Motivations for & Benefits of Concurrency in Android (Part 1)

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 Module

• Understand key motivations for developing concurrent software on Android
Learning Objectives in this Part of the Module

- Understand key motivations for developing concurrent software on Android
- Leverage advances in hardware & software components
Learning Objectives in this Part of the Module

• Understand key motivations for developing concurrent software on Android
• Leverage advances in hardware & software components
• Improve software quality attributes
Learning Objectives in this Part of the Module

- Understand key motivations for developing concurrent software on Android
- Leverage advances in hardware & software components
- Improve software quality attributes
- Support many popular services, apps, & capabilities
Motivations for Concurrent Software
Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components…

Commodity components are affordable & easy to obtain

See [www.dre.vanderbilt.edu/~schmidt/commodization.html](http://www.dre.vanderbilt.edu/~schmidt/commodization.html)
Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components…
  - Moore’s Law
    - # of transistors in a dense integrated circuit doubles every ~2 years

See en.wikipedia.org/wiki/Moore's_law
Motivations for Concurrent Software

• Leverage advances in commodity hardware/software components…
  • Moore’s Law

• Multi-core processors
  • Consist of 1 processor that contains 2+ CPUs that read & run instructions in parallel

See en.wikipedia.org/wiki/Multi-core_processor
Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components…
  - Moore’s Law
  - Multi-core processors
    - Consist of 1 processor that contains 2+ CPUs that read & run instructions in parallel
    - Have become essential as clock speeds haven’t kept pace with transistor density

See [www.gotw.ca/publications/concurrency-ddj.htm](http://www.gotw.ca/publications/concurrency-ddj.htm)
Motivations for Concurrent Software

• Leverage advances in commodity hardware/software components…
  • Moore’s Law
  • Multi-core processors
    • Consist of 1 processor that contains 2+ CPUs that read & run instructions in parallel
    • Have become essential as clock speeds haven’t kept pace with transistor density
  • It’s getting hard to buy a computer with only one core!

Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components…
  - Moore’s Law
  - Multi-core processors
- Multi-threaded operating systems & virtual machines
  - Manage concurrent access to multi-core hardware & system resources

See en.wikipedia.org/wiki/Operating_system & en.wikipedia.org/wiki/Virtual_machine
Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components...
  - Moore’s Law
  - Multi-core processors
- Multi-threaded operating systems & virtual machines
  - Manage concurrent access to multi-core hardware & system resources
  - A thread is a unit of execution for instruction streams that run on one or more processor cores

See en.wikipedia.org/wiki/Thread_(computing)
Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components...
  - Moore’s Law
  - Multi-core processors
  - Multi-threaded operating systems & virtual machines
  - Multi-threaded middleware
    - Enhance productivity & performance via reusable app-oriented services

See www.dre.vanderbilt.edu/~schmidt/concurrency-patterns.html
Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components…
  - Moore’s Law
  - Multi-core processors
  - Multi-threaded operating systems & virtual machines
  - Multi-threaded middleware

Knowledge of concurrency is needed to program multi-thread & multi-core systems effectively & efficiently
Motivations for Concurrent Software

- Leverage advances in commodity hardware/software components…
  - Moore’s Law
  - Multi-core processors
  - Multi-threaded operating systems & virtual machines
  - Multi-threaded middleware

Android supports concurrency at multiple layers!
Motivations for Concurrent Software

• ...to improve software quality attributes...

Software quality is the degree to which software possesses a desired combination of attributes

See www.sei.cmu.edu/reports/95tr021.pdf
Motivations for Concurrent Software

• ...to improve software quality attributes...
• Increase performance
  • e.g., by leveraging multi-core processors & overlapping computing & communication
Motivations for Concurrent Software

• ...to improve software quality attributes...
  • Increase performance
  • Improve responsiveness
  • e.g., don’t ignore user input while other processing is occurring in background threads
Motivations for Concurrent Software

• ...to improve software quality attributes...
  • Increase performance
  • Improve responsiveness
  • Simplify program structure
  • e.g., by using synchronous calls that are easier to understand & sustain over the lifecycle
Motivations for Concurrent Software

- ...in support of many popular services, apps, & capabilities
Motivations for Concurrent Software

• ...in support of many popular services, apps, & capabilities
• Scalable web servers for e-commerce & social media
Motivations for Concurrent Software

- ...in support of many popular services, apps, & capabilities
  - Scalable web servers for e-commerce & social media
  - Responsive UIs on mobile devices
Motivations for Concurrent Software

- ... in support of many popular services, apps, & capabilities
- Scalable web servers for e-commerce & social media
- Responsive UIs on mobile devices
- Sustainable architectures over the lifecycle
End of Motivations for & Benefits of Concurrency in Android (Part 1)
Motivations for & Benefits of Concurrency in Android (Part 2)

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 Module

• Understand the motivations for developing concurrent software on Android

• Recognize how concurrency can improve performance on Android
Using Concurrency to Increase Performance
Using Concurrency to Increase Performance

• “Performance” is a characterization of the amount of useful work that can be accomplished

See en.wikipedia.org/wiki/Computer_performance
Using Concurrency to Increase Performance

• “Performance” is a characterization of the amount of useful work that can be accomplished, e.g.
  • Decreasing response time for handling requests

- **Service time** – How long it takes to do requested work
- **Wait time** – How long the request has to wait for before it gets to run
- **Transmission time** – How long it takes to move request to computer doing the work & the response back to requestor

Using Concurrency to Increase Performance

• “Performance” is a characterization of the amount of useful work that can be accomplished, e.g.
  • Decreasing response time for handling requests
  • Increasing the amount of work that can be performed within a given time

See en.wikipedia.org/wiki/Computer_performance#Throughput
Using Concurrency to Increase Performance

• “Performance” is a characterization of the amount of useful work that can be accomplished, e.g.
  • Decreasing response time for handling requests
  • Increasing the amount of work that can be performed within a given time

See en.wikipedia.org/wiki/Up_to_eleven for more on maximizing performance
Using Concurrency to Increase Performance

- Decreasing response time & increasing the amount of work performed within a given time are common motivations for using concurrency.
Using Concurrency to Increase Performance

- Decreasing response time & increasing the amount of work performed within a given time are common motivations for using concurrency.

Optimizing these properties requires an understanding of patterns & tradeoffs amongst constraints & quality attributes.
Using Concurrency to Increase Performance

- Performance can be accelerated via parallel processing

See en.wikipedia.org/wiki/Parallel_computing
Using Concurrency to Increase Performance

- Performance can be accelerated via parallel processing, e.g.
- Performing computations simultaneously
Using Concurrency to Increase Performance

- Performance can be accelerated via parallel processing, e.g.
- Performing computations simultaneously
  - Particularly for computations with no inter-dependencies

See en.wikipedia.org/wiki/Embarrassingly_parallel
Using Concurrency to Increase Performance

- Performance can be accelerated via parallel processing, e.g.
  - Performing computations simultaneously
  - Dividing a large problem into multiple smaller problems that can be processed in parallel

See en.wikipedia.org/wiki/MapReduce
Using Concurrency to Increase Performance

- Performance can be accelerated via parallel processing, e.g.
  - Performing computations simultaneously
- Dividing a large problem into multiple smaller problems that can be processed in parallel
  - e.g., image rendering

See [en.wikipedia.org/wiki/Parallel_rendering](http://en.wikipedia.org/wiki/Parallel_rendering)
Using Concurrency to Increase Performance

- Android enables parallelism by overlapping computation & communication via two concurrency frameworks.
Using Concurrency to Increase Performance

- Android enables parallelism by overlapping computation & communication via two concurrency frameworks, e.g.
  - HaMeR framework
    - “Handlers, Messages, & Runnables”

Operations run in one or more threads & publish their results to the UI thread

See code.tutsplus.com/tutorials/concurrency-on-android-using-hamer-framework--cms-27129
Using Concurrency to Increase Performance

- Android enables parallelism by overlapping computation & communication via two concurrency frameworks, e.g.
  - HaMeR framework
    - “Handlers, Messages, & Runnables”
  - Works with Java threads

See [docs.oracle.com/javase/tutorial/essential/concurrency/threads.html](docs.oracle.com/javase/tutorial/essential/concurrency/threads.html)
Android enables parallelism by overlapping computation & communication via two concurrency frameworks, e.g.

- HaMeR framework
- AsyncTask framework

**Using Concurrency to Increase Performance**

Operations run in one or more threads & publish results to the UI thread without using threads, handlers, messages, and/or runnables

Using Concurrency to Increase Performance

- Android enables parallelism by overlapping computation & communication via two concurrency frameworks, e.g.
  - HaMeR framework
  - AsyncTask framework
- Can be used with the Java Executor framework to run tasks in a pool of threads

See docs.oracle.com/javase/tutorial/essential/concurrency/executors.html
End of Motivations for & Benefits of Concurrency in Android (Part 2)