웹 로그 분석을 위한 OLAP 시스템 및 성능 평가

김지현[†] · 용환승[††]

요 약

CRM을 위해서는 다차원 분석이 가능한 OLAP (On-Line Analysis Processing)기술을 적용한 방법 그리고 데이타 마이닝을 이용한 방법들이 각광 받고 있다. 고객 데이터 중에서 웹 로그 데이터를 실시간에 다차원 분석을 하기 위해서는 OLAP을 사용해야 한다. 그러나 OLAP을 적용하게 되면 웹 로그 데이터 자체가 가지고 있는 특성에 의해 희박성이 발생되고, 사전 집계 연산을 수행 할 시 데이터의 폭발(Explosion)현상이 일어난다. 이는 저장공간의 낭비 뿐 아니라 다차원 질의 시 성능 저하를 발생 시킨다. 본 논문에서는 웹 로그 데이터의 희박성에 대한 체계적인 접근을 위해 희박성을 발생시키는 원인과 2,3 차원의 희박성 형태들에 대해 밝혀 보고, 이러한 분석을 기반으로 성능 평가를 위한 테스트 데이터 모델과 질의 모델을 설계하였다. 그리고 희박성 처리를 위해 청크 방식을 사용한 MOLAP시스템을 구현해 보고, 이 시스템과 MS SQL 2000 Analysis Services, Oracle Express의 성능을 평가 및 분석 해보았다. 이는 웹 로그 데이터내의 희박성을 효율적으로 처리할 수 있는 저장구조와 인덱스 방식을 발견하는데 토대가 될 수 있다.

OLAP System and Performance Evaluation for Analyzing Web Log Data

Ji-Hyun Kim[†] and Hwan-Seung Yong[††]

ABSTRACT

Nowadays, IT for CRM has been growing and developed rapidly. Typical techniques are statistical analysis tools, on-line multidimensional analytical processing (OLAP) tools, and data mining algorithms (such neural networks, decision trees, and association rules). Among customer data, web log data is very important and to use these data efficiently, applying OLAP technology to analyze multi-dimensionally. To make OLAP cube, we have to precalculate multidimensional summary results in order to get fast response. But as the number of dimensions and sparse cells increases, *data explosion* occurs seriously and the performance of OLAP decreases. In this paper, we presented why the web log data sparsity occurs and then what kinds of sparsity patterns generate in the two and t.he three dimensions for OLAP. Based on this research, we set up the multidimensional data models and query models for benchmark with each sparsity patterns. Finally, we evaluated the performance of three OLAP systems (MS SQL 2000 Analysis Service, Oracle Express and C-MOLAP).

Key words: 웹 로그 분석, MOLAP, 희박성, 성능평가

1. 서 론

최근의 경쟁적 비즈니스 환경에서 CRM(Customer Relationship Management)을 위한 정보 기술의 발전이 급속도로 진전되고 있다. CRM의 목적은 각 기업이나 회사들이 새로운 고객을 유치하고, 기존의 고객들을 유지하면서 장기간의 이익 관계를 이어나가기 위해 고객의 취향과 필요를 이해하는 것이다 [1]. CRM을 위해 최근 많이 적용, 개발되고 있는 분야는 간단하게는 통계적 방식과 좀더 정교하게는 다차원 분석이 가능한 OLAP(On-Line Analysis Proc-

접수일 : 2002년 7월 11일, 완료일 : 2003년 3월 12일
[†] 동양 Systems 연구원
[††] 종신회원, 이화여자대학교 컴퓨터학과 부교수

essing)기술을 적용한 방법 그리고 데이타 마이닝을 이용한 방법들이 각광 받고 있다. 이에 적용되는 데이터들로는 고객, 구매, 공급업체, 경쟁사 등 외부 환경 자료들이 사용되고 있다. 특히 고객 데이터 중에서 각 기업이나 회사의 웹 서버에 저장되어 있는 웹 로그 데이터가 가장 쉽게 얻을 수 있고, 유용하게 사용될 수 있다.

웹 로그 데이터는 최근 웹 기술의 엄청난 발전과 사용자들의 수요가 많아 짐에 따라 그 양이 매일 수십 수백 메가 바이트까지 증가 하고 있다. 이러한 대규모 데이터를 사용해 실시간에 다차원 분석을 하기 위해서는 OLAP을 사용해야 한다. 하지만 OLAP을 사용하는데 있어 웹 로그 데이터 자체가 가지고 있는 특성에 의해 희박성이 발생되고, 이러한 데이터를 이용하여 사전 집계 연산을 수행 할 시 데이터의 폭발(Explosion) 현상이 일어난다. 이는 저장공간의 낭비 뿐 아니라 다차원 질의 시 성능 저하를 발생 시킨다[2-4].

따라서 본 논문에서는 웹 로그 데이터의 희박성에 대한 체계적인 접근을 위한 기초 단계로 희박성을 발생시키는 원인과 2,3 차원 내에서의 희박성 형태들에 대해 밝혀 보고, 이러한 분석을 기반으로 성능 평가를 위한 테스트 데이터 모델과 질의 모델을 설계하였다. 그리고 희박성 처리를 위해 청크 방식을 사용한 MOLAP시스템을 구현해 보고, 이 시스템과 MS SQL 2000 Analysis Services, Oracle Express의 성능을 평가 및 분석 해보았다.

본 논문의 구성은 다음과 같다. 2장은 웹 로그 데이터를 OLAP적용 시 발생되는 희박성 원인에 대해 분석해 보고, 실제 웹 로그 데이터에 대해 간단히 설명한다. 그리고 이를 사용하여 희박성 원인에 대해 확인하고, 2,3 차원 내에서의 희박성 형태에 대해 알아본다. 3장에서는 성능 평가를 위해 사용된 OLAP 시스템 중 청크 방식을 사용한 C-MOLAP시스템의 특징 및 주요 알고리즘에 대해 설명 하고, 4장에서는 성능 평가를 위한 테스트 데이터 생성기에 대한 설명과 OLAP제품들의 성능을 측정하여 그 결과를 기술한다. 마지막으로 5장에서는 본 연구의 결과 및 향후 연구 방향을 제시한다.

2. 웹 로그 데이터의 OLAP적용 시 희박성 원인 및 형태 분석

2.1 웹 로그 데이터의 OLAP적용 시 희박성 원인

웹 로그 데이터를 사용하여 OLAP을 적용할 시 희박성을 일으키는 원인들을 몇 가지로 분석해 볼 수 있다[11]. 첫번째로는 각 차원의 항목들이 밀집 항목과 희박 항목으로 나눠지고, 희박 항목의 수가 밀집 항목의 수보다 많을 경우 이러한 차원들의 결합으로 인해 희박성이 발생 되게 된다. 예를 들어 스키장 사이트의 시간차원의 경우 늦가을과 겨울에 성수기를 이루고, 나머지 시간대에는 방문자의 수가 희박하게 나타난다. 둘째로, 데이터 베이스 설계에 있어서 실제 로그 데이터 형식의 변화로 인해 발생하게 된다. 세번째는 시스템 에러에 의해서 즉 웹 서버의 에러에 의해서 발생 될 수 있다. 마지막으로는 특정 기간동안만 존재하고 없어져 버리는 데이터 즉 이벤트 페이지들과 같은 것에 의해 발생된다. 이 중 희박성을 일으키는 가장 주 원인이 되는 첫번째 경우를 실제 세가지 웹 로그 데이터 사용하여 확인해 본다. 다음 절에서는 분석을 위해 사용된 실제 웹 로그 데이터에 대한 대략적인 설명과 데이터 모델 구조에 대해 설명한다.

2.1.1 실제 웹 로그 데이터와 OLAP 적용을 위한 데이터 모델

분석에 사용된 웹 로그 데이터의 형식은 W3C 확장 로그 형식 가지며, 분량은 일주일 분량의 웹 로그 데이터를 사용하여 사전처리모듈[5]을 거친 후 OLAP 서비스에 적용한다. 그리고 OLAP 서비스에 적용하기 위한 데이터 모델은 스타 스키마 구조를 따른다. 스타 스키마 구조는 4개의 차원 - 방문자, 페이지, 검색엔진, 시간 - 을 가지며, 방문횟수를 측정값으로 한다.

첫번째 사례에 사용된 사이트는 세계 각국의 스포츠 소식을 전하는 사이트로서 회원가입을 통해서 주요 정보 서비스를 받을 수 있다. 분석에서 사용한 이 사이트의 웹 로그 데이터는 표 1에 간단히 설명 하였다.

사전처리가 끝난 로그 데이터와 고객 데이터 정보와 결합하여 데이터 큐브를 만들기 위한 데이터 모델은 아래 그림 1과 같다.

표 1. 사례 1의 웹 로그 데이터의 설명

기 간	2000년 10월 28일~11월 4일
데이터용량	실제 로그 데이터 → 2513975 records (163M) 사전 처리된 로그 데이터 → 35635 records (1.83M)

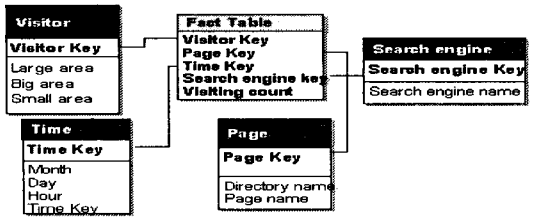

그림 1. 사례 1의 다차원 데이터 모델 항목 및 레벨 구조

두 번째 웹 사이트는 이화여대 데이터베이스 연구실의 웹 서버의 로그 데이터를 사용하였다. 이 사이트에서는 회원 가입을 요구하지 않으며, 데이터의 용량이 적은 관계로 15일 간의 웹 로그 데이터를 사용하였다. 분석에 사용된 로그 데이터의 설명은 아래 표 2에 나타내었다.

본 사이트에서는 고객 데이터가 존재 하지 않으므로 순수 웹 로그 데이터를 사용하였다. 데이터 모델은 그림 2와 같은 구조를 가진다. 세 번째 사례에 사용된 웹사이트는 자연 다큐멘터리 인터넷 방송국 사이트로서 첫번째 사례에 나온 스포츠 정보회사 사이트와 같이 회원 가입을 통해 사이트의 중요 정보들을 볼 수 있다. 사이트의 로그 데이터에 대한 설명은 아래 표 3에 나타낸다. 사례 3의 데이터 모델은 순수 웹 로그 데이터만을 사용하여 그림 2의 것과 같은 구조를 갖는다.

2.1.2 각 차원별 데이터 항목을 이용한 희박성 원인 분석

앞에서 설명한 것처럼 희박성이 발생하는 주 원인은 각 차원의 데이터 항목들이 밀집 항목과 희박 항목으로 나뉘고, 희박 항목이 밀집 항목에 비해 넓은 분포로 존재할 경우에 나타나게 된다. 본 절에서는 위와 같은 사실을 실제 로그 데이터를 사용하여 확인해 보도록 하겠다. 우선 첫번째로 방문자 차원을 기준으로 보면 사례1의 방문자 차원의 계층 구조 중 대권역을 기준으로 하여 방문 횟수에 대한 분포를 그림 3로 나타내었다. 방문자 차원의 경우 지역적 특징에 따라 밀집 항목 즉 서울과 광역시 그리고 그 외의 희박 항목으로 나눠 볼 수 있고, 희박 항목의 수가 매우 많음을 알 수 있다.

두 번째는 페이지 차원을 기준으로 각 페이지에 방문한 방문자의 횟수를 알아봄으로써 페이지 차원에서도 자주 방문 되는 인기 페이지 항목 즉 로그인 페이지, 주제 페이지, 관리자 페이지와 그 외의 비인기 항목을 구별해 볼 수 있다. 그림 4은 사례3의 페이지 차원별 방문횟수를 그래프로 보여준 그림이다.

세 번째로는 시간 차원의 각 시간대별 계층구조를 기준으로 하여 시간대의 방문횟수를 가지고 바쁜 시간대와 한가한 시간대를 알아본다. 그림 5는 사례2의 시간대별 방문횟수를 그래프화 한 것이다.

표 2. 사례 2의 웹 로그 데이터의 설명

기 간	2001년 02월 01일~02월 15일
데이터용량	실제 로그 데이터 → 198591 records (120M) 사전 처리된 로그 데이터 → 72309 records(25.1M)

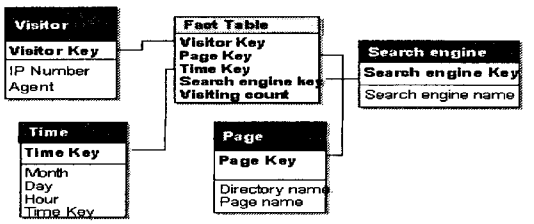

그림 2. 사례 2,3의 다차원 데이터 모델 항목 및 레벨 구조

표 3. 사례 3의 웹 로그 데이터의 설명

기 간	2001년 03월 28일~04월 03일
데이터용량	실제 로그 데이터 → 21575 records (13.1M) 사전 처리된 로그 데이터 → 19739 records(4.72M)

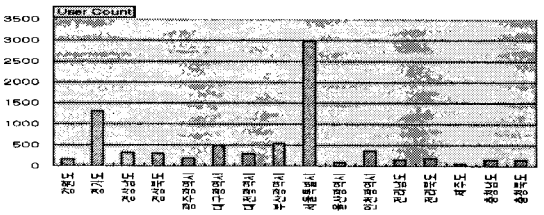

그림 3. 사례1 방문자 차원

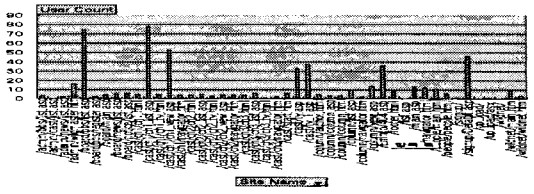

그림 4. 사례3 페이지 차원

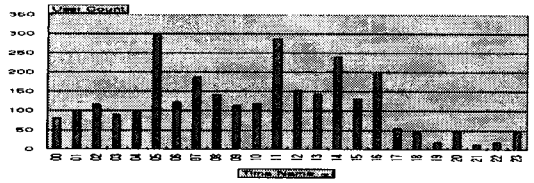

그림 5. 사례2 시간 차원

위의 예에서 각 시간대별 접속자수의 접속현황을 보면 저녁시간을 제외한 나머지 시간대에 접속자 수가 많은 바쁜 시간대 임을 볼 수 있다. 시간 차원의 경우 나머지 차원들과는 달리 밀집 항목의 수가 희박 항목에 비해 많은 비율을 차지 하고 있다. 마지막으로 검색엔진 차원의 경우 각 검색엔진에 의해 참조되어 본 사이트에 오게 된 방문자의 수를 알아봄으로써 밀집 항목과 희박 항목을 알 수 있다. 그림 6은 사례 1의 검색 엔진에 의해 참조된 횟수를 그래프화 한 것이다.

검색 엔진 차원의 경우 위와 같은 밀집 항목과 희박 항목의 뚜렷한 구분이 일어나는 원인은 각 검색엔진에 검색어를 다양하고 적절하게 제시 함으로써 자신의 사이트를 찾는 사람들이 여러 루트를 통하여 본 사이트를 발견할 수 있게 도와줌으로써 발생하게 된다.

세가지 실제 로그 데이터를 사용하여 위 4가지 차원들에 대해 우리는 각 차원에 희박 항목과 밀집 항목이 존재 하고, 희박 항목의 수가 밀집 항목의 수보다 많음을 확인할 수 있었다. 다음 장에서는 이러한 밀집 항목과 희박 항목으로 구성된 각 차원들의 결합에 의해 만들어 지는 희박성 형태에 대해 알아보도록 하겠다.

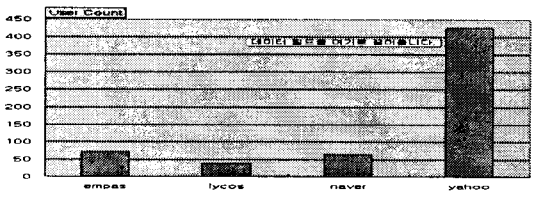

그림 6. 사례1 검색 엔진차원

2.2 웹 로그데이터의 OLAP 적용시 희박성 형태

본 장에서는 실제 웹 로그데이터를 OLAP적용 했을 시 만들어 질 수 있는 2,3차원 데이터 모델에서의 희박성 형태에 대해 알아보도록 하겠다. 희박성 형태를 확인하기 위해 MS SQL 2000 Analysis Service와 DBMiner 3D Cube Explore를 사용하였다.

2.2.1 2차원 데이터 모델에서의 희박성 형태

위에서 제시한 3가지 실제 웹 사이트의 로그 데이터를 사용하여 OLAP적용을 위한 다차원 모델 생성시, 2차원 모델을 기본으로 하여 만들어 지는 희박성 형태에 대해 알아 보도록 하겠다. 그림 7은 사례 3의 페이지 차원과 시간 차원을 이용하여 큐브를 생성했을 때 만들어 지는 희박성 형태이다.

위 그림에서 볼 수 있듯이 줄무늬(Stripe)형태의 희박성 형태가 시간 차원과 페이지 차원 내에서 교차가 이루어지면서 그리드(Grid)형태 와 어느 한 지역에 뭉쳐서 나타나는 클러스터(Cluster) 형태를 나타내게 된다. 이러한 희박성 형태는 시간차원에 있어서 바쁜 시간대에 해당되는 항목이 페이지 차원의 항목 종류와 관계없이 꾸준히 방문한 사람이 많고, 또한 인기 페이지에 해당되는 항목들에 대해서는 시간대에 제한 받지 않고 방문횟수가 많은 경우 발생된다. 그림 8은 사례 1의 검색 엔진 차원과 시간 차원에 의해 만들어진 희박성 형태를 보여준다. 여기에서도 위의 경우와 마찬가지로 그리드(Grid)형태와 클러스터(Cluster) 형태의 희박성 형태를 발견할 수 있다.

이러한 희박성의 형태는 페이지차원과 시간차원 뿐만 아니라 대부분의 2차원 데이터 모델에서 빈번하게 발생된다.

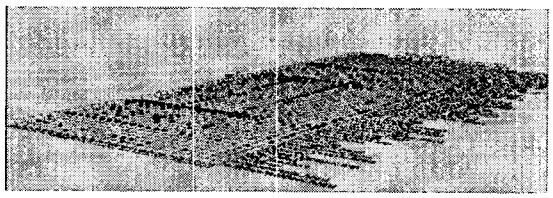

그림 7. 사례 3의 2차원 희박성 형태 : 페이지-시간차원

그림 8. 사례 1의 2차원 희박성 형태 : 검색엔진-시간차원

2.2.2 3차원 데이터 모델에서의 희박성 형태

이번 절에서는 3차원 내에서 발생되는 희박성의 형태에 대해 알아본다. 그림 9와 10은 사례 3과 2의

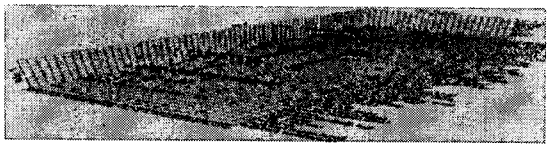

그림 9. 사례 3의 3차원 희박성 형태 : 시간-페이지-검색엔진 차원

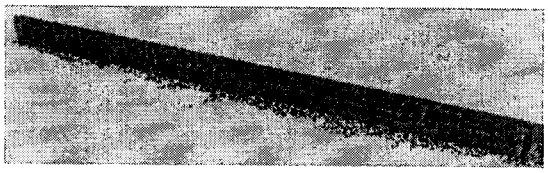

그림 10. 사례 2의 3차원 희박성 형태 : 시간-페이지-검색엔진차원

페이지 차원, 시간차원, 검색엔진차원의 희박성 패턴을 보여 주는 그림이다.

2차원 데이터 모델과 같이 3차원 데이터 모델에서도 그리드(Grid)형태와 클러스터(Cluster)형태의 희박성 형태가 발견됨을 볼 수 있다.

다음 장에서는 이러한 희박성을 다루기 위해 청크 방식을 사용하여 구현한 MOLAP시스템에 대해 알아 보도록 하겠다.

3. C-MOLAP시스템의 특징

OLAP에서 사용자의 질의에 빠르게 응답하기 위해서는 사전 집계 연산을 해야 한다. 따라서 여러 개의 관련 그룹 연산을 계산하는 것은 OLAP 어플리케이션의 주요 연산자가 된다. 하지만 이러한 과정에서 중요하게 다루어져야 할 점은 이러한 사전 집계연산을 수행하는 데 있어서 무효 셀들을 어떻게 처리 할 것이며 연산의 속도는 얼마나 빠르게, 적은 메모리 용량을 사용하여 수행할 수 있느냐 하는 것이 중요하다 할 수 있다. 본 장에서는[6]를 바탕으로 구현한 사전 집계 연산을 효율적으로 할 수 있고, 희박 데이터에 대해 압축 방식을 사용하는 MOLAP시스템을 C-MOLAP이라 명명한다. 다음절에서는 C-MOLAP의 특징 및 주요 알고리즘을 저장 방식과 집계 연산 수행 측면에서 제시한다.

3.1 C-MOLAP시스템의 저장 방식

첫 번째 저장 방식 측면에서 봤을 때 이 시스템은 기본적으로 MOLAP시스템을 기반으로 하기 때문에 모든 데이터들을 배열 형태로 저장 하게 된다. 배열에 의해 데이터들이 저장 될 경우 각 데이터들의 위치가 처음부터 결정됨으로 인덱스에 영향을 미치지 않고, 갱신 될 수 있다는 장점을 가지게 된다. 하지만 이는 큐브 내의 희박성을 일으키는 주 원인이 되기도 한다. 현재 C-MOLAP 시스템에서는 이러한 희박한 데이터 셀들을 처리 하기 위해 n차원의 데이터 배열을 더 작은 단위의 n차원 배열 형태인 청크 단위로 나누고, 각 청크에 대해 희박 청크와 밀집 청크로 나누어 희박 청크에 대해 Chunk-Offset Compression을 적용한다[12,13].

본 시스템에서는 OLAP연산 수행 시 청크 단위로 디스크 내에서 메모리로 읽어오고, 쓰기 때문에 디스크 I/O단위인 디스크 블록 사이즈로 청크 크기를 결정 하게 된다.

그림 11은 청크 사이즈를 결정하는 수식으로[6]에서 가정하는 것처럼 각 차원의 사이즈와 관계없이 일정 크기로 청크 사이즈를 결정하는 것과는 달리 차원의 사이즈를 고려하여 청크 사이즈를 결정할 수 있다. 위의 수식을 통해 제시된 각 청크 사이즈들의 조합이 디스크 블록 사이즈보다 적거나 같아질 때까지, 각 차원의 항목의 수를 $(1/2)^n$으로 나눠가며 청크 사이즈를 결정 할 수 있다. 현 시스템에서는 하나의 디스크 블록 사이즈를 50K로 가정하고, 차원의 사이즈가 50*2160*10000이라 할 경우 그림 3.1의 알고리즘에 따라 7번의 순환을 지난 후에 3*16*78사이즈의 청크 사이즈를 결정 할 수 있다. 이렇게 나누어진 각각의 청크들에 대해서는 청크내 셀들이 40%이상 값을 가지는 지에 따라 밀집 청크와 희박 청크를 정의 한다. 밀집 청크에 대해서는 모든 셀 내의 값만을 저장한다. 반면, 희박 청크에 대해서는 "Chunk-Offset Compression"방식을 사용한다(그림 12).

이 방식에서는 청크내의 데이터 값의 인덱스 값을 계산하는 행 우선 방식이나 열 우선 방식을 사용하여 offset값을 정수 값으로 계산하고, offset과 데이터 값들의 쌍을 저장한다. 희박 청크의 경우 청크의 크기가 항상 변하게 됨으로 데이터 파일의 시작에 메타

$$|Ci| = |Di| * (1/2)^n$$

|Ci| = i번째 차원의 청크 사이즈,
|Di| = i번째 차원의 청크 사이즈, n : 반복 횟수

그림 11. 청크 사이즈 결정 식

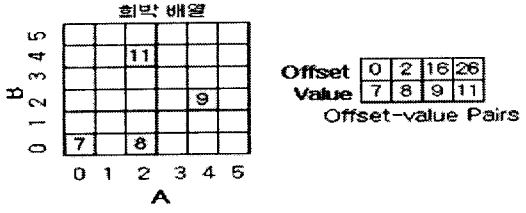

그림 12. Chunk-Offset Compression

데이터로서 각 청크의 길이를 저장하게 된다. 아래 그림 13은 각 청크들의 각 차원별 시작항목과 끝 항목을 계산해서 청크의 범위를 저장하는 메타 데이터와 실제 데이터 값들의 저장 구조이다.

```
typedef struct Chunk_Numbering{
        int start_x int end_x;
}Chunk_N // 각 청크의 차원들의 시작항목과 마지막 항목을 계산한다.
typedef struct A_Mem_Save{
   int offset;          // 각 chunk영역 내에서의 data offset
   int data_val;
   int chunk_num}A_Mem
```

그림 13. 각 청크 내의 메타 데이터와 실제 데이터 저장 구조

3.2 C-MOLAP시스템의 집계연산 수행방식

본 절에서는 사전 집계 연산을 수행하는 측면에서 살펴 보겠다. N개의 차원으로 이루어진 데이터 큐브에 대해 그룹 연산은 2^n개가 성립되게 된다. 2^n개의 모든 그룹 연산들을 메모리 내에 모두 유지 시키기 위해서는 충분한 메모리 용량이 있을 경우에만 가능하다. 따라서 [6]에서 제시하고 있는 사전 집계 연산 알고리즘은 최소한의 메모리를 사용하여 여러 개의 그룹 연산을 동시에 계산할 수 있도록 하는데 그 목표를 두고 있는데 본 C-MOLAP시스템에서는 이 방법을 사용하여 집계연산을 수행한다. 이는 사전 집계 연산을 수행하는데 있어서 최소의 메모리를 사용하여 매 단계 마다 관련된 부분만을 로드해서 한번의 디스크 스캔으로 모든 사전 집계 연산을 수행한다. 다음은 이 알고리즘 내에서 주요하게 다루어지는 개념이다.

• Minimum Memory Spanning Tree(MMST)

MMST는 큐브 내의 group_bys를 동시에 계산할 수 있도록 하기 위해 필요한 최소한의 메모리를 구하는 자료 구조를 가르킨다. 그림 14에서 보여 주는 것처럼 3개의 차원을 가지며, 큐브 사이즈는 16*16*16

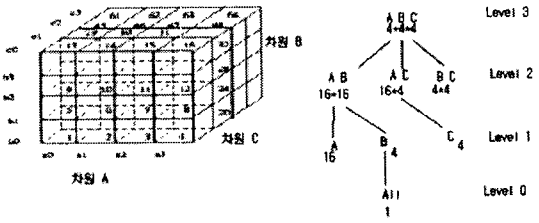

그림 14. 3차원 큐브와 MMST

이고, 청크 사이즈는 4*4*4이며 A,B,C차원 순서대로 읽어 가는 큐브가 있다고 가정하자.

위에 그림에서 우리가 BC라는 서브그룹 연산을 계산한다고 할 때, 1~4번까지의 청크를 읽어서 b_0c_0의 그룹연산 값을 계산할 수 있다. 따라서 BC를 위해서는 4*4크기의 메모리를 가지고서 모든 BC관련 그룹연산을 계산할 수 있게 된다. 이와 같이 AC의 경우 a_0c_0계산을 위해서는 4*16의 메모리를 할당해야 한다. 따라서 전체 서브그룹 연산에 대한 메모리 할당은 다음과 같은 식으로 나타낼 수 있다.

$\prod_{i=1}^{n} |D_i| \times \prod_{i=1}^{n-1} |C_i|$ $|D_i|$: 차원 i의 사이즈 $|C_i|$: 차원 i의 청크 사이즈

그림 15. 서브 group_bys의 메모리 할당량

p는 부모 노드와 현재 자신의 노드의 접두사가 같은 부분의 길이를 의미한다. 각 차원의 항목의 수는 각 청크의 항목의 수에 비해 매우 크다는 것을 우리는 알고 있다. 따라서 이 수식에서 볼 수 있듯이 전체 메모리를 할당하는데 있어서 중요한 영향을 미치는 요소는 각 차원의 순서를 어떻게 정하느냐에 따라 좌우 되게 된다. 아래 그림 16은 이를 위한 최적의 차원 순서를 정하는 수식이다.

최적의 차원의 순서 : |D1| <= |D2| <= .... <= |Dn|

그림 16. 최적의 차원 순서

그림 16에서 볼 수 있듯이 항목 수가 적은 차원에서 큰 차원 순서대로 읽는 것이 가장 작은 메모리 양을 요구한다. 그림 15와 16의 수식을 통해 우리는 전체 group_by를 계산하는데 필요한 최소한의 메모리 양과 각각의 전group_by를 계산하기 위해 필요한 메모리 양을 계산해 낼 수 있다.

4. 테스트 데이터 생성과 성능 평가 시스템 구현 및 결과 분석

본 장에서는 2장에서 제시한 웹 로그 데이터의 희

박성 형태를 기반으로 성능 평가를 위해사용 된 데이터 모델 및 질의 모델을 제시하고, 생성된 테스트 데이터를 이용하여 MS SQL 2000 Analysis Service, Oracle Express Server, C-MOLAP 시스템들에 대해 실제 OLAP모델을 구성한다. 그리고 각각의 제품에 대해 설정한 질의 모델들을 수행하여 웹 로그 데이터 희박성에 따른 성능을 측정한다.

4.1 데이터 모델 및 질의 모델 설계

성능 평가를 위해 사용된 데이터 모델들은 차원 테이블로는 웹 로그데이터를 분석하는데 가장 핵심이 되는 방문자, 페이지, 시간을 기반으로 하고, 측정값은 방문 횟수로 한다. 각각의 데이터 모델들의 계층 구조와 자세한 정보는 아래 표 4와 같다.

표 4. 차원정보

차원	방문자	페이지	시간
항목 수	10,000	50	2,160
계층 수	1	4	3

위에서 제시한 3가지 차원 정보를 이용하여 본 논문에서 사용하게 될 2,3 차원 데이터 모델은 다음과 같다.

● 2차원

2차원 모델은 방문객과 시간, 방문객과 페이지 차원들로 이루어 진다. 시간 차원의 경우 달, 날짜, 시간 3개의 계층으로 이루어 지고, 페이지 차원의 경우 루트 디렉토리, 2개의 서브 디렉토리, 페이지 4개의 계층으로 이루어 진다. 방문객과 페이지 차원으로 이루어진 2차원 데이터 모델은 사례 A로 방문객과 시간 차원으로 이루어진 것에는 사례 B로 가정한다. 사례 A와 B에 대한 데이터 모델과 자세한 정보는 그림 17과 표 5과 같다.

● 3차원

3 차원 모델은 방문객, 시간, 페이지 3가지 차원들로 이루어 진다. 3가지 차원 테이블의 계층 구조는 2차원에 사용된 정보와 같고, 3차원 데이터 모델은 사례 C로 가정한다. 사례 C에 대한 데이터 모델과 자세한 정보가 그림 18와 표 5과 같다.

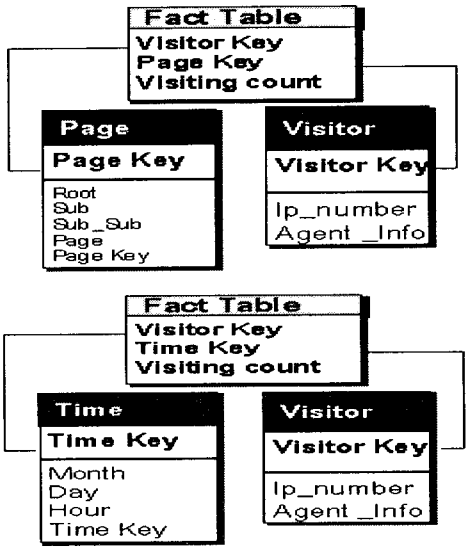

그림 17. 2차원 데이터 모델

표 5. 2,3차원 데이터 모델의 상세정보

사례	차원	전체 레코드 수	희박성(10%)
A	방문객, 페이지	500,000	50,000
B	방문객, 시간	21,600,000	2,160,000
C	방문객, 시간, 페이지	1,080,000,000	108,000,000

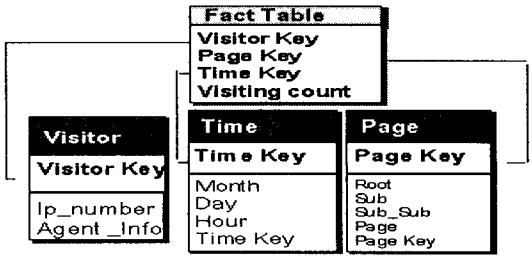

그림 18. 3차원 데이터 모델

● 질의 모델

2,3 차원 데이터 모델에 의해 만들어진 테스트 데이터를 사용하여 OLAP시스템들의 성능을 평가하기 위해 사용되는 질의 모델은 다음과 같다. 우선 OLAP 시스템에서 일반적으로 자주 사용하고 있는 질의로는 그림 19와 같다[6,7].

본 논문에서는 위의 7가지 일반적인 OLAP 질의와 더불어 OLAP 질의 중 많은 항목들 사이에 숨어 있는 데이터를 쉽게 파악하고자 할 경우 사용되는 랭킹 질의를 사용하여 웹 로그 데이터 분석을 하는데 있어 주 방문객들을 파악할 수 있게 하는 Top500질

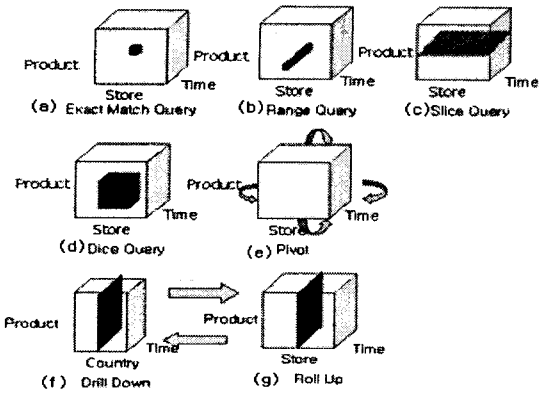

그림 19. OLAP의 일반적인 7가지 질의 종류

의를 설계하였다. 위에서 제시 했던 사례 A,B,C 데이터 모델에 대해 7가지 일반적인 OLAP질의 모델은 아래와 같다.

표 9는 3가지 사례에 대해 각각 우수 고객을 얻기 위해 수행되는 Top500 질의에 대해 보여준다.

4.2 시스템 구현 환경 및 시스템 전체 구조도

본 장에서는 앞에서 분석한 희박성 형태와 데이터 모델에 의해 생성한 테스트 데이터를 사용하여 MS SQL 2000 Analysis Service, Oracle Express Server, C-MOLAP 3가지 시스템에 대해 다차원 질

표 6. 사례 A의 일반적인 OLAP질의 모델

질의 종류	질의 내용
Exact Match	이벤트 페이지(PK00003)에 방문자(VK03018)가 방문한 횟수
Range	단골 방문자VK03000~VK03600가 메뉴S00001에 방문한 횟수
Slice	방문자(VK03018)가 전체 페이지에 방문한 횟수
Dice	전체 방문자에 대해 메뉴 S00001에 방문한 횟수
Pivot	열에 방문자, 행에 페이지 차원으로 보여 주던 것을 열에 페이지, 행에 방문자 차원으로 보여주시오.
Drill Down	방문자 VK0318이 페이지차원의 R0001디렉토리를 기준으로 하여 한 차원 내린 디렉토리의 방문 횟수를 보여주시오.
Roll Up	방문자 VK0318의 페이지 차원의 페이지 레벨을 기준으로 한 차원 위인 상위 디렉토리의 방문 횟수를 보여주시오.

표 7. 사례 B의 일반적인 OLAP질의 모델

질의 종류	질의 내용
Exact Match	시간(TK00042)에 방문자(VK03018)가 방문한 횟수
Range	단골방문자VK03000~VK03600가 8월2일 하루동안 방문한 횟수
Slice	방문자(VK03018)가 전체 시간대에 방문한 횟수
Dice	전체 방문자에 대해 8월 2일 하루동안의 방문한 횟수
Pivot	열에 시간, 행에 방문자차원으로 보여주던 것을 열에 사용자, 행에 시간 차원으로 보여주시오.
Drill Down	방문자 VK3018의 시간 차원의 8월 2일을 기준으로 하여 한 차원 내린 시간대의 방문 횟수를 보여주시오.
Roll Up	방문자 VK3018의 시간 차원의 8월 2일을 기준으로 한 차원 위인 8월달의 방문 횟수를 보여주시오.

표 8. 사례 C의 일반적인 OLAP질의 모델

질의 종류	질의 내용
Exact Match	이벤트페이지(PK00003)에 방문자(VK00012)가 TK02041시간대에 방문한 횟수
Range	8월 1일에 자주 방문하는 방문자그룹(VK00001 - VK00100)이 새로 단장한 페이지 PK00001 - PK00030에 방문한 횟수
Slice	방문자(VK00001)가 전시간대에 전 페이지에 걸친 방문 횟수
Dice	자주 방문하는 방문자그룹이 (VK00300~VK00330) 8월 1일날 전 페이지에 대한 방문 횟수
Pivot	방문자 VK00012에 대해 행에 페이지 차원와 열에 시간 차원로 보여 줬던 것을 행에 시간 차원 열에 페이지 차원으로 방문 횟수를 보여 주시오.
Drill Down	사용자 'VK000012'와 'PK00008' 페이지에 대해 시간차원에서 8월 1일을 기준으로 보여 주었던 것을 한 차원 내린 시간을 기준으로 방문 횟수를 보여 주시오.
Roll Up	사용자 'VK000012'와 'PK00008' 페이지에 대해 시간차원에서 8월 1일을 기준으로 보여 주었던 것을 한 차원 올려 보여 주도록 한다.

표 9. 사례A, B, C의 Top500질의 모델

사례 A	현재 이벤트가 수행되는 페이지(PK00003)에 대해 가장 많이 방문한 사람의 Top500을 구하시오.
사례 B	이번 한달 동안(10월) 가장 많이 방문한 사람의 Top500을 구하시오.
사례 C	이번 한달 동안(10월) 새롭게 단장한 메뉴(PK00003)에 대해 가장 많이 방문한 사람의 Top500을 구하시오.

의를 수행하여 응답 성능을 측정 할 수 있는 전체 구조도에 대해 설명한다. OLAP 제품들의 성능 평가 프로그램의 구현환경은 아래 표 10과 같다.

MS SQL 2000 Analysis Service는 성능 평가 프로그램을 구현하기 위해 개발 언어로는 Visual Basic 6.0를 사용하였다. Visual Basic 프로그램내에서는 이미 만들어져 있는 OLAP큐브에 접근 하기 위해 ADO MD API를 사용하고, MDX구문을 통해 질의를 수행한다[7]. Oracle Express Server의 경우는 Oracle Express Administrator에 의해 다차원 모델을 구성하고, Oracle Express Object시스템과 Oracle Basic언어를 사용하여 성능평가 인터페이스를 구현하였다. 그리고 Oracle Language를 사용해 큐브에 대한 접근 및 질의를 수행하였다[8,9]. 마지막으로 C-MOLAP시스템의 경우 MOLAP시스템 구현에 사용했던 언어와 동일한 Visual C++6.0을 사용하여 성능 평가 프로그램을 구현하였다. 생성한 테스트 데이터를 로드해서 성능 평가를 하여 분석하는 전체 시스템 구조는 그림 20과 같다.

표 10. 시스템 구현환경

운영 체제	Windows NT 4.0 Server
OLAP 서버	Oracle Express, MS SQL 2000 Analysis Service
개발 도구 및 언어	Visual Basic 6.0 MDAC (Microsoft Data Access Components) 2.5 SDK (ADO MD: ActiveX Data Objects Multidimensional), MDX (Multidimensional Expressions) OEO (Oracle Express Object) Express Basic, Express Language Visual C++ 6.0

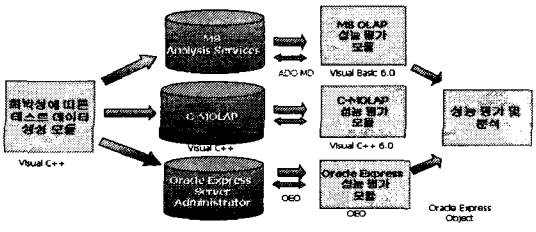

그림 20. 시스템 전체 구성도

앞에서 정의한 데이터 모델과 희박성 형태에 따라 생성된 데이터를 파일 형태로 저장하고, 각각 세가지 OLAP서버 내에 로드 한다. MS SQL 2000 Analysis Service의 경우 MS SQL 2000내부의 DTS기능을 사용하여 각각 RDB형태로 저장하고, 저장된 데이터들을 사용하여 Analysis Service내에서 2,3차원에 대한 큐브를 정의한다[10]. Oracle Express의 경우는 Oracle Express Administrator내의 Database위저드를 사용하여 파일 형태의 데이터들을 큐브로 변환하여 저장한다. 세가지 OLAP시스템 내부에서는 2,3 차원 형태의 큐브들에 대해 사전 연산을 수행하고, 각각의 큐브들에 대해 성능 평가 모듈을 통해 앞에서 정의한 질의 모델에 대한 수행 시간을 측정한다. 마지막 과정으로 응답 시간을 통해 3가지 시스템의 성능을 평가하고 분석한다.

4.3 성능 평가 프로그램

MS SQL 2000 Analysis Service의 경우 성능 측정 프로그램은 Visual Basic으로 구현하였다. 본 프로그램에서는 큐브를 생성하는 방법으로 수동적으로 Analysis Service내부의 위즈드를 사용하여 큐브 생성 및 사전 연산을 처리 하였다. 이와 같이 생성된 큐브에 질의를 수행하기 위해서 Microsoft MDAC SDK 2.5를 통해 제공하는 ADO MD API를 사용하였다. 그림 21은 MS SQL 2000 Analysis Service의

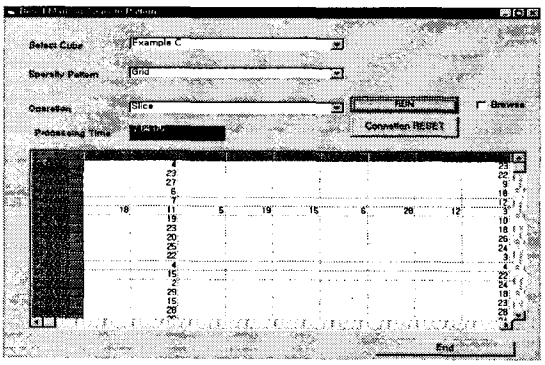

그림 21. MS SQL 2000 Analysis Service 성능 평가 인터페이스

성능 평가 프로그램의 인터페이스 이다. 이 그림은 사례C의 그리드 형태의 희박성 형태를 지닌 데이터 모델에 대해 일반적인 7가지 OLAP 질의 중 Slice 연산을 수행한 수행 결과와 수행 시간을 보여준다.

두 번째 Oracle Express Server 또한 큐브 생성은 수동적으로 수행하고 이들 큐브에 대해 질의를 수행하기 위해 Express Language를 사용하였고, 성능 평가 인터페이스를 위해서 Oracle Express Object와 Express Basic언어를 사용하였다. 그림 22는 사례A의 클러스터 형태에 대해 Top500질의를 수행한 결과 테이블과 수행시간을 보여준다.

그림 23은 C-MOLAP을 사용하여 클러스터 형태를 가진 사례 A에 대하여 Range질의를 수행 했을 때의 화면을 보여준다.

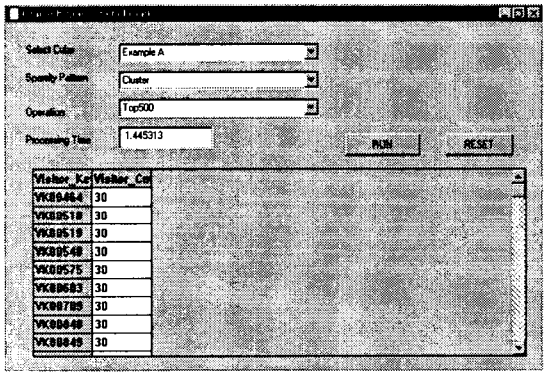

그림 22. Oracle Express 성능 평가 인터페이스

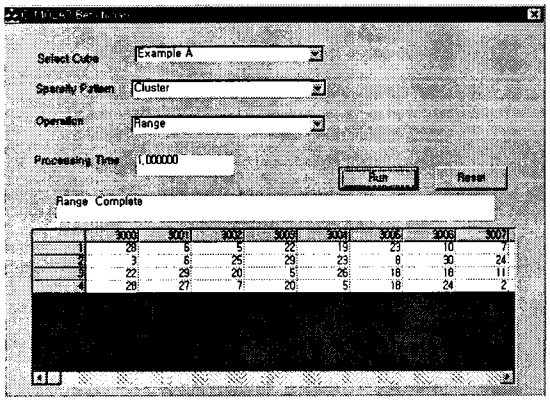

그림 23. C-MOLAP 성능 평가 인터페이스

4.4 성능 평가 결과 및 분석

3가지 OLAP시스템에 대한 성능 평가를 하기 위해 앞에서 제시 했던 질의 모델을 수행할 시 소요되는 시스템 시간을 측정 하였다. 각 시스템 시간은 초(Sec)단위로 측정하였고, 각 제품별 성능 측정치는 캐쉬에 의한 영향을 최소화하기 위해 처음 수행되어 나오는 수치들의 3회 수행결과의 평균을 사용하였다. 측정된 결과를 분석하기 위해 각각의 데이터 모델들에 대해 OLAP시스템과 희박성 형태에 따라 질의 결과를 보여 주었다.

그림 24은 Top500 질의를 수행한 결과이다. Top500 질의의 경우 OLAP질의 모델 중 랭킹 질의의 일종으로 정렬 알고리즘이 그 성능에 주요한 영향을 미치게 된다. 따라서 C-MOLAP시스템의 경우 이 질의에 대한 성능 평가에서는 제외 시킨다. 위 그림에서 볼 수 있듯이 희박성 형태에 관계없이 사례 A와 B에서와 같이 큐브 사이즈가 적을 경우 Oracle Express가 1초 이내의 빠른 결과를 보여 주지만 사례 C와 같이 큐브 사이즈가 커짐에 따라 MS SQL 2000 Analysis Service가 10배 이상의 향상된 성능을 보여 준다.

다음에서는 OLAP의 일반적인 질의 7가지에 대해 성능 평가한 결과를 분석해 보도록 하겠다. 본 측정 결과에서는 3가지 시스템에서 성능측정 결과가 2초 이내인 경우 큐브 사이즈와 희박성 형태에 따라 해당 질의에 대해서는 좋은 성능을 나타낸 다고 가정하고, 그래프화 하지 않았다. 그림 25, 26, 27에서는 사례A,

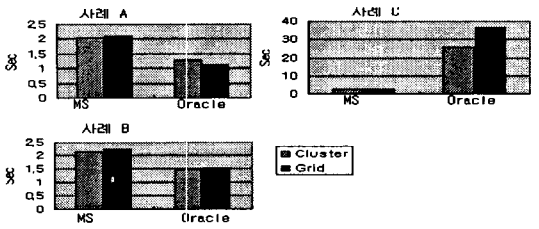

그림 24. Top500 성능 평가 결과

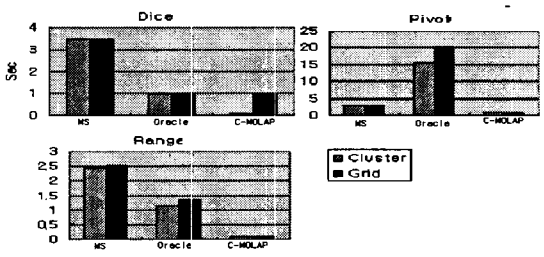

그림 25. 사례A의 성능 평가 결과

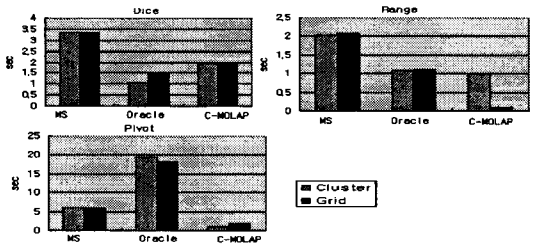

그림 26. 사례B의 성능 평가 결과

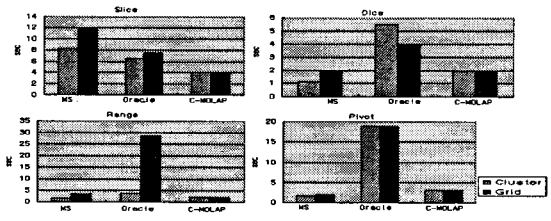

그림 27. 사례C의 성능 평가 결과

B, C의 성능 측정 결과를 보여준다. 위의 그림에서 알 수 있듯이 사례 A와 B에서 각 차원의 부분 집합의 조합에 의해 사용자 질의에 응답하는 Dice와 Range질의의 경우 MS SQL 2000 Analysis Service가 다른 두 가지 시스템에 비해 1~2초의 성능 저하를 보여 준다. 하지만 데이터 항목이 많고, 차원의 수가 하나 더 많아진 사례C의 경우 Oracle Express의 성능이 2~25초의 성능 저하를 보여준다.

특히 그리드 형태의 Range질의에 있어서 Oracle Express의 성능은 크게 저하 됨을 볼 수 있다. 이에 반해 Pivot연산에 있어서는 Oracle Express가 큐브 사이즈와 희박성 형태와 관계없이 가장 나쁜 성능을 보여줌을 확인할 수 있다.

결론적으로, 항목의 수와 차원의 수가 많아짐에 따라 희박성의 형태와 관계없이 랭킹 질의에서도 그리고 Dice와 Range, Pivot 질의에서도 Oracle Express의 성능이 다른 두 가지 시스템에 비해 현저하게 저하됨을 볼 수 있다.

5. 결론 및 향후 과제

CRM의 목적은 각 기업이나 회사들이 새로운 고객을 유치하고, 기존의 고객들을 유지하면서 장기간의 이익 관계를 이어 나가기 위해 고객의 취향과 필요를 이해하는 이다. CRM을 위해 웹 로그 데이터를 대상으로 OLAP을 적용할 시 사용자의 응답 성능을 빠르게 하기 위해 사전 연산은 반드시 이루어져야 하는 과정이다. 하지만 이러한 과정에서 데이터의 희박성이 커지고, 차원의 수가 많아짐에 따라 OLAP의 성능은 저하되고, 데이터 폭발(Data Explosion) 현상이 발생한다.

본 논문에서는 웹 로그 데이터를 분석하는데 있어서 OLAP을 적용할 시 희박성이 발생되는 원인에 대해 알아 보고, 이들 원인 중 주요한 요인에 대해 실제 로그 데이터를 사용하여 확인해 보았다. 그리고 2,3 차원내의 데이터 모델에서 그리드(Grid)와 클러스터(Cluster)형태의 희박성 형태가 주로 발생됨을 알 수 있었다. 분석된 희박성 형태를 기반으로 데이터 모델을 설계하고, 테스트 데이터 생성 모듈을 구현하여 데이터를 생성하였다. 7가지 OLAP의 일반적인 질의 모형과 Top500질의를 설계하고, 생성된 데이터와 함께 3가지 OLAP시스템(MS SQL 2000 Analysis Service, Oracle Express, C-MOLAP)에 대한 성능을 평가해 보았다. 성능 평가 결과 큐브의 사이즈가 적을 시에는 Top500연산, Dice연산, Range연산에 있어서 MS SQL 2000 Analysis Service의 성능이 다른 시스템들에 비해 약간의 성능 저하를 나타내지만, 차원의 수가 증가하고 항목의 수가 많아짐에 따라 3가지 연산에서 모두 Oracle Express의 성능이 현저하게 저하됨을 볼 수 있었다. 또한 Pivot연산에 있어서는 Oracle Express의 성능이 희박성 형태와 큐브 사이즈와 관계없이 가장 나쁜 성능을 보임을 알 수 있었다. 본 논문의 결과를 기반으로 다음과 같은 향후 연구 과제를 제시 할 수 있다.

- 실제 로그 데이터와 유사한 로그 데이터의 생성 모듈의 개발은 웹 로그 데이터를 사용하여 성능 평가를 하거나 Data Mining알고리즘에 적용하여 CRM을 하기 위한 작업의 기초가 될 수 있다.
- C-MOLAP시스템에 있어서 효율적인 청크의 사이즈를 결정 할 수 있는 알고리즘의 구현을 향후 과제로 제시 할 수 있다. 청크 사이즈가 너무 작게 나누어져 있을 시에는 각 청크를 가져오는데 디스크 I/O 타임이 많아지고, 순차 검색을 하는 것과 별반 다를 것이 없는 결과를 가져 오게 된다. 이에 반해 청크 사이즈가 너무 클 시에는 하나의 디스크 페이지 내에 불필요한 여과 용량을 낭비 할 수 있다.
- 본 논문에서 분석한 웹 로그 데이터의 희박성 형태와 성능 평가 결과를 기반으로 이들을 저장하고,

처리하기 위한 일반화되고 효율적인 웹 로그 데이터 분석기의 개발을 향후 연구 과제로 제시 한다.

참 고 문 헌

[1] MaxScan Corp, White Paper. "An Accelerator for Click Stream Data Analysis Applications".

[2] *White Paper*, http://www.olapreport.com/DatabaseExplosion.htm

[3] Michael L. Gonzales, "Estimating the Explosion of Derived Cells", DB2 magazine, pp 14-17, 2000.

[4] Sanjay Goil and Alok Choudhary, "Sparse Data Storage of Multi-Dimensional Data for OLAP and Data Mining", Technical Report CPDC-TR-9801-005, Center for Parallel and Distributed Computing, Northwestern University, 1997.

[5] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava, "Data Preparation for mining world wide web browsing patterns", the Journal of Knowledge and Information System, Vol. 1, No. 1, 1999.

[6] Y. Zhao, P. M. Deshpande, and J. F. Naughton, "An Array-Based Algorithm for Simultaneous Multidimensional Aggregates", In Proc. of ACM SIGMOD, pp 159-170, 1997.

[7] MDAC(Microsoft Data Access Component)Documentation, http://www.microsoft.com/data/whatcom.htm

[8] Oracle Corp, "Oracle Express Objects User's Guide Release 6.3", Aug 1999.

[9] Oracle Corp, "Oracle Express Programmer's Guide to the Express Language Release 6.3", Sep 1999.

[10] Microsoft Corp, "Microsoft Corp. SQL Server 2000 Analysis Services", Nov 2000.

[11] Ju-young Kang, "Classification of sparsity patterns and performance evaluation in OLAP system", the Master's Thesis of Department of Computer Science and Engineering, Ewha Institute of Science and Technology, 2000.

[12] Sanjay Goil and Alok Choudhary, "Sparse Data Storage of Multi-Dimensional Data for OLAP and Data Mining, *Technical Report CPDC-TR-9801-005*, Center for Parallel and Distributed Computing, Northwestern University, 1997.

[13] Robert J. Earle. Arbor Software corporation, "Method and Apparatus for Storing and Retrieving Multi-dimensional Data in Computer Memory", *U.S. patent* #5359724, Oct. 1994.

김 지 현

1999년 전남대학교 전산학과 학사

2002년 이화여자대학교 컴퓨터학과 공학석사

2002~현재 동양 Systems 연구원

관심분야 : OLAP, CRM

용 환 승

1983년 서울대학교 컴퓨터공학과 학사

1985년 서울대학교 컴퓨터공학과 공학석사

1985년~1989년 한국전자통신연구소 연구원

1994년 서울대학교 컴퓨터공학과 공학박사

2002년 IBM T.J. Watson Research Center Visiting Researcher

1995년~현재 이화여자대학교 컴퓨터학과 부교수

관심분야 : 웹기반 멀티미디어 데이터베이스시스템, 데이터마이닝, XML 데이터베이스

교 신 저 자

용 환 승 120-750 서울시 서대문구 대현동 이화여자대학교 컴퓨터학과