

# 그리드 기반의 토목공학 하이브리드 원격 실험의 리코딩 및 리플레이 서비스

장 선<sup>o</sup> 이장호  
홍익대학교 컴퓨터 공학과  
[realjangsun<sup>o</sup>@gmail.com](mailto:realjangsun@gmail.com), [janghol@cs.hongik.ac.kr](mailto:janghol@cs.hongik.ac.kr)

## Recording and Replay Service for a Grid-Based Hybrid Remote Experiment in Civil Engineering

Sun Jang<sup>o</sup> Jang Ho Lee  
Department of Computer Engineering, Hongik Univ.  
요 약

그리드 컴퓨팅 기술을 기반으로 한 원격 실험 환경 구축에 있어서 원격 실험만큼이나 실험결과 데이터를 저장하고 재연하는 것이 중요하게 대두되고 있다. 본 논문에서는 KOCED 프로젝트의 건설 연구 실험시설 중 하나인 실시간 하이브리드 다자유도 실험시설에 대한 프로토타입인 원격 하이브리드 실험에 대하여 나라다 브로커링 이라는 발간 및 구독 패러다임의 스트리밍 서버와 글로벌 톨킷에 기반 한 리코딩 및 리플레이 서비스를 통하여 실험결과 데이터를 저장하고 재연하는 시스템을 구축 하였다. 기존에 진행된 실험결과를 검색하여 볼 수 있게 함으로써 중복된 실험으로 인한 비용을 줄이고, 사용자가 원하는 데이터에 대한 토픽정보를 통하여 재연함으로써 실험결과 데이터의 효율성을 높일 수 있을 것으로 판단된다.

### 1. 서 론

그리드 컴퓨팅(Grid Computing)[1] 기술이 발전함에 따라 물리학, 천체학, 화학 연구, 건축, 토목 등 다양한 과학 연구 분야에서 기존의 실험 시설들을 그리드 컴퓨팅 기술과 접목시켜 활용 하고자 하는 노력들이 계속되고 있다.

국내외에서 그리드 컴퓨팅 기술을 사용한 e-Science[2] 환경을 구축하기 위한 노력들이 계속되고 있다. 대표적으로 16개의 지진공학 연구 실험 시설을 그리드 시스템으로 연결 한 미국의 NEES(Network for Earthquake Engineering Simulation)[3]와 NEES의 기본 개념을 모델로 국내의 건설 연구 실험시설들을 그리드 기술로 공유 하는 KOCED(Korea Construction Engineering Development)[4]가 있다. KOCED는 국내의 건설 연구 분야의 실험시설들을 그리드 시스템으로 공유하여 사용하는 프로젝트로서 본 논문에서는 KOCED의 실험 시설 중 하나인 실시간 하이브리드 다자유도 실험 시설을 대상으로 그리드 시스템을 구축 하였다.

이러한 그리드 기반의 실험 시설 공유 환경은 다양한 실험 시설의 구축에 많은 비용을 소모하고 실험 시설의 이용 또한 많은 비용이 필요하다. 따라서 최대한 중복된 실험을 피하여 비용을 절감하기 위해서는 실험의 결과 데이터를 저장하고 검색, 재연 하는 것이 원격으로 실험이 가능하게 하는 것만큼 이나 매우 중요 하다고 할 수 있다[5]. 또한 실험을 통해 발생하는 방대한 양의 데이터

들을 관리하고 데이터의 검색, 공유 등의 환경을 제공하려면 해당 데이터들은 반드시 저장되고 검색, 재연되어야 한다[6-7].

본 논문에서는 효율적으로 실험 결과를 저장하고 재연하기 위한 시스템을 설계 및 구현 하였다. 2장에서 관련 연구부분을 살펴보고, 3장에서는 시스템 구축의 대상 모델이 되는 실험시설의 설명과 원격실험에 대하여 알아보고 4장에서는 그리고 실험결과를 저장 및 재연하기 위한 그리드 서비스들의 구현과 구축에 대한 설명을 하고 5장에서 간단한 성능 평가를 살펴 본 후 마지막 6장에서 결론을 맺고 향후 연구 방안을 소개한다.

### 2. 관련 연구

그리드 컴퓨팅 기술에 기반 한 연구 환경에서는 다양하고 방대한 데이터들이 효과적으로 저장되고 관리 되어야 한다[8]. 현재 이와 관련된 연구가 다양하게 진행 되고 있으며 본 논문의 목적과 유사하게 데이터를 효율적으로 공유하고 관리하기 위한 시스템을 구축 해 놓은 대표적인 프로젝트로는 ESG(The Earth System Grid)[9]와 NEES를 예로 들 수 있다.

본 논문에서 실험 결과 데이터를 저장하고 재연하기 위해 사용된 기본적인 패러다임은 나라다 브로커링(Narada Brokering)[10-11]이 채택하고 있는 발간 및 구독(Publish and Subscribe) 패러다임[12] 이다. 발간 및 구독 패러다임이란 스트리밍 서버에 데이터를 전송해주는 공표자가 어떤 데이터를 전송하는가를 메타정보인 토픽으로 정의하여 전송하고 데이터를 스트리밍 받기를 원하는 구독자는 자신이 관심 있는 데이터에 대한 토픽 정보

본 논문은 2007학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음

를 이용하여 스트리밍 서버로부터 전송되는 다양한 데이터 중에서 원하는 데이터만 스트리밍 받는 클라이언트/서버 구조이다[13].

여러 가지 자원들을 공유하고 다양하고 방대한 양의 데이터들이 스트리밍 되는 그리드 컴퓨팅 기반의 공유 환경에 있어서 발간 및 구독 패러다임은 스트리밍 되는 다양한 데이터 중에서 사용자가 원하는 데이터를 토픽에 기반 하여 획득할 수 있게 함으로써 그리드 컴퓨팅 환경에서 적합한 스트리밍 서버라고 할 수 있다.

또한 본 논문에서 구현된 그리드 서비스들은 개방형 그리드 표준인 Open Grid Services Architecture (OGSA)[14]에 기반 한 글로벌 툴킷(Globus Toolkit) [15-17]을 사용하여 구축되었다. 그리드 컴퓨팅 기술이 발전하면서 다양한 표준화 작업이 진행 되었으며 그 중에서 가장 일반적이고 표준으로 자리 잡아가고 있는 것이 OGSA다. 글로벌 툴킷은 이러한 표준화 동향에 따라 OGSA에 기반 하여 그리드 컴퓨팅 환경 구축을 위한 핵심 서비스들을 그 목적에 따라서 다양한 컴포넌트 형태로 구현해 놓은 툴킷이다. 본 논문에서 저장된 데이터들을 사용자가 공유할 수 있는 환경으로 이동시키는데 사용된 주된 컴포넌트들에는 GridFTP[18]와 Reliable File Transfer (RFT)[19]가 있다.

GridFTP는 Global Grid Forum에 의해 정의된 파일 전송 프로토콜로서 인증된 사용자에게 빠르고 효과적이며 안전한 파일 전송 메커니즘을 제공해 준다. 또한 RFT는 대용량 데이터의 전송에서 GridFTP를 보완해 주기위한 컴포넌트로서 두 개의 GridFTP를 연결하여 원하는 데이터를 안전하게 이동시켜 준다. 데이터의 이동 상태를 데이터베이스를 사용하여 관리함으로써 데이터 전송 도중 발생 할 수 있는 네트워크 장애나 장비의 결함에 의한 데이터 손실을 효과 적으로 막아준다.

### 3. 원격 하이브리드 실험 시스템

본 논문에서 제안하는 시스템은 하이브리드 실험의 원격 실험을 비롯하여 실험 결과 데이터를 리코딩하고 리플레이하는 시스템 구축에 있다.

3.1절에서는 하이브리드 실험이란 무엇인가와 하이브리드 실험 중에서 본 논문에서 기반하고 있는 모델은 어떤 것인가에 대한 설명을 하고 3.2절에서는 이러한 하이브리드 실험을 원격지에서 수행하기 위하여 구성된 아키텍처를 살펴 볼 것이다.

#### 3.1 하이브리드 실험 모델

본 논문에서 원격 실험의 대상으로 하는 모델이 바로 하이브리드 실험 모델이다. 하이브리드 실험 모델이란 건설, 토목 공학 분야의 실험에 있어서 전체 실험 구조물을 여러 개의 부분 구조물로 나눈 뒤 일부 구조물들은 실제로 물리적으로 구축 하고 일부 구조물들은 컴퓨터에서 수치해석을 통하여 실험을 수행하는 실험으로서 물리적인 구조물과 수치해석 구조물의 결합으로 전체 실험이 수행되는 의사 동적 실험(Pseudodynamic test)에 기반

한 실험을 뜻한다[20].

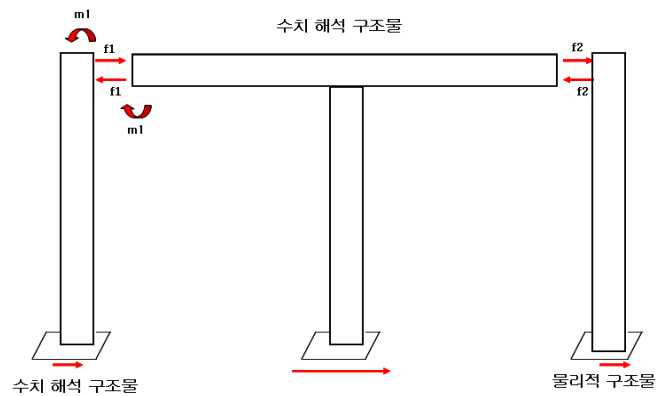


그림 1 하이브리드 실험 모델

본 논문에서는 KOCED 프로젝트의 실시간 다자유도 실험 시설의 프로토타입 형태인 그림 1과 같은 구조를 가지고 있는 Mini-MOST[21] 실험 구조를 모델로 하였다. 실제로 실험이 수행 될 때 물리적으로 힘을 받아 움직이는 구조물은 오른쪽 구조물이고 나머지 구조물들은 수치해석으로 대신하게 된다[22].

#### 3.2 프로토타입 구조물을 이용한 원격 실험

3.1절과 같은 모델을 원격으로 실험하는데 있어서 본 논문에서는 Open System for Earthquake Engineering Simulation (OpenSees)[23] 라는 수치해석 툴을 사용하였다.

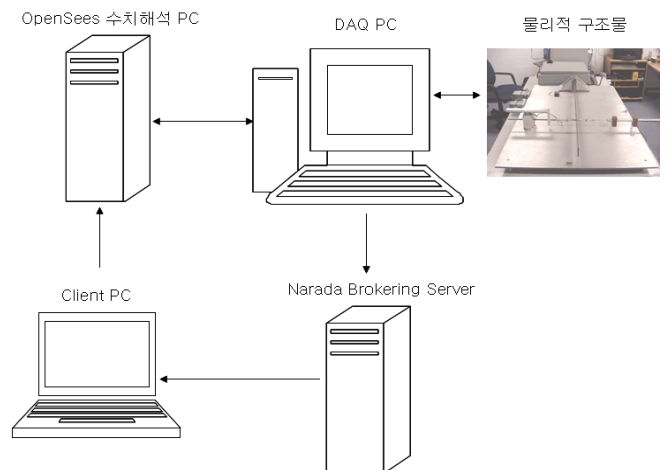


그림 2 원격 하이브리드 실험 구조도

그림 2는 원격 실험의 구조도를 나타내고 있다. 사용자가 클라이언트 프로그램을 통해 실험을 시작하길 원하는 명령을 OpenSees 수치해석 컴퓨터에 요청하면 OpenSees 수치해석 컴퓨터에서는 실제로 움직여야 할 물리적 구조물에 데이터획득 컴퓨터를 거쳐 명령을 전달하게 된다. 물리적 구조물이 명령에 따라 움직이게 되면 데이터획득 컴퓨터(Data Acquisition PC)가 센서를

통해 측정된 결과 값을 OpenSees 수치해석 컴퓨터와 나라도 브로커링 스트리밍 서버로 보내주게 된다. OpenSees 수치해석 컴퓨터는 받은 결과를 통해 수치해석을 하고 다음 명령을 결정하게 되며 클라이언트 프로그램은 나라도 브로커링 서버로부터 데이터를 받아 사용자에게 보여주게 된다.

그림 3은 실제 클라이언트 프로그램에서 원격 실험을 수행하면서 결과를 모니터링 하는 화면을 보여주고 있다.

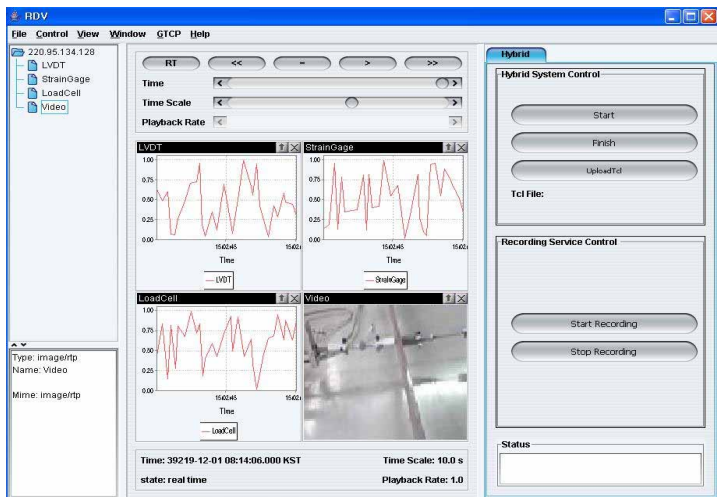


그림 3 원격 실험 및 원격 관찰

#### 4. 원격 하이브리드 실험 시스템의 리코딩 및 리플레이 서비스의 설계 및 구현

3.1절의 모델을 대상으로 3.2절과 같은 원격 실험을 진행 할 때 그 결과 데이터의 효율적인 저장 및 재연을 위하여 본 논문에서는 리코딩 서비스(Recording Service), 리플레이 서비스(Replay Service), HEMS(Hybrid Experiment Managing Service)를 구현하였고 글로비스 툴킷의 RFT 서비스와 GridFTP 서버를 이용하였다. 그림 4는 리코딩 및 리플레이 서비스를 위한 전체 시스템의 연관 관계를 나타내고 있다.

리코딩 및 리플레이 서버는 리코딩, 리플레이 서비스가 서비스 되는 머신으로서 클라이언트나 HEMS 서버로부터 서비스 요청을 받아 나라도 브로커링 서버로부터 데이터를 구독해서 데이터를 저장하거나 포털 서버로부터 저장된 데이터를 읽어 와서 나라도 브로커링 서버에 데이터를 발간하는 역할을 한다.

나라도 브로커링은 클라이언트/서버 구조의 스트리밍 서버로서 데이터 획득 컴퓨터(Data Acquisition PC)나 리코딩 및 리플레이 서버 내부의 리플레이 서비스에 의해 발간되는 데이터들을 스트리밍 해 주는 역할을 한다.

데이터 획득 컴퓨터는 물리적 구조물에 부착되어있는 센서로부터 읽은 데이터를 나라도 브로커링 서버로 옮겨주는 역할을 한다.

HEMS는 Hybrid Experiment Managing Service로서 클라이언트로부터 받은 리코딩 명령에 따라 리코딩 및

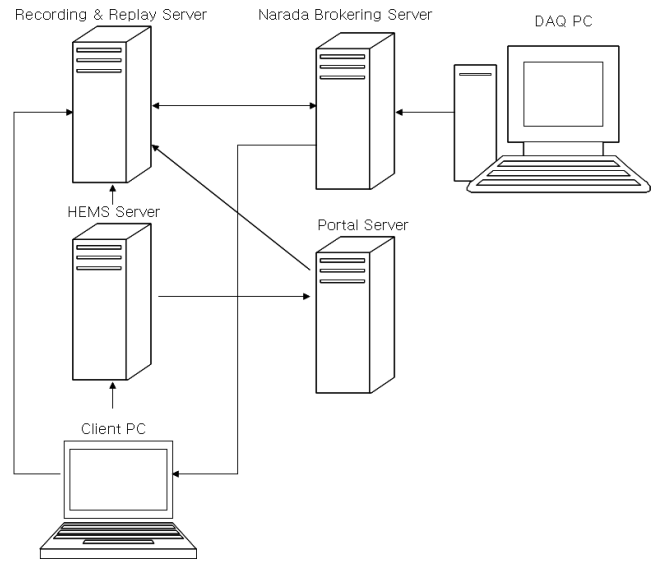


그림 4 원격 하이브리드 실험 시스템의 리코딩 및

리플레이 서비스 시스템 연관도  
리플레이 서버 내부의 리코딩 서비스를 호출하고 RFT와 GridFTP를 통하여 리코딩 서비스가 저장해 놓은 실험결과 데이터를 포털 서버의 저장소로 옮겨주는 역할을 한다.

포털 서버는 사용자 공유 환경인 웹 포털이 제공되는 웹 서버로서 클라이언트 프로그램이 이곳에서부터 다운로드 받아 자바 웹 스타트 형식으로 시작되고 리코딩 서비스가 저장 하는 실험결과 데이터가 최종적으로 위치하게 되는 곳이다.

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

#### 4.1 리코딩 서비스

그림 5는 리코딩 서비스를 서비스 하기위한 소프트웨어들의 구조를 나타내고 있다.

실험결과 데이터의 저장 순서를 살펴보면 제일 처음 사용자가 실험결과 데이터의 저장 시작 요청을 클라이언트 프로그램에 내리게 된다.

명령을 받은 클라이언트 프로그램은 현재 모니터링하고 있는 데이터들의 토픽 정보를 포함하여 HEMS에 데이터 저장 시작 요청을 전달하게 된다.

HEMS는 클라이언트 프로그램으로부터 받은 토픽 정보와 함께 리코딩 서비스를 호출하여 데이터 저장의 시작을 요청하게 된다.

리코딩 서비스는 HEMS로부터 전달받은 토픽 정보에 따라서 저장하기를 원하는 데이터를 나라도 브로커링에서 구독해 온다.

리코딩 서비스는 구독해온 데이터를 로컬 저장소에 임시 저장하게 된다.

사용자로부터 데이터 저장의 중지 요청이 들어올 때 까

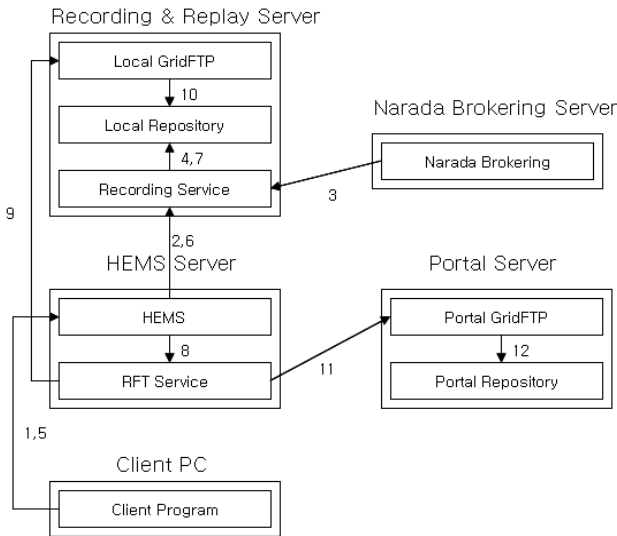


그림 5 원격 하이브리드 실험 시스템의 리코딩

서비스 소프트웨어 구조도  
리코딩 서비스는 해당 토픽에 대한 데이터들을 나라다 브로커링으로부터 스트리밍 받아 주기적으로 로컬 저장소에 저장한다.

실험결과 데이터를 저장 하던 중에 사용자로부터 데이터의 저장을 중지하는 요청이 들어오게 되면 클라이언트 프로그램은 이를 HEMS로 전달하게 된다.

HEMS는 리코딩 서비스에게 데이터 저장 중지 요청을 내려 데이터의 저장을 멈추게 되고 RFT서비스를 호출한다.

RFT서비스는 리코딩 서비스가 로컬 저장소에 임시로 저장해 놓은 데이터를 최종적인 사용자 공유 환경인 포털서버 쪽의 포털 저장소 로 옮기게 된다. 이때 RFT는 로컬 GridFTP를 통해 로컬 저장소에 접근하고 포털 GridFTP를 통해 포털 저장소에 접근하여 안전하고 효율적인 데이터 전송을 담당하게 된다.

#### 4.2 리플레이 서비스

그림 6은 리플레이 서비스를 서비스 하기위한 소프트웨어들의 구조를 나타내고 있다.

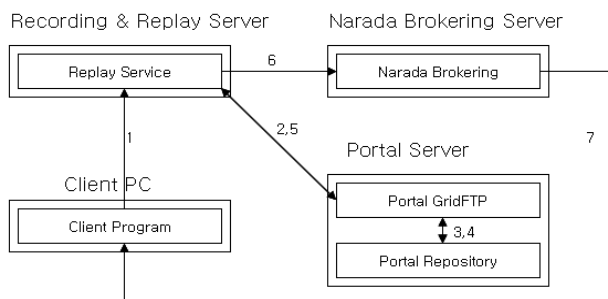


그림 6 원격 하이브리드 실험 시스템의 리플레이 서비스 소프트웨어 구조도

사용자는 이미 진행 되었던 실험의 결과 데이터를 보고 싶을 경우 아래와 같은 순서에 의해 해당 데이터를 관측할 수 있다.

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

리플레이 서비스는 포털 GridFTP에 접속하여 포털 저장소에 저장되어 있는 데이터들을 읽어 온 후 나라다 브로커링에 해당 데이터들을 순차적으로 발간하게 되고 클라이언트 프로그램 에서는 나라다 브로커링에 접속하여 스트리밍 되는 데이터들을 관찰 할 수 있다.

그림 7은 실제로 클라이언트 프로그램에서 데이터를 관찰하는 모습을 보여준다.

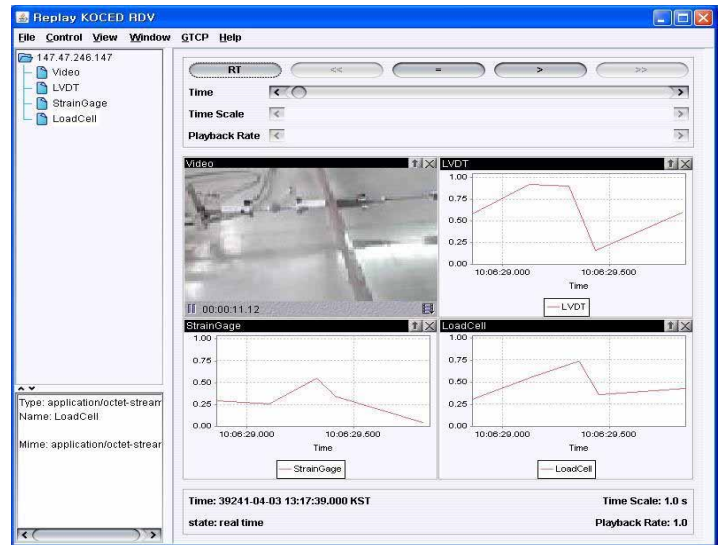


그림 7 원격 하이브리드 실험 시스템의 리플레이

#### 5. 성능 평가

리플레이 서비스를 사용하여 실험 결과 데이터를 재연하여 사용자들이 공유 하는 환경에 대한 성능평가를 수행하였다. 실험을 수행한 서버 환경으로는 P4-3.0Ghz의 CPU와 2GB의 메모리, 10MB의 랜카드를 사용하였다.

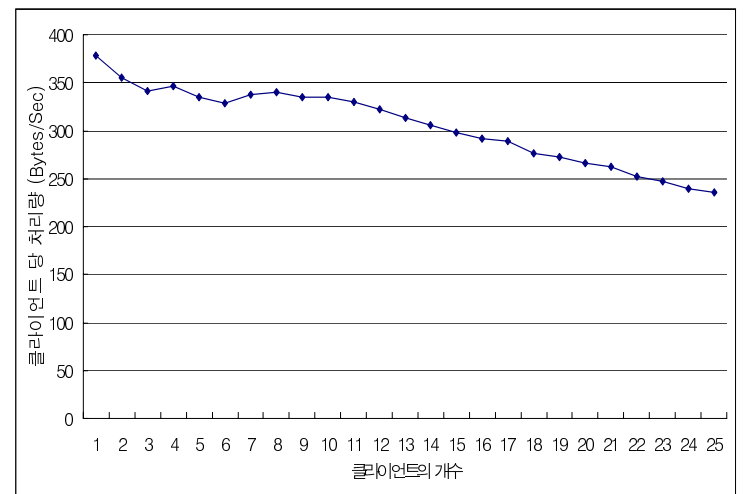


그림 8 클라이언트 개수 변화에 따른 나라다 브로커링 스트리밍 서버의 클라이언트 당 처리량



그림 8은 나라다 브로커링 스트리밍 서버에서 측정된 클라이언트 당 스트리밍 처리량을 나타내 주는 그림이다. x축은 클라이언트의 개수를 나타내고 있고 y축은 클라이언트 하나당 서버의 처리량을 나타내고 있다. 그림 8에서 볼 수 있듯이 클라이언트 개수가 늘어남에 따라 한 클라이언트 당 서버에서 처리해 주는 처리량이 감소하는 것을 알 수 있다. 10개의 클라이언트를 동시에 처리 할 때의 처리량은 하나의 클라이언트를 처리 할 때의 약 89퍼센트였으며 15개의 클라이언트를 동시에 처리 할 때에는 약 79퍼센트, 20개의 클라이언트를 동시에 처리 할 때에는 약 70퍼센트, 25개의 클라이언트를 동시에 처리 할 때에는 약 62퍼센트의 처리량을 보였다. 따라서 하나의 나라다 브로커링 서버로 사용자가 큰 성능 저하를 느끼지 않고 원활하게 서비스를 이용 할 수 있게 하기 위해서는 약 15개 정도의 클라이언트가 연결되어 서비스를 사용하는 것이 적당한 한계 범위라고 판단된다.

## 6. 결 론

본 논문에서는 KOCED 프로젝트의 실험 시설 중 하나인 원격 하이브리드 실험을 모델로 하여 발간 및 구독 패러다임의 나라다 브로커링 스트리밍 서버와 그리드 서비스를 보다 쉽게 구축 할 수 있는 글로벌 툴킷에 기반 하여 실험 결과 데이터의 저장과 재연을 가능하게 해주는 리코딩 및 리플레이 서비스를 구현 하였고 그리드 기반의 실험 환경에서의 실험 데이터 저장 및 재연을 위한 시스템을 설계 및 구축하였다.

본 논문에서 설계 및 구축한 시스템이 실제 시스템에서 사용되기 위해서는 다양한 지역에서 접속하는 사용자들에게 최소한의 지연 시간을 제공하기 위하여 하나의 나라다 브로커링 서버가 아니라 분산된 나라다 브로커링 서버의 네트워크가 형성되어야 할 것이다.

또한 현재 리플레이 서비스의 경우 포털의 GridFTP에서 결과 데이터를 가져오면서 나라다 브로커링에 데이터를 발간하기 때문에 사용자가 원하는 부분으로 이동 시 버퍼링 과 동기화 때문에 지연 시간이 발생하는데 이는 향후 개선되어야 할 부분이다.

본 논문에서 설계 및 구축한 시스템을 통해 기존의 원격 실험 시스템에서는 제공되지 않았던 기존에 진행되었던 실험의 결과 데이터를 검색을 통해 볼 수 있게 할 수 있다. 따라서 중복된 실험의 가능성을 줄일 수 있고 실험 수행에 수반되는 비용절감의 효과도 기대 할 수 있다. 이러한 데이터 공유 환경 시스템의 구축으로 인하여 실험결과 데이터 사용의 효율성을 높일 수 있을 것으로 판단된다.

## 7. 참고 문헌

[1] Foster, I., Kesselman, C., Tuecke, S., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," , Int'l J. High-Performance Computing Applications, vol. 15, pp. 200-222, 2001  
 [2] Reading e-Science Centre, <http://www.resc.rdg.ac.uk>  
 [3] The NEEsgrid System Integratioin Team,

"Introduction to NEEsgrid : TR2004-08-23", <http://it.nees.org> , August, 2004.

[4] Korea Construction Engineering Development, <http://www.koced.net/>

[5] 신수봉, 강수용, 김철영, 염현영, 김재관, "KOCED: 건설분야의 그리드 기술 활용", 한국정보과학회지 , 1229-6821 , 제24권5호 , pp. 46-52 , 2006

[6] Radic, B. Imamagic, E. Dobrenic, D. "Grid data and replica management system", 27th International Conference on Information Technology Interfaces, pp. 541-546, 2005

[7] R. Moore, A. Rajasekar, and M. Wan, "Data Grids, Digital Libraries and Persistent Archives: An Integrated Approach to Publishing, Sharing and Archiving Data", Special Issue of the Proceedings of the IEEE on Grid Computing, IEEE Computer Society, Vol. 93, No.3, pp. 578-588, March 2005.

[8] Malcolm Atkinson, Ann L. Chervenak, Peter Kunszt, Inderpal Narang, Norman W. Paton, Dave Pearson, Arie Shoshani, and Paul Watson, "Data Access, Integration, and Management" , The GRID 2 : Blueprint for a New Computing Infrastructure, pp 391-430, 2003

[9] Foster, I., Aplper, E., Chervenak, A., Drach, B., Kesselman, C., Nefedova, V., Middleton, D., Shoshani, A., Sim, A., and Williams, D., "The Earth System Grid II: Turning climate datasets into community resources", Annual Meeting of the American Meteorological Society, 2002

[10] Shrideep Pallickara, and Geoffrey Fox, "NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids." Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware-2003, pp 41-61, <http://www.naradabrokering.org> 2003

[11] Narada Brokering, <http://www.naradabrokering.org/>

[12] Eisenhauer, G. Schwan, and K. Bustamante, F.E, "Publish-subscribe for high-performance computing", IEEE Internet Computing Magazine Volume: 10 Issue: 1, pp.40-47, 2006

[13] EUGSTER, P., GUERRAOUI, R., and SVENTEK, J. "Distributed Asynchronous Collections: Abstractions for publish/subscribe interaction" , Proceedings of the 14th European Conference on Object-Oriented Programming (ECOOP'2000), 2000.

[14] Foster, I., Kesselman, C., Nick, J. M., Tuecke, S., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", <http://www.globus.org/research/papers/ogsa.pdf> , 2002.

[15] Brunett, S., Czajkowski, K., Fitzgerald, S., Foster, I., Johnson, A., Kesselman, C., Leigh, J., Tuecke, S., "Application experiences with the Globus Toolkit", in 7th IEEE International Symposium on High

Performance Distributed Computing, pp. 81-89, 1998.

[16] Foster, I., Kesselman, C., "GLOBUS: A metacomputing infrastructure toolkit", International Journal of Supercomputer Applications 11(2), pp.115-129, 1998.

[17] Sotomayor, B., Childers, L., "Globus Toolkit 4", pp.3-39, MORGAN KAUFMANN, 2006.

[18] GridFTP, <http://www.globus.org/toolkit/docs/4.0/data/gridftp/>

[19] Reliable File Transfer (RFT), <http://www.globus.org/toolkit/docs/4.0/data/rft/>

[20] M. Nakashima et al., "Integration Techniques for Substructure Pseudodynamic Test", Proceedings of the 4th U.S. National Conference on Earthquake Engineering, vol. 2, pp.515-524, Palm Springs, CA, 1990.

[21] Mini-MOST, <http://cive.seas.wustl.edu/wusceel/minimost>

[22] Jang Ho Lee, Dong Wook Kim, Sun Jang, Taikyeong Jeong, Song-Yi Yi, "A Grid-Based Research Environment for Civil Engineering", Proceedings of the 2007 WSEAS International Conference on Computer Engineering and Applications, pp.100-105, 2007.

[23] OpenSees, <http://opensees.berkeley.edu>