает БД в сжатый столбцовый формат, что позволяет экономить на памяти для работы с данными и ускорять выполнение аналитических запросов (см. [8]).

Примером полностью промышленного использования FPGA можно считать разработку Alibaba X-DB компании Alibaba Cloud (Китай). Alibaba X-DB создана на основе MySQL и использует FPGA для ускорения выполнения запросов. Alibaba X-DB дает эффект на транзакционных нагрузках. FPGA задействованы в упаковке LSM-дерева, что освобождает CPU и оставляет ему непосредственно транзакционную работу. На одном узле распределенной СУБД с ускорителем FPGA на TPC-C-подобном тесте получено около 15% прироста производительности по сравнению с обычной реализацией без FPGA (см. [8]).

Примером «железной» машины БД выступает разработка Netezza (приобретена компанией IBM в 2010 году, см. [8]). Матрицы FPGA применялись для эффективного фактически аппаратного выполнения операций сжатия данных БД. Над упакованным сегментом БД Netezza умела выполнять операции обработки данных для запросов: проекции и фильтрации. Все прочие операции, включая агрегации и группировки, осуществлялись с помощью CPU. Несмотря на успехи проекта, IBM прекратила выпуск Netezza.

Еще одним примером промышленного использования FPGA является отечественная ERP-система 1С:Предприятие 8.3. Для ускорения обработки больших массивов данных БД используется FPGA Xilinx Alveo U280 для PostgreSQL v.11 на ОС Ubuntu (см. [13]). Ускорение обработки данных достигается за счет распараллеливания обработки аналогично многоядерной схеме обработки на

CPU. Вообще же принцип распараллеливания вычислений с помощью FPGA не нов, так, например, возможно ускорения выполнения программ за счёт параллельного выполнения программ на FPGA (см., например, [14]).

Безусловно, примером промышленного использования FPGA является проект AQUA компании Amazon для СУБД Redshift (см. подробнее [15]).

По сути, AQUA – это распределенный кэш с аппаратным ускорением для узлов СУБД Redshift. AQUA использует чипы AWS Nitro, адаптированные для ускорения шифрования и сжатия данных, а также записи данных на SSD для аппаратного ускорения сжатия, шифрования и обработки (поиска, агрегирование и фильтрация) данных «рядом» с хранилищем данных. Обеспечивается примерно десятикратное ускорение запросов, выполняющих сканирование (поиск), агрегирование и фильтрацию данных с помощью SQL-предикатов LIKE и SIMILAR_TO. Использование архитектуры AQUA (рис. 4) позволяет увеличить количество обращений к кэшу с 99,60% до 99,95% при отправке на узел СУБД Redshift критически важных запросов.

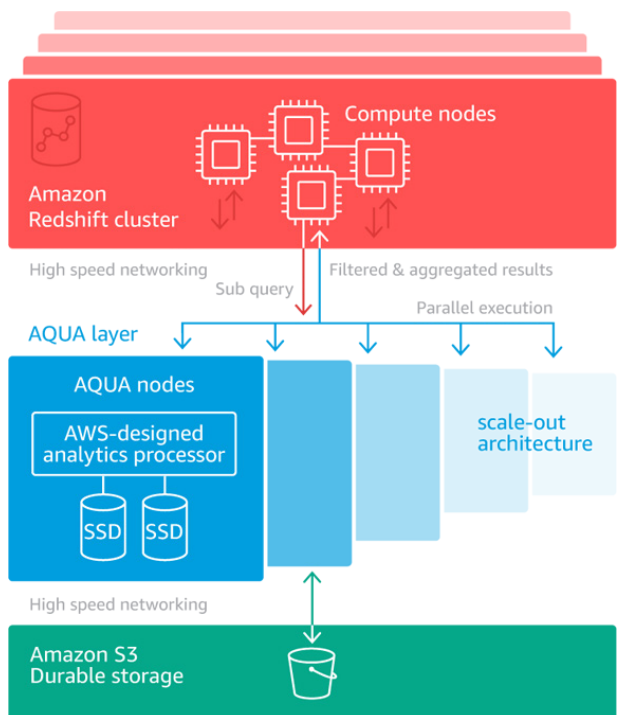


Рис. 4. Архитектура AQUA для СУБД Redshift

Если в проекте AQUA можно увидеть удачное сочетание аппаратного ускорения операций обработки данных и использования высокоскоростных накопителей SSD, то следующим проектом, который может быть использован для ускорения операций обработки данных может служить разработка

компании Samsung – SmartSSD, в состав которого включен FPGA.

Его отличительной особенностью от обычных твердотельных дисков (SSD) является возможность обрабатывать данные напрямую, сводя к минимуму обмен данными между процессором, оперативной памятью, графическим процессором и т.д., что приводит к значительному увеличению производительности и повышению энергоэффективности. Подробнее о SmartSSD см., например, [16-18].

Поэтому, теоретически, SmartSSD может использоваться для обработки больших массивов данных, алгоритмов поиска (в том числе с помощью искусственного интеллекта, машинного обучения).

По оценке Samsung при использовании SmartSSD время обработки запросов к большим БД сокращается примерно в 2 раза, при этом энергопотребление уменьшается на 70%, загруженность CPU на 87% по сравнению с использованием обычных SSD.

Можно предположить сценарии использования SmartSSD для СУБД:

- теоретически возможно реализовать проверку прав на строки таблиц БД прямо на SmartSSD, за счет чего можно получить выигрыш в скорости обращения к данным БД;
- выполнение полнотекстовой индексации и алгоритмов поиска непосредственно на программируемой матрице SmartSSD без участия CPU и передачи данных между SSD и оперативной памятью;
- на SmartSSD можно хранить часто используемые витрины данных или материализованные представления для ускорения доступа к часто используемым данным пользователей СУБД;
- реализация алгоритмов сжатия/распаковки данных, шифрования данных непосредственно на программируемой матрице SmartSSD, что даст выигрыш в скорости обработки и позволит хранить защищенные данные на SmartSSD.

Безусловно, все эти возможности должны быть подтверждены экспериментально.

3. Проблемы и недостатки использования FPGA

На основании проведенного краткого обзора использования FPGA для ускорения операций СУБД, которая, казалось бы, бесспорна, можно выделить следующие проблемы и недостатки использования FPGA:

- 1) Первая проблема лежит не совсем в технической сфере. Это проблема зависимости от поставок оборудования FPGA, которое может

оказаться недоступным из-за сложной международной обстановки. Отечественных производителей FPGA не существует.

- 2) Настораживает тот момент, что большинство достигнутых результатов ускорения производительности обработки данных так и находятся на стадии экспериментов.
- 3) Компания IBM, которая приобрела Netezza в 2010 году, больше не выпускает машины БД с FPGA.
- 4) В последнее время появились альтернативные подходы к поисковой оптимизации запросов SQL на основе шаблонов (см., например, [19]) или ИИ (см., например, [20]). Анализ этих подходов будут посвящены дальнейшие исследования автора.
- 5) В последнее время большую, по сравнению с FPGA, популярность приобрели подходы, связанные с использованием GPU для распараллеливания и ускорения выполнения запросов к БД. Хотя FPGA считаются более энергоэффективными, GPU значительно более производительны (см., например, [21]). Кроме того, FPGA достаточно сложно программировать, и GPU/DPU стали альтернативной технологией ускорения обработки. Исследование возможностей ускорения СУБД за счет использования GPU и DPU будут также посвящены дальнейшие исследования автора.
- 6) Из-за повышения производительности компьютеров и увеличения пропускной способности каналов связи и передачи данных, в СУБД последнее время чаще используются технологии in-memory и NVM (Non-Volatile Memory (энергонезависимая память)). В связи с этим коммуникационные издержки на передачу данных, связанные с ускорителями FPGA, стали более критичными. Связь между CPU и FPGA гораздо менее быстрая по сравнению со схемой связи между ядрами CPU. Поэтому может не увеличиваться производительность в схеме совместного использования CPU и FPGA (см., например, [22]).
- 7) FPGA достаточно сложно программировать:
 - а. ограничены ресурсы FPGA на объемы программного кода;
 - б. сложно исправлять ошибки непосредственно на FPGA, т.к. требуется перепрограммирование, а количество циклов перезаписи программируемой матрицы ограничено;
 - с. генерация планов запросов, которые могут быть выполнены в системе с ускорением на FPGA, требуется компилятор запросов, согласованный по модели с FPGA;

- d. использование специализированных FPGA приводит к необходимости перенастройки устройств, когда, например, требуется другой набор функций, т.е. требуется частое перепрограммирование, что невозможно, либо замена устройств.

Возможные проблемы использования FPGA можно показать на примере критики проекта AQUA компании Amazon [15]:

- 1) Redshift ограничен по масштабам, потому что даже с типами узлов СУБД, поддерживающими AQUA, он не может эффективно распределять различные рабочие нагрузки по кластерам.
- 2) Как отмечает Firebolt, хотя Redshift может автоматически масштабироваться до 10 кластеров для поддержки параллелизма запросов, он может обрабатывать не более 50 запросов в очереди во всех кластерах.
- 3) Кроме того, поскольку Redshift блокируется на уровне таблицы, он лучше подходит для пакетного приема и ограничен в пропускной способности записи.

Заключение

В статье выполнен обзор способа ускорения выполнения операций обработки данных в СУБД с помощью FPGA. Выполнено рассмотрение возможностей FPGA, выполнен обзор архитектур использования FPGA, а также рассмотрены практические примеры использования FPGA.

Определены достоинства и недостатки применения FPGA. Отмечен ряд проблем и альтернативных технологий, которые существенно затормозили применение FPGA в промышленных СУБД и системах хранения данных.

В качестве вывода по применению FPGA для ускорения обработки данных в СУБД, можно утверждать, что подход позволяет ускорять операции в БД, связанные с обработкой запросов, сжатием и шифрованием данных, параллельной обработкой данных, в том числе записи на SSD. Однако, применение FPGA также усложняет систему в целом и увеличивает общую стоимость владения.

В дальнейших исследованиях планируется рассмотреть альтернативные технологии ускорения выполнения операций обработки данных в СУБД с помощью GPU, DPU, методов искусственного интеллекта.

Литература

1. Alcaín E., Fernández P.R., Nieto R., Montemayor A.S., Vilas J., Galiana-Bordera A., Martinez-

- Girones P.M., Prieto-de-la-Lastra, C., Rodriguez-Vila B., Bonet M., Rodriguez-Sanchez C. Hardware Architectures for Real-Time Medical Imaging // *Electronics*, 10(24): 3118, 2021. doi:10.3390/electronics10243118. ISSN 2079-9292.
2. Nagornov N.N., Lyakhov P.A., Valueva M.V., Bergerman M.V. RNS-Based FPGA Accelerators for High-Quality 3D Medical Image Wavelet Processing Using Scaled Filter Coefficients // *IEEE Access*, 10: 19215–19231, 2022. doi:10.1109/ACCESS.2022.3151361. ISSN 2169-3536.
3. Huffmire T., Brotherton B., Sherwood T., Kastner R., Levin T., Nguyen T.D., Irvine C. Managing Security in FPGA-Based Embedded Systems // *IEEE Design & Test of Computers*, 25 (6): 590–598, 2008. doi:10.1109/MDT.2008.166.
4. Babaei A., Schiele G., Zohner M. Reconfigurable Security Architecture (RESA) Based on PUF for FPGA-Based IoT Devices // *Sensors*, 22(15): 5577, 2022. doi:10.3390/s22155577. ISSN 1424-8220.
5. Simpson P.A. *FPGA Design, Best Practices for Team Based Reuse*, 2nd edition. Switzerland: Springer International Publishing AG, 2015. ISBN 978-3-319-17924-7.
6. Как работает FPGA? : RUVDs.com, habr. URL: <https://habr.com/ru/companies/ruvds/articles/736060/>. Дата публикации: 18.05.2023.
7. Fang J., Mulder Y.T.B., Hidders J. et al. In-memory database acceleration on FPGAs: a survey. *The VLDB Journal* 29, 33–59, 2020. <https://doi.org/10.1007/s00778-019-00581-w>.
8. Волков Д., Николаенко А. На пути к «железным» СУБД // *Открытые системы. СУБД*, №02, 2019 : OSP. URL: <https://www.osp.ru/os/2019/02/13054946>. Дата публикации: 19.05.2019.
9. Chen S., Chen Y., Wang Z., Qin W., Zhang J., Nand H., Zhang J., Li J., Zhang X., Liang X., Xu M. Efficient sequencing data compression and FPGA acceleration based on a two-step framework. *Front. Genet.*, 14: 1260531, 2023. doi: 10.3389/fgene.2023.1260531.
10. Lu A., Fang Z. SQL2FPGA: Automatic Acceleration of SQL Query Processing on Modern CPU-FPGA Platforms. 2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Marina Del Rey, CA, USA, 184-194, 2023. doi: 10.1109/FCCM57271.2023.00028.
11. Owaida M., Sidler D., Kara K., Alonso G. Centaur: A Framework for Hybrid CPU-FPGA Databases. 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Napa, CA, 211-218, 2017. doi: 10.1109/FCCM.2017.37.

12. *Wei X.* Sailfish: Exploring Heterogeneous Query Acceleration on Discrete CPU-FPGA Architecture. IEEE 39th International Conference on Data Engineering Workshops (ICDEW), Anaheim, CA, USA, 198-204, 2023. doi: 10.1109/ICDEW58674.2023.00036.
13. *Хасанов В.* Ускорение обработки данных 1С на FPGA-ускорителях Xilinx Alveo : rutube. URL: <https://rutube.ru/video/c246d709685406649e-5730c23a95e416/>. Дата публикации: 2021.
14. *Oikawa S.* Operating System Framework for Transparent Execution on a CPU and FPGA. 2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA), Kanazawa, Japan, 97-101, 2021. doi: 10.1109/SERA51205.2021.9509041.
15. Amazon throws SSDs, FPGAs, Nitro chips at Redshift with “AQUA” : TheStack. URL: <https://www.thestack.technology/aws-aqua-redshift-ga/>. Дата публикации: 15.04.2021.
16. Intelligent self-processing for the era of big data. SmartSSD : Samsung. URL: <https://semiconductor.samsung.com/emea/ssd/smart-ssd/> (дата обращения: 26.04.2025).
17. SmartSSD® Computational Storage Drive : AMD. URL: <https://www.amd.com/content/dam/xilinx/publications/product-briefs/xilinx-smartssd-computational-storage-drive-product-brief.pdf> (дата обращения: 26.04.2025).
18. Samsung представила SmartSSD второго поколения на базе процессоров AMD : overclockers. URL: <https://overclockers.ru/blog/TechRanch/show/71223/samsung-predstavila-smartssd-vtorogo-pokoleniya-na-baze-proces-sorov-amd>. Дата публикации: 22.07.2022.
19. *Feng J., Li Z., Chen Q.* Towards Exploratory Query Optimization for Template-Based SQL Workloads. 2024 IEEE 40th International Conference on Data Engineering (ICDE), Utrecht, Netherlands, 151-164, 2024. doi: 10.1109/ICDE60146.2024.00019.
20. *Chen T., Gao J., Tu Y., Xu M.* GLO: Towards Generalized Learned Query Optimization. 2024 IEEE 40th International Conference on Data Engineering (ICDE), Utrecht, Netherlands, 4843-4855, 2024. doi: 10.1109/ICDE60146.2024.00368.
21. *Fang J.* Database Acceleration on FPGAs. Dissertation (TU Delft), Delft University of Technology, 2019. <https://doi.org/10.4233/uuid:84dfc577-ca6f-43ea-9b24-4dc160c103f5>.
22. Практическое применение сервера с FPGA. Блог компании Selectel : habr. URL: <https://habr.com/ru/companies/selectel/articles/565190/>. Дата публикации: 10.07.2021.

Соловьев Александр Владимирович. Федеральный исследовательский центр «Информатика и управление» Российской академии наук» г. Москва, Россия. Главный научный сотрудник, доктор технических наук. Область научных интересов: системный анализ, системы управления базами данных, теория надежности, математическое моделирование, долговременное хранение электронных документов. E-mail: soloviev@isa.ru

Increasing the speed of DBMS operations using hardware FPGA accelerators

A.V. Solovyev

Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, Moscow, Russia

Abstract. The article provides an overview of a way to speed up data processing operations in a DBMS using an FPGA. An overview of FPGA capabilities, FPGA usage architectures is performed, and practical examples of FPGA usage are considered. The advantages and disadvantages of using FPGA are determined. A number of problems and alternative technologies were noted, which significantly slowed down the use of FPGAs in industrial DBMS and data storage systems. Using FPGA to speed up data processing in a DBMS, it can be argued that the approach allows you to speed up database operations related to query processing, data compression and encryption, and parallel data processing. However, the use of FPGAs also complicates the system as a whole, does not allow for flexible and quick reconfiguration of the system’s functionality, and increases the total cost of ownership. In further research, it is planned to consider alternative technologies for accelerating data processing operations.

Keywords: *database management systems, DBMS, FPGA, acceleration of data processing operations.*

DOI: 10.14357/20790279250208 **EDN:** XQCEYD

References

1. Alcaín E., Fernández P.R., Nieto R., Montemayor A.S., Vilas J., Galiana-Bordera A., Martinez-Girones P.M., Prieto-de-la-Lastra, C., Rodriguez-Vila B., Bonet M., Rodriguez-Sanchez C. Hardware Architectures for Real-Time Medical Imaging // *Electronics*, 10(24): 3118, 2021. doi:10.3390/electronics10243118. ISSN 2079-9292.
2. Nagornov N.N., Lyakhov P.A., Valueva M.V., Bergerman M.V. RNS-Based FPGA Accelerators for High-Quality 3D Medical Image Wavelet Processing Using Scaled Filter Coefficients // *IEEE Access*, 10: 19215–19231, 2022. doi:10.1109/ACCESS.2022.3151361. ISSN 2169-3536.
3. Huffmire T., Brotherton B., Sherwood T., Kastner R., Levin T., Nguyen T.D., Irvine C. Managing Security in FPGA-Based Embedded Systems // *IEEE Design & Test of Computers*, 25 (6): 590–598, 2008. doi:10.1109/MDT.2008.166.
4. Babaei A., Schiele G., Zohner M. Reconfigurable Security Architecture (RESA) Based on PUF for FPGA-Based IoT Devices // *Sensors*, 22(15): 5577, 2022. doi:10.3390/s22155577. ISSN 1424-8220.
5. Simpson P.A. *FPGA Design, Best Practices for Team Based Reuse*, 2nd edition. Switzerland: Springer International Publishing AG, 2015. ISBN 978-3-319-17924-7.
6. How does FPGA work? 2023. : RUVDs.com, habr. URL: <https://habr.com/ru/companies/ruvds/articles/736060/>. Published: 18.05.2023.
7. Fang J., Mulder Y.T.B., Hidders J. et al. 2020. In-memory database acceleration on FPGAs: a survey. *The VLDB Journal* 29, 33–59. <https://doi.org/10.1007/s00778-019-00581-w>.
8. Volkov D., Nikolayenko A. 2019. On the way to the “iron” DBMS. *Open systems. DBMS*, №02: OSP. URL: <https://www.osp.ru/os/2019/02/13054946>. Published: 19.05.2019.
9. Chen S., Chen Y., Wang Z., Qin W., Zhang J., Nand H., Zhang J., Li J., Zhang X., Liang X., Xu M. Efficient sequencing data compression and FPGA acceleration based on a two-step framework. *Front. Genet.*, 14: 1260531, 2023. doi: 10.3389/fgene.2023.1260531.
10. Lu A., Fang Z. SQL2FPGA: Automatic Acceleration of SQL Query Processing on Modern CPU-FPGA Platforms. 2023 IEEE 31st Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Marina Del Rey, CA, USA, 184-194, 2023. doi: 10.1109/FCCM57271.2023.00028.
11. Owaida M., Sidler D., Kara K., Alonso G. Centaur: A Framework for Hybrid CPU-FPGA Databases. 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Napa, CA, 211-218, 2017. doi: 10.1109/FCCM.2017.37.
12. Wei X. Sailfish: Exploring Heterogeneous Query Acceleration on Discrete CPU-FPGA Architecture. IEEE 39th International Conference on Data Engineering Workshops (ICDEW), Anaheim, CA, USA, 198-204, 2023. doi: 10.1109/ICDEW58674.2023.00036.
13. Khasanov V. Acceleration of 1C data processing on FPGA accelerators Xilinx Alveo : rutube. URL: <https://rutube.ru/video/c246d709685406649e-5730c23a95e416/>. Last access: 25.04.2025.

14. *Oikawa S.* 2021. Operating System Framework for Transparent Execution on a CPU and FPGA. 2021 IEEE/ACIS 19th International Conference on Software Engineering Research, Management and Applications (SERA), Kanazawa, Japan, 97-101. doi: 10.1109/SERA51205.2021.9509041.
15. Amazon throws SSDs, FPGAs, Nitro chips at Redshift with “AQUA” : TheStack. URL: <https://www.thestack.technology/aws-aqua-redshift-ga/>. Published: 15.04.2021.
16. Intelligent self-processing for the era of big data. SmartSSD : Samsung. URL: <https://semiconductor.samsung.com/emea/ssd/smart-ssd/>. Last access: 26.04.2025.
17. SmartSSD® Computational Storage Drive : AMD. URL: <https://www.amd.com/content/dam/xilinx/publications/product-briefs/xilinx-smartssd-computational-storage-drive-product-brief.pdf>. Last access: 26.04.2025.
18. Samsung has introduced a second-generation Smart SSD based on AMD processors : overclockers. URL: <https://overclockers.ru/blog/TechRanch/show/71223/samsung-predstavila-smartssd-vtorogo-pokoleniya-na-baze-processorov-amd>. Published: 22.07.2022.
19. *Feng J., Li Z., Chen Q.* Towards Exploratory Query Optimization for Template-Based SQL Workloads. 2024 IEEE 40th International Conference on Data Engineering (ICDE), Utrecht, Netherlands, 151-164, 2024. doi: 10.1109/ICDE60146.2024.00019.
20. *Chen T., Gao J., Tu Y., Xu M.* GLO: Towards Generalized Learned Query Optimization. 2024 IEEE 40th International Conference on Data Engineering (ICDE), Utrecht, Netherlands, 4843-4855, 2024. doi: 10.1109/ICDE60146.2024.00368.
21. *Fang J.* Database Acceleration on FPGAs. Dissertation (TU Delft), Delft University of Technology, 2019. <https://doi.org/10.4233/uuid:84dfc577-ca6f-43ea-9b24-4dc160c103f5>.
22. Practical application of an FPGA server. Selectel company blog : habr. URL: <https://habr.com/ru/companies/selectel/articles/565190/>. Published: 10.07.2021.

Aleksandr V. Solovyev. Doctor of Technical Sciences. Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, Moscow, Russia. Research interests: system analysis, database management systems, reliability theory, mathematical modeling. E-mail: soloviev@isa.ru