



ENPHAROS JAVA v5.0 표준제안서

2015. 11





Agenda

1. 제품개요
2. 아키텍처 및 핵심기술
3. 주요 특징점
4. 차별성
5. 주요기능
6. 기대효과
7. 활용사례

1. 제품 개요

ENPHAROS JAVA는 Web Application Server(WAS)와 WAS에서 실행되는 JAVA 프로그램의 성능, 자원소모량 등 성능관리 지표를 검출하여 성능감시, 성능진단 분석, 장애 관제 및 전파를 수행하는 애플리케이션 성능관리(APM) 솔루션입니다.

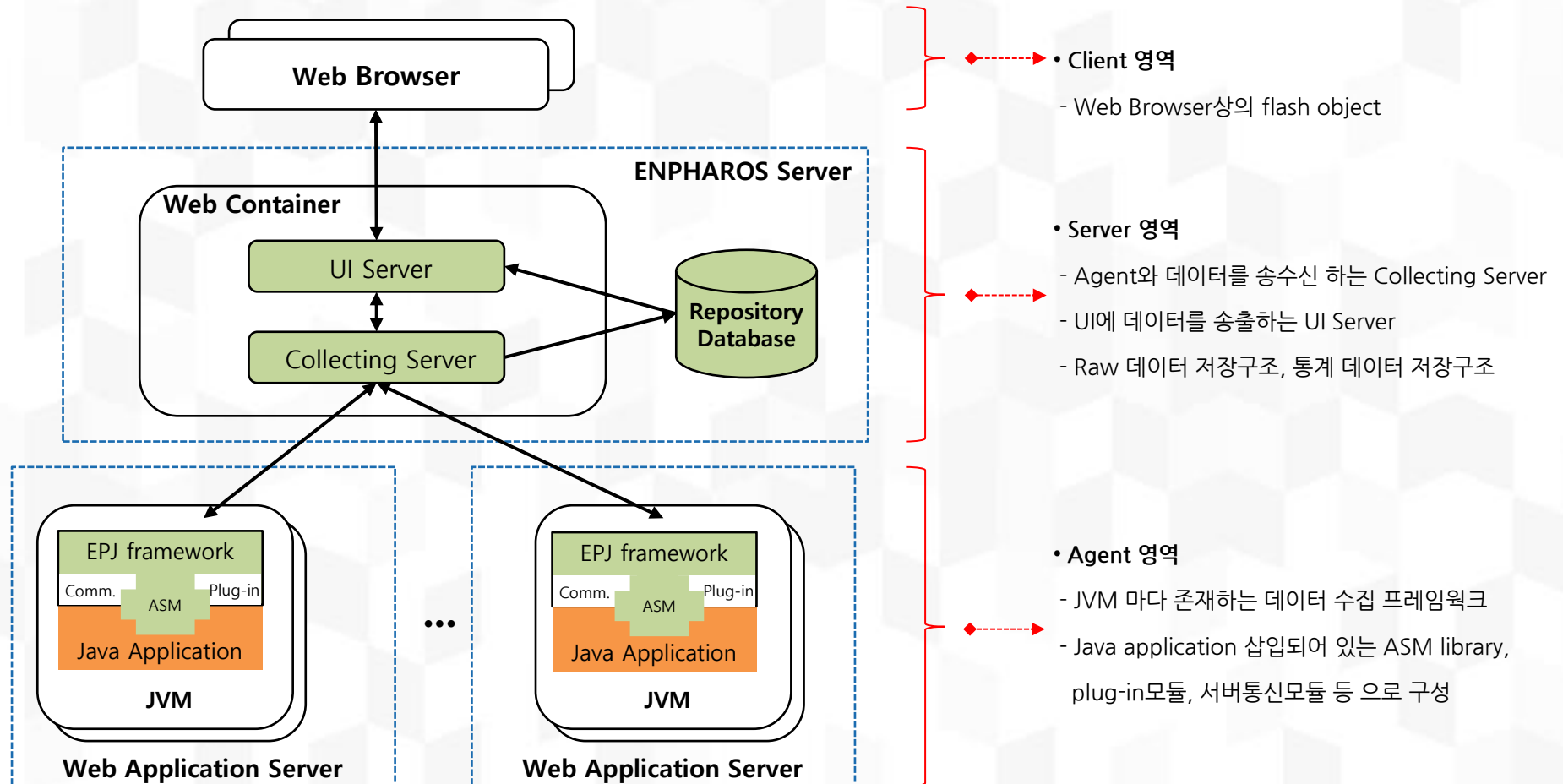
제품명	ENPHAROS JAVA
버전	Version 5.0
제조사	(주)다봄소프트
주요기능	<ul style="list-style-type: none">▪ 대시보드를 통한 WAS 관제▪ 실시간 서비스 성능모니터링▪ JVM 상태 모니터링▪ 자바 애플리케이션 분석▪ 통계기반의 정량적 분석 및 성능 튜닝▪ 분석 데이터에 대한 정형, 비정형 보고서▪ 장애 원인 진단 및 분석▪ 자동화된 운영 관리▪ 솔루션 자체 보안

특징	내용
완전 실시간 모니터링	<ul style="list-style-type: none">▪ 초 단위 간격에 기반한 성능 / 장애 데이터 수집▪ 성능자료의 실시간 가공 및 모니터링▪ 자원 사용현황에 대한 실시간 모니터링▪ 장애 이벤트 발생 즉시 전파
최고 안정성 확보	<ul style="list-style-type: none">▪ Agent, Server, UI의 3계층 기반 아키텍처 채용으로 Agent의 최소역할▪ 업계 최경량 라이브러리인 ASM 기술 채택▪ 5대 시중은행 중 4곳에서 사용중인 검증된 안정성
통계 분석 및 통합관리	<ul style="list-style-type: none">▪ 업계 최소 통계 주기인 1분 주기 분석데이터 제공으로 정밀한 통계추이 산출▪ 추이분석, 비교분석 차트 지원▪ SLA 설정 방식에 의한 수준별 관리방안 제공▪ 사용자 정의에 의거한 보고서 출력



2.1 제품 아키텍처

ENPHAROS JAVA는 모니터링 대상 시스템에 최소의 간섭으로 안정성있게 관리되도록 경량구조의 Agent 영역과 수집된 정보를 가공하고 조작하는 Server 영역, Client 영역의 3 티어 아키텍처로 구성됩니다.



2.2 ENPHAROS Agent 핵심기술

ENPHAROS JAVA는 JAVA 모니터링을 위하여 널리 이용되는 BCI(Byte Code Instrumentation)기술 중에서 가장 성능이 우수한 ASM 라이브러리를 채택하여 다른 유사 Library대비 운영시스템 부하를 최소화 하였으며 Agent에서는 최소한의 필수 동작을 하도록 설계하여 모니터링 대상서버에 대한 안정성을 높였습니다.

BCI 라이브러리 종류 및 비교



【ASM】



【BCEL】



【SERP】

【 BCI library 크기 및 로딩 타임 상호 비교 】

항목	ASM	BCEL	SERP
Size ¹⁾	25KB	350KB	150KB
Time ²⁾	Short	Middle	Long

※ 비교

1) 크기 : 라이브러리 크기가 작을수록 CPU Memory 소비가 적음

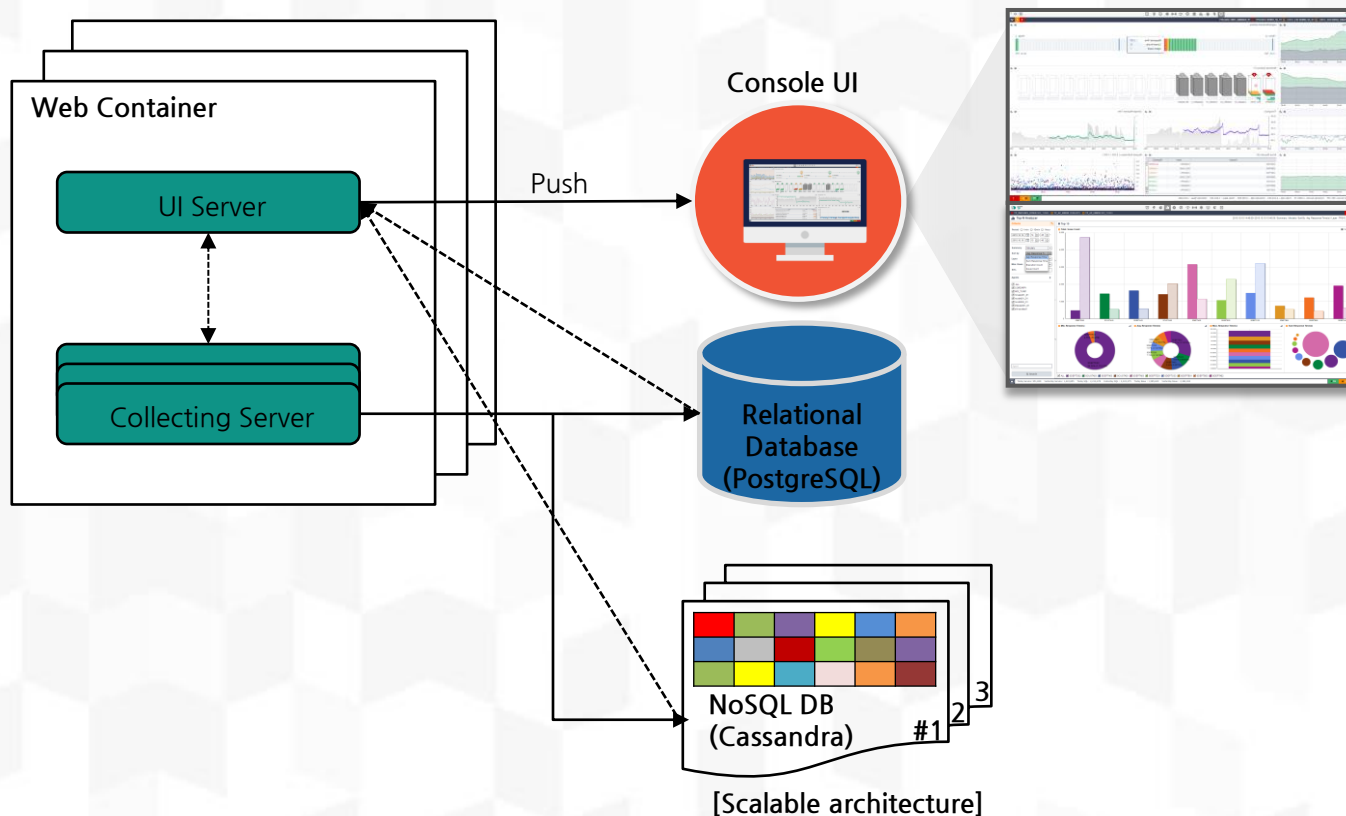
2) 시간 : class loading시 Instrumentation time 기준

(출처 : <http://java-source.net/open-source/bytecode-libraries>)

- BCI Library에는 ASM, BCEL, SERP등 다수가 존재하며 ENPHAROS JAVA는 ASM Library를 기본으로 채택하여 적용함
- ASM은 비교 대상인 BCEL의 7%, SERP의 17% 정도 크기를 갖는 경량의 library로서 상대적으로 CPU ,Memory 소비가 작은 특징임
- ASM은 JVM위에 모니터링 대상 application class들이 loading시 dynamic하게 이루어지는 instrumentation 시간이 짧아 상대적으로 application의 실행이 빠른 특징이 있음

2.3 ENPHAROS Server 핵심기술

ENPHAROS 를 구성하는 서버모듈에는 수천 tps급 성능 데이터를 안정적으로 가공 후 저장하기 위한 분산 병렬 storage 기술이 적용 되며 실시간으로 UI에 표현하기 위한 Push 기술이 내장되어 있습니다.



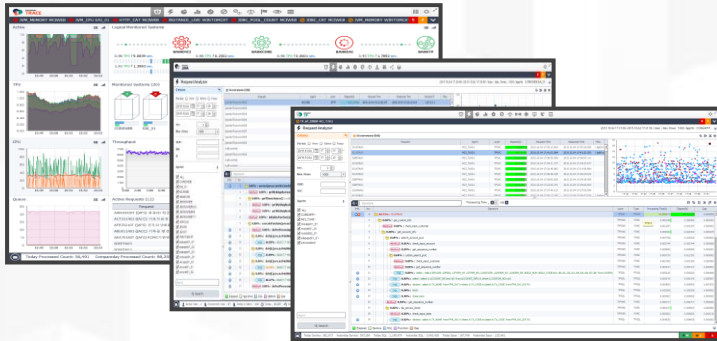
- 고속 대용량 성능데이터를 수집 저장하기 위하여 ENPHAROS Server는 병렬처리 구조를 채택함 (Web Container)
- 데이터를 안정적으로 저장하기 위하여 통계용 데이터는 RDBMS에 저장하며 Raw 데이터는 NoSQL DB에 저장하며 늘어나는 데이터를 안정적으로 처리하기 위하여 Scale-out 방식의 Scalable architecture를 채택함
- Console UI에 고속, 실시간 데이터 전송을 위해 Push 기술을 적용함
- 대용량 데이터 적재 및 삭제등 관리용 self automation 기술을 적용함

3.1 주요 특징점 (1/2)

애플리케이션에 대한 추가 공정과 오브젝트 변경 없이 단순 설치와 설정만으로도 WAS의 애플리케이션에 대해 초단위로 모니터링 데이터를 수집하여 저장, 분석, 가공, UI 표현을 함으로써 24시간 365일 수많은 트랜잭션을 처리하는 대규모 WAS 서버 환경에서 특히 효용성이 탁월합니다.

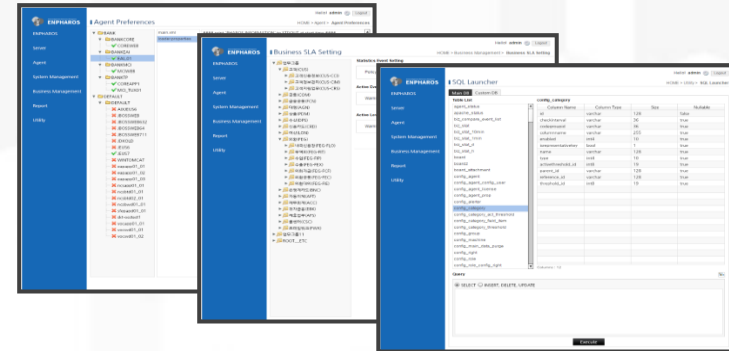
강력한 실시간 모니터링과 연결 상태 분석 모니터링

- 하나의 화면에서 모든 실시간 유형의 모니터링이 가능(시스템 리소스, JVM 상태, 거래 처리 성능, 지연 처리 거래들에 대한 목록 및 거래 내용, 완료된 거래에 대한 성능지표)
- 각 프로세스들에 대한 실시간 시스템 리소스 사용률 제공
- Connected 사용자수, Active 사용자수, Long 트랜잭션 검출 등



애플리케이션 변경 없이 설정만으로 구성 및 제어

- WAS 서비스에 대한 소스 변경 없이 설치 후 바로 적용이 가능하며, 삭제 또한 WAS 서비스와의 종속성이 없어 손쉽게 가능함
- Admin 화면을 통한 WAS 모니터링 설정 및 구성에 대한 중앙집중 관리
- SLA를 통한 손쉬운 정책관리 (이벤트설정, 업무구성설정, 통계설정 등)

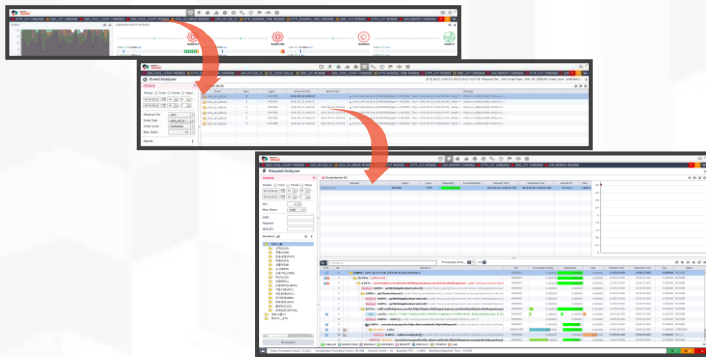


3.2 주요 특징점 (2/2)

ENPHAROS는 대규모로 수집되는 애플리케이션 성능정보를 빅 데이터 기반의 시계열 분석기술을 이용하여 사용자 관점의 모니터링이 될수 있도록 구성하였으며 업계 최소인 1분 통계 주기를 적용하여 정밀한 통계분석 자료를 제공합니다. 이를 통해 보다 안정적인 장애 예방과 신속한 장애 감지가 가능합니다.

장애 예방/감지/대응을 위한 직관적 모니터링 및 탁월한 분석 기능

- Time Slice 에 따른 다양한 관점의 통계 차트 제공으로 거래성능의 손쉬운 패턴화와 이를 비교/분석 할 수 있는 기능
- 각 종 통계 차트에 따른 거래성능 분포도 연계와 거래성능 분포도를 구성하고 있는 개별 거래에 대해 상세분석 화면과의 연계
- 시인성이 강조된 이벤트의 손쉬운 관리와 이벤트 발생 시 거래 상세분석 화면과의 원스텝 연계를 통한 근본 원인 분석



사용자 관점에 따른 최적의 실시간 모니터링 환경

- 사용자 관점의 실시간 통합 대시보드 화면 제공(기본 4개)
- 거래부하와 처리 응답시간에 따른 여러 실시간 차트들과 현재 처리 중인 거래에 대한 정보와 이들 거래들이 각 시스템에 미치는 영향을 실시간으로 다양한 이미지로 표현하여 제공
- 사용자 편의성을 강조한 꾸준한 모니터링 환경 업그레이드 정책



4. 차별성

ENPHAROS JAVA는 WAS와 웹 애플리케이션의 성능 분석 뿐만 아니라, 각 트랜잭션에 대하여 호출관계도(Call Tree) 내 최하위 메소드와 SQL 단위까지 완벽한 추적 기능과 서비스 통계 데이터를 제공하여 업무 서비스의 안정성과 성능개선에 도움을 줄 수 있는 전문 APM 솔루션으로서 단위 모니터링 수준에 머무르는 타 제품과 큰 차별성을 제공합니다.

실시간 모니터링

- 거래의 발생률, TPS, 응답시간, JVM, CPU/MEM 사용률 등의 실시간 차트들을 통한 종합적인 처리 성능 모니터링
- 과거 특정일 처리 성능 대비 당일 성능 실시간 비교
- 처리 진행 중인 거래에 대한 모니터링

거래추적

- 서비스간 호출 시에 거래 관점에서의 완벽한 전체 Call-Tree 제공
- 바인드 값이 적용 된 SQL 문 추적
- 거래를 구성하는 각종 파라미터 값 제공하여 거래가 어떠한 코드값으로 구성되어 있는지 파악이 가능

다양한 통계

- 다양한 성능지표를 반영한 통계 데이터 및 차트
- 업무코드별 통계 분석(거래코드, 채널코드, IP, 에러코드 ...)
- 함수단위, SQL문 단위의 성능 통계

연계분석

- 철저한 사후분석을 위해 통계 데이터를 구성하는 개개의 거래를 거래상세분석 화면과 연계
- 이를 통해 각 거래의 데이터 값과 Call-Tree 확인으로 장애예방을 위한 조치가 가능

차별성





5. 주요기능

- 5.1 WAS 관제용 대시보드
- 5.2 실시간 모니터링
- 5.3 JVM 모니터링
- 5.4 애플리케이션 상세분석
- 5.5 통계 기반 성능분석
- 5.6 장애관리
- 5.7 보고서
- 5.8 자체보안

5.1.1 대시보드 개요

ENPHAROS JAVA는 소규모에서 부터 대규모로 운영되는 WAS를 통합적으로 관리하기 위한 통합 관제에 최적화된 대시보드를 제공하여 관리포인트를 일원화 하도록 합니다. 성능데이터, 접속자 데이터, 부하량 데이터, 자원 데이터와 같은 관제 핵심적인 정보를 실시간 및 5분 통계에 기초하여 제공 함으로써 관리자에게 높은 가독성과 편의성을 드립니다.



메뉴 표시 영역

이벤트알림 표시 영역

실시간 통합 모니터링 화면은 현재 처리중인 액티브 서비스에 대한 성능정보, 접속자현황, 부하량 정도, 자원소모량과 같은 데이터를 실시간과 최근 5분 통계로 제공하므로써 업무 시스템이 어느 정도의 부하를 받고 있고, 어느 정도의 성능을 발휘하며 처리하고 있는지를 표현하므로 업무 시스템의 전 상황을 확인 할 수 있는 매우 역동적인 화면입니다.

통계정보 표시 영역



5.1.2 대시보드 종류

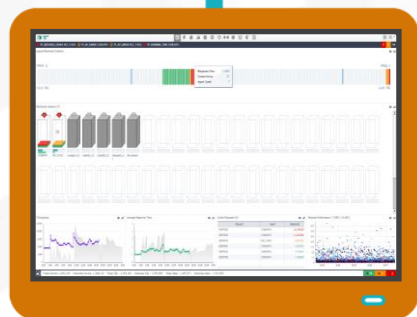
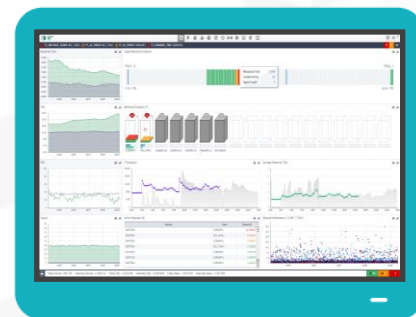
5.1 WAS 관제용 대시보드

WAS 관제를 위해 제공되는 대시보드는 사용자가 보고자 하는 중요 관점에 최적화된 4가지 종류가 존재하며 사용자는 관제해야 할 WAS JVM의 규모에 따라 시스템 중심으로 관제에 최적화된 레이아웃, 또는 서비스 처리 상태 관제에 최적화된 레이아웃을 선택 적용할 수 있어 사용자 선택에 의해 최적화된 대시보드를 사용할 수 있습니다.



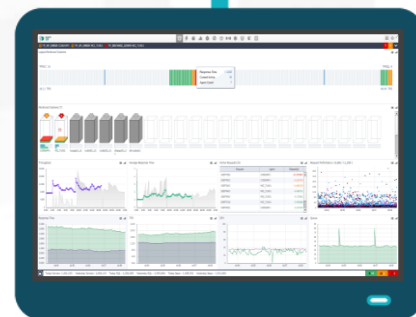
다수의 모니터링 대상
시스템 관점의 관제 - 1

업무 및
거래처리 중심



다수의 모니터링 대상
시스템 관점의 관제 - 2

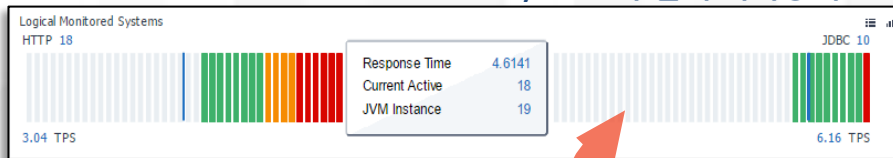
시스템 및
거래처리 중심



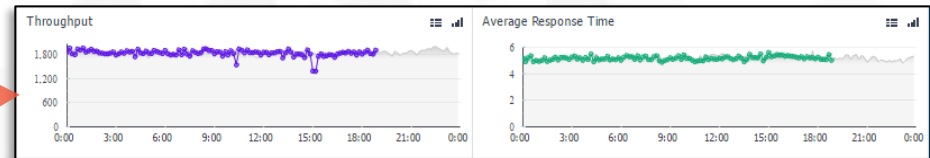
5.2.1 실시간 모니터링 개요

24시간 중단 없는 기업의 웹서비스를 담당하는 WAS는 실시간경영 (Real Time Enterprise, RTE)을 가능하게 하는 핵심 인프라입니다. 따라서 ENPHAROS JAVA는 이 같은 실시간 기업 인프라의 장애발생이나 성능 저하 증상을 즉시 검출하여 관리자가 인지하고 조치하도록 수집되는 대다수의 정보를 real time 기반으로 모니터링하도록 설계되어 있습니다.

DB, WAS의 현재 처리상태



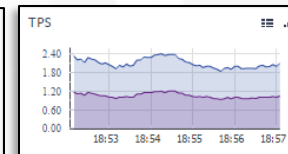
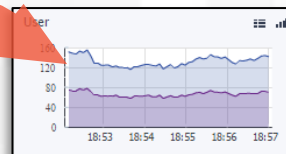
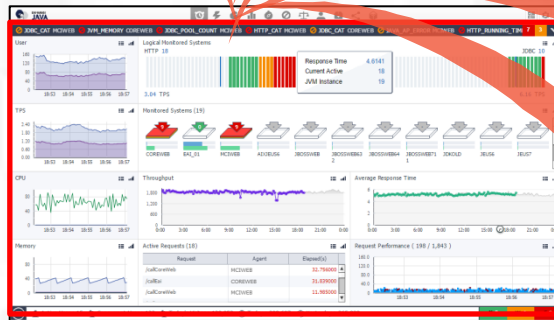
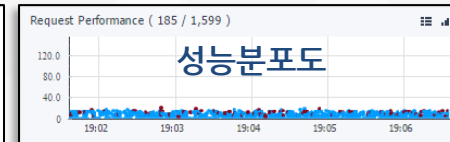
Throughput



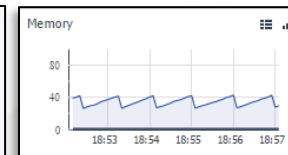
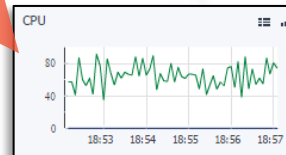
평균적 처리시간

Active Requets

Request	Agent	Elapsed(s)
/calCoreWeb	MCIWEB	138.834000
/calEai	COREWEB	138.373000
/calCoreWeb	MCIWEB	99.324000



접속자, TPS



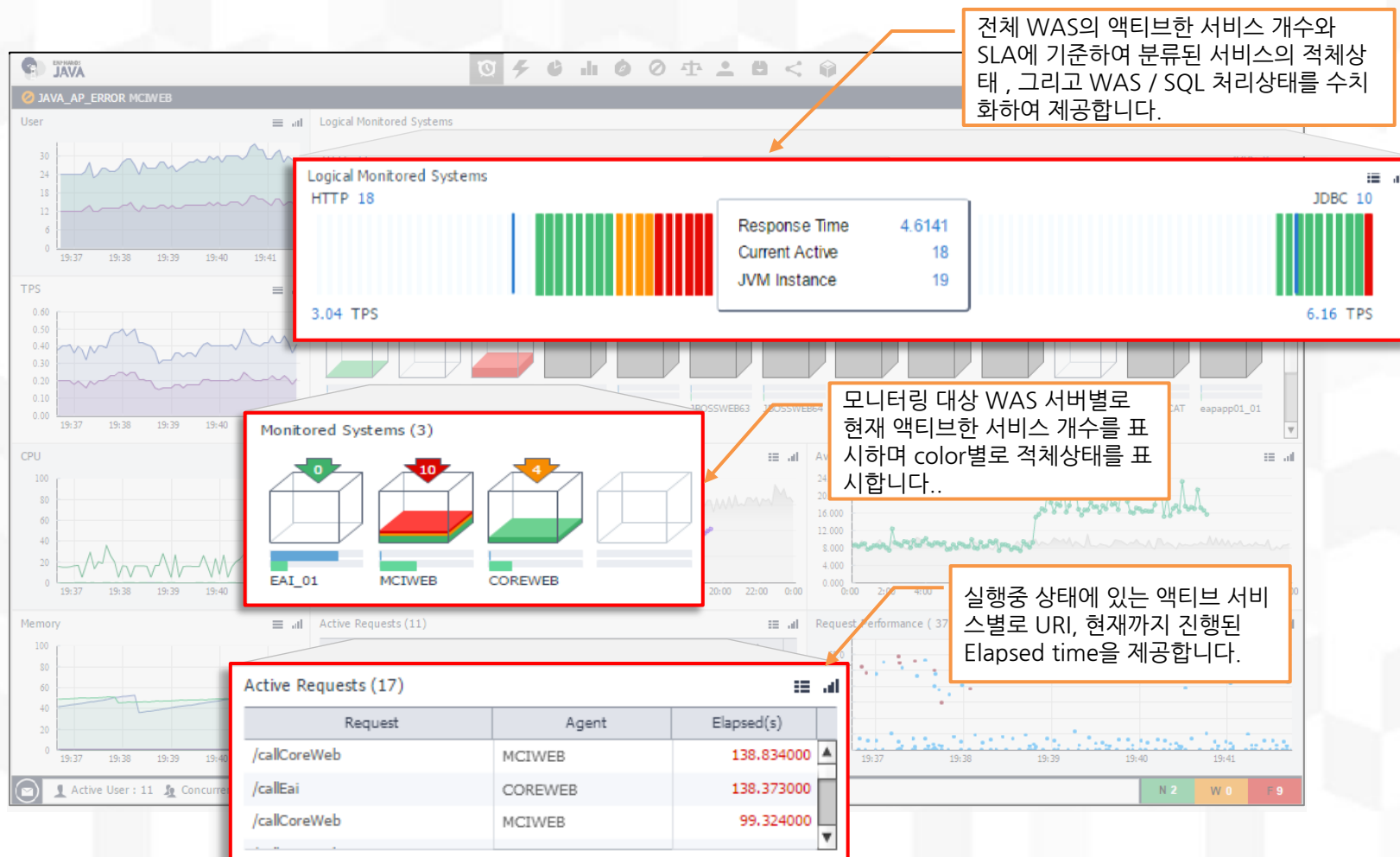
자원사용량



5.2.2 액티브서비스(Active Service)

5.2 실시간 모니터링

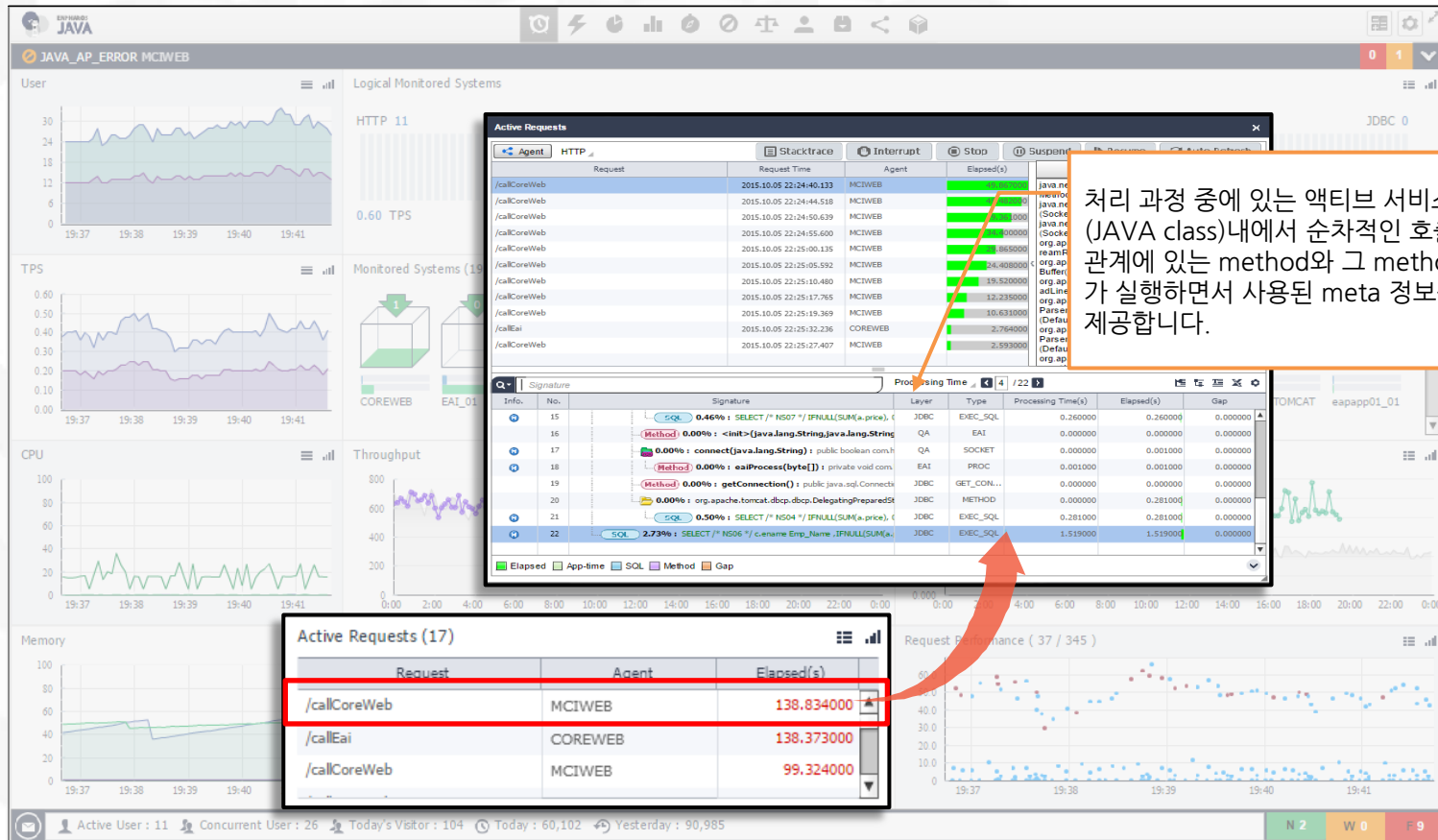
현재 WAS에서 실행중인 서비스를 효과적으로 표현하기 위하여 ENPHAROS JAVA는 3가지 차트를 통하여 현재 실행 중인 서비스의 속도, 서비스의 개수, 서비스명(URI) 등을 제공합니다. 이를 통해 WAS관리자는 모니터링 대상 WAS의 서비스 안정성을 판단할 수 있습니다.



5.2.3 서비스 프로파일링

5.2 실시간 모니터링

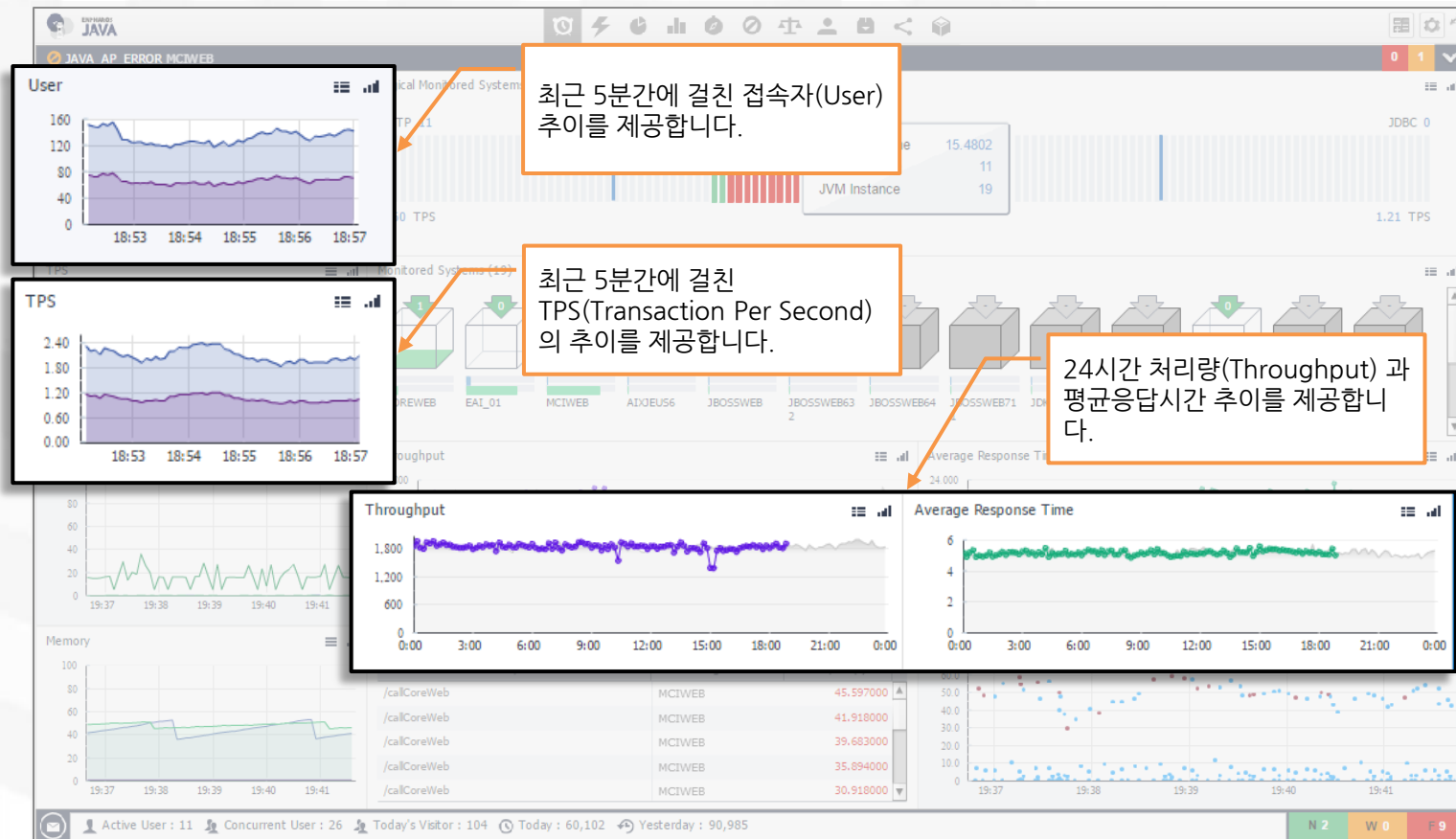
ENPHAROS JAVA의 액티브서비스 기능에서는 현재 수행이 지연 처리되고 있는 서비스를 클릭하여 그 서비스의 내부 처리 로직을 프로파일링 한 메소드 별 콜트리 정보와 소요된 시간을 분석할 수 있습니다. 이를 통해 지연처리 되는 서비스가 있다면 내부 로직의 어떤 지점에서 지연이 일어나는지 진단할 수 있어 효과적인 장애의 해결이 가능합니다.



5.2.4 처리성능 모니터링

5.2 실시간 모니터링

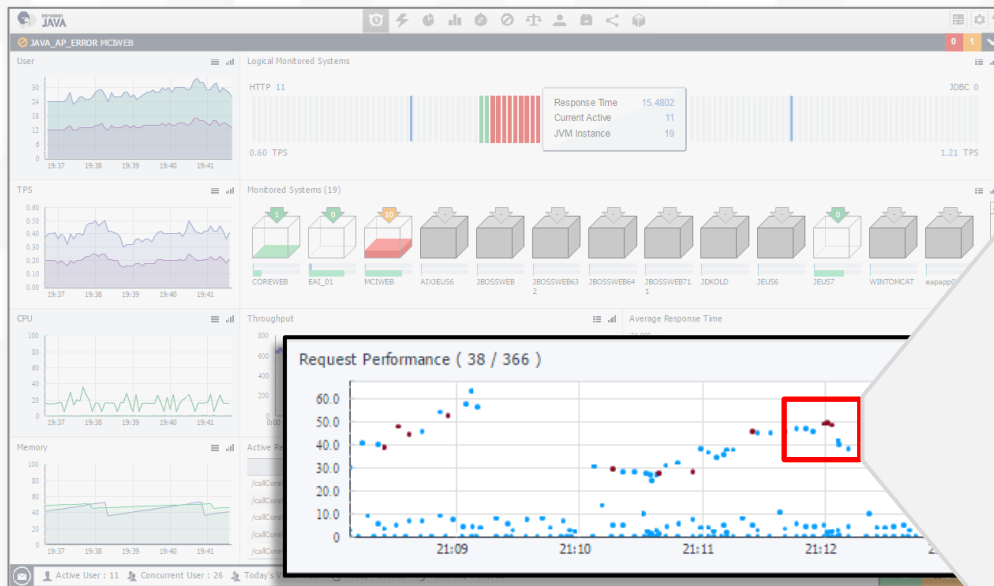
WAS를 모니터링시 현재의 시시각각 변하고 있는 성능정보 수치를 확인하는 것만으로 관제대상 시스템의 건강함을 판단하기 어려울 수 있습니다. 따라서 ENPHAROS JAVA에서는 성능정보가 갖고 있는 가치와 의미에 의거하여 최근 5분간의 추이, 최근 1일간의 추이를 제공하여 성능지표가 어떤 추이를 나타내는지 판단할 수 있도록 도와드립니다.



5.2.5 퍼포먼스뷰

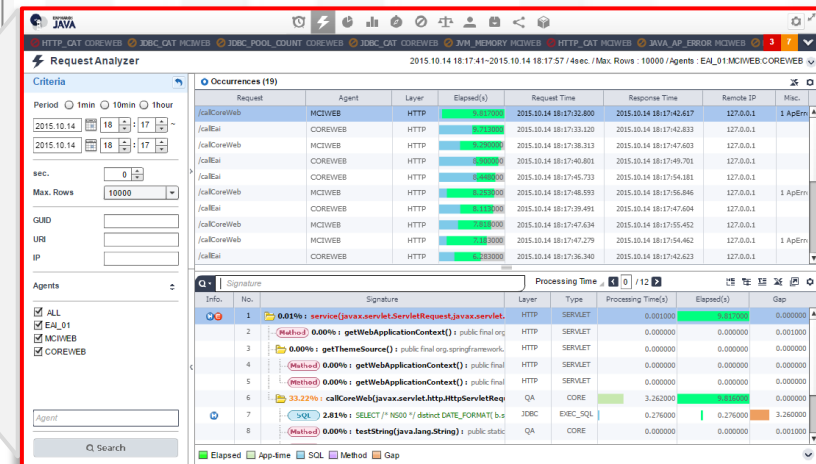
5.2 실시간 모니터링

ENPHAROS JAVA는 서비스의 전체 성능 분포를 표시하기 위하여 WAS에서 처리가 완료된 서비스에 대해 시간대(X축)별 응답시간분포(Y축)를 하나의 점으로 표시하는 성능분포도(Performance View)를 제공합니다.마우스로 분석대상 서비스를 드래그 할 경우 범위에 포함된 점들의 상세한 서비스 분석이 될 수 있습니다.



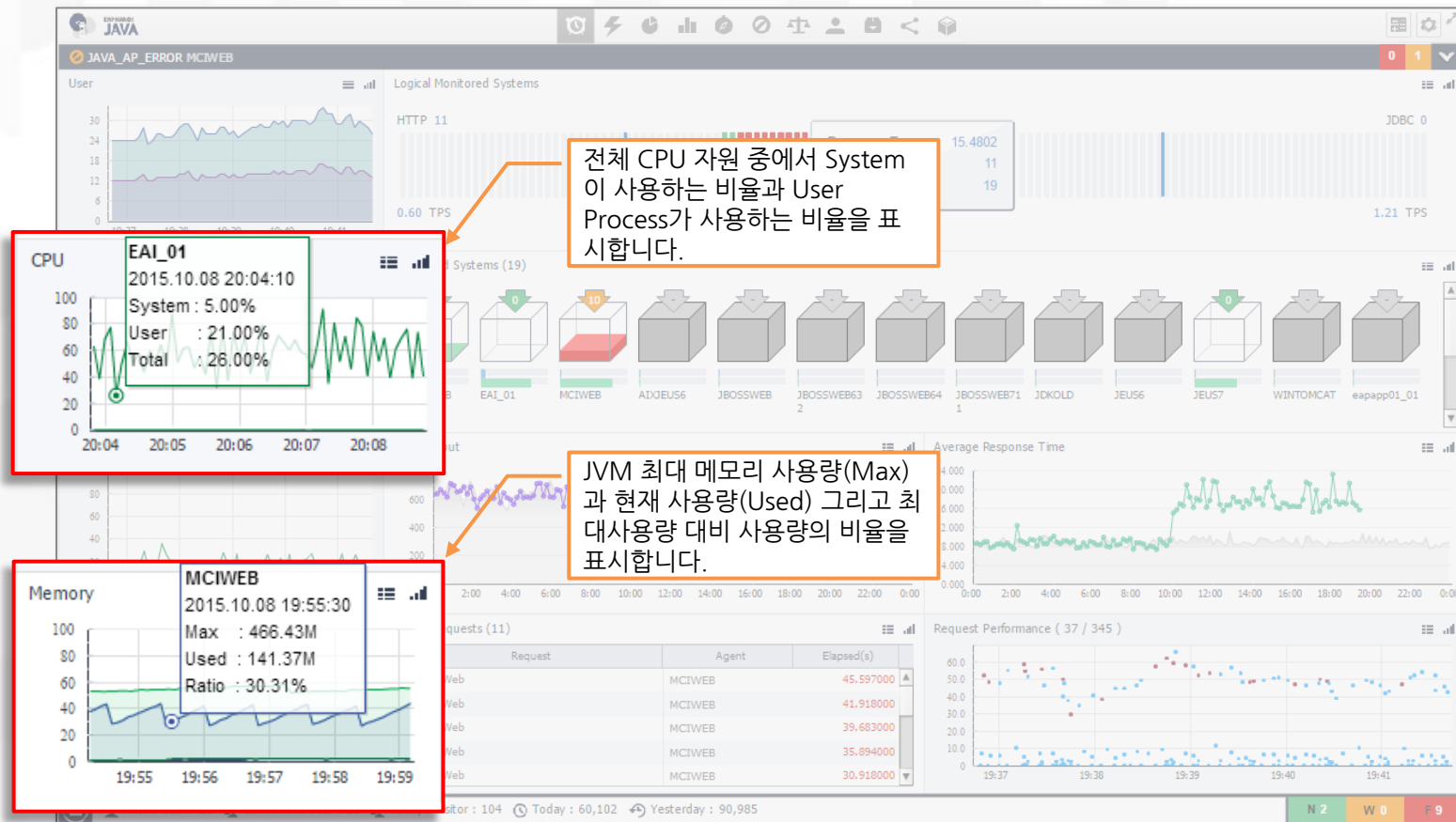
※ 특징점

- 입체적인 분포를 통해 서비스 식별력이 높음
- 마우스 드래그를 통해 분석대상을 쉽게 선별함
- 윈스톱으로 상세 분석이 가능함
- 성능분포 이외에 오류(issue) 분포도 제공됨



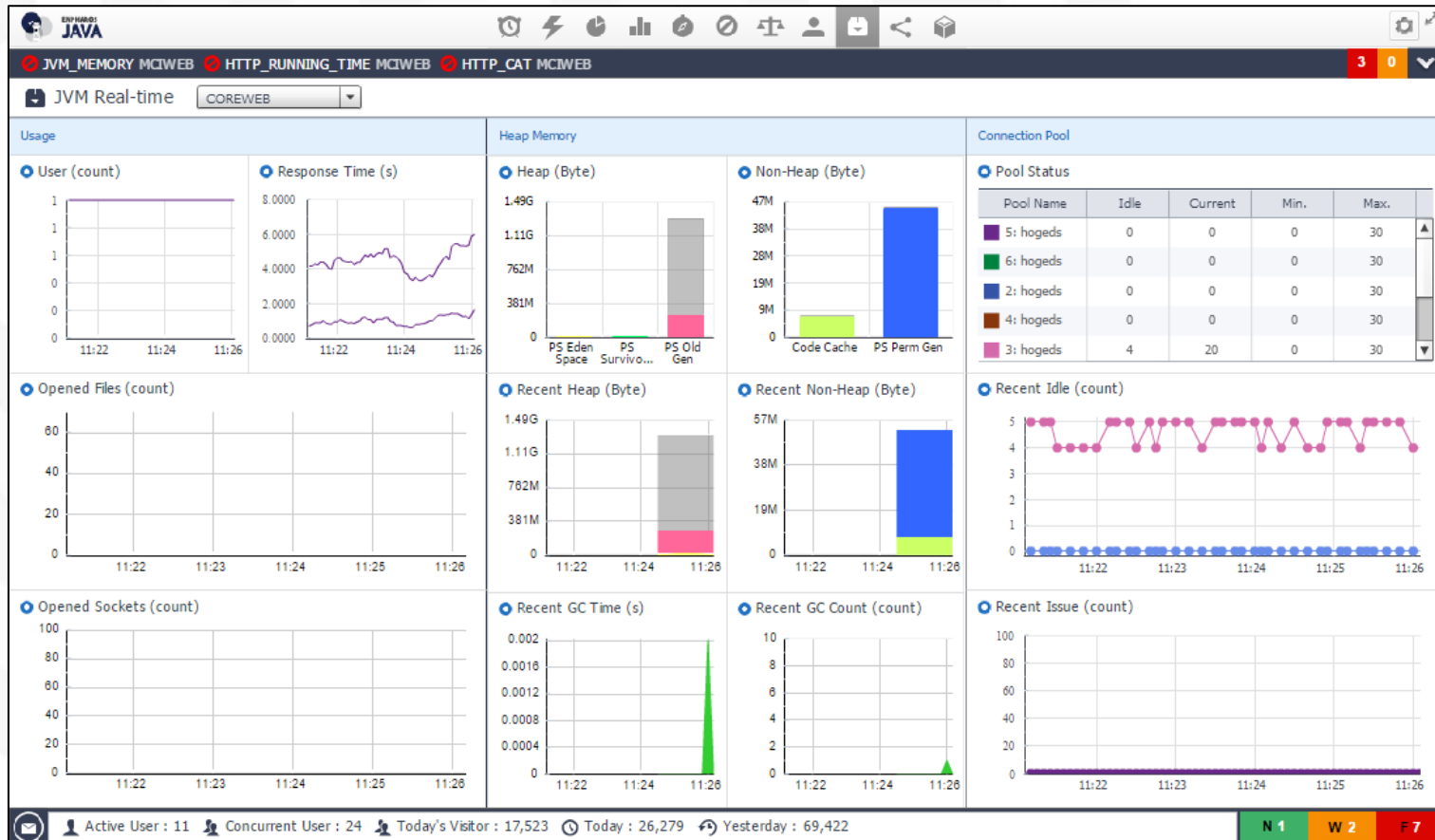
5.2.6 자원 사용량

WAS는 OS의 시스템 자원을 이용하여 서비스를 처리하므로 WAS가 소모하는 시스템 자원 사용량은 장애를 식별하거나 서비스 지체를 발생하는 중요한 요인 되므로 ENPHAROS JAVA는 시스템자원 중에서 CPU, Memory에 대한 사용량을 실시간으로 모니터링합니다.



5.3.1 JVM모니터링 개요

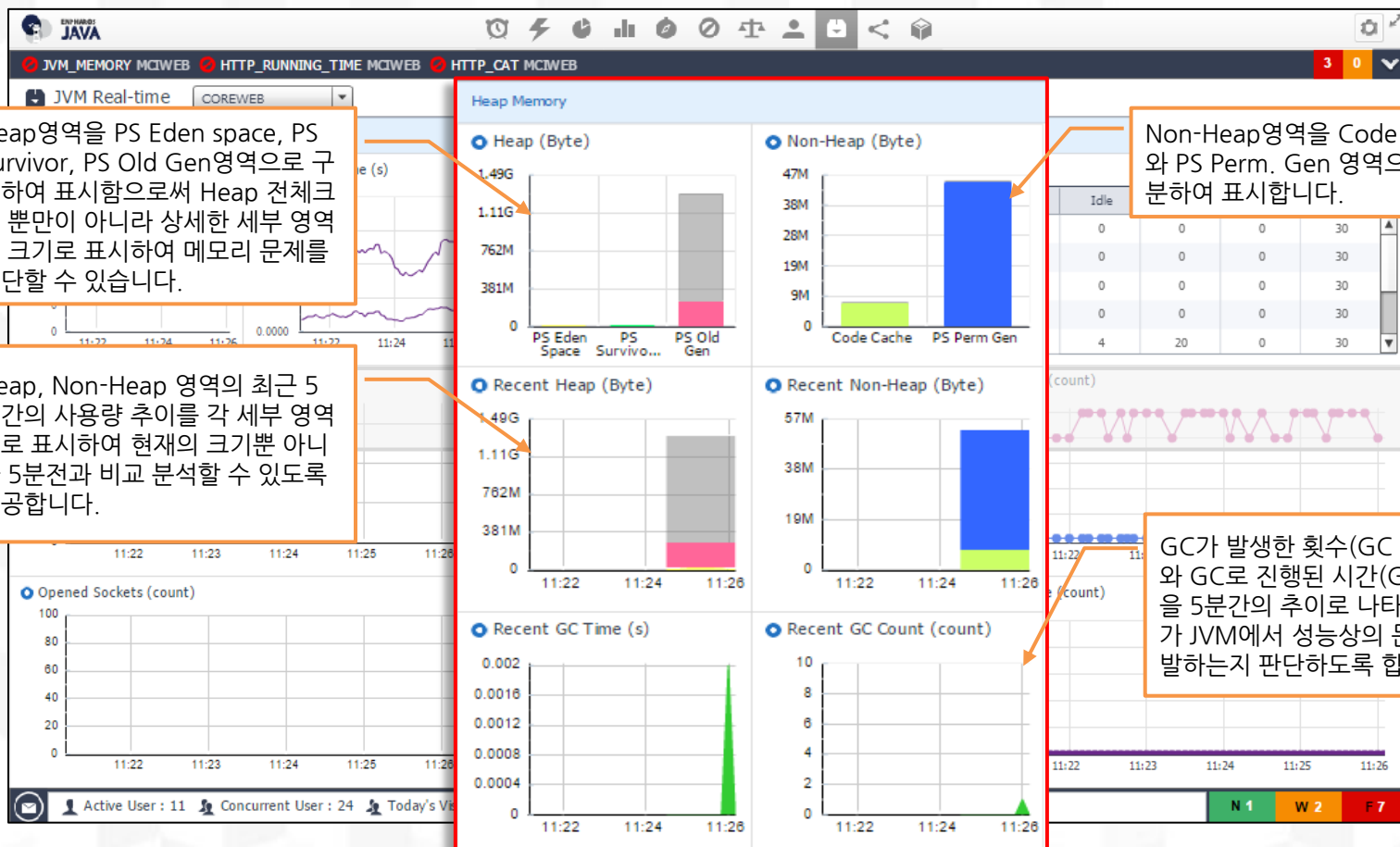
WAS는 한대의 H/W 위에 통상 다수의 JVM 인스턴스를 기동하여 사용할 수 있으므로 서비스가 이루어지는 최소 단위는 JVM이라 할수 있습니다. 따라서 ENPHAROS JAVA에서는 서비스가 통제되는 최소 단위인 JVM별로 내재된 상태정보를 제공하며 JVM엔진이 원활히 동작해야 그 위에서 실행되는 서비스도 원활히 수행되어 JVM의 내재된 상태정보는 서비스 모니터링에 있어 중요한 의미를 갖습니다.



5.3.2 힙메모리(Heap memory) 사용량

5.3 JVM 모니터링

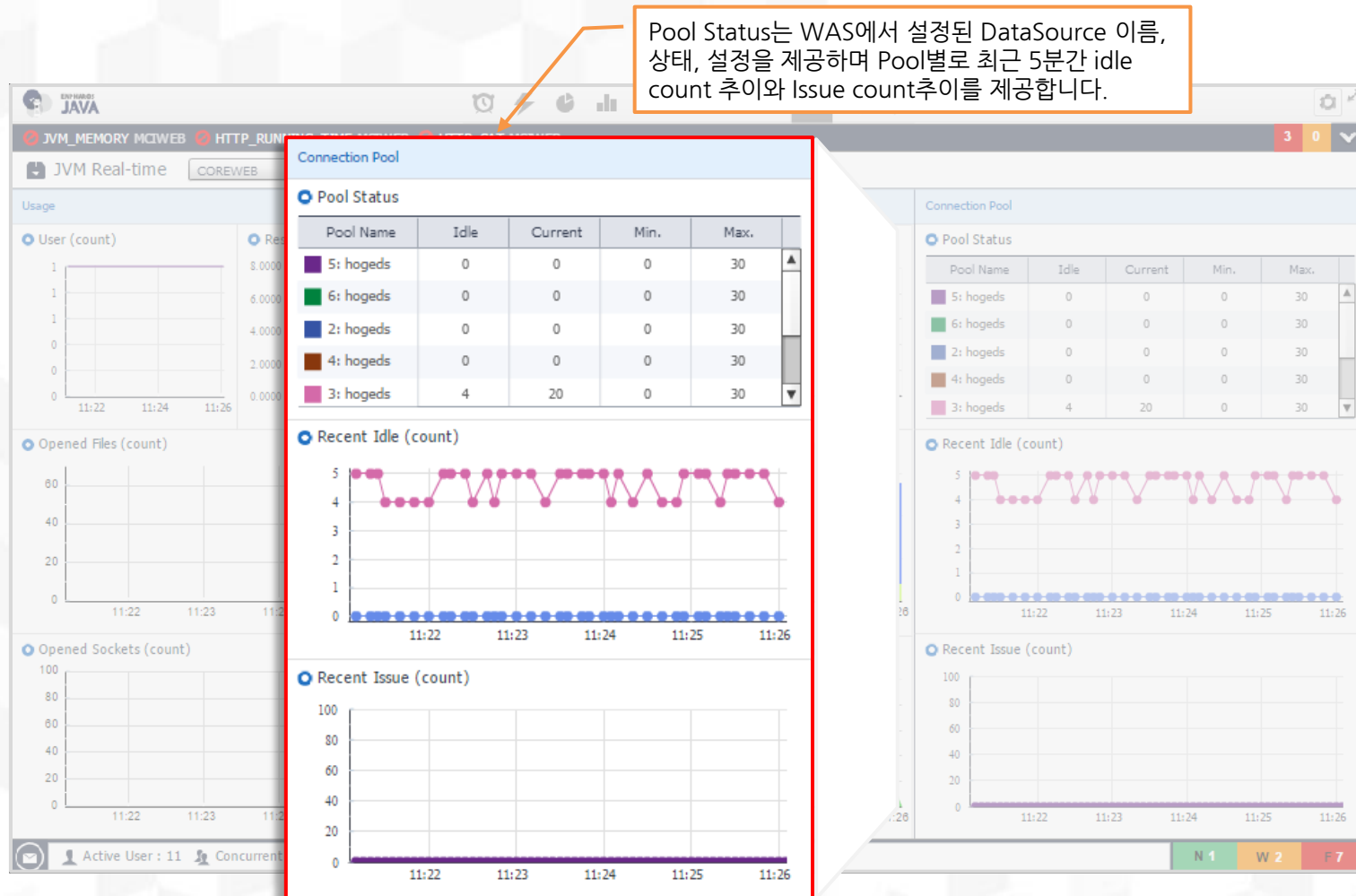
JVM 내의 메모리는 내부 관리 메커니즘에 의거하여 사용된 후 GC 과정을 거쳐 다시 회수되는 등의 반복적인 재사용이 일어납니다. 여유 메모리가 부족할 경우 서비스 지체와 같은 현상이 발생 가능하므로 ENPHAROS JAVA에서는 JVM 내부에서 관리되는 힙 메모리를 주요한 영역별로 세부 구획하여 상세한 사용량과 사용추이를 제공하여 관리자가 좀 더 향상된 관리를 할 수 있도록 합니다.



5.3.3 DataSource 모니터링

5.3 JVM 모니터링

WAS에서 실행되는 대부분의 서비스의 비즈니스로직은 테이블에 보관된 데이터를 읽고 쓰고 가공하는 역할을 하도록 개발되어 있습니다. 이때 테이블에 대한 접근은 JDBC 를 통해 이루어지는데 이때 WAS에서 Database와의 연결통로는 ConnectionPool 설정에 의해 이루어 집니다. 따라서 DB Connection과 관련된 DataSource 모니터링은 서비스 성능에 있어 중요한 병목이 될 수 있습니다.



5.3.4 Thread 모니터링

5.3 JVM 모니터링

JVM은 내부적으로 여러 개의 Thread를 설정하여 Thread 단위로 서비스를 실행합니다. ENPHAROS JAVA는 JVM에서 생성된 Thread가 현재 어떠한 서비스를 처리하는지와 그 서비스를 실행하면서 소모하는 CPU 자원량, 상태정보(State)를 제공함으로써 WAS의 장애나 hang-up현상 발생시 어떤 Thread가 무슨 서비스를 처리 도중에 문제가 발생하였는지 모니터링하고 조치를 할 수 있습니다.

Dump History

Dump 내용에 대한 히스토리 기능을 통해 자동으로 설정된 조건에 의해 Thread Dump가 발생시 과거 이력 조회가 가능합니다.

Thread Overview

Thread Overview graph showing CPU usage over time.

Thread List

Name	CPU(%)	State	Layer	Type	Request	Signature
http-bio-8080-exec-125	0.0	RUNNABLE	HTTP			
http-bio-8080-exec-109	0.0	TIMED_WAITING				
http-bio-8080-exec-42	0.0	RUNNABLE				
http-bio-8080-exec-122	0.0	RUNNABLE				
Abandoned connection cleanup thread	0.0	TIMED_WAITING				
http-bio-8080-exec-118	0.0	TIMED_WAITING				
main	0.0	WAITING				
http-bio-8080-exec-88	0.0	RUNNABLE				
http-bio-8080-exec-96	0.0	RUNNABLE				
[THREAD-UFINITE Plug-in Manager]	0.0	TIMED_WAITING				

Thread Details

Full Java Thread dump with locks info

http-bio-8080-exec-125 is in RUNNABLE (running in native)

at java.net.SocketInputStream.read(SocketInputStream.java:129)

at com.mysql.jdbc.io.ReadAheadInputStream.read(ReadAheadInputStream.java:100)

at com.mysql.jdbc.io.ReadAheadInputStream.readFromUnderlyingStreamIfNeeded(ReadAheadInputStream.java:143)

at com.mysql.jdbc.io.ReadAheadInputStream.read(ReadAheadInputStream.java:173)

locked com.mysql.jdbc.io.ReadAheadInputStream@76830

at com.mysql.jdbc.MysqlIO.readFully(MysqlIO.java:211)

at com.mysql.jdbc.MysqlIO.readAndReadPacket(MysqlIO.java:337)

at com.mysql.jdbc.MysqlIO.readAndReadPacket(MysqlIO.java:337)

at com.mysql.jdbc.MysqlIO.checkErrorPacket(MysqlIO.java:354)

at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:2562)

at com.mysql.jdbc.MysqlIO.sendQuery(MysqlIO.java:2530)

locked com.mysql.jdbc.JDBC4Connection@7668279

at com.mysql.jdbc.PreparedStatement.executeInternal(PreparedStatement.java:1907)

locked com.mysql.jdbc.JDBC4Connection@7668279

Thread Control

Stacktrace Interrupt Stop Suspend Resume

Dump

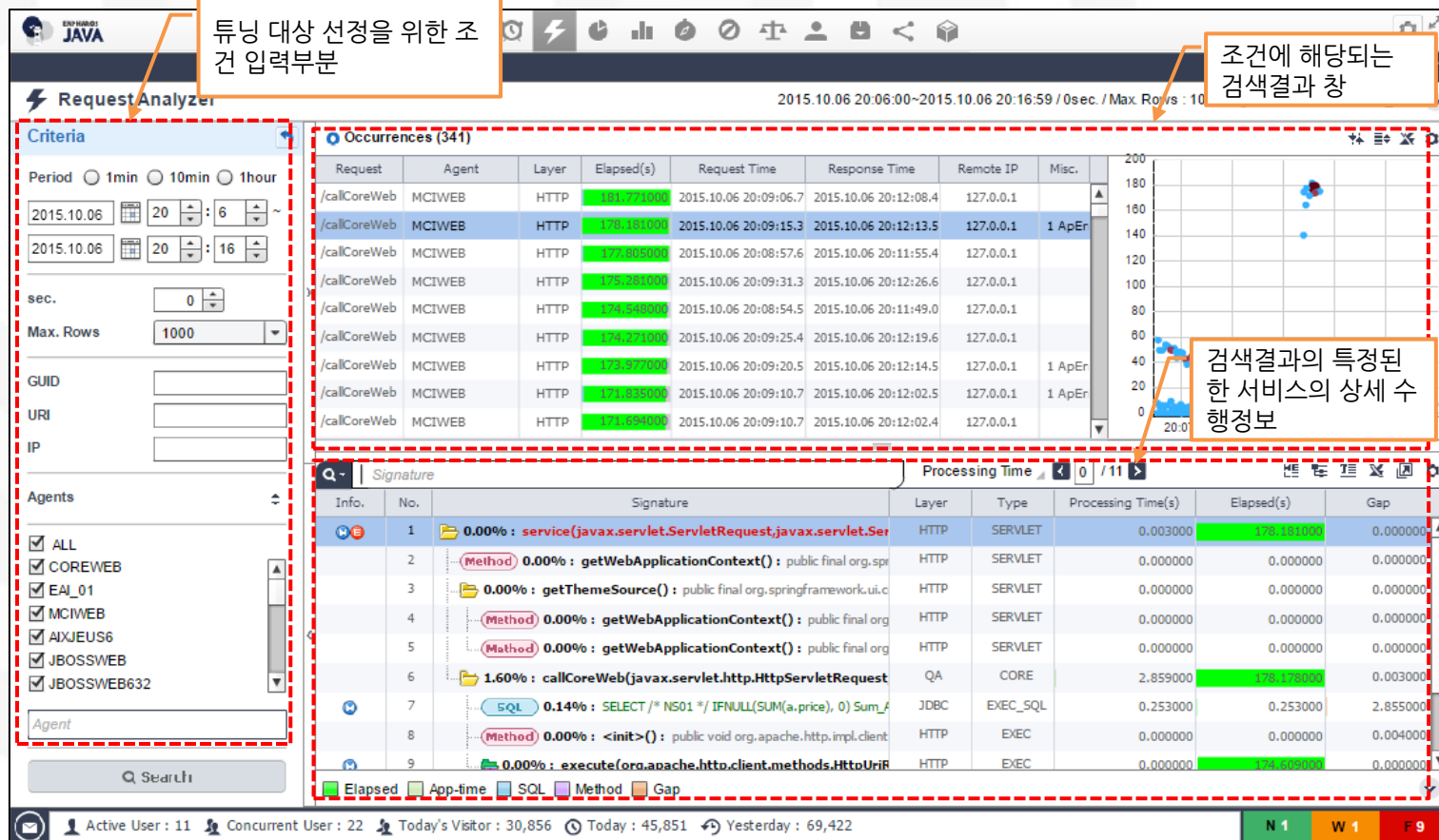
Dump 명령어를 통해 당시 수행되던 서비스의 Full Thread Dump를 팝업으로 제공합니다..



5.4.1 애플리케이션 성능분석 개요

5.4 애플리케이션 성능분석

WAS에서 처리되는 개개의 서비스는 ENPHAROS JAVA에서 Request로 표시합니다. Request Analyzer는 개별 Request 튜닝을 목적으로 제공되는 기능으로서 관리자가 애플리케이션 성능을 튜닝시 수행관련 정보(내부 콜트리 구조, 수행시간, 자원사용량, 요청한 ip등등)를 제공하므로 애플리케이션 개발자나 WAS 운영자가 성능을 향상시키기 위한 도구로 사용할 수 있습니다.



5.4.2 Application 검색기능

대용량 WAS에서 처리되는 수 많은 서비스들의 수행 히스토리에서 문제가 되는 부분을 선별하는 것은 튜닝을 하는 관리자에게 고된 작업입니다. ENPHAROS JAVA에서는 간단한 조건을 입력하므로 관리자가 원하는 대상을 선별하는 Application Search 기능을 내장하여 제공합니다.

WAS에서 처리된 서비스는 너무 많으므로 관리자는 특정한 조건에 맞는 서비스를 대상으로 검색하도록 Criteria 설정을 통한 Search 기능을 제공합니다.

조건에 의해 검색된 서비스는 URI request, 실행된 WAS, 수행시간, 요청한 IP 등과 같이 Request List를 제공합니다.

The screenshot displays the ENPHAROS JAVA Application Search interface. On the left, the 'Criteria' sidebar allows users to filter search results by Period (1min, 10min, 1hour), GUID, URI, IP, and Agents. The main area shows a table of 'Occurrences (341)' with columns for Request, Agent, Layer, Elapsed(s), Request Time, Response Time, and Remote IP. A red box highlights the 'Criteria' sidebar and the 'Occurrences' table. A red arrow points from the 'Search' button in the sidebar to the table. A line graph on the right shows performance metrics over time.

Request	Agent	Layer	Elapsed(s)	Request Time	Response Time	Remote IP	Misc.
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:09:06.7	2015.10.06 20:12:08.4	127.0.0.1	
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:09:15.3	2015.10.06 20:12:13.5	127.0.0.1	1 ApEr
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:08:57.6	2015.10.06 20:11:55.4	127.0.0.1	
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:09:31.3	2015.10.06 20:12:26.6	127.0.0.1	
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:08:54.5	2015.10.06 20:11:49.0	127.0.0.1	
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:09:25.4	2015.10.06 20:12:19.6	127.0.0.1	
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:09:20.5	2015.10.06 20:12:14.5	127.0.0.1	1 ApEr
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:09:10.7	2015.10.06 20:12:02.5	127.0.0.1	1 ApEr
/callCoreWeb	MCIWEB	HTTP	0.000000	2015.10.06 20:09:10.7	2015.10.06 20:12:02.4	127.0.0.1	

0.00% : getThemeSource() : public final org.springframework.util.
(Method) 0.00% : getWebApplicationContext() : public final org.
(Method) 0.00% : getWebApplicationContext() : public final org.
1.60% : callCoreWeb(javax.servlet.http.HttpServletRequest
5SQL 0.14% : SELECT /* NS01 */ IFNULL(SUM(a.price), 0) Sum_f
(Method) 0.00% : <init>() : public void org.apache.http.impl.client.
0.00% : execute(org.apache.http.client.methods.HttpUriR

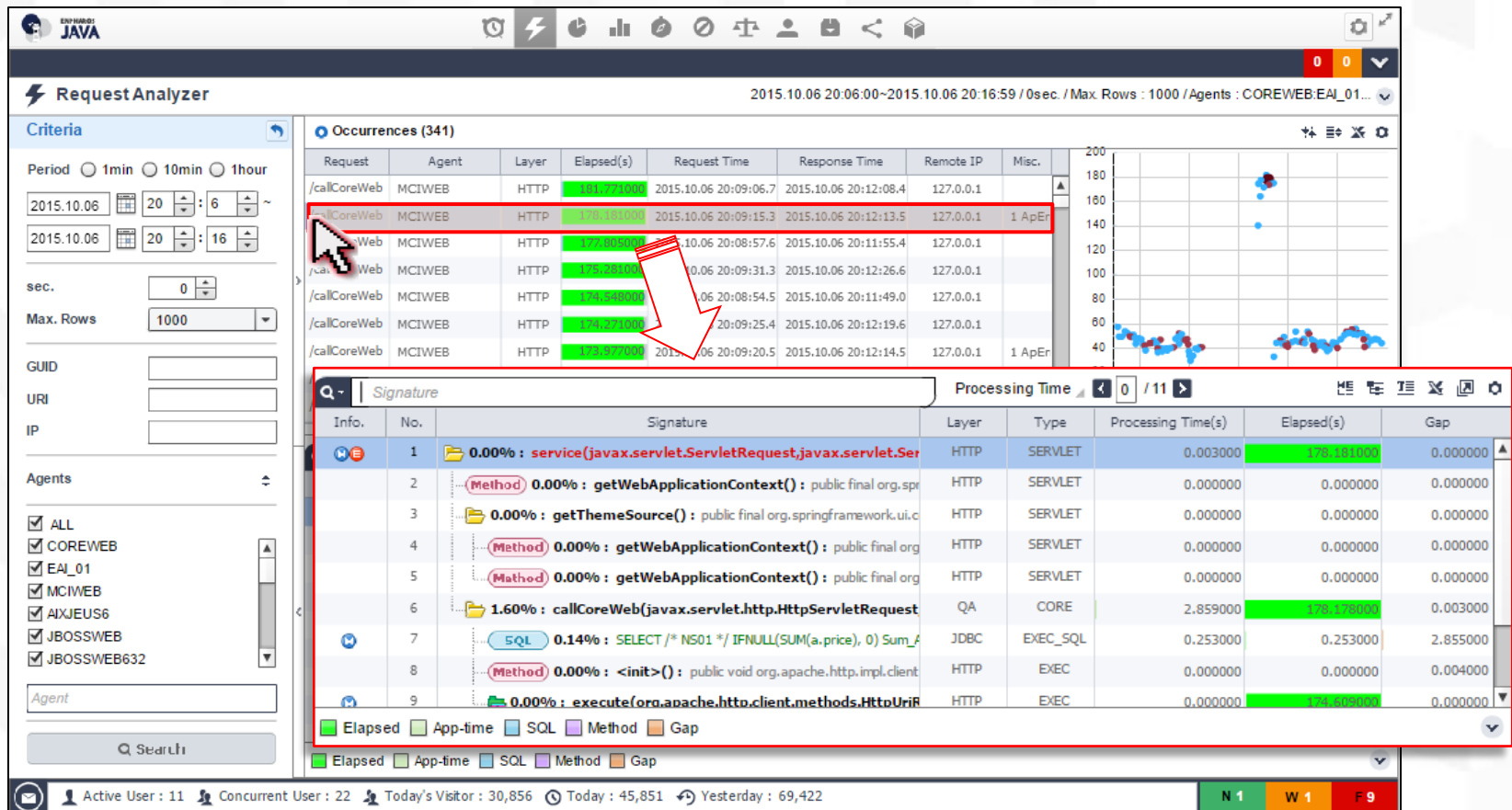
Elapsed App-time SQL Method Gap

Active User : 11 Concurrent User : 22 Today's Visitor : 30,856 Today : 45,851 Yesterday : 69,422



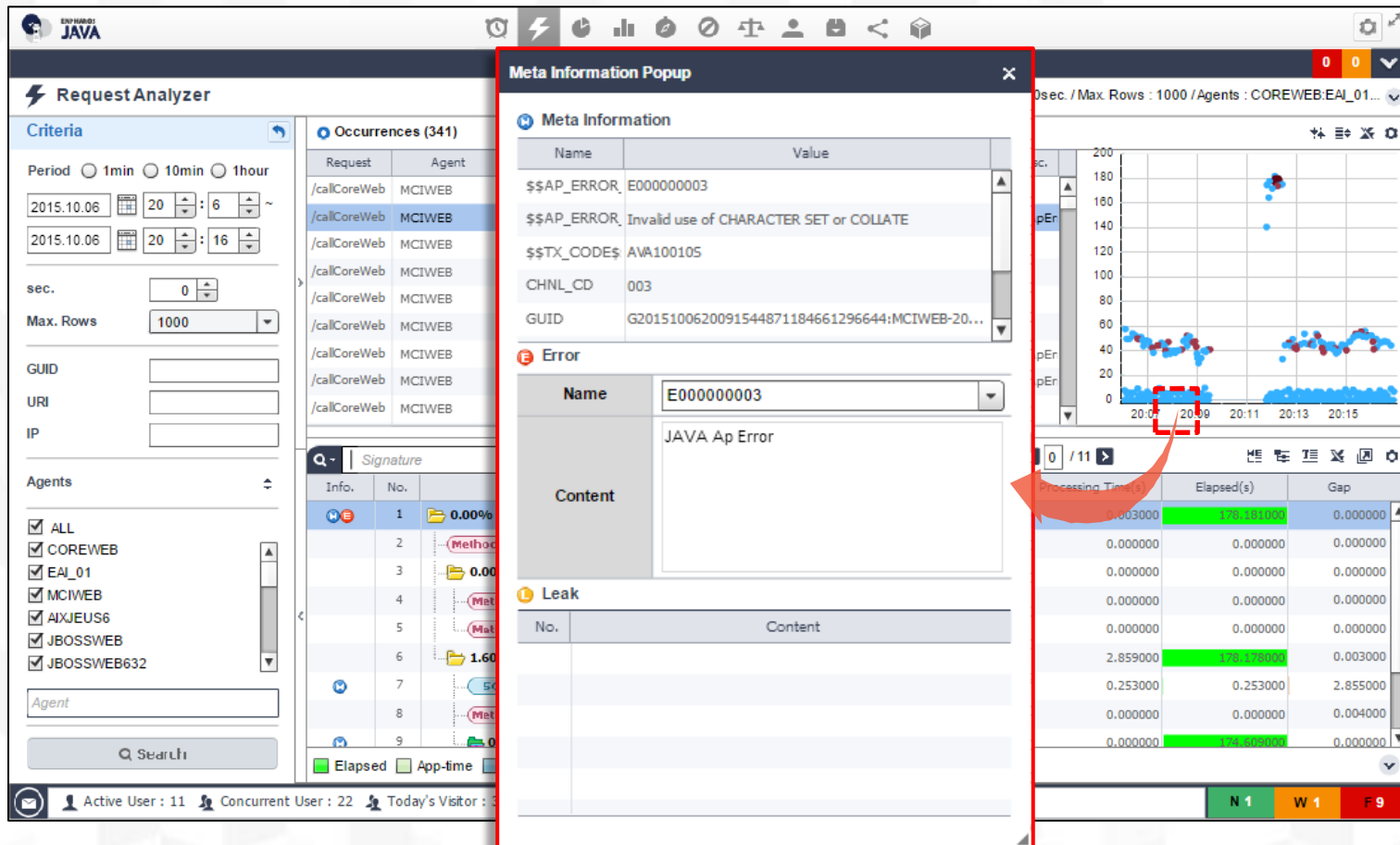
5.4.3 Call Tree 분석기능

ENPHAROS JAVA는 애플리케이션 코드에 변경이나 인위적인 수고 없이 애플리케이션 내부의 호출된 method, SQL문을 순서대로 수집하여 Call Tree 관계도를 제공합니다. Call Tree는 서비스내에 호출된 순서를 인지하도록 계층 구조도 형식으로 제공되며 한 서비스 내에서 호출 단위별로 수행 시간, 백분율을 표시하여 문제된 호출을 식별하는데 높은 가독성을 제공합니다.



5.4.4 실행 메타정보

특정 애플리케이션 서비스가 실행시 관계된 메타정보로는 해당 서비스를 특정 짓는 각종 GUID(Global Unique ID) 와 채널코드와 같은 것이 있어 다른 서비스와 연계되어 처리된 경우 연관성의 추적에 용이하며 또한 그 서비스가 실행시 발생된 오류코드나 오류 메시지 지점코드, 입력 전문등 다양한 설정정보를 제공하므로 분산 처리되는 환경에서 높은 유용성을 제공합니다.



5.4.5 SQL 상세분석

기업 업무용 서비스는 대부분 DB 테이블을 액세스하기 위하여 SQL statement가 서비스 내부에 존재하게 되는데 많은 경우성능의 저하를 일으키는 중요한 곳이 SQL 코드 부분입니다. 따라서 ENPHAROS JAVA는 서비스 내에 존재하는 SQL 코드가 무엇이고 DBMS에 전달된 binding variable는 무엇이며 DBMS에서 SQL 실행에 필요한 실행계획, TABLE정보, INDEX정보를 제공합니다.

DBMS 엔진에 의해 작성된 SQL Execution Plan(실행 계획)과 SQL문이 참조한 TABLE, INDEX의 Analyzed statistics 정보는 SQL Tuning의 기초정보로서 실제적인 튜닝에 강력한 정보로 사용됩니다.

Meta Information

Name	Value
param-count	2
query	SELECT /*+ NO1 */ IFNULL(SUM(a.price), 0)

Instant Query

Summary

Execution Time: 0.000100 sec. Gap: 8 ms. Agent: COREAPP1
End-time: 2015.10.10 14:46:09.638
Start-time: 2015.10.10 14:46:09.637
fetched-rows: 11
param-count: 2
params: p0=[10], p1=[20]
return-value: SqlCode: -3114
source-line-no: 296
source-name: dbio_custom.pc
updated-rows: 0
GID: 20151010dan1028556314460415FA00.COREAPP1-7591320

Original SQL text

select SEQ, ZIPCODE, ZIPSEQ from CUSTOM_INFO where SEQ between '10' and '20'

Applied SQL text

select SEQ, ZIPCODE, ZIPSEQ from CUSTOM_INFO where SEQ between '10' and '20'

Instant Query - SQL Plan

Operation	Options	Object	Object Type	Cost	Rows	Bytes	CPU Cost	I/O Cost
SELECT STATEMENT (OPTIMIZER)				4	0	192	33166	4
TABLE ACCESS	BY INDEX ROWID	BARBARIAN_CUSTOM_	TABLE	4	0	192	33166	4
INDEX	RANGE SCAN	BARBARIAN_ZIPCODE_	INDEX (UNIQUE)	3	0	0	23764	3

Table	Rows	Blocks	Non-Empty	Last Analyzed	Chain	Partitioned	Temporary	Table Space	Avg. Row Len.
CUSTOM_INF	0	13163	13163	02-MAR-15	0	NO	TABSPACE_	N	145

id	name	type	nullable	cardinality	selectivity	low-value	high-value	density	num-nulls	buckets	last-analyze	avg-column	char-length
1	SEQ	NUMBER(8)	NO	1	0	C102	C3384D61	0	0	1	02-MAR-1	5	0
2	ZIPCODE	VARCHAR(10)	NO	114	0	31303030	31353838	0	0	1	02-MAR-1	7	10
3	ZIPSEQ	NUMBER(8)	NO	10424	0	C102	C20416	0	0	1	02-MAR-1	4	0
4	CITYNM_K	VARCHAR(10)	YES	583701	1	BCAD6FEF	BCAD6FEF	1	0	1	02-MAR-1	11	50
5	CITYNM_F	VARCHAR(10)	YES	583684	0	E3C56E75	E3C56E75	1	17	1	02-MAR-1	11	50

name	column-name	type	uniqueness	tablespace	ini-trans	max-trans	blevel	leaf-blocks	distinct-key	clustering-f	num-rows	last-analyze	partitioned
ZIPCODE_	(SEQ)	NORMAL	UNIQUE	TABSPACE_	2	255	2	1094	583701	12300	583701	02-MAR-1	NO

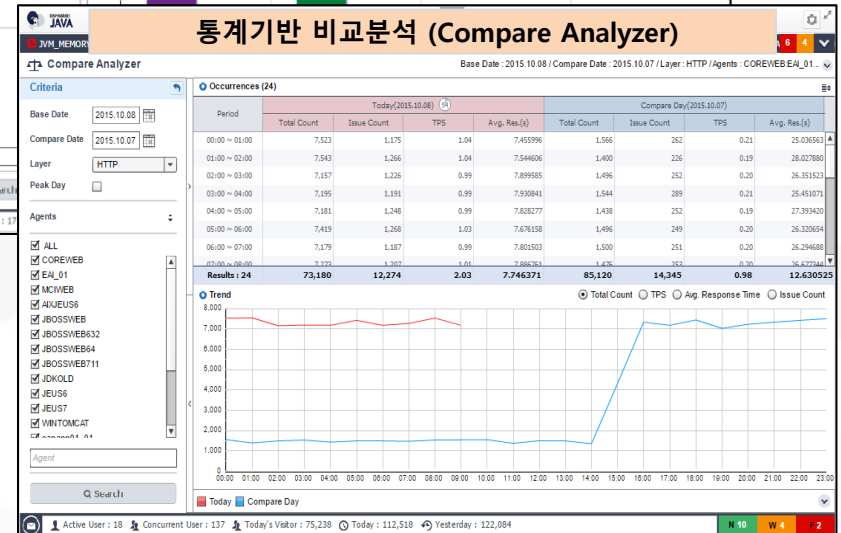
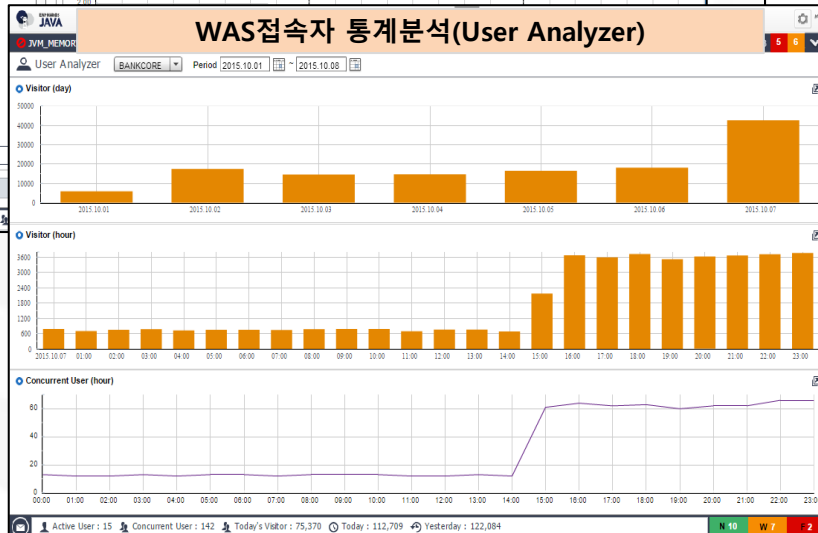
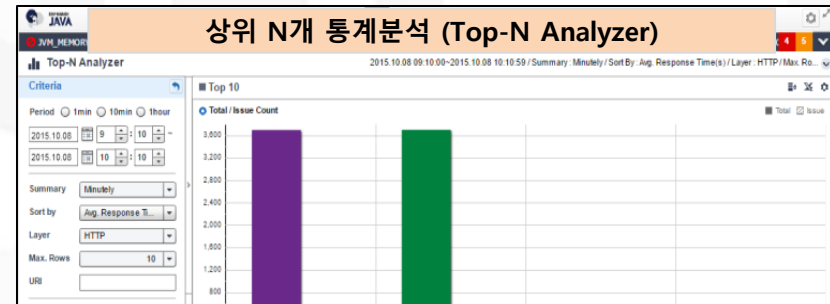
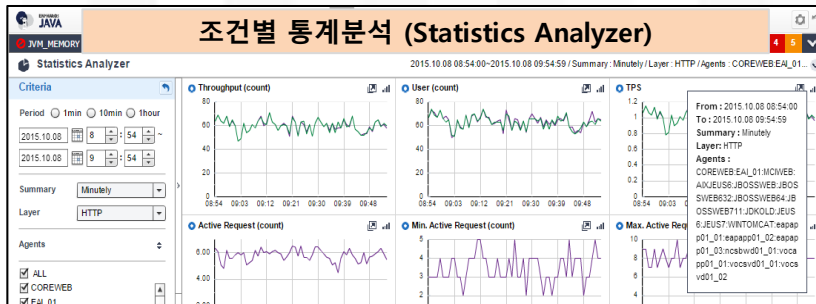
Explain **Apply**



5.5.1 통계기반 성능분석 개요

5.5 통계 기반 성능분석

ENPHAROS JAVA는 WAS에서 발생하는 성능관련 다양한 정보들을 실시간으로 수집되어 모니터링 제공하며 이때 성능관련 시계열성 정보들은 1분, 10분, 1시간과 같은 의미를 부여할 수 있는 시간 단위로 통계화 하여 저장하므로 각종 정보들의 추이분석에 유용합니다.



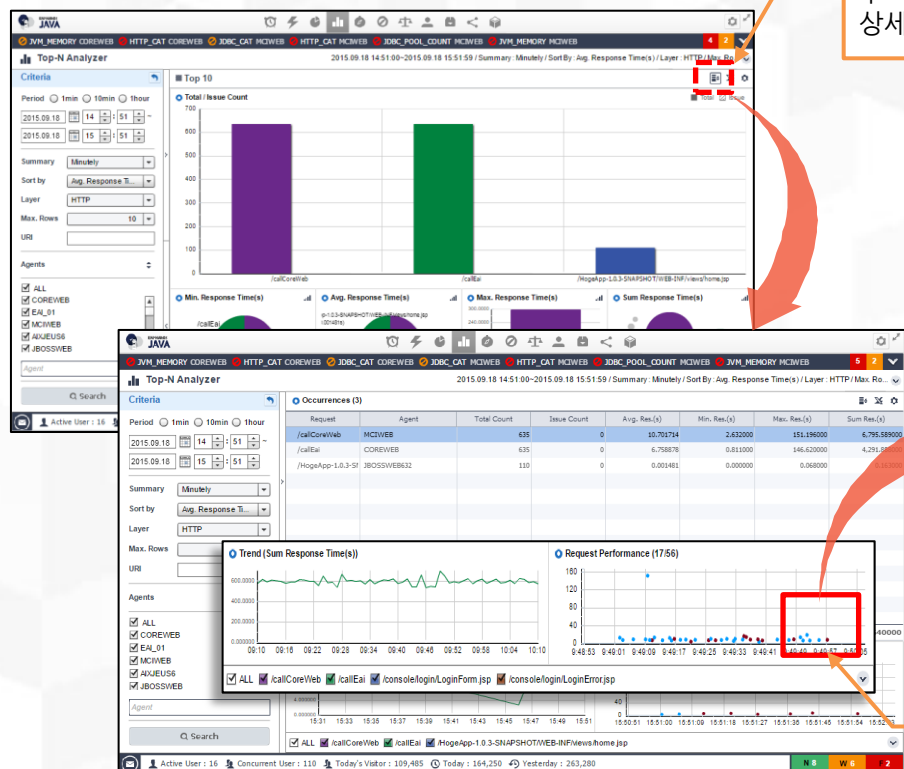
5.5.2 성능 통계용 지표

WAS의 성능을 시계열 통계화 하는 지표로는 Throughput, Active Request, 평균응답시간, 접속자수, CPU 사용율, 초당서비스수, 최대액티브요청건수, 메모리사용량 등 WAS1대를 기준으로 관리하며 검색시 선택한 시계열 구간에 대하여 추이그래프를 통해 표시합니다.



ENPHAROS JAVA의 통계분석 전략은 WAS에서 실행되는 많은 서비스 중에서 반드시 관심을 갖고 개선이 필요한 대상 서비스를 원스톱으로 찾아 제공하는 것으로서 이를 위해 중요도 상위 10개 (설정에 의거 적용함)를 자동으로 선정하여 제공합니다. 이 기능을 통하여 사용자는 성능저하 서비스, 과부하 서비스, 오류발생 서비스를 손쉽게 선별하여 개선하도록 합니다.

호출횟수, 평균응답시간, 이슈발생건수 기준 TOP 10으로 선정된 결과의 상세 리스트를 확인함

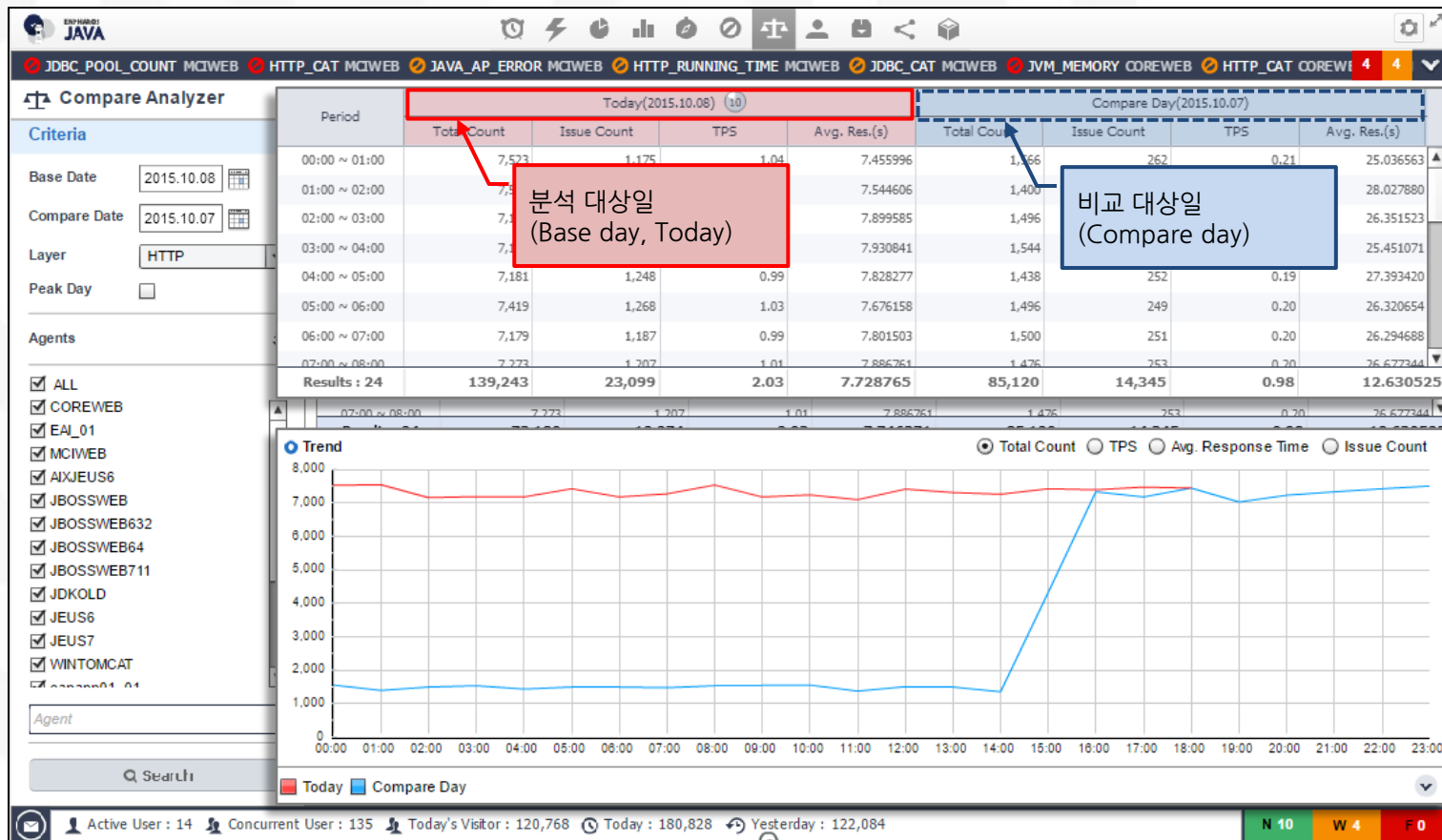


응답시간 기준의 퍼포먼스뷰를 통해
애플리케이션 성능분석 뷰로 원스톱
연계분석을 함

[illegible]

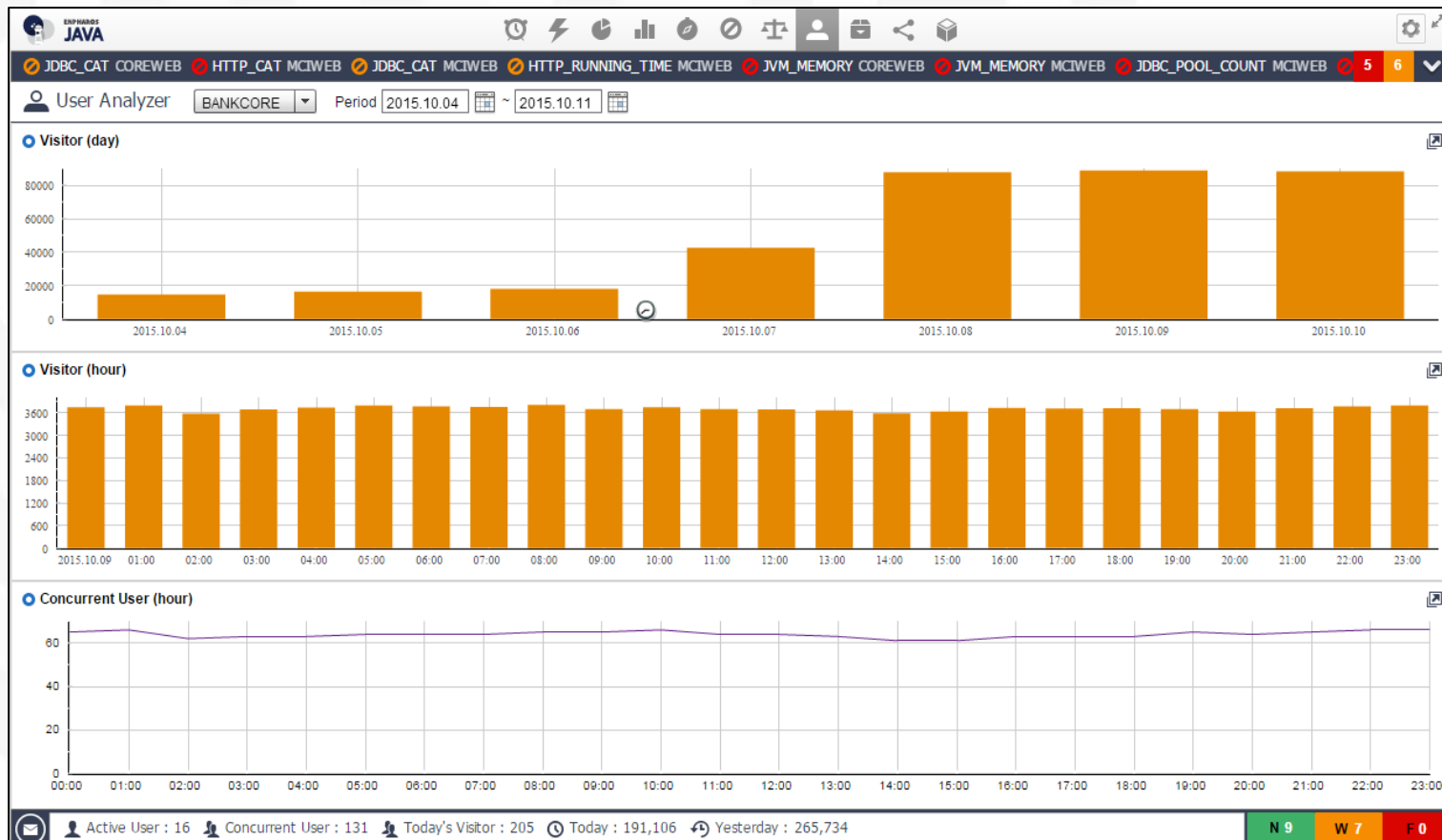
5.5.4 통계기반 비교분석

WAS에서 실시간으로 수집되는 시계열 성능 수치는 높고, 낮음이 발생하더라도 전체적인 Trends를 보기 위해 통계를 생성하여 모니터링하나 이 경우 현재의 통계치가 정상인지 비정상인지를 판단하는 것은 쉽지 않습니다. 따라서 ENPHAROS JAVA에서는 이 같은 통계값으로 부터 비교판단을 위하여 특정일자과 비교하는 상관분석을 실시하며 이때 Compare Analyzer가 사용됩니다.



5.5.5 사용자 통계분석

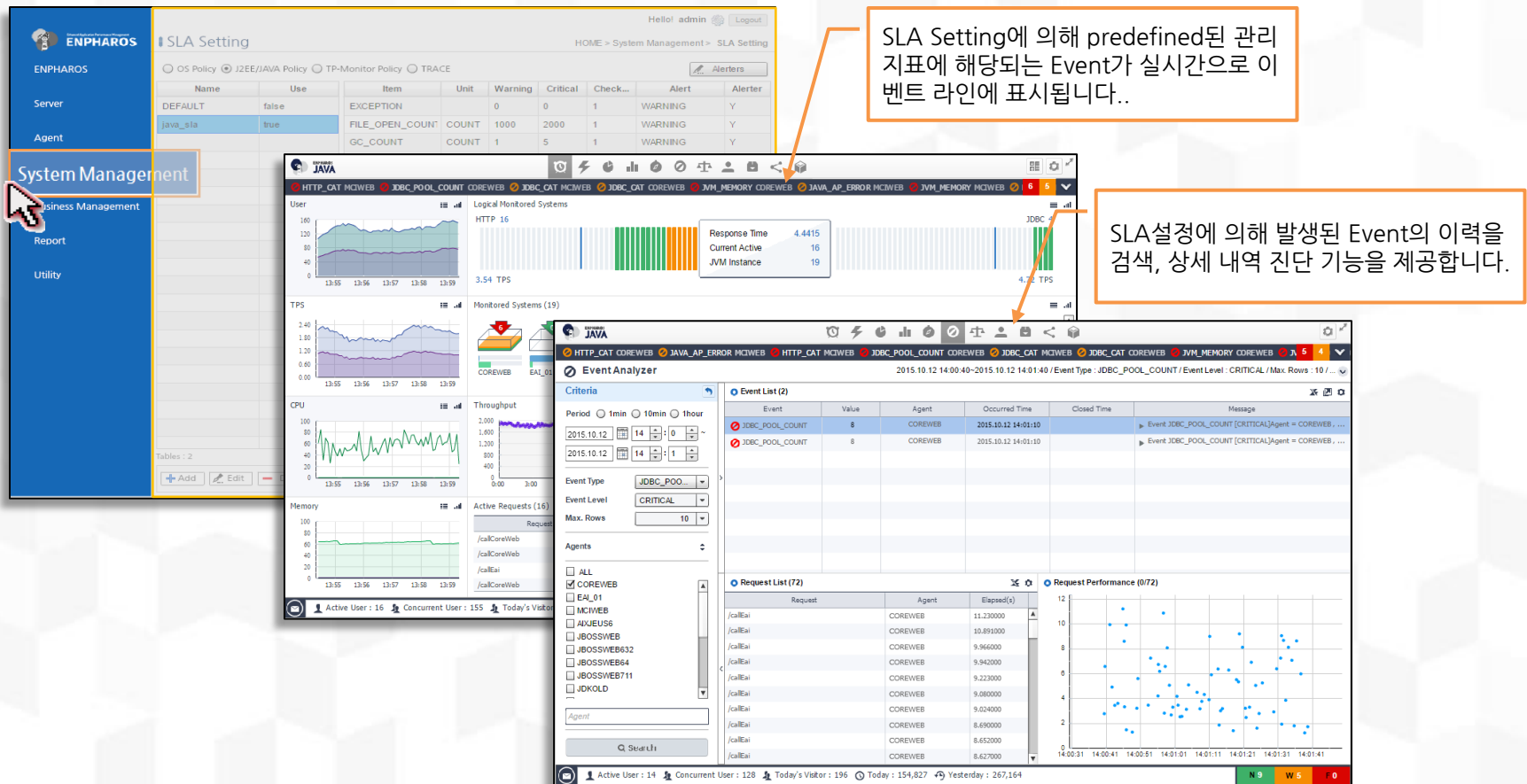
ENPHAROS JAVA는 WAS에 서비스를 요청하는 Client의 연결을 User(접속자)로 정의하여 관리합니다. 많은 동시 접속자수는 WAS에 비례적으로 Active Requests를 높이며 결과적으로 WAS서버의 CPU, Memory자원 고갈을 불러오므로 User수는 WAS의 성능과 장애에 높은 상관성을 갖는 성능지표가 됩니다. 따라서 Concurrent User수의 관리와 더불어 일당, 시간당 방문자수를 통해 Trend를 관리합니다.



5.6.1 장애관리 개요

5.6 장애 관리

ENPHAROS JAVA는 WAS에서 자주 발생하는 장애 원인 항목을 미리 정의하여(Predefined) 임계치를 설정하도록 합니다. 이를 통해 서비스수준을 관리하도록 하며 SLA setting값에 부합되는 상황 발생시 Event를 발생시켜 화면에 송출하고 각종 외부 전파 장치를 통해 상황을 전파합니다. 또한 발생한 Event는 Event Analyzer 화면을 통해 이력을 진단할 수 있도록 합니다.



5.6.2 장애이벤트 설정관리

5.6 장애 관리

ENPHAROS JAVA에서 설정하는 장애 이벤트 관리항목은 크게 JVM 자체의 관리항목, ENPHAROS JAVA Agent의 생사감시 항목, WAS에서 처리되는 서비스의 성능과 관련된 항목으로 구분합니다. 각 항목은 관리용 UI화면을 통해 수정, 삭제, 추가를 할 수 있으며 SLA 항목에 대한 임계 값 도달 시 전파 방법도 설정에 의거 관리합니다.

ENPHAROS

SLA Setting

☐ OS Policy ☒ J2EE/JAVA Policy ☐ TP-Monitor Policy ☐ TRACE

Alerters

Name	Use	Item	Unit	Warning	Critical	Check...	Alert	Alserter
DEFAULT	false	EXCEPTION		0	0	1	WARNING	Y
java_sla	true	FILE_OPEN_COUNT	COUNT	1000	2000	1	WARNING	Y
		GC_COUNT	COUNT	1	5	1	WARNING	Y
		GC_TIME	μs	5000	10000	1	WARNING	Y
		HTTP_CAT	COUNT	1	5	1	WARNING	Y
		HTTP_RUNNING_TIME	μs	5000000	10000000	1	WARNING	Y
		INSTANCE_LIVE		0	0	1	CRITICAL	Y
		JAVA_API_ERROR		0	0	1	WARNING	Y
		JDBC_CAT	COUNT	1	6	1	WARNING	Y
		JDBC_FETCH_COUNT	COUNT	100	170	1	WARNING	Y
		JDBC_POOL_COUNT	COUNT	3	5	1	WARNING	Y
		JDBC_RUNNING_TIME	μs	5000000	10000000	1	WARNING	Y
		JVM_CPU	%	5	10	1	WARNING	Y
		JVM_MEMORY	MB	10	100	1	WARNING	Y
		JVM_MEMORY_RATIO	%	70	90	1	WARNING	Y
		LEAK		0	0	1	NONE	N
		SOCKET_OPEN_COUNT	COUNT	2	10	1	WARNING	Y
		THREAD_CPU	%	5	10	1	WARNING	Y
		TP_POOL_COUNT	COUNT	50	100	1	NONE	N

Tables : 2

Tables : 19

+ Add Edit Delete Edit

Save Apply

SLA로 관리하는 대표적인 Predefined Item으로서 수집되는 정보 중에서 임계치로 관리 가치가 높은 항목을 선별하여 관리합니다.



5.6.3 장애 이벤트 분석

5.6 장애 관리

ENPAHROS JAVA는 Event가 발생시 상단 이벤트라인에 발생 즉시 발생한 이벤트의 심각도에 따라 붉은색, 황색 등으로 구분하여 전파합니다. 상단 이벤트라인의 이벤트 메시지를 클릭시 상세한 이벤트 List를 제공하여 장애에 대한 대처를 유도합니다.

The screenshot displays the ENPAHROS JAVA Event Analyzer interface. The top bar shows the application name and various status indicators. The main area is divided into several sections:

- Criteria:** Includes filters for Period (1min, 10min, 1hour), Event Type (JDBC_RUN...), Event Level (CRITICAL), and Max. Rows (10).
- Event List (4):** A table showing event details. The first row is highlighted in blue.
- Request List (33):** A table showing request details. The first row is highlighted in blue.
- Request Performance (0/33):** A scatter plot showing request performance over time.

An orange callout box points to the Event List (4) table, stating: "특정기간 동안 선택한 Event가 선택된 Level(Critical, Warning...)로 발생한 Event의 이력을 제공합니다." (Provides the history of events that occurred at the selected level (Critical, Warning...) during the specified period.)

Event	Value	Agent	Occurred Time	Closed Time	Message
JDBC_RUNNING_TIME	20.582000	COREWEB	2015.09.21 18:14:50		Event JDBC_RUNNING_TIME [CRITICAL] (20582000 / 1000000...
JDBC_RUNNING_TIME	20.582000	COREWEB	2015.09.21 18:14:50		Event JDBC_RUNNING_TIME [CRITICAL] (20582000 / 1000000...
JDBC_RUNNING_TIME	10.579000	COREWEB	2015.09.21 18:14:40	2015.09.21 18:12:09	Event JDBC_RUNNING_TIME [CRITICAL] (10579000 / 1000000...
JDBC_RUNNING_TIME	10.579000	COREWEB	2015.09.21 18:14:40	2015.09.21 18:12:14	Event JDBC_RUNNING_TIME [CRITICAL] (10579000 / 1000000...

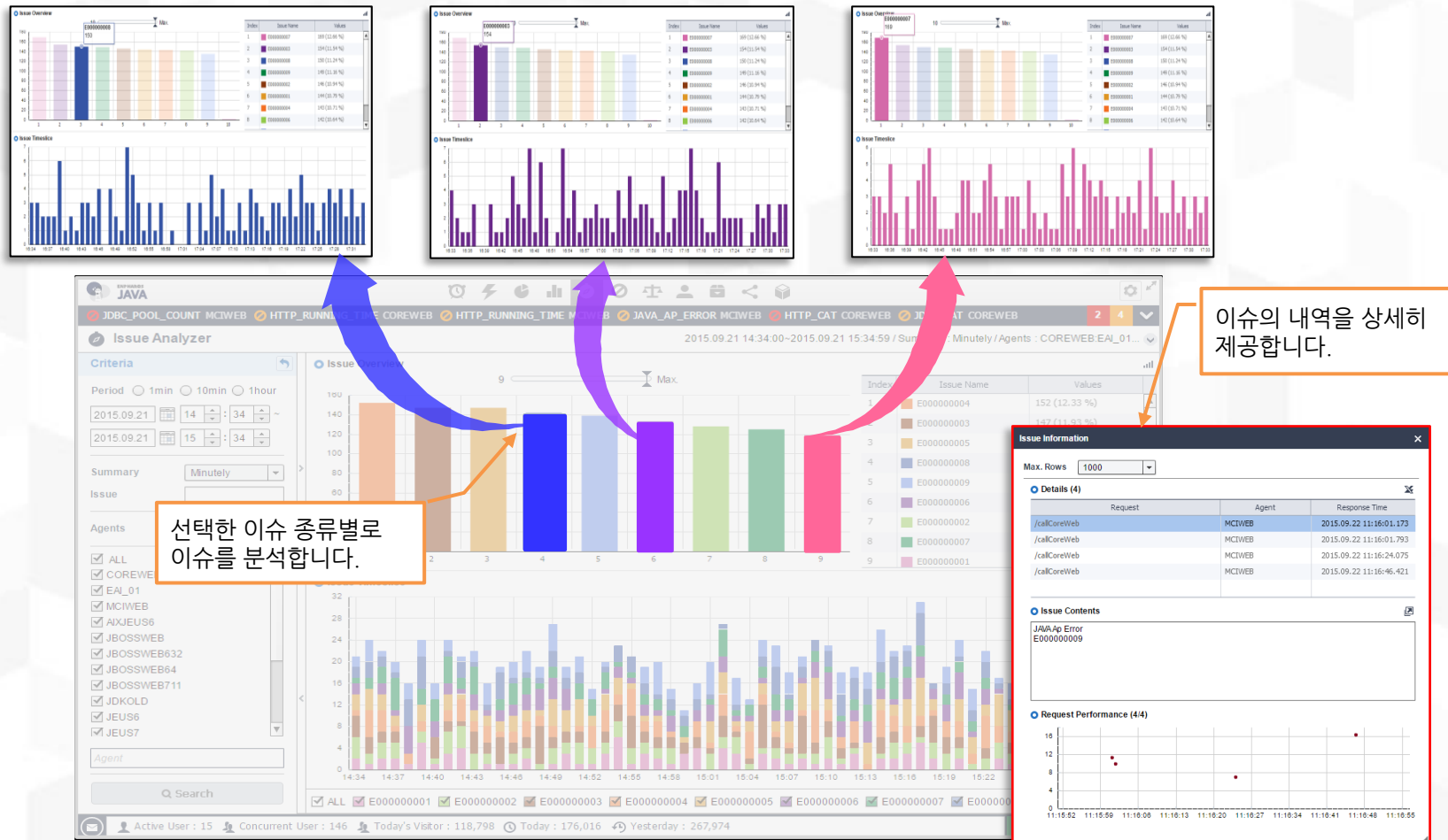
Request	Agent	Elapsed(s)
/callEai	COREWEB	9.878000
/callEai	COREWEB	9.351000
/callEai	COREWEB	9.303000
/callEai	COREWEB	9.182000
/callEai	COREWEB	8.825000
/callEai	COREWEB	8.545000
/callEai	COREWEB	7.624000
/callEai	COREWEB	7.559000
/callEai	COREWEB	6.897000



5.6.4 장애내역 통계분석

5.6 장애 관리

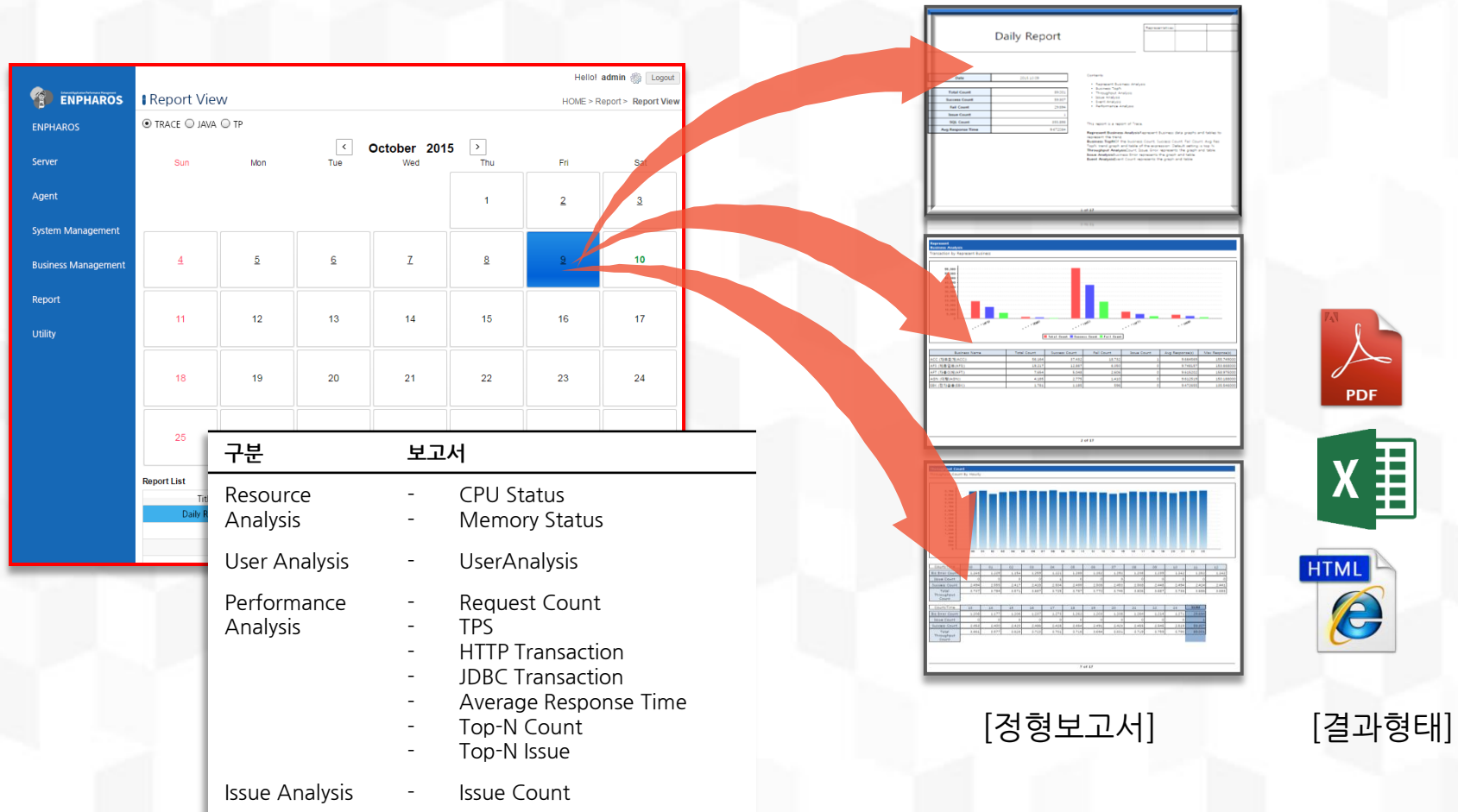
성능 데이터를 기반으로 통계 축적에 의해 성능을 분석하는 방식과 유사한 방법으로 WAS에서 발생한 장애, 오류, exception 등 성능을 저하시키는 장애 관련 정보들을 통계 축적한 정보를 최소의 navigation으로 찾을 수 있도록 가독성이 높은 UI를 통해 제공하여 ENPHAROS JAVA는 이미 발생한 장애를 효과적으로 찾고 다시 재발되지 않도록 합니다.



5.7.1 성능보고서 개요

5.7 성능 보고서

ENPHAROS JAVA는 수집 저장된 성능정보중에 IT부서의 업무 성격에 맞는 정형화된 성능진단 보고서를 제공하며 이를 Calendar 설정을 통하여 주기적으로 자동 생성이 되도록 스케줄관리 기능을 제공하며 생성된 결과물의 형태로는 HTML, EXCEL, PDF등을 통해 선택 사용이 가능합니다. 또한 사용자 정의형 보고서도 기술지원에 의거 간단히 제공이 가능합니다.



5.8 자체보안기능

ENPHAROS JAVA는 접속한 사용자를 식별하고 사용자의 권한별 관리를 위하여 자체 접속 ID와 PASSWORD를 관리 하며 접속자 별 부여된 권한에 따라 볼수 있는 화면과 없는 화면을 구분 관리할 수 있는 접근제어기능을 제공합니다.



에이전트 접근 제한

사용자 별 접근 가능한 에이전트 지정
로그인시 권한이 없는 에이전트로의 접근 통제



역할별 제어 권한 부여

제어 기능 : SQL 수행, SQL 상세정보 보기
역할 별, 사용자 별 권한 부여



비밀번호 암호화

강력한 암호화 알고리즘 사용: AES-128, SHA-256
단순 패턴의 비밀번호 설정 방지



비밀번호 기한 설정

역할별 비밀번호 변경 기한 설정
미 변경 시 접속 제한



비밀번호 인증 실패제한

지정된 횟수 이상의 비밀번호 인증 실패 시 사용자 잠금
Admin만이 잠금 해제



6. 기대효과

ENPHAROS JAVA에서 제공되는 성능 모니터링 기능은 업무시스템의 개발 및 운영 시에 활용됨으로써 도입 전 대비 높은 투자대비 효과가 다수의 사례에서 입증되고 있으며 기존 H/W의 증설로 대응하던 많은 IT시스템들이 합리적으로 적절한 투자규모를 제시하게 되어 높은 ROI를 나타내고 있습니다.



서비스 가치 실현

- 실시간 애플리케이션 서비스 관리를 통한 안정적인 서비스 가치 실현

고가용성 IT 서비스 구축

- 애플리케이션 상시 관제를 통한 장애 예방으로 시스템 가용성 극대화

개발코드의 품질 제고

- 애플리케이션 개발 시 품질 측정의 도구로 사용함으로써 품질 개선 효과

장애 분석 및 예방

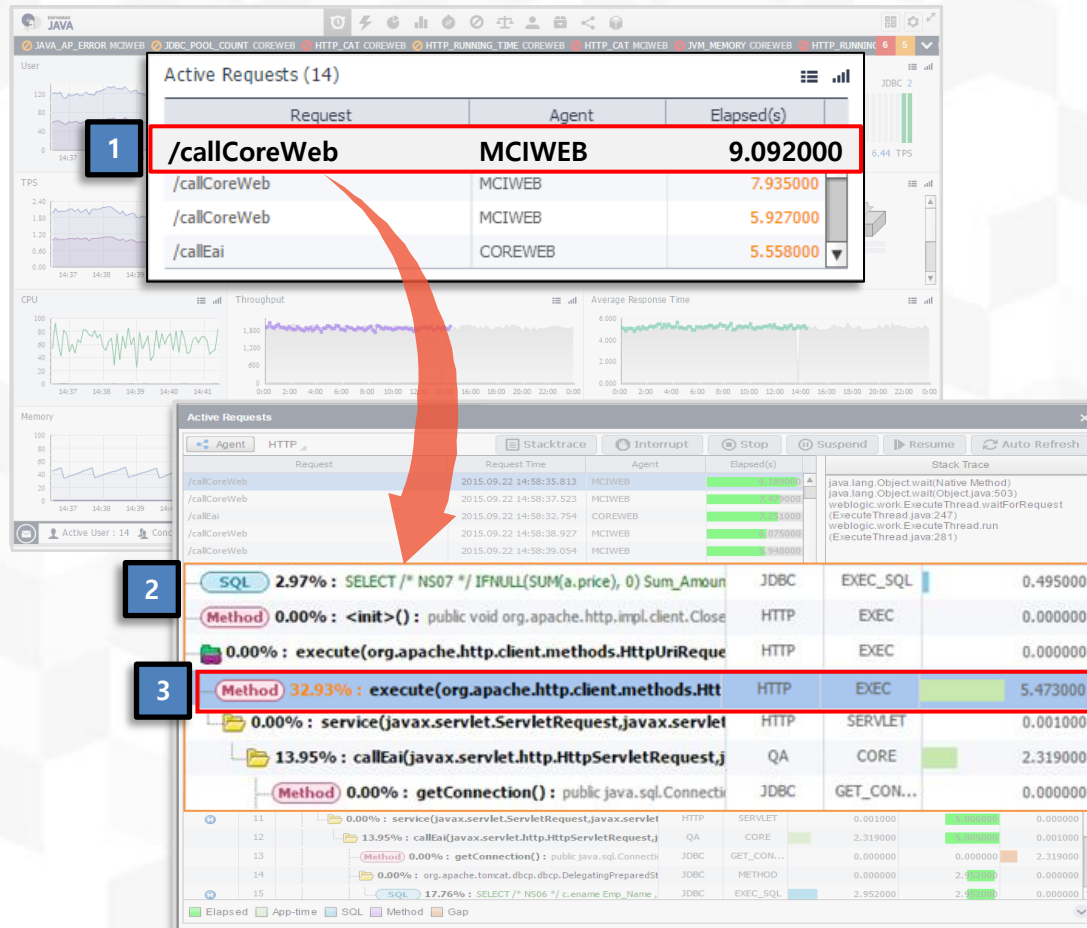
- 실시간 모니터링, 통계 분석을 통한 신속한 장애 분석과 사전 예방

IT 자원의 합리적 투자

- 정기적 튜닝 개선을 통한 자원 사용량 절감, IT 자원 증설의 합리화

7.1 실행중인 지연 서비스 진단분석

WAS로 업무를 서비스할 경우 가장 일반적으로 겪게 되는 현상은 사용자들이 서비스가 느다는 불평이며 이 경우 WAS관리자 또는 IT부서는 신속히 어떤 프로그램 때문에 서비스가 느린지 확인하는 일입니다. 이 같은 지연 거래 (Long Transaction)가 발생시 ENPHAROS를 통해 다음과 같이 손쉽게 isolation과 tuning을 할 수 있습니다.



1 'Active Request'에서는 현재 수행되고 있는 모든 거래에 대하여 경과 시간에 따라 List-up된 서비스를 제공하며 특정한 하나의 서비스를 선택함.

2 선택한 지연 서비스의 현재시간까지 수행되었던 Call-tree 데이터 모두가 표현이 되고 현재 진행 중인 로직이 표현되며 수행을 위해 참조한 메타 정보나 기타 파라미터 정보가 제공됨.

3 Call-tree의 단계 중에서 전체 수행시간 중에서 가장 비중이 높은(32.93%) 단계의 step을 적시하므로 신속히 튜닝을 할 수 있음.



7.2 수행완료 서비스의 성능 튜닝

7. 활용 사례소개

WAS 상에서 처리된 업무서비스는 다양한 조건으로 검색을 통해 튜닝 개선이 필요한 서비스를 선별할 수 있도록 Analyzer(분석도구)를 제공하며 이를 통하여 개선 대상 서비스를 선별하며 원스텝 클릭을 통해 Back end의 Query문 튜닝까지를 할 수 있습니다.

The screenshot displays the Request Analyzer tool interface. On the left, the 'Criteria' panel shows search filters for 'Period' (1min, 10min, 1hour) and 'Agents' (ALL, COREWEB, EAL_01, MCIWEB, AIXEUS6, JBOSSWEB, JBOSSWEB632). A red arrow points from the 'Search' button to the 'Occurrences' table. The 'Occurrences' table lists requests with columns: Request, Agent, Layer, Elapsed(s), Request Time, Response Time, Remote IP, and Misc. A red box highlights a specific request. A red arrow points from this request to the 'Signature' table. The 'Signature' table shows the call tree for the selected request, with columns: Info, No., Signature, Layer, and Type. A red box highlights a specific SQL query in the call tree. A red arrow points from this SQL query to the 'Processing Time' table. The 'Processing Time' table shows the execution time for the selected SQL query, with columns: Info, No., Signature, Layer, and Type. A red arrow points from this table to the 'Meta Information' panel, which displays details about the selected SQL query, including the query text and execution time.

1 Search

2 Occurrences (2,079)

3 Signature

1 튜닝 대상을 선별하기 위해 기간을 포함 검색 조건을 입력한 후 Search 버튼 클릭을 통해 선별함.

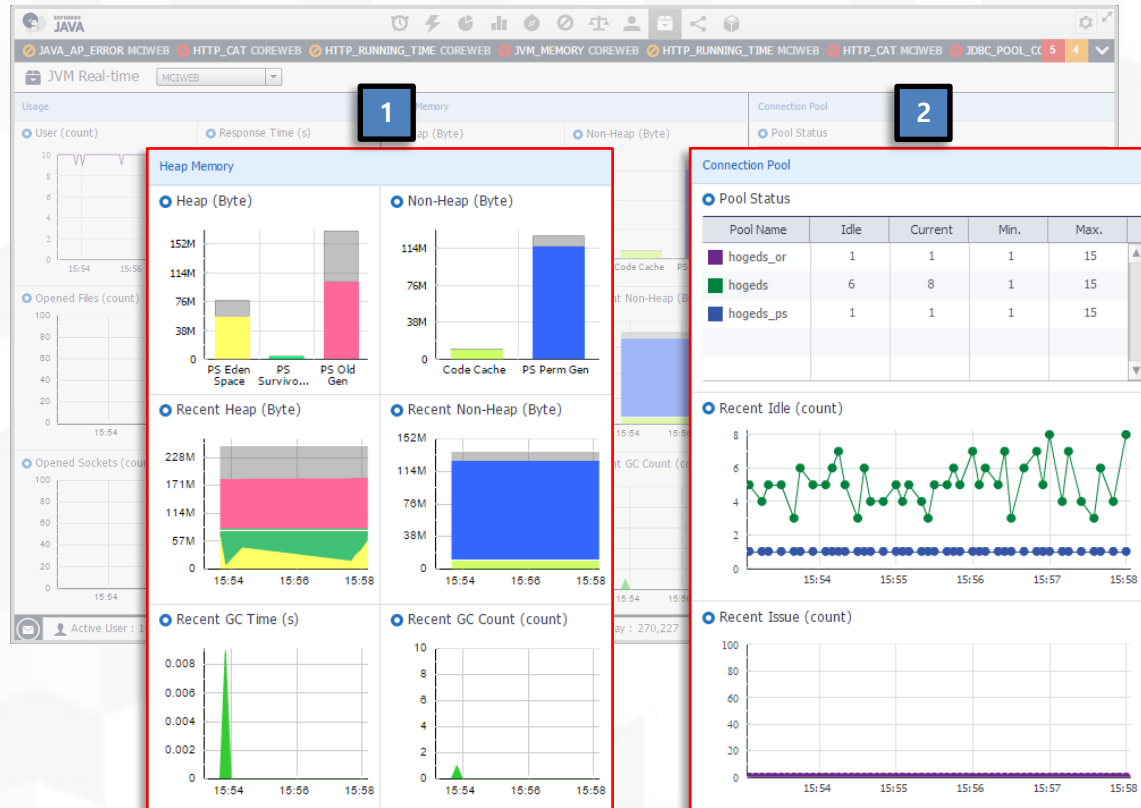
2 특정된 기간내 선별된 서비스 중에서 개선이 필요한 서비스 하나를 선택할 수 있으며 선택한 서비스를 클릭만으로 내부의 Call-tree(호출관계)를 확인하며 어느 step에서 많은 시간이 소요되었는지 확인함.

3 특정 step(예시:특정 SQL수행)을 선택시 선택한 SQL Query가 수행시 수행된 각종 메타정보를 제공하므로 성능 튜닝 함.



7.3 WAS내부 Resource 사용 진단

기업용 WAS 솔루션은 JAVA 프로그램을 운영하기 위한 핵심 인프라 솔루션으로서 업무 서비스의 가용성을 제고를 위해서 필수적인 관리 대상이며 WAS관리자에게 JVM이 사용하는 자원(메모리, CPU, Thread등등)의 사용량 관리는 안정적 서비스를 위해 반드시 관리하는 항목이며 ENPHAROS를 통해 관리자는 WAS 핵심관리 항목을 원시스템으로 진단이 가능합니다.



JVM의 힙메모리(Heap memory) 영역을 내부 구조별로 세분하여 설정값, 사용량, 여유공간, GC발생 등으로 제공하며 추이를 제공하므로 현재 값의 이상여부 판단을 할 수 있습니다.

JVM에서 Data Source와 연결을 위해 미리 연결한 connection pool의 종류와 pool에서 사용하는 connection 개수, 여유 개수 등을 제공하므로 자주 발생하는 connection pool의 제반 문제를 one click으로 확인할 수 있습니다.

감사합니다.

Q&A