Toward integration of multi-SPMD programming model and advanced cyberinfrastructure platform

Miwako Tsuji
RIKEN Center for Computational Sceince

Agenda : In this paper, we introduce a multi SPMD (mSPMD) programming model, which combines a workflow paradigm and a distributed parallel programming model. Then, we discuss about current issues in the mSPMD regarding data transfer. At the end, we describe future plan to integrate the mSPMD and advanced cyberinfrastructure platform (ACP).

A multi-SPMD programming model

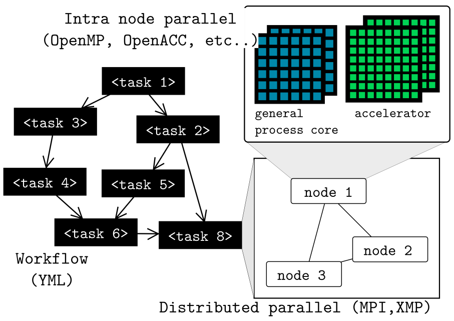

Figure 1: Overview of the multi SPMD programming model

In order to address reliability, fault tolerance, and scaling problems in future large scale systems, we have proposed a multi SPMD (mSPMD) programming model and have been developing a development and execution environment for the mSPMD programming model[4, 3]. The mSPMD programming model combines a workflow paradigm and a distributed parallel programming model for future large scale systems, which would be highly hierarchical architecture with nodes of many-core processors, accelerators and other different architectures.

Figure 1 shows the overview of the mSPMD programming model. Each task in a workflow can be a distributed parallel program. A PGAS language called XcalableMP [5] has been supported to describe tasks in a workflow. To describe and manage dependency between tasks, we adopt YML[1] workflow development and execution environment.

Since tasks in a workflow can be executed in parallel in the mSPMD programming model, some "heavy" tasks can be executed in parallel to speed up the workflow. Another advantage of the mSPMD is that a huge distributed parallel program can be decomposed into several moderate sized sub-programs based on the workflow paradigm to avoid the communication overhead.

Current "BigData" issues in mSPMD programming model

One of important pieces that the mSPMD programming model is missing is an intelligent implementation of the data transfer method between tasks. As shown in Figure 2, our current implementation of the data transfer between tasks strongly relies on a network file system (NFS) and MPI-IO functions. After a task writes a data to a NFS, the other tasks which use the data are started and read the data from the NFS. The advantage of using NFS and MPI-IO are (1) portability (2) auto check-pointing and (3) ease of use for application developers since the MPI-IO function calls can be generated automatically based on XcalableMP declarations.

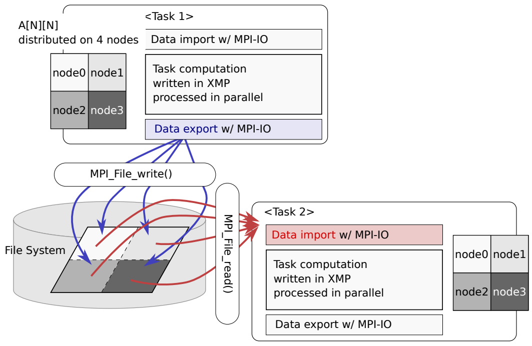

Figure 2: Data transfer between tasks

The disadvantages of using NFS are speed and performance instability. To overcome this, we are investigating the combination of file-IO and data servers[2]. Additionally, as future works, we will investigate advanced software and hardware infrastructures such as ADIOS library, data-compression hardware, burst buffer.

Toward integration of multi-SPMD programming model and advanced cyberinfrastructure platform

In addition to the advantages described above, the mSPMD can combine several parallel libraries and existing parallel programs easily to compose a complex application for a heterogeneous architecture.

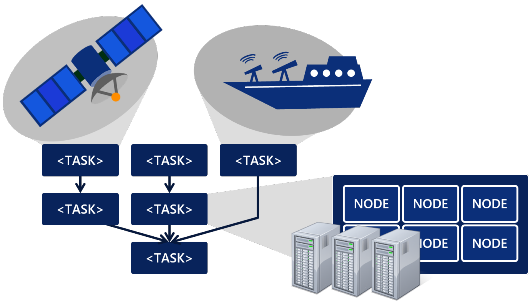

Figure 3: Integration of mSPMD and ACP

The mSPMD may provide a new method for data processing and data logistic among different systems. So far, we have focused on the data processing in an HPC cluster and the data dependencies between them. However, we consider the workflow paradigm is also useful to manage data dependencies in/from cyber-physical systems. An unified workflow approach to describe dependencies among data generations, data processings, HPC simulations to update data, etc... should be important to develop complicated applications. As future works, we will integrate the mSPMD programming model and ACP. As shown in Figure 3, our workflow paradigm will orchestrate data dependencies not only between traditional distributed parallel programs, but also from/to various sensing and processing devices.

References

[1] Olivier Delannoy, Nahid Emad, and Serge Petiton. Workflow global computing with yml. In *The 7th IEEE/ACM International Conference on Grid Computing*, pages 25–32, 2006.

[2] Thomas Dufaud, Miwako Tsuji, and Mitsuhisa Sato. Design of data management for multi-spmd workflow programming model. In *Proceedings of the 4th International Workshop on Extreme Scale Programming Models and Middleware, SC18*. ACM, 2018.

[3] Miwako Tsuji, Serge Petiton, and Mitsuhisa Sato. Fault tolerance features of a new multi-spmd programming/execution environment. In *Proceedings of the First International Workshop on Extreme Scale Programming Models and Middleware, SC15*, pages pp.20–27 doi:10.1145/2832241.2832243. ACM, 2015.

[4] Miwako Tsuji, Mitsuhisa Sato, Maxime Hugues, and Serge Petiton. Multiple-SPMD programming environment based on PGAS and workflow toward post-petascale computing. In *Proceedings of the 2013 International Conference on Parallel Processing (ICPP-2013)*, pages 480–485. IEEE, 2013.

[5] XcalableMP. http://www.xcalablemp.org/.