공개SW-09-05

공개SW Test 결과 보고서

공개SW 역량프라자 운영 사업

2009. 12. 15

주관연구기관 **정보통신산업진흥원**

지 식 경 제 부

| CentOS

목　　　차

1. 개요

본 문서는 공개SW 역량프라자 운영 사업의 공개SW 테스트 환경 구축을 통하여 테스트한 공개SW에 대한 기능, 성능, 호환성 등을 검증하고, 향후 공공 기관 및 기업에서 도입 시 참고할 수 있는 자료로 활용하기 위하여 작성되었다.

가. 목적

□ 공개SW를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 공개SW의 기능, 성능, 안정성, 호환성 등 검증
- 공개SW 기술의 도입 가이드라인 제공
- 효과적인 공개SW 도입 유도
- 안정적인 공개SW 시스템 도입

나. 선정 기준

□ 공개SW 테스트 대상은 다음 5가지 사항을 고려하여 선정되었다.

- 1번 : 국내에서 개발된 제품
- 2번 : 국내 기업에서 많이 도입되는 제품
- 3번 : Global에서 많이 알려진 제품
- 4번 : 기술지원(업데이트) 가능한 제품
- 5번 : OS간 호환 가능 제품(Package 관리 방식)

[표 1-1. 선정기준]

Linux 운영체제(OS)	1번	2번	3번	4번	5번	결과	비고
Asianux	●	◕	◕	●	●	●	선정*
CentOS	○	◑	●	●	●	◕	선정
Debian	○	◔	◕	●	◔	◑	보류
Fedora	○	◑	●	●	●	◕	선정
Gentoo	○	◔	◕	◕	◔	◑	보류
GINUX	●	◑	◔	●	●	◕	선정
Red Hat Enterprise Linux	○	●	●	●	●	●	선정*
Slackware	○	◔	◕	◕	◔	◑	보류
SULinux	●	◑	◔	●	●	◕	선정
SuSE Linux	○	◑	●	●	●	◕	선정*
Ubuntu	○	◑	●	●	◔	◑	보류

※ 테스트 대상 기준에 근거하여 이번 테스트 대상에서 제외된 **선정*** 표시 제품은 금번 사업의 일정상의 제약으로 다음 사업에서 테스트를 수행하기로 함

2. 테스트 환경 구성

가. 테스트 일정

[표 2-1. 일정표]

구분	월	10				11				12			
	주	1	2	3	4	1	2	3	4	1	2	3	4
테스트 대상 발굴 및 선정													
테스트 환경 구성													
테스트 항목 선정													
테스트 시나리오 작성													
테스트 수행													

나. 테스트 대상 소개

□ CentOS 5.3

CentOS(The Community ENTerprise Operating System)는 리눅스계 운영체제 가운데 하나다. 레드햇 엔터프라이즈 리눅스의 소스코드를 그대로 가져와 빌드해 내놓으며, 이 과정에서 달라지는 점은 레드햇의 상표 대신에 CentOS의 상표가 붙는(상표권 분쟁을 피하기 위해) 것이며 추가적인 기능의 패키지를 Extra 패키지로 제공하고 있다. 버전 또한 레드햇 엔터프라이즈 리눅스의 버전을 그대로 가져오며(소수점 아래 숫자는 업데이트 차수를 가리킨다) 오늘날에는 레드햇 엔터프라이즈 리눅스의 행보를 가장 잘 따라가는 운영체제로 알려져 있다.

□ CentOS 5.3 주요 특징

- 커널과 성능
  - x86-64 환경의 가상화 확장성 강화
  - 인텔 코어 i7 (네할렘) 프로세서 지원
  - 리눅스 2.6.18커널 기반 제공
  - 멀티코어 프로세서 지원
  - 넓은 범위의 새로운 하드웨어 지원
  - Kexec/Kdump에 의해 제공되는 crash dump 성능의 업데이트
  - 인텔 네트워크 가속 기술(IOAT) 지원
  - 엔터프라이즈 SMP 시스템의 향상된 수많은 기능 지원
  - 향상된 파이프 버퍼링
  - IPv4/IPv6 fragmentation 단편화 offload 및 버퍼 관리
  - 동적 switchable per-queue I/O schedulers

▪ 향상된 입출력버퍼 작업을 위한 커널버퍼 splice 성능

- 보안
  ▪ SELinux의 향상은 다양한 수준의 보안과 모든 서버에 대한 목표 정책들을 포함
  ▪ SEtroubleshooter GUI는 SE리눅스 관리를 간편화 함
  ▪ 디렉터리와 안전성의 통합
  ▪ IPSEC 향상은 보안과 성능을 발전시킴
  ▪ Canary word라 불리는 프레임 같이, ExecShield 향상은 해커의 공격도 안전하게 막아 줌
  ▪ 새로운 audit특징은 강력한 서치/리포팅과 실시간 모니터링을 제공
- RHEL5와 100% 호환성 제공
- 네트워킹과 상호운용성
  ▪ 네트워크 스토리지 향상은 Autof, FS-Cache,iSCSI지원을 포함함
  ▪ IPv6 지원과 적합성 향상
  ▪ 마이크로 소프트 파일/프린트와 동적 디렉터리 통합의 향상
- 관리
  ▪ 많은 인스톨러 개선이 시스템 구성을 간단하게 함
  ▪ Red Hat Network를 위한 YUm/Pup 기반 업데이터
  ▪ Conga 클러스터와 저장장치 관리(Advanced Platform을 통한)

다. 테스트 환경

□ 테스트 서버

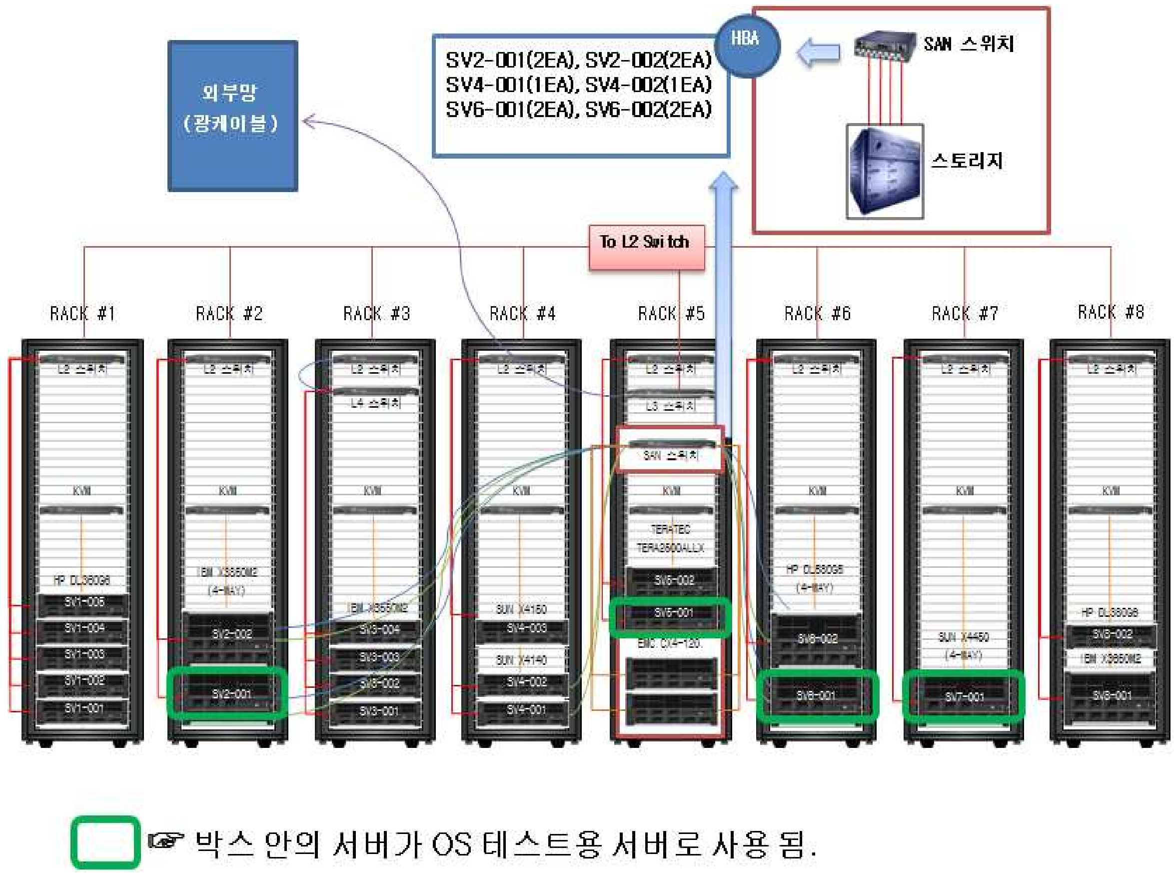
[표 2-2. 테스트 서버 규격]

제조사	모델명	CPU	MEM	Disk	NIC
HP	DL580 G5	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
IBM	X3850 M2	Quad-Core 2.4Ghz 4-Way	8GB	146GB*4	Gigabit 2Port
SUN	X4450	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
TERATEC	TERA2500ALLX	Quad-Core 2.5hz 2-Way	8GB	147GB*5	Gigabit 2Port

□ 테스트 조건

- 32bit 환경에서 64bit환경으로 서버 구성이 전환됨에 따라 64bit 환경에서의 테스트를 수행
- 성능 테스트는 HP DL580 G5 서버에서 수행

☐ Testbed 실장도



[그림 2-1. 테스트 서버 실장도]

☐ 장비 위치

[표 2-3. 서버 위치]

제조사	모델명	Rack 위치	비 고
IBM	X3850 M2	#2	
TERATEC	TERA2500ALLX	#5	
HP	DL580 G5	#6	
SUN	X4450	#7	

3. 테스트 시나리오

가. 설치 및 호환성 검증

□ DVD-ROM을 통한 설치

□ 하드웨어 정보 확인

□ 4대의 하드웨어 벤더별 설치 상태 확인

나. 기능 테스트

□ 네트워크 서비스 지원

- Web, Ftp, NFS, Samba, Openssh, Telnet

□ 파일시스템 지원

- ext3, XFS, GFS 등

□ LVM 지원 확인

□ 보안 기능 지원

- tcp-wrapper
- GUI용 보안 툴

□ 자동 업데이트 기능 지원

- On-Line 업데이트

다. 신뢰성 테스트

□ LTP Tool 사용한 테스트

- ltp-full-20090930 버전 사용

라. 안정성 테스트

□ LTP Stress Tool 사용

- CPU, Memory 등의 사용량 체크

4. 테스트

가. 설치 및 호환성 검증

□ CentOS 설치 환경

- CentOS 설치 버전

[표 4-1. OS 설치 정보]

OS 버전	지원 유형	Kernel 버전
CentOS 5.3	X86_64(EM64T, AMD64)	kernel-2.6.18-128.el5

- 파티션 정보

[표 4-2. 파티션 정보]

디스크 (RAID)	Physical Volume	Volume Group	Logical Volume	Mount Point	용량 (MB)
/dev/sda(580GB) RAID 5	/dev/sda1	-	-	/boot	100
	/dev/sda2	VG0	lv_root	/	20480
			lv_swap	-	8192
			lv_usr	/usr	30720
			lv_home	/home	20480
			lv_var	/var	10240
	/dev/sda3	VG1	lv_data1	/data1	251200
			lv_data2	/data2	51200
			lv_data3	/data3	51200

※ 디스크 용량은 서버별 차이가 있으므로 위의 기준으로 백분율(%) 환산 적용하며 남는 공간은 여유 공간으로 비워 둠

□ 서버별 설치 및 호환성 검증

[표 4-3. OS 설치 방법 및 결과]

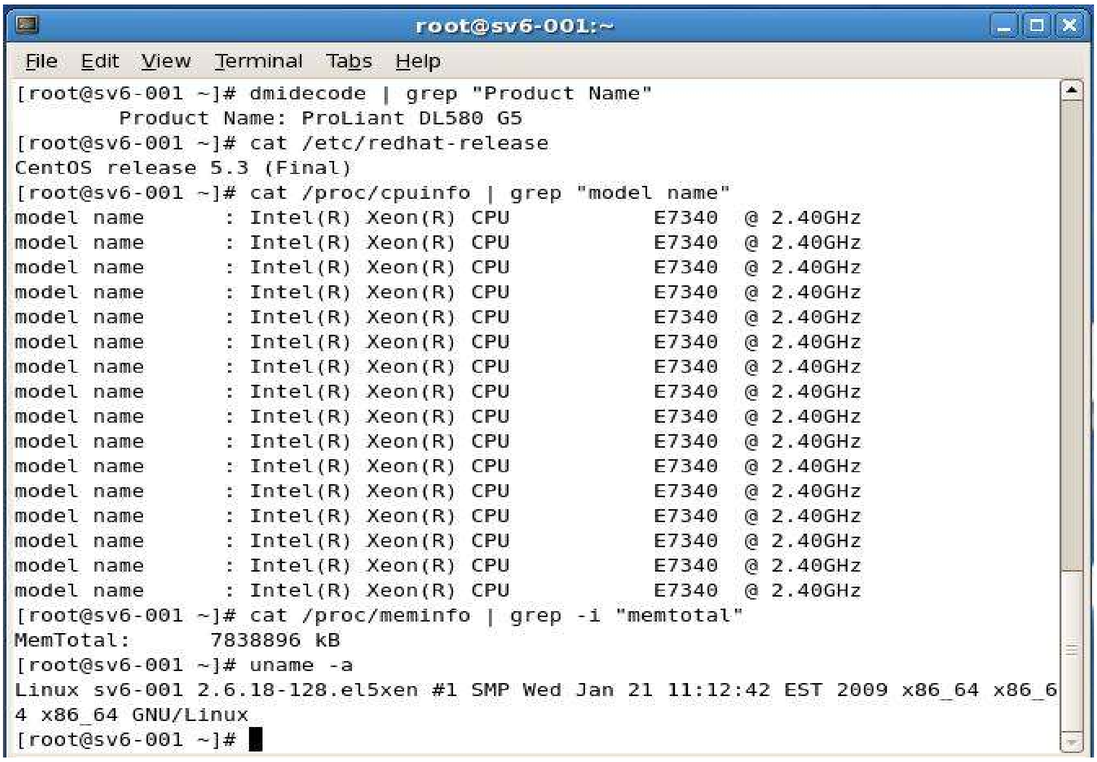
구분	설치 매체	설치 방식	패키지 선택 방식	설치 결과
HP DL580 G5	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
IBM X3850 M2	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
SUN X4450	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
TERATEC TERA2500ALLX	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상

- 서버별 CentOS 설치 상태 확인 명령

[표 4-4. GINUX 설치 상태 확인]

구분	명령어	결과
서버 모델명 확인	dmidecode | grep "Product Name"	[그림4-1~4] 참고
OS 버전 확인	cat /etc/ginux-release	
CPU 정보	cat /proc/cpuinfo | grep -i "model name"	
Memory 정보	cat /proc/meminfo | grep -i "memtotal"	
Kernel 정보	uname -a	

▪ HP DL580 G5 모델에서 CentOS 설치 확인



[그림 4-1. HP DL580G5에서 CentOS 상태 확인]

▪ IBM X3850 M2 모델에서 CentOS 설치 확인

```
root@sv2-001:~
File  Edit  View  Terminal  Tabs  Help
[root@sv2-001 ~]# dmidecode | grep "Product Name"
        Product Name: IBM 3850 M2 / x3950 M2 -[72335RK]-
        Product Name: Node1 Processor Card
        Product Name: Node1 PCI I/O Planar
        Product Name: Node1 Memory Card1
        Product Name: Node1 Memory Card2
        Product Name: Node1 Memory Card3
        Product Name: Node1 Memory Card4
[root@sv2-001 ~]# cat /etc/redhat-release
CentOS release 5.3 (Final)
[root@sv2-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
[root@sv2-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:       8177688 kB
[root@sv2-001 ~]# uname -a
Linux sv2-001 2.6.18-128.el5 #1 SMP Wed Jan 21 10:41:14 EST 2009 x86_64 x86_64 x
86_64 GNU/Linux
[root@sv2-001 ~]#
```

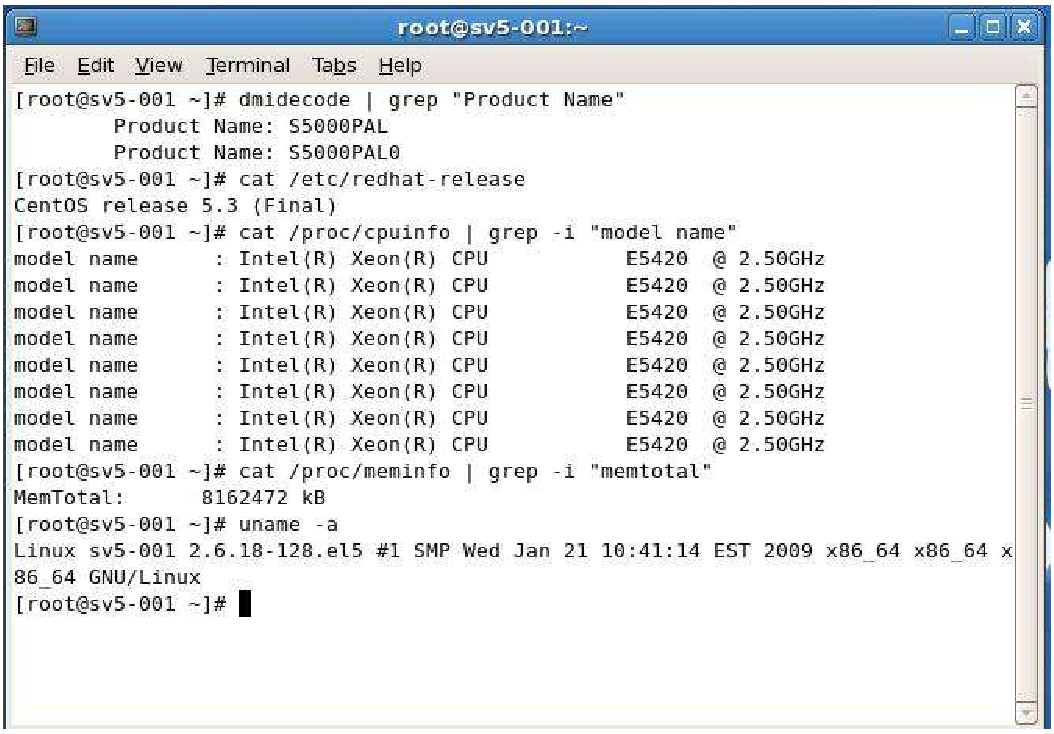
[그림 4-2. IBM X3850M2에서 CentOS 상태 확인]

▪ SUN X4450 모델에서 CentOS 설치 확인

```
root@sv7-001:~
File  Edit  View  Terminal  Tabs  Help

[root@sv7-001 ~]# dmidecode | grep "Product Name"
        Product Name: SUN FIRE X4450
        Product Name: SUN FIRE X4450
[root@sv7-001 ~]# cat /etc/redhat-release
CentOS release 5.3 (Final)
[root@sv7-001 ~]# cat /proc/cpuinfo | grep "model name"
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
[root@sv7-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:       8181544 kB
[root@sv7-001 ~]# uname -a
Linux sv7-001 2.6.18-128.el5 #1 SMP Wed Jan 21 10:41:14 EST 2009 x86_64 x86_64 x
86_64 GNU/Linux
[root@sv7-001 ~]#
```

[그림 4-3. SUN X4450에서 CentOS 상태 확인]

▪ TERATEC TERA2500ALLX 모델에서 CentOS 설치 확인



```
root@sv5-001:~
File Edit View Terminal Tabs Help
[root@sv5-001 ~]# dmidecode | grep "Product Name"
        Product Name: S5000PAL
        Product Name: S5000PAL0
[root@sv5-001 ~]# cat /etc/redhat-release
CentOS release 5.3 (Final)
[root@sv5-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
[root@sv5-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:      8162472 kB
[root@sv5-001 ~]# uname -a
Linux sv5-001 2.6.18-128.el5 #1 SMP Wed Jan 21 10:41:14 EST 2009 x86_64 x86_64 x
86_64 GNU/Linux
[root@sv5-001 ~]#
```

[그림 4-4. TERATEC TERA2500ALLX에서 CentOS 상태 확인]

※ 참고

dmidecode 명령은 하드웨어 업체에서 정보를 제공하는 경우에만 정상적인 결과 값을 얻을 수 있으며, 테라텍 서버의 경우 모델명이 일치하지 않음

나. 기능 테스트

□ 네트워크 서비스 지원

[표 4-5. CentOS 네트워크 서비스 확인]

서비스 명	패키지 명	버전	지원 여부	비고
WEB	httpd	2.2.3-22	지원	
FTP	vsftpd	2.0.5-12	지원	
NFS	nfs-utils	1.0.9-40	지원	
Samba	samba	3.0.33-3.7	지원	
SSH	openssh	4.3p2-29	지원	
Telnet	telnet-server	0.17-39	지원	

- 서비스별 지원 확인 명령

[표 4-6. 서비스별 지원 명령]

구분	명령어	결과
서비스 확인 명령	rpm -qa | grep "*package_name*"	[그림4-5] 참고

▪ 서비스별 지원

[그림 4-5. CentOS 서비스 지원 확인]

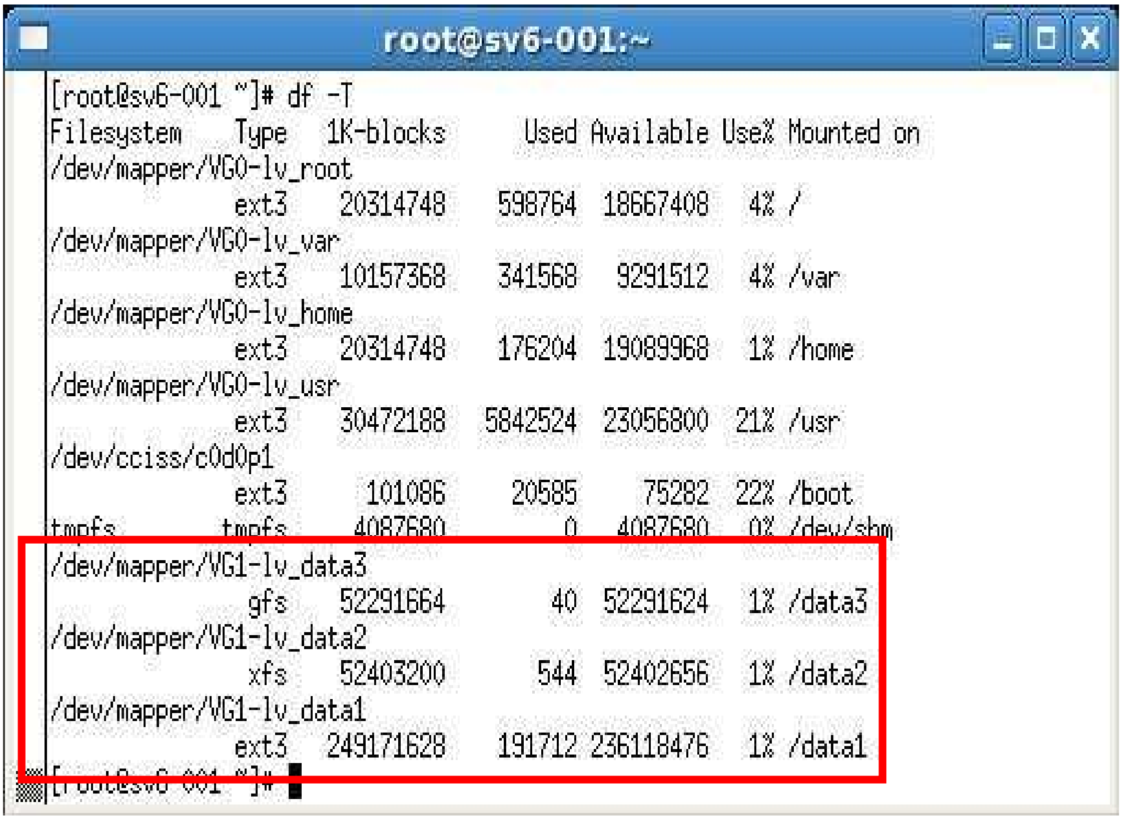
□ 파일시스템 지원 확인

- 파일 시스템 지원 확인 명령

[표 4-7. 파티션별 지원 확인 명령]

구분	명령어	결과
ext3	mkfs -t ext3 /dev/VG1/lv_data1 mount /dev/VG1/lv_data1 /data1	[그림4-6] 참고
xfs	mkfs -t xfs /dev/VG1/lv_data2 mount /dev/VG1/lv_data2 /data2	
gfs	gfs_mkfs -p lock_nolock -j 1 /dev/VG1/lv_data3 mount /dev/VG1/lv_data3 /data3	
파일시스템 확인 명령	df -T	

▪ ext3, xfs, gfs 파일시스템



```
root@sv6-001:~
[root@sv6-001 ~]# df -T
Filesystem    Type   1K-blocks      Used Available Use% Mounted on
/dev/mapper/VG0-lv_root
              ext3    20314748    598764  18667408   4% /
/dev/mapper/VG0-lv_var
              ext3    10157368    341568   9291512   4% /var
/dev/mapper/VG0-lv_home
              ext3    20314748    176204  19089968   1% /home
/dev/mapper/VG0-lv_usr
              ext3    30472188   5842524  23056800  21% /usr
/dev/cciss/c0d0p1
              ext3      101086     20585     75282  22% /boot
tmpfs        tmpfs     4087680         0   4087680   0% /dev/shm
/dev/mapper/VG1-lv_data3
               gfs    52291664        40  52291624   1% /data3
/dev/mapper/VG1-lv_data2
               xfs    52403200       544  52402656   1% /data2
/dev/mapper/VG1-lv_data1
              ext3   249171628    191712 236118476   1% /data1
[root@sv6-001 ~]#
```

[그림 4-6. 파티션별 지원 확인]

□ LVM 지원 확인

[표 4-8. LVM 지원 확인]

패키지명	버전	지원여부	비 고
lvm2	2.02.40-6	**지원**	

- LVM 확인 명령

[표 4-9. LVM 지원 확인 명령]

구분	명령어	결과
LVM 확인 명령	rpm -qa | grep "*package_name*"	[그림4-7] 참고

▪ LVM 확인

[그림 4-7. LVM 지원 확인]

※ LVM(Logical Volume Manager)

저장장치들을 좀더 효율적이고 유연하게 관리할 수 있는 커널의 부분 프로그램

- LVM 사용의 장점
  - 쉬운 관리
  - 서로 다른 많은 디바이스 조합 지원
  - 직관적인 저장 장치로의 접근
  - 뛰어난 확장성
  - 신뢰되는 안정성과 효율성
  - 스냅샷 기능 지원
  - 대용량 단일 파일시스템 구성 지원

□ 보안 기능 지원

[표 4-10. 보안 기능 지원 확인]

패키지명	버전	지원여부	비　고
tcp_wrappers	7.6-40.6	지원	
system-config-securitylevel	1.6.29.1-2.1	지원	

- 보안 기능 확인 명령

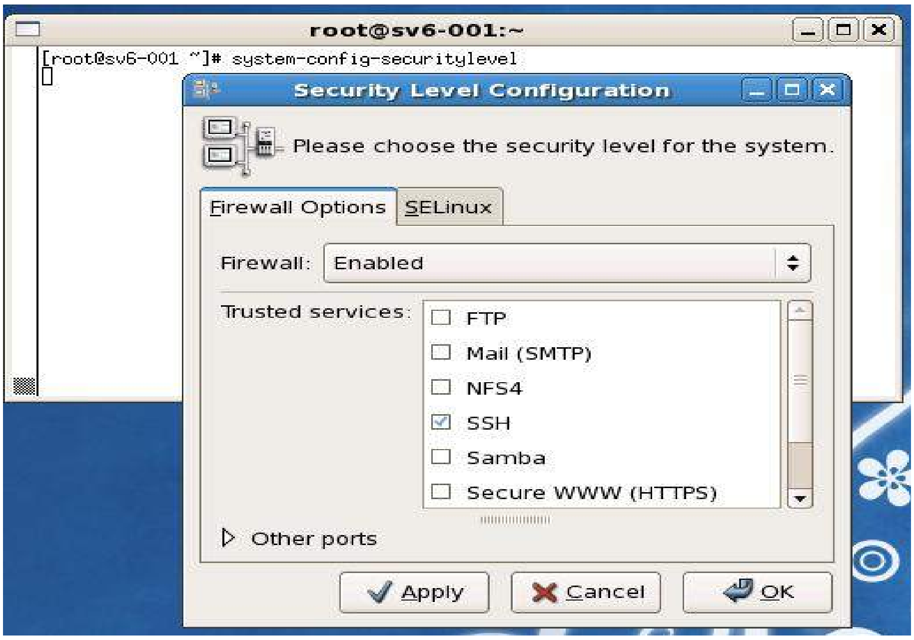
[표 4-11. 보안 기능 지원 명령]

구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "*package_name*"	[그림4-8,4-9] 참고
GUI용 명령	system-config-securitylevel	

▪ 패키지 확인

[그림 4-8. 보안 기능 지원 확인]

▪ GUI용 보안 툴 명령



[그림 4-9. GUI 보안 기능 지원 확인]

□ 자동 업데이트 기능 지원

[표 4-12. 업데이트 지원 확인]

패키지명	버전	지원여부	비 고
yum	3.2.19-18	**지원**	

- 업데이트 기능 확인 명령

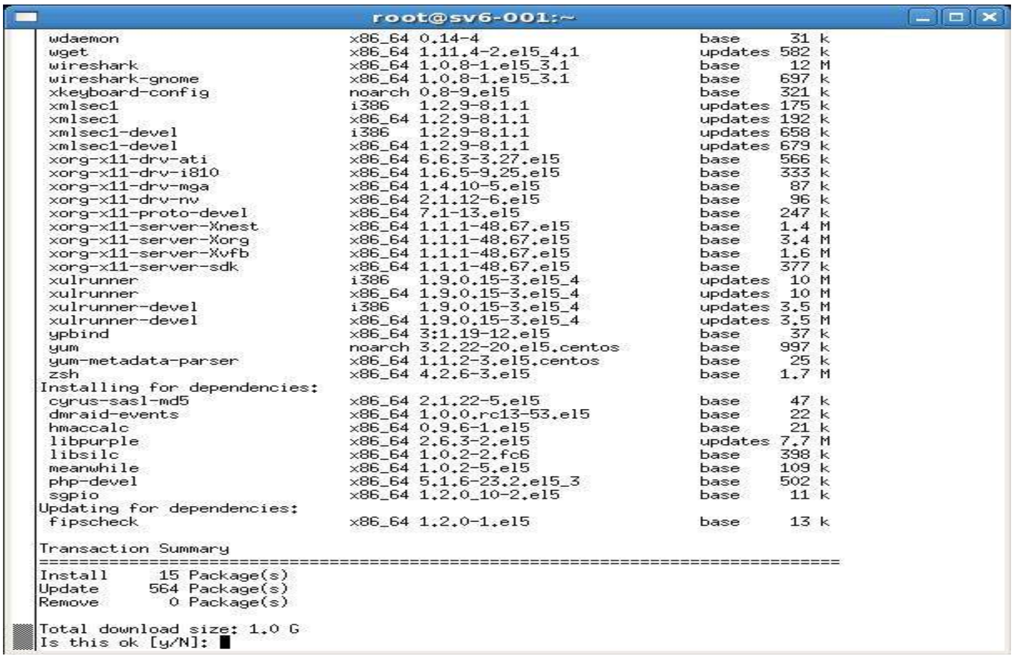
[표 4-13. 업데이트 지원 명령 확인]

구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "*package_name*"	[그림4-10~13] 참고
GUI용 명령	pirut, pup	

▪ 패키지 확인

[그림 4-10. 업데이트 기능 지원 확인]

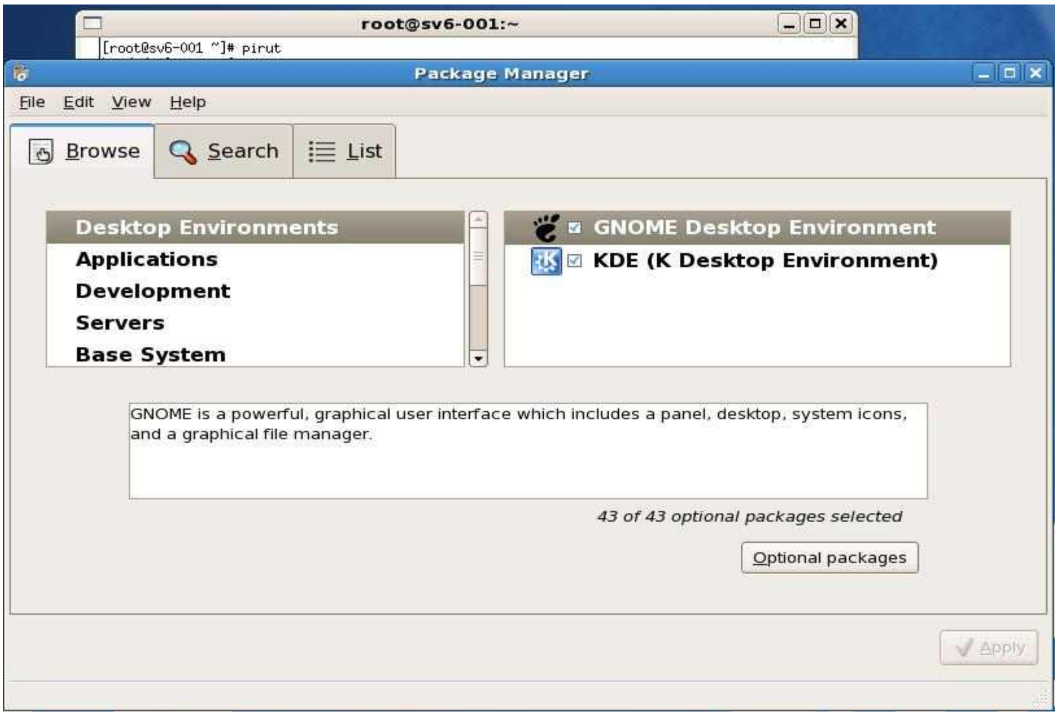
▪ 업데이트 실행 확인 (yum update)



[그림 4-11. 업데이트 기능 실행 확인]

▪ pirut

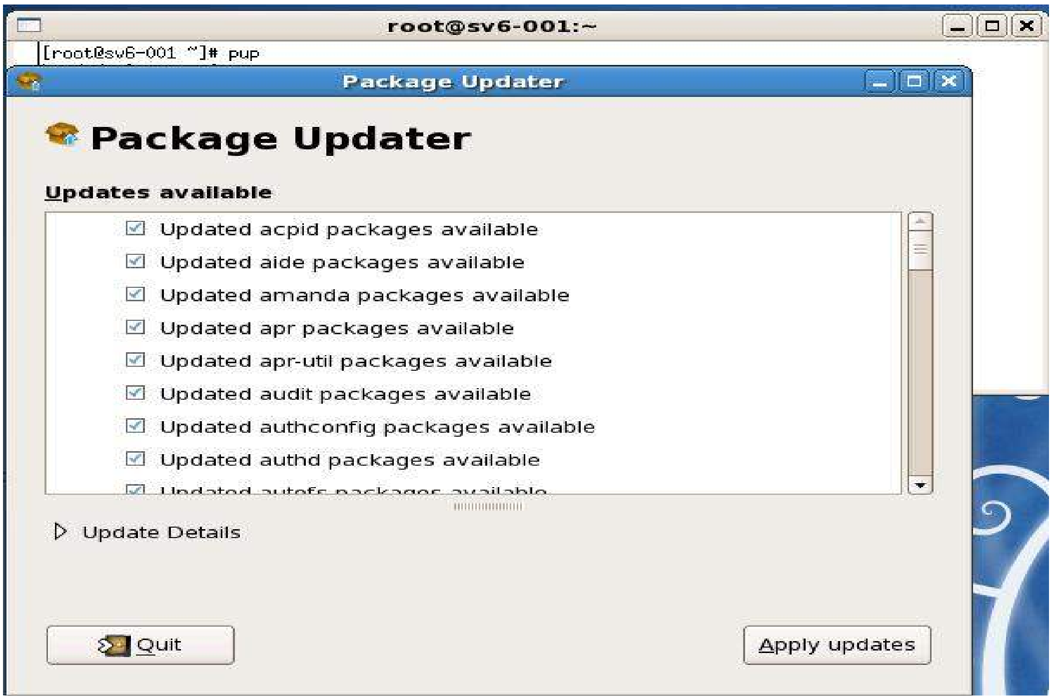
GUI용 소프트웨어 설치 및 삭제 유틸리티



[그림4-12. GUI용 패키지 설치/삭제 기능 실행 확인]

▪ pup

GUI용 패키지 업데이트



[그림 4-13. GUI용 업데이트 기능 실행 확인]

다. 신뢰성 테스트

□ LTP Tool 사용

Linux Test Project (LTP)는 리눅스의 신뢰성, 호환성, 안정성을 입증하여 오픈소스 커뮤니티에 테스트 도구를 제공하는 것을 주 목적으로 하는 오픈소스 프로젝트이다.

- 테스트 방법

runalltests.sh 스크립트를 통한 초기 기본 테스트

▪ 파일시스템 스트레스 테스트
▪ 디스크 I/O 테스트
▪ 메모리 관리 스트레스 테스트
▪ IPC 스트레스 테스트
▪ 스케줄러 테스트
▪ 명령어 함수 확인 테스트
▪ 시스템 호출 함수 확인 테스트

- 실행 방법

```
$ cd /opt/ltp
$ ./runalltests.sh
```

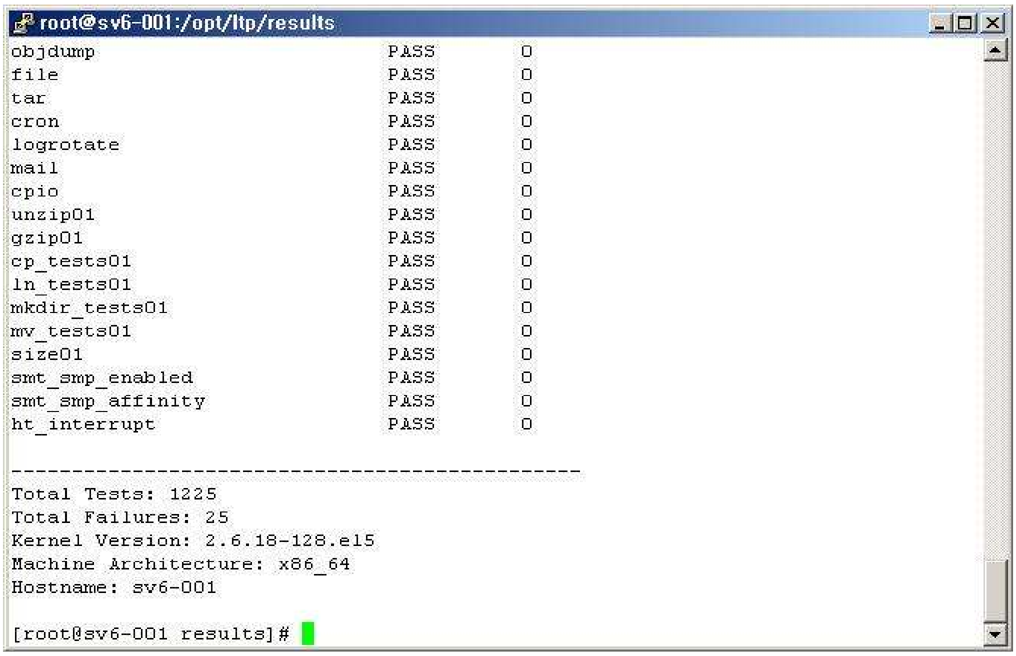
- 테스트 결과

▪ LTP 테스트 도구 버전 : ltp-full-20090930
▪ 테스트 시험 결과

[표 4-14. LTP 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	1225	25

※ 실패 목록은 CentOS에서 발생한 문제가 아닌 기능상 구현되지 않은 경우 이거나 LTP 툴의 자체 버그일 가능성이 높은 것으로 판단함([표 4-15] 참고)



[그림 4-14. LTP 시험 결과 화면]

▪ 테스트 시험 실패 목록

[표 4-15. LTP 시험 실패 항목]

항 목	설 명
eventfd01	eventfd test는 임시 파일을 임시 디렉터리(tmp)가 아니라 현재 디렉터리에 생성하는데, 현재 디렉터리는 nonwriteable로 설정되어 오류 발생 가능
ppoll01	ltp_signal.h에서 SIGSETSIZE 매크로를 사용하고, poll.h의 헤더 위치를 바꾸어 테스트하면 정상이지만 다른 테스트와의 호환을 위해서 수정하지 않고 테스트 수행 시 오류 발생
quotactl01	__NR_set_tid_address 시스템 호출에 하드코딩해서 258번을 대치해야 정상 수행되지만 수정되지 않은 커널은 테스트 수행 시 오류 발생
rt_sigaction01	Functionality test 방식으로 SIGRTMIN(34)에서 SIGRTMAX(64)까지 사용한다. 만약 rt_sigaction으로 1번 시그널부터 64번 시그널을 테스트하고자 한다면, 별도의 패치를 기반으로 커널 2.6.23 이후의 버전으로 테스트를 한다.
rt_sigprocmask01	rt_sigprocmask 시스템 호출에 대한 test로 real time signal test인 경우 별도의 패치를 통해 test를 수행하지만 신뢰성 약함
rt_sigsuspend01	rt_sigsuspend 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 현재 segmentation fault가 발생해서 test 항목의 신뢰성 약함
set_thread_area01	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)
set_thread_area02	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)
stime01	time() 함수가 리턴한 data 값보다 stime() (x86-64에서 settimeofday())이 리턴한 값이 더 적을 경우 FAIL
sysconf01	err# 대신에 “actual" return code를 리턴하면서 sysconf의 fail 오류가 0이 아닐 경우 오류 발생

utimes01	EACCESS times가 NULL이고 호출자의 EUID가 파일에 접근하는 호출자와 일치하지 않은 경우 오류 발생, 패치를 적용할 경우에 오류 해결 (최신버전)
su01	passwd -e 명령이 지원되지 않은 경우 FAIL 발생, 시스템에 영향 없음
hugemmap03	hugetlb 기반으로 normal page가 high memory region에 매핑할 수 없는지를 테스트하며 비정상적인 경우 FAIL
hugemmap04	적어도 하나의 테스트용 hugepage가 필요하고 hugepage의 개수에 대한 정보가 필요한데, 유효한 hugepage가 아닐 경우 FAIL
hugeshmat01	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패 시 FAIL
hugeshmat02	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패 시 FAIL
hugeshmat03	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패 시 FAIL
hugeshmctl01	임의의 usleep time은 부정확한 실행 순서를 야기하며 FAIL
hugeshmctl02	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패 시 FAIL
hugeshmctl03	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패 시 FAIL
hugeshmdt01	hugepage가 가용할 경우에만 hugepages를 hugeshmdt에 할당, 실패 시 FAIL
hugeshmget01	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget02	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget03	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget05	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL

라. 안정성 테스트

□ LTP Stress Tool 사용

LTP 테스트 도구를 사용한 스트레스 테스트를 수행한다.

- NFS 스트레스 테스트
- 메모리 관리 스트레스 테스트
- 파일시스템 스트레스 테스트
- Math (floating point) 테스트
- pthread 스트레스 테스트
- 디스크 I/O 테스트
- IPC (pipeio, semaphore) 테스트
- 시스템 호출 함수 확인 테스트
- 네트워킹 스트레스 테스트

- 실행 방법

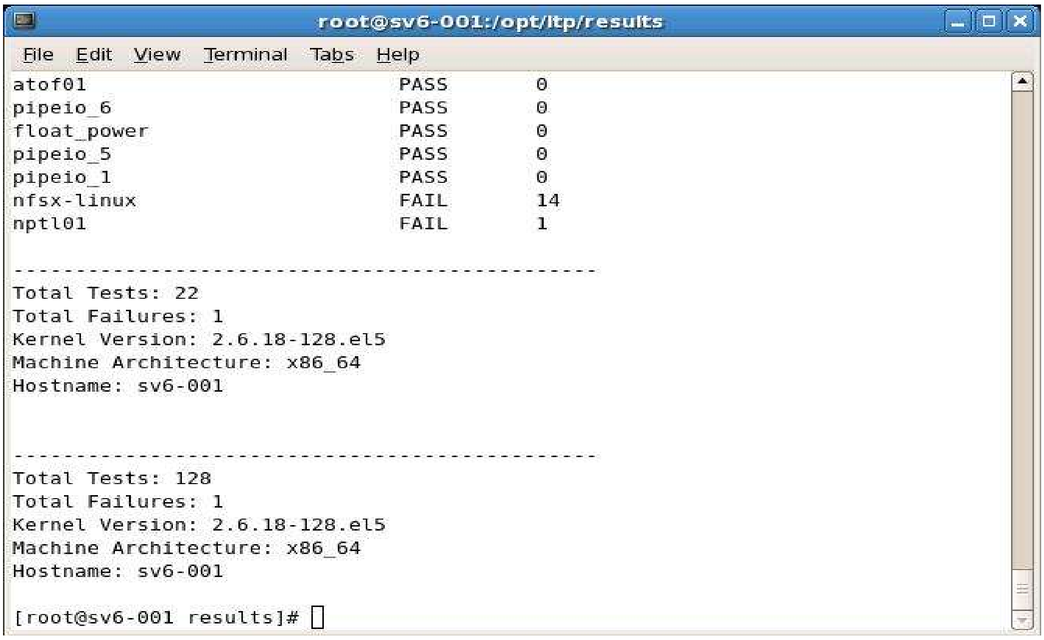
```
$ cd /opt/ltp/testscripts
$  ./ltpstress.sh -d /tmp/sardata -l /tmp/ltplog.1122 -m 1024 -t 24 -S
```

- 테스트 결과
  - LTP 테스트 도구 버전 : ltp-full-20090930
  - 테스트 시험 결과

[표 4-16. LTP Stress 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	22	1
	128	1

※ 실패 목록은 CentOS에서 발생한 문제가 아닌 기능상 구현되지 않아 발생한 오류일 가능성이 높은 것으로 판단함([표 4-17] 참고)



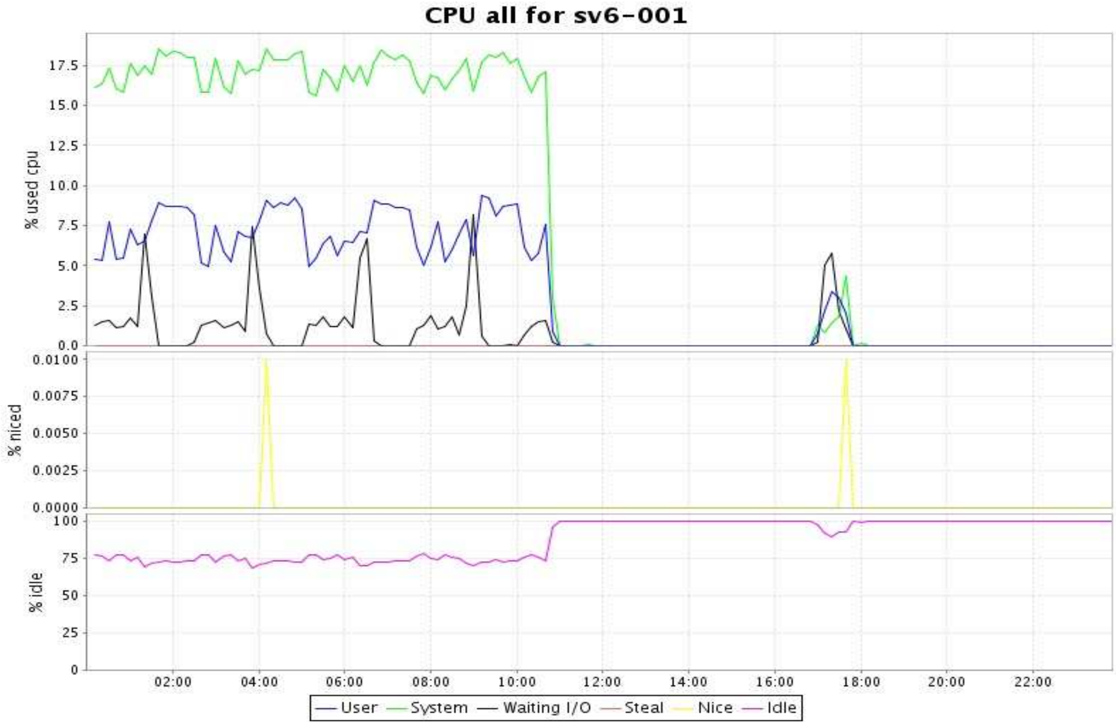
[그림 4-15. LTP Stress 시험 결과 화면]

▪ 테스트 시험 실패 목록

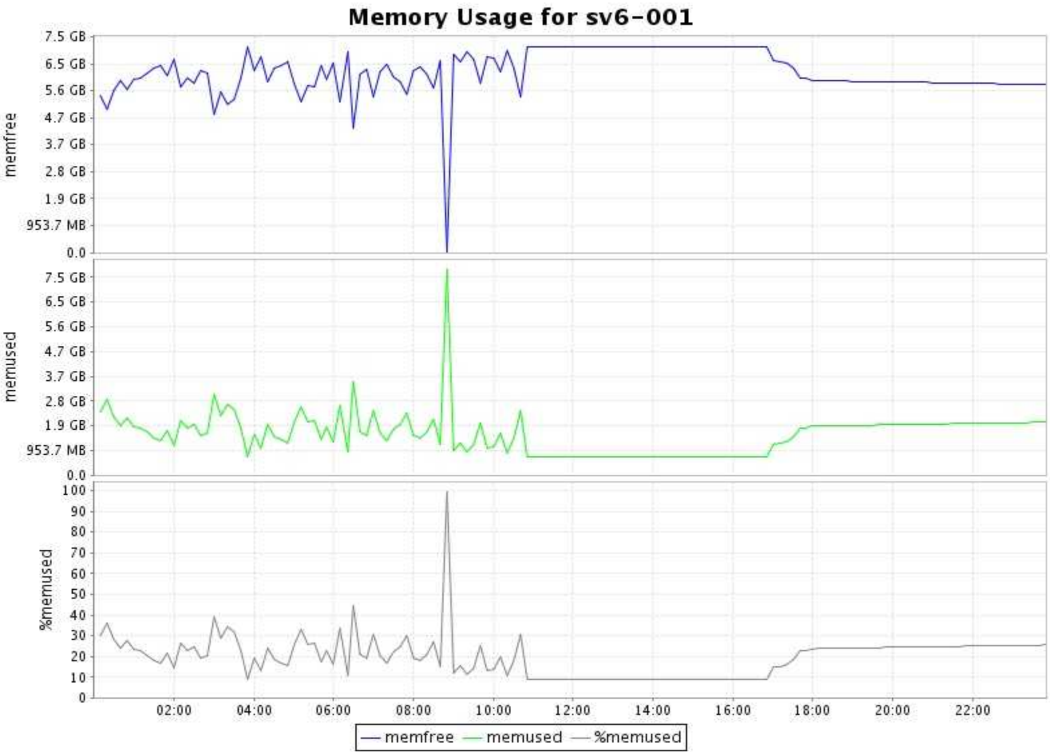
[표 4-17. LTP Stress 실패 항목]

항　목	설　명
nfsx-linux	nfsx-linux를 찾지 못해서 발생하는 오류로 nfs v3과 v4에서 버전 미스매치로 인해 발생하는 문제로 시스템에 영향 없음
nptl01	nptl의 pthread_cond_timedwait() 시스템 호출에서 발견되는 버그를 탐지하는 test이며 현재 버그 발생이며 FAIL임을 표시

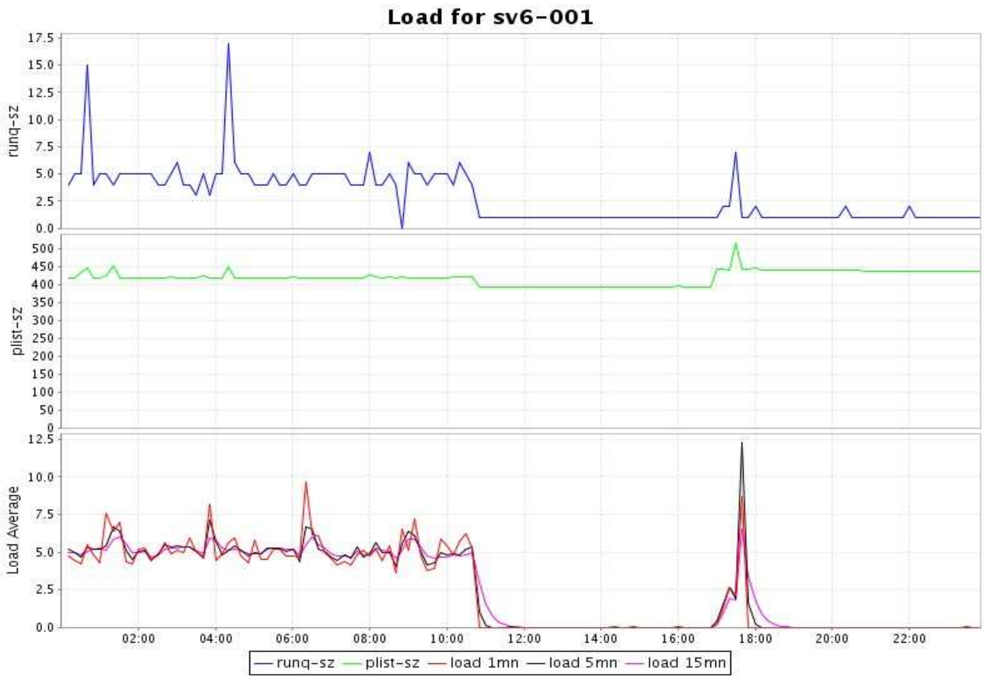
▪ 시스템 사용률



[그림 4-16. CPU 사용률 확인]



[그림 4-17. Memory 사용률 확인]



[그림 4-18. 평균 사용률 확인]

[표 4-18. 시스템 사용량 요약]

구분	최대 사용량	평균 사용량
CPU	100%	17.5%
Memory	8GB	2GB
Load Average	16	5

※ 참고 사항

- [그림4-16~18] 그래프는 24시간을 기준으로 작성되었다.
- 스트레스 테스트 그래프는 00시부터 ~ 11시까지 해당 된다.
- 수치가 낮을수록 시스템 부하가 적은 것임

5. 결론

□ 설치

최신의 하드웨어 테스트 환경에서 드라이버가 기본으로 제공되므로 누구나 쉽게 설치가 가능하며, Red Hat Enterprise 제품의 클론 제품이므로 안정적으로 사용이 가능하나, 국내에서 배포하는 제품이 아니므로 기술지원을 받기가 어려울 수 있다.

□ 기능

Red Hat Enterprise에서 지원하는 대부분의 기능을 지원하며, 다른 Unix나 Windows에서 지원하는 기능 또한 부족함이 없으므로 정부기관 및 기업에서 도입이 가능한 제품이다.

□ 신뢰성

LTP 툴에서 보여주듯 1000가지가 넘는 기능 테스트 중 대부분이 문제없이 테스트되었으며, 일부 테스트의 경우 기능상 구현되지 않았거나 지원하지 않는 부분의 오류이므로 운영체제에 문제가 있다고 판단하기는 어려우며 서비스용으로 사용하더라도 문제가 없을 것으로 보인다.

□ 안정성

스트레스 테스트 결과에서 보이듯 시스템이 안정적으로 운영되기 위한 기본 요건을 갖춘 제품이다.

CentOS는 Global에서 많이 알려진 Red Hat Enterprise Linux의 클론 제품으로서, 테스트 결과, 사용 중 문제점을 발견할 수 없었으며, 국내 IDC나 소규모 서비스 업체에서 많이 사용되고 있는 제품이므로 안정적이고 신뢰할 수 있는 제품이라 할 수 있다. 그러나, 자체적으로 기술력을 확보하거나, 기술지원 업체를 확보해야 하는 문제점이 있다.

* 참고 자료

http://www.centos.org/

http://ko.wikipedia.org/wiki/CentOS

* 별첨

별첨1. LTP 활용 가이드

Ⅱ CUBRID

목 차

1. 개요

본 문서는 공개SW 역량프라자 운영 사업의 공개SW 테스트 환경 구축을 통하여 테스트한 공개SW에 대한 기능, 성능, 호환성 등을 검증하고, 향후 공공 기관 및 기업에서 도입 시 참고할 수 있는 자료로 활용하기 위하여 작성되었다.

가. 목적

□ 공개SW를 테스트하여 다음과 같은 목적을 달성하고자 한다

- 공개SW의 기능, 성능, 안정성, 호환성 등 검증
- 공개SW 기술의 도입 가이드라인 제공
- 효과적인 공개SW 도입 유도
- 안정적인 공개SW 시스템 도입

나. 선정 기준

□ 공개SW 테스트 대상은 다음 5가지 사항을 고려하여 선정되었다.

- 1번 : 역량프라자 내 대표 공개SW DBMS 발굴 제품
- 2번 : 국내에서 개발된 제품
- 3번 : 국내 기업에서 많이 도입되는 제품
- 4번 : Global에서 많이 알려진 제품
- 5번 : 기술지원(업데이트) 가능 제품

[표 1-1. 선정 기준]

DataBase	1번	2번	3번	4번	5번	결과	비고
CUBRID	●	●	◑	◔	●	◕	선정
MySQL	●	○	●	●	●	◕	선정
Postgre SQL	○	○	◑	◕	●	◑	보류

※ Postgre SQL은 국내 도입된 사례가 극히 적은 제품으로 파악됨

2. 테스트 환경 구성

가. 테스트 일정

[표 2-1. 일정표]

구분	월	10				11				12			
	주	1	2	3	4	1	2	3	4	1	2	3	4
테스트 대상 발굴 및 선정													
테스트 환경 구성													
테스트 항목 선정													
테스트 시나리오 작성													
테스트 수행													

나. 테스트 대상 소개

□ CUBRID

CUBRID는 국내 유일의 오픈소스 DBMS로서, 인터넷 서비스 최적의 DBMS를 지향하고 있다.

미션 크리티컬(Mission-critical)한 인터넷 응용에서 요구하는 고성능, 안정성, 확장성 및 가용성을 보장함은 물론, GUI 기반의 클라이언트 도구를 제공하여 개발자가 편리하게 DBMS를 사용하고 관리할 수 있도록 한다.

또한, 대규모 인터넷 서비스 영역에서 반드시 필요한 복제 (Replication), 분할 (Partition), 그리고 자바 기반의 저장 프로시저를 지원한다.

CUBRID는 국내 최대 규모의 인터넷 포털 업체인 NHN에서 제공하는 네이버 서비스에 적용되어 안정적으로 운영되고 있으며, 교육인적자원부를 비롯한 여러 공공기관과 해외 정부기관에서 사용 중에 있다.

□ CUBRID 특징

- 대용량 트랜잭션

CUBRID는 대용량 DB에서도 만족할만한 트랜잭션 성능을 보여주는 관계형 데이터베이스로서, DB개수나 크기, 테이블 개수에 관계없이 원활한 트랜잭션을 보장함

▪ DB 개수/크기, 테이블 개수/크기 무제한
▪ 멀티쓰레드 서버 – 멀티 CPU 사용 최적화
▪ 복제 (replication) – 서버 추가
▪ 분할 (partition) – 데이터 확장 대처
▪ 완벽한 트랜잭션 ACID 지원
▪ 복제 간 트랜잭션 일치성 보장
▪ 다중 단위 잠금 (multiple granularity locking)

▪ 분산 트랜잭션 (2PC) 지원

- 웹 서비스에 특화된 성능 및 기능 제공

▪ 인터넷 서비스의 게시판 유형의 웹 서비스에 대해 개발자 편의성을 제공하기 위한 Click Counter(클릭 카운터)기능이나, 인터넷 기반 서비스의 특성상 특정 질의가 반복적으로 사용되는 상황을 보다 쉽게 해결하기 위해 Query Plan Cache(쿼리 플랜 캐시)와 같은 타 DBMS와 차별되는 기능을 제공함

▪ 24시간, 365일 웹 서비스 동안 연속성을 유지하고 운영의 편의성을 극대화하기 위해 HA(high Availability) 기능과 서버 분산을 통한 성능 향상 및 장애 발생 시 긴급 대처를 위한 Replication 기능도 제공함

- 고성능의 서버/클라이언트 구조

멀티 쓰레드, 멀티 서버구조로, Cost Based Optimizer (CBO)를 포함하고 있으며, 데이터를 복제 하고 분할하여 DB간의 부하를 최소화 함

▪ 멀티쓰레드, 멀티 서버

▪ Cost-Based Optimizer (CBO)

▪ 질의 플랜 캐시

▪ 분할(partition) – 질의 분할 최적화

▪ 복제(replication) – 질의 분산

▪ CAS 미들웨어 포함

▪ 쓰레드 풀 관리

▪ 자동 부하 최적화 기능

▪ 객체 메모리 캐시 – pointer chasing

- 장애 처리 시, 실시간 백업/복구

▪ CUBRID는 3단계 백업 레벨을 지원함으로써, 장애 발생시점 및 특정시점에서 복구 지원이 가능함

▪ 온, 오프라인에서의 백업 지원은 물론, 다양한 백업 시스템과의 연동을 통해 최단시간 내에 재해 복구 및 백업 기능을 제공함

다. 테스트 환경

□ 테스트 서버

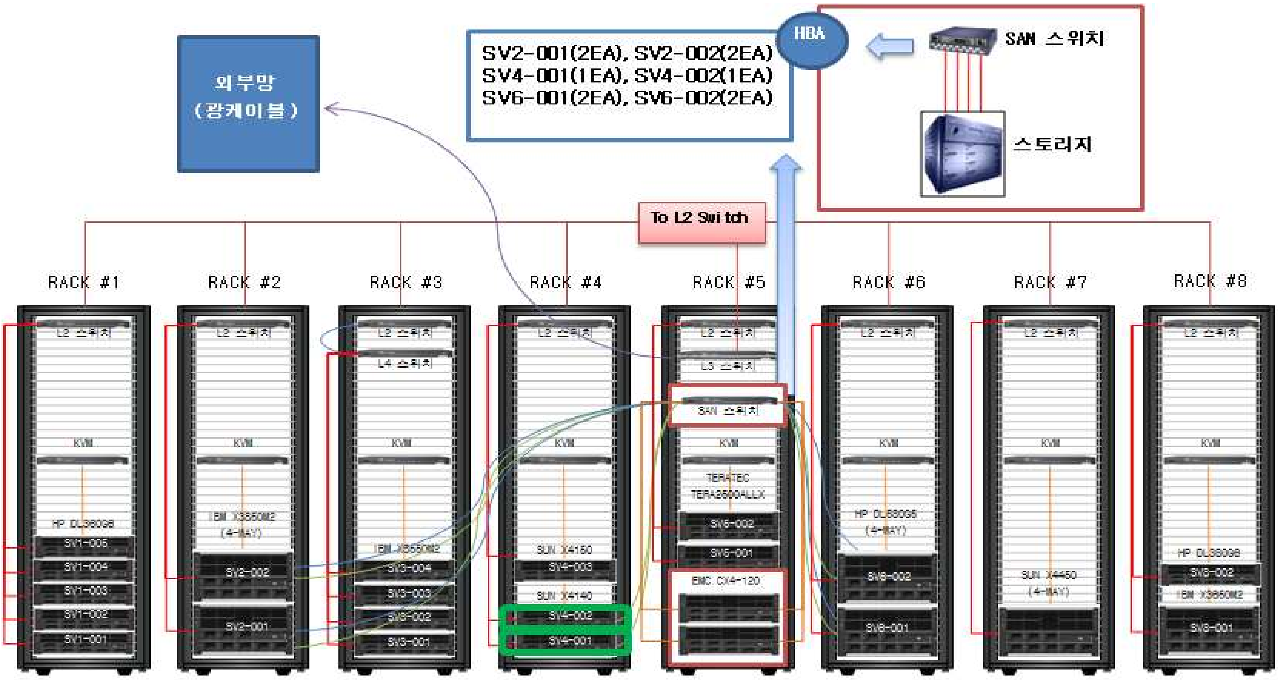
[표 2-2. 테스트 서버 규격]

구분	제품명	CPU	MEM	Disk	NIC	수량
SUN	X4140	Quad-Core 2.5Ghz 2-Way	8GB	146GB*5	Gigabit 2Port	2

□ 테스트 조건

- 32bit 환경에서 64bit환경으로 서버 구성이 전환됨에 따라 64bit 환경에서의 테스트를 수행
- Linux 환경에서 테스트

□ 테스트 실장도



☞ 박스 안의 서버가 DB 테스트용 서버로 사용 됨.

[그림 2-1. 테스트 서버 실장도]

□ 서버 위치

[표 2-3. 서버 위치]

제조사	모델명	Rack 위치	비 고
SUN	X4140	#4	

3. 테스트 시나리오

가. 호환성 테스트

□ 설치 지원 플랫폼(HW&OS)
- 운영체제별 지원 여부
- Architecture별 지원 여부

□ Installation packages
- Platform 별 package 제공 여부
- Binary Package 제공 여부
- Source Package 제공 여부

□ Installation Deployment
- 기본 설치 Testing
- Installation Customizable 여부
- 기본 작업 테스트
  ▪ Installation 방법
  ▪ Installation procedure listup(기본 스키마)

나. 기능 및 성능 테스트

□ CUBRID QA Tool 사용

□ JMeter Tool 사용

다. 가용성 테스트

□ Active/Active High Availability
- Active/Active Solution 제공 유무
- Active/Standby Solution 제공 유무

라. 관리 도구 테스트

□ GUI Tools

□ Monitoring Tools

4. 테스트

가. 호환성 테스트

□ CUBRID 홈페이지를 통한 설치 지원 플랫폼(HW&OS) 확인

[표 4-1. 플랫폼별 지원 내역]

<table>
<tr><th>운영체제</th><th>Architecture</th><th>호환 운영체제(OS)</th><th>CUBRID 2008 2.0</th></tr>
<tr><td rowspan="10">Linux</td><td rowspan="7">x86</td><td>CentOS 4이상</td><td rowspan="7">○</td></tr>
<tr><td>Fedora 4 이상</td></tr>
<tr><td>Ubuntu 6.10 이상</td></tr>
<tr><td>Gentoo 2007 이상</td></tr>
<tr><td>openSUSE 11 이상</td></tr>
<tr><td>Asianux 2 이상</td></tr>
<tr><td>Debian 4 이상</td></tr>
<tr><td rowspan="3">x86_64</td><td>CentOS 4이상</td><td rowspan="3">○</td></tr>
<tr><td>Fedora 11</td></tr>
<tr><td>Ubuntu 9.04</td></tr>
<tr><td rowspan="5">Windows</td><td rowspan="4">x86(32bit)</td><td>Windows XP Home 32bit</td><td rowspan="4">○</td></tr>
<tr><td>Windows XP Pro</td></tr>
<tr><td>Windows 2003</td></tr>
<tr><td>Windows Vista 32bit</td></tr>
<tr><td>x86_64(64bit)</td><td>Windows Vista 64bit</td><td>○</td></tr>
</table>

□ 호환성 테스트 대상 및 지원 확인

[표 4-2. 호환성 테스트 결과]

운영체제	Architecture	MySQL 버전	지원 여부
CentOS 5.3	x86_64	5.4	지원
Fedora 11	x86_64	5.4	지원
GINUX 2.2	x86_64	5.4	지원
SULinux 2.0	x86_64	5.4	지원

☐ Installation packages

- Platform 별 Binary Package 제공 여부

▪ CentOS

[표 4-3. CentOS 패키지]

PLATFORM	Bynary Package	다운로드 위치
CentOS	x86, x86_64	http://www.cubrid.com/zbxe/download_2008

▪ Microsoft

[표 4-4. MS Windows 패키지]

PLATFORM	Bynary Package	다운로드 위치
Windows	32bit, 64bit 제공	http://www.cubrid.com/zbxe/download_2008

- Source Package 제공 여부

[표 4-5. Source 패키지]

PLATFORM	Source Package	다운로드 위치
Linux x86/x86_64	제공	http://cubrid.svn.sourceforge.net/viewvc/cubrid/

☐ Installation Deployment

- 기본 설치 Testing

[표 4-6. CUBRID 설치 버전]

OS(X86_64)	Install Package	CUBRID 버전
CentOS 5.3 Fedora11 Ginux 2.2 SULinux 2.0	CUBRID-8.2.0.2005-linux.x86_64.tar.gz	2008 R2.0

▪ Installation 명령

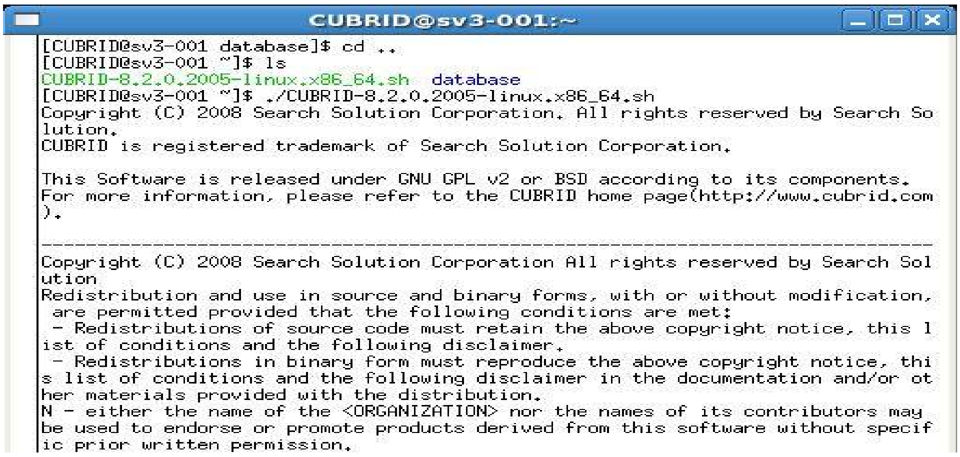
[표 4-7. CUBRID 설치 방법]

구분	명령어	결과
CUBRID 설치	tar xvfz CUBRID-8.2.0.2005-linux.x86_64.tar.gz chmod a+x CUBRID-8.2.0.2005-linux.x86_64.sh ./CUBRID-8.2.0.2005-linux.x86_64.sh	정상 설치 (그림4-1, 4-2 파일 참고)

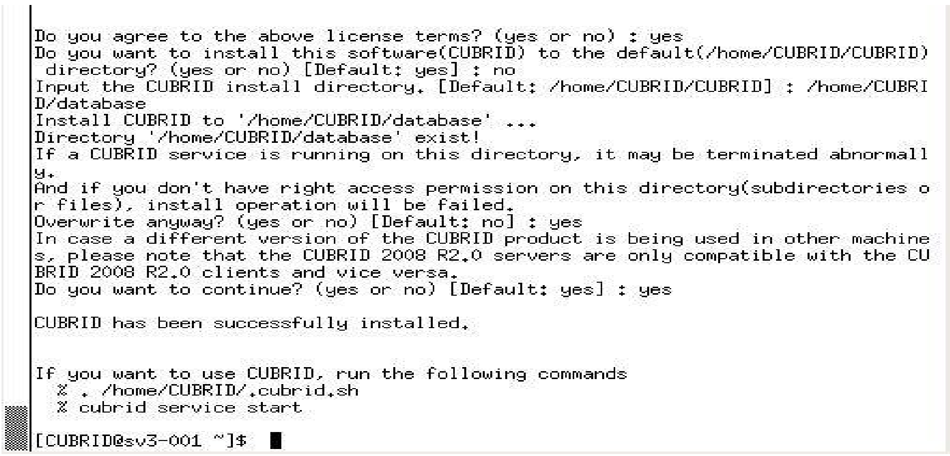
※ 참고 사항

- CUBRID는 Binary 형태로 제공하며 압축파일을 해제하는 형태로 설치하도록 구성됨
- cubrid 일반 사용자를 시스템에 추가하여 설치하는 것을 권장함

▪ CUBRID 설치 화면



[그림 4-1. 설치 실행 화면]



[그림 4-2. 설치 완료 화면]

- Installation Customizable 여부

Binary 압축 버전을 다운로드 받아서 원하는 위치에 압축 풀고 사용

```
tar xvfz CUBRID-8.2.0.2005-linux.x86_64.tar.gz
sh CUBRID-8.2.0.2005-linux.x86_64.sh
```

- 기본 작업 테스트

▪ Installation 방법

```
CUBRID 설치 파일을 다음과 같이 설치 진행
=====================================================
$ sh CUBRID-8.2.0.2005-linux.x86_64.sh
   중략
Do you agree to the above license terms? (yes or no) : yes
Do you want to install this software(CUBRID) to the default(/home/CUBRID/CUBRID)
 directory? (yes or no) [Default: yes] : yes

If you want to use CUBRID, run the following commands
  % . /home/CUBRID/.cubrid.sh
  % cubrid service start
=====================================================
```

※ 참고 사항

압축을 해제하고 실행 후 라이선스 동의 및 설치 위치만 지정하면 정상 설치됨

▪ Installation procedure listup

○ 설치 디렉터리에 databases 하위 디렉터리 생성

예) /home/CUBRID에 설치할 경우 /home/CUBRID/databases라는 디렉터리를 생성함

○ 환경변수 설정

사용자 shell에 맞춰 아래의 내용이 담긴 shell파일(.cubrid.sh, .cubrid.csh 등)을 등록함

○ DB 생성

ISV형태 설치는 기본 제공되는 demodb가 생성되지 않으므로 테스트를 위해 별도 DB가 필요함

(1) ~/.cubrid.sh 파일의 내용

```
CUBRID=/home/CUBRID
CUBRID_DATABASES=/home/CUBRID/databases
CUBRID_LANG=en_US
ld_lib_path='printenv LD_LIBRARY_PATH'
if [ "$ld_lib_path" = "" ]
then
LD_LIBRARY_PATH=$CUBRID/lib
else
LD_LIBRARY_PATH=$CUBRID/lib:$LD_LIBRARY_PATH
fi
SHLIB_PATH=$LD_LIBRARY_PATH
LIBPATH=$LD_LIBRARY_PATH
PATH=$CUBRID/bin:$CUBRID/cubridmanager:$PATH
export CUBRID
export CUBRID_DATABASES
export CUBRID_LANG
export LD_LIBRARY_PATH
export SHLIB_PATH
export LIBPATH
export PATH.
```

(2) 사용자 shell 파일에 아래와 같이 등록함

```
#.bash_profile(bash shell일 경우) 파일의 내용
    …
. /home/CUBRID/ .cubrid.sh
```

(3) CUBRID환경변수를 아래와 같이 적용하고 cubrid service를 구동함

```
$. /home/CUBRID/ .cubrid.sh
$cubrid service start
```

(1) DB 생성

DB를 생성할 곳을 선택하고 아래와 같이 명령어를 수행하여 생성함

예) /home/CUBRID/databases에 TestDB 폴더를 생성한다.(원하는 경로로 변경 가능)

cubrid createtdb TestDB ☞ DB생성 시 사용 옵션에 대해서는 매뉴얼을 참조함

cubrid addvoldb -p data TestDB 10000 ☞ addvoldb의 옵션은 매뉴얼을 참조함

cubrid addvoldb -p index TestDB 5000

cubrid addvoldb -p temp TestDB 5000.

※ 각각의 volume은 필요에 따라 추가해 줄 수 있으며 32bit는 2G, 64bit는 파일크기에 제한이 없음

(2) 생성한 DB 확인 방법

cubrid server start TestDB ☞ DB구동 방법이며 실행 위치는 어디든 상관없음

cubrid server stop TestDB ☞ 구동된 DB를 종료하는 방법이며 실행 위치는 어디든 상관없음

cubrid server status 생성된 모든 DB의 구동 상태를 보여줌

cubrid spacedb -C TestDB ☞ -C는 DB구동 시 -S는 DB가 구동되어 있지 않을 경우 사용함

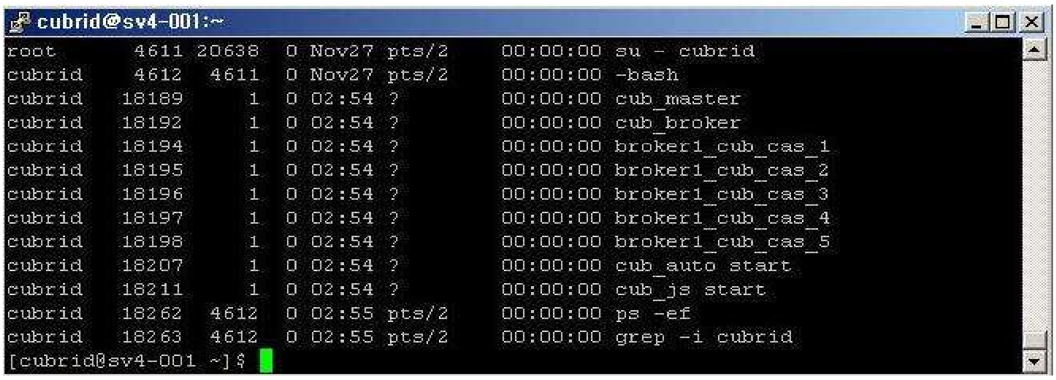
▪ CUBRID 실행

$ cubrid service start 명령으로 실행함

[그림 4-3. CUBRID 실행]

▪ CUBRID 실행 프로세스 확인

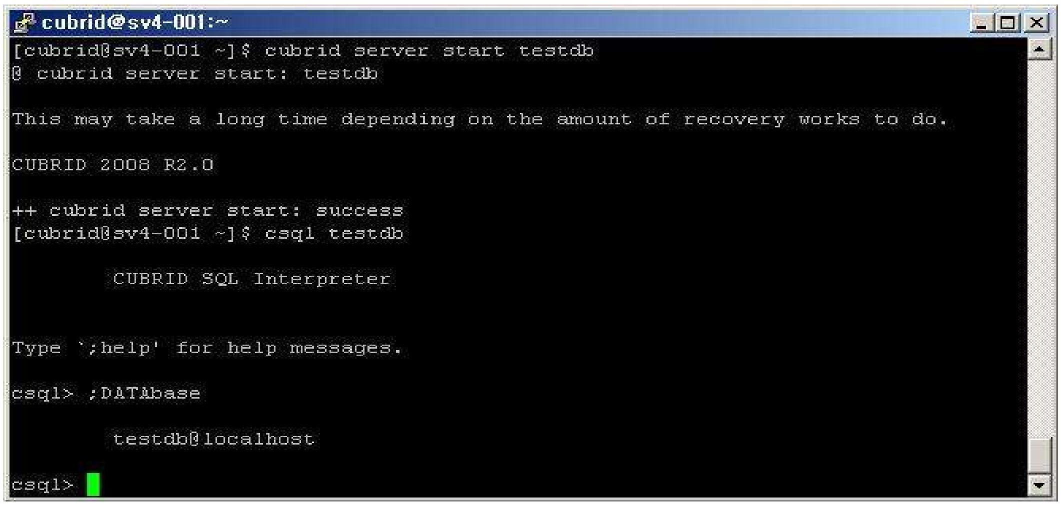
$ ps -ef | grep -i cubrid 명령으로 확인함



[그림 4-4. CUBRID 프로세스 확인]

▪ CUBRID testdb 구동 및 접속 화면

$ cubrid server start testdb 명령으로 접속함



[그림 4-5. CUBRID TestDB 구동]

※ CUBRID는 cubrid 사용자에서 실행하도록 설계가 되어 있으며, Binary의 배포도 압축된 상태로 배포되어 사용자가 손쉽게 설치 및 운영할 수 있는 구조임

나. 기능 및 성능 테스트

□ 기능 테스트

- CUBRID QA Tool

CUBRID QA 시험 도구는 CUBRID DBMS를 위한 시험 시나리오를 관리하고 시험을 수행하는 도구이다. Java Eclipse RCP 기반으로 개발되고 있으며, 오픈되어 있는 시험 도구에는 약 7000개의 시나리오 파일이 포함되어 있다. 이 시나리오 파일에는 CUBRID 의 다양한 기능 시험을 위한 SQL 들이 저장되고 시험된다.

- 테스트 시나리오

CUBRID DBMS 의 기능 측면에서의 적합성 검증을 수행하기 위해 CUBRID QA 도구는 크게 세 가지 형태의 시험을 수행함

▪ SQL 구문을 이용한 시험
▪ shell 환경에서 command를 이용한 시험
▪ 복제 기능 시험

이번 테스트에서는 SQL 구문을 이용한 기능 시험만 진행 함

SQL 구문을 수행하여 시험도구에 저장된 대로 기대하는 결과가 반환되는지 여부를 검증하고 SQL을 이용한 시험 시나리오는 아래 그림과 같이 크게 sql, medium, site 로 구분됨

[그림 4-6. CUBRID QA Tool 시험 시나리오]

sql, medium, site 시나리오 중 sql 시나리오를 실행함

- 시험 방식 및 sql 시나리오 설명

▪ 시험 방식

시험을 수행하는 방식은 세 가지 디렉터리 모두 동일한 방식으로 수행됨

시험 시나리오 위에 autocommit [on|off] 라는 구문을 이용하여 한 시나리오 파일에 기록된 sql 구문을 수행할 때 auto commit on/off 여부를 지정할 수 있음

지정 구문이 없으면 기본적으로 auto commit off 로 수행됨

sql 시험은 sql 구문이 기록된 파일 단위로 수행되며, 시험 도구는 파일의 마지막에 무조건 commit 을 호출함

▪ sql 시나리오 설명

DDL, DML, DCL 등 CUBRID 가 지원하는 모든 SQL 구문의 시험을 수행함 SQL 노드는 기능 별로 노드의 이름을 구분하고 각 노드는 다시 서브 노드로 구성됨

최상위 노드의 구조는 다음과 같음(경로는 $QA_REPOSITORY/scenario를 기준으로 함)

[표 4-8. CUBRID QA 시나리오 설명]

시나리오 이름	경 로	설 명
01_object	sql/_01_object	CUBRID가 지원하는 객체들에 대한 기능 시험을 점검하는 부분으로 3200 여개의 시나리오를 가지는 가장 많은 부분이다.
02_user_authorization	sql/_02_user_authorization	사용자 관리 및 권한 관리 기능에 대한 기능 시험을 수행하는 부분이다.
03_object_oriented	sql/_03_object_oriented	CUBRID는 관계형 데이터베이스에 객체 지향 개념을 확장한 DBMS 시스템으로 이 는 확장된 객체 지향 개념에 대한 시험을 수행한다.
04_operator_function	sql/_04_operator_function	CUBRID 가 지원하는 기본 함수 및 연산자에 대한 기능 시험을 담당하는 부분이다.
06_manipulation	sql/_06_manipulation	DML 구문의 가장 기본 SQL 구문인 insert, update, delete, select 구문을 시험하는 부분이다. 기본 구문부터 서브 질의 및 각종 조인 질의를 시험하는 부분이 포함된다.
07_misc	sql/_07_misc	주로 DCL(Data Control Language)에 포함된 기능을 시험하는 부분으로 통계 정보 갱신 및 기타 기능에 대한 시험을 수행하는 부분이다.
08_javasp	sql/_08_javasp	자바 stored procedure의 모든 기능 시험을 수행하는 부분이다
09_64bit	Sql/_09_64bit	Bigint, datetime 두 가지 타입에 대한 기능 시험을 수행하는 부분이다.

▪ sql 시나리오 실행

X-windows 에서 root 로 로그인 후 다음과 같이 실행함

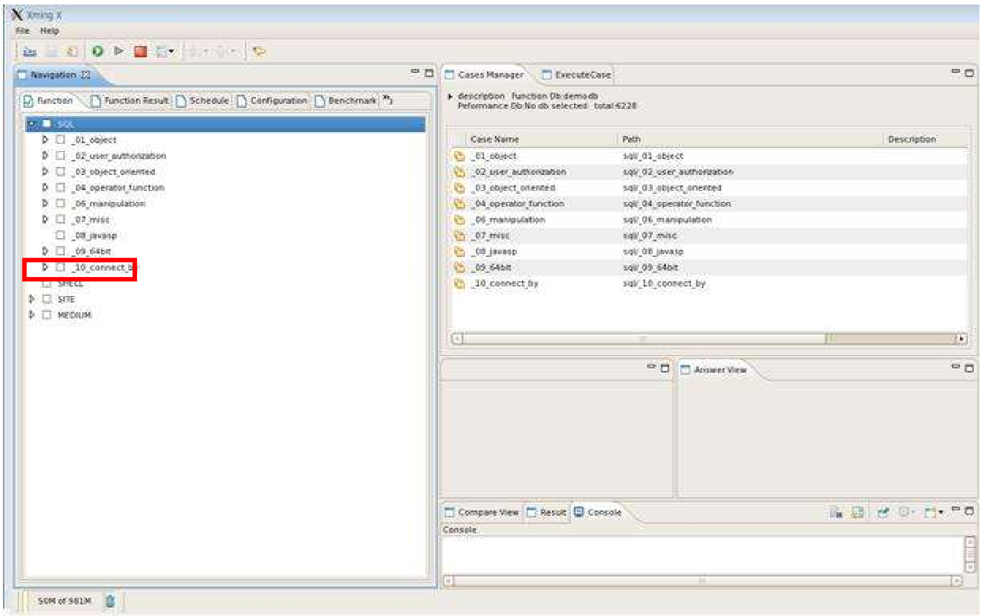
```
# source    /home/cubrid/.bash_profile
# cd   /home/cubrid/trunk/qa_repository/qatool_bin/qamanager/
cubridqa_centos64_kde
# ./cubridqa
```

X-windows 화면에서 QA 창 실행 후 SQL을 클릭함

※ 주의 사항

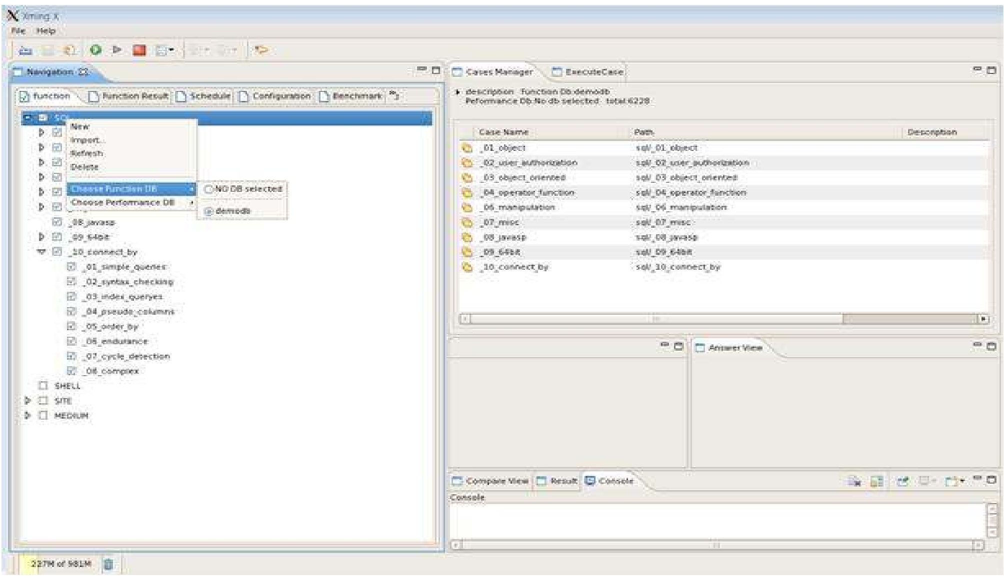
○Java sp를 실행하기 위해서는 DB에 먼저 java sp를 적재해야 함

○Connect by 의 경우 R2.1에 release되므로 선택하면 안 됨



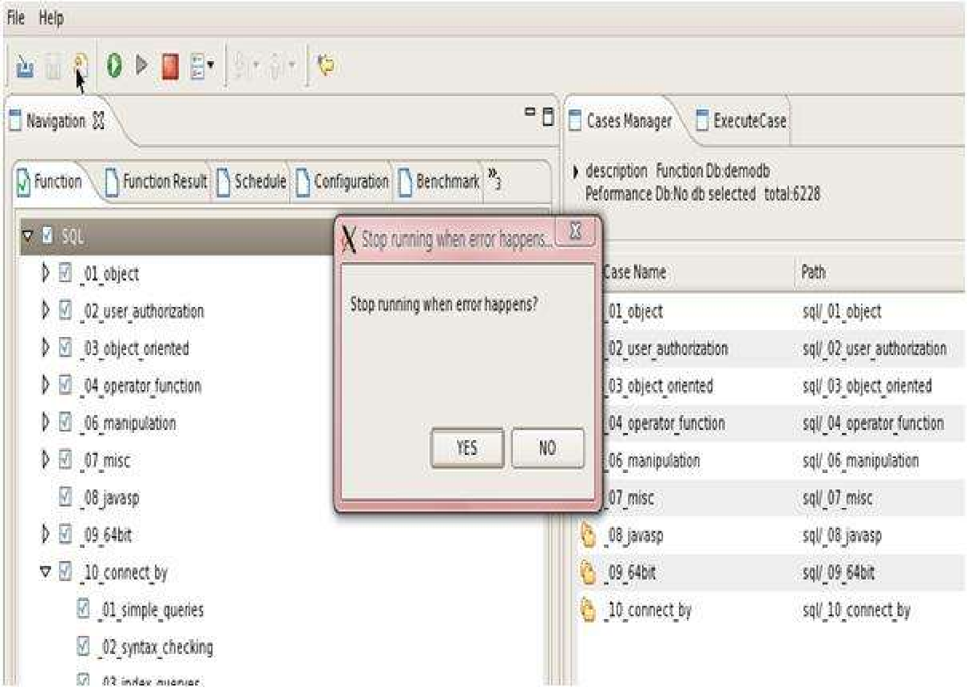
[그림 4-7. CUBRID QA Tool 실행 화면]

마우스 오른쪽 버튼을 클릭 후 테스트 DB를 선택 후 실행함



[그림 4-8. CUBRID QA Tool 테스트 DB 실행]

실행 버튼 클릭 후 새로운 dialog box가 나타난 후 No를 선택하면 QA 시나리오가 실행됨

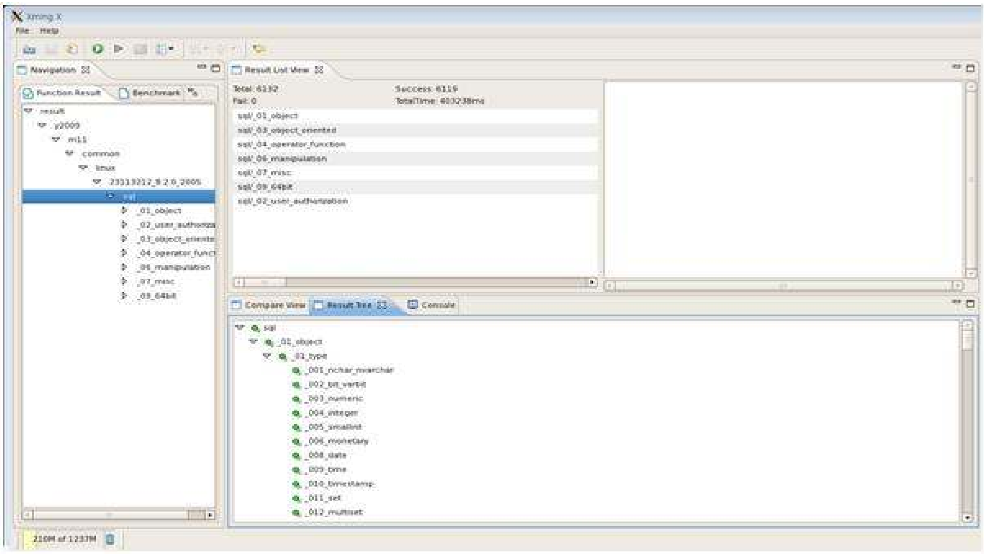


[그림 4-9. CUBRID QA Tool 실행 화면]

▪ 테스트 결과

[표 4-9. CUBRID QA Tool 테스트 결과]

테스트 합계	성공	실패	해당 없음
6132	6119	0	13
결과	결함 없음		



[그림 4-10. CUBRID QA Tool 완료 화면]

□ 성능 테스트

- JMeter Tool

Apache JMeter 는 테스트 기능과 퍼포먼스를 측정하는 기능을 갖는 100% 순수 자바 데스크탑 애플리케이션이다. 원래는 웹 애플리케이션의 테스트를 위해서 설계가 되었지만 현재는 다른 기능의 테스트를 위해 확장되고 있다.

- 사용 방법

```
$ /home/webtest/jakarta-jmeter-2.3.4/bin/jmeter.sh
CTRL+O Test Plan.jmx 을 열어 시나리오 로딩
CTRL+R 시나리오 실행
```

- 테이블 구조(Schema)

```
csql> ;sc sbtest

=== <Help: Schema of a Class> ===
 <Class Name>
     sbtest
 <Attributes>

     id                    INTEGER NOT NULL
     k                     INTEGER DEFAULT 0 NOT NULL
     c                     CHARACTER VARYING(120) DEFAULT '' NOT NULL
     pad                   CHARACTER VARYING(60) DEFAULT '' NOT NULL

 <Constraints>
     PRIMARY KEY pk_sbtest_id ON sbtest (id)
```

- 데이터 건수

  100만 건

- 테이블 용량

```
[cubrid@sv4-001 ~]$ cubrid spacedb --size-unit=M sdb
Space description for database 'sdb' with pagesize 8192.

Volid  Purpose   total_size   free_size  Vol Name
    0  GENERIC       390.6 M     178.6 M  /data/cubdata/sdb
Space description for temporary volumes for database 'sdb' with pagesize 8192.

Volid  Purpose   total_size   free_size  Vol Name
32766     TEMP       159.0 M     158.7 M  /data/cubdata/sdb_t32766
```

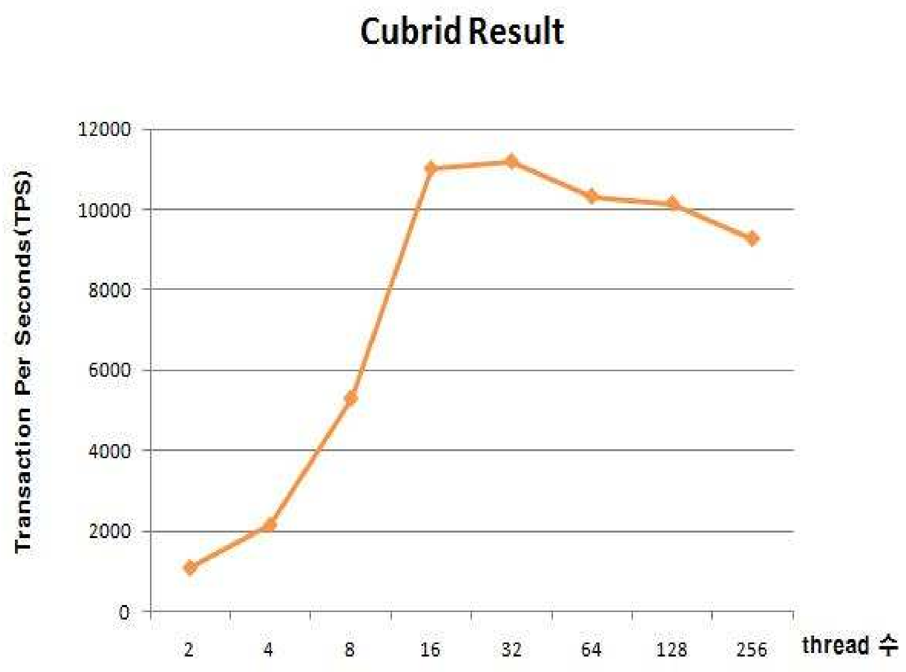
- 테스트 시나리오

▪ 특정 데이터 지정(10회) : SELECT c FROM sbtest WHERE id=N ▪ 특정 범위 지정(1회) : SELECT c FROM sbtest WHERE id BETWEEN N AND M ▪ 범위 지정 합계(1회) : SELECT SUM(c) FROM sbtest WHERE id BETWEEN N and M ▪ 범위 지정 후 Order by 수행(1회) : SELECT c FROM sbtest WHERE id between N and M ORDER BY c ▪ 범위 지정 후 DISTINCT 수행(1회) : SELECT DISTINCT c FROM sbtest WHERE id BETWEEN N and M ORDER BY c

- 테스트 결과

[표 4-10. 테스트 결과]

Thread	실행 횟수	평균응답(ms)	TPS	테스트 시간(Sec)
2	28,000	1	1092.64	25.626
4	56,000	1	2158.33	25.946
8	112,000	1	5299.52	21.134
16	224,000	1	10998.7	20.366
32	448,000	2	11182.7	40.062
64	896,000	6	10317	86.847
128	1,792,000	12	10137	176.778
256	3,584,000	27	9266.35	386.776



[그림 4-11. CUBRID 테스트 결과]

※ 위의 결과에서 보면 32 thread까지는 성능이 올라간 후 떨어지는 것을 볼 수 있으며, 이 부분은 시스템의 Spec과 튜닝 포인트에 따라 달라질 수 있음

□ 테스트 결과 요약

큐브리드 QA 툴 테스트 결과 6000개 이상의 기능이 정상적으로 수행되었으며, 이러한 결과는 큐브리드 DBMS가 상용 DBMS와 비교하여 기능적인 면에서 문제 되지 않는다는 점을 증명하고 있음

JMeter를 사용하여 sysbench 툴에서 제공하는 기능과 동일하게 테스트 한 결과 thread수를 늘려가며 테스트 할 경우 32 thread를 기준으로 최고 성능치에서 점점 떨어지는 결과가 나왔음

이러한 결과는 큐브리드 DBMS가 일정한 부하에서는 정상적인 성능을 발휘할 수 있음을 의미하며, 이는 Enterprise 환경에서도 충분히 사용될 수 있는 성능을 가진 DBMS임을 증명함

다. 가용성 테스트

□ High Availability

High Availability Cluster (= Failover Cluster)는 가용성을 향상시켜 서비스의 안정성을 확보하는데 중점을 둔 클러스터이다. 따라서 준비된 여러 node들을 통해, 구성 node들 중 일부에 장애가 발생을 하여도 중단 없는 서비스를 제공하며, 한곳의 장애로 인하여 서비스의 중단(single point of failure)이 되는 상황에 잘 대처할 수 있다.

또한 HA Cluster는 시스템이나 소프트웨어의 장애 발생 시 이를 검출하고, 일정 시간 내에 자동으로 이를 복구 (Failover) 하여 운영자의 개입 없이 중단 없는 서비스를 제공할 수 있게 한다.

- Cluster 기능 제공 유무

[표 4-11. Cluster 제공 확인]

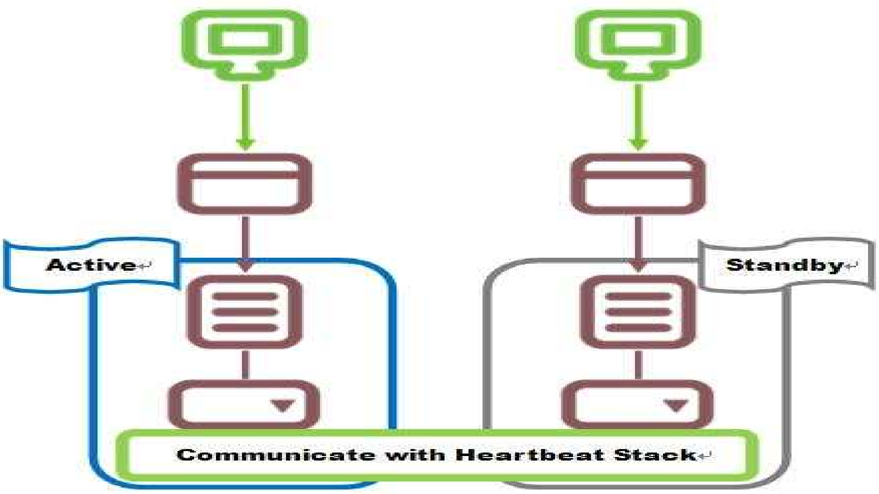
구분	Package	제공 여부
Active/Active	기능 지원 예정	△
Active/Passive	Heartbeat	○

- 운영체제별 지원

[표 4-12. Cluster 운영체제 지원 확인]

구분	운영체제 종류	지원 여부
Cubrid HA	Linux 배포판	○

- Cluster 구성도



[그림 4-12. Cluster 구성도]

- Cluster 구성
  - 환경 구성

```
CUBRID HA 기능 설정을 위하여 다음과 같이 각 호스트의 설정 파일을 수정한다.

 ○ $CUBRID_DATABASES/databases.txt 수정

shell> cat $CUBRID_DATABASES/databases.txt
#db-name vol-path db-host log-path
demodb /home/cubrid/ databases/demodb host1:host2 \
/home/cubrid/databases/demodb

 ○ $CUBRID/conf/cubrid.conf 수정

shell> cat $CUBRID/conf/cubrid.conf
…
ha_mode=on
```

  - heartbeat 설치

```
각 호스트에 heartbeat 패키지를 설치한다.
shell> $ rpm -ivh heartbeat-pils-2.1.3-3.el5.centos.x86_64.rpm
shell> $ rpm -ivh heartbeat-ldirectord-2.1.3-3.el5.centos.x86_64.rpm
shell> $ rpm -ivh heartbeat-stonith-2.1.3-3.el5.centos.x86_64.rpm
shell> $ rpm -ivh heartbeat-gui-2.1.3-3.el5.centos.x86_64.rpm
shell> $ rpm -ivh heartbeat-devel-2.1.3-3.el5.centos.x86_64.rpm
shell> $ rpm -ivh heartbeat-2.1.3-3.el5.centos.x86_64.rpm
```

  - Cubrid HA 스크립트 다운로드 및 설정

```
Cubrid HA 기능을 위행 제공하는 heartbeat 관련 스크립트를 다운로드하고 설정한다.
shell> $ cd $HOME
shell> $ wget http://www.cubrid.com/files/heartbeat_config.tar.gz
shell> $ tar xvfz heartbeat_config.tar.gz
shell> $ ls heartbeat
config   ocf   resources
```

  - Cubrid HA 스크립트 주요 디렉터리 구성

○ config : heartbeat daemon 구동을 위한 파일
○ ocf : heartbeat OCF(Open Cluster Framework) 스크립트 파일
○ resources : heartbeat 리소스 파일

▪ Cubrid용 OCF는 3개가 제공되며 역할은 다음과 같다.(Cubrid Broker Process는 관리되지 않는다)

○ cubrid : DB 서버를 start/stop/monitor/promote/demote하는 스크립트

○ cubrid-lw : Log writer 용 스크립트

○ cubrid-la : Log applier 용 스크립트

▪ Heartbeat 데몬을 위한 ha.cf 설정 파일 수정 및 데몬 등록

○ ha.cf : 기본 Heartbeat 설정 파일

○ authkeys : Heartbeat가 설치된 노드 사이의 통신을 위한 인증 키

○ logd.cf : 로깅 관련 설정 파일

▪ ha.cf 파일 수정 및 복사

```
shell> $ cd $HOME/heartbeat/config
shell> cat ha-1to1.cf
use_logd yes
mcast eth0 239.0.0.1 694 1 0
node host1 host2
#autojoin any
crm on
keepalive 1
warntime 2
deadtime 3

shell> $ cp -f ha-1to1.cf /etc/ha.d/ha.cf
shell> $ cp authkeys /etc/ha.d/authkeys
shell> $ cp -f logd.cf /etc/logd.cf
shell> chmod 600 /etc/ha.d/authkeys
```

▪ OCF 스크립트 복사 및 실행 권한 추가

```
OCF 스크립트를 통해 heartbeat에서 DB를 제어하게 된다.

shell> $ cd $HOME/heartbeat
shell> $ cp -f ocf/* /usr/lib/ocf/resource.d/heartbeat/
shell> $ chmod +x /usr/lib/ocf/resource.d/heartbeat/cubrid*
```

▪ CIB(Cluster Information Base) 리소스 파일 생성 및 등록

cub_constraints-1to1.xml 리소스 파일을 다음과 같이 생성한다(sed를 사용하는 것은 일부 문자열을 변경하기 위함이다).

```
shell> $ cd $HOME/heartbeat/resources
shell> $ sed 's/cdnt10v1.cub/host1/g' constraints-1to1.xml
| sed 's/cdnt10v2.cub/host2/g' \ | sed 's/xdb/demodb/g' > cub_constraints-1to1.xml
```

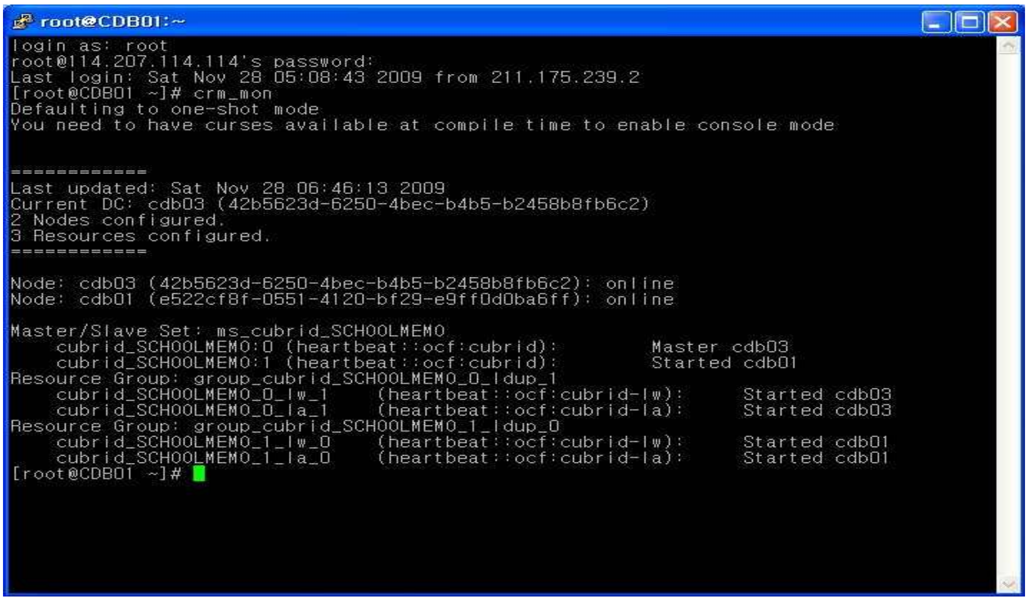
▪ heartbeat에서 제공하는 관리용 유틸리티(cibadmin)를 통해 리소스를 등록

리소스는 최초 한 번만 등록하면 되고 서버나 heartbeat가 재실행될 때 자동으로 과거에 등록한 리소스를 사용한다. cibadmin 유틸리티로 리소스를 등록하기 위해선 Heartbeat가 먼저 실행되어 있어야 한다(cibadmin으로 리소스를 등록할 때 너무 빨리 등록하면 Heartbeat가 오작동을 일으킬 수 있으므로 10초 정도의 여유 간격을 두고 등록해야 한다)

```
shell> $ /etc/init.d/heartbeat start
shell> $ cd $HOME/heartbeat/resources
shell> $ /usr/sbin/cibadmin -V -o crm_config -U -x crm_config-M2.xml
shell> $ /usr/sbin/cibadmin -V -o constraints -R -x cub_constraints-1to1.xml
shell> $ /usr/sbin/cibadmin -V -o resources -R -x cub_resources-1to1.xml
```

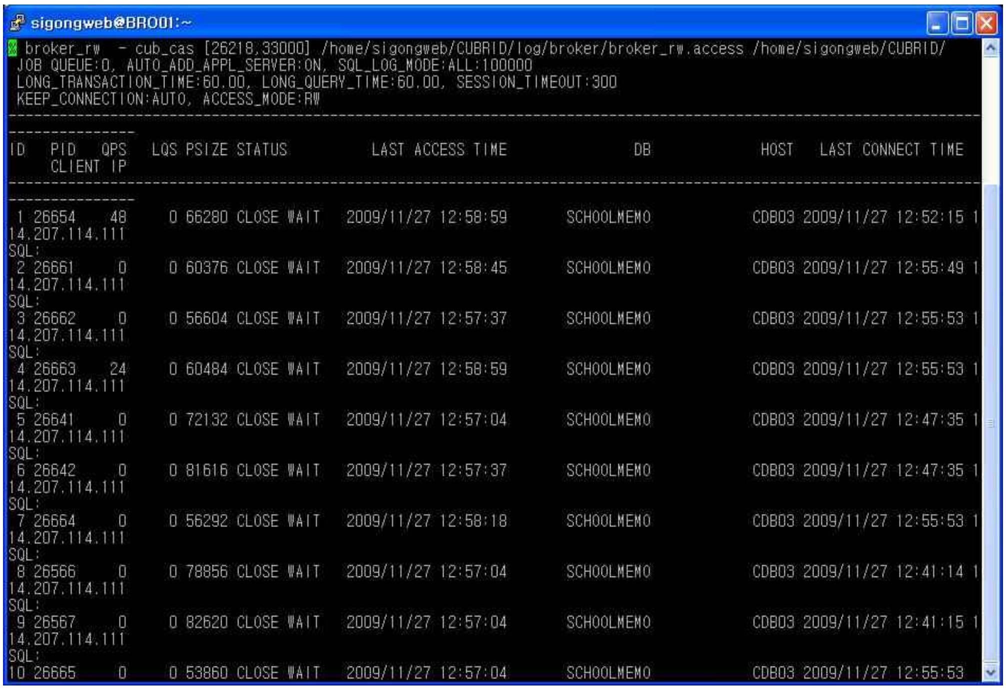
- Cluster 테스트

▪ Cluster 상태 확인(crm_mon 명령)



[그림 4-13. Cluster 상태 확인]

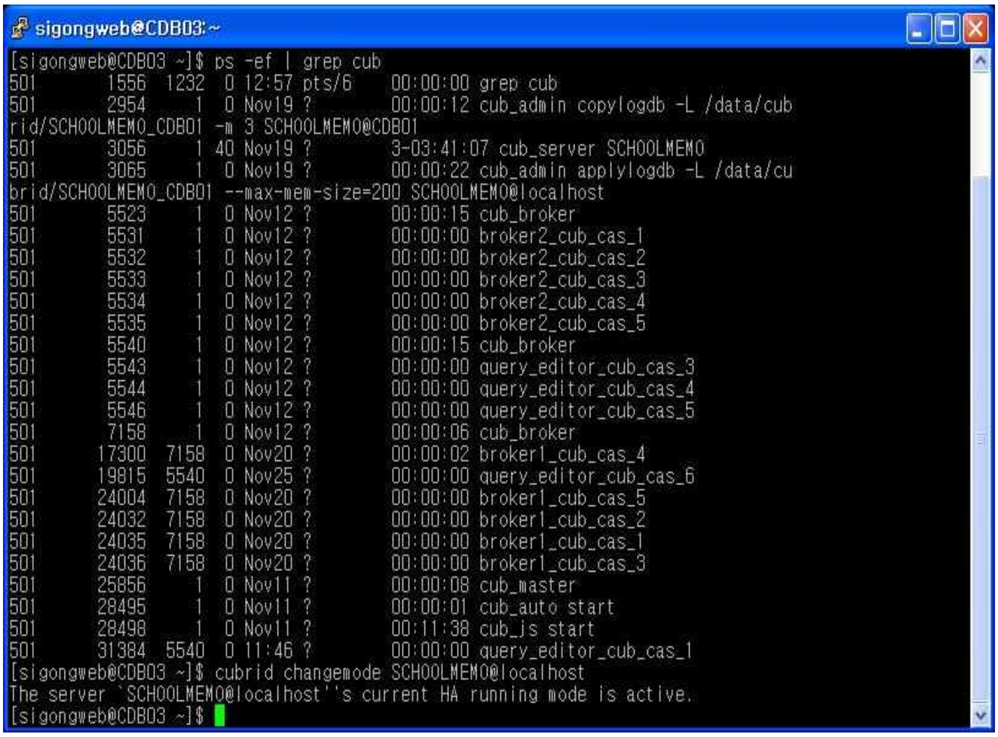
▪ JDBC 를 통한 CUBRID BROKER TEST(Active DB인 CDB03 접근)



[그림 4-14. CUBRID BROKER 테스트 화면]

▪ Active 서버 확인

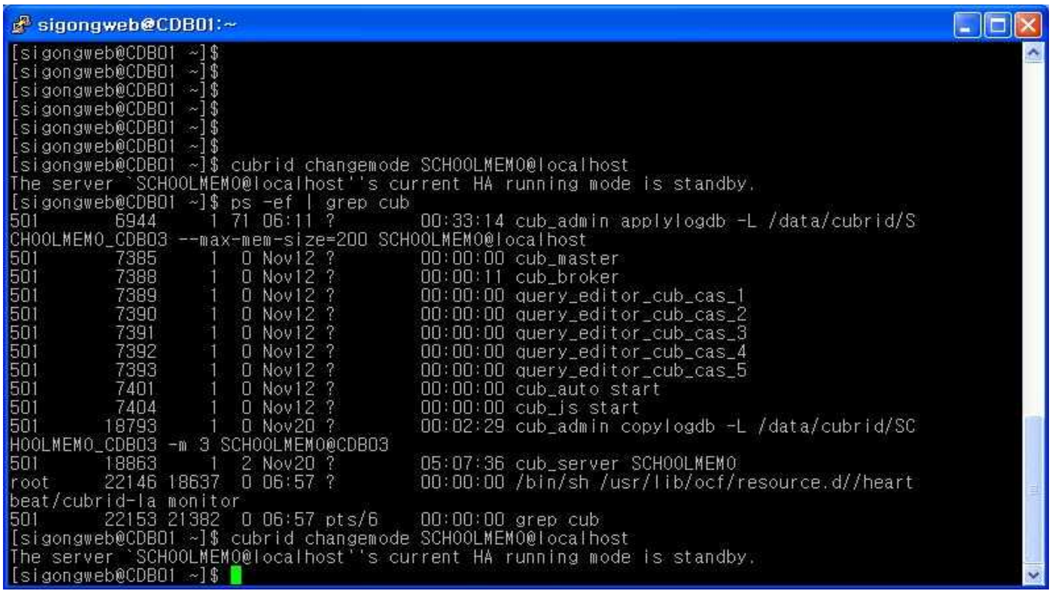
$cubrid changemode SCHOOLMEMO@localhost



[그림 4-15. Active 서버 상태 확인]

▪ StandBy 서버 확인

$cubrid changemode SCHOOLMEMO@localhost

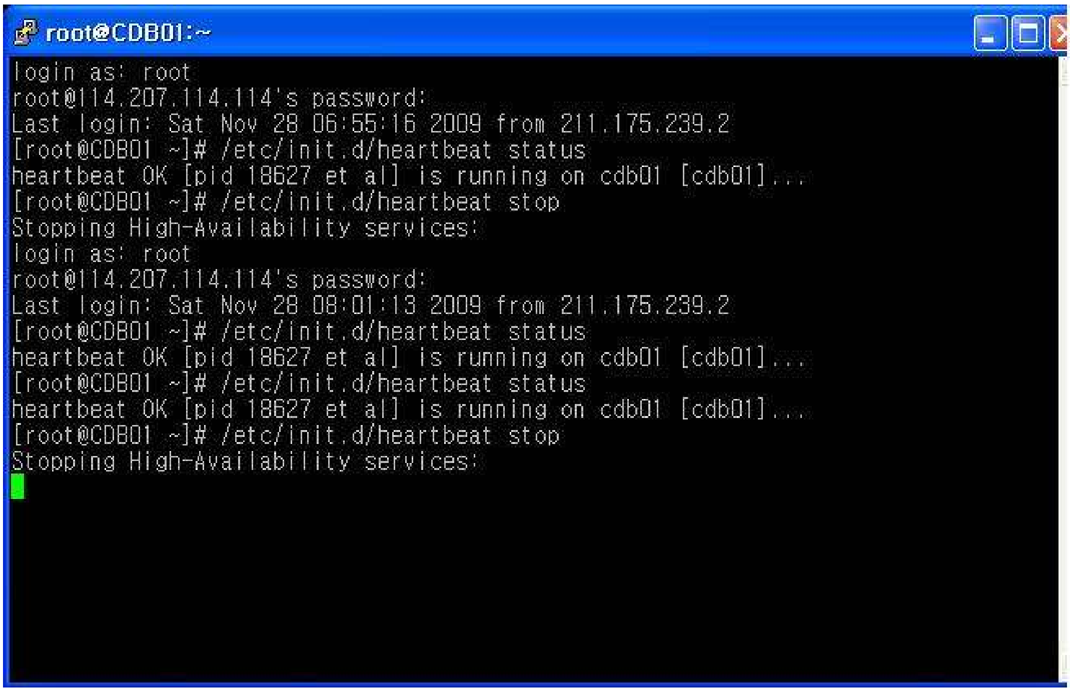


```
sigongweb@CDB01:~
[sigongweb@CDB01 ~]$
[sigongweb@CDB01 ~]$
[sigongweb@CDB01 ~]$
[sigongweb@CDB01 ~]$
[sigongweb@CDB01 ~]$
[sigongweb@CDB01 ~]$
[sigongweb@CDB01 ~]$ cubrid changemode SCHOOLMEMO@localhost
The server `SCHOOLMEMO@localhost''s current HA running mode is standby.
[sigongweb@CDB01 ~]$ ps -ef | grep cub
501       6944     1 71 06:11 ?        00:33:14 cub_admin applylogdb -L /data/cubrid/S
CHOOLMEMO_CDB03 --max-mem-size=200 SCHOOLMEMO@localhost
501       7385     1  0 Nov12 ?        00:00:00 cub_master
501       7388     1  0 Nov12 ?        00:00:11 cub_broker
501       7389     1  0 Nov12 ?        00:00:00 query_editor_cub_cas_1
501       7390     1  0 Nov12 ?        00:00:00 query_editor_cub_cas_2
501       7391     1  0 Nov12 ?        00:00:00 query_editor_cub_cas_3
501       7392     1  0 Nov12 ?        00:00:00 query_editor_cub_cas_4
501       7393     1  0 Nov12 ?        00:00:00 query_editor_cub_cas_5
501       7401     1  0 Nov12 ?        00:00:00 cub_auto start
501       7404     1  0 Nov12 ?        00:00:00 cub_js start
501      18793     1  0 Nov20 ?        00:02:29 cub_admin copylogdb -L /data/cubrid/SC
HOOLMEMO_CDB03 -m 3 SCHOOLMEMO@CDB03
501      18863     1  2 Nov20 ?        05:07:36 cub_server SCHOOLMEMO
root     22146 18637  0 06:57 ?        00:00:00 /bin/sh /usr/lib/ocf/resource.d//heart
beat/cubrid-la monitor
501      22153 21382  0 06:57 pts/6    00:00:00 grep cub
[sigongweb@CDB01 ~]$ cubrid changemode SCHOOLMEMO@localhost
The server `SCHOOLMEMO@localhost''s current HA running mode is standby.
[sigongweb@CDB01 ~]$
```

[그림 4-16. StandBy 서버 상태 확인]

▪ StandBy 서버 절체 확인

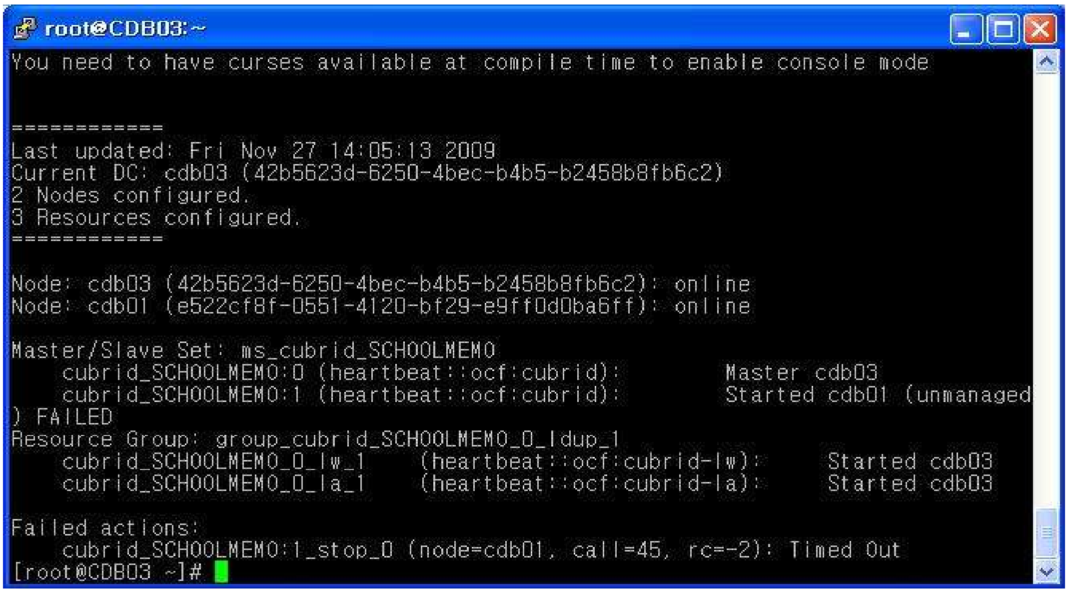
$ /etc/init.d/heartbeat stop



```
root@CDB01:~
login as: root
root@114.207.114.114's password:
Last login: Sat Nov 28 06:55:16 2009 from 211.175.239.2
[root@CDB01 ~]# /etc/init.d/heartbeat status
heartbeat OK [pid 18627 et al] is running on cdb01 [cdb01]...
[root@CDB01 ~]# /etc/init.d/heartbeat stop
Stopping High-Availability services:
login as: root
root@114.207.114.114's password:
Last login: Sat Nov 28 08:01:13 2009 from 211.175.239.2
[root@CDB01 ~]# /etc/init.d/heartbeat status
heartbeat OK [pid 18627 et al] is running on cdb01 [cdb01]...
[root@CDB01 ~]# /etc/init.d/heartbeat status
heartbeat OK [pid 18627 et al] is running on cdb01 [cdb01]...
[root@CDB01 ~]# /etc/init.d/heartbeat stop
Stopping High-Availability services:
```

[그림 4-17. StandBy 서버 절체 확인]

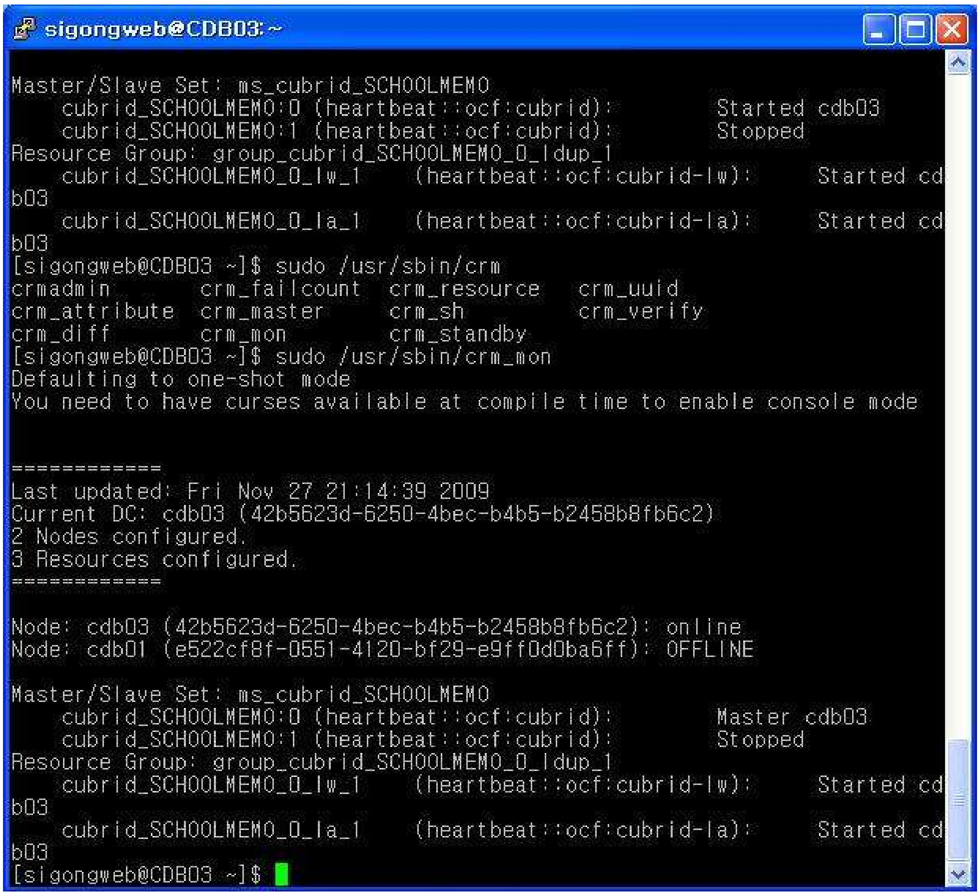
Timed Out이 되면서 StandBy 서버를 찾지 못함



```
root@CDB03:~
You need to have curses available at compile time to enable console mode


============
Last updated: Fri Nov 27 14:05:13 2009
Current DC: cdb03 (42b5623d-6250-4bec-b4b5-b2458b8fb6c2)
2 Nodes configured.
3 Resources configured.
============

Node: cdb03 (42b5623d-6250-4bec-b4b5-b2458b8fb6c2): online
Node: cdb01 (e522cf8f-0551-4120-bf29-e9ff0d0ba6ff): online

Master/Slave Set: ms_cubrid_SCHOOLMEMO
    cubrid_SCHOOLMEMO:0 (heartbeat::ocf:cubrid):        Master cdb03
    cubrid_SCHOOLMEMO:1 (heartbeat::ocf:cubrid):        Started cdb01 (unmanaged
) FAILED
Resource Group: group_cubrid_SCHOOLMEMO_0_Idup_1
    cubrid_SCHOOLMEMO_0_lw_1     (heartbeat::ocf:cubrid-lw):     Started cdb03
    cubrid_SCHOOLMEMO_0_la_1     (heartbeat::ocf:cubrid-la):     Started cdb03

Failed actions:
    cubrid_SCHOOLMEMO:1_stop_0 (node=cdb01, call=45, rc=-2): Timed Out
[root@CDB03 ~]#
```

[그림 4-18. StandBy 상태 변화 확인]

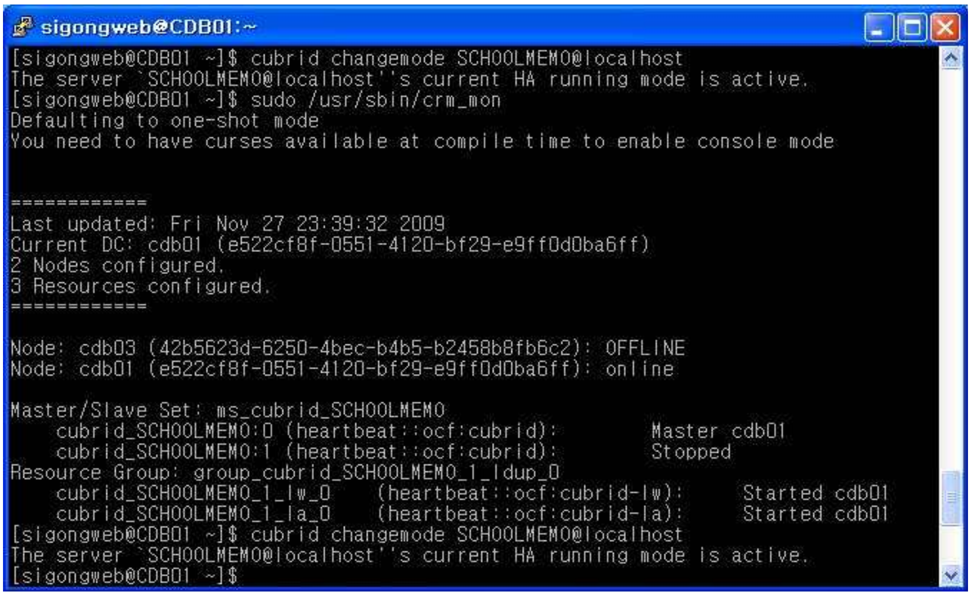
▪ crm_mon을 통한 StandBy 서버 절체 확인



```
sigongweb@CDB03:~

Master/Slave Set: ms_cubrid_SCHOOLMEMO
    cubrid_SCHOOLMEMO:0 (heartbeat::ocf:cubrid):        Started cdb03
    cubrid_SCHOOLMEMO:1 (heartbeat::ocf:cubrid):        Stopped
Resource Group: group_cubrid_SCHOOLMEMO_0_Idup_1
    cubrid_SCHOOLMEMO_0_lw_1     (heartbeat::ocf:cubrid-lw):     Started cd
b03
    cubrid_SCHOOLMEMO_0_la_1     (heartbeat::ocf:cubrid-la):     Started cd
b03
[sigongweb@CDB03 ~]$ sudo /usr/sbin/crm
crmadmin       crm_failcount  crm_resource   crm_uuid
crm_attribute  crm_master     crm_sh         crm_verify
crm_diff       crm_mon        crm_standby
[sigongweb@CDB03 ~]$ sudo /usr/sbin/crm_mon
Defaulting to one-shot mode
You need to have curses available at compile time to enable console mode


============
Last updated: Fri Nov 27 21:14:39 2009
Current DC: cdb03 (42b5623d-6250-4bec-b4b5-b2458b8fb6c2)
2 Nodes configured.
3 Resources configured.
============

Node: cdb03 (42b5623d-6250-4bec-b4b5-b2458b8fb6c2): online
Node: cdb01 (e522cf8f-0551-4120-bf29-e9ff0d0ba6ff): OFFLINE

Master/Slave Set: ms_cubrid_SCHOOLMEMO
    cubrid_SCHOOLMEMO:0 (heartbeat::ocf:cubrid):        Master cdb03
    cubrid_SCHOOLMEMO:1 (heartbeat::ocf:cubrid):        Stopped
Resource Group: group_cubrid_SCHOOLMEMO_0_Idup_1
    cubrid_SCHOOLMEMO_0_lw_1     (heartbeat::ocf:cubrid-lw):     Started cd
b03
    cubrid_SCHOOLMEMO_0_la_1     (heartbeat::ocf:cubrid-la):     Started cd
b03
[sigongweb@CDB03 ~]$
```

[그림 4-19. crm_mon명령으로 상태 확인]

▪ Active 서버 절체 확인(reboot 명령을 통하여 절체)



[그림 4-20. Active 서버 절체 확인]

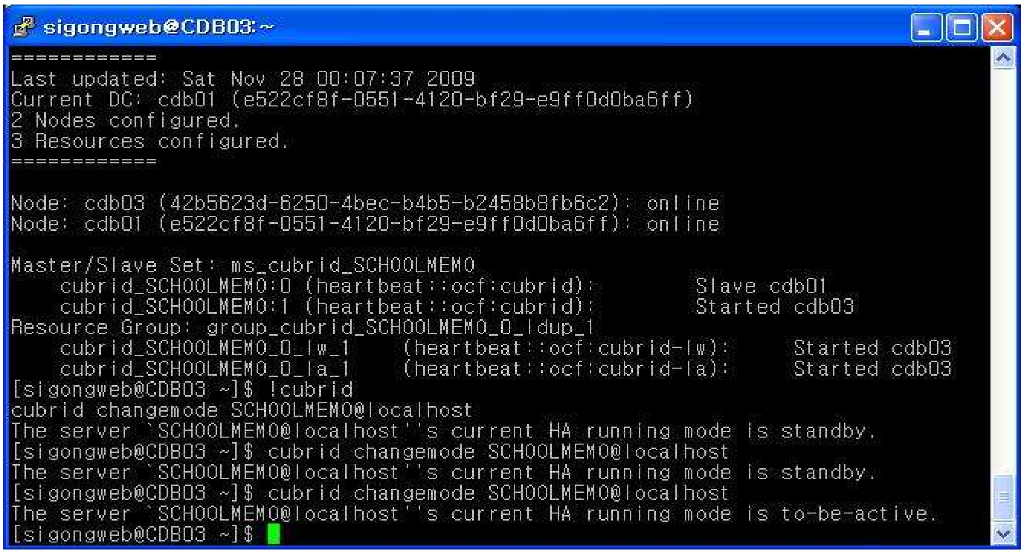
▪ HA Active 서버 절체 화면

Active 서버 절체후 JDBC 연결이 role change 되어 CDB01로 연결 되는 화면

[그림 4-21. JDBC 연결 role change 화면]

▪ HA role change 화면

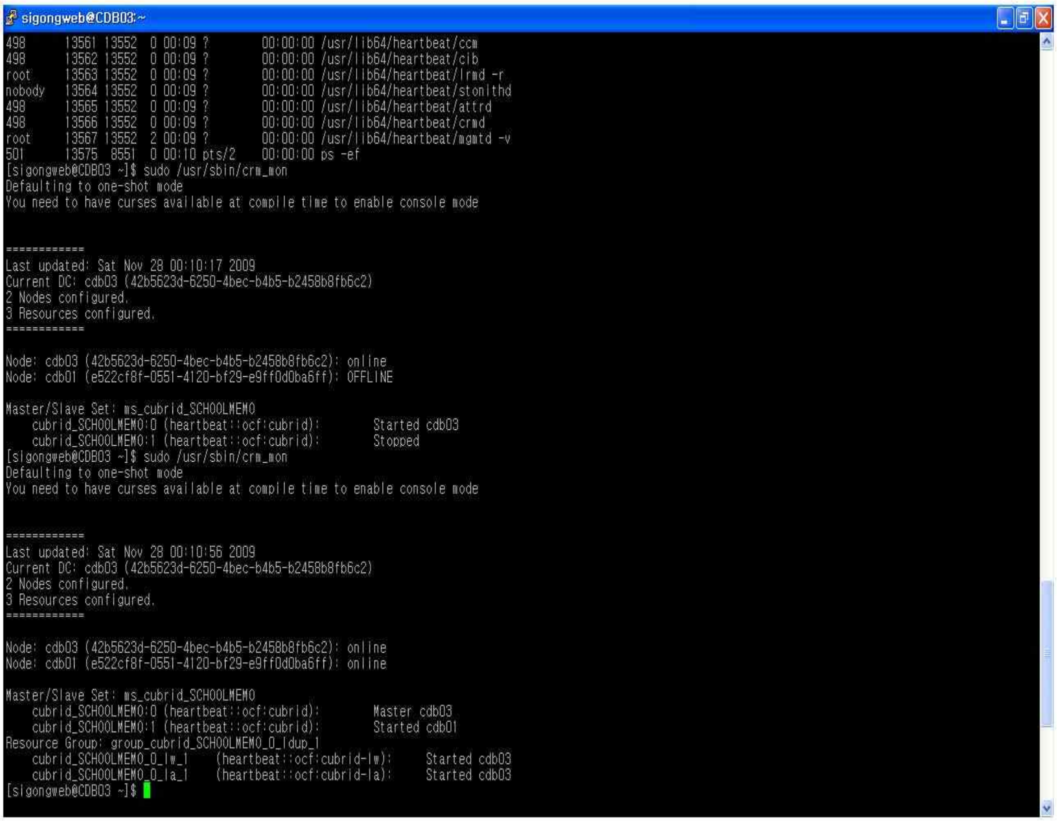
Master 절체로 인한 Slave-to-be-Active



[그림 4-22. StandBy 서버가 Active로 바뀌는 화면]

▪ HA Active 복구 후 role change 화면

Active 서버가 복구되어 처음 상태로 변경된 화면



[그림 4-23. Active 서버 정상 복구 화면]

라. 관리 도구 테스트

□ GUI Tool 지원

[표 4-13. GUI용 관리 툴]

구　분	GUI Tool 버전	지원 여부
Cubrid Manager	2008 R2.0 Build id: 8.2.0.2005	○

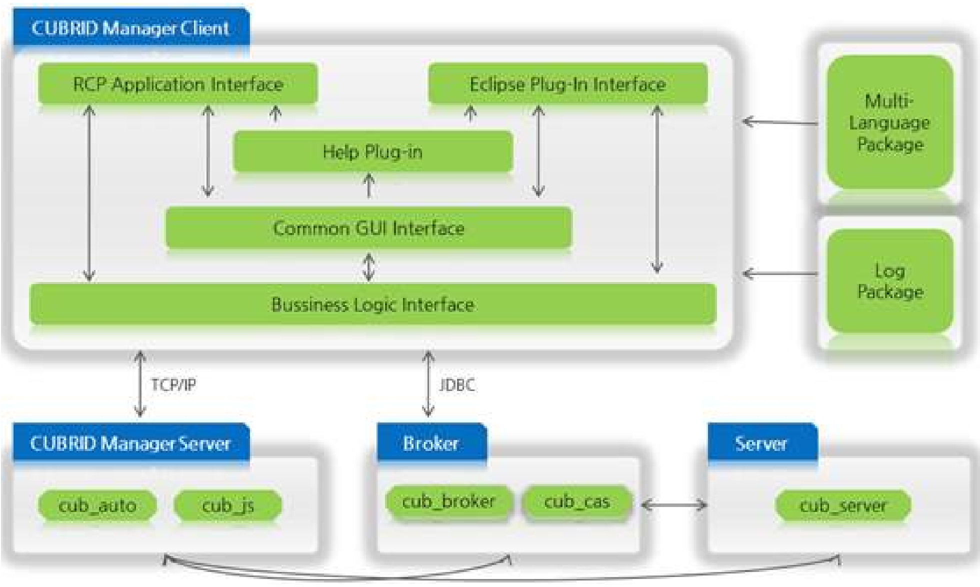
- Cubrid Manager

CUBRID 매니저는 CUBRID 데이터베이스 관리 및 질의 기능을 GUI 환경에서 제공하는 CUBRID 데이터베이스 전용 관리 도구이다. CUBRID 매니저를 사용하면 다양한 관리 작업을 쉽게 할 수 있다. CUBRID 매니저는 데이터베이스 관리 및 응용 개발에 필수적인 질의 편집기 기능을 제공한다.

▪ 데이터베이스 관리 및 응용 개발에 필수적인 질의 편집기 기능 제공

- Cubrid Manager 구조

데이터베이스 서버와 브로커가 설치된 호스트에 CUBRID 매니저 서버가 동작하고, 이 CUBRID 매니저 서버에 CUBRID 매니저 클라이언트가 접속한다.



[그림 4-24. CUBRID Manager 구조도]

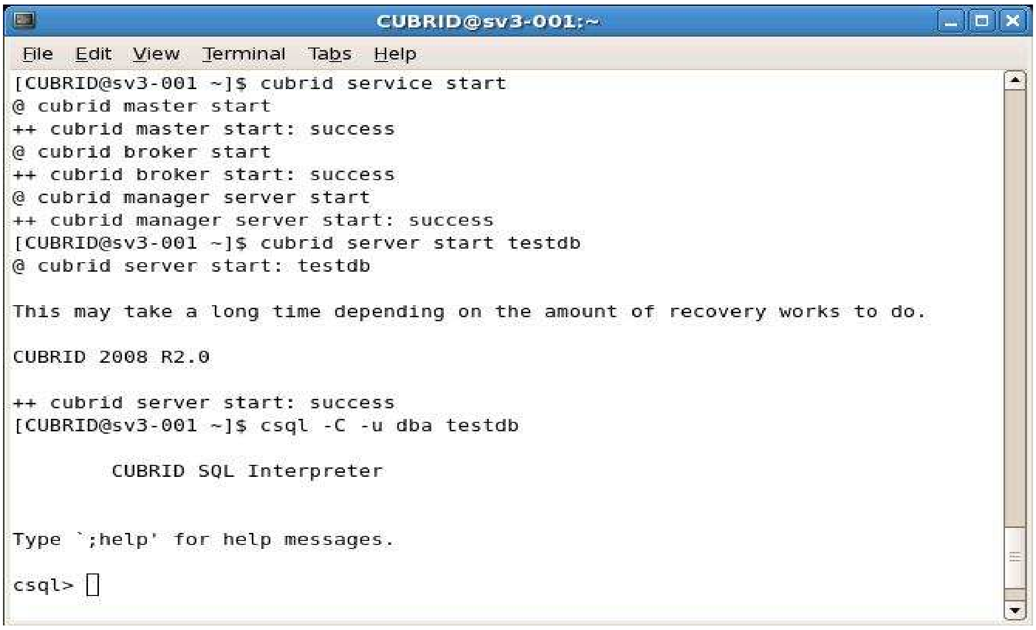
▪ cub_auto 프로세스는 CUBRID 매니저 클라이언트 사용자의 인증처리 및 주기적인 자동화 작업과 진단 정보를 수집하는 기능을 수행한다.

▪ cub_js 프로세스는 CUBRID 매니저 클라이언트로부터 전송된 사용자의 요구를 수행한다.

- Cubrid Manager 설정

```
$ cd ~/CUBRID/conf
$ cp cubrid.conf-dist cubrid.conf
$ cp cm.conf-dist cm.conf
$ cp cm.pass-dist cm.pass
$ cp cmdb.pass-dist cmdb.pass
$ cubrid service start
$ cubrid createdb testdb ☞ testdb 생성
$ cubrid server start testdb ☞ 서버를 구동
$ csql -C -u dba testdb ☞ csql로 접속
```

▪ Cubrid Manager 설정 화면



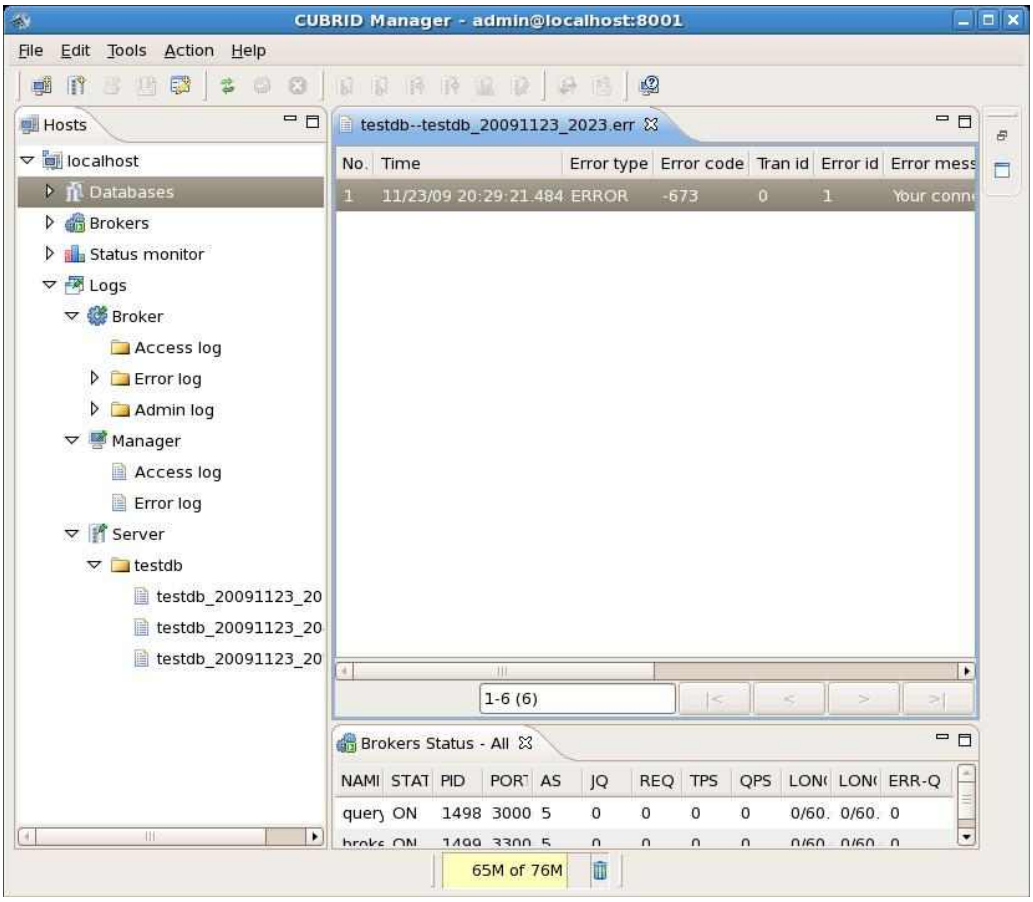
[그림 4-25. CUBRID Manager 시작 화면]

▪ Cubrid Manager Client실행

[표4-14. CUBRID Manager Client 실행]

구 분	실행 방법
Windows	[시작] > [모든 프로그램] > [CUBRID] > [CUBRID Magnger Client]를 선택한다.
Linux	cd /home/CUBRID/cubridmanager ./cubridmanager

▪ Cubrid Manager Client GUI 실행 화면



[그림 4-26. CUBRID Manager Client GUI 화면]

※ 참고 사항

Management Tool은 Cubrid DBMS를 통한 서비스 제공시 서버를 제어하고 오류를 감시하는 기능을 제공하므로 매우 중요한 기능이라 할 수 있다.

운영자는 서비스를 감시 및 제어할 수 있는 GUI기반의 관리 툴 제공을 선호한다.

5. 결론

□ 설치

CUBRID DBMS는 비교적 쉬운 설치 방법을 제공한다. 현재는 리눅스와 윈도우 기반에서의 Binary 설치 버전을 제공하지만 앞으로 유닉스 기반의 Binary 설치 버전도 제공할 것으로 보인다.

□ 성능

소프트웨어 성능은 하드웨어 성능과 비례하는데, 이번 테스트에서 32 thread를 기준으로 가장 좋은 성능을 보였으며, 32개보다 높은 thread값을 적용하였을 경우 성능이 저하되었다.

□ 기능

데이터베이스의 기능 중 클러스터 기능은 매우 중요한 기능 중 하나이다. 장애에 대비한 이중화 구성을 기본으로 하는 클러스터를 제공함으로서 큐브리드의 장애 대처 능력이 다른 상용 DBMS와 크게 차이가 없음을 보여 주었다.

※ Active/Active 구성의 클러스터 기능은 향후 추가 계획에 있음을 확인함

□ 관리

CUBRID Management Tool은 GUI 환경에서 쉽게 제어가 가능하여 일반 사용자들도 쉽게 관리가 가능하므로 사용자들에게 큰 호응을 얻을 것으로 보인다.

다만, 일부 기능이 현재 개발 중이거나 지원이 아직 미흡하여 지속적인 개발과 기능 업그레이드가 필요해 보인다.

국산 DBMS 솔루션으로 NHN이 인수하여 후원을 하고 있으며, NHN 내부의 상용 서비스에 이미 사용되고 있음을 감안할 때 Oracle과 같은 글로벌 DBMS와의 경쟁에서도 손색이 없는 우수한 제품이 될 것으로 기대한다.

* 참고 자료

http://www.cubrid.com
http://jakarta.apache.org/jmeter/

Ⅲ Fedora

목 차

1. 개요

본 문서는 공개SW 역량프라자 운영 사업의 공개SW 테스트 환경 구축을 통하여 테스트한 공개SW에 대한 기능, 성능, 호환성 등을 검증하고, 향후 공공 기관 및 기업에서 도입 시 참고할 수 있는 자료로 활용하기 위하여 작성되었다.

가. 목적

□ 공개SW를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 공개SW의 기능, 성능, 안정성, 호환성 등 검증
- 공개SW 기술의 도입 가이드라인 제공
- 효과적인 공개SW 도입 유도
- 안정적인 공개SW 시스템 도입

나. 선정 기준

□ 공개SW 테스트 대상은 다음 5가지 사항을 고려하여 선정되었다.

- 1번 : 국내에서 개발된 제품
- 2번 : 국내 기업에서 많이 도입되는 제품
- 3번 : Global에서 많이 알려진 제품
- 4번 : 기술지원(업데이트) 가능한 제품
- 5번 : OS간 호환 가능 제품(Package 관리 방식)

[표 1-1. 선정기준]

Linux 운영체제(OS)	1번	2번	3번	4번	5번	결과	비고
Asianux	●	◕	◕	●	●	●	선정*
CentOS	○	◑	●	●	●	◕	선정
Debian	○	◔	◕	●	◔	◑	보류
Fedora	○	◑	●	●	●	◕	선정
Gentoo	○	◔	◕	◕	◔	◑	보류
GINUX	●	◑	◔	●	●	◕	선정
Red Hat Enterprise Linux	○	●	●	●	●	●	선정*
Slackware	○	◔	◕	◕	◔	◑	보류
SULinux	●	◑	◔	●	●	◕	선정
SuSE Linux	○	◑	●	●	●	◕	선정*
Ubuntu	○	◑	●	●	◔	◑	보류

※ 테스트 대상 기준에 근거하여 이번 테스트 대상에서 제외된 **선정*** 표시 제품은 금번 사업의 일정상의 제약으로 다음 사업에서 테스트를 수행하기로 함

2. 테스트 환경 구성

가. 테스트 일정

[표 2-1. 일정표]

구분	월	10				11				12			
	주	1	2	3	4	1	2	3	4	1	2	3	4
테스트 대상 발굴 및 선정													
테스트 환경 구성													
테스트 항목 선정													
테스트 시나리오 작성													
테스트 수행													

나. 테스트 대상 소개

□ Fedora

레드햇에서 후원하는 페도라 프로젝트에서 개발한 RPM 기반의 리눅스 배포판이며, 개인과 기업의 다양한 컴퓨터 환경에서 사용될 수 있도록 만들어져 있으며, 특히 그래픽 기반의 설치 프로그램인 아나콘다와 시스템-환경설정 도구를 이용해 누구나 쉽게 운영체제를 설치할 수 있는 것이 특징이다. 페도라는 GNOME과 KDE 환경을 모두 지원하고 있다. 현재 페도라의 주 목적은 자유 소프트웨어와 오픈 소프트웨어를 포함하는 것만 아니라, 그러한 기술들의 첨단을 이끌어 나가는 데에 있다.

□ Fedora 주요 특징

- ext4를 기본 파일 시스템으로 채택
- Btrfs로 포맷하고 설치 가능(부팅할 때 IcantbelieveitsnotBTR 옵션 추가)
- 컴퓨터 전원 가동부터 데스크톱 진입까지 걸리는 시간: 20초 내외
- System 보안 서비스 제공(SSSD)
- 가상화 : KVM, Xen 지원 및 많은 가상화 플랫폼 지원
  - Virtual Machine 관리를 위한 VNC 인증 개선
  - Virtual Machine을 위한 GUI 콘솔 개선
  - KVM PCI 장치 할당
  - KVM과 QEMU의 Merge
  - Svirt 명령의 접근 제어(가상화에서의 SELinux 사용)
  - libbirt 0.6.1로 기능 개선
  - Linux 및 Solaris 호스트에서 Xen 하이퍼바이저
  - QEMU 에뮬레이터
  - KVM Linux 하이퍼바이저
  - LXC Linux 컨테이너 시스템

▪ OpenVZ Linux 컨테이너 시스템
▪ IDE/SCSI/USB 디스크, FibreChannel, LVM, iSCSI, NFS 상의 스토리지
- GNU C 컴파일러 4.4
- K 데스크톱 환경 4.2.2
- GNOME 2.26.1
- 리눅스 2.6.29-4
- Eclipse 3.4.2
- Netbeans 6.5
- 오픈 소스 기반의 nVidia 드라이버의 커널 모드 세팅
- OpenOffice 3.1.0
- Phython 2.6
- Xfce 4.6
- X Window 시스템 서버 1.6
- KMS(Kernel Mode Setting)

다. 테스트 환경

□ 테스트 서버

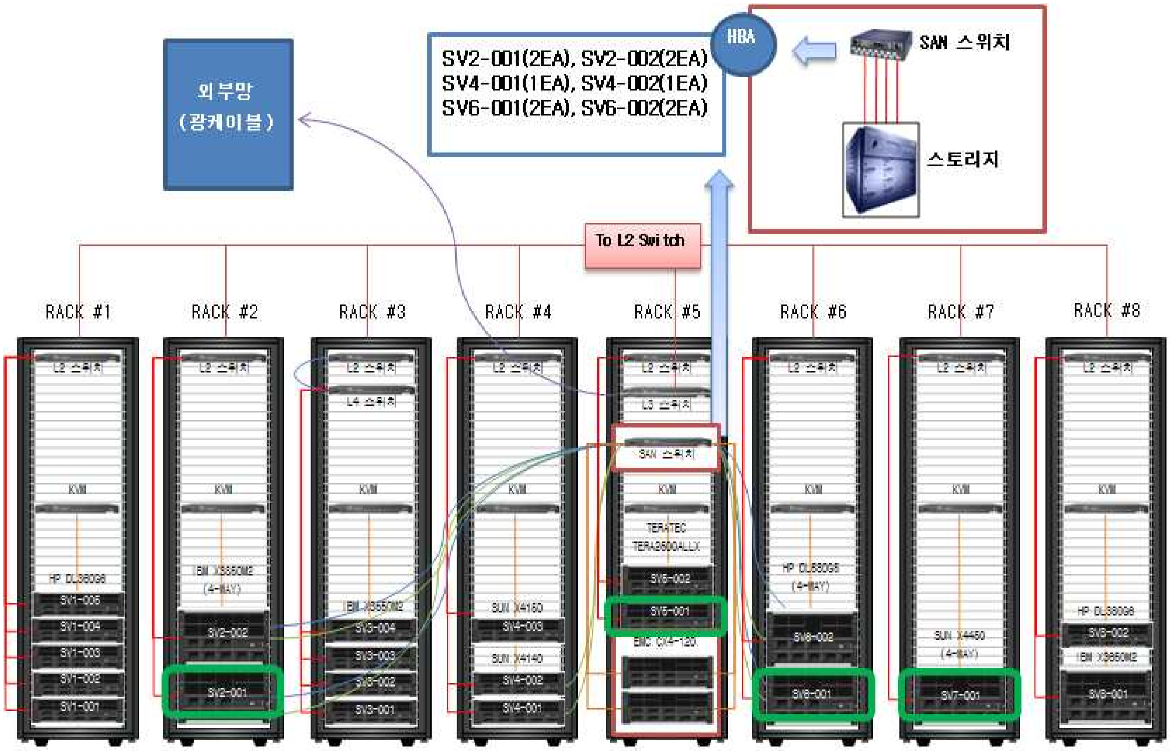
[표 2-2. 테스트 서버 규격]

구분	제품명	CPU	MEM	Disk	NIC
HP	DL580 G5	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
IBM	X3850 M2	Quad-Core 2.4Ghz 4-Way	8GB	146GB*4	Gigabit 2Port
SUN	X4450	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
TERATEC	TERA2500ALLX	Quad-Core 2.5Ghz 2-Way	8GB	147GB*5	Gigabit 2Port

□ 테스트 조건

- 32bit 환경에서 64bit환경으로 서버 구성이 전환됨에 따라 64bit 환경에서의 테스트를 수행
- 성능 테스트는 SUN X4450 서버에서 수행

□ 테스트 실장도



☞ 박스 안의 서버가 OS 테스트용 서버로 사용 됨.

[그림 2-1. 테스트 서버 실장도]

□ 서버 위치

[표 2-3. 서버 위치]

제조사	모델명	Rack 위치	비 고
IBM	X3850 M2	#2	
TERATEC	TERA2500ALLX	#5	
HP	DL580 G5	#6	
SUN	X4450	#7	

3. 테스트 시나리오

가. 설치 및 호환성 검증

□ DVD-ROM을 통한 설치

□ 하드웨어 정보 확인

□ 4대의 하드웨어 벤더별 설치 상태 확인

나. 기능 테스트

□ 네트워크 서비스 지원

- Web, Ftp, NFS, Samba, Openssh, Telnet

□ 파일시스템 지원

- ext3, XFS, GFS 등

□ LVM 지원 확인

□ 보안 기능 지원

- tcp-wrapper
- GUI용 보안 툴

□ 자동 업데이트 기능 지원

- On-Line 업데이트

다. 신뢰성 테스트

□ LTP Tool 사용한 테스트

- ltp-full-20090930 버전 사용

라. 안정성 테스트

□ LTP Stress Tool 사용

- CPU, Memory 등의 사용량 체크

4. 테스트

가. 설치 및 호환성 검증

□ Fedora 설치 환경

- Fedora 설치 버전

[표 4-1. OS 설치 정보]

OS 버전	지원 유형	Kernel 버전
Fedora 11	X86_64(EM64T, AMD64)	2.6.29.4-167.fc11

- 파티션 정보

[표 4-2. 파티션 정보]

<table>
<tr><th>디스크
(RAID)</th><th>Physical
Volume</th><th>Volume
Group</th><th>Logical
Volume</th><th>Mount
Point</th><th>용량
(MB)</th></tr>
<tr><td rowspan="9">/dev/sda(580GB)
RAID 5</td><td>/dev/sda1</td><td>-</td><td>-</td><td>/boot</td><td>100</td></tr>
<tr><td rowspan="5">/dev/sda2</td><td rowspan="5">VG0</td><td>lv_root</td><td>/</td><td>20480</td></tr>
<tr><td>lv_swap</td><td>-</td><td>8192</td></tr>
<tr><td>lv_usr</td><td>/usr</td><td>30720</td></tr>
<tr><td>lv_home</td><td>/home</td><td>20480</td></tr>
<tr><td>lv_var</td><td>/var</td><td>10240</td></tr>
<tr><td rowspan="3">/dev/sda3</td><td rowspan="3">VG1</td><td>lv_data1</td><td>/data1</td><td>251200</td></tr>
<tr><td>lv_data2</td><td>/data2</td><td>51200</td></tr>
<tr><td>lv_data3</td><td>/data3</td><td>51200</td></tr>
</table>

※ 디스크 용량은 서버별 차이가 있으므로 위의 기준으로 백분율(%) 환산 적용하며 남는 공간은 여유 공간으로 비워 둠

□ 서버별 설치 및 호환성 검증

[표 4-3. OS 설치 방법 및 결과]

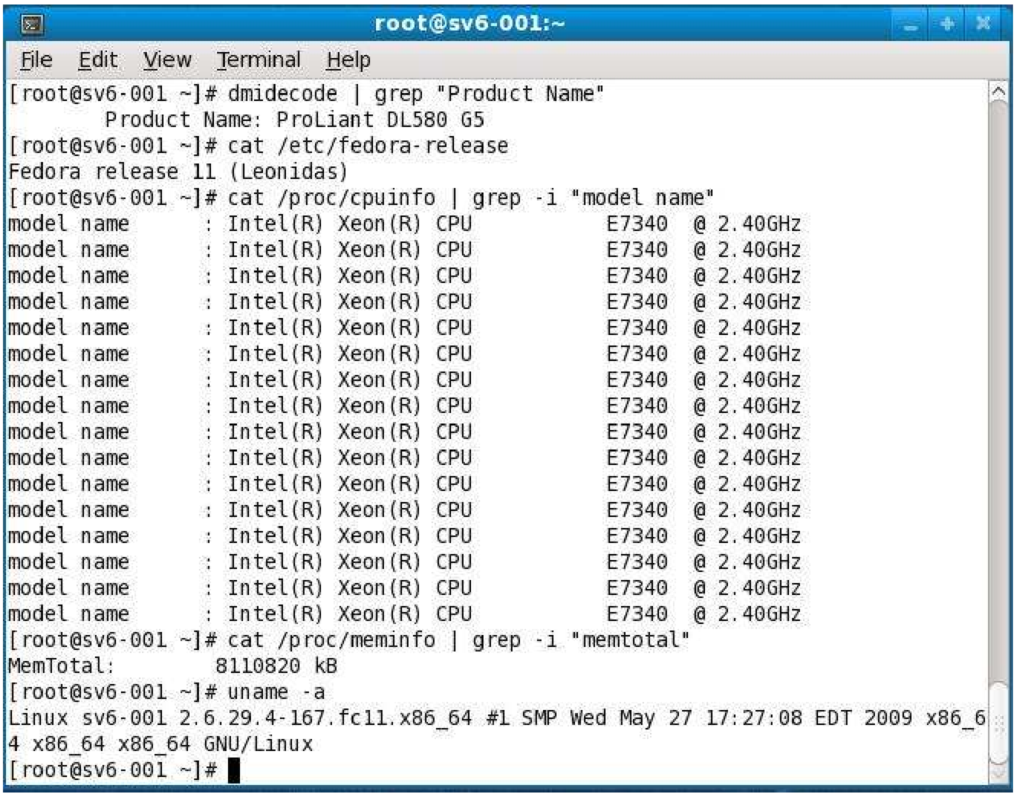
구분	설치 매체	설치 방식	패키지 선택 방식	설치 결과
HP DL580 G5	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
IBM X3850 M2	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
SUN X4450	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
TERATEC TERA2500ALLX	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상

- 서버별 Fedora 설치 상태 확인 명령

[표 4-4. SULinux 설치 상태 확인]

구분	명령어	결과
서버 모델명 확인	dmidecode | grep "Product Name"	[그림4-1~4] 참고
OS 버전 확인	cat /etc/ginux-release	
CPU 정보	cat /proc/cpuinfo | grep -i "model name"	
Memory 정보	cat /proc/meminfo | grep -i "memtotal"	
Kernel 정보	uname -a	

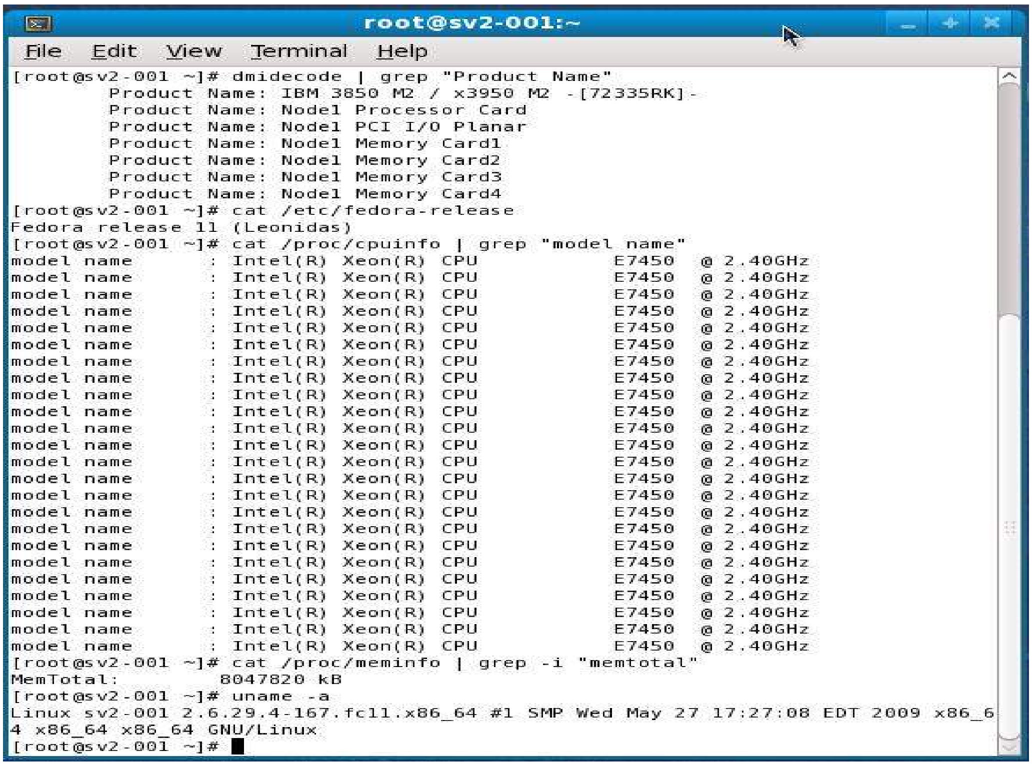
▪ HP DL580 G5 모델에서 Fedora 설치 확인



```
root@sv6-001:~
File Edit View Terminal Help
[root@sv6-001 ~]# dmidecode | grep "Product Name"
        Product Name: ProLiant DL580 G5
[root@sv6-001 ~]# cat /etc/fedora-release
Fedora release 11 (Leonidas)
[root@sv6-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
[root@sv6-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:        8110820 kB
[root@sv6-001 ~]# uname -a
Linux sv6-001 2.6.29.4-167.fc11.x86_64 #1 SMP Wed May 27 17:27:08 EDT 2009 x86_6
4 x86_64 x86_64 GNU/Linux
[root@sv6-001 ~]#
```

[그림 4-1. HP DL580G5에서 Fedora 상태 확인]

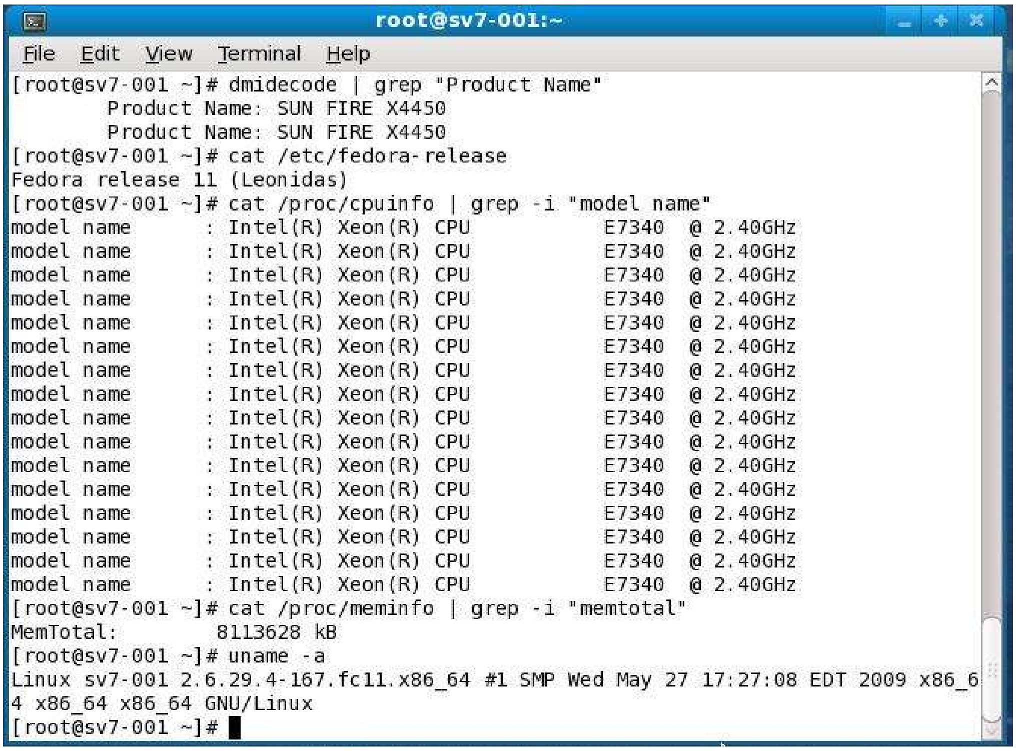
▪ IBM X3850 M2 모델에서 Fedora 설치 확인



```
root@sv2-001:~
File  Edit  View  Terminal  Help
[root@sv2-001 ~]# dmidecode | grep "Product Name"
        Product Name: IBM 3850 M2 / x3950 M2 -[72335RK]-
        Product Name: Node1 Processor Card
        Product Name: Node1 PCI I/O Planar
        Product Name: Node1 Memory Card1
        Product Name: Node1 Memory Card2
        Product Name: Node1 Memory Card3
        Product Name: Node1 Memory Card4
[root@sv2-001 ~]# cat /etc/fedora-release
Fedora release 11 (Leonidas)
[root@sv2-001 ~]# cat /proc/cpuinfo | grep "model name"
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
[root@sv2-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:        8047820 kB
[root@sv2-001 ~]# uname -a
Linux sv2-001 2.6.29.4-167.fc11.x86_64 #1 SMP Wed May 27 17:27:08 EDT 2009 x86_6
4 x86_64 x86_64 GNU/Linux
[root@sv2-001 ~]#
```

[그림 4-2. IBM X3850M2에서 Fedora 상태 확인]

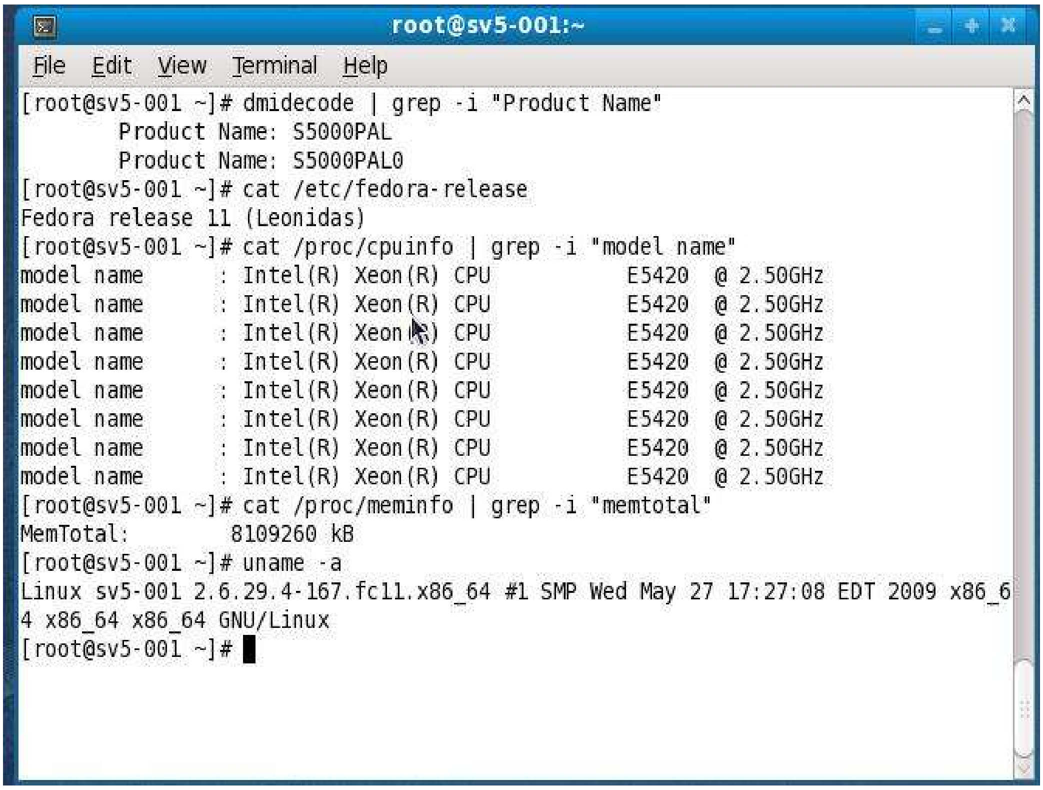
▪ SUN X4450 모델에서 Fedora 설치 확인



```
root@sv7-001:~
File  Edit  View  Terminal  Help
[root@sv7-001 ~]# dmidecode | grep "Product Name"
        Product Name: SUN FIRE X4450
        Product Name: SUN FIRE X4450
[root@sv7-001 ~]# cat /etc/fedora-release
Fedora release 11 (Leonidas)
[root@sv7-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
[root@sv7-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:        8113628 kB
[root@sv7-001 ~]# uname -a
Linux sv7-001 2.6.29.4-167.fc11.x86_64 #1 SMP Wed May 27 17:27:08 EDT 2009 x86_6
4 x86_64 x86_64 GNU/Linux
[root@sv7-001 ~]#
```

[그림 4-3. SUN X4450에서 Fedora 상태 확인]

▪ TERATEC TERA2500ALLX 모델에서 Fedora 설치 확인



```
root@sv5-001:~
File  Edit  View  Terminal  Help
[root@sv5-001 ~]# dmidecode | grep -i "Product Name"
        Product Name: S5000PAL
        Product Name: S5000PAL0
[root@sv5-001 ~]# cat /etc/fedora-release
Fedora release 11 (Leonidas)
[root@sv5-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
[root@sv5-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:        8109260 kB
[root@sv5-001 ~]# uname -a
Linux sv5-001 2.6.29.4-167.fc11.x86_64 #1 SMP Wed May 27 17:27:08 EDT 2009 x86_6
4 x86_64 x86_64 GNU/Linux
[root@sv5-001 ~]#
```

[그림 4-4. TERATEC TERA2500ALLX에서 Fedora 상태 확인]

※ 참고 사항

dmidecode 명령은 하드웨어 업체에서 정보를 제공하는 경우만 정상적인 결과 값을 얻을 수 있는데 테라텍 서버의 경우 모델명이 일치하지 않음

나. 기능 테스트

□ 네트워크 서비스 지원

[표 4-5. Fedora 네트워크 서비스 확인]

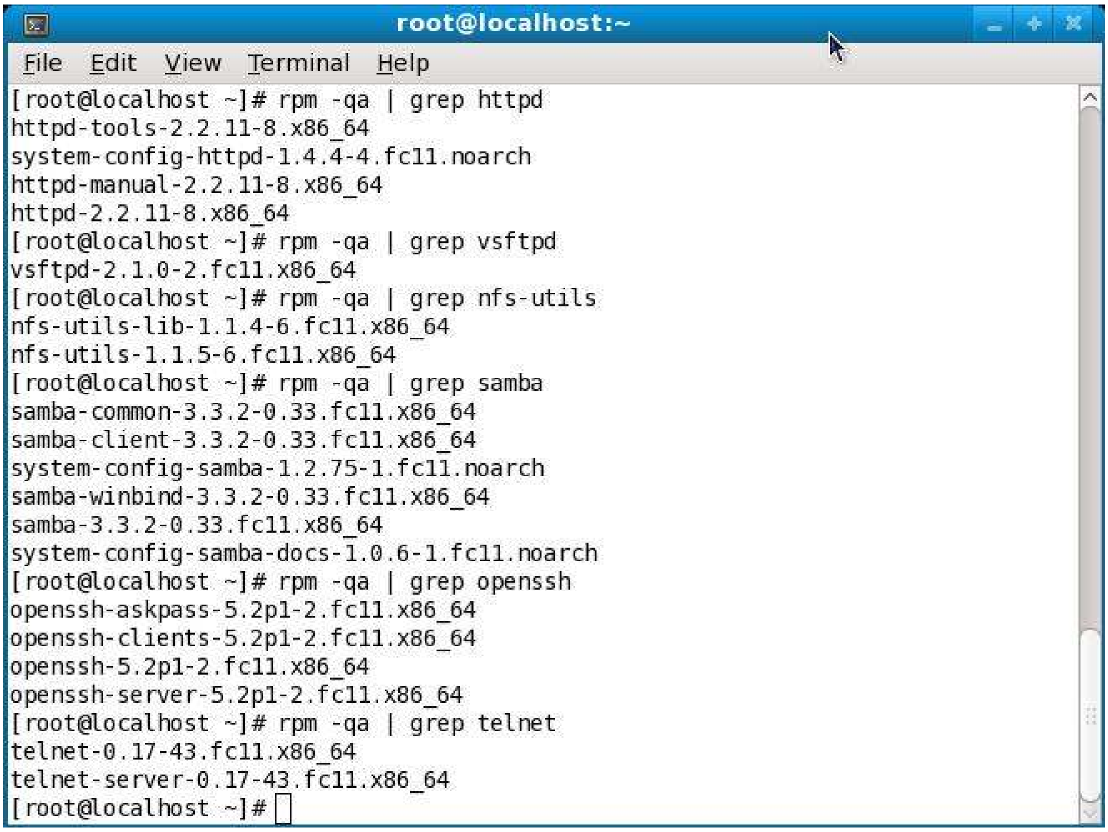
서비스 명	패키지 명	버전	지원 여부	비고
WEB	httpd	2.2.11-8	지원	
FTP	vsftpd	2.1.0-2	지원	
NFS	nfs-utils	1.1.5-6	지원	
Samba	samba	3.3.2-0.33	지원	
SSH	openssh	5.2p1-2	지원	
Telnet	telnet-server	0.17-43	지원	

- 서비스별 지원 확인 명령

[표 4-6. 서비스별 지원 명령]

구분	명령어	결과
서비스 확인 명령	rpm -qa | grep "*package_name*"	[그림4-5] 참고

▪ 서비스별 지원



[그림 4-5. Fedora 서비스 지원 확인]

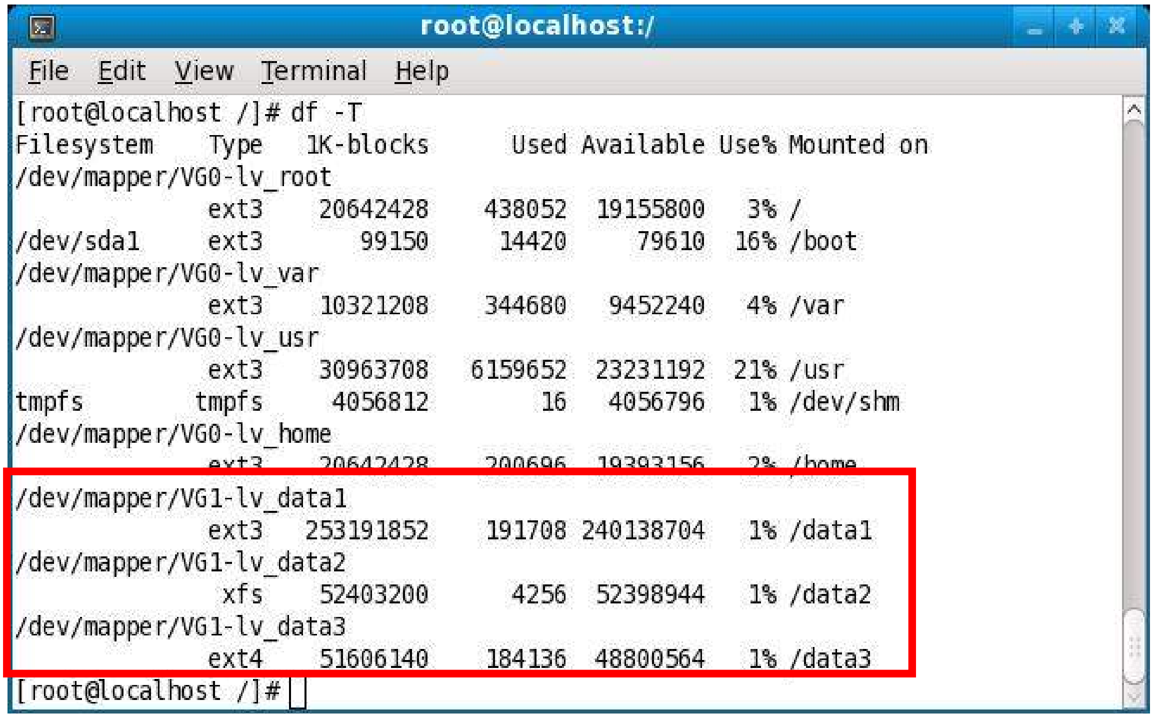
□ 파일시스템 지원

- 파일 시스템 지원 확인 명령

[표 4-7. 파티션별 지원 확인 명령]

구분	명령어	결과
ext3	mkfs -t ext3 /dev/VG1/lv_data1 mount /dev/VG1/lv_data1 /data1	[그림4-6] 참고
xfs	mkfs -t xfs /dev/VG1/lv_data2 mount /dev/VG1/lv_data2 /data2	
ext4	mkfs -t ext4 /dev/VG1/lv_data3 mount /dev/VG1/lv_data3 /data3	
gfs	N/A(지원 안함)	
파일시스템 확인 명령	df -T	

▪ ext3, xfs, ext4 파일시스템



[그림 4-6. 파티션별 지원 확인]

□ LVM 지원 확인

[표 4-8. LVM 지원 확인]

패키지명	버전	지원여부	비 고
lvm2	2.02.45-4	**지원**	

- LVM 확인 명령

[표 4-9. LVM 지원 확인 명령]

구분	명령어	결과
LVM 확인 명령	rpm -qa | grep "*package_name*"	[그림4-7] 참고

▪ LVM 확인

[그림 4-7. LVM 지원 확인]

※ LVM(Logical Volume Manager)

저장장치들을 좀 더 효율적이고 유연하게 관리할 수 있는 커널의 부분 프로그램.

- LVM 사용의 장점
  - ▪ 쉬운 관리
  - ▪ 서로 다른 많은 디바이스 조합 지원
  - ▪ 직관적인 저장 장치로의 접근
  - ▪ 뛰어난 확장성
  - ▪ 신뢰되는 안정성과 효율성
  - ▪ 스냅샷 기능 지원
  - ▪ 대용량 단일 파일시스템 구성 지원

□ 보안 기능 지원

[표 4-10. 보안 기능 지원 확인]

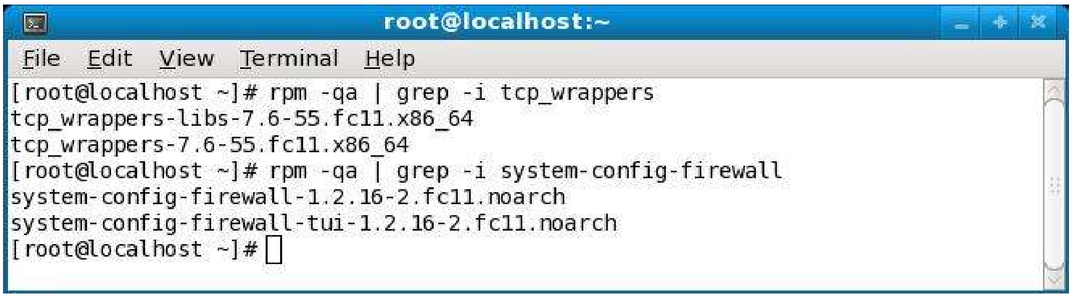
패키지명	버전	지원여부	비 고
tcp_wrappers	7.6-55	지원	
system-config-securitylevel	1.2.16-2	지원	

- 보안 기능 확인 명령

[표 4-11. 보안 기능 지원 명령]

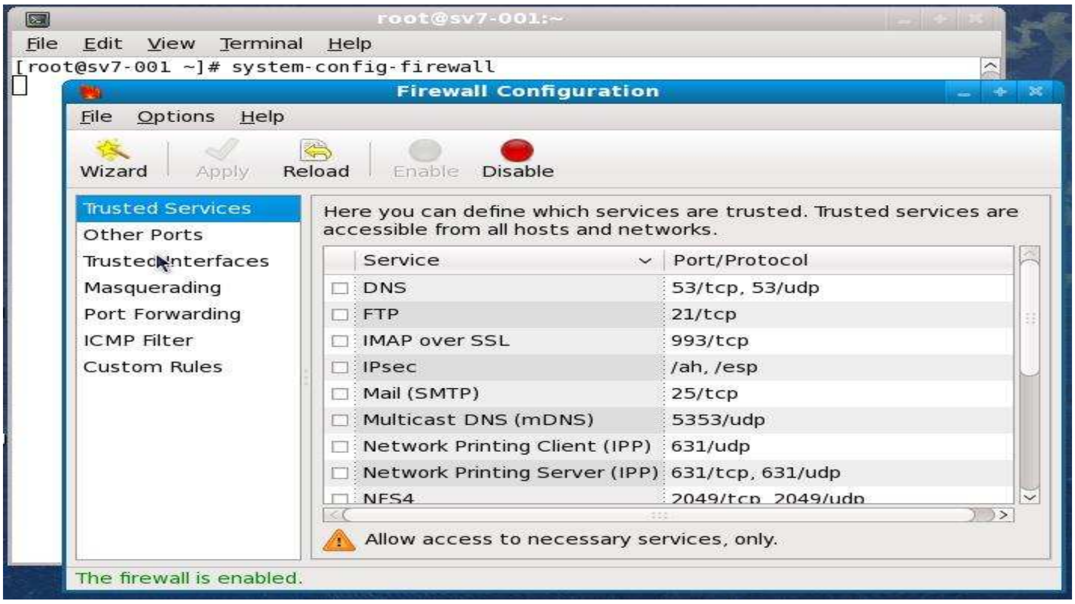
구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "*package_name*"	[그림4-8,4-9,4-10] 참고
GUI용 명령	system-config-firewall system-config-selinux	

▪ 패키지 확인



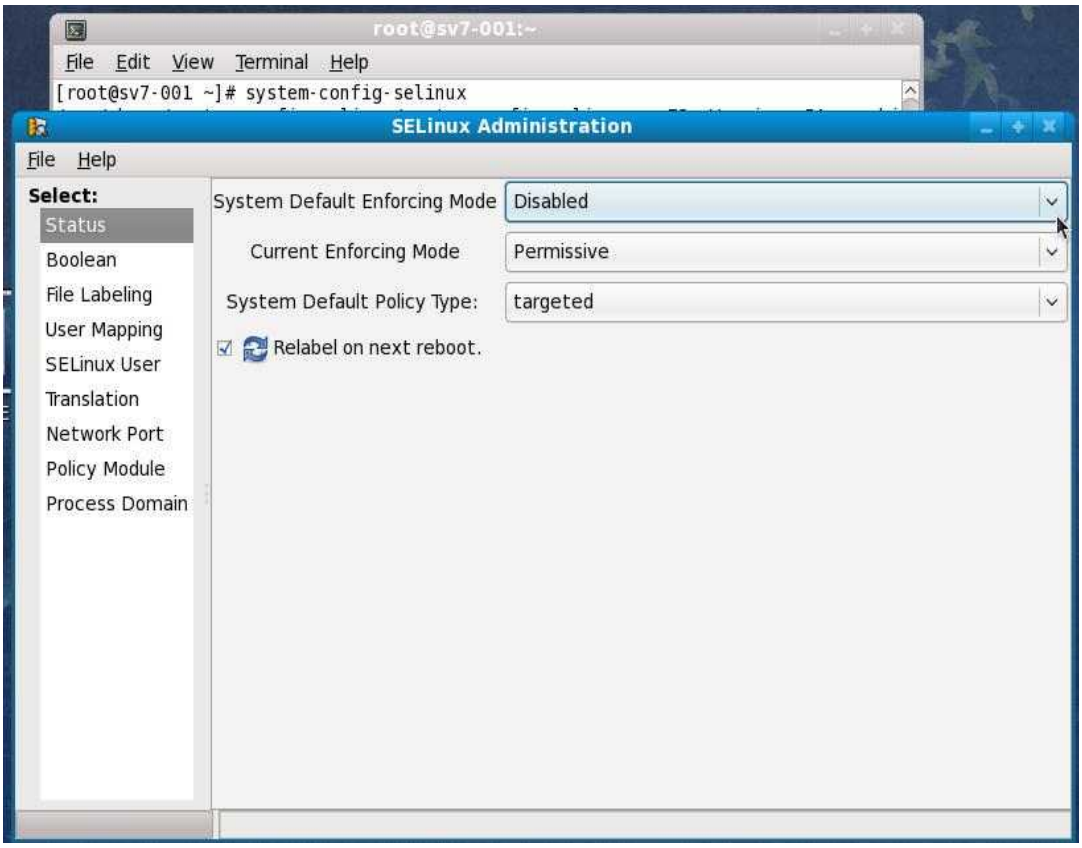
[그림 4-8. 보안 기능 지원 확인]

▪ GUI용 보안 툴 명령(방화벽)



[그림 4-9. GUI 보안 기능 지원 확인]

▪ GUI용 보안 툴 명령(SELinux)



[그림 4-10. GUI SELinux 기능 지원 확인]

☐ 자동 업데이트 기능 지원

[표 4-12. 업데이트 지원 확인]

패키지명	버전	지원여부	비 고
yum	3.2.22-4	지원	

- 업데이트 기능 확인 명령

[표 4-13. 업데이트 지원 명령 확인]

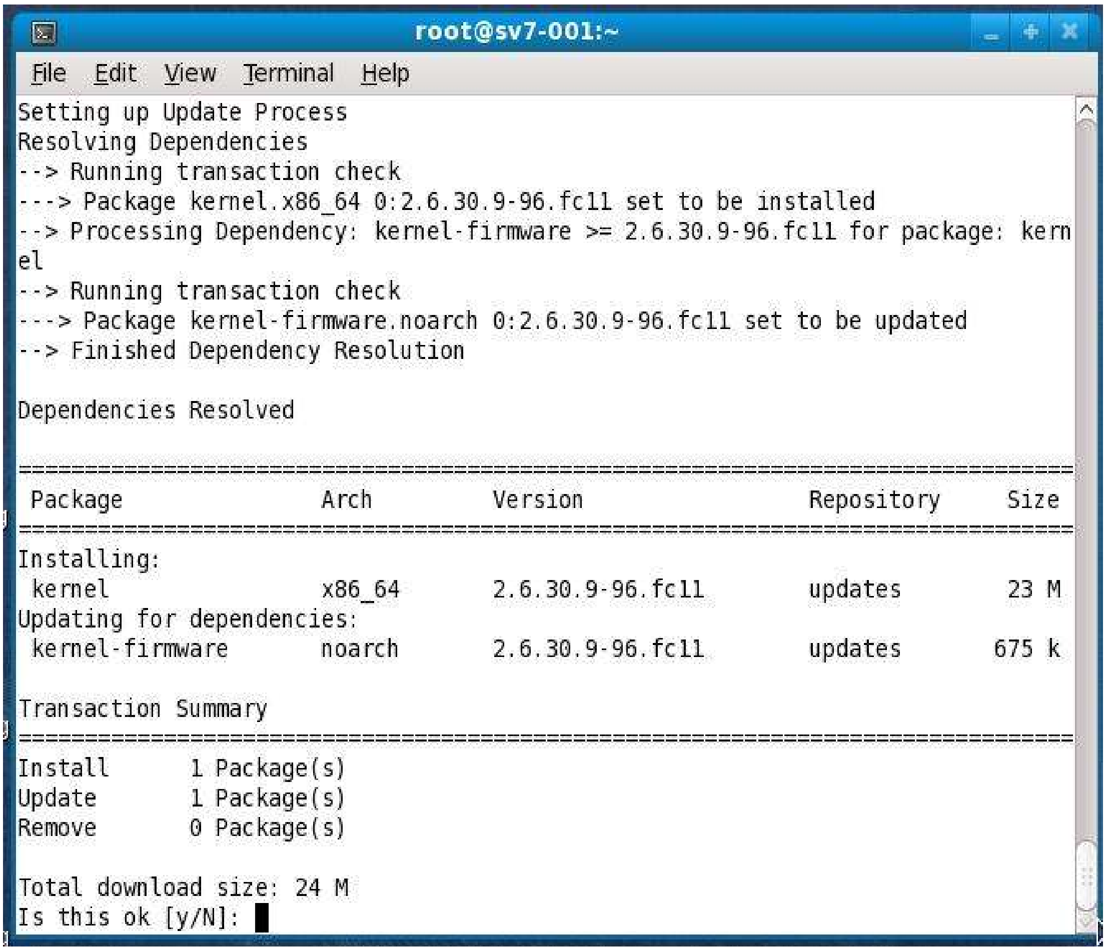
구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "*package_name*"	[그림4-11~14] 참고
GUI용 명령	gpk-application, gpk-update-viewer	

▪ 패키지 확인

```
root@localhost:~
File  Edit  View  Terminal  Help
[root@localhost ~]# rpm -qa | grep yum
yum-3.2.22-4.fc11.noarch
PackageKit-yum-plugin-0.4.9-1.fc11.x86_64
yum-utils-1.1.21-2.fc11.noarch
anaconda-yum-plugins-1.0-4.fc11.noarch
yum-metadata-parser-1.1.2-12.fc11.x86_64
PackageKit-yum-0.4.9-1.fc11.x86_64
You have new mail in /var/spool/mail/root
[root@localhost ~]#
```

[그림 4-11. 업데이트 기능 지원 확인]

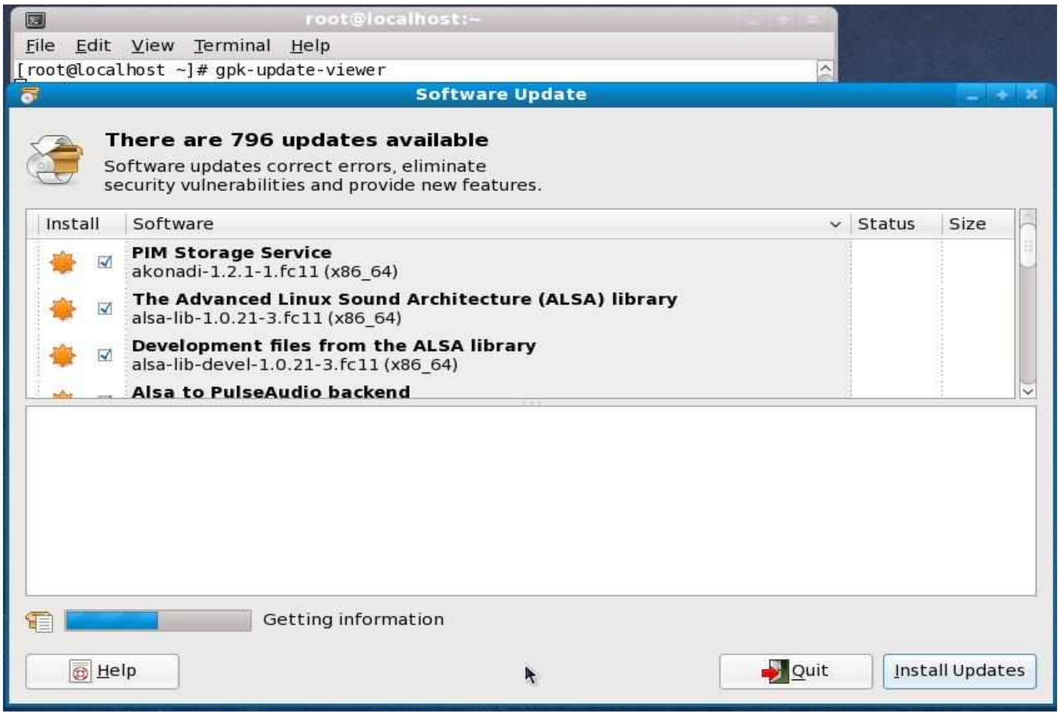
▪ 업데이트 실행 확인 (yum update)



```
root@sv7-001:~
File  Edit  View  Terminal  Help
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package kernel.x86_64 0:2.6.30.9-96.fc11 set to be installed
--> Processing Dependency: kernel-firmware >= 2.6.30.9-96.fc11 for package: kern
el
--> Running transaction check
---> Package kernel-firmware.noarch 0:2.6.30.9-96.fc11 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                Arch           Version                 Repository   Size
================================================================================
Installing:
 kernel                 x86_64         2.6.30.9-96.fc11        updates      23 M
Updating for dependencies:
 kernel-firmware        noarch         2.6.30.9-96.fc11        updates     675 k

Transaction Summary
================================================================================
Install       1 Package(s)
Update        1 Package(s)
Remove        0 Package(s)

Total download size: 24 M
Is this ok [y/N]:
```

[그림 4-12. 업데이트 기능 실행 확인]

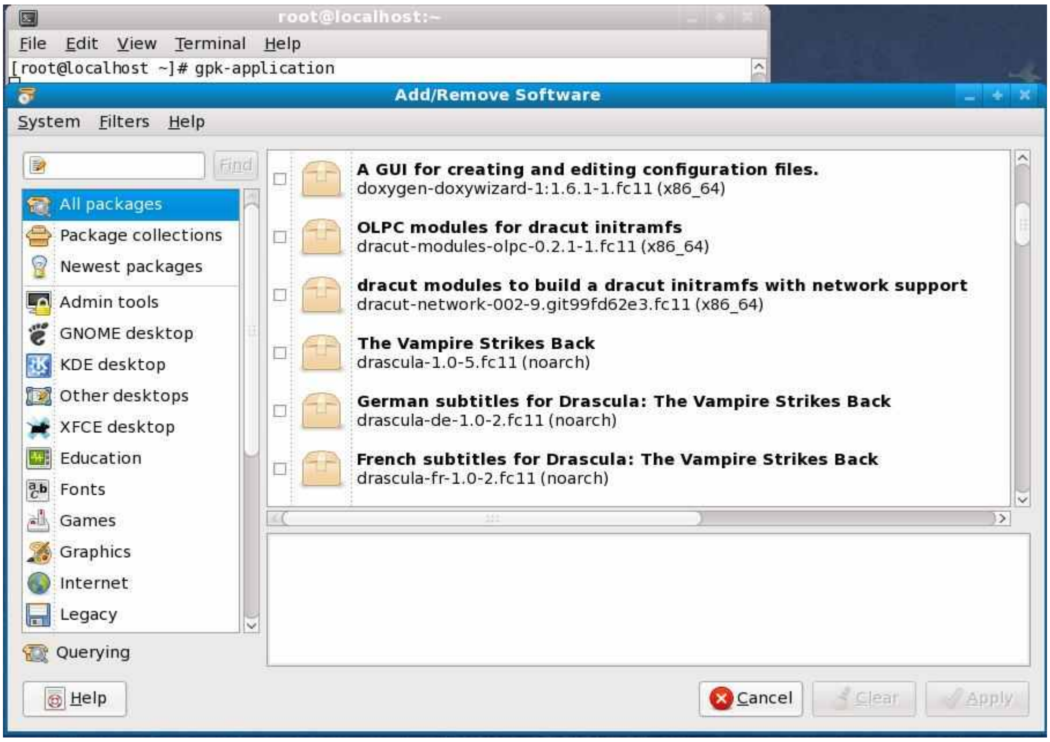
- gpk-application



[그림 4-13. GUI용 소프트웨어 설치/삭제 기능 실행 확인]

- gpk-update-viewer

GUI용 패키지 업데이트



[그림 4-14. GUI용 업데이트 기능 실행 확인]

다. 신뢰성 테스트

□ LTP Tool 사용

Linux Test Project (LTP)는 리눅스의 신뢰성, 호환성, 안정성을 입증하여 오픈소스 커뮤니티에 테스트 도구를 제공하는 것이 주 목적인 오픈소스 프로젝트이다.

- 테스트 방법

runalltests.sh 스크립트를 통한 초기 기본 테스트

- ▪ 파일시스템 스트레스 테스트
- ▪ 디스크 I/O 테스트
- ▪ 메모리 관리 스트레스 테스트
- ▪ IPC 스트레스 테스트
- ▪ 스케줄러 테스트
- ▪ 명령어 함수 확인 테스트
- ▪ 시스템 호출 함수 확인 테스트

- 실행 방법

```
$ cd /opt/ltp
$ ./runalltests.sh
```

- 테스트 결과

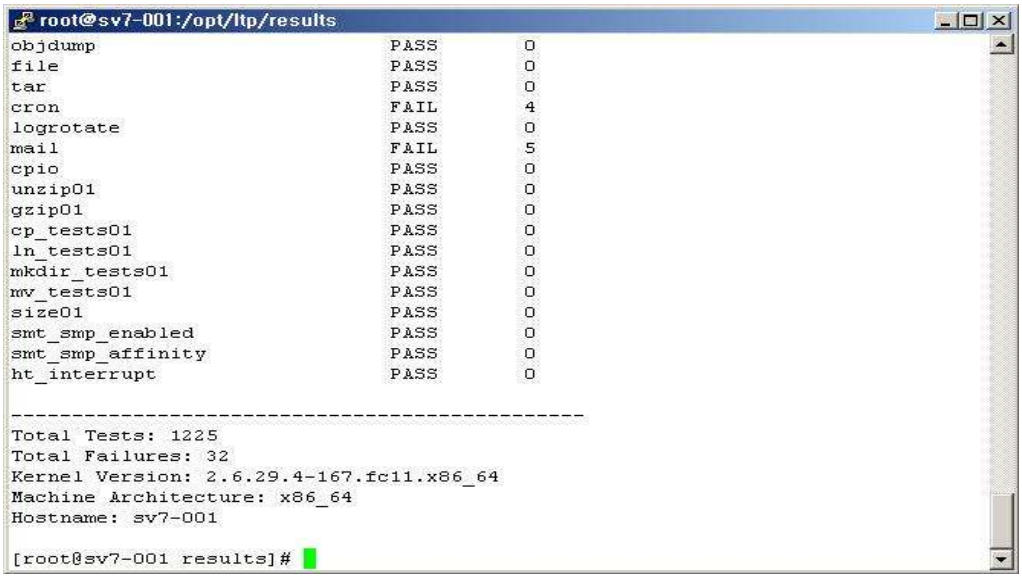
- ▪ LTP 테스트 도구 버전 : ltp-full-20090930
- ▪ 테스트 시험 결과

[표 4-14. LTP 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	1225	32

※실패 목록은 Fedora에서 발생한 문제가 아닌 기능상 구현이 안되거나, LTP 툴의 자체 버그일 가능성이 높음([표4-15] 참고)

※다른 Linux 배포판보다 실패 항목이 조금 더 높게 나타났는데, 가장 최신의 커널과 새로운 기능이 많이 포함되어 있는 것이 원인으로 파악 됨



```
root@sv7-001:/opt/ltp/results
objdump                              PASS       0
file                                 PASS       0
tar                                  PASS       0
cron                                 FAIL       4
logrotate                            PASS       0
mail                                 FAIL       5
cpio                                 PASS       0
unzip01                              PASS       0
gzip01                               PASS       0
cp_tests01                           PASS       0
ln_tests01                           PASS       0
mkdir_tests01                        PASS       0
mv_tests01                           PASS       0
size01                               PASS       0
smt_smp_enabled                      PASS       0
smt_smp_affinity                     PASS       0
ht_interrupt                         PASS       0

-----------------------------------------------
Total Tests: 1225
Total Failures: 32
Kernel Version: 2.6.29.4-167.fc11.x86_64
Machine Architecture: x86_64
Hostname: sv7-001

[root@sv7-001 results]#
```

[그림 4-15. LTP 시험 결과 화면]

▪ 테스트 시험 실패 목록

[표 4-15. LTP 시험 실패 항목]

항 목	설 명
eventfd2_03	eventfd2() 시스템 호출에 대한 test로 시스템 호출시 정상적인 리턴이 아닐 경우 FAIL
get_mempolicy01	numa 관련 시스템 호출을 핸들링하지 않는데, 이유는 overengineered(과도한 기술 중심)의 시스템 호출이기 때문이다.
mbind01	정상적인 경우 오류 유형이 EFAULT이지만 접근 가능한 주소 범위 이상으로 포인터가 설정되어있을 경우 오류 발생
msgctl11	pid의 범위를 벗어나서 실행되는 프로세스가 있을 경우 FAIL
ppoll01	ltp_signal.h에서 SIGSETSIZE 매크로를 사용하고, poll.h의 헤더 위치를 바꾸어서 테스트하면 정상적이지만 다른 테스트와의 호환을 위해서 수정하지 않고 테스트 수행, 오류 발생
quotactl01	__NR_set_tid_address 시스템 호출에 하드코딩해서 258번을 대치해야 정상 수행되지만 수정되지 않은 커널은 테스트 수행시 오류 발생
rt_sigaction01	Functionality test 방식으로 SIGRTMIN(34)에서 SIGRTMAX(64)까지 사용한다. 만약 rt_sigaction으로 1번 시그널부터 64번 시그널을 테스트하고자 한다면, 별도의 패치를 기반으로 커널 2.6.23 이후의 버전으로 테스트를 한다.
rt_sigprocmask01	rt_sigprocmask 시스템 호출에 대한 test로 real time signal test인 경우 별도의 패치를 통해 test를 수행하지만 신뢰성 약함
rt_sigsuspend01	rt_sigsuspend 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 현재 segmentation fault가 발생해서 test 항목의 신뢰성 약함

set_thread_area01	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)
set_thread_area02	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)
utimes01	EACCESS times가 NULL이고 호출자의 EUID가 파일에 접근하는 호출자와 일치하지 않은 경우 오류 발생, 패치를 적용할 경우에 오류 해결 (최신버전)
utimensat01	test 도구 안에 __NR_utimensat() 시스템 호출이 설정되지 않아서 빌드 및 링커 오류가 발생, 시스템에 영향을 주지는 않음
Containers	커널 쓰레드 생성과 관련있는 clone() 시스템 호출 관련 오류로 최신 커널 버전과 LTP 툴과의 버전 미스매치로 인해 발생
memcg_function	메모리의 자원 제어를 위한 커널에서 제공하는 기능으로 추가된지 얼마되지 않아 버그의 가능성이 있음(툴 자체의 버그 가능성)
Cap_bounds	커널의 capability 관련 기능에 대한 test로 시스템에 영향을 주지는 않으나 보안 관련 위험성이 있음
su01	passwd -e 명령이 지원되지 않은 경우 FAIL 발생, 시스템에 영향 없음
hugemmap03	hugetlb 기반으로 normal page가 high memory region에 매핑할 수 없는지를 테스트하며 비정상적인 경우 FAIL
hugemmap04	적어도 하나의 테스트용 hugepage가 필요하고 hugepage의 개수에 대한 정보가 필요한데, 유효한 hugepage가 아닐 경우 FAIL
hugeshmat01	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmat02	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmat03	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmctl01	임의의 usleep time은 부정확한 실행 순서를 야기하며 FAIL
hugeshmctl02	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패땐 FAIL
hugeshmctl03	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패땐 FAIL
hugeshmdt01	hugepage가 가용할 경우에만 hugepages를 hugeshmdt에 할당, 실패땐 FAIL
hugeshmget01	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget02	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget03	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget05	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
cron	crond 데몬에 대한 autodetect 기능을 test하는 부분으로 데몬 미실행시 발생
mail	메일 발송 정보 오류 /etc/xxx-release 파일 참조

라. 안정성 테스트

□ LTP Stress Tool 사용

LTP 테스트 도구를 사용한 스트레스 테스트를 수행한다.

- NFS 스트레스 테스트
- 메모리 관리 스트레스 테스트
- 파일시스템 스트레스 테스트
- Math (floating point) 테스트
- pthread 스트레스 테스트
- 디스크 I/O 테스트
- IPC (pipeio, semaphore) 테스트
- 시스템 호출 함수 확인 테스트
- 네트워킹 스트레스 테스트

- 실행 방법

```
$ cd /opt/ltp/testscripts
$  ./ltpstress.sh -d /tmp/sardata -l /tmp/ltplog.1122 -m 1024 -t 24 -S
```

- 테스트 결과

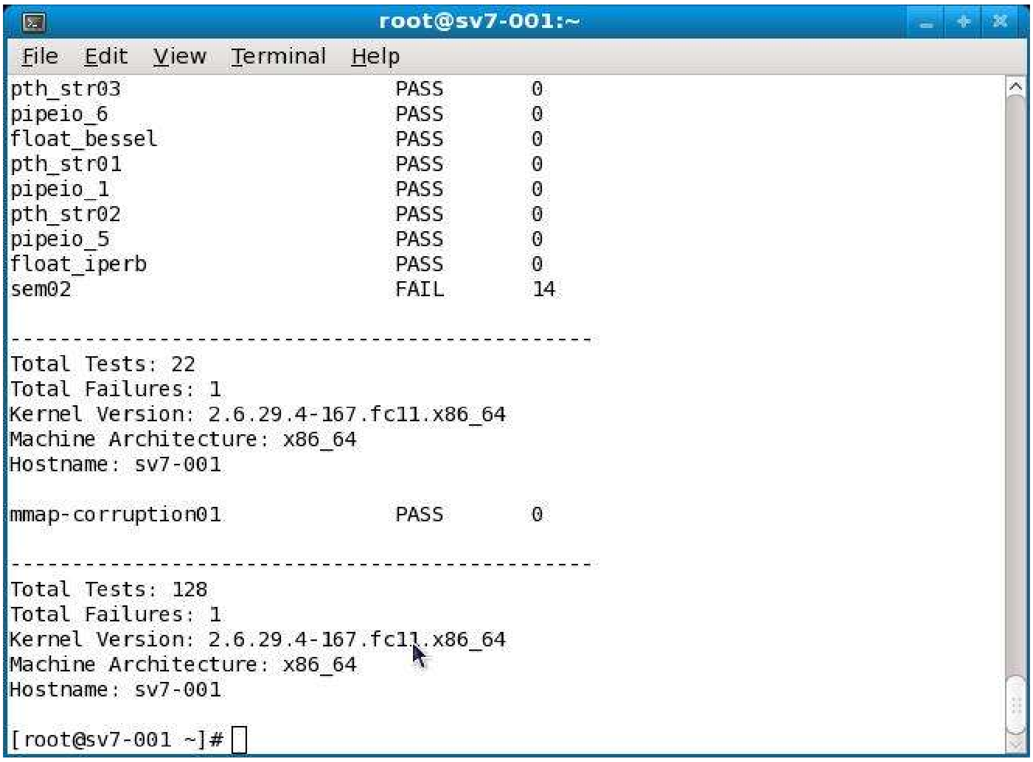
- LTP 테스트 도구 버전 : ltp-full-20090930
- 테스트 시험 결과

[표 4-16. LTP Stress 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	22	1
	128	1

※ 실패 목록은 Fedora에서 발생한 문제가 아닌 기능상 구현이 안 되어 발생한 오류일 가능성이 높음([표 4-17] 참고)

※ 기본 테스트인 runalltests.sh의 결과와 달리 다른 배포판과 동일한 성공률을 보였음



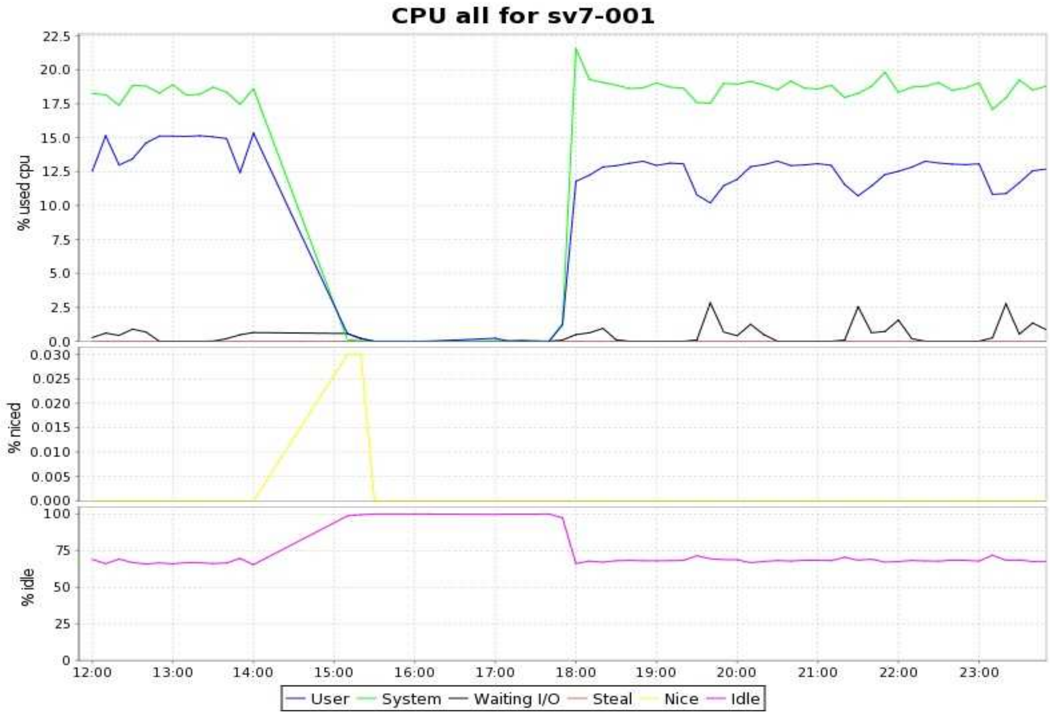
[그림 4-16. LTP Stress 시험 결과 화면]

▪ 테스트 시험 실패 목록

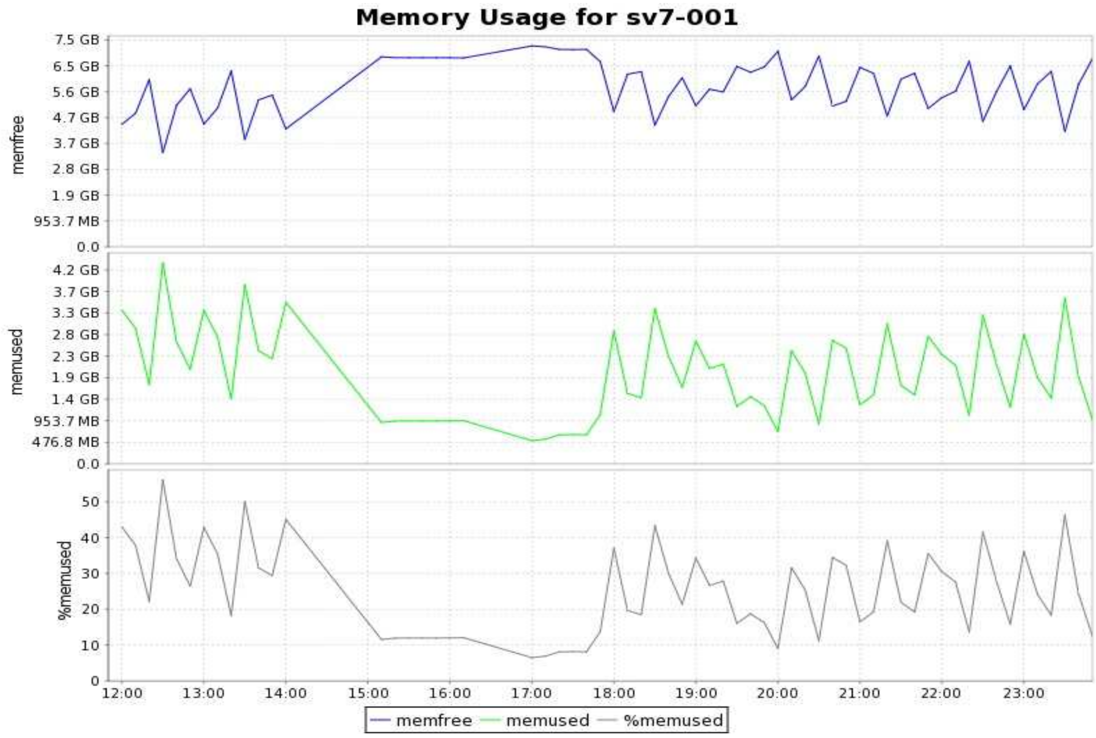
[표 4-17. LTP Stress 실패 항목]

항 목	설 명
mc_cmds	ltp network test일 경우 /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts 값이 1로 설정된 경우 FAIL 출력
nfsx-linux	nfsx-linux를 찾지 못해서 발생하는 오류로 nfs v3과 v4에서 버전 미스매치로 인해 발생하는 문제로 시스템에 영향 없음
sem02	semaphore lock을 테스트하는 부분으로 세마포어 획득 및 해제 관련 기능 테스트로 FAIL시 시스템 영향 적음

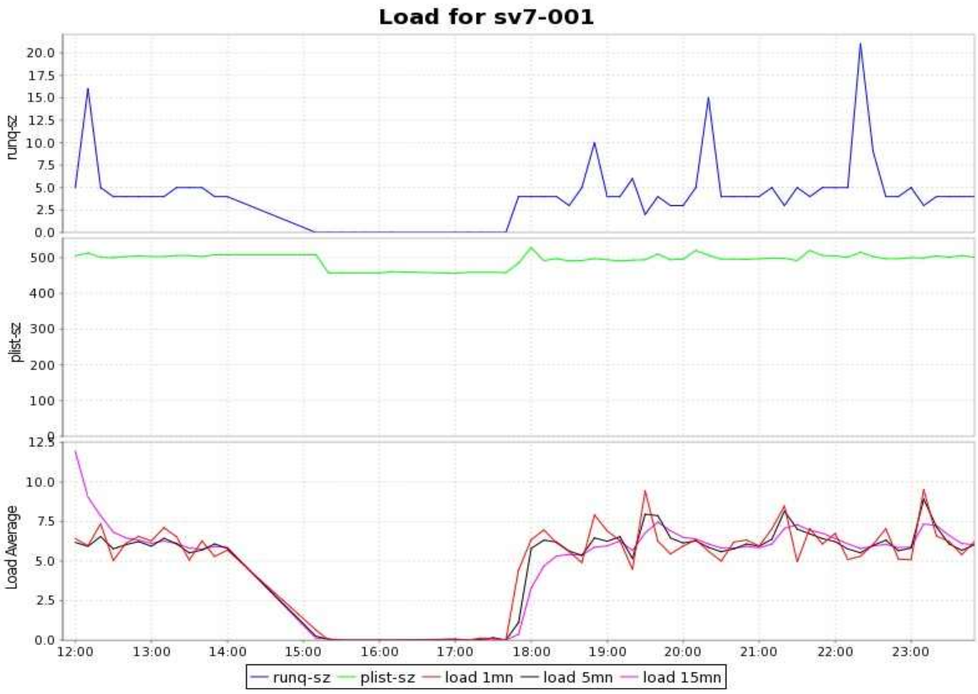
▪ 시스템 사용률



[그림 4-17. CPU 사용률 확인]



[그림 4-18. Memory 사용률 확인]



[그림 4-19. 평균 사용률 확인]

[표 4-18. 시스템 사용량 요약]

구분	최대 사용량	평균 사용량
CPU	100%	18%
Memory	8GB	2.3GB
Load Average	16	7

※ 참고 사항

- [그림4-17~19] 그래프는 하루를 기준으로 작성되었다.
- 스트레스 테스트 그래프는 17시 50분부터 ~ 23시 59분까지 해당 된다.
- 수치가 낮을수록 시스템 부하가 적은 것이다.

5. 결론

□ 설치

최신의 하드웨어 테스트 환경에서 드라이버가 기본으로 제공되므로 설치 시 문제가 발생할 가능성이 적으며, Global에서 많이 알려진 Linux 운영체제로 설치 및 사용 중 발생하는 문제를 커뮤니티 및 프로젝트 홈페이지에서 확인할 수 있다.

□ 기능

새로운 기술이 가장 많이 적용되어 배포되는 제품으로 Red Hat Enterprise Linux의 모태가 되며, 앞으로 발전 가능성이 높은 신기술을 미리 체험할 수 있는 장점이 있다.

□ 신뢰성

LTP 툴에서 보여주듯 1000가지가 넘는 기능 테스트 중 대부분이 문제없이 테스트 되었으며, 일부 테스트의 경우 기능상 구현이 안 되거나 지원하지 않는 부분의 오류이므로 운영체제에 문제가 있다고 판단하기는 어려우며 다른 Linux 배포판과 비교하여도 크게 문제될 부분은 없다.

□ 안정성

스트레스 테스트 결과에서 시스템이 문제가 발생할 가능성은 적어 보였지만, 테스트 중 시스템이 느려지는 증상이 다른 배포판에 비하여 높게 나타났다.

Fedora 리눅스 운영체제는 테스트를 진행하면서 최신기술의 평가 항목 및 많은 어플리케이션이 포함되어 있어 테스트 수행을 위한 기본 환경 설정에 있어서 다소 어려운 점이 있었다.

공공기관 및 기업, 그리고 교육기관 등에 서비스 제품으로 도입이 되더라도 크게 문제될 내용은 없을 것으로 판단된다.

다만, 많은 어플리케이션이 포함되어 있어 각각의 구축 서비스에 맞는 최적화 작업이 필요하며, 책임 있는 기술지원 서비스를 받을 수 있는 체계가 구축된다면 엔터프라이즈 환경에서 서비스를 할 수 있는 운영체제가 될 것으로 보인다.

* 참고 자료

http://www.fedoraproject.org/

http://ko.wikipedia.org/wiki/Fedora

http://ltp.sourceforge.net/

* 별첨

별첨1. LTP 활용 가이드

Ⅳ FOSSology

목 차

1. 개요

Hewlett Packard Company (HP)에서 시작한 FOSSology 프로젝트는 오픈소스 프로젝트에 라이선스를 주는 방법에 대해서 빠르고 정확하게 설명하는 툴이다.

본 테스트에 사용될 라이선스 검증 툴인 FOSSology에 대하여 다음과 같은 특성을 열거한다.

□ HP 에서 개발한 오픈 소프트웨어이다

□ GPL로 개발
소스의 다운에서 부터 이용까지 모든 것을 무료로 이용할 수 있다.

□ 가벼운 설치 파일
설치 파일은 10MB면 충분하며, 분석대상인 총 프로젝트 크기에 따라 필요한 하드 용량의 크기가 달라진다.

□ 주석 부분에 의한 분석
코드 매치가 아닌, 주석 부분에 의한 분석이 이루어짐으로, 주석 부분에 라이선스와 관련된 내용이 없는 파일들은 오픈소스로 분석이 되지 않는다.

□ 356개의 기본적인 라이선스 정보만 제공

□ 제품의 버전과 라이선스에 대한 업데이트는 이전 버전을 모두 삭제 후 새로운 버전을 설치해야 한다.

□ 각 기술적인 부분은 Fossology Community에서 지원 받는다.

2. 테스트 환경

테스트 대상 Application	FOSSology_sample.c
테스트 툴	FOSSology-1.1.0-1-e15

<FOSSology_sample.c>

```
/*FOSSology_sample
* Copyright(C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.
*
*This File is part of GnuPG
*
*GnuPG is free software; you can redistribute it and/or modify
*it under the terms of the GNU General Public License as published by
*the Free Software Foundation; either version 2 of the License, or
*(at your option) any later version.
*
*GnuPG is distributed in the hope that it will be useful,
*but WITHOUT ANY WARRANTY; without even the implied warranty of
*MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*GNU General Public License for more details.
*
*You should have received a copy of the GNU General Public License
*along with this program; if not, write to the Free Software
*Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

#include "util.h"
#include "mpi.h"
#include "i18n.h"

#define STACKSIZE 100
static MPI stack[STACKSIZE];
static int stackidx;
```

```
const char *
strusage(int level)
{
const char *p;
swich(level){
case 10:
case 0: p = "mpicalc - v" VERSION ";"
      "Copyright 2009 Werner Koch (dd9jn)" ; break;
case 13: p = "mpicalc"; break;
case 14: p = "VERSION"; break;
case 1:
case 11: p = "Usage: "mpicalc (-h for help)";
      break;
case 2:
case 12: p = "\nSyntax: mpicalc [options][files]\n" "MPI RPN calculator\n";
      break;
default: p = default_strusage(level);
}
return p;
}

static void
i18n_init(void)
{
#ifdef ENABLE_NLS
#ifdef HAVE_LC_MESSAGES
setlocale(LC_MESSAGES, "");
#else
setiocale(LC_ALL, "");
#endif
bindtextdomain(PACKAGE, G10_LOCALEDIR);
textdomain(PACKAGE);
#endif
}
```

3. 테스트 절차

가. 소프트웨어 설치

1) Installing FOSSology Dependencies

(1) Libextractor

메타데이터에 사용되는 라이브러리이다.

다운로드 장소

```
ftp://ftp.suse.com/pub/people/ke/9.3-i386/libextractor-0.5.2-1.1.i586.rpm
ftp://ftp.suse.com/pub/people/ke/9.3-i386/libextractor-devel-0.5.2-1.1.i586.rpm
```

(2) Sleuthkit

파일시스템 데이터를 조사하도록 하는 C 자료와 명령라인 툴 컬렉션이다.

다운로드 장소

```
ftp://ftp.pbone.net/mirror/www.spenneberg.org/Forensics/sleuthkit/sleuthkit-2.0
1-oss_fc2_1.i386.rpm
```

(3) ucl

ANSI C에서 작성된 휴대용 데이터 압축 라이브러리이다.

다운로드 장소

```
ftp://ftp.univie.ac.at/systems/linux/dag/redhat/el5/en/i386/RPMS.dag/ucl-1.03-
2.el5.rf.i386.rpm
```

(4) upx

다른 실행가능 포맷에서도 실행 가능한 packer(압축, 압축해제 알고리즘)이다.

다운로드 장소

```
ftp://ftp.univie.ac.at/systems/linux/dag/redhat/el5/en/i386/RPMS.dag/upx-3.03
-1.el5.rf.i386.rpm
```

(5) unrar

파일 압축 알고리즘이다.

다운로드 장소

```
ftp://ftp.pbone.net/mirror/atrpms.net/sl5-i386/atrpms/stable/unrar-3.8.4-1.el5.i38
6.rpm
```

(6) cabextract

Microsoft cabinet 파일을 추출하기 위한 무료 소프트웨어이며 UNIX나 UNIX와 같은 시스템에서는 .CAB files이라고도 불린다.

다운로드 장소

```
ftp://ftp.univie.ac.at/systems/linux/dag/redhat/el5/en/i386/dag/RPMS/cabextra
ct-1.2-1.el5.rf.i386.rpm
```

2) Installing FOSSology from the RPM

(1) 다운로드

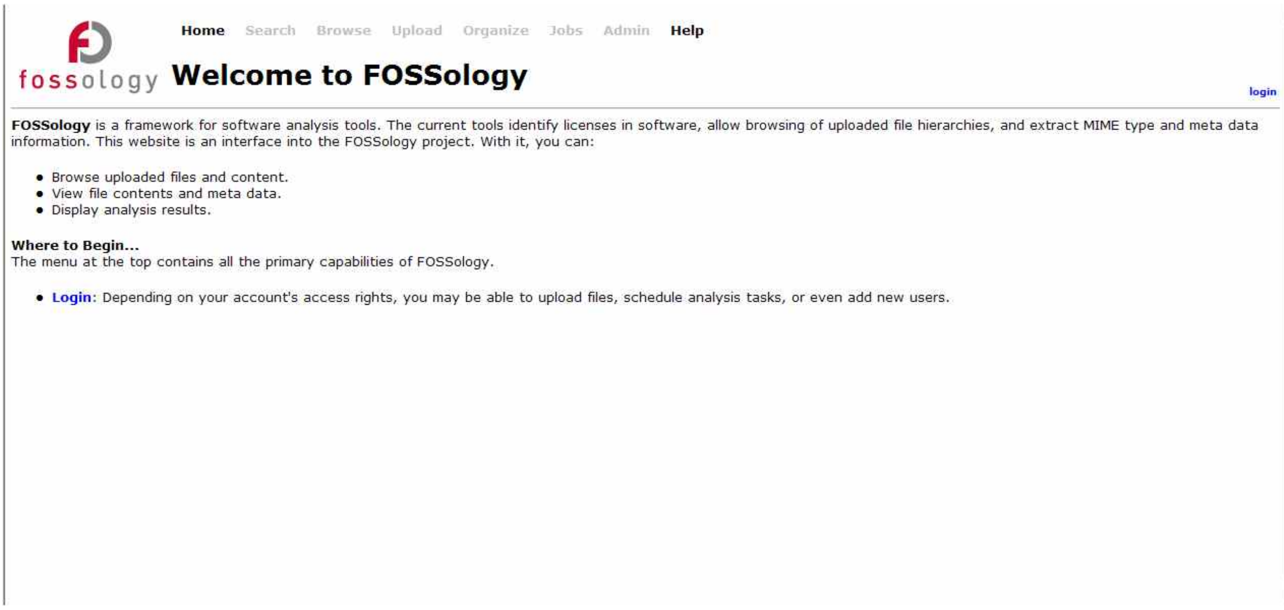
```
https://sourceforge.net/projects/fossology/files/
fossology-1.1.0-1-el5.i386.rpm and fossology-1.1.0-1-el5.x86_64.rpm
```

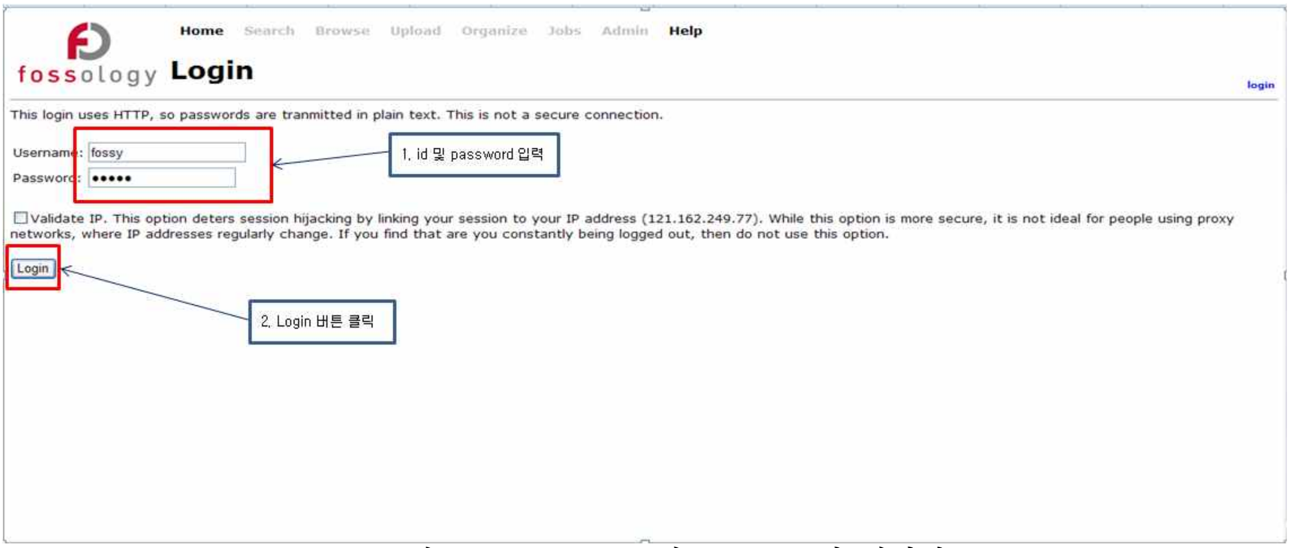
(2) 설치

```
$ sudo rpm -ivh fossology-1.1.0-1-el5.i386.rpm
```

나. 테스트 실시

1) FOSSology UI 페이지를 표시하여 로그인한다.

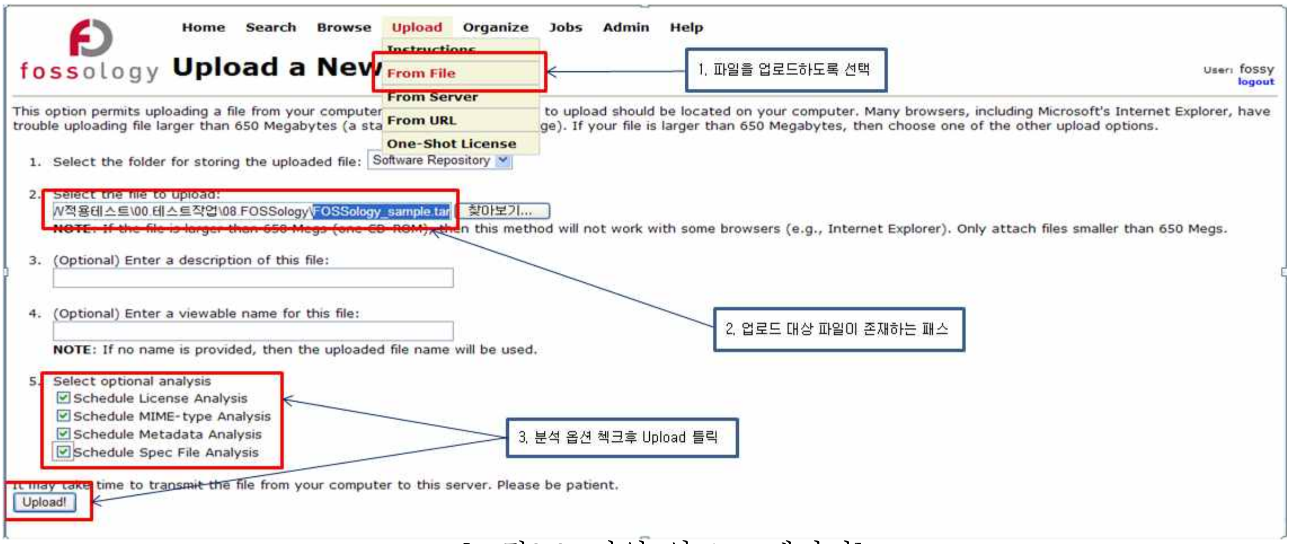




[그림3-1. FOSSology 테스트 로그인 페이지]

2) 라이선스 검증을 위한 파일을 업로드 한다.

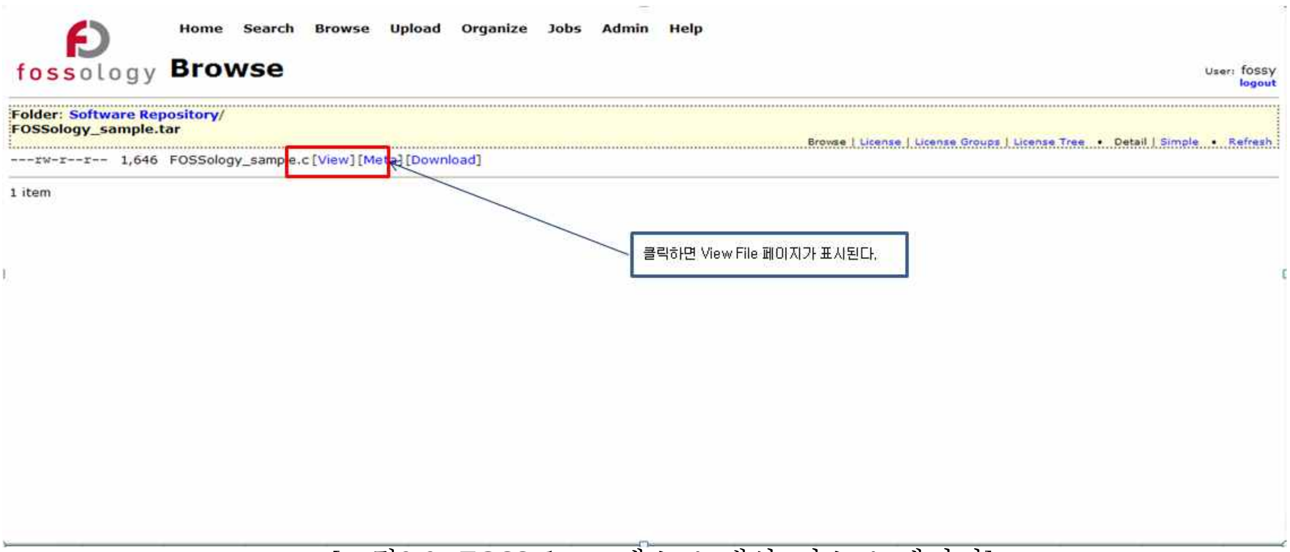
업로드하면 스케줄에 따라 라이선스 검증이 시작된다.



[그림3-2. 파일 업로드 페이지]

4. 테스트 결과

1) Browse 페이지 - 라이선스 검증 테스트 대상 리스트 표시

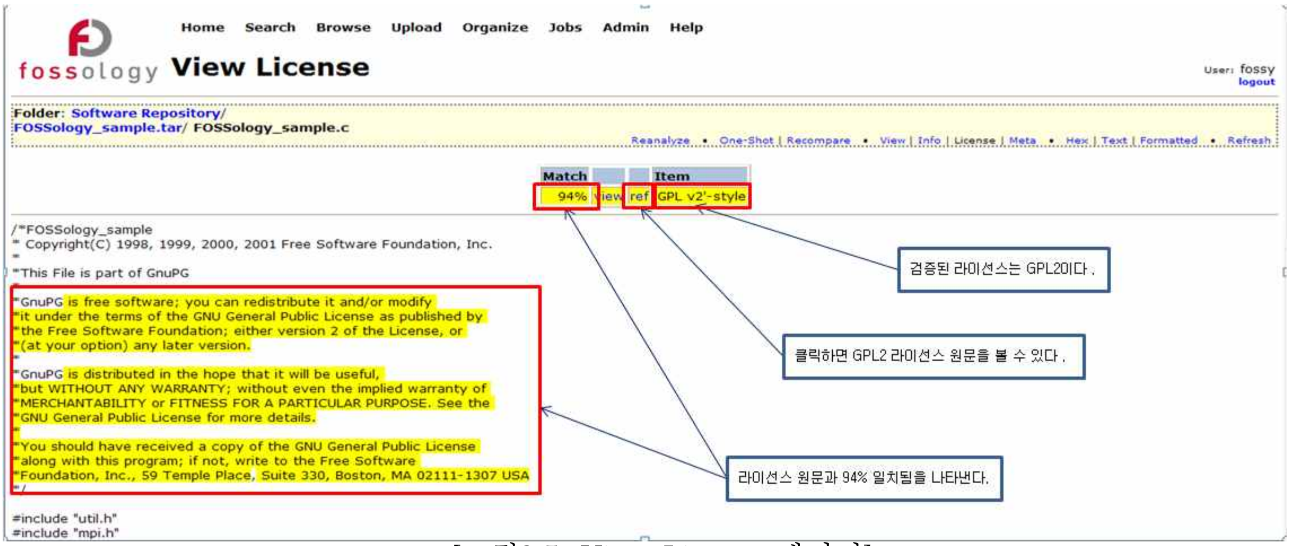


[그림3-3. FOSSology 테스트 대상 리스트 페이지]

2) View File 페이지 - 라이선스 검증 테스트 대상 소스 표시(1)

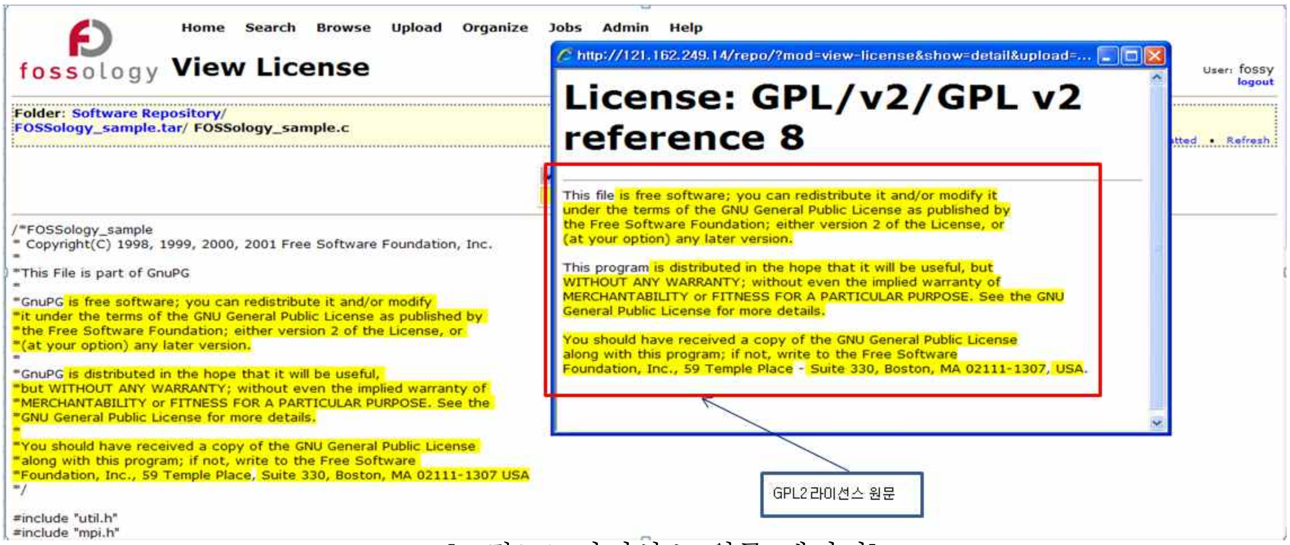
[그림3-4. View File 페이지]

3) View License 페이지 - 라이선스 검증 테스트 대상 소스 표시(2)



[그림3-5. View License 페이지]

4) 라이선스 원문 페이지



[그림3-6. 라이선스 원문 페이지]

5. 종합

FOSSology 툴을 사용한 분석에 의하여 소프트웨어 패키지 내에서 사용하고 있는 소프트웨어 라이선스 정보의 리포트를 얻을 수 있다.

라이선스 정보를 추출하는 것 이외에도 메타 데이터의 전개 및 MIME 타입을 식별하는 기능도 이용할 수 있다.

그러나 FOSSology는 코드 매치가 아닌, 주석 부분에 의한 분석이 이루어짐으로, 주석 부분에 라이선스와 관련된 내용이 없는 파일들은 오픈소스로 분석이 되지 않는다.

즉, **파일에 라이선스 원문이 포함되지 않는다면, 일반적인 주석과 코드들은 분석 대상에서 제외되어 해당 파일을 검증할 수 없게 된다**.

의도적으로 주석 부분에 라이선스를 반영/삭제함에 의하여 본질에 위배되는 라이선스 정책을 악용하는 사례가 발생할 수 있다.

Ⅴ GINUX

목 차

1. 개요

본 문서는 공개SW 역량프라자 운영 사업의 공개SW 테스트 환경 구축을 통하여 테스트한 공개SW에 대한 기능, 성능, 호환성 등을 검증하고, 향후 공공 기관 및 기업에서 도입 시 참고할 수 있는 자료로 활용하기 위하여 작성되었다.

가. 목적

□ 공개SW를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 공개SW의 기능, 성능, 안정성, 호환성 등 검증
- 공개SW 기술의 도입 가이드라인 제공
- 효과적인 공개SW 도입 유도
- 안정적인 공개SW 시스템 도입

나. 선정 기준

□ 공개SW 테스트 대상은 다음 5가지 사항을 고려하여 선정되었다.

- 1번 : 국내에서 개발된 제품
- 2번 : 국내 기업에서 많이 도입되는 제품
- 3번 : Global에서 많이 알려진 제품
- 4번 : 기술지원(업데이트) 가능한 제품
- 5번 : OS간 호환 가능 제품(Package 관리 방식)

[표 1-1. 선정기준]

Linux 운영체제(OS)	1번	2번	3번	4번	5번	결과	비고
Asianux	●	◕	◕	●	●	●	선정*
CentOS	○	◑	●	●	●	◕	선정
Debian	○	◔	◕	●	◔	◑	보류
Fedora	○	◑	●	●	●	◕	선정
Gentoo	○	◔	◕	◕	◔	◑	보류
GINUX	●	◑	◔	●	●	◕	선정
Red Hat Enterprise Linux	○	●	●	●	●	●	선정*
Slackware	○	◔	◕	◕	◔	◑	보류
SULinux	●	◑	◔	●	●	◕	선정
SuSE Linux	○	◑	●	●	●	◕	선정*
Ubuntu	○	◑	●	●	◔	◑	보류

※ 테스트 대상 기준에 근거하여 이번 테스트 대상에서 제외된 **선정*** 표시 제품은 금번 사업의 일정상의 제약으로 다음 사업에서 테스트를 수행하기로 함

2. 테스트 환경 구성

가. 테스트 일정

[표 2-1. 일정표]

구분	월	10				11				12			
	주	1	2	3	4	1	2	3	4	1	2	3	4
테스트 대상 발굴 및 선정													
테스트 환경 구성													
테스트 항목 선정													
테스트 시나리오 작성													
테스트 수행													

나. 테스트 대상 소개

☐ GINUX 2.2

GINUX는 SK C&C 가 엔터프라이즈 서버용으로 개발한 리눅스 운영체제로 기업 비즈니스 환경에서 필요한 하드웨어의 성능 및 기존 소프트웨어와의 호환성과 범용성에 초점을 두어 개발되었다.

☐ GINUX 중요 특징

- 성능개선
  - 개선된 프로세싱 파워
  - 대용량 볼륨 활용
  - 32/64비트 동시지원 플랫폼
- 가상 스토리지
  - Distributed Parallel Fault Tolerant File System
  - Locking Mechanism
  - 대용량/고성능 파일처리
  - 디스크 효율성 극대화
- 관리 효율성
  - 최신 패키지 유지
  - 손쉬운 패키지 업데이트
  - Bug Tracking System(BTS)
  - 효율적인 시스템 관리
- 시스템 보안
  - 강력한 암호화 및 인증 서비스
  - 한정적인 시스템 허가(SELinux)

- ▪ 표준 보안 프로토콜
- 서비스 이중화
  - ▪ 정교한 서비스 자원 모니터링
  - ▪ 손쉽고 편리한 서비스 관리
  - ▪ 멀티노드 클러스터링
  - ▪ 최적 리소스 선점 기능
  - ▪ 서비스 운용 시간 최대화
  - ▪ 서비스 유지 수준 향상
- 신뢰성과 안정성
  - ▪ 안정적인 H/W 추상화, 쓰레드 제어
  - ▪ 정교한 멀티 쓰레딩
  - ▪ 멀티 코어 최적화
  - ▪ 우수한 장애처리
  - ▪ 디스크 효율성 극대화
  - ▪ 고성능 & 안정적인 데이터 처리
  - ▪ 우수한 서버, 네트워크, 데이터 보호
- 호환성
  - ▪ IHV / ISV 인증을 통한 HW/SW 호환성 확보
  - ▪ LSB, LTP 를 통한 표준 준수, OS 기능 및 성능 검증
  - ▪ 표준 UNIX 적합성

다. 테스트 환경

□ 테스트 서버

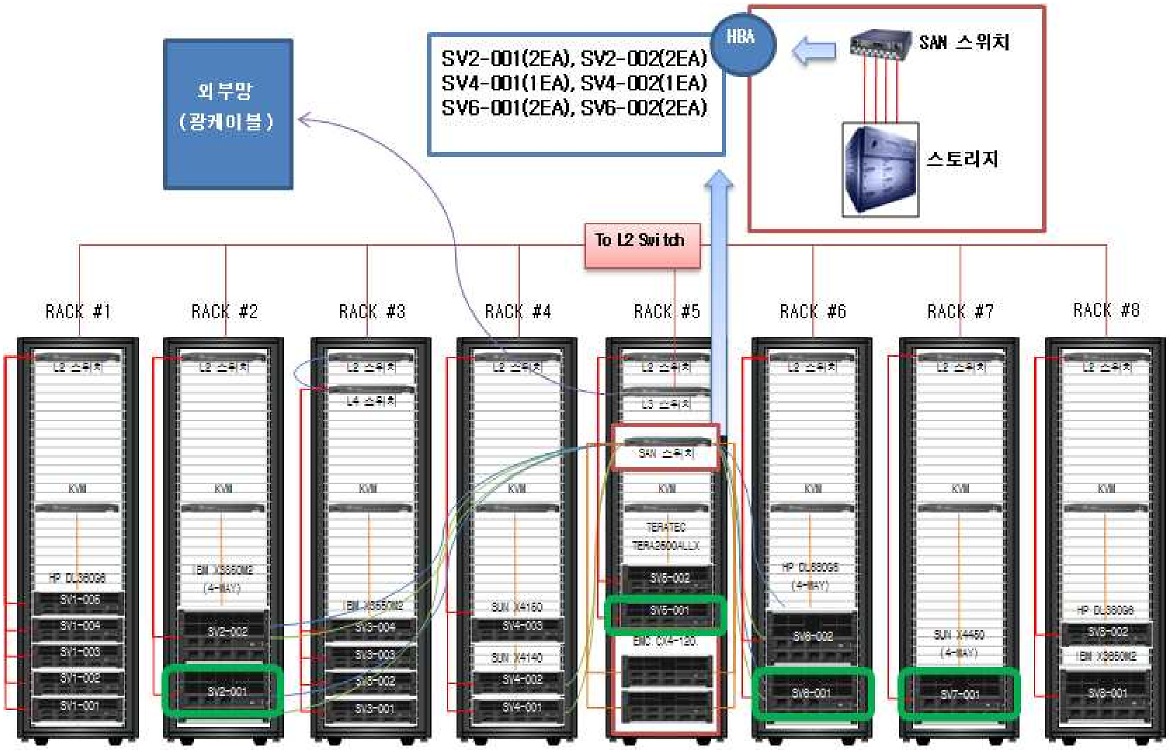
[표 2-2. 테스트 서버 규격]

제조사	모델명	CPU	MEM	Disk	NIC
HP	DL580 G5	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
IBM	X3850 M2	Quad-Core 2.4Ghz 4-Way	8GB	146GB*4	Gigabit 2Port
SUN	X4450	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
TERATEC	TERA2500ALLX	Quad-Core 2.5Ghz 2-Way	8GB	147GB*5	Gigabit 2Port

□ 테스트 조건

- 32bit 환경에서 64bit환경으로 서버 구성이 전환됨에 따라 64bit 환경에서의 테스트를 수행
- 성능 테스트는 IBM X3850 M2 서버에서 수행

□ 테스트 실장도



☞ 박스 안의 서버가 OS 테스트용 서버로 사용 됨.

[그림 2-1. 테스트 서버 실장도]

□ 서버 위치

[표 2-3. 서버 위치]

제조사	모델명	Rack 위치	비 고
IBM	X3850 M2	#2	
TERATEC	TERA2500ALLX	#5	
HP	DL580 G5	#6	
SUN	X4450	#7	

3. 테스트 시나리오

가. 설치 및 호환성 검증

□ DVD-ROM을 통한 설치

□ 하드웨어 정보 확인

□ 4대의 하드웨어 벤더별 설치 상태 확인

나. 기능

□ 네트워크 서비스 지원

- Web, Ftp, NFS, Samba, Openssh, Telnet

□ 파일시스템 지원

- ext3, XFS, GFS 등

□ LVM 지원 확인

□ 보안 기능 지원

- tcp-wrapper
- GUI용 보안 툴

□ 자동 업데이트 기능 지원

- On-Line 업데이트

다. 신뢰성

□ LTP Tool 사용한 테스트

- ltp-full-20090930 버전 사용

라. 안정성

□ LTP Stress Tool 사용

- CPU, Memory 등의 사용량 체크

4. 테스트

가. 설치 및 호환성 검증

□ GINUX 설치 환경

- GINUX 설치 버전

[표 4-1. OS 설치 정보]

OS 버전	지원 유형	Kernel 버전
GINUX 2.2	X86_64(EM64T, AMD64)	kernel-2.6.18-92.GX2

- 파티션 정보

[표 4-2. 파티션 정보]

<table>
<tr><th>디스크
(RAID)</th><th>Physical
Volume</th><th>Volume
Group</th><th>Logical
Volume</th><th>Mount
Point</th><th>용량
(MB)</th></tr>
<tr><td rowspan="9">/dev/sda(580GB)
RAID 5</td><td>/dev/sda1</td><td>-</td><td>-</td><td>/boot</td><td>100</td></tr>
<tr><td rowspan="5">/dev/sda2</td><td rowspan="5">VG0</td><td>lv_root</td><td>/</td><td>20480</td></tr>
<tr><td>lv_swap</td><td>-</td><td>8192</td></tr>
<tr><td>lv_usr</td><td>/usr</td><td>30720</td></tr>
<tr><td>lv_home</td><td>/home</td><td>20480</td></tr>
<tr><td>lv_var</td><td>/var</td><td>10240</td></tr>
<tr><td rowspan="3">/dev/sda3</td><td rowspan="3">VG1</td><td>lv_data1</td><td>/data1</td><td>251200</td></tr>
<tr><td>lv_data2</td><td>/data2</td><td>51200</td></tr>
<tr><td>lv_data3</td><td>/data3</td><td>51200</td></tr>
</table>

※ 참고사항

디스크 용량은 서버별 차이가 있으므로 위의 기준으로 백분율(%) 환산 적용 하였으며 남은 용량은 여유 공간으로 비워 둠

□ 서버별 설치 및 호환성 검증

[표 4-3. OS 설치 방법 및 결과]

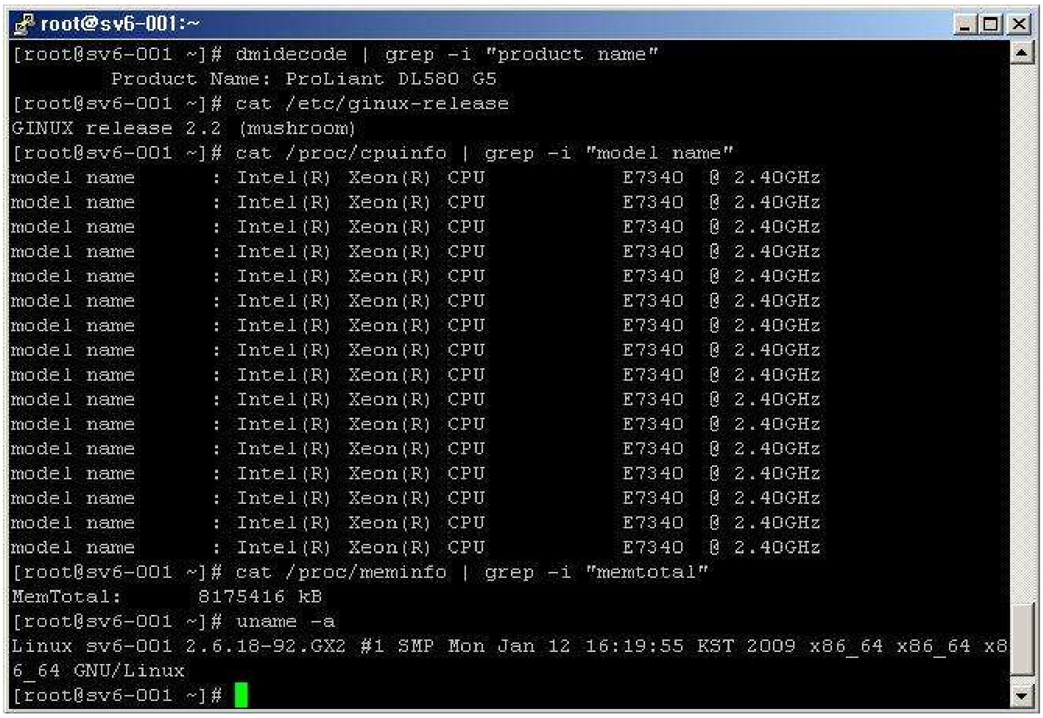
구분	설치 매체	설치 방식	패키지 선택 방식	설치 결과
HP DL580 G5	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
IBM X3850 M2	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
SUN X4450	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
TERATEC TERA2500ALLX	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상

- 서버별 Ginux 설치 상태 확인 명령

[표 4-4. GINUX 설치 상태 확인]

구분	명령어	결과
서버 모델명 확인	dmidecode | grep "Product Name"	[그림4-1~4] 참고
OS 버전 확인	cat /etc/ginux-release	
CPU 정보	cat /proc/cpuinfo | grep -i "model name"	
Memory 정보	cat /proc/meminfo | grep -i "memtotal"	
Kernel 정보	uname -a	

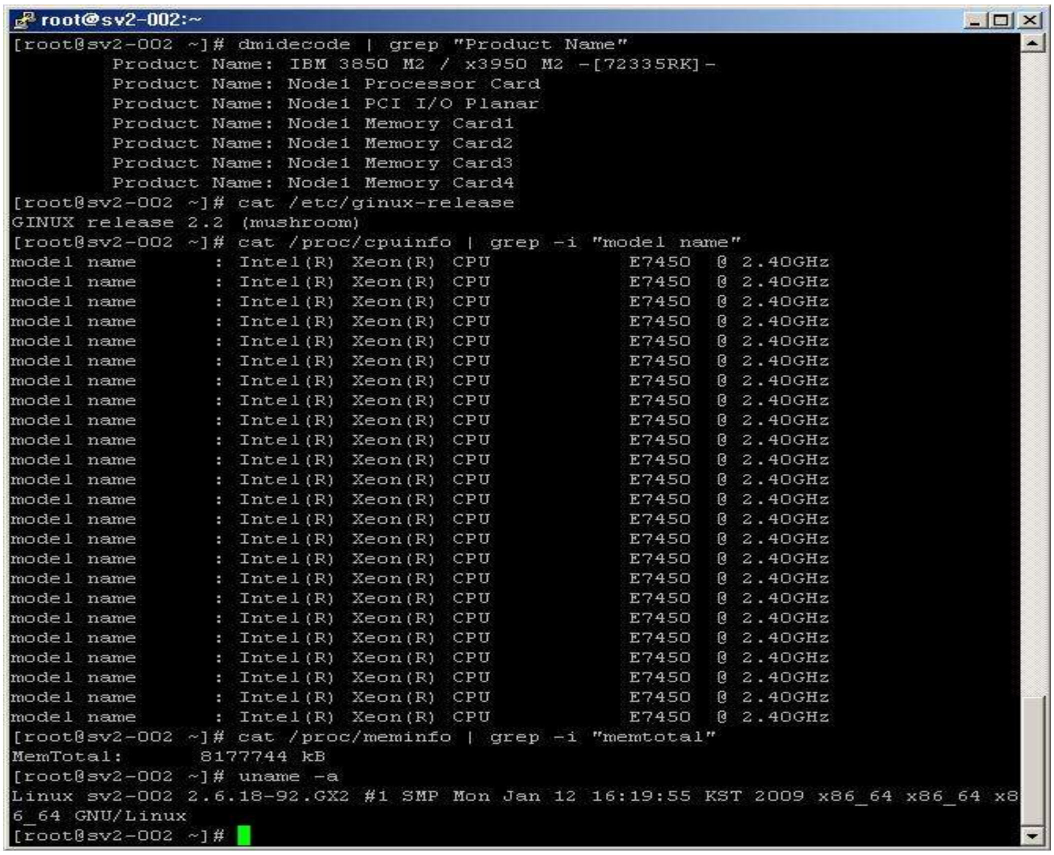
▪ HP DL580 G5 모델에서 Ginux 설치 확인



```
root@sv6-001:~
[root@sv6-001 ~]# dmidecode | grep -i "product name"
        Product Name: ProLiant DL580 G5
[root@sv6-001 ~]# cat /etc/ginux-release
GINUX release 2.2 (mushroom)
[root@sv6-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
[root@sv6-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:       8175416 kB
[root@sv6-001 ~]# uname -a
Linux sv6-001 2.6.18-92.GX2 #1 SMP Mon Jan 12 16:19:55 KST 2009 x86_64 x86_64 x8
6_64 GNU/Linux
[root@sv6-001 ~]#
```

[그림 4-1. HP DL580G5에서 GINUX 상태 확인]

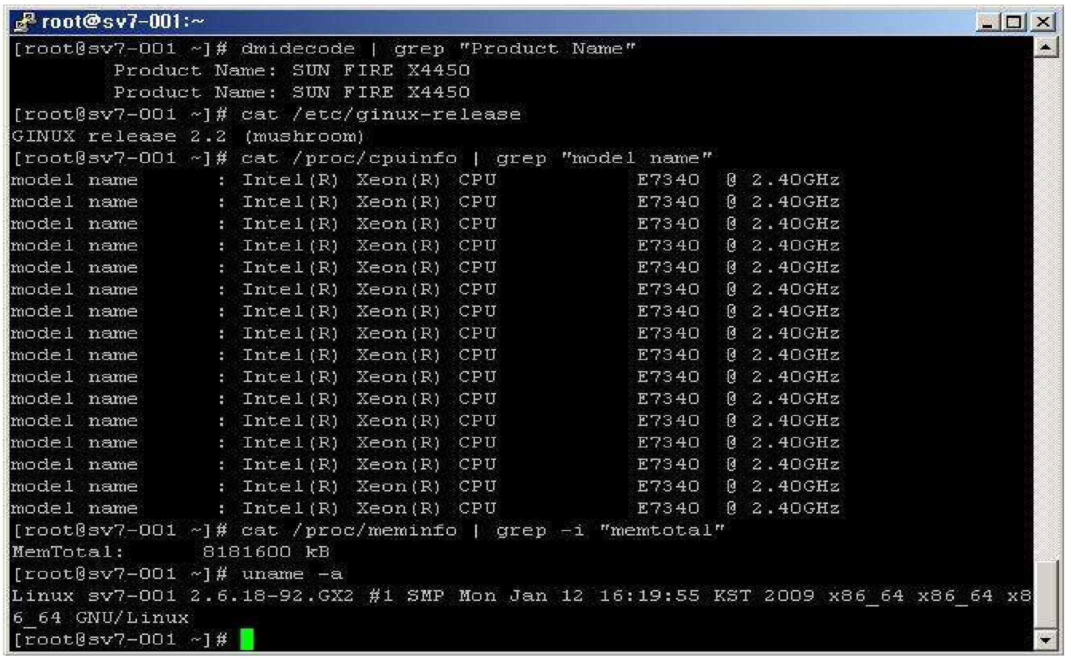
▪ IBM X3850 M2 모델에서 Ginux 설치 확인



```
root@sv2-002:~
[root@sv2-002 ~]# dmidecode | grep "Product Name"
        Product Name: IBM 3850 M2 / x3950 M2 -[72335RK]-
        Product Name: Node1 Processor Card
        Product Name: Node1 PCI I/O Planar
        Product Name: Node1 Memory Card1
        Product Name: Node1 Memory Card2
        Product Name: Node1 Memory Card3
        Product Name: Node1 Memory Card4
[root@sv2-002 ~]# cat /etc/ginux-release
GINUX release 2.2 (mushroom)
[root@sv2-002 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
[root@sv2-002 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:      8177744 kB
[root@sv2-002 ~]# uname -a
Linux sv2-002 2.6.18-92.GX2 #1 SMP Mon Jan 12 16:19:55 KST 2009 x86_64 x86_64 x8
6_64 GNU/Linux
[root@sv2-002 ~]#
```

[그림 4-2. IBM X3850M2에서 GINUX 상태 확인]

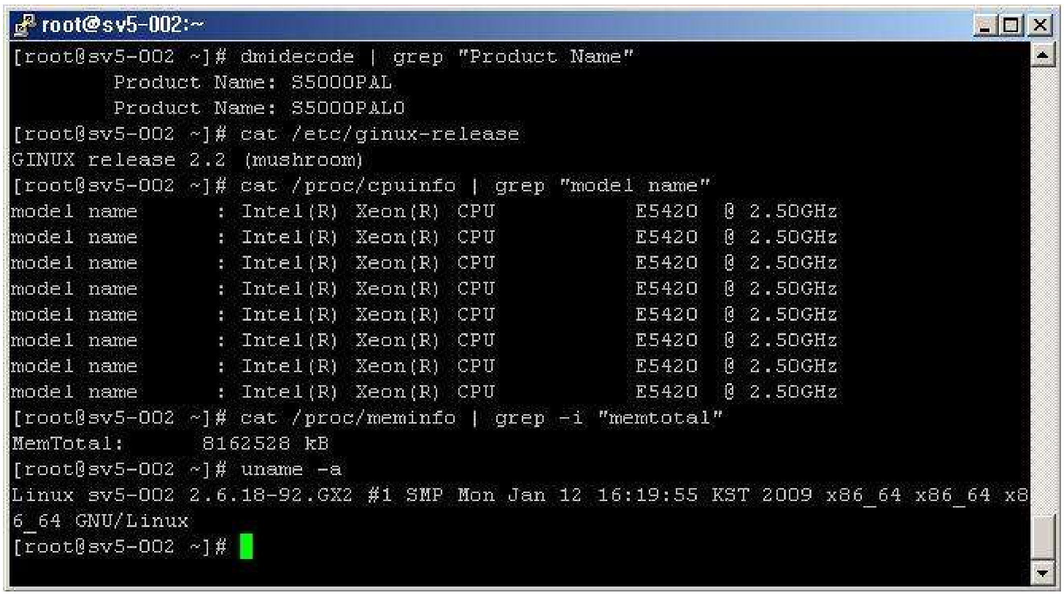
▪ SUN X4450 모델에서 Ginux 설치 확인



```
root@sv7-001:~
[root@sv7-001 ~]# dmidecode | grep "Product Name"
        Product Name: SUN FIRE X4450
        Product Name: SUN FIRE X4450
[root@sv7-001 ~]# cat /etc/ginux-release
GINUX release 2.2 (mushroom)
[root@sv7-001 ~]# cat /proc/cpuinfo | grep "model name"
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
[root@sv7-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:      8181600 kB
[root@sv7-001 ~]# uname -a
Linux sv7-001 2.6.18-92.GX2 #1 SMP Mon Jan 12 16:19:55 KST 2009 x86_64 x86_64 x8
6_64 GNU/Linux
[root@sv7-001 ~]#
```

[그림 4-3. SUN X4450에서 GINUX 상태 확인]

▪ TERATEC TERA2500ALLX 모델에서 Ginux 설치 확인



```
root@sv5-002:~
[root@sv5-002 ~]# dmidecode | grep "Product Name"
        Product Name: S5000PAL
        Product Name: S5000PAL0
[root@sv5-002 ~]# cat /etc/ginux-release
GINUX release 2.2 (mushroom)
[root@sv5-002 ~]# cat /proc/cpuinfo | grep "model name"
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
[root@sv5-002 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:      8162528 kB
[root@sv5-002 ~]# uname -a
Linux sv5-002 2.6.18-92.GX2 #1 SMP Mon Jan 12 16:19:55 KST 2009 x86_64 x86_64 x8
6_64 GNU/Linux
[root@sv5-002 ~]#
```

[그림 4-4. TERATEC TERA2500ALLX에서 GINUX 상태확인]

※ 참고 사항

dmidecode 명령은 하드웨어 업체에서 정보를 제공하는 경우만 정상적인 결과 값을 얻을 수 있는데, 테라텍 서버의 경우 모델명이 일치하지 않음

나. 기능 테스트

□ 네트워크 서비스 지원

[표 4-5. GINUX 네트워크 서비스 확인]

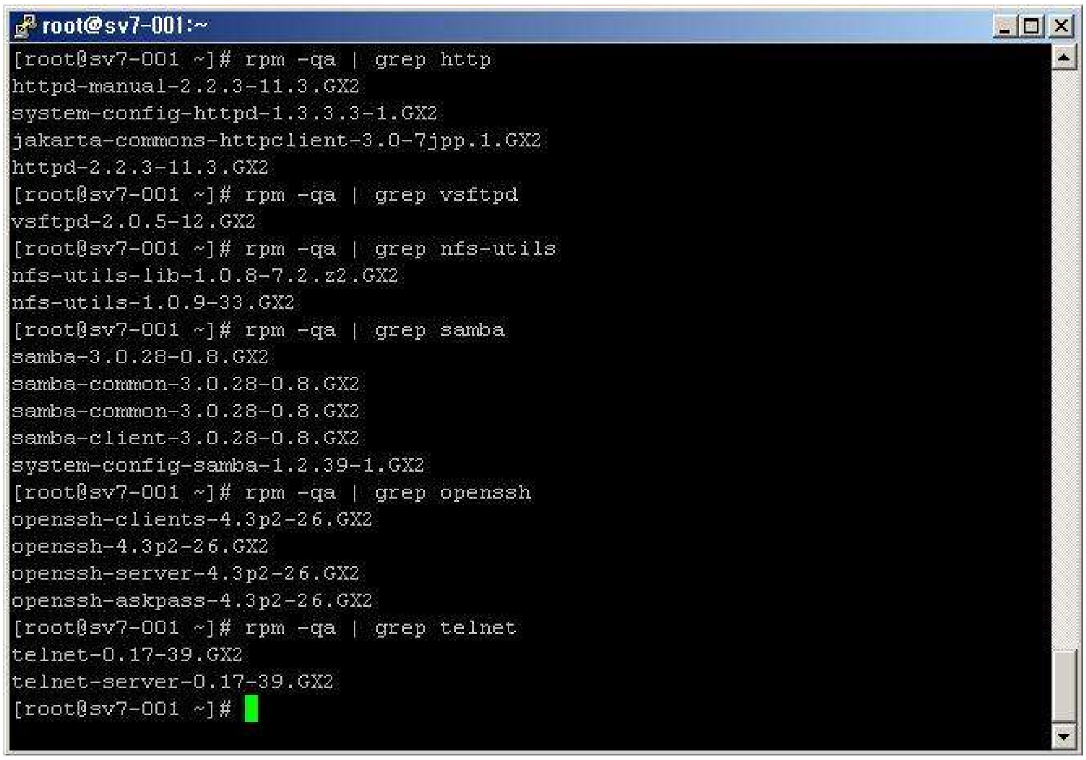
서비스 명	패키지 명	버전	지원 여부	비고
WEB	httpd	2.2.3-11.3	지원	
FTP	vsftpd	2.0.5-12	지원	
NFS	nfs-utils	1.0.9-33	지원	
Samba	samba	3.0.28-0.8	지원	
SSH	openssh	4.3p2-26	지원	
Telnet	telnet-server	0.17-39	지원	

- 서비스별 지원 확인 명령

[표 4-6. 서비스별 지원 명령]

구분	명령어	결과
서비스 확인 명령	rpm -qa | grep "package_name"	[그림4-5] 참고

▪ 서비스별 지원



[그림 4-5. GINUX 서비스 지원 확인]

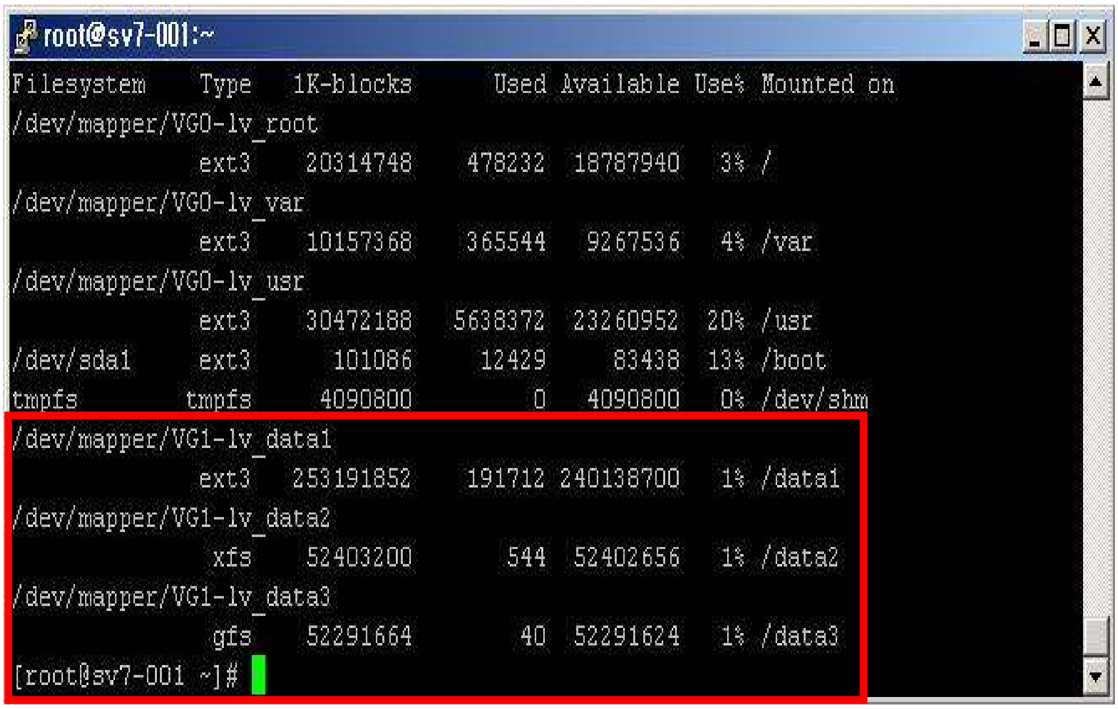
□ 파일시스템 지원

- 파일 시스템 지원 확인 명령

[표 4-7. 파티션별 지원 확인 명령]

구분	명령어	결과
ext3	mkfs -t ext3 /dev/VG1/lv_data1 mount /dev/VG1/lv_data1 /data1	[그림4-6] 참고
xfs	mkfs -t xfs /dev/VG1/lv_data2 mount /dev/VG1/lv_data2 /data2	
gfs	gfs_mkfs -p lock_nolock -j 1 /dev/VG1/lv_data3 mount /dev/VG1/lv_data3 /data3	
파일시스템 확인 명령	df -T	

▪ ext3, xfs, gfs 파일시스템



[그림 4-6. 파티션별 지원 확인]

□ LVM 지원 확인

[표 4-8. LVM 지원 확인]

패키지명	버전	지원여부	비 고
lvm2	2.02.32-4	지원	

- LVM 확인 명령

[표 4-9. LVM 지원 확인 명령]

구분	명령어	결과
LVM 확인 명령	rpm -qa | grep "package_name"	[그림4-7] 참고

▪ LVM 확인

[그림 4-7. LVM 지원 확인]

※ LVM(Logical Volume Manager)

저장장치들을 좀 더 효율적이고 유연하게 관리할 수 있는 커널의 부분 프로그램

- LVM 사용의 장점
  - ▪ 쉬운 관리
  - ▪ 서로 다른 많은 디바이스 조합 지원
  - ▪ 직관적인 저장 장치로의 접근
  - ▪ 뛰어난 확장성
  - ▪ 신뢰되는 안정성과 효율성
  - ▪ 스냅샷 기능 지원
  - ▪ 대용량 단일 파일시스템 구성 지원

□ 보안 기능 지원

[표 4-10. 보안 기능 지원 확인]

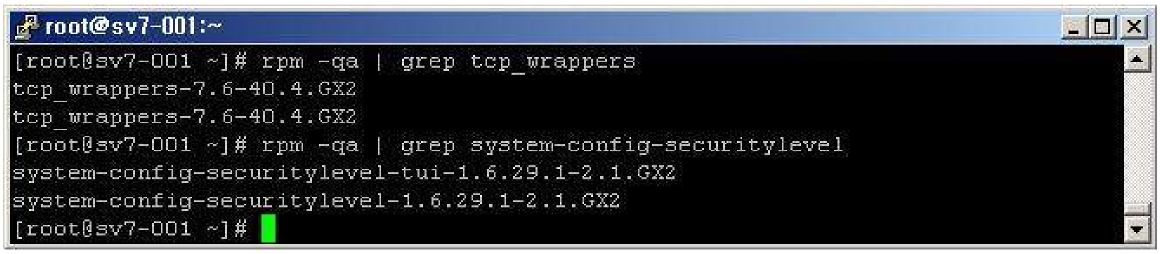
패키지명	버전	지원여부	비 고
tcp_wrappers	7.6-40.4	지원	
system-config-securitylevel	1.6.29.1-2.1	지원	

- 보안 기능 확인 명령

[표 4-11. 보안 기능 지원 명령]

구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "package_name"	[그림4-8,4-9] 참고
GUI용 명령	system-config-securitylevel	

▪ 패키지 확인



[그림 4-8. 보안 기능 지원 확인]

▪ GUI용 보안 툴 명령

[그림 4-9. GUI 보안 기능 지원 확인]

□ 자동 업데이트 기능 지원

[표 4-12. 업데이트 지원 확인]

패키지명	버전	지원여부	비 고
yum	3.2.8-9.1	**지원**	

- 업데이트 기능 확인 명령

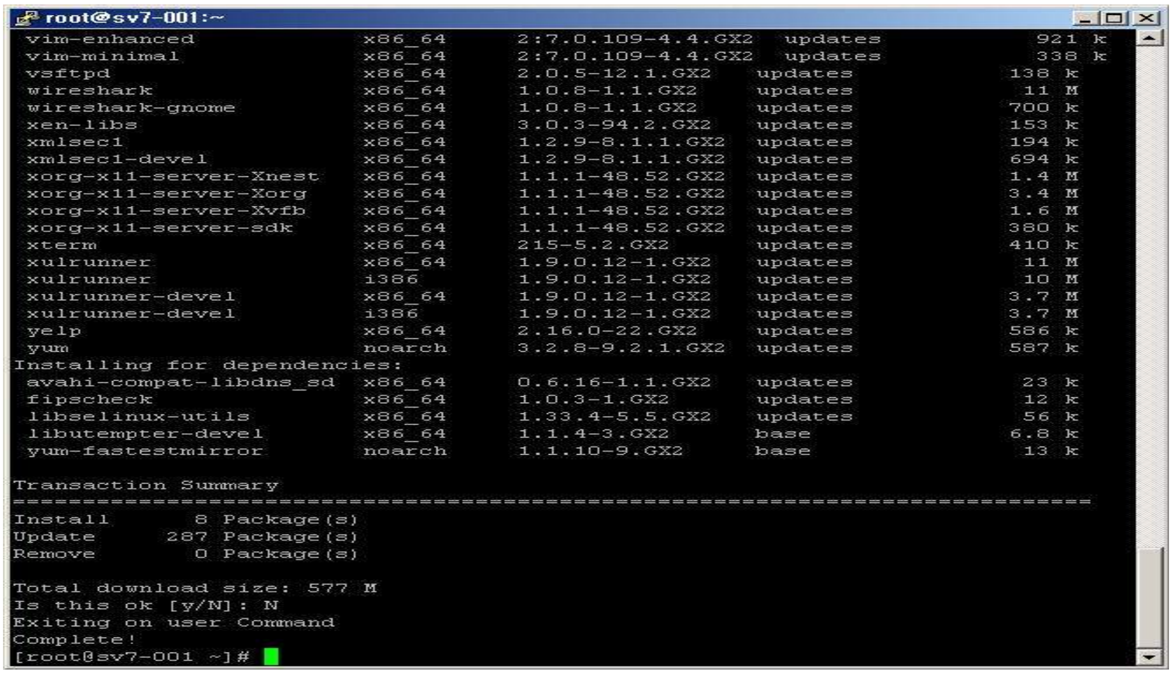
[표 4-13. 업데이트 지원 명령 확인]

구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "*package_name*"	[그림4-10~13] 참고
GUI용 명령	pirut, pup	

▪ 패키지 확인

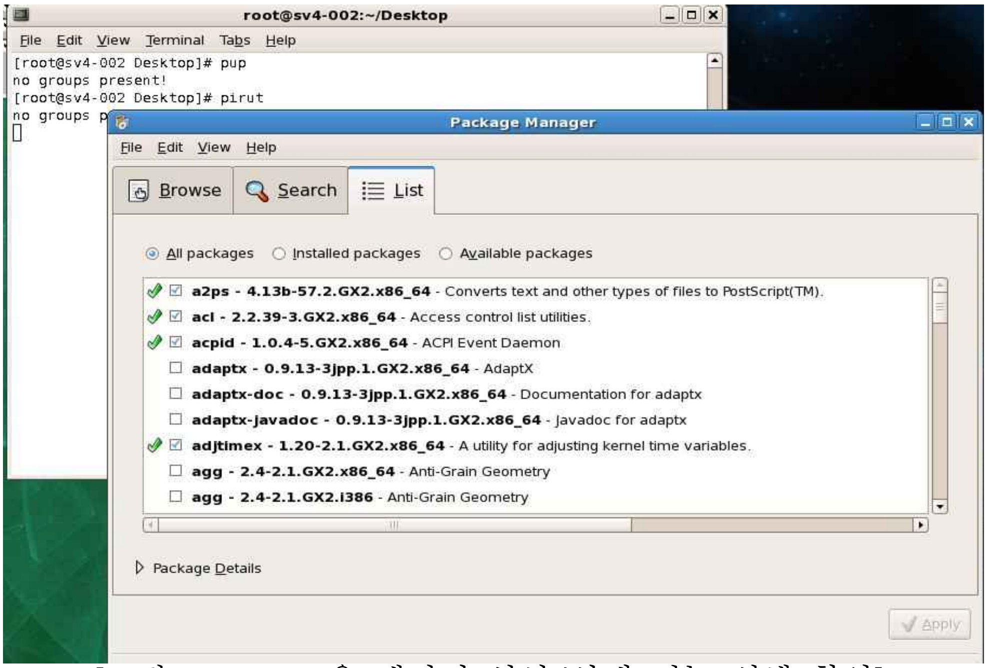
[그림 4-10. 업데이트 기능 지원 확인]

▪ 업데이트 실행 확인 (yum update)



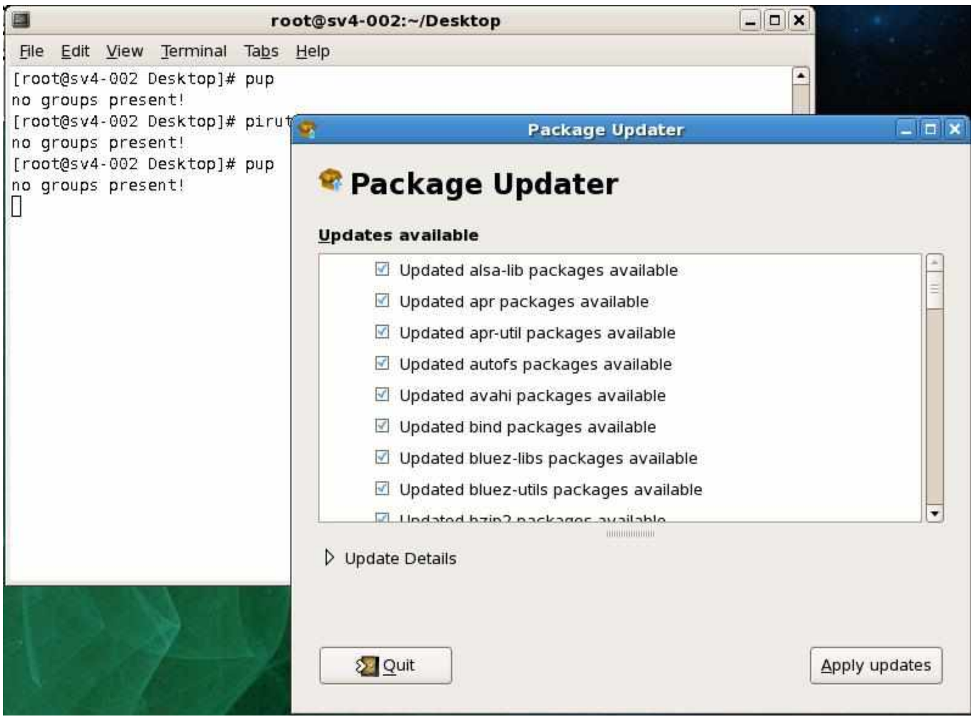
[그림4-11. 업데이트 기능 실행 확인]

▪ pirut
GUI용 소프트웨어 설치 및 삭제 유틸리티



[그림 4-12. GUI용 패키지 설치/삭제 기능 실행 확인]

▪ pup
GUI용 패키지 업데이트



[그림 4-13. GUI용 업데이트 기능 실행 확인]

다. 신뢰성 테스트

□ LTP Tool 사용

Linux Test Project (LTP)는 리눅스의 신뢰성, 호환성, 안정성을 입증하여 오픈소스 커뮤니티에 테스트 도구를 제공하는 것을 주 목적으로 하는 오픈소스 프로젝트이다.

- 테스트 방법

runalltests.sh 스크립트를 통한 초기 기본 테스트

▪ 파일시스템 스트레스 테스트
▪ 디스크 I/O 테스트
▪ 메모리 관리 스트레스 테스트
▪ IPC 스트레스 테스트
▪ 스케줄러 테스트
▪ 명령어 함수 확인 테스트
▪ 시스템 호출 함수 확인 테스트

- 실행 방법

```
$ cd /opt/ltp
$ ./runalltests.sh
```

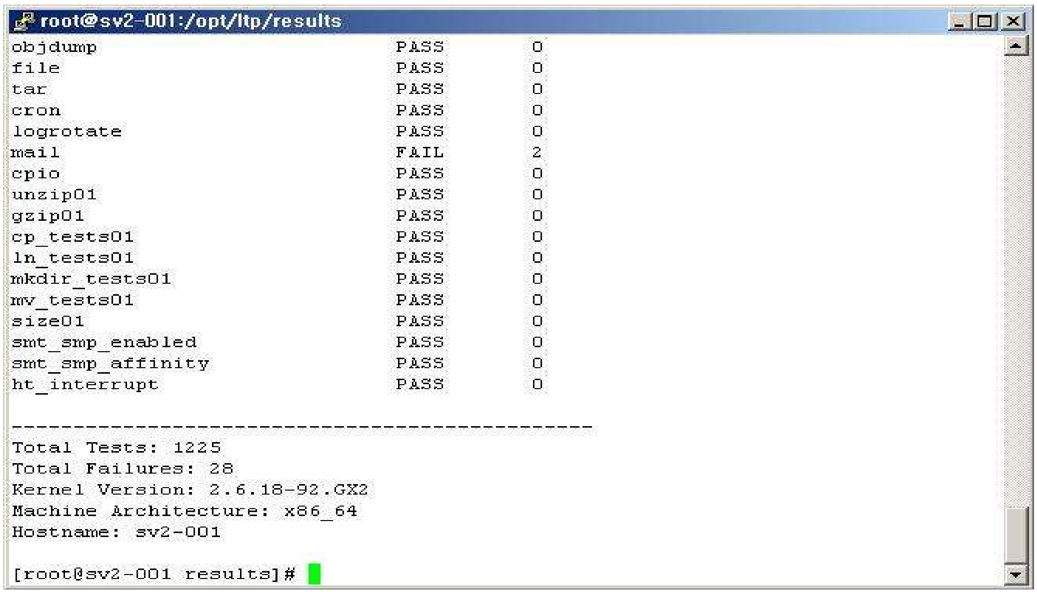
- 테스트 결과

▪ LTP 테스트 도구 버전 : ltp-full-20090930
▪ 테스트 시험 결과

[표 4-14. LTP 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	1225	28

※ 실패 목록은 GINUX에서 발생한 문제가 아닌 기능상 구현되지 않은 경우 이거나, LTP 툴의 자체 버그일 가능성이 높음([표 4-15] 참고)



```
root@sv2-001:/opt/ltp/results
objdump                          PASS      0
file                             PASS      0
tar                              PASS      0
cron                             PASS      0
logrotate                        PASS      0
mail                             FAIL      2
cpio                             PASS      0
unzip01                          PASS      0
gzip01                           PASS      0
cp_tests01                       PASS      0
ln_tests01                       PASS      0
mkdir_tests01                    PASS      0
mv_tests01                       PASS      0
size01                           PASS      0
smt_smp_enabled                  PASS      0
smt_smp_affinity                 PASS      0
ht_interrupt                     PASS      0

-----------------------------------------------
Total Tests: 1225
Total Failures: 28
Kernel Version: 2.6.18-92.GX2
Machine Architecture: x86_64
Hostname: sv2-001

[root@sv2-001 results]#
```

[그림 4-14. LTP 시험 결과 화면]

▪ 테스트 시험 실패 목록

[표 4-15. LTP 시험 실패 항목]

항 목	설 명
eventfd01	eventfd test는 임시 파일을 임시 디렉터리(tmp)가 아니라 현재 디렉터리에 생성하는데, 현재 디렉터리는 nonwriteable로 설정되어 오류 발생 가능
inotify02	kernel 소스 기반의 non-x86 arch system에서 오류 발생
ppoll01	ltp_signal.h에서 SIGSETSIZE 매크로를 사용하고, poll.h의 헤더 위치를 바꾸어서 테스트하면 정상적이지만 다른 테스트와의 호환을 위해서 수정하지 않고 테스트 수행, 오류 발생
quotactl01	__NR_set_tid_address 시스템 호출에 하드코딩해서 258번을 대치해야 정상 수행되지만 수정되지 않은 커널은 테스트 수행 시 오류 발생
rt_sigaction01	Functionality test 방식으로 SIGRTMIN(34)에서 SIGRTMAX(64)까지 사용한다. 만약 rt_sigaction으로 1번 시그널부터 64번 시그널을 테스트하고자 한다면, 별도의 패치를 기반으로 커널 2.6.23 이후의 버전으로 테스트를 한다.
rt_sigprocmask01	rt_sigprocmask 시스템 호출에 대한 test로 real time signal test인 경우 별도의 패치를 통해 test를 수행하지만 신뢰성 약함
rt_sigsuspend01	rt_sigsuspend 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 현재 segmentation fault가 발생해서 test 항목의 신뢰성 약함
set_thread_area01	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)
set_thread_area02	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)

stime01	time() 함수가 리턴한 data 값보다 stime() (x86-64에서 settimeofday())이 리턴한 값이 더 적을 경우 FAIL
sysconf01	errno 대신에 “actual" return code를 리턴하면서 sysconf의 fail 오류가 0이 아닐 경우 오류 발생
utimes01	EACCESS times가 NULL이고 호출자의 EUID가 파일에 접근하는 호출자와 일치하지 않은 경우 오류 발생, 패치를 적용할 경우에 오류 해결(최신버전)
proc01	/proc/self 파일을 읽지만 /proc/PID 디렉토리를 SKIP하는 경우가 많아 오류 발생 가능성, test 항목의 신뢰성 약함
su01	passwd -e 명령이 지원되지 않은 경우 FAIL 발생, 시스템에 영향 없음
cron_allow01	permission 문제가 발생하는 경우 FAIL, 시스템에 영향 없음
hugemmap01	errno 2번 No such file or directory 오류로 파일을 오픈하지 못할 경우 FAIL
hugemmap03	hugetlb 기반으로 normal page가 high memory region에 매핑할 수 없는지를 테스트하며 비정상적인 경우 FAIL
hugemmap04	적어도 하나의 테스트용 hugepage가 필요하고 hugepage의 개수에 대한 정보가 필요한데, 유효한 hugepage가 아닐 경우 FAIL
hugeshmat01	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmat02	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmat03	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmctl01	임의의 usleep time은 부정확한 실행 순서를 야기하며 FAIL
hugeshmctl02	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패땐 FAIL
hugeshmctl03	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패땐 FAIL
hugeshmdt01	hugepage가 가용할 경우에만 hugepages를 hugeshmdt에 할당, 실패땐 FAIL
hugeshmget01	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget02	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget03	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget05	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
mail	메일 발송 정보 오류 /etc/xxx-release 파일 참조

라. 안정성 테스트

□ LTP Stress Tool 사용

LTP 테스트 도구를 사용한 스트레스 테스트를 수행한다.

- NFS 스트레스 테스트
- 메모리 관리 스트레스 테스트
- 파일시스템 스트레스 테스트
- Math (floating point) 테스트
- pthread 스트레스 테스트
- 디스크 I/O 테스트
- IPC (pipeio, semaphore) 테스트
- 시스템 호출 함수 확인 테스트
- 네트워킹 스트레스 테스트

- 실행 방법

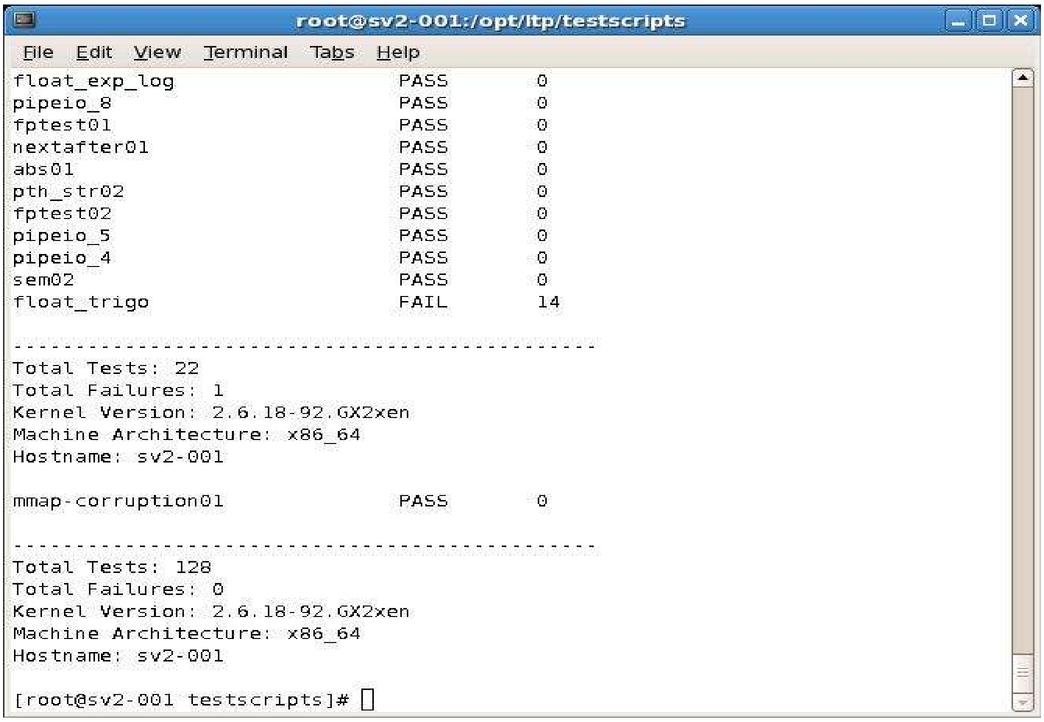
```
$ cd /opt/ltp/testscripts
$  ./ltpstress.sh -d /tmp/sardata -l /tmp/ltplog.1122 -m 1024 -t 24 -S
```

- 테스트 결과
  - LTP 테스트 도구 버전 : ltp-full-20090930
  - 테스트 시험 결과

[표 4-16. LTP Stress 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	22	1
	128	0

※ 실패 목록은 GINUX에서 발생한 문제가 아닌 기능상 구현되지 않아 발생한 오류일 가능성이 높음([표 4-17] 참고)



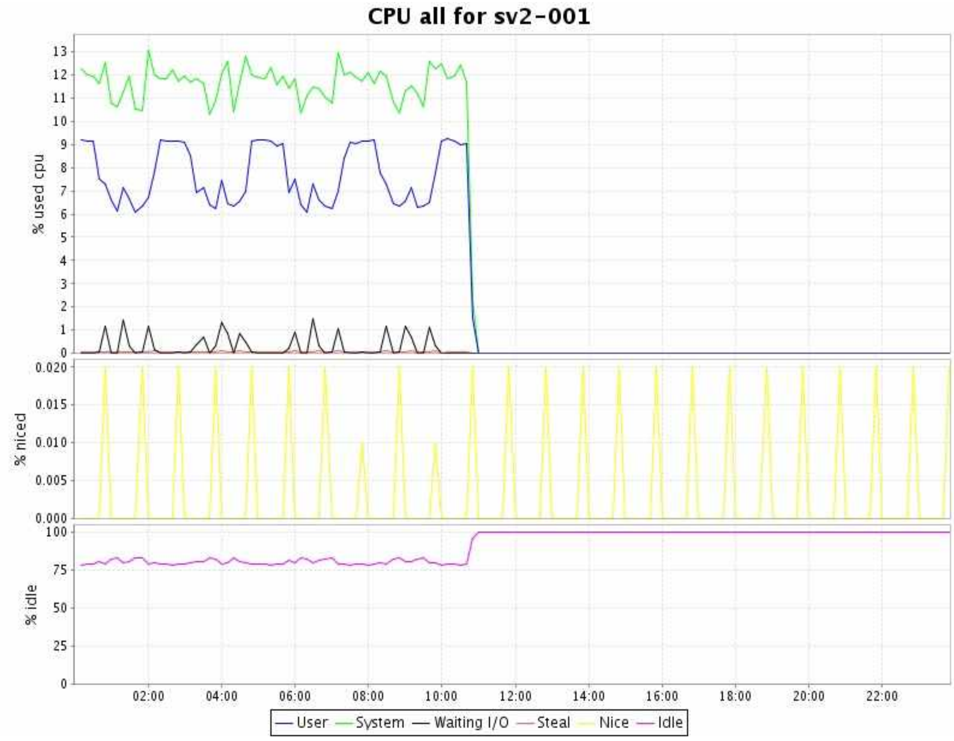
[그림 4-15. LTP Stress 시험 결과 화면]

▪ 테스트 시험 실패 목록

[표 4-17. LTP Stress 실패 항목]

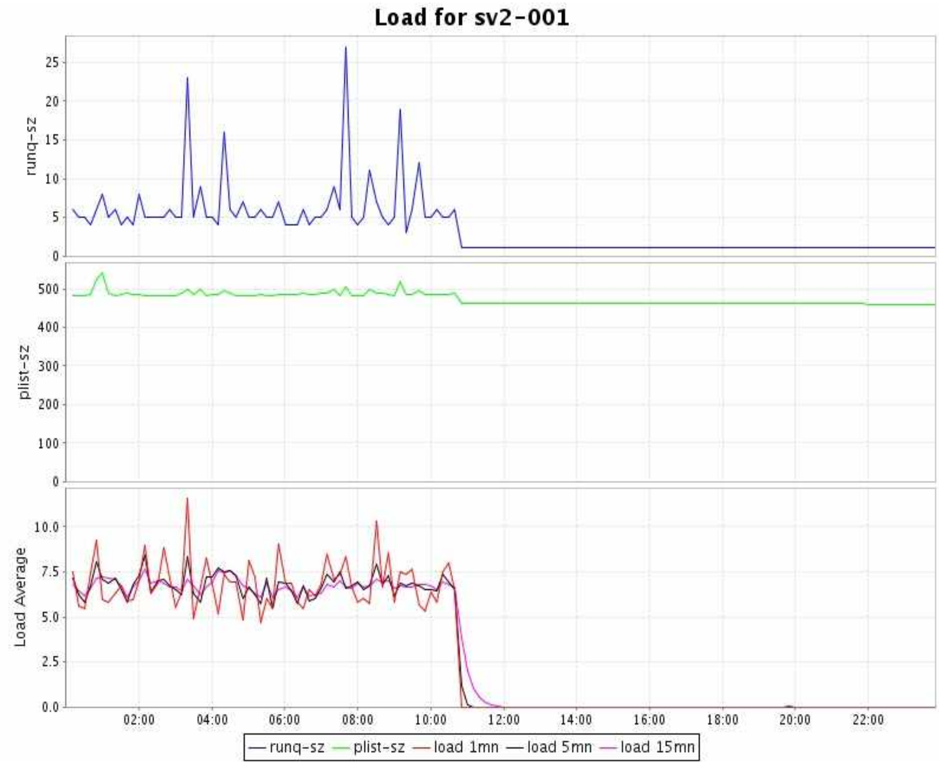
항　목	설　명
mc_cmds	ltp network test일 경우 /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts 값이 1로 설정된 경우 FAIL 출력
sysconf01	errno 대신에 “actual" return code를 리턴하면서 sysconf의 fail 오류가 0이 아닐 경우 오류 발생
float_trigo	커널 oom_killer 발생시에도 ltp 테스트가 수행되는 환경을 테스트하는 항목으로 FAIL일 경우 커널 oom_killer 기반으로 float연산 부분의 오류 가능성 제기

▪ 시스템 사용률



[그림 4-16. CPU 사용률 확인]

[그림 4-17. Memory 사용률 확인]



[그림 4-18. 평균 사용률 확인]

[표 4-18. 시스템 사용량 요약]

구분	최대 사용량	평균 사용량
CPU	100%	20%
Memory	8GB	2GB
Load Average	24	7

※ 참고 사항

- [그림 4-16~18] 그래프는 하루를 기준으로 작성되었다.
- 스트레스 테스트 그래프는 00시부터 ~ 11시까지 해당 된다.
- 수치가 낮을수록 시스템 부하가 적은 것이다.

5. 결론

□ 설치

최신의 하드웨어 테스트 환경에서 드라이버가 기본으로 제공되므로 누구나 쉽게 설치가 가능하며, SK C&C에서 제작 및 배포하는 제품이므로 기술지원도 어렵지 않게 받을 수 있다.

□ 기능

Red Hat Enterprise에서 지원하는 대부분의 기능을 지원하며, 다른 Unix나 Windows에서 지원하는 기능 또한 부족함이 없으므로 정부기관 및 기업에서 도입이 가능한 제품이다.

□ 신뢰성

LTP 툴에서 보여주듯 1000가지가 넘는 기능 테스트 중 대부분이 문제없이 테스트 되었으며, 일부 테스트의 경우 기능상 구현되지 않았거나 지원하지 않는 부분의 오류이므로 운영체제에 문제가 있다고 판단하기는 어려우며 서비스용으로 사용하더라도 문제가 없을 것으로 보인다.

□ 안정성

스트레스 테스트 결과에서 보이듯 시스템이 안정적으로 운영되기 위한 기본 요건을 갖춘 제품이다.

Ginux는 국내 몇 개 안되는 배포판 중 하나로 테스트 결과 사용 중 중요 문제점을 발견할 수 없었으며 기술지원을 받을 수 있는 제품이므로 더욱 안정적이고 신뢰할 수 있는 제품이라 할 수 있다.

*참고 자료

http://www.ginux.co.kr/

http://ltp.sourceforge.net/

* 별첨

별첨1. LTP 활용 가이드

Ⅵ Glassfish

목 차

1. 개요

보편적으로 개발/구축된 Web시스템 중에는 기능사양에는 만족하고 있지만, 퍼포먼스라는 점에서는 사용자가 큰 불만을 품고 있는 사례가 흔하게 발생하고 있으며, 그 이유로서는 부하테스트(성능테스트)가 거의 이루어지지 않는 프로젝트가 많았기 때문이다.
불특정다수의 사용자가 이용하는 web애플리케이션에 있어서는 만족할만한 편리한 기능이 갖추어져 있다고 하여도, 충분한 퍼포먼스가 나오지 않으면 사용자의 만족도가 저감하게 된다.

이에, 본 테스트 보고서에서는 부하테스트 툴인 JMeter를 사용하여, WAS(Web Application Server)의 대표적인 공개SW인 GlassFish를 대상으로 부하테스트를 실행하여 결과를 기술하도록 한다.

가. 목적

GlassFish 내장의 어플리케이션의 부하를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 현재 구축된 시스템이 어느 정도의 부하에 견딜 수 있는가를 테스트한다.
- 상정한 동시 사용자수를 만족하는지 검증한다.
- 현재 시스템의 변경으로 성능향상의 정도를 측정한다.

나. 테스트 대상 및 테스트 툴의 선정기준

본 테스트에 있어서의 테스트 대상 및 테스트 툴을 선정함에 있어서 다음과 같은 특성을 열거한다.

1) 테스트 툴 - JMeter

JMeter는 Apache Jakarta Project에서 개발이 진행되고 있는 부하테스트 애플리케이션이다.
Web애플리케이션(HTTP베이스의 애플리케이션)뿐만 아니라, FTP, JDBC, LDAP 등

의 인터페이스기반의 애플리케이션에 대하여 부하테스트를 실시하는 것이 가능하다. Web애플리케이션의 부하테스트를 실시하는 도구로서는 JMeter 이외에도 OpenSTA, Mercury LoadRunner 등이 있지만, 다음의 관점으로 JMeter를 사용하였다.

- 100% pure Java이다.
  Java로 기술되어 있으므로 OS환경에 구애받지 않는다.
- 오픈소스이다.
  오픈소스이므로 도입 비용이 거의 안 들고, 커뮤니티도 활발하다.
- 조작이 간단하다.
  조작이 간단하고 CUI는 물론 GUI로 테스트를 실시할 수가 있다.

2) 테스트 대상 - GlassFish

Glassfish Project는 2005년 6월에 Sun에 의해서 Launch 된 Open Source 기반의 J2EE RI(Reference Implementation) 구현을 위한 Project로써, 기존 J2EE RI 가 J2EE 개발자들만을 위한 Application 테스트 환경을 제공하는 데 그 주목적이 있었던 반면 Glassfish는 J2EE SPEC에 부합하며 Mission Critical 한 기업환경에서도 운영될 수 있는 J2EE Container를 구현하고 제공한다.

- J2EE 5 지원
- 확장성 있는 서버 애플리케이션을 위한 Java New IO 프레임워크
- J2EE Technology 에 대한 Reference Implementation
  - J2EE 5 SDK 내에 포함되어 제공
- Enterprise Quality
  - 탁월한 성능, Session Clustering , 중앙 집중 관리 기능
- Community 기반의 Open Source Project
  - http://glassfish.dev.java.net 을 통해 개발 Project 진행
  - 다양한 Sub Project 운영
    (Core : Grizzly, OpenMQ, SailFin, Shoal, HK2
    Web Tier : Phobos, WADL, jMatki, JSF, Woodstock, Jersey
    Web Service : Metro, JAX-WS, Tango(WSIT), JAXB, JAXP)

다. 전제조건

- 본 부하 테스트는 Hardware 및 네트워크에 관련된 모니터링에 대해서는 언급하지 않는다.
- 본 부하 테스트는 JMeter를 사용한 Glassfish 내장 애플리케이션의 성능을 검증하기 위한 목적이므로 테스트 결과에 대한 분석 및 반영으로 의한 성능향상 방법은 별개의 과제로 한다.

2. 테스트 환경

OS	Linux (2.6.18)
테스트 대상	GlassFish / glassfish-v2.1.1-b32g
애플리케이션	GlassFish 내장 애플리케이션
테스트 툴	JMeter

3. 테스트 절차

가. 소프트웨어 설치

(1) 서버

1) JDK

> JAVA_HOME=/usr/java/jdk1.6.0_17

2) Glassfish

① http://java.net/download/javaee5/v2.1.1_branch/promoted/Linux/ 에서 glassfish-installer-v2.1.1-b31g-linux.jar 을 다운로드

② 설치 디렉터리를 결정하고, 설치파일의 압축을 푼다.

> # cd /data

> # java -Xmx256m -jar glassfish-installer-v2.1.1-b32g-linux.jar

> GUI 환경의 라이선스 동의화면이 팝업 되고 동의를 클릭하면 압축이 풀림

③ ant 바이너리를 위한 실행 권한을 부여한다.

> # cd glassfish

> # chmod -R +x lib/ant/bin

④ glassfish 설치

> # lib/ant/bin/ant -f setup.xml

⑤ glassfish 기동

> # /data/glassfish/bin/asadmin start-domain

⑥ glassfish 기동 확인

웹브라우저에서 http://서버주소:4848/을 입력한다.

(관리자 ID : admin 관리자 / PW: adminadmin)

⑦ glassfish 서버 중지

> # /data/glassfish/bin/asadmin stop-domain

(2) 클라이언트

1) JDK

> JAVA_HOME=/usr/local/java/jdk1.5.0_21

2) JMeter

> JMeter_Manual.pdf 참조

나. 테스트 시나리오

(1) 시나리오

1) 초기화면을 표시한다.
2) NIPA로 인사한다.
3) SKCnC로 인사한다.

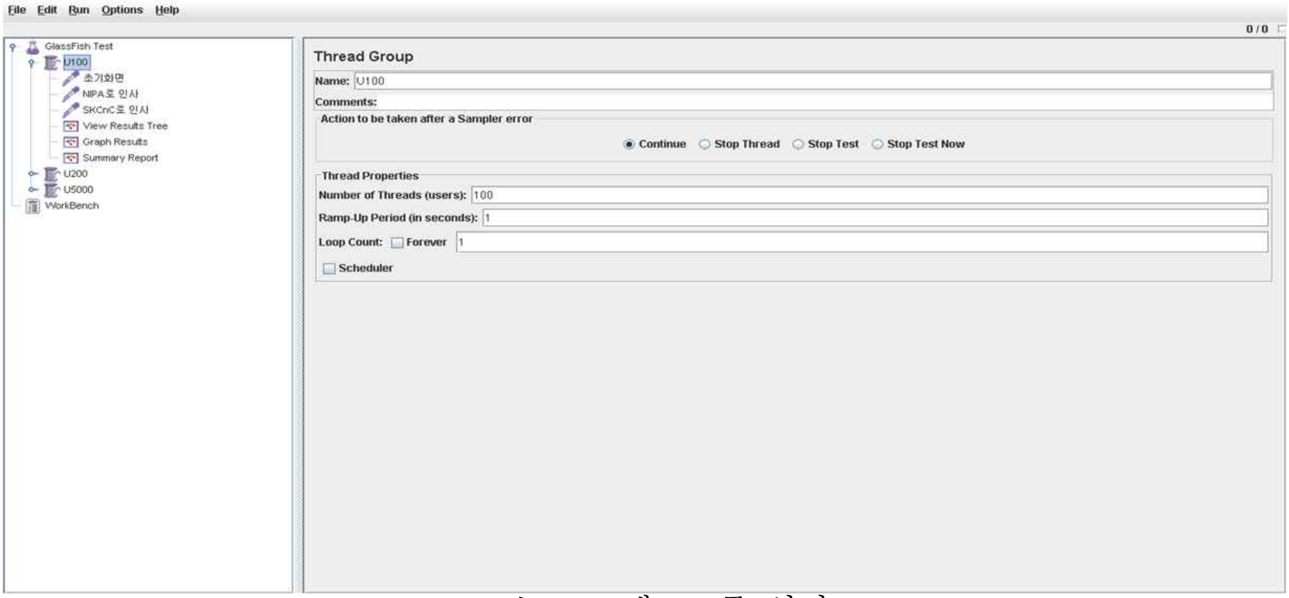
(2) 부하레벨의 설정

시스템의 부하레벨을 높여 가면서 테스트를 반복하여, 한계치를 조사하기 위하여, Ramp-Up기간을 1sec 고정으로, loop counter를 1로 하여 thread수(접속자수)를 증가시켜 1초당 동시 접속수를 조정하여 간다.

(3) JMeter 설정

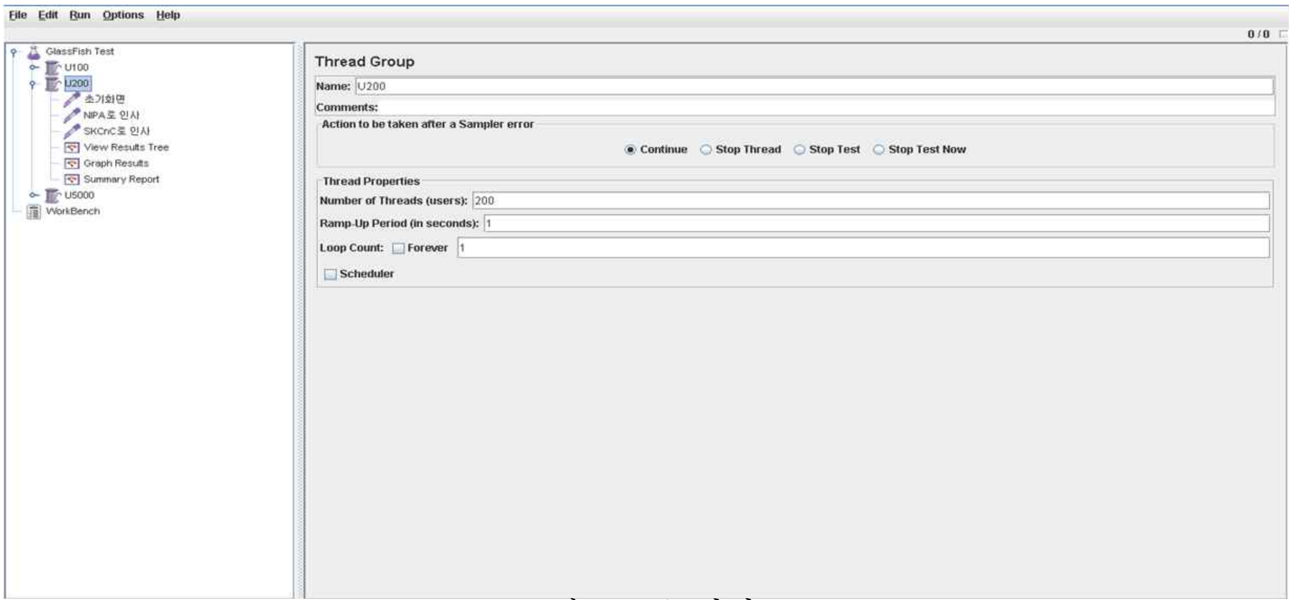
1) 테스트 플랜을 작성
2) Thread Group을 생성

● 100 User 동시 접속



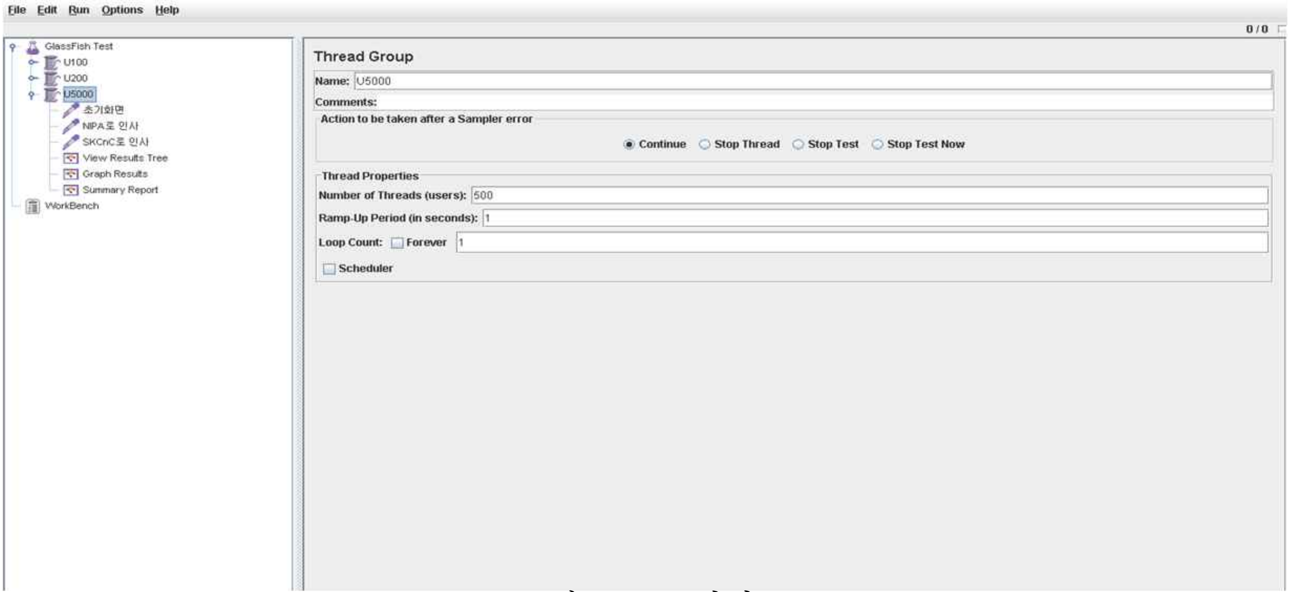
[그림3-1. 쓰레드그룹 설정 - 100 User]

● 200 User 동시 접속



[그림3-2. 쓰레드그룹 설정 - 200 User]

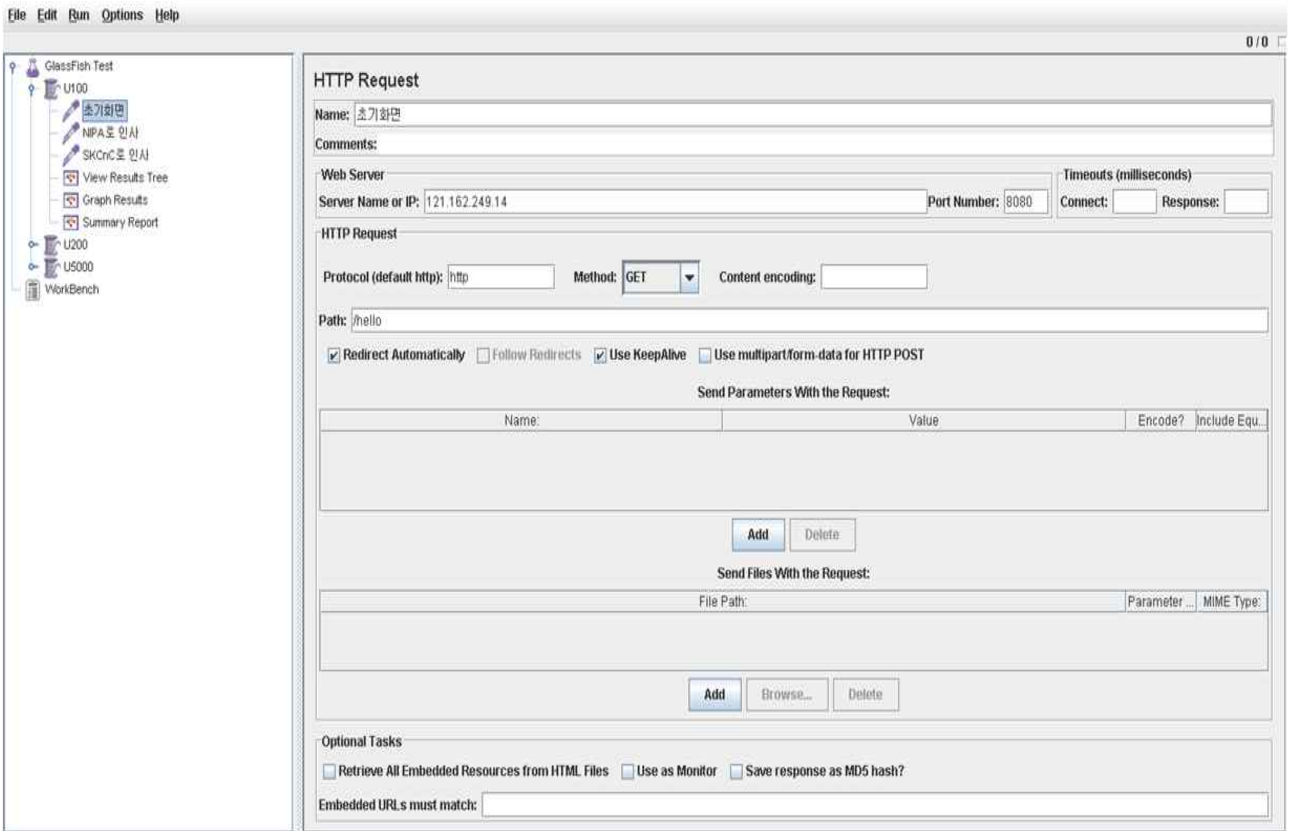
● 500 User 동시 접속



[그림3-3. 쓰레드그룹 설정 - 500 User]

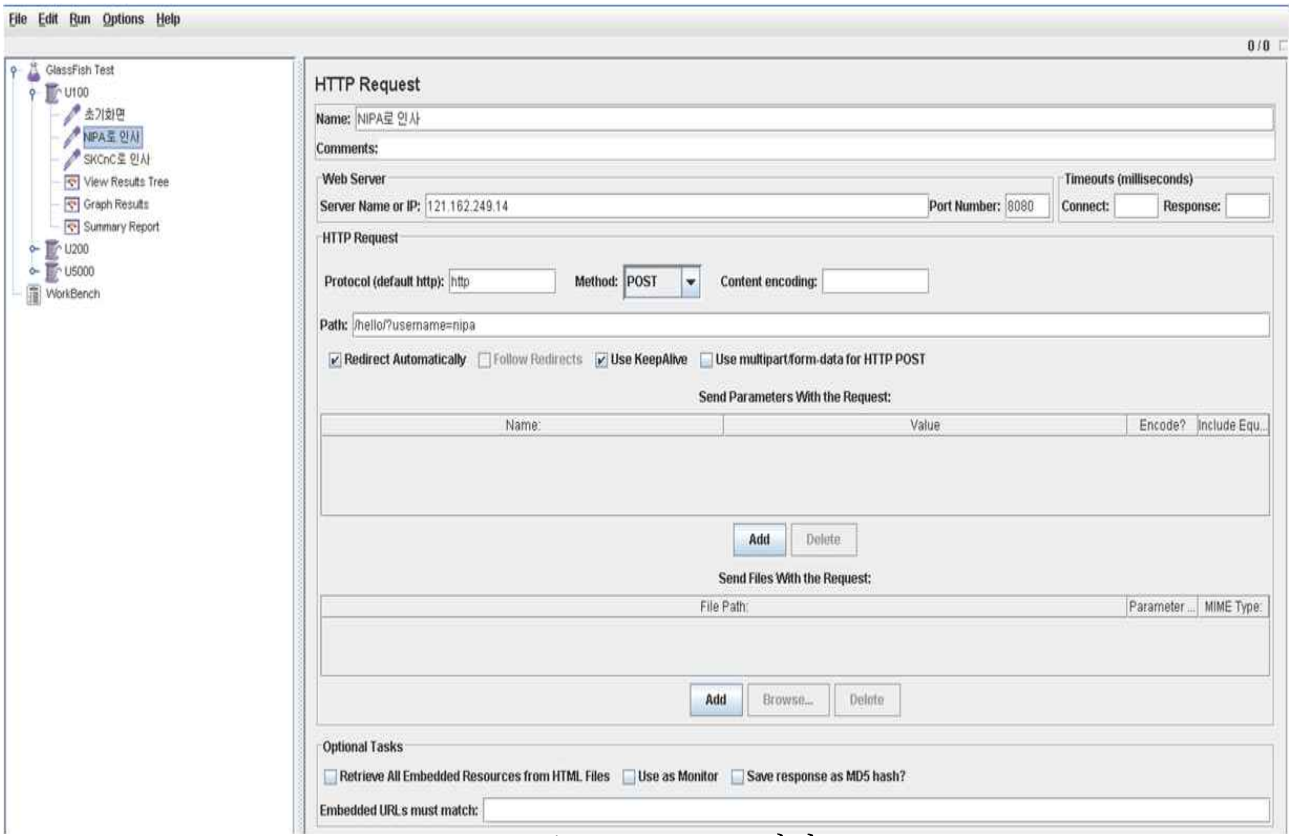
3) Http Request를 추가 (각 Thread Group에 공통)

● 초기화면



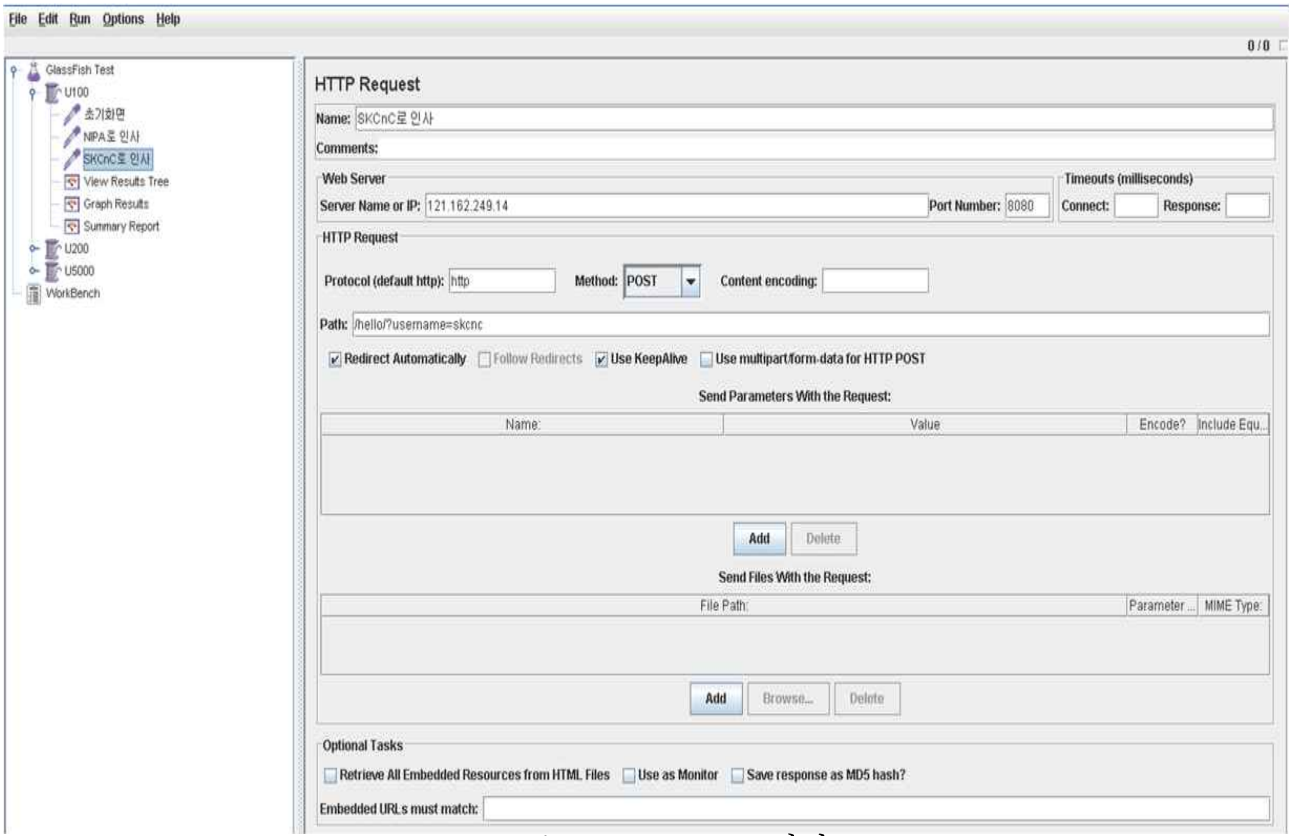
[그림3-4. Http Request 설정]

● NIPA로 인사



[그림3-5. NIPA로 인사]

● SKCnC로 인사



[그림3-6. SKCnC로 인사]

4) Listener를 추가 (각 Thread Group에 공통)

● View Results Tree
● Graph Results
● Summary Report

다. 테스트 실시

(1) GlassFish의 기동

> # /data/glassfish/bin/asadmin start-domain

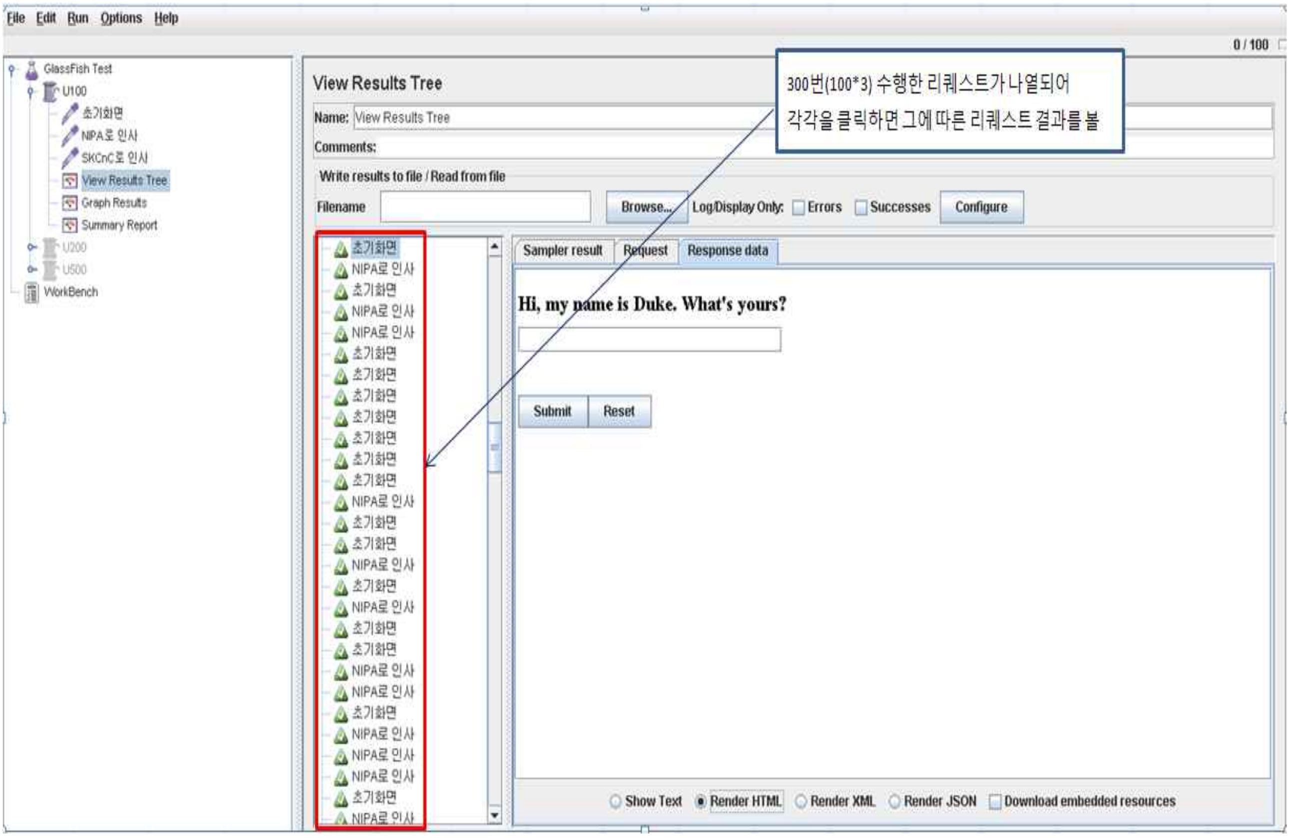
(2) Jmeter의 Run > Start

4. 테스트 결과

테스트 결과는
① View Results Tree - 초기화면
② View Results Tree - NIPA로 인사
③ View Results Tree - SKCnC로 인사
④ Graph Results
⑤ Summary Report
의 Listener로 확인한다.

가. 100 User가 동시 접속

① View Results Tree - 초기화면

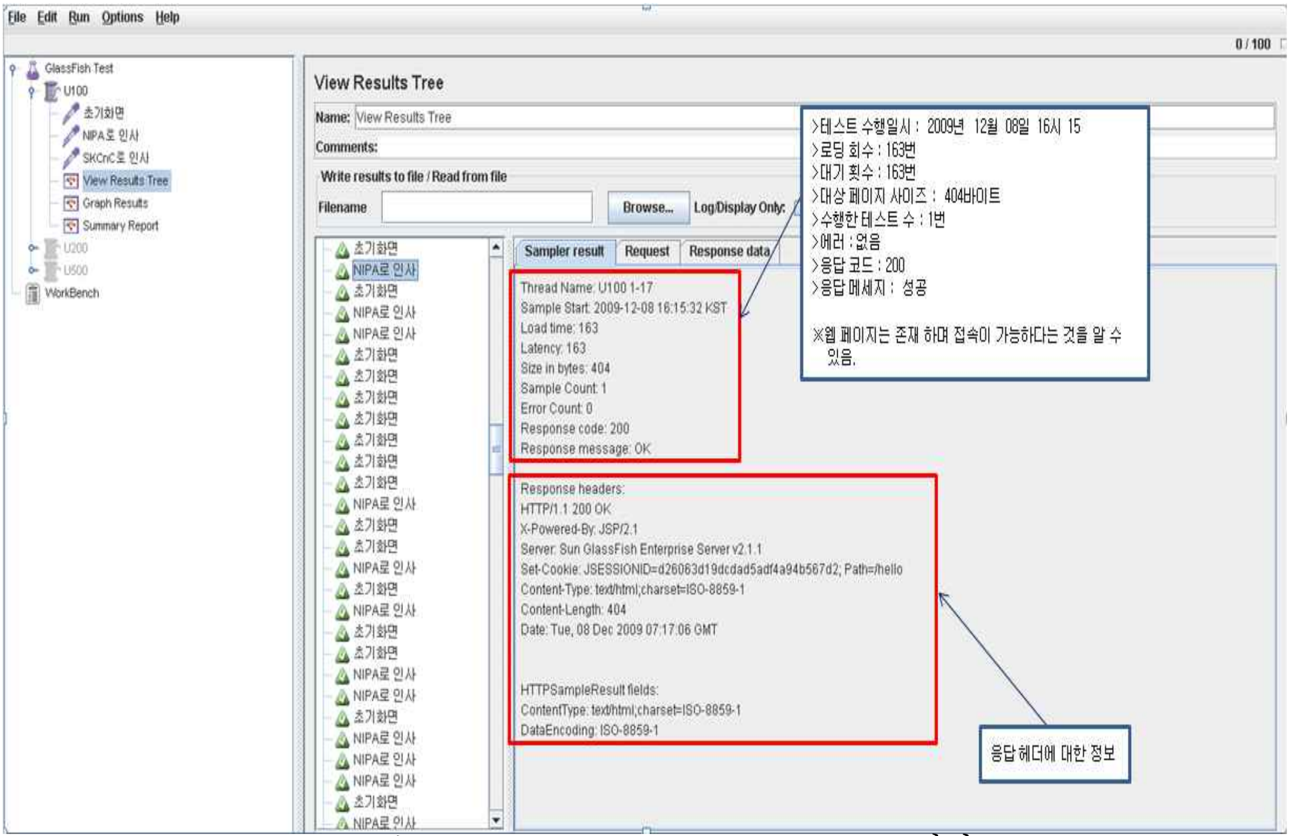


[그림4-1. View Results Tree - 초기화면]

방문자를 입력하기 위한 초기화면이다.

총 300번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

② View Results Tree - NIPA로 인사



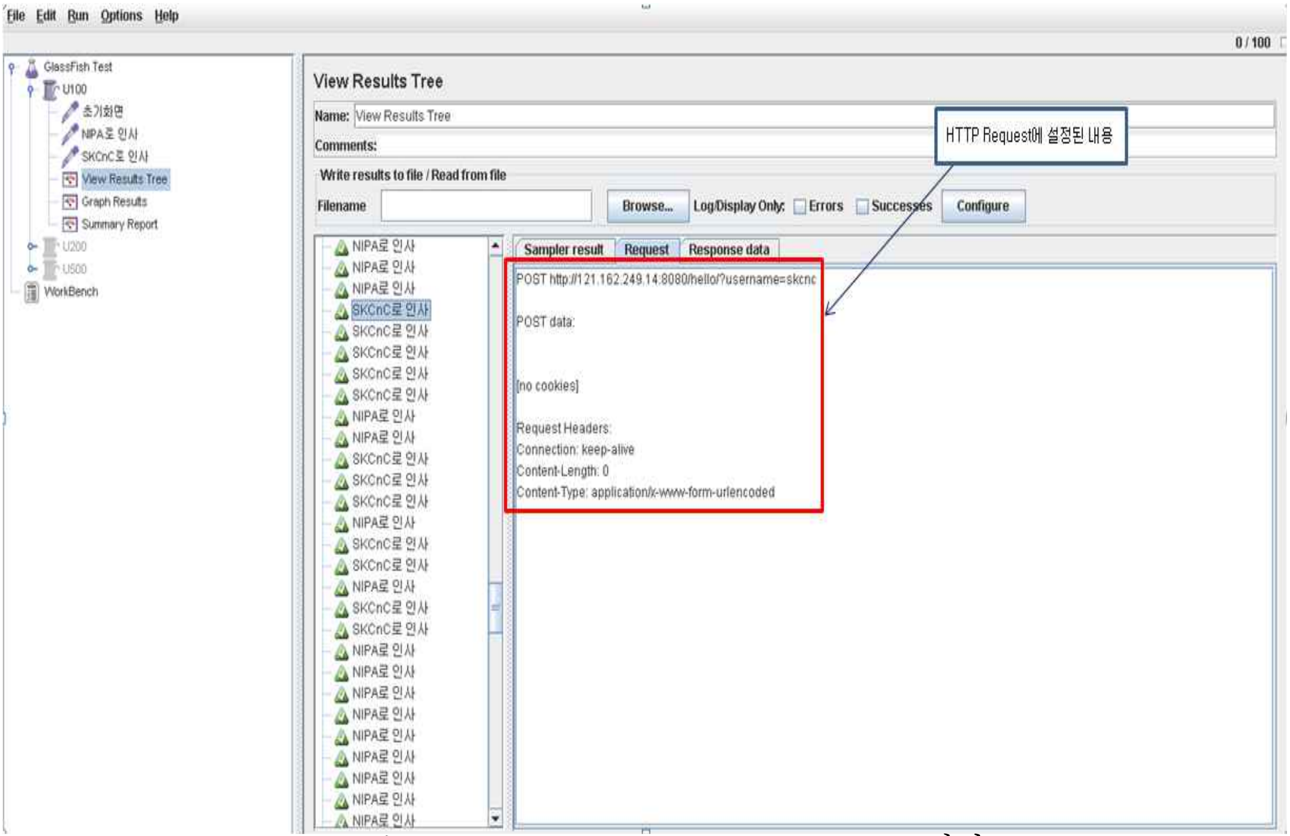
[그림4-2. View Results Tree - NIPA로 인사]

NIPA로 인사한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

- 로딩 회수 : 163번
- 대기 횟수 : 163번
- 대상 페이지 사이즈 : 404바이트
- 수행한 테스트 수 : 1번
- 에러 : 없음
- 응답 코드 : 200
- 응답 메시지 : 성공

③ View Results Tree - SKCnC로 인사

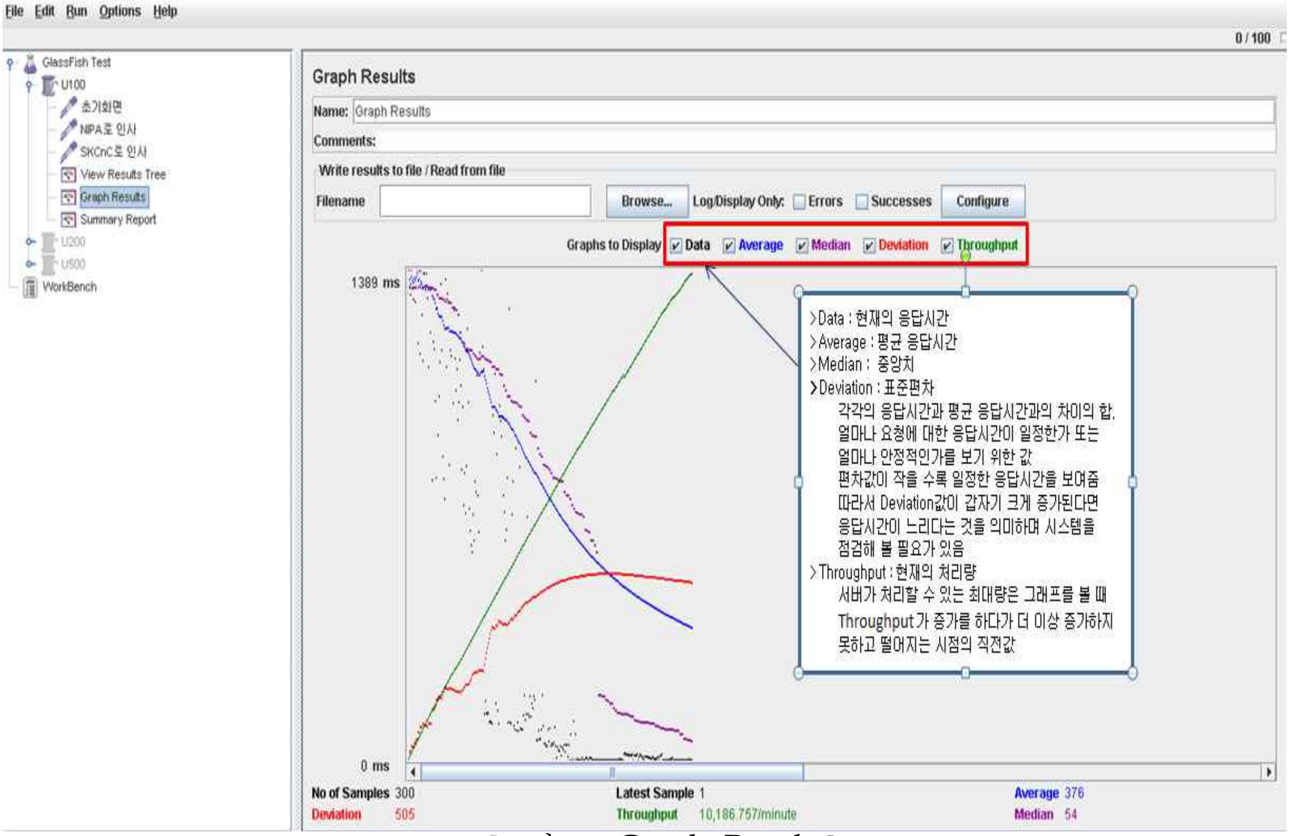


[그림4-3. View Results Tree - SKCnC로 인사]

SKCnC로 인사한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

④ Graph Results

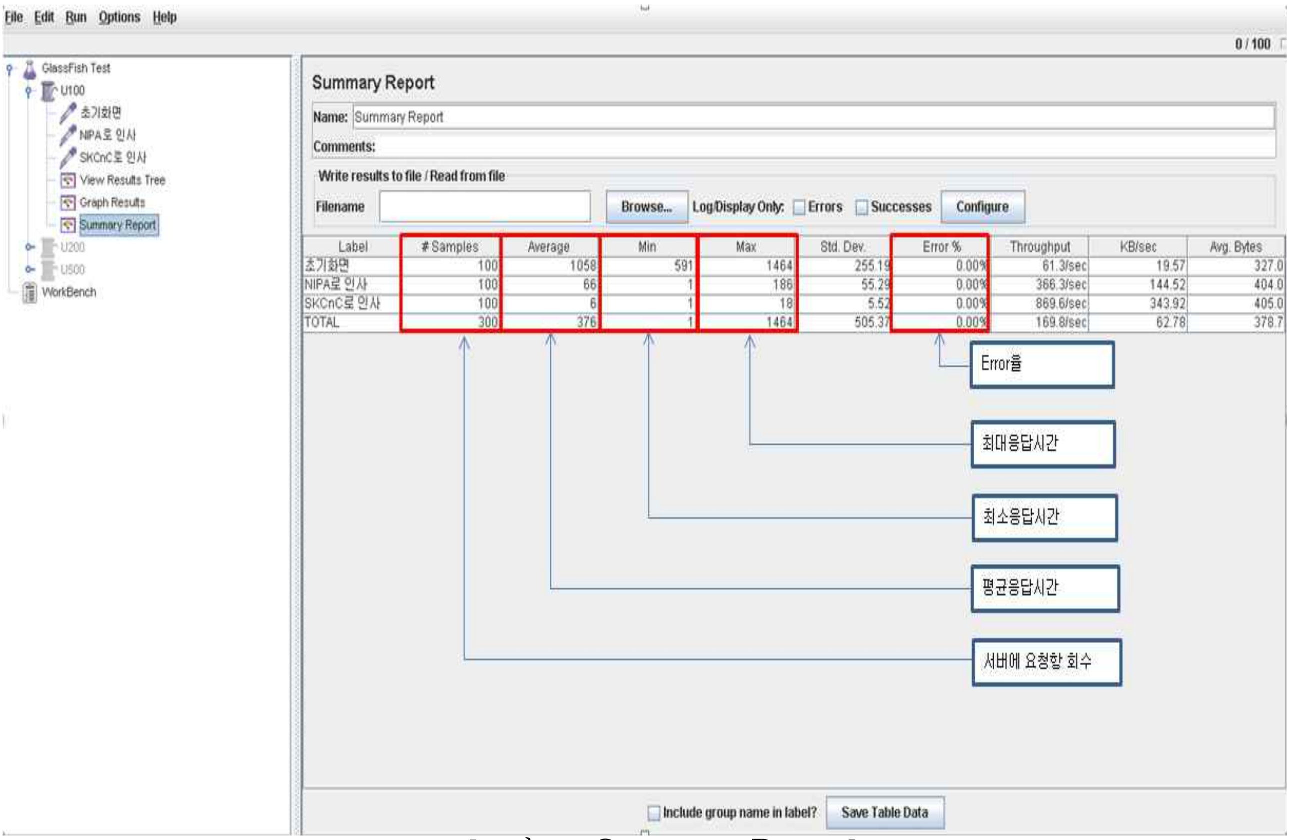


[그림4-4. Graph Results]

그래프를 분석하여 보면 응답시간, 처리량이 안정적인 것을 확인할 수 있다. 단, 표준편차가 갑자기 크게 증가되므로 시스템을 점검해볼 필요가 있다.

- Data : 300
- Average : 376
- Median : 54
- Deviation : 505
- Throughput : 10186.757/min

⑤ Summary Report



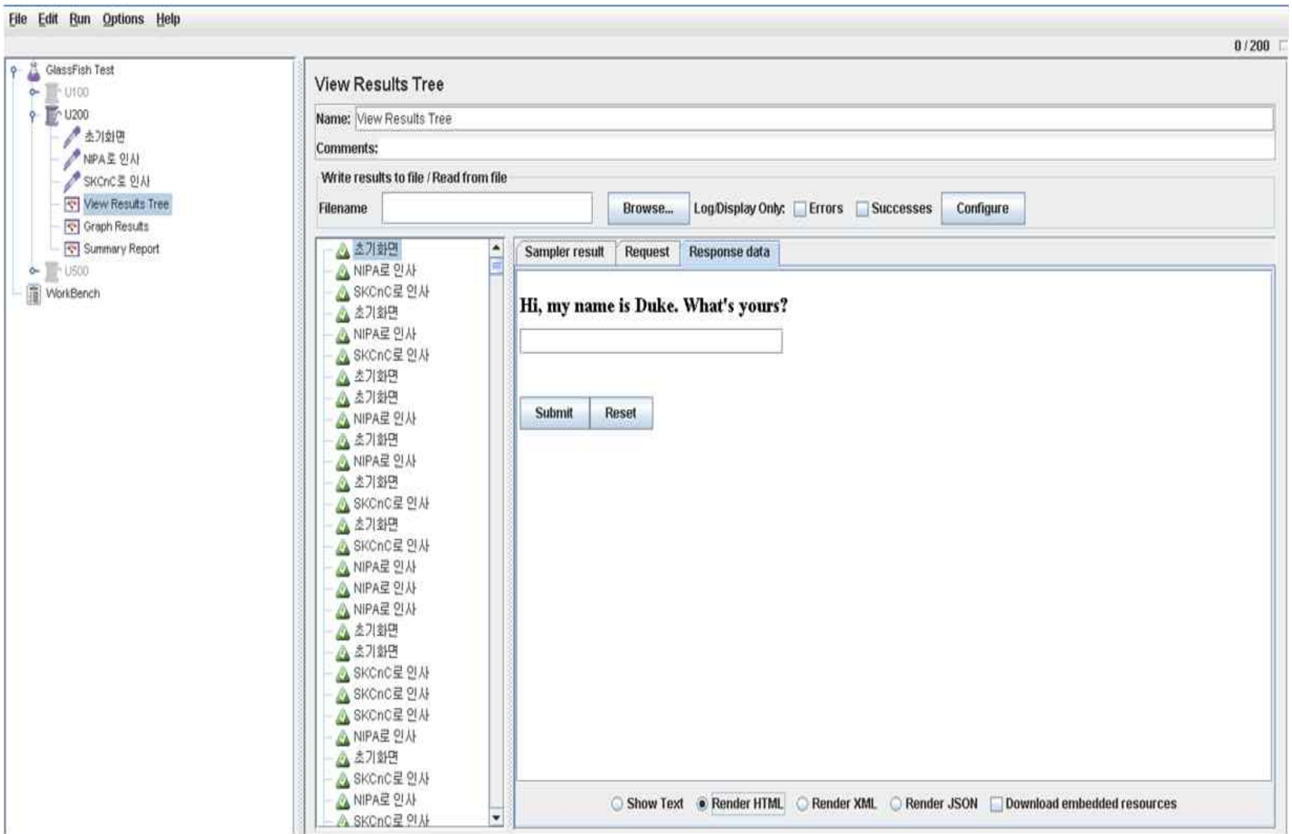
[그림4-5. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 300
- Average : 376
- Min : 1
- Max : 1464
- Error : 0.00%
- Throughput : 169.8/sec

나. 200 User가 동시 접속

① View Results Tree - 초기화면

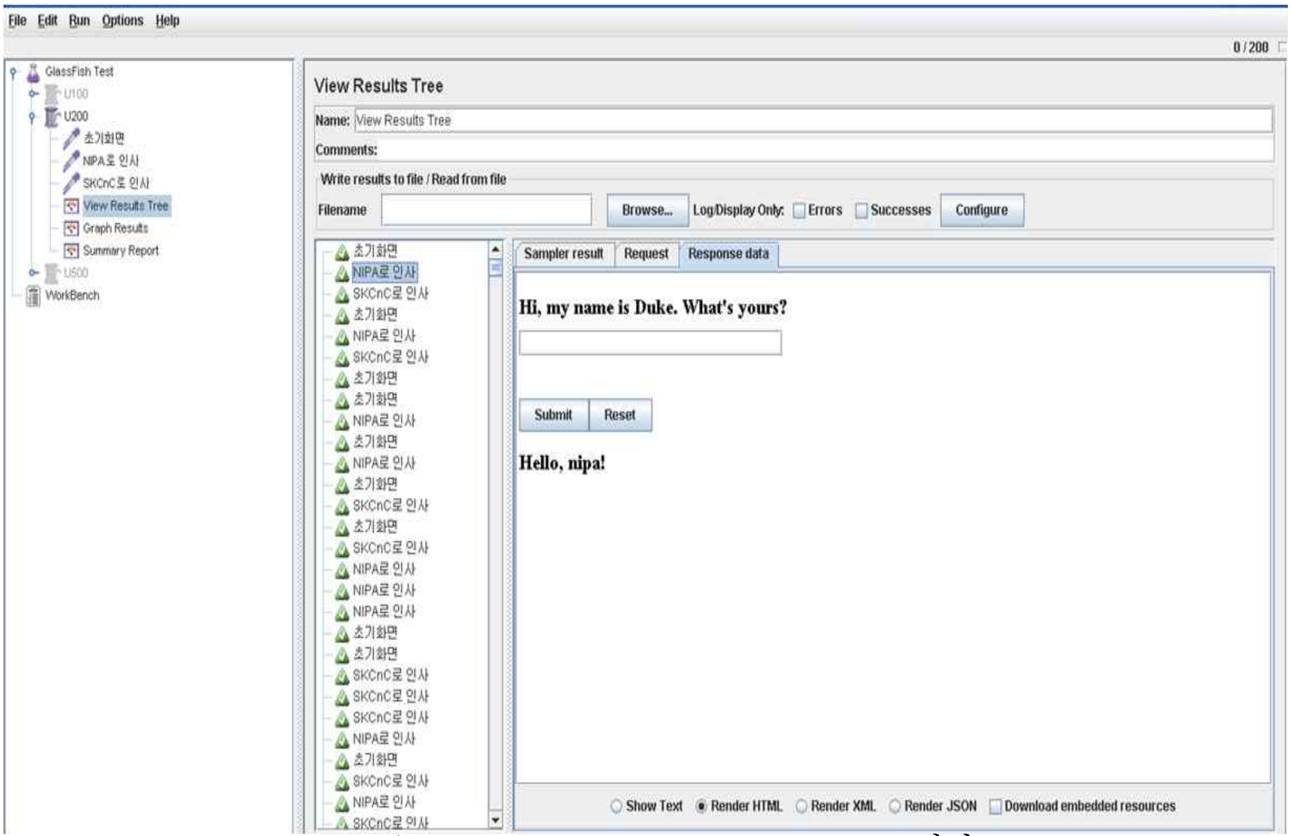


[그림4-6. View Results Tree - 초기화면]

방문자를 입력하기 위한 초기화면이다.

총 600번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

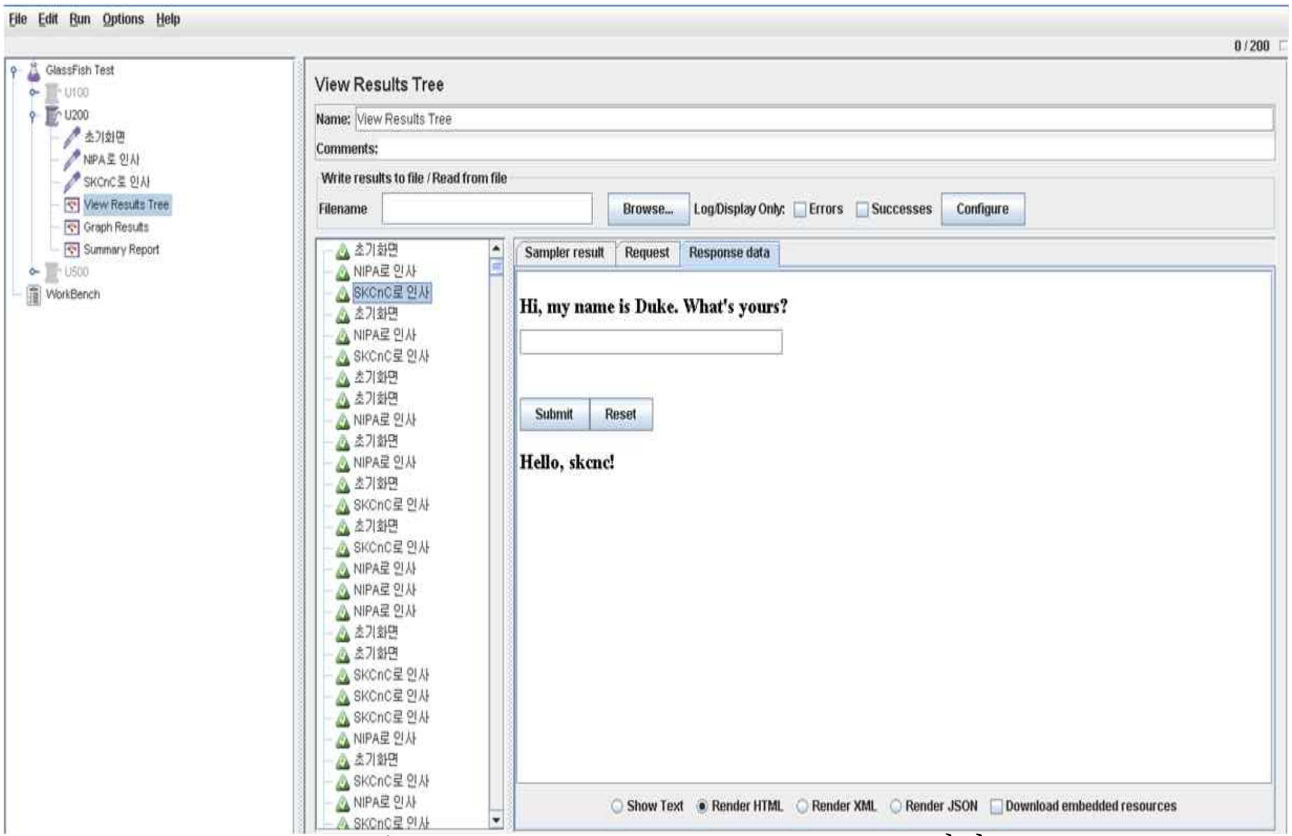
② View Results Tree - NIPA로 인사



[그림4-7. View Results Tree - NIPA로 인사]

NIPA로 인사 페이지를 정상적으로 표시하고 있음을 알 수 있다.

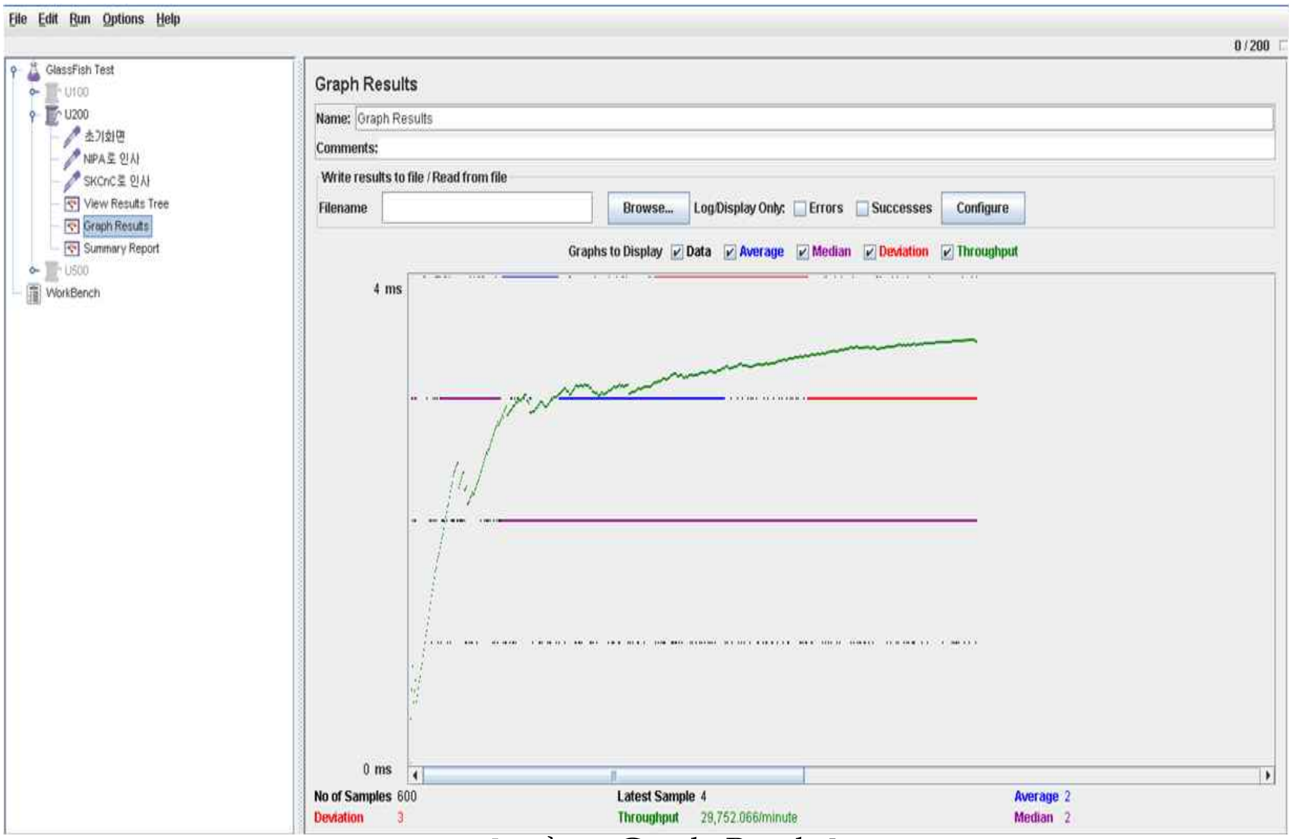
③ View Results Tree - SKCnC로 인사



[그림4-8. View Results Tree - SKCnC로 인사]

SKCnC로 인사 페이지를 정상적으로 표시하고 있음을 알 수 있다.

④ Graph Results

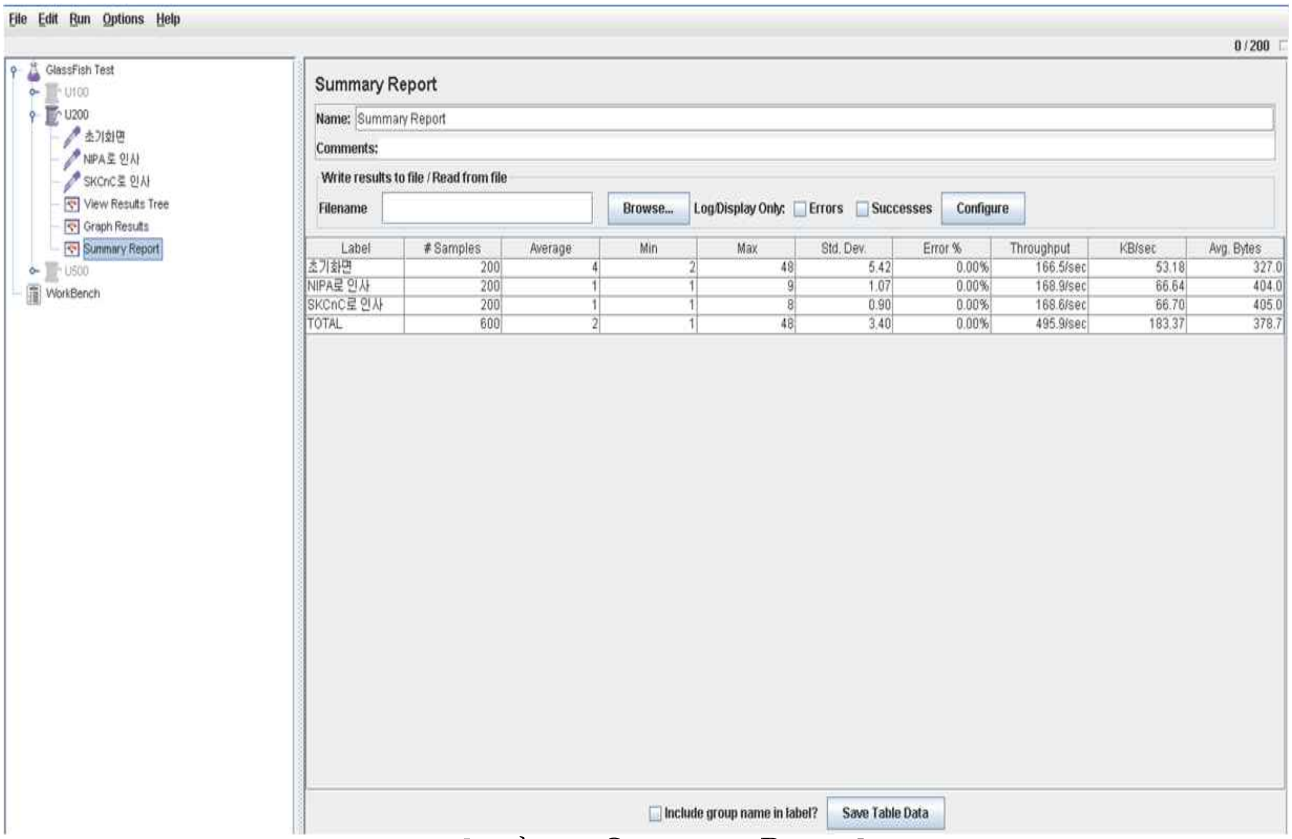


[그림4-9. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 600
- Average : 2
- Median : 2
- Deviation : 3
- Throughput : 29752.066/min

⑤ Summary Report



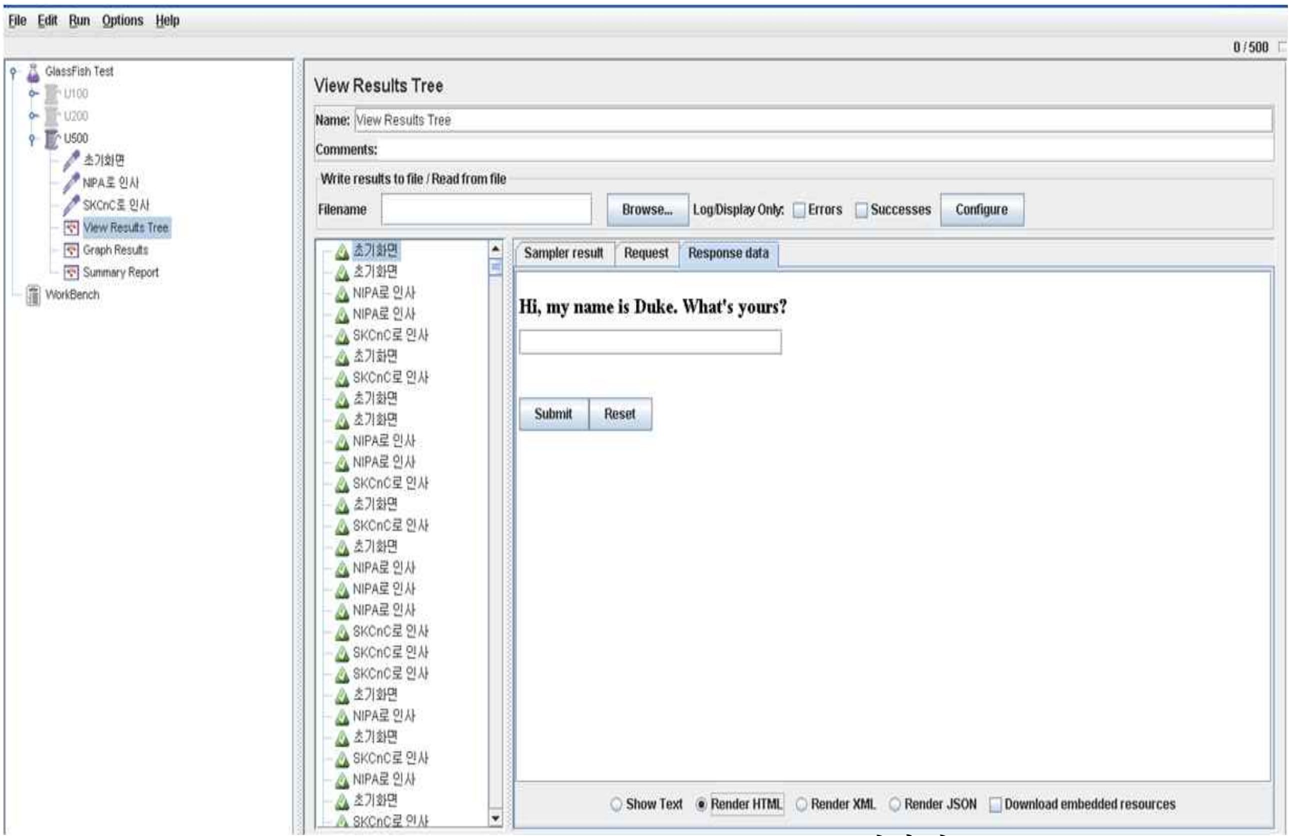
[그림4-10. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 600
- Average : 2
- Min : 1
- Max : 48
- Error : 0.00%
- Throughput : 495.9/sec

다. 500 User가 동시 접속

① View Results Tree - 초기화면

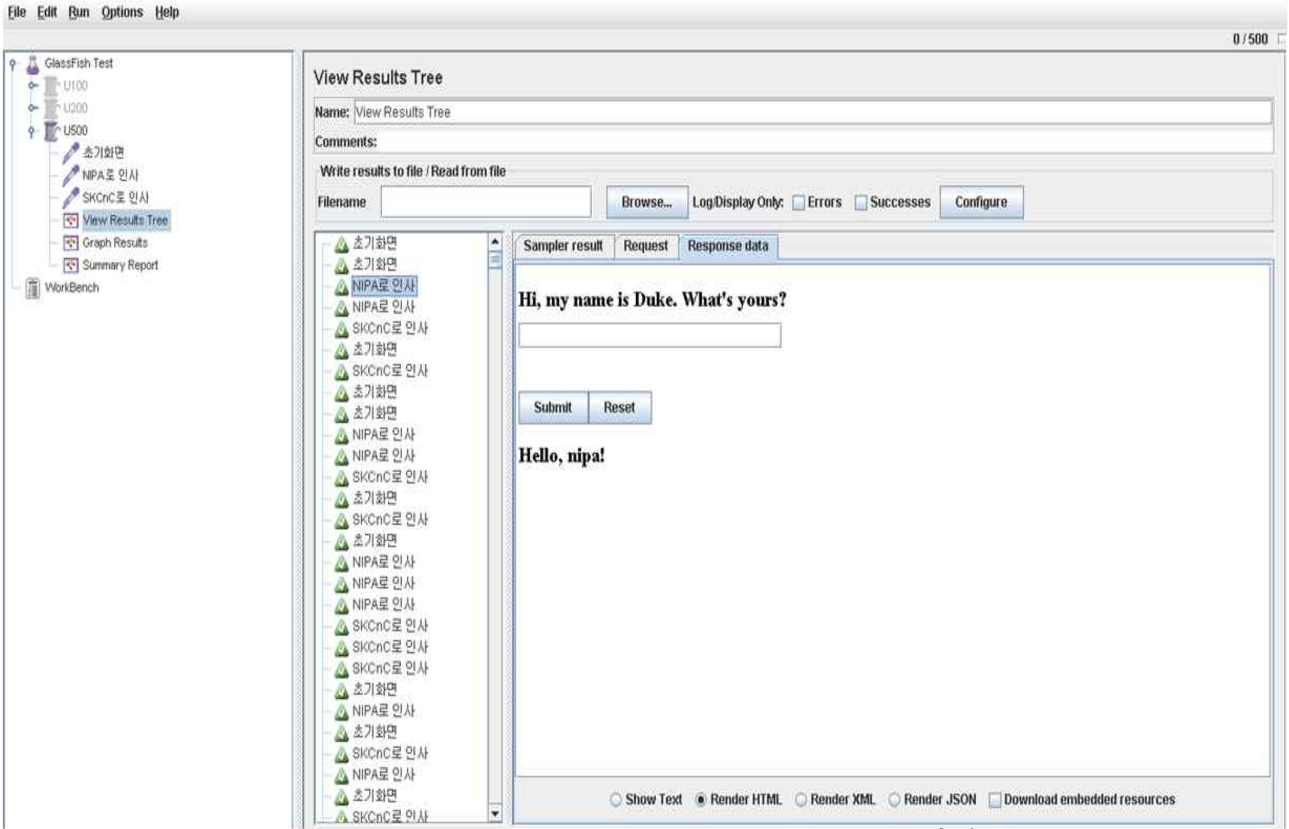


[그림4-11. View Results Tree - 초기화면]

방문자를 입력하기 위한 초기화면이다.

총 1500번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

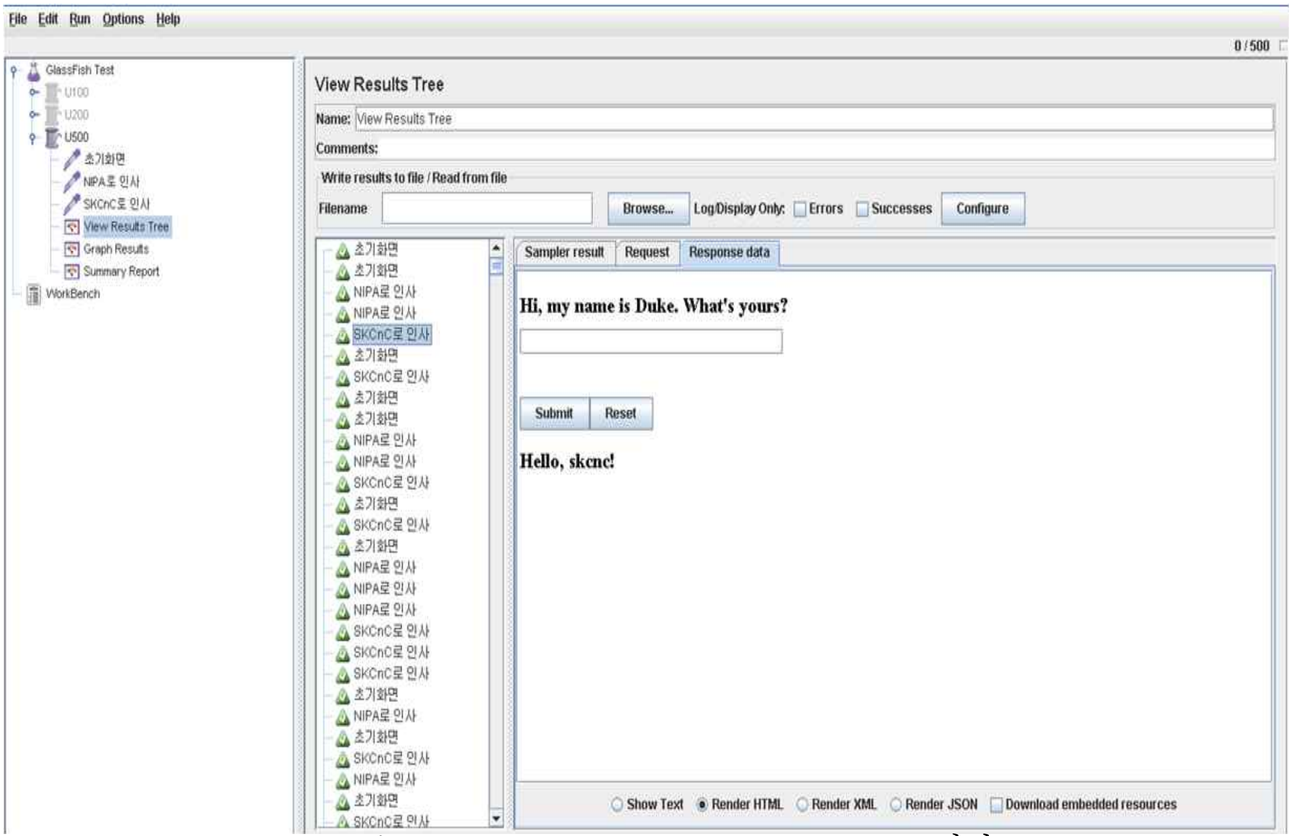
② View Results Tree - NIPA로 인사



[그림4-12. View Results Tree - NIPA로 인사]

NIPA로 인사 페이지를 정상적으로 표시하고 있음을 알 수 있다.

③ View Results Tree - SKCnC로 인사



[그림4-13. View Results Tree - SKCnC로 인사]

SKCnC로 인사 페이지를 정상적으로 표시하고 있음을 알 수 있다.

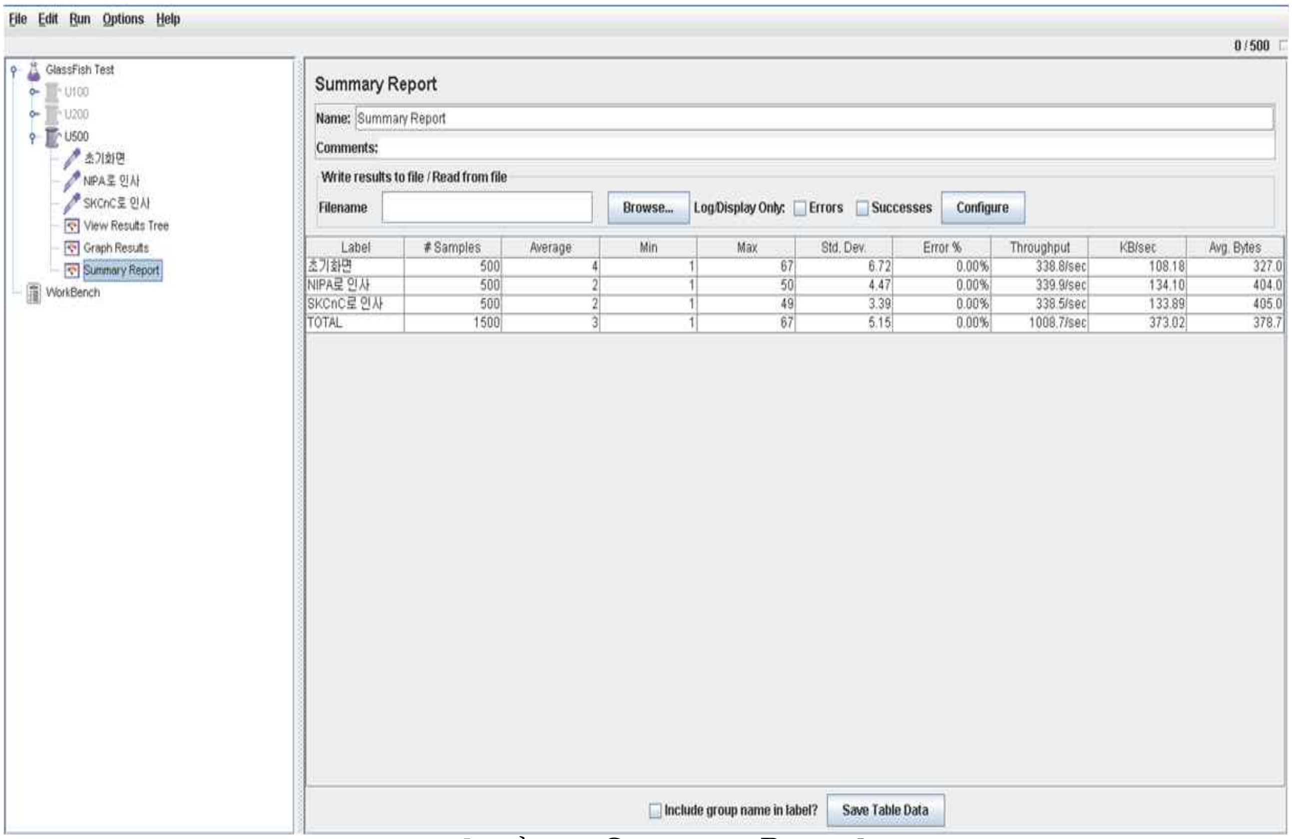
④ Graph Results

[그림4-14. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 1500
- Average : 3
- Median : 2
- Deviation : 5
- Throughput : 60524.546/min

⑤ Summary Report



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
초기화면	500	4	1	67	6.72	0.00%	338.8/sec	108.18	327.0
NIPA로 인사	500	2	1	50	4.47	0.00%	339.9/sec	134.10	404.0
SKCnC로 인사	500	2	1	49	3.39	0.00%	338.5/sec	133.89	405.0
TOTAL	1500	3	1	67	5.15	0.00%	1008.7/sec	373.02	378.7

[그림4-15. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 1500
- Average : 3
- Min : 1
- Max : 67
- Error : 0.00%
- Throughput : 1008.7/sec

5. 종합

Summary Report

Request	Average	Min	Max	Error	Throughput	KB/sec
300	376	1	1464	0.00%	169.8/sec	62.78
600	2	1	48	0.00%	495.9/sec	183.37
1500	3	1	67	0.00%	1008.7/sec	373.02

● 동시 접속 수 1500 요청에도 Throughput의 안정과 에러율의 0로서 시스템이 정상적으로 동작하는 것을 알 수 있다.

● 금번 테스트는 GlassFish 단독구성으로 하였으나 JVM, Web서비스, mod_jk 등의 튜닝으로 보다 나은 퍼포먼스의 향상을 기대할 수 있다.

Ⅷ JBoss

목 차

1. 개요

보편적으로 개발/구축된 Web시스템 중에는 기능사양에는 만족하고 있지만, 퍼포먼스라는 점에서는 사용자가 큰 불만을 품고 있는 사례가 흔하게 발생하고 있다. 그 이유로서는 부하테스트(성능테스트)가 거의 이루어지지 않는 프로젝트가 많았기 때문이다. 불특정다수의 사용자가 이용하는 web애플리케이션에 있어서는 만족할만한 편리한 기능이 갖추어져 있다고 하여도, 충분한 퍼포먼스가 나오지 않으면 사용자의 만족도가 저감하게 된다.

이에, 본 테스트 보고서에서는 부하테스트 툴인 JMeter를 사용하여, WAS(Web Application Server)의 대표적인 공개SW인 JBoss를 대상으로 부하테스트를 진행하고자 한다. 또한, 차세대 Web2.0을 구현하기 위한 애플리케이션 프레임워크인 JBoss Seam기반의 web 애플리케이션을 대상으로 테스트를 진행하고자 한다.

가. 목적

JBoss Seam 기반 어플리케이션의 부하를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 현재 구축된 시스템이 어느 정도의 부하에 견딜 수 있는가를 테스트한다.
- 상정한 동시 사용자수를 만족하는지 검증한다.
- 현재 시스템의 변경으로 성능향상의 정도를 측정한다.

나. 테스트 대상 및 테스트 툴의 선정기준

본 테스트에 있어서의 테스트 대상 및 테스트 툴을 선정함에 있어서 다음과 같은 특성을 열거한다.

1) 테스트 툴 - JMeter

JMeter는 Apache Jakarta Project에서 개발이 진행되고 있는 부하테스트 애플리케이션이다. Web애플리케이션(HTTP베이스의 애플리케이션)뿐만 아니라, FTP, JDBC, LDAP 등의 인터페이스 기반의 애플리케이션에 대하여 부하테스트를 실시하는 것이 가능하다.

Web애플리케이션의 부하테스트를 실시하는 도구로서는 JMeter 이외에도 OpenSTA, Mercury LoadRunner 등이 있지만, 다음의 관점으로부터 JMeterf를 사용하였다.

- 100% pure Java이다.
  Java로 기술되어 있으므로 OS환경에 구애받지 않는다.
- 오픈소스이다.
  오픈소스이므로 도입 비용이 거의 들지 않고, 커뮤니티도 활발하다.
- 조작이 간단하다.
  조작이 간단하고 CUI는 물론 GUI로 테스트를 실시할 수가 있다.

2) 테스트 대상 - JBoss

JBoss는 Java EE(Java Platform, Enterprise Edition) 어플리케이션 서버이다.
JBoss는 EJB3.0이나 JSF등의 최신 사양에 대응하고 있을 뿐만 아니라, JBoss Seam과 Hibernate의 프레임워크와 연계하여 높은 개발 생산성을 실현하고 있다.
JBoss는 ONJava 2008 Reader Survey에서, 가장 많이 사용되고 있는 J2EE 제품으로서 1위(33%)를 획득하고 있다.

JBoss는 오픈 소스 제품이면서 애플리케이션 서버에 요구되는 높은 기능 요건을 만족하고 있다.
JBoss는 차세대 애플리케이션 프레임워크와 가장 대중적인 클러스터 Java EE 서버 애플리케이션 서버의 통합을 통해 엔터프라이즈 급의 안정성에 대한 혁신적인 균형을 맞추고 있다.

- J2EE1.4이상 준거
- 생산성이 높은 프레임워크
- 클러스터링 기능
- 기업 레벨의 서포트
- 비용 절감

3) 테스트 대상 - JBoss Seam

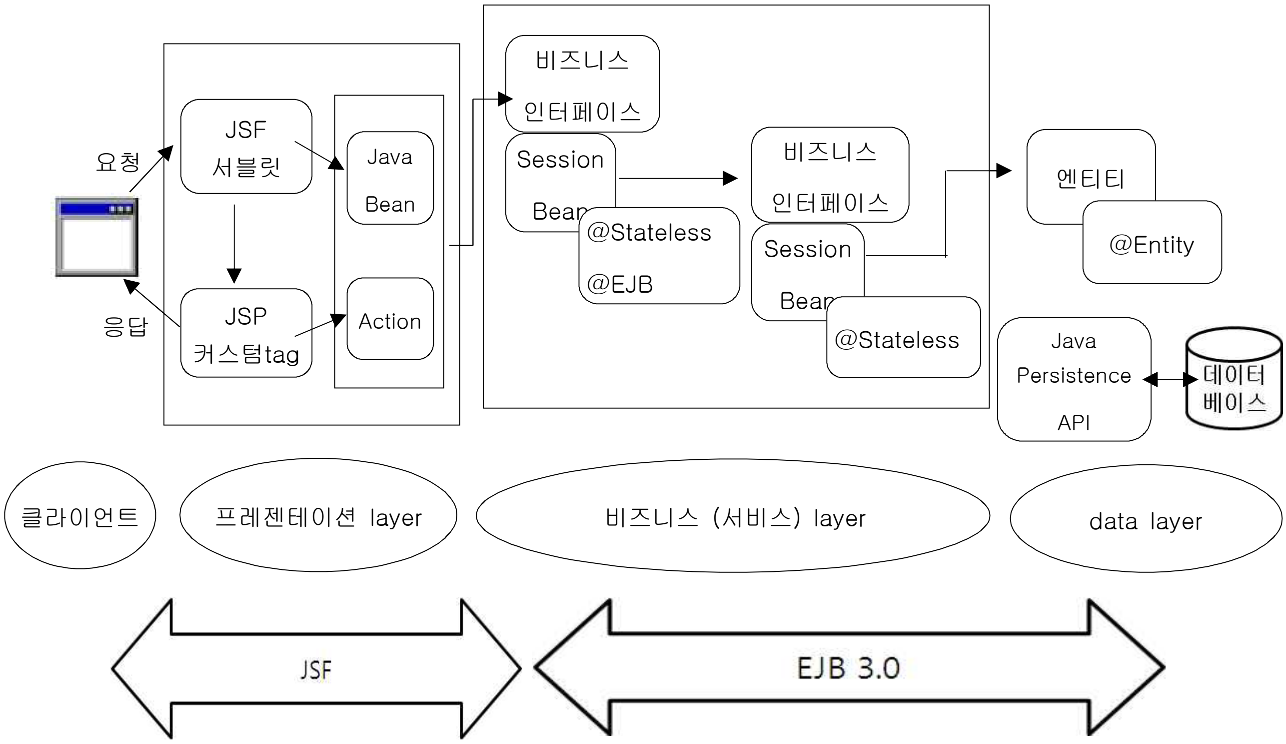
JBoss Seam는 복잡화하는 어플리케이션 개발을 심플하게 바꾸는 차세대 표준프레임워크이다.
Java Server Faces(JSF)에 의한 화면으로부터 Enterprise Java Beans(EJB)의 비즈니스 로직을 풀로 접착하듯이 간단하게 연계 가능하다. 또, AJAX와의 연계나, BPM이나 포탈 프레임워크와의 연계, 한층 더 Enterprise Service Bus(ESB)에 의한 외부 시스템과의 연계까지, 다양한 연계가 가능하다.

- EJB 3.0을 적용한 최초의 애플리케이션 프레임워크
- J2EE 지원
- AJAX 구현용이
- 혁신적인 상태 관리 기능
- 화면 흐름의 관리 가능 (jBPM)
- 통합 테스트의 용이 (JUnit or TestNG)

다. 전제조건

- 본 부하 테스트는 Hardware 및 네트워크에 관련된 모니터링에 대해서는 언급하지 않음
- 본 부하 테스트는 JMeter를 사용한 JBoss Seam기반의 애플리케이션의 성능을 검증하기 위한 목적이므로 테스트 결과에 대한 분석 및 반영으로 의한 성능향상 방법은 별개의 과제로 한다.

라. 참고사항(JBoss Seam Application 구성도)



[그림1-1. JSF + EJB에 의한 Web 애플리케이션]

2. 테스트 환경

OS	Linux (2.6.18)
테스트 대상	JBoss / jboss-5.1.0.GA
애플리케이션	Blog(RESTful Seam 애플리케이션)
테스트 툴	JMeter

3. 테스트 절차

가. 소프트웨어 설치

1) 서버

(1) JDK

> JAVA_HOME=/usr/local/java/jdk1.5.0_21

(2) JBoss

> JBOSS_HOME=/usr/local/jboss/jboss-5.1.0.GA

> <JBOSS_HOME>/bin/run.sh에 JAVA_HOME 환경변수 추가

```
#!/bin/sh
..... 생략
JAVA_HOME=/usr/local/java/jdk1.5.0_21
..... 생략
```

> JBoss 기동

```
#run.sh -b 121.162.249.17 <- 외부접속 허용하기 위한 IP 바인딩
```

2) 클라이언트

(1) JDK

> JAVA_HOME=/usr/local/java/jdk1.5.0_21

(2) JMeter

> JMeter_Manual.pdf 참조

나. 테스트 시나리오

1) 시나리오

(1) 초기화면을 표시한다.

(2) 북마크 aaa로 검색한다.

(3) 북마크 bbb로 검색한다.

2) 부하레벨의 설정

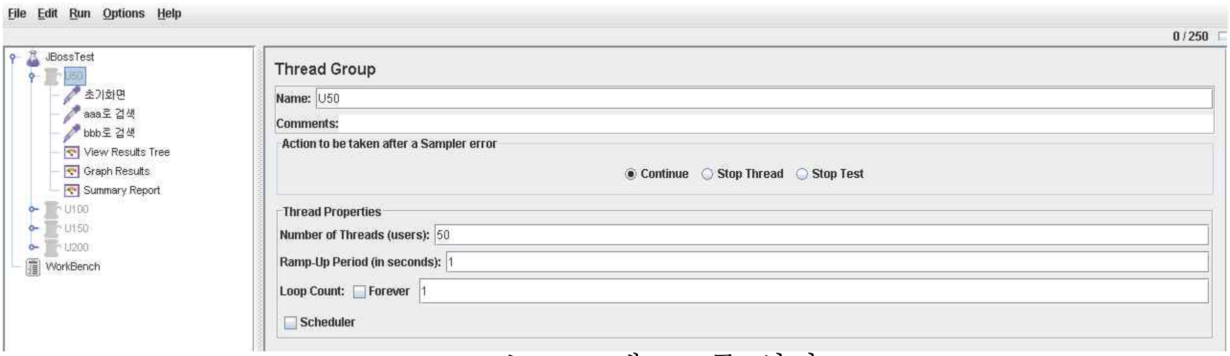
시스템의 부하레벨을 높여 가면서 테스트를 반복하며, 한계치를 조사하기 위하여, Ramp-Up기간을 1sec 고정으로 loop counter를 1로 하여 thread수(접속자수)를 증가시켜 1초당 동시 접속수를 조정한다. 금번 구축한 시스템에서의 성능저하를 확인할 수 있는 동시 접속자수=100 부근의 쓰레드그룹을 작성한다.

3) JMeter 설정

(1) 테스트 플랜 작성

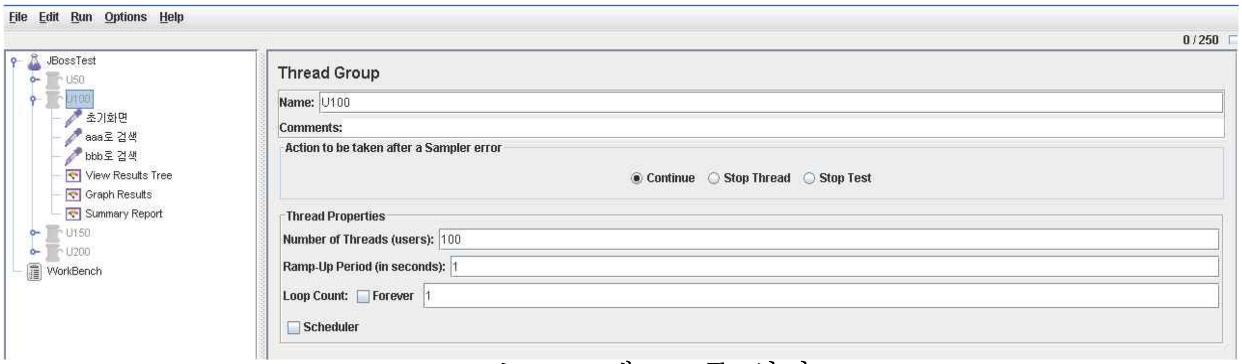
(2) Thread Group 생성

● 50 User 동시 접속



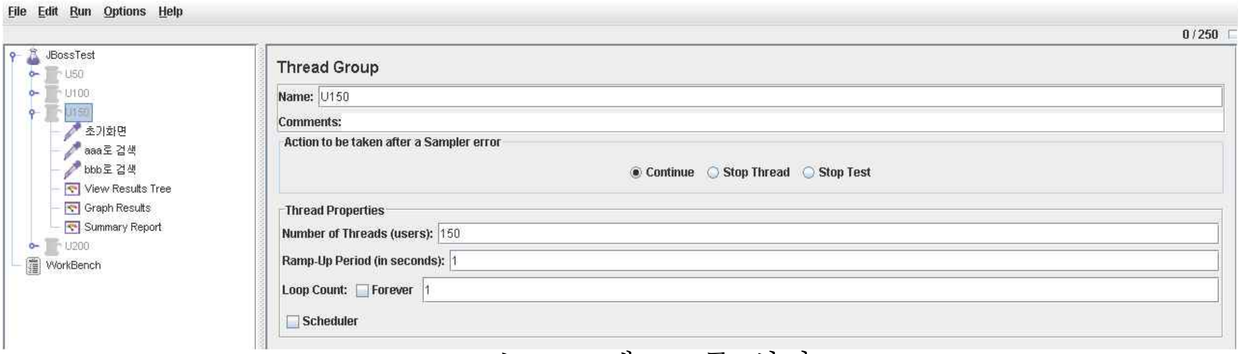
[그림3-1. 쓰레드그룹 설정 - 50 User]

● 100 User 동시 접속



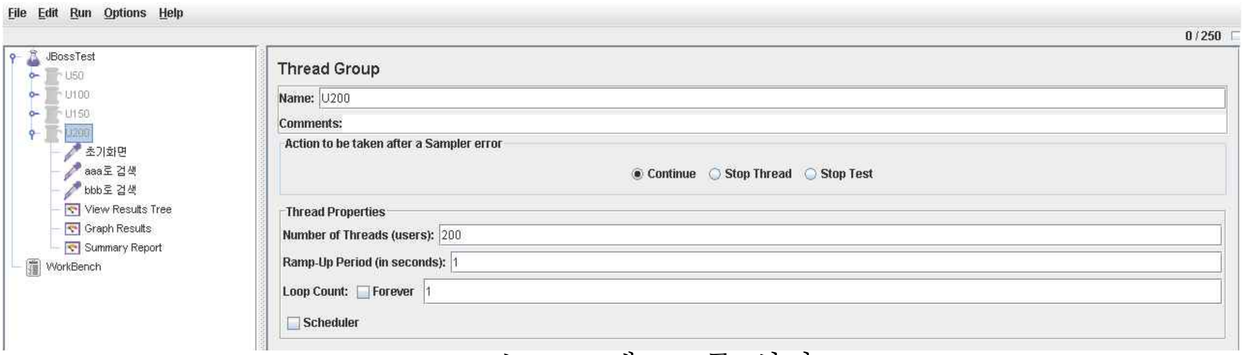
[그림3-2. 쓰레드그룹 설정 - 100 User]

● 150 User 동시 접속



[그림3-3. 쓰레드그룹 설정 - 150 User]

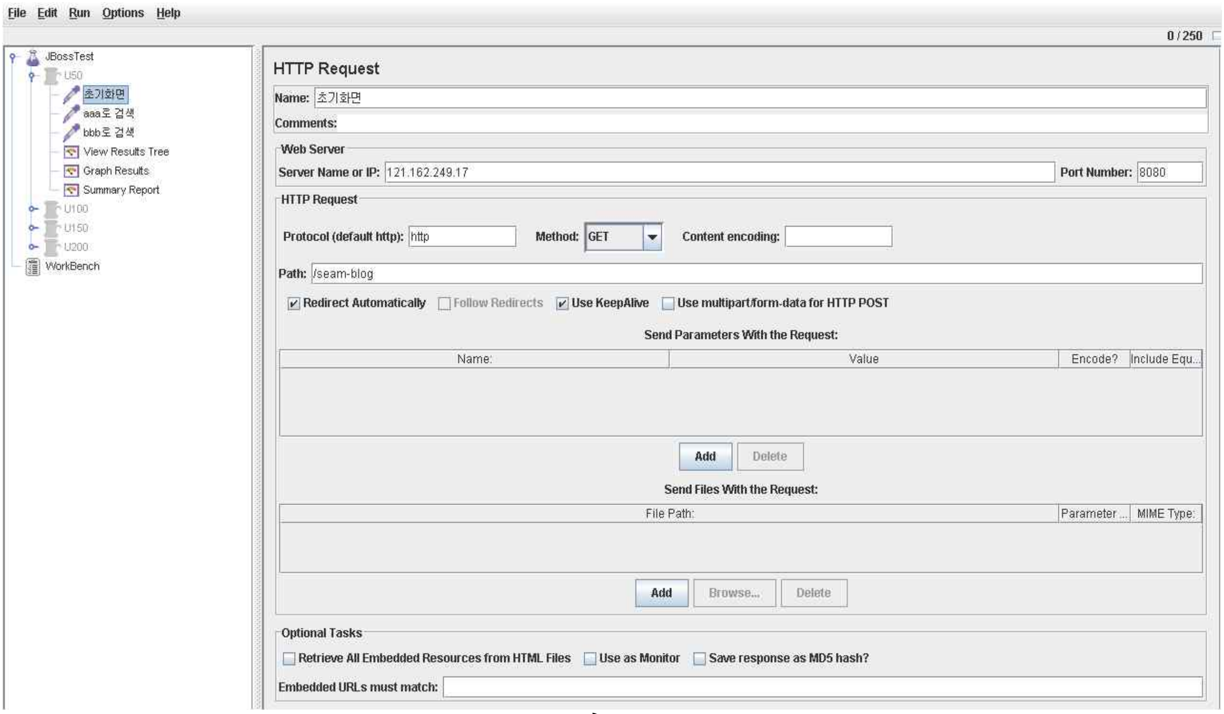
● 200 User 동시 접속



[그림3-4. 쓰레드그룹 설정 - 200 User]

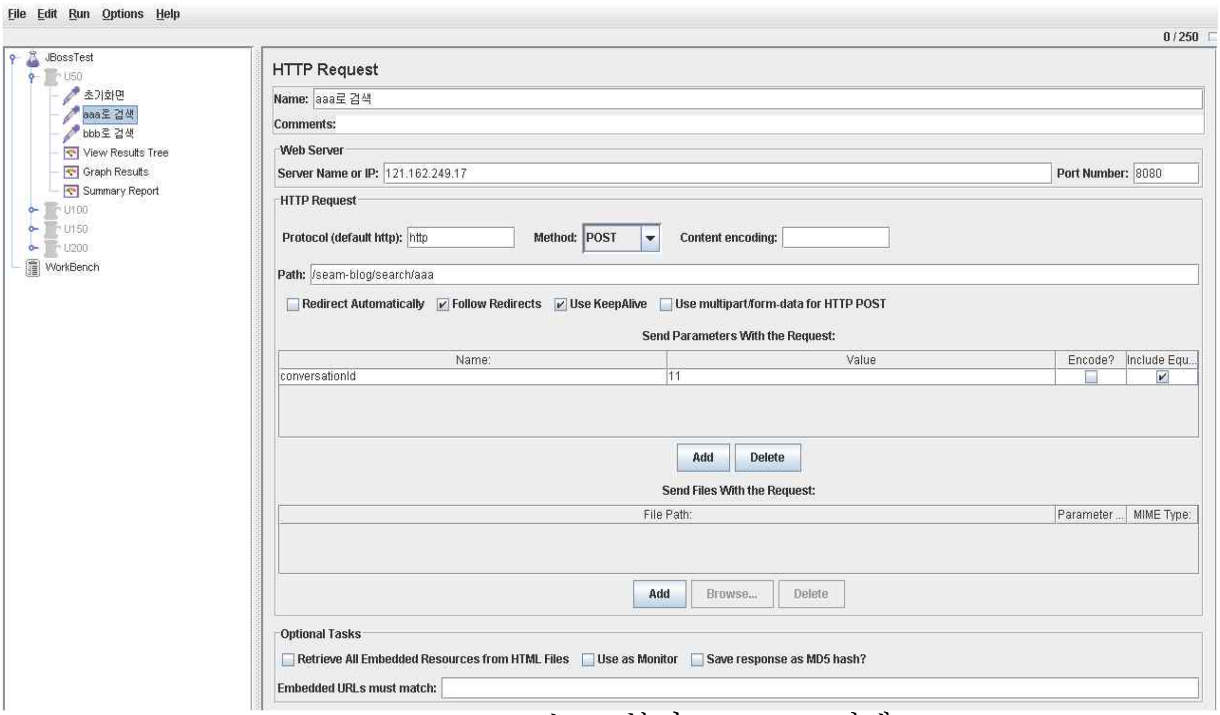
(3) Http Request 추가 (각 Thread Group에 공통)

● 초기화면



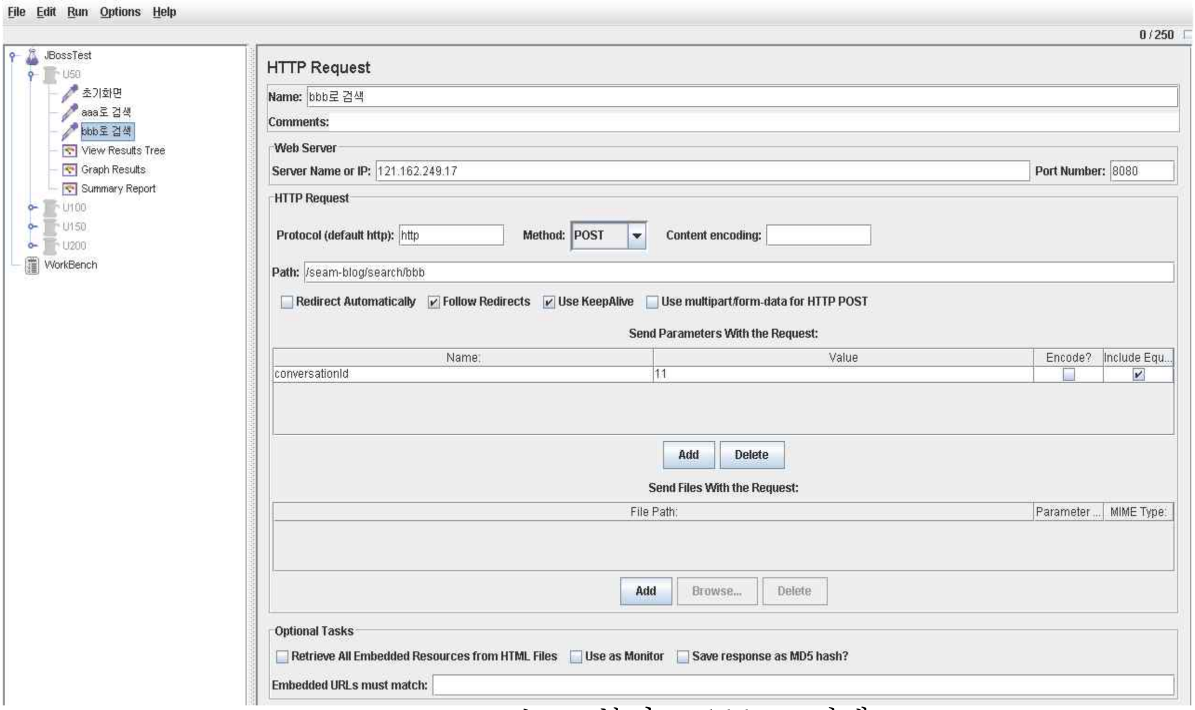
[그림3-5. Http Request]

● 북마크 aaa로 검색



[그림3-6. 북마크 aaa로 검색]

● 북마크 bbb로 검색



[그림3-7. 북마크 bbb로 검색]

(4) Listener 추가 (각 Thread Group에 공통)

● View Results Tree

● Graph Results

● Summary Report

다. 테스트 실시

(1) JBoss의 기동
- JBOSS_HOME/bin 디렉터리에서
  > ./run.sh -b 0.0.0.0 실행

(2) 테스트 대상 seam 애플리케이션을 Deploy
- /usr/local/jboss/jboss-seam-2.2.0.GA/examples/blog 디렉토리에서
  > ant 실행

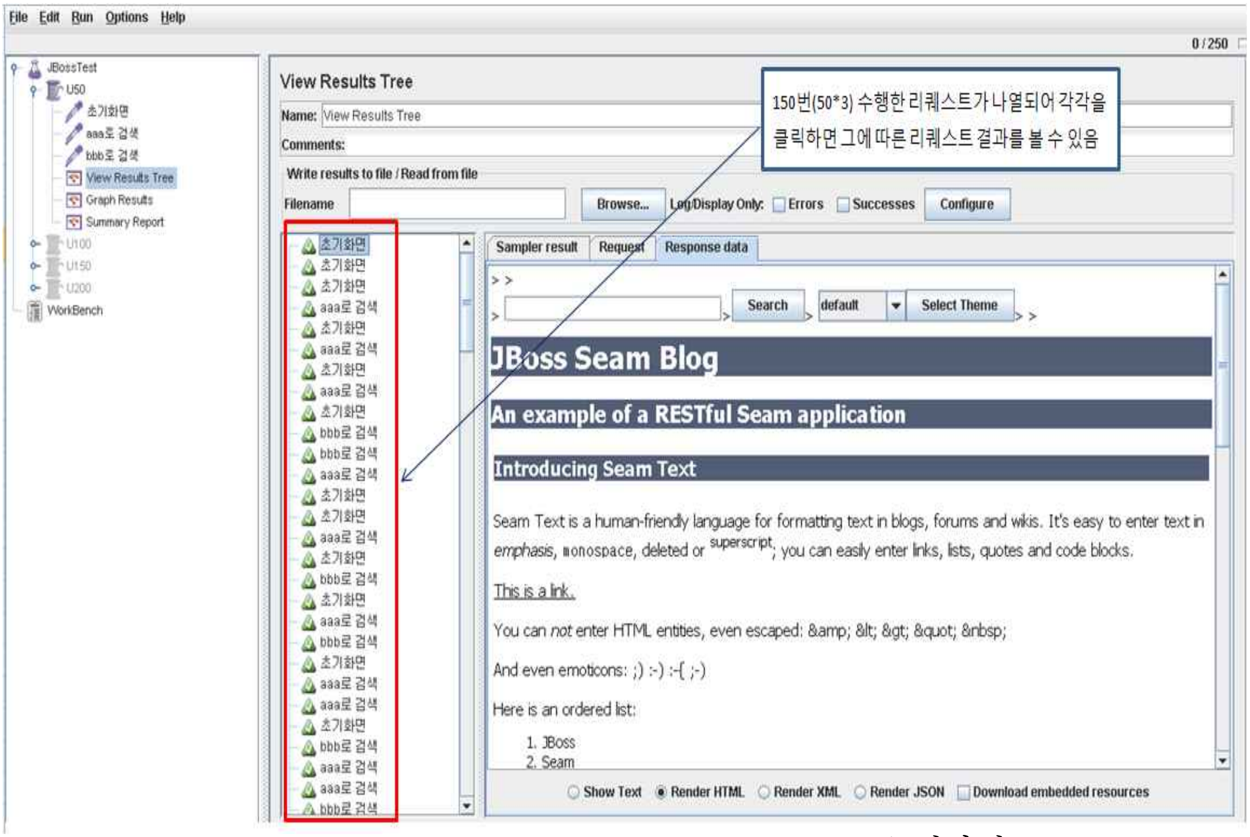
(3) Jmeter의 Run > Start

4. 테스트 결과

테스트 결과는
① View Results Tree - 초기화면
② View Results Tree - 북마크 aaa로 검색
③ View Results Tree - 북마크 bbb로 검색
④ Graph Results
⑤ Summary Report
의 Listener로 확인한다.

가. 50 User 동시 접속

① View Results Tree - 초기화면

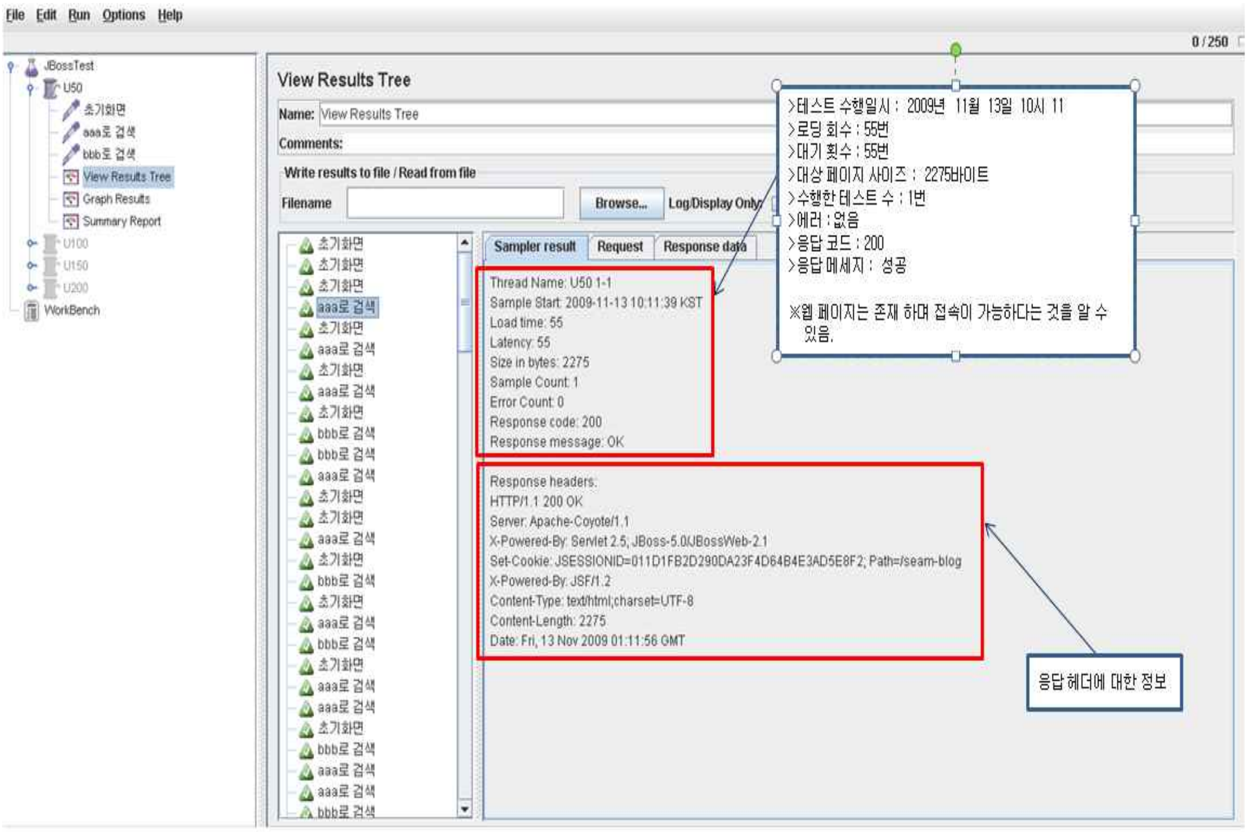


[그림4-1. View Results Tree - 초기화면]

블로그의 북마크를 검색하기 위한 초기화면이다.

총 150번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

② View Results Tree - 북마크 aaa로 검색



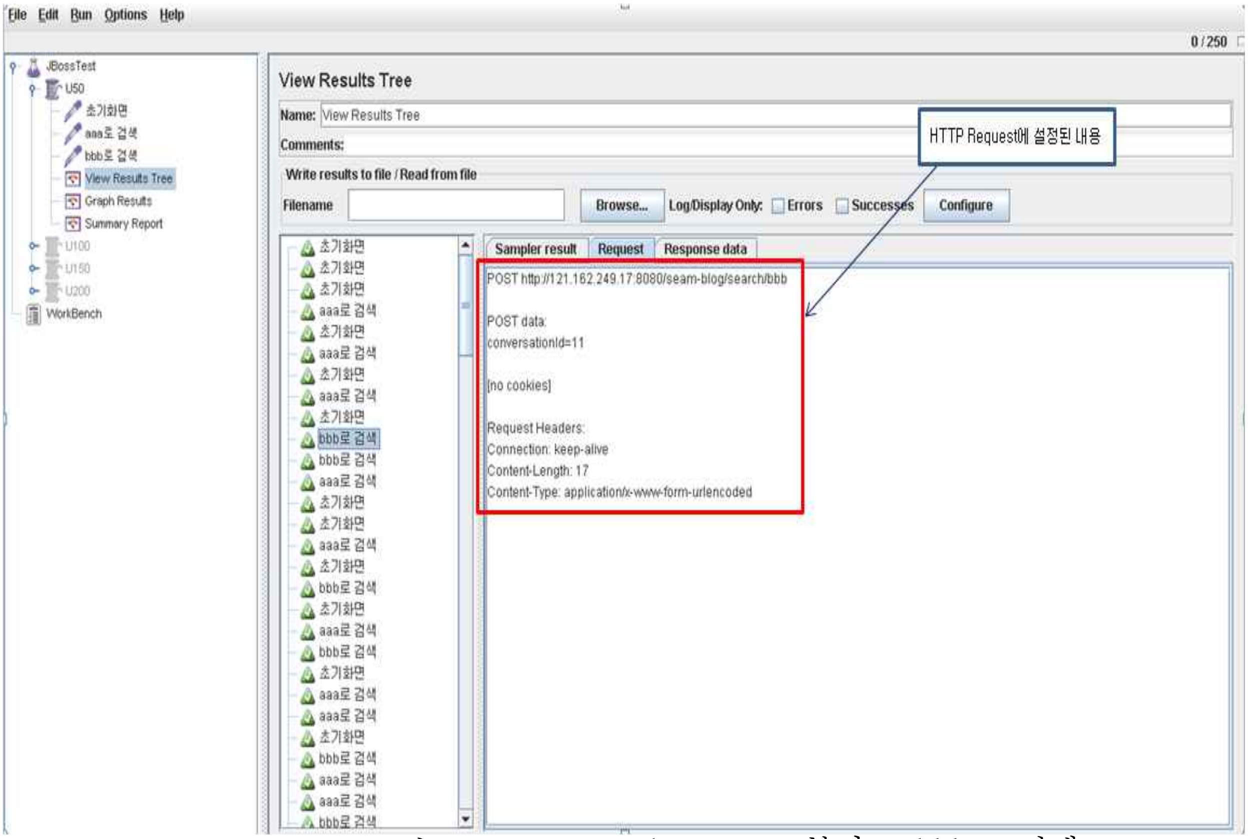
[그림4-2. View Results Tree - 북마크 aaa로 검색]

북마크 aaa로 검색한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

- 로딩 회수 : 55번
- 대기 횟수 : 55번
- 대상 페이지 사이즈 : 2275바이트
- 수행한 테스트 수 : 1번
- 에러 : 없음
- 응답 코드 : 200
- 응답 메시지 : 성공

③ View Results Tree - 북마크 bbb로 검색

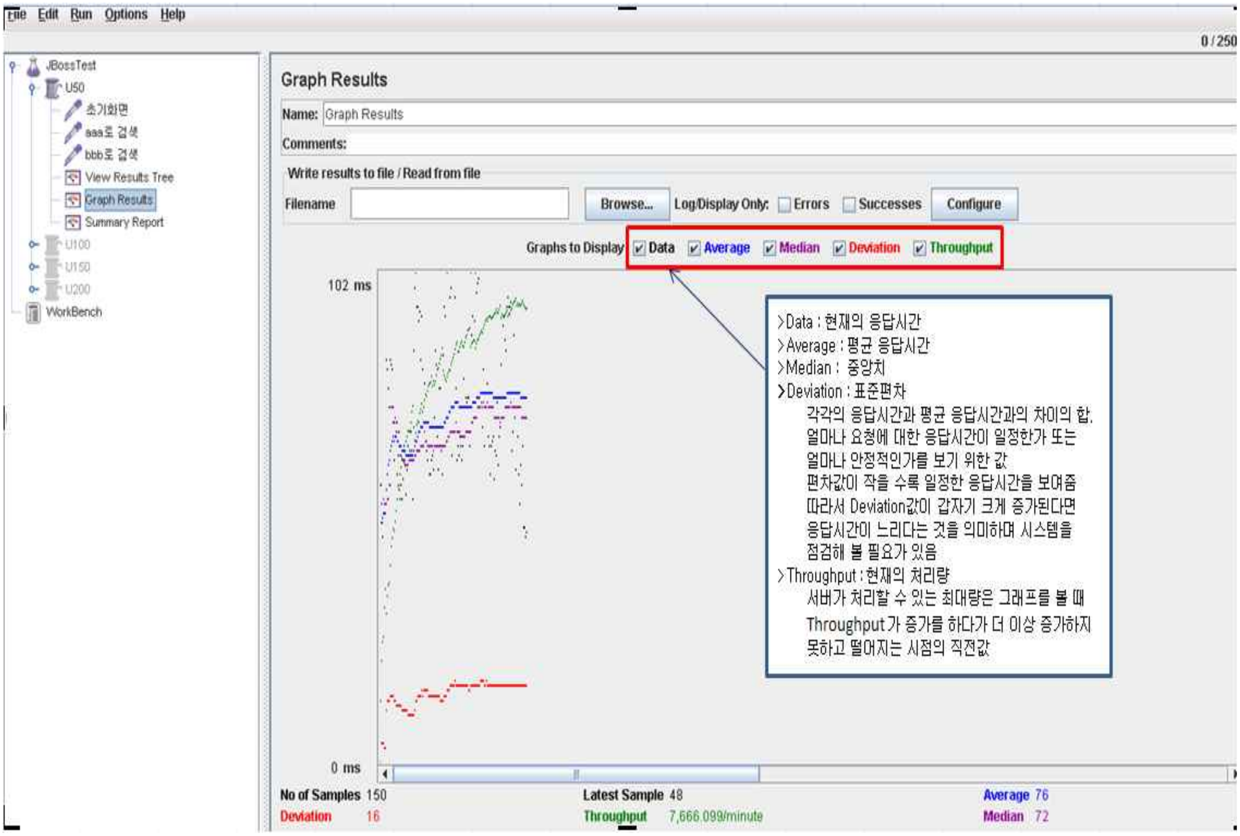


[그림4-3. View Results Tree - 북마크 bbb로 검색]

북마크 bbb로 검색한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

④ Graph Results

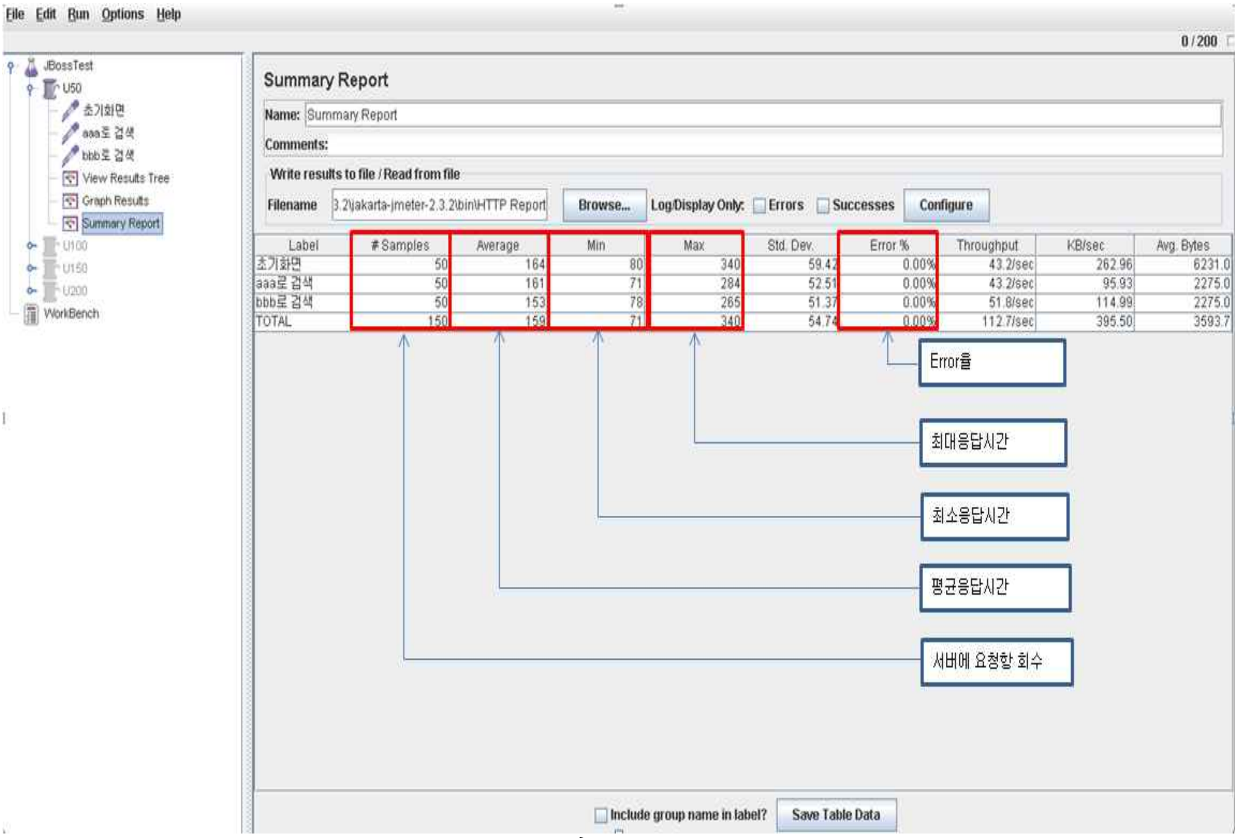


[그림4-4. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 150
- Average : 76
- Median : 72
- Deviation : 16
- Throughput : 7666.099/min

⑤ Summary Report



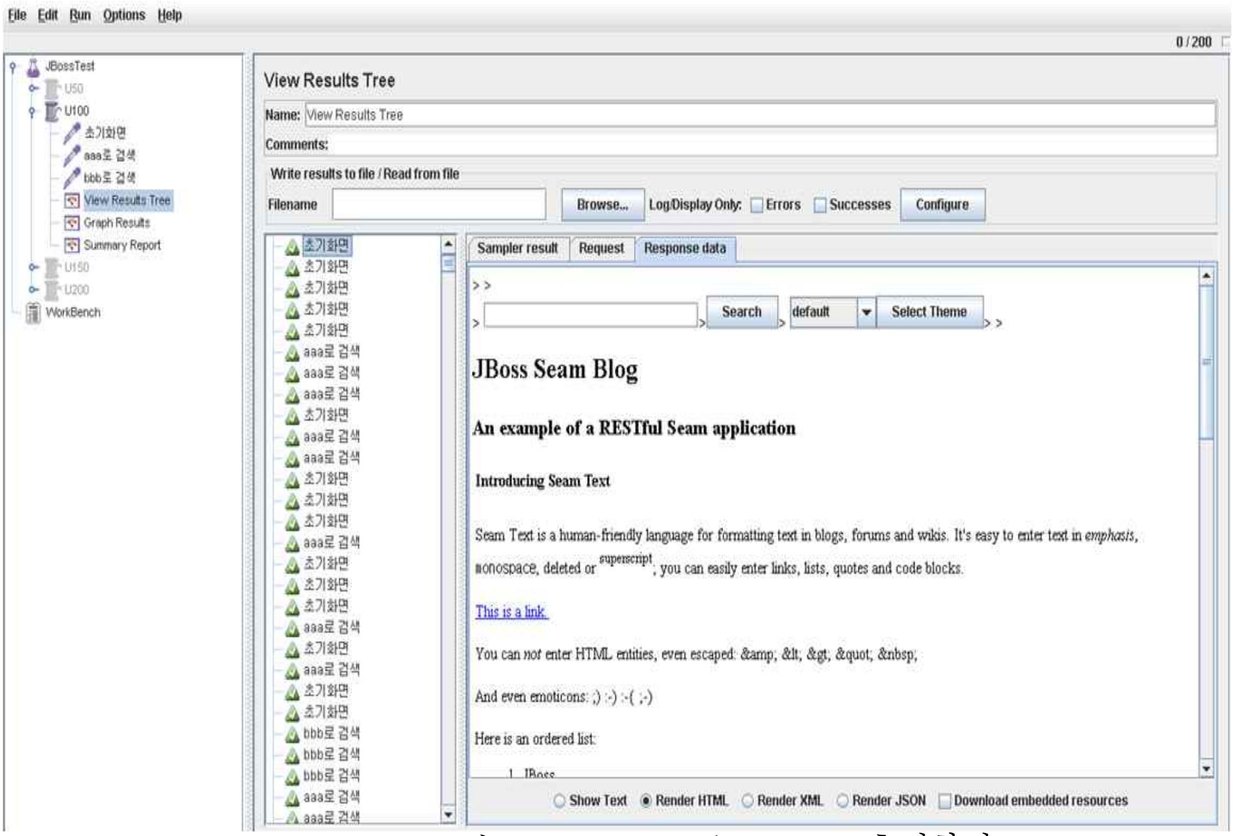
[그림4-5. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 150
- Average : 159
- Min : 71
- Max : 340
- Error : 0.00%
- Throughput : 112.7/sec

나. 100 User 동시 접속

① View Results Tree - 초기화면

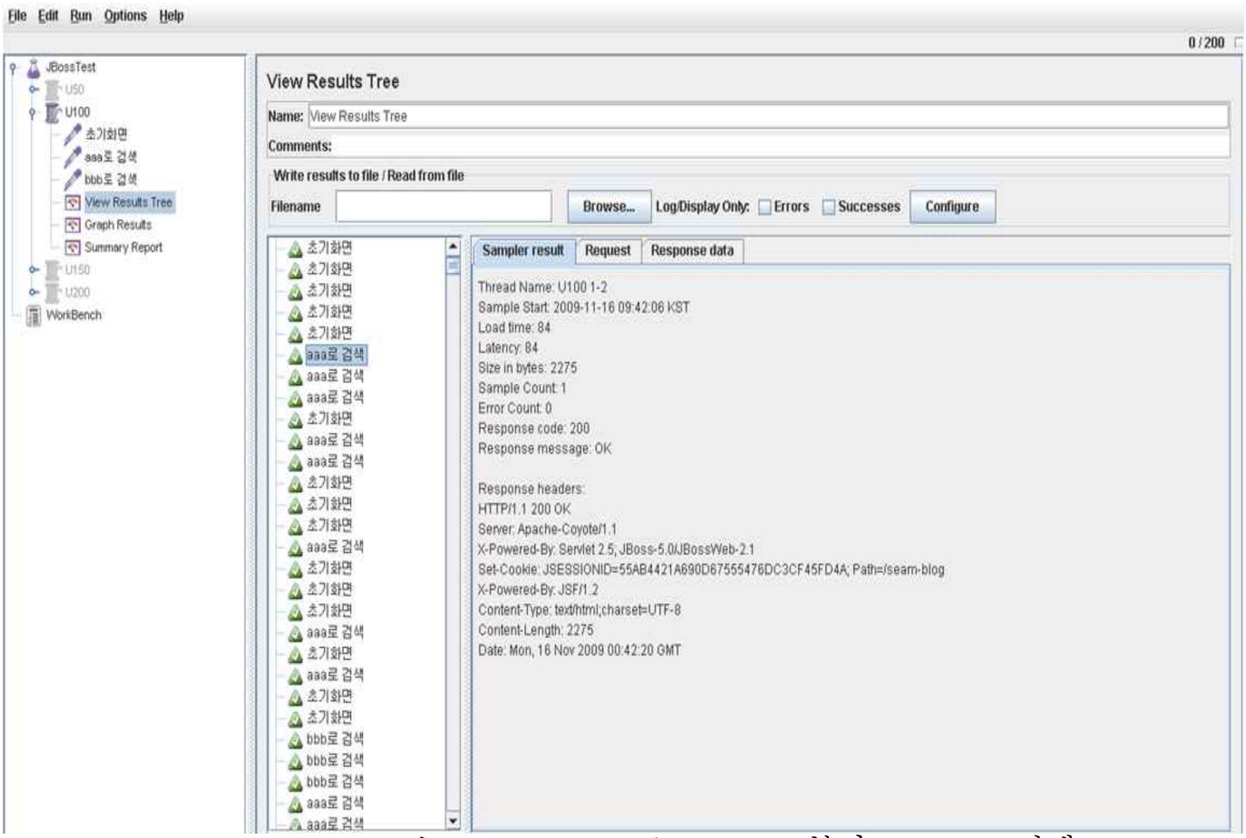


[그림4-6. View Results Tree - 초기화면]

블로그의 북마크를 검색하기 위한 초기화면이다.

총 300번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

② View Results Tree - 북마크 aaa로 검색



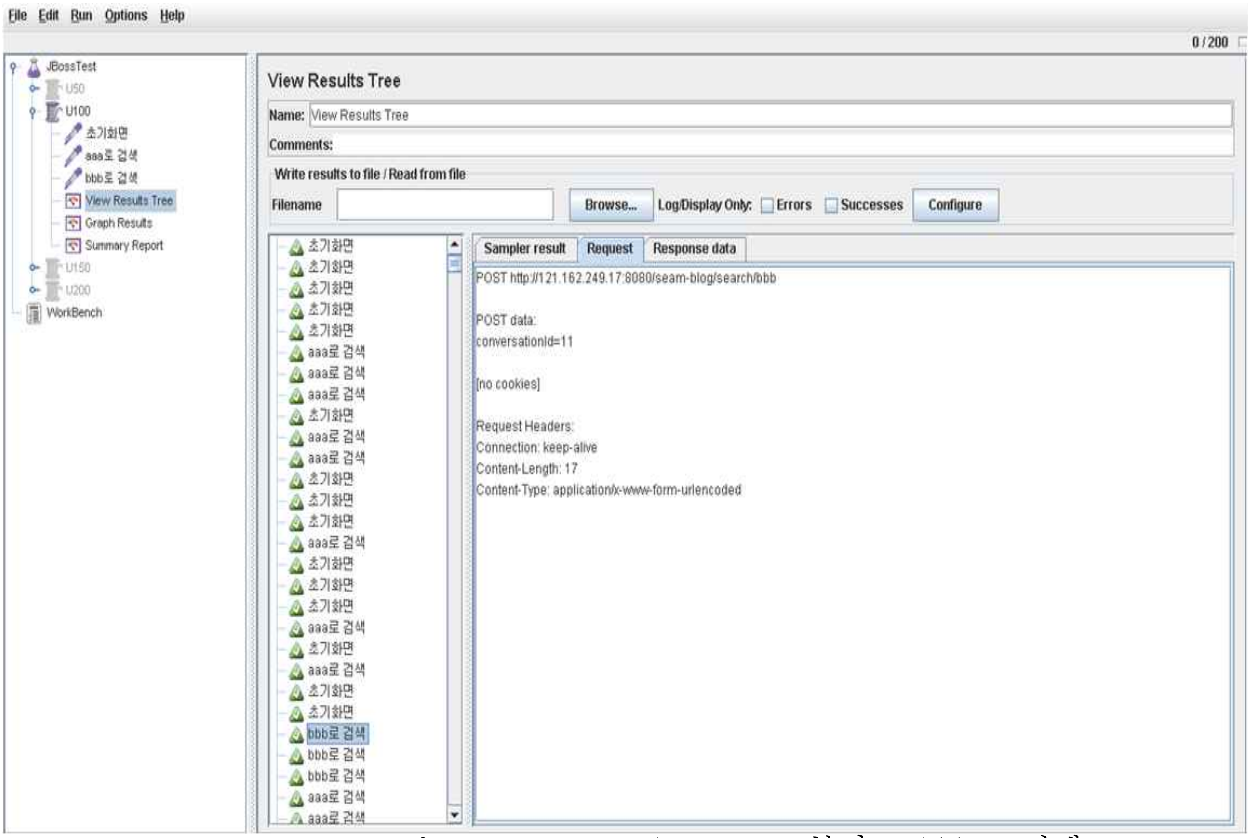
[그림4-7. View Results Tree - 북마크 aaa로 검색]

북마크 aaa로 검색한 상세 정보를 알 수 있다.

각 요청(초기화면, aaa로 검색, bbb로 검색)에 대한 정상적인 결과를 알 수 있다.

- 로딩 회수 : 84번
- 대기 횟수 : 84번
- 대상 페이지 사이즈 : 2275바이트
- 수행한 테스트 수 : 1번
- 에러 : 없음
- 응답 코드 : 200
- 응답 메시지 : 성공

③ View Results Tree - 북마크 bbb로 검색

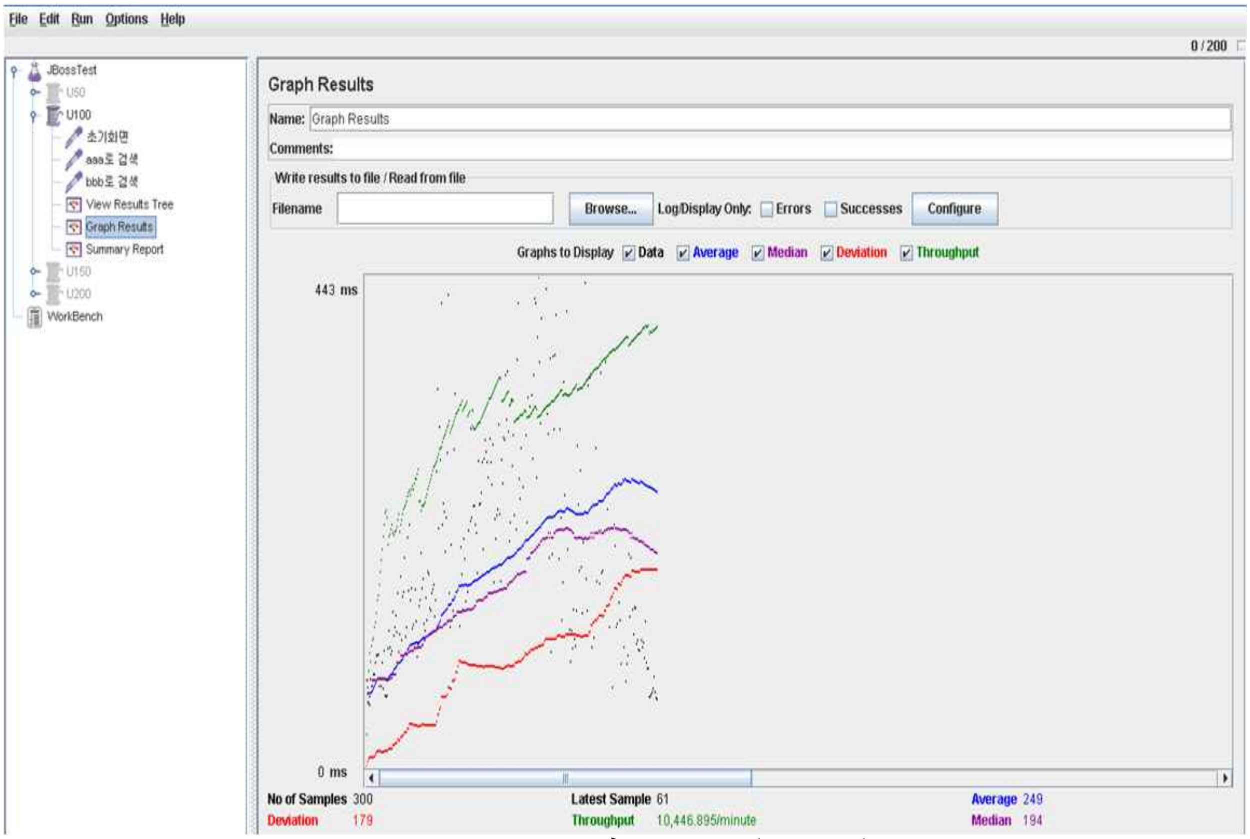


[그림4-8. View Results Tree - 북마크 bbb로 검색]

북마크 bbb로 검색한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

④ Graph Results

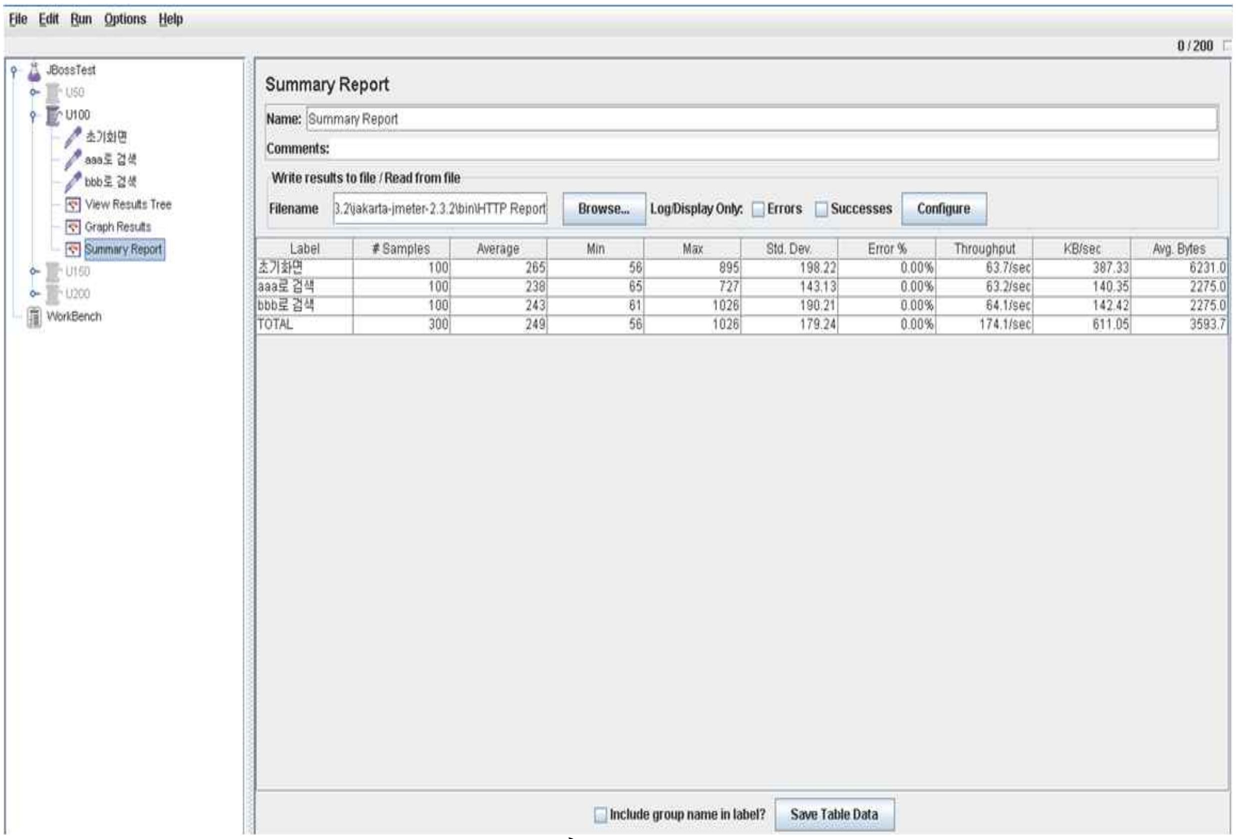


[그림4-9. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 300
- Average : 249
- Median : 194
- Deviation : 179
- Throughput : 10446.895/min

⑤ Summary Report



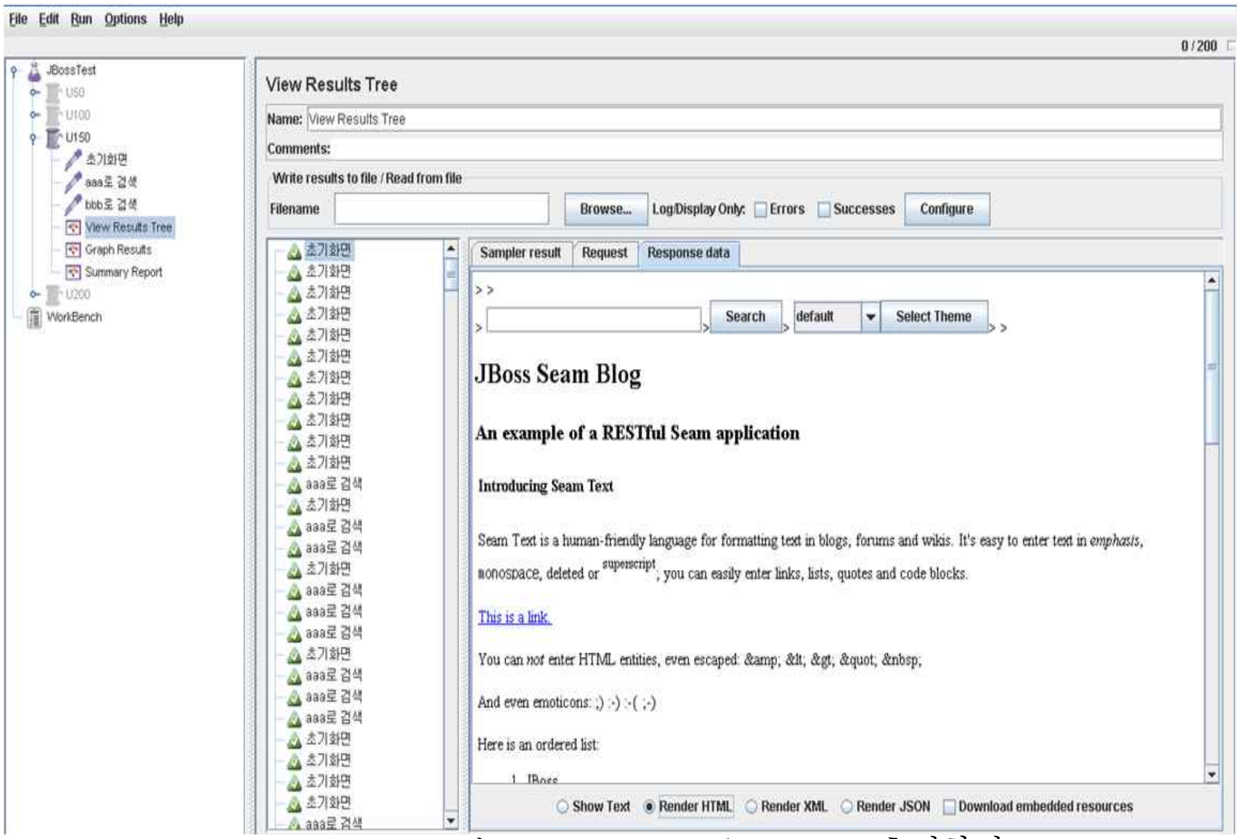
[그림4-10. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 300
- Average : 249
- Min : 56
- Max : 1026
- Error : 0.00%
- Throughput : 174.1/sec

다. 150 User 동시 접속

① View Results Tree - 초기화면

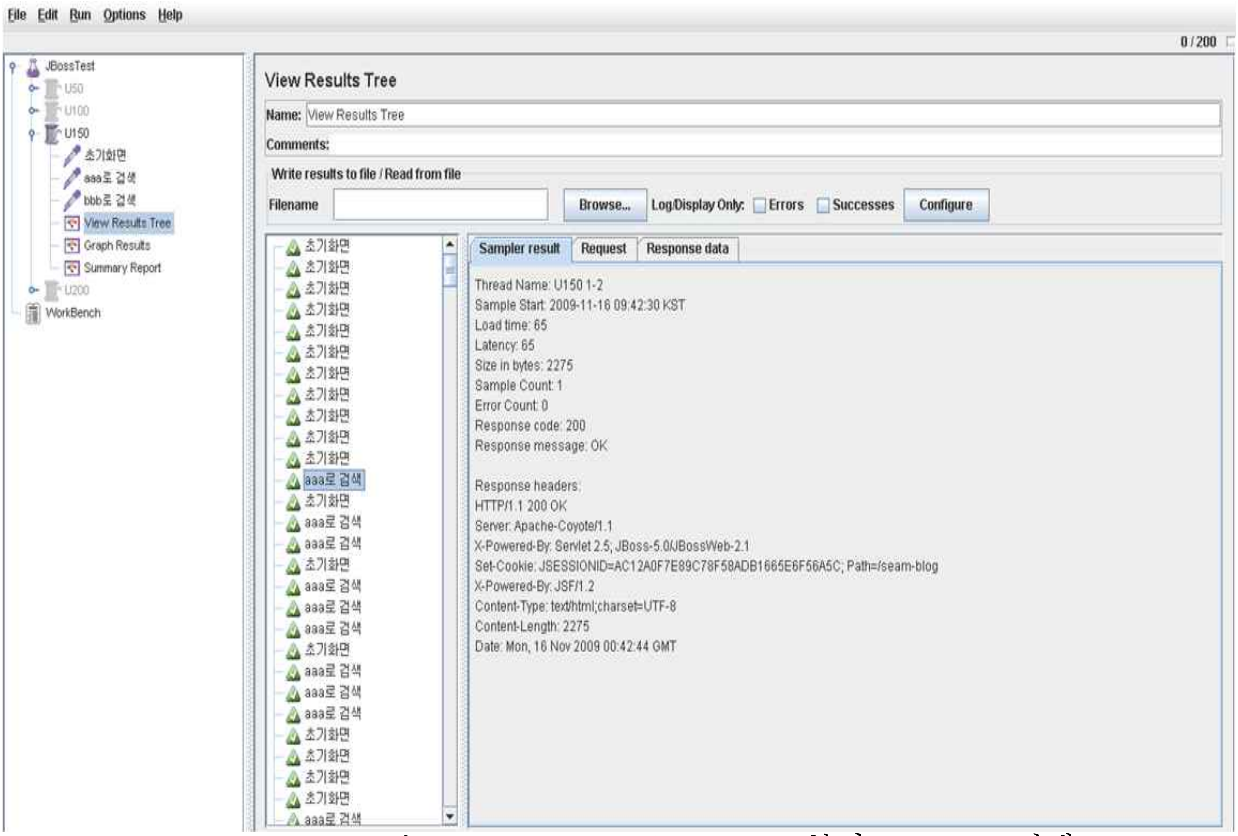


[그림4-11. View Results Tree - 초기화면]

블로그의 북마크를 검색하기 위한 초기화면이다.

총 450번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

② View Results Tree - 북마크 aaa로 검색



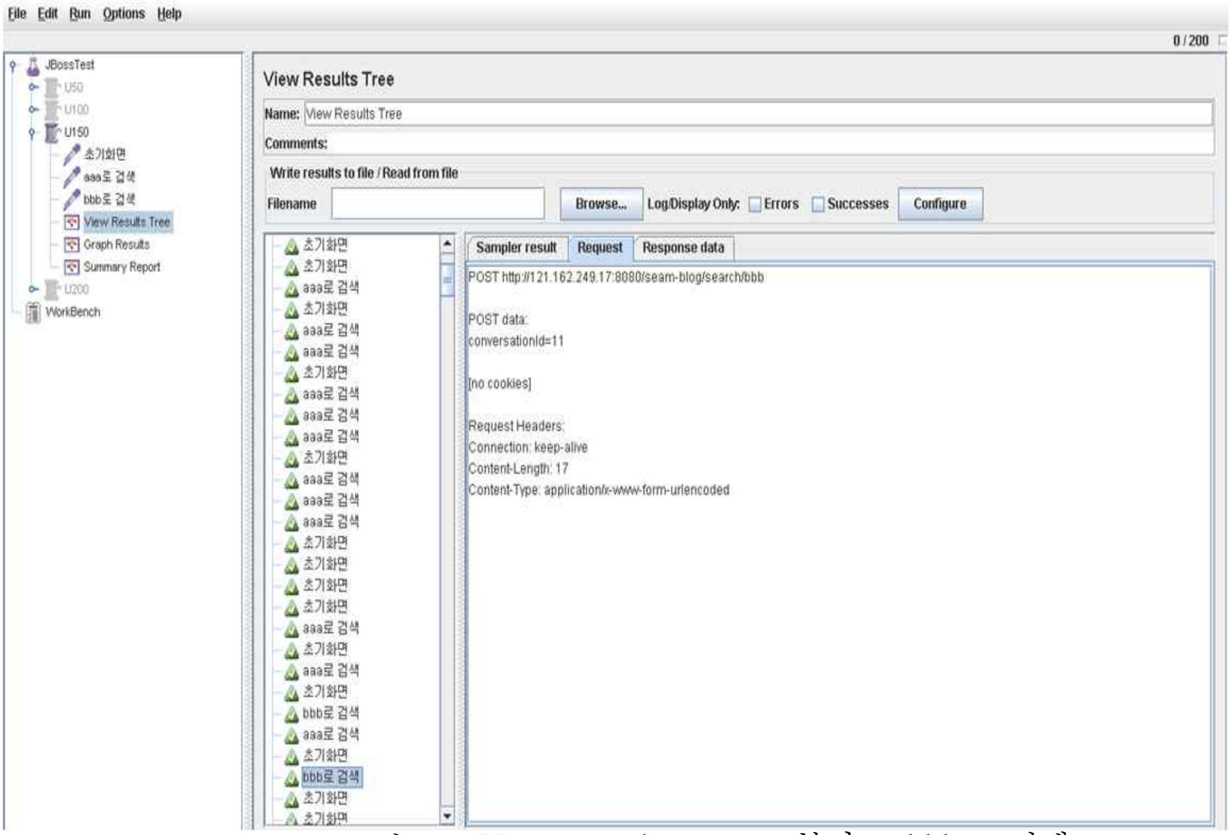
[그림4-12. View Results Tree - 북마크 aaa로 검색]

북마크 aaa로 검색한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

- 로딩 회수 : 65번
- 대기 횟수 : 65번
- 대상 페이지 사이즈 : 2275바이트
- 수행한 테스트 수 : 1번
- 에러 : 없음
- 응답 코드 : 200
- 응답 메시지 : 성공

③ View Results Tree - 북마크 bbb로 검색

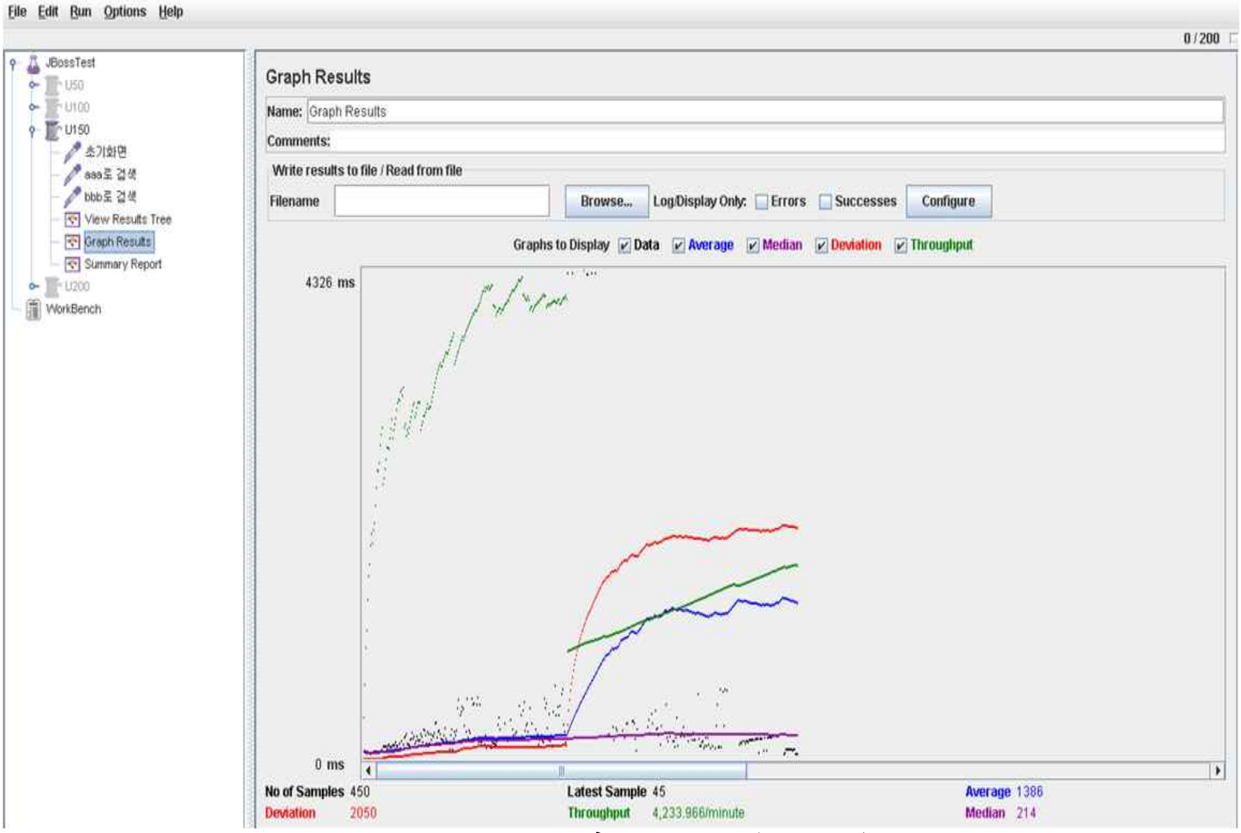


[그림4-13. View Results Tree - 북마크 bbb로 검색]

북마크 bbb로 검색한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

④ Graph Results

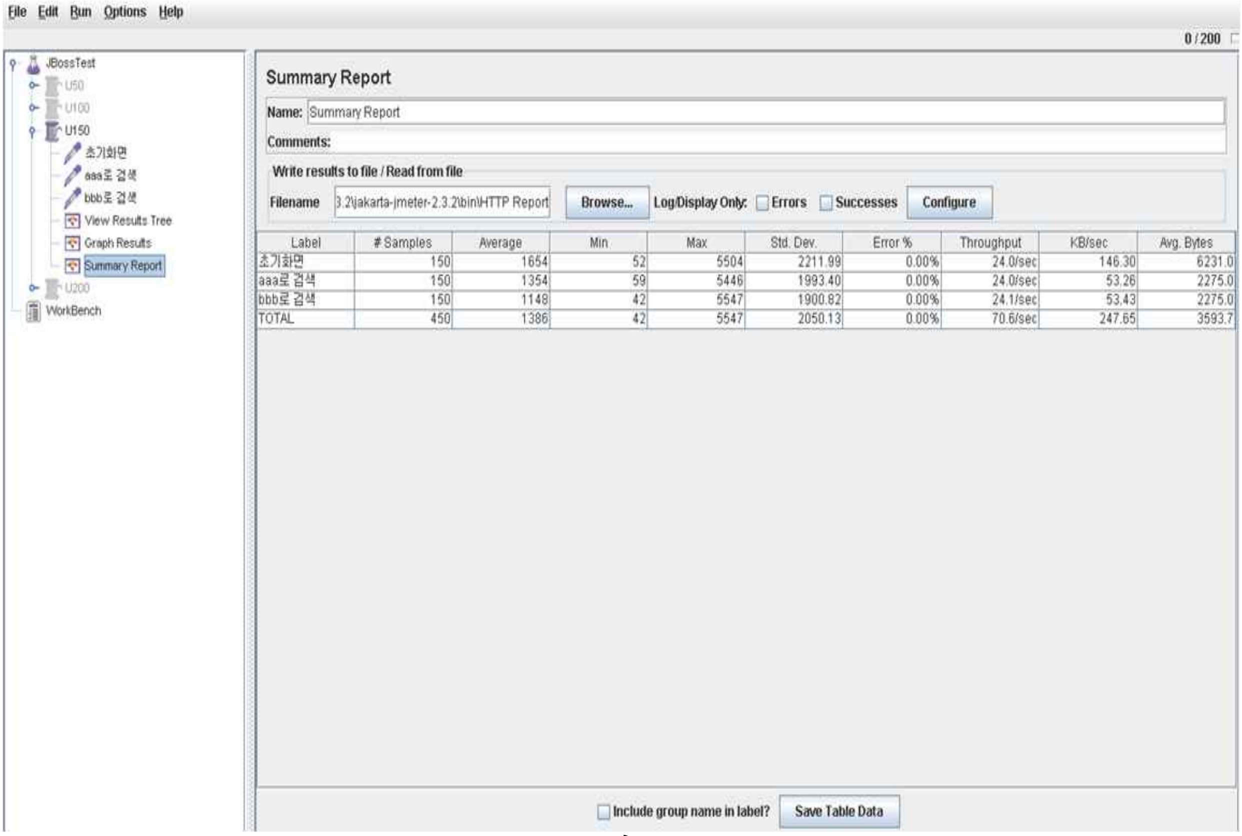


[그림4-14. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 450
- Average : 1386
- Median : 214
- Deviation : 2050
- Throughput : 4233.966/min

⑤ Summary Report



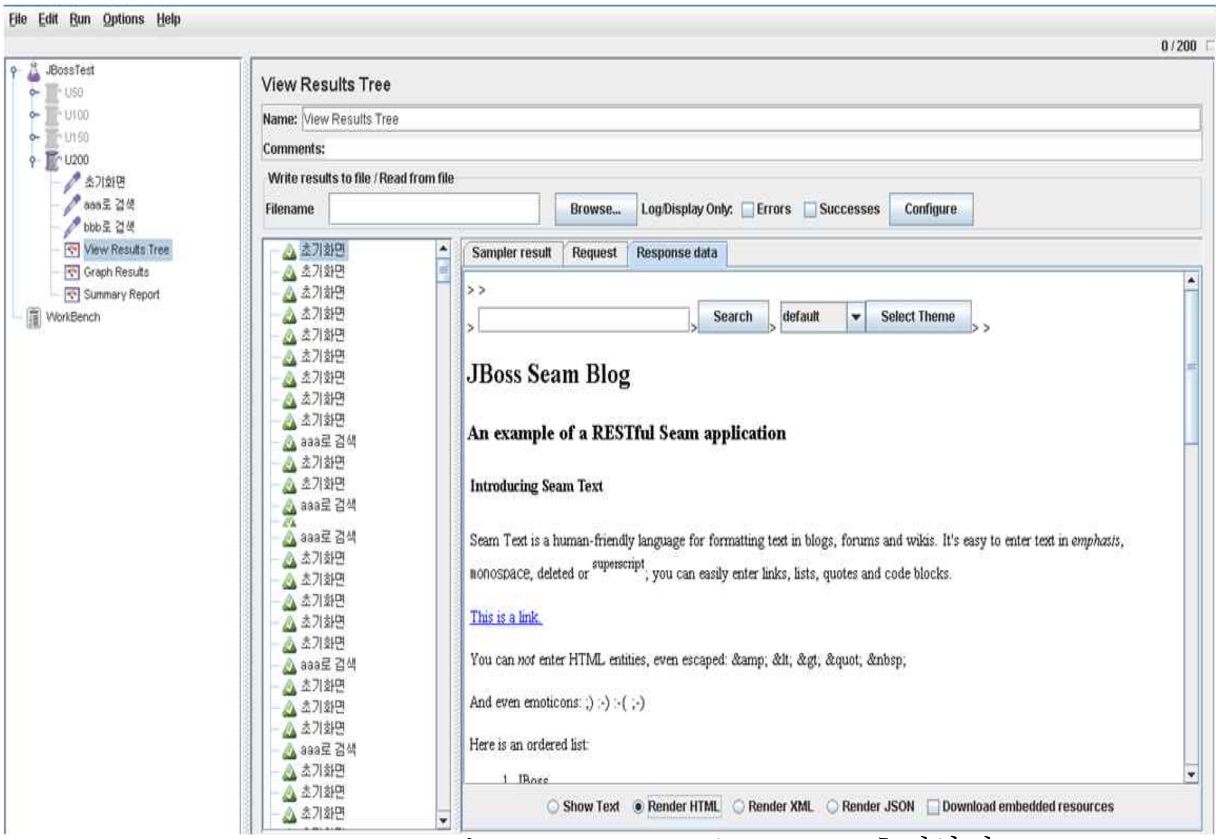
[그림4-15. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 450
- Average : 1386
- Min : 42
- Max : 5547
- Error : 0.00%
- Throughput : 70.6/sec

라. 200 User 동시 접속

① View Results Tree - 초기화면

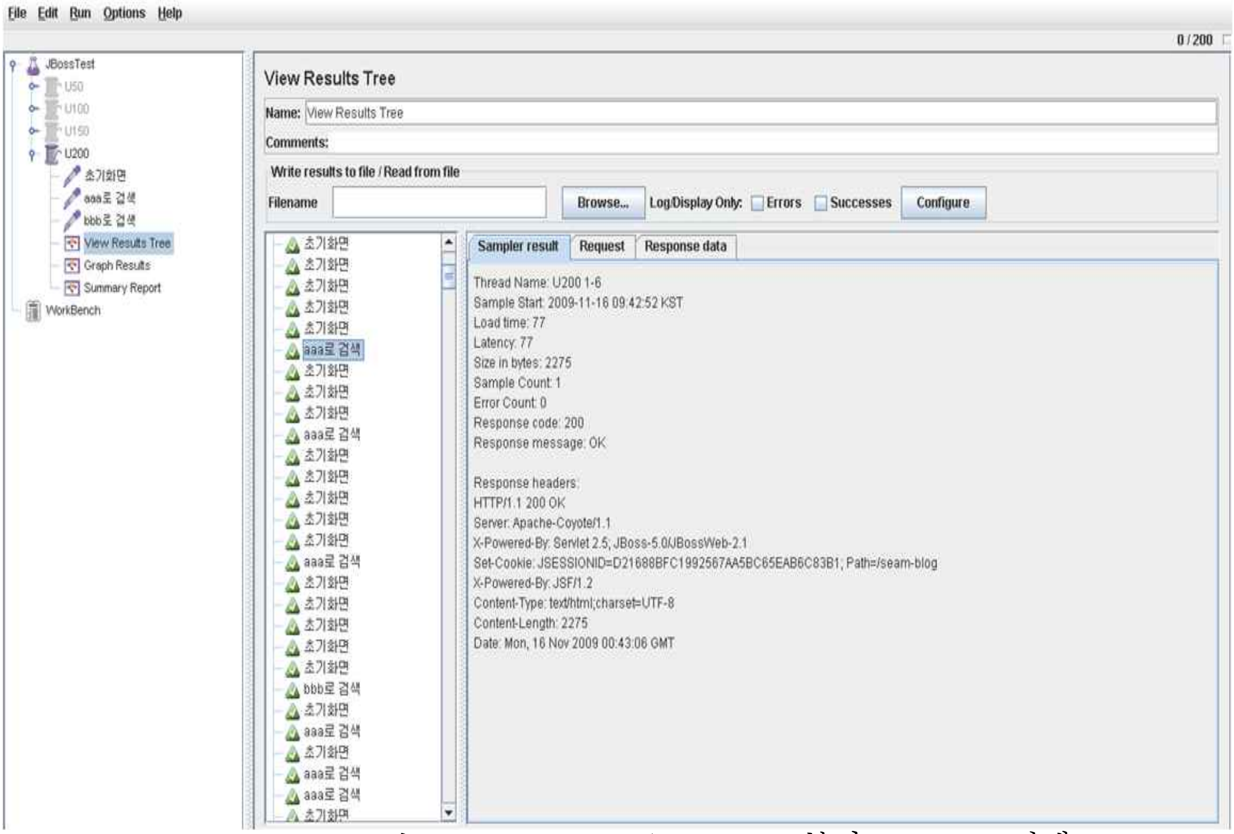


[그림4-16. View Results Tree - 초기화면]

블로그의 북마크를 검색하기 위한 초기화면이다.

총 600번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

② View Results Tree - 북마크 aaa로 검색



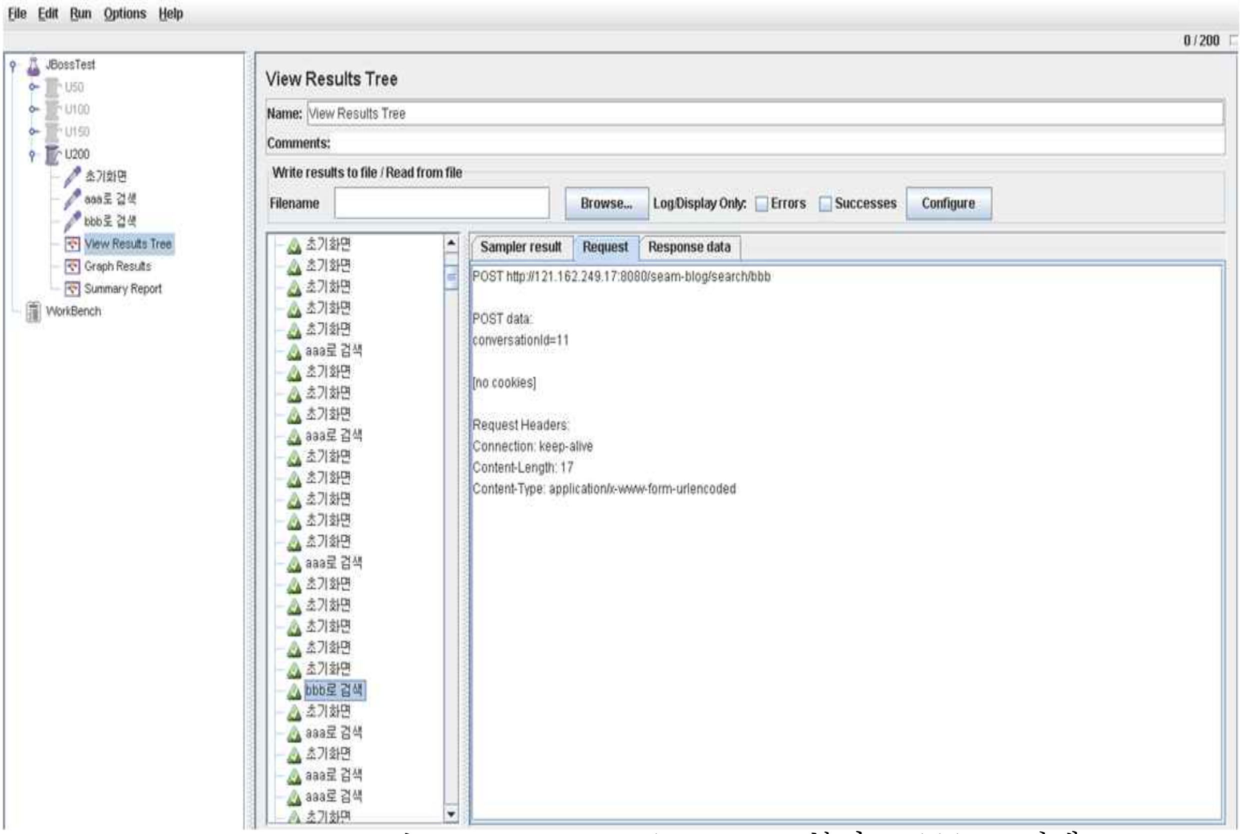
[그림4-17. View Results Tree - 북마크 aaa로 검색]

북마크 aaa로 검색한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

- 로딩 회수 : 77번
- 대기 횟수 : 77번
- 대상 페이지 사이즈 : 2275바이트
- 수행한 테스트 수 : 1번
- 에러 : 없음
- 응답 코드 : 200
- 응답 메시지 : 성공

③ View Results Tree - 북마크 bbb로 검색

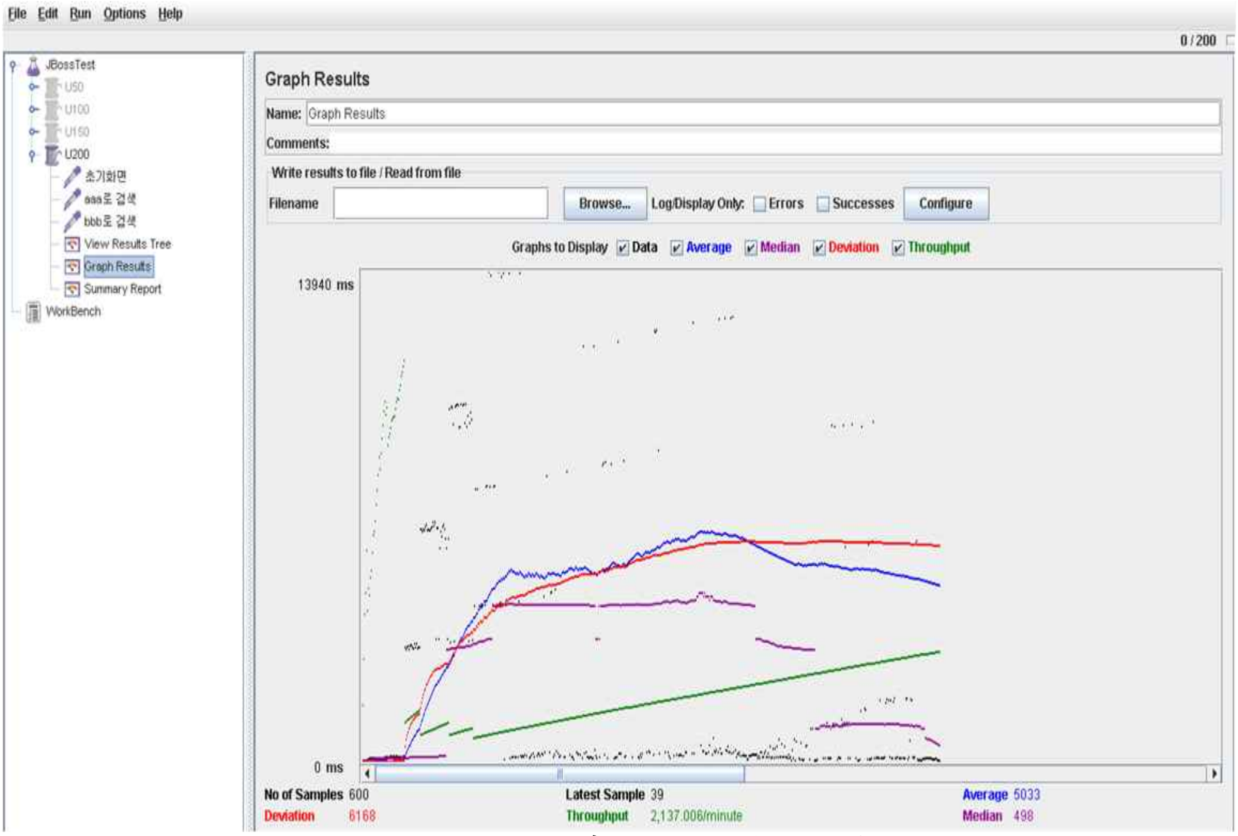


[그림4-18. View Results Tree - 북마크 bbb로 검색]

북마크 bbb로 검색한 상세 정보를 알 수 있다.

각 요청에 대한 정상적인 결과를 알 수 있다.

④ Graph Results

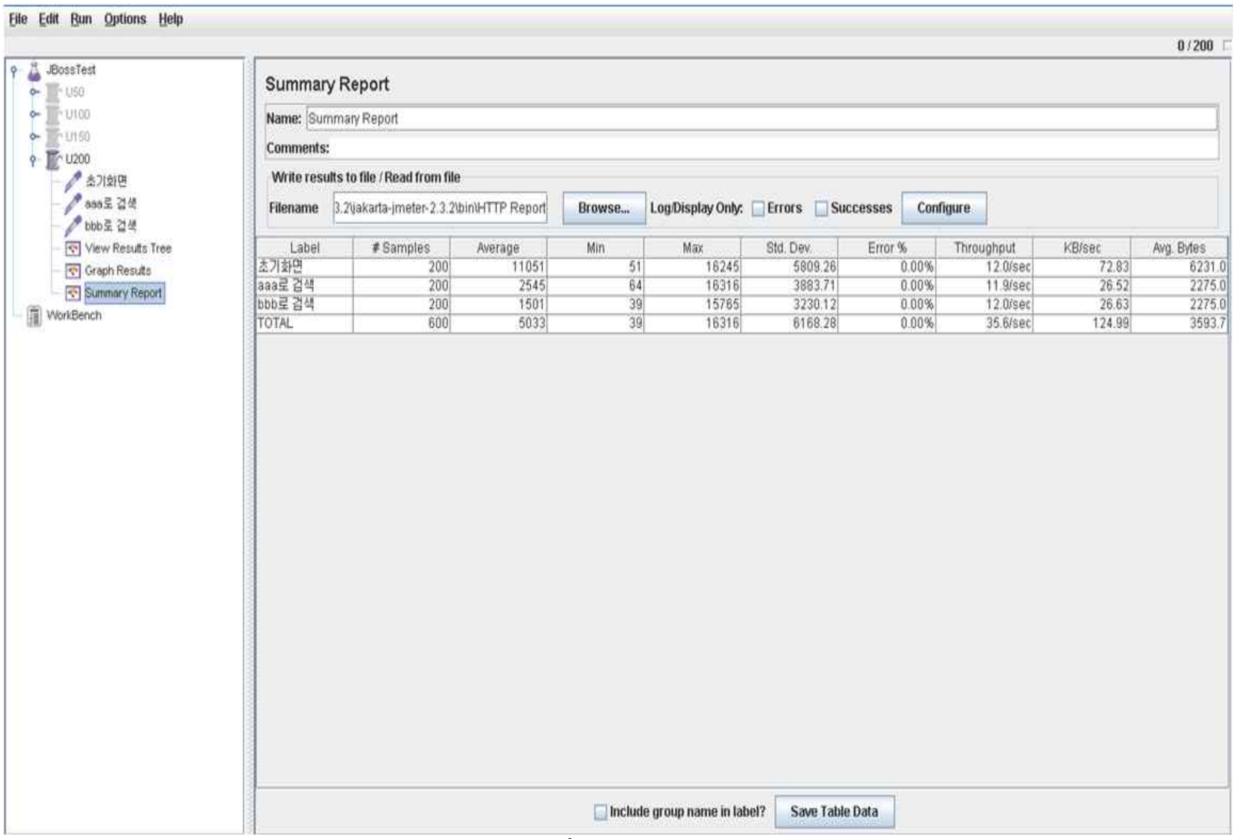


[그림4-19. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 600
- Average : 5033
- Median : 498
- Deviation : 6160
- Throughput : 2137.006/min

⑤ Summary Report



[그림4-20. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 600
- Average : 5033
- Min : 39
- Max : 16316
- Error : 0.00%
- Throughput : 35.6/sec

5. 종합

Summary Report

Request	Average	Min	Max	Error	Throughput	KB/sec
150	159	71	340	0.00%	112.7/sec	395.5
300	194	50	640	0.00%	194.7/sec	694.37
450	1131	47	5123	0.00%	78.8/sec	281.1
600	2871	50	9616	0.00%	58/sec	206.75

● 동시 접속 수 450 요청 부근부터 Throughput의 감소현상이 보이나 에러율은 0로서 시스템이 정상적으로 동작하는 것을 알 수 있다.

● 금번 테스트는 JBoss 단독 시스템 구성으로 하였으나 JVM, Web서비스, 클러스터링, mod_jk 등의 튜닝으로 보다 나은 퍼포먼스의 향상을 기대할 수 있다.

Ⅷ MySQL

목 차

1. 개요

본 문서는 공개SW 역량프라자 운영 사업의 공개SW 테스트 환경 구축을 통하여 테스트한 공개SW에 대한 기능, 성능, 호환성 등을 검증하고, 향후 공공 기관 및 기업에서 도입 시 참고할 수 있는 자료로 활용하기 위하여 작성되었다.

가. 목적

□ 공개SW를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 공개SW의 기능, 성능, 안정성, 호환성 등 검증
- 공개SW 기술의 도입 가이드라인 제공
- 효과적인 공개SW 도입 유도
- 안정적인 공개SW 시스템 도입

나. 선정 기준

□ 공개SW 테스트 대상은 다음 5가지 사항을 고려하여 선정되었다.

- 1번 : 역량프라자 내 대표 공개SW DBMS 발굴 제품
- 2번 : 국내에서 개발된 제품
- 3번 : 국내 기업에서 많이 도입되는 제품
- 4번 : Global에서 많이 알려진 제품
- 5번 : 기술지원(업데이트) 가능 제품

[표 1-1. 선정 기준]

DataBase	1번	2번	3번	4번	5번	결과	비고
CUBRID	●	●	◑	◔	●	◕	선정
MySQL	●	○	●	●	●	◕	선정
Postgre SQL	○	○	◑	◕	●	◑	보류

※ Postgre SQL은 국내 도입된 사례가 극히 적은 제품으로 파악됨

2. 테스트 환경 구성

가. 테스트 일정

[표 2-1. 일정표]

구분	월	10				11				12			
	주	1	2	3	4	1	2	3	4	1	2	3	4
테스트 대상 발굴 및 선정													
테스트 환경 구성													
테스트 항목 선정													
테스트 시나리오 작성													
테스트 수행													

나. 테스트 대상 소개

□ MySQL

MySQL 데이터베이스 서버는 세계에서 가장 널리 사용되는 오픈소스(open source) 데이터베이스 중 하나이다. 매우 효율적인 아키텍처를 채택하여 빠른 성능과 쉬운 사용자 구성을 가능하게 해준다. 소프트웨어 내적으로 다량의 코드 재사용과 풍부한 기능 생산에 대한 최소화 기법은 최고의 속도와 compactness, 안정성과 용이한 deployment를 보장해주는 데이터베이스 관리 시스템을 탄생시켰다. 테이블 핸들러(table handler)에서 코어 서버(core server)를 분리하는 독창적인 방식은 정교한 트랜잭션 제어(transaction control) 또는 트랜잭션이 필요 없는 초고속의 디스크 액세스(disk access)를 구현하도록 해주었고, 이와 같은 기능은 모든 상황에 제일 적합한 방식이다.

MySQL 데이터베이스 서버는 GPL(GNU General Public License)의 규약에 의해 자유롭게 사용할 수 있으며, 상업용 라이센스는 GPL의 규약에 제한을 받지 않고자 하는 사용자들이 선택할 수 있다.

□ MySQL 특징

- 확장성 및 유연성 ; 다음 환경에서도 실행
  ○ 불과 1MB의 공간에 내장된 어플리케이션
  ○ 테라바이트 규모의 정보를 담은 대형 데이터 웨어하우스
- 뛰어난 성능
  ○ 테이블 및 색인 파티셔닝, 초고속 로드 유틸리티
  ○ 뛰어난 메모리 캐시, 텍스트 전체 색인 등
- 고가용성
  ○ 로우 기반 및 하이브리드 복제 기능으로 고속 마스터/슬레이브 복제 구성 실행

○ 즉각적인 복구 기능을 제공하는 특화된 클러스터 서버
- 강력한 트랜잭션 지원
○ 완벽한 ACID(정밀성, 일관성, 격리성, 내구성) 특성의 트랜잭션 지원
○ 제한 없는 행 단위(row-level) 잠금 기능
○ 분산 트랜잭션 기능
○ 멀티버전 트랜잭션 지원
- 뛰어난 웹 및 데이터 웨어하우스 기능
○ 고성능 쿼리 엔진
○ 초고속 데이터 삽입 기능
○ 신속한 텍스트 전체 검색 등 전문적 웹 기능에 대한 강력한 지원
- 강력한 데이터 보호 기능
○ 권한을 가진 사용자만 액세스할 수 있도록 하는 강력한 메커니즘
○ 안전한 보안 연결을 지원하는 SSH 및 SSL
○ 강력한 데이터 암호화 및 해독 기능
- 포괄적인 어플리케이션 개발
○ 저장된 프로시저, 트리거, 함수, 뷰, 커서, ANSI-표준 SQL 등 다양한 지원
○ 거의 모든 애플리케이션에 MySQL 데이터베이스 지원 기능을 내장할 수 있는 플러그인 라이브러리
- 관리 편이성
○ 일반적으로 반복되는 SQL 기반 작업을 데이터베이스 서버에서 실행하도록 자동으로 예약하는 Event Scheduler 사용
○ 소프트웨어 다운로드부터 설치까지 평균 15분 내에 완료
- 최소 수준의 총 소유비용
○ 시스템 가동 중단 시간을 줄이면서도 데이터베이스 라이선스 비용과 하드웨어 지출 절감

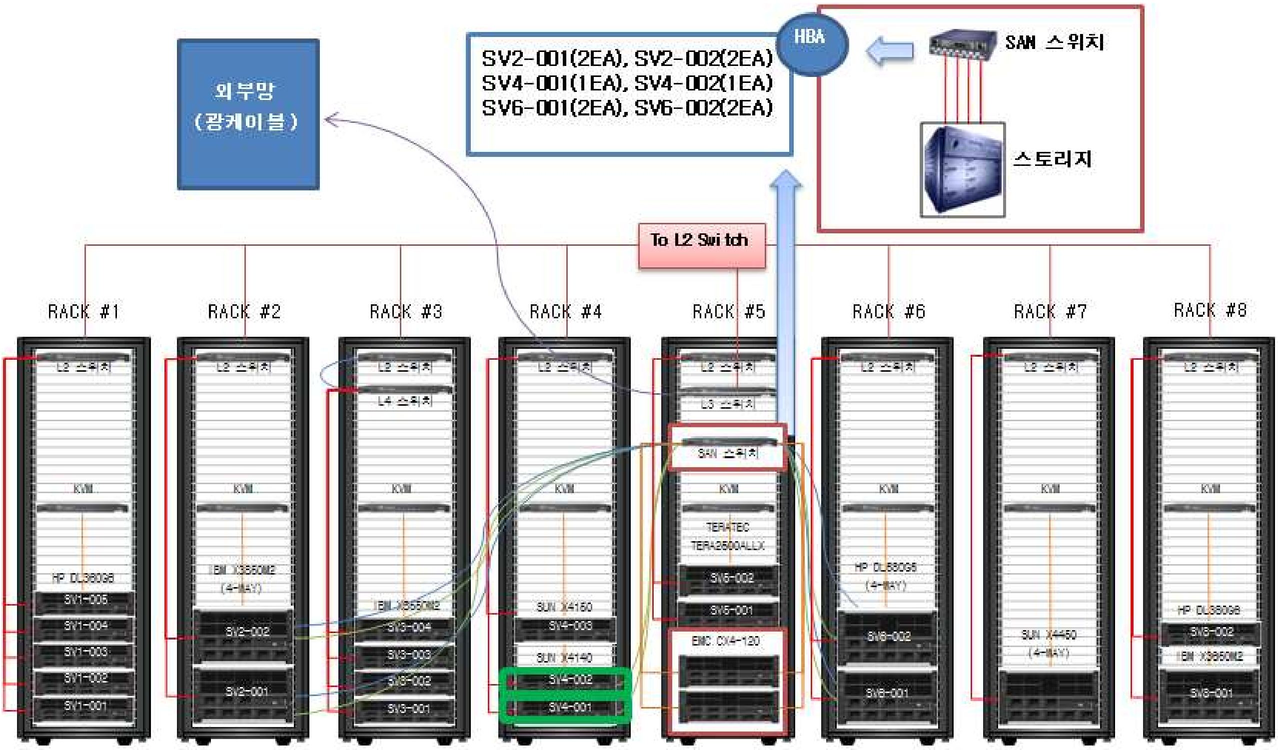
다. 테스트 환경

□ 테스트 서버

[표 2-2. 테스트 서버 규격]

구분	제품명	CPU	MEM	Disk	NIC	수량
SUN	X4140	Quad-Core 2.5Ghz 2-Way	8GB	146GB*5	Gigabit 2Port	2

□ 테스트 베드 실장도



☞ 박스 안의 서버가 DB 테스트용 서버로 사용 됨.

[그림 2-1. 테스트 서버 실장도]

□ 서버 위치

[표 2-3. 서버 위치]

제조사	모델명	Rack 위치	비 고
SUN	X4140	#4	-

3. 테스트 시나리오

가. 호환성 테스트

□ 설치 지원 플랫폼(HW&OS)
- 운영체제별 지원 여부
- Architecture별 지원 여부

□ Installation packages
- Platform 별 package 제공 여부
- Binary Package 제공 여부
- Source Package 제공 여부

□ Installation Deployment
- 기본 설치 Testing
- Installation Customizable 여부
- 기본 작업 테스트
  ▪ Installation 방법
  ▪ Installation procedure listup(기본 스키마)

나. 성능 테스트

□ Sysbench Tool을 사용한 테스트

다. 가용성 테스트

□ Active/Active High Availability
- Active/Active Solution 제공 유무
- Active/Standby Solution 제공 유무

라. 관리 도구 테스트

□ Administration tools

□ Query Tools(Client Side)

□ Monitoring Tools

□ ETC Mgmt Tools

4. 테스트

가. 호환성 테스트

□ MySQL 홈페이지에서 설치 지원 플랫폼 확인(HW&OS)

[표 4-1. 플랫폼별 지원 내역]

<table>
<tr><th>제조사</th><th>운영체제</th><th>Architecture</th><th>MySQL 5.4</th></tr>
<tr><td rowspan="3">Red Hat</td><td>RHEL5</td><td>x86, x86_64, Intel IA64</td><td rowspan="3">○</td></tr>
<tr><td>RHEL4</td><td>x86, x86_64, Intel IA64</td></tr>
<tr><td>RHEL3</td><td>x86, x86_64, Intel IA64</td></tr>
<tr><td rowspan="3">Novell</td><td>SuSe Enterprise Linux 11</td><td>x86, x86_64</td><td rowspan="3">○</td></tr>
<tr><td>SuSe Enterprise Linux 10</td><td>x86, x86_64, Intel IA64</td></tr>
<tr><td>SuSe Enterprise Linux 9</td><td>x86, x86_64, Intel IA64</td></tr>
<tr><td>Microsoft</td><td>Windows</td><td>x86, x86_64</td><td>○</td></tr>
<tr><td rowspan="2">SUN</td><td>Solaris 10</td><td>SPARC (32 and 64 bit), x86_64, x86/32-bit</td><td rowspan="2">○</td></tr>
<tr><td>Solaris 9</td><td>SPARC (32 and 64 bit), x86/32-bit</td></tr>
<tr><td rowspan="2">HP</td><td>HP-UX 11.23</td><td>Intel IA64</td><td rowspan="2">○</td></tr>
<tr><td>HP-UX 11.11</td><td>PA-RISC 2.0, 64-bit; PA-RISC 1.1 and 2.0</td></tr>
<tr><td rowspan="2">IBM</td><td>AIX 5.3</td><td>POWERPC_64</td><td rowspan="2">○</td></tr>
<tr><td>IBM i5/OS</td><td>POWERPC_64, POWERPC_32</td></tr>
<tr><td rowspan="4">Others</td><td>Fedora</td><td>x86, x86_64</td><td rowspan="4">○</td></tr>
<tr><td>OpenSuSE</td><td>x86, x86_64</td></tr>
<tr><td>CentOS</td><td>x86, x86_64</td></tr>
<tr><td>Ginux</td><td>x86, x86_64</td></tr>
</table>

□ 호환성 테스트 대상 및 지원 확인

[표 4-2. 호환성 테스트 결과]

운영체제	Architecture	MySQL 버전	지원 여부
CentOS 5.3	x86_64	5.4	지원
Fedora 11	x86_64	5.4	지원
GINUX 2.2	x86_64	5.4	지원
SULinux 2.0	x86_64	5.4	지원

□ Installation packages

- Platform 별 Binary Package 제공 여부

○ Red Hat

[표 4-3. RHEL 패키지]

PLATFORM	Bynary Package	다운로드 위치
RHEL5	x86, x86_64, Intel IA64 제공	http://dev.mysql.com/downloads/mysql/5.4.html
RHEL4		
RHEL3		

○ Novell

[표 4-4. SuSE 패키지]

PLATFORM	Bynary Package	다운로드 위치
SuSE Enterprise Linux 11	x86, x86_64 제공	http://dev.mysql.com/downloads/mysql/5.4.html
SuSE Enterprise Linux 10	x86, x86_64, Intel IA64 제공	
SuSE Enterprise Linux 9		

○ Microsoft

[표 4-5. MS Windows 패키지]

PLATFORM	Bynary Package	다운로드 위치
Windows	x86, x86_64 제공	http://dev.mysql.com/downloads/mysql/5.4.html

○ SUN

[표 4-6. Solaris 패키지]

PLATFORM	Bynary Package	다운로드 위치
Solaris 10	SPARC (32 and 64 bit), x86_64, x86/32-bit 제공	http://dev.mysql.com/downloads/mysql/5.4.html
Solaris 9	SPARC (32 and 64 bit), x86/32-bit 제공	

○ HP

[표 4-7. HP-UX 패키지]

PLATFORM	Bynary Package	다운로드 위치
HP-UX 11.23	Intel IA64, 64bit 제공	http://dev.mysql.com/downloads/mysql/5.4.html
HP-UX 11.11	PA-RISC 2.0, 64-bit; PA-RISC 1.1 and 2.0 제공	

○ IBM

[표 4-8. AIX 패키지]

PLATFORM	Bynary Package	다운로드 위치
AIX 5.3	POWER_PC 64bit 제공	http://dev.mysql.com/downloads/mysql/5.4.html
i5/OS	PA-RISC 2.0, 64-bit; PA-RISC 1.1 and 2.0 제공	

○ Others

[표 4-9. Others 리눅스 배포판 패키지]

PLATFORM	Bynary Package	다운로드 위치
Fedora	x86, x86_64, Intel IA64 제공	http://dev.mysql.com/downloads/mysql/5.4.html
OpenSuSE		
CentOS		
Ginux		

※ 참고 사항
Others 제품들은 Generic RPM 버전을 사용 가능

- Source Package 제공 여부

[표 4-10. Source 패키지]

PLATFORM	Source Package	다운로드 위치
ALL	제공	http://dev.mysql.com/downloads/mysql/5.4.html

☐ Installation Deployment

- 기본 설치 Testing

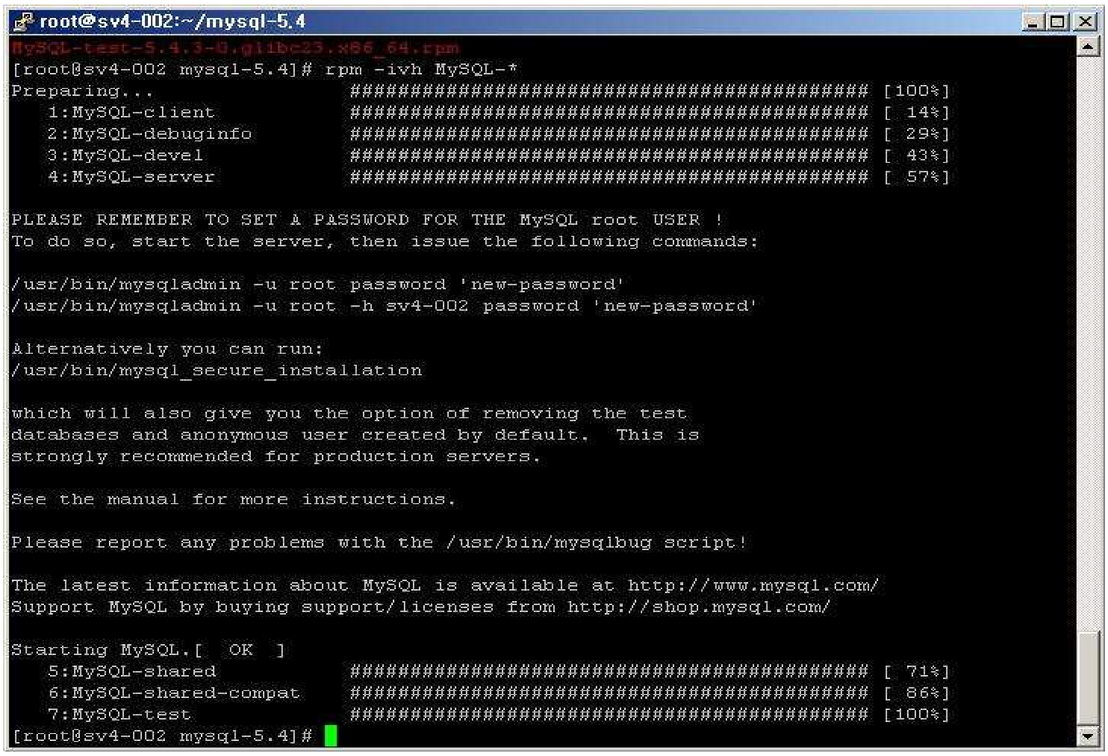
[표 4-11. MySQL 설치 버전]

OS(X86_64)	Install Package	MySQL 버전
CentOS 5.3 Fedora11 Ginux 2.2 SULinux 2.0	MySQL-client-5.4.3-0.glibc23.x86_64.rpm MySQL-debuginfo-5.4.3-0.glibc23.x86_64.rpm MySQL-devel-5.4.3-0.glibc23.x86_64.rpm MySQL-server-5.4.3-0.glibc23.x86_64.rpm MySQL-shared-5.4.3-0.glibc23.x86_64.rpm MySQL-shared-compat-5.4.3-0.glibc23.x86_64.rpm MySQL-test-5.4.3-0.glibc23.x86_64.rpm	5.4.3

○ Installation 명령

[표 4-12. MySQL 설치 방법]

구분	명령어	결과
MySQL 설치	rpm -ivh "Package Name".rpm	정상 설치 ([그림 4-1] 참고)



[그림 4-1. MySQL 설치 화면]

- Installation Customizable 여부

○ MySQL glibc23 Compile Binary 사용 방법

Binary 압축 버전을 다운로드 받아서 원하는 위치에 압축 풀고 사용하면 됨

```
tar xvfz mysql-5.4.3-beta-linux-x86_64-glibc23.tar.gz -C /usr/local/mysql
```

- 기본 작업 테스트

○ Installation 방법

MySQL 모든 설치 Package를 하나의 디레토리로 이동 후 아래와 같이 설치 진행 함

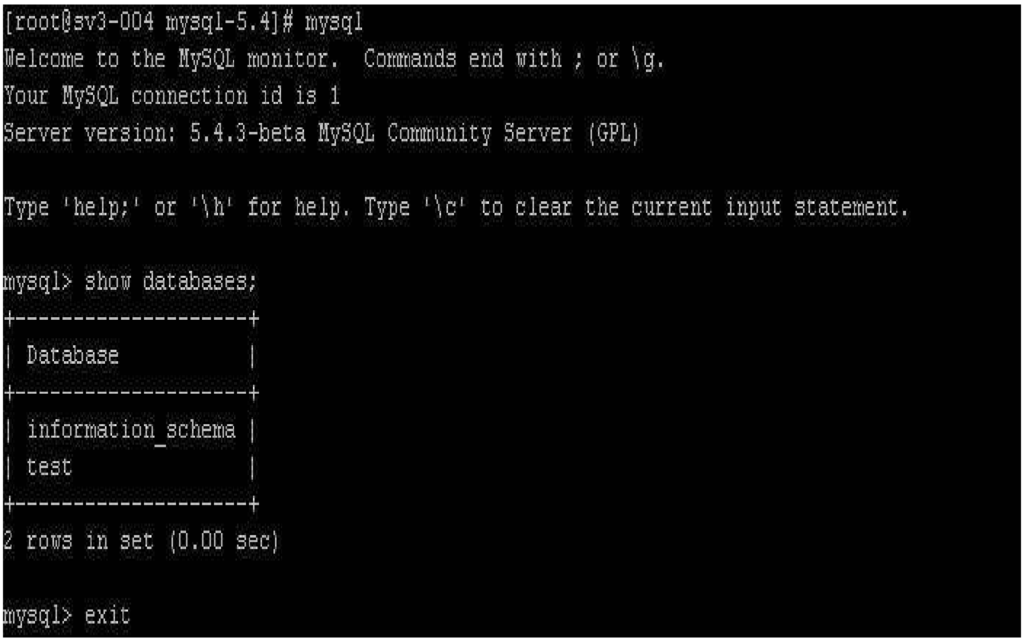
```
$ rpm -ivh MySQL-*.rpm
```

※ 참고 사항

- MySQL 5.4 버전을 설치 후 mysql이 자동으로 시작되는데 서비스를 정지 후 MySQL DB 데이터를 재생성 후 설정 파일 수정 및 기타 작업을 하면 됨
- 세부 작업은 아래의 표에 설명 됨

○ Installation procedure listup(기본 스키마)

```
$ service mysql stop ↵
$ mysql_install_db ↵
$ cp /usr/share/mysql/my-medium.cnf /etc/my.cnf ↵
$ vi /etc/my.cnf ↵
  중략
[client]
#password        = your_password
user             = mysql      ==> 설정 추가
port             = 3306
socket           = /var/lib/mysql/mysql.sock
   중략
:wq! ==> 저장 후 빠져 나감
$ service mysql start ↵
$ mysql ↵ ==> [그림 4-2]처럼 접속이 되어 Database 목록이 확인되면
                정상
```

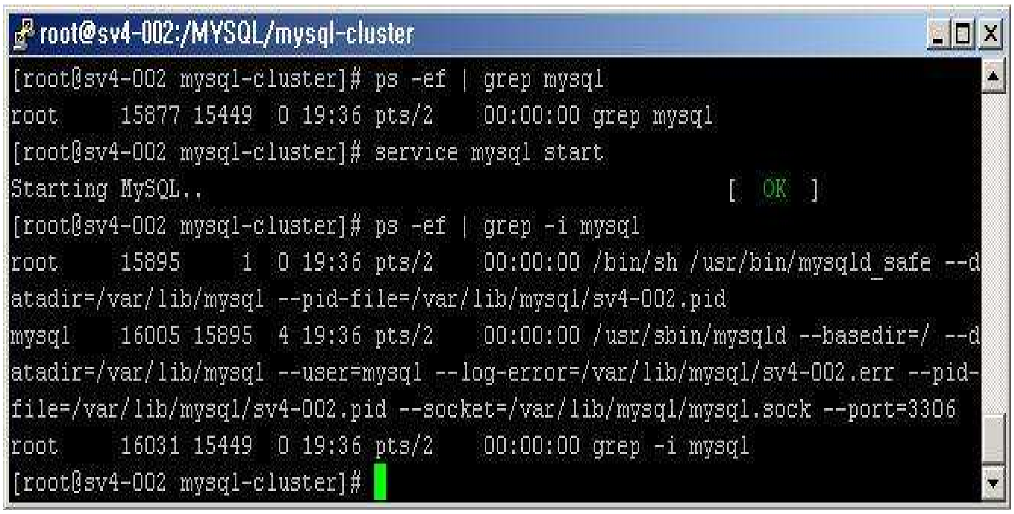


```
[root@sv3-004 mysql-5.4]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.4.3-beta MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| test               |
+--------------------+
2 rows in set (0.00 sec)

mysql> exit
```

[그림 4-2. 데이터베이스 목록 확인]

○ MySQL 프로세스 확인

\# ps -ef | grep -i mysql



```
root@sv4-002:/MYSQL/mysql-cluster
[root@sv4-002 mysql-cluster]# ps -ef | grep mysql
root     15877 15449  0 19:36 pts/2    00:00:00 grep mysql
[root@sv4-002 mysql-cluster]# service mysql start
Starting MySQL..                                           [  OK  ]
[root@sv4-002 mysql-cluster]# ps -ef | grep -i mysql
root     15895     1  0 19:36 pts/2    00:00:00 /bin/sh /usr/bin/mysqld_safe --datadir=/var/lib/mysql --pid-file=/var/lib/mysql/sv4-002.pid
mysql    16005 15895  4 19:36 pts/2    00:00:00 /usr/sbin/mysqld --basedir=/ --datadir=/var/lib/mysql --user=mysql --log-error=/var/lib/mysql/sv4-002.err --pid-file=/var/lib/mysql/sv4-002.pid --socket=/var/lib/mysql/mysql.sock --port=3306
root     16031 15449  0 19:36 pts/2    00:00:00 grep -i mysql
[root@sv4-002 mysql-cluster]#
```

[그림 4-3. MySQL 프로세스 확인]

나. 성능 테스트

□ sysbench tool을 사용한 성능 테스트

- sysbench tool 소개

기본적으로 MySQL의 성능을 측정하기 위해 만들어진 오픈소스 벤치마크 툴인 Sysbench(sysbench.sourceforge.net)를 기반으로 수행되었다. Sysbench는 데이터베이스가 동작하는 서버의 파일 I/O 성능, 스케쥴러 성능, 메모리 할당과 전송 속도 및 데이터베이스의 OLTP(Online Transaction Processing) 성능을 측정할 수 있으며 처음 만들어질 당시에는 MySQL을 위한 것이었으나 현재는 테스트의 스펙을 만족시키는 각종 데이터베이스들을 지원하고 있다.

- 테스트 방법

○ 테이블 크기

100만건(300MB)

○ 데이터베이스 엔진

▪ MyISAM

주로 트랜잭션이 필요 없고 Select 쿼리가 많은 Database에 사용됨

▪ InnoDB

트랜잭션과 DB를 변경하는 작업이 많은 Database에 사용됨

○ 테이블 스키마(Schema)

```
| sbtest | CREATE TABLE ‘sbtest‘ (
  ‘id‘ int(10) unsigned NOT NULL AUTO_INCREMENT,
  ‘k‘ int(10) unsigned NOT NULL DEFAULT ’0’,
  ‘c‘ char(120) NOT NULL DEFAULT ’’,
  ‘pad‘ char(60) NOT NULL DEFAULT ’’,
  PRIMARY KEY (‘id‘),
  KEY ‘k‘ (‘k‘)
) ENGINE=MyISAM AUTO_INCREMENT=1000001 DEFAULT CHARSET=latin1 MAX_ROWS=1000000 |
```

※ 동일한 테이블에서 Engine을 InnoDB로 변경하여 동일한 방식으로 테스트를 진행함

```
mysql> alter table sbtest engine=innodb;
```

○ 테스트 명령

```
sysbench 사용 옵션
--num-threads=사용할 스레드 개수
--max-requests=최대 Request 개수
--oltp-table-size=Test할 Table의 Data 수
--mysql-db=사용할DB명
--oltp-table-name=Test할 Table명
--mysql-table-engine=사용할 DB Engine
```

```
sysbench 사용 명령
$ sysbench --mysql-user=root --mysql-password='kipa!@#$' \
 --num-threads=" " --max-requests=100000 \
 --test=oltp --oltp-table-size=1000000 \
 --mysql-db=sysbench --oltp-table-name=sbtest \
 --mysql-table-engine="  "\
 --mysql-socket=/var/lib/mysql/mysql.sock run
```

○ 테스트 설명

OLTP(On Line Transaction Processing) 기반에서 10만 건의 처리량을 기준으로 클라이언트 thread를 늘려가며 처리하는 시간을 확인

--num-threads 옵션 값 ☞ 2,4,8,16,32,64,88,128,196,256

--mysql-table-engine 옵션 값 ☞ myisam, innodb

○ thread별 TPS(Transaction Per Second) 결과

▪ MyISAM Engine(Sec 단위)

[표 4-13. MyISAM 테스트 결과]

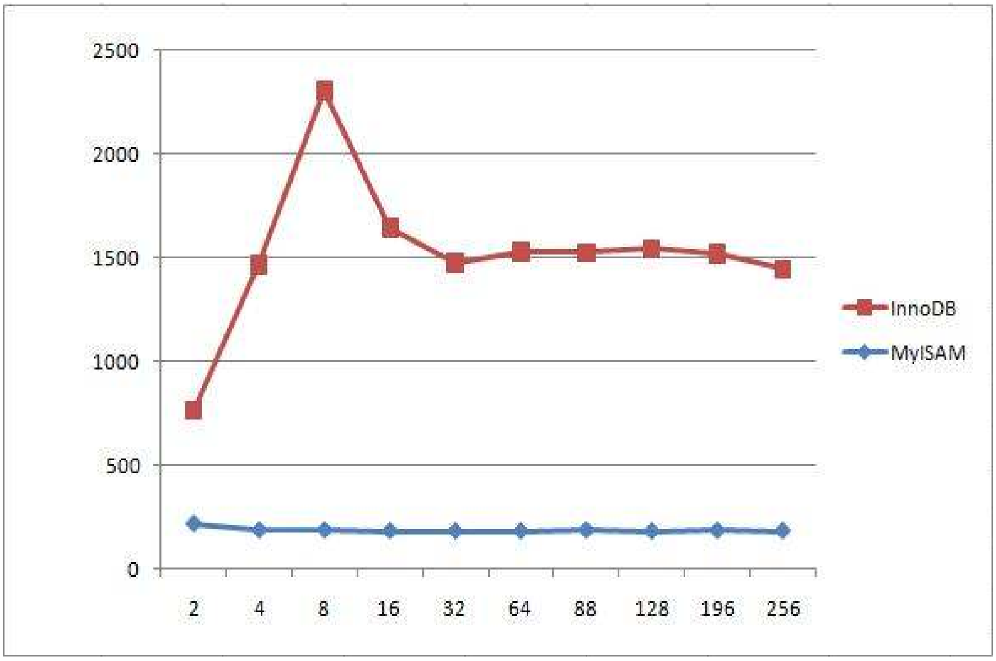
Thread	Transaction	R/W Requests	Total Time
2	220.27	4185.18	453.9829
4	188.10	3573.88	531.6352
8	187.89	3569.82	532.2397
16	184.55	3506.40	541.8667
32	183.51	3486.74	544.9218
64	182.81	3473.34	547.0242
88	187.87	3569.55	532.2802
128	183.01	3477.14	546.4258
196	185.53	3525.06	538.9986
256	184.78	3510.83	541.1822

▪ InnoDB Engine(초(Sec) 단위)

[표 4-14. InnoDB 테스트 결과]

Thread	Transaction	R/W Requests	Total Time(sec)
2	767.44	14581.42	130.3028
4	1466.64	27866.12	68.1832
8	2304.86	43792.31	43.3866
16	1644.82	31251.54	60.7982
32	1473.74	28000.99	67.8561
64	1527.16	29016.30	65.4843
88	1524.89	28972.84	65.5800
128	1542.06	29299.08	64.8524
196	1517.54	28833.23	65.8995
256	1445.84	27471.25	69.1687

▪ Transaction 결과

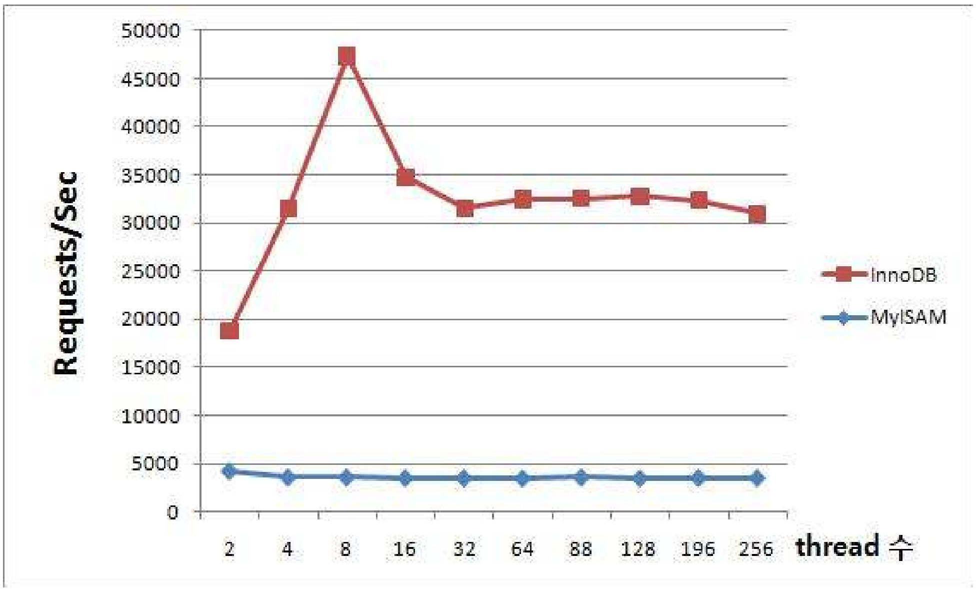


[그림 4-4. Transaction 성능]

※ Transaction 처리 테스트를 수행한 결과, MyISAM은 Transaction을 지원하지 않으며, 성능의 변화가 거의 없이 일정함을 알 수 있다.

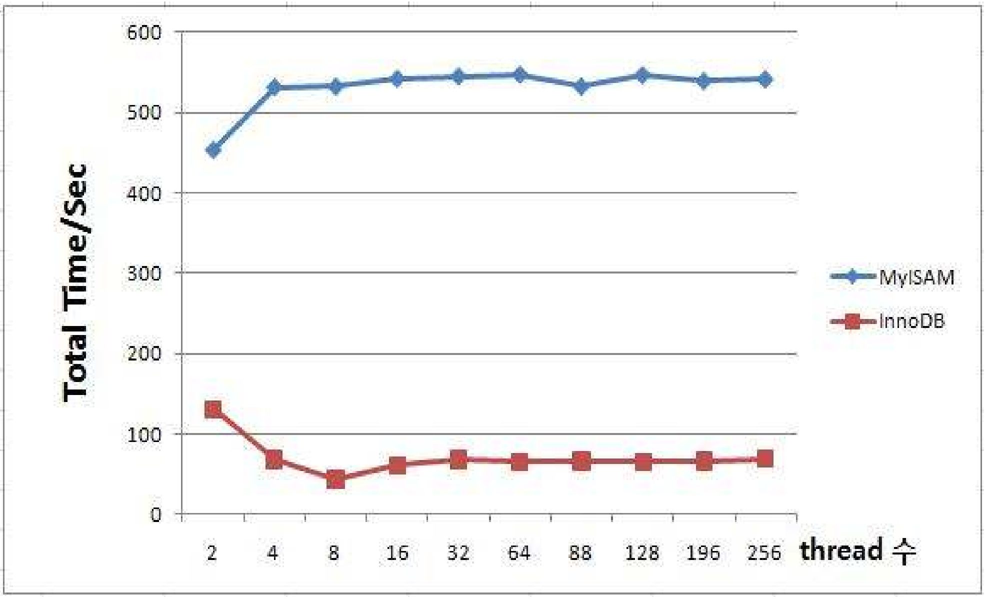
※ InnoDB는 8Thread에서 가장 좋은 성능을 냈으며, Thread 증가에 따라 성능은 1500TPS/Sec를 기준으로 일정하게 나타났다.

▪ Read/Write Requests 결과



[그림 4-5. R/W Request 성능]

▪ Total Time 결과



[그림 4-6. Total Time 결과]

테스트 결과 MyISAM의 경우, thread 수 증가에 따른 성능 저하는 크게 발생하지 않았다.
MyISAM 엔진의 경우, 이미 알려진 바와 같이 Transaction 처리는 없고 Select Query가 많은 데이터베이스 시스템에 적합하다.
InnoDB 엔진의 경우, 그래프에서 보이는 것처럼 Transaction 처리와 데이터베이스 변경 작업이 많은 시스템에 적합한 것을 알 수 있다.

다. 가용성 테스트

□ High Availability

MySQL 클러스터는 분산 컴퓨팅 환경에서 high-availability와 high-redundancy를 채택하였다. MySQL 클러스터는 NDB 클러스터 스토리지 엔진을 사용하여, 클러스터에서 여러 개의 서버가 함께 돌아가도록 한다.

- Cluster 기능 제공 유무

[표 4-15. Cluster 기능 제공 확인]

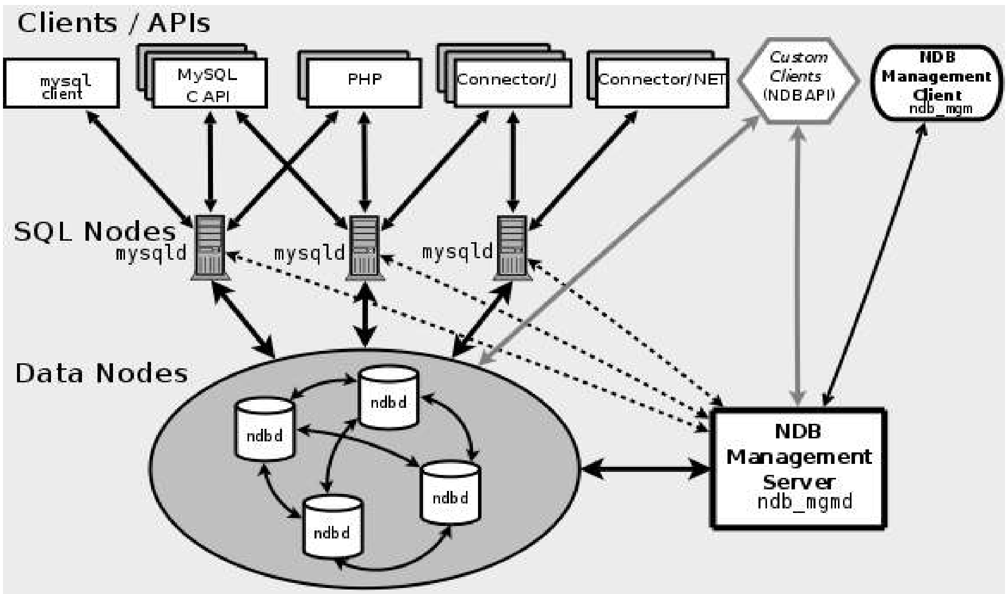
구분	Package	제공 여부
Active/Active	mysql-cluster-gpl-7.0.8a-linux-x86_64-glibc23.tar.gz (Generic Linux Version용)	○
Active/Standby		

- 운영체제별 지원

[표 4-16. Cluster 기능 운영체제별 확인]

구분	운영체제 종류	지원 여부
mysql-cluster	Windows, Linux, Red Hat Enterprise Linux, SuSE Enterprise Linux, Mac OS X, Sun Solaris	○

- Cluster 구성도



[그림 4-7. Cluster 구성도]

- Cluster 구성 설치

○ 서버 구성

[표 4-17. 테스트 서버 구성 요건]

서버 구성	수 량	비 고
Data node	2	Management node는 data node 1 서버에 구성
Management node	1	

○ Cluster Binary 설치

▪ Binary 다운로드 위치

http://dev.mysql.com/downloads/select.php?id=14&display=current#downloads

※ Binary 버전에는 ndb가 포함되어 있음

▪ 설치 순서

☞ Binary를 설치할 디렉터리에 압축을 해제한다.

tar xvfz mysql-cluster-gpl-7.0.9-linux-x86_64-glibc23.tar.gz

☞ MySQL 사용자 생성

useradd mysql –d /MySQL/mysql-cluster

☞ MySQL Home Directory 권한 변경

chown mysql.mysql -R /MySQL/mysql-cluster

☞ MySQL DB 설치

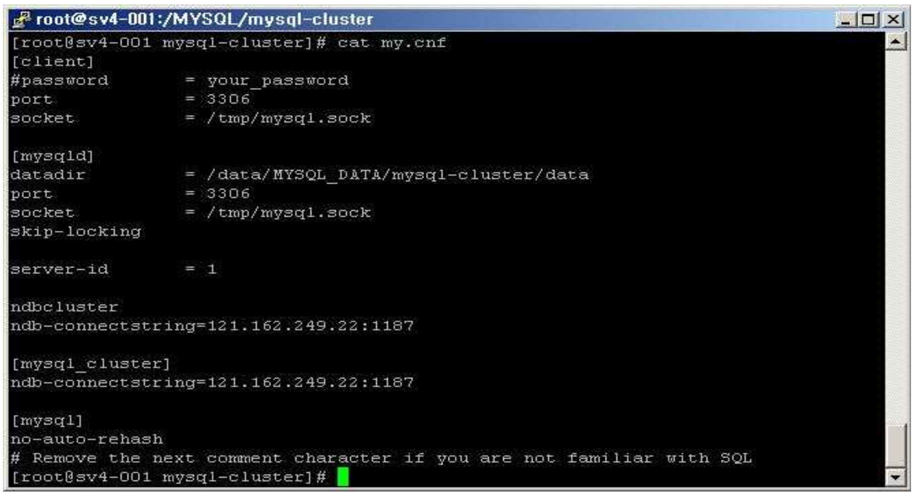
scripts/mysql_install_db --user=mysql

☞ my.cnf 생성 및 설정

```
# vi /MYSQL/mysql-cluster/my.cnf
[mysqld]
ndbcluster
ndb-connectstring=Management_IP_Address:Port

[mysql_cluster]
ndb-connectstring=Management_IP_Address:Port
```

☞ Data 디렉터리 생성

mkdir /data/MYSQL_DATA/mysql-cluster/data



```
root@sv4-001:/MYSQL/mysql-cluster
[root@sv4-001 mysql-cluster]# cat my.cnf
[client]
#password       = your_password
port            = 3306
socket          = /tmp/mysql.sock

[mysqld]
datadir         = /data/MYSQL_DATA/mysql-cluster/data
port            = 3306
socket          = /tmp/mysql.sock
skip-locking

server-id       = 1

ndbcluster
ndb-connectstring=121.162.249.22:1187

[mysql_cluster]
ndb-connectstring=121.162.249.22:1187

[mysql]
no-auto-rehash
# Remove the next comment character if you are not familiar with SQL
[root@sv4-001 mysql-cluster]#
```

[그림 4-8. my.cnf 설정 내용]

※ 위의 내용은 Data node 서버 2대에서 모두 동일하게 수행한다.

▪ Management Node Config 설정

☞ config.ini 생성

cp /MYSQL/mysql-cluster/support-files/ndb-config-2-node.ini \
/data/MYSQL_DATA/mysql-cluster/config.ini

```
[ndbd default]
NoOfReplicas= 2
LockPagesInMainMemory=1
DataMemory=1536M
IndexMemory= 512M
DataDir= /data/MYSQL_DATA/mysql-cluster

NoOfFragmentLogFiles=300
MaxNoOfConcurrentOperations=100000

SchedulerSpinTimer=400
#ScheduleExecutionTimer=100
RealTimeScheduler=1

TimeBetweenGlobalCheckpoints=1000
TimeBetweenEpochs=200
DiskCheckpointSpeed=10M
```

```
DiskCheckpointSpeedInRestart=100M
RedoBuffer=32M

MaxNoOfTables=1024
MaxNoOfOrderedIndexes=256

[ndb_mgmd  default]
DataDir= /data/MYSQL_DATA/mysql-cluster
portNumber=1187

[ndb_mgmd]
Id=1
HostName=121.162.249.22

[ndbd]
Id=2
HostName=121.162.249.22

LockExecuteThreadToCPU=1
LockMaintThreadsToCPU=0

[ndbd]
Id=3
HostName=121.162.249.23

LockExecuteThreadToCPU=1
LockMaintThreadsToCPU=0

[mysqld]
Id=4

[mysqld]
Id=5

[tcp  default]
SendBufferMemory=2M
ReceiveBufferMemory=2M
```

☞ Management Node 시작

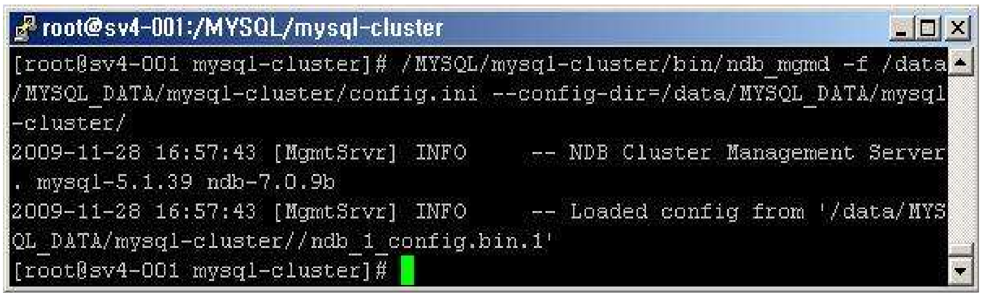
 # /MYSQL/mysql-cluster/bin/ndb_mgmd -f /config.ini 위치 \
 --configdir=/configdir 위치

※ config file 지정 : -f /config.ini 위치

 config dir 지정 : --configdir=/configdir 위치

 Binary configuration file을 위한 디렉터리 지정을 안 할 경우,

 /usr/local/mysql/mysql-cluster 디렉터리 생성



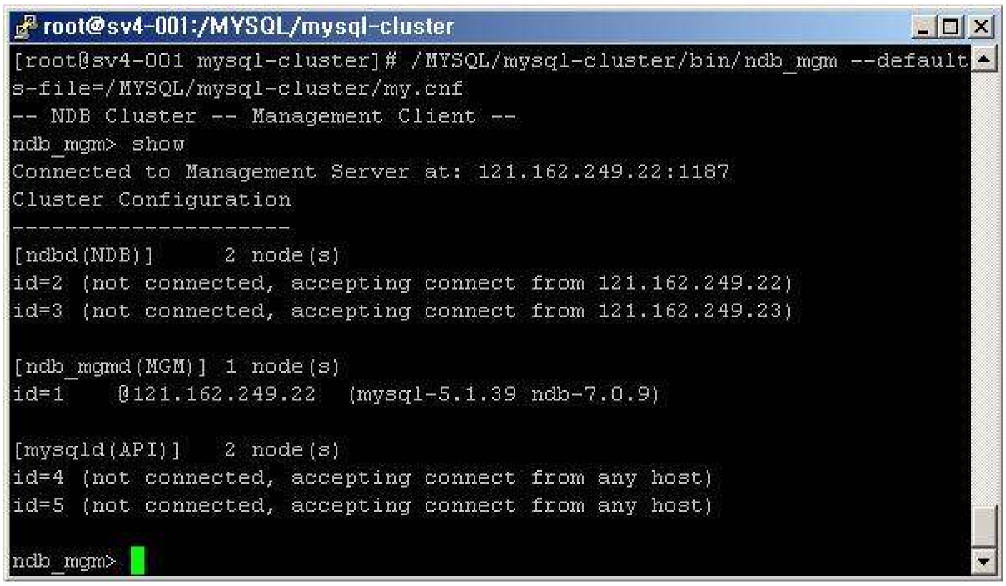
[그림 4-9. Cluster 시작 화면]

☞ Management Client 시작

 # /MYSQL/mysql-cluster/bin/ndb_mgm\
 --defaults-file=/MYSQL/mysql-cluster/my.cnf

☞ Cluster 정보 확인

 ndb_mgm> SHOW



[그림 4-10. Cluster 상태 확인]

▪ Data Node 실행 및 확인

☞ 각각의 Data node에서 ndb 구동

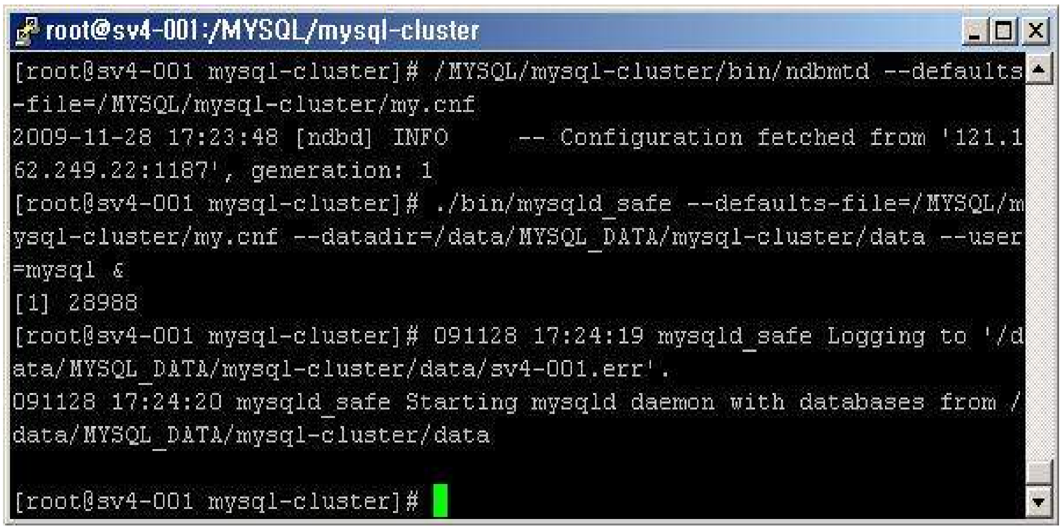
/MYSQL/mysql-cluster/bin/ndbmtd \
--defaults-file=/MYSQL/mysql-cluster/my.cnf

※ Management node와 Data node가 같은 서버에 있을 경우, node-id가 충돌하는 경우가 발생할 수 있다. 그럴 경우에는 Management host와 Bind address를 옵션에 명시해주면 에러 없이 실행된다.

$MySQL_Cluster_Bin/ndbmtd --initial\
--ndb-mgmd-host=Management_node_IP:Port--bind-address=Data_node_IP

☞ 각각의 Data node에서 MySQL 구동

/MYSQL/mysql-cluster/bin/mysqld_safe \
--defaults-file=/MYSQL/mysql-cluster/my.cnf \
--datadir=/data/MYSQL_DATA/mysql-cluster/data --user=mysql &



[그림 4-11. MySQL 실행 화면]

☞ Management node에서 Cluster 정보 확인

ndb_mgm> SHOW

```
root@sv4-001:/MYSQL/mysql-cluster
[root@sv4-001 mysql-cluster]# /MYSQL/mysql-cluster/bin/ndb_mgm --defaults-file=/MYSQL/mysql-cluster/my.cnf
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: 121.162.249.22:1187
Cluster Configuration
---------------------
[ndbd(NDB)]     2 node(s)
id=2    @121.162.249.22  (mysql-5.1.39 ndb-7.0.9, Nodegroup: 0, Master)
id=3    @121.162.249.23  (mysql-5.1.39 ndb-7.0.9, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1    @121.162.249.22  (mysql-5.1.39 ndb-7.0.9)

[mysqld(API)]   2 node(s)
id=4    @121.162.249.22  (mysql-5.1.39 ndb-7.0.9)
id=5    @121.162.249.23  (mysql-5.1.39 ndb-7.0.9)

ndb_mgm>
```

[그림 4-12. Management Node에서 Cluster 정보 확인]

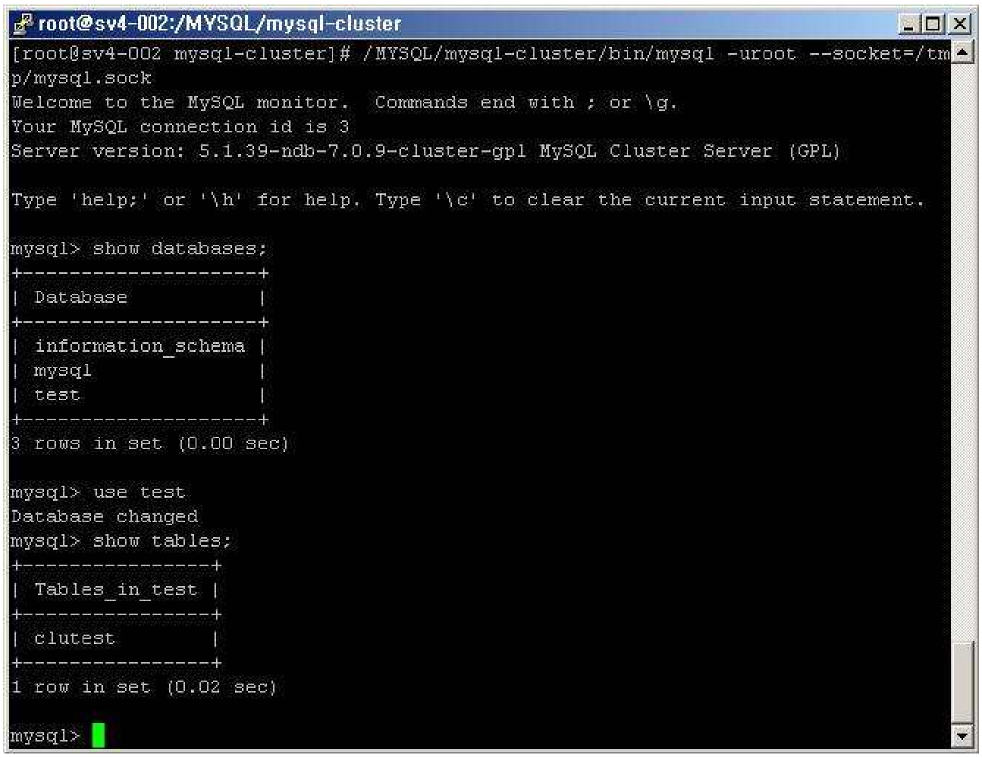
▪ Cluster 테스트

☞ Data node 1에서 clutest Table 생성

```
root@sv4-001:/MYSQL/mysql-cluster
[root@sv4-001 mysql-cluster]# /MYSQL/mysql-cluster/bin/mysql -uroot --socket=/tmp/mysql.sock
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 68
Server version: 5.1.39-ndb-7.0.9-cluster-gpl MySQL Cluster Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)

mysql> use test
Database changed
mysql> create table clutest(i int) engine=ndbcluster;
Query OK, 0 rows affected (0.56 sec)

mysql>
```

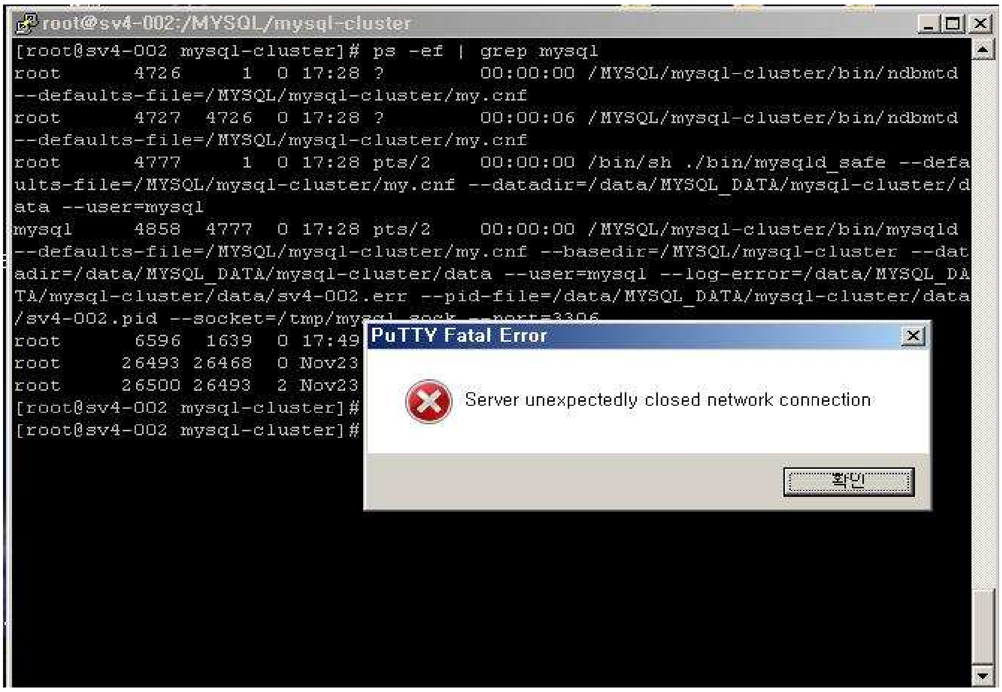
[그림 4-13. Table 생성]

☞ Data node 2에서 clutest Table 생성 확인



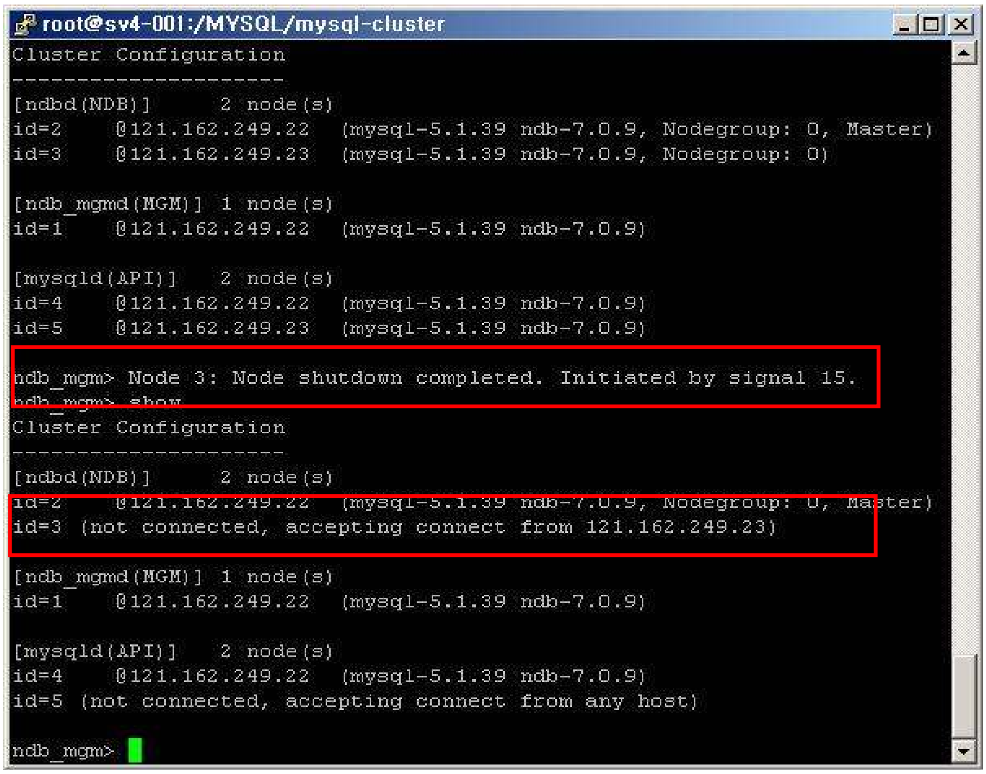
[그림 4-14. Table 생성 확인]

☞ Data node 2 시스템 Shutdown 후 상태 확인
Data node 2 서버가 종료되어 연결이 끊어진 화면



[그림 4-15. Data Node2 서버 장애 화면]

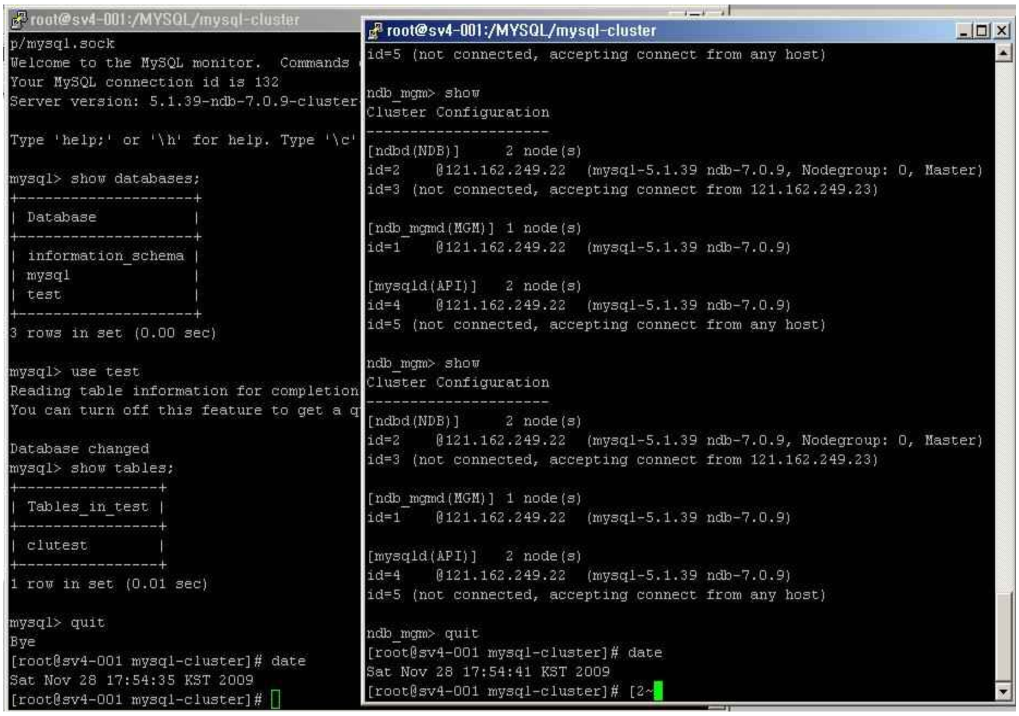
Data node 1 시스템에서 cluster 상태 확인



[그림4-16. Cluster 상태 확인]

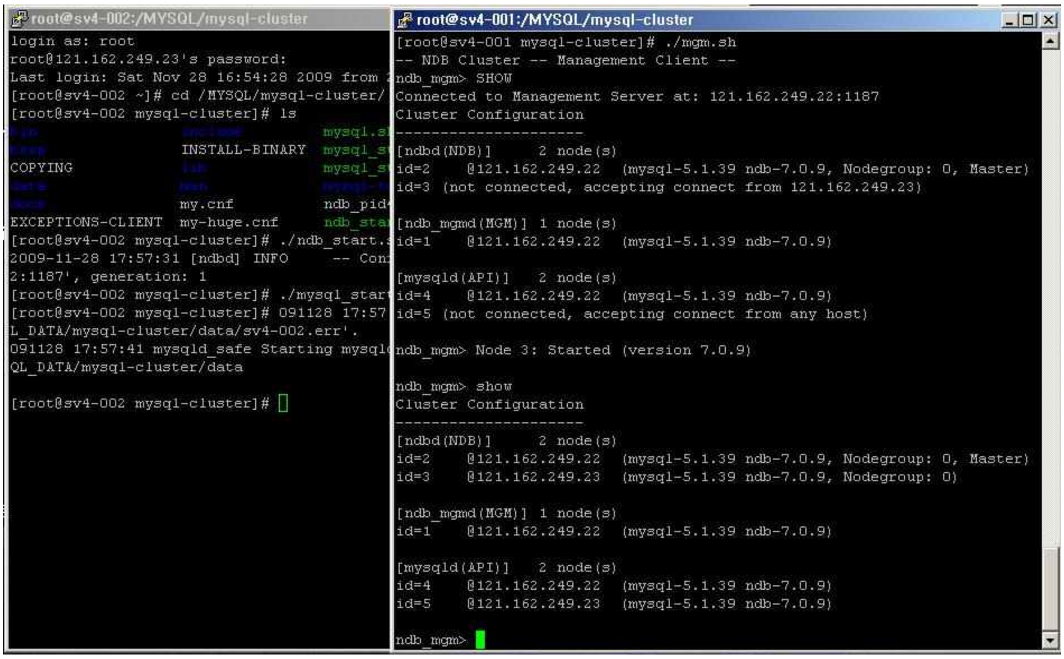
※ [그림 4-16]에서 확인 시 not connected 로 나오고 있음

☞ Data node 1 시스템 Database 상태 확인



[그림 4-17. 데이터베이스 상태 확인]

☞ Data node 2 시스템 정상 구동 후 상태 확인



[그림 4-18. 시스템 Cluster Join 확인]

※ mysql 클러스터 구성 테스트는 정상적으로 구동되는 것을 확인하였다. 단, 실제 상용 서비스에 적용할 경우, L4 Switch와 함께 연동을 하거나 Virtual IP 설정을 별도로 구성해야 클러스터 방식의 서비스가 가능하다.

☞ Active-Active 방식은 L4 Switch와 함께 연동 필요

☞ Active-Standby 방식은 Heartbeat 패키지와 함께 연동 필요

라. 관리 도구 테스트

□ GUI Tool 지원

[표 4-18. 관리 도구 지원]

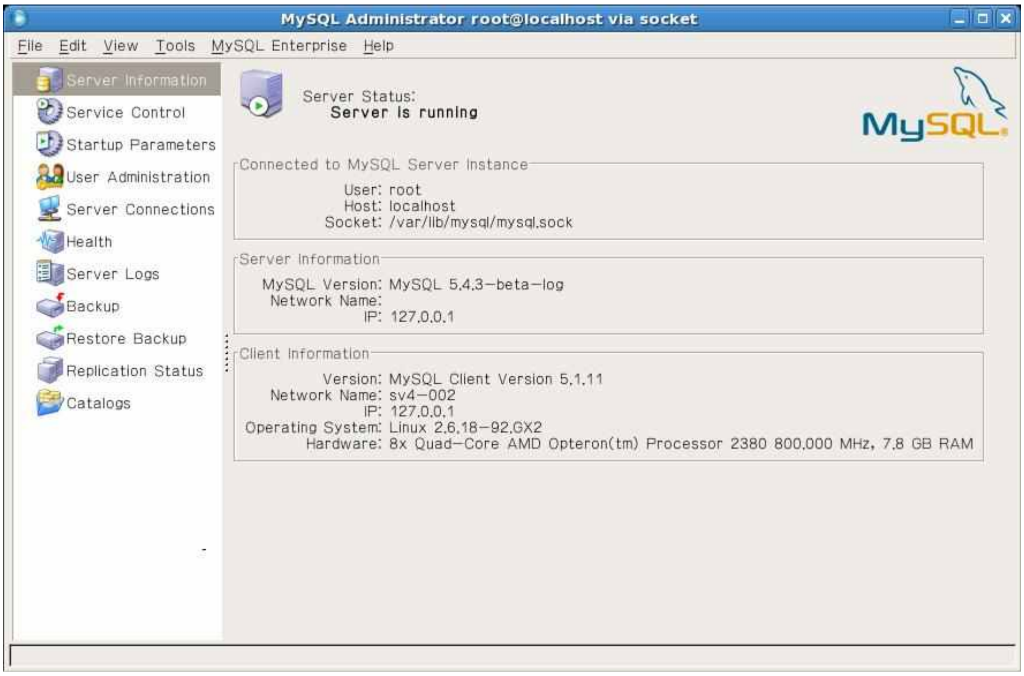
구　분	GUI Tool 버전	지원 여부
Administration Tools	MySQL Administrator 1.2	○
Query Tools(Client Side)	MySQL Query Browser 1.2	
Monitoring Tools	MySQL Administrator 1.2	
ETC Mgmt Tools	MySQL Migration Toolkit 1.1	

- MySQL Administrator
  MySQL Database를 운영하는데 필요한 사항을 관리한다.
  ○ MySQL 설정 편집
  ○ MySQL Monitoring
  ○ MySQL Server Start/Stop
- Query Tools(Client Side)
  그래픽 환경에서 Query의 최적화와 생성, 실행을 제공하는 GUI 툴
  MySQL 4.0 이상 버전에서 지원 됨
- Monitoring Tools
  MySQL Administrator에서 함께 제공 함
- ETC Mgmt Tools
  MySQL Migration Tool은 MySQL 여러 Database 시스템으로부터 Data와 스키마(Schema)를 마이그레이션 할 수 있는 툴을 말함

- MySQL GUI Tool 설치

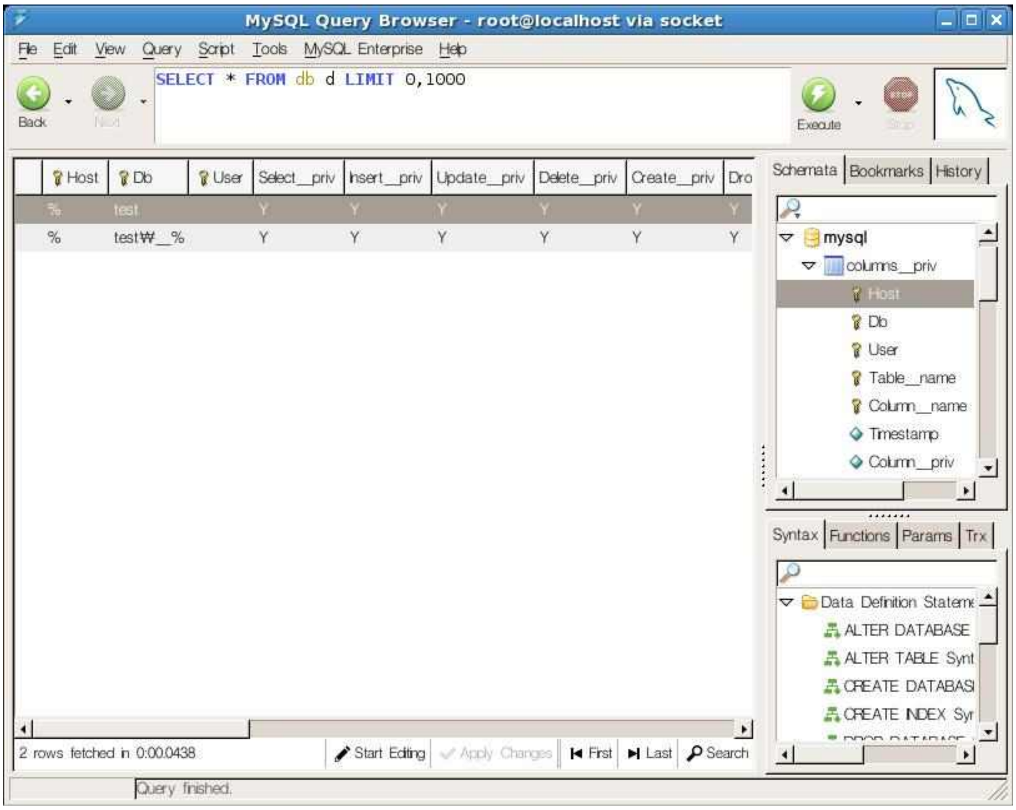
○ MySQL GUI Tool 다운로드(Binary Version)
http://dev.mysql.com/downloads/gui-tools/5.0.html
○ 압축 풀기
$ tar xvfz mysql-gui-tools-5.0r12-linux-x86_64.tar.gz -C /opt
$ cd /opt/mysql-gui-tools-5.0
$ ./mysql-query-browser --update-paths ☞ 최초 실행 시 설치 디렉터리 경로 업데이트를 위한 명령
$ ./mysql-administrator (Administrator Tool GUI 명령 실행)
$ ./mysql-query-browser (Query Tool GUI 명령 실행)

○ mysql-administrator GUI 화면



[그림 4-19. MySQL Administrator GUI 화면]

○ mysql-query-browser GUI 화면



[그림 4-20. MySQL Query Browser GUI 화면]

5. 결론

□ 설치

운영체제별 Binary 버전별 제공 및 Source 제공으로 대부분의 운영체제에 설치 및 운영이 가능한 우수한 공개SW 제품이다.

□ 성능

MyISAM 엔진과 InnoDB 엔진에서 OLTP 환경에서의 성능을 측정하였는데, 결과는 역시 Transaction 처리가 지원되는 InnoDB의 성능이 우수하게 나왔으며, 시스템 사용 환경에 따라 엔진 선택이 필요하며, 부하가 많은 시스템의 경우는 적절한 시스템 최적화가 필요해 보인다.

□ 기능

MySQL Cluster는 기업에서 도입되어 사용되고 있는 사례가 많으며, 테스트 결과에서도 보여지 듯 구성이 까다롭지 않고 매우 간편하면서도 장애에 효과적으로 대비할 수 있는 이중화 솔루션으로 평가된다.

□ 관리

MySQL administrator와 query-browser 등은 GUI 방식으로 구동되어 사용자가 쉽게 사용이 가능하다.
그러나, 사용자가 꼭 필요로 하는 기능은 상용으로 판매되는 제품에서 지원되는 점이 다소 아쉬운 점이라 할 수 있다.

세계적으로 알려진 MySQL 데이터베이스는 테스트에서 나온 결과 값을 보더라도 매우 안정적인 서비스를 할 수 있는 소프트웨어이다.
국내에서도 이미 많은 업체들이 사용하고 있으며, 공개SW 사용의 장점을 가장 잘 보여주는 대표적인 DBMS 제품이라 할 수 있다.
오라클과 같은 상용 DBMS와의 경쟁에서도 점차 그 영역을 넓혀 가고 있으며, 데이터베이스 안정화를 위한 이중화 기능도 지원함으로 엔터프라이즈 환경에서도 안심하고 사용할 수 있다.
데이터베이스 성능을 측정하는데 있어서, 많은 지식과 경험이 필요한데 이번 테스트에서는 MyISAM 엔진과 InnoDB 엔진을 이용하여 단위 테스트인 Transaction 성능만 테스트 하였다.
MySQL을 테스트함에 있어서 sysbench 테스트 툴은 많은 레퍼런스를 통한 신뢰도를 가지고 있지만, 공개SW의 신뢰성을 위한 검증을 위하여 상업용 테스트 툴을 이용하여 테스트하는 부분도 필요할 것으로 보인다.

* 참고 자료

http://www.mysql.com/

http://www.mysql.com/support/supportedplatforms/enterprise.html

http://sysbench.sourceforge.net/

Ⅸ SULinux

목 차

1. 개요

본 문서는 공개SW 역량프라자 운영 사업의 공개SW 테스트 환경 구축을 통하여 테스트한 공개SW에 대한 기능, 성능, 호환성 등을 검증하고, 향후 공공 기관 및 기업에서 도입 시 참고할 수 있는 자료로 활용하기 위하여 작성되었다.

가. 목적

□ 공개SW를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 공개SW의 기능, 성능, 안정성, 호환성 등 검증
- 공개SW 기술의 도입 가이드라인 제공
- 효과적인 공개SW 도입 유도
- 안정적인 공개SW 시스템 도입

나. 선정 기준

□ 공개SW 테스트 대상은 다음 5가지 사항을 고려하여 선정되었다.

- 1번 : 국내에서 개발된 제품
- 2번 : 국내 기업에서 많이 도입되는 제품
- 3번 : Global에서 많이 알려진 제품
- 4번 : 기술지원(업데이트) 가능한 제품
- 5번 : OS간 호환 가능 제품(Package 관리 방식)

[표 1-1. 선정기준]

Linux 운영체제(OS)	1번	2번	3번	4번	5번	결과	비고
Asianux	●	◕	◕	●	●	●	선정*
CentOS	○	◑	●	●	●	◕	선정
Debian	○	◔	◕	●	◔	◑	보류
Fedora	○	◑	●	●	●	◕	선정
Gentoo	○	◔	◕	◕	◔	◑	보류
GINUX	●	◑	◔	●	●	◕	선정
Red Hat Enterprise Linux	○	●	●	●	●	●	선정*
Slackware	○	◔	◕	◕	◔	◑	보류
SULinux	●	◑	◔	●	●	◕	선정
SuSE Linux	○	◑	●	●	●	◕	선정*
Ubuntu	○	◑	●	●	◔	◑	보류

※ 테스트 대상 기준에 근거하여 이번 테스트 대상에서 제외된 **선정*** 표시 제품은 금번 사업의 일정상의 제약으로 다음 사업에서 테스트를 수행하기로 함

2. 테스트 환경 구성

가. 테스트 일정

[표 2-1. 일정표]

구분	월	10				11				12			
	주	1	2	3	4	1	2	3	4	1	2	3	4
테스트 대상 발굴 및 선정													
테스트 환경 구성													
테스트 항목 선정													
테스트 시나리오 작성													
테스트 수행													

나. 테스트 대상 소개

□ SULinux 2.0

SULinux는 “보안 최적화된 서버 전용 리눅스 운영체제”이다. SULinux는 커널 2.6 기반의 보안이 강화된 안정된 버전이며, 서버로서 최고의 성능을 발휘 할 수 있게 최적화되어 탄생한 서버 전용 리눅스 OS이다.

□ SULnux 주요 특징

- 서버용으로 최고 성능 발휘
- GUI와 TUI 버전으로 구분되어 배포
- 서버용으로 최적화된 리눅스 OS
  ▪ 설치 후 서버용으로 자동 최적화
  ▪ 서버용으로 최적화 된 패키지 구성
  ▪ 360여개의 관리자 전용 명령어에 대한 기본 보안 설정
  ▪ 서버 관리 전용 명령어 내장(SSU)
  ▪ 뛰어난 안정성, 향상된 보안성
  ▪ 커널 2.6 기반의 향상된 성능 및 안정성
  ▪ SELinux 기술 및 자체보안강화
  ▪ 안정성이 검증된 패키지들만 포함
  ▪ DDOS방어에 최적화된 리눅스
- 쉽고 빠른 설치와 빠른 서비스 구현
  ▪ 3단계로 설치 완료되며, 누구나 쉽게 설치가능
  ▪ APM 자동 설치를 제공(APM_Install-tools)
  ▪ yum을 통한 빠르고 지속적인 업데이트 지원
- 향상된 호환성과 쉬운 인터페이스 제공

- 다양한 최신 RAID 컨트롤러 지원
- 8TB 이상의 대용량 파일 시스템 제공
- 국제 표준 언어셋(UTF-8) 기본 채택, 국제 리눅스 표준(LSB) 준수
- 64bit 지원으로 64bit 서버 성능 향상 및 대용량 메모리 사용
- 통합 서버 상태 상세 모니터링 지원(SSU)
- 편리한 GUI 환경 제공(GUI 버전)
- 편리한 TUI 환경 제공
- 엔터프라이즈급 가상화 지원(Xen)

다. 테스트 환경

□ 테스트 서버

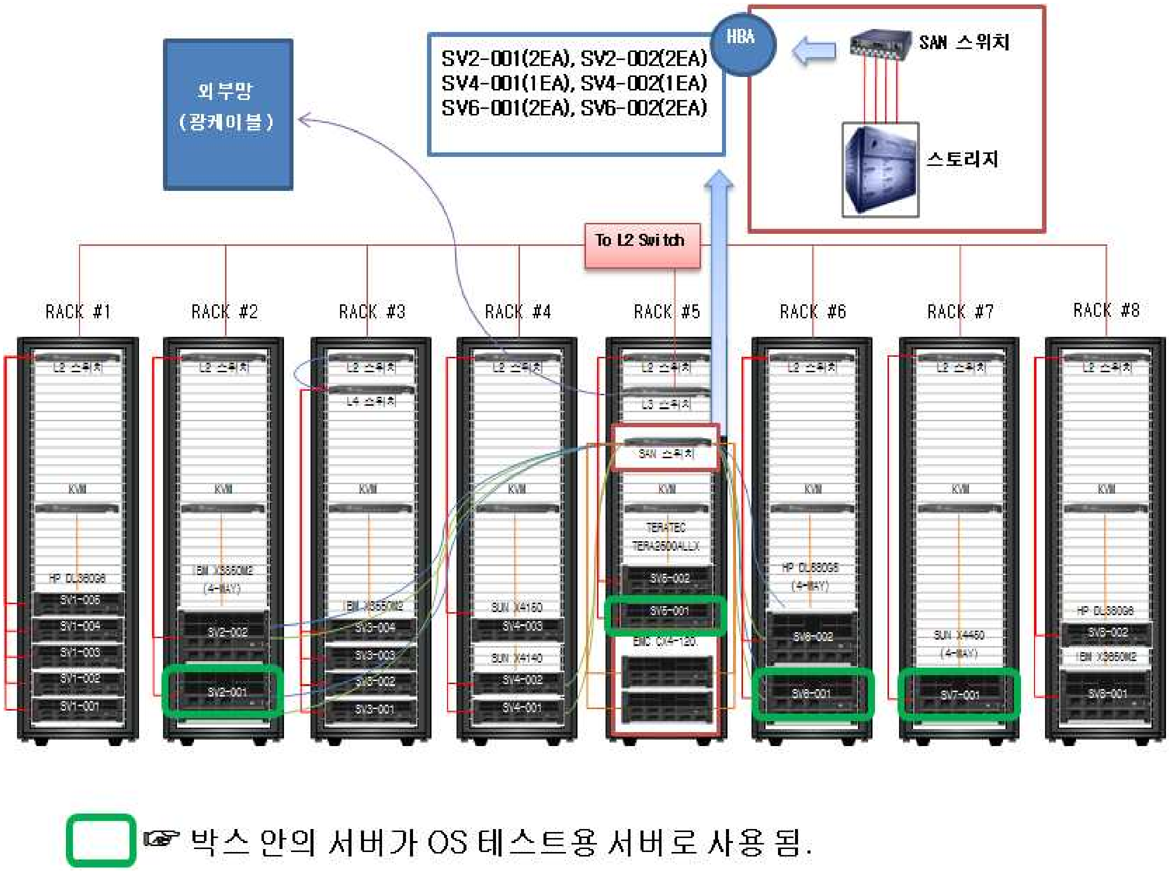
[표 2-2. 테스트 서버 규격]

구분	제품명	CPU	MEM	Disk	NIC
HP	DL580 G5	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
IBM	X3850 M2	Quad-Core 2.4Ghz 4-Way	8GB	146GB*4	Gigabit 2Port
SUN	X4450	Quad-Core 2.4Ghz 4-Way	8GB	146GB*5	Gigabit 2Port
TERATEC	TERA2500ALLX	Quad-Core 2.5Ghz 2-Way	8GB	147GB*5	Gigabit 2Port

□ 테스트 조건

- 32bit 환경에서 64bit환경으로 서버 구성이 전환됨에 따라 64bit 환경에서의 테스트를 수행
- 성능 테스트는 HP DL580 G5 서버에서 수행

□ 테스트 실장도



[그림 2-1. 테스트 서버 실장도]

□ 서버 위치

[표 2-3. 서버 위치]

제조사	모델명	Rack 위치	비 고
IBM	X3850 M2	#2	
TERATEC	TERA2500ALLX	#5	
HP	DL580 G5	#6	
SUN	X4450	#7	

3. 테스트 시나리오

가. 설치 및 호환성 검증

□ DVD-ROM을 통한 설치

□ 하드웨어 정보 확인

□ 4대의 하드웨어 벤더별 설치 상태 확인

나. 기능 테스트

□ 네트워크 서비스 지원

- Web, Ftp, NFS, Samba, Openssh, Telnet

□ 파일시스템 지원

- ext3, XFS, GFS 등

□ LVM 지원 확인

□ 보안 기능 지원

- tcp-wrapper
- GUI용 보안 툴

□ 자동 업데이트 기능 지원

- On-Line 업데이트

다. 신뢰성 테스트

□ LTP Tool 사용한 테스트

- ltp-full-20090930 버전 사용

라. 안정성 테스트

□ LTP Stress Tool 사용

- CPU, Memory 등의 사용량 체크

4. 테스트

가. 설치 및 호환성 검증

□ SULinux 설치 환경

- SULinux 설치 버전

[표 4-1. OS 설치 정보]

OS 버전	지원 유형	Kernel 버전
SULinux 2.0	X86_64(EM64T, AMD64)	kernel-2.6.18-92.el5

- 파티션 정보

[표 4-2. 파티션 정보]

<table>
<tr><th>디스크
(RAID)</th><th>Physical
Volume</th><th>Volume
Group</th><th>Logical
Volume</th><th>Mount
Point</th><th>용량
(MB)</th></tr>
<tr><td rowspan="9">/dev/sda(580GB)
RAID 5</td><td>/dev/sda1</td><td>-</td><td>-</td><td>/boot</td><td>100</td></tr>
<tr><td rowspan="5">/dev/sda2</td><td rowspan="5">VG0</td><td>lv_root</td><td>/</td><td>20480</td></tr>
<tr><td>lv_swap</td><td>-</td><td>8192</td></tr>
<tr><td>lv_usr</td><td>/usr</td><td>30720</td></tr>
<tr><td>lv_home</td><td>/home</td><td>20480</td></tr>
<tr><td>lv_var</td><td>/var</td><td>10240</td></tr>
<tr><td rowspan="3">/dev/sda3</td><td rowspan="3">VG1</td><td>lv_data1</td><td>/data1</td><td>251200</td></tr>
<tr><td>lv_data2</td><td>/data2</td><td>51200</td></tr>
<tr><td>lv_data3</td><td>/data3</td><td>51200</td></tr>
</table>

※ 디스크 용량은 서버별 차이가 있으므로 위의 기준으로 백분율(%) 환산 적용 하며 남는 용량은 여유 공간으로 비워 둠

□ 서버별 설치 및 호환성 검증

[표 4-3. OS 설치 방법 및 결과]

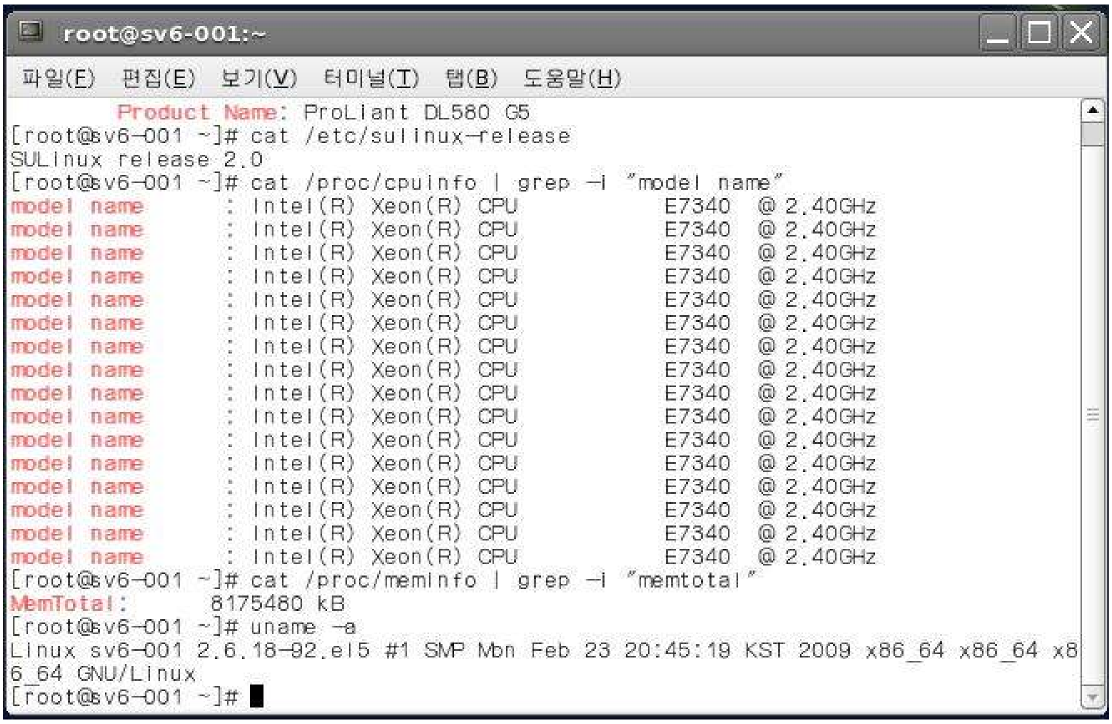
구분	설치 매체	설치 방식	패키지 선택 방식	설치 결과
HP DL580 G5	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
IBM X3850 M2	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
SUN X4450	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상
TERATEC TERA2500ALLX	DVD-ROM	GUI Mode 설치	전체 패키지 설치	정상

- 서버별 SULinux 상태 확인 명령

[표 4-4. SULinux 설치 상태 확인]

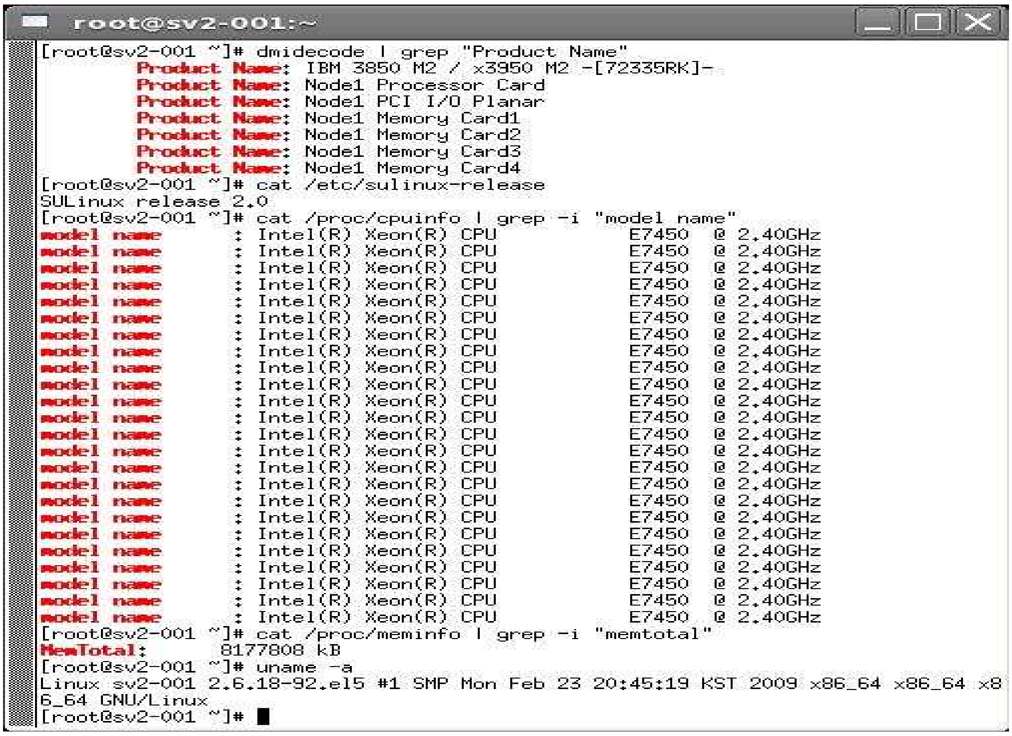
구분	명령어	결과
서버 모델명 확인	dmidecode | grep "Product Name"	[그림 4-1~4] 참고
OS 버전 확인	cat /etc/ginux-release	
CPU 정보	cat /proc/cpuinfo | grep -i "model name"	
Memory 정보	cat /proc/meminfo | grep -i "memtotal"	
Kernel 정보	uname -a	

▪ HP DL580 G5 모델에서 SULinux 설치 확인


```
root@sv6-001:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
        Product Name: ProLiant DL580 G5
[root@sv6-001 ~]# cat /etc/sulinux-release
SULinux release 2.0
[root@sv6-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
[root@sv6-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:      8175480 kB
[root@sv6-001 ~]# uname -a
Linux sv6-001 2.6.18-92.el5 #1 SMP Mon Feb 23 20:45:19 KST 2009 x86_64 x86_64 x8
6_64 GNU/Linux
[root@sv6-001 ~]#
```

[그림 4-1. HP DL580G5에서 SULinux 상태 확인]

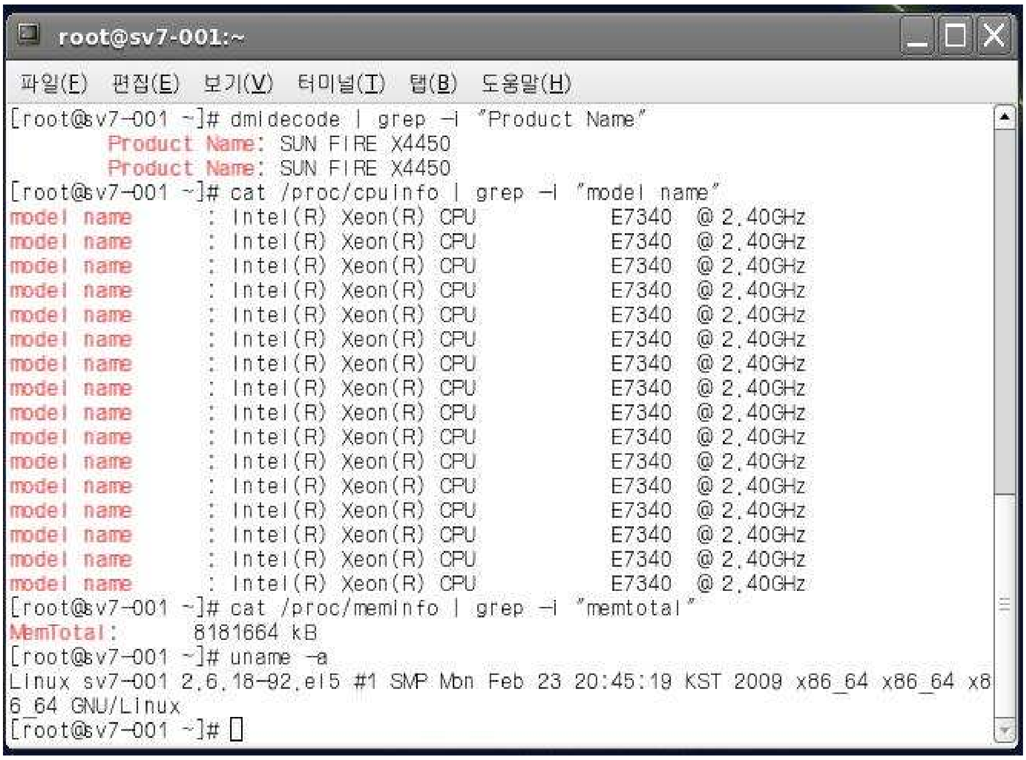
▪ IBM X3850 M2 모델에서 SULinux 설치 확인



```
[root@sv2-001 ~]# dmidecode | grep "Product Name"
          Product Name: IBM 3850 M2 / x3950 M2 -[72335RK]-
          Product Name: Node1 Processor Card
          Product Name: Node1 PCI I/O Planar
          Product Name: Node1 Memory Card1
          Product Name: Node1 Memory Card2
          Product Name: Node1 Memory Card3
          Product Name: Node1 Memory Card4
[root@sv2-001 ~]# cat /etc/sulinux-release
SULinux release 2.0
[root@sv2-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7450  @ 2.40GHz
[root@sv2-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:        8177808 kB
[root@sv2-001 ~]# uname -a
Linux sv2-001 2.6.18-92.el5 #1 SMP Mon Feb 23 20:45:19 KST 2009 x86_64 x86_64 x8
6_64 GNU/Linux
[root@sv2-001 ~]#
```

[그림 4-2. IBM X3850M2에서 SULinux 상태 확인]

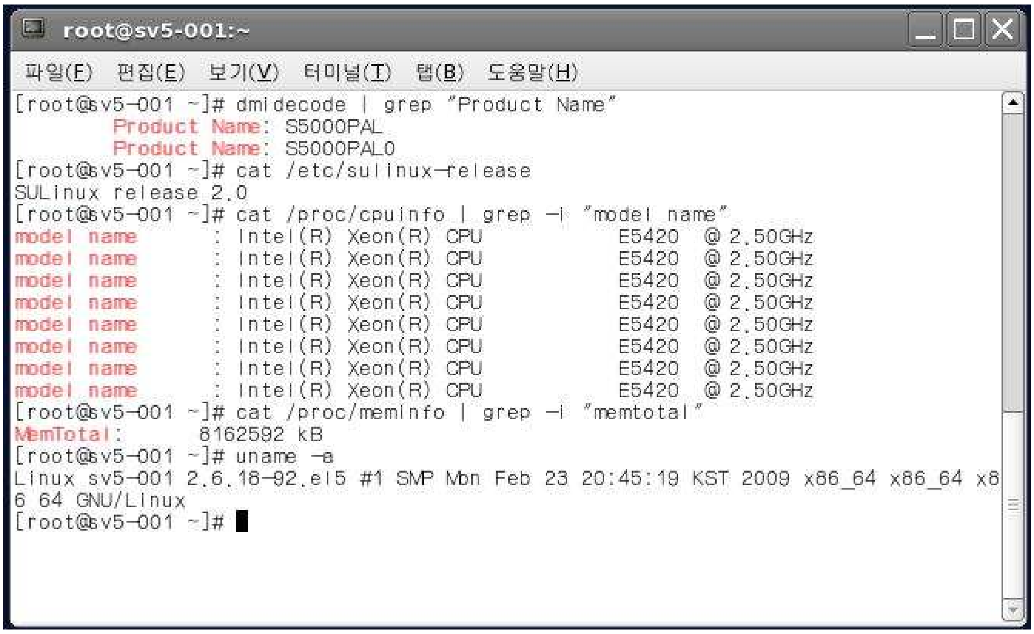
▪ SUN X4450 모델에서 SULinux 설치 확인



```
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@sv7-001 ~]# dmidecode | grep -i "Product Name"
        Product Name: SUN FIRE X4450
        Product Name: SUN FIRE X4450
[root@sv7-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
model name      : Intel(R) Xeon(R) CPU           E7340  @ 2.40GHz
[root@sv7-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:        8181664 kB
[root@sv7-001 ~]# uname -a
Linux sv7-001 2.6.18-92.el5 #1 SMP Mon Feb 23 20:45:19 KST 2009 x86_64 x86_64 x8
6_64 GNU/Linux
[root@sv7-001 ~]#
```

[그림 4-3. SUN X4450에서 SULinux 상태 확인]

▪ TERATEC TERA2500ALLX 모델에서 SULinux 설치 확인



```
root@sv5-001:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@sv5-001 ~]# dmidecode | grep "Product Name"
        Product Name: S5000PAL
        Product Name: S5000PAL0
[root@sv5-001 ~]# cat /etc/sulinux-release
SULinux release 2.0
[root@sv5-001 ~]# cat /proc/cpuinfo | grep -i "model name"
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
model name      : Intel(R) Xeon(R) CPU           E5420  @ 2.50GHz
[root@sv5-001 ~]# cat /proc/meminfo | grep -i "memtotal"
MemTotal:       8162592 kB
[root@sv5-001 ~]# uname -a
Linux sv5-001 2.6.18-92.el5 #1 SMP Mon Feb 23 20:45:19 KST 2009 x86_64 x86_64 x8
6 64 GNU/Linux
[root@sv5-001 ~]#
```

[그림 4-4. TERATEC TERA2500ALLX에서 SULinux 상태 확인]

※ 참고 사항

dmidecode 명령은 하드웨어 업체에서 정보를 제공하는 경우만 정상적인 결과값을 얻을 수 있는데 테라텍 서버의 경우 모델명이 일치하지 않음

나. 기능 테스트

□ 네트워크 서비스 지원

[표 4-5. SULinux 네트워크 서비스 확인]

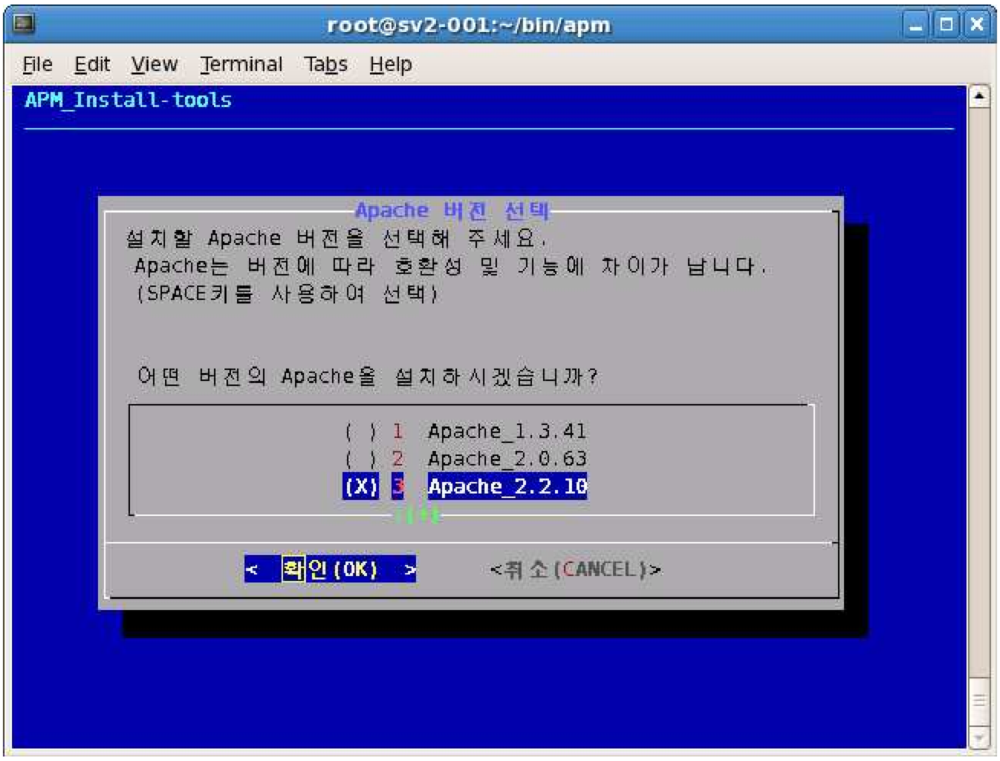
서비스 명	패키지 명	버전	지원 여부	비고
WEB	httpd	2.2.10	지원	apm_install 사용
FTP	vsftpd	2.0.5-13	지원	
NFS	nfs-utils	1.0.9-33	지원	
Samba	samba	3.0.28-0	지원	
SSH	openssh	4.3p2-26	지원	
Telnet	telnet-server	0.17-39	지원	

- 서비스별 지원 확인 명령

[표 4-6. 서비스별 지원 명령]

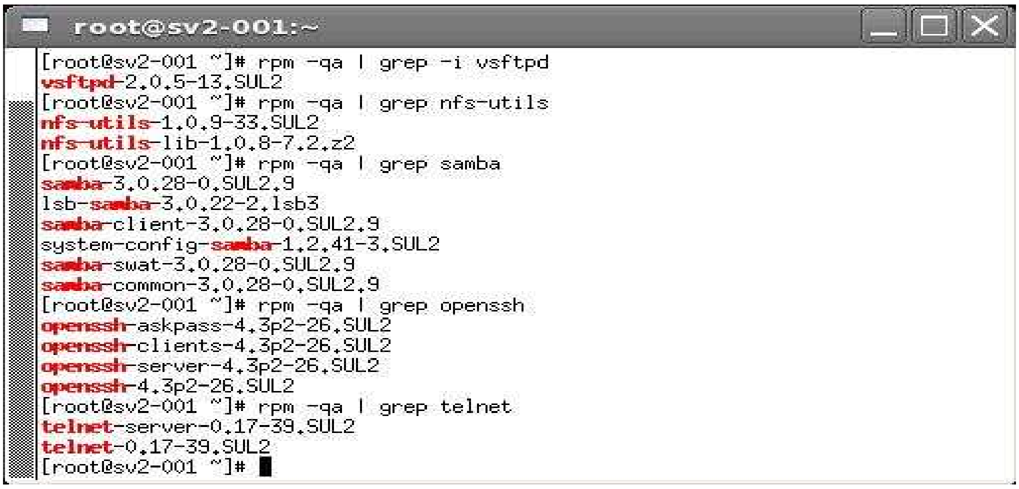
구분	명령어	결과
WEB 확인 명령	/root/bin/apm/apm_install	[그림4-5] 참고
서비스 확인 명령	rpm -qa | grep "*package_name*"	[그림4-6] 참고

▪ WEB 설치 화면



[그림 4-5. Web 설치 화면]

▪ 서비스 지원 확인



```
[root@sv2-001 ~]# rpm -qa | grep -i vsftpd
vsftpd-2.0.5-13.SUL2
[root@sv2-001 ~]# rpm -qa | grep nfs-utils
nfs-utils-1.0.9-33.SUL2
nfs-utils-lib-1.0.8-7.2.z2
[root@sv2-001 ~]# rpm -qa | grep samba
samba-3.0.28-0.SUL2.9
lsb-samba-3.0.22-2.lsb3
samba-client-3.0.28-0.SUL2.9
system-config-samba-1.2.41-3.SUL2
samba-swat-3.0.28-0.SUL2.9
samba-common-3.0.28-0.SUL2.9
[root@sv2-001 ~]# rpm -qa | grep openssh
openssh-askpass-4.3p2-26.SUL2
openssh-clients-4.3p2-26.SUL2
openssh-server-4.3p2-26.SUL2
openssh-4.3p2-26.SUL2
[root@sv2-001 ~]# rpm -qa | grep telnet
telnet-server-0.17-39.SUL2
telnet-0.17-39.SUL2
[root@sv2-001 ~]#
```

[그림 4-6. SULinux 서비스 지원 확인]

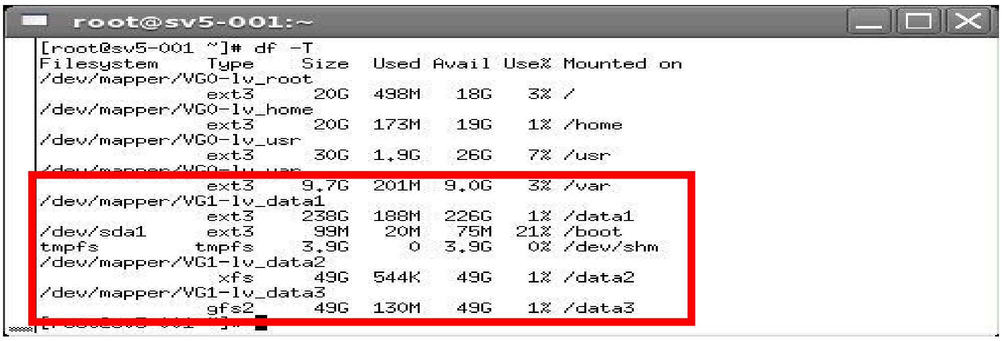
☐ 파일시스템 지원

- 파일시스템 지원 확인 명령

[표 4-7. 파티션별 지원 확인 명령]

구분	명령어	결과
ext3	mkfs -t ext3 /dev/VG1/lv_data1 mount /dev/VG1/lv_data1 /data1	[그림4-7] 참고
xfs	mkfs -t xfs /dev/VG1/lv_data2 mount /dev/VG1/lv_data2 /data2	
gfs	N/A(지원 안함)	
gfs2	gfs_mkfs -p lock_nolock -j 1 /dev/VG1/lv_data3 mount /dev/VG1/lv_data3 /data3	
파일시스템 확인 명령	df -T	

▪ ext3, xfs, gfs 파일시스템



```
[root@sv5-001 ~]# df -T
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/mapper/VG0-lv_root
              ext3     20G  498M   18G   3% /
/dev/mapper/VG0-lv_home
              ext3     20G  173M   19G   1% /home
/dev/mapper/VG0-lv_usr
              ext3     30G  1.9G   26G   7% /usr
              ext3    9.7G  201M  9.0G   3% /var
/dev/mapper/VG1-lv_data1
              ext3    238G  188M  226G   1% /data1
/dev/sda1     ext3     99M   20M   75M  21% /boot
tmpfs        tmpfs    3.9G     0  3.9G   0% /dev/shm
/dev/mapper/VG1-lv_data2
               xfs     49G  544K   49G   1% /data2
/dev/mapper/VG1-lv_data3
              gfs2     49G  130M   49G   1% /data3
```

[그림 4-7. 파티션별 지원 확인]

□ LVM 지원 확인

[표 4-8. LVM 지원 확인]

패키지명	버전	지원여부	비 고
lvm2	2.02.32-4	**지원**	

- LVM 확인 명령

[표 4-9. LVM 지원 확인 명령]

구분	명령어	결과
LVM 확인 명령	rpm -qa | grep "*package_name*"	[그림4-8] 참고

▪ LVM 확인

[그림 4-8. LVM 지원 확인]

※ LVM(Logical Volume Manager)

저장장치들을 좀더 효율적이고 유연하게 관리할 수 있는 커널의 부분 프로그램.

- LVM 사용의 장점
  - ▪ 쉬운 관리
  - ▪ 서로 다른 많은 디바이스 조합 지원
  - ▪ 직관적인 저장 장치로의 접근
  - ▪ 뛰어난 확장성
  - ▪ 신뢰되는 안정성과 효율성
  - ▪ 스냅샷 기능 지원
  - ▪ 대용량 단일 파일시스템 구성 지원

□ 보안 기능 지원

[표 4-10. 보안 기능 지원 확인]

패키지명	버전	지원여부	비　고
tcp_wrappers	7.6-40.4	지원	
system-config-securitylevel	1.6.29.1-2.1	지원	

- 보안 기능 확인 명령

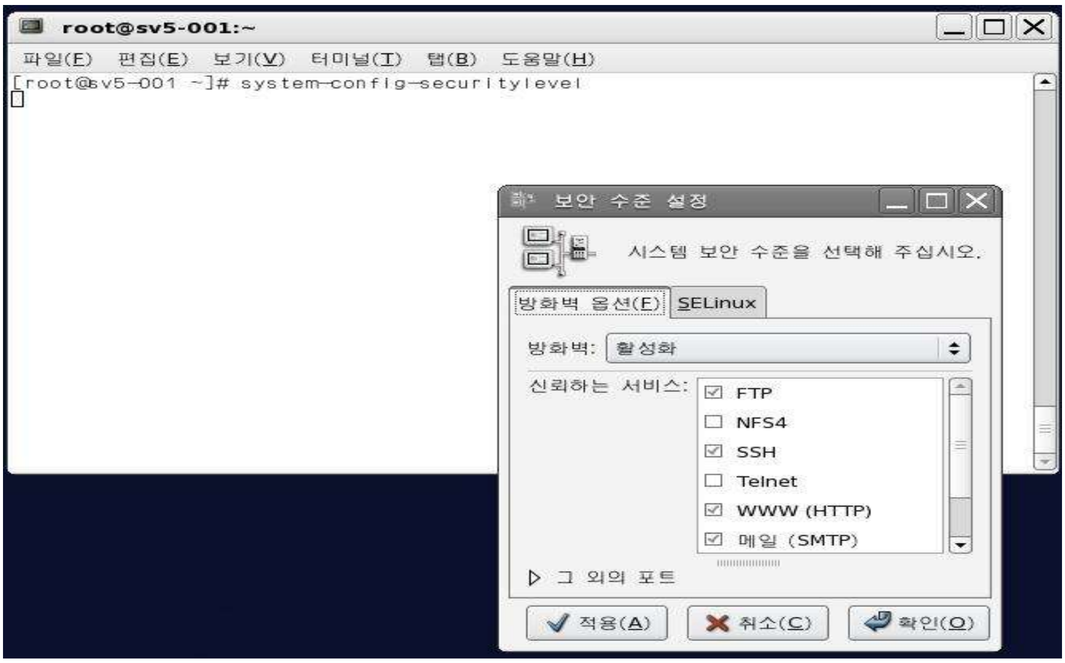
[표 4-11. 보안 기능 지원 명령]

구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "*package_name*"	[그림4-9, 4-10] 참고
GUI용 명령	system-config-securitylevel	

▪ 패키지 확인

[그림 4-9. 보안 기능 지원 확인]

▪ GUI용 보안 툴 명령



[그림 4-10. GUI 보안 기능 지원 확인]

□ 자동 업데이트 기능 지원

[표 4-12. 업데이트 지원 확인]

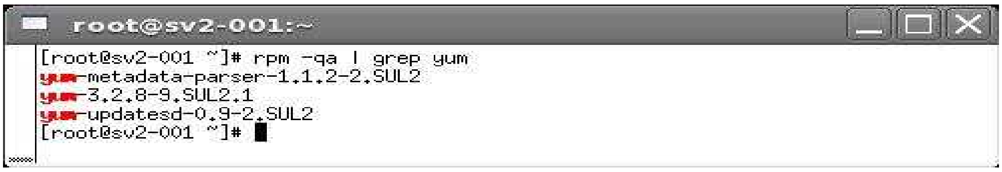
패키지명	버전	지원여부	비 고
yum	3.2.8-9.1	**지원**	

- 업데이트 기능 확인 명령

[표 4-13. 업데이트 지원 명령 확인]

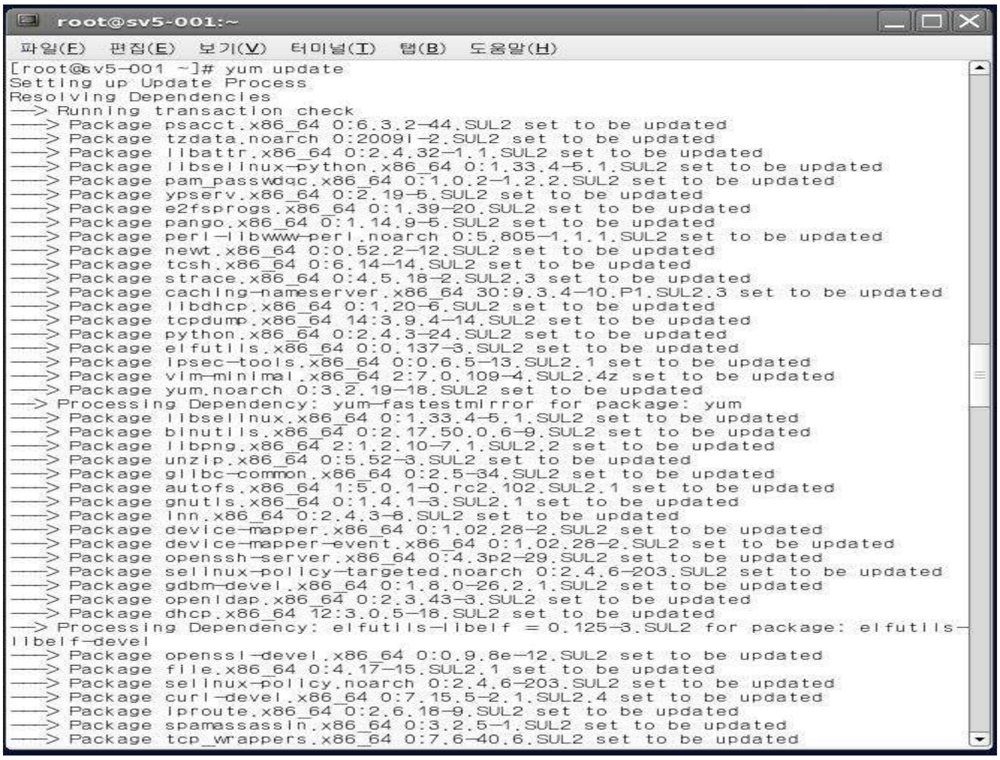
구분	명령어	결과
패키지 확인 명령	rpm -qa | grep "*package_name*"	[그림4-11, 4-12] 참고
GUI용 명령	pirut, pup	

▪ 패키지 확인



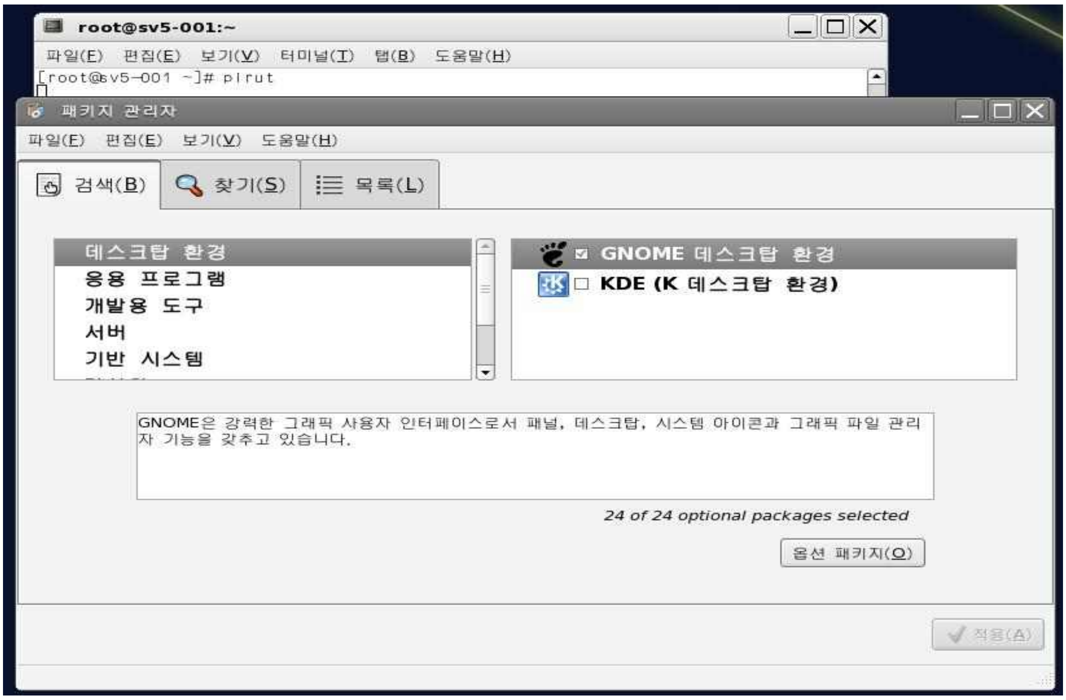
[그림 4-11. 업데이트 기능 지원 확인]

▪ 업데이트 실행 확인 (yum update)



[그림 4-12. 업데이트 기능 실행 확인]

▪ pirut
GUI용 소프트웨어 설치 및 삭제 유틸리티



[그림4-13. GUI용 패키지 설치/삭제 기능 실행 확인]

▪ pup
GUI용 패키지 업데이트

[그림4-14. GUI용 업데이트 기능 실행 확인]

다. 신뢰성 테스트

□ LTP Tool 사용

Linux Test Project (LTP)는 리눅스의 신뢰성, 호환성, 안정성을 입증하여 오픈소스 커뮤니티에 테스트 도구를 제공하는 것을 주 목적으로 하는 오픈소스 프로젝트이다.

- 테스트 방법

runltp 스크립트를 통한 초기 기본 테스트

▪ 파일시스템 스트레스 테스트
▪ 디스크 I/O 테스트
▪ 메모리 관리 스트레스 테스트
▪ IPC 스트레스 테스트
▪ 스케줄러 테스트
▪ 명령어 함수 확인 테스트
▪ 시스템 호출 함수 확인 테스트

- 실행 방법

```
$ cd /opt/ltp
$ ./runalltests.sh
```

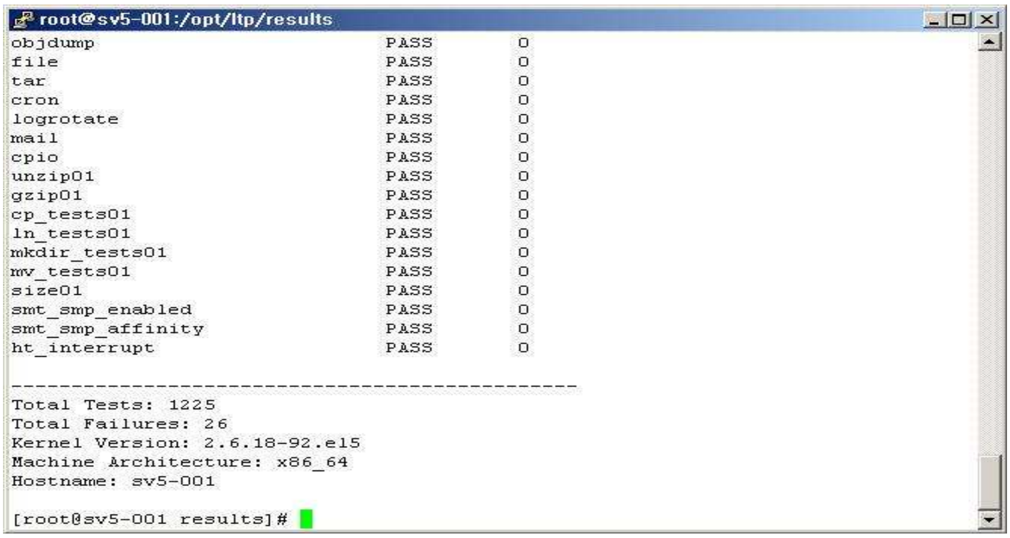
- 테스트 결과

▪ LTP 테스트 도구 버전 : ltp-full-20090930
▪ 테스트 시험 결과

[표 4-14. LTP 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	1225	26

※실패 목록은 SULinux에서 발생한 문제가 아닌 기능상 구현되지 않았거나, LTP 툴의 자체 버그일 가능성이 높음([표 4-15] 참고)



[그림4-15. LTP 시험 결과 화면]

▪ 테스트 시험 실패 목록

[표 4-15. LTP 시험 실패 항목]

항 목	설 명
eventfd01	eventfd test는 임시 파일을 임시 디렉터리(tmp)가 아니라 현재 디렉터리에 생성하는데, 현재 디렉터리는 nonwriteable로 설정되어 오류 발생 가능
inotify02	kernel 소스 기반의 non-x86 arch system에서 오류 발생
ppoll01	ltp_signal.h에서 SIGSETSIZE 매크로를 사용하고, poll.h의 헤더 위치를 바꾸어서 테스트하면 정상적이지만 다른 테스트와의 호환을 위해서 수정하지 않고 테스트 수행, 오류 발생
quotactl01	__NR_set_tid_address 시스템 호출에 하드코딩해서 258번을 대치해야 정상 수행되지만 수정되지 않은 커널은 테스트 수행 시 오류 발생
rt_sigaction01	Functionality test 방식으로 SIGRTMIN(34)에서 SIGRTMAX(64)까지 사용한다. 만약 rt_sigaction으로 1번 시그널부터 64번 시그널을 테스트하고자 한다면, 별도의 패치를 기반으로 커널 2.6.23 이후의 버전으로 테스트를 한다.
rt_sigprocmask01	rt_sigprocmask 시스템 호출에 대한 test로 real time signal test인 경우 별도의 패치를 통해 test를 수행하지만 신뢰성 약함
rt_sigsuspend01	rt_sigsuspend 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 현재 segmentation fault가 발생해서 test 항목의 신뢰성 약함
set_thread_area01	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)
set_thread_area02	set_thread_area 시스템 호출에 대한 test로 별도의 패치를 적용해서 test를 진행하지만 핸들링 코드의 중복으로 오류 발생(툴 자체의 버그일 가능성)
stime01	time() 함수가 리턴한 data 값보다 stime() (x86-64에서 settimeofday())이 리턴한 값이 더 적을 경우 FAIL
sysconf01	errno 대신에 "actual" return code를 리턴하면서 sysconf의 fail 오류가 0이 아닐 경우 오류 발생

utimes01	EACCESS times가 NULL이고 호출자의 EUID가 파일에 접근하는 호출자와 일치하지 않은 경우 오류 발생, 패치를 적용할 경우에 오류 해결 (최신버전)
su01	passwd -e 명령이 지원되지 않은 경우 FAIL 발생, 시스템에 영향 없음
hugemmap03	hugetlb 기반으로 normal page가 high memory region에 매핑할 수 없는지를 테스트하며 비정상적인 경우 FAIL
hugemmap04	적어도 하나의 테스트용 hugepage가 필요하고 hugepage의 개수에 대한 정보가 필요한데, 유효한 hugepage가 아닐 경우 FAIL
hugeshmat01	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmat02	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmat03	hugepage가 가용할 경우에만 hugepages를 hugeshmat에 할당, 실패땐 FAIL
hugeshmctl01	임의의 usleep time은 부정확한 실행 순서를 야기하며 FAIL
hugeshmctl02	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패땐 FAIL
hugeshmctl03	hugepage가 가용할 경우에만 hugepages를 hugeshmctl에 할당, 실패땐 FAIL
hugeshmdt01	hugepage가 가용할 경우에만 hugepages를 hugeshmdt에 할당, 실패땐 FAIL
hugeshmget01	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget02	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget03	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL
hugeshmget05	segment size가 정확하지 않거나 호출이 실패하는 경우 FAIL

라. 안정성 테스트

□ LTP Stress Tool 사용

LTP 테스트 도구를 사용한 스트레스 테스트를 수행한다.

- NFS 스트레스 테스트
- 메모리 관리 스트레스 테스트
- 파일시스템 스트레스 테스트
- Math (floating point) 테스트
- pthread 스트레스 테스트
- 디스크 I/O 테스트
- IPC (pipeio, semaphore) 테스트
- 시스템 호출 함수 확인 테스트
- 네트워킹 스트레스 테스트

- 실행 방법

```
$ cd /opt/ltp/testscripts
$  ./ltpstress.sh -d /tmp/sardata -l /tmp/ltplog.1122 -m 1024 -t 24 -S
```

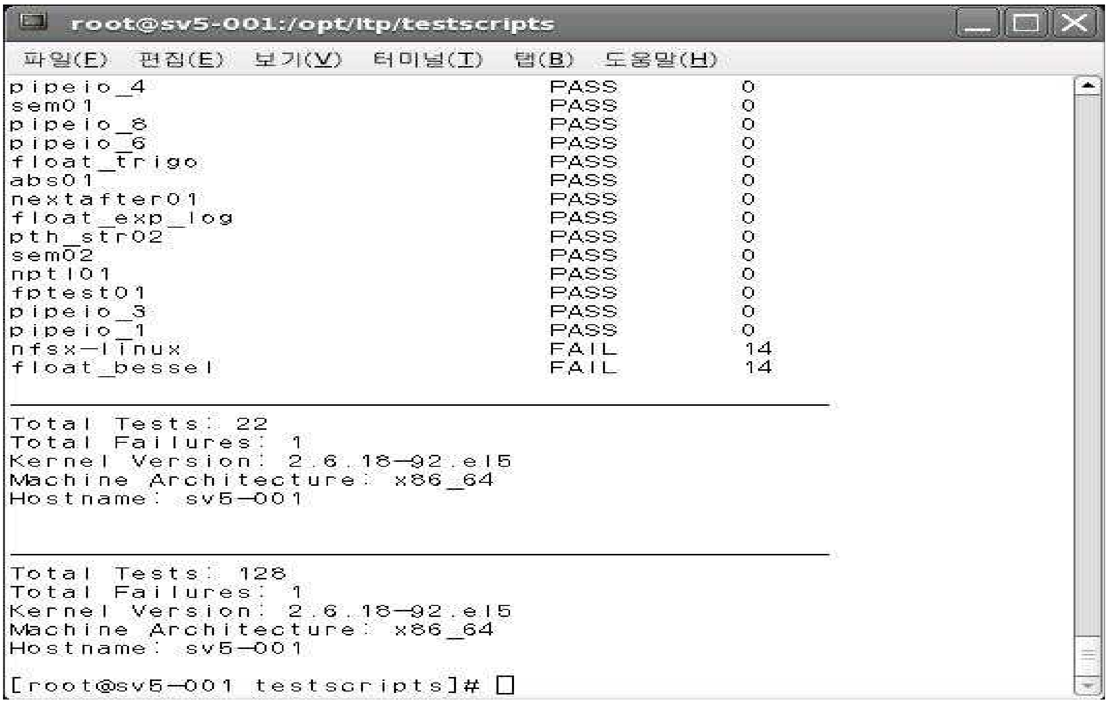
- 테스트 결과

- LTP 테스트 도구 버전 : ltp-full-20090930
- 테스트 시험 결과

[표 4-16. LTP Stress 시험 결과]

구분	시험 결과	
	성공	실패
x86_64(64bit)	22	1
	128	1

※ 실패 목록은 SULinux에서 발생한 문제가 아닌 기능상 구현되지 않아 발생한 오류일 가능성이 높음([표4-17] 참고)



[그림 4-16. LTP Stress 시험 결과 화면]

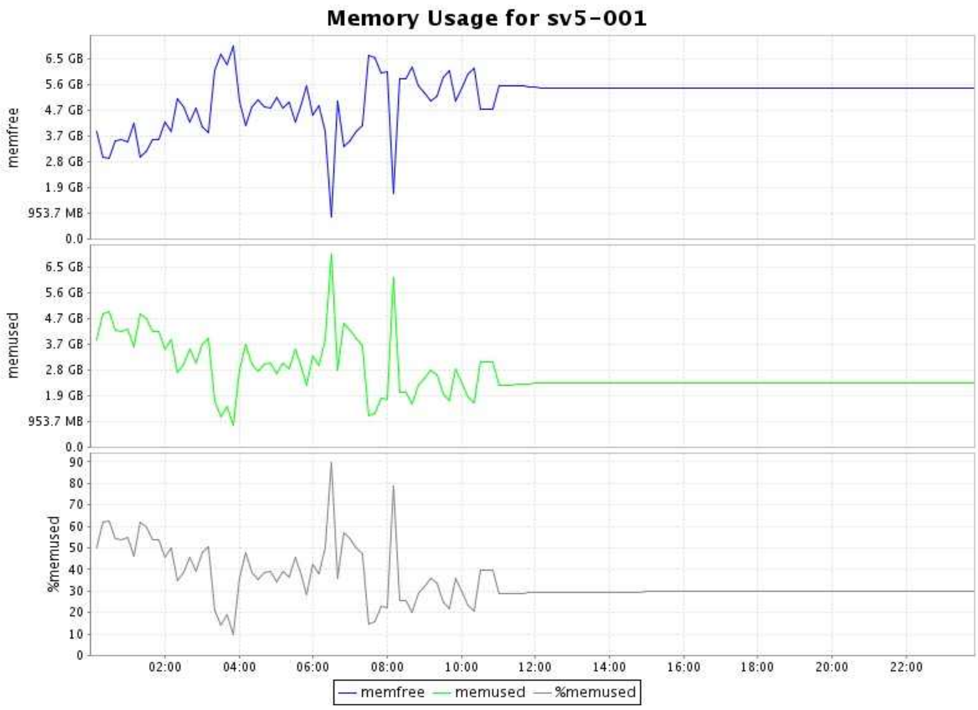
▪ 테스트 시험 실패 목록

[표 4-17. LTP Stress 실패 항목]

항 목	설 명
sysconf01	errno 대신에 “actual" return code를 리턴하면서 sysconf의 fail 오류가 0이 아닐 경우 오류 발생
mc_cmds	ltp network test일 경우 /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts 값이 1로 설정된 경우 FAIL 출력
nfs-linux	nfsx-linux를 찾지 못해서 발생하는 오류로 nfs v3과 v4에서 버전 미스매치로 인해 발생하는 문제로 시스템에 영향 없음
float_bessel	커널 oom_killer 발생 시에도 ltp 테스트가 수행되는 환경을 테스트하는 항목으로 FAIL일 경우 커널 oom_killer 기반으로 float연산 부분의 오류 가능성 제기

▪ 시스템 사용률

[그림 4-17. CPU 사용률 확인]



[그림 4-18. Memory 사용률 확인]

[그림 4-19. 평균 사용률 확인]

[표 4-18. 시스템 사용량 요약]

구분	최대 사용량	평균 사용량
CPU	100%	20%
Memory	8GB 이상	3.5GB
Load Average	8 이상	5.5

※ 참고 사항

- [그림4-17~19] 그래프는 하루를 기준으로 작성되었다.
- 스트레스 테스트 그래프는 00시부터 ~ 11시까지 해당 된다.
- 수치가 낮을수록 시스템 부하가 적은 것이다.

5. 결론

□ 설치

최신의 하드웨어 테스트 환경에서 드라이버가 기본으로 제공되므로 누구나 쉽게 설치가 가능하며, 초보자도 설치가 가능하도록 설정 기능을 최소화 하였고, 국내 기업인 (주)수퍼유저코리아에서 제작 및 배포하는 제품이므로 기술지원도 어렵지 않게 받을 수 있다.

□ 기능

서버용으로 사용하도록 최적화 하였고, Red Hat Enterprise에서 지원하는 서버 기능을 지원하며, 기업이나 공공기관에서 도입하여 상용 서비스를 하더라도 문제가 없을 것으로 보인다.

단, 서버용으로 사용 시 필요한 패키지로 최적화 되어 있어서 일부 기능은 별도 구성이 필요할 수 있다.

□ 신뢰성

LTP 툴에서 보여주듯 1000가지가 넘는 기능 테스트 중 대부분이 문제없이 테스트되었으며, 일부 테스트의 경우 기능상 구현되지 않았거나 지원하지 않는 부분의 오류이므로 운영체제에 문제가 있다고 판단하기는 어려우며 서비스용으로 사용하더라도 문제가 없을 것으로 보인다.

□ 안정성

스트레스 테스트 결과에서 보이듯 시스템이 안정적으로 운영되기 위한 기본 요건을 갖춘 제품이다.

SULinux는 국내 몇 개 안되는 배포판 중 하나로 테스트 결과, 사용 중 특이사항이 될 만한 문제점을 발견할 수 없었으며, 특히 초보자도 설치가 가능하도록 설치 방법을 단순화 한 것이 특징이다. 기술지원을 받을 수 있는 제품이므로 안심하고 사용할 수 있는 제품이라 할 수 있다.

* 참고 자료

http://www.sulinux.net

http://ltp.sourceforge.net

* 별첨

별첨1. LTP 활용 가이드

Ⅹ Tomcat

목　　　차

1. 개요

보편적으로 개발/구축된 Web시스템 중에는 기능사양에는 만족하고 있지만, 퍼포먼스라는 점에서는 사용자가 큰 불만을 품고 있는 사례가 흔하게 발생하고 있다.
그 이유로서는 부하테스트(성능테스트)가 거의 이루어지지 않는 프로젝트가 많았기 때문이다.
불특정다수의 사용자가 이용하는 web애플리케이션에 있어서는 만족할만한 편리한 기능이 갖추어져 있다고 하여도, 충분한 퍼포먼스가 나오지 않으면 사용자의 만족도가 저감하게 된다.

이에, 본 테스트 보고서에서는 부하테스트 툴인 JMeter를 사용하여, 서블릿 컨테이너의 대표적인 공개SW인 Tomcat을 대상으로 부하테스트를 실행하여 간다.

가. 목적

Tomcat 내장의 어플리케이션의 부하를 테스트하여 다음과 같은 목적을 달성하고자 한다.

- 현재 구축된 시스템이 어느 정도의 부하에 견딜 수 있는가를 테스트한다.
- 상정한 동시 사용자수를 만족하는지 검증한다.
- 현재 시스템의 변경으로 성능향상의 정도를 측정한다.

나. 테스트 대상 및 테스트 툴의 선정기준

본 테스트에 있어서의 테스트 대상 및 테스트 툴을 선정함에 있어서 다음과 같은 특성을 열거한다.

1) 테스트 툴 - JMeter

JMeter는 Apache Jakarta Project에서 개발이 진행되고 있는 부하테스트 애플리케이션이다.
Web애플리케이션(HTTP베이스의 애플리케이션)뿐만 아니라, FTP, JDBC, LDAP 등의 인터페이스기반의 애플리케이션에 대하여 부하테스트를 실시하는 것이 가능하다.

Web애플리케이션의 부하테스트를 실시하는 도구로서는 JMeter 이외에도 OpenSTA, Mercury LoadRunner 등이 있지만, 다음의 관점으로부터 JMeter를 사용하였다.

- 100% pure Java이다.
  Java로 기술되어 있으므로 OS환경에 구애받지 않는다.
- 오픈소스이다.
  오픈소스이므로 도입 비용이 거의 안 들고, 커뮤니티도 활발하다.
- 조작이 간단하다.
  조작이 간단하고 CUI는 물론 GUI로 테스트를 실시할 수가 있다.

2) 테스트 대상 - Tomcat

전 세계적으로 사실상의 서블릿 컨테이너(또는 웹 컨테이너)의 표준인 아파치 톰캣(Apache Tomcat)은 아파치 소프트웨어 재단에서 개발된 서버이다.
톰캣은 자바 서블릿(Servlet)과 자바 서비스 페이지(JSP)를 지원하며 표준 스펙을 만족하고 있다.

- 오픈 소스 소프트웨어
- JAAS 및 SSL 지원
- http 세션 replication
- 웹기반 콘솔을 이용한 어플리케이션 deploy 세션 expire
- 웹 브라우저가 URL 경로 요청 시 UTF-8 Encode 여부에 상관없이 저장
- Tomcat 6.0부터 Servlet 2.5, JSP2.1 지원
- Request 필터로 동작
- New I/O를 범용적으로 사용함으로 기능상 대폭 적 향상
- 설치와 테스트가 편리

다. 전제조건

- 본 부하 테스트는 Hardware 및 네트워크에 관련된 모니터링에 대해서는 언급하지 않음
- 본 부하 테스트는 JMeter를 사용한 Tomcat 내장 애플리케이션의 성능을 검증하기 위한 목적이므로 테스트 결과에 대한 분석 및 반영으로 의한 성능향상 방법은 별개의 과제로 한다.

2. 테스트 환경

OS	Linux (2.6.18)
테스트 대상	Tomcat / apache-tomcat-6.0.20
애플리케이션	Tomcat 내장 애플리케이션
테스트 툴	JMeter

3. 테스트 절차

가. 소프트웨어 설치

1) 서버

(1) JDK

> JAVA_HOME=/usr/local/java/jdk1.5.0_21

(2) Tomcat

① http://tomcat.apache.org/download-60.cgi에서
apache-tomcat-6.0.20.tar.gz 을 다운로드

② 압축을 푼다.
> tar xvfz apache-tomcat-6.0.20.tar.gz

③ tomcat을 실행
> cd /usr/local/tomcat/apache-tomcat-6.0.20/bin
> ./startup.sh

④ tomcat 기동 확인

```
Using CATALINA_BASE:    /usr/local/tomcat/apache-tomcat-6.0.20
Using CATALINA_HOME:    /usr/local/tomcat/apache-tomcat-6.0.20
Using CATALINA_TMPDIR:  /usr/local/tomcat/apache-tomcat-6.0.20/temp
Using JRE_HOME:         /usr/local/java/jdk1.5.0_21
```

2) 클라이언트

(1) JDK

> JAVA_HOME=/usr/local/java/jdk1.5.0_21

(2) JMeter

> JMeter_Manual.pdf 참조

나. 테스트 시나리오

1) 시나리오

(1) 서블릿 구성인 Request Info 화면을 표시한다.
(2) JSP 구성인 Basic Arithmetic 화면을 표시한다.

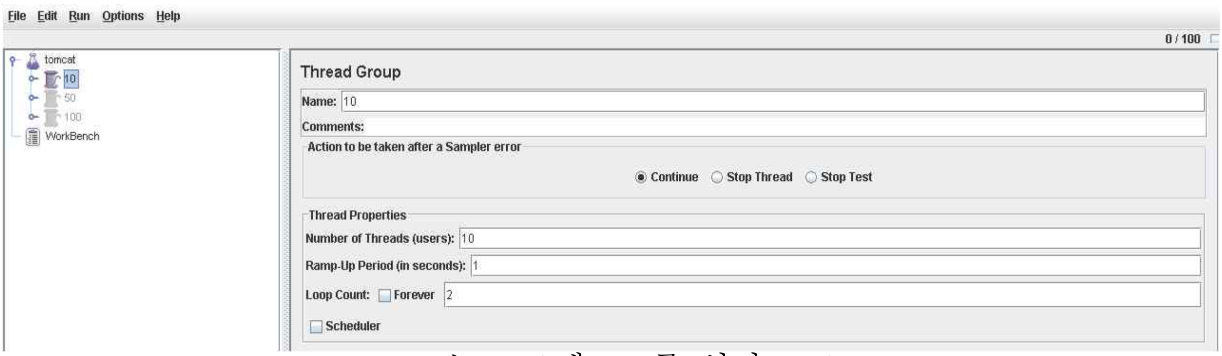
2) 부하레벨의 설정

시스템의 부하레벨을 높여 가면서 테스트를 반복하여, 한계치를 조사하기 위하여, Ramp-Up기간을 1sec 고정으로, loop counter를 2로 하여 thread 수(접속자 수)를 증가시켜 1초당 동시 접속수를 조정하여 간다.

3) JMeter 설정

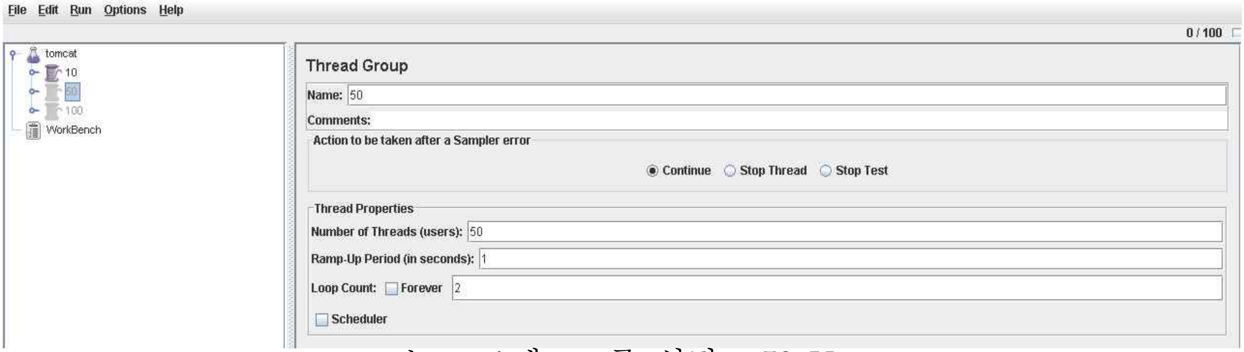
(1) 테스트 플랜 작성
(2) Thread Group 생성

● 10 User가 동시 2회 접속



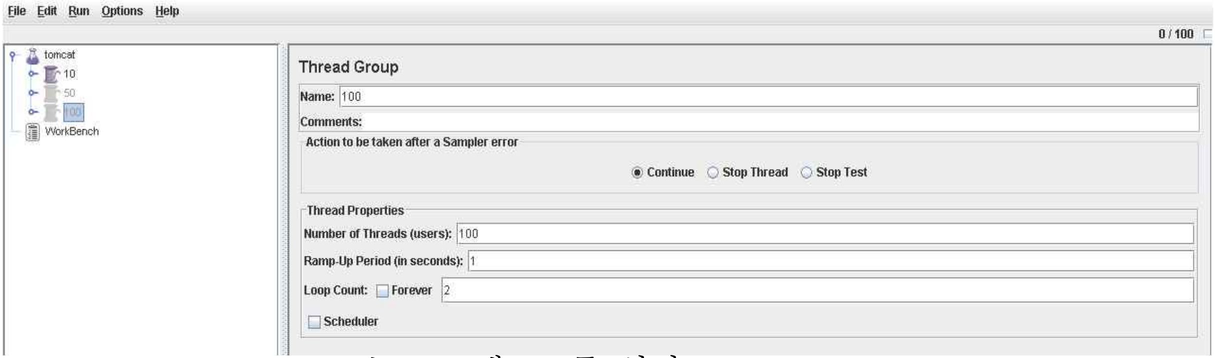
[그림3-1. 쓰레드그룹 설정 - 10 User]

● 50 User가 동시 2회 접속



[그림3-2. 쓰레드그룹 설정 - 50 User]

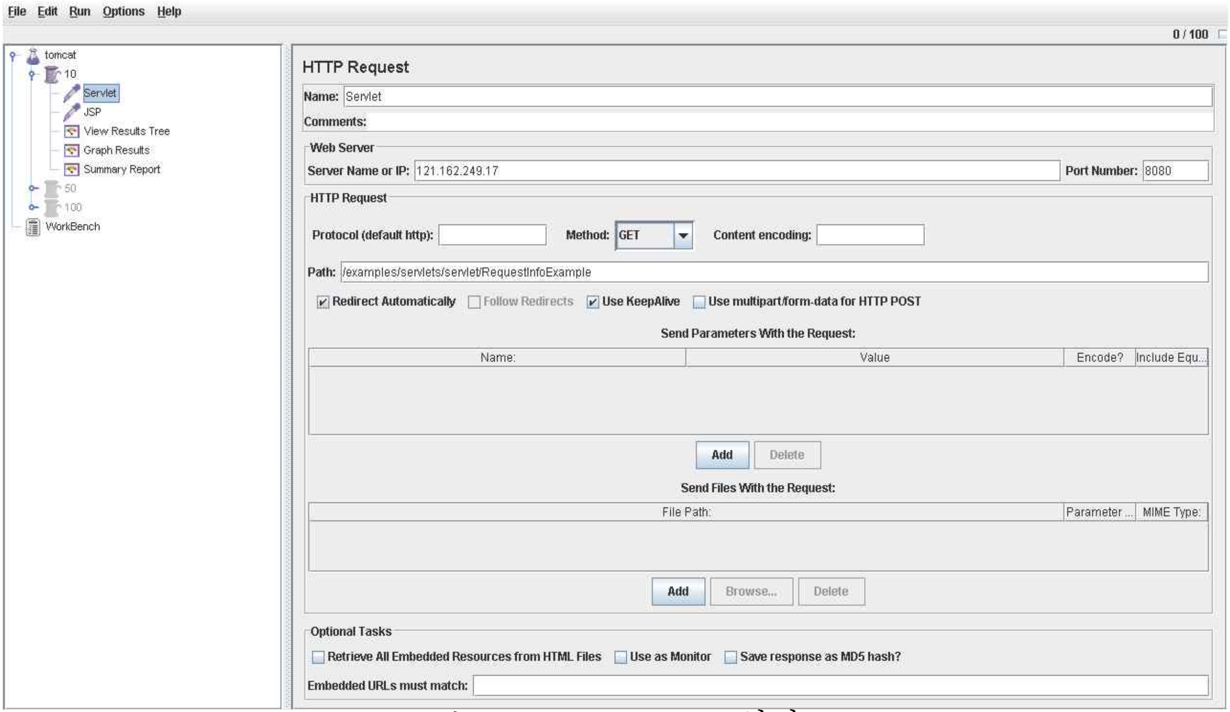
● 100 User가 동시 2회 접속



[그림3-3. 쓰레드그룹 설정 - 100 User]

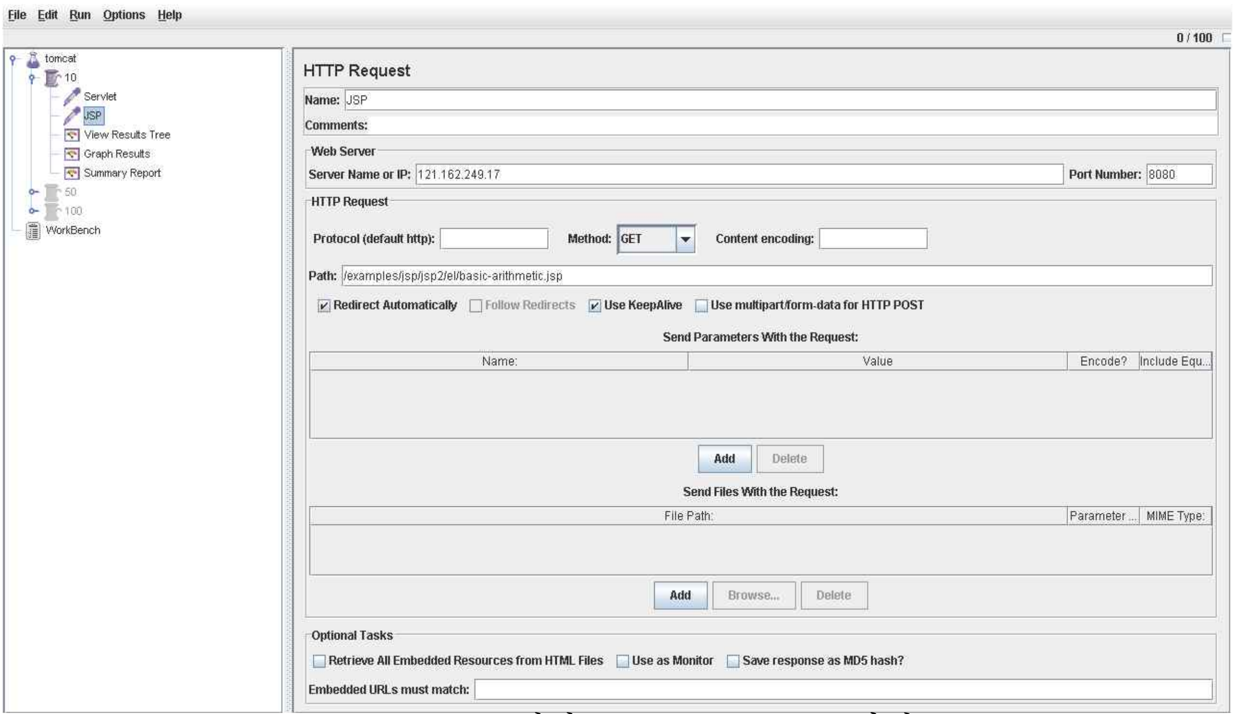
(3) Http Request를 추가 (각 Thread Group에 공통)

● 서블릿 구성인 Request Info 화면



[그림3-4. Http Request 설정]

● JSP 구성인 Basic Arithmetic 화면



[그림3-5. JSP 구성인 Basic Arithmetic 화면]

(4) Listener 추가 (각 Thread Group에 공통)
- View Results Tree
- Graph Results
- Summary Report

다. 테스트 실시

(1) JBoss의 기동
- apache-tomcat-6.0.20/bin 디렉터리에서
  > ./startup.sh 실행

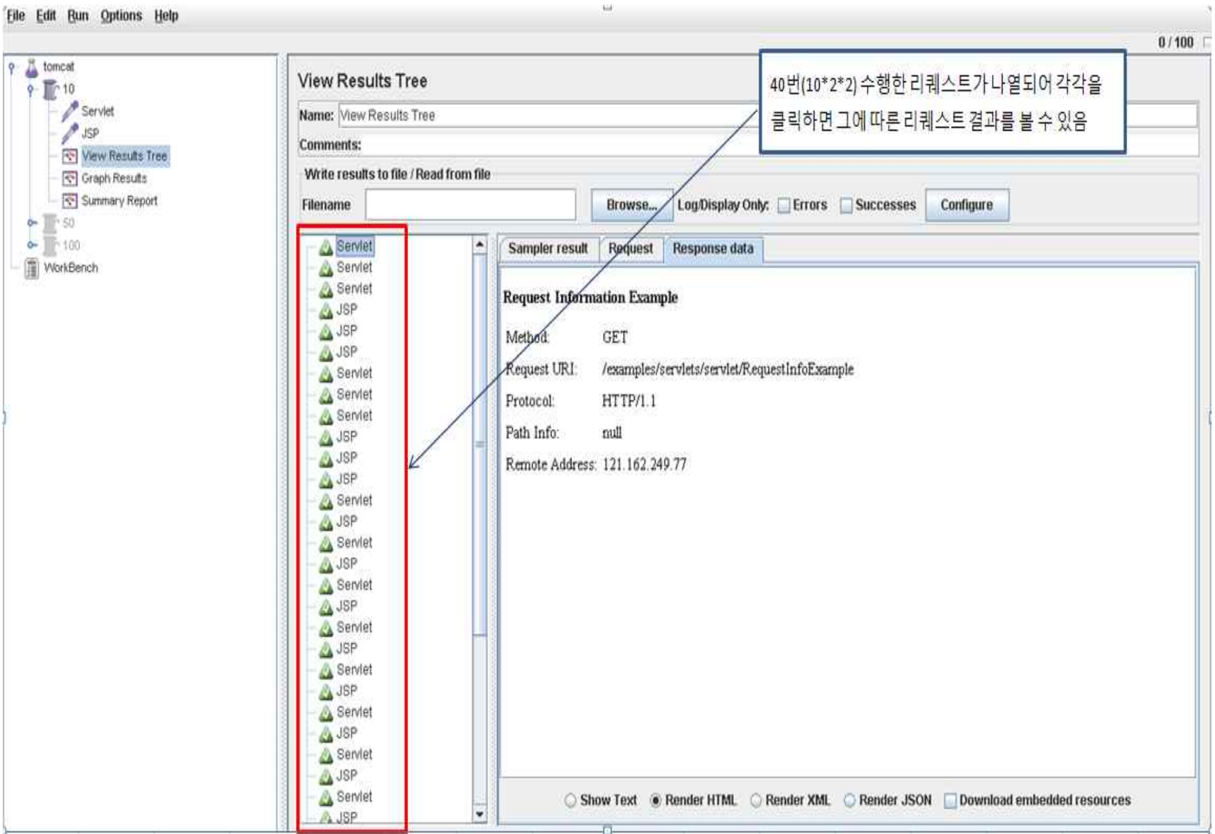
(2) Jmeter의 Run > Start

4. 테스트 결과

테스트 결과는
① View Results Tree - 서블릿 구성인 Request Info 화면
② View Results Tree - JSP 구성인 Basic Arithmetic 화면
③ Graph Results
④ Summary Report
의 Listener로 확인한다.

가. 10 User가 동시 2회 접속

① View Results Tree - 서블릿 구성인 Request Info 화면

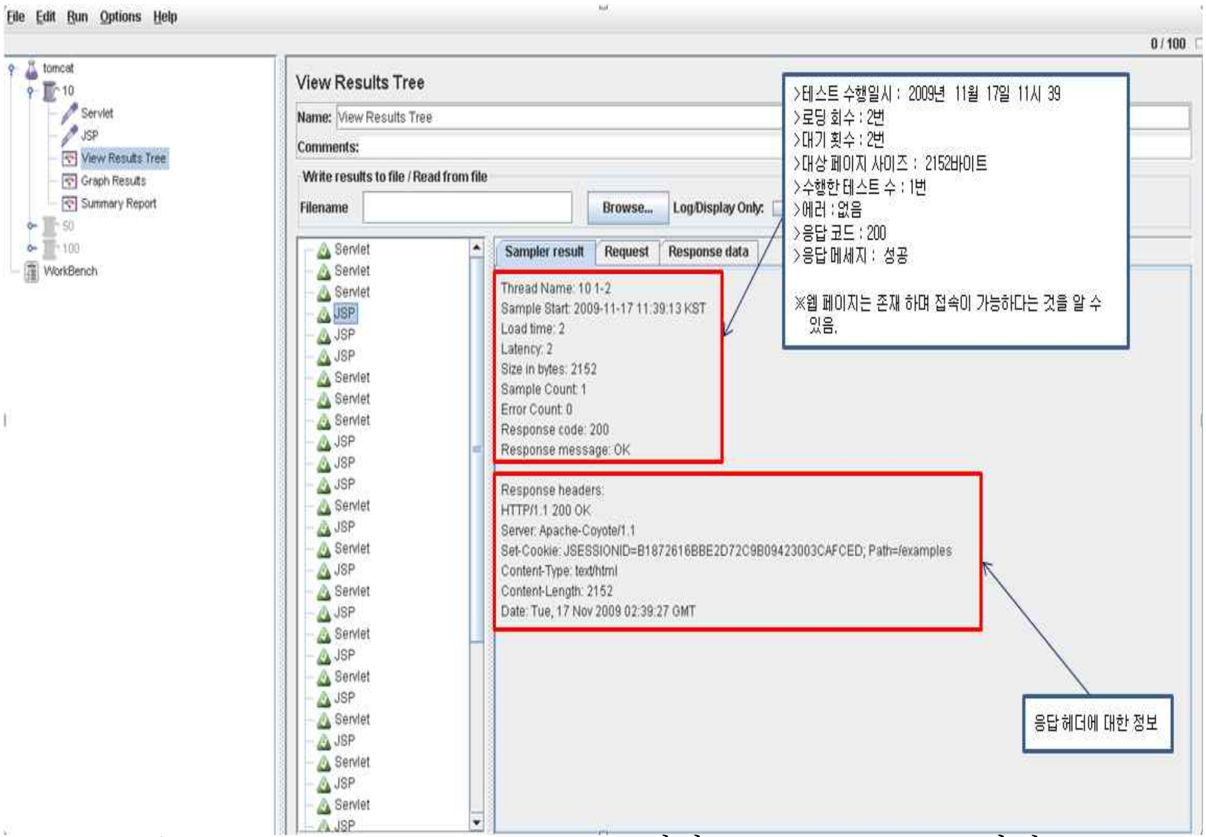


[그림4-1. View Results Tree - 서블릿 구성인 Request Info 화면]

서블릿으로 구성된 화면이다.

총 40번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

② View Results Tree - JSP 구성인 Basic Arithmetic 화면

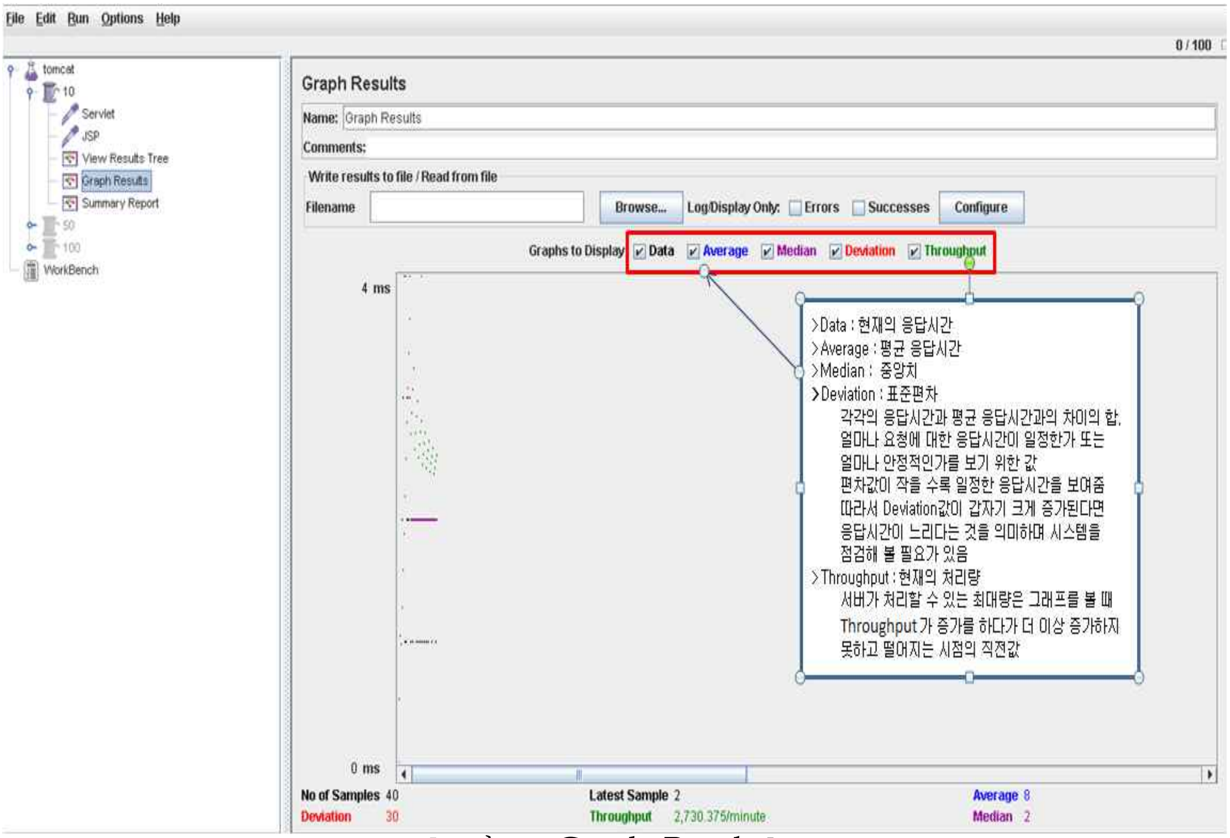


[그림4-2. View Results Tree - JSP 구성인 Basic Arithmetic 화면]

JSP로 구성한 화면을 정상적으로 표시하고 있음을 알 수 있다.

- 로딩 회수 : 2번
- 대기 횟수 : 2번
- 대상 페이지 사이즈 : 2152바이트
- 수행한 테스트 수 : 1번
- 에러 : 없음
- 응답 코드 : 200
- 응답 메시지 : 성공

③ Graph Results

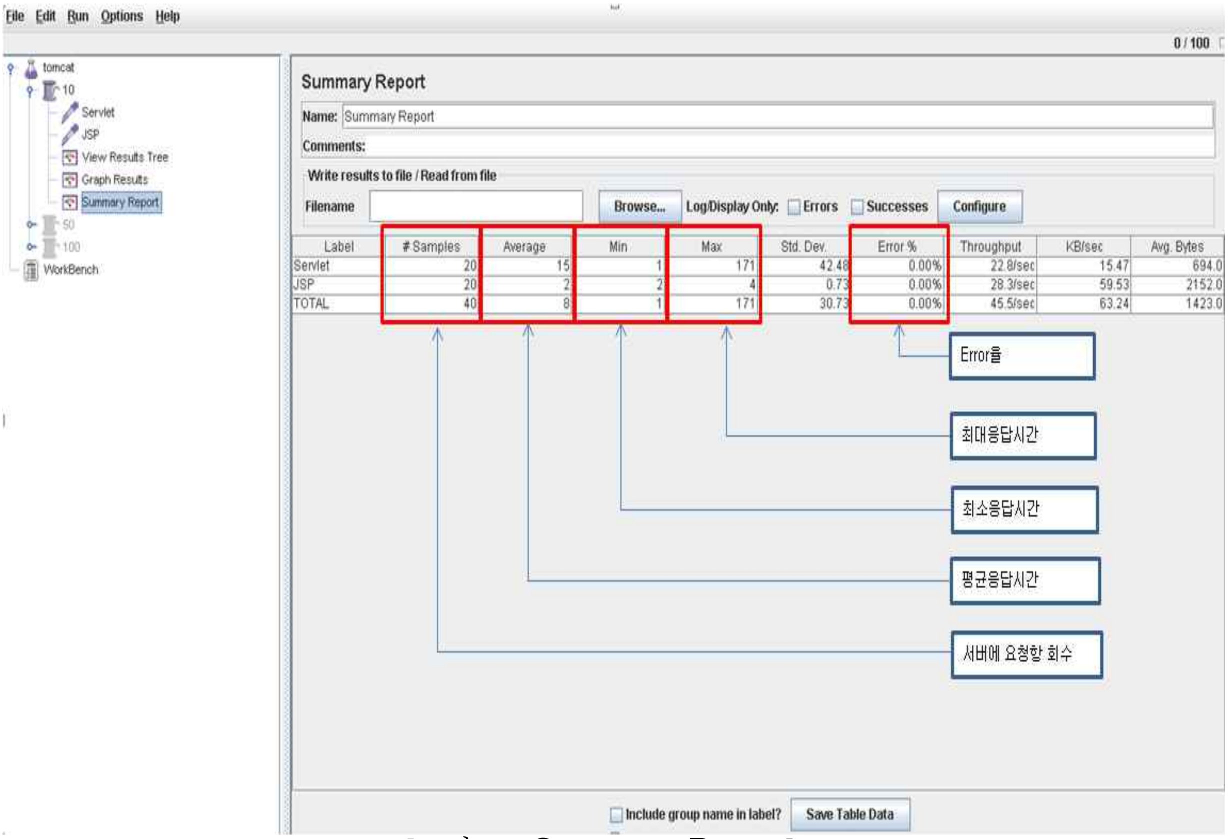


[그림4-3. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 40
- Average : 8
- Median : 2
- Deviation : 30
- Throughput : 2730.375/min

④ Summary Report



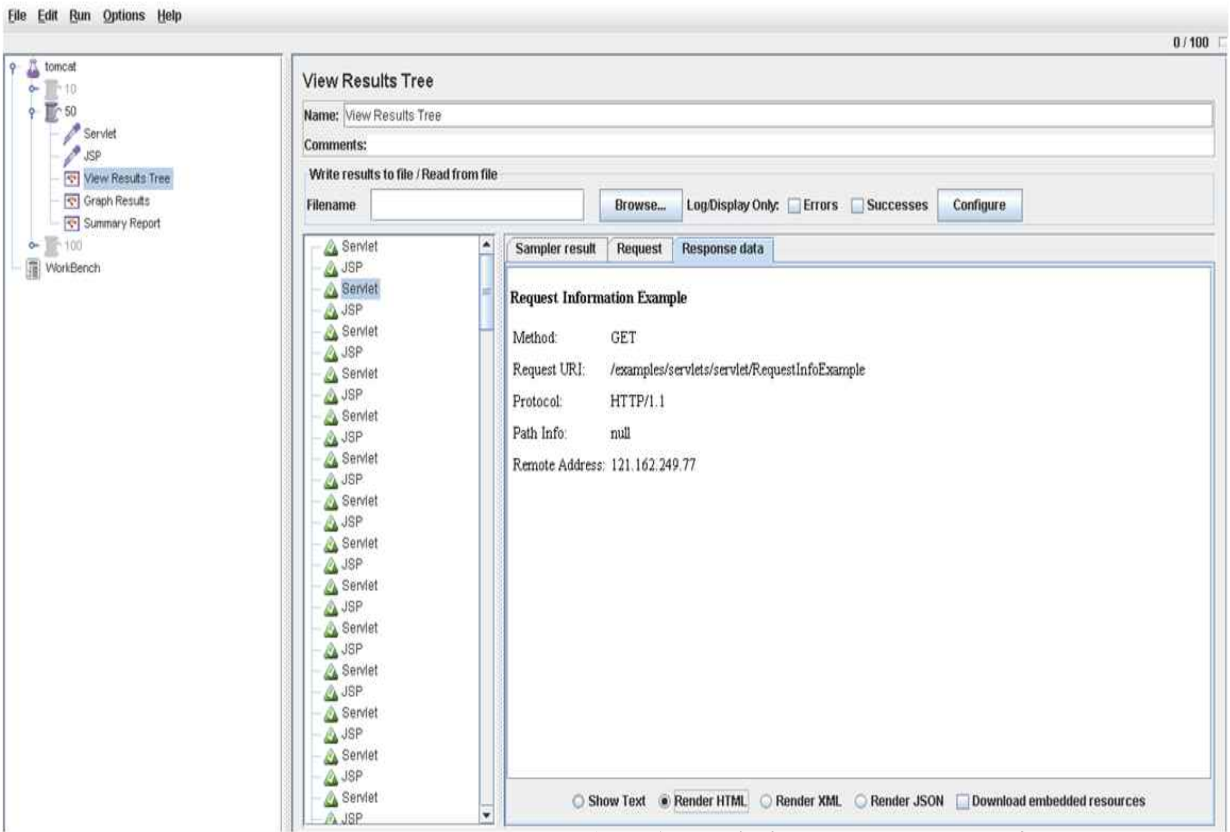
[그림4-4. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 40
- Average : 8
- Min : 1
- Max : 171
- Error : 0.00%
- Throughput : 45.5/sec

나. 50 User가 동시 2회 접속

① View Results Tree - 서블릿 구성인 Request Info 화면

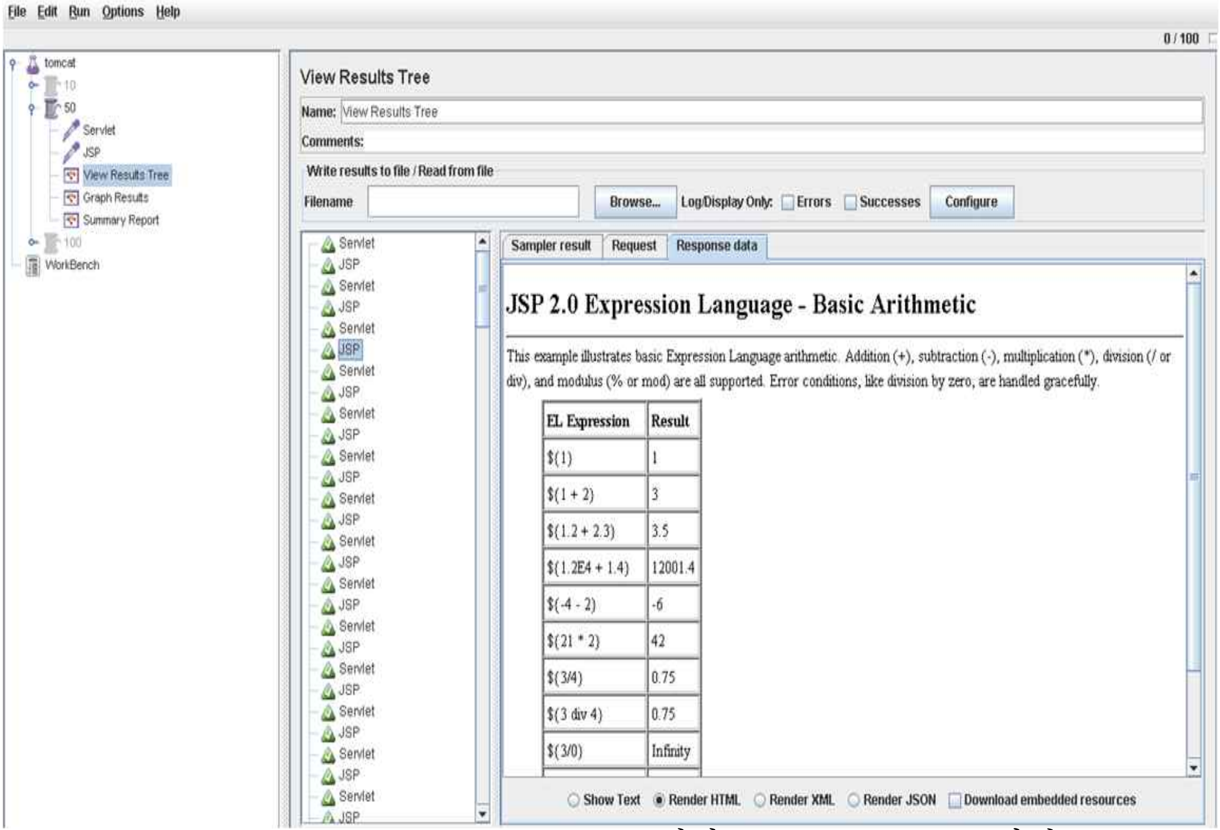


[그림4-5. View Results Tree - 서블릿 구성인 Request Info 화면]

서블릿으로 구성된 화면이다.

총 200번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

② View Results Tree - JSP 구성인 Basic Arithmetic 화면



[그림4-6. View Results Tree - JSP 구성인 Basic Arithmetic 화면]

JSP로 구성한 화면을 정상적으로 표시하고 있음을 알 수 있다.

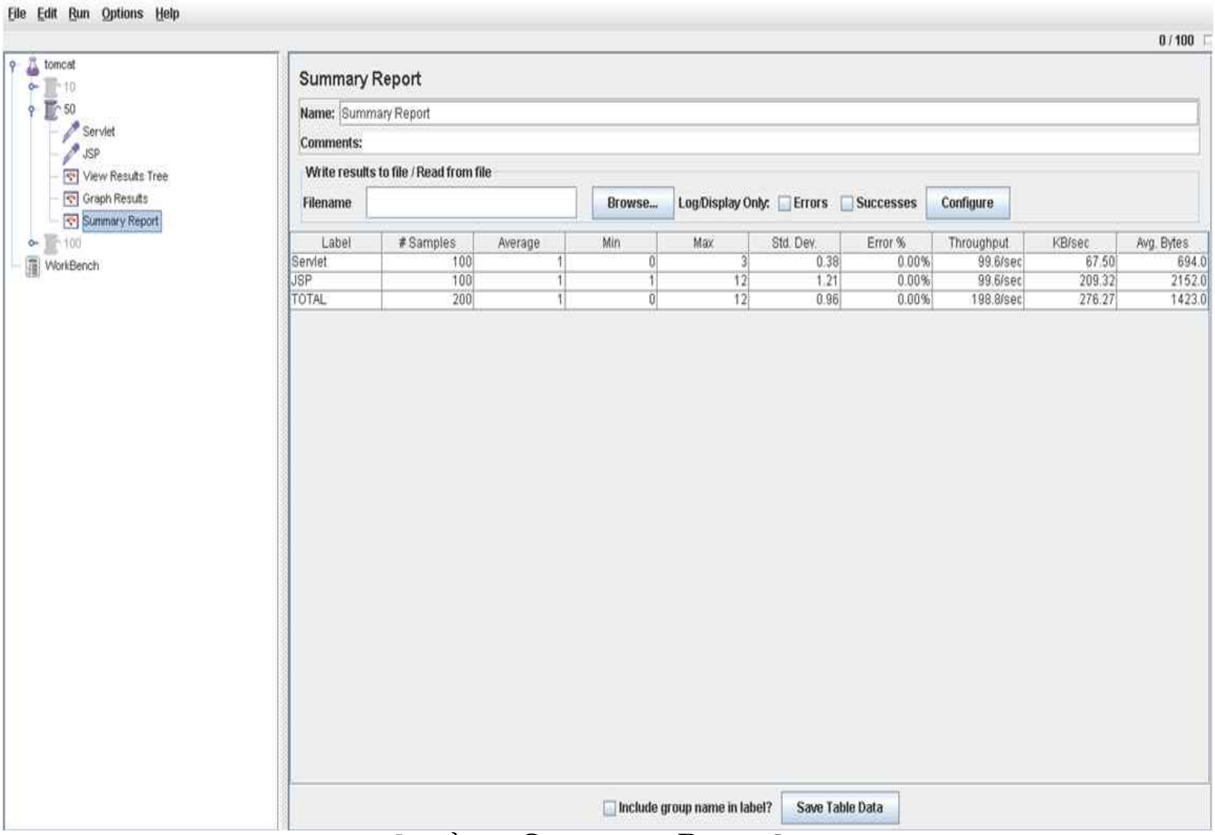
③ Graph Results

[그림4-7. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 200
- Average : 1
- Median : 1
- Deviation : 0
- Throughput : 11928.429/min

④ Summary Report



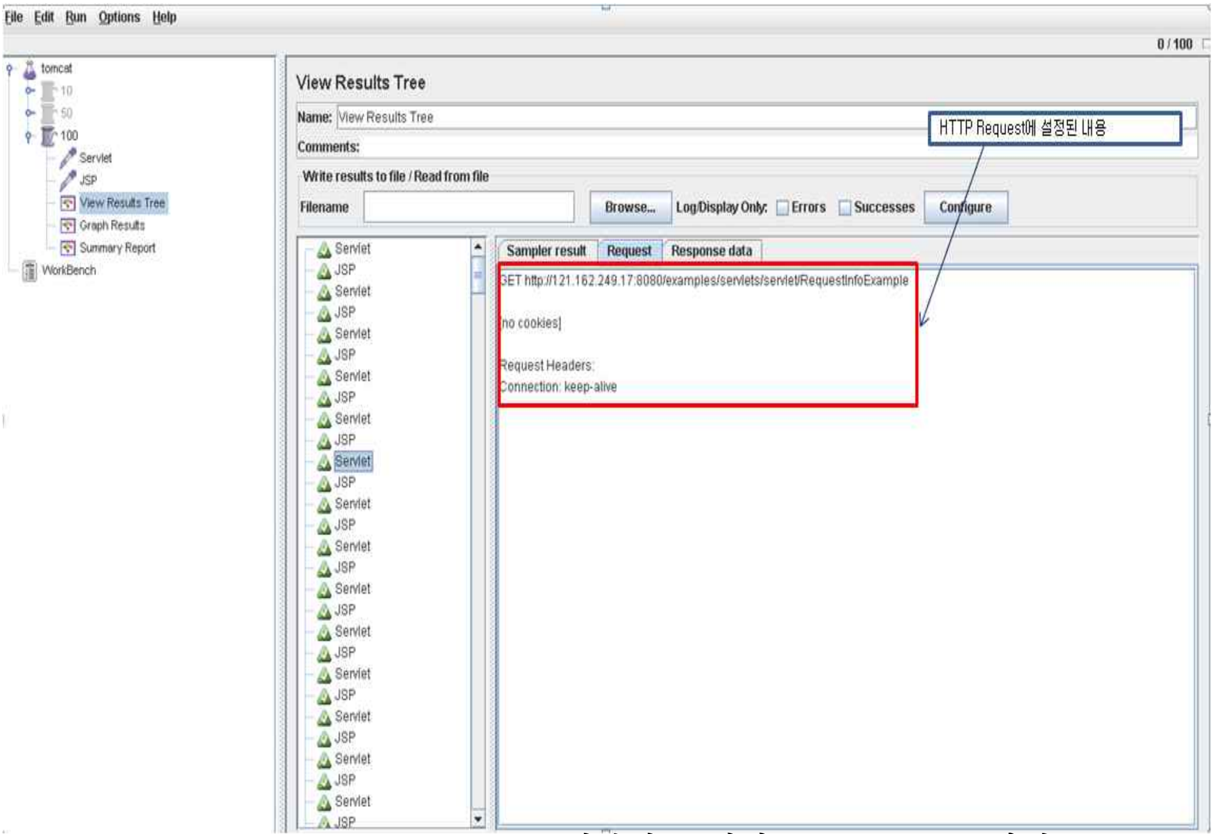
[그림4-8. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 200
- Average : 1
- Min : 0
- Max : 12
- Error : 0.00%
- Throughput : 198.8/sec

다. 100 User가 동시 2회 접속

① View Results Tree - 서블릿 구성인 Request Info 화면

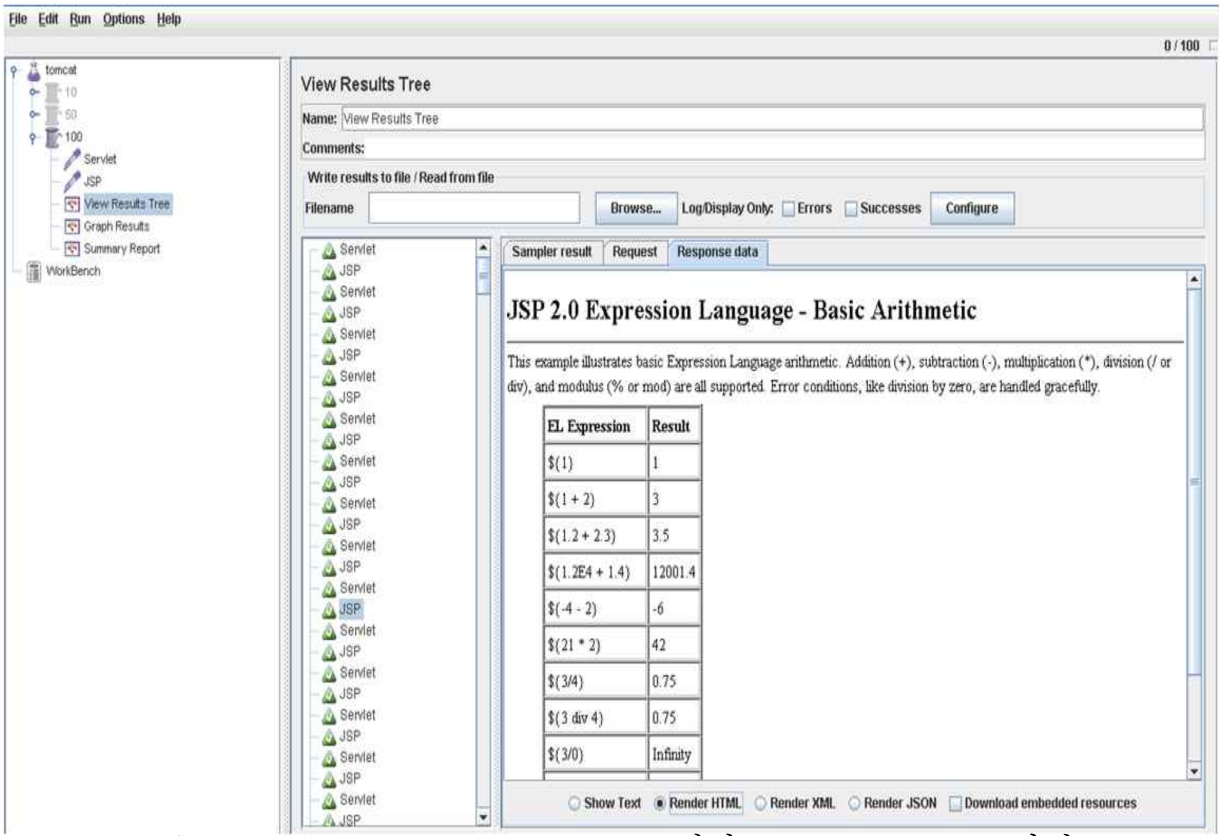


[그림4-9. View Results Tree - 서블릿 구성인 Request Info 화면]

서블릿으로 구성된 화면이다.

총 400번 동시 접속한 결과 응답이 정상인 것을 알 수 있다.

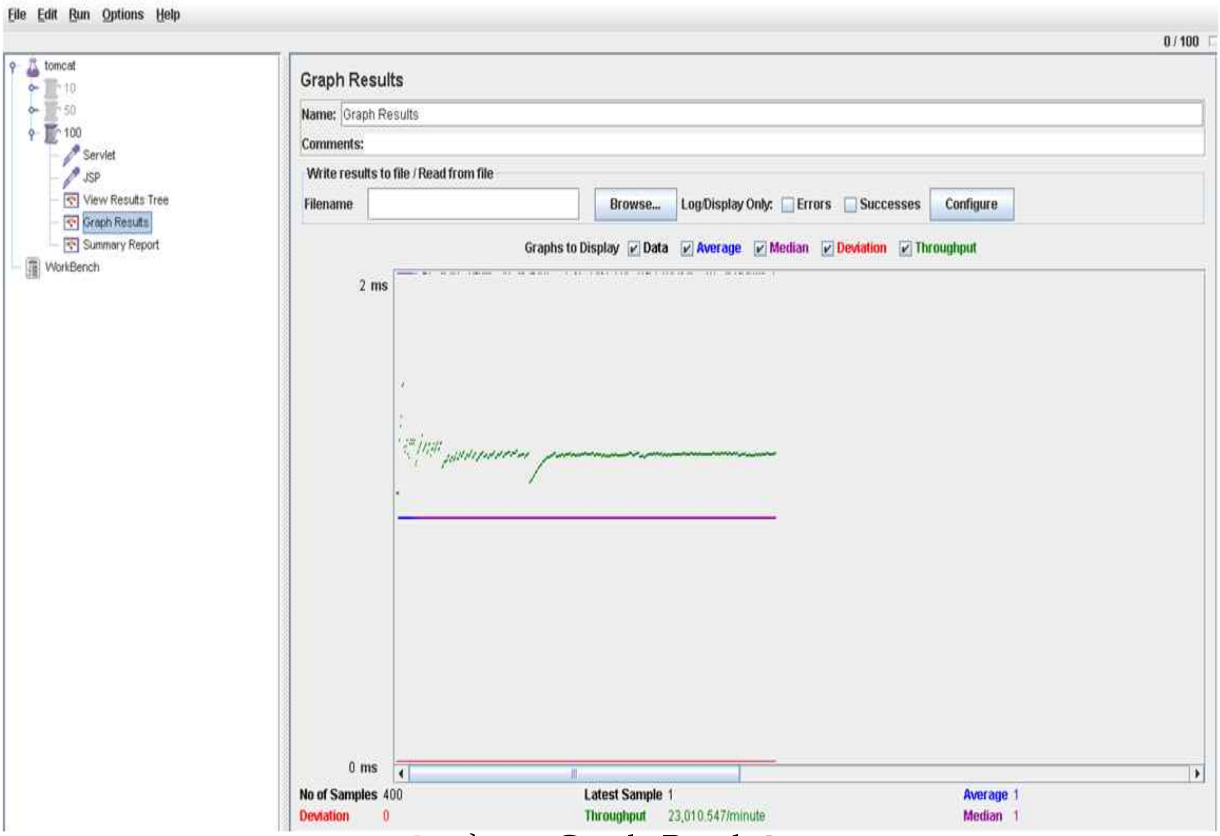
② View Results Tree - JSP 구성인 Basic Arithmetic 화면



[그림4-10. View Results Tree - JSP 구성인 Basic Arithmetic 화면]

JSP로 구성한 화면을 정상적으로 표시하고 있음을 알 수 있다.

③ Graph Results

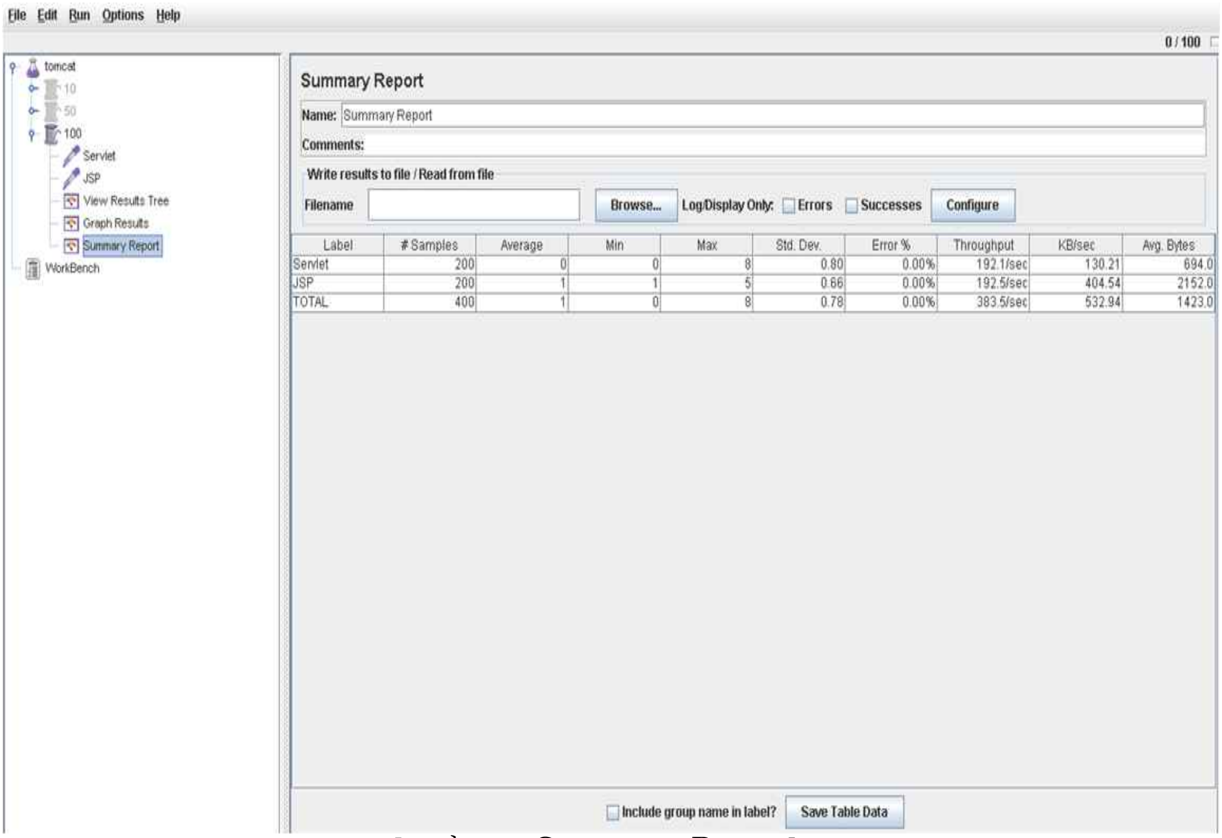


[그림4-11. Graph Results]

그래프를 분석하여 보면 응답시간, 표준편차, 처리량이 안정적인 것을 확인할 수 있다.

- Data : 400
- Average : 1
- Median : 1
- Deviation : 0
- Throughput : 23010.547/min

④ Summary Report



[그림4-12. Summary Report]

그래프의 데이터를 표 형식으로 확인 가능하다.

- Request : 400
- Average : 1
- Min : 0
- Max : 8
- Error : 0.00%
- Throughput : 383.5/sec

5. 종합

Summary Report

Request	Average	Min	Max	Error	Throughput	KB/sec
40	8	1	171	0.00%	45.5/sec	63.24
200	1	0	12	0.00%	198.8/sec	276.27
400	1	0	8	0.00%	383.5/sec	532.94

● 동시 접속 수 400 요청에도 Throughput의 안정과 에러 율 0 로서 시스템이 정상적으로 동작하는 것을 알 수 있다.

● 금번 테스트는 Tomcat 구성으로 하였으나 JVM, Web서비스, mod_jk 등의 튜닝으로 보다 나은 퍼포먼스의 향상을 기대할 수 있다.

별첨 1.

Linux Test Project
활용 가이드

2009년 12월 15일

정보통신산업진흥원

목　　　차

1. LPT Test Suite 소개

☐ LTP(Linux Test Project)는 리눅스의 신뢰성, 호환성, 안정성을 입증하여 오픈소스 커뮤니티에 Test Suite를 제공하는 오픈소스 프로젝트이다.

☐ LTP Test Suite는 리눅스 운영체제를 다양한 관점에서 테스트 할 수 있도록 개발된 자동, 반자동 Test Suite들의 집합체이다.

☐ LTP를 구성하는 프로그래밍언어는 ANSI-C(94%), Bash(5%), Perl(0.62%)이다.

☐ 본 문서에서는 LTP Test Suite의 다운로드, 컴파일링, 실행, 결과 출력에 대한 전반적인 내용을 설명한다.

2. LPT Test Suite의 다운로드

☐ LTP Test Suite 소스의 안정 버전은 기본적으로 매월 Sourceforge.net에 ltp-intermediate-yyyymmdd.tgz 형식으로 릴리즈 된다. Test Suite는 리눅스 커널의 안정성을 입증하고 개발SW를 릴리즈 하는데 사용되며, Source와 RPM 형식으로 제공된다.

☐ 패키지 ltp-intermediate-yyyymmdd.tgz는 LTP 웹사이트나 wget 커맨드를 사용하여 다운로드 할 수 있다.

☐ 통합 다운로드 사이트
- ○ http://sourceforge.net/projects/ltp/files/

☐ wget을 사용한 소스파일 다운로드 예
- ○ wget http://downloads.sourceforge.net/project/ltp/LTP%20Source/ltp- 20090831/ltp-intermediate-20090921.tgz?use_mirror=ufpr

☐ 다음의 익명 CVS를 통해서 LTP Test Suite 소스의 최신 CVS 스냅샷을 다운로드 할 수 있다.
- ○ cvs -d:pserver:anonymous@ltp.cvs.sourceforge.net:/cvsroot/ltp login
- ○ cvs -z3 -d:pserver:anonymous@ltp.cvs.sourceforge.net:/cvsroot/ltp co ltp

☐ 각 하위 디렉터리에 생성된 것들을 제외하고 CVS를 다운로드 하려면 아래의 명령어를 사용한다.

- ○ cvs -d:pserver:anonymous@ltp.cvs.sourceforge.net:/cvsroot/ltp login
- ○ cvs -z3 -d:pserver:anonymous@ltp.cvs.sourceforge.net:/cvsroot/ltp export -D "MONTH DAY, YEAR HH:MMam/pm" ltp

3. LPT Test Suite의 콘텐츠

☐ LTP 루트 디렉터리 /ltp-intermediate-yyyymmdd 에는 쉘 스크립트와 다양한 소스 디렉터리들이 포함되어 있다. 이 스크립트들은 테스트 케이스를 수행하는데 사용된다. 스크립트는 pan이라 불리는 테스트 드라이버를 사용하는 테스트 케이스를 실행한다.

가. 스크립트

1) IDcheck.sh

Test Suite는 시스템에 생성된 유저와 그룹을 요구한다. 이 스크립트는 유저와 그룹을 생성하며, LTP 루트 디렉터리에 위치한 최상위 Makefile을 불러온다. 또한, 루트 유저로 실행하여야 한다.

2) runltp

본 드라이버 스크립트는 LTP에서 중요한 테스트를 수행한다. 필요 시 테스트의 사용자가 지정한 테스트를 수행하기 위해 쉽게 수정할 수 있다. 기본적으로 이 스크립트는 다음의 테스트를 수행한다.

- ○ filesystem stress tests
- ○ disk I/O tests
- ○ memory management stress tests
- ○ ipc stress
- ○ scheduler tests
- ○ commands functional varification tests
- ○ system call functional varification tests

3) runalltest.sh

본 스크립트는 기본적으로 모든 테스트를 수행한다. 그러나 일부 수행하지 않는 테스트가 있다.

4) networktests.sh

네트워킹 연결 테스트를 수행한다. 이 스크립트를 수행하기 전에 두 개의 환경 변수인 RHOST(Remote Host Name)와 PASSWD(원격 HOST의 루트 패스워드)를 설정해야 한다.

5) diskio.sh

플로피 드라이브와 CD-ROM 드라이브의 연결 테스트를 수행한다. 이 테스트를 수행하기 위해서는 각 드라이브의 매체(플로피 디스크, CD-ROM 미디어)가 필요하다.

나. 디렉터리 설명

1) testcases

모든 테스트 케이스의 소스 코드가 포함되어 있다. 테스트 케이스는 서브시스템이나 서비스가 계획된 이후에 이름 지어진 디렉터리 하위에 위치한다. 예를 들어, 메모리 관리 테스트는 testcases/mem 등의 하위에 위치한다.

2) runtest

테스트 수행을 위한 pan에 의해 사용되어 지는 커맨드 파일들이 위치한다. 각 파일에는 테스트를 수행하기 위해 실행해야만 하는 인수인 테스트 이름과 커맨드의 이름이 포함되어 있다. 예를 들어, syscalls 파일은 아래 예제1 과 같은 입력들을 포함한다.

예제1. Command File의 내부 구조

```
# Name of test          Command
access01                access01
alarm04                 cp -f $LTPROOT/test-cases/bin/sig_rev $TMP;alarm04
chdir01A                symlink01 -T chdir01; chdir01
```

3) lib

LTP-harness API's (libltp.a)를 위한 소스코드가 위치한다. libltp.a 라이브러리는 로그 테스트 결과와 상태를 위해 이 API 함수들이 사용하는 모든 테스트 케이스와의 링크를 제공한다.

4) include

LTP API-harness 함수를 사용하는 모든 테스트 케이스에 포함된 HEADER 파일이 위치한다.

5) testcases/bin

컴파일링 되어 설치된 실행 가능한 테스트 케이스가 위치한다.

4. LPT Test Suite의 컴파일링

□ Test Suite를 컴파일 하기 위해서는 LTP Test Suite의 루트 디렉터리인 /ltp-intermediate-yyyymmdd로 디렉터리를 이동하고 configure를 입력한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# ./configure
```

□ 컴파일을 진행하기 위해서는 gcc와 lex 패키지가 필요하다. 해당 패키지가 설치되어 있지 않다면 다음을 수행하여 설치를 진행한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# yum install gcc flex
```

□ 컴파일이 완료되었으면 이제 Test Suite를 설치하기 위해 make와 make install을 순서대로 실행한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# make
[root@localhost ltp-intermediate-yyyymmdd]# make install
```

□ 설치는 루트 유저로 수행해야한다. 이 작업의 수행은 /ltp-intermediate-yyyymmdd/test-cases/bin 디렉터리에 해당 파일을 실행시키는 Symbolic 링크를 생성한다.

□ 테스트 스크립트(예:runltp)는 커맨드 파일 안에 열거된 테스트를 기반으로 이 디렉터리에 제공된 바이너리를 실행한다.

5. LPT Test Suite의 실행

□ 일반적으로 LTP Test Suite에는 테스트 케이스를 실행하기 위해 사용하는 3개의 스크립트(runltp, networktests.sh, diskio.sh)가 있다. 가장 대표적으로 사용하는 스크립트는 runltp 이며, [표1]과 같은 옵션을 제공한다. 만약 옵션 없이 실행한다면 아래처럼 기본적인 테스트만을 수행한다.

○ 기본 테스트 설정의 실행
○ 화면에 테스트에 대한 자세한 정보를 출력
○ 로그파일이나 테스트 결과를 생성하지 않음

[표1. LTP 테스트 슈트 옵션]

옵션	기능
-a EMAIL_TO	입력한 E-mail 주소로 모든 리포트를 전송
-c NUM_PROCS	부가적인 백그라운드 CPU 부하 아래에서 LTP를 실행 [NUM_PROCS = no. of processes creating the CPU Load by spinning over sqrt() (Defaults to 1 when value)]
-C FAILCMDFILE	모든 실패한 테스트 케이스를 포함한 커맨드 파일을 지정
-d TMPDIR	임시파일이 생성될 디렉터리 지정
-D NUM_PROCS,NUM_FILES,NUM_BYTES,CLEAN_FLAG	세컨드 스토리지의 부가적인 백그라운드 부하 아래에서 LTP를 실행 (구분자:‘,’) [NUM_PROCS = no. of processes creating Storage Load by spinning over write()] [NUM_FILES = Write() to these many files (Defaults to 1 when value 0 or undefined)] [NUM_BYTES = write these many bytes (defaults to 1GB, when value 0 or undefined)] [CLEAN_FLAG = unlink file to which random data written, when value 1]
-e	현재 LTP release의 날짜를 출력
-f CMDFILES	유저가 선언한 테스트 케이스의 리스트를 실행(구분자:‘,’)
-g HTMLFILE	추가된 HTML 출력 포맷을 생성
-h	도움말 출력(가능한 모든 옵션을 출력)

옵션	기능
-i NUM_PROCS	IO Bus위의 부가적인 백그라운드 부하 아래에서 LTP를 실행 [NUM_PROCS = no. of processes creating IO Bus Load by spinning over sync()]
-l LOGFILE	테스트 결과를 로그파일로 생성
-m NUM_PROCS,CHUNKS,BYTES,HANGUP_FLAG	주 메모리 위의 부가적인 백그라운드 부하 아래에서 LTP를 실행(구분자:‘,’) [NUM_PROCS = no. of processes creating main Memory Load by spinning over malloc()] [CHUNKS = malloc these many chunks (default is 1 when value 0 or undefined)] [BYTES = malloc CHUNKS of BYTES bytes (default is 256MB when value 0 or undefined)] [HANGUP_FLAG = hang in a sleep loop after memory allocated, when value 1]
-N	모든 네트워크 테스트를 수행
-n	백그라운드 네트워크 트래픽과 함께 LTP를 수행
-o OUTPUTFILE	파일로 출력
-p	ASCII 포맷으로 로그파일을 생성
-q	화면에 최소한의 정보만 출력
-r LTPROOT	LTP 루트 디렉터리의 경로를 지정
-s PATTERN	PATTERN과 매치되는 테스트 케이스만을 수행
-t DURATION	시간설정 ex) -t 60s = 60 초 -t 45m = 45 분 -t 24h = 24 시간 -t 2d = 2 일
-T REPETITION	Test Suite를 반복해서 수행
-v	테스트 케이스의 유저 리스트를 얻기 위해 wget을 사용
-w CMDFILEADDR	이 Test Suite의 다양한 인스턴스를 수행

가. 테스트의 기본 설정 실행

□ LTP의 루트 디렉토리로 이동 후 runltp 스크립트를 실행한다.

```
[root@localhost /]# cd ltp-intermediate-yyyymmdd
[root@localhost ltp-intermediate-yyyymmdd]# ./runltp
```

□ 원하는 옵션이 있을 경우 상단의 옵션 리스트를 참고하여 옵션을 추가한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# ./runltp -p -l logfile_name.log
```

예제2. 결과 로그 파일의 내용

```
Test Start Time: Mon Sep 28 13:38:45 2009
-----------------------------------------
Testcase                    Result     Exit Value
---------                   -------    ----------
abort01                     PASS       0
accept01                    PASS       0
accept4_01                  PASS       0
access01                    PASS       0
access02                    PASS       0
access03                    PASS       0

... 중략 ...

mkdir_tests01               PASS       0
mv_tests01                  PASS       0
size01                      PASS       0
smt_smp_enabled             PASS       0
smt_smp_affinity            PASS       0
ht_interrupt                PASS       0
-----------------------------------------------
Total Tests: 1225
Total Failures: 75
Kernel Version: 2.6.18-128.GX2
Machine Architecture: x86_64
Hostname: skcc_oss
```

나. 테스트의 유저 설정 실행

□ 테스트 케이스의 설정을 변경하여 테스트를 수행하고자 할 경우에는 다음의 내용을 수행한다.

1) 커맨드 파일을 생성한다.
커맨드 파일은 인수를 포함한 테스트 케이스와 테스트 케이스 Tag를 포함해야 한다.

예제3. Command File의 내부 구조

```
#Tag        Test case
#-------------------------------------
mtest01      mtest01 -p 10
mmstress    mmstress -x 100
fork01       fork01
chdir01      symlink01 -T chdir01
#--------------------------------------
```

2) LTP 루트 디렉터리인 /ltp-intermediate-yyyymmdd 디렉터리로 이동한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# yum install gcc flex
```

3) 이 테스트를 수행하기 위한 커맨드 파일 옵션인 -f를 사용하여 runltp를 실행한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# ./runltp -p -l logfile_name.log -f
my_command_file
```

6. LPT Test Suite의 실행결과 확인

가. 화면 출력

□ 테스트 케이스, 테스트 드라이버, 스크립트는 다양한 정보 메시지를 화면에 출력한다.
□ 테스트 케이스는 정보를 프린트하기 위해 USCTEST 함수를 사용한다.
□ 테스트 드라이버 pan은 또 다른 테스트로부터 별도의 출력인 tags를 프린트 한다.
□ 옵션에 따라, 스크립트를 실행하는 동안 이 메시지들의 내용을 조절할 수 있다.

나. 로그파일 출력

□ 테스트 드라이버 pan은 테스트의 성공 또는 실패를 결정하기위해 테스트 케이스의 종료 값을 사용한다. 만약 테스트 케이스가 zero pan 값으로 종료될 경우 PASS로 기록하고 zero pan이 아닐 경우 FAIL로 기록한다.

□ 테스트 드라이버 pan은 로그 파일을 생성한다. 테스트 케이스의 실행으로부터 특정 결과인 이 로그파일은 로그 결과를 위하여 runalltest.sh 스크립트에 -l 로그파일_이름 옵션을 사용할 수 있다. 로그파일은 기본적으로 results 디렉터리에 생성되며, 결과 로그파일은 2개의 포맷으로 제공된다.

1) ASCII 포맷 : -p 옵션

예제4. ASCII 포맷의 결과 화면

```
Test Start Time: Mon Sep 28 11:06:37 2009
-----------------------------------------
Testcase                  Result     Exit Value
--------                  ------     ----------
mtest01                   FAIL        1
mmstress                   PASS        0
-----------------------------------------------
Total Tests: 2
Total Failures: 1
Kernel Version: 2.4.19
Machine Architecture: i686
Hostname: lazy
```

2) SCRIPTS 포맷 : 기본 값

기본 포맷 방식이며 테스트 결과 분석 보고서를 생성하는 스크립트에 의해 사용될 수 있다.

예제5. SCRIPTS 포맷의 결과 화면

```
tag=mtest01 stime=1043947511 dur=0 exit=exited stat=1 core=no cu=0 cs=11
tag=mmstress stime=1043947511 dur=41 exit=exited stat=0 core=no cu=671 cs=3383
tag=fork01 stime=1043947552 dur=0 exit=exited stat=0 core=no cu=0 cs=1
tag=chdir01A stime=1043947552 dur=0 exit=exited stat=0 core=no cu=0 cs=0
```