

# EJB 컴포넌트 성능 측정을 위한 Test Program 생성 방법

임병진, 이궁혜  
한국항공대학교 컴퓨터공학과  
(nomja, khlee@mail.hankong.ac.kr)

## Generating Performance Test Programs for EJB Components using Their Signatures

Byungjin Lim, Keung Hae Lee  
Department of Computer Engineering, Hankuk Aviation University

### 요 약

컴포넌트의 성능 측정을 위해서는 크게 화이트박스 모델과 블랙박스 모델에 기초한 측정 방법을 고려해볼 수 있다. 우리의 이전 연구는 소스코드에 성능 측정을 위한 코드를 삽입하는 방법으로 EJB 컴포넌트의 성능을 측정하고자 하였다. 이러한 화이트박스 모델에 기초한 테스트 방식은 컴포넌트의 소스코드에 접근이 가능할 경우에만 적용이 가능하다. 연구 초기에는 바이너리 형태의 EJB 컴포넌트를 역컴파일하여 소스코드를 얻어내는 것이 가능하였으나 점점 더 많은 컴포넌트들이 역컴파일 방지하기 위한 수단을 사용하는 것이 관찰되었다. 본 논문은 EJB 컴포넌트를 블랙박스로 보고 성능을 측정하는 방법에 대하여 논한다. EJB 컴포넌트의 성능을 측정하기 위한 테스트 프로그램을 EJB 컴포넌트의 외부 명세로부터 자동 생성하기 위한 방법을 연구중으로써, 동일 명세를 만족하는 다수 컴포넌트의 성능을 측정하기 위하여 EJB 컴포넌트의 시그내처로부터 테스트 프로그램을 생성하는 방법에 대하여 설명한다.

## 1. 서 론

컴포넌트 형태로의 소프트웨어 개발 기술의 발전은 재사용성을 높이고, 개발 시간 단축과 비용의 절감한다는 장점을 가진다. 한국 소프트웨어 컴포넌트 컨소시엄에서 발표한 2000년 12월 자료에 의하면, CBD(Component Base Development) 기반 소프트웨어 개발 경험이 있는 업체는 조사 대상 중 75.7%를 차지하고 있다[1]. 이는 컴포넌트 시장의 성장을 보여주고 있다.

컴포넌트의 사용으로 개발된 소프트웨어는 컴포넌트의 성능에 영향을 받는다. 최적화되지 않은 컴포넌트는 응용프로그램의 성능을 떨어뜨리는 원인이 된다. 따라서 소프트웨어 개발자는 객관적으로 평가된 컴포넌트의 성능을 미리 인지하고, 컴포넌트를 선택하여 소프트웨어를 개발하여야 한다[2].

본 논문에서는 분산환경에서 EJB 컴포넌트 성능을 측정하는 방법을 제안하고 이러한 측정을 수행하기 위한 도구를 설계한다. 2장에서는 이전의 연구와 문제점에 대해 설명한다. 3장에서는 본 논문에서 제안하는 컴포넌트 성능 측정 방법에 대해 설명한다. 4장에서는 결론 및 향후 연구에 대해 논한다.

## 2. 이전 연구 및 문제점

### 2.1 역컴파일을 통한 J2EE 컴포넌트 성능 측정 방법

AUTOPEC(AUTO tool of the Performance measuring for the j2Ee Component)의 컴포넌트 성능 측정 방법은 바이너리 형태의 컴포넌트를 역컴파일 과정을 통해 소스 파일을 얻어내고 소스 파일에 성능 측정을 위한 코드를 삽입하여 재컴파일하는 것이다. 재컴파일된 컴포넌트는 다시 J2EE Server에 Deploy하여야 한다. 또 컴포넌트의 성능 측정을 위해서 컴포넌트를 사용하여 소프트웨어를 개발하여야 한다. 그림 1은 AUTOPEC에서의 성능 측정 과정을 도식화 한 것이다[3][4].

이러한 방법은 컴포넌트의 성능 측정을 위하여 컴포넌트를 사용한 소프트웨어를 개발해야만 하는 2차적인 노력을 요구하게 된다. 또 최근에는 java class 파일에 대한 역컴파일 방지 기술이 연구되고 있다[5]. Commercial 컴포넌트들은 이러한 역컴파일 방지 기술을 통해 자신들의 바이너리 컴포넌트를 역컴파일하지 못하도록 할 것이다. 이러한 역컴파일 방지 기술을 택한 컴포넌트에 대해서는 컴포넌트의 소스 파일을 얻어 낼 수 없다. 따라서 성능 측정 코드를 삽입 할 수 없고, 컴포넌트의 성능을 측정할 수 없게 된다.

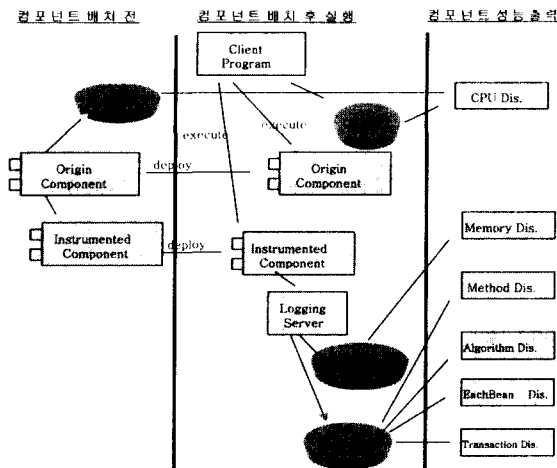


그림 1. AUTOPEC 시나리오

### 3. 컴포넌트 성능 측정 도구 설계

#### 3.1 성능 측정 요소

EJB에서 컴포넌트의 성능 측정 요소로 빈 처리 응답 시간, 트랜잭션 처리 응답 시간, 메모리 사용률, CPU 사용률을 들 수 있다.

빈 처리 응답 시간은 컴포넌트의 메소드의 실행 시간을 측정하여 기능 수행에 얼마만큼의 시간이 걸리는지를 알아내는데 사용한다. 트랜잭션 처리 응답 시간은 기능 수행에서 트랜잭션이 얼마만큼의 시간을 소모하는지를 측정한다. 메모리 사용률과 CPU 사용률은 컴포넌트를 사용하는 동안 얼마만큼의 메모리를 사용하고, CPU를 점유하는지를 측정한다.

#### 3.2 성능 측정 시스템 설계

그림 2와 같이 성능 측정 시스템은 크게 컴포넌트의 Spec을 입력 받고, Test Program을 자동 생성하는 Performance Test Program Generator Part, 파라미터에 대입 될 값을 생성하는 Input Data Set Generator Part, Component를 성능 측정하는 Test Program Part, GUI를 통해 측정 결과를 사용자에게 보여주는 Performance Result Reporter Part의 네 부분으로 구성된다.

컴포넌트의 성능을 측정하기 위해서는 우선 컴포넌트에서 제공하는 메소드들의 정보를 알아야 한다. 컴포넌트의 메소드를 알아내는 방법은 사용자가 컴포넌트의 메소드를 입력하는 것과 자바에서 제공하는 reflection기능을 사용하는 것이다. reflection은 컴포넌트의 class 파일이 필요하다. 사용자의 입력이나 reflection을 통해 알아내는 컴포넌트의 메소드 정보는 메소드의 이름, 리턴 타입, 파라미터의 종류등이다.

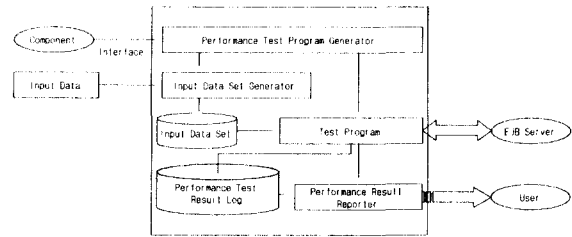


그림 2. 성능 측정 시스템의 구성

파라미터가 존재하는 메소드를 사용하기 위해서는 파라미터에 대입될 실제 값이 필요하다. Input data set generator에서는 컴포넌트에서 추출한 메소드 정보를 가지고 파라미터에 대입될 값을 생성해 낸다. 생성된 데이터는 동일한 기능을 수행하는 다른 컴포넌트의 성능 측정에서 사용할 수 있도록 파일에 기록한다.

Input Data Set Generator는 컴포넌트의 비즈니스 메소드를 호출함에 있어서 필요한 Parameter를 생성한다. 분석된 클래스 파일에서 얻을 수 있는 Parameter 정보는 Parameter type의 이름 정도에 불과하다. Input Data Set Generator는 Parameter type을 입력받아 해당 타입의 랜덤한 실제 값을 생성시켜 그 값을 리턴한다. 리턴된 Parameter 값은 실행할 메소드의 Parameter로 사용한다.

Input Data Set Generator를 설계함에 있어서 기본적으로 고려해야 할 사항은 다음과 같다.

1. 리모트에 위치한 Client에서 EJB 컴포넌트를 접근하기 위해서 반드시 필요한 요소로는 stub class 들과 Remote Interface와 Home Interface의 class file 등이 있다.
2. Primitive type을 처리해 주기 위한 Data file이 필요하다.
3. Primitive type을 처리해 주는 루틴과 Object type을 처리해 주는 루틴은 분리한다.

Input data set의 생성이 완료되면 Performance Test Program Generator는 컴포넌트의 메소드와 Input data set, Test Program Template을 사용하여 Test Program을 생성한다. 컴포넌트의 메소드가 존재하면 메소드를 호출하기 위한 코드를 작성한다. 메소드가 파라미터를 사용한다면 그 파라미터에 해당하는 값을 Input data set에서 읽어와 대입한다. 성능 측정 코드는 메소드의 호출 전과 후에 배치하고 메소드의 실행이 끝나면 그 차를 계산하여 기록한다. 컴포넌트의 메소드가 존재하지 않을 때까지 이러한 과정을 반복하여 Test Program을 생성한다.

Performance Test Program Generator에 의해 생성된 Test Program은 기본적으로 EJB Client Program의 형태를 가지게 된다. Test Program은 EJB Server에 Deploy되어 있는 컴포넌트와 서로 통신하게 되며 컴포넌

트의 메소드를 호출함으로써 메소드에 대한 성능 측정 코드를 통해 컴포넌트의 성능을 측정하고 그 결과를 Performance Test Result Log로 기록한다.

Test Program은 내부적으로 import Part, EJB Component Initialize Part, Input Data Set Initialize Part, Performance Analysis Part, Program Destroy Part등의 5가지 부분으로 나눌 수 있다.

import part는 Test Program에서 사용하는 Java Package와 측정할 컴포넌트의 Package를 import 하는 부분이다. 성능 측정을 위한 Runtime class가 기본적으로 import 되어야 하며, EJB component를 사용하기 위해 javax.naming, javax.rmi, javax.ejb.EJBHome등의 package를 포함하여야 한다. 그 외 성능 측정 시스템에서 만든 package들을 import 하여야 한다.

EJB Component Initialize Part에는 측정하려는 컴포넌트를 생성하기 위한 코드가 위치한다. EJB 컴포넌트를 사용하기 위한 과정은 먼저 JNDI(Java Naming & Directory Interface) 초기화를 위해 System Property를 얻는다. 얻어낸 System Property로 JNDI를 통해 홈 객체에 대한 레퍼런스를 얻어내고, EJB Object를 생성하고 remote Interface에 대한 레퍼런스를 넘겨준다. 생성된 EJB Object의 remote Interface를 통해 컴포넌트를 사용할 수 있다.

Input Data Set Initialize Part에서는 Input Data Set Generator를 사용하기 위해 Input Data Set Generator 객체를 생성한다. 생성된 객체는 컴포넌트의 메소드 호출시 Parameter의 값을 설정하기위해 호출된다.

Performance Analysis Part에는 컴포넌트의 비즈니스 메소드들에 대해 성능 측정 코드가 위치하게 된다. 성능 측정은 메소드가 시작되기 전 현재 상태를 저장하고 메소드의 실행이 종료되었을 때의 상태의 차이를 계산함으로써 이루어진다.

Test Program의 실행을 종료하면서 EJB 컴포넌트 객체를 제거한다. Test Program은 성능 측정 시스템과 독립적으로 실행되기 때문에 Test Program의 실행이 종료되었음을 사용자나 성능 측정 시스템에 알려줄 필요가 있다. 따라서 실행이 종료되었을 때 제어권을 성능 측정 시스템에 넘겨주는 부분을 Program Destroy Part에서 수행한다.

Test Program의 실행이 종료되어 성능 측정이 종료되면 Test Program에 의해 생성된 Performance Test Result Log를 가지고 Performance Result Reporter는 사용자에게 성능 측정 결과를 보인다.

#### 4. 결론

본 논문에서는 컴포넌트의 성능 측정을 자동화하도록 Test Program을 자동 생성하는 방법에 대해 연구하였다. 이 방법은 컴포넌트의 성능을 측정하기 위해 따로 소프트웨어를 구현해야 하는 문제점을 해소하고, 역컴파일의 불가능하도록 한 상업 컴포넌트에 대해서도 성능을 측정 할 수 있다. 동일 명세를 만족하는 다수 컴포넌트에 대해 적용이 가능한 테스트 프로그램을 생성할 수 있으므로 컴포넌트의 성능을 객관적으로 비교할 수 있는

수단이 될 수 있을 것이다.

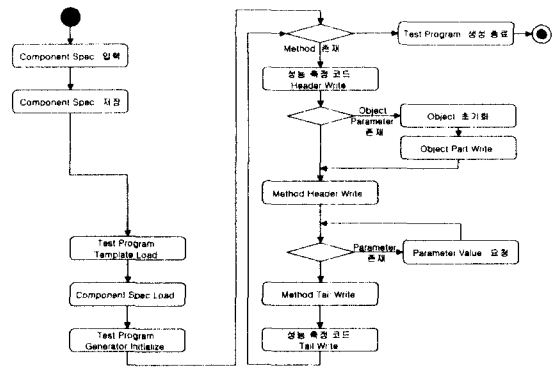


그림 3. Test Program 생성 방법

#### 5. 참고 문헌

- [1] 한국 소프트웨어 컴포넌트 컨소시엄, "컴포넌트 산업 육성 계획수립을 위한 컴포넌트 산업 및 기술 동향 조사", <http://www.component.or.kr/>, 12. 2000.
- [2] Jerry Ga, "Component testability and component testing challenges", CMU Software Engineering, Carnegie Mellon University, 2000
- [3] 오창남, 이궁해, Enterprise JavaBeans(EJB) 컴포넌트의 성능 측정 방법, 추계학술 발표회, 한국정보과학회, 2000년 10월
- [4] 오창남, 이궁해, Enterprise JavaBeans(EJB) 컴포넌트의 성능 측정 시스템 설계, 2000년 한국정보처리학회 추계학술 발표회, 한국정보처리학회, 2000년 10월.
- [5] "SourceGuard", URL <http://www.4thpass.com/sourceguard/support/index.html>, 4thpass Inc.