

論文 / 著書情報  
Article / Book Information

題目(和文)	GPU上のオブジェクト指向プログラミングのメモリ効率化
Title(English)	Memory-Efficient Object-Oriented Programming on GPUs
著者(和文)	シュプリンガー マティアス
Author(English)	Matthias Springer
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第11319号, 授与年月日:2019年9月20日, 学位の種別:課程博士, 審査員:増原 英彦,遠藤 敏夫,首藤 一幸,南出 靖彦,脇田 建
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第11319号, Conferred date:2019/9/20, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	審査の要旨
Type(English)	Exam Summary

(博士課程)

## 論文審査の要旨及び審査員

報告番号	甲第	号	学位申請者氏名	SPRINGER, Matthias		
論文審査 審査員		氏名	職名		氏名	職名
	主査	増原英彦	教授	審査員	脇田建	准教授
	審査員	遠藤敏夫	教授			
		首藤一幸	准教授			
南出靖彦		教授				

### 論文審査の要旨 (2000 字程度)

論文は「Memory-Efficient Object-Oriented Programming on GPUs」と題し、Graphics Processing Unit (GPU) 上でのオブジェクト指向プログラミングを、メモリ管理に関して性能向上させる手法を提案している。論文は英文で以下の 8 章で構成されている。

第 1 章「Introduction」は、現在の汎用 GPU 計算では、性能上の問題からオブジェクト指向プログラミングがほとんど用いられていないことを述べた上で、それを解決することを論文の主旨としている。さらにその解決にあたって、以降の章が果たす役割を述べている。

第 2 章「Background」は、論文の背景知識として、GPU の実行モデル、オブジェクト指向プログラミング言語の実行モデル、構造データのメモリ配置方式を解説している。解説はメモリ性能の面から、実際の GPU を用いた性能測定結果を交えて整理されている。

第 3 章「Expressing Parallelism in Object-oriented Programs」は、GPU でオブジェクト指向プログラミングを実現するための領域特化言語 Ikra-Ruby および Ikra-Cpp を提案している。Ikra-Ruby は Ruby 言語のデータ並列ライブラリである。Ikra-Cpp は、本論文が新たに提案する Single-Method Multiple-Objects (SMMO) 並列モデルに基づく領域特化言語で、C++ ライブラリとして提供される。ここではそれぞれの言語の設計方針を詳細な API と例題の記述を示しながら述べている。

第 4 章「Optimizing Memory Access」はメモリ最適化手法を 2 つ提案している。1 つは GPU カーネル処理を融合する最適化で、Ikra-Ruby 上に実現している。もう 1 つは Ikra-Cpp において C++ オブジェクトを Structure-of-Arrays (SoA) 方式でメモリ上に配置するコンパイル手法である。前者は融合変換として知られている技法を、Ruby のような動的な言語において実現する方法を示しており、画像処理アプリケーションを用いた性能評価によって人手で最適化されたプログラムに匹敵する速度を達成することが示されている。後者は、クラス、フィールドおよび添字情報を疑似ポインタに埋め込む方式と、C++ テンプレートメタプログラミングによる実現方式から成っており、配列を用いたプログラムに等しい性能を達成している。SoA 方式の実現は、GPU に対するものとしては初めてであり、第 5,6 章の提案の重要な土台となっている。

第 5 章「Dynamic Memory Allocation with SOA Performance Characteristics」は Ikra-Cpp のための動的メモリ割り当て手法 DynaSOAr を提案している。GPU での動的なメモリ割り当ては、多数の割り当て要求が同時発生するため、それを衝突させない工夫が欠かせない。本章は、先行研究が用いているランダムな方法では、引き続く計算の速度低下を引き起こすことを指摘している。そして解決策として、前章の SoA 方式をブロック単位に変更し、さらに階層的なビットマップを走査・探索するための並列アルゴリズムを用いる手法を提案している。これによって、高速な割り当てと、連続性の高い割り当て結果を同時に実現している。性能評価では、アプリケーションプログラムの実行速度を先行研究から 3 倍以上に高速化するという大きな改善を得ている。

第 6 章「GPU Memory Defragmentation」はメモリの断片化除去手法を提案している。動的にオブジェクトが削除されるアプリケーションでは、次第にオブジェクトの配置が断片化して実行速度が低下することが知られている。本章では、断片化したブロックを発見し、断片化したブロック間でオブジェクトを移動し、移動したオブジェクトへの参照を更新するアルゴリズムを、それぞれ GPU 上で並列に行

うものとして提案し、DynaSOAr の拡張として実現している。性能評価では、いくつかのアプリケーションプログラムにおいて、断片化除去を行うことで全体の実行時間が短縮されることを示している。このような断片化除去手法は GPU 向けとしては初めての提案であり、アルゴリズムの提案だけでなく、全体性能の向上までを示している本章の意義は大きい。

第 7 章「SMMO Examples」は、SMMO の記述力評価を、8 つのアプリケーションプログラムを用いて行っている。考察は記述力、安全性などと動的メモリ割り当てや継承機構の利用の観点から行っている。GPU 向けのオブジェクト指向アプリケーションプログラムの例はこれまでほとんどない中で、具体的なアプリケーションを通じた評価を行っており、その知見は貴重なものだと言える。

第 8 章「Conclusion」は、本論文の総括を述べている。

以上のように、本論文は、オブジェクト指向プログラミングを GPU 上で実行する際に、メモリ管理が実行速度上の問題となることを指摘し、複数の抽象レベルで新しい手法を提案することで解決している。解決策の有効性は実際のアプリケーションプログラムの性能測定によって確かめられており、学術上貢献するところ大である。よって本論文は博士(学術)の学位論文として十分価値があるものと認める。

注意：「論文審査の要旨及び審査員」は、東工大リサーチポジトリ(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。