

Learn When to Use Java Parallel 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

- Learn when to use parallel streams



Parallelism is Not a Panacea

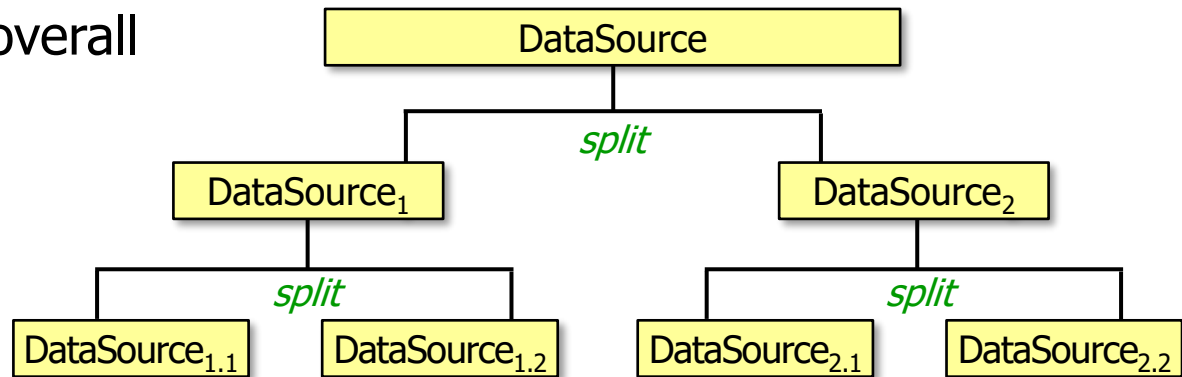
Parallelism is Not a Panacea

- A parallel program *always* does more work than a non-parallel program



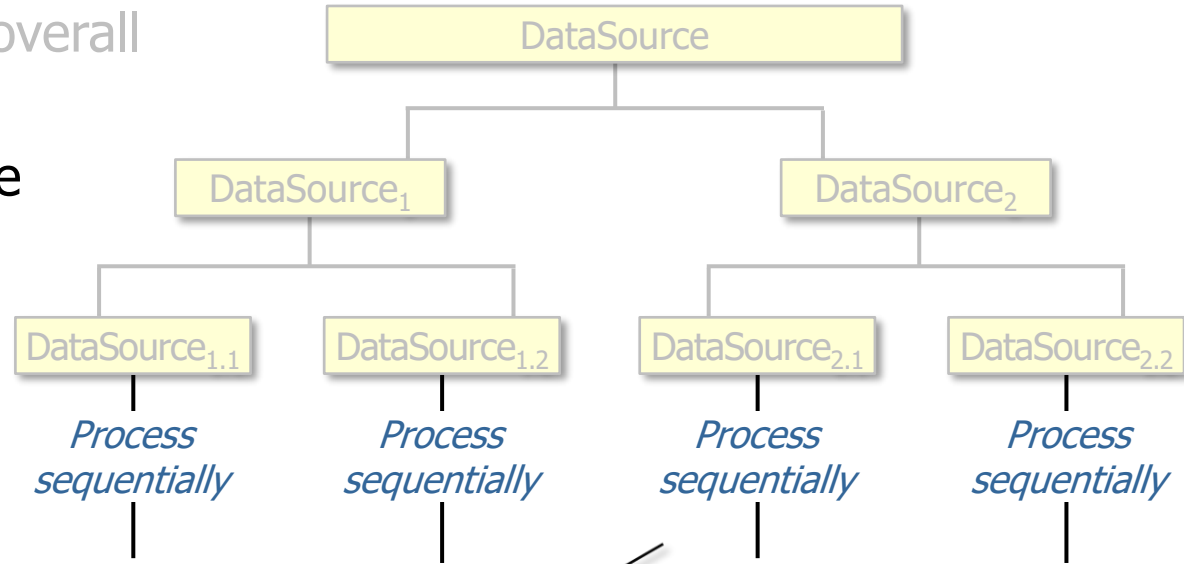
Parallelism is Not a Panacea

- A parallel program *always* does more work than a non-parallel program, e.g.
 - It needs to partition the overall task into sub-tasks



Parallelism is Not a Panacea

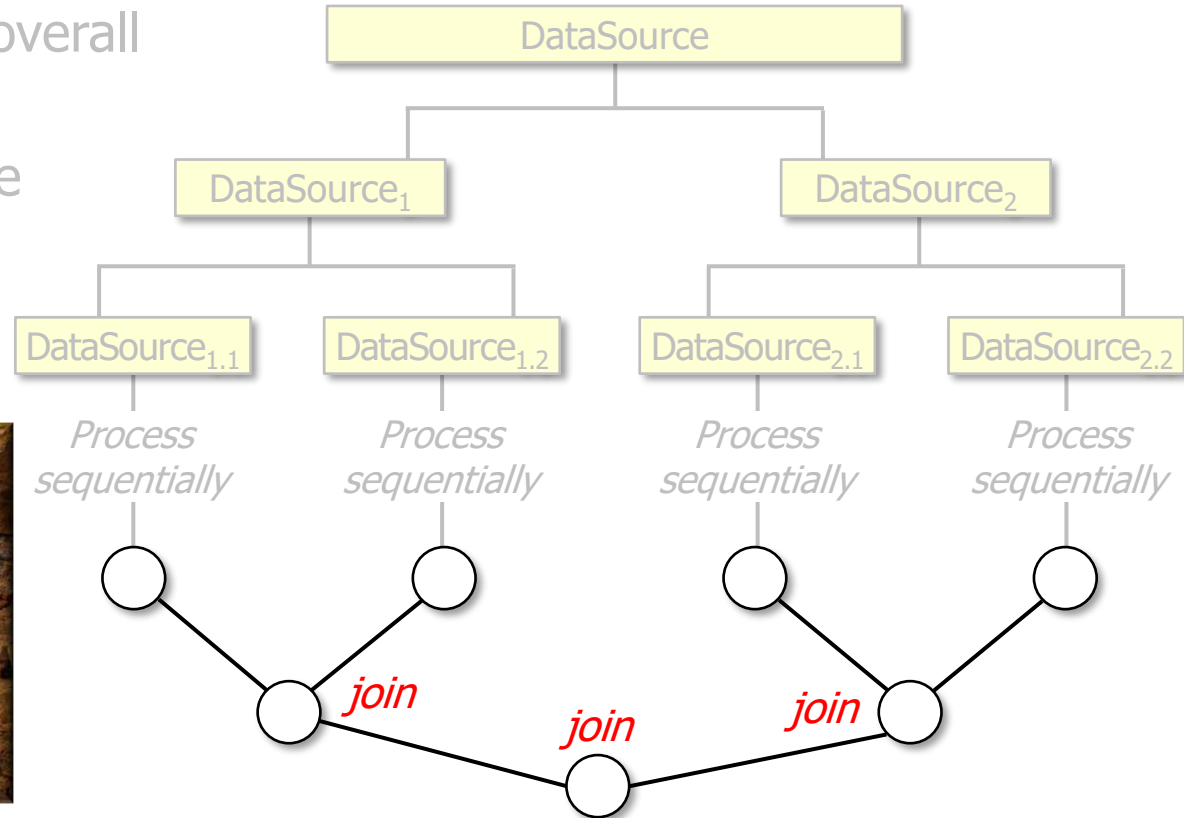
- A parallel program *always* does more work than a non-parallel program, e.g.
 1. It needs to partition the overall task into sub-tasks
 2. It needs to process all the sub-tasks



This step is typically *all* that a sequential program does!

Parallelism is Not a Panacea

- A parallel program *always* does more work than a non-parallel program, e.g.
 - It needs to partition the overall task into sub-tasks
 - It needs to process all the sub-tasks
 - It needs to combine the sub-task results



Parallelism is Not a Panacea

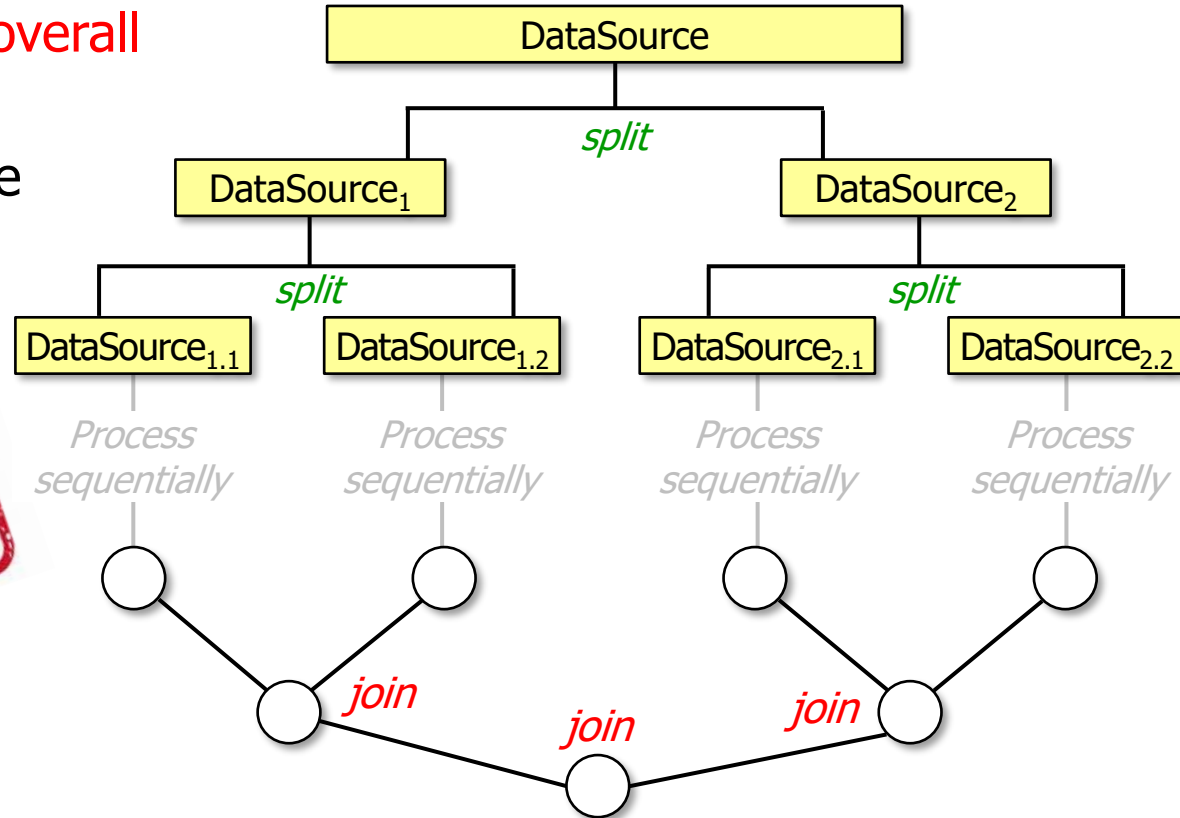
- A parallel program *always* does more work than a non-parallel program, e.g.

1. It needs to partition the overall task into sub-tasks

2. It needs to process all the sub-tasks

3. It needs to combine the sub-task results

EXTRA COST



A sequential program needn't do steps 1 & 3..

Parallelism is Not a Panacea

- Java parallel streams are thus useful in some (but not all) conditions



CAUTION
THIS MACHINE
HAS NO BRAIN
USE YOUR OWN

See gee.cs.oswego.edu/dl/html/StreamParallelGuidance.html

When to Use Java Parallel Streams

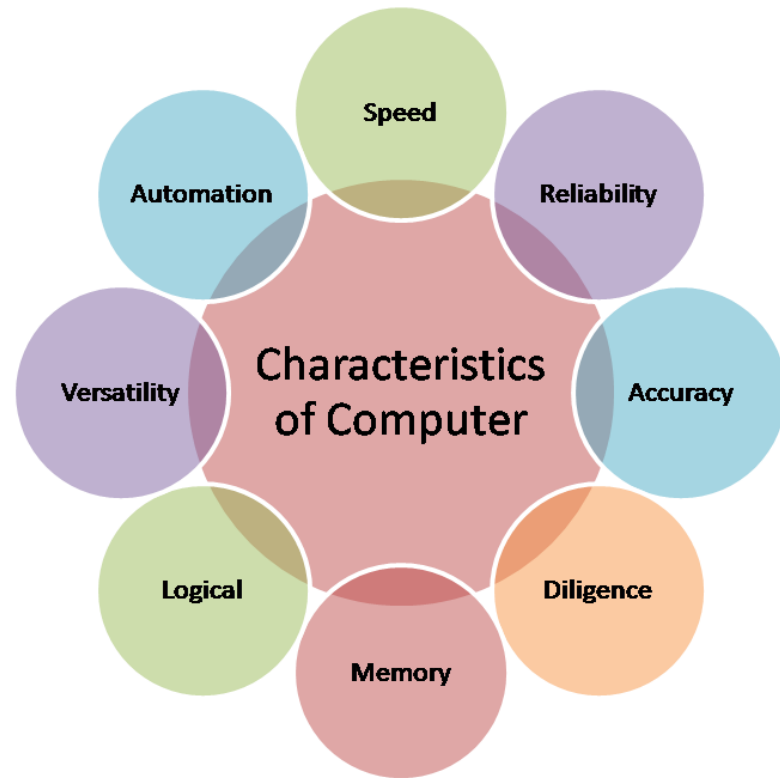
When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions



When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics



When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent

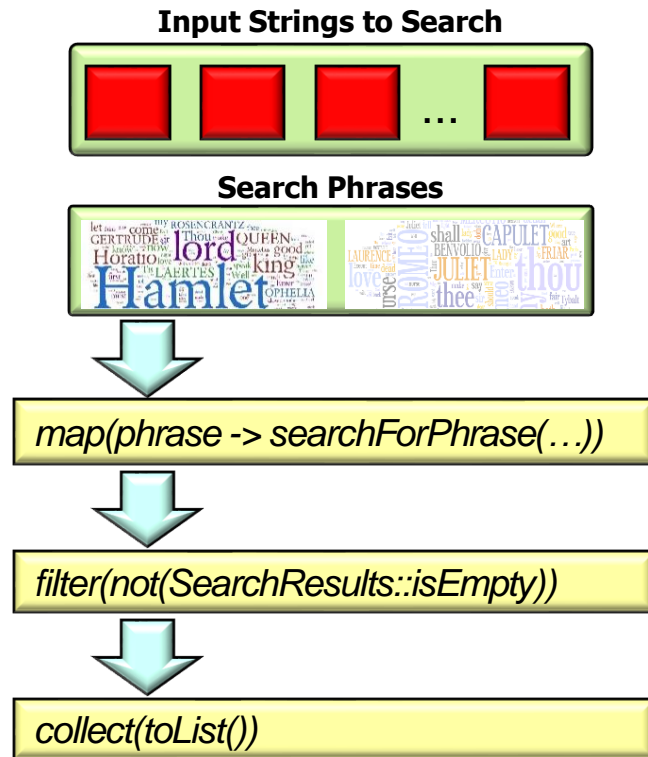
"Embarrassingly parallel" tasks have little/no dependency or need for communication between tasks or for sharing results between them



See en.wikipedia.org/wiki/Embarrassingly_parallel

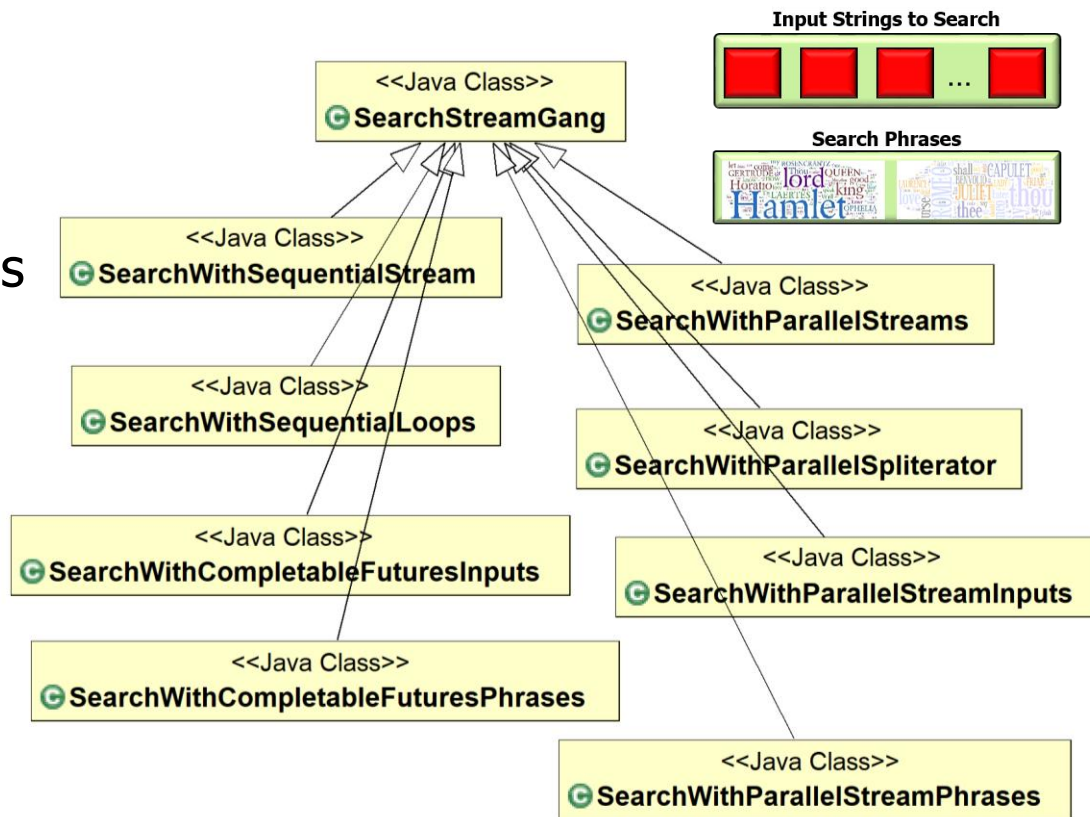
When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - e.g., searching for phrases in a list of input strings



When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - e.g., searching for phrases in a list of input strings

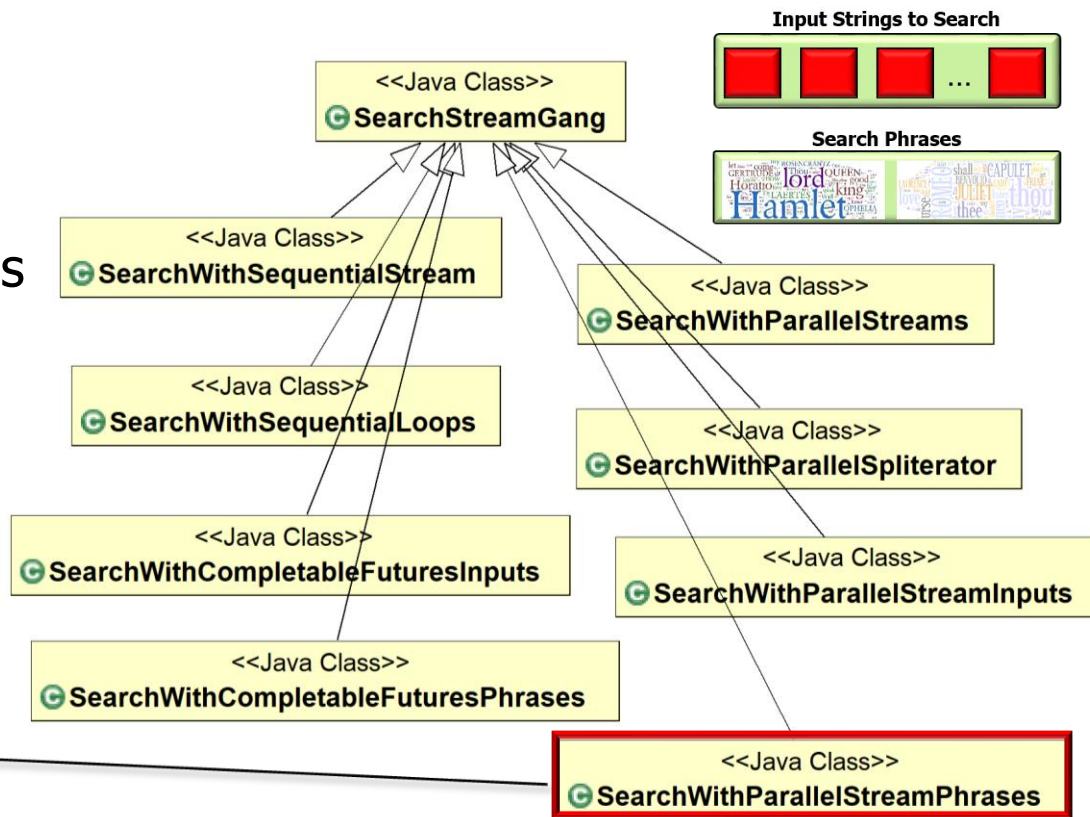


When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - e.g., searching for phrases in a list of input strings

Parallel streams can:

- search each phrase in parallel*
- search each input string in parallel*
- search chunks of each input string in parallel*

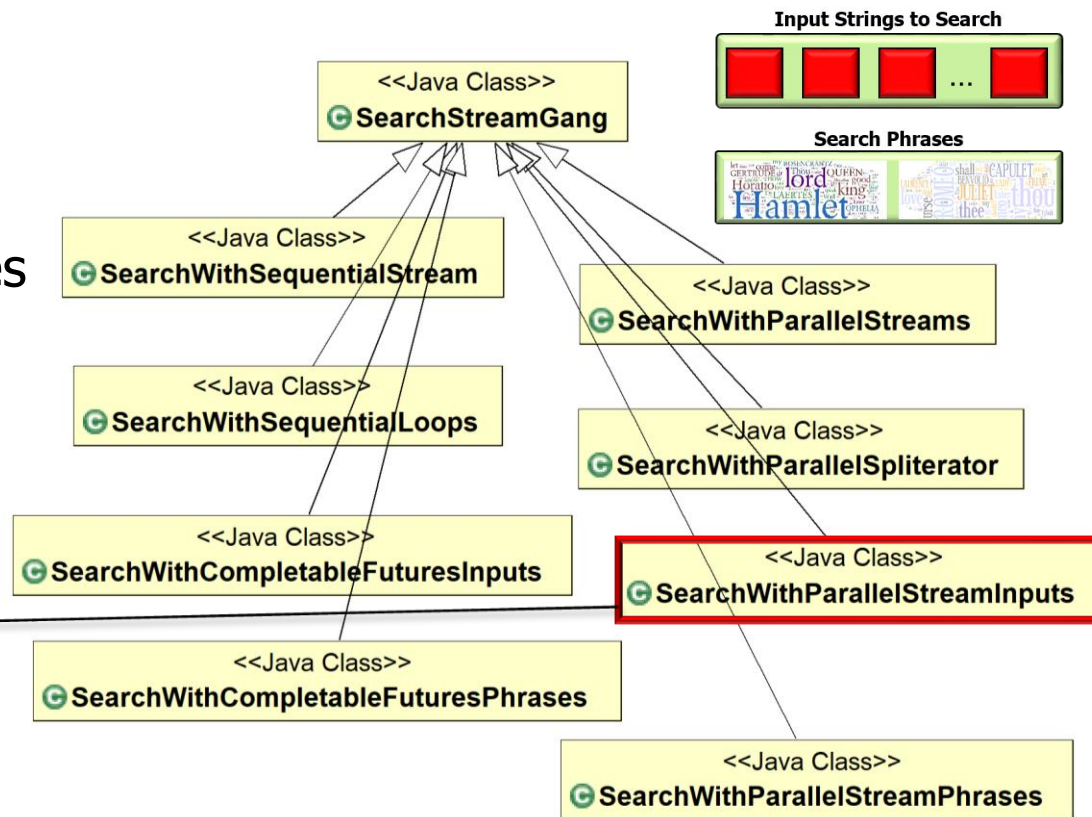


When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - e.g., searching for phrases in a list of input strings

Parallel streams can:

- search each phrase in parallel*
- search each input string in parallel*
- search chunks of each input string in parallel*

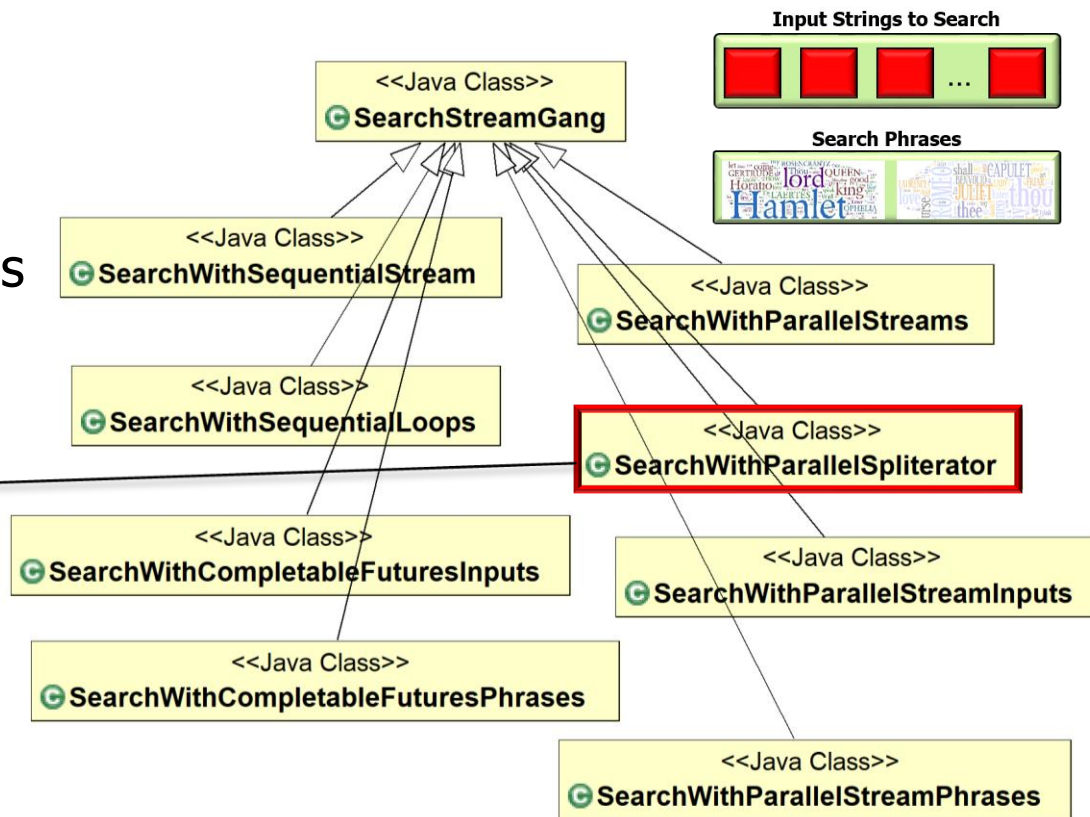


When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - e.g., searching for phrases in a list of input strings

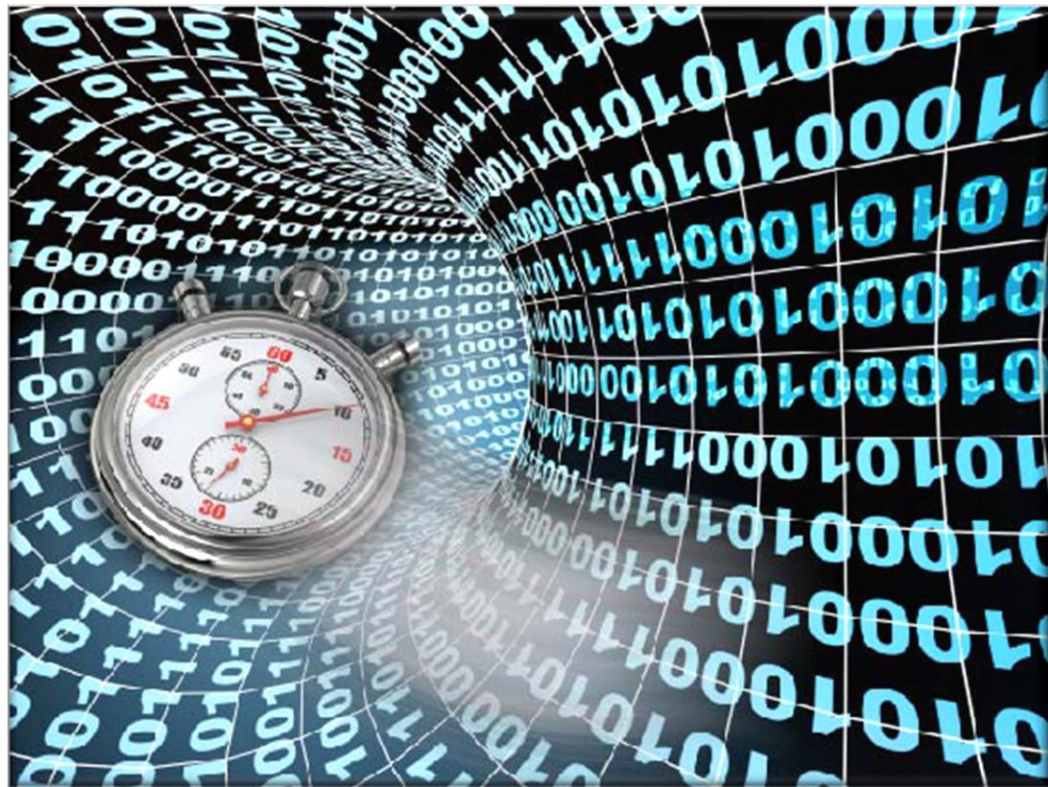
Parallel streams can:

- search each phrase in parallel*
- search each input string in parallel*
- search chunks of each input string in parallel*



When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - Computationally expensive
 - e.g., when behavior(s) applied to each element take a “long-time” to run



See developer.ibm.com/articles/j-java-streams-5-brian-goetz

When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - Computationally expensive
 - Applied to many elements of data sources



See developer.ibm.com/articles/j-java-streams-5-brian-goetz

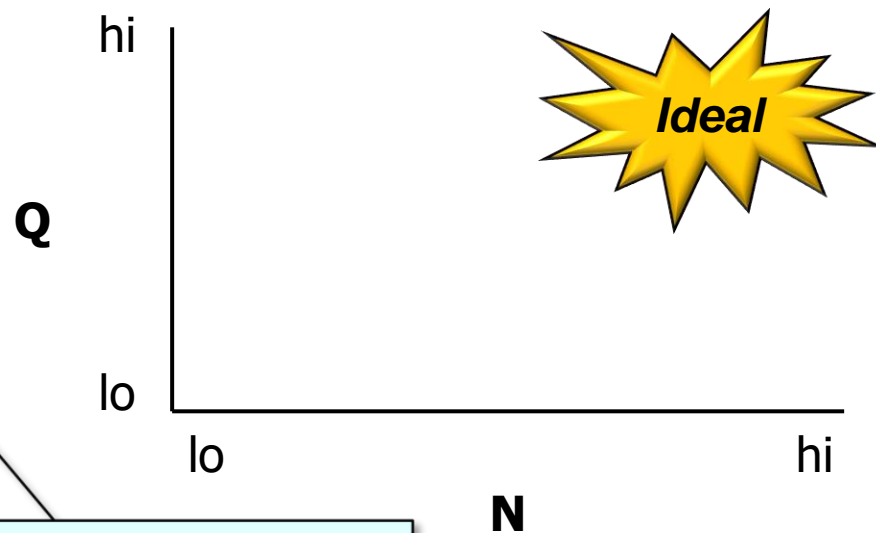
When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - Computationally expensive
 - Applied to many elements of data sources
 - Where these sources can be split efficiently/evenly



When to Use Java Parallel Streams

- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - Computationally expensive
 - Applied to many elements of data sources

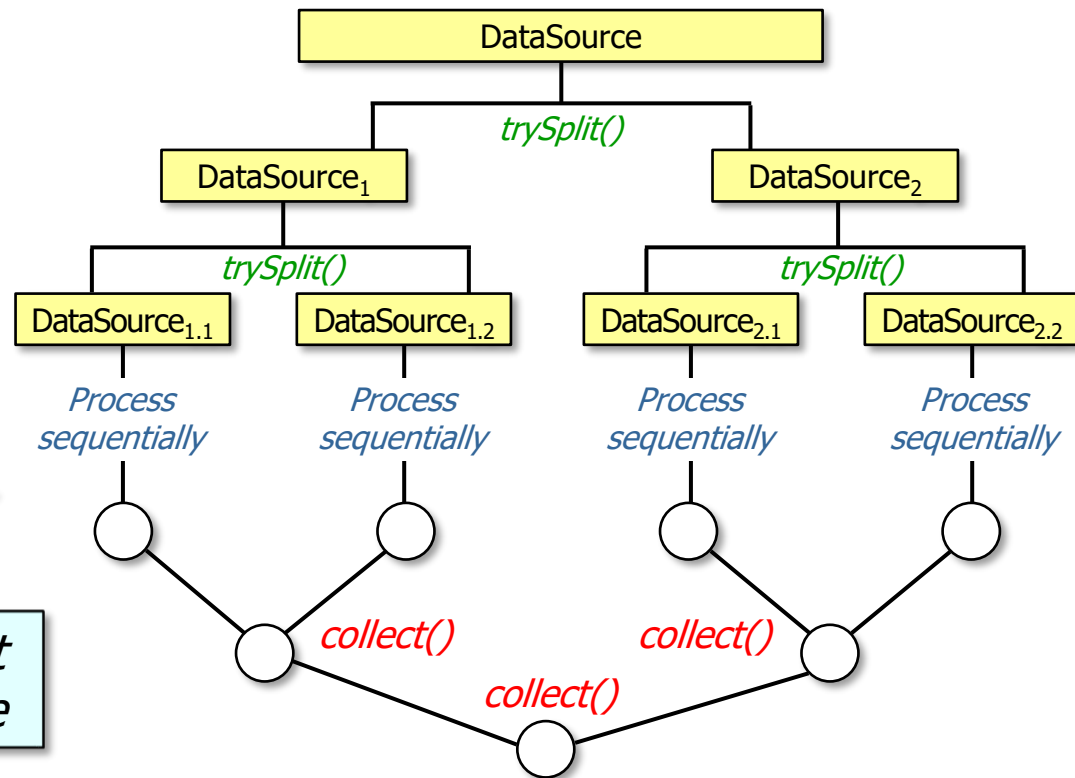


The "NQ" model:

- *N is the # of data elements to process per thread*
- *Q quantifies how CPU-intensive the processing is*

When to Use Java Parallel Streams

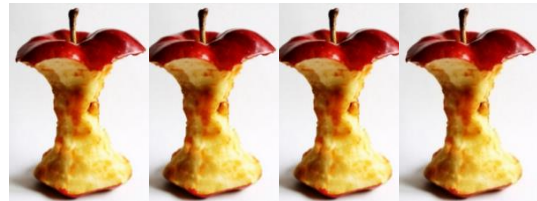
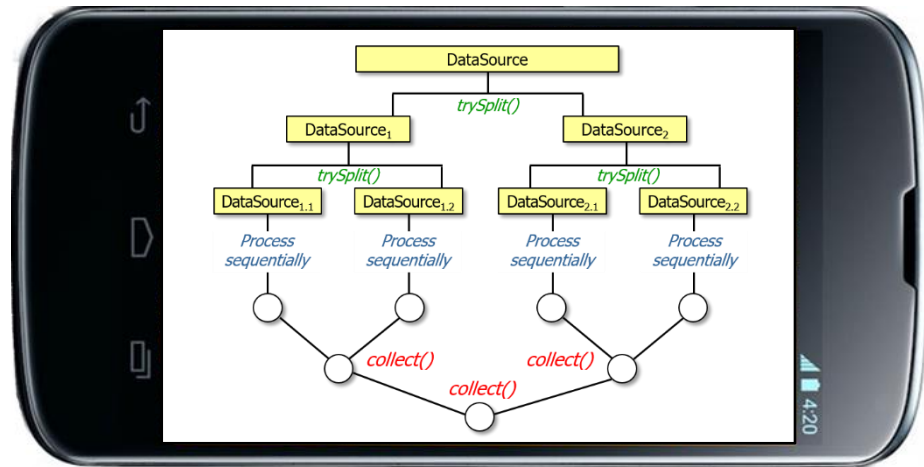
- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - Independent
 - Computationally expensive
 - Applied to many elements of data sources



e.g., searching for phrases that match in works of Shakespeare

When to Use Java Parallel Streams

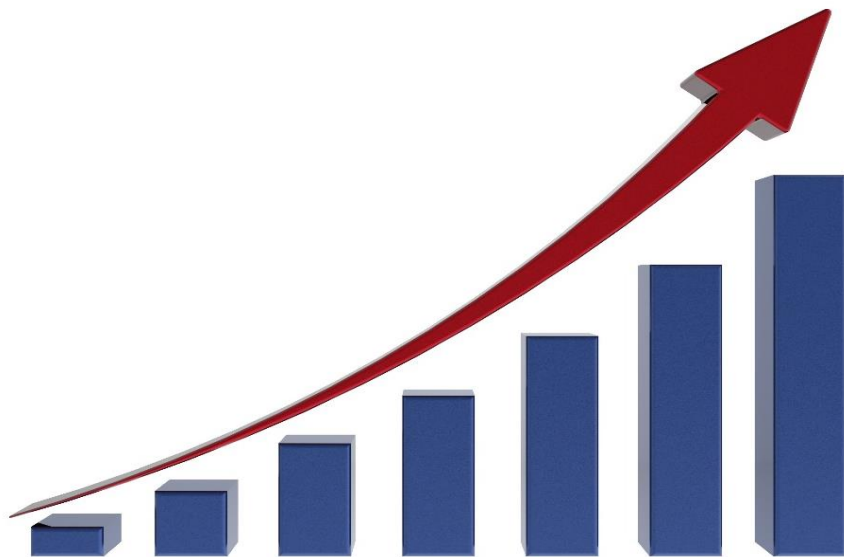
- Java parallel streams are most useful under certain conditions, e.g.
 - When behaviors have certain characteristics
 - If there are multiple cores



See blog.oio.de/2016/01/22/parallel-stream-processing-in-java-8-performance-of-sequential-vs-parallel-stream-processing

When to Use Java Parallel Streams

- Under the right conditions Java parallel streams can scale up nicely on multi-core & many-core processors



Input Strings to Search



Search Phrases



Starting SearchStreamGangTest

PARALLEL_SPLITTERATOR 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 www.infoq.com/presentations/parallel-java-se-8

End of Learn When to Use Java Parallel Streams