Setup

Log in to Salomon

Update (or clone) the git repository
$ (git clone https://code.it4i.cz/jansik/SummerOfHPC.git)
$ git pull

Get interactive allocation on Salomon
$ qsub -A DD-17-18 -q R1075290 -l select=8:ncpus=1,place=scatter -I

$ qstat –a
$ qstat –u username –n

Lab 1

MPI Hello world!

The goal is to write and run a simple MPI Hello world program

MPI Hello world

File: helloworld.c

Tasks:

1. Run mpirun hostname; mpirun –n XX hostname
2. Modify helloworld.c so that it it runs with multiple MPI, writing out ***Hello world rank XX of YY on host ZZZ***
3. Change number of processes

Ingredients:

- https://www.open-mpi.org/doc/v1.10/
- MPI_Init(), MPI_Comm_size(), MPI_Comm_rank()
- MPI_Get_processor_name(), MPI_COMM_WORLD

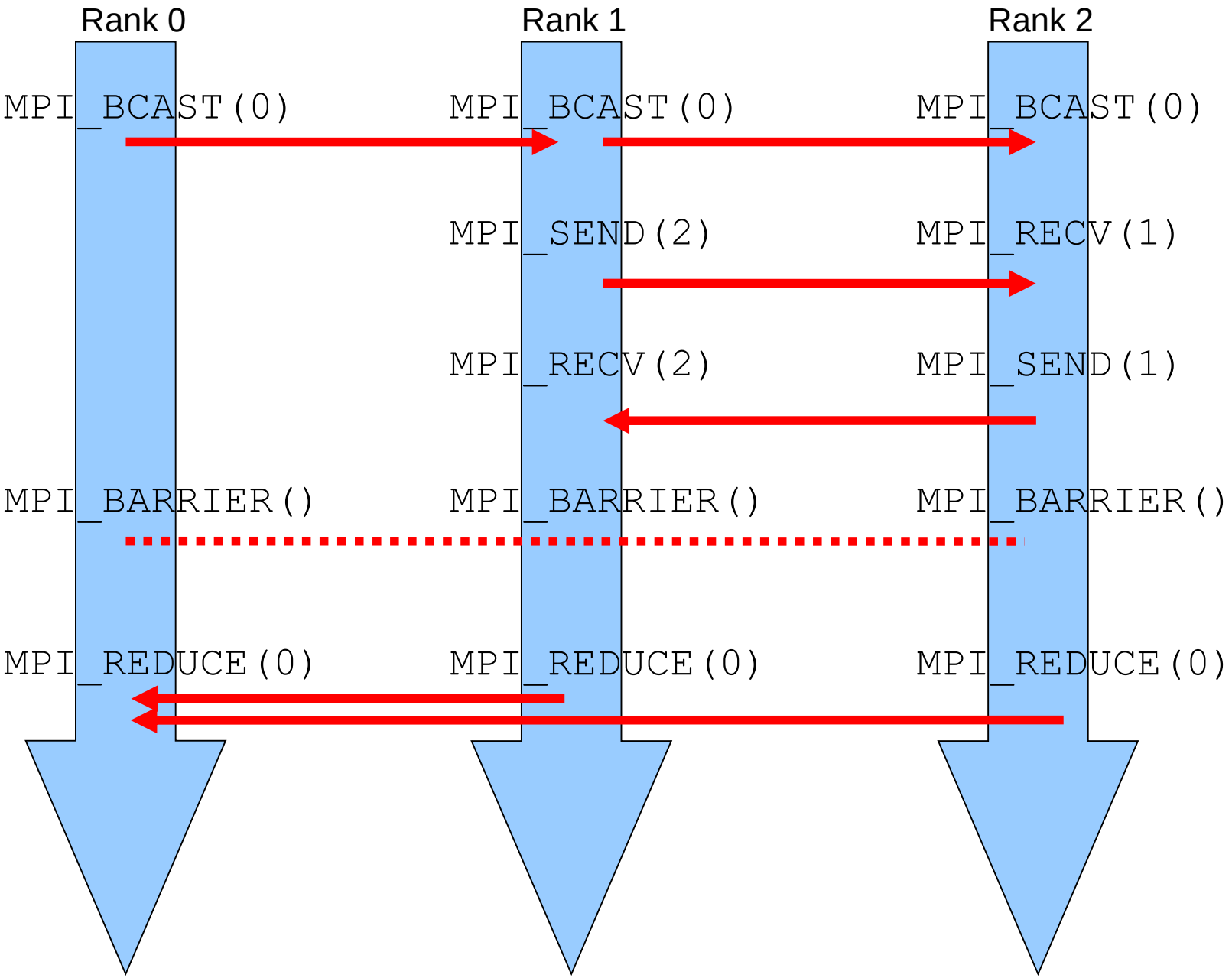
Compilation and execution:

- mpiicc helloworld.c –o helloworld.x
- mpirun ./helloworld.x
- mpirun –n XX ./helloworld.x

Lab 2

Send messages among the processes

Try out blocking point to point and collective communications


Rank 0
Rank 1
Rank 2
MPI_BCAST(0)
MPI_BCAST(0)
MPI_BCAST(0)
MPI_SEND(2)
MPI_RECV(1)
MPI_RECV(2)
MPI_SEND(1)
MPI_BARRIER()
MPI_BARRIER()
MPI_BARRIER()
MPI_REDUCE(0)
MPI_REDUCE(0)
MPI_REDUCE(0)

Exchanging messages

File: msgexchane.c

Tasks:

1. Review the msgexchange.c, compare to the message exchange picture
2. Modify msgexchange.c so that it works
3. Run msgexchange.x using 3 processes
4. Run msgexchange.x using other number of processes

Ingredients:

- https://www.open-mpi.org/doc/v1.10/
- MPI_Bcast(), MPI_Barrier
- MPI_Send(), MPI_Recv()
- MPI_Reduce()

Compilation and execution:

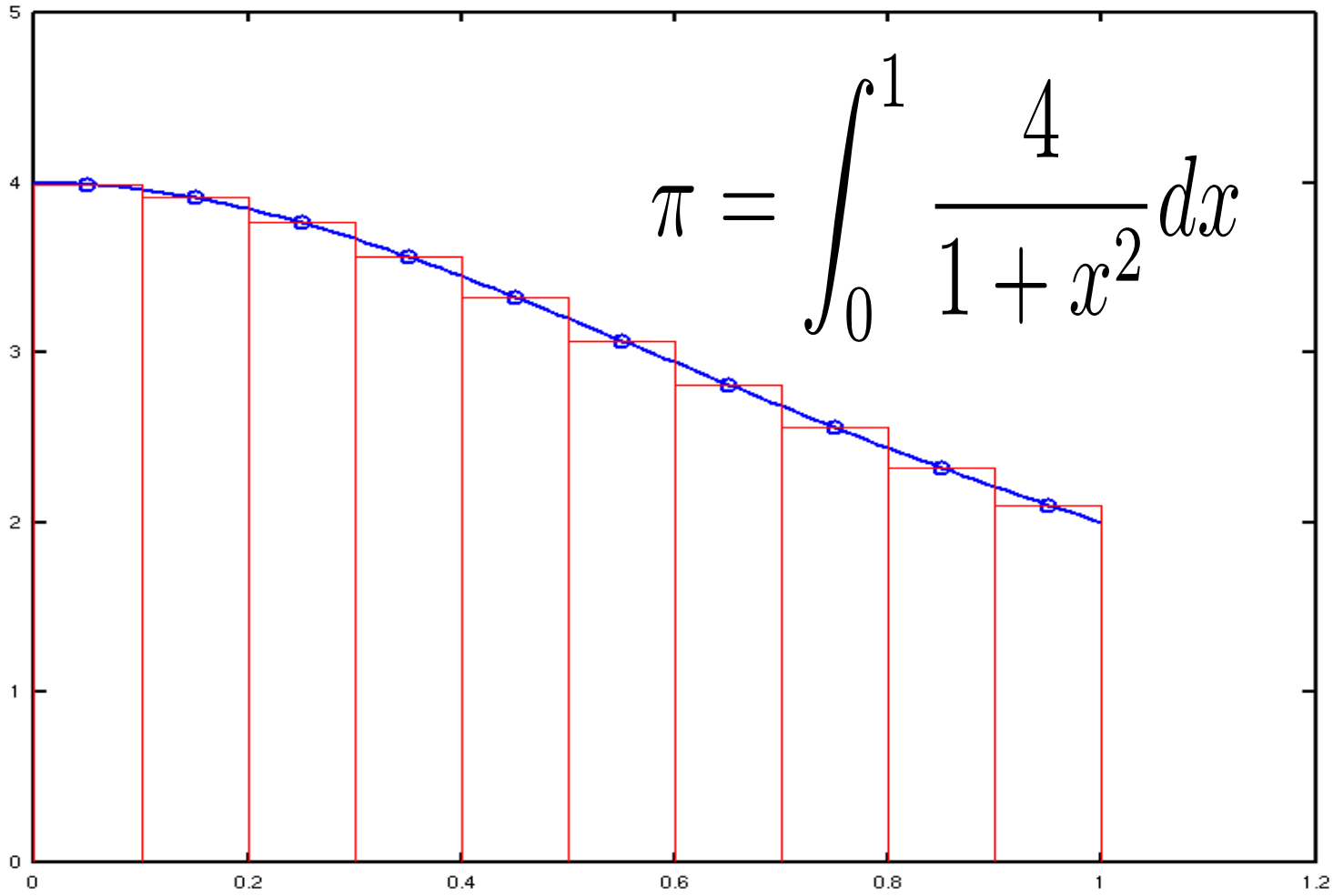
- mpiicc msgexchange.c –o msgexchange.x
- mpirun –n 3 ./msgexchange.x
- mpirun –n XX ./msgexchange.x

Lab 3

Parallelize pi integration using MPI

The goal is to have MPI parallel program integrating value of pi.

OpenMP pi integration



OpenMP pi integration

Files: pi3serial.c, pi3halfdone.c

Tasks:

1. Review pi3serial.c, compile and run, rationalize the results.
2. Parallelize using MPI (use pi3halfdone.c)
3. Find out which iterations are executed by which process
4. Add OpenMP on top of MPI, use #pragma **omp parallel for,** use reduction clause
5. Find out which threads on what ranks execute which loop iterations.
6. Replace MPI_Reduce by MPI_Send/MPI_Recv.

Ingredients:

- MPI_Bcast()
- MPI_Reduce()
- #pragma omp parallel
- omp_get_thread_num()
- omp_get_num_threads()
- clause private(var), shared(var)
- clause reduction(+:var)
- #pragma omp parallel for
- MPI_Send()/MPI_Recv()

Compilation:

- mpiicc –qopenmp pi3halfdone.c –o pi3halfdone.x

Thank you!