Apply Spliterator to the Java Sequential SearchStreamGang Case Study (Part 1)

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

- SearchStreamGang uses PhraseMatchSpliterator that works for both sequential & parallel streams

• SearchStreamGang uses PhraseMatchSpliterator that works for both sequential & parallel streams
• We focus on the sequential portions now
Applying Java Spliterator in SearchStreamGang

- SearchStreamGang uses PhraseMatchSpliterator that works for both sequential & parallel streams
- We focus on the sequential portions now
- We’ll cover the parallel portions later

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

See lessons on the "Java SearchWithParallelSpliterator Example"
searchForPhrase() uses PhraseMatchSpliterator to find all phrases in input & return SearchResults

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

- `searchForPhrase()` uses `PhraseMatchSpliterator` to find all phrases in input & return `SearchResults`

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

`StreamSupport.stream()` creates a sequential or parallel stream via `PhraseMatchSpliterator`

See [docs.oracle.com/javase/8/docs/api/java/util/stream/StreamSupport.html#stream](http://docs.oracle.com/javase/8/docs/api/java/util/stream/StreamSupport.html#stream)
searchForPhrase() uses PhraseMatchSpliterator to find all phrases in input & return SearchResults

SearchResults searchForPhrase(String phrase, CharSequence input, String title, boolean parallel) {
    return new SearchResults(..., phrase, ..., StreamSupport.stream(new PhraseMatchSpliterator(input, phrase), parallel)
        .collect(toList()));
}

For SearchWithSequentialStreams “parallel” is false, so we’ll use a sequential spliterator

See docs.oracle.com/javase/8/docs/api/java/util/stream/StreamSupport.html#stream
searchForPhrase() uses PhraseMatchSpliterator to find all phrases in input & return SearchResults

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

Convert the stream into a list of Result objects
Applying Java Spliterator in SearchStreamGang

- Here's the context of PhraseMatchSpliterator for processInput() in SearchWithSequentialStreams

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

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

```
stream()
map(phrase -> searchForPhrase(…))
filter(not(SearchResults::isEmpty))
collect(toList())
```
Applying Java Spliterator in SearchStreamGang

- Here’s the context of PhraseMatchSpliterator for processInput() in SearchWithSequentialStreams

“...
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” matches at index [54739]
Applying Java Spliterator in SearchStreamGang

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

```java
class PhraseMatchSpliterator implements Spliterator<Result> {
    private CharSequence mInput;
    private final String mPhrase;
    private final Pattern mPattern;
    private Matcher mPhraseMatcher;
    private final int mMinSplitSize;
    private int mOffset;
    ...
}
```

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> {
    private CharSequence mInput;
    private final String mPhrase;
    private final Pattern mPattern;
    private Matcher mPhraseMatcher;
    private final int mMinSplitSize;
    private int mOffset;
    ...

    Spliterator is an interface that defines eight methods, including tryAdvance() & trySplit()
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> {
    private final String mPhrase;
    private final Pattern mPattern;
    private Matcher mPhraseMatcher;
    private final int mMinSplitSize;
```

Some fields are updated in the trySplit() method, which is why they aren’t final
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> {
    private CharSequence mInput;
    private final String mPhrase;
    private final Pattern mPattern;
    private Matcher mPhraseMatcher;
    private final int mMinSplitSize;
    private int mOffset;
    ...
```

Contains a single work of Shakespeare
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<
  private final String mPhrase;
  private final Pattern mPattern;
  private Matcher mPhraseMatcher;
  private final int mMinSplitSize;
  private int mOffset;
  ...
```
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> {
    private CharSequence mInput;
    private final String mPhrase;
    private final Pattern mPattern;
    private Matcher mPhraseMatcher;
    private final int mMinSplitSize;
    private int mOffset;
    ...
}
```

See docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html

Contains the regular expression representation of the phrase
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> {
    private CharSequence mInput;

    private final String mPhrase;

    private final Pattern mPattern;

    private Matcher mPhraseMatcher;

    private final int mMinSplitSize;

    private int mOffset;

    ...

    Contains a matcher that searches for the phrase in the input

    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<Result> {
    private CharSequence mInput;

    private final String mPhrase;

    private final Pattern mPattern;

    private Matcher mPhraseMatcher;

    private final int mMinSplitSize;

    private int mOffset;

    ...
}
```

- `Dictates the minimum size to perform a split`
• 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> {
    private CharSequence mInput;

    private final String mPhrase;

    private final Pattern mPattern;

    private Matcher mPhraseMatcher;

    private final int mMinSplitSize;

    private int mOffset;

    ...

    Track the offset needed to return the index into the original string

    This field is used by the parallel streams spliterator
End of Apply Spliterator to the Java Sequential Search StreamGang Case Study (Part 1)