Using Shared Computing Resources

Joseph R. Peterson

CATMS Lunch

Oct. 6th, 2016

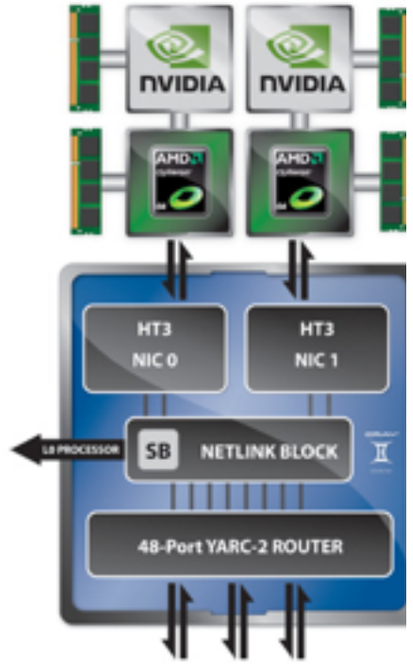
Supercomputers

“Shared Computing Resources”

- Clusters
  - Many connected computers
  - Shared among by few researchers (group, department, institution)
- Supercomputers
  - High performance interconnect
  - High performance storage
  - Shared among many researchers around the country (world)

Supercomputer Terminology

- Node
- Rack
- Computer Allocation
  - XSEDE
  - PRAC
  - INCITE/ALCC
  - SU - Service Unit (resource/time)



Super Computing Facilities

- NSF
  - XSEDE - e.g. PSC, SDSC, TACC,
  - NCSA - e.g. Blue Waters

XSEDE
Extreme Science and Engineering Discovery Environment

NCSA

- DOE
  - ORNL's OLCF - e.g. Titan, Eos, Rhea
  - PNNL - e.g. Cascade, Summit (coming soon)
  - ANL - e.g. Mira
  - LANL - e.g. Cielo, Trinity
  - LLNL - e.g. Vulcan, Sequoia

OLCF
Oak Ridge Leadership Computing Facility

Top 500

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	National Supercomputing Center in Wuxi China	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway NRCPC	10,649,600	93,014.6	125,435.9	15,371
2	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
3	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
4	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
5	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660

Logging In

- Use secure shell and forward the display:
  - `ssh -X <username>@h2ologin.ncsa.illinois.edu`
- Maybe forward a port:
  - `ssh -X -L 8888:localhost:8888 <username>@maverick.tacc.xsede.org`

Logging In

```
[jpeterson@wirelessprv-10-194-25-236:~/Programming/bootcamp/src$ ssh bw

 Access by OTP or Two Factor Certificate Authority only.
 Use myproxy-logon -s tfca.ncsa.illinois.edu -p 7512 for gsissh access.
 gsissh or ssh -o PreferredAuthentications=keyboard-interactive for otp access.


 Blue Water Admin Team

[Enter PASSCODE:
[Enter PASSCODE:
 Last login: Wed Oct  5 11:45:04 2016 from dagr.scs.uiuc.edu

    ___  __                 _      __     __
   / _ )/ /_ _____   | | /| / /__ _/ /____ _______
  / _  / / // / -_)  | |/ |/ / _ `/ __/ -_) __(_-<
 /____/_/\_,_/\__/   |__/|__/\_,_/\__/\__/_/ /___/
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Batch and Scheduler configuration.
  Queues:  normal (default), high, low, debug
  Features:  "xe" (default), "xk", "x" (xe or xk non-specific)
  30 min default wall time,
  -lnodes=X:ppn=Y syntax supported.

All SSH traffic on this system is monitored.

06-10-15 17:57
Blue Waters Discounts Available: For details, see:
https://bluewaters.ncsa.illinois.edu/manage-news/-/blogs/blue-waters-discounts-a
vailable

01-10-16 23:27
The maximum walltime for all queues, except the debug queue, is now 48 hours. We
will be evaluating the impact of a longer wall clock time for jobs on the system
workload and science throughput.

Questions?  Mail help+bw@ncsa.illinois.edu to create a support ticket.
For known issues:  https://bluewaters.ncsa.illinois.edu/known-issues
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

josephp@h2ologin3:~>
```

Queueing System

- Schedules
  - Portable Batch System
  - SLURM (F#*$ing eeew!)
  - Moab
  - [Sun/Oracle/Univa] Grid Engine
- Commands
  - **`qstat`** - check the queue status
  - **`qsub`** - submit a job to the queue
  - **`qdel/qalter`** - delete/modify a job
  - **`showq`** - sometimes this exists…

Queueing 1 liners

- qstat - check the queue status
  - Display aggregate statistics: **`qstat -g c`**
  - Display full information: **`qstat -f`**
  - Info about my jobs: **`qstat -u <username>`**
- qsub - submit a job to the queue
  - Submit a job: **`qsub input.pbs`**
  - Submit a job with a dependency: **`qsub -hold_jid <jid1> <jid2> …`**
  - Run an interactive job: **`qsub -I`**
- qdel/qalter - delete/modify a job
  - Delete a job: **`qdel <jid>`**
  - Put a job on hold: **`qalter -h u <jid>`**
  - Remove a hold: **`qalter -U <jid>`**

PBS scripts

```
##############################
# Scheduler Configuration #
##############################
# Set Up Email Preferences #
#PBS -M <email address>
#PBS -m bea
#PBS -q normal
#PBS -l walltime=24:00:00
#PBS -l nodes=16:ppn=32:xe
#PBS -N <job name>
#PBS -V
#PBS -cwd
#PBS -A <account number>


############################
# Pre-Job Configuration #
############################
# Output file
OUTPUT=$PBS_JOBID.log

# Alternative to (-cwd)
cd $PBS_O_WORKDIR
```

```
# Save some information about the job to the file
echo "#############" >> $OUTPUT
echo "# JOB SETUP #" >> $OUTPUT
echo "#############" >> $OUTPUT
echo "Job Start Time:" >> $OUTPUT
date >> $OUTPUT
echo "Job ID: $PBS_JOBID" >> $OUTPUT
echo "Queue: $PBS_O_QUEUE" >> $OUTPUT
echo "Node List:" >> $OUTPUT
cat $PBS_NODEFILE >> $OUTPUT
echo "Work Directory: $PBS_O_WORKDIR" >> $OUTPUT
echo "" >> $OUTPUT

#################
# Execute Work #
#################
echo "###############" >> $OUTPUT
echo "# JOB OUTPUT #" >> $OUTPUT
echo "###############" >> $OUTPUT
aprun -n 512 program.exe >> $OUTPUT

####################
# Post-Processing #
####################
echo "Job End Time:" >> $OUTPUT
date >> $OUTPUT
```

Modules

- “Dynamic Modification of User’s environment via *modulefiles*”
- **`module avail`** - Show available modules
- **`module list`** - Show currently loaded modules
- **`module load <mod name>`** - Load a module
- **`module swap <mod1> <mod2>`** - Swap one module for another (e.g. intel for gcc compilers)

Modules

- `module avail`

```
------------------------------------------------------------ /usr/local/modulefiles ------------------------------------------------------------
firefox/31.0(default) globus/5.2.0          globus/5.2.4          globus/5.2.5(default) gsissh/5.8p2          gsissh/6.2p2(default)

------------------------------------------------------------ /opt/cray/gem/modulefiles ------------------------------------------------------------
alps/5.2.0-2.0502.8594.12.4.gem                krca/1.0.0-2.0401.33562.3.95.gem                rca/1.0.0-2.0502.53711.3.125.gem
alps/5.2.1-2.0502.9041.11.6.gem                krca/1.0.0-2.0401.36792.3.70.gem                rca/1.0.0-2.0502.53711.3.125.gem-get-fix
alps/5.2.1-2.0502.9205.18.1.gem                krca/1.0.0-2.0401.37429.5.1.gem                 rca/1.0.0-2.0502.60530.1.63.gem(default)
alps/5.2.1-2.0502.9649.23.1.gem                krca/1.0.0-2.0401.37429.5.21.gem                sdb/1.0-1.0401.38148.4.27.gem
alps/5.2.1-2.0502.9710.28.1.gem                krca/1.0.0-2.0402.38880.1.110.gem               sdb/1.0-1.0402.41597.10.13.gem
```

- `module list`

```
[josephp@h2ologin3:~> module list
Currently Loaded Modulefiles:
  1) modules/3.2.10.4                          11) gni-headers/4.0-1.0502.10859.7.8.gem  21) moab/9.0.2-1469837953_f87b286-sles11
  2) eswrap/1.1.0-1.020200.1231.0              12) xpmem/0.1-2.0502.64982.5.3.gem        22) java/jdk1.8.0_51
  3) cce/8.4.6                                 13) dvs/2.5_0.9.0-1.0502.2188.1.113.gem   23) globus/5.2.5
  4) craype-network-gemini                     14) alps/5.2.4-2.0502.9774.31.12.gem      24) gsissh/6.2p2
  5) craype/2.5.4                              15) rca/1.0.0-2.0502.60530.1.63.gem       25) darshan/2.3.0.1
  6) cray-libsci/16.03.1                       16) atp/2.0.1                             26) scripts
  7) udreg/2.3.2-1.0502.10518.2.17.gem         17) PrgEnv-cray/5.2.82                    27) user-paths
  8) ugni/6.0-1.0502.10863.8.28.gem            18) cray-mpich/7.3.3                      28) xalt/0.7.5
  9) pmi/5.0.10-1.0000.11050.179.3.gem         19) craype-interlagos                     29) bwpy/0.3.0
 10) dmapp/7.0.1-1.0502.11080.8.74.gem         20) torque/6.0.1                          30) bwpy-mpi/0.3.0
```

...

SCS Triton (5 queues)

- ge1 - 34 nodes
  - 2-core AMD Opteron 175 @ 2.2 GHz
  - 4 GB Memory
- ge2 - 10 nodes
  - 2-core AMD Opteron 1222 @ 3 GHz
  - 4 GB Memory
- ge3 - 8 nodes
  - 4-core AMD Opteron 2354 @ 2.2 GHz
  - 8 GB Memory
- ib1 - 14 nodes
  - 8-core AMD Opteron 6136 @ 2.4 GHz
  - 32 GB Memory
- webmo - 4 nodes
  - 2-core AMD Opteron 175 @ 2.2 GHz
  - 4 GB Memory

Parallel Programming

Parallel Computing Methodologies

- Types
  - SIMD - Single Instruction Multiple Data
  - MIMD - Multiple Instruction Multiple Data
- Parallelism Levels
  - Thread-level parallelism
  - Process-level parallelism

Parallel Programming Technologies

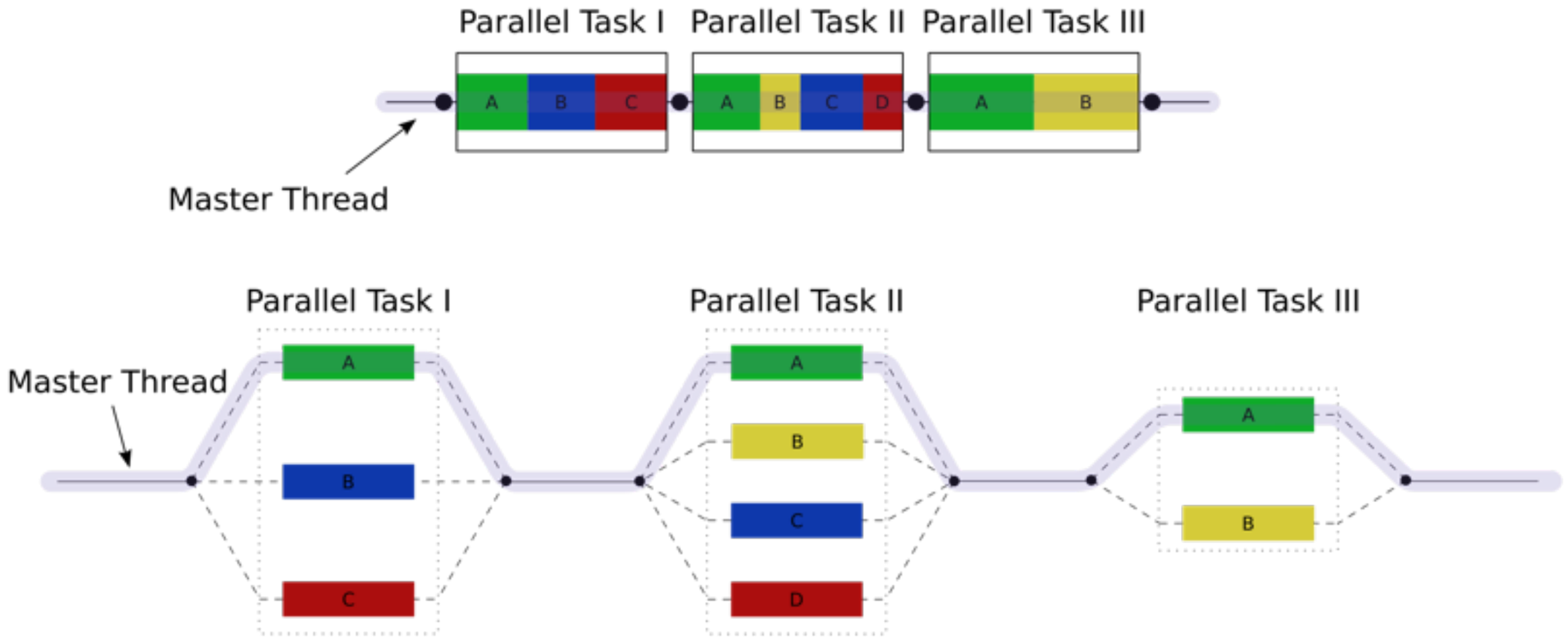
- Thread-level
  - pthreads
  - C++ threads
  - OpenMP
- Process-level
  - ~~Fork~~
  - MPI
- Accelerator APIs
  - CUDA
  - OpenCL
  - OpenACC
- Hybrid APIs/Programming Languages
  - ~~Charm++~~
  - ~~Kokkos~~
  - ~~Unified Parallel C~~

Parallel Programming Technologies

- Thread-level
  - pthreads
  - C++ threads
  - OpenMP
- Process-level
  - ~~Fork~~
  - MPI
- Accelerator APIs
  - CUDA
  - OpenCL
  - OpenACC
- Hybrid APIs/Programming Languages
  - ~~Charm++~~
  - ~~Kokkos~~
  - ~~Unified Parallel C~~

OpenMP Basics

- Compiler “directives”
  - automatically parallelize portions of your code
  - based on user specified rules



https://upload.wikimedia.org/wikipedia/commons/thumb/f/f1/Fork_join.svg/2000px-Fork_join.svg.png

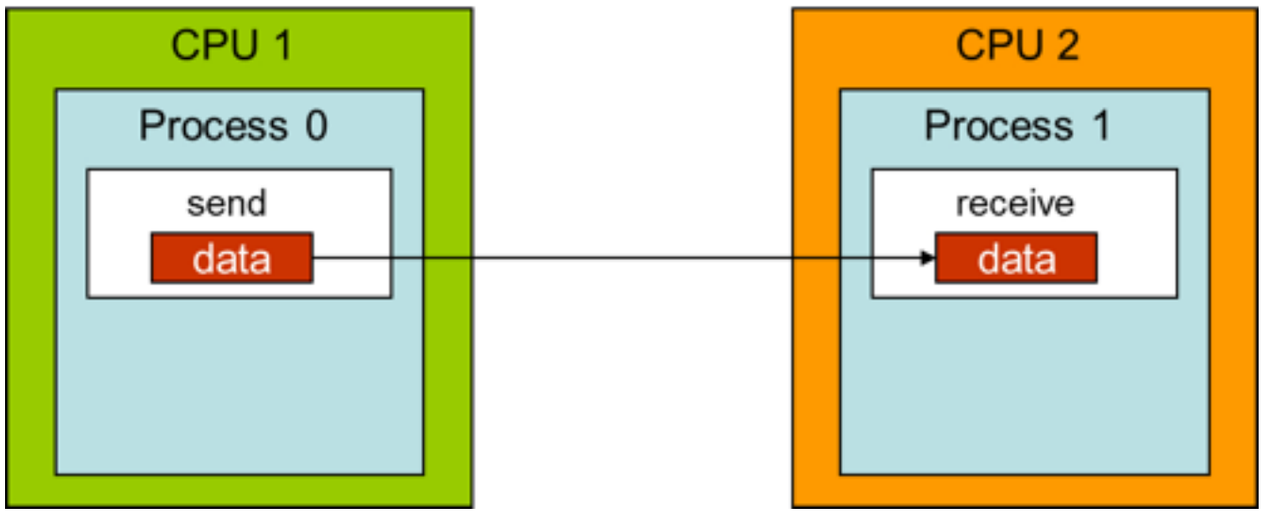
OpenMP Application (C++)

```
int main(int argc, char**argv) {
    if(argc>1)
        omp_set_num_threads(atoi(argv[1]));

    int i = 0;      // Loop counter
    int count = 0; // Number of successful trials
    int niter;      // Total trials
    unsigned seed; // To store the seed state of each random number

    niter = 1000000000;

    double x; // X coordinate
    double y; // Y coordinate

    double start_time;
    double end_time;

    start_time = omp_get_wtime();
    #pragma omp parallel private(i,x,y,seed)
    {
        seed = 25234 + 17*omp_get_thread_num();
        #pragma omp parallel for reduction(+:count) schedule(static)
        for(i=0; i < niter; i++) {
            x = (double)rand_r(&seed)/RAND_MAX;
            y = (double)rand_r(&seed)/RAND_MAX;
            if(x*x + y*y <= 1.0)
                count++;
            //printf("Thread %d, Iter: %d\n",omp_get_thread_num(),i);
        }
    }
    end_time = omp_get_wtime();

    // Compute Pi
    double pi = 4.0*(double)count/(double)niter;
    printf("# of trials= %d , estimate of pi is %g, time= %f \n",niter,pi, end_time-start_time);
    return 0;
}
```

MPI Basics

- Communicator
  - Rank - Process ID
  - Size - size of communicator
- MPI messages:
  - Are BETWEEN processes
  - Have:
    - type (e.g. data type)
    - tag (message ID)
    - source rank
    - target rank

MPI Send/Receive

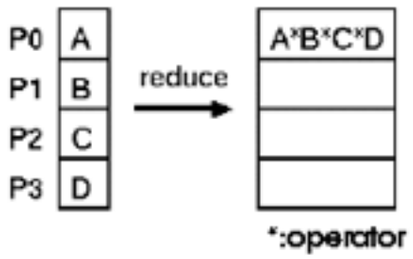


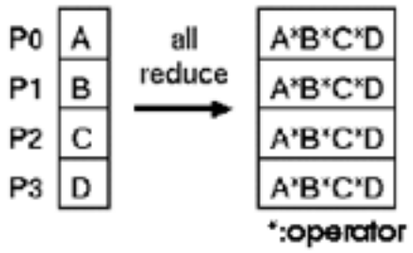
https://cvw.cac.cornell.edu/mpip2p/images/SimpleSendAndRecv.jpg

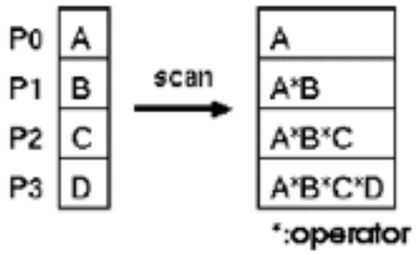
- Sends and receives can be synchronous, asynchronous, and/or immediate

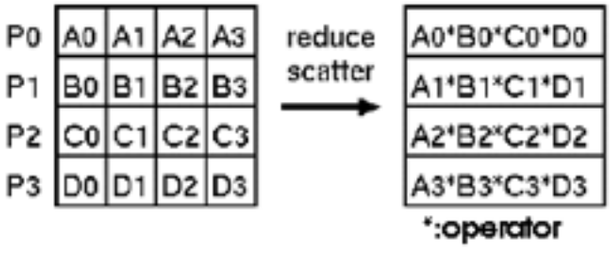
MPI Basics

MPI Reduce







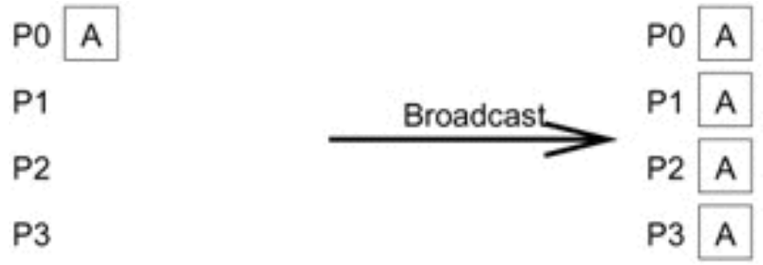


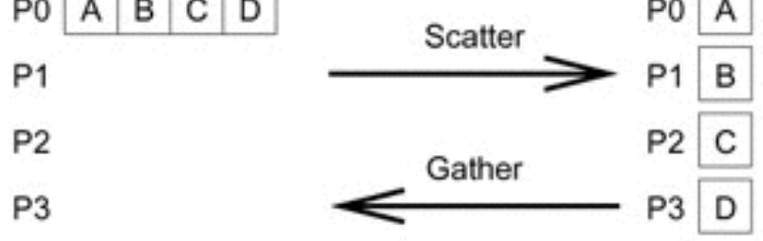
Operators

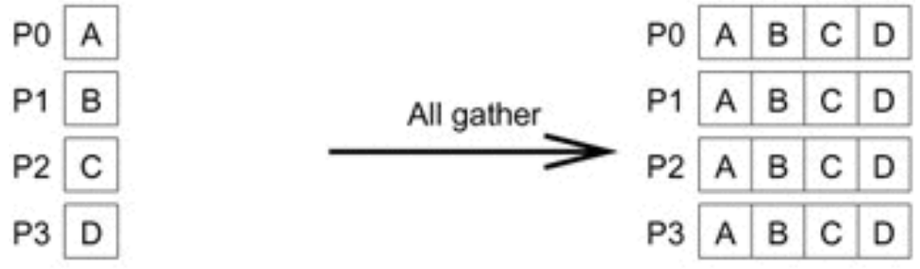
max
min
sum
product
logical

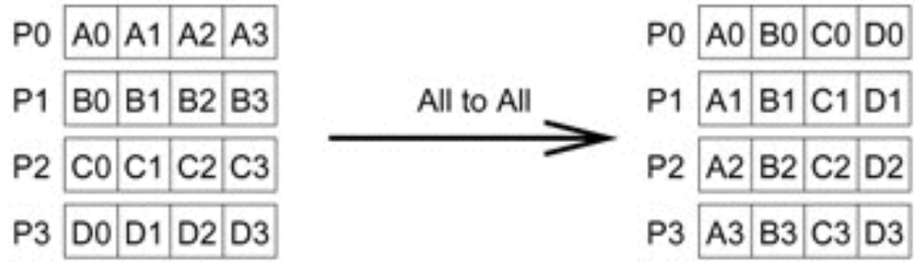
http://artemis.wszib.edu.pl/~jkocot/pwir/mpi/cp2.gif

MPI Collective Operations









http://users.dcc.uchile.cl/~ynakanis/pagina/collec~1.gif

Monte Carlo MPI Application (Python)

```
from mpi4py import MPI

# Global communicator
comm = MPI.COMM_WORLD
# Process rank and communicator size
rank = comm.Get_rank()
size = comm.Get_size()

def simulate(k1,k2,d,t):
# Set up initial conditions
if rank == 0:
    ICs = npr.poisson(k1/d,size)
else:
    ICs = None
# Send initial conditions to each rank
IC = comm.scatter(ICs, root=0)

# Run simulation
print("Running on rank: {rank} with initial count: {ic}".format(rank=rank, ic=IC))
soln = gillespie(IC, t, [k2,d])

# Gather results and return
soln = comm.gather(soln, root=0)

return ICs, soln
```

Finite Difference MPI Application (Python)

```
def heatEquation(a, T0, TL, TR, t, delt, ts, x, L):
    # Initialization
    dx = float(L)/float(x) # Width of cells
    dxdx = dx*dx
    cellsPerRank = int(x/size) # Count of cells per rank
    cellsPerRankTuple = size*[cellsPerRank]
    offsets = size*[0]
    for i,_ in enumerate(cellsPerRankTuple):
        if i == 0:
            continue
        offsets[i] = offsets[i-1] + cellsPerRankTuple[i-1]

    # Time variables
    tcur = 0.0
    timesteps = ts * np.arange(int(t/ts),dtype=np.float64)
    temperatures = []

    # Set up initial conditions
    if rank == 0:
        domain = np.full(x, T0, dtype=np.float64)
    else:
        domain = None

    # Set up local chunks for computation
    domainLocal_0 = np.zeros(cellsPerRank)
    domainLocal_1 = np.zeros(cellsPerRank)
    # Send initial conditions to each rank
    comm.Scatterv([domain, cellsPerRankTuple, offsets, MPI.DOUBLE], domainLocal_0, root=0)
```

Finite Difference MPI Application (Python)

```
lastSave = ts
while tcur <= t:
    # Update time
    tcur += delt
    lastSave += delt

    L = None # Left boundary
    R = None # Right boundary

    # Send left
    if rank > 0:
        comm.send(domainLocal_0[0], dest=rank-1, tag=2)
    else:
        L = TL
    if rank < size-1:
        R = comm.recv(source=rank+1, tag=2)

    # Send right
    if rank < size-1:
        comm.send(domainLocal_0[-1], dest=rank+1, tag=3)
    else:
        R = TR
    if rank > 0:
        L = comm.recv(source=rank-1, tag=3)

    # Compute heat flux
    for i in range(1,cellsPerRank-1):
        # Central difference
        dTdt = (domainLocal_0[i+1] - 2.0*domainLocal_0[i]+ domainLocal_0[i-1])/(dxdx)
        # Update temperature
        domainLocal_1[i] = a*delt*dTdt + domainLocal_0[i]

    # Fix boundaries
    domainLocal_1[0]  = a*delt*(domainLocal_0[1] - 2.0*domainLocal_0[0] + L) /(dxdx) + domainLocal_0[0]
    domainLocal_1[-1] = a*delt*(R - 2.0*domainLocal_0[-1] + domainLocal_0[-2])/(dxdx) + domainLocal_0[-1]
```

Finite Difference MPI Application (Python)

```
    # Save if necessary
    if lastSave > ts:
        lastSave = 0.0
        domainSave = np.zeros(x)
        comm.Gatherv(domainLocal_1,[domainSave,cellsPerRankTuple, offsets, MPI.DOUBLE],root=0)
        if rank == 0:
            print("Time:",tcur,"s",int(tcur/delt),"of",int(t/delt))
            temperatures.append(domainSave)

        # Swap array pointers
        tmp = domainLocal_0
        domainLocal_0 = domainLocal_1
        domainLocal_1 = tmp

        comm.Barrier()

    # Return solutions
    return timesteps, temperatures
```

```
mpirun -np 4 ./heatEqnMPI.py -a 0.01 -T0 300 -TL 350 -TR 310 -t 100.0 -tdel 0.001 -ts 2.0 -x 120 -L 1
```

Parallelism Frameworks

Parallelism/Multiphysics/ PDE Frameworks

- BoxLib
- Cactus
- Charm++
- Chombo
- deal.II
- FEniCS
- MOOSE
- Uintah
- Wasatch