Implementation and Performance of NPB

In order to investigate the performance of parallel programs written in XcalableMP, we have implemented NAS Parallel Benchmarks, the Integer Sort (IS) benchmark and the Conjugate Gradient (CG) benchmark.

Integer Sort

The IS benchmark tests a sorting operation that is important in particle method codes. IS is based on the “bucket sort” algorithm to sort large arrays of integers.

```
#pragma xmp coarray key_buff2
...  
#pragma xmp loop on t(i)
for(i=0; i<NUM_KEYS; i++)
    bucket_size[key_array[i] >> shift]++;
...  
for(i=0;i<NUM_PROCS;i++)
  key_buff2[a[i]:b[i]].i = key_buff1[c[i]:d[i]];
```

First line is to declare coarray for local view programming. Next “loop” directive and “for statement” are to create a histogram which is used by the sort. The array key_array[] is a distributed array. Last “for statement” is to exchange the keys at each node with using coarray. Array a[], b[], c[] and d[] are information of send points and receive points of each node.

Conjugate Gradient

In the CG benchmark, the conjugate gradient method is used to compute an approximation to the smallest eigenvalue of a large, sparse, symmetric positive definite matrix.

```
#pragma xmp distribute(block,block) on pros
...  
#pragma xmp loop on t(*,j)
for (j=1; j<=lastrow-firstrow+1; j++) {
    sum = 0.0;
    for (k=rowstr[j]; k<rowstr[j+1]; k++) {
        sum = sum + a[k]*p[colidx[k]];
    }
    w[j] = sum;
}
#pragma xmp reduction(+:w) on p(*,:)
#pragma xmp gmove
q[:] = w[:];
```

Two-dimensional template is declared in first line and one-dimensional working arrays(p[], q[], w[]) are aligned to one dimension of the template. Index arrays(rowstr[], colidx[]) are calculated to refer the array a[] before entering the loop in each node. This operation enables the nested loop statements to be calculated which causes an equivalent processing with the MPI version of CG. The “gmove” directive copies all elements of array w[] to array q[] with different distributions.

Performance Results

<table>
<thead>
<tr>
<th></th>
<th>PC Cluster</th>
<th>T2K Tsukuba</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Core2 Quad CPU Q9650 3.0GHz</td>
<td>AMD Opteron Quad Core 8000 series 2.3GHz</td>
</tr>
<tr>
<td>Network</td>
<td>Gigabit Ethernet</td>
<td>Infiniband DDR (4 rails)</td>
</tr>
</tbody>
</table>

Integer Sort

Conjugate Gradient

The results indicate that the performance of XcalableMP is comparable to that of MPI.

Reference