Overview of the Java CompletableFuture ImageStreamGang Case Study

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

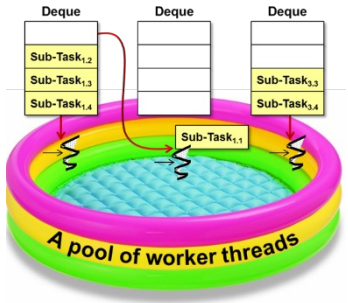
Professor of Computer Science

Institute for Software Integrated Systems

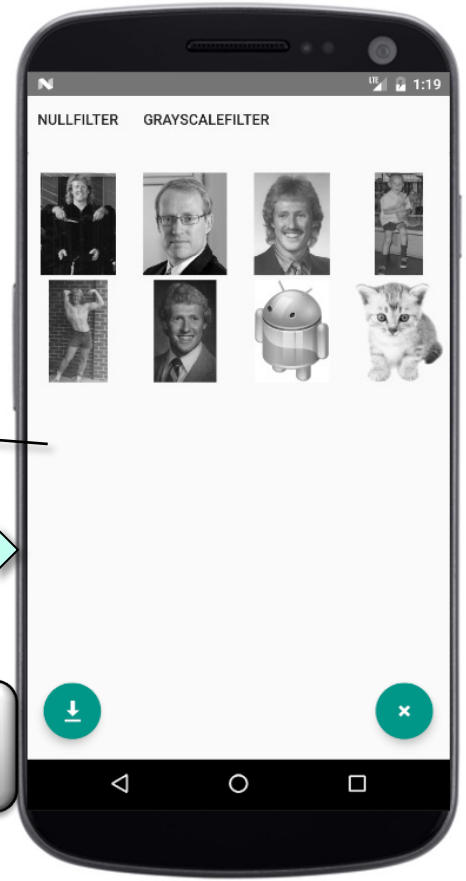
Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

- Understand the design of the Java completable future version of the ImageStreamGang app

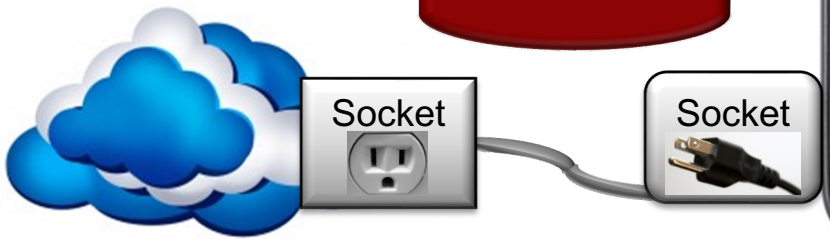


Process a list of URLs to images that aren't already cached & download/ transform/store images asynchronously



List of URLs to Download

List of Filters to Apply

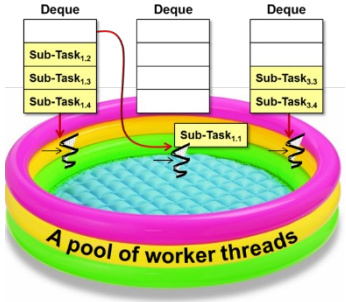


See github.com/douglascraigschmidt/LiveLessons/tree/master/ImageStreamGang

Overview of the Completable Futures ImageStreamGang

Overview of Completable Futures ImageStreamGang

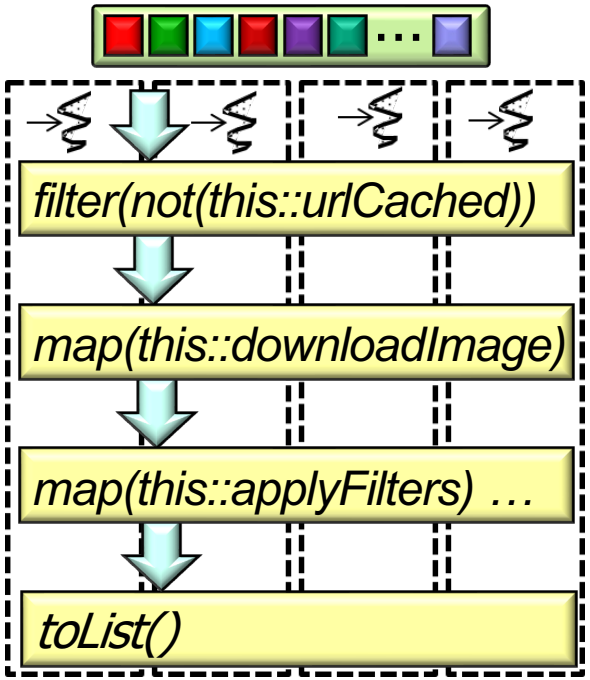
- This app applies several Java parallelism frameworks that do the following
  - Ignore cached images
  - Download non-cached images
  - Apply a list of filters to each image
  - Store filtered images in the file system
  - Display images to the user



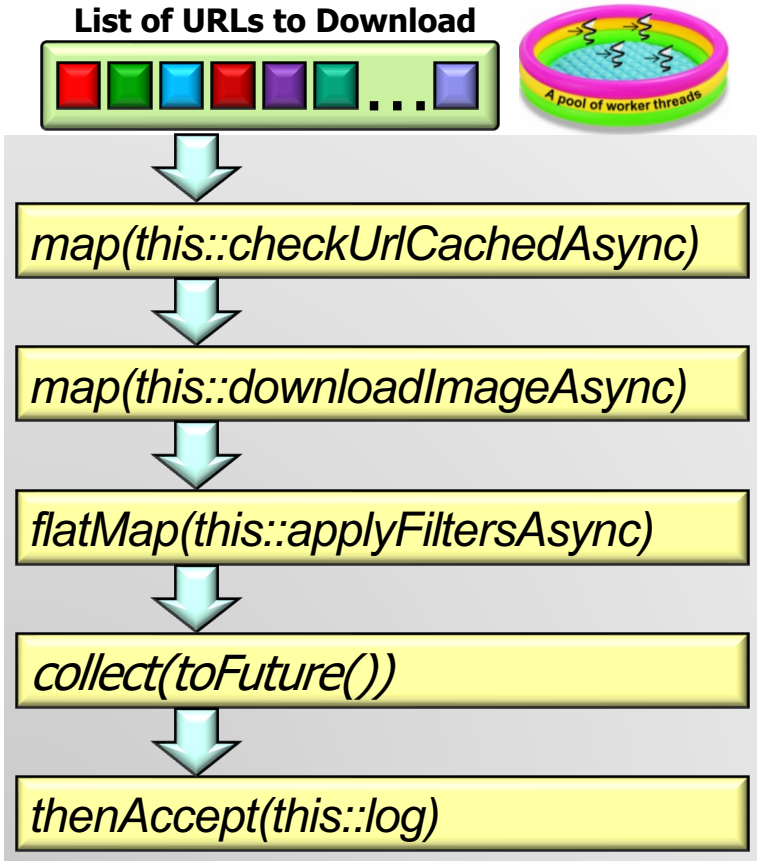
See github.com/douglascraigschmidt/LiveLessons/tree/master/ImageStreamGang

Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant



Parallel Streams

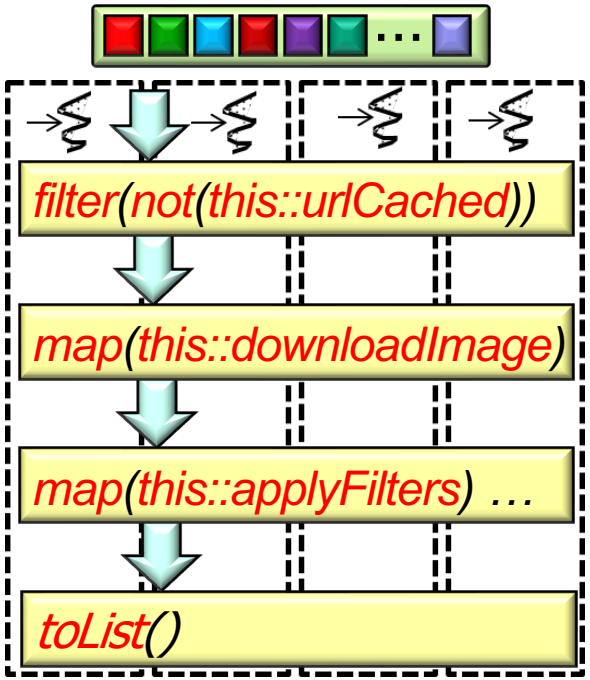


Completable Futures

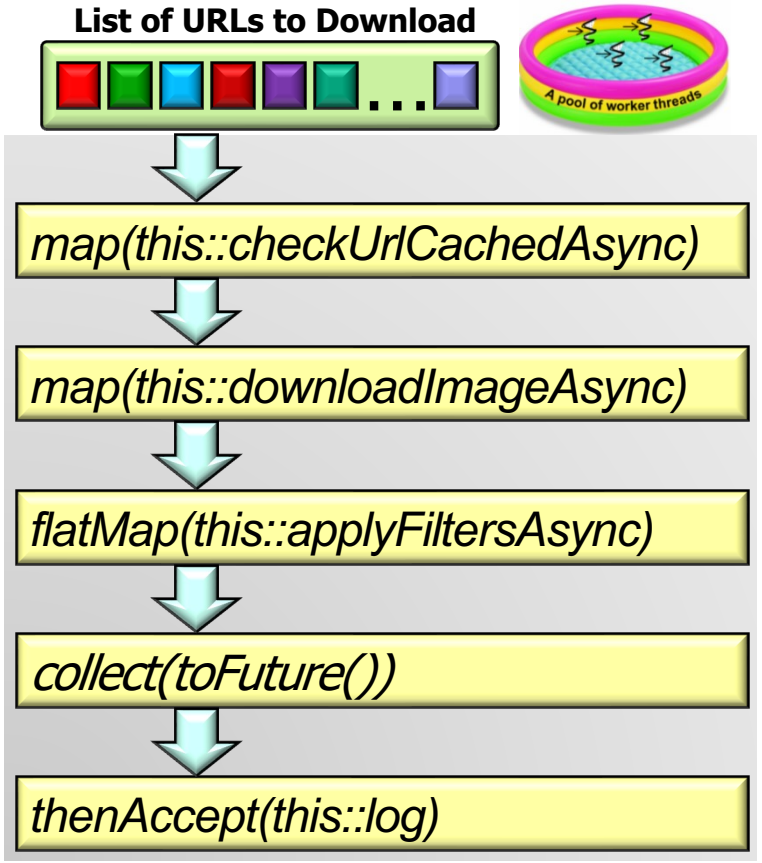
See earlier lesson on "*The Java Parallel ImageStreamGang Example*"

Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant



Parallel Streams

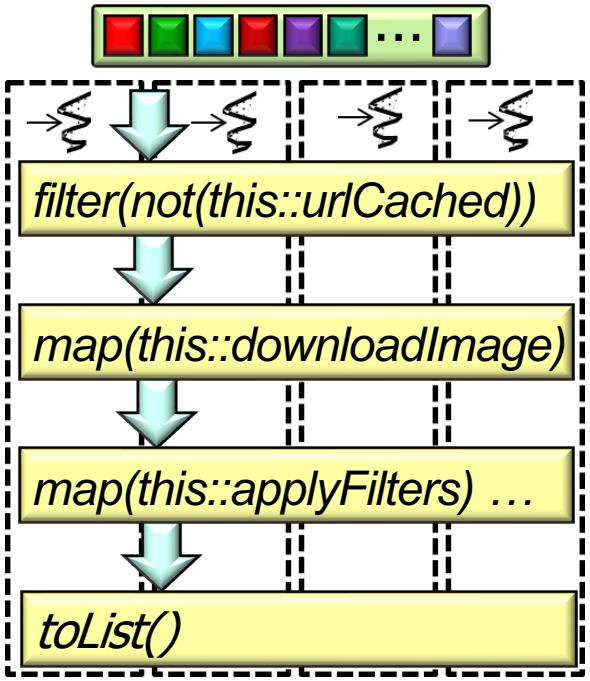


Completable Futures

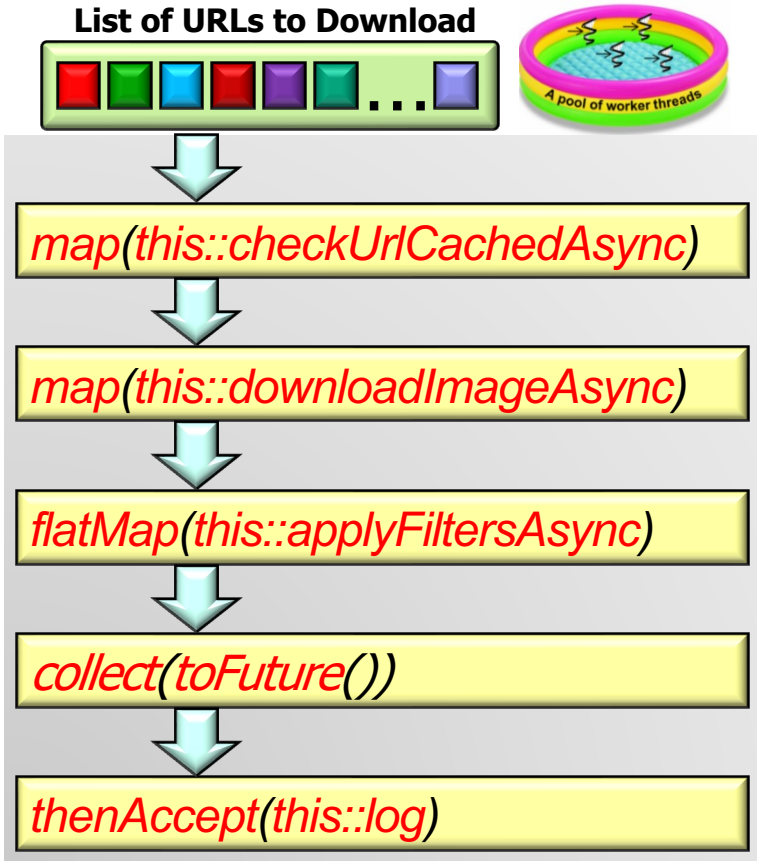
All behaviors in the parallel stream variant are synchronous

Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant



Parallel Streams

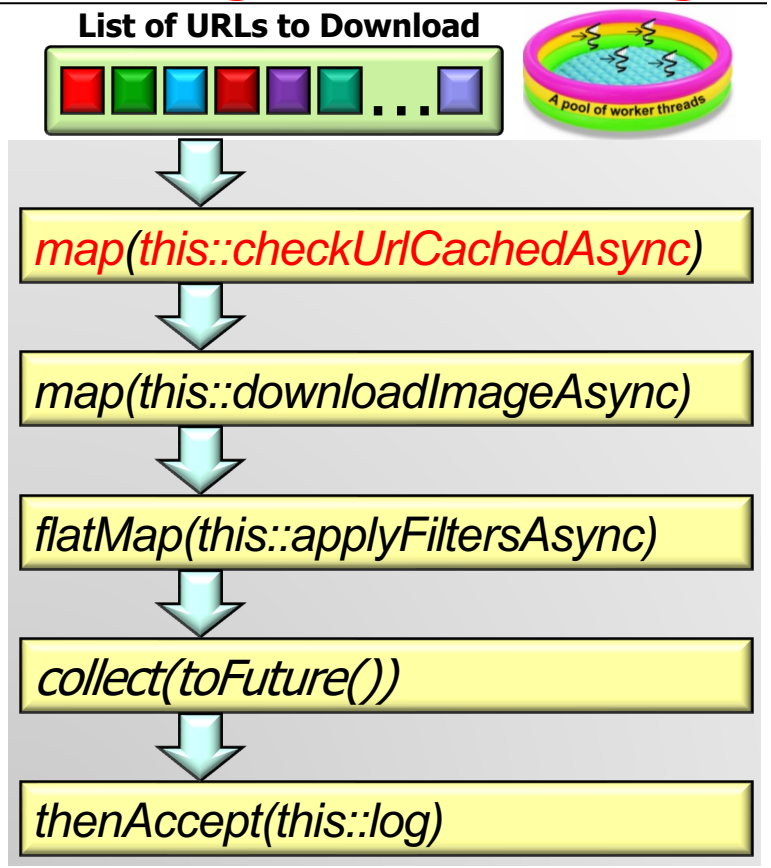


Completable Futures

All behaviors in the completable futures variant are asynchronous

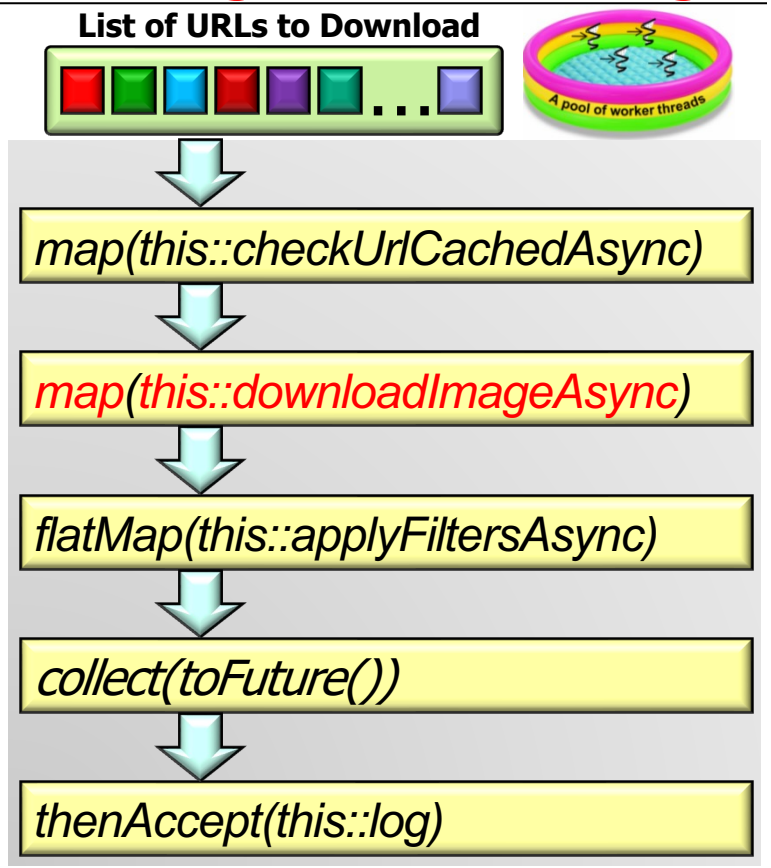
Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant, e.g.
  - Ignore cached images *asynchronously*



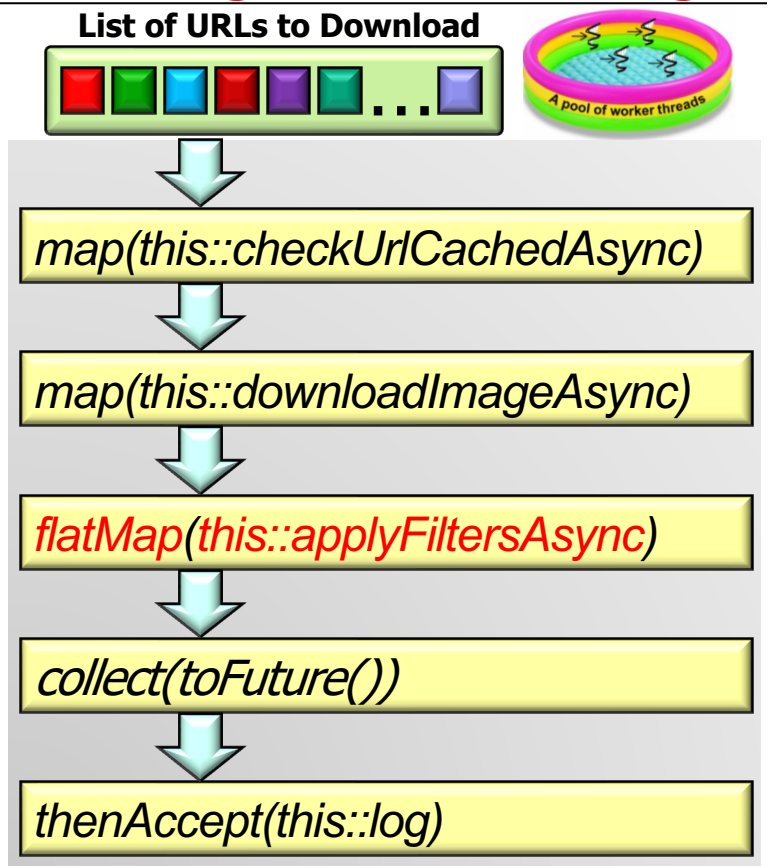
Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant, e.g.
  - Ignore cached images *asynchronously*
  - Download non-cached images *asynchronously*



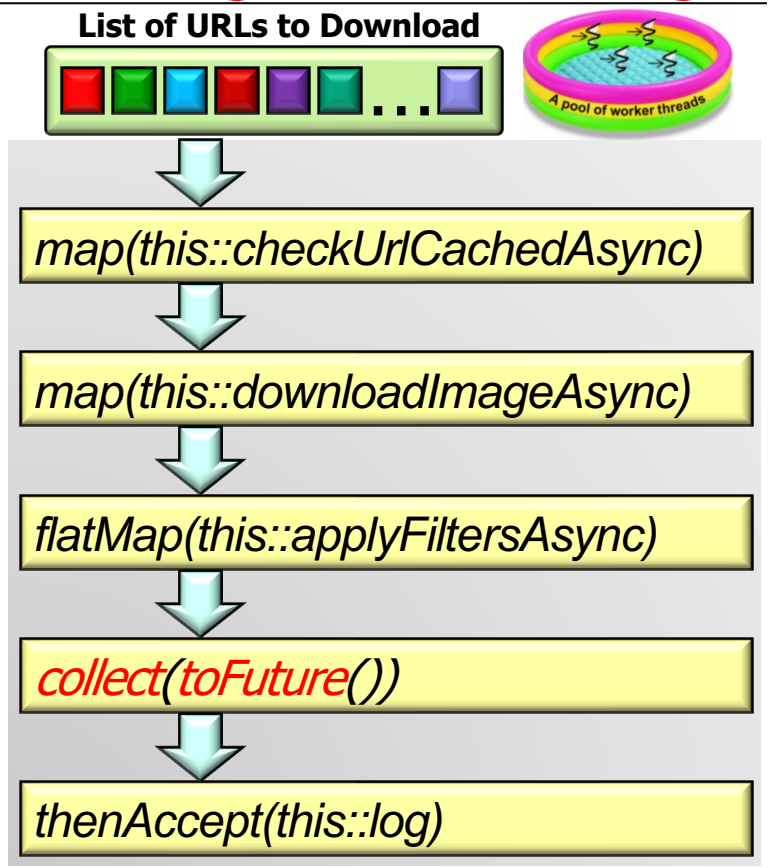
Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant, e.g.
  - Ignore cached images *asynchronously*
  - Download non-cached images *asynchronously*
  - As downloads complete apply a list of filters & store filtered images in file system *asynchronously*



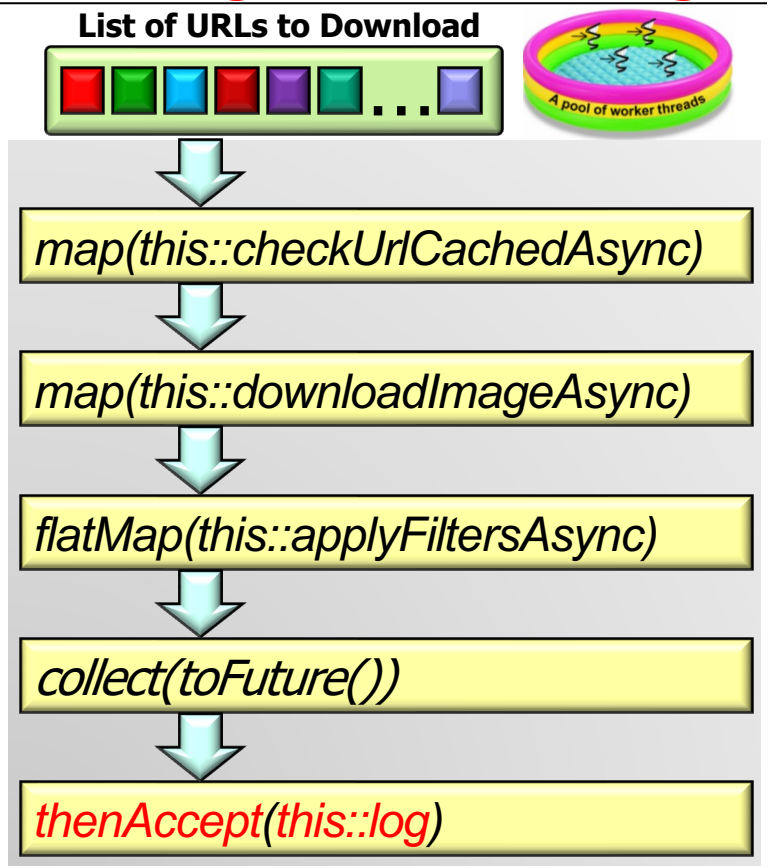
Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant, e.g.
  - Ignore cached images *asynchronously*
  - Download non-cached images *asynchronously*
  - As downloads complete apply a list of filters & store filtered images in file system *asynchronously*
  - Trigger all the stream processing to run *asynchronously*



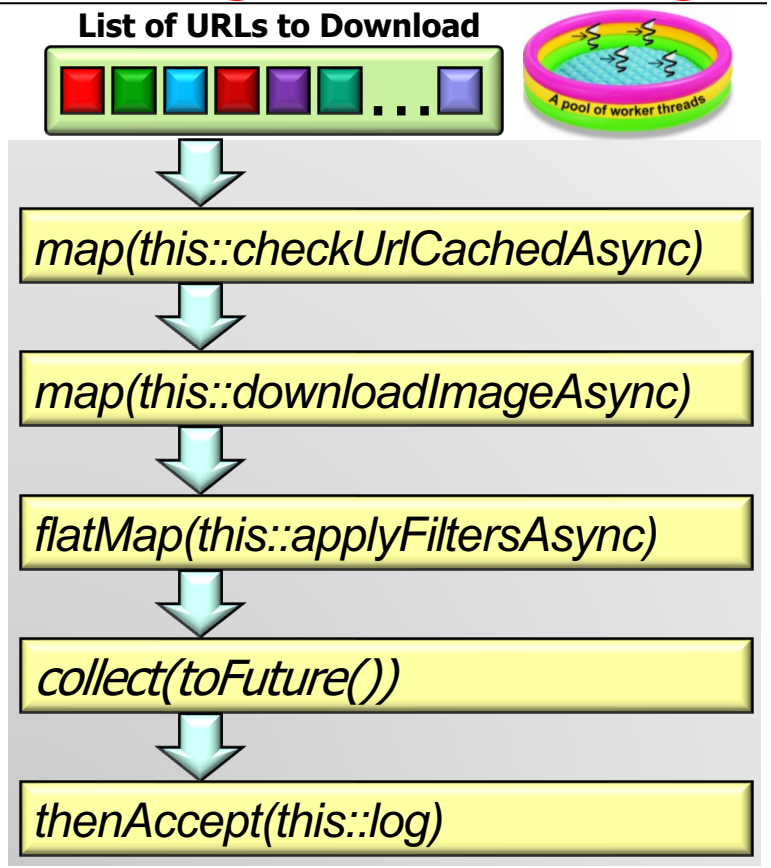
Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant, e.g.
  - Ignore cached images *asynchronously*
  - Download non-cached images *asynchronously*
  - As downloads complete apply a list of filters & store filtered images in file system *asynchronously*
  - Trigger all the stream processing to run *asynchronously*
  - Get results of asynchronous computations



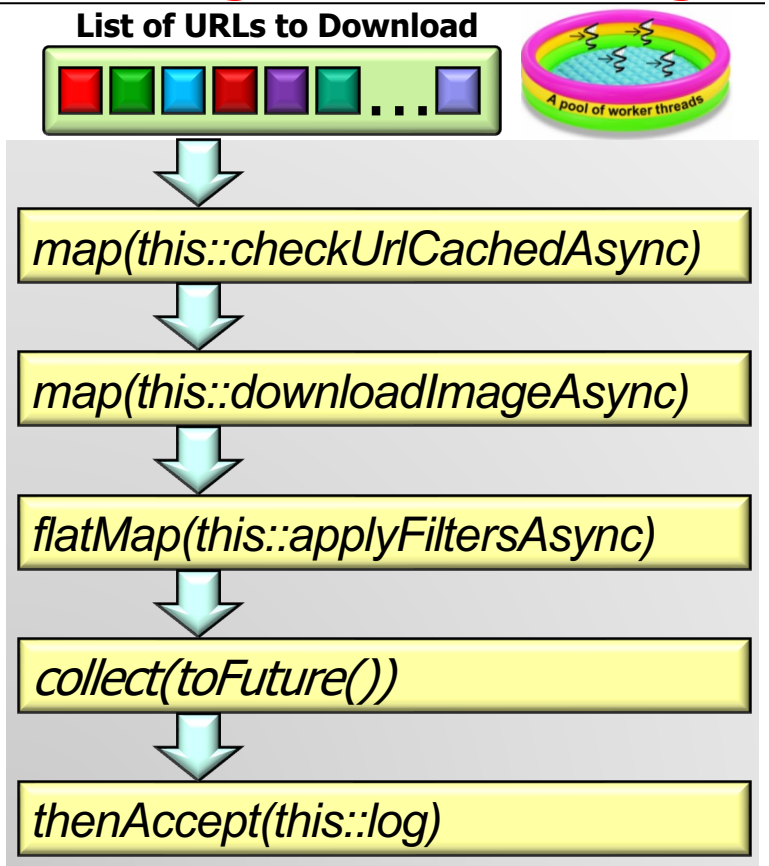
Overview of Completable Futures ImageStreamGang

- The behaviors in this pipeline differ from the earlier parallel streams variant, e.g.
  - Ignore cached images *asynchronously*
  - Download non-cached images *asynchronously*
  - As downloads complete apply a list of filters & store filtered images in file system *asynchronously*
  - Trigger all the stream processing to run *asynchronously*
  - Get results of asynchronous computations
    - Ultimately display images to user



Overview of Completable Futures ImageStreamGang

- Combining completable futures & streams helps to *efficiently* close the gap between the domain intent & the computations



End of Overview of the Java Completable Future Image StreamGang Case Study