論文 / 著書情報
Article / Book Information

題目(和文)	メニーコアアーキテクチャにおける疎行列計算の性能最適化
Title(English)	Performance Optimization of Sparse Matrix Kernels for Many-core Architectures
著者(和文)	長坂 侑亮
Author(English)	Yusuke Nagasaka
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11066号, 授与年月日:2019年3月26日, 学位の種別:課程博士, 審査員:松岡 聡,遠藤 敏夫,増原 英彦,額田 彰,横田 理央
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第11066号, Conferred date:2019/3/26, 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

系・コース： Department of, Graduate major in	数理・計算科学　系 数理・計算科学　コース	申請学位（専攻分野）： Academic Degree Requested	博士（　理学　） Doctor of
学生氏名： Student's Name	長坂　侑亮	指導教員（主）： Academic Supervisor(main)	松岡　聡
		指導教員（副）： Academic Supervisor(sub)	

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

Many-core architectures such as Graphic Processing Units (GPUs) and Many Integrated Core architectures (MICs) are becoming mainstream in high performance computing platform, accelerating the performance of various kinds of applications such as simulations, Big Data analytics and Machine Learning. Currently, areas for improving application performance on many-core processors include operations on sparse matrices, which are mainly occupied by zero elements. Sparse matrices are used in various fields, such as the problem matrix in simulations or the adjacency matrix in graph processing, and it is crucial to accelerate the kernels for processing sparse matrices in a wide range of applications. However, the kernels for sparse matrix operations on many-core processors still have several performance issues.

There are two main issues with optimizing sparse matrix kernels on many-core processors. The first issue deals with memory access. A sparse matrix is usually compressed using sparse matrix format, holding only non-zero elements with value and indices information. This increases byte per flop ratio while the byte per flop ratio of many-core processors is relatively low compared to modern multi-core CPUs, resulting in much waste of computing capability of many-core processors. Furthermore, a kernel based on a sparse matrix format yields indirect memory access resulting in frequent cache misses, especially on many-core processors, whose cache capacity is small. The second issue is concerned with load balancing. The computation complexity is determined by the pattern or number of non-zero elements in the sparse matrix. Non-sophisticated task assignment causes load imbalance wasting the massive parallelism of many-core architectures. In addition, another load-balancing issue emerges when processing many kernels for sparse matrices in parallel.

Firstly, for sparse matrix vector multiplication (SpMV), we propose the Non-Uniformly Segmented (NUS) format and the Adaptive Multi-level Blocking (AMB) format to tackle the memory access and load balancing issues of sparse matrix operations on many-core architectures. By dividing the matrix along the column, the memory access to input vector elements attains better cache locality. In the AMB format, furthermore, the number of bits for holding the indices of non-zero elements is reduced, and the contiguous non-zero elements in the matrix is treated as “one block” in order to reduce the amount of bytes accessed, thereby alleviating high byte per flop ratio. Performance evaluation using a variety of sparse matrix datasets shows that our proposed sparse matrix formats and algorithms for SpMV achieve significant speedups of up to 1.4x compared to existing state-of-the-art algorithms.

Secondly, for sparse general matrix-matrix multiplication (SpGEMM), we propose a hash table-based algorithm optimized for GPU with perfect load balance. In SpGEMM, memory allocation is also one of the issues due to the large memory required for temporary use and storing output matrix despite the limited memory capacity of many-core processors. Our approach efficiently uses shared memory on the GPU for hash table, and the evaluation on NVIDIA GPU shows significant speedups of up to 4.4x compared to existing approaches for GPU. The proposed optimization strategy for SpGEMM has also been applied to Intel Xeon Phi. We executed microbenchmarks on Intel Xeon Phi in order to expose the performance bottlenecks of SpGEMM, and optimize hash-based and heap-based approaches for Intel Xeon Phi. We built the performance model of SpGEMM and the guide for selecting the best algorithm for specific input and scenario based on empirical analysis.

Thirdly, for processing many sparse matrix multiplications (SpMMs) mainly between small matrices, we propose Batched SpMM and Batched SpMM Dynamic. Recently proposed Graph Convolutional Networks (GCN) can deal with the graph structure as input of the neural networks.

In GCN applications, the graph structure is expressed as adjacency matrix, and many SpMM kernels are executed. However, the graph sizes are often very small and the parallelism of many-core processors is hardly exploited. To improve the occupancy of many-core processors and achieve high performance, Batched approaches execute tens or hundreds of SpMM kernels with a single kernel launch. The evaluation results show that Batched approaches largely accelerate the GCN application and the speedup in training is up to 1.64x.
This thesis provides several contributions to performance optimizations across a wide-area of computer science from traditional simulations to Bigdata and Machine Learning related to the kernels for sparse matrix.