

MPI Parallel Prefix Sum (1A)

•
•

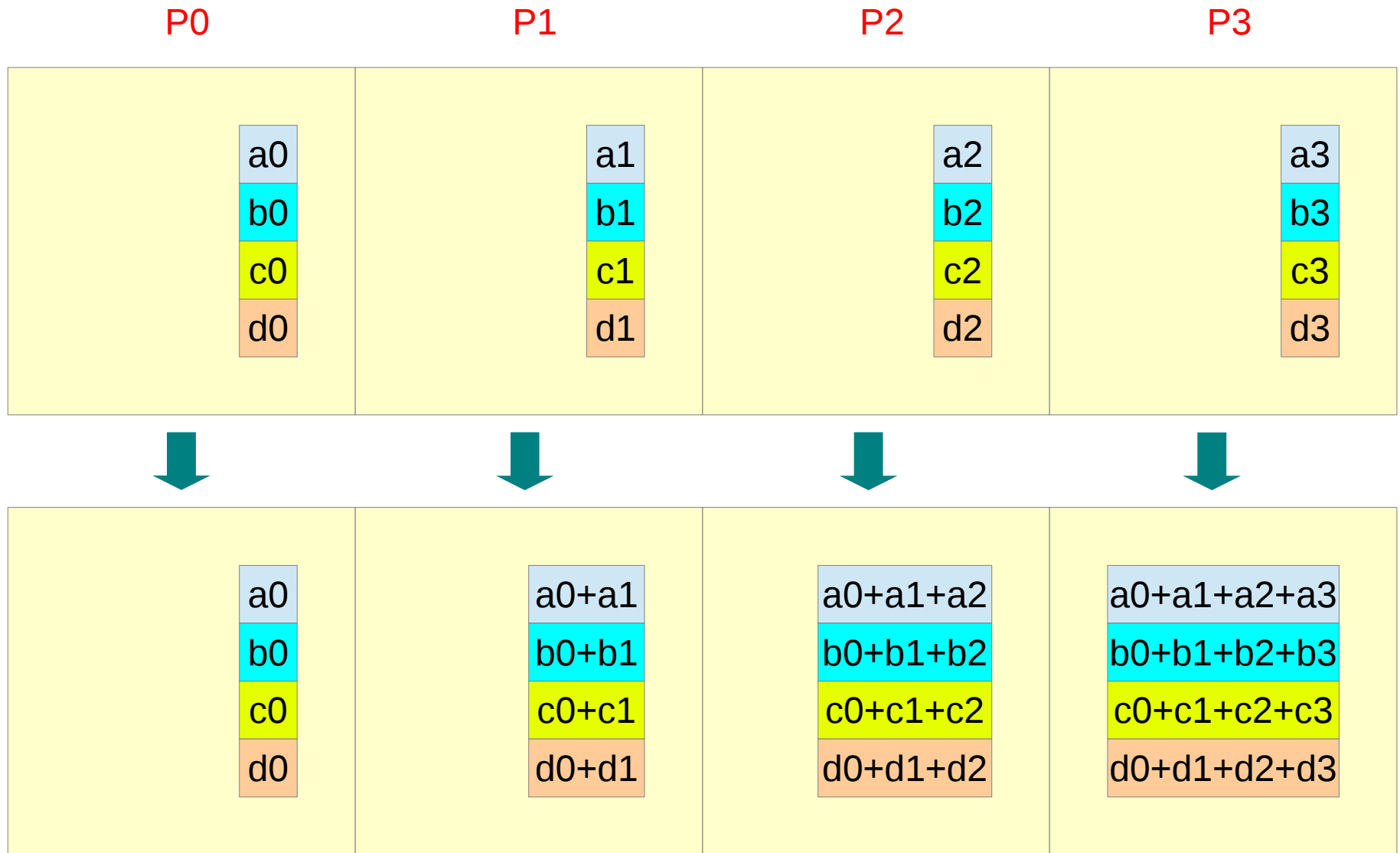
Copyright (c) 2013 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

MPI Parallel Prefix - Scan



Parallel Prefix Sum

```
MPI_Scan( void *sbuf,  
          void *rbuf,  
          int count,  
          MPI_Datatype stype,  
          MPI_Op op,  
          MPI_Comm comm )
```

Parallel Prefix Sum

MPI_OP	Operation	C	Fortran
MPI_MAX	maximum		
MPI_MIN	minimum		
MPI_SUM	sum		
MPI_PROD	product		
MPI_LAND	logical and		
MPI_BAND	bit-wise and		
MPI_LOR	logical or		
MPI BOR	bit-wise or		
MPI_LXOR	logical xor		
MPI_BXOR	bit-wise xor		
MPI_MAXLOC	max val and loc		
MPI_MINLOC	min val and loc		

Parallel Prefix Sum

<http://www.mpi-forum.org/docs/mpi-1.1/mpi-11-html/node85.html>

```
typedef struct {
    double val;
    int log;
} SegScanPair;

/* the user-defined function */
void segScan
( SegScanPair *in, SegScanPair *inout, int *len, MPI_Datatype *dptr ) {

    int i; SegScanPair c;

    for (i=0; i< *len; ++i) {
        if ( in->log == inout->log ) c.val = in->val + inout->val;
        else c.val = inout->val;

        c.log = inout->log;

        *inout = c;
        in++;
        inout++;
    }
}
```

Parallel Prefix Sum

<http://www.mpi-forum.org/docs/mpi-1.1/mpi-11-html/node85.html>

```
int i, base;
    SeqScanPair  a, answer;
    MPI_Op       myOp;
    MPI_Datatype type[2] = {MPI_DOUBLE, MPI_INT};
    MPI_Aint     disp[2];
    int         blocklen[2] = { 1, 1};
    MPI_Datatype sspair;

/* explain to MPI how type SegScanPair is defined */
MPI_Address( a, disp);
MPI_Address( a.log, disp+1);
base = disp[0];

for (i=0; i<2; ++i) disp[i] -= base;

MPI_Type_struct( 2, blocklen, disp, type, &sspair );
MPI_Type_commit( &sspair ); /* create the segmented-scan user-op */

MPI_Op_create( segScan, False, &myOp );
...
MPI_Scan( a, answer, 1, sspair, myOp, root, comm );
```

References

- [1] en.wikipedia.org
- [2] R. v. Engelen, www.cs.fsu.edu/~engelen/courses/HPC/MessagePassing2.pdf