The glas half full – Präsentation

Felix Grenzing

4. Januar 2025

1 Einführung

Motivation: Traditionell skeptische Haltung gegenüber spezialisierter Hardware in Datenbanken. Herausforderungen: Kosten, Komplexität und eingeschränkte Anwendbarkeit. Neue Entwicklungen machen Hardware-Beschleunigung wieder interessant.

Kontext: Stagnation der CPU-Leistung und steigende Netzwerkgeschwindigkeiten. Heterogene Hardwarearchitekturen (z. B. FPGAs, TPUs) etablieren sich in Rechenzentren.

Moore's Law und Stagnation der CPU-Leistung.

2 Herausforderungen früherer Ansätze

Limitierungen: Hohe Latenz durch Kommunikation über PCIe. Um PCIe zu vermeiden, wird der gesamte Algorithmus auf den Beschleuniger ausgelagert, wobei Geschwindigkeitsvorteile verloren gehen, da nur Teile des Algorithmus gut zu dem Beschleuniger passen. Außerdem ist komplizierte Hardware nötig, um große if-else-Strukturen abzubilden.

Zahlen nennen

Einschränkungen der FPGA-Ressourcen (Speicher, Logikgatter). Komplexität bei der Optimierung für Datenbank-Workloads.

Vergleich zu CPUs: CPUs oft schneller bei iterativen oder verzweigten Algorithmen. Unsicherheiten in der Query-Optimierung durch Hardware-Limitierungen.

Spaltenbasierte Datenbanken können vielkern CPUs gut verwenden und reduzieren daher die Notwendigkeit von Beschleunigern.

FPGA bisschen erklären und Daten nennen

Warum?

3 Gründe für Optimismus

Technologische Entwicklungen: Verfügbarkeit von programmierbaren Co-Prozessoren (z. B. Intel Xeon+FPGA). In-Datenpfad-Beschleunigung zur Reduktion der Datenbewegung.

Neue Workloads: Datenbanken profitieren von hardwarebeschleunigter Verarbeitung für maschinelles Lernen und analytische Workloads, da dies häufig compute-bound sind, bspw. für low-latency Inferenz. Beispiel: SQL-Server mit Machine-Learning-Plugins.

Beispiele nennen

Ansätze, welche unabhängig von den Daten sind, helfen dem Optimizer, die beste Ausführungsstrategie zu finden. (LIKE Operator)

4 Ansätze

- Ibex: FPGA-basierte Datenbankbeschleunigung. (partielle Verarbeitung von Anfragen auf dem FPGA)
- DoppioDB: Defintion von ML-Operatoren, welche auf Slots des FPGAs ausgeführt werden.
- Corner cases müssen in Software gelöst werden

Erklärung

5 Zentrale Konzepte

Heterogene Hardwarearchitekturen: Drei Kategorien: On-the-Side (z. B. GPUs), In-Data-Path (z. B. Smart NICs) und Co-Processor (z. B. Oracle DAX).

Bilder

- On the Side (traditionell): CPU besitzt die Daten und sendet sie an den Beschleuniger über bspw. PCIe. (Datenfilterung, Dekompression)
- in Data-Path: Beschleuniger ist Teil des Datenpfads und verarbeitete die Daten live. (vor allem Reduktion der Datenmenge)
- Co-Processor: Beschleuniger ist Teil des Prozessors und kann auf den Arbeitsspeicher zugreifen.

FPGAs als Beispiel: Flexible Konfiguration und Kombination von Parallelitätsansätzen (Pipeline, Data-Parallelismus). Teilweise Neukonfiguration ermöglicht dynamische Anpassung. (Partial Reconfiguration; benötigt allerdings einige Millisekunden)

Verweis auf Bitweaving

Erklärung

6 Offene Fragen

Integration in Datenbanken: Wie kann Query-Planung und Ressourcenmanagement angepasst werden? Zusammenarbeit mit anderen Forschungsbereichen notwendig (z. B. Hardwaredesign).

Warum?

7 Fazit

Neue Perspektive: Fortschritte in Hardware und veränderte Workloads eröffnen Chancen für spezialisierte Hardware. Zukunftsvision: Kooperation zwischen Hardware- und Datenbankforschung notwendig, um das Potenzial voll auszuschöpfen.