

개발 생산성 향상을 위한 분산 시스템 기반의 프레임워크 설계

최병하* 최승교** 조경산***

A Design of Framework based on Distributed System for Enhancing Productivity

Byungha Choi* Sung-kyo Choi** Kyungsan Cho***

요약

본 논문은 분산 시스템 기반에서 경량급 프레임워크의 간단한 개발 주기를 지원하며 분산 처리가 가능한 IPO(Input-Process-Output) 모델 기반의 프레임워크를 제안한다. 제안 프레임워크는 입력 함수, 처리 함수, 출력 함수를 가진 클래스를 개발하면 분산 시스템의 표현 계층과 중간 계층에서 하나의 응용 프로그램으로 실행되도록 지원해준다. 또한 이 기법의 특징으로 응용 프로그램 개발시 분산 시스템 의존적인 부분을 제거하며, 개발-구축-배치-검증의 개발 주기를 개발-검증 주기로 감소시킨다. 이 기법의 생산성 향상과 성능 검증하기 위해 기존의 분산 시스템과 경량급 프레임워크의 응용 프로그램을 제안 기법의 응용 프로그램과 비교 분석한다. 그 결과 상당한 코드 감소와 분산 시스템 의존적인 부분이 감소하여 개발 생산성을 향상시키고 경량급 시스템에 비해 상대적으로 안정적인 성능을 보인다.

▶ Keyword : 분산 객체 시스템, 경량급 프레임워크, XML 웹 서비스, IPO 모델, 개발주기

Abstract

In this paper, we propose a framework which supports the simple development cycle based on Distributed System using IPO(Input-Process-Output) model. In our proposed framework, once a single class with Input, Process and Output functions is developed, it can be executed like a single application in both presentation layer and middle layer of a distributed system. Our proposed

· 제1저자 : 최병하 교신저자 : 조경산

· 투고일 : 2011. 10. 20. 심사일 : 2011. 11. 05. 게재확정일 : 2011. 11. 15.

* 단국대학교 일반대학원 컴퓨터학과(Dept. of Computer, Dankook University)

** 강원대학교 컴퓨터공학과 교수(Dept. of Computer Engineering, Kangwon University)

** 단국대학교 소프트웨어학과 교수(Dept. of Software Science, Dankook University)

※ 본 과제는 정보통신산업진흥원의 SW공학 요소기술 연구개발사업의 결과물임을 밝힙니다.

Framework removes dependency on distributed system and makes development cycle decreased from Develop-Build-Deploy-Test cycle to Develop-Test cycle when application programs is developed. The verification of productivity and performance is performed by comparing our Framework with distributed system or lightweight Framework. The results show high productivity due to the considerable amount of reduction in lines of source code and reduction in dependency on distributed system, and show stabler performance than lightweight framework.

▶ Keyword : Distributed Object System, Lightweight framework, XML Web Service, IPO Model, Development Cycle

1. 서론

현존하는 대부분의 응용 프로그램은 입력(Input), 처리(Process), 출력(Output)으로 구성된 IPO (Input-Process-Output) 모델로 설계될 수 있다[1]. 이를 이용한 최근의 응용 프로그램은 다양한 시스템 환경과 프로그래밍 언어로 그 구조가 복잡해지고 대형화 되어 대규모 기업용 응용 프로그램에 이르러서는 복잡한 처리 과정을 분산하여 과부하를 감소시키며 성격이 다른 다양한 응용 프로그램의 처리를 위해 원격 시스템과 협동작업의 필요성까지 대두되었다.

초기의 원격처리는 RPC(Remote Procedure Call)로 처리했으나 원인 미상의 예외와 병렬성이 결여된 순차적 동기화의 단점으로 최근에는 원격지의 객체를 동일한 시스템의 객체처럼 호출하는 편리한 기능을 가진 분산 객체 시스템으로 발달하게 되었다. 분산 객체 시스템은 입출력을 담당하는 표현 계층(Presentation Tier), 업무를 처리하는 중간 계층(Middle Tier), 데이터를 저장하는 데이터 계층(Data Tier)의 논리적인 3계층으로 나눌 수 있으며[2], 견고한 설계를 지원하여 응용 프로그램을 명확한 계층적인 모듈로 개발할 수 있다. 또한 중간 계층의 트랜잭션, 메모리 관리 등 자원 관리의 효율성을 지원하는 장점도 있다. 그러나 복잡한 다단계 개발 주기 즉, 개발(Develop), 구축(Build), 배치(Deploy), 검증(Test) 등으로 개발과 유지보수에 분산 객체 시스템에 숙달된 개발자가 요구되지만 신속한 개발과 변경이 용이하지 않아 유지보수 비용이 과도하게 든다는 단점이 있다[3]. 또한 최근 SOA (Service Oriented Architecture)라는 서비스 위주의 개념으로 분산 시스템에 적용할 수 있는 XML 웹서비스가 제안되었으며 분산 객체 시스템 보다 유연하지만 개발시에 동일한 단점이 존재한다. 이들 단점을 해결하기 위해 분산 시스템을 사용하지 않고 간단한 개발 주기와 편리한 API로 신속하고 우수한 개발 생산성을 보여주는 벨로시티(Velocity), 스트러츠(Struts), 스프링(Spring),

하이버네이트(Hibernate)의 경량급(lightweight) 프레임워크가 각각 제안되었다[4, 5]. 그러나 이들은 원격처리 기능이 없는 한계를 가져 대규모의 응용 프로그램 개발에서 분산 객체 시스템과 함께 사용되는 기법도 제시되었으나[6], 이는 분산 객체 시스템의 단점을 상쇄시키지 못한다.

대규모 응용 프로그램 개발시에는 분산 객체 시스템의 원격 처리와 명확한 계층적인 모듈화, 효과적인 자원관리 등의 장점을 추구하는 동시에 경량급 프레임워크가 제공하는 개발의 신속성과 편의성을 함께 충족시켜 개발 생산성을 향상시킬 필요가 있다.

본 연구는 금융 시스템 같은 복잡하고 대규모 응용 프로그램의 개발을 위해 명료한 모듈화 및 견고한 설계를 지원하는 분산 객체 시스템 기반이면서도 동시에 경량급 프레임워크의 장점인 간단하고 신속한 개발과 검증의 2 단계 개발 주기를 지원해 주는 IPO 프레임워크 설계 기법을 제안한다. 제안 기법에서는 하나의 클래스에 입력, 처리, 출력의 3가지 함수를 갖는 응용 프로그램을 구현하면 IPO 프레임워크는 대규모 분산 객체 환경에서 서버 설정이나 배치 등이 없이 간단한 개발 주기로 응용 프로그램을 개발과 유지보수가 가능하여 개발 생산성을 향상시키는 기법이다.

본 연구에서는 개발생산성과 안정적인 성능을 검증한다. 개발 생산성을 검증하기 위해 기존 분산 시스템으로 개발된 금융 위기 조기 경보 시스템과 3가지 다른 파생상품 가격 결정 시스템들을 제안 기법과 비교 분석한다. 구축과 배치의 생략으로 신속하고 효율적인 개발, 그리고 소스코드의 감소와 클래스의 재사용등으로 기존 분산 처리 기법에 비해 개발 생산성의 향상을 보인다. 또한 경량급 프레임워크와 성능 비교 분석을 통해 분산 처리 기법의 자원 관리 효율성과 안정성을 그대로 사용할 수 있다.

본 논문의 구성은 다음과 같다. 2장은 관련연구로 분산 객체 시스템과 기존의 프레임워크를 분석하여 개선된 프레임워크의 요구사항을 제시한다. 3장에서는 2장의 분석의 요구사항을 충족하는 프레임워크를 제안한다. 4장에서는 제안 프

레이아웃을 XML 웹 서비스에 적용하여 기존 기법과 비교 분석하여 검증하고 요구사항의 충족여부를 제시한다, 5장은 결론으로 본 논문을 마무리 짓는다.

II. 관련 연구

본장에서는 대규모 응용 프로그램 개발을 위해 제안된 분산 시스템과 이의 단점으로 제안된 경량급 프레임워크의 구현 기법을 분석한다. 이들을 기반으로 개선된 프레임워크의 요구사항을 제안한다.

1. 분산 객체 시스템의 구현 기법

객체 지향적 프로그래밍을 분산된 원격 시스템에 적용할 수 있도록 한 분산 객체 시스템의 널리 알려진 기법으로는 CORBA(Common Object Request Broker Architecture), 자바의 EJB (Enterprise Java Bean)가 있다. 최근에는 객체 지향보다 SOA(Service Of Architecture)의 개념을 이용한 분산 시스템인 XML 웹 서비스 등이 제시되었으며, 그림 1과 같은 논리적인 3계층 구조가 대표적인 예이다[2,7,8,9].

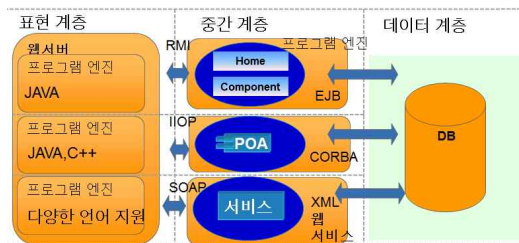


그림 1. 분산 객체 시스템의 구조
Fig. 1. Architecture of Distributed Object System

이들은 이기종 또는 동종의 분산된 환경에서 원격 객체의 생성, 소멸, 트랜잭션, 메모리 관리 등 자원 관리의 효율성을 지원하는 장점이 있지만, 각 기법에 따른 계층간 통신을 위해 인터페이스 정의가 필요하고 이를 이용하여 원격 객체의 호출에 사용되는 스텝(stub), 스켈레톤(skeleton)을 생성하여 서버에 설정해주는 배치가 필요하다[7,8,9,10].

그러므로 CORBA, EJB, XML 웹 서비스는 모두 배치라는 과정으로 4단계의 개발주기가 되고, 표현 계층과 중간 계층의 응용 프로그램을 각각 개발해야 된다. 특히 중간 계층은 각 아키텍처 구조에 따른 설정, 인터페이스 정의 등의 분산 시스템의 의존적인 부분은 개발을 더 어렵게 한다. 하나의 서비스를 위해 개발 비용과 개발 시간이 과도하게 소요되

어 결국 개발 생산성 저하 및 품질 저하가 발생된다.

2. 경량급 프레임워크의 구현 기법

앞서 분석된 분산 시스템이 갖는 다단계의 개발 주기와 분산 시스템의 의존적인 부분 등으로 유발된 개발 생산성의 취약성에 대해 편리한 API와 간단한 개발주기를 함께 지원하고 우수한 개발 생산성을 제공하는 프레임워크가 제안되었다[4,5].

프레임워크는 응용 프로그램의 생성에 적용되어 고객의 요구에 신속히 제작할 수 있도록 미리 만들어진 미완성의 프로그램을 의미하며 모듈화, 재사용성, 확장성을 지원한다. 또한 프레임워크가 제시하는 코드 규칙이나 인터페이스에 의하여 응용 프로그램의 제어권을 프레임워크에 위임하는 역제어(Inversion of Control)를 용이하게 해주는 특징이 있다[11].

경량급 프레임워크는 단순히 가볍고 빠르다는 의미보다는 분산 시스템에 의존적인 별도의 인터페이스 정의와 설정 등이 없고, 독립적으로 실행 가능하며, 구동이 빠르고, 배치를 위한 복잡한 과정이 없다는 의미를 가진다[3].

분산 시스템을 대체하는 경량급 프레임워크의 예로써 벨로시티, 스트러츠, 스프링, 하이버네이트라는 프레임워크를 계층별로 구성하는 기법이 제안되었다[12,13,14]. 벨로시티와 스트러츠는 표현 계층을 위한 것이며, 스프링과 하이버네이트는 중간계층을 대체하여 단독으로 또는 혼용하여 각각 사용 가능하다. 이들은 개발과 검증라는 간단한 2 단계의 주기와 편리한 API로 신속하게 개발할 수 있지만 분산 처리가 되지 않아 자원 관리나 트랜잭션 관리 등에서 과부하가 발생할 수 있으며 표준이 정해져 있지 않았다는 단점이 있다.

이를 위해 분산 시스템과 경량급 프레임워크를 같이 사용하는 기법도 제시되었다[6]. 이 기법은 중간 계층에 하이버네이트 프레임워크를 삽입하여 소스코드의 개발편의성은 제공하지만, 분산 시스템과 동일한 4 단계의 개발주기를 가지므로 개발 생산성과 품질 저하가 발생할 수 있다.

3. 개선된 프레임워크의 요구사항

앞에서 제시된 분산 시스템과 경량급 프레임워크의 장점을 분석하면 분산 환경에서 성능과 개발생산성이 개선된 프레임워크의 요구사항이 제시될 수 있다.

첫째, 프레임워크의 특징인 모듈화, 재사용성, 확장성, 역제어를 지원하는 프레임워크이어야 한다.

둘째, 경량급 프레임워크의 장점인 간단한 개발주기와 분산 시스템에 의존적인 인터페이스, 설정등의 생략으로 효율적인 개발 생산성을 가져야 한다.

셋째, 과부하를 대비하여 분산 처리가 가능하며 분산 시스

템의 장점인 안정적인 성능을 가져야 한다.

III. IPO 프레임워크 기법 제안

본 장에서는 앞장의 제안 요구사항을 기반으로 분산 환경에서 경량급 프레임워크의 장점을 가진 IPO 프레임워크를 제안한다.

1. IPO 프레임워크의 전체 구조 제안

기존의 분산 시스템은 표현 계층의 클라이언트 응용 프로그램과 중간 계층의 분산 객체 응용 프로그램을 별도로 개발해야 된다. 그러나 제안 IPO 프레임워크는 응용 프로그램의 배치 없이 표현 계층과 중간 계층의 2 계층에 걸쳐 하나의 응용 프로그램으로 그림 2와 같이 실행되도록 한다. 즉 개발자는 계층과 무관하게 하나의 응용 프로그램으로 개발하여 생산성을 높일 수 있다.

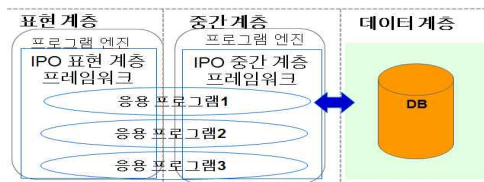


그림 2 IPO 프레임워크 이용한 응용 프로그램
Fig. 2. Application Program Using IPO Framework

제안된 구조의 특징은 다음과 같다.

첫째, IPO 프레임워크가 제공하는 공통 분산 객체만 배치하고 원격 처리 작업은 그림 3에서 설명되는 공통 클라이언트와 공통 분산 객체를 통해서만 이루어진다.

둘째, 동일한 응용 프로그램 클래스를 표현 계층과 중간 계층에 삽입하거나 공유하게 한다.

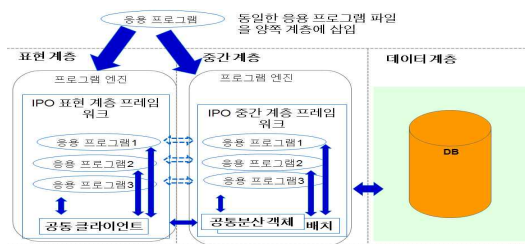


그림 3 IPO 프레임워크의 원격처리
Fig. 3. Remote Processing of IPO Framework

공통 분산 객체와 공통 클라이언트를 통해 표현 계층과 중간 계층의 동일한 클래스들 간에 호출하고 응답하게 하면 실

제 처리는 원격으로 그림 3처럼 이루어지나, 그림 2와 같이 응용 프로그램 자신의 함수를 호출하는 것으로 실행된다. 즉 IPO 프레임워크는 논리적으로는 하나의 응용 프로그램이 표현 계층과 중간 계층에 걸쳐 존재하는 실행 환경을 제공해준다.

2. 응용 프로그램 개발과 실행 과정

IPO 프레임워크를 이용하는 응용 프로그램을 개발하려면 IPO 프레임워크가 제공하는 IPOBased 클래스를 상속받아야 한다. IPOBased는 입력과 처리 그리고 출력 함수가 정의만 되어있고 내용이 없는 함수로 선언된 추상 클래스(Abstract Class)이다. 공통 분산 객체는 IPOBased를 실행시키는데, 이를 상속한 개발자의 클래스는 객체지향의 특성 중 다형성에 의해 공통 분산 객체가 IPOBased로 인식되게 할 수 있다. 그러므로 공통 분산 객체는 그 새로운 응용 프로그램의 클래스가 이미 배치된 IPOBased 클래스로 판단하기 때문에 그에 따른 배치가 필요 없어진다. IPOBased를 상속한 응용 프로그램의 재정의(overriding)에 관하여, 입력 함수는 IPO 프레임워크가 정리한 입력값을 필요시에 추가하거나 수정하는 내용을 구현하고, 처리 함수는 응용 프로그램이 실행할 내용인 DB 연동이나 업무에 관련된 내용을 구현한다. 출력 함수는 처리 함수가 처리한 결과를 필요시 수정하거나 추가하는 내용으로 구현한다.

IPO 프레임워크에서 개발된 응용 프로그램은 그림 4와 같은 실행 단계를 가진다. 즉 IPO 프레임워크가 외부에서 들어온 입력값을 정리하여 동적으로 해당 응용 프로그램의 클래스를 호출하고 입력함수로 입력값을 넘겨주면서 실행시킨다. 입력 함수의 결과 값을 공통 클라이언트와 공통 분산 객체를 통해 중간 계층에 전달하여 표현 계층과 동일한 중간 계층의 해당 클래스를 동적으로 호출하여 처리함수를 실행시킨다. 다시 처리 함수의 결과 값을 공통 분산 객체와 클라이언트를 통해 표현 계층에 넘겨 출력함수를 실행시킨다. 이때 IPO 프레임워크의 입출력과 처리 함수의 인자들은 {키, 값} 형태의 배열로 계층 간에 전달한다.

제안된 프레임워크는 원격 호출이나 인터페이스 정의 등에 관련된 분산 시스템의 의존적인 부분은 IPO 프레임워크에 위임하고, 개발자는 IPO에 관련된 함수만 구현한다. 또한 기존 기법처럼 표현 및 중간 계층의 응용 프로그램을 각각 개발할 필요가 없으며 공통 분산 객체와 클라이언트를 재사용함으로써 응용 프로그램에 대한 배치의 과정이 생략되며 배치를 위한 구축 과정도 제거된다. 그러므로 분산 시스템에서 개발과 검증의 개발주기도 응용 프로그램이 개발 가능해진다. 또한 분산 처리가 가능하므로 경량급 프레임워크보다 더 안정적

인 성능을 나타낸다.

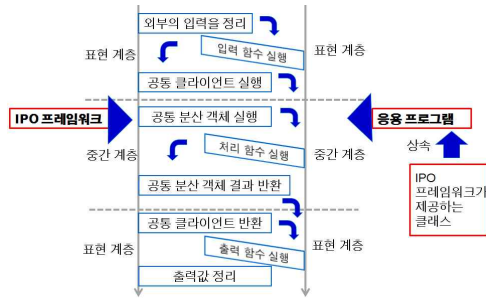


그림 4. IPO 프레임워크와 응용 프로그램의 실행과정
Fig. 4. Workflow for IPO Framework and Application Program

표 4. 시스템 사양
Table 1. Details of System

항목	내용
운영체제	MS Vista Business K 64bit
Java Developer Kit	IBM JDK 1.6
WAS	IBM Websphere 7.0
표현계층의 프로그램 엔진	Websphere 7.0의 Servlet Container
중간계층의 프로그램 엔진	SOAP Engine - IBM Websphere JAX-WS RI 2.1.1(SOAP v 1.1)
데이터 계층	Oracle Database 10g Express
응용 프로그램	금융 조기경보 시스템, 파생상품 가격 결정 시스템 3가지 DB 데이터 출력 테스트 응용 프로그램

IV IPO 프레임워크 검증

본 장에서는 분산 환경의 XML 웹 서비스 기반 위에서 기존 분산 처리 기법과 제안 기법의 응용 프로그램을 비교 분석하여 개발생산성을 검증하고, 또한 기존 경량급 프레임워크의 기법과 제안 기법에 대한 과부하 테스트를 비교 분석하여 제안 기법의 안정성과 견고한 성능을 검증한다.

1. IPO 프레임워크 구현

XML 웹서비스의 사양은 표1과 같으며 이를 이용하여 표현 계층, 중간 계층, 데이터 계층을 구축한다.

그리고 표 1의 시스템에 기반하여 IPO 프레임워크는 3장의 제안 기법에 따라 XML 웹 서비스 기반 위에 표현 계층에는 웹에서 작동하는 자바 클래스인 서블릿(Servlet)으로, 중간 계층에는 공동 분산 객체를 공동 분산 객체로 배치하여 IPO 프레임워크를 그림 5와 같이 구성한다.

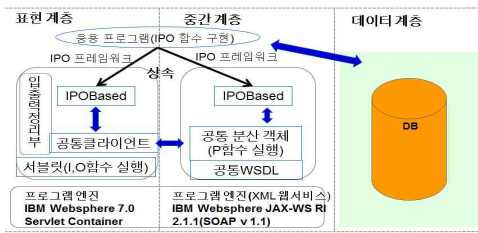


그림 5. IPO 프레임워크의 구조
Fig. 5. IPO Framework Architecture

2. 검증을 위한 응용 프로그램 구현

제안 프레임워크의 개발 생산성 검증을 위한 응용 프로그램은 금융 위기 조기 경보 시스템과 3가지 파생상품 가격 결정 서비스의 총 4개의 응용 프로그램으로 기존 분산 시스템 기법과 IPO프레임기법의 2가지로 구현하고, 안정적인 성능을 검증을 위한 응용 프로그램은 DB에서 특정 내용을 출력하는 응용 프로그램으로 스프링과 IPO 프레임워크의 2가지로 구현한다.

금융 위기 조기 경보 시스템은 주가 지수를 기반으로 인공 신경망 알고리즘으로 주가 지수 등락률, 등락률의 이동 분산 등으로 학습하고 이를 바탕으로 금융 위기를 예측하는 시스템이다. 또한 파생상품 가격 결정 시스템은 기초자산에 대한 최적 포트폴리오를 선택하여 미래의 기초자산 가격의 변동성 위험을 줄이기 위하여 3가지 기법의 응용 프로그램들이 배치된다. 몬테카를로(MonteCarlo) 시뮬레이션, 블랙쇼울즈(BlackScholes), 이항모형(Binomial Model) 방법론의 3개의 응용 프로그램이며 이들은 그림 6과 같이 왼쪽의 기존 기법과 오른쪽의 IPO 프레임워크 기법의 2가지로 3 계층으로 구현한다.

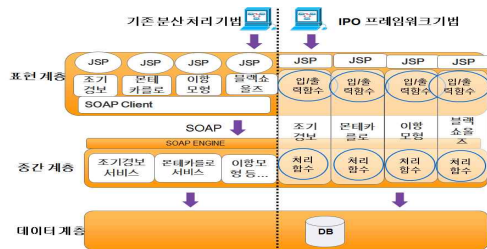


그림 6. 기존 분산 처리 기법과 IPO 프레임워크의 응용 프로그램 구조

Fig. 6. Application Programs of Existing Scheme in Distributed System and IPO Framework

기존 분산 환경의 기법은 데이터 계층의 DB는 금융 위기 조기 경보 시스템에 대한 데이터가 입력되어 있고 중간 계층은 조기 경보 시스템과 파생상품 가격 결정 서비스의 3가지 응용 프로그램으로 배치된다. 표현 계층은 원격 호출을 위한 SOAP Client, 데이터를 처리하는 서블릿과 HTML로 생성하는 JSP(Java Server Page)로 구현된다.

IPO 프레임워크로 개발된 응용 프로그램은 그림 6의 오른쪽 부분과 같으며 이들은 하나의 클래스로 입력, 처리, 출력의 함수를 구현하여 표현 계층, 중간 계층에 걸쳐 실행된다. 출력 함수로 생성된 출력값을 HTML 문서로 생성하기 위해 JSP 파일에서 HTML 태그와 함께 처리한다.

또한 성능 검증을 위한 응용 프로그램은 표현 계층과 데이터 계층의 2 계층으로 구성된 그림 7과 같은 경량급 프레임워크의 응용 프로그램과 3 계층의 IPO 프레임워크의 응용 프로그램 2가지를 구현한다. 이들이 처리하는 기능은 단순히 데이터베이스의 특정 내용을 JSP파일로 화면에 출력하는 기능을 가진다.

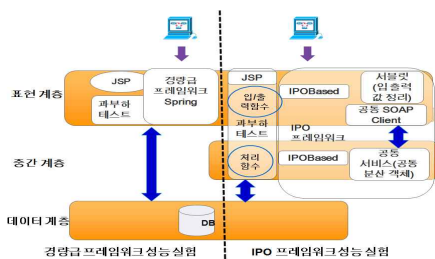


그림 7. 경량급 프레임워크와 IPO 프레임워크의 응용 프로그램

Fig. 7. Application Programs of IPO Framework and Lightweight Framework

3. IPO 프레임워크 개발생산성의 비교 분석

표 2. 1개 서비스의 응용 프로그램 개수
Table 2. Number of Application Programs per Service

항목	기존 기법	제안 프레임워크
표현 계층 응용프로그램	클래스 1개, JSP 1개	클래스 1개, JSP 1개
중간 계층 클래스	클래스 1개	

표 3. 클래스와 코드라인 감소율
Table 3. Reduction Ratio Of Classes and Line Of Code

항목	조기 경보	몬테 카를로	이항 모형	블랙 쇼울즈
클래스 감소율	50%	50%	50%	50%
코드라인 감소율	62%	40%	54%	77%

표 4. 분산 객체 환경의 의존성 비교
Table 4. Comparison For Dependency on particular environment or context

항목	기존 기법	IPO 프레임워크
배치	N1) 번	0 번
인터페이스 정의(WSDL)	N 개	0 개
개발주기	개발,구축,배치,검증	개발, 검증
분산 객체 호출 소스 코드	1회 이상	없음

1) N은 응용 프로그램의 개수

그림 6의 기존 분산 처리 기법은 각 계층에서 수행되는 응용 프로그램을 별도로 개발하지만 IPO 프레임워크 기법은 표현과 중간 계층에서 1개의 응용 프로그램을 개발하므로 서비스 1개에 대해 개발할 클래스와 JSP는 표 2와 같이 제시되며 결과적으로 표 3의 클래스와 코드라인 감소율을 나타내며 평균 58%의 코드 라인 감소율을 보인다.

또한 제안 프레임워크에 의해 분산 시스템의 환경에 의존적인 부분은 표 4처럼 배치와, 인터페이스 정의문서, 분산 객체 호출에 대한 소스 코드 등이 생략되며 개발주기도 2 단계로 단축되어 개발과정의 절차도 상당히 감소한다. 또한 IPO 프레임워크는 기본적인 프레임워크의 특징인 재사용성, 모듈화, 확장성, 역제어를 표 5와 같이 지원한다.

제안 프레임워크는 클래스와 소스코드를 감소시키며 시스템에 의존적인 부분을 제거한다. 그리고 간단한 개발주기와 기본적인 프레임워크의 특징을 제공하여 개발자에게 우수한 개발 생산성을 제공해준다.

표 5. 프레임워크의 특징 지원
Table 5. Supporting Features of Framework

재사용	표현, 중간 계층에서 재사용
모듈화	입력, 처리, 출력으로 모듈화
확장성	IPO 모델이 지원하는 응용 프로그램은 모두 개발가능
역제어	IPO 프레임워크가 제어

4. IPO 프레임워크의 성능 비교 분석

그림 7의 응용 프로그램의 구성으로 경량급 프레임워크인 스프링의 기법과 IPO 프레임워크 기법을 적용하여 성능을 비교 한다. 이는 DB에 저장된 다량의 데이터를 조회하는 응용 프로그램을 JMeter라는 과부하 생성 도구로 실험하는 것이다. 10명의 사용자가 동시에 50번씩 응용 프로그램의 URL을 요청하도록 JMeter로 수행하였으며 그 결과 표 6과 같이 제시된다.

표 6. 성능 분석
Table 6. Performance Analysis

항목	스프링	IPO 프레임워크
평균 응답 시간	125 ms	109 ms
최소 응답 시간	14 ms	12 ms
최대 응답 시간	1096 ms	125 ms

경량급 프레임워크인 스프링의 응용프로그램은 최대, 최소 응답시간의 편차가 심하다. 그러나 IPO 프레임워크의 응용 프로그램은 최대, 최소의 응답 시간의 편차가 상대적으로 적은 안정적인 결과를 나타내며 평균 응답시간도 IPO 프레임워크가 더 짧은 결과를 보인다.

5. 제안 요구 사항의 만족

제안 프레임워크 기법은 2장 3절에서 제시된 개선된 프레임워크의 제안 요구사항을 다음과 같이 만족시킨다.

첫째, IPO 프레임워크는 표 5에 제시된 대로 모듈화, 재사용성, 확장성, 역제어 등의 프레임워크 특징을 우수하게 지원한다.

둘째, IPO프레임워크는 표 3과 같이 소스코드와 클래스의 감소를 보이며, 표 4에서 제시한 대로 인터페이스 정의와 같은 분산 시스템에 의존적인 부분과 배치의 생략이 가능하며, 개발과 검증의 간단한 개발주기를 가지므로, 분산 시스템에 비해 효율적인 개발 생산성을 가진다.

셋째, IPO 프레임워크의 기법은 XML 웹 서비스를 기반

하므로 과부하를 분산 처리하여 경량급 프레임워크에 비해 표 6과 같이 안정적인 성능을 제공한다.

V 결론 및 향후 연구

본 논문에서는 기존 분산 시스템과 경량급 프레임워크의 단점들을 해결하기 위해 개선된 프레임워크가 가져야할 3가지 요구사항을 제시하고, 이를 만족시키는 개선된 프레임워크를 제안한다. 즉 제안기법은 모듈화, 재사용성, 확장성과 역제어와 같은 프레임워크의 기본적인 기능을 제공하며, XML 웹서비스의 분산 처리 기능을 가지면서도 경량급 프레임워크 처럼 간단한 개발주기와 분산 시스템에 의존성이 없는 개발을 지원한다. 기존 기법과의 비교 분석을 통해, 제안 기법은 기존 분산 처리 환경에 비해 응용 프로그램의 소스코드가 평균적으로 58%로 감소되는 우수한 개발 생산성을 나타내었으며, 경량급 프레임워크에 비해 과부하에 대해서도 안정적인 성능을 보여 신속한 대형 응용 프로그램의 개발과 유지보수에 적합한 제안임을 보였다.

그러나 제안 프레임워크는 XML 웹 서비스의 인터페이스 정의문서에서 계층간에 함수를 호출할 때 함수 인자들을 {키, 값}의 배열 형태로 사용하여 다른 형태의 데이터 구조를 적용할 수 없는 단점이 있다.

향후 연구로는 이 단점을 제거하고, Tomcat 같은 서블릿 컨테이너로 개발한 응용 프로그램을 수정없이 EJB나 XML 웹 서비스로 전환 가능한 IPO 프레임워크의 기법을 제시한다.

참고문헌

- [1] Sabah Al-F edaghi, "Diagrammatical Framework for Information System Attacks," Proc. of *Internet Technology and Secured Transactions (ICTTST)*, pp. 1-7, Nov. 2010.
- [2] Microsoft, <http://msdn.microsoft.com/en-us/library/ms685068.aspx>
- [3] Rod Johnson, Juergen Hoeller, "Expert One-on-One J2EE Development without EJB," WROX Press, 2004.
- [4] Young-Oh Kwon, Se-Hoon Lee, "Design of Open Framework Extended Module for ASP Software Development," Proc. of the Korean Society of Computer Information Conference, Vol.

- 16, No 1, pp 91-28, Jun. 2008.
- [5] Myeong Ho Lee, "Design and Implementation of Lightweight Container Architecture Using Spring 2.0 and Hibernate 3.2," Journal of The Korean Institute Of Plant Engineering, Vol. 14, No. 2, Feb. 2009.
- [6] Jae-Won Lee, Kum-Joong Kim, Man-Cheol Shin, Chul-Woo Park, Min-Uk Yang, "Study of CIM data exchange using Hibernate(J2EE)," Proc. of The Korean Institute of Electrical Engineers, pp. 60-61, Jul. 2010.
- [7] Myungnam Bae, Byungbog Lee, Aesoon Park, Inhan Lee, Naesoo Kim, "The Middleware Extension for guaranteeing the Implementation Independency between C++ and VHDL," The Institute of Electronics Engineers of Korea -TC, Vol. 46, No. 6, pp. 66-77, Jun. 2009.
- [8] Soo Dong Kim, Hyun Gi Min, Jin Yeal Lee, Seong An Kim, "An Effective Method to Design CBD Components in Enterprise JavaBeans (EJB)," Journal of KISS:Software and Applications, Vol. 33, No. 1, pp. 32-44, Jan. 2006.
- [9] Gun Min Lee, Jung-Hyun Kim, Sang Jun Park, Sung-Ho Lee, Dong-Wook Shin, Young-Wan Koo, "Design of Web Services based on SOA for provide to Realtime Services," Proc. of Korean Society for Internet Information, pp. 343-348, May. 2009.
- [10] Kyung-Young So, Jong-Goo Park, "A Study on the Modeling for Component Integration in the java Bean-based System", Journal of the Korea Society of Computer and Information, vol. 5, no.2, pp. 37-42, Jun. 2000.
- [11] G. Edwards, N. Medvidovic, "A Methodology and Framework for Creating Domain-Specific Development Infrastructures," Proc. of Automated Software Engineering, pp. 168-177, Sep. 2008.
- [12] Yong Hwan Lee, "A Flexible and Extensible CBD Application Framework and Performance Analysis," Journal of KISS : computing practices, Vol. 13, No. 1, pp. 46-58, Feb. 2007.
- [13] HaiLan Pan, AnBao Wang, WenRong Jiang, "Discussion of Course of E-commerce Website Construction based on Java EE lightweight framework," Proc. of Education Technology and Computer (ICETC), Vol 1, pp. 442-445, Jun. 2010.
- [14] Wu Oil, Dong Xin, Huang Penghe, Bai Yanl, Li Yanhui, "Design & Implementation of Devanning/Container Loading Management System Based on Lightweight Framework," Proc. of Computer Science and Information Technology (ICCSIT), pp. 446-449, Jul. 2010.

저자 소개



최 병 하

2009~현재 : 단국대학교 컴퓨터학과
(박사과정)

관심분야 : 네트워크 보안

Email : notanything@hanmail.net



최 승 교

1982 : 단국대학교 전기공학과(학사)

1992 : 단국대학교 대학원 전산통계학
과(석사)

2001 : 단국대학교 대학원 전산통계학
과(박사)

1994~현재 : 삼척/강원대학교 컴퓨터
공학과 교수

관심분야 : 컴퓨터구조, 성능평가, 시뮬
레이션

Email : skchoi@kangwon.ac.kr



조 경 산

1979 : 서울대학교 전자공학과(학사)

1981 : 한국과학기술원 전기전자공학과
(공학석사)

1988 : 텍사스 대학교(오스틴) 전기전
산공학과(Ph.D.)

1988~1990 : 삼성전자 컴퓨터부문
책임연구원, 실장

1990~현재 : 단국대학교 컴퓨터학부
교수

관심분야 : 네트워크시스템 및 이동통
신보안, 컴퓨터시스템

Email : kscho@dankook.ac.kr