ORACLE

Deep Java Diagnostics Can be Low Overhead

Rajagopal Marripalli
Principal Product Manager

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.
The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

1. Production Diagnostics Challenges
2. Application Diagnostics for Java
3. Architecture
4. Platform Support
5. Features
6. Resources
7. Summary

Sample Production Application Problems

1. Key business app is slow → Java/DB? Which Java method or SQL? Is it long running I/O or network calls or locks?
2. Mid-tier CPU utilization is high → don't know which Java classes or methods burning CPU
3. User requests are hanging → which Java method call is causing the hang? What is the reason?
4. GC is running often, consuming resources & slowing application. Is this a memory leak? Which objects are leaking?
5. DBA – Database is slow because of a bad SQL. Which application request is associated with the DB session? What is the associated Java code?

Why Production Diagnostics

- Accelerate problem resolution
- No need to reproduce problems - save money and time
- Empower IT teams in diagnosing application problems

Production Diagnostics Challenges

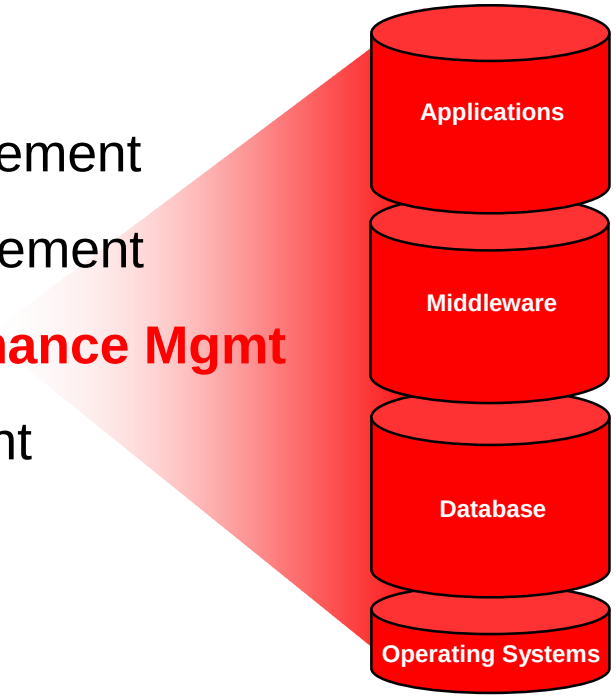
- Limited Production visibility
- Not enough depth to diagnose production issues
- Diagnostic tools not suited for production
- Inability to reproduce problems on other environments
- Hard to isolate culprit tier and component
- Impacts Time to Resolution

Java is a black box

Enterprise Manager – Application Diagnostics for Java (AD4J)

Strengthening APM solution

- Configuration Management
- Service Level Management
- **Application Performance Mgmt**
- Lifecycle Management
- Dashboards

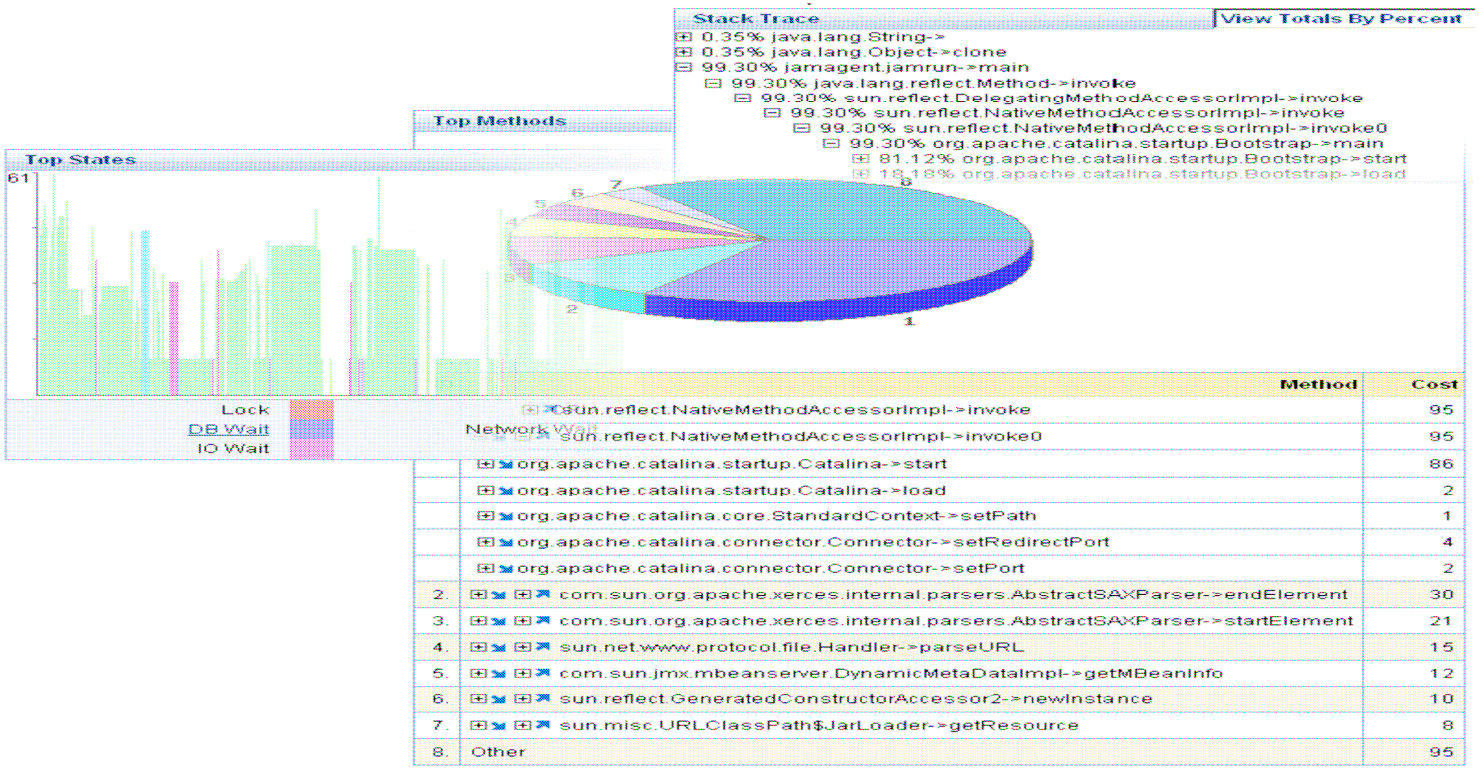


AD4J

- Diagnose problems in production
- View all java activity: instrumentation not needed
- Trace transactions across tiers, into the Oracle DB

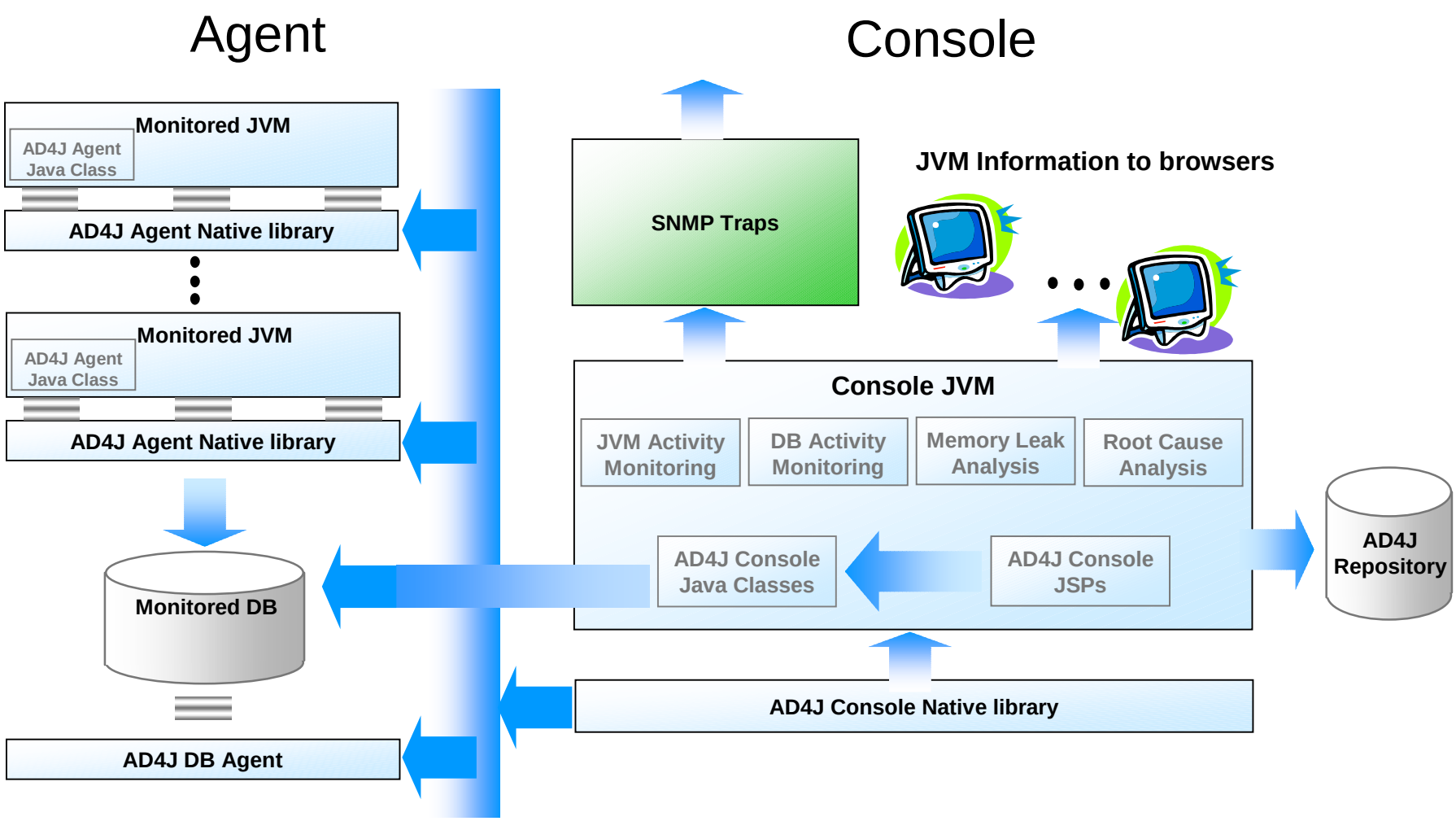
Oracle AD4J – Value Proposition

- Diagnose Issues in Production, not in Test → Reduce Time to Resolution
- “Always-on” monitoring with more informative alerts → Early warnings to minimize outages



10gR4

AD4J Architecture



Platform Support

Standalone Java

IBM WebSphere

ORACLE APPLICATION SERVER

Apache Tomcat

JBoss

SAP

bea

ORACLE DATABASE 8i-11g

1.3.1 1.4.2 1.5.0 1.6.0

Java

HP-UX 11i version 2

SOLARIS

AIX

Microsoft Windows

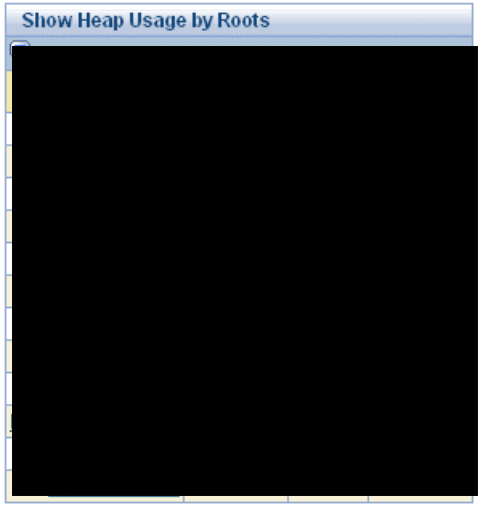
Linux

AD4J – In-depth Application View

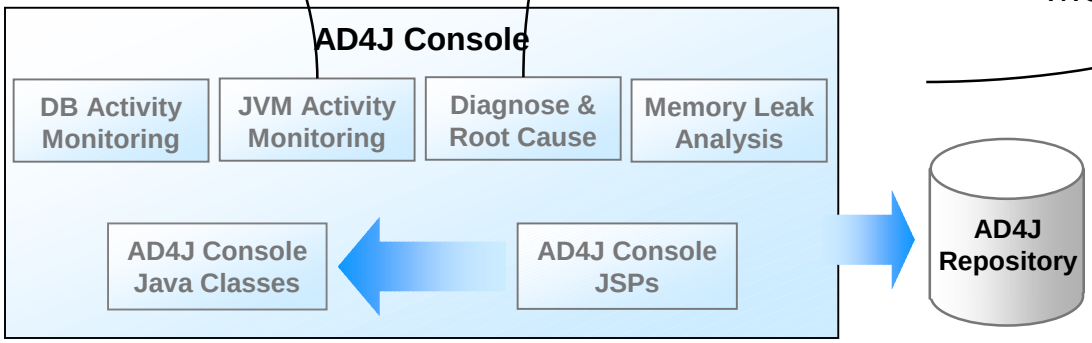
Monitor The Platform 24X7

CPU	Mem	Run	DB	Lock	Net	IO	Obj Wait	Sleep	Total
20	16								16
83	9								11
44	3								50
44	25								12

Diagnose & Root Cause



Memory Analysis



NOC	App Support & QA	DEV

AD4J vs BCI based tools

BCI Based Tools

1. Complicated to install, deploy, use
   - Code instrumentation and server restarts
   - Experts needed to instrument and diagnose
2. High overhead (>20%) – not suitable for production
3. Silo oriented – No visibility from App Server through to DB

Application Diagnostics for Java (AD4J)

1. Very easy to install, deploy, and use
   - No modification of code or server restarts
   - Immediate visibility with no Application expertise required
2. Minimal Overhead (<1%) – Runs in production Servers
3. Transaction Visibility – Follow transactions from AS to Oracle database

Feature-Benefit Summary

Key Features	Benefit
1. Hot deployment, without server restarts or configuration changes	• Quick time to value • Simple to deploy • Easy to use
2. Monitoring & Dashboard	• Low overhead – “always-on” production diagnosis • Events and notification on all performance, load, usage metrics & resource bottlenecks (Java Diagnostics) • Historical trends
3. Real-time visibility into JVM Application state without instrumentation overhead	• Complete Java stack including thread state and Java method/line visibility • View of resource bottlenecks (DB, I/O, CPU, Locks, thread hangs etc) • No need to reproduce problems in test environment. • No application knowledge required
4. Trace in-flight transactions	• View request performance problems in real time • Analyze time spent in each resource bottleneck (extend to each java method)

Feature-Benefit Summary - Continued

Key Features	Benefit
5. Cross-tier trace with Oracle DB	• Cross tier analysis of transactions between DB and middle-tier
6. Differential heap analysis	• Simplify memory growth and leak analysis

Download from OTN

- Download
  - http://www.oracle.com/technology/software/products/oem/htdocs/jade.html
- Documentation
  - download.oracle.com/docs/cd/B16240_01/doc/nav/portal_booklist.htm
  - Installation Guide
  - Administration and Reference Guide

Resources

Oracle Fusion Middleware Packs

www.oracle.com/enterprise_manager/fusion-middleware-management-packs.html

- Data-sheets
- White Papers
- Product Download link

Oracle Technology Network (OTN)

- Application Performance Management - http://www.oracle.com/technology/products/oem/mgmt_solutions/apm.html
- Middleware Management - http://www.oracle.com/technology/products/oem/as_mgmt/index.html

Application Diagnostics for Java (AD4J) Summary

- Production diagnostics
  - Accelerates problem resolution
  - Reduce cost
- Monitor & diagnose production apps with very low overhead
- Identify deep down application problems without any instrumentation and application knowledge
- Trace transactions from Java to the Database and vice versa
- Hot pluggable and heterogeneous including standalone Java apps
- Memory leak detection and analysis

10gR4

Question & Answers

Thank You

- Please complete evaluation form
  - Speaker Name: Rajagopal Marripalli
  - Session Name: Deep Java diagnostics can be low overhead
  - Session Number: Q541
- For further questions, contact Rajagopal Marripalli at rajagopal.marripalli@oracle.com