

Interstage / ENdoSnipe 動作検証結果報告書(Windows)

平成21年6月5日

株式会社ヒューマンクレスト

品質コンサルティング部

第1.0版

動作検証結果報告書 目次

■ 1. 本プロジェクトの目的及び方針	・・・P 2
■ 2. 検証環境	・・・P 3
■ 3. 検証内容	・・・P 5
■ 4. 検証結果	・・・P 6
◆ メモリリーク	・・・P 7
◆ マルチプロセス	・・・P 11
◆ 負荷試験	・・・P 14
◆ ENdoSnipeの動作有無によるリソース使用量比較	・・・P 18
■ 5. 「見える化」でシステムの問題解決を効率化する	・・・P 19
■ 6. 問い合わせ先	・・・P 20
■ 別紙 検証内容	・・・P 21

1. 本プロジェクトの目的及び方針

本プロジェクトは以下の目的及び基本方針に基づいて実施する。

■ 1. 本プロジェクト発足の経緯

- ◆ ソフトウェア事業本部・サポート技術統括本部様における、ENdoSnipe動作確認作業において、高評価をいただき、Interstageパートナー登録の運びとなった。
- ◆ Interstageパートナー登録にあたって、ENdoSnipeの機能が正しく動作するかを再度確認するために、今回のプロジェクトが発足した。

■ 2. 本プロジェクトの実施目的

- ◆ Interstage Application Server上で動作するJavaアプリケーションに対し、ENdoSnipeの機能が正しく動作するかを確認する。

■ 3. 基本方針

- ◆ 上記実施目的を達成するために、弊社にて準備したJavaアプリケーションを使用し、動作検証を実施する。

■ 4. 本プロジェクトの達成目標

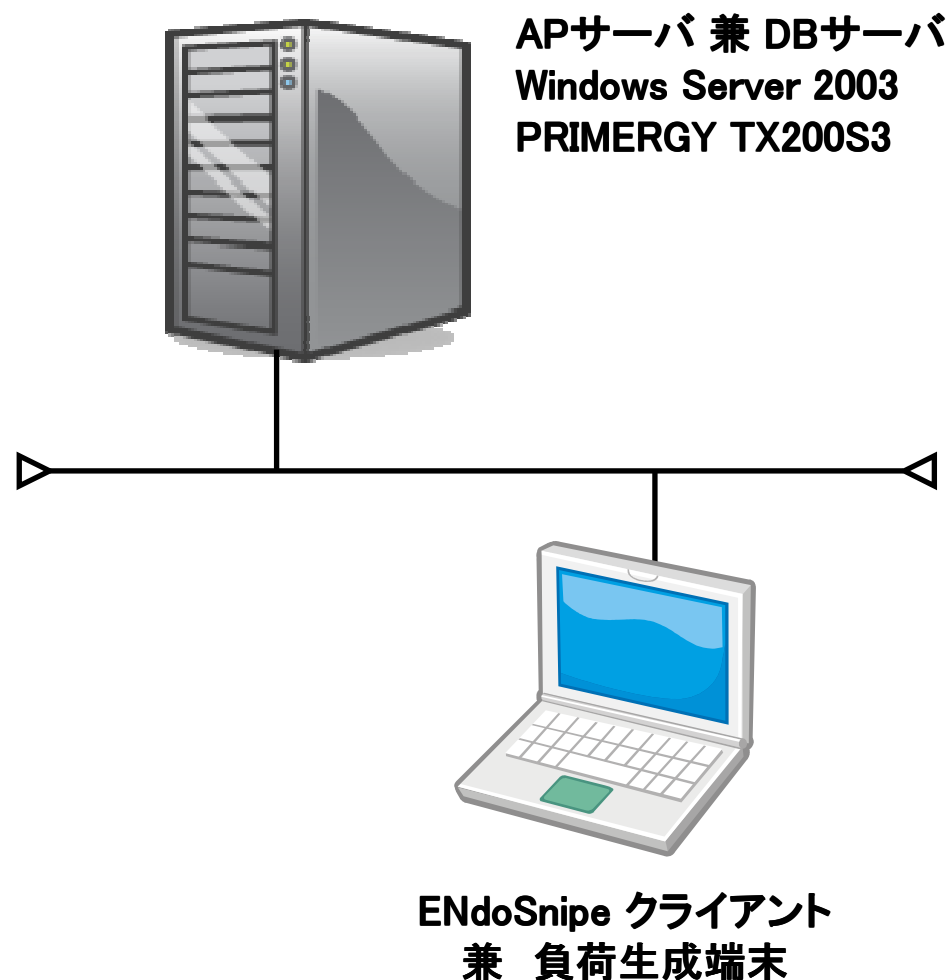
- ◆ ENdoSnipeの機能が正しく動作することを明らかにする。

■ 5. 検証のポイント

- ◆ Interstage Application Server上で動作するJavaアプリケーションに対し、ENdoSnipeの機能が正しく動作すること。
- ◆ マルチコンテナ(マルチプロセス)で動作するJavaアプリケーションに対し、ENdoSnipeの機能が正しく動作すること。
- ◆ Symfowareにおいても、ENdoSnipeの各機能が正常に動作すること。

2. 検証環境 (1/2)

■ 1. ハードウェア構成



マシン	仕様	
APサーバ 兼 DBサーバ PRIMERGY TX200S3	CPU	QuadXeon X5355 2.66GHz/2x4MB × 2CPU
	MEM	2GB
	HDD	147GB × 3
ENdoSnipe クライアント	CPU	Pentium M 1.7GHz × 1
	MEM	2GB
ThinkPad X31	HDD	70GB × 1

2. 検証環境 (2/2)

■ 2. ソフトウェア構成

マシン名	機種	OS	Application	DB	Java Ver
評価用サーバ APサーバ 兼 DBサーバ	PRIMERGY TX200S3	Windows Server 2003, Enterprise Edition	・Interstage Application Server Enterprise Edition V9.1.0 ・ENdoSnipeエージェント(Javelin) ・検証プログラム(Javaアプリ)	Symfoware Server Enterprise Edition V9.1.1	1.5.0_13 (Fujitsu)
評価用クライアント ENdoSnipeクライアント 兼 負荷生成端末	ThinkPad X31	Windows XP SP3	・Eclipse 3.4.2 ・ENdoSnipeプラグイン ・Jmeter(負荷生成ツール)	—	1.6.0_13

※Java verについて Interstageサーバは富士通JDKのバージョンを、ENdoSnipeクライアントは、SunJDKのバージョンを記載

3. 検証内容

■以下の観点で検証を実施

機 能	観 点
BottleneckEye	アプリケーションの実行状況を、クラス図表示、グラフ表示できること
	メソッドの呼び出し回数、平均処理時間などの統計情報を参照できること
PerformanceDoctor	メモリーリークの発生を検知できること
	DatabaseのSQLを解析できること
	Databaseの実行計画を解析できること
	スレッドのCPU占有、スリープを検知できること
ArrowVision	アプリケーションの処理の流れをシーケンスダイアグラムとして表示できること
	Javaなどの複数ログの情報が1つのシーケンス図で表示できること
	ボトルネック解析として、時間が掛かった処理部分を表示できること

その他の観点として、以下のものを実施。

- ・マルチプロセスで動作しているアプリの、プロセス毎の情報を取得できること
- ・Serverに負荷が掛かった状態でも正常に動作ログが出力されること
- ・ENdoSnipeの動作有無によるリソース差異を確認する

4. 検証結果 (1/13)

Interstage Application Server上で動作するJavaアプリケーションに対し、ENdoSnipeの機能が正しく動作することが確認できた。(※1)

結果として、代表的なものを以降に示す。

- ◆ メモリリーク
メモリリークの発生を検知できること
- ◆ マルチプロセス
マルチプロセスで動作しているJavaアプリケーションに対して、プロセス毎の情報を取得できること
- ◆ 負荷試験
Serverに負荷が掛かった状態でも正常に動作ログが出力されること
- ◆ ENdoSnipeの動作有無によるリソース使用量比較
ENdoSnipeの動作有無によるリソース差異を確認する

(※1)ただし、以下については確認できていない。

- ・スリープ回数、時間は測定不能のため、スリープ時間の診断はできない。(ENdoSnipe次期バージョン(Ver4.12009/09リリース予定)で対応予定)
- ただし、アイドル時間の診断ルールを用いて、CPUを使用していない時間を診断する事で代用が可能。
- ・Symfoware Serverのアクセスプラン(実行計画情報)の取得は現在のバージョンでは未対応。(ENdoSnipe次期バージョン(Ver4.12009/09リリース予定)で対応予定)
- ◆ DB種別による差異

機 能 (SQL関連)	Symfoware Server	Oracle
DataBaseのSQLを解析できること	○	○
Databaseの実行計画を解析できること	—	○

4. 検証結果 (2/13)

●メモリリーク

▪BottleneckEye画面

/endosnipe-demo
/employee/employeeList.html(131338196)
/employee/employeeSearch.html(121914816)
/employee/employeeConfirm.html(120321406)
/employee/employeeEdit.html(115616731)
/index.html(10788631)
/foreach3/foreach3List.html(18838143)
/validator/validator.html(17255857)
/postback/postback.html(9423400)
/foreach/foreachList.html(7921625)
/converter/converter.html(6231000)

examples.jsf
logic.impl.EmployeeLogicImpl(154531495)
sample.MultiAccessThread(1506215062000)
logic.impl.EmployeeLogicImpl\$Stopper(60006000000)
dao.EmployeeDtoDao(22172291)
dao.DepartmentDtoDao(18131379)
logic.impl.EmployeeLogicImpl\$1(547432439)
util.RequestDumpFilter(3900850)
util.RequestDumpUtil(3440336)
action.impl.EmployeeEditActionImpl(620036)
action.impl.EmployeeListActionImpl(480003)

jdbc:syconfd:///PARTNER.SERVLETuser=administrator;password=root0000;schema=SERV...
SELECT DEPT.deptno, DEPT.dname, DEPT.loc, DEPT.versionno FROM DEPT(1720571)
select 1(2030653)
select count(*) from emp e left outer join dept d on e.deptno = d.deptno (1250522)
select dname from dept where deptno = ?(1250587)
select empno,ename,job,mgr,hiredate, sal,comm,e.deptno,d.dname from emp e left o...(1410665)
select empno,ename,job,mgr,hiredate, sal,comm,e.deptno,e.versionno,d.dname from ...(2810859)

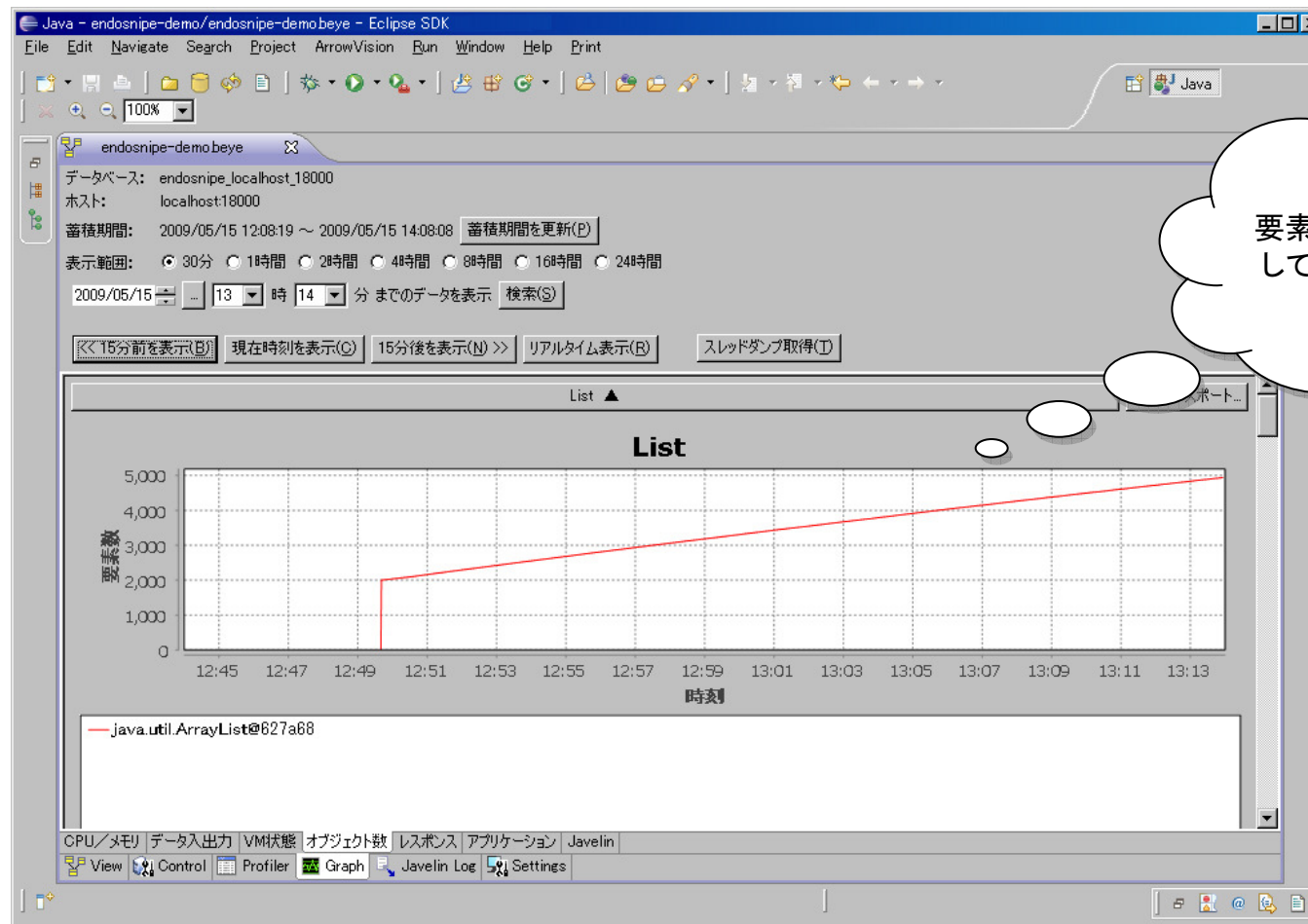
メモリリーク検知時、関連するクラス部分が点滅

Time	Node	Name	Duration
2009/05/19 15:41:05.756	EmployeeEditInitActionIm...	initialize	5047
2009/05/19 15:41:05.756	EmployeeLogicImpl	getEmployeeDto	5016
2009/05/19 15:41:05.756	EmployeeLogicImpl	proceed	5016
2009/05/19 15:41:05.756	EmployeeLogicImpl	proceed	5016
2009/05/19 15:41:05.756	EmployeeLogicImpl	getEmployeeDto	5016

4. 検証結果 (3/13)

●メモリリーク

▪BottleneckEye画面

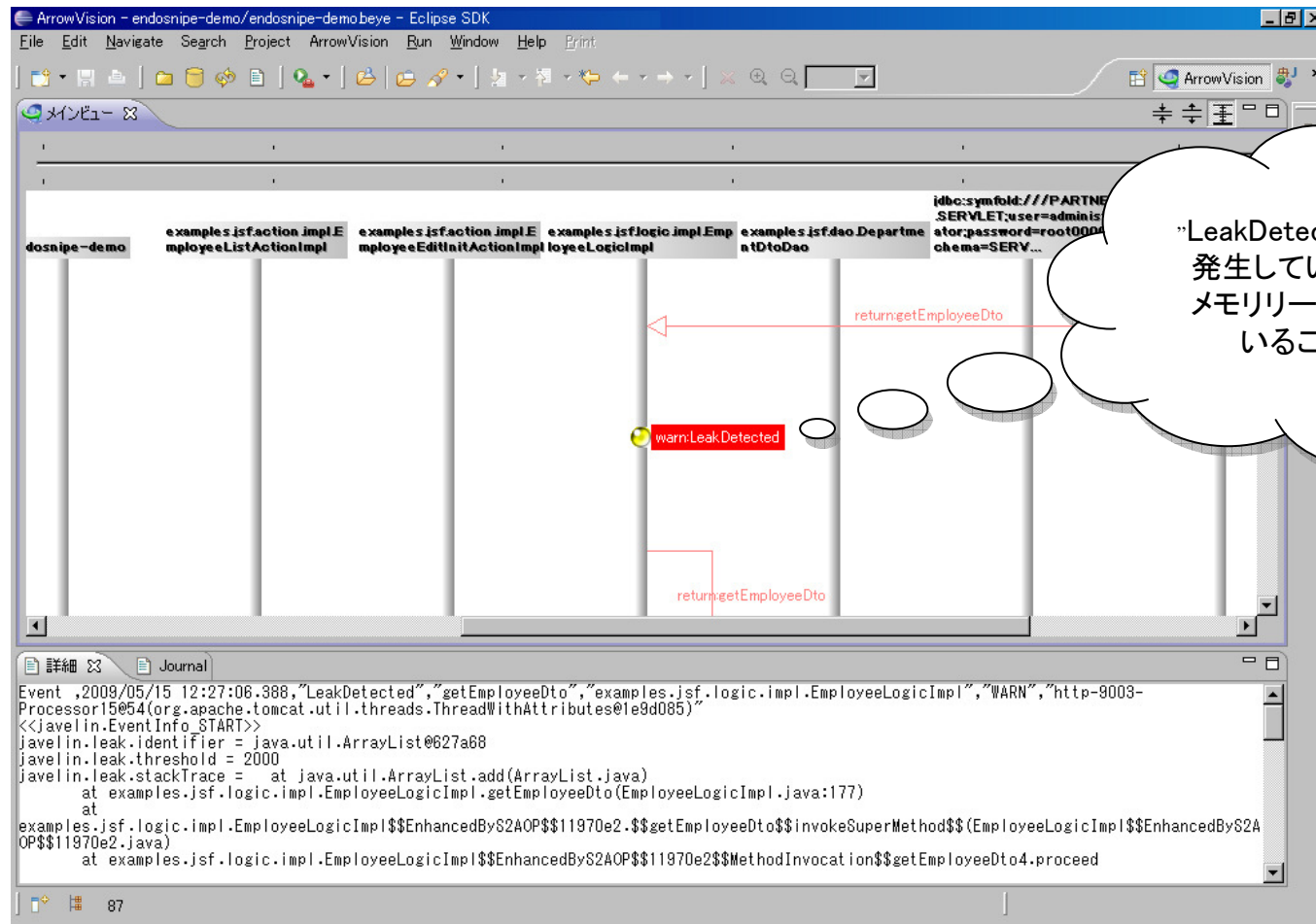


要素のインスタンスが存在
していることがわかります

4. 検証結果 (4/13)

●メモリリーク

▪ ArrowVision画面



4. 検証結果 (5/13)

●メモリリーク

■PerformanceDoctor画面

ID	概要	重要度	クラス	メソッド
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:500msec、検出値:5.031 msec)	ERROR	examples.jsf.logic.impl.EmployeeLog...	proceed
COD.MTRC.METHOD_CPU	メソッドの開始から終了までの間のCPU時間が閾値を超えています。(閾値:200ms...	ERROR	examples.jsf.logic.impl.EmployeeLog...	proceed
examples.jsf.action.impl.EmployeeEditInit...				
initialize				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:500msec、検出値:5.031 msec)	ERROR	examples.jsf.action.impl.EmployeeE...	initializ
COD.MTRC.METHOD_CPU	メソッドの開始から終了までの間のCPU時間が閾値を超えています。(閾値:200ms...	ERROR	examples.jsf.action.impl.EmployeeE...	initializ
proceed				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:500msec、検出値:5.031 msec)	ERROR	examples.jsf.action.impl.EmployeeE...	proceed
COD.MTRC.METHOD_CPU	メソッドの開始から終了までの間のCPU時間が閾値を超えています。(閾値:200ms...	ERROR	examples.jsf.action.impl.EmployeeE...	proceed
/endosnipe-demo				
/employee/employeeEdit.html				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:500msec、検出値:5.328 msec)	ERROR	/endosnipe-demo	/emp
COD.MTRC.METHOD_CPU	メソッドの開始から終了までの間のCPU時間が閾値を超えています。(閾値:200ms...	ERROR	/endosnipe-demo	/emp
/employee/employeeList.html				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:500msec、検出値:5.328 msec)	ERROR	/endosnipe-demo	/emp
COD.MTRC.METHOD_CPU	メソッドの開始から終了までの間のCPU時間が閾値を超えています。(閾値:200ms...	ERROR	/endosnipe-demo	/emp
examples.jsf.util.RequestDumpFilter				
doFilter				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:500msec、検出値:5.531 msec)	ERROR	examples.jsf.util.RequestDumpFilter	doFilter
COD.MTRC.METHOD_CPU	メソッドの開始から終了までの間のCPU時間が閾値を超えています。(閾値:200ms...	ERROR	examples.jsf.util.RequestDumpFilter	doFilter
警告				
examples.jsf.util.RequestDumpUtil				
dumpRequestParameters				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:200msec、検出値:203msec)	WARN	examples.jsf.util.RequestDumpUtil	dumpR
examples.jsf.util.RequestDumpFilter				
dumpRequest				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:200msec、検出値:203msec)	WARN	examples.jsf.util.RequestDumpFilter	dumpR
examples.jsf.logic.impl.EmployeeLogicImpl				
getEmployeeDto				
COD.THRD.MEM_LEAK	メモリリークが発生している可能性があります。(閾値:サイズ:2000、対象コレクション...	WARN	examples.jsf.logic.impl.EmployeeLog...	getEmp
examples.jsf.util.RequestDumpUtil				
dumpRequestParameters				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:100msec、検出値:172msec)	INFO	examples.jsf.util.RequestDumpUtil	dumpR
examples.jsf.util.RequestDumpFilter				
dumpRequest				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:100msec、検出値:172msec)	INFO	examples.jsf.util.RequestDumpFilter	dumpR
doFilter				
COD.MTRC.METHOD_TAT	メソッドのTATが閾値を超えています。(閾値:100msec、検出値:198msec)	INFO	examples.jsf.util.RequestDumpFilter	doFilter

メモリリークが発生
していることが判明

ID	概 要	重要度
COD.THRD.MEM_LEAK	メモリリークが発生している可能性があります。 (閾値:サイズ:2000、対象コレクション識別子:java.util.ArrayList@627a68、オブジェクトサイズ:nullbytes、スレッド:http-9003-Processor15@54(org.apache.tomcat.util.threads.ThreadWithAttributes@1e9d085))	WARN

4. 検証結果 (6/13)

●マルチコンテナ(マルチプロセス)

▪BottleneckEye画面

↓ プロセス2側の画面
(10.20.105.11:18001)

それぞれのプロセスで
問題が発生したことがわかります

↑ プロセス1側の画面
(10.20.105.11:18000)

4. 検証結果 (8/13)

●マルチコンテナ(マルチプロセス)

・PerformanceDoctor画面

↓ プロセス2側の画面
(10.20.105.11:18001)

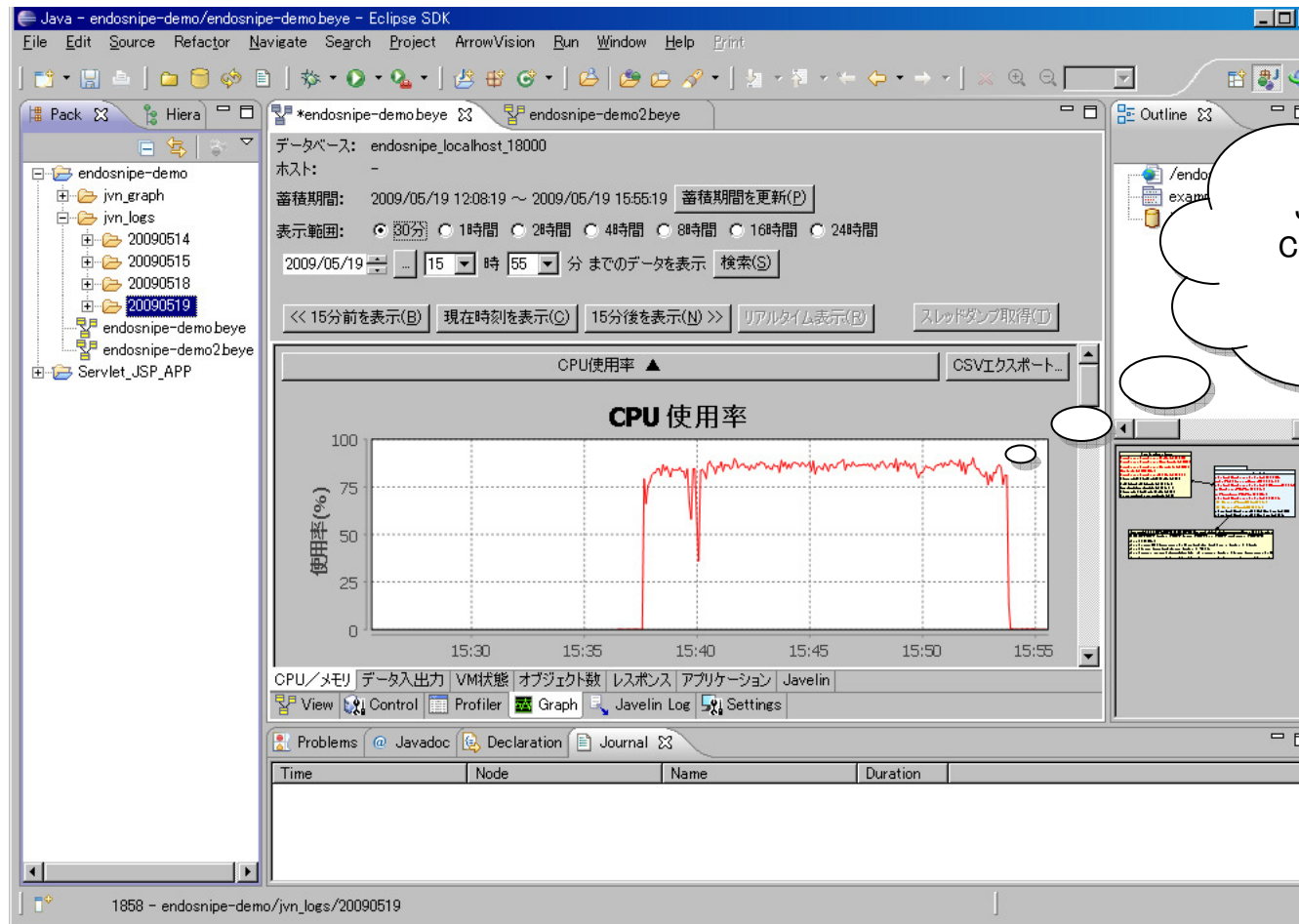
↑ プロセス1側の画面
(10.20.105.11:18000)

それぞれのプロセスで
問題が発生したことがわかります

4. 検証結果 (9/13)

● 負荷試験

▪ BottleneckEye画面



4. 検証結果 (10/13)

● 負荷試験

▪ BottleneckEye画面

The screenshot displays the BottleneckEye application interface. The main window shows a call graph with nodes representing different parts of the application, such as `/endosnipe-demo`, `examples.jsf`, and `jdbc:symfoid:///PARTNER.SERVLET`. The `examples.jsf` node is highlighted, showing its internal structure and associated methods. The `jdbc:symfoid:///PARTNER.SERVLET` node is also highlighted, showing its internal structure and associated methods. The bottom panel displays a table of log entries.

Time	Node	Name	Duration
2009/05/19 15:41:05.756	EmployeeEditInitActionIm...	initialize	5047
2009/05/19 15:41:05.756	EmployeeLogicImpl	getEmployeeDto	5016
2009/05/19 15:41:05.756	EmployeeLogicImpl	proceed	5016
2009/05/19 15:41:05.756	EmployeeLogicImpl	proceed	5016
2009/05/19 15:41:05.756	EmployeeLogicImpl	getEmployeeDto	5016

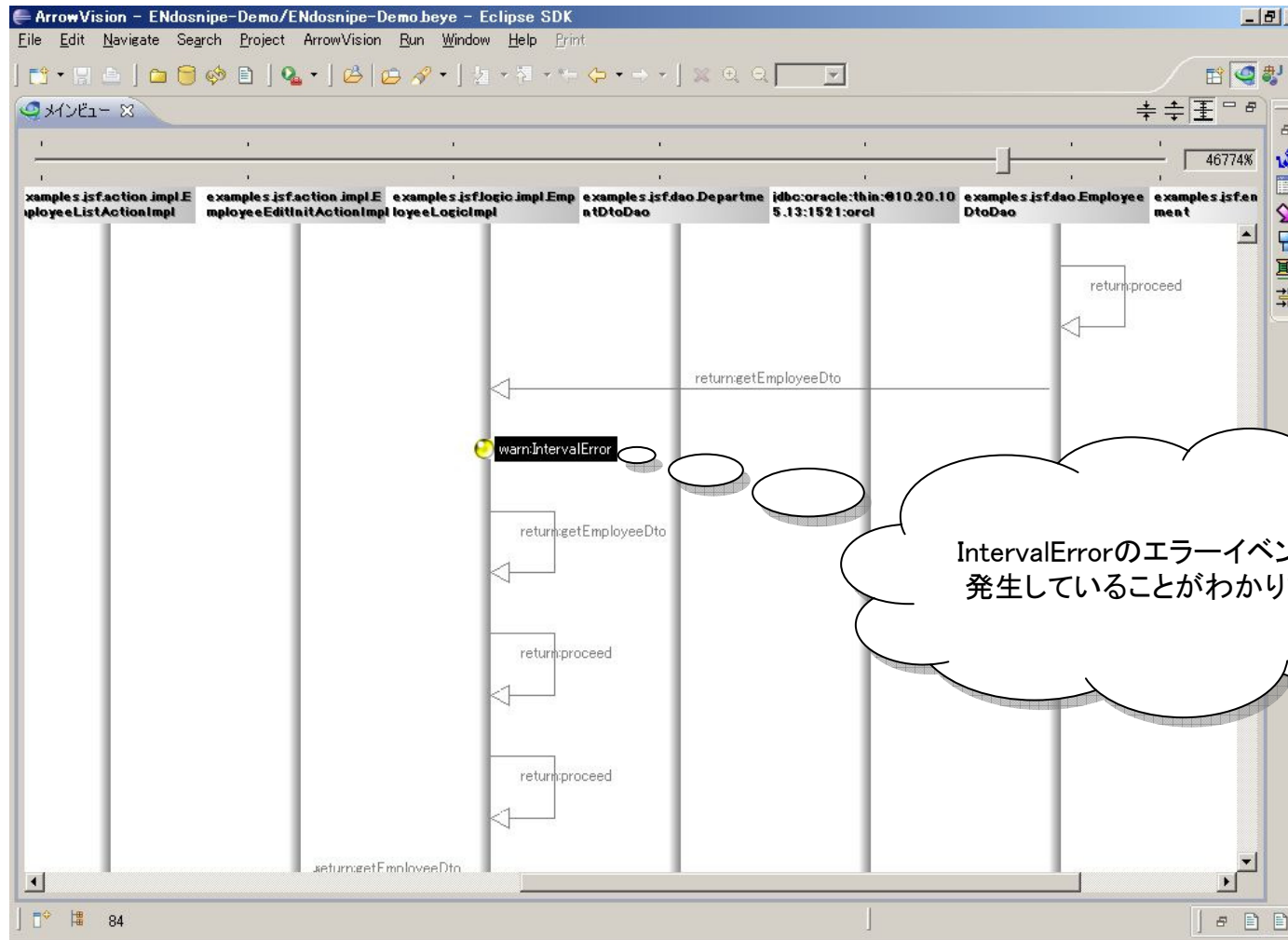
負荷が掛かっている状態で
初期化メソッドを2回呼出す
処理を実行

負荷が掛かっているでも、
初期化を複数回行っている
状態を取得できています

4. 検証結果 (11/13)

● 負荷試験

▪ ArrowVision画面



4. 検証結果 (12/13)

● 負荷試験

・ PerformanceDoctor画面

ID	概要	重要度
エラー		
examples.jsf.logic.impl.EmployeeLogicImpl getEmployeeDto		
警告		
情報		
jdbcoracle:thin@10.20.105.13:1521:orcl		
select empno,ename,job,mgr,hiredate, s...		
DBA.MTRC.SQL_TAT	SQLの実行時間が閾値を超えています。(閾値:1msec、検出値:1msec)	INFO
SQLPLAN.ORA_JOIN	テーブル結合を閾値以上実行している可能性があります。(閾値:1回、検出値:1...	INFO
SQLPLAN.ORA_COST	実行時のコストが閾値を超えています。(閾値:1、検出値:2)	INFO
SELECT DEPT.deptno, DEPT.dname, D...		
DBA.MTRC.SQL_TAT	SQLの実行時間が閾値を超えています。(閾値:1msec、検出値:1msec)	INFO
SQLPLAN.ORA_FULLSCAN	テーブルのフルスキャンが発生している可能性があります。(テーブル名:DEPT、Dura...	INFO
SQLPLAN.ORA_COST	実行時のコストが閾値を超えています。(閾値:1、検出値:3)	INFO

初期化が複数回
発生していることが
わかります

ID	概要	重要度
LIB.SEASAR2.INIT_DUPLICATION	S2コンテナの初期化を複数回行っている可能性があります。 (閾値:100msec、検出値:0msec、クラス名:org.seasar.framework.container.factory.S2ContainerFactory、メソッド名:create)	ERROR

4. 検証結果 (13/13)

●ENdoSnipeの動作有無によるリソース使用量比較

	物理メモリ空き容量	CPU使用率
ENdoSnipe適用下	1,098,436 [Kbytes]	1.0%
ENdoSnipe非適用下	1,123,236 [KBytes]	1.0%

※ENdoSnipe適用下 : Javelinがログを取得する状態で、「数秒間のスリープ発生」をさせる

※ENdoSnipe非適用下 : Javelinがログを取得しない状態で、「数秒間のスリープ発生」をさせる

物理メモリ空き容量は多少差があるものの、CPU使用率はともに1%程度であることから、ENdoSnipeの動作有無によるリソース使用量に大きな違いは見られない。

よって、アプリケーション実行時に致命的なパフォーマンス低下は発生しないと考えられる。

5. 「見える化」でシステムの問題解決を効率化する

解析

ArrowVision

システムの開発・保守効率を向上させたい！

▶ ArrowVision とは

ArrowVisionは、アプリケーションのソースコードを一切変更することなく、その動作をシーケンスダイアグラムによって「見える化」します。

- ・システムの処理の流れを、自動的にシーケンスダイアグラムにします。
- ・Javaシステム、ネットワークなど、複数のログを1つのシーケンス図にまとめて表示します。
- ・時間の掛かる処理がすぐにわかるため、大量のログを調査する必要がありません。

診断

PerformanceDoctor

手早く品質を診断したい！

▶ PerformanceDoctor とは

PerformanceDoctorは、取得したログからシステムに潜む性能問題の芽を「見える化」します。

- ・性能問題の原因となる処理を自動的に見つけます。
- ・スレッドのCPU占有や頻繁なスリープなども見つけます。

監視

BottleneckEye

予期せず発生した障害をすぐに解決したい！

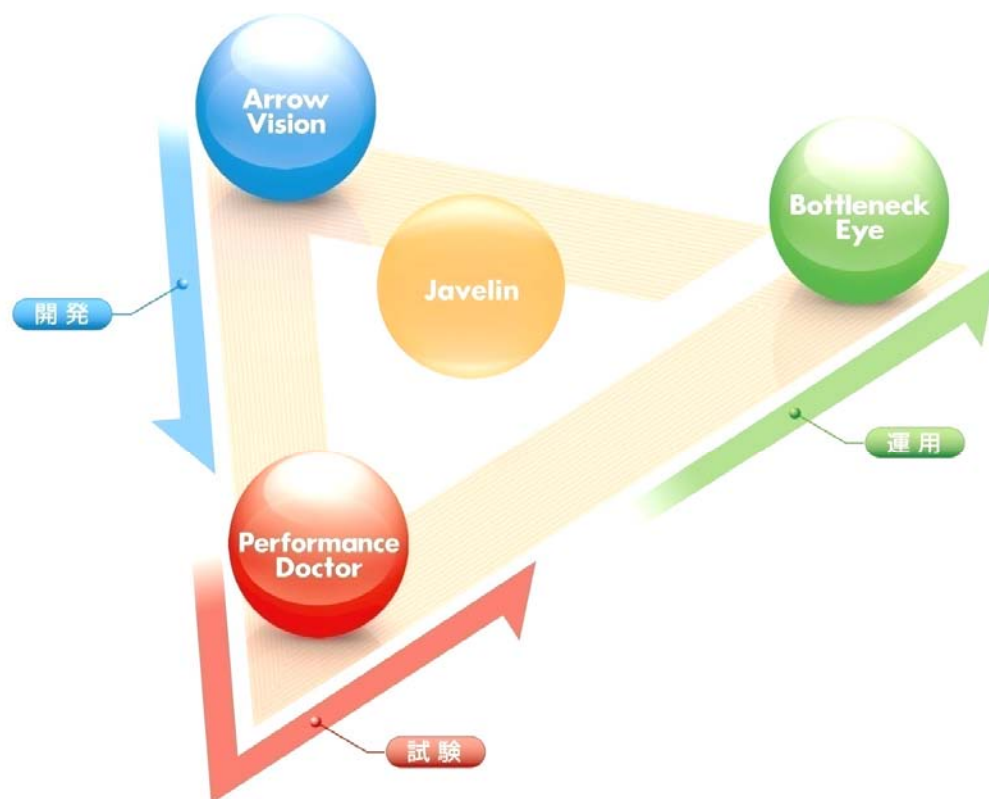
▶ BottleneckEye とは

BottleneckEyeは、運用中のシステムを監視し、「アプリケーションの構造」と「システムのボトルネック」を「見える化」します。

- ・異常や性能問題の発生を即座に知らせます。
- ・アプリケーションの構造を自動的にクラス図化します。
- ・アプリケーション中のメモリリークを発見できます。

ENdoSnipeがソフトウェアの解析／診断／監視に「革命的効率化」をもたらします

6. 問い合わせ先



変革と研鑽
Human Crest

株式会社ヒューマンクレスト

TEL: 045-226-0714

URL: <http://www.humancrest.co.jp>

MAIL: endosnipe@humancrest.co.jp



エスエムジー株式会社

TEL: 045-476-3171

URL: <http://www.smg.co.jp>

MAIL: endosnipe@smg.co.jp

別紙. 検証内容(1/3)

◆ BottleneckEye: システムの構造をクラス図で「見える化」

アプリケーションの実行状況を、 クラス図表示、グラフ表示できること	CPU使用率
	物理メモリ使用量
	仮想マシンメモリ使用量
	ヒープメモリ
	非ヒープメモリ
	GC停止時間
	スレッド数
	スワップ領域使用量
	ネットワーク経由でのデータ受信量
	ネットワーク経由でのデータ送信量
	ファイル出力量
メソッドの呼び出し回数、平均処理時間などの統計情報を参照できること	ファイル入力量

別紙. 検証内容(2/3)

◆ PerformanceDoctor: 性能問題の原因を「見える化」

メモリリークの発生を検知できること	
DatabaseのSQLを解析できること	SQL実行回数と時間の閾値チェック
	同一SQL実行回数の閾値チェック
	SQL中のUNION句登場回数の閾値チェック
	SQL中のOR句登場回数の閾値チェック
Databaseの実行計画を解析できること	SQL実行計画中のテーブル・フルスキャン
	SQL実行計画中のコスト閾値チェック
	SQL中の結合利用回数の閾値チェック
スレッドのCPU占有、スリープを検知できること	メソッド処理時間の閾値チェック
	同一メソッド呼び出し回数の閾値チェック
	Threadの平均CPU使用率の閾値チェック
	Threadスリープ回数と時間の閾値チェック
	Thread同期待ち回数と時間の閾値チェック
	GC発生回数と時間の閾値チェック
	通信量(バイトサイズ)の閾値チェック
	ファイルアクセス量(バイトサイズ)の閾値チェック
	ハンドリングされない例外の発生のチェック

別紙. 検証内容(3/3)

◆ ArrowVision: システムの動作をシーケンス図で「見える化」

アプリケーションの処理の流れをシーケンスダイアグラムとして表示できること
Javaなどの複数ログの情報が1つのシーケンス図で表示できること
ボトルネック解析として、時間が掛かった処理部分を表示できること

その他の観点として、以下のものを実施。

- ・マルチプロセスで動作しているアプリの、プロセス毎の情報を取得できること
- ・Serverに負荷が掛かった状態でも正常に動作ログが出力されること
- ・ENdoSnipeの動作有無によるリソース差異を確認する