

Java SearchWithParallelSpliterator Example: PhraseMatchSpliterator Constructor & tryAdvance()

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

- Be aware of how a parallel spliterator can improve parallel stream performance
- Know the intent of—& fields in—the PhraseMatchSpliterator
- Recognize the PhraseMatchSpliterator constructor & tryAdvance() method implementation

```
class PhraseMatchSpliterator
    implements Spliterator<Result> {
    ...
    PhraseMatchSpliterator (CharSequence input,
                           String phrase) { ... }

    boolean tryAdvance (Consumer<? super Result> action) { ... }
    ...
}
```

These methods are identical w/the SearchWithSequentialStreams class

See "Java Sequential SearchStreamGang Example: Applying Spliterator"

Analysis of PhaseMatch Spliterator Constructor

Analysis of PhraseMatchSpliterator Constructor

- 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+", "\\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](https://github.com/Netflix-Serliano/search-stream-gang/blob/master/src/main/java/livelessons/utils/PhraseMatchSpliterator.java)

Analysis of PhraseMatchSpliterator Constructor

- 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+", "\\s+\\b\\s+\\b")  
                        + "\\b"; ...
```

*One work of Shakespeare & a
phrase to search for in this work*

```
    mPattern = Pattern.compile(regexPhrase,  
                               Pattern.CASE_INSENSITIVE | Pattern.DOTALL);
```

```
    mPhraseMatcher = mPattern.matcher(input);
```

```
    mInput = input; mPhrase = phrase;
```

```
    mMinSplitSize = input.length() / 2;
```

```
} ...
```

Analysis of PhraseMatchSpliterator Constructor

- 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+", "\\s+\\b\\s+\\b")  
                                + "\\b"; ...  
    }  
}
```

Create a regex that matches phrases

```
    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

Analysis of PhraseMatchSplitter Constructor

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

```
class PhraseMatchSplitter implements Splitter<Result> {
```

```
...
```

```
PhraseMatchSplitter(CharSequence input, String phrase) {
```

```
    String regexPhrase = "\\b" + phrase.trim().replaceAll  
                        ("\\s+", "\\s+\\b\\s+\\b")  
                        + "\\b"; ...
```

*A regex is compiled into a pattern
that matches a phrase across lines*

```
    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

Analysis of PhraseMatchSpliterator Constructor

- 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+", "\\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;  
    } ...  
}
```

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

Analysis of PhraseMatchSpliterator Constructor

- 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+", "\\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

Analysis of PhraseMatchSpliterator Constructor

- 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+", "\\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

This field is used by the trySplit() method for a parallel spliterator

Analysis of PhraseMatcherSpliterator tryAdvance() Method

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

```
class PhraseMatcherSpliterator 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 Java streams framework,
which attempts to advance the
spliterator by one matching phrase*

See docs.oracle.com/javase/8/docs/api/java/util/Spliterator.html#tryAdvance

Analysis of PhraseMatcherSpliterator tryAdvance() Method

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

```
class PhraseMatcherSpliterator 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

Analysis of PhraseMatcherSpliterator tryAdvance() Method

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

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

*Check if any remaining phrases
in the input match the regex*

Analysis of PhraseMatcherSpliterator tryAdvance() Method

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

```
class PhraseMatcherSpliterator 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

Analysis of PhraseMatcherSpliterator tryAdvance() Method

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

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

If there is a match, then accept() keeps track of which index in the input string the match occurred

Analysis of PhraseMatcherSpliterator tryAdvance() Method

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

```
class PhraseMatcherSpliterator 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()*

End of Java SearchWith
ParallelSpliterator Example:
PhraseMatchSpliterator
Constructor & tryAdvance()