This course has already ended.
The latest instance of the course can be found at: Concurrent Programming: 2023

« 1 Futures and Promises-- High-level mechanisms for asynchronous prog... Course materials 3 Feedback »

CS-E4110 / Round 5 - Composable structures / 2 Parallel Collections -- High-level mechanisms for data parallelism.

Assignment description | My submissions (1/10)

Parallel Collections -- High-level mechanisms for data parallelism.

Parallel Strings

In the earlier modules, we have seen ways of composing multiple threads of computation into safe concurrent programs. In this exercise, we will briefly focus on achieving better performance while requiring minimal changes in the organization of legacy programs. One specific way of achieving better performance is data parallelism which relies on doing computation in parallel on different portions of a data collection.

Code

Download the assignment template here

Task

In this task, we implement a custom parallel collection called ParString which is a counter part of Java's String class that supports parallel operations:

```
class ParString(val str: String) extends immutable.ParSeq[Char] {
    def apply(i: Int)
    def length
    def seq
    def splitter
    override def newCombiner
}
```

A parallel string is a sequence of characters, so it is only logical to extend the ParSeq parallel collection trait with the Char type argument. When we extend a parallel collection, we need to implement its apply, length, splitter, and seq methods. Follow the Hint links below for more detail.

Hint

Consult chapter 5 of the book: Learning concurrent programming in Scala by Aleksandar Prokopec and/or take a look at this tutorial.

ParString.scala

Choose File No file chosen

Submit

Earned points
25 / 25

Exercise info

Assignment category
Programming exercises

Your submissions
1 / 10

Deadline
Wednesday, 8 December 2021, 14:00

Late submission deadline
Wednesday, 15 December 2021, 14:00 (-30%)

Total number of submitters
40

« 1 Futures and Promises-- High-level mechanisms for asynchronous prog... Course materials 3 Feedback »