Introduction to Java 8
Parallelism Frameworks

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 Lesson

- Recognize how Java 8 applies functional programming features to its parallelism frameworks
Learning Objectives in this Lesson

• Recognize how Java 8 applies functional programming features to its parallelism frameworks, e.g.
  • **Fork-join pools**
Learning Objectives in this Lesson

• Recognize how Java 8 applies functional programming features to its parallelism frameworks, e.g.
  • Fork-join pools
  • Parallel streams

```java
filter(not(this::urlCached))
map(this::downloadImage)
flatMap(this::applyFilters)
collect(toList())
```
Learning Objectives in this Lesson

- Recognize how Java 8 applies functional programming features to its parallelism frameworks, e.g.
  - Fork-join pools
  - Parallel streams
  - Completable futures

Completable futures also provide a reactive asynchrony programming model
Learning Objectives in this Lesson

• Recognize how Java 8 applies functional programming features to its parallelism frameworks, e.g.

• Know how these features are applied in several example case study apps

New & Improved

List of Filters to Apply
NullFilter GrayScaleFilter

List of URLs to Download

Persistent Data Store
Overview of Java 8 Parallelism Frameworks
Overview of Java 8 Parallelism Frameworks

- Java 7 introduced the fork-join pool

See [www.infoq.com/interviews/doug-lea-fork-join](http://www.infoq.com/interviews/doug-lea-fork-join)
Overview of Java 8 Parallelism Frameworks

- Java 7 introduced the fork-join pool
- This framework provides high performance, fine-grained task execution for data parallelism

See www.dre.vanderbilt.edu/~schmidt/PDF/DataParallelismInJava.pdf
Overview of Java 8 Parallelism Frameworks

- Java 7 introduced the fork-join pool
- This framework provides high performance, fine-grained task execution for data parallelism
- It supports parallel programming by solving problems via “divide & conquer”

```java
Result solve(Problem problem) {
    if (problem is small)
        directly solve problem
    else {
        a. split problem into independent parts
        b. fork new sub-tasks to solve each part
        c. join all sub-tasks
        d. compose result from sub-results
    }
}
```

See en.wikipedia.org/wiki/Divide_and_conquer_algorithm
Overview of Java 8 Parallelism Frameworks

- Java 7 introduced the fork-join pool
  - This framework provides high performance, fine-grained task execution for data parallelism
  - It supports parallel programming by solving problems via “divide & conquer”
  - Employs *work-stealing* to optimize multi-core processor performance

See gee.cs.oswego.edu/dl/papers/fj.pdf
• Java 8 adds two new parallelism frameworks related to functional programming

See www.ibm.com/developerworks/library/j-jvmc2
Java 8 adds two new parallelism frameworks related to functional programming

1. **Parallel streams**

   ```java
   filter(not(this::urlCached))
   map(this::downloadImage)
   flatMap(this::applyFilters)
   collect(toList())
   ```

See [docs.oracle.com/javase/tutorial/collections_streams/parallelism.html](docs.oracle.com/javase/tutorial/collections_streams/parallelism.html)
Overview of Java 8 Parallelism Frameworks

• Java 8 adds two new parallelism frameworks related to functional programming

1. Parallel streams
• Partitions a stream into multiple substreams that run independently & combine into a “reduced” result

```
filter(not(this::urlCached))
map(this::downloadImage)
flatMap(this::applyFilters)
collect(toList())
```
Overview of Java 8 Parallelism Frameworks

- Java 8 adds two new parallelism frameworks related to functional programming

1. Parallel streams
   - Partitions a stream into multiple substreams that run independently & combine into a “reduced” result
   - Chunks of data in the substreams can be mapped to multiple threads (& cores)

```
filter(not(this::urlCached))
map(this::downloadImage)
flatMap(this::applyFilters)
collect(toList())
```
Overview of Java 8 Parallelism Frameworks

- Java 8 adds two new parallelism frameworks related to functional programming

1. Parallel streams
   - Partitions a stream into multiple substreams that run independently & combine into a “reduced” result
   - Chunks of data in the substreams can be mapped to multiple threads (& cores)
   - Leverage the common fork-join pool

See dzone.com/articles/common-fork-join-pool-and-streams
Java 8 adds two new parallelism frameworks related to functional programming

1. **Parallel streams**
   - Partitions a stream into multiple substreams that run independently & combine into a “reduced” result
   - Chunks of data in the substreams can be mapped to multiple threads (& cores)
   - Leverage the common fork-join pool

Parallel streams provides fine-grained data parallelism functional programming
Java 8 adds two new parallelism frameworks related to functional programming:

1. **Parallel streams**

2. **CompletableFuture**

See [docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html](docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html)
Overview of Java 8 Parallelism Frameworks

- Java 8 adds two new parallelism frameworks related to functional programming

1. Parallel streams

2. Completable futures
   - Supports dependent actions that trigger upon completion of async operations

See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletionStage.html
Overview of Java 8 Parallelism Frameworks

- Java 8 adds two new parallelism frameworks related to functional programming
  
  1. Parallel streams
  
  2. Completable futures
     - Supports dependent actions that trigger upon completion of async operations
     - Async operations can run in parallel in thread pools

See [www.nurkiewicz.com/2013/05/java-8-definitive-guide-to.html](http://www.nurkiewicz.com/2013/05/java-8-definitive-guide-to.html)
Overview of Java 8 Parallelism Frameworks

- Java 8 adds two new parallelism frameworks related to functional programming

1. Parallel streams
2. Completable futures

- Supports dependent actions that trigger upon completion of async operations
- Async operations can run in parallel in thread pools

Java 8 completable futures & streams can be combined to good effects!!
Overview of Java 8 Parallelism Frameworks

- These frameworks often eliminate the use of synchronization or explicit threading when developing parallel apps!

Alleviates many accidental & inherent complexities of parallel programming
Overview of Java 8 Parallelism Frameworks

- Both Java 8 frameworks use the fork-join pool framework by default

See [www.oracle.com/technetwork/articles/java/fork-join-422606.html](http://www.oracle.com/technetwork/articles/java/fork-join-422606.html)
Summary of Example Case Study Apps
Summary of Example Case Study Apps

- SearchStreamGang case study uses regular expressions to find phrases in the complete works of William Shakespeare in parallel.

Summary of Example Case Study Apps

- ImageCounter recursively crawls web pages counting # of images in parallel

See github.com/douglascraigschmidt/LiveLessons/tree/master/Java8/ex19
Summary of Example Case Study Apps

- ImageStreamGang shows how the StreamGang framework can be combined with Java 8 streams & completable futures to download, filter, store, & display images in parallel

```
List of URLs to Download

map(this::checkUrlCachedAsync)

map(this::downloadImageAsync)

flatMap(this::applyFiltersAsync)

collect(toFuture())

thenAccept(this::log)

Parallel Streams

filter(not(this::urlCached))

map(this::downloadImage)

flatMap(this::applyFilters)

collect(toList())
```

End of Introduction to Java 8
Parallelism Frameworks