



Scouter 매뉴얼

Focus Area:

Product/Process:

Prepared By:

Document Owner(s)	Project/Organization Role

Project Status Report Version Control

Version	Date	Author	Change Description

Software Version

Version	Product
	Scouter Server
	Scouter client

1. 시작하기.....	4
1.1 애플리케이션 성능 관리	4
1.2 스카우터란?.....	4
1.3 스카우터 개요.....	4
1.4 지원 환경	7
2. Scouter 아키텍처.....	8
2.1 스카우터 기본 구조.....	8
2.2 Scouter Agent.....	8
2.3 Scouter Collector	9
3. Scouter Install.....	10
3.1 Server Installation	10
3.2 Window Preferences.....	14
4. Scouter 실시간 모니터링.....	16
4.1 실시간 대시보드	16
4.2 모니터링 인터페이스	17
4.3 XLog - 상세 트랜잭션 프로파일링.....	36
5. Scouter 추가 기능.....	49
5.1 클라이언트 스레드 리스트	49
5.2 클라이언트 환경 설정 정보.....	50
5.3 서비스 덤프.....	51
6. Scouter Alert.....	52
6.1 Alert 가능한 종류.....	52
6.2 Email	53
6.2.1 Scouter 서버 라이브러리 추가.....	53
6.2.2 scouter.conf 설정 파일의 Properties.....	53

6.3	Telegram	54
6.3.1	Scouter 서버 라이브러리 추가.....	54
6.3.2	Scouter.conf 설정 파일의 Properties	55
6.3.3	Telegram Application 설정	56

1. 시작하기

1.1 애플리케이션 성능 관리

- 애플리케이션 성능관리는 애플리케이션 서비스에 대한 효율적인 성능 모니터링 및 장애 대응 전략을 수립하고 사고 방지에 대해 지속적으로 성능 관리 체계를 구축 하는 것이다.
- APM은 application Performance Monitoring 또는 Application Performance Management 를 의미 한다.

1.2 스카우터란?

스카우터는 복잡한 Multi Tier 환경의 웹 기반 시스템에 대한 효율적인 성능 모니터링 체계를 마련함으로써 주어진 시스템 자원의 활용도를 극대화 시키는 오픈소스 APM이다.

- **웹 애플리케이션 관리 (사전 장애 예측)**

- WAS 구간의 모니터링을 활용한 WEB/DB의 명확한 원인 규명
- 구체적인 시스템의 개선 방향 지시
- 고객 서비스에 대한 품질 향상 제공

- **비즈니스 트랜잭션 관리**

- 요청에 대한 서비스 중심의 성능 관리 제공
- 트랜잭션 응답시간 분포도(XLog)를 통한 관리
- 트렌드 분석을 통한 관리 및 리포팅

- **오픈소스 모니터링 및 확장팩**

- 표준화된 운영관리, 최적의 모니터링 구현 방안 제시
- 업무 특성에 적합한 최적화된 분석, 점검 방안 제시
- 오픈소스 인프라 확장 추진 로드맵 제시

1.3 스카우터 개요

Web Application Server(WAS)에서 동작하는 Java Application의 성능 관리를 위한 오픈소스 프로젝트 APM 솔루션이다.

1.3.1 스카우터 서비스 모니터링

서비스 모니터링은 애플리케이션의 사용자와 실시간 서비스 처리 현황과 관련된 성능 데이터를 수집하는 것이다.

- 실시간 접속 중인 동시단말 사용자
- 실시간 수행 중인 액티브 서비스
- 실시간 업무 처리량
- 실시간 트랜잭션 응답 시간 분포도
- 애플리케이션 이름 동적 치환 기능
- 외부 트랜잭션과 바인딩 변수를 포함한 SQL 추적
- HTTP POST 호출 파라미터 키와 값 추적

1.3.2 스카우터 리소스 모니터링

리소스 모니터링은 애플리케이션이 서비스를 수행하는 과정에서 사용하는 CPU와 메모리와 같은 논리적 혹은 물리적 자원과 관련된 성능 데이터를 수집하는 것이다.

- DB 커넥션 및 커넥션 풀 상태
- 시스템 및 프로세스 CPU 사용률
- 시스템 및 프로세스 메모리 사용량
- 프로세스 힙 메모리 사용량
- 파일 IO 상태
- TCP 소켓 IO 추적
- 콜렉션 객체와 자바 라이브 오브젝트
- JMX(Java Management Extension)를 활용한 WAS 내부 리소스 상태

1.3.3 장애 진단 및 분석

웹 애플리케이션의 성능과 관련된 장애 유형은 부하량 증가에 의한 상대적 성능 장애와 비정상적인 특정 조건 상황에서 발생하는 조건적 성능 장애로 구분된다. 이런 성능 장애를 진단하고 분석하기 위한 기능을 제공한다.

1.3.4 실시간 액티브 서비스 모니터링

자바 애플리케이션이 현재 처리중인 트랜잭션을 액티브 서비스라고 한다. 그리고 이 액티브 서비스 목록을 경과 시간대 별로 나누어서 실시간 이퀄라이저 차트와 액티브 서비스 목록으로 제공한다. 이를 통해 사용자는 현 시점에서의 자바 애플리케이션 내부의 액티브 서비스에 대한 스냅 샷 정보를 확인 할 수 있다.

1.3.5 XLog와 트랜잭션 프로파일링

응답 시간 분포도는 모든 트랜잭션을 트랜잭션의 시작 시간과 응답 시간을 기준으로 차트에 점으로 표현한 것이다. 스카우터에서는 XLog라고 표현한다.

- 임의의 메소드 호출 다이내믹 프로파일링
- 메소드 파라미터와 반환 값 추적
- DB와 바인딩 변수를 포함한 SQL 추적
- 레거시 시스템과의 연동 같은 외부 트랜잭션 추적
- 파일과 소켓 IO추적

1.3.6 다이내믹 프로파일링

자바 애플리케이션을 재기동 없이 임의의 자바 메소드에 대해서 트랜잭션 프로파일링을 활성화 혹은 비활성화 할 수 있는 다이내믹 프로파일링 기능을 제공한다.

1.3.7 다이내믹 스택트레이스

자바 애플리케이션에서 스택트레이스를 추출하는 전통적인 방법은 임의의 예외가 발생하도록 소스 코드를 수정하고 의도적으로 해당 예외를 발생시켜서 관련 스택트레이스를 명시적으로 출력하는 것이다. 자바 애플리케이션을 재시작하지 않은 상태에서 소스코드의 수정 없이 스택트레이스를 수집할 수 있다. 스택트레이스를 기록할 메소드를 지정하고 임의의 트랜잭션이 해당 메소드에 진입하면 관련 스택트레이스가 다이내믹하게 기록된다.

1.3.8 도메인 구성을 통한 통합 모니터링

하나의 스카우터 서버가 감당할 수 있는 스카우터 에이전트의 숫자와 업무 처리량에는 한계가 있다. 이 한계를 초과하는 경우 여러 개 서버를 운영하면서 부하를 분산 시켜야 한다. 그러나 여러 개의 서버를 운영하려면 관리 비용은 증가하고 사용자 편의성은 감소된다. 이런 단점을 해결하기 위해서 스카우터에서는 도메인 구성을 통해 여러 개의 서버를 통합해서 관리하는 통합 사용자 인터페이스를 제공한다.

1.3.9 애플리케이션과 SQL 튜닝

애플리케이션의 호출 건수와 평균 응답 시간 통계를 기반으로 성능 저하를 야기하는 병목 애플리케이션을 도출하고 프로파일링 기능을 통해서 해당 애플리케이션을 수행한 트랜잭션의 병목 구간을 구체적으로 파악할 수 있다.

1.3.10 경보와 예외 모니터링

자바 애플리케이션에서 발생하는 예외를 감지하고 자바 애플리케이션 상태를 분석하여 적절한 경보를 발생 시킨다. 경보는 심각, 에러, 경고 등으로 구분되는데, 사용자는 클라이언트를 통해서 실시간으로 경보 발생을 확인하고 과거 경보 내역을 분석 할 수 있다.

1.3.11 모니터링 확장 어댑터

모니터링 확장 어댑터를 통해서 미리 정의된 규약과 형식 없는 비정형 성능 데이터를 수집할 수 있다. 모니터링 확장 어댑터는 다음과 같다.

1.3.12 모니터링 확장 어댑터

수집한 성능 데이터를 데이터베이스에 저장하기 때문에 유연한 통계 분석을 가능하게 한다. 그리고 해당 데이터를 2차 가공한 다양한 사용자 정의형 보고서 작성 기능을 제공한다.

1.4 지원 환경

IT 환경 내에서 운영중인 대부분의 OS(Operating System), Java(Sun, IBM, BEA, etc), WAS(Web Application Server)를 지원하고 있다.

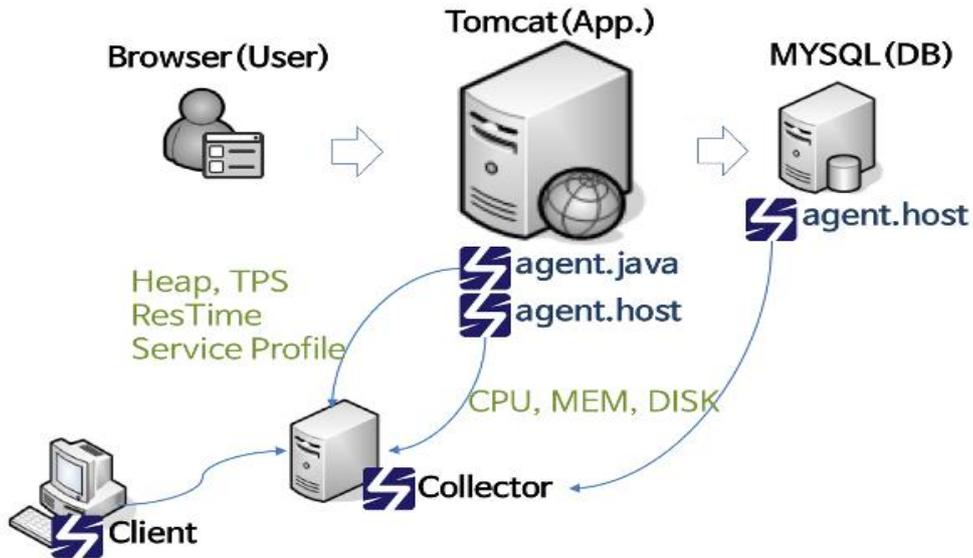
지원 운영 체제(Operating System)	웹 애플리케이션 서버(WAS)
● AMD64 Free BSD	● Apache Tomcat 6.X, 7.x, 8.x
● AMD64 Linux	● Oracle WebLogic 8.x, 9.x, 10.x, 11.x, 12.x
● HP-UX 11.x 32bit, 64bit, Itanium 64bit	● IBM WebSphere 6., 7.x
● PPC 64 Linux, AIX	● Tmaxsoft JEUS 5.x, 6.x
● Sun Solaris 32bit, 64bit, x86	● Caucho Technology Resin 4.x, 5.x
● Universal MacOSX 32bit, 64bit	● JBoss EAP 5.x,6.x
● Compaq Tru54 UNIX OSF1	● WildFly 7.x, 8.x
● Microsoft Windows 2000, XP, 2003, Vista	

*신규로 추가되는 시스템의 경우 github로 문의 시 해당 플랫폼에 대한 테스트 후 지원 대상에 포함 시킴

2. Scouter 아키텍처

2.1 스카우터 기본 구조

스카우터는 에이전트/컬렉터 구조를 가지고 있으며, 별도의 클라이언트 애플리케이션을 통해 시스템에 대한 모니터링을 수행한다.



- ✓ Server(Collector): Agent가 전송한 데이터 수집/처리
- ✓ Host Agent: OS의 CPU, Memory, Disk등의 성능 정보 전송
- ✓ Java Agent: 실시간 서비스 성능 정보, Heap Memory, Thread 등 Java 성능 정보

2.2 Scouter Agent

모니터링 대상 시스템에 구성되며, 각종 성능 정보를 수집하여 스카우터 서버로 전달하는 역할을 한다.

- Host Agent: 모니터링 대상 Host에 대한 CPU, Memory, Disk 등등 대한 시스템 성능 정보를 Collector 서버에 전달하는 역할을 담당한다.

```
${SCOUTER_HOME}/agent.host
```

파일	설명
host.sh	Host 에이전트 기동을 위한 스크립트

stop.sh	Host 에이전트 종료용 위한 스크립트
lib	Host 에이전트에서 사용하는 라이브러리 위치
conf	스카우터 host 에이전트 설정을 위한 설정파일 위치

- Java Agent: 모니터링 대상 Java 애플리케이션과 별도로 동작하며 Java 애플리케이션 hooking 방식으로 Java 애플리케이션의 응답시간, TPS, GC, JVM 상태정보 등등을 Collector 서버에 전달하는 역할을 담당한다.

```

${SCOUTER_HOME}/agent.java

```

파일	설명
conf	스카우터 host 에이전트 설정을 위한 설정파일 위치

2.3 Scouter Collector

에이전트로부터 실시간 모니터링 정보를 수집, 가공하여 실시간 대시보드 제공 및 통계정보 관리, 장애/에러 경보 등 관리에 필요한 각종 기능을 수행한다.

- Collector Server: 각 Agent에서 받은 데이터를 저장하고 Scouter Client 화면상으로 모니터링 할 수 있게 대시 보드에 제공한다.

```

${SCOUTER_HOME}/server

```

파일	설명
startup.sh	Collector 서버 기동을 위한 스크립트
stop.sh	Collector 서버 종료용 위한 스크립트
database	서버에서 가공된 데이터를 File System 형태로 저장하기 위한 위치
lib	Collector 서버에서 사용하는 라이브러리 위치
conf	스카우터 서버 설정을 위한 설정파일 위치

- 서버에서 가공된 데이터는 데이터 형태에 따라 DB File System에 날짜별로 저장되며 이러한 데이터를 통해 시스템 분석에 필요한 각종 데이터를 확보할 수 있다.

```

${SCOUTER_HOME}/server/database

```

3. Scouter Install

3.1 Server Installation

3.1.1 Collector Server Installation

- ✓ JDK 1.6+ (1.8+ recommended)

3.1.1.1 Download Collector Server

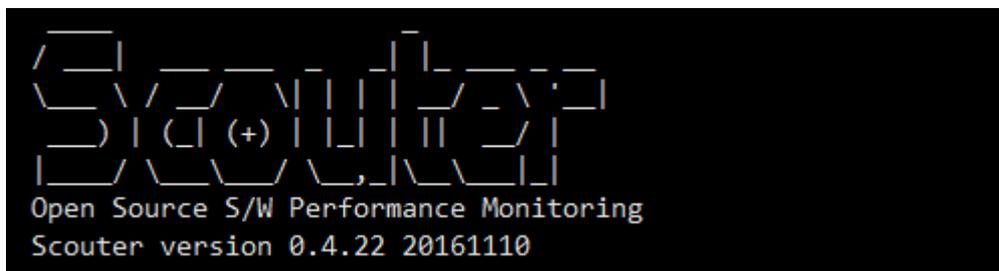
<https://github.com/scouter-project/scouter/releases>

3.1.1.2 Extract the file (./scouter/server)

3.1.1.3 Start script 실행

ex)

```
$ cd ${SCOUTER_HOME}/server
$ ./startup.sh
```



3.1.1.4 Collector server 기본 사용 포트

- ✓ UDP Receive Port: 6100 (성능 metric을 얻어 오기 위한 포트)
- ✓ TCP Service Port: 6100 (클라이언트와 에이전트가 통신하기 위한 포트로서 사용되며 에이전트로 명령을 요청하고 응답을 받는 역할)

3.1.1.5 Configuration

- ✓ scouter.conf file

스카우터 설정 관련된 포트, 데이터베이스, 로그에 관련된 설정은 scouter.conf 파일을 통해서 설정을 변경할 수 있다.

ex)

```
$ vi ${SCOUTER_HOME}/server/conf/scouter.conf
```

```
# Agent Control and Service Port(Default : TCP 6100)
net_tcp_listen_port=6100
# UDP Receive Port(Default : 6100)
net_udp_listen_port=6100
# DB directory(Default : ./database)
db_dir=./database
# Log directory(Default : ./logs)
log_dir=./logs
```

3.1.2 Host Agent

- ✓ JDK 1.5+(Required), 1.6(Recommended)

3.1.2.1 Download Host agent

<https://github.com/scouter-project/scouter/releases>

3.1.2.2 Host Agent 기동을 위한 스크립트 실행

ex)

```
$ cd ${SCOUTER_HOME}/agent.host
$ ./host.sh
```

3.1.3 Java Agent

- ✓ JDK 1.5+(Required), 1.6(Recommended)

3.1.3.1 Java agent JVM 옵션 추가

WAS 기동 Start 쉘 스크립트 부분에 아래와 같이 Option을 추가

ex) Java Agent의 startup.sh

```
export JAVA_OPT="$JAVA_OPT -javaagent:${SCOUTER_HOME}/agent.java/scouter.agent.jar
export JAVA_OPT="$JAVA_OPT -Dscouter.config=${SCOUTER_HOME}/agent.java/conf/scouter_java.conf
export JAVA_OPT="$JAVA_OPT -Dobj_name=jboss_standalone_ha_11
```

설 정	설 명
-javaagent	Byte Code Instrumentation 기법으로 실제 실행 환경의 동작 모니터링을 위한 agent library loading 위한 설정
-Dscouter.config	Agent.java 의 설정 파일 위치를 설정
-Dobj_name	Agent name 설정

3.1.3.2 Java Agent 기본 설정 내용

ex)

```
$ cd ${SCOUTER_HOME}/agent.java
$ vi scouter_java.conf
```

```
# Scouter Server IP Address (Default : 127.0.0.1)
net_collector_ip=127.0.0.1
# Scouter Server Port (Default : 6100)
net_collector_udp_port=6100
net_collector_tcp_port=6100
# Scouter Name(Default : tomcat1)
obj_name=jboss_standalone_ha_11
```

설 정	설 명
net_collector_ip	Data 수집 서버(collector server) IP address
net_collector_udp_port	Data 수집 서버(collector server) TCP Port
net_collector_tcp_port	Data 수집 서버(collector server) UDP Port
obj_name	Agent name 설정

3.1.4 Client Install

✓ JRE 1.8+

3.1.4.1 Download Client

<https://github.com/scouter-project/scouter/releases>

3.1.4.2 Scouter.client.product-{OS버전}.zip 파일 압축 해제

3.1.4.3 Scouter.exe 실행 후 해당 서버 IP:Port 입력

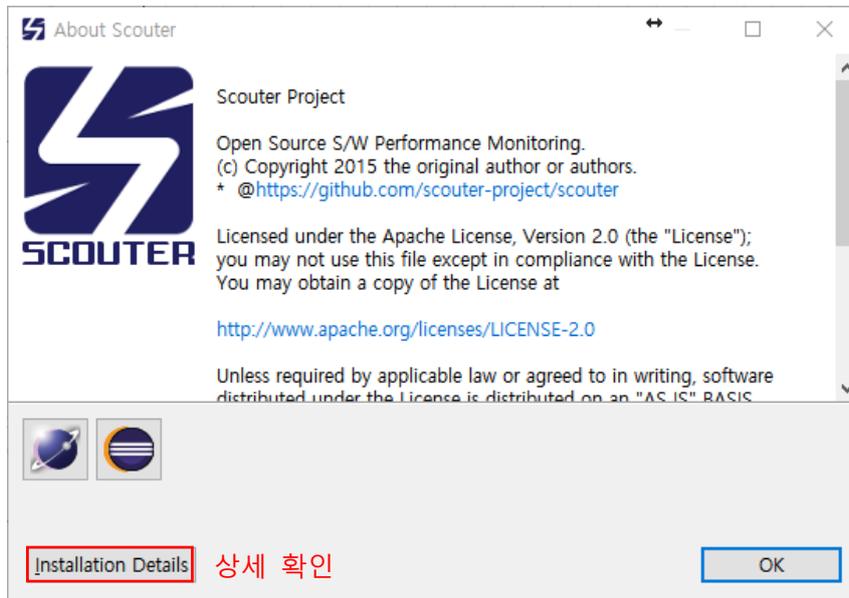
ex)



Default id : admin / admin

3.1.5 About Scouter

Scouter 프로젝트에 대한 정보를 메뉴 바에 About을 통해 확인이 가능하다.



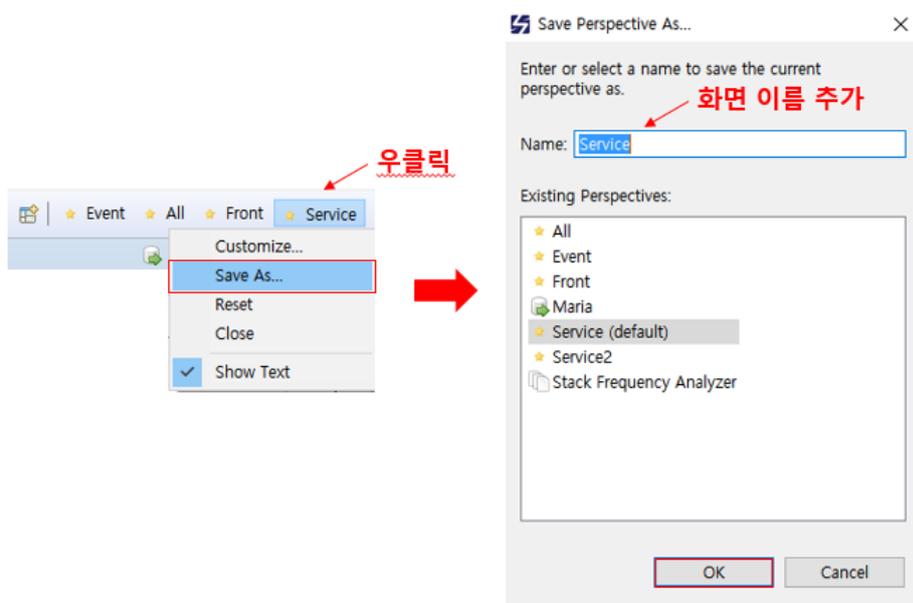
3.2 Window Preferences

3.2.1 Add Perspective



우측 상단에 위와 같은 업무/용도별 Perspective 지정 기능

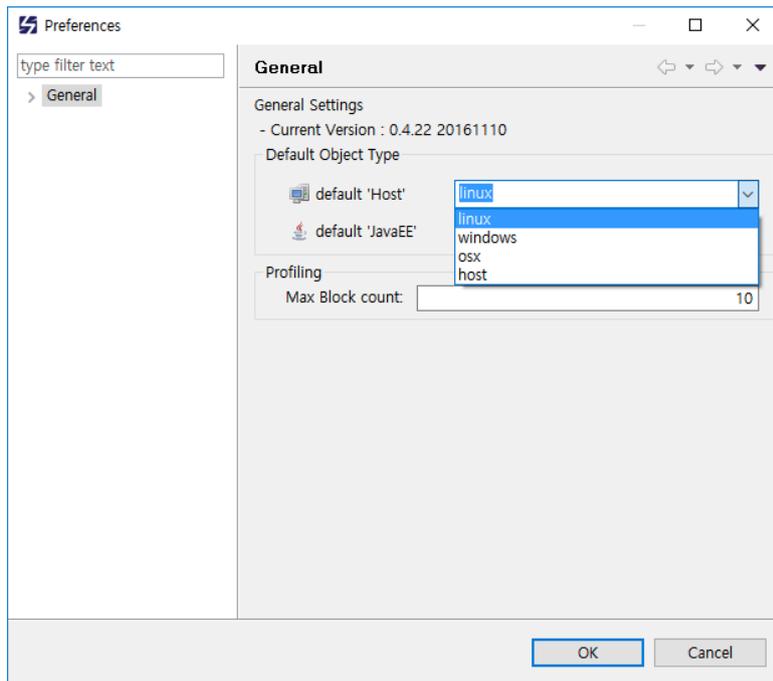
ex)



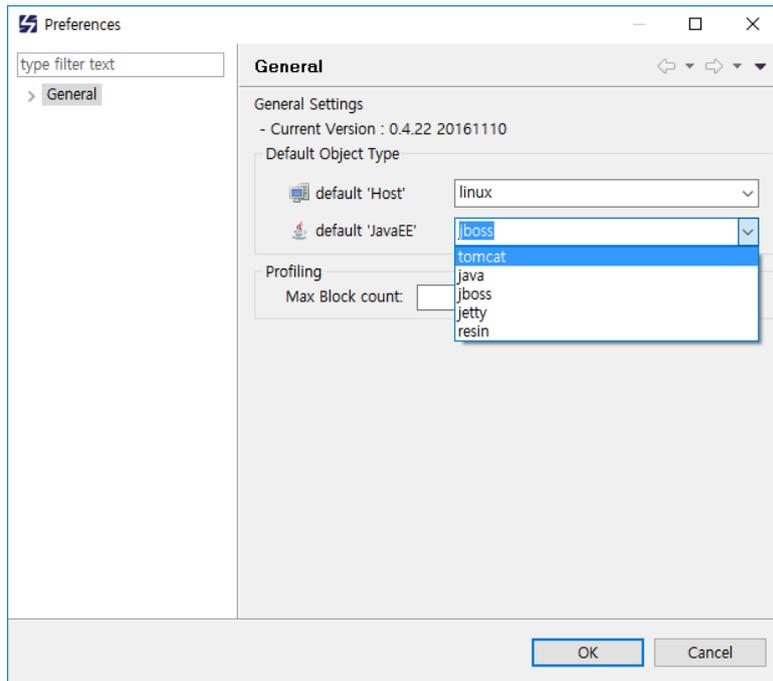
3.2.2 Preferences

최초 실행 시 Linux / Tomcat이 아닌 경우 그래프 등의 항목이 표시되지 않는 경우가 있으며, Preference를 통해 각각의 타입을 변환하여 정상적으로 표시될 수 있다. 해당 host와 java에 대해서 확인 후 타입을 변경

Default 'Host' 설정 (linux, windows, osx, host)



Default 'JavaEE' (tomcat, java, jboss, jetty, resin)

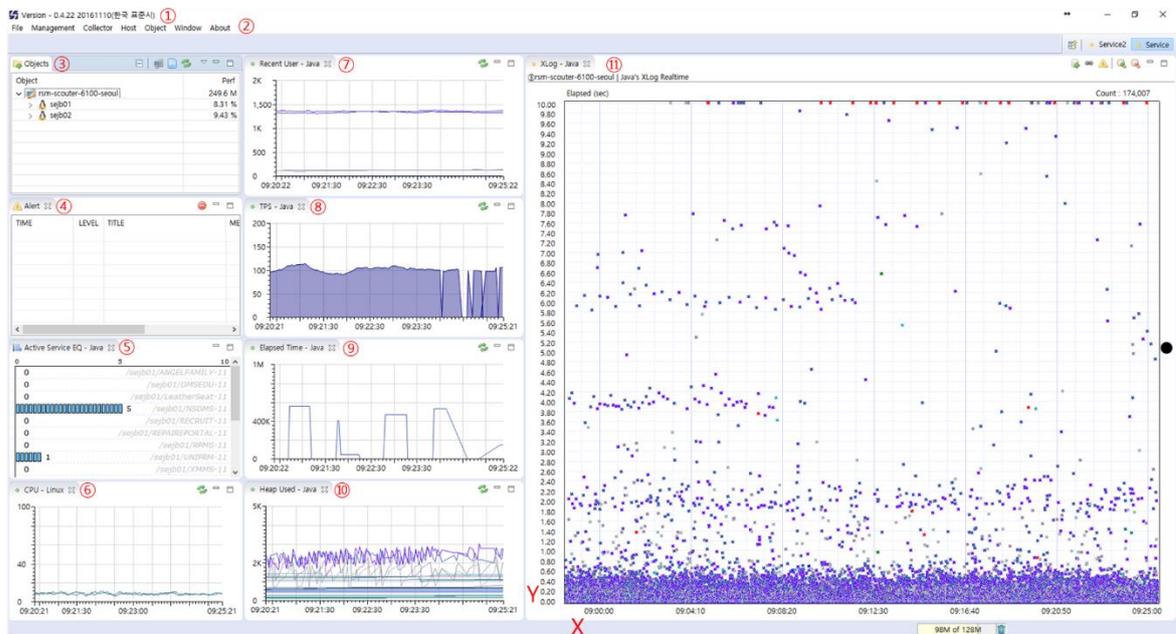


4. Scouter 실시간 모니터링

4.1 실시간 대시보드

J2EE 시스템 운영 상태 모니터링을 위해 필요한 각종 데이터를 서로 유기적으로 배치하여 제공하고 있으며, 유기적으로 배치된 시스템 상태에 대한 빠른 판단 및 효과적으로 제어할 수 있도록 돕고 있다.

시스템 사용자 정보	<ul style="list-style-type: none"> • 통신 단말 사용자 수 • 방문자 수 • 액티브 사용자 수
시스템 처리 상태	<ul style="list-style-type: none"> • 초당 처리 건수 • 시간당 호출 건수 • 평균 응답시간
지원상태	<ul style="list-style-type: none"> • 시스템 CPU 사용량 • JVM CPU 사용량 • 힙 메모리 사용량
장애/에러	<ul style="list-style-type: none"> • 시스템 장애 • 프로그램 에러



<설치 후 기본 화면>

① Version: 스카우터 클라이언트 버전 정보

Version - 0.4.22 20161110(한국 표준시)

② 스카우터 설정 메뉴

File Management Collector Host Object Window About

- ③ Object: Host, agent들의 목록
- ④ Alert: Alert를 통해서 사용자가 원하는 알람을 화면에서 확인이 가능
- ⑤ Active Service EQ: 액티브 서비스 모니터링을 이퀄라이저 형태로 보여짐
- ⑥ CPU – Linux: Host에 대한 CPU 사용률
- ⑦ Recent User – Java: 최근 사용자 카운트
- ⑧ TPS – Java 초당 수행 건수
- ⑨ Elapsed Time – Java 수행 시간
- ⑩ Heap Used – Java: Heap 메모리 사용률
- ⑪ XLog – Java: 종료된 요청을 점 형태로 보여주며, 드래그 하면 상세 프로파일을 확인 가능

-세로축(Y축): 트랜잭션의 수행 시간을 의미, 위쪽방향키를 누르면 Y축 구간 크기를 증가하고, 아래쪽방향키를 누르면 Y축 구간크기가 감소

-가로축(X축): 트랜잭션이 수행 된 시간을 의미, 오른쪽 방향키를 누르면 X축 구간 크기가 1분 증가하고, 왼쪽 방향키를 누르면 X축 구간 크기가 감소

(자세한 내용은 모니터링 인터페이스에서 다뤄진다.)

4.2 모니터링 인터페이스

4.2.1 공통 인터페이스

4.2.1.1 스카우터에서는 공통으로 우측 상단에 새로 고침, 최소화, 최대화 기능을 제공한다.



4.2.1.2 Frame 추가, 변경 방법

- ✓ 마우스 클릭 상태에서 원하는 위치로 이동 후 마우스 오버 하면 위할 영역을 확인 할 수 있다.

ex)

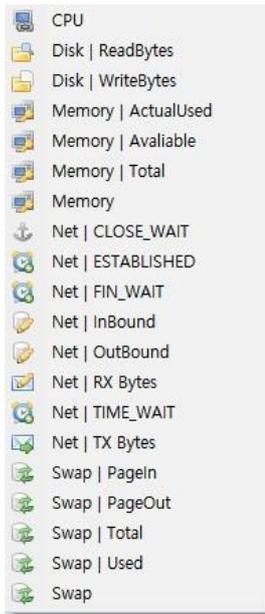


마우스 왼쪽 버튼을 누르고 있는 상태에서 화면을 나누기, 붙이기가 가능하고, 데스크톱에서 제외 하고 싶을 경우 해당 탭에 X를 클릭한다.

4.2.1.3 시스템 모니터링(OS)

Host object 클릭 후 우클릭 또는 상단 메뉴에서 Host -> Performance Counter

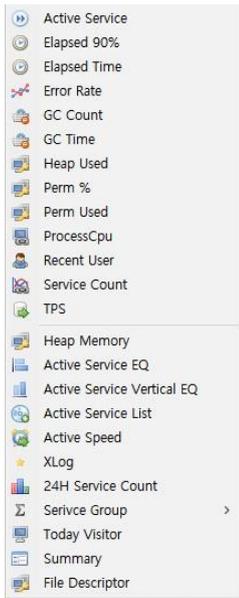
Object	Perf
192.168.56.105:6100	
52.78.104.114:16100	
centvm7	30.7 M
centvm7	1.51 %
jboss_standalone_ha_11	0 cnt



성능 지표	단위	설명
CPU	%	CPU 사용률
Memory ActualUsed	MB	실제 사용 메모리
Memory Available	MB	사용 가능 메모리
Memory Total	MB	전체 메모리
Memory	%	Host 별 메모리 사용률
Net CLOSE_WAIT	Cnt	CLOSE 대기
Net ESTABLISHED	Cnt	ESTABLISHED 된 Connection 수
Net FIN_WAIT	Cnt	FIN_WAIT 상태 Connection 수
Net InBound	Cnt	InBound 패킷 수
Net OutBound	Cnt	OutBound 패킷 수
Net TIME_WAIT	cnt	TIME_WAIT 상태의 Connection 수
Swap PageIn	Page	Swap 파일에 write한 Page 수
Swap PageOut	Page	Swap 파일에 read한 Page 수
Swap Total	MB	Swap 메모리 사이즈
Swap Used	MB	Swap 메모리 사용량
Swap	%	Host 별 Swap 메모리 사용률

4.2.1.4 액티브 서비스 모니터링(Java)

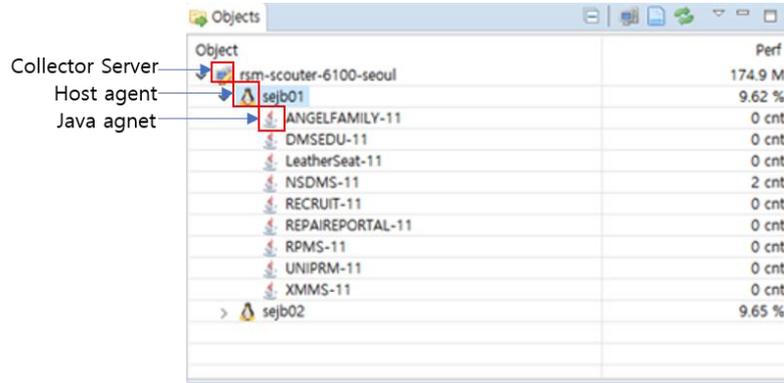
Java object 클릭 후 우클릭 또는 상단 메뉴에서 Collector -> Performance Counter



성능 지표	단위	설명
Active Service	Cnt	현재 처리중인 서비스 개수
Elapsed 90%	Ms	상위 90% 서비스의 평균 응답시간
Elapsed Time	ms	평균 응답시간
Error Rate	%	에러가 발생한 서비스 비율
GC Count	Cnt	Garbage Collection 횟수
GC Time	ms	GC 수행에 소요된 시간
Heap Used	Ms	Heap 메모리 사용량
Perm %	%	Permanent Generation 사용 비율
Perm Used	Mb	Permanent Generation 사용량
ProcessCpu	%	JVM 프로세스 CPU 사용률
Recent User	cnt	최근 5분 동안 방문한 Unique 사용자 수
Service Count	cnt/min	서비스 호출 건수
TPS	Tps	초당 트랜잭션 처리 건수
Heap Memory	MB	Heap 메모리 사용률 통계
Active Service EQ		실행 중인 서비스 이퀄라이저
Active Service Vertical EQ		실행 중인 서비스 가상 이퀄라이저
Active Service List		현재 실행중인 서비스 목록
Active Speed		현재 실행중인 서비스 속도
XLog	sec	트랜잭션 수행 지표
24H Service Count	cnt/min	24시간 서비스 호출 건수
Today Visitor		금일 방문자 수
Summary		요약정보

4.2.2 Object

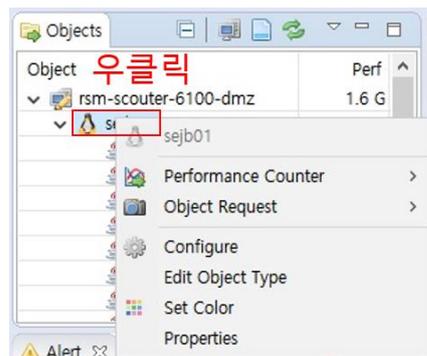
- Host 별 Process 모니터링을 제공한다.



- 항목 설명

상단 메뉴	설 명
Collapse All	모든 리스트를 축소 버튼
Server Manager	Server를 추가 또는 import하는 버튼
Filter object	모니터링 하고 자하는 필터 버튼

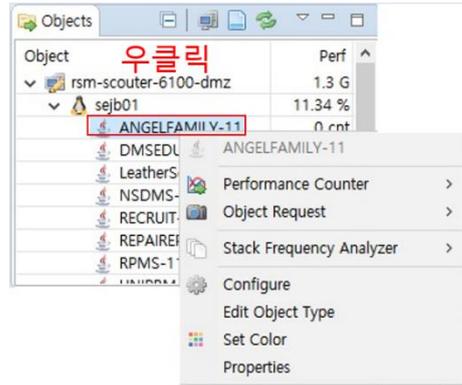
- Host 우클릭 메뉴



항 목	설 명
Performance Counter	CPU/Memory/Network/Disk에 대한 수집된 정보를 모니터링
Object Request	Env/Top/Disk Usage에 대한 수집된 정보를 모니터링
Configure	모니터링 설정을 변경 할 수 있음

Edit Object Type	Object 관련된 정보를 수정
Set Color	색을 변경
Properties	현재 설정된 Properties를 확인

- Java 프로세스 우클릭 메뉴



항 목	설 명
Performance Counter	CPU/Memory/Network/Disk에 대한 수집된 정보를 모니터링
Object Request	Env/Top/Disk Usage에 대한 수집된 정보를 모니터링
Stack Frequency Analyzer	
Configure	모니터링 설정을 변경 할 수 있음
Edit Object Type	Object 관련된 정보를 수정
Set Color	색을 변경
Properties	현재 설정된 Properties를 확인

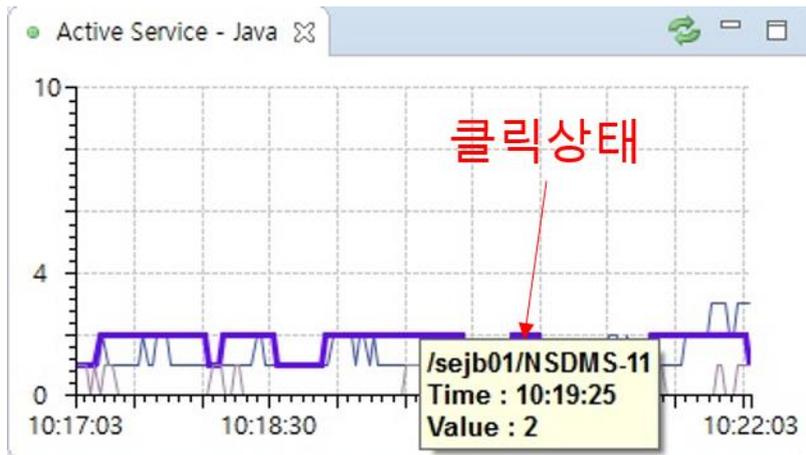
4.2.3 Alert

모니터링 시 설정한 임계치에 대한 알람 발생시 이를 감지하여 해당 이벤트를 슬랙, 텔레그램, 이메일 등을 통해 관리자에게 통지함으로써 빠른 장애에 대한 대비를 수행 하도록 도와줍니다. 또한 필요 시 임의의 사용자 정의형 "경고"를 확장 어댑터를 통해 추가 등록할 수 있다. (6. Scouter Alert 설정 내용 추가)

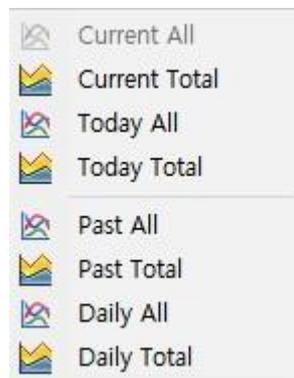
TIME	LEVEL	TITLE	ME

4.2.4 Active Service

Java 애플리케이션이 현재 처리중인 트랜잭션을 액티브 서비스라고 한다. 이 화면을 통해 사용자는 현 시점에서의 Java 애플리케이션 내부의 액티브 서비스에 대한 스냅 샷 정보를 확인할 수 있다.



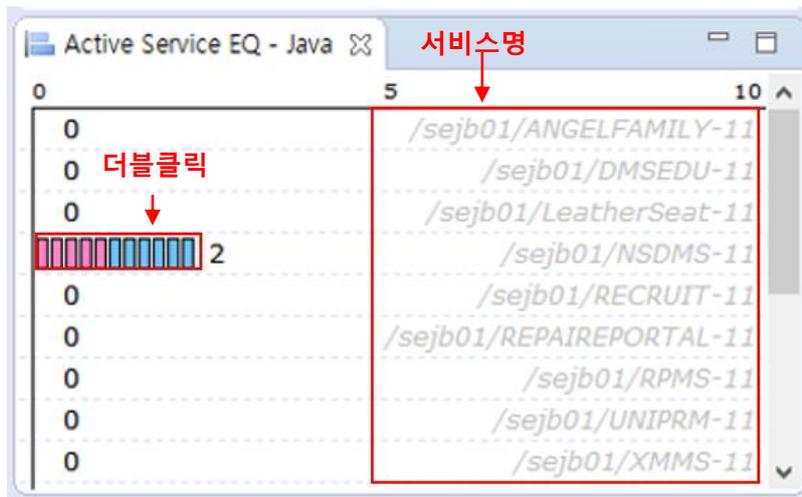
- 우클릭 메뉴



항 목	설 명
Current All	현재 모든 액티브 서비스
Current Total	현재 액티브 서비스 통계
Today All	금일 액티브 서비스 모니터링
Today Total	금일 액티브 서비스 통계
Past All	과거 액티브 서비스
Past Total	과거 액티브 서비스 통계
Daily All	날짜 별 모든 액티브 서비스
Daily Total	날짜 별 액티브 서비스 통계

4.2.5 Active Service EQ

액티브 서비스의 개수 및 서비스 경과 시간을 색깔로 구분하여 서비스 정상 여부를 직관적으로 판단할 수 있도록 실시간 이퀄라이저 그래프를 제공한다.



항 목	설 명
■	3초이하 서비스 수행
■	3초이상 5초 사이 서비스 수행
■	6초 이상 서비스 수행

- ① 모니터링 상태에서 실시간 동시 서비스 요청 건수를 표시한다.
- ② 더블 클릭 시 Active Service 관련 상태 리스트 확인이 가능하다

ObjectName	Service	Elapsed
/sejb01/UNIPRM-11	/uniprm/main/PDailyClosing	202,871
/sejb01/UNIPRM-11	/uniprm/main/InsuranceBillPubli...	1,241

- ③  버튼을 통해서 5초 단위 Auto refresh 화면 설정 가능하다
- ④ 해당 서비스를 더블 클릭 시에 해당 Thread Detail 확인이 가능하다

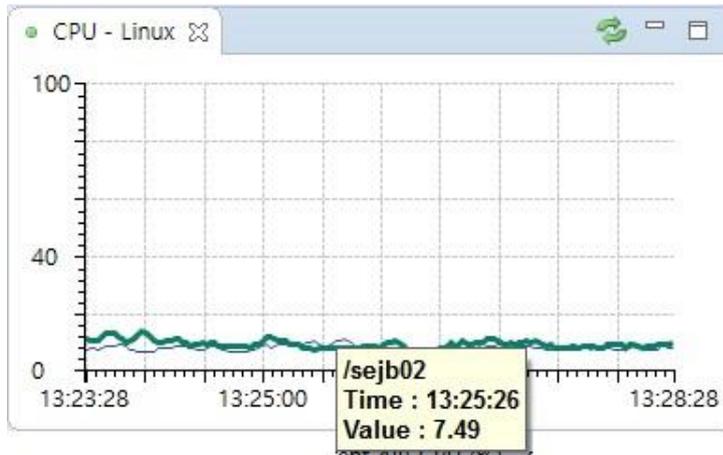
Key	Value
Blocked Count	432,962
Blocked Time	-1
Lock Name	
Lock Owner Id	-1
Lock Owner Name	
State	RUNNABLE
Thread Cpu Time	1,602,760
Thread Id	165
Thread Name	ajp-/0.0.0.0:8009-12
Thread User Time	1,340,010
Waited Count	429,092
Waited Time	-1

```

java.net.SocketInputStream.socketRead0(Native Method)
java.net.SocketInputStream.read(SocketInputStream.java:129)
com.javaservice.lwst.SocketWrapper$InputStreamWrap.read(Unknown Source)
org.apache.coyote.ajp.AjpProcessor.read(AjpProcessor.java:1124)
    
```

4.2.6 CPU – OS

WAS 성능 모니터링을 위해서는 시스템의 CPU의 성능 모니터링을 확인 하는 것이 필요하다. 이를 위해 해당 서버의 CPU 사용률을 나타낸다.



그래프에서 마우스 클릭 시 해당 host에 대한 cpu 정보 확인이 가능하다

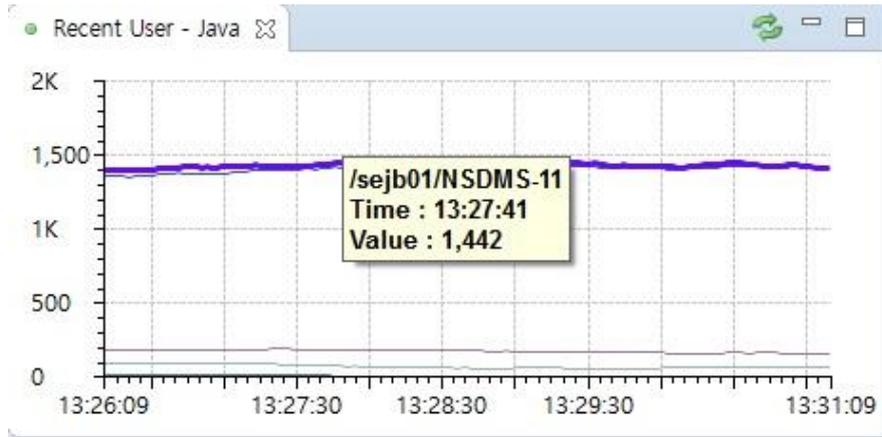
- 우클릭 메뉴



항 목	설 명
Current All	현재 CPU의 사용률
Today All	금일 액티브 서비스 모니터링
Past All	과거 CPU 사용률
Daily All	날짜 별 CPU 사용률

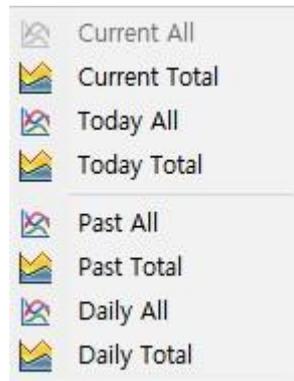
4.2.7 Recent User – Java

서비스 별 최근 5분동안 방문한 사용자 수를 나타낸다.



그래프에서 마우스 클릭 시 해당 Recent User 정보 확인이 가능하다

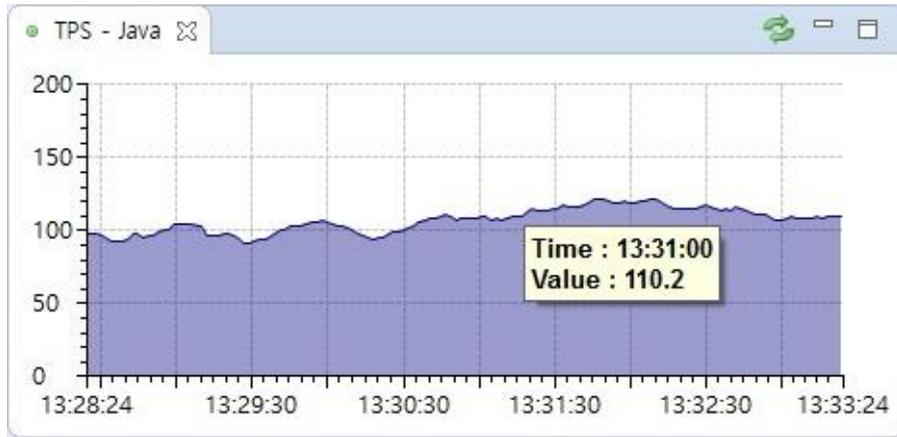
- 우클릭 메뉴



항 목	설 명
Current All	현재 방문한 모든 사용자 수
Current Total	현재 방문한 사용자 수 통계
Today All	금일 방문한 모든 사용자 수
Today Total	금일 방문한 사용자 수 통계
Past All	과거 방문한 사용자 수
Past Total	과거 방문한 사용자 수 통계
Daily All	날짜 별 방문한 모든 사용자 수
Daily Total	날짜 별 방문한 사용자 수 통계

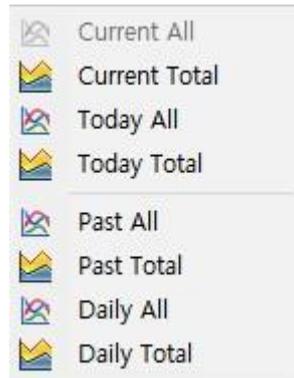
4.2.8 TPS – Java

Java 애플리케이션에서 TPS(Transaction Per Second: 초당 처리건수)의 수치 모니터 링화면에 나타낸다.



그래프에서 마우스 클릭 시 해당 시점 TPS 정보 확인이 가능하다

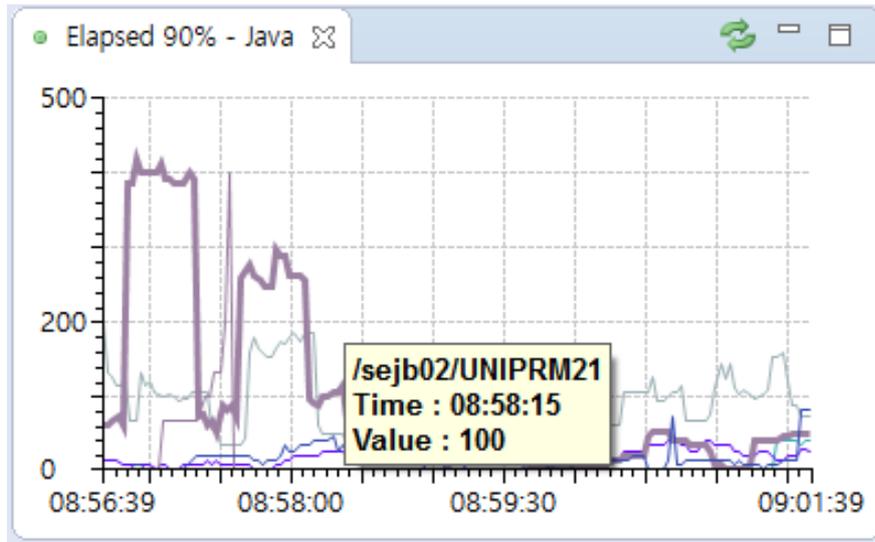
- 우클릭 메뉴



항 목	설 명
Current All	현재 모든 TPS
Current Total	현재 TPS의 통계
Today All	금일 모든 TPS
Today Total	금일 TPS의 통계
Past All	과거 TPS
Past Total	과거 TPS의 통계
Daily All	날짜 별 모든 TPS
Daily Total	날짜 별 TPS의 통계

4.2.9 Elapsed 90%

Java 애플리케이션의 상위 90% 수행시간을 나타낸다.



그래프에서 마우스 클릭 시 해당 시점 수행시간 정보 확인이 가능하다

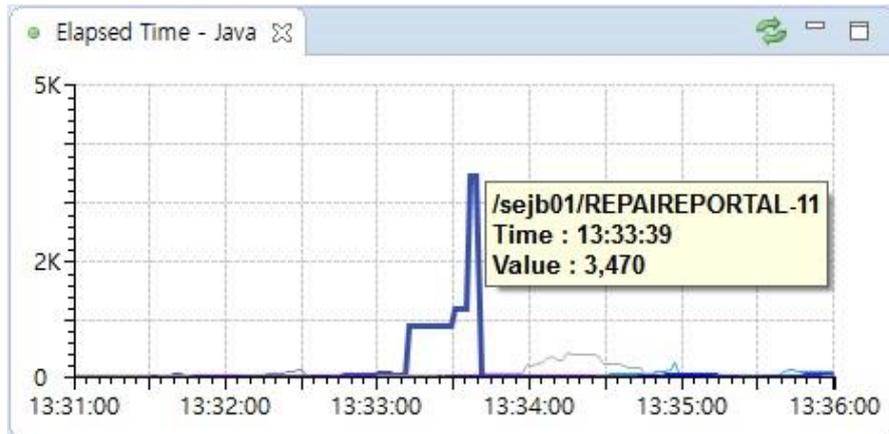
- 우클릭 메뉴



항 목	설 명
Current All	현재 모든 Transaction 수행시간
Today All	금일 모든 Transaction 수행시간
Past All	과거 Transaction 수행시간
Daily All	날짜 별 모든 Transaction 수행시간

4.2.10 Elapsed Time – Java

Java 애플리케이션의 Transaction의 총 수행 시간을 나타낸다.



그래프에서 마우스 클릭 시 해당 시점 수행시간 정보 확인이 가능하다

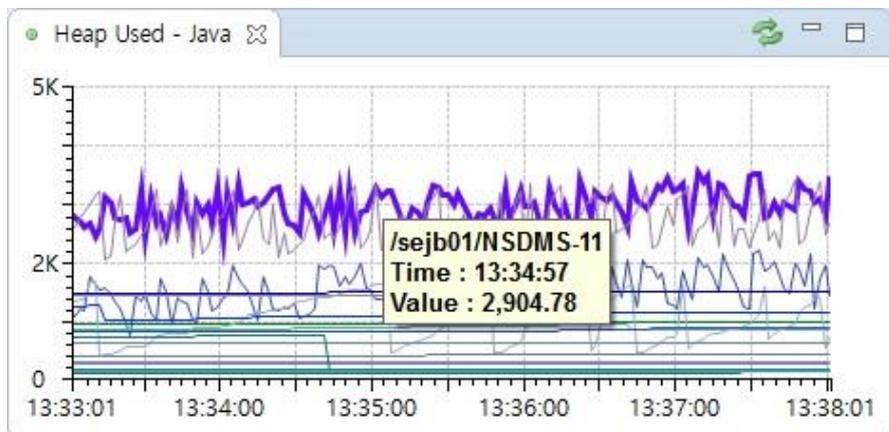
- 우클릭 메뉴



항 목	설 명
Current All	현재 모든 Transaction 수행시간
Today All	금일 모든 Transaction 수행시간
Past All	과거 Transaction 수행시간
Daily All	날짜 별 모든 Transaction 수행시간

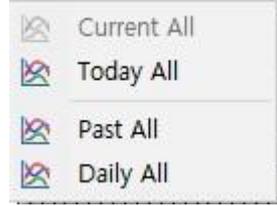
4.2.11 Heap Used – Java

Heap 영역 사용량이나 Garbage Collection 지표 등을 나타낸다.



그래프에서 마우스 클릭 시 해당 시점 Heap 메모리 정보 확인이 가능하다

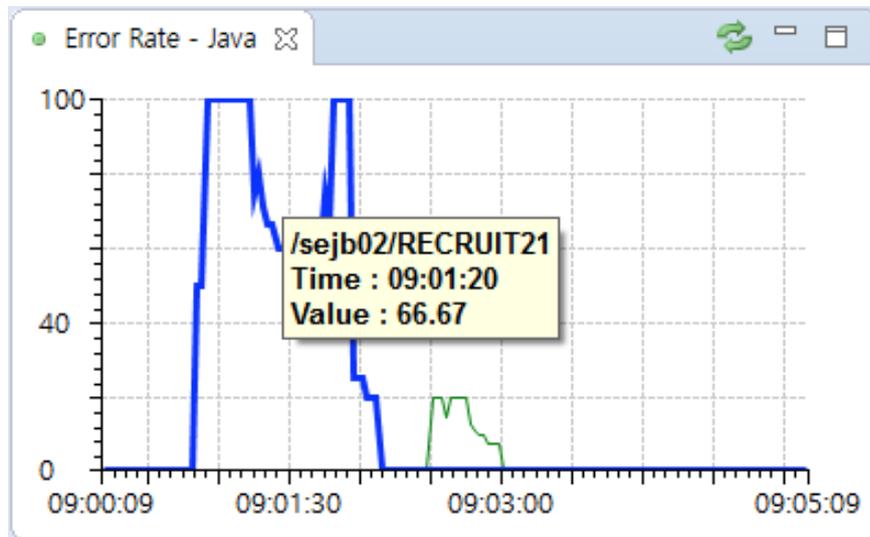
- 우클릭 메뉴



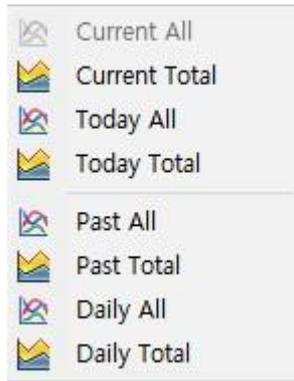
항 목	설 명
Current All	현재 모든 Heap 영역 사용량
Today All	금일 모든 Heap 영역 사용량
Past All	과거 Heap 영역 사용량
Daily All	날짜 별 모든 Heap 영역 사용량

4.2.12 Error Rate

에러가 발생한 서비스의 비율을 화면에 나타낸다



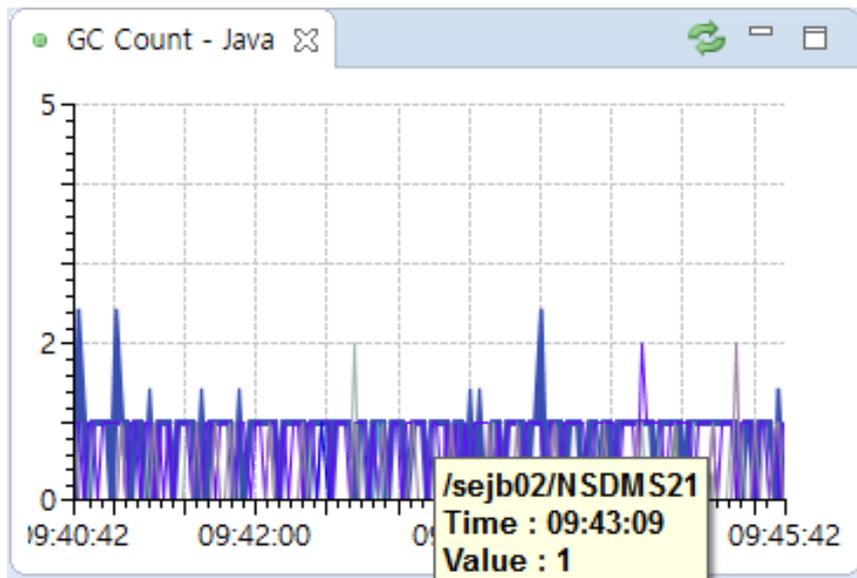
- 우클릭 메뉴



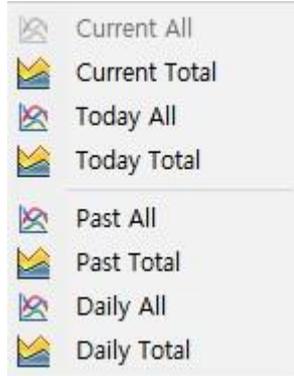
항 목	설 명
Current All	현재 모든 액티브 서비스
Current Total	현재 액티브 서비스 통계
Today All	금일 액티브 서비스 모니터링
Today Total	금일 액티브 서비스 통계
Past All	과거 액티브 서비스
Past Total	과거 액티브 서비스 통계
Daily All	날짜 별 모든 액티브 서비스
Daily Total	날짜 별 액티브 서비스 통계

4.2.13 GC Counter – java

Garbage Collection 횟수를 나타낸다.



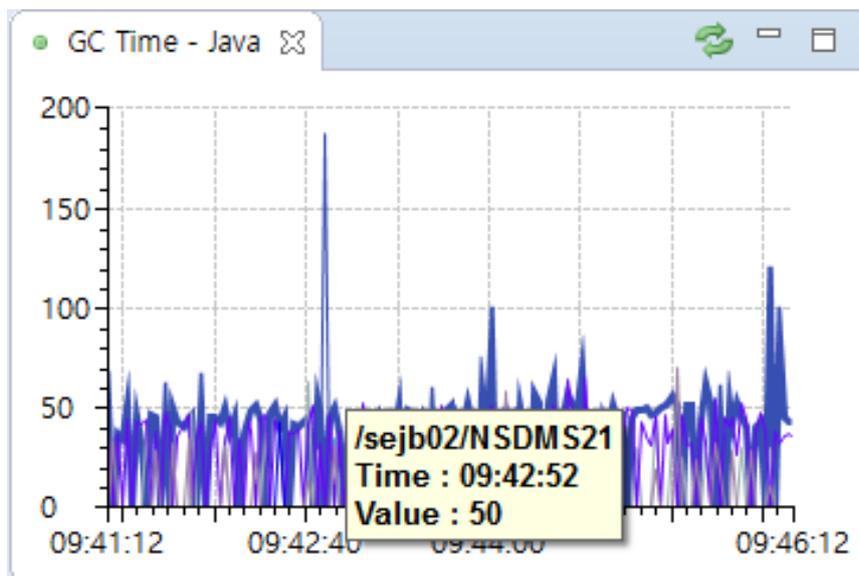
- 우클릭 메뉴



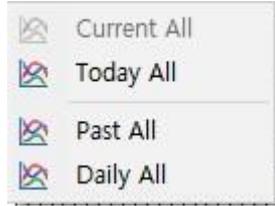
항 목	설 명
Current All	현재 모든 액티브 서비스
Current Total	현재 액티브 서비스 통계
Today All	금일 액티브 서비스 모니터링
Today Total	금일 액티브 서비스 통계
Past All	과거 액티브 서비스
Past Total	과거 액티브 서비스 통계
Daily All	날짜 별 모든 액티브 서비스
Daily Total	날짜 별 액티브 서비스 통계

4.2.14 GC Time – java

Garbage Collection 수행에 소요된 시간을 나타낸다.



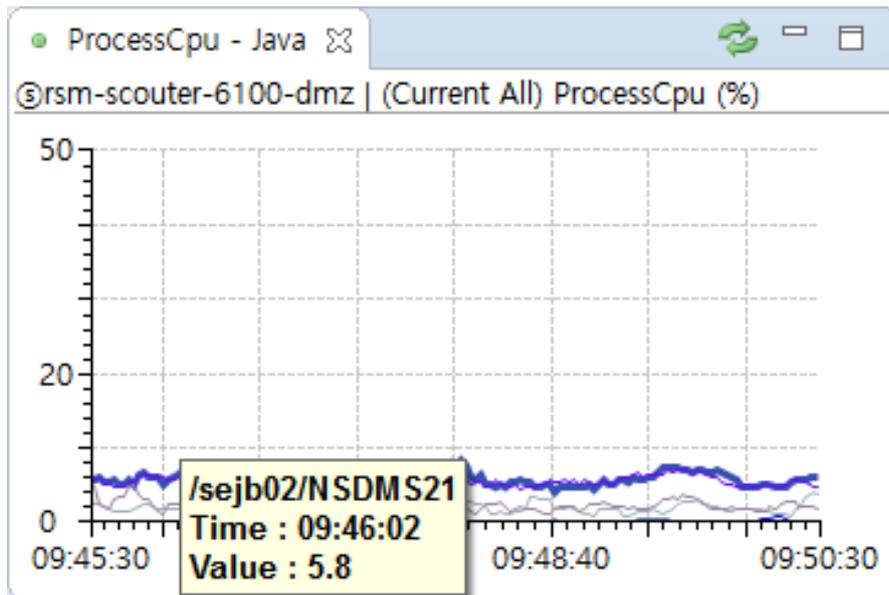
- 우클릭 메뉴



항 목	설 명
Current All	현재 모든 Heap 영역 사용량
Today All	금일 모든 Heap 영역 사용량
Past All	과거 Heap 영역 사용량
Daily All	날짜 별 모든 Heap 영역 사용량

4.2.15 Process CPU

JVM 프로세스의 CPU 사용률을 나타낸다.



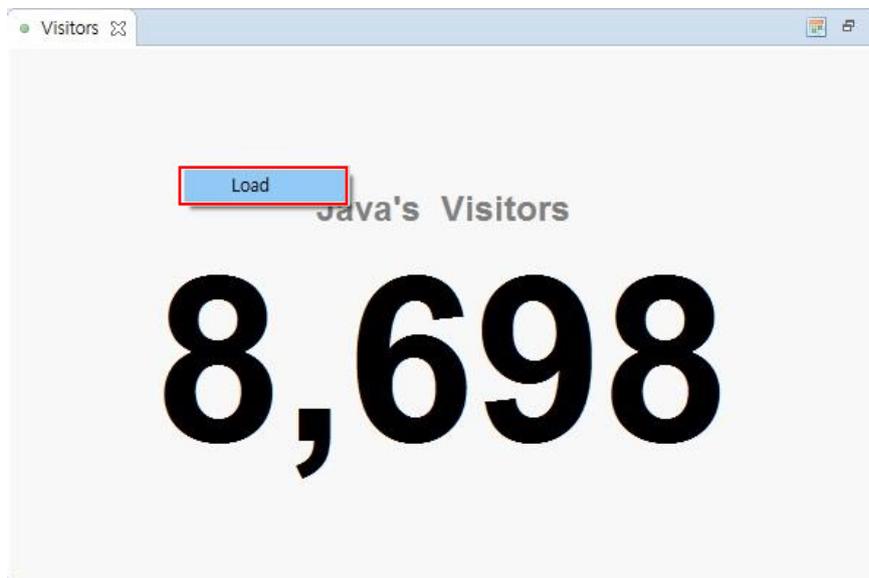
- 우클릭 메뉴



항 목	설 명
Current All	현재 모든 Heap 영역 사용량
Today All	금일 모든 Heap 영역 사용량
Past All	과거 Heap 영역 사용량
Daily All	날짜 별 모든 Heap 영역 사용량

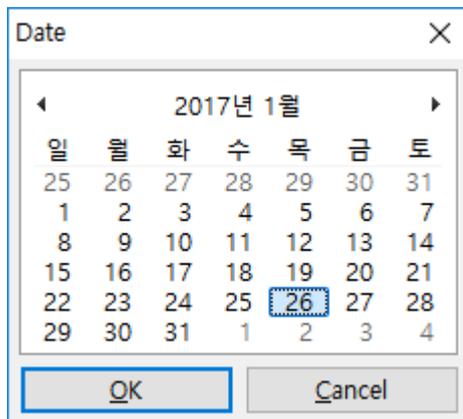
4.2.16 Visitors

금일 방문자가 몇 명이었는지를 실시간으로 카운트하여 화면에 보여준다.



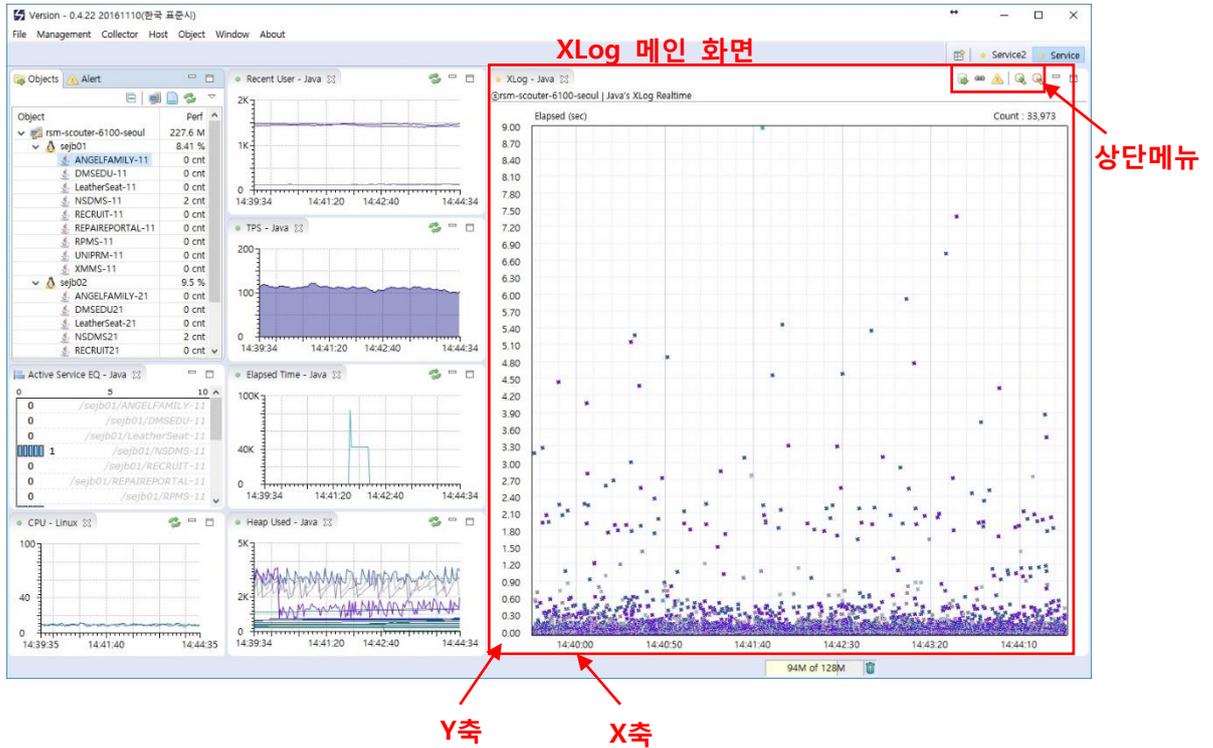
- 우클릭 메뉴(Load)

해당 날짜를 클릭하여 날짜 별 방문자수 확인이 가능하다.



4.3 XLog – 상세 트랜잭션 프로파일링

XLog 그래프는 각 개별 트랜잭션의 응답시간(ELAPSED)관점 뿐만 아니라, 세로축(Y축)을 각각 CPU소요시간. SQL수행시간. SQL패치(Fetch)시간 등의 트랜잭션 인터페이스 시간을 기준으로 분포도의 형태로 나타낸다.



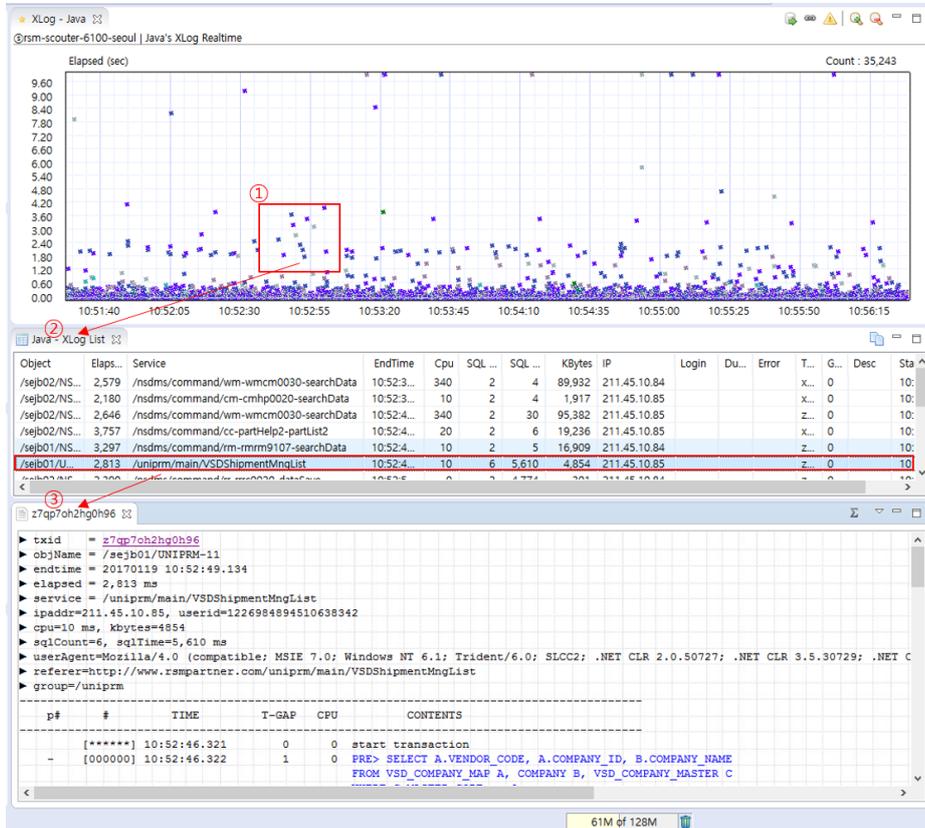
4.3.1 XLog 상단 메뉴(우클릭 메뉴와 동일)

항 목	설 명
Only SQL	SQL에 대한 XLog 보고자 할 경우 클릭
Only API Call	API Call에 대한 XLog를 보고자 할 경우
Only Error	Error에 대한 XLog를 보고자 할 경우
Zoom In	X축 시간을 기준으로 화면 크기를 확대
Zoom Out	X축 시간을 기준으로 화면 크기를 축소

4.3.2 XLog 프로파일링

트랜잭션 종료시간을 X축, 응답시간을 Y축으로 한 점이 트랜잭션을 의미하고 이 점을 선택하여 트랜잭션에 대한 상세 수행 내역을 조회할 수 있다.

정상적인 트랜잭션은 푸른색 계열 색으로 표시되고 예외가 발생한 트랜잭션은 붉은색 계열 색으로 표시된다.



- ① 화면 에서처럼 왼쪽 마우스 버튼을 이용하여 드래그 하면 일부 점들을 선별하여 선택할 수 있다.
- ② 이렇게 선택된 트랜잭션들은 아래 화면처럼 리스트로 나타낸다.
- ③ 리스트에서 하나를 선택하면 해당 트랜잭션에 대한 상세 정보를 볼 수 있다.

4.3.3 트랜잭션 필터링(XLog 상단 메뉴와 동일)

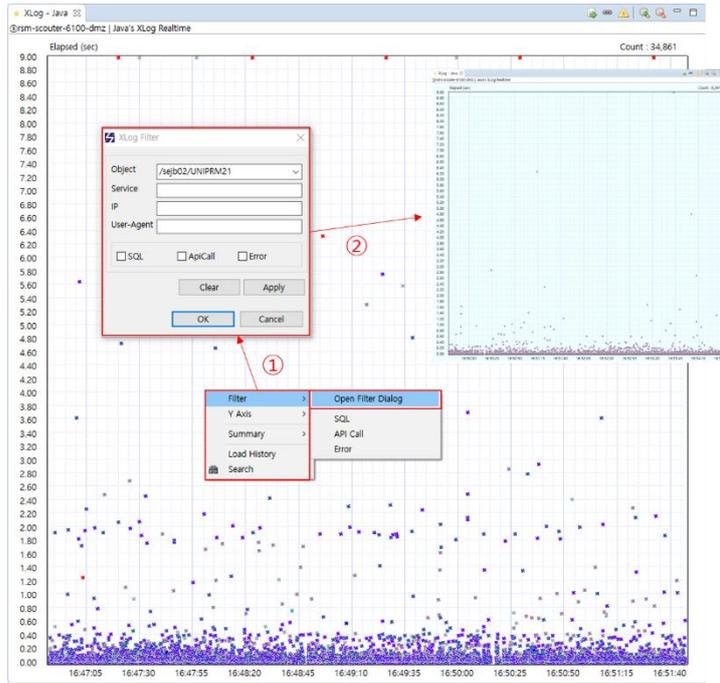
XLog 차트에 표시된 트랜잭션을 여러가지 방법 등을 이용해서 필터링할 수 있다.

- Filter 메뉴(우클릭)

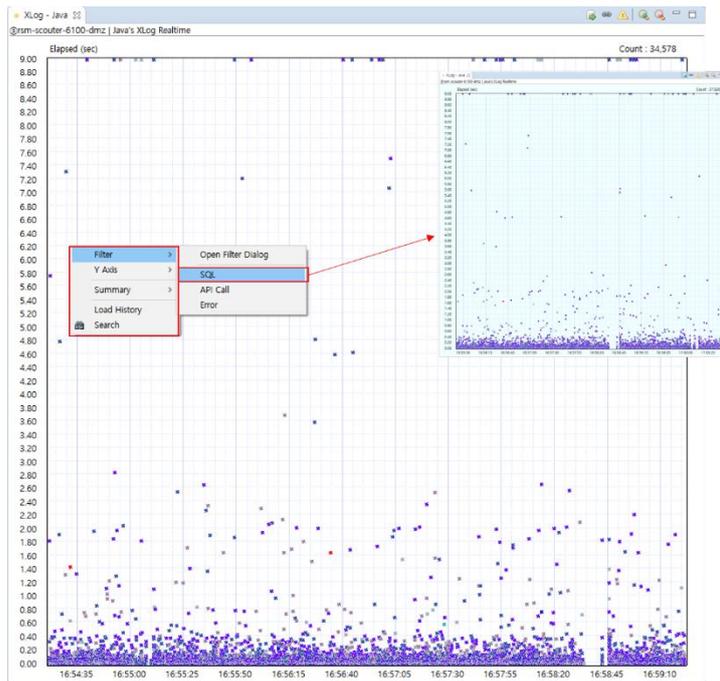
필터를 통해서 SQL로그, API호출 정보, 에러 관련 정보와 함께 문제가 발생하는 상황에 대한 내용으로 애플리케이션에 대한 가시성을 높일 수 있다.

Open filter dialog를 통해서 object, service, ip, agent등을 분류 할 수 있고 sql, apicall, error를 확인이 가능하다.

- ① Open Filter Dialog 메뉴를 통해 특정 트랜잭션에 대해서 모니터링을 위한 필터링이 가능하다.

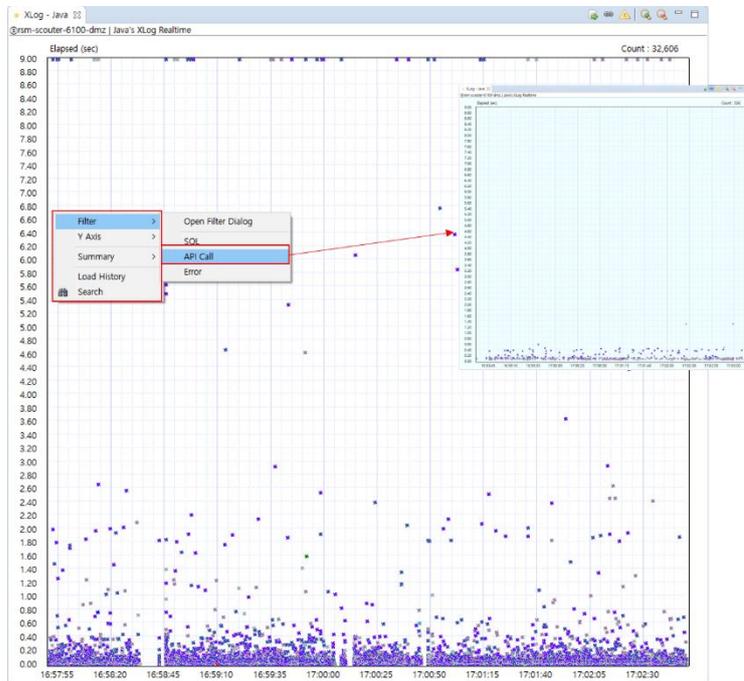


- ② Filter -> SQL 설정으로 SQL만 모니터링이 가능하다.



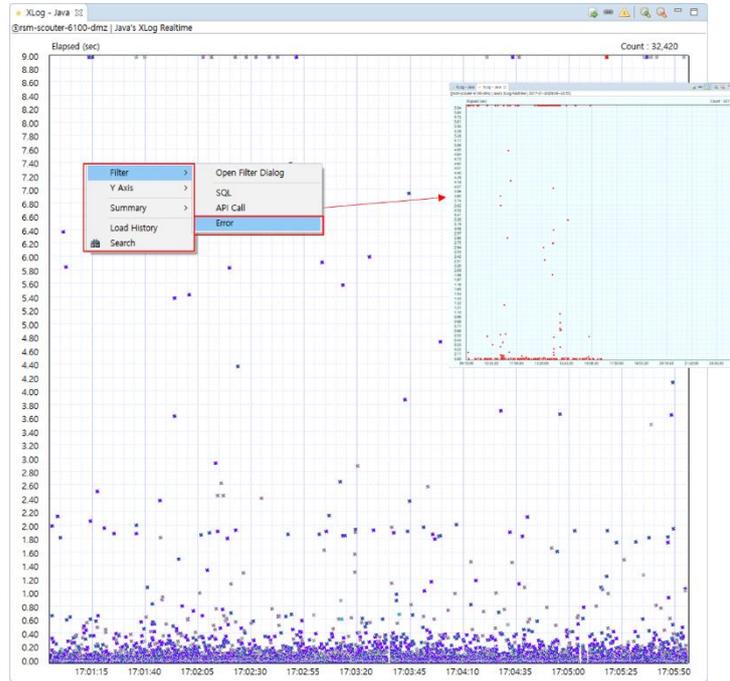
해제 시에는 XLog 탭 우측 상단  클릭

- ③ Filter -> API Call 을 통해 API call에 대한 부분만 모니터링이 가능하다.



해제 시에는 XLog 탭 우측 상단  클릭

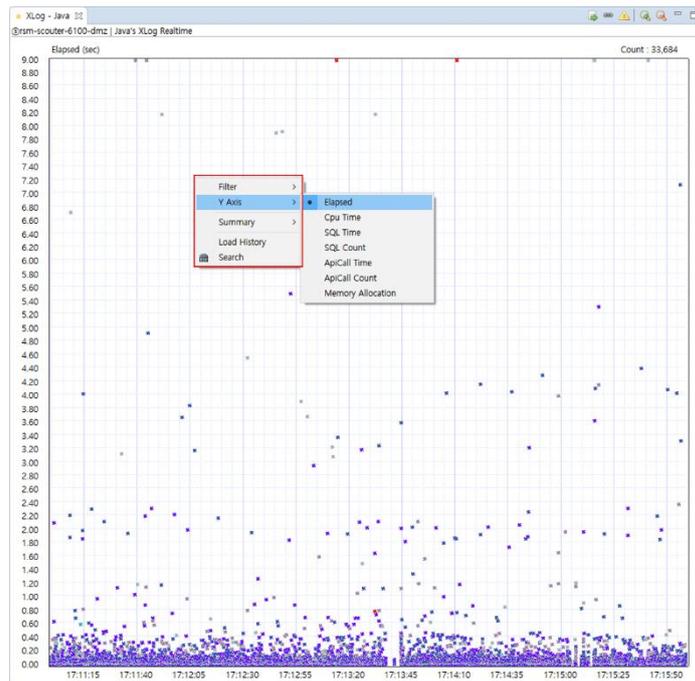
④ Error



해제 시에는 XLog 탭 우측 상단  클릭

● Y Axis 메뉴

Y축을 Elapsed, Cpu Time, SQL time, SQL Count, ApiCall Time, ApiCall Count, Memory Allocation 등 필요시에 모니터링 해야 할 정보를 변경한다

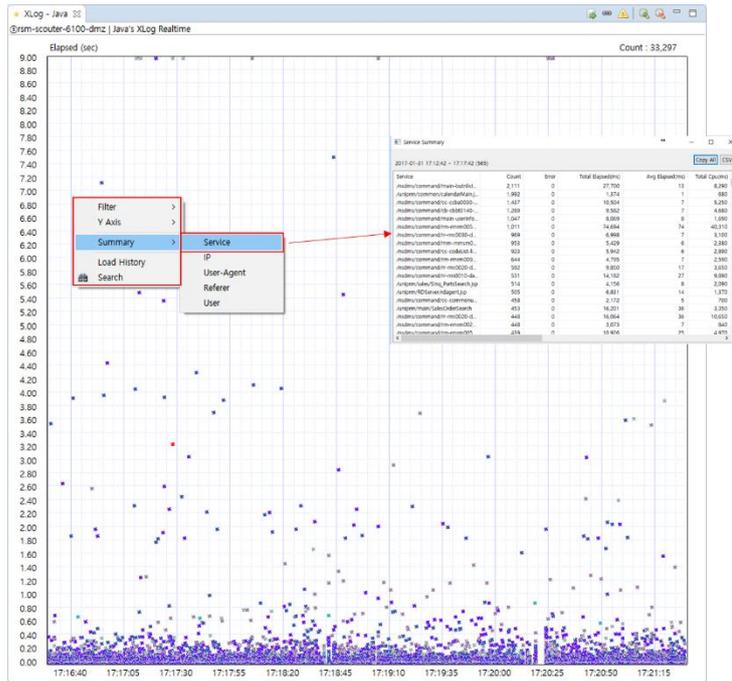


- Summary 메뉴

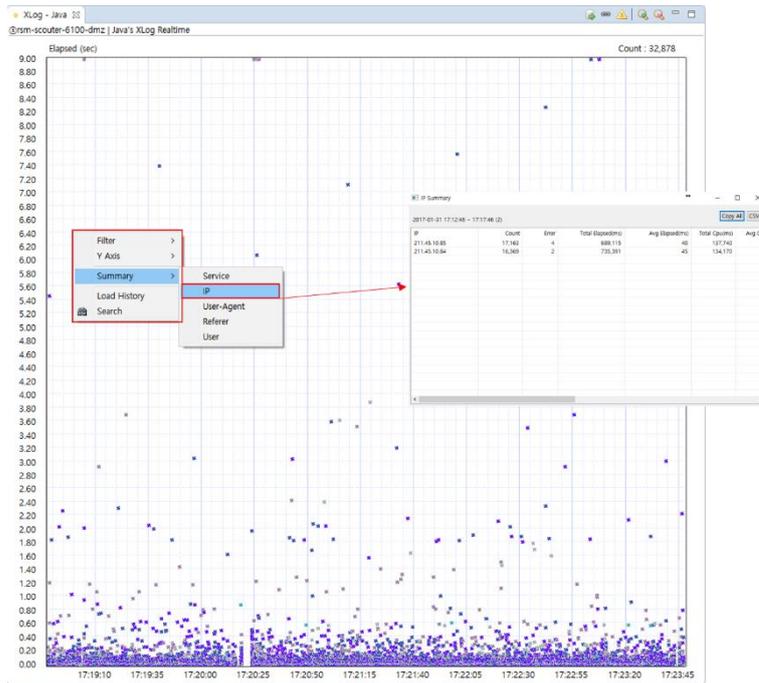
Service, IP, User-Agent, Referer, User 등 모니터링에 대한 Summary를 제공한다.

Copy All을 클릭하면 전체 리스트에 대해 클립보드 복사가 되고 CSV를 클릭하게 되면 리스트의 내용을 파일 형태로 내려 받기가 가능하다.

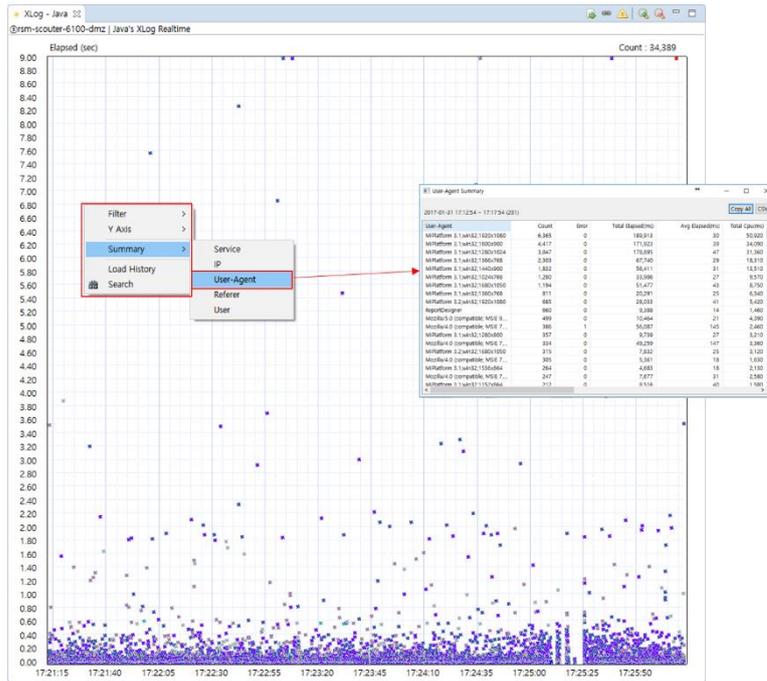
- ① Summary – Service



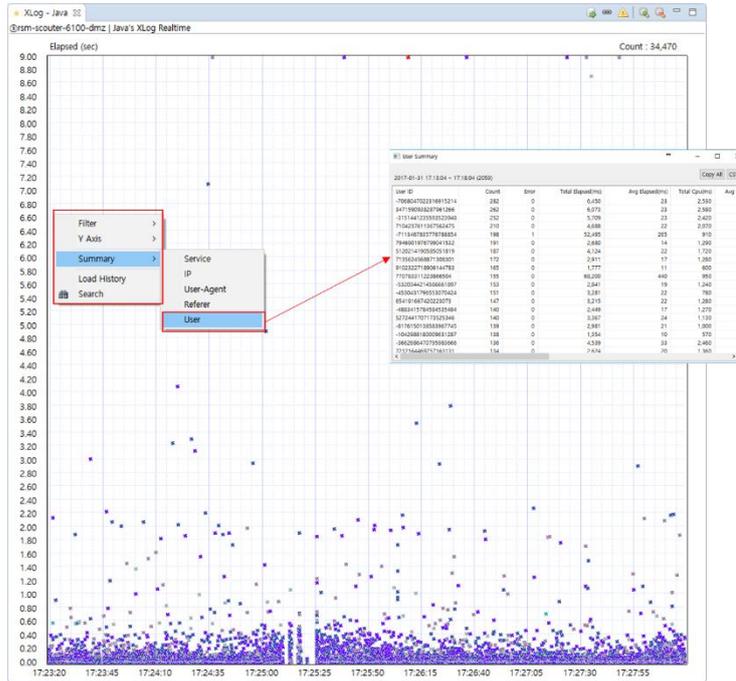
② Summary – IP



③ Summary – User Agent



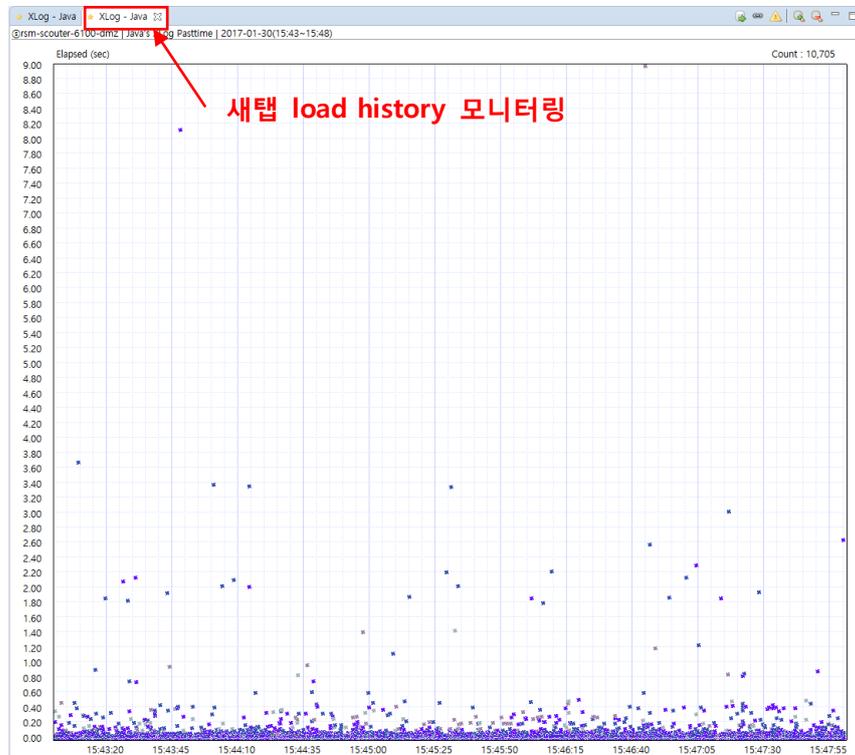
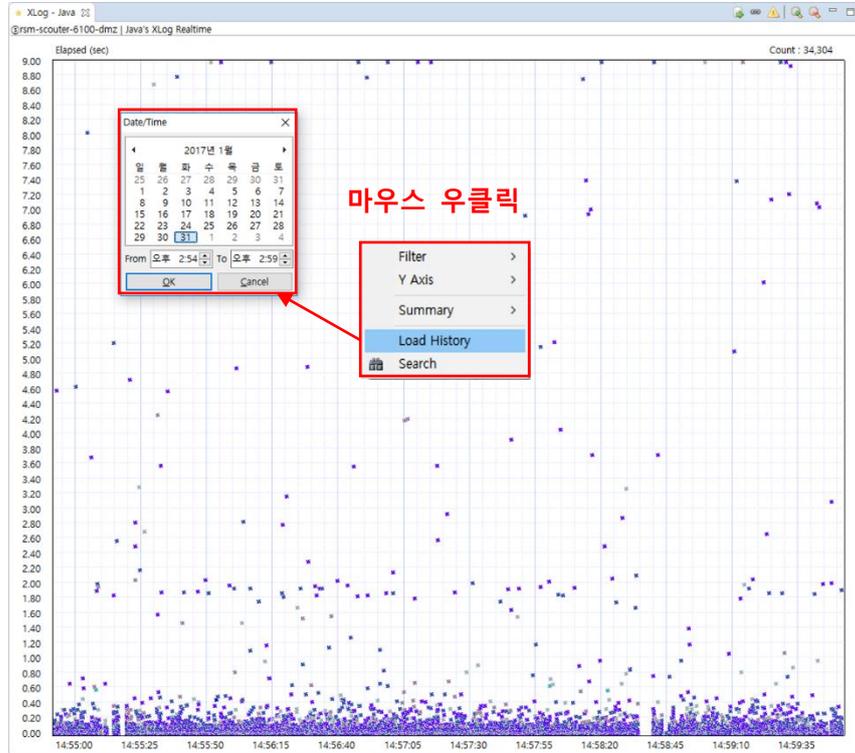
④ Summary – User



● Load History (우클릭)

날짜 별, 시간대별 과거에 대한 데이터를 확인하는 기능을 제공한다.

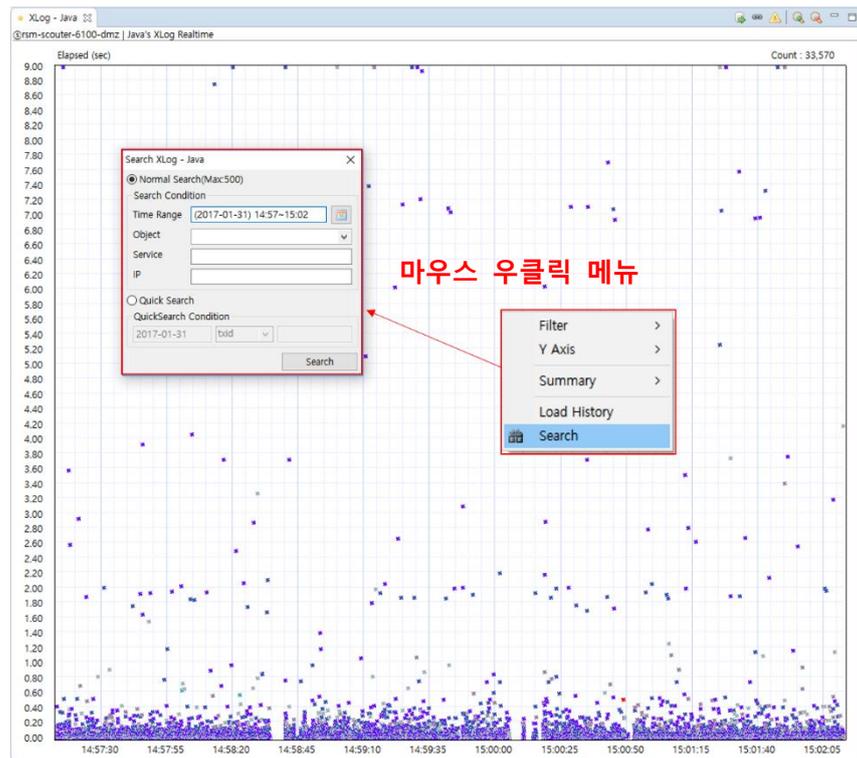
해당 날짜를 선택하면 XLog 형태 그대로 과거 트랜잭션에 대한 화면을 새 탭으로 볼 수 있다.



- Search 메뉴

특정 트랜잭션에 대한 object, service 검색 기능을 제공한다.

선택한 날짜에서 object, Service, IP등을 통해 필터링하면 XLog List를 통해 검색한 트랜잭션에 대한 상세 정보를 제공한다.



Object	Elapsed	Service	EndTime	Cpu	SQL	SQL...	KBytes	IP	Login	Du...	Er...	T...	G...	I...
/sej01/NSDMS-11	17	/nsdms/command/rm-rrc0020-dataComboSearch	15:35:1...	0	2	6	3,958	211.45.10.84				X...	0	
/sej02/NSDMS21	40	/nsdms/command/cc-ccr0010-searchData	15:35:1...	20	2	4	4,868	211.45.10.85				Z...	0	
/sej01/NSDMS-11	7	/nsdms/command/cc-ccb0030-searchLaborData	15:35:1...	20	2	4	2,054	211.45.10.84				Z...	0	
/sej02/NSDMS21	13	/nsdms/command/rm-rrc0030-dataComboSearch	15:35:1...	0	2	4	2,708	211.45.10.84				Z...	0	
/sej02/NSDMS21	8	/nsdms/command/rm-rrm0025-initData	15:35:1...	10	2	2	2,538	211.45.10.85				Z...	0	
/sej02/NSDMS21	19	/nsdms/command/rm-rrm0030-dataSearch	15:35:1...	10	2	4	4,482	211.45.10.85				Z...	0	
/sej02/NSDMS21	6	/nsdms/command/rm-rrm0010-popData	15:35:1...	0	2	4	632	211.45.10.84				X...	0	
/sej01/NSDMS-11	7	/nsdms/command/cc-codetList-CodetAll	15:35:1...	10	2	0	1,975	211.45.10.84				Z...	0	
/sej01/NSDMS-11	13	/nsdms/command/rm-rrm0050-saveData	15:35:1...	10	2	10	163	211.45.10.84				X...	0	
/sej01/NSDMS-11	7	/nsdms/command/rm-rrm0090-freeHourSearch	15:35:1...	0	2	6	1,005	211.45.10.85				X...	0	
/sej01/NSDMS-11	9	/nsdms/command/rm-rrm0024-selectLaborParts	15:35:1...	0	2	4	2,698	211.45.10.84				Z...	0	
/sej01/UNIPRM-11	1	/uniprm/common/calendarMain.jsp	15:35:1...	0	0	0	52	211.45.10.84				X...	0	
/sej02/NSDMS21	6	/nsdms/command/cc-codetList-ListOnlyCode	15:35:1...	0	2	4	1,345	211.45.10.85				Z...	0	
/sej02/UNIPRM21	24	/uniprm/main/SalesOrderSearch	15:35:1...	0	4	38	1,794	211.45.10.85				X...	0	
/sej02/NSDMS21	7	/nsdms/command/rm-rrm0090-stallListSearch	15:35:1...	0	2	4	625	211.45.10.84				Z...	0	
/sej01/NSDMS-11	7	/nsdms/command/rm-rrm0010-popData	15:35:1...	0	2	4	632	211.45.10.85				X...	0	
/sej02/RECRUIT21	121	/recruitApp/SubmitSV	15:35:1...	70	18	202	77,199	211.45.10.85				X...	0	
/sej02/NSDMS21	23	/nsdms/command/cc-ccr0010-searchData	15:35:1...	10	2	6	2,631	211.45.10.85				X...	0	
/sej02/UNIPRM21	1	/uniprm/common/calendarMain.jsp	15:35:1...	0	0	0	52	211.45.10.85				X...	0	
/sej02/NSDMS21	8	/nsdms/command/cc-ccb0030-searchLaborData	15:35:1...	10	2	4	2,055	211.45.10.85				Z...	0	
/sej01/NSDMS-11	139	/nsdms/command/cc-ccb0050-searchData	15:35:1...	40	2	6	66,700	211.45.10.84				X...	0	
/sej02/NSDMS21	61	/nsdms/command/rm-rrm0050-selectData	15:35:1...	30	2	6	46,717	211.45.10.84				Z...	0	
/sej01/UNIPRM-11	56	/uniprm/main/PurchaseList	15:35:1...	30	4	96	4,321	211.45.10.85				X...	0	
/sej02/NSDMS21	14	/nsdms/command/rm-rrs0030-estl_m_j	15:35:1...	0	2	18	259	211.45.10.85				X...	0	
/sej01/NSDMS-11	7	/nsdms/command/main-userinfo-USER_MENU_LOG	15:35:1...	0	2	8	119	211.45.10.84				X...	0	
/sej01/NSDMS-11	10	/nsdms/command/rm-rrs0030-estl_lbr_j	15:35:1...	0	2	4	140	211.45.10.85				X...	0	
/sej02/NSDMS21	13	/nsdms/command/main-butnlst-butnlst	15:35:1...	0	2	4	2,951	211.45.10.85				X...	0	
/sej01/NSDMS-11	16	/nsdms/command/rm-rrs0030-estl_part_j	15:35:1...	0	2	24	145	211.45.10.84				Z...	0	
/sej01/UNIPRM-11	1	/uniprm/common/calendarMain.jsp	15:35:1...	0	0	0	52	211.45.10.85				X...	0	
/sej01/NSDMS-11	198	/nsdms/command/rm-rrc0020-dataSearchVIN	15:35:1...	40	2	4	62,710	211.45.10.85				X...	0	
/sej01/NSDMS-11	27	/nsdms/command/rm-rrc0020-dataSearch	15:35:1...	30	2	4	6,100	211.45.10.85				Z...	0	
/sej01/NSDMS-11	23	/nsdms/command/rm-rrs0010-dataSave	15:35:1...	10	2	26	262	211.45.10.85				X...	0	
/sej02/NSDMS21	4	/nsdms/command/cc-commonunlock-unlock	15:35:1...	0	2	4	188	211.45.10.84				X...	0	
/sej01/NSDMS-11	7	/nsdms/command/cc-ccb0030-searchLaborData	15:35:1...	0	2	4	2,064	211.45.10.84				Z...	0	
/sej01/NSDMS-11	6	/nsdms/command/cc-codetList-ListOnlyCode	15:35:1...	0	2	2	1,340	211.45.10.85				X...	0	
/sej01/NSDMS-11	10	/nsdms/command/pm-pmpo020-getShopData	15:35:1...	20	2	2	13,654	211.45.10.84				Z...	0	
/sej02/NSDMS21	5	/nsdms/command/rm-rrm0023-disc_combolist_adv	15:35:1...	0	2	4	850	211.45.10.85				Z...	0	
/sej02/NSDMS21	14	/nsdms/command/rm-rrs0030-estl_lis	15:35:1...	10	2	4	3,468	211.45.10.84				X...	0	
/sej02/NSDMS21	17	/nsdms/command/cc-partHelp2-partList2	15:35:1...	20	2	6	16,456	211.45.10.84				X...	0	
/sej02/NSDMS21	5	/nsdms/command/cm-cmh0021-initData2	15:35:1...	0	2	4	731	211.45.10.85				Z...	0	
/sej02/NSDMS21	10	/nsdms/command/rm-rrm0023-searchData	15:35:1...	0	2	4	5,309	211.45.10.84				Z...	0	
/sej01/UNIPRM-11	0	/uniprm/PRM_Top.jsp	15:35:1...	0	0	0	59	211.45.10.84				X...	0	
/sej01/UNIPRM-11	20	/uniprm/main/SalesOrderSearch	15:35:1...	10	4	28	1,803	211.45.10.84				Z...	0	
/sej02/NSDMS21	15	/nsdms/command/rm-rrm0025-searchLabor_SSM	15:35:1...	0	2	18	3,041	211.45.10.85				X...	0	
/sej02/NSDMS21	11	/nsdms/command/pm-pmpo020-getShopData	15:35:1...	10	2	4	13,655	211.45.10.84				X...	0	
/sej01/UNIPRM-11	1	/uniprm/selectModule.jsp	15:35:1...	0	0	0	67	211.45.10.84				X...	0	
/sej01/NSDMS-11	103	/nsdms/command/cb-smss010-smCreditCheck	15:35:1...	0	0	0	156	211.45.10.85				Z...	0	
/sej01/UNIPRM-11	1	/uniprm/leftmenu.jsp	15:35:1...	0	0	0	67	211.45.10.84				Z...	0	

4.3.4 XLog List

트랜잭션 데이터 상세 내용은 XLog 차트에서 임의의 점을 선택했을 때 확인할 수 있다.

Object	Elapsed	Service	EndTime	Cpu	SQL	Count	SQL Time	KBytes	IP	Login	Dump	Error	Txid	Gxid	Desc	StartTime
/sej01/NSDMS-11	24	/nsdms/command/rm-rrm0050-initData	18:21:28.662	10	2	6	9,619	211.45.10.85					19qk38dovg9a9	0		18:21:28.638
/sej02/NSDMS21	68	/nsdms/command/pm-pmpo020-searchData	18:21:28.662	60	2	8	135,829	211.45.10.85					z19qkhhcstb0t	0		18:21:28.594
/sej01/NSDMS-11	7	/nsdms/command/rm-rrs0040-labor_gantt_s	18:21:28.665	0	2	4	1,800	211.45.10.85					z197f9m4pwh03	0		18:21:28.658
/sej02/NSDMS21	31	/nsdms/command/rm-rrc0020-dataSearch	18:21:28.670	30	2	4	23,828	211.45.10.85					z197f9m4pwh03	0		18:21:28.667
/sej02/NSDMS21	11	/nsdms/command/rm-rrm0020-saveData	18:21:28.673	10	4	8	275	211.45.10.85					z25zseefg9l8p	0		18:21:28.662
/sej01/NSDMS-11	7	/nsdms/command/cc-ccb0030-searchLaborData	18:21:28.679	0	2	2	2,324	211.45.10.85					z26g967vlgia	0		18:21:28.672
/sej02/NSDMS21	6	/nsdms/command/rm-rrc0010-getNotice	18:21:28.717	20	2	4	3,522	211.45.10.85					z427h30whe790e	0		18:21:28.711
/sej02/NSDMS21	7	/nsdms/command/cc-codetList-ListOnlyCode	18:21:28.723	0	2	4	1,337	211.45.10.84					z426w4h3m9eudl	0		18:21:28.716
/sej02/NSDMS21	6	/nsdms/command/rm-rrm0025-searchLaborData	18:21:28.727	0	2	2	3,033	211.45.10.85					z534r52k69eet	0		18:21:28.721
/sej02/NSDMS21	1	/nsdms/reports/rmd/ANSR0026_unintofed_ASC.mxd	18:21:28.738	0	0	0	35	211.45.10.84					z291f9ub9p0r	0		18:21:28.737
/sej01/NSDMS-11	8	/nsdms/command/rm-rrm0020-searchP_adv	18:21:28.739	10	2	10	226	211.45.10.85					z29z8w9e9tall	0		18:21:28.731
/sej01/NSDMS-11	60	/nsdms/command/rm-rrm0010-selectData	18:21:28.747	40	2	6	46,667	211.45.10.85					z29z12oaw07f4	0		18:21:28.697
/sej02/NSDMS21	6	/nsdms/command/rm-rrm0025-searchFacts	18:21:28.767	0	2	2	1,752	211.45.10.85					z29z8w9e9tall	0		18:21:28.761
/sej01/NSDMS-11	7	/nsdms/command/rm-rrm0090-bookingDataOneSearch2	18:21:28.776	10	2	4	2,123	211.45.10.85					z569g0k644fr	0		18:21:28.769
/sej02/NSDMS21	10	/nsdms/command/main-butnlst-butnlst	18:21:28.781	0	2	2	2,966	211.45.10.84					z716l4oq00ddas	0		18:21:28.771
/sej01/NSDMS-11	37	/nsdms/command/rm-rrm0020-searchListPData	18:21:28.782	0	2	4	2,522	211.45.10.85					z53f5ed9e9j4id	0		18:21:28.755
/sej01/NSDMS-11	11	/nsdms/command/ia-sas0050-searchData	18:21:28.853	0	2	2	2,649	211.45.10.85					z53f5ed9e9j4id	0		18:21:28.842
/sej01/NSDMS-11	25	/nsdms/command/rm-rrs0010-sataSearch	18:21:28.867	20	2	4	18,463	211.45.10.85					x6me9h7tuss6g	0		18:21:28.842
/sej01/NSDMS-11	21	/nsdms/command/rm-rrm0020-initData	18:21:28.867	10	2	4	9,182	211.45.10.84					z58m93h2421	0		18:21:28.846
/sej01/NSDMS-11	5	/nsdms/command/rm-rrm0010-popData	18:21:28.904	0	2	4	623	211.45.10.84					x6m172ag3k314t	0		18:21:28.899
/sej02/NSDMS21	7	/nsdms/command/cc-ccb0030-searchLaborData	18:21:28.912	10	2	4	2,153	211.45.10.85					z604z2ow9992ch	0		18:21:28.905
/sej02/NSDMS21	7	/nsdms/command/rm-rrc0010-getNotice	18:21:28.916	0	2	4	3,523	211.45.10.85					x6m172ag3k314t	0		18:21:28.909
/sej02/NSDMS21	13	/nsdms/command/rm-rrs0000-vch_lbr_j	18:21:28.916	10	2	2	3,004	211.45.10.85					x6v4z2159ag0j	0		18:21:28.903
/sej02/NSDMS21	7	/nsdms/command/rm-rrm0025-searchData	18:21:28.917	10	2	2	2,538	211.45.10.85					z53f5ed9e9j4id	0		18:21:28.910
/sej01/NSDMS-11	4	/nsdms/command/cc-commonunlock-unlock	18:21:28.920	0	2	4	188	211.45.10.85					x52h71h6vgvth	0		18:21:28.916
/sej02/NSDMS21	122	/nsdms														

항 목	설 명
Object	Object 명
Elapsed	수행시간
Service	서비스 명
EndTime	종료 시간
Cpu	Cpu
SQL Count	SQL 수행 건수
SQL Time	SQL 수행 시간
IP	클라이언트 IP
Txid	트랜잭션 ID
Gxid	글로벌 트랜잭션 ID
StartTime	Start 시간

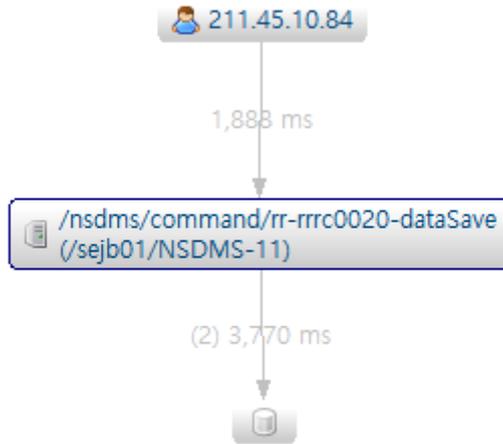
4.3.5 트랜잭션 상세 정보

XLog list에서 해당 트랜잭션을 더블 클릭하면 트랜잭션에 대한 정보를 상세하게 확인 할 수 있다.

```

▶ txid = x1q5u6epqf8p5p
▶ objName = /sejb01/NSDMS-11
▶ endtime = 20170126 14:03:36.493
▶ elapsed = 4,439 ms
▶ service = /nsdms/command/cm-cmhp0020-searchData
▶ ipaddr=211.45.10.84, userid=7484654359174877364
▶ cpu=0 ms, kbytes=2039
▶ sqlCount=2, sqlTime=2 ms
▶ userAgent=MiPlatform 3.2;win32;1280x800
▶ group=/nsdms
-----
p#      #          TIME          T-GAP    CPU    CONTENTS
-----
-      [*****] 14:03:32.054    0       0    start transaction
-      [0000000] 14:03:32.055    1       0    PRE> {call PG_CMHP0020.pr_searchData(?,?,?,?,?,?)}
[0000000] [0000001] 14:03:32.055    0       0    PRE> {call PG_CMHP0020.pr_searchData(?,?,?,?,?,?)}
-      [0000002] 14:03:36.492    4,437   0    RESULT-SET-FETCH #19 1 ms
-      [*****] 14:03:36.493    1       0    end of transaction
-----
    
```

트랜잭션 아이디를 클릭 하였을 경우 아래와 같이 클라이언트 요청 구간 확인 이 가능하다.

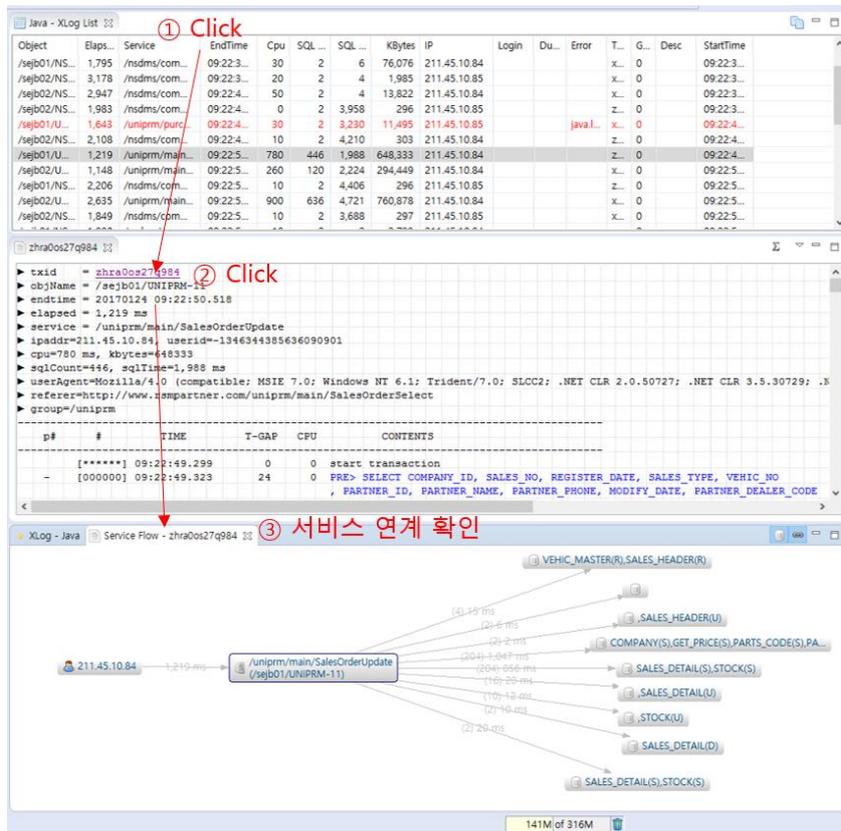


해당 트랜잭션을 클릭 후 상단  클릭 시 클립보드에 복사 기능도 제공됩니다.

항 목	설 명
txid	트랜잭션 id
Objname	오브젝트 이름
Endtime	수행 종료 시간
Elapsed	수행 시간
Service	서비스 이름
Ipaddr	Ip 주소
Sqlcount	Sql 수행 건수
Sqltime	Sql 수행 시간
group	소속 그룹

4.3.6 서비스 연계 추적

WAS 시스템의 장애 분석 시 문제 고립화 작업을 통해 문제의 원인에 접근하게 된다. 문제 고립화 작업 시 클래스, 메소드 사용에 대한 상세 프로파일링을 어느 수준까지 상세히, 그리고 시스템에 부담을 주지 않고 할 수 있는지가 이 작업의 핵심이라 할 수 있다. 서비스 연계와 관련된 화면을 보기 위해서는 java agent 설정 conf파일 또는 스카우터 클라이언트 configure에서 `race_interservice_enabled=true` 설정이 있어야 한다.

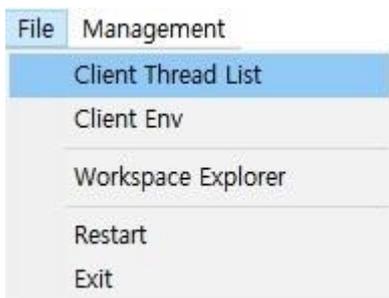


- ① XLog List를 보면 transaction 요청을 확인할 수 있다.
- ② 실제 transaction의 txid를 선택하면 서비스 호출 연계 내용이 확인 된다.
- ③ 서비스 연계 내용을 확인 후 그 하위의 연계를 확인하기를 원한다면, 해당 transaction의 call 연계 부분을 클릭하여 이후 서비스 연계를 확인 할 수 있다.

5. Scouter 추가 기능

5.1 클라이언트 스레드 리스트

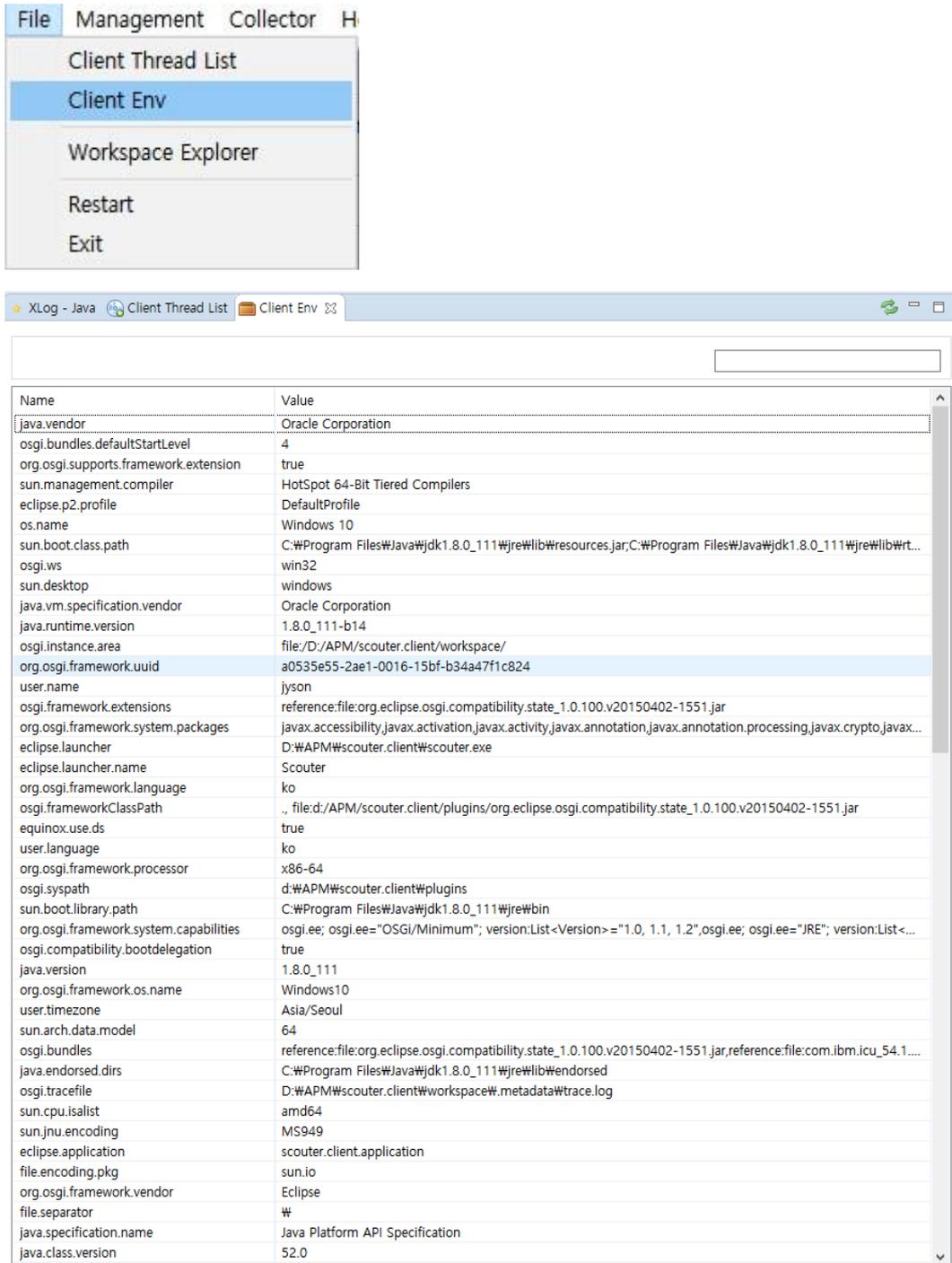
스레드 리스트를 통해 해당 서버에 스레드의 상태와 cpu 사용률을 확인 할 수 있다.



No	Name	Stat	Cpu
001	main	RUNNABLE	4,360,546
002	Reference Handler	WAITING	375
003	Finalizer	WAITING	218
005	Signal Dispatcher	RUNNABLE	0
006	Attach Listener	RUNNABLE	0
012	Active Thread: Equinox Container: a0535e55-2ae1-0016-15bf-b34a47f1c824	TIMED_WAITI...	0
014	Framework Event Dispatcher: Equinox Container: a0535e55-2ae1-0016-15bf-b34a47f1c824	WAITING	62
015	Start Level: Equinox Container: a0535e55-2ae1-0016-15bf-b34a47f1c824	WAITING	156
016	Worker-JM	WAITING	0
018	[Timer] - Main Queue Handler	TIMED_WAITI...	15
021	EMF Reference Cleaner	WAITING	796
023	scouter.client.server.ServerManager	TIMED_WAITI...	1,718
027	scouter.client.remote.CheckMyJob	TIMED_WAITI...	1,765
028	SessionObserverThread	TIMED_WAITI...	250
029	scouter.client.threads.AlertProxyThread	TIMED_WAITI...	2,500
030	EventAdmin Async Event Dispatcher Thread	WAITING	109
031	scouter.client.model.AgentModelThread	TIMED_WAITI...	4,562
032	scouter.client.views.ObjectNavigationView@5e8cda75	TIMED_WAITI...	78
035	scouter.client.views.EQView@62db3891	TIMED_WAITI...	628,781
036	scouter.client.counter.views.CounterRealTimeAllView@6eb17ec8 - objType:linux, counter:Cpu, serverid:1839464977	TIMED_WAITI...	2,156
037	scouter.client.counter.views.CounterRealTimeAllView@4649d70a - objType:java, counter:RecentUser, serverid:1839464977	TIMED_WAITI...	3,562
038	scouter.client.counter.views.CounterRealTimeTotalView@18d910b3 - objType:java, counter:TPS, serverid:1839464977	TIMED_WAITI...	3,218
039	scouter.client.counter.views.CounterRealTimeAllView@52d97ab6 - objType:java, counter:ElapsedTime, serverid:1839464977	TIMED_WAITI...	3,375
040	scouter.client.counter.views.CounterRealTimeAllView@15639440 - objType:java, counter:HeapUsed, serverid:1839464977	TIMED_WAITI...	2,968
041	scouter.client.xlog.views.XLogRealTimeView@142213d5 - objType:java, serverid:1839464977	TIMED_WAITI...	15,093
054	[ThreadPool Manager] - Idle Thread	WAITING	0
1038	scouter.client.views.ObjectActiveServiceListView@9de1b85	TIMED_WAITI...	0
1044	scouter.client.views.ObjectActiveServiceListView@43fbc2bf	TIMED_WAITI...	31
1051	scouter.client.views.ObjectActiveServiceListView@1cca3e8c	TIMED_WAITI...	0
1059	scouter.client.views.ObjectActiveServiceListView@76134251	TIMED_WAITI...	15
1062	scouter.client.views.ObjectActiveServiceListView@3151277f	TIMED_WAITI...	15
1066	scouter.client.views.ObjectActiveServiceListView@3d7ca9c	TIMED_WAITI...	0
1074	scouter.client.views.ObjectActiveServiceListView@64e89bb2	TIMED_WAITI...	15
1202	scouter.client.views.ObjectActiveServiceListView@7d563c13	TIMED_WAITI...	0
12556	scouter.client.views.ObjectDashboardView@2a4c1ef3	TIMED_WAITI...	62
13812	scouter.client.views.ObjectDashboardView@67e36a47	TIMED_WAITI...	46
13815	scouter.client.views.ObjectActiveServiceListView@594d6436	TIMED_WAITI...	15
13819	scouter.client.views.ObjectActiveServiceListView@5c0c7c98	TIMED_WAITI...	0
13821	scouter.client.views.ObjectActiveServiceListView@484180a7	TIMED_WAITI...	0
13823	scouter.client.views.ObjectActiveServiceListView@ba39e5	TIMED_WAITI...	0
25887	Worker-23	TIMED_WAITI...	1,281
285	scouter.client.views.ObjectActiveServiceListView@6dca31eb	TIMED_WAITI...	0
289	scouter.client.views.ObjectActiveServiceListView@3d9f0a5	TIMED_WAITI...	0
291	scouter.client.views.ObjectActiveServiceListView@3c19592c	TIMED_WAITI...	15
294	scouter.client.views.ObjectActiveServiceListView@58dad04a	TIMED_WAITI...	0
297	scouter.client.views.ObjectActiveServiceListView@1f7cec93	TIMED_WAITI...	15
302	scouter.client.views.ObjectActiveServiceListView@241fbec	TIMED_WAITI...	0
307	scouter.client.views.ObjectActiveServiceListView@626e0c86	TIMED_WAITI...	0
310	scouter.client.views.ObjectActiveServiceListView@372461a9	TIMED_WAITI...	15
3543	scouter.client.views.ObjectActiveServiceListView@5b979374	TIMED_WAITI...	15
54554	scouter.client.views.ObjectActiveServiceListView@6f4b7d5f	TIMED_WAITI...	0
54870	scouter.client.counter.views.CounterRealTimeAllView@2ef63356 - objType:java, counter:ErrorRate, serverid:1839464977	TIMED_WAITI...	484
55947	scouter.client.counter.views.CounterRealTimeAllView@5ce7162c - objType:java, counter:TPS, serverid:1839464977	TIMED_WAITI...	390
56091	scouter.client.views.VerticalEQView@2a1c70b8	TIMED_WAITI...	69,765
56476	scouter.client.counter.views.CounterRealTimeAllView@37bb7cda - objType:java, counter:ActiveService, serverid:1839464977	TIMED_WAITI...	453
57121	scouter.client.counter.views.CounterRealTimeTotalView@5bd4551a - objType:java, counter:ActiveService, serverid:1839464977	TIMED_WAITI...	296
61318	scouter.client.views.ObjectActiveServiceListView@744b6793	TIMED_WAITI...	0
61379	scouter.client.views.ObjectActiveServiceListView@268cd075	TIMED_WAITI...	0
61386	scouter.client.views.ObjectActiveServiceListView@75497b35	TIMED_WAITI...	0
61389	scouter.client.views.ObjectActiveServiceListView@17eac67b	TIMED_WAITI...	0
61635	scouter.client.views.ObjectActiveServiceListView@756deebc	TIMED_WAITI...	0
61640	scouter.client.views.ObjectActiveServiceListView@2f6a26be	TIMED_WAITI...	0
61658	scouter.client.views.ObjectActiveServiceListView@8d8a7ff	TIMED_WAITI...	0
61664	scouter.client.views.ObjectActiveServiceListView@6da0714f	TIMED_WAITI...	15
61675	scouter.client.views.ObjectActiveServiceListView@2954ca09	TIMED_WAITI...	15

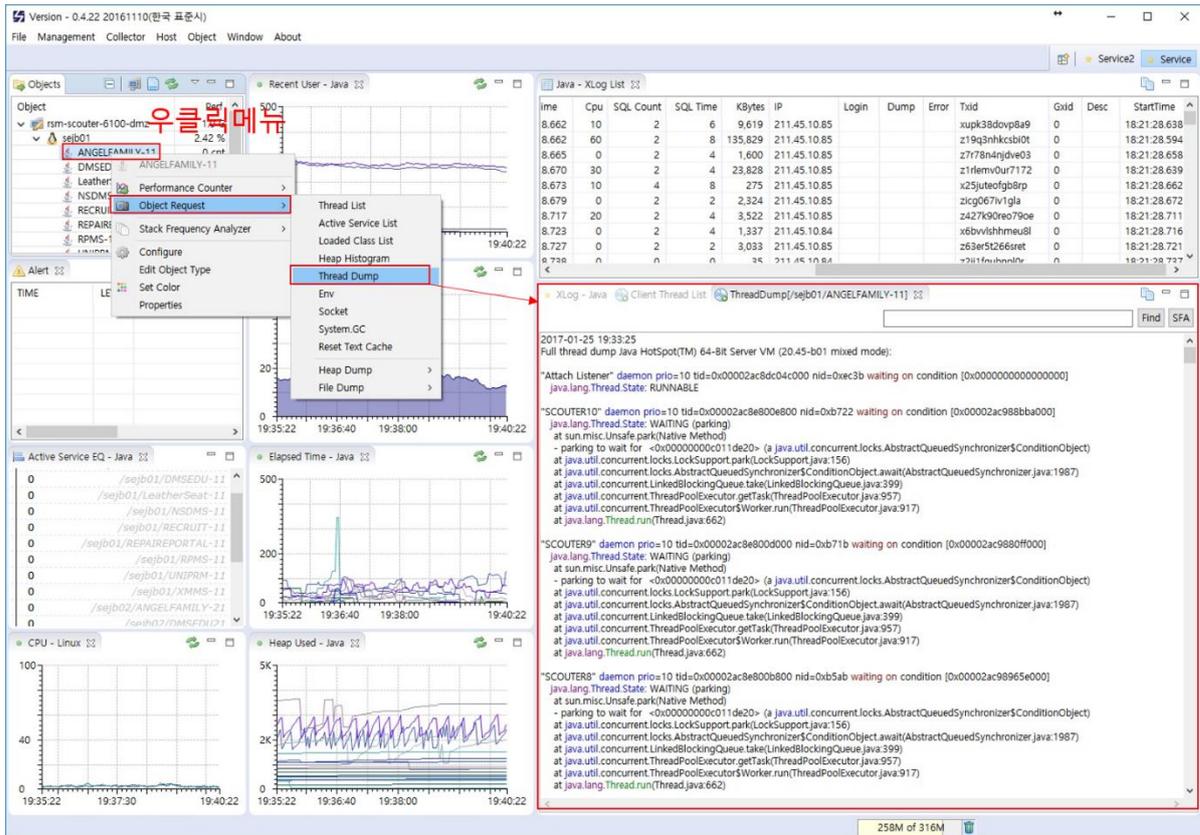
5.2 클라이언트 환경 설정 정보

실행중인 스카우터 클라이언트에 대한 OS, java, 디렉토리 등 환경 정보를 확인 할 수 있다.



5.3 서비스 덤프

상대적 성능 장애(Relative Performance Problem)이 발생하면 수행 중인 서비스의 응답시간 지연이 발생하기 때문에 문제의 소지가 있는 상황을 파악하기 위해 다양한 덤프 형태의 목록을 제공한다.



스레드 덤프를 통한 서비스 장애 시 애플리케이션 문제를 확인

6. Scouter Alert

6.1 Alert 가능한 종류

- Agent의 CPU (warning/fatal)
- Agent의 Memory (warning/fatal)
- Agent의 Disk (warning/fatal)
- 신규 Agent 연결
- Agent의 연결 해제
- Agent의 재접속
- 응답시간 임계치 초과
- GC Time의 임계치 초과

- Thread 개수의 임계치 초과

6.2 Email

스카우터 서버 플러그인 방식으로 Email을 통해서 Alert 메시지를 Email으로 전송하는 기능이다.

6.2.1 Scouter 서버 라이브러리 추가

- Activation-1.1.1.jar
- Commons-email-1.4.jar
- Javax.mail-1.5.2.jar
- 다운로드 링크

<https://github.com/scouter-project/scouter-plugin-server-alert-email/tree/master/lib>

```
$ cd ${SCOUTER_HOME}/server/lib
$ ls
activation-1.1.1.jar commons-email-1.4.jar javax.mail-1.5.2.jar
```

6.2.2 scouter.conf 설정 파일의 Properties

```
$ vi ${SCOUTER_HOME}/server/conf/scouter.conf
```

- ext_plugin_email_send_alert: Email 메시지 발송 여부 (true/false) – default: false
- ext_plugin_email_debug: 로깅 여부 – default: false
- ext_plugin_email_level: 수신 레벨 0: INFO, 1:WARN, 2:ERROR, 3:FATAL - default: 0
- ext_plugging_email_smtp_hostname: SMTP 서버의 IP 또는 Domain
- ext_plugging_email_smtp_port: SMTP Port – default 587
- ext_plugging_email_username: Email 사용자 계정
- ext_plugging_email_password: Email 사용자 비밀번호
- ext_plugin_email_tls_enabled: TLS 사용 여부 – default: true

- ext_plugin_email_from_address: Email 발신자 계정
- ext_plugin_email_to_address: Email 수신 계정 (다중 사용자 지정시 ' ' 구분자 사용)
- ext_plugin_email_cc_address: Email 참조 수신 계정 (다중 사용자 지정시 ' ' 구분자 사용)
- ext_plugin_elapsed_time_threshold: 응답시간의 임계치(ms) – default 값은 0으로, 0일 때 응답시간의 임계치 초과 여부를 확인하지 않는다.
- ext_plugin_gc_time_threshold: GC Time의 임계치(ms) – default 값은 0으로, 0일 때 GC Time의 임계치 초과 여부를 확인하지 않는다.
- ext_plugin_thread_count_threshold: Thread Count의 임계치 - default 값은 0으로, 0일 때 Thread Count의 임계치 초과 여부를 확인하지 않는다.
- Example

```
# External Interface (Email)
ext_plugin_email_send_alert=true
ext_plugin_email_debug=true
ext_plugin_email_level=0
ext_plugin_email_smtp_hostname=smtp.gmail.com
ext_plugin_email_smtp_port=587
ext_plugin_email_username=noreply@scouter.com
ext_plugin_email_password=password
ext_plugin_email_tls_enabled=true
ext_plugin_email_from_address=noreply@scouter.com
ext_plugin_email_to_address=receiver1@scouter.com,receiver2@scouter.com
ext_plugin_email_cc_address=creceiver@scouter.com
ext_plugin_elapsed_time_threshold=5000
ext_plugin_gc_time_threshold=5000
ext_plugin_thread_count_threshold=300
```

6.3 Telegram

스카우터 서버 플러그인 방식으로 Telegram을 통해서 Alert 메시지를 Telegram으로 전송하는 기능이다.

6.3.1 Scouter 서버 라이브러리 추가

- commons-codec-1.9.jar

- commons-logging-1.2.jar
- gson-2.6.2.jar
- httpclient-4.5.2.jar
- httpcore-4.4.4.jar
- 다운로드 링크

<https://github.com/scouter-project/scouter-plugin-server-alert-telegram/tree/master/lib>

```
$ cd ${SCOUTER_HOME}/server/lib
$ ls
commons-codec-1.9.jar commons-logging-1.2.jar gson-2.6.2.jar
httpclient-4.5.2.jar httpcore-4.4.4.jar
```

6.3.2 Scouter.conf 설정 파일의 Properties

```
$ vi ${SCOUTER_HOME}/server/conf/scouter.conf
```

- ext_plugin_telegram_send_alert: Telegram 메시지 발송 여부 (true/false) - default false
- ext_plugin_telegram_debug: 로깅 여부 - default false
- ext_plugin_telegram_level: 수신 레벨 0: INFO, 1:WARN, 2:ERROR, 3:FATAL - default 0
- ext_plugging_telegram_bot_token: Telegram Bot Token
- ext_plugin_telegram_chat_id: chat_id(Integer) 또는 채널이름(String)
- ext_plugin_elapsed_time_threshold: 응답시간의 임계치(ms) - default 값은 0으로, 0일 때 응답시간의 임계치 초과 여부를 확인하지 않는다.
- ext_plugin_gc_time_threshold: GC Time의 임계치(ms) - default 값은 0으로, 0일 때 GC Time의 임계치 초과 여부를 확인하지 않는다.
- ext_plugin_thread_count_threshold: Thread Count의 임계치 - default 값은 0으로, 0일

때 Thread Count의 임계치 초과 여부를 확인하지 않는다.

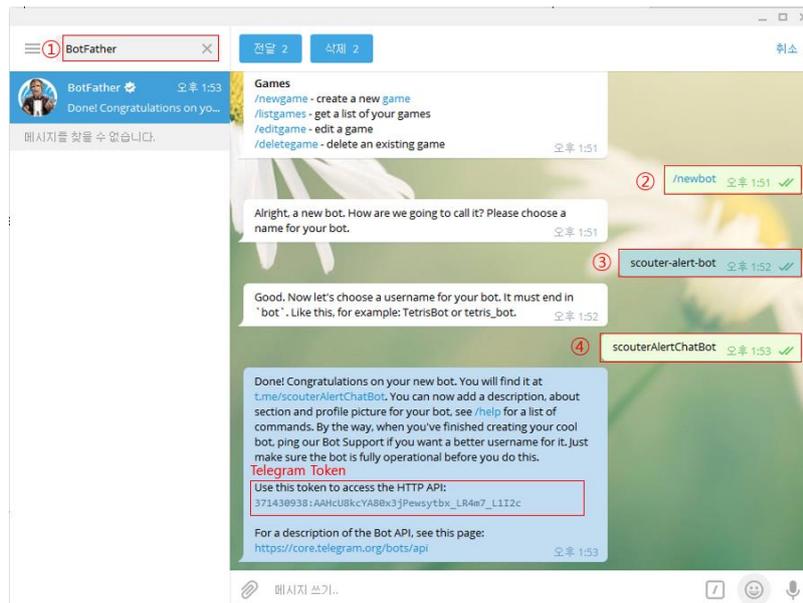
- Example

```
# External Interface (Telegram)
ext_plugin_telegram_send_alert=true
ext_plugin_telegram_debug=true
ext_plugin_telegram_level=0
ext_plugin_telegram_bot_token=371430938:AAHcU8kcYA80x3jPewsytbx_LR4m7_L1I2c
ext_plugin_telegram_chat_id=@scouterAlertChannel
ext_plugin_elapsed_time_threshold=5000
ext_plugin_gc_time_threshold=5000
ext_plugin_thread_count_threshold=300
```

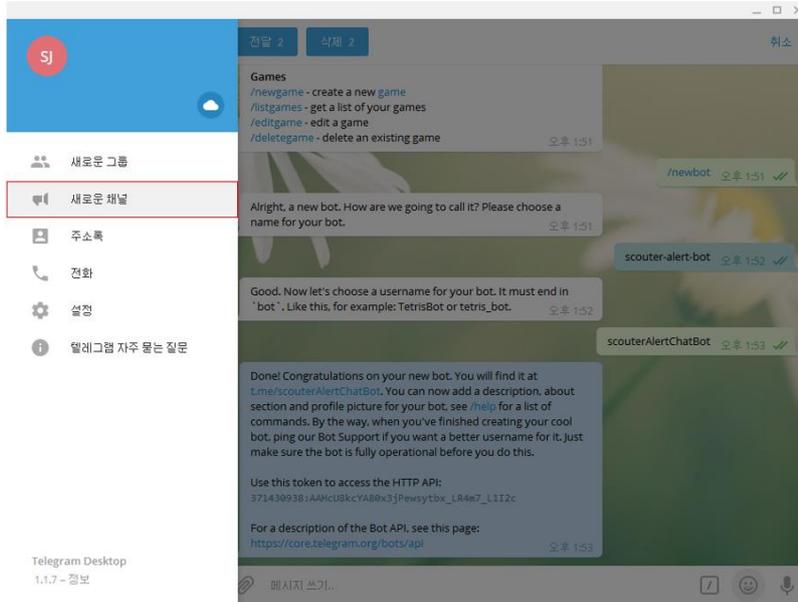
6.3.3 Telegram Application 설정

스카우터와 Telegram 연동 시에 아래와 같이 순서대로 진행하면 Telegram을 통해서 Alert 메시지를 받아 확인 할 수 있다.

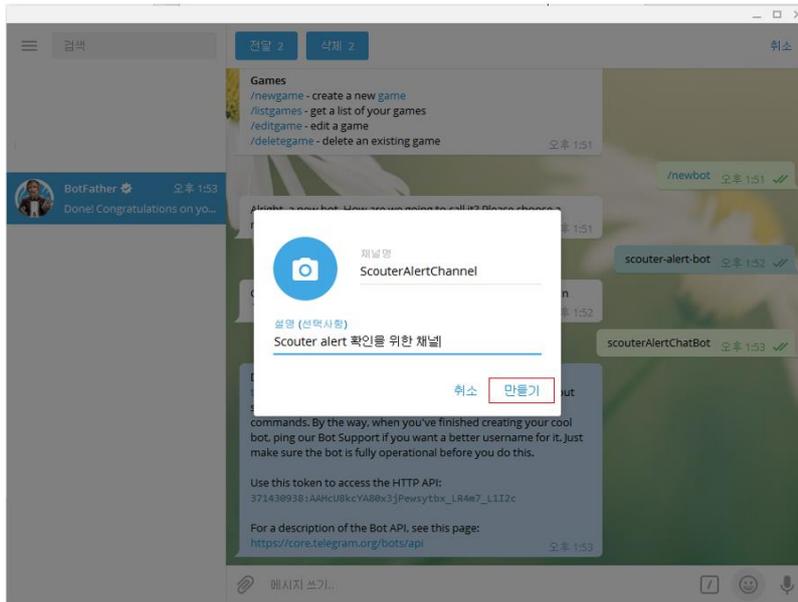
- ① 검색란에 BotFather를 검색하여 사용 가능한 명령어를 확인
- ② /newbot 실행하여 새로운 bot을 생성
- ③ bot 이름을 지정
- ④ bot의 username을 지정



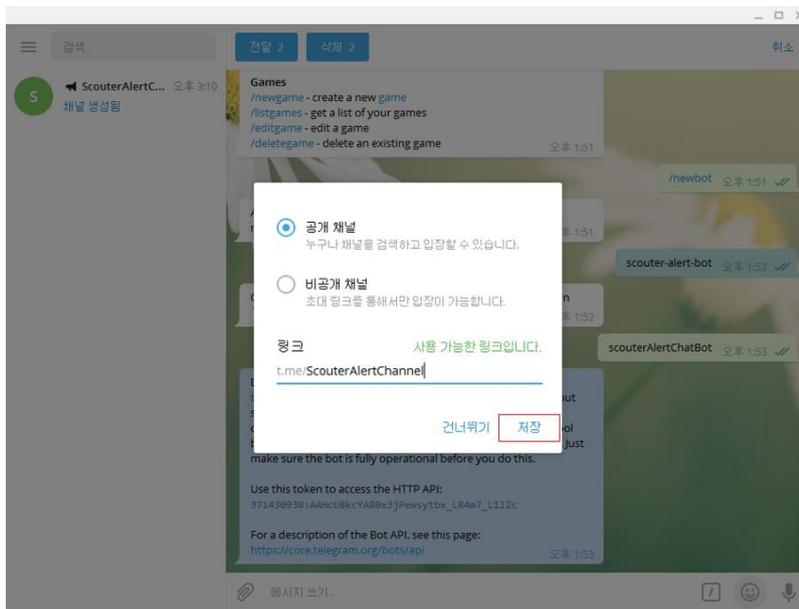
- ⑤ Bot 생성 이후 새로운 채널 생성하여 alert 메시지를 받을 수 있는 채널을 생성



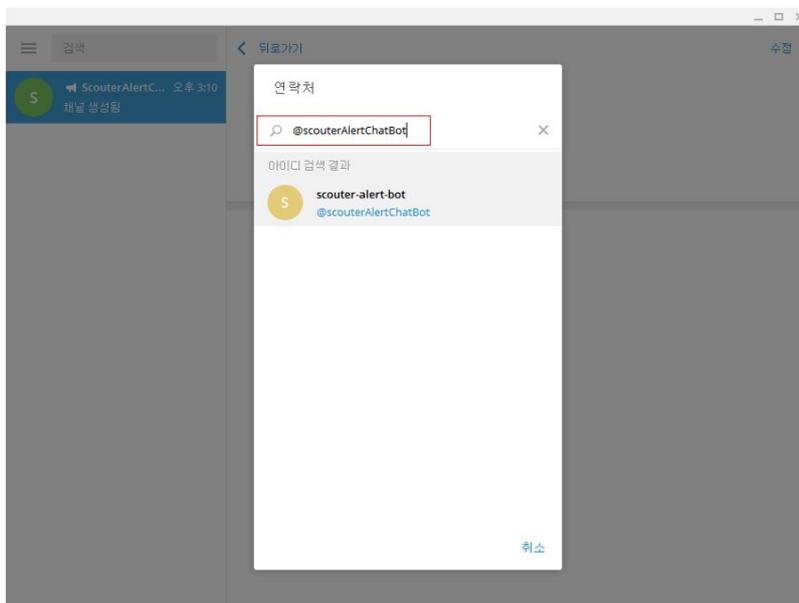
- ⑥ 채널명과 채널에 대한 설명을 작성 (설명은 선택사항)



- ⑦ 공개, 비공개 채널 설정 후 링크에 대한 내용 설정 (비공개 설정 시 채널 ID필요)



- ⑧ 구성원 추가를 통해 BotFather을 통해 생성한 bot의 username을 검색, bot을 통해 alert 메시지를 받을 수 있고 추가적으로 구성원이 필요 하다면 동일하게 구성원을 추가



- ⑨ 스카우터 서버를 기동 시에 출력되는 메시지 아래와 같으며, 설정 파일 수정을 통해 alert에 대한 변경이 가능

