
분산 파일 시스템 분야 Stack 통합 테스트 결과보고서 [XtremFS]

2012. 09.

목 차

I. Stack 통합 테스트 개요	3
1. 목적	3
II. 분산 파일 시스템 소개	4
1. 분산 파일 시스템	4
2. 분산 파일 시스템 분야 주요 공개SW	6
III. 테스트 대상 소개	7
1. XtreamFS 소개	7
IV. Stack 통합 테스트	9
1. 테스트 환경	9
2. 주요 테스트 방법	10
3. 기능 테스트 수행 결과	11
4. 성능 테스트 수행 결과	12
V. 종합	27
※ 참고자료	28
[별첨 1] XtreamFS 테스트 케이스	

I. Stack 통합 테스트 개요

공개SW Stack 통합테스트는 여러 공개SW들의 조합으로 시스템 Stack을 구성한 후 Stack을 구성하는 공개SW의 상호운용성에 중점을 두고 기능 및 성능테스트 시나리오를 개발하여 테스트를 진행한다.

본 통합테스트를 통해 안정된 Stack 정보를 제공하여 민간 및 공공 정보시스템 도입 시 활용될 수 있도록 한다.

1. 목적

☐ 공개SW Stack 통합 테스트 수행 목적

- 공개SW로 구성된 Stack이 유기적으로 잘 동작함을 확인
- 다양한 Stack 구성에 기반을 둔 테스트를 통해 안정된 Stack 조합 규명
- 공개SW 시스템 도입을 위한 Stack 참조모델의 신뢰성 정보로 활용
- 공개SW의 신뢰성과 범용성에 대한 사용자 인식 제고

II. 분산 파일 시스템 소개

1. 분산 파일 시스템(Distributed File System)

분산 파일 시스템은 컴퓨터 네트워크를 공유하는 여러 호스트 컴퓨터에게 동일하게 보이는 파일 접근 공간을 제공해 주는 시스템으로, 이는 여러 사용자가 공통된 파일 및 스토리지 자원을 공유할 수 있음을 의미한다.

클라이언트 노드는 기본 블록 스토리지에 직접 액세스 할 수 없지만, 네트워크 프로토콜을 사용하여 상호작용 할 수 있으며, 프로토콜의 설계 방법에 따라 서버, 클라이언트, 기능에 따른 파일 시스템에 대한 액세스 권한을 설정할 수 있다.

기존 시스템은 디스크 액세스 시간과 CPU 처리량으로 구성되어 있지만, 분산 파일 시스템은 원격 액세스 구조로 인해 추가 오버 헤드가 있다. 이는 서버에 요청을 전달하는 시간과 클라이언트에서 응답을 제공하는 시간 간의 통신 프로토콜 소프트웨어를 실행하는 CPU 오버헤드이다.

(출처 : WIKIPEDIA)

□ 고려사항

분산 파일 시스템은 단일 제어 체계를 가진 하나의 컴퓨터 시스템이 아니라, 서로 독립적이고 분리된 시스템 사이에 파일 서비스를 제공함으로써 다음을 고려하여야 한다.

○ 동시제어

동시에 서로 다른 컴퓨터에서 파일 작업이 이루어지기 때문에 작업간의 순서를 관리하여야 한다.

○ 복구

서버의 운영이나 네트워크에서 문제가 발생하여 서버 간의 통신이 불가능하게 되었을 경우의 대처 방안이 고려되어야 한다.

○ 확장성

분산 파일 시스템을 구성하는 서버와 클라이언트의 수가, 파일 서비스의 성능에 어떤 영향을 미치는지 고려되어야 한다.

○ 보안 및 액세스 제어

분산 파일 시스템은 다수의 사용자가 다수의 컴퓨터 시스템 사이에서 파일을 서비스 받기 때문에 보안과 접근 제어 방식에 대한 정책이 고려되어야 한다.

2. 분산 파일 시스템 분야 주요 공개SW

[표 II-1. 주요 공개SW]

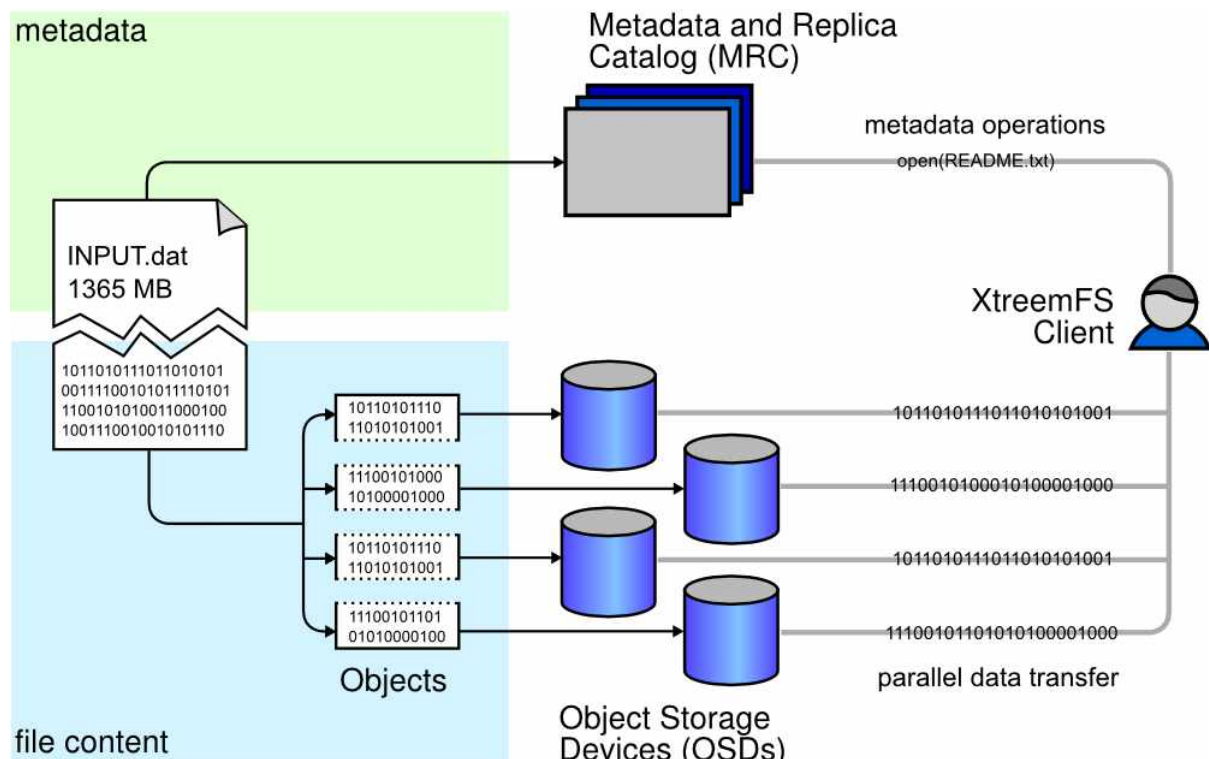
제품명	Stack 환경		홈페이지	비고
XtreemFS	OS	Linux	http://www.xtreemfs.org/	New BSD
	JAVA	JRE(JDK) 1.6.0		
Gluster	OS	Linux	http://www.gluster.org	GPL v3
	JAVA	JDK 1.6.0		
Hadoop	OS	Cross-platform	http://hadoop.apache.org	Apache License 2.0
	JAVA	JRE 1.6.0		
Sector	OS	Linux	http://sector.sourceforge.net	Apache License 2.0

III. 테스트 대상 소개

1. XtreamFS 소개

XtreamFS는 광역 네트워크에 대한 객체 기반의 분산 파일 시스템으로, 대기 시간의 성능 개선과 데이터의 결함 회피를 위해 파일 시스템의 개체복제를 이용한다. XtreamFS는 공용 네트워크를 통해 SSL¹⁾ 및 X.509²⁾ 인증서를 사용할 수 있다.

XtreamFS는 고성능 컴퓨터 클러스터에서 사용하기 위한 것이 아니라, 다수의 작은 파일 저장 공간을 활용하여 안정성과 가용성을 확보하는 그리드 파일 시스템이다.



[그림 III-1. XtreamFS 구조]

1) Secure Socket Layer의 약자로 인터넷상에서 교환되고 있는 각종 데이터를 암호화 하는 통신 기술

2) PKI(Public Key Infrastructure)에서 사용하는 표준 인증서 형식

□ XtreamFS 구성

o DIR(Directory Service) Server

XtreamFS의 모든 서비스에 대한 레지스트리 서버

o MRC(Metadata and Replica catalog) Server

자신의 디렉토리 트리와 볼륨에 있는 파일에 대한 모든 메타 데이터와 볼륨을 저장하며, 사용자를 인증하고 파일에 대한 액세스를 승인하는 서버

o OSD(Object Stores Device) Server

파일 내용을 저장하고 복제를 구현하는 서버

□ XtreamFS Stable Packages 지원

[표 III-1. packages 지원 OS]

구분	OS
Linux	- openSUSE - Fedora - CentOS - SuSE Linux Enterprise - Redhat Enterprise Linux - Mandriva - Ubuntu - Debian - Gentoo
Windows(client only)	- Windows XP - Windows Vista - Windows 7
MAC(client only)	- Mac OS X

※ 추가적인 자세한 정보는 아래의 링크 정보 참조

- <http://www.xtreamfs.org/download.php>

IV. Stack 통합 테스트

1. 테스트 환경

☐ XtreamFS 환경

[표 IV-1. XtreamFS 환경]

모듈	Version
XtreamFS Server	1.3.1-4.2
XtreamFS backend	1.3.1-4.2
XtreamFS client	1.3.1-4.2
XtreamFS tools	1.3.1-4.2

☐ Stack 환경

[표 IV-2. Stack 환경]

구성	Host OS	Java(JRE)	FUSE ³⁾ (kernel)
DIR/MRC Server	CentOS 6.3(64bit)	1.7.0_05	2.8.3(2.6.32)
OSD Server_1	CentOS 6.3(64bit)	1.7.0_05	2.8.3(2.6.32)
OSD Server_2	CentOS 6.3(64bit)	1.7.0_05	2.8.3(2.6.32)

☐ HW 환경

[표 IV-3. HW 환경]

제조사	모델명	CPU	MEM	Disk	NIC
IBM	X3550M2	Intel Xeon(R)CPU 2.40GHz * 4	8GB	320GB	Gigabit 1Port

※ 동일 사양의 HW Stack 구성

3) Filesystem in Userspaced의 약자로 코드 작성자가 커널코드를 수정하지 않고 파일 시스템을 만들 수 있도록 리눅스에서 제공하는 커널모듈 및 라이브러리

2. 주요 테스트 방법

□ 시나리오 테스트

시나리오 테스트 기법은 단일 기능에 대한 결함 여부를 확인하는 것이 아니라, 서로 다른 컴포넌트 사이의 상호작용과 간섭으로 발생할 수 있는 결함을 발견하기 위한 기법이다.

본 테스트에서는 사용자 시나리오 테스트 기법을 적용하여 XtreamFS를 사용하는 사용자들이 사용할 수 있는 항목 중 Volumes, Policy, Services Management, Replication, Striping에 대한 사용자 시나리오를 도출하였다. 각각의 항목에서 도출한 세부 시나리오는 사용자가 일반적으로 수행할 수 있는 시나리오를 추출하여 테스트케이스로 작성하였다.

□ 상호 운용성 기반 테스트

상호 운용성은 서로 다른 기술로 이루어진 제품이나 서비스가 상호작용 상의 오류가 없는지 검증하는 기법으로, 본 테스트에서는 애플리케이션이 지원하는 Stack을 구성하여 애플리케이션과 Stack 환경 사이의 상호작용 상의 동작여부를 검증하였다.

3. 기능 테스트 수행 결과

기능 테스트 수행 관련 세부 시나리오는 별첨 「XtreemFS 테스트 케이스」 문서를 참고한다.

□ 테스트 시나리오 현황

[표 IV-4. 테스트 시나리오 현황]

기능	테스트 시나리오	테스트 케이스
Services Management	3	12
Volumes	5	15
Policy	4	16
Replication	1	8
Striping	1	7
합 계	14	58

□ 테스트 결과

기능 테스트 시나리오를 통한 테스트 수행 결과 Replication, Striping 등 시나리오 상의 모든 기능이 예상 결과와 동일하게 동작함을 확인하였다.

[표 IV-5. 테스트 결과]

분류		PASS	FAIL	N/A
기능	개수			
Services Management	12	12	0	0
Volumes	15	15	0	0
Policy	16	16	0	0
Replication	8	8	0	0
Striping	7	7	0	0

4. 성능 테스트 수행 결과

성능 테스트의 경우 하드웨어 사양뿐 아니라, OS 및 애플리케이션 환경 구성에 따라 성능 측정 결과가 상이하므로, 실제 운영 시스템 환경에 따라 테스트 결과가 다를 수 있다.

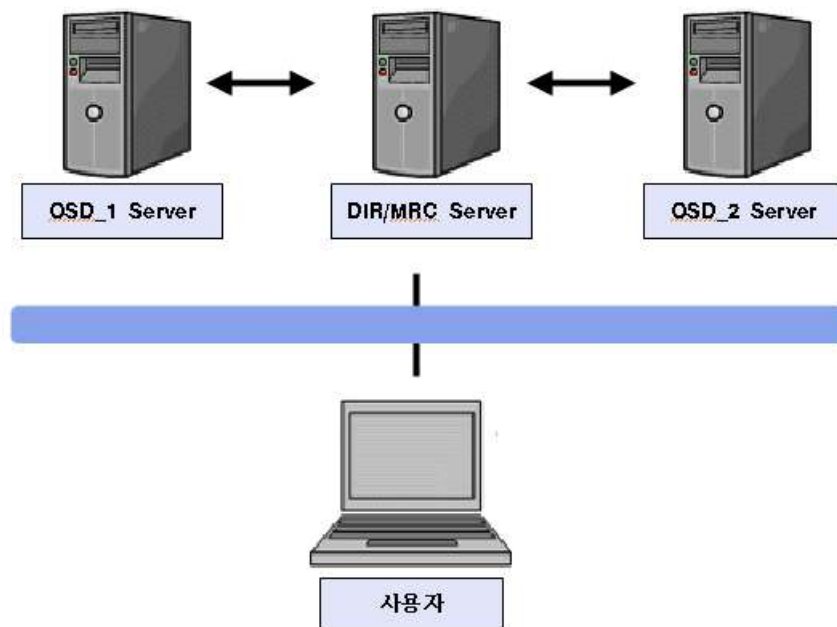
본 성능 테스트는 XtreamFS 시스템이 가동되는 상황에서 부하 시나리오를 재현하여, 각 Server의 자원 사용률을 측정하였다.

□ 테스트 시나리오

[표 IV-6. 테스트 시나리오]

시나리오 ID	시나리오
SC_DFS_X_1	Replication_128(파일이 128byte로 분할되어 복제)
SC_DFS_X_2	Replication_256(파일이 256byte로 분할되어 복제)
SC_DFS_X_3	Striping_128(파일이 128byte로 분산되어 저장)
SC_DFS_X_4	Striping_256(파일이 256byte로 분산되어 저장)

□ 서버 구성



[그림 IV-1. 성능 테스트 환경 정보]

□ 측정 항목

[표 IV-7. 측정 항목]

항목	내용
CPU 사용률	프로세스에서 CPU(Central Processing Unit)를 사용한 비율(%)
메모리 사용률	Physiscal 메모리를 사용량
네트워크 I/O	네트워크 입출력 작업의 양(K/B)

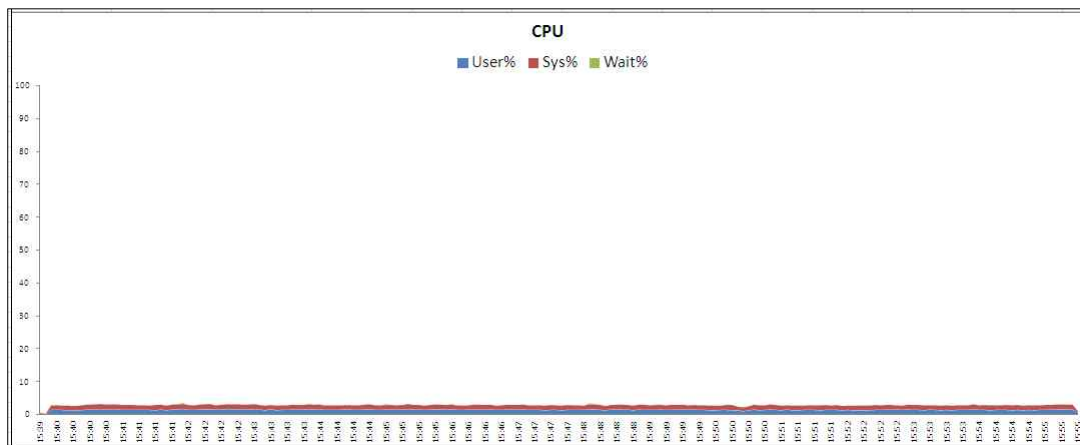
□ 테스트 결과

o Replication_128

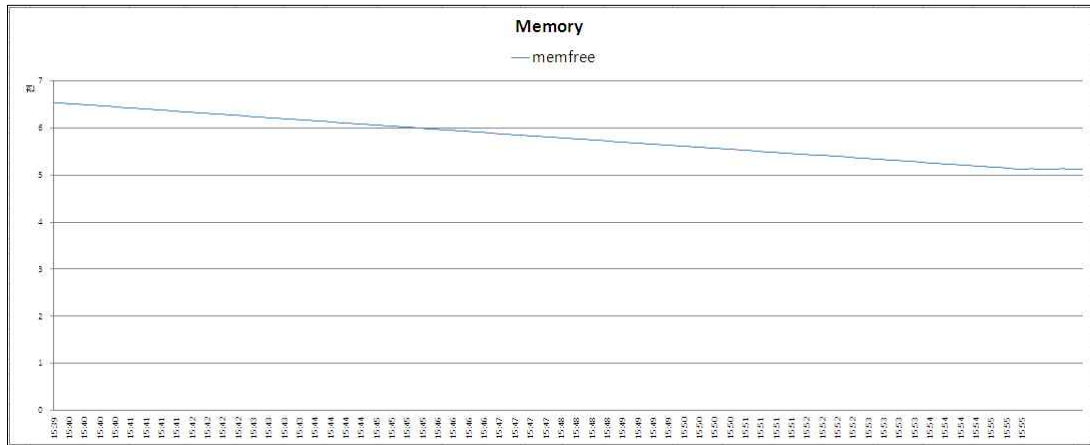
- 수행 정보

수행 조건	- Data를 128byte로 분할 - OSD_1 Server에 저장 - OSD_1 Data를 OSD_2 Server로 복제
-------	---

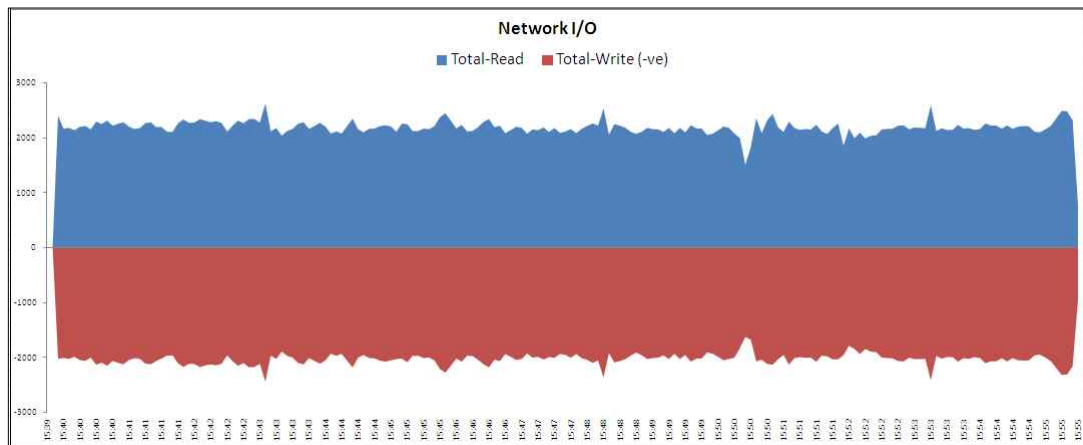
- DIR/MRC Server 측정 결과



[그림 IV-2. CPU 자원사용률]



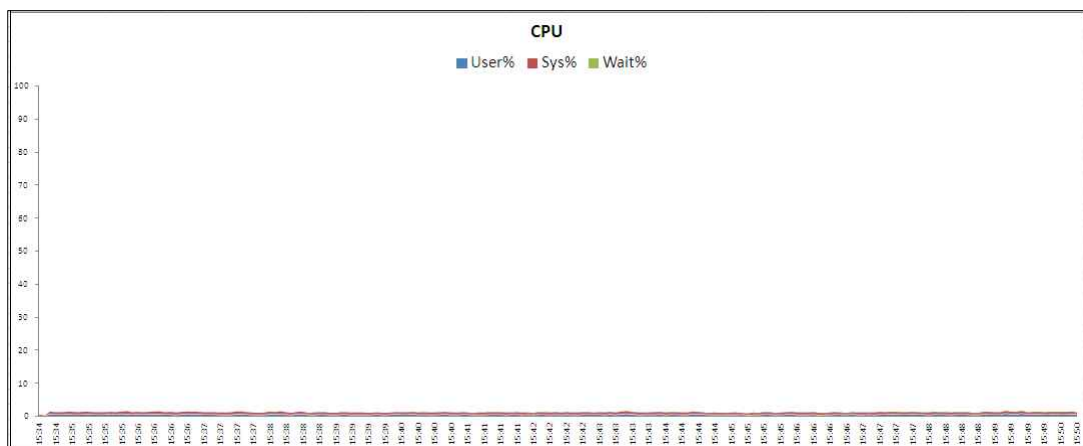
[그림 IV-3. Memory 자원사용률]



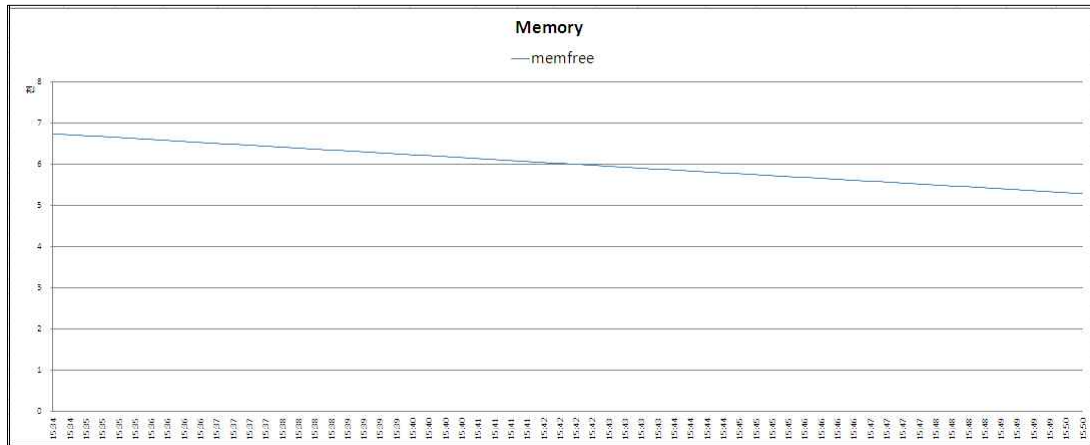
[그림 IV-4. Network 자원사용률]

(공통) DIR/MRC Server는 FileZilla로부터 1.37GB 파일을 직접 전송받기 때문에 Total-Read가 다소 높게 나타남

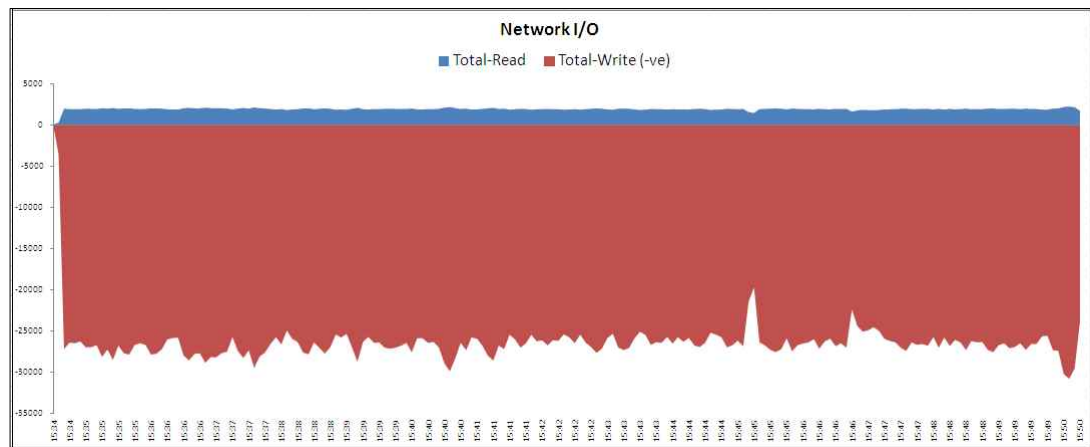
- OSD_1 Server 측정결과



[그림 IV-5. CPU 자원사용률]



[그림 IV-6. Memory 자원사용률]

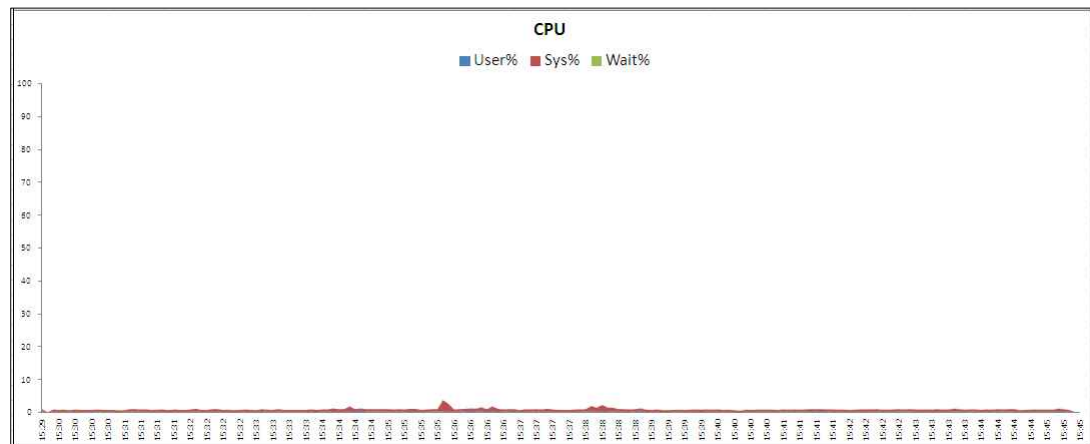


[그림 IV-7. Network 자원사용률]

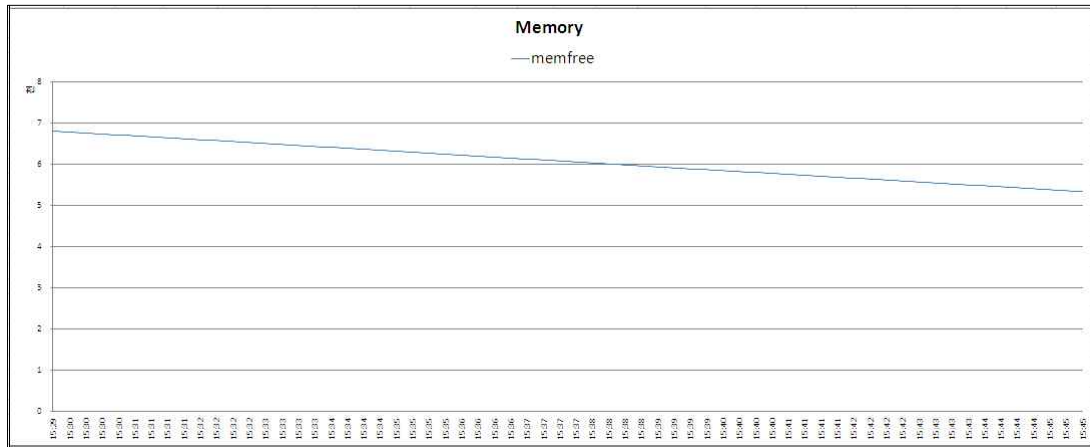
(Read) DIR/MRC Server로부터 데이터를 전송 받는 양

(Write) OSD_2 Server로 데이터를 복제하기 위해 보내는 데이터의 양

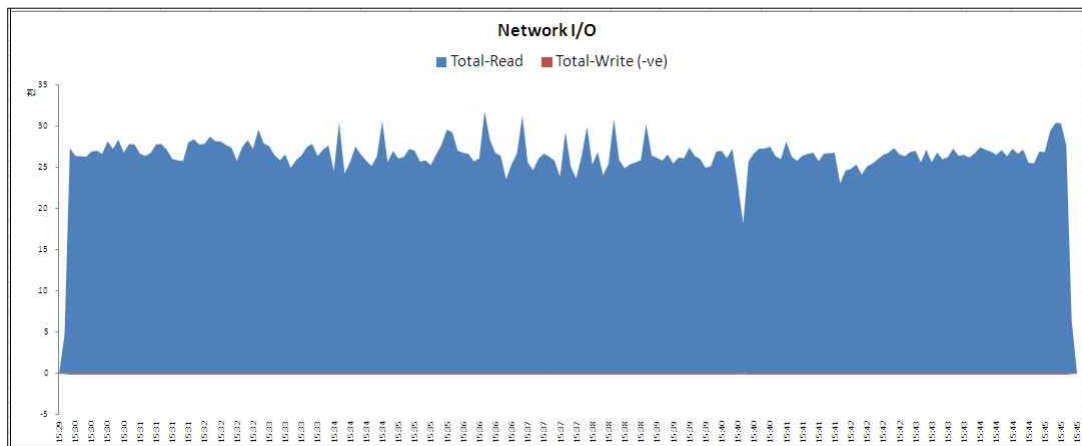
- OSD_2 Server 측정결과



[그림 IV-8. CPU 자원사용률]



[그림 IV-9. Memory 자원사용률]



[그림 IV-10. Network 자원사용률]

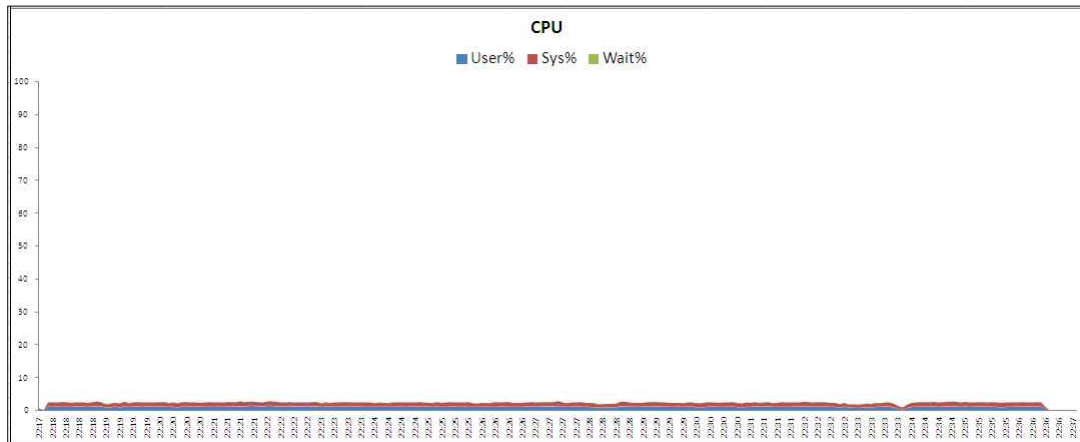
(Read) OSD_1 Server의 데이터를 복제하기 위해 데이터를 전송받는 양

o Replication_256

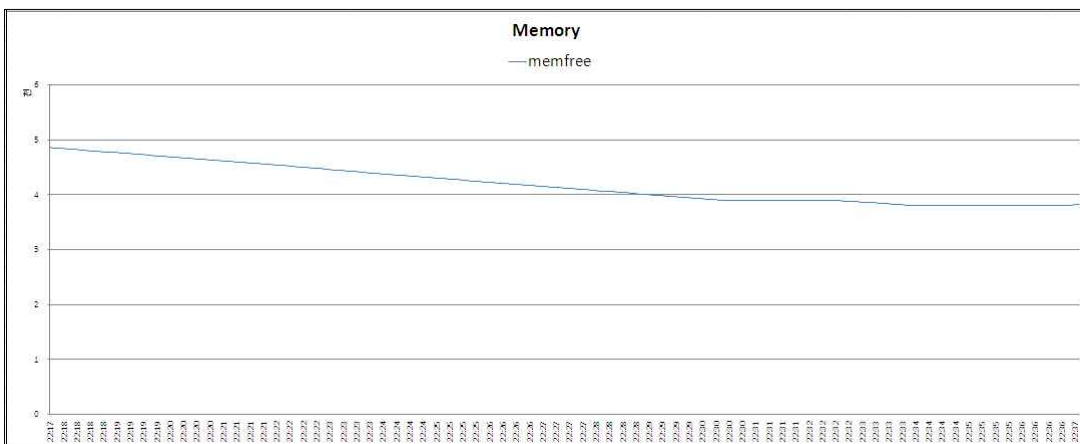
- 수행정보

수행 조건	- Data를 256byte로 분할 - OSD_1 Server에 저장 - OSD_1 Data를 OSD_2 Server로 복제
-------	---

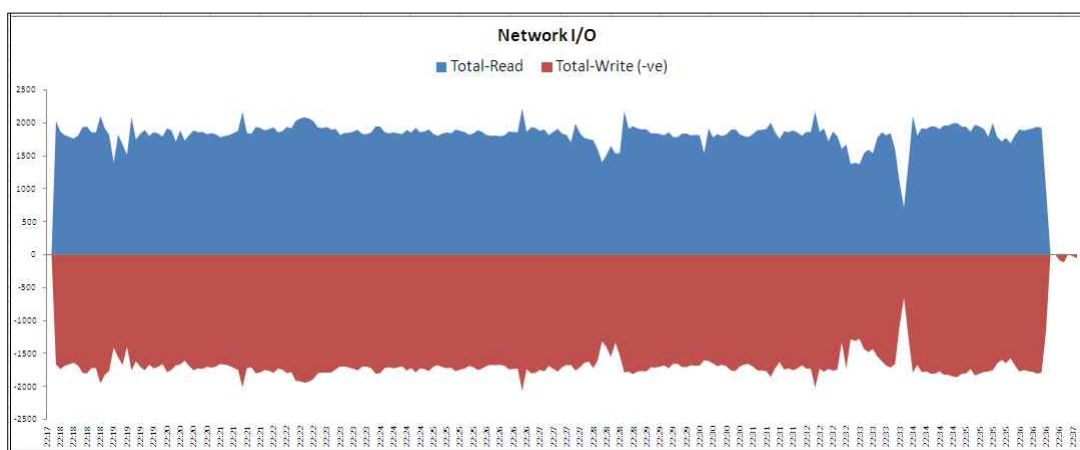
- DIR/MRC Server 측정결과



[그림 IV-11. CPU 자원사용률]



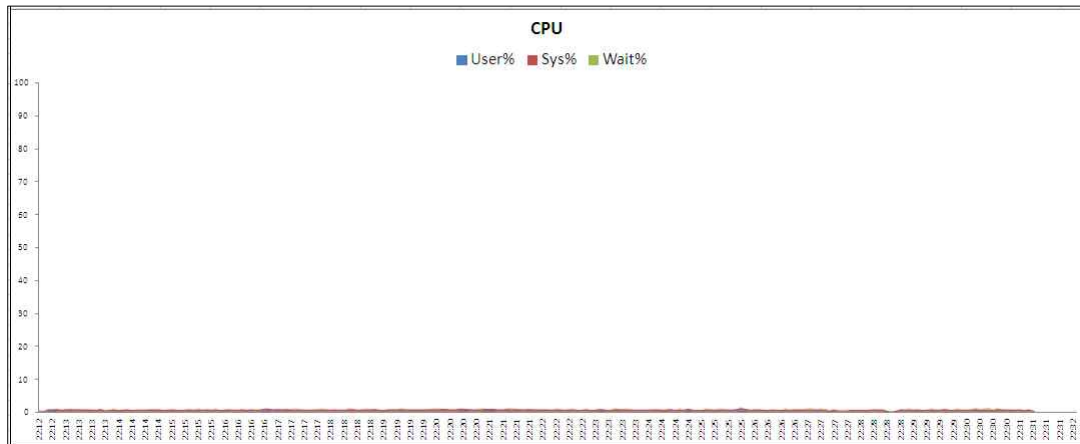
[그림 IV-12. Memory 자원사용률]



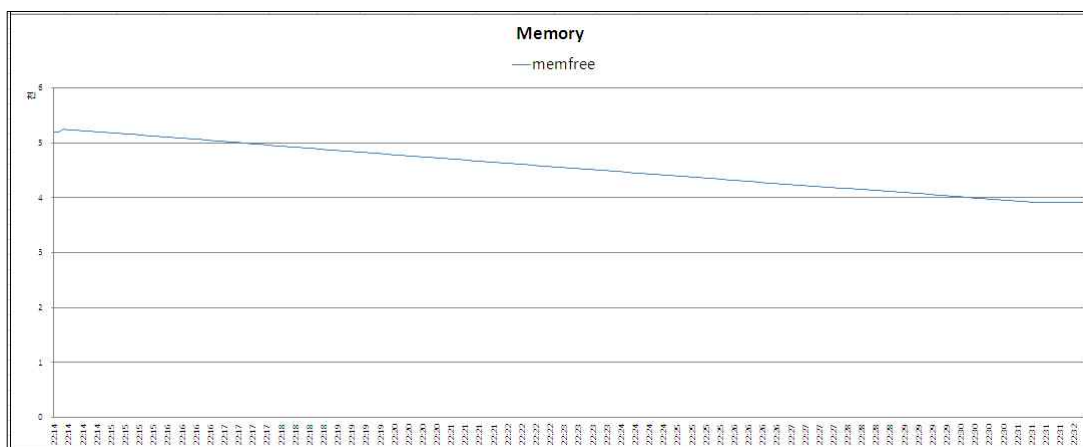
[그림 IV-13. Network 자원사용률]

(공통) DIR/MRC Server는 FileZilla로부터 1.37GB 파일을 직접 전송받기 때문에 Total-Read가 다소 높게 나타남

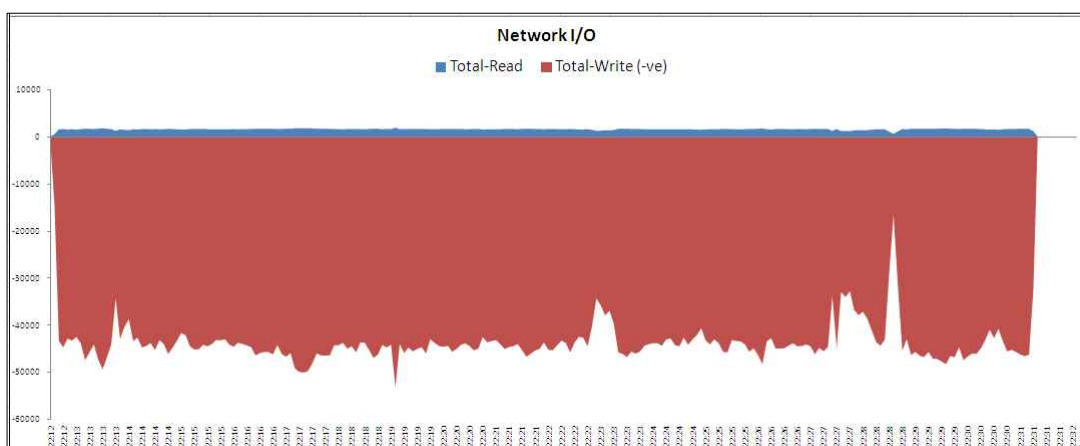
- OSD_1 Server 측정결과



[그림 IV-14. CPU 자원사용률]



[그림 IV-15. Memory 자원사용률]

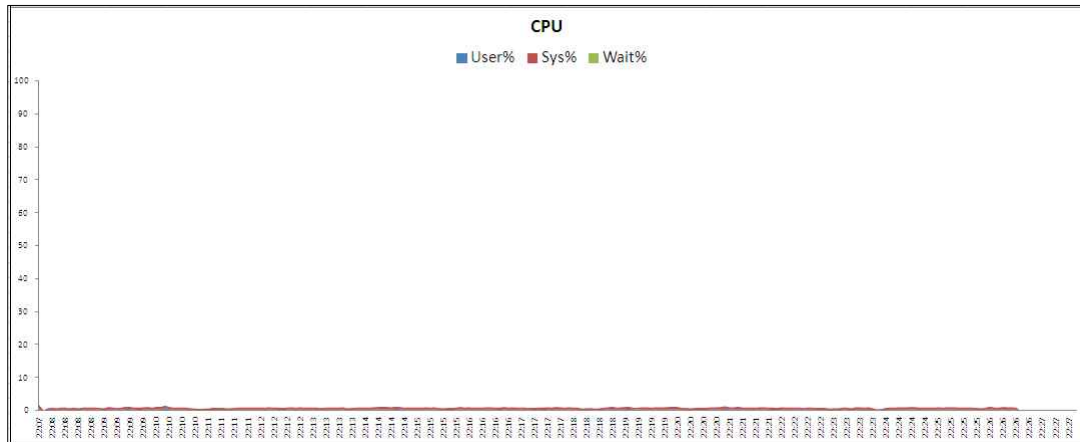


[그림 IV-16. Network 자원사용률]

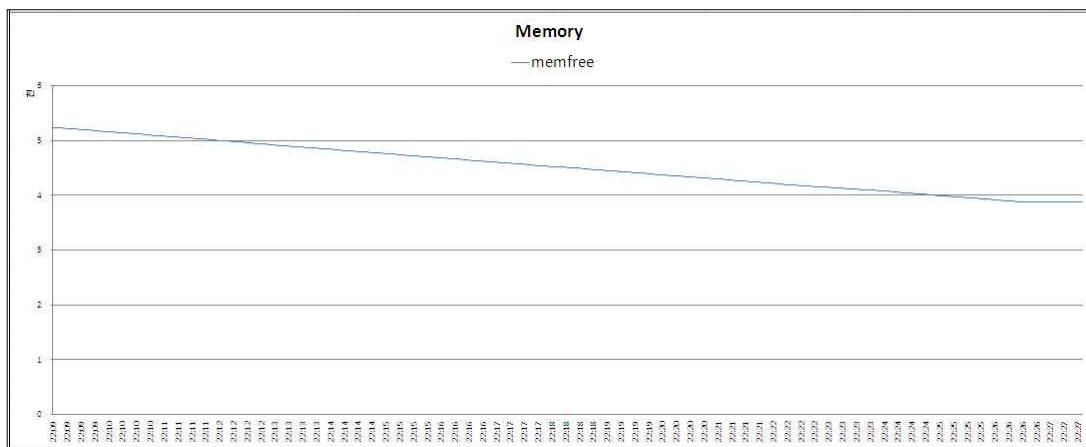
(Read) DIR/MRC Server로부터 데이터를 전송 받는 양

(Write) OSD_2 Server로 데이터를 복제하기 위해 보내는 데이터의 양

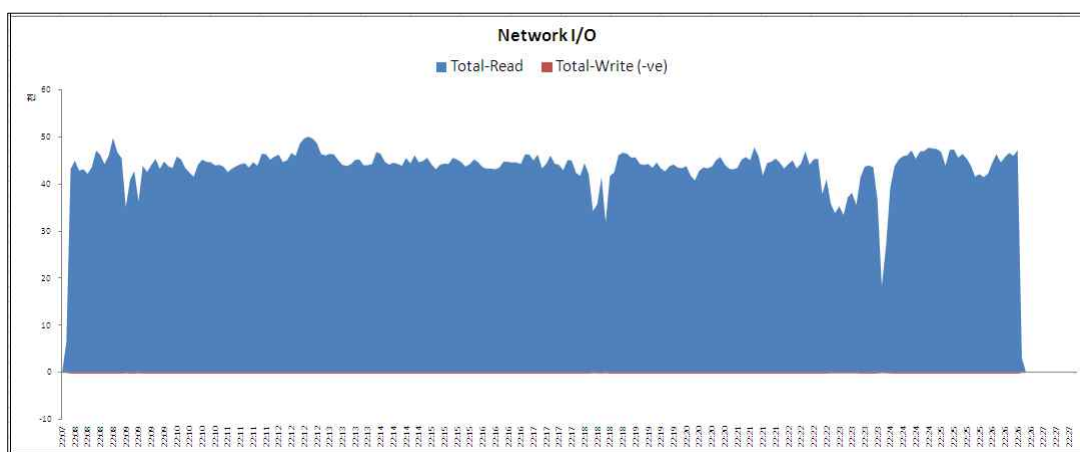
- OSD_2 Server 추정결과



[그림 IV-17. CPU 자원사용률]



[그림 IV-18. Memory 자원사용률]



[그림 IV-19. Network 자원사용률]

(Read) OSD_1 Server의 데이터를 복제하기 위해 데이터를 전송받는 양

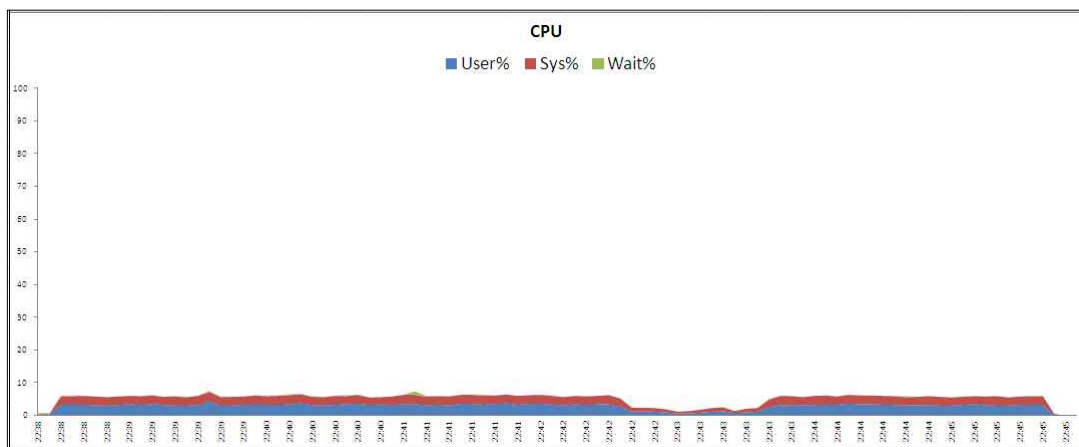
o Striping_128

- 수행 정보

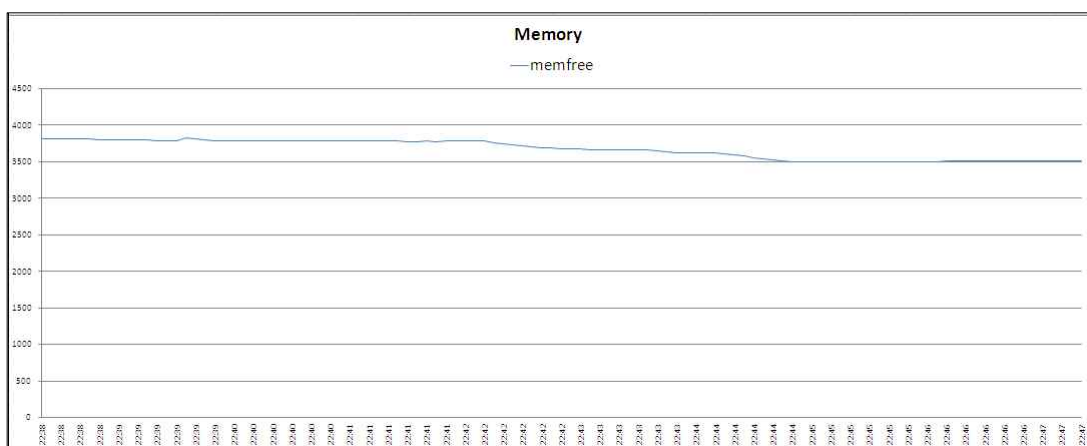
수행 조건	- Data를 128byte로 분할 - OSD_1 Server와 OSD_2 Server에 분산
-------	---

- DIR/MRC Server 측정 결과

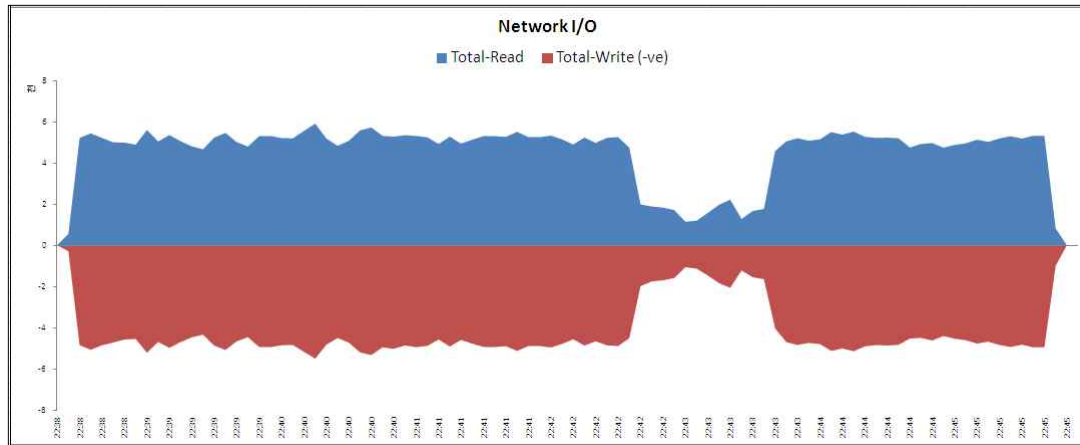
Striping 시 DIR/MRC Server의 DISK I/O 병목 현상으로 인한 Wait 상황이 발생하여, CPU, NetWork 사용 비율이 약 30초간 감소하는 현상이 발생하였으나, 이후 정상 수치를 회복하였으며, 실제 데이터는 정상적으로 Striping 됨



[그림 IV-20. CPU 자원사용률]



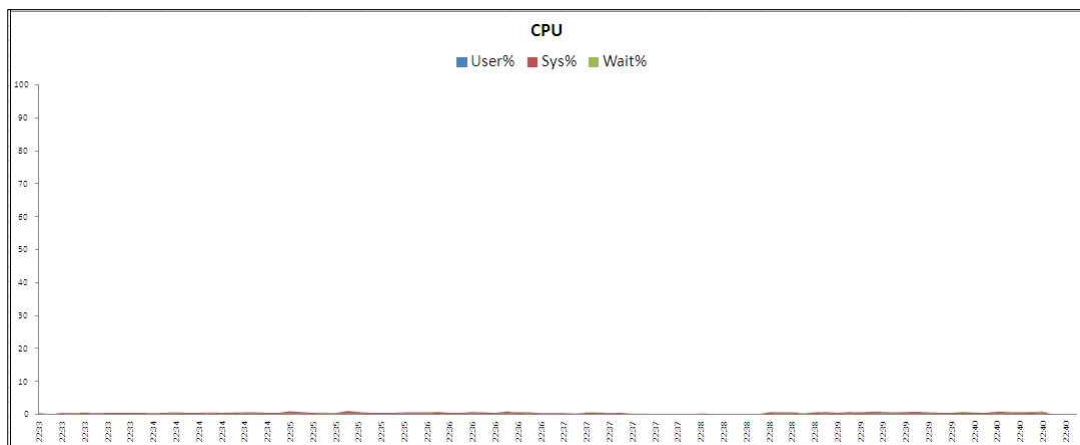
[그림 IV-21. Memory 자원사용률]



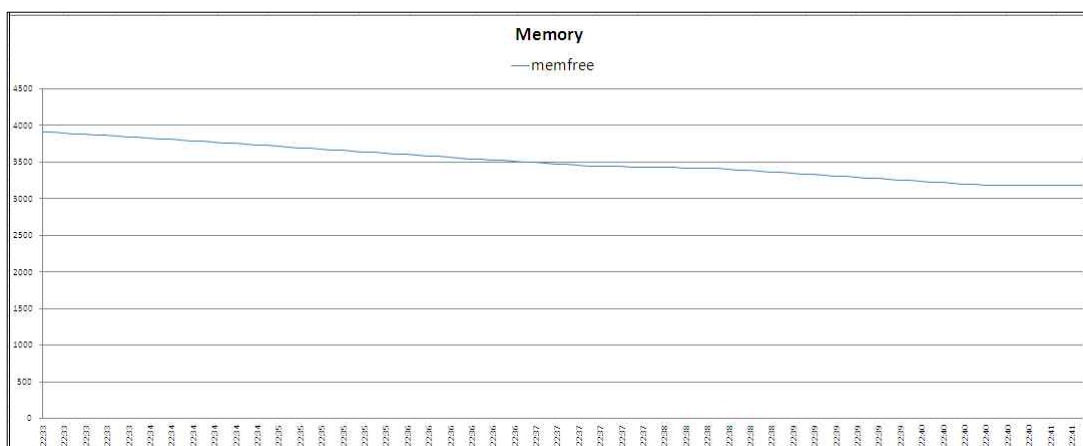
[그림 IV-22. Network 자원사용률]

(공통) DIR/MRC Server는 FileZilla로부터 1.37GB 파일을 직접 전송받기 때문에 Total-Read가 다소 높게 나타남

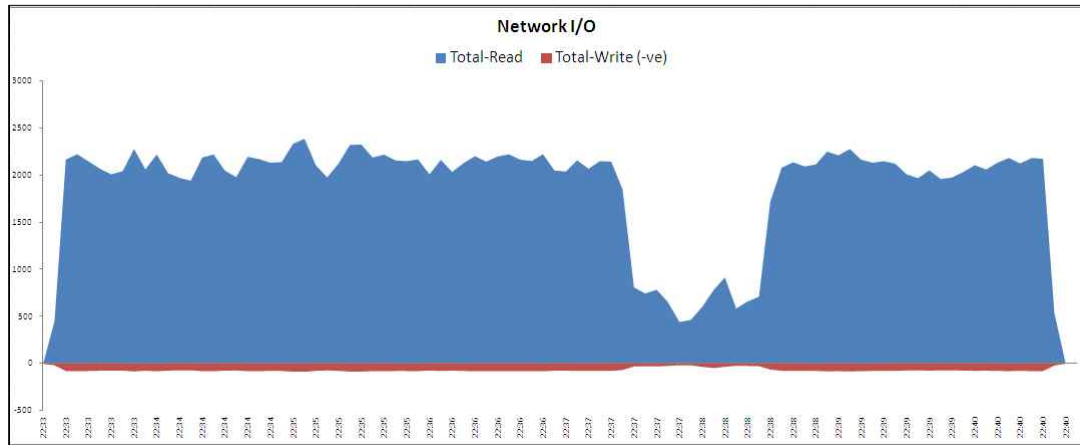
- OSD_1 Server 측정결과



[그림 IV-23. CPU 자원사용률]



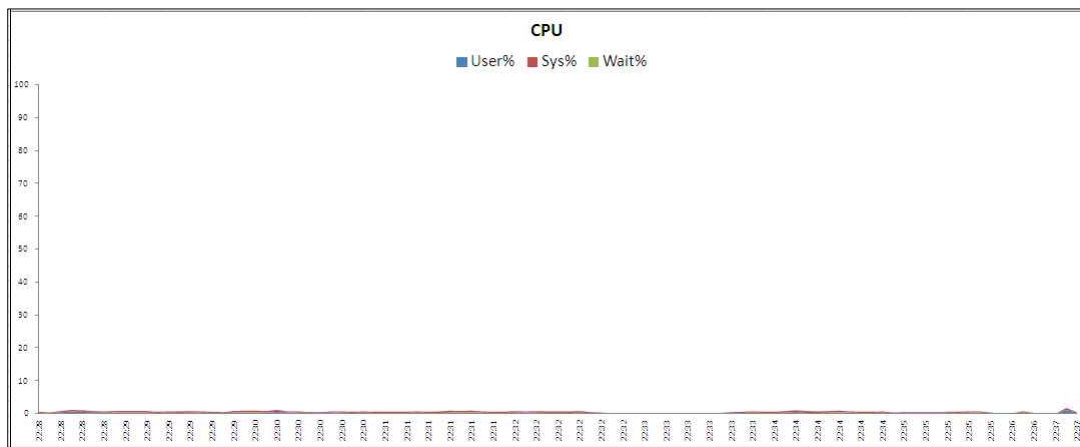
[그림 IV-24. Memory 자원사용률]



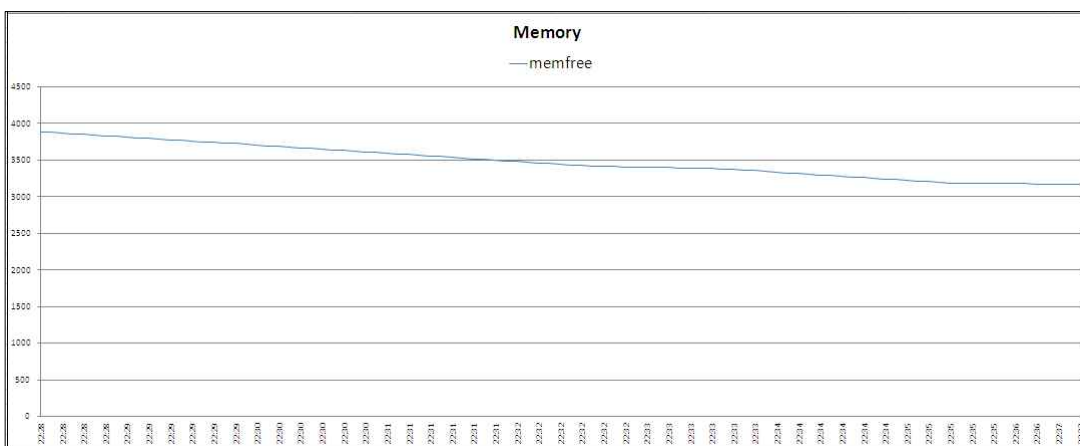
[그림 IV-25. Network 자원사용률]

(Read) DIR/MRC Server로부터 분산되는 데이터를 전송받는 양

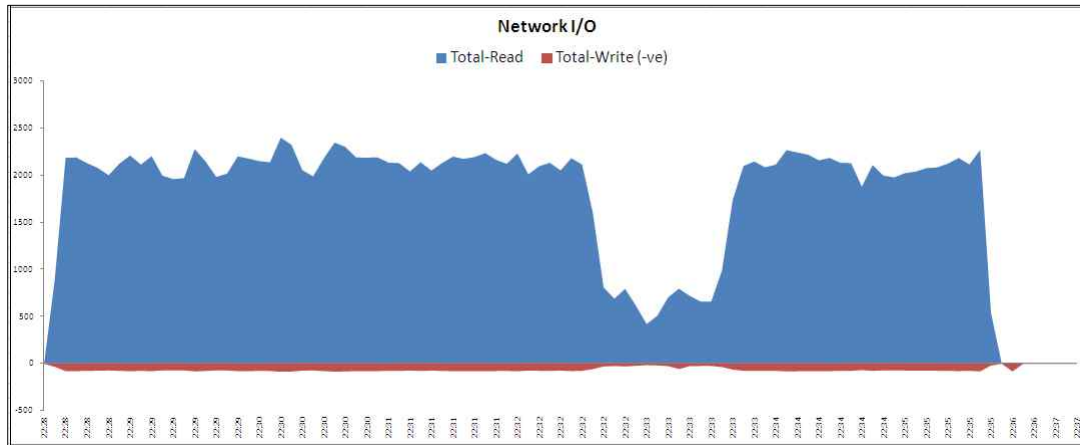
- OSD_2 Server 측정결과



[그림 IV-26. CPU 자원사용률]



[그림 IV-27. Memory 자원사용률]



[그림 IV-28. Network 자원사용률]

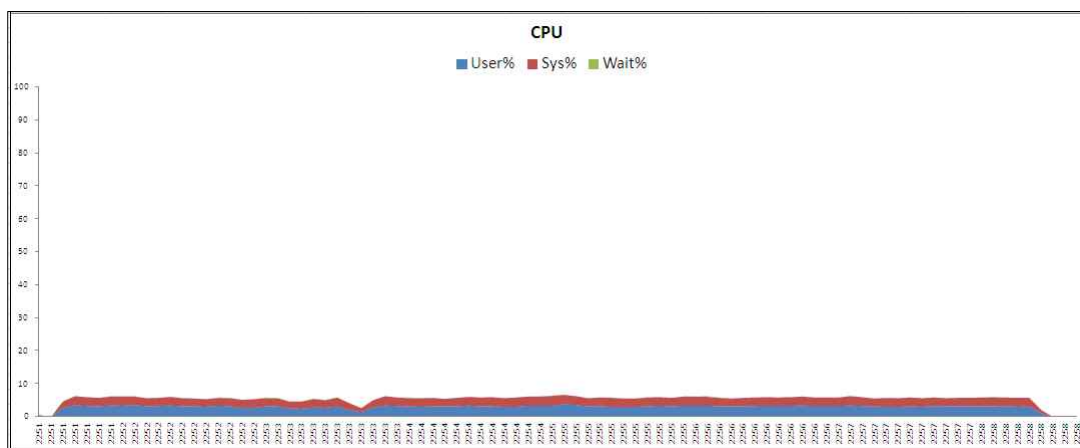
(Read) DIR/MRC Server로부터 분산되는 데이터를 전송받는 양

o Striping_256

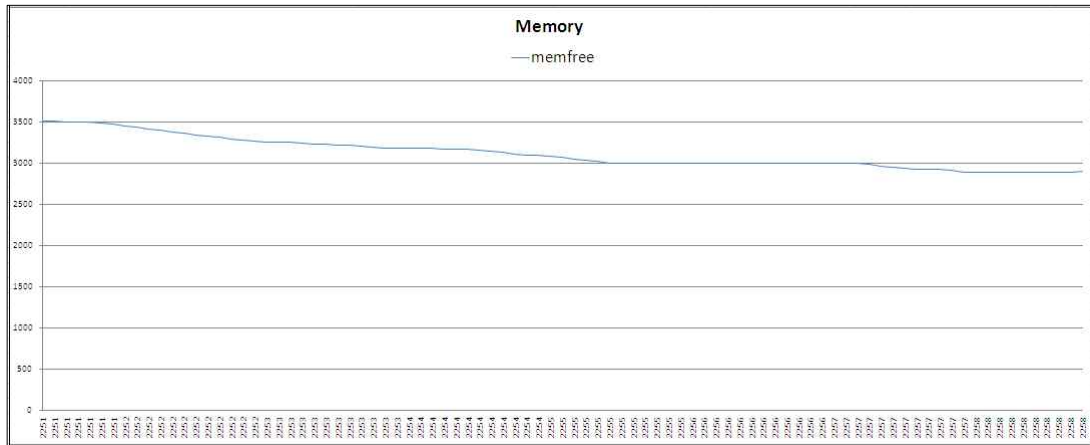
- 수행정보

수행 조건	- Data를 256byte로 분할 - OSD_1 Server와 OSD_2 Server에 분산
-------	---

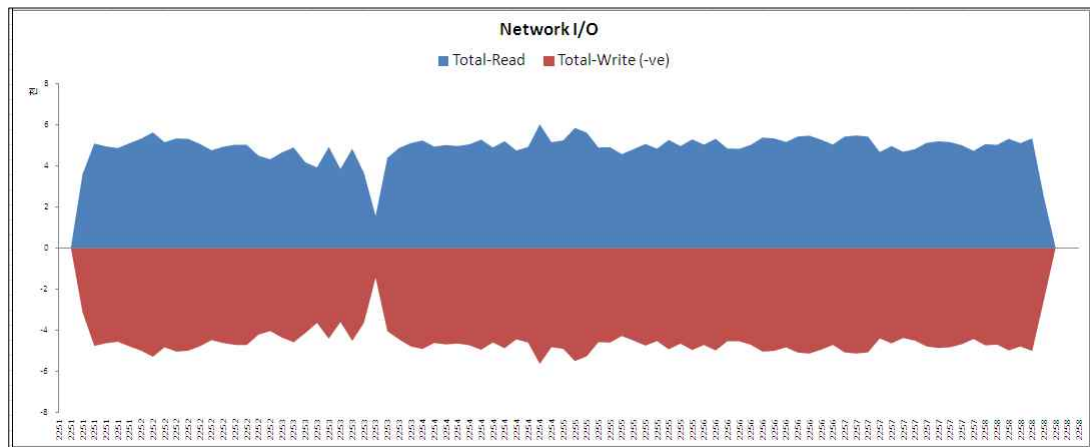
- DIR/MRC Server 측정결과



[그림 IV-29. CPU 자원사용률]



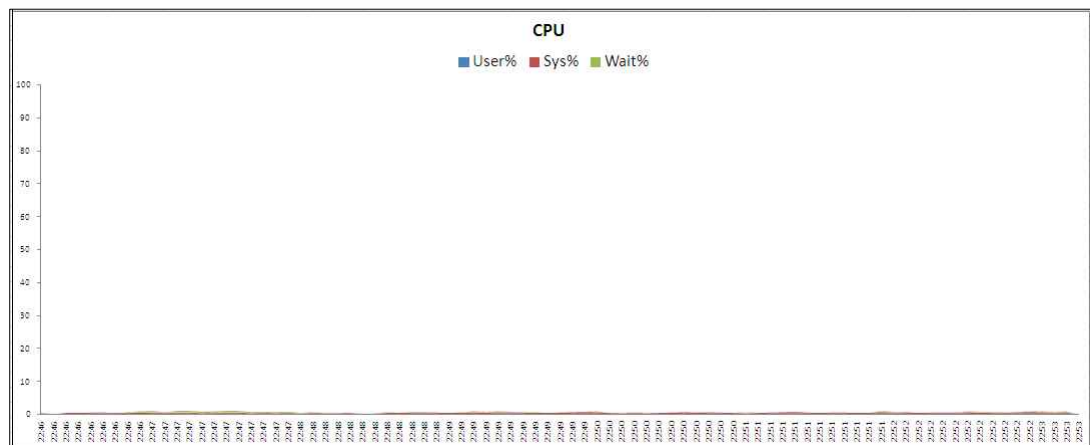
[그림 IV-30. Memory 자원사용률]



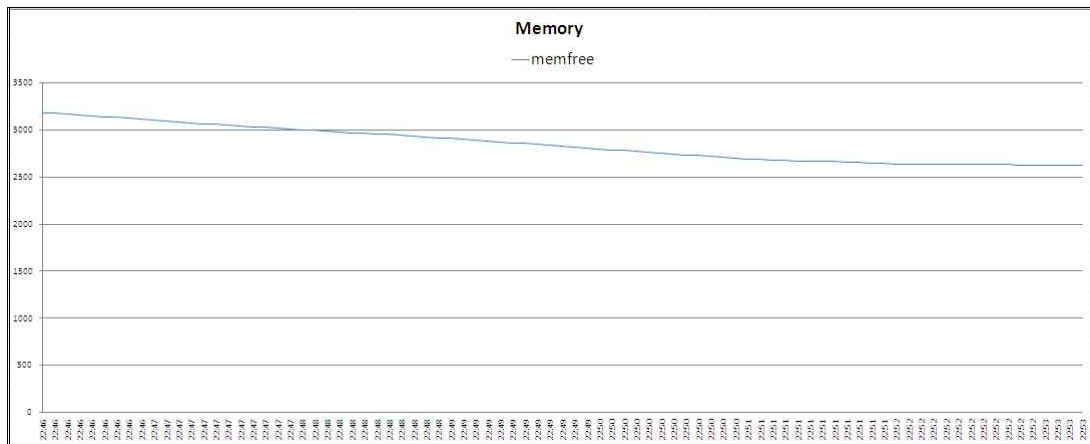
[그림 IV-31. Network 자원사용률]

(공통) DIR/MRC Server는 FileZilla로부터 1.37GB 파일을 직접 전송받기 때문에 Total-Read가 다소 높게 나타남

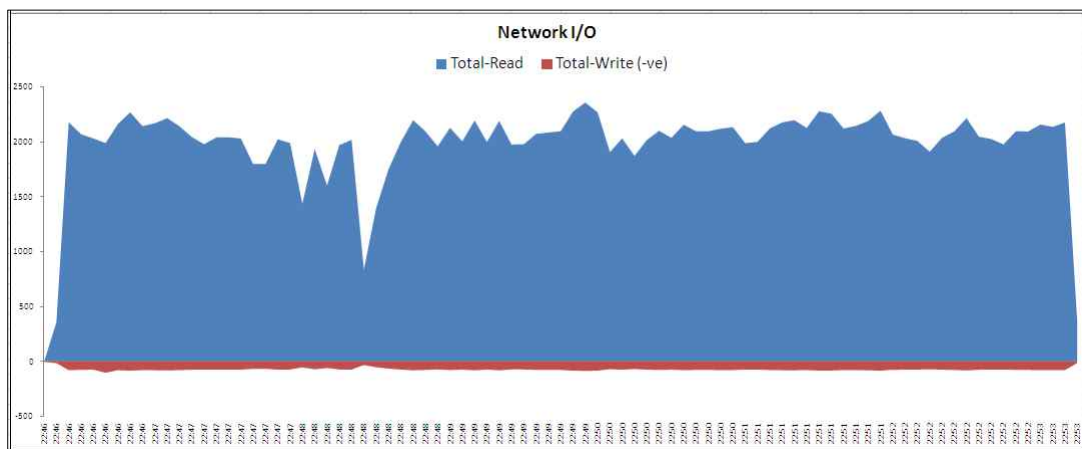
- 측정결과 (OSD_1 Server)



[그림 IV-32. CPU 자원사용률]



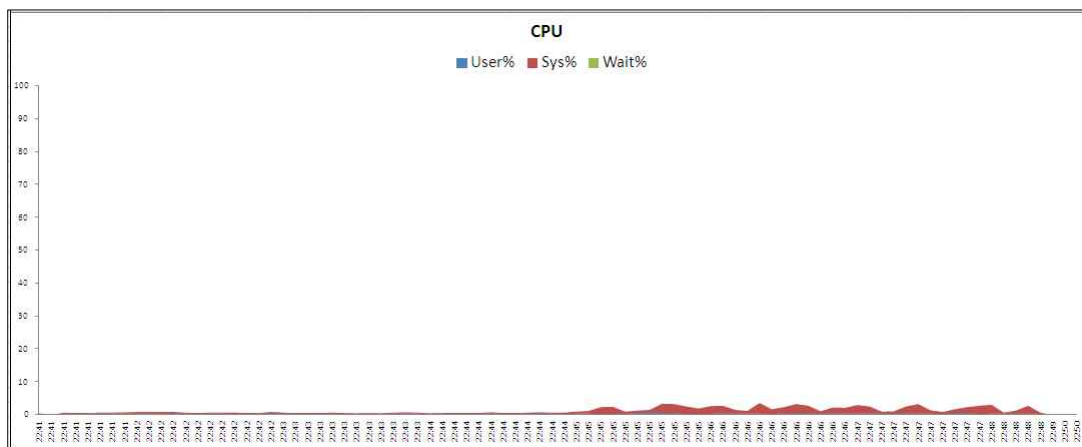
[그림 IV-33. Memory 자원사용률]



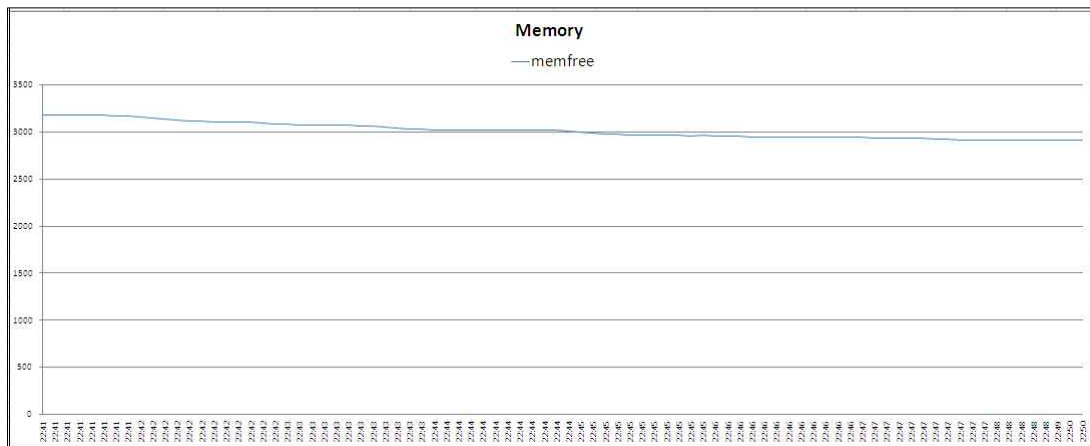
[그림 IV-34. Network 자원사용률]

(Read) DIR/MRC Server로부터 분산되는 데이터를 전송받는 양

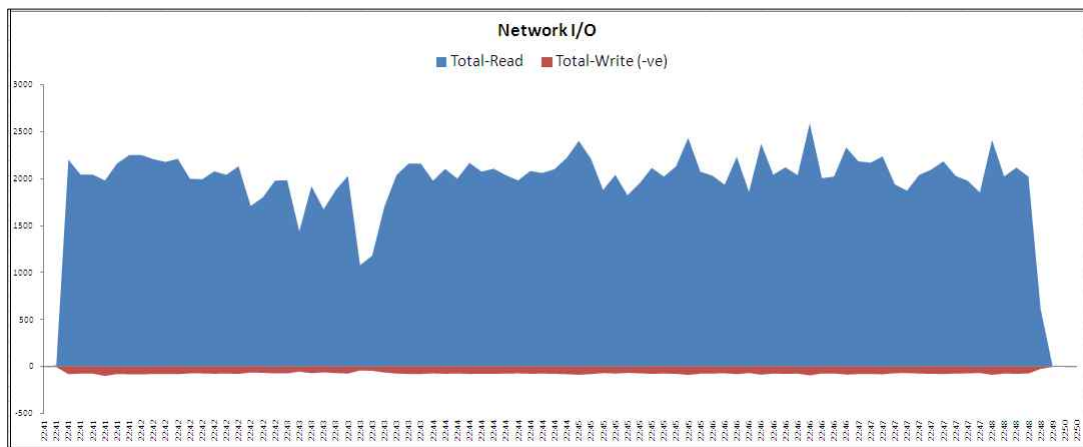
- 측정결과 (OSD_2 Server)



[그림 IV-35. CPU 자원사용률]



[그림 IV-36. Memory 자원사용률]



[그림 IV-37. Network 자원사용률]

(Read) DIR/MRC Server로부터 분산되는 데이터를 전송받는 양

V. 종합

- XtreamFS 기능 테스트 수행 결과 공개SW로 구성된 Stack 상에서 각 기능 시나리오 수행 시 치명적 오류 또는 심각한 장애가 발생하지 않았으며, Stack을 구성하는 각 공개SW가 유기적으로 동작함을 확인하였음

- XtreamFS 성능 테스트 수행 결과 CPU, Memory, NetWork 사용률은 전체적으로 안정적 수치를 보였으나, Striping 시 DIR/MRC Server의 DISK I/O 병목 현상으로 인한 Wait 상황이 발생하였다. 이에 XtreamFS 도입 시 사용 중인 시스템 환경에서의 DISK I/O에 따른 CPU, NetWork 사용률에 대한 분석이 필요할 것으로 추정함

※ 참고 자료

- [1] <http://www.xtreemfs.org/>
- [2] <http://www.centos.org/>
- [3] <http://www.java.com/>
- [4] <http://dpnm.postech.ac.kr/>
- [5] <http://www.filezilla-project.org/>
- [6] <http://en.wikipedia.org/wiki/>
- [7] <http://www.mke.go.kr/>
- [8] The XtreamFS Installation and User Guide