This paradigm features functions for array distribution and work typical scientific applications using a concept similar to OpenMP. XcalableMP introduces simple, but effective features to describe languages as to describe distributed memory parallel program, systems, and the directives extend the C and Fortran base parallel programming for distributed memory multiprocessor and Fortran program. This specification provides a model of collection of compiler directives, runtime library routines that can however limited to shared memory architectures.

On the other hand, OpenMP is widely used on shared memory programming models have not been commonly used any more. The most important feature of OpenMP from the perspective of programmability is to enable parallelization with simple directives that helps users extend their codes relatively easily from sequential ones. The target platform of OpenMP is CPUs. The most important feature of OpenMP is that it can be executed as MPI processes. These features can be coded using directives similar to OpenMP.

The specification has been being designed by XcalableMP Specification Working Group which consists of members from academia and research labs to industries in Japan. Features of XcalableMP are summarized as follows:

- **XcalableMP supports typical parallelization based on the data parallel paradigm and work mapping for loop on parallel processes, which are normally executed as MPI processes. These features can be coded using directives similar to OpenMP.**
- **XcalableMP Application Program Interface (XcalableMP API) is a collection of compiler directives, runtime library routines that can be used to specify distributed-memory parallel programming in C and Fortran program. This specification provides a model of parallel programming for distributed memory multiprocessor systems, and the directives extend the C and Fortran base languages as to describe distributed memory parallel program, as in OpenMP.**
- **XcalableMP introduces simple, but effective features to describe typical scientific applications using a concept similar to OpenMP. This paradigm features functions for array distribution and work mapping for loop on parallel processes, which are normally executed as MPI processes. These features can be coded using directives similar to OpenMP.**
- **We want better solutions!!**
- **We need better solutions!!**
- **Current solution for programming clusters?**
  ```c
  #pragma xmp template T[10] 
  #pragma xmp distributed T[block]
  int array[10][10];
  main()
  int i,j,res;
  #pragma xmp aligned array[i][j] T[10]
  #pragma xmp loop on j() reduce:
  for(i=0; i<10; ++i) 
  for(j=0; j<10; ++j) 
  array[i][j] = func(i, j);
  res = array[i][j];
  ```
- **Only way to program is MPI, but MPI programming seems difficult … we have to rewrite almost entire program and it is time-consuming and hard to debug… mmm**
- **Features of XcalableMP are summarized as follows:**
- **XcalableMP also includes CAF-like PGAS (Partitioned Global Address Space) feature as “local view” programming.**
- **Extention of existing base languages with directives is useful for rewriting cost and education cost. XcalableMP APIs are defined on C and Fortran 95 as a base language.**
- **For flexibility and extensibility, the execution model allows to combine with explicit MPI coding for more complicated and tuned parallel codes and libraries. For multi-core and SMP clusters, OpenMP directives can be combined into XcalableMP for thread programming inside each node as a hybrid programming model.(Under discussion)**

XcalableMP

Directive-based Language eXtension for Scalable and Performance-aware Parallel Programming

University of Tsukuba
Programming model of XcalableMP

The parallel execution model of XcalableMP is a Single Program Multiple Data (SPMD) model. As in MPI, all parallel processes in nodes start their execution from the same main function. When the thread encounters XcalableMP directives, the synchronization and communication occurs between nodes. That is, no synchronization and communications happen without directives. In this case, the program does duplicated execution of the same program on local memory in each node. As default, data declared in the program is allocated in each node, and is referenced locally by threads executed in the node.

The global-view programming model is useful when, starting from sequential version of the program; the programmer parallelizes it in data-parallel model by adding directives incrementally with minimum modifications. As these directives can be ignored as a comment by the compilers of base languages (C and Fortran), an XcalableMP program derived from a sequential program can preserve the integrity of original program when it is run sequentially.

The global-view programming model shares major concepts with HPF. Nodes directive declares a node array to express a set of nodes. The programmer describes the data distribution of data shared among the nodes by data distribution directives. To specify the data distribution, the template is used as a dummy array distributed on nodes. A distributed array is declared by aligning the array to the template by align directive.

The work mapping in loop iteration is described by the loop directive as in OpenMP. Loop construct maps iterations to the node where referenced data is located. Template is used to specify the mapping of iteration. By using the same template used for the data distribution, iterations are assigned to the node of the data. It should be noted that in XcalableMP the programmer must control all data reference required computations done locally by any appropriate directives.

In order to describe communication, the gmove construct is a powerful operation in global-view programming in XcalableMP: It copies data of a distributed array in global-view. This directive is followed by the assignment statement of scalar value and array sections.

In addition to the "global view" programming described above, XcalableMP also includes CAF-like PGAS (Partitioned Global Address Space) feature as "local view" programming. XcalableMP adopts coarray notations as an extension of languages for local view programming. In case of Fortran as the base language, most coarray notations are compatible to that of Coarray Fortran(CAF) expect that the task constructs are used for task parallelism. In order to use coarray notations in C, we propose some language extension of the language. A coarray is declared by the coarray directive in C.

References


This research is carried out as a part of “Seamless and Highly-productive Parallel Programming Environment for High-performance computing” project funded by Ministry of Education, Culture, Sports, Science and Technology, JAPAN.