**What’s XcalableMP?**

- 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., and Kyusyu U.), research labs (RIKEN, NIFS, JAXA, and JAMSTEC/ES), and industries (Fujitsu, NEC, Hitachi) in Japan.
- Omni XcalableMP compiler is developed in "Seamless and Highly-productive Parallel Programming Environment for High performance computing" project funded by MEXT in Japan.

**Implementation Status**

- XcalableMP specification ver. 1.0 is available.
- Omni XcalableMP compiler ver. 0.5.3 is available from University of Tsukuba.
- Supported platforms are Linux cluster, Cray platform,..
- Interface of Scalasca & tlog profiling tools.
- For accelerators (GPU, etc).
- XcalableMP Parallel I/O.
- Interface of MPI library.
- For K computer.

XcalableMP will be used to program to K computer.

**Programming Model**

**Language Features**

- Language extension of C99 and Fortran 95.
- SPMD as a basic execution model.
- 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 HPF).

**Global-view Programming**

- Supports typical parallelization based on the data parallel paradigm and work mapping.

**Example:**

```fortran
int a[12];
#pragma xmp loop on t(i) reduction(+s)
for(i = 1; i < 10; i++) {
    s += a[i];
}
```

**Local-view Programming**

- Also includes Co-Array Fortran like feature.

**Example:**

```fortran
double a[5], b[5];
#pragma xmp coarray a   // Declaration
b[0:2] = a[3:2]:[1];       // Communication
```

**Benchmarks**

**Laplace Solver by Global-view**

- Define two-dimensional process grid.
- Define shadow area and its width.
- Specify additional thread parallelization.
- XcalableMP also supports hybrid parallelization for multicore cluster.
- Synchronize data only on shadow area.

**Example:**

```fortran
int a[YSIZE][XSIZE];
#pragma xmp nodes p(0, Y, X)
#pragma xmp loop on t(i) reduction(+s)
for(i = 1; i < XSIZE-1; i++)
    for(j = 1; j < YSIZE-1; j++)
        s += a[i][j];
```

**Integer Sort of NPB by Local-view**

- Define two-dimensional process grid.
- Define shadow area and its width.
- Specify additional thread parallelization.
- XcalableMP also supports hybrid parallelization for multicore cluster.
- Synchronize data only on shadow area.

**Example:**

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

**For more information, please visit**

- T2K Open Supercomputer Alliance (#5007@Level 6).
- Center for Computational Sciences, University of Tsukuba (#923@Level 4).

**Performance (GFlops)**

<table>
<thead>
<tr>
<th>Number of Cores</th>
<th>Omni XcalableMP</th>
<th>XcalableMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1000</td>
<td>1200</td>
</tr>
<tr>
<td>32</td>
<td>1500</td>
<td>1800</td>
</tr>
<tr>
<td>64</td>
<td>2000</td>
<td>2500</td>
</tr>
<tr>
<td>128</td>
<td>2500</td>
<td>3000</td>
</tr>
<tr>
<td>256</td>
<td>3000</td>
<td>3500</td>
</tr>
</tbody>
</table>

**Number of CPUs (x 4Cores)**

- Flat MPI
- multi-threaded XcalableMP

**Number of Cores**

- 16 32 64 128 256