Prefix scan

Gauthier Castro, David Beley, Hoang Bich Ngoc

5/12/2018

1. Prefix scan
2. Applications
3. Parallelization methods
4. Benchmark

Section 1

Prefix scan

Algorithm

Input : $x_0, x_1, \ldots, x_{n-1}$

Output : $s_0, s_1, \ldots, s_{n-1}$

Operator: $\otimes$

Prefix scan general formulation:

$$\begin{gathered} s_0 = x_0, \\ s_1 = x_0 \otimes x_1, \\ s_2 = x_0 \otimes x_1 \otimes x_2, \\ \ldots, \\ s_{n-1} = x_0 \otimes x_1 \otimes \ldots \otimes x_{n-1} \end{gathered}$$

Simple example with cumsum

```r
x <- c(12, 5, 13)
cumsum(x)
```

```
## [1] 12 17 30
```

Inclusive vs. Exclusive scan

In inclusive scan, x_i is included in s_i. In exclusive prefix scan, x_i is not included.

Examples with cumsum

```
x <- c(12, 5, 13)
cumsum_inclusive(x)
```

```
## [1] 12 17 30
```

```
x <- c(12, 5, 13)
cumsum_exclusive(x)
```

```
## [1]  0 12 17
```

Section 2

Applications

Polynomial calculation

$$P = 7 + 5x - 3x^2 - 6x^3 + 3x^4$$

Exclusive prefix scan with product operator

$$\begin{array}{ccccc} x & x & x & x & x \\ s_0 = x^0 & s_1 = x^1 & s_2 = x^2 & s_3 = x^3 & s_4 = x^4 \end{array}$$

Multiplication of the input vectors and the coefficient vectors

$$\begin{array}{ccccc} x^0 & x^1 & x^2 & x^3 & x^4 \\ * & * & * & * & * \\ 7 & 5 & -3 & -6 & 3 \end{array}$$

Calculation of polynomial

$$x = 7$$

$$P = 7 + 5x - 3x^2 - 6x^3 + 3x^4$$

```r
x
```

```
## [1] 7 7 7 7
```

```r
c(1,cumprod(x))
```

```
## [1]    1    7   49  343 2401
```

```r
coef
```

```
## [1]  7  5 -3 -6  3
```

Calculation of polynomial

```
res
```

```
## [1]     7    35  -147 -2058  7203
```

```
sum(res)
```

```
## [1] 5040
```

Section 3

Parallelization methods

Algorithm: log-based method

```
for i ← 0 to $[\log_2 n]$ - 1 do
    • for j ← 0 to n - 1 do in parallel
        - ***if*** j < 2^i^ ***then***
            - x~j~^i+1^ $\leftarrow$ x~j~^i^
        - ***else***
            - x~j~^i+1^ $\leftarrow$ x~j~^i^ + x^i^~j-2^i^~
```

Illustration

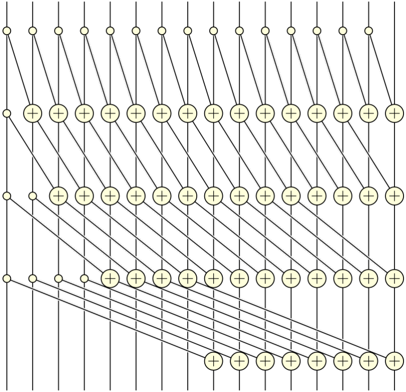


Figure 1: log-based Illustration

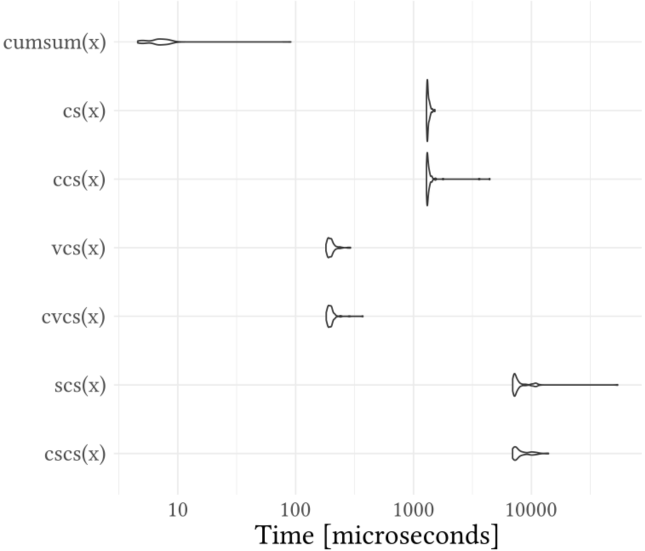
Section 4

Benchmark

Functions implemented

- cs : sequential cumsum
- vcs : vectorized cumsum
- scs : “sapply” cumsum
- pscs : parallel “sapply” cumsum
- fcs : “foreach” cumsum
- pfcs : parallel “foreach” cumsum
- c-cs : compiled cs
- c-scs : compiled scs
- c-vcs : compiled vcs

Benchmark


cumsum(x)
cs(x)
ccs(x)
vcs(x)
cvcs(x)
scs(x)
cscs(x)
10
100
1000
10000
Time [microseconds]