

FPGAs in DBMS

1st Felix Grenzing
Universität Hamburg
Hamburg, Deutschland
felix.grenzing@studium.uni-hamburg.de

Zusammenfassung—Abstract here.

Index Terms—FPGAs, DBMS, Hardware Acceleration

I. EINFÜHRUNG

II. BEGRIFFE

In [1] und [2] werden verschiedene Begriffe verwendet, die Erklärung bedürfen.

A. FPGAs

B. Einsatzmöglichkeiten

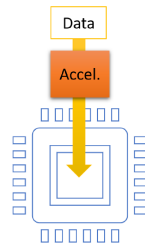


Abbildung 1. In Datapath Architektur

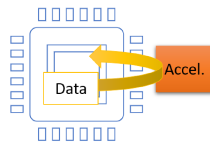


Abbildung 2. On The Side Architektur

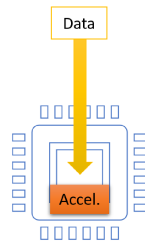


Abbildung 3. Koprozessor Architektur

C. Pipelining und Datenparallelität

Zwei wichtige Arten der Parallelität sind die Pipelineparallelität und die Datenparallelität. Pipelineparallelität beschreibt die Aufteilung der Befehlsabarbeitung in mehrere Stufen, die parallel arbeiten. Jede Stufe bearbeitet dabei Teil des Befehls

und gibt die Ergebnisse an die nächste Stufe weiter. Ein Beispiel für eine Pipeline ist in Abbildung 4 zu sehen. Die Befehle werden in vier Stufen aufgeteilt. Bearbeitet die Fetch Stufe den ersten Befehl. Sobald dies geschehen ist, wird der Befehl an die nächste Stufe weitergegeben. Die Fetch Stufe kann nun den nächsten Befehl bearbeiten. Die Befehle werden so stufenweise durch die Pipeline geschoben.

Datenparallelität beschreibt die Aufteilung der Daten in mehrere Teile, die mit mehreren Bearbeitungseinheiten parallel verarbeitet werden. Beide Ansätze können die Performance von Algorithmen verbessern, und können auch kombiniert werden um die Vorteile beider Ansätze zu nutzen. Um beide Ansätze zu kombinieren, könnten mehrere Pipelines entworfen werden, die jeweils verschiedene Datenbearbeiten.

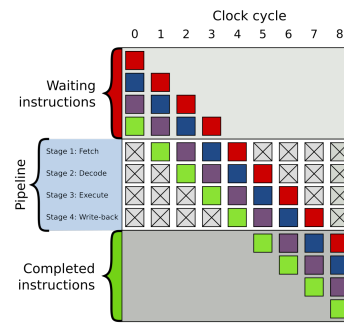


Abbildung 4. Prozessor Pipeline

D. Partial Reconfiguration

Partial Reconfiguration (PR) ist ein Konzept, welches aktuelle FPGAs unterstützen. PR bietet die Möglichkeit, Teile des FPGAs zur Laufzeit zu verändern, ohne das gesamte FPGA neu zu konfigurieren oder das FPGA offline nehmen zu müssen. Dies ermöglicht es, verschiedene Funktionen auf einem FPGA zu implementieren und bei Bedarf zu wechseln. Umkonfiguration während der Laufzeit bietet offensichtlich Potenzial für Beschleunigung, da mehr Algorithmen implementiert werden können, als gleichzeitig auf dem FPGA Platz haben. Eine Problematik ist jedoch, dass die Umkonfiguration Zeit in Größenordnungen von Millisekunden benötigt.

Das Projekt DoppioDB [1] nutzt PR, um verschiedene Algorithmen auf einem FPGA zu implementieren und bei Bedarf zu wechseln.

E. BitWeaving

TODO: Faktcheck BitWeaving ist ein Algorithmus, der in [1] vorgestellt wird. Der Algorithmus stellt eine Methode dar, Spaltenscanoperationen durchzuführen, indem die Daten mehrerer Zeilen als Bitvektor codiert in einem Prozessorwort gespeichert und verarbeitet werden. Zwei Varianten des Algorithmus werden vorgestellt, BitWeaving H und BitWeaving V, die sich in der Art und Weise unterscheiden, wie die Daten in den Prozessorworten gespeichert werden. BitWeaving H speichert die Daten zeilenweise, BitWeaving V spaltenweise.

III. EINORDNUNG DER SITUATION

A. Pessimismus

B. Optimismus

Andere Architekturen.

IV. ANDERES PAPER

Im Kontext veränderter Architekturen ordnet sich auch [2] ein. Dort wurde auf einem FPGA ein Column Store implementiert, der den BitWeaving H Algorithmus nutzt. Es wurden verschiedenen Hardwareansätze verglichen, um die beste Performance zu erzielen.

A. Architektur

Die Forschung von [2] baut auf der Zynq Ultrascale+ Architektur von Xilinx auf. Die Plattform ist aus zweiteilig aus Steuersystem mit 4 ARM Cortex-A53 Kernen und 4GB DDR4 Speicher und Logikbereich mit FPGA und 500MB DDR4 Speicher aufgebaut [2].

B. Grundlegendes Vorgehen

Pipeline
Processing Elements
Combiner

C. Verschiedene Ansätze

Mehrere parallele Pipelines -> Datenparallelität
Limitation des DDR4-Controllers
Optimale Nutzung von Combiner
Hybridansatz

D. Ergebnisse

V. ANDERE ANSÄTZE

A. IBEX

B. REGEXP_LIKE

VI. THE ROAD THAT LIES AHEAD

LITERATUR

- [1] Y. Li and J. M. Patel, "BitWeaving: fast scans for main memory data processing," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '13. New York, NY, USA: Association for Computing Machinery, Jun. 2013, pp. 289–300. [Online]. Available: <https://doi.org/10.1145/2463676.2465322>
- [2] N. J. Lisa, A. Ungethüm, D. Habich, W. Lehner, T. D. Nguyen, and A. Kumar, "Column Scan Acceleration in Hybrid CPU-FPGA Systems," in *ADMS@ VLDB*, 2018, pp. 22–33.