Parallelizing non-associative sequential reductions

By Federico Pizzuti
Supervisor: Cristophe Dubach

Associative reductions are parallelizable

Summation of Numbers: $\sum_{0}^{n} x_n = x_1 + \ldots + x_n$

```
reduce(+,[1,2,3,4,5,6,7,8])
```

The operator + is **associative**, so this reduction is **parallelizable**

What about non-associative reductions?

Polynomial evaluation: $\sum_{0}^{n} x_n \cdot k^n$

Define operator: $a \odot b = k \cdot a + b$

```
reduce(⊙,[1,2,3,4,5,6,7,8])
```

The operator $\odot$ is **not associative**, so this reduction must be executed **sequentially**

How to parallelize non-associative reductions?

Key insight: Matrix multiplication ($\times$) is associative*

We rewrite the reduction using $\odot$ in terms of a reduction that uses $\times$

The derived reduction is then **parallelizable**

*This works for all operators expressed in terms of semiring operations, not just $+$ and $\cdot$

Rewrite reduction operator as matrix multiplication

Rearrange data as matrices

`[1,2,3,4,5,6,7,8]`

$$\begin{bmatrix} k & 1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} k & 2 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} k & 3 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} k & 4 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} k & 5 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} k & 6 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} k & 7 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} k & 8 \\ 0 & 1 \end{bmatrix}$$

$$\texttt{reduce}(\odot, [1,2,3,4,5,6,7,8]) = x$$

$$\texttt{reduce}\left(\odot, \begin{bmatrix} k & 1 \\ 0 & 1 \end{bmatrix}, \ldots, \begin{bmatrix} k & 8 \\ 0 & 1 \end{bmatrix}\right) = \begin{bmatrix} a & x \\ 0 & 1 \end{bmatrix}$$