Android Concurrency Frameworks: Motivation

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

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

• Know the motivations for Android concurrency & its concurrency frameworks
Motivation for Android Concurrency
Motivation for Android Concurrency

• Many Android apps require and/or benefit from concurrency
Motivation for Android Concurrency

• Many Android apps require and/or benefit from concurrency
• These apps perform long-duration operations and/or access remote resources in the background

Often (but not always), apps interact with servers that reside in the cloud
Motivation for Android Concurrency

• Many Android apps require and/or benefit from concurrency
• These apps perform long-duration operations and/or access remote resources in the background, e.g.
  • Play multimedia content on a device
    • e.g., music or videos
Motivation for Android Concurrency

• Many Android apps require and/or benefit from concurrency
  • These apps perform long-duration operations and/or access remote resources in the background, e.g.
    • Play multimedia content on a device
    • Synchronize contents of phone databases with cloud servers
      • e.g., email, contacts, calendar, MMS/SMS, etc.
Motivation for Android Concurrency

• Many Android apps require and/or benefit from concurrency

• These apps perform long-duration operations and/or access remote resources in the background, e.g.
  • Play multimedia content on a device
  • Synchronize contents of phone databases with cloud servers
  • Download & store images
Motivation for Android Concurrency

- Many Android apps require and/or benefit from concurrency
- These apps perform long-duration operations and/or access remote resources in the background, e.g.
  - Play multimedia content on a device
  - Synchronize contents of phone databases with cloud servers
  - Download & store images
  - Access web services
Motivation for Android Concurrency

• Concurrency benefits apps by overlapping *communication & computations*

See earlier lessons on “Motivation for Concurrency”
Motivation for Android Concurrency

• Concurrency benefits apps by overlapping *communication* & *computations*, e.g.
• Increase performance via multi-core parallelism

Motivation for Android Concurrency

- Concurrency benefits apps by overlapping *communication & computations*, e.g.
  - Increase performance via multi-core parallelism
  - Improve responsiveness by running long-duration operations in background thread(s)

See developer.android.com/training/articles/perf-anr.html
Motivation for Android Concurrency

- Concurrency benefits apps by overlapping *communication* & *computations*, e.g.
  - Increase performance via multi-core parallelism
  - Improve responsiveness by running long-duration operations in background thread(s)
  - Simplify program structure by allowing threads to block synchronously

[Image of a phone application interface with buttons labeled "Get Weather Sync" and "Get Weather Async", and text "nashville,tn".]
Motivation for Android Concurrency

• Concurrency benefits apps by overlapping *communication* & *computations*, e.g.
  • Increase performance via multi-core parallelism
  • Improve responsiveness by running long-duration operations in background thread(s)
  • Simplify program structure by allowing threads to block synchronously
    • Can yield more natural control flow & collaboration within an app

See [en.wikipedia.org/wiki/Control_flow](en.wikipedia.org/wiki/Control_flow)
Motivating Android’s Concurrency Frameworks
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints

See developer.android.com/guide/components/processes-and-threads.html#Threads
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
- “ANR” dialog is generated if the UI thread blocks too long

The UI thread can’t block for more than several seconds, so it can’t be used for long-duration operations

See developer.android.com/training/articles/perf-anr.html
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default

See blog.vogella.com/2012/02/22/android-strictmode-networkonmainthreadexception
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default
  - Non-UI threads can’t access UI toolkit components directly

*UI toolkit components aren’t thread-safe*

See android-developers.blogspot.com/2009/05/painless-threading.html
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default
  - Non-UI threads can’t access UI toolkit components directly

Java concurrency mechanisms alone cannot address these constraints!!

See [www.dre.vanderbilt.edu/~schmidt/LiveLessons/CPiJava](http://www.dre.vanderbilt.edu/~schmidt/LiveLessons/CPiJava)
Motivating Android Concurrency Frameworks

- Android’s concurrency frameworks also address design constraints, e.g.
  - “ANR” dialog is generated if the UI thread blocks too long
  - Network calls are disallowed on the UI thread by default
  - Non-UI threads can’t access UI toolkit components directly

See developer.android.com/guide/components/processes-and-threads.html#WorkerThreads
Motivating Android Concurrency Frameworks

- The “Buggy Downloader” app motivates the need for Android’s concurrency frameworks

See github.com/douglasraigschmidt/POSA/tree/master/ex/M4/BuggyDownloader
Motivating Android Concurrency Frameworks

- The “Buggy Downloader” app motivates the need for Android’s concurrency frameworks
  - “Buggy1” throws an exception since the image is downloaded in the UI thread
Motivating Android Concurrency Frameworks

• The “Buggy Downloader” app motivates the need for Android’s concurrency frameworks
  • “Buggy1” throws an exception since the image is downloaded in the UI thread
  • “Buggy2” throws an exception since a UI component is accessed via a background thread
End of Android Concurrency Frameworks: Motivation