

# .NET 기반의 Remoting 시스템과 Socket 통신을 이용한 이 기종 DB 동기화

## Synchronization of Heterogeneous Database Using .NET-Based Remoting System and Socket Communication

- ▶ 투고자 : \*조호목 (육군 제2군사령부 정보체계지원실)  
          김길래 (육군 제2군사령부 정보체계지원실)
- ▶ 원고 매수 : 표지(1), 본문 및 참고문헌(7), 저자약력(1)
- ▶ 그림의 수 : 15 개

# .NET 기반의 Remoting 시스템과 Socket 통신을 이용한 이 기종 DB 동기화

조호목\*, 김길래

## Synchronization of Heterogeneous Database Using .NET-Based Remoting System and Socket Communication

Ho-Mook, Cho\*, Kil-Lae, Kim

### 요 약

본 논문에서는 .NET 기반의 Remoting 시스템과 Socket 통신을 이용하여 이 기종 DB Server 간의 Data 동기화를 구현하였다. 이 시스템은 Client의 웹 브라우저를 이용 Socket 통신으로 Data를 전송하여 데이터베이스에 저장하는 방식으로 구현하였고, Data 읽기는 Remoting 기술인 중계 어플리케이션 서버의 클래스를 원격으로 실행시켜 Data를 가져오는 방식으로 구현하였다. DB 동기화는 Data를 저장할 때 중계 어플리케이션 서버와 DB 서버의 Data 관리 어플리케이션 간 비동기 통신으로 DB 서버의 Data 처리 수준을 실시간으로 감시하여 중계 어플리케이션 서버에서 선택적으로 Data를 전송하고, 이후 Data 처리 수준이 정상화가 되면 다시 Data를 전송케 함으로써 이 기종 DB 서버 간의 동기화를 구현하였다.

본 논문은 1장에 서론, 2장에 .NET Remoting 기술과 네트워크 프로그래밍, 3장에 이 기종 DB 서버 간 동기화 시스템 설계 및 구현, 4장에 결론을 기술하였다.

## 1. 서 론

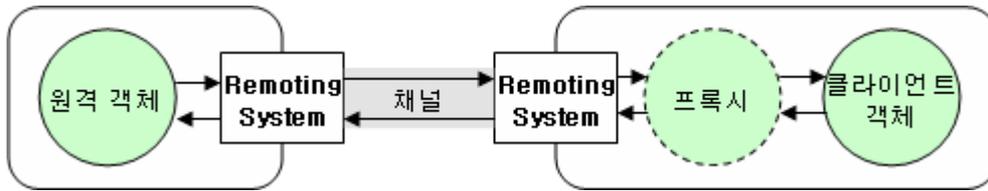
인터넷과 정보기술의 발전으로 인하여 데이터베이스 역할과 관심도가 점차 증가되고 있다. 이에 따라 사용자들은 더욱 양질의 Data 서비스와 안정된 서비스를 제공 받기 원하고, 서비스 제공자는 사용자들이 원하는 서비스를 제공하기 위해 많은 비용과 시간을 투자하여 안정된 서비스를 제공하고자 한다. 이와 같이 많은 비용과 시간을 투자하여 구축된 시스템은 대형 시스템 기반의 동 기종 소프트웨어를 선호하며, Data 관리를 위해 상용 데이터베이스 시스템을 사용한다. 또한 Data의 안정적 서비스를 위하여 필수적으로 백업 체계를 구축한다. 물론 많은 비용과 시간을 투자할 수 있다면 최적의 하드웨어와 소프트웨어로 시스템을 구축하여 최상의 서비스를 제공할 수 있을 것이다. 하지만 대형 시스템을 갖추지 못하는 환경에서는 본 논문에서 제시한 이 기종 DB 서버 간의 동기화 방식을 적용한다면 안정된 서비스를 제공하는 것은 물론 이중화된 백업 체계를 구축 할 수 있다. 또한 이 기종 DB 서버 간의 동기화 문제도 중계 어플리케이션 서버를 이용하여 Data 저장 정보를 분석하고 Data 처리수준을 실시간 Socket 통신으로 감시하기 때문에 해결이 가능하다.

이에 본 논문은 이 기종 DB 서버 간의 동기화를 위해 소프트웨어적 라우터 역할과 분산된 각 DB 서버의 관리자 역할을 수행하는 중계 어플리케이션 서버를 구현하여 안정된 서비스 시스템 설계와 백업 시스템 설계의 기초를 마련하고자 한다.

## 2. .NET Remoting 기술과 네트워크 프로그래밍

### 2.1 .NET Remoing 기술

.NET Remoting 기술은 (그림 1)과 같이 객체를 원격지에서 생성하고, 원격지에 생성된 객체를 사용할 수 있도록 하는 분산 객체 기술로 다양한 AppDomains, 프로세스, 컴퓨터에 있는 객체들이 서로 문제없이 통신할 수 있도록 풍부하고 확장 가능한 프레임워크를 제공하는 기술이다.[박상현 2006] 원격 객체 코드는 서버 활성화 객체와 클라이언트 활성화 객체 모두의 경우에서와 같이 서버에서 실행되거나 원격 객체가 클라이언트/서버 연결을 통해 일원화 된 클라이언트에서 실행된다.



(그림 1) Remoting 아키텍처

.NET Remoting 기술을 최적의 상태로 사용하기 위해서는 다음과 같이 두 가지 방식이 있다. 첫 번째는 클라이언트가 HTTP-SOAP를 통해 Remoting 객체를 호출하는 방식으로 HTTP 채널은 기본적으로 SOAP 형식 지정자를 사용하므로 클라이언트가 인터넷상에서 객체를 액세스하는 경우에 사용할 수 있다. 이 과정에서 HTTP를 사용하기 때문에 이 구성을 사용하여 방화벽을 통해 .NET 객체를 원격으로 액세스할 수 있다. 두 번째는 클라이언트가 TCP 채널을 통해 컴퓨터 경계에서 Remoting 객체를 호출하는 방식으로 TCP 채널은 Binary 형식의 지정자를 기본으로 사용한다. 이 형식 지정자는 Data를 Binary 형태로 순서화하고 원시 소켓을 사용하여 Data를 네트워크를 통해 전송한다. 이 방식은 객체가 방화벽이 있는 폐쇄된 환경에서 배포될 때 이상적이며, 객체 사이에서 소켓을 사용하여 Binary Data를 통신할 때 더욱 최적화된다.[David Conger 2003]

.NET Remoting 기술은 시스템 요구사항에 따라 강력한 원격 상호작용을 가능하게 해 주는 여러 방법을 제공하고, 웹 서비스와 자연스럽게 통합되며 여러 플랫폼에서 액세스할 수 있도록 .NET 객체를 노출하는 기능을 제공한다.

## 2.2 네트워크 프로그래밍

TCP/IP 프로토콜을 사용하는 네트워크 프로그램을 구현하는데 표준으로 사용하는 네트워크 프로그래밍은 소켓 프로그래밍이다. 네트워크 프로그램을 구현 할 때 기본이 되는 요소에는 IP, PORT, TCP/IP 프로토콜 3가지가 있다.[정의현, 김성진 2000]

### 2.2.1 IP

IP(Internet Protocol)는 호스트들에게 유일한 IP 주소인 논리적 주소를 지정한다. 이 IP 주소를 사용하여 시작지 컴퓨터와 목적지 컴퓨터를 구분하고 Data 전송을 위한 경로를 설정한다. 또한 IP 주소는 단순히 호스트에 할당된 논리적인 주소뿐만 아니라 해당 호스트가 위치하고 있는 논리적인 서브 네트워크를 나타낸다.[장중환, 이재현 2004]

### 2.2.2 PORT

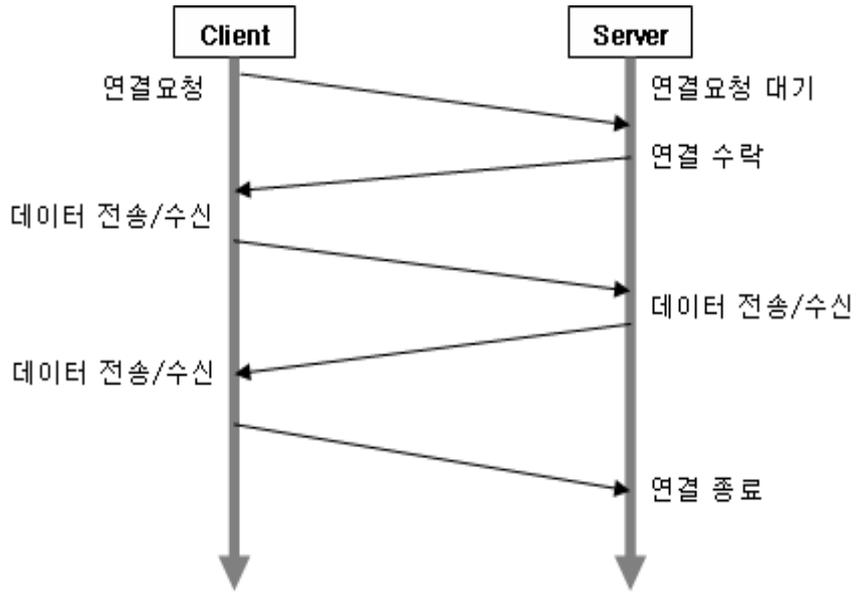
IP를 통해 수신된 Data가 어디로 가야하는지를 알 수 있게끔 하는 것이 포트이다. 컴퓨터에는 네트워크 서비스를 사용하는 여러 서비스들이 존재하며 해당 컴퓨터의 IP 주소를 통해 수신된다. 그런데 해당 컴퓨터의 네트워크 서비스들은 모두 같은 IP 주소를 사용하고 있는 관계로 수신된 Data가 어떤 서비스로 가야할 지를 알아야 한다. 즉 이렇게 수신된 Data가 각 서비스로 정확히 갈 수 있도록 도와주는 것이 포트이다.[장중환, 이재현 2004]

### 2.2.3 TCP/IP

TCP/IP(Transmission Control Protocol/Internet Protocol) 프로토콜은 인터넷의 기반을 이루는 프로토콜이다. 우리가 자주 사용하는 HTTP, FTP, Telnet이나 채팅, 파일교환 프로그램 등이 이 프로토콜로 동작한다.[박상현 2006]

TCP/IP 프로토콜은 기본적으로 Client/Server 모델을 기반으로 동작한다. (그림 2)처럼 Server는 Client의 요청을 기다리고 있다가 연결 요청이 들어오면 이것을 수락하고 연결을 맺는다. 연결이 이루어지면 Client와 Server는 각각 작업을 진행하고, 작업이 끝나면 연결을 끊는다. 이 과정이 TCP/IP 프로토콜의 기

본 동작모델이다.[장중환, 이재현 2004]

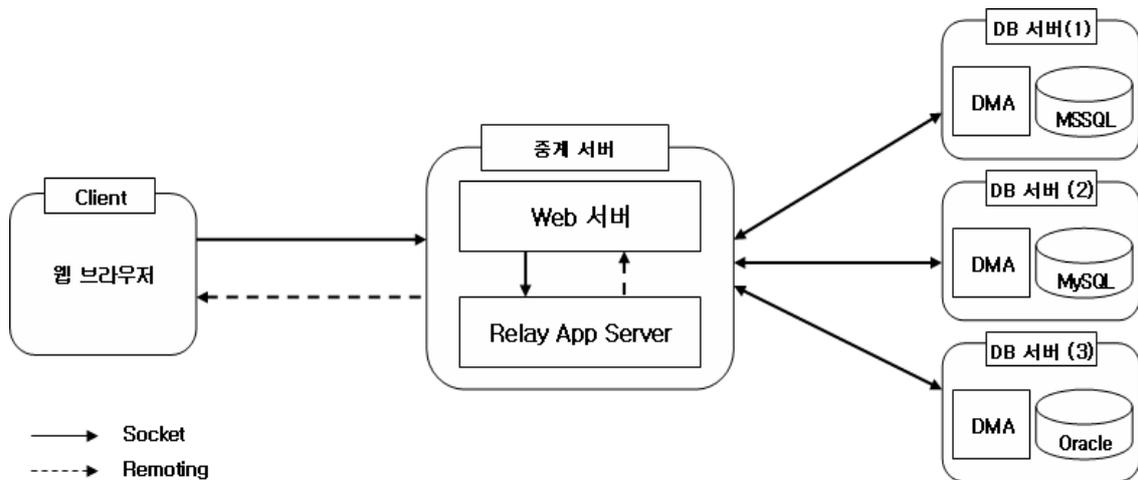


(그림 2) TCP/IP 동작 과정

### 3. 이 기종 DB 서버 간 동기화 시스템 설계 및 구현

#### 3.1 구성

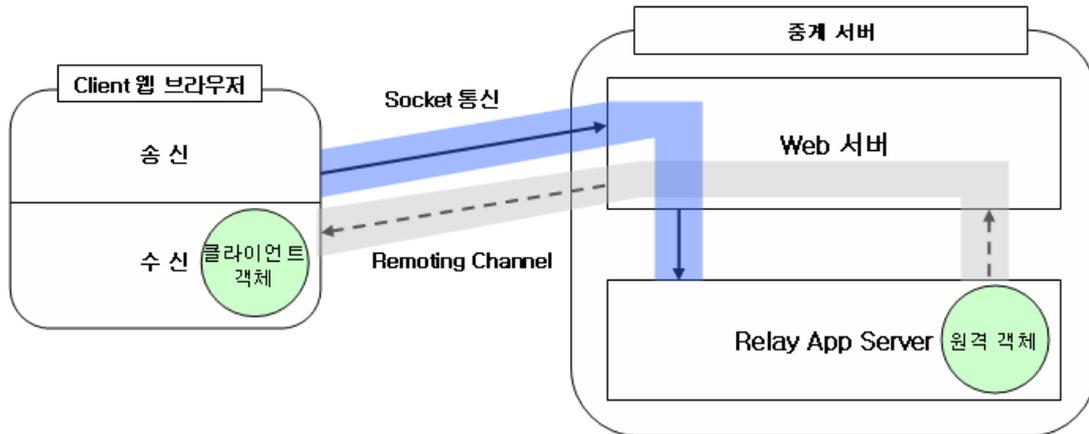
시스템은 Web 서버와 데이터베이스, RAS(Relay Application Server), DMA(Data Management Application)으로 구성된다. 중계 서버 상에 RAS를 배치하여 Client 웹 브라우저와 RAS 간의 Socket 통신과 Remoting 시스템으로 주고받은 Data를 DMA에 전송하며 DMA는 해당 DB Server에 Data를 저장하는 방식으로, Data가 DB Server에 저장되는 과정에서 RAS는 각각의 DB Server의 Data 처리 수준을 실시간으로 감시하면서 Data를 전송하게 된다. 만약 Data 처리수준이 기준 이하이거나 DB Server가 정상 또는 비정상 종료되었을 경우는 해당 DMA를 제외한 나머지 DMA에 Data를 전송하게 된다. 이후 Data 처리수준이 향상되거나 정상 상태로 되었을 때, RAS는 각각의 DB Server의 Data를 동기화 시켜주는 작업을 한다. 다음의 (그림 3)은 시스템의 구성을 보여준다.



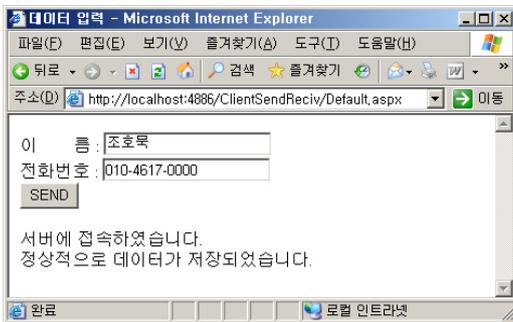
(그림 3) 이 기종 DB 서버 간의 동기화 시스템 구성

### 3.2 사용자와 중계서버 간 Data 처리

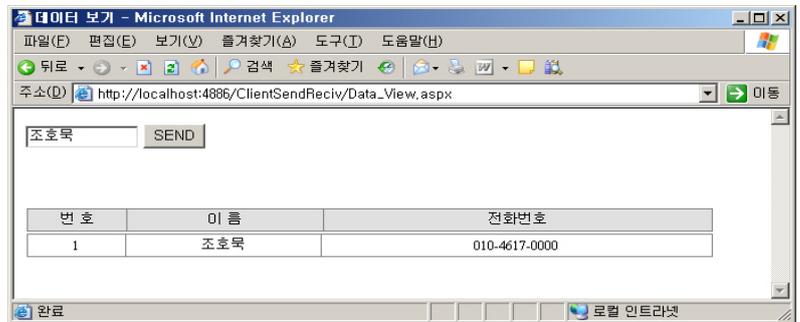
사용자와 중계 서버 간의 Data 처리 과정은 (그림 4)와 같이 웹 브라우저의 Socket 통신과 Remoting Channel을 이용하여 Data를 송·수신하여 처리된다. Data 송신은 Web 서버를 중심으로 Client에 웹 서비스를 제공하고, Client가 [SEND] 버튼을 누르면 Socket을 열고 Data를 RAS에 전달한다. 이 때 RAS에 Data 전달이 완료되면 RAS는 연결 해제 키워드를 Client에 전달하고 Socket 클래스에서 제공 받아 생성된 통신 객체들을 제거하여 다중 접속과 통신이 가능하도록 구현하였다. Data 수신은 Web 서버의 .NET Remoting 서비스를 이용하여 RAS의 원격 객체를 Remoting Channel로 Client 웹 브라우저에서 실행하는 것처럼 사용하고 이 원격 객체에 DB 서버의 DMA에서 받은 Data를 Remoting Channel로 가져와 Client 웹 브라우저에 나타내도록 구현하였다.



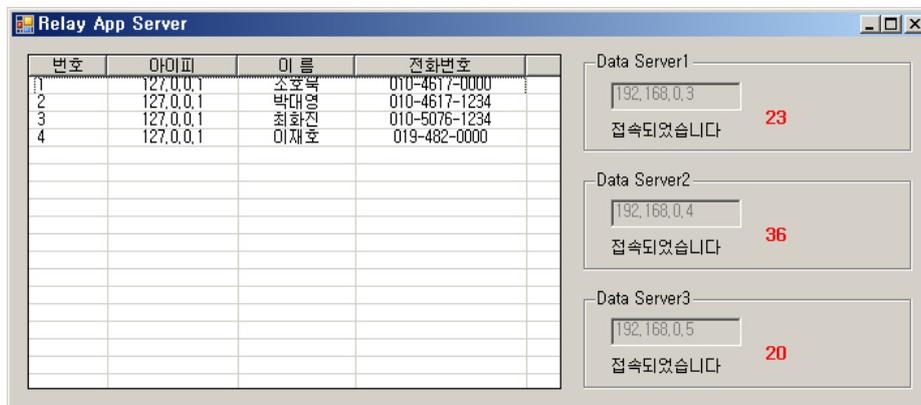
(그림 4) 사용자와 중계서버 간의 Data 처리 과정



a. Data 송신



b. Data 수신

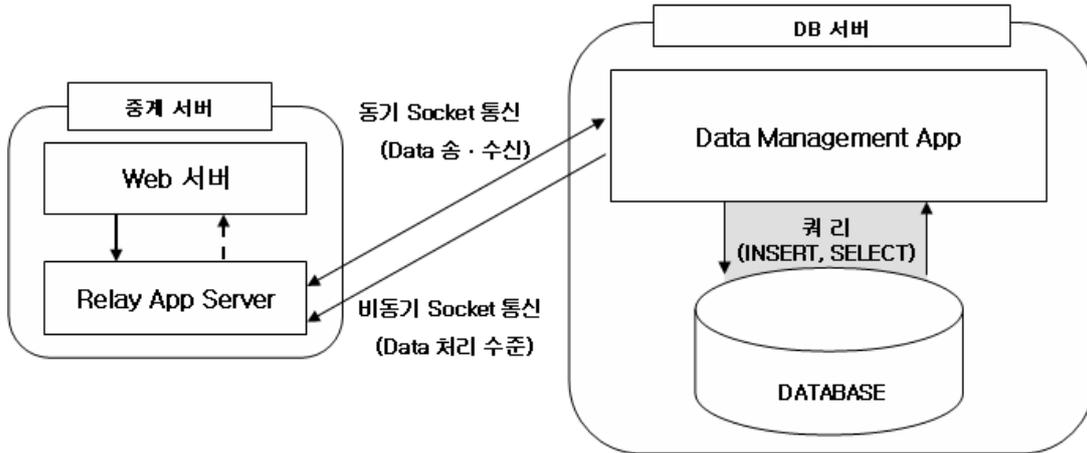


c. Relay Application Server

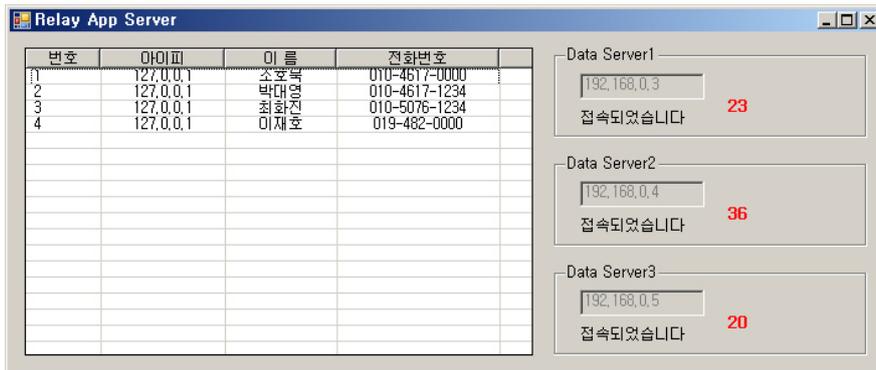
(그림 5) Client 웹 브라우저를 이용한 Data 송·수신 처리

### 3.3 중계서버와 DB 서버 간의 Data 처리

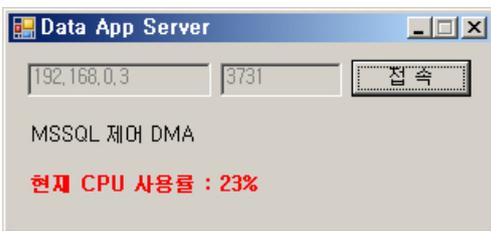
중계서버와 DB 서버 간의 Data 처리 과정은 (그림 6)와 같이 Socket 통신으로 Data를 송·수신한다. DB 서버에 Data를 쓰고 읽는 방식은 Client에서 전송받은 Data를 RAS가 동기 Socket 통신으로 DMA에 전달하고 DMA는 INSERT, SELECT 쿼리 구문으로 Data를 DB 서버에 읽고 쓰는 형태로 구현하였다. 이 방식은 중계서버의 로컬에서 DB를 직접 액세스 할 수 있어 중계서버와 DB 서버의 부하를 줄일 수 있다. DB 서버에서 Data를 가져오는 경우는 비동기 Socket 통신으로 DB 서버의 Data 처리 수준을 실시간에 RAS로 전달한다. Data 처리 수준을 고려하면서 DB 서버들의 Data를 동기화하는 과정을 (그림 8)에서 자세히 설명한다.



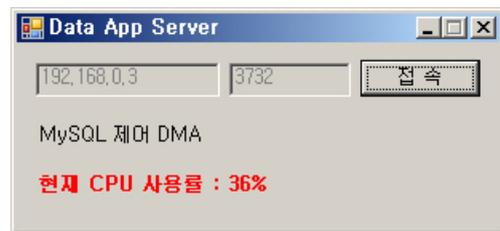
(그림 6) 중계서버와 DB 서버 간의 Data 처리 과정



a. Relay Application Server



b. MSSQL 제어 Data Management Application



c. MySQL 제어 Data Management Application

(그림 7) RAS와 DMA 간의 접속 및 Socket 통신

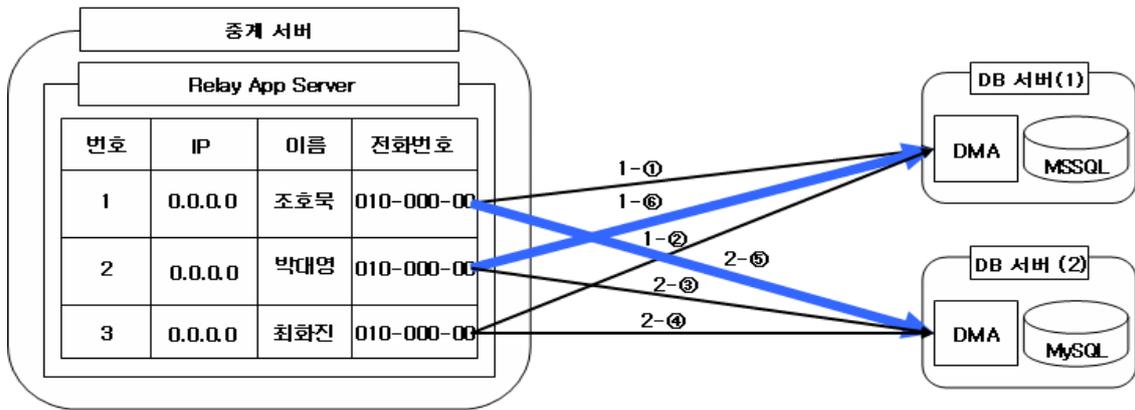
### 3.4 이 기종 DB 서버 간의 Data 동기화

서버의 부하 및 성능 평가 기준으로 다음과 같은 지표가 있다.

- 동시 접속 유저 수(Number of Virtual Users)
- 시간당의 유저 시나리오 실행 수(Transactions per hour)

- 시간당의 페이지 처리 수(Received pages per hour)
- 시간당의 히트수(Hits per hour)
- 서버 평균 응답 시간(Average Server times, etc. )
- 서버 측의 CPU 사용률

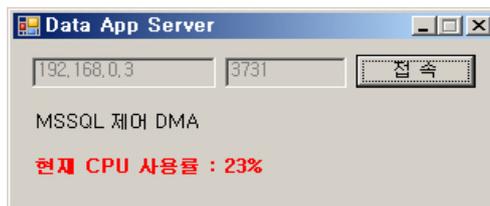
위의 부하 및 성능 평가 기준을 모두 적용하여 시스템을 구현 할 수도 있겠으나 본 논문에서 위의 기준에서 서버 측의 CPU 사용률을 지표로 삼아 구현하였다. CPU 사용률이 100% 이상으로 한계에 도달하면 일단 서비스를 중지하는 등의 판단을 하지만 그 때에도 일정 시간 이내에 응답하는 시스템이 존재하기 때문에, 어디까지나 시스템 관리자가 지표를 정하여 최적화 된 서비스를 구현하는 것이 적당하리라 판단된다.[http://www.empirix.co.kr 엠펜릭스 코리아] 이와 같이 (그림 8)에서는 DB 서버의 처리 수준 즉, CPU 사용률을 DMA에서 RAS로 보내주고 분석된 Data 처리 수준을 판단하여 DMA에 Data를 전송하여 DB에 저장되게 된다. (그림 8)에서 볼 수 있듯이 ‘1-⑥’, ‘2-⑤’ Data 처리 순서가 Sequential Access가 아닌 DB 서버의 부하도를 판단하여 정상 상태가 될 때 입력된 결과를 알 수 있다. 이 때 RAS에서는 ListView 윈도우 컨트롤을 이용하여 Client에서 전송된 Data를 저장하게 되고 이를 기준으로 DB 서버에 Data를 다시 전송하는 방식으로 구현하였다. RAS와 DMA의 Data 처리 순서 정보를 잃지 않고 검색을 빠르게 하기 위해 레지스트리 정보를 사용하였다.



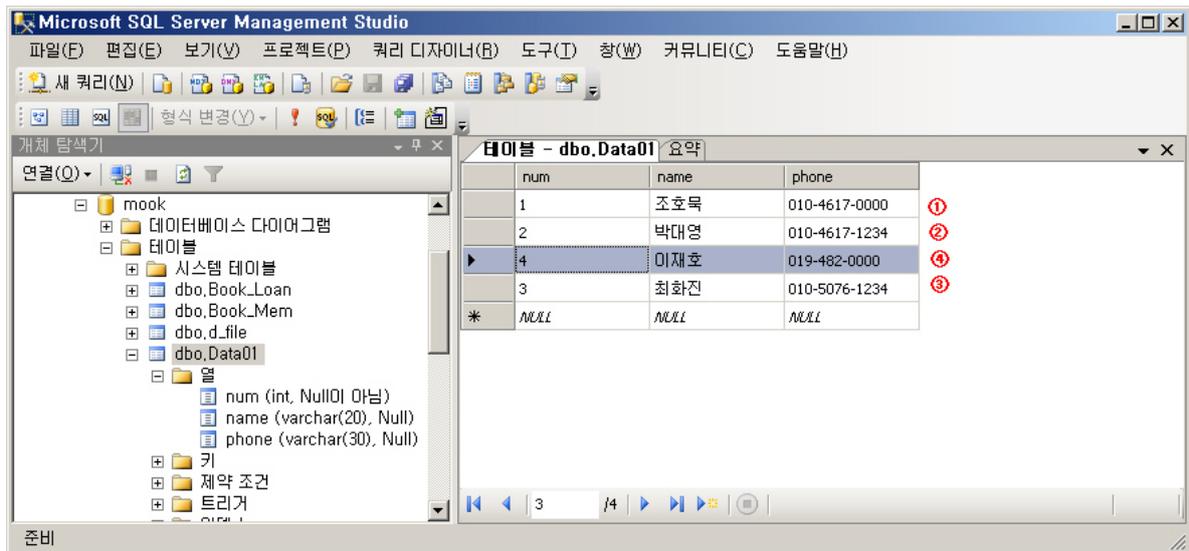
(그림 8) 이 기종 DB 서버 간의 Data 동기화 과정



a. Relay Application Server



b. MSSQL 제어 Data Management Application



c. MSSQL 제어 DMA에 의해 동기화 된 MSSQL 2005의 결과 화면

(그림 9) Data 동기화 결과

#### 4. 결론

본 논문에서는 .NET Remoting 기술과 네트워크 프로그래밍을 이용하여 Data 처리를 소프트웨어적 라우터 역할과 분산된 이 기종 DB 서버의 관리자 역할을 수행하는 중계 어플리케이션 서버를 구현하여 안정된 서비스와 백업 시스템을 설계하고 구현하였다. 하지만 본 논문에서는 대형 시스템이 아닌 중소형 시스템에 적합한 모델을 제시하였고 여러 기술을 집약하여 대형 시스템에 적합한 중계 어플리케이션 서버를 구현한다면 더욱 많은 분야에서 활용이 가능 하리라 확신한다. 또한 본 논문에서는 서버의 부하 및 성능 평가 기준을 CPU 사용률만으로 구현하였지만 보다 정확한 동기화 시스템을 구현하려면 3.4에서 제시한 여러 지표를 복합적으로 적용함으로써 정확한 서버 부하도를 측정함이 타당하겠다.

#### 참고문헌

- [1] 김연홍 「SQL Server 2005를 위한 데이터베이스 모델링」, 1판, 디비바다미디어, 2007
- [2] 김정훈 「TCP/IP 소켓 프로그래밍」, 1판, 교학사, 2003
- [3] 박상현 「비주얼 C# 2005 익스프레스로 배우는 C# 2.0 프로그래밍」, 1판, 도서출판 대림, 2006
- [4] 신재영 「기초에서 활용까지 한글 Visual C# .NET」, 1판, 정보문화사, 2002
- [5] 이상엽 「Visual C++ Programming Bible Ver.6.X」, 1판, 영진출판사, 2000
- [6] 장중환, 이재현 「Visual Basic.NET 프로젝트 따라하기」, 1판, 컴스페이스, 2004
- [7] 정성권, 최성우, 하민규 「about .NET XML 웹서비스」, 1판, 영진출판사, 2002
- [8] 정의현, 김성진 「자바 2」, 1판, 도서출판 대림, 2000
- [9] David Conger 「Remoting with C# and .Net」, 1판, WILEY, 2003
- [10] Marco Bellinaso, Kevin Hoffman 「ASP.NET WebSite Programming」, 1판, 정보문화사, 2002
- [11] Sean Dillon, Christopheher Beck, Thomas Kyte, Joel Kallman, Howard Roges 「Beginning Oracle Programming」, 1판, 정보문화사, 2005
- [12] Rebecca M.Riordan 「ADO.NET」, 1판, 정보문화사, 2002
- [13] <http://www.empirix.co.kr>, 웹 어플리케이션 성능관리 및 테스트



Ho-Mook, Cho

e-mail : mook79@empal.com

2003년 한림대학교 컴퓨터공학과(학사)

2006년 아주대학교 정보통신대학원 정보보호학과(석사)

현재 : 육군 제2군사령부 정보체계지원실 정보보호반장(대위)

관심분야 : Steganography, Wireless Network Security



Kil-Lae, Kim

e-mail : brad0212@unitel.co.kr

1987년 경북대학교 문헌정보학과(학사)

1992년 동국대학교 경영대학원 전자계산학과(석사)

2002~2006년 육군과학화훈련장(KCTC) 구축사업 체계통합팀장

현재 : 육군 제2군사령부 정보체계지원실장(중령)

관심분야 : Modeling & Simulation, S/W Life Cycle