

High-Performance Computing

– Exercises: Distributed Memory Programming with MPI –

Christian Plessl

High-Performance IT Systems Group

Paderborn University

Pacheco Exercise 3.11

- 3.11. Finding **prefix sums** is a generalization of global sum. Rather than simply finding the sum of n values,

$$x_0 + x_1 + \cdots + x_{n-1},$$

the prefix sums are the n partial sums

$$x_0, x_0 + x_1, x_0 + x_1 + x_2, \dots, x_0 + x_1 + \cdots + x_{n-1}.$$

- Devise a serial algorithm for computing the n prefix sums of an array with n elements.
- Parallelize your serial algorithm for a system with n processes, each of which is storing one of the x_i s.
- Suppose $n = 2^k$ for some positive integer k . Can you devise a serial algorithm and a parallelization of the serial algorithm so that the parallel algorithm requires only k communication phases?
- MPI provides a collective communication function, `MPI_Scan`, that can be used to compute prefix sums:

```
int MPI_Scan(
    void*      sendbuf_p  /* in */,
    void*      recvbuf_p  /* out */,
    int        count      /* in */,
    MPI_Datatype datatype  /* in */,
    MPI_Op     op         /* in */,
    MPI_Comm   comm       /* in */);
```

It operates on arrays with `count` elements; both `sendbuf_p` and `recvbuf_p` should refer to blocks of `count` elements of type `datatype`. The `op` argument is the same as `op` for `MPI_Reduce`. Write an MPI program that generates a random array of `count` elements on each MPI process, finds the prefix sums, and prints the results.

Change log

- **1.1.0 (2017-11-07)**
 - removed programming exercises (now on dedicated webpage)
- **1.0.0 (2016-11-24)**
 - initial version of slides