

平成 30 年 6 月 15 日現在

機関番号：32503

研究種目：挑戦的萌芽研究

研究期間：2014～2017

課題番号：26540031

研究課題名(和文)事例に基づく性能チューニング支援システムの研究

研究課題名(英文)A Study of Evidence-Based Performance Tuning

研究代表者

橋本 政朋 (HASHIMOTO, Masatomo)

千葉工業大学・人工知能・ソフトウェア技術研究センター・上席研究員

研究者番号：60357770

交付決定額(研究期間全体)：(直接経費) 2,900,000円

研究成果の概要(和文)：様々な大規模科学技術計算プログラムに対してその実行効率を向上させるための詳細な助言を得ることのできるシステムの構築に向けて、専門家により実際に行われた性能チューニング事例を収集し、プロファイリング結果や、ソースコード変更履歴を解析することにより得られた最適化パターン情報などを蓄積したデータベースを構築した。これに基づき最適化パターン予測モデルを構築しその評価を行った。また、1000もの科学技術計算オープンソースプロジェクトに対して、プログラムに含まれるループ構造に注目した大規模調査を実施し、その結果に基づきチューニング対象とすべきループ構造を自動同定するツールを開発し公開した。

研究成果の概要(英文)：Performance tuning is still a demanding manual task. To improve an application's efficiency, we have to identify its computational kernels, each of which is typically composed of one or more loops. Then various empirical attempts such as loop transformations are made. Thus, it is crucial to learn from the experience of performance tuning experts. As a proof-of-concept, we extracted various facts from performance tuning histories of a few real-world scientific applications, and then constructed a database of that facts. Based on the database, we constructed a few experimental predictive models for promising loop transformation patterns. We also explored a thousand computation-intensive applications to reveal the distribution of kernel classes, each of which is related to expected efficiency and specific tuning patterns. In addition, we constructed a binary classifier for identifying loop kernels and a multi-class classifier for predicting kernel classes.

研究分野：情報学

キーワード：高性能計算 性能チューニング 最適化パターン推定 計算カーネル同定 目標実行効率推定 プログラム構造理解支援 プログラム解析 機械学習

1. 研究開始当初の背景

大規模科学技術計算プログラムをスーパーコンピュータにおいて意図通り効率的に実行するためには、職人芸を駆使した人手によるパラメタ調整やプログラム書き換えを行う性能チューニングと呼ばれる工程が必要となる。この工程の完全な自動化は非常に困難であり現時点では不可能であるものの、その支援に関する研究が行われている。

自動チューニング技術[1]は、同一の処理を実現する複数の実装候補を用意し、それらに対し測定した性能値を比較することで最適な実装を選択する技術である。その特徴から主に計算ライブラリの付加機能として実装される。PerfExpert[2]は米国テキサス大学 Texas Advanced Computing Center で研究されている HPC (High Performance Computing) アプリケーションのための性能診断ツールである。PerfExpert は HPC アプリケーションの性能測定値からボトルネックを推定し、あらかじめ用意された知識ベースに照らし一般的な最適化パターン例を提案する機能を持っている。

研究代表者らは、細粒度チューニング事例データベースの構築に関する予備的研究を行っている。この研究は、機械的処理が可能な形でチューニング事例の要点を自動的に抽出しデータベースに蓄積することを目的としており、書き換えられた一連のソースコードやプロファイリングデータを含むチューニング事例から、適用されたと推定されるチューニングパターンを同定し、ソースコードの抽象構文木やプロファイリングデータとリンクさせたグラフ構造としてデータベースに蓄積することを目指すものである。

2. 研究の目的

大規模科学技術計算プログラムの性能(実行効率)を向上させる性能チューニング専門家の知識をデータベース化し、それを検索することで様々なプログラムに対して性能向上のための詳細な助言を得ることのできるシステムの構築に向けて、(1) 細粒度事例データベースに基づく高度な事例検索を実現するための新しいソースコード分類手法と、(2) プログラム書換えの正当性(計算結果を変えないなど)の検証手法の効率化とを目指す。

3. 研究の方法

目的(1)の達成のため、研究代表者らが研究を重ねている、抽象構文木のノード単位でソースコードの構造比較を行う細粒度差分解析手法[3]とそれに基づく系統分析手法とを用いる。ソースコード構造比較に関する研究自体多くはなく、さらに細粒度構造比較による系統分析の試みは知る限り存在しない。また、目的(2)の達成のため、細粒度差分解析により、

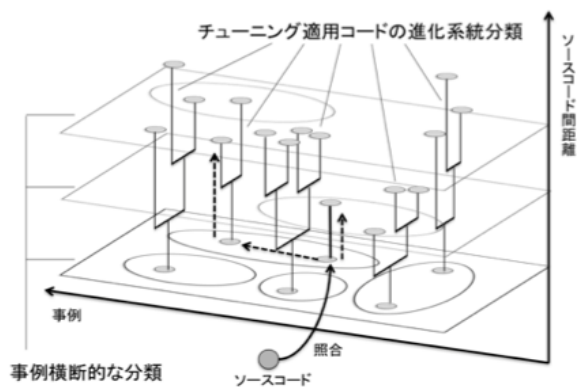


図1 高度事例検索

似通った一連のソースコード群の最大共通部分を見つけ、計算コストの大きい検証の対象を削減することで効率化を図る。大目標達成に向けて歩を進めることで、性能チューニング行程の効率化が促進されることが期待される。

(1) 事例ソースコードの新しい分類手法

手元のプログラムに似たものは過去の事例においてどのようにチューニングされてきたか、といったような、チューニング作業の助けとなる高度な検索を可能とするためには、事例に含まれるソースコードを的確に分類する必要がある。研究代表者らは細粒度ソースコード差分解析手法を用いて、分子系統学の手法を応用し、祖先・子孫関係にある一連のプログラム間の進化距離を計算し、進化系統樹を復元することに成功しており、良好な結果を得ている。この手法を用いることで、事例に含まれるソースコード群のみから元のソースコードを祖先とするチューニング過程を復元することができる。この情報をデータベース化することで、任意のソースコードをキーとしたチューニング過程の検索を可能とする

(図1)。ところが当然ながら、キーとなるソースコードは事例に含まれているもののみとはならず、多岐にわたると予想される。そこで本研究では、チューニング対象となるカーネルと呼ばれるプログラムの構造のみを比較対象とし、さらに本質的でない詳細を隠蔽するための正規化を行う。カーネルは、チューニングの際にプロファイリング結果に基づき手作業で抽出され既に事例に含まれている場合もあるが、そうでない場合には、新しい試みとして、機械学習を用いたカーネルの抽出にも挑戦する。

(2) ソースコード書換えの正当性検証手法

チューニングによるプログラムの書換えの正当性の検証については、階差的シンボリック実行(differential symbolic execution)と呼ばれる手法を基に、我々の細粒度ソースコード差分解析手法を組み合わせることで改良を試みる。シンボリック実行とは、プログラムに対して、実際の入力代わりに任意の入

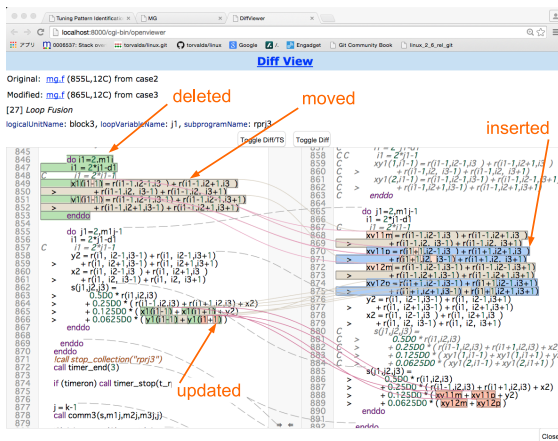


図 2 細粒度差分解析

力を表現する記号を与え、実行パスを可能な限り網羅的に探索することで仮想的に実行するもので、テストケースの生成などに用いられる。階差的シンボリック実行は2つのプログラムをシンボリック実行する際に、共通部分に対する探索を重複して行うことを避けることで、インクリメンタルにテストケースを生成したり、プログラムの振る舞いの差を計算したりするために用いられる。この共通部分を見いだすために差分解析が必要となるが、これまでの単純な比較手法に換え、細粒度差分解析を組み合わせることで効率化や探索範囲の拡大に挑戦する。

4. 研究成果

(1) 性能チューニング事例の収集とそのデータベースへの蓄積、並びに最適化パターン予測モデルの構築

事例の収集に関しては、理化学研究所計算科学研究機構運用技術部門ソフトウェア技術チーム(当時)の協力により、NICAMと呼ばれる気象シミュレータを含む幾つかの大規模科学技術計算アプリケーションの性能チューニング事例の収集を行った。それらの事例には実際にスーパーコンピュータ上で実行するために施された性能チューニングの記録、具体的には試行錯誤の過程で生じたソースコードのバリエーションとそれらを実行し測定した性能プロファイリングデータが含まれている。細粒度差分解析によりバリエーション間の差分を解析し(図2)、適用されたと思いき最適化パターンを同定(図3)した上で、性能プロファイリングデータと共に RDF(Resource Description Framework)形式と呼ばれるデータ形式に変換した後、OpenLink Software社製の Virtuoso と呼ばれるデータベース管理ソフトウェアを用いてデータベースに蓄積した。

さらに当初の研究計画の範囲を超えるが、事例検索の応用として、機械学習を用いた最適化パターン予測モデル構築にも取り組んだ。これは、ソースコードの特徴を表現する数値ベクトルから、その性能を向上させると見込

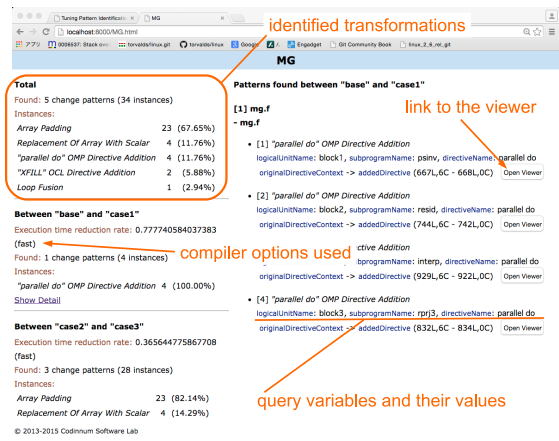


図 3 最適化パターン同定

まれる(既知の)ソースコード変更パターンを予測するものである。ソースコードの特徴ベクトルとして性能プロファイリングデータと、配列変数の数などの幾つかのソースコードメトリクスとを用いた。実行時間の短縮に成功した変更パターンとソースコード特徴ベクトルとの組合せを BR-kNN と呼ばれる機械学習アルゴリズムに教師データとして与え、予測モデルを構築した。事例数がまだ十分とは言えずパラメータ調整などの課題も残るものの、交差確認による評価から、その有効性を見て取ることができた。実際の事例から最適化パターン予測に至る手法にまで踏み込むことができたことは計画以上の進展であり意義深い。以上の成果は学会発表④(招待講演)として発表し、さらに論文②としてまとめ国際会議において発表③を行った(採択率30.2%)。

成果の位置づけ

Javaで記述されたプログラムに対して適用されたリファクタリングパターンを同定する試みはあったが、Fortranで記述されたプログラムに対して適用された最適化パターンを同定する試みは知る限り存在しない。リファクタリングにおいて主眼とされるクラスの移動やメソッドの抽出などの比較的大きな構造単位の変更と比べ、性能チューニングで行われるループの分割や配列の融合などは、より細かい変数や式単位の変更であり、それを正確に捉えることは、より困難であると考えられる。

今後の展望

まず考えられるのは、より多くの事例に対して提案手法を適用することであるが、後述するようにそれは容易なことではない。

本課題ではこれまで蓄積されてきた性能チューニング事例に基づき、究極の目標としてチューニング作業の一部の自動化を行うことを目指しているが、実際には、事例はあれどもその作業履歴の記録が行われているとは限らず、また行われていたとしても、ソースコードや作業履歴などが、機密情報に当たり入手不可能であるという事態に直面した。この事例の絶対数不足という状況を打開するため、

Class	Traits	Typical Applications	Efficiency
M1	Matrix-matrix multiplication	Density functional theory	
M2	Simple loop body	Molecular dynamics	
M3	Simple loop body + stencil	Particular high-order accuracy stencil	
M4	Complex loop body	Climate model (physical), plasma (particle-in-cell)	
M5	Stencil	Climate model (dynamical), fluid dynamics, earthquake	
M6	Stencil (indirect array reference)	Fluid dynamics (finite element method)	

図4 カーネルクラス

当初研究計画にあったパッチ正当性検証手法に関する研究を取り止め、それに代わる新たな試みとして、オープンソースプロジェクトに対する大規模調査を実施することとした。

(2) カーネルの同定とそれに対する目標実行効率の推定

性能チューニング工程の初期段階において実施する重要な作業の一つに、チューニング対象となるループ構造（計算ループカーネル、以後単にカーネルと呼ぶ）の同定と分類がある。この分類に基づき目標とすべき実行効率を見積もることができる。性能チューニング全体もさることながら、この特定と分類という作業も専門家の経験と感とに依存している。この作業の支援のため、プログラムに含まれるループを、その構造から抽出された特徴ベクトルに基づき機械学習を適用することで自動分類する実験を行った。

専門家による京コンピュータ向けの性能チューニングにおいては、カーネルを期待実行効率の範囲に基づき6つのクラス(図4)に分類するということが行われている。この分類をモデル化するために、GitHubから科学技術計算と思しきFortranアプリケーションを1020個収集し、そこに含まれる175,963個のループ構造の中から無作為抽出した100個について2名の専門家に上述クラスに基づく分類作業を依頼した。

この分類作業をスムーズに進めるために、専門家が実際作業を行う上でどのような情報を参照し、どのように判断してゆくのかを備に聴取し、それに基づき情報提示ツールを新たに開発した。このツールは、GitリポジトリのURLを与えるだけで、そこに含まれるFortranプログラムの構文解析、RDF形式でのデータベース構築、情報閲覧ならびに分類結果入力のためのウェブアプリケーション生成を自動的に行う。これにより、プログラム構造や副プログラム呼び出しツリーのアウトライン、ループ構造の特徴値を容易に確認することができる(図5)。

数ヶ月を経て得られた専門家によるループ分類の結果に基づき、分類データセットを作成した。その結果、ループの母集団において、70-85%はカーネルであり、その中でM5と呼ばれるクラスに属するものが45-70%と突出して多いことが統計的に推定された(図6)。M5ク

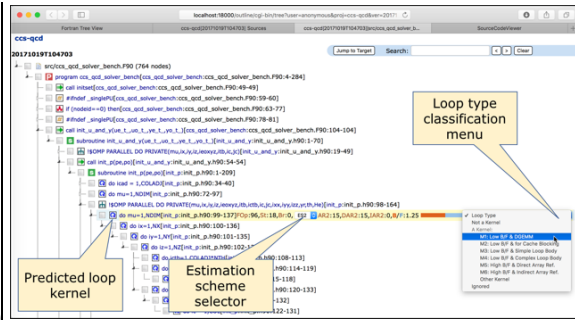


図5 プログラム構造理解支援ツール

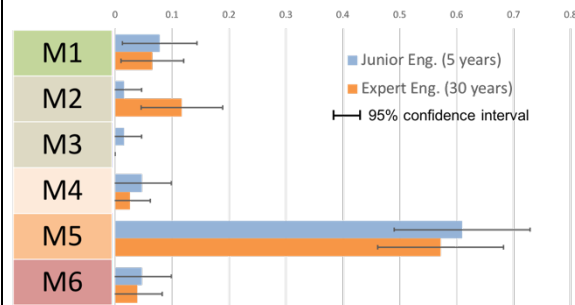


図6 カーネルクラス分布

ラスのカーネルは浮動小数点演算に対するメモリアクセス要求量の比率が比較的大きく、京コンピュータにおける実行効率（ピーク性能比）は高々15%ほどとされている。

また、このデータセットを機械学習の訓練データとして使い、カーネル判定モデルとM5クラス判定モデルの構築も行った。このモデルを交差確認により評価したところ、ループに対するカーネルか否かの判定正解率は81%、カーネルのM5クラスか否かの判定正解率は94%という結果が得られた。これらの成果を論文①としてまとめ、国際会議において発表②を行った(採択率33.8%)。

成果の位置づけ

これまで数十程度の科学技術計算アプリケーション開発プロジェクトに対する調査はあったが、1000ものプロジェクトの調査は類を見ない。ましてやその1000ものプロジェクトに含まれるFortranで記述されたプログラムを、ビルドすることなく完全自動で構文解析するなど有り得なかった。さらに、対象となった母集団における結論とはなるが、カーネルクラスの分布が非常に偏っており、大部分のカーネルは京コンピュータにおいて高々15%のピーク性能比を期待できる、ということが今回初めて明らかとなった。

今後の展望

今回の大規模調査ではGitHubにおいて公開されているFortranで記述された計算プログラムのみを対象としたが、他のソースコードホスティングサービスや他のプログラミング言語を対象とするなど、より広範な調査を実施し、統計的な確度を向上させてゆくことが望ましい。

(3) プログラム理解支援ツールの公開

専門家によるループ分類データセット作成作業支援のために、コード理解支援並びに分類結果収集を行うツールの開発を行ったが、性能チューニング現場への成果還元という観点から、これを単体のアプリケーションとして動作するように改修し、オープンソースソフトウェアとして公開した。改修にあたっては、専門家の意見を取り入れつつ、研究成果に基づくカーネル予測機能を盛り込み、実際の性能チューニング作業におけるプログラム構造理解支援ツールとしての利用を想定した改良を行い、その概要をポスターとしてまとめ、国際会議において発表①を行った。このツールは、まだ数例ではあるものの、ポスト京の開発に資する性能チューニング業務において実際に専門家に利用されるに至っている。

成果の位置づけ

専門家による性能チューニング作業は、通常第三者によって書かれたプログラムを対象とするため、作業初期におけるプログラム構造理解、カーネルの同定・分類は重要かつ手間のかかる作業となっている。プログラム構造理解支援ツールは、市販のものも含めて数多あるが、性能チューニングに特化し、カーネル予測機能を備えたものは知る限り存在しない。

今後の展望

専門家による実際の性能チューニング業務における使用に耐えるツールとして、またさらなる研究成果を展開するための枠組みとして、発展させてゆくことが望ましい。

<引用文献>

[1]汎用自動チューニング機構を実現するためのソフトウェア基盤の研究, 須田礼仁, 科研費 研究課題番号: 23240005

[2]PerfExpert: An Easy-to-Use Performance Diagnosis Tool for HPC Applications, M. Burtscher, B.D. Kim, J. Diamond, J. McCalpin, L. Koesterke, and J. Browne, SC2010, DOI:10.1109/SC.2010.41

[3]Diff/TS: A Tool for Fine-Grained Structural Change Analysis, M. Hashimoto, A. Mori, WCRE2008, DOI:10.1109/WCRE.2008.44

5. 主な発表論文等

[雑誌論文] (計 2 件)

- ① Masatomo Hashimoto, Masaaki Terai, Toshiyuki Maeda, Kazuo Minami, An Empirical Study of Computation-Intensive Loops for Identifying and Classifying Loop Kernels, Proceedings

of the 8th ACM/SPEC on International Conference on Performance Engineering, Refereed, 2017, pp. 361-372
DOI:10.1145/3030207.3030217

- ② Masatomo Hashimoto, Masaaki Terai, Toshiyuki Maeda, Kazuo Minami, Extracting Facts from Performance Tuning History of Scientific Applications for Predicting Effective Optimization Patterns, Proceedings of the 12th Working Conference on Mining Software Repositories, Refereed, 2015, pp. 13-23, DOI:10.1109/MSR.2015.9

[学会発表] (計 4 件)

- ① Masatomo Hashimoto, Masaaki Terai, Toshiyuki Maeda, Kazuo Minami, CCA/EBT: Code Comprehension Assistance Tool for Evidence-Based Performance Tuning, International Conference on High Performance Computing in Asia-Pacific Region, Poster, Tokyo, January 28-31, 2018
- ② Masatomo Hashimoto, An Empirical Study of Computation-Intensive Loops for Identifying and Classifying Loop Kernels, 8th ACM/SPEC International Conference on Performance Engineering, L'Aquila, Italy, April 22-27, 2017
- ③ Masatomo Hashimoto, Extracting Facts from Performance Tuning History of Scientific Applications for Predicting Effective Optimization Patterns, 12th Working Conference on Mining Software Repositories, Florence, Italy, May 16-17, 2015
- ④ 橋本政朋、根拠に基づく性能チューニングの支援に向けて、第 6 回自動チューニング技術の現状と応用に関するシンポジウム、招待講演、東京、2014.12.25

[その他]

ホームページ等

<https://stair.center/archives/projects/ebt>

6. 研究組織

(1) 研究代表者

橋本 政朋 (HASHIMOTO, Masatomo)

千葉工業大学・人工知能・ソフトウェア技術
研究センター・上席研究員

研究者番号: 60357770