Profiler, the better Debugger?

Heiko W. Rupp
Senior Developer JBoss ON
Red Hat
Overall Presentation Goal

Provide a different viewpoint on tackling software issues
Heiko W. Rupp is senior developer with JBoss Operations Network, Jopr and RHQ.

He wrote the first German book on JBoss AS and one of the first German books on EJB3.

He is and has been involved in other Open Source and commercial software projects.

Heiko also gave trainings and was consulting at customers.
It is obvious:

If the software does not work we use a debugger.
If it is slow, we use a profiler.
It is obvious:

If the software does not work we use a debugger.
If it is slow, we use a profiler.

But: is this always the right thing to do?
Short outline

- Questions to solve
- Differences Profiler <-> Debugger
- Why shall I use a profiler as a debugger?
- Demo
- Conclusion
Questions to solve

- When working with unknown software
- Where do I start?
- What happens when I click here?
- What is really going on in this piece of code:

```java
Query q = "SELECT p FROM Person AS p WHERE ...
q.getSingleResult();
```
A look at a Debugger

- Tool to
  - Step through code
  - Look at variable content
- Allows to stop the application
  - On Exception
  - On Breakpoint (perhaps with condition)
- But: issues with timeouts in larger applications
A look at a Profiler

- Analyzing of
  - CPU usage
  - Memory Usage
- Application runs while profiling
- No view on content of variables

Products (Free and commercial)

- NetBeans Profiler, Eclipse TPTP, JProfiler, JProbe
Why a profiler for debugging?

- Start of a call-chain is unknown
- New, unknown application
- Call stack uses reflection
- Hard to follow the call tree
- Use of big, complex frameworks
- Hard to step through a deep call stack
- Transaction timeouts render values invalid
Let's see how to find out what happens "when you click here"

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Total</th>
<th>Avail</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>snert</td>
<td>1</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>File System</td>
<td>4</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>2</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>HttpCheck</td>
<td>1</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td><strong>JBossAS Server</strong></td>
<td>2</td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>snert RHQ Agent</td>
<td>1</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Network Adapter</td>
<td>6</td>
<td>!</td>
<td></td>
</tr>
</tbody>
</table>

Example from Jopr/RHQ
Concluding statement

A profiler can help to view a debugging task from a different angle.
A profiler does not replace a debugger, but can add a lot of value.
Thanks for your attention!

http://www.rhq-project.org/
http://www.jboss.org/jopr/

http://pilhuhn.blogspot.com/
http://twitter.com/pilhuhn