

데이터베이스 벤치마크를 위한 통합 도구

정 회 진[†] · 이 상 호^{**}

요 약

데이터베이스 개발자나 사용자는 새로운 데이터베이스 시스템이 개발되거나 기존 데이터베이스 시스템에 새로운 기능이 추가되면 벤치마크를 통해 해당 시스템이나 기능의 성능을 점검하게 된다. 본 논문에서는 데이터베이스 시스템을 위한 벤치마크 통합 도구를 제안한다. 벤치마크 통합도구는 범용 벤치마크, 주문 벤치마크, 혼합 벤치마크를 사용자가 하나의 사용자 인터페이스를 사용하여 쉽게 수행할 수 있도록 지원한다. 텍스트 데이터 생성과 관련하여, 벤치마크 통합도구는 3가지 데이터 형에 대해 8가지 데이터분포를 제공하여 실세계와 유사한 데이터 생성을 수행하도록 돕는다. 또한 벤치마크 통합도구에서는 3가지 서로 다른 방법을 사용하여 XML 데이터를 생성하도록 지원한다. 사용자들은 벤치마크 통합도구의 작업부하 생성기능을 사용하여 실세계와 유사한 환경에서 벤치마크를 수행할 수 있다. 벤치마크 통합도구는 사용자가 새로운 범용 벤치마크나 주문 벤치마크를 쉽게 구현할 수 있도록 지원한다. 본 논문에서는 새로운 주문 벤치마크를 벤치마크 통합도구에 쉽게 추가할 수 있음을 예를 통해 보인다.

키워드 : 데이터베이스 벤치마크, 벤치마크 도구, 통합도구, 주문 벤치마크

A Integrated Suite for Database Benchmarks

Hoe Jin Jeong[†] · Sang Ho Lee^{**}

ABSTRACT

As new database systems are developed or new functions are added to existing database systems, database developers or users would like to evaluate new database systems or new functions. This paper presents an integrated database benchmark suite. The integrated suite offers generic benchmarks, custom benchmark, and hybrid benchmarks to users on a unified Web user interface. With regard to text data generation, the integrated suite supports eight data distributions with three data types. The integrated suite can also generate XML data in three different ways. Users can run benchmarks in realistic environments by performing the workload generation facility of the integrated suite, which generates composite workloads similar to real-world workloads. Using supporting tools, users can easily implement new generic and custom benchmarks in the integrated suite. An illustrative demonstration to add a new custom benchmark into the integrated suite is presented.

Key Words : Database Benchmark, Benchmark Suite, Integrated Suite, Custom Benchmark

1. 서 론

새로운 데이터베이스 시스템이 개발되거나 기존 데이터베이스 시스템에 새로운 기능이 추가됨에 따라 데이터베이스 개발자나 사용자는 새로운 데이터베이스 시스템이나 새로운 기능을 여러 환경에서 체계적으로 평가하고자 한다. 이를 위해서는 데이터베이스 벤치마크가 필요하고, 지금까지 관련된 많은 연구가 이루어져왔다. 데이터베이스 벤치마크는 크게 범용 벤치마크(generic benchmark)와 주문 벤치마크

(custom benchmark)로 분류할 수 있다[7]. 범용 벤치마크는 어플리케이션 영역에서 공통적으로 사용되고 인지되고 있는 전형적인 예에 기반을 두어 개발된다. 주문 벤치마크는 정해진 사용자 환경에서 사용되는 특정 어플리케이션에 기반을 두어 사용자의 요구에 의해 개발되어진다. 널리 알려진 기존의 데이터베이스 벤치마크 중 대부분은 범용 벤치마크에 속한다. 하지만 범용 벤치마크는 모든 사용자의 요구사항을 완벽하게 만족시킬 수 없다는 단점을 가진다. 따라서 주문 벤치마크는 이러한 사용자의 특정 요구사항들을 만족시키기 위해 필요하다. 일반적으로 주문 벤치마크는 구현 비용이 많이 들지만, 사용자들은 주문 벤치마크를 통해 어떤 범용 벤치마크로도 평가되지 않을 특정 기능들에 대해 집중적으로 시험을 수행할 수 있다. 주문 벤치마크의

※ 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음. (KRF-2004-005-D00172)

† 준 회 원 : (주)헨디소프트 기술전략실, 수석연구위원

** 종 신 회 원 : 숭실대학교 컴퓨터학부 교수

논문접수 : 2005년 7월 19일, 심사완료 : 2006년 2월 7일

단점을 극복하기 위한 노력도 있다. 즉, 범용 벤치마크의 기본적인 구조에 기반을 두고 사용자 요구사항을 반영한 시험 질의를 추가하거나 수정하는 등의 노력을 통해서 보다 값싼 비용으로 주문 벤치마크를 개발하고자 한다. 이를 혼합 벤치마크(hybrid benchmark)라 부른다.

범용 벤치마크는 지금까지 많은 문헌을 통해 발표되었다. Wisconsin 벤치마크[5]는 관계형 데이터베이스 시스템을 위한 대표적인 벤치마크이다. Wisconsin 벤치마크는 관계형 데이터베이스 시스템에 의해 제공되어야 하는 기능들에 대해 벤치마크를 수행할 수 있도록 제안하고 있다. 관계형 데이터베이스 시스템을 위한 또 다른 벤치마크인 Set Query 벤치마크[10]는 전략 데이터 어플리케이션(strategic data applications)을 통해 성능 데이터를 요구하는 의사결정자(decision maker)를 돕기 위해 개발되어졌다. 객체지향형 데이터베이스 시스템의 출현과 더불어 OO1 벤치마크(Object Operations version 1)[4]와 OO7 벤치마크[3]가 개발되었다. OO1 벤치마크와 OO7 벤치마크는 CAD/CAM 어플리케이션과 같은 엔지니어링 데이터베이스 어플리케이션의 주요 기능과 관련된 객체지향형 데이터베이스 시스템의 성능을 측정하고자 하였다. BUCKY 벤치마크[1]와 BORD 벤치마크[9]는 객체관계형 데이터베이스 시스템의 성능을 측정하기 위해 제안되었다. BUCKY 벤치마크는 하나의 데이터베이스 시스템 내에서 관계형 데이터 모델과 객체관계형 데이터 모델에 의해 초래된 성능 차이에 초점을 맞추어 객체관계형 데이터베이스 시스템의 성능을 측정하고자 하였다. BORD 벤치마크는 객체관계형 데이터 모델에서 지원하는 독특한 기능들에 초점을 맞추어 해당 기능의 성능을 측정하고자 개발되었다.

약 20여개의 시스템 공급사가 모여서 구성된 컨소시엄(consortium)인 TPC(transaction processing performance council)[11]는 다양한 데이터베이스 어플리케이션에 기반을 두어 많은 데이터베이스 벤치마크를 제안하였다. 2005년 6월을 기준으로 현재 4개의 벤치마크가 실제 사용되고 있다. 4개의 벤치마크는 TPC-C, TPC-H, TPC-W, TPC-APP이다. TPC-C는 복잡한 온라인 트랜잭션 프로세싱 어플리케이션 환경에서의 데이터베이스 시스템 성능을 측정하고자 제안되었고, TPC-H는 사용자가 주로 사용하는 질의에 대한 사전 지식 없이 수행되는 의사결정 지원 어플리케이션 영역에서의 데이터베이스 시스템 성능을 측정하고자 개발되었다. TPC-W는 인터넷 환경에서 데이터베이스 시스템을 사용한 전자상거래 시스템의 성능을 측정하고자 개발되었다. TPC-APP는 가장 최근에 개발된 벤치마크로서, B2B(business-to-business) 트랜잭션이 주를 이루는 어플리케이션 서버의 수행을 시뮬레이션하는 환경에서의 어플리케이션 서버와 웹 서비스의 성능을 측정한다.

범용 벤치마크와 주문 벤치마크 모두 벤치마크 시험에서 최고의 성능을 획득하고자 다른 어떤 어플리케이션 프로그램도 허용치 않은 환경에서 벤치마크 시험을 수행한다. 하지만 이런 벤치마크 환경은 다양한 어플리케이션 프로그램

이 동시에 수행될 수 있는 실세계 환경과는 차이를 보인다. 이런 현실과 다소 동떨어진 벤치마크 환경에서 수행된 시험의 결과는 데이터베이스 시스템 구입 결정을 위한 올바른 척도로서의 역할을 수행치 못할 가능성이 높다. 자신의 사용 환경에 가장 적합한 데이터베이스 시스템을 구입코자 하는 사용자들은 보다 의미 있는 성능 결과를 얻기 위해 현실적인 실험 환경에서 데이터베이스 벤치마크를 수행코자 요구한다. 다양한 작업부하를 생성하여 벤치마크 환경을 보다 실세계 환경과 유사하게 만들 수 있는 작업부하생성 기능은 이러한 사용자의 요구사항을 만족시킬 수 있을 것이다.

데이터베이스 시스템에 대해 벤치마크를 수행하는 것은 쉬운 작업이 아니다. 데이터베이스 시스템을 벤치마크하기 위해 새로운 벤치마크를 설계하고 구현하기 위해서는 전문적인 지식을 필요로 하며, 많은 시간을 요구한다. 때로는 예상한 시간보다 더욱 많은 시간을 소요하며, 지속적으로 수정해나가야 하는 시간 소비적인 작업이다. 또한 벤치마크를 위해 꼭 필요한 데이터 생성 기능의 경우만 보더라도 명세(specification)에 언급된 다양한 요구사항에 맞춰 데이터 생성을 수행할 수 있는 데이터 생성기를 구현하는 것이 간단한 작업은 아니다. 즉, 일반 사용자들이 데이터베이스 시스템의 성능 측정을 위해 범용 벤치마크를 새로 개발하는 것은 쉽지 않다. 또한 혼합 벤치마크를 처음부터 개발하는 것은 더욱 어렵다. 이를 통해 사용자들의 손쉬운 벤치마크 구현 및 수행을 돕는 도구가 필요함을 알 수 있다.

본 논문에서는 데이터베이스 시스템을 위한 벤치마크 통합도구에 대해 기술한다. 벤치마크 통합도구는 범용 벤치마크, 주문 벤치마크, 혼합 벤치마크까지 모든 벤치마크 종류를 수행하기에 편리한 환경을 제공하고, 보다 실세계에 근접한 환경에서의 벤치마크 수행을 돕는다. 벤치마크 통합도구는 4종류의 범용 벤치마크와 4종류의 주문 벤치마크를 제공하고, 새로운 주문 벤치마크를 구현할 수 있는 환경을 제공한다. 또한 기존 제공되는 범용 벤치마크와 주문 벤치마크를 활용하거나 새로운 벤치마크 추가를 통해 혼합 벤치마크를 쉽게 수행할 수 있는 환경을 제공한다. 이 모든 벤치마크의 수행은 하나의 통합된 사용자 인터페이스를 통해 이루어지므로 벤치마크 사용자가 다양한 벤치마크 수행을 위해 여러 사용자 인터페이스를 익혀야 하는 번거로움을 제거하였다. 벤치마크 통합도구는 실세계 데이터와 유사한 데이터의 생성을 수행할 수 있도록 3가지 데이터 형과 8가지 데이터분포를 지원한다. 벤치마크 통합도구는 텍스트 데이터 생성은 물론 XML 데이터 생성도 가능하다. XML 데이터 생성 시 사용되는 생성방법에 따라 데이터분포의 지원도 가능하다. 벤치마크 통합도구는 실세계 환경과 유사한 벤치마크 환경 구성을 위해 작업부하 생성기능을 지원한다. 작업부하생성기능은 운영체제가 제어하는 자원 상태에 기초하여 이루어진다.

벤치마크 통합도구는 데이터베이스 개발자와 사용자 모두의 요구사항을 만족시킬 수 있도록 개발되었다. 데이터베이스 개발자는 자신들이 개발한 데이터베이스 시스템과 이미

사용되고 있는 다른 데이터베이스 시스템의 비교를 통한 상대적 성능을 알고자 하며, 이를 위해 범용 벤치마크를 일반적으로 수행하고자 한다. 한편 사용자들은 데이터베이스 시스템에서 자신들이 주로 사용하는 기능들에 대한 집중적인 시험을 통해 해당 기능의 우수성을 확인하고, 데이터베이스 시스템 구입결정 시 이를 반영하고자 주문 벤치마크를 주로 수행하고자 한다. 벤치마크 통합도구는 개발자와 사용자 모두의 요구 사항을 만족시키기 위해 하나의 도구를 통해 범용 벤치마크와 주문 벤치마크를 모두 수행할 수 있도록 설계 및 구현되었다.

본 논문의 구성은 다음과 같다. 2장에서는 벤치마크 통합도구의 설계단계에서 고려된 설계원칙에 대해 기술한다. 벤치마크 통합도구의 모듈구성과 파라미터 파일은 3장에서 설명한다. 텍스트 데이터 및 XML 데이터의 생성과 작업부하 생성도 3장에서 기술한다. 4장에서는 새로운 주문 벤치마크를 벤치마크 통합도구에 어떻게 쉽게 추가할 수 있는지를 보이며, 작업부하생성 기능을 활용하여 실세계와 유사한 환경에서 벤치마크를 수행하는 과정 및 실험 결과를 기술한다. 마지막으로 5장에서는 결론 및 향후 연구 계획을 기술한다.

2. 설계 원칙

벤치마크 통합도구는 데이터베이스 시스템 벤치마크 수행 시 대부분의 사용자들이 직면하고 있는 어려운 점을 인식하고, 벤치마크에 대한 사용자들의 다양한 요구사항을 효과적으로 만족시킬 수 있도록 구조기반에서의 접근을 통한 노력의 결과이다. 즉, 벤치마크 통합도구의 목적은 사용자들에게 다양한 방법을 통해 데이터베이스 시스템의 성능을 측정하는데 도움이 될 수 있는 벤치마크 환경을 제공하고자 하는 것이다. 사용자들은 벤치마크 통합도구를 사용하여 범용 벤치마크, 주문 벤치마크는 물론 혼합 벤치마크까지 모든 종류의 벤치마크를 하나의 사용자 인터페이스를 사용하여 수행할 수 있다.

벤치마크 통합도구의 설계단계에서 고려된 주요 사항 중 하나는 확장성이다. 확장 가능한 벤치마크 도구는 사용자들이 어떤 벤치마크라도 쉽게 수행할 수 있는 환경을 제공한다. 벤치마크 통합도구는 사용자들이 다음과 같은 작업을 설계 구조 변경 없이도 수행할 수 있다는 점에서 확장가능하다.

- 사용자들은 벤치마크 통합도구에 어떤 범용 벤치마크라도 추가 가능하다.
- 사용자들은 기존 데이터베이스 시스템에 새로 추가된 특징을 평가하기 위해 기존 범용 벤치마크의 질의를 확장할 수 있다.
- 사용자들은 새로운 주문 벤치마크를 쉽게 벤치마크 통합도구에 추가할 수 있다.

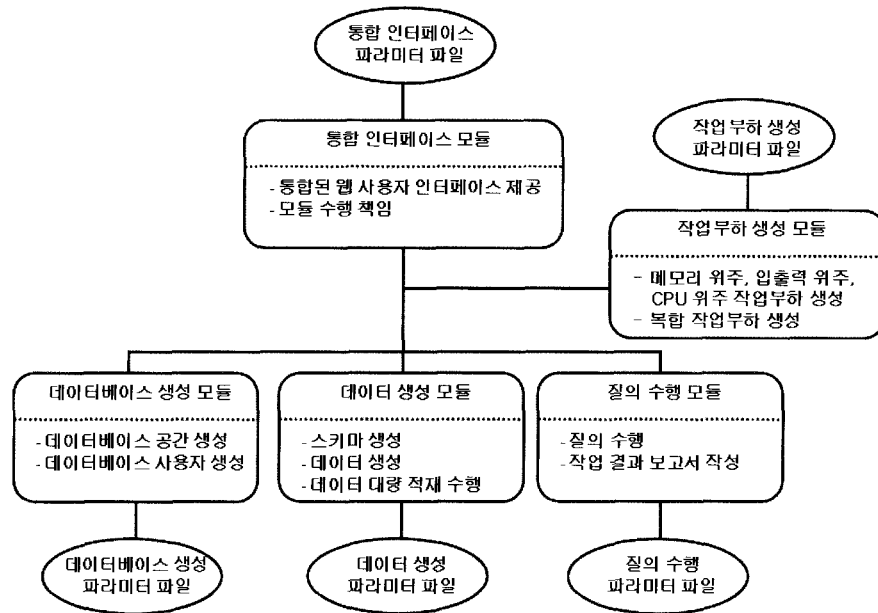
모든 데이터베이스 벤치마크는 그 구성요소를 크게 두 부

분으로 나눌 수 있다. 벤치마크 수행을 위해 필요한 시험 데이터를 생성하는 부분과 시험 질의를 수행하는 부분이다. 우리는 데이터베이스 벤치마크를 구성하고 있는 이 두 부분이 서로 완전히 독립적이어서 별개로 구현될 수 있으며, 또한 단독으로 수행될 수 있음에 주목하였다. 다른 벤치마크의 시험 데이터 생성 부분과 다른 벤치마크의 질의 수행 부분이 동시에 수행될 수 있다면 이를 통해 다양한 벤치마크의 수행이 가능해지고, 혼합 벤치마크 수행도 가능해지기 때문이다. 벤치마크 통합도구는 시험 데이터를 생성하는 부분과 시험 질의를 수행하는 부분이 하나의 운영체제 프로세스로 수행되어 확장성을 가지도록 설계되었다.

벤치마크 통합도구는 데이터베이스 시스템에 의존적이지 않도록 설계되었다. 각 데이터베이스 시스템은 자체의 수행 모델을 가진다. 예를 들어, 로그 관리자는 프로세스 형태로 구현될 수도 있으며, 수행 라이브러리 형태로 구현될 수도 있다. 그러나 데이터베이스 벤치마크 도구는 다양한 데이터베이스 시스템에서 잘 동작되어야 하므로 이러한 특정 수행 모델에도 의존적이지 않아야 한다. 이를 위해 데이터베이스 시스템은 자신의 프로세스에 대해 책임을 지고, 벤치마크 통합도구는 데이터베이스 시스템의 프로세스에 관여하지 않은 채 통합 도구 자신의 프로세스에만 책임을 진다. 예를 들어, 시험 질의로 인해 야기된 에러 처리는 데이터베이스 시스템이 처리한다. 벤치마크 통합도구는 사용자들에게 에러가 발생하였음을 알리고 필요한 경우 벤치마크 수행을 종료하는 작업만을 진행한다.

어떤 벤치마크든지 실세계와 유사한 시험 환경에서 벤치마크를 수행하여 그 결과를 도출하고자 하나 본질적으로 실세계를 시뮬레이션 하는데 제한을 가진다. 실세계와 유사하지 않은 환경에서 수행된 결과는 사용자들이 평소 접하는 환경에서의 데이터베이스 시스템 수행을 대변할 수 없다. 벤치마크 통합도구는 가능한 한 실세계와 유사한 시험 환경을 구축할 수 있도록 실세계의 작업부하와 유사한 인위적인 작업부하를 생성할 수 있도록 설계되었다.

작업 부하 생성은 크게 두 가지 방법으로 이루어진다[2]. 한 가지 방법은 분석 기반 접근 방법이고, 다른 한 가지 방법은 수집 기반 접근 방법이다. 분석 기반 접근 방법은 사용자의 행동이나 특정 작업부하의 특징을 시뮬레이션 하기 위해 수학적 모델을 사용하는 방법이다. 수학적 모델의 예로는 대표적으로 데이터 분포를 들 수 있다. 수집 기반 접근 방법은 작업부하를 생성시키고자 하는 시스템에서의 자원 사용 상태에 대한 통계 자료를 수집하고, 수집된 정보에 기초하여 작업부하를 생성하는 방법이다. 분석 기반 접근 방법은 각 작업부하를 시뮬레이션 하기 위해 해당 작업부하의 특징에 대한 수학적 모델의 선정 및 이에 대한 증명 등 일반 사용자들이 쉽게 수행할 수 없는 작업을 수행해야 한다는 단점을 가진다. 실세계 작업부하는 매우 다양하기 때문에 제한된 수학적 모델을 사용하여 실세계 작업부하 모두를 시뮬레이션하기에는 매우 어렵다. 수집 기반 접근 방법은 분석 기반 접근 방법보다 다양한 작업 부하를 시뮬레이



(그림 1) 벤치마크 통합도구의 모듈과 관련 파라미터 파일

선 할 수 있으며, 보다 단순한 방법을 사용하여 작업부하를 생성할 수 있다.

데이터 생성 작업은 매우 어렵고 시간소비적인 벤치마크 수행 단계이다. 사용자들은 데이터베이스 벤치마크의 보다 정확한 수행을 위해 실세계 데이터를 사용하기를 원하기도 한다. 벤치마크 통합도구는 조합된 데이터뿐만 아니라 그 종류에 있어 다소 제한적이기는 하지만 실세계 데이터를 생성할 수 있도록 지원한다. 벤치마크 통합도구는 실세계 데이터와 유사한 데이터를 생성하기 위해 데이터 분포를 적용하여 데이터를 생성한다. 또한 사용자가 지정한 비율로 널 데이터(null data)를 생성한다. 이는 실세계에서는 대부분의 데이터들이 널 데이터를 가지고 있기 때문에 실세계와 유사한 데이터 생성을 위해 벤치마크 통합도구 설계 시 반영하였다.

벤치마크 통합도구 설계 시 마지막으로 주요하게 고려한 사항은 편리성이다. 사용자는 벤치마크 통합도구를 사용하여 다양한 벤치마크를 수행할 수 있으므로 사용자 인터페이스와 관련하여 통합된 하나의 웹 사용자 인터페이스를 제공할 수 있도록 설계하였다. 각 벤치마크 수행 시마다 해당 벤치마크에 특화된 사용자 인터페이스를 사용한다면 사용자들은 사용하고자 하는 벤치마크마다 사용자 인터페이스를 새로 배워야 한다는 문제점을 가지기 때문이다. 어떤 벤치마크를 수행하더라도 하나의 통일된 사용자 인터페이스를 사용할 수 있도록 편리성을 고려하여 벤치마크 통합도구를 설계하였다.

3. 벤치마크 통합도구

3.1 구성 모듈 및 관련 파라미터 파일

벤치마크 통합도구는 통합 인터페이스 모듈, 데이터베이스

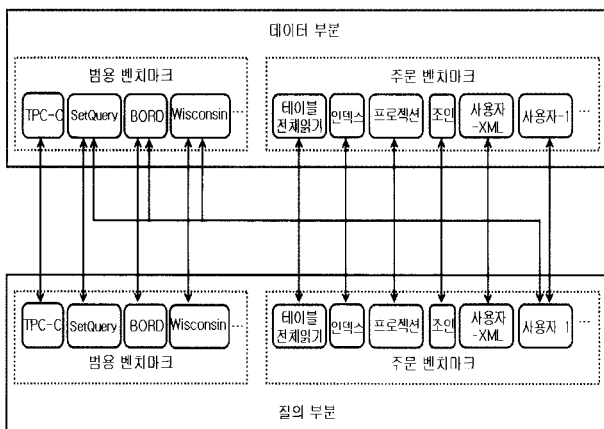
스 생성 모듈, 데이터 생성 모듈, 질의 수행 모듈, 작업부하 생성 모듈로 구성된다. 이들 다섯 모듈들은 상호 독립적이다. 각 모듈들은 각각 하나의 파라미터 파일을 가지며, 하나의 운영체제 프로세스로 동작된다. 앞서 언급한 바와 같이 데이터베이스 벤치마크는 기본적으로 데이터 생성 부분과 질의 수행 부분의 구성 요소를 가진다. 혼합 벤치마크를 쉽게 구현하기 위해서는 이 두 구성 요소가 상호 독립적으로 수행될 수 있도록 구현되어야만 한다. 벤치마크 통합도구는 데이터 생성 모듈과 질의 수행 모듈이 상호 독립적으로 수행될 수 있도록 각각 하나의 운영체제 프로세스로 동작하며 상호 모듈의 수행에 영향을 미치지 않기 때문에 혼합 벤치마크 구현이 쉽다.

(그림 1)은 벤치마크 통합도구의 다섯 모듈과 각 모듈과 관련된 파라미터 파일들을 나타낸다. 각 파라미터 파일은 영역(section), 항목(entry), 값 등 세 개의 구성요소로 이루어진다. 각 파라미터 파일은 미리 정의된 방법에 의해 작성되며, 사용자 인터페이스를 사용하여 구성할 수 있다. 단 사용자가 직접 작성하는 경우에는 미리 정의된 방법에 따라 작성한다.

통합 인터페이스 모듈은 통합된 하나의 사용자 인터페이스를 제공한다. 사용자는 사용자 인터페이스를 사용하여 벤치마크 통합도구에서 제공되는 범용 벤치마크, 주문 벤치마크, 혼합 벤치마크를 수행하거나 새로운 주문 벤치마크의 구현 및 실행이 가능하다. 데이터베이스 생성 모듈은 시험 환경을 구축하기 위한 많은 초기화 작업을 권장한다. 예를 들어, 데이터베이스가 저장되어야 하는 물리적 공간의 확보라든지 데이터베이스 사용자의 생성, 시험 대상 데이터베이스 시스템에 대한 튜닝 파라미터(tuning parameter)의 설정을 수행할 수 있도록 돕는다. 데이터 생성 모듈은 스키마와 텍스트 데이터 및 XML 데이터 생성을 수행한다. 또한 데이

터 생성 모듈은 각 데이터베이스 시스템에서 제공되는 데이터 적재 기능을 활용하여 벤치마크 수행에 필요한 스키마와 데이터를 데이터베이스에 적재한다. 질의 수행 모듈은 사용자가 시험 질의를 정의하거나 수행할 수 있도록 한다. 더불어 사용자가 벤치마크 수행 결과를 볼 수 있도록 보고서를 작성한다. 작업부하 생성 모듈은 수집 기반 접근 방법에 기초하여 실세계와 유사한 작업부하를 생성하는 역할을 담당한다. 작업부하 생성 모듈이 생성할 수 있는 작업부하는 메모리 위주 작업부하, CPU 위주 작업부하, 입출력 위주 작업 부하로 구분되며, 이들이 조합된 형태인 혼합 작업부하가 실세계 작업부하와 유사하다.

벤치마크 통합도구는 네 개의 범용 벤치마크 즉, Wisconsin 벤치마크, Set Query 벤치마크, TPC-C 벤치마크, BORD 벤치마크를 제공한다. 더불어 데이터베이스 시스템의 기본적인 네 가지 기능을 집중적으로 평가할 수 있는 네 개의 주문 벤치마크도 제공한다. (그림 2)는 벤치마크 통합도구에서 제공되는 네 개의 범용 벤치마크와 네 개의 주문 벤치마크 및 이를 활용한 혼합 벤치마크를 보이고 있다.



(그림 2) 벤치마크 통합도구에서 제공되는 벤치마크

(그림 2)에서 보듯이 벤치마크 통합도구는 데이터 부분과 질의 부분을 완전히 분리하여 특정 데이터 생성 부분과 특정 질의 수행 부분이 조합되어 수행될 수 있도록 지원한다. 또한 이러한 독립적 구성은 기본 내부 구조의 변경 없이도 “사용자-1”, “사용자-XML”과 같은 주문 벤치마크의 추가를 용이하게 할 뿐 아니라, 벤치마크 통합도구에서 기본적으로 제공되는 범용 벤치마크와 주문 벤치마크 외에 혼합 벤치마크의 수행도 가능하게 한다. 벤치마크 통합도구에서는 범용 벤치마크의 데이터 생성 부분과 주문 벤치마크의 질의 부분을 조합하여 수행하는 것이 가능하기 때문이다. 벤치마크 통합도구에서 제공되는 범용 벤치마크 외에 사용자가 벤치마크 통합도구의 설계 고려 사항에 따라 구현한 다른 범용 벤치마크가 있는 경우, 구현된 해당 모듈들을 자신이 원하는 디렉터리에 위치시킨 후 해당 모듈들의 위치를 벤치마크 통합도구의 통합 인터페이스 모듈에 등록함으로써 추가가 가능하다.

3.2 데이터 생성

데이터 생성 기능은 데이터베이스 벤치마크에 있어 중요한 기능임을 앞서 설명한바 있다. 벤치마크 통합도구는 데이터 생성 모듈을 통해 데이터 생성 기능을 수행한다. 벤치마크 통합도구는 실세계 데이터와 유사한 데이터를 생성하기 위해 여덟 가지의 데이터 분포를 지원한다. 벤치마크 통합도구에서 지원되는 데이터 분포는 정규 분포, 음지수 분포, 일련번호 분포, 상수 분포, 균일 분포, 난수 분포, 포아송 분포(Poisson distribution), 지프 분포(Zipfian distribution)이다. 이들 분포 중 정규 분포, 음지수 분포, 포아송 분포, 지프 분포는 [6]에서 제시하는 알고리즘을 재구성하여 데이터 생성에 적용한다.

벤치마크 통합도구는 정수형 데이터, 실수형 데이터, 문자형 데이터 등 세 가지 종류의 데이터 종류를 생성할 수 있다. 숫자형 데이터 즉 정수형 데이터와 실수형 데이터에 대해서는 앞서 언급한 여덟 가지 데이터 분포를 모두 지원하며, 문자형 데이터에 대해서는 일련번호 분포, 상수 분포를 제외한 여섯 가지의 데이터 분포를 지원한다. 벤치마크 통합도구는 혼합 기능을 사용하여 생성된 데이터의 순서를 완전히 변경할 수 있다. 때면 혼합 기능이 호출되어 사용될 때마다 생성된 데이터의 순서가 변경되며, 이를 통해 실세계 데이터와 유사한 형태의 데이터를 생성할 수 있다. 즉, 생성되는 데이터의 내용은 동일하지만 그 순서만 달라야 하는 경우 유용하게 사용할 수 있다. 생성되는 데이터의 내용이 달라야 하는 경우에는 서로 다른 데이터 분포를 적용하거나 같은 데이터 분포 내에서 분포에 영향을 미치는 변수 값의 변경을 통해 그 목적을 달성할 수 있다. 벤치마크 통합도구는 이처럼 조합된 데이터뿐만 아니라 실세계의 데이터를 직접 생성할 수 있다. 주민번호, 전화번호, 우편번호에 대해서 실 데이터를 생성한다. 전화번호는 지역번호, 국번, 일련번호로 구성되며, 지역번호 대신 핸드폰 고유 번호를 사용하여 핸드폰 번호도 생성 가능하다. 우편번호의 경우에는 우편번호를 포함하여 우편번호로 식별 가능한 주소까지 생성 가능하다.

문자형 데이터 생성과 관련하여 사용자는 선택율(selectivity)을 지정할 수 있다. 선택율은 질의가 주어졌을 때 추출되어야 하는 데이터의 비율을 의미한다. 예를 들어, 남자가 절반씩 섞여 있는 모임을 시뮬레이션하는 데이터가 있는 경우, 사용자는 남자를 선택하기 위해 선택율의 값을 “50%”로 설정하면 된다. 선택율을 사용자가 지정하게 되면, 사용자가 지정한 문자열 길이만큼의 특정 문자열을 가지는 데이터가 전체 데이터 대비 지정 비율만큼 생성된다. 사용자는 미리 정해진 특정 문자열을 질의에 사용하면 원하는 선택율만큼의 데이터를 추출하게 된다.

일단 데이터가 생성되게 되면, 벤치마크 통합도구에서는 데이터 검증이 가능하다. 데이터 검증 기능은 생성된 데이터의 신뢰도를 높일 수 있는 기능이며, 정확한 벤치마크 결과 획득을 위해 필요한 기능이다. 데이터 검증은 두 단계로 이루어진다. 첫 번째 단계는 생성된 데이터의 수를 점검한

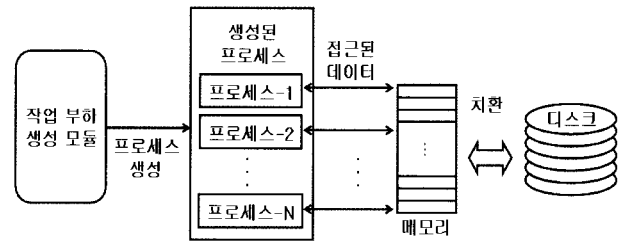
다. 사용자가 지정한 개수만큼의 데이터가 생성되었는지 여부를 점검한다. 두 번째 단계에서는 생성된 데이터가 사용자가 지정한 분포에 맞게 생성되었는지를 검정(test)을 통해 알아본다. 각 데이터 분포에 따라 검정을 수행하는 방법은 [14]에 자세히 기술되어 있으므로 본 논문에서는 생략한다.

벤치마크 통합도구는 XML 데이터를 생성하기 위하여 세 가지 방법을 제공한다. 첫 번째 방법은 텍스트 데이터와 데이터 구조 정의 파일을 이용하여 XML 데이터를 생성하는 방법이다. 텍스트 데이터는 데이터베이스 시스템의 데이터 내보내기(export) 기능을 사용하거나 텍스트 데이터 생성기를 이용하여 생성할 수 있다. 두 번째 방법은 텍스트 데이터와 파라미터 파일을 이용하여 XML 데이터를 생성하는 방법이다. 파라미터 파일은 생성될 데이터의 구조에 대한 간략한 기본 정보를 가지는 파일이다. 세 번째 방법은 사용자가 XML 데이터를 생성하기 위해 텍스트 데이터와 데이터 구조 정의 파일이 없는 경우 사용할 수 있는 방법으로 벤치마크 통합도구의 데이터 생성 모듈에서 제공하는 데이터 자체 생성 기능을 활용하는 방법이다. XML 데이터 생성과 관련된 자세한 사항은 [13]을 참조하기 바란다. 벤치마크 통합도구는 앞서 기술한 텍스트 데이터 생성 기능과 XML 데이터 생성 기능을 모두 수행할 수 있도록 각 기능 별 수행과 더불어 두 기능이 서로 융합하여 수행할 수 있다.

3.3 작업부하 생성

벤치마크 통합도구의 작업부하 생성 모듈은 메모리 위주 작업부하, 입출력 위주 작업부하, CPU 위주 작업부하를 생성할 수 있다. 이들 작업부하의 종류는 독립적으로 생성될 수도 있고, 또한 각 작업부하의 종류가 조합되어 생성될 수도 있다. 작업부하 생성 모듈은 운영체제가 제어하는 자원을 직접 소비함으로써 작업부하를 생성한다. 사용자 프로세스는 운영체제가 제어하는 자원을 정확하게 자신이 필요한 만큼만 소비하는 것이 사실상 불가능하다. 따라서 벤치마크 통합도구는 작업부하 생성 시 허용 오차 한계(tolerance margin)를 두어 사용자가 생성하고자 한 작업부하의 크기와 실제 작업부하 생성 모듈이 생성한 작업부하 크기간의 차이를 일정 정도 허용한다. 허용 오차 한계는 벤치마크 통합도구의 작업부하 생성 모듈이 사용하는 각 작업부하 종류 별 생성 알고리즘에 따라 발생할 수 있는 최대 오차 범위를 의미한다.

(그림 3)은 작업부하 생성 모듈이 메모리 위주 작업부하를 생성하는 과정을 보인다. (그림 3)에서 보이는 바와 같이 메모리 위주 작업부하는 가상 메모리 체계 환경에서 운영체제가 실제 메모리와 스왑 메모리(swap memory) 사이에 데이터를 치환하여 생성한다. 이를 위해 작업부하 생성 모듈은 먼저 실제 메모리에서 자신의 작업 공간을 확보하기 위해 서로 경쟁하는 많은 프로세스를 생성한다. 생성된 프로세스들이 접근하고자 하는 전체 데이터의 크기는 실제 메모리 크기의 1.5배다. 전체 데이터의 크기는 많은 실험을 거쳐 경험적으로 결정된 값이다. 전체 데이터의 크기를 결정



(그림 3) 메모리 위주 작업부하 생성 과정

하는 방법은 [12]에 기술되어 있다. 생성되는 프로세스들의 수는 작업부하의 크기에 따라 유동적으로 조정되며, 각 프로세스들은 메모리 위주 작업부하의 생성을 위한 작업 중 작업부하 크기의 조절을 위해 휴지 시간(pause time)을 가질 수 있다.

사용자는 시스템 명령어인 “top”이나 “iostat”를 통해 확인 가능한 “iowait” 값을 사용하여 메모리 위주 작업부하의 크기를 나타낸다. 작업부하 생성 모듈은 사용자가 지정한 작업부하 크기에 최대한 가깝게 작업부하를 생성하기 위하여 프로세스의 수와 각 프로세스마다의 휴지 시간을 조정한다. 사용자가 작업부하 크기를 지정하면 작업부하 생성 모듈은 최대 5%의 허용 오차 한계에서 작업부하를 생성할 수 있다. 입출력 위주 작업부하, CPU 위주 작업부하, 혼합 작업부하의 생성 방법이나 자세한 기술은 [12]를 참조하기 바란다.

4. 벤치마크 통합도구를 사용한 주문 벤치마크 수행 예

본 장에서는 벤치마크 통합도구에 새로운 주문 벤치마크 추가의 용이함과 작업부하 생성 기능을 활용하여 실세계와 유사한 시험 환경에서 벤치마크를 어떻게 수행할 수 있는지를 사례를 통해 살펴본다. 벤치마크 통합도구로의 추가를 위해 사용되는 주문 벤치마크는 “사용자 관점에서의 조인 연산 평가 방법론”[15]이다. 사용자들은 새로운 주문 벤치마크를 벤치마크 통합도구에 추가하기 위하여 사용자 인터페이스를 사용하거나 직접 작성하여 다섯 개의 파라미터 파일을 생성해야 한다.

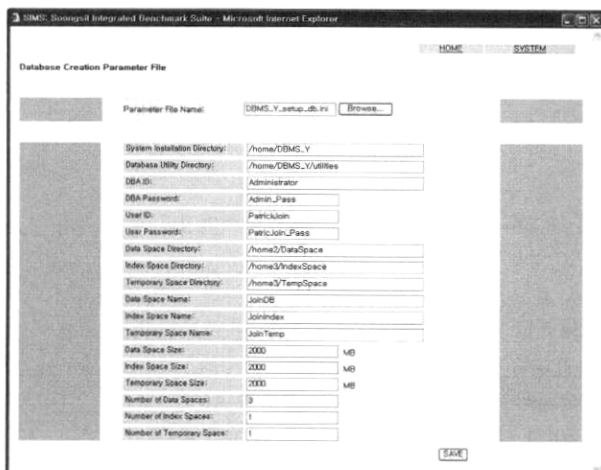
(그림 4)는 통합 인터페이스 파라미터 파일을 작성하는 사용자 인터페이스를 보이고 있다. “DBMS_Types” 항목은 벤치마크의 대상이 되는 데이터베이스 시스템을 지정하는데 사용된다. 대상 데이터베이스 시스템의 이름은 실험의 편의상 “DBMS_Y”로 설정하였다. “Benchmark_Types” 항목은 수행하고자 하는 벤치마크의 종류를 의미하며, 주문 벤치마크를 수행하고자 하는 경우에는 “User”값을 선택한다. 범용 벤치마크를 수행하고자 하는 경우에는 수행하고자 하는 범용 벤치마크의 이름을 선택하고, 혼합 벤치마크를 수행하고자 하는 경우에는 주문 벤치마크와 마찬가지로 “User”값을 선택하면 된다. (그림 4)에 설정된 값들의 의미는 각 시험



(그림 4) 통합 인터페이스 파라미터 파일의 작성 예

질의 수행 사이에 데이터 버퍼를 시험 질의 결과와 무관한 데이터로 채우는 버퍼 플러싱(buffer flushing)을 사용하지 않고, 시험 질의를 10번 반복 수행하며, 고립화 수준으로 “read committed”를 사용함을 의미한다. 마지막으로 기술된 “Execution Stages”는 사용자가 수행하고자 하는 작업의 진행 범위를 결정한다. 각 작업 진행 범위는 벤치마크 통합도구의 각 구성 모듈과 연관된다. 데이터 생성 모듈이 데이터 생성 및 데이터 적재의 두 작업 범위를 가진다는 점을 제외하면 모든 모듈이 각각 하나의 작업 범위를 가진다.

(그림 5)는 데이터베이스 생성 파라미터 파일의 작성 예를 보인다. “System Installation Directory” 항목과 “Database Utility Directory” 항목은 벤치마크 대상이 되는 데이터베이스 시스템의 설치 경로와 각 데이터베이스 시스템이 보유하고 있는 유틸리티 프로그램들의 위치 정보를 의미한다. 각 경로는 루트 디렉터리(root directory)로부터 완전한 경로(full path)를 작성하여야 한다. 사용자는 데이터베이스 내에서 데이터가 저장될 영역, 인덱스가 저장될 영역, 임시 작업을 위해 사용할 영역에 대한 정보를 (그림 5)에 기술된 각 항목들에 적절한 값을 기술함으로써 지정할 수 있다. 사용자가 지정한 데이터 영역, 인덱스 영역, 임시 영역에 대한 각 영역의 크기는 기술된 각 영역의 단위 크기와 각 영역에



(그림 5) 데이터베이스 생성 파라미터 파일의 작성 예

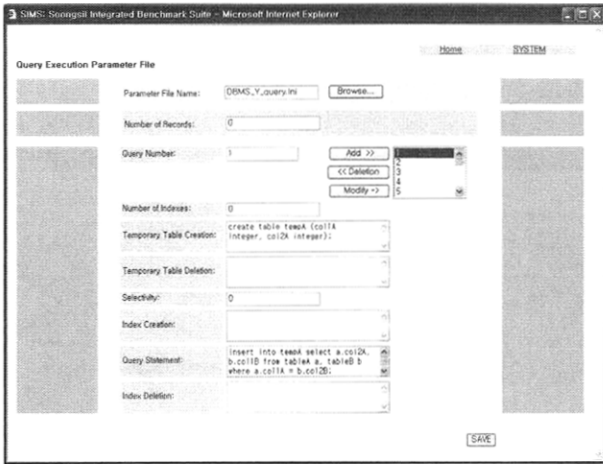
위치할 파일의 개수를 곱하여 얻어진다. 예를 들어, 데이터 영역의 전체 크기는 “Data Space Size” 항목의 값과 “Number of Data Spaces” 항목의 값을 곱하여 구할 수 있다.



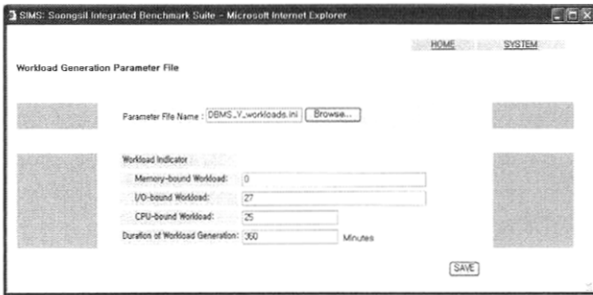
(그림 6) 데이터 생성 파라미터 파일의 작성 예

(그림 6)은 데이터 생성 파라미터 파일의 작성 예를 보이고 있다. “Tables” 영역에 기술되는 각 항목들은 각 테이블들에 대한 정보를 의미하며, “Columns” 영역은 지정된 특정 테이블에서의 각 칼럼에 대한 정보를 의미한다. 현재 (그림 6)에서 보이는 “Columns” 영역은 “Tables” 영역에 기술된 세 개의 테이블(“tableA”, “tableB”, “tableC”) 중에서 “tableB” 테이블의 세 개 칼럼(“col1B”, “col2B”, “chpadB”) 중 “col2B” 칼럼에 대한 상세 내용을 보이고 있다. “Tables” 영역의 “Number of Records” 항목은 “tableB” 테이블의 레코드 수가 208,000개로 지정되었음을 의미한다. “Columns” 영역에 기술된 각 항목들은 “col2B” 칼럼이 균일 분포를 가지는 정수형 데이터 형이며, 널 값을 가지지 않고 서플링을 수행하지 않음을 보인다.

(그림 7)은 질의 수행 파라미터 파일의 작성 예이며, 현재 “질의 1”에 대한 내용을 보이고 있다. 질의 수행 파라미터 파일의 작성을 통해 사용자는 벤치마크의 시험 질의를 수행하기 필요한 인덱스와 임시 테이블을 생성할 수 있다. “Temporary Table Creation” 항목에서 임시 테이블 생성을 위한 SQL문을 기술하고, 인덱스는 “Index Creation” 항목에 인덱스 생성을 위한 적절한 SQL문을 기술함으로써 생성할 수 있다. 생성된 임시 테이블이나 인덱스의 삭제는 각각 “Temporary Table Deletion” 항목과 “Index Deletion” 항목에 삭제와 관련된 적절한 SQL문을 기술함으로써 가능하다. 시험 질의 수행을 위한 SQL문은 “Query Statement” 항목에 기술한다. “Number of Records” 항목과 “Selectivity” 항목에 값을 지정함에 의해 사용자는 수행된 질의에 대한 결과 검증을 수행할 수 있다. “Number of Records” 항목은 질의 수행 결과 튜플 수이며 “Selectivity” 항목은 질의 대상 전체 튜플 수 중 선택될 튜플의 비율을 의미한다. “Number of Records” 항목과 “Selectivity” 항목이 “0”으로 지정되는 경우에는 데이터 검증 절차를 수행하지 않는다.

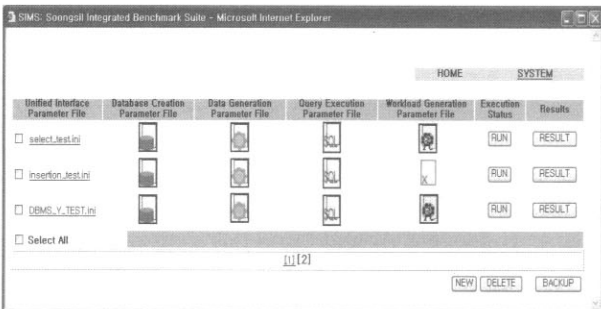


(그림 7) 질의 수행 파라미터 파일의 작성 예



(그림 8) 작업부하 생성 파라미터 파일의 작성 예

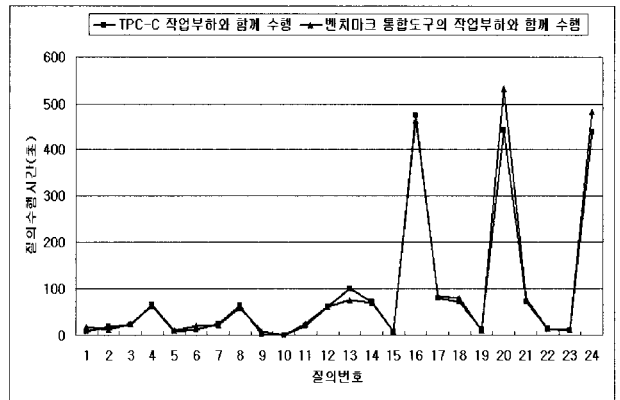
(그림 8)은 작업부하 생성 파라미터 파일의 작성 예를 보인다. “Workload Indicator” 영역은 “Memory-bound Workload” 항목, “I/O-bound Workload” 항목, “CPU-bound Workload” 항목을 가진다. 각 항목들은 세 가지 작업부하의 크기를 지정하는 항목이다. (그림 8)에 기술된 값들이 의미하는 바는 메모리 위주 작업부하는 생성하지 않으며, 입출력 위주 작업부하의 크기는 27, CPU 위주 작업부하의 크기는 25를 가지는 복합 작업부하를 생성한다는 것이다. 작업부하를 생성하는 총 수행 시간은 “Duration of Workload Generation” 항목에 기술하며, 360분을 수행하도록 지정하였다. 작업부하를 생성하기 위해 작업부하 생성 모듈이 수행된 후 360분 후에는 작업부하 생성 모듈이 작업부하 생성을 중단한다.



(그림 9) 벤치마크 통합도구의 주화면

사용자 인터페이스를 사용하여 다섯 개의 파라미터 파일의 작성이 완료된 후, 사용자는 (그림 9)의 “RUN” 버튼을 클릭함에 의해 주문 벤치마크를 수행할 수 있다. (그림 9)는 벤치마크 통합도구의 주 화면을 보이고 있다. 주 화면은 이미 작성된 파라미터 파일에 대한 연결(link)을 유지하며 각 파일에 대한 작성과 수정을 주 화면을 거쳐 수행할 수 있도록 환경을 제공한다. (그림 4)~(그림 8)의 파라미터 파일들에 대해 (그림 9)의 맨 마지막 줄의 아이콘(icon)들이 각각 그 연결을 유지하고 있다. (그림 9)의 두 번째 줄에서 “Workload Generation Parameter File” 항목에 대한 아이콘의 모양은 다른 아이콘과 다르다. 이는 해당 주문 벤치마크에 대한 작업부하 생성 파라미터 파일이 작성되지 않았음을 의미한다.

벤치마크 통합도구에 새로 추가된 주문 벤치마크를 수행하기 위해 본 논문에서는 Sun사의 E3500 서버에서 특정 회사의 상용 데이터베이스 시스템을 사용하였다. 데이터베이스 시스템의 종류 및 버전은 데이터베이스 시스템 설치 시 동의한 라이선스(license)로 인해 밝히지 못한다. 벤치마크 통합도구는 데이터베이스 생성 모듈의 수행과 함께 벤치마크 수행을 시작한다. 사용자가 지정한 내용대로 데이터를 생성하고 이를 데이터베이스에 적재한 후, 작업부하의 생성이 시작된다. 작업부하가 사용자가 지정된 크기에 이르게 되면 시험의 질의 수행 모듈이 시작하게 된다. 작업부하가 일정크기에 이르게 되면 작업부하 생성 모듈은 미리 지정된 파일의 특정 위치에 정해진 문자를 쓰게 되고, 이를 통합 인터페이스 모듈이 체크하여 질의 수행 모듈을 시작한다. 사용자가 지정한 질의가 수행된 이후에는 수행 결과를 사용자에게 보고하기 위한 파일을 정해진 위치에 작성하게 된다.

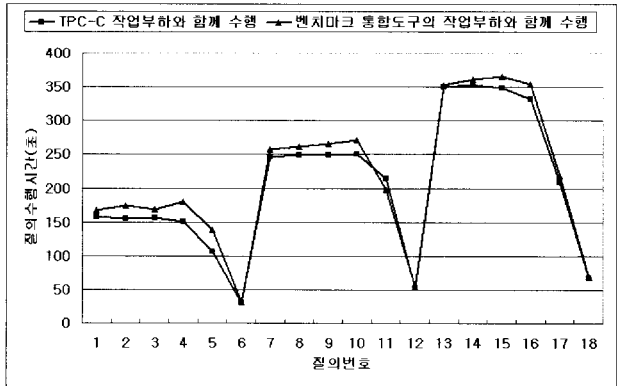


(그림 10) 새로이 추가된 주문 벤치마크의 수행 결과

(그림 10)은 작성된 다섯 개 파라미터 파일들을 참고하여 새롭게 벤치마크 통합 도구에 추가된 주문 벤치마크를 수행한 결과와 TPC-C 벤치마크의 작업부하 그리고 새로운 주문 벤치마크를 동시에 수행한 결과를 함께 보이고 있다. (그림 10)의 작성 목적은 사용자가 벤치마크 통합도구를 통해 실제 작업 환경과 유사한 환경에서 벤치마크를 수행할 수 있음을 보이기 위함이다.

TPC-C 벤치마크는 여러 데이터베이스 시스템 개발 업체와 하드웨어 개발 업체가 모여서 구성한 비영리 단체인 TPC에서 다년간의 논의를 거쳐 개발되었다. TPC-C 벤치마크는 온라인 트랜잭션 처리 환경을 시뮬레이션하고, 이를 통해 데이터베이스 시스템이나 서버 시스템의 성능 측정을 목적으로 한다. 즉, 실세계의 온라인 트랜잭션 처리 환경을 대표적으로 표현할 수 있는 창고 관리 업무를 시뮬레이션하고 이를 통해 온라인 트랜잭션 처리 환경에서의 각 시스템의 성능을 측정하고자 한 것이다. TPC-C 벤치마크는 산업계에서 표준(de facto)과 같이 인정받고 있으며 다른 시스템들과의 비교 평가를 위해 현재도 널리 사용되고 있기 때문에 TPC-C 벤치마크로부터 얻어진 작업부하를 실세계 환경에서의 작업부하로 가정하였다.

(그림 8)에 기술된 복합 작업부하의 크기는 TPC-C 벤치마크의 작업부하 크기를 미리 측정된 값이며, 벤치마크 통합도구는 사용자가 지정한 크기의 작업부하를 일정크기의 허용 오차 한계 내에서 생성할 수 있으므로 (그림 10)의 결과 도출을 위해 사용된 두 작업부하의 크기는 동일하다 할 수 있다. (그림 10)의 두 그래프는 비슷한 모양을 보이고 있다. 이 사실은 벤치마크 통합도구에 새로운 주문벤치마크의 적용이 원활히 수행되어 실세계 작업 환경과 유사한 환경에서 그 수행 결과를 도출하고 있음을 의미한다.



(그림 11) 또 다른 주문 벤치마크의 수행 결과

(그림 11)은 (그림 10)의 결과를 보이기 위한 수행 환경 중 작업부하 생성 파라미터 파일의 내용은 동일하게 하고, 나머지 파라미터 파일들의 내용을 달리하여 벤치마크 통합도구를 수행한 결과를 보이고 있다. 이 결과 또한 (그림 10)과 마찬가지로 두 그래프가 비슷한 모양을 보이고 있어, 벤치마크 통합도구가 실세계 작업 환경과 유사한 환경에서 수행 결과를 도출하고 있음을 알 수 있다.

5. 결 론

본 논문에서는 데이터베이스 시스템을 위한 벤치마크 통합도구에 대해 기술하였다. 벤치마크 통합도구는 지금까지 따로 수행되던 범용 벤치마크, 주문 벤치마크, 혼합 벤치마크

를 사용자가 하나의 통합된 사용자 인터페이스를 이용하여 쉽게 수행할 수 있는 환경을 제공한다. 또한 사용자가 범용벤치마크나 주문벤치마크를 새로 개발한 경우 이를 쉽게 추가할 수 있는 확장된 구조를 제공한다. 현재 벤치마크 통합도구에서는 네 개의 범용 벤치마크와 네 개의 주문 벤치마크를 기본적으로 제공하고 있으며, 이를 이용하여 혼합 벤치마크도 수행할 수 있다. 벤치마크 통합도구는 사용자들에게 실세계의 작업 환경에 보다 가까운 벤치마크 수행 환경을 제공하기 위해 작업부하 생성 기능을 제공한다.

벤치마크 통합도구의 개발은 국내 데이터베이스 시스템 개발사의 요구사항 충족을 위해 시작되었으며, 여러 해에 걸쳐 새로운 요구사항 수용 및 시행착오를 거쳐 현재에 이르고 있다. 처음 개발된 통합도구[8]에서는 TPC-C 벤치마크, BORD 벤치마크, Wisconsin 벤치마크 등 3종류의 범용 벤치마크와 4종류의 주문 벤치마크가 제공되었으며, 사용자 인터페이스가 사용자 대화방식의 텍스트 형태였다. 본 논문에서 기술하고 있는 통합도구에서는 범용 벤치마크인 Set Query 벤치마크를 추가하였고, 사용자 인터페이스를 웹 기반으로 변경하였다. 또한 데이터 생성 시 대용량 데이터 생성을 위한 알고리즘 추가와 XML 데이터 생성 기능, 작업부하 생성 기능이 추가되었다. 특히 작업부하 생성 기능은 데이터베이스 시스템 업계에서의 요구사항을 반영하기 위해 꼭 필요한 사항이었다.

벤치마크 통합도구는 작업부하 생성 기능을 통해 다중 사용자 환경을 시뮬레이션하고 있다. 그러나 벤치마크 통합도구는 단일 사용자용으로 설계되었기 때문에 현재 단일 사용자 환경으로만 벤치마크 수행이 가능하다. 단일 사용자 환경에서의 벤치마크도 나름대로의 의미를 가지지만, 데이터베이스 시스템 및 하드웨어의 복잡화, 사용자 수의 급속한 증가, 매우 복잡한 업무 프로세스가 다중 사용자 환경에서의 데이터베이스 벤치마크를 요구하고 있는 게 현실이다. 벤치마크 통합도구의 질의 수행 모듈 개선 및 확장을 통해 다중 사용자 환경에서 쉽게 수행될 수 있는 벤치마크 통합도구의 개발을 추후 수행할 것이다. 또한 다중사용자 환경 외에 XML 데이터베이스 시스템 벤치마크 등 다양한 벤치마크 환경에서의 실험도 추후 수행할 것이다.

현재 기업들에서는 실시간 환경에서의 업무 처리와 빠른 의사 결정을 위해 노력하고 있다. 이런 환경에서는 디스크 기반 데이터베이스보다는 주기억장치 기반 데이터베이스가 보다 효과적이다. 현재 개발되어진 벤치마크 통합도구는 디스크 기반의 데이터베이스 벤치마크를 손쉽게 수행할 수 있는 환경을 제공하고 있다. 이를 주기억장치 기반 데이터베이스의 벤치마크를 위해 적용할 수 있도록 개선하고 확장하는 것이 추후 연구 과제가 될 것이다.

참 고 문 헌

[1] M. Asgarian, M.J. Carey, D.J. DeWitt, J. Gehrke, J.F. Naughton, and D.N. Shah, "The BUCKY Object-Relational

Benchmark”, Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, pp.135-146, 1997.

[2] P. Barford and M. Crovella, “Generating Representative Web Workloads for Network and Server Performance Evaluation”, Proceedings of the 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp.151-160, 1998.

[3] M.J. Carey, D.J. DeWitt, and J.F. Naughton, “The OO7 Benchmark, Proceedings of the 1993 ACM SIGMOD Conference on Management of Data”, pp.12-21, 1993.

[4] R.G.G Cattel and K. Skeen, “Object Operations Benchmark”, ACM Transactions on Database Systems, Vol.17, No.3, pp.1-31, 1992.

[5] D.J. DeWitt, “The Wisconsin Benchmark: Past, Present, and Future”, In: The Benchmark Handbook for Database and Transaction Processing Systems 2nd Ed., J. Gray Ed., Morgan Kaufmann, pp.269-316, 1993.

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

[7] U. Hohenstein, V. Plessner, and R. Heller, “Evaluating the Performance of Object-Oriented Database Systems by Means of a Concrete Application”, Proceedings of the 8th Database and Expert Systems Applications Workshop, pp.496-501, 1997.

[8] 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.

[9] S.H. Lee, S.J. Kim, and W. Kim, “The BORD Benchmark for Object-Relational Database”, Proceedings of the 11th International Conference on Database and Expert Systems Applications, pp.6-20, 2000.

[10] P.E. O’Neil, “The Set Query Benchmark”, In: The Benchmark Handbook for Database and Transaction Processing Systems 2nd Ed., J. Gray Ed., Morgan Kaufmann, pp.359-395, 1993.

[11] The TPC home page, <http://www.tpc.org/>.

[12] 김기욱, 정희진, 이상호, “데이터베이스 시스템을 위한 실세계 부하 생성 도구”, 정보처리학회논문지D, 제11-D권, 제7호, pp.1427-2434, 2004.

[13] 신선미, 정희진, 이상호, “WeXGene: 웹 기반 XML 데이터 생성기”, 정보처리학회논문지D, 제12-D권, 제2호, pp.199- 210, 2005.

[14] 옥은택, 정희진, 이상호, “데이터베이스 벤치마크를 위한 데이터 생성기와 성능 평가”, 한국정보처리학회 논문지, 제10-D권, 제6호, pp.907-916, 2003.

[15] 정희진, 이상호, “사용자 관점에서의 조인 연산 평가 방법론”, 정보처리학회논문지D, 제12-D권, 제1호, pp.13-20, 2005.

정희진



e-mail : sinclear@dreamwiz.com

1993년 우석대학교 전산학과(학사)

1995년 숭실대학교 대학원 컴퓨터학과 (석사)

2005년 숭실대학교 대학원 컴퓨터학과 (박사)

1995년~2000년 (주)헨디소프트 기술연구소, 선임연구원

2002년~2003년 숭실대학교 정보미디어 연구소, 전임연구원

2005년~현재 (주)헨디소프트 기술전략실, 수석연구위원

관심분야 : 데이터베이스 시스템 성능 평가 및 튜닝, 비즈니스 프로세스 관리

이상호



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

1984년 서울대학교 전산공학과(학사)

1986년 미국 노스웨스턴대 전산학과 (석사)

1989년 미국 노스웨스턴대 전산학과 (박사)

1990년~1992년 한국전자통신연구원, 전임연구원

1999년~2000년 미국 조지메이슨대, 소프트웨어정보공학과, 교환 교수

1992년~현재 숭실대학교 컴퓨터학부 교수

관심분야 : 인터넷 데이터베이스, 데이터베이스 시스템 성능 평가 및 튜닝