Apply Spliterator to the Java Sequential SearchStreamGang Case Study (Part 2)

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 to apply sequential streams to the SearchStreamGang program
- Recognize how a Spliterator is used in SearchWithSequentialStreams

```java
SearchResults searchForPhrase(String phrase, CharSequence input, String title, boolean parallel) {
    return new SearchResults(..., phrase, ..., StreamSupport.
    stream(new PhraseMatchSpliterator
    (input, phrase),
    parallel)
    .collect(toList()));
}
```
Applying Java Spliterator in SearchStreamGang
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

```java
class PhraseMatchSpliterator implements Spliterator<Result> {
    ...

    PhraseMatchSpliterator( CharSequence input, String phrase) {
        String regexPhrase = "\\b" + phrase.trim().replaceAll("\\s+", "\\\b\\\s+\\\b") + "\\b"; ...

        mPattern = Pattern.compile(regexPhrase, Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
        mPhraseMatcher = mPattern.matcher(input);
        mInput = input; mPhrase = phrase;
        mMinSplitSize = input.length() / 2;
    }
    ...
}

See SearchStreamGang/src/main/java/livelessons/utils/PhraseMatchSpliterator.java
```
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string

class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    PhraseMatchSpliterator(CharSequence input, String phrase) {
        String regexPhrase = "\b" + phrase.trim().replaceAll("\s+", "\\\b\\\s+\\\b") + "\b"; ...

        mPattern = Pattern.compile(regexPhrase, Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
        mPhraseMatcher = mPattern.matcher(input);
        mInput = input; mPhrase = phrase;
        mMinSplitSize = input.length() / 2;
    }
    ...

See docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string

class PhraseMatchSpliterator implements Spliterator<Result> {
  ...
  PhraseMatchSpliterator(CharSequence input, String phrase) {
    String regexPhrase = "\b" + phrase.trim().replaceAll("\s+", "\b\s+\b") + "\b"; ...
    mPattern = Pattern.compile(regexPhrase,
      Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
    mPhraseMatcher = mPattern.matcher(input);
    mInput = input; mPhrase = phrase;
    mMinSplitSize = input.length() / 2;
  }
  ...

See docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    PhraseMatchSpliterator(CharSequence input, String phrase) {
        String regexPhrase = "\\b" + phrase.trim().replaceAll("\\s+", "\\b\\b\\s+\\b") + "\\b"; ...

        mPattern = Pattern.compile(regexPhrase,
                                    Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
        mPhraseMatcher = mPattern.matcher(input);
        mInput = input; mPhrase = phrase;
        mMinSplitSize = input.length() / 2;
    } ...

    A matcher is created to search the input for the regex pattern.

See docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

```java
class PhraseMatchSpliterator implements Spliterator<SearchResults> {
    ...
    PhraseMatchSpliterator(CharSequence input, String phrase) {
        String regexPhrase = "\\b" + phrase.trim().replaceAll("\\s+", "\\\\b\\\\s+\\\\b") + "\\b"; ...

        mPattern = Pattern.compile(regexPhrase, Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
        mPhraseMatcher = mPattern.matcher(input);
        mInput = input; mPhrase = phrase;
        mMinSplitSize = input.length() / 2;
    }
    ...
}
```

Set key fields with params
PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    PhraseMatchSpliterator(CharSequence input, String phrase) {
        String regexPhrase = "\\b" + phrase.trim().replaceAll("\\s+", "\\\\b\\\\s+\\\\b") + "\\b"; ...

        mPattern = Pattern.compile(regexPhrase, Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
        mPhraseMatcher = mPattern.matcher(input);
        mInput = input; mPhrase = phrase;
        mMinSplitSize = input.length() / 2;
    }
    ...

    Define the min split size

    The minimum split size is used by the parallel streams version of this program
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

```java
class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    boolean tryAdvance(Consumer<? super Result> action) {
        if (!mPhraseMatcher.find())
            return false;
        else {
            action.accept(new Result(mOffset + mPhraseMatcher.start()));
            return true;
        }
    }
    ...
}
```

Called by the Java streams framework to attempt to advance the spliterator by one phrase match.

See docs.oracle.com/javase/8/docs/api/java/util/Spliterator.html#tryAdvance
• PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string

```java
class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    boolean tryAdvance(Consumer<? super Result> action) {
        if (!mPhraseMatcher.find())
            return false;
        else {
            action.accept(new Result(mOffset + mPhraseMatcher.start()));
            return true;
        }
    }
    ...
}
```

Passes the result (if any) back “by reference” to the streams framework

See [docs.oracle.com/javase/8/docs/api/java/util/function/Consumer.html](http://docs.oracle.com/javase/8/docs/api/java/util/function/Consumer.html)
**Applying Java Spliterator in SearchStreamGang**

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

```java
class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    boolean tryAdvance(Consumer<? super Result> action) {
        if (!mPhraseMatcher.find())
            return false;
        else {
            action.accept(new Result
                        (mOffset + mPhraseMatcher.start()));
            return true;
        }
    }
    ...
```

See [docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html#find](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html#find)
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string

class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    boolean tryAdvance(Consumer<? super Result> action) {
        if (!mPhraseMatcher.find())
            return false;
        else {
            action.accept(new Result(
                mOffset + mPhraseMatcher.start()));
            return true;
        }
    }
    ...

Inform the streams framework to cease calling tryAdvance() if there's no match

See docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html#find
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

```java
class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    boolean tryAdvance(Consumer<? super Result> action) {
        if (!mPhraseMatcher.find())
            return false;
        else {
            action.accept(new Result(mOffset + mPhraseMatcher.start()));
            return true;
        }
    }
    ...
}
```

Stores the index in the input string where the match occurred, which is returned to the streams framework.

See [docs.oracle.com/javase/8/docs/api/java/util/function/Consumer.html#accept](http://docs.oracle.com/javase/8/docs/api/java/util/function/Consumer.html#accept)
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string

class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    boolean tryAdvance(Consumer<? super Result> action) {
        if (!mPhraseMatcher.find())
            return false;
        else {
            action.accept(new Result
                    (mOffset + mPhraseMatcher.start()));
            return true;
        }
    }
    ...
    
Inform the streams framework to continue calling tryAdvance()
Applying Java Spliterator in SearchStreamGang

- PhraseMatchSpliterator uses Java regex to create a stream of SearchResults Result objects that match the # of times a phrase appears in an input string.

class PhraseMatchSpliterator implements Spliterator<Result> {
    ...
    public Spliterator<SearchResults.Result> trySplit() {
        ...
    }
    ...
}

We will analyze trySplit() when we discuss SearchWith ParallelSpliterator (it’s not used for the sequential version)

See docs.oracle.com/javase/8/docs/api/java/util/Spliterator.html#trySplit
End of Apply Spliterator to the Java Sequential Search StreamGang Case Study (Part 2)