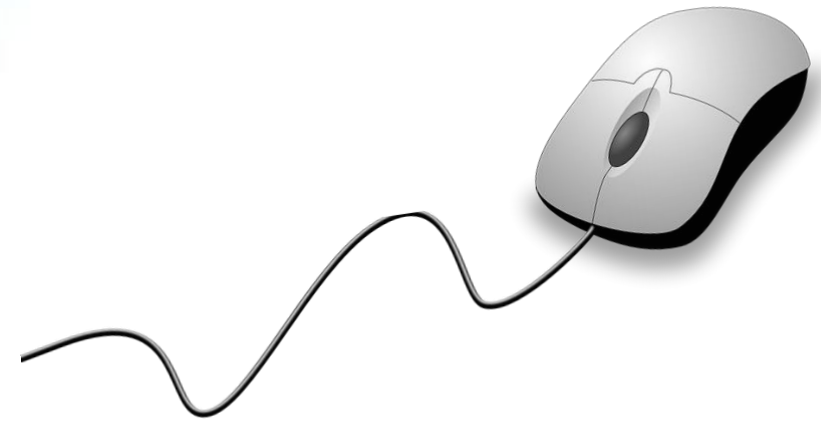


공개SW 솔루션 설치 & 활용 가이드

시스템SW > 자원관리

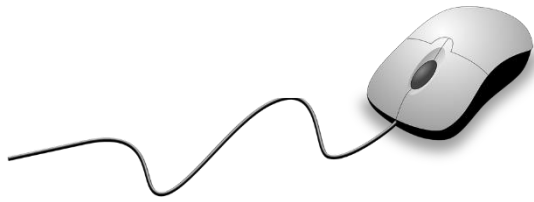


제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide





CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개 및 활용가이드
6. FAQ
7. 용어정리

1. 개요



소개	• 오픈소스 어플리케이션 성능 모니터링 솔루션 (APM)		
주요기능	• Java 기반 어플리케이션에 대한 상세 모니터링을 위한 Trace 정보 제공 • OS 모니터링 텔레그라프 확장 플러그인을 통해 솔루션에 대한 모니터링 기능 제공 • 집킨 플러그인을 통한 다양한 언어로 개발된 어플리케이션에 대한 분산 트레이스 기능 제공		
대분류	• 시스템SW	소분류	• 자원관리
라이선스형태	• Apache License v2.0	사전설치 솔루션	• 없음
		버전	• 2.10 (2020년 10월 기준)
특징	• 오픈소스 기반의 APM • XLOG 스캐터 차트를 통한 직관적인 어플리케이션 모니터링 기능 제공 • 상용 APM를 대체할 수 있는 실시간 어플리케이션 모니터링 • 다양한 플러그인을 통한 기능 확장 및 스크립팅을 통한 사용자 정의 기능 추가 가능		
개발회사/커뮤니티	• Scouter Project		
공식 홈페이지	• https://github.com/scouter-project		



2. 기능요약



- SCOUTER 주요 기능

실시간 대시보드	<ul style="list-style-type: none">• 실시간 성능 모니터링 대시보드• XLOG, TPS, ERROR, JVM GC, Heap 사용량 등
시스템 모니터링	<ul style="list-style-type: none">• OS 자원에 대한 모니터링 (CPU, 메모리, 네트워크)
액티브 서비스 모니터링	<ul style="list-style-type: none">• 실시간 액티브 서비스 모니터링
알림	<ul style="list-style-type: none">• 자원 사용 이상 등에 대한 기본 알림• 스크립트를 통한 사용자 정의 알림
사용자 정의 플러그인	<ul style="list-style-type: none">• 사용자가 스카우터의 기능을 확장할 수 있도록 제공
웹 대시보드	<ul style="list-style-type: none">• 스카우터 클라이언트를 확장하는 3rd 파티 웹 대시보드
텔레그라프 확장	<ul style="list-style-type: none">• 텔레그라프를 통해 다양한 솔루션 모니터링을 통합
집킨 확장	<ul style="list-style-type: none">• 집킨 확장을 통해 다양한 언어로 개발된 어플리케이션 트레이싱

3. 실행환경



- 서버 실행 환경
 - OS 제한 없음
 - Java 1.8 이상
 - 저장공간 500G 이상 권장
- Java 에이전트 실행 환경
 - OS 제한 없음
 - Java 1.6 이상
- Host 에이전트 실행 환경
 - Linux, Windows, Unix 계열
 - Java 1.6 이상



4. 설치 및 실행

세부 목차



4.1 S/W 다운로드

4.2 콜렉터 서버 설치

4.3 클라이언트 실행

4.4 호스트 에이전트 설치

4.5 자바 에이전트 설치



4. 설치 및 실행

4.1 S/W 다운로드



- Scouter 공식 Github의 릴리즈 페이지에서 최신 버전을 다운로드 하여 압축 해제 한다.
 - 가이드 기술 시점 최신 버전 : 2.10

The screenshot shows the GitHub release page for the Scouter project. The browser address bar displays 'github.com/scouter-project/scouter/releases'. The page header includes the GitHub logo, a search bar, and navigation links: 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, the repository name 'scouter-project / scouter' is shown, followed by tabs for '<> Code', 'Issues 117', 'Pull requests', 'Actions', 'Projects', 'Security 1', 'Insights', and 'Settings'. The 'Releases' tab is selected, showing the latest release 'v2.10.0' with commit hash 'b2a9ffc' and a 'Verified' status. A 'Compare' button is also visible. The release details indicate it was released by 'gunlee01' on September 14, with 12 commits to master since the previous release. The 'Features' section lists updates for the collector and agent, including bug fixes and support for HikariDataSource, Java 14, and various tracing capabilities.

Latest release

v2.10.0
b2a9ffc
Verified
Compare

gunlee01 released this on 14 Sep · 12 commits to master since this release

Features

[collector]

- some bugs are fixed.

[agent.java]

- support HikariDataSource metric on tomcat & spring boot. (thx for @ms29seo)
- support java 14
- add traceability of webflux
- add traceability of webflux in kotlin coroutine
- add reactor specific active service list & live stack trace
- add traceability of WebClient
- add traceability of ElasticSearch query
- add traceability of MongoDB query (default false, hook_mongodb_enabled = false)
- add option - xlog_sampling_exclude_patterns
- add option - xlog_consequent_sampling_ignore_patterns



4. 설치 및 실행



4.2 콜렉터 서버 설치

- 적절한 위치에 scouter-all-[version].tar.gz 의 압축을 풀어준다.
Scouter Server를 실행한다
- startup.sh 또는 startup.bat 실행 (실행 중지시 shutdown.sh, shutdown.bat을 실행한다.)
- 스카우터 콜렉터는 모니터링 대상들의 성능 정보를 수집하고 저장하는 역할을 하므로 모니터링 대상이 되는 서버들이 접근할 수 있는 네트워크에 위치하여야 한다.

```
[gunlee@GUN ~/demo/scouter/server]
$ ./startup.sh

Scouter
Open Source S/W Performance Monitoring
Scouter version 1.7.1

[gunlee@GUN ~/demo/scouter/server]
$
```



4. 설치 및 실행



4.2 콜렉터 서버 설치

- netstat 명령어를 통해 scouter가 어떤 port를 사용하고 있는지 확인한다.
기본적으로 tcp, udp 6100 포트를 사용한다.
- 따라서 방화벽이 있다면 tcp/udp 6100 포트를 열어야 한다.

```
[gunlee@GUN ~/demo/scouter/server]
$ netstat -an | grep 6100
tcp46      0      0 *.6100          *.*              LISTEN
udp46      0      0 *.6100          *.*
```



4. 설치 및 실행



4.2 콜렉터 서버 설치

- 콜렉터 서버의 주요 옵션들은 다음과 같으며 아래 경로에서 수정이 가능하다.
 - 콜렉터 서버 설치 위치 하위 `conf/scouter.conf`
- `net_udp_listen_port` : udp 포트 (기본값: 6100)
- `net_tcp_listen_port` : tcp 포트 (기본값: 6100)
- `db_dir` : 콜렉터로 수집된 성능 메트릭, XLOG의 저장 경로 (기본값: `./database`)
- `log_dir`: 콜렉터 로그 저장 경로 (기본값: `./logs`)
- 위 값들을 변경하면 콜렉터 서버를 재시작 해야 한다.








4. 설치 및 실행



4.3 Scouter 클라이언트 실행

- Scouter 릴리즈 페이지에서 본인 PC에 맞는 클라이언트를 다운로드 한다.
 - Windows와 맥OSX, 리눅스용 클라이언트를 제공한다.

▼ Assets 8

 scouter-all-2.10.0.tar.gz	55.4 MB
 scouter-min-2.10.0.tar.gz	21.2 MB
 scouter.client.product-linux.gtk.x86_64.tar.gz	41.8 MB
 scouter.client.product-macosx.cocoa.x86_64.tar.gz	41.6 MB
 scouter.client.product-win32.win32.x86_64.zip	41.9 MB
 scouter.client.product-win32.win32.x86.zip	41.9 MB
 Source code (zip)	
 Source code (tar.gz)	



4. 설치 및 실행



4.3 Scouter 클라이언트 실행

- 다운로드 받은 클라이언트 파일을 압축해제하고 실행한다.
- 접속할 스카우터서버(컬렉터)의 IP나 도메인 그리고 ID/PW를 입력한다.
 - 초기 설치시 ID/PW는 admin/admin 이다.

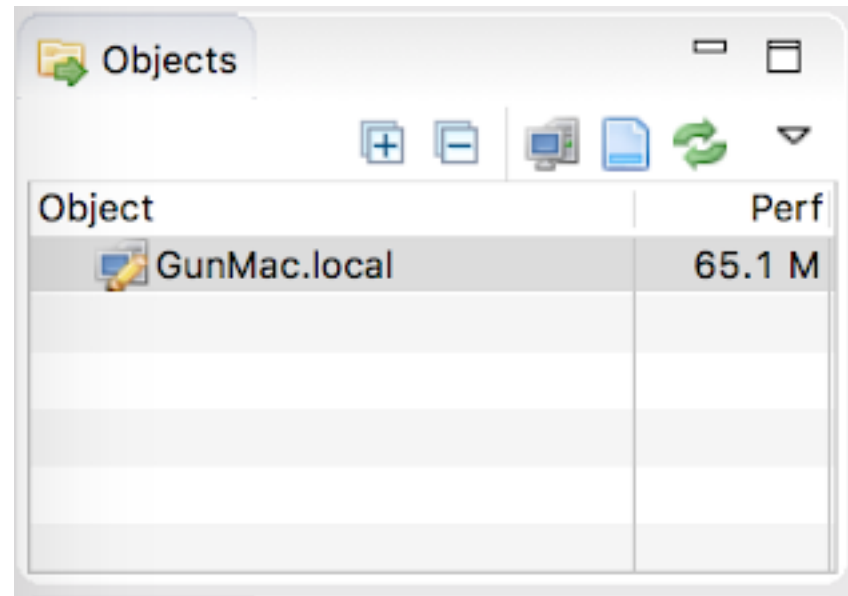


4. 설치 및 실행



4.3 Scouter 클라이언트 실행

- 콜렉터에 클라이언트가 접속되면 좌측 상단의 오브젝트뷰에서 접속한 콜렉터 서버 하나가 있는것을 확인할 수 있다. 이후 에이전트를 실행할때마다 이 콜렉터 서버 하위 모니터링 대상에 나타나게 된다.
- 클라이언트를 통해 정상적으로 콜렉터에 접속이 되면 이제 에이전트를 설치한다.



4. 설치 및 실행



4.3 호스트 에이전트 설치

- 호스트 에이전트는 스카우터 압축 해제한 위치의 ./agent.host 디렉토리에 위치한다.
- Host Agent 실행 전에 먼저 Collector 접속 설정이 필요하다.
- 설정 파일에 Collector의 IP와 Port 정보를 설정한다.
 - 설정 파일(default) : [scouter-dir]/agent.host/conf/scouter.conf
- net_collector_ip에는 Collector 접속 IP나 도메인을 기입한다.
 - 예제 그림에서와 같이 127.0.0.1을 넣지 않고 실제 콜렉터가 설치된 IP를 넣는다.

```
### scouter host configuration sample
net_collector_ip=127.0.0.1
net_collector_udp_port=6100
net_collector_tcp_port=6100
```



4. 설치 및 실행



4.3 호스트 에이전트 설치

- 호스트 에이전트를 실행한다.
- ./host.sh(또는 host.bat)
- 정지할 경우는 ./stop.sh (또는 stop.bat)를 수행한다.

```
[gunlee@GUN ~/demo/scouter/agent.host]
$ ./host.sh

Scouter
Open Source S/W Performance Monitoring
Scouter version 1.7.1

[gunlee@GUN ~/demo/scouter/agent.host]
$ █
```

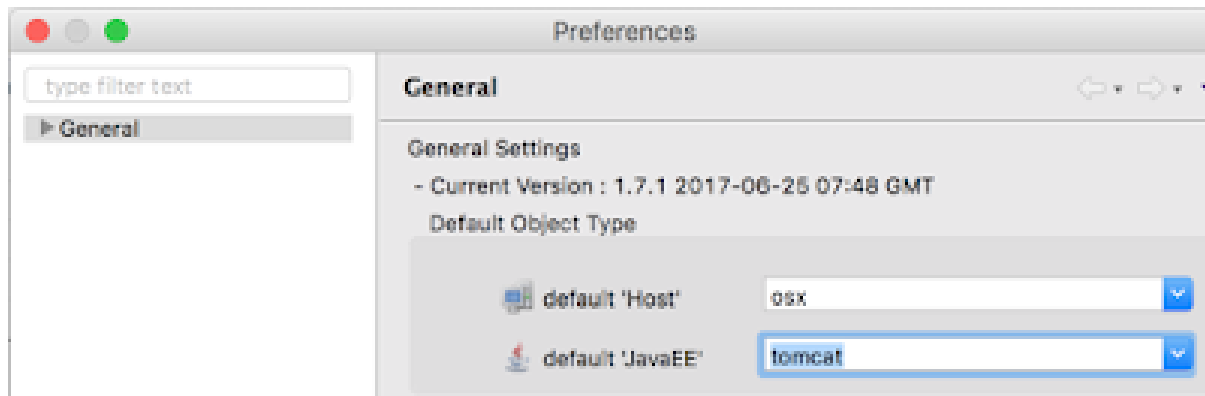
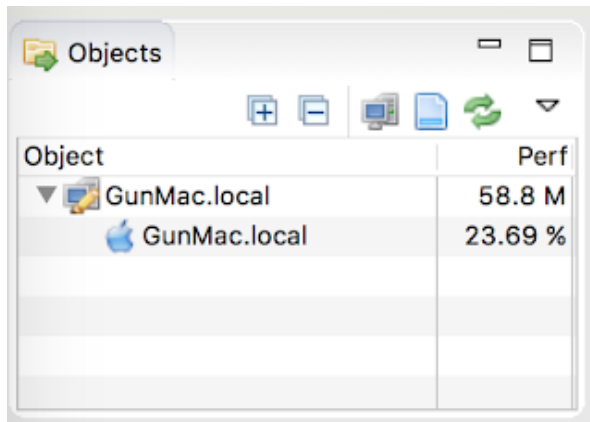


4. 설치 및 실행



4.3 호스트 에이전트 설치

- Host Agent가 잘 실행되었다면, 앞에서 실행한 Scouter Client에서 확인이 가능하다.
- 예제에서는 OSX에서 테스트 중이라 Mac 아이콘이 나온다.
- 이쯤에서 Default Object Type을 맞춰주는 것이 좋은데, Client의 메뉴중 환경설정(Preferences) 메뉴로 들어갑서 설정한다. 예제는 Mac에서 진행하는 중이므로, default Host를 OSX 그리고 default JavaEE를 tomcat으로 설정하였다.

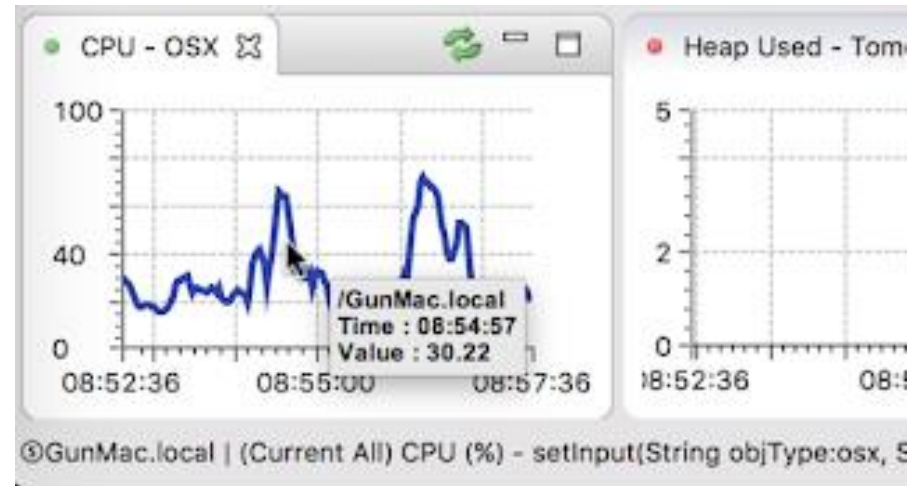
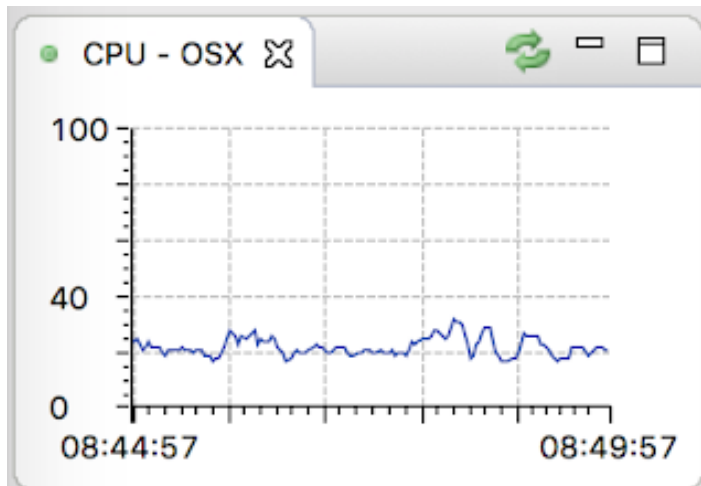


4. 설치 및 실행



4.3 호스트 에이전트 설치

- "Default Object Type"을 적절히 설정한 후 저장하면 설정한 type에 맞게 Perspective가 재구성 된다.
(reset은 기본 Perspective에서만 가능)
- 그러면 이제 "CPU-OSX" View에서 CPU가 모니터링 되는 것을 확인할 수 있다.
- 아래 View에서 그래프를 클릭하게 되면 시간 및 값이 보여지게 된다.
- 그리고 Status Bar에는 선택한 View에 대한 상세 정보가 나타난다.



4. 설치 및 실행



4.4 Java 에이전트 설치

- Java Agent는 단독으로 실행되는 것이 아니라, 모니터링할 Java Program이 실행될때 attach되어 모니터링을 수행한다.
- 따라서 실행하는 Java Program에서 scouter agent를 인식할 수 있도록 명령행 옵션을 설정해야 한다.
 - "-javaagent:[somewhere]/scouter.agent.jar"
- 그리고 하나의 VM에 여러개의 java 프로그램을 실행한다면 보통 scouter agent의 환경설정 파일경로와 이름을 지정해주게 된다.
- 설정 파일 경로 지정
 - "-Dscouter.config=/[somewhere]/mySampleConfig.conf"
 - 설정 파일 경로를 지정하지 않은 경우 기본값
 - [directory of scouter.agent.jar]/conf/scouter.conf
- 모니터링 대상(object) 이름 지정
 - "-Dobj_name=scouterWebDemo1"
 - 기본값 : java1 또는 tomcat1

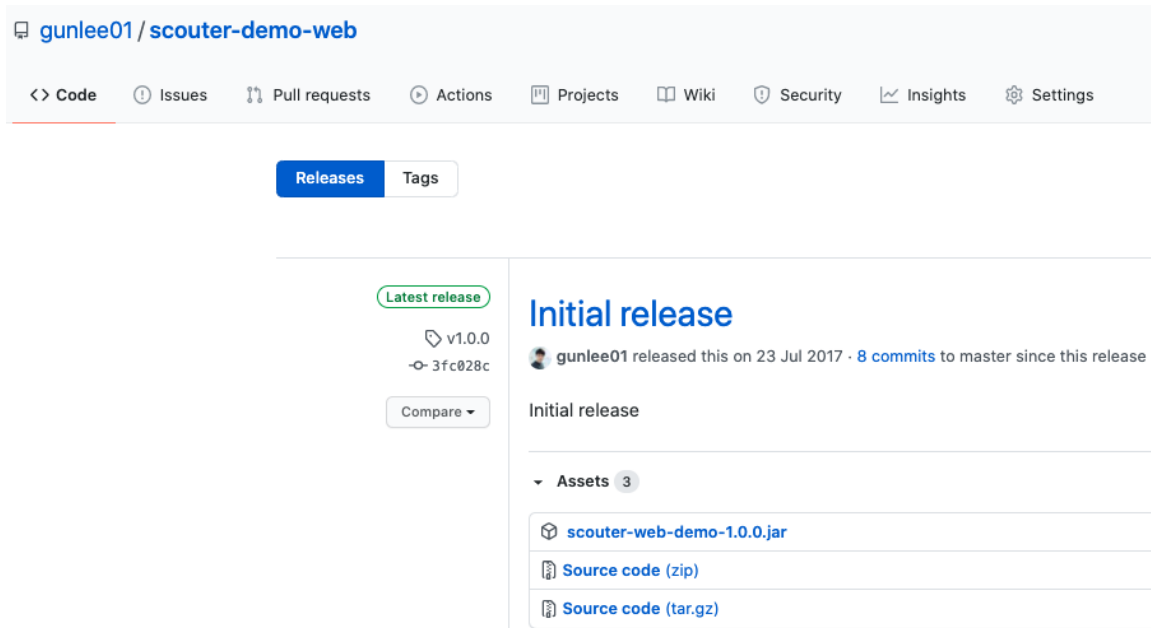
4. 설치 및 실행



4.4 Java 에이전트 설치

- 데모를 위해 Spring boot로 작성된 간단한 샘플 프로그램에 java agent를 실행해 본다.
- 샘플 프로그램은 다음 경로에서 다운로드 받을 수 있다.
 - Sample web application for Scouter agent demo

<https://github.com/gunlee01/scouter-demo-web/releases/tag/v1.0.0>



gunlee01 / scouter-demo-web

<> Code ⓘ Issues 🔗 Pull requests 🔄 Actions 📁 Projects 📖 Wiki ⓘ Security 📈 Insights ⚙ Settings

Releases Tags

Latest release

v1.0.0
3fc028c

Compare

Initial release

gunlee01 released this on 23 Jul 2017 · 8 commits to master since this release

Initial release

Assets 3

- scouter-web-demo-1.0.0.jar
- Source code (zip)
- Source code (tar.gz)



4. 설치 및 실행



4.4 Java 에이전트 설치

- 제공되는 어플리케이션은 `java -jar scouter-web-demo-[versions].jar` 명령어로 실행할 수 있으며, 8080 포트로 웹 어플리케이션이 시작된다.
- 여기서는 `scouter` 명령행 옵션을 포함하여 `jar` 파일을 실행한다.

```
java -javaagent:/demo/scouter/agent.java/scouter.agent.jar -Dobj_name=scouterWebDemo1 -
DScouter.config=/demo/scouter/sample/conf/scouterWebDemo1.conf -jar ./scouter-web-demo-
1.0.0.jar
```

```
[gunlee@GUN ~/demo/scouter/sample]
$ java -javaagent:/Users/gunlee/demo/scouter/agent.java/scouter.agent.jar -Dobj_name=scouterWebDemo
-DScouter.config=/Users/gunlee/demo/scouter/sample/conf/scouterWebDemo.conf -jar ./scouter-web-dem
o-1.0.0.jar
objc[59822]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8
.0_121.jdk/Contents/Home/bin/java (0x10545e4c0) and /Library/Java/JavaVirtualMachines/jdk1.8.0_121.
jdk/Contents/Home/jre/lib/libinstrument.dylib (0x1055804e0). One of the two will be used. Which one
is undefined.

SCOUTER
Open Source S/W Performance Monitoring
Scouter version 1.7.1

20170723 16:44:00 [SCOUTER] Version 1.7.1 2017-06-25 07:48 GMT
```

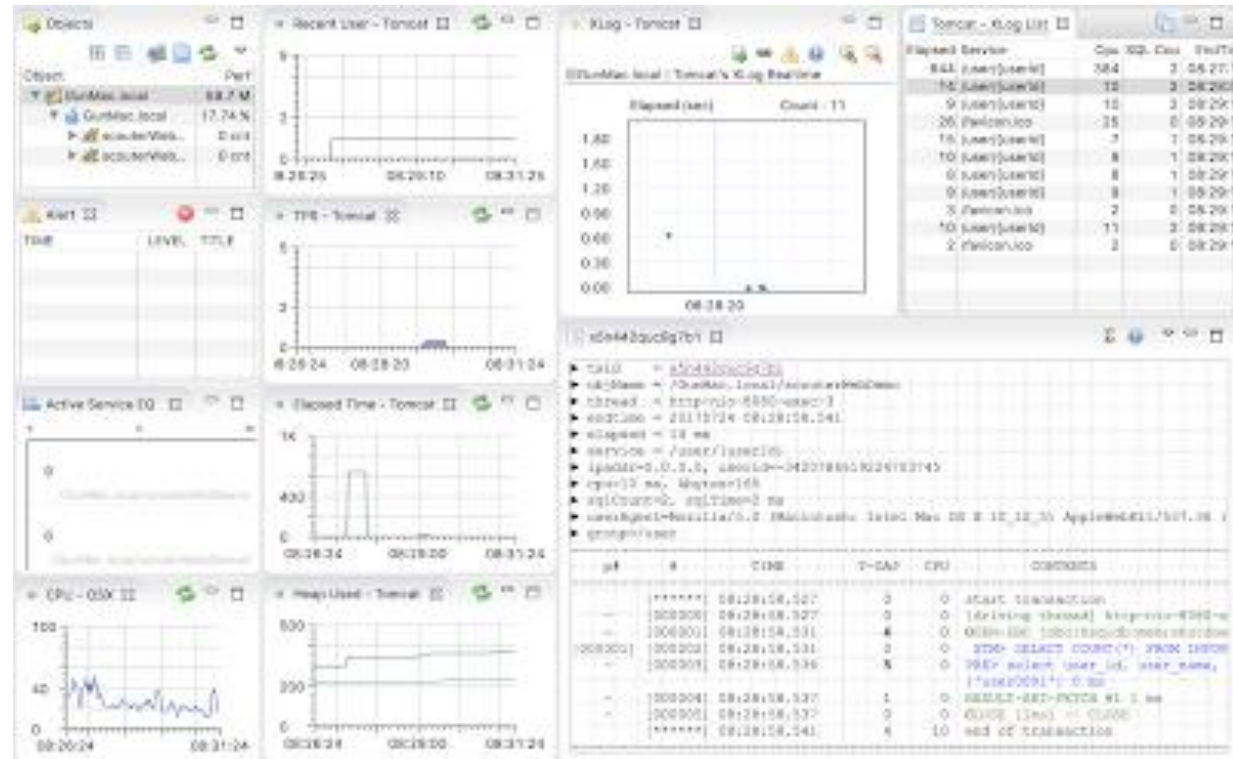
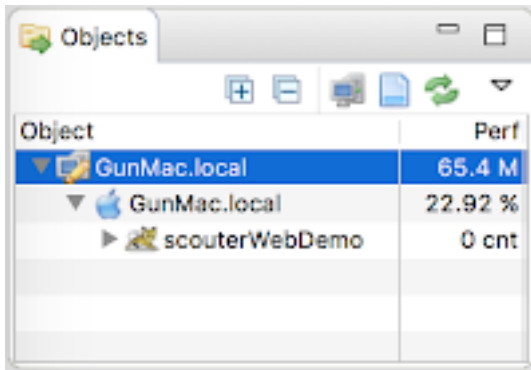


4. 설치 및 실행

4.4 Java 에이전트 설치



- 정상적으로 scouter agent가 포함된 상태로 spring boot 어플리케이션이 실행되면, object view에서 이를 확인할 수 있다.
- 브라우저를 열고, <http://localhost:8080/user/user0001> 을 호출해보면 Scouter client를 통해 요청의 상세 정보를 확인할 수 있다.



5. 기능소개 및 활용가이드



세부 목차

5.1 차트의 종류

5.2 실시간 대시보드

5.3 XLOG 및 액티브 서비스 모니터링



5. 기능소개 및 활용가이드



5.1 차트의 종류

- Scouter의 performance counter는 8가지 종류의 차트를 제공하며, 처음에 열리는 차트는 실시간 성능 정보를 제공하는 "Realtime-Current" 차트이다.
- Real Time 차트 : 현재 시점까지의 데이터를 보여주며 실시간 갱신되는 차트
 - Current : 최근 5분간의 데이터를 2초 간격으로 갱신(Default)
 - Today : 오늘의 0시 부터 현재까지의 데이터를 보여준다.



5. 기능소개 및 활용가이드

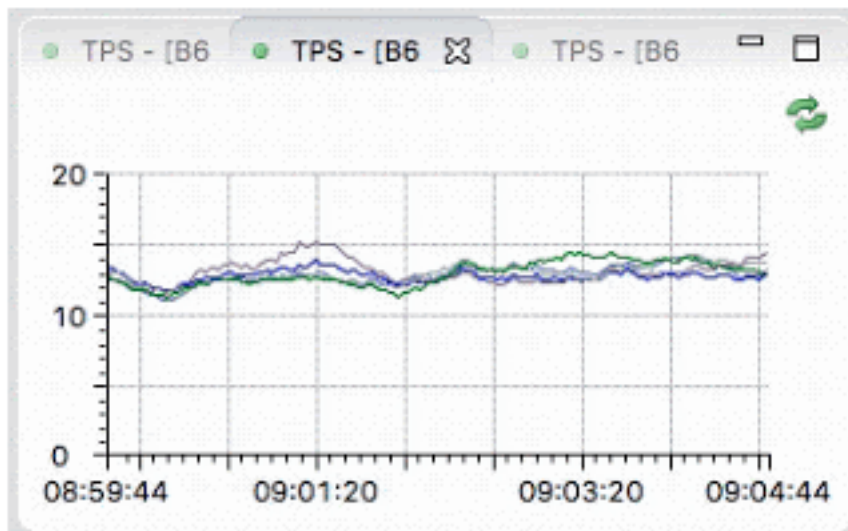


5.1 차트의 종류

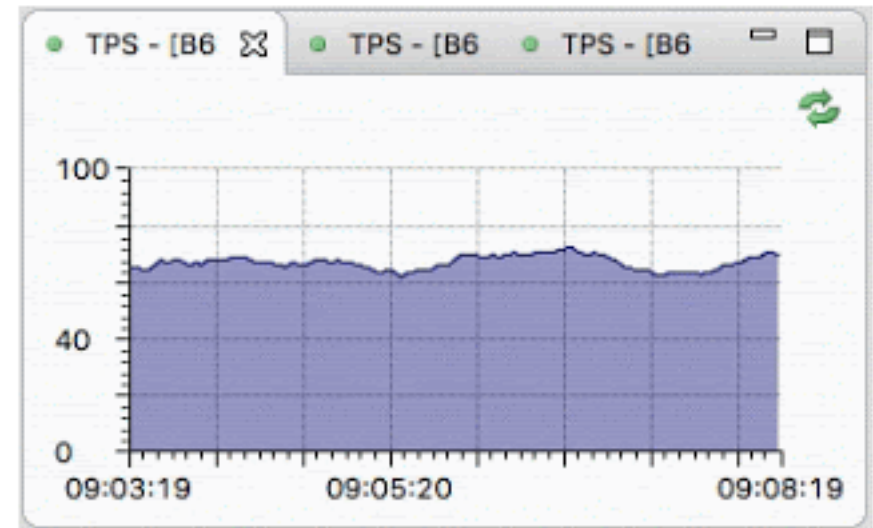
- Current Time 차트

All : (각 인스턴스별 값을 보여준다.)

Total : (모든 인스턴스 값의 합계를 보여준다.)



<그림. Real Time Current>



<그림. Real Time Total>



5. 기능소개 및 활용가이드

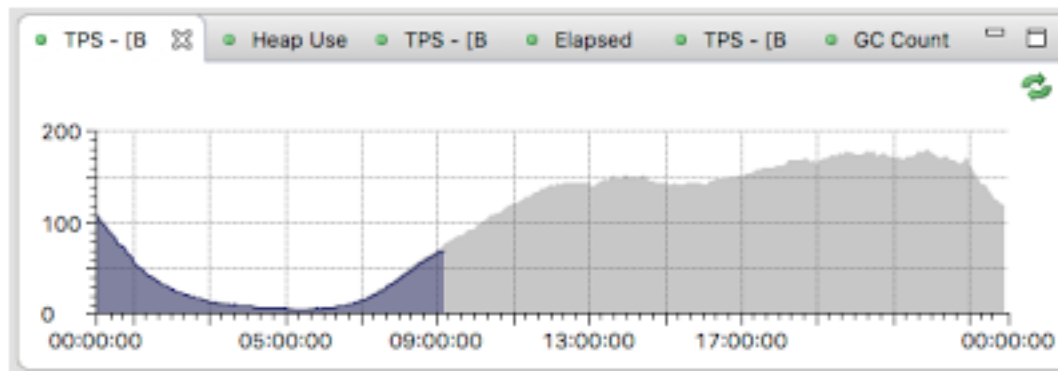


5.1 차트의 종류

- Today 차트

All : (각 인스턴스별 값을 보여준다.)

Total : (모든 인스턴스 값의 합계를 보여준다.)



<그림. Today Total>

"Today Total" 차트에서 어제 데이터는 회색으로 표시됩니다.



5. 기능소개 및 활용가이드



5.1 차트의 종류

- History 차트 : 과거 특정 시점의 데이터를 조회한다.
 - Daily : 과거 특정 시점의 1일~수개월 데이터를 5분 통계로 보여준다.
 - All : (각 인스턴스별 값을 보여준다.)
 - Total : (모든 인스턴스 값의 합계를 보여준다.)
 - Past : 과거 특정 시점의 5분~4시간의 데이터를 2초 정밀도로 보여준다.
 - All : (각 인스턴스별 값을 보여준다.)
 - Total : (모든 인스턴스 값의 합계를 보여준다.)

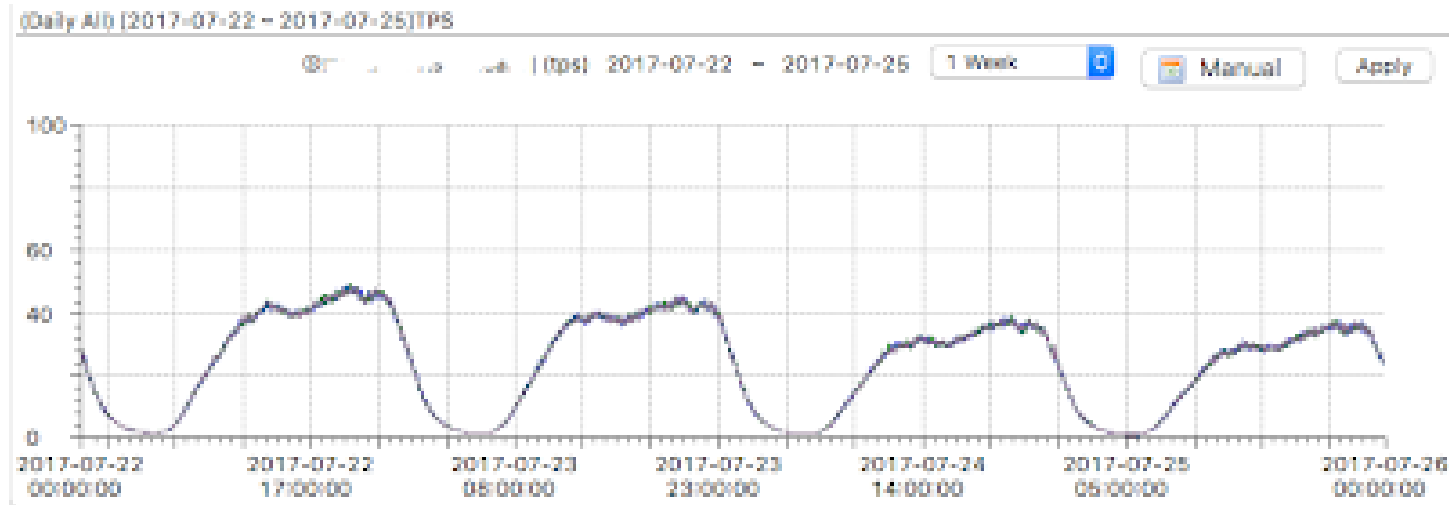


5. 기능소개 및 활용가이드



5.1 차트의 종류

- Daily 차트 - 과거 특정 시점의 1일~수개월 데이터를 5분 통계로 보여준다.

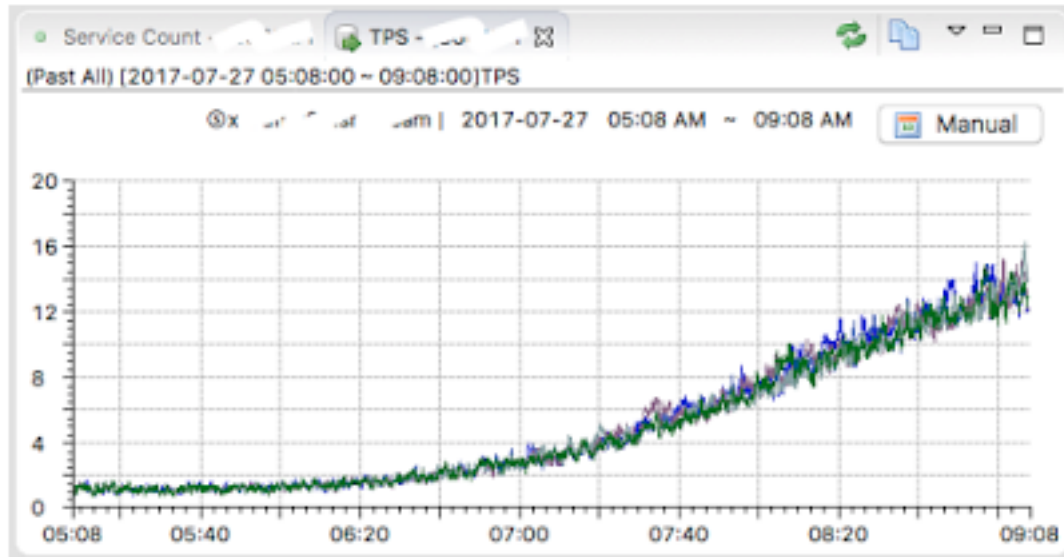


5. 기능소개 및 활용가이드



5.1 차트의 종류

- Past 차트 - 과거 특정 시점의 5분~4시간의 데이터를 2초 정밀도로 보여준다.



<그림. Past All>



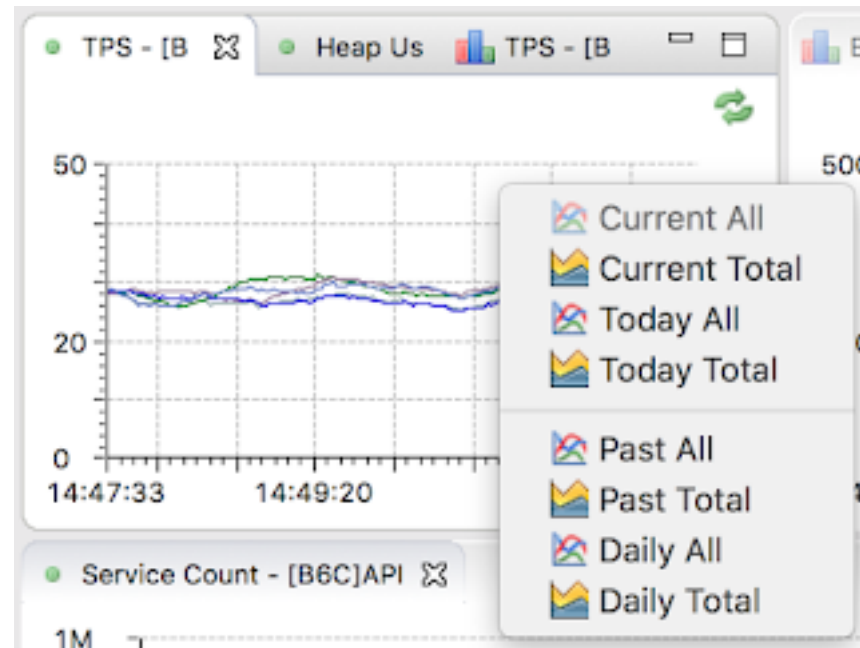
5. 기능소개 및 활용가이드



5.1 차트의 종류

- 차트 다루기

하나의 차트에서 다른 유형의 차트를 열 수 있다.



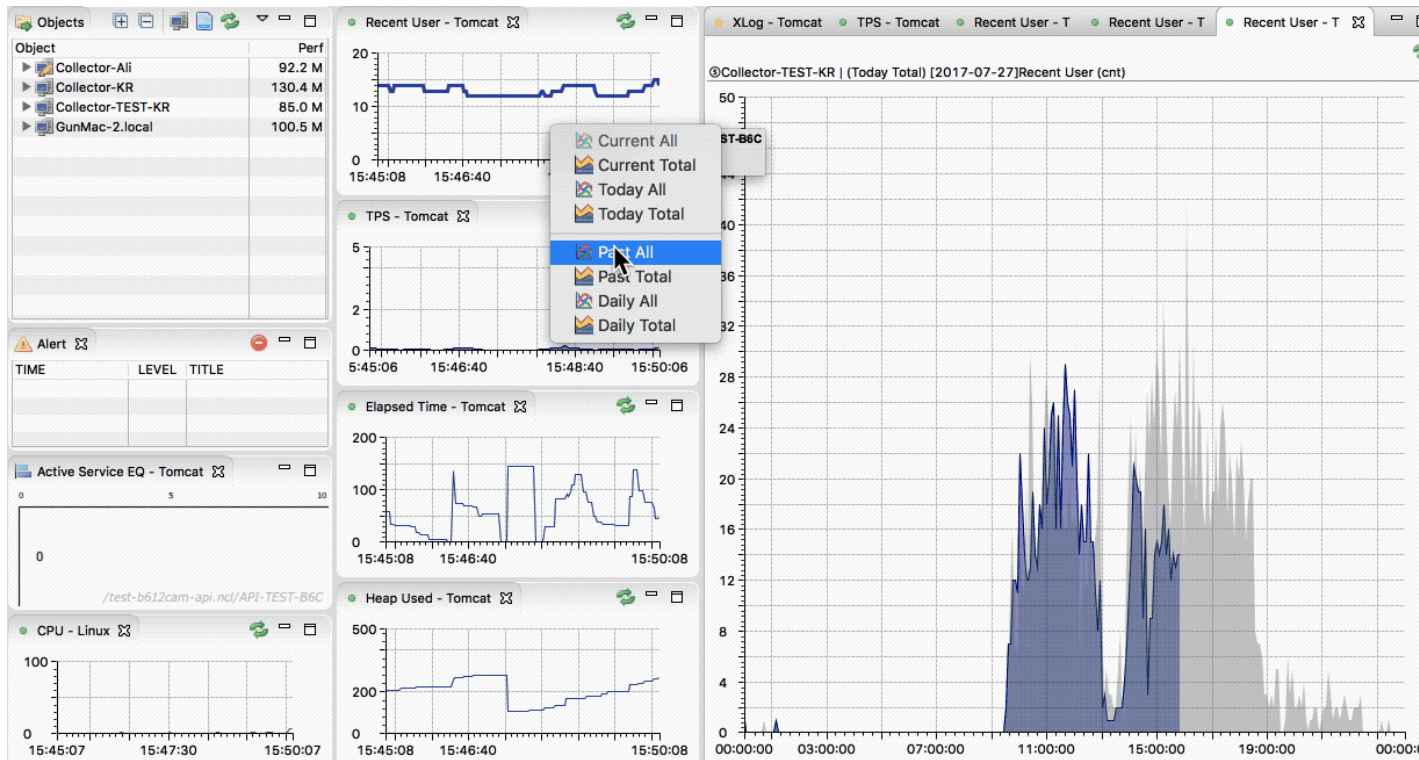
5. 기능소개 및 활용가이드



5.1 차트의 종류

• 차트 배열하기

- 차트는 드래그 드롭하여 손쉽게 대시보드를 구성할 수 있다.
- 또한 재구성한 퍼스펙티브는 새이름으로 저장하여 퍼스펙티브간 전환하여 사용할 수 있다.

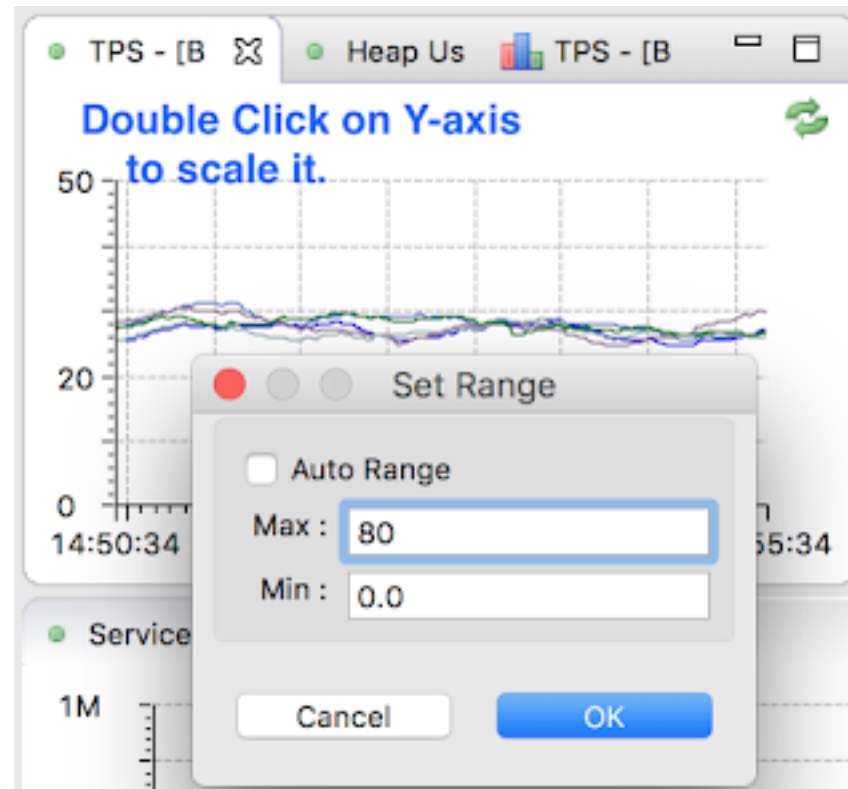


5. 기능소개 및 활용가이드



5.1 차트의 종류

- 차트의 라인을 클릭하면 상세 정보를 볼 수 있다.
- 어떤 차트는 Y축을 더블클릭하면 Y축의 높이를 조절할 수 있다.



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

- 실시간 대시보드 오버뷰
- 시스템 상태에 대한 빠른 판단을 위한 실시간 대시보드 기능



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

• 성능 카운터

• 시계열의 성능 메트릭 제공

• Counter View를 Collector의 콘텍스트 메뉴에서 여는 경우, 해당 Object Type의 인스턴스 전체에 대해 하나의 차트로 보여준다.

(예를 들어 TPS 차트를 선택한 경우 동일한 Object Type인 tomcat1, tomcat2, ... tomcatX 에 대한 TPS가 하나의 차트로 보여진다.)

• 각 Object의 콘텍스트 메뉴에서 여는 경우에는 해당 Object에 대한 Counter View만 보여준다.

어떠한 View는 Collector 메뉴에서만 열 수 있거나 혹은 Object 메뉴에서만 열 수 있다.

XLog 및 Active Service EQ등은 Collector 메뉴에서만 열 수 있다.

Heap Total Usage, Sys/User CPU등은 Object 메뉴에서만 열 수 있다.

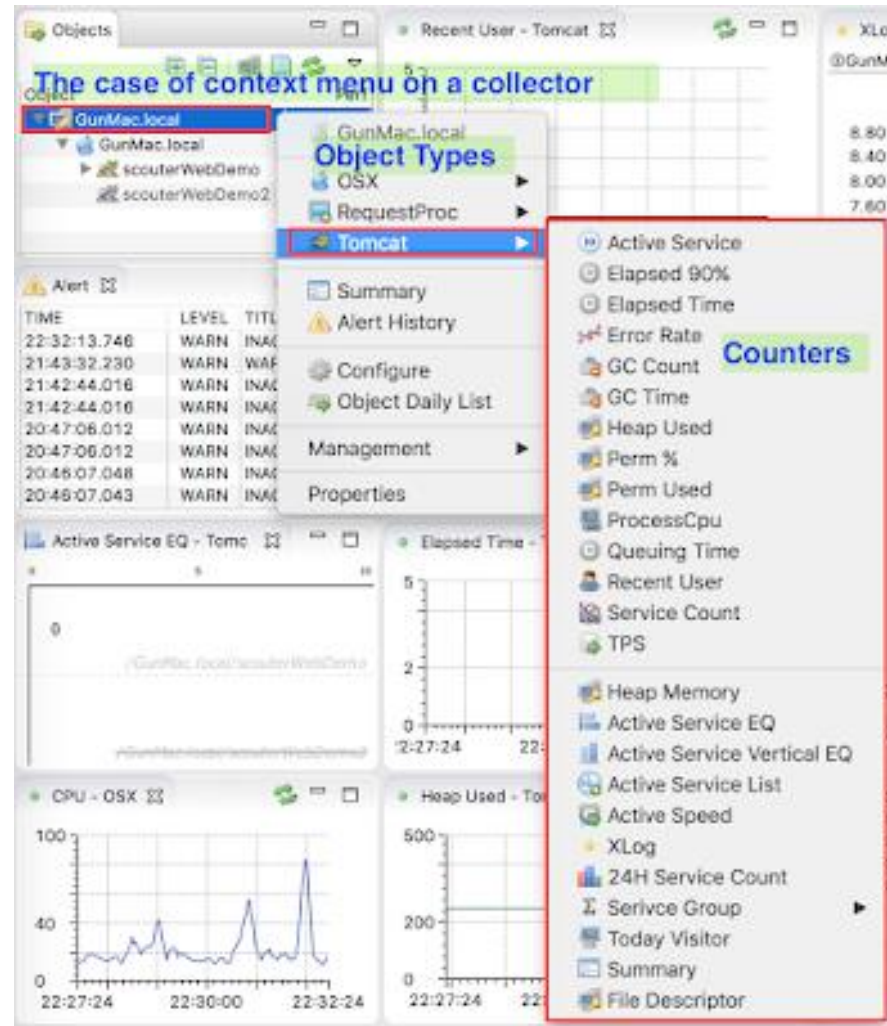


5. 기능소개 및 활용가이드



5.2 실시간 대시보드

• 성능 카운터 – Java 의 성능 카운터 목록



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

• 성능 카운터

• Java 성능 카운터 (1/3)

- Active Service

액티브 서비스 개수

- Elapsed 90%

90% 응답 시간

- Elapsed

평균 응답 시간

- Error Rate

에러율

- GC Count

JVM GC 수행 회수

- GC Time

JVM GC 수행 시간



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

• 성능 카운터

• Java 성능 카운터 (2/3)

- Heap Used

힉메모리 사용량

- Perm %

퍼메모리 사용율

- Perm Used

퍼메모리 사용량

- Process CPU

JVM 프로세스의 CPU 사용율

- Queuing Time

웹서버에서 WAS까지 지연 시간

- Recent User

최근 5분간의 접속 사용자

- Service Count

- TPS



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

• 성능 카운터

• Java 성능 카운터 (3/3)

- Service Count

단위시간당 서비스 요청 수

- TPS

초당 요청량

- 24H Service Count

당일 시간당 서비스 요청수

- Today Visitor

당일 방문자 수

- Summary

통계 정보

- File Descriptor

FD 개수

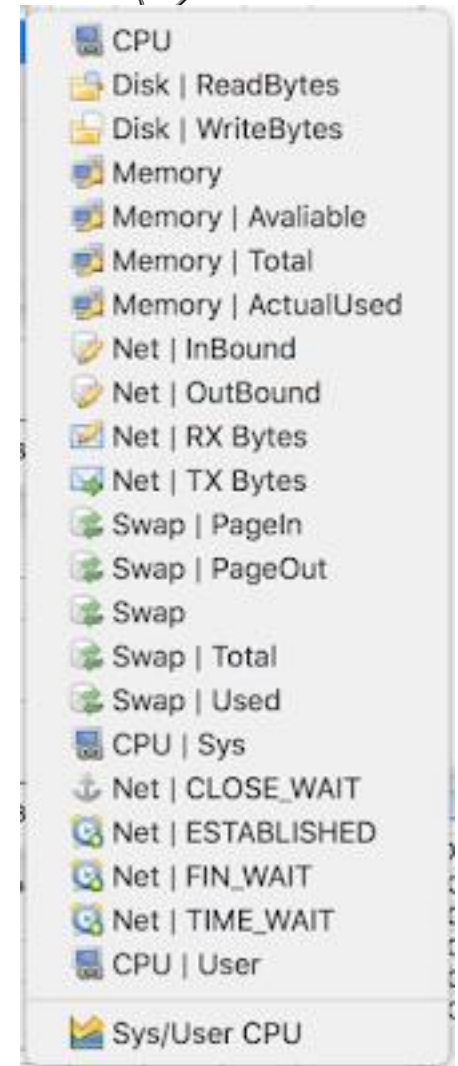


5. 기능소개 및 활용가이드



5.2 실시간 대시보드

- 성능 카운터
- Host 성능 카운터
 - Host 성능 카운터는 이름으로 알 수 있으므로 따로 설명하지 않는다.



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

- **오브젝트 리퀘스트**
- 모니터링 대상의 특정 오브젝트에 요청을 보내 특정 성능 정보를 얻어오는 기능

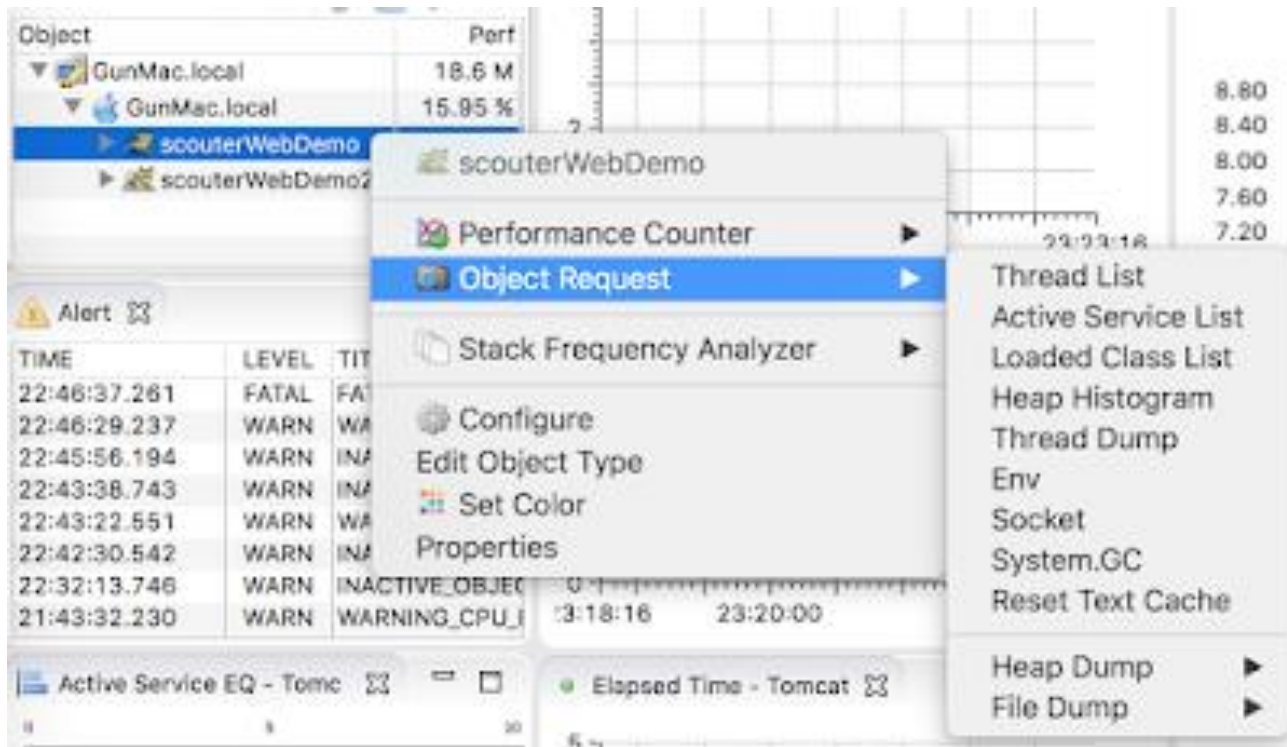


5. 기능소개 및 활용가이드



5.2 실시간 대시보드

- 오브젝트 리퀘스트
- Java 오브젝트 리퀘스트



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

• 오브젝트 리퀘스트

• Java 오브젝트 리퀘스트 (1/2)

- Thread List

Thread 목록을 요청한다. 간단한 스레드 정보를 확인할 수 있다.

- Active Service List

현재 수행중인 서비스 목록을 요청한다.

- Loaded Class List

JVM에 로드된 클래스 목록을 요청한다.

- Heap Histogram

힙 메모리에 대한 현재 시점의 히스토그램을 요청한다.

- Thread Dump

JVM에서 제공하는 스레드 덤프를 요청한다.

특정시점 문제가 있는 스레드를 분석하기 위해 사용된다.



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

• 오브젝트 리퀘스트

• Java 오브젝트 리퀘스트 (2/2)

- Env

JVM의 환경 변수 정보를 요청한다.

- Socket

현재 열려있는 소켓 정보를 요청한다.

- System.GC

강제로 GC를 수행하도록 요청한다.

- Reset Test Cache

스카우터의 텍스트 캐시를 리셋한다.

- Heap Dump

힙덤프를 요청한다. 시스템 부하가 발생하므로 장애 상황에서만 사용하여야 한다.

- File Dump

스레드 목록등을 파일로 덤프한다.

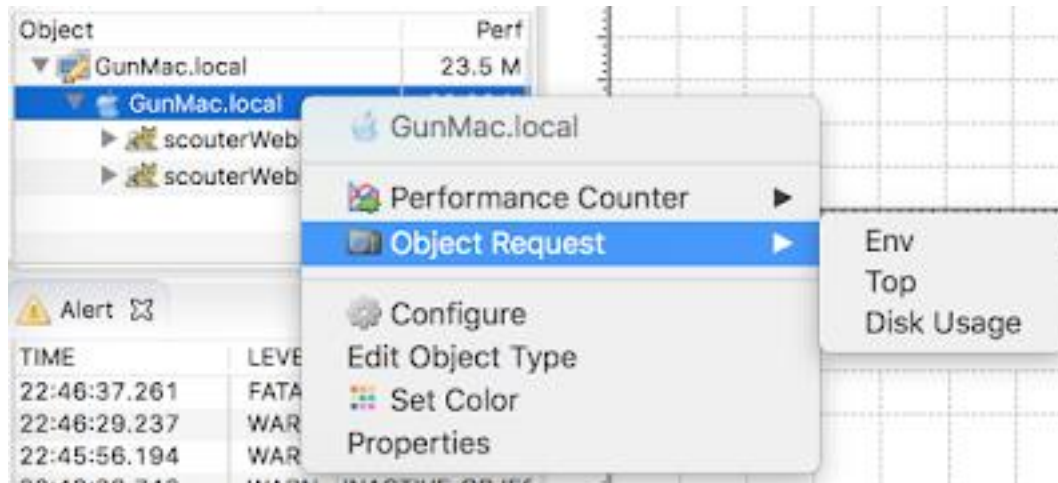


5. 기능소개 및 활용가이드



5.2 실시간 대시보드

- 오브젝트 리퀘스트
- Host 오브젝트 리퀘스트



5. 기능소개 및 활용가이드



5.2 실시간 대시보드

- **오브젝트 리퀘스트**
- Host 오브젝트 리퀘스트
 - Env : OS의 환경 변수를 조회한다.
 - Disk Usage : 디스크 사용량을 조회한다.
 - Top : 요청 시점의 Process 별 CPU 사용율을 조회한다.

PID	USER	CPU%	MEM	TIME	NAME
55786	gunlee	8.9%	378.8M	59:45.801	/Users/gunlee/demo/scouter/scouter.client.app/Conte
58119	gunlee	1.2%	130.8M	24:21.213	java:scouter.boot.Boot
37040	gunlee	1.0%	1.6G	41:57.339	/Applications/IntelliJ IDEA.app/Contents/MacOS/idea
82933	gunlee	0.7%	262.6M	57:29.471	/Applications/Google Chrome.app/Contents/Versions/5
74057	gunlee	0.6%	348.2M	40:19.590	/Applications/Google Chrome.app/Contents/Versions/5
58086	gunlee	0.6%	133.1M	24:12.486	java:scouter.boot.Boot
38090	gunlee	0.4%	73.1M	42:22.740	/Applications/CheatSheet.app/Contents/MacOS/CheatS
76612	gunlee	0.4%	49.5M	43:18.879	/Applications/Scroll Reverser 2.app/Contents/MacOS/

5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

• XLOG

- 시계열의 성능 메트릭 제공
- Counter View를 Collector의 콘텍스트 메뉴에서 여는 경우, 해당 Object Type의 인스턴스 전체에 대해 하나의 차트로 보여준다.
(예를 들어 TPS 차트를 선택한 경우 동일한 Object Type인 tomcat1, tomcat2, ... tomcatX 에 대한 TPS가 하나의 차트로 보여진다.)
- 각 Object의 콘텍스트 메뉴에서 여는 경우에는 해당 Object에 대한 Counter View만 보여준다.
어떠한 View는 Collector 메뉴에서만 열 수 있거나 혹은 Object 메뉴에서만 열 수 있다.
XLog 및 Active Service EQ등은 Collector 메뉴에서만 열 수 있다.
Heap Total Usage, Sys/User CPU등은 Object 메뉴에서만 열 수 있다.

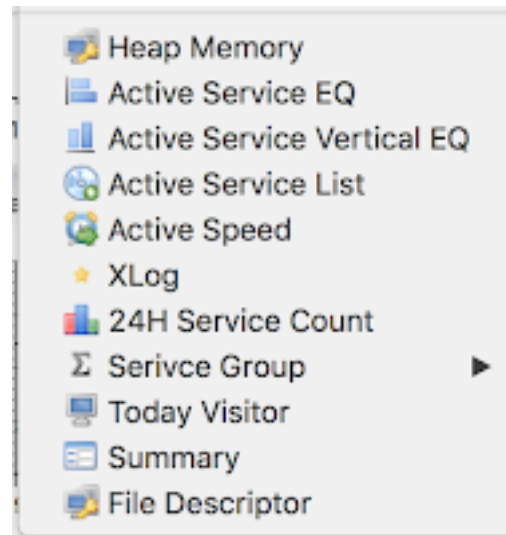


5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- XLOG와 액티브 서비스 모니터링은 어플리케이션 관점에서 수행되는 코드 수준의 가시성을 제공하며 이를 통해 즉시 문제 발견 및 해결이 가능하도록 도와주는 기능이다.



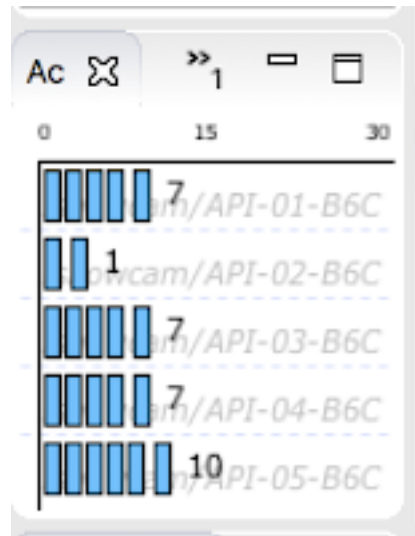
5. 기능소개 및 활용가이드



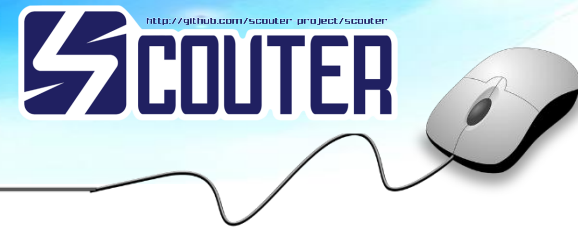
5.3 XLOG 및 액티브 서비스 모니터링

- Active Service EQ

- 현재 시점에 얼마나 많은 서비스가 동시에 수행되는지를 한 눈에 파악하기 위한 이퀄라이저 형태의 차트이다. 이 차트를 더블 클릭 하게되면 현재 수행되는 서비스를 "Active Service View"에서 목록 형태로 보여지게 된다.



5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- Active Service List
 - Active Service View 에서 실행중인 서비스의 정보를 알고 싶다면 상세히 보고싶은 인스턴스를 더블 클릭하면 "Active Service List" 화면이 열리게 된다.
 - 여기서는 수행중인 서비스의 현재까지 진행중인 응답시간등이 보여지게 된다

★ XLog - [B6C]API		Active Service List[/xva... /API-01-B6C]					
Count = 3							
Object	Service	Elapsed	Note	Cpu	IP	State	Name
/xva...	LS...infrastr...	325	service_mem...	202,726		RUNNABLE	defaultExecutor-1
/xva...	/user/sns<POST>	150	/sns/userinfo	1,434,908	10.10.0.163	RUNNABLE	http-bio-8889-ex
/xva...	/user/sns<POST>	128	/sns/userinfo	1,433,879	10.10.0.163	RUNNABLE	http-bio-8889-ex



5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- Active Service View

- 현재 시점에 서버에서 수행중인 서비스 목록 및 이에 대한 상세 정보이다.
- 서비스명 및 수행중인 SQL이나 호출하고 대기중인 원격 Service URL이 보여지게 되며, 코드의 어떤 부분이 수행중인지 알 수 있도록 StackTrace 및 Thread의 Lock 정보등이 제공된다.

The screenshot shows the 'Active Service List' window in SCOUTER. It displays details for a specific thread (ID 0) associated with the service '/user/snsJoin<POST>'. The window includes control buttons (Interrupt, Stop, Suspenc, Resume) and a table of key-value pairs for thread attributes. Below the table is a stack trace showing the current execution path through various Java classes, including SocketInputStream, BufferedInputStream, and HttpParser.

Key	Value
Blocked Time	-1
Lock Name	
Lock Owner Id	-1
Lock Owner Name	
Service Elapsed	138
Service Name	/user/snsJoin<POST>
Service Txid	x7ous3omvumbh7
State	n/a
Thread Cpu Time	777,107
Thread Id	0

```
java.net.SocketInputStream.socketRead0(Native Method)
java.net.SocketInputStream.socketRead(SocketInputStream.java:116)
java.net.SocketInputStream.read(SocketInputStream.java:170)
java.net.SocketInputStream.read(SocketInputStream.java:141)
java.io.BufferedInputStream.fill(BufferedInputStream.java:246)
java.io.BufferedInputStream.read(BufferedInputStream.java:265)
org.apache.commons.httpclient.HttpParser.readRawLine(HttpParser.java:78)
org.apache.commons.httpclient.HttpParser.readLine(HttpParser.java:106)
org.apache.commons.httpclient.HttpConnection.readLine(HttpConnection.java:1116)
```



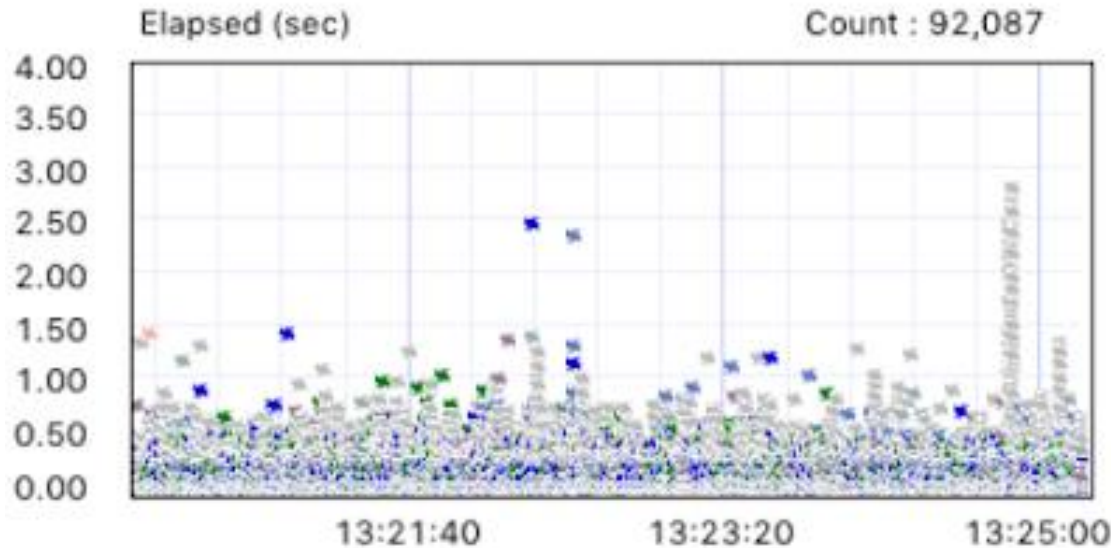
5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- XLog

- 모든 요청에 대해 scatter 차트형태로 보여주며 이를 통해 어플리케이션의 상태를 직관적으로 파악할 수 있다.
- X축은 수행이 완료된 시간을 Y축은 응답시간을 나타낸다.
- Xlog를 선택하게 되면 선택된 목록이 나타나게 된다



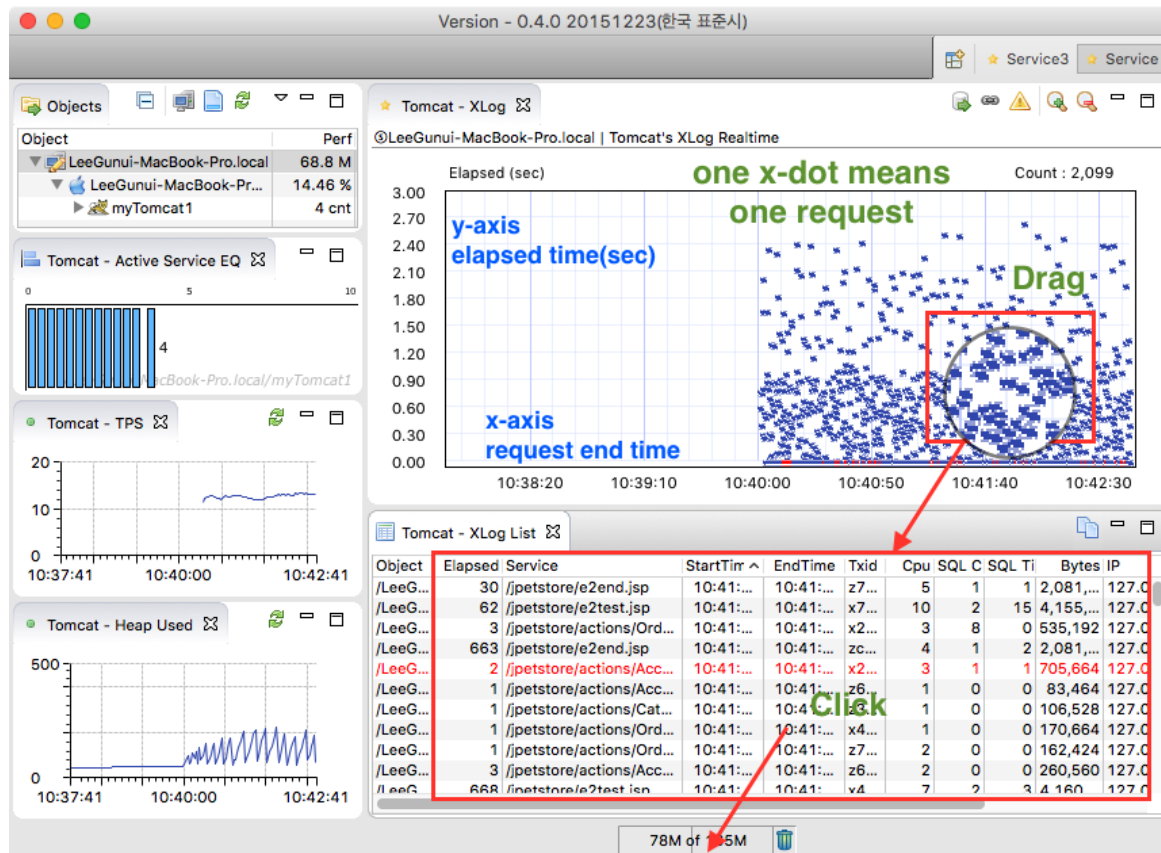
5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- XLog

- 각 점의 색은 각 object 마다 다르며 빨간색은 에러를, 회색은 비동기 thread를 나타낸다. (scouter는 비동기 thread에 대한 연결 추적을 지원한다)



5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- XLog
 - Xlog 목록 뷰이다.
 - 개략적인 요청, 응답시간 등을 알 수 있다.

Tomcat - XLog List										
Object	Elapsed	Service	EndTime	Cpu	SQL Cou	SQL Time	KBytes	IP	Error	Login
/GunMac-2.l...	3	/user/{userId}	13:05:26...	3	2	2	112	0.0.0.0		
/GunMac-2.l...	3	/favicon.ico	13:05:26...	3	0	0	239	0.0.0.0		
/GunMac-2.l...	4	/user/{userId}	13:05:28...	3	1	0	101	0.0.0.0		
/GunMac-2.l...	2	/favicon.ico	13:05:28...	2	0	0	237	0.0.0.0		
/GunMac-2.l...	4	/user/{userId}	13:05:33...	4	2	1	114	0.0.0.0		
/GunMac-2.l...	3	/favicon.ico	13:05:33...	2	0	0	237	0.0.0.0		
/GunMac-2.l...	5	/user/{userId}	13:05:54...	5	2	2	112	0.0.0.0		
/GunMac-2.l...	2	/favicon.ico	13:05:55...	2	0	0	237	0.0.0.0		
/GunMac-2.l...	20	/user/{userId}	13:06:20...	4	2	1	126	0.0.0.0		
/GunMac-2.l...	36	/user/{userId}	13:06:29...	33	2	1	2,776	0.0.0.0		



5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- Xlog 주요 메타 정보

service : service명

elapsed : 응답시간

endtime : 요청의 종료시간

ipaddr: 요청자의 ip

userAgent

sqlCallCount, sqlCallTime : 실행한 SQL 개수와 수행시간

apiCallCount, apiCallTime : 외부로 http call을 수행한 개수와 시간

cpu: 해당 요청을 처리하는 동안 사용한 cpu time

kbytes: 해당 요청을 처리하는 동안 사용한 heap memory

thread: 처리한 thread명

country, city : geoip 정보(서버에 *geoip_data_city_file* 옵션 설정시)

queuing time : proxy 서버로 부터 인입까지 걸린 시간(설정시)

login, desc, text1~5 : agent plugin을 통해 사용자가 설정한 정보



5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

• XLog

- Xlog 프로파일 뷰이다. 목록에서 하나의 항목을 선택하면 나타난다.
- 여기서는 각 요청이 어떤 흐름으로 수행되는지 상세한 내용을 확인할 수 있다.

```
XLog - Tomcat x659bjiv9e7qog X
▶ txid      = x659bjiv9e7qog
▶ objName   = /GunMac-2.local/scouterWebDemo1
▶ thread    = http-nio-8081-exec-1
▶ endtime   = 20170724 13:06:20.749
▶ elapsed   = 20 ms
▶ service   = /user/{userId}
▶ ipaddr=0.0.0.0, userid=-7911099780103871554
▶ cpu=4 ms, kbytes=126
▶ sqlCount=2, sqlTime=1 ms
▶ userAgent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.307
▶ group=/user
```

p#	#	TIME	T-GAP	CPU	CONTENTS
	[*****]	13:06:20.729	0	0	start transaction
-	[000000]	13:06:20.729	0	0	[driving thread] http-nio-8081-exec-1
-	[000001]	13:06:20.730	1	0	OPEN-DBC jdbc:hsqldb:mem;shutdown=true [1ms] -- OPEN-DBC jdbc:hsqldb
[000001]	[000002]	13:06:20.730	0	0	STM> SELECT COUNT(*) FROM INFORMATION_SCHEMA.SYSTEM_USERS 1 ms
-	[000003]	13:06:20.731	1	0	PRE> select user_id, user_name, created from user where user_id=? ['user1001'] 0 ms
-	[000004]	13:06:20.732	1	0	CLOSE [0ms] -- CLOSE
	[*****]	13:06:20.749	17	4	end of transaction



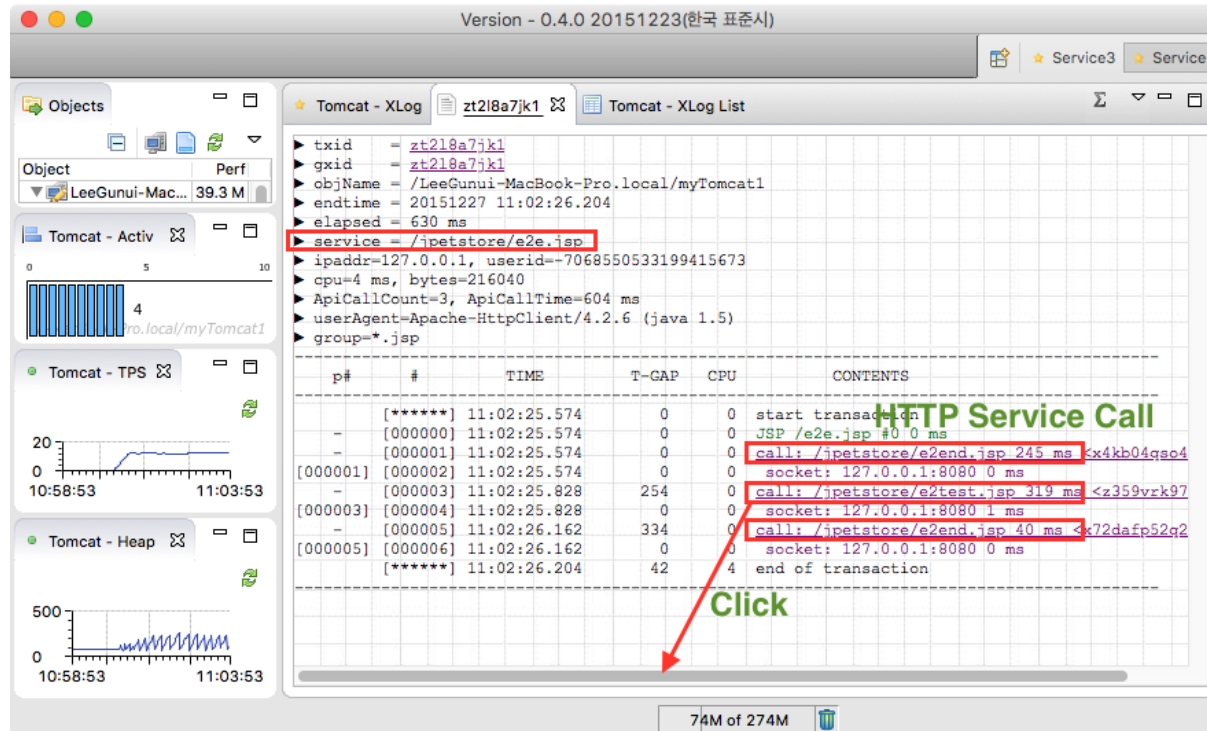
5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

- Xlog 프로파일

- 상세 프로파일은 기본적으로 SQL 및 Socket 연결, http call 및 redis call, async thread call 등과 같은 연계에 대한 정보를 보여주며, 추가적인 설정을 통해 특정한 패턴의 메소드를 프로파일하게 할 수도 있다.



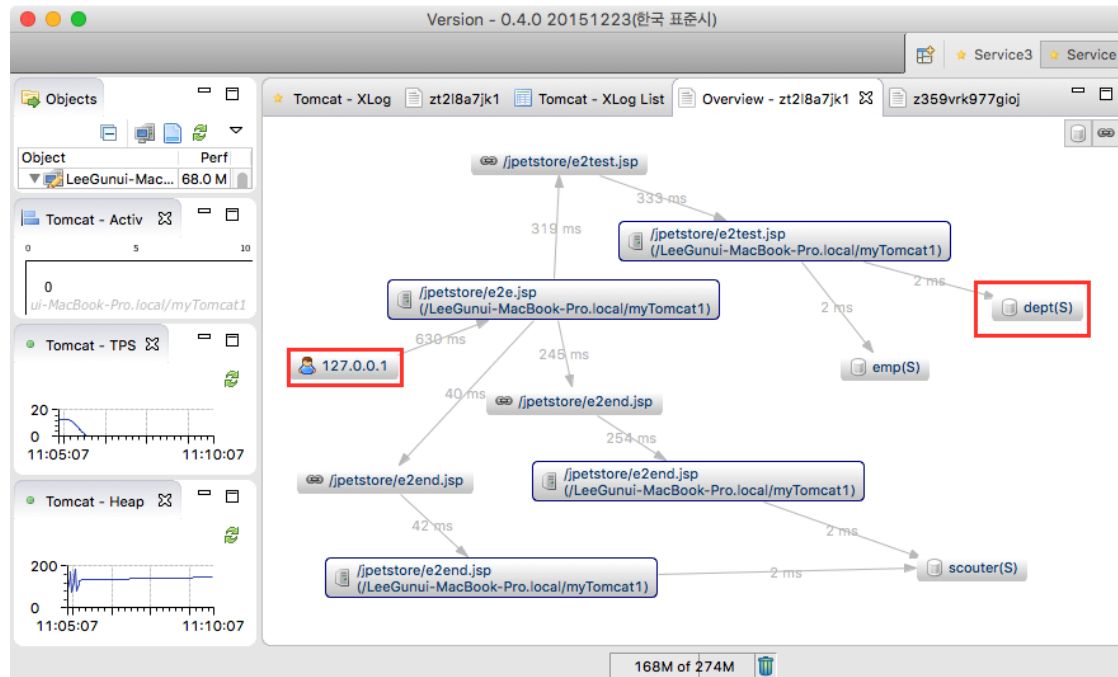
5. 기능소개 및 활용가이드



5.3 XLOG 및 액티브 서비스 모니터링

• Xlog 프로파일

- 만약 호출되는 API의 서버에도 Scouter가 설치되어있다면, Scouter는 spanid를 발행하여 각 서비스들(혹은 분기된 비동시 thread들)을 연결하여 추적할 수 있게 한다.
- 또한 gxid 링크를 클릭하게 되면 서비스의 시작점부터 호출된 서비스 및 각 서비스가 사용한 테이블의 CURD 현황을 하나의 다이어그램으로 보여주게 된다.





Q Scouter를 구성할때 필요한 최소 사양은 어떻게 되나요?

&

A 콜렉터 서버는 IO 바운드한 성격의 솔루션입니다. 따라서 CPU는 일반적으로 2core 정도면 수십개의 인스턴스를 모니터링 할 수 있습니다.
다만 데이터 저장소는 수집량에 따라 달라지므로 오랜기간 보관하기 위해서는 큰 저장소가 필요하며 최소 500G 이상을 권장합니다.
또한 HDD 보다는 SSD를 사용하기를 권장드립니다.

Q Java 모니터링시에 자원 사용량에 영향이 있나요?

&

A 어플리케이션의 정보를 수집해야 하기때문에 당연히 해당 어플리케이션에 오버헤드가 발생합니다. 다만 오버헤드는 크지 않으며 일반적으로 3%~5%로 보면 됩니다. 이는 스카우터가 아닌 다른 APM도 마찬가지 입니다.



8. 용어정리



용어	설 명
컬렉터	스카우터의 수집서버를 지칭한다. 에이전트가 설치된 모니터링 대상들의 성능 정보가 모이는 곳이며 이를 수신하기 위한 네트워크 레이어와 수집된 정보를 저장하기 위한 스토리지 레이어로 구성된다.
뷰	스카우터 화면의 단위 요소를 "뷰"라고 부른다. 각각의 뷰는 각각의 이름을 가지며, 하나의 성능 정보를 표현한다.
오브젝트	스카우터 모니터링 대상을 오브젝트라고 부른다. 따라서 호스트, 자바 어플리케이션, 자바 어플리케이션의 하위 모니터링 대상등이 모두 오브젝트이다.
액티브 서비스	현재 서비스에서 처리가 시작되고 끝나지 않은 요청을 의미한다. 종종 서비스 장애나 부하시점에 액티브 서비스의 숫자와 오래 진행중인 액티브 서비스가 많아지므로 가장 중요한 어플리케이션 모니터링 요소중의 하나이다.
프로파일	하나의 요청이 처리된 발자취를 시간순으로 코드 수준에서 나열한 정보를 프로파일이라고 한다. 어플리케이션 문제 발생시 원인 분석을 위한 가장 중요한 정보이다.



Open Source Software Installation & Application Guide



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.