Parallel Processing

Dr. Guy Tel-Zur

Agenda

- Sending a 2D array in MPI
- Network Performance – Revisited
- Collective commands
- Send/Recv types and summary
- Performance Evaluation of Parallel Programs
- PBS demo
- mpiP
- Parallel Computing environments for this course

Recommended online references

- **Designing and Building Parallel Programs**, by *Ian Foster* http://www.mcs.anl.gov/~itf/dbpp/
- http://www.mhpcc.edu/training/workshop/parallel_intro/MAIN.html
- https://computing.llnl.gov/tutorials/parallel_comp/ by Blaise Barney, Livermore Computing

Another Linux Reference

- People who still don’t feel comfortable with linux please consult this reference:

http://sc.tamu.edu/shortcourses/SC-unix/introToUnix.pdf

About the status in MPI_Recv

- It is possible to Receive messages in non-selective ways:
  - source = MPI_ANY_SOURCE
  - tag = MPI_ANY_TAG
- This is useful for example in a Master-Worker style where the master holds a workpool and it gives a work unit upon each "recv".

- `status.MPI_SOURCE` specifies the rank of the sending process;
- `status.MPI_TAG` specifies the tag of the message received;
- `status.MPI_ERROR` contains an error code.

M-W Demo

- **Embarrassingly Parallel Computation**
  - π calculation by Monte Carlo (Previous lecture)
  - Demo under:

/users/agnon/misc/tel-zur/mpi/pi_monte_carlo

- Execution:

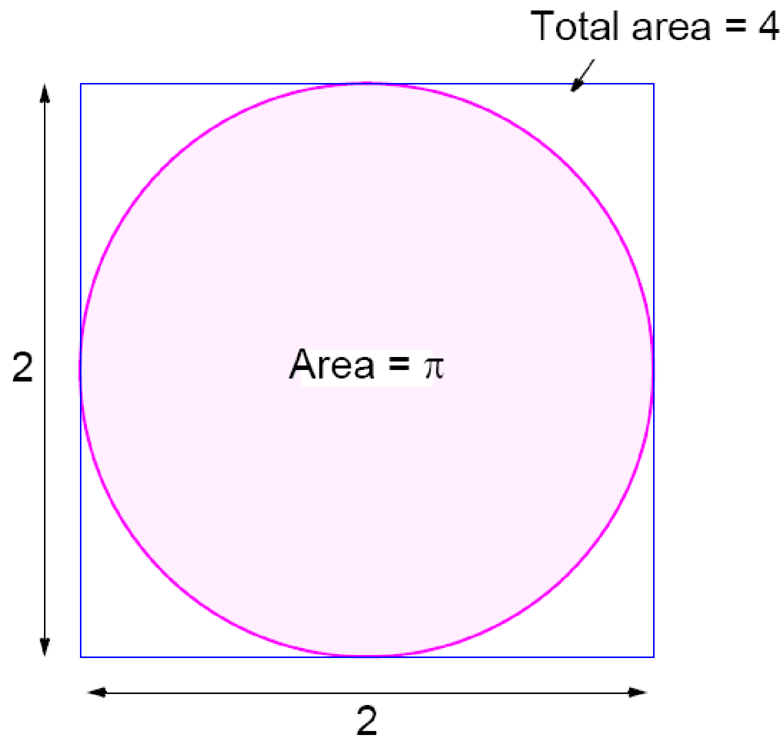
```
mpirun -np 4 ./pi_reduce
```

- The source code – open IDE

Master-Worker (M-W) Demo

- Embarrassingly Parallel Computing paradigm
- Calculation of π

$$\frac{\text{Area of circle}}{\text{Area of square}} = \frac{\pi(1)^2}{2 \times 2} = \frac{\pi}{4}$$



Pseudo Code - Serial Version

```
npoints = 10000
circle_count = 0
do j = 1,npoints
 generate 2 random numbers between 0 and 1
 xcoordinate = random1
 ycoordinate = random2
 if (xcoordinate, ycoordinate) inside circle
then
    circle_count = circle_count + 1
end do
PI = 4.0*circle_count/npoints
```

Pseudo Code - Parallel Version

```
npoints = 10000
circle_count = 0
p = number of tasks
num = npoints/p
find out if I am MASTER or WORKER
do j = 1,num
   generate 2 random numbers between 0 and 1
   xcoordinate = random1
   ycoordinate = random2
   if (xcoordinate, ycoordinate) inside circle then
      circle_count = circle_count + 1
end do
if I am MASTER
   receive from WORKERS their circle_counts compute PI
(use MASTER and WORKER calculations)
else if I am WORKER
   send to MASTER circle_count
endif
```

Must use different seeds

M – W Monte-Carlo calculation of π

eesrv.ee.bgu.ac.il - PuTTY

```
vdwarf20.ee.bgu.ac.il> mpirun -np 4 ./pi_reduce
MPI task ID = 0
MPI task ID = 1
MPI task ID = 2
MPI task ID = 3
   After 5000 throws, average value of pi = 3.14360000
   After 10000 throws, average value of pi = 3.13500000
   After 15000 throws, average value of pi = 3.14506667
   After 20000 throws, average value of pi = 3.14670000
   After 25000 throws, average value of pi = 3.14196000
   After 30000 throws, average value of pi = 3.14516667
   After 35000 throws, average value of pi = 3.14880000
   After 40000 throws, average value of pi = 3.14612500
   After 45000 throws, average value of pi = 3.14673333
   After 50000 throws, average value of pi = 3.14674000
vdwarf20.ee.bgu.ac.il>
```

- Reference to the source code:
http://www.pdc.kth.se/training/Tutor/MPI/Templates/pi/index.html#top
- pi_send.c
- pi_reduce.c
- dboard.c
- make.pi.c

- However, I had to modify the scaling of random numbers – see next slide

0<r<1

- Instead of:

```
cconst = 2 << (31 - 1);
r = (double)random()/cconst;
```

- I had to change the code to:

```
r = ((double)rand() / ((double)
(RAND_MAX) + (double)(1)));
```

Sending a 2D array in MPI between two tasks

- A simple demo – *turn to the demo*
- The source code is enclosed – next slide
- Implementation on Windows Vista using **DeinoMPI** and **DevC++**

```
// Guy Tel-Zur (c) 2009
// This is a demo for PP2010A course
// sending a 2D array - study pointers
#include <mpi.h>
int main (int argc, char **argv) {
    int dim = 3; // array dimension
    int smat[dim][dim];  //send matrix[row]
[col]
    int rmat[dim][dim];  // recv matrix
    int i,j;
    int mytid, nproc;
    int MASTER = 0;
    int WORKER = 1;
    int tag = 99;
    MPI_Status status;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD,
&mytid);
    MPI_Comm_size(MPI_COMM_WORLD,
&nproc);
    printf ("MPI task ID = %d\n", mytid);

    if (nproc != 2) {
              printf("This program needs
exactly 2 processes\n");
              exit (1);
    }
```

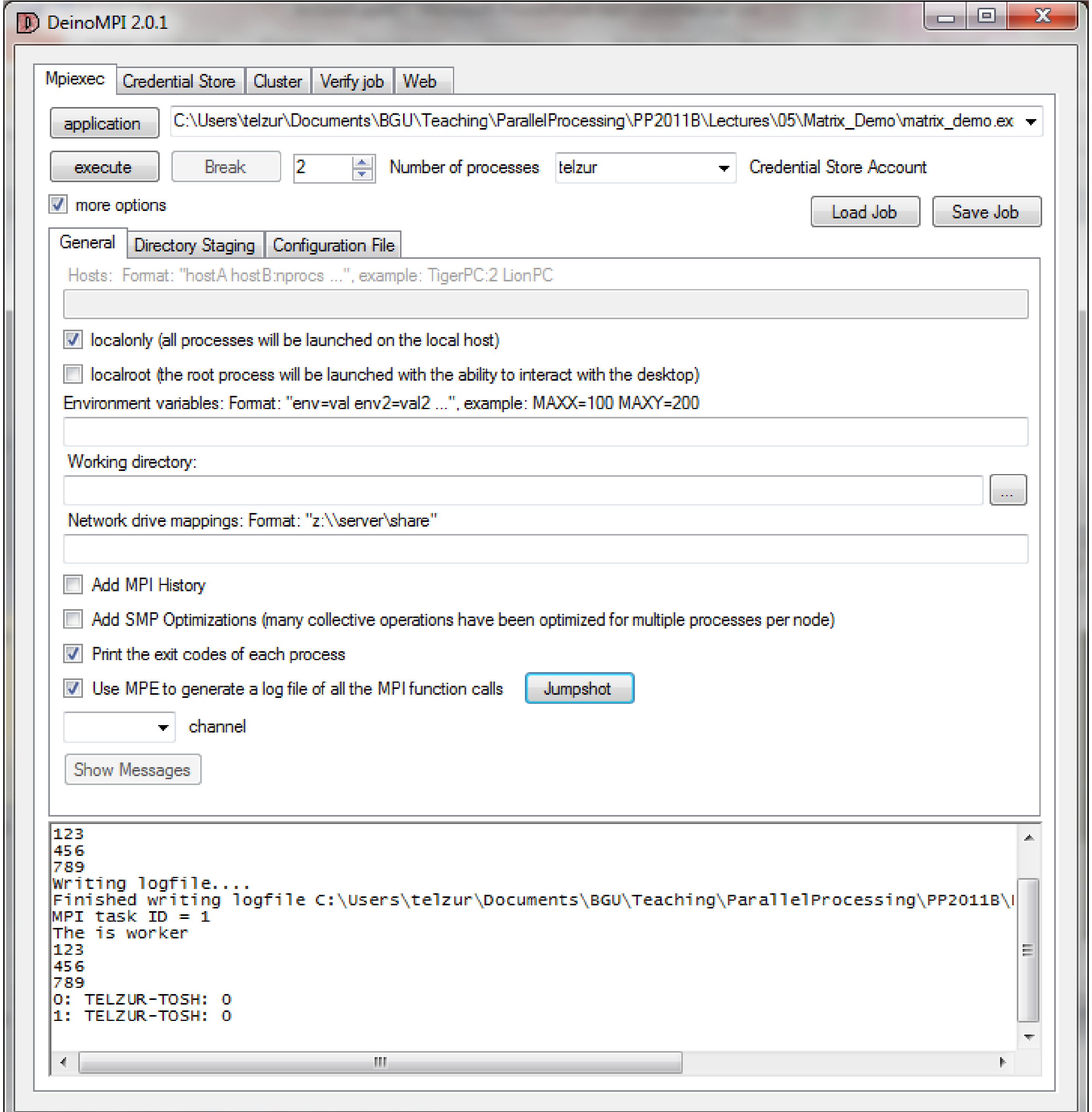
```
 if (mytid == 0) {
       // fill the matrix
       smat[0][0] = 1; smat[1][0] = 4;
smat[2][0] = 7;
       smat[0][1] = 2; smat[1][1] = 5;
smat[2][1] = 8;
       smat[0][2] = 3; smat[1][2] = 6;
smat[2][2] = 9;
printf("Thie is master\n");
       for (i=0;i<dim;i++) {
           for (j=0;j<dim;j++)
               printf("%i",smat[i][j]);
       printf("\n");
       }

       // send the 2D matrix as a linear array
to the Worker

MPI_Send(&smat,dim*dim,MPI_INT,WORKE
R,tag,MPI_COMM_WORLD);

    } else {

MPI_Recv(rmat,dim*dim,MPI_INT,MASTER,t
ag,MPI_COMM_WORLD,&status);
       printf("The is worker\n");
       for (i=0;i<dim;i++) {
           for (j=0;j<dim;j++)
               printf("%i",rmat[i][j]);
       printf("\n");
       }
     }
// That's it!
```



Use DeinoMPI for the demo!!!

Network Performance

The time needed to transmit data

$$cost = L + \frac{N}{B}$$

L = Latency [s]
N = number of bytes [byte]
B = Bandwidth [byte/s]
cost [s]

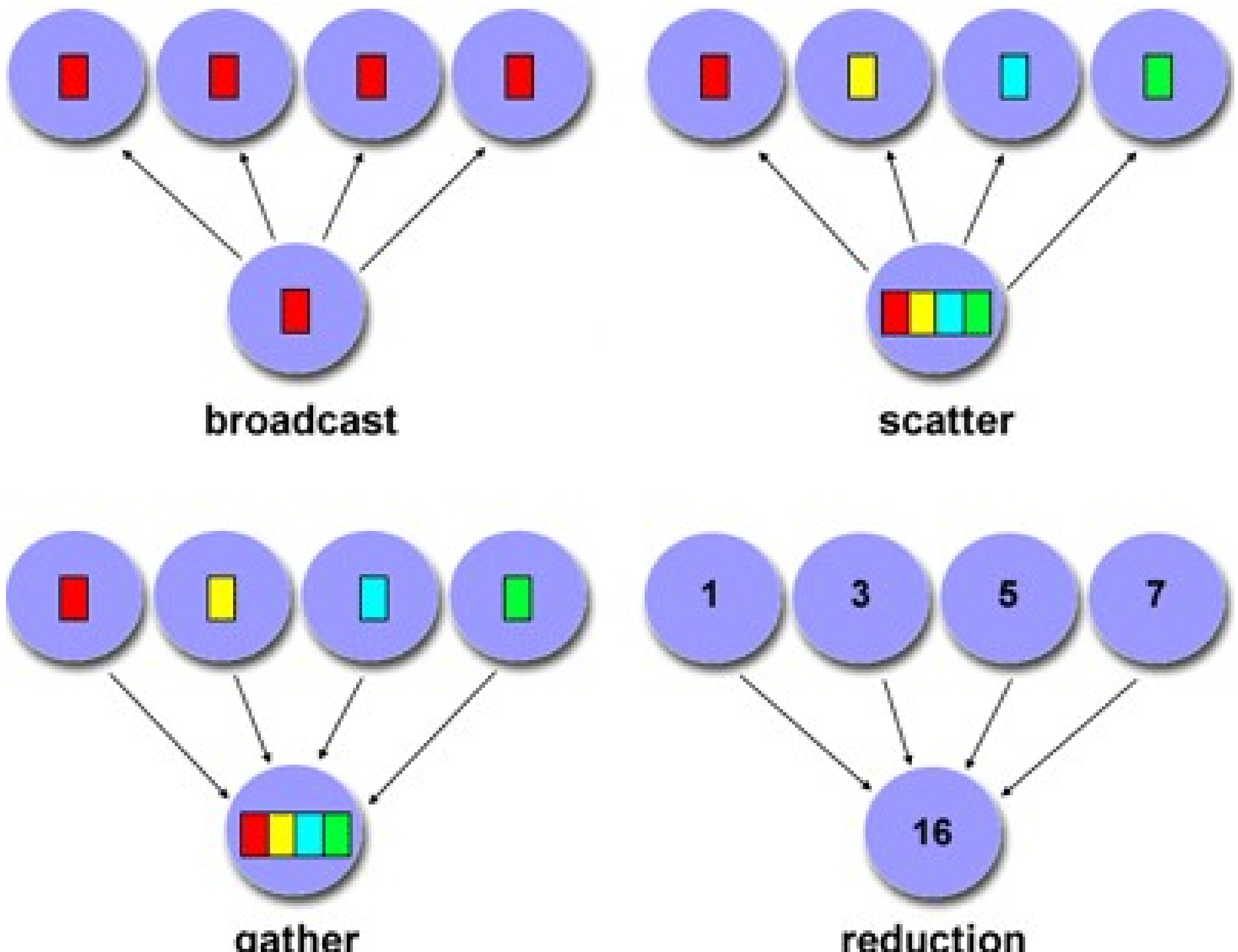
Source: Stanford

MPI_Isend

“Latency Hiding”

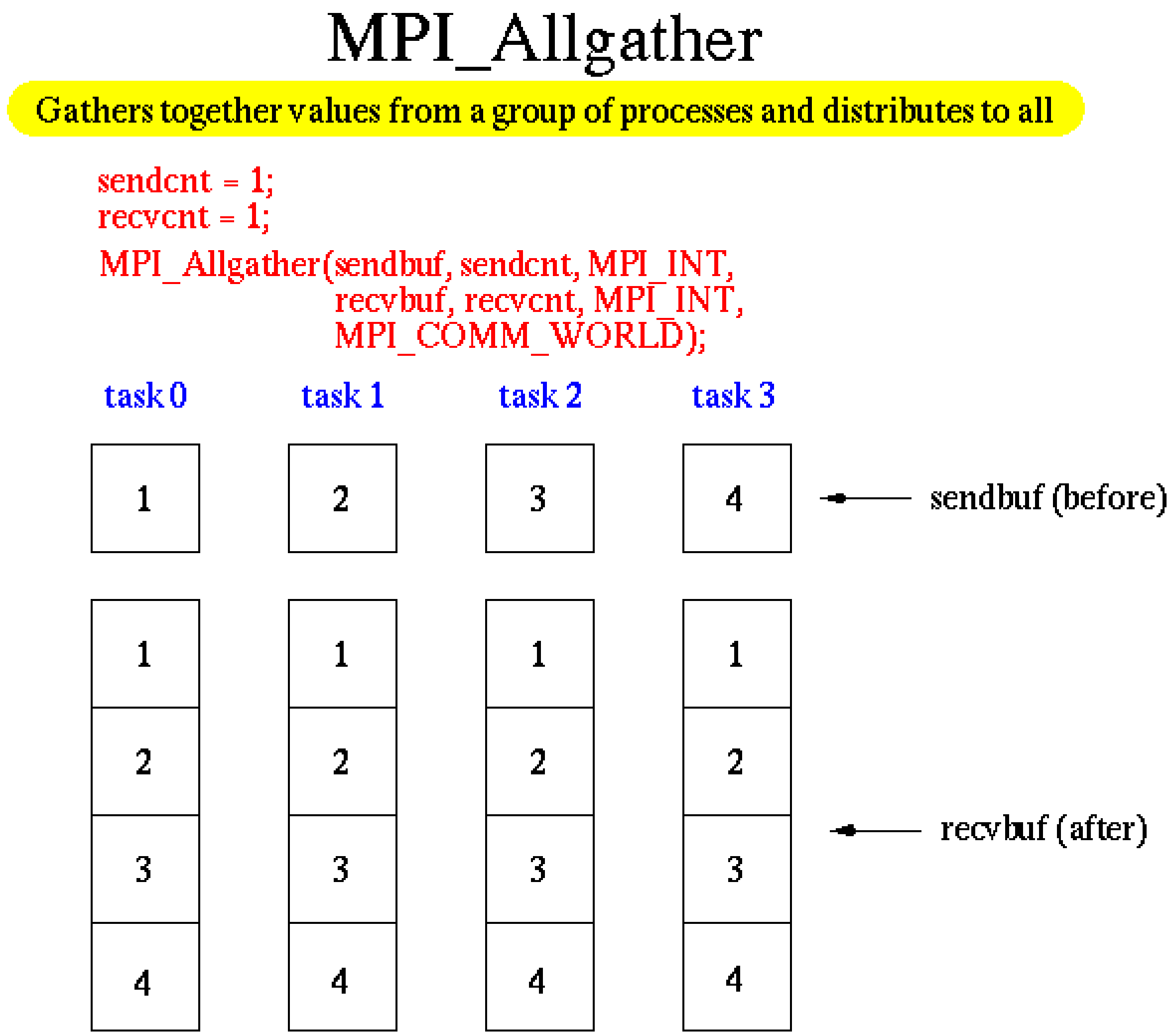
Identifies an area in memory to serve as a send buffer. Processing continues immediately without waiting for the message to be copied out from the application buffer. A communication request handle is returned for handling the pending message status. The program should not modify the application buffer until subsequent calls to **MPI_Wait** or **MPI_Test** indicate that the non-blocking send has completed.

Some Common Collective Commands



Source: https://computing.llnl.gov/tutorials/parallel_comp/

Allgather

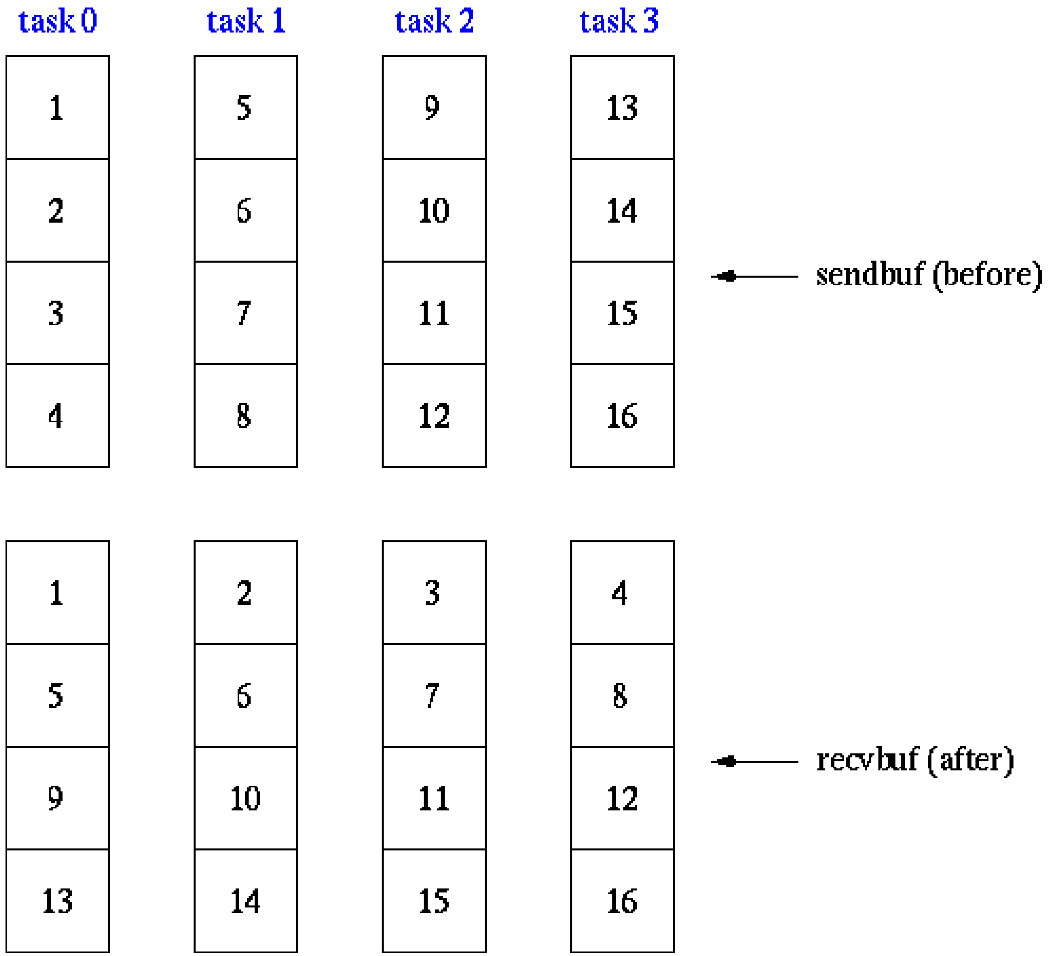


MPI_Allgather
Gathers together values from a group of processes and distributes to all
sendcnt = 1;
recvcnt = 1;
MPI_Allgather(sendbuf, sendcnt, MPI_INT,
recvbuf, recvcnt, MPI_INT,
MPI_COMM_WORLD);
task 0
task 1
task 2
task 3
1
2
3
4
sendbuf (before)
1
2
3
4
1
2
3
4
1
2
3
4
1
2
3
4
recvbuf (after)

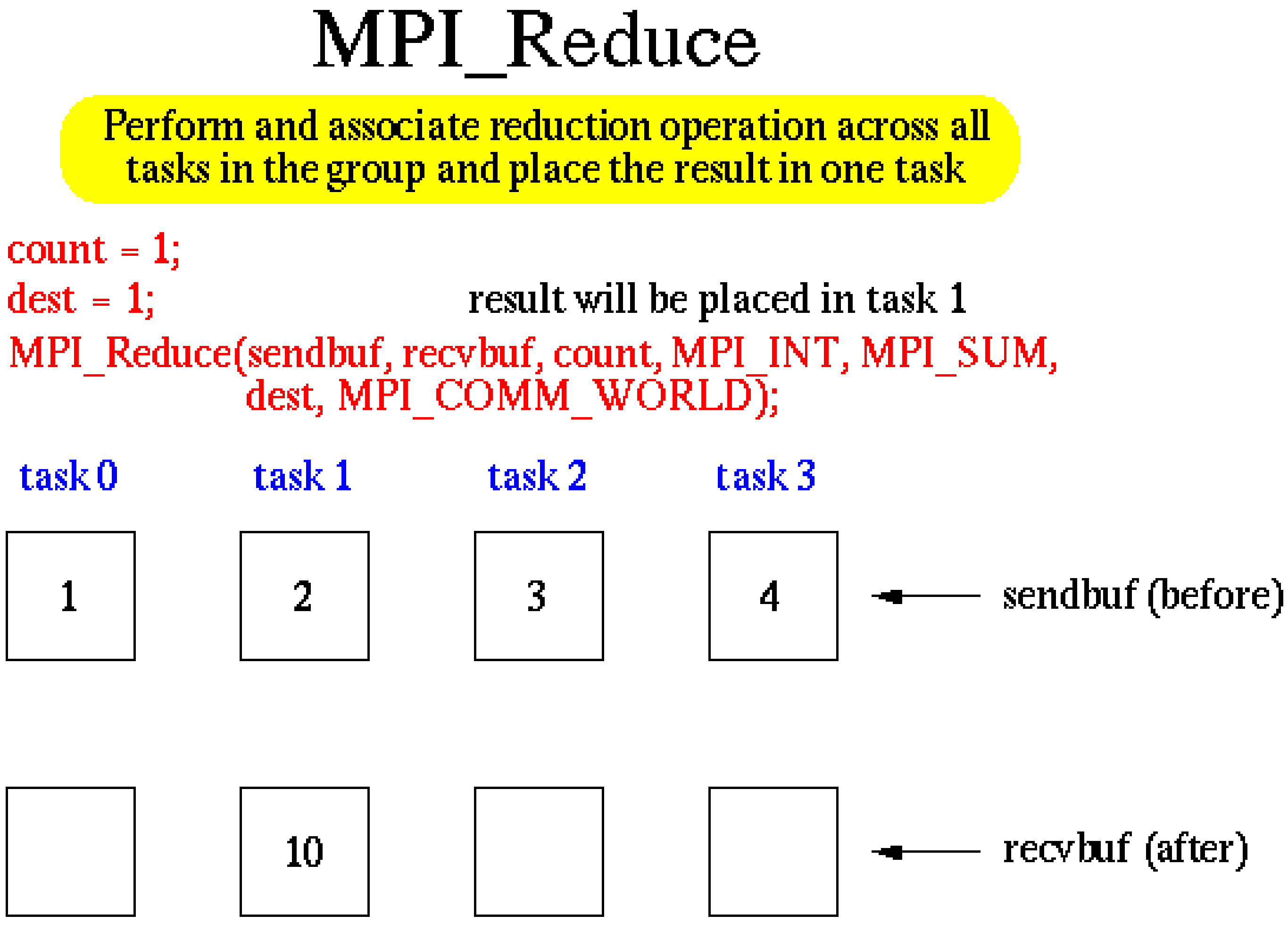
MPI_Alltoall

Sends data from all to all processes. Each process performs a scatter operation.

```
sendcnt = 1;
recvcnt = 1;
MPI_Alltoall(sendbuf, sendcnt, MPI_INT,
             recvbuf, recvcnt, MPI_INT,
             MPI_COMM_WORLD);
```



MPI_Reduce



MPI_Reduce
Perform and associate reduction operation across all tasks in the group and place the result in one task
count = 1;
dest = 1;
result will be placed in task 1
MPI_Reduce(sendbuf, recvbuf, count, MPI_INT, MPI_SUM, dest, MPI_COMM_WORLD);
task 0
task 1
task 2
task 3
1
2
3
4
sendbuf (before)
10
recvbuf (after)

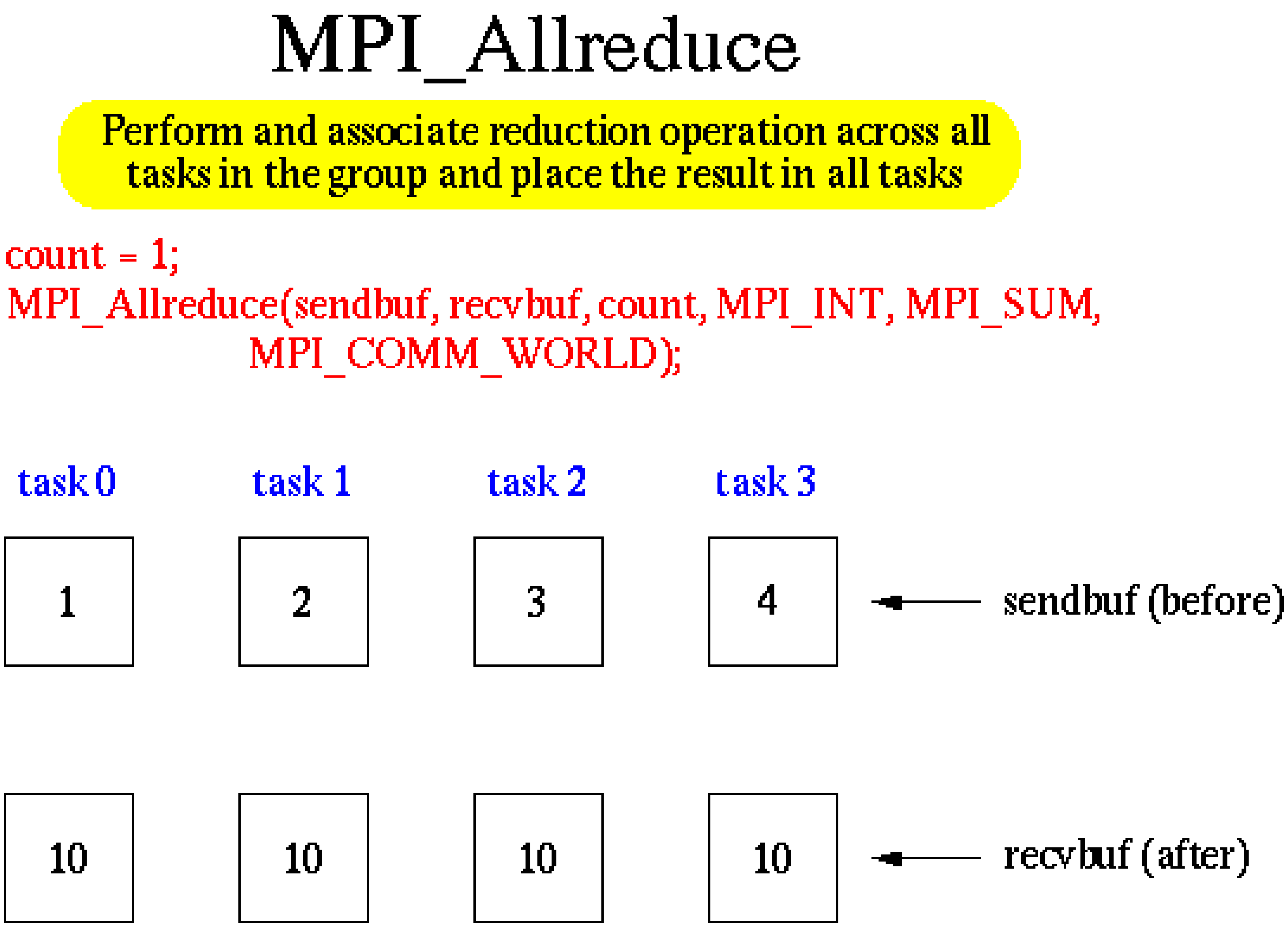
MPI_Reduce

MPI Reduction Operation		C Data Types
MPI_MAX	maximum	integer, float
MPI_MIN	minimum	integer, float
MPI_SUM	sum	integer, float
MPI_PROD	product	integer, float
MPI_LAND	logical AND	integer
MPI_BAND	bit-wise AND	integer, MPI_BYTE
MPI_LOR	logical OR	integer
MPI_BOR	bit-wise OR	integer, MPI_BYTE
MPI_LXOR	logical XOR	integer
MPI_BXOR	bit-wise XOR	integer, MPI_BYTE
MPI_MAXLOC	max value and location	float, double and long double
MPI_MINLOC	min value and location	float, double and long double

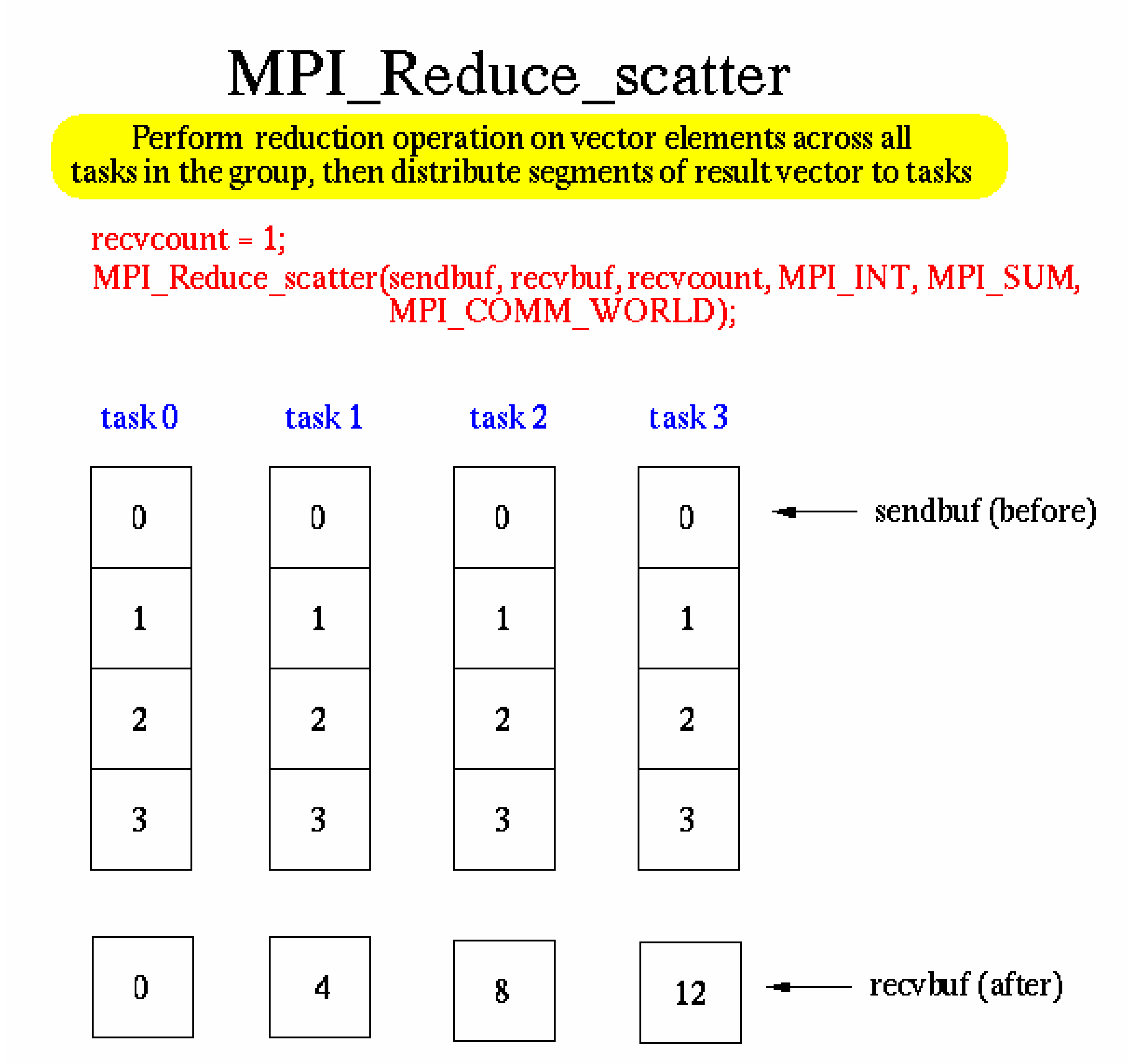
MPI_Reduce

Users can also define their own reduction functions by using the MPI_Op_create routine

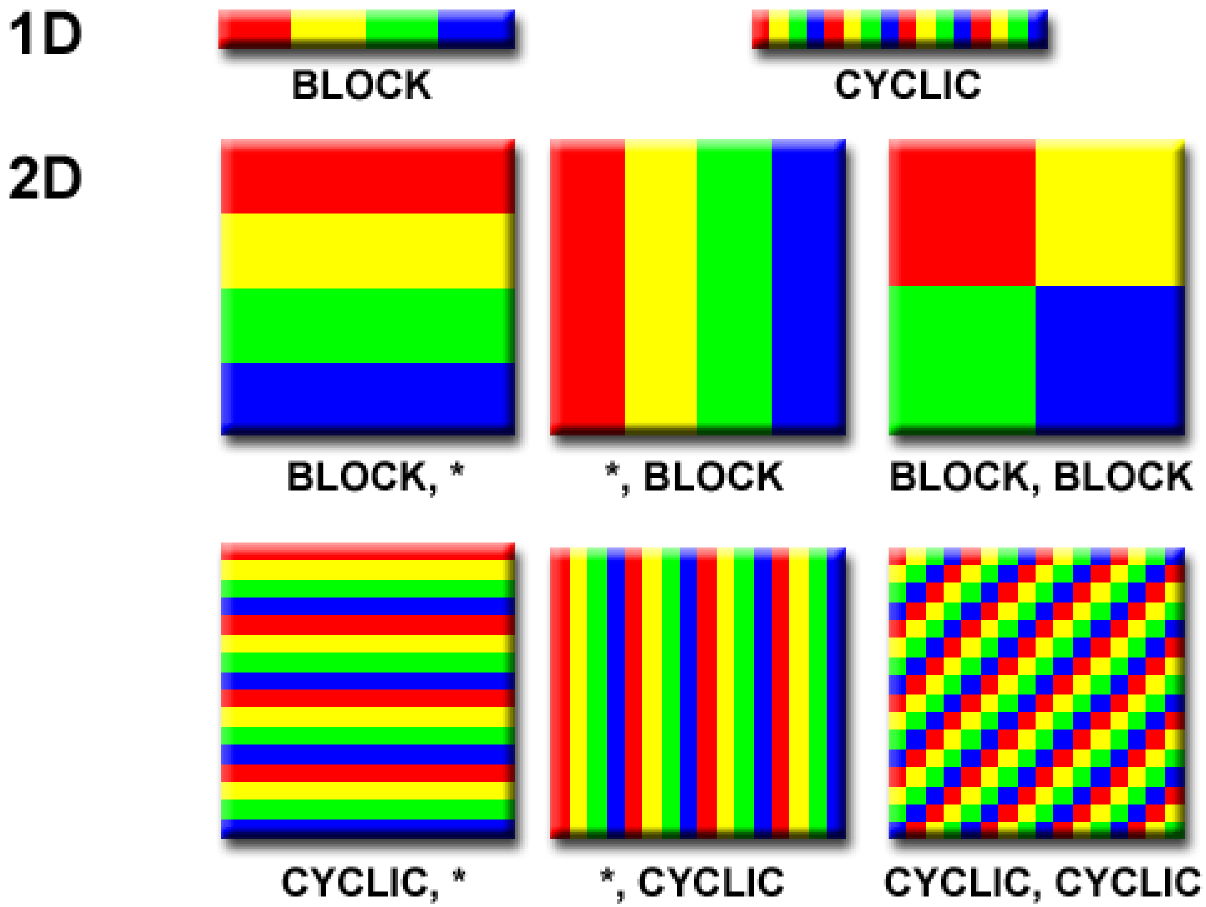
MPI_Allreduce



MPI_Reduce_scatter

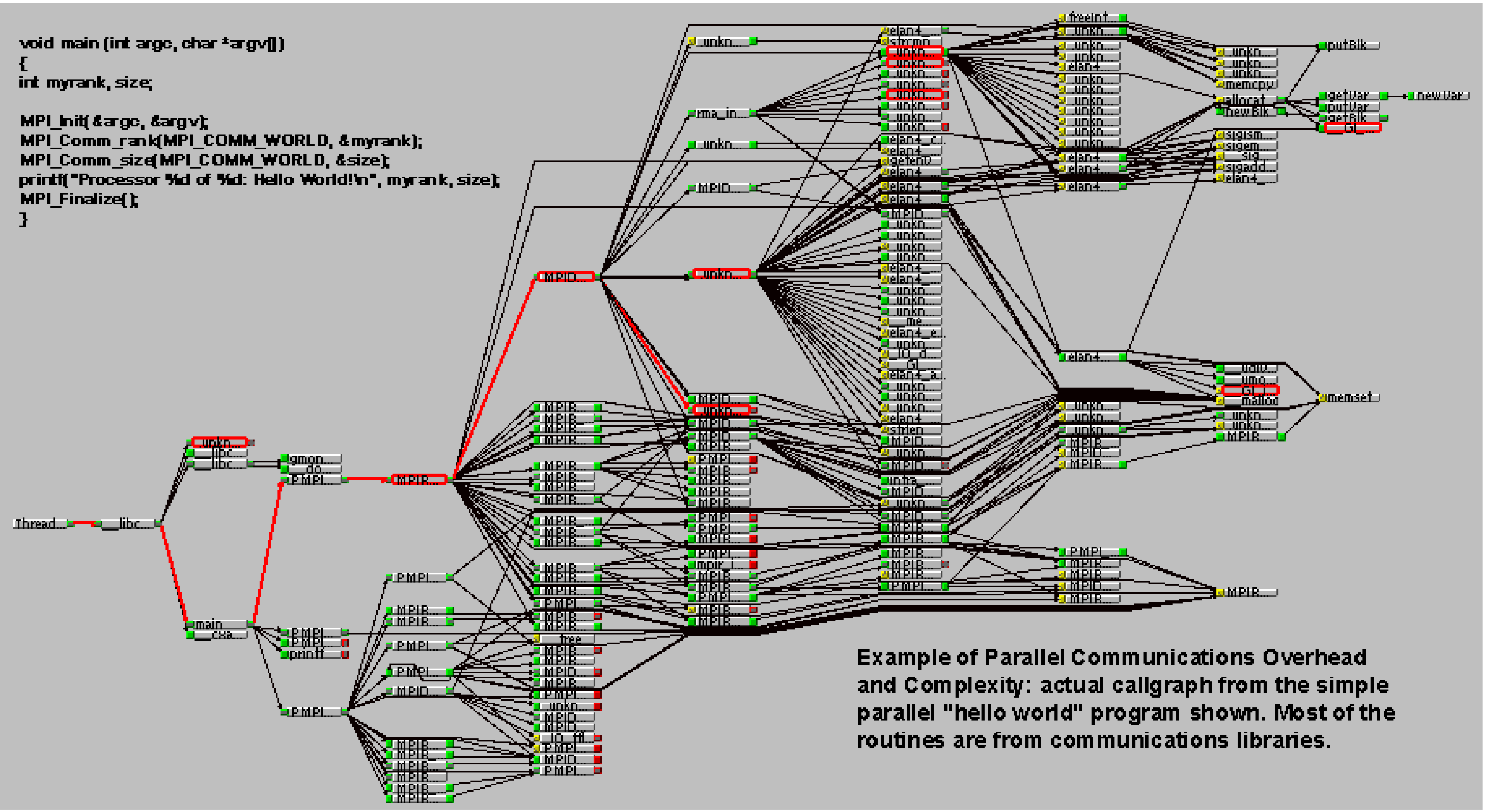


Domain Decomposition: Different ways to partition data



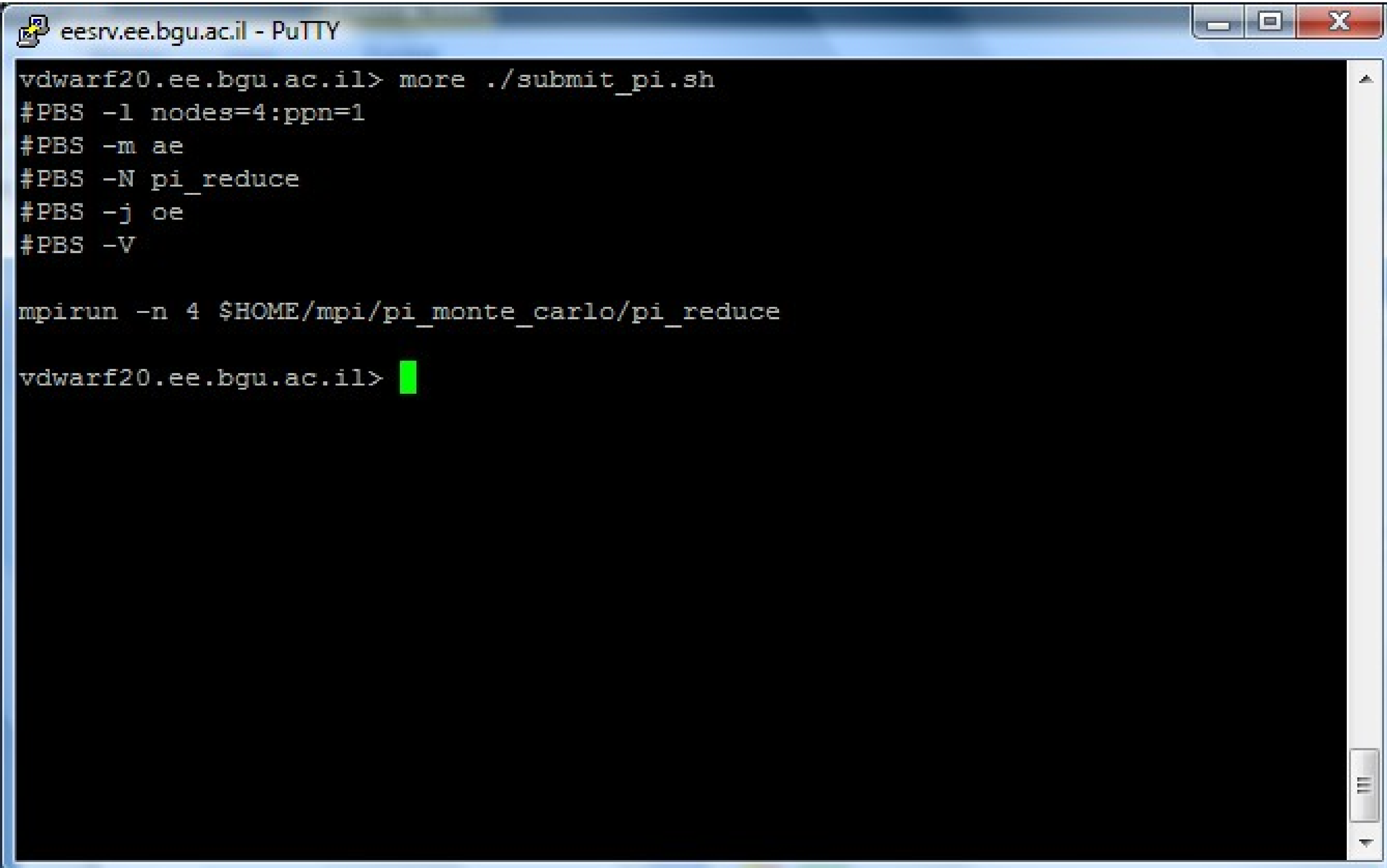
Source: https://computing.llnl.gov/tutorials/parallel_comp/

Overhead and Complexity



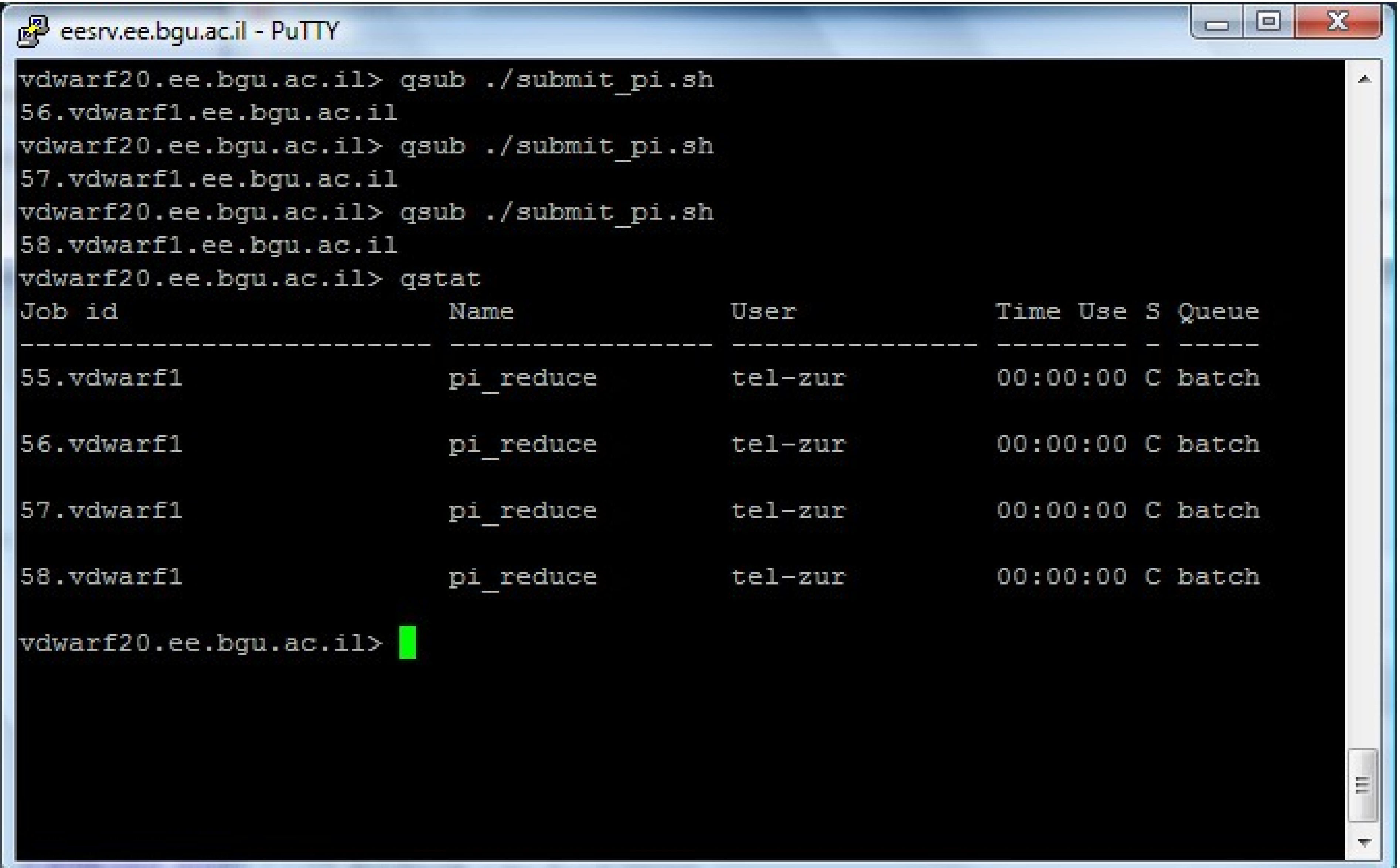
Source: https://computing.llnl.gov/tutorials/parallel_comp/

PBS/Torque – job scheduling



- **-N myjob15** specifies the name of the job will be myjob15
- **-l nodes=1:ppn=1** specifies that the job will use 1 node and that there is 1 processor per node.

PSB/Torque


eesrv.ee.bgu.ac.il - PuTTY
vdwarf20.ee.bgu.ac.il> qsub ./submit_pi.sh
56.vdwarf1.ee.bgu.ac.il
vdwarf20.ee.bgu.ac.il> qsub ./submit_pi.sh
57.vdwarf1.ee.bgu.ac.il
vdwarf20.ee.bgu.ac.il> qsub ./submit_pi.sh
58.vdwarf1.ee.bgu.ac.il
vdwarf20.ee.bgu.ac.il> qstat
Job id Name User Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
55.vdwarf1 pi_reduce tel-zur 00:00:00 C batch
56.vdwarf1 pi_reduce tel-zur 00:00:00 C batch
57.vdwarf1 pi_reduce tel-zur 00:00:00 C batch
58.vdwarf1 pi_reduce tel-zur 00:00:00 C batch
vdwarf20.ee.bgu.ac.il>

PSB/Torque

eesrv.ee.bgu.ac.il - PuTTY

```
Thus no job control in this shell.
stty: standard input: Invalid argument
************************************************
DISPLAY IS SET TO       : 132.72.53.100:0.0
************************************************
tset: standard error: Inappropriate ioctl for device

stty: standard input: Invalid argument
Sun Nov 15 14:39:12 IST 2009
WARNING: cannot identify machine, uname=Linux
MPI task ID = 0
MPI task ID = 1
MPI task ID = 2
MPI task ID = 3
   After 5000 throws, average value of pi = 3.14360000
   After 10000 throws, average value of pi = 3.13500000
   After 15000 throws, average value of pi = 3.14506667
   After 20000 throws, average value of pi = 3.14670000
   After 25000 throws, average value of pi = 3.14196000
   After 30000 throws, average value of pi = 3.14516667
   After 35000 throws, average value of pi = 3.14880000
   After 40000 throws, average value of pi = 3.14612500
   After 45000 throws, average value of pi = 3.14673333
--More--(94%)
```

eesrv.ee.bgu.ac.il - PuTTY

```
59.vdwarf1.ee.bgu.ac.il
vdwarf20.ee.bgu.ac.il> qsub submit_pi.sh
60.vdwarf1.ee.bgu.ac.il
vdwarf20.ee.bgu.ac.il> qsub submit_pi.sh
qs61.vdwarf1.ee.bgu.ac.il
vdwarf20.ee.bgu.ac.il> qstat -an

vdwarf1.ee.bgu.ac.il:
                                                                   Req'd  Req'd
  Elap
Job ID               Username Queue    Jobname    SessID NDS   TSK Memory Time
S Time
-------------------- -------- -------- ---------- ------ ----- --- ------ -----
- -----
59.vdwarf1.ee.bgu.ac tel-zur  batch    pi_reduce     --      4  --     --  01:00
C 00:00
   vdwarf4+vdwarf3+vdwarf2+vdwarf1
60.vdwarf1.ee.bgu.ac tel-zur  batch    pi_reduce     --      4  --     --  01:00
C 00:00
   vdwarf4+vdwarf3+vdwarf2+vdwarf1
61.vdwarf1.ee.bgu.ac tel-zur  batch    pi_reduce     --      4  --     --  01:00
C 00:00
   vdwarf4+vdwarf3+vdwarf2+vdwarf1
vdwarf20.ee.bgu.ac.il>
```

References

- **Torque**: http://www.clusterresources.com/products/torque-resource-manager.php

- **OpenPBS**: http://openpbs.org

Parallel Computing environments for this course

- Our Linux Cluster: **hobbit**
- Your own resources:
  - A dedicated Linux installation with MPI
  - Install MPI on Windows
  - Dual boot Linux/Windows
  - Windows + Cygwin
  - Virtualization & **Vi-HPS**!!!!:
    - VMware Player, VirtualBox…
  - Parallel Computing on the Cloud (**$**)

Profiling with mpiP

Lightweight, Scalable MPI Profiling

אפשר לעצור כאן
(לקורסים עתידיים)

This tool isn't installed yet on the hobbits

Check on Vi-HPS

http://mpip.sourceforge.net/

On Lifebook ~/tests/mpiP

```
mpicc -o cpi ./cpi.c \
-L /home/telzur/Downloads/mpiP-3.4.1/lib \
-lmpiP -L/usr/local/lib -lunwind -lm

mpirun -np 8 ./cpi
```

more ./cpi.8.32737.1.mpiP

```
@ mpiP
@ Command : ./cpi
@ Version                  : 3.4.1
@ MPIP Build date          : Dec 15 2014, 13:17:00
@ Start time               : 2014 12 15 13:28:16
@ Stop time                : 2014 12 15 13:28:16
@ Timer Used               : PMPI_Wtime
@ MPIP env var             : [null]
@ Collector Rank           : 0
@ Collector PID            : 32737
@ Final Output Dir         : .
@ Report generation        : Single collector task
@ MPI Task Assignment      : 0 LIFEBOOK
@ MPI Task Assignment      : 1 LIFEBOOK
@ MPI Task Assignment      : 2 LIFEBOOK
@ MPI Task Assignment      : 3 LIFEBOOK
@ MPI Task Assignment      : 4 LIFEBOOK
@ MPI Task Assignment      : 5 LIFEBOOK
@ MPI Task Assignment      : 6 LIFEBOOK
@ MPI Task Assignment      : 7 LIFEBOOK
```

Cont'

```
        Bcast                    2       34.3    5.81   90.31    0.59
        Reduce                   1       3.68    0.62    9.69    1.83
---------------------------------------------------------------------
                                                                -----
    @--- Aggregate Sent Message Size (top twenty, descending, bytes)
                                                           ----------
---------------------------------------------------------------------
                                                                -----
     Call                 Site      Count      Total       Avrg  Sent%
     Reduce                  1          8         64          8  50.00
     Bcast                   2         16         64          4  50.00
---------------------------------------------------------------------
                                                                -----
               @--- Callsite Time statistics (all, milliseconds): 16
                                                 --------------------
---------------------------------------------------------------------
                                                                -----
Name                 Site Rank  Count      Max     Mean      Min   App%
                                                                   MPI%
 Bcast                  2    0      2   0.0715   0.0385   0.0055  0.11
                                                                  67.10
 Bcast                  2    1      2    0.564    0.401    0.238  1.09
                                                                  97.07
```

Cont'

```
Reduce                  1     0       1   0.0377   0.0377   0.0377   0.05
                                                                    32.90
Reduce                  1     1       1   0.0243   0.0243   0.0243   0.03
                                                                     2.93
Reduce                  1     2       1     1.25     1.25     1.25   1.70
                                                                    19.12
Reduce                  1     3       1   0.0245   0.0245   0.0245   0.03
                                                                     0.46
Reduce                  1     4       1     2.25     2.25     2.25   3.05
                                                                    32.14
Reduce                  1     5       1    0.025    0.025    0.025   0.03
                                                                     0.52
Reduce                  1     6       1    0.027    0.027    0.027   0.04
                                                                     0.45
Reduce                  1     7       1   0.0343   0.0343   0.0343   0.05
                                                                     0.47
Reduce                  1     *       8     2.25     0.46   0.0243   0.62
                                                                     9.69
---------------------------------------------------------------------
                                                               ------
              @--- Callsite Message Sent statistics (all, sent bytes)
                                                   -------------------
---------------------------------------------------------------------
                                                               ------
```