

혼합 워크로드 처리를 위한 컬럼 그룹 기반 동적 페이지 저장 관리 설계

박경현^{1,2} · 원희선¹ · 류근호^{2*}

¹한국전자통신연구원 스마트데이터연구그룹

²충북대학교 데이터베이스/바이오인포매틱스 연구실

A Design of Column-Group based Dynamic Page Storage Model for Mixed Workloads

Kyoung Hyun Park^{1,2} · Hee Sun Wonk¹ · Keun Ho Ryu^{2*}

¹Smart Data Research Group, ETRI, Daejeon, 34129, Korea

²*Database/Bioinformatics Lab., Chungbuk National University, Cheongju, 28644, Korea

[요약]

기존의 데이터베이스시스템은 정적 페이지 저장 모델을 사용하기 때문에 최근 데이터베이스 시장에서 요구하는 혼합 워크로드를 효율적으로 처리하기에는 한계를 가진다. 이에 본 논문에서는 혼합 워크로드의 특성을 반영할 수 있는 동적 페이지 저장 모델을 소개한다. 또한 혼합 워크로드를 분석하여 최적화된 컬럼 그룹을 추출한 후 동적으로 페이지를 구성하는 방법을 소개한다. 마지막으로 실험을 통해 본 논문에서 제안한 컬럼 그룹 기반의 동적 페이지 저장 모델이 기존의 페이지 저장 모델과 비교할 때 혼합 워크로드를 처리하는데 보다 효율적임을 보인다.

[Abstract]

There exists a limit in efficient processing of mixed workloads that database markets requires in recent years since existing database systems utilize a static page storage model. In this paper we propose a dynamic page storage model that can reflect the characteristics of mixed workloads. We also describe how to extract optimized column groups from given mixed workloads and how to construct pages dynamically. Finally, we show in our experiments that the proposed model is more efficient than the existing model in processing given mixed workloads.

색인어 : OLTP/OLAP, 페이지 저장 모델, 혼합 워크로드, 비용모델, 컬럼 그룹 선택 알고리즘

Key word : OLTP/OLAP, Page Storage Model, Mixed Workload, Cost Model, Column Group Selection Algorithm

<http://dx.doi.org/10.9728/dcs.2018.19.2.335>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 07 January 2018; Revised 10 February 2018

Accepted 26 February 2018

*Corresponding Author; Keun Ho Ryu

Tel: +82-43-267-2254

E-mail: khryu@chungbuk.ac.kr

I. 서론

데이터베이스 시장은 전통적으로 데이터 저장 구조에 따라 레코드 기반의 데이터 스토어와 컬럼 기반의 데이터 스토어로 양분되어 발전되어 왔다. 레코드 기반의 데이터 스토어는 레코드 단위로 데이터를 저장하기 때문에 트랜잭션 처리에 중점을 두는 OLTP(Online Transaction Processing) 워크로드에 적합하고 대부분의 관계형 데이터베이스가 여기에 속한다. 반면 컬럼 기반의 데이터 스토어는 데이터하우스와 같이 대용량의 데이터를 대상으로 특정 컬럼 또는 컬럼 그룹에 접근하여 데이터를 처리, 분석하는 OLAP(Online Analytical Processing) 워크로드에 더 적합한 특성을 보인다.

하지만 최근 데이터베이스 시장이 실시간 데이터 및 빅데이터 관리 및 분석 시장으로 발전하면서 워크로드의 구분이 모호해지기 시작하였다. 이와 같이 현재의 데이터베이스 시장이 특정 워크로드가 아닌 OLTP와 OLAP가 혼합된 워크로드를 처리해야 하는 요구가 계속 증가하고 있지만 현재의 데이터베이스 시스템들은 이러한 시장의 기대에 부응하지 못하고 있다.

이에 대한 주요한 원인중의 하나는 데이터베이스의 데이터 저장 단위인 데이터 페이지 저장구조가 특정 워크로드를 처리하는데 적합하도록 설계되었기 때문이다. 따라서 시간에 따라 그 특성이 변하는 워크로드의 경우에는 데이터베이스의 처리 성능이 저하될 수 밖에 없다는 문제가 발생한다[1].

이에 본 논문에서는 이러한 문제를 해결하기 위한 방법으로 혼합 워크로드를 효율적으로 처리하기 위한 동적 페이지 저장 모델과 페이지의 레이아웃을 관리하는 방법을 제안하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 기존의 데이터 저장 모델을 소개하고 혼합 워크로드를 처리하는 시스템에 대해 기술한다. 3장에서는 혼합 워크로드를 효율적으로 처리할 수 있는 동적 페이지 저장 모델의 구조에 대해 기술하고 4장에서는 워크로드를 모니터링하고 주기적으로 컬럼 그룹을 추출하여 데이터 저장 모델에 반영하는 데이터 레이아웃 관리 방법에 대해 기술한다. 5장에서는 기존 데이터 모델과 컬럼 그룹 기반의 데이터 모델의 성능을 평가함으로써 제안한 데이터 저장 모델의 우수성을 보이며 6장에서는 데이터 저장 모델에 대한 결론과 향후 연구에 대해 기술한다.

II. 관련 연구

데이터 모델은 데이터 저장 방식에 따라 크게 레코드 중심의 데이터 모델과 컬럼 중심의 데이터 모델로 구분할 수 있다. NSM(N-ary Storage Model)[2]은 레코드 중심의 대표적인 데이터 모델로 데이터 페이지 내에 순차적으로 레코드를 저장하는 데이터 모델이다. NSM은 레코드 단위로 데이터를 읽고 쓰기 때문에 레코드 단위로 데이터를 처리하는 워크로드에 적합한 데이터 모델이다.

DSM(Decomposition Storage Model)[2]은 하나의 데이터 페이지에 하나의 컬럼만을 저장하는 데이터 저장 모델로 대용량의 데이터에 대해 특정 컬럼을 접근하여 처리할 때 효율적인 데이터 모델이다. 하지만 DSM 데이터 구조의 특성상 하나의 레코드를 구성하는 컬럼들이 여러 데이터 페이지에 나누어 저장되어 있기 때문에 데이터 레코드를 처리할 경우에는 레코드를 재구성하는 비용이 높기 때문에 이로 인한 성능저하가 발생하게 된다.

PAX(Partition Attribute Across)[3]은 NSM과 DSM의 장점을 가지는 하이브리드 성격을 지닌 데이터 모델로 하나의 데이터 내에 컬럼 그룹 단위로 데이터를 저장하는 구조이다. 따라서 PAX는 혼합 워크로드를 처리할 때 NSM과 DSM보다 효율적인 데이터 저장 모델이다. 하지만 NSM, DSM과 마찬가지로 정적 데이터 저장 모델이기 때문에 워크로드의 특성이 동적으로 변하는 혼합 워크로드의 경우에는 효율적으로 관리할 수 없다는 한계를 가진다.

혼합 워크로드를 처리하는 시스템으로는 SAP의 HANA[4]와 Hyrise[5]가 있다. HANA는 로우 스토어(row store)와 컬럼 스토어(column store) 엔진을 별도로 두어 워크로드의 특성에 맞게 선택하여 처리한다. 즉, OLTP 워크로드일 경우는 로우 스토어를 이용하고 OLAP 워크로드일 경우는 컬럼 스토어를 이용하여 워크로드를 처리한다.

이에 반해 Hyrise 시스템[4]은 인메모리 기반의 데이터 저장 시스템으로 워크로드의 특성에 맞게 컬럼 그룹을 생성하여 데이터를 저장함으로써 혼합 워크로드를 효율적으로 처리할 수 있게 해준다.

III. 컬럼 그룹 기반 데이터 저장 모델

본 장에서는 혼합 워크로드를 효율적으로 저장 관리할 수 있는 컬럼 그룹 기반의 동적 데이터 저장 모델에 대하여 소개한다.

3-1 데이터 저장 구조

본 논문에서는 컬럼단위로 데이터를 저장하는 방식으로 설계하였다. 컬럼 그룹 기반의 데이터 저장 모델은 다음과 같이 몇 가지 특징을 가진다.

첫째, 하나의 데이터 페이지내에 컬럼들을 모두 저장한다. 따라서 컬럼기반의 데이터 모델(DSM)에 비해 상대적으로 적은 비용으로 레코드를 재구성할 수 있다.

둘째, 컬럼들은 컬럼 그룹 단위로 서버페이지에 저장된다. 따라서 컬럼 그룹의 개수만큼 서버페이지가 생성된다.

셋째, 컬럼 그룹에 접근할 때 레코드 전체를 접근하지 않고 컬럼 그룹에 접근하기 때문에 캐시미스의 발생이 감소하여 성능을 향상을 가져온다.

그림 1은 컬럼 그룹 기반의 동적 데이터 저장 모델의 구조를 보여준다. 데이터 페이지는 아래와 같이 데이터 헤더와 서버페이지들

로 구성되는데 서브페이지는 컬럼 데이터의 속성에 따라 고정 서브페이지(fixed length subpage)와 가변 서브페이지(variable length subpage)로 구분할 수 있다.

• 페이지 헤더

페이지를 처리하기 위한 기본 정보를 관리하는 부분으로 페이지 헤더의 세부정보는 표 1과 같다.

표 1. 페이지 저장 모델 구성
Table 1. Page Storage Model Parameters

항 목	내 용
페이지 아이디	페이지 아이디로 유일한 값을 가진다.
서브페이지 수	페이지를 구성하는 서브 페이지의 수를 나타낸다.
서브페이지 포인터 배열	각 서브페이지의 시작주소를 가지는 포인터 배열
애트리뷰트 수	레코드를 구성하는 애트리뷰트의 수를 나타낸다.
애트리뷰트의 서브페이지 번호	애트리뷰트가 저장된 서브페이지의 번호를 나타낸다.
애트리뷰트 크기	애트리뷰트 데이터 타입의 크기를 나타낸다. 크기가 가변일 경우는 V 표시를 한다.
레코드 수	페이지내에 저장된 레코드의 수를 나타낸다.

• 고정 서브페이지(F-subpage)

데이터의 크기가 고정된 컬럼값들을 저장한다. 데이터는 순차적으로 저장되고 서브페이지의 마지막 영역에는 데이터의 존재유무를 체크하는 비트배열을 가진다.

• 가변 서브페이지(V-subpage)

데이터의 크기가 고정되지 않은 컬럼값들을 저장한다. 서브페이지내의 데이터는 컬럼수만큼의 포인터와 저장된 컬럼값들로 구성되고 각 컬럼 포인터는 해당 데이터의 시작부분을 가리킨다. 서브페이지의 마지막 영역에는 포인터 배열을 두어 해당 데이터의 마지막 부분을 가리킨다. 따라서 해당 데이터의 크기는 이전 포인터와 해당 포인터의 차이값으로 구할 수 있다.

3-2 페이지 기본 연산

컬럼 그룹 기반의 페이지모델은 기본연산을 지원한다.

첫째, 레코드 삽입(insert) 연산은 레코드를 페이지에 저장하는 연산으로 데이터는 컬럼 그룹 단위로 해당 서브페이지내에 저장한다. 만약 페이지내에 공간이 없을 경우에는 새로운 페이지를 생성하여 저장한다.

둘째, 레코드 또는 특정 컬럼을 검색할 경우 헤더 정보를 이용하여 서브페이지 아이디와 레코드위치를 검색한다. 만약 서브페이지의 특정 컬럼을 검색할 경우에는 서브페이지의 마지막 영역에 위치한 포인터 배열에서 위치를 찾아 검색할 수 있다.

셋째, 레코드의 수정은 데이터의 크기가 고정일 경우와 가변

일 경우를 나누어 수행한다. 데이터의 크기가 고정일 경우, 해당 컬럼의 값을 새로운 값으로 수정한다. 데이터의 크기가 가변인 경우에는 고정페이지와 동일한 방법으로 값을 수정한다. 하지만 새로운 데이터의 크기가 기존의 데이터의 크기보다 크면 새로운 값을 저장할 수 없기 때문에 페이지를 재구성하여 값을 수정해야 한다.

넷째, 레코드의 삭제는 해당 서브페이지의 포인터 배열에서 엔트리를 삭제함으로써 레코드를 삭제한다. 이때 고정 데이터의 경우에는 기존의 저장 공간을 그대로 유지하지만 가변 크기의 데이터일 경우에는 페이지를 재구성할 때 해당 공간을 삭제한다.

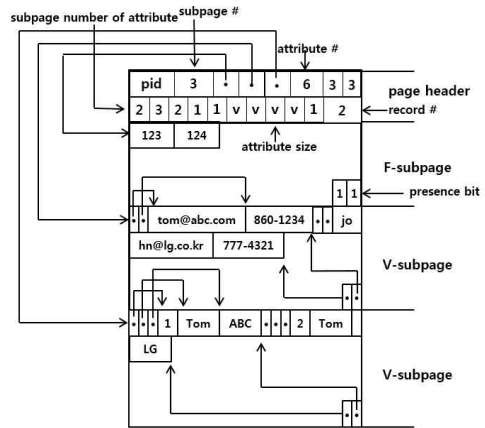


그림 1. 컬럼 그룹 기반 페이지 저장 모델 구조
Fig. 1. Page Storage Model Structure based on Column Group

IV. 동적 페이지 레이아웃 관리

동적 페이지 레이아웃 관리는 워크로드 분석을 통해 워크로드의 특성을 반영한 페이지 레이아웃을 생성하여 관리하는 것으로 페이지의 접근 패턴을 바탕으로 페이지별 최적의 컬럼 그룹을 추출하여 제공하고 동적으로 페이지의 레이아웃을 구성함으로써 혼합 워크로드를 효율적으로 처리할 수 있다. 동적 페이지 레이아웃 관리는 다음과 같이 페이지 모니터링, 페이지 레이아웃 관리, 페이지 레이아웃 동적 재구성으로 구성된다.

4-1 페이지 모니터링

페이지 모니터링은 그림 2와 같이 테이블을 구성하는 페이지들의 접근 컬럼 정보를 모니터링하고 질의 처리기로부터 질의 수행시 접근하는 컬럼들과 각 컬럼들의 선택도(selectivity) 정보를 수집한다. 여기서 선택도는 전체 데이터중에서 접근해야 하는 데이터에 대한 비율을 나타낸다. 또한 시스템 카탈로그로부터 테이블과 컬럼들에 대한 기본정보를 수집한다. 이외에도 페이지 모니터링은 데이터 저장 관리기로부터 질의를 실행

할 때 접근하는 페이지들의 정보를 수집한다.

따라서 페이지 모니터링은 테이블을 구성하는 페이지별로 컬럼 그룹을 생성하는데 필요한 정보들을 수집하고 관리하는 기능을 수행한다.

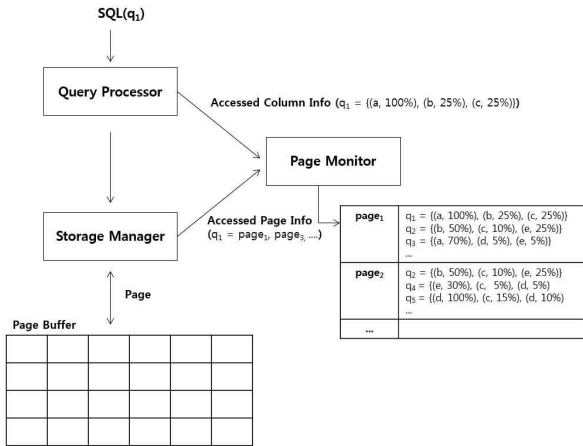


그림 2. 페이지 모니터링
Fig. 2. Page Monitoring

4-2 페이지 레이아웃 관리

페이지 레이아웃 관리는 그림 3과 같이 페이지 모니터링 정보를 바탕으로 주기적으로 해당 워크로드에 적합한 컬럼 그룹을 추출하고 데이터 저장 관리기에 정보를 제공하는 기능을 수행한다. 페이지 레이아웃 관리기가 컬럼 그룹을 생성하기 위해서는 컬럼 그룹 선택 알고리즘이 필요하고 질의 비용을 계산하기 위한 비용모델이 필요하다.

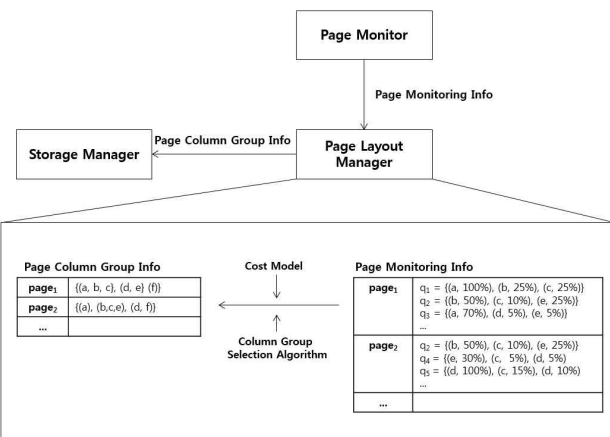


그림 3. 페이지 레이아웃 관리
Fig. 3. Page Layout Management

컬럼 그룹을 생성하기 위해서는 우선 페이지 모니터링으로부터 페이지 리스트 정보를 얻어온다. 각 페이지는 질의별 접근 컬럼과 선택도 정보를 가지고 있기 때문에 이를 바탕으로 비용 모델과 컬럼 그룹 선택 알고리즘을 이용하여 새로운 컬럼 그룹

을 생성한다. 비용 모델은 메모리내 캐시미스가 발생하는 횟수를 기반으로 한다. 따라서 페이지 레이아웃 관리기는 비용 모델을 통해 캐시미스가 적게 발생하는 컬럼 그룹의 조합을 계산하여 선택한다.

4-3 페이지 레이아웃 동적 재구성

페이지 레이아웃 동적 재구성이란 그림 4와 같이 주기적으로 페이지 레이아웃 관리기로부터 새로운 페이지 컬럼 그룹 정보를 얻어온 접근 횟수가 많은 페이지들만을 대상으로 페이지를 재구성하는 것을 의미한다. 즉, 임계치(threshold)를 설정하여 임계치 값 이상인 - 특정 기간동안 페이지 접근 횟수가 특정 횟수 이상인 - 페이지들만을 대상으로 페이지를 재구성한다.

후보 페이지 관리는 재구성할 페이지들을 선택하는 기능을 수행한다. 선택된 페이지들은 후보 페이지 큐(candidate page queue)로 전달한다.

후보 페이지 재구성은 페이지를 재구성하는 작업을 수행하는데 후보 페이지 큐로부터 후보 페이지를 읽어와 재구성하고자 하는 페이지를 확인한다. 그리고 메인 메모리로부터 해당 페이지의 헤더 정보를 읽어서 후보 페이지의 컬럼 그룹 정보와 동일하지 확인한다. 이때 메인 메모리내의 페이지 컬럼 그룹 정보와 후보 페이지의 컬럼 그룹 정보가 동일하면 페이지 재구성 없이 후보 페이지 큐로부터 다음 후보 페이지를 읽어와 비교 작업을 반복한다. 만약 메인 메모리내의 페이지 컬럼 그룹 정보와 후보 페이지의 컬럼 그룹 정보가 동일하지 않으면 메인 메모리내의 페이지를 새로운 컬럼 그룹에 맞게 새로운 페이지로 재구성한다.

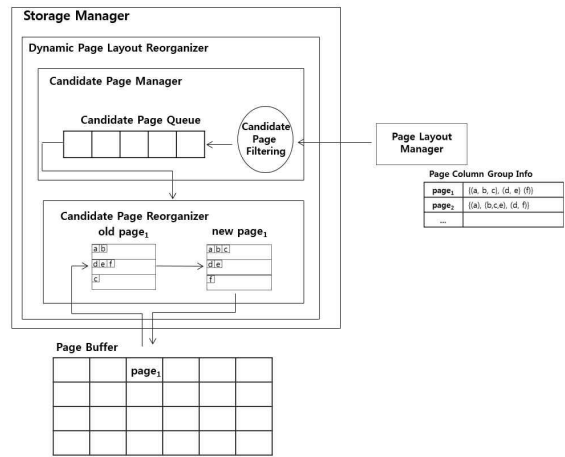


그림 4. 동적 페이지 재구성
Fig. 4. Dynamic Page Reconstruction

IV. 성능 평가

본 장에서는 페이지 저장 모델인 NSM, PAX와 본 논문에서 제안한 컬럼 그룹 기반의 페이지 모델의 성능을 평가한다. 본

장에서 보여지는 그림에서는 제안한 페이지 모델을 CGM(Column Group based Page Storage Model)으로 표시하였다.

실험은 2.83 GHz Intel(R) Core(TM) 2 Quad Processor와 4GB의 메인 메모리를 가지는 데스크탑에서 수행하였고 운영체제는 우분투 12.04 64bit를 사용하였다. 실험의 대상이 되는 페이지의 크기는 8KB로 하였다.

그림 5는 OLAP 워크로드 환경에서 각 페이지 모델별 성능을 보여준다. PAX와 CGM이 NSM보다 성능이 뛰어나함을 알 수 있다. 이는 NSM이 레코드를 기반으로 되어 있기 때문에 특정 컬럼을 접근할 때 모든 레코드를 읽어야 하기 때문에 캐시미스가 많이 발생하기 때문이다.

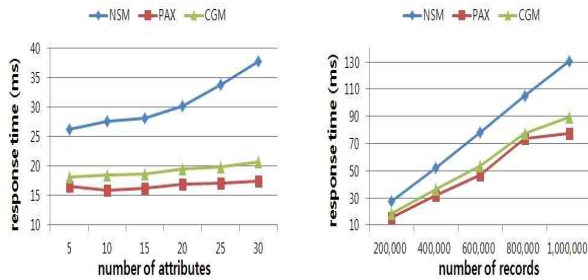


그림 5. OLAP 워크로드 환경에서의 응답시간
Fig. 5. Response time comparison with OLAP workload

그림 5에서 보면 애트리뷰트가 증가할 수록 NSM은 응답속도가 계속 증가하지만 PAX와 CGM의 경우는 응답속도의 증가 비율이 크지 않음을 알 수 있다.

종합하면, OLAP 워크로드의 실험결과 애트리뷰트의 수에 따른 성능은 CGM이 NSM과 비교하여 평균 36%가 빠름을 보였고 레코드 수에 따른 성능은 컬럼 그룹 기반의 페이지 모델이 NSM에 비해 평균 30%가 빠름을 보였다.

그림 6은 OLTP 워크로드 환경에서 각 페이지 모델들의 성능을 보여준다. 그림 6에서 보는바와 같이 PAX는 레코드를 구축하는데 비용이 발생하기 때문에 가장 성능이 좋지 않았다. OLTP 워크로드의 실험결과를 종합해보면, 애트리뷰트 수에 따른 성능은 컬럼 그룹 기반의 페이지 모델이 PAX 보다 평균 16% 좋음을 보였고 레코드의 수에 따른 성능은 약 6% 좋음을 보였다.

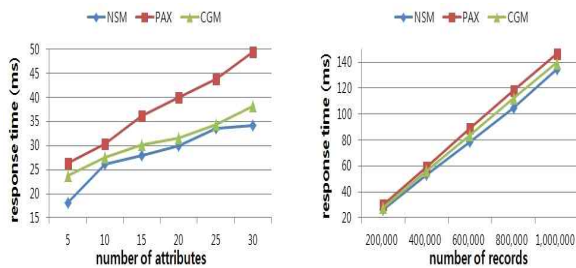


그림 6. OLTP 워크로드 환경에서의 응답시간
Fig. 6. Response time comparison with OLTP workload

마지막으로, 그림 7은 혼합 워크로드 환경에서의 각 페이지

모델별 성능을 보여준다. 그림 7에서와 같이 CGM이 다른 페이지 모델보다 성능이 우수함을 알 수 있다. 이는 컬럼 그룹 기반의 페이지 모델이 워크로드의 특성을 반영하여 동적으로 구성되기 때문에 캐시미스의 발생이 낮기 때문이다.

요약하면, CGM은 혼합워크로드에서 가장 좋은 성능을 보였고 OLAP, OLTP 워크로드 환경에서도 허용할만한 성능을 보였다.

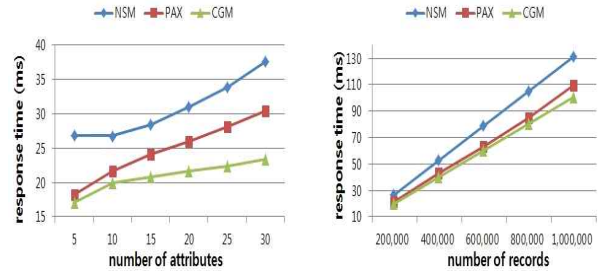


그림 7. 혼합 워크로드 환경에서의 응답시간
Fig. 7. Response time comparison with mixed workload

IV. 결론

본 논문에서는 혼합 워크로드를 효율적으로 처리하기 위한 동적 페이지 저장 모델을 제안하였고 동적으로 페이지 레이아웃을 관리하는 방법을 제안하였다.

기존의 페이지 저장 모델들은 모두 특정 워크로드 대해서는 좋은 성능을 보이지만 혼합 워크로드를 처리하는데 있어서는 성능상 한계를 가진다. 이에 본 논문에서는 워크로드의 특성을 고려하여 페이지를 동적으로 재구성함으로써 OLTP와 OLAP이 혼합된 워크로드를 효율적으로 처리할 수 있는 동적 페이지 저장 모델을 제안하였다. 또한 워크로드 처리에 최적화된 컬럼 그룹을 추출하고 동적으로 페이지를 재구성하는 페이지 레이아웃 관리 방법에 대해서도 소개하였다.

향후 연구로는 페이지 레이아웃을 결정하는데 필요한 비용 모델과 컬럼 그룹 선택 알고리즘에 대한 연구가 필요하고 다양한 혼합 워크로드를 대상으로 하는 벤치마크에 대한 연구가 필요하다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (2017-0-00253, 국제표준 기반 오픈 데이터 유통 플랫폼 확장 기술 개발)

참고문헌

- [1] Hasso Plattner, "A Common Database Approach for OLTP and OLAP using an In-Memory Column Database", SIGMOD Conference, 2009.
- [2] George P. Copeland and Setrag Khoshafian, "A Decomposition Storage Model", SIGMOD Conference, pp. 268-279, 1985.
- [3] Anastassia Ailamaki, David J. DeWitt, Mark D. Hill and Marios Skounakis, "Weaving Relations for Cache Performance", VLDB 2001, pp. 169-180, 2001.
- [4] Franz Farber, Norman May, Wolfgang Lehner, Philipp Große, Ingo Müller, Hannes Rauhe and Jonathan Dees, "The SAP HANA Database - An Architecture Overview", IEEE Data Eng. Bull. 35, No. 1, pp. 28-33, 2012.
- [5] Martin Grund, Jene Kruger and Hasso Plattner, "HYRISE - A Main Memory Hybrid Storage Engine", PVLDB, pp.105-116, 2010.

박경현(Kyoungyun Park)



1999년 : 충북대학교 컴퓨터공학과 (공학사)
2001년 : 충북대학교 전산학과 대학원 (이학석사)
2015년 : 충북대학교 전산학과 대학원 수료

2001년~현 재: 한국전자통신연구원 선임연구원
※관심분야 : 대용량 분산 데이터관리시스템, 빅데이터 플랫폼, 클라우드 컴퓨팅

원희선(Hee Sun Won)



1990년 : 연세대학교 전산학과 (이학사)
1992년 : KAIST 전산학과 대학원 (이학석사)
2016년 : KAIST 전산학과 대학원 (이학박사)

1992년~1999년: 한국방송(KBS) 기술연구소 연구원
2000년~현 재: 한국전자통신연구원 책임연구원
※관심분야 : 빅데이터, 클라우드 플랫폼, 오픈 데이터 거버넌스

류근호(Keun Ho Ryu)



1976년 : 숭실대학교 전산학과 (이학사)
1980년 : 연세대학교 대학원 전산전공 (공학석사)
1988년 : 연세대학교 대학원 전산전공 (공학박사)

1976년~1986년: 육군군수 지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송통신대학교 전산학과(조교수)
1989년~1991년: Univ. of Arizona Research Staff
1986년~현 재: 충북대학교 전기전자 컴퓨터공학부 교수
※관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 지식기반 정보검색 시스템, 유비쿼터스 컴퓨팅 및 스트림데이터 처리, 데이터 마이닝, 데이터베이스, 보안, 바이오 인포메틱스