PMS : 다단계 저장장치를 고려한 효율적인 선반입 정책

이규형° 이효정 노삼혁
홍익대학교 컴퓨터 공학과
lkh@cs.hongik.ac.kr, hjlee@mail.hongik.ac.kr, samhnoh@cs.hongik.ac.kr

PMS : Prefetching Strategy for Multi-level Storage System

Kyuhyung Lee° Hyojeong Lee Sam H. Noh
Department of Computer Science, Hongik University

최근의 저장 장치는 많은 사용자들이 동시에 요청하는 대용량의 자료를 처리하기 위해 다단계로 확장하는 추세이다. 하지만 CPU와 메모리의 발달로 인해 데이터 I/O 는 전체 시스템의 성능을 해치는 병목구간이며, 다단계 저장장치의 경우 데이터에 접근하기 위한 과정이 늘어남으로 전체 시스템의 성능을 더욱 떨어뜨릴 수 있다. 이러한 저장장치의 약점을 극복하기 위해 이미 요청된 데이터를 버퍼 캐쉬에 저장해 놓음으로써 성능 향상을 꾀하는 방법과 필요한 데이터를 미리 가져다 놓음으로 성능을 향상시키는 선반입 기법이 오래 전부터 연구되어 왔다. 다단계 저장장치를 고려한 버퍼 캐쉬 교체 정책은 여러 가지로 연구가 되어 왔으나, 다단계 저장 장치를 고려한 선반입 기법에 대한 연구는 현재까지 거의 이루어 지지 않았다. 하지만 선반입 기법을 잘 활용한다면, 다단계 저장장치의 각 단계에서 걸리는 I/O 시간을 효과적으로 감출 수 있기 때문에 다단계 저장장치에서 선반입 기법은 성능향상을 위해 대단히 중요한 역할을 할 수 있을 것으로 기대한다. 선반입 기법은 일반적으로 현재 필요한 데이터를 디스크에서 읽어올 때 추가로 일정량을 미리 읽어 옴으로써 향후 연속적인 데이터 요청에 대비를 하는 방법을 사용한다. 그러나 앞으로의 요청을 정확하게 예측하지 못할 경우엔 필요하지 않을 데이터를 미리 가져옴으로써 캐쉬 공간과 디스크 사용의 낭비를 초래하거나 미리 데이터를 준비하지 못함으로써 선반입으로 인한 이득을 누릴 수 없게 된다. 과거의 1단계 시스템이라면, 부정확한 예측으로 인한 선반입의 부작용이 크지 않기 때문에 과감한 선반입 정책을 사용할 수 있었다. 그러나 다단계 시스템의 각 단계에서 개별적인 선반입을 수행한다면, 부정확한 선반입으로 인한 낭비가 증폭되어 1단계 시스템일 때와는 비교할 수 없이 크게 늘어나게 된다.

한 시스템 내에서의 선반입 기법은 많은 연구가 진행되어 왔다. 그러나 현재까지 다단계의 저장장치를 고려한 다단계 선반입 정책은 특별한 연구 결과가 없다. 한 시스템 내에서의 선반입 기법에 대한 많은 연구는 "언제", "무엇을" 선반입 할 것인가에 초점을 맞추어 왔다. 대부분의 선반입 연구는 순차적인 선반입을 기반으로 선반입 시기와 양을 효과적으로 결정하는데 주력하여 왔다. 일어난 요청에 연속적으로 이어지는 블록을 순차적으로 선반입을 하게 되면, 매우 작은 비용으로 디스크 접근을 할 수 있기 때문에 선반입 기법의 대부분은 순차적 접근을 하고, 본 연구에서 제안한 PMS 기법도 순차적 선반입 기반으로 효율적인 선반입 알고리즘을 개발하였다. 기존의 1레벨 시스템을 위한 선반입 연구는 다양하게 이루어져 왔으며, 본 연구에서는 그 중 가장 널리 쓰이는 선반입 정책인 AMP[1], SARC[2], 리눅스 선반입 정책, Read Ahead(RA)[3] 정책을 사용하여 실험하고, PMS의 성능을 평가하였다.

다단계 저장 장치 시스템에서 효율적인 선반입을 수행하기 위하여, 본 연구에서는 상위 시스템의 선반입 기법에 의존하지 않는, 넓은 용도로 사용할 수 있는 일반적인 하위 레벨 선반입 기법을 제안 하였다. 본 연구에서 제안하는 다단계를 고려한 선반입 기법은 상위 레벨 시스템의 선반입 정책, 혹은 캐쉬 교체 정책이 무엇이던 전체 시스템의 성능을 향상 시킬 수 있는 독립적인 기법이다. 다단계 저장장치에

서 각 단계 시스템이 독립적인 선반입 정책을 사용한다면, 상위 레벨에서의 낭비가 작더라도 하위 레벨 시스템에선 상위 시스템의 선반입이 포함된 요청을 처리하며 추가적인 선반입을 수행하기 때문에 전체 시스템에서는 캐쉬 공간과 디스크 I/O 측면에서 큰 낭비를 초래할 수 있다. 반대로, 상위 시스템에서 앞으로 일어날 요청보다 너무 작은 선반입을 수행한다면, 하위 레벨에서도 역시 충분하지 않은 선반입을 수행할 것이기 때문에, 전체 시스템의 측면에서 보면 선반입의 효과를 충분히 누리지 못한다. 그래서 본 연구에서 제안하는 PMS (Prefetch strategy for Multi-level Storage system)는 상위 레벨의 선반입이 포함된 요청을 분석하여 이제까지의 히스토리를 바탕으로 최대한 적절한 시기에 적절한 양을 선반입하는 기법이다. 즉, PMS 기법은 하위 레벨의 시스템에 적용되는, 상위 레벨의 선반입에 독립적으로 항상 좋은 성능을 이끌어 낼 수 있는 선반입 기법이다. 이를 위해 PMS 기법에서는 각각의 요청에 대해 다음의 두 가지 동작 중 하나, 혹은 두 개의 동작을 취해 선반입을 행함으로써, 상위 레벨의 선반입 정책이 적절하지 못한 수행을 하더라도 하위 레벨에서는 이를 최대한 효율적으로 수행한다.

- Bypass : 상위 레벨에서 온 요청 중 적중되지 않은 블록의 일부는 캐쉬에 저장하지 않고 바로 하위 레벨, 혹은 디스크로 요청한다. 상위 레벨의 캐쉬가 모자라다면, Bypass를 최소화 하여 향후 상위레벨에서 캐쉬 미스로 요청되는 블록을 빠르게 제공할 수 있다.
- Readmore : 요청 블록에 연속되는 선반입을 요청한다. 얼마나 많은 블록을 선반입 할지에 대한 결정은 이제까지의 요청에 대한 히스토리를 기반으로 결정한다.

PMS 기법에서는 어떤 블록을 bypass하고, 어떤 블록을 readmore 할지를 결정하기 위해 Bypass큐와 Readmore 큐를 사용한다. 두 개의 큐에는 실제 데이터가 아닌, 블록 번호만이 저장되기 때문에 메모리 사용량은 극히 적다. 또한 두 개의 큐는 각각 전체 캐쉬 크기의 10%에 해당하는 블록번호 만을 저장하며 이는 LRU 정책에 의해 교체된다. Bypass 큐에는 이전까지 bypass 되었던 블록 번호가 저장되어 이후 상위 레벨에서 요청된 블록이 Bypass 큐에 있는 것이 확인된다면, 이는 상위 레벨 캐쉬의 크기가 부족하여 이전 선반입 된 블록이 사용되기 전에 교체되었다고 판단하여 앞으로의 bypass 블록 개수를 줄인다. Readmore 큐에는 readmore 가 일어난 블록 이후의 일정 개수 블록 번호를 저장하여 이후 요청이 Readmore 큐에 있는 블록 중 하나라면 readmore 가 효과적으로 수행되고 있다는 증거이므로 readmore를 적극적으로 수행한다.

본 연구에서는 제안한 기법의 성능을 평가하기 위하여, 정밀한 2단계 저장장치 시뮬레이터를 제작하여, 1단계 시스템에 서로 다른 4가지의 선반입 정책을 적용하여 실험하였다. SPC의 Web trace와 Purdue대학의 Multi의 2가지의 널리 쓰이는 트레이스를 이용하여 실험한 결과, 본 연구에서 제안한 선반입 기법을 이용할 경우, 상위 시스템의 선반입 정책이 무엇이든 상관없이 좋은 성능을 나타냄을 보였다. 버퍼 캐쉬의 크기, 상위 레벨의 선반입 정책등을 달리하여 실험한 총 32가지의 실험 모두에서 하위 레벨 시스템에 기존에 연구된 선반입 시스템을 사용하였을 때 보다, 본 연구에서 제안된 일반적인 선반입 시스템을 적용할 경우, 응답시간을 측정하였을 때 최대 35%, 평균 16.56%의 성능 향상을 확인 할 수 있다.

참고문헌

[1] B. Gill and L. Bathen. AMP: Adaptive multi-stream prefetching in a shared cache. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)*, 2007.

[2] B. Gill and D. Modha. SARC: Sequential prefetching in adaptive replacement cache. In *Proceedings of the 2005 USENIX Annual Technical Conference*, pages 293–308, 2005.

[3] A. Smith. Cache memories. In *ACM Computing Surveys (CSUR)*, volume 14, pages 473–530. ACM Press, 1982.