

# Collaboration with MPI

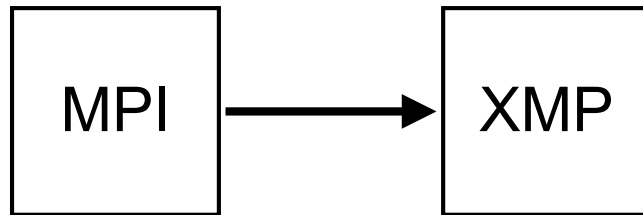
---

Masahiro Nakao (RIKEN AICS, Japan)

# Agenda

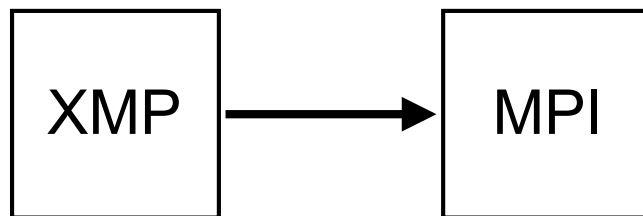
---

- How to call an XMP program from an MPI program



This feature is useful that an existing MPI code calls a new function in XMP

- How to call an MPI program from an XMP program



XMP program can call HPC libraries implemented in MPI (FFTE and so on).

Sorry, these functions are implemented in only XMP/C.  
Now implementing XMP/Fortran too.

# Agenda

---

Please get sample files from <http://xcalablemp.org/lecture.html>  
"source code" in "Collaboration with MPI"

# How to call XMP from MPI

---

./collaboration/MPItoXMP/mpi2xmp.c

```
#include "xmp.h"

int main(int argc, char **argv)
{
    MPI_Init(&argc, &argv);
    xmp_init(MPI_COMM_WORLD);
    int sum = foo(10);
    xmp_finalize();
    MPI_Finalize();

    printf("%d\n", sum);

    return 0;
}
```

`xmp_init(MPI_Comm comm)`

Initialization for XMP.

`xmp_finalize()`

Finalization for XMP.

`xmp_init()` should be used after `MPI_Init()`.

`xmp_finalize()` should be used before `MPI_Finalize()`.

`XMP function call` should be used between `xmp_init()` and `xmp_finalize()`.

# How to call XMP from MPI

---

./collaboration/MPItoXMP/xmp.c

```
#pragma xmp nodes p[*]
#pragma xmp template t[20]
#pragma xmp distribute t[block] onto p
int a[20];
#pragma xmp align a[i] with t[i]

int foo(int ulimit)
{
    int sum = 0;
#pragma xmp loop on t[i] reduction(+:sum)
    for(int i=0;i<ulimit;i++){
        a[i] = i;
        sum += a[i];
    }
    return sum;
}
```

The **node set** is created from the mpi communicator indicated in xmp\_init().

This function returns total value.

# How to call XMP from MPI

---

```
$ mpiicc mpi2xmp.c -c -I[your install path]/include
```

```
$ xmpcc -c xmp.c
```

```
$ xmpcc mpi2xmp.o xmp.o -o mpi2xmp.x
```

```
$ mpirun -np 4 ./mpi2xmp.x
```

```
45
```

```
45
```

```
45
```

```
45
```

[your install path] is a  
install directory of omni compiler

# How to call MPI from XMP

collaboration/XMPtoMPI/xmp2mpi.c

```
#pragma xmp nodes p[4]

int main(int argc, char **argv)
{
    xmp_init_mpi(&argc, &argv);

    MPI_Comm comm = xmp_get_mpi_comm();
    foo(comm);

    xmp_finalize_mpi();
    return 0;
}
```

`xmp_init_mpi(int* argc, char ***argv)`

Initialization for MPI

`xmp_finalize_mpi()`

Finalization for MPI

`xmp_get_mpi_comm()`

Get MPI communicator associated with the executing node set.

collaboration/XMPtoMPI/mpi.c

```
void foo(MPI_Comm comm){
    int rank, size;
    MPI_Comm_rank(comm, &rank);
    MPI_Comm_size(comm, &size);
    printf("[%d] %d\n", rank, size);
}
```

# How to call MPI from XMP

---

```
$ xmpcc xmp2mpi.c -c
```

```
$ mpiicc mpi.c -c
```

```
$ xmpcc xmp2mpi.o mpi.o -o xmp2mpi.x
```

```
$ mpirun -np 4 ./xmp2mpi.x
```