

나만의 성능 모니터링 만들기

APM을 구현하는 기술과
오픈소스 APM Scouter의 고급 활용 기법



Gunhee Lee
SCOUTER commiter

이건희 - **그분은 아닙니다.**

그래서 구글서 검색 안됨

LG CNS S/W architect



LG CNS APM - TUNA 개발자리더



OSS APM Scouter co-founder & commiter



KOSSLab 3gi - 오픈소스포커팀리더



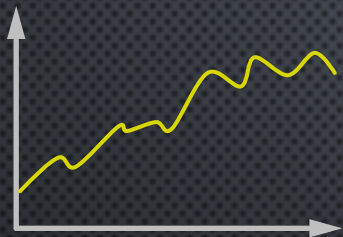
이야기 할 것

- APM은 어떻게 만들어지나, 구현하는 기술?
- Scouter 를 통한 어플리케이션 모니터링
- Scouter 고급활용: 플러그인

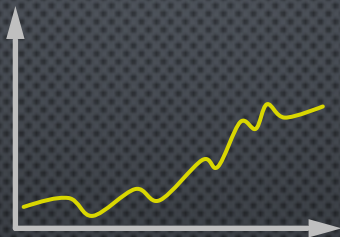
Runtime에서 메소드 아규먼트 값 확인하기

- 플러그인을 통해서 나만의 모니터링 시스템 만들기

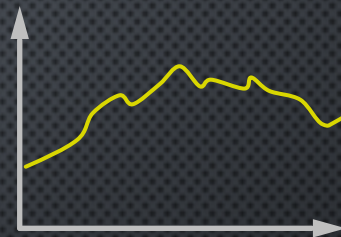
??



CPU



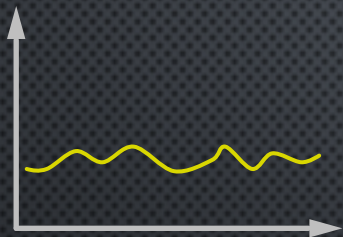
응답시간



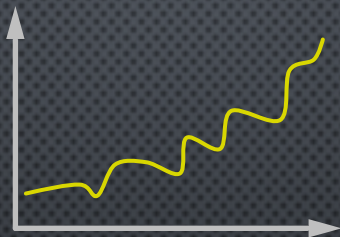
처리량

이건 뭐...

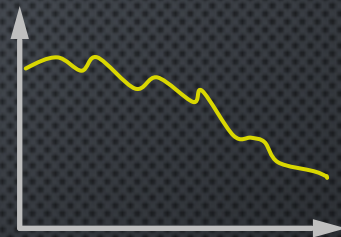
??



CPU



응답시간



처리량

이건 뭔가 좀 이상하다...

??



시간에 따른 숫자들 - 성능메트릭
-> visualization

??



이것들은 결과일 뿐...

??



원인을 모를때 우리는 이런 일
들을 하지요...

??

RESTART!











??

그럼 왜 ???
원인이 뭐냐!



APM이란?

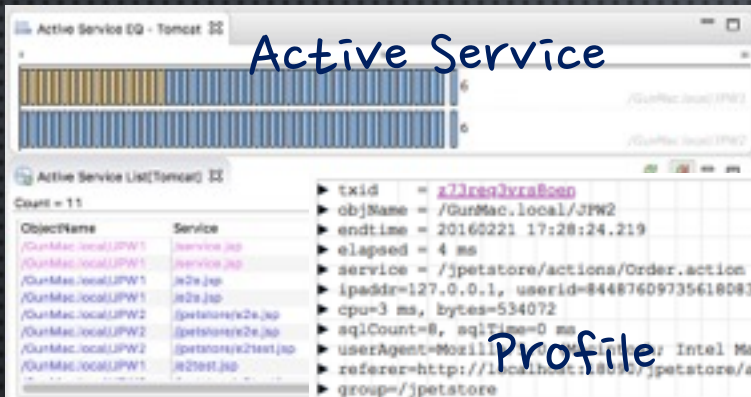
Application performance management

성능 메트릭 + 원인을 찾기위한 α

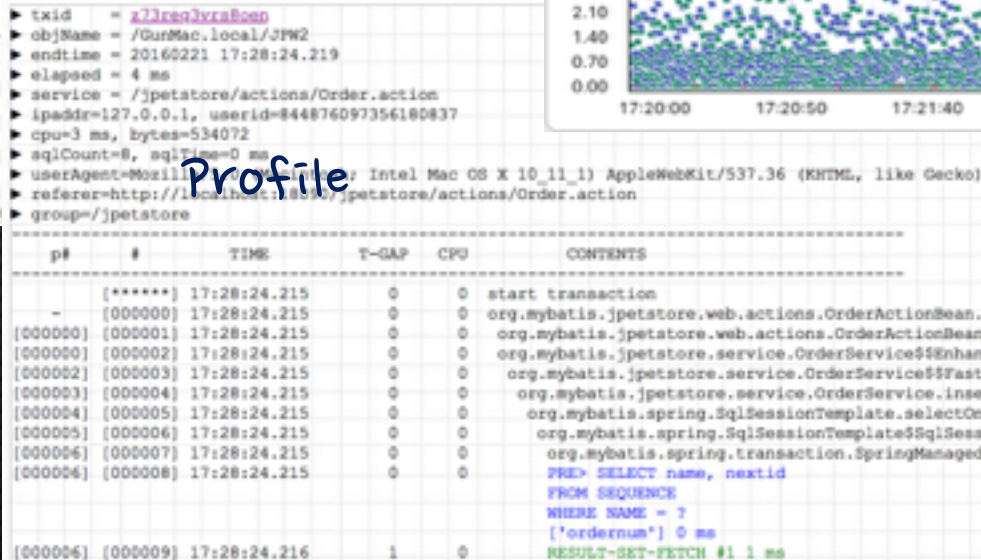


APM

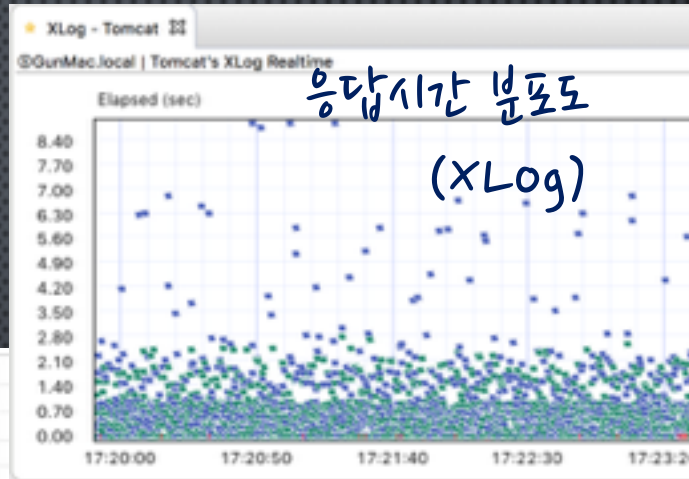
Active Service



Profile



응답시간 분포도 (XLog)



APM을 구현하는 기술

어떻게 성능 정보를 수집하는가?

```
▶ txid = 233e9339a8000
▶ objName = /GurMac.local/JFW2
▶ endTime = 20160221 17:28:24.219
▶ elapsed = 4 ms
▶ service = /jpetstore/actions/Order.action
▶ ipaddr=127.0.0.1, userid=844876097356180837
▶ cpu=3 ms, bytes=534072
▶ sqlCount=8, sqlTime=0 ms
▶ userAgent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) AppleWebKit/537.36 (KHTML, like Gecko)
▶ referer=http://localhost:18090/jpetstore/actions/Order.action
▶ group=/jpetstore
```

p#	#	TIME	T-GAP	CPU	CONTENTS
-	[*****]	17:28:24.215	0	0	start transaction
-	[000000]	17:28:24.215	0	0	org.mybatis.jpstest.web.actions.OrderActionBean.i
[000000]	[000001]	17:28:24.215	0	0	org.mybatis.jpstest.web.actions.OrderActionBean.i
[000000]	[000002]	17:28:24.215	0	0	org.mybatis.jpstest.service.OrderService\$\$Enhanc
[000002]	[000003]	17:28:24.215	0	0	org.mybatis.jpstest.service.OrderService\$\$FastC
[000003]	[000004]	17:28:24.215	0	0	org.mybatis.jpstest.service.OrderService.inser
[000004]	[000005]	17:28:24.215	0	0	org.mybatis.spring.SqlSessionTemplate.selectOne
[000005]	[000006]	17:28:24.215	0	0	org.mybatis.spring.SqlSessionTemplate\$\$SqlSess
[000006]	[000007]	17:28:24.215	0	0	org.mybatis.spring.transaction.SpringManagedT
[000006]	[000008]	17:28:24.215	0	0	PRE> SELECT name, nextid FROM SEQUENCE WHERE NAME = ? ['ordernum'] 0 ms
[000006]	[000009]	17:28:24.216	1	0	RESULT-SET-FETCH #1 1 ms

APM을 구현하는 기술

Super Power

BCI (Byte code instrumentation)

- bytecode에 직접 변경을 가해

소스코드의 수정없이 원하는 기능을

삽입할 수 있는 방법



Bci를 통한 Method의 수행 시간 측정

```
methodA() {  
    (시작시간); // <- Bci  
    ... 원래 코드 ...  
    (end시간-시작시간) // <- Bci  
    (출력 or 전송) // <- Bci  
}
```

BCI

어려워서 어렵지 않다.



premain() 만 기억하자!

스피어
Super Power

BCI

premain()

- premain()은 main() 전에 수행된다.
- 여기서 transformer를 통해 class 변경

BCI 예제

Thread name enhancer

HttpServlet.service()

http-bio-7 :

뭔가 모자른다...

uri ??

Start time ?? 특정 parameter?

```
"http-bio-18080-exec-10" #39 daemon prio=5 os_prio=31 tid=0x00007fe8add5
java.lang.Thread.State: WAITING (parking)
  at sun.misc.Unsafe.park(Native Method)
  - parking to wait for <0x00000006c0da1370> (a java.util.concurrent.locks.Ab
  at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
  at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(LockSupport.java:175)
  at java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
  at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:104)
  at org.apache.tomcat.util.threads.TaskQueue.take(TaskQueue.java:32)
  at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1076)
  at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1145)
  at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
  at org.apache.tomcat.util.threads.TaskThread$WrappingRunnable.run(TaskThread.java:61)
  at java.lang.Thread.run(Thread.java:745)
```


BCI 예제

Thread name enhancer

<https://github.com/gunleeol/thread-name-enhancer>

BCI - ByteScope

Runtime java process 에

Thread name enhancer ^{기능}을

attach 할수 있는 CLI 방식의 utility

BCI - ByteScope

```
bytescope> jps
jps
92462 scouterx.toys.bytescope.runner.RunnerMain
92461 org.jetbrains.jps.cmdline.Launcher /Applications/
92123 org.apache.catalina.startup.Bootstrap start

bytescope> use 92123
use 92123
send agent log dir : /Users/gunlee/Documents/workspace/
dynamically loading javaagent
bytescope-92123> attach threadNameEnhancer
attach threadNameEnhancer
attache thread name enhancer : to 92123@GunMac
[attached] EnhancedThreadNameForServlet
bytescope-92123>
```

BCI - Bytescope

```
[gunlee@GUN ~/Programs]
```

```
$ jstack 92123
```

```
2016-10-21 18:56:25
```

```
Full thread dump Java HotSpot(TM) 64-Bit Server VM (20.65-b04-462 mixed mode):
```

```
"http-bio-8090-exec-4 [bytescope][uri]/e2end.jsp [start at] 2016.10.21 18:56:24 [1477043784950]"
```

```
daemon prio=5 tid=7f9db8801800 nid=0x7000002303000 waiting on condition [7000002301000]
```

```
java.lang.Thread.State: TIMED_WAITING (sleeping)
```

```
at java.lang.Thread.sleep(Native Method)
```

```
at org.apache.jsp.e2end_jsp._jspService(e2end_jsp.java:87)
```

```
at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
```

```
at javax.servlet.http.HttpServlet.service(HttpServlet.java:731)
```

```
at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:439)
```

```
at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:395)
```

```
at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:339)
```

```
at javax.servlet.http.HttpServlet.service(HttpServlet.java:731)
```


BCI - ByteScope

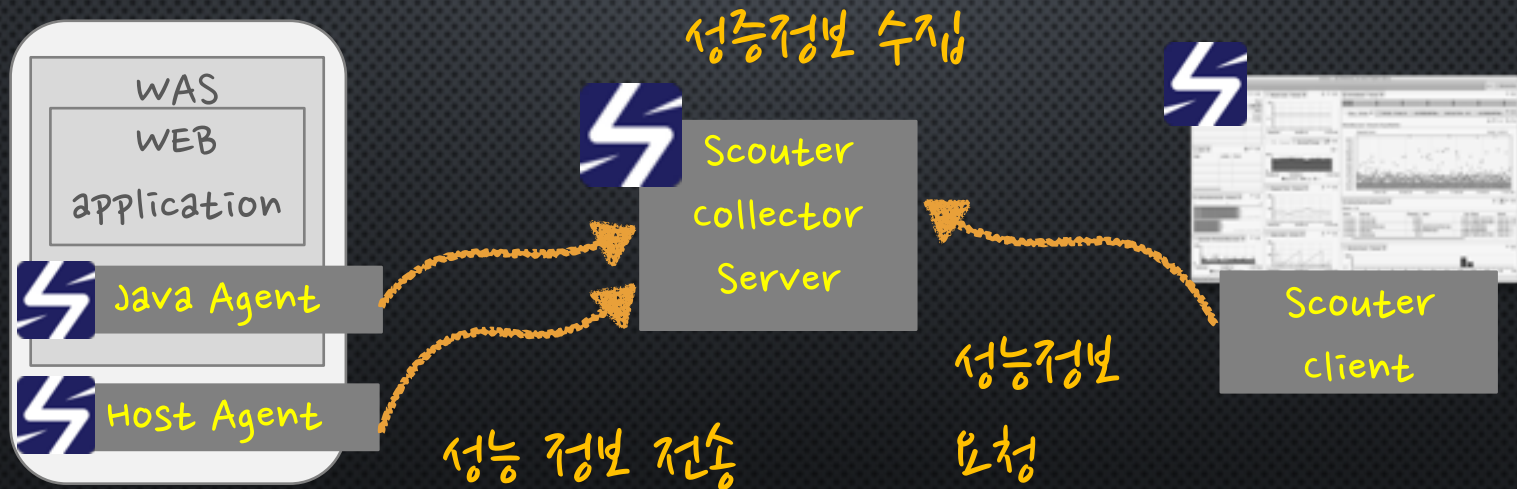
<https://github.com/scouter-project/bytescope>

opensource APM

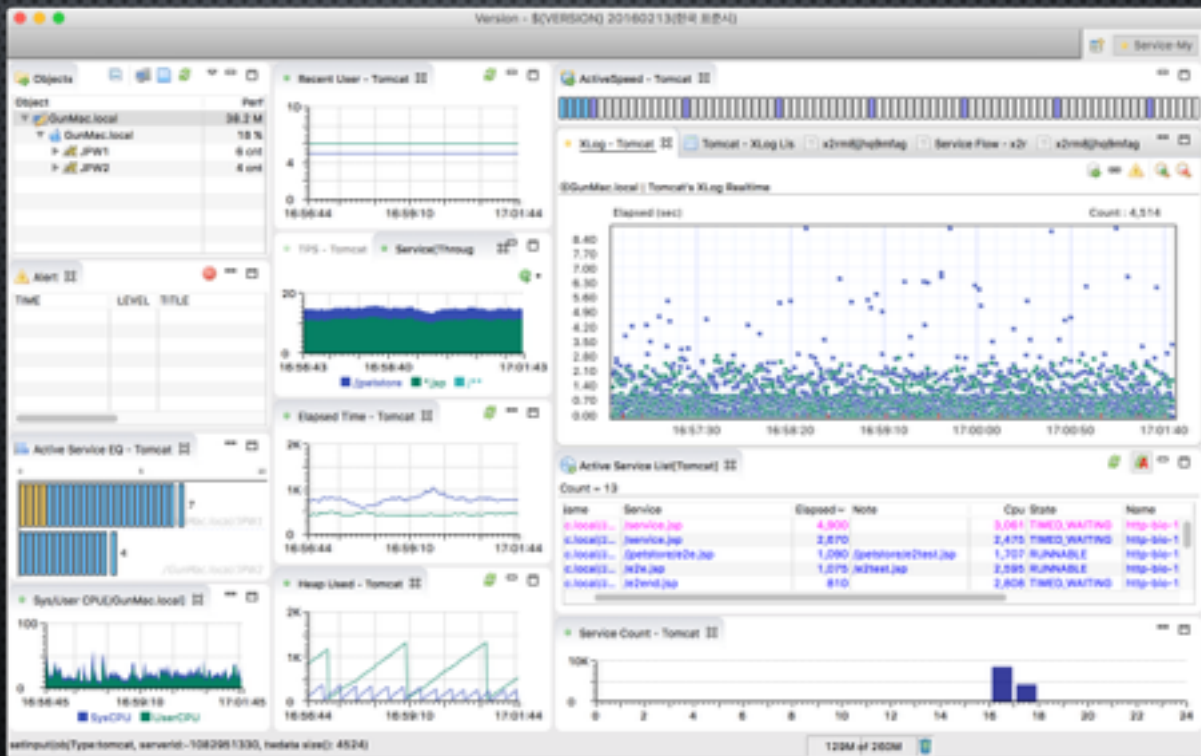
Scouter Basics

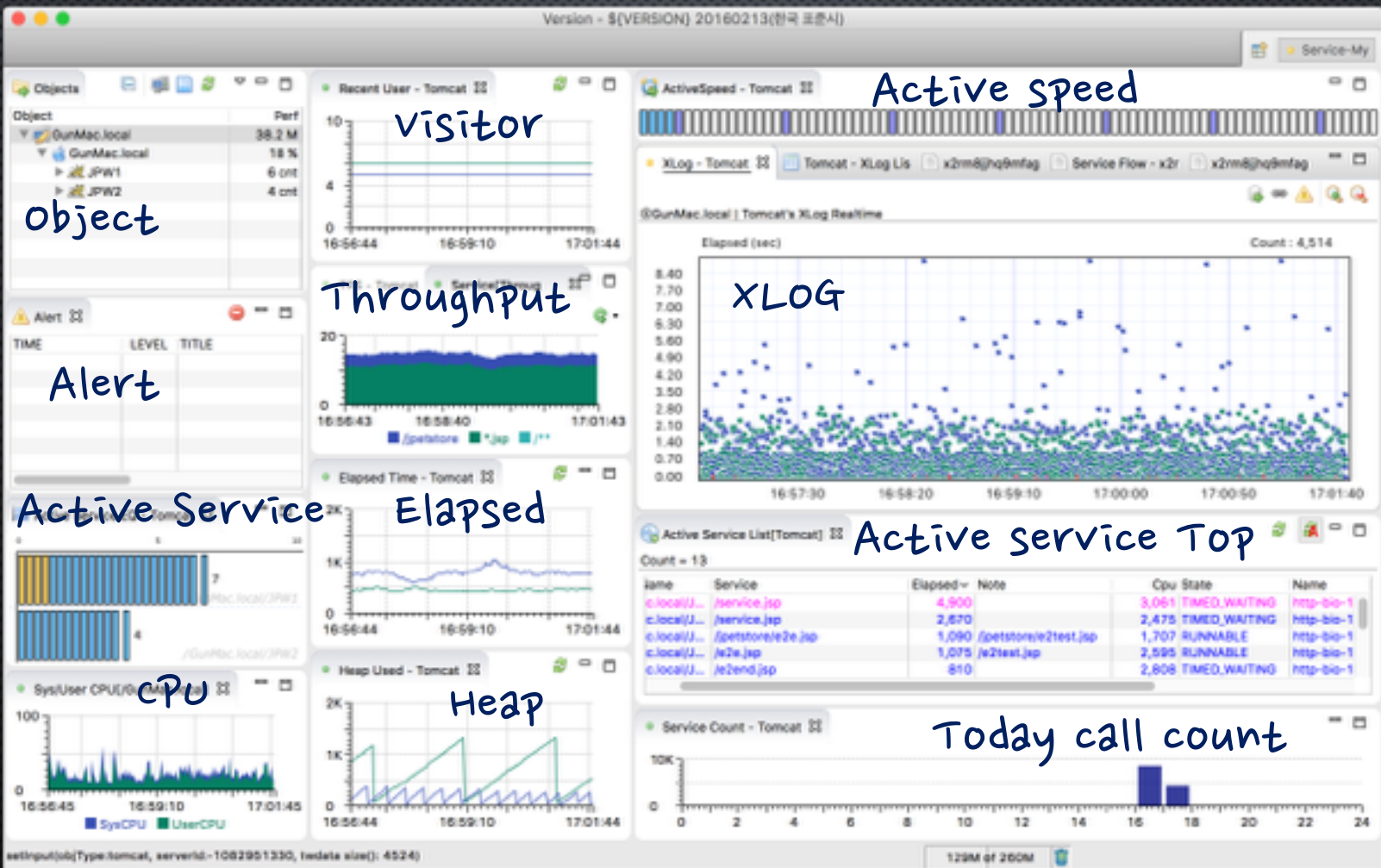
SCOUTER

아키텍처는? Agent, collector, client

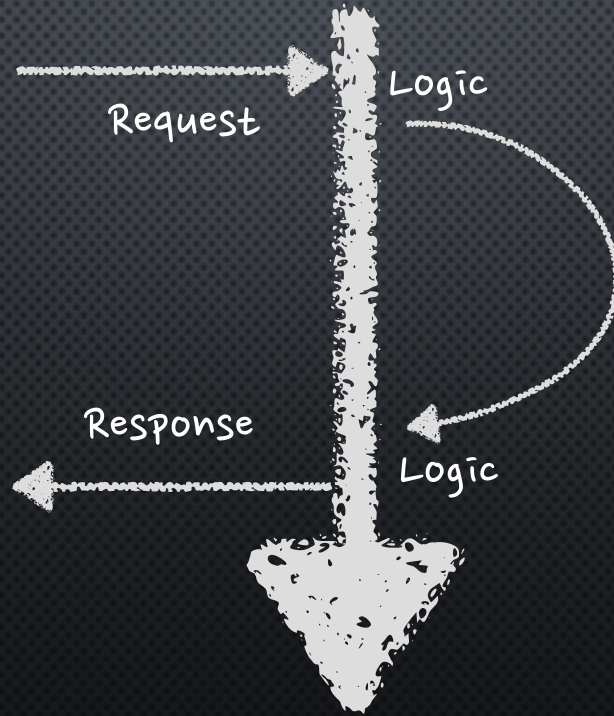


SCOUTER



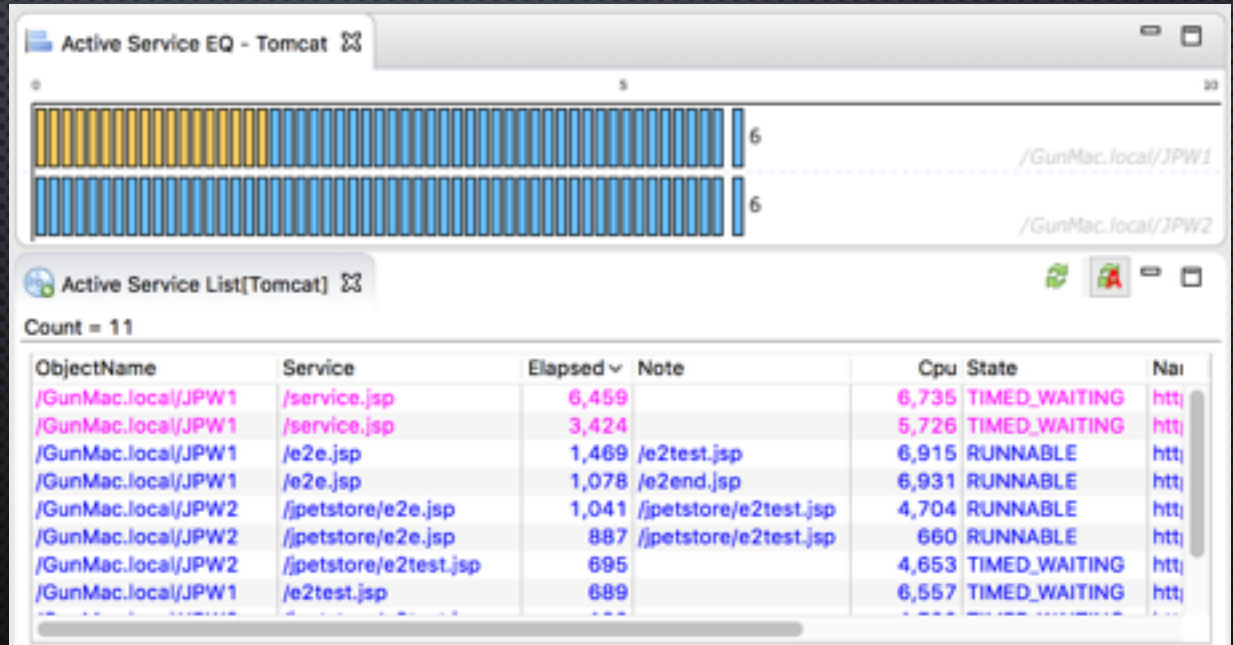


Application monitoring with scouter



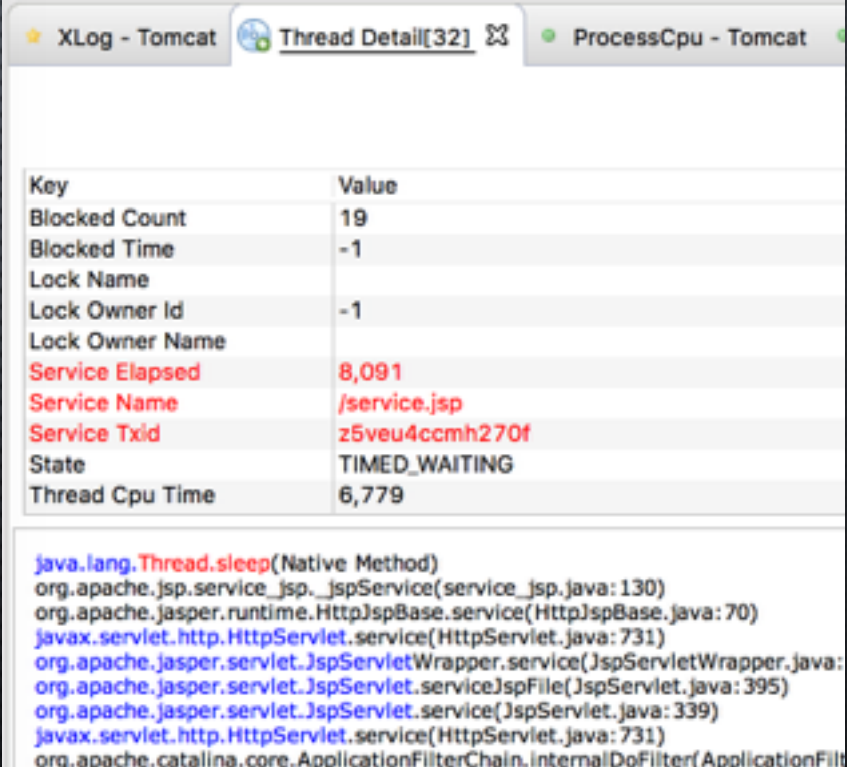
Application monitoring with scouter

Active Service



Application monitoring with scouter

Active Service



The screenshot displays the Scouter application monitoring interface. The top navigation bar includes tabs for 'XLog - Tomcat', 'Thread Detail[32]', and 'ProcessCpu - Tomcat'. The 'Thread Detail[32]' tab is active, showing a table of thread statistics and a stack trace.

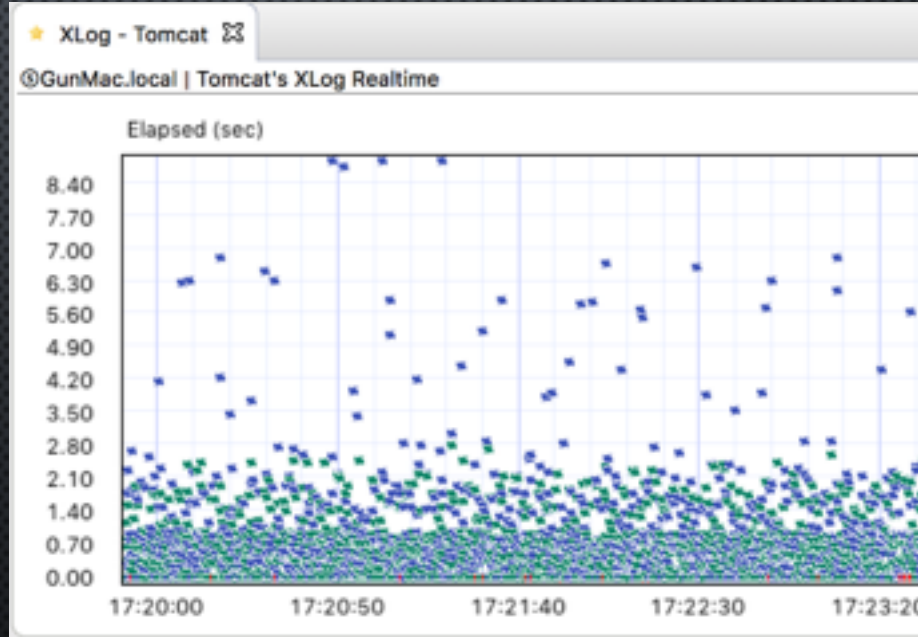
Key	Value
Blocked Count	19
Blocked Time	-1
Lock Name	
Lock Owner Id	-1
Lock Owner Name	
Service Elapsed	8,091
Service Name	/service.jsp
Service TxId	z5veu4ccmh270f
State	TIMED_WAITING
Thread Cpu Time	6,779

Below the table, the stack trace is displayed:

```
java.lang.Thread.sleep(Native Method)
org.apache.jsp.service_jsp._jspService(service_jsp.java:130)
org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:70)
javax.servlet.http.HttpServlet.service(HttpServlet.java:731)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:395)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:339)
javax.servlet.http.HttpServlet.service(HttpServlet.java:731)
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilt
```


Application monitoring with scouter

XLOG



Application monitoring with scouter

```

▶ txid = x73req3vrs8oen
▶ objName = /GunMac.local/JPN2
▶ endtime = 20160221 17:28:24.219
▶ elapsed = 4 ms
▶ service = /jpetstore/actions/Order.act
▶ ipaddr=127.0.0.1, userid=8448760973561
▶ cpu=3 ms, bytes=534072
▶ sqlCount=8, sqlTime=0 ms
▶ userAgent=Mozilla/5.0 (Macintosh; Inte
▶ referer=http://localhost:18090/jpetsto
▶ group=/jpetstore

```

Elapsed	Service	StartTime	EndTime	Txid	Cpu	SQL Cou	SQL Time
83	/jpetstore/e2test.jsp	11:09:...	11:09:...	zld...	8	2	7
18	/service.jsp	11:09:...	11:09:...	z7...	7	1	5
120	/e2test.jsp	11:09:...	11:09:...	z5...	9	2	5
32	/jpetstore/e2test.jsp	11:09:...	11:09:...	z5...	9	2	5

p#	#	TIME	T-GAP	CPU	CONTENTS
	[*****]	17:28:24.215	0	0	start transaction
-	[000000]	17:28:24.215	0	0	org.mybatis.jpetest.web.actions.OrderActionBean.i
[000000]	[000001]	17:28:24.215	0	0	org.mybatis.jpetest.web.actions.OrderActionBean.i
[000000]	[000002]	17:28:24.215	0	0	org.mybatis.jpetest.service.OrderService\$\$Enhanc
[000002]	[000003]	17:28:24.215	0	0	org.mybatis.jpetest.service.OrderService\$\$FastC
[000003]	[000004]	17:28:24.215	0	0	org.mybatis.jpetest.service.OrderService.inser
[000004]	[000005]	17:28:24.215	0	0	org.mybatis.spring.SqlSessionTemplate.selectOne
[000005]	[000006]	17:28:24.215	0	0	org.mybatis.spring.SqlSessionTemplate\$SqlSess
[000006]	[000007]	17:28:24.215	0	0	org.mybatis.spring.transaction.SpringManaged
[000006]	[000008]	17:28:24.215	0	0	PRE> SELECT name, nextid FROM SEQUENCE WHERE NAME = ? ['ordernum'] 0 ms
[000006]	[000009]	17:28:24.216	1	0	RESULT-SET-FETCH #1 1 ms

Profile

Scouter In Practice

- Scouter Plugin

Scouter Plugin



Scripting plugin



Scripting
plugin

- http service plugin
http의 요청 시작 & 끝
- http call plugin
http client로 원격 서비스 호출시
- capture plugin
지정된 메소드의 시작 & 끝

Agent Plugin을 통한 사용자ID 프로파일

HTTP service plugin

1. 세션에서 user id 획득

String userId =

`$req.getSessionAttribute("userId");`

Agent Plugin을 통한 사용자ID 프로파일

Java Agent - HTTP service plugin

2. XLog 항목에 id를 추가

```
$ctx.login(userId);
```

Agent Plugin을 통한 사용자ID 프로파일

Java Agent - HTTP service plugin

3. Profile에 사용자 정의 문장 기입

```
$ctx.profile("Login ID = " + userId);
```


Agent Plugin을 통한 사용자ID 프로파일

```
▶ txid = x5lnibgmh74occ
▶ objName = /GunMac.local/JPW1
▶ endtime = 20160221 21:52:06.966
▶ elapsed = 9 ms
▶ service = /actions/Cart.action
▶ ipaddr=0.0.0.0, userid=2552704822505855920
▶ cpu=7 ms, bytes=313856
▶ sqlCount=1, sqlTime=0 ms
▶ userAgent=Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko)
▶ referer=http://localhost:18080/actions/Catalog.action?viewItem=&itemId=EST-27
▶ group=/actions
```

p#	#	TIME	T-GAP	CPU	CONTENTS
	[*****]	21:52:06.957	0	0	start transaction
-	[000000]	21:52:06.957	0	0	#### It's user defined profile gunlee
-	[000001]	21:52:06.959	2	0	org.mybatis.jp.petstore.web.actions.CartActionBean.a
[000001]	[000002]	21:52:06.959	0	0	org.mybatis.jp.petstore.web.actions.CartActionBean.a
[000001]	[000003]	21:52:06.959	0	0	org.mybatis.jp.petstore.web.actions.CartActionBean.a
[000003]	[000004]	21:52:06.959	0	0	org.mybatis.spring.SqlSessionTemplate.selectOne
[000004]	[000005]	21:52:06.959	0	0	org.mybatis.spring.SqlSessionTemplate.selectOne
[000005]	[000006]	21:52:06.959	0	0	org.mybatis.spring.transaction.SpringManagedTr
[000005]	[000007]	21:52:06.959	0	0	PRE> SELECT QTY AS value FROM INVENTORY WHERE ITEMID = ? ['EST-27'] 0 ms
[000005]	[000008]	21:52:06.959	0	0	RESULT-SET-FETCH #1 0 ms

Agent Plugin을 통한 Parameter debugging

Agent Plugin을 통한 Parameter debugging

AI서비스가 가끔 에러가 발생하거나 느려지는데...

파라미터 확인만 되면 해결할 수 있을 듯...

근데 로깅 불가... 재기동 불가...

Agent Plugin을 통한 Parameter debugging

capture plugin

+ hook_args_patterns

+ redefine class

Built-in Plugin (Server plugin)



Built-in Plugin (Server plugin)

Scouter-Alert-Telegram

Scouter-Alert-Slack

Scouter-Alert-email

Scouter-Influxdb

Influxdb-plugin

<https://github.com/scouter-project/scouter-plugin-server-influxdb>

성능 카운터를 Influxdb로 전송



Influxdb-plugin

Scouter -> InfluxDB -> Grafana

연동해 보기 참고 url

<https://goo.gl/fSd8Dd>

Influxdb-plugin

Demo

<http://demo.scouterapm.com:3000>

(scouter/scouter)



Built-In Plugin 제작

1. `scouter.plugin.server` 패키지로 시작
2. Method annotation 명시
3. jar로 만든 후 `./lib` 디렉토리에 넣는다.

* Dependency

- `scouter.common`
- `scouter.server`

Built-In Plugin 제작

```
public class NullPlugin {
    Configure conf = Configure.getInstance();

    @ServerPlugin(PluginConstants.PLUGIN_SERVER_ALERT)
    public void alert(AlertPack pack){
        if(conf.getBoolean("ext_plugin_null_alert_enabled", true)) {
            println("[NullPlugin-alert] " + pack);
        }
    }

    @ServerPlugin(PluginConstants.PLUGIN_SERVER_COUNTER)
    public void counter(PerfCounterPack pack){
        if(conf.getBoolean("ext_plugin_null_counter_enabled", true)) {
            println("[NullPlugin-counter] " + pack);
        }
    }
}
```

```
@ServerPlugin(PluginConstants.PLUGIN_SERVER_OBJECT)
public void object(ObjectPack pack){
    if(conf.getBoolean("ext_plugin_null_object_enabled", true)) {
        println("[NullPlugin-object] " + pack);
    }
}

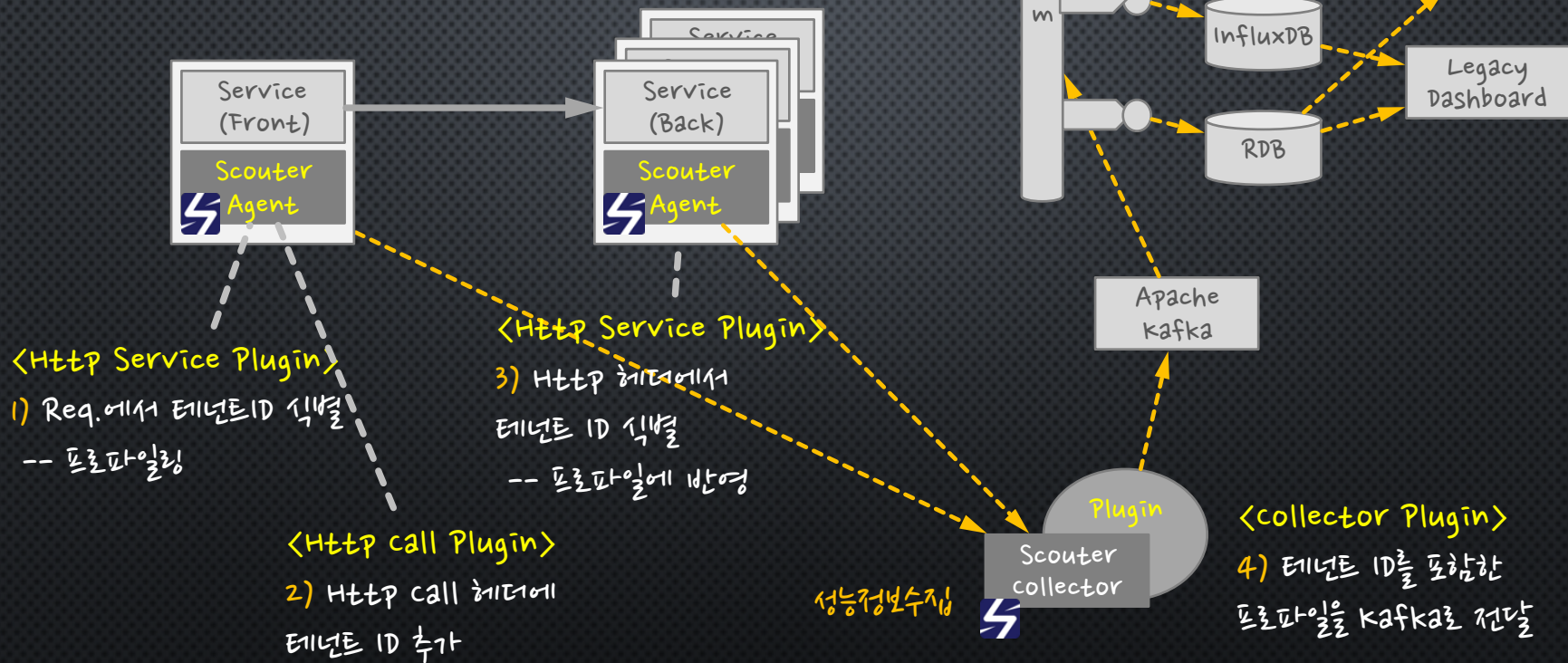
@ServerPlugin(PluginConstants.PLUGIN_SERVER_SUMMARY)
public void summary(SummaryPack pack){
    if(conf.getBoolean("ext_plugin_null_summary_enabled", true)) {
        println("[NullPlugin-summary] " + pack);
    }
}
}
```

Built-In Plugin 제작

Guide 문서

[https://github.com/scouter-project/scouter/blob/master/
scouter.document/main/Plugin-Guide.md](https://github.com/scouter-project/scouter/blob/master/scouter.document/main/Plugin-Guide.md)

Plugin 활용 사례



정리해 보자 !

Application 모니터링

문제의 원인으로 접근

Scouter 확장 - Plugin

- debugging, 사용자정의 프로파일
- 다른 오픈소스와 결합, 확장

SCOUTER의 확장

그리고..

Plugin을 활용하여

centralized logging / monitoring

물론 Application의 수정 없이...



만들고 싶은 것 ...

kafka-plugin

elasticsearch-Plugin (xlog, counter)

만들고 싶은 것 ...

Redis agent

Kafka agent

Cassandra agent

Nodejs agent

...

SCOUTER가 하고 싶은 것

오픈 소스 중심의 통합 모니터링

node.js(w/ express.js)

Apache HTTPD

Redis / memcached

NoSQL(Mongo DB, cassandra...)

opensource RDB (Maria, cubrid...)

client side monitoring(script error, dom rendering)

...

다양한 형태의 contribution을 기다립니다~

쉬운것 부터...

매뉴얼

Plugin

간단한 Agent

SCOUTER에 기여하기

GitHub

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

Facebook 사용자 모임

<https://www.facebook.com/groups/scouterapm>

감사합니다

gunleeol@gmail.com

