임베디드 시스템 상에서의 고속 트랜잭션을 위한 메인메모리 기반 데이터베이스 시스템 구현

김영환[12], 손재기[1], 박창원[1]
전자부품연구원 지능형정보 연구센터
e-mail : *yhkim93, jgson, parkcw@keti.re.kr*

Implementation of Maim Memory DBMS for Efficient Transactions based on Embedded System

Young-Hwan Kim[12], Jae-Gi Son[1], Chang-Won Park[1]

[1]Intelligent IT Research Center
Korea Electronics Technology Institute

[2]Embedded Operating Systems Lab., School of Computer
SoongSil University

Abstract

Mani Memory DataBase(MMDB) system store their data in main physical memory and provide very high-speed access. Conventional database system are optimized for the particular characteristics of disk storage mechanism. Memory resident systems, on the other hand, use different optimizations to structure and organize data, as well as to make it reliable.

This paper provides a brief overview on MMDBs and the results after evaluating the performance of our simple MMDB based on Embedded system.

I. 서론

MMDB 기술은 현재까지 빠른 시간 동안 급성장해 다양한 분야에서 사용되고 있지만 아직 일반화되지는 못한 상황이다. 또 기술 및 기능에 있어서도 기존 디스크 DB에 비해 아직 미흡한 점이 있는 것도 사실이다. 그러나 최근에 MMDB에 관심이 크게 늘고 있는데, 이는 기존 네트워크 장비 및 속도의 발전과 온라인 환경의 대중화로 인해 빠른 트랜잭션 처리에 대한 사용자의 요구가 증대되고 있기 때문이다.

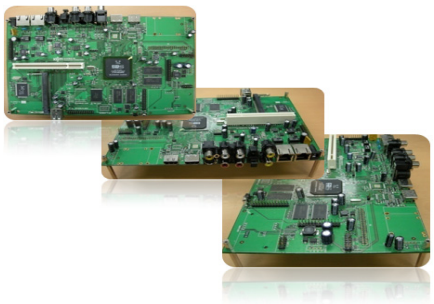
일반적으로 MMDB는 셋톱박스, 산업용제어, 가전기기 등과 같이 특수 목적의 소형 하드웨어시스템에 적용될 수 있는데 이는 데이터의 고속 처리에 초점을 맞추고 있다. 이 때 적용되는 메모리 DBMS는 범용 DBMS보다 사이즈가 작아 꼭 필요한 기능만을 제한적으로 제공하게 되는데 이와 같은 목적으로 사용되는 소용량 메모리 DBMS 제품을 임베디드 메모리 DBMS 또는 플랫폼 내장형 DBMS라고 한다. 본 논문에서 사용된 임베디드 시스템 또한 IP 셋톱박스에서 사용되는 임베디드 플랫폼으로 Sigma Designes 사의 MISP 계열인 SMP8634 미디어 프로세서를 사용하고 있다.

본 논문에서는 이와 같은 임베디드 시스템 상에서 단순한 데이터베이스 기능만을 제공하는 MMDB를 구현하여 다양한 성능평가 파라미터를 통한 실제 임베디드 시스템 상에서의 MMDB의 성능을 평가한다.

II. 본론

2.1 임베디드 시스템

본 논문에서 사용된 플랫폼의 메인 프로세서는 SMP 8634 200MHz 미디어 프로세서로 메인메모리는 64MB, 64MB의 부트 플래시를 사용한다. 이와 같은 제한된 리소스를 가진 임베디드 시스템에 많은 I/O 및 고속 트랜잭션을 요구하는 응용에 적합한 MMDB를 구현함으로 실제 MMDB에 성능을 평가할 수 있을 것이다.



[그림 1] SMP8634 임베디드 플랫폼

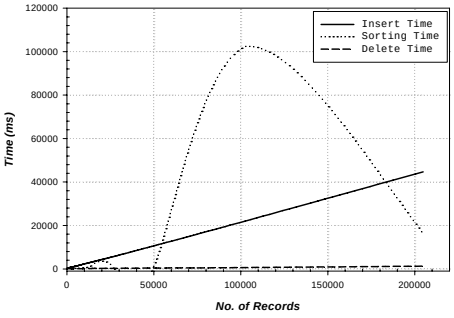
2.2 Simple MMDB

위의 임베디드 시스템에서 사용될 Simple MMDB의 특징은 SQL Syntax와 유사한 Query 언어를 지원하고, C++ 인터페이스를 포함하고 있어 이를 이용한 DB기반의 응용 개발에 있어 편리성을 제공한다. 또한 16바이트의 Chunk 사이즈로 메모리를 할당하며, 할당된 메모리는 Bitmap 테이블을 통해 관리된다. 메모리상에서 직접 데이터 처리가 가능함으로 실시간성을 보장하며, 시스템의 다운 시 자동 복구 및 온라인 백업 기능을 포함하고 있다. 이와 같은 특징을 통해 기존의 디스크 기반의 데이터베이스 시스템보다 향상된 성능을 나타낸다. 다음은 Simple MMDB에서 제공하는 SQL 명령어를 정리한 것이다.

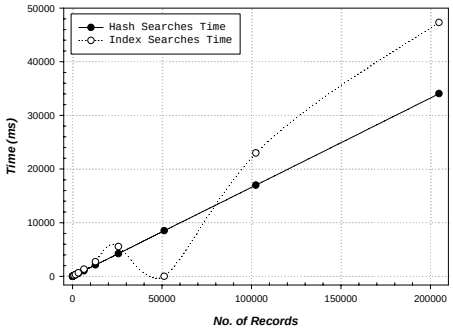
III. 성능평가

임베디드 시스템 상에서 Simple MMDB의 성능평가를 위한 metric은 다양한 개수의 레코드를 생성함으로서 Insert, Hash Search, Index, Sorting, Delete Time을 확인하게 된다. 성능평가를 위해 생성 레코드 수를 입력 받아 각각의 metric에 대한 결과를 ms(밀리초) 단위로 나타낸다. 아래의 코드는 각각의 metric에 대한 시간을 확인하기 위해 사용된 코드이다.

```
start = clock();

[Insert, Hash Search, Index Search, Sorting, Delete]

end = clock();
msec = 1000.0 *(end-start) / CLOCKS_PER_SEC;

printf("[Insert, Hash Search, Index Search, Sorting, Delete] Time
                %d : %10.0fms₩n", 레코드 수, msec);
```



[그림 2] Insert, Sorting Delete metrics



[그림 3] Hash, Index Searches

Ⅳ. 결론 및 향후 연구 방향

본 논문에서는 최근 임베디드 분야와 DB분야에서 새로운 기술로 주목받고 있는 임베디드 기반의 MMDB를 개발하여 실제 임베디드 시스템 상에서 성능평가를 수행하여 그 결과를 보여주었다. 아직 완전한 성능을 보이기에는 다소 수정할 부분이 있다. 특히 delete metric에서 레코드 수에 따른 latency의 변화가 매우 심하다는 것을 알 수 있는데 이 부분도 향후 디버깅을 통해 해결해야할 부분이다. 또한 메모리와 디스크 간의 스왑은 전체 MMDB의 성능을 떨어뜨리고 있어 이 부분에 대한 알고리즘 개선이 필요하며 시스템 다운의 경우 신속·정확한 복구를 위해 Shadow paging 메커니즘에 대한 수정도 필요할 것으로 생각된다. 마지막으로 임베디드 시스템 상에서의 디스크 기반 DB와의 성능평가는 향후 Simple MMDB의 최적화 이후 수행할 계획이다.

참고문헌

[1]F.Raja, M.Rahgozar, N.Razavi, and M.Siadaty, "A Comparative Study of Main Memory Databases and Disk-Resident Databases", PWASET Volume 14, August 2006.

[2] Inseon Lee, Heon Y.Yeon, Taesoon Park, “"A New Approach for Distributed Main Memory Database Systems: A Causal Commit Protocol”", IEICE Trans. Inf. & Syst., Vol.ES7, No.1 January 2004.