**XcalableACC**

**Directive-based language extension for accelerated clusters**

### What's XcalableACC?

**XcalableACC (XACC)** is a PGAS language for accelerated clusters, which is a directive-based language extension of C and Fortran based on an XcalableMP PGAS language by using OpenACC.

XACC supports typical parallelization under “global-view model” programming and enables parallelizing the original sequential code by using simple directives.

XACC also includes coarray features for “local-view model” programming.

### Global-view model

Array `a[]` is distributed onto an accelerator on each node.

```c
int a[12];
#pragma xmp nodes p(3)
#pragma xmp template t(0:11)
#pragma xmp distribute t(block) onto p
#pragma xmp align a[i] with t(i)
#pragma acc enter data copyin(a)
```

Array `a[]` is distributed onto an accelerator on each nodes.

```c
#pragma xmp nodes p(3)
#pragma xmp loop on t(i)
#pragma acc parallel loop
for(i = 0; i < 12; i++)
a[i] = func(i);
```

**Data mapping**

<table>
<thead>
<tr>
<th>Global index</th>
<th>Distributed Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 10 11</td>
<td>0 1 2 4</td>
</tr>
</tbody>
</table>

**Work mapping**

```c
#pragma xmp nodes p(3)
#pragma xmp loop on t(i)
#pragma acc parallel loop
for(i = 0; i < 12; i++)
a[i] = func(i);
```

### Local-view model

XACC C language defines array section and codimension equivalent to CAF.

```c
x[start:length][node_number]
```

Above code means length elements from `x[start]` to `x[start+length-1]` located on `node_number` are referred.

```c
double a[5][*], b[5][*]; // Declare
#pragma acc declare create(a, b)
if(me==2){
#pragma acc host_data use_device(a, b)
b[0:2][1] = a[3:2]; // Put
}
```

**Productivity (Source lines of Code)**

<table>
<thead>
<tr>
<th></th>
<th>HIMENO</th>
<th>NPB CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>XcalableACC (a, b)</td>
<td>198 (34, 9)</td>
<td>609 (48, 20)</td>
</tr>
<tr>
<td>MPI + OpenACC (b)</td>
<td>328 (13)</td>
<td>772 (24)</td>
</tr>
</tbody>
</table>

(a) Number of XcalableMP directives
(b) Number of OpenACC directives

### Evaluation

**Benchmark**

- HIMENO is a stencil code which evaluates the performance of incompressible fluid analysis code.
- NPB CG is a benchmark to solve minimum eigenvalue of symmetric and positive definite sparse matrix using the conjugate gradient method.

**Performance on HA-PACS/TCA system**

**Node Specification**

- CPU : Ivy Bridge E5-2680v2 x 2
- Mem. : DDR3 128GB 59.7GB/s x 2
- GPU : NVIDIA K20X GDDR5 6GB 250GB/s x 4
- Network : InfiniBand 8GB/s

For more information, please visit **RIKEN AICS (#2521)** and **Univ. of Tsukuba (#1927)**