openmaru
APM

WAS에서 발생하는 주요 문제에 대한 트러블슈팅 방법

- Confidential -
Opennaru, Inc. © 2016 | All Rights Reserved.

Troubleshooting 공통 프로세스

장애발생

상황수집
(긴급조치,로그수집)

진단&모니터링

원인분석&재현

조치

복구

증상과 원인

진단 및 모니터링

조치

방지대책

Opennaru, Inc. © 2017 | All Rights Reserved.

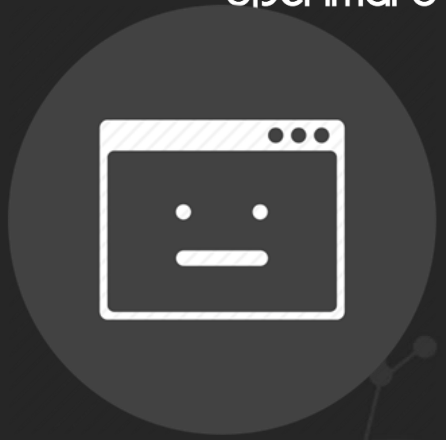
트러블 슈팅의 기본

- **로그 등 관련 자료는 충분히 확보해야**
  - 로그는 첫 번째 문제부터 살펴봐야
- **섣부른 추측은 금물**
  - 전해들은 말을 100% 믿지 말라
  - 증거에 의존하여 분석
  - 문서를 100% 믿지 말라
- **문제분석**
  - 문제를 분리하는 방법(Divide & Conquer)
  - 코드 샘플링을 통한 분리 / 재현
  - 재현 & 테스트 & 재현 & 테스트

Opennaru, Inc. © 2017 | All Rights Reserved.

Server Hang이란?

openmaru



- 서버가 전혀 Response를 주지 못함 ➔ 서버 Hang
- 느리더라도 Response를 준다 ➔ 서버 Slowdown

- 서버 Hang의 증상
  - Request가 처리되지 않는다
  - 새로운 Request를 받지 못하고 기존 Request가 Timeout되거나 응답이 없는 상황
  - 서버가 아무 작업도 하지 않는 것 같다
  - 서버의 증상:
    - Hang이 오래 지속되면 JVM Crash될 수도 있다(항상은 아님)
    - 그냥 두면 Hang이 풀리는 경우도 있다(리소스 경합이 발생하여 Hang이 발생한 경우에는 Resource가 풀리면 Hang이 풀린다)
    - 대부분 Hang이 발생된 후에는 관리자의 조치가 없으면 Hang 상태로 또 한이 남아있게 된다

증상과 원인

Opennaru, Inc. © 2017 | All Rights Reserved.

APM = Application Performance Management

openmaru

3티어구조에서성능문제발생시신속한진단과조치를위한성능관리도구!

- OPENMARU APM 은 실시간으로 어플리케이션 성능을 모니터링 하여 안정적인 시스템 운영을 지원하는 어플리케이션 성능 관리(Application Performance Management, APM) 솔루션
- 웹 애플리케이션의 내부를 실시간으로 모니터링 하면서 사용자들에게 쾌적하게 서비스가 제공되고 있는지를 "가용성"과 "성능"의 관점에서 분석
- 애플리케이션 서버에 에이전트를 배포 페이지 요청의 응답 시간과 쿼리에 대한 데이터베이스 응답 시간을 모니터링하여 "서비스 장애 감지", "장애 원인 특정 시간 단축, 서비스 부하의 원인을 파악"

Opennaru, Inc. © 2017 | All Rights Reserved.

정상적인 상황에서의 APM 대시보드

Opennaru, Inc. © 2017 | All Rights Reserved.

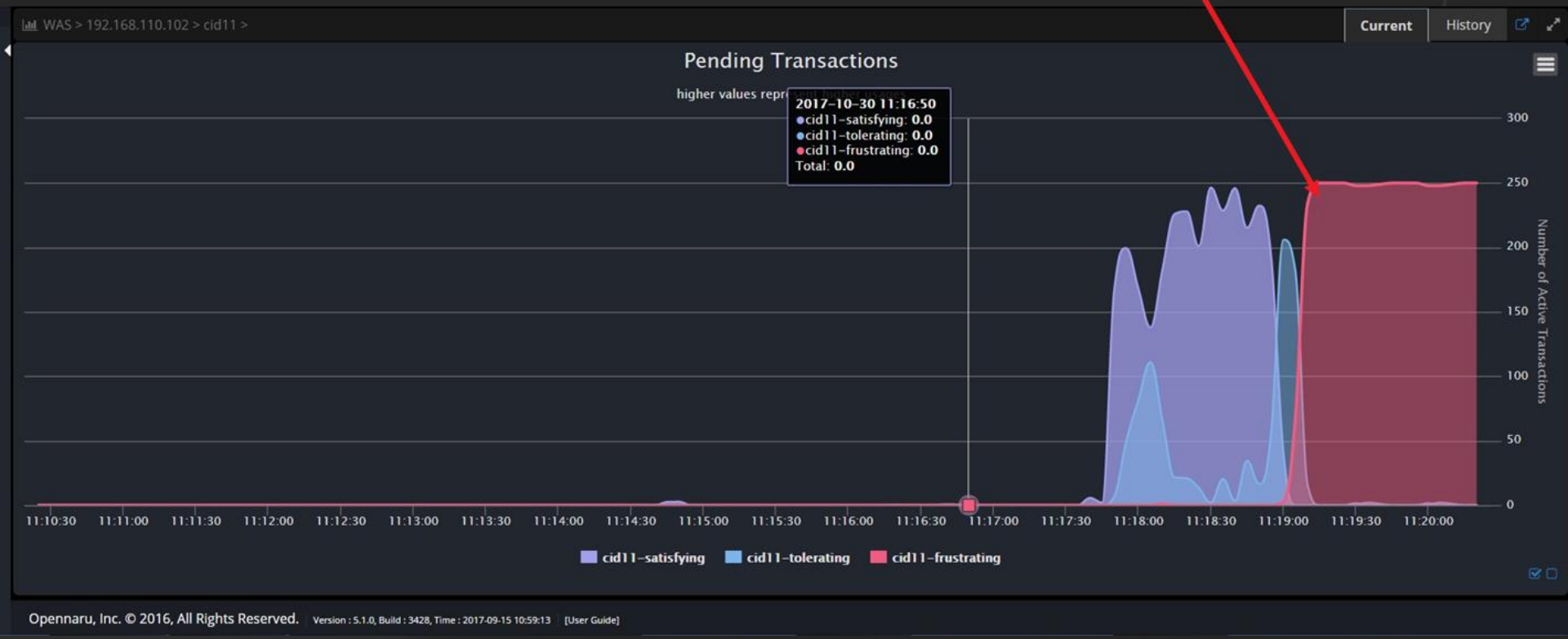
장애 상황에서의 APM 대시보드

Opennaru, Inc. © 2017 | All Rights Reserved.

대시보드에서 Pending 상황을 나중에 확인방법

- WAS → Instance → [호스트명] → [인스턴스 이름] → 애플리케이션 → 지연 트랜잭션
- 애플리케이션 그룹 → [그룹명] → 지연 트랜잭션

Pending Request가 250개 급격히 쌓임



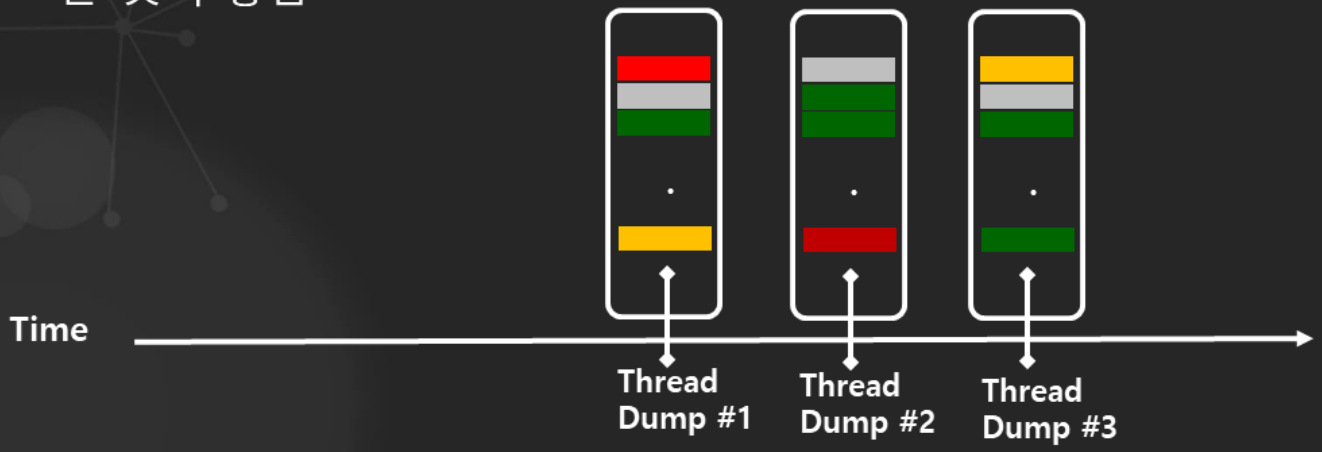
Opennaru, Inc. © 2017 | All Rights Reserved.

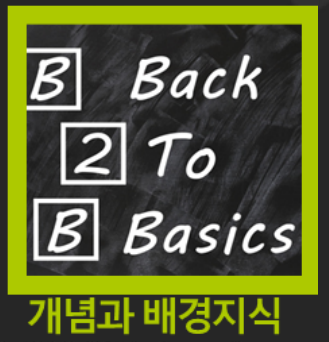
Opennaru, Inc. © 2016 | All Rights Reserved.

Java 스레드 덤프

- **스레드 덤프란?**
  - JVM 내부에서 현재 수행중인 스레드의 SnapShot을 텍스트로 출력
  - 서버 Hang, 서버 Slowdown등 서버에서 발생하는 문제를 해결하는 가장 유용한 도구
  - 약 3~5초간격으로 3~5회 연속해서 스레드 덤프를 받아야 분석이 가능하다

- **언제 받아야 하나?**
  - 운영 중인 서버에서도 수행가능(많은 시간이 걸리지 않는 작업임)
  - 서버 Hang, Slowdown시, 서버가 Crash되거나 OutOfMemory가 발생한 직후에 받는 것이 좋음

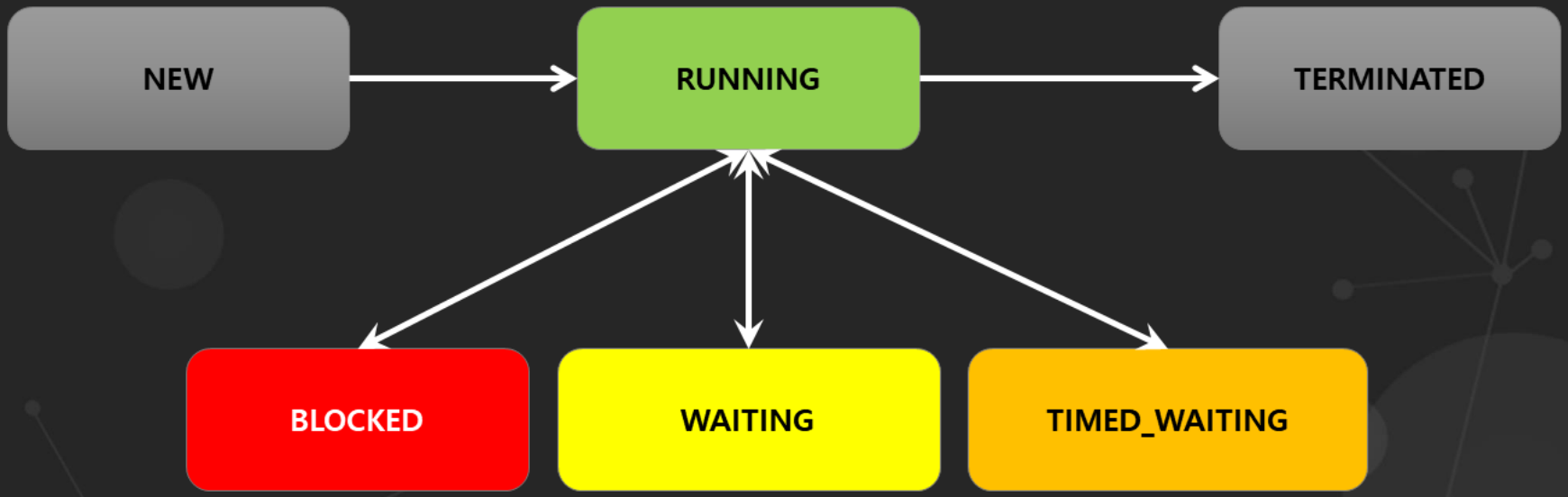




Opennaru, Inc. © 2017 | All Rights Reserved.

Thread 상태

- java.lang.Thread.State

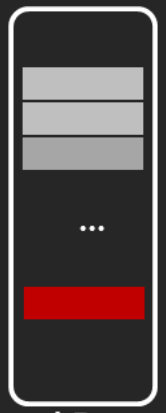


- NEW – 생성되어있지만 실행되지 않은 상태
- RUNNING – 작업 수행 중
- BLOCKED – 다른 스레드의 Lock 해제를 기다리는 중
- WAITING – wait(), join(), park() 메소드로 대기 중
- TIMED_WAITING – WAITING과 같지만, 정해진 시간만 대기
- TERMINATED – 종료됨

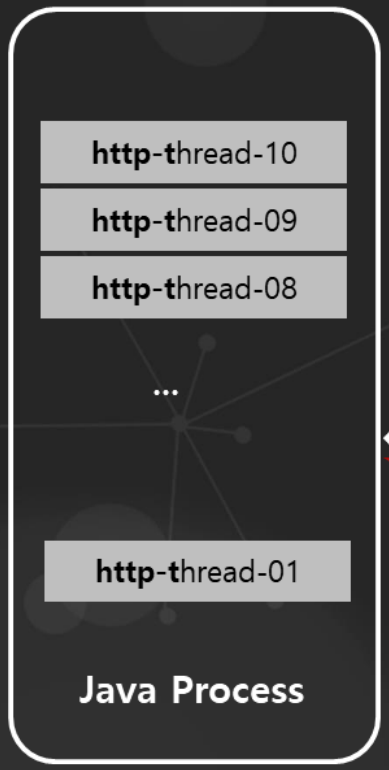
Back 2 To Basics
개념과 배경지식
- Confidential -
Opennaru, Inc. © 2017 | All Rights Reserved.

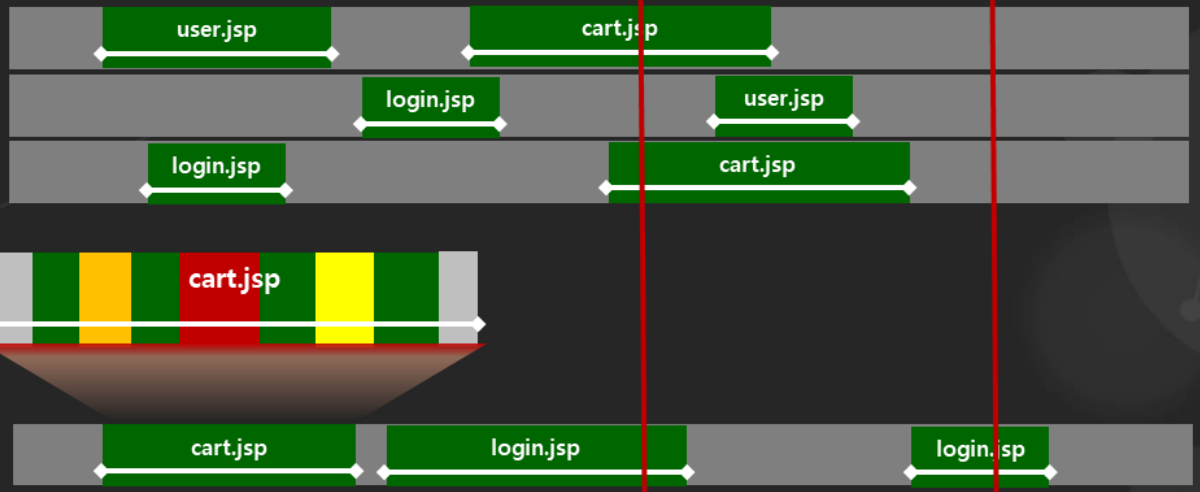
Java Thread Dump의 이해

Thread Dump #1



Thread Dump #2





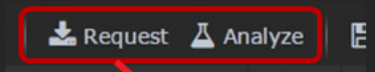
Back 2 To Basics

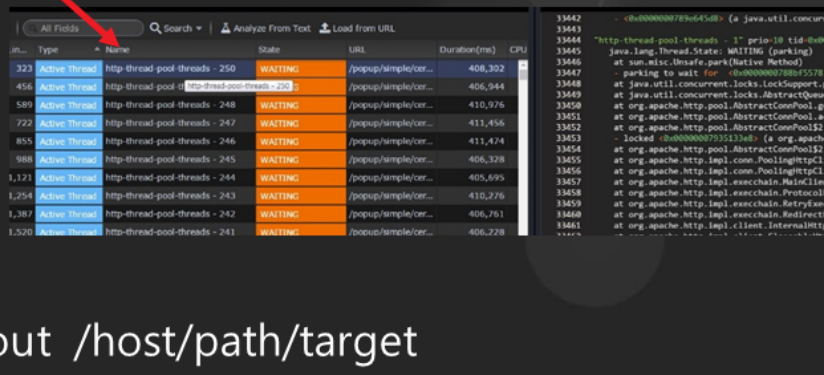
개념과 배경지식

Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프 받는 방법은?

- **Linux / UNIX 시스템에서 스레드 덤프**
  - kill -3 <*PID*>
  - 일반적을 nohup 명령을 이용하여 시작하기 때문에 nohup.out파일에 stdout 로그가 남게 됨
  - 윈도우에서는 Ctrl + Break
  - OPENMARU Installer 설치 WAS : s1 → ./dump.sh
  - **OPENMARU APM 화면에서 버튼 클릭(장애시 자동 생성)**





- **Docker와 Openshift 같은 환경**
  - **스레드 덤프을 어떻게 받나?**
    - docker exec -it c4343003c1 /bin/bash
      ~# kill -3 <PID>
  - **내용을 어떻게 복사? nohup 로그는 남기나?**
    - docker cp c4343003c1:/file/path/nohup.out /host/path/target
  - OPENMARU APM은 Openshift, Docker를 지원 → 버튼 클릭 & 분석 UI 제공

Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프의 정보 구성

- 스레드 덤프의 각 스레드의 정보 구성

스레드 이름 | 우선순위 | 스레드ID | OS 스레드ID | 스레드 상태

```
"http-thread-pool-threads - 250" prio=10 tid=0x00007fab9c192800 nid=0xdbc9 waiting on condition [0x00007faa6ce2c000]
   java.lang.Thread.State: WAITING (parking)
        at sun.misc.Unsafe.park(Native Method)
        - parking to wait for  <0x000000078b428170> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
        at java.util.concurrent.locks.LockSupport.park(LockSupport.java:186)
        at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.await(AbstractQueuedSynchronizer.java:2043)
        at java.util.concurrent.LinkedBlockingQueue.take(LinkedBlockingQueue.java:442)
        at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1068)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1130)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:615)
        at java.lang.Thread.run(Thread.java:744)
        at org.jboss.threads.JBossThread.run(JBossThread.java:122)
```

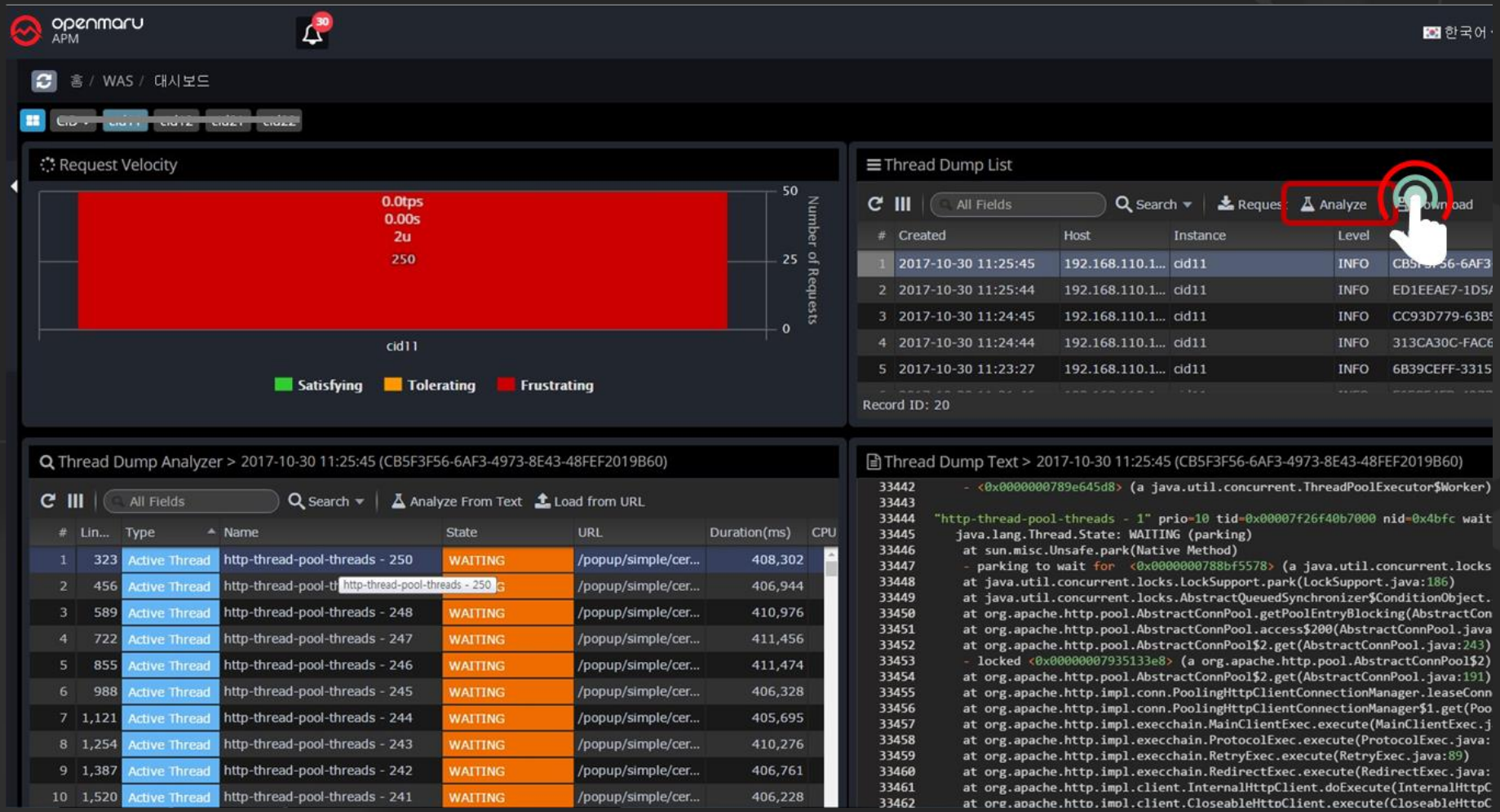
스레드 호출 Stack

B Back 2 To B Basics
개념과 배경지식

Opennaru, Inc. © 2017 | All Rights Reserved.

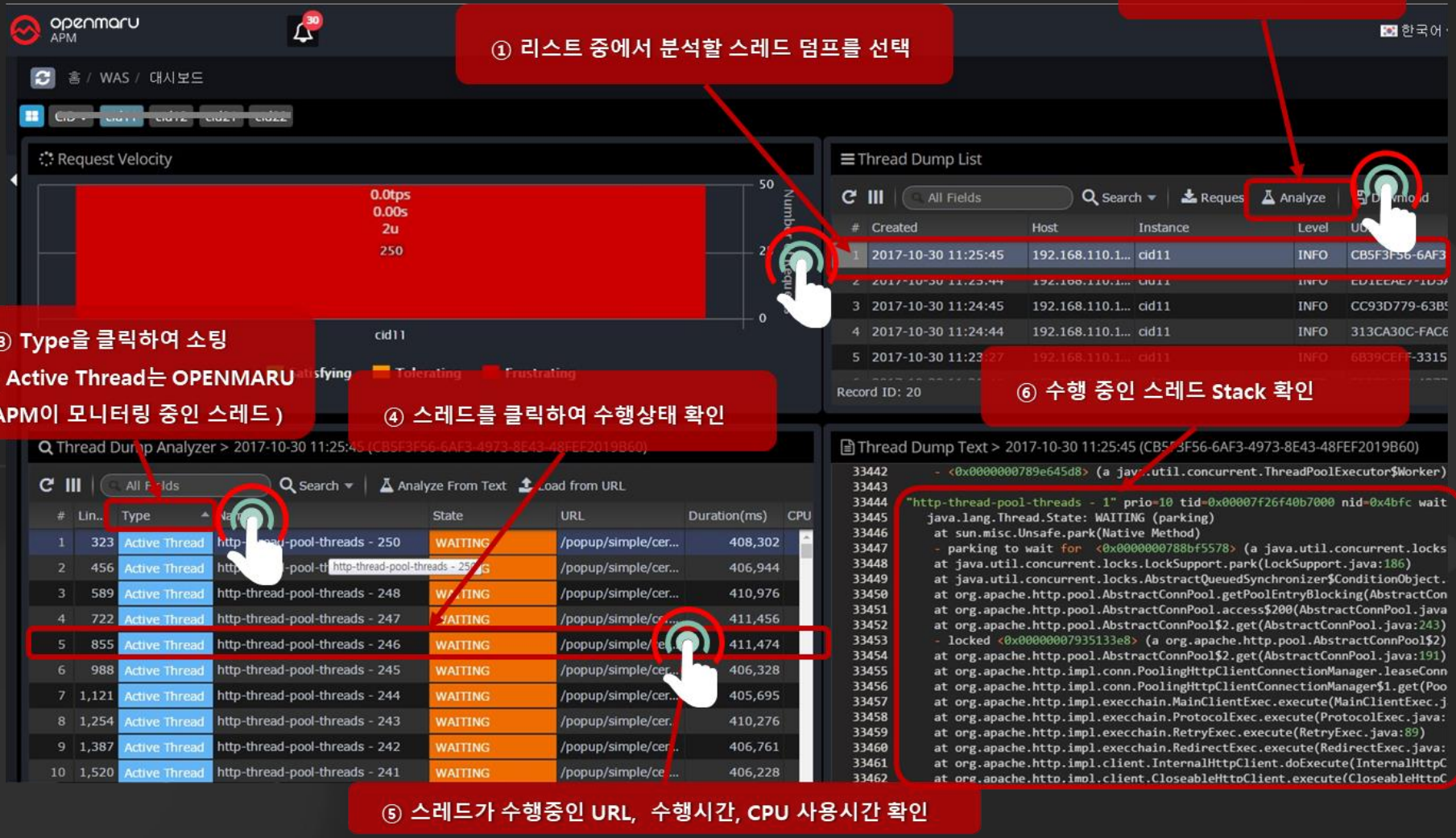
OPENMARU APM에서 스레드 덤프 생성 방법

- 대시보드 → Request Velocity의 빨간색 클릭 → '요청' 버튼 클릭
- WAS → Instance → [호스트명] → [인스턴스 이름] → 스레드 덤프 분석 → '요청' 버튼 클릭
- <자동생성 조건> 이벤트 → 경고정책 → Pending Transaction이 Critical 조건이 되면 자동생성



Opennaru, Inc. © 2017 | All Rights Reserved.

OPENMARU APM에서 스레드 덤프 간단 분석 방법



Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프 분석 패턴

openmaru
APM

- Confidential -

Opennaru, Inc. © 2016 | All Rights Reserved.

스레드 덤프 분석 주요 패턴

DB 쿼리가 느릴 때

HTTP 호출이 느릴 때

Logging 처리가 느릴 때

파일 I/O가 느릴 때

Lock이 걸릴 때

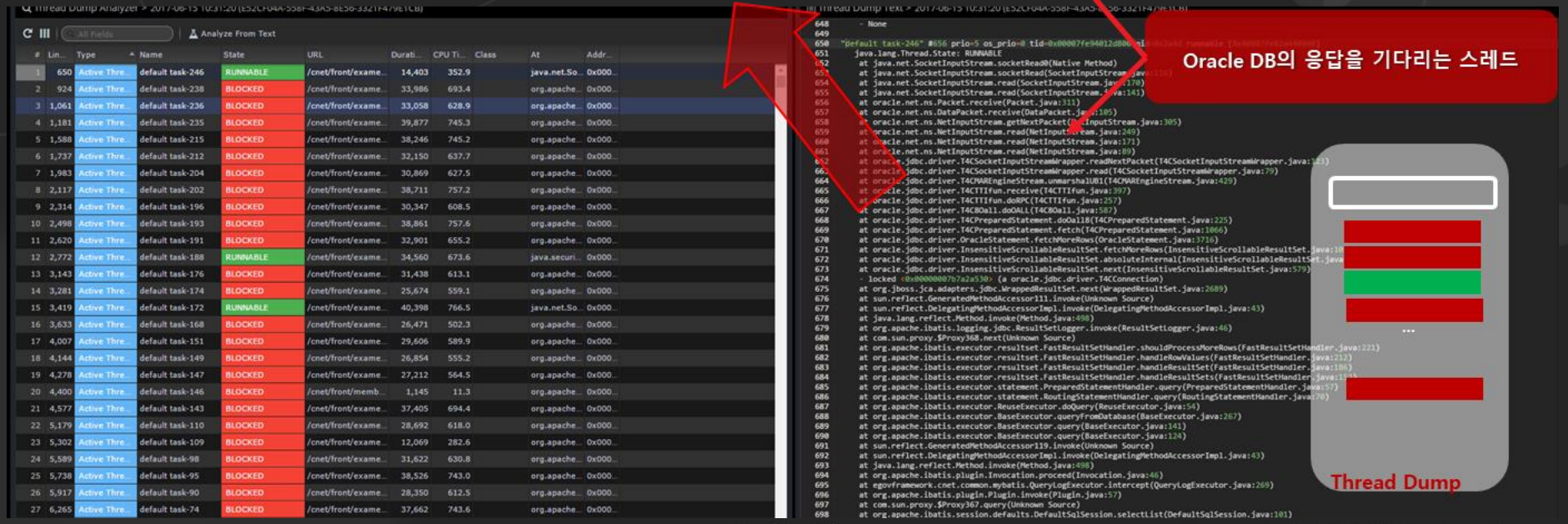
Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프 분석 주요 패턴

openmaru

DB 쿼리가 느릴 때

```
650  "default task-246" #656 prio=5 os_prio=0 tid=0x00007fe94012d800 nid=0x2a4d runnable [0x00007fe82a440000]
651     java.lang.Thread.State: RUNNABLE
652        at java.net.SocketInputStream.socketRead0(Native Method)
653        at java.net.SocketInputStream.socketRead(SocketInputStream.java:116)
654        at java.net.SocketInputStream.read(SocketInputStream.java:170)
655        at java.net.SocketInputStream.read(SocketInputStream.java:141)
656        at oracle.net.ns.Packet.receive(Packet.java:311)
657        at oracle.net.ns.DataPacket.receive(DataPacket.java:105)
658        at oracle.net.ns.NetInputStream.getNextPacket(NetInputStream.java:305)
659        at oracle.net.ns.NetInputStream.read(NetInputStream.java:249)
660        at oracle.net.ns.NetInputStream.read(NetInputStream.java:171)
661        at oracle.net.ns.NetInputStream.read(NetInputStream.java:89)
662        at oracle.jdbc.driver.T4CSocketInputStreamWrapper.readNextPacket(T4CSocketInputStreamWrapper.java:123)
663        at oracle.jdbc.driver.T4CSocketInputStreamWrapper.read(T4CSocketInputStreamWrapper.java:79)
```



Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프 분석 방법 - DB 연결이 오래 걸릴 때

openmaru

DB getConnection이 느릴 경우

```
1894  "defa...task-2...
1895      java.lang.Thre...
1896      at sun.misc.Unsafe.park(Native Method)
1897      - parking to wait for  <0x00000006c2832210> (a java.util.concurrent.Semaphore$FairSync)
1898      at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport...
1899      at java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedNanos(AbstractQueuedSynchroni
1900      at java.util.concurrent.locks.AbstractQueuedSynchronizer.tryAcquireSharedNanos(AbstractQueuedSynchron
1901      at java.util.concurrent.Semaphore.tryAcquire(Semaphore.java:409)
1902      at org.jboss.jca.core.connectionmanager.pool.api.Semaphore.tryAcquire(Semaphore.java:68)
1903      at org.jboss.jca.core.connectionmanager.pool.mcp.SemaphoreConcurrentLinkedDequeManagedConnectionPool.
1904      at org.jboss.jca.core.connectionmanager.pool.AbstractPool.getSimpleConnection(AbstractPool.java:632)
1905      at org.jboss.jca.core.connectionmanager.pool.AbstractPool.getConnection(AbstractPool.java:604)
1906      at org.jboss.jca.core.connectionmanager.AbstractConnectionManager.getManagedConnection(AbstractConnec
1907      at org.jboss.jca.core.connectionmanager.tx.TxConnectionManagerImpl.getManagedConnection(TxConnectionM
1908      at org.jboss.jca.core.connectionmanager.AbstractConnectionManager.allocateConnection(AbstractConnecti
1909      at org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection(WrapperDataSource.java:138)
1910      at org.jboss.as.connector.subsystems.datasources.WildFlyDataSource.getConnection(WildFlyDataSource.ja
1911      at org.springframework.jdbc.datasource.DataSourceUtils.doGetConnection(DataSourceUtils.java:111)
1912      at org.springframework.jdbc.datasource.DataSourceUtils.getConnection(DataSourceUtils.java:77)
1913      at org.mybatis.spring.transaction.SpringManagedTransaction.openConnection(SpringManagedTransaction.ja
```

Connection Pool 획득 요청 이후 Pool이 부족하여 대기하고 있는 상태

스레드 덤프 분석기 > 2019-03-11 18:03:20 (26DFF023-4EB4-4654-8C17-06C06CDFAFE3)

모든 필드 | 검색 | 텍스트로 분석 | URL에서 읽기

#	Line#	Type	Name	State	URL	Durati...	CPU
1	536	Active Thread	taskExecutor-8	TIMED_WAITING		5,747	
2	963	Active Thread	default task-248	TIMED_WAITING	/bms/dct/getNotic...	3,797	
3	1,204	Active Thread	default task-246	TIMED_WAITING	/blg/message/mes...	6,297	
4	1,343	Active Thread	default task-244	TIMED_WAITING	/blg/message/mes...	7,399	
5	1,510	Active Thread	default task-240	RUNNABLE	/blg/message/mes...	11,773	
6	1,680	Active Thread	default task-236	TIMED_WAITING	/blg/message/mes...	2,496	
7	1,894	Active Thread	default task-229	TIMED_WAITING	/bms/dct/editrepo...	4,287	
8	2,049	Active Thread	default task-225	TIMED_WAITING	/blg/message/mes...	3,917	
9	2,671	Active Thread	default task-203	WAITING	/bms/dctenf/Bms...	52	
10	2,804	Active Thread	default task-199	TIMED_WAITING	/blg/message/mes...	5,460	
11	3,198	Active Thread	default task-180	TIMED_WAITING	/blg/message/mes...	528	
12	3,397	Active Thread	default task-174	TIMED_WAITING	/bms/dct/editrepo...	4,912	

Record ID: 695

스레드 덤프 텍스트 > 2019-03-11 18:03:20 (26DFF023-4EB4-4654-8C17-06C06CDFAFE3)

```
1892      - None
1893
1894  "default task-229" #720 prio=5 os_prio=0 tid=0x000000000c411000 nid=0x79bb waiting on condition [0x00007f
1895      java.lang.Thread.State: TIMED_WAITING (parking)
1896      at sun.misc.Unsafe.park(Native Method)
1897      - parking to wait for  <0x00000006c2832210> (a java.util.concurrent.Semaphore$FairSync)
1898      at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
1899      at java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedNanos(AbstractQueuedSynchroni
1900      at java.util.concurrent.locks.AbstractQueuedSynchronizer.tryAcquireSharedNanos(AbstractQueuedSynchron
1901      at java.util.concurrent.Semaphore.tryAcquire(Semaphore.java:409)
1902      at org.jboss.jca.core.connectionmanager.pool.api.Semaphore.tryAcquire(Semaphore.java:68)
1903      at org.jboss.jca.core.connectionmanager.pool.mcp.SemaphoreConcurrentLinkedDequeManagedConnectionPool.
1904      at org.jboss.jca.core.connectionmanager.pool.AbstractPool.getSimpleConnection(AbstractPool.java:632)
1905      at org.jboss.jca.core.connectionmanager.pool.AbstractPool.getConnection(AbstractPool.java:604)
1906      at org.jboss.jca.core.connectionmanager.AbstractConnectionManager.getManagedConnection(AbstractConnec
1907      at org.jboss.jca.core.connectionmanager.tx.TxConnectionManagerImpl.getManagedConnection(TxConnectionM
1908      at org.jboss.jca.core.connectionmanager.AbstractConnectionManager.allocateConnection(AbstractConnecti
1909      at org.jboss.jca.adapters.jdbc.WrapperDataSource.getConnection(WrapperDataSource.java:138)
1910      at org.jboss.as.connector.subsystems.datasources.WildFlyDataSource.getConnection(WildFlyDataSource.ja
1911      at org.springframework.jdbc.datasource.DataSourceUtils.doGetConnection(DataSourceUtils.java:111)
1912      at org.springframework.jdbc.datasource.DataSourceUtils.getConnection(DataSourceUtils.java:77)
1913      at org.mybatis.spring.transaction.SpringManagedTransaction.openConnection(SpringManagedTransaction.ja
1914      at org.mybatis.spring.transaction.SpringManagedTransaction.getConnection(SpringManagedTransaction.jav
1915      at org.apache.ibatis.executor.BaseExecutor.getConnection(BaseExecutor.java:279)
1916      at org.apache.ibatis.executor.SimpleExecutor.prepareStatement(SimpleExecutor.java:72)
1917      at org.apache.ibatis.executor.SimpleExecutor.doUpdate(SimpleExecutor.java:47)
1918      at org.apache.ibatis.executor.BaseExecutor.update(BaseExecutor.java:105)
```


Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프 분석 주요 패턴

openmaru

HTTP 호출이 느릴 때

```
2589        - None
2590
2591    "ajp-thread-pool-threads - 363" prio=10 tid=0x00007f977c250800 nid=0x3bbb runnable [0x
2592       java.lang.Thread.State: RUNNABLE
2593        at java.net.SocketInputStream.socketRead0(Native Method)
2594        at java.net.SocketInputStream.read(SocketInputStream.java:153)
2595        at java.net.SocketInputStream.read(SocketInputStream.java:122)
2596        at java.io.BufferedInputStream.fill(BufferedInputStream.java:235)
2597        at java.io.BufferedInputStream.read1(BufferedInputStream.java:275)
2598        at java.io.BufferedInputStream.read(BufferedInputStream.java:334)
2599        - locked <0x000000079998c960> (a java.io.BufferedInputStream)
2600        at sun.net.www.MeteredStream.read(MeteredStream.java:134)
2601        - locked <0x000000079998c8d0> (a sun.net.www.http.KeepAliveStream)
2602        at java.io.FilterInputStream.read(FilterInputStream.java:133)
2603        at sun.net.www.protocol.http.HttpURLConnection$HttpInputStream.read(HttpURLConnect
2604        at sun.net.www.protocol.http.HttpURLConnection$HttpInputStream.read(HttpURLConnect
2605        at com.nhncorp.keris.ebook.common.controller.FileDownloadURLView.renderMergedOutpu
2606        at org.springframework.web.servlet.view.AbstractView.render(AbstractView.java:250)
2607        at org.springframework.web.servlet.DispatcherServlet.render(DispatcherServlet.java
```

Java의 HttpUrlConnection가 호출한 URL의 Socket 응답을 기다리는 중

스레드 덤프 시점에 559초 가량 실행중

스레드 덤프 분석기 > 2018-10-26 22:09:52 (F563FFAE-CC45-4533-89F4-E4CD2DE9784F)

모든 필드 | 검색 | 텍스트로 분석 | URL에서 읽기

#	Line#	Type	Name	State	URL	Durati...	CPU T
1	1,862	Active Thread	ajp-thread-poo...	RUNNABLE	/securityFileDownl...	264,67...	
2	2,591	Active Thread	ajp-thread-poo...	RUNNABLE	/securityFileDownl...	559,79...	
3	17	Lock		locked			
4	32	Lock		locked			
5	50	Lock		Locked ownable ...			
6	283	Lock		locked			
7	291	Lock		Locked ownable ...			
8	300	Lock		locked			
9	308	Lock		Locked ownable ...			
10	1,870	Lock		locked			
11	1,872	Lock		locked			

264,67...

559,79...

Java의 HttpUrlConnection은 Timeout / Pooling을 지원하지 않아 HttpClient로 변경해야 함

Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프 분석 주요 패턴

openmaru

HTTP 호출이 느릴 때

```
42873
42874  "http-10.129.2.144:8080-257" daemon prio=10 tid=0x00007f38a01f8800 nid=0x11dae runnable [0x00007f37d2a2f000]
42875     java.lang.Thread.State: RUNNABLE
42876        at java.net.SocketInputStream.socketRead0(Native Method)
42877        at java.net.SocketInputStream.read(SocketInputStream.java:153)
42878        at java.net.SocketInputStream.read(SocketInputStream.java:122)
42879        at sun.security.ssl.InputRecord.readFully(InputRecord.java:442)
42880        at sun.security.ssl.InputRecord.read(InputRecord.java:480)
42881        at sun.security.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:944)
42882        - locked <0x00000007ada7a698> (a java.lang.Object)
42883        at sun.security.ssl.SSLSocketImpl.readDataRecord(SSLSocketImpl.java:901)
42884        at sun.security.ssl.AppInputStream.read(AppInputStream.java:102)
42885        - locked <0x00000007ada8be90> (a sun.security.ssl.AppInputStream)
42886        at org.apache.http.impl.io.SessionInputBufferImpl.streamRead(SessionInpu
42887        at org.apache.http.impl.io.SessionInputBufferImpl.fillBuffer(SessionInpu
42888        at org.apache.http.impl.io.SessionInputBufferImpl.readLine(SessionInputB
```

Http Client가 호출한 URL의 Socket 응답을 기다리는 중

스레드 덤프 시점에 650초 가량 실행중

#	Line #	Type	Name	State				
1	42,874	Active Thread	http-10.129.2...					
2	54,758	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	652,996	9.1	java.net.So... 0x00...
3	43,443	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	652,743	8.6	java.net.So... 0x00...
4	38,954	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	652,292	9.1	java.net.So... 0x00...
5	37,328	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	648,523		
6	36,558	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	648,069		
7	41,684	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	647,699		
8	60,198	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	647,561		
9	60,058	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	645,937		
10	50,316	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	645,861		
11	48,566	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	644,622		
12	44,996	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	640,994	7.8	java.net.So... 0x00...
13	47,306	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	640,812	7.9	java.net.So... 0x00...
14	42,524	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	639,535	10.1	java.net.So... 0x00...
15	43,723	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	638,342	8.7	java.net.So... 0x00...
16	49,616	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	637,802	9.0	java.net.So... 0x00...
17	49,126	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	637,215	9.4	java.net.So... 0x00...
18	52,696	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	636,274	7.8	java.net.So... 0x00...
19	57,317	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	636,165	7.5	java.net.So... 0x00...
20	56,882	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	635,065	8.9	java.net.So... 0x00...
21	44,716	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	634,980	7.5	java.net.So... 0x00...
22	42,594	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	633,974	7.5	java.net.So... 0x00...
23	53,972	Active Thread	http-10.129.2...	RUNNABLE	/common/office365/getMailInfo.ajax	633,256	8.2	java.net.So... 0x00...

653,655

652,996

대부분의 스레드는 /common/office365/getMailInfo.ajax 호출이 응답이 없는 상황

Opennaru, Inc. © 2017 | All Rights Reserved.

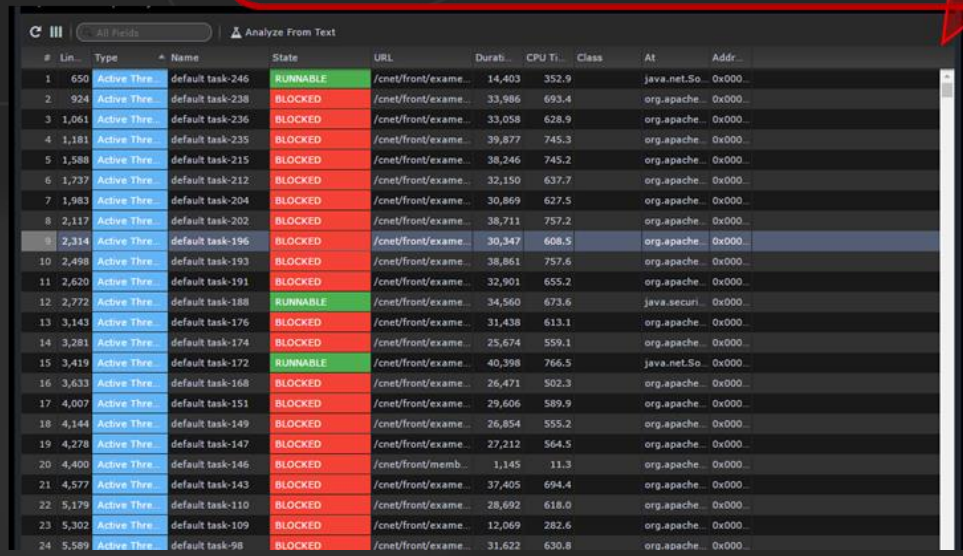
스레드 덤프 분석 주요 패턴

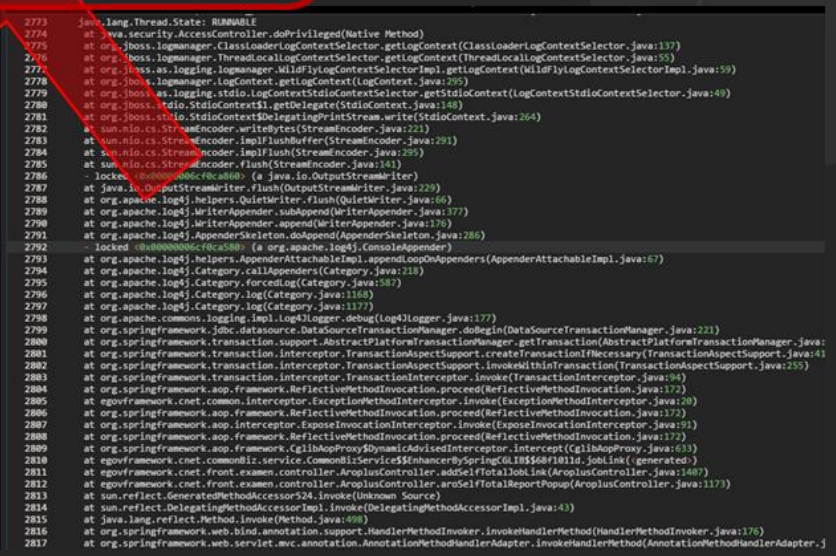
openmaru

Logging 처리가 느릴 때

```
2773    java.lang.Thread.State: RUNNABLE
2774     at java.security.AccessController.doPrivileged(Native Method)
2775     at org.jboss.logmanager.ClassLoaderLogContextSelector.getLogContext(ClassLoaderLogContextSelector.java:137)
2776     at org.jboss.logmanager.ThreadLocalLogContextSelector.getLogContext(ThreadLocalLogContextSelector.java:55)
2777     at org.jboss.as.logging.logmanager.WildFlyLogContextSelectorImpl.getLogContext(WildFlyLogContextSelectorImpl.java:59)
2778     at org.jboss.logmanager.LogContext.getLogContext(LogContext.java:295)
2779     at org.jboss.as.logging.stdio.LogContextStdioContextSelector.getStdioContext(LogContextStdioContextSelector.java:49)
2780     at org.jboss.stdio.StdioContext$1.getDelegate(StdioContext.java:148)
2781     at org.jboss.stdio.StdioContext$DelegatingPrintStream.write(StdioContext.java:264)
2782     at sun.nio.cs.StreamEncoder.writeBytes(StreamEncoder.java:221)
2783     at sun.nio.cs.StreamEncoder.implFlushBuffer(StreamEncoder.java:291)
2784     at sun.nio.cs.StreamEncoder.implFlush(StreamEncoder.java:295)
2785     at sun.nio.cs.StreamEncoder.flush(StreamEncoder.java:141)
2786     - locked <0x00000006cf0ca860> (a java.io.OutputStreamWriter)
2787     at java.io.OutputStreamWriter.flush(OutputStreamWriter.java:229)
2788     at org.apache.log4j.helpers.QuietWriter.flush(QuietWriter.java:66)
2789     at org.apache.log4j.WriterAppender.subAppend(WriterAppender.java:377)
2790     at org.apache.log4j.WriterAppender.append(WriterAppender.java:176)
2791     at org.apache.log4j.AppenderSkeleton.doAppend(AppenderSkeleton.java:286)
2792     - locked <0x00000006cf0ca580> (a org.apache.log4j.ConsoleAppender)
2793     at org.apache.log4j.helpers.AppenderAttachableImpl.appendLoopOnAppenders(AppenderAttachableImpl.java:67)
2794     at org.apache.log4j.Category.callAppenders(Category.java:218)
```

Log4j에서 Log를 Write할 때 느림





Opennaru, Inc. © 2017 | All Rights Reserved.

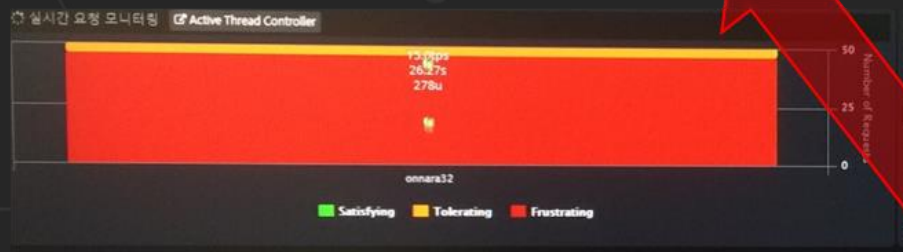
스레드 덤프 분석 주요 패턴

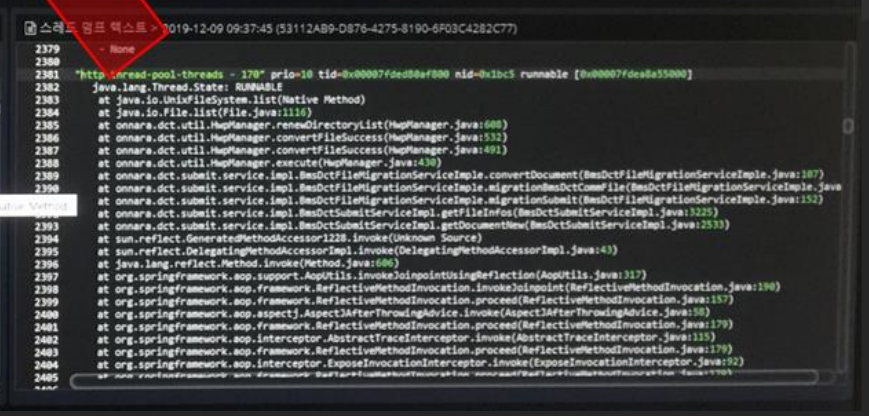
openmaru

파일 I/O가 느릴 때

파일 List를 가져오는 함수에서 응답이 없음

```
2380
2381  "http-thread-pool-threads - 170" prio=10 tid=0x00007fded80ef800 nid=0x1bc5 runnable [0x00007fdea8a55000]
2382     java.lang.Thread.State: RUNNABLE
2383        at java.io.UnixFileSystem.list(Native Method)
2384        at java.io.File.list(File.java:1116)
2385        at [redacted].util.HwpManager.renewDirectoryList(HwpManager.java:608)
2386        at [redacted].util.HwpManager.convertFileSuccess(HwpManager.java:532)
```





Opennaru, Inc. © 2017 | All Rights Reserved.

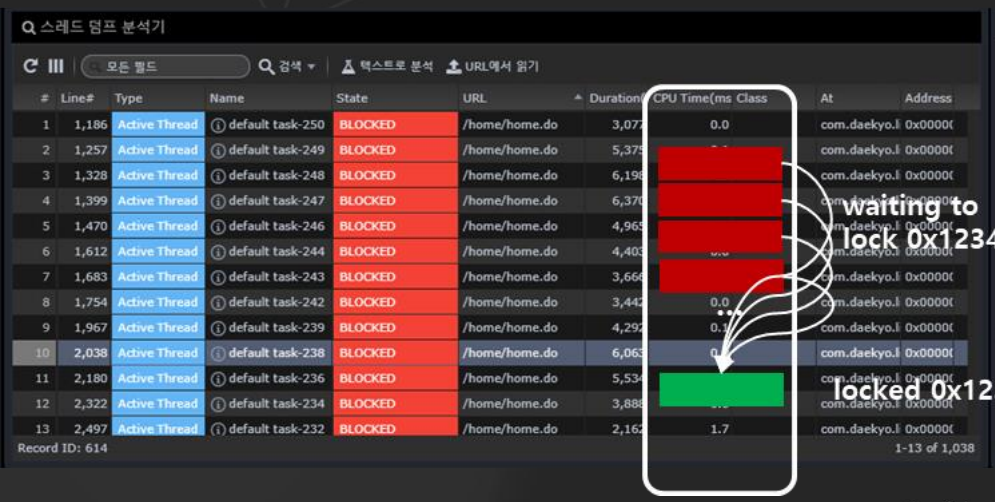
스레드 덤프 분석 주요 패턴

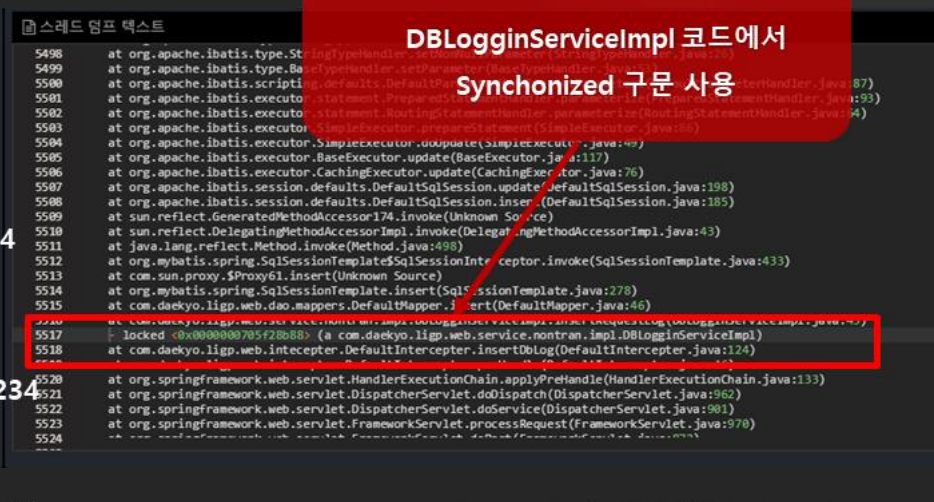
openmaru

Lock이 걸릴 때

```
5506        at org.apache.ibatis.executor.CachingExecutor.update(CachingExecutor.java:76)
5507        at org.apache.ibatis.session.defaults.DefaultSqlSession.update(DefaultSqlSession.java:198)
5508        at org.apache.ibatis.session.defaults.DefaultSqlSession.insert(DefaultSqlSession.java:185)
5509        at sun.reflect.GeneratedMethodAccessor174.invoke(Unknown Source)
5510        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
5511        at java.lang.reflect.Method.invoke(Method.java:498)
5512        at org.mybatis.spring.SqlSessionTemplate$SqlSessionInterceptor.invoke(SqlSessionTemplate.java:433)
5513        at com.sun.proxy.$Proxy61.insert(Unknown Source)
5514        at org.mybatis.spring.SqlSessionTemplate.insert(SqlSessionTemplate.java:278)
5515        at com.[redacted].web.dao.mappers.DefaultMapper.insert(DefaultMapper.java:46)
5516        at com.[redacted].web.service.nontran.impl.DBLogginServiceImpl.insertRequestLog(DBLogginServiceImpl.java:45)
5517      - locked <0x0000000705f28b88> (a com.[redacted].web.service.nontran.impl.DBLogginServiceImpl)
5518        at com.[redacted].web.intecepter.DefaultIntercepter.insertDbLog(DefaultIntercepter.java:124)
5519        at com.[redacted].web.intecepter.DefaultIntercepter.preHandle(DefaultIntercepter.java:46)
5520        at org.springframework.web.servlet.HandlerExecutionChain.applyPreHandle(HandlerExecutionChain.java:133)
```

모두 Lock 0x0000..705f28b88 을 기다리고 있음





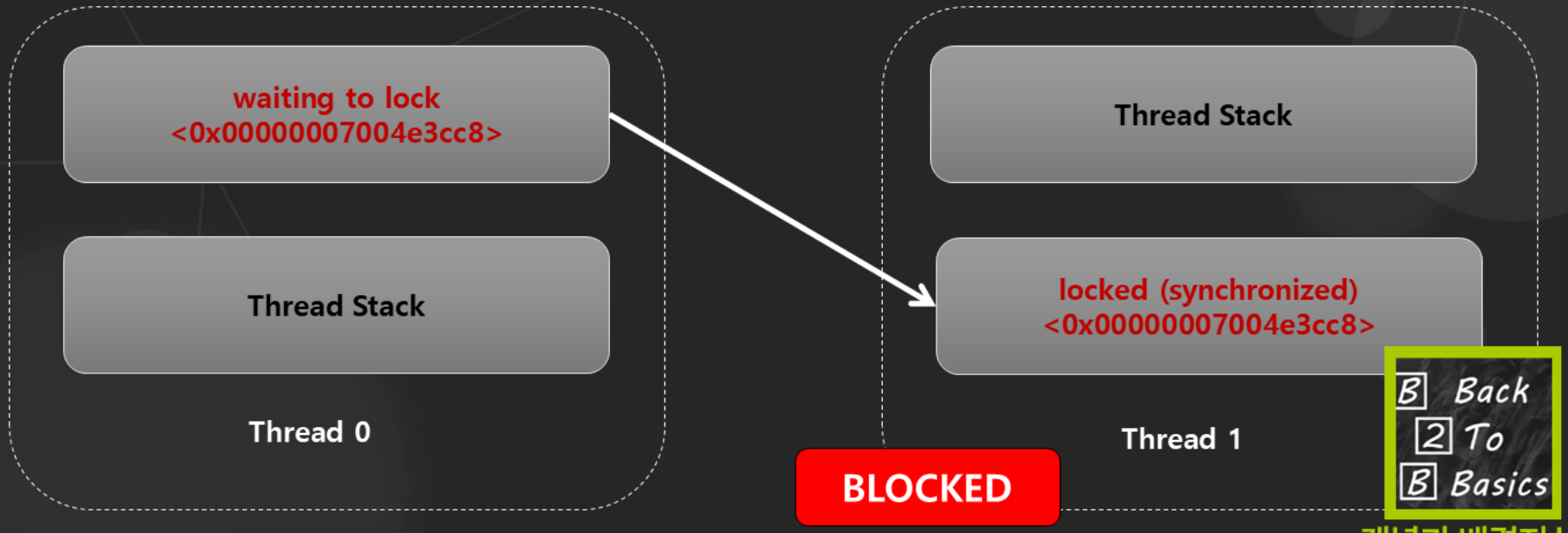
DBLogginServiceImpl 코드에서 Synchonized 구문 사용

Opennaru, Inc. © 2017 | All Rights Reserved.

Thread Dump의 Blocked의 의미

- 여러 스레드에서 같은 변수나 메소드를 변경할 경우
- synchronized로 다른 스레드의 처리가 끝날 때 까지 제어가 넘어가지 않도록 보호

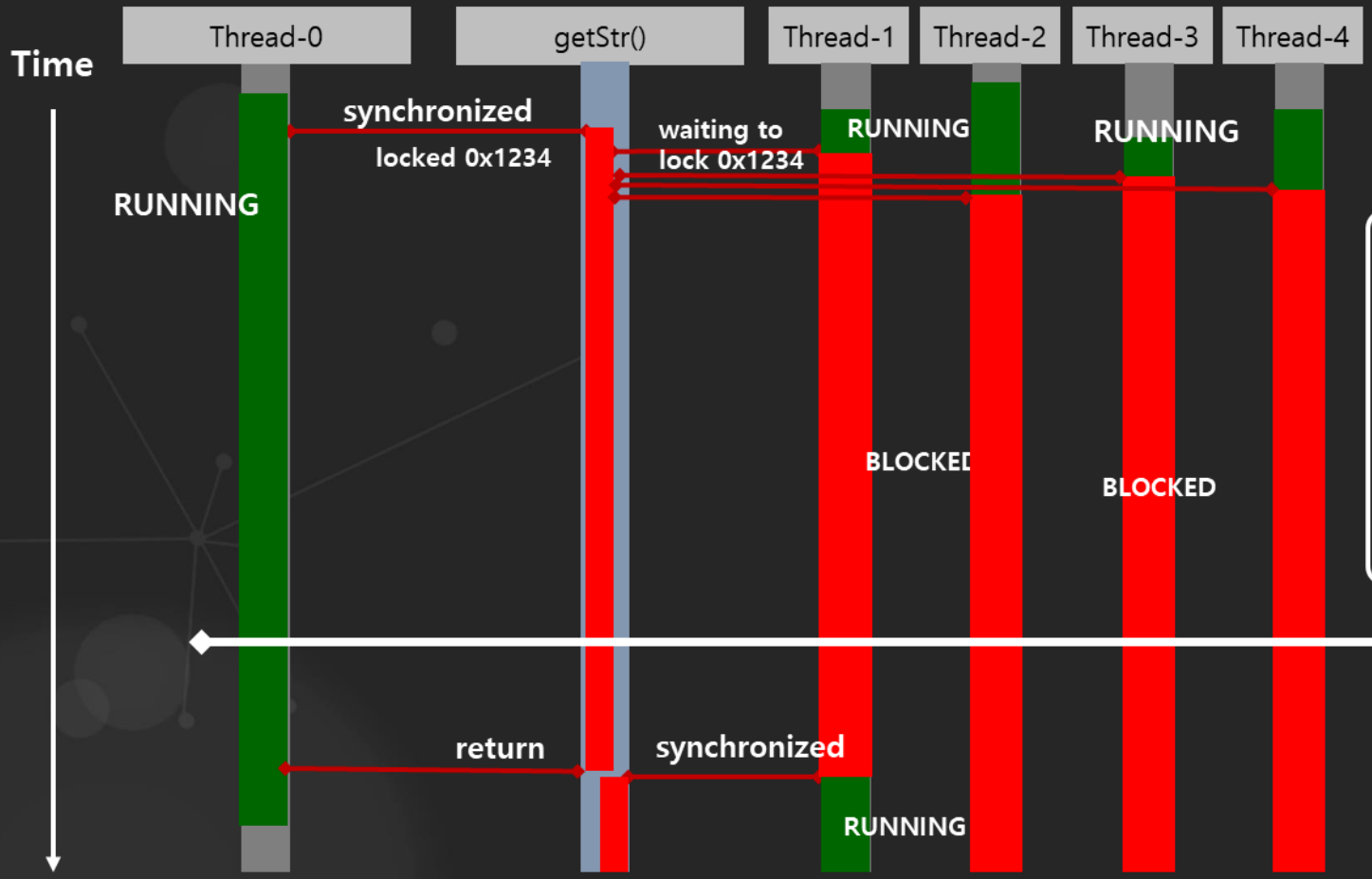
```
// 특정한 객체에 lock을 걸고자 할 때
synchronized(객체의 참조변수){
    // ...
}

// 메서드에 lock을 걸고자할 때
public synchronized void methodName(){
    // ...
}
```

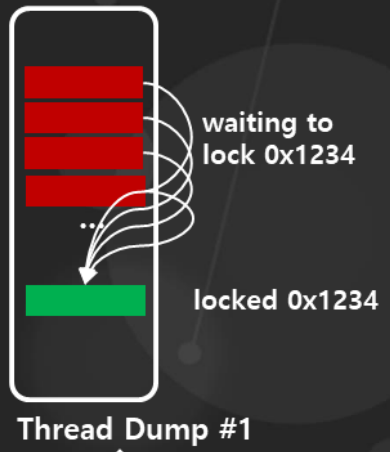


Back 2 To Basics
개념과 배경지식

Opennaru, Inc. © 2017 | All Rights Reserved.

Thread BLOCKED 상태가 길어지면 발생하는 상황

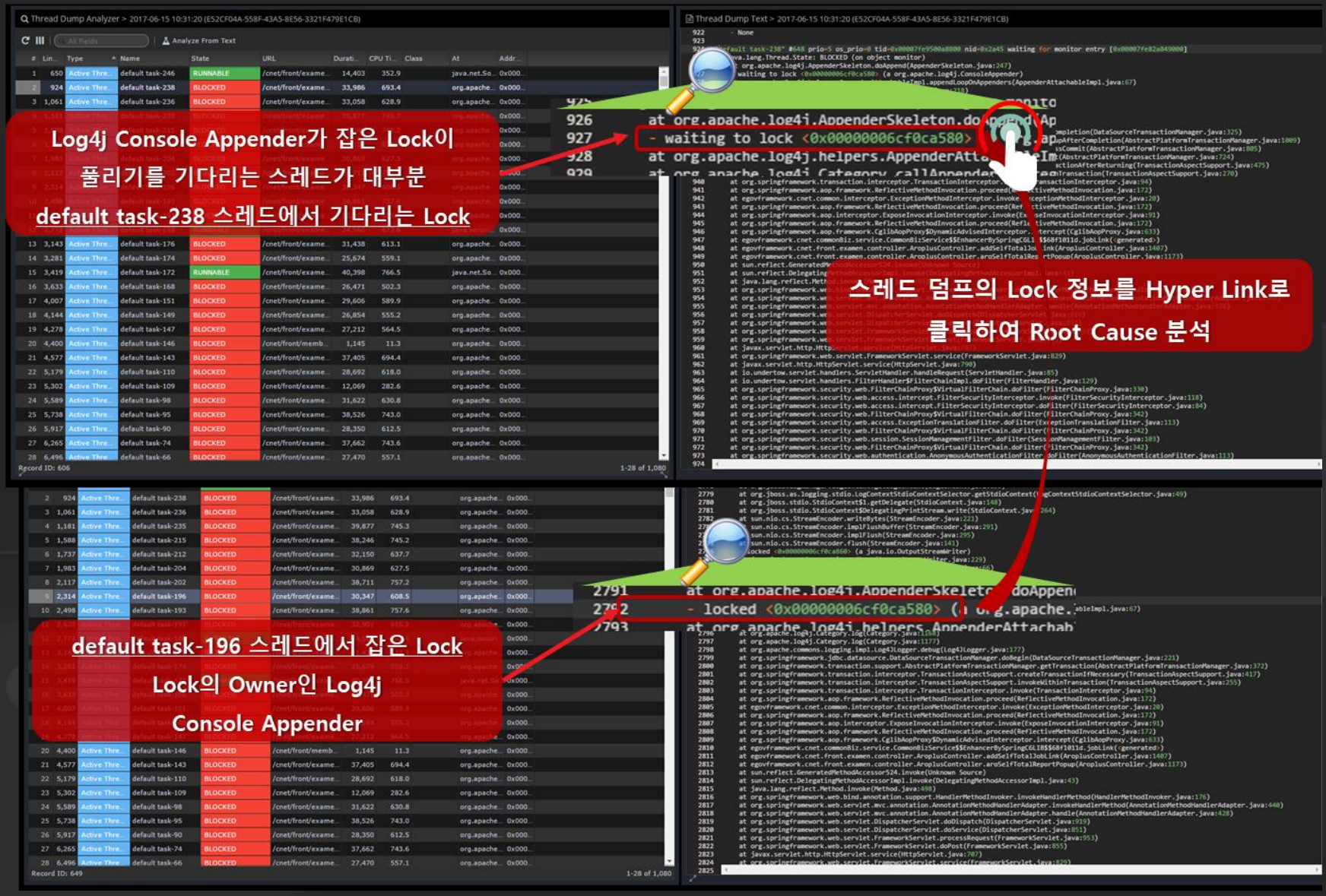




Back 2 To Basics
개념과 배경지식

Opennaru, Inc. © 2017 | All Rights Reserved.

OPENMARU APM 스레드 덤프 분석 예시



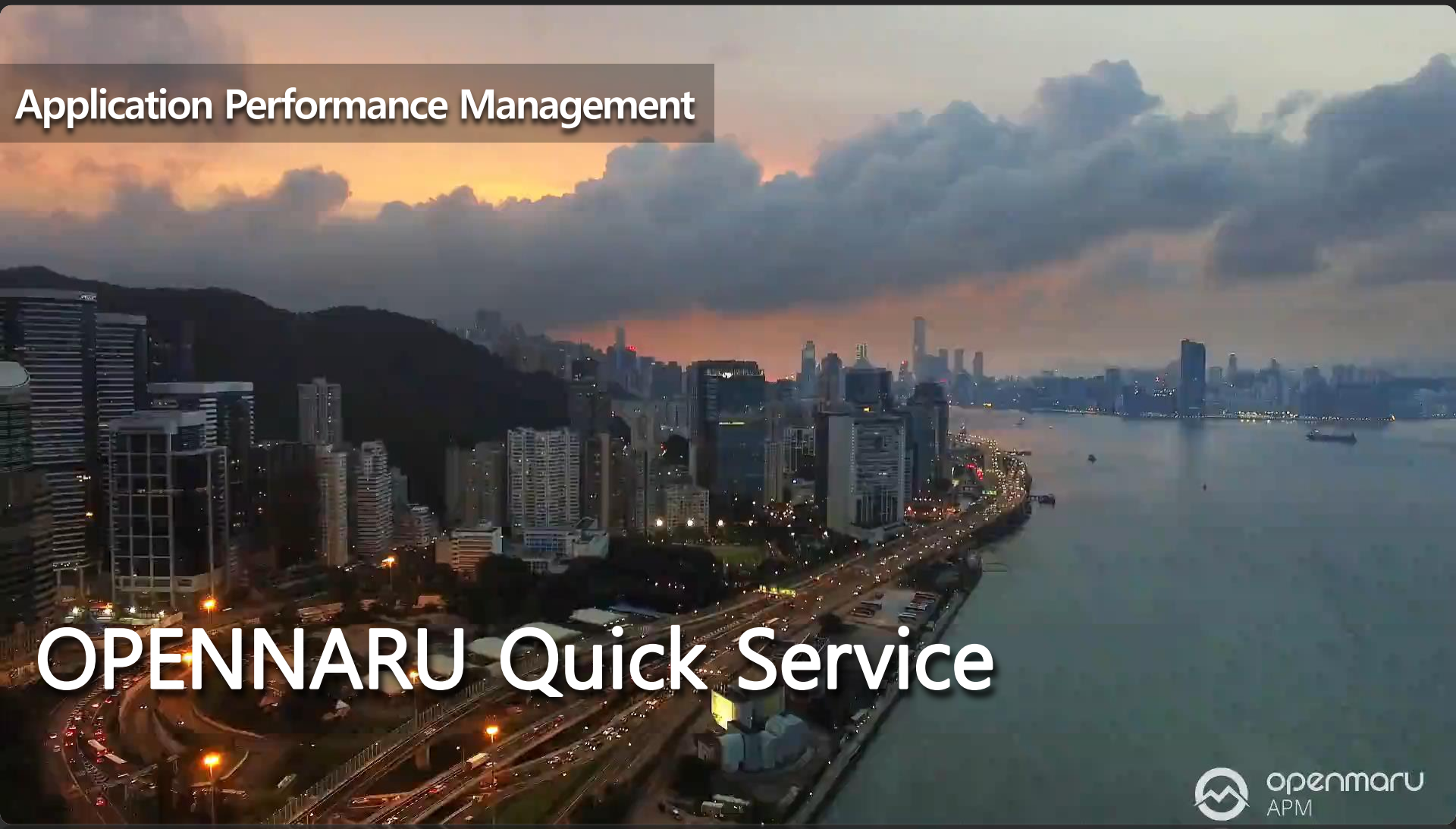
Opennaru, Inc. © 2017 | All Rights Reserved.

스레드 덤프 분석 주요 패턴의 주요 해결방법

openmaru

DB 쿼리가 느릴 때	쿼리 튜닝 DB 연결풀 개수 튜닝
HTTP 호출이 느릴 때	HttpClient로 변경(URLConnection사용하지 말것) Pooling하여 처리 / 개수 튜닝
Logging 처리가 느릴 때	디버그 로그 운영 사용금지 Console/파일로 함께 남기지 말 것
파일 I/O가 느릴 때	파일 I/O 튜닝, 스토리지 확인
Lock이 걸릴 때	애플리케이션 개발시 Lock 최소화

Opennaru, Inc. © 2017 | All Rights Reserved.



- Confidential -
Opennaru, Inc. © 2016 | All Rights Reserved.

OPENNARU Quick Service 란?

고객사

① 시스템 이슈발생

② 쓰레드 덤프와 모니터링 화면 전달

③ 오픈나루 서비스 데스크 접수

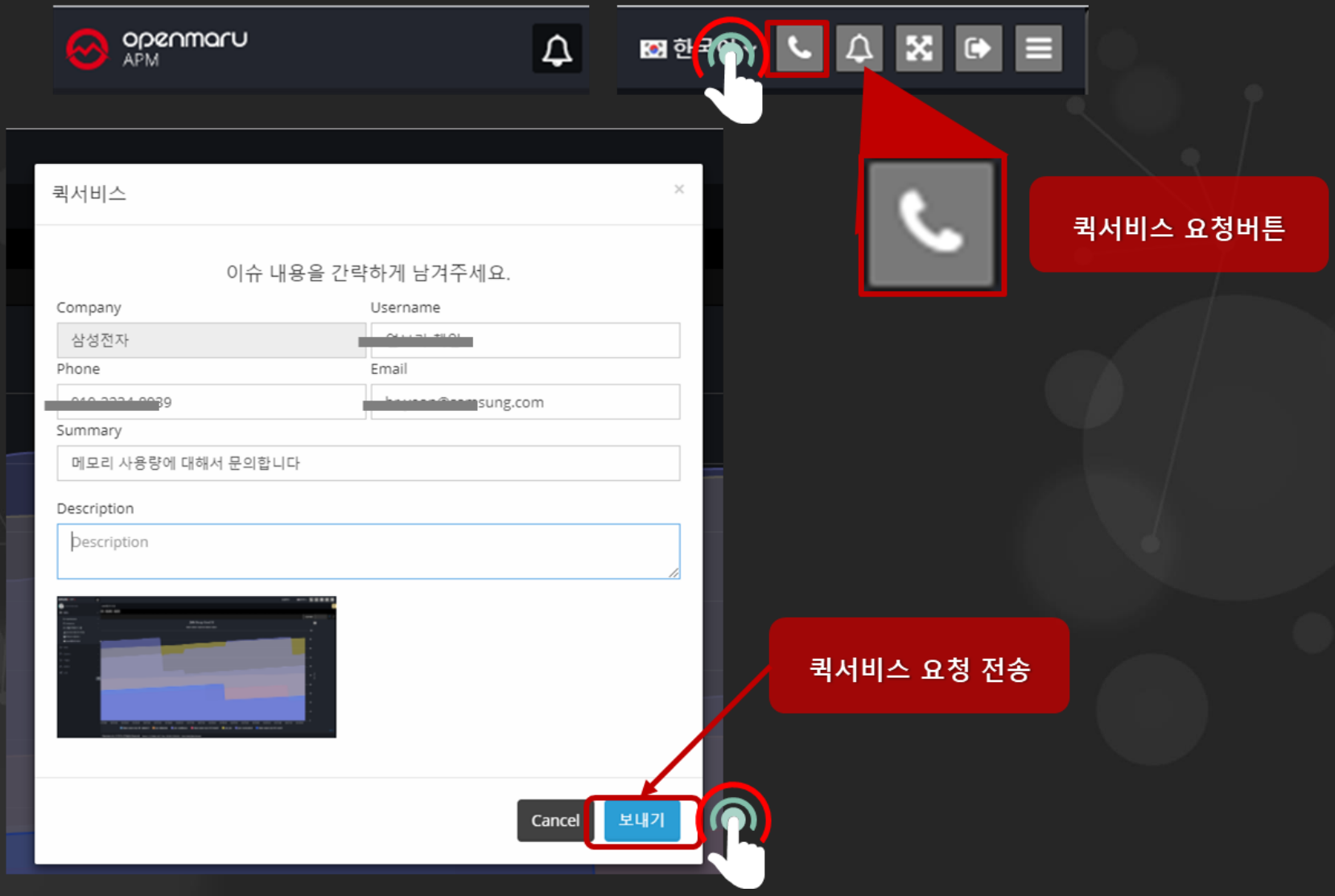
④ 장애 분석

⑤ 장애 분석 보고서 전달

오픈나루

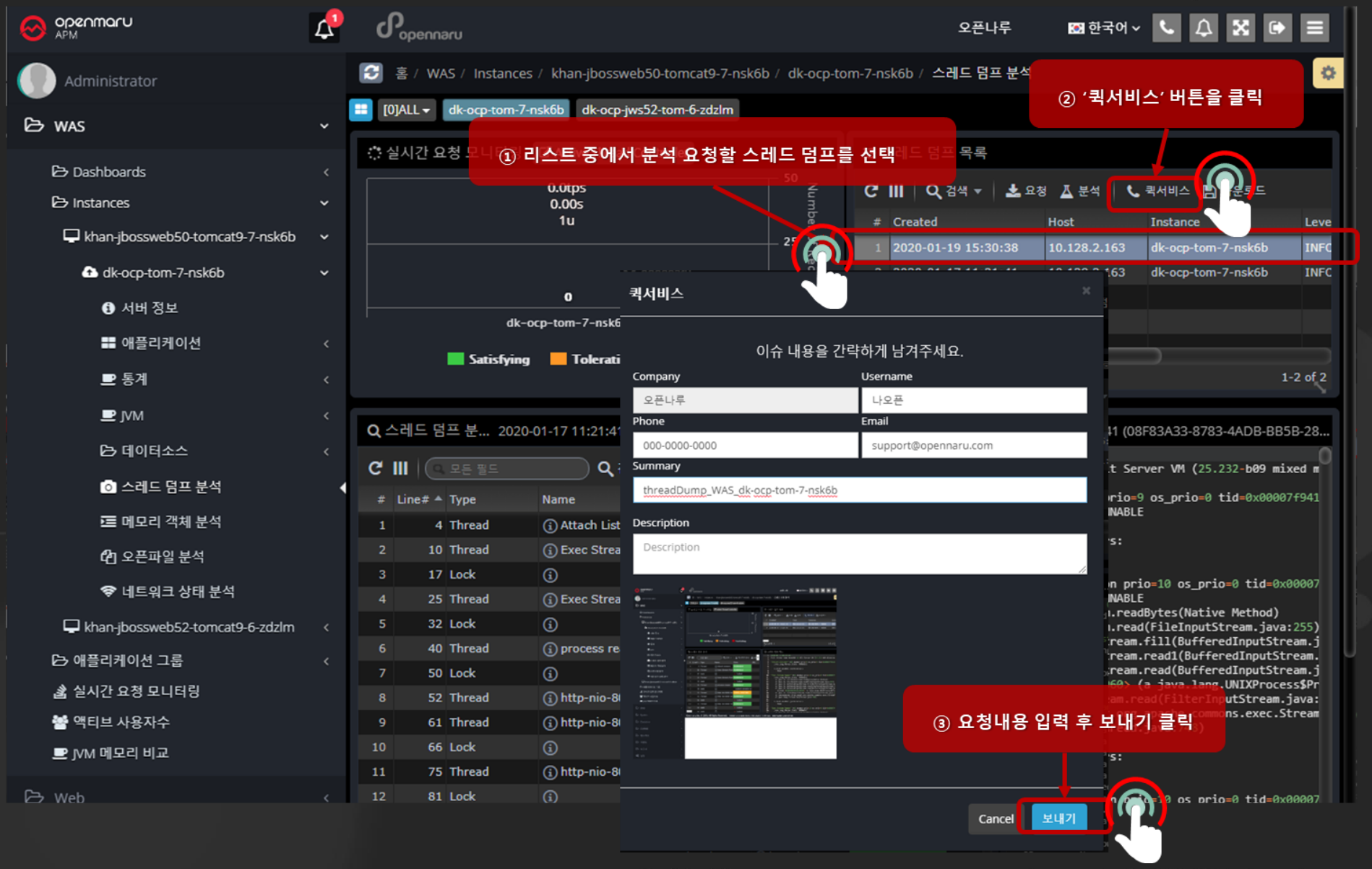
Opennaru, Inc. © 2017 | All Rights Reserved.

Quick Service 원격 진단 요청 – ① 모든 화면에서 가능



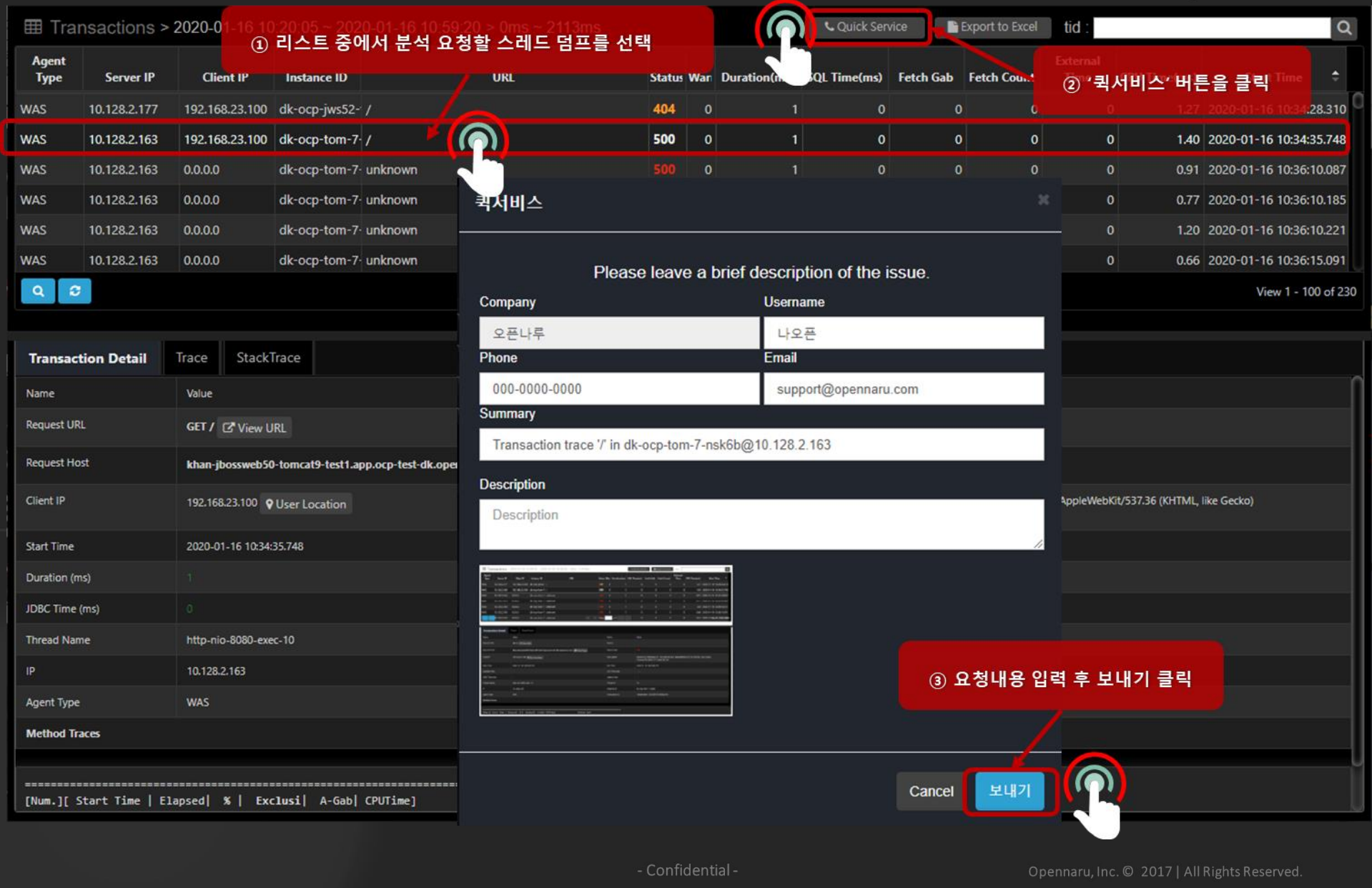
Opennaru, Inc. © 2017 | All Rights Reserved.

Quick Service 원격 진단 요청 - ② 스레드 덤프



Opennaru, Inc. © 2017 | All Rights Reserved.

Quick Service 원격 진단 요청 – ③ Transaction Detail



Opennaru, Inc. © 2017 | All Rights Reserved.

Opennaru, Inc. © 2016 | All Rights Reserved.

OutOfMemory의 유형

증상과 원인

- **메모리 부족**
  - **Heap의 메모리 OutOfMemory 에러**
    - 특정 시각에 급격한 메모리 증가
    - 급격한 증가/점차적인 메모리 증가
  - **Native 메모리의 OutOfMemory 에러**
    - RES 등 OS에서 사용하는 영역에서 발생하는 메모리 부족현상
    - 급격한 증가/점차적인 메모리 증가
- **Memory Leak**
  - 메모리 사용량이 계속 증가해 결과적으로 메모리 부족 상태
  - HTTP Session 오브젝트의 증가, JDBC 관련 오브젝트의 close 새어 캐쉬 오브젝트가 원인 등이 있음
- **Java VM 버그**
  - 매우 드문 경우이지만 JVM 버그의 가능성이 있기 때문에 JVM 버그 정보 조회

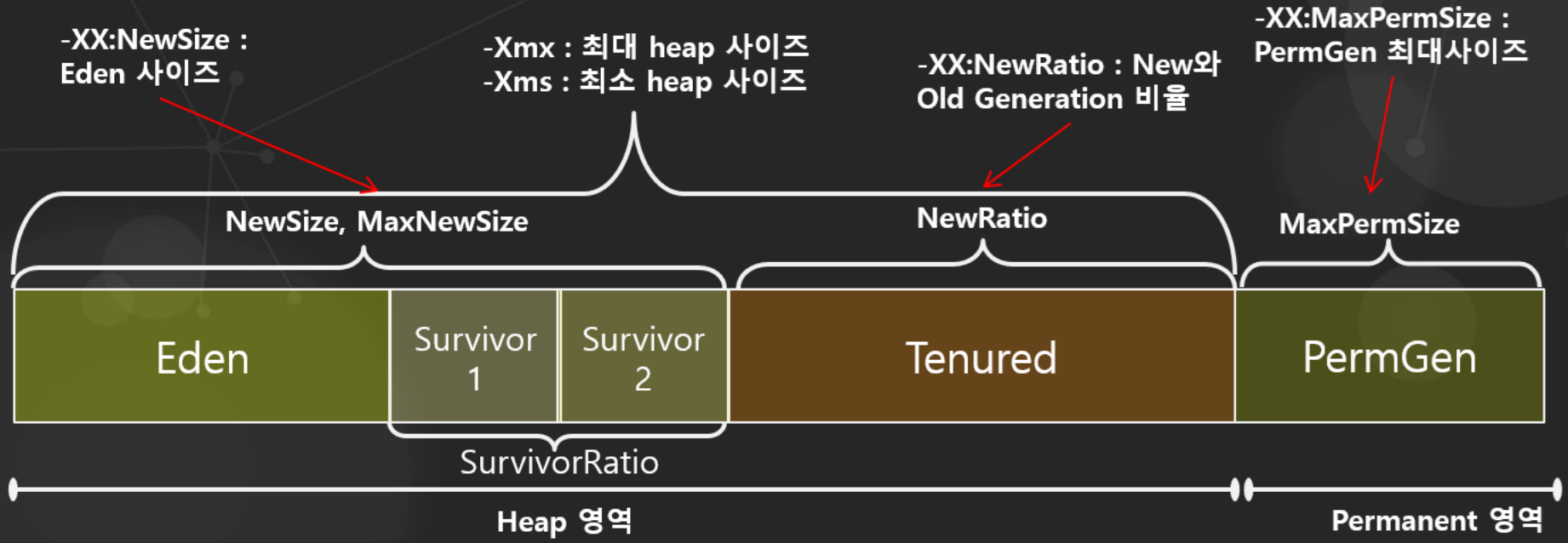
Opennaru, Inc. © 2017 | All Rights Reserved.

Java Heap Memory

openmaru

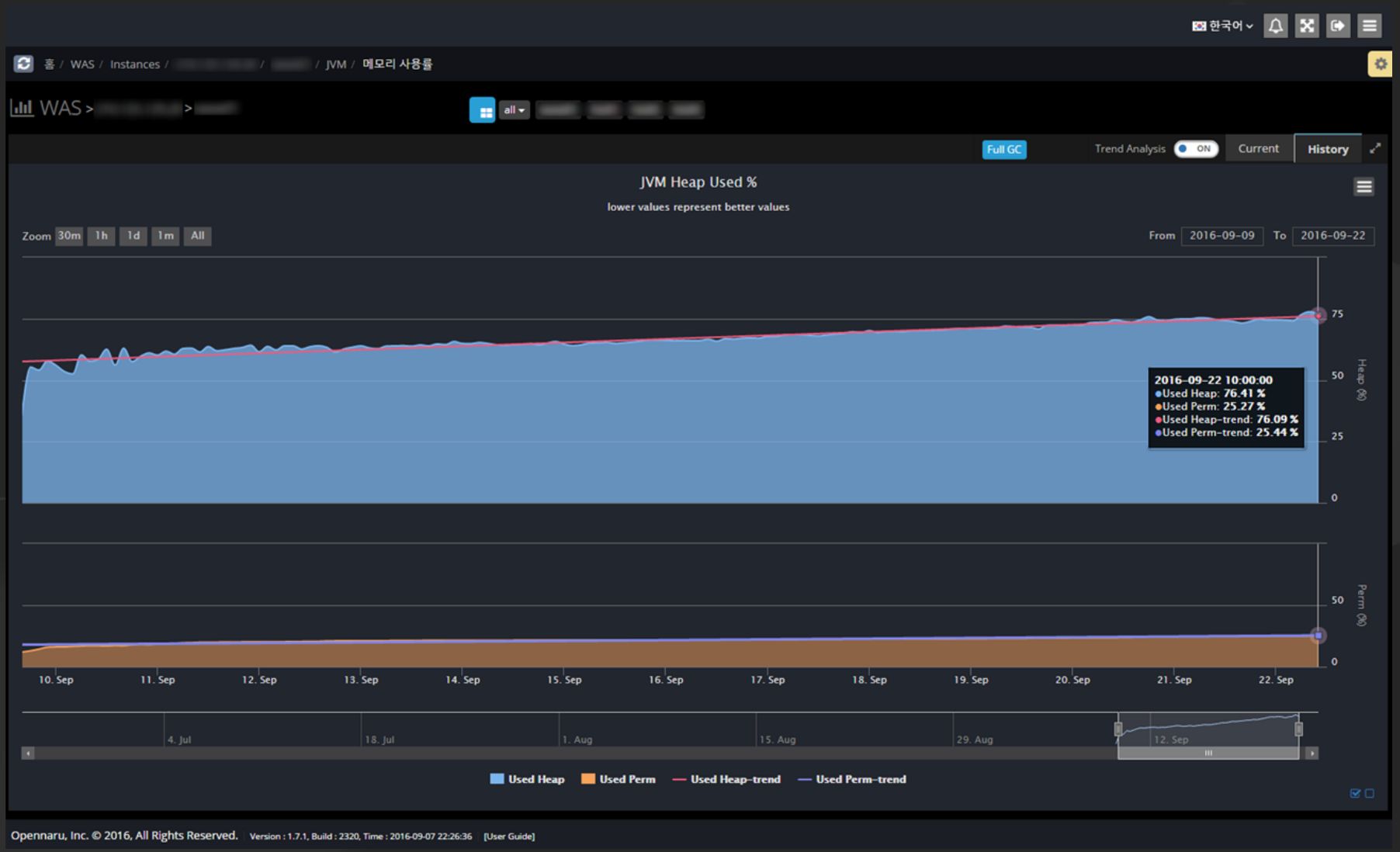
Back 2 To Basics
개념과 배경지식

- **Heap 영역**
  - **Eden 영역**
    - 새롭게 작성된 오브젝트를 저장
    - 생존 기간의 짧은 오브젝트는 Eden 영역에서 GC의 대상
  - **Tenured 영역**
    - Eden 영역에서 GC 대상이 되지 않은 old 오브젝트를 저장
- **Permanent 영역**
  - **PermGen**
    - 클래스 오브젝트와 그 외의 시스템 레벨의 오브젝트를 저장



Opennaru, Inc. © 2017 | All Rights Reserved.

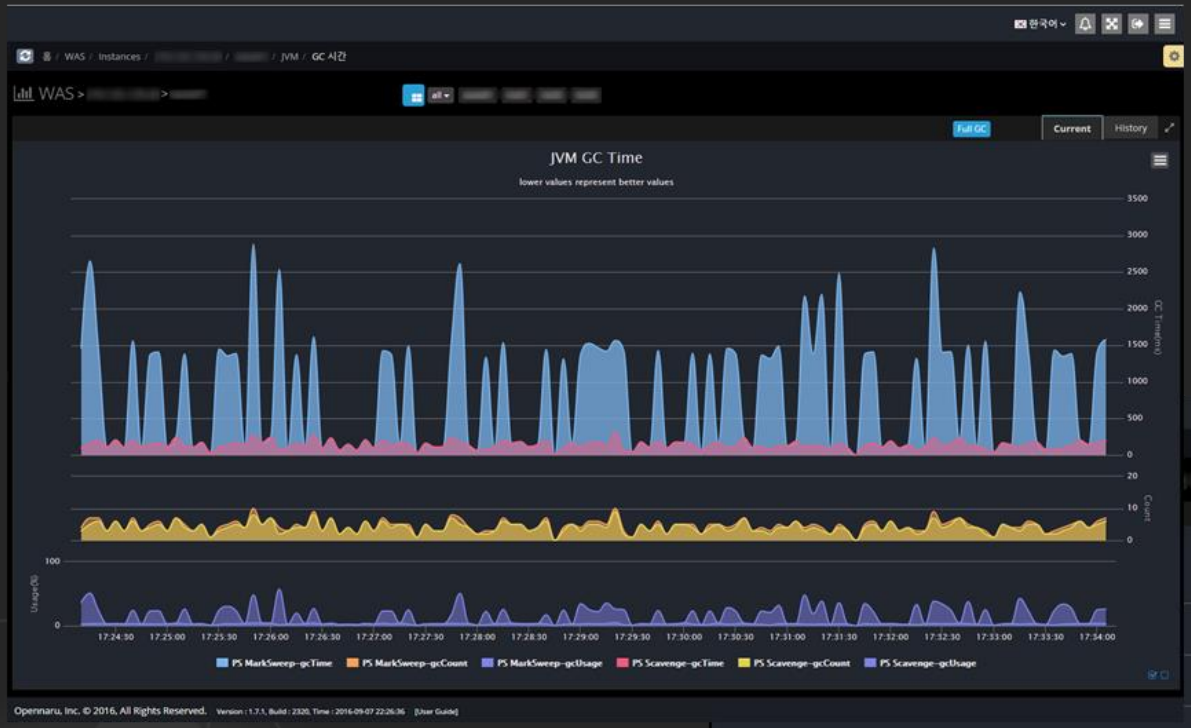
JVM 메모리 상황을 Byte 단위로 분석



Opennaru, Inc. © 2017 | All Rights Reserved.

JVM GC 시간/횟수 상황을 분석

- Java VM Heap GC 현황을 실시간으로 모니터링



- Java VM GC 히스토리

Opennaru, Inc. © 2017 | All Rights Reserved.

메모리가 급격히 부족한 경우

- 이벤트 → 이벤트 목록

URL에서 10만건 데이터를 가져오는 중
20만건의 데이터를 가져오는 중을 알려줌

1	2017-12-15 15:05:04	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/S...	210.90.107.181	WAS	homepage11
2	2017-12-15 15:04:56	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/Se...	210.90.107.181	WAS	homepage11
3	2017-12-15 15:04:52	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/SelectBoardView.do에서 200000개 이상을 fetch하는 중(현재 소요시간:14258m...	210.90.107.181	WAS	homepage14
4	2017-12-15 15:04:46	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/SelectBoardView.do에서 100000개 이상을 fetch하는 중(현재 소요시간:7358ms).	210.90.107.181	WAS	homepage14
5	2017-12-15 14:44:15	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/SelectBoardView.do에서 200000개 이상을 fetch하는 중(현재 소요시간:11544m...	210.90.107.181	WAS	homepage11
6	2017-12-15 14:44:09	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/SelectBoardView.do에서 100000개 이상을 fetch하는 중(현재 소요시간:5897ms).	210.90.107.181	WAS	homepage11
7	2017-12-15 01:17:08	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/SelectBoardView.do에서 200000개 이상을 fetch하는 중(현재 소요시간:13697m...	210.90.107.181	WAS	homepage14
8	2017-12-15 01:17:02	경고-WARN	ALERT_TOO_MANY_FETCH	DB ResultSet Fetch	너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /member/mobile/select/SelectBoardView.do에서 100000개 이상을 fetch하는 중(현재 소요시간:7454ms).	210.90.107.181	WAS	homepage14

너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /me... View.do에서 200000개 이상을 fetch하는 중(현재 소요시간:16211m...

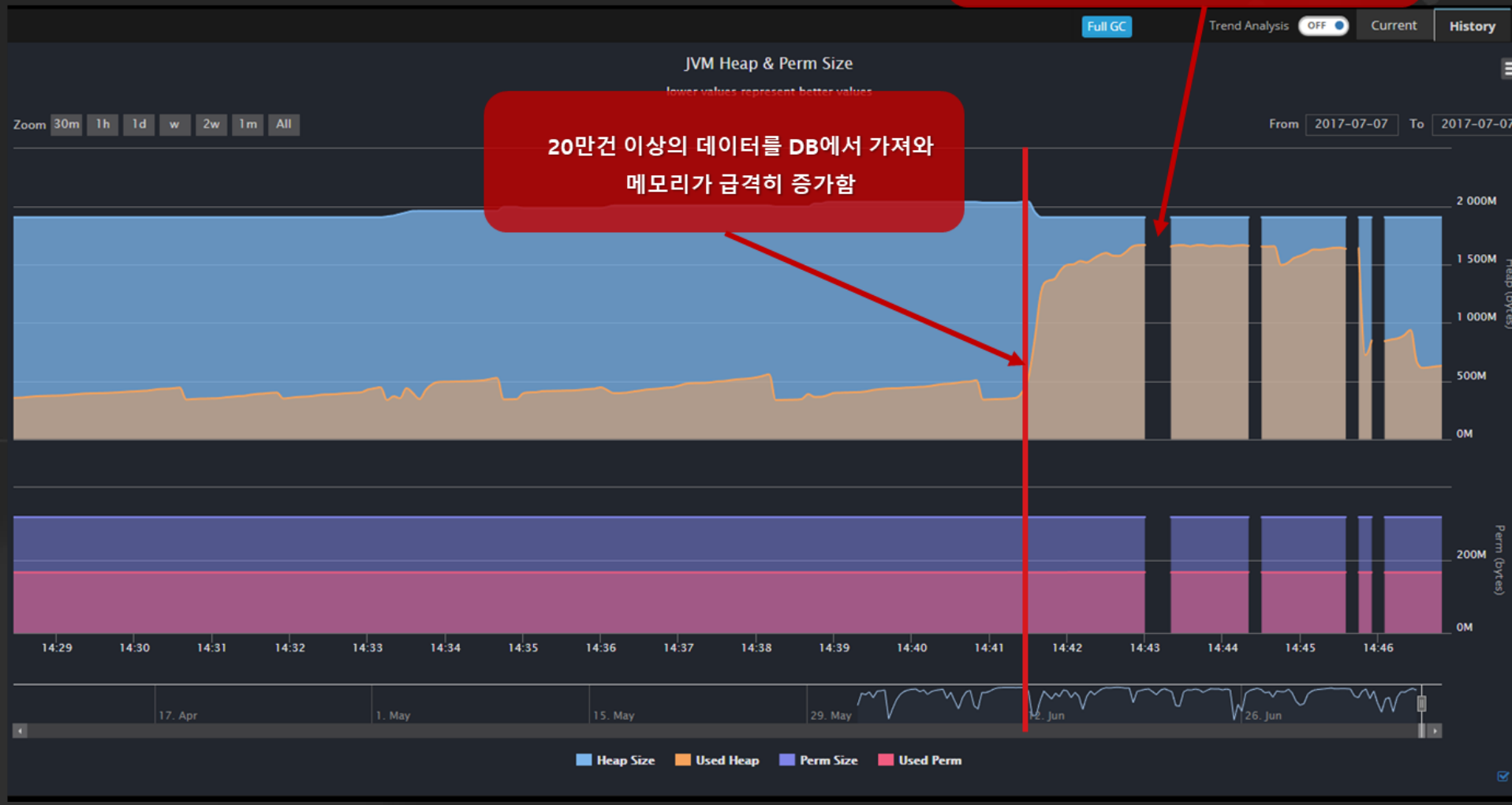
너무 많은 데이터베이스 ResultSet fetch가 감지되었습니다. /me... View.do에서 100000개 이상을 fetch하는 중(현재 소요시간:8169ms).

대용량 Fetch이후에
1) Heap 사용량 경고 발생
2) GC 사용시간 경고 발생
3) WAS 재접속 현상도 발생

경고-WARN	EVENT_WARN_THRESHOLD_EXCEED	GC Usage %	'GC Usage %' (평균값: 20.62)이 경고(WARN) 임계값 '20'을 넘었습니다.	V...
심각-CRIT...	EVENT_CRITICAL_THRESHOLD_EXC...	GC Usage %	'GC Usage %' (평균값: 38.91)이 심각(CRITICAL) 임계값 '35'을 넘었습...	V...
경고-WARN	EVENT_WARN_THRESHOLD_EXCEED	GC Usage %	'GC Usage %' (평균값: 33.77)이 경고(WARN) 임계값 '20'을 넘었습니다.	V...
정보-INFO	EVENT_AGENT_CONNECTED		WAS가 시작되어 에이전트가 연결되었습니다.	V...
경고-WARN	EVENT_WARN_THRESHOLD_EXCEED	APDEX	'APDEX' (평균값: 67.19)이 경고(WARN) 임계값 '70'을 넘었습니다.	V...
경고-WARN	EVENT_WARN_THRESHOLD_EXCEED	Pending Tr...	'Pending Transactions' (평균값: 33)이 경고(WARN) 임계값 '30'을 넘었...	V...
경고-WARN	EVENT_WARN_THRESHOLD_EXCEED	Heap Usag...	'Heap Usage %' (평균값: 86.01)이 경고(WARN) 임계값 '85'을 넘었습니...	V...
심각-CRIT...	EVENT_CRITICAL_THRESHOLD_EXC...	GC Usage %	'GC Usage %' (평균값: 96.7)이 심각(CRITICAL) 임계값 '35'을 넘었습니...	V...
경고-WARN	EVENT_WARN_THRESHOLD_EXCEED	Heap Usag...	'Heap Usage %' (평균값: 85.09)이 경고(WARN) 임계값 '85'을 넘었습니...	V...
심각-CRIT...	EVENT_CRITICAL_THRESHOLD_EXC...	APDEX	'APDEX' (평균값: 46.88)이 심각(CRITICAL) 임계값 '50'을 넘었습니다.	V...
경고-WARN	EVENT_WARN_THRESHOLD_EXCEED	GC Usage %	'GC Usage %' (평균값: 20.61)이 경고(WARN) 임계값 '20'을 넘었습니다.	V...
심각-CRIT...	EVENT_CRITICAL_THRESHOLD_EXC...	GC Usage %	'GC Usage %' (평균값: 37.81)이 심각(CRITICAL) 임계값 '35'을 넘었습...	V...

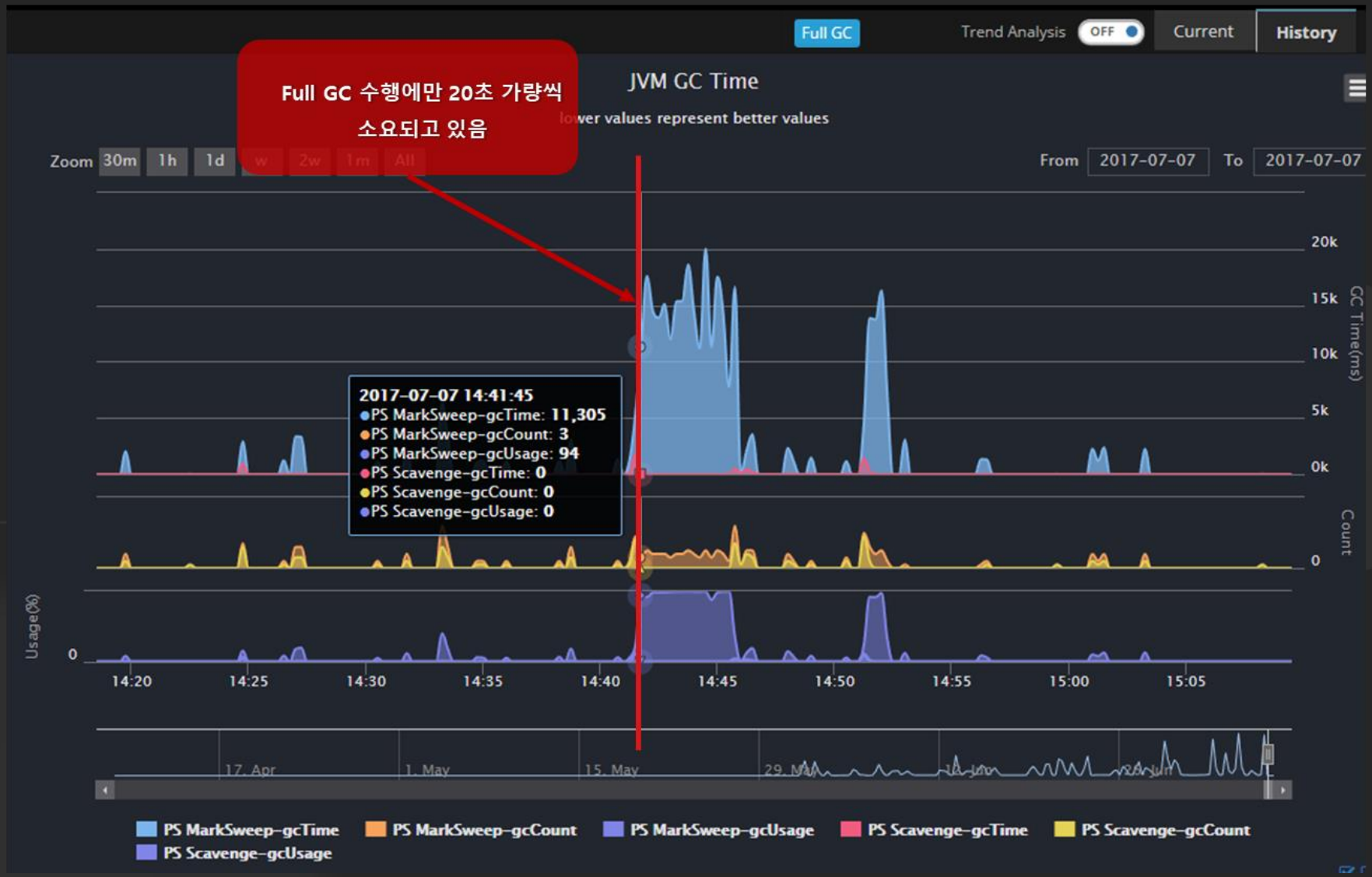
Opennaru, Inc. © 2017 | All Rights Reserved.

대용량 Fetch로 인해 발생하는 현상



Opennaru, Inc. © 2017 | All Rights Reserved.

대용량 Fetch 이후의 메모리 GC 현황



Opennaru, Inc. © 2017 | All Rights Reserved.

OutOfMemory 오류 - 많은 DB ResultSet fetch 감지

Agent Type	Server IP	Client IP	Instance ID	URL	Status	Duration(ms)	SQL Time(ms)	Fetch Gab	Fetch Count	External Time	CPU Time(ms)	Start Time
WAS	192.168.1.25	110.70.56.200	myacv-1-25	/member/auth	500	178,684	93	178,543	92,982	0	3,203.13	2017-12-11 18:30:47.75
WAS	192.168.1.25	223.62.8.206	myacv-1-25	/member/auth	200	146,325	9,265	0	2	0	31.25	2017-12-11 18:31:19.92
WAS	192.168.1.25	223.62.8.206	myacv-1-25	/member/auth	200	143,590	0	0	2	0	31.25	2017-12-11 18:31:22.66
WAS	192.168.1.25	223.62.8.206	myacv-1-25	/log/tracking	200	134,481	0	0	0	0	15.63	2017-12-11 18:31:31.78
WAS	192.168.1.25	220.94.248.31	myacv-1-25	/log/tracking	200	129,779	0	0	0	0	46.88	2017-12-11 18:31:36.50
WAS	192.168.1.25	110.70.56.200	myacv-1-25	/member/auth	200	127,184	4,093	0	1	0	46.88	2017-12-11 18:31:39.06
WAS	192.168.1.25	223.62.8.206	myacv-1-25	/log/tracking	200	124,559	37,827	0	0	0	31.25	2017-12-11 18:31:41.69
WAS	192.168.1.25	219.248.121.3	myacv-1-25	/member/firstAct	200	119,107	0	0	1	0	78.13	2017-12-11 18:31:47.12

View 1 - 22 of

인스턴스에서 OutOfMemory 오류가 발생

로직에 따라 SQL 쿼리의 WHERE 조건절에 필요한 조건이 빠져 테이블 Full Scan이 발생함

Transaction Detail

```
WHERE 1=1

AND A.Status = 'D'
AND ISNULL(A.CsStatus,'')!= 'delete'
```

해결책 : 사용자가 선택한 값이 없을 때에도 WHERE 조건절이 생성되도록 애플리케이션을 변경해야 함

Opennaru, Inc. © 2017 | All Rights Reserved.

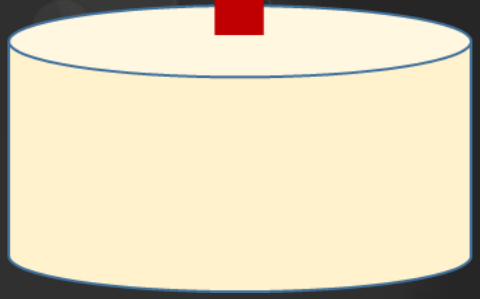
Excel 파일을 출력하는 경우

WAS Server

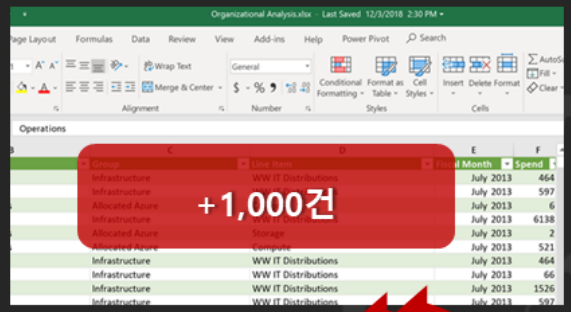
JBOSS EAP 7
test11

JBOSS EAP 7
test12

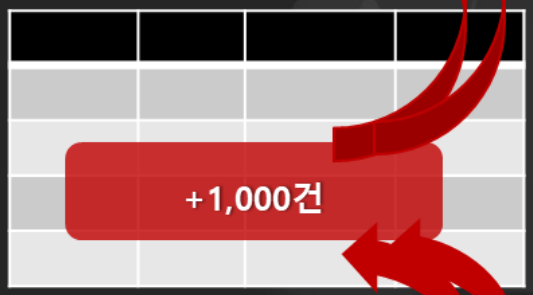
Red Hat Enterprise Linux



OutOfMemoryError



+1,000건



+1,000건

Opennaru, Inc. © 2017 | All Rights Reserved.

Java에서 Excel 파일 생성 방법

openmaru

- Apache POI 라이브러리

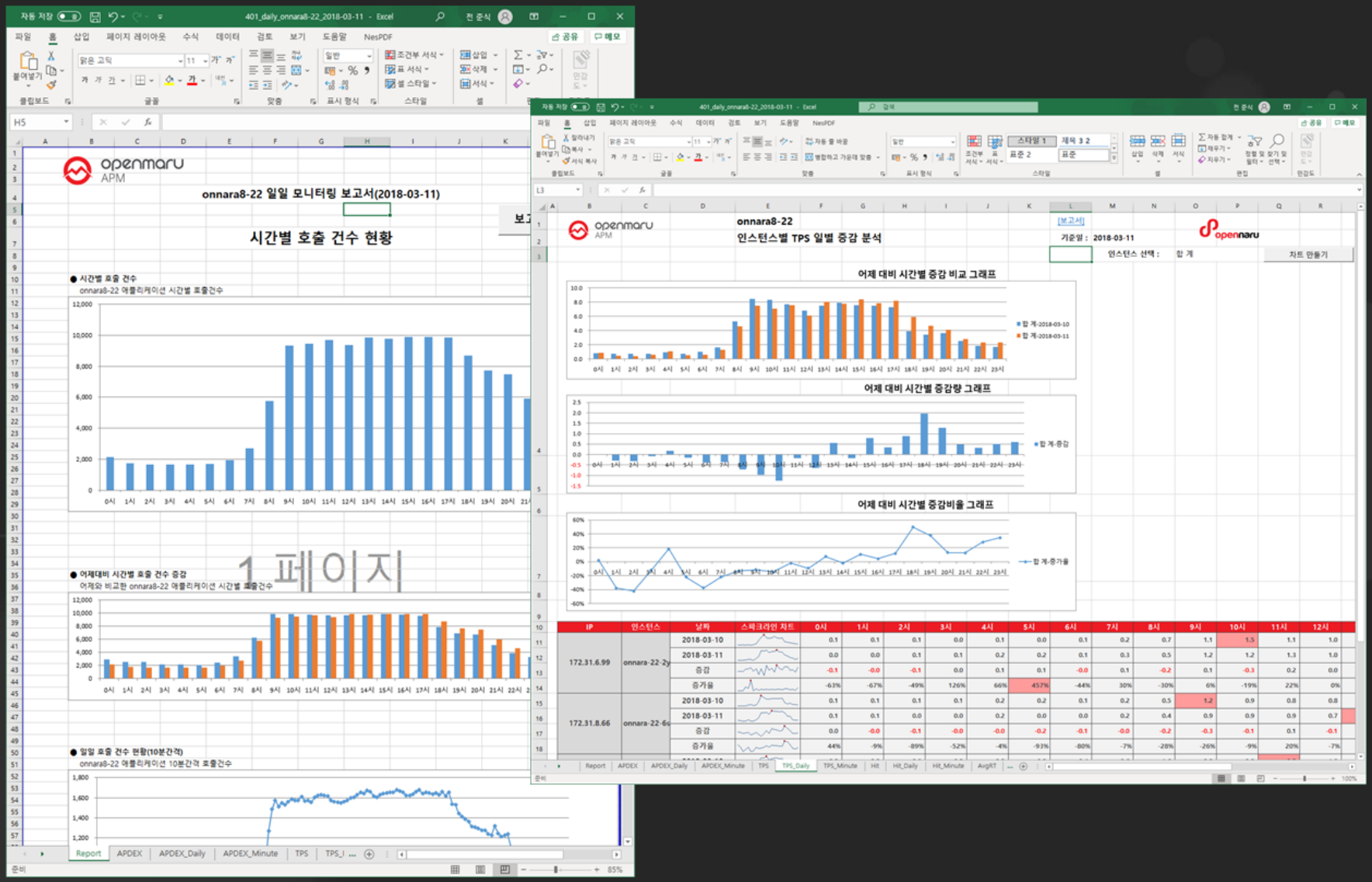
	HSSF (.xls)		XSSF (.xlsx)		SXSSF
	eventmodel	usermodel	eventmodel	usermodel	
API Type	Streaming (ala SAX)	In memory tree (ala DOM)	Streaming (ala SAX)	In memory tree (ala DOM)	Buffered Streaming
CPU and Memory Efficiency	Good	Varies	Good	Varies	Good
Forward Only	Yes	No	Yes	No	Yes
Read Files	Yes	Yes	Yes	Yes	No
Write Files	No	Yes	No	Yes	Yes
Feature:					
create sheets / rows / cells	No	Yes	No	Yes	Yes
styling cells	No	Yes	No	Yes	Yes
delete sheets / rows/cells	No	Yes	No	Yes	No
shift rows	No	Yes	No	Yes	No
cloning sheets	No	Yes	No	Yes	No
formula evaluation	No	Yes	No	Yes	No
cell comments	No	Yes	No	Yes	No
pictures	No	Yes	No	Yes	Yes

- HSSF : 2007이전 버전
- XSSF : 2007이후 버전(라인 수 제한없음)
- SXSSF : 2007이후 속도 빠름, flushRows필요

- JXLS
  - OPENMARU APM에서 엑셀 생성에 사용하는 모듈

Opennaru, Inc. © 2017 | All Rights Reserved.

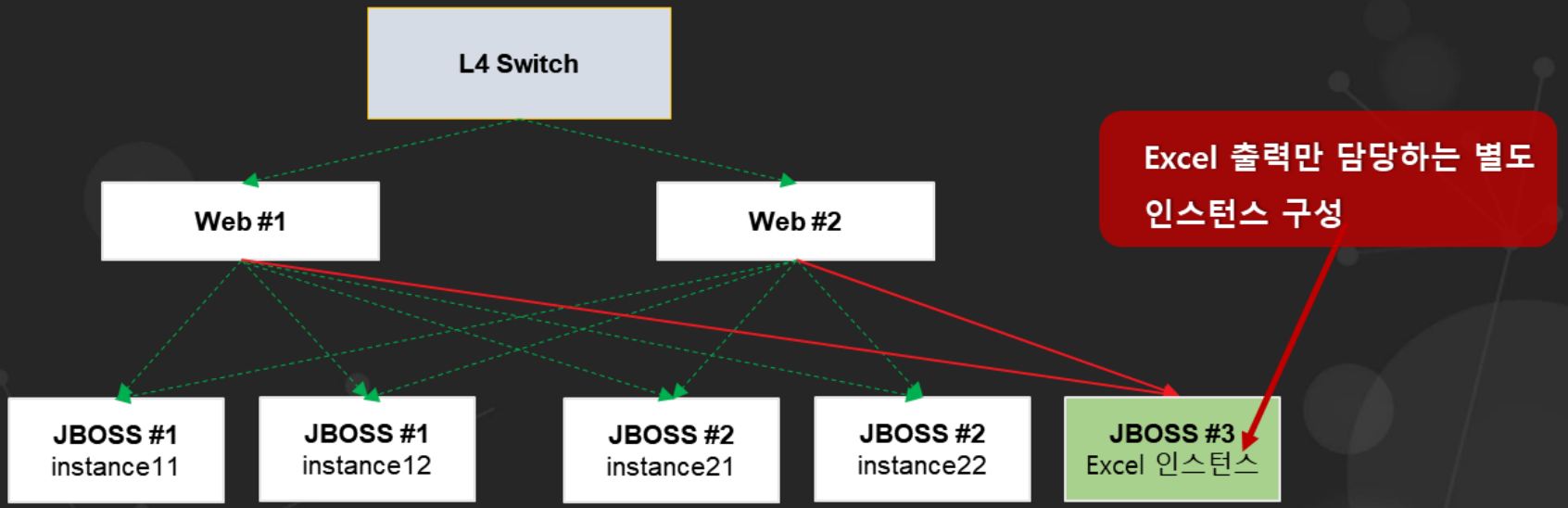
OPENMARU APM의 일일/주간 보고서



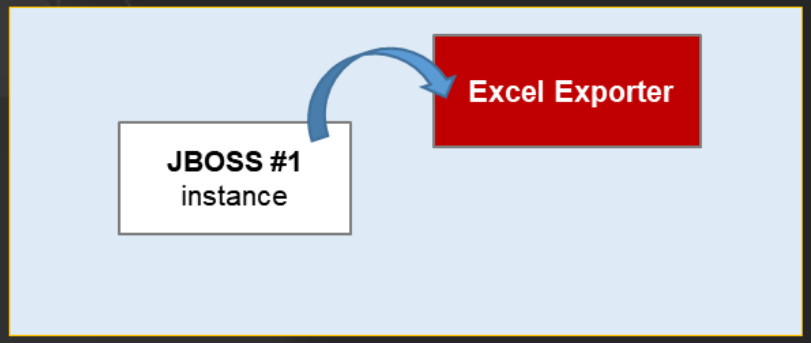
Opennaru, Inc. © 2017 | All Rights Reserved.

SXSSF 사용할 수 없는 경우 대안

- 별도 인스턴스 구성



- 프로세스 Fork



- DOM 방식의 기존 방식을 그대로 사용
- 별도의 프로세스로 실행
- 사용되는 JVM 메모리 크기 지정
- 프로세스 Hang 모니터링 / 강제 종료

Opennaru, Inc. © 2017 | All Rights Reserved.

OPENNARU Quick Service 란?

고객사

① 시스템 이슈발생

② 쓰레드 덤프와 모니터링 화면 전달

③ 오픈나루 서비스 데스크 접수

④ 장애 분석

⑤ 장애 분석 보고서 전달

오픈나루

Opennaru, Inc. © 2017 | All Rights Reserved.

Innovate your cloud with OPENMARU

Application Performance Management
감사합니다.
openmaru
APM

제품이나 서비스에 관한 문의

콜 센터 : 02-469-5426 (휴대폰 : 010-2243-3394)

전자메일 : sales@opennaru.com

2019년도 신입 · 경력 채용 공고

당신과 **함께** 오픈소스 정상에서 **같이** 하고 싶습니다.

· 홈페이지 **www.opennaru.com/recruit**
· 이력서 접수 **apply@opennaru.com**
· 채용문의 **02-469-5426**

opennaru