분산공유 메모리를 위한 성능비교 모델

임승범, 김재훈
아주대학교 정보통신전문대학원
email : prodigy@cesys.ajou.ac.kr , jaikim@madang.ajou.ac.kr

Performance Model for Distributed shared Memory

Seungbum Lim, Jai-Hoon Kim
College of Infomation and Communication
Ajou University

요 약

분산 공유 메모리(Distributed Shared Memory) 시스템은 사용자에게 간단한 공유메모리 개념을 제공하기 때문에 사용자는 노드 사이의 데이터 이동에 관여할 필요가 없다. DSM에서 일치서 유지를 위한 프로토콜을 선택하는 것은 통신 부하를 줄이는데 중요한 역할을 한다. 본 논문은 DSM 프로토콜을 효과적으로 선택하기 위한 새로운 성능평가 모델을 제시한다. 본 연구에서 제안하는 성능평가 모델을 사용함으로써 무효화방식(invalidate protocol), 갱신 방식(update protocol) 그리고 이주방식(migratory protocol)의 성능예측이 가능하다. 본 성능평가모델은 노드들 사이의 데이터 일치성(consistency) 유지를 위한 부담을 최소화하는 최적의 DSM 프로토콜을 결정하는데 사용된다

1. 서론

다중 프로세서 시스템내의 노드들간의 통신은 메시지 전달방식이나 공유메모리를 사용함으로써 행해진다. 메시지 전달방식과 대조적으로 공유 메모리 방식은 한 시스템내의 프로세스에게 공유 메모리 주소공간을 제공한다. 분산시스템에서는 물리적으로 분리되어 있는 메모리들은 소프트웨어 수준에서 가상 공유메모리(shared memory abstraction) 로 구현이 가능하다. loosely coupled 시스템에서의 공유 메모리를 분산 공유 메모리(DSM : distributed shared memory)라고 한다[3].

DSM 시스템에서의 각 노드는 프로세서와 메모리를 가지며 네트워크으로 연결되어 있다. 각 노드는 프로세서, 메모리, 그리고 네트워크 연결장치 등으로 이루어져 있다. 메모리는 페이지 또는 레코드 단위로 구분되며 페이지는 여러 노드에 복제본을 소유할 수 있다. 이들간 일치성을 유지하기 위하여 무효화 방식(invalidate protocol)과 갱신 방식(update protocol)이 전통적으로 많이 사용되었고 성능향상을 위하여 이들의 변형된 알고리즘들이 제안되었다. 여러 프로토콜을 요약하면 다음과 같다.

- 무효화 프로토콜(invalidate protocol): 어떤 노드에서 쓰기 동작이 일어날 경우 다른 노드들에 복제된 블록들을 모두 무효화시킨다. 따라서, 쓰기 이후에는 단일 복제만 존재하게 된다.
- 갱신 프로토콜(update protocol): 어떤 노드에서 쓰기 동작이 일어날 경우 다른 노드들에 복제되어있는 모든 블록들을 쓰기를 한 내용과 동일하게 갱신시킨다. 쓰기 이후에도 복수개의 복제가 존재할 수 있다.[4]
- 경쟁적인 갱신 프로토콜(competitive update protocol) : 가까운 장래에 사용되어질 복제본을 갱신시켜 페이지 오류(page fault)를 줄임으로써 빠른응답성을 가지고 반면, 다른 복제본은 무효화시키어 다른 노드로 부터의 갱신에 따르는 통신 부하를 줄임으로써 결과적으로 전체 시스템의 성능을 높이기 위한 프로토콜이다.
- 이주 프로토콜(migratory protocol) : 페이지 하나에 대해서 오직 하나의 복사본 만이 허락된다. 그 복사본은 데이터에 읽기나 쓰기를 행하는 노드들을 옮겨 다닌다.

DSM의 성능을 예측하고 분석하기 위하여 이제까지 많은 성능평가 모델이 연구되어 왔다. 대부분의 모델들이 읽기/쓰기 비율 또는 메모리 접근의 실패 비율등을 인수로 사용되었다. 사용 형태를 반영하기 위한 방법에는 복제된 블록마다 다른 노드로부터의 갱신 횟수에 한계값을 가지게 하는 경쟁적 갱신 프로토콜[1], 일정한 시간이 지나도록 사용 형태가 일어나지 않는 블록을 무효화시키는 갱신 타임아웃(update timeout) 방식[2], 실행중 연속되는 쓰기 동작의 수를 측정하여, 앞으로도 유사한 형태로 사용을 한다는 가정하에 쓰기 동작이 연속으로 일어날 것으로 예측되는 구간에서는 다른 복제된 블록들을 무효화시키는 write -run 모델의 적응적 프로토콜(adaptive protocol on write -run model) [4], 두 지역 사용 사이에 다른 노드로부터의 갱신수를 고려하는 거리-적응 갱신 프로토콜(distance -adaptive update protocol) 방법[5], 그리고 응용 프로그램에서 데이터들이 노드들을 옮겨가면서 처리되는 이주(migratory) 형태인 경우 옮기기 전의 노드에서 스스로-무효화(self invalidation) 함으로써 무효화 비용을 줄이는 이주성 공유 적응(adapt to migratory sharing) 방법이 있다[8]. 또, 유사한 방법으로 노드 스스로 주어진 주건에 따라 자동으로 무효화됨으로써 무효화 하는데 필요한 통신비용을 줄이도록 하는 가변적 자발적-무효화 (dynamic self -invalidation) 방법이 있다[7].

논문[kim96]에서는 경쟁적 갱신프로토콜(competitive update protocol)을 위한 성능평가 모델을 제안했으며 성능 평가기준은 DSM 의 일치성 유지를 위한 메시지 오버헤드로 정의했다. 이러한 접근은 세그먼트 모델을 바탕으로 하고 있다. " 세그먼트" 란 한 노드에서 한 페이지를 엑세스 하는 데 필요한 통신 비용을 계산하는 기본 단위가 된다. 새로운 " 세그먼트" 의 시작은 원격(remote) 노드가 해당 공유메모리의 페이지에 쓰기를 수행한 후 첫 번째로 지역(local) 노드가 그 페이지를 사용(읽기 또는 쓰기)하는 시점이다. 즉, 세그먼트는 지역 노드의 메모리 사용과 그 사이의 일련의 원격 노드의 쓰기들로 이루어져있다. 본 연구에서는 논문[5]을 바탕으로 하여 DSM을 위한 간편한 성능평가모델을 제시한다. 이전 연구[5]가 경쟁적 갱신 프로토콜(competitive update protocol) 을 위한 성능을 분석했던 반면에 본 연구에서 제시하는 성능평가 모델은 무효화 프로토콜(invalidate protocol), 갱신 프로토콜(update protocol) 그리고 이주 프로토콜(migratory protocol)의 성능을 평가하기 위한 것이다. 본 연구가 제안하는 새로운 평가 모델은 [5] 과 쉽게 결합하여 4개의 프로토콜(update, invalidate, migratory 그리고 competitive update protocol) 의 성능을 비교 평가할 수 있다.

2. 성능 분석 모델

size가 m 인 cost 는 c(m) 으로 정의한다. 한 세그먼트 내에서 평균 갱신(update)수를 u 로 정의한다. 원하는 페이지가 지역(local)에 없으면 페이지 실패(page fault)가 일어난다. 페이지 실패(page fault)에 의한 통신 비용과 다른 노드에서 원하는 페이지를 페치(fetch)하는 비용을 동일하게 c(pf)로 정의한다. 무효화 메시지를 전송시키는 비용을 c(inv)로 정의한다. 페이지를 갱신하는 횟수와 비용을 각각 U 와 c(upd)로 정의한다. 세그먼트당 지역 억세스 사이에 발생한 일련의 평균 원격지 읽기횟수를 r(remote read) 로 정의한다.(이때 원격지 쓰기사이에 발생하는 원격지 읽기는 제외한다.)
각 프로토콜의 성능분석은 아래와 같다.

세그먼트당 지역 노드의 일치성 유지를 위한 비용을 계산한다면 다음과 같다. (비용 계산시 지역 노드에서 원격 노드로 갱신 메시지 전송은 원격 노드의 비용으로 포함한다.)

- 무효화 프로토콜(invalidate protocol) : 세그먼트가 시작될 때 페이지 오류(page fault)가 발생되어 페이지를 다른 노드로부터 전송 받는데 페이지 오류 비용이 필요하다(c(pf)). 또한, 원격 노드에서 첫 번째의 쓰기 발생시 한 개의 갱신 비용(c(inv))이 필요하지만 원격 노드의 두 번째 쓰기 발생시 지역 노드에는 복제가 없으므로 추가 메시지가 필요없다. 따라서 세그먼트당 필요한 비용은 c(pf) + c(inv) 이다. (페이지 오류 비용 + 갱신 비용)

- 갱신 프로토콜(update protocol) : 원격 노드에서 쓰기가 발생될 때 마다 한 개의 갱신 비용(c(upd))이 필요하다. 따라서, 세그먼트에서 총 u 개의 갱신 메시지를 받았다면 비용은 c(upd) * u 이다.(갱신비용*u)

- 이주 프로토콜(migratory protocol) : 이주 프로토콜의 성능을 평가하기 위해서는 앞에서 정의한 인수들 이외에 다른 인수가 필요하다. 이주 프로토콜에서는 오직 하나의 복사본을 허락하므로 다른 노드에서의 세그먼트당 원격지 읽기(remote read) 또한 성능평가에 영향을 미친다. 한 세그먼트내에서 연속되는 지역 읽기/쓰기 사이의 원격지 읽기횟수를 r 이라고 할 때 이주 프로토콜의 성능은 c(cf) + r*c(pf) 이다. (페이지 오류비용 + 평균 원격지 읽기 * 페이지 오류비용)

위에서의 성능분석 모델을 정리하면 다음과 같다.
- 무효화 프로토콜 : c(pf) + c(inv)
- 갱신 프로토콜 : u * c(upd)
- 이주 프로토콜 : c(pf) + r * c(pf) = (r+ 1) * c(pf)

3. 성능 평가

DSM을 위한 성능평가를 위해서 위에서 정의한 파라미터들의 값을 변화시키면서 성능을 비교한 결과는 다음과 같다. 8노드 워크스테이션 클러스터(10M bps)에서 측정한 결과 상대적인 비용은 c(inv) 를 1 이라고 할 때 c(pf)는 8 이였고 c(upd) 는 update 크기에 따라 다르지만 3으로 가정했다. (아래 성능비교에서 이 비용은 고정값으로 사용하였다.) DSM 프로토콜의 성능은 이러한 시스템 환경에 따라 다르며 애플리케이션의 메모리 억세스 형태에 따라 차이가 있다. 다음은 비용에 영향을 미치는 파라메터들을 변화시키면서 비용을 비교하였다.

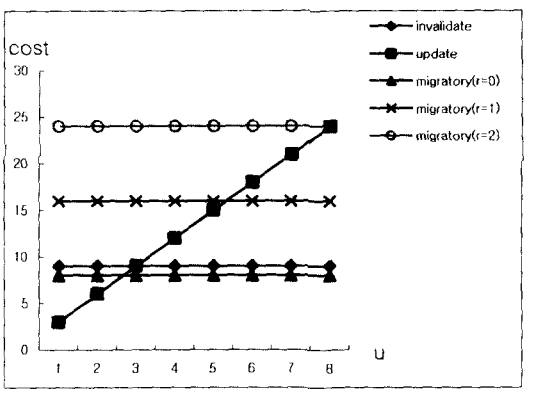

그림 1. u 변화에 따른 성능비교

그림 1 은 세그먼트당 원격 노드들의 평균쓰기 횟수(u)를 변화 시키면서 성능을 비교한 결과이다. 갱신 프로토콜만이 u 에 비례하여 비용이 증가하였다. 그림 2는 페이지 오류(page fault) 비용(c(pf))을 증가시키면서 성능을 비교하였다. 무효화 프로토콜과 이주 프로토콜이

c(pf)가 증가함에 따라 높은 비용이 소요됨을 알 수 있다. 그림 3 은 세그먼트당 평균 원격지 읽기(remote read)횟수(r)에 따른 성능 비교를 나타낸다. 이주 프로토콜의 비용이 r 이 증가함에 따라 높은 비용이 소요된다. 그림 4 는 갱신비용(c(upd))에 따른 성능 비교를 나타낸다. 그림 1 에서와 같이 갱신 프로토콜만이 c(upd)에 비례하여 비용이 증가됨을 나타내었다.

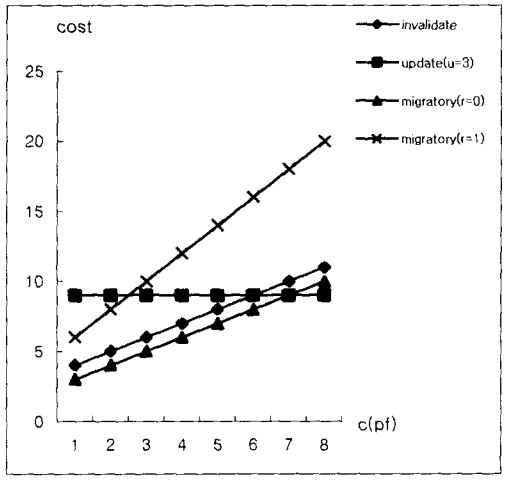

그림 2. c(pf) 의 변화에 따른 성능비교비교

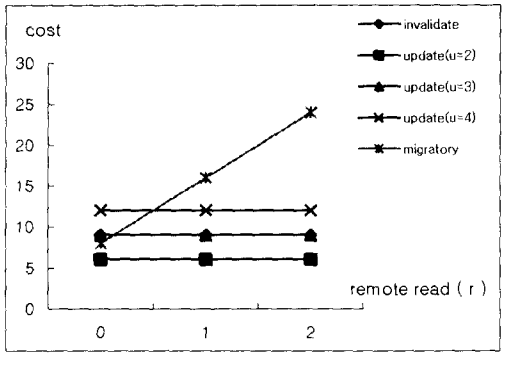

그림 3. remote read 의 변화에 따른 성능비교

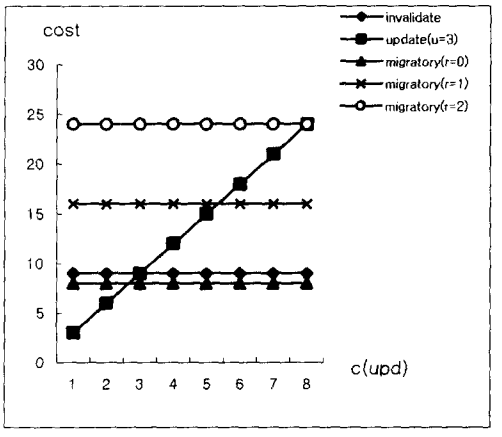

그림 4. c(upd)의 변화에 따른 성능비교

4. 결론

본 논문에서 제시한 비용 모델에 의한 성능 비교는 간단하지만 [6] 에서와 같이 실제 애프리케이션 프로그램을 수행 할때와 결과를 비교하면 일반적으로 비용 모델이 정확함을 알 수 있다. 따라서 프로그램을 1 회 수행하여 애플리케이션 마다 상이한 변수들(u,r)을 구하면 각종 프로토콜(무효화, 갱신, 이주 프로토콜)의 성능을 비교 할 수 있기 때문에 프로토콜들을 각기 수행할 필요 없이 최적의 프로토콜을 선택할 수 있다.

참고 문헌

[1] B. Falsafi, A. Lebeck, S. Reinhart, I. Schoinas, M. Hill, J. Larus, A. Rogers, and D. Wood, " Application-specific protocols for user -level shared memory " in Proc. of Supercomputing' 94, pp. 308-389, Nov. 1994.

[2] H. Grahn, P. Stenstrom, and M. Dubois, "Implementaiton and evaluation of update -based cache protocols under relaxed memory consistency models," Future Generation Computer Systems, vol. 11, pp. 247-271, Jun. 1995

[3]M. Stum and S.Zhou, " algoriths inplementing distributed shared memory," IEEE computer, vol 23, pp. 54 -64, May 1990.

[4] A. Lebeck and D. Wood, " Dynamic self -invalidation: reducing coherence overhead in shared -memory multiprocessors" in Proc. of the 22 nd Annual International Symposium on Computer Architecture, pp. 48-59

[5]Jai-Hoon Kim and Nitin H. Vaidya, "A Cost -Comparison Approach for Adaptive Distributed Shared Memory," 10th ACM International Conference on Supercomputing, Philadelphia, Pennsylvania, pp. 44-51, May 1996.

[6] Jai-Hoon Kim and Nitin H. Vaidya, "A Cost Model for Distributed Shared Memory Using Competitive Update," International Conference on High Performance Computing, Bangalore, India, pp.112-117, December 1997.

[7] U. Ramachandran, G. Shah, A. Sivasubramaniam, A. Singla, an d I. Yanasak, " Architectural mechanisms for explicit communication in shared memory multiprocessors" in Proc. of Supercomputing' 95, Dec. 1995.