

EJB 빈의 성능 메트릭

나학청¹, 김수동
숭실대학교 컴퓨터 학과

hchna@otlab.ssu.ac.kr, sdkim@comp.ssu.ac.kr

Performance Metrics for EJB Beans

Hak Chung Na¹, Soo Dong Kim
Dept. of Computing, Soongsil University

요약 (Abstract)

Java 2 Enterprise Edition(J2EE)의 등장으로 국내의 수많은 기업들은 J2EE의 모델에 맞게 엔터프라이즈 어플리케이션을 개발하고 있다. 이것은 J2EE의 핵심 기술 요소인 Enterprise JavaBeans(EJB)의 컴포넌트 모델이 분산 객체 어플리케이션의 개발 과정을 간단하게 해주기 때문이다. EJB 어플리케이션은 여러 개의 빈들로 구성된다. EJB 어플리케이션의 서비스는 클라이언트의 요청에 따른 빈의 비즈니스 메소드의 실행으로 이루어진다. 따라서 EJB 어플리케이션의 성능은 클라이언트의 요청에 따라 처리하는 빈에서의 측정과 요청을 처리하는 비즈니스 메소드의 측정에 매우 중요하다. 본 논문에서는 EJB 어플리케이션에서 클라이언트의 서비스 요청에 따라 수행하는 빈 단위에서의 성능 메트릭을 제시한다. 클라이언트의 서비스 요청은 요청을 받은 빈에서의 메소드 실행으로 나타난다. 메소드의 유형을 분류하고, 각 유형에 따른 메트릭을 제시한다.

1. 서론

소프트웨어 산업계의 고품질의 소프트웨어의 경제적인 개발로서 컴포넌트 기반 소프트웨어 개발 기법이 부각되고 있다. 1998년 선(SUN) 사에 의해 제안된 EJB는 컴포넌트 기반의 분산 업무 어플리케이션의 개발과 배치를 위한 컴포넌트 아키텍처로서 이러한 상황에 부합한다.

EJB 기반의 어플리케이션은 트랜잭션, 메시징과 비동기적 작업관리, 보안과 인증 등의 미들웨어 서비스를 이용하여 구축한다. 이것은 어플리케이션 개발의 편의성 및 확장성, 신뢰성, 안정성을 보장하여 소프트웨어 산업계에 각광을 받으면서 국내의 많은 기업들은 EJB 기반의 어플리케이션을 개발하고 있다. 그러나 EJB 어플리케이션의 내부는 EJB 서버에 의해 감추어져 있어 기존의 성능 메트릭 적용을 힘들게 만든다. 따라서 EJB 어플리케이션의 이러한 특성이 잘 반영된 보다 적합한 성능 메트릭이 요구된다. 본 논문에서는 EJB 어플리케이션을 구성하는 빈에 대한 성능 메트릭을 제시한다[1].

본 논문의 2장에서는 어플리케이션 성능 평가에 관한 기존 연구를 살펴보고, 3 장에서는 EJB 어플리케이션 아키텍처를, 4장에서는 EJB 어플리케이션에서의 빈을 위한 성능 메트릭을 정의한다.

2. 관련 연구

클라이언트/서버 시스템에서 응답시간(Response time)과 처리량 (Throughput), 비용은 중요한 메트릭이다. 응답시간은 그림 1에서 나타나듯이 반응시간이나 응답시간으로 정의된다. 웹에서 브라우저를 사용하는 경우, 사용자가 브라우저에 나타난 링크를 클릭하는 순간 서버는 트랜잭션을 시작한다. 첫 번째 문서가 나타나는 디스플레이가 되기 시작하는 순간, 반응 간격 (Reaction Interval)이 끝나고, 그 문서와 관련된 텍스트와 이미지 파일이 완전히 디스플레이 되면 응답 시간 간격 끝난다[2].

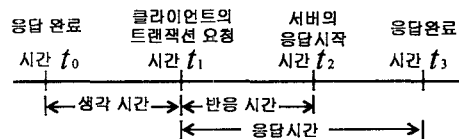


그림 1. 클라이언트/서버 환경에서의 응답시간

객체 지향 방법론에 따라 개발된 어플리케이션에서 메트릭은 실행 시에 객체의 메시지 흐름을 이용하여 실행 시에 측정되는 각 메시지의 부하와 객체 내에 포함되어 있는 함수들의 메시지 전송 회수, 호출하는 객체의 특성에 따른 가중치, 객체 간의 병렬성과 관련성, 호출하는 객체의 함수에서 외부 객체를 참조하는 함수의 메시지 수를 이용하여 동적 결합도와 동적 응집도를 계산한다[3].

Liu T.의 접근은 EJB 기반의 분산 엔터프라이즈 어플리케이션 성능을 예측하기 위해 Layered Queueing Models(LQM) 을

사용한다. 이 접근에서는 서버 프로세스를 모델하기 위해 태스크들을 세 계층으로 나누고, 리소스의 사용은 EJB 빈의 메소드 레벨에서 결정하며, 서비스 요청 대비 부하량에 대한 응답시간을 획득한다[4].

Liad6의 접근은 EJB 서버의 메소드 호출 처리를 위한 분석적 모델을 제안한다. 시스템의 분리된 기능성을 연구하여 그 기능성들을 계층적 방법으로 병합하며, 단위 시간당 메소드 실행 수로서 정의된 처리량과 응답 시간에 관하여 그것의 성능을 예측하기 위해 메소드 호출 실행을 분석적으로 모델하며, 빈 인스턴스가 비 활성화 상태일 때 발생하는 블로킹(Blocking)을 분리한다[5].

위와 같은 기존의 성능 메트릭은 많은 연구가 분석, 설계 단계에서의 성능 예측을 위한 것이었으며, 운영 단계의 EJB 어플리케이션에서 빈의 성능 측정에는 EJB의 여러 가지 특성으로 인해 충분치 않다. 따라서 운영 단계에서의 EJB 어플리케이션에서 빈을 성능 측정하기 특성화된 메트릭이 요구된다.

3. EJB 어플리케이션 아키텍처

EJB 어플리케이션의 아키텍처는 같이 클라이언트 계층과 컨트롤러 계층, 비즈니스 계층, 데이터 액세스 계층, 데이터베이스로 계층화될 수 있다. 그림 2는 계층 간의 상호 관계와 메시지 흐름을 나타내고 있다. 밑줄 그어진 내용이 있는 사각형은 각 계층을 위해 이용되는 빈의 유형을 나타낸다.

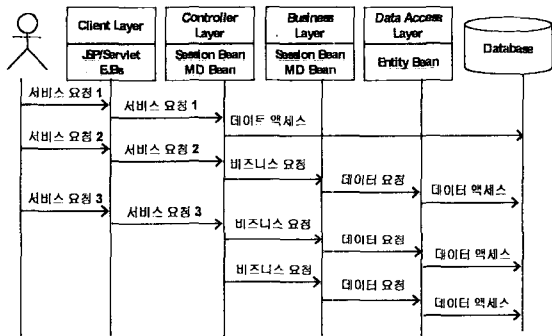


그림 2. EJB 어플리케이션 아키텍처

각 계층은 하위 계층이나 같은 계층으로 서비스를 요청할 수 있으며, 또한 바로 아래 계층을 건너 뛰어 요청할 수 있다. 서비스를 다른 서버 상의 빈에 요청할 때는 RMI를 통해서 이루어지며, 각 계층에서 데이터베이스로의 액세스는 JDBC를 통해 수행한다

4. 메소드의 유형

클라이언트의 서비스 요청에 따라 빈의 메소드 실행은 여러 가지 유형으로 분류될 수 있다. 그림 2에서 보이듯이 클라이언트 계층의 서비스 요청은 유형에 따라 컨트롤러 계층의 빈이 다른 빈을 이끌어 오거나, 클라이언트로부터 요청을 받은

빈이 혼자서 요청된 서비스를 처리하는 경우가 발생한다. 이러한 각각의 경우에 따라 빈 상의 메소드는 여러 유형으로 분류할 수 있으며, 그 유형에 따라 적용되는 성능 메트릭 또한 달라져야 한다.

메소드 타입 I은 클라이언트 프로그램(다른 빈이나 서블릿 같은 자바프로그램)의 서비스 요청이 요청을 받은 빈에서 다른 빈의 서비스를 받지 않고 자체적으로 처리하는 메소드 타입을 말한다. 메소드 타입 II는 클라이언트 프로그램의 서비스 요청을 다른 빈의 서비스를 이용하여 처리하는 메소드 타입을 말한다. 메소드 타입 III는 클라이언트 프로그램의 서비스 요청을 다른 빈에 대해서 트랜잭션적인 메소드들을 실행하는 메소드 타입이다.

5. 빈을 위한 성능 메트릭

클라이언트의 서비스 요청은 빈 상의 메소드 실행으로 이어진다. 상황에 따라 빈의 EJBHome과 EJBObject에 대해서 바인딩 작업, 빈 인스턴스의 활성화 작업, 데이터 베이스에 대한 액세스, 하나 이상의 빈들을 사용함에 따라 발생하는 성능적 영향을 발생한다.

5.1. 메소드 타입 I의 응답 시간

메소드 타입 I에서의 응답시간은 그림 3와 같이 메소드 호출 전송 시간 $I_{send MethodReq}$, 대기시간 $I_{waitForME}$, 실행결과 반환 시간 $I_{send resultOfME}$, 메소드 실행시간 $I_{execOfME}$ 으로 구성되며, 상황에 따라 EJBHome, EJBObject에 대한 바인딩 시간 $I_{binding EJBHome}$, $I_{binding EJBObj}$ 이 추가된다. 대기시간은 빈활성화 시간을 포함하며, 메소드 응답시간 $I_{responseOfMethodOnEJBObj}$ 은 EJBObject에서의 메소드 응답시간으로 비즈니스 요구에 따라 데이터베이스에 대한 액세스 시간 $I_{accessDB}$ 포함할 수 있다.

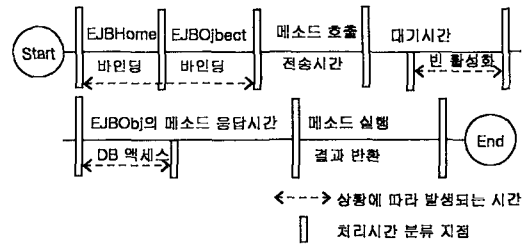


그림 3. 메소드 타입 I의 처리 시간 분류

따라서 응답시간 $I_{responseOfMethodTypeI}$ 은 다음과 같이 정의된다

$$(정의 1) I_{responseOfMethodTypeI} = I_{send MethodReq} + I_{waitForMExec} + I_{responseOfMethodOnEJBObj} + I_{send resultOfME} [[+I_{binding EJBHome} + I_{binding EJBObj}] | [+I_{binding EJBObj}]]$$

5.2. 메소드 타입 II의 응답시간

메소드 타입 II는 그림 2에서 서비스 요청 2를 처리하는 컨트롤러 계층과 같이 클라이언트 프로그램의 서비스 요청을 다른 빈의 서비스를 이용하여 처리하는 메소드 타입을 말한다.

그림 4는 Msg_r을 받은 Bean A가 Bean₁, ..., Bean_n으로 Msg₁, ..., Msg_n으로 요청하는 것을 보이고 있다.

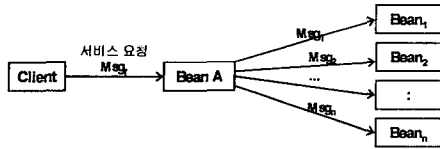


그림 4. 메소드 타입 III에서의 메시지의 분할

Bean A에 의해 이용되는 i 번째 빈 Bean_i에서 j 번째 메소드의 응답 시간을 $I_{responseOfMethod\ j\ OnBean\ i}$ 은 메소드 타입 I의 응답시간을 갖고, EJBHome과 EJBObject에 대한 바인딩 시간을 $I_{bindingEJBHomeOfBean\ i}$, $I_{bindingEJBObjOfBean\ i}$, 병렬적으로 처리되는 빈들 Bean₁, ..., Bean_k을 하나의 노드로 볼 때, p번째 노드의 응답시간을 $I_{responseOfMethodOnPNode\ p}$ 라고 하면 다음과 같이 정의될 수 있다.

(정의 2) $I_{responseOfMethodOnPNode\ p} = MAX(I_{responseOfMethod\ i\ OnBean\ i}, \dots, I_{responseOfMethod\ j\ OnBean\ i})$ 이면,
 $I_{responseOfMethodTypeIII} = \sum_{i=1}^n ((I_{bindingEJBHomeOfBean\ i} + I_{bindingEJBObjOfBean\ i} + \sum_{j=1}^m (I_{responseOfMethod\ j\ OnBean\ i})) + \sum_{p=0}^k I_{responseOfMethodOnPNode\ p}$

5.3. 메소드 타입 III의 응답시간

이 타입은 이끝어지는 다른 빈의 메소드들이 트랜잭션적으로 실행되는 경우이다. 이 때 최초로 요청받은 Bean A는 javax.transaction.UserTransaction 인터페이스의 begin()과 commit() (또는 rollback())을 이용하여 시작과 끝을 결정하며, 그 사이에 일련의 메소드들을 실행한다. 따라서 begin(), commit(), rollback()의 응답시간을 $I_{response\ Tbegin}$, $I_{response\ Tcommit}$, $I_{response\ Trollback}$, 트랜잭션의 준비시간 $I_{prepare\ transaction}$ 이라고 하면 다음과 같이 정의할 수 있다.

(정의 3) $I_{responseOfMethodTypeIII} = \sum_{i=1}^n ((I_{bindingEJBHomeOfBean\ i} + I_{bindingEJBObjOfBean\ i} + I_{response\ Tbegin} + \{ I_{response\ Tcommit} | I_{response\ Trollback} \} + I_{prepare\ transaction} + \sum_{j=1}^m I_{responseOfMethod\ j\ OnBean\ i})) + \sum_{p=0}^k I_{responseOfMethodOnPNode\ p}$

5.3.1. 빈 - 데이터베이스 사용 메트릭

빈 - 데이터베이스 사용 메트릭은 주어진 Workload에 따라 사용되는 빈의 메소드 들이 빈들에서 데이터베이스에 대해서 생성, 읽기, 업데이트, 삭제의 빈도를 보여준다. 표 1은 이 메트릭에서 사용되는 구성 요소들을 보여주고 있다.

이 메트릭은 표의 상단에 주어진 빈의 이름과 표의 좌측 수직 칸들은 Workload의 이름으로 구성한다. 그리고 각

Workload 행과 메소드의 열이 교차되는 칸들은 각 메소드들에 대해서 데이터베이스와의 오퍼레이션 유형인 Create(C), Read(R) Update(U), Delete(D)와 각 유형별 발생 횟수로서 구성된다.

표 1. 서버 - 빈 - 데이터베이스 사용 메트릭

빈		빈 이름					
부하량	빈 이름	메소드_1		...	메소드_M		
부하량 #1		C	R	U	D		R
		1	2	3	1		3
...							
부하량 #n							

이 메트릭을 이용해서 빈의 메소드들이 데이터 리소스(데이터베이스)를 어떻게 사용하는 지 파악하는데 중요하게 사용한다.

6. 맺음말 (Conclusion)

본 논문에서는 EJB 어플리케이션의 특성을 알아보고, 어플리케이션에서 하나의 빈이 운영되었을 때, 운영되는 메소드의 유형을 나누고, 각각의 빈을 위한 메트릭을 정의하였다. 또한 빈에서의 메소드들이 데이터 리소스를 사용하는 유형과 그 빈도를 위한 메트릭을 제시하였다. EJB 어플리케이션은 EJB서버와 빈들로서 구성된다. 따라서 EJB 성능 측정에는 빈 레벨에서의 측정이 요청된다. 기존의 연구가 이런 특성이 반영된 운영상태의 EJB 어플리케이션의 성능 측정에 있어서 연구가 충분치 않은 만큼, 본 논문에서 제시한 빈을 위한 메트릭이 EJB 어플리케이션의 성능 측정에 활용될 것으로 기대된다.

참고문헌

- [1] Java Community, ECperf Specification, Sun Microsystems, at URL : <http://java.sun.com/j2ee/ecperf/download.html>, May 29, 2001.
- [2] Menasce D., Almeida V., Capacity Planning For Web Performance: Metrics, Models and Methods, Prentice Hall, 1998.
- [3] Kim Ch., Cho E., Kim S., Static/Dynamic Metrics for Effective Object-Oriented Design and Performance Measurement, Journal of KISS, Nov., 1998.
- [4] Liu T., et al., Layered Queueing Models for Enterprise JavaBean Applications, IEEE, 2001.
- [5] Lladó C.M., Harrison P.G., Performance evaluation of an enterprise javabean server implementation. In: Proc. 2nd, Int. Workshop on Software and Performance (WOSP 2000, September 17-20, Ottawa, Canada), 2000