



# 성능관리 솔루션 제니퍼(JENNIFER)v4.5

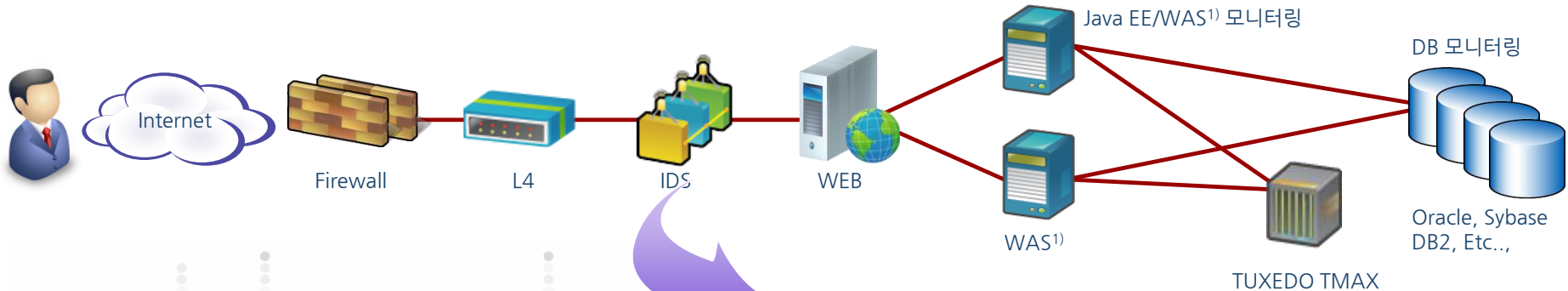
JENNIFER

Performance Management for JAVA EE Platform

# 제니퍼 모니터링 포지셔닝

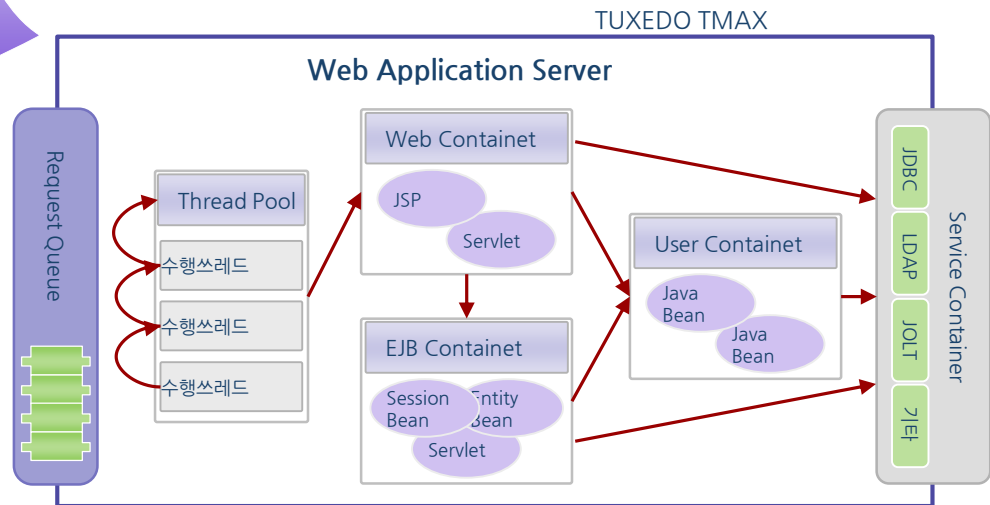


애플리케이션성능관리(APM)는 조직과 보유 시스템의 현실에 맞는 적절한 투자와 가용성 측면을 고려하여, 효과적인 성능 모니터링 및 장애대응 전략을 수립하고 성능관리 체계를 구축하는 것입니다. 애플리케이션성능관리(APM)는 전통적인 시스템관리 솔루션(SMS,NMS)과 달리, 실제 서비스되고 있는 시스템의 서비스 관점에서의 성능적 현황과 내부적 애플리케이션 관점에서의 성능 장애 대응 및 분석 역량을 강화시켜, 보다 지능적인 방법으로 대 고객 서비스의 안정화를 이루어 궁극적으로 총 소유비용(TCO)을 효과적으로 낮출 수 있습니다. 오늘 날의 복잡한 엔터프라이즈 환경에서, End-to-end로 모든 것을 모니터링 하겠다는 것은 자칫 고가의 솔루션 도입에 따른 불필요한 추가 개발용역 비용과 예상치 못한 관리 비용의 증가를 가져올 수 있습니다. 또한 각각 전담 영역별 조직의 전문성 결여를 야기할 수 있기 때문에, 투자 대비 효과 측면에서 신중하게 고려되어야 합니다.



**JENNIFER** 는 "APM+Extension" 입니다

- WAS<sup>1)</sup> 중심적인 특화된 실시간 통합 서비스 모니터링
- 즉시적 성능 장애 진단 및 장애 대응
- 애플리케이션 관점에서의 트랜잭션 추적 및 튜닝
- 외부 트랜잭션 인터페이스 추적 및 확장 모니터링



# 제니퍼 지원 플랫폼



제니퍼는 현재 IT환경 내에서 운영중인 대부분의 OS(Operating System), JAVA(Sun, IBM, BEA, etc), WAS(Web Application Server)를 지원하고 있습니다. 특히 새로운 OS, JAVA, WAS 버전 릴리즈시 24시간 이내에 제품호환성 검증작업을 수행할 수 있는 프로세스 및 지원조직을 보유함에 따라 새로운 시스템 도입 및 변경시 안정적인 관리업무를 수행 할 수 있도록 합니다.

## 운영 시스템(OS)

- AIX 4.3.3, 5.x 32bit, 64bit
- HP-UX 11.x 32bit, 64bit, Itanium 64bit
- Sun Solaris 2.8, 2.9, 10 32bit, 64bit, x86
- Intel Linux 32bit, Redhat Itanium 64bit
- Compaq Tru64 UNIX OSF1
- Microsoft Windows 2000, XP, 2003, Vista
- IBM iSeries(AS400) for WebSphere
- IBM z/OS for WebSphere, zLinux

## 웹 애플리케이션 서버(WAS)

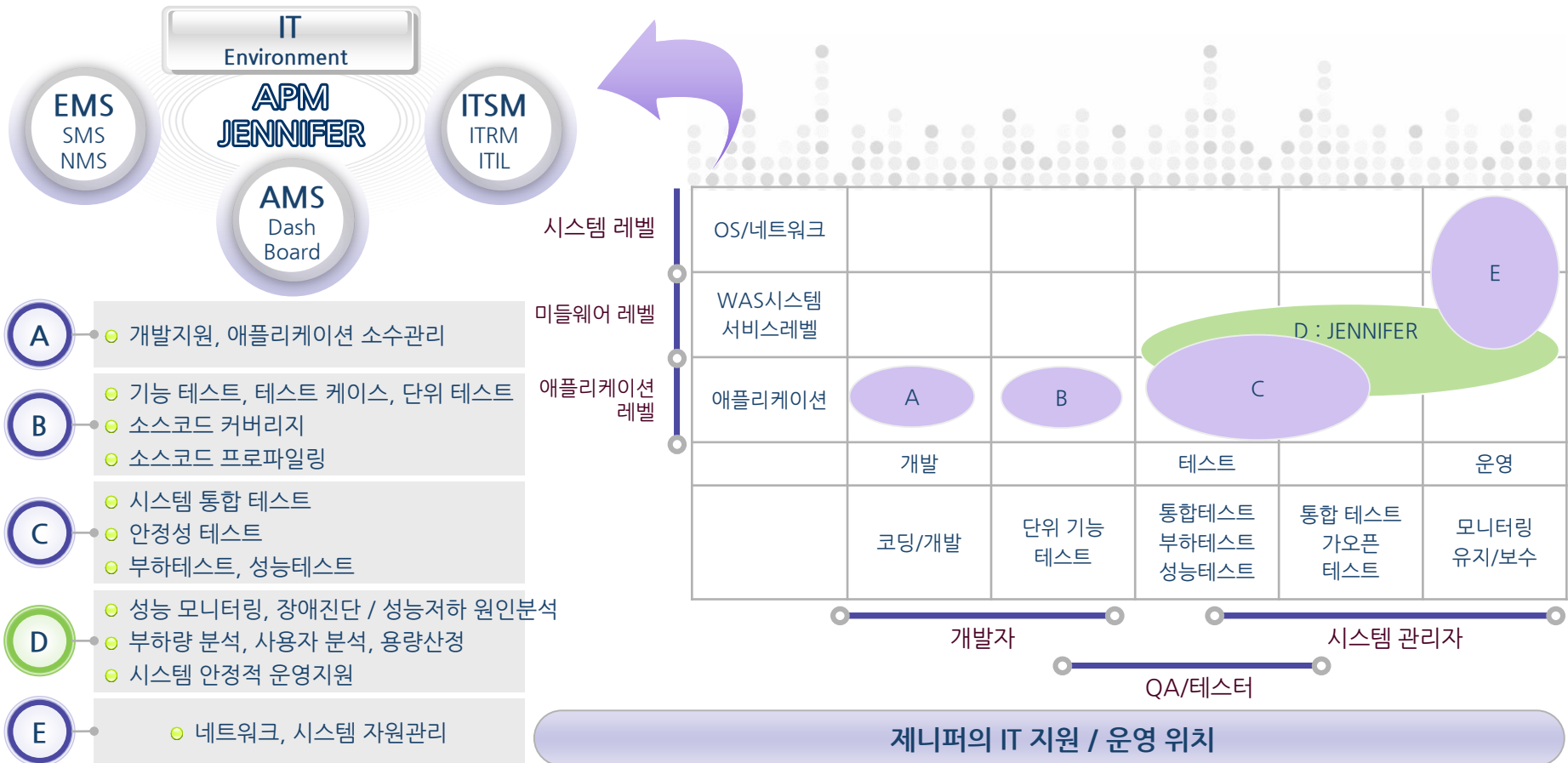
- BEA WebLogic 5.1, 6.x, 8.x, 9.x, 10.x
- IBM WebSphere Application Server 3.5, 4.x, 5.x, 6.x
- Tmaxsoft JEUS 3.x, 4.x, 5.x, 6.x
- Oracle Application Server 9iAS, 10gAS, OC4J, ERP
- SUN Application Server 7.x, 8.x, 9.x
- Fujitsu Interstage 5.x, 6.x, 7.x
- Hitachi Cosminexus 7
- Sybase EAServer 4.x, 5.x
- Macromedia JRun 4.x
- Apache Jakarta Tomcat 3.x, 4.x, 5.x
- Caucho Technology Resin 2.x, 3.x
- RedHat JBoss Application Server 3.x, 4.x
- Apache Jserv
- GlassFish



# 제니퍼 서비스 포지션



시스템 개발 및 운영 단계별로 테스트 및 관리되어야 할 중점 사항은 매우 다양합니다. 제니퍼는 시스템 개발 이후 통합 성능테스트 시점부터, 시스템 오픈 이후 성능 모니터링 및 장기적인 성능관리 운영 단계까지 효과적으로 사용되어 집니다. 타회사의 솔루션 제품 군은 앞 단의 소스 코드/개발 관리 부분만을 지원하거나, 혹은 뒷 단의 시스템 및 네트워크를 모니터링 하는 전통적인 SMS, NMS 제품들이었습니다. 그러나, 막상 운영단계에서 애플리케이션의 성능관리를 제공해 주지는 않았습니다. 제니퍼는 서비스 운영 단계에서의 애플리케이션 서비스 중심적인 성능관리를 제공하는 솔루션입니다.



# 제니퍼의 구조



## 1 JENNIFER Agent

- 모니터링 대상 시스템(WAS)에 설치되며, 각종 성능 정보를 수집, 제니퍼서버로 전달하는 역할을 합니다.

## 2 JENNIFER Server

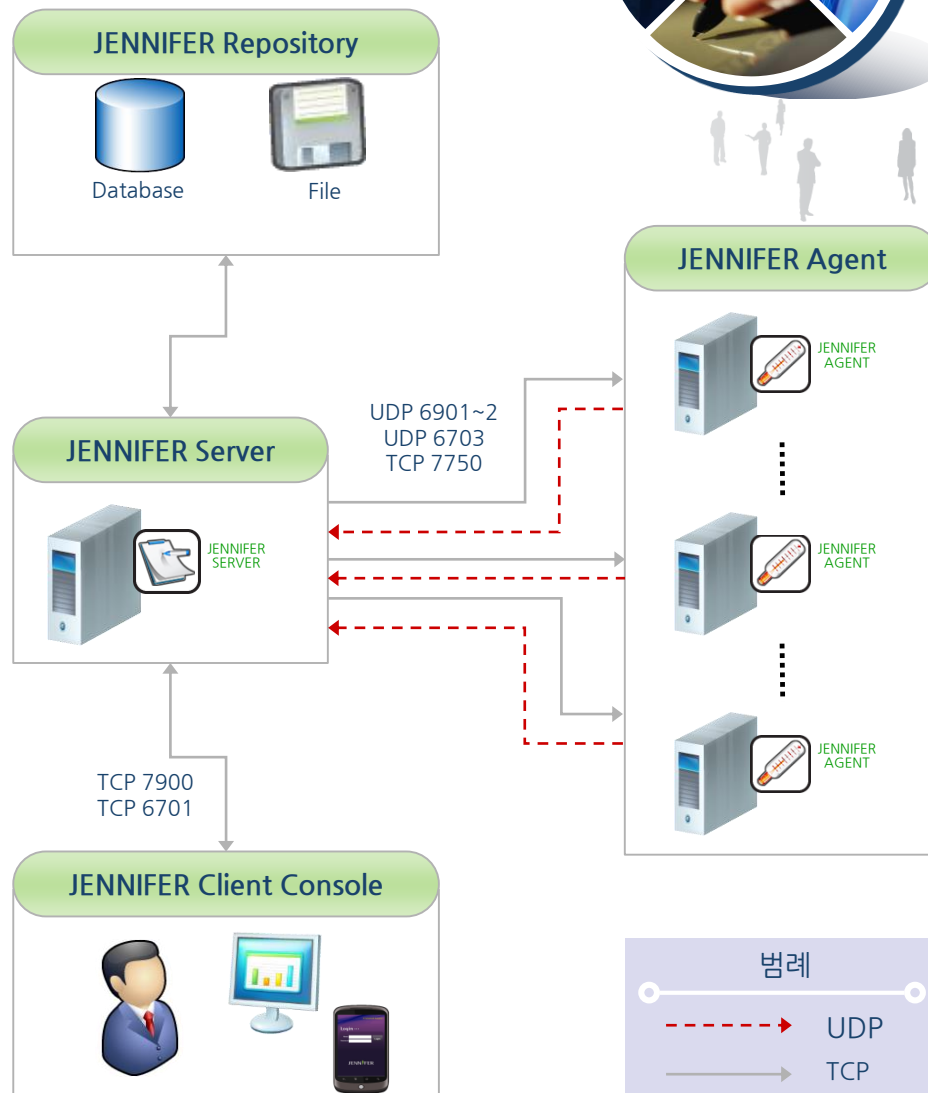
- 제니퍼 에이전트로부터 실시간 모니터링 정보를 수집, 가공하며 실시간 대시보드 제공 및 통계정보관리, 장애/에러 정보 등 관리에 필요한 각종 기능을 수행합니다.

## 3 JENNIFER Repository

- 제니퍼 서버에서 가공된 데이터는 데이터 형태에 따라 DB, File시스템에 저장되며 이러한 데이터를 통해 시스템 분석에 필요한 각종 데이터를 확보할 수 있습니다.

## 4 JENNIFER Client Console

- 웹 브라우저를 통해 장소의 제약 없이 시스템 모니터링 및 제니퍼 관리를 수행 할 수 있습니다.
- 구글 안드로이드, 애플 iOS 기반의 스마트 폰을 통한 성능진단을 수행할 수 있습니다.





# 제니퍼의 핵심기능



실시간(Real-Time) 종합 성능 모니터링

도메인(Domain) 구성기능

서비스 폭주 시 부하 량 제어

정량화된 데이터를 이용한  
용량산정/추이분석

성능장애에 대한 자동경보 및  
SMS 연동기능

인사이트 뷰(Insight View)

스마트폰 지원 (JENNIFER Mobile)

Business Monitoring 기능

Batch Job 모니터링 기능

수행중인 액티브 서비스 모니터링

응답시간 분포 그래프(X-View)

SQL/Application/백엔드  
시스템간의 상호연관관계 분석

사용자 대시보드 화면 구성

정형화된 보고서 및 유연한  
사용자 정의형 보고서

다이나믹 프로파일(Dynamic Profile)

다이나믹 스택트레이스  
(Dynamic Stacktrace)

웹 기반 WYSIWYG 리포팅 지원 기능

제니퍼의 확장기능  
(Remon, Extra Agent, Log Watcher)

JENNIFER

# 제니퍼 통합 대시보드



제니퍼 대시보드는 Java EE 시스템 운영 상태 모니터링을 위해 필수적으로 필요한 각종 데이터를 서로 유기적으로 배치하여 제공하고 있습니다. 이렇게 유기적으로 배치된(시스템 성능, 시스템 자원 리소스 사용상태, 장애/에러발생, 사용자 시스템 사용상태) Java EE 시스템 관리자들이 시스템 상태에 대한 빠른 판단 및 효과적으로 제어할 수 있도록 돕고 있습니다.



## 1 시스템 사용자 정보

- 동시단말사용자수
- 방문자수
- 액티브 사용자수

## 2 시스템 처리상태

- 초당처리건수
- 시간당 호출건수
- 평균응답시간

## 3 자원상태

- 시스템 CPU 사용량
- JVM CPU 사용량
- 힙 메모리 사용량

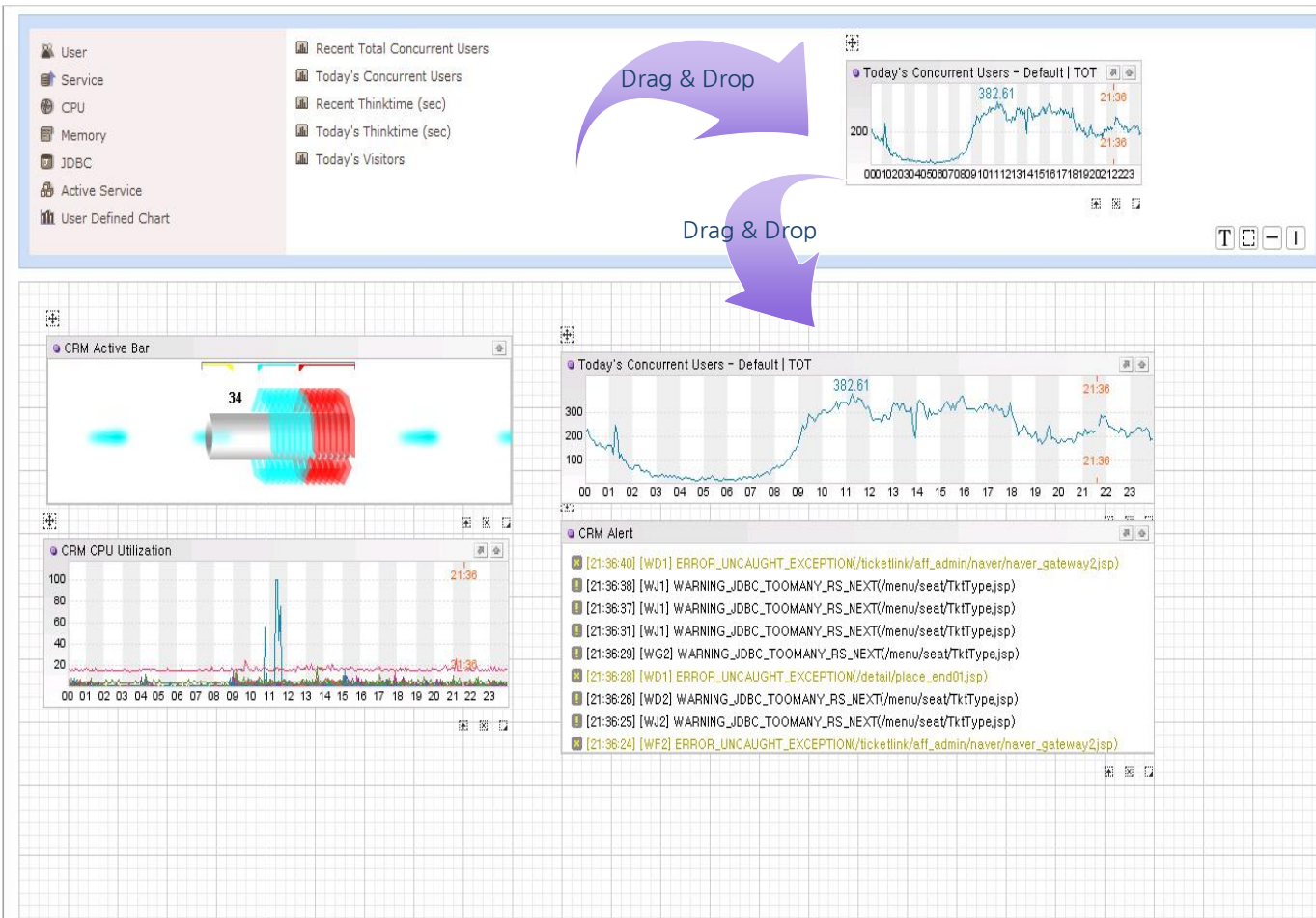
## 4 장애/에러

- 시스템 장애
- 프로그램 에러

# 사용자 정의형 통합 대시보드



제니퍼의 사용자정의화면 구성 기능은 제니퍼에서 제공되는 성능데이터 그래프와 Drag&Drop 기능을 이용하여 사용자가 자신의 환경에 맞는 모니터링을 화면을 구성할 수 있도록 하고 있습니다. 제공되는 성능 모니터링 그래프 종류로는 제니퍼 기본성능데이터 그래프와 사용자정의형 성능데이터 그래프를 제외하고 있습니다.



## ○ 제공되는 그래프 종류

- CPU
- Node
- XViewC
- Alert
- LINE
- STACKED LINE
- EQUALIZER
- STACKED
- EQUALIZER
- HORIZONTAL BAR
- METER
- ON/OFF CHECK
- TABLE
- NUMBER
- 추가개발 중



# 사용자 정의형 통합 대시보드 - 예제



## ◀ 사용자정의화면 구성 예제 - 1

- 프로세스 정상 체크 유무 및 디스크 사용량 거래건수 확인 등 WAS 시스템과 WAS 시스템 외부의 리소스 및 비즈니스 데이터 사용량을 사용자 정의형 대시보드로 구성하여 모니터링 화면을 구성했습니다. 커스터마이징 된 SMS 기능은 문제 발생시 관리자에게 전파 되도록 구성되었습니다.



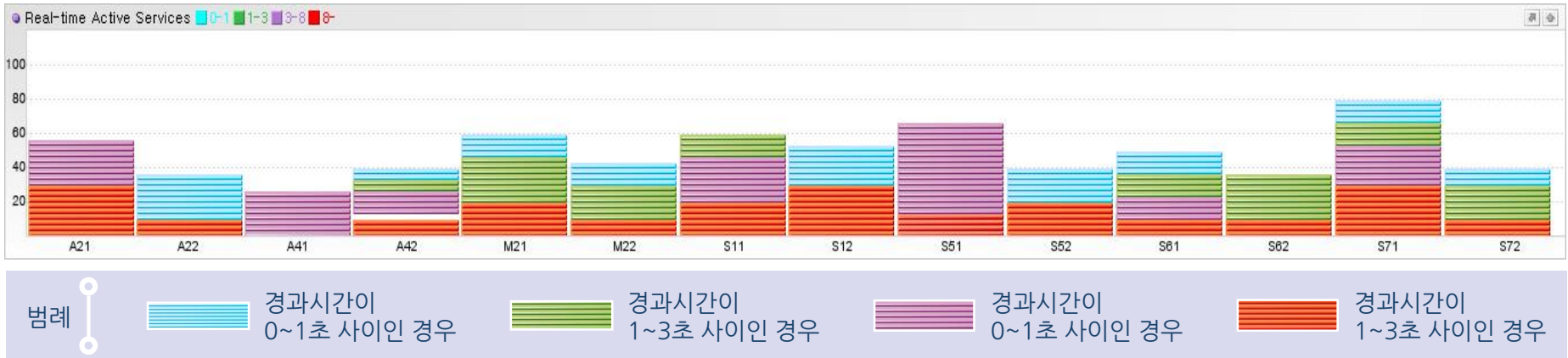
## ◀ 사용자정의화면 구성 예제 - 2

- UNIX머신 4대 각각에 대한 시스템 레벨 모니터링 데이터를 사용자 정의형 성능데이터 그래프를 활용하여 구성한 사용자정의형 통합 대시보드입니다. CPU 사용량, CPU 런큐(runqueue)상태, 네트워크 사용상태, 파일 시스템 별 디스크 사용량 정보를 한 눈에 볼 수 있도록 되었으며, 각 데이터에 따라 관리자가 지정한 임계치에 도달할 경우 SMS기능을 통해 전달받을 수 있도록 커스터마이징 되었습니다.

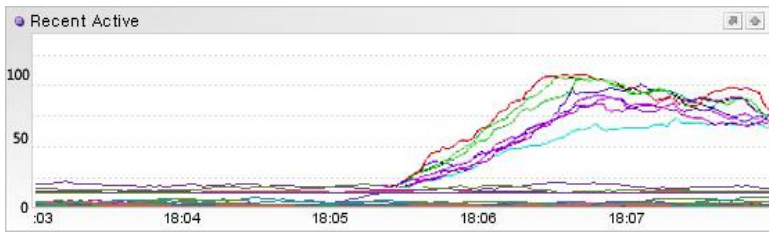
# 직관적인 액티브 서비스 모니터링



성능장애가 발생하면 응답시간이 지연되기 때문에 액티브서비스의 개수가 급증하게 됩니다. 제니퍼는 액티브서비스의 개수 및 서비스 경과시간을 색깔로 구분하여 서비스 정상 여부를 직관적으로 판단할 수 있도록 실시간 이퀄라이저 그래프를 제공합니다. 다수의 WAS 시스템을 동시에 통합 모니터링 하면서 어느 WAS 시스템에 성능장애가 발생하는지 한눈에 확인할 수 있습니다.

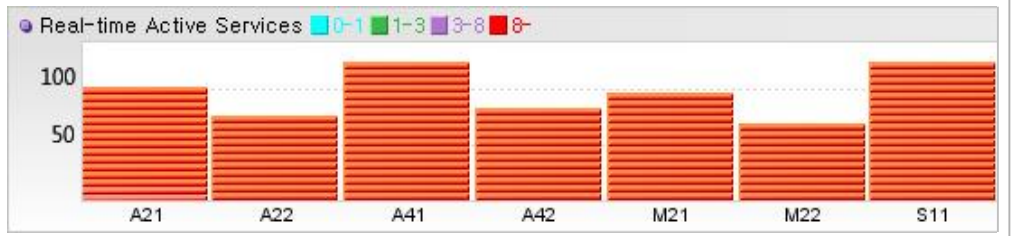


최근 액티브 서비스 개수 변화 그래프



액티브 서비스 개수

붉은색: 8초 이상, 분홍색: 3-8초, 초록색: 1-3초, 하늘색: 0-1초

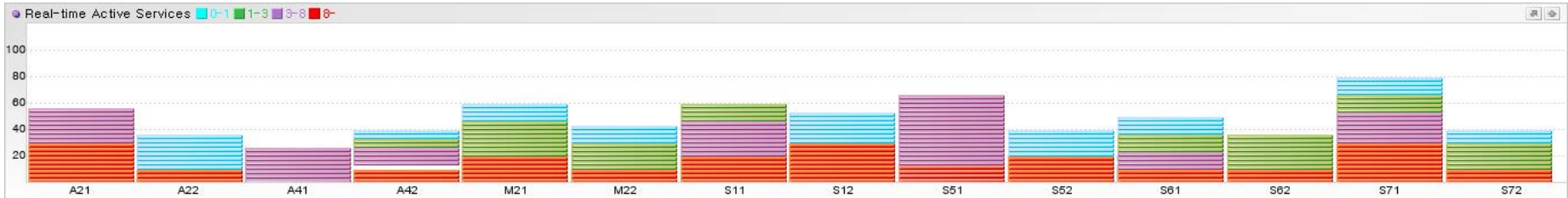


위 그래프는 갑작스럽게 전체 WAS 시스템의 액티브 개수가 증가하여, 각 WAS 서버마다 8초 이상 경과된 애플리케이션들이 100개 이상 존재하고 있다는 것을 나타내고 있습니다.

# 제니퍼의 주요기능 - 실시간 트랜잭션 분석



액티브서비스 이퀄라이저 그래프를 더블 클릭하여 해당 WAS시스템에서 현재 수행중인 액티브서비스의 상세목록을 실시간으로 확인할 수 있으며, 수행중인 애플리케이션의 클래스/메소드 레벨의 스택트레이스(Stack-trace)와 Profiling 정보를 제공합니다. 또한 수행중인 SQL 쿼리, 사용자 IP Address, 경과시간, CPU 수행시간을 제공합니다. 경우에 따라, 장시간 펜딩(pending)된 쓰레드(Thread)를 강제로 중단(kill) 시킬 수도 있습니다.



No	Time	Message
041	11:29:03.878	SOCKET-ISTREAM Socket[addr=211.233.81.17,port=80,localport=37937]
042	11:29:07.483	GET-CONNECTION DataSource.getConnection()
043	11:29:07.483	SQL-PREPARE_STMT(0)[SELECT A.EVENT_CD  B.NO AS SEQ_NO, A.EVENT_CD, B.NO, B.PRO_CD, A.EVENT_TITLE, A.START_DATE, A.END_DATE, A.UPDATE_DATE, A.EVENT_STYLE, A.TARGET_USER, A.WINNER_DATE, A.INVITE_PLACE, A.CONTENT, A.LINK_CATEGORY, A.EVENT_STATUS FROM TBL_EVENT_MAIN A, TBL_EVENT_PROC B WHERE A.EVENT_CD = B.EVENT_CD AND A.EVENT_STATUS = '\$' AND B.PRO_CD=?] [0 ms] param1:[R]
044	11:29:07.483	SQL-EXECUTE-QUERY(0) [2 ms] param1:[R] param2:['B0032222']
045	11:29:07.486	FETCH [0/100]
046	11:29:07.487	CLOSE-CONNECTION
047	11:29:07.489	SOCKET-OSTREAM Socket[addr=211.233.81.17,port=80,localport=37952]
048	11:29:07.489	SOCKET-ISTREAM Socket[addr=211.233.81.17,port=80,localport=37952]
049	11:29:07.489	GET-CONNECTION DataSource.getConnection() [1 ms]
050	11:29:07.490	SQL-PREPARE_STMT(1)[SELECT A.EVENT_CD  B.NO AS SEQ_NO, A.EVENT_CD, B.NO, B.PRO_CD, A.EVENT_TITLE, A.START_DATE, A.END_DATE, A.UPDATE_DATE, A.EVENT_STYLE, A.TARGET_USER, A.WINNER_DATE, A.INVITE_PLACE, A.CONTENT, A.LINK_CATEGORY, A.EVENT_STATUS FROM TBL_EVENT_MAIN A, TBL_EVENT_PROC B WHERE A.EVENT_CD = B.EVENT_CD AND A.EVENT_STATUS = '\$' AND B.PRO_CD=?] [0 ms] param1:[R]
051	11:29:07.490	SQL-EXECUTE-QUERY(1) [3 ms] param1:[R] param2:['B0032213']
052	11:29:07.493	FETCH [0/100]
053	11:29:07.494	CLOSE-CONNECTION
054	11:29:07.497	SOCKET-OSTREAM Socket[addr=211.233.81.17,port=80,localport=37953]
055	11:29:07.497	SOCKET-ISTREAM Socket[addr=211.233.81.17,port=80,localport=37953]
056	11:29:07.546	GET-CONNECTION DataSource.getConnection()
057	11:29:07.546	SQL-PREPARE_STMT(2)[SELECT A.EVENT_CD  B.NO AS SEQ_NO, A.EVENT_CD, B.NO, B.PRO_CD, A.EVENT_TITLE, A.START_DATE, A.END_DATE, A.UPDATE_DATE, A.EVENT_STYLE, A.TARGET_USER, A.WINNER_DATE, A.INVITE_PLACE, A.CONTENT, A.LINK_CATEGORY, A.EVENT_STATUS FROM TBL_EVENT_MAIN A, TBL_EVENT_PROC B WHERE A.EVENT_CD = B.EVENT_CD AND A.EVENT_STATUS = '\$' AND B.PRO_CD=?] [0 ms] param1:[R]
058	11:29:07.546	SQL-EXECUTE-QUERY(2) [2 ms] param1:[R] param2:['B00322480']
059	11:29:07.548	FETCH [0/100]
060	11:29:07.549	CLOSE-CONNECTION
061	11:29:07.549	SOCKET-OSTREAM Socket[addr=211.233.81.17,port=80,localport=37954]
062	11:29:07.549	SOCKET-ISTREAM Socket[addr=211.233.81.17,port=80,localport=37954]
063	11:29:07.557	GET-CONNECTION DataSource.getConnection()
064	11:29:07.557	SQL-PREPARE_STMT(3)[SELECT A.EVENT_CD  B.NO AS SEQ_NO, A.EVENT_CD, B.NO, B.PRO_CD, A.EVENT_TITLE, A.START_DATE, A.END_DATE, A.UPDATE_DATE, A.EVENT_STYLE, A.TARGET_USER, A.WINNER_DATE, A.INVITE_PLACE, A.CONTENT, A.LINK_CATEGORY, A.EVENT_STATUS FROM TBL_EVENT_MAIN A, TBL_EVENT_PROC B WHERE A.EVENT_CD = B.EVENT_CD AND A.EVENT_STATUS = '\$' AND B.PRO_CD=?] [0 ms] param1:[R]
065	11:29:07.557	SQL-EXECUTE-QUERY(3) [2 ms] param1:[R] param2:['B0028448']
066	11:29:07.560	FETCH [0/100]
067	11:29:07.560	CLOSE-CONNECTION
068	11:29:07.564	SOCKET-OSTREAM Socket[addr=211.233.81.17,port=80,localport=37955]
069	11:29:07.565	SOCKET-ISTREAM Socket[addr=211.233.81.17,port=80,localport=37955]

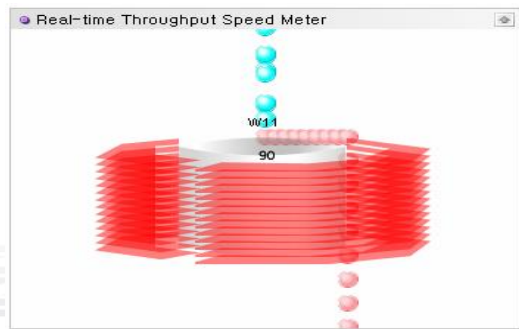
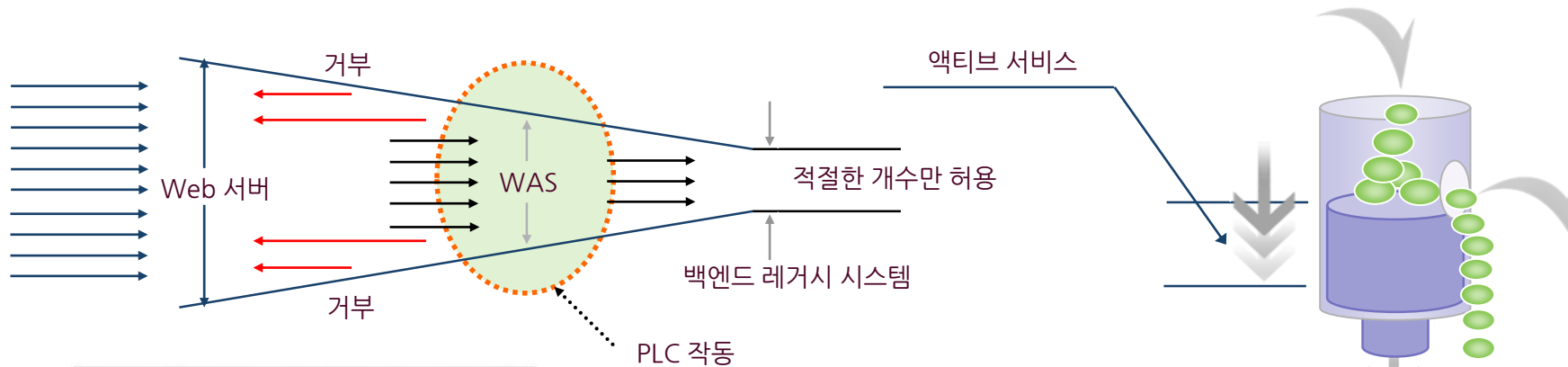
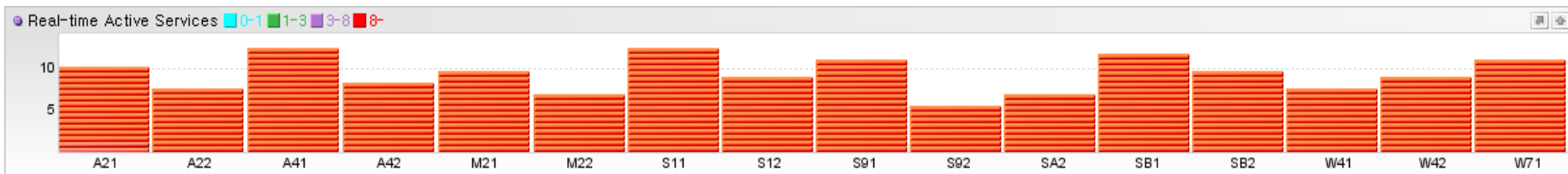
Agent	IP	Call Time	Response	CPU	Thread	Status	Business	Application	SQL	Fetch
WG1	211.233.81.59	12:44:55	8.849	0.000	6802362 / 0x3...	ICONCD	/search.jsp?BASIC_CD=101&VIEWU...	13	100	
WG1	211.233.81.59	12:45:01	2.52	0.000	6802370 / 0x3...	ISTCLD	/menu/ria/dnm/Plc_Inf_Dsc.jsp	1	1	
WG1	211.233.81.59	12:45:03	0.301	0.000	6802328 / 0x2...	APPRUN	/menu/ria/dnm/Ctt_Imm_Dsc.jsp	0	0	
WG1	211.233.81.59	12:45:03	0.299	0.000	6802346 / 0x3...	APPRUN	/menu/ria/dnm/Login_Check.jsp	0	0	
WG1	211.233.81.59	12:45:03	0.199	0.000	6802354 / 0x3...	APPRUN	/menu/ria/dnm/Ctt_Plc_List.jsp	0	0	

- 클래스/메소드 단위의 상세 스택트레이스(Stack-Trace)
- 현재 수행중인 SQL 쿼리(BIND 변수포함)
- 펜딩(Pending)된 쓰레드 중단(Kill) 기능
- 쓰레드 우선순위 변경 기능
- 사용자 IP Address, 수행시간/CPU/SQL시간 제공

# 제니퍼의 주요기능 - 서비스 폭주 시 부하량 제어



애플리케이션의 상대적 성능저하 원인으로 인해 응답을 주지 못하고 Running되고 있는 애플리케이션의 개수가 갑작스럽게 (혹은 점진적으로) 증가했을 때, 불과 몇 분 이내에 WAS에서 설정된 최대치의 Max Thread개수에 도달하게 되면, 최종사용자는 모래시계만 바라보게 되는 "서비스행(hang)" 상태에 이르게 됩니다. 이러한 문제를 해결하기 위해 일반적으로 WAS 재기동을 통한 임시방편적인 조치를 취해 왔습니다.





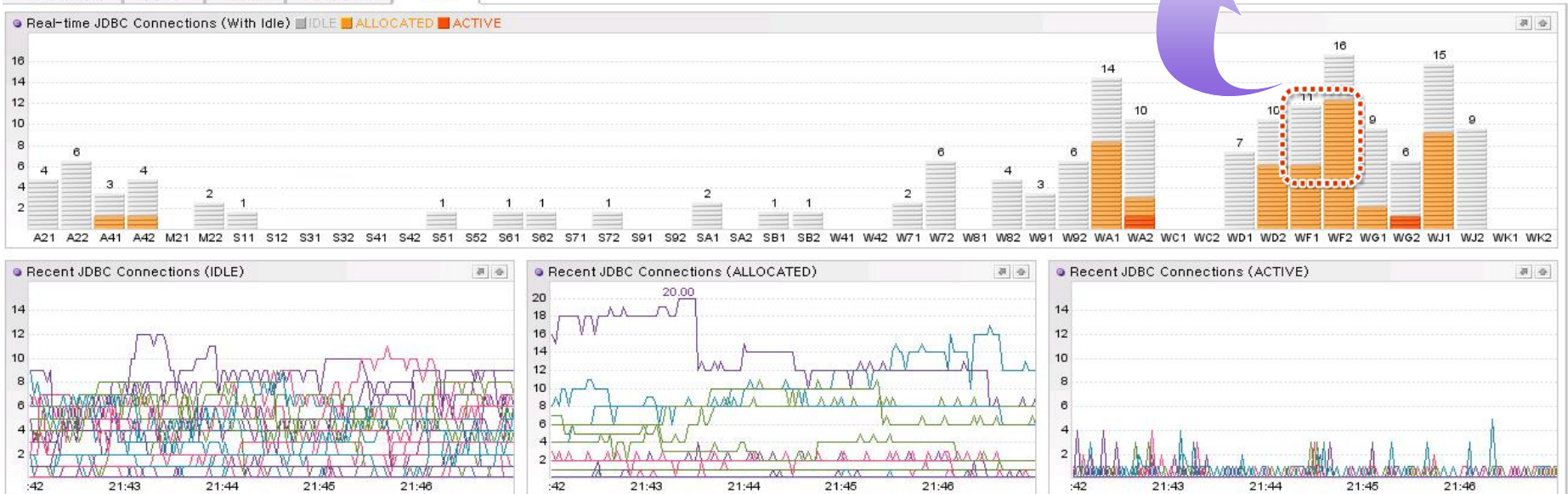
# 제니퍼의 주요기능 - 액티브 JDBC 모니터링



웹 애플리케이션의 주요 외부 거래 시스템인 데이터베이스(DB)와의 연동상태를 모니터링 하기 위해 JDBC Pool(idle), allocation connection, active connection 개수 정보를 실시간 제공합니다. 만약 Connection이 처리되지 않고 큐잉(Queuing)상태로 장시간 응답을 주지 않을 경우, 해당을 JDBC Connection을 선택하면 Connection을 사용하고 있는 애플리케이션에 대한 상세정보를 확인 할 수 있습니다.

Active Service	Application	SQL	External Transaction	Exception	JDBC				
Hash Code	DB SID	Class	Name	IP	Call Time	ResponseT	Thread	Status	Application
1a59d52	0	oracle.jdbc.driver.O	java:comp/e						
da2652	0	oracle.jdbc.driver.O	java:comp/e						
216884	0	oracle.jdbc.driver.O	java:comp/e						
1d5d4ef	0	oracle.jdbc.driver.O	java:comp/e						
156a7e5	0	oracle.jdbc.driver.O	java:comp/e						
b5b71e	0	oracle.jdbc.driver.O	java:comp/e	211.233.81.5	21:35:18	0.143	68021057/0:	JEXENG	<a href="#">/detail/place_end01.jsp</a>

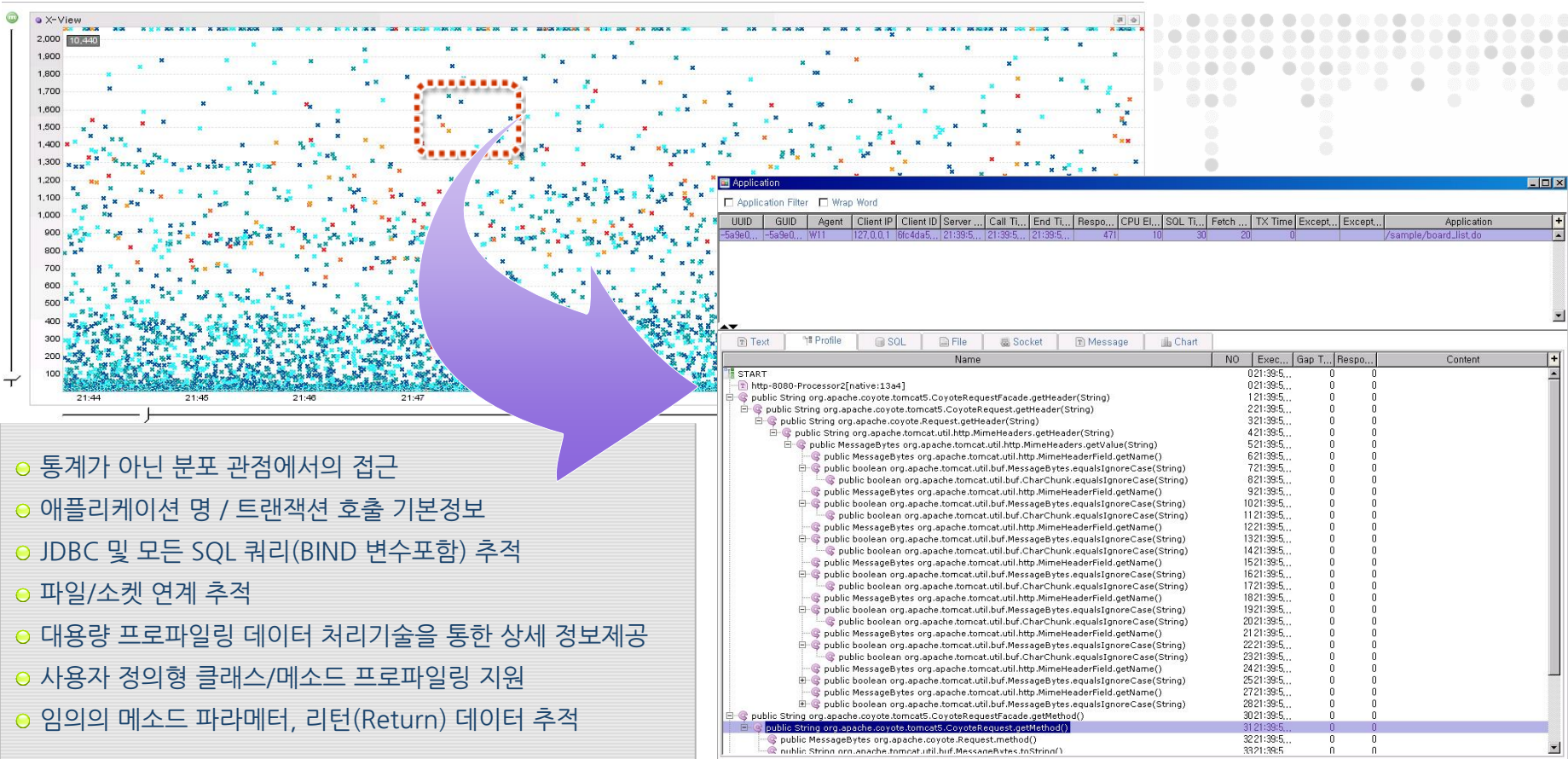
- JDBC 연결 총 개수
- 대기상태/할당/실행 상태
- 액티브 SQL 쿼리 확인
- JDBC 세션 아이디
- JDBC 클래스 이름
- 사용 중인 애플리케이션



# 제니퍼의 주요기능 - X-View와 상세 트랜잭션 프로파일링



제니퍼의 X-View 그래프는 각 개별 트랜잭션의 응답시간(ELAPSED) 관점 뿐만 아니라, 세로축(Y축)을 각각 CPU소요시간, SQL수행 시간, SQL 패치(Fetch)시간, TUXEDO/CICS/Tmax등과의 트랜잭션 인터페이스 시간을 기준으로 분포도의 형태를 변화시킬 수 있습니다. 즉, SQL쿼리 튜닝을 위한 용도라면, 비단 응답시간이 느린 관점 보다는 SQL수행 시간이 가장 높은 관점에서 분포도를 보는 것이 더욱 효과적이기 때문입니다. 또한 Mixed 타입을 통해 각 수행 영역별로 점유시간을 비교해 볼 수 있습니다. 이를 통해 어느 구간의 점유시간이 가장 많이 소요되었는지를 직관적으로 확인할 수 있습니다.



- 통계가 아닌 분포 관점에서의 접근
- 애플리케이션 명 / 트랜잭션 호출 기본정보
- JDBC 및 모든 SQL 쿼리(BIND 변수포함) 추적
- 파일/소켓 연계 추적
- 대용량 프로파일링 데이터 처리기술을 통한 상세 정보제공
- 사용자 정의형 클래스/메소드 프로파일링 지원
- 임의의 메소드 파라미터, 리턴(Return) 데이터 추적

# 제니퍼의 주요기능 - 다이나믹 프로파일(Dynamic Profile)



WAS시스템의 장애 분석 시 문제고립화 작업을 통해 문제의 원인에 접근하게 됩니다. 문제고립화 작업 시 클래스, 메소드 사용에 대한 상세 프로파일링을 어느 수준 까지 상세히, 그리고 시스템에 부담을 주지 않고 할 수 있는지가 이 작업의 핵심이라 할 수 있습니다. 제니퍼의 프로파일링 기능은 다이나믹 프로파일링을 표방하고 있는 타 APM의 한계로 존재하는 높은 CPU사용률 문제를 적극적으로 해결하면서 WAS의 재시작이 없이 최고의 분석데이터를 제공합니다.

Application	UUID	GUID	Agent	Client IP	Client ID	Server	Call Ti...	End Ti...	Respo...	CPU EL...	SOL Ti...	Fetch ...	TX Time	Except...	Except...	Application
...	3a190...	-3a190...	W11	127.0.0.1	8c44a5...	21:38:2...	21:38:2...	21:38:3...	4.627	30	1.150	161	0			/sample/board_list.do

Name	NO	Execute Time	Gap Time	ResponseTL	Content
START	0	21:38:27.730	0	0	
HTTP-8080-Processor[native:1b8]	0	21:38:27.730	0	0	
GET-CONNECTION [javaservice.edu.db.Connect]	1	21:38:27.730	0	3.114	
SOCKET-OSTREAM Socket[addr=127.0.0.1.p]	2	21:38:27.740	10	0	
SOCKET-ISTREAM Socket[addr=127.0.0.1.p]	3	21:38:27.740	0	0	
SQL-EXECUTE-QUERY(0) [493 ms]	4	21:38:31.504	3.184	481	SELECT boardno, title, name, email, createdate, hitcount FROM BOARD.

Name	NO	Exec...	Gap T...	Respo...	Content
START	0	21:38:27.730	0	0	
http-8080-Processor2[native:1344]	0	21:38:27.730	0	0	
public String org.apache.coyote.tomcat5.CoyoteRequestFacade.getHeader(String)	1	21:38:27.730	0	0	
public String org.apache.coyote.Request.getHeader(String)	2	21:38:27.730	0	0	
public String org.apache.tomcat.util.http.MimeHeaders.getHeader(String)	3	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaders.getValue(String)	4	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	5	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	6	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	7	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	8	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	9	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	10	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	11	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	12	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	13	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	14	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	15	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	16	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	17	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	18	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	19	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	20	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	21	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	22	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	23	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	24	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	25	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	26	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	27	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	28	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	29	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	30	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.MessageBytes.equalsIgnoreCase(String)	31	21:38:27.730	0	0	
public boolean org.apache.tomcat.util.buf.CharChunk.equalsIgnoreCase(String)	32	21:38:27.730	0	0	
public MessageBytes org.apache.tomcat.util.http.MimeHeaderField.getName()	33	21:38:27.730	0	0	

① 상세분석을 위한 추가적인 프로파일링 설정이 필요

② WAS의 재시작없이 상세 프로파일링이 가능

Class	Profile	Stacktrace	Application	Index	Stacktrace
org.apache.struts.action.Action.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionError.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionErrors.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionForm.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionFormBean.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionFormBeans.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionForward.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionForwards.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionMapping.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionMappings.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionMessage.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionMessages.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionServlet.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.ActionServletWrapper.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.DynaActionForm.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	
org.apache.struts.action.RequestProcessor.class	<input checked="" type="checkbox"/>	<input type="checkbox"/>		0	

Save Refresh

Activate Profile - All Classes Deactivate Profile - All Classes

프로파일 대상 선택/적용 후 X-View 화면



# 제니퍼의 주요기능 - 다이내믹 스택트레이스(Dynamic Stacktrace)



프로파일링 데이터에서 누락이 의심되는 성능저하구간이나, 트랜잭션 내 추가적인 상세분석이 필요한 일정 지점에 대해 WAS의 재시작 없이 풀 스택트레이스(Full Stacktrace)를 제공합니다. 이는 문제분석을 위해 시스템명령어(kill -3)를 사용해서 덤프파일을 기록, 분석하던 이전의 순차적인 방식을 벗어나, 실시간 시스템 상황과 연계하여 병렬적으로 문제원인에 보다 빠르게 접근 할 수 있도록 합니다.

The screenshot displays the Application Performance Monitoring (APM) interface. At the top, a table lists application transactions with columns for UID, GUID, Agent, Client IP, Server, Call Time, End Time, Response Time, CPU Elapsed Time, SQL Time, Fetch Time, TX Time, and Exception. Below this, a detailed view of a transaction is shown, including its ID, agent information, and timing data.

The main part of the screenshot shows a stack trace for a transaction. The stack trace is a list of method calls, with the following line highlighted in red:

```
[0002][21:42:32 722][ 0][ 0] public List javaservice.edu.model.BoardManager.findBoardList(int, int) [341 ms]
```

Below the stack trace, a table lists the classes and methods involved in the transaction. The table has columns for Class Name, Profile, Stacktrace, Application, Index, and Stacktrace. The class `BoardManager` is highlighted, and its `findBoardList` method is checked for stack tracing.

A callout box with a purple arrow points to the highlighted line in the stack trace, stating: "BoardManager.class가 사용되어지는 시점에 Full Stacktrace를 보여줍니다."

● BoardManager.class가 사용되어지는 시점에 Full Stacktrace를 보여줍니다.



# 사후분석을 위한 서비스 덤프 및 스냅샷 기능



상대적 성능 장애(Relative Performance Problem)이 발생하면 수행 중인 서비스의 응답시간 지연이 발생하기 때문에 항상 액티브서비스의 개수 증가로 이어집니다. 제니퍼는 액티브 서비스 개수 변화를 감지하여 설정된 임계치를 넘었을 때, 향후 문제 분석(사후분석)에 필요한 핵심적인 정보를 스냅샷(성능 데이터/그림파일) 형태로 자동으로 저장해 둡니다. 이를 통해 관리자 부재 시 발생한 문제에 대한 사후분석이 가능할 수 있도록 합니다. 특히 시각적 데이터(화면)를 손쉽게 확보, 유지할 수 있는 기능은 덤프파일분석 및 보고서 작성시 보다 효과적인 작업이 이루어 질 수 있도록 합니다.

- 1 시스템 장애발생 시 사후분석이 필요한 각종 시스템리소스 정보를 애플리케이션 수행정보 저장

## IIS 시스템 장애상황



```
서비스 덤프
(PULL TEXT)
/user/look/realtime-2.1.147/./dump_M2_20080504_09019.txt
Dump Time: 2008050409019
Host Name: win13
IP Address: 111.239.81.33
System CPU(%) : 2.78 (user: 1.9%, system: 1.9%, idle: 0.2%, wait: 0.0%, sleep: 96.2%)
System Memory(MB) total: 2054.0(MB), used: 1390.0(MB), free: 224.0(MB)
Agent Name: 822
Java Process Memory(MB) : 922.7(MB)
Java Process CPU(%) : 0.28 (user: 0.0%, system: 0.2%)
Java Heap Memory(MB) total: 254.1(MB), used: 149.1(MB), free: 105.1(MB)
Today VMs: Check: 907
Today GCs: 14959
Connections: Check: 174
Active Sessions: 174
Current Arrival Rate(qps): 0.5
Current Service Rate(qps): 1.0
Average Response Time(ms): 250.7
Trunk Time(ms): 6.3
[Current Active Sess:]
0:11:23:81.99:2008050409019:48827:0.0%:ICONEO/detail/wed_products.jsp:12 wq:78 fshd:
GET-CONNECTED DataSource.getConnection()
0:11:23:81.99:2008050409019:48827:0.0%:JSPDT/detail/wed_products.jsp:10 wq:66 fshd:
SOCKET-READ:Socket(addr=111.239.81.33,port=821,localport=7982)
public Object java.sql.CallableStatement.getObject(int)
0:11:23:81.99:2008050409019:48828:0.0%:ICONEO/detail/wed_products.jsp:10 wq:70 fshd:
SOCKET-READ:Socket(addr=111.239.81.33,port=821,localport=7977)
0:11:23:81.99:2008050409019:48828:0.0%:JSPDT/detail/wed_products.jsp:10 wq:68 fshd:
SOCKET-READ:Socket(addr=111.239.81.33,port=821,localport=7942)
public Object java.sql.CallableStatement.getObject(int)
0:11:23:81.99:2008050409019:48828:0.0%:ICONEO/detail/wed_products.jsp:10 wq:70 fshd:
SOCKET-READ:Socket(addr=111.239.81.33,port=821,localport=7982)
0:11:23:81.99:2008050409019:48829:0.0%:ICONEO/detail/wed_products.jsp:10 wq:65 fshd:
GET-CONNECTED DataSource.getConnection()
0:11:23:81.99:2008050409019:48829:0.0%:JSPDT/detail/wed_products.jsp:10 wq:65 fshd:
SOCKET-READ:Socket(addr=111.239.81.33,port=821,localport=7960)
0:11:23:81.99:2008050409019:48829:0.0%:ICONEO/detail/wed_products.jsp:10 wq:65 fshd:
SOCKET-READ:Socket(addr=111.239.81.33,port=821,localport=7960)
public Object java.sql.CallableStatement.getObject(int)
0:11:23:81.99:2008050409019:48830:0.0%:ICONEO/detail/wed_products.jsp:10 wq:65 fshd:
GET-CONNECTED DataSource.getConnection()
0:11:23:81.99:2008050409019:48830:0.0%:JSPDT/detail/wed_products.jsp:10 wq:66 fshd:
SOCKET-READ:Socket(addr=111.239.81.33,port=821,localport=8101)
```

## 제니퍼 서비스 덤프 주요내용

- 해당 시점에 액티브서비스 수행 목록
- 시스템 CPU(%)
- 시스템 메모리(MB)
- 에이전트(Agent) 명
- 자바 프로세스 메모리
- 자바 프로세스 CPU(%)
- 자바 힙(Heap) 메모리(MB)
- 해당 시점의 동시단말사용자 수
- 해당 시점의 호출빈도 (Arrival Rate)
- 해당 시점의 처리빈도 (Service Rate)

- 2 장애발생 시의 대시보드 데이터를 저장 및 유지



# 메모리 누수(Leakage) 추적 - 콜렉션/라이브 오브젝트 객체



힙 메모리릭을 유발시키는 주요원인으로서 특정 오브젝트의 지속적인 증가와 콜렉션류가 포함하고 있는 엘리먼트의 증가를 들 수 있습니다. 이러한 오브젝트를 사용하는 트랜잭션을 추적하기 위해 JVMPI(JVMTI)기법을 사용하는 경우가 있으나 기술 특성상 높은 CPU 사용률을 필요로 하게 됩니다. 대다수의 메모리릭 문제가 개발상보다는 실제 운영 시 일정기간 지속적인 메모리 증가와 함께 발생된다는 점을 고려했을 때 해당 기술을 사용한 툴로 메모리릭 분석을 한다는 것을 쉽지 않은 일이었습니다. 제니퍼의 경우 JVMPI(JVMTI)기술을 적용 없이 문제를 유발시키는 콜렉션 혹은 오브젝트를 감지하여 해당 오브젝트를 사용하는 애플리케이션 풀스택 (Full StackTrace)정보를 제공함에 따라 추가적인 시스템 리소스 부담 없이 문제의 원인에 접근할 수 있도록 합니다.

No.	Class	Live Objects	Stacktrace
[0000]	com.caucho.sql.UserConnectionAdapter	218	[Clear] [Trace Again] Stacktrace Added: at com.caucho.sql.UserConnectionAdapter. < init > (UserConnectionAdapter.java:111) at com.caucho.sql.XAConnectionAdapter.getConnection(XAConnectionAdapter.java:130) at com.caucho.sql.PoolItem.getConnection(PoolItem.java:303) at com.caucho.sql.DBPool.getConnection(DBPool.java:1230) at com.caucho.sql.DBPool.getConnection(DBPool.java:1177) at com.javaservice.jennifer.trace.JdbcTrace.getConnection(Unknown Source) at com.javaservice.jennifer.trace.HookImpl.getConnection(Unknown Source) at com.javaservice.jennifer.jdbc.DataSource.getConnection(DataSource.java:20) at common.DBMgr.getConnection(Unknown Source) at chn.the.CTheMain000CMD.processListAvi(CTheMain000CMD.java:194) at chn.the.CTheMain000CMD.processListAvi(CTheMain000CMD.java:71) at _reserve._category_0musical_0list_0i_0jsp.jspService(_category_0musical_0list_0i_0jsp.java:104) at com.caucho.jsp.JavaPage.service(JavaPage.java:75) at com.caucho.jsp.Page.subservice(Page.java:506) at com.caucho.server.http.FilterChainPage.doFilter(FilterChainPage.java:182) at com.caucho.server.http.Invocation.service(Invocation.java:315) at com.caucho.server.http.CacheInvocation.service(CacheInvocation.java:135) at com.caucho.server.http.RunnerRequest.handleRequest(RunnerRequest.java:346) at com.caucho.server.http.RunnerRequest.handleConnection(RunnerRequest.java:274)
[0001]	com.caucho.server.TcpConnection		
[0002]	tmax.webt.io.WebtTransceiver		
[0003]	com.caucho.util.Alarm\$AlarmThread		
[0004]	com.caucho.server.TcpServer		
[0005]	tmax.webt.WebtConnectionPoolMonitor		
[0006]	org.apache.commons.httpclient.MultiThreadedHttpConnectionManager		
[0007]	com.caucho.server.TcpServer\$Start		
[0008]	com.caucho.util.Cron\$CronThread		

Collection Monitoring Minimum Size:  Collection Auto Stacktrace Size:

Search(E) Snap Shot Clear Stack Accept All Stack Garbage Collection(G)

No.	Created Time	Total	Delta	Collection Name	Hash Code	Stack	URL
[0000]	20080620/170047	1,860	1,860	java.util.Hashtable	1c8968f	[Clear] [New Trace] Element Type : java.lang.String(3) at java.util.Hashtable.put(Hashtable.java:421) at lib.dic.SeatInfoDic.initialize(Unknown Source) at lib.dic.SeatInfoDic. < clinit > (Unknown Source) at _menu_ria_dnm_Booking_0List__jsp.jspService(_Booking_0List__jsp.java:3508) at com.caucho.jsp.JavaPage._service(Unknown Source) at com.caucho.jsp.JavaPage.service(Unknown Source) at com.caucho.jsp.Page.subservice(Page.java:506) at com.caucho.server.http.FilterChainPage.doFilter(FilterChainPage.java:182) at com.caucho.server.http.Invocation.service(Invocation.java:315) at com.caucho.server.http.CacheInvocation.service(CacheInvocation.java:135)	/menu/ria/dnm/Booking_List.jsp

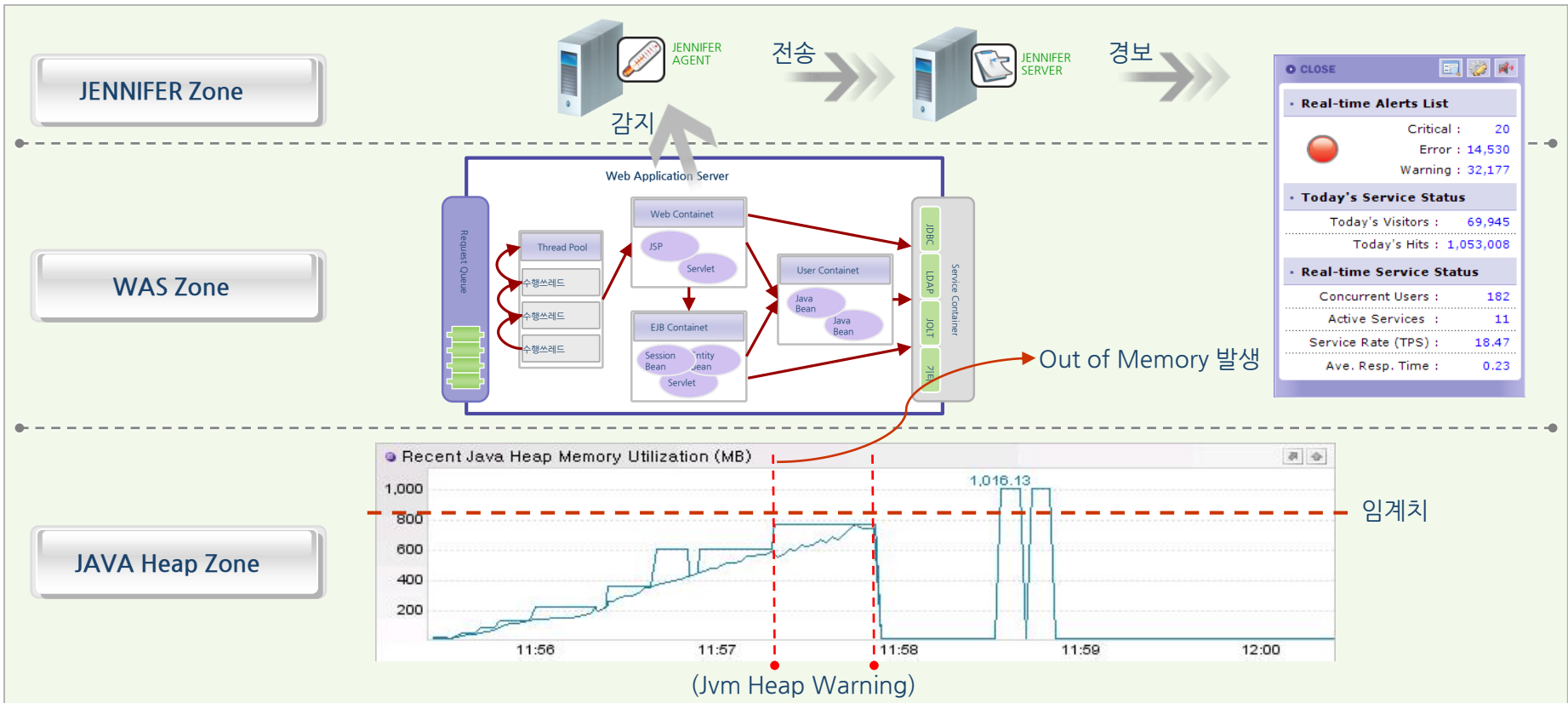
라이브 오브젝트

메모리 콜렉션

# 메모리 누수(Leakage) 감지



자바에서는 GC에 의해 메모리가 자동 관리되어 memory leak이 없다고 하지만, 사실은 memory leak문제는 장애의 많은 원인이 되고 있습니다. 그 이유는 실제로 사용되지 않는 객체의 reference를 프로그램에서 잡고 있으면 그 객체는 GC에 의해 처리되지 않고 프로그램 내에서도 접근하여 사용될 수 없는 사실상 쓰레기로서 메모리(보다 정확하게는 주소 공간)를 점유하게 됩니다. 이렇게 쓰레기메모리가 증가하다 보면 지정된 제한된 크기를 가지고 있는 힙 메모리의 여유 공간이 줄게 되고, 결국에는 OutofMemory 메모리를 유발, 서비스정지라는 장애로 이어질 수 있습니다. 제니퍼는 아래의 그림에서 제시된 것과 같이 OutofMemory발생직전 높은 힙 메모리가 점유되는 시점이 장시간 지속될 경우 경보를 통해 관리자에게 통보함으로써 사전조치를 취할 수 있는 시간적 여유를 제공합니다.



Real-time Alerts List	
Critical :	20
Error :	14,530
Warning :	32,177
Today's Service Status	
Today's Visitors :	69,945
Today's Hits :	1,053,008
Real-time Service Status	
Concurrent Users :	182
Active Services :	11
Service Rate (TPS) :	18.47
Ave. Resp. Time :	0.23

# TCP/IP 소켓(Socket) & 파일 추적 기능



제니퍼는 WAS 프로세스가 열고 있는 모든 TCP/IP 소켓(Socket) 상태를 모니터링 합니다. 또한 제니퍼는 대량 파일로깅(Logging) 작업이나 파일 업로드/다운로드 상황을 감지하기 위해 WAS 프로세스가 열어둔 파일(File)에 대한 접근상태(Read/Write)를 모니터링 합니다. 특히, 각 TCP/IP 포트별, 파일(File)별로 어떤 애플리케이션이 해당 자원을 사용하고 있는지에 대한 애플리케이션 스택트레이스(StackTrace)를 제공합니다. 이를 통해 다양한 시스템들과 복잡하게 연동되어 있는 외부 시스템과의 TCP/IP 소켓 현황을 보다 손쉽게 파악할 수 있을 뿐만 아니라, 프로세스당 최대 파일 디스크립트 (File Descriptors) 초과와 같은 시스템 장애를 감지할 수 있습니다.

[0027]	2008-11-24/12:41:25	[Waiting]	52823	211.233.81.17	80	0 / 350	0 / 214
[0028]	2008-11-24/12:41:32	[Trace Again]	52876	211.233.81.50	7160	0 / 26111	0 / 17365
[0029]	2008-11-24/12:41:4	Stacktrace Added: at oracle.net.ns.DataPacket.send(Unknown Source) at oracle.net.ns.NetOutputStream.flush(Unknown Source) at oracle.net.ns.NetInputStream.getNextPacket(Unknown Source) at oracle.net.ns.NetInputStream.read(Unknown Source) at oracle.net.ns.NetInputStream.read(Unknown Source) at oracle.net.ns.NetInputStream.read(Unknown Source) at oracle.jdbc.ttc7.MAREngine.unmarshalUB1(MAREngine.java:931) at oracle.jdbc.ttc7.MAREngine.unmarshalSB1(MAREngine.java:893) at oracle.jdbc.ttc7.Opens.receive(Opens.java:105) at oracle.jdbc.ttc7.TTC7Protocol.open(TTC7Protocol.java:614) at oracle.jdbc.driver.OracleStatement.open(OracleStatement.java:578) at oracle.jdbc.driver.OracleStatement.doExecuteWithTimeout(OracleStatement.java:2806) at oracle.jdbc.driver.OraclePreparedStatement.executeUpdate(OraclePreparedStatement.java:609) at oracle.jdbc.driver.OraclePreparedStatement.executeQuery(OraclePreparedStatement.java:537) at com.javaservice.jennifer.trace.sql.PreparedStatement4Oracle.executeQuery(Unknown Source) at mem.Login.getUserInfo(Login.java:368) at _cookieDel_jsp._jspService(_cookieDel_jsp.java:51) at com.caucho.jsp.JavaPage.service(JavaPage.java:75) at com.caucho.jsp.Page.subservice(Page.java:506) at com.caucho.server.http.FilterChainPage.doFilter(FilterChainPage.java:182) at com.caucho.server.http.Invocation.service(Invocation.java:315) at com.caucho.server.http.CacheInvocation.service(CacheInvocation.java:135) at com.caucho.server.http.RunnerRequest.handleRequest(RunnerRequest.java:346) at com.caucho.server.http.RunnerRequest.handleConnection(RunnerRequest.java:274) at com.caucho.server.TcpConnection.run(TcpConnection.java:139) at java.lang.Thread.run(Thread.java:534)	52937	211.233.81.33	1521	0 / 1995370	0 / 574564
[0030]	2008-11-24/12:41:59	[Waiting]	52998	211.233.81.50	10301	0 / 344	0 / 364
[0031]	2008-11-24/12:41:59	[Waiting]	52999	211.233.81.50	10301	0 / 344	0 / 364
[0032]	2008-11-24/12:41:59	[Waiting]	53000	211.233.81.50	10301	0 / 1904	0 / 1828
[0033]	2008-11-24/12:42:05	[Waiting]	53057	211.233.81.7	1521	0 / 1813	0 / 1968
[0034]	2008-11-24/12:42:05	[Trace]	53058	211.233.81.7	1521	0 / 1868	0 / 2600
[0035]	2008-11-24/12:42:12	[Trace]	53084	211.233.81.17	80	0 / 350	0 / 196
[0036]	2008-11-24/12:42:16	[Trace]	53103	211.233.81.17	80	0 / 350	0 / 194
[0037]	2008-11-24/12:42:27	[Trace]	53182	211.233.81.17	80	0 / 348	0 / 195
[0038]	2008-11-24/12:42:29	[Trace]	53196	211.233.81.50	7030	0 / 7739	0 / 6785



# 경보(Alert)



제니퍼는 WAS운영에 영향을 미칠 수 있는 시스템 및 애플리케이션 처리간 발생하는 다양한 다양한 에러 및 예외사항에 대해 실시간 감지할 수 있습니다. 이와 함께 대시보드 경고 창을 통한 실시간 알람 발생 및 관리자에게 다양한 매체(핸드폰, E-mail, 타 SMS솔루션 등)를 통해 실시간 문제발생여부를 전달 하여, WAS시스템에 대한 장애/에러 대처능력을 향상시킬 수 있도록 합니다. 에러/예외 사항은 "심각", "에러", "경고"로 각각 그룹핑되어 있으며, 일자 별 관리가 가능합니다. 필요하다면 임의의 사용자 정의형 "경고" 를 확장 어댑터를 통해 추가 등록할 수 있습니다.



## 심각(Critical)

ERROR\_SYSTEM\_DOWN  
ERROR\_PROCESS\_DOWN  
ERROR\_OUTOFMEMORY  
  
ERROR\_SERVICE\_QUEUING



경보 →



## 에러(Error)

ERROR\_HTTP\_IO\_EXCEPTION  
ERROR\_UNCAUGHT\_EXCEPTION  
  
ERROR\_PLC\_REJECTED



전송 ↓



## 경고(Warning)

WARNING\_DB\_CONN\_UNCLOSED  
WARNING\_JDBC\_STMT\_UNCLOSED  
  
WARNING\_APP\_BAD\_RESPONSE  
  
USER\_DEFINED\_FATAL  
USER\_DEFINED\_ERROR  
USER\_DEFINED\_WARNING



[SMS, E-Mail]

## 제니퍼 서버의 경고

Alert Type	Today's Alerts	
ERROR_SYSTEM_DOWN	1	[Reset]
ERROR_JVM_DOWN	5	[Reset]
ERROR_UNCAUGHT_EXCEPTION	8721	[Reset]
ERROR_JDBC_CONNECTION_FAIL	114	[Reset]
ERROR_MAYBE_GC_TIME_DELAY	52	[Reset]
WARNING_JDBC_CONN_UNCLOSED	527	[Reset]
WARNING_JDBC_STMT_UNCLOSED	712	[Reset]
WARNING_JDBC_PSTMT_UNCLOSED	3218	[Reset]
WARNING_JDBC_RS_UNCLOSED	5895	[Reset]
WARNING_JDBC_TOOMANY_RS_NEXT	7713	[Reset]
WARNING_JDBC_STMT_EXCEPTION	8	[Reset]
WARNING_JDBC_PSTMT_EXCEPTION	97	[Reset]
WARNING_JDBC_BAD_RESPONSE	3	[Reset]
WARNING_APP_BAD_RESPONSE	60	[Reset]

Alert

- 12:39:30 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:34 [W42] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:33 [W72] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(ticket/bin/admin/aver/main.jsp)
- 12:39:32 [W72] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(ticket/bin/af/easoadu/main.jsp)
- 12:39:32 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:30 [W62] WARNING\_JDBC\_PSTMT\_EXCEPTION(ticket/levi\_pjace.jsp)
- 12:39:22 [W42] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:19 [W22] ERROR\_UNCAUGHT\_EXCEPTION/minigame/minigame.jsp)
- 12:39:18 [W22] ERROR\_UNCAUGHT\_EXCEPTION/minigame/minigame.jsp)
- 12:39:15 [W22] ERROR\_UNCAUGHT\_EXCEPTION/admin/aver/aver\_gateway2.jsp)
- 12:39:14 [W01] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:07 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:06 [W41] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:05 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:39:05 [W41] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:59 [W42] ERROR\_UNCAUGHT\_EXCEPTION/ticket/bin/admin/aver/aver\_gateway2.jsp)
- 12:38:59 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:58 [W42] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:57 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:57 [W01] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:57 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:56 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:45 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:45 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:45 [W22] WARNING\_JDBC\_TOOMANY\_RS\_NEXT(menu/see7k.ftype.jsp)
- 12:38:43 [W01] ERROR\_UNCAUGHT\_EXCEPTION/ticket/bin/admin/aver/aver\_gateway2.jsp)
- 12:38:42 [W01] ERROR\_UNCAUGHT\_EXCEPTION/ticket/bin/admin/aver/aver\_gateway2.jsp)
- 12:38:40 [W41] ERROR\_UNCAUGHT\_EXCEPTION/ticket/bin/admin/aver/aver\_gateway2.jsp)

Real-time Alerts List

Critical : 20  
Error : 14,530  
Warning : 32,177

Today's Service Status

Today's Visitors : 69,945  
Today's Hits : 1,053,008

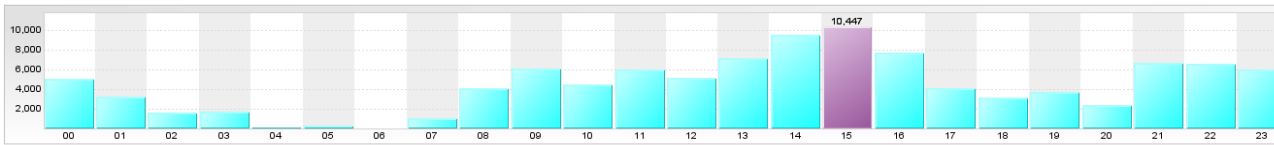
Real-time Service Status

Concurrent Users : 182  
Active Services : 11  
Service Rate (TPS) : 18.47  
Ave. Resp. Time : 0.23

# 애플리케이션 및 SQL 튜닝 데이터 제공



실행된 애플리케이션의 클래스/메소드 별로 응답시간, CPU사용시간 등을 추적할 수 있어, 어떤 모듈에서 병목이 발생하였는지를 세세하게 확인할 수 있습니다. 실행된 모든 SQL쿼리의 응답시간을 SQL 실행 시 사용된 BIND변수와 함께 성능저하 없이 추적합니다. 해당 SQL쿼리가 애플리케이션에서 응답시간 비중이 얼마나 차지하는지, 해당 SQL쿼리는 어떤 애플리케이션에 의해 사용되고 있는지 연관관계 분석을 할 수 있습니다. 제니퍼는 WAS서버를 경유하여 호출되는 백엔드 시스템의 트랜잭션 거러 내역을 모두 추적합니다. TMAX WebT, TUXEDO의 WTC/Jolt, 메인 프레임 CICS연동을 위한 CTG 모듈을 추적하여 WAS로부터 발생한 모든 트랜잭션의 응답시간 및 호출 건수를 실시간 모니터링이 가능하며, 통계화 과정을 통한 성능분석 데이터를 제공합니다.

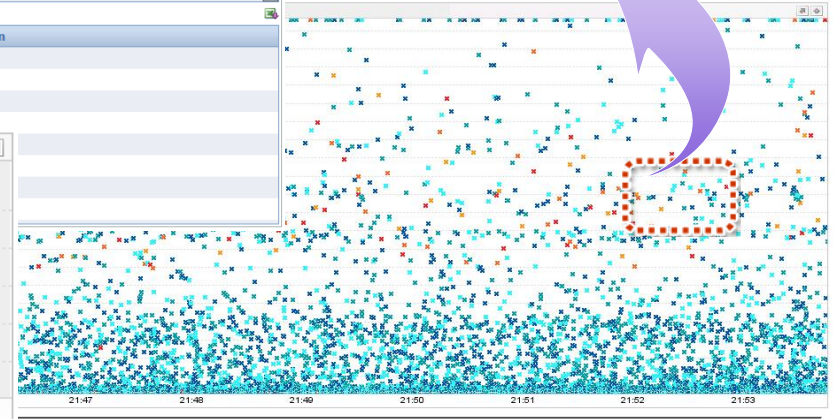
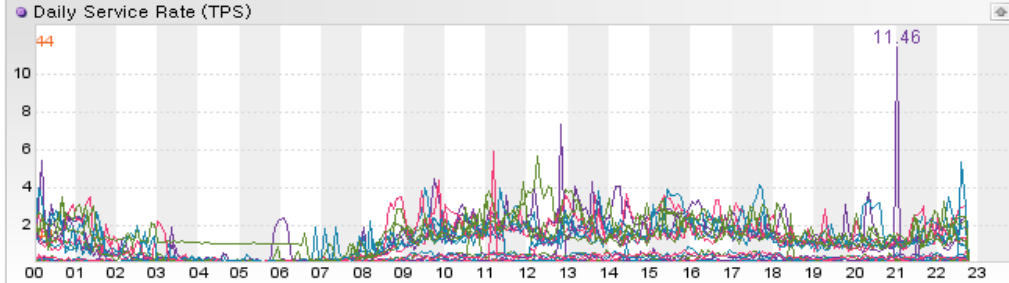


Time	End Time	Response	CPU El.	SQL El.	Fetch	TX Time	Except.	Except.	Application
5	21:35:5	471	10	30	20	0			/sample/board_list.do

Application	SQL	External Transaction	Exception									
Hits	Success	Fail Cou	Total El.	Occupai	Ave. Re	Std.Dev	Min. Re	Max. Re	Ave.CPI	CPU (tp	CPU Su	Application
5	5	0	67.328	7.458	13.466	0.543	12.557	14.177	0.013	0.000	0.067	/rcvdata.jsp
505	505	0	101.516	11.245	0.201	0.461	0.000	3.024	0.000	0.000	0.099	/index.jsp
31	31	0	62.33	6.904	2.011	0.039	1.966	2.149	0.002	0.000	0.062	/reserve/category_now_list_i.jsp
36	36	0	42.836	4.745	1.19	0.921	0.814	5.057	0.001	0.000	0.045	/reserve/timetable_list_i.jsp
129	129	0	43.479	4.816	0.337	0.703	0.052	3.389	0.000	0.000	0.045	/search.jsp
38	38	0	44.451	4.924	1.17	0.029	1.134	1.241	0.001	0.000	0.045	/auction/lottery1.jsp
59	59	0	49.372	5.469	0.837	0.368	0.019	2.725	0.001	0.000	0.051	/menu/seat/TktType.jsp
354	260	94	103.753	11.493	0.293	0.229	0.032	1.162	0.000	0.000	0.111	/detail/place_end01.jsp

NO	Exec...	Gap T...	Respo...	Content
021:39:5...	0	0	0	
021:39:5...	0	0	0	
121:39:5...	0	0	0	
221:39:5...	0	0	0	
321:39:5...	0	0	0	
421:39:5...	0	0	0	
521:39:5...	0	0	0	
621:39:5...	0	0	0	
721:39:5...	0	0	0	
821:39:5...	0	0	0	
921:39:5...	0	0	0	
1021:39:5...	0	0	0	
1121:39:5...	0	0	0	
1221:39:5...	0	0	0	
1321:39:5...	0	0	0	
1421:39:5...	0	0	0	
1521:39:5...	0	0	0	
1621:39:5...	0	0	0	

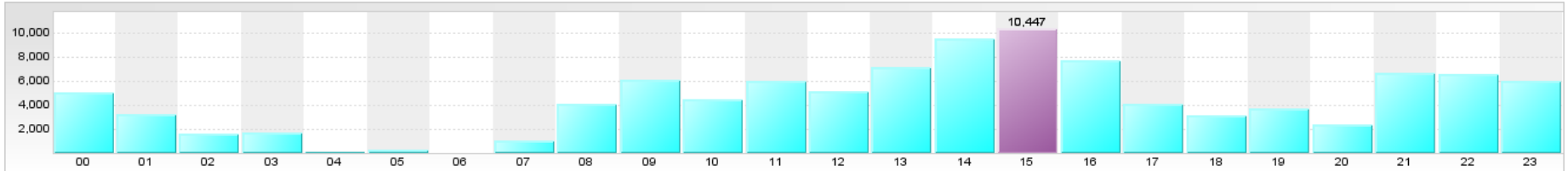
Type	Occupancy Ratio	Hits	Total El.	Ave. Re	Std.Dev	Min. Re	Max. Re	SQL / External Transaction
SQL	99.498	78	38.816	0.498	0.028	0.475	0.721	{ call PR_CIN_RES_NOW_LIST ( ?, ?, ?, ?, ?, ?, ? ) }
SQL	0.233	39	0.091	0.002	0.003	0.001	0.017	SELECT PRO_CD,
SQL	0.236	39	0.092	0.002	0.001	0.002	0.005	SELECT CATEGORY, KEYWORD, LPAD(RANK, #, #) AS RANK, COUNT
SQL	0.033	8	0.013	0.002	0.000	0.001	0.002	begin ? := PW_EM_ACCOUNT_USER_LST ( ?, ?, ?, ? );end;



# DB 연결 자원 미 반환 추적



제니퍼는 DB 액세스를 위해 사용되는 객체들(Connection, Statement, PreparedStatement, ResultSet 등)들을 애플리케이션에서 정상적으로 반환(Close)하지 않았을 때, 이를 자동으로 감지하여 애플리케이션 이름과 발생 위치를 스택트레이스(StackTrace) 정보와 함께 제공합니다. 아래 예와 같이 “JDBC ResultSet NOT CLOSED”를 선택하면, ResultSet을 정상적으로 닫지 않은 애플리케이션명과 관련 통계정보를 확인할 수 있으며, 해당 애플리케이션을 선택하면 문제 발생 지점의 상세 정보가 추가로 제공됩니다.



Daily All

Application	SQL	External Transaction	Exception
Occurrence	Occurrence Ratio	Exception	
308	43.688	JDBC ResultSet NOT CLOSED	
190	26.95	JDBC PreparedStatement NOT CLOSED	
140	19.858	UNCAUGHT APPLICATION EXCEPTION	
56	7.943	TOO MANY ResultSet FETCHED	
11	1.56	JDBC Statement NOT CLOSED	

Occurrence	Occurrence Ratio	Application	Content
152	49.351	/detail/place_end01.jsp	sql:[ SELECT COUNT(*) FROM TBL_TPROG_BASIC WHERE PRO_CD = ? ], rs created public java.sql.ResultSet<init> public ResultSet java.sql.PreparedStatement.executeQuery()
48	15.584	/detail/place_end01.jsp	sql:[ SELECT A.CSEQ_NO FROM TBL_COMM_TOURLIST A, TBL_CC rs created public java.sql.ResultSet<init> public ResultSet java.sql.PreparedStatement.executeQuery()
38	12.338	/auction/lottery1.jsp	sql:[ SELECT /*+ INDEX_desc (TBL_MEM_MASTER TBL_MEM_MASTER_IDX02) */ USER_ID,USER_NM,RING_POI rs created public java.sql.ResultSet<init> public ResultSet java.sql.PreparedStatement.executeQuery()

# 어플리케이션 그룹 모니터링 : BizMon



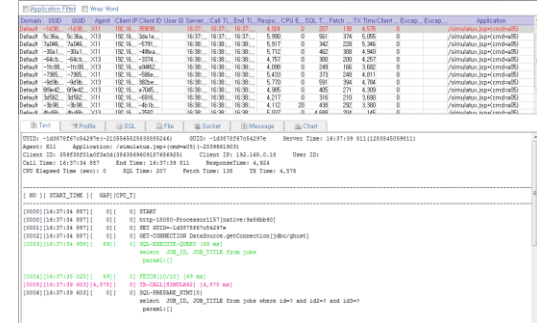
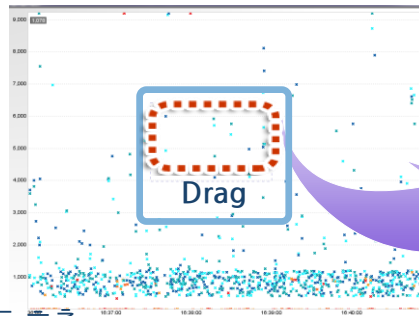
제니퍼는 주요 업무별 어플리케이션 그룹을 지정하여 이에 대한 성능 지표를 모니터링 하는 기능을 제공하며, 이에 대한 기능을 BizMon이라 합니다. 또한, 각 어플리케이션 그룹별 응답시간분포도를 확인할 수 있습니다.

## BizMon상의 성능 지표

- Client : 웹 브라우저에서 시스템 사이의 평균소요 시간
- Service : 평균 응답시간
- TP : 외부 트랜잭션의 평균 소요 시간
- DB : DB 평균 소요 시간 (SQL Query+Fetch Time)
- TPM : 분당 실행 건수

NAME	CLIENT	SERVICE	TP	DB	TPM
Human Resource	0.0	13.933	3.471	6.276	5.4
Payment	0.0	6.518	1.422	3.057	17.2
Account	0.0	2.782	0.9	1.144	63.4
Administration	0.0	1.294	0.553	0.457	110.2
Inventory	0.0	1.081	0.457	0.345	192.8
Maintenance	0.0	0.844	0.461	0.244	228.2
Production	0.0	0.658	0.421	0.133	316.79
Education	0.0	0.503	0.413	0.061	412.6
Procurement	0.0	0.481	0.368	0.067	320.2
Marketing	0.0	0.952	0.459	0.29	421.0

Click



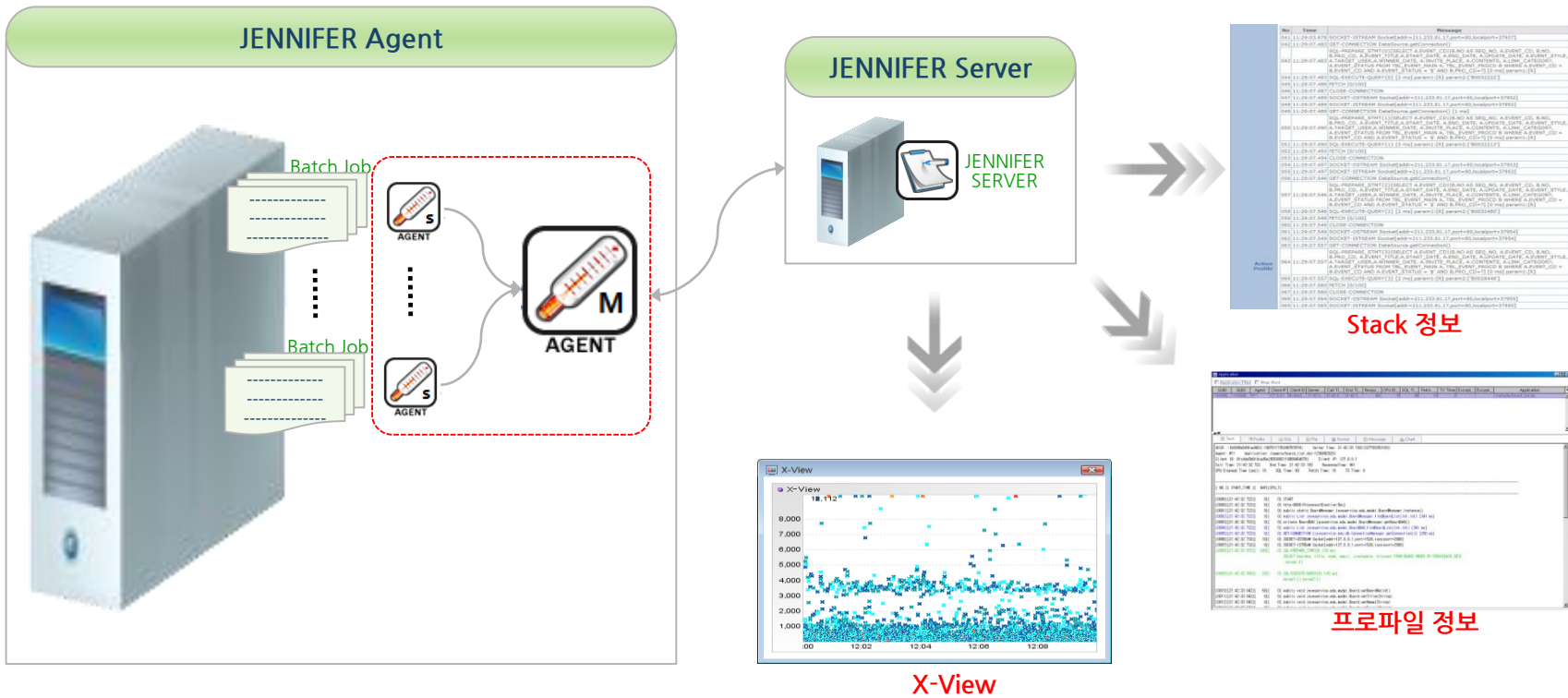
① 해당 어플리케이션 그룹의 응답시간 분포도 호출



# Batch Job Monitoring



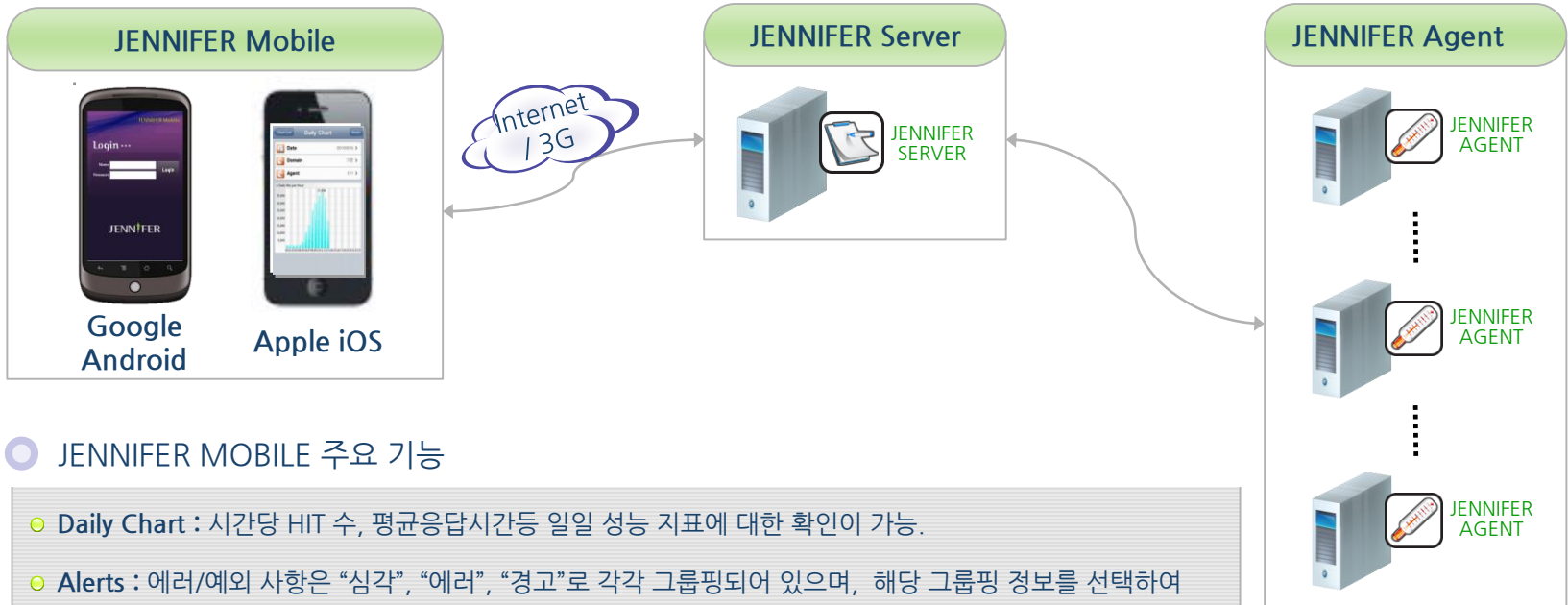
제니퍼는 Java 또는 .NET Component로 구축된 Batch Job에 대한 모니터링이 가능합니다. 해당 Batch Job의 기본적인 프로파일 정보 및 응답시간분포도를 제공할 뿐만 아니라 Stack 조회도 가능합니다.



# 스마트폰 지원 기능 : JENNIFER Mobile



JENNIFER는 스마트폰 환경에 대한 장점을 그대로 제품에 적용하여 '모바일 전자 웹 시스템 모니터링 기능'을 추가하였으며, 이에 대한 구성 체계 및 모듈은 JENNIFER Mobile이라 칭합니다. 시스템 운영자는 스마트폰 만을 이용하여 장소 및 시간에 구애 받지 않고, 좀더 빠르게 해당 전자 시스템의 성능 이슈를 감지하고, 이에 따른 확인 및 조치를 할 수 있습니다.



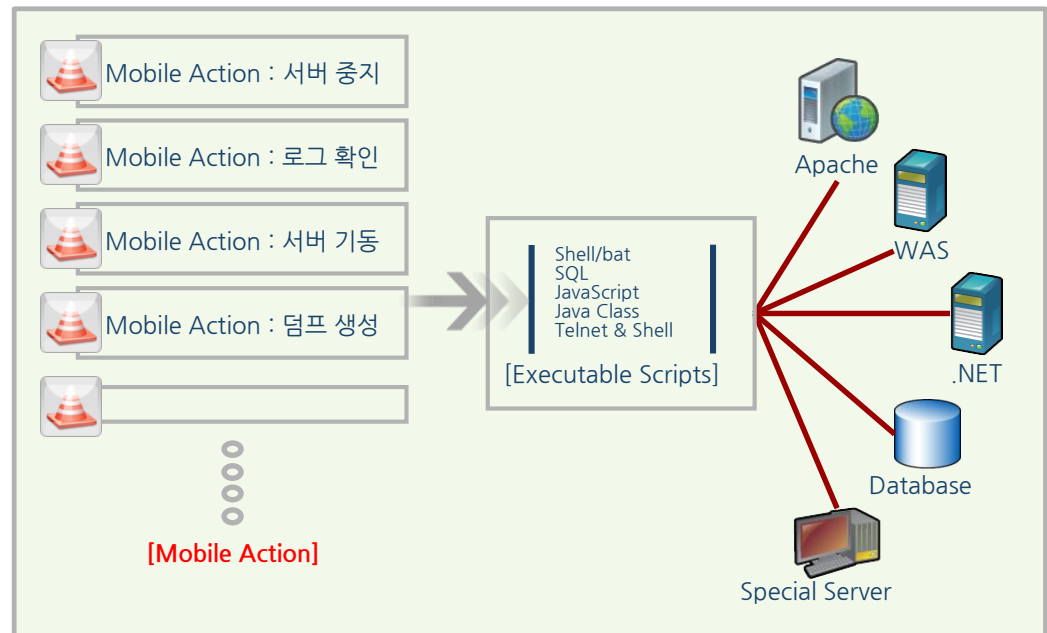
## ○ JENNIFER MOBILE 주요 기능

- **Daily Chart** : 시간당 HIT 수, 평균응답시간등 일일 성능 지표에 대한 확인이 가능.
- **Alerts** : 에러/예외 사항은 “심각”, “에러”, “경고”로 각각 그룹핑되어 있으며, 해당 그룹핑 정보를 선택하여 최근의 경보 상황에 대한 조회 기능.
- **Active Service** : 현재의 Active Service의 상황을 파악할 수 있으며, 액티브서비스 이퀄라이저 그래프를 더블 클릭하여 수행중인 애플리케이션의 클래스/메소드 레벨의 스택트레이스(Stack-trace)와 Profiling 정보에 대한 조회 기능
- **Mobile Action** : JENNIFER상에서 Action을 지정하고, 이를 스마트폰에서 구동할 수 있도록 하는 기능.

# 스마트폰 지원 기능 : Mobile Action



사용자(운영자)가 JENNIFER상에서 User Action을 설정하고, 이를 스마트 폰에서 구동하도록 하는 기능입니다. Mobile Action이란, 운영자가 모바일 환경에서 원하는 실행 코드를 JENNIFER상에 등록하여, 이를 JENNIFER Mobile Action으로 Wrapping/구동하는 방식입니다. User Action은 쉘 스크립트, 자바(Java)프로그램, C 라이브러리, 텔넷(Telnet, SSH지원), SQL 쿼리로 등록 가능합니다.



## Example : Server Restart

- 서버를 재기동할 경우, 우선 서버를 중단할 것이고 - 'Stop Server' Action, 해당 Process가 Stop 되었는지 확인할 것이고 - 'View the Log of Process', 이후에 서버를 기동할 것임 - 'Start Server' Action. 사용자가 원하는 서버 작업을 Action에 등록하고 이를 모바일 환경에서 구동 할 수 있음.

# 통계분석 및 보고서 지원



제니퍼는 접속자수, 응답시간 등 서비스 통계 및 자원사용량, 서비스 호출건수, SQL 수행내역, 백엔드 트랜잭션 거래 등에 대한 각 통계 데이터를 실제 업무에서 활용도가 높은 성능데이터 중심으로 일일, 주간, 월간 보고서는 제공하고 있습니다. 추가적으로 사용자정의형 보고서 작성 기능을 제공함으로써 DB에 저장되어 있는 제니퍼 모니터링 데이터를 자유롭게 가공, 사용자 요구에 맞는 보고서를 작성 할 수 있도록 합니다.

## 일일보고서

Category	Item	Value	Unit
A. Summary Data	Accessed Requests	1,000	Requests
	Accessed Bytes	1,000,000	Bytes
	Accessed Connections	1,000	Connections
B. Response Status	200	1,000	Count
	300	1,000	Count
	400	1,000	Count
C. Program Status	Program A	1,000	Count
	Program B	1,000	Count
	Program C	1,000	Count

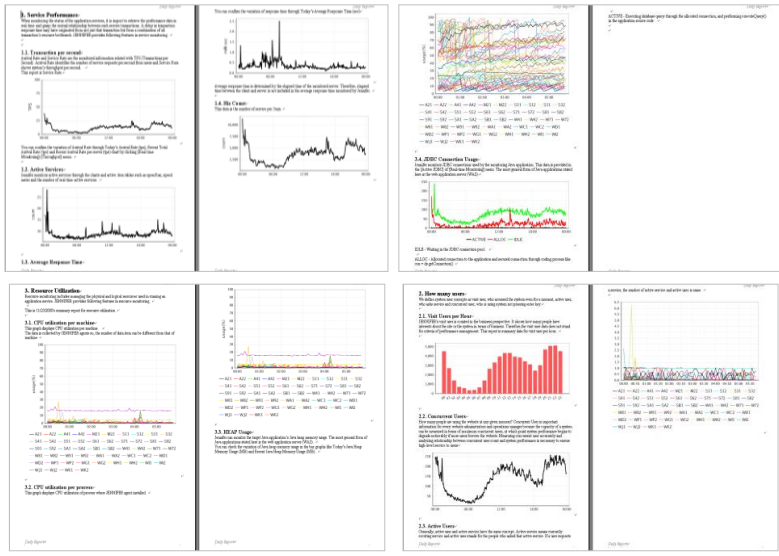
## 주간보고서

Category	Item	Value	Unit
A. Summary Data	Accessed Requests	1,000	Requests
	Accessed Bytes	1,000,000	Bytes
	Accessed Connections	1,000	Connections
B. Response Status	200	1,000	Count
	300	1,000	Count
	400	1,000	Count
C. Program Status	Program A	1,000	Count
	Program B	1,000	Count
	Program C	1,000	Count

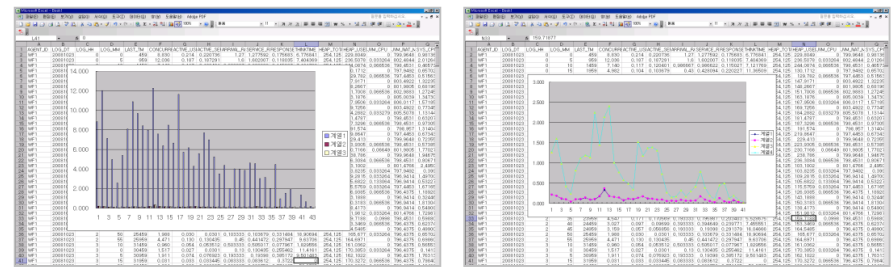
## 월간보고서

Month	Accessed Requests	Accessed Bytes	Accessed Connections
Jan	1,000	1,000,000	1,000
Feb	1,000	1,000,000	1,000
Mar	1,000	1,000,000	1,000
Apr	1,000	1,000,000	1,000
May	1,000	1,000,000	1,000
Jun	1,000	1,000,000	1,000
Jul	1,000	1,000,000	1,000
Aug	1,000	1,000,000	1,000
Sep	1,000	1,000,000	1,000
Oct	1,000	1,000,000	1,000
Nov	1,000	1,000,000	1,000
Dec	1,000	1,000,000	1,000

## 사용자 정의형 보고서



## 엑셀 데이터

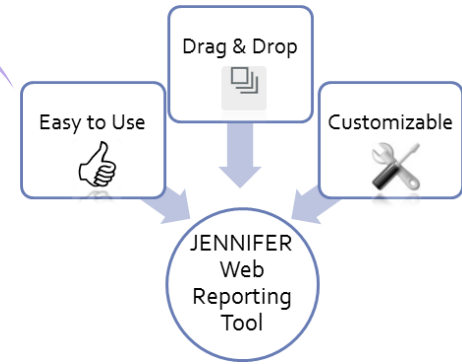




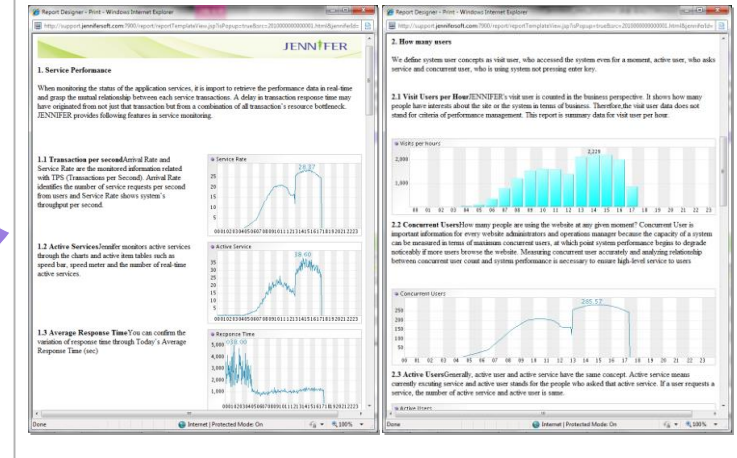
# 통계분석 및 보고서 지원: Jennifer WebReport



성능지표에 대한 보고서는 분야, 목적, 시점에 따라 해당 보고서의 형태 및 형식을 달리하여야 하며, 모니터링 툴은 사용자 정의 형 보고서에 대한 저작기능이 수반되어야 합니다. 이에 JENNIFER는 웹 기반의 WYSIWYG 보고서 저작기능(Jennifer Web Reporting Tool)을 통하여, 빠르고 쉽게 보고서를 저작할 수 있는 환경을 제공합니다. Jennifer Web Reporting Tool은 별도의 툴을 설치할 필요 없이 웹 브라우저만으로 리포팅 저작이 가능하며, 해당 Component를 Drag&Drop 형태로 보고서를 구성할 수 있습니다.



## 사용자 정의 보고서



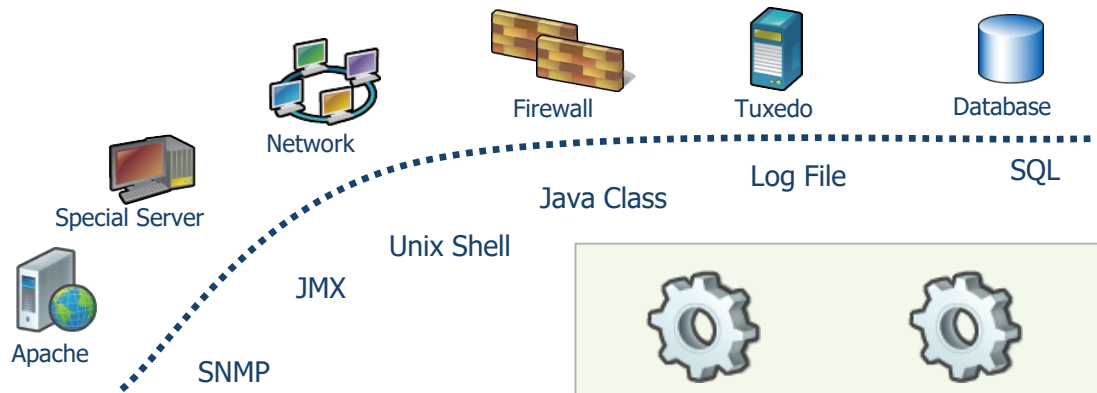
## JENNIFER Web Reporting Tool의 구성

- 1 Pool Area : 리포트 컴포넌트의 선택 영역
- 2 Reporting Settings : 해당 리포트에 대한 신규/미리보기/저장/프린트 등 명령 툴바
- 3 Design Area : 각 컴포넌트의 조합 패널
- 4 Properties : 각 컴포넌트의 설정 창

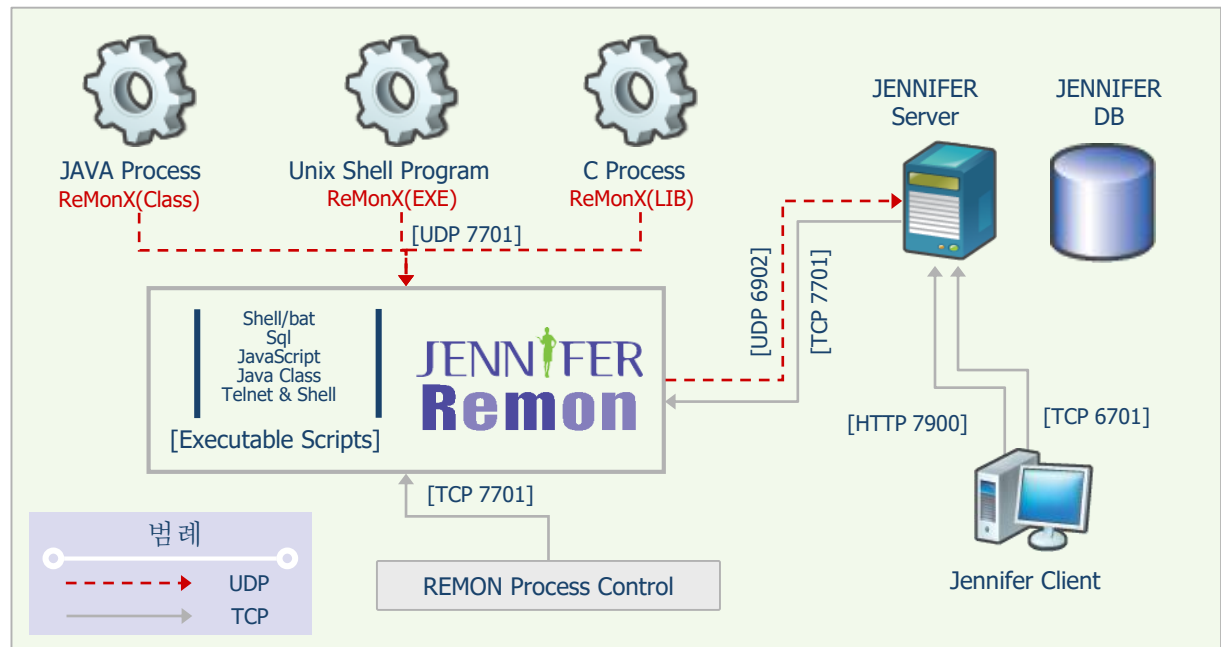
# 제니퍼의 모니터링 확장 기능 - 사용자 정의형 어댑터



제니퍼는 WAS/Java EE 중심적인 효과적인 성능 모니터링을 제공할 뿐만 아니라, 사용자 정의형 어댑터를 추가함으로써, 성능 모니터링 확장을 지원합니다. 제니퍼의 사용자 정의형 확장 인프라스트럭처인 레몬(REMON)은 쉘 스크립트, 자바(Java) 프로그램, C 라이브러리, 텔넷(Telnet, SSH지원), SQL 쿼리 등 다음과 같은 다양한 형태의 사용자 정의형 어댑터를 지원합니다. 제니퍼는 엔터프라이즈 모니터링을 위해, 실시간 데이터 전송, 저장, 통합 대시보드 등 성능 모니터링 인프라스트럭처 환경을 기본적으로 제공합니다. 제니퍼는 이 처럼 사용자 정의형 어댑터 기반의 모니터링 확장이 유연하여, WAS를 중심으로 엔터프라이즈 모니터링 아키텍처를 쉽게 구현 할 수 있습니다.



## EASY & FLEXIBLE MONITORING INFRASTRUCTURE



# 제니퍼의 모니터링 확장 기능 – 임의의 시스템/비즈니스 데이터 모니터링 어댑터 지원



제니퍼의 핵심역할인 WAS서비스 모니터링 및 리소스 모니터링 이외에 WAS시스템과 연동 혹은 독립적으로 운영중인 시스템의 사용상태를 통합 모니터링 할 수 있도록 레몬기능을 제공합니다. 시스템 운영자가 현재 모니터링 업무를 위해 사용하고 있는 Shell, SQL 포맷 등의 모니터링 명령어를 그대로 사용할 수 있도록 함으로써 모니터링 데이터 일관성 유지 및 시각적 대시보드 화, 모니터링 데이터의 데이터베이스화를 쉽게 구축할 수 있도록 합니다.

# Remon

Executable Scripts

Shell/Batch File

SQL

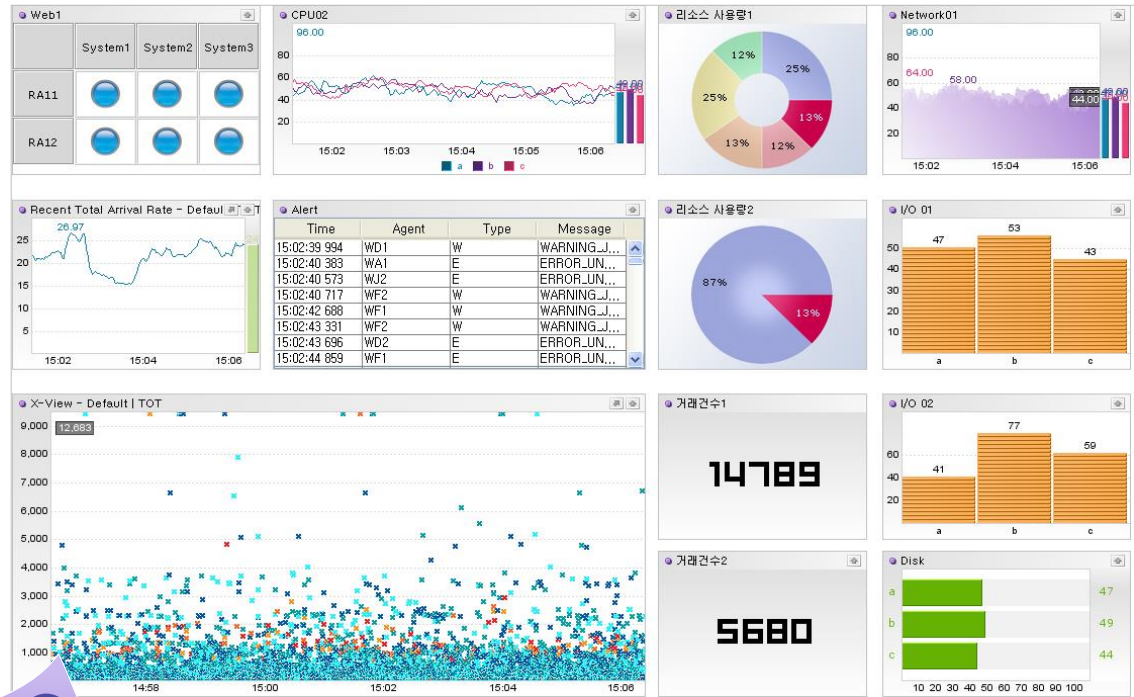
Java Script

Java Class

Telnet & Shell

```

File Edit View Options Transfer Script Tools Window Help
Remon [JFS] Host=EIPAX2 [Ref refresh=2] [VT100]
CPU User% Sys% Wait% Idle%
0 3.0 2.0 0.0 95.0
1 1.0 1.5 0.0 97.5
2 0.0 2.5 0.0 97.5
3 4.0 2.5 0.0 93.6
System Averages
RA1 2.0 2.2 0.0 95.8
Memory
Physical PageSpace | Pages/sec In Out | FileSystemCache
% Used 61.1% 0.1% | to Paging Space 0.0 0.0 | (numperm) 28.4%
% Free 38.9% 99.9% | to File System 0.0 0.0 | Process 27.3%
MB Used 5003.3MB 6.9MB | Page Scans 0.0 | System 5.4%
MB Free 3189.7MB 8185.1MB | Page Cycles 0.0 | Free 38.9%
Total(MB) 8192.0MB 8192.0MB | Page Steals 0.0 | Total 100.0%
| Page Faults 344.9 | numclient 26.3%
| | User 52.0%
| | Pinned 10.2%
Network
Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv Trans
en2 0.1 0.1 1.0 0.5 104.0 118.0 0.1 0.3
en1 0.4 1.5 6.4 5.9 63.5 265.2 73.5 72.0
lo0 7.3 7.3 49.4 49.4 151.6 151.6 54.5 54.5
Warning: Some Statistics may not shown
Ready Telnet 1. 1 31 Rows, 102 Cols | VT100
    
```

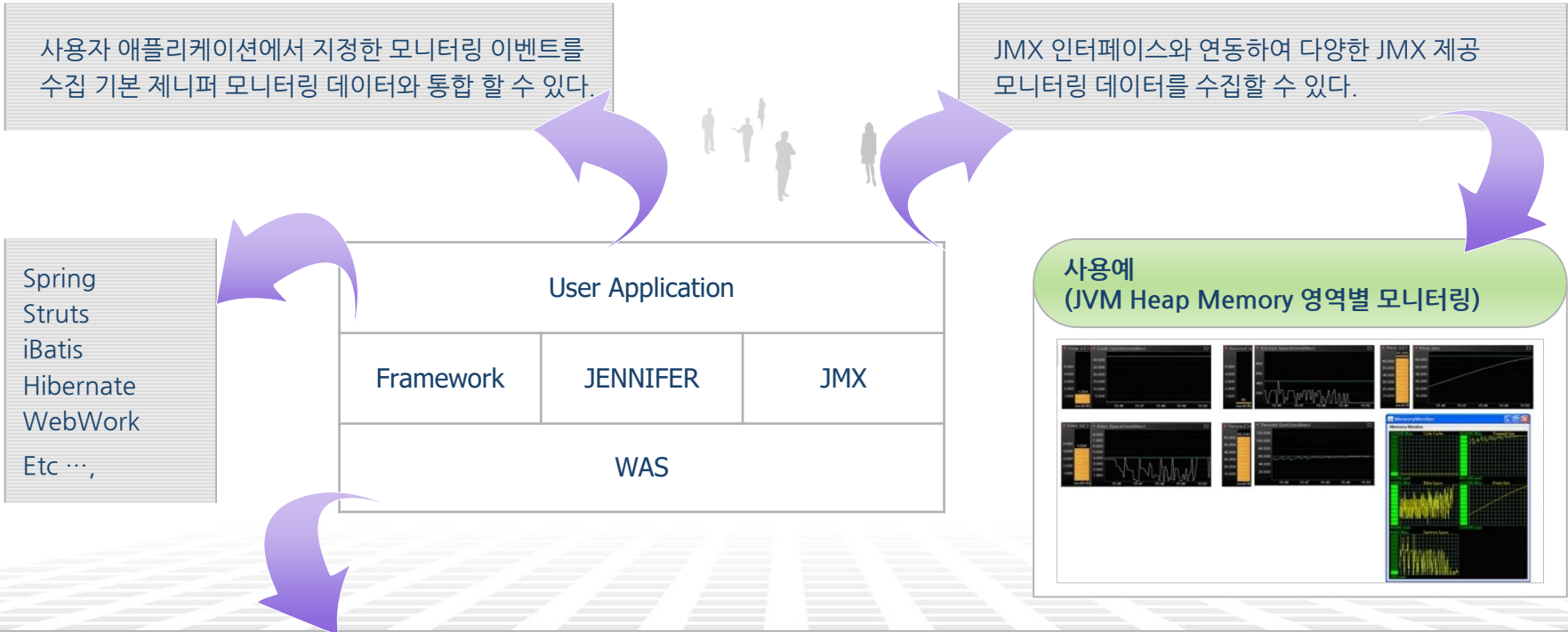


1 WAS 시스템 이외의 시스템 리소스 사용량 수집을 기존의 사용하던 셸 스크립트 등을 이용하여 손쉽게 수집/대시보드 화면을 구성할 수 있습니다. 또한 수집된 데이터는 데이터베이스에 저장됨으로써 사용자 업무프로세스에 적합한 보고서 형태로 2차 가공이 가능합니다.

# 제니퍼의 모니터링 확장 기능 - Extra Agent



엑스트라에이전트는 WAS 인스턴스 내부에 임베디드 되어 제니퍼 기본 모니터링 데이터 이외에 사용자가 필요한 다양한 데이터를 손쉽게 수집 할 수 있도록 함으로써, 애플리케이션(혹은 프레임워크)별 특징에 따른 특정 데이터 수집 및 임베디드 특징을 활용한 부하 없는 JMX 데이터 수집 등을 통해 한단계 높은 모니터링 수집 아키텍처를 구현하실 수 있습니다.



- |                    |  |  |   |  |
|--------------------|--|--|---|--|
| 제니퍼 서비스<br>덤프 주요내용 | <ul style="list-style-type: none"> <li>○ 시스템 사용자 정보                     <ul style="list-style-type: none"> <li>- 동시단말사용자수</li> <li>- 방문자수</li> <li>- 액티브 사용자수</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>○ 시스템 처리상태                     <ul style="list-style-type: none"> <li>- 초당처리건수</li> <li>- 시간당 호출건수</li> <li>- 평균응답시간</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>○ 자원상태                     <ul style="list-style-type: none"> <li>- 시스템 CPU 사용량</li> <li>- JVM CPU 사용량</li> <li>- 힙 메모리 사용량</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>○ 장애/에러                     <ul style="list-style-type: none"> <li>- 시스템 장애</li> <li>- 프로그램 에러</li> </ul> </li> </ul> |
|--------------------|--|--|---|--|



# 제니퍼의 모니터링 확장 기능 - 로그왓쳐(Log Watcher)



로그파일로 기록되는 다양한 데이터 및 에러를 모니터링 하는 작업은 WAS 서비스 모니터링에서 놓칠 수 있는 사각지대를 보완해줄 수 있는 중요한 작업입니다. 제니퍼의 로그 와쳐 기능은 다양한 포맷의 로그파일을 모니터링 하면서 모니터링에 필요한 정보를 필터링을 통해 통계화, 경보발생 등의 모니터링 작업을 지원합니다.

Apache Log
System Log
Business Log

Apache Log / System Log / Business Log 등 각종 로그파일들을 로그왓쳐를 사용하여 모니터링 정책에 맞는 경보발생 및 2차 로그관리를 효율적으로 수행할 수 있습니다.

통계화

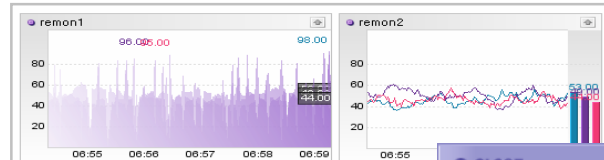


모니터링

## 통계화된 로그 데이터를 레몬을 통해 모니터링

## 통계화된 로그 데이터를 레몬을 통해 모니터링

[12/11/07 16:51:37:059 KST] 00000027 CoordinatorIm I HMGR0228I: The Coordinator is not an Active Coordinator for core group DefaultCoreGroup.  
 [12/11/07 16:51:37:065 KST] 0000000e ViewReceiver I DCSV1033I: DCS Stack DefaultCoreGroup at Member daou570Cell01Wdaou570Node01Wserver1: Confirmed all new view members in view identifier (4:0.daou570Cell01Wdaou570CellManager01Wdmgr). View channel type is View|Ptp.  
 [12/11/07 16:51:37:096 KST] 00000027 CoordinatorIm I HMGR0218I: A new core group view has been installed. The core group is DefaultCoreGroup. The view identifier is (4:0.daou570Cell01Wdaou570CellManager01Wdmgr). The number of members in the new view is 3.  
 [12/11/07 16:51:37:102 KST] 00000027 CoreGroupMemb I DCSV8050I: DCS Stack DefaultCoreGroup at Member daou570Cell01Wdaou570Node01Wserver1: New view installed, identifier (4:0.daou570Cell01Wdaou570CellManager01Wdmgr), view size is 3 (AV=3, CD=3, CN=3, DF=3)



remon3				
Time	a	b		
06:54:06 334	43	53		
06:54:08 338	44	52		
06:54:11 344	46	49		
06:54:12 346	45	50		
06:54:14 350	47	48		
06:54:16 354	46	51		
06:54:18 358	47	52		
06:54:20 362	45	48		
06:54:22 365	46	49		
06:54:24 369	46	49		
06:54:26 373	43	50		
06:54:28 377	42	51		

**Real-time Alerts List**

Critical : 20  
 Error : 14,530  
 Warning : 32,177

---

**Today's Service Status**

Today's Visitors : 69,945  
 Today's Hits : 1,053,008

---

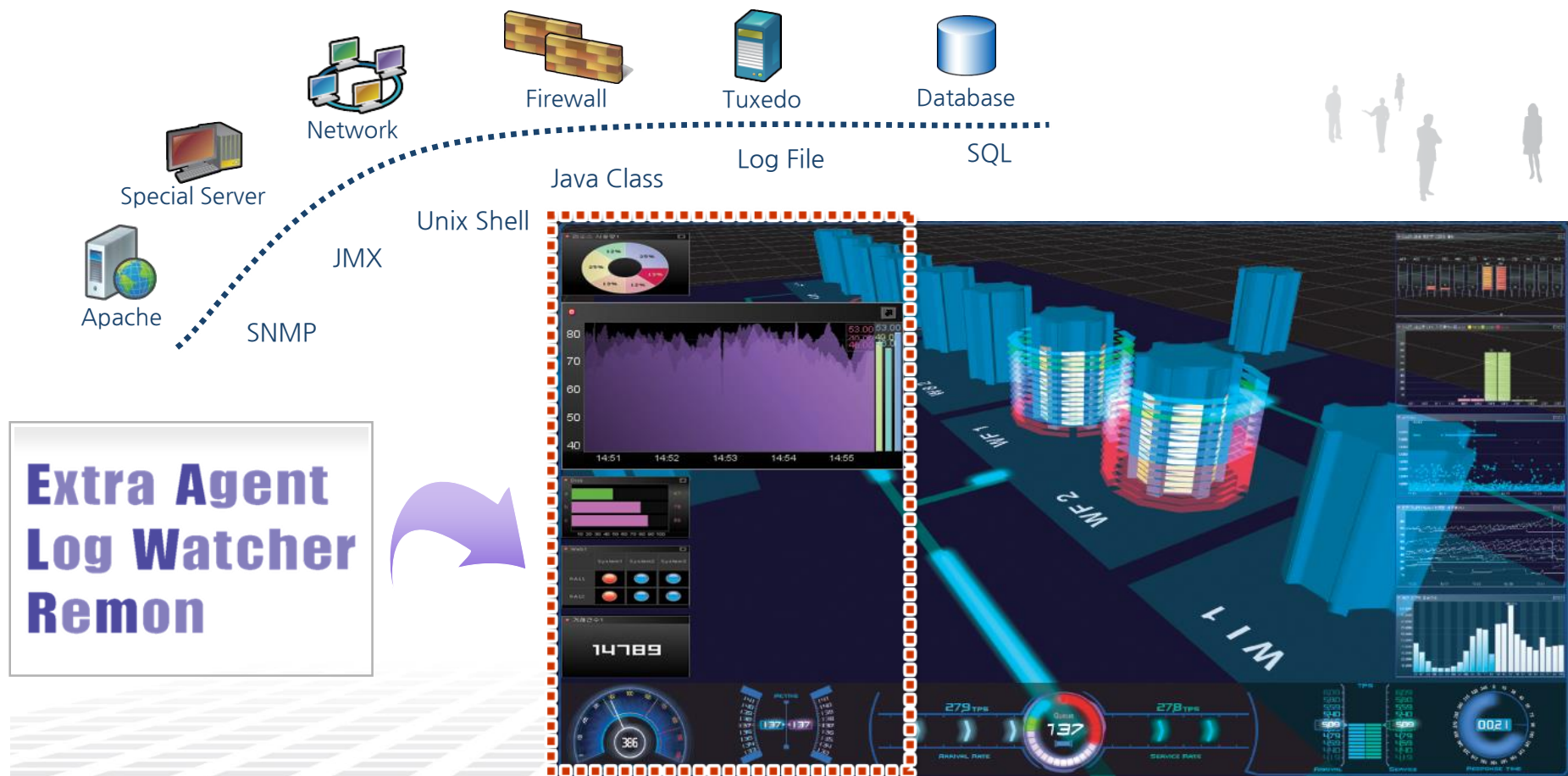
**Real-time Service Status**

Concurrent Users : 182  
 Active Services : 11  
 Service Rate (TPS) : 18.47  
 Ave. Resp. Time : 0.23

# 제니퍼의 확장 어댑터와 이클루스 연동



앞서 소개된 이러한 확장 어댑터를 활용하면 시스템 모니터링 범위는 사용자 시스템에 적합하도록 계속 확대해 나갈 수 있습니다. 모니터링 대상과 데이터가 증가될 수록 모니터링 데이터 표현을 2D로 하는 것은 2D가 가지고 있는 특성상 제한적일 수 밖에 없으며, 결국에는 모니터링 업무의 효율성을 떨어뜨리는 결과를 가져 옵니다. 데이터를 입체적으로 표현할 수 있는 3D기술을 적용한 이클루스는 2D표현방식의 제한을 극복하고 모니터링 시스템 대상을 제한 없이 자유롭게 확대, 구성함으로써 새로운 차원의 모니터링 환경을 제공할 것입니다.



**Extra Agent  
Log Watcher  
Remon**

# 모니터링 데이터 공유를 위한 Open API



엔터프라이즈 시스템 각 요소에 이미 포지셔닝 되어 있는 각종 솔루션과의 원활한 데이터 연동을 위해 제니퍼가 모니터링하고 있는 시스템 리소스 모니터링 데이터, 서비스 모니터링 데이터, 비즈니스 모니터링 데이터들을 XML데이터 형태로 제공되며, 필요에 의해 개별 트랜잭션 데이터를 데이터베이스 저장하거나 경보발생시 SNMP Trap전송과 같은 다양한 모니터링 연동방식을 제공합니다.

### WAS



JENNIFER AGENT

### WAS



JENNIFER AGENT

### WAS



JENNIFER AGENT

### 업무별/WAS별 성능

### WAS List



JENNIFER SERVER

Polling

Push



ESM, ITSM, ...

### TX별 성능

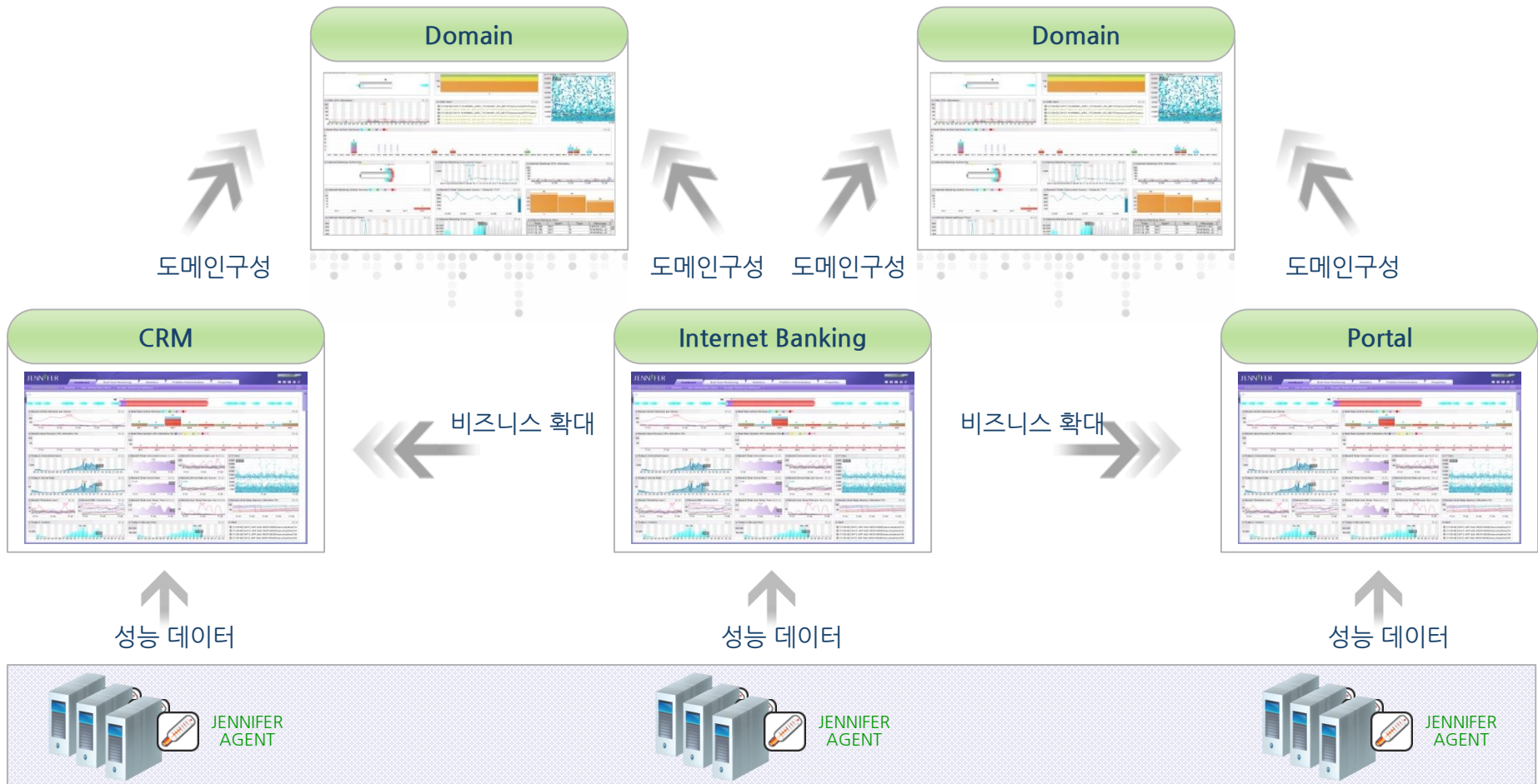
### Alert 이벤트

- [12:11:56] [WH1] UNCAUGHT EXCEPTION
- [12:11:56] [WF2] UNCAUGHT EXCEPTION
- [12:11:55] [WF1] UNCAUGHT EXCEPTION
- [12:11:50] [WJ2] UNCAUGHT EXCEPTION
- [12:11:50] [W11] JDBC PSTMT EXCEPTION
- [12:11:36] [WF1] UNCAUGHT EXCEPTION
- [12:11:27] [WF2] UNCAUGHT EXCEPTION
- [12:11:27] [WG1] UNCAUGHT EXCEPTION

# 제니퍼 도메인 구성



대규모 엔터프라이즈 환경에서는 서로 다른 비즈니스 업무 시스템이 존재하며, 비즈니스 환경변화 및 조직변화에 따른 WAS시스템 모니터링 환경의 변화가 요구될 때 추가적인 대시보드 개발에 따른 인력, 비용의 낭비 없이 시스템 통합 대시보드 환경을 구축할 수 있도록 도메인 아키텍처 기반의 사용자정의형 통합 대시보드 구성 환경을 지원합니다.







## End-User 만족을 위한 최적의 환경 구축

### 안정적인 시스템 운영

- 성능장애 현상 발생시 즉각적인 원인분석으로 신속한 해결을 통해 다운타임을 최소화하여 시스템을 보다 안정적으로 운영

### 장애대응 능력 확보

- 지속적인 시스템의 성능 모니터링을 통해 향후 발생할 가능성이 있는 위험을 예측하고 미연에 방지할 수 있으며 자동경보 기능을 제공

### 정량화된 성능근거 자료

- 접속자/부하량 등에 대한 정량화된 데이터 수집 및 통계를 근거로 시스템 확장 및 개편 시점과 증설/ 튜닝 작업에 따른 정량화된 근거자료를 확보

### 통합 대시보드 구축

- 서비스관점에서의 성능 모니터링 시스템을 손쉽게 구축하여 서비스 현황을 실시간으로 통합 관제

### 고객 서비스 만족도 향상

- 다운타임 최소화, 장애대응능력확보 및 자동 부하량 제어 기능으로 안정적이고 신뢰성 높은 보다 근접한 24x365 시스템 운영이 가능하게 되어, 궁극적으로 대 고객 만족도를 증대

### 효율적인 애플리케이션 성능관리

- 서비스 레벨 목표치인 SLA를 수립하여 그 임계치를 넘어 문제발생 시 문제점을 빠르게 해결할 수 있도록 함으로써 효율적인 애플리케이션 성능관리에 기여

# 제니퍼 경쟁우위



## 1 제니퍼 경쟁 우위 관점

- 리얼타임(Real-Time) & 인사이트(Insight)
- 전거래 응답시간분포(X-View) 인터페이스
- 엔터프라이즈 성능관리 관점 제시
- EASY & POWERFUL

## 2 제니퍼 차별적 특장점

- 다이내믹 프로파일링 ON/OFF 설정
- 다이내믹 스택트레이스 추출
- 액티브 프로파일링 기능
- 도메인별 통합모니터링 관리
- 모니터링 확장 어댑터 기능
- 실시간 통합 로그 모니터링





# 감사합니다



서울시 구로구 디지털로 30길 28 마리오타워7층 701호  
TEL: 02-558-9130 / FAX: 02-558-7868 / [www.in-con.biz](http://www.in-con.biz)