누전 에너지 관리를 고려한 공유 캐시 파티션 정책

*강희준, 김지홍
서울대학교 전기컴퓨터공학부
e-mail : *kanghj@davinci.snu.ac.kr, jihong@davinci.snu.ac.kr*

Leakage-Aware Shared Cache Partitioning

*Hee-Joon Kang, Jihong Kim
School of Computer Science and Engineering
Seoul National University

Abstract

Cache leakage management techniques based on cache decay were devised on a single processor environment. But they can be applied to multi-processor environment such as CMP.

However, their leakage saving will be decrease on multi-processor environment because cache interferences in shared cache disturb sleeping cache lines.

We eliminated cache interferences via leakage-aware cache partitioning and saved leakage energy up to 8% over cache decay technique without cache partitioning.

I. 서론

최근의 CPU 디자인은 CMP/MPSoC와 같은 단일 칩 멀티프로세서 환경으로 변화했다. 기존의 캐시 누전 관리 기법들은 단일 프로세서 환경에서 제안되었기 때문에, 환경의 변화에 맞추어 기존 기법들이 고려하지 못한 점을 찾아 수정해야 할 필요가 있다.

CMP의 2차 공유 캐시는 단일 프로세서의 캐시와 다르게 여러 프로세서에서 발생하는 캐시 간섭 때문에, 기존의 cache decay[1] 기반 캐시 라인 턴오프 기법을 통한 누전 에너지 절감을 방해받는다.

이 연구는 누전 에너지 소비를 고려한 캐시 파티션을 통해서 캐시 간섭을 제거했고, 이를 통해 기존의 캐시 라인 턴오프 기법만을 사용했을 때보다 더 많은 누전 에너지 소비 감소를 얻었다.

본 연구에선 캐시 라인 턴오프 기법으로 cache decay 기법을 변형하여 사용했으며, 파티션 매커니즘으로 column caching[2] 기법을 사용했다.

본 논문은 예측 모델과 캐시 파티션 알고리즘에 초점을 맞추어 기술한다.

이 연구를 위해 연구 장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사드립니다. 이 논문은 2007 년도 두뇌한국21 사업과 2007년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구입니다(No. R0A-2007-000-20116-0).

II. 본론

2.1 캐시 히트 / 누전 감소 예측 모델

캐시 히트 수와 누전 감소량 예측은 회귀분석 모델을 통해 이루어진다. 일정 시간 간격 동안 모니터한 데이터에 근거해서 회귀 분석을 수행한다.

[3]의 marginal gain counter와 유사한 모니터 하드웨어를 이용해서 프로세스별 직전 주기의 전체 캐시 히트 및 MRU 블록에서의 캐시 히트를 측정한다. 이 데이터를 이용해서 지수 함수 형태의 LRU 위치에 따

른 예상 캐시 히트 수 회귀 분석 모델을 만든다. 이 모델을 통해서 할당한 공간에 따라서 예상되는 캐시 히트 수를 알 수 있다.

캐시 누전 발생량을 수행 중에 측정하기가 어렵기 때문에 cache decay를 통한 캐시 누전 감소량은 각 캐시 라인이 sleep 상태에 머물러 있는 시간의 총합에 비례한다고 가정하고, 직전 주기에 캐시 라인들이 sleep 상태에서 머물러 있는 시간을 측정한다. 이렇게 얻어낸 데이터를 선형 회귀 분석 모델에 적용해서 프로세스별 공간 할당에 따른 예상 캐시 라인 sleep 시간을 얻을 수 있다.

2.2 캐시 파티션 알고리즘

```
proc repartition(num_way,sleep_weight)
  max_value  := 0
  allocation := (0,0)
  hit_weight := 1-sleep_weight
  for i=1 to num_way-1
    hit := HIT(p0,i)+HIT(p1,num_way-i)
    sleep := SLEEP(p0,i)+SLEEP(p1,num_way-i)
    t := hit_weight*hit + sleep_weight*sleep
    if  t > max_value
      max_value  := t
      allocation := (i, num_way-i)
  return allocation
```

그림 1. 파티션 알고리즘

매 주기가 끝날 때마다 예측 모델을 이용해서 다음 주기의 공간 할당량을 새로 계산한다. 캐시 히트 수와 캐시 블록 sleep 시간을 모두 고려해서 성능과 에너지 양 쪽에 균형 있는 결과를 얻을 수 있는 할당량을 찾는다.

Ⅲ. 실험 환경 및 결과

실험은 SimpleScalar3.0[4] 기반의 2 개의 프로세서 코어를 갖는 CMP 시뮬레이터에서 수행하였다. 1M, 16개의 way를 갖는 L2 공유 캐시에서 실험을 했다.

Spec cpu2000 integer 벤치마크[5]에서 임의로 2개씩 프로그램을 뽑아서, 한 개의 프로세서에서 한 개의 프로그램이 수행되도록 설정했다.

그림 2는 cache decay만을 적용했을 때의 누전 에너지 소비량과, 제안한 파티션 기법과 cache decay 기법을 함께 적용했을 때의 누전 에너지 소비량을 어떤 누전 관리 기법도 적용하지 않았을 때의 누전 에너지 소비량에 대해서 정규화한 결과를 나타낸다. 제안한 파

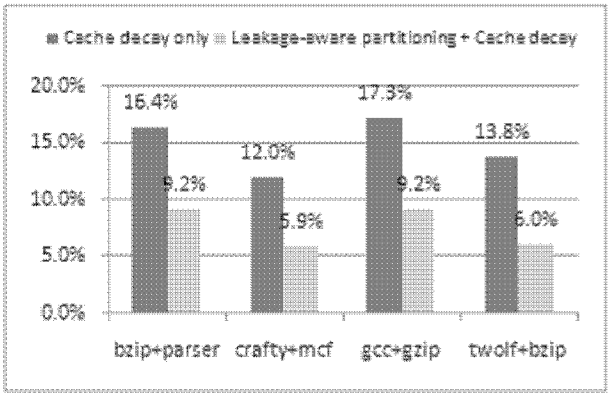

그림 2. 파티션 적용에 따른 누전 에너지 감소

티션 기법을 적용하였을 때 누전 에너지 소비가 6~8% 감소한 것을 확인할 수 있다. 캐시 파티션에 의한 캐시 미스 증가는 거의 cache decay의 부하에 가려지기 때문에 파티션 자체의 성능 부하는 크지 않다. 자세한 캐시 미스 변화는 지면 관계로 생략한다.

Ⅳ. 결론 및 향후 연구 방향

본 논문에서는 캐시 누전 감소와 전체 캐시 히트를 함께 고려한 캐시 파티션 기법을 제안했다. 캐시 파티션을 통해 공유 캐시에서 cache decay 기법을 통해서 얻을 수 있는 캐시 누전 감소의 효과를 증대 시킬 수 있음을 실험을 통해서 보였다.

이번 연구에선 수행 중인 프로세스의 특성에 관계없이, 동일한 decay timeout을 적용했지만, 앞으로는 수행 중인 프로세스의 특성을 고려하여 decay timeout을 조절함으로써, 캐시 미스를 크게 늘리지 않고도 더욱 많은 누전 에너지를 줄이는 방법을 연구할 예정이다.

참고문헌

[1] Kaxiras 외, “Cache decay: exploiting generational behavior to reduce cache leakage power”, Proc. of ISCA, pp. 240-251, 2001.

[2] Chiou 외, “Application-specific memory management for embedded systems using software-controlled caches”, Proc. of DAC, pp.416-419, 2000.

[3] Suh 외, “Dynamic partitioning of shared cache memory”, Journal of Supercomputing, vol.28, no.1, pp7-26, 2004.

[4] www.simplescalar.com

[5] www.spec.org