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

< ソフトウェア自動チューニング技術の応用 >

7 自動チューニングの適用事例：量子化学計算と信号処理

山本 有作　名古屋大学大学院工学研究科

はじめに

科学技術計算における自動チューニングは，数値計算ライブラリへの適用を中心として発展してきた．実際，自動チューニングの成功事例として広く知られている ATLAS，FFTW は，それぞれ行列乗算，高速フーリエ変換のための数値計算ライブラリである．数値計算ライブラリは，他のソフトウェア部品と切り離してその部分だけを最適化することが可能であり，かつ，プログラミングには，数値計算とハイパフォーマンスコンピューティング技術の両方に精通した専門家が携わることが多い．そのため，自動チューニング技術を組み込むことが比較的容易な状況にある．数値計算ライブラリにおける自動チューニングに関しては，その後も活発に研究が進められ，行列計算ライブラリの ILIB，固有値計算のための EigAdept など，さまざまな成果が生まれている．

一方，一般の科学技術計算においては，処理内容が用途によって大きく異なるなど，ライブラリとしての切り出しが必ずしも容易でない応用も数多く存在する．このような応用に対しても自動チューニング技術を適用できれば，実行速度の向上，所要記憶量の削減などの面で大きなメリットが期待できる．

そこで本稿では，数値計算ライブラリ以外の科学技術計算に自動チューニングを適用し，大きな成果を挙げている事例として，量子化学計算のためのシステム TCE（Tensor Contraction Engine）と信号処理のためのシステム SPIRAL を紹介する．

以下では，まず自動チューニングが有効に働くための条件について，数値計算ライブラリの事例を振り返りつつ検討する．次に TCE と SPIRAL を紹介し，両者がなぜ成功を収めたのかを，上記の条件に基づいて考察する．最後に，これら 2 つの事例に基づき，科学技術計算の他の分野における自動チューニングについても言及したい．

自動チューニングが有効に働く条件

自動チューニングは，うまく活用できれば効果が大きいが，一方で，プログラムに自動チューニング機能を組み込むにはコストがかかる．したがって，自動チューニングが有効となるには，得られる効用とコストを比べたとき，前者の方が大きい必要がある．自動チューニングの効用が大きくなるための条件，コストが小さくて済む条件を考えてみると，表 -1 のような条件が挙げられる．

数値計算ライブラリの場合は，条件（E1），（E3），（E4），

効用が大きくなるための条件	コストが小さくて済むための条件
（E1）プログラムの実行時間が非常に長い，あるいは多くの計算機資源を必要とする場合	（C1）計算の大部分が少数の計算カーネルに集中している場合
（E2）プログラムの実行時間，あるいは計算機資源に関する制約が非常にきつい場合	（C2）パラメタを変化させることで，プログラムをさまざまな異なるハードウェア環境向けに最適化することが可能な場合
（E3）パラメタ／アルゴリズムの選択により，性能が大きく変化する場合	（C3）異なるハードウェア環境に適するさまざまなアルゴリズムを系統的に自動生成できる場合
（E4）同じプログラムが，さまざまな異なるハードウェア環境で使われる場合	（C4）プログラムの構造が明快で，ドキュメントがきちんとしている場合
（E5）同じプログラムが長期間にわたり使われる場合	

表 -1　自動チューニングの効用が大きくなるための条件と，コストが小さくて済むための条件

目的関数	チューニングすべきパラメタ
(o1) 縮約の計算量を削減する (o2) 縮約演算の中間結果を記憶する領域を削減し，できればメモリ上に格納できるようにする (o3) 中間結果がメモリ上に載らない場合でも，ディスクには収まるようにする (o4) ディスク I/O の量を最小化する (o5) 分散メモリ型並列計算機上で実行する場合は，プロセッサ間通信を削減する	(p1) ループの順番（ループの実行順序，ループのネスティング順序など） (p2) ループの融合 (p3) ループのタイリング (p4) 分散メモリ型計算機の場合のデータ分散方法

表 -2　多電子積分の自動チューニングにおける目的関数とチューニングすべきパラメタ

(E5) を満たすため，自動チューニングの効用は非常に大きい．さらに，条件 (C1)，(C2) が成り立つため，適用のコストは比較的小さい．他の科学技術計算についても，これらの条件を満たすならば，自動チューニングを活用できる余地があることになる．

量子化学計算への適用事例：TCE

そのような科学技術計算応用の例として，本章では量子化学計算における多電子積分を取り上げる．量子化学は，量子力学に基づいて分子構造，物性，化学反応などを解明する分野であり，そこではシュレーディンガー方程式を数値的に解くことが重要となる．方程式を解くにあたっては，電子の波動関数を基底関数で展開することにより，方程式中に出てくる微分演算子のハミルトニアン (全エネルギー演算子) を行列で表し，方程式を行列の固有値問題として書き直す必要がある．このための計算が多電子積分であり，量子化学計算において実行時間の大部分を占める処理である．本応用は，応用分野の専門家である量子化学者と計算機科学者との協働により，AT 応用の実用化に向けた研究が進められている数少ない例の 1 つである [1]．

■ 多電子積分の処理の概要

多電子積分では，ハミルトニアン行列の要素を求めるため，複数個の電子軌道とポテンシャルの積に対して多重積分を行って A_{ijkl} のような高階テンソルを求める．ここで，高階テンソルとは，多次元の配列と考えてよい．さらに，複数個のテンソルに対して縮約を行う．たとえば，次の式は 4 個の 4 階テンソルを 6 個の添字に関して縮約し，1 個の 4 階テンソルを求める計算を示す．

$$S_{abij} = \Sigma_{cdefkl} A_{acik} \times B_{befl} \times C_{dfjk} \times D_{cdel}$$

これを単純にプログラム化すると，添字が 10 種なので 10 重のループになるが，各添字は数十〜数千個の値を取るため，計算量が膨大となる．この計算を高速かつ使用可能なメモリ／ディスクの範囲で行うことが課題である．

■ 目的関数およびチューニングすべきパラメタ

上記の課題を達成するための目的関数としては，表 -2 の (o1) 〜 (o5) が考えられる．一方，最適化すべきパラメタとしては表 -2 の (p1) 〜 (p4) が挙げられる．この問題は多目的最適化問題となるため，制約条件 (o3) を満たしつつ，各目的関数 (o1)，(o2)，(o4)，(o5) に適当な重みを付けて最適化を行うことになる．

■ 自動チューニングの効用と適用における課題

多電子積分は分子軌道法において計算時間の 95% 以上を占めると言われる負荷の重い計算である (表 -1 の条件 E1)．また，計算法によっては必要とする中間結果の大きさがディスク容量を超えてしまい，計算不可能になるなど，計算機資源に関する制約がきつい計算でもある (E2)．また，計算法，特にループの順番により，計算量と所要メモリ量は数千倍以上の幅で変動する (E3)．これに対する最適化は，従来，経験の豊富な量子化学者が長時間かけて手作業で行っていたが，これを自動化できれば，その意義は大きい．さらに，分子軌道法は広く使われているため，汎用的な AT システムが構築できれば，その恩恵は広い範囲に及ぶ (E4)．以上より，自動チューニングが適用できれば，その効用はきわめて大きい．

一方，多電子積分は，対象とする系あるいは量子化学の手法ごとに計算すべき式が異なる．したがって，あらかじめ決めた何種類かの計算式を最適化し，ライブラリとして提供するだけでは解決にならない．ユーザが定義した任意の式の計算を最適化する仕組みが必要であり，ここに数値計算ライブラリとは違った難しさがある．

最適化を行うに当たっては，ループの順番変更や融合の可能性をシステムが自動的に抽出しなければならない．しかし，計算式が通常のプログラミング言語で記述されていると，複雑なループに対しては解析が困難となり，これら最適化のための情報を抽出できない可能性がある．また，パラメタ数が多く，各パラメタの取り得る値の数が多いため，パラメタの組合せの数が爆発的に増大し，最適パラメタの探索が困難となるという点も課題である．

■ 自動チューニングシステム TCE

これらの問題点を解決し，多電子積分への自動チューニングの適用を可能にした例として，TCE と呼ばれるシステムがある [1)]．このシステムでは，(a) 多電子積分の記述のための高水準言語，(b) パラメタ最適化エンジン，(c) コード生成装置，の 3 つを組み合わせることにより，多電子積分のための自動チューニング支援ツールを構築している．各部分の役割は次の通りである．

◆ 多電子積分の記述のための高水準言語

多電子積分を，なるべく数式(テンソルの縮約)に近い形で記述することを可能にする言語である．計算内容を高水準で記述することにより，通常のプログラミング言語で書いたプログラムからは抽出困難な，ループの順番変更や融合の可能性などに関する情報を抽出することが可能となる．

◆ パラメタ最適化エンジン

高水準言語で書かれたプログラムをもとに，パラメタの最適化を行う．組合せ論的爆発に対処するため，階層的な最適化，動的計画法の利用，メタヒューリスティクスの利用などの工夫を用いる．TCE では，演算量最小化，演算量を保ったままでのメモリ使用量削減，中間結果がディスク容量を超える場合のディスク使用量削減（演算量増加を許容），ディスク I/O の最適化，データ分散方式の決定，のステップをこの順に（必要なら前のステップに戻って）行うという階層的な最適化の枠組みを用いている（図 -1[1)]）．また，各層の最適化では，必要に応じて動的計画法やメタヒューリスティクスを利用する．

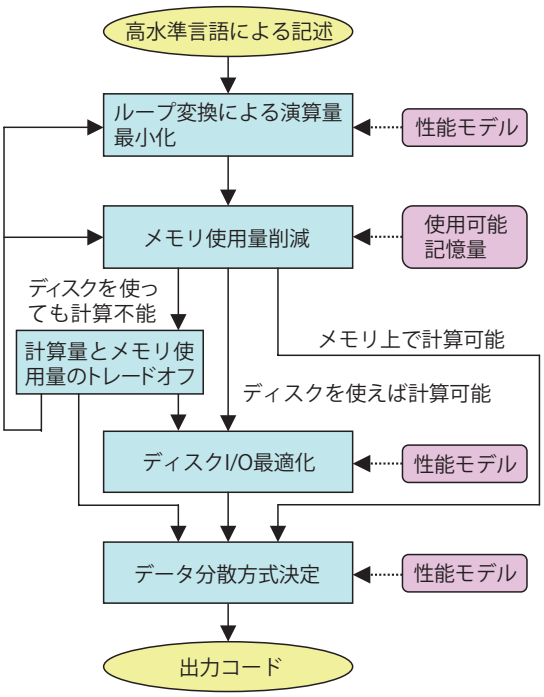

図 -1　TCE におけるパラメタ最適化エンジンの処理フロー

◆ コード生成装置

最適化エンジンで求めたパラメタに基づき，通常のコンパイラに入力可能なプログラムを自動生成する．

■ TCE による自動チューニングの例

TCE による自動チューニングの例として，縮約演算

$$B_{abcd} = \sum_{pqrs} C^{(1)}_{sd} \times C^{(2)}_{rc} \times C^{(3)}_{qb} \times C^{(4)}_{pa} \times A_{pqrs}$$

の高水準言語による記述を図 -2 に，それに最適なループ融合とタイリングを施して得られたコードを図 -3 に示す [1)]．ただしタイリングとは，多重の For ループにおいて，各ループ変数の動く範囲をブロックに分割することにより，内側ループでの配列の参照範囲を狭める最適化手法のことである．この例では，まず，計算量削減のため，縮約演算が次のように 4 段階に分割される．

$$B_{abcd} = \sum_{s} C^{(1)}_{sd} \times (\sum_{r} C^{(2)}_{rc} \times (\sum_{q} C^{(3)}_{qb} \times (\sum_{p} C^{(4)}_{pa} \times A_{pqrs})))$$

```
procedure Transform(in disk A[N,N,N,N], in disk C1[N,V], in disk C2[N,V], in disk C3[N,V],
                    in disk C4[N,V], out disk B[V,V,V,V]) =
  index a,b,c,d : V;    index p,q,r,s : N
begin
  B[a,b,c,d] = sum[C1[s,d] * C2[r,c] * C3[q,b] * C4[p,a] * A[p,q,r,s], {p,q,r,s}];
end
```

図 -2　TCE の高水準言語によるテンソル縮約演算の記述例

```
Read C4,C3,C2,C1
For sT
 For rT
  For qT
   For pT
    Read A
     For a,sI,rI,qI,pI
      T1[a,sI,rI,qI]
          += C4[p,a]*A[pI,qI,rI,sI]
   For a,sI,rI,qI,b
    T2[a,b,sI,rI] += T1[a,sI,rI,qI]*C3[q,b]
  For a,sI,rI,b,c
   T3[a,b,c,sI] += T2[a,b,sI,rI]*C2[r,c]
 Write T3
For aT
 For sT
  Read T3
  For aI,b,c,sI,d
   B[aI,b,c,d] += T3[aI,b,c,sI]*C1[s,d]
 Write B
```

図 -3　TCE による自動最適化で得られたコード

次に，分割後の縮約演算

$$T^{(1)}_{aqrs} = \Sigma_p C^{(4)}_{pa} \times A_{pqrs}$$

について調べる．この演算は 4 次元配列を用いる演算であり，計算に必要なメモリ量を算定してシステムの使用可能メモリ量と比較することで，ディスク I/O が必要と判定される．そこでタイリングを適用し，内側ループでの配列参照範囲を狭めて，内側ループの演算をメモリ上で行えるようにする．タイルサイズは，計算機環境と問題サイズに応じて最適に決定する．さらに，ループの融合を行うことでディスク I/O の量をさらに削減する．

Itanium 2，メモリ 4GB の計算機上での自動チューニングの効果を表 -3 に示す．量子化学計算で標準的に用いられるタイルサイズを用いた計算に比べ，TCE の出力したコードでは，ディスク I/O 時間が約 1/5，全実行時間が約 1/2 と高速化されている．

最適化手法	ディスク I/O 時間	全実行時間
タイルサイズ =1/3×(全メモリ量)$^{1/4}$	1240.85 秒	1957.18 秒
最適タイルサイズ＋ループ融合(TCE)	248.43 秒	954.87 秒

表 -3　TCE による自動チューニングの効果

信号処理への適用事例：SPIRAL

■ 信号処理における線形変換

信号処理においては，入力ベクトル x に対して定数行列 F をかけて出力ベクトル y を求める線形変換がよく用いられる．変換の具体例としては，離散フーリエ変換，ウェーブレット変換，線形フィルタなどがある．

x，y のサイズをそれぞれ m，n とすると，単純な行列ベクトル積と見た場合の計算量は $O(mn)$ である．しかし，多くの線形変換では，F を対角行列，置換行列，サイズ 2 のデータに対する行列の直和などの基本変換行列の積として表せる．基本変換行列による乗算の計算量は一般に $O(\max(m, n))$ なので，F を少数個の基本変換行列の積に分解できれば，それらを順に x にかけることで，変換の計算量は大きく削減できる [2]．与えられた線形変換を行うアルゴリズムは，この分解を決めることで決定される．

この分解の仕方により，メモリアクセスのパターンや計算精度も異なる．そこで，与えられたサイズの線形変換を与えられた計算機上で行うに際して，なるべく演算量が少なく，対象計算機に適したアクセスパターンを持ち，かつ精度の高い分解を見つけることが課題となる．さらに，変換を FPGA 等のハードウェアで実現する場合は，回路規模と消費電力を小さくすることも課題となる．線形変換における目的関数とパラメタの例を表 -4 に示す．

■ 自動チューニングの効用と適用における課題

線形変換も表 -1 の（E1）～（E5）の条件を満たし，自動チューニングの効用がきわめて大きい計算である．また，多電子積分と同様，用途によって計算すべき式が異なるため，ライブラリ化のみでは対応が困難である．

■ 自動チューニングシステム SPIRAL

この線形変換に対して自動チューニングの適用を可能としたシステムとして，SPIRAL[3] がある．SPIRAL は，TCE とよく似た構成をとっており，(a) 線形変換を記述するための高水準言語 SPL，(b) 指定された変換行列のさまざまな分解を系統的に導出し，アルゴリズム／パラメタの候補群を生成するシステム，(c) それらの候補の中から目的に応じて最適なものを選ぶ最適化エンジン，(d) コード生成装置，からなっている．

SPIRAL によって生成されたコードは，多くの場合に人手による最適化コードを上回る性能を達成しており，また，分散メモリ型並列計算機や Cell プロセッサ向けの最適化，FPGA による実装のためのロジック自動生成などの成果が発表されている．また，SPIRAL の Web ペ

(o1) 変換の演算量を削減する (o2) 対象計算機に適したメモリアクセスパターン（連続参照など）をできるだけ用いる (o3) 変換の精度を一定値以上に保つ (o4) 回路規模の最小化（ハードウェア化の場合） (o5) 消費電力の最小化（ハードウェア化の場合）	(p1) 変換の基本変換への分解法 (p2) 各基本変換の実装法（ループの実行順序，ループのネスティング順序，ループ融合など） (p3) ハードウェアによる実現の場合の種々のパラメタ（パイプラインの深さ，並列性など）

表 -4　線形変換の自動チューニングにおける目的関数とチューニングすべきパラメタ

ージ[4]では，ユーザが指定した線形変換に対し，さまざまなプロセッサ向けの最適化コードを自動生成し，提供している．

まとめと今後の展望

各応用分野におけるアルゴリズムを専用の高水準言語で記述し，最適化エンジンによりアルゴリズム／パラメタを最適化してコードを自動生成するというアプローチは，他の科学技術計算にも適用可能であり，条件（E1）～（E5）を満たすような応用では有用だと考えられる．本アプローチのより広い分野への応用が期待される．

参考文献

1) Baumgartner, G. et al. : Synthesis of High-Performance Parallel Programs for a Class of Ab Initio Quantum Chemistry Models, Proceedings of IEEE, Vol.93, No.2, pp. 276-292 (2005).
2) van Loan, C. F. : Computational Frameworks for the Fast Fourier Transform, SIAM (1992).
3) Pueschel, M. et al. : SPIRAL : Code Generation for DSP Transforms, Proceedings of IEEE, Vol.93, No.2, pp.1-42 (2005).
4) http://www.spiral.net/

（平成 21 年 3 月 30 日受付）

山本　有作（正会員）　yamamoto@na.cse.nagoya-u.ac.jp

1966 年生．1992 年東京大学大学院工学系研究科物理工学専攻修士課程修了．同年（株）日立製作所中央研究所入所．2003 年名古屋大学大学院工学研究科助手．現在，同准教授．博士（工学）．