

기존 어플리케이션의 웹 기반 제어 시스템

홍상현^{0*} 황신태^{*} 정갑주^{**} 이지수^{***}

^{0*}국민대학교 컴퓨터학부, ^{*}진국대학교 인터넷미디어공학부, ^{**}한국과학기술연구원
{ibmmsx2^{0*}, sthwang^{*}}@cs.kookmin.ac.kr, {jeongk^{**}}@konkuk.ac.kr, {jysoo^{***}}@hpcnet.net.kr

Web-based Control System for Existing Applications

Sanghyun Hong^{0*} Suntae Hwang^{*} Karpju Jeong^{**} Jysoo Lee^{***}

^{*}Department of Computer Science, Kookmin University

^{**}College of Information and Communication, Konkuk University

^{***}Korea Institute of Science and Technology Information

요 약

기존에 작성된 어플리케이션은 독립적이거나 클라이언트-서버 모델을 기초로 하고 있다. 하지만 인터넷의 대중화와 더불어 원격지에 있는 정보를 웹을 통해 접속하여 처리하는 기술이 보편화됨에 따라 기존 어플리케이션도 하나의 서비스 개념으로 발전해 나가고 있다. 본 논문에서는 이러한 웹기반 기술을 이용하여 기존의 어플리케이션을 웹 브라우저를 통해 원격에서 제어할 수 있는 구조를 제안한다. 이 구조는 클라이언트, 웹서버, 미들웨어의 세 부분으로 구성되어 있으며 각 요소별로 원격 호스트에 존재하기 때문에 다양한 응용이 가능하다.

1. 서 론

어플리케이션 기반의 소프트웨어는 로컬의 컴퓨터 상에서 수행해야 하는 한계성이 있기 때문에 원격에서 이러한 소프트웨어를 사용하려면 관련 패키지를 가져와서 설치해야 하는 단점이 있다. 하지만, 현재 인터넷의 보편화와 더불어 웹 기술의 발달로 인해 위와 같은 어플리케이션도 웹을 통해 하나의 서비스를 제공하는 개념으로 발전해 가고 있으며, 사용자는 원격지에서도 단순히 웹 브라우저를 통해 자신이 원하는 어플리케이션을 실행하여 관련 기능의 수행 여부를 조절할 수 있는 기술이 가능하게 되었다.

본 논문은 인터넷을 이용하여 사용자가 웹 환경하에 기존의 어플리케이션을 수행하고, 원하는 job을 제어할 수 있는 구성을 제안하고 그에 대한 장점을 소개한다. 그리고 HTTP 프로토콜을 이용한 각 요소 간의 통신 기법에 대하여 설명하며 마지막으로 제안된 기술에 대한 그리드 컴퓨팅 환경에서의 활용가능성을 언급하고자 한다.

2. 관련연구

2.1 M3C

M3C (Managing and Monitoring Multiple Clusters)는 여러 도메인에 존재하는 클러스터를 통합, 관리하기 위해 만들어진 도구이다[1].

M3C는 클러스터 관리자라 위해 웹환경의 GUI를 제공하며 클러스터에 적용할 수 있는 다양한 back-end 툴을 사용할 수 있도록 고안되었다. 사용자는 웹 브라우저를 통해 여러 도메인에 존재하는 클러스터의 계산노드를 모니터링 할 수 있고 job을 수행할 수 있다.

2.2 Cactus

Cactus는 고성능 다차원 시뮬레이션을 협업환경하에

수행하기 위해 제작된 프레임워크로서 천체물리학등의 분야에서 계산량이 많이 요구되는 문제 (블랙홀의 충돌 문제등)를 다루기 위해 구현되었고, 현재 그리드 컴퓨팅 환경하에 수행될 수 있도록 확장되었다[2].

Cactus는 스케줄링 및 기반환경을 조성하는 flesh와 응용 모듈인 thorn으로 구성되어 있다. 사용자가 원하는 기능을 부여한 thorn은 flesh에 통합되어 다른 thorn과 같이 문제를 해결하고 다른 호스트에 존재하는 cactus와 연동되어 손쉽게 병렬처리가 가능하다.

여러 가지의 thorn이 현재 개발되어 있으며 그중에서도 HTTP thorn은 웹서버를 내장하여 사용자가 웹을 통하여 원하는 job을 제어할 수 있는 환경을 제시하며 각 thorn의 파라미터 파일을 수정함에 따라 사용자 관리와 job에 대한 명시등이 손쉽게 가능하다.

2.3 그리드 포털

여러 곳에 분산되어 있는 컴퓨팅 자원을 이용하여 많은 양의 계산량과 데이터를 필요로 하는 문제를 해결하는 그리드 어플리케이션은 웹 기반의 포털 사이트를 통해 배포하는 방법이 효과적이다. 포털 사용자는 웹이 가지는 pervasive한 면과 플랫폼에 독립적인 면, 그리고 분산된 자원관리가 용이하다는 점에서 손쉽게 접속하여 배포된 어플리케이션을 이용할 수 있고, 포털은 접근할 수 있는 단일의 중앙 어플리케이션 서버에 무수히 많은 그리드 어플리케이션을 배포하여 서비스를 제공할 수 있는 매커니즘을 지원한다[3].

3. 제어시스템의 설계

본 시스템은 기능상으로 다음과 같이 명확하게 구별된 세 개의 요소로 이루어져 있다. 첫째, 사용자에게 어플리케이션을 제어할 수 있도록 보여주는 웹기반의 유저 인터페이스이며, 둘째로는 이러한 인터페이스로부터의 요

청을 처리하는 웹서버와 CGI, 마지막으로 기존의 어플리케이션과 연결되어 있어 사용자의 요청을 실제로 처리하는 XML 미들웨어이다.

3.1 기본 구성

본 논문에서 구현한 기본 구조는 그림 1과 같다.

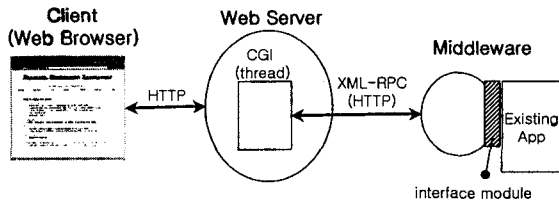


그림 1. 전체 구성도

본 시스템의 전체적인 흐름은 사용자가 웹 브라우저를 통하여 해당 어플리케이션의 제어 명령을 보내면 웹서버는 이 요청을 받아서 해석한 후, 해당 어플리케이션이 존재하는 호스트의 미들웨어에게 명령을 전달한다. 그 후 명령에 따라 기존 어플리케이션을 적절하게 수행하고 그 결과를 반대의 경로를 통해 사용자에게 보여준다. 즉, 원격지에 있는 사용자로부터 해당 호스트에 있는 어플리케이션까지의 연결점은 세 개의 파트로 구성되어 있으며 (클라이언트, 웹서버, 미들웨어) 각 요소는 인터넷으로 연결된 원격지 호스트에 존재한다.

3.2 장점

위 그림 1과 같은 구성에 대한 장점은 크게 다음과 같이 나눌 수 있다.

첫째, 각각의 요소가 동일 호스트가 아닌 원격지에 존재할 수 있기 때문에 각각의 기능이 명확하게 구분되어 다양한 관점에서의 응용이 가능하다는 점이다. 예를들면 단일 웹서버와 여러개의 미들웨어를 구성하여 사용자에게 다양한 어플리케이션을 동시에 웹을 통해 제어할 수 있는 서비스가 가능하다.

둘째, 각 요소간의 연결 종단점이 HTTP 프로토콜 및 XML-RPC의 표준을 이용하기 때문에 수행환경과 플랫폼의 제약없이 어느 곳이라도 구축이 가능하다.

셋째, 사용자의 관점에서 볼 때 웹 서버의 후면에 연결되어 있는 다양한 구성요소를 고려할 필요없이 인터넷이 연결된 곳이면 어디에서나 손쉽게 접근하여 어플리케이션의 제어 및 결과를 받아 볼 수 있다는 투명성을 보장한다는 점이다. 이러한 접속의 용이함은 시간이 많이 소요되는 batch job이나 계산량이 많은 job을 수행할 때 한 장소에서 job을 시작하고 다른 장소에서 결과를 받아 볼 수 있는 편리함을 제공한다.

4. 제어 시스템의 구현

본 시스템은 객체지향언어인 파이썬을 이용하여 주요 기능을 구현하였다[4].

4.1 클라이언트와 웹 서버

웹 브라우저를 통한 유저 인터페이스는 HTML을 이용

한 웹 페이지와 더불어 동적인 표현을 위하여 자바 스크립트를 사용하였다. 간단한 GUI는 HTML 만으로도 작성 가능하기 때문에 구현의 오버헤드가 줄어들었다.

웹 서버는 파이썬으로 구현된 medusa를 사용하였는데 그 이유는 기존에 존재하는 웹 서버와 비교해 볼 때 작고 단순할 뿐만 아니라 성능면에서 뛰어나기 때문이다 [5]. 사용자의 요청을 처리하는 CGI의 후면에는 어플리케이션이 구동되고 있는 호스트의 미들웨어에게 정보를 전달하는 RPC 클라이언트가 존재하고 있으며 해당 요청의 처리 결과를 다시 사용자에게 보여주는 스크립트가 존재한다.

4.2 XML 미들웨어

XML-RPC는 이기종의 운영체제위에서 소프트웨어의 수행이 가능하게 하며, 인터넷을 통하여 원격지의 함수를 호출하기 위해 다양한 환경하에 수행되도록 만들어진 표준 프로토콜이다. 원격지의 함수를 호출하기 위해 HTTP 프로토콜을 이용하여 XML로 인코딩된 정보를 주고 받기 때문에 플랫폼에 독립적이며 단순하면서도 복잡한 정보를 주고 받을 수 있는 장점이 있다[6].

본 시스템의 미들웨어는 XML-RPC를 사용한 서버로 구현되어 있으며 어플리케이션을 제어하기 위하여 기능 단위로 함수를 정의한다.

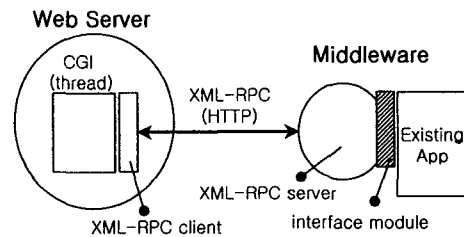


그림 2. XML-RPC를 이용한 어플리케이션 제어

그림 2에서와 같이 CGI 스크립트의 RPC 클라이언트로 부터 원하는 함수의 호출요청이 이루어지면 미들웨어에서 수행되고있는 RPC 서버는 해당 함수안에 정의되어 있는 어플리케이션의 제어 루틴을 수행하게 된다. 인터페이스 모듈은 파이썬으로 구현된 RPC 서버에서 기존 어플리케이션의 함수로 명령을 전달하기 위해 호환성을 담당하는 역할을 한다. 따라서, 기존 어플리케이션을 제어하는 명령만 알면 본 시스템을 적용하여 웹 제어기능을 추가 할 수 있다.

XML-RPC를 이용한 서버와 클라이언트는 표준에 준수되어 구현되어 있기 때문에 특정 언어 및 환경에 종속되지 않는 독립적인 미들웨어 역할을 수행할 수 있다. 그러므로, 위의 구성요소는 어떠한 이종의 환경에서도 구축이 용이하다.

4.3 제어시스템의 적용 예

본 논문에서 구현한 시스템은 plinda[7]라는 장애허용 병렬 프로그램을 위한 시스템 소프트웨어에 웹을 통한 제어 기능을 추가하기 위해서 사용되었다. 그림 3은 클

러스터 환경에서 수행되고 있는 plinda를 제어하기 위하여 만들어진 Tcl/Tk 기반의 관리자용 어플리케이션이다. 이것은 클러스터에서 수행되는 plinda 서버 및 데몬과 통신하여 사용자가 원하는 job을 처리한다.

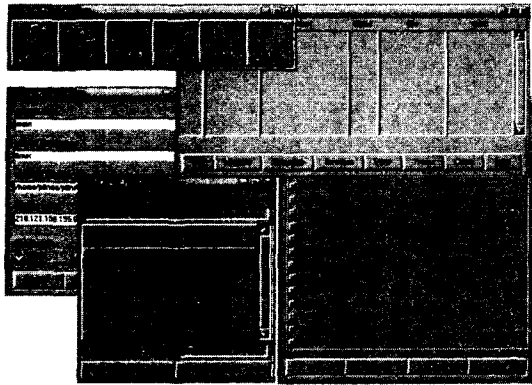


그림 3. Tcl/Tk 기반의 plinda 관리자용 어플리케이션

본 논문에서 구현된 시스템을 위의 관리자 프로그램에 적용한 결과 유저 인터페이스 부분은 그림 4와 같이 웹 페이지로 변경되었다.

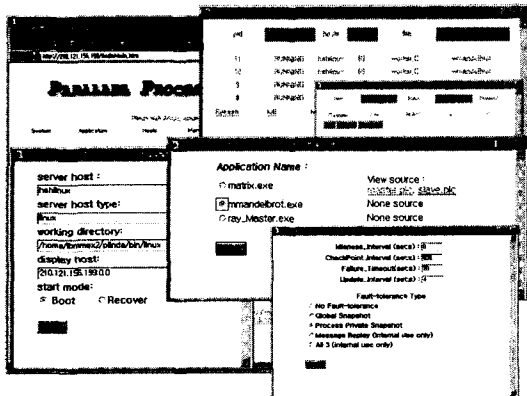


그림 4. 웹 기반으로 변경된 관리자용 어플리케이션

변경된 웹 페이지를 통하여 사용자는 원격지에서 자신이 원하는 제어명령을 보내며 이러한 명령은 그림 1에 명시된 경로를 통하여 미들웨어에서 수행되고 있는 plinda 어플리케이션에게 전달된다. 이렇게 웹 제어 기능을 추가하는 과정에서 관리자 어플리케이션의 소스는 수정하지 않았으며 기존의 Tcl/Tk에 의해서 plinda로 전달되는 명령체계만 이해하면 되었다.

따라서, 위에서 예로 제시한 어플리케이션 이외의 다른 것도 똑같이 적용이 가능하다. 이 경우 해당 어플리케이션의 주요 기능을 담당하는 함수를 웹 서버의 CGI 스크립트에서 호출할 수 있도록 RPC 서버의 함수와 맵핑시킨다. 그러므로, 외부에서는 RPC 함수를 호출하지만 내부적으로는 해당 어플리케이션의 함수가 호출되어 그 기능을 수행할 수 있도록 연결되는 것이다. 그리고 어플리케이션에 맞는 유저 인터페이스를 웹 기반으로 새로 작

성하여 제공한후 사용자의 명령을 전달하는 CGI 스크립트와 RPC 서버와 연동시키면 된다.

5. 결론 및 향후 계획

본 논문에서는 기존 어플리케이션을 웹을 통해 제어할 수 있도록 세 부분의 구조를 가지는 시스템을 제안하였다. 제안된 시스템을 적용하기 위해서는 기존 소프트웨어를 수정할 필요가 없으며, 각 요소는 기능상으로 명확하게 구분되어 있고 HTTP 프로토콜과 XML-RPC를 이용하여 원격지에서 서로 통신이 가능하기 때문에 장소 및 환경에 대한 제약성이 사라지는 장점이 있어 앞으로 많은 응용분야에 적용이 가능하다.

이러한 구조상의 장점은 차세대 컴퓨팅 모델인 그리드 컴퓨팅 환경에서 수행될 수 있는 어플리케이션이나 job에 적용될 수 있다[8]. 그리드를 이용하는 사용자의 관점에서 볼 때, 그들이 요구하는 job은 단순히 포털사이트에 접속하여 본인이 원하는 작업을 지시하고 결과를 받아보기를 원할 것이다. 하지만 그러한 포털사이트 후면에는 이기종의 방대한 자원이 그리드를 통해 통합되어 있고 사용자에게는 마치 단일 시스템처럼 보여질 것이다. 이러한 상황에서 포털사이트와 그리드 환경을 잇는 중간계층에서 본 논문에서 제시한 시스템이 적용될 수 있으며 이기종 환경의 그리드 특성과 잘 부합되는 결과를 가져올 것이다.

향후과제로 본 시스템과 그리드 미들웨어인 Globus Toolkit[9]을 연동시켜 그리드 포털사이트를 제작하는 일과, 이를 통해 신약개발시 사용되는 대규모의 단백질 분자모사 프로그램을 원격으로 제어할 수 있는 환경을 개발 중에 있다.

참고문헌

- [1] Michael Brim, Al Geist, Brian Luethke, Jens Schwidder, Stephen L. Scott, "M3C: Managing and Monitoring Multiple Cluster" Cluster Computing and the GRID, p386, 2001
- [2] <http://www.cactuscode.org>
- [3] Gabrielle allen, Edward seidel, John shalf, "Scientific Computing on the GRID" <http://www.byte.com>, 2002 Spring.
- [4] <http://www.python.org>
- [5] A High-Performance Internet Server Architecture <http://www.nightmare.com/medusa>
- [6] <http://www.xmlrpc.com>
- [7] Karpjoo Jeong, "Fault-tolerant Parallel Processing Combining Linda, Checkpointing, and Transactions" Ph.D. Thesis, Department of Computer Science, New York University, 1996.
- [8] I. Foster, C. Kesselman, "The Grid: Blueprint for a Future Computing Infrastructure". Morgan Kaufmann Publishers, 1999.
- [9] <http://www.globus.org>