Evaluation of 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

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

Diagram:
- Operating System Kernel
- System Libraries
- Execution Environment (JVM, Dalvik/ART, etc.)
- Concurrency/Parallelism Frameworks
- Java Threads & Synchronizers
- Additional Application Frameworks
- Applications
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

- Mobile app developers may want to program with higher-level frameworks

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

- Mobile app developers may want to program with higher-level frameworks
  - **Pros**: Productivity & robustness
  - **Cons**: Time/space overhead & overly prescriptive

---

### Applications
- Additional Application Frameworks
- Concurrency/Parallelism Frameworks
  - Java Threads & Synchronizers
- Execution Environment (JVM, Dalvik/ART, etc.)
- System Libraries
- Operating System Kernel
“Full stack” developers should understand concepts & mechanisms at each layer.
End of Evaluation of Concurrency & Parallelism in Java