Introduction to Java 8 Concurrency & Parallelism Frameworks

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

Professor of Computer Science

Institute for Software Integrated Systems

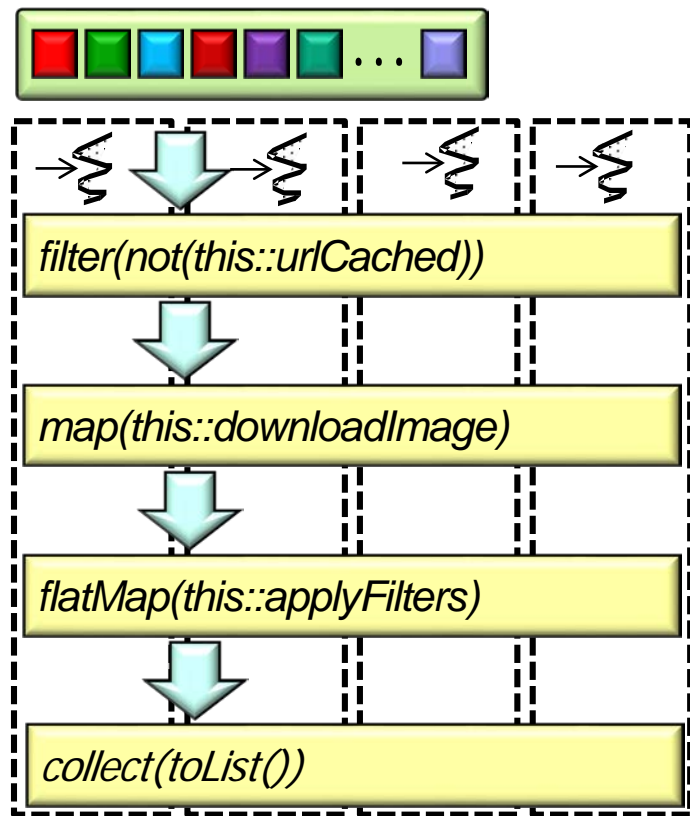**Vanderbilt University Nashville, Tennessee, USA**

Learning Objectives in this Lesson

- Recognize how Java 8 applies functional programming features for its concurrency & parallelism frameworks

Learning Objectives in this Lesson

- Recognize how Java 8 applies functional programming features for its concurrency & parallelism frameworks, e.g.
  - Parallel streams

filter(not(this::urlCached))

map(this::downloadImage)
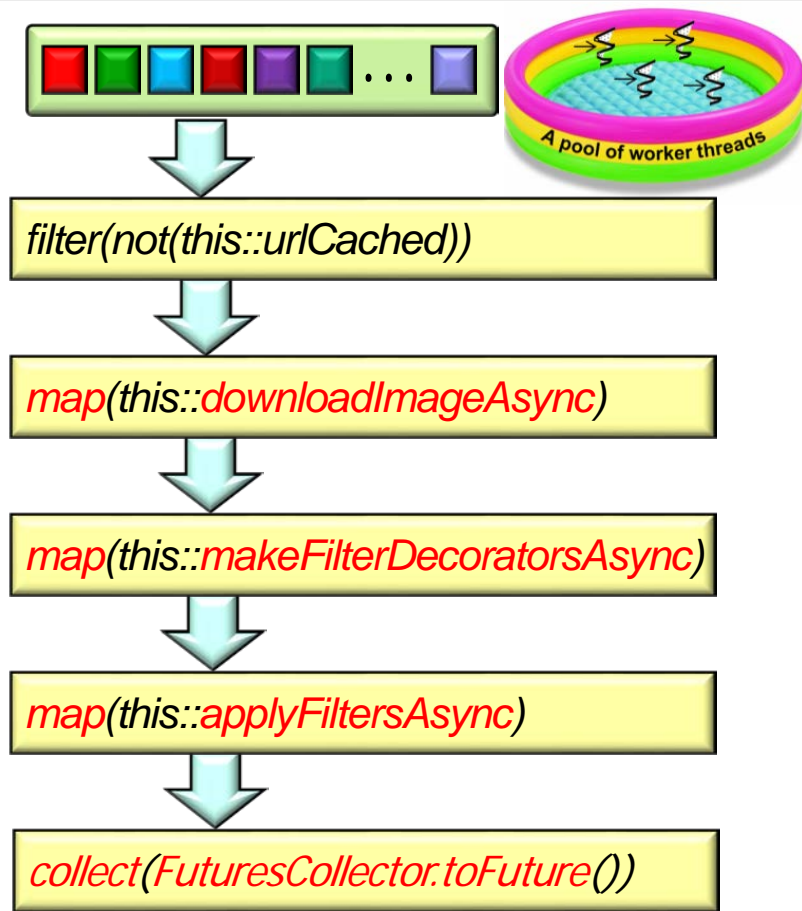
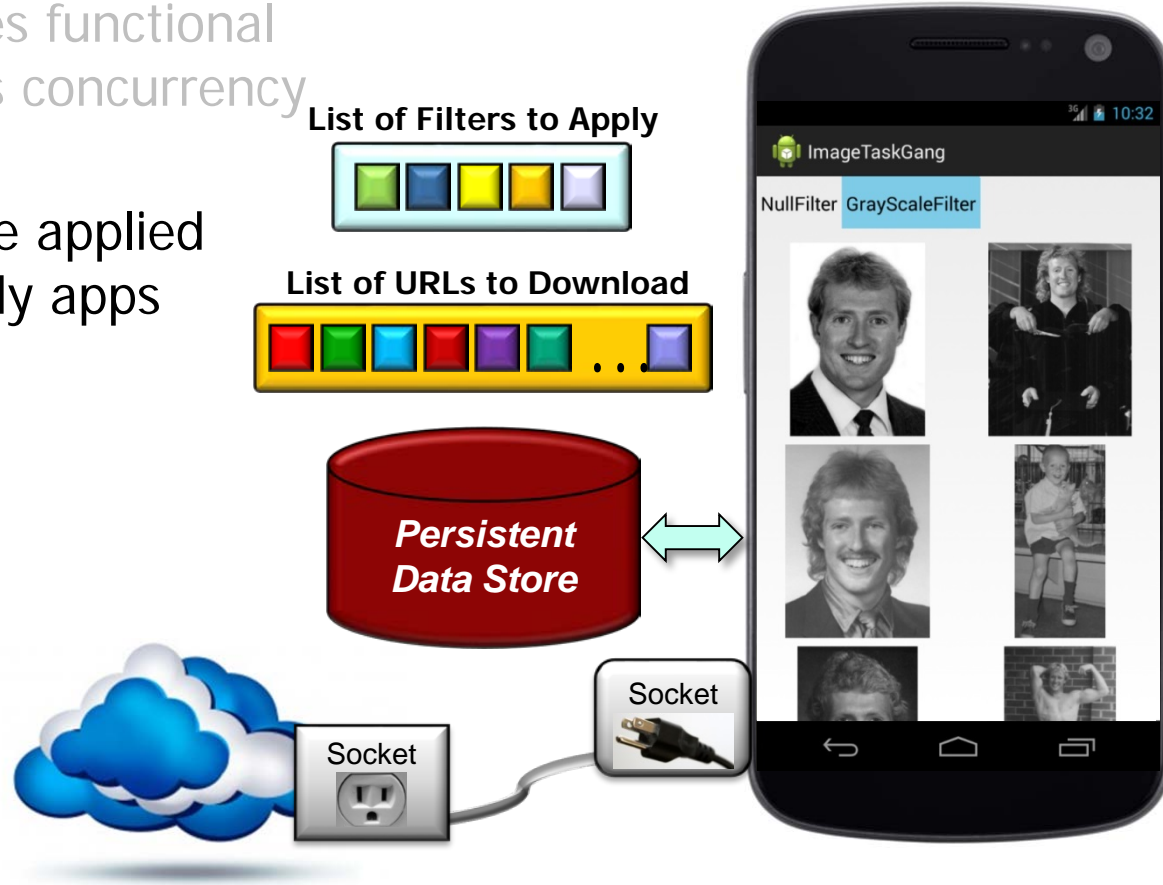flatMap(this::applyFilters)

collect(toList())

Learning Objectives in this Lesson

- Recognize how Java 8 applies functional programming features for its concurrency & parallelism frameworks, e.g.

  - Parallel streams

  - Completable futures


```
filter(not(this::urlCached))
```

```
map(this::downloadImageAsync)
```

```
map(this::makeFilterDecoratorsAsync)
```

```
map(this::applyFiltersAsync)
```

```
collect(FuturesCollector.toFuture())
```

Learning Objectives in this Lesson

- Recognize how Java 8 applies functional programming features for its concurrency & parallelism frameworks

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

New & Improved

List of Filters to Apply

List of URLs to Download

...

Persistent Data Store

ImageTaskGang

NullFilter GrayScaleFilter

Socket

Socket

10:32

Overview of Java 8 Concurrency & Parallelism Frameworks

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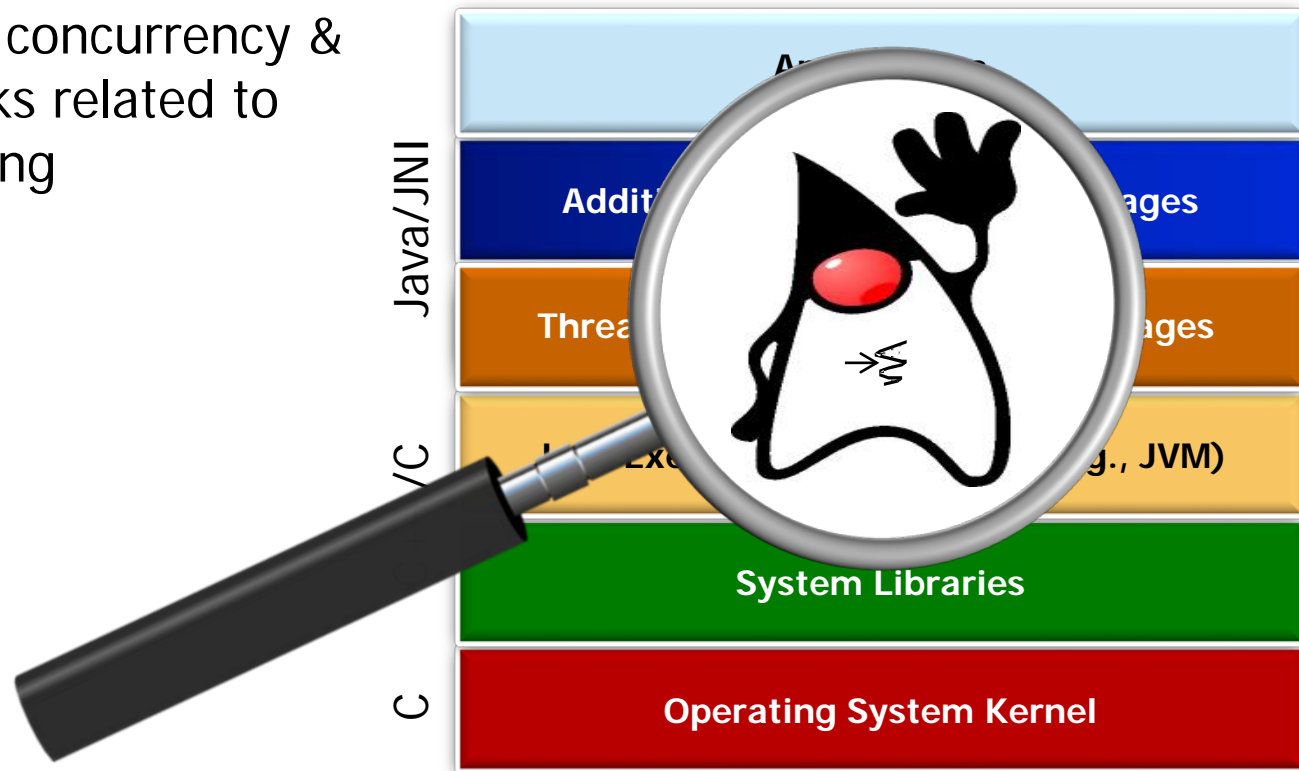
See www.ibm.com/developerworks/library/j-jvmc2

Overview of Java 8 Concurrency & Parallelism Frameworks

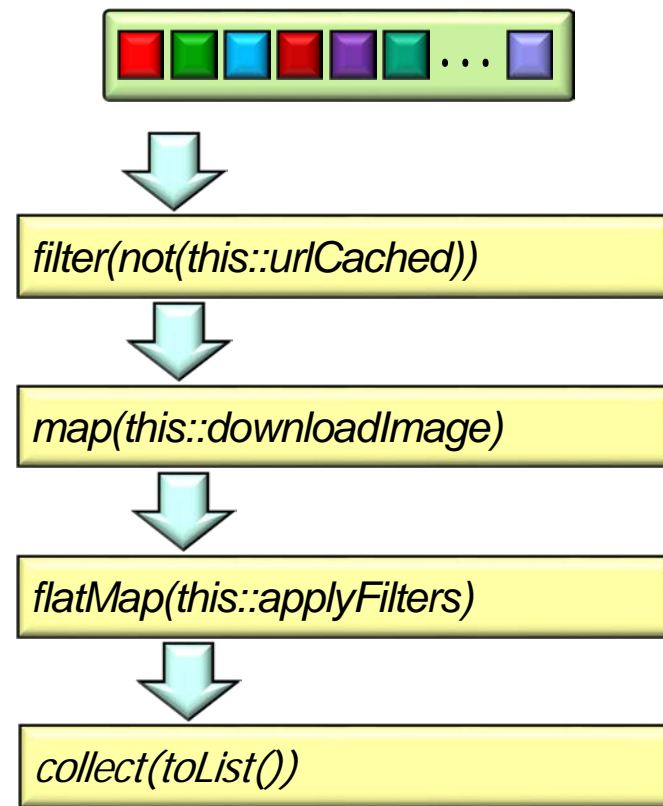- Java 8 adds two new concurrency & parallelism frameworks related to functional programming

1. **Parallel streams**


```
filter(not(this::urlCached))

map(this::downloadImage)

flatMap(this::applyFilters)

collect(toList())
```
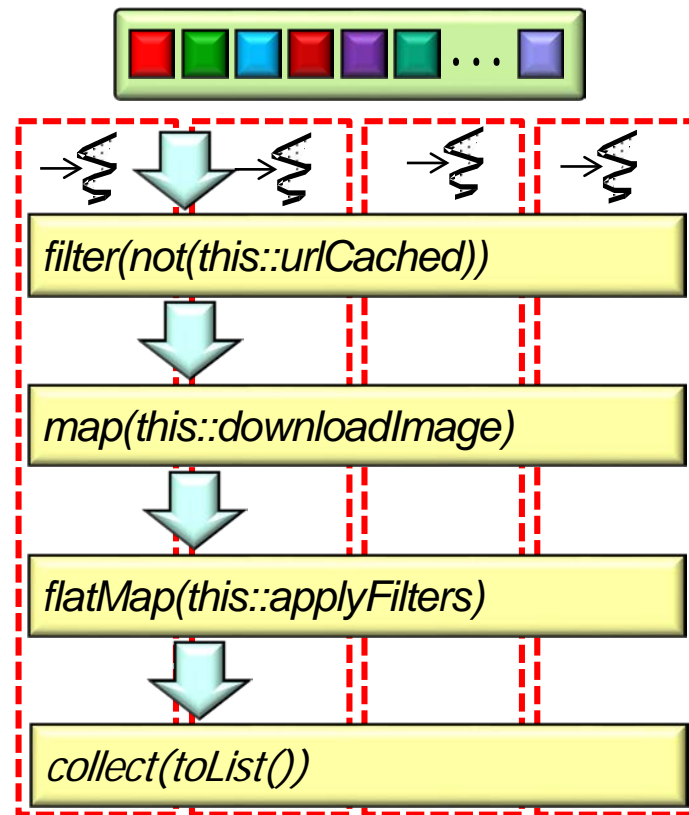
See docs.oracle.com/javase/tutorial/collections/streams/parallelism.html

Overview of Java 8 Concurrency & Parallelism Frameworks

- Java 8 adds two new concurrency & 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)

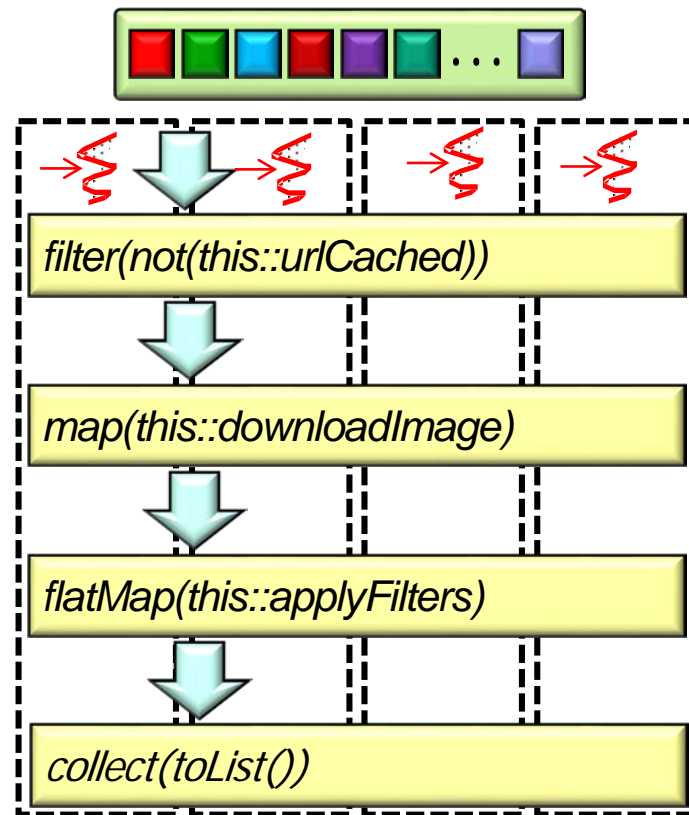*flatMap(this::applyFilters)*

collect(toList())

Overview of Java 8 Concurrency & Parallelism Frameworks

- Java 8 adds two new concurrency & 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)

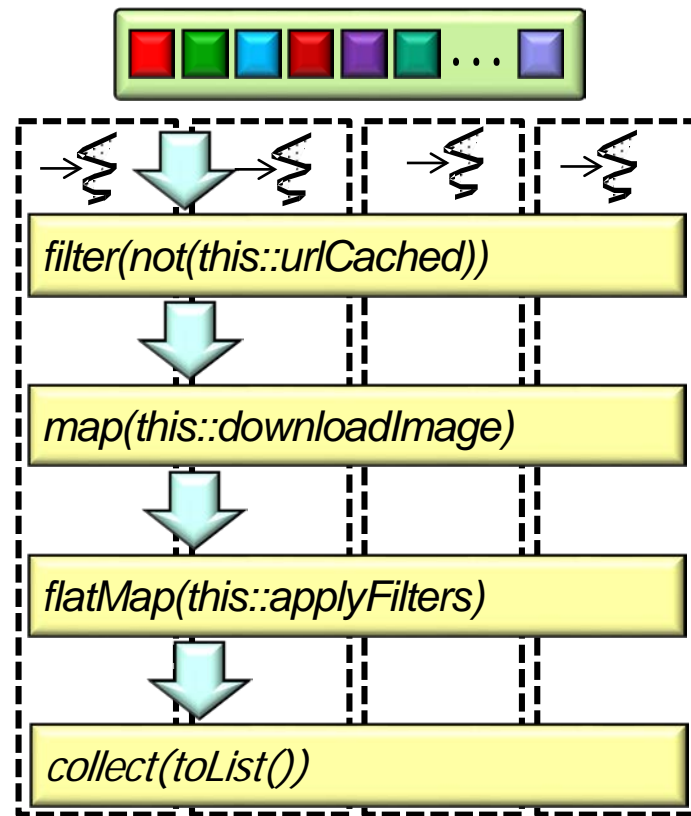flatMap(this::applyFilters)

collect(toList())

Overview of Java 8 Concurrency & Parallelism Frameworks

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

1. Parallel streams

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

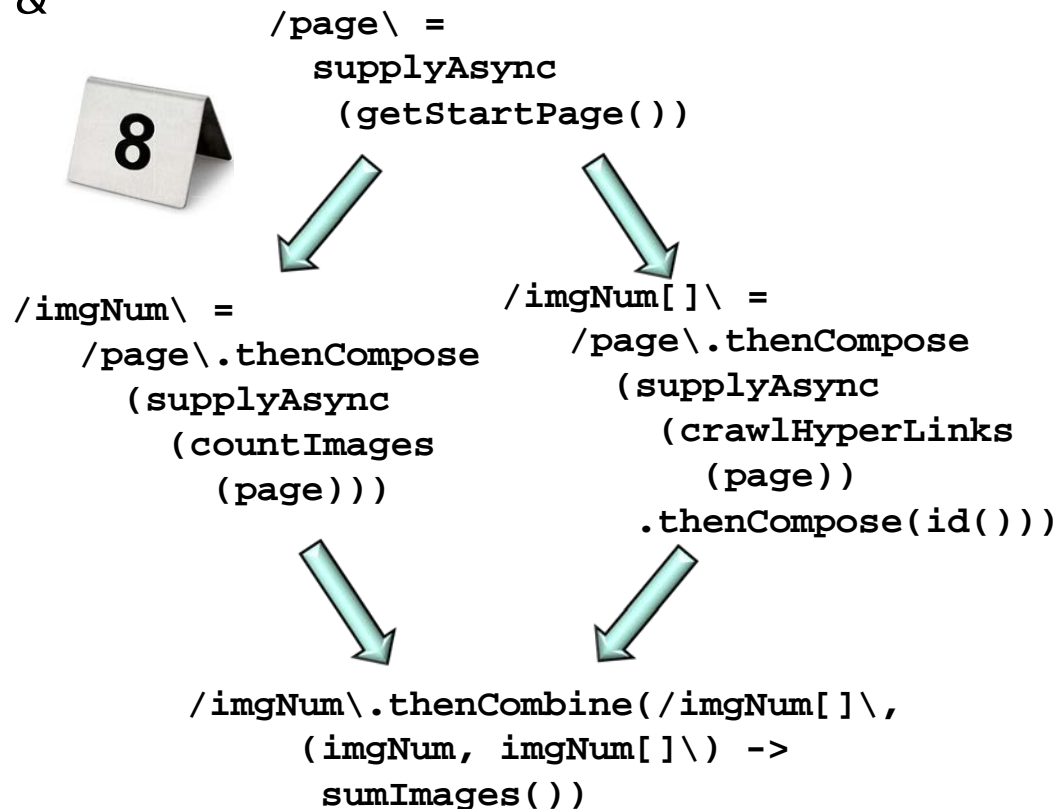 - Chunks of data in the substreams can be mapped to multiple cores (& cores)

```
filter(not(this::urlCached))

map(this::downloadImage)

flatMap(this::applyFilters)

collect(toList())
```

Parallel streams provides a fine-grained data parallelism programming model

Overview of Java 8 Concurrency & Parallelism Frameworks

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

  1. **Parallel streams**

  2. **Completable futures**

```
/page\ =
    supplyAsync
        (getStartPage())
```

```
/imgNum\ =
    /page\.thenCompose
        (supplyAsync
            (countImages
                (page)))
```

```
/imgNum[]\ =
    /page\.thenCompose
        (supplyAsync
            (crawlHyperLinks
                (page))
        .thenCompose(id()))
```

```
/imgNum\.thenCombine(/imgNum[]\,
    (imgNum, imgNum[]\) ->
        sumImages())
```

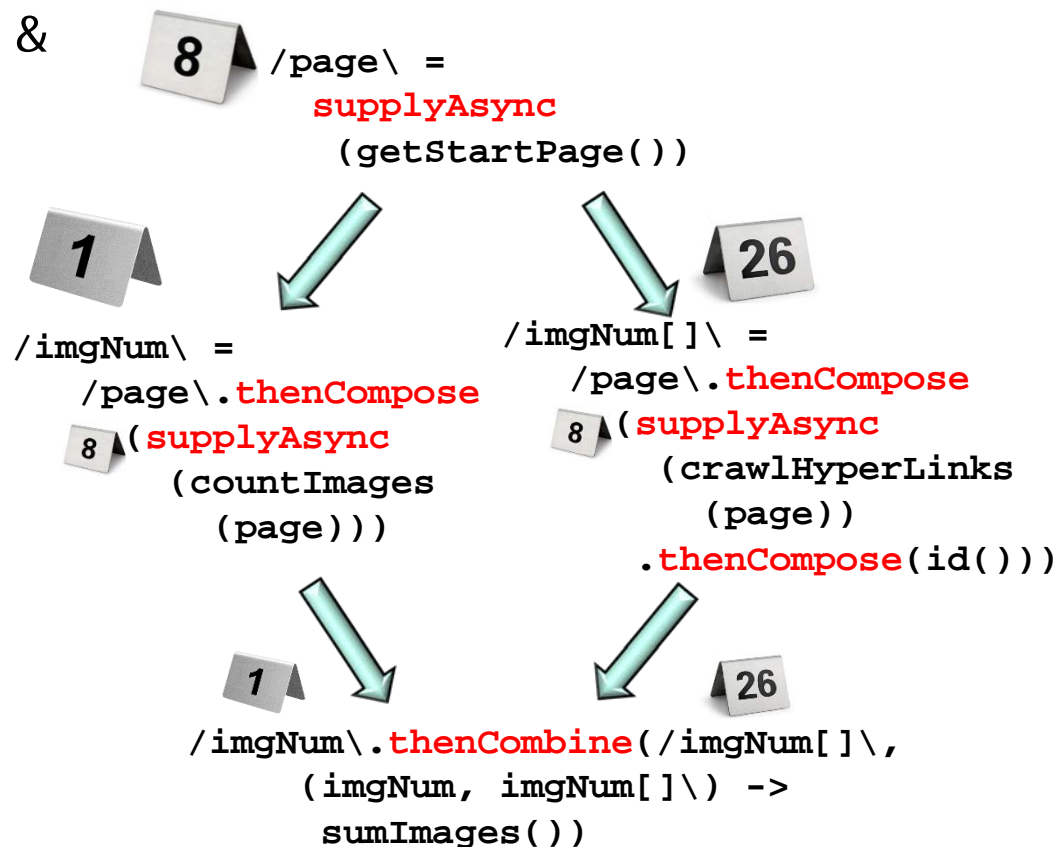See docs.oracle.com/javase/8/docs/api/java/util/concurrent/CompletableFuture.html
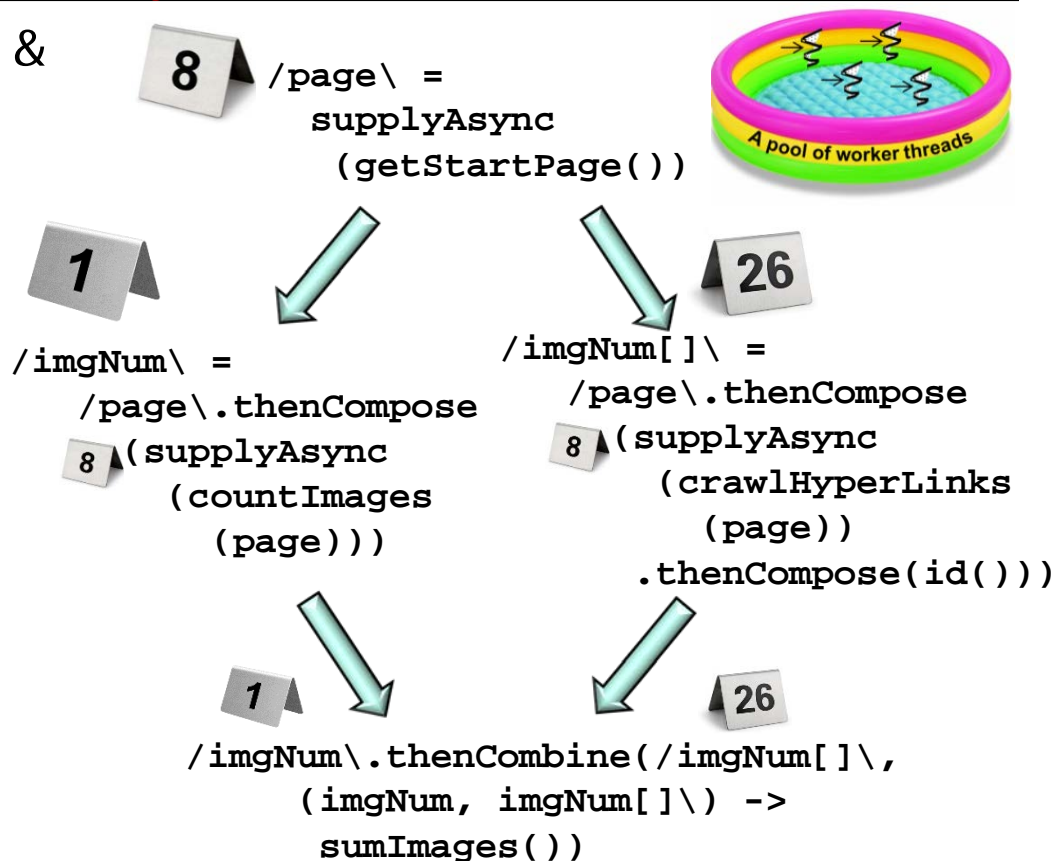
Overview of Java 8 Concurrency & Parallelism Frameworks

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

  1. **Parallel streams**

- **2. Completable futures**

  - Supports dependent actions that trigger upon completion of async operations

```
8  /page\ =
     supplyAsync
       (getStartPage())

1
/imgNum\ =
   /page\.thenCompose
8 (supplyAsync
     (countImages
       (page)))

26
/imgNum[]\ =
   /page\.thenCompose
8 (supplyAsync
     (crawlHyperLinks
       (page))
   .thenCompose(id()))

1                    26
/imgNum\.thenCombine(/imgNum[]\,
     (imgNum, imgNum[]\) ->
       sumImages())
```

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

- Java 8 adds two new concurrency & 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 concurrently in thread pools

```
8  /page\ =
      supplyAsync
        (getStartPage())
```

A pool of worker threads

```
1
/imgNum\ =
    /page\.thenCompose
  8 (supplyAsync
      (countImages
        (page)))
```

```
26
/imgNum[]\ =
    /page\.thenCompose
  8 (supplyAsync
      (crawlHyperLinks
        (page))
      .thenCompose(id()))
```

```
1  26
/imgNum\.thenCombine(/imgNum[]\,
    (imgNum, imgNum[]\) ->
      sumImages())
```
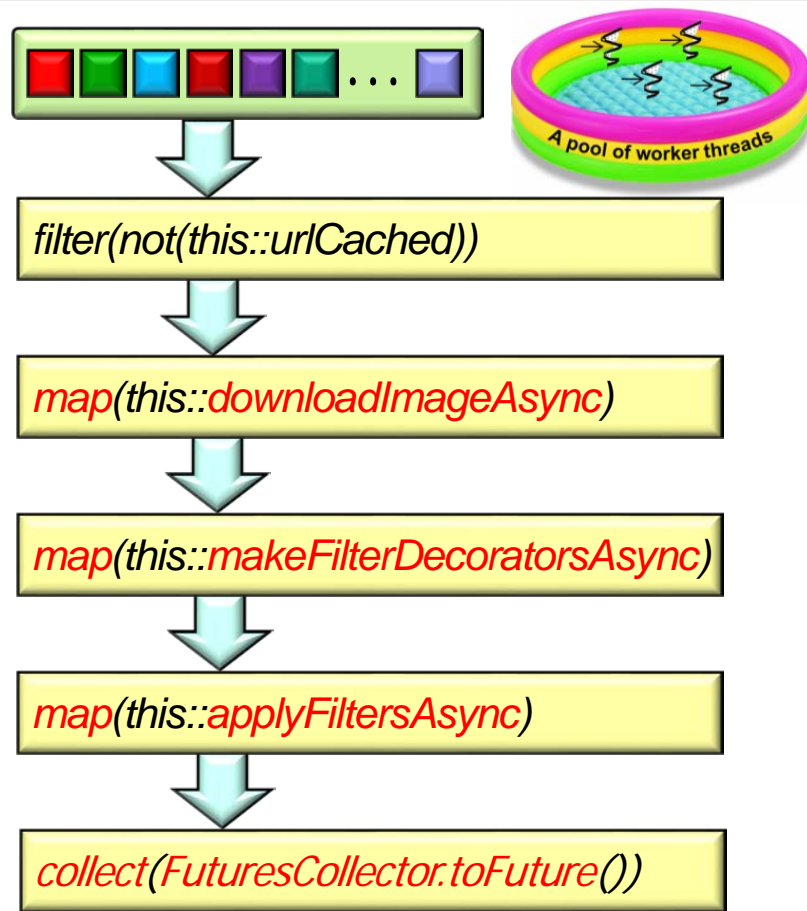
Overview of Java 8 Concurrency & Parallelism Frameworks

- Java 8 adds two new concurrency & 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 concurrently in thread pools

filter(not(this::urlCached))

map(this::downloadImageAsync)

map(this::makeFilterDecoratorsAsync)

map(this::applyFiltersAsync)

collect(FuturesCollector.toFuture())

Java 8 completable futures & streams can be combined to good effects!!

Overview of Java 8 Concurrency & Parallelism Frameworks

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

Alleviates many accidental & inherent complexities of concurrency/parallelism

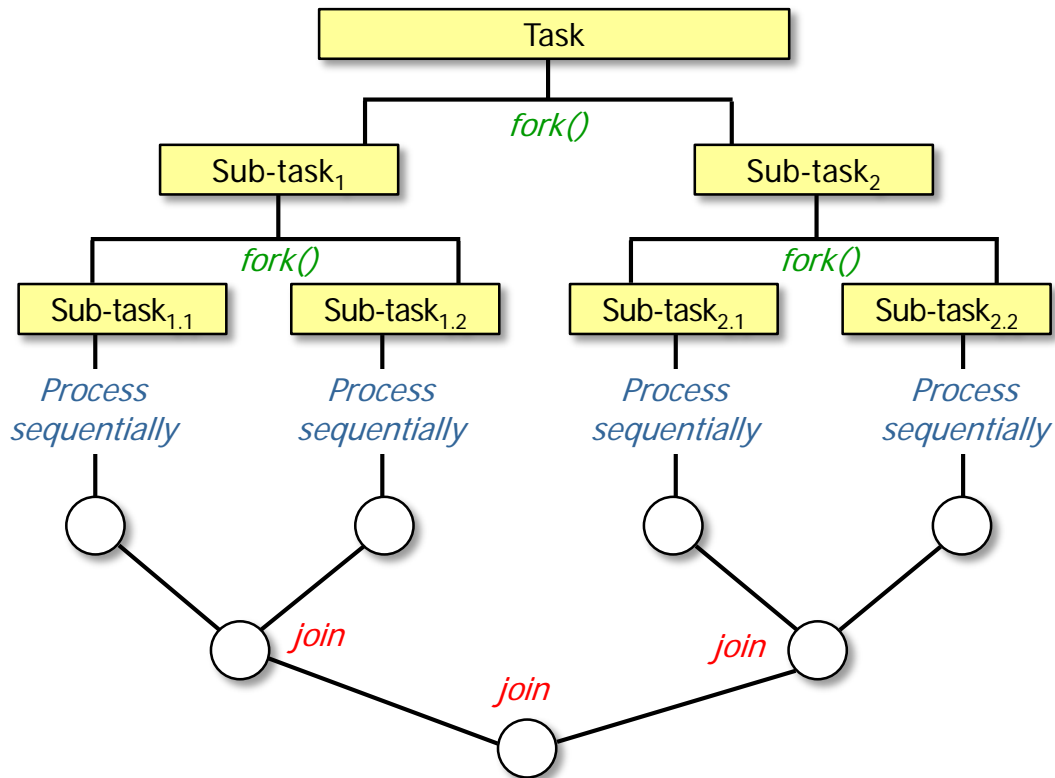# Overview of Java 8 Concurrency & Parallelism Frameworks
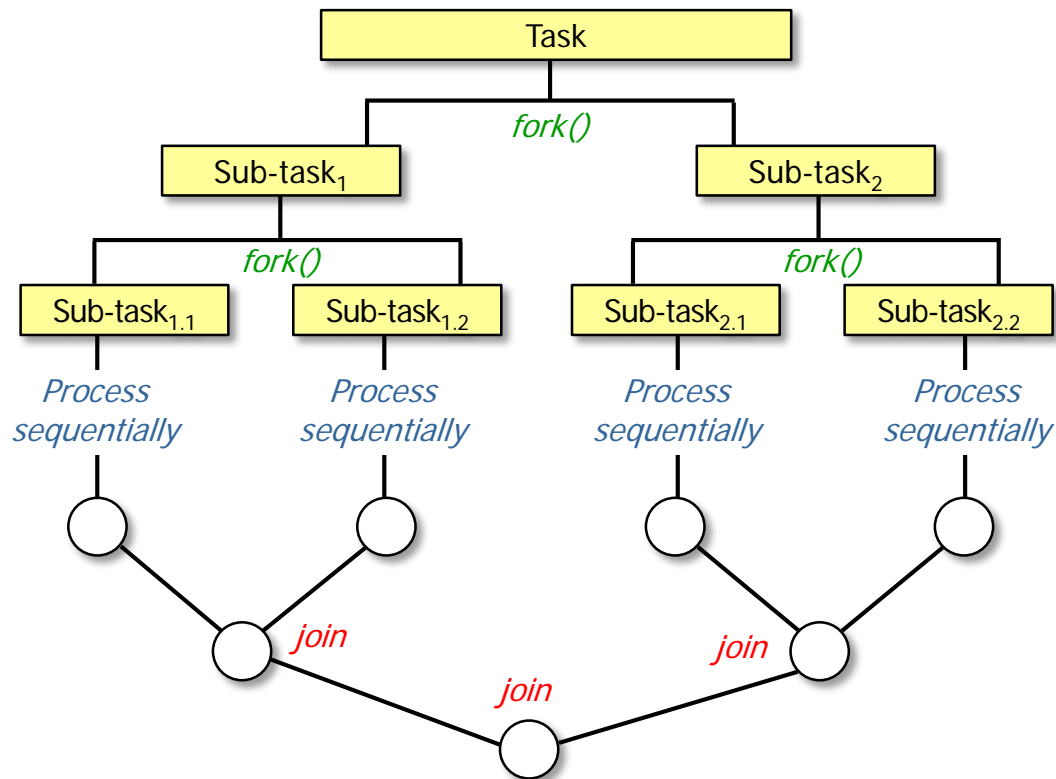
- Both frameworks use the fork-join pool framework by default

Overview of Java 8 Concurrency & Parallelism Frameworks

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

  - Employs *work-stealing* to accelerate performance on multi-core processors

See gee.cs.oswego.edu/dl/cpjslides/fj.pdf
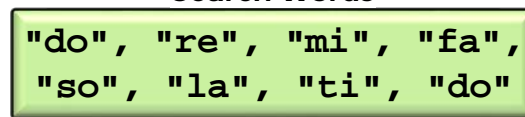
Summary of Example Case Study Apps

Summary of Example Case Study Apps

- SearchStreamGang case study enhances the SearchTaskGang from LiveLessons lesson 2.3

**Input Strings to Search**

Search Words

```
"do", "re", "mi", "fa",
"so", "la", "ti", "do"
```

Starting SearchStreamGangTest
PARALLEL_SPLITERATOR executed in 409 msecs
COMPLETABLE_FUTURES_INPUTS executed in 426 msecs
COMPLETABLE_FUTURES_PHASES executed in 427 msecs
PARALLEL_STREAMS executed in 437 msecs
PARALLEL_STREAM_PHASES executed in 440 msecs
RXJAVA_PHASES executed in 485 msecs
PARALLEL_STREAM_INPUTS executed in 802 msecs
RXJAVA_INPUTS executed in 866 msecs
SEQUENTIAL_LOOPS executed in 1638 msecs
SEQUENTIAL_STREAM executed in 1958 msecs
Ending SearchStreamGangTest

See github.com/douglascraigschmidt/LiveLessons/tree/master/SearchStreamGang

Summary of Example Case Study Apps

- ImageStreamGang case study enhances the ImageTaskGang from Live Lessons lesson 4

List of Filters to Apply

List of URLs to Download

Persistent Data Store

Socket

Socket
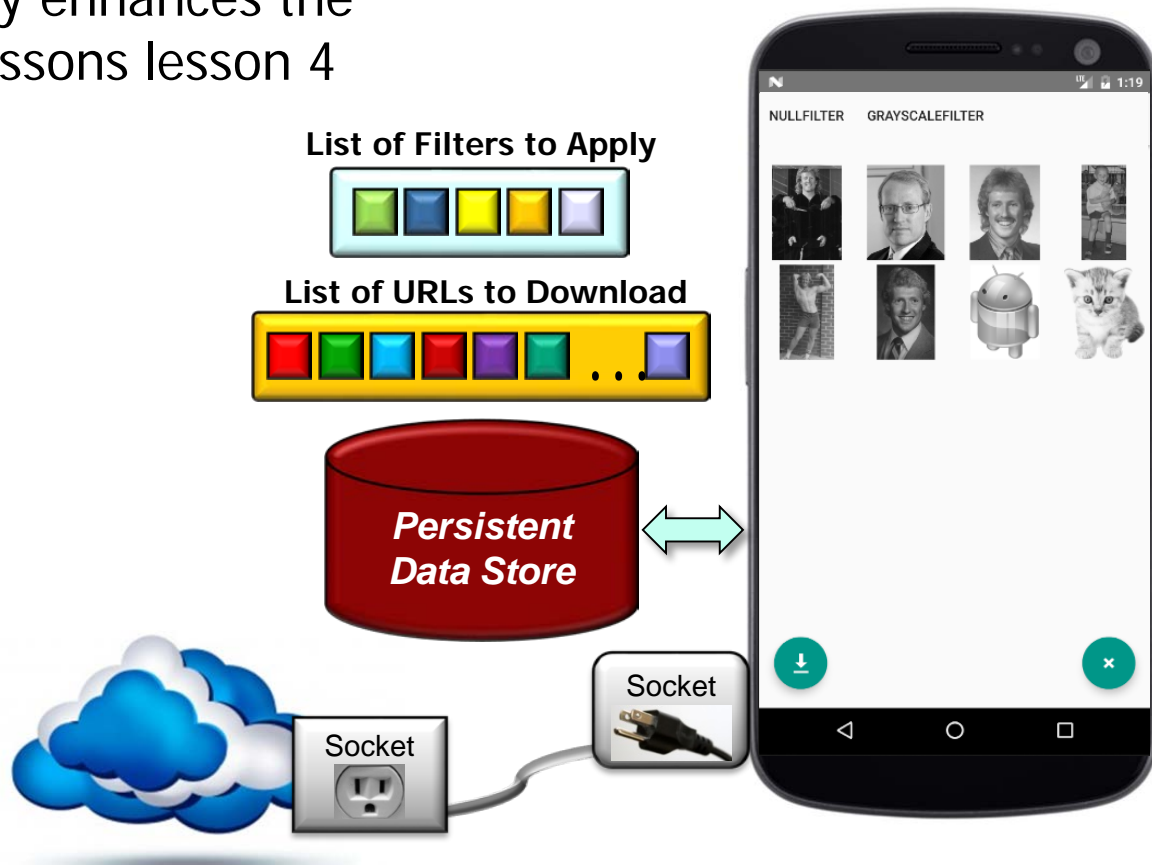
See github.com/douglascraigschmidt/LiveLessons/tree/master/ImageStreamGang

End of Java 8 Concurrency & Parallelism Frameworks Intro