

동적 임계치 적용을 통한 이기종 멀티프로세서의 소비전력 개선 기법

신동익, 신동군
성균관대학교 정보통신공학부
dongig@skku.edu, dongkun@skku.edu

Dynamic Threshold Technique for Power Optimization of Heterogeneous Multi Processor

Dongig Sin, Dongkun Shin
Sungkyunkwan University

요 약

이기종 멀티프로세서는 고성능 프로세서와 저전력 프로세서를 혼용하여 소비전력을 줄이기 위해 설계된 아키텍처이다. 하지만 기존 스케줄러는 불필요하게 유휴 CPU를 사용하게 되는 경우가 있고, 고성능 코어들의 부하 불균형을 유발하여 비 효율적으로 전력을 소비하는 문제점이 있다. 이 논문에서는 동적 임계치 기법을 적용하고, 스케줄링 기법을 개선하여 부하 불균형을 낮춤으로써 CPU의 소비전력을 최대 10.7% 감소시켰다.

1. 서 론

동일 명령어 집합을 사용하는 이기종 멀티프로세서는 고성능 프로세서와 저전력 프로세서로 구성된 아키텍처로써, ARM사의 big.LITTLE을 대표적인 예로 들 수 있다. 일반적인 big.LITTLE은 고성능 CPU인 4개의 big과 저전력 CPU인 4개의 LITTLE로 구성된다. big은 높은 성능과 함께 많은 전력을 소비하는 반면, LITTLE은 제한적인 성능을 갖지만 낮은 전력을 소비하고 특히 LITTLE은 big과 동일한 성능을 갖는 구간에서도 적은 전력을 소비한다는 특징이 있다[1].

이러한 구조를 효율적으로 사용하기 위해서는 새로운 스케줄링 기법이 요구된다. 초기 모델인 삼성 엑시노스 5에서 사용되었던 클러스터 마이그레이션(Cluster migration)은 4개의 big과 4개의 LITTLE을 각각 클러스터로 구성하고, 태스크(task)들의 부하에 따라 한 클러스터만 사용할 수 있다. 따라서, 두 타입의 코어를 동시에 혼용할 수 없어, 스케줄링의 유연성과 효율성이 떨어진다는 한계점을 가진다. 이를 극복하기 위해, IKS (In-Kernel Switch)과 HMP (Heterogeneous Multi-Processing) 기법을 적용한 Linaro 커널이 공개되었다[2]. IKS 기법은 big과 LITTLE 하나씩을 가상 프로세서로 묶어 4개의 대형형 가상 프로세서로 구성하고, 각 가상 프로세서는 부하에 따라 big과 LITTLE 중 하나만 선택하여 사용한다. 따라서 8개의 코어 중 최대 4개만 사용할 수 있다는 단점이 있다. 반면에, HMP 기법은 타입 별 4개의 코어를 각각의 도메인으로 구성하고 8개의 코어를 독립적으로 사용하도록 구현된 스케줄러로, 모든 코어를 동시에 사용할 수 있어 최대 성능을 낼 수 있다. 하지만 더 많은 코어를 사용하면서 CPU 소비전력이 증가하게 된다.

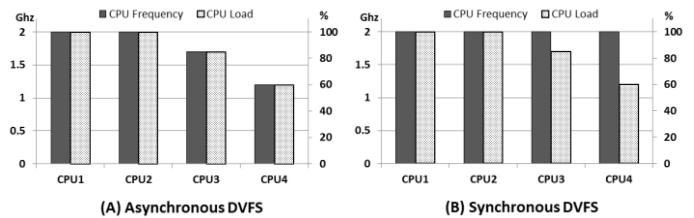


그림 1. 비동기식 DVFS와 동기식 DVFS

CPU의 소비전력은 스케줄링을 통한 각 코어의 부하를 기준으로 CPU 클럭을 조절하는 DVFS(dynamic voltage and frequency scaling)기법에 의해 관리된다. 그림 1과 같이 DVFS에는 동기식과 비동기식의 2가지 방식이 있는데, 비동기식 DVFS는 CPU 부하에 따른 클럭을 각각 다르게 조정한다. 반면, 동기식 DVFS는 도메인 단위로 CPU 클럭을 조절하는데, 제조 비용의 이유로 그림 1-(B)와 같이 동기식 DVFS를 사용하는 경우가 많다. 예를 들어 최근 보편적으로 사용되는 인터랙티브 거버너 (Interactive governor)를 기준으로 4개의 big 도메인 코어가 자신의 부하에 따라 각각 1.2, 1.4, 1.6, 2.0Ghz의 클럭을 요청했다면, 가장 높은 클럭을 요청한 CPU를 기준으로 big 도메인 전체를 2.0Ghz로 설정한다. LITTLE 도메인 또한 같은 방식으로 클럭을 조정한다. 다시 말해, 8개 프로세서들의 클럭은 big과 LITTLE의 도메인 단위로만 조절된다. 이러한 구조에서 CPU의 소비전력을 줄이기 위해서는 DVFS와 CPU 스케줄러가 모두 고려되어야 한다.

본 논문은 Linaro 커널 3.10의 HMP 스케줄러에 동적 임계치를 적용하고 새로운 프로세스 생성 기법을 개선하여, 동기식 DVFS와 스케줄러간 발생하는 전력소비의 비효율을 개선하며, 또한 유휴 CPU를 불필요하게 사용하여 발생하는 소비전력을 개선시키는 기법을 제안한다.

이 논문은 2010년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. 2010-0020724).

2. 관련 기술

2.1 Linaro HMP 스케줄러

Linaro HMP 스케줄러는 이기종 멀티프로세서의 사용을 지원하기 위해 기존 CFS(Completely Fair Scheduler)를 확장 구현하였다. HMP 스케줄러의 특징은 CPU 클럭을 고려한 태스크 부하 추적(power-aware task load tracking)과 big-LITTLE 도메인 간 태스크 이주(task migration) 기능이다.

CFS는 각 태스크 별로 실제 CPU 자원을 사용한 시간을 통해 부하(Task load)를 추적한다. 태스크 부하는 아래 식을 통해 0~1023의 값으로 환산되고, n_{period} 는 태스크의 총 생애시간, $n_{runnable}$ 은 생애 시간 중 실행 혹은 실행 대기상태 시간의 합을 나타낸다. 즉, 태스크가 생성된 후 현재까지의 시간 중 실제 CPU를 사용한 시간과 대기 큐에서 기다린 시간의 비중을 해당 태스크의 부하로 산출한다.

$$Taskload = 1024 \times \frac{T(n_{runnable})}{T(n_{period})}$$

추가로, HMP 스케줄러는 실행 시점의 CPU 클럭 정보를 태스크 부하에 반영하여, 낮은 클럭에서 실행된 태스크의 부하를 낮게 측정하는 기법을 적용하였다[3].

또한, 실시간으로 추적된 태스크 별 부하를 기준으로 big과 LITTLE 중 적합한 CPU를 할당하기 위해 태스크 이주 정책이 추가되었다. 태스크가 새로운 도메인의 CPU를 할당 받는 경우는 새로운 태스크 생성, 블록 후 깨어날 때, 그리고 로드밸런스 발생으로 나눌 수 있다. 태스크가 새로 생성되면 초기 부하는 최대(1023)로 설정되고 할당 받은 big에서 실행되면서 그 부하를 추적한다. 초기 부하추적 시간은 약 50ms를 최소한으로 보장하고 있는데, 이는 최대 부하가 down-threshold(512)까지 도달 하는데 걸리는 최소 시간이다. 그 후 태스크의 실행 도중에, 블록 후 깨어나는 시점 혹은 로드밸런스 호출 시점에 해당 태스크 부하가 임계치에 도달하면 도메인간 이주가 발생한다. 즉, big에서 실행되는 낮은 부하의 태스크(down threshold 이하)와 LITTLE에서 실행되는 높은 부하의 태스크(up threshold 이상)가 이주의 대상이 된다. 따라서 임계치는 태스크가 자신의 부하에 따라 big과 LITTLE 중 어떤 CPU에서 실행될 지 비교하는 기준이고, 전력-성능간 trade-off의 기준이 된다. 예를 들어, 임계치를 낮게 설정하면 더 낮은 부하의 태스크들이 big을 활용하게 되어 높은 성능을 낼 수 있지만, 많은 전력을 소비하게 된다. HMP 스케줄러에서 임계치는 LITTLE→big(up threshold)은 700, big→LITTLE(down threshold)는 512로 설정되어 있다.

3. 본 론

3.1 동기식 DVFS의 문제점

동기식 DVFS는 도메인 내의 부하 불균형이 커지면 소비 전력이 증가하는 단점이 있고, 이런 현상은 많은 전력을 소비하는 big 도메인에서 발생한다. 즉, 그림 1-(B)의 부하 불균형은 부하가 작은 CPU3과 4에서 불필요한 전력소비를 유발하는 문제점이 있다[4]. 대형형 멀티 프로세서와는 달리 이기종 멀티 프로세서에서는 저전력 CPU를 활용해 고성능 CPU의 부하 불균형 상황을 줄여 전력 효율을 높일 수 있다.

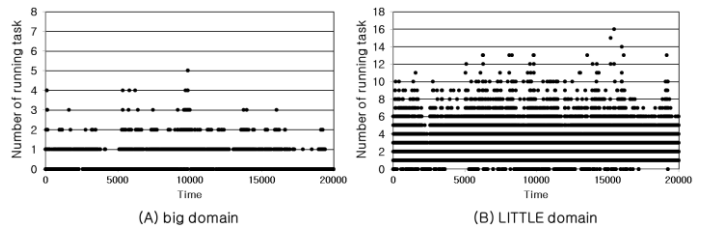


그림 2. 도메인 별 실행 태스크 수

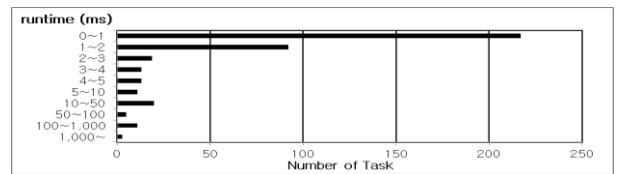


그림 3. 새로 생성된 태스크의 runtime

3.2 HMP 스케줄러의 문제점

HMP 스케줄러의 알고리즘 관찰을 위해, 음악 재생, 사진, 게임(Angry bird), 웹 브라우저(gmail, facebook, cnn, youtube)을 실행하면서 태스크들의 변화를 관찰하였다.

그림 2는 각 도메인에서 실행 중인 태스크 수를 나타낸다. 대다수의 낮은 부하 태스크들은 LITTLE 도메인에서 실행되고, 소수(1~2)의 태스크만 big 도메인을 사용하고 나머지 big CPU는 유휴 상태로 관리된다. 따라서 태스크가 새로운 도메인으로 이주 시 가장 idle한 CPU로 할당되는 것을 고려할 때, big으로의 태스크 할당은 매번 저전력 상태의 유휴 CPU를 깨워 소비전력을 높이는 원인이 된다. 하지만 유휴 상태의 big을 높은 부하의 태스크에게 할당하는 다른 이주 시점과는 달리, 부하 추적이 되지 않은 새로 생성된 태스크를 big으로 할당하는 것은 3.1에서 지적한 big 도메인 내 부하 불균형을 유발하는 원인이 되고, 매번 유휴 CPU를 깨워 소비전력을 증가시킨다. 그림 3은 생성된 태스크가 처음 할당 받은 big의 CPU 사용 시간을 측정하여 분류하였고, 이는 생성된 시간부터 big CPU 반납시(LITTLE로 이주 혹은 종료)까지 측정하였다. 최소 부하 추적시간인 50ms이상 CPU 자원을 사용한 태스크는 약 9.6%밖에 되지 않았다. 따라서, 대다수의 태스크는 CPU를 많이 사용하지 않아 LITTLE로 이주되는 낮은 부하의 태스크이거나, 총 생애시간 중에 1~2ms의 짧은 시간만 CPU를 사용하고 종료되는 경량 태스크로 측정되었다.

즉, 현재 Linaro HMP 스케줄러는 부하가 추적되지 않은 새로운 태스크의 성능을 위해 유휴 CPU를 깨워 big으로 할당하고 있지만, 대다수의 태스크가 처음 할당 받은 big을 거의 사용하지 않고 있어 비효율적인 전력 소비를 야기한다.

3.3 동적 임계치를 통한 도메인 불균형 해소 기법

동적 임계치 기법은 big의 클럭이 높아짐에 따라 이주 임계치를 동적으로 증가시켜 big 도메인의 부하 불균형을 줄이는 기법이다. 예를 들어, big 도메인 최대 클럭(2.0Ghz)시 기존 700/512(up/down threshold)를 최대 900/712까지 동적으로 올리도록 구현하였다. 이를 통해, 그림 1과 같이 도메인 내 부하 불균형을 초래하는 낮은 부하의 태스크(CPU 4)를 LITTLE로 이주시키고, CPU4를 저전력 유휴상태로 전환하여 전력 소모를 줄일 수 있다. 추가로, big으로 이주 될 LITTLE의

낮은 부하 태스크의 이주를 지연 시켜 부하 불균형을 유발하는 big의 사용을 막을 수 있다. 다시 말해, 최대 클럭 상황에서 big 도메인의 불균형을 야기하는, big의 부하 512에서 712 사이인 태스크와 LITTLE의 부하 700에서 900사이인 태스크들이 동적 임계치 기법의 적용 대상이 된다.

반면, big의 낮은 클럭을 요구하는 태스크를 LITTLE의 높은 클럭에서 실행시킴으로 인해 성능의 저하가 생길 수 있다. 하지만 big.LITTLE의 성능-소비전력 비교[2]에서 볼 수 있듯이, LITTLE의 높은 클럭은 big의 낮은 클럭에 비해 많은 성능 차이를 보이지 않고 심지어 낮은 전력으로 같은 성능을 내는 구간도 포함하고 있다. 이를 통해 동적 임계치 기법은 최소한의 성능 감소로 더 큰 전력소비를 막을 수 있다.

3.4 새로 생성된 태스크의 CPU 할당 기법 수정

3.3장에서 지적한 바와 같이, 새로 생성된 태스크의 CPU 할당 기법은 매번 유휴 상태의 프로세서를 깨우기 때문에 전력 측면에서 비효율적이다. 오히려 big으로 할당되던 기법을 LITTLE로 할당하도록 수정하는 것이 유휴 상태인 big의 사용을 줄여서 소비전력의 효율을 더 높일 수 있다.

반면 큰 부하 태스크를 생성하는 경우 성능이 감소할 수 있지만, 실제 사용 시나리오에서 big의 높은 성능을 이용하는 태스크는 약 9%의 소수 태스크이고 초기 태스크 부하를 0이 아닌 일정 값을 설정하여 50ms내에 부하 추적을 끝내고 big으로 이주할 수 있다. 따라서, LITTLE로의 할당은 실험에서도 큰 성능저하를 유발하지 않았다. 그러나 이를 효율적으로 하기 위해서는, 큰 부하를 갖는 태스크를 따로 구분하여 생성시 big을 할당하는 기법이 필요하고 이에 관한 기법은 향후 연구로 진행하려고 한다.

4. 실험

4.1 실험 환경

본 논문에서는 ODROID-XU3(exynos 5422 big.LITTLE)의 안드로이드 4.2.2 버전에서 Linaro 커널 3.10을 적용하였다. 추가로, 새로 생성된 태스크의 CPU 할당 기법을 수정하고, 동적 임계치 기법을 추가하였다. 소비전력 측정은 on-board 소비전력 분석 툴을 사용하여 100ms마다 각 도메인의 CPU 전력만을 측정하였고, 실험은 웹 벤치마크인 Vellamo와 Antutu HTML5를 사용하였다. 위 벤치마크들은 최대 부하의 태스크를 생성하여 CPU 최고성능을 측정하는 CPU 벤치마크들과는 달리, 웹 페이지 로딩이나 자바스크립트 연산과 같이 실제 사용 시나리오에 더 근접한 벤치마크이다.

4.2 소비전력과 성능 평가

그림 5은 벤치마크를 실행하는 동안의 소비전력을 나타낸다. Vellamo의 경우, big 도메인의 소비전력이 14.2% 감소하였지만, big의 부하 불균형 상황에서 LITTLE을 더 활용하였기 때문에 LITTLE 도메인의 소비전력은 5% 증가하였다. 하지만 LITTLE은 저전력 CPU이기 때문에 전체 증가 양은 크지 않아 총 30.4J (10.7%)의 전력 소비를 줄일 수 있었다. 마찬가지로 Antutu HTML5의 경우 big 도메인의 전력은 11.4% 개선된 반면, LITTLE의 소비전력이 6.7% 늘어 총 11.88J (7.78%)의 소비전력을 개선하였다.

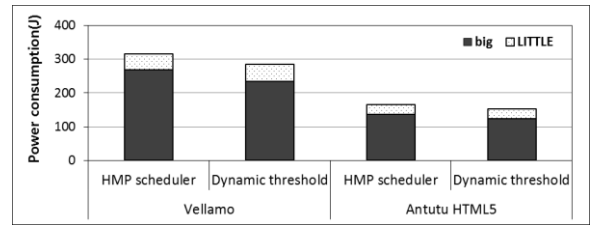


그림 5. 벤치마크 실행간 소비전력

표 1. 성과와 소비전력의 평가

	Vellamo		Antutu HTML5	
	Performance Score	Normalized EDP	Performance Score	Normalized EDP
HMP scheduler	2966	1.0000	12662	1.0000
Dynamic threshold	2918	0.9245	12247	0.9231

표 1은 벤치마크 점수(성능)를 토대로 에너지 지연 시간의 곱(Energy-Delay Product)을 통해 성과와 소비전력을 평가하였다. EDP는 소비전력/성능으로 계산하였고, 수정된 기법을 통해 전력 효율이 더 개선되었음을 확인할 수 있다.

4.3 big 도메인의 부하 불균형

소비전력 증가를 유발하는 big 도메인의 부하 불균형을 개선하기 위해 새로운 fork기법과 동적 임계치 기법을 적용하였고, 그림 6은 Vellamo 실행 간 big 도메인 CPU들에 부하간 표준편차의 CDF를 나타낸다. 큰 표준편차는 도메인 내의 큰 부하 불균형 상태를 의미하고, big의 사용빈도가 적기 때문에, 불균형이 0인 경우가 많다. 새 기법 적용을 통해 big 도메인 부하 불균형이 개선됨을 볼 수 있다.

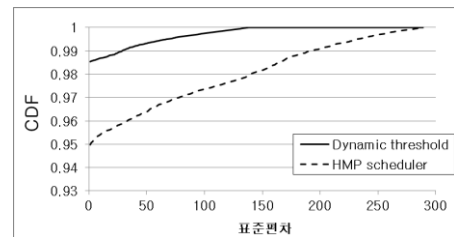


그림 6. big domain내 CPU의 부하간 표준편차(CDF)

5. 결론

본 논문에서는 big.LITTLE 구조에서 Linaro HMP 스케줄러의 전력 비효율을 개선하기 위해, 새로 생성된 태스크의 CPU 할당 기법을 수정하고 동적 임계치 기법을 제안하였다. 실험 결과, HMP 스케줄러와 비교해서, 비슷한 성능을 유지하면서 최대 10.7%의 전력 소비를 감소할 수 있었다.

참고문헌

[1] Peter Greenhalgh, "big.LITTLE processing with ARM Cortex-A15 and Cortex-A7," White paper (2011)
 [2] Linaro Stable Kernel(LSK), <https://wiki.linaro.org/LSK>
 [3] Yu, Kisoo, et al., "Power-aware task scheduling for big. LITTLE mobile processor," *SoC Design Conference (ISOC), 2013 International*. IEEE, 2013.
 [4] Cho, Hyun-Duk, et al. "Benefits of the big. LITTLE Architecture," *EETimes, Feb(2012)*.