
분산DB 분야 Stack 통합 테스트 결과보고서 [MongoDB]

2012. 09.

목 차

I. Stack 통합 테스트 개요	3
1. 목적	3
II. 분산DB 소개	4
1. 분산DB	4
2. 분산DB 분야 주요 공개SW	6
III. 테스트 대상 소개	7
1. MongoDB 소개	7
IV. Stack 통합 테스트	8
1. 테스트 환경	8
2. 주요 테스트 방법	9
3. 기능 테스트 수행 결과	10
4. 성능 테스트 수행 결과	11
V. 종합	27
※ 참고자료	28
[별첨 1] MongoDB 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. 분산DB 소개

1. 분산 DB(NoSQL)

NoSQL은 관계형 데이터베이스 관리 시스템(RDBMS)과 다른 새로운 형태의 데이터베이스 관리 시스템 클래스이다. NoSQL과 같은 데이터 저장소는 고정된 테이블 스키마를 필요로 하지 않으며, Join 연산이 없고, 수평적 확장을 지원한다.

NoSQL 구조는 eventual consistency¹⁾ 같은 일관성을 보장하며 트랜잭션은 1개의 항목에 대해서만 지원한다. 하지만 일부 시스템에서는 AppScale²⁾이나 CloudTPS³⁾같은 보조 미들웨어를 통해서 ACID⁴⁾ 기능을 보장한다.

NoSQL 시스템은 여러 서버에 이중화 방식의 분산 구조를 사용하며 데이터는 분산 해시 테이블을 이용하여 여러 대의 서버에 중복되어 저장된다. 이러한 방식으로 시스템은 필요 시 서버를 추가하는 방식으로 확장 가능하다.

일부 NoSQL은 연관배열(associative arrays) 혹은 키 값(key-value) 쌍 같은 단순한 인터페이스를 선호한다. 반면 XML 데이터베이스 같은 시스템에서는 XQuery 표준을 지원한다. CloudTPS 같은 시스템은 Join 쿼리를 지원하기도 한다. (출처 : WIKIPEDIA)

1) 특정 시점에는 일관성이 결여되는 경우가 발생하지만, 일정 시간이 경과 후에는 일관성을 유지함

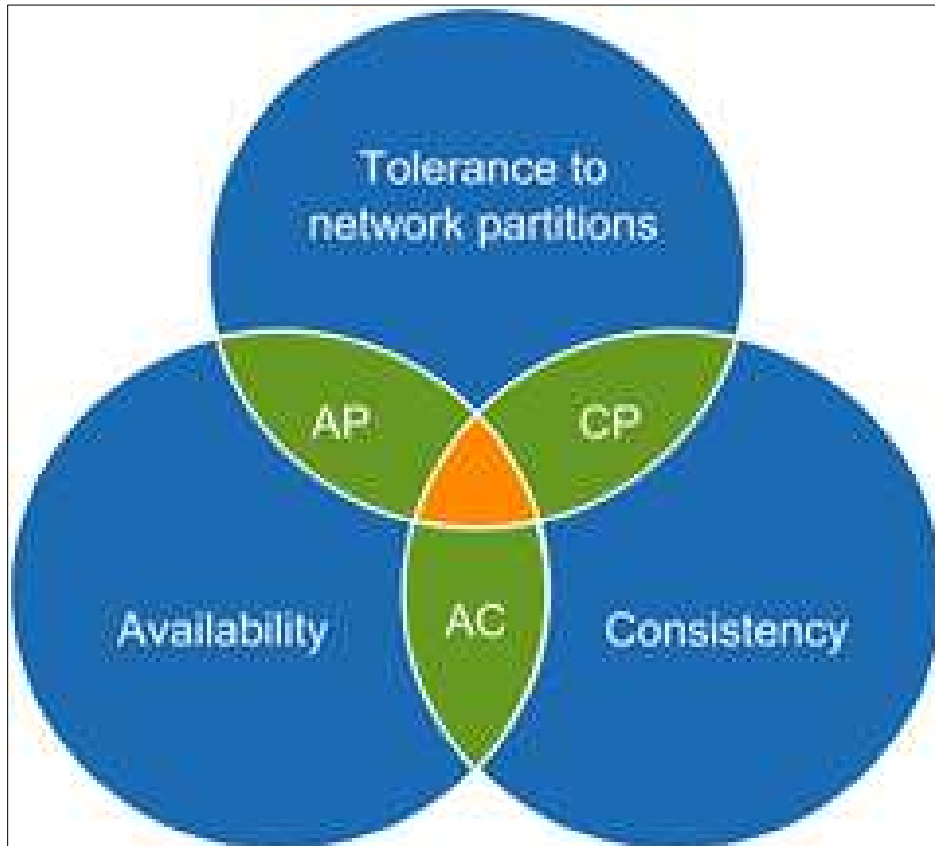
2) IaaS 클라우드에서 Google App Engine 애플리케이션을 쉽게 실행할 수 있게 하는 클라우드 컴퓨팅 플랫폼

3) 클라우드 웹 응용 프로그램을 위한 확장 가능한 트랜잭션

4) 데이터베이스 트랜잭션이 안전하게 수행된다는 것을 보장하기 위한 성질을 가리키는 약어로 원자성, 일관성, 독립성, 지속성을 지칭한다.

□ CAP 정리

CAP 정리에 따르면, 대규모 분산데이터시스템의 속성을 3가지로 구분하였으며, 최소 2가지를 만족해야한다.



[그림 II-1. CAP 이론 다이어그램]

- 일관성(Consistency)
모든 노드는 동시에 같은 데이터를 참조해야 함
- 가용성(Availability)
모든 요청이 성공하거나, 실패에 상관없이 응답을 보장해야 함
- 지속성(Partition tolerance)
일부 데이터를 손실하더라도 사용자는 사용이 가능해야 함

2. 분산DB 분야 주요 공개SW

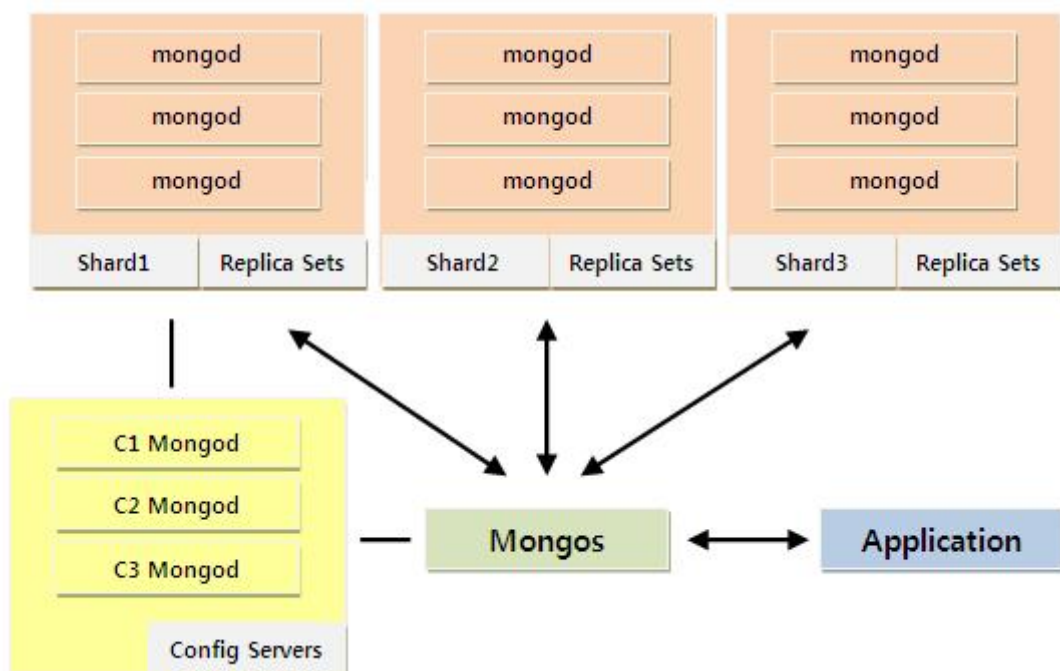
[표 II-1. 주요 공개SW]

제품명	Stack 환경		홈페이지	비고
MongoDB	OS	Cross-platform	http://www.mongodb.org	GNU AGPL v3.0
Cassandra	OS	Cross-platform	http://cassandra.apache.org	Apache License 2
	JAVA	JDK 6		
Cloudata	OS	Linux	http://www.cloudata.org	Apache License 2
	JAVA	JDK 6		
CouchDB	OS	Cross-platform	http://couchdb.apache.org	Apache License 2
HBase	OS	Cross-platform	http://hbase.apache.org	Apache License 2
	JAVA	JDK 6		

III. 테스트 대상 소개

1. MongoDB 소개

MongoDB는 C++ 언어로 작성된 NoSQL 데이터베이스로, SQL문을 MongoDB 쿼리 함수 호출로 매우 쉽게 변환할 수 있기 때문에 관계형 데이터베이스를 MongoDB로 쉽게 마이그레이션 할 수 있는 강력한 문서 지향 쿼리 언어이다.



[그림 III-1. MongoDB 구성 요소]

IV. Stack 통합 테스트

1. 테스트 환경

☐ MongoDB 환경

[표 IV-1. MongoDB 환경]

모듈	Version
MongoDB	2.0.6

☐ Stack 환경

[표 IV-2. Stack 환경]

구성	OS	비고
A Stack	CentOS 6.2 (64bit)	Master (DL360G6)
	CentOS 6.2 (64bit)	Slave (X3850M2)
B Stack	Windows Server 2008 R2 (64bit)	Master (DL360G6)
	Windows Server 2008 R2 (64bit)	Slave (X3850M2)

☐ HW 환경

[표 IV-3. HW 환경]

제조사	모델명	CPU	MEM	Disk	NIC
IBM	X3850M2	2.40GHz (Six Core) x 4, 12MB L3 90w	8GB	146G x 4	Gigabit
HP	DL360G6	2.40 GHz (Quad-Core) x 2	8GB	146G x 5	Gigabit

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 MongoDB를 사용하는 사용자들이 사용할 수 있는 항목 중 MongoDB 시작/종료, Monitoring, Collection, 계정 관리, 데이터 관리, 복제 등에 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 「MongoDB 테스트 케이스」 문서를 참고한다.

□ 테스트 시나리오 현황

[표 IV-4. 테스트 시나리오 현황]

기 능	테스트 시나리오	테스트 케이스
시작/종료	1	6
Monitering	2	4
Collection	4	12
계정 관리	4	27
데이터 관리	2	18
복제	3	46
합 계	16	113

□ 테스트 결과

기능 시나리오를 통한 테스트 수행 결과 데이터 관리, 복제 등 모든 기능이 예상 결과와 동일하게 동작함을 확인하였다.

[표 IV-5. 테스트 결과]

분류		PASS		FAIL		N/A	
기능	개수	A Stack	B Stack	A Stack	B Stack	A Stack	B Stack
시작/종료	6	6	6	0	0	0	0
Monitering	4	4	4	0	0	0	0
Collection	12	12	12	0	0	0	0
계정 관리	27	27	27	0	0	0	0
데이터 관리	18	18	18	0	0	0	0
복제	46	46	46	0	0	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 웹서비스(mongo-php-driver⁵⁾)를 활용하여, MongoDB 데이터 조회 및 입력에 대한 시간반응성 및 자원효율성을 측정하였다.

□ 테스트 시나리오

[표 IV-6. 테스트 시나리오]

시나리오 ID	시나리오
SC_NS_M_1	Replication_Insert(데이터 순차 입력)
SC_NS_M_2	Replication_Insert_Random(데이터 랜덤 입력)
SC_NS_M_3	Replication_Select(데이터 검색)

□ 서버 구성

[표 IV-7. 서버 설정 정보]

구분		항목
WEB	PHP	memory_limit = 512M max_input_time = 60 max_execution_time = 60
	HTTP	MaxClients 150 KeepAliveTimeout 5 Timeout 300
MongoDB (Java Heap Memory)		A Stack : 2G B Stack : 1G

5) MongoDB와 연동하기 위한 PHP Library

□ 측정 항목

[표 IV-8. 측정 항목]

항목	내용
ThroughPut	단위 시간당 처리하는 프로세스의 개수
Response Time	입력이 주어지고 나서 반응하기까지 걸린 시간
자원사용률	CPU, Memory 등 서버 자원사용 현황

□ 테스트 결과

o Replication_Insert

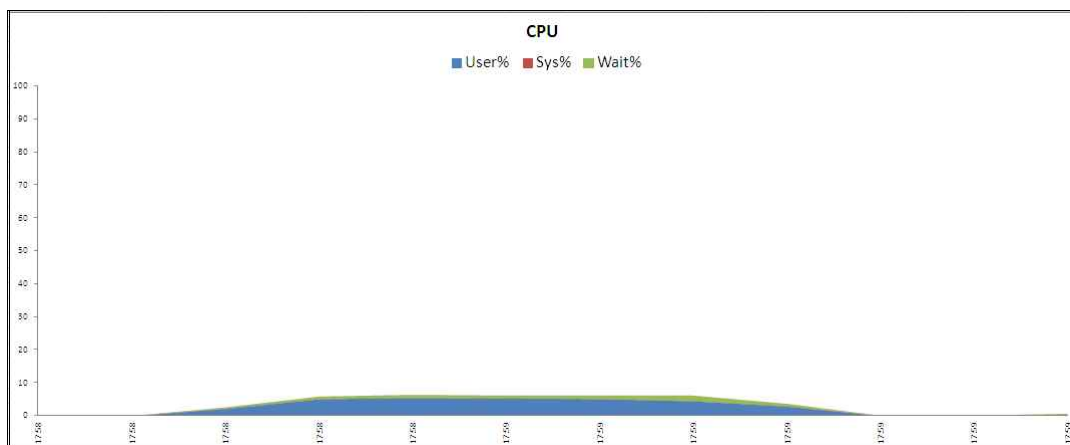
- 수행 정보

수행 조건	<ul style="list-style-type: none"> - Master Server와 Slave Server로 구성 - PHP에서 500,000건 데이터 입력 - Master Server의 데이터가 Slave Server로 복제
-------	--

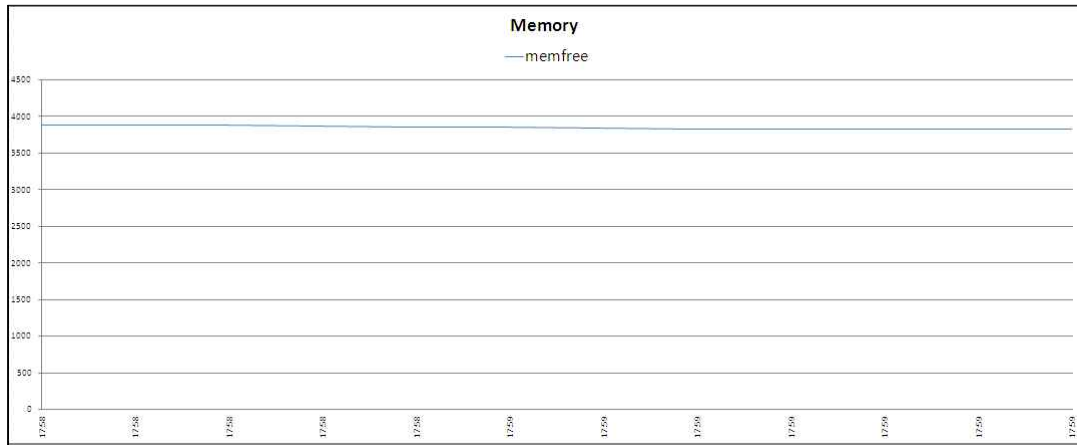
- 측정 결과(A Stack)

데이터 입력 시간은 29.5초 소요되었으며, 총 40000348byte의 데이터가 저장되었음

• Master

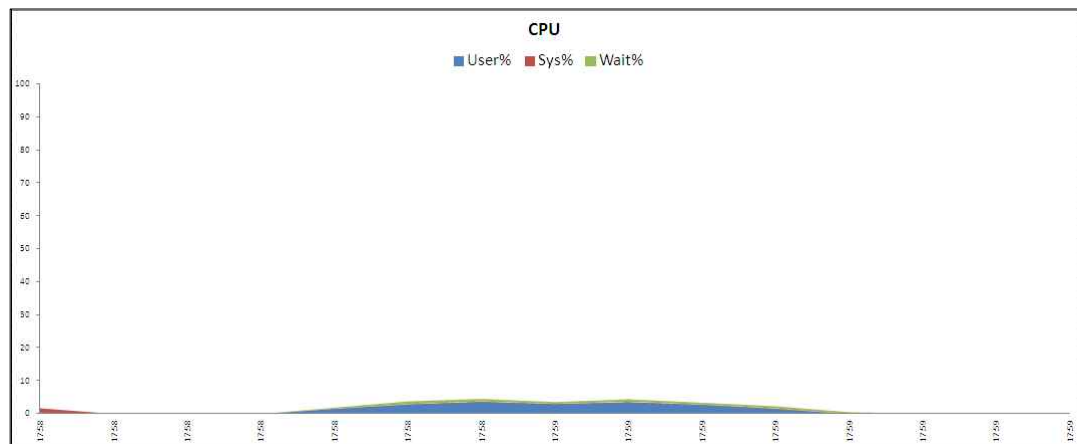


[그림 IV-1. CPU 자원사용률]

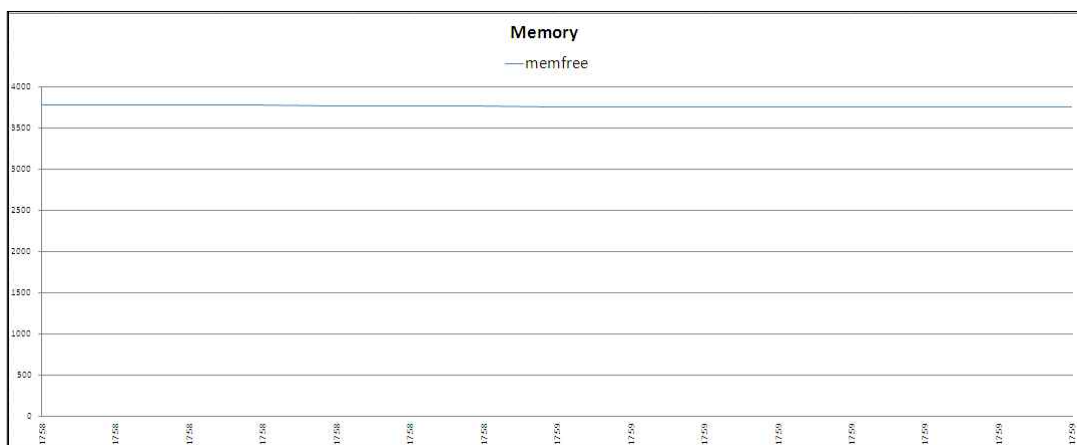


[그림 IV-2. Memory 자원사용률]

- Slave



[그림 IV-3. CPU 자원사용률]

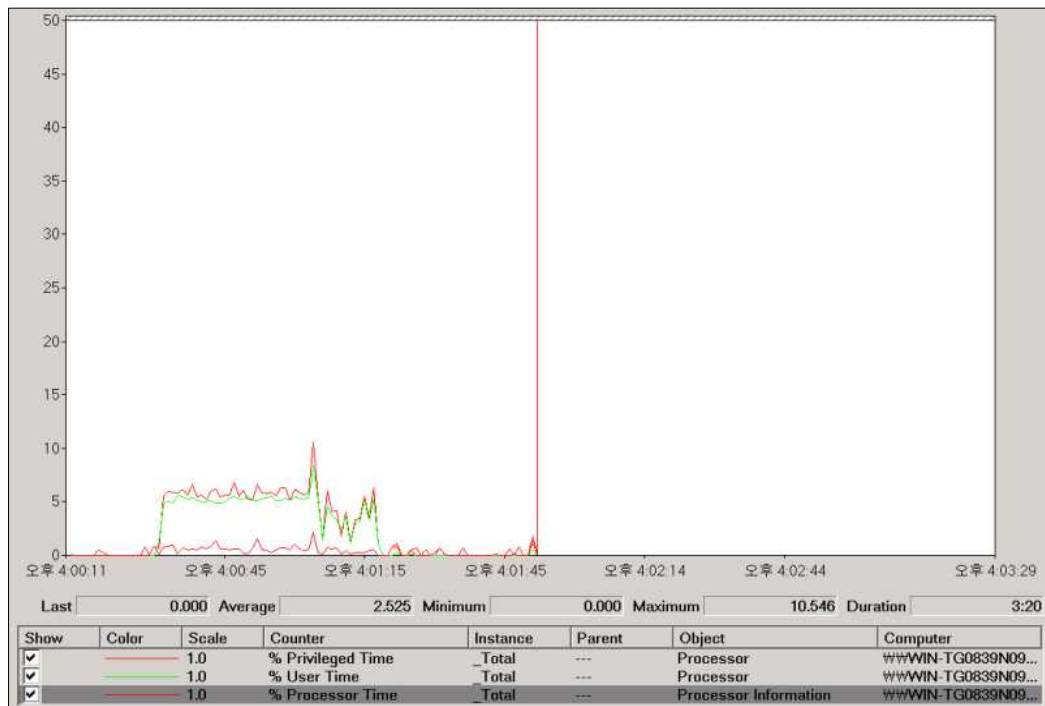


[그림 IV-4. Memory 자원사용률]

- 측정결과(B Stack)

데이터 입력 시간은 46.1초 소요되었으며, 총 40000348byte의 데이터가 저장되었음

• Master

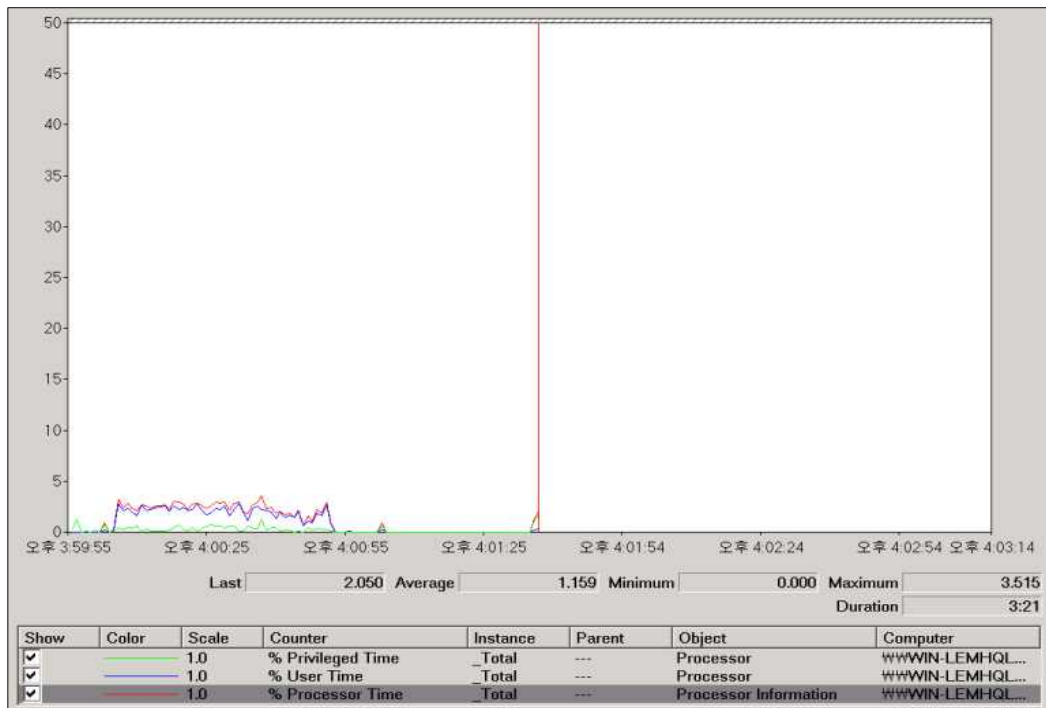


[그림 IV-5. CPU 자원사용률]



[그림 IV-6. Memory 자원사용률]

- Slave



[그림 IV-7. CPU 자원사용률]



[그림 IV-8. Memory 자원사용률]

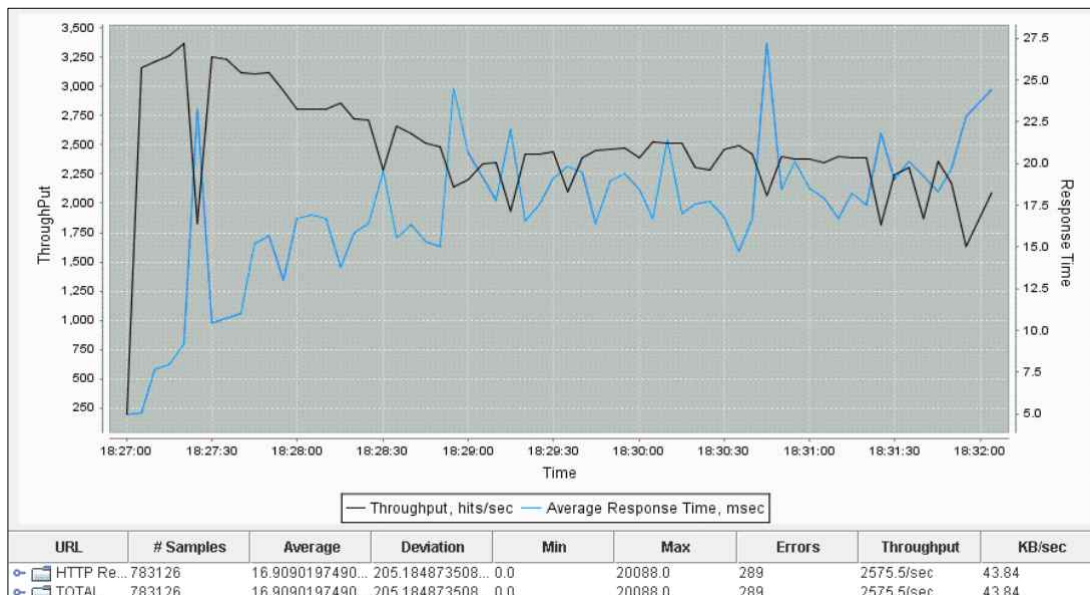
o Replication_Insert_Random

- 수행 정보

수행 조건	<ul style="list-style-type: none"> - Master Server와 Slave Server로 구성 - Ramp-Up : 5User - Running-Time : 5분 - 동시사용자 : 50User - UUID⁶⁾를 생성하여 중복되지 않은 키 생성 및 데이터 입력
-------	--

- 측정 결과(A Stack)

총 90809440byte의 데이터가 저장되었으며, 0.04%의 Connect Error가 발생함

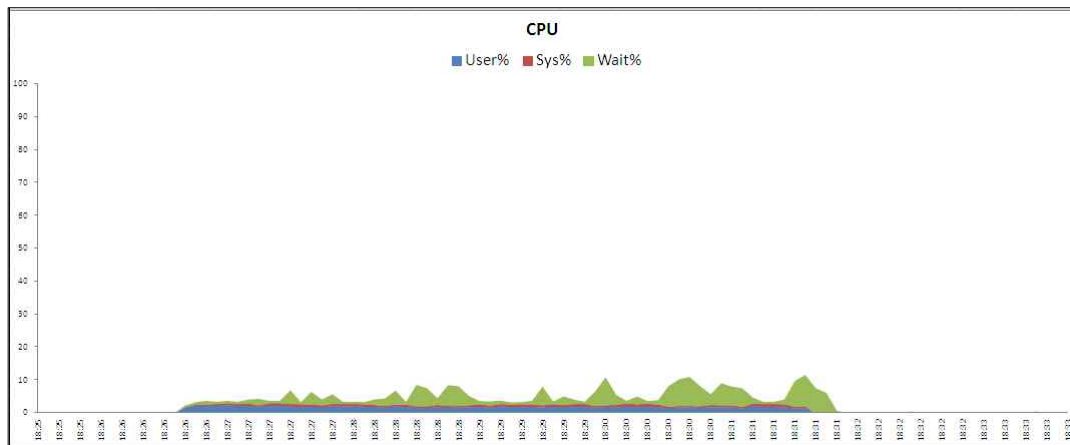


[그림 IV-9. Statistical Aggregate Report]

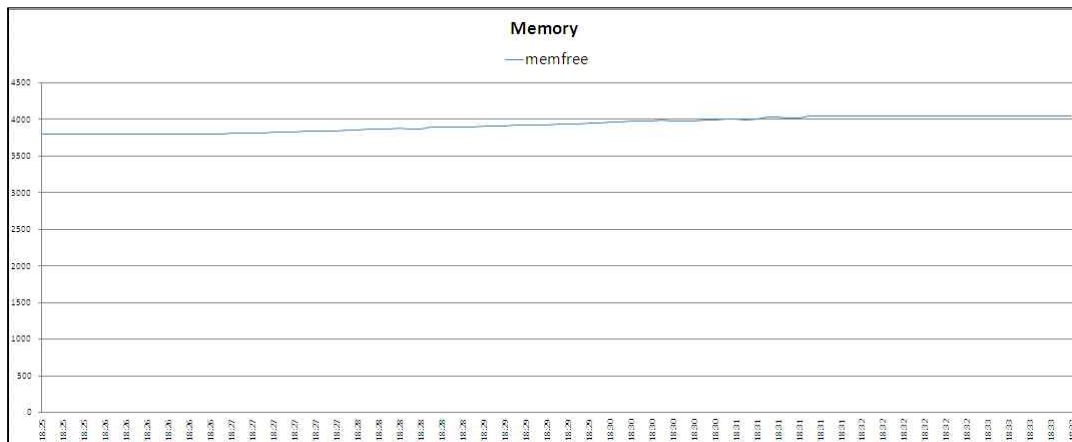
응답시간은 평균 1초 이내의 응답을 보이고 있음

6) Universally Unique Identifier의 약자로 128비트 고유한 데이터 값

- Master

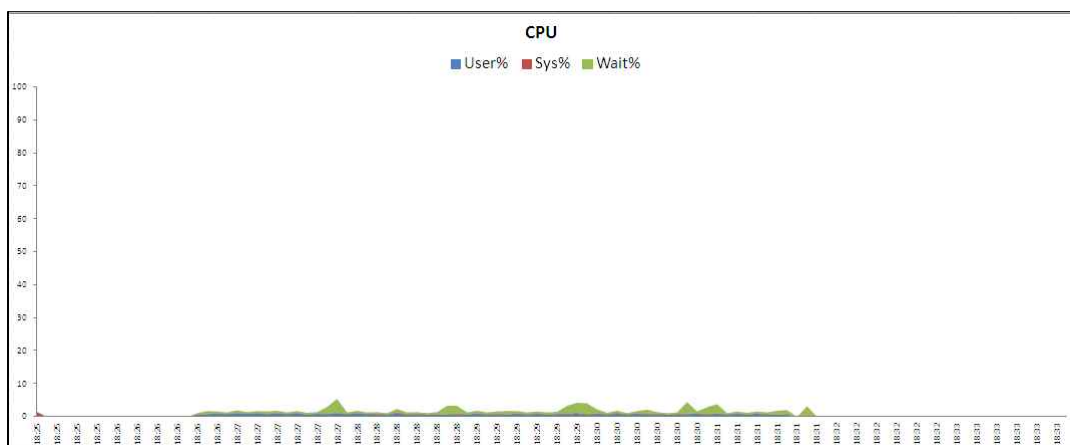


[그림 IV-10. CPU 자원사용률]

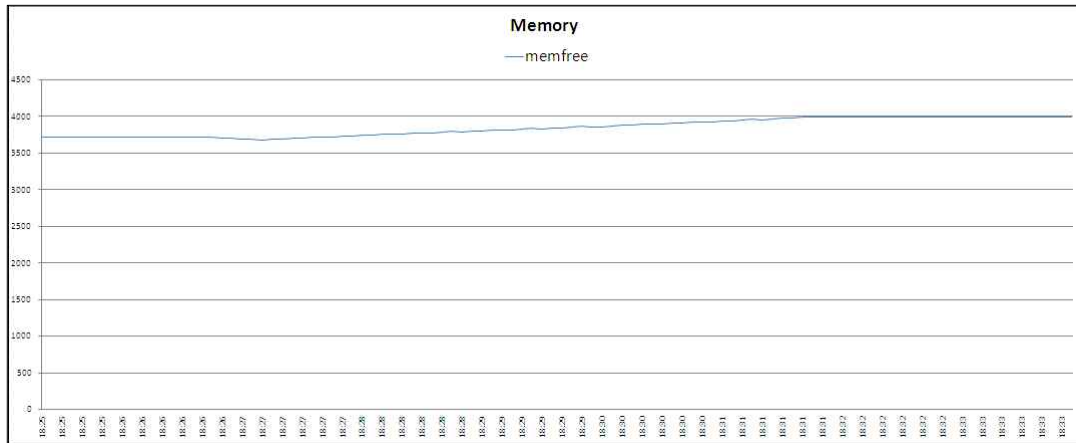


[그림 IV-11. Memory 자원사용률]

- Slave



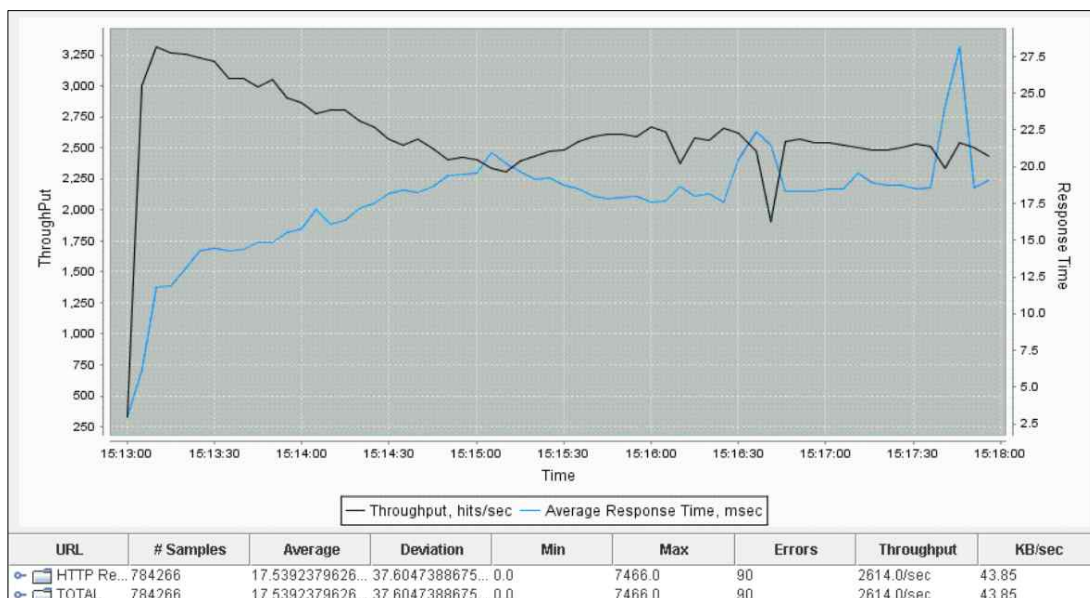
[그림 IV-12. CPU 자원사용률]



[그림 IV-13. Memory 자원사용률]

- 측정결과(B Stack)

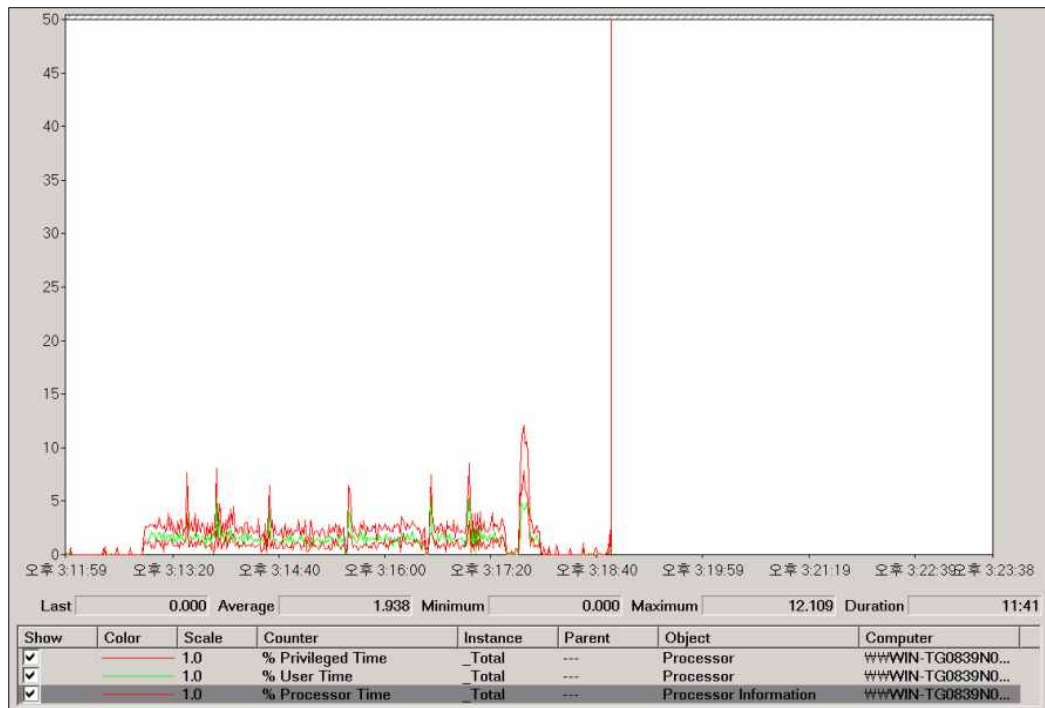
총 90964764byte의 데이터가 저장되었으며, 0.01%의 Connect Error가 발생함



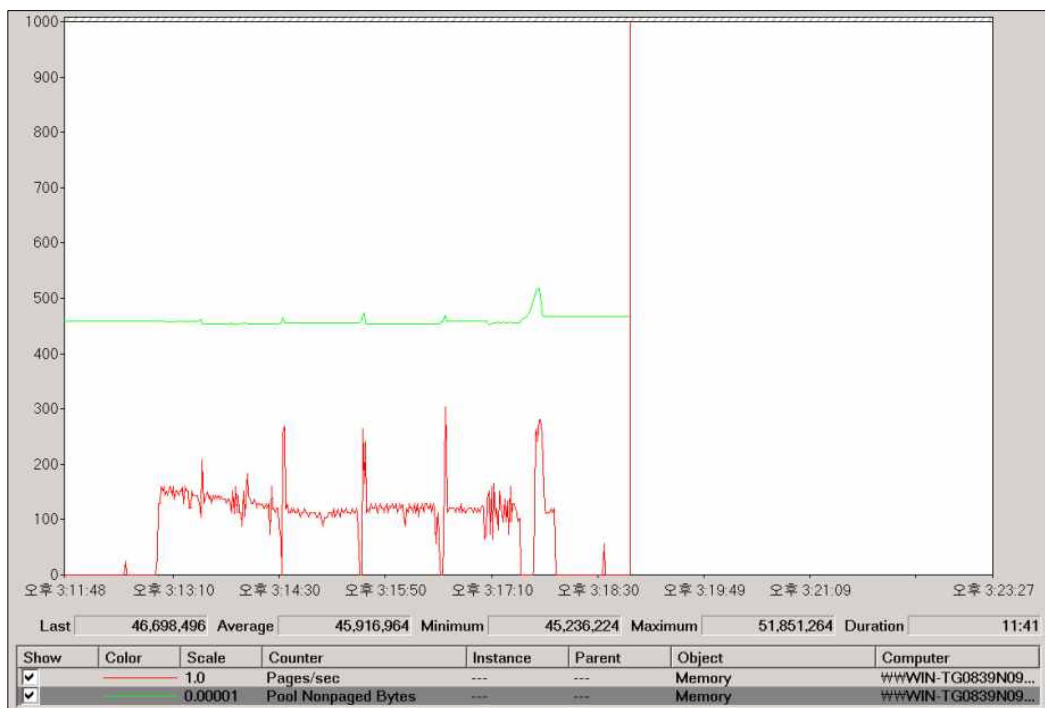
[그림 IV-14. Statistical Aggregate Report]

응답시간은 평균 1초 이내의 응답을 보이고 있음

- Master



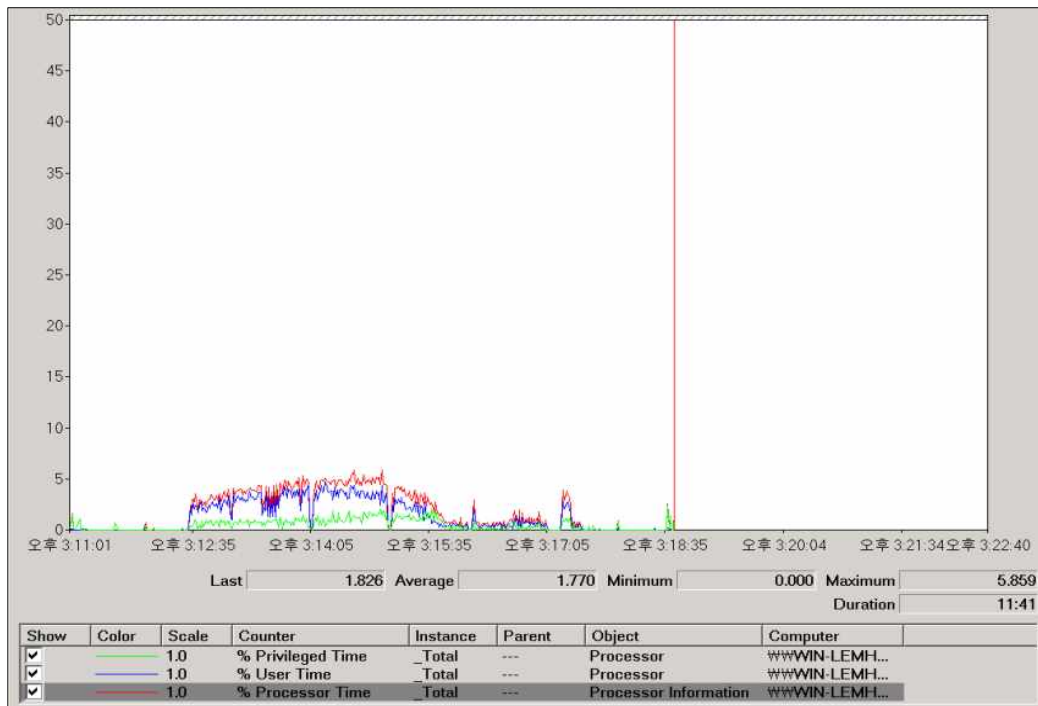
[그림 IV-15. CPU 자원사용률]



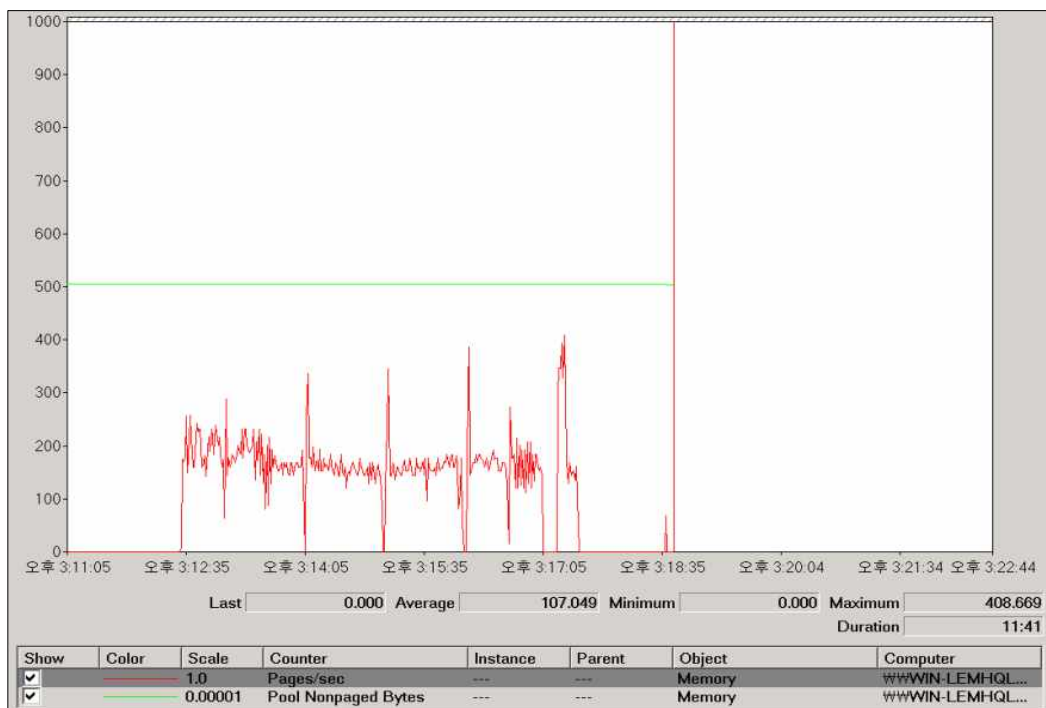
[그림 IV-16. Memory 자원사용률]

Slave Server로의 주기적인 데이터 복제로 인해 일부 구간에서 Memory 사용률이 급증하는 현상이 발생함

- Slave



[그림 IV-17. CPU 자원사용률]



[그림 IV-18. Memory 자원사용률]

Master Server로부터 주기적인 데이터 복제로 인해 일부 구간에서 Memory 사용률이 급증하는 현상이 발생함

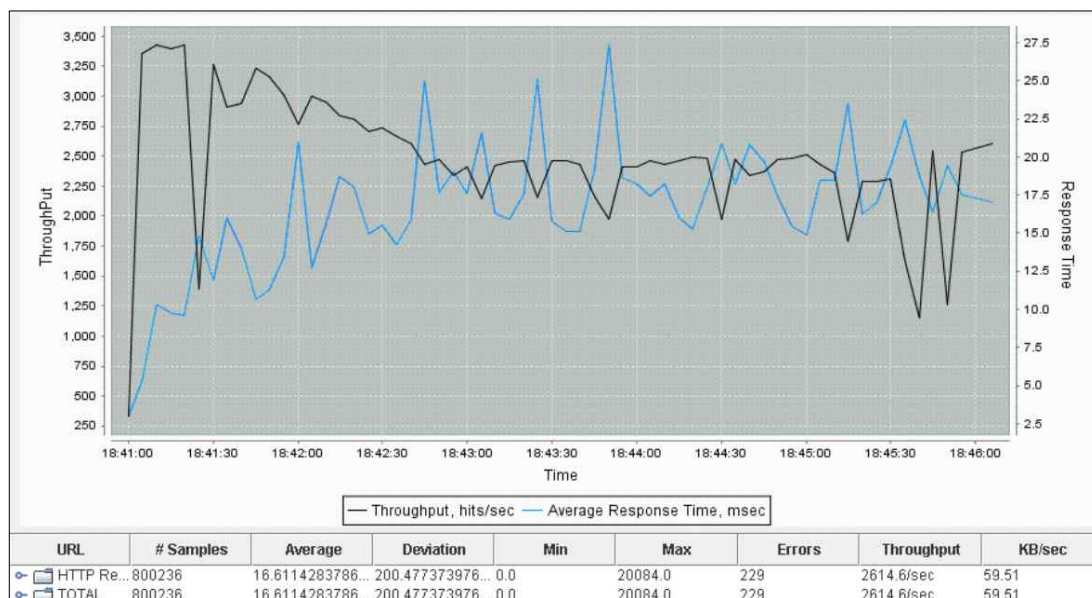
o Replication_Select

- 수행 정보

수행 조건	<ul style="list-style-type: none"> - Master Server와 Slave Server로 구성 - Ramp-Up : 5User - Running-Time : 5분 - 동시사용자 : 50User - 등록된 데이터 개수 : 500,000건 - 등록 된 키 값을 랜덤하게 생성한 후 데이터 조회
-------	---

- 측정결과(A Stack)

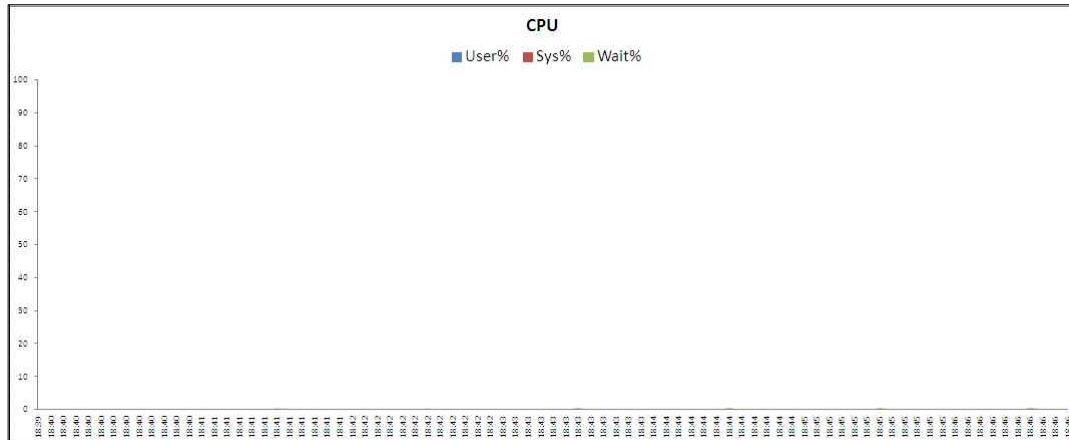
800,240건의 Sample을 수행하였으며, 0.03%의 Connect Error가 발생함



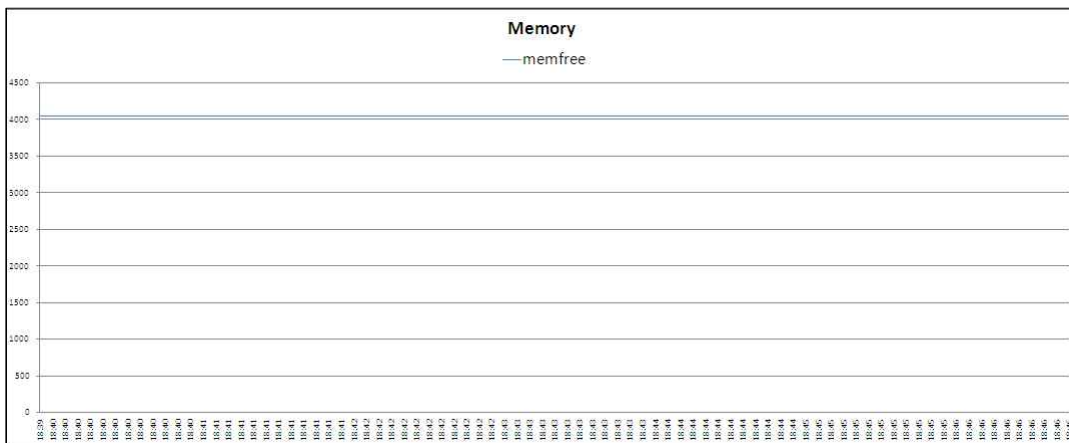
[그림 IV-19. Statistical Aggregate Report]

응답시간은 평균 1초 이내의 응답을 보이고 있음

- Master

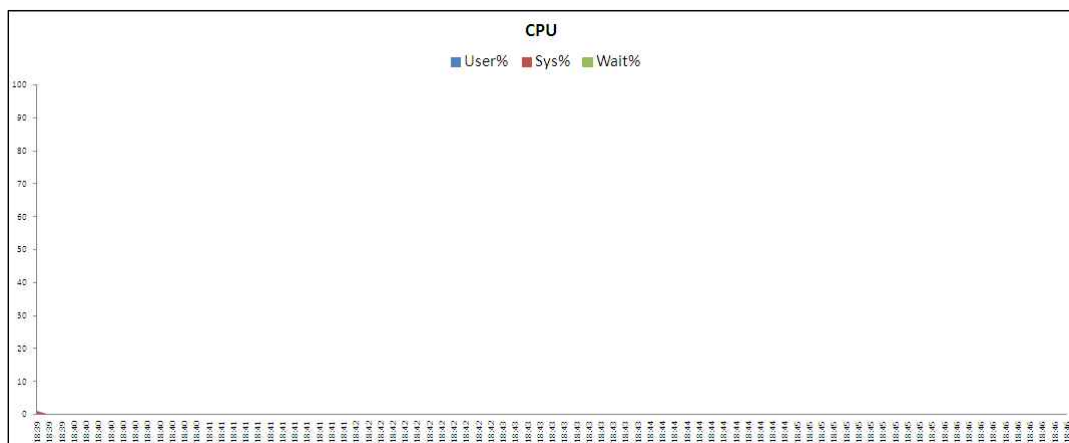


[그림 IV-20. CPU 자원사용률]

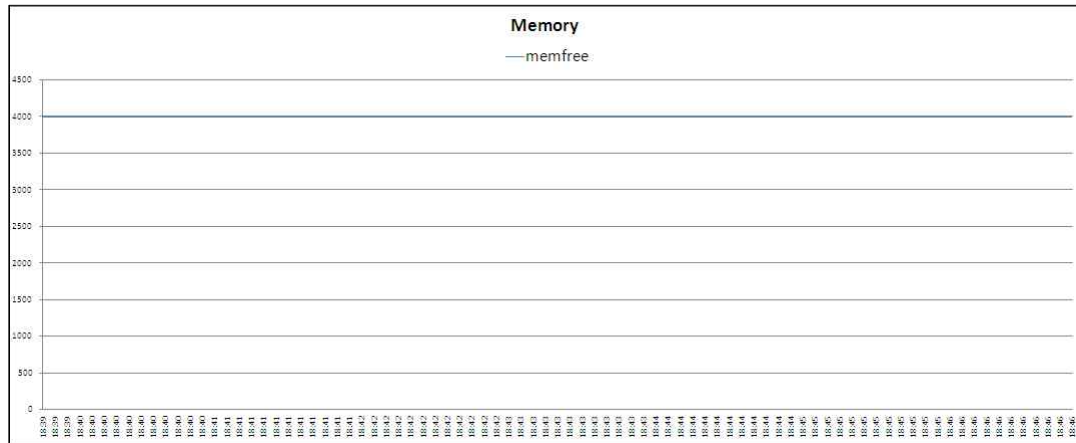


[그림 IV-21. Memory 자원사용률]

- Slave



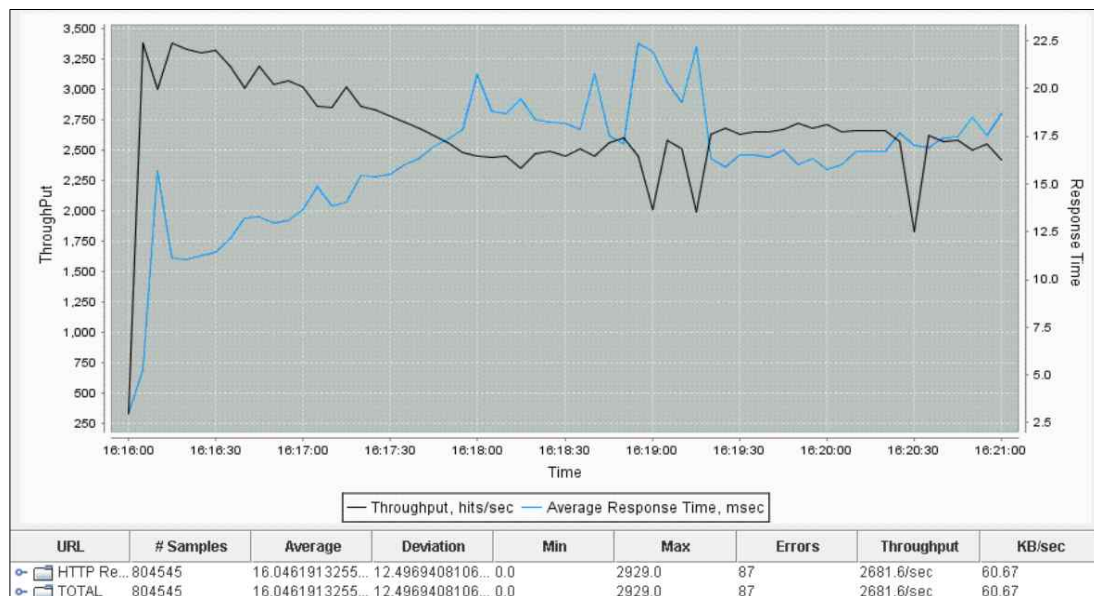
[그림 IV-22. CPU 자원사용률]



[그림 IV-23. Memory 자원사용률]

- 측정결과(B Stack)

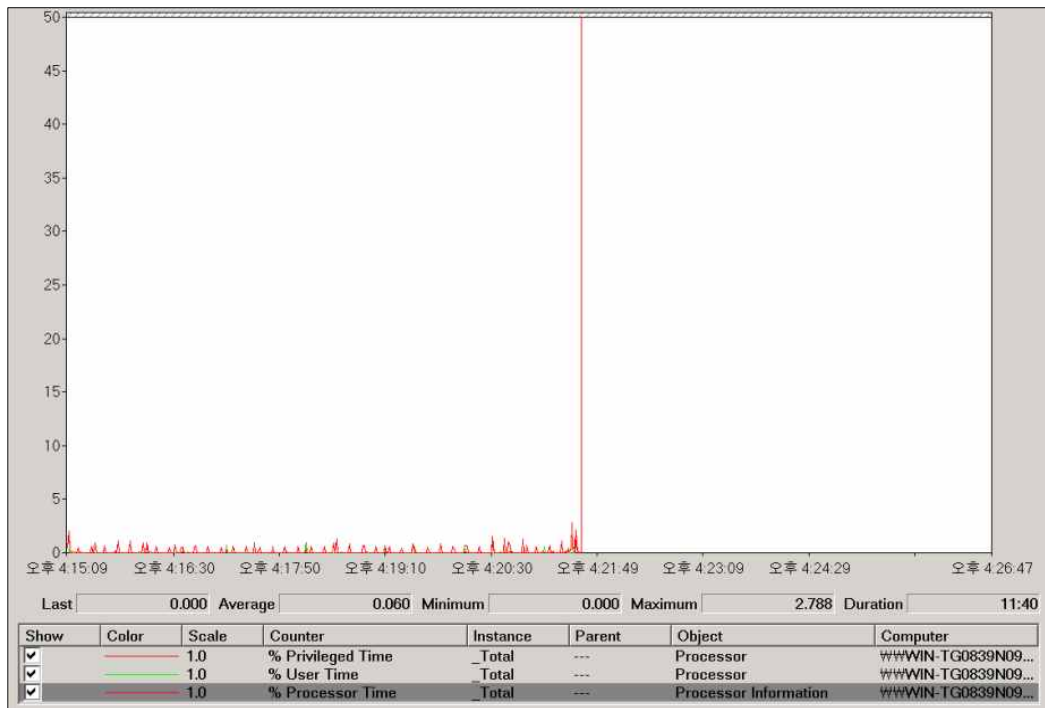
804,545건의 Sample을 수행하였으며, 0.01%의 Connect Error가 발생함



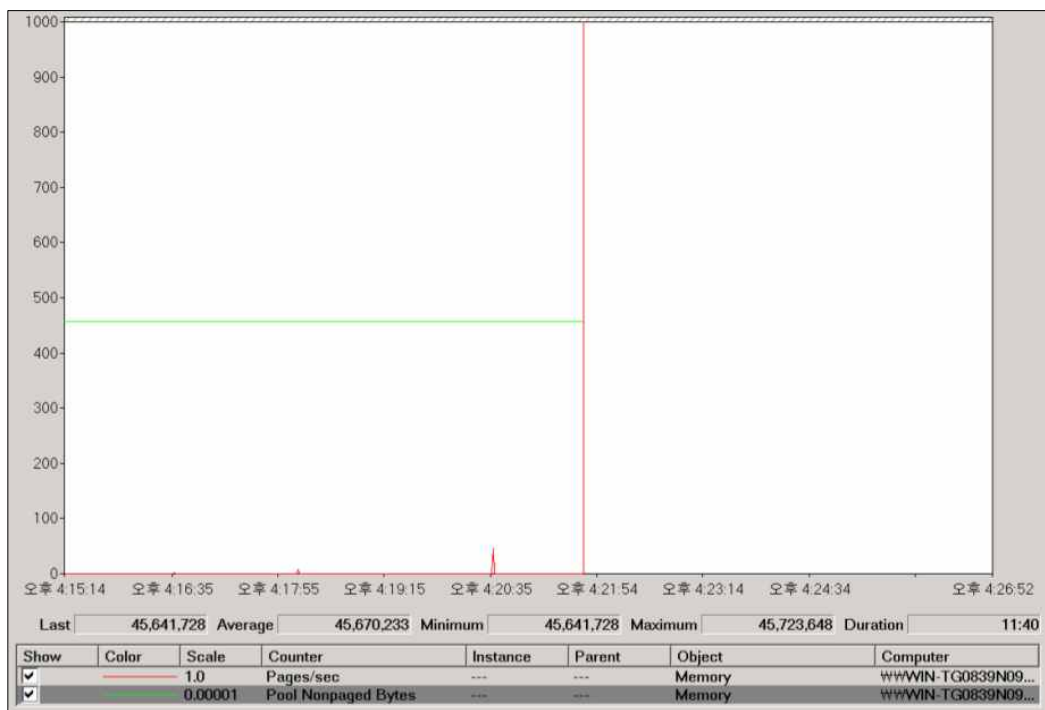
[그림 IV-24. Statistical Aggregate Report]

응답시간은 평균 1초 이내의 응답을 보이고 있음

- Master

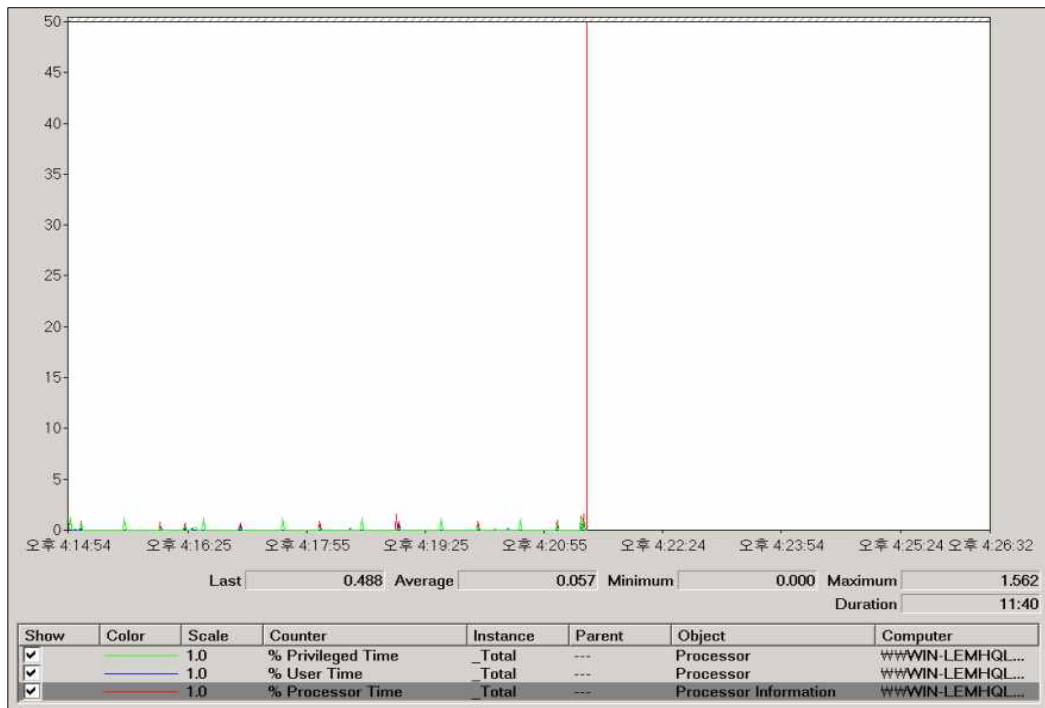


[그림 IV-25. CPU 자원사용률]

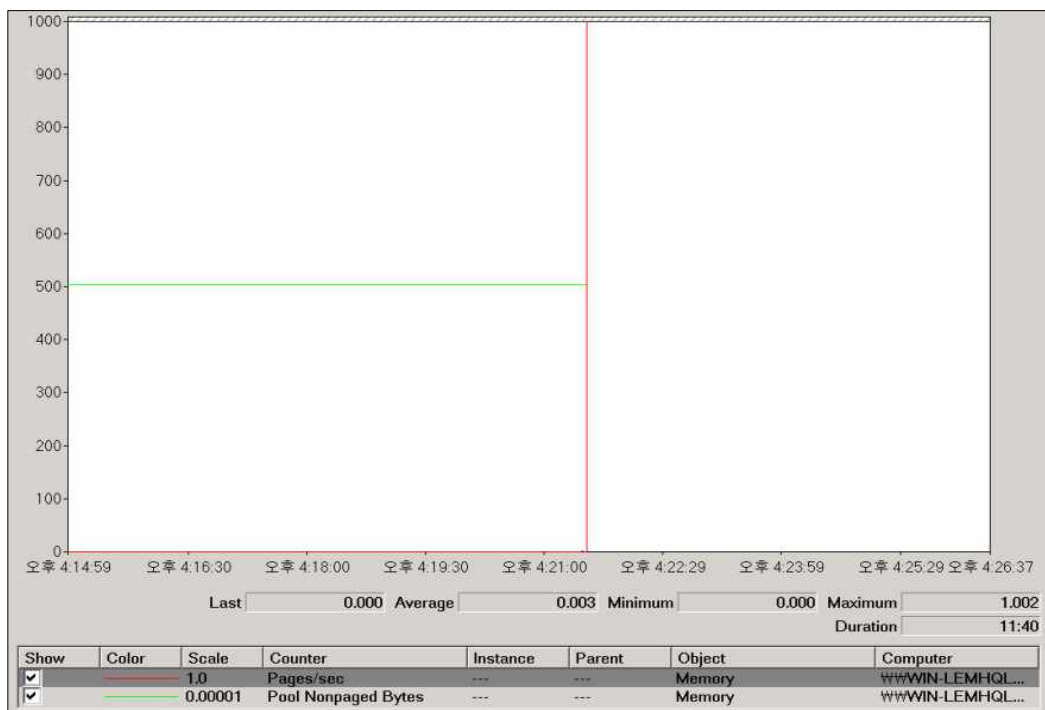


[그림 IV-26. Memory 자원사용률]

- Slave



[그림 IV-27. CPU 자원사용률]



[그림 IV-28. Memory 자원사용률]

□ 결함 내용

Jmeter⁷⁾를 이용한 Replication_Insert_Random, Replication_Select 시나리오 수행 결과 A Stack과 B Stack에서 동일한 Connect Error가 발생하였으며, Error의 내용은 아래와 같다.

o Jmeter Sampler result

- > Response code: Non HTTP response code: java.net.BindException
- > Response message: Non HTTP response message: Address already in use: connect

o Jmeter Response data

- > java.net.BindException: Address already in use: connect

o MongoDB Log

- > Socket recv() conn closed? php-host
- > SocketException: remote: php-host error: 9001 socket exception [0] server [php-host]
- > end connection php-host

7) Apache Jakarta 프로젝트의 일환으로 만들어진, 테스트 기능과 퍼포먼스를 측정하는 순수 자바로 작성된 공개SW

V. 종합

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

- MongoDB 성능 테스트 수행 결과 데이터 입력과 데이터 검색 시 Connect Error가 발생하였으며, 이는 소켓 제한으로 인한 Error로 추정된다. 이에 MongoDB 도입 시 사용 중인 시스템 환경에서 동시 접속자 수에 따른 Connect의 재활용이나, Open 가능한 Port의 개수에 대한 분석이 필요할 것으로 추정됨

※ 참고 자료

- [1] <http://www.ibm.com/developerworks/kr/library/os-mongodb4/index.html>
- [2] <http://cookbook.mongodb.org>
- [3] <http://www.mongodb.org>
- [4] <http://www.centos.org/>
- [5] <http://windows.microsoft.com/>
- [6] <http://www.apache.or.kr/>
- [7] <http://www.php.net>
- [8] <http://jmeter.apache.org/>
- [9] <http://en.wikipedia.org/wiki/Mongodb>