Java Sequential SearchStreamGang

Example: Introduction

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 SearchStreamGang program

See [github.com/douglascraigschmidt/LiveLessons/tree/master/SearchStreamGang](github.com/douglascraigschmidt/LiveLessons/tree/master/SearchStreamGang)

```
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
```

```
45,000+ phrases

Search Phrases

stream()

map(phrase -> searchForPhrase(…))

filter(not(SearchResults::isEmpty))

collect(toList())
```
Learning Objectives in this Part of the Lesson

• Understand the design of the SearchStreamGang program

I DON'T ALWAYS PROGRAM IN JAVA

BUT WHEN I DO, I USE JAVA 8

This example is more interesting than the SimpleSearchStream program
Overview of SearchStreamGang
Overview of SearchStreamGang

- SearchStreamGang revises SearchTaskGang to use functional programming & streams instead of OO programming.

Overview of SearchStreamGang

- SearchStreamGang revises SearchTaskGang to use functional programming & streams instead of OO programming.
- SearchTaskGang showcases the Java executor framework for tasks that are “embarrassingly parallel.”
Overview of SearchStreamGang

- SearchStreamGang is also a more powerful revision of SimpleSearchStream

**Input Strings to Search**

```
map(phrase -> searchForPhrase(…))
filter(not(SearchResults::isEmpty))
collect(toList())
```

**Search Phrases**

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

**Input String to Search**

```
map(word -> searchForWord(…))
filter(not(SearchResults::isEmpty))
collect(toList())
```

Overview of SearchStreamGang

- SearchStreamGang is also a more powerful revision of SimpleSearchStream, e.g.
  - It uses regular expressions to find phrases in works of Shakespeare
Overview of SearchStreamGang

- SearchStreamGang is also a more powerful revision of SimpleSearchStream, e.g.
- It uses regular expressions to find phrases in works of Shakespeare

“...
My liege, and madam, to expostulate
What majesty should be, what duty is,
Why day is day, night is night, and time is time.
Were nothing but to waste night, day, and time.
Therefore, since brevity is the soul of wit,
And tediousness the limbs and outward flourishes,
I will be brief. ...”

"Brevity is the soul of wit"
Overview of SearchStreamGang

- SearchStreamGang is also a more powerful revision of SimpleSearchStream, e.g.

- It uses regular expressions to find phrases in works of Shakespeare

  “... What's in a name? That which we call a rose
By any other name would smell as sweet.
So Romeo would, were he not Romeo call'd,
Retain that dear perfection which he owes
Without that title. ...”

“What’s in a name? That which we call a rose
By any other name would smell as sweet.”

The phrases can also match across multiple lines
Overview of SearchStreamGang

- SearchStreamGang is also a more powerful revision of SimpleSearchStream, e.g.
  - It uses regular expressions to find phrases in works of Shakespeare
  - It defines a framework for comparing Java concurrency & parallelism strategies

- e.g., parallel streams, parallel spliterator, & completable futures
Overview of SearchStreamGang

- SearchStreamGang is also a more powerful revision of SimpleSearchStream, e.g.
  - It uses regular expressions to find phrases in works of Shakespeare
  - It defines a framework for comparing Java concurrency & parallelism strategies

This framework enables “apples-to-apples” performance comparisons
Overview of SearchStreamGang

- SearchStreamGang is also a more powerful revision of SimpleSearchStream, e.g.
- It uses regular expressions to find phrases in works of Shakespeare
- It defines a framework for comparing Java concurrency & parallelism strategies

We’ll cover Java concurrency/parallel strategies after covering sequential streams
Applying Sequential Streams to SearchStreamGang
Applying Sequential Streams to SearchStreamGang

- We show aggregate operations in the SearchStreamGang’s `processStream()` & `processInput()` methods

We show aggregate operations in the SearchStreamGang’s `processStream()` & `processInput()` methods.

```java
return mPhrasesToFind
    .stream()  
    .map(phrase -> searchForPhrase(phrase, input, title, false))  
    .filter(not(SearchResults::isEmpty))  
    .collect(toList());
```

See `SearchStreamGang/src/main/java/livelessons/streamgangs/SearchWithSequentialStreams.java`
Applying Sequential Streams to SearchStreamGang

- We show aggregate operations in the SearchStreamGang’s processStream() & processInput() methods

```java
getInput()
    .stream()
    .map(this::processInput)
    .collect(toList());

return mPhrasesToFind
    .stream()
    .map(phrase -> searchForPhrase(phrase, input, title, false))
    .filter(not(SearchResults::isEmpty))
    .collect(toList());
```

i.e., the map(), filter(), & collect() aggregate operations
Applying Sequential Streams to SearchStreamGang

- We show aggregate operations in the SearchStreamGang’s processStream() & processInput() methods

  - **processStream()**
    - Uses a sequential stream to search a list of input strings in one thread

  ```java
  stream()
  .map(this::processInput)
  .collect(toList())
  ```

  Each input string contains a work of Shakespeare (e.g., Hamlet, MacBeth, etc.)
Applying Sequential Streams to SearchStreamGang

- We show aggregate operations in the SearchStreamGang’s processStream() & processInput() methods

  - **processStream()**
    - Uses a sequential stream to search a list of input strings in one thread

```
Applying Sequential Streams to SearchStreamGang

We show aggregate operations in the SearchStreamGang’s processStream() & processInput() methods

- **processStream()**
  - Uses a sequential stream to search a list of input strings in one thread

Returns a list of lists of SearchResults
```
Applying Sequential Streams to SearchStreamGang

• We show aggregate operations in the SearchStreamGang’s processStream() & processInput() methods

• processStream()

• processInput()

• Uses a sequential stream to search a given input string & locate all the occurrences of phases in one thread
Applying Sequential Streams to SearchStreamGang

- We show aggregate operations in the SearchStreamGang’s `processStream()` & `processInput()` methods
  - `processStream()`
  - `processInput()`
  - Uses a sequential stream to search a given input string & locate all the occurrences of phases in one thread

*Returns a list of SearchResults*
End of Java Sequential SearchStreamGang Example: Introduction