

発表概要

GPUを用いた格子ボルツマン法のループ展開を利用した同期オーバーヘッド削減による高速化

鈴木 宏明^{1,a)} 富永 浩文¹ 中村 あすか¹ 前川 仁孝¹

2015年8月5日発表

本研究では、GPUを用いた格子ボルツマン法を高速化するために、ループ展開を利用して、GPUの実行効率を向上する手法を提案する。GPUを用いた格子ボルツマン法は、時間ステップごとにカーネルを起動し、複数のスレッドブロックが時間発展方程式を求解することで流体の流れを解析する。スレッドブロックは、メモリアクセスのコストを少なくするために、計算に必要なデータをグローバルメモリからシェアードメモリに格納し、シェアードメモリにあるデータのみ参照して計算する。シェアードメモリとグローバルメモリ間のデータ転送のようなスレッド間の同期処理は、計算時間の大部分を占めるため、オーバーヘッドになる。そこで本研究では、同期オーバーヘッドを削減するために、ループ展開を用いることで複数ステップ分の計算を1つのカーネルに割り当てる。本手法は、1回の時間ステップでシェアードメモリに格納するデータは多くなるが、グローバルメモリとシェアードメモリ間のデータ転送の回数やスレッド間の同期回数を削減できるため、高速化できる。最後に、提案手法の有効性を GeForce TITAN Black で評価した。評価の結果、ループ展開した格子ボルツマン法は、ループ展開しない格子ボルツマン法に比べ、最大約 3.1 倍高速化することを確認した。

Speedup of the Lattice Boltzmann Method Using Loop Transformation for Reducing Synchronization Overhead on GPU

HIROAKI SUZUKI^{1,a)} HIROBUMI TOMINAGA¹ ASUKA NAKAMURA¹ YOSHITAKA MAEKAWA¹

Presented: August 5, 2015

This study proposes a speedup method by improving an efficiency of calculation of the Lattice Boltzmann Method on GPU using loop transformation. This method creates GPU kernels at each time step and analyzes fluid flow. GPU kernel creates thread blocks and calculates time evolution equations. Thread blocks store a data from global memory to shared memory in order to reduce a cost of memory access. Therefore, this method calculates by referring only the data in shared memory. Synchronization between threads, such as data transfer between global memory and shared memory is overhead in order to occupy the majority of calculations. So, this study assigns calculation of multiple time steps to one GPU kernel using loop transformation in order to reduce synchronization overhead. The proposed method stores more data in shared memory compared with the conventional method in a single time step. However, it is possible to reduce synchronization times of threads and data transfer between global memory and shared memory. Finally, the results of the evaluation using GeForce TITAN Black shows the proposed method gives us about 3.1 times speedup compared with the conventional method.

¹ 千葉工業大学情報工学科
Department of Computer Science, Chiba Institute of Technology

^{a)} hsuzuki@mae.cs.it-chiba.ac.jp