本文给出的编译框架的并行化效率仍然较低,这 3 个程序获得的加速比平均仅为手工并行程序的 53.44%.通过与手工并行代码对比可以看到,自动并行效率较低的主要原因有两个:一是手工并行主要是外层的循环,而自动并行的循环很多是内层的计算量较小的循环,这主要是因为函数调用等影响了自动并行发掘更粗粒度的并行;二是手工并行时程序员知道更多的信息,可以对代码做更大的改动,而自动并行使用的程序静态分析和变换都有一定的局限性.

8 总 结

高性能计算机在现代科学研究中的作用越来越重要且不可替代.异构众核处理器在构建高性能计算机系统时低功耗、高性能的优势使其成为高性能计算领域处理器发展的重要趋势,但其更为复杂的结构也使得原本就存在的编程难的问题更加突出.面向异构系统的自动并行化研究时间较短,因异构系统多种多样,各种面向异构系统并行化研究工作的侧重点也各不相同,其应用效果取决于具体的程序特征和平台特性.本文的研究旨在通过自动化的编译工具减轻编程人员将应用移植到“太湖之光”计算机上的工作负担.本文基于开源编译器 Open64,设计和实现了面向 SW26010 异构众核处理器的一个并行编译框架,能够实现一些程序到异构众核并行程序的转换且获得较好的加速.但该框架目前的实现仍有很多不足,比如只能处理规则的数组及指针访问形式,而且本框架使用的主要是静态程序分析方法,具有一定的局限性,限制了本框架在一些程序中的应用.后续的研究工作:第一,要改进现有方法,弥补不足,比如用机器学习的方法进行收益评估,能够克服静态代价模型的一些不足;第二,异构众核处理器仍在不断发展,面对新的体系结构,需要不断探索新的方法;第三,过程间分析、指针分析、依赖分析等编译领域的通用技术直接影响并行识别的能力,目前这些技术还比较薄弱,需要继续深入研究.虽然本文提出的具体方法大多是针对 SW26010 处理器的,但其基本思路对于其他异构众核架构处理器的并行化编译工作也有一定的参考价值.

References:

[1] Zang DW, Cao Zh, Sun NH. The development of high-performance computing. Science & Technology Review, 2016,34(14):22–28 (in Chinese with English abstract). [doi: 10.3981/j.issn.1000-7857.2016.14.002]

[2] Zheng F, Xu Y, LI HL, Xie XH, Chen ZN. A homegrown many-core processor architecture for high-performance computing. Sciece China (Information Sciences), 2015,45(4):523–534 (in Chinese with English abstract).

[3] Yang GW, Zhao WL, Ding L. “Sunway TaihuLight” and application system. Science (Shanghai), 2017,69(3):12–16 (in Chinese with English abstract).

[4] Sodani A, Gramunt R, Corbal J. Knights landing: Second-generation Intel Xeon Phi Product. IEEE Micro, 2016,36(2):34–46.

[5] Wu G, Greathouse JL, Lyashevsky A. GPGPU performance and power estimation using machine learning. In: Proc. of the IEEE Int’l Symp. on High Performance Computer Architecture. IEEE, 2015. 564–576.

[6] Ju XG, Yang L, Huang ST. An overview of architecture of cell processor. Engineering Journal of Wuhan University, 2010,43(6): 774–779 (in Chinese with English abstract).

[7] Daga M, Aji AM, Feng W. On the efficacy of a fused CPU+ GPU processor (or APU) for parallel computing. In: Proc. of the 2011 IEEE Symp. on Application Accelerators in High-performance Computing. IEEE, 2011. 141–149.

[8] Keckler SW, Dally WJ, Khailany B. GPUs and the future of parallel computing. IEEE Micro, 2011,31:7–17.

[9] Carter NP, Agrawal A, Borkar S. Runnemede: An architecture for ubiquitous high-performance computing. In: Proc. of the IEEE Int’l Symp. on High Performance Computer Architecture (HPCA). Shenzhen: IEEE, 2013. 198–209.

[10] Lee S, Min SJ, Eigenmann R. OpenMP to GPGPU: A compiler framework for automatic translation and optimization. ACM SIGPLAN Notices, 2009,44(4):101–110.

[11] Lee S, Eigenmann R. OpenMPC: Extended open MP programming and tuning for GPUs. In: Proc. of the 2010 ACM/IEEE Int’l Conf. for High Performance Computing, Networking, Storage and Analysis. IEEE, 2010. 1–11.

[12] Han TD, Abdelrahman TS. hi CUDA: High-level GPGPU programming. IEEE Trans. on Parallel and Distributed Systems, 2011, 22(1):78–90.

[13] Baskaran MM, Ramanujam J, Sadayappan P. Automatic C-to-CUDA code generation for affine programs. Compiler Construction, 2010,6011:244–263.

[14] Nishkam R, Yi Y, Tao B, Srimat C. Apricot: An optimizing compiler and productivity tool for X86-compatible many-core coprocessors. In: Proc. of the ICS 2012. Venice: IEEE, 2012. 1–11.

[15] Eichenberger AE, O'Brien JK, O'Brien KM. Using advanced compiler technology to exploit the performance of the cell broadband Engine™ architecture. IBM Systems Journal, 2006,45(1):59–84.

[16] Wang M, Bodin F, Matz S. Automatic data distribution for improving data locality on the cell BE. In: Proc. of the 22nd Int'l Workshop on Languages and Compilers for Parallel Computing (LCPC 2009). Heidelberg: Springer-Verlag, 2009. 247–262.

[17] Chan SC, Gao GR, Chapman B. Open64 compiler infrastructure for emerging multicore/manycore architecture all symposium tutorial. In: Proc. of the IEEE Int'l Parallel & Distributed Processing Symp. IEEE, 2008. 1.

[18] Fu H, Liao J, Yang J, *et al*. The Sunway TaihuLight supercomputer: System and applications. Sciece China (Information Sciences), 2016,59:1–16.

[19] OpenACC-Standard.org. The openacc application programming interface. v2.5. OpenACC-Standard.org, 2015. 1–118.

[20] Zhao J, Zhao RC, Han L. An MPI backend for Open64 compiler. Chinese Journal of Computers, 2014,37(7):1620–1632 (in Chinese with English abstract).

[21] Liu P, Zhao RC, Pang JM, Yao Y. Prioritizing pointer analysis algorithm based on points-to updating. Ruan Jian Xue Bao/Journal of Software, 2014,25(11):2486−2498 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4596.htm [doi: 10.13328/j.cnki.jos.004596]

[22] Yong SH, Horwitz S. Pointer-range analysis. Lecture Notes in Computer Science, 2004,3148:133–148.

[23] Li YB, Zhao RC, Liu XX, Zhao J. Cost model for automatic OpenMP parallelization. Ruan Jian Xue Bao/Journal of Software, 2014,25(Suppl.(2)):101−110 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14028.htm

[24] Wang Z, Tournavitis G, Franke B. Towards a holistic approach to auto-parallelization integrating profile-driven parallelism detection and machine-learning-based mapping. ACM Trans. on Architecture and Code Optimization, 2014,11(1):2.

[25] Tobias G, Torsten H. Polly-ACC transparent compilation to heterogeneous hardware. In: Proc. of the 2016 Int'l Conf. on Supercomputing. ACM, 2016.

[26] Huang PF, Zhao RC, Yao Y, Zhao J. Parallel cost model for heterogeneous multi-core processors. Journal of Computer Applications, 2013,33(6):1544–1547 (in Chinese with English abstract).

[27] Chunhua L. A compile-time OpenMP cost model [Ph.D. Thesis]. Houston: University of Houston, 2007.

[28] Henning JL. SPEC CPU2006 benchmark descriptions. ACM SIGARCH Computer Architecture News, 2006,34(4):1–17.

[29] Li ZH, Zhang HX. Unified algorithm for three-dimensional complex problems covering various flow regimes based on Boltzmann model equations. Sciece China, 2009,(3):414–427 (in Chinese with English abstract).

[30] Li XL, Fu DX, Ma YW. Development of high accuracy CFD software Hoam-OpenCFD. e-Science Technology & Application, 2010,(1):53–59 (in Chinese with English abstract).

[31] He X, Zhou ZM, Liu X. Design and implemention of multi-level heterogeneous parallel algorithm of 3D acoustic wave equation forwarded. Computer Applications & Software, 2014,(1):264–267 (in Chinese with English abstract).

附中文参考文献:

[1] 臧大伟,曹政,孙凝晖.高性能计算的发展.科技导报,2016,34(14):22–28. [doi: 10.3981/j.issn.1000-7857.2016.14.002]

[2] 郑方,许勇,李宏亮,等.一种面向高性能计算的自主众核处理器结构.中国科学(信息科学),2015,45(4):523–534.

[3] 杨广文,赵文来,丁楠,等."神威·太湖之光"及其应用系统.科学(上海),2017,69(3):12–16.

[6] 巨新刚,杨靓,黄士坦.Cell 处理器结构概述.武汉大学学报(工学版),2010,43(6):774–779.

[20] 赵捷,赵荣彩,韩林,等.面向 MPI 代码生成的 Open64 编译器后端.计算机学报,2014,37(7):1620–1632.

[21] 刘鹏,赵荣彩,庞建民,姚远.基于指向更新的优先权指针分析算法.软件学报,2014,25(11):2486–2498. http://www.jos.org.cn/1000-9825/4596.htm [doi: 10.13328/j.cnki.jos.004596]

[23] 李雁冰,赵荣彩,刘晓娴,赵捷.面向 OpenMP 自动并行化的代价模型.软件学报,2014,25(Suppl.(2)):101–110. http://www.jos.org.cn/1000-9825/14028.htm

© 中国科学院软件研究所 http://www.jos.org.cn

[26] 黄品丰,赵荣彩,姚远,赵捷.面向异构多核处理器的并行代价模型.计算机应用,2013,33(6):1544–1547.
[29] 李志辉,张涵信.基于 Boltzmann 模型方程各流域三维复杂绕流问题统一算法研究.中国科学,2009,(3):414–427.
[30] 李新亮,傅德薰,马延文,等.高精度计算流体力学软件 Hoam-OpenCFD 开发.科研信息化技术与应用,2010,(1):53–59.
[31] 何香,周明忠,刘鑫.三维声波方程正演多级异构并行算法设计与实现.计算机应用与软件,2014,(1):264–267.

李雁冰(1989－),男,甘肃陇西人,博士生,主要研究领域为高性能计算,并行编译优化.

赵荣彩(1957－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为高性能计算,并行编译,反编译.

韩林(1978－),男,博士,副教授,CCF 专业会员,主要研究领域为高性能计算,并行编译优化.

赵捷(1987－),男,博士,讲师,CCF 专业会员,主要研究领域为高性能计算,并行编译优化.

徐金龙(1985－),男,博士,讲师,主要研究领域为高性能计算,并行编译优化.

李颖颖(1984－),女,讲师,CCF 专业会员,主要研究领域为高性能计算,并行编译优化.

© 中国科学院软件研究所 http://www.jos.org.cn