

Seasar Conference 2007 Autumn



S2Javelinによるトラブルシューティング -問題発見から、パフォーマンス監視まで-

2007/11/11

S2Javelinコミッタ
エスエムジー株式会社
山崎 政憲



トラブル募集中!

名前: 山崎 政憲 (yamasaki@smg.co.jp)

所属: エスエムジー株式会社 (<http://www.smg.co.jp/>)

職業: Javaシステムトラブルシューター

- 期限内に解決できない場合は半額をお返しします。

<http://itpro.nikkeibp.co.jp/article/NEWS/20061214/256956/>

<http://www.smg.co.jp/service/consulting/jats/index.html>

ニュース **SYSTEMS**
Inpro B はてな A/B お気に入りへ 印刷 ヘルプ

「トラブルを解決できない場合は、半額返します」、SMGがJavaの障害解決サービスを提供開始

記事一覧へ▶

システム・インテグレータのエスエムジーは12月15日、Javaシステムの障害解決サービス「JaTS」を開始すると発表した。Javaで開発したシステムに障害が発生した場合、原因の特定と解決を代行する。サービスの特徴は、料金体系に返金制度があること。あらかじめ定めた期間内に障害が解決できなかった場合、料金の半額を返金する。同社の山崎政憲氏(システムズコンサルティングディヴィジョンテクニカルコンサルタント)は、「これまでの開発経験で培ったJavaの知識や技術を生かして、是が非でも障害を解決するという決意表明である」と返金制度導入の理由を語る。

サービス料金は、期間が1週間の場合は155万円、2週間の場合は300万円、1ヶ月の場合は580万円。これらは、障害に再現性があるかどうかや、障害の数、緊急度に応じて設定する。サービス提供開始は12月18日から。

記事一覧へ▶

SMG
JaTS システム障害時に、ご連絡ください。
045-349-3322
Javaシステム障害解決サービス

サービス内容・料金 | 信頼の証し | 解決事例 | お問い合わせ

Javaシステムの トラブルを 即、解決!

★スタンダードコース
3万円/1時間~

★即行期間内解決保障コース
1週間148万円~

解決できない場合、半額返金! ※即行期間内解決保障コースの場合のみ



本セッションのゴール

1. S2Javelin(とその関連ツール)の効果を知る
システムの中身が、見えるようになります。
2. S2Javelinを使ったトラブルシューティングの流れを知る。
トラブル解決！ヒーローになれます。
3. 開発の中で、S2Javelinを効果的に使う方法を知る。
プロセスに組み込んで、生産性を上げましょう。



1. なぜS2Javelinを作ったか
2. 本日の目玉！
 1. JDBC Javelin (Seasar Conference Edition)
 2. ArrowVision (InetSpyコンバータ、ソースコードジャンプ)
3. **トラブルシューティング！ (デモ)**
4. 効果的な使い方
5. セットアップ (とっても簡単になりました。)
6. よくある質問
7. まとめ

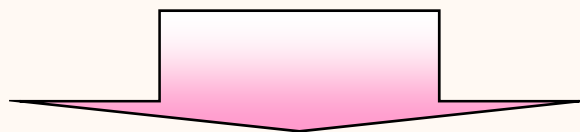


1-1. なぜS2Javelinを作ったか

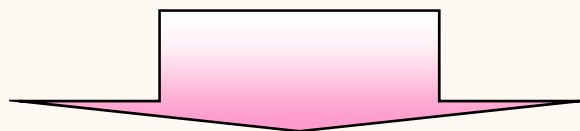
トラブルシュータとしての「困った」がスタート

大量のソースコードの中から問題点を探し出す

思いもよらないSQL実装・・・インデクス漏れ、膨大な結合



実装の生産性 ≠ 開発の生産性
「作りやすさ」「柔軟性」だけ追求しても、
システム開発は成功しない。



1. 障害解析／性能解析の効率を上げたい。
2. システムの動作を「見える化」したい。



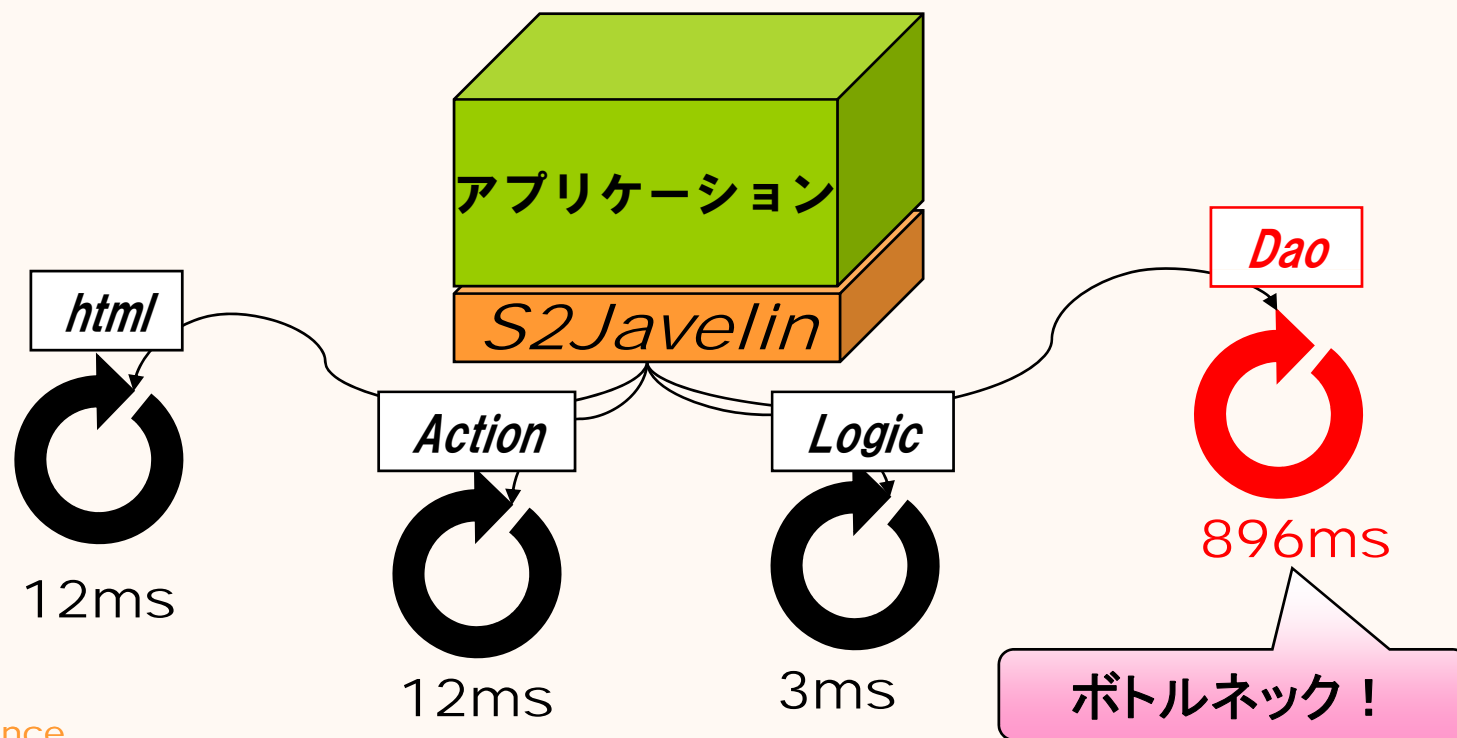
1-2. こうすれば「見える化」できる

1. Seasar 2のDIxAOPを使おう。
 1. Action、Logic、Dao・・・粒度の揃ったコンポーネントは切り分けの単位にピッタリ！
 2. (デバッガ等と比較して)低オーバーヘッドでメソッドをフックできる。
2. Servlet Filterでサーブレットをフックしよう。
3. 集めた情報はクラス図、シーケンス図にしよう。



1-3. やって見た(S2Javelin)

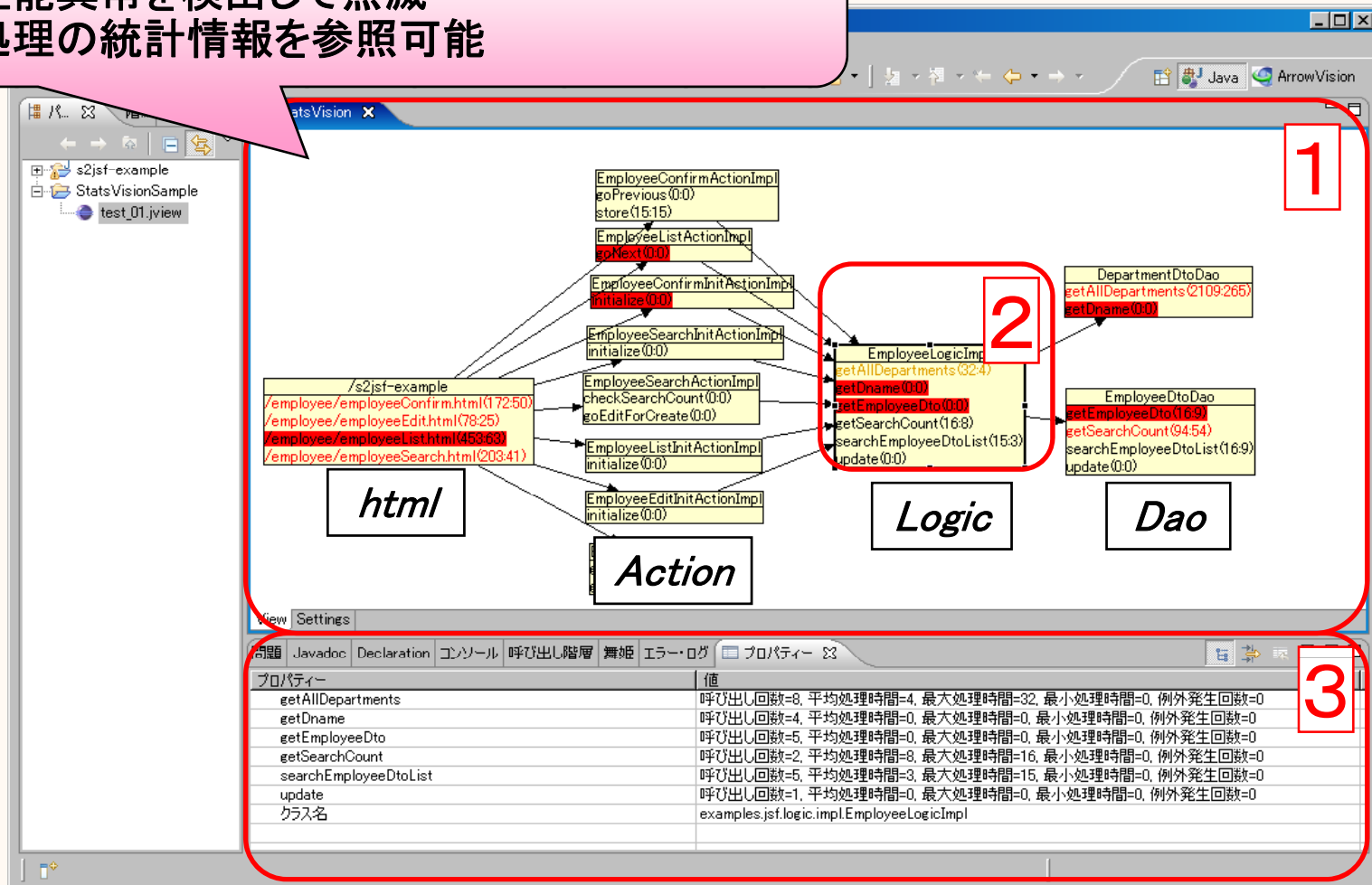
1. メソッドの開始／終了で「呼び出し回数」「処理時間」を記録する。
2. 性能異常や例外を見つけた場合、「アラームを送信」し、解析用のログを出力する。





1-3. やって見た (StatsVision)

1. クラス図形式でシステムの構造を図示
2. 性能異常を検出して点滅
3. 処理の統計情報を参照可能





1-3. やって見た (ArrowVision)

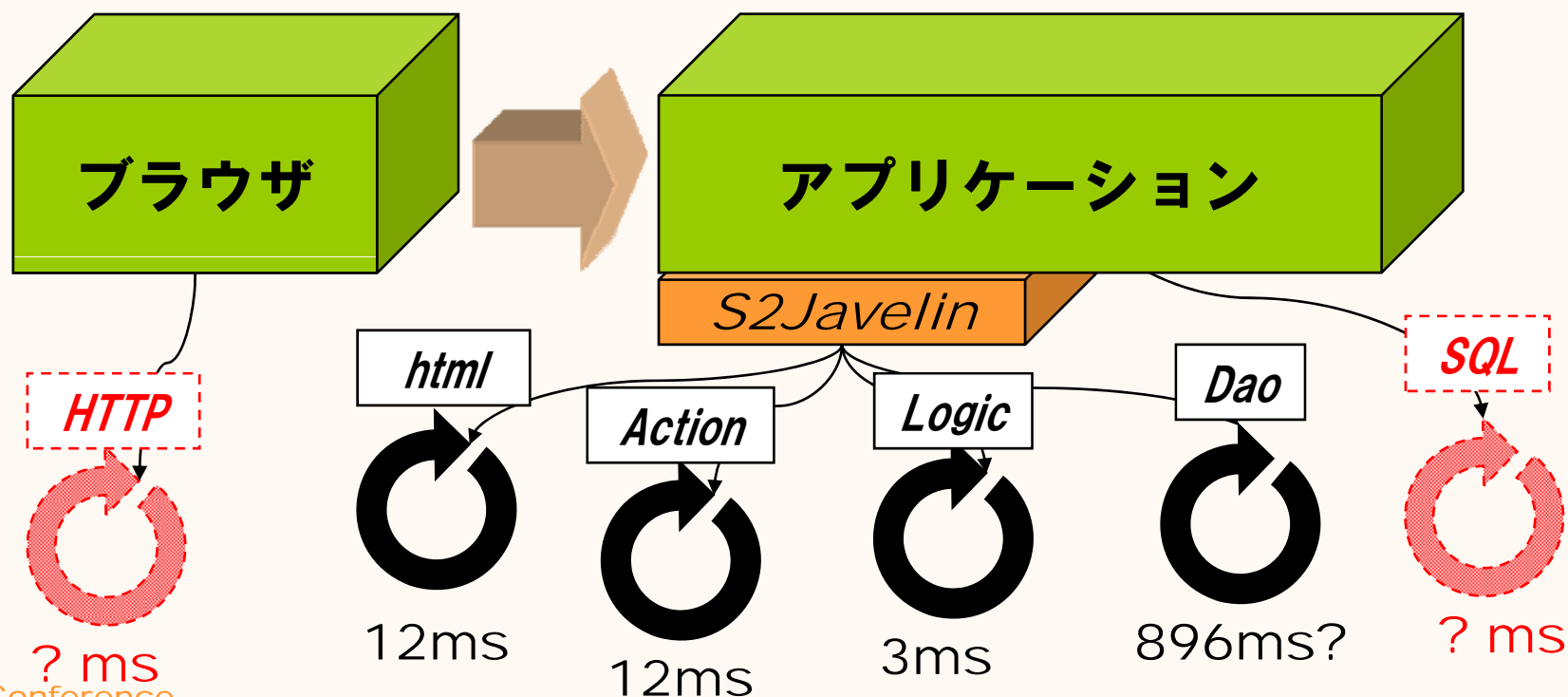
The screenshot shows the ArrowVision interface with three main components:

- 1:** A sequence diagram showing interactions between several objects: `examples.jsf.action.impl.EmployeeSearchInitActionImpl`, `examples.jsf.dao.EmployeeDtoDao`, `examples.jsf.dao.DepartmentDtoDao`, and `examples.jsf.action.impl.EmployeeEditInitActionImpl`. The diagram includes messages like `call getAllDepartments`, `return getAllDepartments`, `call getEmployeeDto`, and `return getEmployeeDto`.
- 2:** A message log table on the right side of the window, listing messages with their names, start times, and end times. The messages include `initialize`, `searchEmployee...`, `return searchEmployee...`, `return initialize`, `return /employee/emp...`, `call /employee/emp...`, `goNext`, `return goNext`, `call initialize`, `call getAllDepartme...`, `return getAllDepartme...`, `return getAllDepartme...`, `call getEmployeeDto`, and `return getEmployeeDto`.
- 3:** A call detail view at the bottom showing the parameters and return values for a specific call. The call is `getEmployeeDto` on `2007/11/09 22:13:42.578`. The arguments are `examples.jsf.action.impl.EmployeeEditInitActionImpl`, `unknown`, and `examples.jsf.action.impl.EmployeeEditInitActionImpl`. The return value is `examples.jsf.action.impl.EmployeeEditInitActionImpl`.

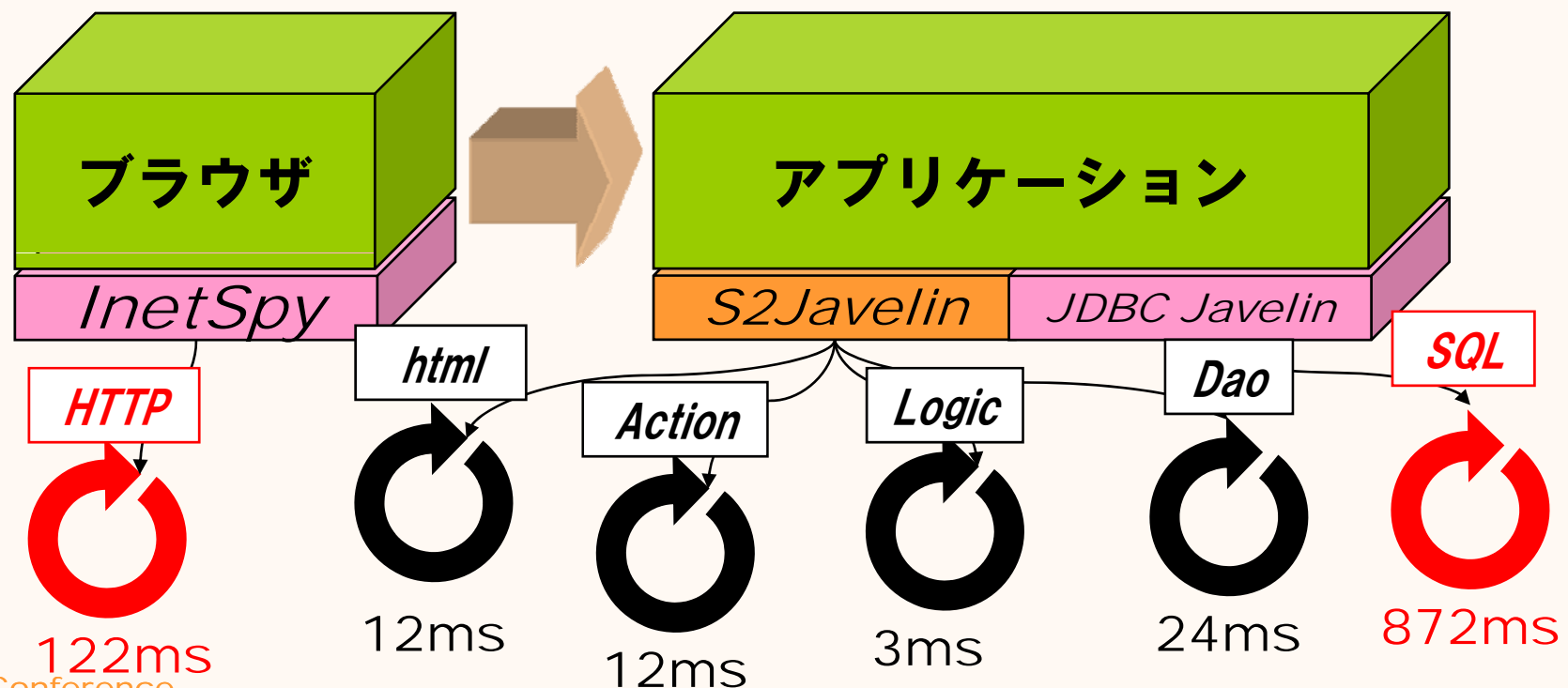
1. シーケンス図形式でシステムの動作を図示
2. 等間隔モードで流れをつかみ、実時間モードで処理の内訳をつかむ
3. 引数や戻り値などの詳細情報を参照可能。

1-4. ネットワークとDBも見たい！

1. ネットワークにも問題があるんじゃないか？
2. Daoが遅いのは、本当はSQLが遅いのではないか？



1. ブラウザからのネットワークアクセスは、InetSpyを使う事で、性能測定を可能に。
2. DBアクセスはBCIを用いたエージェント、JDBC Javelinを開発。





2-1. 本日の目玉！（JDBC Javelin）

1. JDBC Javelin特別版 （Seasar Conference 2007 Autumn Edition）

- 機能
 1. クエリをログに出力します。
 2. クエリの実行時間がわかります。
 3. プリペアドステートメントのパラメータもわかります。
 4. 実行計画も自動的にとっちやいます（**!**）。



2-1. 本日の目玉！（JDBC Javelin）

- JDBC Javelinは当社の障害解決スイートである、「**ENdoSnipe**」の1機能です。
- 有償で提供しているものですが、Seasar 2を使用した案件のみ、無償にてご利用頂けるようにしました。
- ご利用を希望される方は、本セッション後、私、山崎まで、ご連絡ください。

ENdoSnipe

システムの問題解決を効率化

システムの
問題解決を効率化する



2-2. 本日の目玉！（ArrowVision）

1. ArrowVision Ver. 3.2 (New!)

1. InetSpyコンバータ

クライアントとの通信も見えちゃいます。

2. ソースコードジャンプ

「→」をクリックすると、ソースコードの該当箇所を開きます。

デモでお見せします！



3-1. トラブルシュート！（デモ）

1. 環境

1. Seasar 2.4.17
2. S2JSF Example 1.1.2 (Employee Management)
3. Oracle 10g R2

2. ケース

1. SQLのチェック
2. 遅い！
3. 止まる！

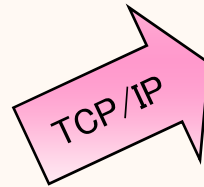


3-1. トラブルシューティング！ (デモ)

概観

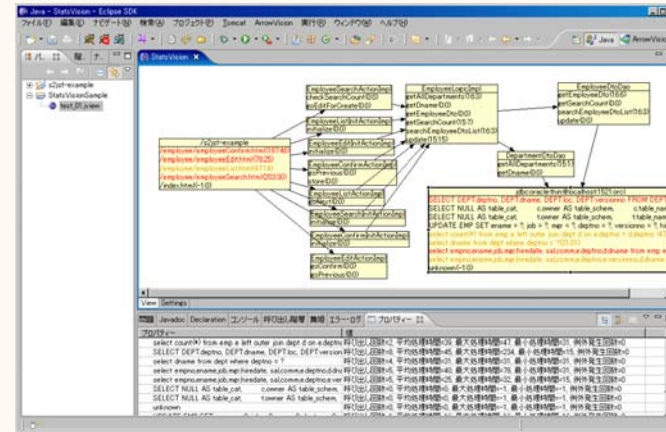
EmployeeNo	EmployeeName	Job	Manager	HireDate	Salary	Commission	DepartmentName	Detail
7904	MILLER	CLERK	7782	1982/01/23	1300		ACCOUNTING	Emp
7839	KING	PRESIDENT		1981/11/17	5000		ACCOUNTING	Emp
7782	CLARK	MANAGER	7839	1981/06/09	2450		ACCOUNTING	Emp
7902	FORD	ANALYST	7566	1981/12/03	3000		RESEARCH	Emp
7676	ADAMS	CLERK	7788	1983/01/12	1100		RESEARCH	Emp
7789	SCOTT	ANALYST	7566	1982/07/09	3000		RESEARCH	Emp
7566	JONES	MANAGER	7839	1981/04/02	2975		RESEARCH	Emp
7869	SMITH	CLERK	7902	1980/12/17	850		RESEARCH	Emp
7900	JAMES	CLERK	7698	1981/12/03	950		SALES	Emp
7844	TURNER	SALESMAN	7698	1981/09/08	1500	0	SALES	Emp

Employee Management

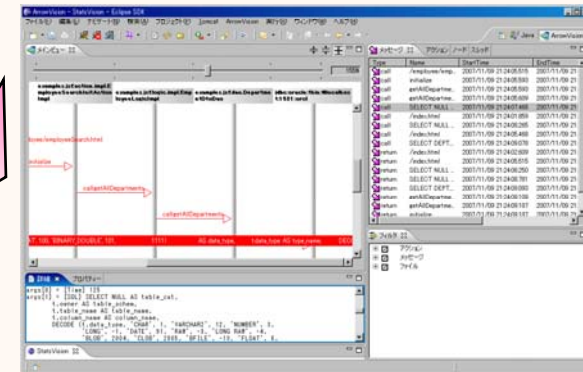


Javelin
ファイル
(jvn)

InetSpy
ログ



StatsVision



ArrowVision



1. SQLをチェック！

②問題のありそうなSQLは、実行計画を確認する。

```

DELETE FROM EMP WHERE empno = ? AND versio = ?(1515)
INSERT INTO EMP (empno, ename, versio) VALUES (?, ?, ?)(1515)
SELECT DEPT.deptno, DEPT.dname, DEPT.loc, DEPT.versio FROM DEPT(4727)
SELECT NULL AS table_cat, c.owner AS table_schem, c.table_name, c.(765648)
SELECT NULL AS table_cat, t.owner AS table_schem, t.table_name AS ...(422265)
UPDATE EMP SET ename = ?, job = ?, mgr = ?, deptno = ?, versio = ?, hiredate ... (1515)
select count(*) from emp e left outer join dept d on e.deptno = d.deptno (32:31)
select dname from dept where deptno = ?
select empno,ename,job,mgr,hiredate, sal,comm,e.deptno,d.dname from emp e left o...(6340)
select empno,ename,job,mgr,hiredate, sal,comm,e.deptno,e.versio,d.dname from ...(47:34)
unknown(-1:0)
  
```

```

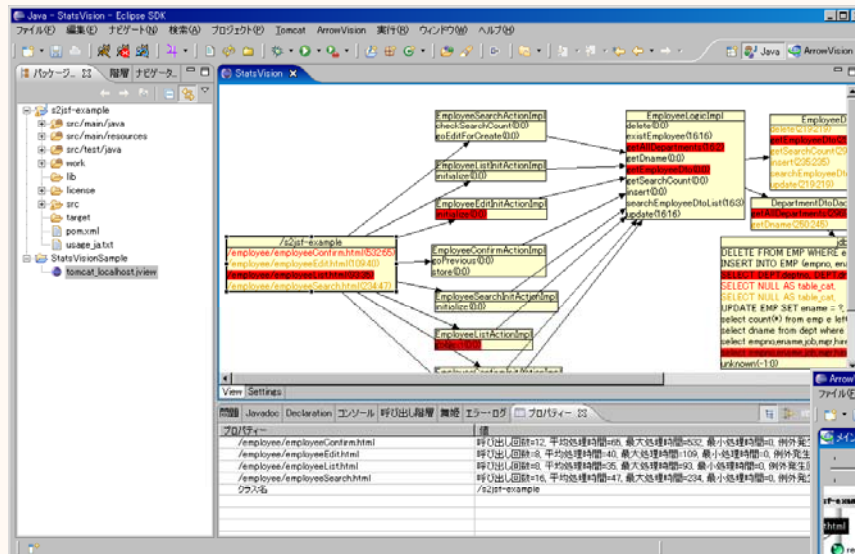
args[0] = [Time] 94
args[1] = [SQL] SELECT NULL AS table_cat,
t.owner AS table_schem,
t.table_name AS table_name,
t.column_name AS column_name,
DECODE (t.data_type, 'CHAR', 1, 'VARCHAR2', 12, 'NUMBER', 3,
'LONG', -1, 'DATE', 31, 'RAW', -3, 'LONG RAW', -4,
'BLOB', 2004, 'CLOB', 2005, 'BFILE', -13, 'FLOAT', 8,
'TIMESTAMP(6)', 38, 'TIMESTAMP(6) WITH TIME ZONE', -101,
'TIMESTAMP(6) WITH LOCAL TIME ZONE', -102,
'INTERVAL YEAR(2) TO MONTH', -103,
'INTERVAL DAY(2) TO SECOND(6)', -104,
'BINARY_FLOAT', 100, 'BINARY_DOUBLE', 101,
1111)
AS data_type,
t.data_type AS type_name,
0 SELECT STATEMENT
1 TABLE ACCESS BY INDEX ROWID OBJ$
2 INDEX UNIQUE SCAN I_OBJ1
3 TABLE ACCESS BY INDEX ROWID OBJ$
4 INDEX UNIQUE SCAN I_OBJ1
5 TABLE ACCESS BY INDEX ROWID OBJ$
6 INDEX UNIQUE SCAN I_OBJ1
7 TABLE ACCESS BY INDEX ROWID OBJ$
8 INDEX UNIQUE SCAN I_OBJ1
9 TABLE ACCESS BY INDEX ROWID OBJ$
10 INDEX UNIQUE SCAN I_OBJ1
11 TABLE ACCESS BY INDEX ROWID OBJ$
12 INDEX UNIQUE SCAN I_OBJ1
13 TABLE ACCESS BY INDEX ROWID OBJ$
14 INDEX UNIQUE SCAN I_OBJ1
15 TABLE ACCESS BY INDEX ROWID OBJ$
16 INDEX UNIQUE SCAN I_OBJ1
17 TABLE ACCESS BY INDEX ROWID OBJ$
18 INDEX UNIQUE SCAN I_OBJ1
19 TABLE ACCESS BY INDEX ROWID OBJ$
20 INDEX UNIQUE SCAN I_OBJ1
21 SORT ORDER BY
22 FILTER
23 NESTED LOOPS OUTER
24 NESTED LOOPS OUTER
25 NESTED LOOPS OUTER
26 NESTED LOOPS OUTER
27 NESTED LOOPS
28 NESTED LOOPS
29 TABLE ACCESS BY INDEX ROWID USER$
30 INDEX RANGE SCAN I_USER1
31 TABLE ACCESS BY INDEX ROWID OBJ$
32 INDEX RANGE SCAN I_OBJ2
33 TABLE ACCESS CLUSTER COL$
34 INDEX UNIQUE SCAN I_OBJ#
35 TABLE ACCESS CLUSTER COLTYPE$
36 INDEX RANGE SCAN I_HH_OBJ#_INTCOL#
37 TABLE ACCESS BY INDEX ROWID OBJ$
38 INDEX RANGE SCAN I_OBJ3
39 TABLE ACCESS CLUSTER USER$
40 INDEX UNIQUE SCAN I_USER#
41 TABLE ACCESS CLUSTER TAB$
42 INDEX UNIQUE SCAN I_OBJ#
43 NESTED LOOPS
44 FIXED TABLE FULL X$KZSR0
45 INDEX RANGE SCAN I_OBJAUTH2
  
```

①SQLの実行回数や時間を確認する。
(S2Daoがメタデータ取得用に発行したSQLまで見えます)



3-3. トラブルシューティング！（ケース2）

1. 「遅い処理」を検出！

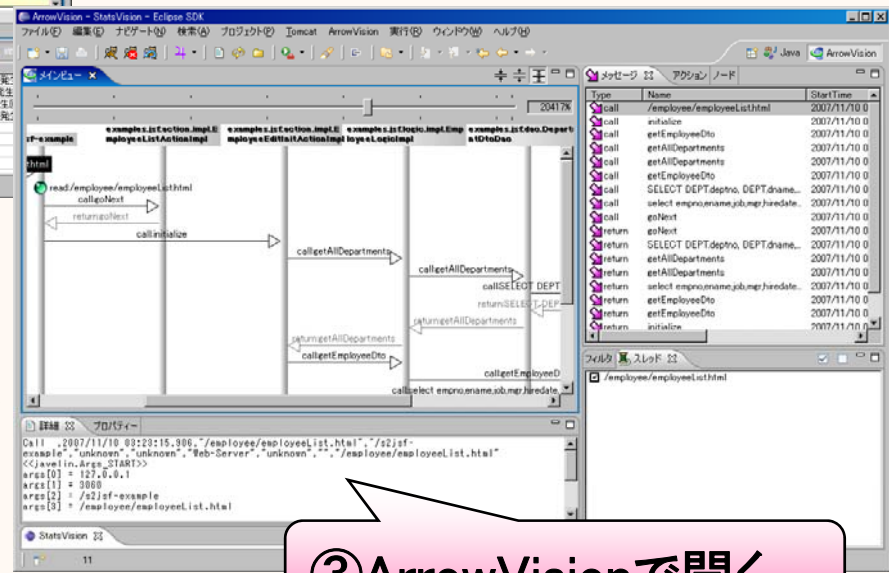


① ブリンクで異常を検出

Javelin
ファイル
(jvn)

InetSpy
ログ

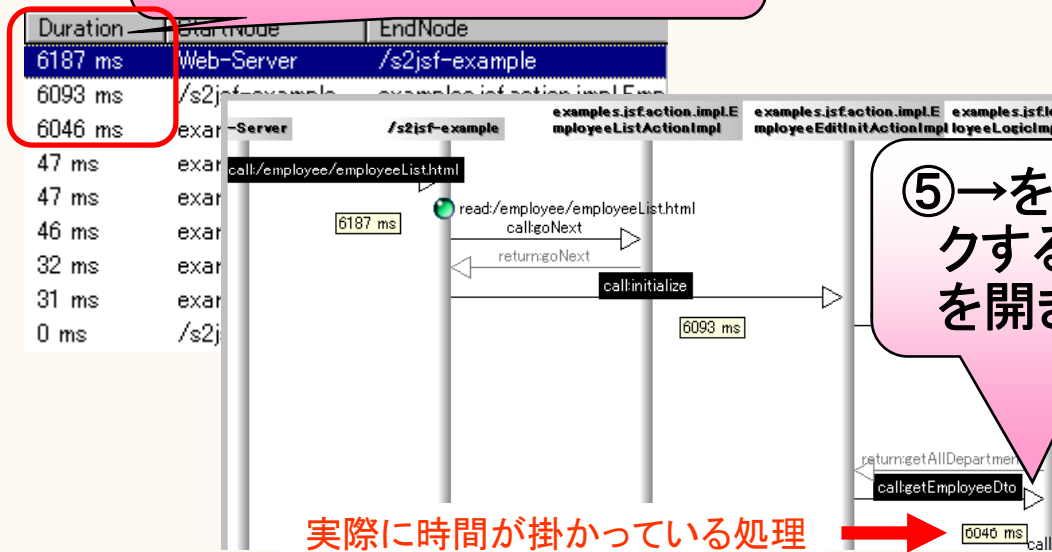
② ファイルを回収



③ ArrowVisionで開く

2. ソースコードジャンプでおかしな場所を発見！

④ Durationでソートして、遅い処理を特定する。



⑤ →をダブルクリックすると、ソースを開きます。

```

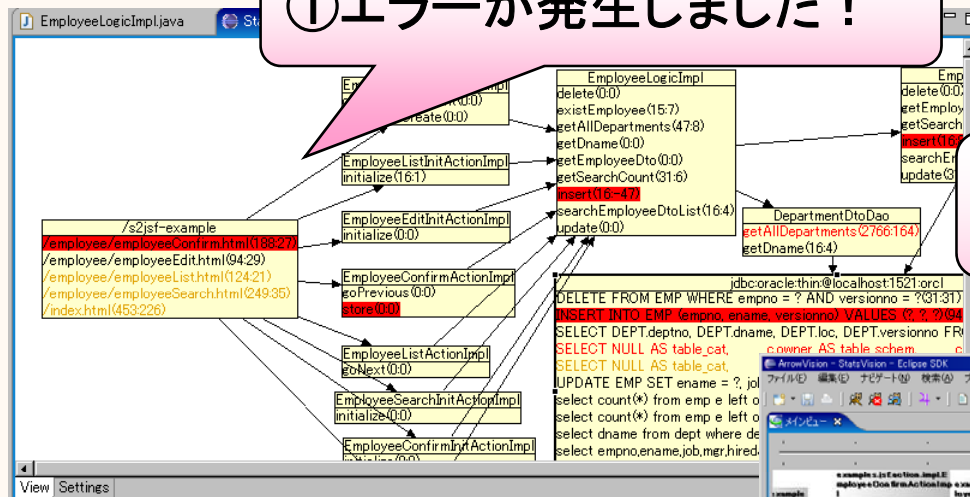
48 public EmployeeDto getEmployeeDto(Integer empno) {
49     EmployeeDto dto = employeeDtoDao.getEmployeeDto(empno);
50     if (dto != null && "ALLEN".equals(dto.getEname()))
51     {
52         try
53         {
54             Thread.sleep(8000);
55         }
56         catch (InterruptedException e)
57         {
58             e.printStackTrace();
59         }
60     }
61     return dto;
62 }
63

```

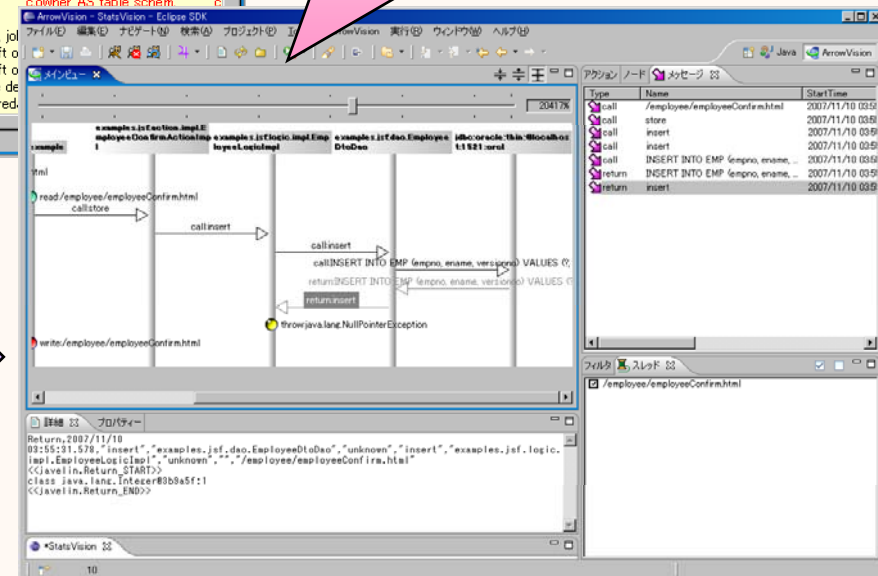
⑥ 問題箇所を発見！

1. 例外発生！

①エラーが発生しました！



③ArrowVisionで開きます。



Javelin
ファイル
(jvn)

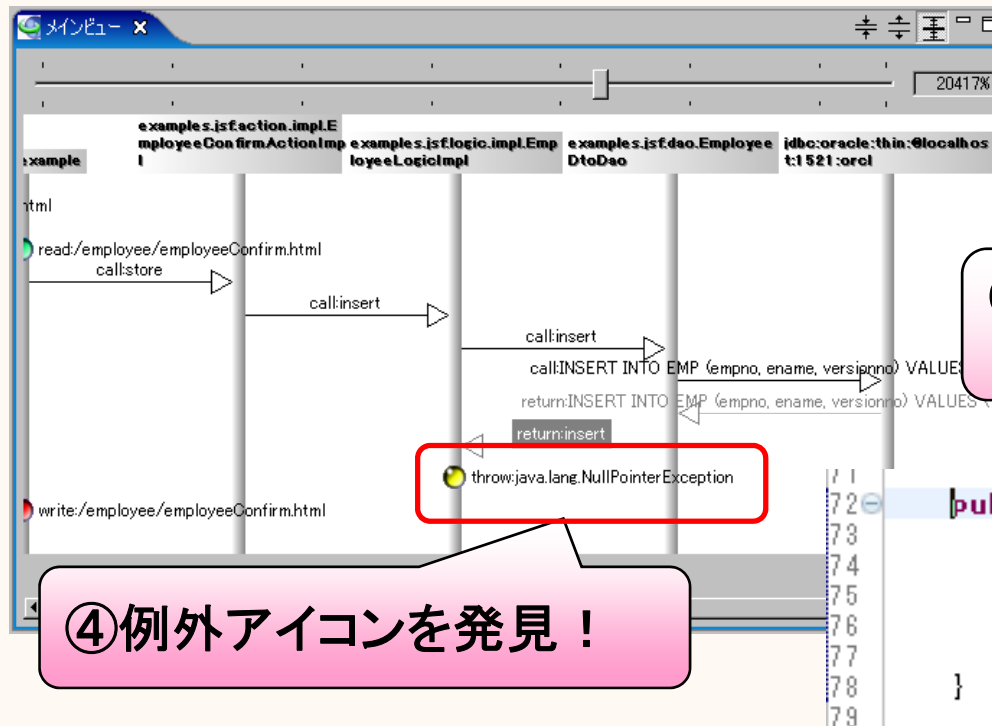
InetSpy
ログ

②ファイルを回収



3-4. トラブルシュート！（ケース3）

2. 原因がわかる！



⑤→をダブルクリックすると、ソースを開きます。

④例外アイコンを発見！

```
71  
72 public void insert(EmployeeDto dto) {  
73     employeeDtoDao.insert(dto);  
74     if ("DICK".equals(dto.getEname()))  
75     {  
76         throw new NullPointerException();  
77     }  
78 }  
79
```

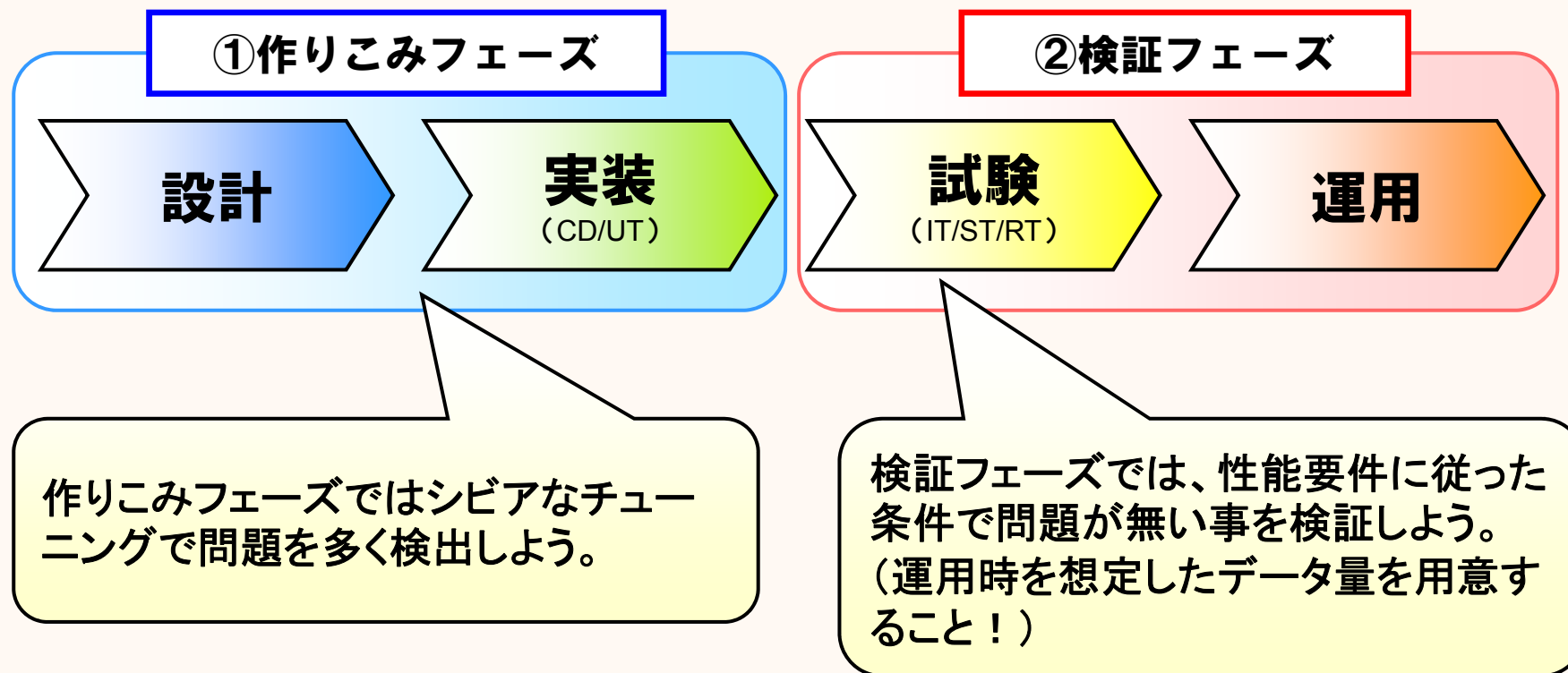


4. テクニック

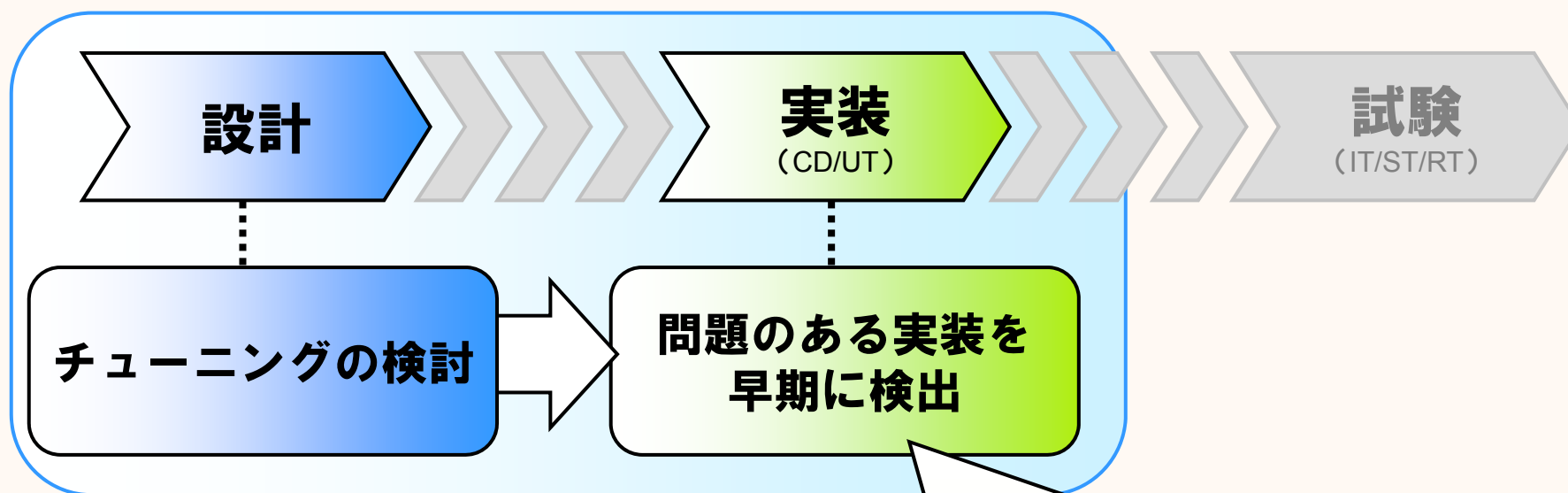
javelin.propertiesのチューニングポイント

1. 性能異常として検出する時間を変えるには？
 - javelin.alarmThreshold
処理時間がこの値を超えれば、StatsVisionに性能異常を通知する。
2. 時間の短い処理のログは不要。
 - javelin.recordThreshold
処理時間がこの値を超えれば、javelinファイルを出力する。
3. 時間のかかるSQLだけ、実行計画を取得したい。
 - javelin.jdbc.execPlanThreshold
SQLの処理時間がこの値を超えれば、実行計画を取得する。
4. 問題無い処理は無視したい。
 - javelin.statisticsThreshold
処理時間がこの値を超えれば、
統計処理(平均、最小、最大の算出)を行う。

フェーズごとの使い分けが効果的！

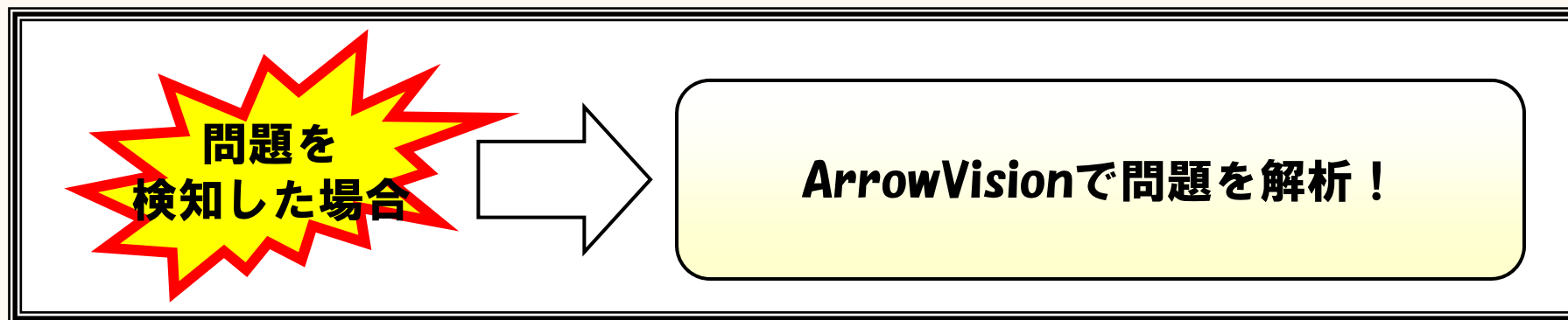
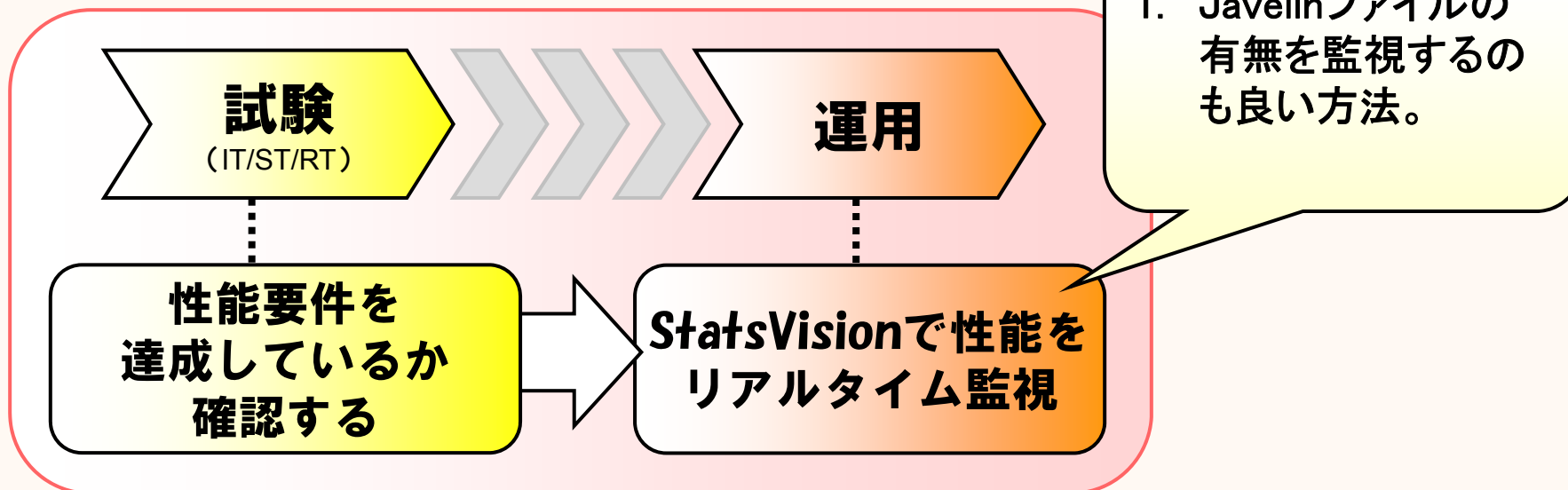


1. 作りこみフェーズ



1. 処理に時間がかかり過ぎている。
2. メソッドの呼び出し回数が多い。
3. SQLの実行回数が多い。
4. 実行計画中に、フルスキャンやソートが頻繁に登場する。

1. 検証フェーズ





5-1. S2Javelinのセットアップ

1. S2Javelin

1. WEB-INF/libにs2javelin.jarを置く。
2. allaop.diconでインタセプターを追加

```
<components>  
  <include path="aop.dicon" />  
  <include path="dao.dicon" />  
  <include path="j2ee.dicon" />  
  <include path="javelin.dicon" />
```

(中略)

```
<component name="actionInterceptorChain"  
  class="org.seasar.framework.aop.interceptors.InterceptorChain">  
  <initMethod name="add"><arg>s2JavelinInterceptor</arg></initMethod>  
  <initMethod name="add"><arg>aop.traceInterceptor</arg></initMethod>  
  <initMethod name="add"><arg>actionThrowsInterceptor</arg></initMethod>  
</component>
```

この要領でLogic、Daoにもインターセプターを追加すればOK！



5-2. Servlet Filterのセットアップ

2. Servlet Filterを追加する。

```
<filter>
  <filter-name>s2javelinfilter</filter-name>
  <filter-class>org.seasar.javelin.S2JavelinFilter</filter-class>
</filter>
(中略)
<filter-mapping>
  <filter-name>s2javelinfilter</filter-name>
  <url-pattern>*.html</url-pattern>
</filter-mapping>
```

3. VMの起動引数で、システム・プロパティを指定する。(Tomcatならstartupスクリプトを編集する。)

```
-Djavelin.property=C:/Javelin/conf/javelin.properties
```



5-3. JDBC Javelinのセットアップ

2. JDBC Javelin

1. VMの起動引数に、Java Agentを指定するだけ。
(Tomcatならstartupスクリプトを編集する。)

```
-javaagent:C:/Javelin/lib/jdbcjavelin.jar
```



5-4. StatsVisionセットアップ

3. StatsVision

1. ダウンロードしてEclipse 3.2にインストール。
 - <http://s2javelin.sandbox.seasar.org/s2javelin.html>
2. 設定ファイルをテキストエディタで作成する。
(拡張子は.jview)

```
127.0.0.1  
18000  
default  
100  
500
```

1. IPアドレス
2. ポート番号
3. ドメイン(任意の文字列)
4. 警告用閾値(黄文字で表示)
5. エラー用閾値(赤文字で表示)

3. Eclipse上から開ける事を確認する。



5-5. ArrowVisionセットアップ

4. ArrowVision

1. ダウンロードしてEclipse 3.2にインストール。

- http://www.smg.co.jp/service/products/arrow_vision/index.html

ソースコードジャンプが使えるEclipseプラグイン版をダウンロード！

2. ArrowVisionパースペクティブを開く。



5-6. InetSpyセットアップ

5. 横取り丸 & InetSpy

1. ダウンロード & インストール

“秀まるお”さんのホームページからダウンロード

<http://hide.maruo.co.jp/software/ydm.html>

<http://hide.maruo.co.jp/software/inetspy.html>

2. ブラウザのプロキシ設定を変更する

3. ログ取り開始

4. ブラウズ

5. ログ保存

URL	リクエスト時刻	転送時刻	応答時刻	終了時刻	メソッド	リクエストサイズ	応答サイズ
• http://localhost:8080/s2jsf-example/employee/employeeSearch.html	23:38:40.51	+0.03	+0.04	+0.04	POST	742	247
• http://localhost:8080/s2jsf-example/employee/employeeSearch.html	23:38:40.57	+0.00	+0.09	+0.09	GET	574	8148
• http://localhost:8080/s2jsf-example/employee/employeeSearch.html	23:38:41.89	+0.00	+1.11	+1.12	POST	861	23483
• http://localhost:8080/s2jsf-example/employee/employeeList.html	23:38:45.20	+0.06	+0.31	+0.31	POST	762	8399
• http://localhost:8080/s2jsf-example/employee/employeeEdit.html	23:38:46.60	+0.00	+0.14	+0.14	POST	1001	7805
• http://localhost:8080/s2jsf-example/employee/employeeConfirm.html	23:38:47.90	+0.00	+0.10	+0.10	POST	921	247
• http://localhost:8080/s2jsf-example/employee/employeeSearch.html	23:38:48.01	+0.00	+0.12	+0.12	GET	572	8148



6. よくある質問

1. 性能への影響はどれくらい？

1. CPU使用率は数パーセント
2. TATへの影響はチューニング次第。
 1. ログ出力が多ければ、劣化。
 2. 実行計画を頻繁に取得すれば、劣化。
3. メモリ使用量は10～20%は増加する。
ヒープエリアだけでなく、パーマネントエリアの増加にも注意が必要。

2. 対応しているDBMSは？

1. JDBCに準拠していれば動きます。
 - 動作確認は、HSQLDB、Oracle 10g、PostgreSQL 8.2、SQL Server 2005、MySQLで実施。
2. 実行計画の取得はOracle 10g、PostgreSQL 8.2、SQL Server 2005に対応。

3. 対応しているJavaのバージョンは？

Java 5.0以降。



7. まとめ

1. S2Javelinは「見える化」ツールです。
 1. SeasarアプリケーションはS2Javelin、JDBC Javelinで「見える化」しよう。
 2. JDBC Javelinが(無料で)使えるのはSeasarだけ。
2. トラブルシュート／保守作業の効率が上がります。
3. フェーズごとに使い分けることで効果アップ。



8. お願い

できるだけ多くのフィードバックをお願いします。

1. StatsVision / S2Javelin

seaser-user MLへ。

2. ArrowVision / JDBC Javelin

endosnipe@smg.co.jpへ。

(JDBC Javelin特別版を使ってみたい方は、
こちらへリクエスト願います。)



ご清聴、
ありがとうございました。