Korea@Home 데스크톱 그리드 시스템에서 REST 웹 서비스 구현†

한창환°[1], 한연희[1], 길준민[2], 강상원[3], 최장원[4]
[1]한국기술교육대학교 인터넷미디어공학부
{atin84, yhhan}@kut.ac.kr
[2]대구가톨릭대학교 컴퓨터교육과
jmgil@cu.ac.kr
[3]고려대학교 컴퓨터학과
swkang@disys.korea.ac.kr
[4]한국과학기술정보연구원 고성능네트워크사업단
jwchoi@kisti.re.kr

Implementation of REST Web Service in Korea@Home Desktop Grid System

Chang-Hwan Han°[1], Youn-Hee Han[1], Joon-Min Gil[2], Sang-Won Kang[3], Jang-Won Choi[4]
[1]Korea University of Technology and Education, School of Internet-Media
[2]Catholic University of Daegu, Dept. of Computer Science Education
[3]Korea University, Dept. of Computer Science and Engineering
[4]Korea Institute Science and Technology Information, HPcN Project Division

요 약

기존의 P2P기반의 분산컴퓨팅 시스템을 일반 응용수행자가 사용하기 위해서는 수동적으로 시스템에 위탁하여 수행하여 왔다. 이러한 구조에서 응용수행자가 자신이 원하는 응용을 계산도구로서 분산컴퓨팅 시스템을 활용하여 수행하고 최종 결과를 받아오는 것은 복잡하고 어려운 과정이었다. 본 논문에서는 이러한 과정을 간편화하고 분산컴퓨팅 시스템의 참여도를 높이기 위하여 웹 서비스 Open API를 사용한 개방형 분산컴퓨팅 시스템을 구현하였다. 본 논문은 국내의 P2P기반 분산컴퓨팅 시스템인 Korea@Home 데스크톱 그리드 시스템을 소개하고 이 시스템에서 효율적으로 Open API를 제공하기 위하여 REST 웹 서비스를 구현하고 응용수행자의 편의를 위한 인터페이스로 자바 클래스를 제공하였다. 응용수행자는 제공되는 인터페이스를 통하여 간단한 프로그래밍 지식으로 Korea@Home 데스크톱 그리드 시스템을 웹 서비스 방식으로 이용할 수 있다.

1. 서 론

P2P기반 분산컴퓨팅 시스템(혹은 데스크톱 그리드 시스템)은 기존의 슈퍼컴퓨터에서 수행하던 응용을 대체할 수 있는 수단으로 최근에 활발한 연구가 진행되고 있다[1-4]. P2P기반 분산컴퓨팅은 다양한 분야에 활용될 수 있는데, 대표적 적용 응용으로는 바이오 분야의 유전체 서열비교, 나노 시뮬레이션, 반도체, PDP 등의 전자부품 시뮬레이션, 기상 및 기후 시뮬레이션, 영상 애니메이션, 항공우주 비행체 및 자동차의 설계 해석 등을 예로서 들 수 있다.

1990년 후반부터 SETI@Home의 성공적인 구축과 활용에 힘입어 국내외적으로 많은 분산컴퓨팅 프로젝트들이 시작되었다. 한편, 국내에서도 2003년부터 P2P기반 분산컴퓨팅 시스템인 Korea@Home 프로젝트가 시작되었고, 현재까지 바이오분야의 신약후보물질탐색, 유전체 서열비교, IT 분야의 TFT-LCD 광특성 해석, 기상분야의 한반도 기후예측 등의 다양한 응용을 수행해 왔다[1-5]. 다양한 응용 적용이 가능하고 일반 인터넷연결 PC를 컴퓨팅 자원으로 활용할 정도로 Korea@Home 시스템이 안정화되고 신뢰성 있게 구축·운영되고 있음에도 불구하고, 현재까지 일반 응용수행자가 Korea@Home 시스템을 계산도구로 활용하기 위해서는 별도의 표준 API를 사용하지 않고 자신의 응용을 수동적으로 시스템에 위탁하여 응용을 수행하여 왔다. 이러한 점은 응용수행자가 자신의 응용을 위탁하여 분산컴퓨팅에서 확보한 유휴자원

†본 연구는 한국과학기술정보연구원 “컴퓨터연계활용기반 구축(Korea@Home)" 사업의 지원에 의해서 수행되었음.

을 활용하는 과정과 최종결과를 분산컴퓨팅 시스템으로부터 다시 받아오기까지의 일련의 과정이 일반 응용수행자 입장에서는 매우 복잡하고 어려운 과정을 거치는 것이다. 따라서 응용수행자가 복잡하고 어려운 과정 없이 능동적으로 분산컴퓨팅시스템에 참여할 수 있고 응용결과를 활용할 수 있는 체계적인 방안이 요구된다.

본 논문에서는 사용자 생산 컴퓨팅(User-Created Computing) 환경을 응용수행자에게 제공하는 새로운 기능을 갖는 분산컴퓨팅 시스템을 구현한다. 본 논문의 구현 시스템은 응용수용자가 자신이 필요로 할 때 언제든지 자신의 응용을 분산컴퓨팅 시스템을 활용하여 수행할 수 있는 웹 서비스 Open API를 제공한다. 이러한 웹 서비스 Open API를 활용하면 분산컴퓨팅 시스템의 관점에서는 응용수행의 항시성을 보장해 주며 유휴자원의 활용관점에서는 확보된 유휴자원을 최대한 활용할 수 있다는 장점이 있다. 또한, 응용수행자 입장에서는 응용의 위탁부터 결과의 활용까지 단순하고 단일화된 Open API 방식으로 처리할 수 있어서 현재보다 많은 응용의 참여가 예상되므로 결과적으로 응용결과의 활용도 증대를 통한 P2P기반 분산컴퓨팅의 응용 활성화에 기여할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대하여 살펴보고, 3장에서는 본 논문에서 구현하는 시스템의 개요를 살펴본다. 4장에서는 REST 웹서비스 구현에 대해서 자세히 기술하며, 구현결과는 5장에서 보여준다. 마지막으로 6장에서 본 논문의 결론을 맺는다.

2. 관련연구

2.1 Korea@Home 데스크톱 그리드 시스템

2003년부터 한국과학기술정보연구원에서 개발·구축·운영 중인 Korea@Home 데스크톱 그리드 시스템은 인터넷에 연결되어 있는 수많은 컴퓨팅 자원을 이용하여 대량의 연산을 요구하는 정보를 분산 처리하여 그 결과를 얻을 수 있는 시스템이다. Korea@Home 시스템은 우리나라의 초고속 정보통신 인프라를 최대로 활용하여 저비용, 고효율의 그리드 컴퓨팅 환경을 구축하고자 하는 목적을 가지고 있다. 만약 이 시스템에 3만명의 PC 자원제공자(P-III 500MHz 기준)가 동시에 참여할 경우, 이론적으로 ASCI White 슈퍼컴퓨터에 버금가는 성능인 12.7 테라플롭스 정도의 성능을 얻을 수 있다[5].

Korea@Home 시스템에 참여하는 자원제공자는 일반 네티즌이다. 현재는 대다수 사람들이 호응할 수 있는 공공적인 성격의 애플리케이션을 발굴하고 기반 소프트웨어를 개발하여 관련 산업의 활성화를 유도하기 위하여 많은 노력을 기울이고 있다.

2.2 웹 서비스 Open API와 REST

웹 서비스 Open API는 원격에 위치한 서버 내의 유용한 자원을 서버 외부에서 쉽게 사용할 수 있게 해주는 웹 기반 서비스이다[6]. 소프트웨어를 구성할 때 웹 서비스 Open API를 통하여 자신이 가지고 있지 못한 정보나 서비스 등을 다른 서버로 요구하여 그 결과를 가져와 활용이 가능하다. 현재 Google, Amazon, eBay 등을 비롯하여 국내의 네이버, 다음 등의 포탈사이트에서 독자적인 웹 서비스 Open API를 제공하고 있다.

웹 서비스 Open API의 기반 구조로 활용될 수 있는 REST(Representational State Transfer)는 대규모 네트워크 시스템을 위한 아키텍처로서 2000년 Roy Fielding의 박사 학위 논문에서 처음 제안되었다[7]. REST 아키텍처는 웹상의 모든 것들은 URL로 표현하며 잘 정의된 URL을 통해 웹 애플리케이션을 구동하고 그 결과를 전달 받아 처리하는 것에 중점을 두고 있다. REST는 웹에 존재하는 리소스의 상태가 핵심이며 주고받는 메시지의 형식이 자유롭다. 같은 리소스도 POX(Plain Old XML), JSON(JavaScript Object Notation), RSS 및 ATOM 등으로 다양하게 표현할 수 있다. 하나의 리소스에 대해 CRUD(Create, Retrieve, Update, Delete - 만들고, 읽고, 고치고, 지우는 처리)를 HTTP의 메소드인 POST, GET, PUT, DELETE와 대응하여 리소스의 URL로 리소스를 요청한다[8]. 본 논문에서 제안하는 시스템에서는 웹 서비스 Open API로서 REST 아키텍처를 사용하였다.

3. 시스템 개요

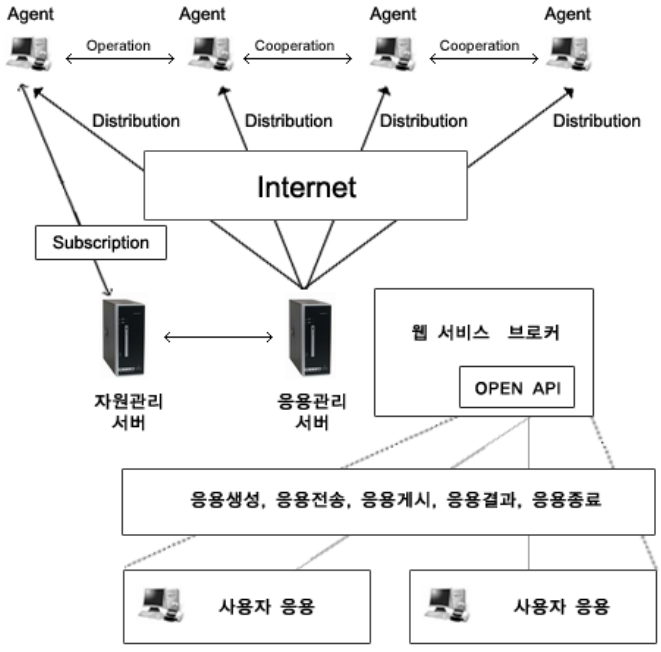

그림 1. 웹 서비스 Open API를 활용한 분산컴퓨팅 모델

본 논문에서는 웹 서비스 Open API를 활용하여 응용수행자가 Korea@Home 시스템에 자신의 응용을 위탁하고 결과를 활용할 수 있는 환경을 제공하는 것을 목표로 한다. 기존에 Korea@Home의 응용수행자는 자신의 응용을 Korea@Home 시스템에 수행하기 위해서 복잡한 과정을 거쳐야 했다. 본 논문에서 제공되는 웹 서비스 Open API를 통하여 프로그래밍을 할 수 있는 기본적인 능력을 가진 응용수행자는 웹 서비스 API 및 라이브러리

를 활용하여 쉽게 응용을 생성하고 응용의 상태확인 및 결과를 다운받을 수 있다. 그림 1은 본 논문에서 구현한 웹 서비스 Open API를 활용한 분산컴퓨팅 모델을 간략히 보여준다. 본 논문에서 구현된 웹 서비스의 주요기능은 다음과 같다.

○ 응용수행자 인증 : 웹 서비스를 이용하기 위한 수행자의 인증을 처리한다.
○ 응용생성 및 응용관련 파일 전송 : 응용생성을 위해 필요한 파라미터와 파일들을 전송받고 응용을 생성한다.
○ 응용상태 확인 : 응용의 실행상태, 처리된 수, 수행시간 등에 대하여 확인한다.
○ 응용의 실행 또는 중지 : 응용을 실행시키거나 중지시킨다.
○ 응용결과 다운로드 : 작업이 완료된 응용의 결과를 다운받는다.

허가된 응용수행자만이 웹 서비스를 사용할 수 있도록 반드시 인증 과정을 거친다. Stateless 방식으로 웹 서비스를 구현하기 위하여 인증은 Request를 보낼 때마다 이루어지며 인증에 성공하지 못하는 Request는 처리하지 않는다. 응용의 상태를 확인하는 URL에서는 해당 데이터베이스의 값을 가져와서 XML 문서 형태로 응답을 보내준다. 응용수행자는 프로그램에서 XML을 파싱해서 사용할 수 있으며 웹브라우저를 통한 URL 접근을 통하여 값을 확인할 수도 있다.

응용수행자는 구현된 REST 웹 서비스에 URL 접근을 통한 응용의 생성 및 수행에 관련된 모든 서비스를 받을 수 있다. 하지만 REST 방식을 잘 모르는 일반 응용 수행자들에게 편리성을 제공하기 위해서 수행자 인터페이스로서 KathWebService라는 자바 클래스도 제공한다. 따라서 응용수행자는 웹 서비스와 XML에 대한 지식이 없을지라도 KathWebService 클래스를 이용하여 객체를 생성하고 메소드를 통하여 REST 방식의 웹 서비스를 쉽게 이용할 수 있다. 그림 2는 응용수행자가 응용수행자에게 제공된 인터페이스인 KathWebService를 이용하여 웹 서비스를 이용하는 구조를 보여준다.

4. REST 웹 서비스 구현

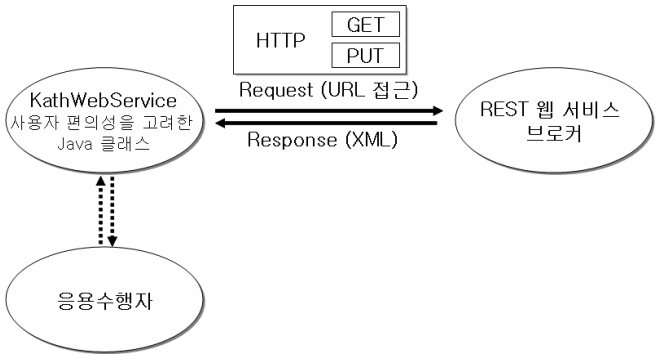

그림 2. 구현 시스템 구조

4.1 Restlet 생성

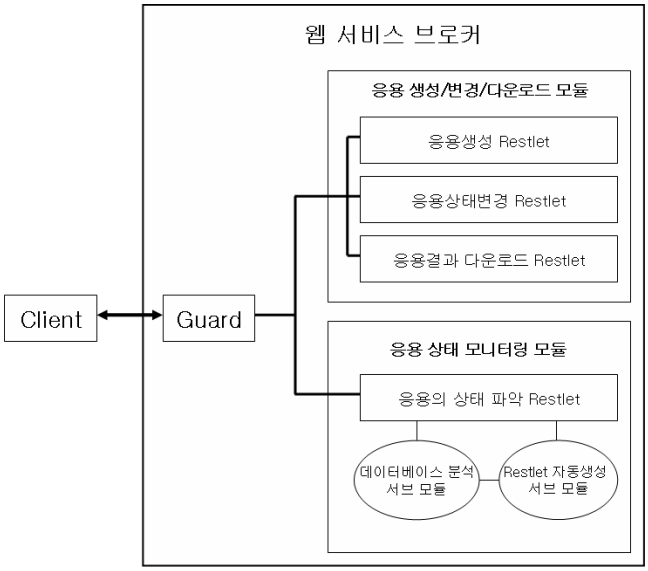

그림 3. 웹 서비스 브로커 구조

본 논문에서는 REST 웹 서비스 구현을 위하여 자바 기반의 REST 웹 서비스를 제공하는 Restlet[9]을 이용하였다. 그림 3은 본 논문의 분산컴퓨팅 시스템에서 구현된 웹 서비스 브로커의 구조를 보여준다. 그림 3의 웹 서비스 브로커 구조에서 웹 서비스 브로커로 들어오는 모든 Request는 Guard를 통하여 응용 수행자 인증을 하게 된다. 여기서 Guard의 역할은 인증되지 않은 Request를 접속할 수 없게 해준다. 만일 응용 수행자 인증에 실패한다면 인증되지 않은 응용 수행자로 판별하고 Bad Request로 처리한다.

또한 Guard는 웹 서비스 브로커에 접속하는 URL을 분석하여 알맞은 모듈로 분배해 주는 역할을 한다. 그림 3에서 볼 수 있듯이, 웹 서비스 브로커 내에 구현된 모듈은 크게 “응용 생성/변경/다운로드” 모듈과 “응용 상태 모니터링 모듈”로 나누어 구현되어 있다. “응용 생성/변경/다운로드 모듈”은 표 3에서 제시하는 응용의 생

표 1. DB 내용을 URL 접근을 통해 얻는 방법과 예제

http://서버명/웹 서비스명/DB명/테이블명/주요키값/필드명	
예1	http://grid.cu.ac.kr:8888/ws/kathDB/App_group/1/state/
	Korea@Home 데이터베이스에 접속하여 1번째 응용의 상태를 받는다.
예2	http://grid.cu.ac.kr:8888/ws/kathDB/VirtualScreen_work
	Korea@Home 데이터베이스에 접속하여 VirtualScreen응용의 작업리스트를 받는다.
예3	http://grid.cu.ac.kr:8888/ws/kathDB/VirtualScreen_work/1/state
	Korea@Home 데이터베이스에 접속하여 VirtualScreen응용의 1번째 작업의 상태를 받는다.

성, 응용의 상태 변경, 응용의 수행결과를 다운로드하는 역할을 수행한다. Interface Application에 포함된 Restlet들은 표 3에서 보이는 응용의 상태파악을 제외한 인터페이스들의 수와 동일하다. 한편, “응용 상태 모니터링 모듈”은 표 3에서 응용의 상태 파악을 위한 인터페이스를 구현하기 위해 만들어졌다.

응용의 상태 파악은 Korea@Home 데이터베이스 내의 응용작업 정보를 추출 및 해석하여 이루어진다. 그래서 응용 상태 모니터링 모듈은 Korea@Home의 데이터베이스의 내용을 확인할 수 있는 Restlet들을 제공한다. Restlet은 Korea@Home 데이터베이스 내의 테이블 목록과 테이블 내용을 모두 확인할 수 있도록 생성된다.

“응용 상태 모니터링 모듈”을 통해서 클라이언트는 Korea@Home 데이터베이스의 내용을 표 1에 제시된 바와 같이 URL 접속을 통하여 확인할 수 있다.

이와 같은 URL 접속 시에도 응용 수행자의 인증을 요구하며 인증에 성공하면 해당 데이터베이스의 내용을 XML 문서로 확인할 수 있다.

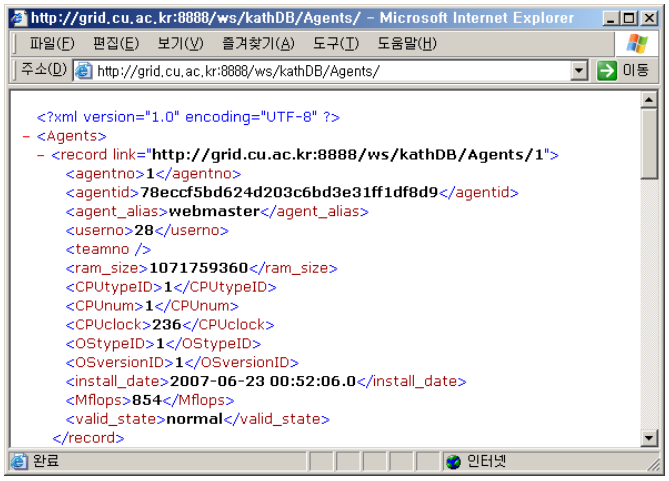

그림 4. URL 접근을 통한 데이터베이스 접근의 예

그림 4는 상태 모니터링 Restlet에 URL로 접속하는 과정을 웹브라우저를 통하여 테스트한 화면을 보여준다.

이미 구축되어 있는 Koera@Home 데이터베이스에 REST 방식으로 웹 기반 Open API 서비스를 제공하기 위하여 “응용 상태 모니터링 모듈” 내의 “데이터베이스 분석 서브 모듈”은 Koera@Home 데이터베이스 내에 존재하는 데이터베이스 단위, 테이블 단위, 레코드 단위, 필드 단위의 각 정보를 분석한다. 이러한 분석 이후, “Restlet 자동 생성 서브 모듈”은 REST 방식의 접근 URL 생성 및 관련 Restlet 생성을 한다.

표 2. 생성된 Restlet의 수

테이블 목록 전송 Restlet	1개
각 테이블내의 레코드 목록 전송 Restlet	83개
특정 레코드 전송 Restlet	83개
레코드 내의 특정 필드 전송 Restlet	741개

표 2는 이와 같은 과정을 통하여 생성된 Restlet 종류 및 개수를 보여준다. 테스트를 통해 얻은 Restlet의 수는 데이터베이스에 쌓인 각각의 정보의 수와 동일하다.

4.2 응용 수행자 편의성을 위한 인터페이스 제공

표 3은 응용수행자를 위한 인터페이스로서 제공되는 KathWebService 클래스의 메소드들을 보여준다. 첫 번째로 응용생성 API는 응용을 최초시작을 위해 사용된다. 응용수행자는 appAuthenticate를 반드시 제일 처음 호출하여 인증을 받아야 한다. 그렇지 않으면 다른 모든 메소드들도 사용할 수 없다. appCreate는 Korea@Home 데이터베이스에 응용정보를 등록시키는 역할을 수행하며, programTransfer와 projectTransfer를 통하여 작업수행을 위한 작업프로그램과 작업데이터 파일을 서버로 전송시킨다. 마지막으로 appReadyComplete를 이용하여 응용관련 데이터베이스를 자동으로 생성하고 응용을 Korea@Home에서 정상적으로 수행하게 한다.

표 3. 응용수행자에게 제공하는 인터페이스

인터페이스	기능설명	비고
appAuthenticate	응용수행자의 인증	응용의 생성 API
appCreate	응용의 생성 및 이름 설정	
programTransfer	응용프로그램 및 응용데이터 전송	
projectTransfer	응용입력파일 전송	
appReadyComplete	응용의 준비 완료 (응용 관련 테이블 자동 생성)	
appEnd	응용의 종료를 보고받음	응용의 상태 파악 API
appStart	응용의 시작을 알림	
appStatus	응용의 상태를 파악	
getCompleteWorkList	응용의 완료된 작업리스트 얻어옴	
getCompleteWorkTime	응용의 완료작업의 총 수행시간을 얻어옴	
getCompleteWorkUnit	응용의 완료 작업개수 얻어옴	
getTotalWorkList	응용의 작업리스트 얻어옴	
getTotalWorkUnit	응용의 총 작업개수 얻어옴	
isAppEnd	응용의 종료상태를 알아봄	
isAppStart	응용의 시작상태를 알아봄	
appRestart	멈춰진 응용을 재개시킴	응용의 상태 변경 API
appPause	응용의 수행을 잠시 멈춤	
appResultDownload	응용결과를 다운로드	응용결과 다운로드 API

그림 5. 용용수행자에게 제공되는 인터페이스를 통해 구현한 새로운 응용을 생성하는 코드

```
public class client {
    public static void main(String[] args) throws IOException{
        // 응용 객체 생성
        KathWebService app = new KathWebService();

        // 응용 수행자 인증
        app.appAuthenticate("id", "pass");

        // 응용 생성
        app.appCreate("testapp", "테스트응용", "테스트 응용입니다.", "1");

        // 응용관련 program 전송
        app.programTransfer(FileType.SOLVER, "project");
        app.programTransfer(FileType.INPUT, "project");
        app.programTransfer(FileType.OTHER, "project");

        // 응용관련 데이터 전송
        app.projectTransfer("project", "D:₩₩app₩₩input₩₩input₩₩", "kath2_", "sdf.gz", "43400001", "10000");

        // 응용의 준비 완료
        app.appReadyComplete("98", "16", ".NET", "5", "2", "180", "100");
    }
}
```

두 번째로 응용의 상태파악, 상태변경, 응용결과 다운로드 API들은 모두 응용 생성이 완료된 이후에 호출할 수 있다. 응용의 상태파악 API는 응용의 처리과정을 모니터링하기 위해 사용되며, 응용의 작업진행 정도와 응용의 상태를 각 메소드들의 호출을 통하여 파악할 수 있다. 응용의 상태 변경 API는 Korea@Home에서 수행중인 응용을 응용수행자가 중단시키거나 재개시키는 일을 한다. 응용결과 다운로드 API는 모든 작업이 완료된 응용에 대해 응용수행자가 응용결과를 다운로드받을 수 있도록 해준다.

5. 구현결과

그림 5에서 보여주는 자바 코드는 응용수행자에게 제공되는 인터페이스를 통하여 새로운 응용을 생성하는 코드이다. 표 3에서 제시된 인터페이스를 이용하여 새로운 응용을 생성하게 되면 응용은 Korea@Home에 등록되어 응용의 작업을 수행하게 된다. 이 코드의 테스트를 통하여 Korea@Home 데이터베이스에 응용에 관련된 실행파일 및 10,000개의 작업데이터를 업로드 하였다. 테스트를 위해 사용된 응용은 바이오 분야의 신약후보물질탐색이다. 이 응용은 데스크톱 그리드 컴퓨팅 시스템에서 대표적으로 사용되는 응용으로서 가상탐색 기술을 활용한 컴퓨터 시뮬레이션을 통해 신약 가능성이 있는 물질을 도출해 내는 응용이다.

응용이 생성된 이후에 응용수행자는 응용의 상태를 파악하는 인터페이스를 통하여 응용의 작업진행 상황을 모니터링할 수도 있고 응용의 작업이 모두 완료되었을 경우 응용결과를 다운로드 받아서 결과자료를 분석하는 프로그램을 만들 수도 있다.

6. 결론

본 논문에서는 데스크톱 그리드 시스템의 응용수행자를 위한 Restlet기반의 REST 방식의 웹 서비스에 대해 소개하고 Korea@Home 데스크톱 그리드 시스템에서 REST 웹 서비스의 구현에 대해 기술하였다. 또한, 응용수행자의 편의성을 고려하여 자바 클래스 형태의 인터페이스를 개발하였다. REST 웹 서비스 구현을 위하여 자동으로 데이터베이스 내의 내용을 다수의 Restlet으로 생성하는 모듈을 작성하고 URL 접근을 통하여 Korea@Home 내에서 수행되는 응용의 상태를 확인하는 방법을 제시하였다.

본 논문의 구현을 통하여 응용수행자는 스스로 응용을 Korea@Home 시스템으로 등록하고 상태 파악 및 결과를 다운로드 받을 수 있다. 즉 응용수행자는 자신이 필요로 할 때 자신의 로컬 컴퓨터에서 본 논문에서 구현된 REST 웹 서비스를 이용하여 유휴 PC자원을 최대한 활용할 수 있다. 또한 웹 서비스를 통해 응용의 중간 및 최종결과를 받아서 다른 정보와 접목하여 가치 있는 또 다른 결과를 산출하는 작업을 할 수 있다.

참고 문헌

[1] 박찬열, 길준민, 최장원, 박학수, 이필우, “Korea@Home: P2P형 인터넷기반 분산컴퓨터,” *한국정보과학회지*, 제22권, 제3호, pp. 26-34, 2004년 3월.

[2] D. Anderson, "BOINC: A System for Public-Resource Computing and Storage," *Proc. of the 5th IEEE/ACM Int. Workshop on Grid*

Computing, pp. 4-10, 2004.
[3] A. Chien, B. Calder, S. Elbert, and K. Bhatia, "Entropia: Architecture and Performance of an Enterprise Desktop Grid System," *J. Parallel and Distributed Computing*, Vol. 63, pp. 597-610, 2003.
[4] G. Fedak, C. Germain, V. Neri, and F. Cappello, "XtremWeb: A Generic Global Computing Systems," *Proc. of the 1st Int. Symp. on Cluster Computing and the Grid*, pp. 582-587, 2001.
[5] 길준민, 박찬열, 최장원, 박학수, 황일선, "Korea @ Home의 현황과 성능," *한국정보과학회 2004년도 가을학술발표논문집*, 제31권, 제2호(III), pp. 637-639, 2004년 10월.
[6] F. Liu, G. Wang, L. Li, and W. Chou, "Web Service for Distributed Communication Systems," *IEEE Int. Conf. on Service Operations and Logistics, and Informatics*, pp. 1030-1035, 2006.
[7] R. T. Thomas. Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine, 2000.
[8] Xuan Shi, "Sharing Service Semantics using SOAP-Based and REST Web Services," *IT Professional*, Vol. 08, No. 2, pp. 18-24, Mar/Apr, 2006.
[9] Lightweight REST framework for Java, http://www.restlet.org.