

웹 어플리케이션의 성능 시험 평가 모델에 관한 연구

신현정*°, 이남용**

*바산네트워크, **송실대학교

Performance Test Evaluation Model of Web Application

Shin, Hyun Jung°, Lee, Nam Yong

Bashan Network, SoongSil University

E-mail : hjshin@bashanet.com, nylee@computing.ssu.ac.kr

요 약

웹/인터넷의 등장 이후, IT환경에서는 웹 사이트 수, 사용자 수의 폭발적인 증가로 하드웨어와 소프트웨어의 구성이 복잡하게 되었다. 적용 업무의 장애가 발생시에 신속하고 정확한 장애원인을 파악하기가 어려워진다. 따라서, 웹 어플리케이션에 맞는 전문적인 성능 테스트와 성능 관리가 필요하게 된다. 본 논문에서는 웹 어플리케이션의 특성에 맞는 성능 시험 평가 모델을 제시한다.

1. 서론

최근의 컴퓨터 시스템은 더욱 복잡해지고 있으며, 이러한 기술은 인터넷상의 많은 네트워크 기술들에 의해서 종속된다. 이에 따른, 웹 사이트의 증가, 사용자 수의 증가, 복잡한 시스템 구조, 잦은 변화, 업무의 중요성, 제한된 예산 때문에 웹 사이트의 성능 테스트의 중요성이 대두되고 있다. 그러나, 성능 테스트가 중요시되고 있으나 아직 웹 어플리케이션의 유형에 적합한 체계적인 성능 테스트 및 평가가 미흡하다. 따라서, 웹 어플리케이션의 성능 테스트 및 평가 시 고려되어야 할 성능 목표, 기준 등을 식별을 하게 되며, 체계적

인 성능 테스트 평가 모델을 제시한다. 본 논문의 구성은 다음과 같다. 먼저 제 2장에서는 본 논문에서 제시하는 웹 어플리케이션에서 주요한 성능 테스트의 목표를 설명하며, 기존의 웹 어플리케이션의 성능 테스트의 연구에 대해서 살펴본다. 3장에서는 본 논문에서 제시하고 있는 성능 테스트 모델을 설명한다. 4장에서는 3장의 성능 테스트 모델에 대한 사례 연구를 설명한다. 끝으로 5장에서는 결론과 앞으로 해결해야 할 연구 방향에 대해서 기술한다.

2. 관련연구

2.1 성능 테스트

성능 테스트는 블랙박스 테스트로 볼 수 있으며, 이는 프로그램의 코드의 지식이 없이 외부로부터 시스템과 어플리케이션의 행위에 초점을 맞춘다. 성능 테스트의 유형으로는 로드 테스트, 스트레스 테스트, 볼륨 테스트들이 있다.

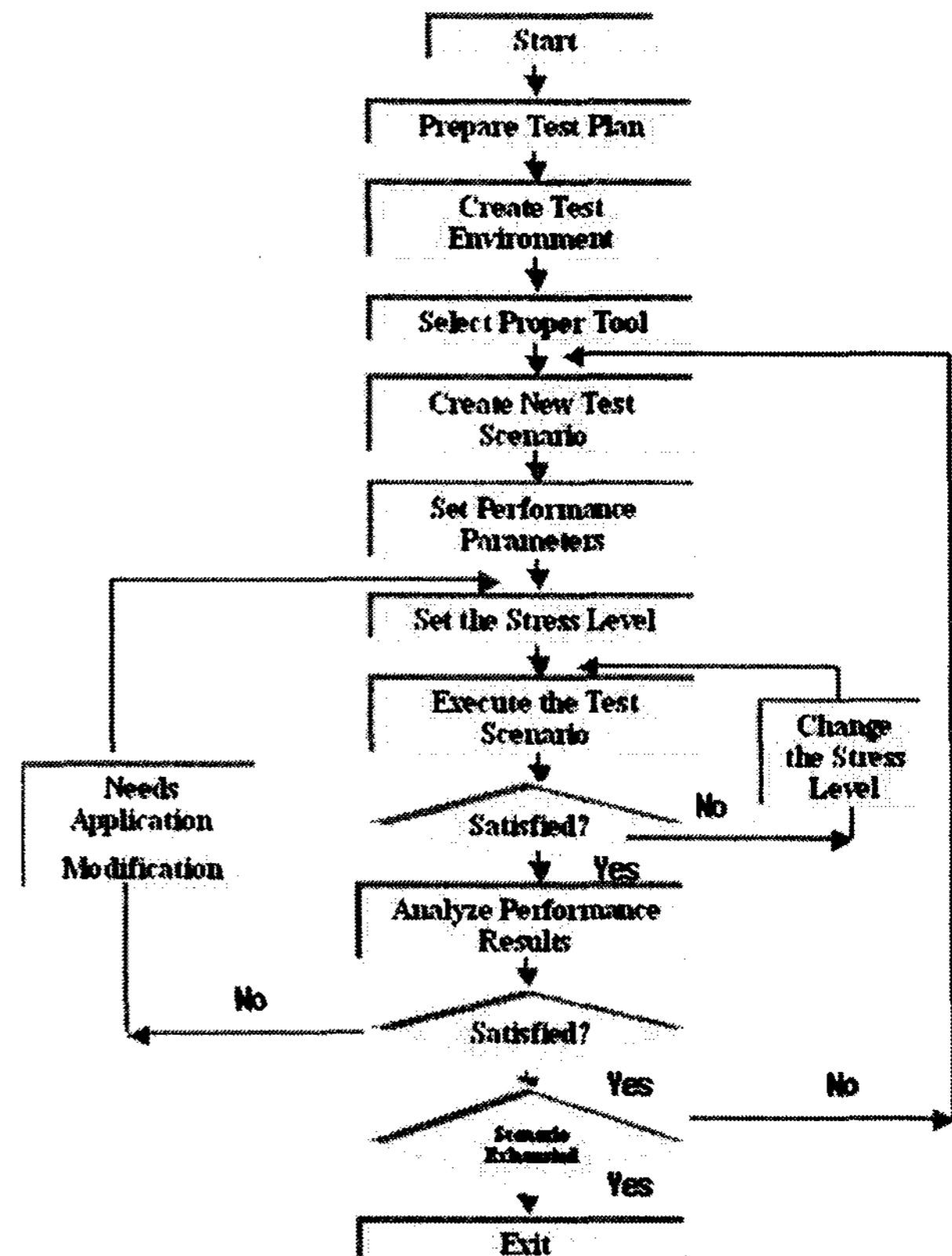
웹 어플리케이션에서 성능 테스트의 주요 목표는 다음과 같다. [1]

- 수락 가능한 성능을 지원하는 동안의 최대 동시사용자 수
- 시스템이 실패가 일어나기 직전의 최대 동시사용자 수
- 어플리케이션 아키텍처의 병목 현상 발견
- 최적화된 하드웨어와 어플리케이션 구성을 결정
- 시스템의 확장성 검증
- 성능의 요구사항을 검증

성능 테스트는 실제 사용자가 수행하는 것과 동일한 웹 사이트의 행위를 레코딩하고 스크립터화한다. 또한 자동화되고 컨트롤 되어진 방식으로 테스트 대상 시스템에 레코드 되어진 행위들을 테스트 하게 된다.

2.2 기존의 성능 테스트

기존의 성능 테스트 논문에서는 웹 어플리케이션의 특성에 맞는 성능 테스트 수행을 제시하고 있다. 다음의 [그림 1][2]과 같이 테스트를 수행하기 전에 계획을 먼저 한 뒤 테스트를 수행하기 위한 환경을 구축을 한 후 적절한 테스트 도구를 선택을 하여서 테스트를 수행을 한다. 또한 테스트가 수행된 후에 테스트된 결과를 분석을 하게 된다.



[그림 1] 웹 어플리케이션의 테스트 프로세스

2.3 기존의 성능 테스트의 문제점

기존의 성능 테스트 연구는 웹 어플리케이션의 특성에 맞게 성능 테스트를 제안하고 있으나, 실제 웹 어플리케이션의 성능 테스트의 다양한 요구사항들을 반영하고 있지는 못하고 있다.

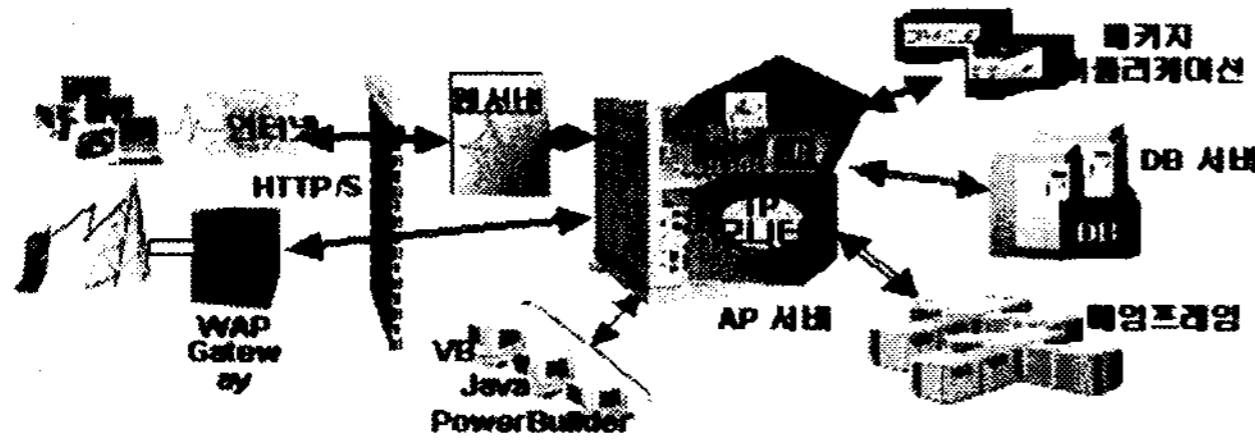
또한, 전체적인 프로세스는 제안을 하고 있으나, 실제 각 단계마다 고려해야 할 부분에 대해서는 언급이 미약하다. 따라서 본 논문에서는 성능테스트를 수행하는데 고려해야 할 점들을 좀 더 자세하게 제시를 하려고 한다.

3. 성능 시험 평가 모델

3.1 웹 어플리케이션 구조

오늘날, 웹 어플리케이션은 사용자 수의 증가와 시스템 구조의 복잡도가 커지고 있다. 다음 그림은 웹 어플리케이션의 구조[그림 2]를 나타낸다. 이 어플리케이션의 구조에서 보는 바와 같이 우리는 데이터베이스 서버, 웹 서버, 어플리케이션 서

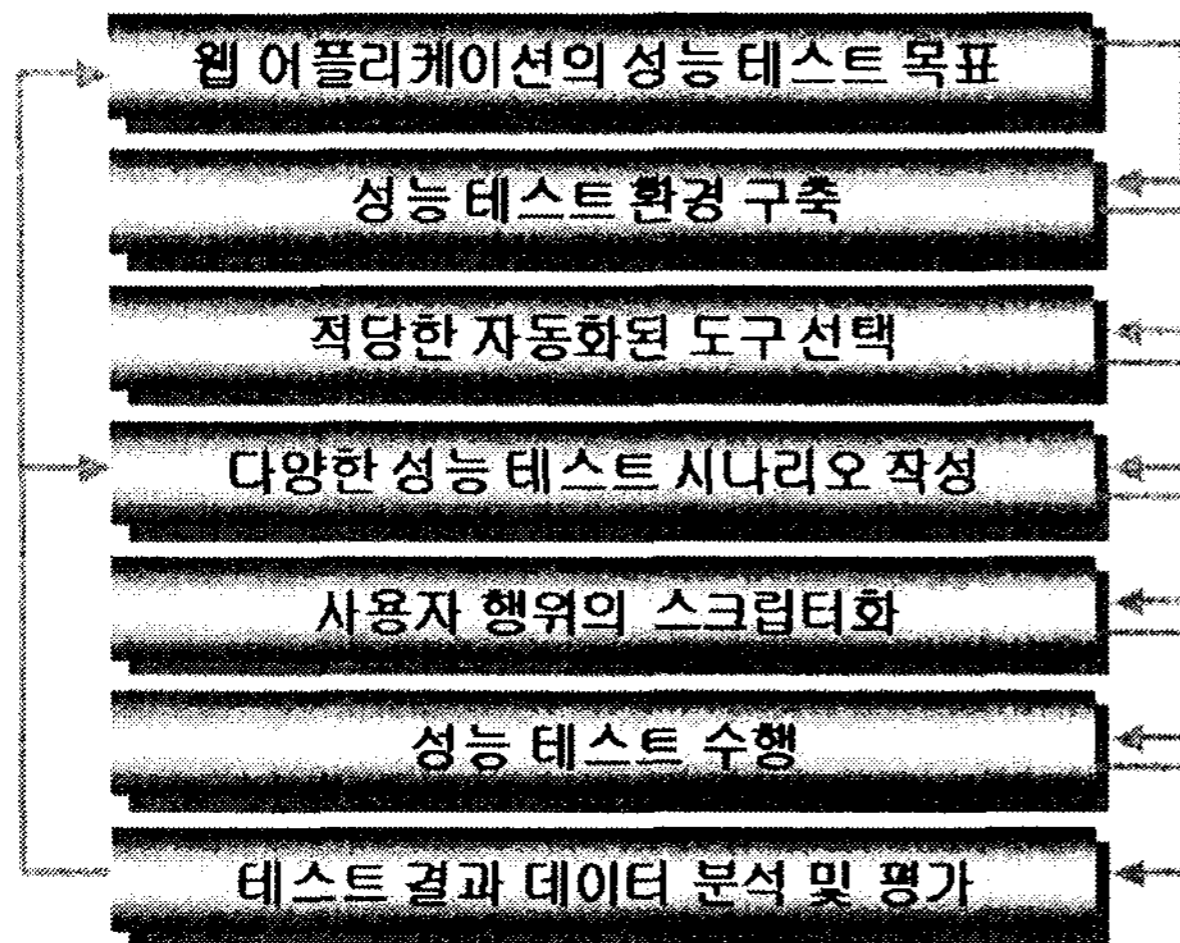
버, 클라이언트에 따른 응답 시간 등이 상당히 중요하다라는 것을 알 수 있다.



[그림 2] 웹 어플리케이션 구조

3.2 웹 어플리케이션의 성능 시험 평가 모델

웹 어플리케이션의 성능 평가 모델은 다음 [그림3]과 같다.



[그림 3] 웹 어플리케이션의 성능 평가 모델

3.2.1 성능 테스트 목표

웹 어플리케이션의 특성에 맞는 성능 테스트 목표를 식별하며, 테스트하고 싶은 목적이 무엇인지 정확하게 분석을 하여야 한다.

3.2.2 성능 테스트 환경 구축

테스트 전과정에 있어서 구축된 환경이 사용이 되어지기 때문에 매우 중요하다. 성능 테스트의 목표가 식별이 되면 그에 따른 테스트 환경을 구축 해야 한다. 다음의 고려할 사항들이 있다.

- 소프트웨어가 실행되어지는 곳은 어떤 유형의 플랫폼인지
- 시스템 아키텍처에서 얼마나 많은 티어

를 구성하고 있는지

- 웹 페이지가 정적인 페이지만으로 구성되어 있는지
- 데이터베이스에 정보를 저장하는지
- 어떠한 브라우저에서도 웹 페이지가 보이는지
- 웹 어플리케이션 개발에 쓰이는 소프트웨어, 하드웨어, 펌웨어와 동일한지

3.2.3 적당한 자동화된 도구를 선택

웹 어플리케이션에 적합한 성능 테스트 도구들은 현재 국외에 상용화되어 제품이 많이 있다. 웹 어플리케이션의 성능 테스트는 다수의 사용자가 실제 웹 사이트에서 수행하는 것과 같은 행위를 시뮬레이션 해야 하며, 좀더 빠르고 정확한 성능 테스트를 수행하기 위해서 자동화된 성능 테스트 도구는 반드시 필요하다. 웹 어플리케이션의 유형에 적합한 성능 테스트 도구를 선정하는 것이 비용면에서도 상당히 중요하다. 가상 유저 수의 정도에 따라서도 성능 테스트 도구의 비용은 차이가 크므로 신중해야 한다. 다음의 [표 1]은 성능 테스트 도구에 대한 특징을 나열한 것이다.

성능 테스트 도구	특징
Rational Software 사의 Rational Robot	<ul style="list-style-type: none"> ● 손쉬운 가상 유저 생성 ● 응답시간 측정 ● 테스트 대상이 되는 시스템에 리소스(CPU, DISK, MEMORY)를 읽어옴 ● 가상 유저 수에 따른 시스템 리소스의 시각적인 그래픽화
Mercury 사의 Road Runner	<ul style="list-style-type: none"> ● 쉬운 가상 유저를 생성 ● 부하 테스트 시나리오 작성 및 실행 ● 시스템 병목의 확인 및 조치 ● 전사적인 로드 테스트

Web Performance Trainer	<ul style="list-style-type: none"> 가상 유저수에 따른 웹 페이지 로딩 시간, 응답시간 웹 페이지의 처음 전달된 Byte에 최소, 최대, 평균 시간 초당 Hit 수 초당 Byte 수 에러가 발생한 수 가상 유저가 상용된 수
-------------------------	---

[표 1] 웹 어플리케이션의 성능 테스트 도구

3.2.4 다양한 성능 테스트 시나리오 작성

성능 테스트의 시나리오는 목표에 따라서 결정이 되어진다. 사용자의 요구사항을 충분히 반영한 성능 테스트가 요구된다.

3.2.5 사용자 행위의 스크립터화

성능 테스트의 목표에 적합한 시나리오를 작성한 뒤 최종 성능 테스트를 수행하기 위한 스크립터를 작성하게 된다. 다음의 표는 성능 테스트 스크립터 작성시에 고려되어야 할 사항들이다.

스크립터 작성시 고려할 사항	설명
가상 유저 수	실제 웹 사이트에 방문하게 될 사용자 수
가상 유저에 따른 동기화	가상 유저들의 동시 접속
분당 가상 유저 수의 증가 폭 설정	가상 유저들이 증가하게 되는 폭을 설정
Think Time	웹 사이트에서 사용자들이 머무는 시간

[표 2] 테스트 스크립터 작성 시 고려 사항

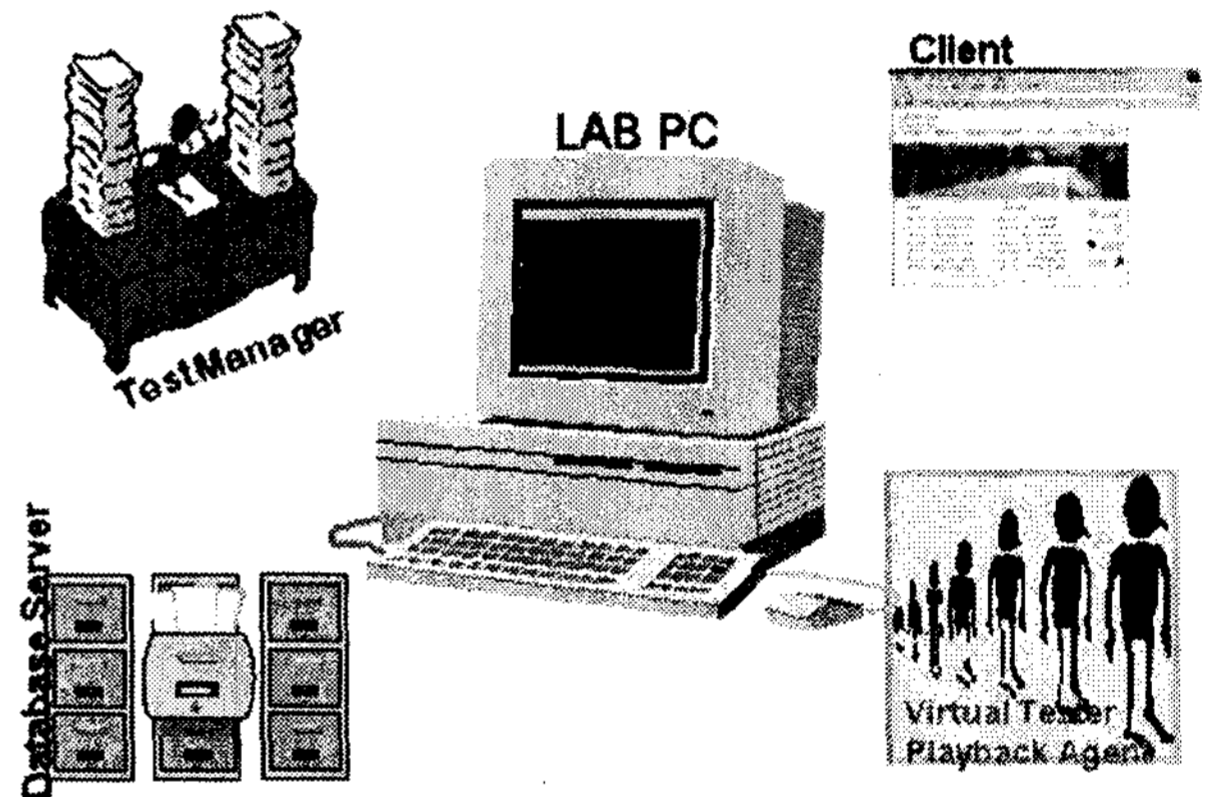
3.2.7 테스트 결과 데이터 분석 및 평가

앞에서의 스크립터가 생성이 되면 성능 테스트를 수행을 하게 된다. 성능 테스트의 수행 결과 얻어진 테스트 결과를 바탕으로 분석을 하게 된다. 얻어진 결과 데이터를 분석하여 평가를 내리게 된다. 이때, 얻어진 테스트 결과는 유용한 데이터로 남게 된다. 최초 성능 테스트의 목표와 맞지 않는 잘못된 데이터를 얻었거나, 원치 않는 결과를 얻

었다면 다시 성능 테스트의 목표를 설정하거나 시나리오를 작성을 해야 한다.

4. 사례 연구

본 성능 테스트 평가 모델을 적용하기 위한 현재 수행하고 있는 사이트를 대상으로 성능 테스트를 수행한다. 가장 많이 사용되는 웹 페이지상에서 가상 유저의 수에 따른 응답시간을 측정한다. 다음 그림은 성능 테스트를 수행하는데 필요한 자원을 나타낸다. 이에 따른 환경을 구축한다.



[그림 4] 필요한 지원

Rational Software사의 Rational Robot의 자동화된 테스트 도구를 선택하여 테스트를 수행하도록 한다.

다음 그림은 웹 상에서의 사용자 행위를 시뮬레이션을 스크립터화 한 그림이다.

```

File Edit View Record Debug Insert Tools Window Help
[Icons]
Sub Main
Dim Result As Integer

'Initially Recorded: 2802-07-02 오전 2:40:37
'Script Name: Test

Window SetContext, "Class=Shell_TrayWnd", ""
SetThinkAvg 1993
TabControl Click, "ObjectIndex=1;ItemIndex=2", ""

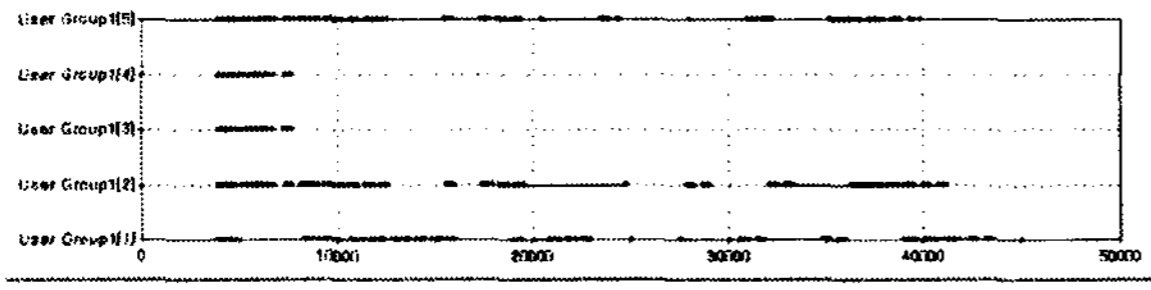
Window SetContext, "Caption=바산네트웍(주)에 오신것을 환영합니다. - Microsoft Intern
Browser SetFrame, "Type=HTMLFrame;HTMLId=head", ""
Browser NewPage, "HTMLTitle=Untitled Document", ""
SetThinkAvg 5548
HTMLImage Click, "Type=HTMLImage;Index=7", "Coords=51,26"
SetThinkAvg 1392
HTMLImage Click, "Type=HTMLImage;Name=image9", "Coords=45,18"
SetThinkAvg 971
HTMLImage Click, "Type=HTMLImage;Name=image7", "Coords=29,7"

End Sub

```

[그림 5] 사용자 시뮬레이션 스크립터

성능 테스트를 수행한 결과를 다음과 같다.



[그림 6] 응답 시간

	Cmd ID	Ending TS	Response	Status	Virtual Tester	Script
1	blank1.001	4146	220	Pass	User Group1[1]	Notepad
2	blank1.002	4336	190	Pass	User Group1[1]	Notepad
3	blank1.003	4377	231	Pass	User Group1[1]	Notepad
4	blank1.004	4796	190	Pass	User Group1[1]	Notepad
5	blank1.005	8392	3596	Pass	User Group1[1]	Notepad
6	blank1.006	8392	3596	Pass	User Group1[1]	Notepad
7	blank1.007	8592	190	Pass	User Group1[1]	Notepad
8	blank1.008	8642	250	Pass	User Group1[1]	Notepad
9	blank1.009	8842	250	Pass	User Group1[1]	Notepad
10	blank1.010	9063	221	Pass	User Group1[1]	Notepad
11	blank1.011	9284	231	Pass	User Group1[1]	Notepad
12	blank1.012	9294	231	Pass	User Group1[1]	Notepad
13	blank1.013	9524	190	Pass	User Group1[1]	Notepad
14	blank1.014	9774	190	Pass	User Group1[1]	Notepad

[그림 7] 테스트 결과 데이터

5. 결론 및 향후 연구

웹 어플리케이션의 성능 테스트에서 고려되어야 할 성능 목표, 환경 구축, 테스트 도구 선정, 시나리오작성, 스크립터화, 테스트 수행에 대한 모델을 제시하였다. 이 모델은 웹 어플리케이션의 정확하고 신뢰성 있는 성능 테스트를 수행하게 될 것이다.

향후 연구과제로서는 웹 어플리케이션의 자동화된 성능 테스트를 개발하여 좀더 정확하고 신뢰성 있는 테스트를 수행할 것이며, 다양한 성능 테스트의 목표와 시나리오를 선정하여 테스트를 수행함으로써 통계학적인 모델을 제시할 것이다.

[참고문헌]

- [1] Key Considerations in Performance Testing, Leslie Segal President, Testware Associates, Inc, 2002
- [2] Object driven Performance Testing of Web Applications, Subraya, B.M.; Subrahmanya, S.V., Quality Software, 2000. Proceedings. First Asia-Pacific Conference on , 2000 Page(s): 17 -26
- [3] Quality Web Systems, Elfriede Dustin, Jeff Rashka, Douglas McDiamid, Addison-Wesley, 2002
- [4] Testing Applications on the Web, Hung Q. Nguyen, Wiley&Sons, 2001
- [5] Rational Performance Engineering White Paper,

Rational Corporation, <http://www.rational.com>

[6] Mercury Interactive White Paper, Mercury Corporation, <http://www.mercuryinteractive.com>

[7] Web Performance Trainer Guide, <http://www.WebPerformance.com>