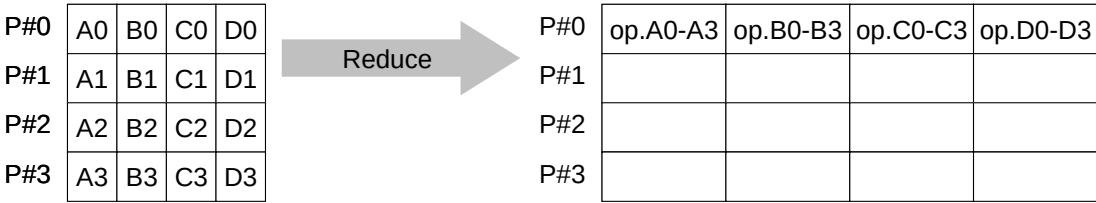


MPI_REDUCE



- Reduces values on all processes to a single value
 - Summation, Product, Max, Min etc.
- **call MPI_REDUCE**
(sendbuf, recvbuf, count, datatype, op, root, comm, ierr)
 - **sendbuf** choice I starting address of send buffer
 - **recvbuf** choice O starting address receive buffer
type is defined by "**datatype**"
 - **count** I I number of elements in send/receive buffer
 - **datatype** I I data type of elements of send/recive buffer
 - FORTTRAN MPI_INTEGER, MPI_REAL, MPI_DOUBLE_PRECISION, MPI_CHARACTER etc.
 - C MPI_INT, MPI_FLOAT, MPI_DOUBLE, MPI_CHAR etc
 - **op** I I reduce operation
 - MPI_MAX, MPI_MIN, MPI_SUM, MPI_PROD, MPI_LAND, MPI_BAND etc
 - Users can define operations by **MPI_OP_CREATE**
 - **root** I I rank of root process
 - **comm** I I communicator
 - **ierr** I O completion code

Send/Receive Buffer (1/3)

A: Scalar

```
call MPI_REDUCE  
(A, recvbuf, 1, datatype, op, root, comm, ierr)
```

```
MPI_Reduce  
(A, recvbuf, 1, datatype, op, root, comm)
```

Send/Receive Buffer (2/3)

A: Array

```
call MPI_REDUCE  
(A, recvbuf, 3, datatype, op, root, comm, ierr)
```

```
MPI_Reduce  
(A, recvbuf, 3, datatype, op, root, comm)
```

- Starting Address of Send Buffer
 - $A(1)$: Fortran, $A[0]$: C
 - 3 (continuous) components of A ($A(1) - A(3)$, $A[0] - A[2]$) are sent

A(:)	1	2	3	4	5	6	7	8	9	10
A[:]	0	1	2	3	4	5	6	7	8	9

Send/Receive Buffer (3/3)

A: Array

```
call MPI_REDUCE
  (A(4), recvbuf, 3, datatype, op, root, comm, ierr)
```

```
MPI_Reduce
  (A[3], recvbuf, 3, datatype, op, root, comm)
```

- Starting Address of Send Buffer
 - $A(4)$: Fortran, $A[3]$: C
 - 3 (continuous) components of A ($A(4) - A(6)$, $A[3] - A[5]$) are sent

A(:)	1	2	3	4	5	6	7	8	9	10
A[:]	0	1	2	3	4	5	6	7	8	9