

# CPU 부하가 큰 쓰레드를 가진 모바일 게임에서 QoS를 고려한 전력 관리 기법

김민성<sup>o</sup>                      김지홍

서울대학교

minsungk@davinci.snu.ac.kr, jihong@davinci.snu.ac.kr

## QoS-Aware Power Management of Mobile Games with High-Load Threads

Minsung Kim<sup>o</sup>                      Jihong Kim

Seoul National University

### 요 약

스마트폰을 비롯한 다양한 모바일 플랫폼에서 게임은 가장 많은 전력을 소비하여 배터리 사용시간을 단축시키는 응용 중 하나이다. 따라서 모바일 게임은 모바일 플랫폼의 배터리 사용시간을 늘리기 위한 중요한 전력관리 대상이다. 그러나 모바일 게임 실행 중 배터리 용량이 급격히 감소하는 현상은 빈번하게 발생하는 문제이며 실제 사용자가 많은 여러 모바일 게임의 동작을 살펴보았을 때 멀티 쓰레드로 동작하는 동안 특정 쓰레드가 불필요하게 CPU 사용량을 높게 점유하여 에너지가 낭비되는 경우가 발생하였다. 따라서 본 논문은 게임 실행 도중에 사용자 Quality of Service(QoS)와 무관하게 비정상적으로 CPU 사용량이 높은 쓰레드를 검출하고 이종 멀티프로세싱(Heterogeneous Multi-Processing)의 특성을 활용하여 QoS 저하 없이 에너지 이득을 달성하는 최적화 기법을 제안한다. 제안된 기법을 실제 스마트폰(삼성 Galaxy S6)에 적용하여 QoS 저하 없이 최대 58%의 에너지 효율 향상이 가능하였다.

### 1. 서 론

게임은 스마트폰, 태블릿 PC와 같은 모바일 플랫폼에서 가장 인기 있는 응용 중 하나이다. 따라서 모바일 게임에 의한 에너지 소모는 제한된 배터리 용량을 가진 모바일 플랫폼에서 사용시간을 단축시키는 핵심 요소이다. 모바일 플랫폼이 고화질, 고사양으로 발전되어감에 따라 최근 게임은 더욱 높은 성능과 많은 에너지를 요구하게 되었으며 이로 인해 에너지 효율을 극대화하는 전력관리 기법이 요구되고 있다. 모바일 게임의 전력관리에 관련한 기존 연구들[1], [2]은 주로 CPU 주파수 또는 GPU 주파수에 따른 성능과 소모 전력에 대한 모델을 만들고 이에 최적화된 dynamic voltage and frequency scaling(DVFS)를 수행하여 에너지 효율을 향상시키는 기법을 다루고 있다. 그러나 기존 연구들은 모바일 게임 실행 중에 동작하는 쓰레드가 필요한 성능 이상으로 CPU를 사용하는 경우와 같은 상황에서는 지나치게 높은 CPU 주파수로 동작하여 에너지가 낭비되는 것을 제어할 수 없다. 본 연구는 모바일 게임 동작 중 비정상적으로 CPU 사용량이 높은 쓰레드를 검출한 뒤 해당 쓰레드를 에너지 효율이 우수한 CPU 코어에 할당하여 QoS 저하 없이 소모 전력을 감소시키는 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구의 동기를

제시하고 3장에서는 비정상 쓰레드의 검출과 검출된 쓰레드를 강제 마이그레이션 시키는 기법을 제안한다. 4장에서는 기법에 대한 평가를 수행하고 5장에서는 본 연구의 결론을 맺는다.

### 2. 연구 동기

최근 모바일 플랫폼에는 이종 멀티프로세서(Heterogeneous Multiprocessor) SoC가 탑재되고 있다. 이종 멀티프로세서 SoC는 성능과 소모 전력 특성이 서로 다른 마이크로 아키텍처의 CPU 코어들로 구성되어 있으나 동일한 명령어 집합 구조(Instruction Set Architecture)를 가진다. 그림 1은 삼성 Galaxy S6에 탑재된 Exynos 7 Octa의 블록 다이어그램이며 이 SoC는 ARM사의 이종 멀티프로세서 구조인 big.LITTLE 구조[3]를 기반으로 하고 있다. big은 소모 전력이 큰 고성능의 코어이며

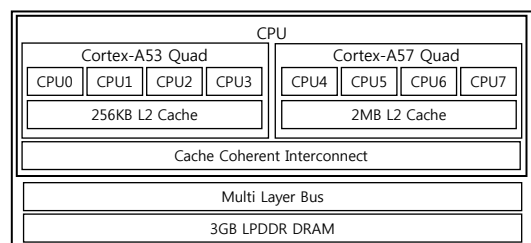


그림 1 Exynos 7 Octa(7420)[4] 블록 다이어그램

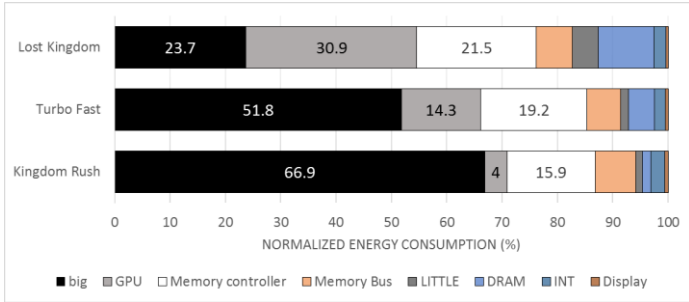


그림 2 게임 실행 시 스마트폰 SoC의 주요 컴포넌트별 에너지 소비량

LITTLE은 전력효율이 우수하지만 상대적으로 성능이 낮은 코어이다. big.LITTLE 구조에서 CPU 부하가 큰 스레드는 고성능을 요구하는 것으로 간주되어 big 코어를 사용하게 된다. 따라서 모바일 게임 실행 중에 CPU 부하가 큰 스레드들은 big 코어에서 동작하며 에너지 소모에 높은 비중을 차지하게 된다.

모바일 게임의 복잡도가 높아짐에 따라 점점 더 많은 스레드로 구현이 되고 있으며 이 과정에서 지나친 성능 위주의 최적화가 이뤄지거나 다양한 버그에 의해 특정 스레드가 불필요하게 높은 CPU 사용량을 보이는 경우가 나타나고 있다. 사용자가 많은 여러 게임들을 검토해 본 결과 단일 스레드가 하나의 CPU 코어를 100% 점유하는 현상이 관찰되었다. Kingdom Rush, Turbo fast, Lost kingdom, Assassin's Creed Pirates, Dragon village TCG, Zombie WW 등의 게임에서 이 현상을 확인하였으며 해당 스레드에 의해 big 코어가 최고 주파수에서 지속적으로 동작하여 전력 소모가 커지고 발열이 심해지는 문제가 발생하였다. 이중 일부 게임은 업데이트를 통해 개선된 경우도 있으나 이러한 문제는 개발자의 실수와 같은 버그나 전력소모를 고려하지 않은 최적화 등 여러 원인에 의해 발생할 수 있다. 그림 2는 이러한 문제가 발생하는 3개의 게임을 Galaxy S6에서 1분 동안 수행하여 주요 컴포넌트의 에너지 소모량을 실측한 결과이다. 단일 스레드에 의해 big 코어가 포함된 클러스터가 최고 주파수로 유지되어 전체 에너지 소모량 중 최대 66.9%의 비중을 차지하였다. 그림 3은 Kingdom Rush 실행 후 하나의 big 코어를 100% 점유하는 스레드를 임의의 LITTLE 코어에 강제로 할당하였을 때 소모전력 변화를 보여준다. 강제 할당 후 big 클러스터의 소모전력은 거의 없어지고 LITTLE 클러스터의 소모전력은 크게 증가하였으나 코어 간 에너지 효율의 차이 때문에 전체 SoC 소모전력은 크게 개선되었다. 따라서 QoS의 저하가 없는 수준에서 비정상 스레드를 LITTLE 코어에서 동작하게 할 수 있다면 소모전력을 최소화하여 에너지 효율을 향상시킬 가능성이 있음을 알 수 있다.

본 논문에서는 게임의 스레드별 부하를 추적하고 특정 스레드가 정상적인 범주를 벗어나 지나치게 높은 부하를 가지고 CPU를 점유하고 있는지 검출하는 방법을 제안한다. 또한

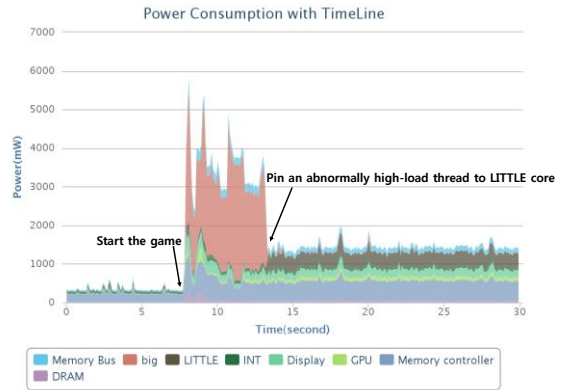


그림 3 Kingdom Rush 실행 후 단일 big 코어를 100% 점유하는 스레드를 LITTLE 코어에 할당 시 소모전력 변화

검출된 스레드를 LITTLE 코어로 강제 마이그레이션한 뒤 QoS가 동일하게 유지되는지 검사하여 이를 통해 QoS 저하 없이 에너지 효율 향상에 기여하는 것이 목표이다.

### 3. 비정상 스레드 검출 및 QoS를 고려한 에너지 최적화 기법

big.LITTLE 구조에서 이종의 코어 간 마이그레이션을 수행하는 기법은 클러스터 마이그레이션, CPU 마이그레이션, 글로벌 태스크 스케줄링(GTS)[5]이 있으며 본 연구에서는 GTS에 기반한 기법을 제안하였다. 스레드별 부하(L)는 스케줄러의 per-entity load tracking[6]에 의해 산출되며 다음 식(1)과 같다.

$$L = L_0 + L_1 \cdot y + L_2 \cdot y^2 + L_3 \cdot y^3 + L_4 \cdot y^4 + \dots \quad (1)$$

$$y^{32} = 0.5$$

$L_0, L_1, L_2, \dots$ 는 1밀리초 간격으로 스레드가 실행중 또는 실행 가능한 상태에 머문 시간을 측정된 값이며 최근 값에 가중치를 높이기 위해 감쇠계수  $y$ 를 곱한 뒤 합산하여 부하를 산출하였다. 스레드의 부하가 역치를 넘어설 경우 해당 스레드는 LITTLE 코어에서 big 코어로 마이그레이션 된다. 본 연구에서는 스케줄러가 산출한 각 스레드의 부하를 0에서 1023사이의 값을 갖도록 변환하고 이를 이용하여 비정상적으로 CPU를 많이 사용하는 스레드의 후보를 검출한다. 그림 4는 제안된 기법의 플로우 차트이다. 스케줄러는 동작 중인 스레드의 부하를 지속해서 산출하고 갱신한다. 스레드의 부하는 식 (1)에 따라

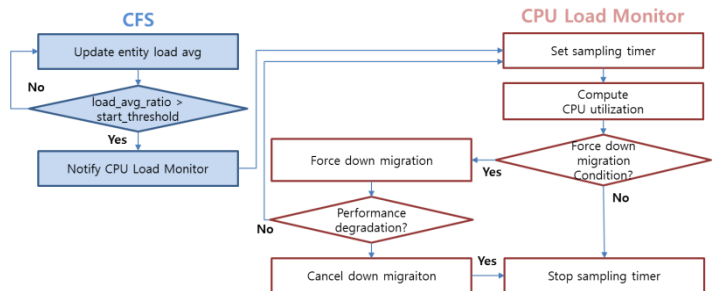


그림 4 QoS를 고려한 비정상 스레드 검출 및 강제 마이그레이션 기법 플로우 차트

감쇠되어 산출되므로 부하가 1000이 넘는 값을 가진 채로 장시간 유지되는 스레드는 극소수이다. 부하가 1000이 넘는 스레드가 발견되면 스케줄러는 CPU Load Monitor(CLM)에 해당 스레드를 통보하고 CLM은 타이머를 설정하여 100밀리초 주기로 해당 스레드의 부하와 CPU 사용량을 측정한다. 4초 이상 CPU 사용량이 95%를 초과하고 해당 스레드의 부하가 1000이상일 경우 해당 스레드를 비정상 스레드로 판단하여 LITTLE 코어로 강제 마이그레이션 시킨다. 게임에서 QoS의 측정 기준은 Frame per Second(FPS)로 정의하며[5] 강제 마이그레이션 수행 후에 FPS가 이전 대비 하락하거나 스레드의 부하가 700 이하로 낮아질 경우 강제 마이그레이션을 취소한다. 그렇지 않을 경우 지속적으로 타이머를 재설정하여 100밀리초 주기로 스레드의 부하와 FPS를 검사한다. FPS는 display control device driver를 통해 측정하였으며 이 기법의 알고리즘은 모두 커널 영역에 구현되었다.

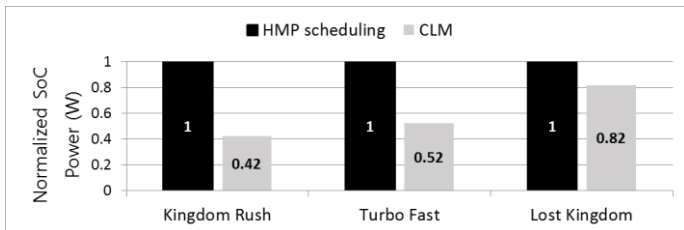
#### 4. 제안 기법의 평가

##### 4.1. 실험환경

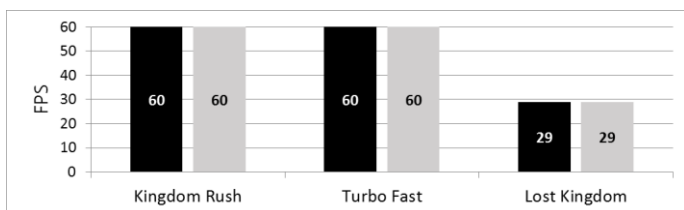
실험은 Exynos 7 Octa(7420)이 탑재된 삼성 Galaxy S6 실험보드에서 진행하였다. 기존 GTS 방식과 본 연구에서 제안한 QoS를 고려한 비정상 스레드 강제 마이그레이션 방식을 비교하기 위하여 단일 스레드가 하나의 CPU를 100% 점유하는 현상이 확인된 게임 중 3개를 선정하여 5분 동안 플레이 시 SoC의 소모전력과 평균 FPS를 측정하였다.

##### 4.2. 실험결과

그림 5는 기존 GTS와 제안한 기법(CLM) 간의 소모 전력과 FPS를 비교한 그래프이다. (a)는 제안한 기법에서 기존 대비 평균 소모전력이 최대 58%, 평균 45% 감소한 것을 보여 준다.



(a) SoC Power consumption



(b) 평균 FPS

그림 5 기존 GTS 대비 제안한 기법(CLM)의 소모전력, FPS 비교

CLM에 의해 비정상 스레드가 강제 마이그레이션 되면 big 클러스터의 소모전력이 감소하여 전체 SoC 소모전력이 크게 감소한 것을 볼 수 있다. 그림 2에서 big 클러스터의 소모전력 비중이 가장 높았던 Kingdom Rush가 가장 개선이 많이 된 것을 확인 할 수 있다. 강제 마이그레이션이 된 스레드는 LITTLE 클러스터의 단일 CPU를 여전히 100% 점유하지만 LITTLE 코어의 에너지 효율이 big 코어에 비해 훨씬 우수하기 때문에 전체 소모 전력은 감소한 것으로 볼 수 있다. 그림 5의 (b)에서 나타나는 바와 같이 FPS는 기존 대비 동일한 수준을 유지하므로 본 기법으로 인한 QoS 저하는 발생하지 않았다.

#### 5. 결론 및 논의

본 논문에서는 모바일 게임에서 확인된 과도하게 높은 CPU 사용량을 요구하는 스레드를 이중 멀티프로세서 SoC의 특성과 스레드의 부하를 추적하는 기법을 이용하여 에너지 효율이 우수한 코어로 강제 마이그레이션 시키는 기법을 통해 에너지 효율을 향상시켰다.

개별 스레드의 특성과 QoS와의 관계를 명확히 분석하고 모델링하여 비정상적으로 부하가 큰 스레드 이외의 다른 스레드도 최적의 코어와 주파수에 할당하는 것이 향후 연구할 과제이다.

#### 참고 문헌

- [1] A. Pathania, Q. Jiao, A. Prakash, and T. Mitra, "Integrated CPU-GPU Power Management for 3D Mobile Games," in *proceeding of the Design Automation Conference*, 2014
- [2] A. Pathania, A. E. Irimiea, A. Prakash, and T. Mitra, "Power-Performance Modelling of Mobile Game Workloads on Heterogeneous MPSoCs," in *proceeding of the Design Automation Conference*, 2015
- [3] ARM Ltd., big.LITTLE technology, <https://www.arm.com/products/processors/technologies/biglittleprocessing.php>
- [4] Samsung Electronics Co., Ltd., Exynos 7 Octa, <http://www.samsung.com/exynos>
- [5] B. Jeff, "big.LITTLE Technology Moves Towards Fully Heterogeneous Global Task Scheduling Improving Energy Efficiency and Performance in Mobile Devices", *ARM white paper*, 2013
- [6] P. Turner, "CFS Per-Entity Load Tracking", <https://lwn.net/Articles/504013>, 2012
- [7] M. Claypoola, K. Claypoolb, and F. Damaab, "The Effects of Frame Rate and Resolution on Users Playing First Person Shooter Games," in *proceeding of the Electronic Imaging*, 2006