

# 그래픽 프로세서 환경에서의 병렬 FP-growth 알고리즘 구현

\*정의석<sup>1</sup>, 방관후<sup>1</sup>, 윤성로<sup>2</sup>, 정의영<sup>1</sup>

연세대학교 전기전자공학과<sup>1</sup>, 고려대학교 전기전자전파공학과<sup>2</sup>

e-mail : {krieitiv, khbang}@dtl.yonsei.ac.kr, sryoon@korea.ac.kr, eychung@yonsei.ac.kr

## An Efficient Implementation of the FP-growth Algorithm using Graphics Processing Units

\*Eesuk Jung<sup>1</sup>, Kwanhu Bang<sup>1</sup>, Sungroh Yoon<sup>2</sup>, Eui-Young Chung<sup>1</sup>  
School of Electrical and Electronic Engineering, Yonsei University<sup>1</sup>  
School of Electrical Engineering, Korea University<sup>2</sup>

### Abstract

Graphics processing unit (GPU) was originally designed as a co-processor of CPU for graphics applications such as games and image rendering. Recently, the use of GPU in other non-graphics applications is becoming increasingly popular. In this paper, we present an efficient GPU-based implementation of the FP-growth algorithm used in frequent itemset mining. Our implementation exploits SIMD multi-processors of GPU to maximize parallelism. Our experimental results show that the proposed method can run up to 14% faster than the CPU-based implementation.

### I. 서론

최근 전통적으로 컴퓨터 그래픽을 위한 연산만을 수행하던 GPU를 사용하여 CPU가 담당하던 응용 프로그램들의 계산까지 수행하는 범용 GPU 컴퓨팅 기술이 대두되었다. 그러나 GPU 기반의 컴퓨팅을 위해서는 지금까지와는 다른 병렬 프로그래밍 패러다임을 필요로 하게 되었다. 본 논문에서는 NVIDIA의 CUDA 환경

에서 FP-growth 알고리즘의 효율적인 C 언어 구현을 통하여, 기존의 CPU 기반의 프로그램 대비 성능의 향상을 보였다.

### II. 본론

#### 2.1 NVIDIA CUDA 환경

과거에는 게임과 같은 그래픽 응용 프로그램을 위한 CPU의 보조 연산장치로 특화되었던 GPU가 NVIDIA에서 선보인 CUDA 환경을 통하여 범용 병렬 컴퓨팅 구조로 발전되었다. 기존의 멀티코어 CPU와는 달리 다수의 SIMD 기반의 프로세서를 사용하는 GPU는 대단히 많은 쓰레드의 관리를 하드웨어에서 자동적으로 수행하며, GPU가 가진 높은 대역폭의 메모리를 활용할 수 있다. 이러한 GPU의 특징을 이용하여 성능을 극대화하기 위해서는 병렬성을 가진 데이터 집합의 크기가 크고, 각 데이터 요소 간의 의존성이 최소화되어야 한다[1].

#### 2.2 FP-growth 알고리즘

연관 분석의 빈발 항목집합 마이닝 문제는 주어진 트랜잭션 속에서 minimum support 임계값을 만족하는 모든 항목집합을 찾아 높은 신뢰도의 규칙을 찾아내는 것이다. 이를 가장 빠르게 풀 수 있는 것으로 알려진 FP-growth 알고리즘은 FP-tree라 불리는 트랜잭션의

사상을 분할-정복 전략을 적용하여 조건 테이블과 조건 트리를 재귀적으로 구성하여 특정 접미사로 끝나는 모든 빈발 항목집합을 찾는다[2].

### III. 구현 및 결과

CUDA API를 이용하여 기존의 FP-growth 알고리즘을 데이터 병렬성을 극대화하는 메모리 구조와 스레드 병렬화에 중점을 두어 구현하였다. 우선 GPU 하드웨어의 특징적인 계층적 구조와 대역폭의 한계를 극복하기 위한 데이터 구조를 설계하였다. FP-tree의 불규칙한 데이터 구조를 2차원의 규칙적인 구조로 변환하여 멀티프로세서의 각 스레드 단위 읽기에 적합한 메모리 액세스 패턴을 구성하였으며, 제한된 메모리 대역폭의 병목을 피하기 위하여, 계층적 메모리 간의 데이터 이동을 최소화하였다. 이에 더하여, 모든 프로세서 코어들에 의해 공유되는 읽기 전용의 텍스처 메모리를 사용하여 속도를 향상시켜주는 텍스처 캐시를 각각의 멀티프로세서들이 텍스처 유닛을 통해서 접근할 수 있도록 하여 32비트 float형의 배열의 효율적 읽기 성능을 구현하였다.

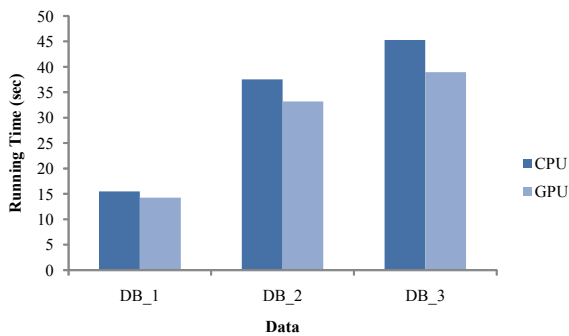


그림 1. 다양한 데이터에서의 실행 속도 비교

실험은 27개의 SIMD 기반 멀티프로세서를 가진 NVIDIA GeForce GTX 260 GPU와 Intel Core 2 Duo CPU가 갖추어진 PC와 Microsoft Windows Vista SP2 환경 하에서 수행되었다. 그림 1은 동일한 데이터를 분석했을 경우의 두 가지 구현의 수행 속도의 비교를 보여준다. GPU를 활용한 구현은 속도 면에서 기존의 CPU에서 실행되는 최적화된 프로그램 대비 최대 14% 빠른 결과를 얻을 수 있었다[3]. 또한 그림 2는 텍스처 메모리를 전체 FP-tree의 저장 공간으로 사용한 결과로 GPU의 전역 메모리만 사용한 구현보다 약 11% 나은 성능을 나타내었다.

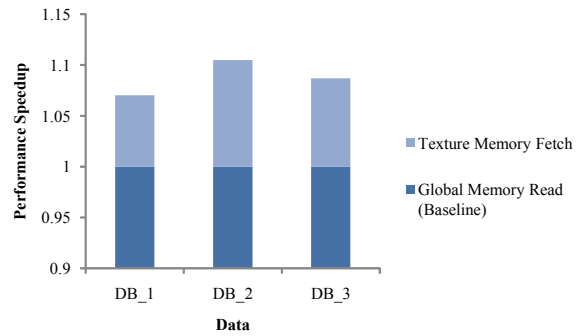


그림 2. 텍스처 메모리를 통한 성능 향상

### IV. 결론 및 향후 연구 방향

본 논문에서는 GPU 환경에서 빈발 항목집합의 마이닝을 위한 연관 분석 알고리즘인 FP-growth 알고리즘을 GPU 구조에 적합한 형태로 구현함으로써, 기존의 CPU 기반 프로그램 대비 성능의 향상을 보였다. 대부분의 알고리즘은 데이터 구조가 불규칙하기 때문에 병렬 프로세서의 구조를 최대한 활용하여 구현해야 한다. 앞으로 다양한 알고리즘이 GPU의 병렬 구조에 적합한 형태로 구현된다면 복잡한 문제를 더욱 빠른 시간 안에 해결 할 수 있을 것으로 기대한다.

### 감사의 글

“이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 임(No. 2009-0068969).”

### 참고문헌

- [1] S. Che, M. Boyer, J. Meng, D. Tarjan, J. Sheaffer, and K. Skadron, "A performance study of general-purpose applications on graphics processors using cuda," *Journal of Parallel and Distributed Computing*, vol. 68, no. 10, pp. 1370-1380, October 2008.
- [2] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Min. Knowl. Discov.*, vol. 8, no. 1, pp. 53-87, January 2004.
- [3] C. Borgelt, "An implementation of the fp-growth algorithm," in *OSDM '05: Proceedings of the 1st international workshop on open source data mining*.