韓國컴퓨터情報學會 論文誌
第15卷 第8號, 2010. 8.

2010-15-8-1-1

DBMS와 결합된 데이터스트림관리시스템을 위한 성능 평가 도구 개발

김 경 배*

Development of the Performance Benchmark Tool for Data Stream Management Systems Combined with DBMS

Kim, Gyoung-Bae*

요 약

데이터스트림 관리시스템(DSMS)의 많은 응용분야에서는 단순한 실시간 스트림 데이터의 효율적인 처리뿐만 아니라 기존의 DBMS와 결합하여 데이터마이닝이나 데이터웨어하우징 같은 고급 서비스를 사용자에게 제공하는 것을 요구하고 있다. 본 논문에서는 고급 서비스를 위하여 DSMS와 DBMS를 결합한 시스템의 성능평가를 위한 도구를 개발하였다. 기존 연구 개발된 대표적인 DSMS와 DBMS를 결합하여 네트워크 모니터링 스트림 데이터를 기반으로 통합된 시스템의 성능평가를 수행하였다. 통합된 시스템의 평가를 위하여 JAVA로 통합 시스템 성능 평가 툴을 개발하였으며, 개발 된 툴을 이용하여 DSMS(STREAM, Coral8)와 DBMS(MySQL, Oracle10g)를 결합한 시스템의 성능평가를 수행하였다.

Abstract

Many applications of DSMS(Data Stream Management System) require not only to process real-time stream data efficiently but also to provide high quality services such as data mining and data warehouse combining with DBMS(Database Management System) to users. In this paper we execute the performance benchmark of the combined system of DSMS and DBMS that is developed for high quality services. We use the stream data of network monitoring application system and combine the traditional representative DSMSs and DBMSs in a single system for the performance testing. We develop the total performance benchmark tool implementing JAVA language for the our testing. For our performance testing, we combine DSMS such as STREAM and Coral8 and DBMS such MySQL and Oracle10g respectively.

▸ Keyword : 데이터스트림 관리시스템(DSMS), 성능평가도구(performance benchmark tool), 스트림데이터(stream data)

• 제1저자 : 김경배
• 투고일 : 2010. 04. 21, 심사일 : 2010. 06. 08, 게재확정일 : 2010. 07. 02.
* 서원대학교 컴퓨터교육과 부교수

Ⅰ. 서 론

고속의 네트워크 기술과 각종 센서기술의 발달로 인해 증권정보, 네트워크 상태 관리, 웹 패킷 감시, 그리고 물류 관리 등의 응용 분야에서 네트워크를 통해 지속적으로 발생되는 대용량의 스트림 형태 데이터에 대한 실시간 처리 요구가 증가하고 있다[1]. 이와 같이 지속적으로 입력되는 대량의 데이터 스트림을 처리하기 위한 시스템을 데이터 스트림 관리 시스템(Data Stream Management System: DSMS)[2]이라고 한다.

기존의 데이터베이스 관리시스템(DataBase Management System: DBMS)은 다양한 비즈니스 응용의 요구사항들을 성공적으로 지원하기 위해 효율적인 업무용 데이터 처리를 목적으로 설계되었다. 따라서 기존의 데이터베이스 시스템에서는 대용량의 데이터 집합들이 데이터베이스에 저장되고, 사용자들은 디스크에 안전하게 저장된 데이터에 대한 질의와 트랜잭션을 요구한다. 그러나 기존 DBMS 용용과는 달리 DSMS 응용에서 발생되는 데이터는 무한하고(unbounded) 연속적으로(continuous) 발생하는 특징을 지니고 있으며, 응용에서 요구되는 질의 또한 데이터 스트림에 대한 질의도 순간 질의보다는 연속 질의(continuous query)[3]의 처리를 요구한다. 연속 질의는 사용자에 의해 입력된 후 지속적으로 실행되는 질의로 이러한 연속질의의 예로는 네트워크 모니터링 시스템에서 네트워크의 평균트래픽양이나 최근 1시간 동안 가장 높은 접속이 발생하는 서버를 검색하는 질의 등을 들 수 있다.

최근 DSMS에 대한 연구가 활발히 진행되어 그 결과로 STREAM[4], TelegraphCQ[5], Aurora[6], NiagaraCQ[7], Nile[8]과 같이 대학에서 개발된 프로토타입 형태의 시스템과 연속적인 이벤트 처리(Continuous Event Processing: CEP)의 장점을 내세우는 Coral8[9], StreamBase[10]와 같은 상용 시스템들이 개발되었다. 연구개발 된 초기의 DSMS는 새로운 응용 분야에서의 적용을 위해 Nile 기반으로 GIS 기술과 결합한 PLACE[11]와 TelegraphCQ를 기반으로 네트워크 모니터링에 적용한 Data Triage[12]와 같은 시스템으로 확장 개발되었다. 그러나 실제 DSMS의 많은 응용분야에서는 단순한 실시간 스트림 데이터의 효율적인 처리뿐만 아니라 기존의 DBMS와 결합하여 유용한 정보를 추출하여 데이터베이스에 저장하고 데이터마이닝이나 데이터 웨어하우징

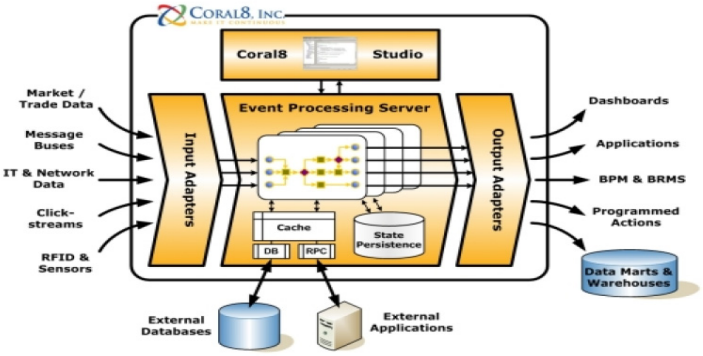

그림 1. Coral8의 시스템 구조
Fig. 1. Architecture of Coreal8

같은 고급의 처리를 요구하고 있다[12, 13].

따라서, 본 논문에서는 DSMS와 DBMS를 고급의 서비스 개발을 위하여 결합한 시스템에서의 성능평가를 수행하였다. 구현 알고리즘과 스트림 데이터의 조건이나 특성에 따라서 성능의 차이가 발생하는 특성으로 인해 일반적으로 DSMS의 성능 평가에는 많은 어려움이 있다[14]. 기존에 수행된 DSMS에 대한 성능평가 방법은 도시고속도로와 톨게이트 시스템을 시뮬레이션 한 Linear Road[15]와 온라인 전자상거래를 시뮬레이션 한 NEXMark[16]가 있으나, 이는 특정 응용 환경을 시뮬레이션 하여 DSMS 중심의 성능평가 방법만을 제공하였다. 따라서 본 논문에서는 현재 연구 개발된 DSMS와 DBMS를 단일 시스템에서 결합하여 네트워크 모니터링 스트림 데이터를 기반으로 통합 시스템의 성능평가를 수행하였다. 통합된 시스템의 평가를 위하여 JAVA 기반의 통합 시스템 성능 평가 툴을 개발하였으며, 성능평가에 사용된 시스템으로

는 DSMS로는 대표적인 프로토타입 시스템인 STREAM과 상용 DSMS인 Coral8을 사용하였고, DBMS로는 MySQL[17]과 Oracle10g[18]를 결합하여 성능평가를 수행하였다. 성능평가를 위해 사용된 데이터는 [19]에서 적용된 네트워크 모니터링 시스템을 위한 스트림 데이터 스키마를 적용하였다.

본 논문의 구성은 다음과 같다. 2장에서 관련연구로 기존에 개발된 DSMS에 대한 시스템 소개와 성능평가 방법에 대하여 설명하고, 3장에서는 성능평가를 위해 사용되는 데이터에 대하여 설명하고, 4장에서는 DSMS와 DBMS를 결합하여 성능평가 하기 위해 개발된 통합 성능 평가 시스템의 구현과 기능에 대하여 설명한다. 5장에서 성능평가의 결과와 이에 대한 분석을 수행하고, 6장에서 본 논문의 결론 향후 연구 방향에 대하여 논한다.

II. 관련 연구

1. DSMS(Data Stream Management System)

다양한 센서들로 부터의 스트림 데이터에 대한 연속적인 질의 처리를 위한 데이터 스트림 처리 시스템에 대한 연구가 활발히 진행되었다. 대표적인 DSMS로는 스탠포드대학의 STREAM[4], 버클리대학의 TelegraphCQ[5], 위스콘신매디슨대학의 NiagaraCQ[7], 그리고 브라운대학과 M.I.T의 Aurora[6] 등이 있다. STREAM은 센서 스트림의 연속적인 질의를 처리하기 위해 SQL을 확장한 CQL(Continuous Query Language)를 정의하고, 관계형 연산자들을 연속적인 질의에 맞게 확장하였다. TelegraphCQ 는 관계형 연산자의 시맨틱(semantic)을 스트림으로 확장하는 StreaQuel을 제공하며, 다량의 스트림을 동시에 처리할 때 발생할 수 있는 성능 저하를 개선하기 위해 자원 공유 기반의 질의 처리 방법을 제안하였다. NiagaraCQ는 분산 환경에서 XML을 이용하여 센서 스트림의 연속 질의를 처리하기 위한 방법을 제공한다. Aurora는 데이터 스트림의 흐름(flow)을 직접 지정할 수 있도록 하여 센서 스트림의 전송 속도가 가변적일 때 실시간적인 연속 질의를 처리하기 위한 메모리 연산 스케줄링 방법을 제공한다.

대표적인 상용시스템으로 Coral8[9]과 StreamBase[10]가 있다. Coral8은 그림 1과 같은 구조를 가지고 있으며, 복합적인 이벤트 처리 응용(Complex Event Processing: CEP)을 위해 개발되었다. Coral8은 스트림 데이터를 위한 프로그래밍 모델과 고성능, 고 확장성의 신뢰성 있는 플랫폼을 제공하고, 통합된 프레임워크와 툴을 제공한다.

StreamBase는 네트워크 보안관리 및 투자 관리의 실시간 분석 요구사항이 증가하는 환경에서 비즈니스 이벤트의 실시간 처리를 위해 대용량 데이터의 실시간 처리에 적합하도록 DSMS를 상용화한 새로운 데이터 관리 플랫폼이다.

StreamBase는 필요한 데이터만을 선별적으로 저장하여 실시간 처리 성능을 지원하고, 가능한 컴포넌트수를 최대한 줄여 프로세스의 수를 감소시켰으며, Failover의 고가용성을 유지하고, 프로그램을 위한 다양한 툴을 제공한다.

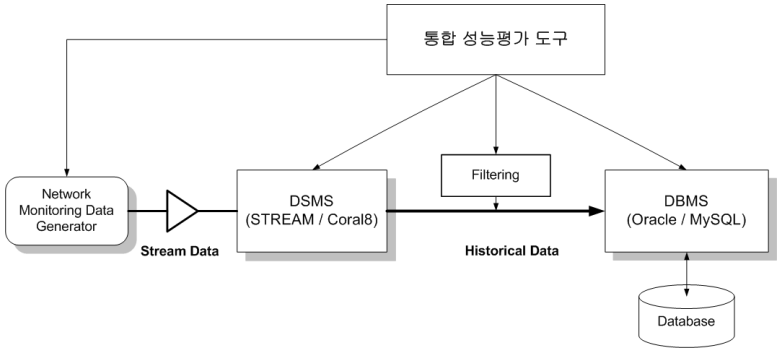

그림 2. 통합 성능평가 시스템의 데이터 처리 및 흐름
Fig 2. Data processing and flow of total performance testing system

2. 성능평가 방법

기존의 데이터베이스 관리시스템의 성능평가 방법으로는 TPC(transaction Processing Performance Council)[20]을 많이 사용하고 있다. TPC에서 TPC-C는 신뢰도 높은 온라인 트랜잭션 프로세싱(OLTP) 작업의 성능과 확장성 측정 벤치마크(산업 표준)로 널리 쓰이고 있다. TPC-C 테스트는

약 0.6GB 크기의 데이터베이스 스키마상에서 트랜잭션을 수행 하여 초당 처리 가능한 트랜잭션(TPS)의 수를 측정한다. TPC 성능평가 방법에는 의사결정 지원테스트를 위한 TPC-H, 웹에서의 성능평가를 위한 TPC-W등이 있다.

Linear Road는 스트림 데이터 관리시스템을 위해 개발된 성능평가 프로그램으로 DSMS와 관계형 데이터베이스 시스템("System X") 등에 대하여 스트림 데이터 관리의 특성을 성능평가 할 수 있도록 개발되었다. Linear Road는 대도시의 고속도로에서 이동 차량에 대한 톨게이트 시스템을 시뮬레이션 하였으며, 현재의 교통 상황과 이력 데이터 통계관리를 지원하는 교통 모니터링 시스템을 통하여 Aurora와 STREAM의 성능평가에 사용되었다. [21]에서는 Linear Road 를 기반으로 스트림 처리를 위한 미들웨어인 스트림 프로세싱 코어(stream processing core: SPC)를 개발하여 대규모의 분산 스트림 프로세싱 시스템의 특징과 성능의 병목 현상에 대한 성능평가를 수행하였다. NEXMark(Niagara Extension to XMark)는 연속데이터스트림에 대한 질의를 평가하기 위해 온라인 전자상거래의 모델을 구현하여 개발되었으며, 현재 version 0.8이 소개되고 있으며, 웹에서의 대용량 XML 데이터를 위해 개발된 XMark를 DSMS를 위한 연속질의 등에 대한 성능평가를 할 수 있도록 확장하였다.

III. 성능평가 데이터

본 논문의 성능평가에서 사용하는 데이터는 Frederick Reiss[19]에서 네트워크 모니터링 시스템의 성능평가에 사용된 데이터 기반으로 하였다. 표 1은 각 스트림 데이터의 타입과 의미를 설명하고 있다. DSMS에 입력되는 스트림 데이터는 네크워크 패킷에 포함되어 있는 9개의 속성(attribute)과 DSMS에서 도달된 시간을 나타내는 1개의 타입스탬프(timestamp) 속성으로 스트림 레코드를 구성한다.

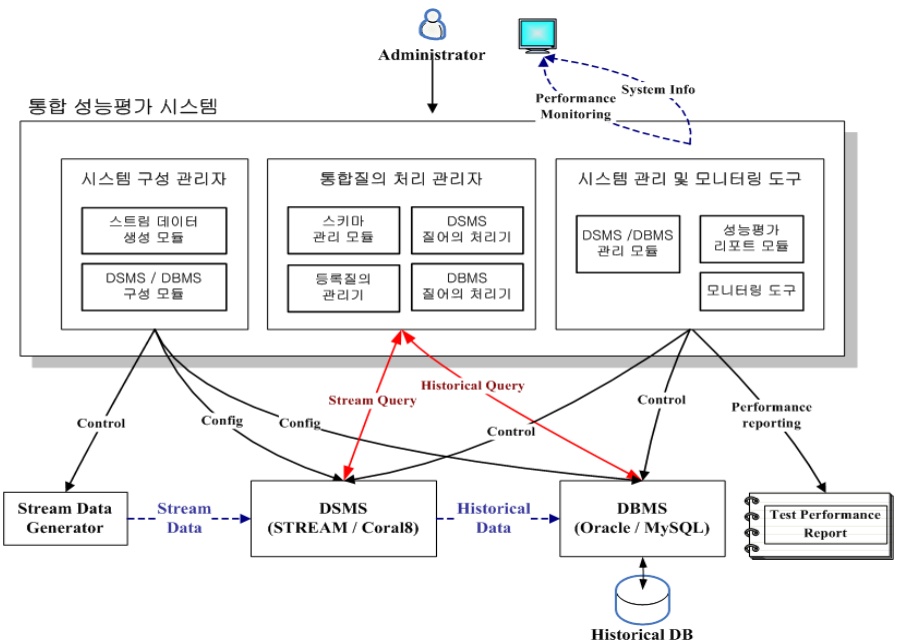

그림 3. 통합 성능평가 시스템의 구조
Fig. 3. System Architecture of Performance Benchmark Tool

표 1. 성능평가에 사용된 스트림 데이터의 스키마
Table 1. Schema of stream data using performance evaluation

Field Name	Field Type	Field Description
timeStamp	Integer (TimeStmap)	스트림 데이터의 타임스탬프
protocol	Integer	프로토콜 타입 0:TCP, 1:UDP
source_Ip	String	패킷의 소스 IP 주소
destination_Ip	String	패킷의 목적지 IP 주소
source_Port	Integer	소스 포트 번호(1-65535)
destination_Port	Integer	목적 포트 번소(1-655535)
bytes_Send	Integer	전송데이터의 크기
bytes_received	Integer	수신데이터의 크기
connection	Integer	연결표시 0:비연결, 1: 연결
duration	Integer (TimeStamp)	접속 시간

성능 평가에서 사용하는 스트림 데이터인 net 입력 스트림의 스키마를 위해서 STREAM에서는 입력 스트림을 위해 register stream 키워드를 이용하여 스트림을 선언하고, 상업용인 Coral8은 CREATE구문에서 스트림 입력을 위한 INPUT STREAM, 스트림 출력을 OUT STREAM 을 각각 구분해서 선언할 수 있다.

그림 2는 통합 성능평가 시스템에서의 성능평가를 위해서 발생되는 데이터와 발생된 데이터를 처리하는 DSMS와 DBMS 간의 데이터 이동과 처리 과정을 보여 주고 있다. 본 성능평가에서는 연구용 프로토타입으로 개발된 대표적인 DSMS인 Stanford 대학의 STREAM과 상업용 DSMS인 Coral8이 사용되며, 다양한 사용자의 응용서비스 및 고급 서비스를 지원하기 위한 DBMS로는 Oracle과 MySQL를 적용하였다.

데이터 생성기에 의해서 발생되는 스트림 형태의 네트워크 모니터링 데이터는 DSMS에서 입력되어 통합 성능 평가 시스템에 DSMS에 요구되는 CQL과 같은 스트림 질의를 처리한다. DSMS에 도달된 스트림 데이터 중에서 데이터베이스 시스템에 저장되어 향후 데이터웨어하우징이나 마이닝과 같은 고급의 서비스를 위해 사용될 데이터는 DSMS의 필터링 기능을 거쳐서 DBMS에 전달되며 DBMS에서는 이들 데이터를 데이터베이스에 저장되어 응용에서 요구되는 SQL과 같은 데이터베이스 질의를 처리한다.

Ⅳ. 통합 성능평가 시스템의 구현

DSMS와 DBMS의 통합 성능평가 시스템의 구현은 RedHat Enterprise 5에서 개발되었으며, 3.4GB의 메모리 크기와 500GB의 7200rpm의 STAT 디스크로 구성되어 있다. 통합 성능평가 시스템 통합 사용자 GUI 및 시스템의 개발에 사용된 언어는 JAVA이며, version 1.6으로 구현하였다.

구분	환경
CPU	Intel(R) Pentium(R) Dual E2180 @ 2.0GHz
메모리 크기	3.4GB
운영체제	RedHat Enterprise 5
디스크 크기	500GB
디스크 타입 및 종류	SATA Barracuda (7200rpm)

성능평가에 사용된 DSMS와 DBMS의 버전은 표 2와 같다. DSMS로는 STREAM 0.6 버전과 Coral8 5.4.1이 사용되었고, DBMS로는 Oracle 10g Enterprise release 10.2.0.1.0과 MySQL 5.5.51a가 통합 성능평가에 사용되었다.

표 2. 성능평가에 사용된 DSMS와 DBMS
Table 2. DSMS and DBMS using performance evaluation

시스템명	유 형	사용시스템
STREAM	DSMS	Stream-0.6.0
Coral8	DSMS	5.4.1 rhel4-x86.release
Oracle	Relational DBMS	10g enterprise release 10.2.0.1.0
MySQL	Relational DBMS	mysql-5.0.51a

통합 성능평가 시스템은 그림 3과 같이 데이터생성기와 DSMS 및 DBMS의 구성을 관리하기 위한 시스템구성관리자, DB 스키마 관리 및 질의어를 처리하는 통합 질의 처리 관리자, 그리고 시스템의 관리 기능과 성능평가 모니터링 기능을 제공하는 시스템 관리 및 모니터링 도구로 구성된다.

구성관리자는 네트워크 모니터링을 위한 스트림 데이터를 생성하는 생성기를 제어하는 스트림 데이터 생성 모듈과 성능평가를 위해 수행할 DSMS와 DBMS를 선택할 수 있는 기능을 제공한다. DSMS로는 STREAM과 Coral8을 선택할 수 있고, DBMS로는 Oracle10g와 MySQL을 조합하여 선택할 수 있다. 각각의 DSMS와 DBMS를 선택한 후에 실행을 하면 자동적으로 선택된 DSMS와 DBMS에 적절한 환경설정과 함께 각각의 프로세서가 실행되며 성능평가를 수행할 준비를 완료한다. 그림 4는 통합 성능평가 시스템의 실행화면을 보여주고 있다.

통합 질의처리 관리자는 성능평가를 위해 수행되는 질의를 입력 받아서 관리하고 이를 DSMS와 DBMS에서 실행할 수 있는 기능을 지원한다. 통합 성능평가 시스템과 DSMS와 DBMS의 연결 방식은 JDBC를 이용하여 질의 실행과 결과의 전달이 처리된다. 통합 질의처리 관리자에 의해 실행되는 질의는 DSMS를 위한 CQL과 같은 스트림 데이터를 위한 질의와 Oracle이나 MySQL의 디스크기반 데이터베이스에 저장되어 있는 데이터를 위한 이력 질의로 구분된다.

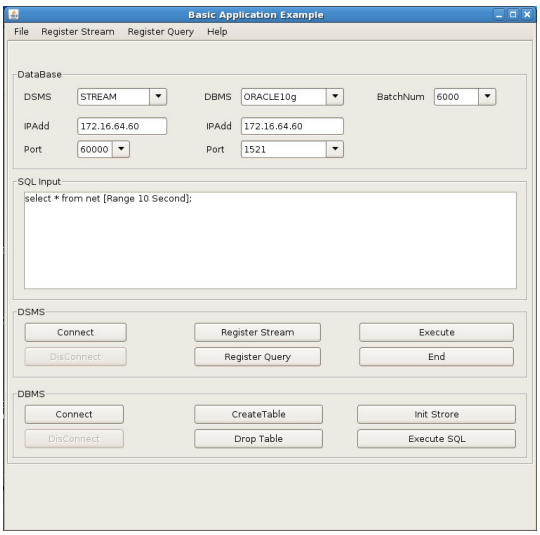

그림 4. 통합 성능평가 시스템의 실행 화면
Fig. 4. Execution screen of total performance testing system

시스템 관리 및 모니터링 도구는 DBMS와 DBMS의 관리자에게 제공되는 데이터베이스의 생성 및 삭제, DBMS/DSMS의 시작 및 종료 등과 같은 관리기능과 성능측정결과에 대한 모니터링 기능을 제공한다. 실행되는 성능 측정의 결과는 모니터를 통해서 관리자에게 제공되며, 그 결과 및 데이터는 파일형태로 디스크에 저장된다.

V. 성능평가 결과

성능 평가는 시스템의 사양이나 구현 방법에 영향을 받는다. 따라서 본 논문에서는 성능평가를 수행하는 시스템 환경에서의 최대 성능을 알아보기 위한 DSMS와 DBMS를 단일 성능 평가를 수행하였고, 이를 기반으로 구현된 통합 성능평가 툴에서 DSMS와 DBMS의 결합 시스템의 성능 평가를 수행하였다.

1. 단일 시스템의 성능 평가 결과

1.1 DBMS의 성능 평가

통합 성능 평가 시스템에서 DSMS에서 처리된 데이터를 이력질의 등의 처리를 위해서 DBMS에 전달되어 저장된다. 본 성능평가에서 사용되는 시스템에서 JDBC를 이용하여 한 번에 5,000개의 튜플을 배치 삽입(batch insert)한 경우에 그림 5와 같은 성능을 보였다.

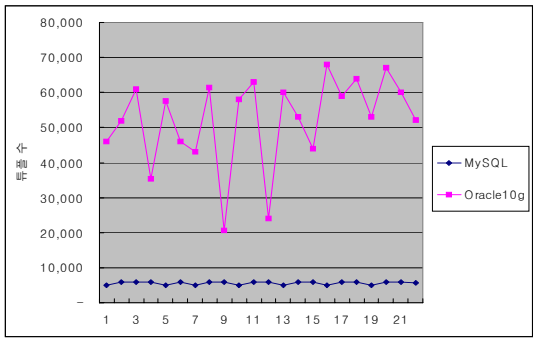

그림 5. MySQL과 Oracle의 배치삽입 성능 비교(bach 크기= 5,000)
Fig. 5 Bach insertion performance of MySQL and Oracle

Oracle 평균 52,190개의 튜플을 삽입할 수 있고, MySQL은 5,667개의 튜플을 삽입할 수 있는 것으로 측정되었다. Oracle과 MySQL이 약 9.2배의 큰 성능 차이를 보이는 것은 JDBC를 이용하여 DBMS에 대한 배치 삽입을 수행하는 경우에 Oracle이 효율적인 배치 삽입을 지원하는 반면, MySQL은 효율인 배치 삽입을 지원하지 않기 때문에 발생한 것으로 판단된다. 동일 환경에서 배치 삽입을 사용하지 않고 1개씩 튜플을 삽입하는 경우에는 유사한 성능을 보였다.

1.2 STREAM과 Coral8에서 제공되는 도구를 이용한 성능 평가

STREAM에서 제공되는 성능 평가 도구인 STREAM Visualizer를 이용하여 테스트를 수행하면 그림 6과 같이 최대 초당 180,000개의 스트림 데이터가 처리됨을 볼 수 있다. 그러나 STREAM에서의 최대 성능은 질의 처리를 위한 연산에서 발생되는 positive 스트림 데이터와 negative 스트림 데이터를 모두 포함한 스트림 데이터의 개수이다. Coral8에서 지원되는 도구인 Coral8 Studio를 이용하여 Windows XP 환경에서 수행한 성능평가에서는 그림 7과 같이 최대 초당 188,336개의 스트림이 처리됨을 볼 수 있다.

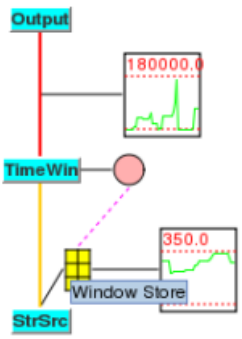

그림 6. STREAM Visualizer를 이용한 성능 평가 결과
Fig. 6. Performance result using STREAM Visualizer

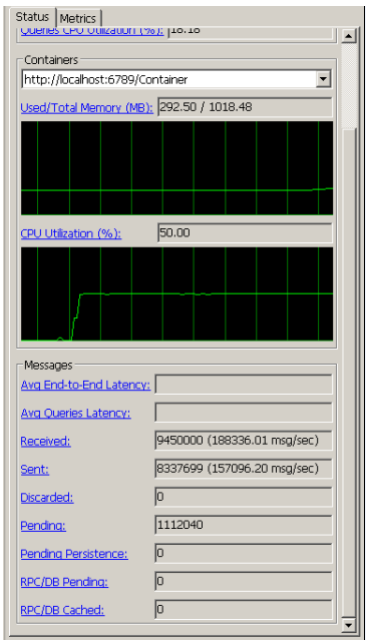

그림 7. Coral8 Studio를 이용한 성능 평가
Fig. 7. Performance Testing using Coral8 Studio

1.3 통합 성능 평가 시스템에서 STREAM과 Coral8의 성능 평가 결과

그림 8은 본 논문의 통합 성능평가 시스템에서 수행한 STREAM과 Coral8의 filter 연산 수행 시 최대 스트림 데이터의 처리 개수를 구한 것이다. STREAM의 경우 평균 27,583개가 처리 되고, Coral8의 경우에는 36,070개의 스트림 데이터가 처리되었다. 초기에 연구용으로 개발된 STREAM보다 상업용 시스템인 Coral8이 약 30%정도 우수한 성능을 보였다.

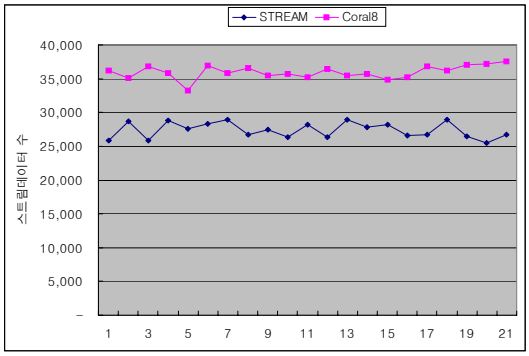

그림 8. STREAM 과 Coral8의 Filter 연산 시 최대 스트림 처리 수
Fig. 8. Number of Maximum stream in Filter operation of STREAM and Coral8

2. DSMS와 DBMS의 결합 시스템의 성능 평가 결과 및 분석

다음은 본 논문의 통합 성능평가 툴을 이용하여 DSMS와 DBMS가 결합된 시스템의 성능평가 결과이다. 각 성능평가는 스트림 데이터 생성기에서 발생되어 DSMS에 입력되는 데이터를 DSMS에서 처리하여 50%와 25%의 스트림 데이터를 DBMS에 저장하여 성능을 평가 하였다. 본 성능평가에서 사용되는 배치 삽입의 크기는 성능평가를 통해서 최적의 값을 구하였으며, MySQL은 5,000개에서 Oracle은 9,000개에서 가장 효율적이었다.

2.1 STREAM 과 MySQL

그림 9와 그림 10은 STREAM과 MySQL을 결합하여 수행한 성능 평가의 결과이다. 생성된 스트림 데이터의 50%를 저장하는 경우 평균적으로 20,079개의 생성된 스트림 데이터에서 STREAM은 50%인 10,533개의 데이터가 처리되었으나, MySQL에서는 3,600개의 스트림 데이터만이 처리되어 약 34%의 데이터 밖에 저장하지 못했다. 25%의 경우에는 평균적으로 25,800개의 스트림 데이터가 생성되었고 STREAM에서는 약 25%인 6,722개의 스트림 데이터가 처리되었고 MySQL에서는 4,200개의 스트림 데이터가 처리되어 약 62%가 처리되었다.

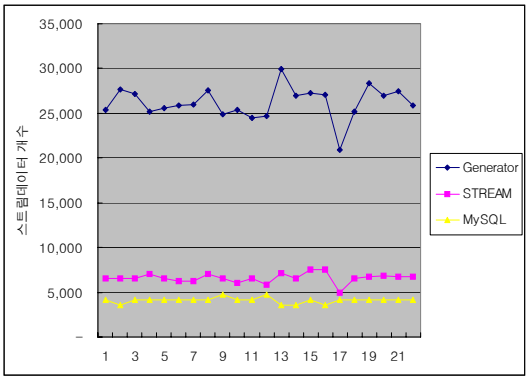

그림 9. 25%의 스트림 데이터를 MySQL에 저장하는 경우의 성능
Fig. 9. Performance of MySQL storing 25% stream data

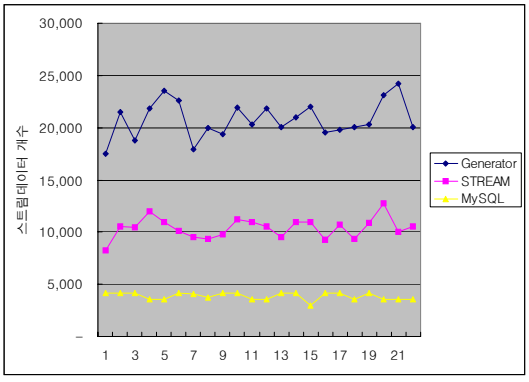

그림 10. 50%의 스트림 데이터를 MySQL에 저장하는 경우의 성능
Fig. 10. Performance of MySQL storing 50% stream data

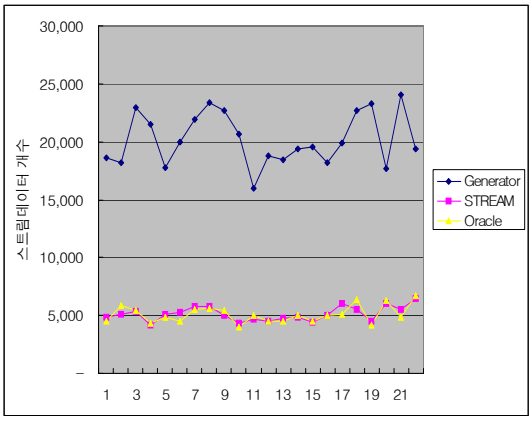

그림 11. 25%의 스트림 데이터를 Oracle에 저장하는 경우의 성능
Fig. 11. Performance of Oracle storing 50% stream data

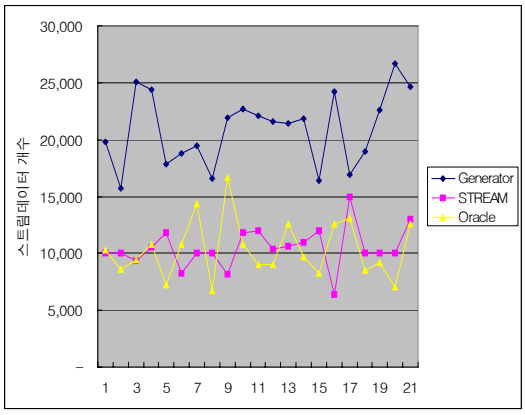

그림 12. 50%의 스트림 데이터를 Oracle에 저장하는 경우의 성능
Fig. 12. Performance of Oracle storing 25% stream data

2.2 STREAM과 Oracle

STREAM과 Oracle을 결합한 시스템의 성능은 그림 11과 그림 12에서 볼 수 있다. 50%인 경우에는 평균적으로 20,939개의 스트림 데이터가 생성이 되어 STREAM에서 10,488개의 스트림 데이터가 처리되고, Oracle에서 10,348개의 스트림 데이터가 처리되어 STREAM에서 처리된 99%의 데이터가 Oracle에서 처리됨을 볼 수 있다.

그림 12의 25%인 경우에는 평균적으로 20,216개의 스트림 데이터가 생성이 되어 STREAM에서 5,129개의 스트림 데이터가 처리되고, Oracle에서 5,091개의 스트림 데이터가 처리되어 99%의 처리율을 보였다.

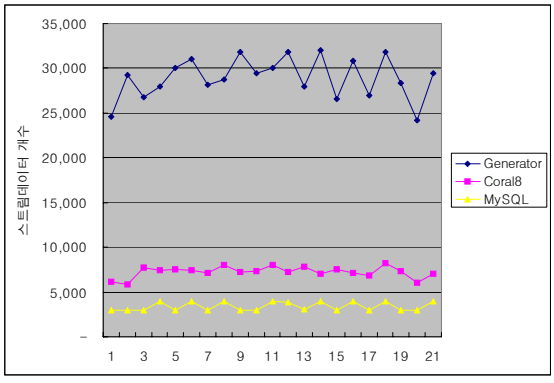

그림 13. 25%의 스트림 데이터를 MySQL에 저장하는 경우의 성능 (배치 삽입 = 5,000튜플)
Fig. 13. Performance of MySQL storing 25% stream data(batch insertion = 5,000tuples)

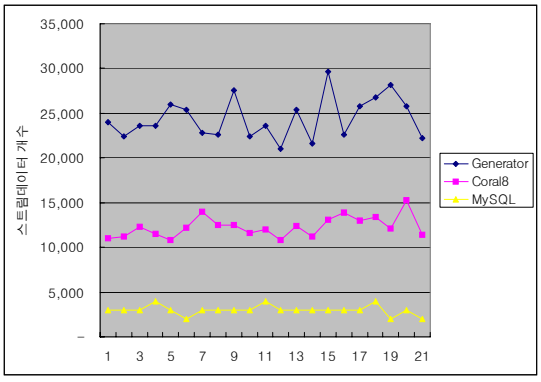

그림 14. 50%의 스트림 데이터를 MySQL에 저장하는 경우의 성능 (배치 삽입 = 5,000튜플)
Fig. 14. Performance of MySQL storing 50% stream data(batch insertion = 5,000tuples)

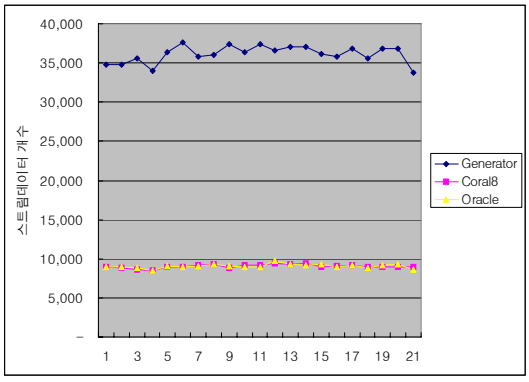

그림 16. 25%의 스트림 데이터를 Oracle에 저장하는 경우의 성능(배치 삽입 = 5,000튜플)
Fig. 16. Performance of Oracle storing 50% stream data(batch insertion = 5,000tuples)

2.3 Coral8과 MySQL

Coral8과 MySQL을 결합한 시스템의 성능 평가는 그림 13과 그림 14와 같다. 50%의 경우에는 그림 14와 같이 평균적으로 24,429개의 스트림 데이터가 생성이 되어 Coral8에서 12,291개의 스트림 데이터가 처리되었으나, MySQL에서는 3,000개의 스트림 데이터만이 처리되었다. Coral8에서 요구된 스트림 데이터의 약 25% 만이 MySQL을 통해서 저장되었다. 25%의 경우에는 그림 13과 같이 평균적으로 28,943개의 스트림 데이터가 생성이 되어 Coral8에서 7,219개의 스트림 데이터가 처리되었으나, MySQL에서는 3,428개의 스트림 데이터만이 처리되어 Coral8에서 요구한 스트림 데이터의 48%만이 처리되었다.

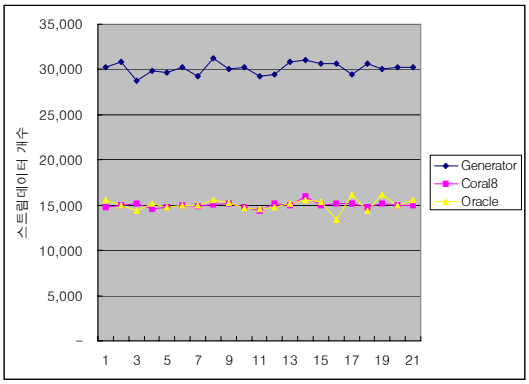

그림 15. 50%의 스트림 데이터를 Oracle에 저장하는 경우의 성능(배치 삽입 = 5,000튜플)
Fig. 15. Performance of Oracle storing 25% stream data(batch insertion = 5,000tuples)

2.4 Coral8과 Oracle

Coral8과 Oracle을 결합한 경우에는 50%에서는 평균적으로 30,095개의 스트림 데이터가 생성이 되어 Coral8에서 15,019개의 스트림 데이터가 처리되었고, Oracle에서도 거의 모든 데이터인 15,083개의 스트림 데이터가 그림 15와 같이 처리되었다. 25%의 경우에는 평균적으로 35,124개의 스트림 데이터가 생성이 되어 Coral8에서 9,048개의 스트림 데이터가 처리되었고, Oracle에서도 9,066개의 스트림 데이터가 그림 16과 같이 처리되었다.

본 실험 성능 평가의 결과는 다음과 같이 분석할 수 있다. 첫째, 통합 시스템의 전체 성능은 스트림 데이터의 처리율에 50%인 경우에는 약 23,886개의 스트림을 처리할 수 있으며, 25%의 처리율인 경우에는 27,521개의 스트림 데이터를 처리할 수 있어 통합 시스템의 처리율이 15% 증가 되었다. 둘째, DSMS의 성능은 50%인 경우에 STREAM이 10,511개의 스트림 데이터를 처리하고 Coral8인 13,655개의 스트림 데이터를 처리하여 Coral8이 30% 우수하였다. 25%인 경우에는 STREAM과 Coral8이 각각 5,926개와 8,134개의 스트림 데이터를 처리하여 37% 더 처리하였다. 셋째, DBMS의 경우에는 Oracle의 경우에는 DSMS에서 요구되는 대부분의 스트림 데이터에 대한 처리를 만족하였으나, MySQL의 경우에는 50%인 경우에는 DSMS의 요구의 27%정도가 처리되었고 25%의 경우는 54%정도만이 처리가 되었다.

Ⅵ. 결 론

고속의 네트워크 기술과 각종 센서기술의 발달로 등장한 DSMS는 대용량의 스트림 형태 데이터에 대한 효율적인 실시간 처리를 지원하다. 그러나 DSMS의 많은 응용분야에서는 단순한 실시간 스트림 데이터의 효율적인 처리뿐만 아니라 기존의 DBMS와 결합하여 유용한 정보를 추출하여 데이터베이스에 저장하고 데이터마이닝이나 데이터 웨어하우징 같은 고급의 처리를 요구하고 있다.

본 논문에서는 DSMS와 DBMS를 고급의 서비스 개발을 위하여 결합한 시스템에서의 성능평가를 수행하였다. 기존 연구 개발된 대표적인 DSMS와 DBMS를 단일 시스템에서 결합하여 네트워크 모니터링 스트림 데이터를 기반으로 통합 시스템의 성능평가를 수행하였다. 통합된 시스템의 평가를 위하여 JAVA 기반의 통합 시스템 성능 평가 툴을 개발하였으며, 성능평가에 사용된 시스템으로는 DSMS로는 STREAM과 Coral8을 사용하였고, DBMS로는 MySQL과 Oracle10g를 각각 결합하여 성능평가를 수행하였다.

성능평가 결과 DSMS와 DBMS의 통합 시스템의 전체 성능은 DSMS에 처리되어 DBMS에 저장되는 스트림 데이터의 처리율을 50%에서 25%의 축소하는 경우 통합 시스템의 처리율이 15%가 증가 되었다. 통합 시스템에서 DSMS의 성능은 상업용인 Coral8이 STREAM보다 약 30% 정도 우수함을 보였고, DBMS의 성능은 Oracle이 실험에서 수행한 DSMS의 요구를 만족하였으나, MySQL의 경우에는 50%인 경우에는 DSMS의 요구의 27%정도가 처리되었고 25%의 경우는 54% 정도 만 처리되었다.

참고문헌

〔1〕 Golab, L. and Ozsu, M. T., "Issues in Data Stream Management," ACM SIGMOD Record, Vol. 32, No. 2, pp. 5-14, 2003.

〔2〕 Babcock, B. et al., "Models and Issues in Data Stream Systems," PODS2002, pp. 1-16, 2002.

〔3〕 Motwani, R. et al., "Query Processing, Approximation, and Resource Management in a Data Stream Management System," In Proc. the First Biennial Conf. on Innovative Data Systems Research, Asiloma, California, pp. 245-256, Jan. 2003.

〔4〕 A. Arasu, B. Babcock, S. Babu, J. Cieslewicz, M. Datar, K. Ito, R. Motwani, U. Srivastava and J. Widom, "Stream: The Stanford Data Stream Management System," in IEEE Data Engineering Bulletin, vol. 4(1), pp.19-26, 2003.

〔5〕 S. Chandrasekaran, O. Copper, A. Deshpande, M. J. Franklin, H. Joseph M, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss and M. Shah, "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World," in Proceedings of the Conference on Innovative Data Systems Research, pp.11-18, 2003.

〔6〕 Zdonik, S. et al., "The Aurora and Medusa Projects," IEEE Data Engineering Bulletin, Vol. 26, No. 1, pp. 3-10, 2003.

〔7〕 Jianjun Chen, David J. DeWitt, Feng Tian and Yuan Wang, "NiagaraCQ: A calable Continuous Query Stream for Internet Databases," ACM SIGMOD Record, pp.379-390, 2000.

〔8〕 M. A. Hammad, M. F. Mokbel, M. H. Ali, W. G. Aref, A. C. Catlin, A. K. Elmagarmid, M. Eltabakh, M. G. Elfeky, T. M. Ghanem, R. Gwadera, I. F. Ilyas, M. Marzouk, and X. Xiong. "Nile: A Query Processing Engine for Data Streams", In ICDE, p.851, 2004.

〔9〕 *http://www.coral8.com*

〔10〕 *http://www/streambase.com*

〔11〕 Mohamed F. Mokbel , Xiaopeng Xiong, Moustafa A. Hammad , Walid G. Aref, "Continuous Query Processing of Spatio-Temporal Data Streams in PLACE," Geoinformatica, v.9 n.4, p.343-365, 2005.

〔12〕 F. Reiss and J. Hellerstein. "Data Triage: An Adaptive Architecture for Load Shedding in TelegraphCQ," In IEEE ICDE Conference, pp.155-156, April 2005.

〔13〕 이충호, 안경환, 이문수, 김주완, "u-GIS 공간정보 기술동향," 전자통신동향분석, 제 22권, 제 3호, 110-123쪽, 2007년.

〔14〕 Coral8, "Guide to Evaluating Complex Event Processing Engines," *http://www.coral8.com/system/files/assets/pdf/GuideToEvaluatingCEPEngines.pdf*

〔15〕 A. Arasu, M. Cherniack, E. Galvez, D. Maier, A. Maskey, E. Ryvkina, M. Stonebraker, and R. Tibbetts, "Linear Road: A Stream Data Management Benchmark," VLDB 2004, pp.480-491, 2004.
〔16〕 *http://datalab.cs.pdx.edu/niagara/NEXMark/*
〔17〕 *http://www.mysql.org*
〔18〕 *http://www.oracle.com*
〔19〕 Frederick Reiss, Kurt Stockinger, Kesheng Wu, Arie Shoshani, Joseph M. Hellerstein, "Enabling Real-Time Querying of Live and Historical Stream Data," Proceedings of the 19th International Conference on Scientific and Statistical Database Management, p.28, July 09-11, 2007.
〔20〕 *http://www.tpc.org*
〔21〕 Navendu Jain, Lisa Amini, Henrique Andrade, Richard King, Yoonho Park, Philippe Selo, Chitra Venkatramani, "Design, implementation, and evaluation of the linear road benchmark on the stream processing core," Proceedings of the ACM SIGMOD Conference, pp.431-442, 2006.

저 자 소 개

김 경 배

1992: 인하대학교 전자계산공학과 학사
1994: 인하대학교 전자계산공학과 석사
2000: 인하대학교 전자계산공학과 박사
2000-2004: 한국전자통신연구원 선임연구원
2004-현재: 서원대학교 컴퓨터교육과 교수
관심 분야: 데이터베이스, GIS, 모바일컴퓨팅, 공항IT, 컴퓨터교육론