

Title	GPU向きStrassenアルゴリズムの最適化
Author(s)	大塚, 達史
Citation	
Issue Date	2020-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/16385
Rights	
Description	Supervisor: 井口 寧, 先端科学技術研究科, 修士 (情報科学)

修士論文

GPU 向き Strassen アルゴリズムの最適化

1810032 大塚 達史

主指導教員 井口 寧 教授
審査委員主査 井口 寧 教授
審査委員 金子 峰雄 教授
田中 清史 准教授
本郷 研太 准教授

北陸先端科学技術大学院大学
(情報科学)

2020 年 2 月

Abstract

GPU を用いた Strassen アルゴリズムによる行列の乗算の高速化を行った．高速化を行う上で二つの問題があることを見出した．第一の問題は部分行列の乗算の並列実行数が行列サイズが増加するにつれて減少してしまうことである．これまでの GPU による Strassen アルゴリズムの研究は計算途中の結果を保存するための Temporary 行列を使用していた．第二の問題は GPU では Temporary 行列を使用するためのメモリ確保と解放の時間を要することである．本研究では第一の問題については GPU リソースの影響を明らかにすることにより行列サイズごとの最適な並列実行数を決定した．第二の問題は Temporary 行列を削除した演算スケジューリングを作成することでメモリ確保と解放の時間を削減することができた．本研究は NVIDIA GPU Tesla V100, Tesla P100 および Tesla K20 を使用し, 部分行列の乗算に CUBLAS-10.1 を用いて性能比較を行った．その結果本研究のプログラムは Strassen アルゴリズムの GPU による計算の研究で最速の計算プログラムよりもすべての行列のサイズで高速となった．GPU V100 を用いた行数 4096×4096 の計算で先行研究のプログラムより 11 % 高速になった．

Abstract

We made acceleration of matrix multiplication by the Strassen algorithm on GPU. In achieving the acceleration we found two problems. The first problem is a decrease in the number of parallel executions of multiplications of submatrices. Earlier studies on the Strassen algorithm with GPUs employed temporary matrices for storing intermediate results. The second problem is the time spent by allocation and deallocation of memories for these temporary matrices. For overcoming the first problem we determined the optimal number of parallel executions of submatrices multiplications by clarifying the influence of GPU resources. For overcoming the second problem we reduced the time for the memory allocation and deallocation by making an operation schedule without a temporary matrix. NVIDIA GPU Tesla V100, Tesla P100 and Tesla K20 were used, and the performance evaluation was made by using CUBLAS library 10. 1. For all matrix sizes the program made in the present study is faster than programs made in earlier studies on the Strassen algorithm with GPUs. For multiplication of 4096×4096 matrices our program is faster than the fastest program in earlier studies by 11 %.