EU-ASEAN HPC School 2022

5-10 December 2022
Kasetsart University, Bangkok, Thailand

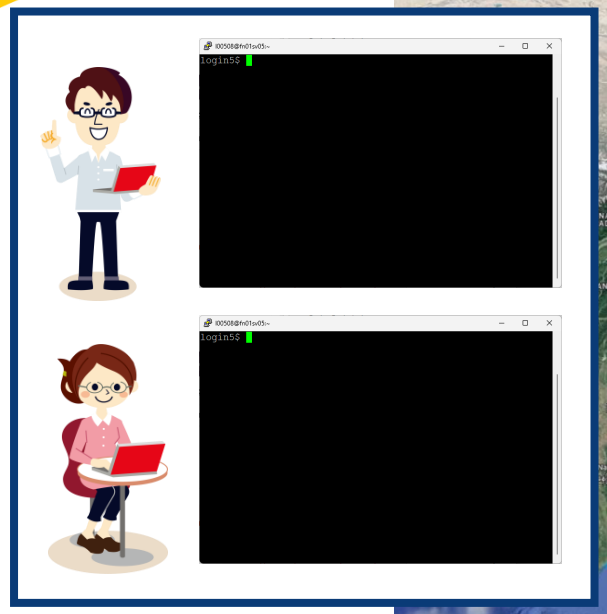
Introduction of Parallel Programming using Fugaku

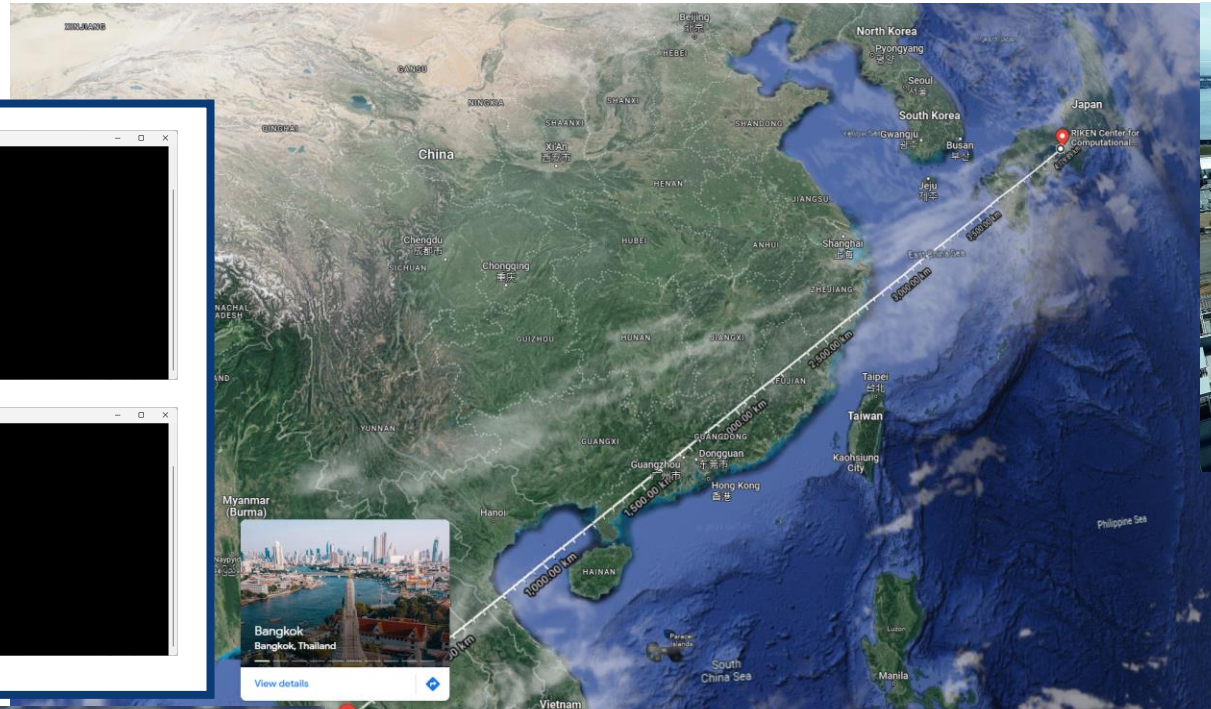
Jorji Nonaka (RIKEN R-CCS)

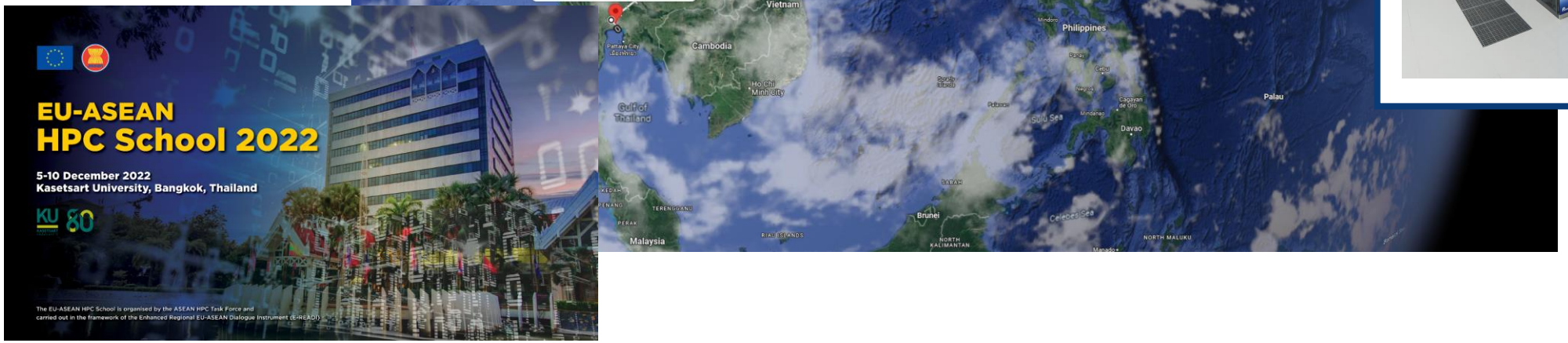
The EU-ASEAN HPC School is organised by the ASEAN HPC Task Force and carried out in the framework of the Enhanced Regional EU-ASEAN Dialogue Instrument (E-READI)

EU-ASEAN
HPC School 2022

RIKEN
R-CCS
RIKEN
Center for
Computational Science







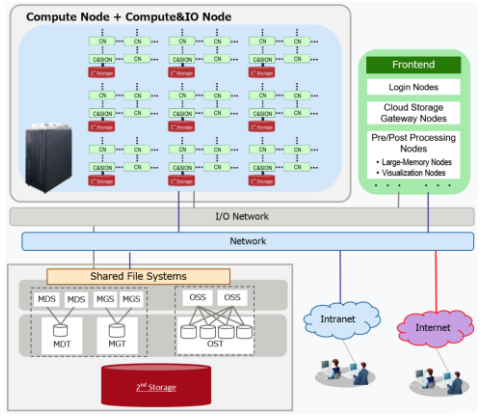
富岳
Fugaku

Users

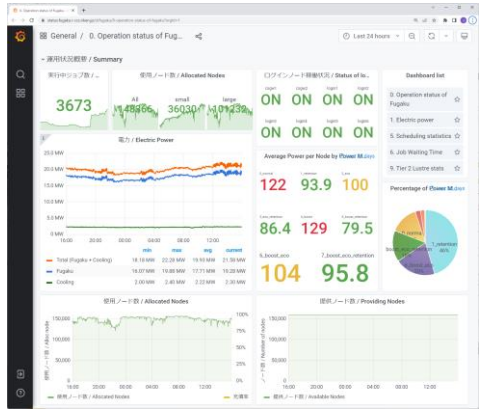
- Scientists
- Engineers
- …

Operations and Computer Technologies Div.

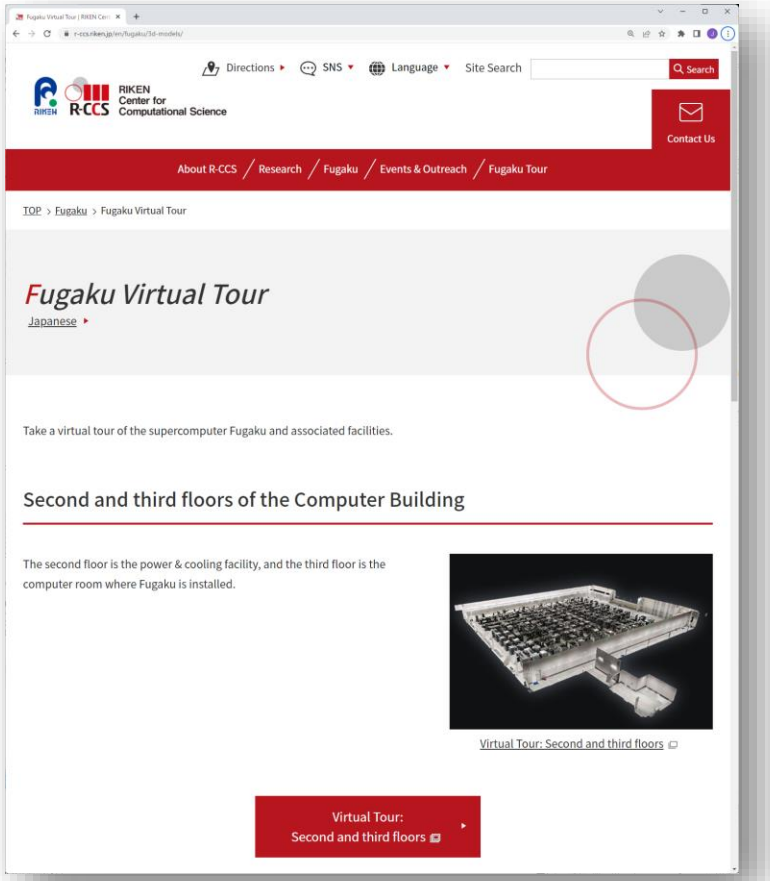
- Power System
  - Substation
  - Power Generator
- Cooling System
  - Air
  - Water

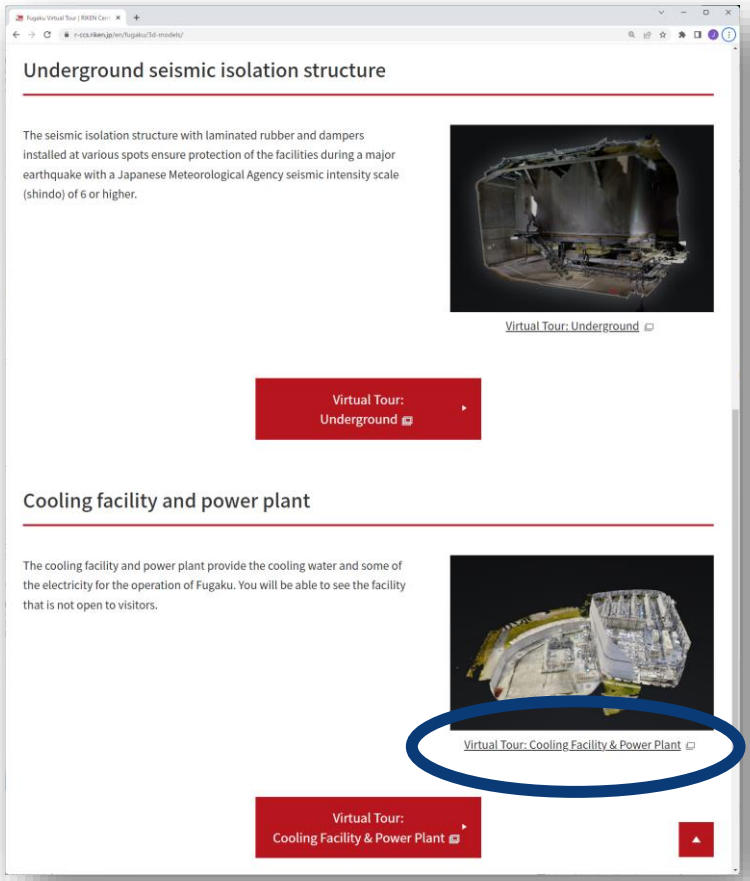


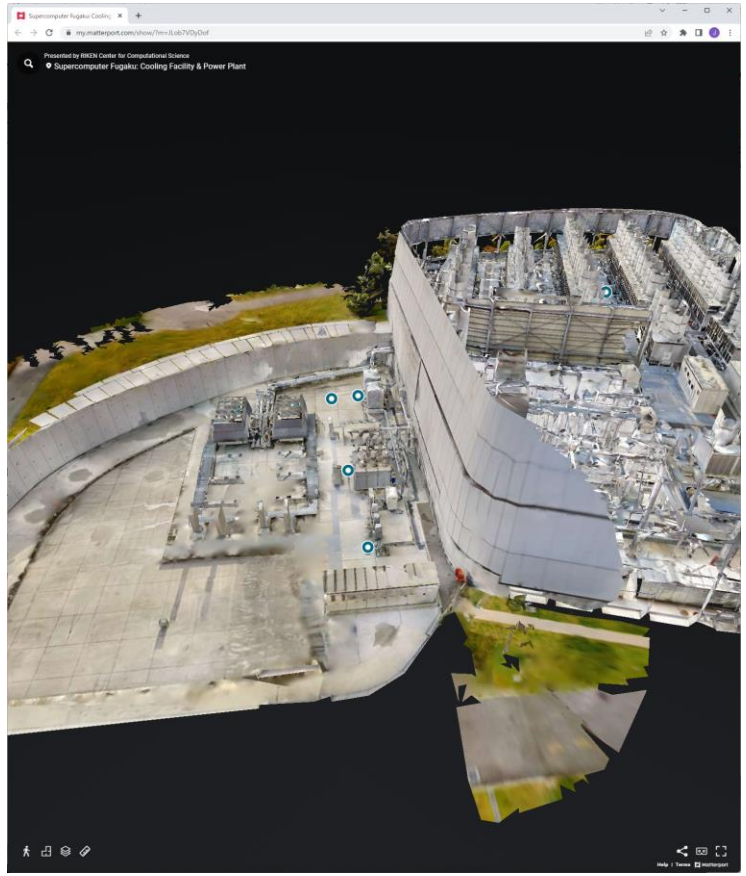
- Management
  - Account
  - Hardware
  - Software
  - Storage
  - …



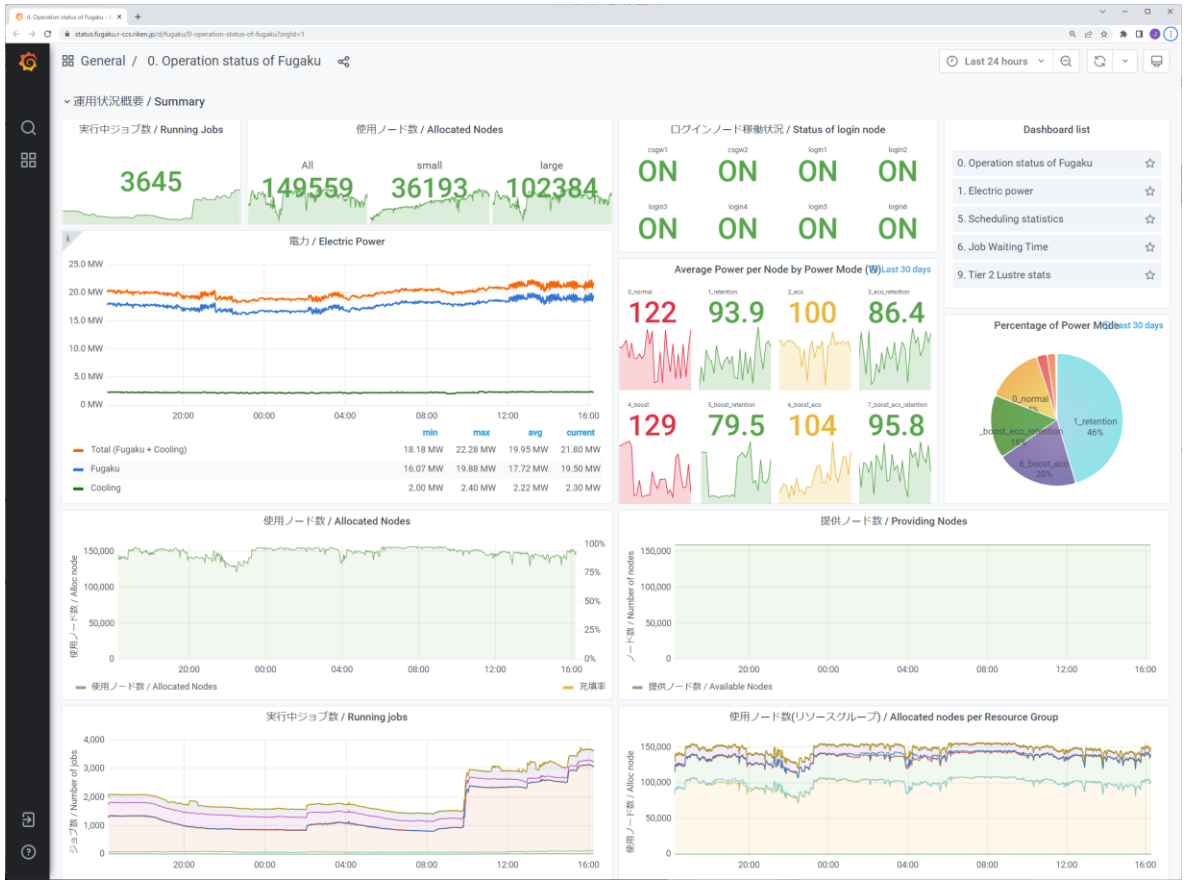
https://www.r-ccs.riken.jp/en/fugaku/3d-models/







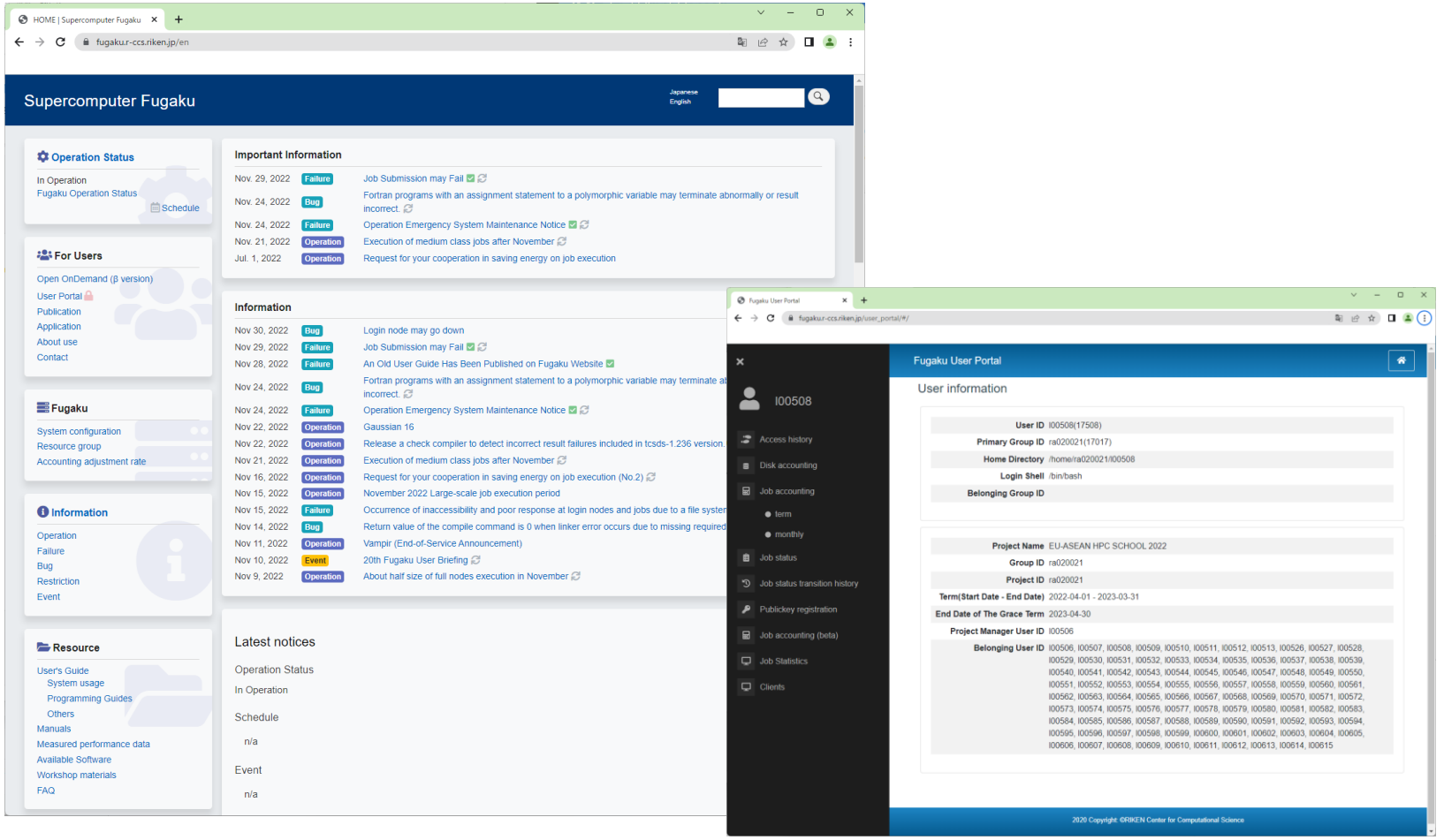
https://status.fugaku.r-ccs.riken.jp/

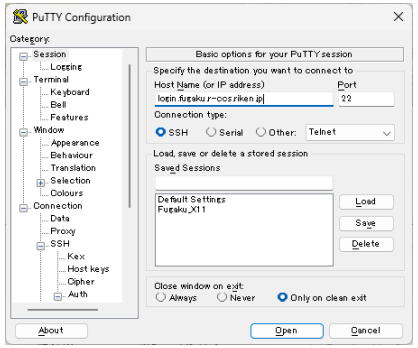


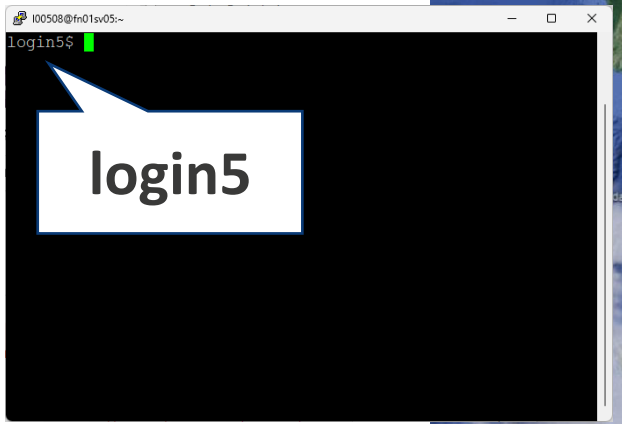
https://fugaku.r-ccs.riken.jp/

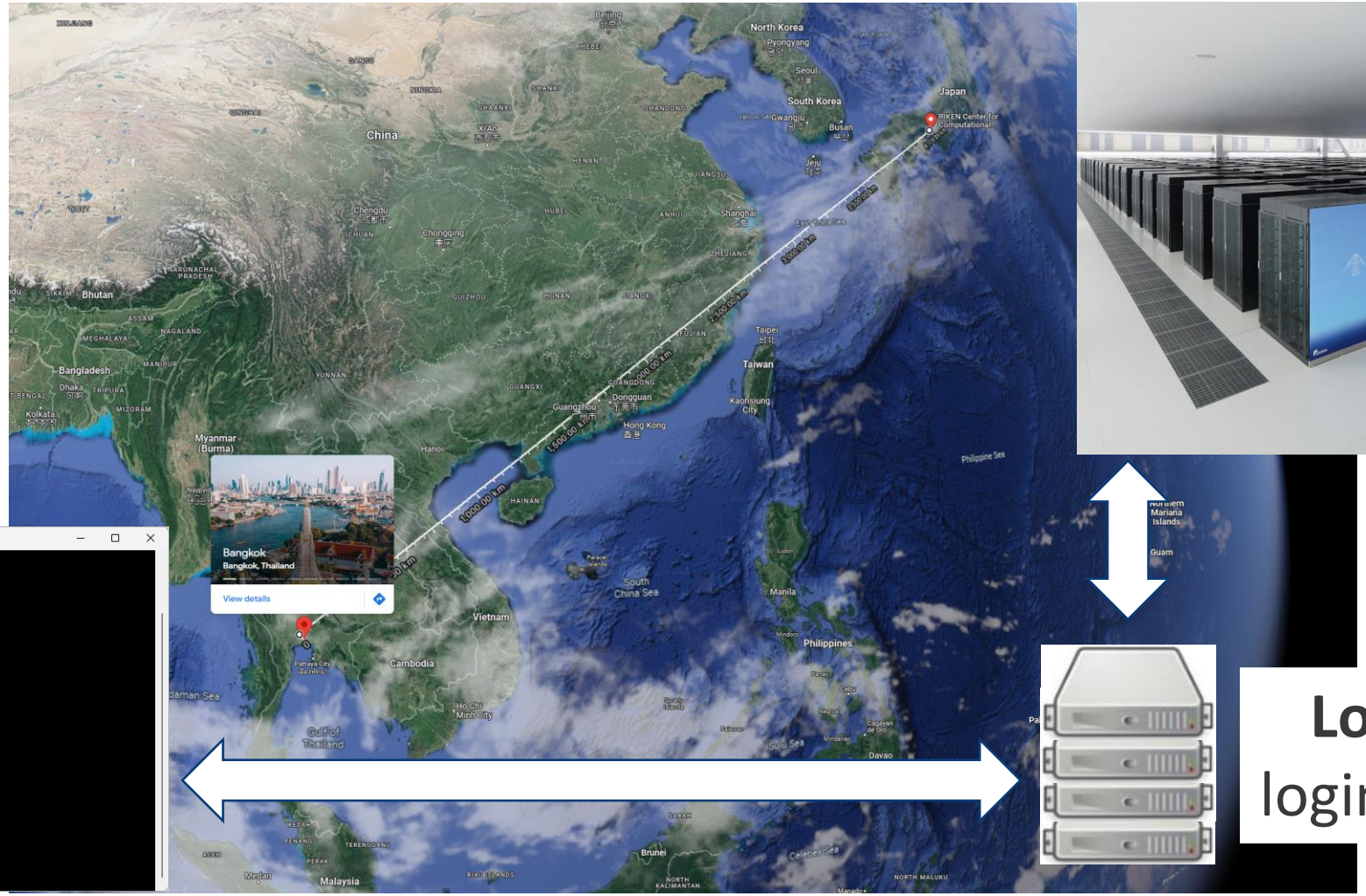
Fugaku Web Page

- User Portal
- User's Guide
- Operation Status
- Information
- …



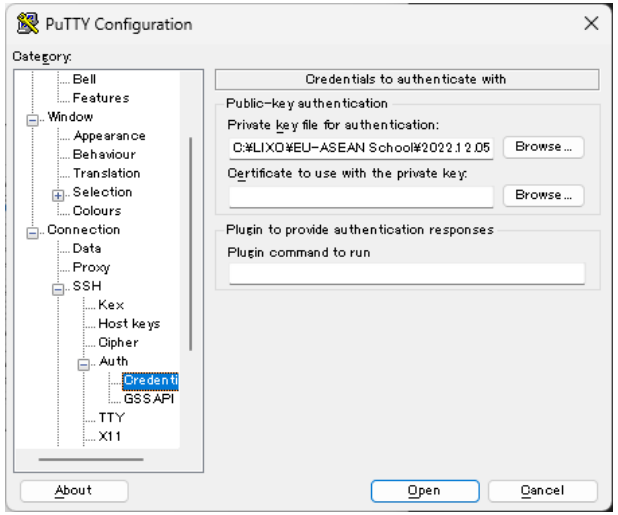






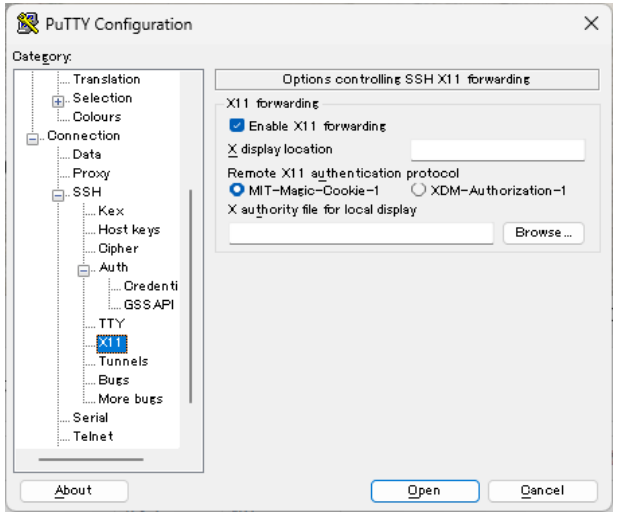
Login Nodes
login1, …, login6

PuTTY (SSH Client)



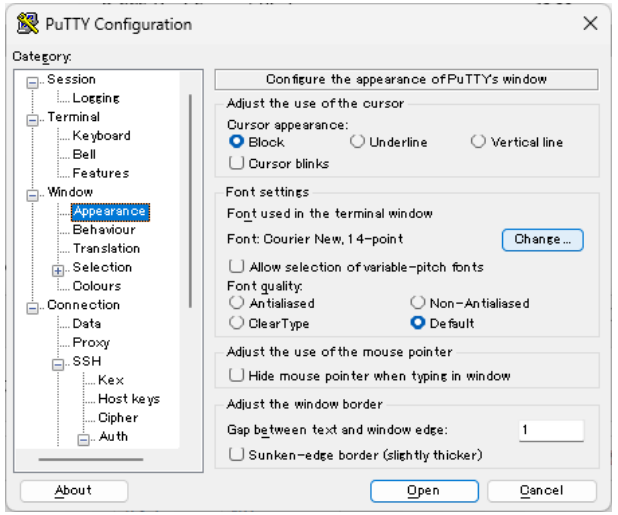
Private Key Setting

- Connection
  - SSH
    - Credentials



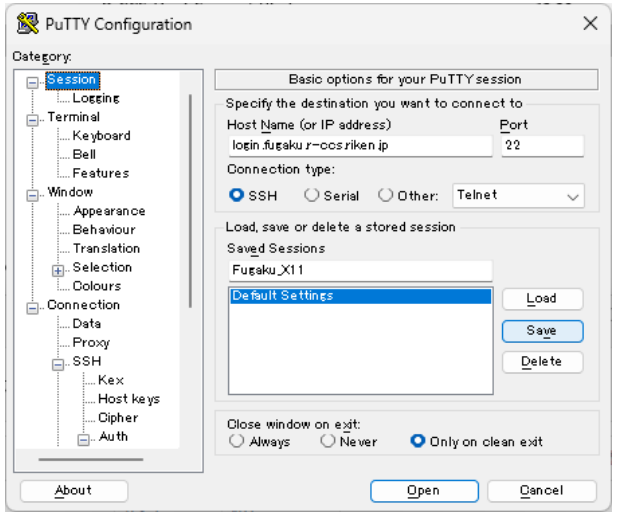
X11 Forwarding

- Connection
  - SSH
    - Auth
      - X11



Appearance (Font settings)

- Window
  - Appearance

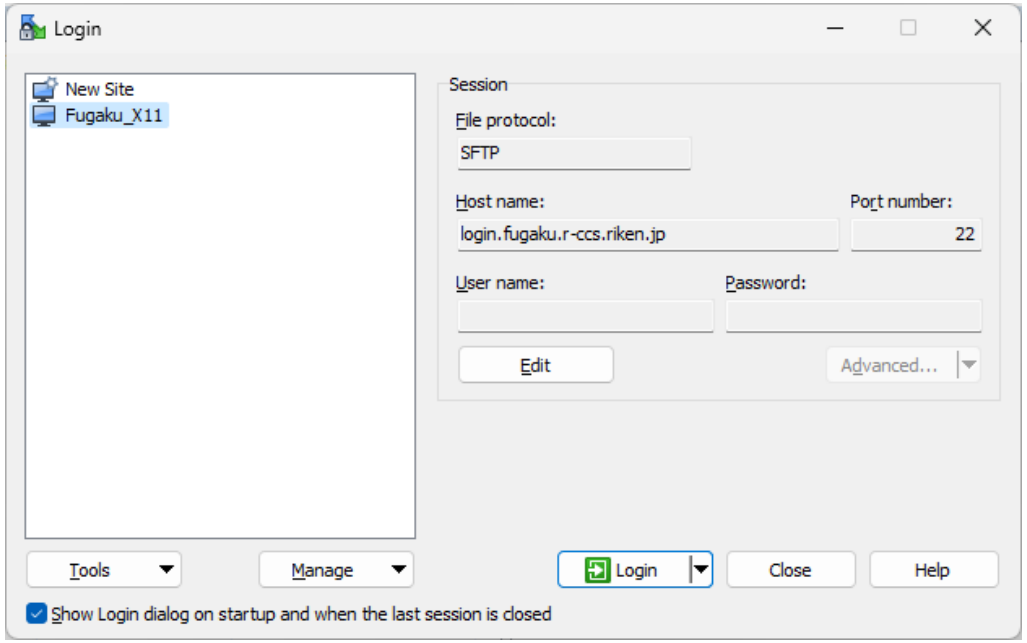


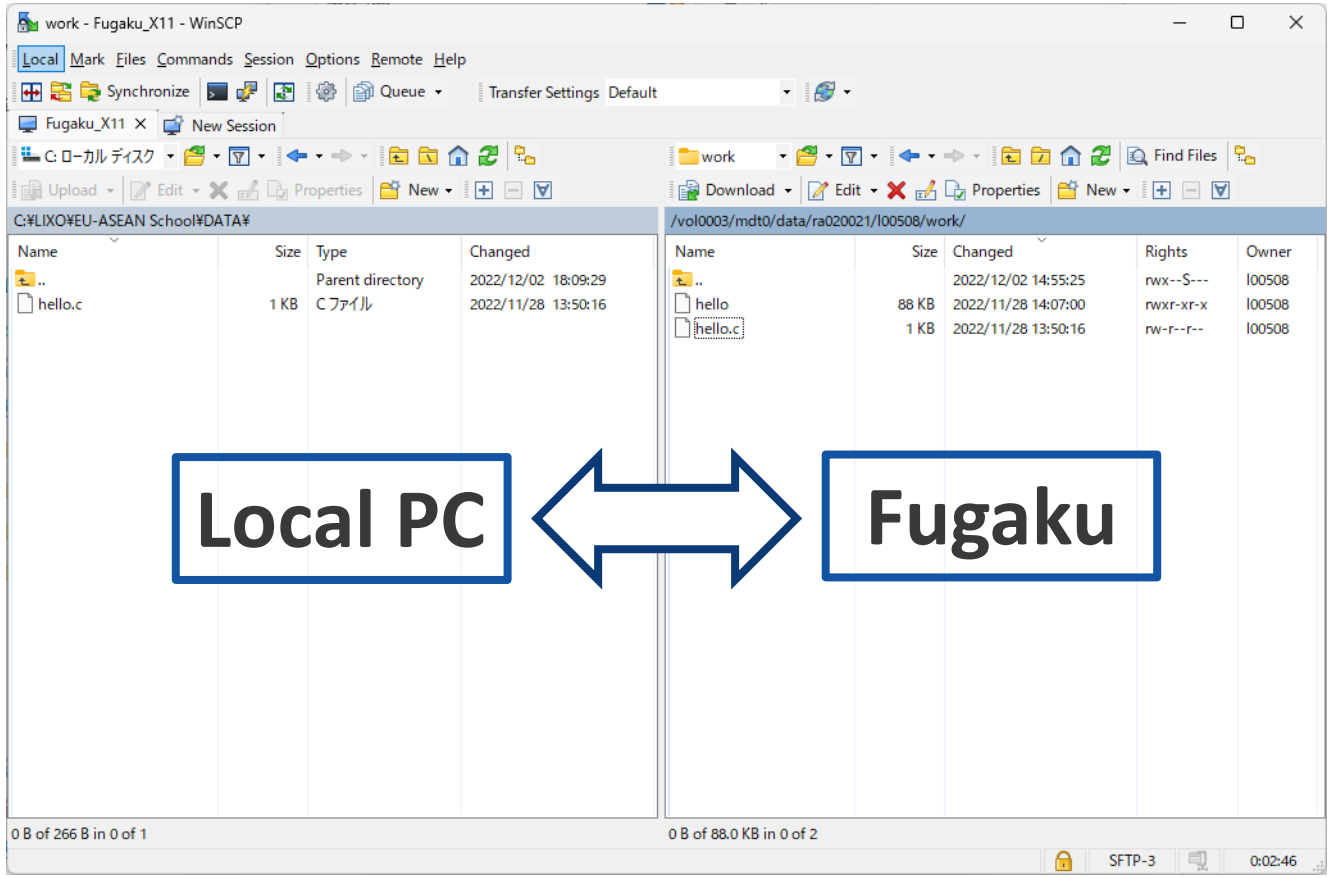
Save Session (Host Name)

- Session

WinSCP (FTP Client)

Supercomputer Fugaku
Startup Guide Version 1.06



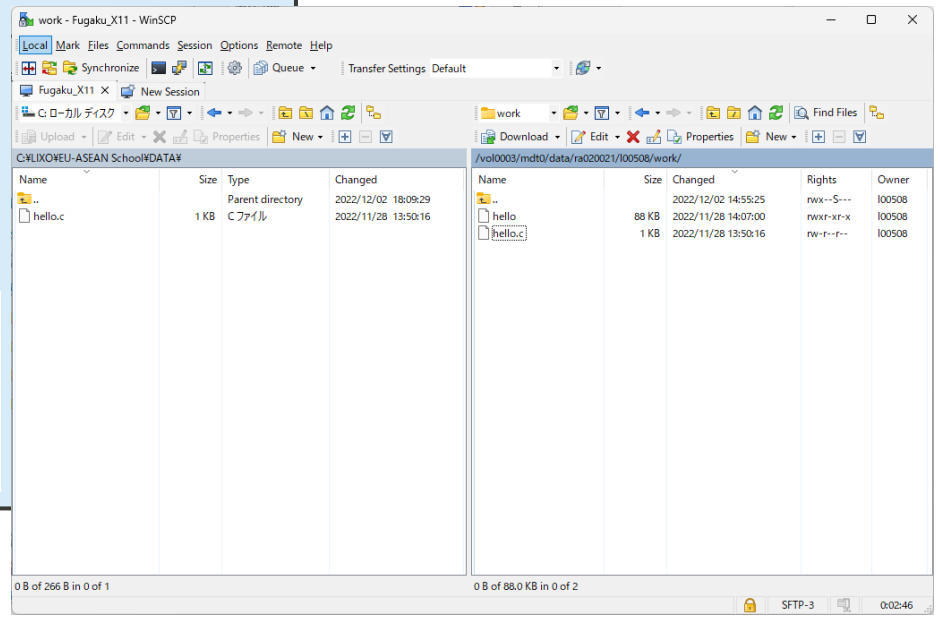


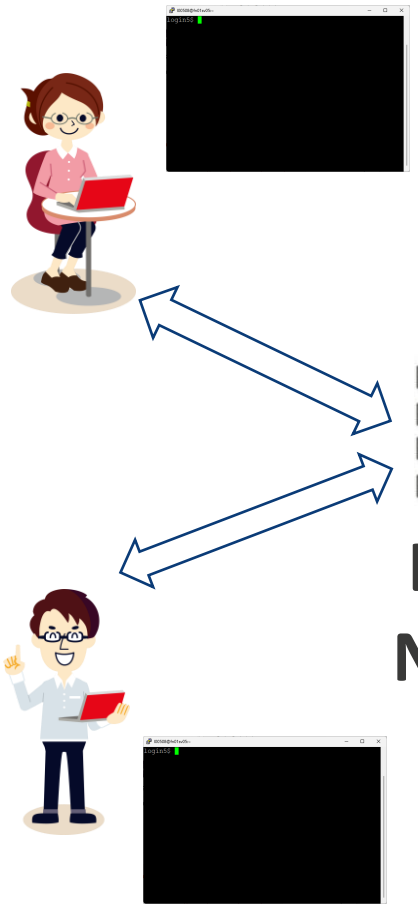
Lecture Slides and Hands-on Data

```
$ cd /vol0601/share/ra020021/
$ cd ComputerScience/20221205_Nonaka
$ ls
examples slides

$ cp –r examples ~/
$ cp –r slides ~/
```

Home Directory (Fugaku)

Your PC





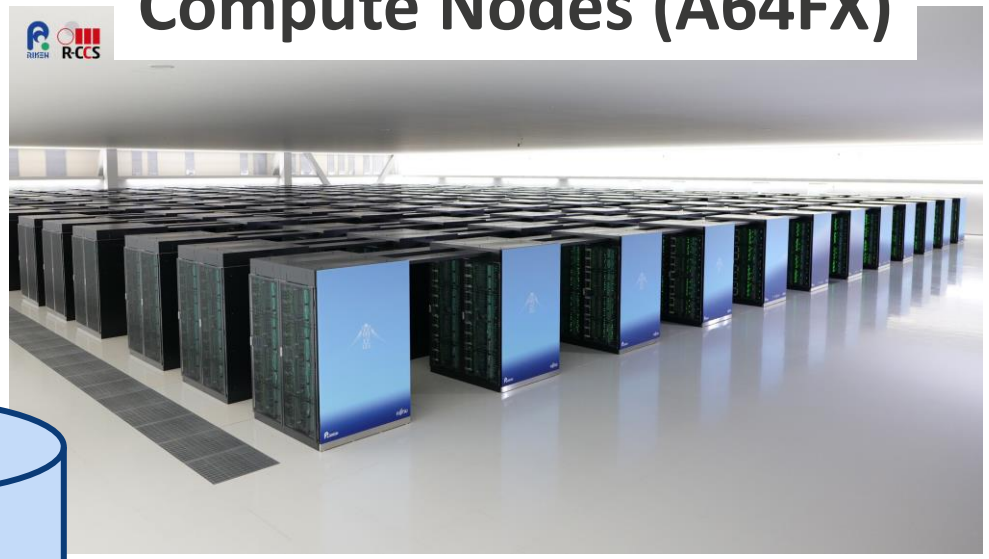
Login Nodes

• Batch Job
• Interactive Job

Storage

• Batch Job
• Interactive Job

Compute Nodes (A64FX)



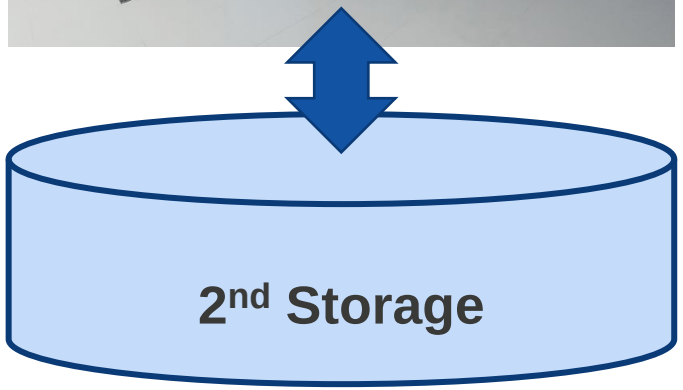
Pre/Post Environment (x86 CPU / GPU)

- Large-Memory Nodes
- GPU Nodes

Pre/Post Environment Users Guide

$ accountd -E

```
l00508@fn01sv05:~
login5$ accountd -E
COLLECTDATE : 2022/12/02 13:01:24    unit[GiB]
USER : l00508
*----------------------------------------------------[GROUP]-----------------------------------------------------*
GROUP           VOLUME                          LIMIT            USAGE        AVAILABLE          FILES  USE_RATE
*ra020021       vol0001                         5,120                1            5,119             15      0.0%
*ra020021       vol0601                         5,120               15            5,105        232,237      0.3%
  /vol0601/data/ra020021
  /vol0601/share/ra020021
*----------------------------------------------------[USER]------------------------------------------------------*
USER            VOLUME                          LIMIT            USAGE        AVAILABLE          FILES  USE_RATE
l00508          vol0300                            20                1               19            331      0.1%
  /vol0300/data/ra020021/l00508
login5$
```



- Home Directory (20 GB limit)
  - **/vol0003**/mdt0/data/**ra020021**/lxxxxx
- Data and Share Directory (5TB limit each)
  - **/vol0006**/mdt1/data/**ra020021**
  - **/vol0006**/mdt1/share/**ra020021**

Spack

- Spack
  - **/vol0004**

Users Guide – Use and job execution

Interactive Job

```
$ pjsub -x PJM_LLIO_GFSCACHE=/vol0004:/vol0006 --interact …
```

Batch Job

```
$ pjsub batch_job.sh
```

Batch Script: batch_job.sh

```
#PJM -g ra020021
#PJM –L “rscgrp=excl_HPCS_2212-1”
#PJM -x PJM_LLIO_GFSCACHE=/vol0004:/vol0006
…
```

Fugaku Spack User Guide

```
$ . /vol0004/apps/oss/spack/share/spack/setup-env.sh
```

Pre-built packages

```
$ spack find -x
```

Loading packages

```
$ spack load NAME
```

```
$ spack load NAME arch=xxx
$ spack load /HASH
```

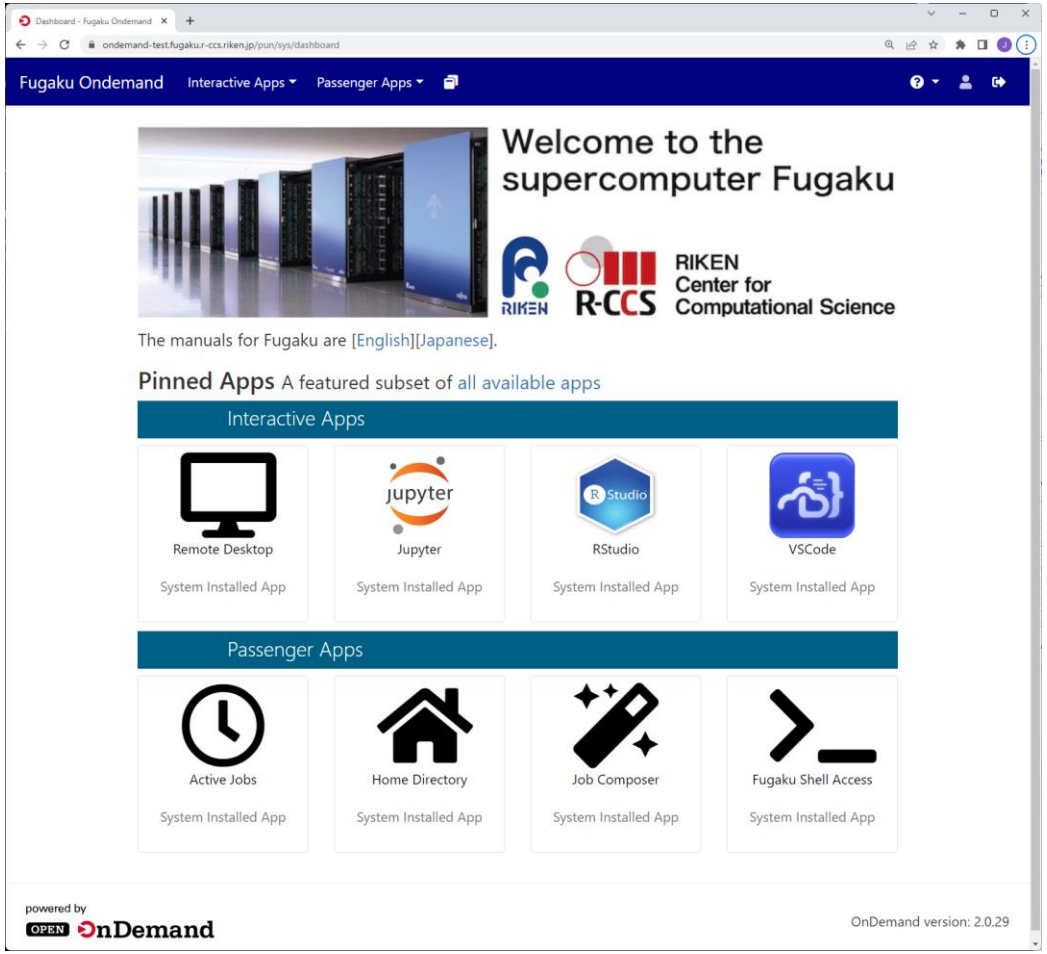
```
l00508@fn01sv05:~
login5$ . /vol0004/apps/oss/spack/share/spack/setup-env.sh
login5$ spack find -x
==> 186 installed packages
-- linux-rhel8-a64fx / fj@4.7.0 ---------------------------------
abinitmp@1-22          grads@2.2.1          n2p2@2.1.4           picard@2.26.2          py-toml@0.10.2
abinitmp@2-4           gromacs@2020.6       netcdf-c@4.8.1       povray@3.7.0.8         py-xarray@0.18.2
adios2@2.7.1           gromacs@2021.4       netcdf-cxx@4.2       py-ase@3.21.1          python@3.8.12
bcftools@1.12          gromacs@2021.5       netcdf-cxx4@4.3.1    py-dask@2021.6.2       quantum-espresso@6.5
bedtools2@2.30.0       hdf5@1.10.7          netcdf-fortran@4.5.3 py-h5py@3.4.0          quantum-espresso@6.6
biobambam2@2.0.177     hdf5@1.10.7          netlib-lapack@3.9.1  py-jupyterhub@0.9.4    quantum-espresso@6.7
blitz@1.0.1            htslib@1.12          netlib-scalapack@2.1.0 py-keras@2.4.3       quantum-espresso@6.8
boost@1.77.0           julia@1.6.3          ntchem@12.2.0        py-matplotlib@3.4.3    r@4.1.1
bwa@0.7.17             kim-api@2.2.1        openblas@0.3.18      py-mpi4py@3.1.2        raja@0.14.0
cblas@2015-06-06       kokkos@3.4.01        openfoam@2012        py-netcdf4@1.5.3       rdkit@2021_03_2
charmpp@6.10.2         lammps@20201029      openfoam@2106        py-numpy@1.21.3        rist-fftw@3.3.9-272-g63d6bd70
cmake@3.21.4           lis@2.0.27           openfoam-org@8       py-pandas@1.3.4        samtools@1.12
dssp@3.1.4             llvm@12.0.1          openjdk@11.0.12_7    py-pip@21.1.2          scale@5.4.3
eigenexa@2.6           mapsplice2@2.2.1     openmx@3.8           py-pydmd@0.3           screen@4.8.0
fujitsu-fftw@1.0.0     mesa@21.2.3          parallel-netcdf@1.12.2 py-pygps@1.3.5       siesta@4.0.1
fujitsu-frontistr@5.0  metis@5.1.0          paraview@5.9.1       py-pysam@0.15.3        star@2.7.6a
fujitsu-mpi@head       mpich-tofu@master    parmetis@4.0.3       py-scikit-learn@1.0.1  suite-sparse@5.10.1
fujitsu-ssl2@head      mpich-tofu@master    petsc@3.16.1         py-scipy@1.6.1         tmux@3.2a
genesis@1.5.1          mptensor@0.3.0       pfapack@2014-09-17   py-seaborn@0.11.2

-- linux-rhel8-a64fx / fj@4.8.0 ---------------------------------
abinit@8.10.2          fds@6.7.7            gromacs@2022.3       octa@8.4               py-numpy@1.21.3      smash@3.0.0
akaikkr@2002v010       fujitsu-fftw@1.1.0   h5z-zfp@1.0.1        onednn@2.5.2           py-numpy@1.21.3      tau@2.30.2
alamode@1.3.0          fujitsu-frontistr@5.0 hphi@3.5.0          openfdtd@2.6.0         py-pymol@2.5.0       wrf@3.9.1.1
batchedblas@1.0        genesis@1.5.1        improved-rdock@main  openfoam@2012          py-pytest@6.2.4      wrf@4.3.3
cp2k@7.1               genesis@2.0.0        libgridxc@0.8.5      openfoam@2106          py-spglib@1.16.1     xios@2.5
cp2k@9.1               genesis@2.0.0        libpsml@1.1.7        openfoam-org@8         py-tomli@1.2.1       xmlf90@1.5.2
darshan-runtime@3.3.1  genesis@2.0.0        metis@4.0.3          openmx@3.9.9           salmon-tddft@2.0.2   zpares@0.9.6a
darshan-runtime@3.3.1  gmake@4.2.1          modylas-new@1.1.0b   phase0@2021.01         scsumma25d@1.0a
darshan-util@3.3.1     gmt@6.2.0            mvmc@1.2.0           py-horovod@0.23.0      siesta@4.0.1
eigen@3.4.0            gromacs@2022         nwchem@master        py-hypothesis@6.23.1   siesta-MaX@1.0.1

-- linux-rhel8-a64fx / gcc@8.4.1 --------------------------------
fujitsu-mpi@head  gcc@10.3.0  gcc@11.2.0  mpich-tofu@master  mpich-tofu@master

-- linux-rhel8-a64fx / gcc@11.2.0 -------------------------------
fujitsu-mpi@head

-- linux-rhel8-cascadelake / gcc@11.2.0 -------------------------
boost@1.78.0        git-lfs@2.11.0  gnuplot@5.4.2  mercurial@5.8   netcdf-fortran@4.5.3  py-pip@21.1.2  tmux@3.2a
cmake@3.21.4        gmt@6.2.0       hdf5@1.10.7    ncview@2.1.8    ninja@1.10.2          python@3.8.12  xterm@353
darshan-util@3.3.1  gnuplot@5.4.2   libxml2@2.9.7  netcdf-c@4.8.1  openjdk@11.0.12_7     screen@4.8.0   zsh@5.8

-- linux-rhel8-skylake_avx512 / gcc@8.4.1 -----------------------
gcc@11.2.0  imagemagick@7.0.8-7  llvm@12.0.1  mesa@21.2.3  omni-compiler@1.3.3  paraview@5.9.1  py-numpy@1.21.3  py-phonopy@2.12.0
login5$
```

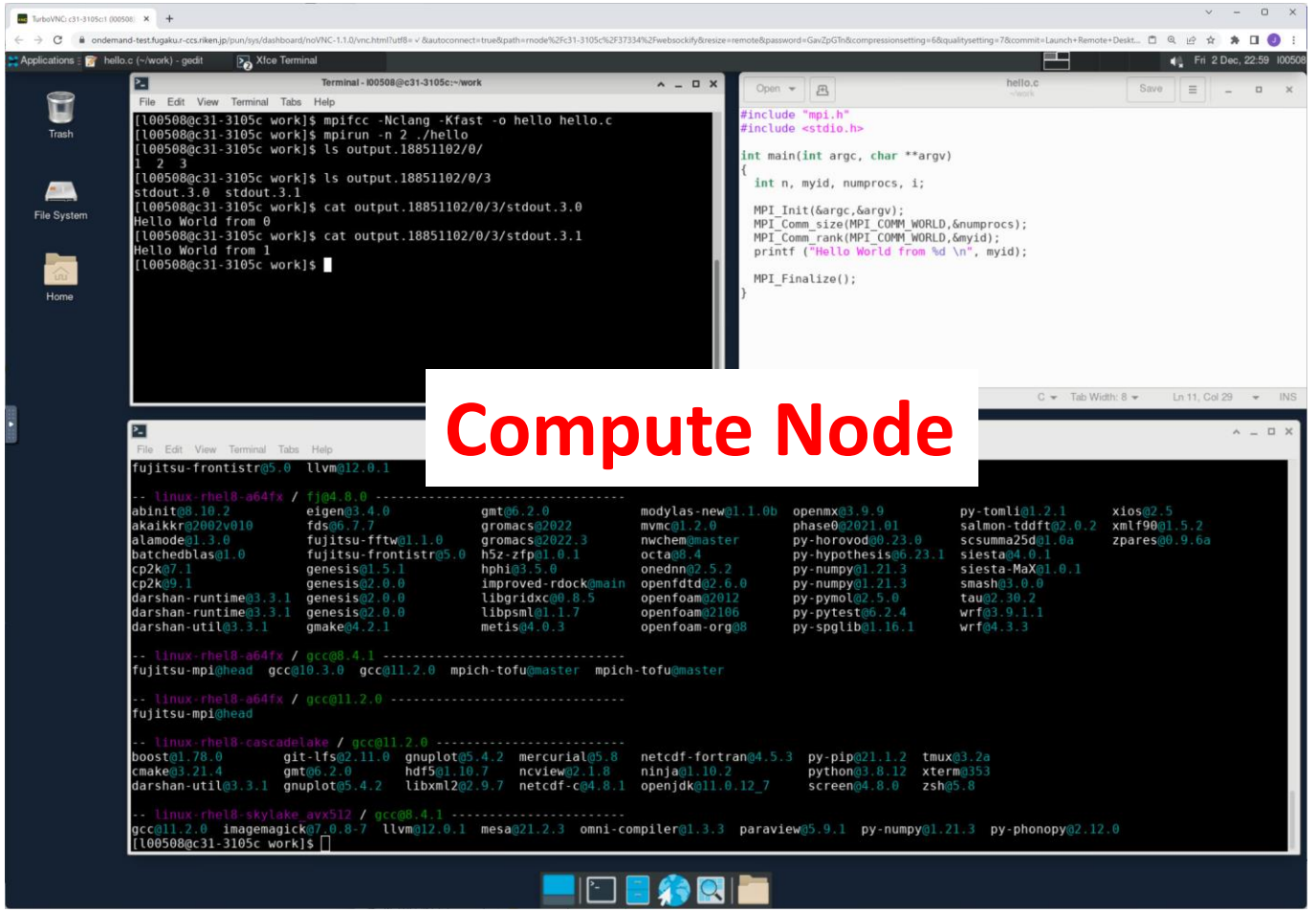
Compute Node
- **a64fx**

Pre/Post Env.
- **cascadelake**
- **skylake**

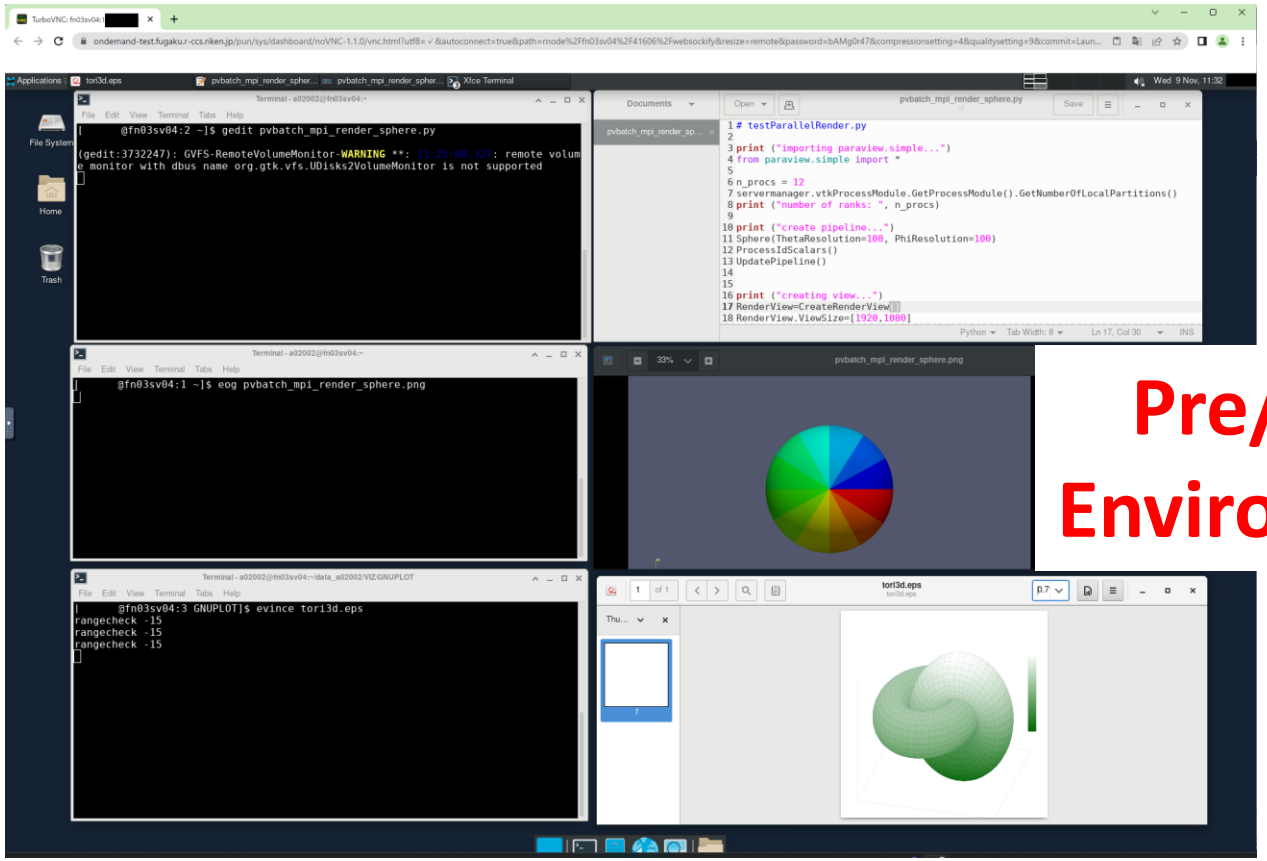
Fugaku Open OnDemand Guide

https://ondemand-test.fugaku.r-ccs.riken.jp

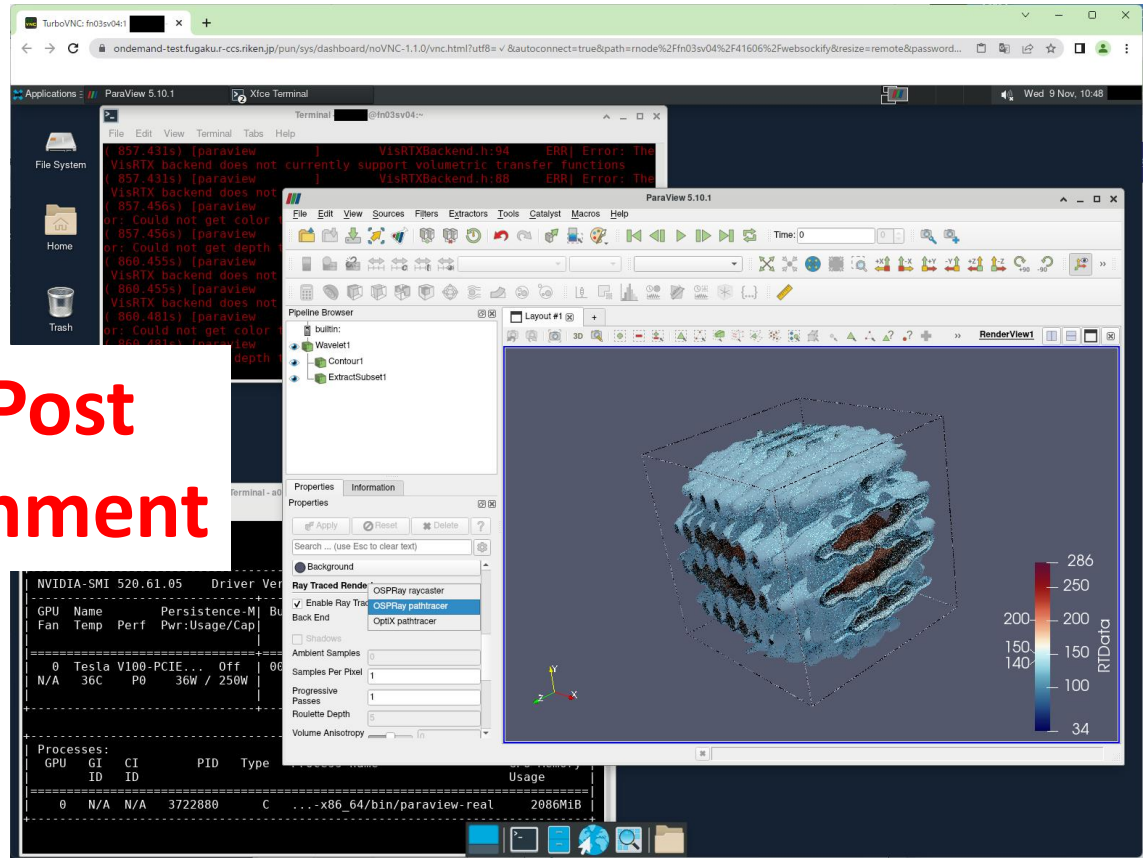




Visualization User Guide



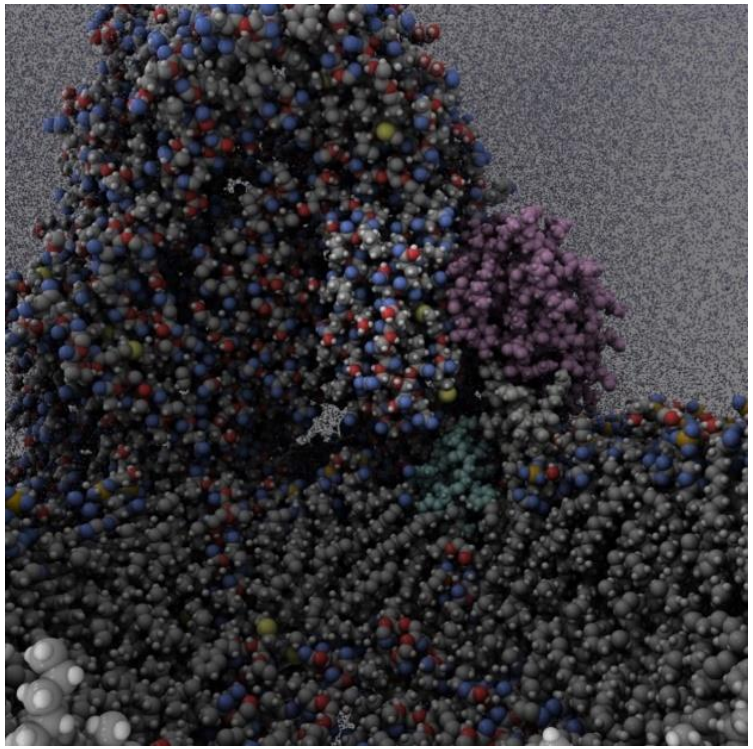
Pre/Post Environment

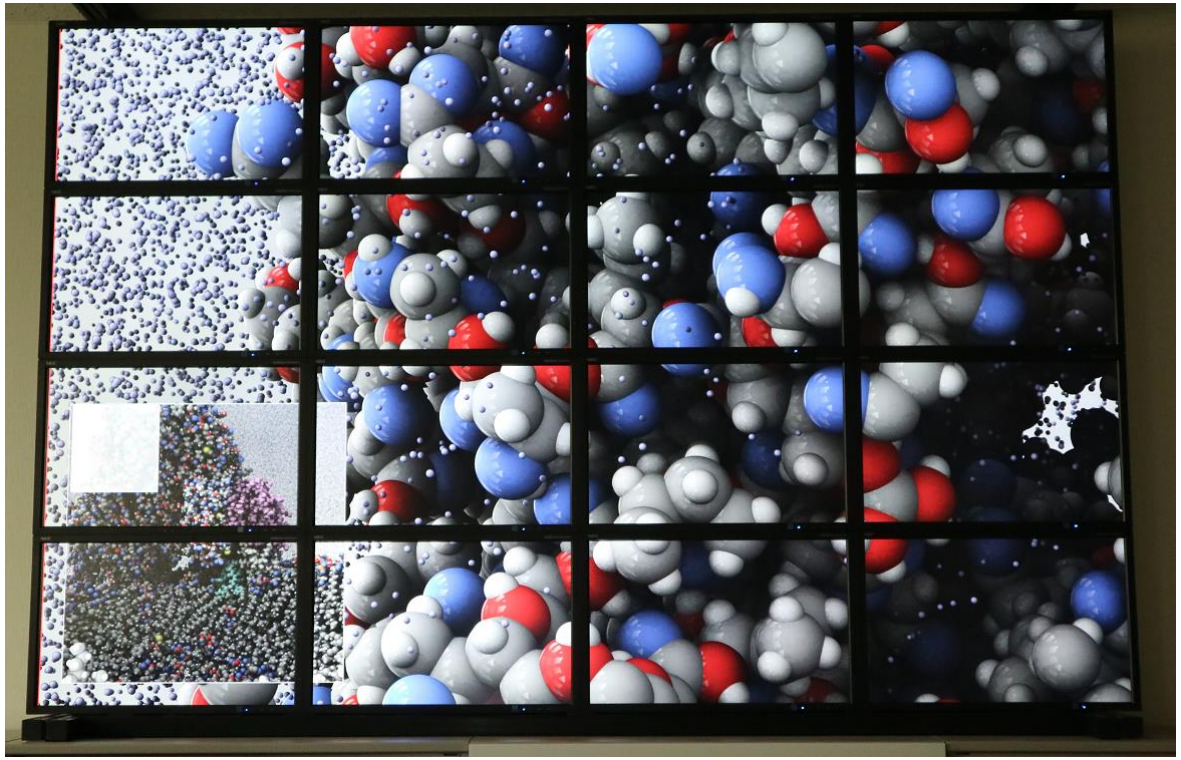


Parallel Processing

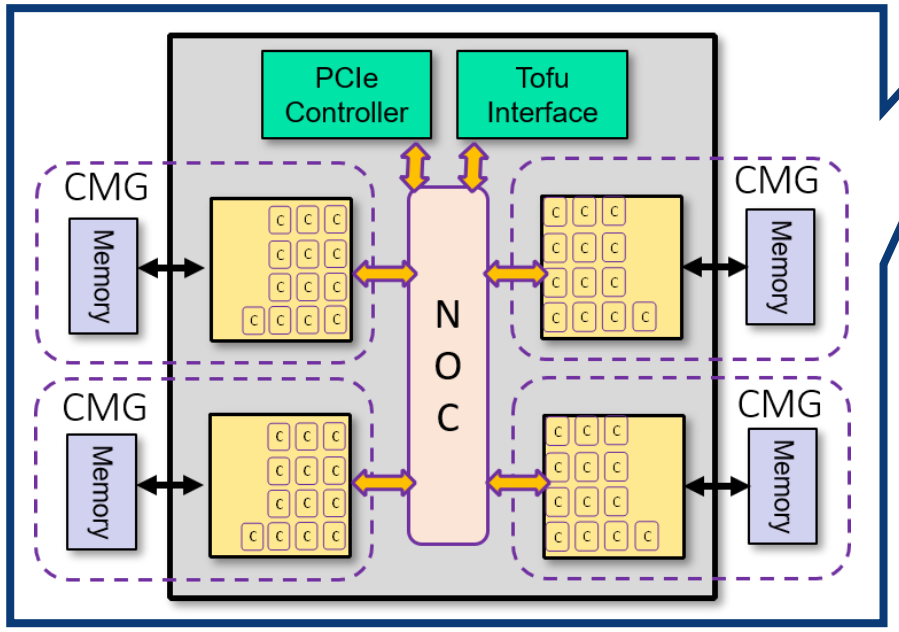
Divide-and-conquer

82,944 Nodes
- 288 x 288 (2D subdivision)
  - 64 x 64 pixels



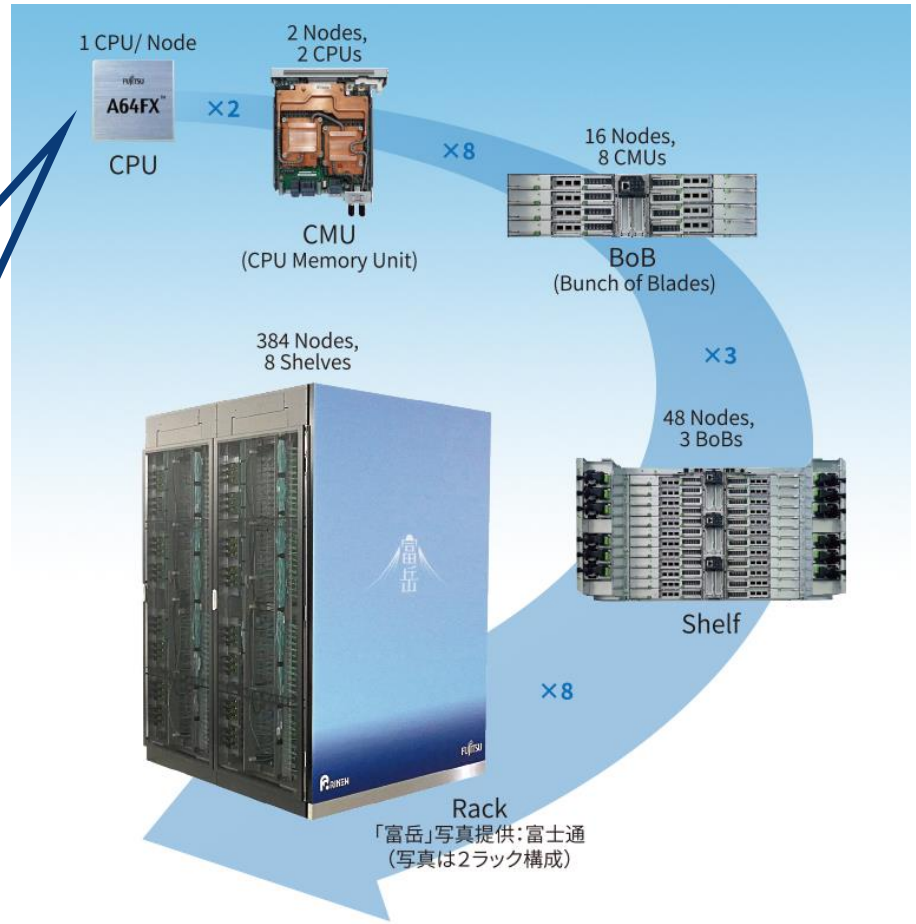


Shared Memory



48 Computer Cores per CPU

Distributed Memory



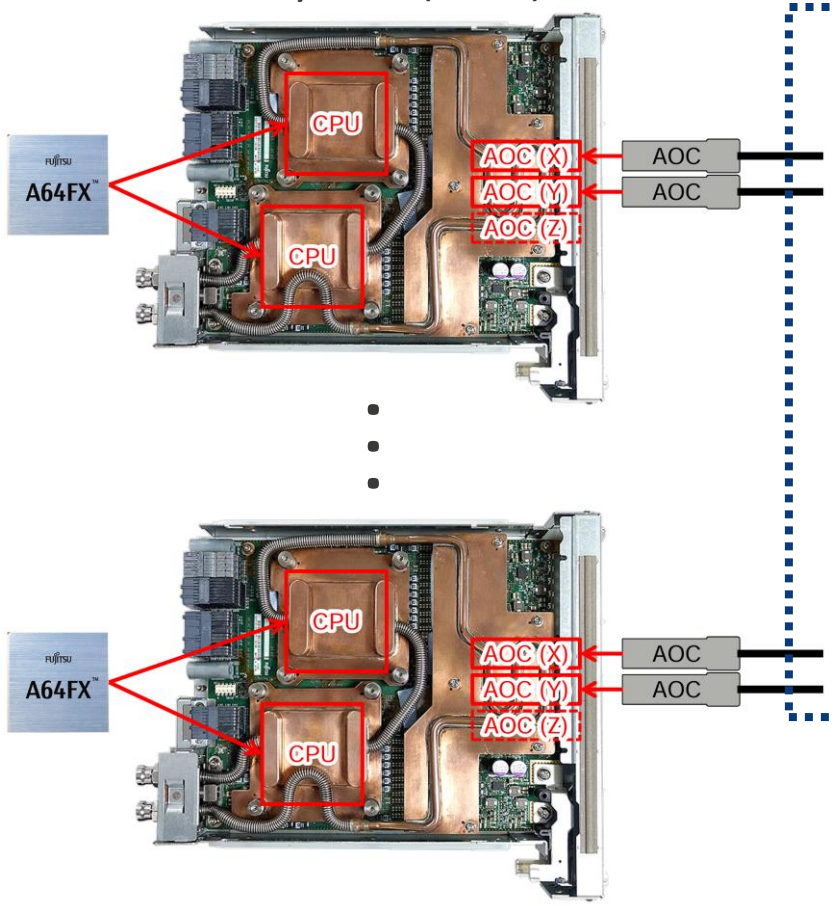
432 Racks
- 396 (Full)
- 36 (Half)

Max. 384 Nodes per Rack

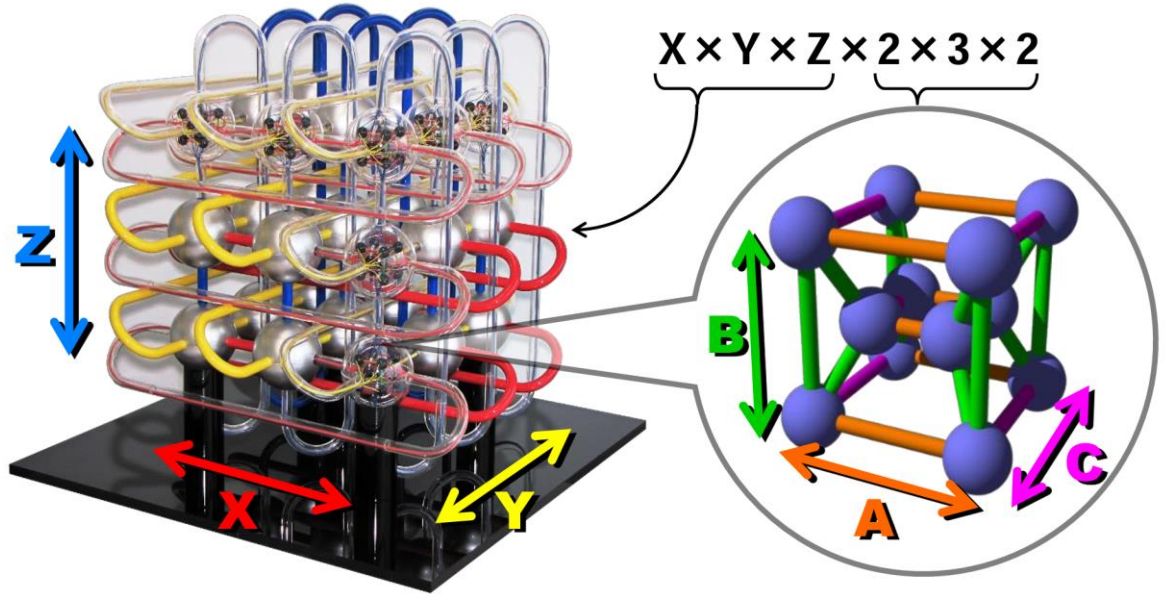
Distributed Memory

CPU Memory Unit (CMU)

Independent Nodes



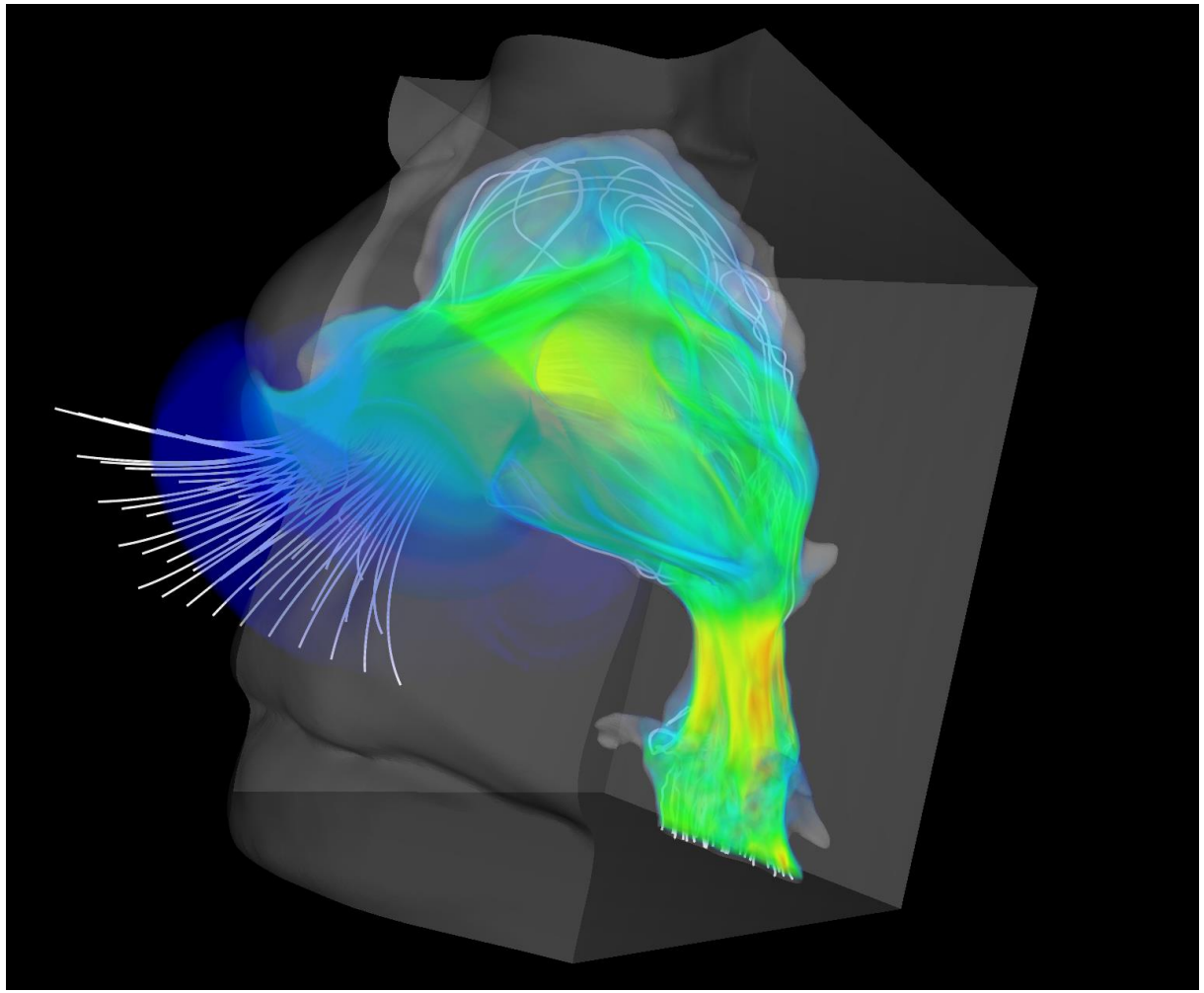
Independent Nodes



https://www.fujitsu.com/global/imagesgig5/the-tofu-interconnect-d-for-supercomputer-fugaku.pdf

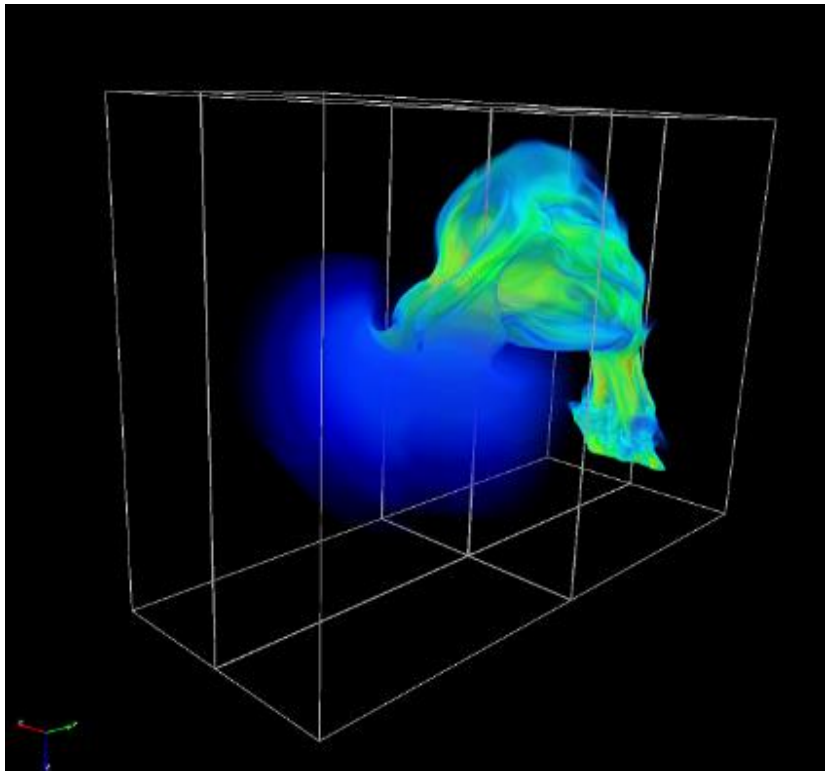
Shape (1D / 2D / 3D)

Rendering

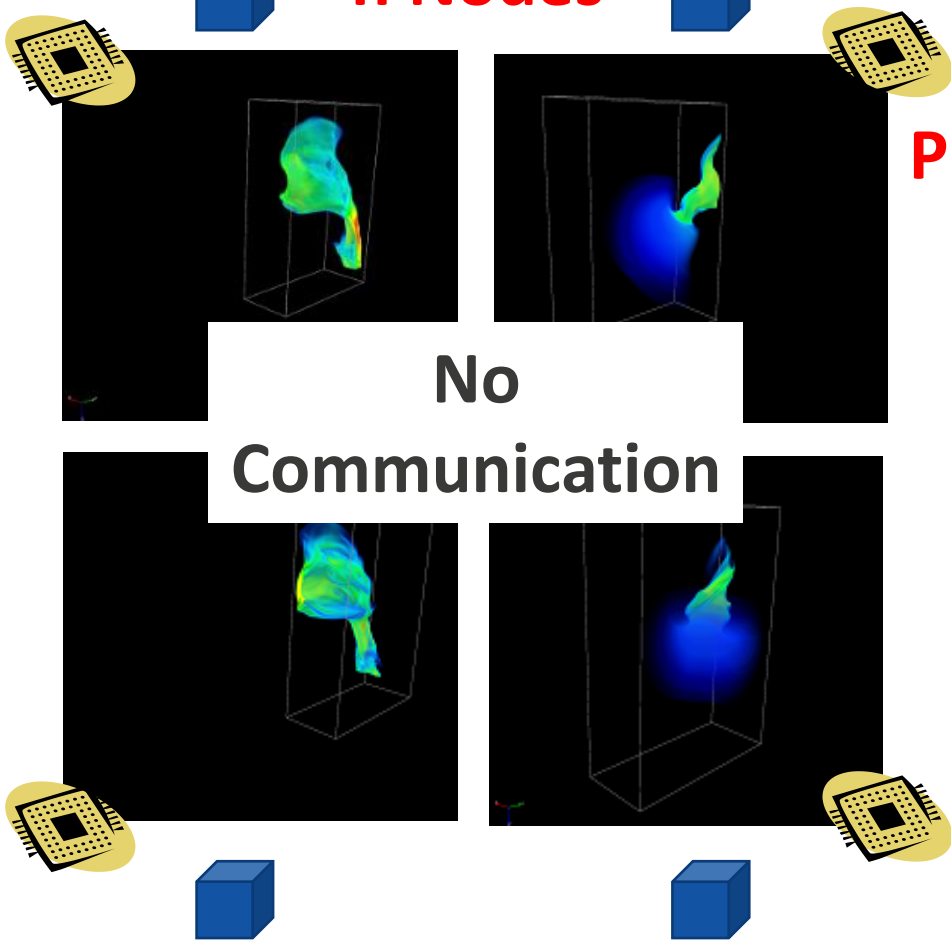


Parallel Rendering

1



n Nodes

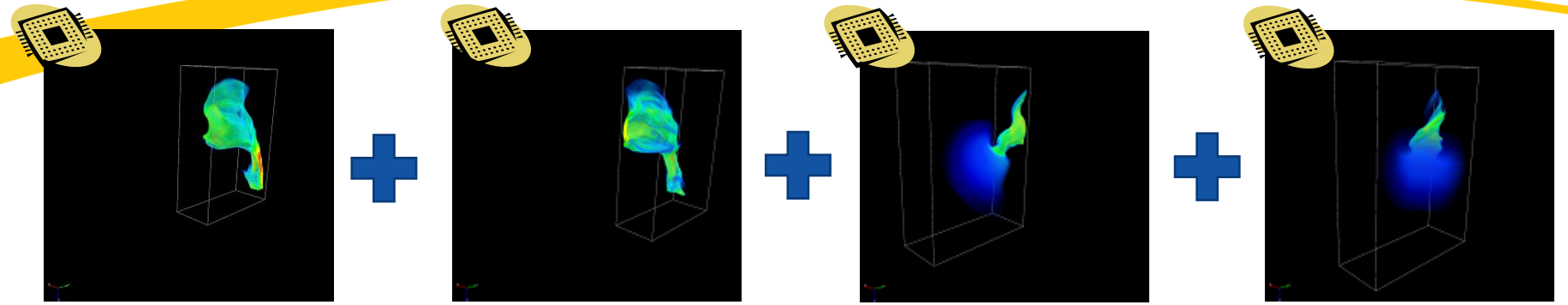


No Communication

Problem Size 1/n

Embarrassingly Parallel

Parallel Image Composition

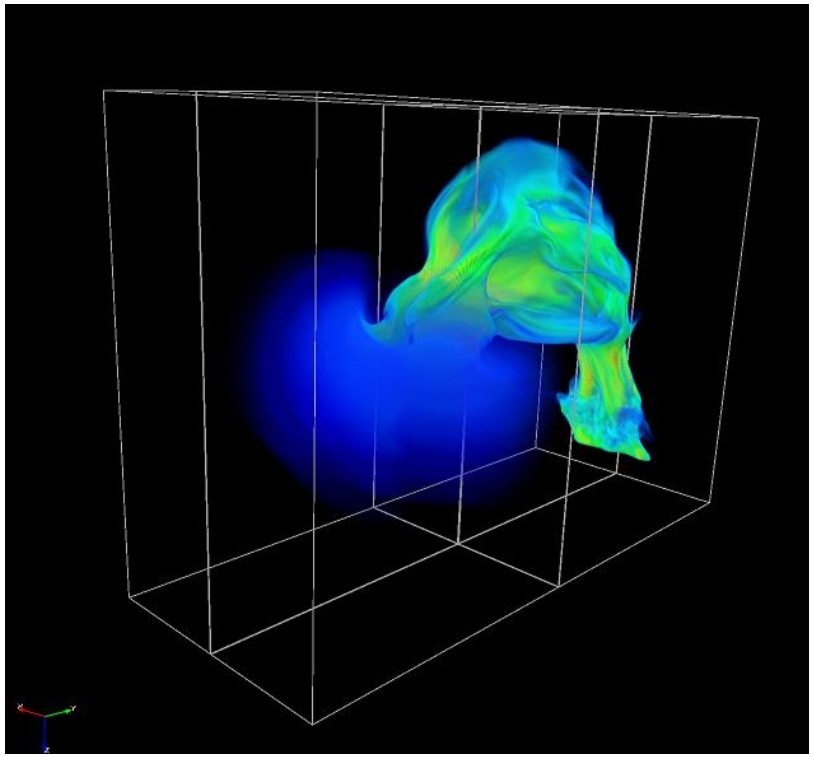


Problem Size 1 x n

1

n Nodes

Communication
(Send/Recv Image Data)



Computation
(Alpha Blending)

Message Passing Interface (MPI)

- "Specification" of message passing API for distributed memory environment
  - http://www.mcs.anl.gov/mpi/www
  - https://www.mpi-forum.org/docs
- History
  - **1992** MPI Forum; **1994** MPI-1.0; **1997** MPI-2.0; **2012** MPI-3.0
  - **2015** MPI-3.1
  - **2021** MPI-4.0

Fugaku Users Guide

- Fujitsu MPI
  - MPI-3.1
  - MPI-4.0 subset

Based on Prof. Kengo Nakajima's Slides (Introduction to Programming by MPI for Parallel FEM)

Message Passing Interface (MPI)

- Implementation
  - MPICH
  - OpenMPI (Fujitsu Compiler)

- Cross-Compilers
  - C language
    - mpifccpx –Nclang -Kfast
  - C++ language
    - mpiFCCpx –Nclang -Kfast
  - FORTRAN language
    - mpifrtpx

- Native Compilers
  - C language
    - mpifcc
  - C++ language
    - mpiFCC
  - FORTRAN language
    - mpifrt

Spack
(MPICH and OpenMPI)

-- linux-rhel8-a64fx / fj@4.7.0 ----------
fujitsu-mpi@head
mpich-tofu@master

-- linux-rhel8-a64fx / gcc@8.4.1 ----------
fujitsu-mpi@head
mpich-tofu@master

Hello World (C Program)

```
$ cd /vol0601/share/ra020021/ComputerScience/20221205_Nonaka
$ cp -r examples ~/
```

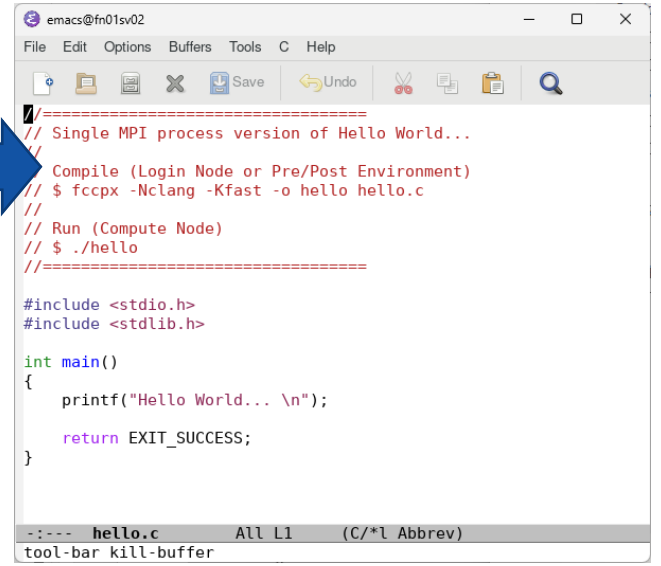
hello.c

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
   printf("Hello World... \n");

   return EXIT_SUCCESS;
}
```

```
$ cd
$ cd examples

$ cat mpi_hello.c

$ fccpx –Nclang –Kfast –o hello hello.c
$ ./hello
```

To Edit

```
$ emacs hello.c (GUI)
$ vi hello.c
```



MPI Processes (Distributed Memory)

mpi_hello.c

```
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>

int main()
{
   int my_rank;      // My MPI process number (Rank)
   int num_procs; // Total number MPI processes

   // Initialize MPI environment
   MPI_Init(NULL, NULL);
    // Get the total numberof MPI processes
   MPI_Comm_size( MPI_COMM_WORLD, &num_procs );
   // Get my rank number (Start from ZERO)
   MPI_Comm_rank( MPI_COMM_WORLD, &my_rank );
   printf("Hello World... from %d of %d \n", my_rank, num_procs);

   // Finalize MPI environment
   MPI_Finalize();
}
```

$ cat mpi_hello.c

$ mpifccpx -Nclang -Kfast -o mpi_hello mpi_hello.c
$ mpirun -n 48 ./mpi_hello

Batch Job (MPI)

job_mpi_hello.sh

```
#!/bin/bash
#PJM -g ra020021  # Group name

#PJM -L "rscgrp=small"   # Resource group small (up to 384 nodes)
# #PJM -L "rscgrp=excl_HPCS_2212-1" # EU-ASEAN HPC School: 12/5 0:00 to 12/9 24:00

#PJM -L "node=1"                        # Number of nodes
#PJM --mpi "max-proc-per-node=48"   # Max. MPI processes per node

#PJM -L "elapse=5:00"  # Running time
#PJM -L "eco_state=2"  # Eco mode setting

# #PJM -x PJM_LLIO_GFSCACHE=/vol0004:/vol0006  # Volume names used by the job
#PJM -s # Output statistical information


export PLE_MPI_STD_EMPTYFILE=off

cd ${HOME}/intro_pp
mpiexec -n 48 ./mpi_hello
```

// Job submission
$ pjsub job_mpi_hello.sh

// Check the job status
$ pjstat

// Check the job output
$ cat job_mpi_hello.sh.xxxxxxx.err
$ cat job_mpi_hello.sh.xxxxxxx.stats

$ cd output.xxxxxxxx
$ cd x/x
$ cat stdout.x.x

OpenMP (Shared Memory)

- "Specification" for a set of compiler directives, library routines, and environment variables that can be used to specify high-level parallelism in Fortran and C/C++ programs
  - https://www.openmp.org/specifications/

- History
  - 1998 C/C++ version 1.0; 2002 C/C++ version 2.0
  - 2011 OpenMP-3.1
  - 2013 OpenMP-4.0
  - 2015 OpenMP-4.5
  - 2020 OpenMP-5.0

Fugaku Users Guide

- Fujitsu Compiler
  - OpenMP 3.1
  - OpenMP 4.0
  - OpenMP 4.5
  - OpenMP 5.0 subset

OpenMP Threads

openmp_hello.c

```
#include <omp.h>

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char* argv[])
{
   int my_thread;   // My OpenMP thread
   int num_threads; // Total number of OpenMP threads

   // Beginning of OpenMP Parallel Region
   #pragma omp parallel
   {
      my_thread   = omp_get_thread_num();
      num_threads = omp_get_num_threads();
      printf("Hello World... from thread %d of %d \n", my_thread, num_threads);
   }
   // End of OpenMP parallel region

   return EXIT_SUCCESS;
}
```

$ cat openmp_hello.c

$ fccpx -Nclang -Kfast –Kopenmp -o openmp_hello openmp_hello.c
$ export OMP_NUM_THREADS=xx
$ ./openmp_hello

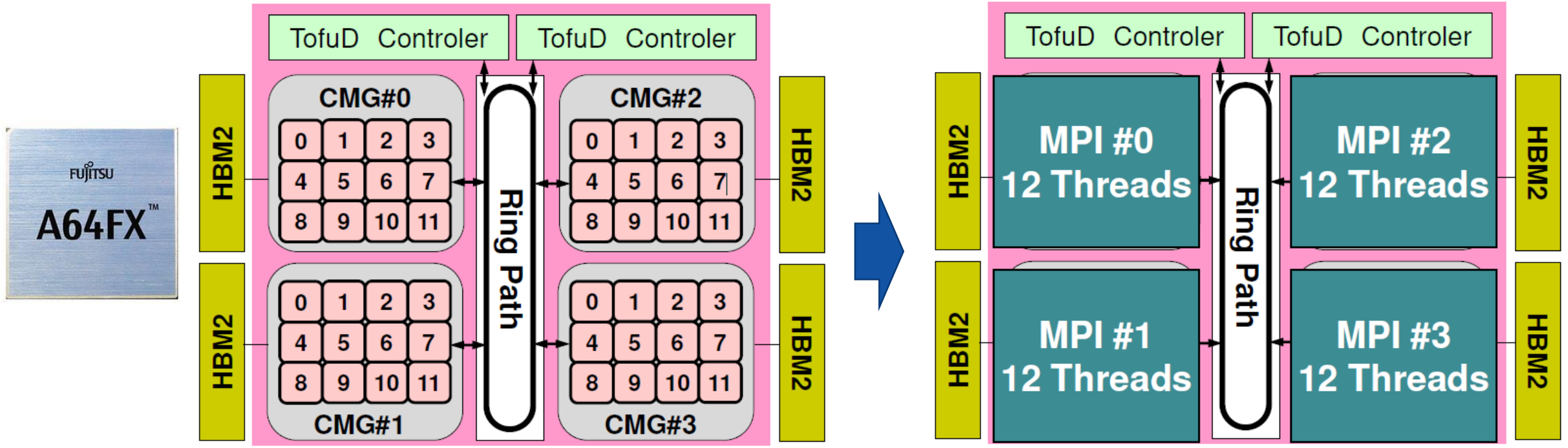
Batch Job (OpenMP)

job_openmp_hello.sh

```
#!/bin/bash
#PJM -g ra020021  # Group name

#PJM -L "rscgrp=small"            # Resource group small (up to 384 nodes)
# #PJM -L "rscgrp=excl_HPCS_2212-1" # EU-ASEAN HPC School: 12/5 0:00 to 12/9 24:00

#PJM -L "node=1"                  # Number of nodes

#PJM -L "elapse=5:00"  # Running time
#PJM -L "eco_state=2"  # Eco mode setting

# #PJM -x PJM_LLIO_GFSCACHE=/vol0004:/vol0006  # Volume names used by the job
#PJM -s # Output statistical information

export OMP_NUM_THREADS=48

cd ${HOME}/intro_pp
./openmp_hello
```

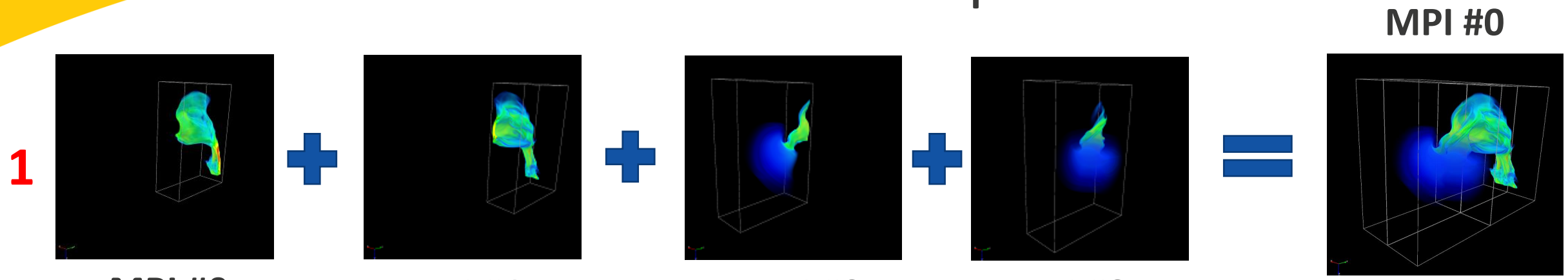
```
// Job submission
$ pjsub job_openmp_hello.sh

// Check the job status
$ pjstat

// Check the job output
$ cat job_openmp_hello.sh.xxxxxxx.err
$ cat job_openmp_hello.sh.xxxxxxx.stats

$ cd output.xxxxxxxx
$ cd x/x
$ cat stdout.x.x
```

MPI + OpenMP (Distributed and Shared Memory)



Based on Prof. Kengo Nakajima's Slides (Introduction to Programming by MPI for Parallel FEM)

Communication & Computation



MPI_Recv()
AlphaBlend()
MPI_Send()

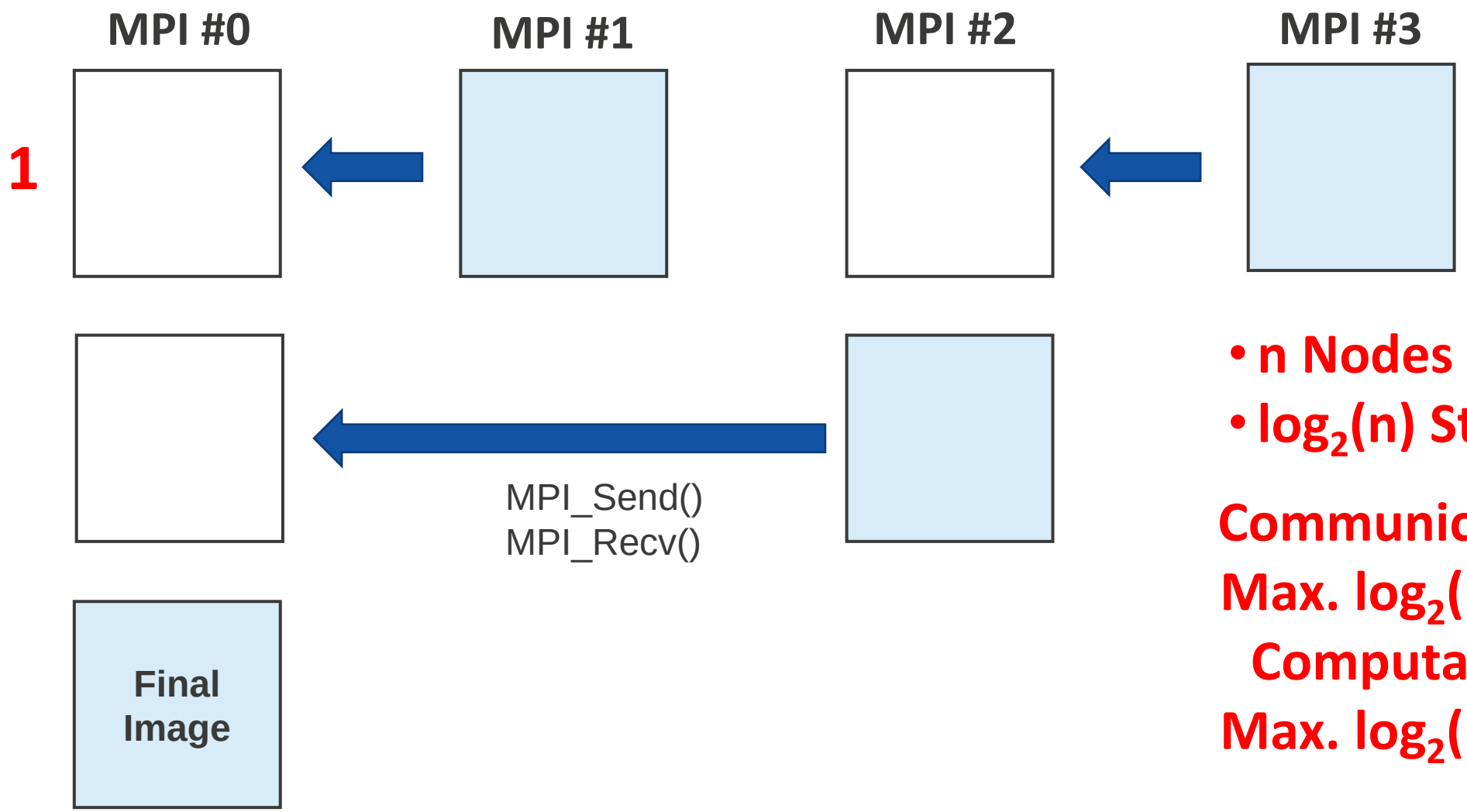
MPI_Recv()
AlphaBlend()
MPI_Send()

MPI_Recv()
AlphaBlend()
MPI_Send()

- **n Nodes**
- **n-1 Stages**

MPI #0
Communication
Computation
(n-1) x 1

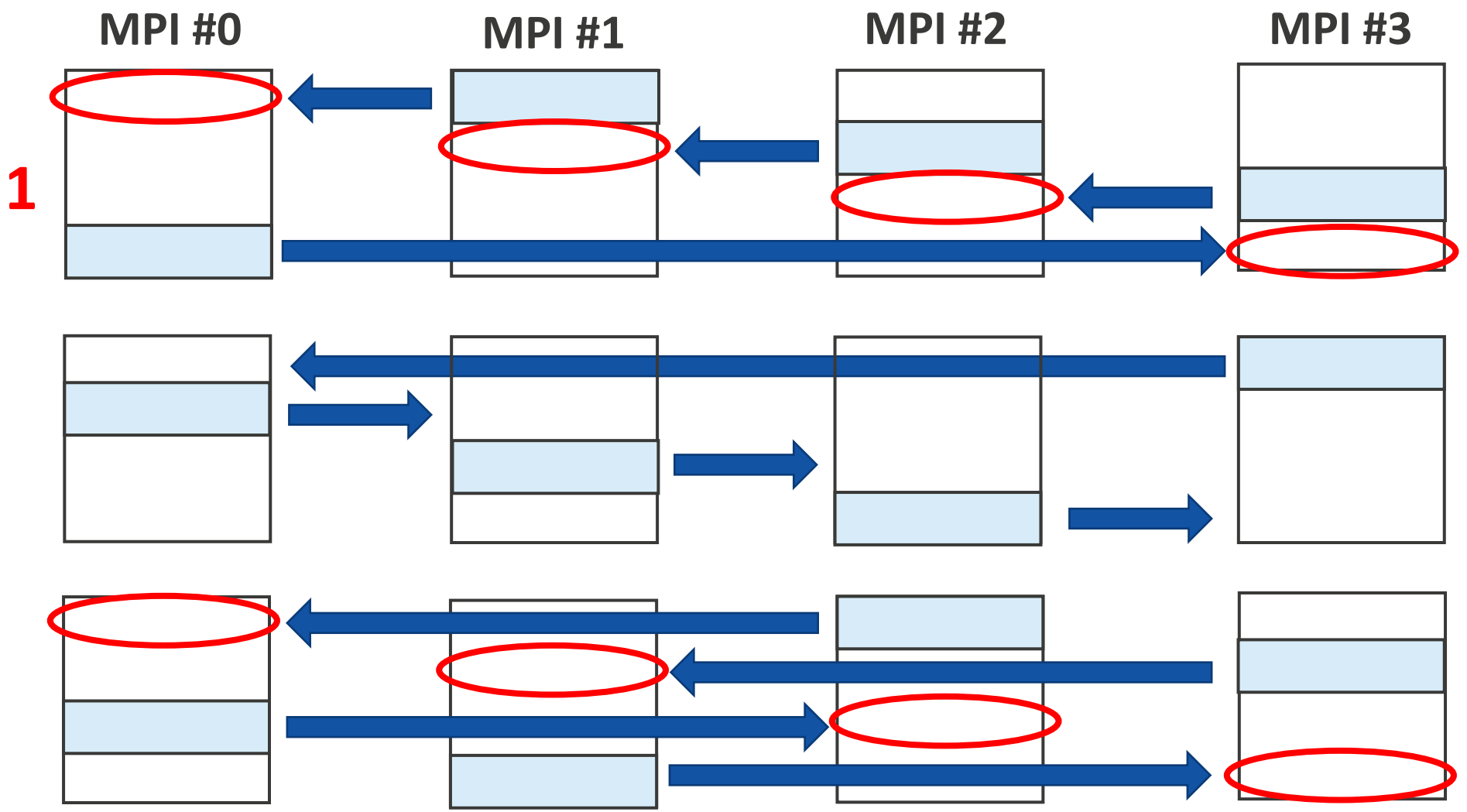
Binary Tree (2^n)



- n Nodes
- $\log_2(n)$ Stages

Communication
Max. $\log_2(n)$ x 1
Computation
Max. $\log_2(n)$ x 1

Direct Send

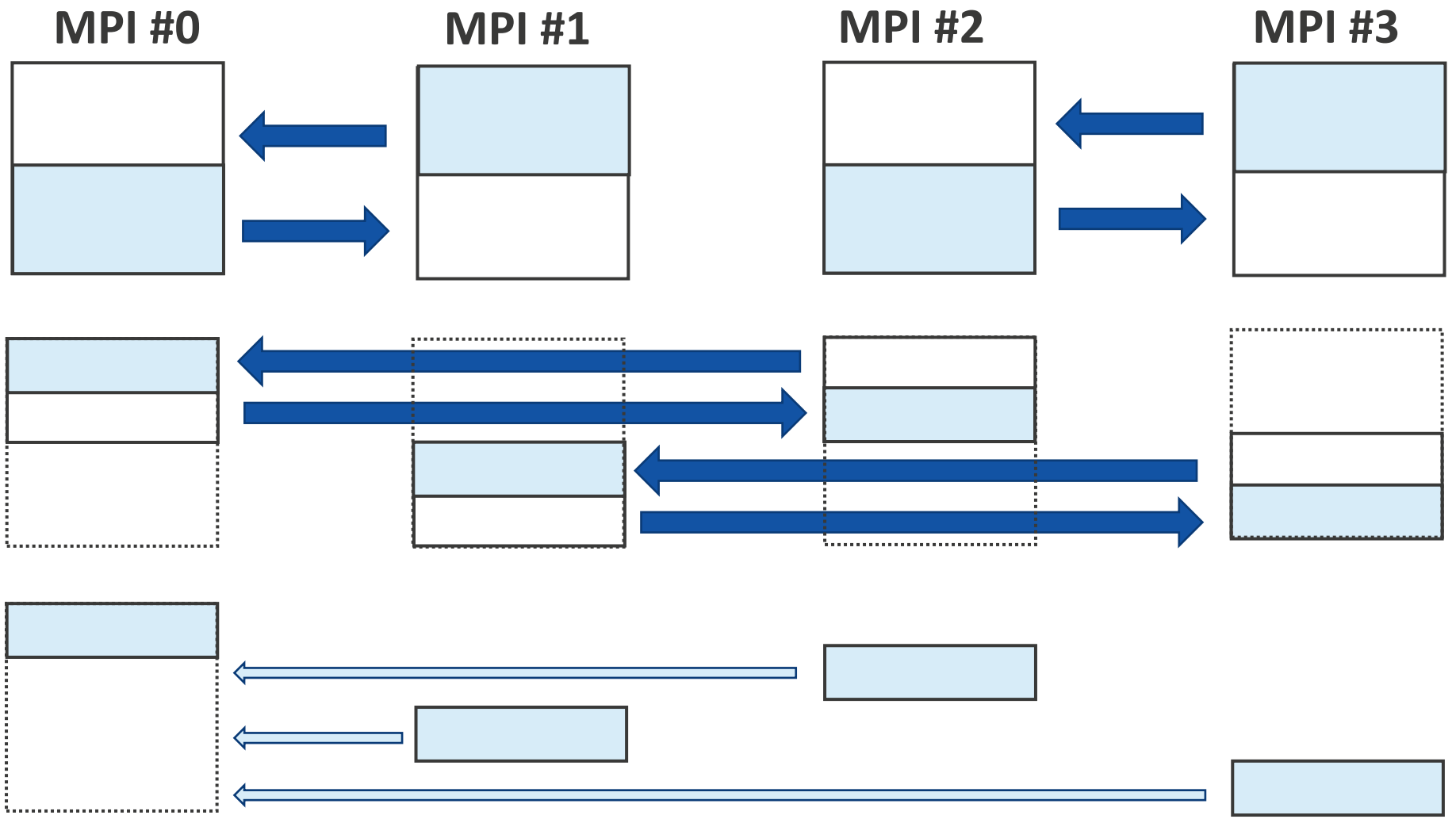


- n Nodes
- n-1 Stages

Communication
Computation
(n-1) x 1/n

Final Gather
Communication
1- (1/n)

Binary-Swap



- n Nodes
- $\log_2(n)$ Stages

Communication

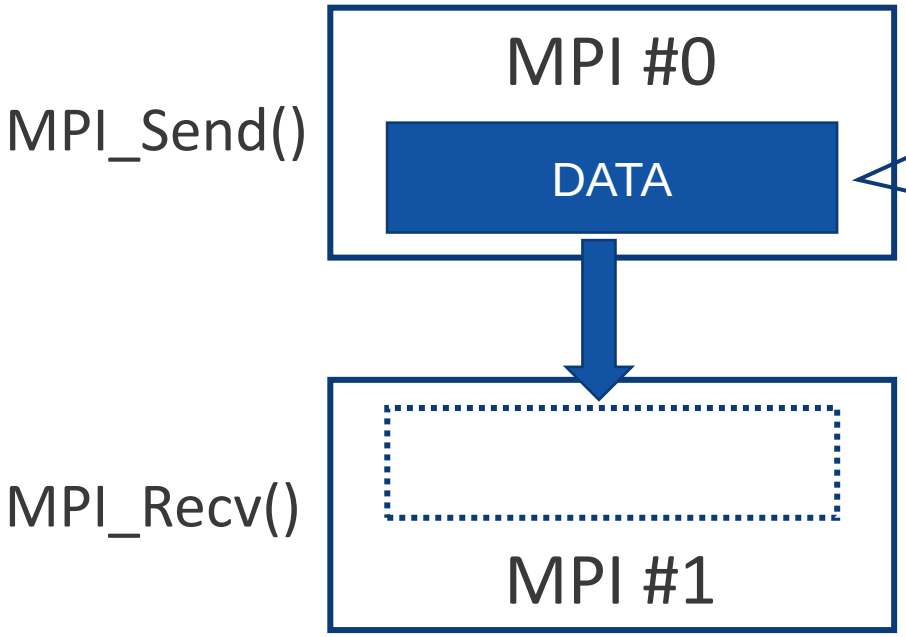
Computation

$$\sum_{i=1}^{\log_2 n} \left(\frac{1}{2^i}\right)$$

Final Gather Communication

1- (1/n)

Point-to-Point Communication



Data Type

MPI_CHAR	MPI_UNSIGNED_CHAR	MPI_FLOAT
MPI_SHORT	MPI_ UNSIGNED_ SHORT	MPI_DOUBLE
MPI_INT	MPI_ UNSIGNED_ INT	MPI_LONG_DOUBLE
MPI_LONG	MPI_ UNSIGNED_ LONG	MPI_BYTE
…	…	…

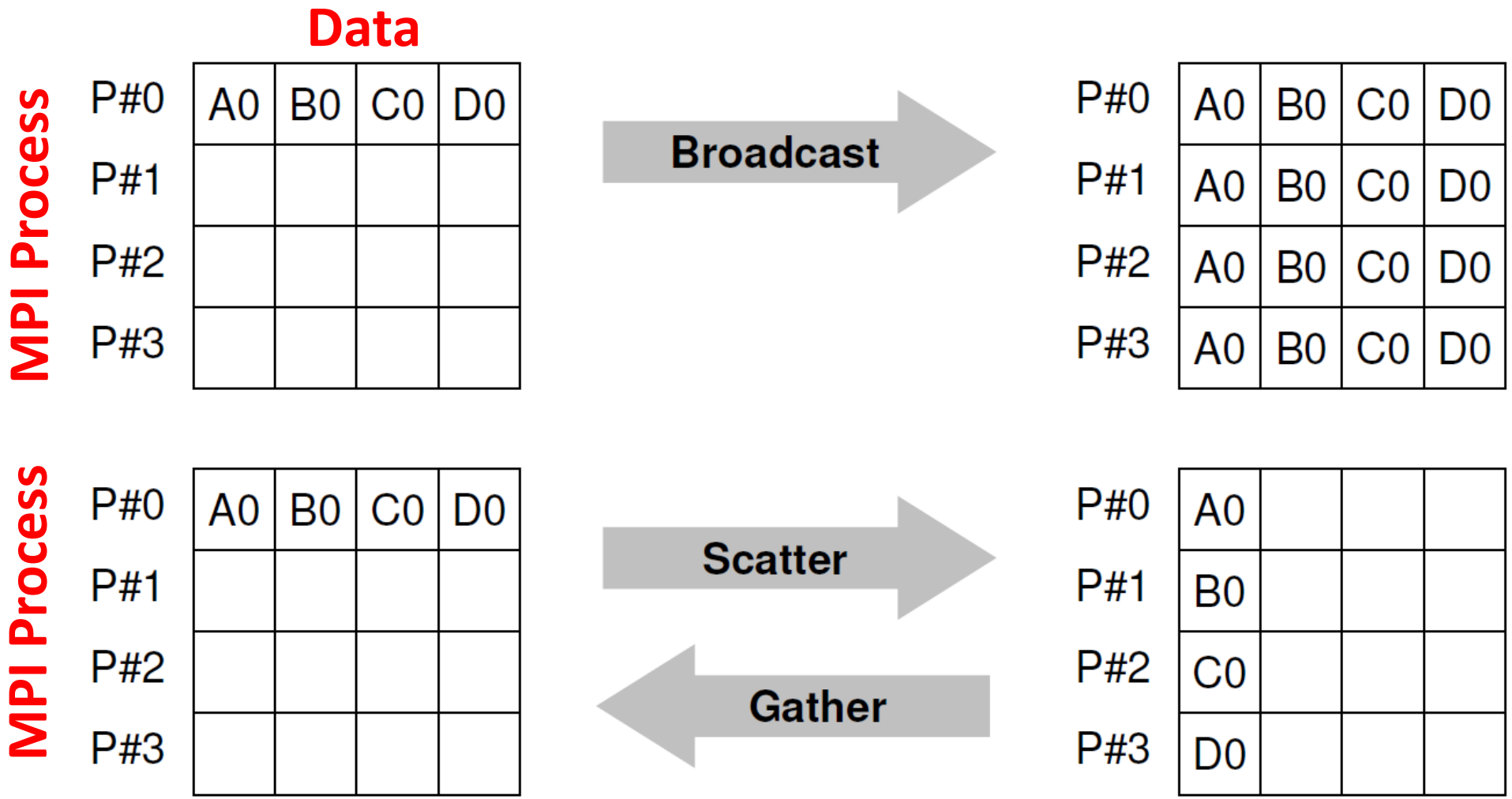
Example C code: mpi_send_recv_hello.c

Example Code (Point-to-Point Communication)

MPI_Send / MPI_Recv

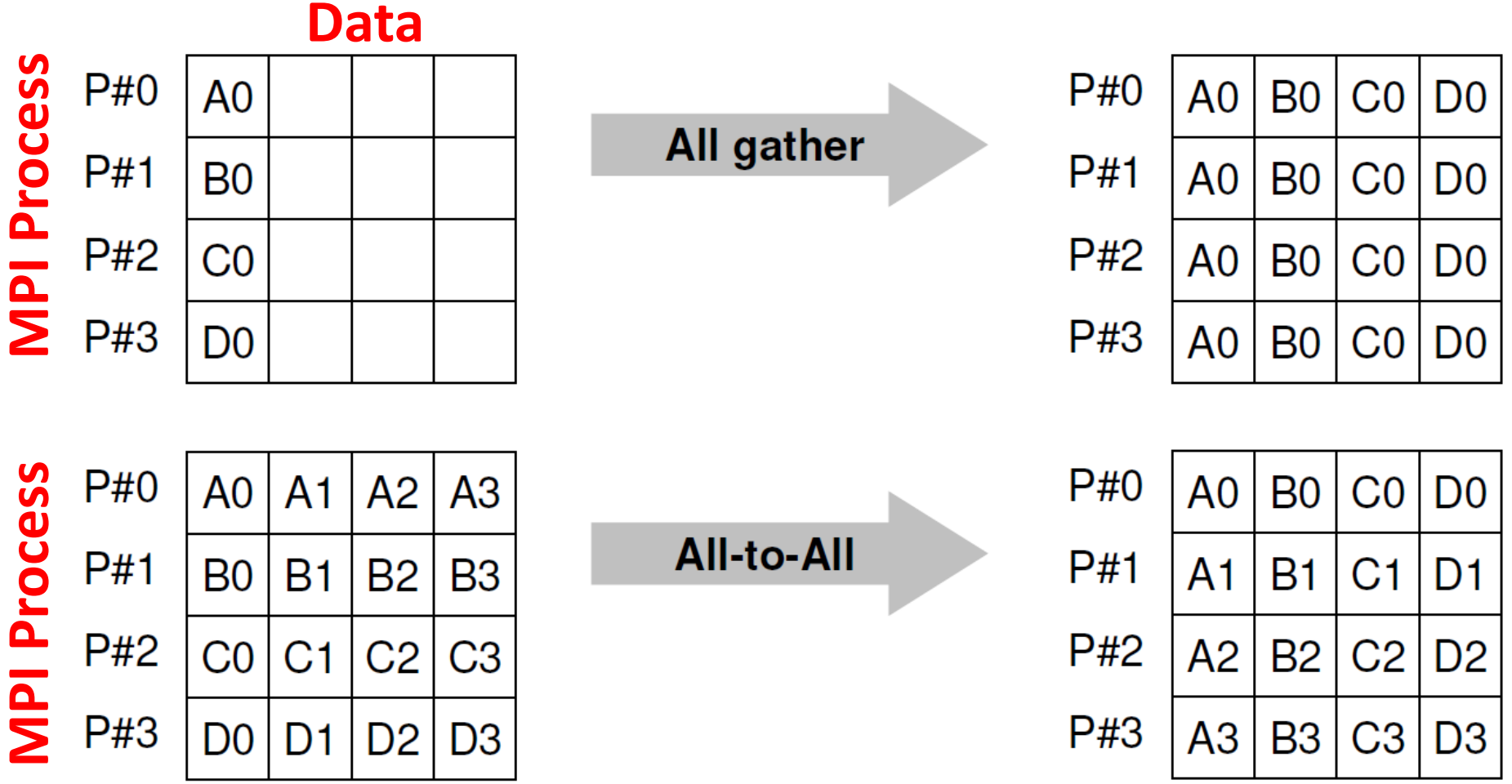
```
$ cat mpi_sed_recv_hello.c

$ mpifccpx -Nclang -Kfast -o mpi_sed_recv_hello mpi_sed_recv_hello.c

$ pjsub mpi_sed_recv_hello.c
$ pjstat

$ cat output.xxxxxxxx/x/x/stdout.x.x
```

Collective Communication



Based on Prof. Kengo Nakajima's Slides (Introduction to Programming by MPI for Parallel FEM)

Collective Communication



Based on Prof. Kengo Nakajima's Slides (Introduction to Programming by MPI for Parallel FEM)

Collective Communication

op: Operation

MPI Process

P#0	A0	B0	C0	D0
P#1	A1	B1	C1	D1
P#2	A2	B2	C2	D2
P#3	A3	B3	C3	D3

Reduce

P#0	op.A0-A3	op.B0-B3	op.C0-C3	op.D0-D3
P#1				
P#2				
P#3				

MPI Process

P#0	A0	B0	C0	D0
P#1	A1	B1	C1	D1
P#2	A2	B2	C2	D2
P#3	A3	B3	C3	D3

All reduce

P#0	op.A0-A3	op.B0-B3	op.C0-C3	op.D0-D3
P#1	op.A0-A3	op.B0-B3	op.C0-C3	op.D0-D3
P#2	op.A0-A3	op.B0-B3	op.C0-C3	op.D0-D3
P#3	op.A0-A3	op.B0-B3	op.C0-C3	op.D0-D3

Based on Prof. Kengo Nakajima's Slides (Introduction to Programming by MPI for Parallel FEM)

Collective Communication

op: Operation

MPI Process

P#0	A0	B0	C0	D0
P#1	A1	B1	C1	D1
P#2	A2	B2	C2	D2
P#3	A3	B3	C3	D3

Reduce scatter

P#0	op.A0-A3			
P#1	op.B0-B3			
P#2	op.C0-C3			
P#3	op.D0-D3			

Some predefined reduce operations

MPI_MAX	MPI_LAND	MPI_LXOR
MPI_MIN	MPI_BAND	MPI_BXOR
MPI_SUM	MPI_LOR	MPI_MAXLOC
MPI_PROD	MPI_BOR	MPI_MINLOC
…	…	…

Based on Prof. Kengo Nakajima's Slides (Introduction to Programming by MPI for Parallel FEM)

Example Codes (Collective Communication)

MPI_Bcast

```
$ cat mpi_bcast_random.c
$ mpifccpx -Nclang -Kfast -o mpi_bcast_random mpi_bcast_random.c
$ pjsub job_mpi_bcast_random.sh
```

MPI_Allreduce

```
$ cat mpi_allreduce_random.c
$ mpifccpx -Nclang -Kfast -o mpi_allreduce_random mpi_allreduce_random.c
$ pjsub job_mpi_allreduce_random.sh
```

Send the output of the MPI_AllReuce job

/vol0601/share/ra020021
ComputerScience/20221205_Nonaka
job_output

$ cp –r output.xxxxx /vol0601/share/ra020021/ComputerScience/20221205_Nonaka/job_output