

다차원 데이터의 실시간 검색을 위한 내장형 주기억장치 자료 저장시스템의 구현 및 성능평가

권오수*, 정재보*, 홍동권*,
*계명대학교 컴퓨터공학과
e-mail: dkhong@kmu.ac.kr

Implementation and performance evaluation of embedded main-memory storage system for real-time retrieval of multidimensional data

Oh-Su Kwon*, Jae-Bo Jung*, Dong-Kweon Hong*
*Dept of Computer Engineering, Kei-myung University

요약

이동 단말기 관리, 무인 항공 제어 시스템 등의 시스템에서는 검색 대상의 정보(위치, 여러 가지 상태 등)가 시시각각으로 빠르게 변화하므로 현재의 상태를 정확히 파악하기 위하여 많은 양의 자료 검색, 변경 요청이 빈번히 발생한다. 이와 같은 시스템에서의 상태 정보 검색은 자료의 효용성이 사라지기 전에 이루어져야 하므로 디스크 I/O가 많은 디스크 상주형 데이터베이스로는 한계점을 안고 있다. 또한 빠른 검색을 지원할 수 있는 주기억장치 상주형 데이터베이스로는 다량의 데이터를 저장해야 하는 어려움을 안고 있다. 본 논문에서는 위와 같은 실시간 검색 기능과 대용량 자료 저장의 2가지 요구 사항을 만족시키기 위한 내장형 주기억장치 저장 시스템을 개발하였다.

1. 서론

최근 반도체 메모리 부문에서는 비약적인 발전이 이루어져 데이터베이스 기술에도 영향을 끼치고 있다. 이러한 기술은 많은 양의 데이터를 사용하는 인공 위성, 이동 단말기, 무인 로봇, 레이더 시스템 등의 실시간 시스템에서 현재 위치와 같이 검색 대상이 시간적으로 이동할 경우 빠른 응답을 위해 디스크 I/O가 많은 디스크 상주형 데이터베이스를 대체하기 위한 연구로 진행되고 있다. 실시간 시스템을 구축하기 위해 현재 많은 연구 기관과 대학에서 주 메모리 상주형 데이터베이스에 대한 연구가 활발히 진행중이고, 이미 일부에서는 그 결과를 실제 업무에 적용중이다.

주 메모리 상주형 데이터베이스 시스템은 데이터를 주기억장치에 상주시킴으로써 디스크 상주형 데이터베이스 시스템에 비해 높은 성능과 빠른 응답 시간을 제공해 준다. 그러나 데이터를 주기억장치에 상주시켜 처리하는 것이 디스크 상주형 데이터베이스 시스템에서 단순히 버퍼만을 크게 하는 것과

다르다. 디스크 상주형 데이터베이스 시스템에서의 버퍼는 데이터의 복사본을 가지는 보조적 역할밖에 하지 못하나, 주 메모리 데이터베이스 시스템의 경우는 데이터를 관리, 조직화할 수 있다. 따라서 이런 시스템은 명시된 데드라인 내에 작업이 완료되어야 하는 실시간 응용에 잘 적용될 수 있다.

본 논문에서는 기존의 일반 디스크 상주형 데이터베이스 시스템을 변경 없이 그 하부 구조로 사용하면서 검색과 연산의 속도를 높이고자, 내장형 주기억장치 자료 저장 시스템을 제안하고, 어느 정도의 성능이 더 개선될 수 있는지 기술한다.

2장에서는 관련 연구 및 다차원 데이터의 검색 방법에 대해서 설명하고, 3장에서는 주기억장치 자료 저장 시스템의 구조에 대해서 기술한다. 4장에서는 내장형 주기억장치 자료 저장 시스템의 구현에 대해 기술하고, 5장에서는 구현된 시스템의 실험 평가를 하고, 마지막으로 6장에서는 결론 및 향후 연구 방향을 기술한다.

2. 관련 연구

2.1 주 메모리 데이터베이스 시스템

주 메모리 데이터베이스 시스템은 주 물리적 메모리에 데이터를 저장하는 시스템이다. 디스크 상주형 데이터베이스에서 디스크에 저장된 데이터를 액세스하기 위해 메모리로 캐쉬하지만 주 메모리 데이터베이스 시스템은 주 물리적 메모리에 있는 데이터의 백업본을 디스크에 보관한다.

주 메모리 데이터베이스 시스템의 특징은 디스크 상주형 데이터베이스에 비해 높은 성능과 빠른 응답 시간을 제공한다.

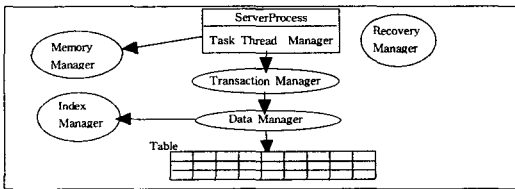
이미 발표된 시스템으로 스탠포드 대학의 STRIP (STandford Real-time Information Processor), Bell 연구소의 Dali 등이 있다.

2.2 다차원 데이터의 검색 방법

다차원 데이터를 검색하는 방법으로 R-Tree, 범위 검색(Range Query), Quadtree 등이 있다. 이중 R-Tree는 디스크 상주형 시스템에 적당한 색인방법이고, 주 메모리 상주형 시스템에서는 분포가 균일하지 않고, 데이터의 종류가 다양한 경우에 사용하는 색인 중 정형적인 공간 분할을 사용함으로써 구현하기 쉽고, 중간 노드에도 객체를 저장할 경우 객체를 분할하지 않아도 되는 Quadtree를 자주 사용한다. Quadtree를 이용하여 이웃 노드 찾기 연산을 수행할 경우, 탐색되어야 하는 링크가 많다는 것이 단점이 된다.

3. 주기억장치 자료 저장 시스템의 구조

3.1 주기억장치 자료 저장 시스템의 구성요소



[그림 1] 시스템의 구성요소

3.1.1 메모리 관리자(Memory Manager)

메모리를 할당하고 사용하지 않는 메모리는 회수한다. 최적의 메모리가 할당되어야 하는 것도 중요하다. 이 논문에서는 사용되고 난 메모리의 해체에 더 중점을 두고 시간 주기에 따라 실행되거나, 현재 메모리의 상태를 관리자가 파악한 뒤 사용하지 않거나 사용되어 더 이상 필요 없는 메모리의 경우 강제로 해제 할 수 있도록 설계되었다.

3.1.2 트랜잭션 관리자(Transaction Manager)

트랜잭션 ID객체는 구별되어야 하기 때문에 순서를 나타내는 것과 재실행 시를 위해 트랜잭션이 최초로 시작되어진 시간을 가지고 있다. 트랜잭션이 시작 메소드와 종료 메소드, 그리고 현재 상태를 가지고 있는 상태 메소드를 기본적으로 가지고 있다.

본 논문에서는 여러 개의 작업을 처리하지 않고, 하나의 SQL문을 처리하는 것을 하나의 트랜잭션을 처리하는 것으로 하였다.

3.1.3 데이터 관리자(Data Manager)

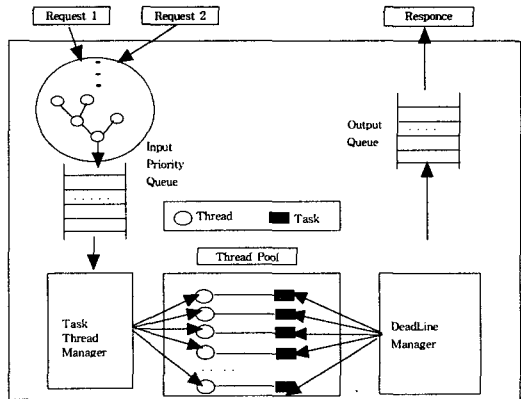
트랜잭션의 처리에 따라 할당되어진 메모리에 데이터를 갱신하거나 입력하거나 검색한다. 데이터 객체는 칼럼 객체와 테이블 객체로 이루어진다. 칼럼 객체는 테이블의 ID와 칼럼 이름, 칼럼 타입, 칼럼 ID 등의 정보를 가지고, 테이블은 이 칼럼 객체들의 리스트와 ID에 대한 정보와 테이블 타입의 정보를 유지하면서, 칼럼에 대한 정보를 참조한다. 칼럼과 테이블은 객체가 생성시 유일한 ID를 부여받아 중복된 객체가 생성되지 않도록 구별한다.

3.1.4 인덱스 관리자(Index Manager)

데이터 관리자의 요청에 따라 빠른 검색을 위한 인덱스를 생성, 관리한다. 기본적으로 지원하는 인덱스는 해쉬이다. 일반 정보의 경우에는 해쉬 인덱스를 사용하여 정보 검색을 속도를 높이고, 다차원 데이터의 경우에는 앞에서 살펴본 Quadtree로 구현한다.

3.1.5 회복 관리자(Recovery Manager)

회복 관리자는 독립적으로 수행되며 회복과 초기화 작업을 담당하며, 디스크 상주형 데이터베이스에 접속하여 디서너리 테이블을 참조하여 모든 정보를 가져온다. 데이터와 테이블 구조 등 모든 정보가 다



[그림 2] 서버 프로세스의 구조

스크 상주형 데이터베이스에 저장되어 있고, 주기억

장치 자료 저장시스템은 가장 최신의 자료들만을 필요로 하기 때문에 일괄처리가 가능해진다.

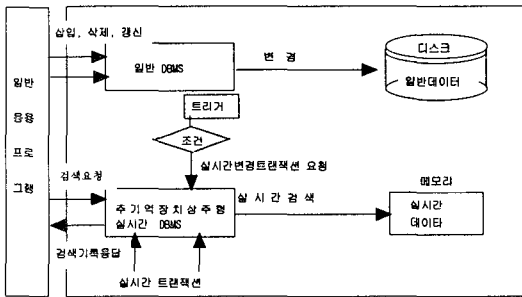
3.2 서버프로세스의 구조

서버 프로세스의 구조에서 일반 DBMS로의 변화 즉, 데이터의 갱신이나 검색의 요청이 발생하였을 경우, 트리거에 의해 조건을 검사하고, 조건을 만족하면 주기억장치 상주형 저장 데이터베이스 시스템으로의 실시간 트랜잭션이 수행된다.

각 트랜잭션에는 중요도에 따른 우선 순위와 데드라인이 부여되고, 입력 우선순위 큐(Input Priority queue)에 들어가게 된다. 우선 순위에 따라 TASK 쓰레드 관리자는 각 쓰레드에게 TASK를 할당하고 주어진 일을 처리한다. 여기서 쓰레드 풀(Thread Pool)은 멀티쓰레드를 지원하기 위함이다. 데드라인 관리자는 데드라인의 정보를 확인하여 데드라인내에 완료가 된 결과는 결과 큐(Output Queue)로 보내게 된다. 또 완료되지 못한 TASK의 경우에는 데드라인내까지 수행되어진 결과를 결과 큐(Output queue)로 보낸다.

4. 내장형 주기억장치 자료 저장 시스템 활용

4.1 전체 시스템 구성도

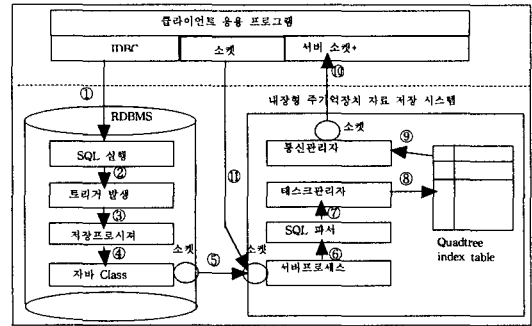


[그림 3] 전체 시스템 구성도

[그림 3]은 기존의 데이터베이스와 내장형 주기억장치 자료 저장 시스템을 활용한 전체 시스템 구성도이다. [그림 4]는 전체 시스템 구성에 따른 프로그램 흐름도이다.

실시간 데이터의 변경은 실시간 트랜잭션이 외부의 변화에 의해서 직접적으로 실시간 데이터베이스로 접근되는 것이 아니라, 먼저 디스크 상주형 데이터베이스에 저장되어 되고 그 내용의 변화가 실시간 데이터베이스의 내용을 변화할 만큼의 조건을 만족하면 주기억장치 상주형 변경 트랜잭션을 일반 데이터베이스가 트리거 기능을 이용하여 자체적으로 생성하여 수행하게 된다. 결국, 메모리에 상주하는 실시간 데이터는 검색조건을 만족하는 정보를 가진

데이터이다. 이 데이터를 검색하여 일반 응용 프로그램의 요구에 대해 응답을 취한다.



[그림 4] 프로그램 흐름도

4.2 시스템을 내장하기 위한 결합 모듈

주기억장치 자료 저장 시스템을 내장하기 위한 환경으로 데이터베이스 시스템은 오라클8i를 사용하였다. 이는 오라클 서버 플랫폼에 Javasoft JDK 1.1.6 표준의 자바 언어 스펙과 자바 가상 머신 스펙을 준수하는 100% 자바 실행 환경을 통합시킴으로써, 데이터베이스 환경 내에서 자바 소스파일, 클래스 파일 및 jar파일을 데이터베이스 내로 로딩하여 사용할 수 있도록 해준다.

이외에도 호스트 환경으로부터의 콜 인/ 콜 아웃(Call In/Call out) 모듈을 지원하는 데 본 논문에서는 이러한 기능들을 바탕으로 내장형 주기억장치 자료 저장 시스템과의 통신을 하여 요구에 대한 응답을 할 수 있도록 구현하였다.

오라클 데이터베이스와 내장형 주기억장치 자료 저장 시스템과의 통신은 오라클 데이터베이스내의 자바 클래스에서 소켓을 생성하여 주기억장치 자료 저장시스템의 대기중인 서버 프로세스에 접속을 하여 작업을 수행토록 한다.

4.3 트리거와 저장 프로시저

트리거란 데이터베이스가 미리 정해 놓은 조건을 만족하거나 또는 어떤 동작이 수행되면, 자동적으로 수행되는 동작을 말한다.

저장 프로시저는 일련의 SQL 명령문이 데이터베이스에 저장되어 특정 문제를 해결하거나 일련의 관련된 작업을 수행하는 하나의 단위로 실행된다.

본 논문에서 트리거는 테이블에 값이 삽입, 삭제, 갱신 시에 발생하게 되는데, 데이터의 갱신 시에는 한계치를 초과할 경우에만 내장형 주기억장치 자료 저장 시스템으로 데이터를 갱신토록 조건 검사를 하게 된다. 또 저장 프로시저를 호출하여 저장 프로

시저에서 자바 클래스를 호출하여 통신을 시도하며 세션이 설정이 되면 요구 사항을 보내게 되고 내장형 주기억장치 자료 저장 시스템에서는 서버 프로세스가 대기하다가 이 요청에 대해 적절한 타스크를 쓰레드에 할당하고 처리하도록 되어 있다.

5. 실험 평가

평가를 위한 방법으로는 범위 검색을 요청하였을 때 응답을 위한 검색시간에 대한 응답속도를 사용하였다. 그리고 메모리 사용시간, 네트워크의 트래픽에 따른 오차는 없는 것으로 가정하여 실험하였다.

2차원 데이터의 전체 영역을 100 * 100 으로 선택을 하고, 이 전체 영역에 포함되는 데이터의 수를 100개에서부터 1000개까지 100개씩 증가시켰고, 2차원의 검색 영역 범위는 10*10, 20*20 등으로 증가 시켜 가면서 실험을 하였다. 데이터의 값과 검색 영역 범위의 시작점은 모두 난수에 의해 발생되 어진 값을 사용하였다.

데이터수(X 100) \ 검색영역	1	2	3	4	5	6	7	8	9	10
10 * 10	1	1	1	1	1	1	1	1	1	1
20 * 20	1	1	1	1	1	2	2	2	2	2
30 * 30	1	1	1	2	2	2	2	3	3	3

[표 1] 검색 범위 크기별 데이터 검색 시간 평가 (단위 : 1/1000초)

결과 값은 각각 10회 측정하여 최대 값과 최소 값을 제외하고 질의에 대한 검색 시간을 구하면 표와 같다. [표 1]에서와 같이 검색 속도는 디스크 I/O를 하지 않고 메모리에서 모든 검색이 이루어지지 때문에 빠른 응답속도를 나타내었다.

6. 결 론

본 논문은 주 메모리 데이터베이스 시스템의 구조 및 응용분야의 한 예로, 다차원 데이터의 검색 방법에 대해 분석하였으며, 기존의 디스크 상주형 데이터베이스 시스템을 그대로 수용하면서, 주 메모리 데이터베이스 시스템의 높은 성능을 이용하기 위한 내장형 주기억장치 자료 저장시스템을 구현하고, 다차원 데이터의 검색 시 디스크 상주형 데이터베이스 시스템보다 얼마만큼 더 성능이 개선될 수 있는지 평가해 보았다. 내장형 주기억장치 자료 저장 시스템은 객체 지향적이며, 플랫폼, 운영체제와 독립적인 자바 언어로 구현되어 이식성이 뛰어나며, 모듈화되어 있어 실시간 시스템과 같은 높은 성능을 요구하

는 분야에서 기존의 시스템을 크게 수정하지 않고도 통신 모듈만을 구현하고, 본 논문에서 구현한 시스템을 내장하여 적합하게 이용될 수 있는 것이 장점이다.

본 논문에서 구현된 내장형 주기억장치 자료 저장 시스템은 주메모리 데이터베이스 시스템을 기본 설계로 하였으나, 메모리에 대한 많은 자원을 차지하게 되므로 좀 더 향상된 시스템을 구현하려면 다음과 같은 사항을 더 보완해야 한다.

먼저, 메모리 최적화 부분이다. 구현된 시스템은 메모리 최적화보다는 사용되어진 메모리를 시스템에 반납하도록 하는 부분에 중점을 두었다. 최적화된 메모리의 사용은 그만큼 더 자원을 활용할 수 있다는 이점이 있다. 그리고 구현된 시스템에서 기존의 디스크 상주형 데이터베이스 시스템과의 통신 모듈의 오버헤드부분이다. 소켓을 사용하여 구현된 통신 모듈은 블로킹 현상에 대한 해결책을 구현하는 부분이 더 보완되어 져야 한다. 이 외에도 다양한 플랫폼, 운영체제 하에서 비교분석이 이루어져야 한다.

참고문헌

[1] Adelberg,B., Kao,B., Garcia-Molina,H., "Overview of the Stanford Real-Time Information Processor(STRIP)" SIGMOD RECORD, 25(1), 1996.
 [2] DeWitt,D. J., Katz,R. H., Olken,F., Shapiro,L. D., Stonebraker,M. R., and Wood,D., "Implementation Techniques for Main Memory Database Systems," in Proc. ACM SIGMOD Conf., June, 1984.
 [3] Garcia-Molina,H., Salem,K., "Main Memory Database Systems : An Overview.", IEEE Transactions on Knowledge and Data Engineering, pp. 50 9-516, Dec, 1992.
 [4] Guttman,A., "R-trees: A Dynamic Index Structure For Spatial Searching", Proc. ACM SIGMOD Inc. Conf. on Management of Data, pp47-57, 1984.
 [5] Jagadish,H. V., Lieuwen,D., Rastogi,R., and Silberschatz,A., "Dali : A High Performance Main Memory Storage Manager." in Proceedings of the 20th VLDB Conference,(Santiago, Chile), 1994.
 [6] Kao,B., Garcia-Molina,H., "An Overview of Real-Time Database Systems", in Advances in Real-Time Systems, pp. 463-486, Prentice Hall, 1995.