한국컴퓨터정보학회 하계학술대회 논문집 제21권 제2호 (2013. 7)

안드로이드 폰에서의 모바일 GPU 성능 분석 및 최적화

조창우[O], Sharma Ashok[*], 김신덕[+]
[O*+]연세대학교 컴퓨터과학과
e-mail: cwjo@yonsei.ac.kr[O], i.ashok.s@gmail.com[*], sdkim@yonsei.ac.kr[+]

Performance Analysis and Optimization of mobile GPU in Android Phone

Chang-Woo Cho[O], Sharma Ashok[*], Shin-Dug Kim[+]
[O*+]Dept. of Computer Science, Yonsei University

● 요 약 ●

본 논문에서는 최신 안드로이드 기반 상용 스마트폰의 모바일 GPU 성능 향상을 위한 방법론을 제안한다. 동일 하드웨어를 가지고 스마트폰을 개발하더라도 제조사의 역량에 따라 소프트웨어 최적화의 정도가 달라서 그래픽 성능 차이가 날 수 있다. 그러므로 우리는 시스템 소프트웨어 레벨에서 그래픽 품질에 아무런 영향을 주지 않고 성능 향상을 이끌어낼 수 있는 기법에 대해 소개한다. 이를 위해 A사, B사 안드로이드 스마트폰을 대상으로 안드로이드 커널에 따른 분석을 수행하였고, GPU 디바이스 드라이버에 따른 분석을 수행하였으며, 마지막으로 타사 휴대폰과의 성능 비교를 통해 이 결과를 비교 분석하였다. 결과적으로 GPU 디바이스 드라이버 변경과 안드로이드 커널 변경을 시도함으로써 B사 대비 68%의 성능을 보인 A사 스마트폰의 성능을 96%까지 향상시킬 수 있었다.

키워드: 모바일 GPU(Mobile GPU), 3차원 그래픽스(3D Graphics), 성능 최적화(Performance Optimization)

I. 서 론

최근 모바일 게임 산업의 시장규모 확대에 따라 내장형 3차원 그래픽 프로세서 유닛 시장은 꾸준히 성장해왔다. 이에 모바일 3차원 그래픽 프로세서 유닛 (이하 모바일 GPU)은 멀티미디어 폰의 매우 중요한 핵심 요소이고, 모바일 GPU 성능에 대한 요구는 나날이 높아져가고 있다. 하지만, 급격한 성능의 발전을 보이는 현재의 휴대용 단말기라도 많은 수의 유저가 요구하는 수준의 고품질 그래픽을 위한, 방대한 양의 연산을 실시간으로 처리하기에는 무리가 있다. 이러한 3차원 그래픽 기술과 휴대기기의 응용 기술은 기존의 CPU의 계산 능력, 메모리, 전력 소모, 하드웨어 구성 및 성능 등 여러 측면에서 서로 상반된 특징을 가지므로, 모바일 환경에 적합한 알고리즘의 선택 및 최적화 작업이 필요하다.

II. 관련 연구

1. 관련연구

전통적으로 3D 그래픽 애플리케이션들은 데스크톱 컴퓨터나 모바일 GPU가 등장하게 되었다. 이에 따라 모바일 3D 게임 산업은 급격하게 성장하고 있으며, 모바일 기기의 제한된 연산능력, 메모리 크기, 파워의 한계에서 고품질 3D 게임을 동작시키기 위해 상당한 기술 혁신이 요구되고 있다.

최근에는 GPU를 그래픽 처리 목적으로만 사용하는 것에서 벗어나 범용 데이터 계산에도 이용할 수 있도록 하기 위한 GPGPU[1](General Purpose GPU) 기술이 널리 연구되고 있다. 특히 NVIDIA사에서 GPGPU 아키텍처인 CUDA(Compute Unified Device Architecture)를 발표하여 보다 유연한 GPU 프로그래밍 환경을 제공하고 있다. 또한, GPGPU에서는 많은 연산을 병렬로 수행함에 따라 메모리 지연시간을 감출 수 없는 상황이 발생하는데, 이를 해결하기 위한 많은 노력들이 있었다. 여기서 Jaekyu Lee. [2]는 MT-prefetching(Many-thread aware prefetching) 라고 불리는 GPGPU 시스템을 위한 새로운 하드웨어, 소프트웨어 프리펫칭 메커니즘을 제안하였다. 현재까지 PC기반 GPGPU는 연구가 활발히 진행되고 있으나, 아직까지는 모바일 GPGPU가 시장에 본격적으로 도입되지 않고 있다.

한편, 시스템 레벨 관점에서 성능 최적화를 위한 방안이 연구되고 있다. Dynamic Voltage and Frequency Scaling(DVFS) 기술은 성능과 전력 최적화를 위해 사용되고 있다. 이와 같은 에너지 절약 기술들은 정확하고 빠른 작업부하 예측을 요구한다. Bren C.

· 본 연구는 정부(교육기술과학부)의 재원으로 이공분야기초연구사업의 일환으로 수행함. [과제번호 2012R1A1A2043400]
† 교신 저자

Mochockit. [3]는 3D 그래픽스 작업부하를 예측하기 위한 시그니처 기반 예측기술을 제안하였다. 시그니처 기반 예측은 간단하며, 정교한 모델 예측을 요구하지 않기 때문에 실시간 예측에 효율적이다. 또한, 표준 히스토리 기반 예측기와 달리 결과를 이끈 원인뿐만 아니라 이전 결과를 사용하여 더 정확한 예측을 할 수 있다.

Ⅲ. 본 론

1. 본론

1.1 성능 비교 분석

본 비교 실험에서는 Adreno 220 GPU을 탑재한 A사 상용 스마트폰을 대상으로 성능 비교 실험을 수행하였다. 그래픽 성능에 영향을 주는 많은 요인 중 본 연구에서는 GPU 드라이버 버전과 안드로이드 커널 버전에 따른 영향을 중점적으로 분석하였다. 또한 동일 하드웨어 스펙을 가진 B사 스마트폰과의 성능 비교를 통하여 성능 저하의 원인을 분석하였고 성능 최적화를 수행하였다. 실험의 전체적인 구조는 그림 1과 같다.

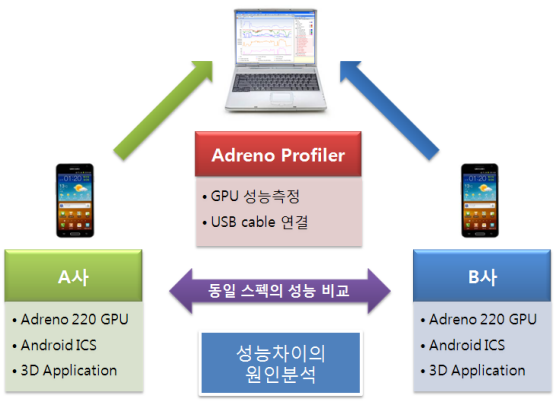


그림 1 - 실험 환경의 전체적인 구조
Fig 1 - Overall structure of experimental environment

1.2 실험 환경

본 비교 실험에서 3D 성능 향상의 정도를 파악하고, 성능을 비교 분석하기 위해서는 성능 측정할 대상이 필요하다. 사용 가능한 다양한 벤치마크가 존재하나 성능에 대한 객관적인 판단을 위해 이름이 널리 알려진 범용 3D 벤치마크를 사용하였다. 아래 그림은 본 연구에 사용한 3D 벤치마크이다.

그림 2 - GLBenchmark-Egypt[4], Nenamark2[5]
Fig 2 - GLBenchmark-Egypt[4], Nenamark2[5]

본 연구에서 성능 비교에 사용한 두 스마트폰의 GPU는 Adreno 220이다. 이 GPU를 지원하는 성능 프로파일러에는 Qualcomm사에서 제공하는 Adreno Profiler[6]가 있다. Adreno Profiler는 모바일 3D Application의 상세한 성능 분석을 위한 GPU Profiler로써 Scrubber 모드와 Grapher 모드를 제공하며, OpenGL ES API의 함수 호출을 추적하고 에뮬레이션 할 수 있다. 이 프로파일러의 주용도는 Application 테스트와 디버그 용도로 사용하지만, 3D 성능 분석을 확인하기 위해 사용하였다. 본 연구의 실험 환경은 표 1과 같다.

표 1. 실험 환경
table 1. Experimental Environment

항목	값
OS	Android 4.0.4
CPU	1.5Ghz Dual-Core
GPU	Adreno 220
RAM	1GB
Profiler	Adreno Profiler 2.7

그림 3 - 프로파일링 환경
Fig 3 - Profiling Environment

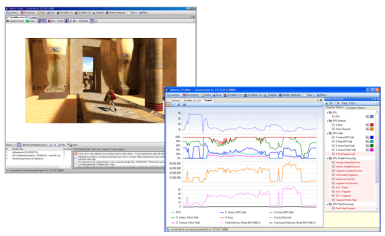


그림 4 - Scrubber 모드와 Grapher 모드
Fig 4 - Scrubber Mode and Grapher Mode

1.3 안드로이드 GPU 드라이버에 따른 성능 영향

이 실험은 GPU 드라이버가 3D 성능에 미치는 영향을 알아보기 위한 것이다. 이 실험을 하기 전에 각 안드로이드 버전에 따른 GPU 디바이스 드라이버의 버전확인이 필요하였다. 디바이스 드라이버는 모바일 기기 내 공유 라이브러리 파일(.so)로 존재하며, 실제 이러한 라이브러리 파일의 변경으로 드라이버 변경이 가능하다.

표 2. Android 버전에 따른 GPU 디바이스 드라이버 버전
table 2. GPU device driver version by android version

Android 버전	GPU 디바이스 드라이버 버전
ICS 4.0.4	2184622 (최신 드라이버)
CM9 RC1	2184622 (최신 드라이버)
CM9 RC1	2131267 (구 드라이버)

드라이버 변경에 따른 성능의 변화를 알아보기 위해 ICS에서 최신 드라이버(2184622)를 다운그레이드하여 그 변화를 살펴보려 하였으나, 구 드라이버(2131267)가 ICS에 호환되지 않아, CM9 커널에서 최신 드라이버(2184622)를 구 드라이버(2131267)로 다운그레이드 하여 그 변화를 살펴보았다.

표 3. GPU 드라이버에 따른 벤치마크 별 성능 측도
table 3. Performance measurement of each benchmark by the GPU device driver

	Egypt		Nenamark2	
Metrics	〈2131267〉	〈2184622〉	〈2131267〉	〈2184622〉
FPS	18.09	17.62	32.18	35.32
% Busy	99.45	99.2	99.99	99.87
Clock/sec	263,682,238	262,239,901	266,475,589	266,552,406
Stall	35.09	32.56	55.17	49.02
BW	404.08	389.19	388.57	334.75

표 3을 보면 Egypt는 모든 지표에서 모두 감소하였으며, Nenamark2는 FPS, Clocks/second를 제외한 나머지 지표의 감소를 보였다. Egypt는 구 드라이버에서, Nenamark2는 최신 드라이버에서 더 높은 FPS 값을 가졌다. 따라서 디바이스 드라이버 버전이 높아질수록 모든 벤치마크의 FPS가 향상되는 것은 아님을 알 수 있다. 다시 말해서, 벤치마크의 특성에 따라 구 버전이 더 적합할 수도 있다.

1.4 안드로이드 커널에 따른 성능 영향

이 실험은 커널 버전이 3D 성능에 미치는 영향을 알아보기 위한 것이다. 1.3과 마찬가지로, 드라이버 버전은 동일하게 두고 커널버전을 다르게 하여 성능을 비교하였다.

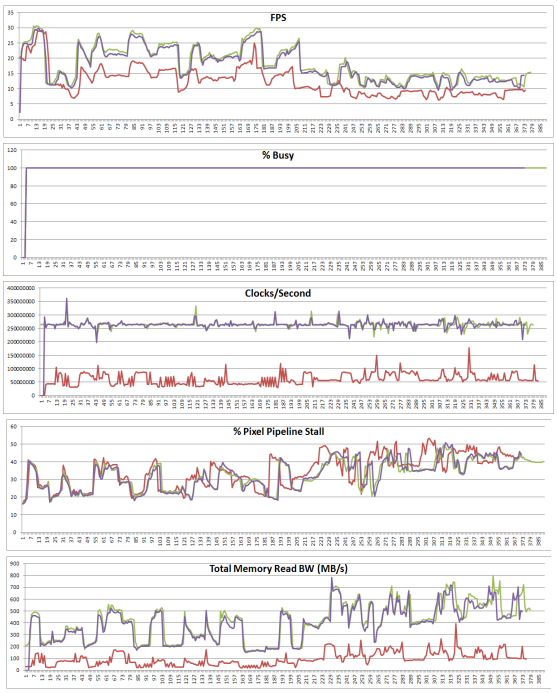
표 4. 안드로이드 커널에 따른 벤치마크 별 성능 측도
table 4. Performance measurement of each benchmark by android kernel

	Egypt		Nenamark2	
Metrics	ICS	CM9	ICS	CM9
FPS	12.58	17.62	33.01	35.32
% Busy	98.92	99.2	97.13	99.87
Clock/sec	60,903,960	262,239,901	23,929,556	266,552,406
Stall	34.73	32.56	47.57	49.02
BW	93.95	389.19	27.92	334.75

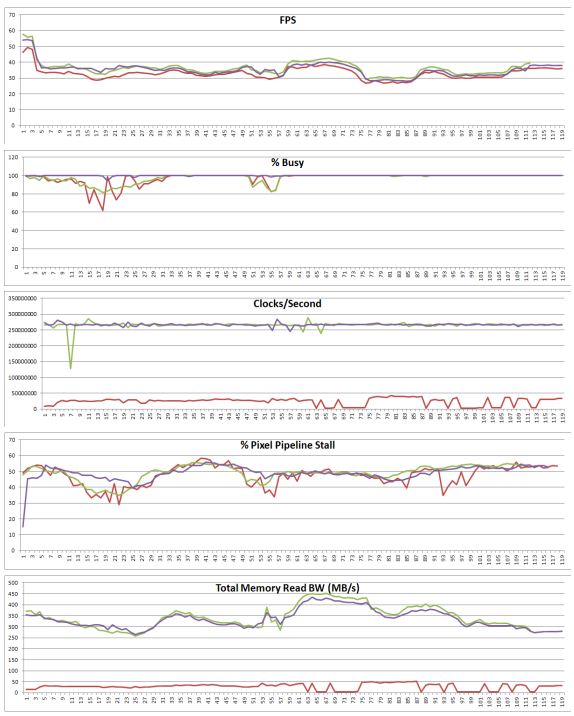
표 4를 보면 Egypt는 stall을 제외한 모든 지표에서 증가하였으며, Nenamark2는 모든 지표에서 증가하였다. 공통적으로 ICS보다 CM9 커널을 사용했을 때 FPS가 향상되었으며, Clocks/sec와 BW가 급격히 증가된 것을 볼 수 있다.

디바이스 드라이버와 커널에 따른 성능 분석 결과 3D 성능은 응용프로그램에 따라 서로 다른 양상을 보인다. 또한 1.3에서 분석한 결과와 비교해 볼 때, 3D 성능은 디바이스 드라이버 버전보다는 어떠한 커널을 쓰느냐에 따라 더 많은 영향을 받는다는 것을 알 수 있다.

1.5 동일 스펙의 두 기기 간 성능 비교



— A사 ICS — A사 CM9 — B사 ICS
그림 5 - GLBenchmark-Egypt 벤치마크 성능 비교
Fig 5 - Performance Comparison of GLBenchmark-Egypt



— A사 ICS — A사 CM9 — B사 ICS
그림 6 - Nenamark2 벤치마크 성능 비교
Fig 6 - Performance Comparison of Nenamark2

그림 5와 그림 6에 있는 그래프를 보면 두 기기의 하드웨어 스펙은 동일하더라도 두 벤치마크에서 ICS 커널에서의 3D 성능은 상당한 차이를 보인다. 그러나 측정결과처럼 A사 ICS 커널을 CM9으로 교체 시 B사의 ICS의 성능에 근접하고 있다. 따라서 A사 ICS 버전의 3D성능을 최소한 CM9까지 더 향상시킬 수 있는 가능성이 존재한다는 것을 알 수 있다.

Ⅳ. 결 론

본 논문에서는 최신 안드로이드 OS를 탑재한 동일 스펙의 두 스마트폰을 대상으로 GPU 디바이스 드라이버 변경과 커널 변경을 시도하였다. 또한, 성능 프로파일링을 수행하여 두 기기 간 성능 차이가 나는 원인을 분석하였고, 방법론 적용 전후의 성능을 비교 분석하였다.

결과적으로 GPU 디바이스 드라이버 변경과 커널을 변경을 시도함으로써 B사 대비 68%의 성능을 보인 A사 스마트폰의 성능을 96%까지 향상시킬 수 있었다. 향후 본 연구 결과를 바탕으로 스마트폰 제조사들이 하드웨어에 최적화된 스마트폰을 개발할 때 본 방법이 성능 최적화에 도움이 될 것을 기대할 수 있다.

참고문헌

[1] Seyong Lee, Seung-Jai Min, and Rudolf Eigenmann, "OpenMP to GPGPU: A Compiler Framework for Automatic Translation and Optimization", In Proc. ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, pp. 101, 2009

[2] Jaekyu Lee, Nagesh B. Lakshminarayana, Hyesoon Kim, and Richard Vuduc, "Many-Thread Aware Prefetching Mechanisms for GPGPU Applications", In Proc. of MICRO, pp. 213-224, December 2010.

[3] B. Mochocki, K. Lahiri, S. Cadambi, and X. S. Hu. "Signature-based workload estimation for mobile 3d graphics", In Design Automation Conference (DAC), pp. 592-597, 2006.

[4] GLBenchmark 2.1.4, Kishonti Informatics Ltd, http://www.glbenchmark.com, (2012.7.8.)

[5] NenaMark2, Nena Innovation AB, retrieved June 26, 2013, http://nena.se/nenamark/view?version=2

[6] Adreno Profiler, Qualcomm, retrieved June 26, 2013, https://developer.qualcomm.com/mobile-development/mobile-technologies/gaming-graphics-optimization-adreno/tools-and-resources