Visualizing the Java Search With Parallel Streams 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

Visualizing the processStream() Method
• processStream() searches a list of input strings in parallel

Visualizing the processStream() Method

Input Strings to Search

Input a list of input strings

List <String>

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

Convert collection to a parallel stream, i.e., substreams with chunks of input strings
Visualizing the `processStream()` Method

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

**Output a stream of input strings**

**Input Strings to Search**

List `<String>`

Stream `<String>`

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.**
Visualizing the `processStream()` Method

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

**Output a stream of lists of search results**

- List `<String>`
- Stream `<String>`
- Stream `<List <SearchResults>>`

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

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

![Diagram of processStream() method]

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

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

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

`List <String>`

`Input a list of phrases to find`

`parallelStream()`

`Search Phrases`
Visualizing the `processInput()` Method

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

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

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

```
List <String>  \rightarrow \n\nOutput a stream of phrases to find
\nParalleled Stream <String>
\nparallelStream()
\nSearch Phrases
```

Different chunks of phrases are processed in parallel on multiple worker threads & cores
Input a stream of phrases to find

processInput() finds phrases in an input string in parallel

List
<Book>
Stream
<Book>

parallelStream()

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

ParallelStream()

Map(phrase -> searchForPhrase(...))

A pool of worker threads

Search Phrases
• processInput() finds phrases in an input string in parallel

Input a stream of search results

List <String>

Stream <String>

Stream <SearchResults>

Visualizing the processInput() Method
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
Visualizing the `processInput()` Method

- `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

Stream <String>
Stream <SearchResults>
Stream <SearchResults>

List <String>

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

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

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

```
List<String>…
```

```
map(phrase -> searchForPhrase(…))
```

```
filter(not(SearchResults::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.

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..

List `<String>`

Stream `<String>`

Stream `<SearchResults>`

Stream `<SearchResults>`

List `<SearchResults>`

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