

論文 / 著書情報
Article / Book Information

題目(和文)	メニーコアプロセッサにおける高性能計算のための高レベル抽象化
Title(English)	High-level Abstractions for High Performance Computing on Many-core Processors
著者(和文)	星野哲也
Author(English)	Tetsuya Hoshino
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第10941号, 授与年月日:2018年9月20日, 学位の種別:課程博士, 審査員:松岡 聡,増原 英彦,遠藤 敏夫,額田 彰,横田 理央
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第10941号, Conferred date:2018/9/20, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	論文要旨
Type(English)	Summary

(博士課程)
Doctoral Program

論文要旨

THESIS SUMMARY

専攻： 数理・計算科学 専攻
Department of
学生氏名： 星野哲也
Student's Name

申請学位(専攻分野)： 博士 (理学)
Academic Degree Requested Doctor of
指導教員(主)： 特任教授松岡聡
Academic Supervisor(main)
指導教員(副)：
Academic Supervisor(sub)

要旨 (英文 800 語程度)
Thesis Summary (approx.800 English Words)

Over the last decade, processor performance has mainly been improved by increasing the number of cores, and the high-performance computing field is correspondingly shifting from multi- to many-core processors. Many-core processors can combine great computational power with excellent energy efficiency; however, programming them effectively takes significant effort. To reduce the difficulty of programming for many-core processors, several high-level programming models have been proposed.

One approach, commonly used in scientific applications, is to rely on compiler-directive-based language extensions such as OpenMP and OpenACC. These provide sets of directives, namely language extensions that abstract away the architectural characteristics of many-core processors. However, although these abstractions allow compiler-directive-based approaches to simplify the writing of parallel programs for many-core processors, they cannot bring out the best possible performance in many-core processors. This is because some of the essential difficulties in programming for many-core processors still remain, such as load balancing and vectorization, which are intimately dependent on the algorithms and data structures used by the target application.

As an alternative, many domain-specific frameworks have also been proposed. Unlike compiler-directive-based approaches, domain-specific approaches can overcome the above issues by focusing on the specific characteristics of their target applications. However, this also means they lack versatility, and a separate framework must be developed for each application domain.

To address these difficulties, we consider both compiler-directive-based and domain-specific approaches in this paper. To evaluate the performance, productivity, and portability of the OpenACC directive-based approach, we port and optimize both kernel benchmarks and real-world application code to both OpenACC and CUDA. We also use the UPACS computational fluid dynamics (CFD) application as a case study. We find that data structure transformations, which require structural changes to the code, even with a directive-based approach, can effectively create efficient vectorized versions of applications.

We also propose extensions to the OpenACC directives for creating efficient vectorizations. The extended directives provide data structure abstraction, which is very important for efficient vectorization. We then develop a source-to-source translator that supports the proposed directives and evaluate it using two real-world applications, namely UPACS and CCS-QCD. The results show that the performance improved by 23% and 20% compared with the baseline for UPACS and CCS-QCD, respectively.

Moving on to domain-specific approaches, we enhance the HACApK open-source H-matrix framework, originally developed for symmetric multiprocessing clusters, to work well on many-core processors. In particular, we propose many-core-focused load-balancing-aware parallel adaptive cross approximation (ACA) algorithms for working with H-matrices. Existing parallel algorithms apply the ACA process to each of the H-matrix's sub-matrices independently. However, since the computational load of each ACA process depends on the sub-matrix's size, which is undefined before the ACA process is applied, it is difficult to balance the load. Because our algorithms are implemented as part of the HACApK framework, framework users can reap the benefits of these changes without even realizing it. We then evaluate the proposed algorithms using boundary element method (BEM) problems on both an NVIDIA Tesla P100 GPU (P100) and an Intel Xeon Broadwell (BDW) CPU. The results show that our algorithms improved performance in all GPU cases.

In addition, we propose a design of framework for abstracting the vectorization process. We then adopt this framework design to the BEM-BB framework, a HACApK wrapper framework for BEM analyses. We then evaluate the adapted framework using two BEM problems, namely, static electric field analysis with a perfect conductor and with a dielectric, on BDW and Intel Xeon Phi Knights Landing (KNL) processors. The results show that this approach can offer good vectorization performance while requiring little vectorization knowledge. Specifically, in perfect conductor analyses conducted using H-matrices, the new framework improved performance by 2.22 and 4.34 times compared with the original BEM-BB framework on the BDW and KNL processors, respectively.

In summary, we hide the difficulties involved in developing efficient programs for many-core processors by proposing abstractions based on both compiler-directive-based and domain-specific approaches. These abstractions also enable programs to be ported to different processors with different numbers of cores and vector lengths without sacrificing performance. Even though the optimal data layout, load-balancing algorithm and vector length may differ among processors, our proposals enable the most suitable approaches to be easily selected. We therefore believe that they are one possible answer to the multi- to many-core paradigm shift.

備考：論文要旨は、和文 2000 字と英文 300 語を 1 部ずつ提出するか、もしくは英文 800 語を 1 部提出してください。

Note: Thesis Summary should be submitted in either a copy of 2000 Japanese Characters and 300 Words (English) or 1 copy of 800 Words (English).

注意：論文要旨は、東工大リサーチリポジトリ (T2R2) にてインターネット公表されますので、公表可能な範囲の内容で作成してください。

Attention: Thesis Summary will be published on Tokyo Tech Research Repository Website (T2R2).