

# 모바일 DBMS를 위한 효율적인 압축 데이터 동적 관리 시스템\*

신영재, 장진근, 이정화, 김학수, 박찬희, 손진현  
한양대학교 컴퓨터공학과  
e-mail:yjshin@cse.hanyang.ac.kr

## Development of the Efficient Compressed Data Dynamic Management System For Mobile DBMS

Youngjae Shin, Jinkun Jang, Jeong Wha Lee, Hak Soo Kim,  
Chan-Hee Park, Jin Hyun Son  
Dept. of Computer Science & Engineering, Hanyang University

### 요 약

최근 사회는 정보의 디지털화로 인해 많은 디지털 정보들이 많아지고 있고, 또한 휴대용 정보기기의 발전으로 인해 휴대용 정보기기의 사용이 보편화 되었다. 이로 인해 휴대용 정보기기에서는 정보들을 효과적으로 관리하기 위해 임베디드 DBMS의 사용이 요구되고 있다. 또한 휴대용 정보기기에서 보편적으로 사용되는 저장장치는 NAND형 플래시 메모리로 단위 공간당 비용이 기존의 하드디스크에 비해 수십배 가량 높아 저장 공간의 효율적인 관리가 요구되고 있다. 따라서 본 논문에서는 플래시메모리를 저장매체로 사용하는 DBMS에서 압축기법을 사용한 효율적인 데이터 동적 관리 시스템을 제안한다. 제안되는 압축 기반 시스템은 저장 공간과 데이터I/O를 줄이며, 데이터I/O를 줄임으로써 DBMS의 성능향상과 플래시 메모리의 수명을 연장시키는 효과를 기대할 수 있다.

### 1. 서론

현대의 휴대용 정보기기는 기본적인 정보의 처리, 저장 수단으로만 사용하던 것에서 높은 휴대성과 광범위한 정보를 저장, 처리할 수 있는 기기로 바뀌고 있다.[1] 이러한 환경의 변화로 휴대용 정보기기에서는 보다 많은 정보의 생성, 처리, 저장이 가능한 다양한 응용프로그램의 사용을 요구 받고 있다.[2] 이로 인해 많은 데이터의 효율적인 관리 시스템이 필요로 하게 되었다. 그래서 휴대용 정보기기에서 정보를 쉽게 접근하여 처리하고 갱신할 수 있도록 구성된 데이터의 집합체인 데이터베이스의 사용이 필요하게 되었다. 세계적인 분석기관인 IDC(International Data Corporation)社は 2004년도에 64%이상의 휴대용 정보기기의 어플리케이션이 임베디드 데이터베이스 관리 시스템이 필요하다고 보고하였다.[3] 하지만 이러한 데이터베이스를 저장해야 할 모바일 컴퓨팅 환경은 높은 휴대성과 내구성, 저 전력소모를 요구한다. 따라서 가격이 저렴하지만 전력을 많이 소모하며 크기, 소음, 진동 등의 단점을 가진 하드디스크의 사용이 어렵게 되었다. 그래서 모바일 정보기

기에서는 이러한 단점을 보완할 수 있는 저전력으로 장시간의 구동이 가능하며, 부피가 작고 경량으로 소음과 진동이 없으며 물리적인 충격에 강해 휴대가 용이한 플래시 메모리를 보조기억장치로 사용한다.[4] 하지만 이 플래시 메모리는 기존의 하드디스크에 비해 고비용과 데이터 I/O에 따른 수명을 가진다는 단점이 있다. 이로 인해 데이터의 효율적인 관리가 필요하다.[1, 3]

본 논문에서는 모바일 정보기기에 데이터베이스의 데이터를 효율적으로 저장하는 방법으로 효율적인 압축 데이터 동적 관리 시스템인 CDDMS (Compressed Data Dynamic Management System)를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 데이터의 압축과 기존 압축데이터 관리기법, 그리고 플래시 메모리에 대한 관련연구를 살펴본다. 3장에서는 압축 데이터 동적 관리 시스템인 CDDMS의 구조와 두가지의 압축 데이터 관리기법을 설명한다. 4장에서는 제안하는 CDDMS의 관리기법의 성능을 실제 DBMS와 연동하여 평가한다. 마지막으로 5장에서는 논문의 결론을 맺는다.

### 2. 관련 연구

이 장에서는 데이터베이스에서 데이터의 압축과 기존

\* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 육성·지원 사업(IITA-2006-C1090-0603-0031)의 연구결과로 수행되었음

압축데이터 관리기법, 그리고 플래시 메모리에 대한 관련 연구를 살펴본다. 이것은 이 논문에서 제안하는 압축 데이터 동적 관리 시스템(CDDMS)에 대한 기본 지식 및 제작 동기가 된다. 2.1절에서는 데이터 압축 및 기존 압축데이터 관리기법에 관해서 알아보고, 2.2절에서는 플래시 메모리의 특성에 대해서 알아본다.

### 2.1 데이터베이스에서 데이터 압축 및 관리

압축된 데이터를 사용하여 시스템의 효율을 높이고자 하는 연구는 현재까지 지속적으로 연구되어 왔다. 이와 관련된 연구는 크게 데이터 압축방식과 압축 데이터의 처리로 나눌 수 있다. 이 중 대부분의 연구는 데이터 압축방식에 대한 연구로써[5], 이와 같은 연구에서 데이터 압축기법을 DBMS에서 사용함으로써 물리적 디스크공간의 절약이다. 데이터베이스의 데이터를 압축 저장함으로써 데이터의 양을 줄일 수 있고 이로 인해 물리적인 저장소의 디스크 공간을 절약할 수 있게 되는 이점과 디스크 I/O 감소의 이점이 있다고 밝히고 있다. 이러한 이점에도 불구하고 실제로 데이터 압축방식은 대부분의 고성능의 DBMS에서 전형적으로 사용되고 있지 않다. 사용되더라도 특정 DBMS에 특화된 방식으로 적용되어 사용되고 있다.[6]

그러나 이러한 기존 연구들은 DBMS 내부에서 데이터 압축을 통해 시스템의 효율을 높이기 위한 연구로, 압축 알고리즘과 압축단위로써 테이블의 필드 혹은 열을 사용하는 것에 초점이 맞추어져 있었다. 또한 기존 연구들은 압축된 데이터의 저장방식에 대해 약간의 언급은 있었으나 그로인해 생겨나는 빈공간의 처리등에 관한 최적의 관리방식을 보여주지 못했다.

따라서 본 논문에서는 기존 DBMS의 수정을 최소화 할 수 있도록 입출력 page단위를 이용한 데이터 압축 구조를 설계함으로써 특정 DBMS에 비종속적 데이터 압축 구조를 제안하고, 효율적인 압축된 데이터의 관리기법을 제안하여 압축 효과를 증대시키도록 하였다.

### 2.2 플래시 메모리의 특성

플래시 메모리는 저전력으로 장시간의 구동이 가능하며, 부피가 작고 가볍다. 또한 소음과 진동이 없으며, 물리적인 충격에 강해 휴대가 용이하다.[4] 또한 플래시 메모리가 RAM에 비해 가격이 저렴하고, HDD보다 성능이 뛰어나다. 이러한 특성은 플래시 메모리가 비싸면서 대용량 저장매체로 사용하기 힘든 휘발성인 RAM과 여러 가지 단점으로 모바일 기기에 사용하기 힘든 하드디스크의 단점을 보완한 저장매체인 것을 보여주고 있다. 따라서 이러한 특성으로 인해 현재 모바일 기기들에 가장 보편적으로 NAND형 플래시 메모리가 사용되고 있다.

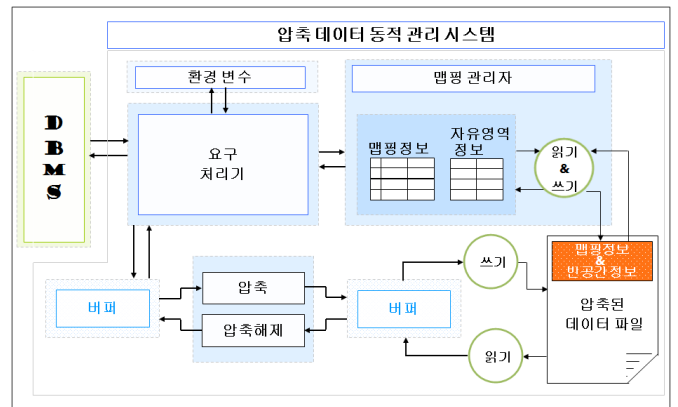
이러한 특성을 가지는 플래시 메모리는 위에서 살펴본 바와 같이 하드디스크보다 비용이 많이 든다. 따라서 비용의 문제로 많은 저장공간을 확보할 수 없게 된다. 이로 인해 본 논문에서 제안하는 CDDMS의 필요성이 대두된다.

## 3. 압축 데이터 관리기법

이 장에서는 본 논문이 제안하는 압축 데이터 관리기법에 대해 기술한다. DBMS와 파일시스템 사이에서 사용하는 압축 데이터 동적 관리 시스템(CDDMS, Compressed Data Dynamic Management System)을 4.1절에서 제안하고 CDDMS에서 압축 데이터를 관리하는 기법에 대해 4.2절에서 기술한다.

### 3.1 CDDMS 구조

모바일 정보기기에서 발생하는 저장공간 부족문제는 데이터를 압축함으로써 해결할 수 있다. 본 논문에서는 데이터 압축기법을 사용하기위해 CDDMS(Compressed Data Dynamic Management System)을 사용한 새로운 구조를 제안한다. 본 연구에서 제안하는 CDDMS는 기존 DBMS의 변경 최소화, 데이터의 무결성 보장, 효율적인 공간 관리라는 설계 목표를 가진다.



(그림 1) CDDMS의 구조

기존의 시스템은 데이터베이스를 관리하기 위해 데이터 파일을 읽고 쓰는 것에 대한 요청을 파일시스템에 직접 하였다. 하지만 본 논문이 제안하는 CDDMS는 기존 DBMS의 변경 최소화를 위해 DBMS의 스토리지 관리자와 파일 시스템 사이에 들어가게 된다. (그림 1)에서 보듯이 DBMS는 기존에 파일시스템으로 보내던 데이터를 CDDMS에게 보내게 되고 이것을 CDDMS가 압축하여 파일시스템에 보내게 된다. 이러한 CDDMS는 DBMS가 요청한 데이터를 압축된 데이터 파일에서 찾기 위해 CDDMS는 데이터파일 내부에 매핑정보를 포함시키고 있다. 이 매핑정보는 논리적 블록번호, 압축된 블록의 위치나 크기 정보를 가지게 되며, 이러한 매핑정보를 이용해 압축되지 않은 상태의 데이터파일과 압축된 데이터파일의 매핑이 가능해진다. 이때 매핑정보에 포함된 블록번호는 압축되지 않은 상태의 데이터파일을 논리적 블록의 크기로 분할했을 때의 할당된 번호이다. 또한 CDDMS는 데이터파일 내부에 자유영역정보(Free Space Information)를 포함시키고 있다. 이 자유영역정보는 데이터의 무결성을 보장하기 위한 처리로 인해 발생한다. 구체적으로 설명하면 압축된 블록에 포함된 데이터가 DBMS에 의해 갱신되어 다시 쓰기를 수행할 때 CDDMS에 의해 해당 데이터

가 포함된 블록은 다시 압축되어 데이터 파일 내에 저장하게 된다. 이때 다시 압축된 블록은 갱신되기 전 압축된 상태의 블록크기와 동일한 크기를 가진다는 것을 보장할 수 없다. 이러한 이유로 압축된 블록이 갱신되어 다시 기록되어야 할 때는 기존위치가 아닌 파일내 새로운 공간에 저장되어야 하고, 압축된 블록이 존재하던 기존위치는 새로운 압축 블록의 쓰기를 위해 저장 가능한 공간으로 표시되어야 한다. 이렇게 파일내부에 발생하는 저장가능공간의 정보를 자유영역정보로 가지고 있게 된다.

CDDMS는 압축된 블록에 대한 쓰기를 진행할 때 자유영역정보를 이용해서 발견된 자유영역에 압축된 블록을 저장하게 된다. 저장하는 블록은 매핑정보에 추가함으로 CDDMS는 매핑정보와 자유영역정보를 활용해 DBMS가 요구하는 데이터에 대한 검색과 저장이 가능하다. 기본적인 DBMS의 데이터 읽기 쓰기 명령에 대한 CDDMS의 처리절차는 <표 1>에서 나타난 바와 같다.

<표 1> CDDMS를 사용한 DBMS의 데이터읽기, 쓰기 요청처리 절차

DBMS의 데이터읽기 요청	DBMS의 데이터쓰기 요청
①요청된 데이터를 포함하는 압축된 블록의 위치 검색	①요청된 데이터 압축
②검색된 위치의 블록 읽기	②저장 가능한 자유영역 검색
③읽어진 블록의 압축해제	③검색된 자유영역에 압축된 블록 쓰기
④압축 해제 된 데이터를 DBMS에 반환	④사용된 자유영역 삭제
	⑤매핑정보 갱신
	⑥자유영역정보 추가 (데이터 갱신시)

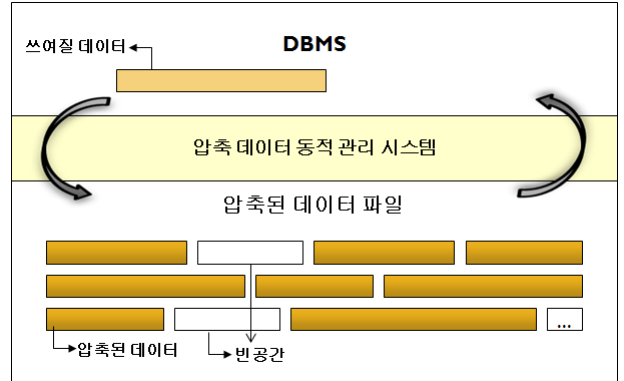
<표 1>에서 나타난 CDDMS의 데이터쓰기 절차는 데이터 무결성을 보장하기 위한 처리절차이다. 다시 말해 어떠한 상황에서 비정상 종료로 하더라도 데이터파일의 손상과 완료되기 전의 데이터를 보호할 수 있다. 이것은 먼저 갱신이 요청된 데이터를 기존의 구역이 아닌 다른 구역에 저장하기 때문에 가능하다. 또한 압축된 블록의 저장이 완료된 후 매핑정보를 갱신함으로 매핑정보를 갱신하고 난후 블록의 저장시 발생하는 오류를 해결해 준다. 그리고 자유영역의 추가를 최대한 뒤쪽에 배정함으로 자유영역이 손실되더라도 데이터에는 문제가 없도록 함으로 데이터의 무결성을 보장한다.

이렇게 무결성을 보장하면서 저장한 데이터는 압축률에 따라 크기가 각각 다르다. 크기가 다른 압축된 데이터로 인해 기존 시스템처럼 정해진 규격에 따른 관리가 어렵게 된다. 따라서 크기가 다른 압축된 데이터들의 효율적인 관리를 위해 본 논문에서는 여러 가지 관리기법을 제안한다. 이 관리기법을 다음 장에서 설명한다.

**3.2 CDDMS의 압축데이터 관리 기법**

CDDMS의 압축 데이터 관리기법은 (그림 2)에서 보듯이 압축률에 따라 달라지는 압축된 데이터의 크기에 따

라 동적으로 처리하는 방식이다. 이 방식은 세부적으로 2가지로 다시 나뉘어진다. 먼저 첫 번째로 압축된 블록이 나뉘지 않고 한곳에 저장되는 방식을 제공하고 두 번째로 빈 공간이 압축된 블록보다 작더라도 앞쪽의 빈 공간부터 채우고 나머지는 또 다른 빈 공간에 넣어서 나뉘진 블록을 링크로 연결하는 방식을 제공하고 있다.



(그림 2) CDDMS의 데이터 이동

첫 번째 방식은 연속공간방식으로 CDDMS가 데이터 쓰기의 요청을 받았을 때 그 데이터의 압축된 블록이 저장하고자 하는 공간은 무조건 블록 전체가 모두 들어갈 수 있는 크기의 공간이면서, 다른 공간보다 잘 맞아 떨어지는 공간에 블록을 쓰게 된다. 이러한 방식은 압축된 데이터가 연속된 공간에 저장됨으로 한번의 읽기로 데이터를 읽어 올수 있는 장점이 있다 하지만 항상 압축된 데이터 전체가 들어가지 않으면서 빈 공간이 많이 생길 수 있다. 이 점을 보완하기 위해 데이터 전체가 모두 들어가지 못해도 나눠서 앞쪽부터 채워 넣는 두 번째 방식을 제안한다.

두번째 방식은 최소공간방식으로 압축된 데이터들에 맞는 공간만을 찾다가 빈 공간이 많이 생기는 것을 제거하고자 압축된 데이터를 나눠서 앞쪽의 빈 공간부터 채워가는 방식이다. 연속된 공간에 써야 할 데이터가 처음에 연속된 공간을 만난다면 그곳에 연속적으로 쓸 수 있지만 그렇지 않다면 먼저 빈공간의 크기만큼 쓰고 나머지는 다음 빈공간에 쓰게된다. 이 때 나뉘지는 데이터는 나뉘진 블록에 대한 정보를 자체적으로 보유하게 된다. 예를 들어 압축된 데이터가 3개의 블록으로 나뉘었다면, 맵핑정보에는 첫 블록의 시작위치와 그 블록의 시작위치를 가지고 있다. 첫 블록은 다음 블록의 시작위치 정보와 다음의 블록 크기를 가지고 있다. 두 번째 블록 또한 다음 블록의 정보를 가지고 있고, 마지막 블록인 세 번째 블록은 압축된 데이터의 전체 크기를 가지고 있게 된다. 이러한 방식을 사용함으로써 파일의 빈 공간을 앞에서부터 차례로 채워 나가기 때문에 빈 공간을 획기적으로 줄일 수 있다. 따라서 파일 용량의 효율적인 관리가 가능함으로 공간효율성이 높아지는 장점이 있다.

본 논문이 제안하는 각각의 방식은 장단점을 가지고 있다. 연속공간방식은 읽기/쓰기 연산에서 한번의 파일 읽기

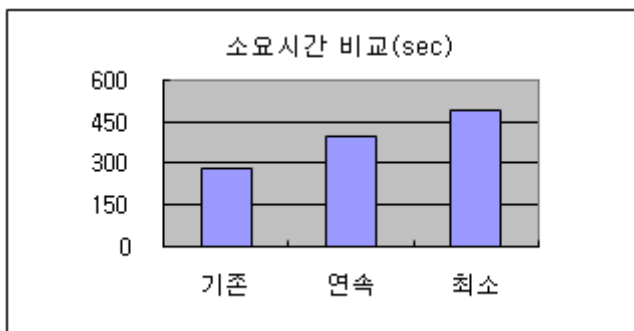
/쓰기만으로 처리된다. 이것은 파일 읽기/쓰기의 초기화 시간이 있는 것을 감안한다면 시간 면에서 좋은 효과를 얻을 수 있을 것이다. 하지만 이 방식은 쓸 블록보다 큰 빈공간을 찾는데, 블록이 계속적으로 일정 크기 이상의 것들이 들어온다면 그 크기 이하의 빈공간은 계속적으로 사용되지 못하는 문제가 생길 수도 있게 된다. 이에 반해 최소공간방식은 파일의 앞쪽 빈공간부터 쓰기 때문에 차례로 채워진다. 이것은 파일을 효율적으로 사용할 수 있게 해준다. 하지만 작은 빈공간이 여러개 생성되었을 때, 하나의 큰 데이터가 나누어 저장되는 방식의 특성 때문에 블록이 항상 연속적으로 저장되지 않는다. 따라서 두 개 이상의 블록에 걸쳐 있는 데이터는 파일 읽기/쓰기를 여러번 할 수밖에 없게 된다. 또한 데이터의 분할을 처리하기 위해 각각의 블록은 링크 정보를 가지고 있어야 하는 단점이 있다.

이러한 장단점을 바탕으로 고려해 볼 때 연속공간방식은 파일의 낭비를 감수하고도 좀 더 나은 속도를 필요로 하는 환경에 적합하고, 최소공간방식은 속도보다는 파일의 효율적인 사용을 필요로 하는 환경에 적합하면서 압축된 블록의 크기가 비슷할 경우 사용하기에 적합하다.

#### 4. 성능 평가

제안하는 CDDMS를 평가하기 위하여 이엠웨어(주)의 MobileLite -Linux 환경을 사용하였다. 이 실험을 통해 기존 DBMS와 CDDMS를 사용한 DBMS의 차이를 확인하고 나아가 Mobile 환경인 플래시 메모리상에서의 결과를 예상 할 수 있도록 하였다.

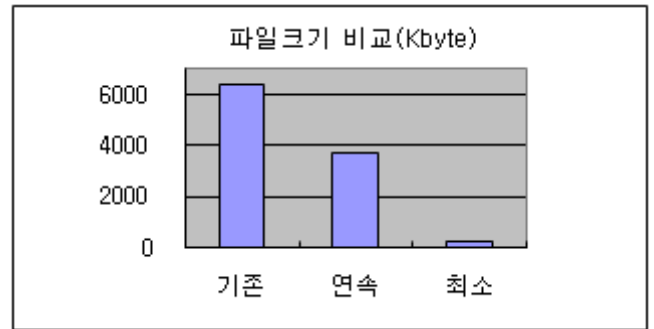
평가를 위해 DBMS의 성능평가를 하기 위한 프로그램을 사용하여 CDDMS를 사용한 DBMS의 성능과 사용하지 않은 DBMS의 성능을 비교하였다. 이 때 사용된 프로그램은 이엠웨어(주)의 MobileLite를 테스트하기 위해 만들어진 프로그램으로써 여러 가지 SQL문을 돌려보면서 결과를 확인하는 프로그램이다.



(그림 3) 기존방식과 CDDMS방식의 소요시간 비교

(그림 3)은 동적 관리의 첫 번째 방식과 두 번째 방식 그리고 기존 시스템의 소요시간을 비교 한 것이다. 이 실험을 통해 속도 면에서 기존방식보다 30%이상의 추가 시간이 필요하였다. CDDMS를 사용한 DBMS의 속도가 크게 나지 않은 것을 알 수 있다.

또한 (그림 4)에서 보이듯이 데이터 파일크기 비교에서 기축방식보다 50%정도 적은 공간을 사용하여 CDDMS의 사용이 우수함을 알 수 있다. 이것은 각 관리방식에 대한 설명이 있는 3장에서의 예측한 결과와 동일한 결과이다.



(그림 4) 기존방식과 CDDMS방식의 파일크기 비교

#### 5. 결론

유비쿼터스 모바일 정보기기에서 내장형 DBMS가 관리하는 데이터 저장 공간을 최소화하기 위해, 본 논문에서는 플래시 메모리를 위한 압축을 이용한 DBMS의 효율적인 데이터 동적 관리를 제안하였다. 또한 여러 가지 방식으로 나누어, 적용 환경의 특성에 따라 더 효율적인 관리를 할 수 있도록 설계하였다. 이 방법은 데이터베이스가 압축되지 않은 상태로 저장되는 현재 상황을 고려해 볼 때 획기적인 저장 공간의 절약 가져온다. 또한 이 방법은 디스크에 써야 되는 데이터를 줄임으로 디스크 I/O를 줄여준다. 이것은 제한적인 수명을 가지는 플래시 메모리의 수명을 연장시키는 효과를 가져온다.

#### 참 고 문 헌

- [1] K.S. Yim, H.K. Bahn, K. Koh, "A Flash Compression Layer for SmartMedia Card Systems", IEEE Transactions on Consumer Electronics, Vol.50 No.1, 2004
- [2] Mark A. Roth, Scott J. Van Horn "Database compression", ACM SIGMOD Record, 22:3, p.31-39, 1993
- [3] 변시우, 노창배, 정명희, "휴대용 정보기기를 위한 플래시 기반 2단계 로킹기법", 한국데이터베이스학회: 정보기술과 데이터베이스 저널 제12권 4호, 2005
- [4] M.-L. Chiang, R.-C. Chang "Cleaning policies in mobile computers using flash memory", Journal of Systems and Software, Vol.48, No.3, p.213-231, 1999
- [5] Till Westmann, Donald Kossmann, Sven Helmer, Guido Moerkotte "The implementation and performance of compressed databases", ACM SIGMOD Record, Vol.29 No.3, p.55-67, Sept. 2000
- [6] Balakrishna R. Iyer, David Willhite, "Data Compression Support in Databases", Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994
- [7] G.V. Cormack. "Data compression on a database system", Communication of the ACM, 28:12, 1985.