

Interfaces of Parallel Programming Language XcalableMP to Parallel Numerical Libraries

Takenori Shimosaka

Programming environment research team

RIKEN AICS

AICS-Café March.23 2012

Contents

- ❑ XcalableMP and Parallel numerical libraries
- ❑ A necessity of XcalableMP interface to parallel numerical libraries
- ❑ A design of the XcalableMP interface

❑ A Location of XcalableMP in Programming environment

Parallel programming language XcalableMP is directive-based language extensions for C and Fortran90 (A XcalableMP processor of our team is a translator from XcalableMP programming model to MPI programming model.)

<e.g. Programming environment>

Programming language : C/C++, Fortran, Java, COBOL, Perl, Lisp ,XcalableMP, ...

Compiler : gcc, gfortran, ...

Debugger : gdb, ...

Editor : Notepad, vi, emacs

Performance analysis tool

Library: BLAS, LAPACK, ScaLAPACK, ...

....

❑ XcalableMP

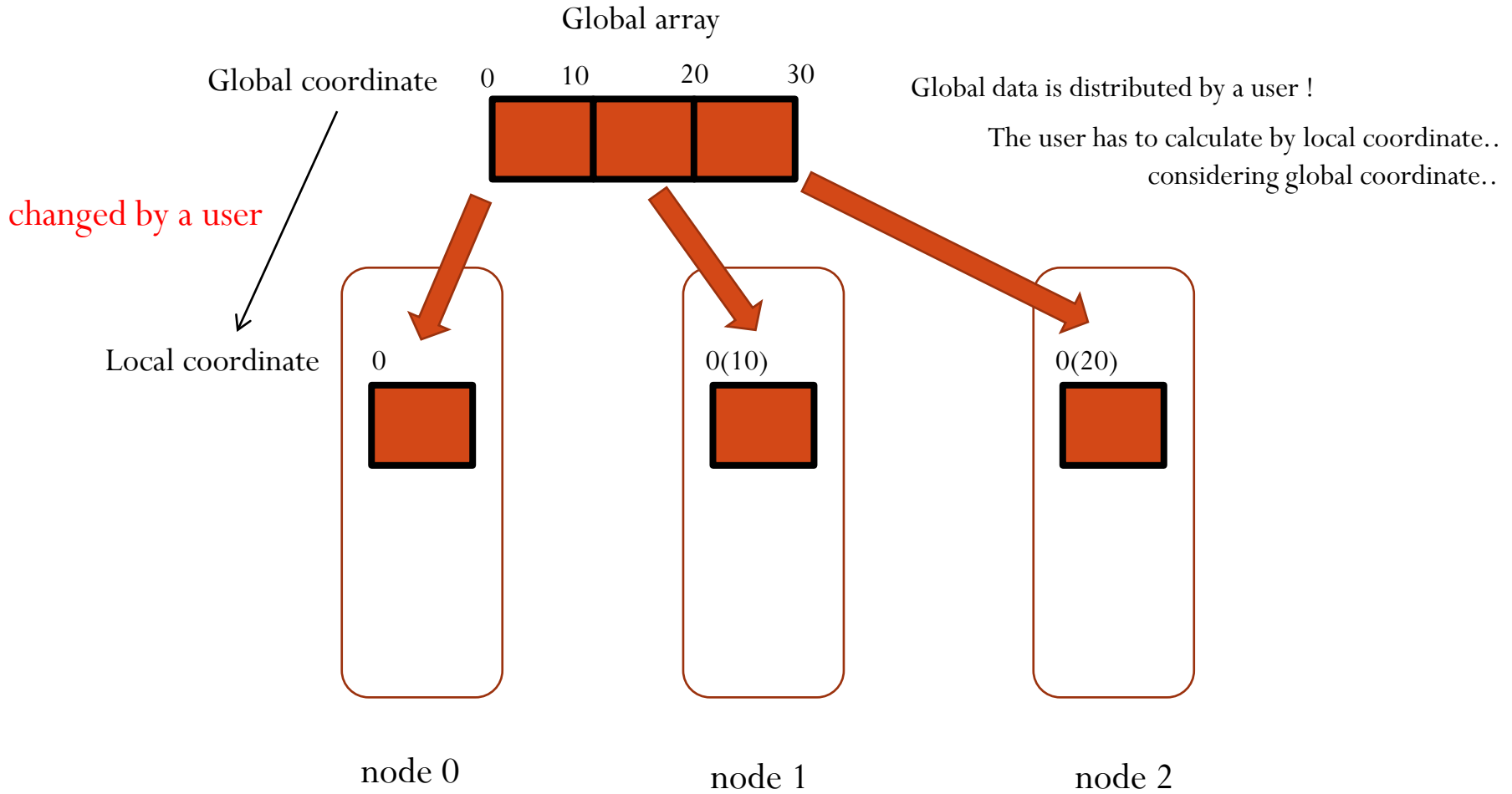
— PGAS(Parallel programming language for distributed memory systems) —

Features :

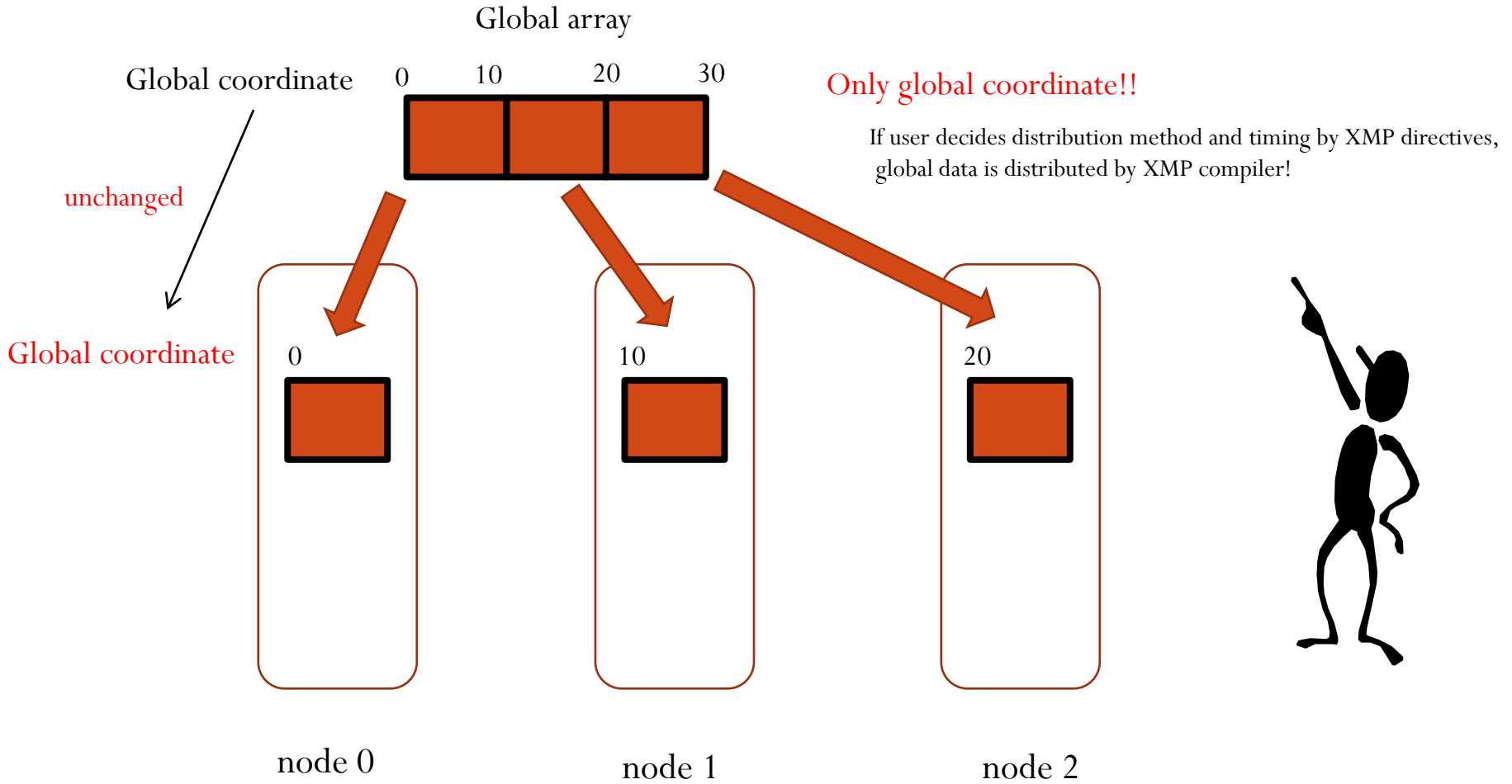
- Directive-based language extensions for C and Fortran90
- Data distribution and communication are specified by directives.
 - A programmer has a **Global view** on distributed data, then he doesn't need to handle directly distributed data using local coordinates.
 - Good readability.
Design changes and program modifications are easy.

- It is possible to mix OpenMP and MPI programming into XMP.
 - We can easily use the existing programs.

❏ MPI programming model



XcalableMP programming model



❑ High performance parallel numerical libraries

ScaLAPACK : Scalable Linear Algebra PACKage(<http://www.netlib.org/scalapack/>)

FFTW : Fast Fourier Transform Library(<http://www.fftw.org/>)

ParMETIS : Parallel Graph Partitioning and Fill-reducing Matrix Ordering

(<http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>)

MUMPS : a MULTifrontal Massively Parallel sparse direct Solver(<http://graal.ens-lyon.fr/MUMPS/>)

Lis : a Library of Iterative Solvers for linear systems(<http://www.ssisc.org/lis/>)

.....

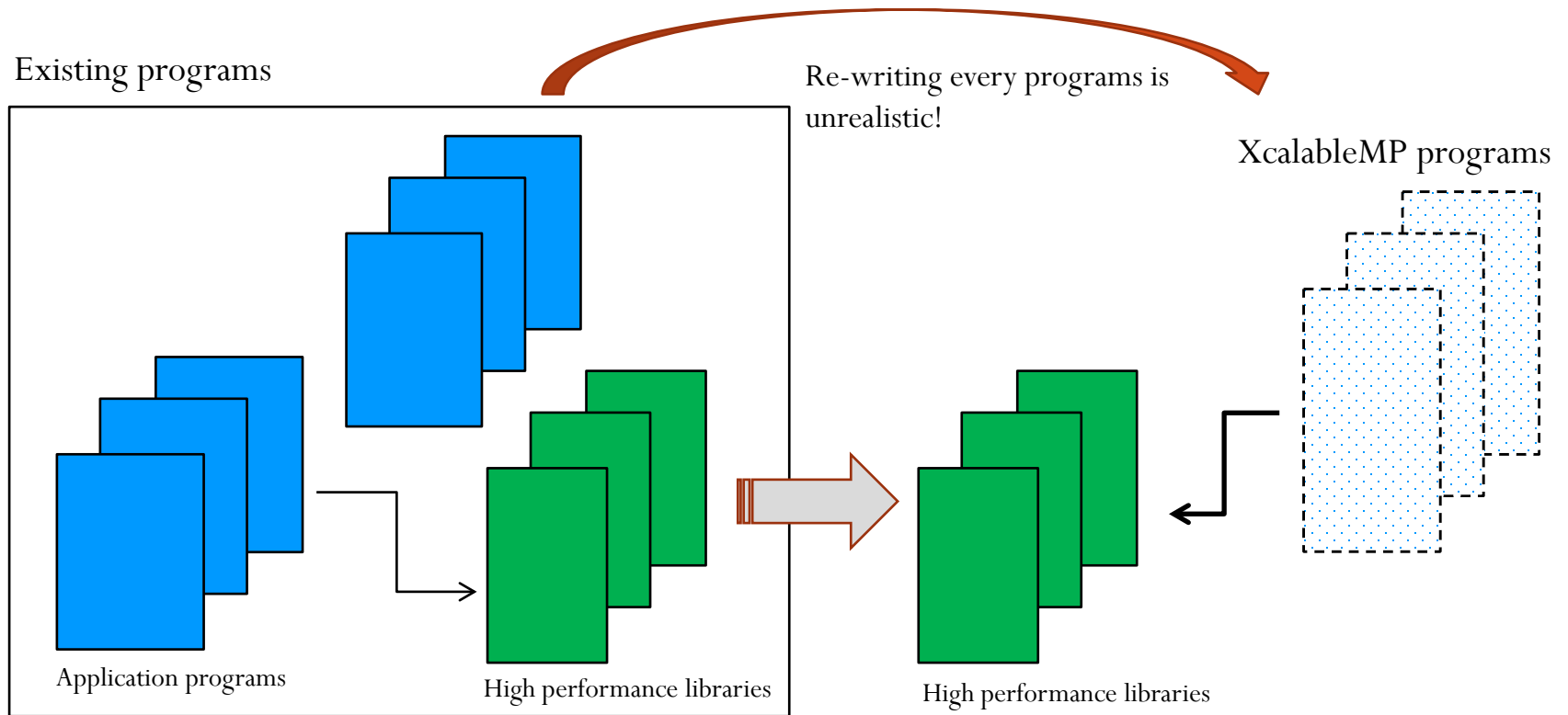
.....

Most of high performance parallel numerical libraries are implemented
by **MPI programming model**.

❑ A necessity of XcalableMP interface to parallel numerical libraries

Issue : Re-writing every application programs using only XMP is unrealistic.

⇒ In order to develop application programs easily, it is important to make used of existing high performance libraries.



Research objective :

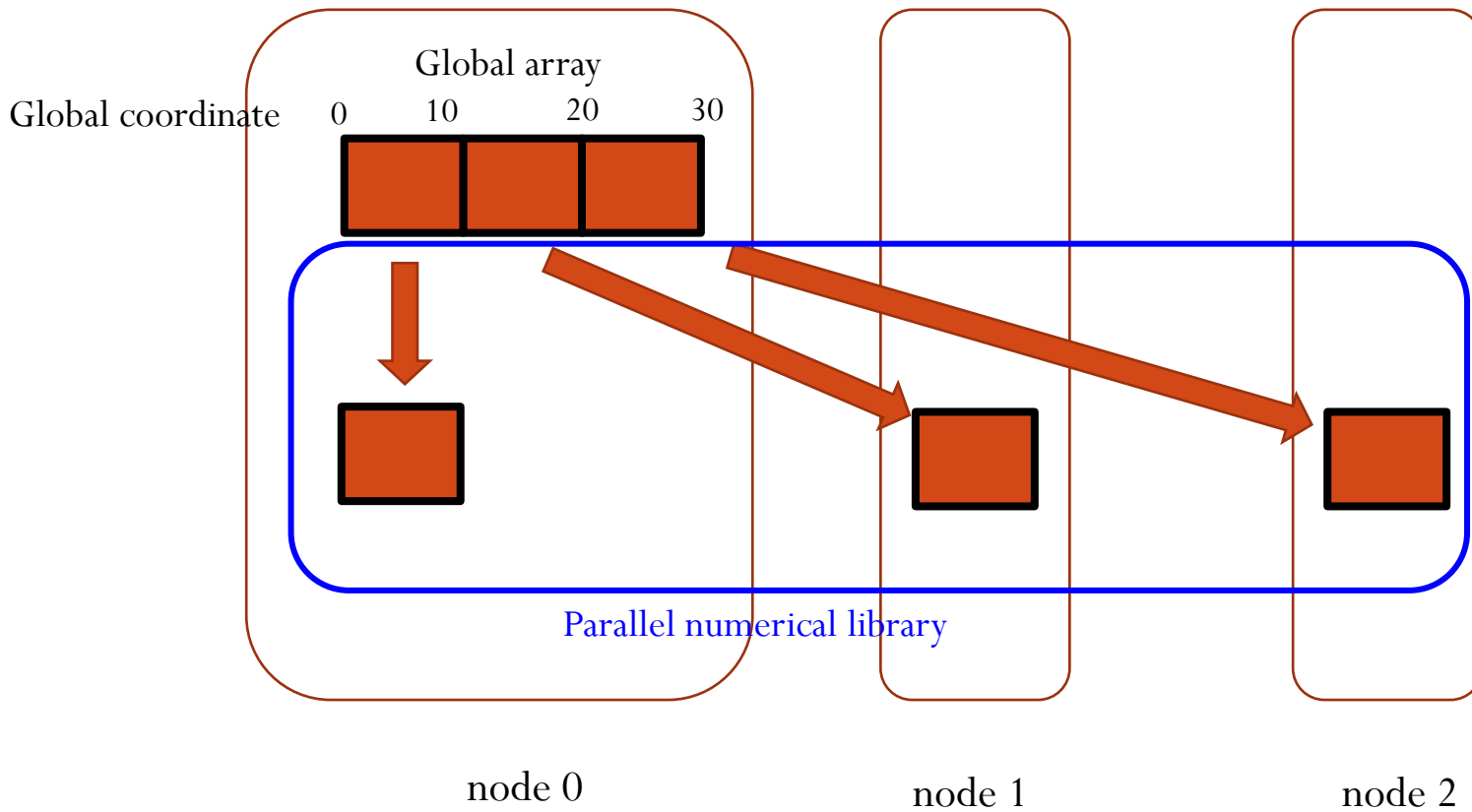
To design a way to make use of MPI parallel libraries in XMP program.

Contents

- ❑ XcalableMP and Parallel numerical libraries
- ❑ A necessity of XcalableMP interface to parallel numerical libraries
- ❑ A design of the XcalableMP interface

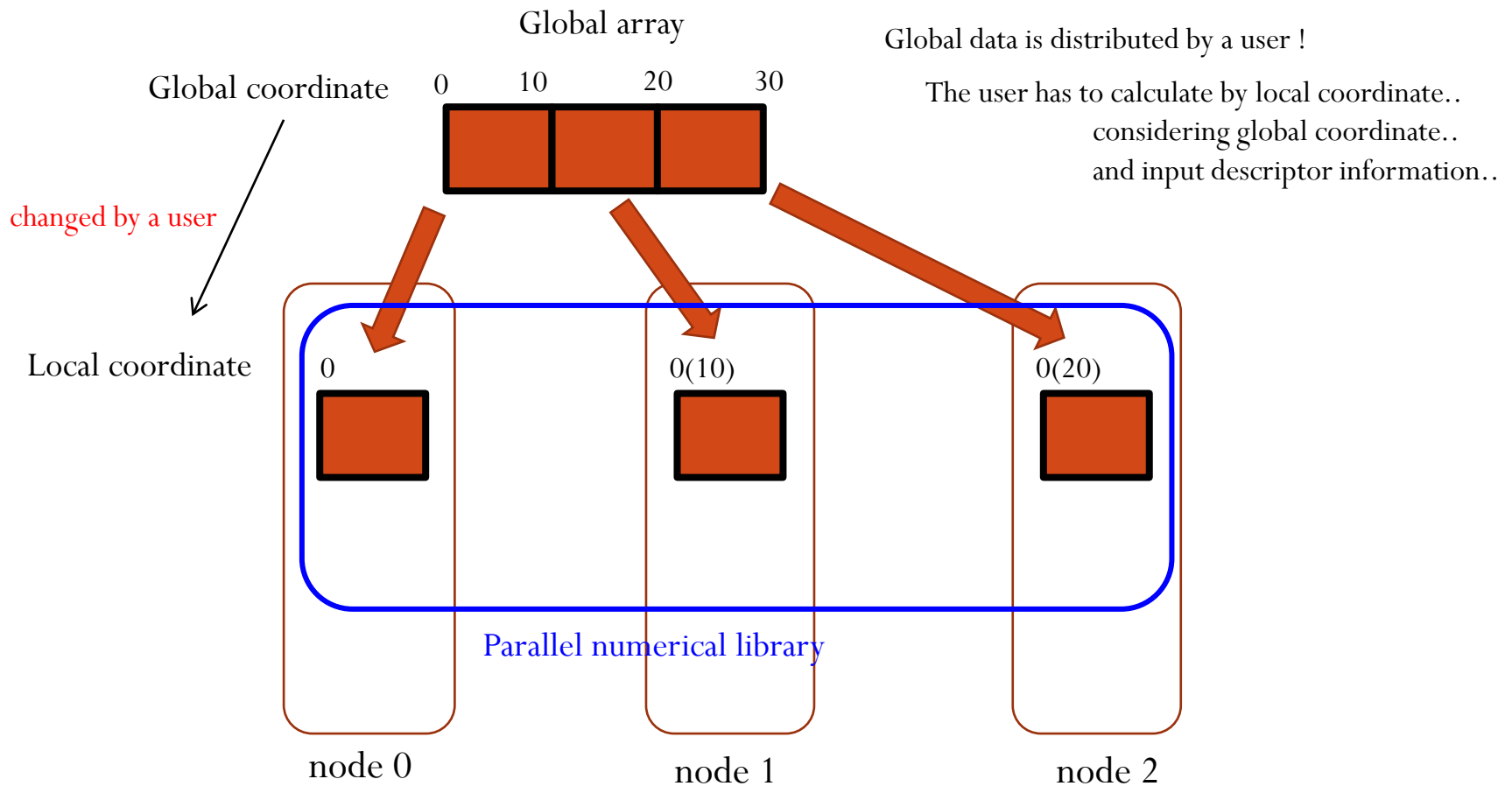
Usage of parallel numerical libraries

<Distribution pattern 1>

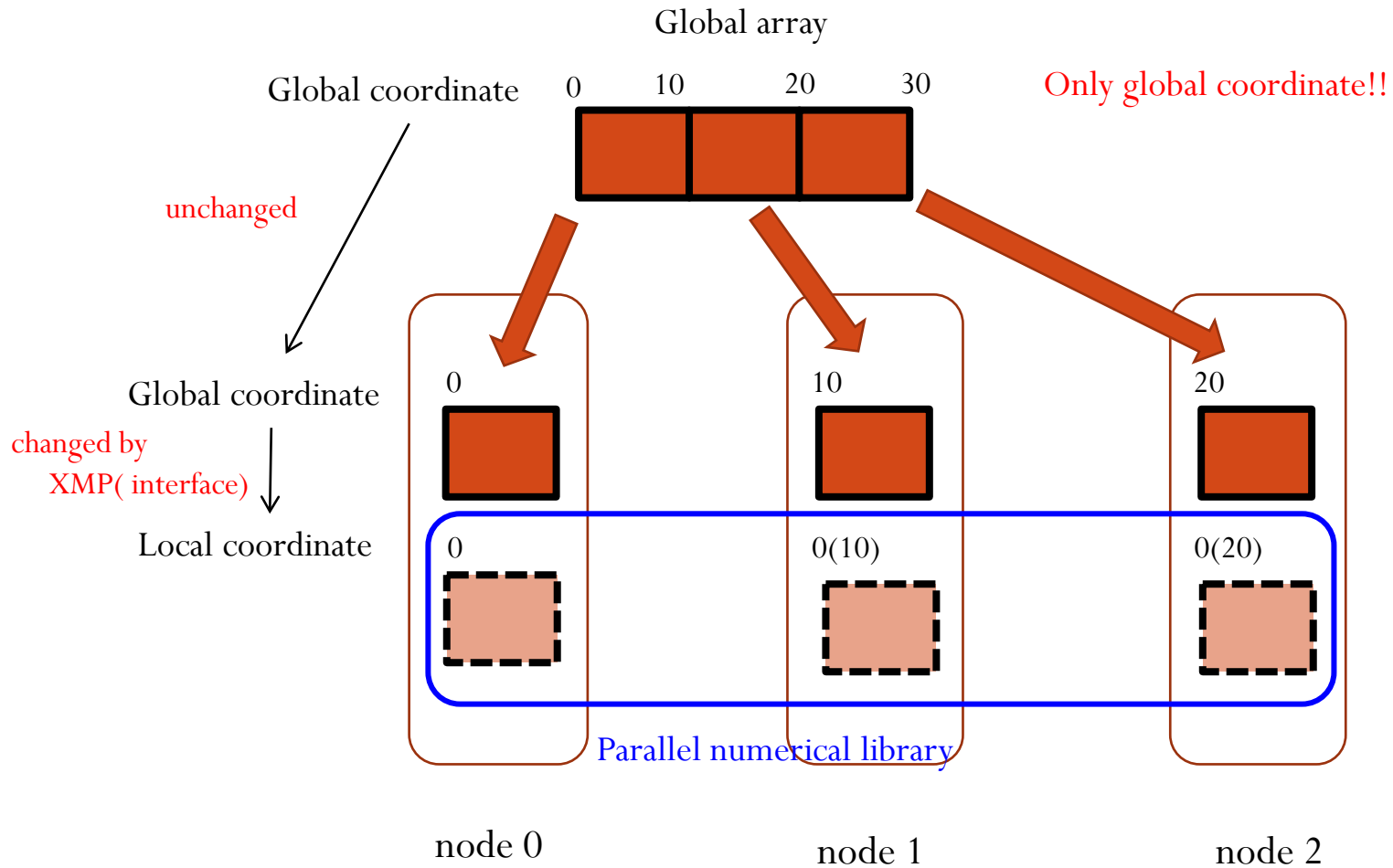


Usage of parallel numerical libraries

<Distribution pattern 2>

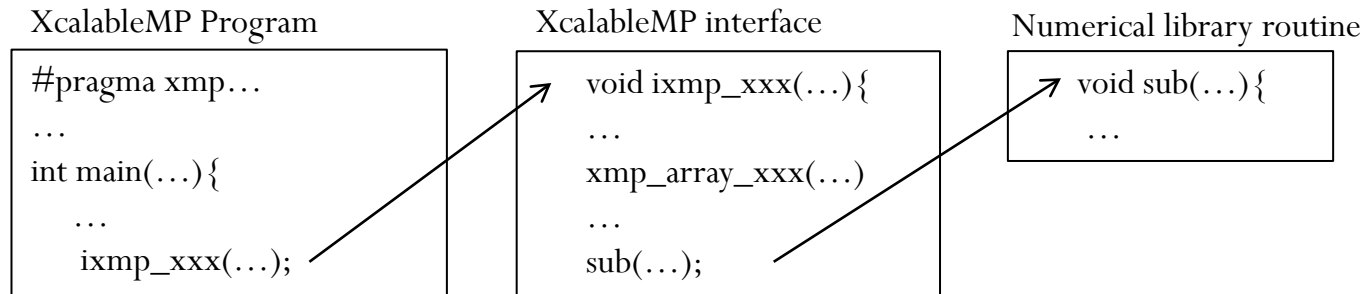


Usage of parallel numerical libraries for XMP



□ Design of the interface

1. Parallel numerical library routines can be invoked as an XMP procedure through the XMP interface procedure.
2. When the parallel numerical library routine needs information on global distributed arrays, the interface extracts it from the descriptor **using some inquiry routines** provided by XMP runtime library and passes it to the numerical library routine as arguments.
3. The interface **does not affect the behavior of numerical library** routines expect for restrictions concerning the XMP specification.



<Inquiry routines>

`xmp_array_xxx(xmp_desc_t d, ..)`

`xmp_align_xxx(xmp_desc_t d, ..)`

`xmp_dist_xxx(xmp_desc_t d, ..)`

❑ Sample program for ScaLAPACK (MPI programming model)

Program code

```
blacs_pinfo_(&mypnum,&nprocs);
blacs_get_(&icontxt,&what,&contxt);
blacs_gridinit_(&contxt,order,&nrow,&npcol);
blacs_gridinfo_(&contxt,&nrow,&npcol,&myrow,&mycol);

    . . .

for (i=0;i<nrow/nrow;i++){
  for (j=0;j<ncol/npcol;j++){
    a[i][j]=a0[myrow+i*nrow][mycol+j*npcol]; ← Calculate by local coordinate !
  }
}

    . . .

desca[0]= array information;
....
descb[8]=array information;
pdgesv_(&n,&nrhs,a,&ia,&ja,desca,ipiv,b,
      &ib,&jb,descb,&info);

    . . .

blacs_gridexit_(&contxt);
blacs_exit_(&doneflag);
```

Flow of execution

Initialization of communication
process of a numerical library



Matrix distribution



Executing solver



Finalization of communication
process of a numerical library

Sample program for ScaLAPACK (XcalableMP programming model)

Program code

```
#pragma xmp nodes p(nprow,npcol)
    . . . . .
#pragma xmp align a[i][j] with t(i,j)
#pragma xmp align ipiv[i][j] with t1(i,j)
#pragma xmp align b[i] with t(*,i)
    . . .

blacs_pinfo_(&myppnum,&nprocs);
blacs_get(&icontxt,&what,&contxt);
blacs_gridinit(&contxt,order,&nprow,&npcol);
blacs_gridinfo(&contxt,&nprow,&npcol,&myrow,&mycol);
    . . .
#pragma xmp loop (i,j) on t(i,j)
for(i=0;i<ncol;i++){
    for(j=0;j<nrow;j++){
        a[i][j] = a0[i][j];
    }
}
    . . .
desca[0]=array information;
...
descb[8]=array information;
ixmp_pdgesv(&n,&nrhs,a,&ia,&ja,xmp_desc_of(a),ipiv,b,
            &ib,&jb,xmp_desc_of(b),&ictxt,&info);
    . . .
blacs_gridexit(&contxt);
blacs_exit_(&doneflag);
```

Calculate by global coordinate !
Reduce distribution programming
cost by directives !

Improve productivity !

Flow of execution

Definition of data distribution



Initialization of communication
process of a numerical library



Matrix distribution
(Execute by directives)



Executing solver



Finalization of communication
process of a numerical library

❑ Descriptor of XcalableMP

Descriptor : A key to retrieve information of specified global array, template, or node array.

xmp_desc_t : The type which of the descriptor is implementation-dependent.

xmp_desc_of : Built-in operator which is evaluated to a descriptor.

xmp_desc_of has a operand which is the name of either a global array, a template or a node array.

The resulting descriptor can be used as an input argument of the inquiry procedures.

<Inquiry routines(XcalableMP C)>

xmp_array_xxx(xmp_desc_t d, ..)

xmp_align_xxx(xmp_desc_t d, ..)

xmp_dist_xxx(xmp_desc_t d, ..)

xmp_template_xxx(xmp_desc_t d, ..)

xmp_nodes_xxx(xmp_desc_t d, ..)

❑ Experiment

Table 1 Experimental environment

	PC Cluster(node environment)
CPU	Intel Xeon X5670(2.93GHz 6core) *2
Memory	24GB
Network	1Gbps Ethernet*2 + Infiniband 4x QDR
OS	Linux 2.6.18
MPI	Intel MPI Library 4.0 Update2
C Compiler	gcc4.1.2
XMP	C language version
ScaLAPACK	Compile netlib original programs by using “gfortran -O3”
BLACS	Compile netlib original programs by using “gcc -O3”
BLAS	Compile netlib original programs by using “gfortran -O3”

❑ Experiment

Table2 Performance results of the sample program

	MPI program	XMP program
Data distribution	0.0052sec	0.0055sec
Solver(PDGESV)	1.02sec	1.02sec

The rank of coefficient matrix: 2000

Data distribution method: Cyclic

Execution nodes: 1

Execution processes: 4

⇒ The performance of XMP program is comparable with that of MPI one.

□ Summary and Future work

- We show that programming productivity is improved by using the XMP programming model and the interface of XMP.
- Currently, we have researched just two parallel numerical libraries. For the other parallel numerical libraries, we need to research the interface of XMP.

Thank you very much for kind attention.