CSE 332: Data Structures and Parallelism

QuickCheck: More Parallelism Solutions

0. I like I

Write pseudocode for a parallelized algorithm using the `ForkJoin` framework that takes an input array containing strings and returns a new array containing only the strings that started with the letter 'i".

For example, if the input array is `["ib", "bb", "ic", "zc", "yc", "ic", "iy", "ic]`, the output array should be `["ib", "ic", "ic", "iy", "ic"]`.

Find the work and span of your algorithm.

Solution:

Our algorithm will:

(a) Perform a parallel `map` over the input array to construct a bitmap. If the string at `input[x]` starts with 'i', then the bit at `bitmap[x]` should be '1'.

Using our initial example array up above, the corresponding bitmap would be `[1, 0, 1, 0, 0, 1, 1, 1]`.

This operation will have $\mathcal{O}(n)$ work and $\mathcal{O}(\lg(n))$ span.

(b) Perform a parallel `scan` over the bitmap, creating a cumulative sum array.

For example, the bitmap up above would result in `[1, 1, 2, 2, 2, 3, 4, 5]`

This operation will have $\mathcal{O}(n)$ work and $\mathcal{O}(\lg(n))$ span.

(c) Perform a parallel `pack` over the cumulative sum array to extract the relevant items from the input array.

This should result in an output of `["ib", "ic", "ic", "iy", "ic"]`, which is what we wanted.

This operation will also have $\mathcal{O}(n)$ work and $\mathcal{O}(\lg(n))$ span.

Since each operation is performed in sequence, and each has a work of $\mathcal{O}(n)$ and a span of $\mathcal{O}(\lg(n))$, we know the combined work and span will be $\mathcal{O}(n)$ and $\mathcal{O}(\lg(n))$ respectively.