Java SearchWithParallelSpliterator

Example: PhraseMatchSpliterator & Fields

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—and fields in—the PhraseMatchSpliterator

```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 = 0; ...
}
```

These fields are identical with the SearchWithSequentialStreams class

See “Java Sequential SearchStreamGang Example: Applying Spliterator”
Overview of PhaseMatchSpliterator
Overview of PhraseMatchSpliterator

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

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

Overview of PhraseMatchSpliterator

- SearchStreamGang uses PhraseMatchSpliterator that works for both sequential & parallel streams
- We focused on the sequential portions earlier
  - & will review them again now briefly

See "Java Sequential SearchStreamGang Example: Applying Spliterator"
Overview of PhraseMatchSpliterator

• SearchStreamGang uses PhraseMatchSpliterator that works for both sequential & parallel streams
  • We focused on the sequential portions earlier
  • We’ll cover the parallel portions next

The goal is to further optimize the performance of the parallel streams solution
Overview of PhraseMatchSpliterator

- Here’s the input/output of PhraseMatchSpliterator for SearchWithParallelSpliterator

List `<String>`

Stream `<String>`

Stream `<SearchResults>`

Stream `<SearchResults>`

List `<SearchResults>`

Search Phrases

parallelStream()

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

filter(not(SearchResults::isEmpty))

collect(toList())

Here’s the input/output of PhraseMatchSpliterator for SearchWithParallelSpliterator

- Overview of PhraseMatchSpliterator

parallelStream()
Here’s the input/output of PhraseMatchSpliterator for SearchWithParallelSpliterator

“… 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. Your noble son is mad. Mad call I it; for, to define true madness, What is't but to be nothing else but mad? But let that go.…“

Overview of PhraseMatchSpliterator

This spliterator splits the input into multiple chunks & searches them in parallel.
Here’s the input/output of PhraseMatchSpliterator for SearchWithParallelSpliterator

"Brevity is the soul of wit\[54739\]

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. Your noble son is mad.
Mad call I it; for, to define true madness,
What is't but to be nothing else but mad?
But let that go...."

When the split occurs efficiently/evenly the speedups can be substantial!
Overview of PhraseMatchSpliterator

• Here’s the input/output of PhraseMatchSpliterator for SearchWithParallelSpliterator

“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. Your noble son is mad. Mad call I it; for, to define true madness,
What is't but to be nothing else but mad?
But let that go....”

However, the spliterator must be careful not to split input across phrases...
Analysis of PhaseMatch Spliterator Fields
Analysis of PhraseMatchSpliterator Fields

- 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 = 0;
    ...
}
```

Splitter is an interface that defines eight methods, including `tryAdvance()` & `trySplit()`.

See SearchStreamGang/src/main/java/livelessons/utils/PhraseMatchSpliterator.java
Analysis of PhraseMatchSpliterator Fields

- 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 = 0;
    ...
}
```

These fields implement PhraseMatchSpliterator for both sequential & parallel use-cases.

Some fields are updated in the `trySplit()` method, which is why they aren’t final.
Analysis of PhraseMatchSpliterator Fields

- 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 = 0;

    ...

    Contains a single work of Shakespeare
Analysis of PhraseMatchSpliterator Fields

- 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 = 0;
    ...

    Contains the phrase to search for in the work.
Analysis of PhraseMatchSpliterator Fields

- 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 = 0;
    ...
}
```

Contains the regular expression representation of the phrase

See docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html
Analysis of PhraseMatchSpliterator Fields

- 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 = 0;
    ...
```

- Contains a matcher that searches for the phrase in the input

See [docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html](https://docs.oracle.com/javase/8/docs/api/java/util/regex/Matcher.html)
Analysis of PhraseMatchSpliterator Fields

- 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 = 0;
    ...
```

*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.

```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 = 0;
    ...
```

**Analysis of PhraseMatchSpliterator Fields**

- **The offset needed to return the appropriate index into the original input string**

- **This value is reset by each spliterator to account for different chunks**
End of Java SearchWith ParallelSpliterator Example: PhraseMatchSpliterator & Fields