

Recognizing Benefits of Java Streams

Douglas C. Schmidt

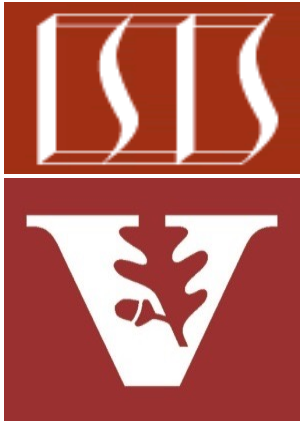
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

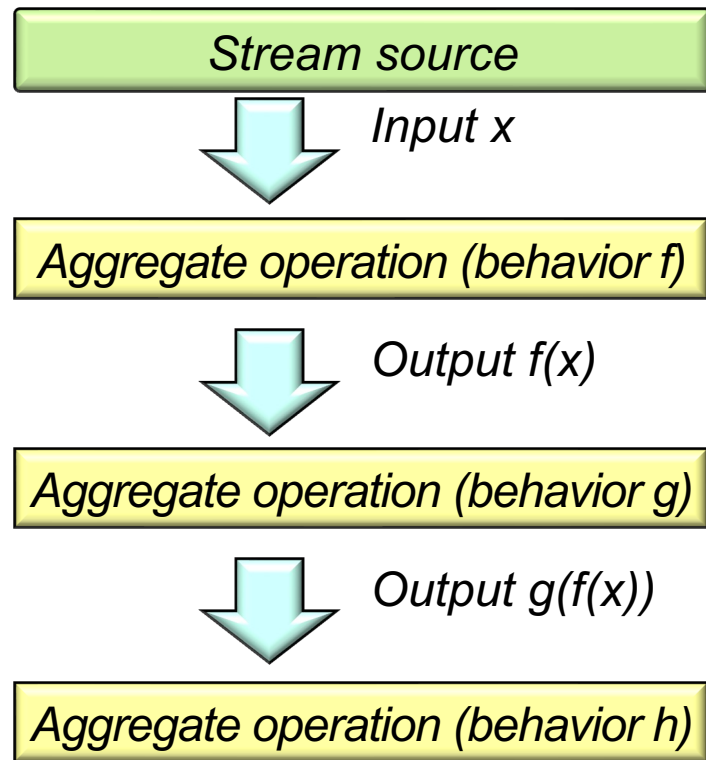
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

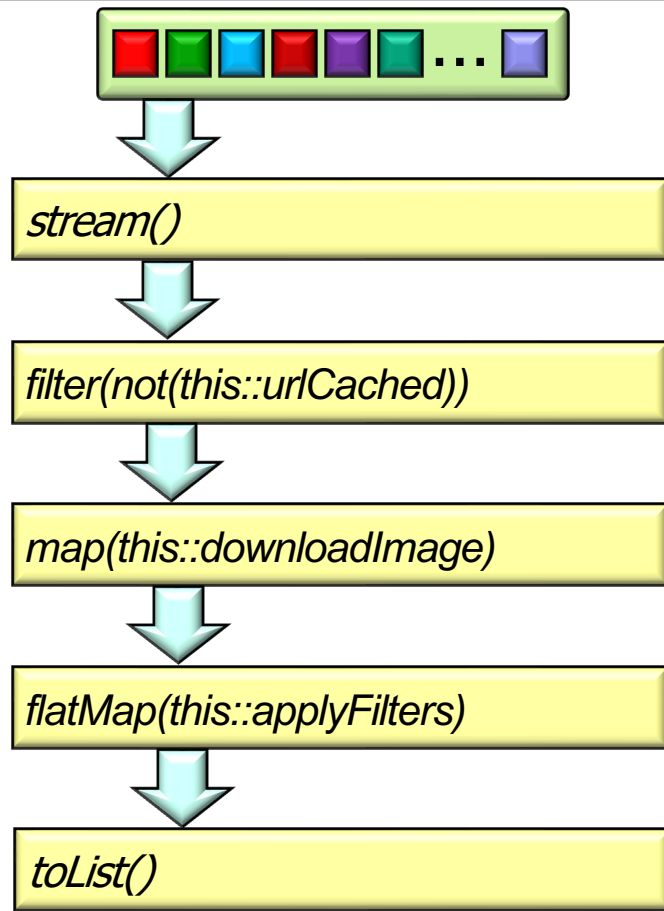
- Understand the structure & functionality of Java streams, e.g.,
 - Fundamentals of streams
 - Benefits of streams
 - Creating a stream
 - Aggregate operations in a stream
 - Applying streams in practice
 - Sequential vs. parallel streams
 - Common programming hazards of parallel streams
- Benefits of streams



Benefits of Java Streams

Benefits of Java Streams

- Java streams provide several key benefits to programs & programmers



Benefits of Java Streams

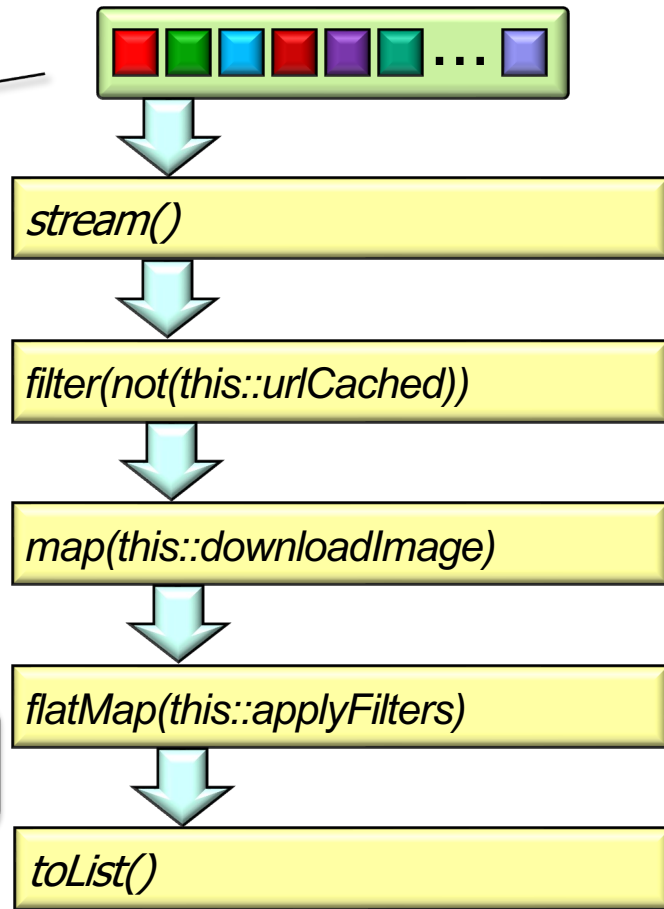
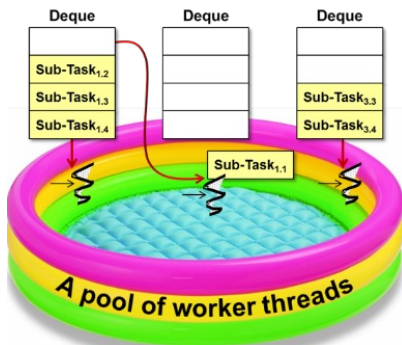
- Java streams provide several key benefits to programs & programmers

This case study program downloads, transforms, stores, & displays images

List of URLs to Download

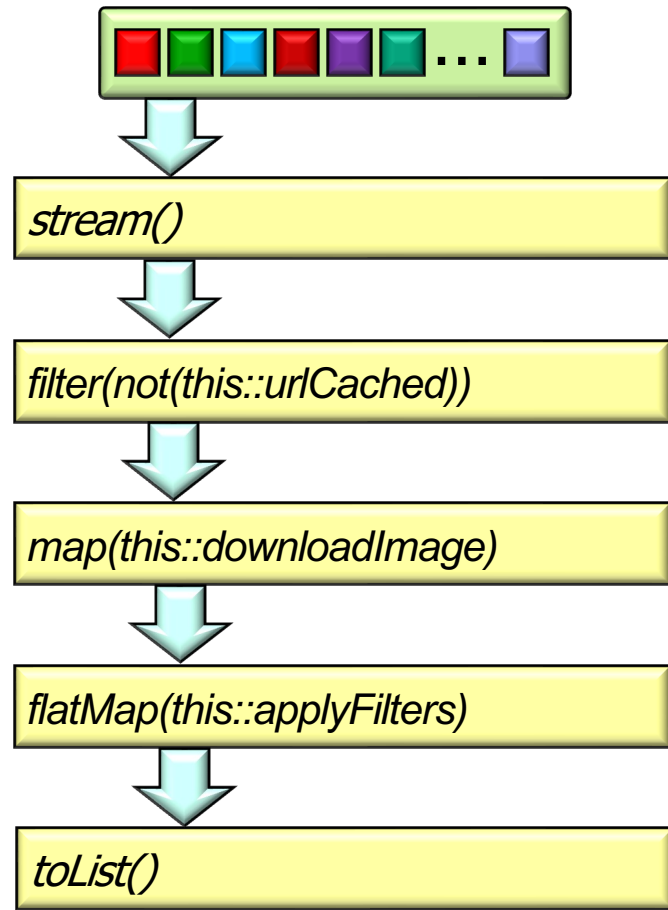


List of Transforms to Apply



Benefits of Java Streams

- Java streams provide several key benefits to programs & programmers, e.g.
 - **Concise & readable**
 - Declarative paradigm focuses on *what* functions to perform, not *how* to perform them



See en.wikipedia.org/wiki/Declarative_programming

Benefits of Java Streams

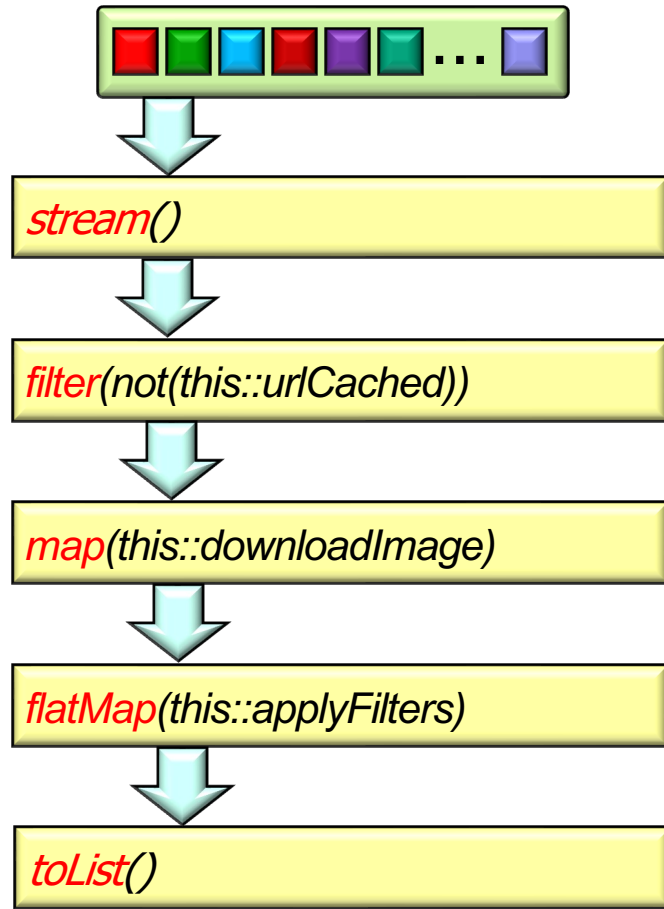
- Java streams provide several key benefits to programs & programmers, e.g.

- **Concise & readable**

- Declarative paradigm focuses on *what* functions to perform, not *how* to perform them



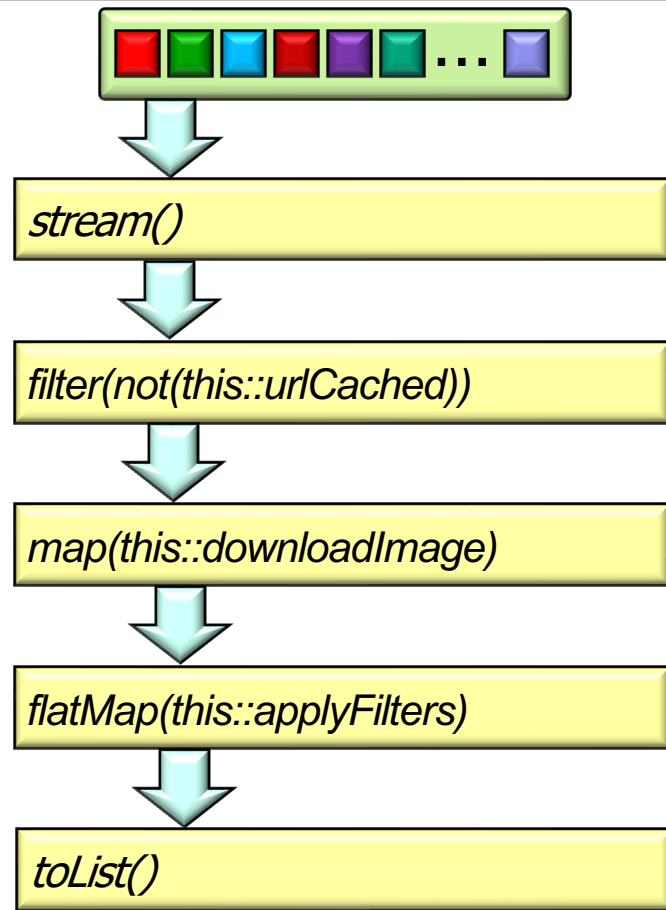
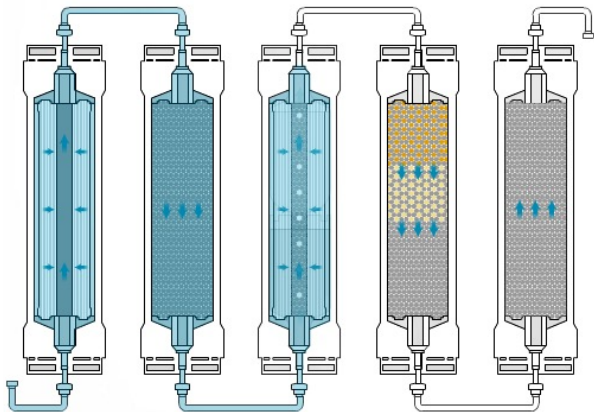
e.g., no Java control-flow operations are applied in this stream



See docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html

Benefits of Java Streams

- Java streams provide several key benefits to programs & programmers, e.g.
 - **Concise & readable**
 - **Flexible & composable**
 - Functions are automatically & fluently connected together

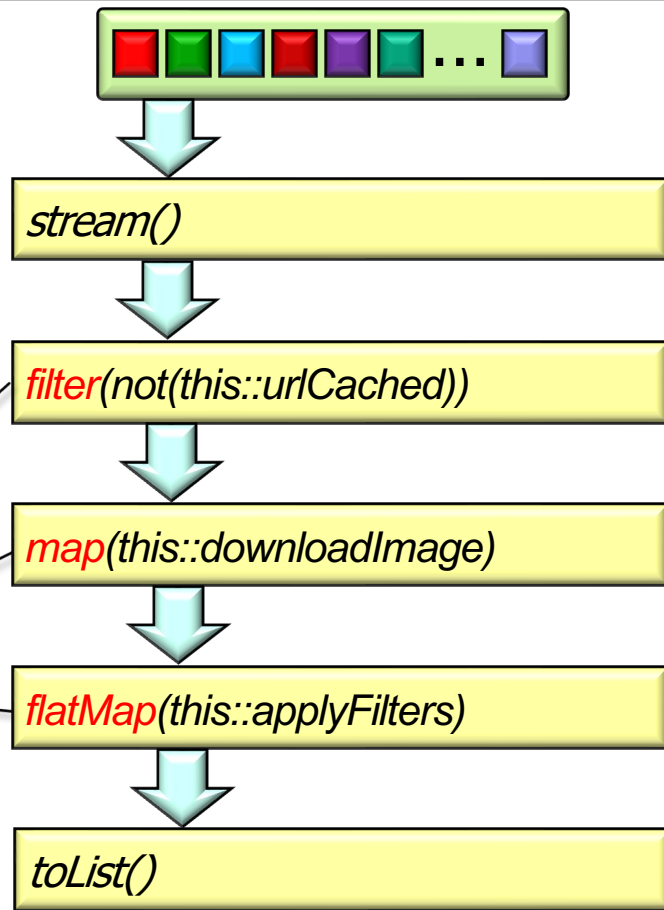


See en.wikipedia.org/wiki/Fluent_interface

Benefits of Java Streams

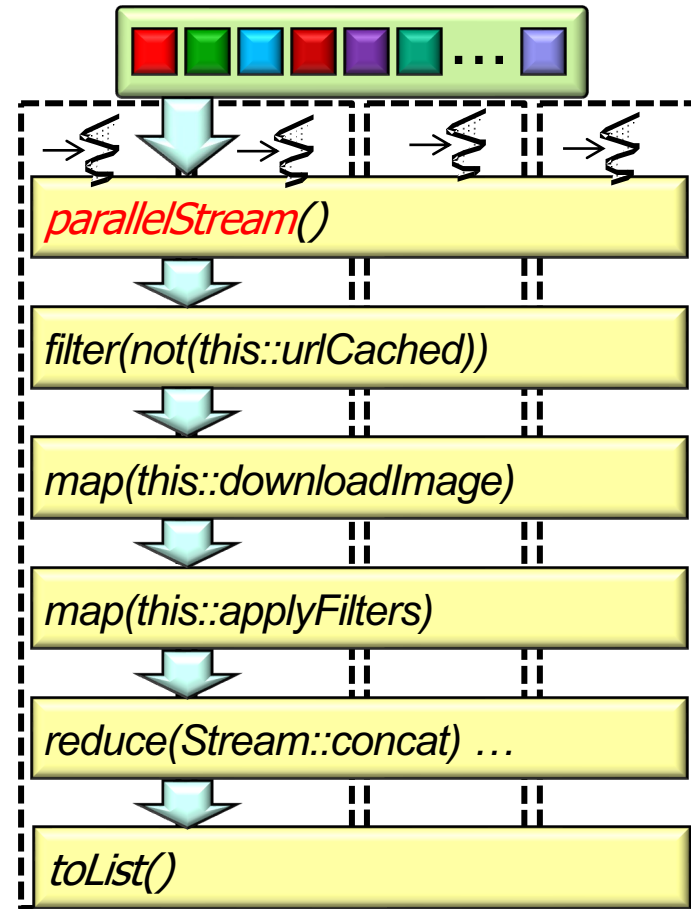
- Java streams provide several key benefits to programs & programmers, e.g.
 - **Concise & readable**
 - **Flexible & composable**
 - Functions are automatically & fluently connected together

e.g., the output from filter() is passed as the input to map(), whose output is passed as the input to flatMap() etc.



Benefits of Java Streams

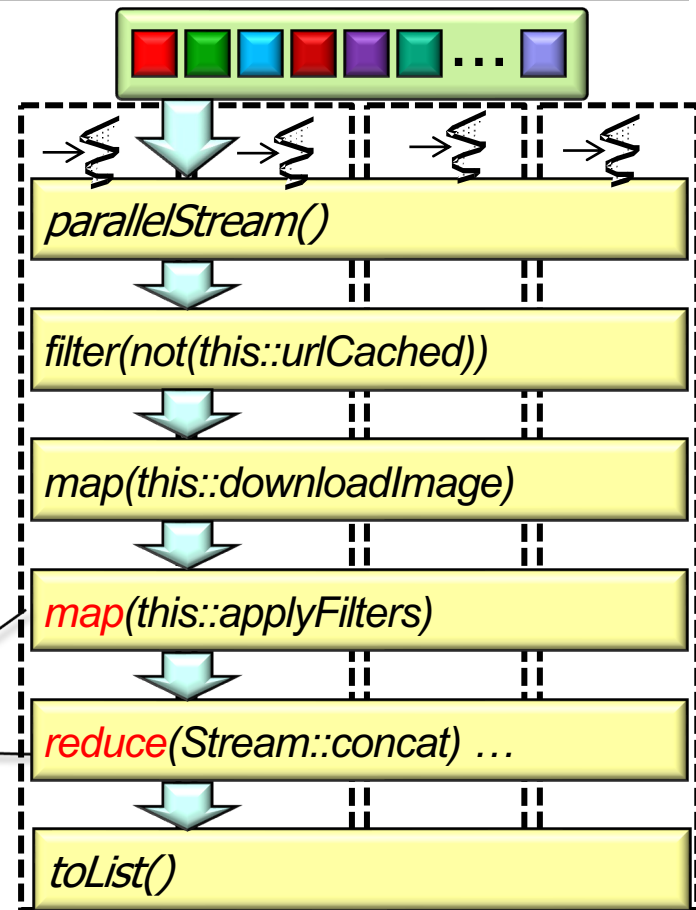
- Java streams provide several key benefits to programs & programmers, e.g.
 - **Concise & readable**
 - **Flexible & composable**
 - **Simplified scalability**
 - Parallelize performance without the need to write any multi-threaded code



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

Benefits of Java Streams

- Java streams provide several key benefits to programs & programmers, e.g.
 - **Concise & readable**
 - **Flexible & composable**
 - **Simplified scalability**
 - Parallelize performance without the need to write any multi-threaded code

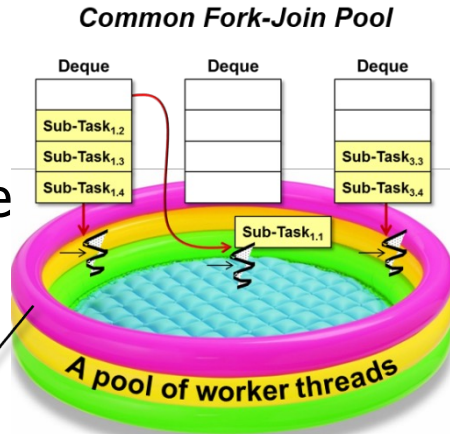


Requires a slight tweak in intermediate operations due to quirks of flatMap()

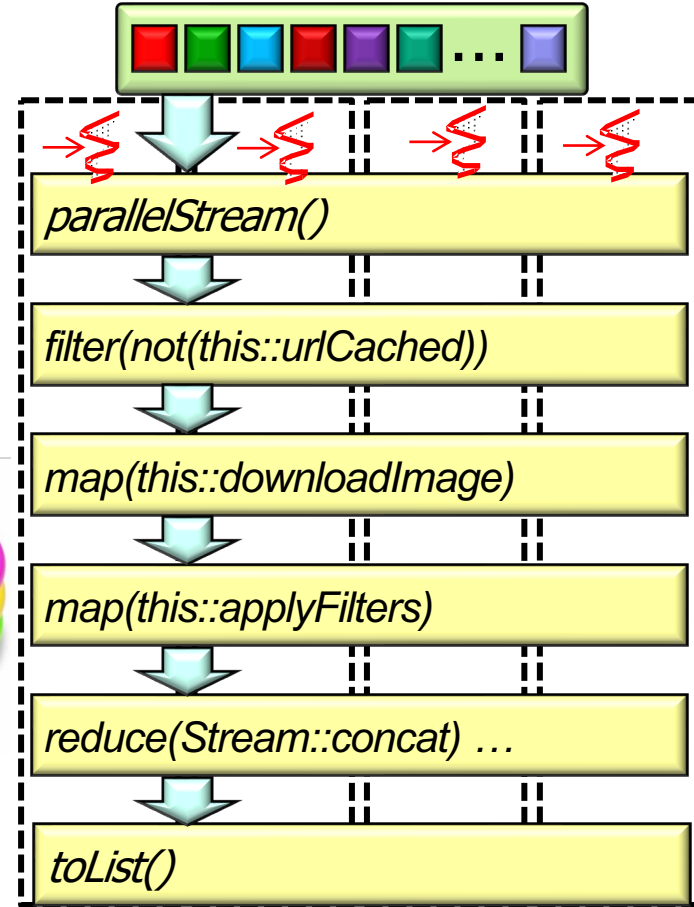
See stackoverflow.com/questions/45038120/parallel-flatmap-always-sequential

Benefits of Java Streams

- Java streams provide several key benefits to programs & programmers, e.g.
 - **Concise & readable**
 - **Flexible & composable**
 - **Simplified scalability**
 - Parallelize performance without the need to write any multi-threaded code



The common fork-join pool of worker threads is applied to process behaviors in parallel



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

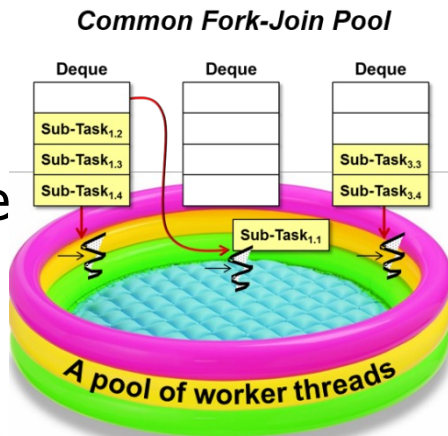
Benefits of Java Streams

- Java streams provide several key benefits to programs & programmers, e.g.

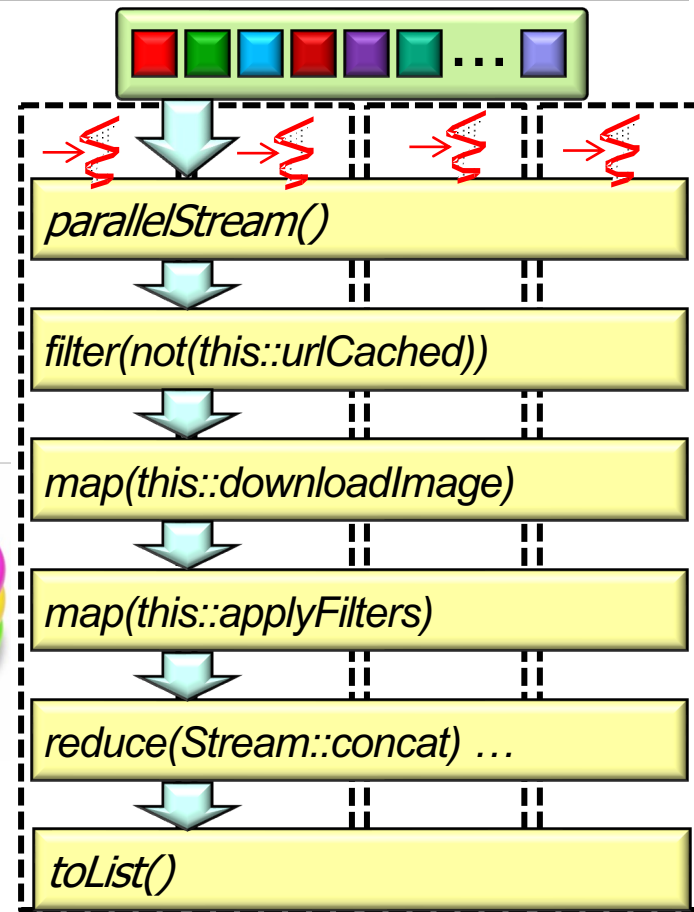
- Concise & readable
- Flexible & composable

- **Simplified scalability**

- Parallelize performance without the need to write any multi-threaded code



Data mapped automatically to underlying processor cores



See gee.cs.oswego.edu/dl/papers/fj.pdf

End of Recognizing Benefits of Java Streams