WLRU: 분산 공유 메모리 구조에 적합한 원격 캐시 관리 정책

서효중, 이병호°
가톨릭대학교 컴퓨터정보공학부
{hjsuh, canine°}@catholic.ac.kr

WLRU: Remote Cache Management Policy for Distributed Shared Memory Architectures

Hyo-Joong Suh, Byong-Ho Lee°
School of Computer Science and Information Engineering, The Catholic University of Korea

요 약

분산 메모리에 기반한 다중 프로세서 시스템은 기존의 중앙 집중형 메모리 구조의 단점인 메모리 접근의 병목현상을 극복하고 프로세서와 메모리의 부가에 따라 메모리 대역폭을 확장시킬 수 있는 구조로써 최근의 다중 프로세서 시스템 구조의 주류로 대두되고 있다. 다중 프로세서 시스템의 성능은 메모리 접근 지연에 의하여 제한 받고 있는데 이러한 이유는 프로세서의 동작 주파수 속도에 비하여 메모리의 접근 지연이 수십 배 이상이 되기 때문이다. 특히 분산 메모리 다중 프로세서 시스템에 있어서 메모리 접근은 지역 메모리 접근과 원격 메모리 접근의 두 가지 유형으로 나눌 수 있는데 이 중 원격 메모리 접근 지연은 시스템의 상호 접속망 구조에 따라 지역 메모리 접근 지연에 비하여 수 배 내지 수십 배에 이르고 있다.

본 논문에서는 분산 메모리 다중 프로세서 시스템에서 상호 접속 망의 구조에 따라 원격 메모리 접근 간에도 시간 지연의 차이가 있음에 착안하여 원격 메모리 접근 시간 지연에 따른 최적화 된 원격 캐시 관리 정책을 제시하며 각 상호 접속 망의 구조에 따라 이러한 캐시 관리 정책에 의한 성능 향상의 정도를 측정한다.

1. 서 론

최근 다양한 형태의 병렬 처리 시스템이 지속적으로 소개되고 있음에도 불구하고 병렬 처리 시스템은 컴퓨터 산업의 주류로 자리 잡고 있지 못하고 있다. 이러한 이유 중, 가장 큰 원인은 프로세서, 메모리 등의 추가에 미치지 못하는 시스템 성능 개선에 있으며, 프로세서와 메모리간의 통신 지연이 성능 개선의 한계로 제시되고 있다. 특히 분산 메모리 다중 프로세서 시스템의 경우에 노드 내(intra-node)의 명령에 의한 지연에 비하여 노드 간(inter-node)의 명령에 의한 지연이 수십 배 이상의 시간을 소모함으로써 노드간의 명령에 의한 지연 시간을 최소화하는 것이 성능 개선의 일차적인 목표로 대두되었다 [1].

본 논문에서는 이러한 분산 다중 프로세서 시스템의 성능 개선을 위해 기존의 LRU 정책을 개선하여 임의의 상호 접속망 구조에 적합한 원격 캐시 라인 교체 정책을 제안한다. 제안하는 원격 캐시 라인 교체 정책은 메모리 참조의 지역성 정보와 각 노드간의 접근 경로 비용에 따라 분산 메모리 다중 프로세서 시스템에 최적화 된 것이다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 분산 메모리 다중 프로세서 시스템의 원격 캐시 상에서 LRU 정책의 비적합성을 지적하며, 제 3 장에서는 제안하는 노드간의 접근 경로에 따른 원격 캐시의 라인 교체 정책에 대하여 설명한다. 제 4 장에서는 원격 메모리 접근에 대한 분석을 기술하였으며, 제 5 장에서 Working Set과의 연관관계를 기술하였다. 마지막으로 6 장에서 결론 및 차후 연구 과제를 제시하였다.

2. 분산 메모리 다중 프로세서 시스템의 원격 캐시

분산 메모리 다중 프로세서 시스템에서 캐시 라인의 교체 정책의 구현에 적합한 알고리즘으로는 최근 최소 사용(Least Recently Used, LRU) 알고리즘이 있다. 그러나 LRU 알고리즘은 모든 메모리가 일정한 시간에 접근될 수 있는 경우에 적합한 알고리즘이며 SCI [2], Myrinet [3]등을 이용한 분산 메모리 시스템과 같이 메모리 접근을 요청한 노드와 메모리 라인을 제공하는 노드의 접근 경로가 가변적일 경우 노드간의 접근 경로에 따른 지연 시간을 고려하지 않는 LRU 알고리즘은 적당하지 않다.

```
int     i, j, k; //all variables are allocated to the registers
word    A[4,3];

for (i=0; i<4; i+=1) {
        for (j=0; j<3; j+=1) {
                k = A[i,j];
        }
        // Check point of the remote cache.
}
for (i=0; i<4; i+=1) {
        for (j=0; j<3; j+=1) {
                k = A[3-i,j];
        }
        // Check point of the remote cache.
}
```

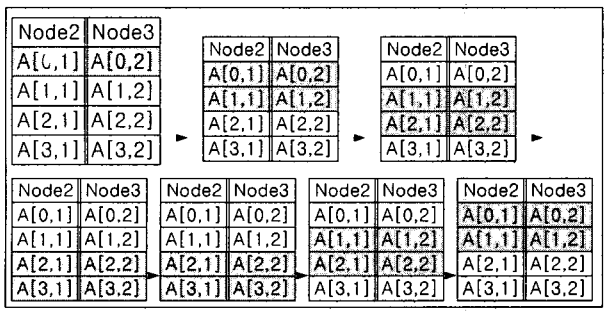
[그림 1] LRU 알고리즘이 적합하지 않은 예제 프로그램

[표 1] 그림 1에서 배열 A의 지역 메모리 할당 모습

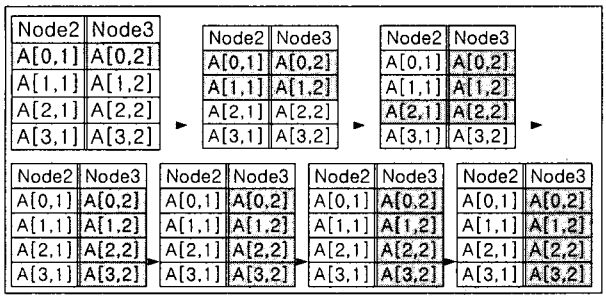
Node 1 의 지역 메모리	Node 2 의 지역 메모리	Node 3 의 지역 메모리
A[0,0]	A[0,1]	A[0,2]
A[1,0]	A[1,1]	A[1,2]
A[2,0]	A[2,1]	A[2,2]
A[3,0]	A[3,1]	A[3,2]

그림 1 은 LRU 알고리즘이 적합하지 않은 경우의 간단한 프로그램 예제 이고, 표 1 은 그림 1의 배열 A의 지역 메모리 할당 모습이다. 원격 캐시[4] 라인의 크기는 워드와 동일하며 full-associative 하고 캐시의 크기는 네 개의 라인이며 프로세서 캐시가 없고 데이터 접근만을 고려한 경우이다. 노드는 세 개를 가정하였으며 프로그램을 수행하는 노드를 노드 1로, 배열 A는 세 노드의 지역 메모리에 표 1 과 같이 할당되어 있다고 가정하였다.

일반적인 LRU 알고리즘을 사용할 경우, 프로그램의 캐시 확인 부분에서 원격 캐시에 캐싱되어 있는 메모리 라인은 그림 2에서 음영으로 표시된 형태로 변화된다.



[그림 2] 일반적인 LRU 알고리즘을 사용한 경우, 원격 캐시에 캐싱되어 있는 메모리 라인의 모습



[그림 3] 변경된 LRU 알고리즘을 사용한 경우, 원격 캐시에 캐싱되어 있는 메모리 라인의 모습

노드 내부의 지연을 1, 노드 1과 노드 2 사이의 접근 지연을 C라 하고 노드 1과 노드 3 사이의 접근 지연을 D라 하며 $D = C*2$의 관계를 가진다고 할 때 이 프로그램에서 총 원격 주소 영역에 대한 데이터 접근 횟수는 16 회이며 이 중 원격 캐시에서 적중된 횟수는 노드 2 주소 영역에 대하여 2 회 노드 3 주소 영역에 대해 2 회이므로 원격 주소 영역에 대한 총 접근 시간은 다음과 같다.

• 원격주소 영역에 대한 총 접근 시간

$$= 6C + 6D + 4 = 18C + 4$$

반면에 어떤 원격 캐시 라인 교체 알고리즘이 노드 2 주소 영역의 라인이 노드 3 주소 영역의 캐시 라인을 교체시키지 않도록 하며, 노드 3 주소 영역의 캐시 라인은 노드 2 영역의 캐시 라인을 우선 교체시키고 노드 2 영역의 캐시 라인이 존재하지 않을 경우만 노드 3 영역의 캐시 라인을 LRU에 따라 교체시킨다고 가정할 경우, 프로그램의 캐시 확인 부분에서 원격 캐시 에 있는 메모리 라인은 다음의 그림 3에서 음영으로 표시된 부분이 된다.

이 알고리즘의 경우 원격 주소 영역에 대한 접근 16 회 중 원격 캐시에 적중된 것은 마찬가지로 4 회이나 노드 2 주소 영역에는 캐시 적중이 되지 않으며, 노드 3 주소 영역에 대해서만 4 회 적중하므로 원격 주소 영역에 대한 총 접근 시간은 다음과 같다.

• 원격주소 영역에 대한 총 접근 시간

$$= 8C + 4D + 4 = 16C + 4$$

LRU 알고리즘과 변형된 알고리즘에서 원격 주소 접근 부분을 제외하고는 동일하므로 알고리즘에 의하여 2회의 노드 3으로의 메모리 접근이 노드 2로의 메모리 접근으로 전환되었음을 알 수 있다.

3. 분산 메모리 구조의 시스템에 적합한 원격 캐시 라인 교체 정책

일반적인 단일 프로세서 시스템의 경우 프로세서로부터 메모리까지의 경로가 정적으로 구성되어 있으므로 프로세서로부터의 메모리 접근 패턴만을 고려하여도 합리적인 결과를 얻을 수 있었다. 그러나 최근의 분산 메모리 구조의 다중 프로세서 시스템의 경우 캐시 실패 시 접근 경로가 메모리 주소 영역에 따라 달라지므로 [5] 메모리 접근 경로 비용에 따른 라인 교체 알고리즘의 채용은 분산 메모리 구조에 가장 합리적인 것이다.

본 논문에서 제안하는 라인 교체 정책은 LRU 정책에 접근 경로 비용에 따른 가중치를 둔 것으로 WLRU (Weighted LRU)정책이라 칭하며, 본 논문에서 WLRU 정책을 기술하기 위하여 접근 순서와 접근 비용, WLRU 값을 정의한다. 일반적인 n-way set associative 캐시에서 LRU 정책을 사용할 경우 set 안의 라인들이 접근된 순서를 유지하여야만 하는데 가장 최근에 접근된 라인은 1로, 가장 오래 전에 접근된 라인을 n으로, 접근 순차대로 큰 값을 갖도록 하고 이 값을 접근 순서라 한다. 접근 비용은 어떤 노드에서 가장 인접한 접근 경로를 가지

고 있는 노드의 지역 메모리를 접근하는 데 필요한 시간 비용을 기준 값 1로 보았을 때 다른 각 노드의 지역 메모리에 접근하는 데 필요한 시간 비용이다. WLRU 값은 각 캐시 라인에 대해 접근 비용에서 접근 순서를 뺀 값이다.

원격 캐시의 WLRU 정책은 다음과 같다.

• 캐싱되는 모든 라인은 접근 순서와 접근 비용을 유지한다. 접근 순서는 LRU의 경우와 같으며 접근 비용은 캐싱되어 있는 라인의 주소 영역에 따른 상수값을 가진다.

• set이 캐싱된 라인으로 차게 되면 차후에 메모리 라인이 캐싱될 경우, 교체 라인은 동일한 set내의 각각의 라인에 대해서 WLRU 값을 계산하여 가장 작은 값을 갖는 라인이 교체된다. 가장 작은 값을 갖는 라인이 복수개 존재할 경우 접근 비용이 작은 라인이 우선 교체된다.

4. 원격 메모리 접근 분석

균일하지 않은 원격 메모리 구조를 갖는 다중 프로세서 시스템 에서 지역 노드의 메모리 접근은 원격 캐시와 무관하다고 가정하고, 캐시 일관성 유지에 의한 영향을 무시하면 평균 원격 메모리 영역 접근 시간은 다음과 같다.

$$R = \frac{D_2}{R_2} + \frac{D_3}{R_3}$$

R : 평균 원격 메모리 영역 접근시간

$D2$: 요청노드의 원격캐시 메모리 접근시간의 총 합

$D3$: 원격노드 접근시간의 총합

$R2$: 요청노드의 원격캐시 메모리 접근횟수

$R3$: 원격노드 접근횟수

따라서, 평균 원격 메모리 영역에 대한 접근 시간은 위의 값을 최소화할 수 있는 경우에 가장 적어진다.

WLRU 알고리즘의 적용으로 꾀하는 것은 원격 메모리 영역에 대한 접근 시간의 평균적인 향상을 유도하는 것이므로 원격 캐시내의 교체될 캐시 라인을 선정하는 WLRU 값은 캐시 라인 내의 LRU 값에 따르는 교체 순서와 각 해당 주소에 해당되는 노드의 물리적 거리로서 설정될 수 있으며, 교체 순서를 N, 노드의 거리를 S로 정의할 때, 다음과 같다.

$$WLRU = kN + (1-k)S$$

상수 k값이 1일 경우, WLRU 는 LRU와 같아지며, k값이 커질수록 교체에 대해 해당 노드의 거리에 의한 비중이 커지게 된다. 실제의 시스템에서 상수 k의 선택은 노드간의 접근 지연 시간과, 원격 캐시에서의 접근 시간의 차이, 각 응용 프로그램의 원격 메모리 접근 형태 등에 의해 달라지게 된다.

5. Working Set 과 WLRU 정책

일반적으로 분산 공유 메모리 구조의 다중 프로세서 시스템에서 원격 캐시는 상대적으로 큰 지연을 발생시키는 원격 메모리 접근을 최소화하기 위해 상대적으로 큰 크기를 유지한다. 원격 캐시가 원격 메모리 접근을 충분히 대응할 수 있을 정도로 적은 Working Set을 구성하게 되는 경우, WLRU 정책에 의해 교체되는 라인이 오히려 원격 캐시에서 Working Set의 적절한 형성을 저해할 수 있다. 이러한 부작용을 방지하기 위해서 WLRU 값의 산정에 있어서 일정 시간 이상 접근되지 않은 라인에 대해서는 노드의 접근 거리에 상관없이 교체되도록 하는 보완이 필요할 수 있다. 즉, 일정 사이클 이내에 접근되지 않은 라인에 대해서는 낮은 WLRU 값을 갖도록 일정한 임계값을 설정할 필요가 있다.

6. 결론 및 차후 연구과제

본 논문은 분산 공유메모리 구조의 다중프로세서 시스템에서 노드간의 물리적 거리 비용의 차이에 따라 원격 캐시 교체의 기회를 다르게 한 WLRU 알고리즘을 제시함으로써 메모리 접근의 거리비용에 따른 적절한 원격 캐시 메모리 운용을 함으로써 보다 고성능의 평균 원격 메모리 영역 접근 시간을 꾀하고자 하였다. 제시한 방법의 실제 성능은 응용 프로그램의 메모리 할당 패턴과 원격 캐시의 크기 및 시간적 지역성의 정도에 따라 다양한 결과가 예상되며, 실 프로그램을 수행한 궤적을 기반으로 WLRU 값을 얻을 수 있는 적절한 계수를 얻을 수 있을 것이다. 차후 연구과제로써 이와 같은 여러 가지 주변 파라미터를 기반으로 한 적절한 계수의 도출 및 동적인 임계값 설정형태로의 알고리즘 확장이 연구 중이다.

7. 참고 문헌

[1] 서효중, "건너뛰 이중링크를 갖는 고확장성 CC-NUMA 시스템", 정보과학회 논문지, 제31권 제9호, pp487-494, 2004.

[2] IEEE Computer Society, IEEE Standard for Scalable Coherent Interface(SCI), Institute of Electrical and Electronics Engineers, Aug. 1993.

[3] http://www.myri.com/myrinet

[4] T. Lovett, R. Clapp, "STiNG : A CC-NUMA Computer System for the Commercial Marketplace", Proc. of the 23th International Symp. on Computer Architecture, pp. 308-317, May 1996.

[5] D. Lenoski, etal., "The Stanford Dash multiprocessor", Computer, Vol. 25 No.3, pp.63-79, Mar. 1992.