様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　元 年　 ６ 月 １０ 日現在

機関番号：１１３０１
研究種目：基盤研究(B)（一般）
研究期間：2016～2018
課題番号：１６Ｈ０２８２２

研究課題名（和文）機械学習技術の活用による職人的プログラミングの知能化

研究課題名（英文）Supporting performance-aware programming with machine learning techniques

研究代表者
滝沢　寛之（Hiroyuki, Takizawa）

東北大学・サイバーサイエンスセンター・教授

研究者番号：７０３２３９９６

交付決定額（研究期間全体）：（直接経費）　　12,800,000円

研究成果の概要（和文）：本研究では，高性能計算(HPC)プログラミングの支援に機械学習を効果的に利用できる事例を示した．すでに機械学習の利用が成功している問題に変換することにより，コード最適化における種々の問題も機械学習で解決できる可能性がある．また，HPCプログラミング分野で膨大な数の訓練データを用意できる問題は稀であり，効率的な収集のためには対象問題を十分に分析する重要性が示された．さらに，HPCプログラミングと同様に，機械学習の利用においても熟練者の経験と勘に頼らなければならないが，すでに数値化されているハイパーパラメータの調整であるため，計算コストの問題に置き換えて考えることが可能であることも明らかになった．

研究成果の学術的意義や社会的意義
従前，熟練のプログラマによる知識と経験に基づいて，高性能計算アプリケーションコードが対象計算システム向けに最適化されてきた．しかし，ポストムーア時代の大規模かつ複雑な計算システム向けにコードを最適化する労力は，今後ますます増大することが予想され，そのための人材を確保し続けることは困難である．この問題に対して，本研究では近年注目されている機械学習技術を有効活用することで，熟練のプログラマに求められる性能最適化の労力を大幅に軽減できる可能性を明確に示すことができた．多様な科学技術分野で必要不可欠なツールとなっている高性能計算アプリケーションの開発の効率化は，学術的にも社会的にも意義深い成果である．

研究成果の概要（英文）：This work has demonstrated some case studies of effectively using machine learning techniques for supporting High-Performance Computing (HPC) programming. Various problems in code optimization can be solved by converting the problems to the problems that have already been proven to be solved by machine learning. Moreover, this work clarified the importance of analyzing the target problems in advance of machine learning, because it is unlikely that a sufficient number of training data are available in code optimization problems. Moreover, as well as HPC programming, machine learning also needs knowledge and experiences of human experts. However, in machine learning, the problem is already parameterized, and hence can be solved if sufficiently-high performance is available.

研究分野：高性能計算

キーワード：並列処理　性能最適化　機械学習　自動性能チューニング

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９、ＣＫ－１９（共通）

１．研究開始当初の背景

近年，計算システムの大規模化と複雑化が進んでいる．そのようなシステムの正確な性能モデルを構築することは難しく，プログラムのどこの部分にどのような最適化を行えば性能が向上するのかをコンパイル時に静的に予測するのは困難である．このことから，コンパイラ最適化による性能向上には原理的な限界があり，現状では，コンパイラが効率のよいコードを生成するようにプログラマ自らが配慮してプログラムを記述・修正することでハードウェアの性能に見合った実行性能を実現している．このように，性能が向上するようにプログラムを修正する作業を性能最適化と呼ぶ．性能最適化のためには，対象システムのハードウェア構成のみならず，コンパイラの特徴や利用可能なライブラリといったシステムソフトウェアに関しても深い理解が必要である．このため，現在，性能最適化作業は職人技とも思える熟練のプログラマたちの経験と勘に基づいて行われている．その方法論の体系化も試みられているが未だ確立されておらず，個々の事例において性能向上に有効な最適化技法を試行錯誤によって探るという非生産的な作業となっている．また，特定のシステムを想定してプログラムを改変することになるため，そのプログラムも特定のシステムに特化したものになりがちである．したがって，対象システムが変わるたびに非生産的な作業を再び行う必要がある．これらの背景に加えて，近い将来に高性能計算のシステム構成は大きく変わらざるを得ないと予想されていることから，性能最適化作業の生産性を高める技術の確立が強く求められている．

２．研究の目的

本研究の目的は，現在多大な労力と時間をかけて手動で行われている職人的プログラミングを，機械学習を用いてモデル化することで，「正しく動くプログラム」を「正しく速く動くプログラム」へと変換する性能最適化作業を知能化し，高性能計算向けソフトウェア開発の生産性を飛躍的に高めることである．熟練のプログラマがソースコードを読むだけで性能最適化の大まかな方針を決め，施すべき高速化技法の候補を絞っていることを考えれば，コード中の何らかの情報と高速化技法との間には規則性があることは明らかである．その規則性を機械学習を使ってモデル化することで，性能への影響が大きい要因を検出・判断し，可変のパラメータとして調整可能とする作業および実際に調整する作業の自動化・知能化を目指す．

３．研究の方法

本研究では，適切なデータ表現とモデルの検討により，機械学習技術の性能最適化への効果的な応用方法を探る．いわゆる教師付き学習で機械学習を行う場合には，入力に対する正しい出力を定義する必要がある．ソフトウェアの性能最適化の世界で使われてきた職人技を機械学習モデルに学ばせる場合には，職人技を体系的に整理する必要がある．本研究計画では，1 年目に機械学習すべき職人技を整理し，学習データを作成するとともに適切な機械学習のモデルを検討する．また，2 年目には自動チューニング(Automatic Performance Tuning，AT)技術等との比較を行いながら，従来では実現できなかった性能最適化を機械学習では実現できることを示すとともに，さらに多くの最適化手法の学習を行う．3 年目には成果を統合し，機械学習技術による実アプリケーションの性能最適化を行い，その有効性を実証する．

４．研究成果

従前，熟練のプログラマによる知識と経験に基づいて，高性能計算(High-Performance Computing, HPC)アプリケーションコードが対象計算システム向けに最適化されてきた．しかし，ポストムーア時代の大規模かつ複雑な計算システム向けにアプリケーションコードを最適化する労力は，今後ますます増大することが予想され，そのための人材を確保し続けることは困難である．この問題に対して，我々は近年注目されている機械学習技術を有効活用することで，熟練のプログラマに求められる性能最適化の労力の軽減を目指してきた．

近年盛んに研究されている機械学習技術の中でも，Convolutional Neural Network (CNN)と呼ばれる機械学習モデルが特に画像認識分野で多くの成功を収めている．CNN の特長として，大量の画像データをそのまま深層学習モデルに提示することで，画像分類のために有用な特徴量を画像データから学習によって習得し，様々な画像分類問題において高い精度を達成できることが挙げられる．画像分類そのものの規則性だけではなく，その分類に必要な特徴量まで画像データから習得する能力は特徴学習(feature learning)あるいは表現学習(representation learning)と呼ばれ，CNN による画像認識の実用性を飛躍的に高める大きな要因となってきた．この特徴学習能力に着目し，本研究では SpMV 実装選択問題に CNN を適用することで高い精度で最適な SpMV 実装を選択できることを明らかにした．具体的には，SpMV 実装問題を画像分類問題へと変換することにより，CNN の特徴学習能力を利用可能であることが示されている．性能評価の結果，常に理想的な SpMV 実装選択を行った場合の 93%以上の性能を達成することが可能であり，機械学習技術が HPC プログラミングの労力軽減のために成功裏に利用できることを明確に示すことができた．

また，数値計算ライブラリにおける反復解法アルゴリズムにおいては，疎行列構造に応じた前処理方式の選択が性能に決定的な影響を及ぼす．しかし，その選択は高度な専門性が必要である．本研究では，上述の SpMV 実装選択のアプローチを反復解法アルゴリズムの前処理選択方式向けに改良し，多種の疎行列事例に対して性能評価を行った．その結果，実用に耐えうる精

度で最適な前処理方式が選択できることを明らかにした．その成果は当該分野の国際ワークショップにて最優秀論文賞を受賞するなど，有用性が高く評価された．

コンパイラは，内部的に様々なコード最適化を行っている．多くの場合，コンパイラの最適化レベルを指示するコンパイラオプションフラグを指定することで，コードに対して適切な最適化技法を自動的に適用する．しかし，より詳細な最適化技法を指示することで，さらに高い性能を実現できることもある．ただし，適切なコンパイラオプションを選択することは，適切なコンパイラ最適化処理を指示することと同じであり，熟練のプログラマであっても試行錯誤なしに最適なコンパイラオプションを言い当てることは難しい．コンパイラオプション選択問題の関連研究として，プログラムの性能プロファイル情報から適切なコンパイラオプションを機械学習モデルにより予測する研究が挙げられる．しかし，実用上の問題として，多様な性能プロファイル情報を得るためには膨大な数のソースコードを収集する必要がある．そこで，本研究では，性能プロファイル情報に加えて，ソースコード自体のコード構造に関する情報も利用することで，少ない訓練データでも機械学習モデルによる予測精度を高める研究を行った．

コードの構造情報を機械学習モデルの入力データとして利用するためには，コード構造情報を何らかの方法で特徴量ベクトルに変換する必要がある．そこで本研究では Tree-Based Convolutional Neural Network (TBCNN)と呼ばれる CNN の一種を利用し，コードの抽象構文木(Abstract Syntax Tree)から特徴量ベクトルへの変換を学習により獲得することを検討した．評価結果から，TBCNN により出力される特徴量ベクトルを用いてコンパイラオプション選択を行うことで，性能プロファイルを使う場合とほぼ同等の予測精度を達成できることが示されており，TBCNN の特徴学習能力によってコード構造情報の特徴量を学習できることが明らかになった．しかし，類似する構造をもつ訓練データ数が少ないコードに関しては，コード構造情報を利用することで予測精度がかえって低下する場合があることも明らかになった．

以上のように，機械学習を用いることで熟練のプログラマの経験と勘に頼っていた部分を自動化できる可能性がある一方で，機械学習モデルの設計に熟練者の経験と勘が必要となる懸念も明らかになった．例えば，CNN は層状のネットワーク構造となっており．層の数や各層のユニット数など，様々なパラメータを事前に決める必要がある．そのように学習以前に定義しなければならないパラメータはハイパーパラメータと呼ばれ，その適切な決定方法は未確立であることから，多くの場合には熟練者が試行錯誤で決定している．すなわち，HPC プログラミングと同様に，機械学習の利用においても熟練者の経験と勘に頼らなければならない．しかし，すでに数値化されているハイパーパラメータの調整であるため，人的労力ではなく計算コストの問題に置き換えて考えることが可能である．そこで本研究では，ベイズ推定に基づいてハイパーパラメータの調整を実行時間の観点から効率化する研究も行った．ハイパーパラメータ調整の１回の試行に要する時間は，そのハイパーパラメータの値によって大きく変化する．このため従前のように試行回数を最小化するではなく，実行時間を最小化することを考慮してハイパーパラメータを調整することにより，より短い時間で適切なハイパーパラメータの値を決定できることが示された．その成果をまとめた論文は機械学習の自動化をテーマとする国際ワークショップで最優秀論文賞を受賞し，同分野の専門家に学術的に高く評価された．

プログラムの高度化や並列化作業では，そのプログラム構造や高い計算コスト部分を知ることが必要である．本研究では，プログラム編集機能を持つ解析支援ツール STView を統合プログラム開発環境 Eclipse 上に開発，公開した．STView は，Fortran プログラムから条件分岐，繰り返し，手続き呼出し等の主要な構成要素を取得し，Eclipse の一つのウィンドウとしてそれらの構造をツリー形式で表示するものである．また，文字列の検索機能，構造ツリーとソースプログラム行との対応付けなど豊富な機能が実装されている．複数の Fortran プログラムに対する適用実験により，その有用性も確認した．STView を地盤-建築物の地震動解析コードや乱流シミュレーションコードに適用し，コード全体のツリー構造が容易に把握できること，および初めてコードを扱う者にとって有益な情報が与えられることが明らかになった．この結果を用いて，コードの並列化の指針が立てやすくなった．また，性能評価すべきコード部分，特にファイル入出力部分の抽出が容易に行え，計算システムの性能評価に用いることができた．

他者が開発したアプリケーションを読み解き，性能を評価し，問題点を発見し，問題点を解決することは難しい技術である．本研究では，このような性能最適化技術を体系的にまとめることにも取り組んだ．アプリケーションの性能最適化技術を高並列性能最適化技術と CPU 性能最適化技術の 2 つに分け，前者については，高並列化に関する問題点を 6 パターンに，後者については，アプリケーションを 6 パターンに分類した．メモリや L2 キャッシュのビジー時間に注目し，アプリケーションタイプごとのビジー時間の傾向を整理した．これまでの体系化の結果を英語の本としてまとめ，現在出版作業中であり 2019 年に出版されることが決まっている．

５．主な発表論文等

〔雑誌論文〕（計１６件）

1. K. Yamaguchi, T. Soga, Y. Shimomura, T. Reimann, K. Komatsu, R. Egawa, A. Musa, H. Takizawa, H. Kobayashi, “Performance Evaluation of Different Implementation Schemes of an Iterative Flow Solver on Modern Vector Machines,” Supercomputing Frontiers and Innovations, 2019 (accepted for publication) 査読有
2. M. Agung, M. A. Amrizal, R. Egawa and H. Takizawa, “The Impacts of Locality and

Memory Congestion-aware Thread Mapping on Energy Consumption of Modern NUMA Systems," 2019 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS), pp.1–3, 2019. 査読有

3. Z. Wang, M. Agung, R. Egawa, R. Suda and H. Takizawa, "Automatic Hyperparameter Tuning of Machine Learning Models under Time Constraints," IEEE Big Data 2018 Workshop, The Second International Workshop on Automation in Machine Learning and Big Data (AutoML 2018), pp. 4967-4973, 2018. 査読有
4. K. Komatsu, S. Momose, Y. Isobe, O. Watanabe, A. Musa, M. Yokokawa, T. Aoyama, M. Sato, and H. Kobayashi, "Performance Evaluation of a Vector Supercomputer SX-Aurora TSUBASA," International Conference for High Performance Computing, Networking, Storage and Analysis (SC'18), 2018. 査読有
5. K. Yamada, T. Katagiri, H. Takizawa, K. Minami, M. Yokokawa , T. Nagai and M. Ogino, "Preconditioner auto-tuning with deep learning for sparse iterative algorithms, "The Sixth International Symposium on Computing and Networking Workshops (CANDARW), pp.257-2622, 2018. 査読有
6. Y. Kawarabatake, M. Agung, K. Komatsu, R. Egawa and H. Takizawa, "Use of Code Structural Features for Machine Learning to Predict Effective Optimizations," 2018 IEEE International Parallel & Distributed Processing Symposium Workshops, pp.1049-1055, 2018. 査読有
7. M. A. Amrizal, P. Li, M. Agung, R. Egawa, and H. Takizawa, "A Failure Prediction-Based Adaptive Checkpointing Method with Less Reliance on Temperature Monitoring for HPC Applications," IEEE International Conference on Cluster Computing (CLUSTER2018), pp.512-523, 2018. 査読有
8. X. Xiao, M. Agung, M. A. Amrizal, R. Egawa and H. Takizawa, "Investigating the Effects of Dynamic Thread Team Size Adjustment for Irregular Applications," The Sixth International Symposium on Computing and Networking (CANDAR), pp.76-84, 2018. 査読有
9. H. Cui, S. Hirasawa, H. Kobayashi and H. Takizawa, "A Machine Learning-Based Approach for Selecting SpMV Kernels and Matrix Storage Formats," IEICE Transactions on Information and Systems, Volume E101-D(9), pp.2307-2314, 2018. 査読有
10. H. Takizawa, M. A. Amrizal, K. Komatsu, and R. Egawa, "An Application-Level Incremental Checkpointing Mechanism with Automatic Parameter Tuning," The Fifth International Symposium on Computing and Networking, International Workshop on Legacy HPC Application Migration (LHAM2017), pp. 389-394, 2017. 査読有
11. M. Agung, M. A. Amrizal, K. Komatsu, R. Egawa and H. Takizawa, "A Memory Congestion-aware MPI Process Placement for Modern NUMA Systems," 24th IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC2017), pp.152-161, 2017. 査読有
12. H. Takizawa, T. Reimann, K. Komatsu, T. Soga, R. Egawa, A. Musa and H. Kobayashi, "Vectorization-Aware Loop Optimization with User-Defined Code Transformations," IEEE International Conference on Cluster Computing (CLUSTER2017), Workshop on Re-Emergence of Vector Architectures (REV-A), pp.685-692, 2017. 査読有
13. M. A. Amrizal and H. Takizawa, "Optimizing Energy Consumption on HPC Systems with a Multi-level Checkpointing Mechanism," 12th International Conference on Networking, Architecture, and Storage(NAS2017), pp.1–9, 2017. 査読有
14. H. Takizawa, D. Sato, S. Hirasawa, and D. Takahashi, "A Customizable Auto-Tuning Scenario with User-Defined Code Transformations," 2017 IEEE International Parallel & Distributed Processing Symposium Workshops, pp.1372-1378, 2017. 査読有
15. X. Xiao, S. Hirasawa, H. Takizawa, and H. Kobayashi, "The Importance of Dynamic Load Balancing among OpenMP Thread Teams for Irregular Workloads," International Journal of Networking and Computing, Volume 7(2), pp.387-404, 2017. 査読有
16. H. Cui, S. Hirasawa, H. Takizawa, and H. Kobayashi, "A code selection mechanism using deep learning," IEEE 10th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC-16), pp.385-392, 2016. 査読有

〔学会発表〕(計２８件)

1. M. A. Amrizal, M. Agung, R. Egawa, and H. Takizawa, "An Energy Optimization Method for Hybrid In-Memory Checkpointing," IEEE Symposuim on Low-Power and High-Speed Chips and Systems (COOL Chips 22), 2019.
2. 滝沢寛之, "NEC SX-Aurora TSUBASA 向けプログラムチューニング技術," 最新アーキテクチャ向けプログラムチューニング技術ワークショップ, 2019.
3. H. Takizawa, N. Ebata, M. Agung, M. A. Amrizal, R. Egawa, Y. Isobe, and R. Takaki, "Memory First! A performance tuning strategy focusing on memory access patterns,"

The 29th Workshop on Sustained Simulation Performance, 2019.

4. 南一生, “スーパーコンコンピュータにおける性能最適化技術,” 最新アーキテクチャ向けプログラムチューニング技術ワークショップ, 2019.
5. T. Katagiri, and K. Yamada, “Auto-tuning of Preconditioners with Deep Learning,” SIAM Conference on Computational Science and Engineering (CSE19), 2019.
6. T. Katagiri, “Toward Auto-tuning of Preconditioners for Sparse Iterative Solvers by Deep Learning,” 2019 Conference on Advanced Topics and Auto Tuning in High-Performance Scientific Computing (ATAT2019), 2019.
7. T. Katagiri, “Auto-tuning of Preconditioner Selection for Sparse Iterative Solvers -- Adaptation of Deep Learning and its Limitations,” 2018 Conference on Advanced Topics and Auto Tuning in High-Performance Scientific Computing (ATAT2018), 2018.
8. K. Minami, “Expectation to Supercomputer Benchmark from the Viewpoint of Performance Optimization Technology,” SIAM Conference on Parallel Processing for Scientific Computing (PP), 2018.
9. T. Katagiri, S. Ichimura and K. Yamada, “High Precision Computing of Matrix-Matrix Multiplications and a New Approach of Auto-Tuning to Numerical Libraries by Deep Learning,” SIAM Conference on Parallel Processing for Scientific Computing (PP) 18, 2018.
10. H. Takizawa, “User-Defined Code Transformation for Separation of Performance-Awareness from Application Codes,” SIAM Conference on Parallel Processing for Scientific Computing (PP), 2018.
11. H. Takizawa, “Towards prediction of effective optimizations in performance engineering,” The 27th Workshop on Sustained Simulation Performance, 2018.
12. H. Takizawa, M. A. Amrizal, K. Komatsu, and R. Egawa, “Automatic Parameter Tuning of Application-Level Incremental Checkpointing,” 2018 Conference on Advanced Topics and Auto Tuning in High-Performance Scientific Computing (ATAT in HPSC 2018), 2018.
13. 塩月信智, 江川隆輔, 滝沢寛之, “SX-Aurora TSUBASA におけるプロセス間通信の性能評価,”情報処理学会ハイパフォーマンスコンピューティング研究会, 2018-HPC-165(21), 2018.
14. H. Takizawa, T. Reimann, K. Komatsu, T. Soga, R. Egawa, A. Musa, and H. Kobayashi, “Expressing the Differences in Code Optimizations between Intel Knights Landing and NEC SX-ACE Processors,” The 13th World Congress on Computational Mechanics/2nd Pan American Congress on Computational Mechanics, 2018.
15. 滝沢寛之, “SX-Aurora TSUBASA の基本性能および機能の初期評価,” SX-Aurora TSUBASA フォーラム, 2018.
16. 後藤啓, 横川三津夫, 坂敏秀, “建物の地震動応答シミュレーションに現れる前処理付き共役勾配法の並列化,” 情報処理学会ハイパフォーマンスコンピューティング研究会, 2018-HPC-167(28), 2018.
17. 山田 賢也, 片桐 孝洋, 永井 亨, 荻野 正雄, “ディープラーニングを用いた数値計算ライブラリにおける反復解法の前処理選択の検討,” 日本応用数理学会 2018 年度年会, 2018.
18. T. Katagiri, “Code Optimization with ppOpen-AT and Preconditioner Selection by Deep Learning,” International Symposium on Research and Education of Computational Science (RECS2018), 2018.
19. 山田賢也, 片桐孝洋, 永井亨, 荻野正雄, “ディープラーニングによる数値計算ライブラリのチューニングパラメタの最適化の試み,” 第 23 回計算工学講演会, 2018.
20. 山田賢也, 片桐孝洋, “ディープラーニングによるカラー画像を用いた疎行列反復解法ライブラリの自動チューニング,” NVIDIA GPU Technical Conference in Japan 2017 (GTC Japan 2017), テクニカルセッション, 2017.
21. 滝沢寛之, 崔航, 平澤将一, “機械学習によるコード最適化の可能性,” 計算工学講演会 C-2-2, pp.20, 2017.
22. 山田賢也, 片桐孝洋, 永井亨, 荻野 正雄, “疎行列形状のカラー画像を入力としたディープラーニングによる数値計算ライブラリの自動チューニング方式,” 情報処理学会ハイパフォーマンスコンピューティング研究会, 162-HPC-2017, 2017.
23. 山田賢也, 片桐孝洋, 永井亨, 荻野正雄, “ディープラーニングを用いた数値計算ライブラリの最適実装選択の検討,” 情報処理学会第 79 回全国大会, 2017.
24. H. Takizawa, “Combining Autotuning and Code Transformations,” 2017 Conference on Advanced Topics and Auto Tuning in High-Performance Scientific Computing (ATAT in HPSC 2017), 2017.
25. H. Takizawa, “Performance Tuning with Machine Learning,” The 25th Workshop on Sustained Simulation Performance, 2017.
26. T. Ohichi, M. Terai, M. Yokokawa, and K. Minami, “STView: An Eclipse Plug-in Tool

for Visualizing Program Structures in Fortran Source Codes," The International Conference for High Performance Computing, Networking, Storage and Analysis (SC2016), 2016.
27. H. Takizawa, D. Sato, S. Hirasawa, and H. Kobayashi, "Making a legacy code auto-tunable without messing it up," The International Conference for High Performance Computing, Networking, Storage and Analysis (SC2016), 2016.
28. 川原畑勇希，平澤将一，滝沢寛之，小林広明，"機械学習を用いたコード変換に関する研究，" 平成 28 年度電気関係学会東北支部連合大会，2016.

〔図書〕（計　2件）
1. M. Geshi, K. Minami, et al, "The Art of High Performance Computing for Computational Science Vol.2," Springer, 頁数未定, 2019(in press).
2. 下司雅章編 南一生他著，"計算科学のための HPC 技術 2"，340 頁，大阪大学出版会 2017.

６．研究組織

(1)研究分担者
研究分担者氏名：片桐 孝洋
ローマ字氏名：Takahiro Katagiri
所属研究機関名：名古屋大学
部局名：情報基盤センター
職名：教授
研究者番号（8 桁）：40345434
研究分担者氏名：横川 三津夫
ローマ字氏名：Mitsuo Yokokawa
所属研究機関名：神戸大学
部局名：先端融合研究環
職名：教授
研究者番号（8 桁）：70358307
研究分担者氏名：南 一生
ローマ字氏名：Kazuo Minami
所属研究機関名：理化学研究所
部局名：計算科学研究機構
職名：チームヘッド
研究者番号（8 桁）：70501998

(2)研究協力者
研究協力者氏名：小林 広明
ローマ字氏名：Hiroaki Kobayashi
研究協力者氏名：須田 礼仁
ローマ字氏名：Reiji Suda
研究協力者氏名：岡谷 貴之
ローマ字氏名：Takayuki Okatani
研究協力者氏名：江川 隆輔
ローマ字氏名：Ryusuke Egawa
研究協力者氏名：大島 聡史
ローマ字氏名：Satoshi Ohshima