

XcalableMP

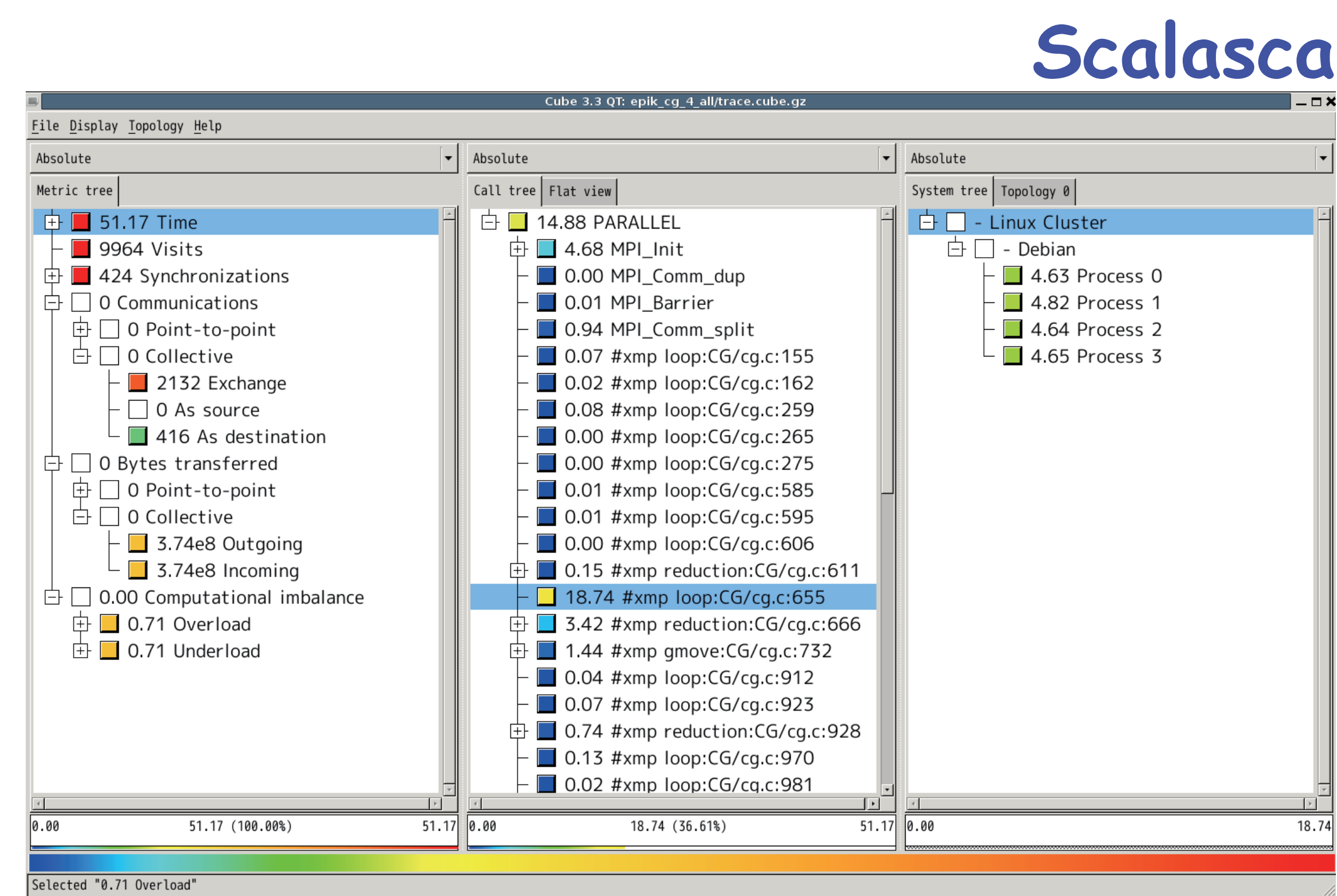
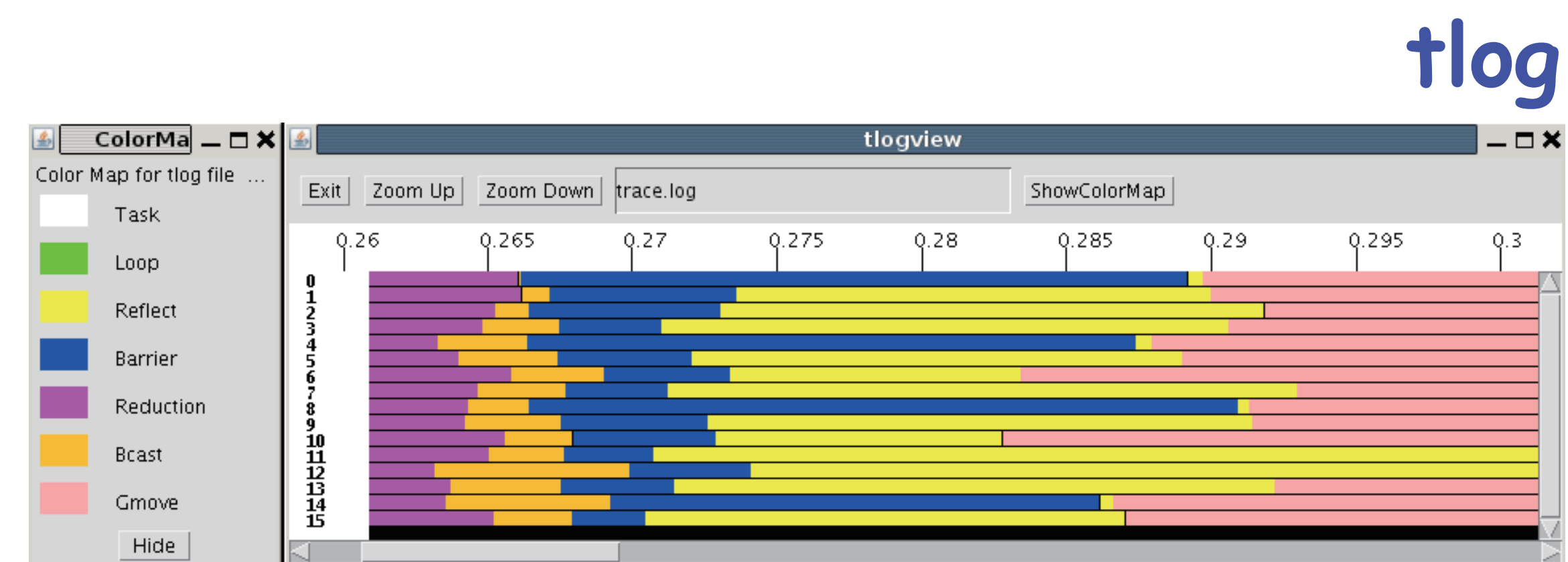
Directive-Based Language eXtension for Scalable Parallel Programming



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

Language Status

- Omni XcalableMP compiler 0.5.3 for C is available from University of Tsukuba
 - Supported platforms are linux cluster, Cray XT5, ...
 - Interface of **tlog** and **Scalasca** profiling tools
 - XcalableMP specification ver. 1.0 is available
 - For accelerators (GPU, etc)
- More details → **Center for Computational Sciences, University of Tsukuba(#923@4F)**
- For K computer
XcalableMP will be used to program to K computer



Examples

Laplace Solver by global-view programming

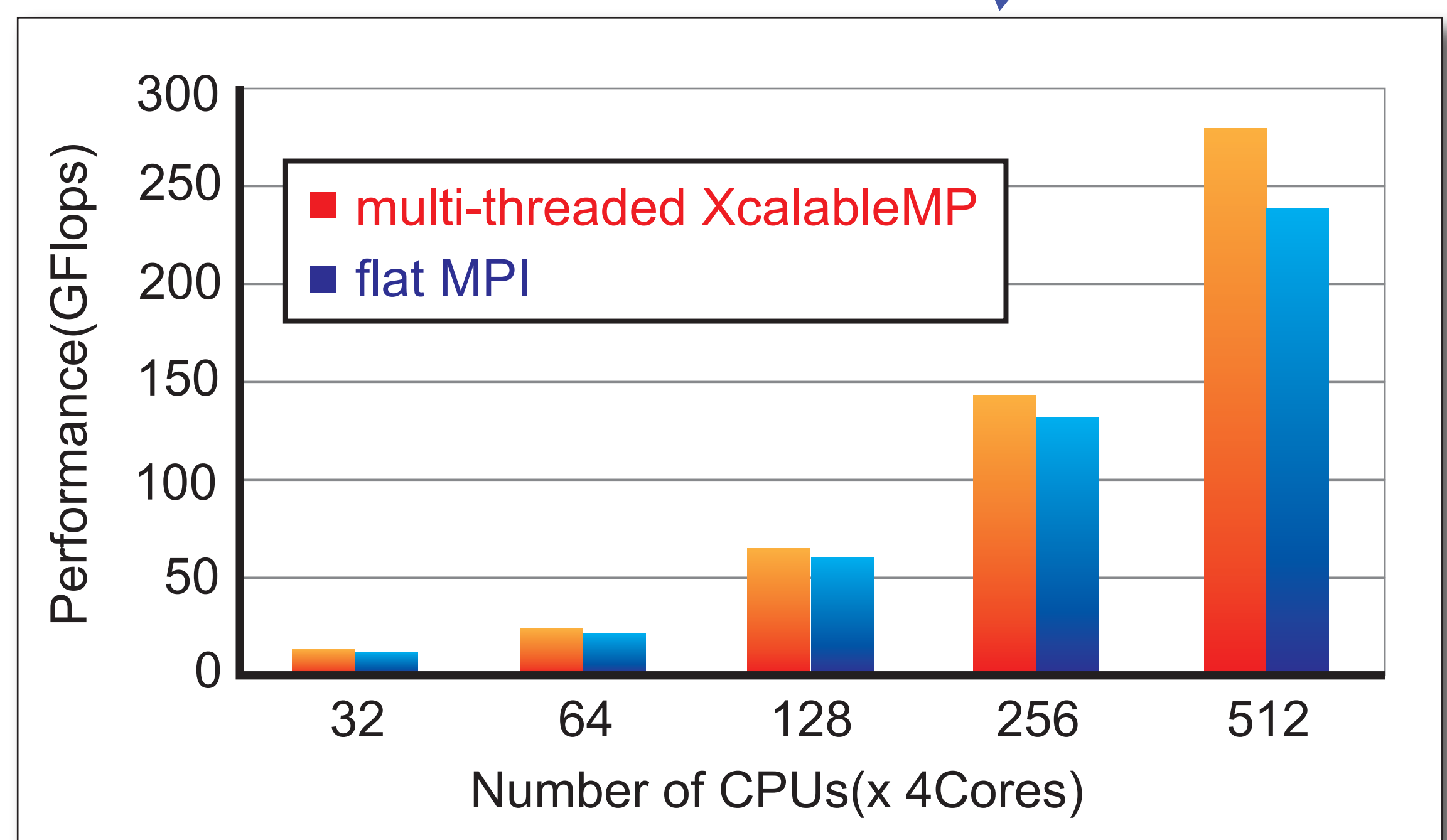
Experimental environment is the T2K Tsukuba system

```
#pragma xmp nodes p(N_Y, N_X)
#pragma xmp template t(O:YSIZE, O:XSIZE)
#pragma xmp distribute t(block, block) onto p
#pragma xmp align u[y][x] with t(x, y)
#pragma xmp align uu[y][x] with t(x, y)
#pragma xmp shadow uu[1:1][1:1]
:
#pragma xmp loop (x, y) on t(x, y) threads
for(y = 1; y < YSIZE-1; y++)
for(x = 1; x < XSIZE-1; x++)
uu[y][x] = u[y][x];

#pragma xmp reflect uu

#pragma xmp loop (x, y) on t(x, y) threads
for(y = 1; y < YSIZE-1; y++)
for(x = 1; x < XSIZE-1; x++)
u[y][x] = (uu[y-1][x] + uu[y+1][x] +
uu[y][x-1] + uu[y][x+1])/4.0;
```

- Define **two dimensional process grid**
- Define **shadow area** and its width
- Specify **additional thread parallelization**
- Synchronize data only on **shadow area**



- XcalableMP also supports hybrid parallelization for multicore cluster
- Little modification from serial source code

Integer Sort of NPB by local-view programming

```
int key[SIZE_OF_BUFFERS];
#pragma xmp coarray key
:
#pragma xmp barrier
for( i=0; i<comm_size; i++ )
key[recv_displ[i]:count[i]]:[i]
= buff[send_displ[i]:count[i]];
#pragma xmp barrier
```

- Define **Co-array**
- Execute **Barrier**
- Exchange data by using **Co-array**

The key[start:length]:[N] means elements from the key[start] to the key[start+length-1] located on compute node N

