

Korea@Home 시스템에서 웹 서비스 Open API 활용을 위한 설계†

한창환* · 한연희* · 길준민** · 최장원*** · 윤준원***

Design for Utilizing Web Service Open API in Korea@Home Systems
Chang-Hwan Han* · Youn-Hee Han* · Joon-Min Gil** · Jang-Won Choi*** · Jun-Weon Yoon***

요 약

기존의 P2P기반의 분산컴퓨팅 시스템을 일반 응용수행자가 사용하기 위해서는 수동적으로 시스템에 위탁하여 수행하여 왔다. 이러한 구조에서 응용수행자가 자신이 원하는 응용을 분산컴퓨팅에서 수행하고 결과를 받아오는 것은 복잡하고 어려운 과정이었다. 본 논문에서는 이러한 과정을 간편화하고 분산컴퓨팅 시스템의 참여도를 높이기 위하여 웹 서비스 Open API를 사용하여 개방형 분산컴퓨팅 시스템을 설계하였다. 본 논문은 국내에서 2003년도부터 개발된 P2P 기반 분산컴퓨팅 시스템인 Korea@Home을 소개하고 이 시스템에서 효율적으로 Open API를 제공하기 위한 시스템 모델을 설정하고 테스트베드를 설계한다. 또한, 응용수행자의 응용과 Korea@Home 사이의 인터페이스를 정의하고 그 인터페이스를 활용한 응용 코드의 실행을 제시한다.

ABSTRACT

In the existing P2P-based distributed computing systems, application users have executed their applications passively. In this architecture, application users suffer from some complex and tedious processes, such as executing their application in the distributed systems, receiving final result from the systems, and so on. In this paper, we will design an open distributed system with web service Open APIs in order to enhance the application participation of Korea@Home by simplifying these complex and tedious processes. First, this paper introduces a P2P-based distributed system, namely Korea@Home, which has been developed in Korea from 2003. Second, we propose the system model to provide Open APIs for Korea@Home and design a testbed under the existing Korea@Home systems. Finally, we define interfaces between applications and Korea@Home systems, and present application codes utilizing the interfaces.

Keyword

분산컴퓨팅, 웹 서비스, Open API

1. 서 론

† 본 연구는 한국과학기술정보연구원 “컴퓨터연계활용기반구축(Korea@Home)” 사업의 지원에 의해서 수행되었음.

* 한국기술교육대학교 인터넷미디어공학부

** 대구가톨릭대학교 컴퓨터교육과

*** 한국과학기술정보연구원(KISTI)

P2P기반 분산컴퓨팅은 기존의 슈퍼컴퓨터에서 수행하던 응용을 대체할 수 있는 수단으로 연구가 활발히 진행되고 있다 [1-4]. P2P기반 분산컴퓨팅은 다양한 분야에 활용될 수 있고 또한, 국가적으로 추진

하고 있는 국가 6대 기술분야, 6Ts(IT, BT, NT, ET, ST, CT) 발전에 필수적인 기반을 제공할 수 있을 것이다. P2P기반 분산컴퓨팅에서 수행되는 응용에는 항공우주 비행체의 설계 해석뿐만 아니라, 자동차 등의 설계 및 개발, 반도체, PDP 등의 전자부품 시뮬레이션, 단백질 운동, 바이오 인포메틱스, 나노 시뮬레이션, 기상 시뮬레이션, 만화, 영화의 애니메이션 등이 있다.

하지만 2003년부터 시작된 국내의 대표적 P2P기반의 분산컴퓨팅 시스템인 Korea@Home 시스템[1,5]을 일반 응용수행자가 사용하기 위해서는 별도의 표준 API를 사용하지 않고 자신의 응용을 수동적으로 시스템에 위탁하여 응용을 수행하여 왔다. 이러한 점은 응용사용자가 자신의 응용을 위탁하여 분산컴퓨팅에서 확보한 유휴자원을 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 체계적이고 종합적인 방안이 요구된다.

본 논문에서는 사용자 생산 컴퓨팅 환경 (User-Created Computing Environment)을 응용수행자에게 제공하는 새로운 시스템을 설계한다. 본 시스템은 응용사용자가 자신이 필요로 할 때 언제든지 자신의 응용을 분산컴퓨팅 시스템을 활용하여 수행할 수 있는 웹 서비스 Open API를 제공한다. 이러한 Open API를 활용하여 분산컴퓨팅 시스템의 관점에서는 응용수행의 항시성을 보장해 주며 유휴자원의 활용 관점에서는 확보된 유휴자원을 최대한 활용할 수 있다는 장점이 있다. 또한, 응용수행자 입장에서는 응용의 위탁부터 결과의 활용까지 단순하고 단일화 된 Open API 방식으로 처리할 수 있어서 현재보다 많은 응용의 참여가 예상되므로 결과적으로 응용결과의 활용도 증대를 통한 P2P기반 분산컴퓨팅의 응용활성화에 기여할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대하여 살펴보고, 3장에서는 시스템 분석 및 설계를 살펴본다. 마지막으로 4장에서 본 논문의 결론을 맺는다.

II. 관련연구

2.1 P2P기반 분산컴퓨팅 시스템 - Korea@Home

한국과학기술정보연구원(KISTI)에서 2003년도부터 개발한 Korea@Home 시스템은 인터넷에 연결되어 있는 수많은 컴퓨팅 자원을 이용하여 대량의 연산을 요구하는 정보를 분산 처리하여 그 결과를 얻을 수 있는 시스템이다. 현재 한국의 초고속 정보통신 인프라를 최대한 활용하여 저비용, 고효율의 그리드 시스템을 구현하려는 목적을 지닌다. 만약 이 시스템에 3만명의 PC 자원 제공자가 동시에 참여할 경우 (P-III 500MHz 기준), 이론적으로 ASCII White 슈퍼컴퓨터에 버금가는 성능인 12.7 테라플롭스 정도의 성능을 얻을 수 있다 [5].

Korea@Home 시스템에 대한 자원제공자는 일반 네티즌이며, 현재는 대다수 사람들이 호응할 수 있는 공공적인 성격의 애플리케이션을 발굴하고 기반 소프트웨어를 개발하여 관련 산업의 활성화를 유도하는 하기 위하여 많은 노력을 기울이고 있다.

2.2 웹 서비스 Open API와 REST

웹 서비스 Open API란 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, eBay등을 비롯하여 국내의 Naver 및 Daum 과 같은 곳에서 독자적인 웹 서비스 Open API를 제공하고 있다.

REST(Representational State Transfer)란 대규모 네트워크 시스템을 위한 아키텍처로서 2000년 Roy Fielding의 박사 학위 논문에서 처음 제안되었다[7]. REST 아키텍처는 웹상의 모든 것들은 URL(Representation)로 표현되며 잘 정의된 URL을 통해 웹 애플리케이션을 구동하고 그 결과를 전달받아 처리하는 것에 중점을 두고 있다. REST는 웹

에 존재하는 리소스의 상태가 핵심이며 주고 받는 메시지의 형식이 자유롭다. 같은 리소스도 POX (Plain Old XML), JSON(JavaScript Object Notation), RSS와 ATOM 등으로 다양하게 표현할 수 있다. 하나의 리소스에 대해 CRUD(Create, Retrieve, Update, Delete - 만들고, 읽고, 고치고, 지우는 처리)를 HTTP의 메소드 POST, GET, PUT, DELETE와 대응하여 리소스의 URI로 리소스를 요청한다[8]. 본 논문에서 제안하는 시스템이 활용하는 웹 서비스 Open API는 REST 아키텍처를 사용할 것이다.

III. 시스템 분석 및 설계

3.1 시스템 모델

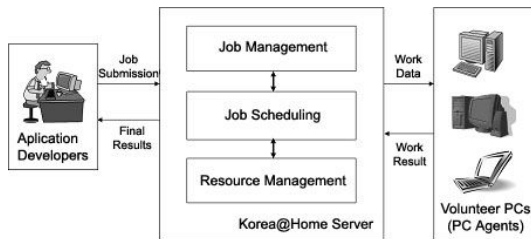


그림 1. Korea@Home 시스템의 물리적 개체

인터넷 서비스의 패러다임의 변환 추세에 맞추어 인터넷기반 분산컴퓨팅 플랫폼인 Korea@Home이 보다 효율적으로 응용수행자에게 응용환경을 제공하여 이들의 참여를 유도하도록 웹서비스 Open API에 기반을 한 시스템을 구축하고자 한다. Korea@Home에서는 응용수행자의 참여 욕구를 충족시키기 위해 웹 서비스 기술과 접목된 컴퓨팅 자원의 활용 극대화가 필요할 것이다. 이를 위해 먼저 현재 Korea@Home 시스템의 주요 물리적 개체별로 기능을 나누면 그림 1과 같으며 각 물리적 개체의 역할은 다음과 같다.

- 분산컴퓨팅 서버 : 자원제공자와 응용을 중재하며 자원제공자에게 작업을 분배하고, 작업과 자원 제공자를 관리하며, 결과수거의 역할을 수행
- 자원제공자 : 분산컴퓨팅 서버에서 제공하는 응용의 단위작업을 수행하고, 수행결과를 서버로 다

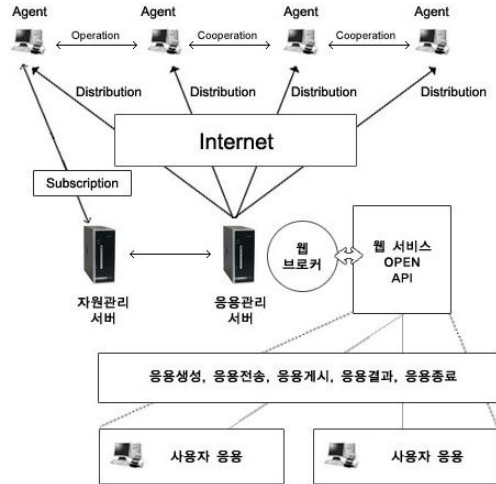


그림 2. 웹 서비스 Open API를 활용한 인터넷기반 분산컴퓨팅 모델

시 되돌려주는 역할을 수행

- 응용수행자 : 자신이 필요한 응용을 서버로 위탁하고, 최종수행결과를 활용하는 주체

현재 응용수행자가 자신의 응용을 Korea@Home 시스템에 수행하기 위해서는 복잡한 과정을 거쳐야 한다. 또한, 대다수의 응용수행자가 자신의 응용분야에 대해서는 전문적인 지식을 갖고 있지만 Korea@Home시스템 등과 같은 고성능 컴퓨터 방식은 단지 계산수행을 위한 도구로 생각한다. 그러므로 계산수행 도구로서 Korea@Home 시스템을 최대한 효율적으로 이용하려면 쉽고 투명성이 보장되는 자동화 방식으로서 웹 서비스 Open API를 사용하는 것이 적합하다. 그림 2는 본 논문을 통해 개발하고자 하는 모델을 간략히 보여준다. 웹 서비스 Open API를 사용하기 위해서 추가되는 모듈에 대한 주요 기능은 다음과 같다.

- 웹 서비스 Open API : 사용자생성컴퓨팅 환경을 지원하는 응용생성, 응용전송, 응용개시, 응용결과, 응용종료 API 등의 웹 서비스 Open API들로 이루어진다.
- 웹 브로커 : 기존 분산컴퓨팅 시스템과 기능적

표 1. 응용수행자가 사용하는 인터페이스

인터페이스명	기능설명	비고
AppAuthenticate	응용수행자의 인증	응용의 최초시작을 위한 응용위탁 API
AppCreate	응용의 생성	
AppTransfer	응용프로그램 및 응용데이터 전송	
AppStart	응용의 시작을 알림	
AppEnd	응용의 종료를 보고받음	
AppResultDownload	응용결과를 다운로드	
AppStatus	응용의 상태를 파악	응용상태 관리 API
IsAppStart	응용의 시작상태를 알아봄	
IsAppEnd	응용의 종료상태를 알아봄	
AppPause	응용의 수행을 잠시 멈춤	
AppRestart	멈춰진 응용을 재개시킴	
GetTotalWorkUnit	응용의 총 작업개수 얻어옴	응용 작업정보 API
GetCompleteWorkUnit	응용의 완료 작업개수 얻어옴	
GetCompleteWorkTime	응용의 완료작업의 총 수행시간을 얻어옴	
GetTotalWorkList	응용의 작업리스트 얻어옴	
GetCompleteWorkList	응용의 완료된 작업리스트 얻어옴	

호환성을 유지하면서 웹 서비스 Open API와 상호기능을 증대한다.

○ 응용수행자의 응용 : 웹 서비스 Open API를 활용하여 응용수행자 자신의 목적에 맞는 응용을 생성, 관리 및 활용한다.

3.2 인터페이스 설계

다음으로 위에서 살펴본 모듈의 주요기능을 구체적으로 실현하고 응용수행자가 자신의 응용생성을 위해 필요한 기능별 인터페이스를 설계한다.

표 1은 응용수행자가 Korea@Home 시스템으로 자신의 응용을 위탁하고 응용을 관리하고자 할 때 사용하는 인터페이스를 보여준다. 표 1에서 볼 수 있듯이, 응용수행자가 사용하는 인터페이스로는 크게 응용의 최초시작을 위해 필요로 하는 응용위탁 API들, 응용수행자가 자신의 응용 상태를 파악하고자 할 때 사용하는 응용 상태 관리 API들, 그리고 응용의 통계정보와 관련된 응용 작업정보 API로 나누어진다.

첫 번째로 응용위탁 API는 응용수행자가 제일 처음에 자신의 응용을 Korea@Home 시스템에 위탁하고 최종 결과를 받을 때까지 사용하는 API들이다. AppTransfer 인터페이스는 현재의 수동방식에 의해 ftp 프로그램 등을 이용하여 응용프로그램과 응용데이터를 전송하는 방식을 탈피하여 전송파일들을 AppTransfer 인터페이스의 파라미터로 설정하면 자동으로 파일들을 전송하는 방식으로 구현한다. 또한, AppStart 인터페이스는 이 인터페이스를 응용수행자가 호출함과 동시에 Korea@Home DB의 작업정보 상태를 시작상태로 설정하도록 하여 Korea@Home 시스템이 응용프로그램과 작업데이터를 에이전트로 할당하도록 한다. 물론 이때의 DB 접근은 본 연구개발을 통해 개발되는 웹 브로커의 제어의 의해서 이루어지는데 외부의 잘못된 요청이나 접근을 차단하는 인증처리 절차를 거치게 하여 불특정 다수로부터 보호하는 보안기능을 수행하도록 한다.

두 번째로 응용 상태관리 API는 응용수행자가 자신의 응용을 위탁한 상태에서 응용의 처리 과정을 모니터링 하는데 필요한 API들을 나타낸다. 응용이

시작상태에 있는지 혹은 응용이 종료상태에 있는지에 대한 처리를 각각 IsAppStart와 IsAppEnd 인터페이스를 호출하여 응용사용자는 알아낼 수 있다. 또한 응용수행을 잠시 멈추는 기능은 AppPause로, 그리고 멈춰진 응용수행을 다시 재개시키는 기능으로 AppRestart를 사용할 수 있다.

세 번째로 제공되는 응용작업 정보 API는 응용의 통계정보를 파악하기 위해 응용을 위탁한 응용수행자나 혹은 일반사용자에게 응용수행에 대한 통계 정보를 제공해 주려는 목적을 갖고 있다. 예를 들어, 응용을 시작한 이후로 현재까지 어느 정도의 작업을 수행했는지의 정보를 파악하기 위해서는 응용사용자는 GetCompleteWorkUnit 인터페이스를 이용할 수 있다.

3.3 테스트베드 설계

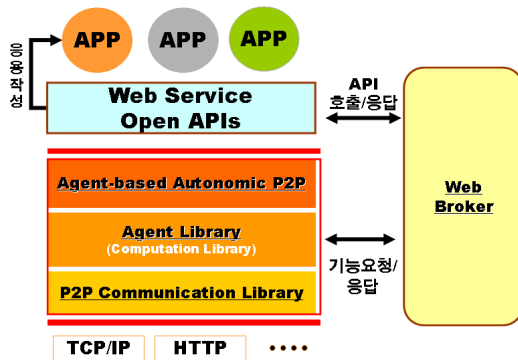


그림 3. 테스트베드의 구성도

테스트베드는 분산컴퓨팅 시스템과 최대한 호환성을 유지하도록 구축해야 한다. 그림 3은 본 논문을 통해 구축되어진 테스트베드의 구성도를 보여준다.

웹 서비스 Open API는 응용계층의 상위에 위치하여 응용수행자가 이 API를 이용하여 자신의 목적에 맞는 응용을 개발하고 테스트베드를 활용하여 응용을 위탁하고 관리하는 것을 도와준다. 작성된 응용은 응용내부에 기술된 웹 서비스 Open API의 호출을 통하여 테스트베드내부기능의 호출을 시도하는데, 이때 웹 브로커가 중간에서 Korea@Home DB로 응용에 관련한 기능을 요청하고 응답을 수행하여 원하는

기능을 수행하고 응답결과를 응용으로 전달해 준다.

그림 3과 같은 테스트베드가 구성되면, 응용수행자는 자신의 응용을 웹 서비스 Open API를 이용하여 생성할 수 있다. 이러한 방식을 사용하는 응용 코드의 실행을 의사코드(pseudo code)로 살펴보면 그림 4와 같다.

```

main() {
    // 응용 자체의 코드
    // 테스트베드 이용부분 시작
    // 응용수행자의 인증
    AppAuthetication();
    // 테스트베드 활용을 위한 응용 생성
    AppCreate();
    // 응용프로그램과 응용데이터 전송
    AppTransfer()
    // 응용 시작
    AppStart()

    do {
        // 응용 자체의 코드 삽입 가능
        // 중간에 처리된 결과를 활용하여 새로운
        // 작업데이터 생성도 가능

        ...
        // 응용의 총수행 작업개수 얻어냄.
        GetCompleteWorkUnit();
        ...
    } while (!IsAppEnd())
    // 응용 종료의 대기
    // 응용 결과파일들의 다운로드
    AppResultDownload();
    // 테스트베드 이용부분 끝
    // 응용 결과의 활용을 위한 코드
    // 응용수행자가 후처리 결과를 GUI 방식의
    // 처리를 위한 코드 등의 추가 가능
}

```

그림 4. 웹서비스 Open API를 이용한 응용 코드의 실행

그림 4와 같은 방식으로 대용량 계산을 필요로 하는 부분을 본 연구과제에서 구축할 테스트베드를 이용함으로써 본 연구과제의 테스트베드가 일종의 계산수행을 위한 도구로 활용되도록 한다. 응용수행자는 아울러 대용량 계산을 위한 작업데이터의 전처리 과정과 최종결과의 후처리 과정을 하나의 단일 프로그램으로 작성할 수 있어서 기존 Korea@Home의 응용수행 방식보다 체계적이고 종합적으로 응용을 준비하고 계산결과를 활용할 수 있는 체계를 구축할

수 있다. 특히, 테스트베드에서 수행한 결과를 하나의 프로그램 코드 내에서 활용하도록 하여 응용자체의 목적에 맞는 새로운 응용결과를 응용수행자 스스로 생성할 수 있다.

IV. 결 론

본 논문에서 설계한 시스템을 통해 Korea@Home의 응용수행자는 자신이 필요한 응용을 언제 어디서나 웹 서비스 Open API를 활용하여 Korea@Home의 유휴 PC자원을 최대한 활용할 수 있다. 또한, 웹서비스 Open API를 통하여 응용수행 중간 및 최종 결과를 원하는 시기에 되돌려 받아 자신의 응용에 다른 정보들과 접목하여 가치 있는 또 다른 결과를 산출할 수 있을 것이다. 이와 같이 본 논문에서 설계한 시스템은 Korea@Home에서 확보한 유휴 PC 자원을 다양한 응용분야로 최대한 활용할 수 있도록 지원하며, 이를 통한 첨단과학 및 기술 분야의 응용분야 확산으로 Korea@Home의 성과확산에 기여할 수 있을 것이다.

참고문헌

- [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] Feng Liu, Gesan Wang, Li Li, Wu Chou, Avaya Labs. Res., Avaya Inc., Basking Ridge, NJ, Web Service for Distributed Communication Systems, 2006. SOLI '06. IEEE

- [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.