
분산 스트리밍 서버를 이용한 원격 실험 재연을 지원하는 그리드 기반 연구 시스템의 구현

장 선* · 이장호**

Implementation of the Grid-Based Research System Supporting Remote Experiment Replay
using Distributed Streaming Server

Sun Jang* · Jang Ho Lee**

이 논문은 2007년도 홍익대학교 학술연구진흥비에 의하여 지원되었음

요 약

원격지로부터 원격 실험을 수행하는데 있어서 수행된 실험 결과 데이터를 저장하고 재사용하여 실험을 재연하는 그리드 기반의 연구 시스템을 제안한다. 제안된 시스템은 Korea Construction Engineering Development (KOCED) 프로젝트에서 구축중인 실시간 하이브리드 실험 시설을 모델로 하여 설계되었다. 다양한 데이터를 사용자에게 제공하기 위하여, 실험 결과 데이터는 Publish/Subscribe 기반의 분산된 스트리밍 서버를 이용하여 관리되었다. 본 프로젝트의 데이터 특성을 고려하면서, 하나의 스트리밍 서버를 사용한 경우와 여러 개의 분산된 스트리밍 서버들을 사용한 경우를 시뮬레이션 하여 성능을 비교 및 분석하였다.

ABSTRACT

We propose an grid-based research system that replays the experiment by storing and reusing the previous experiment result data in performing remote experiment from a distance. The proposed system was designed for the real-time hybrid test facility which is being built in Korea Construction Engineering Development(KOCED) Project. To provide users with various data, the experiment result data are maintained by the distributed streaming servers based on the publish/subscribe paradigm. Based on the data characteristics of our project, we compared and analyzed the performance of the one-streaming-server case and the distributed-multiple-streaming-server case.

키워드

그리드, 원격 실험, Publish/Subscribe, 하이브리드 실험, OGSA

* 동양시스템즈 연구원

접수일자 : 2008. 2. 11

** 홍익대학교 컴퓨터공학과 조교수, 교신저자

I. 서론

전 세계적으로 그리드 컴퓨팅 기술을 사용하여 의학, 물리학, 천체학, 화학, 토목 등 다양한 과학 연구 분야에서 공동 연구 환경을 구축하기 위한 다양한 프로젝트가 진행되고 있다[1]. 그 중에서, 대표적으로 미국의 16개의 지진공학 연구 실험 시설을 그리드 시스템으로 연결한 NEES(Network for Earthquake Engineering Simulation)[2]와 국내의 토목 연구 실험시설들을 그리드 기반의 시스템을 이용하여 공유하는 KOCED(Korea Construction Engineering Development)[3]가 있다.

작은 기관에서는 구축에 막대한 비용이 들어가는 대규모의 토목 실험 시설을 만드는 것이 어렵고, 연구자가 실험을 할 때마다 멀리 떨어진 다른 대규모 실험 시설을 방문하기도 쉽지 않다. 따라서, 원격지에서 네트워크를 통해서 실험에 참가할 수 있는 기능이 필요하다. 그리고, 비용이 많이 드는 실험을 수행함으로써 생기는 다양하고 방대한 데이터는 저장하여 실험 후 나중에 다른 연구자가 공유하고 검색해서 다시 재연(replay)할 수 있게 하면, 불필요한 중복된 실험을 방지하고 연구의 효율성을 높일 수 있다[4].

본 논문에서는 KOCED의 실험 시설 중 하나인 실시간 하이브리드 다자유도 실험 시설을 위해 구축된 그리드 기반의 원격 실험 시스템을 설명하고, 효율적으로 실험 결과를 저장하고 재연하기 위한 시스템의 설계 및 구현을 설명한다.

본 논문에서 구현된 그리드 서비스들은 개방형 그리드 표준인 Open Grid Services Architecture (OGSA)[5]에 기반을 둔 글로버스 툴킷(Globus Toolkit) [6]을 사용하여 구축되었다. 글로버스 툴킷은 그리드 컴퓨팅 환경 구축을 위한 핵심 서비스들을 그 목적에 따라서 다양한 컴포넌트 형태로 구현해 놓은 툴킷이다. 본 논문에서 제안된 시스템에서, 저장된 데이터들을 사용자가 공유할 수 있는 환경으로 이동시키는데 사용된 주요 컴포넌트에는 GridFTP[7]와 Reliable File Transfer (RFT)[8]가 있다.

GridFTP는 Global Grid Forum에 의해 정의된 파일 전송 프로토콜로서 인증된 사용자에게 빠르고 효과적이며 안전한 파일 전송을 제공한다. RFT는 대용량 데이터의 전송에 있어서 두 개의 GridFTP를 연결하여 데이터를 이동시키면서 데이터의 이동 상태를 데이터베이스를 사용하여 관리함으로써 데이터 전송 도중 발생 할 수 있

는 네트워크 장애나 장비의 결함에 의한 데이터 손실을 효과적으로 방지한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구들을 소개하고, 3장에서는 시스템 구축의 대상 모델인 실험시설과 원격실험에 대하여 설명하고 4장에서는 실험 결과를 저장 및 재연하기 위한 그리드 서비스의 구현에 대해 설명한다. 5장에서는 단일 서버 및 분산 서버 환경에서 구축된 시스템의 성능을 비교, 분석하고, 마지막으로 6장에서 결론 및 향후 연구 방향을 설명한다.

II. 관련 연구

본 논문에서 추구하는 바와 같이 그리드 컴퓨팅 기술에 기반 한 연구 환경에서의 다양하고 방대한 데이터들이 효율적인 저장, 관리, 사용하기 위한 대표적인 프로젝트로는 G-Civil[9]과 NEES[2]가 있다.

G-Civil 은 University of Southampton에서 시작된 유럽의 e-Science 프로젝트 중 하나이다. G-Civil의 목적은 사용자가 원격지의 건설 현장과 분석 데이터들을 인터넷 포털을 통하여 실시간으로 관찰할 수 있는 서비스를 제공하는데 있다. 이를 위해 다양한 데이터의 수집, 데이터의 배포, 데이터를 시각적으로 표현해 주는 툴들이 포함된 프로토타입을 제공한다.

NEES는 15개의 지진 공학 연구 실험 시설뿐만 아니라, 다양한 협업 도구들과 데이터 저장소, 지진 시뮬레이션 소프트웨어 등을 그리드 컴퓨팅 기술을 이용해 연결해 놓았다. 또한 연구자들로 하여금 실험 데이터를 공유하게 하기 위하여 Data Turbine을 스트리밍 서버로 사용하여 실험 결과를 재연하는 환경을 제공하고 있다.

실험 데이터들을 저장하고 재연하기 위하여, 데이터는 데이터 획득 컴퓨터(Data Acquisition PC(DAQ PC))로부터 리코딩 서비스로 전달되어야 하고, 리플레이 서비스로부터 사용자에게 전달되어야만 한다.

본 논문에서 실험 결과 데이터를 저장하고 재연하기 위하여 서버 구조로서 Publish/Subscribe 패러다임[10]에 기반한 나라다 브로커링(Narada Brokering)[11]을 사용하고 있다. Publish Subscribe란 스트리밍 서버에 데이터를 전송해주는 Publisher가 토픽이라는 메타데이터와 관련하여 전송하고, 데이터를 스트리밍 받기를 원하는 Subscriber는 관심 있는 데이터에 대한 토픽을 스트리밍

IV. 원격 실험의 재연(Replay)

본 논문에서는 원격 실험의 결과데이터의 저장 및 재연을 위하여 리코딩 서비스(Recording Service), 리플레이 서비스(Replay Service), HEMS(Hybrid Experiment Managing Service)를 구현하였고 글로버스 툴킷의 RFT 서비스와 GridFTP 서버를 이용하였다. 그림 4는 리코딩 및 리플레이 서비스를 위한 전체 시스템의 연관 관계를 나타내고 있다. (그림 4에서 발간은 Publish를, 구독은 Subscribe를 의미한다.)

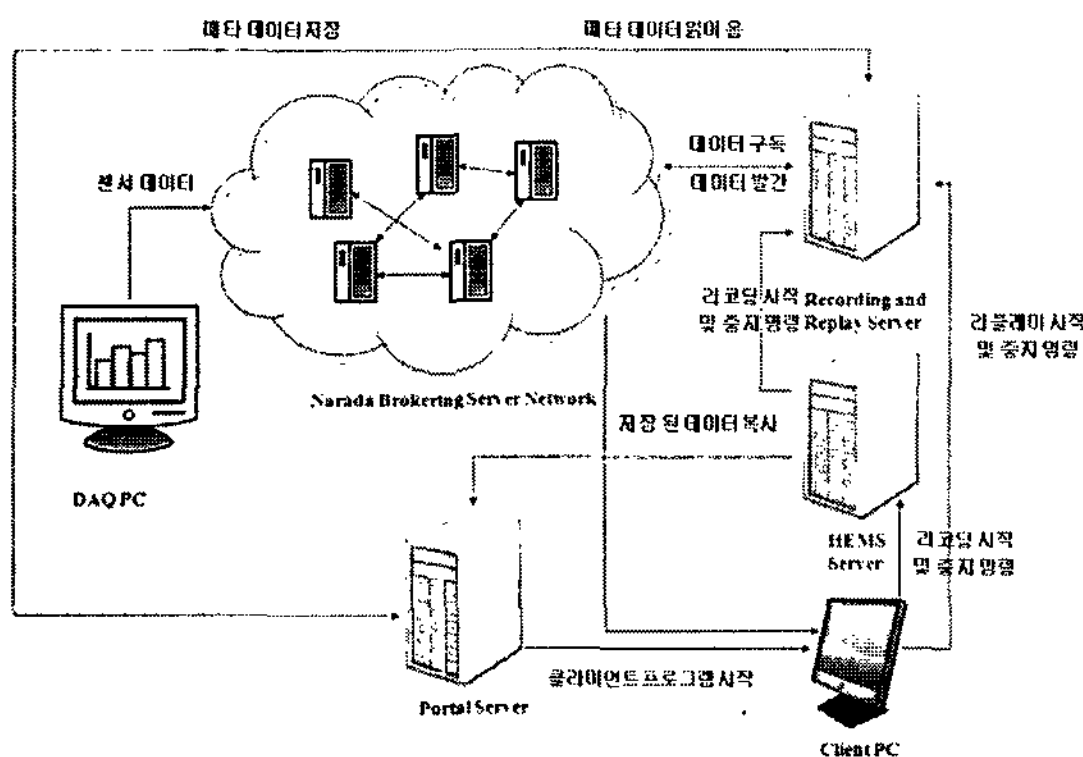


그림 4. 원격 실험의 기록 및 재연

Fig. 4. Recording and replay of remote experiment

리코딩 및 리플레이 서버는 리코딩, 리플레이 서비스를 수행하며, 클라이언트나 HEMS 서버로부터 서비스 요청을 받아 나라다 브로커링 서버네트워크로부터 데이터를 Subscribe해서 데이터를 저장하거나, 포털 서버에 저장된 데이터를 읽어서 나라다 브로커링 서버에 Publish한다.

나라다 브로커링 서버 네트워크는 Publish Subscribe 기반의 스트리밍 서버 네트워크로서 데이터 획득 컴퓨터나 리코딩 및 리플레이 서버 내부의 리플레이 서비스에 의해 Publish되는 데이터들을 스트리밍 해 주는 역할을 한다.

데이터 획득 컴퓨터는 물리적 구조물에 부착되어있는 센서로부터 읽은 데이터를 나라다 브로커링 서버 네트워크로 옮겨주는 역할을 한다.

HEMS(Hybrid Experiment Managing Service)는 클라이언트로부터 받은 리코딩 명령에 따라 리코딩 및 리플

레이 서버 내부의 리코딩 서비스를 호출하고 RFT와 GridFTP를 통하여 리코딩 서비스가 저장해 놓은 실험 결과 데이터를 포털 서버의 저장소로 옮겨준다.

포털 서버는 사용자 공유 환경인 웹 포털이 제공되는 웹 서버로서 클라이언트 프로그램이 이곳에서부터 다운로드 받아 자바 웹 스타트 형식으로 시작되고 리코딩 서비스에 의해 실험 결과 데이터가 저장된다.

클라이언트는 사용자의 컴퓨터를 의미하며 클라이언트 프로그램을 포함하고 있다. 원격 실험의 요청과 원격 모니터링, 그리고 리코딩 및 리플레이 서비스의 시작/중지 요청을 한다.

그림 5에서 1부터 14까지는 리코딩 서비스를 하기 위한 시스템 요소들 간의 상호 작용의 순서를 보여준다.

먼저 사용자가 실험 결과 데이터의 리코딩 시작 요청을 클라이언트 프로그램에 내리게 된다.

명령을 받은 클라이언트 프로그램이 현재 모니터링하고 있는 데이터들의 토픽 정보를 포함하여 HEMS에 실험 데이터 리코딩 시작 요청을 전달하면, HEMS는 클라이언트 프로그램으로부터 받은 토픽 정보와 함께 리코딩 서비스를 호출하여 데이터 저장의 시작을 요청하게 된다.

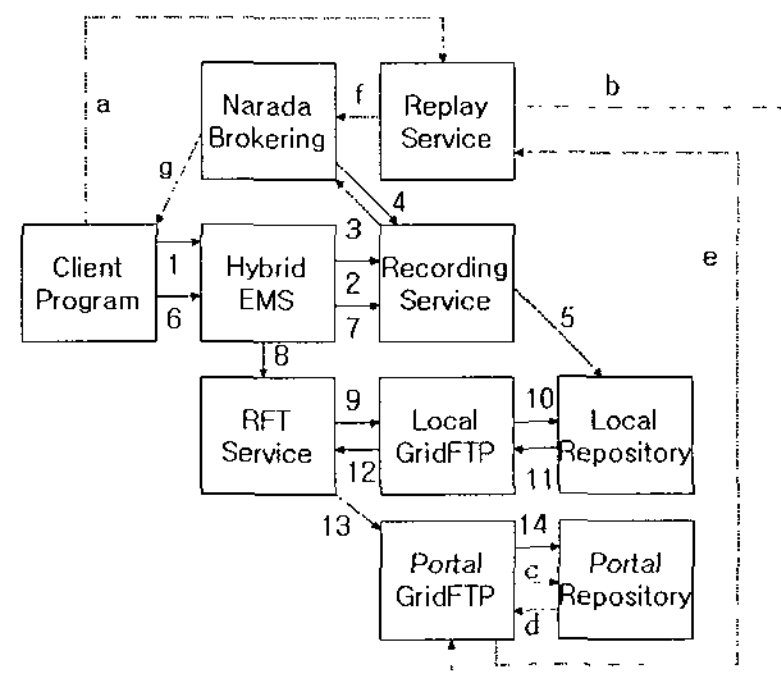


그림 5. 리코딩 및 리플레이 시의 실행 구조 (1-14는 리코딩시의 순서를, a-f는 리플레이시의 순서를 나타냄)

Fig. 5. Run-time architecture of recording and replay (1-14 shows recording sequence while a-f shows replay sequence.)

리코딩 서비스는 HEMS로부터 전달받은 토픽에 해당하는 데이터를 나라다 브로커링에서 Subscribe해 와서 로컬 저장소에 임시 저장한다.

사용자로부터 리코딩 정지 요청이 들어올 때 까지 리코딩 서비스는 해당 토픽에 대한 데이터들을 나라다 브로커링으로부터 스트리밍 받아 주기적으로 로컬 저장소에 저장한다.

리코딩 도중에 사용자로부터 리코딩 정지 요청이 들어오면 클라이언트 프로그램은 이를 HEMS로 전달하고, HEMS는 리코딩 서비스에 데이터 리코딩 정지 요청을 내려 데이터의 저장을 멈추게 되고 RFT서비스를 호출한다.

마지막으로, RFT서비스는 리코딩 서비스가 로컬 저장소에 임시로 저장해 놓은 데이터를 사용자 공유 환경인 포털서버의 저장소로 옮긴다. 이때 RFT는 로컬 GridFTP를 통해 로컬 저장소에 접근하고 포털 GridFTP를 통해 포털 저장소에 접근하여 안전하고 효율적인 데이터 전송을 담당하게 된다.

사용자는 이미 진행되었던 실험의 결과 데이터를 보고 싶을 경우 그림 5의 a부터 f까지의 순서에 이르는 리플레이를 통해서 해당 데이터를 관측 할 수 있다.

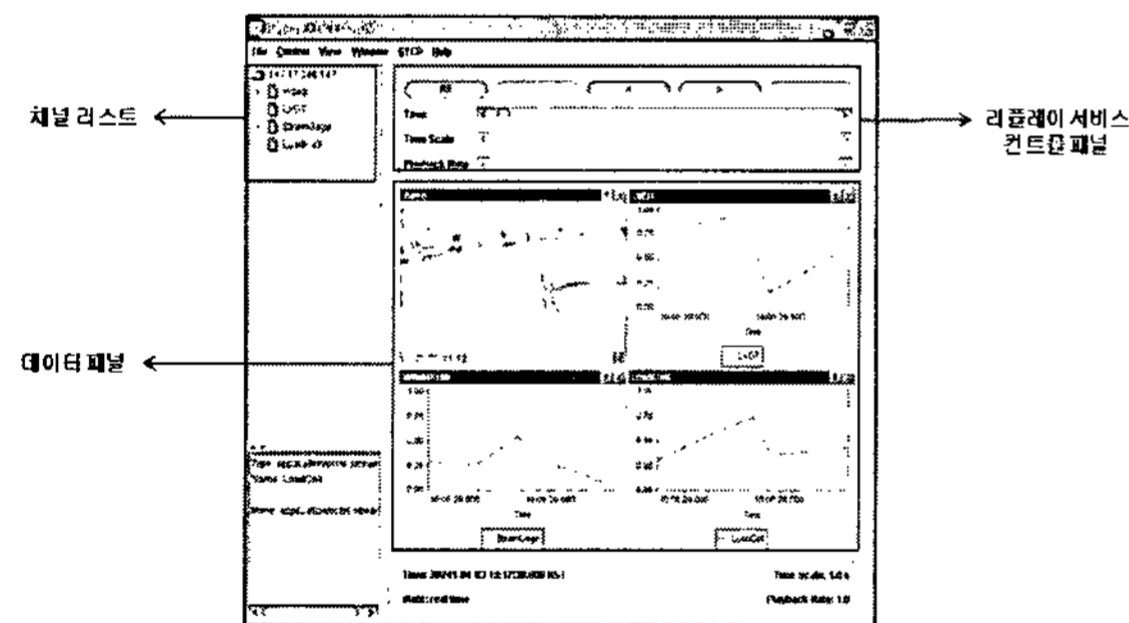


그림 6. 실험 재연 중인 클라이언트
Fig. 6. Client replaying the experiment

먼저 클라이언트 프로그램은 사용자의 리플레이 요청이 들어오면 리플레이 서비스를 호출하여 실험 결과 데이터의 재연을 시작한다.

리플레이 서비스는 포털 GridFTP에 접속하여 포털 저장소에 저장되어 있는 데이터들을 읽어서, 나라다 브로커링에 해당 데이터들을 순차적으로 Publish하게 되고 클라이언트 프로그램에서는 나라다 브로커링에 접속하여 스트리밍 되는 데이터들을 그림 6과 같이 관찰 할 수 있다.

V. 성능 평가

성능 평가를 위한 시뮬레이션을 수행하기 위하여 본 논문에서는 Network Simulator (NS2) [16]를 사용하였다. 서버는 P4-3.0Ghz의 CPU와 2GB의 메모리의 사양을 지녔다.

우선 하나의 나라다 브로커링 스트리밍 서버를 사용하는 경우의 클라이언트의 평균 데이터 전송 양을 측정하였다. 그리고, 데이터 복제를 통한 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용하는 경우의 클라이언트의 평균 데이터 전송 양을 측정하였다. 그리고 두 가지 상황에서의 성능을 비교, 분석 하였다. 그림 7은 단일 나라다 브로커링 스트리밍 서버를 사용하는 경우의 네트워크 토폴로지를 나타낸다.

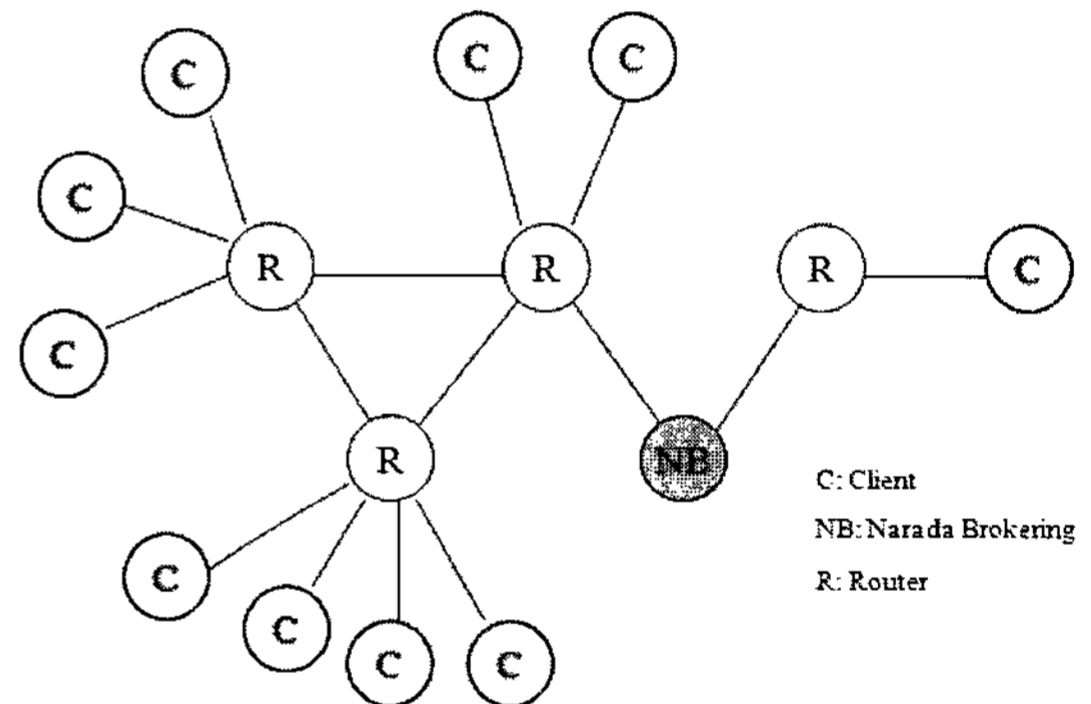


그림 7. 단일 나라다 브로커링 서버를 사용하는 경우의 네트워크 토폴로지

Fig. 7. Network topology for single NaradaBrokering server

‘NB’ 라고 표시된 하나의 나라다 브로커링 스트리밍 서버가 존재한다. 이것은 데이터 스트림의 소스로서, 리플레이 서비스에 의해 데이터가 생산되는 곳이다. ‘C’로 표시된 10개의 클라이언트들은 나라다 브로커링 스트리밍 서버에 접속하여 토픽 정보에 기반 하여 데이터를 획득하게 된다. 또한 ‘R’이라고 표시된 4개의 노드들은 단순한 라우터의 역할을 수행하며 나라다 브로커링 스트리밍 서버로부터 받은 데이터를 클라이언트들에게 전달해 주는 역할을 한다. 이들은 생산된 데이터를 전달해 준다는 부분에 있어서 나라다 브로커링 스트리밍 서버와 비슷한 역할을 하지만, 차이점은 나라다 브로커링 스트리밍 서버와 클라이언트사이의 연결 설정에 전혀

관여 하지 않고 토픽 정보에 의한 데이터 전달이 아니라 모든 데이터를 전달한다는 점에 있다.

모든 연결 링크들은 1Mbps의 대역폭과 10ms의 지연 시간을 가진 duplex-link로 정의되었다. 링크의 공평한 사용을 위하여 라우터에서는 Stochastic Fair Queuing 알고리즘이 사용되었다.

그림 8은 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용하는 경우의 네트워크 토폴로지를 보여주고 있다.

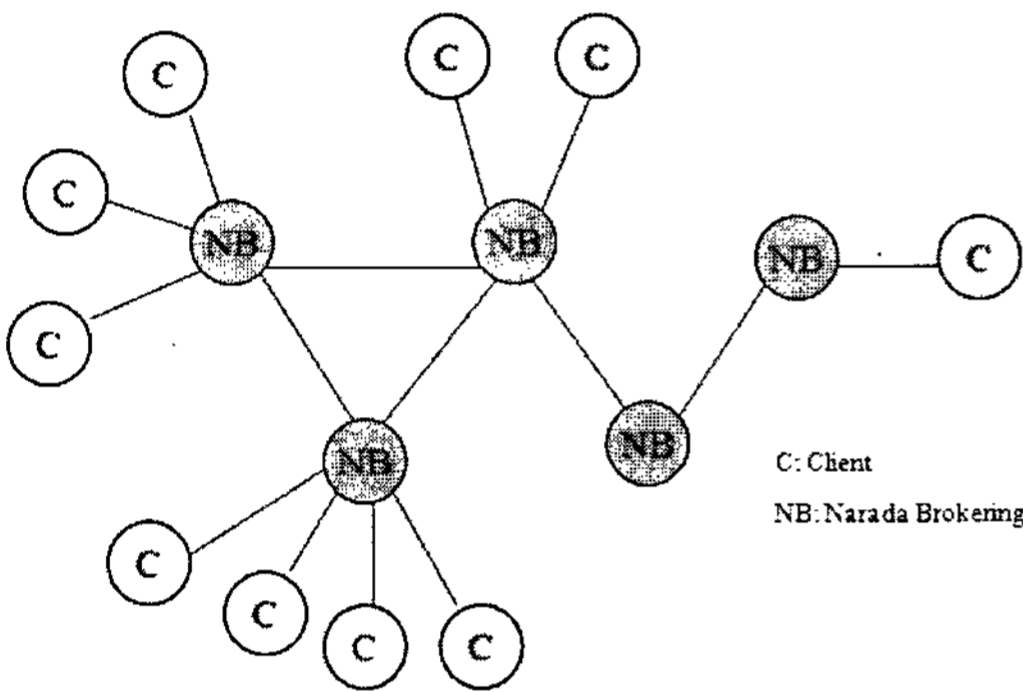


그림 8. 분산 나라다 브로커링 서버를 사용하는 경우의 네트워크 토폴로지
Fig. 8. Network topology for distributed NaradaBrokering servers

하나의 나라다 브로커링 스트리밍 서버를 사용하는 경우와의 차이점은 'NB'라고 표시되어있는 5개의 나라다 브로커링 스트리밍 서버가 분산되어 복제된 데이터를 관리 한다는 점이다. 따라서 클라이언트들은 나라다 브로커링 스트리밍 서버와의 연결설정을 가장 가까운 서버와 함으로써 병목현상을 피할 수 있다.

성능평가에서 사용된 KOCED 프로젝트의 데이터들은 몇 가지 특성을 가지고 있다. 데이터는 비디오 데이터와 센서 데이터를 포함한다.

첫째로, 데이터의 사이즈는 상황에 따라서 커질 수 있다. 스트리밍 되는 데이터의 bit rate는 다음과 같은 식에 의해 계산 될 수 있다.

본 논문에서는 프레임 당 25Kbytes의 크기를 가지고 있고 24Fps를 가지고 있는 하나의 비디오 스트림 데이터를 가정 하였으므로, 비디오 데이터의 bit rate는 이들의 곱인 600 Kbytes이다. 또한 하나의 샘플당 90Bytes의 크기를 가지고 있고 평균 초당 3번의 샘플링 주기를 가지

고 있는 3개의 센서 데이터를 가정하였으므로, 센서 데이터의 bit rate는 이들의 곱인 810 Bytes로 계산될 수 있다. 따라서 하나의 클라이언트에게 전달되어야 하는 UDP 데이터 스트림의 bit rate는 비디오 데이터의 bit rate와 센서 데이터의 bit rate의 합인 약 601Kbytes로 가정하여 시뮬레이션을 수행하였다.

두 번째로 본 논문에서의 센서 데이터의 샘플링 주기는 일정하지 않다. 샘플링 주기는 실험의 종류에 따라서 다양하게 변화 될 수 있다. 따라서 성능 평가를 위해 데이터 스트림의 생성을 2초 동안 Burst 하고 1초 동안 Idle 하게 설정하여 시뮬레이션을 수행하였다.

본 논문에서는 이러한 데이터 특성들에 초점을 맞추어 시뮬레이션을 수행하였다. 그림 7은 시뮬레이션의 결과를 보여준다.

X 축은 시뮬레이션의 시간을 초 단위로 나타내고 있다. Y 축은 클라이언트들의 평균 수신 bit rate를 Kbps 단위로 나타내고 있다.

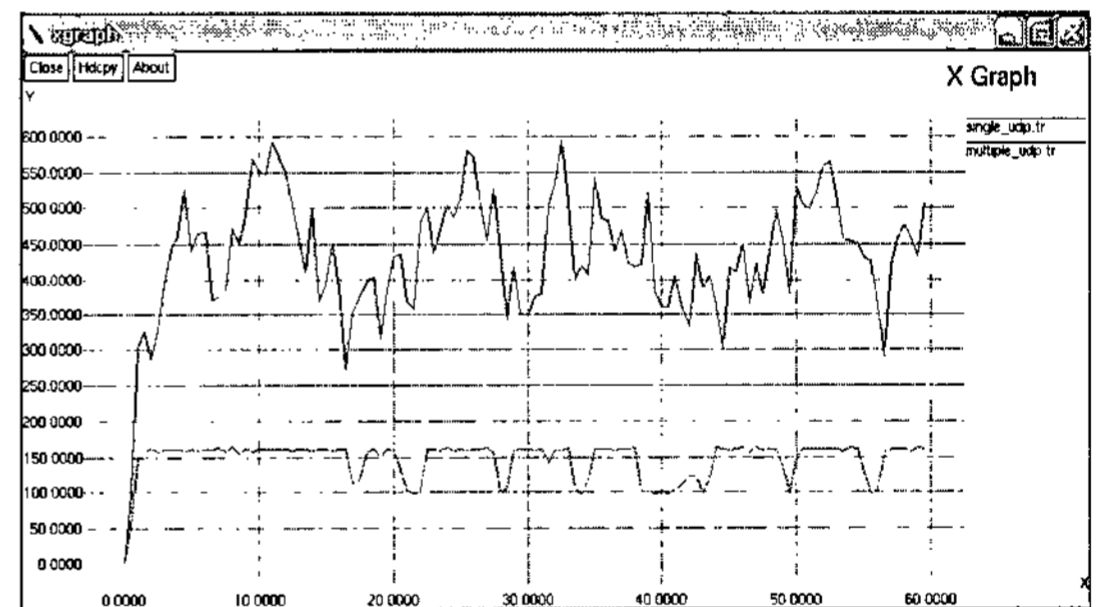


그림 9. 단일 나라다 브로커링 스트리밍 서버를 사용한 경우와 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용한 경우의 클라이언트들의 평균 수신 bit rate
Fig. 9. Average receiving bit rate of clients for single NaradaBrokering server and for distributed NaradaBrokering servers

그림 9에서 아래의 그래프는 단일 나라다 브로커링 스트리밍 서버를 사용한 경우의 클라이언트들의 평균 수신 bit rate를 나타내고 위의 그래프는 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용한 경우의 클라이언트들의 평균 수신 bit rate를 나타낸다.

표 1. 단일 서버와 분산 서버의 평균 및 최대 수신 bit rate

Table 1. Average and maximum receiving bit rate for single server and for distributed servers

	평균 수신 bit rate (Kbps)	최대 수신 bit rate (Kbps)
단일 서버	145.7	164.8
분산 서버	405.6	593.6

표 1은 단일 나라다 브로커링 스트리밍 서버를 사용한 경우와 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용한 경우의 클라이언트들의 평균 수신 bit rate 와 최대 수신 bit rate 값을 나타낸다.

단일 나라다 브로커링 스트리밍 서버를 사용한 경우의 평균 수신 bit rate는 145.7 Kbps로 측정되었다. 반면에 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용한 경우의 평균 수신 bit rate는 405.6 Kbps로 측정되어 단일 서버의 사용 보다 약 178%의 높은 수치를 보였다. 또한 단일 나라다 브로커링 스트리밍 서버를 사용한 경우의 최대 수신 bit rate는 164.8 Kbps이었고, 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용한 경우의 최대 수신 bit rate는 593.6 Kbps로 측정되어 단일 서버를 사용한 경우보다 약 260% 높은 수치를 나타내었다.

그림 9와 표 1에서 나타 난 것처럼, 분산된 나라다 브로커링 서버 네트워크를 사용하는 것이 단일 나라다 브로커링 스트리밍 서버를 사용하는 경우보다 더 높은 수신 bit rate를 제공 할 수 있음을 알 수 있다. 분산된 나라다 브로커링 스트리밍 서버 네트워크 사용함으로써 클라이언트들은 가장 적합한 나라다 브로커링 스트리밍 서버를 선택하여 독립적으로 스트리밍 서버와의 연결 설정을 할 수 있고 이는 병목 현상을 줄일 수 있다. 반면에, 하나의 나라다 브로커링 스트리밍 서버에 모든 클라이언트가 연결 되는 경우에는 병목 현상이 발생하여 많은 수의 UDP 패킷이 버려졌음을 알 수 있다. 이것은 부정확한 데이터의 전달을 야기 시킬 수 있으며 정확한 실험 결과의 분석에 있어서 문제가 될 수 있다.

따라서 클라이언트는 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용함으로써 데이터의 결손을

피하고 보다 정확한 데이터를 받아 볼 수 있다.

VI. 결 론

본 논문에서는 실시간 하이브리드 실험시설을 모델로 하여 원격지에서 실험을 수행하는데 있어서 수행된 실험 데이터를 저장하고 재연하는 그리드 컴퓨팅 기술 기반의 통합된 연구 시스템을 제안했다.

그리고, 높은 신뢰성과 성능을 제공하기 위하여, 본 논문에서는 분산된 나라다 브로커링 스트리밍 서버 네트워크를 사용하였고, KOCED 프로젝트의 데이터 특성에 고려하여 시뮬레이션을 통해 성능평가를 수행하였으며, 이를 통해 분산된 나라다 브로커링 스트리밍 서버 네트워크의 사용이 단일 서버의 사용보다 높은 신뢰성과 성능을 제공 할 수 있다는 것을 보였다.

따라서 본 논문에서 제안한 시스템이 리코딩 및 리플레이 서비스의 사용에 있어서 보다 높은 신뢰성과 성능을 이끌어 낼 수 있다고 판단된다. 그리고, 본 논문에서 제안한 분산 나라다 브로커링 스트리밍 서버의 네트워크는 시뮬레이션을 통해서만 측정된 부분이므로, 향후 실제 구현을 통한 성능 평가를 수행할 예정이다.

참고문헌

- [1] Foster, I., Kesselman, C., "The Grid 2: Blueprint for a New Computing Infrastructure", Morgan Kaufmann, 2004.
- [2] Network for Earthquake Engineering Simulation (NEES), <http://it.nees.org>
- [3] Korea Construction Engineering Development (KOCED), <http://www.koced.net>
- [4] Radic, B. Imamagic, E., Dobrenic, D., "Grid Data and Replica Management System", 27th International Conference on Information Technology Interfaces, pp. 541-546, 2005.
- [5] Foster, I., Kesselman, C., Nick, J., Tuecke, S., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", <http://www.globus.org/research/papers/ogsa.pdf>

- [6] Foster, I., Kesselman, C., "GLOBUS: A Metacomputing Infrastructure Toolkit", International Journal of Supercomputer Applications 11(2), pp.115-129, 1998.
- [7] GridFTP, <http://www.globus.org/toolkit/docs/4.0/data/gridftp>
- [8] Reliable File Transfer (RFT), <http://www.globus.org/toolkit/docs/4.0/data/rft>
- [9] The G-Civil Project, <http://www.soton.ac.uk/~gcivil>
- [10] Eisenhauer, G., Schwan, K., Bustamante, F., "Publish-Subscribe for High-Performance Computing", IEEE Internet Computing Magazine Vol. 10, No. 1, pp.40-47, 2006.
- [11] The Narada Brokering Project, <http://www.naradabrokering.org>
- [12] Nakashima, M. et al., "Integration Techniques for Substructure Pseudodynamic Test", Proc. of the 4th U.S. National Conf. on Earthquake Engineering, Vol. 2, pp.515-524, 1990.
- [13] Mini-MOST, <http://cive.seas.wustl.edu/wusceel/minimost>
- [14] OpenSees, <http://opensees.berkeley.edu>
- [15] Lee, J. et al., "A Grid-Based Research Environment for Civil Engineering", Proceedings of WSEAS International Conference on Computer Engineering and Applications, pp.100-105, 2007.
- [16] The Network Simulator: ns-2, <http://www.isi.edu/nsnam/ns>

저자소개

장 선(Sun Jang)



2005년 동국대학교 정보통신공학과
공학학사

2008년 홍익대학교 컴퓨터공학과
공학석사

2008년~현재 동양시스템즈 연구원

※관심분야: 그리드컴퓨팅, 분산 컴퓨팅

이 장 호(Jang Ho Lee)



1990년 서울대학교 컴퓨터공학과
공학학사

1992년 서울대학교 컴퓨터공학과
공학석사

2000년 University of Michigan, Electrical Eng. and
Computer Science 공학박사

2000년 IBM T.J.Watson Postdoctoral Researcher

2001년 (주)유비쿼스 수석연구원

2002년~현재 홍익대학교 컴퓨터공학과 조교수

※관심분야: CSCW, 그리드컴퓨팅, 모바일컴퓨팅