****

Scouter 장애 대응 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 3](#_Toc26917418)

[**1.1** **스카우터란?** 3](#_Toc26917419)

[2. Scouter 서버 기동 및 정지 4](#_Toc26917420)

[**2.1** **스카우터 서버의 기동** 4](#_Toc26917421)

[**2.2** **스카우터 서버의 정지** 4](#_Toc26917422)

[3. Scouter 서버 서비스 체크 5](#_Toc26917423)

[**3.1** **Scouter Server Process Status** 5](#_Toc26917424)

[**3.2** **Port Listen 정상 여부 확인** 5](#_Toc26917425)

[**3.3** **Scouter 서버 리소스 체크** 6](#_Toc26917426)

[**3.3.1** **CPU 사용량** 6](#_Toc26917427)

[**3.3.2** **Memory 사용량** 7](#_Toc26917428)

[**3.3.3** **프로세스 과부하 cpu 여부 확인** 7](#_Toc26917429)

[**3.3.4** **Disk Usage** 8](#_Toc26917430)

[4. 장애 유형별 대응 방법 9](#_Toc26917431)

[**4.1** **Scouter 접속 실패시** 9](#_Toc26917432)

[**4.1.1** **리소스 및 프로세스 체크** 9](#_Toc26917433)

[**4.1.2** **Too many Openfiles** 9](#_Toc26917434)

[**4.1.3** **Server Crash** 10](#_Toc26917435)

[**4.2** **대시보드 모니터링 실패** 11](#_Toc26917436)

[**4.2.1** **오브젝트 타입 오류에 의한 모니터링 실패** 12](#_Toc26917437)

[**4.2.2** **모니터링 인터페이스 오류** 14](#_Toc26917438)

[5. APPENDICES 16](#_Toc26917439)

[**5.1** **Document Guidelines** 16](#_Toc26917440)

[**5.2** **Project Migration Report Sections Omitted** 16](#_Toc26917441)

# **시작하기**

## **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

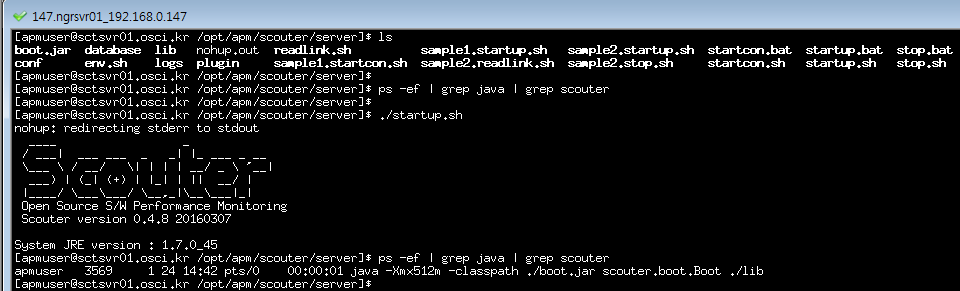
-오픈소스 인프라 확장 추진 로드맵 제시

# **Scouter 서버 기동 및 정지**

## **스카우터 서버의 기동**

***cd /opt/apm/scouter/server***

***sh ./startup.sh***



## **스카우터 서버의 정지**

***cd /opt/apm/scouter/server***

***sh ./stop.sh***



# **Scouter 서버 서비스 체크**

## **Scouter Server Process Status**

서비스 되어지는 Scouter 서버의 실행상태에 대해서 모니터링을 합니다.

***ps –ef | grep ${processName}***

ex) ps –ef | grep java | grep Scouter

항목 설명

ps 명령을 통해서 해당 프로세스의 현재 동작 여부를 확인합니다.

정상 여부 확인 : 프로세스 목록에 해당 내용이 있어야만 합니다.

|  |
| --- |
| **$ ps -ef|grep java|grep Scouter**  jboss 30777 30646 0 May15 ? 00:29:58 /opt/java/jdk1.6.0\_45/bin/java -D[Scouter] XX:+UseCompressedOops -server -Xms1024m -Xmx1024m -XX:MaxPermSize=256m -verbose:gc -XX:+PrintGCDetails |

## **Port Listen 정상 여부 확인**

netstat –an | grep ${servicePort}

ex) netstat –an | grep 6100

항목 설명

netstat 명령을 통해서 해당 프로세스의 현재 Network 동작 여부를 확인합니다.

정상 여부 확인 : Port 목록에 해당 서비스에서 사용되는 포트가 LISTEN 항목으로 되어 있어야 합니다.

|  |
| --- |
| **$ netstat -an | grep 6100**  tcp 0 0 :::6100 :::\* LISTEN |

## **Scouter 서버 리소스 체크**

|  |  |  |  |
| --- | --- | --- | --- |
| **대항목** | **중항목** | **세부항목** | **설명** |
| **OS** | System Resource Usage | CPU 사용량 | CPU 전체 사용량 |
| Memory 사용량 | Memory 전체 사용량 |
| **운영환경 서버 상태** | Process Resource Usage | Process CPU 과부하 사용 | 프로세스 CPU 사용률 |
| Disk Usage | 설치 디스크의 사용량 |
| Server Status | Server Status | Server Life Cycle |

* + 1. **CPU 사용량**

Scouter 서버의 CPU사용량 체크를 위하여 다음의 명령어를 통해 모니터링을 진행합니다.

vmstat 1 100

항목 설명

us : user cpu usage

sy : system cpu usage

id : cpu idle usage

정상 여부 확인 : id 60% 이상일 경우 양호, id 30% 미만인 경우 위험

|  |
| --- |
| **$ vmstat 1 100**  procs -----------memory---------- ---swap-- -----io---- --system-- **-----cpu-----**  r b swpd free buff cache si so bi bo in cs us sy id wa st  0 0 853424 69016 4648 29128 0 0 13 8 8 4 0 0 99 0 0  0 0 853424 69008 4648 29128 0 0 0 0 113 242 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 75 211 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 106 254 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 73 208 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 118 263 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 78 203 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 95 244 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 89 219 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 85 237 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 93 221 0 0 100 0 0  0 0 853424 69008 4648 29128 0 0 0 0 109 244 1 0 100 0 0 |

* + 1. **Memory 사용량**

사용서비스 되어지는 Memory 사용량 체크를 위하여 다음의 명령어를 통해 모니터링을 진행합니다.

vmstat 1 100

항목 설명

total : 전체 메모리 용량

used : 메모리 사용량

free : 여유 메모리 공간

shared : 공유메모리 사용량

buffers : 버퍼 메모리 공간

cached : 캐쉬메모리 사용량

정상 여부 확인 : free 최소 여유공간이 전체 메모리 대비 최소 10% 이상일 경우

|  |
| --- |
| **$ free -m**  total used free shared buffers cached  Mem: 1877 1810 66 0 0 30  -/+ buffers/cache: 1779 98  Swap: 4031 831 3200 |

* + 1. **프로세스 과부하 cpu 여부 확인**

서비스 되어지는 process의 CPU사용량이 과부하 인지 여부를 체크합니다.

***top***

항목 설명

top 명령을 통해서 해당 프로세스의 현재 CPU 정보를 확인합니다.

정상 여부 확인 : 프로세스당 CPU 사용이 많은지 여부를 확인합니다.

|  |
| --- |
| **$ top**  top - 03:14:41 up 14 days, 12:59, 4 users, load average: 0.19, 0.13, 0.09  Tasks: 336 total, 1 running, 335 sleeping, 0 stopped, 0 zombie  Cpu(s): 3.7%us, 3.5%sy, 0.0%ni, 92.1%id, 0.7%wa, 0.0%hi, 0.0%si, 0.0%st  Mem: 16271536k total, 15962368k used, 309168k free, 697304k buffers  Swap: 16777208k total, 529588k used, 16247620k free, 6697584k cached  PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND  28336 jboss 20 0 4003m 798m 15m S 1.0 5.0 20:21.52 /opt/java/jdk1.6.0\_45/bin/java -D[Scouter] -XX:+UseCompr  1732 jboss 20 0 6051m 115m 11m S 0.0 0.7 1:29.62 java -server -cp lib/\* org.ngrinder.NGrinderAgentStarter --  31319 jboss 20 0 915m 7096 1216 S 0.0 0.0 0:00.00 /app/jboss/jboss-ews-2.0.1/httpd/sbin/httpd.worker -f /opt/we |

* + 1. **Disk Usage**

서비스 되어지는 Disk 사용량을 파악하여 여유 공간을 확인합니다.

***df -h***

정상 여부 확인 : 서비스 사용 영역 및 root 파일 시스템의 사용율이 80% 인 경우 위험으로 간주합니다.

|  |
| --- |
| **$ df -h**  Filesystem Size Used Avail Use% Mounted on  /dev/mapper/vg\_rhevmtest-lv\_root 26G 23G 1.9G 93% /scouter  tmpfs 939M 72K 939M 1% /dev/shm  /dev/vda1 485M 38M 422M 9% /boot |

# **장애 유형별 대응 방법**

## **Scouter 접속 실패시**

* + 1. **리소스 및 프로세스 체크**

본 문서 대분류 3. 항목 Scouter 서버 정상 동작 여부 확인을 진행

|  |
| --- |
| ***Thread dump를 이용한 서비스 dump 생성 후 서버 재기동*** |

* + 1. **Too many Openfiles**

에러 상황 :

Too Many Open Files 에러가 발생하는 상황 ( file 이 많이 열리거나, Socket 이 부족한 경우 )

|  |
| --- |
| #### 파일 오픈이 많은 경우  java.io.IOException: Too many open files at java.lang.UNIXProcess.forkAndExec(Native Method) at java.lang.UNIXProcess.(UNIXProcess.java:54) at java.lang.UNIXProcess.forkAndExec(Native Method) at java.lang.UNIXProcess.(UNIXProcess.java:54) at java.lang.Runtime.execInternal(Native Method) at java.lang.Runtime.exec(Runtime.java:551)  #### 소켓 부족시  java.net.SocketException: Too many open files  at java.net.PlainSocketImpl.accept(Compiled Code)  at java.net.ServerSocket.implAccept(Compiled Code |

해결방안 :

|  |
| --- |
| ulimit –a 를 통한 시스템 리소스 확인 후 limits 제한 설정을 변경    [jboss@KVM2 node1]$ ulimit –a  core file size (blocks, -c) 0  data seg size (kbytes, -d) unlimited  scheduling priority (-e) 0  file size (blocks, -f) unlimited  pending signals (-i) 62338  max locked memory (kbytes, -l) 64  max memory size (kbytes, -m) unlimited  open files (-n) 35565  pipe size (512 bytes, -p) 8  POSIX message queues (bytes, -q) 819200  real-time priority (-r) 0  stack size (kbytes, -s) 10240  cpu time (seconds, -t) unlimited  max user processes (-u) 1024  virtual memory (kbytes, -v) unlimited  file locks (-x) unlimited  vi /etc/security/limits.conf  root soft nofile 65536  root hard nofile 65536  root soft nproc 20680  root hard nproc 20680  root soft stack unlimited  root hard stack unlimited  scouter - nofile 65536  scouter - memlock 10240 |

* + 1. **Server Crash**

에러 상황 :

Scouter 서버가 예기치 않게 중단이 되는 경우

|  |
| --- |
| # 에러가 발생하는 상황  WAS Server Crash  JVM Crash  머신 Crash  HotSpot Error  # 에러 분석방법  core파일이 생성되었는지 체크  생성되지 않았다면 생성되도록 설정(OS)  core파일로 발생할 당시의 Crash원인을 분석 |

Core 파일이 생성되지 않았을 경우 :

|  |
| --- |
| # core 파일 사이즈 limit을 체크한다  ulimit –c 로 확인  Solaris의 경우 /etc/system의 sys:coredumpsize 확인  Linux의 경우 /etc/security/limits.conf  HP-UX의 경우 maxdsiz  🡺 OS 엔지니어에게 요청한다 |

*★ 생성 된 core 파일을 통한 분석 진행 필요.*

## **대시보드 모니터링 실패**

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.

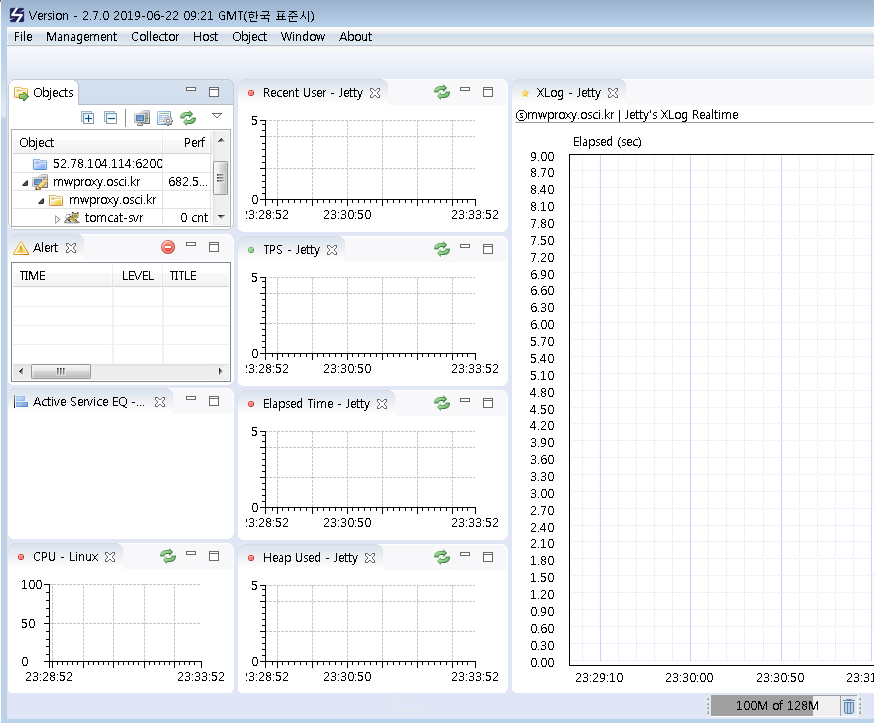
<정상 서비스 화면>

< 주요 대시보드 항목 >

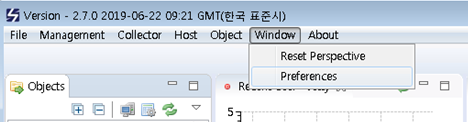
|  |  |
| --- | --- |
| **시스템 사용자 정보** | * 통신 단말 사용자 수 * 방문자 수 * 액티브 사용자 수 |
| **시스템 처리 상태** | * 초당 처리 건수 * 시간당 호출 건수 * 평균 응답시간 |
| **지원상태** | * 시스템 CPU 사용량 * JVM CPU 사용량 * 힙 메모리 사용량 |
| **장애/에러** | * 시스템 장애 * 프로그램 에러 |

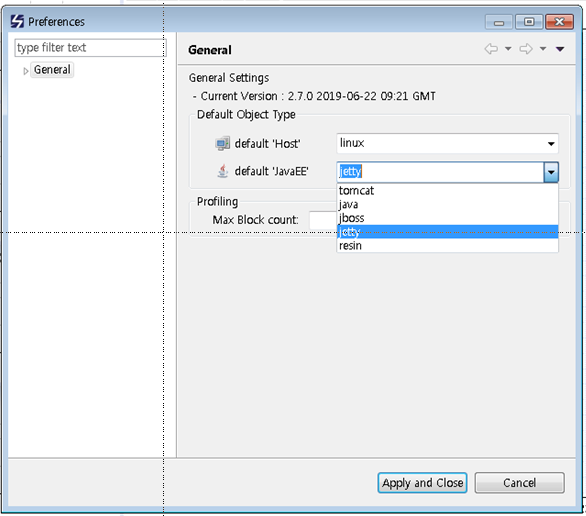
* + 1. **오브젝트 타입 오류에 의한 모니터링 실패**

아래의 빨간색 항목과 같이 정상적인 모니터링이 되지 않을 상황 – OBJ



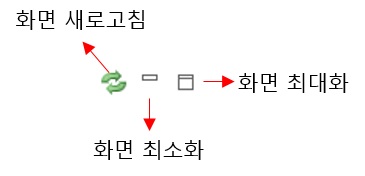
해결 방법 : 서비스 Object type 설정





* + 1. **모니터링 인터페이스 오류**

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



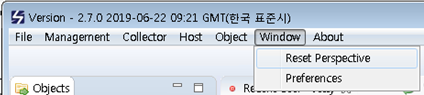
오류 상황 :

해당 설정으로 화면조정을 할 경우 간혹 서비스 오류를 통해서 대시보드 화면이 정상적이지 않은 케이스가 발생한다.

* 해결 방안 :

잘못된 서비스 요청으로 인한 서비스 화면이 오류시 초기 Dashbaord 환경으로 화면을 Reset 할 수 있다.

Windows > Reset Perspective 선택



# **APPENDICES**

## **Document Guidelines**

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

## **Project Migration Report Sections Omitted**

N/A

