Evaluating the Concurrency & Parallelism Mechanisms in Java

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science
Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA
Learning Objectives in this Part of the Lesson

- Know which Java mechanism(s) to understand & apply
Which Java Mechanism(s) to Understand & Apply
Which Java Mechanism(s) to Understand & Apply

- Java’s concurrency & parallelism mechanisms span multiple layers in the software stack
Which Java Mechanism(s) to Understand & Apply

- Java’s concurrency & parallelism mechanisms span multiple layers in the software stack
- Choosing best mechanism(s) depend on various factors
Which Java Mechanism(s) to Understand & Apply

- Developers of low-level classes & performance-sensitive apps may prefer shared object mechanisms.

```java
package java.util.concurrent;

Description

Utility classes commonly useful in concurrent programming. This package includes a few small standardized extensible frameworks, as well as some classes that provide useful functionality and are otherwise tedious or difficult to implement. Here are brief descriptions of the main components. See also the java.util.concurrent.locks and java.util.concurrent.atomic packages.
```

- e.g., java.util.concurrent as per [www.youtube.com/watch?v=sq0MX3fHkro](http://www.youtube.com/watch?v=sq0MX3fHkro)
Which Java Mechanism(s) to Understand & Apply

- Developers of low-level classes & performance-sensitive apps may prefer shared object mechanisms
  - **Pros**: Efficient & lightweight
  - **Cons**: Tedious & error-prone

Shared objects are often best used by infrastructure vs. app developers.
Which Java Mechanism(s) to Understand & Apply

- Framework developers may want to use the Java message passing mechanisms

  e.g., Android AsyncTask/HaMeR frameworks or Java ExecutorCompetitionService
Which Java Mechanism(s) to Understand & Apply

- Framework developers may want to use the Java message passing mechanisms
  - **Pros**: Flexible & decoupled
  - **Cons**: Time/space overhead

May incur higher context switching, synchronization, & data movement overhead
Which Java Mechanism(s) to Understand & Apply

- App developers most likely want to program with higher-level frameworks

**Applications**

**Additional Application Frameworks**

**Concurrency/Parallelism Frameworks**
- Java Threads & Synchronizers

**Execution Environment (JVM, Dalvik/ART, etc.)**

**System Libraries**

**Operating System Kernel**

- e.g., Java parallel streams & completable futures, RxJava, Project Reactor, etc.
Which Java Mechanism(s) to Understand & Apply

- App developers most likely want to program w/higher-level frameworks
  - **Pros**: Productivity & robustness
  - **Cons**: Time/space overhead & overly prescriptive

<table>
<thead>
<tr>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Application Frameworks</td>
</tr>
<tr>
<td>Concurrency/Parallelism Frameworks</td>
</tr>
<tr>
<td>Java Threads &amp; Synchronizers</td>
</tr>
<tr>
<td>Execution Environment (JVM, Dalvik/ART, etc.)</td>
</tr>
<tr>
<td>System Libraries</td>
</tr>
<tr>
<td>Operating System Kernel</td>
</tr>
</tbody>
</table>
Which Java Mechanism(s) to Understand & Apply

“Full stack” developers should be fluent with every layer!!

See en.everybodywiki.com/Full_S...
End of Evaluating the Concurrency & Parallelism Mechanisms in Java