텍스쳐 데이터를 위한 2차 캐쉬 구조를 가지는 그래픽 처리 장치의 성능 분석

김광복[O], 김철홍[*]
[O*]전남대학교 전자컴퓨터공학과
e-mail:loopaz63@gmail.com[O], chkim22@jnu.ac.kr[*]

Analysis of GPGPU Performance by dedicating L2 Cache for Texture Data

Gwang Bok Kim[O], Cheol Hong Kim[*]
[O*]School of Electronics and Computer Engineering, Chonnam National University

● 요 약 ●

최근 그래픽 처리 장치는 DRAM에 대한 접근을 줄이고자 여러 메모리 계층을 사용하고 있다. GPGPU의 L2 캐쉬는 요청 데이터의 타입에 따라 별도로 접근하는 L1 메모리와 다르게 레이턴시가 긴 DRAM에 접근하기 전에 모든 데이터 타입이 접근 가능한 캐쉬이다. 본 논문에서는 애플리케이션에서 명시하는 다양한 데이터 타입에 대하여 접근 및 적재를 허용하는 L2 캐쉬를 오직 텍스쳐 데이터만을 허용하도록 하여 변화하는 성능을 분석하고자 한다. 본 실험을 위해 텍스쳐 데이터 이외의 데이터 타입은 L2 캐쉬를 바이패스하여 바로 DRAM에 접근하도록 구조를 변경한다. 실험을 통한 분석 결과 텍스쳐 데이터만을 허용하는 경우 대부분의 벤치마크에서 성능 감소가 발생하여 기존 구조대비 평균 5.58% 감소율을 확인하였다. 반대로, 본 논문의 실험 환경에서의 L2 캐쉬의 적중률이 낮은 애플리케이션인 needle은 불필요한 L2 접근을 바이패스 함으로써 전체적인 성능 증가를 이끌어낸 것으로 분석된다.

키워드: 그래픽처리장치(GPU), 캐쉬(Cache), 바이패스(Bypass), 텍스쳐(Texture)

I. Introduction

현대 그래픽 처리 장치(GPU)는 범용 애플리케이션을 수행할 때 병렬연산수행을 적극 활용함으로써 높은 처리량을 확보할 수 있는 GPGPU(General Purpose Graphic Processing)에도 사용되고 있다. GPU메모리 구조는 그림 1과 같이 간략하게 표현할 수 있다. 각 SM(Streaming Multiprocessor)은 2차 캐쉬와의 데이터 전송을 위해 내부연결망(Interconnection Network)을 통해 연결되고 각 2차 캐쉬는 뱅크구조로 되어있어 각 DRAM 메모리 채널에 대해 버퍼기능을 수행한다. 이는 일반적으로 오프칩 DRAM에 접근할 때의 400~600사이클이 소모되기 때문에 지역성을 활용하여 접근수를 줄이고자 2차 캐쉬를 활용하고 있다.

NVidia의 Fermi 아키텍처는 메인메모리에서 분리된 주소영역을 가지고 있으며 GPGPU 프로그래밍 단계의 명시에 따라 서로 다른 데이터 타입에 접근할 수 있다. Fermi구조에서의 데이터 타입은 글로벌 데이터, 로컬 데이터, 텍스쳐 데이터, 상수 데이터 등으로 분류되는데, 어떤 스레드가 오프칩 메인 메모리로 데이터를 요청하면 일반적으로 2단계의 캐쉬 계층을 거친다. 또한, 로컬데이터와 글로벌 데이터의 경우 1차 데이터 캐쉬를 거치고 텍스쳐와 상수 데이터는 텍스쳐 캐쉬와 상수캐쉬에 각각 저장된다. 텍스쳐 캐쉬는 현대 GPU가 사실적인 그래픽 묘사를 위해 수행하는 텍스쳐 매핑(Texture Mapping)을 빠르게 수행하기 위한 필수적인 요소이다. NVIDIA G80 아키텍처부터는 텍스쳐 캐쉬를 2레벨 캐쉬로 변경하였다[1].

본 논문에서는 모의실험을 통해 다양한 데이터 타입을 저장하는 2차 캐쉬가 텍스쳐 데이터 타입의 접근만을 허용하는 구조를 가질 때 성능변화를 측정하고 분석하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 실험방법과 실험 환경에 대해 설명하고 3장에서는 실험결과에 대해 상세하게 분석한다. 마지막 4장에서는 본 논문의 결론에 대해서 기술한다.

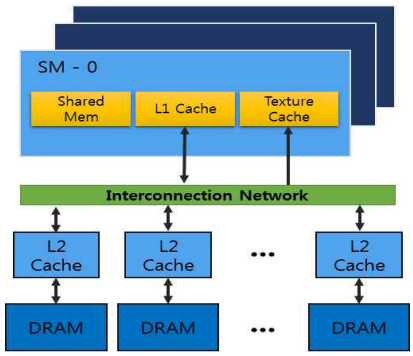

Fig. 1. Memory Hierarchy of GPU

II. Experimental Methodology

본 논문에서는 2차 캐쉬를 텍스쳐 데이터 요청 블록에 대해서 저장하고, 만약 다른 데이터 타입을 요청하는 경우는 바이패스를 통해 바로 오프칩 DRAM에 접근하도록 구성한다. 이 외 본 논문에서 수행된 실험 환경 요소들은 표 1에서 보는 바와 같이 구성한다. 또한, 정확한 성능 측정을 위해서 사이클 단위로 성능을 측정하여 시뮬레이션을 수행할 수 있는 GPGPU-SIM[2]

시뮬레이터를 사용한다.

텍스쳐 캐쉬 접근이 필요한 벤치마크와 그렇지 않은 벤치마크를 모두 포함하여 성능변화를 측정하고자 본 논문에서는 CUDA SDK[3], Rodinia benchmarks[4], ISPASS2009 benchmarks[5]에서 각각 벤치마크를 선정하여 총 6개의 벤치마크를 사용한다.

Table 1. System Configuration

Parameter	Value
Number of SM	16
Warp Size	32
Data L1 Cache	32set, 128bytes, 4way
Data L2 Cache	64set, 128bytes, 8way
Texture Cache	4set, 128bytes, 24way
Clock (Core, InterConnection, DRAM)	700MHz, 700MHz, 924MHz

III. Experiment Results Analysis

그림 2는 기존의 GPU구조와 같이 모든 데이터 타입을 허용하는 2차 캐쉬를 사용할 때 선택된 6개의 벤치마크 프로그램들의 성능결과와 텍스쳐 데이터만을 저장하는 2차 캐쉬 구조를 가지는 구조에서의 성능결과를 각각 보여준다. 그림에서 가로축은 수행하는 응용 프로그램이고 세로축은 성능의 지표로 널리 사용되는 한 사이클 당 수행된 명령의 개수인 IPC(Instructions per Cycle)를 나타내고 있다. 2차 캐쉬를 텍스쳐 데이터만을 위해 사용할 경우 하나의 벤치마크를 제외하고 모두 성능저하가 발생한다. MUM 벤치마크는 텍스쳐 캐쉬에서의 적중률이 99%에 달하기 때문에 재사용성이 상당히 낮았다. 따라서 2차 캐쉬를 텍스쳐 데이터만 접근하도록 허용한다면 성능이 크게 감소한 것을 볼 수 있다. 그림 2에서 보이는 바와 같이 2차 캐쉬를 텍스쳐 데이터만 허용한 경우 기존구조에 비해 18%의 성능저하가 발생하여 가장 큰 감소율을 보였다. 텍스쳐 데이터만을 허용하는 2차 캐쉬를 사용한 경우 유일하게 needle 벤치마크에서 성능향상이 발생하였다. 그 원인은 그림 3에서 보이는 바와 같이 벤치마크의 지역성 특징에 따른 캐쉬의 효율성으로 분석된다. 그림 3은 2차 캐쉬를 텍스쳐 데이터만을 허용하도록 한 구조에서 2차 캐쉬의 적중률을 나타낸다. 벤치마크 needle을 제외한 모든 벤치마크는 2차 캐쉬의 적중률은 23%~95%를 보인다. 특히 MonteCarlo와 MUM 벤치마크는 각각 95%와 87%의 캐쉬 적중률로 상대적으로 높은 캐쉬 효율성을 보여준다. needle 벤치마크는 2차 캐쉬가 모든 데이터를 허용하는 경우에 2차 캐쉬에서의 요청 적중률이 0.78%에 불과했다. 즉, needle 벤치마크는 2차 캐쉬에서 데이터 재사용율이 매우 떨어지기 때문에 2차 캐쉬가 텍스쳐타입만 허용할 경우 불필요한 2차 캐쉬 접근을 줄임으로써 성능증가로 이어진 것으로 분석된다.

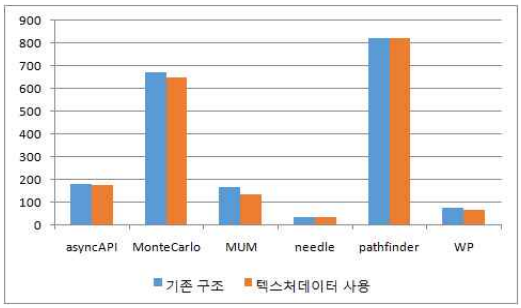

Fig. 2. Performance Comparison

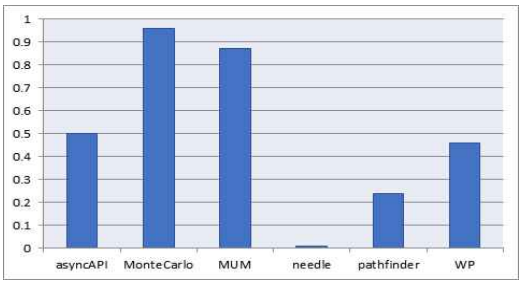

Fig. 3. Cache Hit Rate for L2 Cache

IV. Conclusions

본 논문에서는 내부 연결망에 연결되어 SM에서 요청하는 여러 데이터 타입을 적재할 수 있는 2차 데이터 캐쉬를 오직 텍스쳐 데이터만을 저장하게 변경함으로써 성능 변화를 측정하고 분석하였다. 데이터 재사용성이 매우 낮은 벤치마크의 경우 텍스쳐 데이터만을 허용함으로써 오히려 성능증가를 이끌어낼 수 있음을 확인하였다. 본 논문의 결과를 바탕으로 모든 데이터 타입을 항상 허용하는 캐쉬가 아닌 각 벤치마크 특성을 고려하여 특정 데이터 타입을 허용하도록 하는 기법을 연구하는데 도움이 될 것으로 기대한다.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(NRF-2015R1D1A3A01019454)

References

[1] D. Michael, "Texture Caches", IEEE Micro, vol. 32, No. 32, pp. 136-141, May/June 2012.
[2] GPGPU-SIM, Available at http://gpgpu-sim.org/
[3] NVIDIA SDK Avaiable at https://developer.nvidia.com/cuda-downloads
[4] Rodinia Benchmarks Available at http://lava.cs.virginia.edu/Rodinia/download_links.htm
[5] A. Bakhoda, Gl. L.. Yuan, W. W. W. Fung, H. Wong, and T.M, Adamodt, "Analyzing CUDA Workloads Using a Detailed GPU Simulator" In Proceedings of International Symposium on Performance Analysis of System Software, pp. 163-174, 2009.