

스냅샷을 가지는 다중 레벨 공간 DBMS에서의

효율적인 스냅샷 교체 기법*

오은석⁰ 어상훈 김호석 배해영

인하대학교 컴퓨터 정보공학과

{oes⁰, eosanghun, bluesnow}@dblab.inha.ac.kr hybae@inha.ac.kr

Efficient Snapshot Replacement Technique

in Multi-Level Spatial DBMS with Snapshot

Eun-Seoug Oh⁰ Sang-Hun Eo Ho-Seok Kim Hae-Young Bae

Dept. of Computer Science & Information Engineering, Inha University

요 약

오늘날 우리가 사용할 수 있는 정보의 양은 실로 막대하면서도 지속적으로 증가하고 있는 추세이며, 동시에 데이터들의 신속한 처리에 대한 관심이 늘고 있다. 특히 GIS에서 사용되는 대용량 데이터나 빠른 처리 속도가 요구되는 인중 시스템 데이터와 같은 다양한 종류의 데이터 특성을 고려하여 효율적인 관리를 하는 데이터베이스 관리 시스템을 필요로 하고 있다.

스냅샷 데이터를 갖는 다중 레벨 공간 DBMS는 대용량, 또는 빠른 트랜잭션 처리 속도가 필요한 데이터들을 효율적으로 관리하는 데이터베이스 관리 시스템으로서, 대용량 데이터는 디스크 데이터베이스에서 그리고 빠른 트랜잭션 처리를 요구하는 데이터들은 스냅샷의 형태로 메모리 데이터베이스에서 관리한다. 메모리 데이터베이스에 저장되는 스냅샷은 상대적으로 더욱 중요한 스냅샷들이 집중적으로 생성되거나 스냅샷이 가진 특성이 변화될 경우, 메모리 데이터베이스 내에서의 저장 가치를 잃을 수가 있다. 따라서 메모리 데이터베이스에 불필요한 스냅샷들이 축적되는 문제를 해결하고 메모리의 이용성과 성능을 보존하기 위해서 효율적으로 스냅샷들을 교체하는 기법이 필요하다.

본 논문에서는 다중 레벨 공간 DBMS에서 질의 패턴을 이용한 효율적인 스냅샷 교체 기법을 제안한다. 제안된 기법은 메모리 데이터베이스 내에서 스냅샷에 대한 교체요청이 있을 경우, 메모리 데이터베이스 및 메모리 관리 시스템의 상황을 분석하여 주어진 상황에 대응하는 스냅샷 교체 기법을 수행한다.

1. 서 론

스냅샷 데이터를 가지는 다중 레벨 공간 DBMS는 메인 메모리 데이터베이스와 디스크 데이터베이스를 하나의 시스템에 운영하는 DBMS이다[1, 2, 3]. 본 시스템은 비교적 대용량인면서 빠른 접근속도를 요구하지 않는 데이터의 경우, 디스크 데이터베이스에서 관리하며, 보다 빠른 처리속도와 잦은 참조를 요구하는 데이터는 스냅샷의 형태로 메모리 데이터베이스에서 관리한다. 스냅샷 데이터를 가지는 다중 레벨 공간 DBMS의 메모리 데이터베이스는 디스크 데이터베이스에 비해서 상대적으로 제한된 메모리 공간에 불필요한 스냅샷들이 누적될 경우, 메모리 스냅샷에 대한 참조가 실패하고 디스크로의 접근횟수가 증가하게 된다. 따라서 스냅샷이 가지는 데이터 특성을 정확히 파악하여, 상대적으로 메모리 데이터베이스에 저장될 가치가 적은 스냅샷들을 교체하는 기법이 필요하다.

스냅샷 데이터를 가지는 다중 레벨 공간 DBMS는 현재, 특정 시간대에 걸쳐 질의가 집중적으로 들어오는 상황에 대응하기 위해서 LRU(Least Recently Used) 기법을 사용하고 있다. 그러나 LRU 기법은 스택의 거리가 증가함에 따라 거의 LRU 정책에 무관해지며, 재참조 될 가능성이 없어도 한번 참조되고

난 이후에 긴 시간동안 캐쉬에 저장[4, 5]되는 등의 문제점들을 가지고 있다. 따라서 LRU만을 스냅샷의 교체 기법으로 사용할 경우, 메모리 데이터베이스 내에 저장된 스냅샷의 관리 목적을 전혀 고려하지 않는 교체작업을 수행하게 된다.

본 논문은 메모리 데이터베이스에서 LRU 기법을 교체정책으로 사용할 때 발생하는 여러 문제들을 해결하고 효율적인 교체 정책을 결정하기 위해서 다음과 같은 스냅샷의 특성들을 고려한다. 첫째, 최근참조시간, 참조빈도, 순차참조나 순환참조, 그리고 생성 및 유지비용과 같은 시간적인 참조패턴 내에서 가지는 스냅샷의 중요도를 고려한다. 둘째, 스냅샷 교체시기에 스냅샷 관리자의 높은 부하가 있을 경우, 가변적인 분석기준을 결정하여 교체 작업에 의한 부담을 줄이는 정책을 고려한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로서 시스템 환경과 기존의 버퍼 교체 기법들을 간략히 설명하고, 3장에서는 질의 패턴을 이용한 효율적인 스냅샷 교체 기법을 설명한다. 마지막으로 4장에서는 이 기법을 바탕으로 결론 및 향후 과제를 논의한다.

2. 관련 연구

본 장에서는 제안 기법이 사용하는 다중 레벨 공간 DBMS에 대한 설명과 함께, 메모리 데이터베이스에서의 스냅샷 교체 기법과 유사한 환경을 가진 기존의 버퍼 관리기법에 대한 설명을 한다.

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성, 지원사업의 연구결과로 수행되었음.

2.1 스냅샷을 가지는 다중 레벨 공간 DBMS

스냅샷을 가지는 다중 레벨 공간 DBMS는 대용량 데이터를 지원하는 기존의 디스크 기반 데이터베이스에 빠른 트랜잭션 속도를 위해서 메모리 데이터베이스를 확장한 시스템이다. 본 시스템에서의 디스크 기반 데이터베이스는 대용량의 데이터를 안전하게 지원하며, 이 디스크 데이터베이스에서 비교적 높은 접근 빈도와 빠른 접근속도를 필요로 하는 데이터의 경우, 자동으로 스냅샷을 생성하여 이를 메모리에 저장한다. 자동으로 생성된 스냅샷들은 시간의 경과에 따라서 더 이상의 참조를 가지지 않고 메모리에 축적될 수 있다. 이러한 스냅샷의 축적이 발생할 경우, 새로운 스냅샷을 메모리 데이터베이스에 저장할 수 없는 문제가 발생하여, 수많은 디스크 I/O를 일으킬 수 있다. 따라서 스냅샷 데이터를 저장하는 메모리 데이터베이스는 효율적인 메모리 데이터베이스 교체 알고리즘을 이용하여, 메모리 내에서 더 이상의 가치를 가지지 않는 스냅샷들을 효율적이고 신속하게 교체해야 한다.

2.2 기존의 버퍼 교체 기법

본 시스템에서의 메모리 데이터베이스는 고성능 트랜잭션의 처리를 위해서 메모리 공간을 마치 버퍼와 유사한 역할로 사용한다. 따라서 효율적인 메모리 교체 기법을 위해서 기존의 버퍼 캐쉬 교체 기법들[4, 5, 7, 8, 9, 10]을 알아보도록 한다. 버퍼 관리기법으로는 다음과 같은 기법들이 있다. LRU 기법은 가장 오랫동안 사용되지 않은 페이지를 교체하는 기법이다. 이 기법은 작업집합의 변화에 잘 적응하며, 블록들이 시간 지역성을 보일 때 가장 좋은 교체효율을 가진다[6]. LFU는 각 페이지들이 얼마나 많이 참조되었는지를 교체기준으로 정하고, 페이지 교체가 필요할 경우 이 기법은 참조횟수가 가장 적은 페이지를 교체한다. OPT는 가장 오랫동안 사용되지 않을 페이지를 교체하는 기법으로, 각 페이지의 호출순서나 참조상황 등을 미리 예측해야 하므로 실제 구현은 불가능한 기법이다. NUR은 최근에 사용되지 않은 페이지를 교체하는 기법이며, 이러한 최근사용 여부를 확인하기 위해서 각 페이지마다 참조 비트와 변형비트를 사용한다. 그러나 기존의 교체 기법들은 정의된 몇 가지의 기준에는 효율적으로 대응하지만, 그 이외의 측면에는 효율적인 대응을 할 수 없는 한계를 가지고 있다.

3. 질의 패턴을 이용한 효율적인 스냅샷 교체 기법

스냅샷 관리자는 메모리 데이터베이스의 효율적인 관리를 위해서, 더 이상의 가치를 가지지 못하는 메모리 스냅샷을 신속하게 교체해야 한다. 따라서 스냅샷들이 메모리 내에서 가지는 가치를 정확하게 판단하기 위해 질의 패턴을 이용한 효율적인 스냅샷 교체 기법을 이용하며, 이를 위해서 최근참조시간, 전체참조횟수, 순환참조, 생성비용, 유지비용과 같은 스냅샷 메타데이터들을 이용한다.

3.1 스냅샷이 가지는 측정 요소

메타데이터에 있는 해당 스냅샷의 시간정보인 참조횟수와 최근 참조시간을 분석하고, 단위 시간 내에 걸치는 스냅샷들의 참조주기를 파악한다. 참조주기는 크게 순차참조와 순환참조로 분류될 수 있다. 순차참조는 재참조 가능성이 무한대이므로 높은 우선순위로 교체되어야 한다. 그리고 순환참조의 경우는, 그 순환주기를 측정하여 주기가 클수록 메모리 내에서 높은 우선순위로 교체되어야 한다. 메타데이터에 있는 스냅샷의 공간정보인 비용분석은 생성비용과 유지비용의 합계로 나타낸다. 생성비용은 임의의 시점에서 해당 스냅샷이 교체되었을 경우, 이를 재참조하기 위하여 메모리로 재생성을 하는 데 필요한 비용이다. 또한 유지비용은 해당 스냅샷을 유지하기 위해서 단위 시간 내에 수행되는 리프레시 비용들에 대한 합계이다. 그림 1

은 스냅샷을 측정하기 위한 알고리즘을 나타낸 것이다.

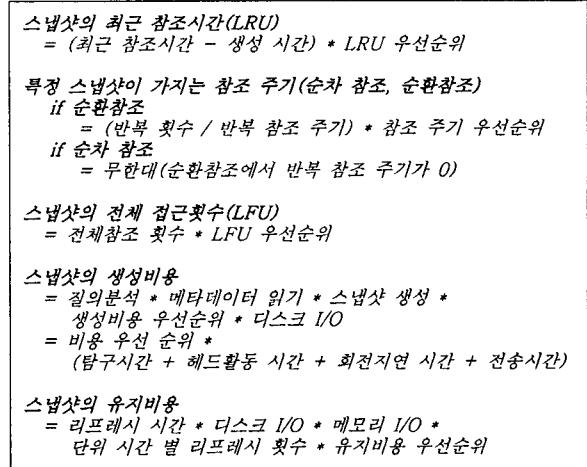


그림 1 제안 기법의 스냅샷 측정 요소

이러한 각각의 스냅샷이 가지는 5개의 기준을 이용하여 스냅샷의 중요도를 분석하고, 이를 기반으로 메모리 내에서 해당 스냅샷이 가지는 교체 순위를 파악한다. 이는 LRU, LFU와 같은 단순 예측 알고리즘보다 정확한 분석을 함으로서 메모리 공간을 효율적으로 관리하고, 스냅샷 교체로 인해 유발될 수 있는 디스크 데이터베이스의 접근 또한 줄일 수 있는 장점을 가진다.

3.2 스냅샷 희생자 선정 알고리즘

본 논문은 스냅샷의 희생자를 선정하기 위해서 그룹 구조로 스냅샷을 관리한다. 또한 스냅샷 관리자가 스냅샷 교체 시점에 걸처서 다른 작업들로 부하가 있을 경우, 위에서 언급한 교체 기준을 조절하여 스냅샷 교체로 인한 시스템 오버헤드를 줄인다. 교체기준은 최근참조시간, 전체참조횟수, 순환참조, 생성비용, 유지비용의 순서를 가지며, 각각의 기준을 위한 그룹 구조를 가진다. 또한 그룹의 순서대로 가치가 적은 스냅샷들을 소환하여, 이들을 다음 그룹의 가치에 연속적으로 적용, 마지막 그룹 내에서 상대적으로 가치가 없다고 최종 스냅샷 삭제 후보들을 교체하는 알고리즘을 가진다.

그림 2는 이러한 스냅샷 교체를 위한 구조를 보여준다. 스냅샷에 대한 교체요청이 있을 경우, 스냅샷 관리자는 본 기법의 기반이 되는 다중 레벨 공간 시스템의 부하상태를 파악한다. 그리고 시스템의 부하상태가 높을 경우, 상대적으로 제한된 측정 요소들을 이용하여 교체될 스냅샷을 선정한다. 예를 들어, 만약 시스템이 낮은 부하 상태를 가진다면, 스냅샷 관리자는 메타데이터의 모든 기준을 이용하여 해당 스냅샷의 가치를 파악한다.

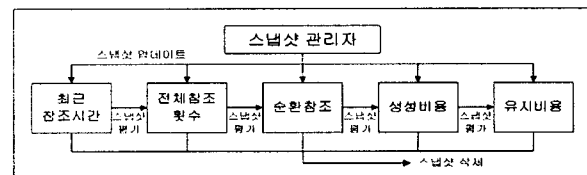


그림 2 제안 기법의 스냅샷 교체 구조

그러나 특정 시간대에 집중적으로 발생하는 스냅샷 리프레시,

질의관련 수행, 또는 업데이트와 같이 극단적으로 시스템에 부하를 주는 상황이 발생할 경우, 스냅샷 관리자는 1번째 참조 기준인 최근참조시간 만을 참조하여 스냅샷을 교체할 수 있다. 교체된 스냅샷에 대한 정보는 메타데이터에 반영되며, 또한 추가적인 메타데이터는 스냅샷 관리자의 하나인 리프레시 관리자가 주기적으로 업데이트한 정보를 이용한다. 그리고 이러한 스냅샷의 교체시기는 메모리 데이터베이스가 디스크 데이터베이스의 접근비용을 줄인다는 기본목적을 고려해야 하며, 교체가 아닌 관리를 목적으로 다루는 문제이므로 본 논문에서는 다루지 않는다. 다만, 스냅샷 관리자는 이러한 주기에 맞추어 해당 그룹 내에 저장되는 스냅샷의 특성이 맞도록 관리해야 한다. 그림 3은 시스템의 부하를 고려하면서 3절에서 언급하였던 측정 요소들을 이용하여 스냅샷을 효율적으로 교체하는 알고리즘을 보여준다.

```

while(스냅샷 교체시기 == true) {
  /* 스냅샷 관리자에서 정의한 시기를 이용 */

  if(스냅샷이 차지하는 메모리 비율 >
    스냅샷 관리자에서 정의한 기준비율) {
    /* 스냅샷 관리자에서 정의한 시기를 이용 */
    DropType
    = round(스냅샷 관리자의 부하상태(%)/100);
    /* 스냅샷 관리자의 부하상태를 고려한 스냅샷 교체 알고리즘 */

    if(DropType >= 80) {
      /* 스냅샷 관리자가 가장 바쁠 경우 */
      최근 참조시간 그룹에서 삭제 후보 스냅샷들을 선정;
      삭제후보로 선정된 스냅샷들을 교체;
    }

    if(DropType >= 60) {
      최근 참조시간 그룹에서 삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 전체참조 횟수 그룹에서의
      삭제 후보 스냅샷들을 선정;
      삭제 후보로 선정된 스냅샷들을 교체;
    }

    if(DropType >= 40) {
      최근 참조시간 그룹에서 삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 전체참조 횟수 그룹에서의
      삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 순환참조 그룹에서의
      삭제 후보 스냅샷들을 선정;
      삭제 후보로 선정된 스냅샷들을 교체;
    }

    if(DropType >= 20) {
      최근 참조시간 그룹에서 삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 전체참조 횟수 그룹에서의
      삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 생성비용 그룹에서의
      삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 유지비용 그룹에서의
      삭제 후보 스냅샷들을 선정;
      삭제 후보로 선정된 스냅샷들을 교체;
    }

    else {
      /* 서버의 부하가 가장 적은 경우 */
      최근 참조시간 그룹에서 삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 전체참조 횟수 그룹에서의
      삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 순환참조 그룹에서의
      삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 생성비용 그룹에서의
      삭제 후보 스냅샷들을 선정;
      선정된 스냅샷들 중에서 유지비용 그룹에서의
      삭제 후보 스냅샷들을 선정;
      삭제 후보로 선정된 스냅샷들을 교체;
    }

    메타데이터 업데이트;
  }
}

```

그림 3 제안 기법의 스냅샷 교체 알고리즘

4. 결론

본 논문은 메모리에 저장되는 스냅샷 교체 알고리즘을 시스템에 기반하여 효율적으로 수행하기 위한 스냅샷 교체 알고리즘을 제안하였다. 본 논문의 스냅샷 교체 알고리즘은 스냅샷이 가지는 고유한 패턴을 파악하기 위하여 해당 스냅샷이 가지는 메타데이터를 메모리에 저장하고 이 데이터들을 분석하여 스냅샷의 교체 순위에 적용한다. 이 알고리즘을 적용한 스냅샷 우선순위 알고리즘은 LRU 기법 이상으로 정확한 스냅샷 단위의 우선순위를 가지면서, 또한 시스템 부하의 측면을 고려하는 효율적인 교체효율을 가지고 있다.

본 논문은 향후 연구과제로서 보다 정확한 질의구문의 파악을 통한 추가적인 분석요소와 함께, 스냅샷의 교체성능을 향상시키는 정책에 관한 연구가 필요하다.

참고 문헌

- [1] S. H. Eo, S. K. Jjang, J. D. Lee and H. Y. Bae: Multi-level SDBMS with Snapshots. In ASGIS, 2005
- [2] S. K. Jjang, S. H. Eo, H. S. Kim and H. Y. Bae: Query Classification Method for Performance Enhancement in Multi-Level Spatial SDBMS with Snapshots. In ASGIS, 2005.
- [3] 주봉, 어상훈, 김명근, 조숙경, 배해영. "스냅샷 데이터를 갖는 다중 레벨 저장 시스템에서의 효율적인 리프레시 기법". 한국정보처리학회 춘계학술대회 논문집. 제12권, 제1호 (2005. 5)
- [4] E. J. O'Neil, P.E. O'Neil, G Weikum. "The LRU-K Page Replacement Algorithm For Database Disk Buffering", Proc. of the 1993 ACM SIGMOD Conference, pp 297-306, 1993.
- [5] T. Johnson, D. Shasha. "2Q: A Low Overhead High Performance Buffer Management Replacement Algorithm", Proc of The 20th International Conference on Very Large Data Bases, pp 439-450, 1994.
- [6] E. G. Coffman, JR. and P. J. Denning. "Operating Systems Theory", Prentice-Hall International Editions, 1973.
- [7] J. T. Robinson and M. V. Devarakonda. Data cache management using frequency-based replacement. In Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems, pages 134-142. ACM Press, 1990
- [8] P. Cao, E. W. Felten, and K. Li. Application Controlled File Caching Policies. In Proceedings of the USENIX Summer 1994 Technical Conference, pages 171-182, 1994.
- [9] R. H. Patterson, G. A. Gibson, E. Gintering, D. Stodolsky, and J. Zelenka. Informed Prefetching and Caching. In Proceedings of the 15th Symposium on Operating System Principles, pages 1-16, 1995.
- [10] D. Lee, J. Choi, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim. On the Existence of a Spectrum of Policies that Subsumes the Least Recently Used (LRU) and Least Frequency Used (LFU) Policies. In Proceedings of the 1999 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pages 134-143, 1999.