

HanTrace™

for JAVA System





1

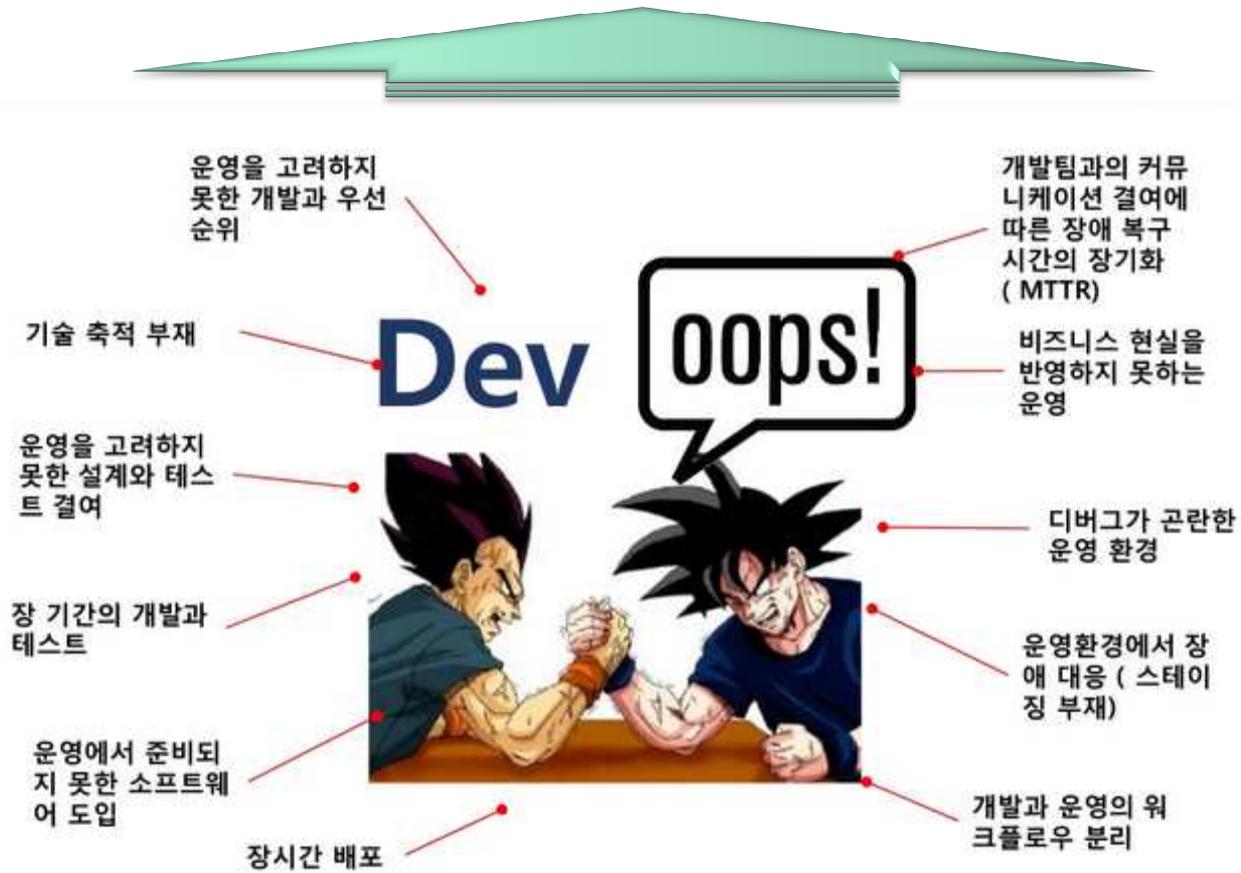
HanTrace™

I. HanTrace™ 제품 개요

II. HanTrace™ 제품 특징

III. HanTrace™ 차별화

운영과 개발이 모두 소통하는 비즈니스 거래 추적 중심의 애플리케이션 품질 모니터링 솔루션



MTTR : Mean Time To Repair (장애 복구 평균 시간)

✓ HanTrace 의 주요 특징

◆ IT Operations

✓ 비즈니스 서비스 Performance

핵심 비즈니스 서비스 성능 모니터링 및 Alert 를 통해 비즈니스 장애에 빠르게 대응

✓ End User 불평

End User 불평이 나오기 전에 문제를 빠르게 해결하고 장애 티켓을 줄이고 허위 경고를 제거하는데 도움

✓ Performance 분석

비즈니스 서비스 기준 상세 성능 및 에러 분석 및 통계 제공

HanTrace™

◆ Developments

✓ 빠르게 Application 문제 발견

비즈니스 서비스 중심으로 Application 문제를 발견

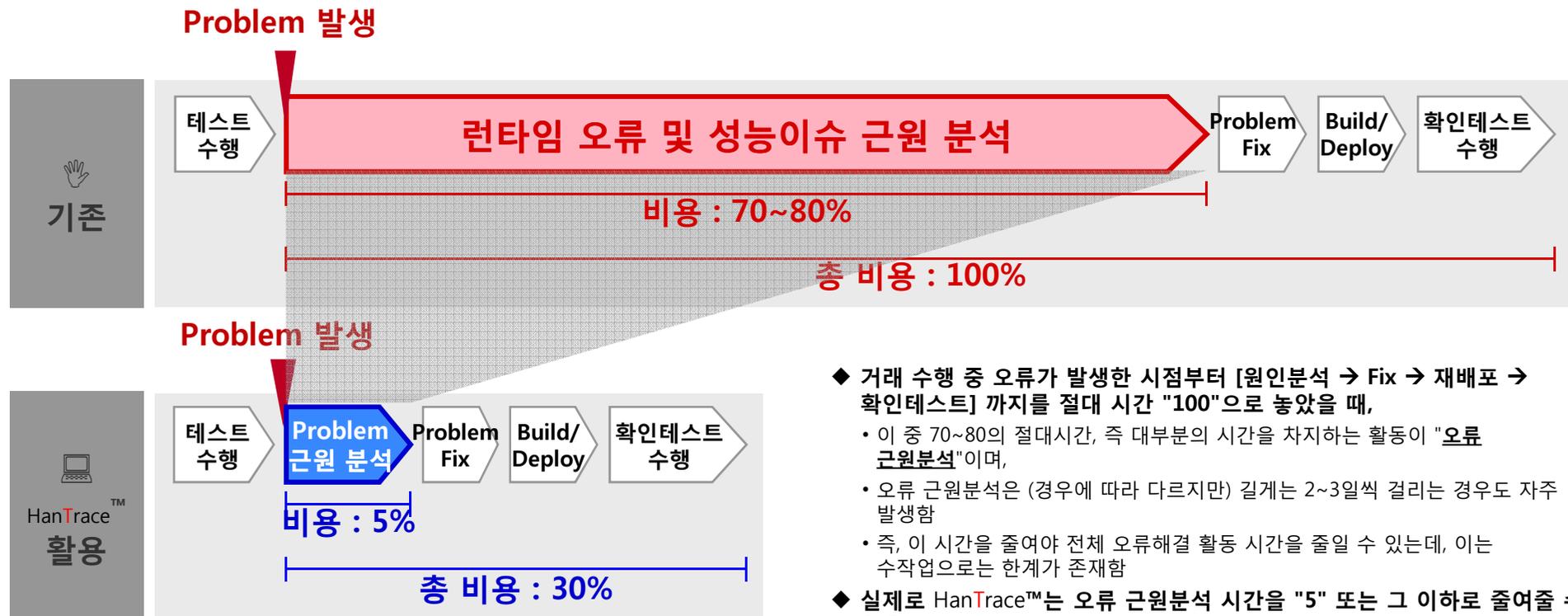
✓ Application 오류 근원 분석

비즈니스 서비스 중심으로 Application 문제의 근원을 상세하게 파악하여 쉽고 빠르게 문제 해결

✓ 비즈니스 서비스 Performance

비즈니스 서비스 에서 성능에 이슈가 되는 거래들을 쉽게 분석하여 Application 품질 개선에 기여

1. 정확한 원인분석으로 결함분석 시간 단축

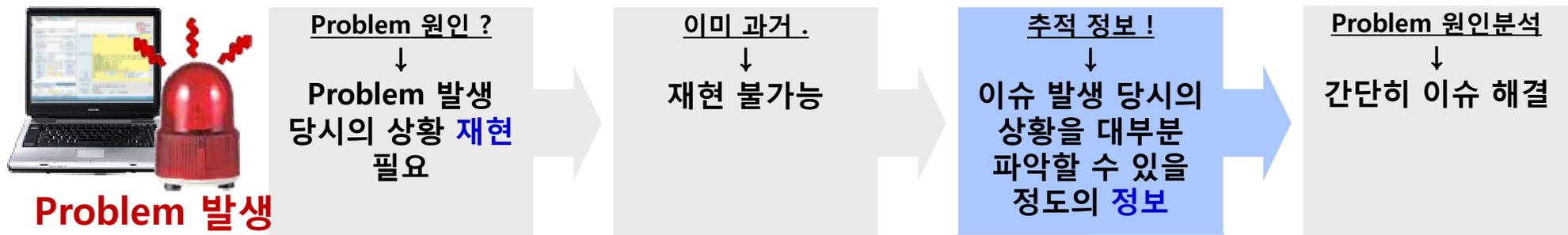


- ◆ 거래 수행 중 오류가 발생한 시점부터 [원인분석 → Fix → 재배포 → 확인테스트] 까지를 절대 시간 "100"으로 놓았을 때,
 - 이 중 70~80의 절대시간, 즉 대부분의 시간을 차지하는 활동이 "오류 근원분석"이며,
 - 오류 근원분석은 (경우에 따라 다르지만) 길게는 2~3일씩 걸리는 경우도 자주 발생함
 - 즉, 이 시간을 줄여야 전체 오류해결 활동 시간을 줄일 수 있는데, 이는 수작업으로는 한계가 존재함
- ◆ 실제로 HanTrace™는 오류 근원분석 시간을 "5" 또는 그 이하로 줄여줄 수 있음
- ◆ 예) 'Null Pointer Exception' 오류 근원분석 및 해결
 - 1) 오류 발생 : Not Null 컬럼에 Null값이 Insert된 오류 발생
→ HanTrace™가 없다면, 이 이상의 정보를 찾기란 매우 어려움. 즉 근원(Root Cause)을 알기 위해 많은 노력 필요
 - 2) 역추적 하면서, 해당 Insert문 필드에 값을 할당하기 위해 사용한 SQL 문장(Select문)을 확인함
 - 3) Select문에서 Fetch된 컬럼의 값이 "Null"임을 확인함
 - 4) Fetch된 해당 컬럼이 Not Null 컬럼이라는 것을 DB 설계를 통해 확인함
 - 5) 데이터 이행팀에, 해당 컬럼에 Null값이 들어가지 않도록 이행 프로그램 수정을 요청함

→ 결국 해당 오류는 프로그램 오류가 아니라 데이터 이행 프로그램 오류로 판명됨

2. 장애시점의 상세 프로파일 기록정보를 통한 빠른 원인분석

- ◆ Problem의 상세한 수행 이력을 각종 부가 정보와 함께 추적하여 통합 레포지터리에 프로파일 정보를 기록함
- ◆ 거래별로 수행된 상세 Flow 기반으로 추적 가능하며, 추적하면서 활용되는 정보들은 입력데이터, 패키지>클래스>메서드, Context, SQL 문장/입력파라미터 등임
- ◆ 무작위로 수행되는 다중 사용자의 거래와 Random하게 처리되는 Multi-WAS 환경에서, Unique하게 거래를 식별하고 정확히 Ordering하여 이슈를 진단할 수 있는 툴임



Ⅲ. 차별화

① 실시간 모니터링의 범위가 어느 정도인가?

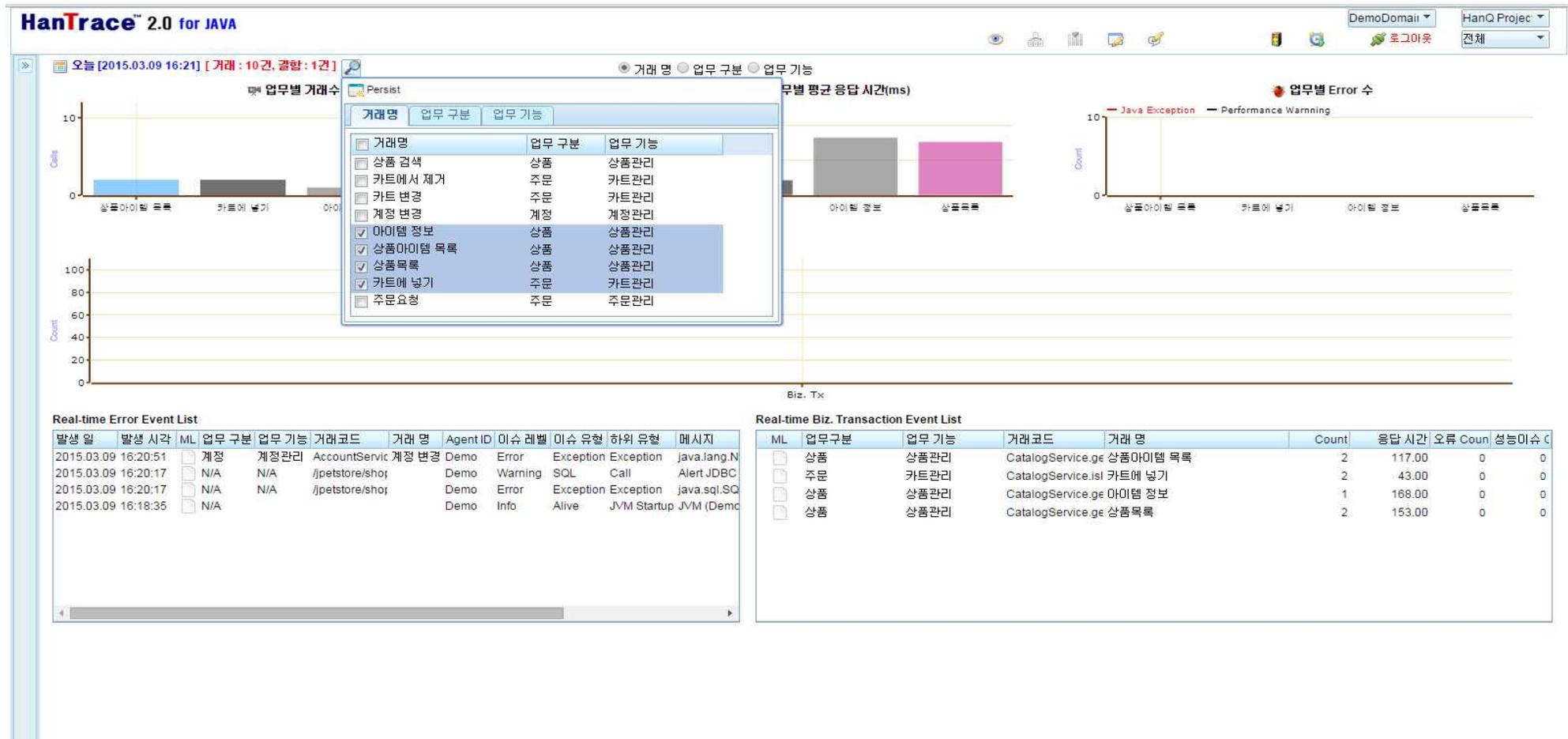
② 장애파악과 분석 내용이 정확하고 신속한가?

③ 원인분석 결과가 장애해결에 부족함이 없는가?

구분		APM	HanTrace™
기준	BSM	<ul style="list-style-type: none"> • 비즈니스 서비스 모니터링을 제공하지 못하거나 미약하다. 	<ul style="list-style-type: none"> • 비즈니스 서비스 Dashboard 를 통해 실시간 모니터링 가능 • 비즈니스 서비스 기준 성능 이슈 모니터링 • 비즈니스 서비스 기준 Application 에러 모니터링
	DevOps	<ul style="list-style-type: none"> • 운영 중심 성능 분석 정보 제공 <ul style="list-style-type: none"> ➢ 실시간 운영 중심 모니터링에 집중되어 있으며, 문제 발생 감지 이후 강력한 분석 및 해결 기능이 미약함 • Application 내 발생하는 오류에 대한 수집 미약 <ul style="list-style-type: none"> ➢ 실시간 Application 내 발생한 오류에 대하여 극히 제약적으로 제공함 • JDBC Resource Leak 에 대한 감지만 제공 	<ul style="list-style-type: none"> • 개발 과 운영을 모두 만족하는 성능 분석 정보 제공 <ul style="list-style-type: none"> ➢ 운영 중에 실시간 발생한 어플리케이션의 업무 성능 및 에러에 대하여 개발 과 운영의 상호 협력이 필요한 상세 정보 제공 ➢ 비즈니스 거래 중심 자료 제공 • Application 내 발생하는 오류에 대하여 전수 감지 및 근원 분석 정보 제공 <ul style="list-style-type: none"> ➢ 모든 Application 오류 감지 및 근원 분석 정보 제공 ➢ SQL Runtime 오류에 대한 쉽고 빠르게 근원 분석 정보 제공 • JDBC Resource Leak 에 대한 감지 및 발생 Source Code 위치 분석 <ul style="list-style-type: none"> ➢ JDBC Resource Leak 에 대하여 감지되면 실시간 경보 및 상세 원인 분석 기능 제공 ➢ Source Coding 오류에 따른 JDBC Flow Error 실시간 감지 및 상세 원인 분석 기능 제공
	기타	<ul style="list-style-type: none"> • 거래 사후 분석이 제한적이다 	<ul style="list-style-type: none"> • 거래 사후 분석이 강력하다 • User Transaction 을 집중적으로 분석 • Dynamic BCI • Easy Configuration

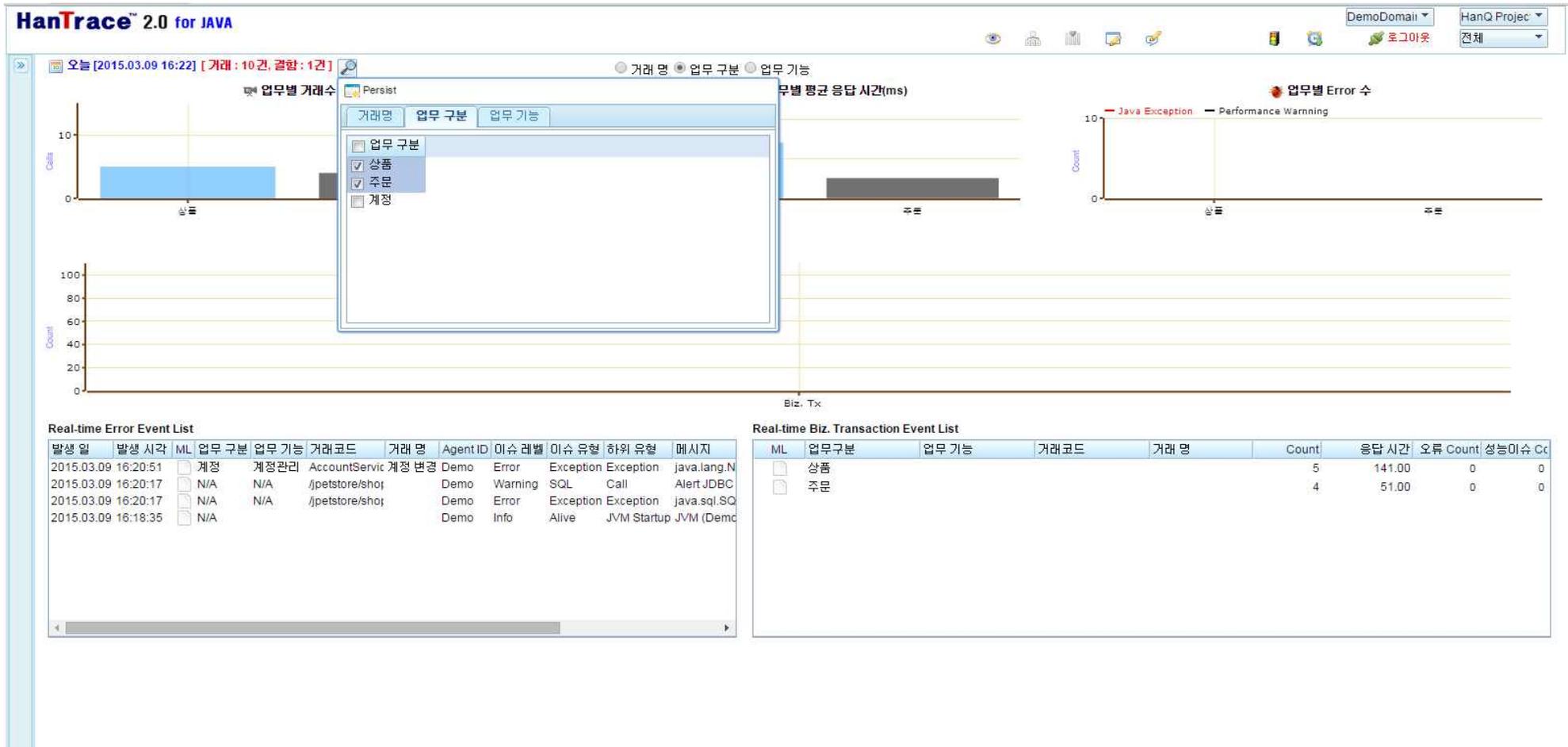
1. 비즈니스 서비스 모니터링 – Transaction 기준

- ◆ Transaction 기준 비즈니스 거래 모니터링
- ◆ Transaction 기준 수행 수, Transaction 기준 응답시간, Transaction 기준 오류 수, 거래 기준 Active 서비스 수 등 실시간 모니터링 제공
- ◆ Real-Time Error Event List / Real-Time Transaction Event List 실시간 모니터링 제공



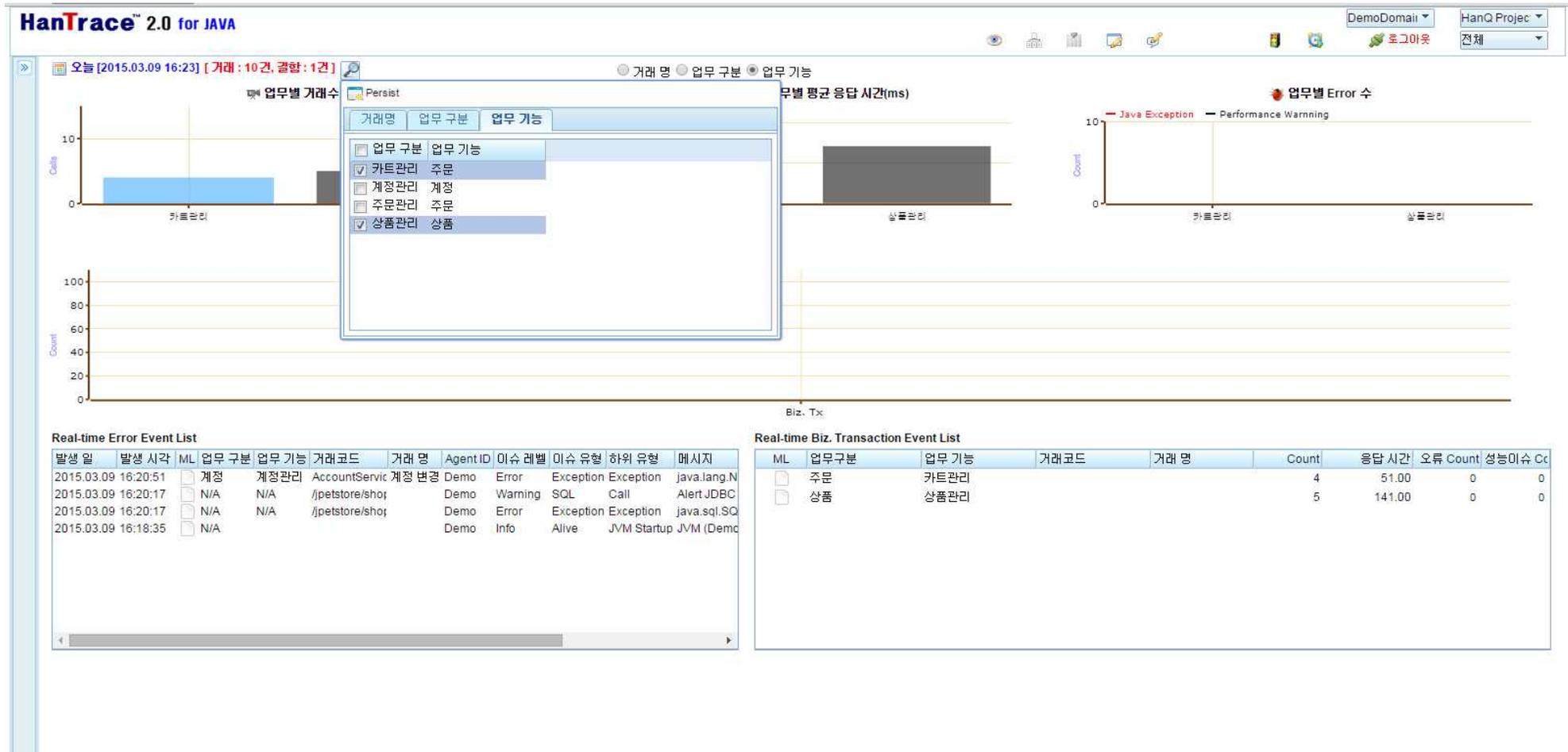
1. 비즈니스 서비스 모니터링 - 업무 구분 기준

- ◆ 업무 구분 기준 비즈니스 거래 모니터링
- ◆ 업무 구분 기준 수행 수, 업무 구분 기준 응답시간, 업무 구분 기준 오류 수, 업무 구분 기준 Active 서비스 수 등 실시간 모니터링 제공



1. 비즈니스 서비스 모니터링 - 업무 기능 기준

- ◆ 업무 기능 기준 비즈니스 거래 모니터링
- ◆ 업무 기능 기준 수행 수, 업무 기능 기준 응답시간, 업무 기능 기준 오류 수, 업무 기능 기준 Active 서비스 수 등 실시간 모니터링 제공



1. 비즈니스 서비스 모니터링 - 업무 관점 이벤트

Real-time Error Event List

발생 일	발생 시각	ML	업무 구분	거래코드	거래 명	Agent ID	이슈 레벨	이슈 유형	하위 유형	메시지
2015.01.18	16:58:08			/petstore/sho		Demo	Error	Exception	Exception	java.sql.SQLException:This function is not supp
2015.01.18	16:58:08			/petstore/sho		Demo	Warning	SQL	Call	Alert JDBC SQL call count too many
2015.01.18	16:48:49					Demo	Info	Alive	JVM Startup	JVM (Demo) is Startup !!!

Problem Level
Error/Warning/Info

Problem 유형
Exception / SQL /
Dump / Leak 등

- ◆ 실시간 Transaction 수행 시 발생하는 Problem 에 대하여 업무/거래 단위로 Event 발행 됨.
- ◆ Error / Warning 에 대하여 어떤 업무/거래에서 발생하였는지 어떤 유형의 Problem 인지 그리고 상세 추적 정보를 제공함으로써 즉시 원인 파악 및 해결을 지원함.

Real-time Biz. Transaction Event List

ML	업무구분	거래코드	거래 명	Count	응답 시간	오류 Count
	N/A	CatalogService.get	카테고리 조회	3	0.0	0

거래 업무 매핑 수정

거래 코드: CatalogService.isItemInStock

거래 명: 상품 카트 체크

업무 분류: N/A

화면 명: DefaultPage

핵심 거래 여부:

공통 거래 여부:

Full 추적 여부:

Biz 거래 여부:

거래 유형: 기본

* 우선 순위: 중

사용 여부:

동록 닫기

거래 목록 및 매핑 관리 화면

거래 코드: /petstore/shop/index.shtml

거래 명: 카테고리 조회

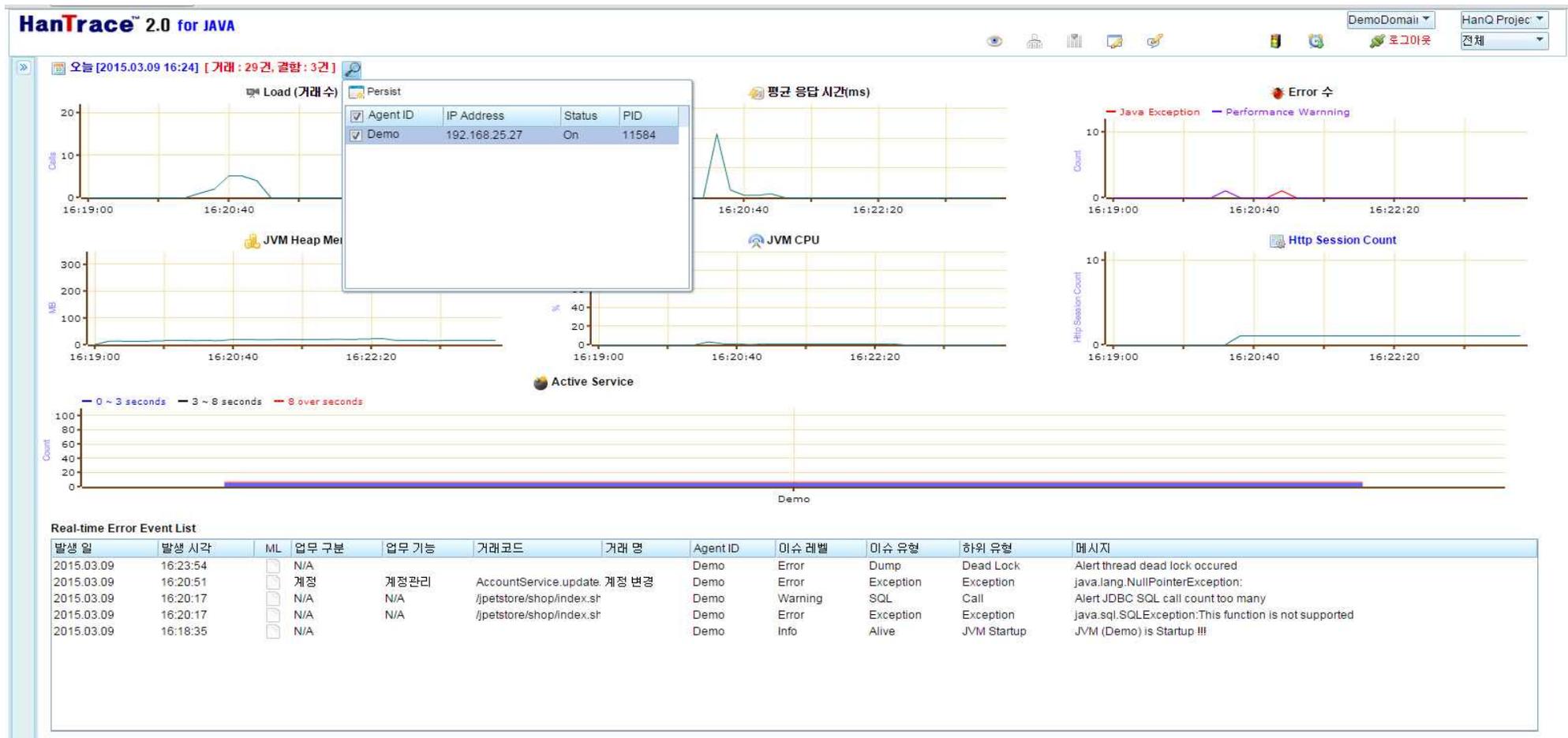
Count: 3

오류 Count: 0

- ◆ 거래 업무 매핑 화면에서 해당 거래에 대하여 Biz 거래 대상인 경우 체크 하면 해당 Biz 거래는 모니터링이 되면 오류에 대하여 상세 정보 제공
- ◆ 거래 업무 매핑 화면에서 해당 거래에 대하여 Full 추적 여부를 체크하면 추적 Depth 관계없이 모두 수집함

1. 비즈니스 서비스 모니터링 - 성능관점

- ◆ Application Performance 모니터링
- ◆ Load / Response Time / Error – Performance Issue / JVM Heap Memory / JVM CPU / Http Session Count / Active Transaction 에 대하여 실시간 모니터링
- ◆ Real-Time Error Event List 를 통해 Application 에서 발생하는 모든 Error / Performance Issue 가 실시간 Alert 된다.



1. 비즈니스 서비스 모니터링 – Error & Performance Issue Event

- ◆ Real-Time Error Event List 에서 각종 오류 정보 실시간 제공
- ◆ Exception, SQL Exception, SQL Performance, JDBC Resource Leak 등 제공

Exception Event Info

...	BM	ML	Age	업무 구분	거래 코드	거래 명	거래 시간	총 SQL 호출 수	최대 SQL 수행시간	최대 SQL Fetch 시간	최대 SQL Row 수	Node 수	Path Count	Object Count	CPU Time	Wa	
...				Dem	주문	/jpetstore/shop/updateCartQuantities.shtml	카드 변경	119	2	0	0	33	0	0	0	15	

유형	수행 시간	From 노드명	To 노드명	Order	Depth	패키지 명	클래스 명	메소드 명	Context	F
...	119	Servlet	ShopServlet	0	0	Root:0:1	192.168.25.20	Demo	http://localhost/jpetstore/sho...	2
...	0 (0/0)	ShopServlet		1	1	java.sql	PreparedStatement	executeQuery()java.sql.ResultSet	SELECT * FROM PRODUCT	
...	0 (0/33)	ShopServlet		2	1	java.sql	PreparedStatement	executeQuery()java.sql.ResultSet	SELECT * FROM PRODUCT	
...	0			3	1	com.ibatis.jpetstore.domain	CartItem	calculateTotal:131	java.sql.SQLException	0

1. 비즈니스 서비스 모니터링 – Dead Lock Event

- ◆ Real-Time Error Event List 에서 Dead Lock Event 상세 정보
- ◆ Dead Lock Thread 상세 Stack 정보 제공

Dead Lock Event Info

- * Dump ID: 5afc2c608adc7d0485aa8bb6dc79cb84
- * 생성일: 2015.02.25 08:20:36
- * Agent ID: Demo

요약

```
[pool-2-thread-2] Id=2538 BLOCKED on java.lang.Object@1b7a8c5 owned by "pool-2-thread-1" Id=2537
at com.ibatis.jpstore.Deadlock.run(Deadlock.java:27)
- blocked on java.lang.Object@1b7a8c5
at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:895)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:918)
at java.lang.Thread.run(Unknown Source)

"pool-2-thread-1" Id=2537 BLOCKED on java.lang.Object@24b943 owned by "pool-2-thread-2" Id=2538
```

내용

```
com.ibatis.jpstore.Deadlock.run(Deadlock.java:27) java.util.concurrent.ThreadPoolExecutor$Worker.runTask
(ThreadPoolExecutor.java:895) java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:918)
java.lang.Thread.run(Unknown Source)
```

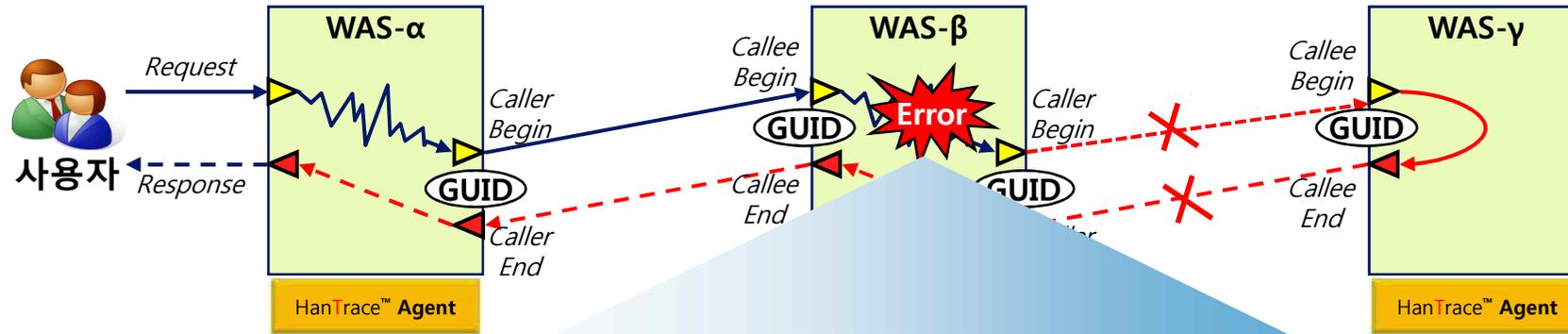
Buttons: 닫기

Real-time Error Event List

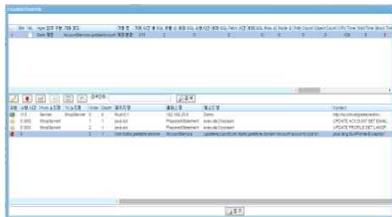
발생 일	발생 시각	ML	업무 구분	업무 기능	거
2015.02.25	08:20:20	N/A			
2015.02.25	08:20:44		계정	계정관리	Ac
2015.02.25	08:20:36	N/A			
2015.02.25	08:20:27		주문	카드관리	/jp
2015.02.25	08:20:27		주문	카드관리	/jp
2015.02.25	08:20:27		주문	카드관리	/jp
2015.02.25	08:20:27		주문	카드관리	/jp
2015.02.25	08:13:59	N/A	N/A	/jpetstore/sho	Demo
2015.02.25	08:13:59	N/A	N/A	/jpetstore/sho	Demo
2015.02.25	08:12:20	N/A			Demo

2. DevOps – Java Runtime Exception

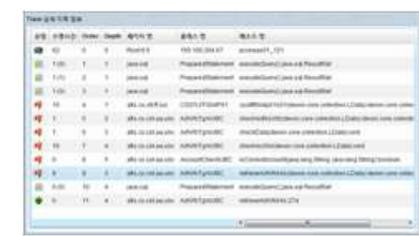
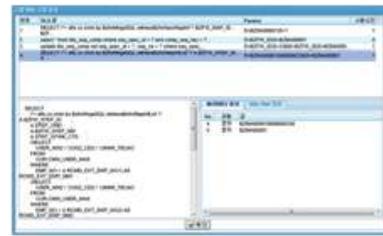
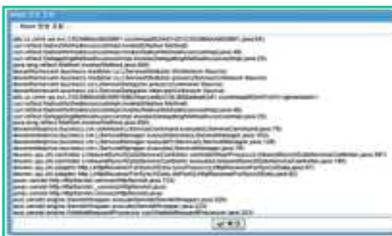
- ◆ 기존 APM 도구들은 Servlet 까지 던져진 Exception 에 대해서만 수집하며, Long Stack 구조에서는 알기도 어렵다.
- ◆ HanTrace 는 비즈니스 처리 모든 구간에서 발생하는 Exception 은 모두 수집되며, Stack 정보 뿐만 아니라 재현이 필요 없이 필요한 모든 정보를 실시간 상세하게 수집한다.



- 1 개별거래 정보
- 2 입력 데이터
- 3 Exception 정보
- 4 Method 입력 값

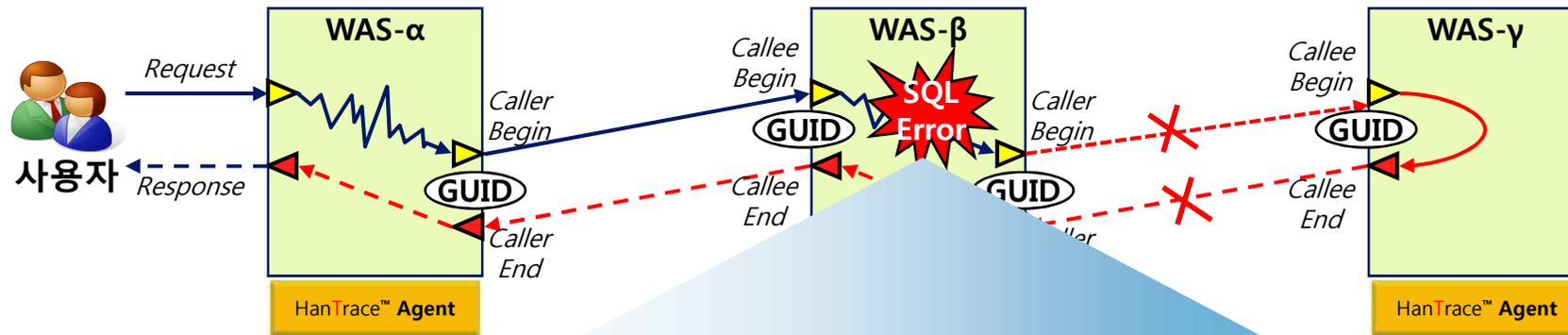


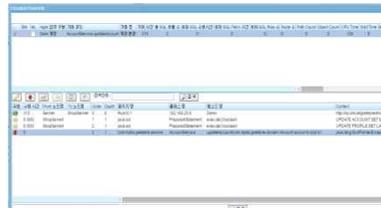
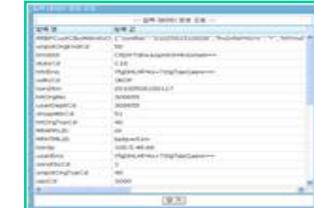
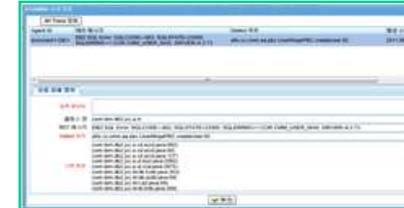
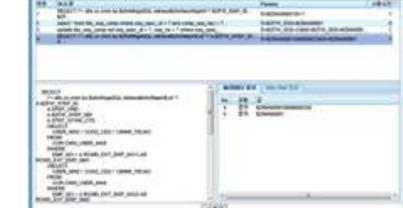
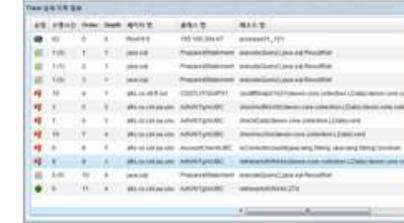
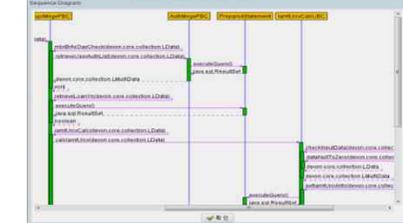
- 5 Stack 정보
- 6 SQL 정보
- 7 결함 발생 소스코드
- 8 추적 상세 이력



2. DevOps – SQL Runtime Exception

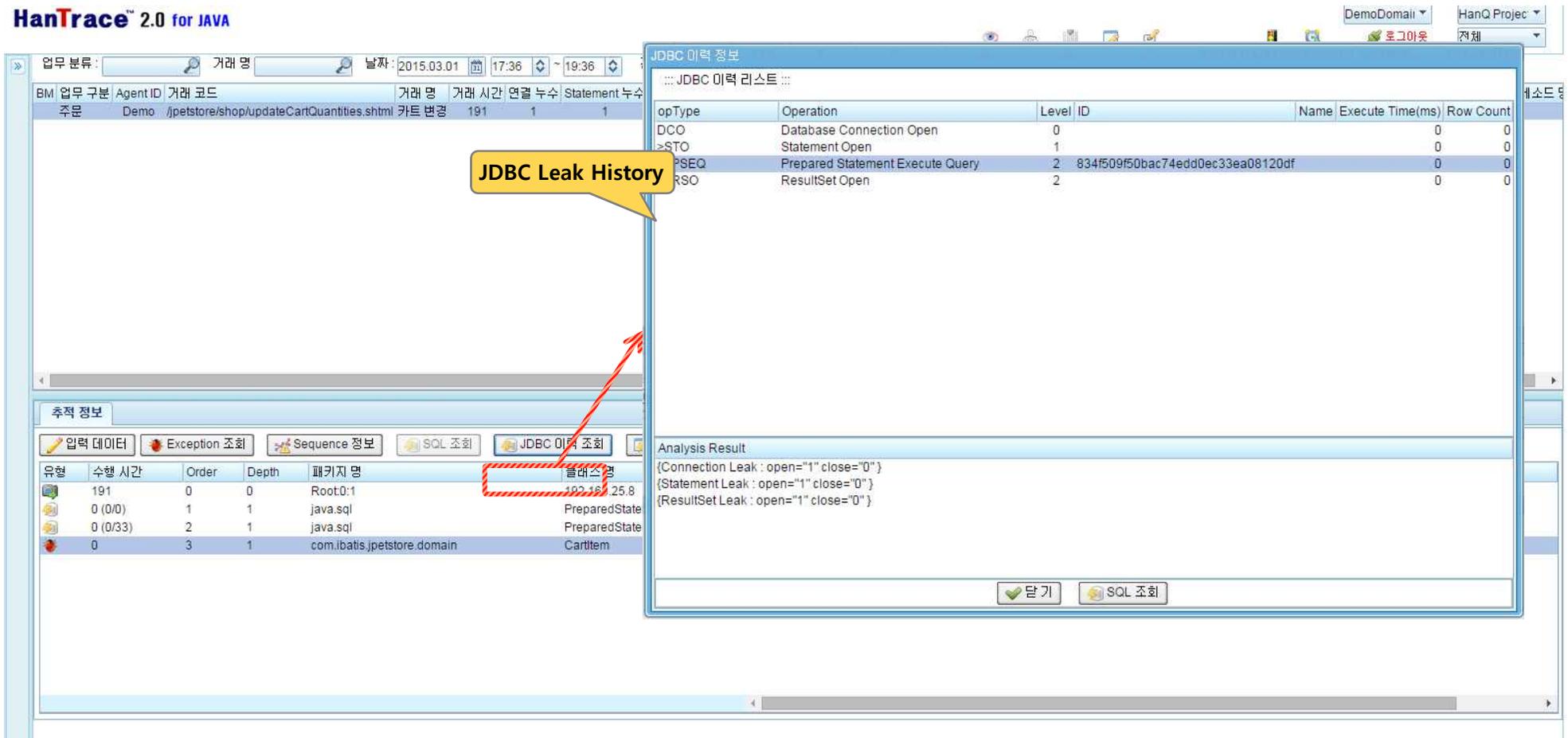
- ◆ 기존 APM 도구들은 SQL Exception 에 대하여 수집하지 않거나 혹은 하더라도 Stack 수준이었음.
- ◆ HanTrace 는 비즈니스 처리 모든 구간에서 발생하는 SQL Exception은 모두 수집되며, Stack 정보 뿐만 아니라 재현이 필요 없이 필요한 모든 정보를 실시간 상세하게 수집한다.



<p>① 개별거래 정보</p> 	<p>② 입력 데이터</p> 	<p>③ Exception 정보</p> 	<p>④ SQL 정보</p> 
<p>⑤ Stack 정보</p> 	<p>⑥ 결함 발생 소스코드</p> 	<p>⑦ 추적 상세 이력</p> 	<p>⑧ 추적 경로 정보</p> 

2. DevOps – JDBC Resource Leak

- ◆ JDBC Resource Leak 에 대하여 발생하면 WAS 성능 저하 발생
- ◆ Leak 발생 지점을 정확하게 파악이 되어야 개발에서 Bug Fix 가 이루어 짐
- ◆ HanTrace 에서는 해당 Leak 발생 소스 코드 라인을 정확하게 분석하여 쉽게 DevOps 가 이루어진다



JDBC 이력 정보

opType	Operation	Level	ID	Name	Execute Time(ms)	Row Count
DCO	Database Connection Open	0			0	0
>STO	Statement Open	1			0	0
>PSEQ	PreparedStatement Execute Query	2	834f509f50bac74edd0ec33ea08120df		0	0
RSO	ResultSet Open	2			0	0

Analysis Result

```
{Connection Leak : open="1" close="0"}
{Statement Leak : open="1" close="0"}
{ResultSet Leak : open="1" close="0"}
```

추적 정보

유형	수행 시간	Order	Depth	패키지 명	클래스 명
	191	0	0	Root0:1	192.168.1.25.8
	0 (0/0)	1	1	java.sql	PreparedStatement
	0 (0/33)	2	1	java.sql	PreparedStatement
	0	3	1	com.ibatis.petstore.domain	CartItem

3. 기타 – User Transaction Snapshot

- ◆ 집중적으로 거래 추적을 통한 Problem 분석 및 거래 매핑을 위해 필요함
- ◆ 관련 사용자 (IP 주소 / User ID / Tx Code / User Define Info) 의 화면에서의 행위를 실시간 분석하여 보여주며, 다른 여러 사용자의 정보를 수집하지 않고 유일하게 해당하는 사용자만 거래 정보를 수집함
- ◆ 거래 레코딩 (Start / Stop) 개념으로 접근되어 초보자도 쉽게 거래를 식별할 수 있으며 , 동시에 빠르게 거래 / 업무를 매핑 할 수 있음
- ◆ SQL 문장 모두 추적 / Method Value 추적 / Dynamic Profile 기능이 제공되어 집중적으로 Debug 할 수 있음

실시간 Snapshot 정보

Snapshot: None | 명: Snapshot_TEST | Key: IP 주소 | 192.168.25.51

BM	ML	Age	업무 구분	거래 코드	거래 명	거래 시간	총 SQL 호출 수	최대 SQL 수행시간	최대 SQL Fetch 시간	...	Node 수	Path Count	Object Count	CPU Time	Wait Tim
Dem	N/A			AccountService.getAccount		62	2	0						62	0
Dem	N/A			/jpetstore/shop/checkout.shtml		296	0	0						62	0
Dem	N/A			CatalogService.isItemInStock		168	3	0						93	0
Dem	N/A			CatalogService.getItem		6	0	0						15	0
Dem	N/A			CatalogService.getItemListByProduct		56	0	0						93	0
Dem	N/A			CatalogService.getProductListByCategory		31	1	0						202	0

관련 사용자 (IP 주소 / User ID / Tx Code / User Define Info) Key 를 선택하여 다양한 경우에 대하여 수집할 수 있다.

거래 업무 매핑 수정

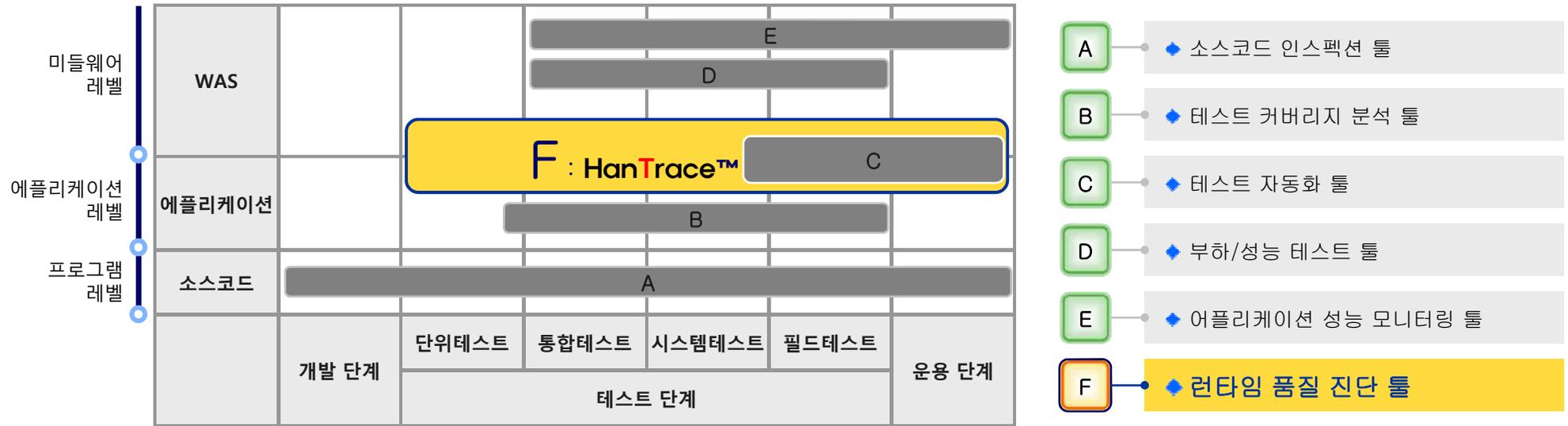
- * 거래 코드: CatalogService.isItemInStock
- * 거래 명: 상품 카드 체크
- 업무 분류: N/A
- 화면 명: DefaultPage
- 핵심 거래 여부:
- 공통 거래 여부:
- Full 추적 여부:
- Biz 거래 여부:
- 거래 유형: 기본
- * 우선 순위: 중
- 사용 여부:

추적 정보

유형	수행 시간	From 노드명	To 노드명	Order	Depth	패키지 명	클래스 명	메소드 명	Context	Param
0		Servlet	ShopServlet	0	0	Root:148:4	192.168.25.51	Demo	http://localhost/jpetstore/sho...	2015.
0		ShopServlet		1	1	com.ibatis.jpetstore.persistence	DaoConfig	getDaoManager():com.ibatis.dao.client.DaoManager		
0		ShopServlet		2	1	com.ibatis.jpetstore.presentation	CartBean	setWorkingItemId(java.lang.String workingItemId):void		
3		ShopServlet		3	1	com.ibatis.jpetstore.presentation	CartBean	addItemToCart():java.lang.String		
0		ShopServlet		4	2	com.ibatis.jpetstore.service	CatalogService	isItemInStock(java.lang.String itemId):boolean		
1		ShopServlet		5	3	com.ibatis.jpetstore.persistence.sqlmapdao	ItemSqlMapDao	isItemInStock(java.lang.String itemId):boolean		
0 (0/1)		ShopServlet		6	4	java.sql	PreparedStatement	execute():boolean	SELECT QTY AS value FROM I...	S=ES
0		ShopServlet		7	2	com.ibatis.jpetstore.service	CatalogService	getItem(java.lang.String itemId):com.ibatis.jpetstore.domain.Item		
5		ShopServlet		8	3	com.ibatis.jpetstore.persistence.sqlmapdao	ItemSqlMapDao	getItem(java.lang.String itemId):com.ibatis.jpetstore.domain.Item		



HanTrace™ 기술



(일반적인) 프로파일 기반 프로그램 테스트 도구

- 정의 : 테스트에 필요한 정보를 테스트 대상 S/W의 소스 코드 또는 바이너리에 삽입하여, 수행시 테스트 정보를 수집하여 S/W 내부의 결함 및 문제점을 진단하는 WhiteBox 테스트 실행 도구이며, SW 내부의 근본적인 문제에 접근 가능함
- 단위(Unit) 거래 프로파일링 : 단위테스트(Unit Test)에서 사용
- 활용 영역 : 런타임 오류 분석, 수행 경로 추적, 단위 모듈 수행 시간 측정, DB 쿼리 오류 분석 등
- 적용 대상 : Dynamic (Run-time 어플리케이션 대상)
- 동작 방식 : (소스 혹은 바이너리) Instrumentation
→ 소스 Instrumentation은 개발자의 부가적인 Cost를 요구하고, 버전의 상이성, 프로파일링의 수준 제약 등 많은 문제를 야기함



HanTrace™: (Advanced) 거래추적기반 런타임 품질진단 도구

- 프로파일 대상 거래 Size 제약 없음 : 단위, 통합, 필드 등 모든 테스트 단계 및 운영유지보수 단계에서 활용 가능
→ Product 환경에서는 Problem 해결용 프로파일 가능
- 활용 영역 : 거래 추적, 결함 원인 분석 및 해결 지원, 거래 및 SQL 레벨 수행성능 최적화 지원, 거래 단위의 Debugging, 이슈 모니터링 지원
- 적용 대상 : Dynamic (Run-time 어플리케이션 대상)
- 동작 방식 : 바이너리 Instrumentation 방식
→ HanTrace™는 Instrumentation Side Effect를 최소화하여, 활용상의 문제를 제거함 (많은 사이트에서 검증됨)

HanTrace™가 타 툴과의 결정적인 차이는? 동적 진단



프로파일러·디버거

- 잘 사용하려면 「풍부한 지식과 경험」이 요구된다
- Java 운용 환경이나 시험 환경에서 사용하려면 부하가 높다

원시코드 정적분석

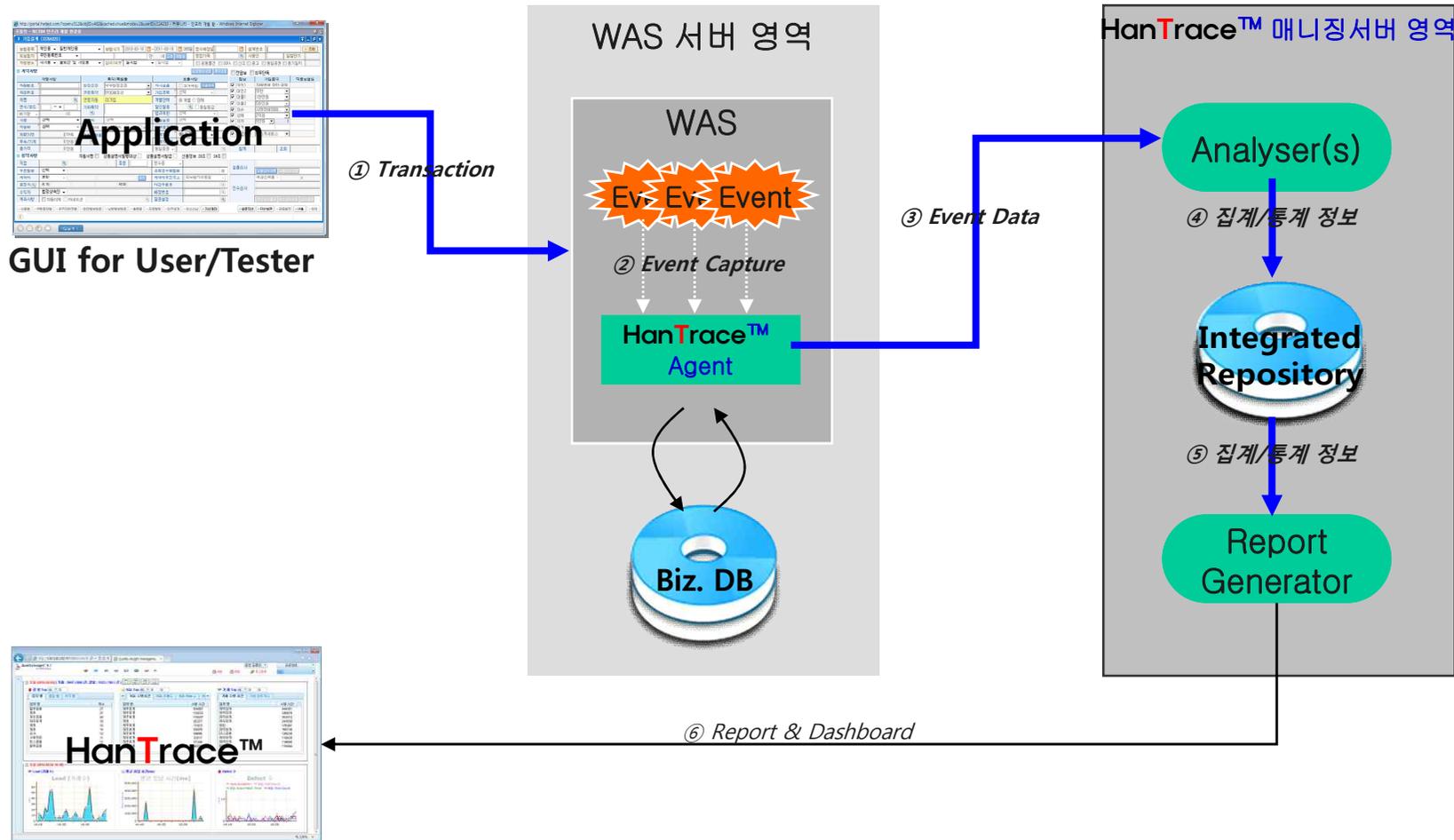
- 중대한 문제는 동작 시켜 보지 않으면 모른다
(Database 액세스, Multi-Thread, Resource Leak)

- 성능 이상의 측정·감지가 주목적이며, 문제해결에는 서툴다
(원인 분석에는 Professional Skill이 필요)
- 어플리케이션 오류(Exception) 분석에 미흡하다

HanTrace™

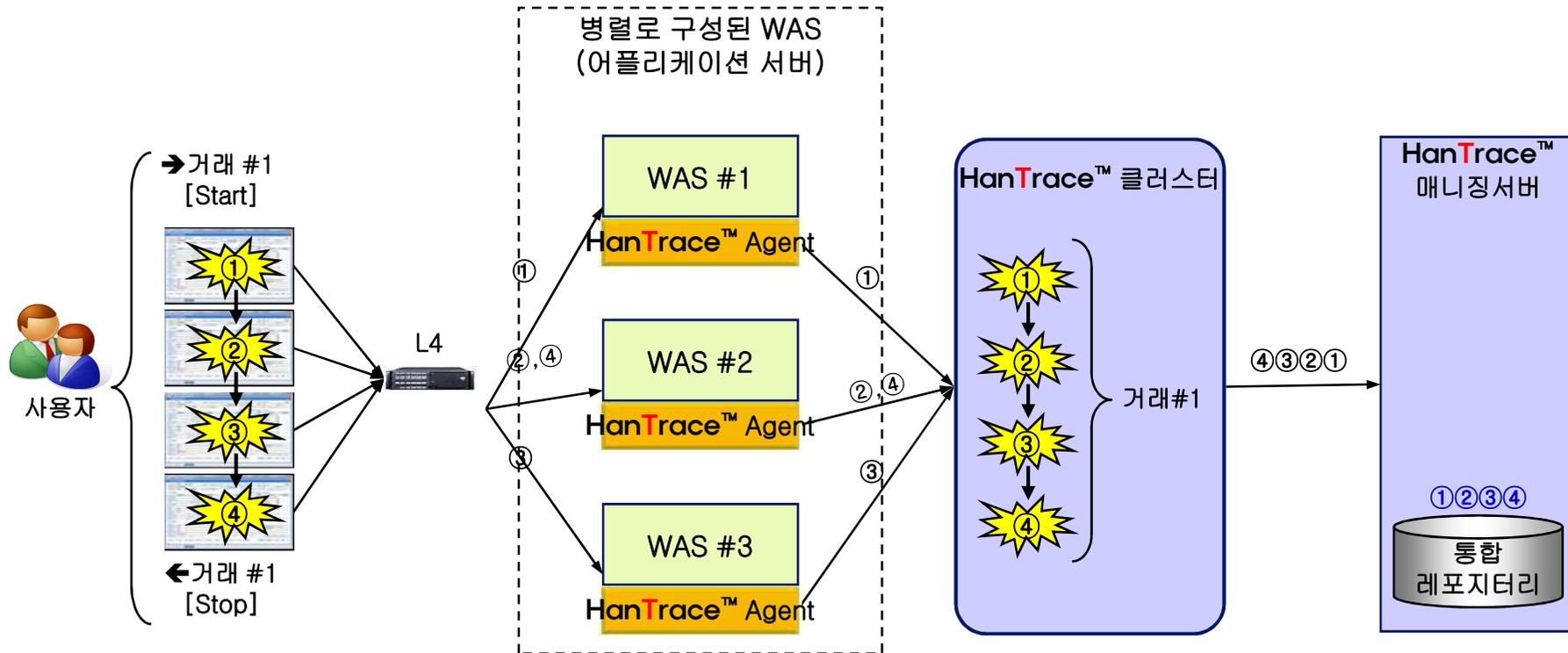
- 「측정·감지」만이 아니고 「진단」까지 실시하여, 문제해결에 지식·경험을 보충해 준다.
- 원시 코드가 아니고, 현실에 동작하고 있는 시스템을 대상으로 한다
- 어플리케이션 오류 및 성능 이슈 분석에 강하다

기능 동작 모형



→ Real-time 프로세스

User Transaction Snapshot



- Java 시스템은 대부분 전체적인 성능을 높이기 위해 WAS를 병렬 처리하는데, 이 때 다수의 사용자가 트랜잭션을 발생시키면 모든 사용자의 거래 내 트랜잭션들이 뒤섞여 WAS에 전송되게 되며, Random하게 WAS 인스턴스로 전송되어, 개별거래의 처리 Order를 식별할 수 없음
- HanTrace™ 클러스터는 각 사용자의 거래 이벤트의 Order를 유지하면서 식별하여 통합레포지터리에 적재함
- 이와 같이 거래 내 이벤트 Ordering이 가능해야 거래별 오류추적이나 업무처리 거래 기반의 수행성능 분석 등의 기능이 가능함

※ WAS Instance가 1개이거나 Local WAS를 사용하는 테스트 단계일 경우는 고려사항이 없음



Thank You

 **코어베이스**

주식회사 코어베이스
02-866-5330
histarpark@corebase.co.kr