

# 3-Tier 웹 어플리케이션에서의 효율적인 테스트에 관한 연구

이재원\*, 최은만\*\*

\*한국교육학술정보원

\*\*동국대학교 컴퓨터공학과

e-mail: \*jaewon@keris.or.kr

\*\*emchoi@dongguk.edu

## Study of an Effective 3-Tier Structured Web Application Test

Jae-Won Lee\*, Eun Man Choi\*\*

\*Korea Educational and Research Information Service

\*\*Dept of Computer Engineering, Dongguk University

### 요 약

웹이 일상생활에서 널리 활용되고 웹 어플리케이션이 종래에 오프라인에서 이루어졌던 다양한 분야를 대신하면서 웹 어플리케이션에 접속하는 사용자가 급속히 증가되었다. 사용자의 급속한 증가는 다시금 웹 어플리케이션을 3-Tier 구조로 구성하여 안정적인 서비스 제공의 기반을 마련하게 만들었다. 그리고 개발된 3-Tier 웹 어플리케이션이 실질적으로 다수의 사용자에게 안정된 서비스를 제공할 수 있는지에 대한 테스트가 널리 수행되고 있다. 이러한 배경 하에서 이 논문은 3-Tier 웹 어플리케이션 테스트 수행 시 고려해야 할 요소들을 분석하고 3-Tier 웹 어플리케이션의 효율적인 성능 테스트 방안을 제시한다.

### 1. 서론

웹(WWW)이 일상생활에서 널리 활용되고 웹 어플리케이션이 종래에 오프라인에서 이루어졌던 다양한 분야를 대신하면서 사용자가 급속히 증가되었다. 사용자의 급속한 증가는 다시금 웹 어플리케이션을 2-Tier구조에서 3-Tier 구조로 구성하여 안정적인 서비스 제공의 기반을 마련하게 만들었다. 그리고 개발된 3-Tier 웹 어플리케이션이 실질적으로 다수의 사용자에게 안정된 서비스를 제공할 수 있는지를 서비스 오픈 전에 검증하기 위한 테스트가 널리 수행되고 있다. 이러한 배경 하에서 이 논문은 웹 어플리케이션의 주된 테스트에 대하여 분류하고 3-Tier 웹 어플리케이션의 테스트 수행 시 고려해야 할 요소들을 분석하여 3-Tier 웹 어플리케이션의 효율적인 성능 테스트 방안을 제시한다.

### 2. 웹 어플리케이션 테스트

웹 어플리케이션 테스트는 기존의 소프트웨어 테스

트와 다른 분류 기준을 가지고 있으며 그 분류 기준에 따라서 기능 테스트, 성능테스트, 사용성 테스트, 적합성 테스트, 보안 테스트로 나눌 수 있다.[1]

#### 가. 기능 테스트

기능 테스트는 명세서에 명시된 동작에 맞게 동작을 하는 지를 테스트하는 것이다. 링크, 폼, 쿠키, 웹 인텍싱, 데이터베이스, 자바 애플릿, 액티브 X 등에 대한 테스트가 기능 테스트에 속한다.

#### 나. 성능 테스트

성능 테스트는 요구되는 테스트로부터 시스템의 감내 능력을 확신할 수 있는가를 테스트하는 것이다. 웹 어플리케이션들을 극한 상황에서 테스트하는 로드 테스트와 스트레스 테스트가 성능 테스트의 범주에 속한다. 성능 테스트의 예로서 많은 사용자의 동시 접속, 많은 양의 데이터 유입 등이 있다. 그 외에 실행 시간, 반응 시간, 커

넥션 속도, 다운 시간 등이 성능 테스트에 속한다.

#### 다. 사용성 테스트

사용성 테스트는 사용자에게 수용되는 정도를 테스트하는 것으로, 다른 대부분의 산업에서는 규격이나 가이드 라인의 형태로 존재한다. 하지만 웹 어플리케이션 테스트에서는 가이드라인이나 규격으로 웹 사이트의 편리성을 보증하기 힘들다. 그 이유는 웹 어플리케이션의 사용성은 사용자의 숙련도에 많은 영향을 받기 때문이다. 다소 추상적인 사용성 테스트의 예로는 웹 사이트에 기입해야 할 사항 중 사용자가 필수 항목을 기입하지 않았다면, 에러 메시지를 출력해야 한다. 이 에러 메시지에 대한 반응의 정도는 사용자의 숙련 정도에 따라 차이를 보인다. 웹에 익숙하지 않은 사용자는 에러 메시지를 이해하는 데 숙련된 사용자 보다 더 많은 시간이 필요하게 된다. 또한, 에러 메시지의 표현 방법도 사용성의 정도에 영향을 준다.

#### 라. 적합성 테스트

서로 다른 구성이나 배열, 예를 들어 클라이언트와 서버(또는 외부 DB)에 대한 서로의 적합성을 테스트하는 것이다. 이와 같은 테스트는 클라이언트와 서버에 대한 완벽한 제어가 어렵기 때문에 테스트가 복잡하고 난해하게 된다. 또한 모든 가능한 조합을 테스트한다는 것은 불가능하기 때문에, 가장 일반적으로 이용되는 것들에 대한 테스트가 이루어진다. 이러한 테스트의 예시로는 웹 페이지 버전이 다른 브라우저나 운영체제에서 올바르게 동작하는가를 체크하는 것 등이 있다.

#### 마. 보안 테스트

사용자가 인터넷 뱅킹이나 전자상거래를 이용하도록 유도하기 위해서는 보안에 대한 사용자의 신뢰가 필요하다. 이러한 신뢰를 얻기 위해서 정보 접근 규제, 사용자 신원에 대한 인증 등의 총괄적인 보안을 통해 사용자의 신원 정보가 안전하게 전송될 수 있도록 해야 한다. 이러한 보안을 위한 테스트는 싱글사인온(SSL), 로그파일, 방화벽, 로그인에 대한 테스트가 있다.

이 논문에서는 다양한 테스트 유형 중에서 웹 어플리케이션의 성능 테스트에 관하여 논하도록 한다.

### 3. 3-Tier 웹 어플리케이션 성능 테스트 고려사항

#### 가. 구조상 고려사항

3-Tier구조는 웹 서버 계층, 미들웨어 계층, 데이터베이스 계층으로 구성되어 있다. 웹서버 계층에서는 이미지, 웹 문서(HTML[3]), 스타일시트파일(CSS[4])등의 정적인 문서가 위치하는 장소이다. 최근에는 웹 문서 자체에 동적인 기능을 삽입하고 있지만[DHTML] 구현할 수 있는 동적 기능에 한계가 있다. 특히 웹 문서의 스크립트를 사용한 동적 기능은 사용자 컴퓨터의 자원을 사용하여 실행되기 때문에 많이 사용할 경우에는 사용자의 컴퓨터 사양에 따라 서비스를 제공받는 속도 차이를 느끼게 된다.

미들웨어 계층에서는 비즈니스 로직을 처리한다. 비즈니스 로직은 데이터베이스에 접근하여 데이터를 조회, 수정, 삭제하는 과정에서 접목되는 업무 로직이 주를 이룬다. 따라서 미들웨어 계층에서는 동적인 업무 로직의 구현 뿐만 아니라 데이터베이스로의 빠르고 안정적인 접근을 통해서 필요한 정보를 쉽게 가공하는 기능을 제공하여야 한다.

데이터베이스 계층은 데이터베이스 엔진이 설치된 서버로서 중요한 자료를 보존하고 미들웨어에서 요청한 CRUD 작업이 빠르게 수행될 수 있어야 한다.

#### 나. 목적상 고려사항

성능 테스트는 요구되는 테스트로부터 시스템의 감내 능력을 확신할 수 있는가를 테스트하는 것이다. 웹 어플리케이션들을 극한 상황에서 테스트하는 로드 테스트와 스트레스 테스트가 성능 테스트의 범주에 속한다.

3-Tier 웹 어플리케이션 성능 테스트는 로드 테스트나 스트레스 테스트를 통하여 시스템의 병목지점을 찾는 데 있다. 즉 서비스 지연이 웹 서버 계층, 미들웨어 계층, 데이터베이스 계층 중 어느 계층에서 생기는지 점검하는 것이다.

다음으로 병목이 생기는 계층에서 병목의 원인을 진단해야 한다. 병목의 원인의 진단을 위해서는 전체 계층에 대한 성능 테스트 뿐만 아니라 해당 계층만의 성능 테스트를 병행하여 병목지점을 좁혀 나가야 한다. 이때에는 실질적인 개발자와 함께 좁혀진 병목 지점에 대한 병목원인에 대한 진단을 함께 수행하며 해결책을 찾아야 한다.

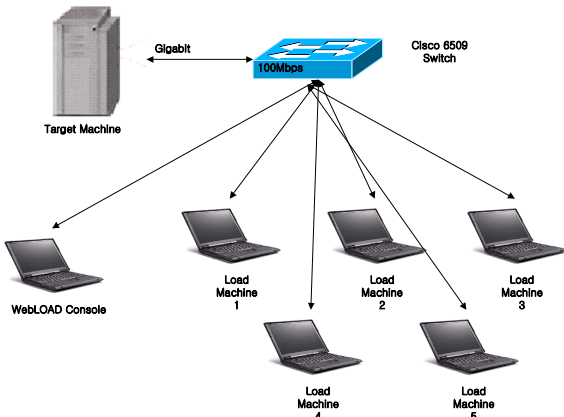
4. 3-Tier 웹 어플리케이션 성능 테스트 사례

가. 배경

교육정보서비스를 담당하는 교육포털사이트에서 기존의 CGI 기반의 2-Tier 구조를 JSP/EJB 기반의 3-Tier 구조로 변경하면서 미들웨어 비교 테스트와 함께 각 계층에서의 병목현상을 찾기 위한 성능 테스트를 수행했다.

나. 환경구축

부하발생을 위한 장비는 6대의 노트북을 사용하였으며 그 중 1대는 로드 발생과 함께 모니터링을 위한 콘솔로 사용하여 [그림 1]과 같은 환경을 구축하였으며 테스트 도구로 Empirix의 e-Load Suite[2]를 사용하였다.



[그림 1] 테스트 환경

[표 1]에서의 웹 서버 사양과 같이 이번 테스트는 네트워크 병목을 찾기 위한 목적이 아니기 때문에 물리적인 3-Tier 구조가 아니라 논리적인 3-Tier 구조를 구축하였다.

구분1	구분2	내역
Target Machine	Model	Model: HP superdome OS: HP-Unix 11i CPU: 4ea Memory: 2GB DISK: 100GB
	web server	① A 사 WebServer ② B 사 WebServer
	Was	① A 사 WAS ② C 사 WAS
	Database	Oracle 8i
Test Machine	용도	console +Load Generators 1대 Load Generators 5대
	Model	Memory: 1GB/512MB/512MB/512MB/512MB Pentium III 933 MHz CPU 256MB Main Memory

[표 1] 테스트 장비 사양

다. 테스트 방법

로드 발생을 [표 2]와 같이 세 가지 형태 구분하여 수행하였다. 첫째는 일반적인 서비스 환경의 사용자 수를 고려한 형태이며, 둘째는 폭발적인 사용자 증가를 고려한 형태이며, 셋째는 웹서버 계층을 제외하고 미들웨어 계층에만 스트레스 테스트를 수행한 형태이다.

구분	로드 발생 내역
첫째	<ul style="list-style-type: none"> <li>동시 유저(Load)의 수를 10명부터 290명까지 20명 단위로 증가시켜 테스트 함.</li> <li>순차적으로 동시 유저 수를 증가시키는 기능을 사용하여 테스트 함</li> <li>총 테스트 시간 30분, 2분단위로 Load의 수를 증가시킴.</li> </ul>
둘째	<ul style="list-style-type: none"> <li>동시 유저(Load)의 수를 500-1000-1500-2000-2500-3000 으로 증가시켜 테스트 함.</li> <li>순차적으로 동시 유저 수를 증가시키는 기능을 사용하여 테스트 함</li> <li>총 테스트 시간 30분, 5분단위로 Load의 수를 증가시킴.</li> </ul>
셋째	<ul style="list-style-type: none"> <li>동시 유저(Load)의 수를 1000-2000-3000-4000-5000 으로 증가시켜 테스트 함.</li> <li>html상의 요소(image, css, js등)들을 제외하고 jsp/ejb 호출 부분을 테스트 함</li> <li>순차적으로 동시 유저 수를 증가시키는 기능을 사용하여 테스트 총 테스트 시간 30분, 6분단위로 Load의 수를 증가시킴.</li> </ul>

[표 2] 로드 발생

또한, 실 서비스에 사용자가 접근하여 이동하는 화면 단위의 주된 이동 경로를 테스트 스크립트로 작성하였다.

서비스 구분	시나리오
교수학습자료 조회	1 초기화면
	2 선생님 채널 초기화면
	3 초등학교 3학년 말하기 · 듣기
	4 초등학교 3학년 말하기 · 듣기 1학기
	5 함께 열어가게 세상
	6 알맞은 말
	7 국어 3학년 1학기 1단원 [함께 열어가게 세상] 총괄
게시판	1 초기화면
	2 선생님 채널 초기화면
	3 공개자료실
	4 수업자료, 지도안, 평가문항, 수업사례 중 선택
	5 외국어(영어) 3학년
	6 목록보기
	7 게시물 선택 - 무작위로 뽑기
	8 중 3년 기초학력 평가문항
	9 목록보기
	10 맨뒤로 가기
	11 최상위 목록
	12 목록 보기
	13 최하위 목록

[표 3] 테스트 시나리오

라. 테스트 결과

테스트 결과로 [표 4]와 같은 테스트 요소를 추출할 수 있었으며 [그림 3]은 테스트 요소들 중에서 사용자가 측면에서 중요한 Page Time과 Round Time에 대한 테스트 결과치이다. 사용자가 평균 4초까지 대기한다면 교수학습자료조회 시나리오에서는 동시접속자 200명을 수용할 수 있으며, 게시판 시나리오에서는 230명을 수용할 수 있음을 확인할 수 있다. 셋째 형태의 테스트를 통해서 웹 서버 계층에서의 처리율과 미들웨어 계층의 처리율을 비교한 결과, 교수학습자료조회 시나리오의 1~3단계에서, 게시판 시나리오의 1~5단계에서는 웹서버 계층에서 처리되었으며 교수학습자료조회 시나리오의 4~7 단계와 게시판 시나리오의 6~12 단계에서는 미들웨어 계층에서 처리되면서 병목이 발생하였다.

Connect Time	[가상 유저]로부터 [ABT]에 접속하는데 걸린 시간
Send Time	[가상 유저]로부터 [ABT]에 HTTP 리퀘스트의 전송에 걸린 시간
Process Time	[ABT]의 HTTP 리퀘스트의 해석에 걸린 시간
Receive Time	가상 유저가 [ABT]로부터 첫 번째 바이트와 마지막 바이트를 받는 사이에 걸린 시간
Response Time	[ABT]가 HTTP 리퀘스트를 받고 나서 [가상 유저]에게 돌아올 때까지의 시간
Hit Time	[ABT]의 HTTP 리퀘스트를 성공적으로 처리하는데 걸린 시간
Page Time	[가상 유저]가 페이지를 읽는 데에 걸ლობ는데 걸린 시간
Round Time	[가상 유저]가 테스트 어젠더(시나리오)를 1회 실행하는데 걸린 시간
Throughput	1 초간 [ABT]로부터 [가상 유저]에게 보내지는 초단위의 byte 전송률
Successful Rounds	1 초간 시나리오를 성공적으로 수행한 횟수
Failed Rounds	1 초간 시나리오를 수행하는데 실패한 횟수
Page Per Second	1 초간 페이지를 읽는 데에 성공한 횟수
Hits Per Second	1 초간 HTTP 리퀘스트를 성공적으로 처리한 횟수
[ABT]	실제 테스트되는 웹 어플리케이션 서버(예:웹서버,DB 서버)를 뜻함
[가상유저]	테스트를 위해 생성된 가상의 로드(Virtual Client)를 뜻함

Round Time $\Sigma$ Page Time $\Sigma$ Hit Time (Connect + Send + Response + Process Time)	
Client	Server

[표 4] 테스트 요소

찾아형터 로드발생-교수학습자료조회 시나리오						찾아형터 로드발생-게시판 시나리오					
Load	1		2		Load	1		2			
	Page Time (평균)	Total Round (Load%)	Page Time (평균)	Total Round (Load%)		Page Time (평균)	Total Round (Load%)	Page Time (평균)	Total Round (Load%)		
10	3.727	1.6	0.269	2.27	10	3.133	1.0	0.199	0.6		
50	2.979	0.6	0.269	2.54	50	2.079	0.4	0.832	1.0		
70	3.793	0.3	1.660	2.09	70	2.082	7.3	1.284	1.31		
90	3.117	1.1	2.072	2.12	90	2.973	0.6	1.595	1.21		
110	3.786	1.43	2.307	2.05	110	3.938	1.4	1.981	1.23		
130	3.469	1.83	2.914	2.03	130	3.607	1.3	2.287	1.22		
150	3.971	1.93	3.932	1.97	150	3.193	1.2	2.693	1.22		
170	4.046	2.00	3.992	1.86	170	3.241	1.6	3.245	1.22		
190	4.287	2.24	4.004	1.89	190	3.565	0.9	3.586	1.19		
210	4.406	2.14	5.309	1.93	210	3.453	1.62	4.149	1.31		
230	6.008	2.11	6.904	1.88	230	4.008	1.65	4.380	1.19		
250	5.196	2.16	6.261	1.97	250	4.709	1.58	4.993	1.25		
270	5.978	2.17	6.759	2.01	270	4.915	1.59	5.533	1.27		
290	5.984	2.20	7.428	1.82	290	4.792	1.49	6.002	1.20		

블록형터 로드발생-교수학습자료조회 시나리오						블록형터 로드발생-게시판 시나리오					
Load	1		2		Load	1		2			
	Page Time (평균)	Total Round (Load%)	Page Time (평균)	Total Round (Load%)		Page Time (평균)	Total Round (Load%)	Page Time (평균)	Total Round (Load%)		
500	10.507	475	12.622	313	500	8.025	316	12.225	231		
1000	18.913	504	22.622	344	1000	14.363	301	16.870	232		
1500	24.621	446	30.100	402	1500	20.963	290	23.775	229		
2000	33.496	514	47.989	355	2000	31.675	297	38.383	221		
2500	47.229	481	64.289	341	2500	40.693	280	52.945	243		
3000	52.350	439	80.440	341	3000	46.703	294	64.480	198		

[그림 3] 테스트 결과표

5. 결론

3-Tier 웹 어플리케이션의 성능 테스트를 수행하기 위해서는 각 계층의 속성을 파악하여 각 계층에 속성에 맞는 스크립트를 작성하여야 한다. 웹서버 계층에서의 정적인 이미지, 사용자 클라이언트에서 수행되는 스크립트 언어, 문자 스타일 등을 정의하는 문서들이 저장되므로 정적인 이미지 처리에 대한 테스트 요소의 결과값으로 웹서버 계층의 병목을 확인하여야 한다. 미들웨어 계층에서는 비즈니스 로직의 기능 구현이 이루어진 형태이므로 이 계층에 대한 성능 테스트를 위해서는 웹서버 계층에서 수행하는 이미지, 스크립트 언어, 문자 스타일에 대한 기능을 스크립트에서 제거하고 성능 테스트를 수행한다. 데이터베이스 계층은 어플리케이션에서 사용하는 데이터들을 저장 관리하고 결과 셋을 미들웨어 계층에 빠르게 제공하는 기능을 담당한다. 따라서 데이터베이스 계층의 병목의 확인하기 위해서는 미들웨어 계층에서의 성능 테스트 과정에서 데이터베이스 계층에서 자료를 요청하는 기능의 수행 시 지연이 발생하는지 확인하여야 한다.

위와 같이 이 논문에서는 3-Tier 웹 어플리케이션 테스트의 효율을 높이기 위해서 웹 어플리케이션의 특성과 3-Tier 구조의 특성에 알맞은 스크립트 작성 및 테스트 효율적인 수행 방안을 실 사례를 바탕으로 제시하였다.

참고문헌

- [1] TTA 한국정보통신기술협회, "2003년 소프트웨어 테스트 전문 기술", 2003. 11.
- [2] <http://www.empirix.com/Empirix/Web+Test+Monitoring/testing+solutions/integrated+web+testing.html>
- [3] <http://www.w3.org/MarkUp/>
- [4] <http://www.w3.org/Style/CSS/>