Journal of Physics: Conference Series **720** (2016) 012046 doi:10.1088/1742-6596/720/1/012046

Parallel processing implementation for the coupled transport of photons and electrons using OpenMP

Edgardo Doerner

Institute of Physics, Pontificia Universidad Católica de Chile, Vicuña Mackenna 4860, Santiago 7820436, Chile

E-mail: edoerner@fis.puc.cl

Abstract. In this work the use of OpenMP to implement the parallel processing of the Monte Carlo (MC) simulation of the coupled transport for photons and electrons is presented. This implementation was carried out using a modified EGSnrc platform which enables the use of the Microsoft Visual Studio 2013 (VS2013) environment, together with the developing tools available in the Intel Parallel Studio XE 2015 (XE2015). The performance study of this new implementation was carried out in a desktop PC with a multi-core CPU, taking as a reference the performance of the original platform. The results were satisfactory, both in terms of scalability as parallelization efficiency.

1. Introduction

The EGSnrc (**E**lectron-**G**amma-**S**hower) corresponds to a system designed for the MC simulation of coupled transport for photons and electrons and therefore allowing the study of many problems in radiotherapy and dosimetry [1]. EGSnrc may be executed in parallel through a Batch-Queuing-System (BQS) [2]. However, this system presents several disadvantages: (a) any code must be extensively modified in order to support this parallelization, (b) the BQS configuration process is not a trivial task and (c) currently the parallel processing process through a BQS is available only on GNU/Linux operative systems (OS). For these reasons is presented a new parallel computing implementation for EGSnrc through the OpenMP API [3], named OMP_EGS.

2. Materials and methods

The different OpenMP directives were added to the MC code using a modified EGSnrc platform [4], allowing the use of XE2015 integrated in VS2013. The use of these programming tools makes possible an effective and safe implementation of the parallel processing in the OMP_EGS platform. The OMP_EGS platform was developed with the aid of the Intel Parallel Advisor XE component of XE2015, which reveals the potential costs and benefits of parallelism by modeling this behavior in the serial program. Then, it was tuned using the Intel Vtune Amplifier XE profiler.

The performance of the OMP_EGS platform was tested in a 8-core AMD FX8350 desktop PC running under GNU-Linux Debian 7 Wheezy. OMP_EGS was compiled with GNU Fortran 4.7.2 (GFORTRAN), part of the GNU Compiler Collection (GCC) and the Intel Fortran Compiler 15.0 (IFORT), inside XE2015. This implementations were compared against the original EGSnrc BQS parallel processing solution in terms of parallel scalability.

To study the parallel scalability different work load partition schemes were considered. Weak scalability reflects the ability to solve larger problems with more resources whereas strong scalability considers a fixed amount of work while increasing the processing resources. For the first scalability test $5x10^4$ particle histories per processing core were considered. For the later, a total amount of $3x10^5$ initial particles were simulated while increasing the number of used processing cores.

3. Results and Discussion

The results for the weak scalability test are shown in Figure 1. Perfect scalability is achieved when the simulation time remains constant. It can be seen the parallel overhead costs of the parallel processing in both implementations. However, the results show that the OMP_EGS platform is faster than the EGSnrc BQS solution for both compilers. The strong scalability study is depicted in Figure 2. Again, the elapsed times are lower for the OMP_EGS platform. As shown by the results, the parallel scalability of the OMP_EGS platform is satisfying, even surpassing the EGSnrc BQS results.

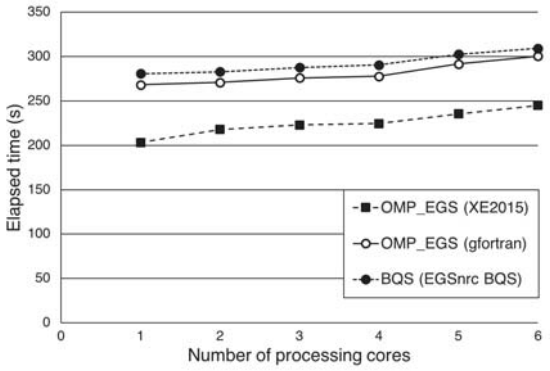

Figure 1. Results for the weak scalability test.

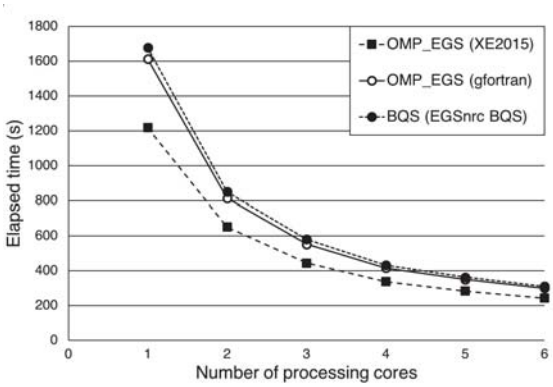

Figure 2. Results for the strong scalability test.

4. Conclusion

OMP_EGS is a portable and compact solution that can be implemented with any compiler that supports OpenMP, between them GFORTRAN and IFORT. Therefore the support of parallel processing depends on the compiler and not on the availability of an external system like a BQS. Thanks to the use of OpenMP most of the serial code is preserved, and moreover, it is possible to recover the serial implementation by turning off the compiler's OpenMP support. This is of great value for debugging purposes. Finally, the performance results have shown that OMP_EGS is a good alternative to the EGSnrc BQS implementation. OMP_EGS obtained better runtimes for all the studied cases.

References

[1] Kawrakow I, Mainegra-Hing E, Rogers D, Tessier F and Walters B 2013 *NRCC Report PIRS-701*
[2] Kawrakow I, Mainegra-Hing E and Rogers D W O 2006 *NRCC Report PIRS-877*
[3] OpenMP Architecture Review Board 2013 Openmp application program interface, version 4.0 URL `http://omp.org/wp/openmp-specifications/`
[4] Doerner E and Hartmann G 2012 *Phys. Med. Biol* **57** 3093