Research 1: Emerging Hardware

1. **Revisiting Co-Processing for Hash Joins on the Coupled CPU-GPU Architecture**

2. **Hardware-Oblivious Parallelism for In-Memory Column-Stores**

5. **The Yin and Yang of Processing Data Warehousing Queries on GPU Devices**

はじめに

- 感想
  - 3本とも微妙でした
  - 3本とも評価頑張りました系論文
    - 1本目はちょっと違うかも？
- GPU 等に興味ない人
  - 聞かなくても良いです
- GPU 等に興味ある人
  - 自分で読んでください

Revisiting Co-Processing for Hash Joins on the Coupled CPU-GPU Architecture

Jiong He†, Mian Lu‡, Bingsheng He†

†Nanyang Technological University
‡A*STAR Institute of HPC, Singapore

概要

- Coupled CPU-GPU architecture
  - 一つのチップにCPUとGPUを搭載
  - AMD APU (Accelerated Processing Unit)

- 貢献
  - APU を用いた効率的な Hash join の提案
  - コストモデルの構築
  - 実験により従来手法より高速であることを示した
  - この結果をもとに,
    APUを用いた効率的な問合せ処理に関して考察

Coupled CPU-GPU Architecture

- 特徴
  - データ転送が不要
  - キャッシュを共有している
  - GPU の性能は，個別のGPUと比べると低い

Table 1: Configuration of AMD Fusion A8-3870K. The last column shows the configuration of AMD Radeon HD 7970 for reference.

	CPU (APU)	GPU (APU)	GPU (Discrete)
# Cores	4	400	2048
Core frequency(GHz)	3.0	0.6	0.9
Zero copy buffer (MB)	512 (shared)		–
Local memory size(KB)	32	32	32
Cache size(MB)	4 (shared)		–

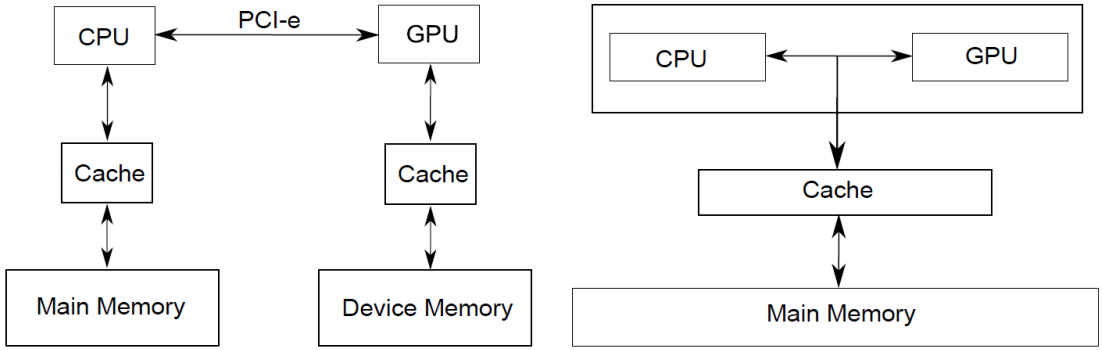

Figure 1: An overview of discrete and coupled CPU-GPU architectures

Fine-Grained Co-Processing

- Coarse-grained co-processing（従来）
  - Hash join 全体を CPU が行う,
  - Hash join 全体を GPU が行う, or
  - データをある割合で分割して CPU と GPU で同時処理

- Fine-grained co-processing
  - Hash join を細かいステップに分割
  - 各ステップで,
    データをある割合で分割して CPU と GPU で同時処理
    - 各ステップごとに割合を変化させる

Hardware-Oblivious Parallelism for In-Memory Column-Stores

Max Heimel†, Michael Saecker‡, Holger Pirk*,
Stefan Manegold*, Volker Markl†

†Technische Universität Berlin
‡ParStream GmbH
*CWI Amsterdam

Hardware-Oblivious

- Hardware-conscious
  - 各ハードウェアに最適化したコードを人手で書く
- Hardware-oblivious
  - 抽象化したモデルに対してのみコードを書く
  - ハードウェアごとの最適化などは
    コンパイラやドライバにまかせる

この論文を一文で表すと…

OpenCL を使って
関係演算子を実装して
MonetDB に組み込んで
評価しました

The Yin and Yang of Processing Data Warehousing Queries on GPU Devices

Yuan Yuan, Rubao Lee, Xiaodong Zhang

The Ohio State Unviersity

概要

- データベース処理の GPU による高速化
  - 様々な研究が行われてきている
  - 実際のシステムで GPU が使われている事例は殆ど無い
- 様々な側面から GPU による
  データウェアハウスクエリの性能を分析
  1. 処理のどの部分に時間がかかっているか？
  2. 既存の最適化手法はどの程度有効か？
  3. GPU はどのような状況において有効か？
  4. GPU 自体の違いが性能にどのような影響を与えるか？
  5. 今後，GPU の性能が向上したときにクエリ処理性能がどのように変化するか？

概要（続き）

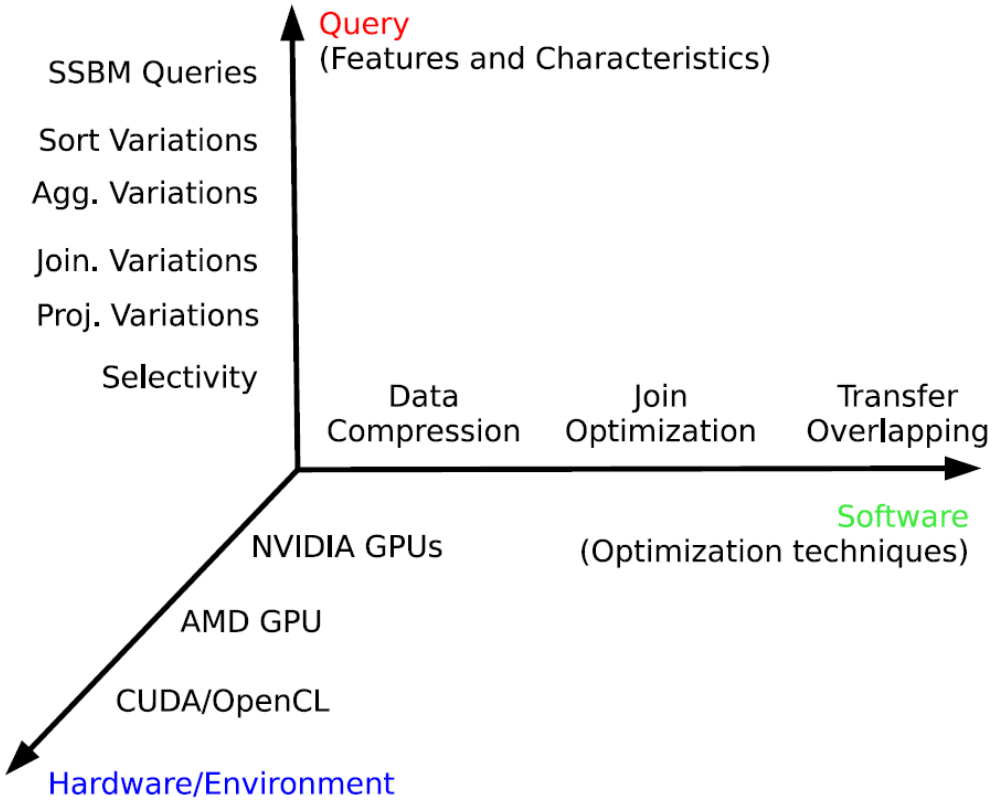

Figure 1: Research Overview: A 3-Dimension Study of Processing Warehousing Queries on GPUs.

分析結果のまとめ

- GPU が CPU より著しく高速 (4.5—6x) になる場合
  - データが pinned ホストメモリ上にあり,
    selection の時間が支配的でない
- GPU による高速化があまり顕著でない (2x) 場合
  - Selection の時間が支配的
  - メモリへのランダムアクセスが多い
  - データが pinned ホストメモリ上にない
- GPU の計算性能の向上は,
  データウェアハウス処理の性能向上には
  つながらない

GPUを変えたときの性能の変化

- GTX 480
  - 1345 GFLOPS
  - 177.4 GB/s
- GTX 580
  - 1581 GFLOPS
  - 192.4 GB/s
- GTX 680
  - 3090 GFLOPS
  - 192.3 GB/s

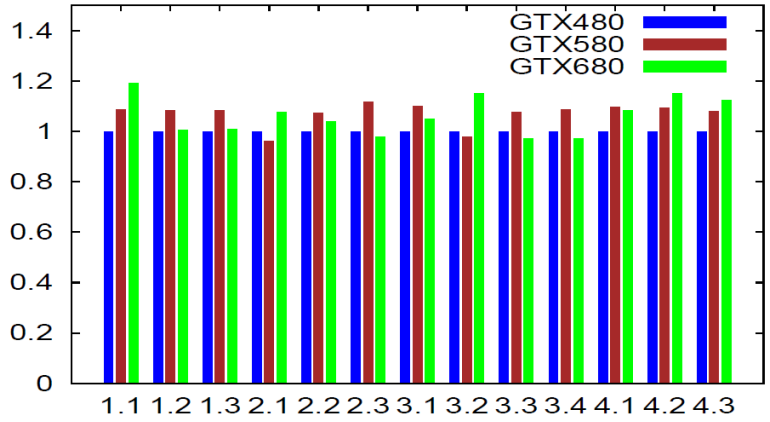

Figure 15: Normalized kernel execution time on GTX 480

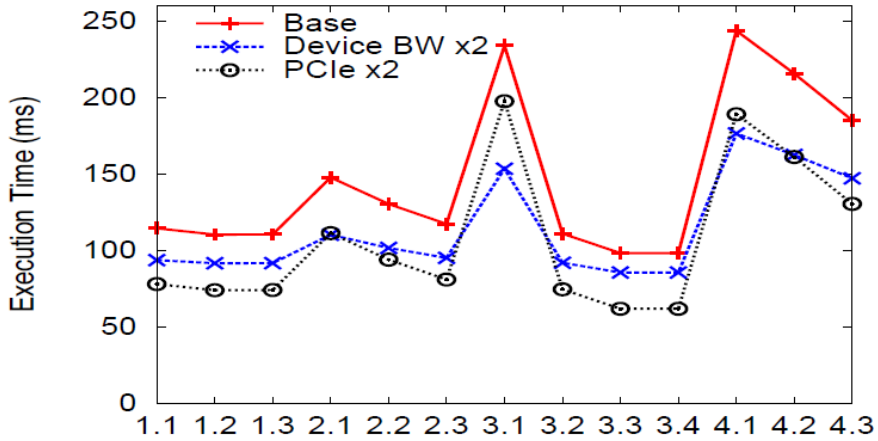

Figure 16: Estimated SSBM performance with different GPU hardware configurations