

◎ 논문

인터넷을 활용하기 위한 수치해석 프로그램의 재구성 방법

송희용¹, 고성호²

Reconstructing Methods of Numerical Analysis Program for Utilizing the Internet

H. Y. Song and S. H. Ko

The present study introduces an architecture for performing efficient numerical analysis by using the Internet and three reconstructing methods of existing numerical analysis codes were presented in order to utilize the architecture. These methods were implemented into a computational fluid dynamics program for solving two-dimensional transient flow problems with free surface. The program was reconstructed with Java technologies and compared with the original one. This study will be a preparation for numerical analysis to participate in web services for engineering.

Key Words: 인터넷(Internet), 수치해석(Numerical Analysis), 전산유체역학(Computational Fluid Dynamics), 자바(Java), 확장성 표기 언어(XML), 애플릿-서블릿 통신(Applet-Servlet Communication), 웹 서비스(Web Services)

1. 서론

인터넷은 그 관련 기술들이 빠르게 발전하면서 다른 많은 산업을 흡수 또는 지원하고 있다. 이는 인터넷을 활용하여 많은 이점을 얻을 수 있기 때문이며 공학 분야 또한 예외일 수 없다. 인터넷의 가장 큰 장점은 거리에 구애받지 않고 필요한 자원에 접근 또는 공유할 수 있다는 것이다. 이러한 장점은 자연스럽게 인터넷을 배포 환경으로 자리잡게 하였으며 개발자들은 여기에 만족하지 않고 인터넷을 개발 및 실행 환경으로 활용하고 있다.

인터넷은 이제 개발 방법 및 시스템에 상관없이 그 안에서 공존하며 쉽게 통합될 수 있도록 발전하고 있다. 이러한 인터넷의 장점을 제대로

활용하기 위해서는 무엇보다 새로운 기술이나 표준을 쉽게 접목시킬 수 있어야 한다. 그러나, 기존의 수치해석 프로그램은 구조적 프로그래밍 방식으로 개발되어 있어 프로그램을 수정하거나 분리하는 작업이 매우 어려우며 이는 새로운 기술이나 표준을 적용하기 어렵게 한다. 게다가 새롭게 만들어지는 프로그램 역시 고전적인 방식으로 개발되고 있다. 이는 토대가 되는 방대한 양의 기존 자원을 전환하는 작업이 쉽지 않기 때문이다. 그러나, 이러한 이유로 새로운 기술이나 표준의 도입을 미룬다면 결국 개발자들은 더 큰 어려움에 봉착하게 될 것이다.

국내의 연구소나 학교에서는 매우 유용한 연구용 코드들을 가지고 있다. 그러나, 폐쇄된 환경에서 사용되고 있어 일부 특정인들에 의해서만 사용되거나 대부분 사장되어 가고 있다. 본 연구는 이러한 자원들을 일반화시키기 위한 시도로 시작되었으며 기존 상용 프로그램과의 차별화를 위해 인터넷 기반의 정보 기술을 활용해서 그 대

* 2003년 01월 05일 접수

*1 비회원, 충남대학교 대학원

*2 정회원, 충남대학교 기계설계공학과

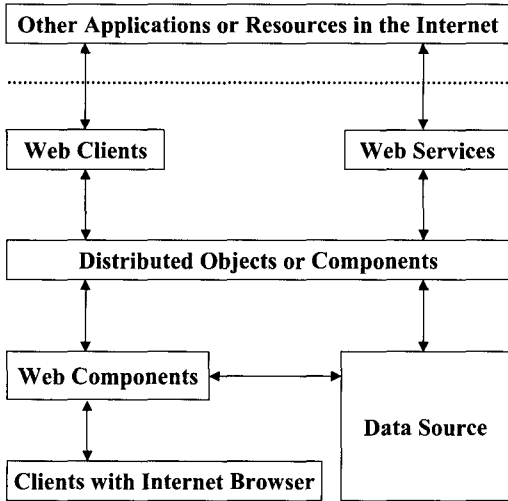


Fig. 1 Architecture for performing efficient numerical analysis by using Internet technology

상을 인터넷 사용자들로 확대하고자 하는 욕심으로 진행되었다.

본 연구의 궁극적인 목적은 제대로 활용되지 못하고 있는 연구용 코드들을 일반화시키기 위한 하나의 접근 방법으로 기존의 수치해석 프로그램에 인터넷의 장점을 활용하고자 하는 것이다. 본 연구에서는 이를 위해 수치해석 프로그램의 재구성 방법을 고안해 보고 일부를 실제 구현해 봄으로써 그 가능성 및 타당성을 확인하고자 하였다. 인터넷의 장점을 수치해석에 활용하고자 하는 연구는 수행 성능을 향상시키기 위한 연구와 수행 절차를 개선하기 위한 연구로 나눌 수 있다. 그러나, 인터넷의 장점을 제대로 활용하기 위해서는 두 연구가 동시에 이루어져야 하며 본 연구는 그 밑바탕이 될 것이다.

2. 고안된 재구성 방법

선행 연구[1]를 통하여 기존의 수치해석 프로그램을 간단한 웹 애플리케이션으로 구현하고 서버 측 계산 과정에서 속도를 향상시키기 위한 시도를 해 봄으로써 그 가능성 및 문제점 등을 확인하였다. 이를 토대로 하여 Fig.1과 같은 구성을 결정하였다.

사용자들은 인터넷 브라우저를 통해 접속하여 전처리 과정을 수행하게 되고 그 결과를 웹 컴포

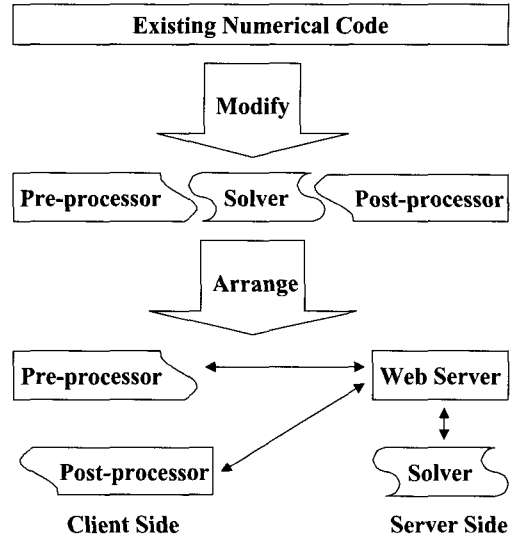


Fig. 2 A reconstructing method using arrangement and communication

넌트에 전송하게 된다. 사용자에 의해 호출된 웹 컴포넌트는 분산 객체 또는 분산 컴포넌트들과 어우러져 계산을 수행하게 된다. 개발자는 성능을 향상시키기 위한 작업을 이 과정에서 이루어지도록 구현해야 하며 공통 저장소로서 DBMS(Database Management System)와 같은 데이터 자원을 활용하게 된다. 또, 이 시스템은 XML(Extensible Markup Language) 기반의 웹 서비스(Web Services)[2]를 통해 다른 애플리케이션 또는 인터넷 안의 다양한 자원과 협업을 할 수 있도록 하고 있다. 본 연구에서는 이와 같은 구성에 적용시킬 수 있도록 기존 수치해석 프로그램의 재구성 방법을 고안해 보았다.

2.1 배치와 통신

방대한 양의 기존 코드에 손을 대는 일은 결코 쉬운 일이 아니다. 따라서, 많은 개발자들은 되도록 쉬운 방법을 원하고 있다. 가장 쉬운 방법은 아마도 기존 프로그램을 수정하지 않고 HTTP(Hypertext Transfer Protocol)를 통한 파일 전송을 이용하는 방법일 것이다. 그러나, 이 방법은 FTP(File Transfer Protocol)를 통한 파일 전송과 텔넷(telnet)을 통한 프로그램 실행 등 사용자의 번거로운 작업을 좀 줄여주는 것 외에 어떠한 의미도 없게 된다. 단지 그 안에서 접근하여 실행할 수 있다는 것만으로 인터넷 환경의

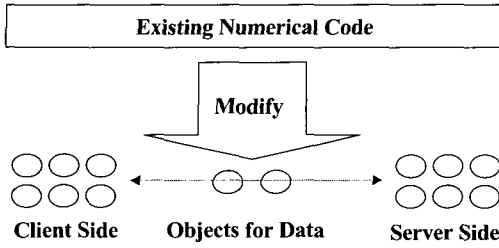


Fig. 3 A reconstructing method using objects for flexibility

장점을 제대로 이용한다고 할 수 없다. 배치와 통신을 이용하는 방법은 비교적 쉽게 기존 수치해석 프로그램에 인터넷 환경을 활용할 수 있는 방법이다. 기존 코드를 인터넷 프로그램에 적합한 언어로 전환하고 이를 사용자 측과 서버 측에 그대로 혹은 간단히 분리하여 배치한 후 이들 사이에 통신을 구현하면 인터넷 환경에서 운영되는 수치해석 프로그램을 비교적 쉽게 개발할 수 있다. 많은 개발자들이 코드 전환이 어려운 작업이라고 생각하는 경향이 있으나 실제로는 생각보다 단순한 작업이며 자동 전환 프로그램도 개발되어 있다.

본 연구에서는 사용자 측 프로그램으로 애플릿(Applet)[3]을 사용하였으며 서버 측 프로그램으로 서블릿(Servlet)[4]을 사용하였다. 애플릿은 인터넷 브라우저 안에서 GUI(Graphic User Interface)를 구성할 수도 있고 서버와 통신할 수도 있다. 따라서, 애플릿은 사용자와 서버 측 프로그램을 연결해 주는 중간 매개체 역할을 수행한다.

애플릿이 Fig. 1과 같은 구성에 참여하려면 반드시 서버와 통신해야 한다. 그러나, 애플릿이 서버와 통신하는 일은 매우 어려운 일이었다. 자바 기반의 서버 측 프로그램인 서블릿이 등장하면서 이러한 작업이 손쉬워졌다. 서블릿은 애플릿과 같은 사용자 측 프로그램이 서버 측 자원에 접근하기 위한 통로이며 다양한 자바 API(Application Programming Interface)를 쉽게 활용할 수 있다는 것이 가장 큰 장점이다.

Fig. 2는 기존 코드를 각각 전처리, 계산, 후처리 과정으로 간단히 분리하고 이를 적절히 배치시킨 후 통신을 구현하는 방법을 간략하게 보여주고 있다. 전처리와 후처리 과정은 애플릿과 같은 사용자 프로그램으로 구현하여 사용자가 호

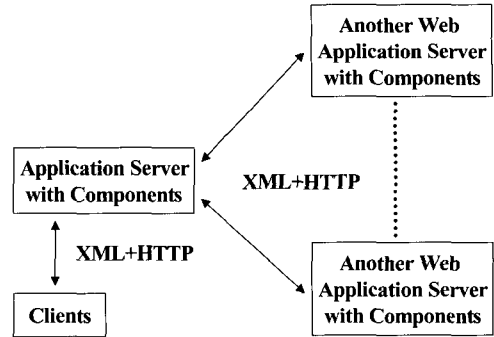


Fig. 4 A reconstructing method using cooperation of components with XML

출할 때 브라우저를 통해 공급될 수 있도록 배치하고 계산 과정은 서버 측에 배치하여 사용자의 요청에 반응할 수 있도록 하고 있다. 계산 또는 후처리 과정에 필요한 결과는 서로 간의 메시지 기반 통신을 이용하여 구현한다.

2.2 응용성을 위한 객체화

배치와 통신을 이용하는 방법의 가장 큰 단점은 구조적 프로그래밍이 갖는 특징을 그대로 가지고 있으며 이것은 결국 유지 및 보수를 어렵게 하여 새로운 기술이나 표준을 쉽게 접목시킬 수 없다는 것이다. 이러한 단점을 보완하기 위해서는 기존 코드를 전환하여 이를 기능과 속성에 따라 객체로 분리해야 한다. 이 방법은 단순히 배치시켜 통신을 구현하는 방법에 비해 많은 시간과 노력이 소요된다. 또, 기존 코드나 개발 언어에 대해 많은 기식을 필요로 하는 작업이기도 하다.

객체화 작업은 우선 배치와 통신을 이용하는 방법에서와 마찬가지로 언어를 전환해야 한다. 제대로 전환되었는지 알아보기 위해 우선 애플리케이션으로 실행하여 그 결과가 기존 프로그램과 일치하는가를 확인한다. 오류 없이 전환이 완료되었음을 확인하고 나서 전환된 코드를 기능과 속성에 따라 분리하는 작업을 해야 한다 코드를 분리하는 작업은 단순 전환작업과 달리 상당히 까다롭다. 적용하려는 구성과 기존 프로그램의 알고리즘을 자세히 이해해야 한다는데 그 어려움이 있으며 이는 많은 개발자들이 이 방법을 꺼려하는 이유이기도 하다. 기존 수치해석 프로그램에 대한 지식을 바탕으로 각각의 변수에 대해 이

해하고 유효 범위를 파악하여 그 성격을 결정해야 한다는 것이다. 즉, 자신이 기존에 알고 있던 지식 외에 추가로 새로운 지식을 습득해야 하는 부담을 안게 되는 것이다. 그러나, 이는 인터넷의 장점을 제대로 활용하기 위해 꼭 필요한 작업이다. 객체 지향의 이론적 틀은 메시지 전달이다. 객체들이 메시지를 주고받는 것으로 시스템이 동작한다. 객체는 자신의 상태를 속성으로 저장하고 있으며 객체의 기능에 해당하는 함수를 통해서 이 속성에 접근할 수 있도록 한다. 이와 같은 구조는 적절한 은폐가 이루어져 견고한 시스템을 구축할 수 있게 한다. 또한, 객체의 직렬화(serialization)[5]를 이용할 수 있어 객체의 상태를 저장하고 이를 인터넷을 통해 전달할 수 있으며 이를 다시 활성화시킬 수 있다. 객체 지향 개념은 이처럼 많은 융통성을 가지므로 성능을 향상시키기 위한 여러 가지 시도를 해볼 수 있게 한다. 위 Fig. 3은 기존 코드를 각각 속성과 기능에 따라 객체로 분리하고 이를 서버 측과 사용자 측에 적절히 배치시키는 방법을 보여주고 있다. 본 구성에서는 입력 값들을 저장한 객체, 격자 구성 객체, 전처리 과정 전반을 포함한 객체, 결과 값들을 저장한 객체 등 기능과 속성에 따라 분리된 다양한 객체가 생성되어 작동하며 각각의 객체는 쉽게 결합되어 하나의 과정을 수행할 수 있고 분리되어 인터넷을 통해 전달될 수도 있다.

객체를 전달하는 방법이 통신으로 데이터를 보내는 방법에 비해 항상 좋은 것만은 아니다. 결과 값들을 저장하는 객체와 같이 실시간으로 생산해내야 하는 데이터보다 저장해야 하는 데이터가 많은 경우에는 통신을 수행하여 전달하는 방법이 훨씬 더 효율적인 것이다. 그러나, 통신을 구현함에 있어서도 객체로 분리된 경우가 통신 관련 API를 사용하기에 용이하므로 보다 쉽게 통신을 구현할 수 있다.

2.3 재사용을 위한 컴포넌트 구현

부품을 조립하듯이 쉽고 빠르게 규모가 큰 프로그램을 개발할 수 있는 장점으로 컴포넌트 기반의 소프트웨어 개발은 각광을 받고 있다. 컴포넌트란 이미 개발된 애플리케이션 조각이며 여러 개가 조립되어 하나의 애플리케이션으로 구성될 수 있다. 컴포넌트는 근본적으로 객체에 그 근간을 두고 있으므로 일단 객체로 시스템을 구성하

게 되면 쉽게 컴포넌트로 전환하여 애플리케이션 서버의 관리를 받을 수 있다. 이의 구현을 위해 J2EE(Java 2 Enterprise Edition)[6]에서는 EJB(Enterprise JavaBeans)를 제공하며 이는 자바 RMI(Remote Method Invocation)[7]에 기반을 두고 있다. 자바 RMI는 자바 기술이 제공하는 분산 객체 기술로 네트워크 상에 떨어져 있는 객체의 함수를 자신의 시스템에 있는 것처럼 호출할 수 있게 한다. Fig. 4는 컴포넌트간의 협업을 위해 XML을 사용하는 방법을 보여주고 있다. XML은 애플리케이션에 종속되지 않은 표준 데이터 양식을 제공함으로써 시스템 개발을 용이하게 한다. 사용자의 요청을 받은 애플리케이션 서버는 다른 공급자와의 협업을 통해 계산을 수행하고 그 결과를 사용자에게 전달하게 된다. 이 과정에서 개발자가 구현하기 어려운 많은 부분을 애플리케이션 서버가 대신 담당하여 수행하게 된다. 이는 규모가 큰 프로그램에 적당한 방식으로 작은 프로그램에 적용하는데는 무리가 있다. 따라서, 본 연구에서는 실제 예제로 구현하지 않았다.

3. 재구성된 예제 프로그램

본 연구에서는 실제 사용되고 있는 프로그램을 재구성하여 구현해 봄으로써 그 가능성을 확인하였다. 예제에는 자유표면을 포함하는 2차원 비정상 유동을 해석하는 전산유체역학 프로그램인 SOLA-VOF(SOLution Alorithm for Volume of Fluid)[8]를 사용하였다. 배치와 통신을 이용하는 방법과 객체로 분리하는 방법으로 재구성하였으며 실제 브라우저를 통해 인터넷에서 수행될 수 있도록 하였다. 객체화를 이용한 예제 프로그램은 여러 개의 GUI 관련 객체들과 입력 값을 저장한 객체, 격자를 구성하고 그 정보를 저장한 객체, 전처리 과정을 수행하고 그 결과를 저장하는 객체, 계산을 수행하는 객체, 결과 값을 저장한 객체 등으로 구성하였다. 추후 웹 컴포넌트 계층에서 수행 성능을 향상시키거나 컴포넌트 기반의 웹 서비스를 구현하려면 계산 과정을 좀 더 세밀하게 분리하여야 할 필요가 있다.

3.1 기존 프로그램

SOLA-VOF는 자유표면의 움직임을 고정된

격자만으로 해석할 수 있게 하여 적은 기억 용량으로 처리할 수 있게 되어 있다. 이때 각 격자를 차지하는 유체의 부피를 나타내는 용적함수를 별도로 계산함으로써 자유 표면의 움직임을 감지할 수 있게 한다. 포트란으로 작성되어 있으며 개별적인 임무를 수행하는 서브루틴으로 분리되어 있다. 그러나, 각 서브루틴은 공통 변수를 사용함으로써 분리를 어렵게 하고 있다. 이는 기존의 거의 모든 코드들이 갖는 특징이라고 할 수 있다.

3.2 재구성된 프로그램

배치와 통신을 이용한 방법과 객체화하여 응용성을 부여한 방법은 겉으로 그 차이점이 드러나지 않으며 내부적으로만 차이를 갖게 된다. 모든 방법은 동일하게 사용자가 브라우저를 통하여 프로그램에 접근해서 전처리 과정, 계산 과정, 후처리 과정을 수행하게 된다.

사용자 측 프로그램의 GUI는 JFC(Java Foundation Class)[9]와 자바 2D[10]를 사용하였다. 초기에 자바가 기본적으로 사용하던 GUI인 AWT(Abstract Window Toolkit)는 응용 프로그램에 사용하기에 적당하지 않았다. 그러나, 순수한 자바 GUI를 만들기 위해 여러 업체들의 공동 작업으로 JFC가 등장하면서 자바가 일반 응용 프로그램에 사용되는데 부족함이 없음을 보여 주었다. 자바 2D 역시 고난도의 그래픽을 쉽게 생성할 수 있도록 하는 여러 가지 클래스들을 가지고 있다.

3.2.1 전처리 과정

사용자가 브라우저를 통해 예제 프로그램에 접근하게 되면 전처리 과정을 위한 GUI를 볼 수 있다. 기본 값으로 격자를 생성해 놓았으며 마우스로 해당 위치의 격자 색인을 확인할 수 있도록 구성하였다. 전처리 과정을 수행하고 서버에 계산을 요청하면 배치와 통신을 이용한 방법에서는 HTTP 요청을 통해 데이터를 서버 측 프로그램에 전송하게 되고 객체를 이용하는 방법에서는 입력 객체를 직렬화하여 서버 측 프로그램에 전송하게 된다.

3.2.2 계산 과정

요청을 받은 서버 측 프로그램은 하나의 쓰레드를 할당하여 계산을 수행한다. 이 과정에서 사용자 측 프로그램은 서버 측 프로그램과 통신하며 계산과정을 지켜볼 수 있도록 하는 객체가 생

성된다. 이를 통해 사용자는 계산의 진행 정도, 수렴성 등을 확인할 수가 있다. 또, 사용자 측에서 서버 측에 계산을 중단하도록 요청할 수도 있게 하였으며 서버 측 계산이 중단되거나 완료되면 이 사실을 사용자가 확인할 수 있도록 구성하였다.

배치와 통신을 이용한 방법에서는 문자열 분석을 통해 사용자의 입력 값을 추출하는 작업이 이루어지고 객체화를 이용한 방법에서는 객체를 추출하여 바로 계산에 사용할 수 있도록 활성화한다. 객체화를 이용한 방법에서 실제로는 전처리 과정에서 계산 과정 클래스의 생성자 인수인 전처리 과정 클래스의 객체가 생성되고 사용자의 입력 값에 의해 전처리 객체의 모든 정보를 설정하게 된다. 그러나, 이 객체는 매우 많은 정보를 갖게 되어 크기가 커지므로 인터넷을 통한 전송 과정에 비효율적이다. 따라서 본 예제 프로그램에서는 사용자의 입력 값만을 담은 입력 정보 객체를 전송하도록 하였고 서버 측 프로그램에서 이를 인수로 받는 격자 구성 클래스의 객체부터 다시 생성하도록 하였다.

3.2.3 후처리 과정

계산이 완료되었을 때 사용자가 결과를 요청하면 서버 측 프로그램은 사용자 측 프로그램에 결과를 전송하게 된다. 배치와 통신을 이용한 방법에서는 결과를 문자열로 전송하게 되며 객체를 이용한 방법에서는 결과를 객체에 저장하고 이를 직렬화하여 전송하게 된다.

결과를 전송 받으면 사용자 측 프로그램의 GUI는 후처리 과정을 위한 GUI로 변경된다. 이를 통해 속도 벡터의 분포, 유체 및 압력 분포, 유선 등을 확인할 수 있도록 하였으며 유동 현상을 동영상으로도 확인할 수 있도록 GUI를 구성하였다.

4. 결과 및 고찰

본 연구에서와 같이 기존의 수치해석 프로그램을 인터넷 안에서 실행할 수 있도록 하는 것은 생각보다 쉬운 일이다. 단지 프로그램을 인터넷 브라우저 및 웹 서버가 수용할 수 있도록 전환하여 적절히 배치하고 그들간의 통신을 구현하면 가능한 일이다. 그러나, 이러한 방식은 좀 더 나은 형태의 프로그램 개발을 위해 지양되어야 한

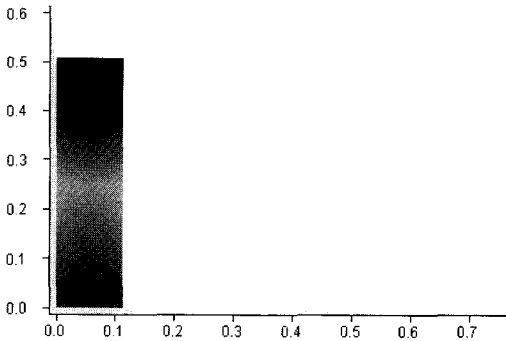


Fig. 5 Fluid distribution for broken dam problem at time 0.0

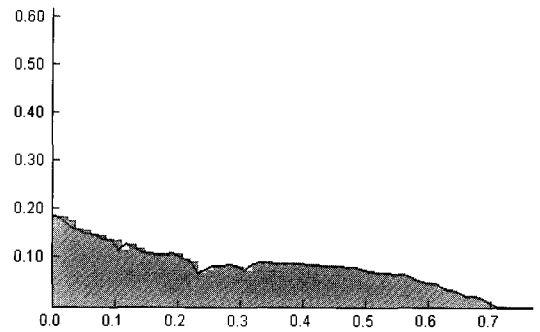


Fig. 6 Fluid distribution predicted by the original program for broken dam problem at time 1.0

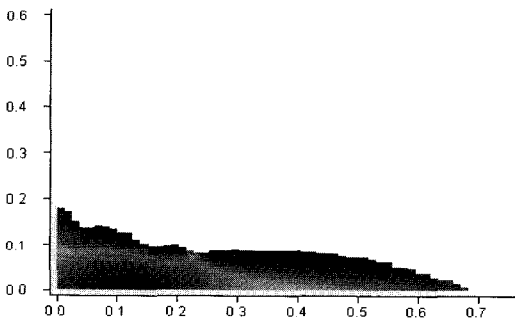


Fig. 7 Fluid distribution predicted by the reconstructed program for broken dam problem at time 1.0

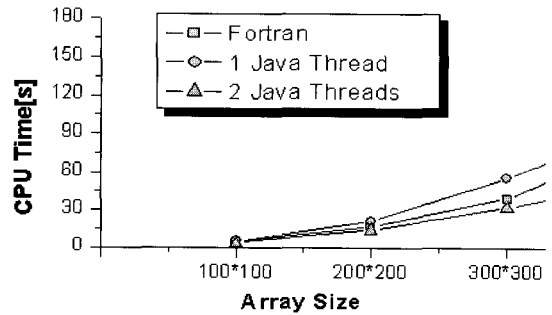


Fig. 8 Performance test of Fortran, a Java thread and two Java threads

다. 단순히 전화하여 재배포하고 서로간의 통신을 구현하게 되면 구조적 프로그램 개발 기법이 갖는 단점을 그대로 떠 안아 결국 더 이상 발전하지 못하는 인터넷 프로그램으로 남게 될 것이다. 본 연구에서 이러한 방식으로 실제 예제를 구현해 보면서 인터넷을 통해 프로그램을 실행하고 결과를 확인할 수 있다는 것 외에 인터넷의 장점을 좀 더 활용하고자 하는 어떠한 시도도 해볼 수가 없었다. 반면, 객체로 분리하여 재배포하는 방식은 구조적 프로그램 개발 기법이 갖는 단점을 개선하였으며 인터넷의 장점을 최대한 활용하기에 충분한 융통성을 가지고 있었다. 특히, 각각의 객체가 배치될 위치의 선정이 자유로웠으며 다양한 API를 쉽게 접목시킬 수 있어 개발 기간 내내 많은 욕심을 갖게 하였다.

4.1 재구성된 프로그램의 검증

재구성을 가장 어렵게 하는 것은 변수의 유효영역이다. 객체를 생성하는 클래스들로 이루어진 프로그램은 여러 개의 객체가 한 개의 클래스

로부터 생성되기도 하고 한 개의 객체 안에 다양한 객체를 포함하기도 한다. 이들의 속성은 그 성격에 따라 공유되기도 하고 자신만의 속성을 갖기도 하므로 변수의 영역을 제대로 구분하는 것이 매우 어려울 때가 많다. 게다가 객체간 속성을 전달함에 있어서도 그 참조를 전달하는지 또는 그 값을 전달하는지가 불분명한 경우가 많이 발생한다. 따라서, 코드의 전환 후에 일치되는 결과를 얻었다 하더라도 재구성 과정에서 오류가 발생할 위험이 있다. 다음 Fig. 5, 6, 7은 재구성된 프로그램의 결과를 기존 프로그램의 결과와 비교한 것으로 예제의 재구성에 오류가 없었음을 보여주고 있다. 무너지는 물기둥을 2차원으로 해석한 것이며 각각 초기 유체의 분포, 시간 1.0에서 각 프로그램의 유체 분포를 보여주고 있다. 그림에 미세한 차이가 있는 것은 출력 시점의 미미한 차이와 가시화 기법 및 방법의 차이에 의한 것으로 실제 결과는 일치하고 있음을 알 수 있다.

4.2 문제점 및 보완 사항

자바나 XML이 갖는 가장 큰 장점은 운용의 효율성이다. 그러나, 이러한 효율성을 얻기 위해 수행 성능에서는 약간의 손해를 보게 된다. 자바와 XML이 성공한 기술로 남을 수 있었던 이유는 현재의 컴퓨터 성능이 이를 보충해 줄 수 있었기 때문이었다. 본 연구에서도 자바와 XML을 사용함으로써 추가되는 수행 시간을 확인할 수 있었다. 게다가 인터넷 전송 속도에 의존하는 통신 시간이 포함되므로 이는 결코 무시할 수 없는 것이다. 이를 보완하기 위해서는 다중 쓰레드와 RMI 또는 RPC(Remote Procedure Call)와 같은 분산 객체 기술이 사용되어야 한다.

Fig. 8은 포트란 언어와 자바 언어로 개발된 프로그램간의 수행 성능 차이를 보여주고 있다. 똑같은 조건에서 같은 계산식을 사용하여 배열의 값을 갱신하는 방법으로 확인하였다. 결과에서 보듯이 자바 언어로 개발된 프로그램의 수행 성능이 조금 뒤지지만 다중 쓰레드(multi-thread)[11]를 활용하여 이를 극복할 수 있고 오히려 좋은 성능을 발휘할 수 있었다. 자바는 다른 언어와 달리 쓰레드를 언어 자체에서 지원함으로써 HPC(High Performance Computing) 구현에 많이 활용되고 있다. Fig. 1에서 제시했던 구성의 웹 컴포넌트 계층에서 이와 같은 기술을 활용하여 수행 성능을 향상시키기 위한 연구가 이루어져야 할 것이다.

4. 결 론

본 연구에서는 수치해석에 인터넷을 활용하기 위한 구성을 제안하고 이를 위한 기존 프로그램의 재구성 방법을 몇 가지 제시하였다. 또, 일부를 실제 구현하여 실행해 봄으로써 그 가능성을 확인하였다. 본 연구를 통해 기능과 속성에 따라 객체로 분리하는 방법이 배치와 통신을 이용한 방법보다 구현에 어려움이 있으나 인터넷의 장점을 제대로 활용하기 위해 충분한 융통성을 갖는

다는 것을 확인하였다. 이러한 융통성은 수행 성능을 향상시키기 위한 연구에 도움이 되며 이와 같은 작업은 곧 공학용 웹 서비스 개발에 수치해석을 참여시키기 위한 준비 과정이 될 것이다.

후기

이 논문은 2000년도 학술진흥재단의 지원에 의해 연구되었습니다.(KRF-2000-E1104-E00006)

참고문헌

- [1] 송희용, 고성호, "인트라넷 안에서 효율적인 수치해석을 구현하기 위한 자바기술," 대한 기계학회 춘계학술대회논문집E (2001), p.488.
- [2] E. Armstrong, et al., The Java™ Web Services Tutorial, Sun Microsystems, Inc. (2002).
- [3] 황정현 역, 자바 2 언리쉬드, 도서출판 대림 (1999), p.85.
- [4] 최환진 역, JAVA 서블릿 프로그래밍, 한빛미디어 (1999), p.37.
- [5] 조현일 역, 자바 I/O, 한빛미디어 (1999), p.327.
- [6] S. Bodoff, et al., The J2EETM Tutorial, Sun Microsystems, Inc. (2002).
- [7] 광용재 역, 자바를 사용한 분산 프로그래밍, 인포북 (2001), p.175.
- [8] B. D. Nichols, et al., "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," J. Comput. Phys. Vol. 39 (1981), p.201.
- [9] 류광 역, The JFC Swing Tutorial, 정보문화사 (2000).
- [10] 정용우, 추형준 역, 자바 2D 그래픽스, 한빛미디어 (2000).
- [11] 진장일, 박성철 역, 자바 쓰레드, 한빛미디어 (2001), p.289.