

데이터베이스 벤치마크를 위한 데이터 생성기와 성능 평가

옥 은 택[†] · 정 회 진[†] · 이 상 호^{††}

요 약

데이터베이스 벤치마크는 많은 양의 데이터에 대한 효과적 생성을 요구한다. 본 논문은 개발된 데이터 생성기의 시스템 구성도, 데이터 생성 동작, 특성을 기술한다. 본 데이터 생성기 특성은 대용량 데이터 생성 지원, 칼럼 단위 데이터 생성, 다양한 데이터 분포 지원 및 검증, 실 데이터 생성 등이다. 또한 타 데이터 생성기들과 기능 관점에서 비교하였다. 마지막으로 대용량 데이터 생성 시의 입출력 병목현상 해결을 위해 RAID 시스템과 비 RAID 시스템간의 성능을 실험적으로 비교하였으며, 실험 결과에 근거하여 시스템 구성 가이드를 제시한다.

A Data Generator for Database Benchmarks and its Performance Evaluation

Eun Taek Ok[†] · Hoe Jin Jeong[†] · Sang Ho Lee^{††}

ABSTRACT

Database benchmarks require efficient generation of large-scale data. This paper presents the system architecture, control flows, and characteristics of the data generator we have developed. The data generator features generation of large-scale data, column-by-column data generation, a number of data distributions and verification, and real data generation. An extensive comparison with other data generators in terms of function is also presented. Finally, empirical performance experiments between RAID systems and non-RAID one have been conducted to alleviate I/O bottleneck. The test results can serve as guidelines to help configure system architecture.

키워드 : 데이터 생성기(Data Generator), 데이터베이스 벤치마크(Database Benchmarks), 성능 평가(Performance Evaluation), RAID

1. 서 론

데이터베이스 시스템이 개발되면 벤치마크 시험을 통하여 개발된 시스템의 성능 및 기능을 검증하여야 한다. 데이터베이스 벤치마크 시험에서는 대용량의 실 데이터(real data)를 사용하는 것이 효과적이나, 대용량의 실 데이터를 얻기 위해서는 많은 시간과 비용이 소모된다. 따라서, 효과적인 데이터베이스 벤치마크 시험을 위해서는 대용량 실 데이터의 종류 및 특성과 비슷한 합성 데이터(synthesized data)를 생성하거나 또는 실 데이터의 일부분을 얻는 방법이 필요하다. 이를 지원하는 데이터 생성기는 데이터베이스 벤치마크 시험에 유용하다.

현재까지 데이터 생성기에 대한 연구는 [1, 2]를 비롯한 다수의 문헌에서 발견되고 있다. [1]은 대용량 데이터 생성의 성능 향상을 위해 다중 프로세서로 구성된 두 개의 시스템

을 구성하여 각 프로세서 단위의 데이터 병렬 생성 방법을 제시하였고, TPC-A 벤치마크 데이터를 이용하여 데이터 생성 알고리즘에 대한 성능 측정을 수행하였다. [2]는 유일한 랜덤 값 생성 알고리즘을 기술하고 있다. 데이터 생성기에 관한 다양한 연구 결과를 기반으로 Datatect[3], DataFactory [4], TurboData[5], DatGen[6] 등 다수개의 데이터 생성기 상용 제품이 현재 출시되어 있다.

대용량 데이터 생성에는 데이터 생성 알고리즘과 데이터 양에 따라 차이가 있겠지만 일반적으로 많은 시간이 소요된다. 특히, 성능에 가장 크게 영향을 미치는 부분은 디스크 입출력 작업이다. 디스크 입출력 작업의 성능 향상을 위해 전산학 분야에서는 디스크 캐싱(disk caching), 디스크 스케줄링(disk scheduling), 디스크 인터리빙(disk interleaving)[7], RAID(redundant arrays of inexpensive disks) 시스템[8] 등에 관한 연구가 있었다.

RAID 시스템은 RAID 계층(level)과 스트라이핑(striping) 크기에 따라 성능 차이가 난다. [9]는 다중 사용자 수, 요구 크기, 요구 시작 위치에 따른 작업 부하를 바탕으로 최적의

* 본 연구는 숭실대학교 교내 연구비 지원으로 이루어졌습니다.

† 준 회 원 : 숭실대학교 대학원 컴퓨터학과

†† 정 회 원 : 숭실대학교 컴퓨터학부 교수

논문접수 : 2003년 6월 27일, 심사완료 : 2003년 9월 8일

스트라이핑 크기에 대한 공식을 제시하였고, [10]은 입출력 작업 부하의 특성(읽기 특성, 쓰기 특성), 동시에 발생하는 작업량의 평균, 디스크 수, 다중 사용자 수 등의 변화에 대하여 계층 5의 RAID 시스템에 대한 최적 스트라이핑 크기를 결정하기 위한 실험을 수행하였다. [11]은 화일 입출력을 위해 요구되는 초당 입출력 크기와 화일 크기에 따라 스트라이핑 크기를 어떻게 변경할 것인지에 관해서 연구를 하였고, [12]는 계층 0와 계층 5의 RAID 시스템에서 미디어 화일 읽기 연산에 대하여 스트라이핑 크기에 따른 데이터 전송률을 측정하였다.

본 논문은 데이터베이스 벤치마크를 위한 데이터 생성기를 기술한다. 구현된 데이터 생성기는 데이터베이스 벤치마크 시험을 위한 데이터 제공을 위해 다양한 기능과 대용량 데이터 생성을 지원한다. 본 데이터 생성기가 가지는 특성은 칼럼 단위 데이터 생성, 다양한 데이터 분포 지원 및 검증, 실 데이터 생성 등이다. 이러한 특성들과 타 데이터 생성기가 가지는 특성 및 기능들을 비교 분석하였다. 또한 대용량 데이터 생성시 디스크 입출력 작업의 성능 향상을 위해 RAID 시스템을 구성하여 본 데이터 생성기에 관한 성능 실험을 하였고, RAID로 구성되어 있지 않는 시스템(이하, No_RAID 시스템)과 성능을 비교 분석하였다. 이러한 성능 평가 결과를 기반으로 사용자가 대용량 데이터를 생성하고자 할 때 참고할 수 있는 튜플 개수, 튜플 길이, 칼럼 개수와 관련된 시스템 구성 가이드를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 효과적인 데이터베이스 벤치마크 시험을 위해 데이터 생성기에 요구되는 기능 및 구조와 분포 데이터 및 검증 방법, 타 데이터 생성기와의 비교 분석을 한다. 3장에서는 데이터 생성에 따른 성능 평가 실험의 환경 및 내용 등에 대해 설명하고, 실험을 통해 얻어진 결과를 기술, 분석한다. 마지막으로, 4장에서는 본 논문의 결론과 향후 계획에 대하여 기술한다.

2. 데이터 생성기

주요 데이터베이스 성능 평가 방법론 - TPC-C 벤치마크 [13], 위스콘신(Wisconsin) 벤치마크[14], Set Query 벤치마크[15], AS³AP 벤치마크[16] 등 - 에서 요구하는 데이터 특성은 다음과 같다. 첫째, 유일한 값(unique value)의 데이터 생성이 필요하다. 특히, 위스콘신 벤치마크의 경우 유일한 값이 랜덤(random)하게 생성되어야 한다. 둘째, 문자열 데이터에 대해서는 임의 선택 추출을 지원하여야 한다. 예를 들어, 문자열 데이터 중 특정 문자열이 사용자가 지정하는 비율로 선택 추출될 수 있도록 생성되어야 한다. 셋째, 생성 가능한 데이터 형으로는 최소한 정수형, 실수형, 문자형

이 필요하다.

2.1 기능 및 구조

본 데이터 생성기는 최대 1억개까지 튜플 생성이 가능하고, 유일한 랜덤 값[2]을 생성할 수 있으며, 선택도(selectivity) 기능을 제공하여 다양한 선택 추출 질의 평가 및 범위 질의 평가를 효과적으로 수행할 수 있도록 지원한다. 또한, 지정된 크기의 문자열 데이터 생성이 가능하고, 정수형, 실수형, 문자형의 데이터 형을 지원한다.

본 데이터 생성기의 주요 특성은 다음과 같다. 첫째, 주어진 비율에 의거하여 널 값 생성 기능을 제공한다. 실 데이터에서는 널 값이 칼럼 값으로 존재하므로 실 데이터와 유사한 데이터 제공을 위하여 반드시 필요하다. 둘째, 생성된 튜플을 무작위로 섞는 셔플링(shuffling) 기능을 제공한다. 유일한 랜덤 값 생성을 하는 경우에는 일반적으로 생성된 랜덤 값들이 동일한 순서로 같은 값을 갖기 때문에 유용한 데이터 생성을 위해서는 셔플링 기능이 필요하다. 셋째, 생성된 분포 데이터에 대한 검증 기능을 지원함으로써 데이터에 대한 신뢰성을 높인다.

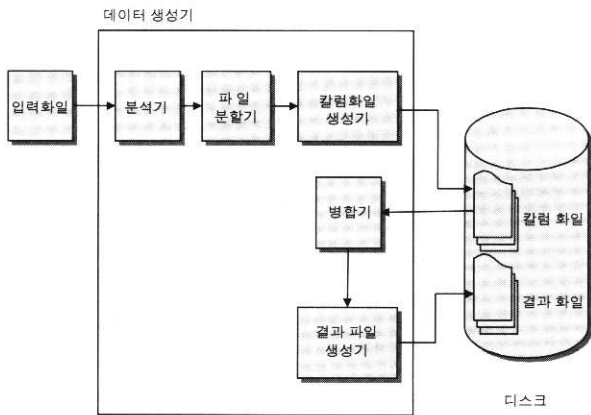
본 데이터 생성기는 데이터 분포의 지원을 통해 다양한 데이터 생성을 제공한다. 포아송 분포, 지프 분포, 정규 분포 등 확률 분포가 포함된다. 분포의 특성을 따르는 데이터 생성 기능은 정수형, 실수형 외에 문자형 데이터형도 지원한다. 또한 전화번호, 우편번호, 주민등록번호 등의 데이터 생성 기능을 제공한다. 전화번호 생성은 지역번호와 국번을 가지는 유선전화번호 및 무선(휴대) 전화번호에 대한 데이터 생성을 가능하게 하고, 우편번호 생성은 실제 우편번호 및 일반 주소를 기반으로 데이터 생성을 한다. 주민등록번호는 실제 사용 가능한 주민등록번호 생성이 가능하며, 사용자의 생년월일 선택 범위와 정해진 규칙에 따라 데이터가 생성된다.

본 데이터 생성기는 결과 화일을 사용자가 지정한 화일로 저장한다. 본 생성기에서 지원하는 테이블은 최대 20개까지 칼럼을 가질 수 있다. 그 이유는 TPC-C 벤치마크, 위스콘신 벤치마크 등 기존의 데이터베이스 벤치마크에서 하나의 테이블에 존재하는 칼럼 개수가 20개 이하이고, 사용자 인터페이스 관점에서의 테이블 및 칼럼 표현 제약성 때문이다.

본 데이터 생성기는 튜플 단위로 데이터를 생성(tuple-by-tuple generation)하지 않고, 칼럼 단위로 데이터를 생성(column-by-column generation)한다. 칼럼 단위 데이터 생성은 칼럼 각각에 대하여 데이터 생성을 한 후, 이를 통합하여 튜플을 생성하는 방식이다. 칼럼 단위 데이터 생성은 튜플 단위 데이터 생성에 비해 두 가지 장점을 제공한다. 첫

제, 칼럼 단위 서플링을 손쉽게 할 수 있다. 튜플 단위 데이터 생성 시 칼럼 단위로 서플링을 하기 위해서는 현재까지 생성한 데이터 분포를 기억하고 있어야 하는 오버헤드(overhead)가 있어, 대규모 데이터 생성시에는 메모리 자원에 따른 제한을 받을 수 있다. 둘째, 칼럼 단위 데이터 생성은 튜플 단위 데이터 생성보다 제한된 메모리 자원 내에서 대규모 데이터 생성이 용이하다.

(그림 1)은 본 데이터 생성기의 구조를 보인다. 입력 화일은 결과 화일 저장 디렉터리, 테이블 이름, 튜플 개수, 칼럼 개수, 칼럼 이름, 생성될 칼럼의 분포 정보, 널 값 정보, 서플링 수행 유무 등 데이터 생성에 필요한 정보를 가진다. 분석기는 입력 화일에 있는 각 정보들을 분석하고 이를 적절한 프로그램 변수에 저장한다. 화일 분할기에서는 분석된 각 칼럼 정보들을 기반으로 생성될 전체 화일 크기를 계산하여 운영체제에서 지원하는 화일 크기로 화일을 분할한다. 칼럼 화일 생성기에서는 분석기에서 프로그램 변수에 저장한 칼럼 정보 값을 바탕으로 칼럼 데이터를 생성하고, 디스크의 칼럼 화일에 각각 저장한다. 병합기는 디스크 상의 칼럼 화일을 병합하며, 결과 화일 생성기는 병합된 하나의 결과 화일을 디스크에 저장한다.

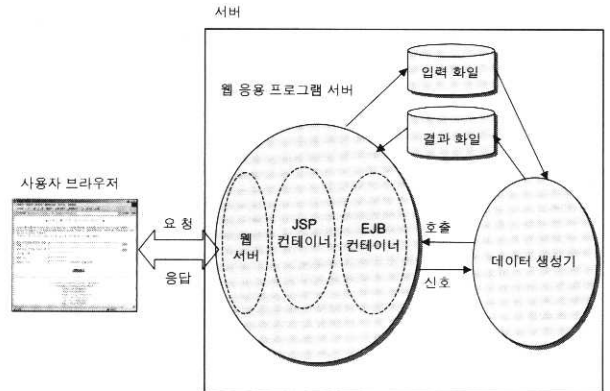


(그림 1) 데이터 생성기 구조

본 데이터 생성기를 통해 데이터를 생성하기까지는 두 번의 쓰기 연산과 한 번의 읽기 연산을 수행한다. 칼럼 화일 생성기가 칼럼 단위로 데이터를 생성할 때 한번의 쓰기 연산이 수행되고, 생성된 각각의 칼럼 화일이 병합기를 통해서 합쳐지고, 결과 화일 생성기를 통해서 최종 데이터를 가지는 결과 화일이 생성될 때 한번의 읽기 연산과 한번의 쓰기 연산이 수행된다. 이러한 구조는 성능 평가 수행 결과를 분석하는데 중요한 열쇠가 된다.

사용자 인터페이스는 웹 환경이다. 관련 웹 페이지는 테이블 정보 입력 페이지, 칼럼 정보 입력 페이지, 서버 경로 페이지, 데이터 수행 결과 성공 여부 페이지, 결과 정보 페

이지, 에러 페이지 등 총 6개로 구성된다. 테이블 정보 입력 페이지와 칼럼 정보 입력 페이지는 입력 화일 내용이 입력된다.



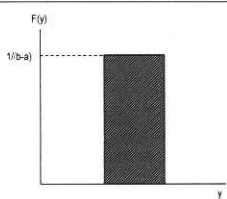
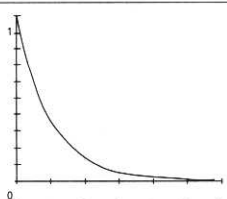
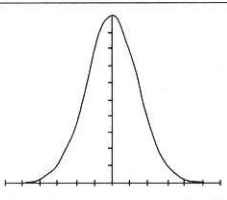
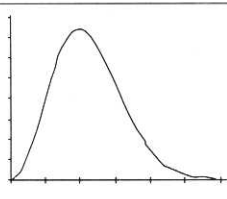
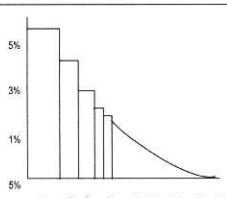
(그림 2) 데이터 생성기 운영

(그림 2)는 데이터 생성 과정을 운영체제 프로세스 관점에서 보인다. 사용자 브라우저의 데이터 생성 관련 웹 페이지는 HTML 및 JSP(Java server pages)로 작성되었다. 웹 응용 서버(web application server)는 웹 서버, JSP 컨테이너(container), EJB (enterprise Java beans) 컨테이너 등 다양한 요소로 구성되고, 하나의 프로세스로 실행된다. 데이터 생성기는 독립된 프로세스로 실행된다. 웹 응용 서버는 사용자 브라우저로부터 테이블과 칼럼 정보를 입력 받아 데이터 생성기에 사용될 입력 화일을 만든 후, 데이터 생성기를 호출한다. 데이터 생성기는 입력 화일의 정보를 바탕으로 데이터를 생성한 후 결과 화일에 데이터를 저장하고, 웹 응용 서버에 종료 신호를 보낸다. 웹 응용 서버는 종료 신호를 받은 후, 데이터 생성기가 생성한 결과 화일의 일부 내용을 사용자 브라우저를 통해 보여준다.

2.2 분포 데이터 및 검증

본 데이터 생성기에서 지원하는 데이터 분포는 선택도 분포, 상수 분포(constant distribution), 일련번호 분포(sequential distribution), 랜덤 분포, 균일 분포(uniform distribution), 정규 분포(normal distribution), 지수 분포(exponential distribution), 포아송 분포(Poisson distribution), 지프 분포(Zipf distribution) 등 모두 9가지이다. 선택도 분포는 전체 데이터 중에서 선택도만큼 동일한 특정 데이터를 생성하고, 나머지는 동일한 특정 데이터를 제외한 의미 없는 데이터를 생성한다. 본 데이터 생성기는 선택도 분포를 문자열 데이터 형에서 지원한다. 상수 분포는 사용자가 지정한 상수 값을 생성하며, 랜덤 분포는 유일한 랜덤 값을 생성한다. 균일 분포는 유한 범위 내의 모든 값들이 발생할

확률이 동일한 분포이므로, 사용자가 지정한 최소값과 최대 값 내에서 균일 분포의 특성에 맞는 데이터를 생성한다. (그림 3)은 본 데이터 생성기가 지원하는 분포 중 일부 분포에 대한 확률 밀도 함수와 분포 그래프를 보여준다.

분 포	확률 밀도 함수	분포 그래프
균일 분포	$f(x) = \frac{1}{b-a} \quad x \in [a, b]$	
지수 분포	$f(x) = \frac{1}{\lambda} e^{-\lambda x}, \quad x \in [0, \infty]$	
정규 분포	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \quad x \in [-\infty, \infty]$	
포아송 분포	$f(x) = \frac{e^{-\lambda} \lambda^x}{x!}, \quad x = 0, 1, 2, \dots$	
지프 분포	없 음	

(그림 3) 분포에 따른 확률 밀도 함수와 그래프

정규 분포, 지수 분포, 포아송 분포, 지프 분포는 [1]의 알고리즘에 따라 구현되었다. 정규 분포는 평균을 중심으로 편차만큼의 데이터가 대칭을 이루면서 생성된다. 지수 분포는 한 사건이 일어난 뒤 다음 사건이 일어날 때까지의 시간 간격을 나타내며, (그림 3)의 확률 밀도 함수에서 모수인 'λ'의 변화에 따라서 데이터 값이 달라지게 된다. 예를 들어, 고객이 은행에 들어오는 시간 간격을 나타낼 때 사용된다. 포아송 분포는 단위 시간당 발생하는 사건의 횟수를 나타내는 분포이며, (그림 3)의 확률 밀도 함수에서 모수인 '1/λ'의 변화에 따라서 데이터 값이 달라지게 된다. 지프

분포는 지프 법칙 - 어떤 문헌 내에 i번째 단어의 빈도가 문헌 내의 최고 빈도 단어의 '1/i^θ'배로 나타나는 특성을 가지는 법칙 - 을 따르는 분포이며, 'θ' 값에 따라서 데이터 값이 달라지게 된다.

본 데이터 생성기는 생성된 분포 데이터에 대한 데이터 검증 기능을 제공한다. 검증 작업은 두 단계로 이루어지며, 첫 번째 단계에서 생성된 데이터를 튜플 개수 관점에서 확인하고, 두 번째 단계에서는 생성된 데이터가 분포 특성에 맞는 지 검증한다. 분포 특성에 맞게 생성 되었는지에 대한 검증은 선택도 분포, 일련번호 분포, 상수 분포, 균일 분포, 정규 분포, 지수 분포, 포아송 분포에 한해서 수행되고, 랜덤 분포, 지프 분포는 특별한 검증 방법이 없기 때문에 첫 번째 단계만 수행한다.

각 분포 특성에 따른 데이터 생성 검증 방법은 다음과 같다. 선택도 분포에서는 전체 튜플 개수에서 선택도만큼의 튜플 개수를 계산하여 해당 개수만큼 특정 문자열이 생성되었는지를 확인한다. 일련번호 분포는 0부터 시작하는 정수 값을 생성하므로 마지막 생성된 데이터가 전체 튜플 수보다 하나 적은 값으로 생성되었는지 여부를 확인한다. 상수 분포는 여러 번의 실험을 통해 적절한 검증 시간을 보인 튜플 개수 20만개를 기준으로 그 검증 방법을 달리한다. 튜플 개수 20만개 미만은 전체 데이터를 검증하고, 20만개 이상은 10만개 단위로 샘플링 검사를 수행하여 검사된 모든 값이 동일한 상수가 나오는지 확인한다. 균일 분포는 생성될 데이터의 최소값과 최대값의 범위에서 데이터 생성 확률에 따른 최소값의 개수가 맞게 생성되었는지 검증한다.

정규 분포, 지수 분포, 포아송 분포는 통계학에서 지원하는 적합도 검증 방법[17]을 사용하여 데이터 분포를 검증한다. 적합도 검증은 모집단이 특정한 분포 형태를 따른다는 가설에 대해 표본의 도수 분포를 이용하여 검증하는 방법이다. 적합도 검증은 'x²-검정 통계량'에 근거하여 식 (1)의 공식에 따라 수행된다.

$$x_0^2 = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}, \quad x_0^2 \geq x^2(k-1; \alpha) \quad (1)$$

식 (1)에서 '(k-1)'은 'x²-검정 통계량'의 자유도를 의미하고, 'α'는 유의 수준의 정도를 나타낸다. 관측 도수 (O_i)는 생성된 데이터를 통해 값을 얻게 되고, 기대 도수 (E_i)는 생성될 것으로 예상되는 값을 계산하여 얻게 된다. 예를 들어, 생성된 데이터가 정규 분포를 따르는지에 관한 적합도 검증을 수행하고자 할 때, 얻고자 했던 평균값이 기대 도수가 되고, 데이터 생성기를 통해 얻어진 데이터의 값들이 관측 도수가 된다.

2.3 타 데이터 생성기와의 비교 분석

상용 데이터 생성기 중 'Datatect', 'DataFactory', 'TurboData', 'DatGen' 4개 제품의 특징을 살펴보면 다음과 같다. 'Datatect'은 다양한 데이터 형을 지원하며, 전자 메일이나 신용 카드 번호 생성 등 다양한 데이터 생성이 가능하다. 'DataFactory'는 데이터베이스로의 직접 접근이 가능하고, 사용하는 데이터베이스의 스키마 정보를 통해서 데이터 생성이 가능하다. 간단한 사용자 인터페이스를 사용하고 있어 편리성에서 다른 제품에 비해 앞서 있다. 'TurboData'는 제품을 통하여 스키마를 만들 수 있으며, 프로그램을 실행할 때는 반드시 만들어진 스키마를 기반으로 해야 한다. 'DatGen'은 유일하게 웹 사용자 인터페이스를 이용하였으나, 다양한 데이터 생성이 불가능하다는 단점이 있다.

상용 데이터 생성기들을 데이터베이스 벤치마크 시험을 위한 데이터 종류와 데이터 생성 관점에서 살펴보면, 본 논문의 데이터 생성기와는 <표 1>과 같은 차이점이 있다. '○'는 해당 기능을 지원함을 나타내고, '×'는 해당 기능을 지원하지 않음, 'N/A'는 해당 기능의 지원 여부를 확인할 수 없음, 그리고 '△'는 직접적으로 해당 기능을 지원하지는 않으나 다른 기능들의 조합을 통해 관련된 데이터의 생성이 가능함을 의미한다.

<표 1> 타 데이터 생성기 제품과 비교

	Datatect	Data Factory	Turbo Data	DatGen	Ours
널 값 유무	×	○	×	×	○
문자열 크기 설정	○	○	○	×	○
선택도	○	△	×	×	○
결과 파일 저장	○	○	○	○	○
칼럼 파일 저장	×	×	×	×	○
랜덤 값	○	○	○	○	○
서플링	×	×	×	×	○
분포 데이터 생성	×	×	×	○	○
웹 인터페이스	×	×	×	○	○
문자 데이터 분포	×	×	×	×	○
분포 데이터 검증	×	×	×	N/A	○

선택 추출이 가능한 데이터 생성을 위해 지원되는 선택도 기능의 경우, 'Datatect'은 생성되는 모든 문자열 데이터에 대해 지정된 선택도만큼 동등하게 생성되는 반면, 본 데이터 생성기는 시스템에서 정한 특정 문자열에 대해서만 선택도만큼의 데이터가 생성되는 차이점이 있다. 생성된 데이터를 아스키(ASCII) 파일로 저장하는 결과 파일 저장 기능은 모두 지원한다. 여러 가지 분포에 따른 데이터 생성

기능은 'DatGen'과 본 데이터 생성기에서만 지원되는 기능이다. 하지만, 'DatGen'의 경우 균일 분포와 정규 분포만 지원하기 때문에 보다 다양한 분포 데이터 생성을 지원하는 본 데이터 생성기에 비해 분포 지원에 있어 한정적이다.

각 칼럼마다 생성된 칼럼 데이터를 파일로 저장하는 기능, 칼럼 단위의 서플링 기능, 분포의 특성을 따르는 문자 데이터 생성 기능, 데이터 신뢰성을 높이기 위한 분포 데이터의 검증 기능 등은 본 데이터 생성기에서만 지원하는 기능이다.

3. 성능 평가 및 분석

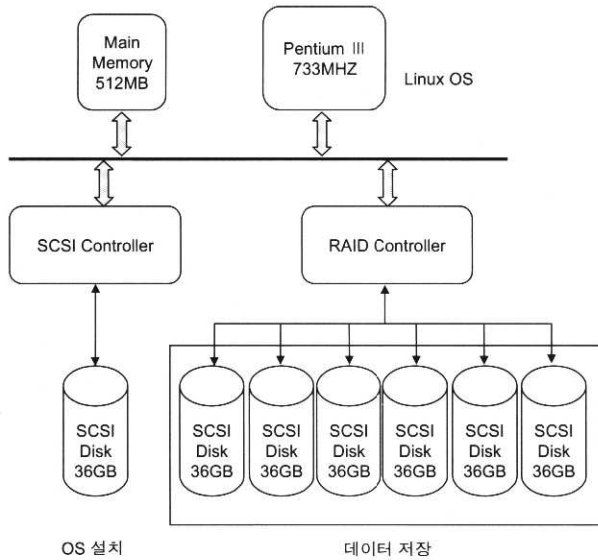
본 데이터 생성기는 칼럼 단위 데이터 생성을 수행한다. 즉 각각의 칼럼을 생성하여 디스크에 쓰기 작업을 수행하고, 디스크에 저장된 칼럼 파일들을 읽어 하나의 파일로 병합하여 최종 결과 데이터를 디스크에 쓰는 작업을 수행한다. 대용량 데이터 생성시 입출력 작업을 많이 수행하게 되므로 입출력 작업의 성능 향상을 위해 RAID 시스템을 구성하고, 이에 대한 성능 평가를 수행하였다. 성능 평가 및 분석을 위해 데이터 생성을 수행하는데 있어 RAID 시스템의 계층과 스트라이핑 크기에 따른 입출력 성능의 향상 정도를 실험하였고, No_RAID 시스템과의 성능 비교 분석을 수행하였다. 이 결과를 바탕으로, 본 논문에서는 데이터 생성 성능에 영향을 미치는 튜플 개수, 튜플 길이, 칼럼 개수 관점에서 사용자가 생성하고자 하는 데이터 특성에 맞는 적절한 RAID 시스템 구성 가이드를 제시한다.

RAID 시스템에는 여러 계층이 존재한다. 그 중에서 계층 0은 빠른 성능이 요구될 때 사용되지만, 데이터 손실 가능성을 가진다. 계층 1은 미러링(mirroring) 기능을 가진 계층으로서, 복구 능력은 탁월하나 처리 속도가 늦고 비용이 많이 드는 단점이 있다. 계층 0+1은 계층 0과 계층 1의 장점이 통합된 계층이다. 계층 5는 데이터에 대한 패리티(parity) 정보를 여러 디스크에 분산 저장함으로써 안정성 측면에서 가장 큰 효과를 볼 수 있고, 가장 많이 사용되는 계층이다. 본 논문에서는 계층 0, 계층 5에 대해서 데이터 생성 성능 평가를 수행하였다.

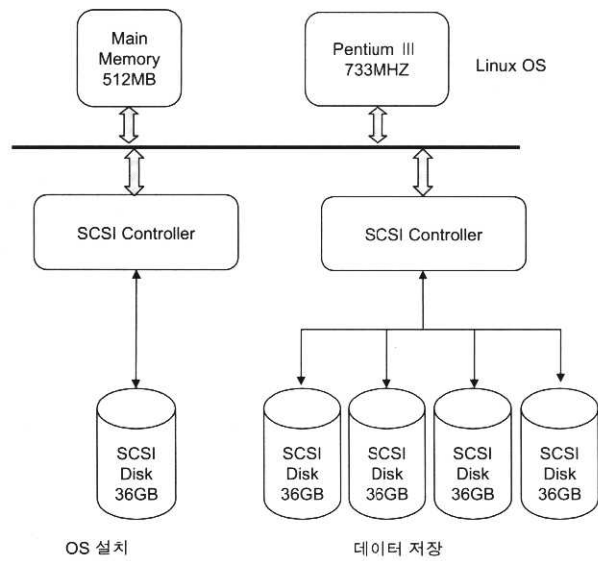
실험 시스템 환경은 다음과 같다. RAID 제어기(controller)는 'Compaq Smart Array 5302' 제품으로 2개 채널 및 32MB의 캐쉬 메모리를 가지고 있다. 또한, RAID 시스템의 계층 0, 계층 0+1, 계층 5를 지원하고, 8KB부터 256KB까지의 스트라이핑 크기를 사용할 수 있도록 지원한다. 실험 대상 서버의 운영체제로는 'Redhat Linux 7.3', 주 메모리는 512MB, 프로세서는 733MHz 2개, 하드디스크는 36GB 7개가 사용되었다. 웹 어플리케이션 서버는 'Web Logic 6.1'이

사용되었다.

(그림 4)는 RAID 시스템의 구성을 보이고 있다. RAID 제어기가 2개 채널을 지원하기 때문에 채널 1개당 3개의 외장 디스크를 제어토록 하여, 6개의 외장 디스크를 하나의 논리적 디스크로 구성하였다. (그림 5)의 No_RAID 시스템에서는 4개의 외장 디스크를 SCSI 제어기에 연결하였다.



(그림 4) RAID 시스템 구성



(그림 5) No_RAID 시스템 구성

튜플 개수, 튜플 길이 등의 데이터 특성을 고려한 실험 내용 구성은 <표 2>와 같다. 본 실험에서는 선택도 분포를 따르는 문자형 데이터를 사용하였고, 튜플 개수가 1억개인 경우에는 실험 장비의 디스크 용량 제한으로 인해 튜플 길이가 512바이트까지만 실험하였다. 칼럼 개수가 5개, 10개인 경우의 각 칼럼이 차지하는 튜플 길이는 칼럼간 구분자(delimiter)와 인용 부호(quotation mark)를 제외한 전체 길이를 각 칼럼에 같은 길이로 분배하고 나머지는 마지막 칼럼에 추가 할당하는 방법을 통해 맞추었다. 튜플 길이가 20바이트의 경우 칼럼 개수 10개에 대해서는 칼럼간 구분자와 문자형 부호가 차지하는 길이로 인해 실험이 불가능하여 제외하였고, 튜플 길이가 2048바이트의 경우 칼럼 개수 1개에 대해서는 일부 데이터베이스 시스템에서 데이터 로딩을 지원하지 않아 제외하였다.

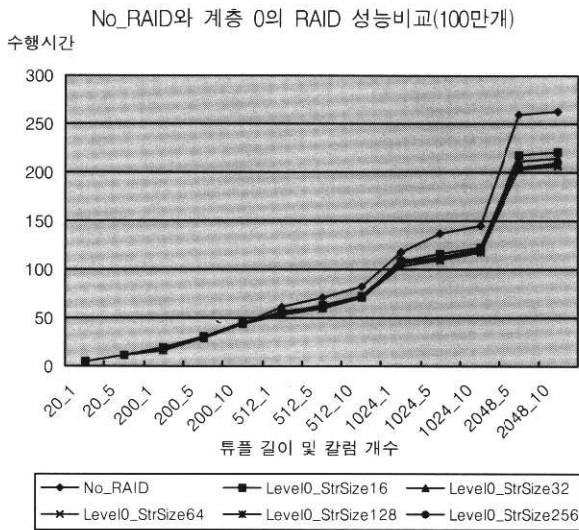
No_RAID 시스템에 대한 성능 평가를 위해 <표 2>에 기술된 실험을 모두 수행하였고, RAID 시스템에 대한 성능 평가를 위해 계층 0의 경우에는 스트라이핑 크기 16KB, 32KB, 64KB, 128KB, 256KB에 대해서, 계층 5의 경우에는 스트라이핑 크기 16KB, 32KB, 64KB에 대해서 <표 2>에 기술된 실험을 모두 수행하였다. 계층 5에 대한 스트라이핑 크기의 경우의 수가 적은 것은 실험에 사용된 RAID 제어기의 성능 제한 때문이다. 실험의 결과값은 사용자가 지정한 튜플 개수만큼 데이터를 생성하는데 소요되는 초(second) 단위 수행 시간이고, 3번의 실험 결과 값에 대한 중앙값(median)을 사용하였다. 결과값에 대한 분석은 No_RAID 시스템의 결과값을 기준으로 스트라이핑 크기에 따른 계층 0과 계층 5의 RAID 시스템에 대한 결과값을 비교하는 방법을 사용하였다.

(그림 6), (그림 7), (그림 8)은 No_RAID 시스템과 계층 0의 RAID 시스템에 대한 성능 평가 결과를 각각 튜플 개수 100만개, 1000만개, 1억개에 대해 보여준다. 그림에서 '20_1'은 칼럼 1개로 구성된 튜플 길이가 20바이트의 데이터를 의미하고, 'Level0_StrSize16'은 계층 0의 RAID 시스템에서 스트라이핑 크기가 16KB임을 나타낸다.

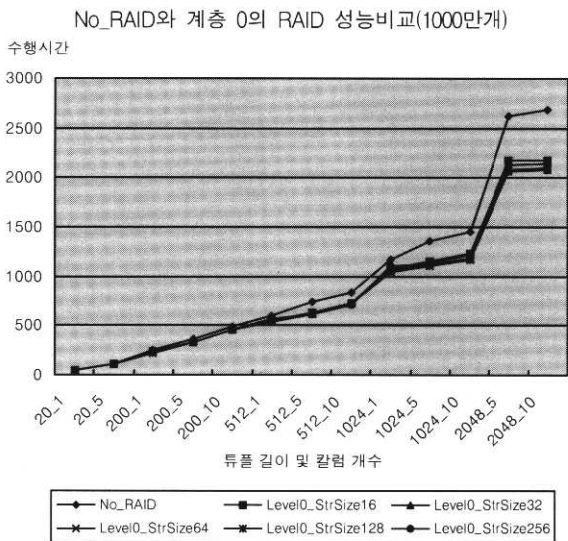
(그림 6) (그림 8)은 계층 0의 RAID 시스템 성능이 No_RAID 시스템의 성능보다 좋으며, 그 차이는 튜플 길이가 길어질수록 더 커지고 있음을 나타내고 있다. 튜플 길이가 길어지면 디스크에 읽고 쓰는 데이터 양이 많아지게 되고,

<표 2> 100만개, 1000만개, 1억개 데이터 특성에 따른 실험 내용

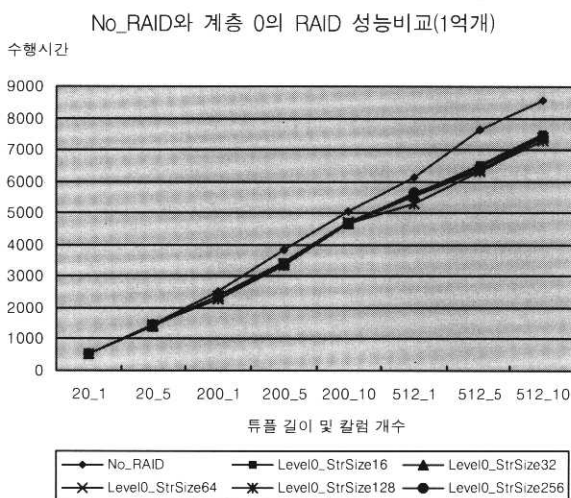
튜플 개수 \ 튜플 길이	20바이트	200바이트	512바이트	1024바이트	2048바이트
100만개	칼럼 1개	칼럼 1개	칼럼 1개	칼럼 1개	-
1000만개	칼럼 5개	칼럼 5개	칼럼 5개	칼럼 5개	칼럼 5개
1억개	-	칼럼 10개	칼럼 10개	칼럼 10개	칼럼 10개



(그림 6) 계층 0의 RAID 시스템에서 100만개 결과



(그림 7) 계층 0의 RAID 시스템에서 1000만개 결과



(그림 8) 계층 0의 RAID 시스템에서 1억개 결과

No_RAID 시스템보다는 스트라이핑 크기 단위로 디스크 입출력을 수행하는 계층 0의 RAID 시스템이 더 많은 양의 데이터를 읽고 쓸 수 있기 때문이다. RAID 시스템에서의 스트라이핑 크기에 따른 성능 차이는 크지 않았지만, 'Level0_StrSize128'이 가장 좋은 성능을 보였다. 튜플 개수가 100만개일 때, 튜플 길이가 2048바이트 이상에서 'Level0_StrSize128'는 No_RAID 시스템보다 20% 이상의 성능 향상 효과를 보였다. 튜플 개수가 1000만개일 때는 튜플 길이가 1024바이트 이상에서 No_RAID 시스템보다 약 11% 이상의 성능 향상 효과를 보였다. 또, 튜플 개수가 1억개일 때는 튜플 길이가 200바이트 이상에서 No_RAID 시스템보다 약 8% 이상의 성능 향상 효과를 보였다.

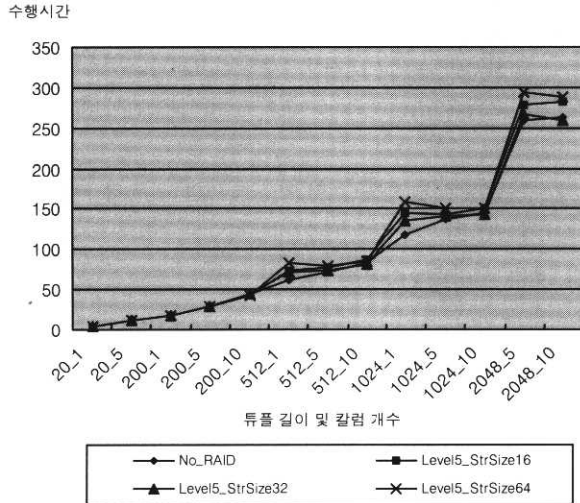
(그림 9), (그림 10), (그림 11)은 No_RAID 시스템과 계층 5의 RAID 시스템에 대한 성능 평가 결과들을 나타내고, (그림 9)는 튜플 개수가 100만개, (그림 10)은 1000만개, (그림 11)은 1억개에 대한 결과이다.

계층 5의 RAID 시스템에 대한 성능 평가 결과, 칼럼 별 데이터를 디스크에 쓸 때는 No_RAID 시스템과 RAID 시스템의 성능 차이가 없었으나, 칼럼 데이터를 병합하기 위해 데이터를 읽을 때 두 시스템간의 성능 차이가 발생하였다. 칼럼 개수가 1개 또는 5개인 경우는 No_RAID 시스템이 RAID 시스템보다 좋은 성능을 보였다. 그 이유는 계층 5의 RAID 시스템이 칼럼 데이터를 병합하기 위해 데이터를 읽을 때 패리티 정보를 갱신하는 시간이 No_RAID 시스템에서 화일 접근 시간(access time)에 소요되는 시간보다 많이 소요되기 때문이다. 칼럼 개수가 10개일 때는 RAID 시스템의 성능이 No_RAID 시스템보다 좋게 나타났다. 이것은 칼럼 개수가 많아질수록 입출력 작업이 많아지기 때문에 계층 5의 RAID 시스템이 패리티 정보를 갱신하는데 사용되는 시간보다 No_RAID 시스템의 화일 접근 시간이 많이 소요되기 때문이다.

계층 5의 RAID 시스템은 스트라이핑 크기에 따라 성능 차이를 보였다. 'Level5_StrSize32'에서 가장 좋은 성능을 보였고, 'Level5_StrSize64'에서는 가장 좋지 않은 성능을 나타내었다. 이후 수행된 계층 5의 RAID 시스템 관련 성능 평가 실험에서는 이 결과를 반영하여 스트라이핑 크기를 32KB로 설정하였다.

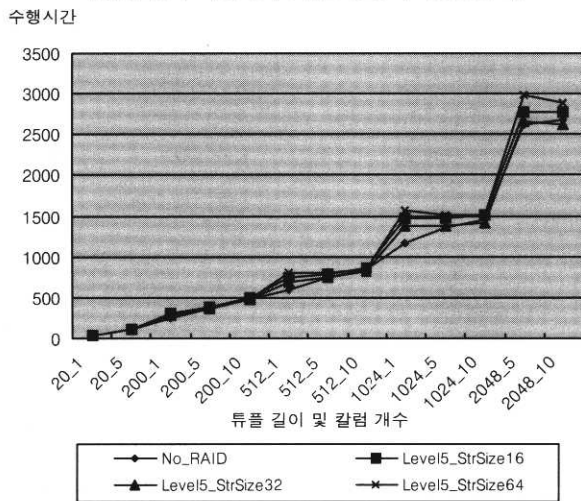
튜플 개수가 100만개와 1000만개인 경우, 튜플 길이가 200바이트 이하에서는 No_RAID 시스템과 'Level5_StrSize32'인 경우의 성능 차이가 거의 없었다. 하지만, 튜플 길이가 512바이트 이상에서 칼럼 개수가 1개일 때에는 No_RAID 시스템이 'Level5_StrSize32'보다 튜플 개수가 100만개일 때는 약 12%, 1000만개일 때는 약 14% 정도의 성능 향상 효과

No_RAID와 계층 0의 RAID 성능비교(100만개)



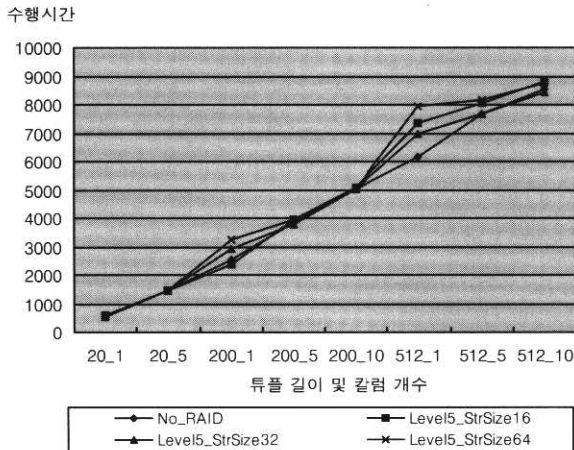
(그림 9) 계층 5의 RAID 시스템에서 100만개 결과

No_RAID와 계층 0의 RAID 성능비교(1000만개)



(그림 10) 계층 5의 RAID 시스템 1000만개 결과

No_RAID와 계층 0의 RAID 성능비교(1억개)



(그림 11) 계층 5의 RAID 시스템에서 1억개 결과

를 보였다. 이러한 성능 차이는 칼럼 개수가 많아질수록 줄어들었으며, 칼럼 개수가 10개일 때는 오히려 'Level5_StrSize32'가 No_RAID 시스템보다 튜플 개수 100만개와 1000만개에 대해서 약 2% 정도 더 좋은 성능을 보였다.

튜플 개수가 1억개인 경우, No_RAID 시스템은 튜플 길이 20바이트에서는 칼럼 개수에 관계없이 'Level5_StrSize32'와 성능 차이가 거의 없었지만, 튜플 길이가 200바이트 이상이고 칼럼 개수가 1개일 때는 'Level5_StrSize32'보다 약 13% 더 좋은 성능을 보였다. 또, 칼럼 개수가 10개일 때는 튜플 길이에 관계없이 'Level5_StrSize32'와 거의 성능 차이가 없었으나, 튜플 길이가 512바이트 이상인 경우에는 'Level5_StrSize32'보다 약 2% 정도 좋지 않은 성능을 보였다.

대용량 데이터 생성시 RAID 시스템을 구성하여 사용하고 자 할 때는 지금까지 언급된 성능 평가 실험의 결과에 근거한 다음과 같은 가이드 라인이 도움이 될 것이다. 계층 0의 RAID 시스템 구성은 튜플 개수에 따라 튜플 길이가 다음과 같은 경우 효과적이다. 튜플 개수가 백만개 정도인 경우에는 튜플 길이가 2048바이트보다 큰 경우이고, 튜플 개수가 천만개 정도인 경우에는 튜플 길이가 1024바이트 이상인 경우, 튜플 개수 1억개 이상은 튜플 길이에 상관없이 효과적이다. 이 때, 스트라이핑 크기는 128KB로 설정하는 것이 가장 효과적이다. 계층 5의 RAID 시스템 구성이 효과적일 때는 데이터 신뢰에 대한 중요성이 크고, 튜플 개수가 100만개 이상인 경우에는 그 개수에 관계없이 튜플 길이가 512바이트 이상이고, 칼럼 개수가 10개 이상인 조건을 만족하는 경우이다. 이 때에는 계층 0의 RAID 시스템과 달리 스트라이핑 크기를 32KB로 구성하는 것이 효과적이다.

4. 결 론

본 논문의 데이터 생성기는 데이터베이스 시스템 벤치마크 시험에 유용한 데이터를 다양하게 생성하는 기능을 지원한다. 칼럼 단위로 데이터를 생성하며, 한번 생성된 입력 화일과 결과 화일에 대한 재사용이 가능하다. 여러 가지 분포 특성을 따르는 숫자 데이터 및 문자 데이터 생성이 가능하며, 분포 특성을 따르는 데이터에 대한 검증 기능을 통해서 데이터에 대한 신뢰성을 제공한다. 또한 전화번호, 주민등록번호, 우편번호 등 실제 세계에서 사용되는 실제 데이터를 생성할 수 있는 기능을 제공한다.

대용량 데이터 생성과 관련하여 RAID 시스템과 No_RAID 시스템의 성능을 평가하여 비교한 결과, 계층 0의 RAID 시스템은 No_RAID 시스템보다 약 8~30% 정도의 성능 향상 효과가 있었으나, 계층 5의 RAID 시스템의 경우는 튜플 길이가 512바이트 이상이고, 칼럼 개수가 10개, 스트라이핑 크

기가 32KB인 경우에 한해서만 No_RAID 시스템 보다 약 2% 정도 성능 개선 효과를 보였다. 본 논문에서는 성능 평가 실험의 결과를 기초로 하여 벤치마크 시험을 위한 대용량 데이터 생성시 RAID 시스템을 구성하는 경우 효과적으로 데이터 생성을 수행할 수 있는 가이드라인을 제시하였다.

추가적으로, RAID 제어기의 캐쉬 메모리 비율을 배분하는 기능인 읽기-쓰기 비율의 조정을 통해 계층 5의 RAID 시스템에 대해 성능 평가 실험을 수행한 결과, 스트라이핑 크기가 32KB, 칼럼 개수가 5개 이상인 경우에서도 No_RAID 시스템 보다 약 2% 정도 성능 향상 효과가 있었다. 계층 0 + 1의 RAID 시스템 경우에는 스트라이핑 크기가 256KB일 때 다른 스트라이핑 크기를 사용한 경우에 비해 가장 좋은 성능을 보였다. 스트라이핑 크기 256KB를 사용한 경우, 칼럼 개수가 1개일 때는 계층 0 + 1의 RAID 시스템이 No_RAID 시스템보다 좋지 않은 성능을 보였지만, 칼럼 개수 5개 이상에서는 No_RAID 시스템보다 약 9~11% 이상 더 좋은 성능 향상 효과를 볼 수 있었다. 따라서, 데이터의 신뢰성을 유지하며 좋은 성능을 얻기 위해서는 읽기-쓰기 비율을 조정하고, RAID 시스템에서 계층 0 + 1을 사용하는 것이 계층 5를 사용하는 것보다 더 나은 성능을 기대할 수 있다.

본 논문에서 기술한 데이터 생성기는 현재 개발 중인 통합 데이터베이스 시스템 성능 평가 도구[18]의 일부분으로 사용되고 있다. 통합 데이터베이스 성능 평가 도구가 제공하는 사용자 정의 성능 평가는 사용자가 명시하는 사양을 만족하는 다양한 형태의 데이터 생성을 요구하며, 본 데이터 생성기는 이러한 요구 사항을 충족한다. 향후 연구로는 보다 다양한 실 데이터, 예를 들면, 이메일 주소, 신용카드 번호와 같은 데이터를 생성할 수 있는 기능을 제공하는 것과 데이터베이스 성능 평가를 위한 대용량 데이터 생성시 최적의 성능을 도출하는 RAID 시스템 구성 방안 설계 등이 있다.

참 고 문 헌

[1] J. Gray, P. Sundaresan, S. Englert, K. Baclawski and P. Weinberger, "Quickly Generating Billion-Record Synthetic Databases," Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, pp.233-242, 1994.

[2] D. Knuth, "The Art of Computer Programming," 2nd Ed., Addison Wesley, 1981.

[3] Datatect, Banner Software Inc, <http://www.datatect.com/>.

[4] DataFactory, Quest Software Inc, <http://www.quest.com/datafactory/>.

[5] TurboData, Canam Software Inc, <http://www.turbodata.ca/>.

[6] DatGen, <http://www.datasetgenerator.com/>.

[7] M. Y. Kim, "Synchronized Disk Interleaving," IEEE Transactions on Computers, Vol.3, No.11, pp.978-988, 1986.

[8] D. A. Patterson, G. Gibson and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data, pp.109-116, 1988.

[9] P. Chen and D. Patterson, "Maximizing Performance in a Striped Disk Array," Proceedings of the 1990 ACM SIGARCH International Conference on Computer Architecture, pp.322-331, 1990.

[10] P. M. Chen and E. K. Lee, "Striping in a RAID Level 5 Disk Array," Proceedings of the 1995 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, pp.136-145, 1995.

[11] G. Weikum and P. Zabback, "Tuning of Striping Units in Disk-Array-Based File Systems," Proceedings of the 2nd International Workshop on Research Issues on Data Engineering : Transaction and Query Processing, pp.80-87, 1992.

[12] 전상훈, 안병철, "실시간 멀티미디어 데이터를 위한 RAID 구조의 실측 분석", 정보처리학회논문지, 제9권 제2호, pp.191-199, 2002.

[13] TPC Home Page, <http://www.tpc.org>.

[14] D. DeWitt, "The Wisconsin Benchmark : Past, Present, and Future," The Benchmark Handbook, 2nd Ed., J. Gray Ed., Morgan Kaufmann, pp.269-316, 1993.

[15] P. O'Neil, "The Set Query Benchmark," The Benchmark Handbook, 2nd Ed., J. Gray Ed., Morgan Kaufmann, pp.359-396, 1993.

[16] C. Turbyfill, C. Orji, and D. Bitton, "AS³AP : An ANSI SQL Standard Scaleable and Portable Benchmark for Relational Database Systems," The Benchmark Handbook, 2nd Ed., J. Gray Ed., Morgan Kaufmann, pp.317-358, 1993.

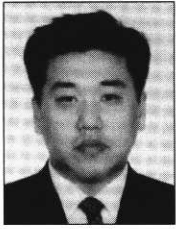
[17] 강근석, 김성철, 김지현, 이윤오, 이정진, 이창수, "디스켓이 들어 있는 PC 통계학", 자유 아카데미, 1993.

[18] H. J. Jeong and S. H. Lee, "An Integrated Benchmark Suite for Database Systems," Proceedings of the IASTED International Conference on Information Systems and Databases, pp.74-79, 2002.



옥 은 택

e-mail : romc0208@bcline.com
 2001년 숭실대학교 산업공학과(학사)
 2003년 숭실대학교 대학원 컴퓨터학과(석사)
 관심분야 : 데이터베이스 시스템 성능 평가 및 튜닝



정 회 진

e-mail : sinclear@dreamwiz.com

1993년 우석대학교 전산학과(학사)

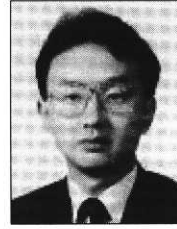
1995년 숭실대학교 대학원 컴퓨터학과(석사)

2002년 숭실대학교 대학원 컴퓨터학과
(박사수료)

1995년~2000년 (주)헨디소프트 기술연구소
선임연구원

2002년~2003년 숭실대학교 정보미디어기술연구소 전임연구원

관심분야 : 데이터베이스 시스템 성능 평가 및 튜닝, XML
데이터베이스



이 상 호

e-mail : shlee@comp.ssu.ac.kr

1984년 서울대학교 전산공학과 졸업(학사)

1986년 미국 노스웨스턴대 전산학과(석사)

1989년 미국 노스웨스턴대 전산학과(박사)

1990년~1992년 한국전자통신연구원 선임
연구원

1999년~2000년 미국 조지메이슨대 소프트웨어정보공학과
교환 교수

1992년~현재 숭실대학교 컴퓨터학부 교수

관심분야 : 인터넷 데이터베이스, 데이터베이스 시스템 성능
평가 및 튜닝