Visualizing the Java SearchWith ParallelStreams Hook Methods

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

- Visualize how Java parallel streams are applied in SearchWithParallelStreams

Starting SearchStreamGangTest
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

Visualizing the `processStream()` Method
Input Strings to Search

• `processStream()` searches a list of input strings in parallel

Visualizing the `processStream()` Method

Input a list of input strings

List
<String>

parallelStream()

Each input string contains a work of Shakespeare (e.g., Hamlet, MacBeth, etc.)
Visualizing the `processStream()` Method

- `processStream()` searches a list of input strings in parallel.

```
List <String>
```

Convert a collection to a parallel stream, i.e., split into chunks of input strings.
• `processStream()` searches a list of input strings in parallel

**Visualizing the `processStream()` Method**

List `<String>`

- Each item is processed in parallel on separate threads/cores

Stream `<String>`

- Represents chunks of input strings

Output a stream of input strings

Chunks of input strings are processed in parallel on separate threads/cores
• `processStream()` searches a list of input strings in parallel

Input a stream of input strings

List
`<String>`

Stream
`<String>`

A pool of worker threads

Visualizing the `processStream()` Method

Input Strings to Search

`parallelStream()`

`map(this::processInput)`
Visualizing the `processStream()` Method

- `processStream()` searches a list of input strings in parallel.

Call `processInput()` to search for phrases in a given input string in parallel.
• `processStream()` searches a list of input strings in parallel

`Output a stream of lists of search results`

List `<String>`

Stream `<String>`

Stream `<List <SearchResults>>`

Input Strings to Search

`parallelStream()`

`map(this::processInput)`

`A pool of worker threads`

Some lists of search results may be empty if no phrases matched an input string
• `processStream()` searches a list of input strings in parallel

Input a stream of lists of search results

List `<String>`

Stream `<String>`

Stream `<List<SearchResults>>`

Input Strings to Search

parallelStream()

map(this::processInput)

collect(toList())

A pool of worker threads
Visualizing the `processStream()` Method

- `processStream()` searches a list of input strings in parallel.

```
List<String>…
```

```
Stream<String>…
```

```
Stream<List<SearchResults>>…
```

Trigger intermediate operation processing to run on multiple worker threads & cores.
Visualizing the `processStream()` Method

- `processStream()` searches a list of input strings in parallel

```
List<String> ...
Stream<String>
Stream<List<SearchResults>>
List<List<SearchResults>>
```

Return a list of lists of search results based on "encounter order"
Visualizing the processInput() Method
processInput() finds phrases in an input string in parallel

Input a list of phrases to find

List
<String>

parallelStream()
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

Convert a collection to a parallel stream, i.e., split into chunks of phrases
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

**Output a stream of phrases to find**

List `<String>`

Stream `<String>`

`parallelStream()`

Search Phrases

Different chunks of phrases are processed in parallel on multiple worker threads & cores
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

**Input a stream of phrases to find**

```
List <String>
Stream <String>
```

```
parallelStream()
map(phrase -> searchForPhrase(...))
```

A pool of worker threads
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

```
List <String> ...
Stream <String> ...
```

```
perform parallel search for phrases in a given input string
```
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

Output a stream of search results

List `<String>`

Stream `<String>`

Stream `<SearchResults>`

Parallel Stream

Search Phrases

Parallel Stream

Map (phrase -> searchForPhrase(…))
• processInput() finds phrases in an input string in parallel

Input a stream of search results

List <String>

Search Phrases

parallelStream()

map(phrase -> searchForPhrase(…))

filter(not(SearchResults::isEmpty))

A pool of worker threads
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel.

List `<String>`

Stream `<String>`

Stream `<SearchResults>`

Remove empty search results from substreams in parallel
• `processInput()` finds phrases in an input string in parallel.

*Output a stream of non-empty search results*

The stream of search results may be empty if no phrases matched an input string.
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

**Input a stream of non-empty search results**

List\n<List<String>>

Stream\n<Stream<String>>

Stream\n<Stream<SearchResults>>

Stream\n<Stream<SearchResults>>

A pool of worker threads

Search Phrases

- `parallelStream()`
- `map(phrase -> searchForPhrase(...))`
- `filter(not(SearchResults::isEmpty))`
- `collect(toList())`
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

```
List <String> ...
Stream <String> ...
Stream <SearchResults> ...
Stream <SearchResults> ...
```

Trigger intermediate operation processing to run on multiple threads/cores
**Visualizing the processInput() Method**

- `processInput()` finds phrases in an input string in parallel.

```java
List<String> inputList = ...;
Stream<String> stream = inputList.stream();
Stream<SearchResults> resultsStream = stream.map(phrase -> searchForPhrase(phrase));
List<SearchResults> resultsList = resultsStream.filter(not(results -> results.isEmpty())).collect(toList());
```

Return a list of search results in the originating thread based on “encounter order”.
Visualizing the `processInput()` Method

- `processInput()` finds phrases in an input string in parallel

```java
List<String>…
Stream<String>…
Stream<SearchResults>…
Stream<SearchResults>…
List<SearchResults>…
```

This list will be empty if none of the phrases matched the input string
Visualizing the `processInput()` Method

- Note that the actual processing of parallel streams differs from this visualization..

See [developer.ibm.com/articles/j-java-streams-3-brian-goetz](developer.ibm.com/articles/j-java-streams-3-brian-goetz)
End of Visualizing the Java SearchWithParallelStreams Hook Methods