2001년 한국정보처리학회 추계 학술발표논문집 제8권 제2호

EJB 애플리케이션을 위한 멀티 쓰레드 구현 방법에 대한 연구

이영지[0] 김태윤
고려대학교 컴퓨터학과
yjlee@netlab.korea.ac.kr

Multithread design of Enterprise Java Beans

Young-Ji Lee[0] ,Tae-Yoon Kim
Dept. of Computer Science, Korea University

요 약

EJB는 Sun사에서 발표한 분산 객체 구조에 맞는 서버쪽 컴포넌트 아키텍처이다. EJB를 사용하면 다중 네트워크 환경에서 애플리케이션의 디자인과 개발, 배포가 쉬워진다. 개발자는 하부 사항에 대해 신경쓰지 않고 상위 레벨에서 애플리케이션을 설계할 수 있다. EJB 1.1 명세서는 그러한 내용을 나타내고 있는 명세서이다. 하지만 이 명세에서는 멀티 스레드를 허용하지 않는다. 멀티 스레드는 프로그래밍에서 상당히 유용한 것으로 멀티 스레드를 사용하면 다른 오브젝트에 영향을 주지 않으면서 작업을 수행할 수 있는 등 여러 가지 이점이 많다. 본 논문에서는 이러한 이점을 살려서 명세서에 따르면서 멀티 스레드를 생성하는 방법에 대해 알아보고자 한다. 본 논문에서 제시하는 방법은 클라이언트 쪽에서 멀티 스레딩을 생성하는 방법, 콜백 서비스를 이용하는 방법, 메세징 서비스를 이용하는 방법 3가지이고 각각에는 장단점이 따른다. 따라서 실제 구현 시에는 각자의 요구 사항에 맞춰 알맞은 방법을 찾아야 한다.

1. 서 론

EJB(Enterprise Java Beans)는 다중 네트워 환경에서 분산 객체 구조에 기반한 자바 응용프로그램의 개발 및 확장을 위한 컴포넌트 모델이다. 컴포넌트 모델이란 재사용 가능한 응용 프로그램 요소의 작성을 지원하기 위한 프로그래밍 방안이며, 컴포넌트란 개발하는 프로그램에 추가될 수 있는 프로그램의 기정의 모듈이라고 볼 수 있다. EJB는 서버 상에서 동작하는 서버 프로그램 요소들의 작성을 지원하기 위한 Java Beans 컴포넌트 모델의 확장형이라 할 수 있다. EJB는 클라이언트/서버 모델의 서버 부분에서 운영되는 자바 프로그램 컴포넌트들을 설정하기 위한 아키텍처로서 네트웍 내의 클라이언트들에 분산되어 있는 프로그램 컴포넌트들을 위한 java beans 기술 위에서 구현된다. EJB 컴포넌트들은 다중 응용프로그램들에서 재 사용되는 장점을 가지고 있다. EJB 빈이나 컴포넌트가 배치되기 위해서는 컨테이너라고 불리는 특정 응용프로그램의 일부가 되어야한다. EJB 아키텍쳐는 비즈니스 로직, 즉 기능적 코드 이외의 부분에 대한 코딩에 대해 프로그래머가 신경 쓰지 않아도 되는 구조를 제공한다[1].

이것은 EJB 구조에서 비즈니스 로직을 담당하는 빈 객체 부분과 그 객체의 기능적이지 않은 속성에 대해 명시하는 배치 디스크립터로 나눈 것과 유사하다[2].

EJB의 1.0 명세서는 EJB의 비 동시적인 쓰레드 생성을 금지하고 있다. 여러 가지 이유가 있지만, 멀티 스레드는 작업을 편리하게 해준다. 따라서 본 논문에서는 EJB 1.0 명세서를 만족하면서 멀티 스레드를 구현하는 방법에 대한 메커니즘을 제시하고자 한다.

2. 본 론

본 논문에서는 EJB 1.1 명세를 따르면서 멀티 스레딩을 구현할 수 있는 방법을 제시한다. 본 논문에서 설명하는 세 가지 방법은 저마다 장단점을 가지고 있다.

첫째, 클라이언트 쪽에서 멀티 스레딩을 생성하는 방법이다. 그림 1은 기본적인 클라이언트 쪽에서의 스레딩 생성 방법을 나타내고 있다. 클라이언트는 스레드 1을 생성시킨다. 클라이언트가 비동시적인 서비스를 호출하는 경우에 클라이언트는 새로운 스레드 2를 생성한다. 스레드 2는 스레드 1의 작업이 끝날 때까지 기다린다. 스레드 1의 작업이 끝난 경우에는 스레드 2에 알림을 주거나 메소드 호출의 결과를 알려주어 스레드 2의 작업이 시작되도록 한다.

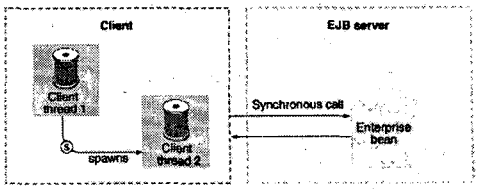

그림 1 클라이언트 쪽에서의 멀티 스레딩

이 방법은 클라이언트 쪽의 스레딩이 그것의 서비스를 사용하는 엔터프라이즈 빈의 설계 과정에서 부가적인 요구를 하지 않는다는 장점이 있는 반면 클라이언트 쪽의 디자인이 멀티 스레딩에 관한 API와 다른 주제들을 다루어야 한다는 단점이 있다. 더구나 이 방법은 시스템을 클라이언트가 EJB 서버가 작업을 끝냈는지 알기 위해 작업 세션을 계속 유지하고 있어야 하기 때문에 시스템에 대한 확장성이 떨어지게 된다.

두 번째는 콜백을 이용하는 방법이다. RMI나 CORBA는 모두 점대 점 아키텍쳐 기반으로 이루어져 있어서 서버와 클라이언트의 역할이 정확하게 구분되어있지 않다. 상황에 따라서 오브젝트에 대한 참조를 넘기는 쪽이 클라이언트가 되고 그 참조에 대한 작업을 수행하는 쪽이 서버가 된다.

그림 2에서 클라이언트 A는 리스너 오브젝트를 생성하고 엔터프라이즈 빈에 등록한다. 클라이언트 B는 엔터프라이즈 빈으로서 동작하고 이것은 이벤트를 발생시킨다. 이것은 CORBA, RMI, EJB의 경우를 따져봤을 때 이 방법은 구현하기에 쉽지 않다[3].

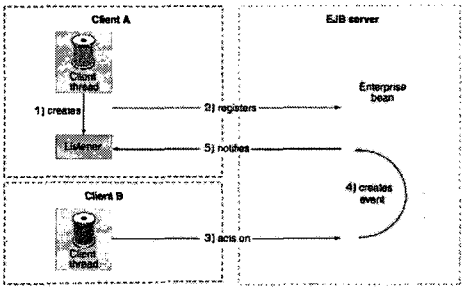

그림 2 콜백을 이용한 멀티 스레딩

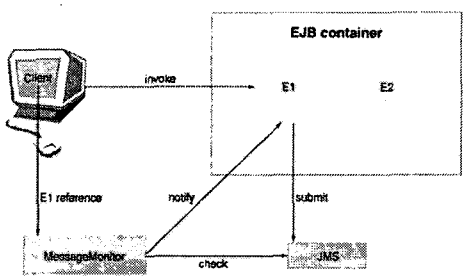

그림 3 메세징을 이용한 비동기화

마지막으로 EJB와 함께 MOM을 이용하는 방법이다. 메시지 지향 미들웨어(MOM)는 애플리케이션 서버와 같은 메세징 자원에 대해 비동시적인 접근을 제공한다. MOM은 메시지를 큐에 저장하고 있다. 자원이 사용가능하게 되면 MOM은 메시지를 받고 메시지의 작업이 진행되었는지 안되었는지 결과를 알려준다. Sun에서는 엔터프라이즈 메세징 시스템에 접근하기 위한 표준 자바 API인 자바 메시징 서비스(JMS) 시스템을 소개하였다. 그것은 IBM의 MQ 시리즈나 다른 것들과 같이 엔터프라이즈 메세징 애플리케이션에 의해 지원되는 쉽고 편리한 방법을 디자인하기 위한 일반적인 API 인터페이스를 제공한다.

하지만 EJB 1.1 명세서에서는 JMS와 같은 메신저 서비스를 쓸 수 없다고 명시되어 있다.

그림 3은 클라이언트가 EJB 컨테이너와 함께 동작하고 있는 엔터프라이즈 빈 E1를 어떻게 호출하는지 나타내는 것이다.E1은 메시지를 JMS에 포워딩 한다. 클라이언트 요청이 JMS 에 도달하면 제어권은 엔터프라이즈 빈 E1과 클라이언트에게 넘어간다. 그 다음에 엔터프라이즈 빈 E1과 클라이언트는 그들의 작업으로부터 자유롭게 된다. 클라이언트는 엔터프라이즈 빈 참조 E1을 메시지 모니터에 넘기게 된다.

이제까지 위에서 언급한 3가지 방법은 멀티 스레딩에 대한 제한점을 극복한다. 각각의 방법은 서로 장단점이 있다. 사용자의 애플리케이션에 따라서 적절한 방법을 선택해야 한다.

3. 결론 및 향후 과제

본 논문에서는 EJB 명세서를 만족시키면서 멀티 스레드를 생성시키는 알고리즘에 대해 알아보았다. 위에서 언급한 세가지 알고리즘은 클라이언트 쪽에서 스레드를 생성하는 방법과 콜백 서비스를 이용하는 방법, MOM과 EJB를 같이 연동시키는 방법이 있다. 각각의 알고리즘은 장단점이 가지고

있기 때문에 상황과 특성에 맞추어서 다른 방법을 적용해야 한다. 앞으로의 향후 과제로는 EJB 아키텍처의 부하를 줄이면서 멀티 스레드를 구현할 수 있는 메커니즘에 대한 연구를 삼았다.

4. 참고 문헌

[1] 이지영, 컴포넌트의 동적 변환을 지원하는 Enterprise JavaBeans 서버, 2000, 고려대학교 석사 학위 논문

[2] Ed Roman, mastering Enterprise Javabeans, wiley, 1999

[3] http://www.devx.com/upload/free/features/javapro/2000