

XcalableMP

Directive-Based Language eXtension for Scalable Parallel Programming



T2K Open Supercomputer Alliance
University of Tsukuba University of Tokyo Kyoto University

Overview

- **XcalableMP** is a directive-based PGAS language for distributed memory system
- Designed by XcalableMP Specification Working Group

Members from academia(U. Tsukuba, U. Tokyo, Kyoto U., Kyusyu U.), research labs(RIKEN, NIFS, JAXA, JAMSTEC/ES), industries(Fujitsu, NEC, Hitachi) in Japan

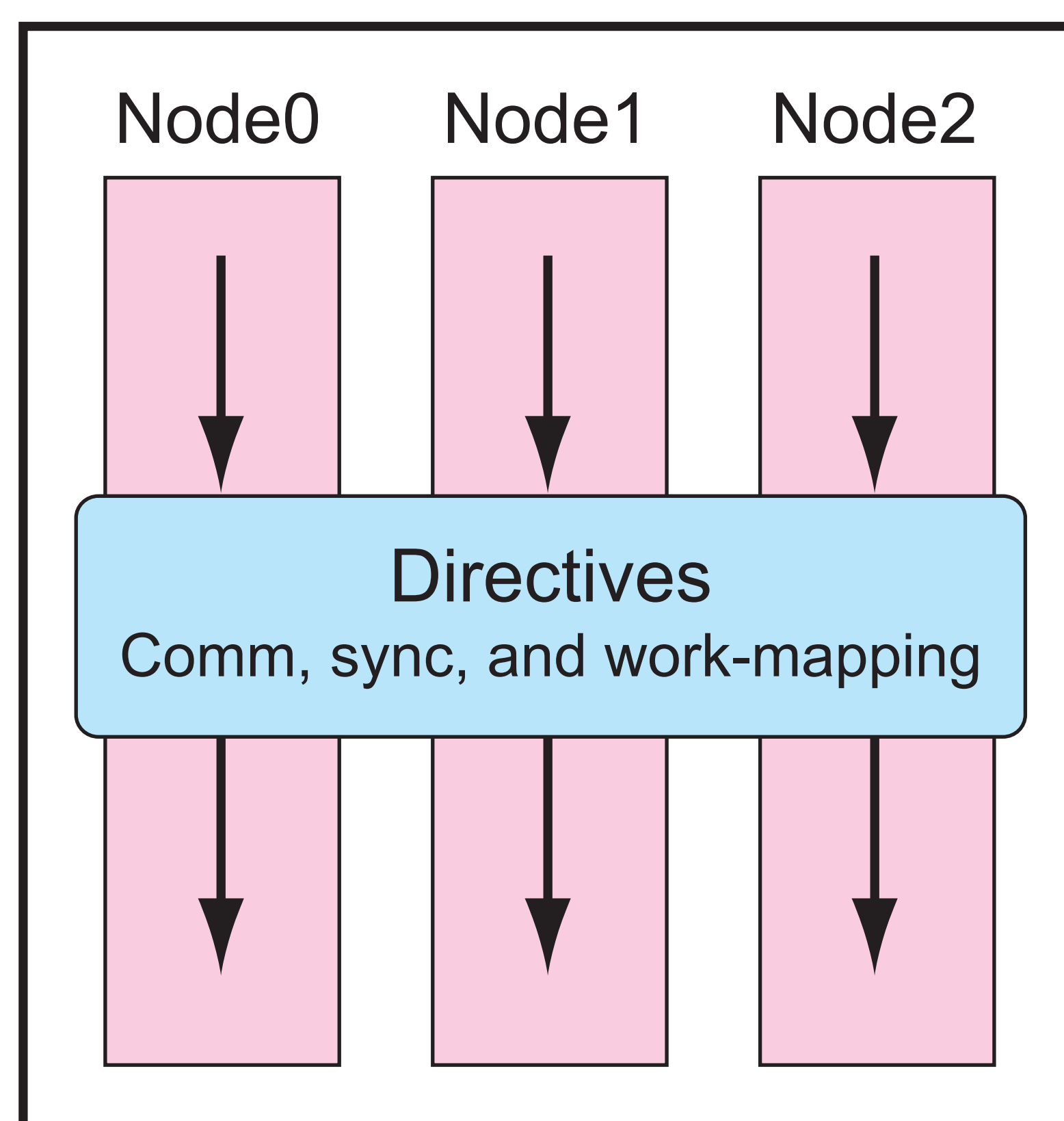
Language Features

To reduce code-writing and educational costs

- Language extension of C99 and Fortran 95
- Supports typical parallelization based on the data parallel paradigm and work mapping under “global-view” programming model
- Also includes Co-Array Fortran like feature as “local-view” programming model

Performance-awareness

- SPMD as a basic execution model
- A thread starts execution in each node independently (as in MPI)
- Communication, synchronization and work-mapping occur when directives are encountered
- All actions are taken by directives for being “easy-to-understand” in performance tuning (different from High Performance Fortran)



Current Solution for parallel programming

```
int array[MAX];

main(int argc, char **argv){
  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &
  MPI_Comm_size(MPI_COMM_WORLD, &
  dx = MAX/size;
  llimit = rank * dx;
  if(rank != (size-1)) ulimit = llimit + dx;
  else ulimit = MAX;

  temp_res = 0;
  for(i=llimit; i < ulimit; i++){
    array[i] = func(i);
    temp_res += array[i];
  }

  MPI_Allreduce(&temp_res, &res, 1, MPI_INT, MPI_SUM, ...);
  MPI_Finalize();
}
```

Only way to program is MPI, but MPI programming seems difficult... we have to rewrite almost entire program and it is time-consuming and hard to debug... mmm



We need better solutions !!

```
int array[MAX];
#pragma xmp nodes p(*)
#pragma xmp template t(0:MAX)
#pragma xmp distribute t(block) onto p
#pragma xmp align array[i] with t(i)

main(){
#pragma xmp loop on t(i) reduction (+res)
for(i = 0; i < MAX; i++){
  array[i] = func(i);
  res += array[i];
}
}
```

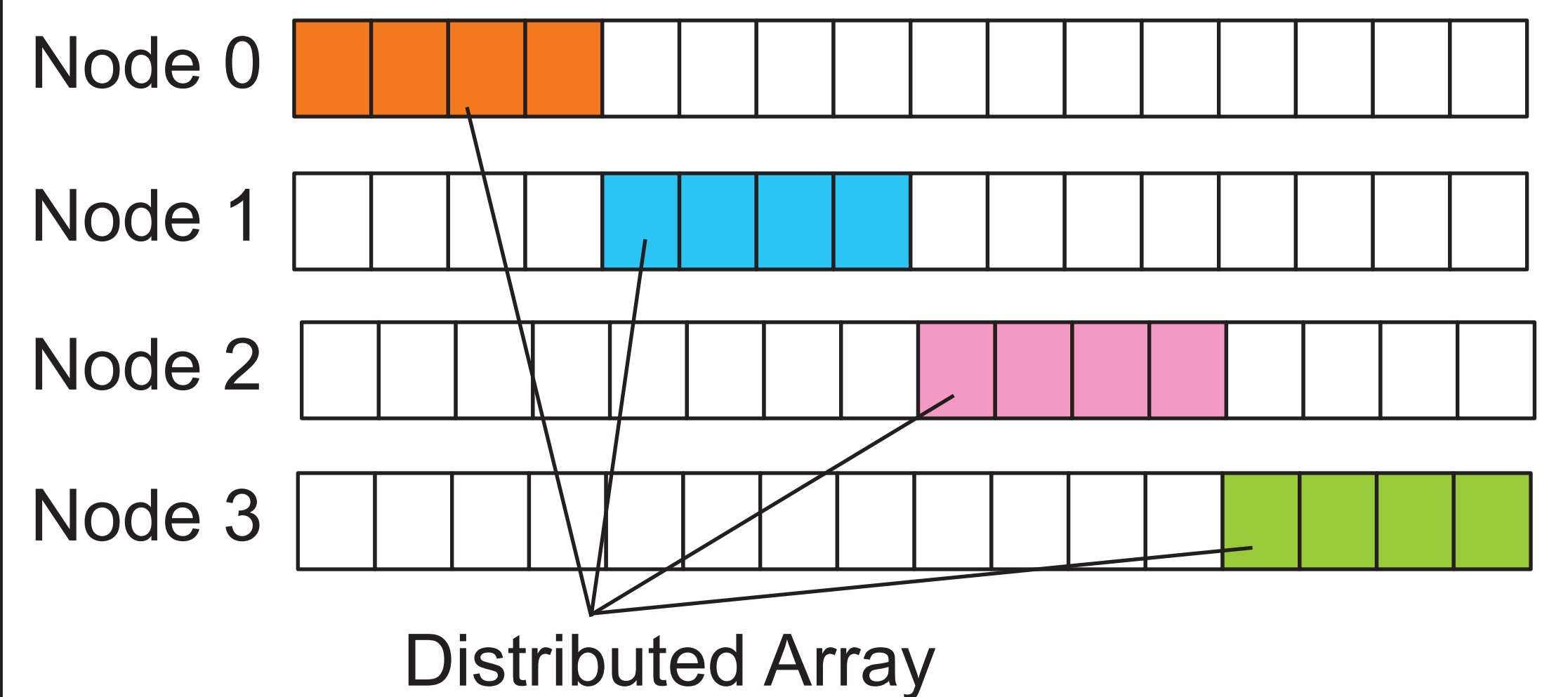
XcalableMP enables users to easily develop parallel programs and to tune performance with minimal and simple notation !!



```
int a[16];
```

Data Mapping

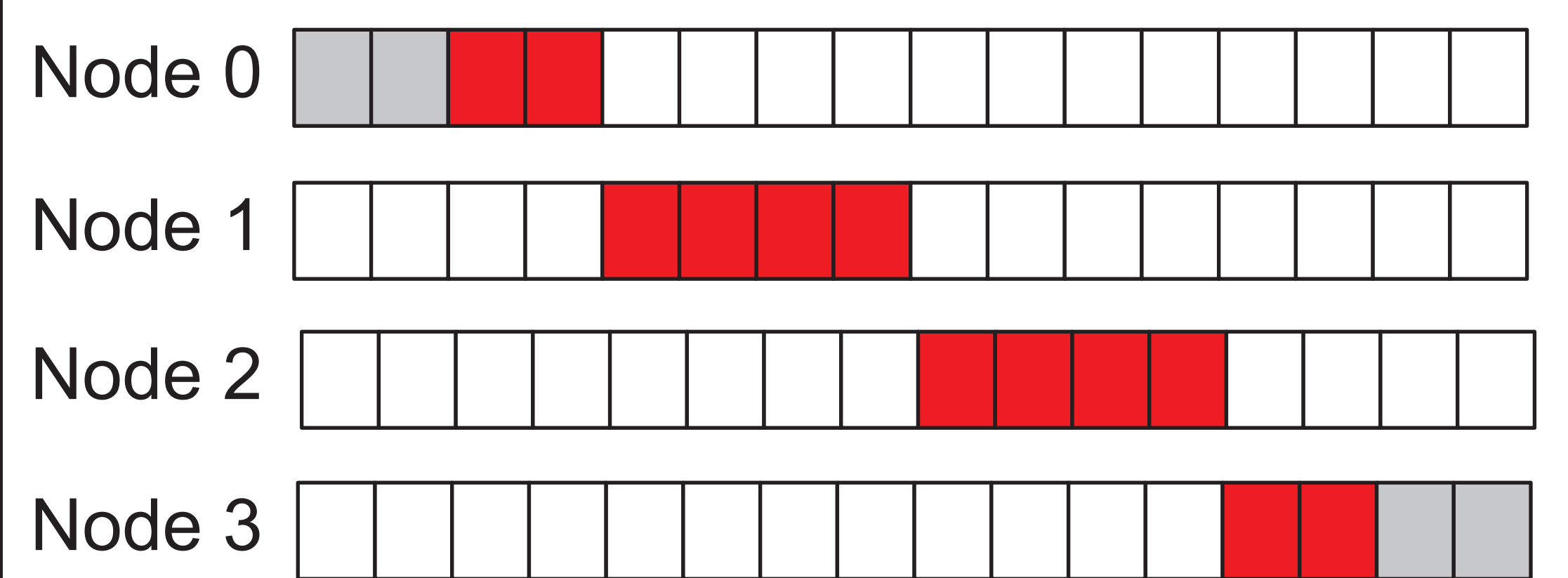
```
#pragma xmp nodes p(4)
#pragma xmp template t(0:16)
#pragma xmp distribute t(block) onto p
#pragma xmp align a[i] with t(i)
```



```
#pragma xmp loop on t(i)
```

Work Mapping

```
for(i = 2; i < 14; i++) {
  a[i] = func(i);
}
```



Each node computes red elements in parallel