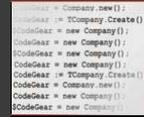


JBuilder 2007によるJEEアプリケーションパフォーマンスチューニング



ボーランド株式会社 Developer Tools Group 高橋智宏

Copyright (C) 2007, CodeGear. 本文書の一部または全部の転載を禁止します。

講師紹介

- 学生の時購入したTurboC++2ndからの熱狂的なボーランドファン
- JDK1.0 の頃からずっと、Javaやっています
- 長年にわたり「Java読書会」を運営
 - <http://www.javareading.com/bof/>

アジェンダ

- 問題のある JEE Webアプリケーション
- JVMPI / JVMTI
- 無償のツール Eclipse / TPTP
- Eclipseベースの JBuilder 2007
- Optimizelt 2007 for Eclipse
- Optimizeltを使った分析手順

- EJB2.x,EJB3.0のビジュアル開発機能の紹介

JEEアプリケーションのプロファイリング

問題のある JEE Webアプリケーション

- Servlet, JSPの処理時間が長い
- 例外が発生しているらしい!?
- JDBC等のリソースの解放処理がお粗末
- HttpSessionの間違った使い方
- メモリが延々と増加し続ける

さて、どうやって原因を見つけようか...

JVMPI / JVMTI

- JavaVMの挙動を監視し、情報収集するためのインターフェース
- .dll, .so等のライブラリ(エージェント)が必要



- ~ JDK1.5 まで
 - Java Virtual Machine Profiler Interface (JVMPI)
- JDK1.5, JDK6 ~
 - Java Virtual Machine Tool Interface (JVMTI)

JVMPI / JVMTI(続き)

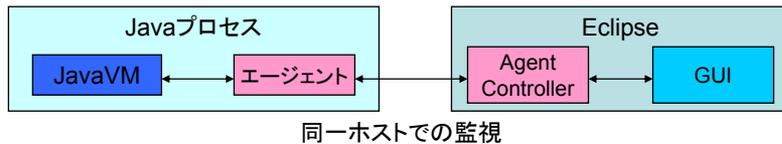
- JVMPIやJVMTIを使うと
 - JavaVM内のメモリの使用状況を把握できる
 - 実行時間の測定が行える
 - 実行されたコードを測定できる
 - スレッドの状態を把握できる

無償のツール Eclipse / TPTP

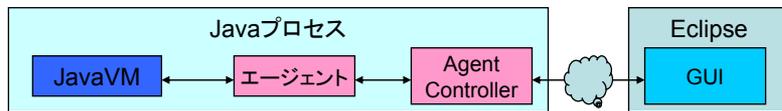
- Eclipseには、JavaVMの解析に役立つオープンソースのプラットフォームや、JVMPI/JVMTI準拠のツールが無償で用意されています
- Test & Performance Tools Platform (TPTP)
 - URL: <http://www.eclipse.org/tptp/>
 - 共通プラットフォーム
 - モニタリングツール
 - テストツール
 - トレースおよびプロファイリングツール

TPTPの仕組み

- JVMPIのエージェントをサーバーに組み込む
- モニタリングツールは、Eclipseに統合されたものを使用
- Agent Controller がエージェントとモニタリングツール (GUI)を仲介



同一ホストでの監視

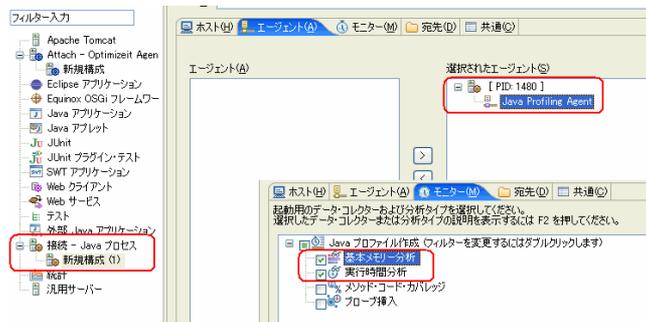


リモートホストから監視

CodeGear
Where developers matter

TPTPを使ってみる

- JBossの起動スクリプトに、piAgent.dllを組み込む
- EclipseからJBossにアタッチ
 - フィルターの追加(「org.jboss.*」を「除外」)
 - 基本メモリー分析
 - 実行時間分析
- メニューから[モニターの開始]



問題のあるJEE Webアプリケーション を動作させてみる

- 実行時間分析
 - メソッド呼び出しの詳細
 - 細かすぎ。フィルターで「org.jboss.*」を「包含」すればかなり見やすくなる。

選択したメソッド

>呼び出し	メソッド	クラス	パッケージ	基本時間 (秒)
4	HotSpotTest0 void	MyServlet	mypack	7.692479

選択したメソッドの呼び出し対象:

>呼び出し	メソッド	クラス	パッケージ	基本時間 (秒)
1	connect(java.lang.String, java.util.Pro...	NonRegisterin...	com.mysql.jdbc	0.102180
2	release0 void	QueuedSemap...	EDUoswego.c...	0.005753
1	containsKey(java.lang.Object) boolean	ConcurrentRe...	EDUoswego.c...	0.000140
2	attempt(long) boolean	QueuedSemap...	EDUoswego.c...	0.049885
2	next0 boolean	Connection	com.mysql.jdbc	0.117895
1	getTransactionIsolation0 int	Connection	com.mysql.jdbc	0.000323
2	get0 boolean	Synchronized...	EDUoswego.c...	0.004378
2	close0 void	ResultSet	com.mysql.jdbc	0.014071
1	acceptsURL(java.lang.String) boolean	NonRegisterin...	com.mysql.jdbc	0.000028
1	-clinit-0	Driver	com.mysql.jdbc	0.000184
1	SynchronizedBoolean(boolean)	Synchronized...	EDUoswego.c...	0.000013
2	executeQuery(java.lang.String) java.s...	Statement	com.mysql.jdbc	0.166393

CodeGear
Where developers matter

問題のあるJEE Webアプリケーション を動作させてみる

- メモリー分析
- オブジェクトの参照 – 遅くて、細か過ぎて、生成場所も特定できない

参照元	パッケージ	サイズ (バイト)	参照の数
MultiMap	org.apache.tomcat.util.coll...	0	0
MultiMap\$Field	org.apache.tomcat.util.coll...	0	0
MyClass	mypack	408	16
Object	mypack	1120	424
CopyOnWriteArrayList	java.lang	23490	408
AbstractKernelDeployment	EDU.oswego.cs.dl.util.concu...	624	0
ConcurrentReaderHashMap\$Ei	org.jboss.kernel.plugins.dep...	40	0
[ConcurrentReaderHashMap	EDU.oswego.cs.dl.util.concu...	24	0
ConcurrentReaderHashMap	EDU.oswego.cs.dl.util.concu...	48848720	277
ConcurrentReaderHashMap	EDU.oswego.cs.dl.util.concu...	6048	103
ConcurrentReaderHashMap	EDU.oswego.cs.dl.util.concu...	37824	7097
CopyOnWriteArraySet	EDU.oswego.cs.dl.util.concu...	576	27
AbstractDependencyInfo	EDU.oswego.cs.dl.util.concu...	576	1
ConcurrentReaderHashMap	org.jboss.dependency.plugi...	24	0
ConcurrentReaderHashMap\$Ei	EDU.oswego.cs.dl.util.concu...	192	7097
[ConcurrentReaderHashM	EDU.oswego.cs.dl.util.concu...	48848720	277
ConcurrentReaderHashM	EDU.oswego.cs.dl.util.concu...	6048	103
ConcurrentReaderHashM	EDU.oswego.cs.dl.util.concu...	37824	7097
UnifiedLoaderRepository3	EDU.oswego.cs.dl.util.concu...	72	0
CopyOnWriteArrayList\$COWrite	java.lang	16	0
Heap	java.lang	96	4
BasicThreadPool	EDU.oswego.cs.dl.util.concu...	224	1
ConcurrentReaderHashM	org.jboss.util.threadpool	24	0
[ConcurrentReaderHashM	EDU.oswego.cs.dl.util.concu...	48848720	277
ConcurrentReaderHashM	EDU.oswego.cs.dl.util.concu...	6048	103
ConcurrentReaderHashM	EDU.oswego.cs.dl.util.concu...	37824	7097

Eclipseベースの JBuilder 2007

- JBuilder 2007 は、Eclipse3.2をベースとした統合開発環境。IDEの最高峰「JBuilder 2006」の後継。
- JEE(Web,EJB2.x,EJB3.0等)のビジュアル開発
 - EJBワークベンチ
 - Webサービスデザイナー
- チーム開発・プロジェクト管理の効率化
 - ProjectAssist
 - TeamInsight
- TogetherによるUMLモデリング機能
- Optimizelt 2007 による分析&チューニング

Optimizelt 2007 for Eclipse

- Optimizeltシリーズは、2002年の「Optimizelt 4」リリース以降、最も優れたJava用プロファイリングツール。
- Eclipse/TPTPに対応
- JVMPI/JVMTIに対応
 - Memory Profiler, CPU Profiler, Quality Analyzer
 - Thread Debugger
 - Code Coverage
 - Request Analyzer(JEE専用)
- 分析中のデータやVMの状態をスナップショットに丸ごと保存できる
 - 後からゆっくりと問題の分析が可能

Memory Profiler

- インスタンス数やオブジェクトの参照関係を把握
 - ある時点(マーク)からの差分だけを追跡可能
 - インスタンスの生成場所を特定可能
 - 解放されない原因となっている参照だけを表示可能
 - メモリリーク自動検出機能

The screenshot displays the Java Memory Profiler interface. On the left, a table lists class names, packages, instance counts, and sizes. The main area shows a tree view of allocation backtraces for a specific memory location. On the right, a 'Reduced reference graph' shows the relationships between objects, including references to arrays, maps, and other objects. The bottom section shows 'Allocation locations' and 'Allocation Backtrace'.

CPU Profiler

- 指定した時間内の実行時間を把握
 - 様々な測定方法
 - スレッドごとに分析可能
 - ホットスポットとソースコードの対応付け

The screenshot displays the Java CPU Profiler interface. The top part shows 'CPU Options' with settings for instrumentation, precision, and sampling. The main area shows a 'Time profiler output' window with a table of threads and their CPU usage. The bottom part shows a detailed view of a hot spot, listing the source code locations and the time spent in each.

Code Coverage

- 実行時に通過したソースコードに関する情報を収集

unknown at JPTOL-TTAKXP [PID: 3156]

Class name	Package	Coverage	Diff.	Total lines	Missed lines
2 Classes/Interfa	1 Packages	17.60%	17.60%	250	206
MyServlet (1)	mypack	35.20%	35.20%	125	81
MyServlet	mypack	0%	None	125	125

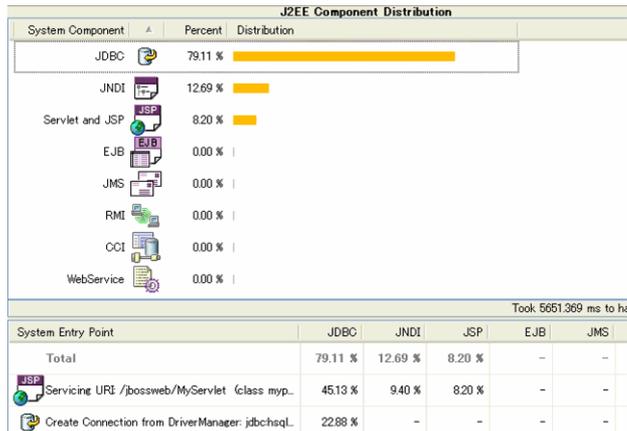
Include/exclude filters: mypack*

Request Analyzer

- JEEアプリケーション特有のパフォーマンス問題の分析や品質問題の検出を支援
 - System Dashboard
 - Component Performance
 - Component Details
 - Quality Analyzer
- 各種アプリケーションサーバーへの対応
 - WebLogic 9.2, 8.1.4
 - JBoss 4.0.5, 3.2.6
 - Tomcat 5.0, 5.5
 - その他の一般的なJ2EEサーバすべて

Request Analyzer System Dashboard

- JEEアプリケーション全体の処理時間を各レイヤごとに分けて表示



CodeGear
Where developers matter

Request Analyzer Component Performance

- アプリケーションのエントリーポイントごとに、各レイヤが消費した時間を分析

Component Performance

Operations	Called	Time %	Time
Latency of external calls to AppServers	140	100.00	5651.369
Servleting URI /bossweb/MyServlet (class mypack.MyServlet)	1	62.74	3545.456
Create Connection from DataSource: java/MySqlDS	500	28.55	1613.274
Execute Statement: SELECT * FROM userinfo;	500	15.35	867.233
JNDI Lookup: java/MySqlDS	500	9.40	531.38
Create Statement for: java/MySqlDS	500	0.67	37.89
Access ResultSet: SELECT * FROM userinfo;	2000	0.57	32.344
Create Connection from DriverManager: jdbc:hsql:C#JBUILDER2007#thirdparty#boss-	1	22.88	1292.757
Create Connection from DataSource: java/DefaultDS	14	4.71	265.937
Execute Statement: SELECT * FROM INFORMATION_SCHEMA.SYSTEM_TABLES WH	1	3.62	204.63
Prepare Statement: CREATE TABLE JMS_USERS (USERID VARCHAR(20) NOT NULL,	1	1.15	65.134
JNDI Lookup: java/jms/jms-console	1	1.04	59.009
JNDI Lookup: java/jms/jms-console	1	0.24	13.401
Execute Statement: SELECT MAX(TXID) TXID FROM (SELECT MAX(TXID) AS TXID I	1	0.83	46.828

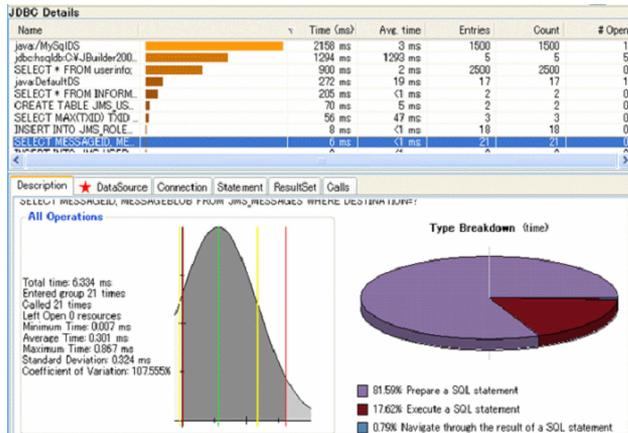
Request Breakdown Hotspots Sub-graph

Hotspots	Type	Time (ms)	Time %	Call Count
Create Connection from DataSource: java/MySqlL...		1613 ms	45.50 %	500
Execute Statement: SELECT * FROM userinfo;		867 ms	24.46 %	500
JNDI Lookup: java/MySqlDS		531 ms	14.99 %	500
Servleting URI /bossweb/MyServlet (class myp...		463 ms	13.07 %	1

CodeGear
Where developers matter

Request Analyzer Component Details

- JEEの各レイヤ(例:JDBC)ごとに、処理時間を詳細に報告



Request Analyzer Quality Analyzer

- JEE特有の品質問題を実行時に検出

Application Quality Analyzer

Error	Count	Description
Not closed	1	A Statement was never closed
An Exception was thrown	1	An exception "javax.naming.NameNotFoundException..."
Session entry failed serialization	2	Session entry data points to a non-serializable object
Had child when closed	1	A Connection left open resources when closed

Description: Connection Closed At ★ Statement Connection DataSource

Forward display

- (left open 1 resource) ApplicationFilterChain.internalDoFilter() (ApplicationFilterChain.java:252)
- (left open 1 resource) Core Java Classes
- (left open 1 resource) Servicing URL: /jbossweb/MyServlet (class mypack.MyServlet)
- (left open 1 resource) MyServlet (2).doGet() (MyServlet.java:156)
- (left open 1 resource) MyServlet (2).ResourceLeakTest1() (MyServlet.java:84)
- ★ (left open 1 resource) Create Statement for java/MySqlDS

MyServlet.java

```
Source Code
private void ResourceLeakTest1() {
    Connection conn = null;
    Statement stat = null;
    ResultSet rs = null;
    try {
        Context ctx = getInitialContext();
        DataSource ds = (DataSource) ctx.lookup("java/MySqlDS");
        conn = ds.getConnection();
        stat = conn.createStatement();
        rs = stat.executeQuery("SELECT * FROM userinfo");
        // rs.close(); 忘れ!!
        // stat.close(); 忘れ!!
        conn.close();
    }
}
```

OptimizeItを使った分析手順

- JBossの起動スクリプトにOptimizeItのエージェント(oii.dll)を組み込む(JVMPI or JVMTI)
 - フィルターの追加で、「org.jboss.*」を「Group」化
- EclipseからJBossにアタッチ

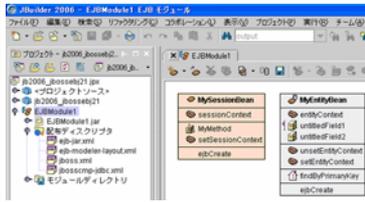


CodeGear
from Borland
Where developers matter

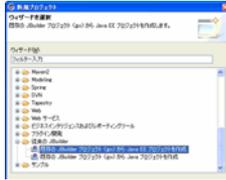
CodeGear
from Borland

EJB2.x, EJB3.0のビジュアル開発機能

従来のJBuilderプロジェクトのインポート



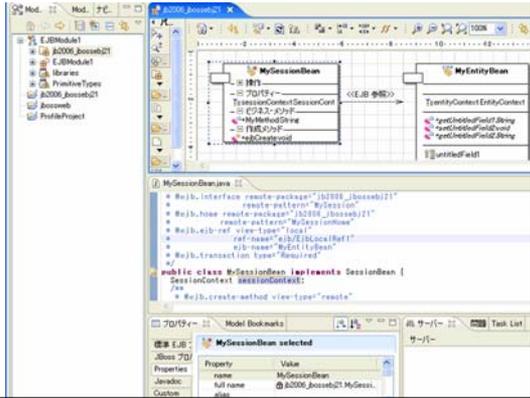
EJB2.xプロジェクト



変換用ウィザード

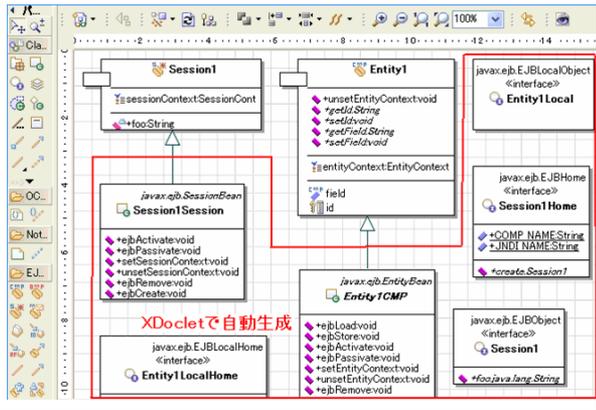
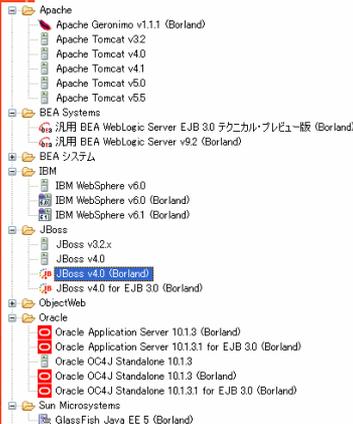


EJB2.x(XDoclet) or EJB3.0(Annotation)



「EJB2.1」ビジュアル開発

- 各種J2EEサーバへの対応
- テーブルからエンティティBeanを自動生成
- EJBの「モデルソースコード/Xdoclet」の完全同期



「EJB3.0」ビジュアル開発

- 各種J2EEサーバへの対応
- テーブルからエンティティBeanを自動生成
- EJBの「モデル/ソースコード/アノテーション」の完全同期

The screenshot displays the Developer Camp IDE interface. On the left, a project browser shows a package structure with 'Session1Remote.java' selected. The central editor shows the source code of 'Session1Remote' implementing 'Session1Remote' interface. The right-hand pane is divided into 'Properties' and 'Persistence' sections. The 'Persistence' section is active, showing settings for 'persistence.xml' and 'JTA' transaction type. The 'JNDI' section is also visible, showing 'java/MySqlDS' as the JNDI name.

CodeGear
from Borland®

Thank you