[별첨 7]

Glassfish
테스트 결과보고서

2010. 11.

공개SW역량프라자
OPEN SOURCE SOFTWARE COMPETENCY PLAZA
nipa 정보통신산업진흥원
National IT Industry Promotion Agency

목 차

1. 테스트 대상 소개

가. Glassfish(v3.0.1)

글래스피시(GlassFish)는 썬 마이크로시스템즈에서 개발하는 Java EE 기반 웹 애플리케이션 서버이며, 썬 社는 이 제품을 Sun Java System Application Server 9.X로 판매하고 있음

글래스피시는 썬 마이크로시스템즈와 오라클의 탑링크(TopLink)를 기반으로 하고 있으며, 물론 웹 콘텐츠를 제공하는 서블릿 컨테이너는 아파치 톰캣을 사용하면서 성능과 확장성을 높이기 위해 자바 NIO을 사용하는 그리즐리(Grizzly)라는 구성 요소를 추가하였음

※ 출처 : 공개SW TRM - 솔루션 프로파일 “솔루션 설명” 참조

2. 테스트 케이스 및 시나리오

Apache의 신뢰성을 검증하기 위하여 테스트 케이스에 기반을 둔 기능 테스트와 테스트 시나리오에 기반을 둔 비 기능 테스트를 수행한다.

가. 기능별 테스트케이스 현황

[표 2-1. 기능별 테스트케이스 현황]

기 능	테스트 케이스 수
설치 및 삭제	3
실행	2
배포	2
로깅	3
보안	6
네이밍	3
모니터링	2
클러스터링	9
가상호스트	5
데이터베이스	5
기타(Default Servlet)	4
합 계	44

나. 비 기능 테스트 시나리오

[표 2-2. 성능 테스트 시나리오]

시나리오ID	설 명
OSS_WASR_01	가상사용자별 서버 응답시간 측정
OSS_WASR_02	프로세스(쓰레드) 개수에 따른 서버 성능 측정
OSS_WASR_03	JDBC Datasource DBCP 개수에 따른 성능 측정
OSS_WASR_04	12시간 장시간 테스트를 진행하여 서버의 안정성 측정

3. 기능 테스트 결과

기능 테스트 수행관련 세부 절차 및 결과는 별첨 「Glassfish 테스트 케이스」를 참고한다.

가. 기능 테스트 결과

[표 3-1. 기능 테스트 결과]

테스트케이스	Pass	Fail	Not Available
44	39	0	5

나. 결함내역

□ 테스트 수행 중 치명적인 결함은 발견되지 않음

다. 특이사항

□ Servlet, JSP같은 기본 샘플 파일을 제공하지 않음

4. 비 기능 테스트 수행 및 결과

가. 비 기능 테스트 결과

[표 4-1. 비 기능 테스트 수행 결과]

테스트 시나리오	내용	결과
OSS_WASR_01	가상사용자별 응답시간 측정	상세내역 참조
OSS_WASR_02	프로세스(쓰레드) 수에 따른 성능 측정	프로세스 수 300에서 최적
OSS_WASR_03	DBCP 개수에 따른 성능 측정	DBCP 수 50에서 최적
OSS_WASR_04	장시간 수행을 통한 안정성 테스트	총 2,263,082번 수행

나. 비 기능 테스트 상세내역

비 기능 테스트의 경우 하드웨어 사양뿐 아니라, OS 및 애플리케이션 구성에 따라 성능 측정 결과가 상이하므로, 실제 운영 환경에서 적용할 경우 테스트 결과가 다를 수 있다.

※ 용어설명
가상사용자 : 테스트 툴에서 생성한 쓰레드 수(사용자)
Average : 테스트 평균 응답 시간
Deviation : 각각의 응답시간과 평균 응답시간과의 차이의 합
Throughput : 서버가 처리하는 작업량

□ OSS_WASR_01 - 가상사용자별 서버 응답시간을 측정한다.

o 테스트 조건
- 최대 쓰레드 수 : 200개
- 가상사용자 : 1000명, 1500명, 2000명
- 웹페이지 크기 : 100KB
- 응답시간 제한 없음

o 테스트 절차
- JMeter 툴을 실행하여 가상사용자 1000, 1500, 2000명 생성 후 테스트 대상 페이지에 접속
  ▪ Number of Threads : 1000, 1500, 2000명, Ramp-Up : 0, Loop Count : 1

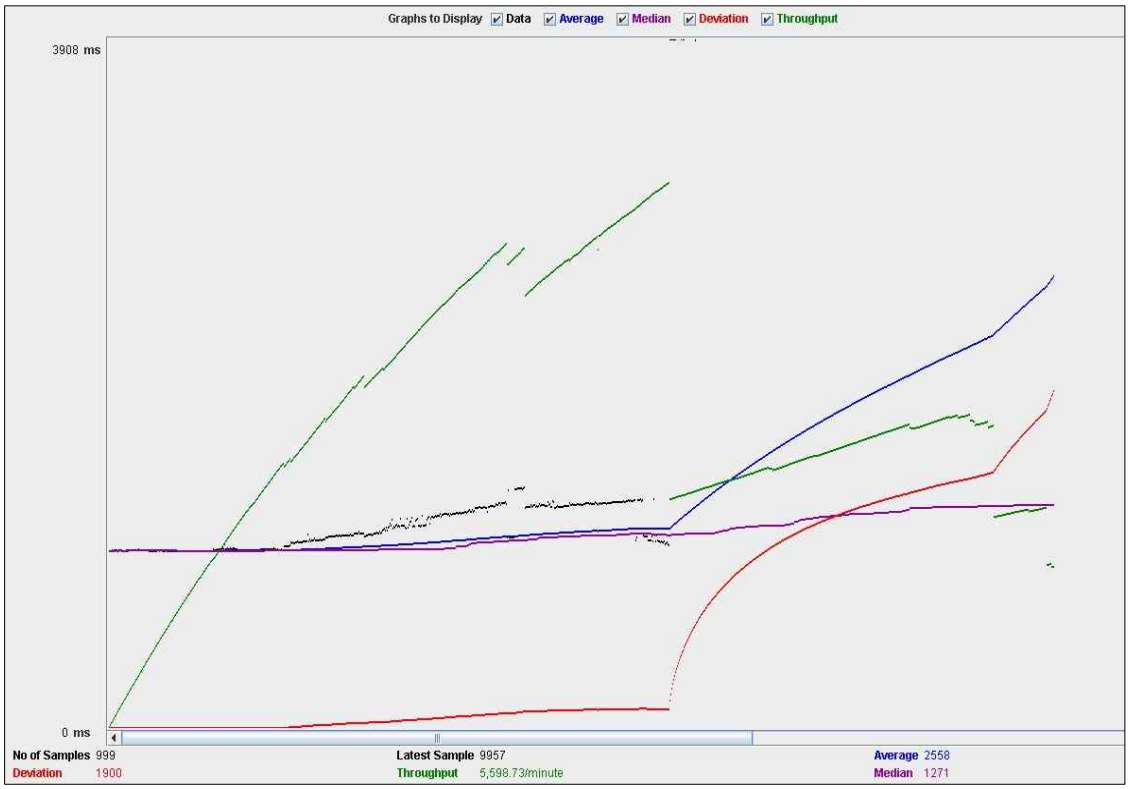
o 테스트 결과
- Summary Report

[표 4-2. 가상사용자별 응답시간]

가상사용자	평균시간/ms	최소시간/ms	최대시간/ms	에러율	TPS	KB/sec
1000	2566	1006	9957	0.00%	93.4	9020
1500	2996	1006	13268	0.00%	108.1	10442
2000	2424	1006	10033	0.00%	182	17582

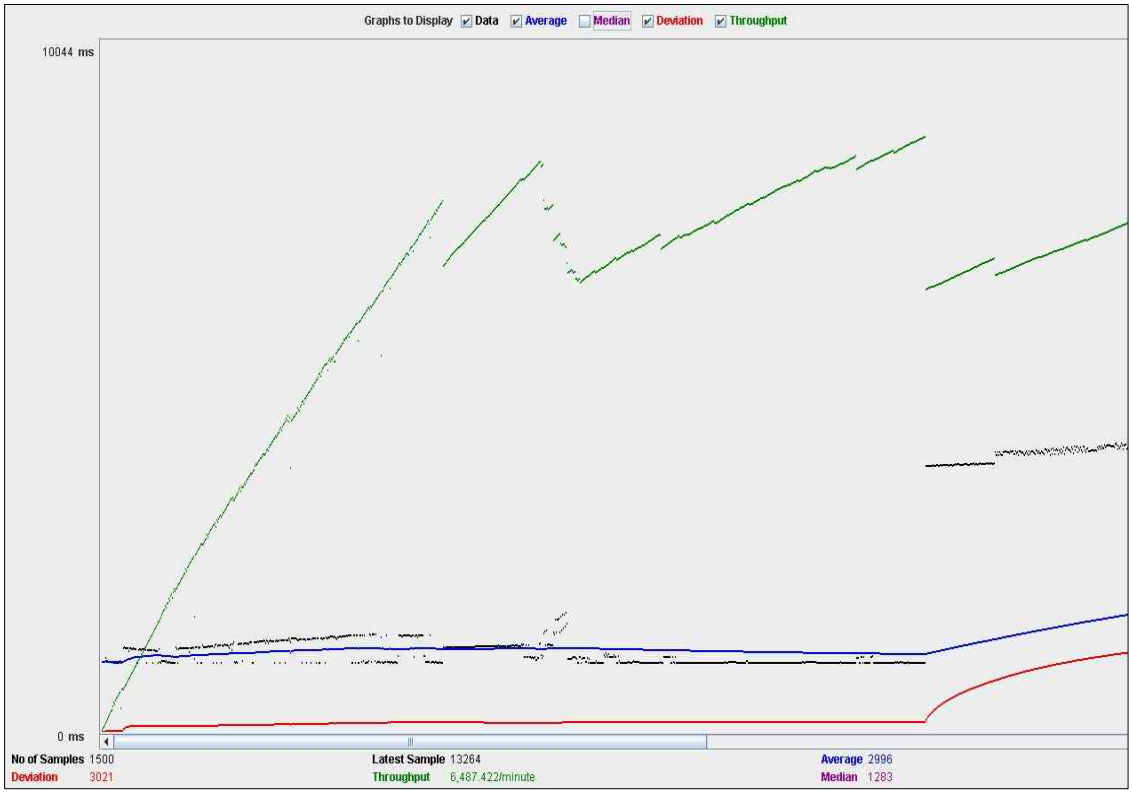
가상사용자가 증가해도 서버 응답시간 및 TPS 수치가 안정적으로 처리되는 현상을 확인할 수 있음

- Graph Results
  - 가상사용자 1000명



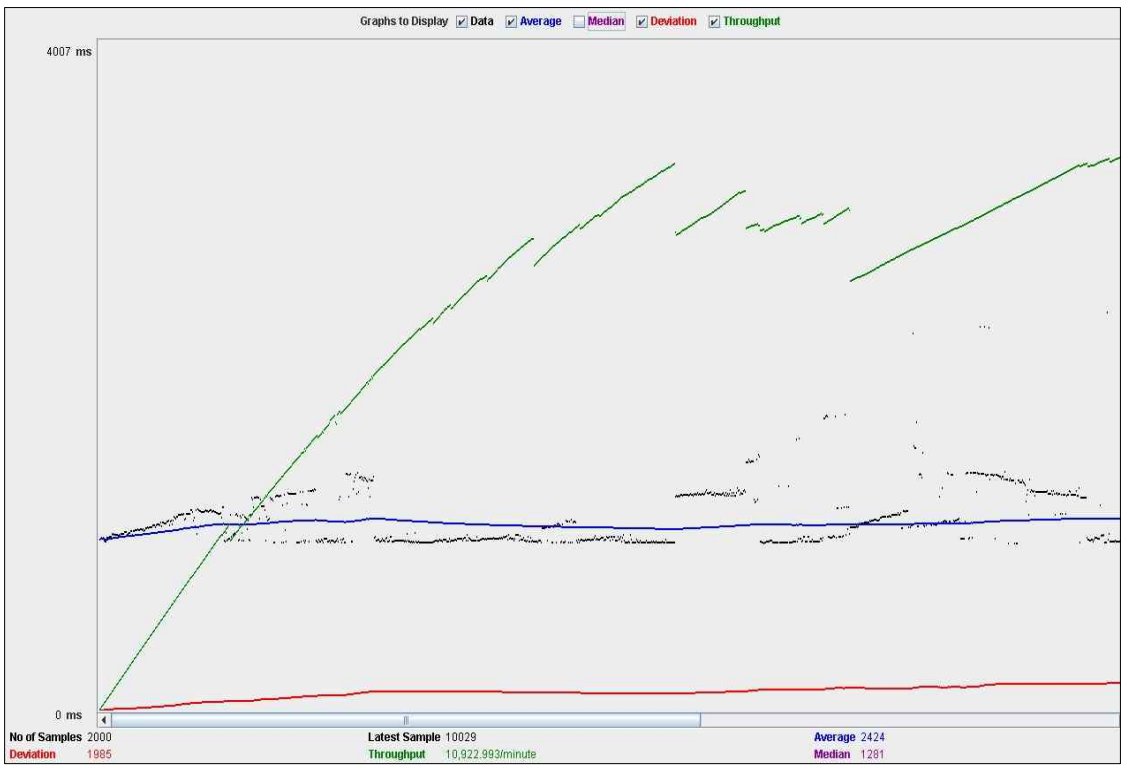
[그림4-1. 가상사용자 1000명]

▪ 가상사용자 1500명



[그림4-2. 가상사용자 1500명]

▪ 가상사용자 2000명



[그림4-3. 가상사용자 2000명]

□ OSS_WASR_02 - 프로세스(쓰레드) 개수에 따른 성능을 측정한다.

o 테스트 조건

- 쓰레드 수 : 200, 250, 300
- 가상사용자 : 1000명

o 테스트 절차

- JMeter 툴을 실행하여 가상사용자 1000명 생성 후 테스트 대상 페이지에 접속
  ▪ Number of Threads : 1000명, Ramp-Up : 0, Loop Count : 1
- 환경설정파일에서 프로세스 수를 조건에 맞게 변경 후 재시작하여 테스트

o 테스트 결과

- Summary Report

[표 4-4. 프로세스(쓰레드)별 응답시간]

쓰레드 수	평균시간/ms	최소시간/ms	최대시간/ms	에러율	TPS	KB/sec
200	2566	1006	9957	0.00%	93.4	9020
250	2319	1006	9133	0.00%	98.1	9527
300	2238	1006	9005	0.00%	100.3	9658

쓰레드 개수가 증가 할수록 일부 응답시간 및 TPS 수치가 다소 향상되는 현상을 확인 할 수 있음

□ OSS_WASR_03 - JDBC Datasource DBCP 개수에 따른 성능을 측정한다.

o 테스트 조건

- DBCP 개수 : 10, 50, 100
- 가상사용자 : 1000명
- 테스트 데이터 : 10만 건

o 테스트 절차

- JMeter 툴을 실행하여 가상사용자 1000명 생성 후 테스트 대상 페이지에 접속
  ▪ Number of Threads : 1000명, Ramp-Up : 0, Loop Count : 1
- 테스트 페이지에서 10만 건 중 100건의 데이터를 질의하여 출력

o 테스트 결과

- Summary Report

[표 4-5. DBCP 개수별 응답시간]

DBCP 수	평균시간/ms	최소시간/ms	최대시간/ms	에러율	TPS	KB/sec
10	315	2	625	0.00%	670.2	2154
50	108	2	342	0.00%	830.7	2687
100	131	2	241	0.00%	804.6	2491

DBCP 개수를 10에서 50으로 증가할 경우 평균응답시간이 3배 정도 개선되었으며, 50에서 100으로 증가할 경우 평균응답시간이 다소 늦어지는 것을 확인 할 수 있음

□ OSS_WASR_04 - 12시간 장시간 테스트를 수행하여 안정성을 측정한다.

o 테스트 조건

- 12시간 연속 서버로 테스트 페이지 요청
- 가상사용자 : 50명

o 테스트 절차

- 테스트 수행 전에 nmon 실행
  - nmon -f -t -r GlassfishTest -s60 -c750 (1분마다 실행하며, 750분간 동작)
- JMeter 툴을 실행하여 가상사용자 50명 생성 후 테스트 대상 페이지에 접속
  - Number of Threads : 50명, Ramp-Up : 60, Loop Count : forever
  - Scheduler : 12시간 실행되도록 설정

o 테스트 결과

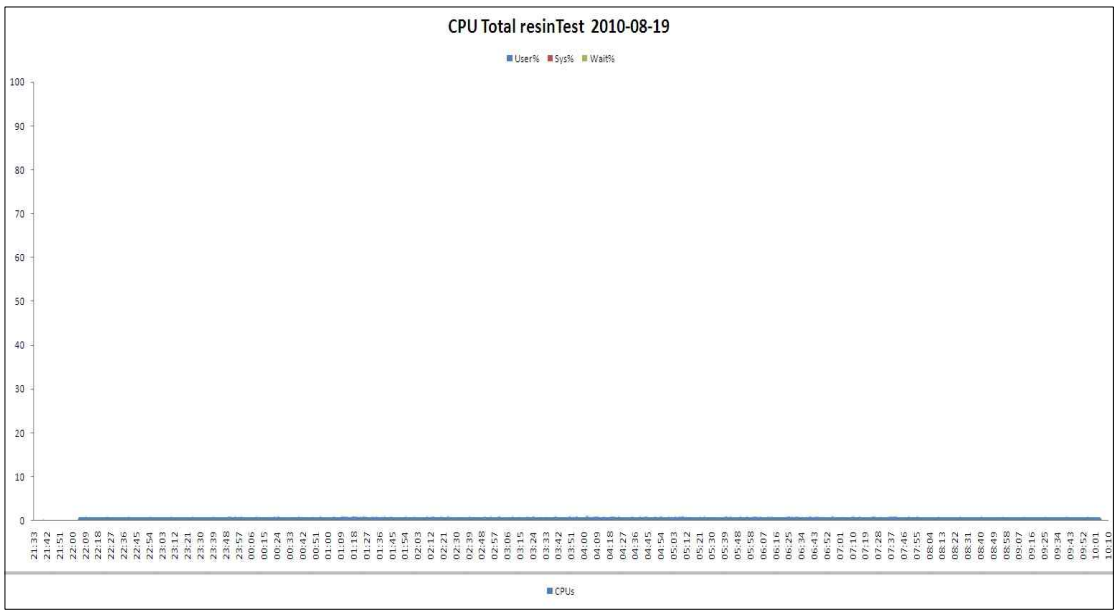
- Summary Report

[표 4-6. 12시간 테스트 결과]

총실행건수	평균시간/ms	최소시간/ms	최대시간/ms	에러율	TPS	KB/sec
2263082	1009	1003	1998	0.00%	49.3	4792

- 서버 자원 현황

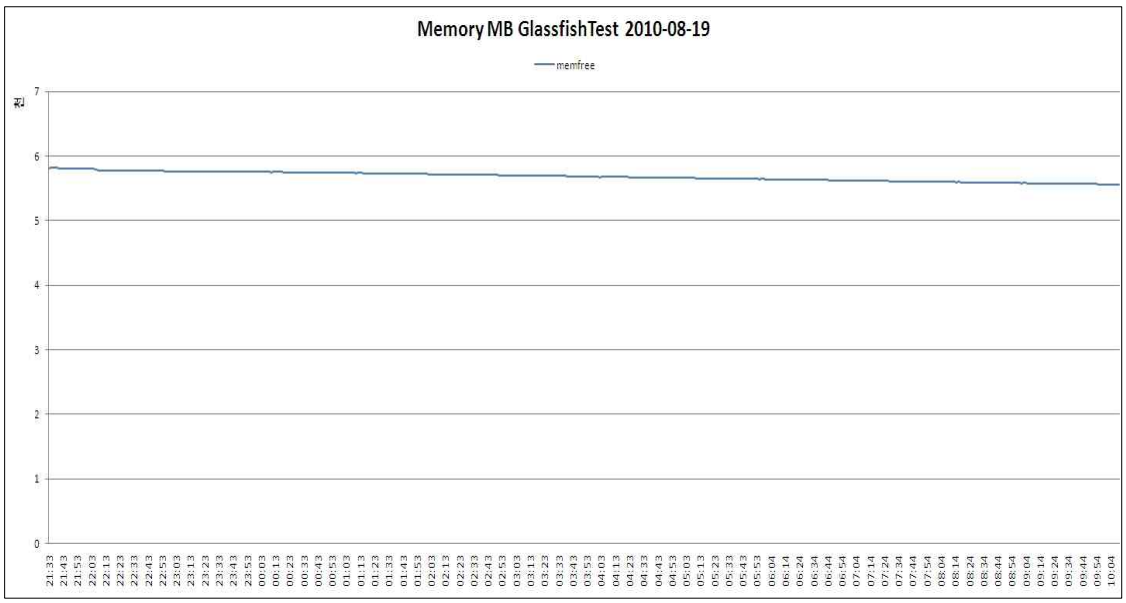
▪ CPU 사용률



[그림4-4. CPU 사용률]

cpu 점유율이 50% 이하로 형성되는 안정적인 모습을 보이고 있음

▪ 메모리 사용량



[그림5-20. 메모리 사용량]

메모리 사용량 변화가 일정하게 형성되는 안정적인 모습을 보이고 있음

5. 종합

□ 테스트 케이스 기반 기능 테스트에 대한 테스트 결과 치명적인 결함이 발생하지 않고 정상적으로 동작함

□ 비 기능 테스트에 대한 테스트 결과 및 여러 가지 상황의 테스트 조건에 대한 테스트 결과 치명적인 결함은 발생하지 않았으며, 전반적인 서버 자원 사용률도 안정적으로 사용함

※ 참고 자료

[1] https://glassfish.dev.java.net

[2] http://jakarta.apache.org/jmeter/

[3] https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-126-17%5E8_4000_100

[4] http://nmon.sourceforge.net/

[5] http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmonanalyser