

特集 科学技術計算におけるソフトウェア自動チューニング

< 概論 >

ソフトウェア自動チューニングの枠組み

弓場 敏嗣 電気通信大学

片桐 孝洋 東京大学

ソフトウェア自動チューニングとは何か？

■ 背景

近年、高性能コンピューティングの世界では、新しいマシン環境に性能を自動的にチューニングすることによって高速化を実現するソフトウェア自動チューニング技術が注目を集めている。半導体技術の進歩により、高性能コンピューティングを支えるプロセッサ(マシン)の技術革新は絶えることなく、その開発サイクルは短くなっている。マシン開発サイクルの短縮化と同時に、マシンアーキテクチャの複雑度は増し、ソフトウェア開発に要する費用と時間の増大化傾向が一層強まっている¹⁾。

高性能マシンを提供する事業者とその性能を享受する利用者にとって、旧世代のマシン環境で構築されたソフトウェア資産を、新世代マシン環境のもとで継承する必然性がある。事業者側では、オペレーティングシステム(OS)、コンパイラなどのシステムソフトウェアについて、対象マシンアーキテクチャの性能を最大限に利用するために、新世代マシンのソフトウェア環境の整備に努力が払われている。しかしながら、システムソフトウェアに比べ、応用ソフトウェアの優先度は相対的に低いという現状がある。応用ソフトウェアを開発する場合、高性能マシンが保持する機能を有効利用するために、応用、ライブラリ、マシンアーキテクチャについての知識と実装のスキルが不可欠である。しかし、そうした高度なプログラミング能力を持つ人材は多くはない。ソフトウェア資産の再利用は、ソフトウェア生産性の向上を企図するソフトウェア工学への期待と合わせて、高性能マシンの事業者にとって急を要する課題となっている。また、数値計算ライブラリなどの応用ソフトウェアの利用者にとっては、永年の資産であるソフトウェアを新世代マシンの環境下で再利用できることが重要である。

このような背景のもとで、ソフトウェア自動チューニング技術は、数値計算ライブラリなどの旧世代ソフトウ

ェア資産を活用する手段として提案された¹⁾。旧世代マシン環境向けに開発されたライブラリを新世代マシン環境に移植(インストール)するとき、新マシン環境の持つ高い性能を自動的に引き出すことが、ソフトウェア自動チューニング技術の狙いである^{☆1)}。

■ 定義

ソフトウェア自動チューニングは、ソフトウェアAをソフトウェアBによってマシン環境に適合させ、自動的に性能をチューニング(あるいは最適化)することと定義できる。ここで、Aはチューニングされるソフトウェア(チューニング対象プログラム)、Bはチューニングを行うソフトウェアである。<自動>の狭義な意味合いは、異なる機種の対象マシンに対しても高性能が得られるように、与えられた応用(数値計算)プログラムをチューニングする過程を自動化することから出発している²⁾。つまり、人手によるチューニング作業を自動化することがそもそもの意味である。また、チューニングの際に基準となる性能は、プログラムの実行速度にかぎらない。マシン環境を構成するネットワークの通信速度、あるいは計算と通信の両者を包含した性能として応答速度が性能基準として用いられる場合もある。速度以外にも、所要記憶量、消費電力量、信頼性など性能基準として設定可能な対象は、ソフトウェア自動チューニングの枠組みに含まれる。たとえば、数値計算応用においては、速度に代わって計算精度の保証が性能基準となる場合がある。評価対象とするマシン環境の範囲を広げたとき、それら

☆1) 自動チューニングは、普遍性、一般性のある技術用語である。たとえば、ラジオ受信機の周波数調整、自動車エンジンのチューンナップ、楽器の音合わせなどに対して、チューン(tune:調整, 調律, 調和)という言葉が用いられる。三省堂国語辞典では、「具合の悪いところを整える」という部分修正の意味合いで説明されている。自動チューニングという語彙には、技術の隙間を追究し、丁寧に磨き上げる感覚がある。本来、科学というより、職人の技術領域を指し示す地味な用語である。計算機科学の分野では、形式化という科学と有効性という工学の両方の視点から、ソフトウェア自動チューニング技術を位置づける必要がある。

1 ソフトウェア自動チューニングの枠組み

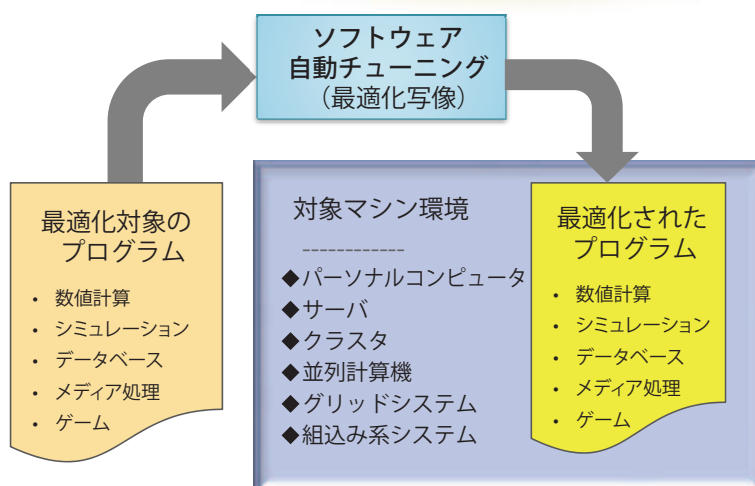


図-1 ソフトウェア自動チューニングのモデル

の間で性能の可搬性が保たれることを性能基準にする場合も考えられる。さらに、プログラムの行列サイズなどの変化によってもたらされる実行時間の変動の程度を考慮して、より少ない変動をもたらす性能安定性を基準とする場合もある。

最適化対象のプログラム(問題)をマシンに実行可能な形に写像する過程で、対象マシン上で当該プログラムが効率よく実行されるように、ソフトウェアによる変換(自動チューニング)が行われる(図-1)。写像という言葉は、プログラムとして記号表現された問題を、対象マシン上で実行可能な表現に変換することを意味する。この対象プログラムの写像過程には、いくつかの段階が存在する。ある段階では、ソースプログラムからソースプログラムへの写像のように、プログラム自身の記述水準は変わらない。評価基準(目的関数)を定め、それに基づき最も高性能を得るための変換(あるいはチューニング)を最適化という。写像過程では最適化が繰り返される。自動チューニングが対象とするプログラムの事例としては、数値計算応用が典型的である。最近では、各種のシミュレーション、データベース検索、メディア処理、ゲームなども、自動チューニングの対象プログラムに含まれる³⁾。対象マシン環境には、それらのプログラムが実行されるパーソナルコンピュータ、サーバ、クラスタ、並列計算機、グリッドシステム、組込み系システムなどが含まれる。

計算機科学における関連領域—コンパイラとの差異について

ソフトウェア自動チューニング技術は、高性能コン

ピューティング分野における計算科学(Computational Science)からの要請を受けて生まれている。自動チューニングの技術的枠組みは、古くは自動制御理論が対象としてきた領域である。今後は、計算機科学(Computer Science)という、より大きな立場からとらえるべきパラダイムと考える。関連の深い計算機科学分野として、オペレーティングシステム(OS)、並列処理、コンパイラが存在する。OSのスケジューリングは、タスク、プロセス、スレッド、ジョブなどの制御単位に対して、システム全体のスループット、当該プログラムの優先度などの評価基準から、最適なマシン割り当てすなわち動的な性能のチューニングを行う。並列処理では、対象とする並列マシン環境に、与えられたプログラムを並列化し、それによって最大限の高速性能を引き出すことを目指すとともに、ソフトウェア自動チューニング技術と密接な関係を持つ。

コンパイラは、ソースプログラムを目的プログラムに静的に変換する過程で最適化を行う。当然ながら、自動チューニングシステムのプログラム水準の最適化では、コンパイラが行う最適化を可能なかぎり利用すればよい。ただし、コンパイラ最適化は、マシン環境が変化する状況への対応を考慮していない。また、コンパイラの最適化技法が適用されるのは、決められたマシン環境における、基本ブロックと呼ばれるプログラム単位に対してである。対象とするプログラム全体から見れば、たとえばループなどきわめて局所的な部分に対して最適化が行われるに過ぎない。一方、自動チューニングの枠組みでは、複数のマシン環境を前提として、もとのプログラムと同じ実行結果をもたらすことが保証されるかぎり、最適化のために命令を全域的に並べかえることを許し、複雑

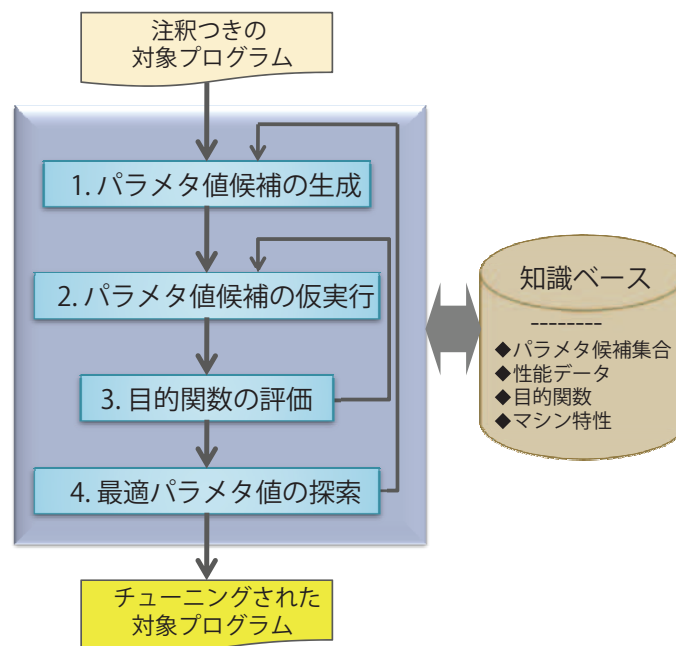


図-2 ソフトウェア自動チューニング機構

な構造を持つプログラムに対して適用し得る。たとえば、ソーティングのアルゴリズムをより高速化が得られるものに置き換えるなど、アルゴリズム水準の変換による最適化を許容する。

自動チューニングにおいて、全域的な最適化を行うためには、部分的な最適化段階の適切な適用順序を決定する。可能な組合せに従って最適化の適用順序を決める必要があり、一般には実験的試行によってしか決定できない。これらの試行結果に基づいて、より効果の大きい最適化を実現する。さらに、自動チューニングでは、動的最適化、すなわち、手続き呼び出し時やプログラム実行時の最適化を許容する。コンパイラと比べた場合、ソフトウェア自動チューニングは知識ベースを利用した学習機能を含め、時空間的に大局的な最適化の枠組みを提供する。

ソフトウェア自動チューニングの実現

ソフトウェア自動チューニングの機能は、チューニング対象の入力プログラムが対象マシンに写像される過程に存在する機構によって実現される(図-2)。対象プログラムを、自動チューニングに必要な情報を<注釈>として付加したプログラムの形で入力すると、ある性能基準に対して最適化されたプログラムが出力される²⁾。注釈付きの入力プログラムはプリコンパイルされ、所要の最適化機能を付加したソースプログラムに変換される。

注釈の言語処理系(プリコンパイラ)は、自動チューニングを支援するプログラミング環境であり、自動チューニング支援ツールといえる。自動チューニング機構は、入力された注釈および知識ベースに置かれた情報、仮実行によって得られた測定データを用いて、パラメタ値ごとにチューニングの有効性を評価し、最適化基準に従って最適パラメタ値を決定する。

注釈で与える情報には、以下のようなものがある。

- マシン環境の特性を表すパラメタ
- 問題を記述するプログラムによって規定される問題パラメタ
- 速度や精度などの性能に影響を及ぼす調整可能な調整パラメタ
- 最適化基準を示す目的関数

ここで、特性パラメタとは、使用するプロセッサの速度、記憶容量、キャッシュサイズ、プロセッサ数、結合網の通信性能などであり、対象マシン環境によって先験的に決まる。問題パラメタは、問題の行列サイズなど、与えられた問題固有の値である。また、調整パラメタは、自動チューニングすべき性能パラメタであり、ループ展開の段数、データ分割のブロック幅などが典型的である。チューニング時の選択候補にはアルゴリズム要素、プログラム部品、目的関数も含まれ、これらの候補集合の中から最適なものを選択することも、調整パラメタの最適化と考える。パラメタ値の仮実行では、部分的にコンパイル、実行し、その性能データを測定する。また、

1 ソフトウェア自動チューニングの枠組み

目的関数のパラメータは、フィードバックにより適切な値が決定される。

ソフトウェア自動チューニングにおいては、得られる高性能化の効用と機構導入の費用のバランスを考慮しなくてはならない。また、同機構は軽量であること、すなわち、最適化に要する空間的、時間的オーバーヘッドが小さいことが必要とされる。さらに、動的最適化を行う場合は、最適化に実時間性をもたさなくてはならない。同機構の一部である自動チューニング支援ツールは、利用者から見てプログラミング言語としての記述性と操作性がよいことが不可欠である。

ソフトウェア自動チューニング技術の世代論的考察

ソフトウェア自動チューニング技術を、10年きざみの世代論として考察する。

◆ 1990年代の第1世代：数値計算ライブラリを指向した自動チューニング

歴史的に見ると、ソフトウェア自動チューニング技術は、スーパーコンピュータを利用する数値計算応用の分野で生まれた。同分野は、過去において自動チューニングを最も必要とし、かつその適用が有効であった。この年代は、マシンの性能向上に追従した応用プログラムの改変作業の自動化の時代である。高性能マシン（スーパーコンピュータ）の頻繁な世代交代への対応が、主として、マシン事業者側から要請された。急速な技術革新に対して、人手による書き直し作業の限界が顕在化した。この世代の自動チューニングシステムは、ATLAS¹⁾によって代表される数値計算ライブラリが中心であり、キャッシュを意識したループ展開により性能をチューニングした。対象マシンへのライブラリのインストール時におけるチューニングであり、ソフトウェア自動チューニング技術の最初の成功例であった。

◆ 2000年代の第2世代：自動チューニング支援ツールを用いる自動チューニング

2000年代に入って、クラスタシステム、分散システム、グリッドシステムなどマシン環境の複雑化、対象アーキテクチャの多様化への対応が必要とされるようになった。多品種化と高性能化が著しい組込み系システム分野においても、マシン環境の開発サイクルの速さは同じ状況にある。多様なマシン環境でのプログラム開発を必要とする、主として応用プログラム開発者から、多様化への対応として自動チューニング支援ツールが要請された。多

様化対応のために、自動チューニング仕様を記述する専用言語の導入が図られた¹⁾。同言語では、対象マシンの特性パラメータのほか、チューニング対象領域、選択対象候補のプログラムおよびアルゴリズムの集合、調整パラメータ、評価に用いる目的関数の記述を与え、チューニングのタイミングなどを指定する。言語による先験的知識の付与と、チューニング実行時のモニタリングによる知識獲得が可能な機能を持つ。プログラム水準の最適化では、コンパイラとの役割分担が課題となっている。

◆ 2010年代の第3世代：後験的知識を利用する学習指向自動チューニング

2010年代には、ソフトウェア自動チューニングの有効性が浸透し、その適用対象が拡大すると予想される。第2世代の成果を踏まえ、プログラム開発者およびシステム管理者は、さらなる自動チューニングの効果を期待する。有効性を格段に向上させるためには、後験的に得られた知識を利用する機能が必要となる。実行してみないと分からない問題の最適化のためには、知識を蓄積する学習能力の適用が不可欠である。マシン事業者の手を離れた世界であり、広い範囲からの学習による組織的な知識獲得とその利用の一般化、有用な知識内容の選別と継承、応用プログラムの動的特性の考慮などが課題となる。

ソフトウェア自動チューニング技術の将来展望

今後の研究開発において考慮すべき方向性は、以下の通りである。

1. ソフトウェア自動チューニングパラダイムの適用範囲を拡大する。
2. ソフトウェア自動チューニング技術の実用化を推進する。
3. ソフトウェアの生産性を高める技術との統合を図る。
4. ソフトウェア自動チューニング技術の標準化を推進する。

ソフトウェア自動チューニング技術は、従来、その適用分野として数値計算応用を想定してきた。考え方の枠組みとしては、数値計算分野以外のさまざまな応用に対して適用可能である³⁾。その意味で、自動チューニング技術の研究は、次世代基盤ソフトウェアの構築に向けた重要な研究課題といえる。今後、自動チューニングパラダイムの適用が有効な結果をもたらす分野を積み上げる努力が必要である。高性能マシン環境に対する数値計算ライブラリから始まり、大規模科学技術計算応用、最終的にはシステムインテグレーション企業が対象とする大

規模ソフトウェアシステム構築への適用が期待される。ライブラリのように閉じたソフトウェアでない場合は、セキュリティ、障害対策などチューニングすべき課題の複雑度が増す。その場合、自動の程度に幅をもたせ、人が介在する部分を内包する形で進展が図られねばならない。パラダイムの適用範囲の拡大によって、情報基盤技術に多大の貢献がもたらされる。

現状では、数値計算応用に限定しても、現場からの要求を高いレベルで満足する自動チューニングの実用的な枠組みは存在しない。一度開発したソフトウェアを長期間にわたって利用するためには、ソフトウェア開発過程と自動チューニング過程が連動して最適化を進め、マシン環境の変化に柔軟に対応できる完成された自動チューニング方式が求められる。その実現のためには、多様化するマシン環境に、柔軟に適応する自動チューニング技術の実用的な枠組みを構築しなくてはならない。ソフトウェア技術者の不足が叫ばれる中で、自動チューニングパラダイムは、ソフトウェア構築の生産性を高めるものとして期待される。ソフトウェア工学におけるソフトウェア部品化とその最適組合せによるソフトウェア構築方法論は、自動チューニングの対象領域である。

自動チューニング技術は、新しく開発するマシンの機種ごとにライブラリを新しく作り直す手間を削減するという意味において、産業技術的観点からも重要である。高性能コンピューティング技術の分野で、外国企業との

競争に勝ち抜くために不可欠な、国際戦略性の高い情報基盤技術と位置づけることができる。そのためには、自動チューニング技術の標準化に力を入れる必要がある。合わせて、オープンソース化によって、多くの人たちの知の集積が企図されねばならない。

参考文献

- 1) 片桐孝洋：ソフトウェア自動チューニング—数値計算ソフトウェアへの適用とその可能性，慧文社，163p. (Dec. 2004).
- 2) 弓場敏嗣：数値計算応用を対象とする自動性能チューニング技術，電気通信大学紀要(解説)，Vol.20, 1・2 合併号，pp.1-14 (2007).
- 3) 自動チューニング研究会編：自動チューニング技術に関する課題調査，自動チューニング研究会技術報告，102p. (Nov. 2008).
(平成 21 年 3 月 19 日受付)

弓場 敏嗣 (正会員) yuba@acm.org

1966 年神戸大学大学院工学研究科修士課程修了，(株)野村総合研究所を経て，1967 年通商産業省工業技術院電子技術総合研究所（現在，産業技術総合研究所）入所。1993 年電気通信大学大学院情報システム学研究科教授。2007 年同退職，名誉教授。本会，電子情報通信学会各フェロー。

片桐 孝洋 (正会員) katagiri@cc.u-tokyo.ac.jp

東京大学情報基盤センター特任准教授。平成 8 年京都大学工学部卒業。平成 13 年東京大学大学院理学系研究科情報科学専攻修了。博士(理学)。平成 14 年電気通信大学助手，平成 17 年米国カリフォルニア大学バークレー校訪問学者を経て，平成 19 年より現職。平成 14 年度本会山下記念研究賞受賞。