CUDA-RA: Uma ferramenta de interpretação de álgebra relacional e estrutura de dados para GPU

Jéssica A. Raposo Seibert[1], **Victor Pedro Corrêa**[1], **Luis Fernando Orleans**[1], **Marcelo Zamith**[1]

[1]Departamento de Ciência da Computação – Universidade Federal Rural do Rio de Janeiro (UFRRJ) – Nova Iguaçu – RJ – Brasil

{jessicaa.raposo,victor.pedrocorreia,zamith}@gmail.com

lforleans@ufrrj.br

Abstract. *Relational databases are an important field of application for organization and analysis of large amounts of data generated by various technologies. At the same time, computer architectures are beginning to move towards hierarchical and parallel architectures emplyoing memory systems optimized with respect to flow, lightweight multi-threading, and organizations of cores in the Single-Instruction Multiple-Data (SIMD) format. This paper explores the mapping of primitive operations of structural relational algebra and the usage of GPUs,in a way assist in the databases discipline , to interpret queries and show the best way to gain performance in data exploration.*

Resumo. *Banco de dados relacionais são um importante campo de aplicação para organização e análise de grandes quantidades de dados gerados por diversas tecnologias. Ao mesmo tempo, arquiteturas de computador estão começando a se deslocar em direção a arquiteturas hierárquicas e paralelas empregando sistemas de memória otimizadas com relação a vazão, multithreading leve, e organizações de núcleos no formato instrução-única múltiplos-dados (SIMD). Esse artigo explora o mapeamento de operações primitivas da álgebra relacional estrutural e o uso de GPUs, de forma auxiliar no ensino da disciplina de banco de dado, interpretar consultas e demonstrar a melhor forma de ganho de desempenho na exploração de dados.*

1. Introdução

A área de computação é constituída por tecnologias novas, em fase de grande expansão, contínuas modificações e estágios de maturidade heterogêneos. Essa situação gera um desafio aos professores; suas técnicas de ensino ainda não estão maduras e, portanto, sofrem adaptações frequentes [Delgado et al. 2004]. Muito se deve ao fato de o aluno não conseguir visualizar a aplicação de determinados conceitos em situações da vida real e ao fato do aluno possuir grande dificuldade em interpretar os exercícios propostos. Percebeu-se que o método tradicional de ensino não deixa claro para os alunos a importância de certos conteúdos para sua formação [Freitas et al. 2014]. O estudo e a compreensão do Modelo Relacional (MR) e da Álgebra Relacional (AR) são fundamentais para estudantes de cursos de Banco de Dados. Para facilitar seu aprendizado, é conveniente ter um ambiente para testes de consultas. Por se tratar de uma linguagem formal, poucas ferramentas interpretam consultas em álgebra relacional, dificultando a aprendizagem.

DOI: 10.5753/cbie.sbie.2016.1290

Um Banco de Dados Relacional (BDR) é representado por uma coleção de tabelas, onde cada tupla representa uma relação entre um conjunto de valores [Silberschatz et al. 2006]. Uma de suas vantagens é a facilidade de acesso aos dados, possibilitando que o usuário utilize uma grande variedade de abordagens no tratamento das informações. A AR consiste em um conjunto de operações para manipulação dos dados em um BDR, onde o procedimento para encontrar recuperar os dados desejados é descrito de forma procedural. O usuário fornece as instruções que representam operações na base de dados e o resultado obtido é uma nova relação.

Uma aplicação da AR é especificada como um grafo de fluxo de dados de operações, tornando capaz o mapeamento natural para uma variedade de modelos de execução paralelos, por exemplo, mapeando os operadores para núcleos *Multi-BSP* e relações para estruturas de dados[Diamos et al. 2012]. A Tabela 1 lista o conjunto das operações mais comuns da AR, onde os quatro primeiros elementos representam operações desenvolvidas especificamente para BDR e as demais compõe as operações de teoria de conjuntos da matemática.

Operações realizadas em AR capturam semânticas de alto nível de uma aplicação em termos de uma série de operações, como junção ou projeção, em relações. Esses sistemas têm sido desenvolvidos com sucesso em processadores *multi-cores* e *clusters* computacionais.

Uma outra tecnologia que vem ganhando bastante destaque no campo de processamento paralelo é a GPU (*Graphics Processing Unit*), seu baixo custo e seu alto poder de processamento massivamente paralelo tornam a GPU acessível e viável de ser utilizada em outros problemas além da Computação Gráfica [Zamith et al. 2007]. Desta forma, há a necessidade de desenvolver e adaptar algoritmos para a GPU de forma a aproveitar todo o poder computacional destes dispositivos.

Diferentemente do procesamento em CPU, a GPU foi desenhada com o objetivo de ser capaz de processar grandes quantidades de dados paralelamente. Enquanto as CPUs contam com alguns cores físicos capazes de instanciar dezenas de *threads*, uma única GPU possui mais de três mil cores físicos que são capazes de instanciar milhares de *threads* [Clua and Zamith 2015], sendo uma única GPU um verdadeiro *cluster* computacional. Este artigo usa NVIDIA CUDA como plataforma de desenvolvimento para GPU, mas os conceitos aqui propostos podem ser aplicados em outras plataformas de desenvolvimento para GPU.

O presente trabalho propõe um interpretador de expressões de álgebra relacional para auxiliar na aprendizagem das operações fundamentais, além de contribuir para o desenvolvimento de algoritmos hierárquicos e de dados paralelos de futuros processadores modernos de propósito geral para alcançar boa escalabilidade e performances quase ótimas. Ao final, espera-se um *software* funcional e de fácil utilização para auxiliar no ensino de Banco de Dados.

2. Algoritmo proposto

O algoritmo proposto para a implementação da ferramenta CUDA-RA será o desenvolvimento dos operadores de AR de forma sequencial, em CPU, da forma convencional para os usuários. Logo após será desenvolvida e apresentada a opção de execução em GPU, possibilitando a análise comparativa de desempenho em ambos os cenários.

Tabela 1. Operadores da álgebra relacional

Operador	Descrição	Simbolo	Exemplo
Projeção	Um operador unário que consome um relação de entrada para produzir uma nova relação de saída.	π	π_{coluna}(tabela)
Seleção	Usada para selecionar um subconjunto de tuplas de uma relação que satisfaça uma condição de seleção	σ	$\sigma_{coluna=1}$(tabela)
Junção	Uma relação binária que intercepta sobre o atributo chave e cruza produtos de atributos chaves.	$\bowtie$	tabela1 $\bowtie$ tabela2
Renomeação	Permite obter uma nova relação com o nome da relação e/ou o nome dos atributos renomeados.	ρ	$\rho_{novatabela}$(tabela1)
Produto	Operador binário que combina o espaço de atributos de duas relações para produzir uma nova relação.	$\times$	tabela1 $\times$ tabela2
Interseção	Operação que consome duas relações para produzir uma nova relação consistindo de tuplas com chaves que estão presentes em ambos as relações de entrada.	$\cap$	tabela1 $\cap$ tabela2
União	Operação que consome duas relações para produzir uma nova relação consistindo de tuplas com chaves que estão presentes em no mínimo uma das relações de entrada.	$\cup$	tabela1 $\cup$ tabela2
Diferença	Operação que consome duas relações para produzir uma nova relação de tuplas com chaves que existem em uma das relações de entrada e não existe na outra relação de entrada.	-	tabela1 - tabela

O interpretador deverá ser de fácil manuseio para os estudantes, contará com os operadores descritos na Tabela 1, será exclusivamente para o ensino de expressões da AR e não possui o intuito de funcionar conforme um SGBDR (Sistema Gerenciador de Banco de Dados Relacional). O aluno terá a liberdade de criar combinações e pesquisas de forma simples, com a tecnologia que desejar e assim aprender o funcionamento de cada operação.

2.1. Metodologia de desenvolvimento

Para realizar as consultas, será feita a leitura de arquivo, no formato de: .txt e .csv. O arquivo será lido e organizado de forma em que as relações serão armazenadas em vetores de tuplas, ordenados utilizando uma função de comparação que opera nos atributos de cada tupla e define uma ordenação em cima de todas as tuplas na relação.

Esta sendo implementado, primeiramente, a forma sequencial para ser executado em uma CPU *single-core*, foi desenvolvido na linguagem C++, utiliza apenas as bibliotecas padrão da linguagem, tais como: *vector, iostream, string, sstream e fstream*. A implementação do programa foi dividida em módulos com o objetivo de seguir o princípio de responsabilidade única, além de possibilitar a troca dos algoritmos de sequencial para multi-core em GPU sem afetar todo o resto do código.

Na GPU, a implementação foi estruturada como uma sequência de estágios de forma que, em alguns estágios, a funcionalidade é compartilhada entre diferentes operadores. A Figura1 ilustra como será o fluxo de funcionamento da ferramenta com o usuário.

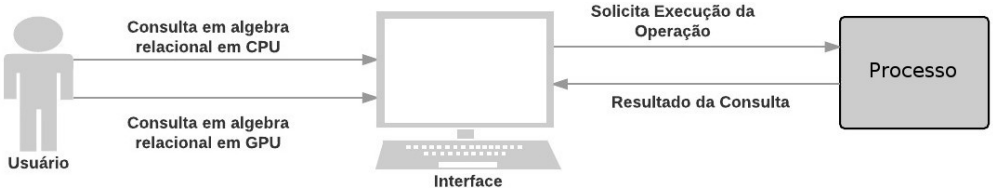

Figura 1. Modelo de execução da ferramenta de Algebra Relacional

3. Primeiros Resultados

A Base de dados escolhida para teste é encontrada em [Data.rio 2013] e possui as transações imobiliárias de arrecadação de impostos por bairros. O arquivo é de fevereiro de 2014 referente as arrecadações de 2013, contém 7 atributos e 457 instâncias.

O primeiro operador implementado, foi o *Project*. A operação de projeção, representada pelo operador π , é unária e filtra os atributos desejados na relação resultante, é simples comparado aos demais algoritmos. Este operador faz uma passagem única pelas tuplas de uma relação armazenando em memória as colunas de cada tupla que serão utilizadas para a criação da nova relação, e descartando as colunas que não foram explicitamente determinadas pelo usuário. A paralelização deste operador se dá pela divisão das tuplas pelas *threads* da GPU.

O resultado obtido com do operador foi a leitura, execução e exibição dos dados do arquivo. De forma a avaliar o desempenho da leitura, ela foi dividida em certas quantidades de dados e comparado o tempo de execução conforme se observa na Tabela 2.

Tabela 2. Tempo de execução em GPU e CPU em segundos

n de Instâncias	CPU	GPU
100	0.0021122	0.000012
200	0.002162	0.000011
300	0.0021369	0.000011
400	0.002115	0.000021
457	0.0021079	0.000010

Podemos observar que, mesmo em uma pequena base de dados, foi possível obter

uma vantagem expressiva ao utilizar a GPU para executar os algoritmos do operador relacional.

4. Considerações finais e trabalhos futuros

O estudo e a compreensão do modelo relacional e da álgebra relacional são fundamentais para o aprendizado de Banco de Dados, sendo assim o presente trabalho apresentou o desenvolvimento de uma ferramenta capaz de interpretar expressões escritas nela. A ferramenta está sendo implementada de forma a permitir que novos operadores sejam criados. Um estudo de caso com o desenvolvimento de um operador utilizando CUDA foi apresentado. Espera-se com isso o incentivo para a propagação do uso da álgebra relacional como ferramenta de ensino nos cursos, assim como a extensão do uso da ferramenta para estudar o desenvolvimento de operadores relacionais que alcançam boa escalabilidade na arquitetura das GPUs.

A utilização de arquitetura paralela da GPU possibilitou a comparação do desempenho, do ponto de vista de tempo de execução, demonstrou que a mesma traz benefícios de ganhos de desempenho e performance. O avanço destas tecnologias envolvendo GPUs é crescente, e trabalhos como este incentivam o aprendizado destas tecnologias e suas aplicações em áreas que podem se beneficiar destes recursos.

O projeto está em crescimento e como trabalhos futuros teremos a implementação dos demais operadores já citados acima, formando assim uma ferramenta completa para interpretação de expressões de álgebra relacional para o aprendizado.

Referências

Clua, E. W. G. and Zamith, M. P. (2015). Programming in cuda for kepler and maxwell architecture. *Revista de Informática Teórica e Aplicada*, 22(2):233–257.

Data.rio (2013). Total de transações com guia não isenta por bairro. Disponível em: `http://data.rio/dataset/itbi-total-de-transacoes-por-periodo-tipo-e-bairro/resource/84606eee-3737-40f0-997d-c3c531a8e009`, Acesso em: 30 maio. 2016.

Delgado, C., Xexeo, J. A. M., SOUZA, I. F., Campos, M., and Rapkiewicz, C. E. (2004). Uma abordagem pedagógica para a iniciação ao estudo de algoritmos. In *XII Workshop de Educação em Computação*.

Diamos, G. F., Wu, H., Lele, A., and Wang, J. (2012). Efficient relational algebra algorithms and data structures for gpu.

Freitas, M. F., Mota, S. D. S., Soares, L. S., and Reis, R. C. D. (2014). Portec: uma ferramenta para auxiliar na abstração dos conceitos de estrutura de dados. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 25, page 872.

Silberschatz, A., Korth, H. F., Sudarshan, S., and Vieira, D. (2006). *Sistema de banco de dados*. Elsevier, 5 edition.

Zamith, M., Clua, E., Pagliosa, P., Conci, A., MONTENE-GRO, A., and Valente, L. (2007). The gpu used as a math co-processor in real time applications. In *Proceedings of the VI Brazilian Symposium on Computer Games and Digital Entertainment*, pages 37–43. Citeseer.