공유 말단 캐시에서의 간섭의 영향을 고려한 멀티코어 프로세서를 위한 가상 머신 스케줄링

김신규, 최찬호, 엄현상, 염헌영
서울대학교 컴퓨터공학부
e-mail:{sgkim, chchoi, hseom, yeom}@dcslab.snu.ac.kr

Virtual Machine Scheduling for Multicores Considering Effects of Shared On-chip Last Level Cache Interference

Shin-gyu Kim, Chanho Choi, Hyeonsang Eom, Heon Y. Yeom
School of Computer Science and Engineering, Seoul National University

요 약

클라우드 컴퓨팅 서비스 시장이 성장하면서, 서비스 제공자들은 전력 사용량 감소와 서비스 수준을 보장하는 등의 여러 가지 문제와 맞딱드리게 되었다. 이런 문제에 대한 원인 중 하나는 자원 효율성을 높이기 위해 도입한 가상머신 기반의 서버 통합 정책이다. 현재의 가상머신 기술들은 아직까지 완벽한 격리수준을 제공하지 못하기 때문에, 같은 노드에 배치된 가상머신들은 자원을 공유하면서 서로 간에 간섭을 일으키게 된다. 본 연구에서는 가상머신끼리 공유하는 자원 중 프로세서의 말단 캐시 (Last-level Cache, LLC) 에서의 간섭을 최대한 줄여서 성능을 극대화하기 위한 방법을 제안한다.

1. 서론

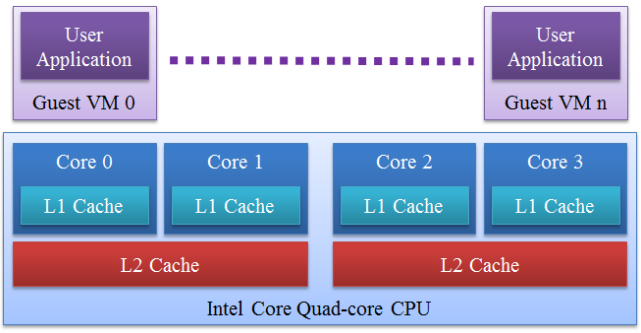
최근 멀티코어 프로세서는 개인용 컴퓨터나 서버 뿐만 아니라 스마트폰이나 태블릿 컴퓨터에도 쓰일 정도로 널리 사용되고 있다. 현재 100개의 코어를 갖는 프로세서도 시장에서 판매되고 있을 뿐만 아니라 향후 그 이상의 코어를 갖는 프로세서도 곧 출시가 예정되어 있다. [1, 2] 클라우드 컴퓨팅 서비스에서는 자원의 효율성을 극대화하는 것이 매우 중요하기 때문에 멀티코어 프로세서와 가상 머신 기술을 이용하여 여러 개의 서버를 하나의 노드에 배치하는 효율적인 서버 통합 정책 (Server Consolidation Policy) 이 매우 중요하다.

클라우드 컴퓨팅 시장이 성장하면서, 서비스 제공자들은 기존에는 중요하게 고려하지 않았던 새로운 문제들을 해결해야한다는 것을 알게 되었다. 전력 소모량을 줄여서 유지비용을 최대한 낮춰야 하며, 사용자가 지불한 요금만큼 성능을 보장해주어야 한다. 이런 문제의 원인들 중 하나는 앞서 언급한 서버 통합 정책이다. 현재의 가상머신 기술들은 가상머신 간에 완벽한 격리수준을 제공하지 못하기 때문에 가상머신들이 자원을 공유하는 상황에서는 가상머신들 간에 성능 간섭이 발생하게 된다. 이로 인해 다수의 사용자들이 예상했던 성능의 서비스를 이용하지 못하게 될 수 있기에, 서비스 제공업자는 추가적인 자원을 투입할 수밖에 없었다. 이로 인해 전력 사용량은 늘어나고, 그로 인한 유지비용의 증가는 서비스 제공업체에게는 큰 부담이 되었다.

본 연구에서는 프로세서의 말단 캐시의 공유로부터 발생하는 성능 간섭을 최소화하여 시스템 전체의 성능을 극대화하는 방법에 대하여 제안한다. 사용자의 각기 다른 작업들은 말단 캐시의 사용량이 다를 수 있는데, 이를 프로세서에서 제공하는 PMU (Performance Monitoring Unit) 을 사용하여 실시간으로 알아내고, 이를 이용하여 가상 머신을 재배치하여 전체적인 성능을 높일 수 있도록 하였다. 본 연구에서 사용한 가상머신 기술은 KVM에 기반하고 있다. [3]

2. 말단 캐시의 공유가 성능에 미치는 영향

그림 1은 인텔의 Nahelem 프로세서의 캐시 구조를 보여주고 있다. 크게 세 단계의 캐시로 구성되어 있으며, L1 캐시와 L2 캐시는 각 코어별로 따로 갖고 있으며, L3 캐시는 모든 코어가 공유하는 구조로 되어 있다. 따라서 L1



(그림 1) 인텔 Nehalem 프로세서의 캐시 구조

캐시와 L2 캐시에서는 성능 간섭이 발생하지 않고, 오직 L3 캐시에서만 성능 간섭이 발생하게 된다. 이 상황에서 성능에 대한 영향은 LLC miss의 증가정도로부터 직접 알 수 있으나, 그보다는 LLC 접근 비율 (LLC reference ratio) 이 더욱 좋은 특성을 갖고 있다는 것을 본 연구팀의 과거 연구 결과에서 이미 밝힌 바 있으며, LLC 접근 비율은 다음과 같이 정의된다. [4]

$$R_{LLCref} = \frac{number\ of\ LLC\ references}{number\ of\ instructions}$$

3. 가상 머신 스케줄링 기법

본 절에서는 앞서 소개한 LLC 접근 비율을 이용한 가상머신 스케줄링 방법을 제안한다. 먼저 본 연구에서 목표로 하는 가상머신의 성능이 극대화되는 경우에 대한 정의부터 시작한다.

[정의 1] 최대 성능을 내는 가상머신 배치에서는 모든 가상머신의 평균 성능 하락 비율이 최소가 된다.

<표 1> LLC 접근 비율 예제

	LLC 접근 비율 (%)
VM1	1.00
VM2	3.00
VM3	8.00

예를 들어 현재 두 개의 노드가 존재하고, 표 1과 같은 LLC 접근 비율을 갖는 가상머신이 세 개 존재할 때의 최대 성능을 내는 가상머신 배치를 알아보겠다. 결론부터 말하자면, VM1과 VM2를 함께 배치하고, VM3을 따로 배치한 경우에 가장 높은 성능을 얻을 수 있다. 이런 배치를 LLC 접근 비율을 이용해서 달성하기 위한 알고리즘은 다음과 같다.

먼저, 모든 가상머신들을 각자의 LLC 접근 비율에 대한 내림차순으로 큰 것부터 작은 순으로 정렬한다. 다음에는 각 노드에 가장 큰 LLC 접근 비율을 갖는 가상머신을 하나씩 배치시킨다. 이후에는 앞서 정렬된 가상머신 리스트로부터 가장 큰 LLC 접근 비율을 가진 가상머신을 LLC 접근 비율의 합이 가장 작은 노드에 배치시킨다. 이 과정을 모든 가상머신에 대해서 반복하면 앞서 언급한 최대 성능을 내는 가상머신 배치에 이를 수 있게 된다. 이에 대한 증명은 다음과 같다.

[증명 1] m개의 가상머신과 n개의 노드가 있고, 각 노드에는 하나의 프로세서가 있으며 해당 프로세서는 모든 코어가 공유하는 하나의 LLC가 존재한다. 각 코어는 오직 하나의 가상머신에만 할당될 수 있으며, 가상머신은 역시 오직 하나의 코어만 사용한다. 하나의 프로세서에는 l개의 코어가 있으며, 하나의 프로세서에는 최대 l개의 가상머신이 배치될 수 있다. S_i를 프로세서 i에 배치된 가상머신들의 LLC 접근 비율의 합이라고 하고, ($0 \le i \le n$) a_t를 t 라운드에서의 모든 S_i의 평균이라고 하자. 따라서 알고리즘의 목적은 모든 라운드에서 $C_t = \sum_{i=0}^{n}(S_i - a_t)^2$ 를 최소화하는 것이 된다. ($0 \le t \le m$)

초기 조건은 모든 프로세서가 하나의 가상머신을 갖고 있는 상황이며, 모든 S_i는 0보다 크다. $t = k$ 인 경우 앞의 조건은 만족한다고 가정한다. 이제 $t = k + 1$ 인 경우에도 조건을 만족하는지 알아보자. 이 경우 선택된 가상머신의 LLC 접근 비율을 x 라고 한다면,

$$a_{t+1} = a_t + \frac{x}{n}$$

이 된다. 이 경우 해당 가상머신이 프로세서 j에 배치된다면, 아래과 같은 관계를 얻을 수 있다.

$$\begin{aligned} C_{t+1} &= \left\{S_j + x - \left(a_t + \frac{x}{n}\right)\right\}^2 + \sum_{i \neq j}\left\{S_i - \left(a_t + \frac{x}{n}\right)\right\}^2 \\ &= \left\{(S_j - a_t) + \left(x + \frac{x}{n}\right)\right\}^2 + \sum_{i \neq j}\left\{(S_i - a_t) + \frac{x}{n}\right\}^2 \\ &= 2x(S_j - a_t) + (\ldots) \end{aligned}$$

결과적으로, 위의 수식의 제일 첫 항이 최소화되어야 C_{t+1}이 최소화될 수 있다. 이를 위해서는 가장 작은 S_j를 갖는 프로세서를 선택하고, 가장 큰 LLC 접근 비율을 갖는 가상머신을 선택해야한다. 따라서 매 라운드마다 이와 같은 선택을 하면 C_t는 항상 최소로 유지될 것이며, 이는 본 연구에서 제안한 알고리즘이 최선의 선택을 할 수 있도록 한다는 것을 증명한다.

4. 성능평가

본 절에서는 본 연구에서 제안된 가상머신 스케줄링 기법의 성능을 평가해본다. 실험은 인텔 Xeon E7-4807 Hexa-core 프로세서와 512GByte의 메모리를 가진 서버를 이용하였으며, 운영체제는 Ubuntu 11.10 (Linux kerenl 3.0.0)을 이용하였다. 가상머신은 KVM환경에서 각기 2Gbyte의 메모리와 8GB의 디스크 공간이 할당되었다. 아래의 표 2는 SPECcpu 2006 벤치마크 중 네 가지 응용프로그램에 대한 LLC 접근 비율을 나타내고, 표 3은 이에 대한 결과를 나타낸다. 스케줄러를 사용한 경우와 사용하지 않은 경우에 대해서 각각 두 번씩 수행하였다. 표 4에서 볼 수 있듯이, 본 연구에서 제안된 알고리즘을 사용한 경우 사용하지 않았을 경우보다 항상 더 낮은 성능 하락

<표 2> SPECcpu 2006 프로그램의 LLC 접근 비율

응용프로그램	LLC 접근 비율 (%)
omnetpp	3.01
hmmer	0.05
soplex	2.06
gcc	2.12

<표 3> 배치에 따른 SPECcpu 2006 응용프로그램의 성능 하락 비율

	평균 성능 하락 비율 (%)								
	VM1 (omnetpp)	VM2 (omnetpp)	VM3 (soplex)	VM4 (soplex)	VM5 (hmmer)	VM6 (hmmer)	VM7 (gcc)	VM8 (gcc)	평균 (%)
스케줄러 사용 하지 않은 경우	32.13	23.57	23.48	16.46	0.13	0.00	16.98	18.43	16.40
	20.07	35.64	25.32	29.13	0.00	2.08	16.77	16.98	18.25
스케줄러 사용 한 경우	4.52	23.91	24.03	27.37	0.13	1.12	18.94	19.14	14.89
	4.39	19.98	23.14	17.17	0.25	0.00	19.24	18.94	14.14

비율을 보여주었으며, 그 편차 또한 크지 않음을 확인할 수 있다.

5. 결론

본 연구에서는 멀티코어 프로세서의 말단 캐시를 공유하는 클라우드 환경에서 최고의 성능을 얻기 위한 가상머신 배치 기법을 제안하였다. 해당 배치 기법은 LLC 접근 비율을 이용하여 실시간으로 가상머신을 최고의 성능을 내는 배치를 알아낼 수 있다. 이 기법을 이용하면 적은 수의 물리적 노드를 갖고도 성능을 높일 수 있기에 전체적인 전력사용량을 감소시킬 수 있으며, 동시에 서비스 수준을 최고로 유지할 수 있게 된다.

감사의 글

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임. (No. 2011-0020521) 이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사드립니다.

참고문헌

[1] Intel Corporation. Single-chip Cloud Computer. http://techresearch.intel.com/ProjectDetails.aspx?Id=1

[2] Tilera Corporation. TILE-Gx Processor Family. http://www.tilera.com/products/processors/TILE-Gx_Family.

[3] KVM (Kernel-based Virtual Machine). http://www.linux-kvm.org/

[4] Seungmin Kang, Shin-gyu Kim, Hyeonsang Eom, Heon Y. Yeom, “Towards Workload-aware Virtual Machine Consolidation on Cloud Platforms”, ACM ICUIMC 2012, Kuala Lumpur, Malaysia