Overview of Parallel Programming 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

- Understand the meaning of key parallel programming concepts
- Know when to apply parallelism
- Recognize the parallel programming frameworks supported by Java

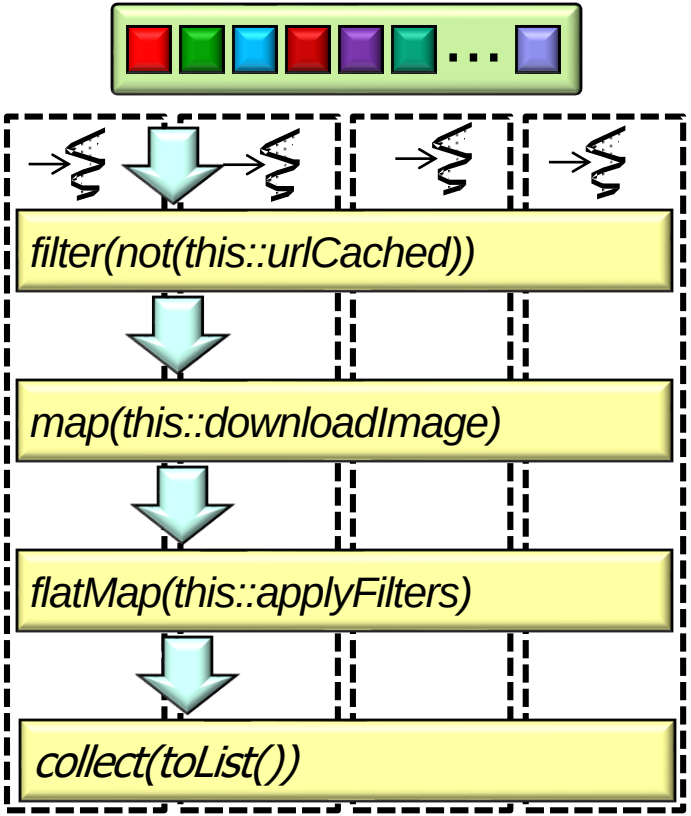
See www.dre.vanderbilt.edu/~schmidt/frameworks.html

Learning Objectives in this Part of the Lesson

- Recognize the parallelism frameworks supported by Java, e.g.
  - **Fork-join pools**

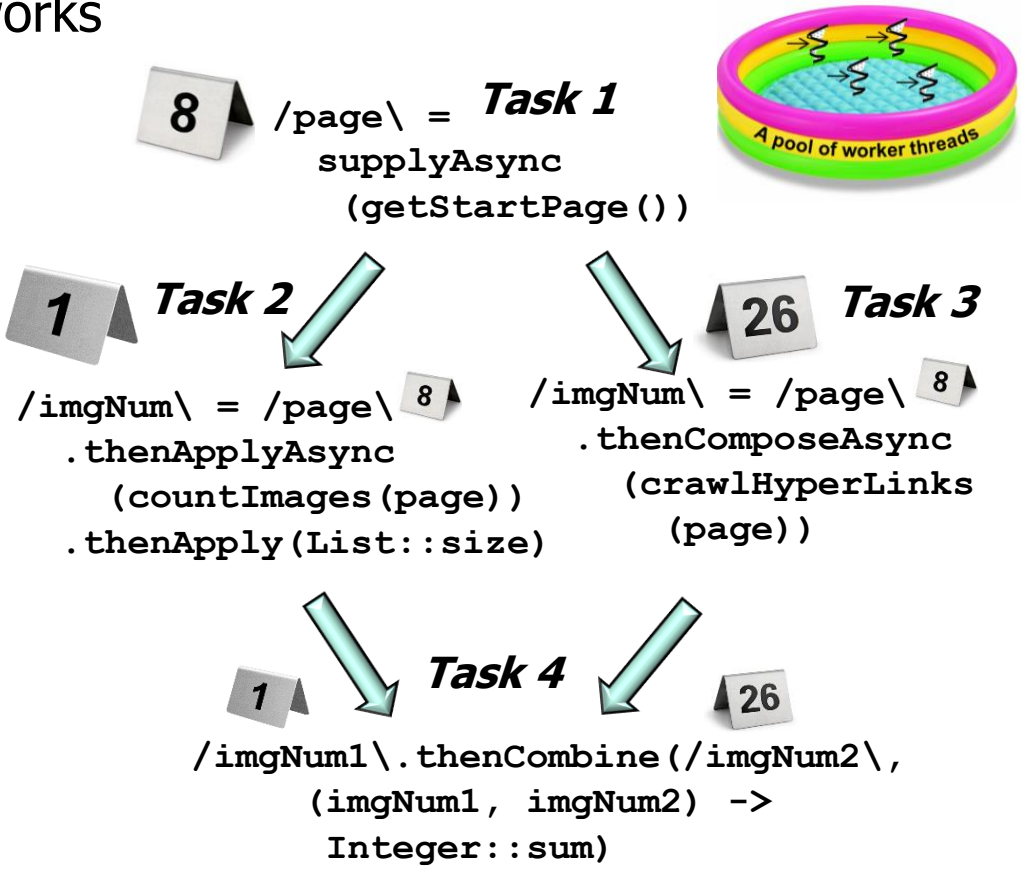
Learning Objectives in this Part of the Lesson

- Recognize the parallelism frameworks supported by Java, e.g.
  - Fork-join pools
  - **Parallel streams**



Learning Objectives in this Part of the Lesson

- Recognize the parallelism frameworks supported by Java, e.g.
  - Fork-join pools
  - Parallel streams
  - **Completable futures**



Completable futures also provide a reactive asynchrony programming model

Overview of Java Parallelism Frameworks

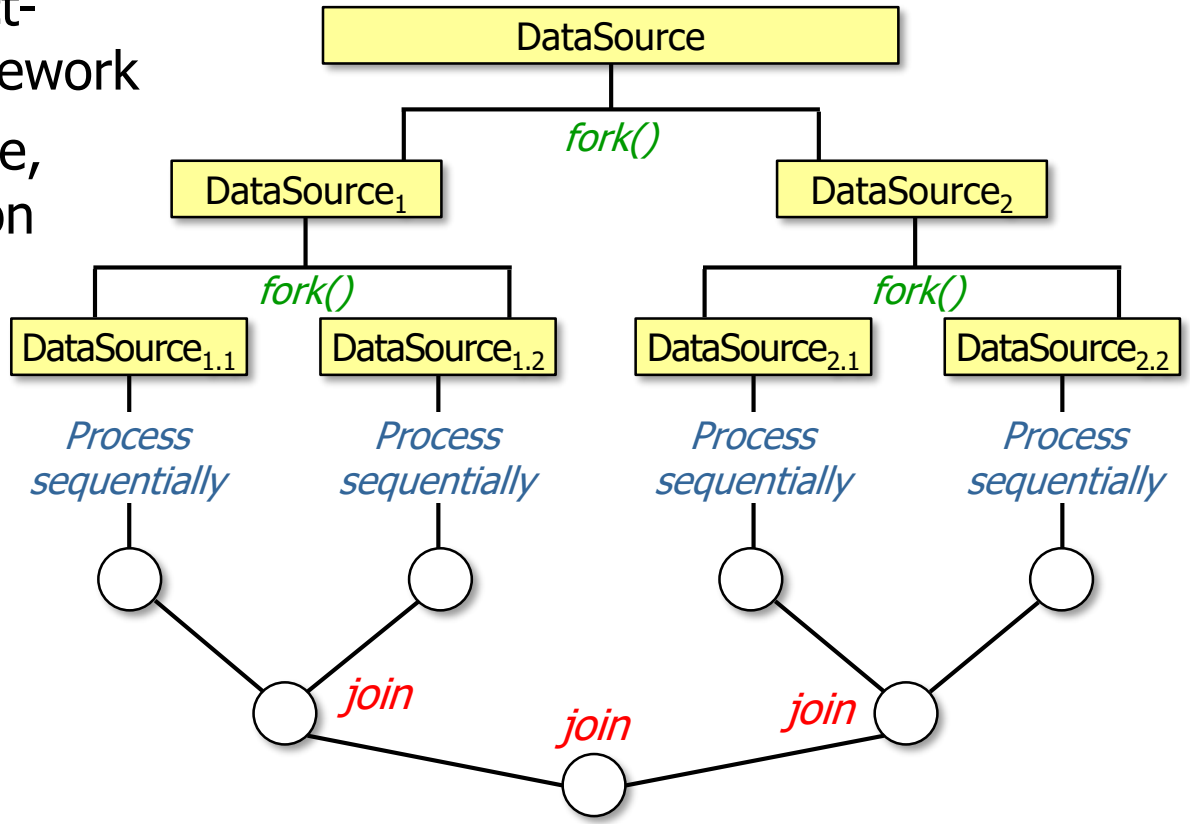
Overview of Java Parallelism Frameworks

- Java 7 introduced the object-oriented fork-join pool framework

See www.infoq.com/interviews/doug-lea-fork-join

Overview of Java Parallelism Frameworks

- Java 7 introduced the object-oriented fork-join pool framework
  - Provides high performance, fine-grained task execution for data parallelism

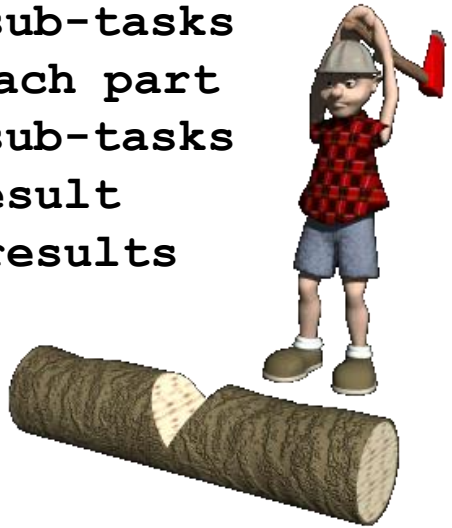


See www.dre.vanderbilt.edu/~schmidt/PDF/DataParallelismInJava.pdf

Overview of Java Parallelism Frameworks

- Java 7 introduced the object-oriented fork-join pool framework
  - Provides high performance, fine-grained task execution for data parallelism
  - Supports parallel programming by solving problems via "divide & conquer"

```
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

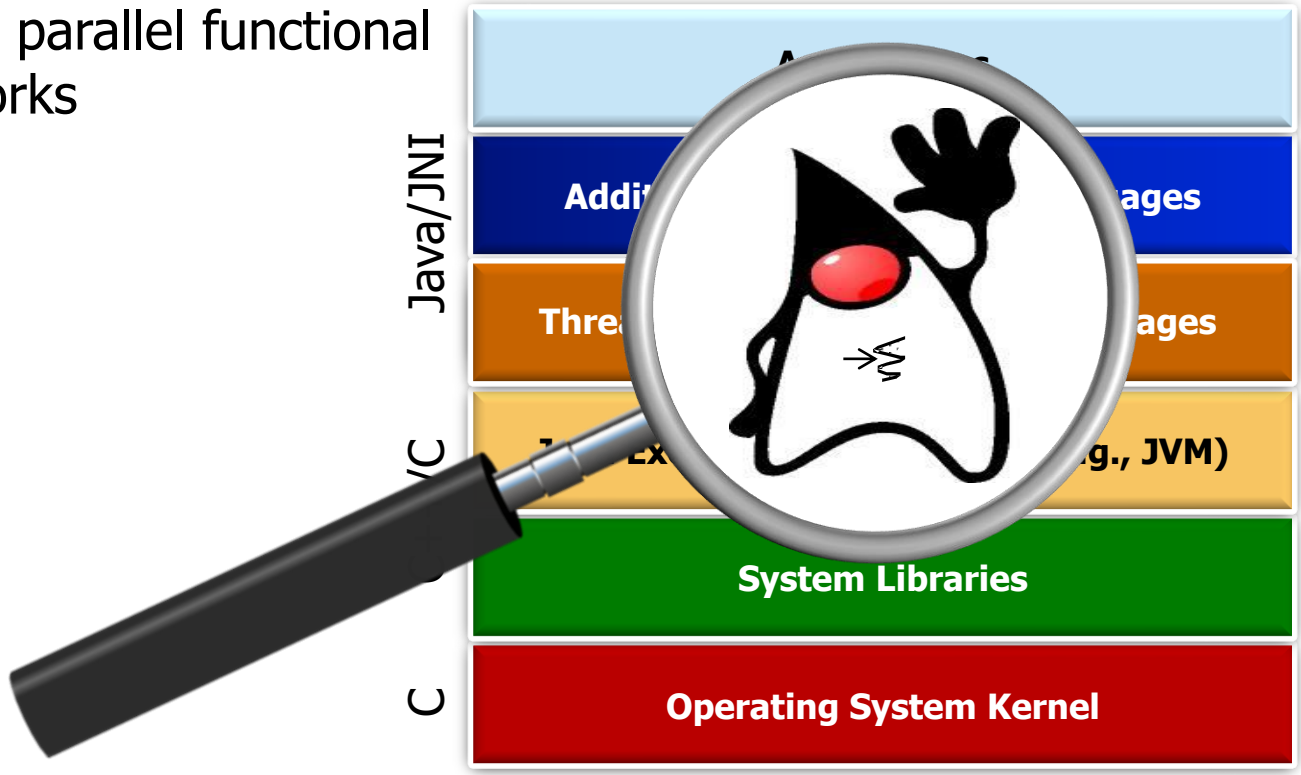
Overview of Java Parallelism Frameworks

- Java 7 introduced the object-oriented fork-join pool framework
  - Provides high performance, fine-grained task execution for data parallelism
  - 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

Overview of Java Parallelism Frameworks

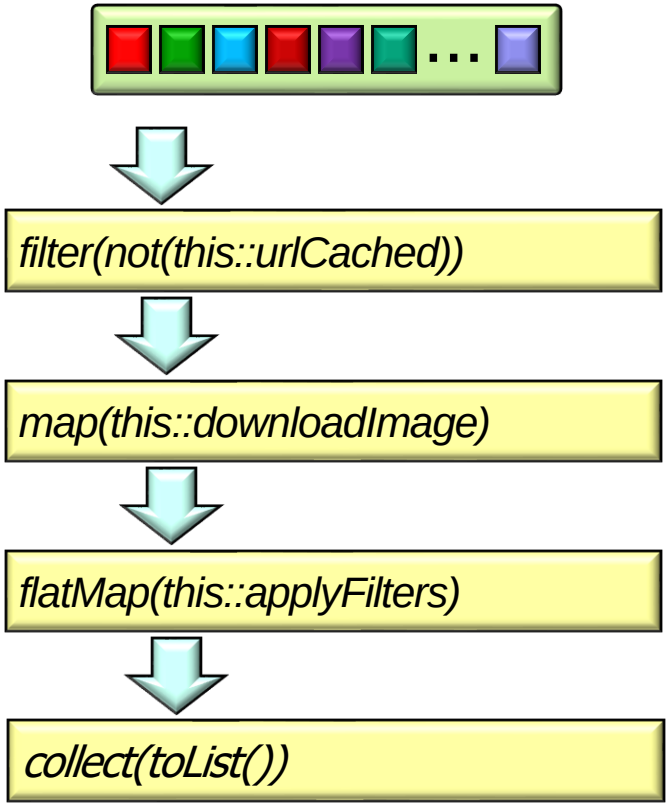
- Java 8 added two new parallel functional programming frameworks



See www.ibm.com/developerworks/library/j-jvmc2

Overview of Java Parallelism Frameworks

- Java 8 added two new parallel functional programming frameworks
  1. **Parallel streams**



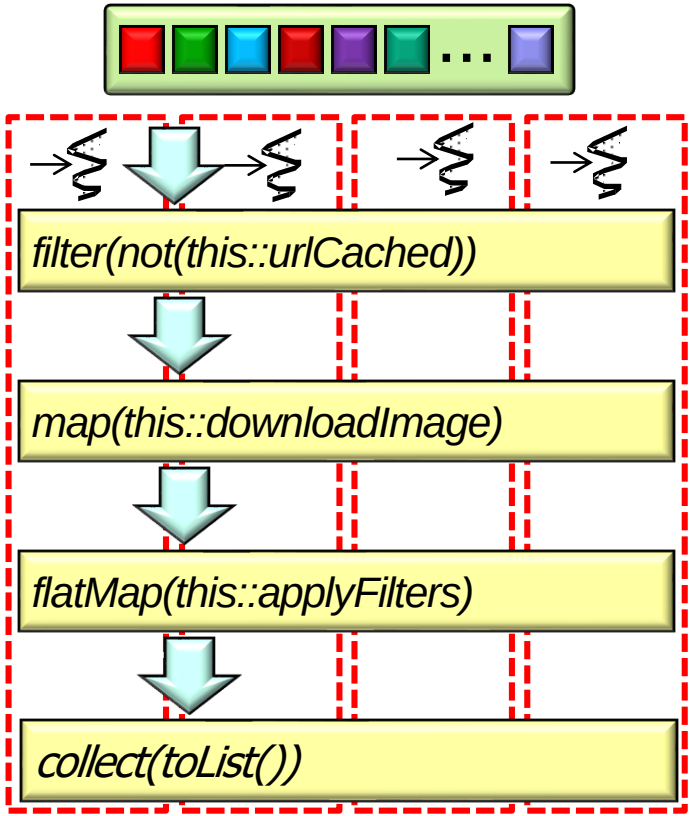
See docs.oracle.com/javase/tutorial/collections/streams/parallelism.html

Overview of Java Parallelism Frameworks

- Java 8 added two new parallel functional programming frameworks

1. Parallel streams

  - Partitions a stream into multiple substreams that run independently & combine into a "reduced" result

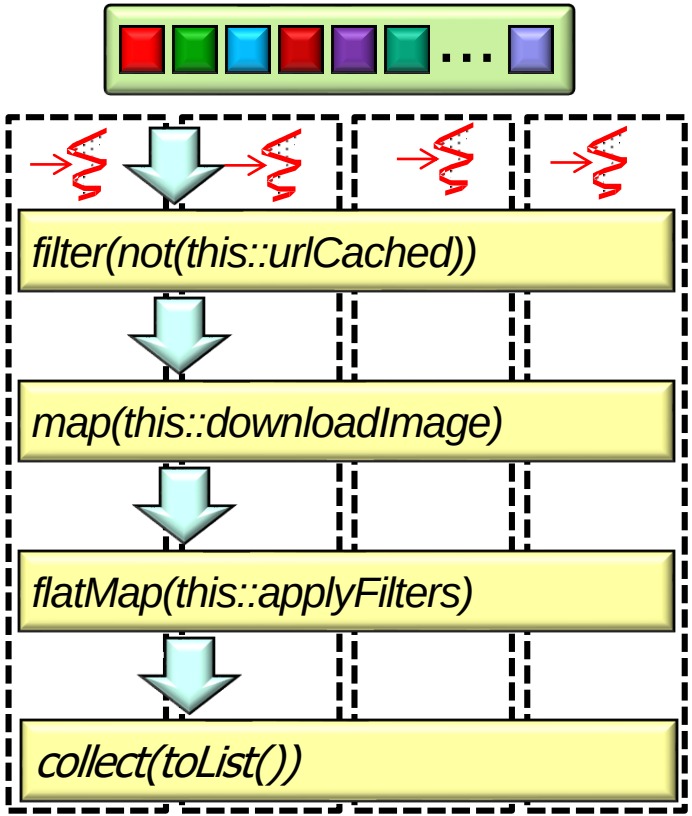


Overview of Java Parallelism Frameworks

- Java 8 added two new parallel functional programming frameworks

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)

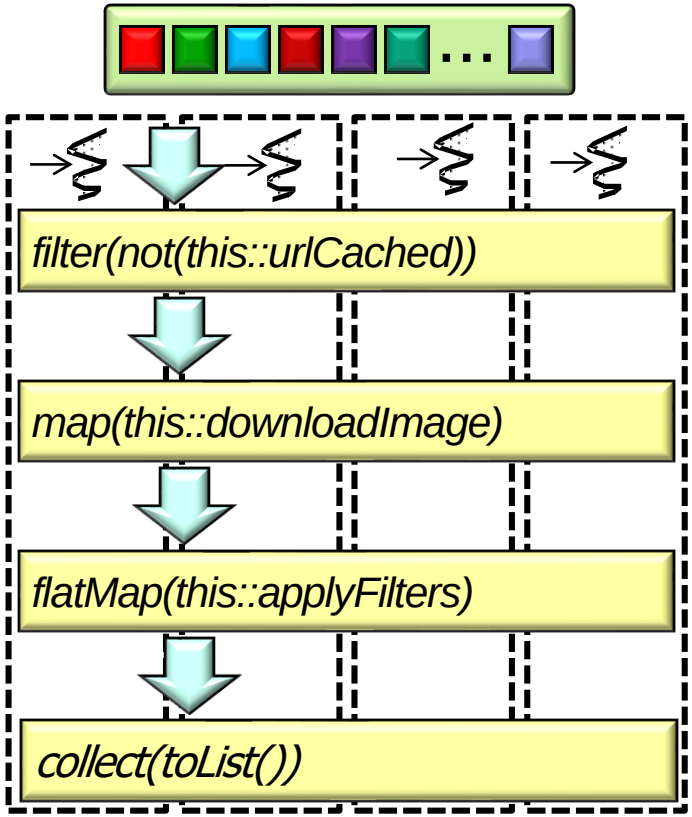


Overview of Java Parallelism Frameworks

- Java 8 added two new parallel functional programming frameworks

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)
  - Leverages the common fork-join pool



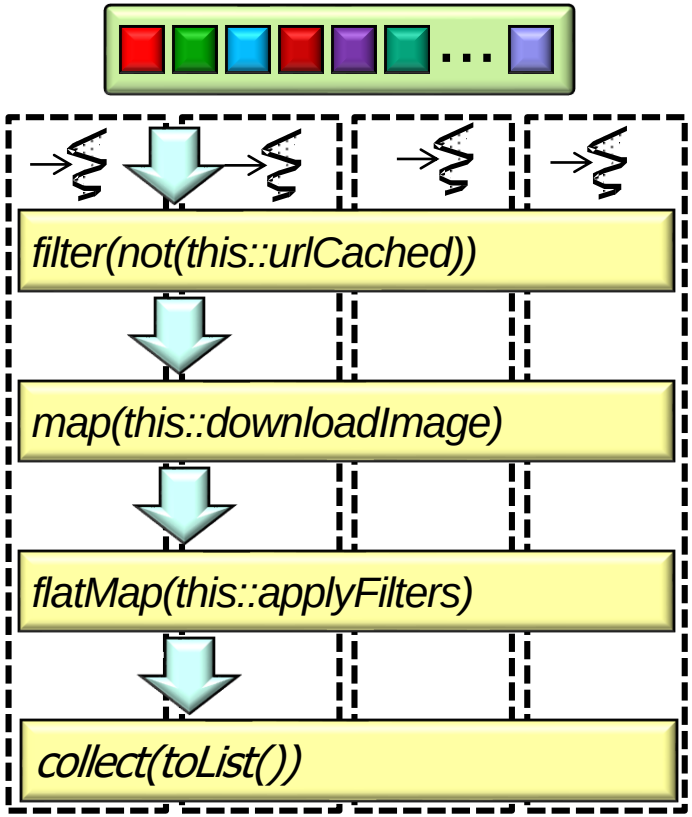
See dzone.com/articles/common-fork-join-pool-and-streams

Overview of Java Parallelism Frameworks

- Java 8 added two new parallel functional programming frameworks

1. Parallel streams

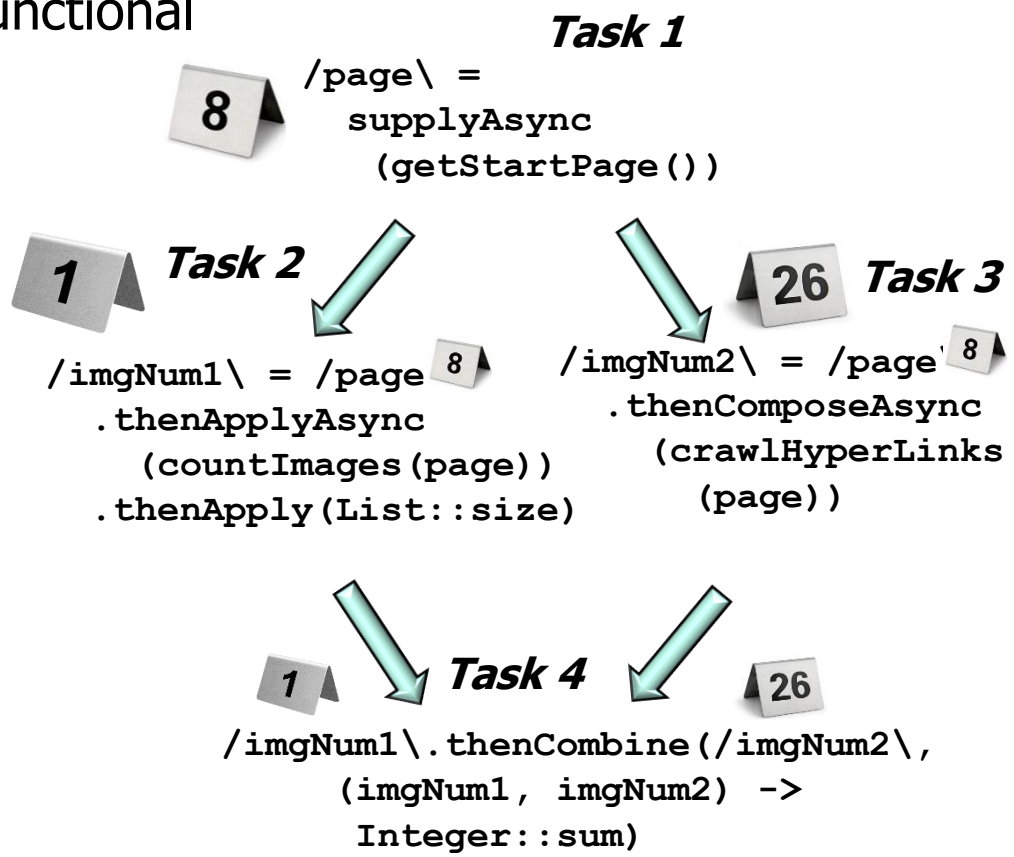
  - Partitions a stream into multiple substreams that run independently & combine into a "reduced" result
  - Chunks of data in the substreams can be mapped to multiple threads (& cores)
  - Leverages the common fork-join pool



Parallel streams provides fine-grained data parallelism functional programming

Overview of Java Parallelism Frameworks

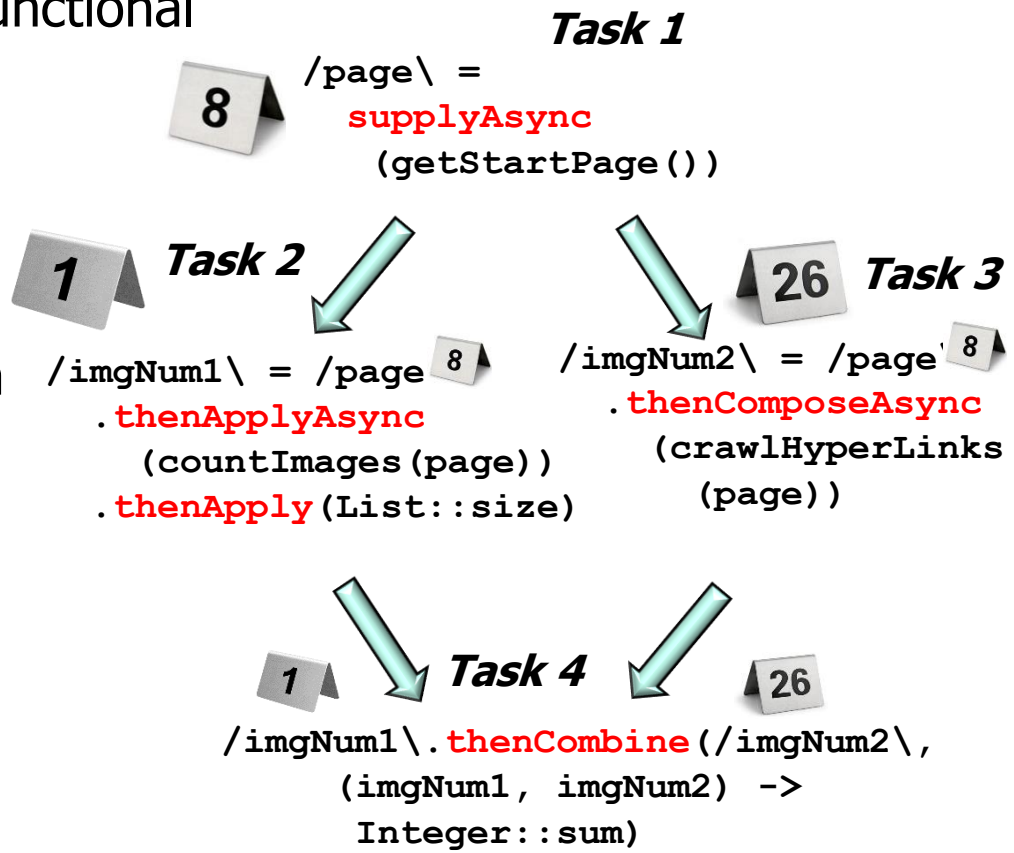
- Java 8 added two new parallel functional programming frameworks
  1. Parallel streams
  2. **Completable futures**



See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html

Overview of Java Parallelism Frameworks

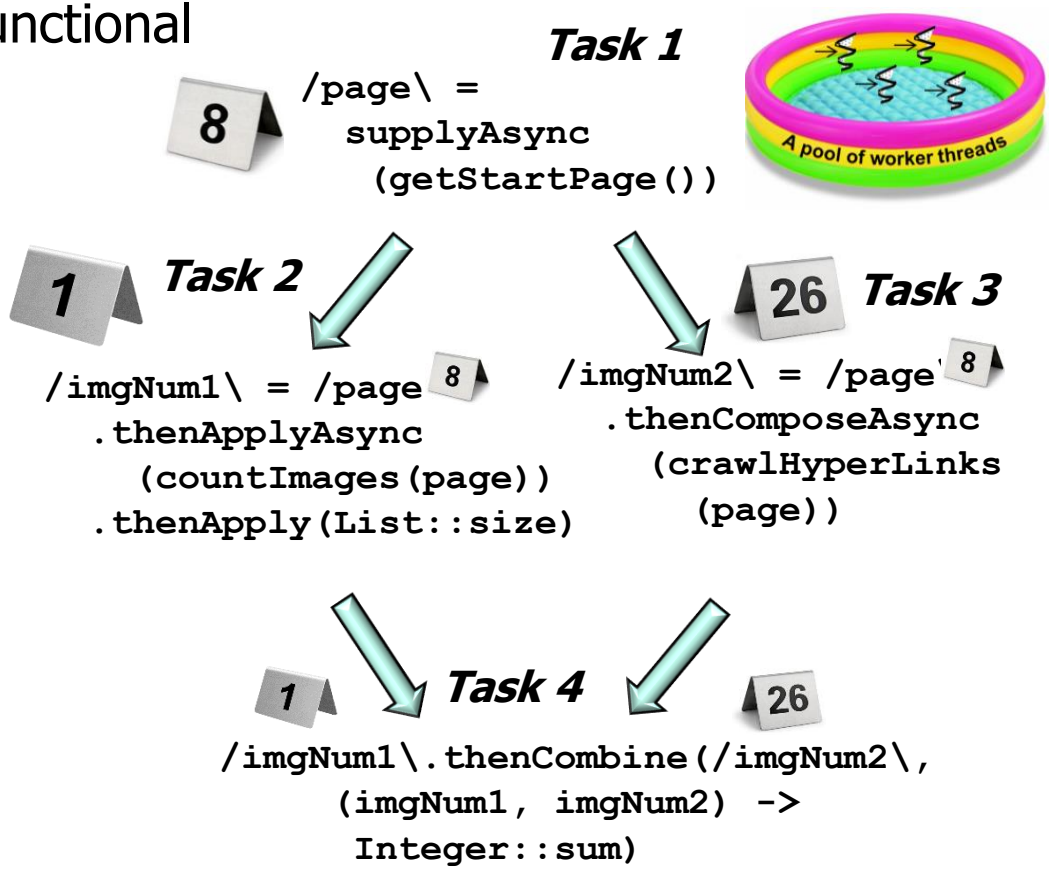
- Java 8 added two new parallel functional programming frameworks
  1. Parallel streams
  2. **Completable futures**
     - Supports dependent actions that trigger upon completion of async operations



See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletionStage.html

Overview of Java Parallelism Frameworks

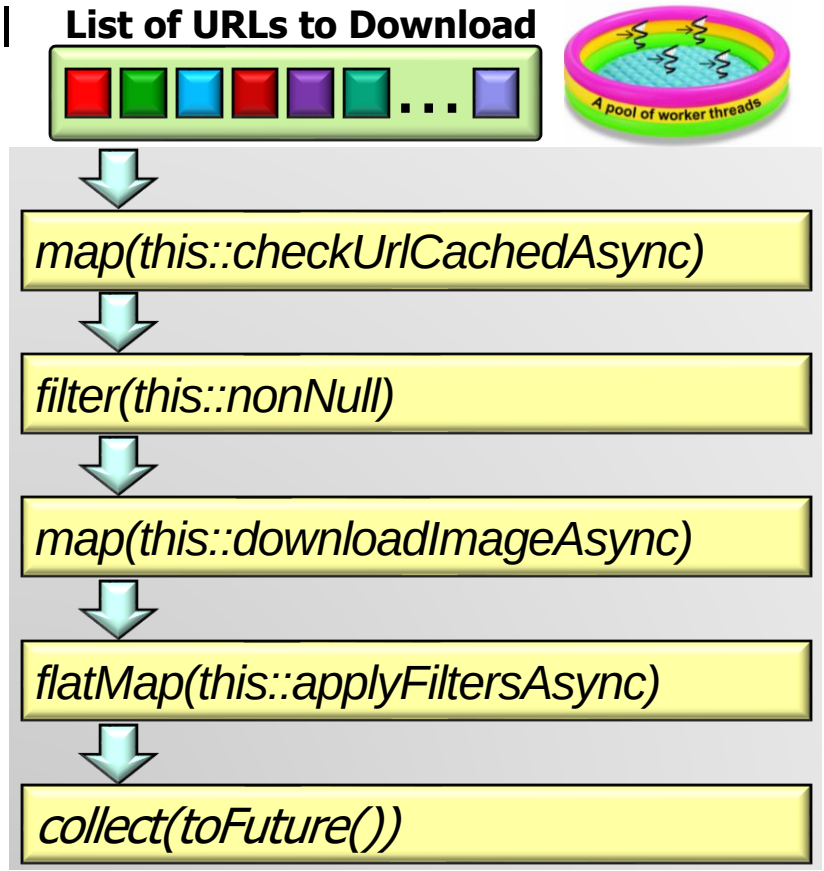
- Java 8 added two new parallel functional programming frameworks
  1. Parallel streams
  2. **Completable futures**
     - Supports dependent actions that trigger upon completion of async operations
     - Async operations can run in parallel in thread pools



See www.nurkiewicz.com/2013/05/java-8-definitive-guide-to.html

Overview of Java Parallelism Frameworks

- Java 8 added two new parallel functional programming frameworks
  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 completable futures & streams can be combined to good effects!!

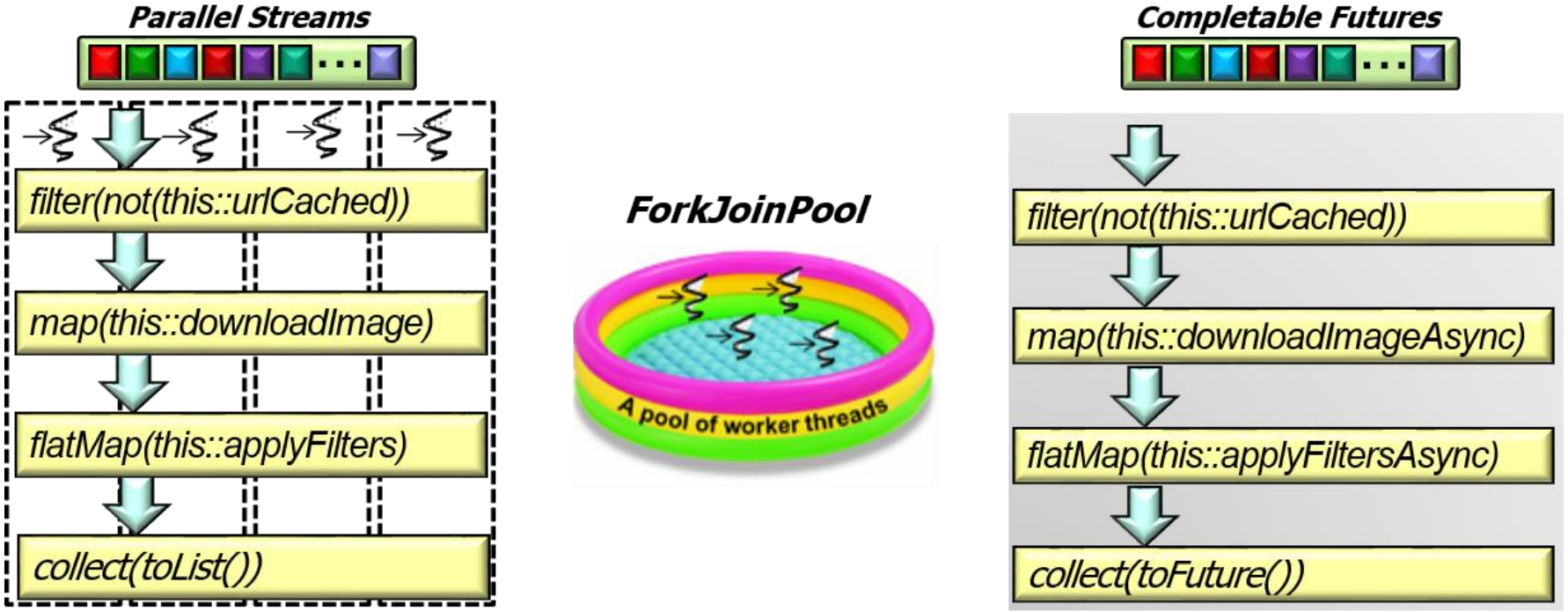
Overview of Java Parallelism Frameworks

- These Java 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 Parallelism Frameworks

- Java parallel streams & completable future functional frameworks use the object-oriented fork-join pool framework by default



See www.oracle.com/technetwork/articles/java/fork-join-422606.html

End of Overview of Parallelism in Java