11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20 - 25, 2014, Barcelona, Spain

MPI/OmpSs programming model and its application in simulation code

Rosa M Badia[1,4], Eduard Ayguadé[2,5] and Jesús Labarta[3,5]

[1] Barcelona Supercomputing Center, Jordi Girona 29, 08034-Barcelona, SPAIN,
rosa.m.badia@bsc.es, www.bsc.es
[2] Barcelona Supercomputing Center, Jordi Girona 29, 08034-Barcelona, SPAIN,
eduard.ayguade@bsc.es, www.bsc.es
[3] Barcelona Supercomputing Center, Jordi Girona 29, 08034-Barcelona, SPAIN,
jesus.labarta@bsc.es, www.bsc.es
[4] Artificial Intelligence Research Institute (IIIA), Spanish National Research Council (CSIC)
[5] Universitat Politècnica de Catalunya

Key Words: *Parallel programming, programming models, simulation codes*

OmpSs is a task-based programming model that aims to provide portability and flexibility to sequential codes while the performance is achieved by the dynamic exploitation of the parallelism at task level [1,2].

OmpSs targets the programming heterogeneous and multi-core architectures, and offers an asynchronous parallelism in the execution of the tasks. The main extension of OmpSs, now incorporated in the recent OpenMP 4.0 standard, is the concept of data dependences between tasks. Tasks in OmpSs are annotated with data directionality clauses that specify the data used by it, and how it will be used (read, write or read&write). This information is used during the execution by the underlying OmpSs runtime to control the synchronization of the different instances of tasks by creating a dependence graph that guarantees the proper order of execution. This mechanism provides a simple way to express the order in which tasks must be executed, without the need of adding explicit synchronization.

Additionally, OmpSs syntax offers the flexibility to express that given tasks can be executed on heterogeneous target cores (i.e., regular processors, GPUs, or FPGAs). The runtime is able to schedule and run those tasks, taking care of the required data transfers and synchronizations that are needed. What is more, more than one implementation can be provided for a given task and the runtime will be able to chose the best suited one [4].

Additionally, OmpSs can be nicely combined with MPI to provide a powerful programming model to enhance current MPI applications. Due to its asynchronous nature and look-ahead capabilities, MPI/OmpSs is a promising programming model approach for future exascale systems, with the potential to exploit unprecedented amounts of parallelism, while coping with memory latency, network latency and load imbalance. Many large-scale applications are already seeing very positive results from their ports to MPI/OmpSs (see EU projects Montblanc, DEEP, TEXT) [3].

The integration of OmpSs with MPI facilitates the migration of current MPI applications and improves the performance of these applications by overlapping computation with communication between tasks. OmpSs is also surrounded by a family of performance and debugging tools.

The talk will present the basics of OmpSs programming model: its syntax, main features of its runtime, and main target computing platforms: multicore, GPUs and clusters, as well as MPI/OmpSs. The talk will be illustrated with examples and results in different target architectures. Examples of applications parallelized with MPI/OmpSs from the simulation field will also be presented.

REFERENCES

[1] Alejandro Duran, Roger Ferrer, Eduard Ayguadé, Rosa M. Badia and Jesús Labarta, "A Proposal to Extend the OpenMP Tasking Model with Dependent Tasks", in the International Journal of Parallel Programming, Volume 37, Issue 3 (2009), pages 292-305.

[2] Alejandro Duran, Eduard Ayguadé, Rosa M. Badia, Jesús Labarta, Luis Martinell, Xavier Martorell and Judit Planas, "OmpSs: A proposal for programming heterogeneous multi-core architectures", in Parallel Processing Letters, Volume 21, Issue 2, pp. 173 - 193, June 2011.

[3] Jesús Labarta, Vladimir Marjanovic, Eduard Ayguadé, Rosa M. Badia and Mateo Valero. Hybrid Parallel Programming with MPI/StarSs. In Applications, Tools and Techniques on the Road to Exascale Computing (Proceedings ParCo 2011), pp. 621-628, IOS Press, May 2012.

[4] Judit Planas, Rosa M. Badia, Eduard Ayguadé and Jesús Labarta. Self-Adaptive OmpSs Tasks in Heterogeneous Environments. In proc. of 27th IEEE International Parallel & Distributed Processing Symposium, IPDPS 2013.