

**Book Review:****Using OpenMP: Portable Shared Memory Parallel Programming**

Barbara Chapman, Gabriele Jost, Ruud van der Pas

The MIT Press, 2008

ISBN-13: 978-0-262-53302-7

Parallel computing is no longer something reserved for the area scientific applications or high performance computing. Today, multiprocessing platforms are widely available with multiple multi-core processors sharing memory. Moreover, it is almost a fact that all software will tend to be parallel at least in multiprocessors, since there is no other way to get better final performance for applications [1]. In this context, OpenMP is one of the best choices at the level of implementation for parallel computing in multiprocessors. Currently, there are multiple options for parallel computing in multiprocessors, but many of which are proprietary and/or conceptually weak. In this sense, OpenMP is one of the few options that do not have been designed to take advantage of the current multicore era. OpenMP is conceptually sound and with a extensive experience in the field of parallel computing/high-performance computing. OpenMP is defined by the ARB (Architecture Review Board) composed of multiple recognized organizations [2] including vendors such as AMD, Cray, Fujitsu, HP, IBM, Intel, NEC and Microsoft. On the other hand, OpenMP is implemented by most major source language compilers since many years ago.

The book *Using OpenMP: Portable Shared Memory Parallel Programming*, presents OpenMP more than appropriately, from the perspective of high performance computing, which traditionally has provided most (if not all) the parallel software development strategies. In this sense, we can identify different points of view related to OpenMP in the book:

- The first three chapters could be regarded as introductory. These chapters provide from a general introduction to parallel computing on multiprocessors to specific simple examples of (multithreaded) parallel programs with OpenMP.
- Chapter four presents both OpenMP directives, library functions and environment variables of the specification. Some of the OpenMP terminology is explained near the beginning and code examples are given in fragments related to each directive/function explained.
- Chapters five and six should be considered as the most directly related to performance and performance optimization of parallel programs with OpenMP. The contents of these chapters are essential from the perspective of parallel computing as well as the primary companion to the specification itself, since they focus on the rationale for parallelization. Computation time reduction and improving scalability are analyzed. It is worth noting that scalability should be taken into account given the growth in the number of cores available on computers as time progresses.
- Chapter seven is related to debugging parallel programs based on shared memory parallel computers. While you can not quite give a complete list of all the potential problems of all applications, the most common problems are referred to. In this chapter, some considerations are specific to OpenMP and others are specific to the model of shared memory parallel computing.
- Chapter eight presents important aspects of OpenMP implementation. Specifically, the specification is interpreted and/or explained in terms of the way OpenMP compilers to recognize and, more importantly, translate the specification into executable code with multiple threads. Some details of some compilers are given, but the presentation is not based on a particular compiler.
- In the last chapter, the authors estimate the inclusions or changes in the specification that were deemed most likely in late 2007/early 2008. It is important to note that tasks were not included in the OpenMP specification by the time the book was written, but are discussed in this chapter and finally included in the current specification, which is Version 3.0 (since May 2008) [3].

The style of the book has significant changes along the chapters, which make interesting its reading. Some chapters are written in teaching style, especially related to

- Concepts implemented in OpenMP or to be implemented, such as tasks (currently included in the specification).
  - Concepts regarding performance or optimization procedures.
  - Concepts related to debugging parallel applications with multiple threads, and the most common problems to be faced.
- Other chapters have a tutorial style, explaining the most important parts of the specification, with specific examples (code fragments), simple yet significant for what they want to show / explain. In some parts, the book can be considered as a reference manual, since it involves the OpenMP specification as an enumeration explained OpenMP directives, library functions and environment variables defined by the specification.

A comparative example of OpenMP with MPI (Message Passing Interface) and pthreads is given in chapter one. As you progress through the chapters, there are some further references to MPI, and almost no other reference to pthreads. Perhaps some extra comparisons and/or comments related to pthreads would clarify the explanations for some guidelines or examples, although it is clear that the book is not about pthreads. Further comments related to pthreads could be useful taking into account the current computer science courses at universities, where operating systems courses often introduce the main concepts related to threads and the scheduling of threads.

Although for the time the book was written tasks were not included in the OpenMP specification, both in chapter six as in the last chapter the most important ideas related to the tasks of the current OpenMP specification are explained. The last chapter in particular explains in great detail the rationale for the existence of the tasks, their usage, and some syntactic ideas which are clear enough to understand the current specification. In this sense, the book is not outdated, at least introduces the most important ideas of parallelization via multiple threads with OpenMP.

In conclusion, this is a book entirely recommended, away from books or proposals of parallelization on shared memory computers that have appeared in recent years, more related to a market advantage than to a strong proposal from a conceptual standpoint. The authors use a broad range of style resources and knowledge of the specification for the presentation of all the material throughout the book.

### References

- [1] H. Sutter, “The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software“, *Dr. Dobbs Journal*, Vol. 30, No. 3., March 2005. Updated version at <http://www.gotw.ca/publications/concurrency-ddj.htm>
- [2] OpenMP.org » About OpenMP and OpenMP.org, <http://openmp.org/wp/about-openmp/>
- [3] OpenMP Architecture Review Board, *OpenMP Application Program Interface*, Version 3.0, May 2008.

Fernando G. Tinetti  
III-LIDI, Facultad de Informática, UNLP  
Investigador Comisión de Inv. Científicas de la Prov. de Bs. As.  
[fernando@info.unlp.edu.ar](mailto:fernando@info.unlp.edu.ar)