

Visualizing the Behaviors of the Java Parallel ImageStreamGang Case Study

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

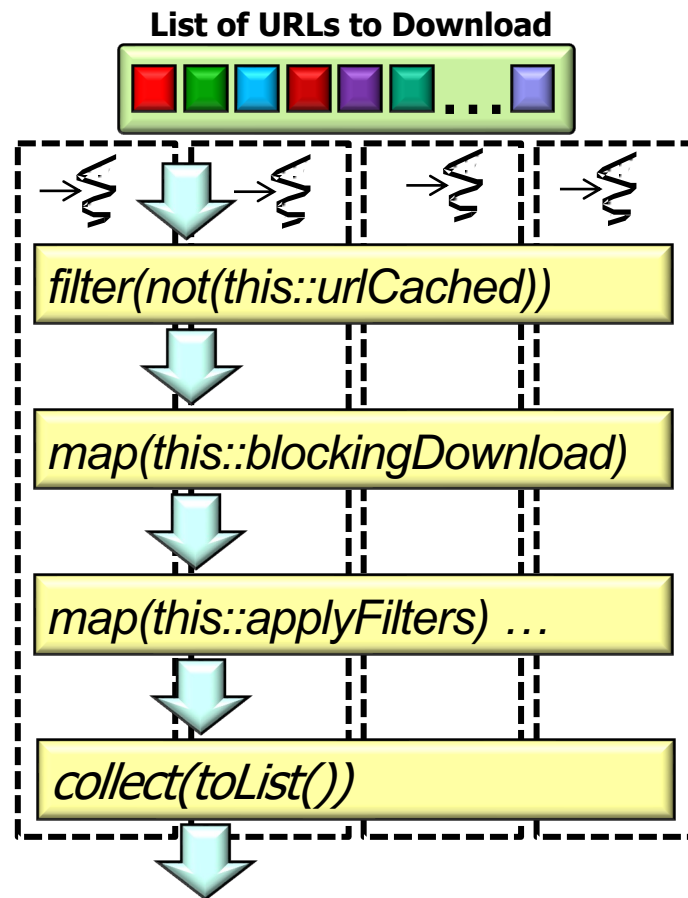
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

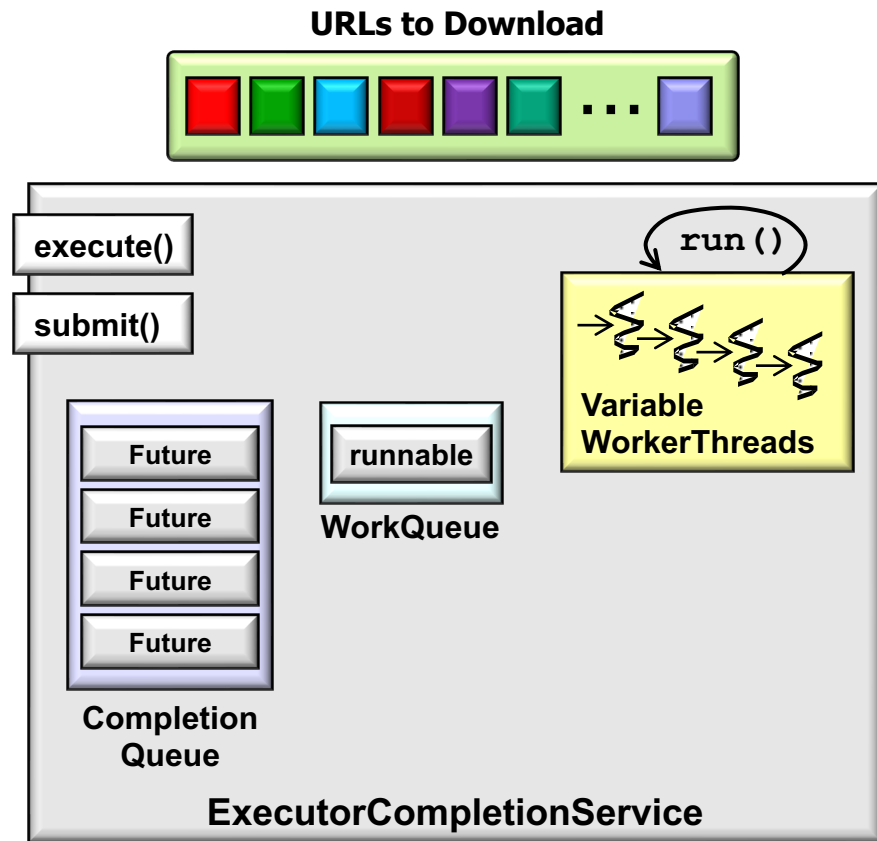
- Understand purpose of the ImageStreamGang app
- Recognize patterns applied in the ImageStreamGang app
- Know how the structure of the ImageStreamGang app
- Visualize how Java parallel streams are applied to the ImageStreamGang app



See github.com/douglasraigschmidt/LiveLessons/blob/master/ImageStreamGang

Learning Objectives in this Part of the Lesson

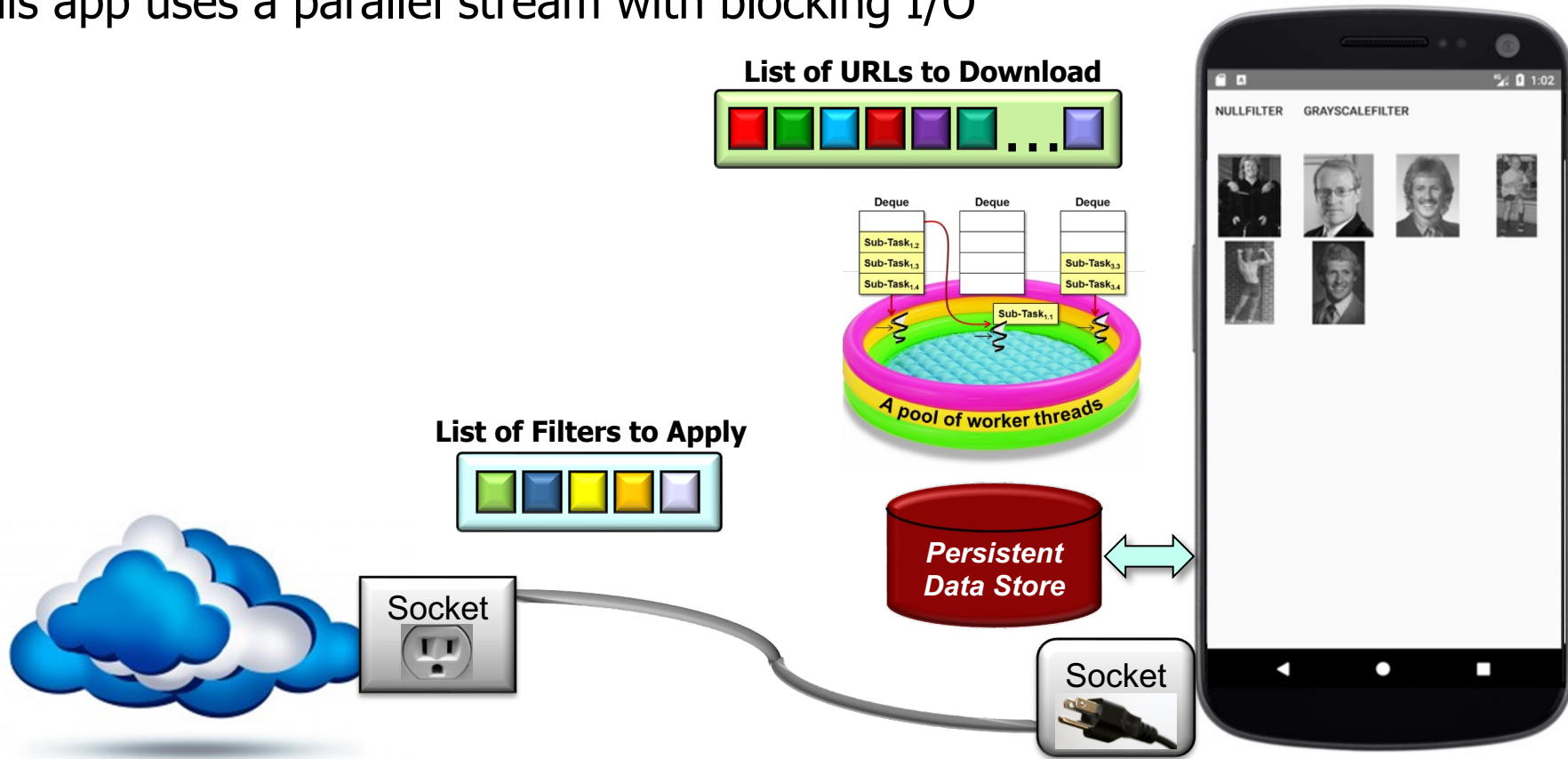
- Understand purpose of the ImageStreamGang app
- Recognize patterns applied in the ImageStreamGang app
- Know how the structure of the ImageStreamGang app
- Visualize how Java parallel streams are applied to the ImageStreamGang app
 - This app enhances the earlier ImageTaskGang app



Visualizing Parallel Streams in ImageStreamGang

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

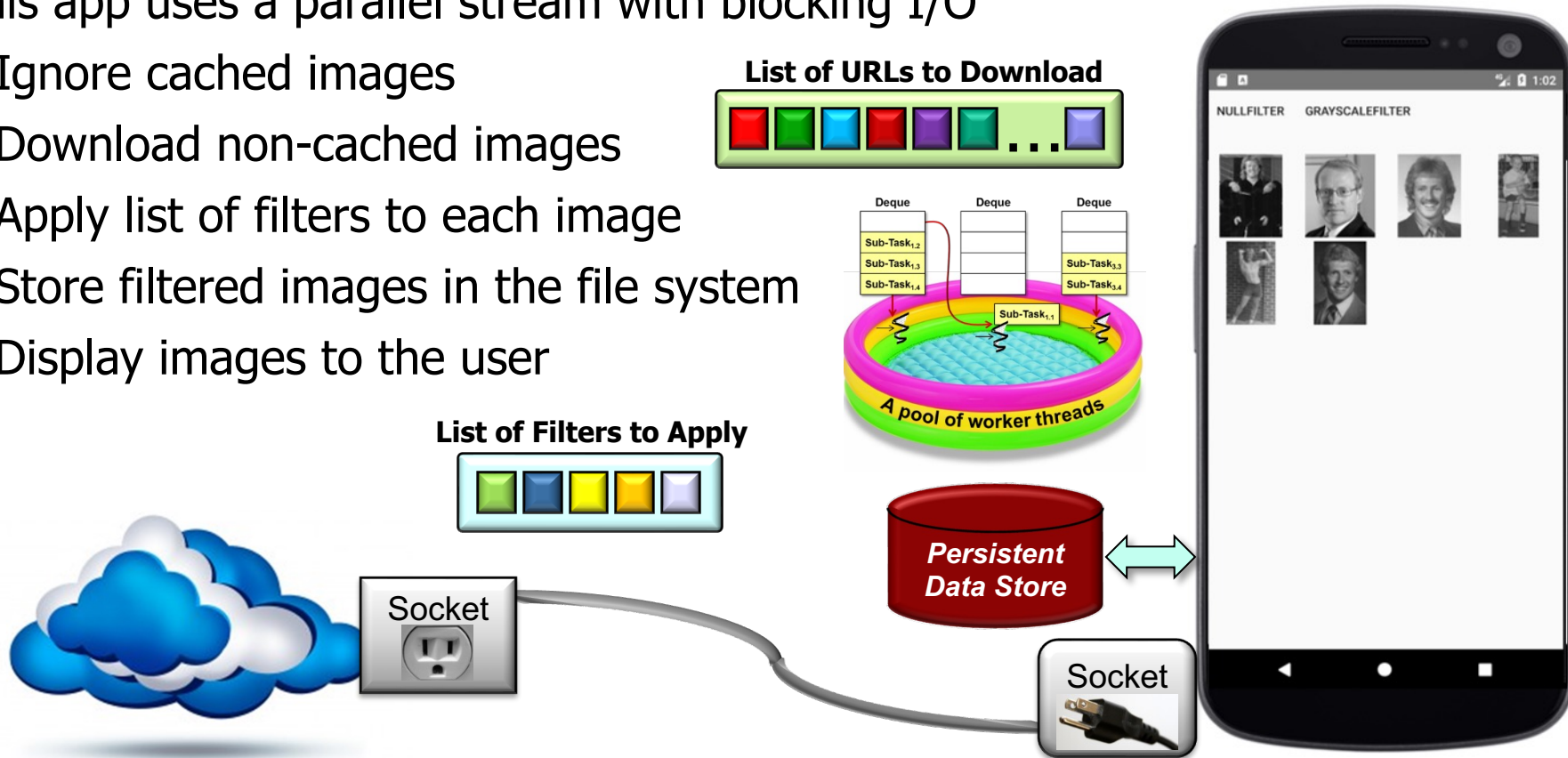


See github.com/douglasraigschmidt/LiveLessons/blob/master/ImageStreamGang/AndroidGUI

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

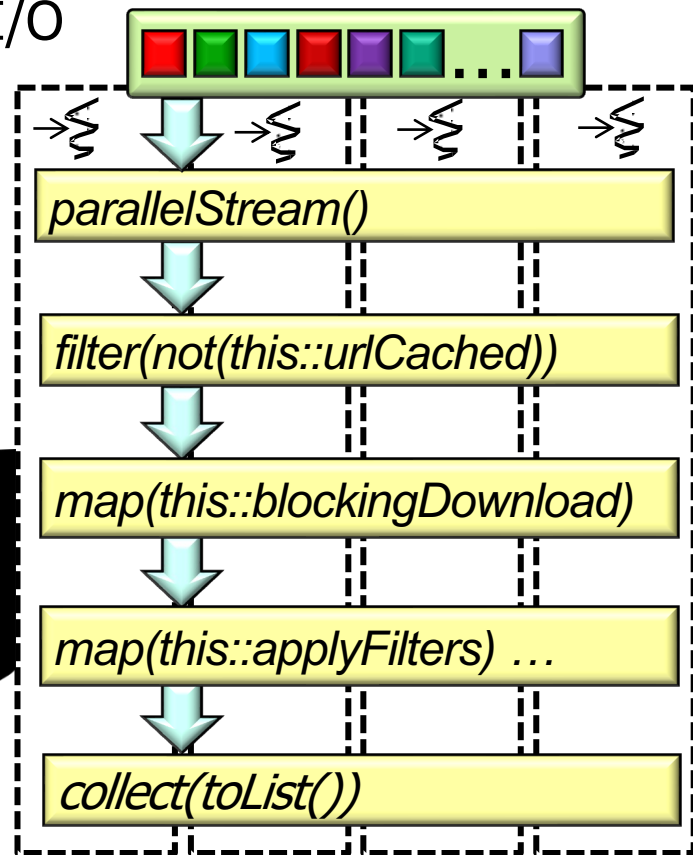
- Ignore cached images
- Download non-cached images
- Apply list of filters to each image
- Store filtered images in the file system
- Display images to the user



Combines Java object-oriented & functional programming features

Visualizing Parallel Streams in ImageStreamGang

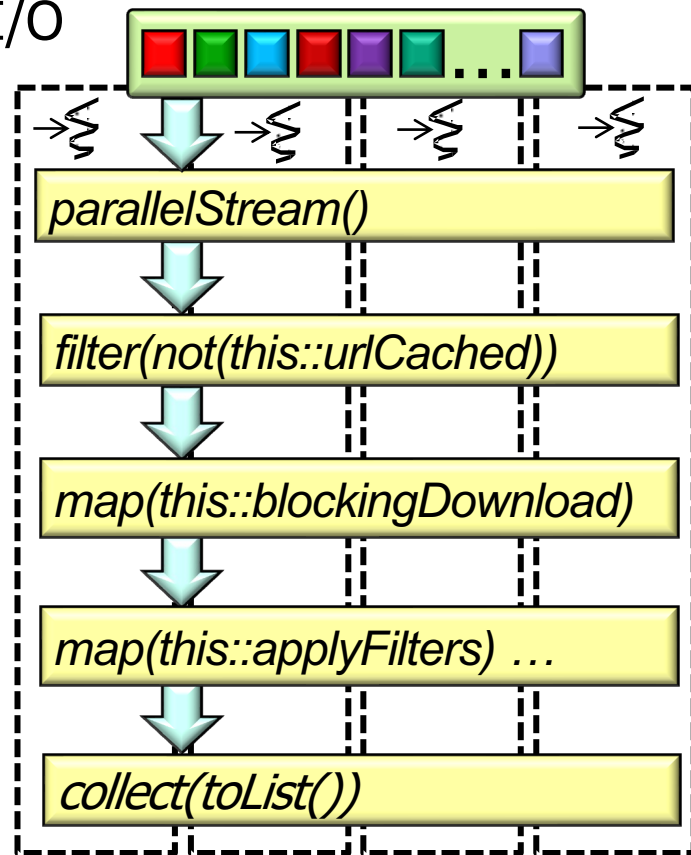
- This app uses a parallel stream with blocking I/O
 - Ignore cached images
 - Download non-cached images
 - Apply list of filters to each image
 - Store filtered images in the file system
 - Display images to the user



Declarative stream pipeline closely aligns with the app description

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O



Closes gap between domain intent & computations that implement the intent

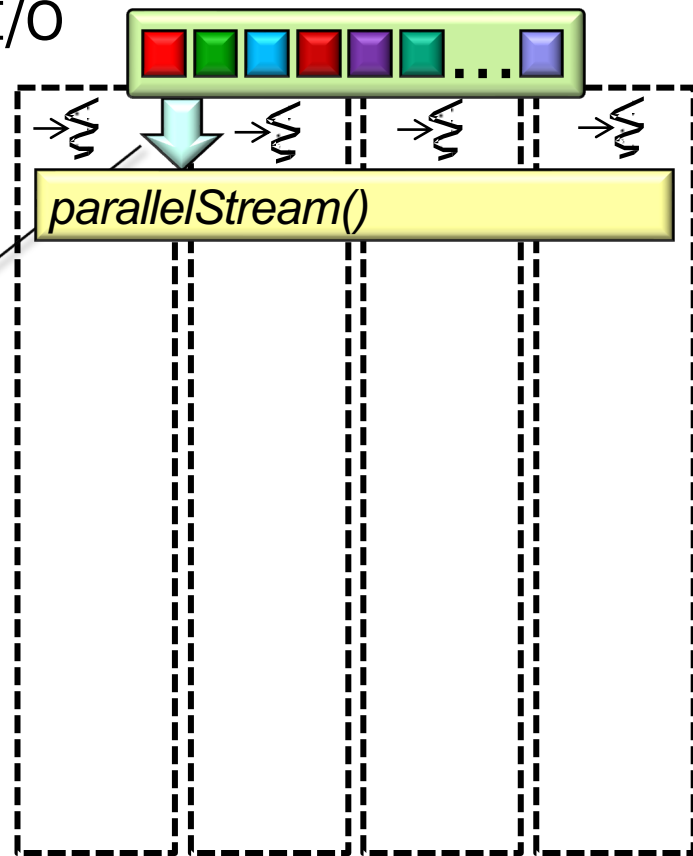
Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

List
<URL>



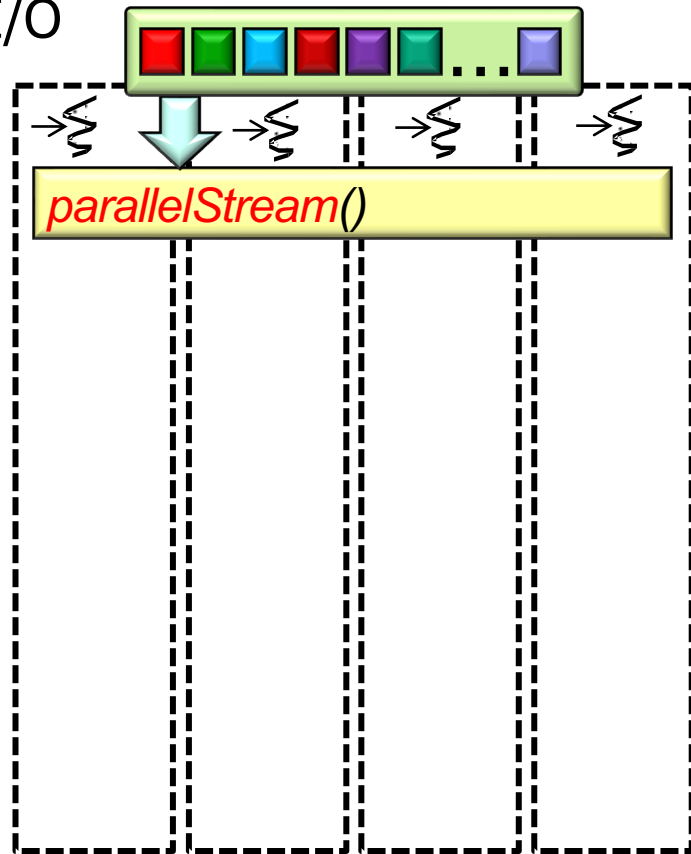
Input a list of image URLs



Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

List
<URL>



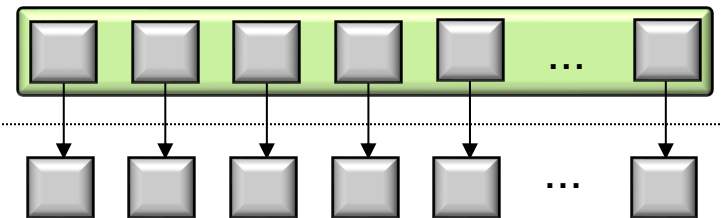
Convert collection to a parallel stream

Visualizing Parallel Streams in ImageStreamGang

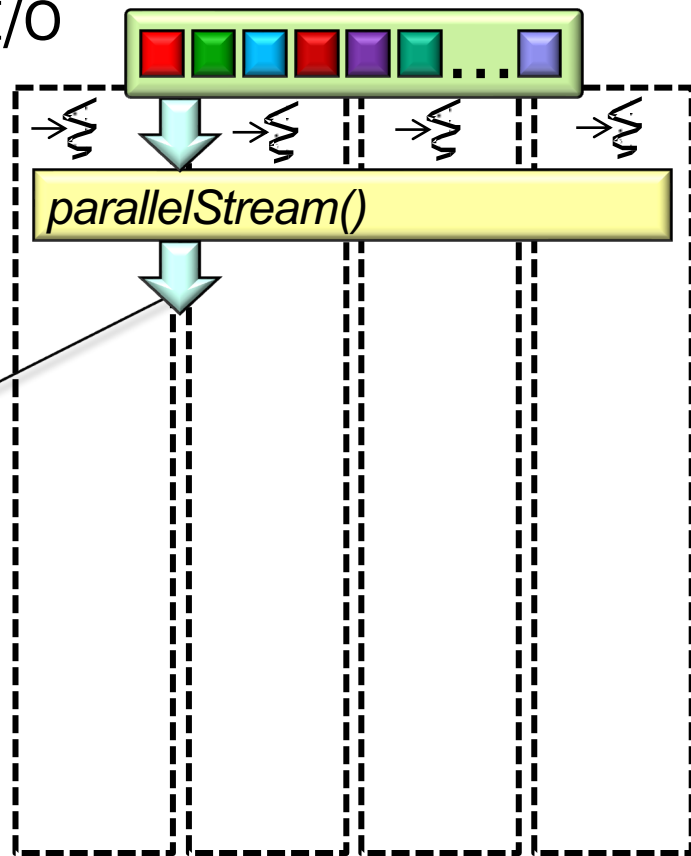
- This app uses a parallel stream with blocking I/O

List
<URL>

Stream
<URL>



Output a stream of image URLs

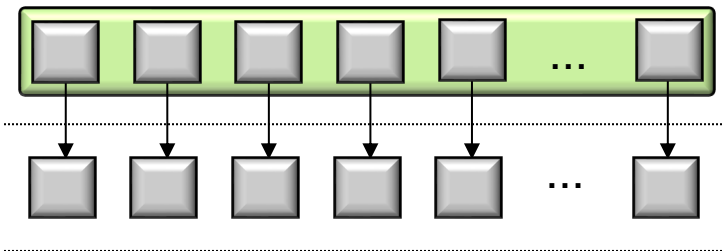


Visualizing Parallel Streams in ImageStreamGang

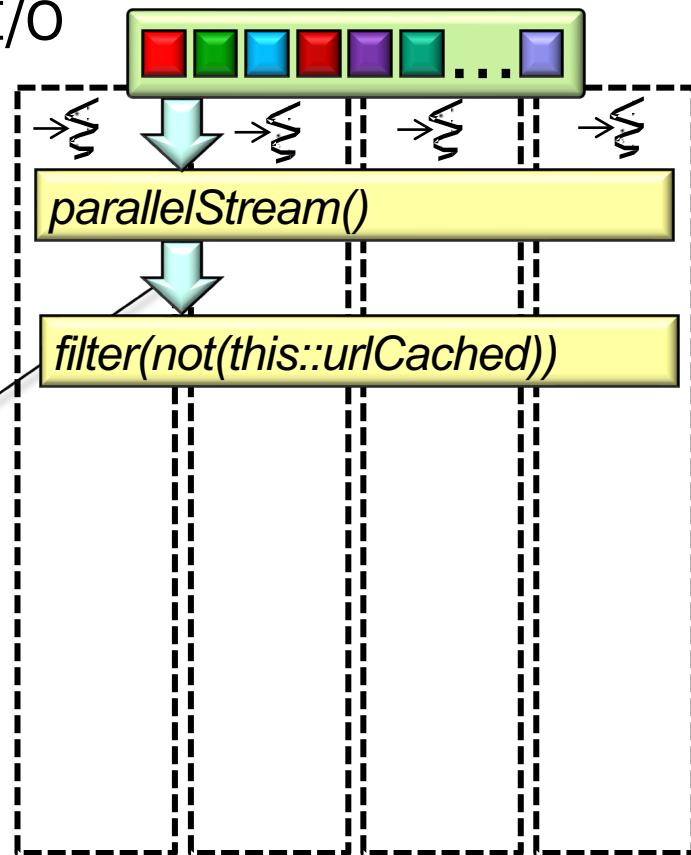
- This app uses a parallel stream with blocking I/O

List
<URL>

Stream
<URL>



Input a stream of image URLs

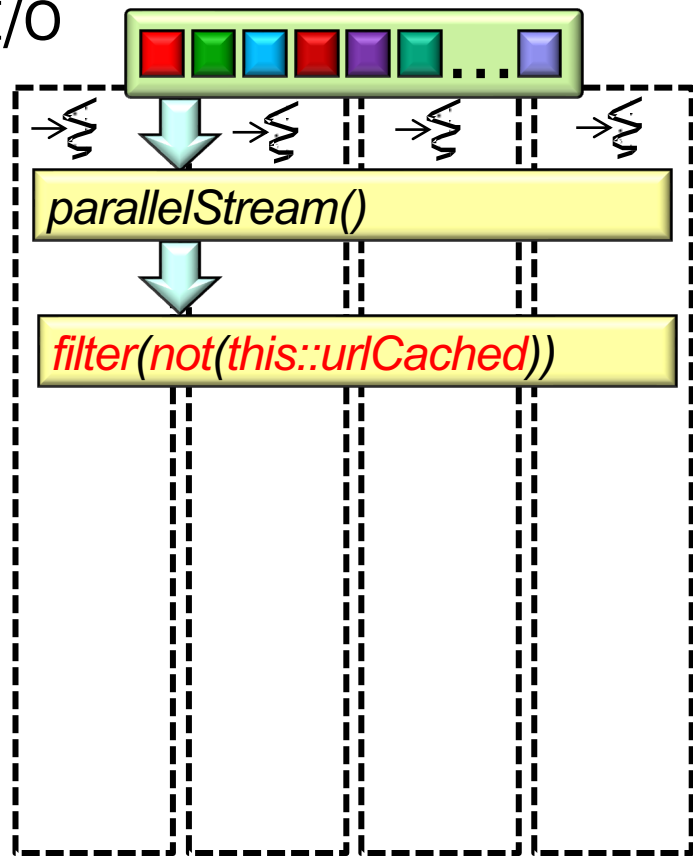
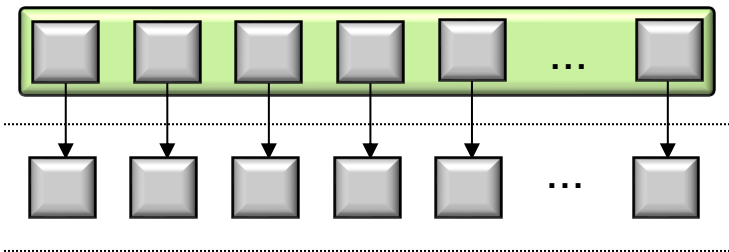


Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

List
<URL>

Stream
<URL>



`filter()` ignores cached images

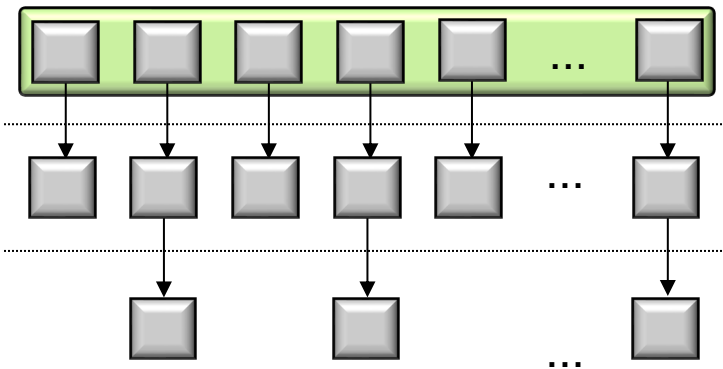
Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

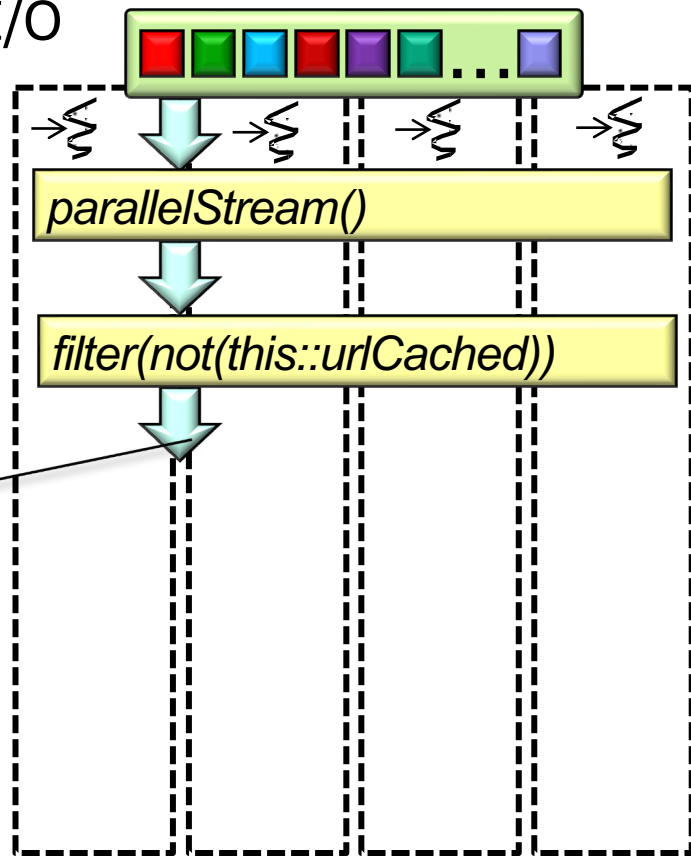
List
<URL>

Stream
<URL>

Stream
<URL>



Output a stream of non-cached image URLs



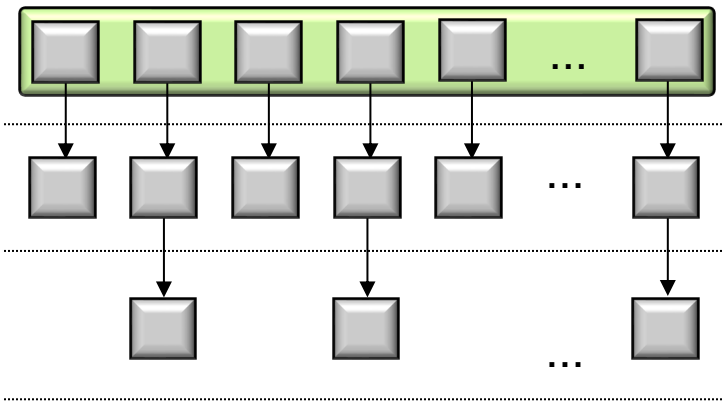
Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

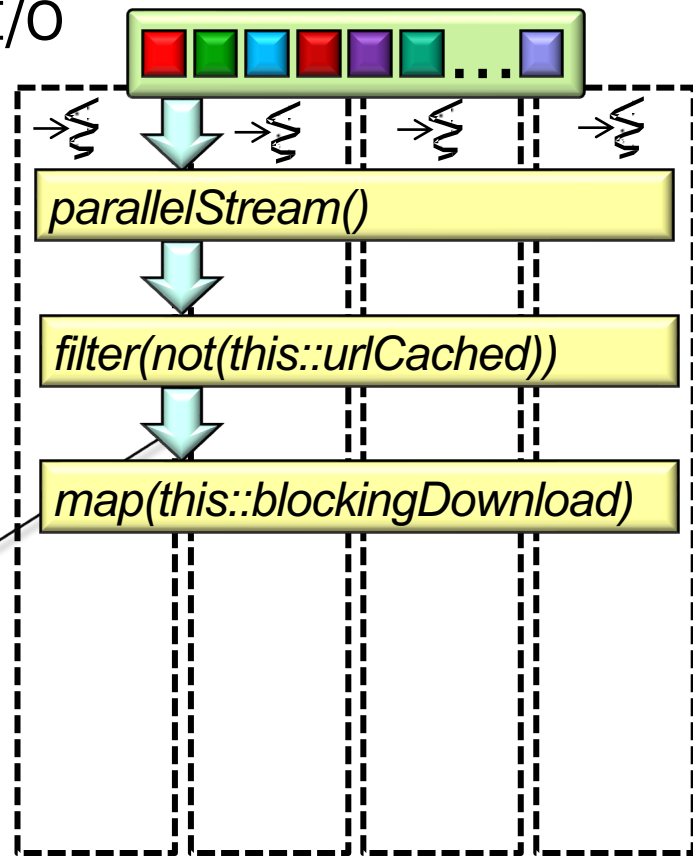
List
<URL>

Stream
<URL>

Stream
<URL>

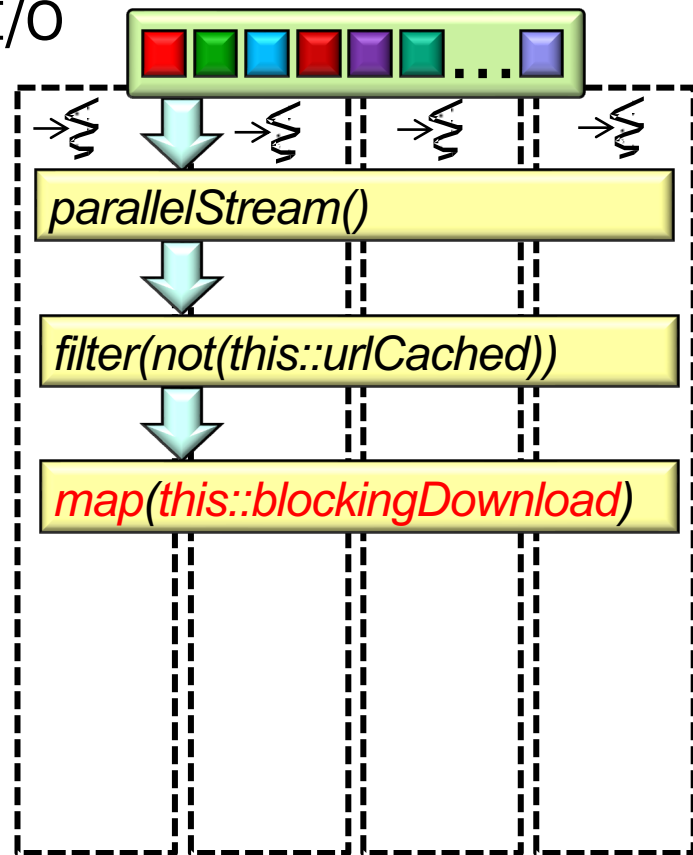
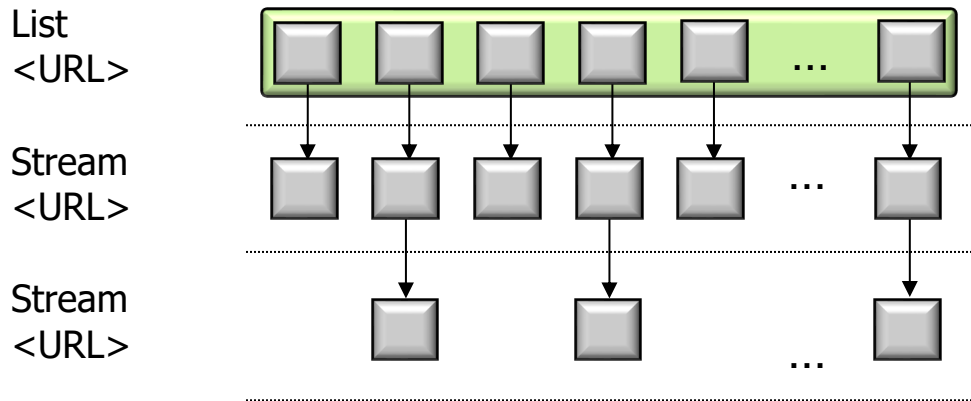


Input a stream of non-cached image URLs



Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O



Download non-cached images in parallel

Visualizing Parallel Streams in ImageStreamGang

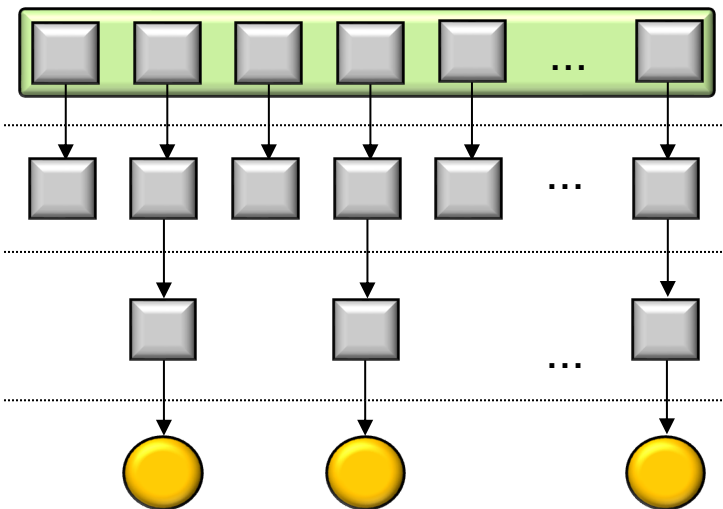
- This app uses a parallel stream with blocking I/O

List
<URL>

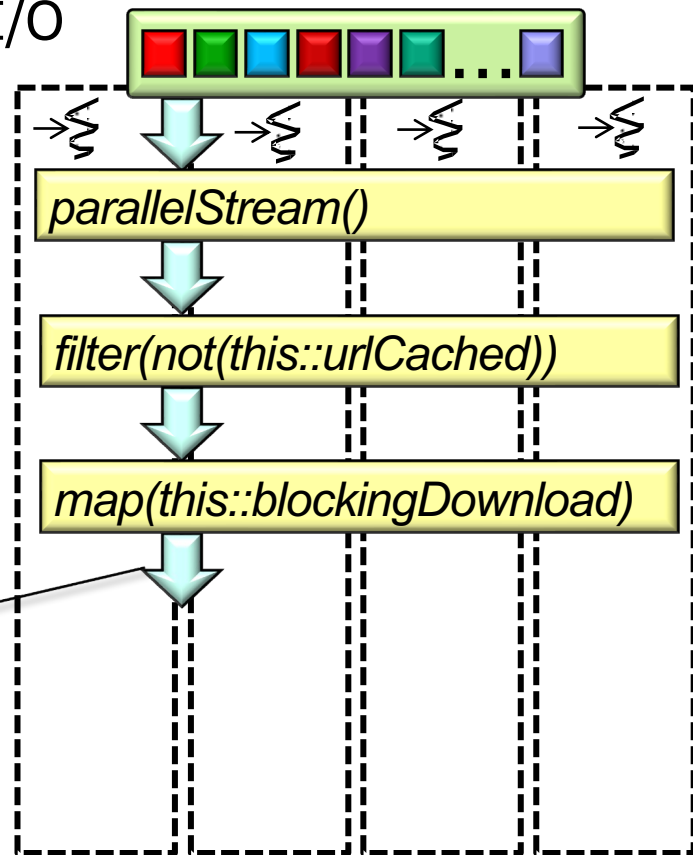
Stream
<URL>

Stream
<URL>

Stream
<Image>



*Output a stream of
downloaded images*



Visualizing Parallel Streams in ImageStreamGang

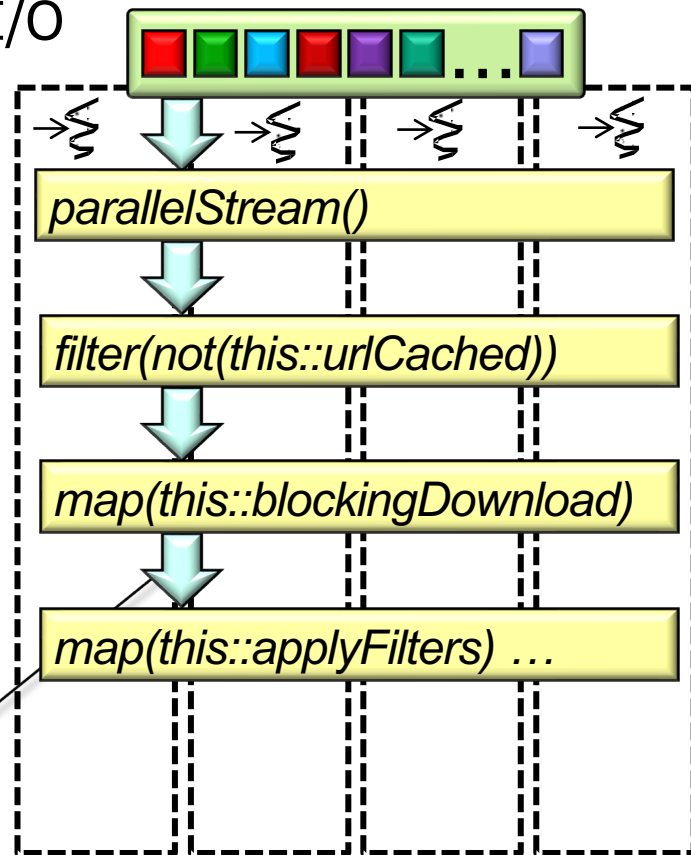
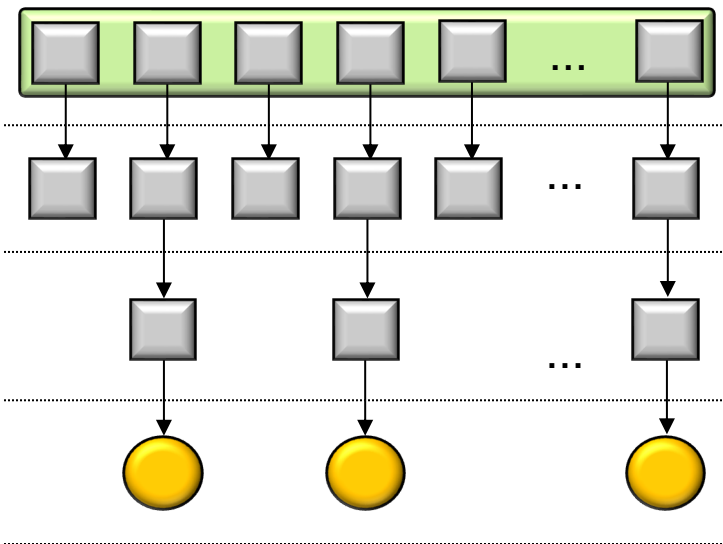
- This app uses a parallel stream with blocking I/O

List
<URL>

Stream
<URL>

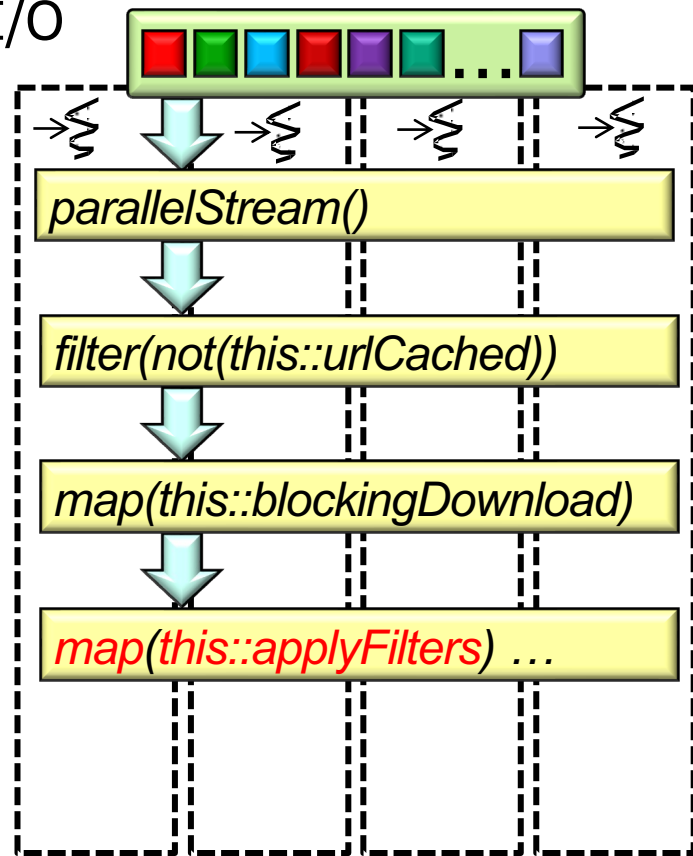
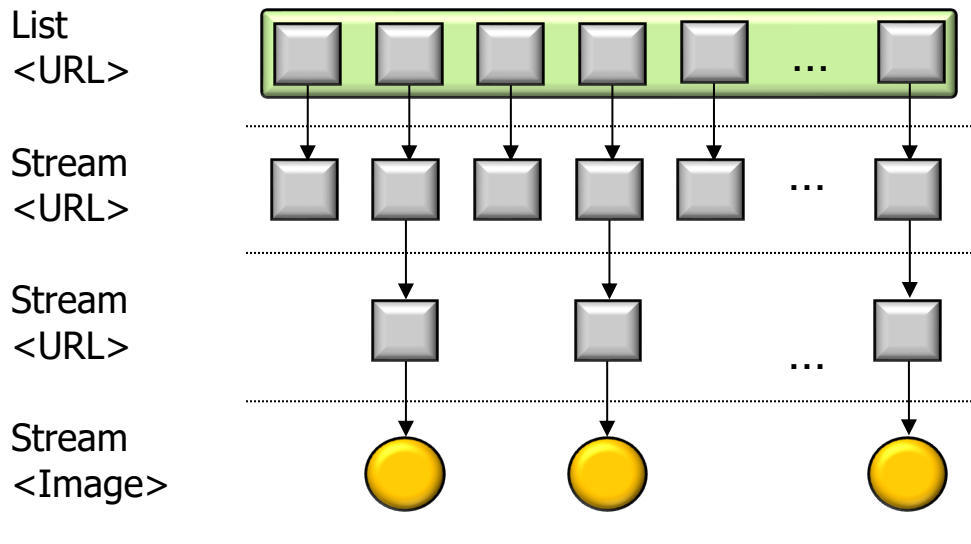
Stream
<URL>

Stream
<Image>



Visualizing Parallel Streams in ImageStreamGang

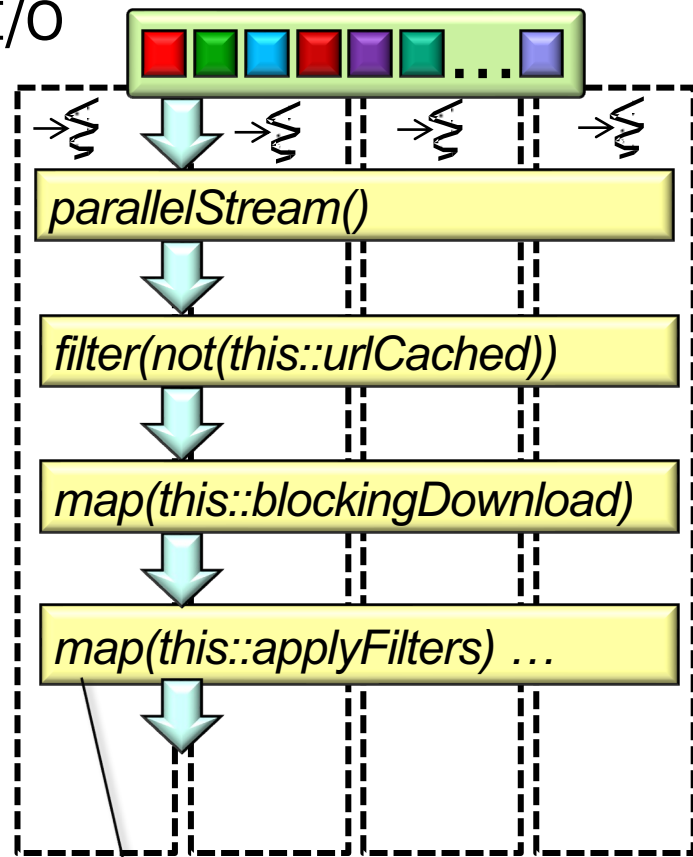
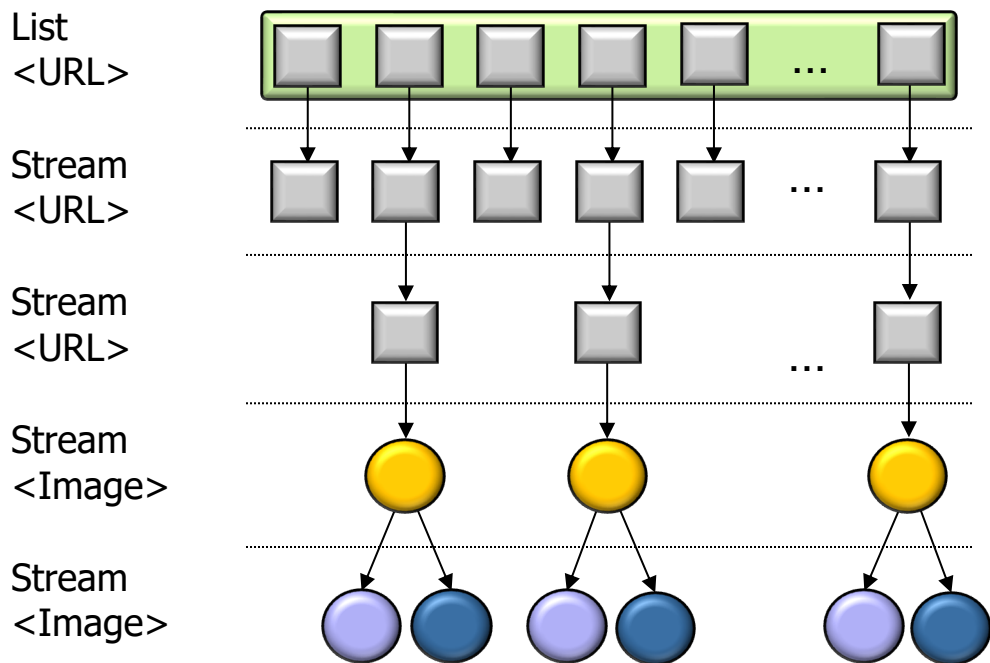
- This app uses a parallel stream with blocking I/O



Apply filters to each image in parallel & store filtered images in file system

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O



Output a stream of streams of filtered & stored images

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

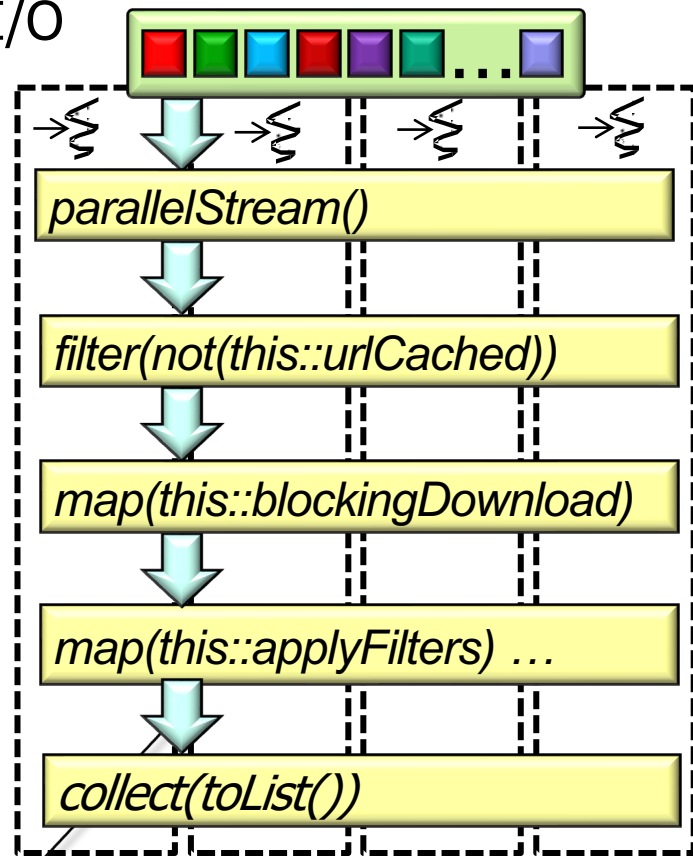
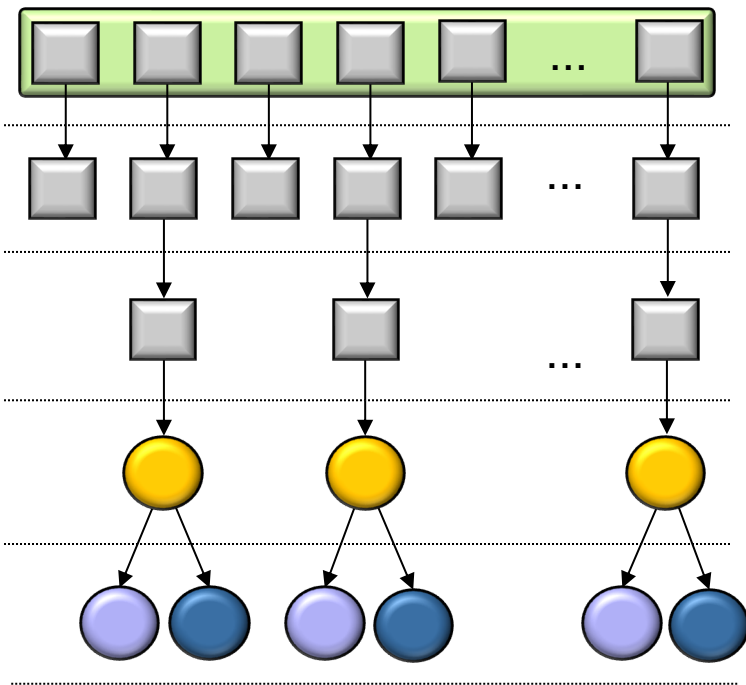
List
<URL>

Stream
<URL>

Stream
<URL>

Stream
<Image>

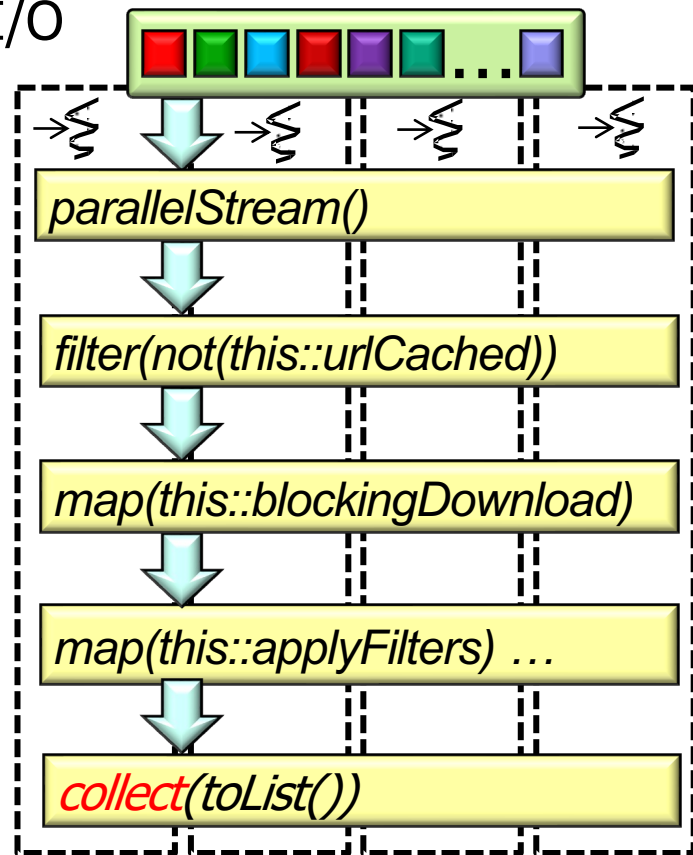
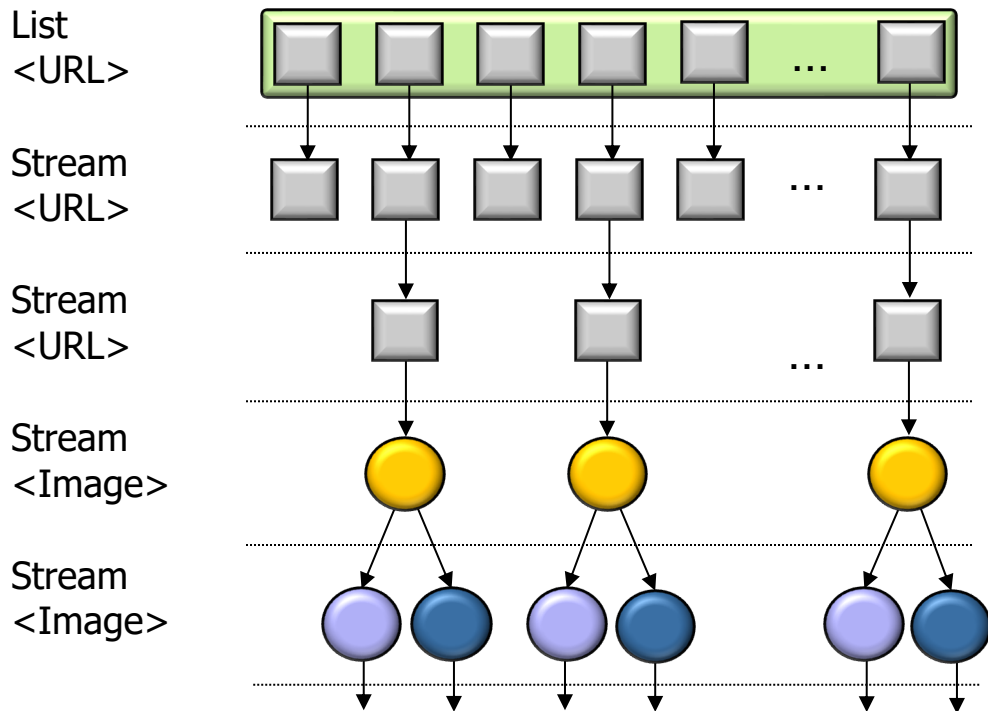
Stream
<Image>



Input a stream of filtered & stored images

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O



Trigger intermediate operation processing

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O

List
<URL>

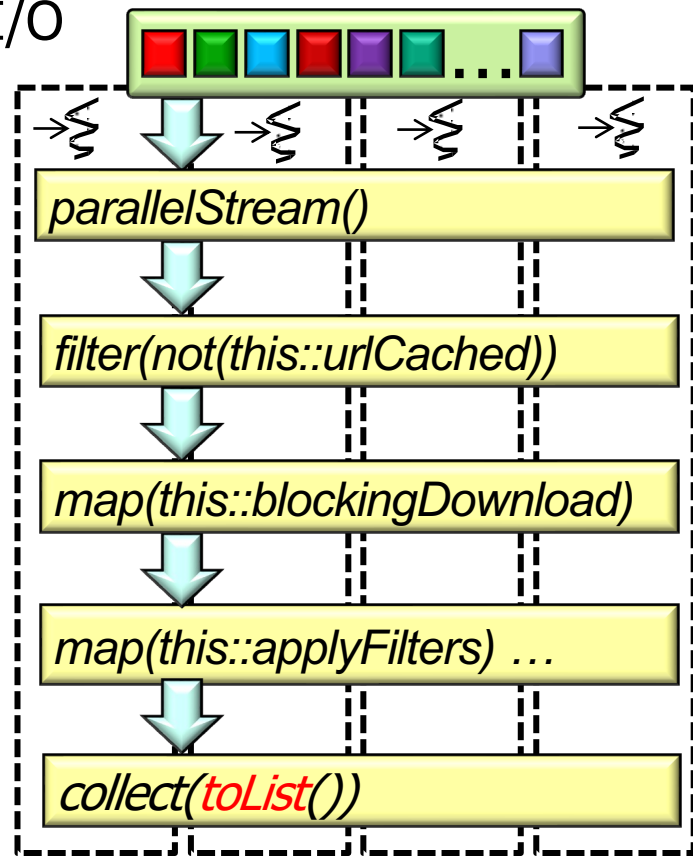
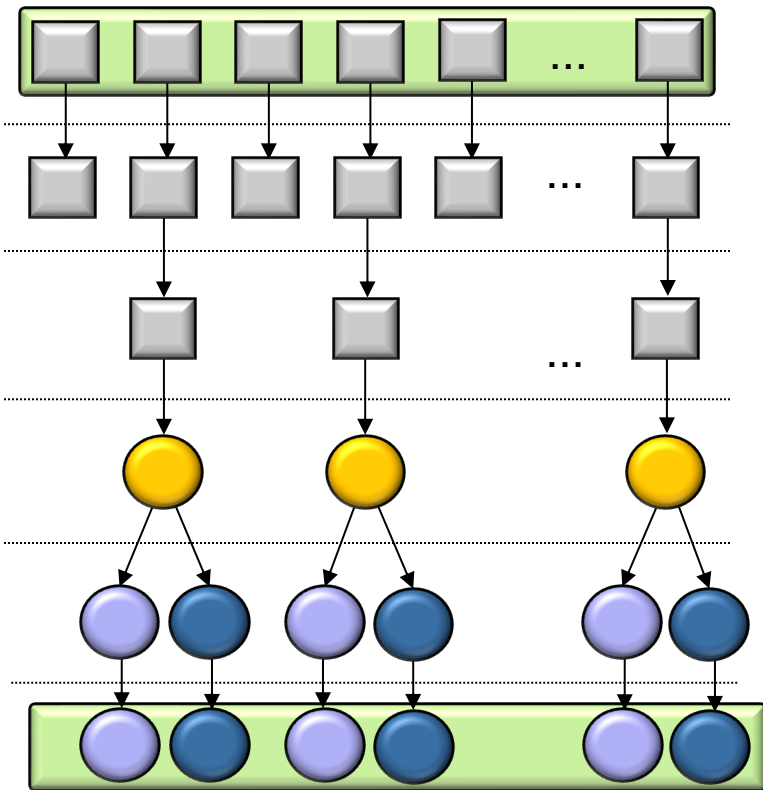
Stream
<URL>

Stream
<URL>

Stream
<Image>

Stream
<Image>

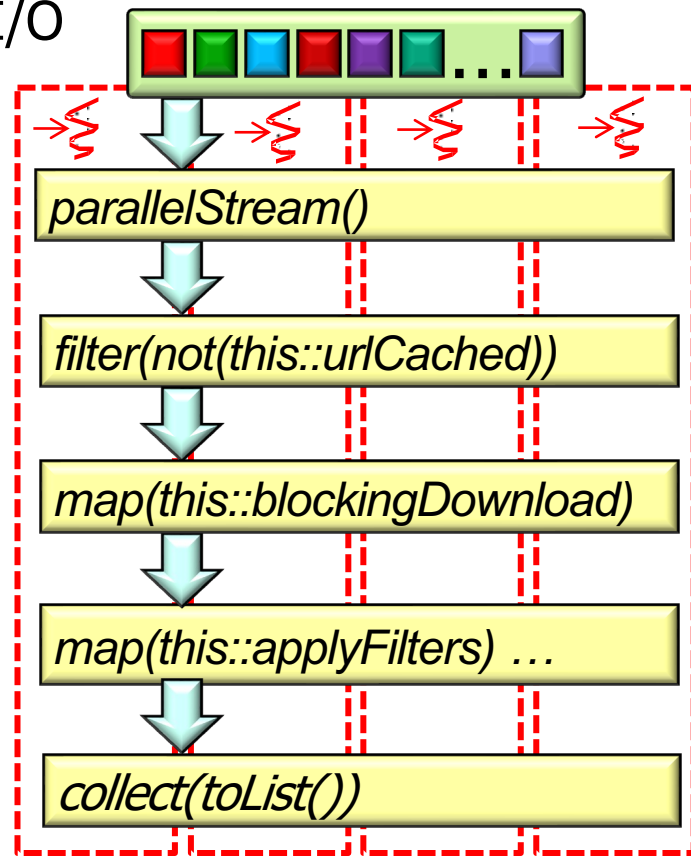
List
<Image>



Return a list of filtered & stored images

Visualizing Parallel Streams in ImageStreamGang

- This app uses a parallel stream with blocking I/O
 - Ignore cached images
 - Download non-cached images
 - Apply list of filters to each image
 - Store filtered images in the file system
 - Display images to the user (after triggering stream processing)



The Java streams framework orchestrates all these steps in parallel

End of Visualizing the Behaviors of the Java Parallel ImageStreamGang Case Study