

令和 5 年 6 月 29 日現在

機関番号：12608  
研究種目：基盤研究(B) (一般)  
研究期間：2018～2022  
課題番号：18H03219  
研究課題名(和文)生産性の高いGPU向けプログラミング言語処理系  
  
研究課題名(英文)High-Productivity GPU Programming Languages  
  
研究代表者  
増原 英彦(Masuhara, Hidehiko)  
  
東京工業大学・情報理工学院・教授  
  
研究者番号：40280937  
交付決定額(研究期間全体)：(直接経費) 14,040,000円

研究成果の概要(和文)：生産性の高いGPGPU向けプログラミング言語処理の構築に向けて以下の成果を得た。まずオブジェクト指向プログラミングをGPU上で行う際に大きな障害となっていた動的なメモリ割り当てについて新しい管理手法を提案しC++領域特化言語DynaSOArとして実現し、これまでのメモリ管理ライブラリと比べて同程度の割り当て速度を保ちつつ連続性の高い割り当てができ、アプリケーションプログラムを大きく高速化できるものを得た。この成果を発展させ、継承機構、モジュール化機構、動的コンパイルなどのより高度なオブジェクト指向言語機能に接続する研究も行った。

#### 研究成果の学術的意義や社会的意義

GPUを搭載した計算機環境がますます一般的になる中で、GPGPUアプリケーションプログラムの記述を容易にすることは研究開始当初から継続して求められており、今後もますます必要となると考えられる。そのような状況下で、これまで性能上の理由から事実上で行えなかった動的メモリ割り当ての効率化に成功したことで、GPU上でのオブジェクト指向プログラミングの応用範囲が大きく広がる可能性を示した意義は大きいと考える。

研究成果の概要(英文)：Towards highly-productive GPGPU programming languages, this project obtained the following results. It proposed a novel algorithm for dynamic memory management allocation, which is one of the primary obstacles that prevents from using object-oriented programming on GPU. The algorithm was implemented as a domain-specific language for C++, called DynaSOAr, which allocates memory as fast as existing memory allocators, yet allocates more densely, resulting in faster overall application execution. Based on this work, the project further studied on advanced object-oriented language features including inheritance, modularity, and dynamic compilation.

研究分野：プログラミング言語

キーワード：GPGPU オブジェクト指向プログラミング 動的メモリ割り当て

## 1. 研究開始当初の背景

研究代表者は本研究開始以前までにも、プログラミング言語の先進的なモジュール機構や、実行時コンパイラなどの実現方式に関する研究を中心に行ってきた。GPGPU (グラフィクス処理装置を用いた汎用計算) は、並列計算を安価に行える装置として当初より注目していたが、その single-instruction/multiple-thread (SIMT) 実行に由来する CUDA などの原始的なプログラミングモデルは、関数型言語やスクリプト言語に備えられている配列内包 (array comprehension) 記法によって記述力を大きく改善できる可能性を持っている。

本研究の主眼の1つである Ruby を基にした Ikra 言語を開発した理由は、GPGPU における「アルゴリズムのプロトタイピング」の重要性に注目したためである。GPGPU 化できる処理には数値計算だけでなく、文字列照合、整列、シミュレーション、グラフ処理などが数多くのものが考えられている。これらのアルゴリズムを作成するためには、データ表現方法や並列化の単位などについての試行錯誤が必要となるため、記述力の高い言語によってプロトタイプを検討できることは大きな効率化につながると思った。また、GPU の性能を 100% 引き出せなくとも良い場合には、Ruby のようなスクリプト言語が提供する豊富な入出力・可視化ライブラリを用いたアプリケーション構築が容易に行える点も大きな利点となると考えた。

記述力の高いプログラミング言語から GPU を利用する実用的な手段は数多く提供されている。その多くは CUDA で記述された並列配列ライブラリを呼び出すもの (Numba, MATLAB 等) であり、行える計算が限定されている。プログラマが GPU カーネルを記述できる GPGPU 領域特化言語としては、Haskell に基づく Accelerate、Python に基づく Copperhead などが研究されている。前節でも述べたように、これらの言語は全て **1 基の GPU を対象としている**。

同様の研究に対する注目の高まりを受けて、International Workshop on Libraries, Languages, and Compilers for Array Programming (ARRAY ワークショップ) が 2014 年から PLDI (Programming Languages Design and Implementation) 国際会議に開催されている。

領域特化言語によって先進的なハードウェアを活用するプログラミングを支援することは、GPGPU に限らず多くなっている。大規模データの分散メモリ処理を目的とした MapReduce と Hadoop や、グラフデータの処理に特化した Pregel、深層学習に特化した TensorFlow や Chainer などが知られている。また、研究レベルではステンスル計算に特化した Physis や粒子計算に特化した FDPS などの試みもある。これらの言語・ライブラリは本研究よりも **限定された応用に特化した設計を行っている**ものの、プログラマが記述した計算カーネルを分散メモリ型環境で並列実行する点で本研究と共通している。

言語処理系の正しさに関しては、CompCert プロジェクトのように C 言語コンパイラの正しさを検証した例が有名であるが、これまでに行われているものは **逐次プログラミング言語がほとんど**であった。GPGPU のように特殊な並列実行モデルに対しても、理論上は CompCert など同様のアプローチが取れるものと予想されるが、テンプレートを使ったコード生成のような特有の方式の扱い

方は興味深い研究課題になっている。

## 2. 研究の目的

GPGPU 領域特化言語の (i) 言語設計、(ii) 実行方式、(iii) 正しさの保証の 3 側面についての要素技術を提案することである。そしてこれらの提案を通じて、GPGPU をはじめとする演算加速器を活用するプログラミング言語の基本技術を前進させることを目指す。

## 3. 研究の方法

研究方法は上記 (i)~(iii) についてそれぞれ以下のように進めた。

### 3.1. 言語設計

これまでの GPGPU 領域特化言語は、並列計算の記述方法として数値配列だけをサポートしていた。このような言語で、オブジェクトが使えるようになり、またオブジェクトを用いたグラフ処理が並列実行可能になれば、計算対象の問題をより直接的に、また再利用性の高い形で扱えるようになる。しかしそのためには、サポートする言語機能を適切に設計すること、GPU 特有の最適化を含めた実現方法を明らかにする必要がある。

そこで本研究では、Ikra 領域特化言語に**オブジェクトのサポート**を追加する。まずは継承機構のないオブジェクトを扱う言語拡張を設計し、それをフィールド変数の配列へと変換する方法を提案・作成する。さらに複数クラスが混在する場合の最適化としてクラスによって再整列するような最適化を試みる。その後、交通シミュレーションのような応用プログラムによって記述力と速度の評価を行う。さらに継承機構を導入し、記述力の向上を行う。

**グラフ処理**に関してはグラフの節点をオブジェクトとして表現する言語設計を行い、実行モデルとしては Pregel と同様に、節点と枝の関係を添字番号とした配列に対する map・reduce 処理を行うプログラムを生成する。大規模グラフ探索等の代表的なアプリケーションに対しての記述力・性能評価を行った後、言語機構の再設計を行う。特にオブジェクトの参照関係を素直に表現できる言語形式を探求する。さらに、偏りの大きいグラフに対する負荷均等化や通信・計算の重畳などの最適化を導入する。

スパコンや携帯端末などにおいて電力消費量をアプリケーションの振舞を変更することで制御するための**資源意識のためのモジュール機構**を提案する。具体的には、プログラマがアプリケーションのいくつかのモジュールについて、異なる電力・品質特性を持つものを用意すると、それを実行中の品質・電力要求に応じて最適な組み合わせに入れ替える機構となる。まず、既存の適応的アルゴリズムを用いて実行時のモジュール入れ替えることで電力最適化を行う実験系を作成し、その効果を測定する。その後、モジュールの記述方法として文脈指向プログラミング言語 JCop の拡張を行う。続けて同じ言語機構を Ikra 領域特化言語にも導入し、アプリケーションプログラムの記述実験によって記述力の評価を行う。

### 3.2. 実行方式

これまでの GPGPU 領域特化言語は、1 基の GPU を利用するものに限られていた。また埋め込み型領域特化言語を用いて GPU が行う計算の中核 (GPU カーネル) を記述させているため、実行時の変換オーバーヘッドが無視できなくなる問題も存在している。スパコンのような大規模分散 GPU システムにおいても手軽にプログラミングを行いたい需要は存在するので、分散 GPU のサポートは 1 つの解となり得る。また、汎用プログラミング言語で一般的となった実行時コンパイル技法を GPU カーネルの抽出に対応させることは、オーバーヘッドの低減だけでなく、高度なカーネル融合などの最適化への道を拓くことにもなる。

そこで本研究では**分散 GPU 環境で効率的に実行するための実行方式**を提案する。本研究の範囲では Ikra 言語を東京工業大学の TSUBAME スパコンで実行可能とすることまでを目標とする。まず GPU 計算を分散 GPU に分配・集約するバックエンドを MPI ライブラリを用いて作成し、分散計算環境で既知の最適化である通信と計算の重畳化や時間ブロック化を導入する。その後、ハードウェアアーキテクチャに対する汎用性を高めるため、GPU コードテンプレートの記述に、Steuer らの中間言語を参考にしつつ、抽象計算モデルを定義して再記述をする。その後最適化手法を整備した上で大規模アプリケーションに対する記述力と性能の評価を行う。

GPU カーネルの抽出とコンパイルのために**履歴型実行時コンパイラ**に基づく手法を提案する。準備段階として履歴型実行時コンパイラの成功例である PyPy 言語の処理系を単純化した実験システムを MinCaml コンパイラを改変して作成する。配列に対する map 処理などを起点とした実行履歴を用いて GPU カーネル部分を抽出する手法を実験システムに組み込み、LLVM を用いて CUDA PTX(中間言語) を生成するシステムを作成する。研究が順調に進んだ場合、PyPy のバックエンドである RPython 上での処理系を構築し、現在の Ikra 処理系のバックエンドを置き換える。

### 3.3. 正しさの保証

言語処理系の正しについては CompCert プロジェクトのような検証済コンパイラの成功例があるが、並列言語について行われた例はほとんどない。GPGPU 領域特化言語処理系の正しさを検証することは、並列化アルゴリズムに混入しやすい誤りを避ける上でも重要な方向性である。

そこで本研究では GPGPU 領域特化言語のための**コンパイラ検証**の基本手法を提案する。具体的にはこれまでに研究した Coq 上に構成した GPGPU 並行分離論理を用いて Accelerate 言語のサブセットに対するコンパイラを検証付きで作成する。当初はホスト言語を単純化した上で枠組を作成し、メモリ管理部分を拡張する。さらに分散 GPU 環境に対応するために、コードテンプレートの記述を Seuer らの中間言語に基づくものに拡張し、中間言語レベルの最適化に対応させる。

コンパイラが使用する CUDA コードテンプレートのために、**プログラム合成による CUDA 最適化**手法を提案する。素朴に記述された CUDA プログラムを人手によってプログラムを書き換えて最適化する際に、複雑になる添字式の記述や同期命令の挿入をプログラム合成技法によって行わせるアイデアである。そのために Rosette プログラム合成系上に CUDA 言語の意味論を構成し、ステンシル計算に対する最適化をプログラム合成によって支援する方法を提案する。実際に Ikra 処理系が用

いているコードテンプレートにこの手法で最適化したものを導入し、記述性と性能を確認する。さらに一般の CUDA プログラムに対しても適用できるように手法の一般化を行う。

#### 4. 研究成果

上記の方法に沿って研究を進め、以下の成果を得た。

まずオブジェクト指向プログラミングを GPU 上で行う際に大きな障害となっていた動的なメモリ割り当てについて新しい管理手法を提案し C++ 領域特化言語 DynaSOAr として実現した。GPU 上のオブジェクト指向プログラミングでは多数のスレッドがオブジェクトのメソッドを呼び出し、それらが新しいオブジェクトの生成や削除を行う。実行効率のためには、各スレッドは他のスレッドとの同期せずに割り当てを行いつつも、結果として同種のオブジェクトを連続したメモリ領域に配置する必要がある。この相反する要請に対してブロック管理、ポインタ埋め込み、階層ビットマップという方法を提案し、これまでのメモリ管理ライブラリと比べて同程度の割り当て速度を保ちつつ連続性の高い割り当てができ、アプリケーションプログラムを大きく高速化できるものを得た。

またこの成果を発展させ、継承機構、モジュール化機構、動的コンパイルなどのより高度なオブジェクト指向言語機能に接続する研究も行った。

GPU を搭載した計算機環境がますます一般的になる中で、GPGPU アプリケーションプログラムの記述を容易にすることは研究開始当初から継続して求められており、今後もますます必要となると考えられる。そのような状況下で、これまで性能上の理由から事実上で行えなかった動的メモリ割り当ての効率化に成功したことで、GPU 上でのオブジェクト指向プログラミングの応用範囲が大きく広がる可能性を示した意義は大きいと考える。

## 5. 主な発表論文等

〔雑誌論文〕 計15件（うち査読付論文 15件 / うち国際共著 3件 / うちオープンアクセス 0件）

1. 著者名 Izawa Yusuke, Masuhara Hidehiko	4. 巻 -
2. 論文標題 Amalgamating different JIT compilations in a meta-tracing JIT compiler framework	5. 発行年 2020年
3. 雑誌名 -	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3426422.3426977	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 Leger Paul, Masuhara Hidehiko, Figueroa Ismael	4. 巻 -
2. 論文標題 Interfaces for Modular Reasoning in Context-Oriented Programming	5. 発行年 2020年
3. 雑誌名 Proceedings of the 12th International Workshop on Context-Oriented Programming and Advanced Modularity (COP 2020),	6. 最初と最後の頁 1-7
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3422584.3423152	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 新美和生, 増原英彦.	4. 巻 13(3)
2. 論文標題 Lift中間言語における動的長配列の追加.	5. 発行年 2020年
3. 雑誌名 情報処理学会論文誌:プログラミング	6. 最初と最後の頁 1-12
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -
1. 著者名 Izawa Yusuke, Masuhara Hidehiko, Aotani Tomoyuki	4. 巻 -
2. 論文標題 Extending a meta-tracing compiler to mix method and tracing compilation	5. 発行年 2019年
3. 雑誌名 Proceedings of the Workshop on Modern Language Runtimes, Ecosystems, and VMs (MoreVMs'19) in Proceedings of the Conference Companion of the 3rd International Conference on Programming,	6. 最初と最後の頁 5:1-5:3
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3328433.3328439	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Adkisson Jeanine Miller、Westlund Johannes、Masuhara Hidehiko	4. 巻 .
2. 論文標題 A shell-like model for general purpose programming	5. 発行年 2019年
3. 雑誌名 Proceedings of the Workshop on Modern Language Runtimes, Ecosystems, and VMs (MoreVMs'19) in Proceedings of the Conference Companion of the 3rd International Conference on Programming,	6. 最初と最後の頁 10:1-10:7
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3328433.3328444	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Springer Matthias、Masuhara Hidehiko	4. 巻 -
2. 論文標題 Massively parallel GPU memory compaction	5. 発行年 2019年
3. 雑誌名 -	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3315573.3329979	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Matthias Springer and Hidehiko Masuhara.	4. 巻 134
2. 論文標題 DynaSOAr: A parallel memory allocator for object-oriented programming on GPUs with efficient memory access.	5. 発行年 2019年
3. 雑誌名 Leibniz International Proceedings in Informatics (LIPICS),	6. 最初と最後の頁 17:1-17:37
掲載論文のDOI (デジタルオブジェクト識別子) 10.4230/LIPIcs.ECOOP.2019.17	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Matthias Springer, Yaozhu Sun, Hidehiko Masuhara	4. 巻 -
2. 論文標題 Inner array inlining for structure of arrays layout	5. 発行年 2018年
3. 雑誌名 Proceedings of the 5th ACM SIGPLAN International Workshop on Libraries, Languages, and Compilers for Array Programming (ARRAY@PLDI 2018)	6. 最初と最後の頁 50-58
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3219753.3219760	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1. 著者名 Yudai Tanabe, Tomoyuki Aotani, Hidehiko Masuhara	4. 巻 -
2. 論文標題 A context-oriented programming approach to dependency hell	5. 発行年 2018年
3. 雑誌名 Proceedings of the 10th International Workshop on Context-Oriented Programming: Advanced Modularity for Run-time Composition, COP '18	6. 最初と最後の頁 14-8
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3242921.3242923	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Akio Oka, Hidehiko Masuhara, Tomoyuki Aotani	4. 巻 -
2. 論文標題 Live, synchronized, and mental map preserving visualization for data structure programming	5. 発行年 2018年
3. 雑誌名 Proceedings of the 2018 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Onward! 2018	6. 最初と最後の頁 72-87
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3276954.3276962	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Anthony Canino, Yu David Liu, Hidehiko Masuhara	4. 巻 -
2. 論文標題 Stochastic energy optimization for mobile GPS applications	5. 発行年 2018年
3. 雑誌名 Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018	6. 最初と最後の頁 703-713
掲載論文のDOI (デジタルオブジェクト識別子) 10.1145/3236024.3236076	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1. 著者名 Yusuke Izawa, Hidehiko Masuhara, Tomoyuki Aotani	4. 巻 -
2. 論文標題 Extending a meta-tracing compiler to mix method and tracing compilation	5. 発行年 2019年
3. 雑誌名 Proceedings of the Workshop on Modern Language Runtimes, Ecosystems, and VMs (MoreVMs'19)	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -



1. 著者名 Jeanine Miller Adkisson, Johannes Westlund, Hidehiko Masuhara	4. 巻 -
2. 論文標題 A shell-like model for general purpose programming	5. 発行年 2019年
3. 雑誌名 Proceedings of the Workshop on Modern Language Runtimes, Ecosystems, and VMs: Track on Programming Across the System Stack (MoreVMs/PASS'19)	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

1. 著者名 Matthias Springer and Hidehiko Masuhara	4. 巻 -
2. 論文標題 Massively parallel GPU memory compaction	5. 発行年 2019年
3. 雑誌名 Proceedings of the ACM SIGPLAN International Symposium on Memory Management (ISMM' 19)	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Matthias Springer and Hidehiko Masuhara	4. 巻 -
2. 論文標題 DynaSOAr: A parallel memory allocator for object-oriented programming on GPUs with efficient memory access	5. 発行年 2019年
3. 雑誌名 Proceedings of of European Conference on Object-Oriented Programming (ECOOP'19)	6. 最初と最後の頁 -
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

〔学会発表〕 計20件 (うち招待講演 2件 / うち国際学会 3件)

1. 発表者名 Yusuke Izawa, Hidehiko Masuhara, and Youyou Cong.
2. 発表標題 An interpreter design for supporting different JIT compilations in RPython framework.
3. 学会等名 The 23rd JSSST Workshop on Programming and Programming Languages,
4. 発表年 2021年

1. 発表者名 Hidehiko Masuhara.
2. 発表標題 Object support for GPU programming: Why and how. (Keynote Talk)
3. 学会等名 the 18th Asian Symposium on Programming Languages and Systems (APLAS 2020) (招待講演) (国際学会)
4. 発表年 2020年

1. 発表者名 Tetsuo Kamina, Tomoyuki Aotani, and Hidehiko Masuhara.
2. 発表標題 Managing persistent signals using signal classes.
3. 学会等名 the 7th Workshop on Reactive and Event-based Languages and Systems (REBLS 2020). (国際学会)
4. 発表年 2020年

1. 発表者名 Naoya Furudono, Youyou Cong, and Hidehiko Masuhara.
2. 発表標題 Koka言語に対するエフェクト割り当て最適化.
3. 学会等名 第23回プログラミングおよびプログラミング言語ワークショップ(PPL2021),
4. 発表年 2021年

1. 発表者名 Kazuki Ikemori, Youyou Cong and Hidehiko Masuhara.
2. 発表標題 代数的エフェクトを特徴に持つ計算体系へのエフェクト強制の導入と健全性の証明.
3. 学会等名 第23回プログラミングおよびプログラミング言語ワークショップ(PPL2021),
4. 発表年 2021年

1. 発表者名 Kazuki Niimi, Youyou Cong, and Hidehiko Masuhara.
2. 発表標題 Effekt言語の双方向エフェクトによる拡張.
3. 学会等名 第23回プログラミングおよびプログラミング言語ワークショップ(PPL2021),
4. 発表年 2021年

1. 発表者名 Yusuke Izawa and Hidehiko Masuhara.
2. 発表標題 Making different JIT compilations dancing to the same tune, acting in the meta-level.
3. 学会等名 The 22nd JSSST Workshop on Programming and Programming Languages,
4. 発表年 2020年

1. 発表者名 Lubis Luthfan Anshar, Yudai Tanabe, Tomoyuki Aotani, and Hidehiko Masuhara.
2. 発表標題 Object-oriented programming with versions.
3. 学会等名 The 22nd JSSST Workshop on Programming and Programming Languages,
4. 発表年 2020年

1. 発表者名 Jizhe Chenxin, Hidehiko Masuhara, and Matthias Springer.
2. 発表標題 Sanajeh: a DSL for GPGPU programming with Python objects.
3. 学会等名 The 22nd JSSST Workshop on Programming and Programming Languages,
4. 発表年 2020年

1. 発表者名 新美和生, 増原英彦.
2. 発表標題 Lift中間言語における動的長配列の追加.
3. 学会等名 情報処理学会第127回プログラミング研究会2019-4-(8),
4. 発表年 2020年

1. 発表者名 田辺裕大, Lubis Luthfan Anshar, 青谷知幸, 増原英彦.
2. 発表標題 コエフェクト多相性に基づいた版多相性の実現.
3. 学会等名 第22回プログラミングおよびプログラミング言語ワークショップ(PPL2020),
4. 発表年 2020年

1. 発表者名 Akio Oka, Hidehiko Masuhara, Tomoyuki Aotani
2. 発表標題 On defining recursive functions in live data structure programming
3. 学会等名 日本ソフトウェア科学会プログラミング論研究会
4. 発表年 2019年

1. 発表者名 Hidehiko Masuhara, Akio Oka, Tomoki Ogushi
2. 発表標題 Programming experiences with a live programming environment for data structures
3. 学会等名 International Conference on the Art, Science, and Engineering of Programming (<Programming>'19) (国際学会)
4. 発表年 2019年

1. 発表者名 Robert Hirschfeld, Tobias Durschmid, Patrick Rein, Marcel Taeumel, Hidehiko Masuhara
2. 発表標題 Narratives for multi-party mechanisms and concerns
3. 学会等名 日本ソフトウェア科学会大会
4. 発表年 2018年

1. 発表者名 Jeanine Miller Adkisson, Johannes Westlund, Hidehiko Masuhara
2. 発表標題 A shell-like model for general purpose programming
3. 学会等名 情報処理学会第122回プログラミング研究会
4. 発表年 2019年

1. 発表者名 蟹暁, 朝倉泉, 増原英彦, 青谷知幸
2. 発表標題 GPGPUプログラム最適化のためのプログラム合成器とその適用手法
3. 学会等名 The 3rd cross-disciplinary Workshop on Computing Systems, Infrastructures, and Programming (xSIG 2018)
4. 発表年 2018年

1. 発表者名 Raffi Khatchadourian, 増原英彦
2. 発表標題 Automated refactoring of legacy Java software to default methods
3. 学会等名 第17回情報科学技術フォーラム(招待講演)
4. 発表年 2018年

1. 発表者名 蟹嶋, 増原英彦, 青谷知幸
2. 発表標題 プロファイル型プログラム合成によるGPGPUプログラムの半自動最適化
3. 学会等名 日本ソフトウェア科学会プログラミング論研究会
4. 発表年 2019年

1. 発表者名 田辺裕大, 青谷知幸, 増原英彦
2. 発表標題 バージョン付きモジュールシステムの提案
3. 学会等名 日本ソフトウェア科学会プログラミング論研究会
4. 発表年 2019年

1. 発表者名 田辺裕大, Lubis Luthfan Anshar, 青谷知幸, 増原英彦
2. 発表標題 コエフェクト計算に基づく量化型の再形式化
3. 学会等名 情報処理学会第124回プログラミング研究会
4. 発表年 2019年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

DynaSOAr: A CUDA Framework for SMMO Applications <a href="https://github.com/prg-titech/dynasoar">https://github.com/prg-titech/dynasoar</a> KaniCUDA <a href="https://github.com/prg-titech/Kani-CUDA">https://github.com/prg-titech/Kani-CUDA</a> BacCaml: The Meta-Hybrid Just-In-Time Compiler <a href="https://github.com/prg-titech/baccaml">https://github.com/prg-titech/baccaml</a>
---

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
--	---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------