Visualize 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

• Know how Java parallel streams are applied in SearchWithParallelStreams

See github.com/douglascraigschmidt/LiveLessons/tree/master/SearchStreamGang
Visualizing the processStream() Method
Visualizing the `processStream()` Method

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

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

List `<String>`

Stream `<String>`

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

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

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

```
Input a stream of input strings
```

```
parallelStream()
```

```
map(this::processInput)
```

Visualizing the processStream() Method

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

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

List `<String>`

Stream `<String>`

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>>`

Input Strings to Search

`parallelStream()`

`map(this::processInput)`

Some lists of search results may be empty if no phrases matched an input string.
Visualizing the `processStream()` Method

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

```
Input a stream of lists of search results

List <String> → parallelStream() → Stream <String> → map(this::processInput) → Stream<List <SearchResults>> → collect(toList())
```

Input Strings to Search

```
Input Streams to Search
```

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

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

```
List<String> ...

parallelStream()
... map(this::processInput)
... collect(toList())

Input Strings to Search

A pool of worker threads
```

Trigger intermediate operation processing to run on multiple worker threads & cores
• `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()
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

Output a stream of phrases to find

List `<String>` → `parallelStream()` → Stream `<String>`

A pool of worker threads

Different chunks of phrases are processed in parallel on multiple worker threads & cores
• processInput() finds phrases in an input string in parallel

Visualizing the processInput() Method

Input a stream of phrases to find

List <String>

Search Phrases

Stream <String>

parallelStream()

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

Visualizing the processInput() Method

List <String>

Stream <String>

perform parallel search for phrases in a given input string

Search Phrases

parallelStream()

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

**Search Phrases**

Parallel Stream

Map (phrase -> searchForPhrase(...))
Visualizing the `processInput()` Method

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

```
Input a stream of search results

List <String> → parallelStream()
Stream <String> → map(phrase -> searchForPhrase(...))
Stream <SearchResults> → filter(not(SearchResults::isEmpty))
```

A pool of worker threads

Search Phrases

Hamlet
• `processInput()` finds phrases in an input string in parallel

Visualizing the `processInput()` Method

```
List <String> →
Stream <String> →
Stream <SearchResults>
```

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

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

List `<String>`

Stream `<String>`

Stream `<SearchResults>`

Stream `<SearchResults>`

A pool of worker threads

The stream of search results may be empty if no phrases matched an input string
• processInput() finds phrases in an input string in parallel

Input a stream of non-empty search results

List <String>

Stream <String>

Stream <SearchResults>

Stream <SearchResults>

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> = ... parallel
Stream <String> = ...
Stream <SearchResults> = ...
```

```
Stream <SearchResults>
```

- `parallelStream()`
- `map(phrase -> searchForPhrase(...))`
- `filter(not(SearchResults::isEmpty))`
- `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))
parallelStream()
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.

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

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

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
End of Visualize Java
SearchWithParallelStreams
Hook Methods