

科学研究費助成事業 研究成果報告書

平成 26 年 6 月 10 日現在

機関番号：14301

研究種目：基盤研究(B)

研究期間：2011～2013

課題番号：23300006

研究課題名(和文) 時空間タイリングによる高性能シミュレーションコードの生成

研究課題名(英文) Composing High-Performance Simulation Codes by Temporal-Spatial Tiling

研究代表者

中島 浩(Nakashima, Hiroshi)

京都大学・学術情報メディアセンター・教授

研究者番号：10243057

交付決定額(研究期間全体)：(直接経費) 14,700,000円、(間接経費) 4,410,000円

研究成果の概要(和文)：本研究で提案した時空間タイリングは、シミュレーションコードの基本構造である時間・空間の多重ループに対し、広大な空間の走査ループをキャッシュメモリに適合可能な小空間(タイル)単位の走査に分割し、かつ空間タイルの時間的更新過程を複数ステップにわたって繰り返すことで、メモリ参照局所性を改善して高速化する技術である。本研究では、局所視点に基づくコード断片記述による時空間タイリング適用を自動化する枠組の開発と、FDTD法電磁場解析、PIC法プラズマシミュレーション、マルチグリッド法ポアソン求解の各応用に対する時空間タイリングの適用実験を行い、各々について目標とした2倍程度の高速化を達成した。

研究成果の概要(英文)：Temporal-Spatial (TS) tiling proposed in this research is an optimization technique to improve the memory access locality in a TS-nested loop, adopted in most of simulation codes as their fundamental structure, by splitting the spatial loop into a set of those for smaller spaces (tiles) fitting to cache memory and by iterating updates on each tile for multiple time steps. In this research, we developed a framework of automatic application of TS tiling based on local-view fragmental coding, and then applied TS tiling to applications including FDTD electromagnetic field analysis, PIC plasma simulation, and multi-grid Poisson solver, to confirm each application is accelerated by factor two as we aimed at.

研究分野：総合領域

科研費の分科・細目：情報学

キーワード：ハイパフォーマンスコンピューティング タイリング コード生成 コード変換

1. 研究開始当初の背景

実用的なシミュレーションコードの多くは、対象とする系の時間的な状態変化を追跡するものであり、一般的には時刻 t における系の状態 $S(t)$ から微小時刻 Δt 後の状態 $S(t+\Delta t)$ を求める計算の繰り返し、すなわち；

```
for t=0 to tend step  $\Delta t$  do
   $S(t+\Delta t) \leftarrow F(S(t));$ 
```

のような形で実現される。またほとんどの場合、状態 $S(t)$ は多数の状態変数の集合、たとえば空間的な系を離散化した格子点・節点に対して定義される変数の集合であり、上記の擬似コードは；

```
for t=0 to tend step  $\Delta t$  do
  for  $\forall s \in S$  do  $s(t+\Delta t) \leftarrow f_s(S(t));$ 
```

というように、時間に関するループ（時間ループ）がその内側に全ての状態変数を走査するループ（空間ループ）を持つ二重ループ（あるいは多重ループ）構造となる。このような時空間ループを、より一般的に運搬依存性を持つ外側ループと多数の変数・配列要素を走査する内側ループからなる構造ととらえると、偏微分方程式の陽的な解法や連立一方程式・固有値などの反復解法などにも見られるなど、高性能計算の分野では頻出するプログラム構造であると言える。

この時空間ループの重大な律速要因は、空間ループが走査するメモリ領域が広大であるため、メモリアクセスの時間的局所性が著しく小さく、高速計算の鍵であるキャッシュの有効利用がほとんど成されないことにある。特に、ある状態変数を更新するために参照すべき状態変数の集合が小さくかつ空間的に局在する問題、すなわち空間分割による並列計算が容易な問題ほど、時間的局所性の欠如が顕著となる。たとえば7点差分解析のように、3次元格子空間の各格子点上の状態変数を隣接6格子点の状態変数により更新する場合、全格子空間（たとえば 1000^3 ）の走査中に各状態変数はわずか7回しか参照されず、時間的局所性をほとんど見出すことができない。

一方、近年のスカラ並列型スーパーコンピュータやそれを支えるマルチコアプロセッサでは、演算性能とメモリ性能の乖離が確実に進行しており、ほとんどの時空間ループの性能はメモリバンド幅によってのみ定まる状況にある。また GPGPU やメニーコアプロセッサにおいても演算性能とメモリ性能の比はマルチコアプロセッサと大差なく、時空間ループの律速要因がメモリバンド幅であることに変わりはない。

2. 研究の目的

本研究では、時空間ループ実行の律速要因である、メモリ参照の時間的局所性の欠如を解決することを目的として行った。具体的には時空間ループを、空間ループを断片化した空間タイルと時間方向の一定回数反復である時間タイルを組み合わせた時空間タイル

を単位として分割し、時空間タイル内ではメモリアクセスの時間的局所性が著しく向上することを利用して高速化する、時空間タイリング技法の確立が目的である。また我々の従来研究成果である並列化技法ライブラリのアプリケーション適用技術を応用し、時間ループ内に多数の関数・手続呼出や複雑な依存関係を持つ一般的なアプリケーションを対象に、時空間タイリングを施したコードの自動変換・自動生成を行うことを目指した。

3. 研究の方法

(1) 時空間タイリングの概要

時空間タイリングは、時間方向のタイルサイズを m 、基準時刻 t からの時間差 $k\Delta t$ に応じて状態変数集合 $S(t+k\Delta t)$ を n 分割した空間タイルを $S_{k,i}$ ($1 \leq i \leq n$) とする；

```
for t=0 to tend step  $m\Delta t$  do
  for i=1 to n do
    for k=1 to m do
      for  $\forall s \in S_{k,i}$  do
         $s(t+k\Delta t) \leftarrow f_s(S(t+(k-1)\Delta t));$ 
```

のように元の時空間二重ループを、外側から順に時間タイル間、空間タイル間、時間タイル内、空間タイル内の四重ループに変換する手法である。 $S_{k,i}$ は図1に模式的に示すように、空間ループの走査順序（左から右）と状態変数間の依存関係（左右の隣接点に依存）により定まり、ほとんどの場合は同じ i に関して大きな共通部分を持つ。この共通部分は元々の状態変数集合に一致させることができ、かつ k のループ中で m 回連続更新されるため、更新時に参照される他の変数も含め、良好な時間的局所性を持つ。また共通部分以外の変数を元々の変数集合に一致させることもしばしば可能であり、一般にタイリングによる空間的オーバーヘッドは微小または皆無である。

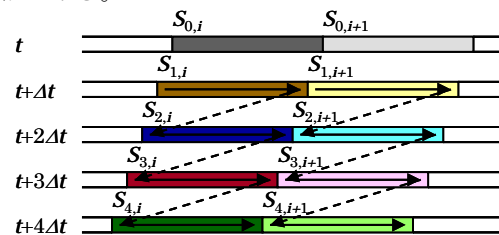


図1: 時空間タイリングの実行過程

(2) 複雑なループのタイリング

上記の模式図のように空間ループが規則的なステンシル計算であって時間ループ内に直接含まれているごく単純な場合であっても、配列で表現される状態変数 $S(t)$ の要素間依存関係を完全に把握し、それに基づく自動的な変換によって時空間タイリングを適用することは容易ではない。また、たとえばプラズマシミュレーションなどで用いられる PIC (Particle-In-Cell) 法では、時空間ループが；

```
for t=0 to tend step  $\Delta t$  do
```

```

begin
  粒子加速(); 粒子移動();
  電流計算(); 電磁場計算();
end

```

のように複数の関数・手続呼出からなり、かつ状態変数である各粒子の位置・速度が別の種類の状態変数である周囲の格子点の電磁場ベクトルと相互作用する複雑な依存関係を持つため、コード解析による自動的な時空間タイリングの適用は実質的に不可能である。

そこで本研究では、空間ループを構成する状態変数更新に関するコード断片と、アプリケーション固有の知識、すなわちデータ構造、制御の流れ、依存関係などの大域情報と、アプリケーションに適合したタイリング手法に基づき、時空間タイリングを施したコードを自動生成することを目指した(図2)。この方法では、空間ループの構造(たとえば粒子集合や3次元格子空間の走査)や意味(たとえば粒子と相互作用する格子点の範囲)がアプリケーション固有の知識として与えられているため、困難な解析を経ずに様々な高速化・最適化技術を適用することができる。

たとえばPIC法のプラズマシミュレーションの場合、前述の「粒子加速()」などのコード断片は概念的には以下のように表現される。

```

粒子加速(){
  p.v[1]+=
    lf(e[se(p.x)],b[sb(p.x)])*
    p.q/p.m; }
粒子位置(){ p.x[1]+=p.v[1]; }
電流計算(){
  j[sj(p.x)]+=cd(p.v)*p.q; }
電磁場計算(){
  e[1]+=(rb(b[1])/mu-j[1])/eps;
  b[1]-=re(e[1]); }

```

このような局所視点に基づくコード断片と大域情報からなるプログラミングを行うためのフレームワークとして、丸山らによるPhysis (Maruyama, et al., SC11)をベース言語として採用し、これに必要な拡張を施すこととした。

一方時空間タイリング技法は一種のライブラリの形で表現され、タイリングを適用した時空間ループのスケルトンなどのアプリケーションに依存しない(擬似)コードと、アプリケーション固有のコード、たとえばPIC法では粒子を空間タイルごとに管理するためのコードなどから構成される。この技法ライブラリと、アプリケーション固有のコード断片・大域情報を融合・結合することにより、タイル化コード生成が行われる。

(3) 自動チューニング

時空間タイリングの有効性は、空間タイル $S_{k,i}$ の大きさ・形状と時間方向のタイルサイズ m に強く依存する。空間タイルの大きさは、基本的にプロセッサのキャッシュサイズに

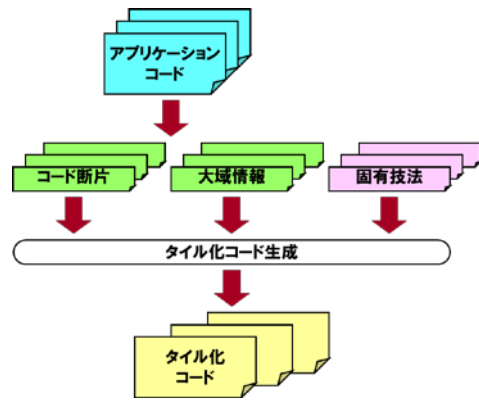


図2: タイル化コードの生成方式

より定まるが、たとえば3次元問題での x, y, z 方向それぞれの最適な大きさは配列データのメモリ上でのレイアウト、多重ループの制御コスト、タイル内の計算を分担するスレッド数、キャッシュの構造・性能やメモリバンド幅など、アプリケーションやプロセッサアーキテクチャの性質に強く依存する。また時間方向のタイルサイズも、プロセス並列実行の際のプロセス間通信のバンド幅や遅延に左右される。

したがって、最適なタイリングパラメータを解析的に求めることは極めて困難であり、何らかのチューニングが必要となる。そこで我々が開発した並列スクリプト言語 Xcrypt (Hiraishi, et al., WHIST2012)を用いて、広大なパラメータ空間を効率的に並列探索する枠組みを構築することとした。

4. 研究成果

本研究で対象とした主なアプリケーションである、FDTD法による3次元電磁場解析、PIC法によるプラズマシミュレーション、およびマルチグリッド法によるポアソン方程式ソルバについて、以下の成果を得た。

(1) FDTD法

本研究に先行して行ったFDTD法の時空間タイリングでは、時間方向の依存性を解決するためにタイル周辺部分の計算を冗長に実施していた(南ほか、情処論ACS, 4(2))。本研究ではまず、図1に模式的に示したように、時間方向の計算の進行とともにタイルを空間的にずらす手法を見出し、冗長計算を完全に排除することに成功した。その結果、4コアプロセッサを用いたスレッド並列実行では、タイリング適用による性能向上を、従来の1.2倍を大きく上回る2.1倍とすることができた(雑誌論文[2])。

また上記の研究では空間タイルの形状を立方体に限定してタイルサイズと性能の関係性を調べたが、メモリアクセスが連続する空間軸やスレッド分割の対象となる空間軸が存在することから、空間的に非対称の直方体形状が優れていることも示唆された。また直方体の形状を様々に変化させて性能との関

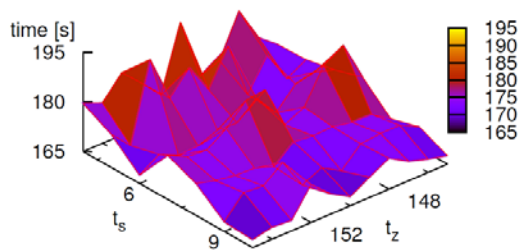


図 3: タイルサイズと性能の関係

係を調査した結果、図 3 に示すように非常に不規則な関係性を有していることが確認された。そこで前述のように Xcrypt を用いて、事前知識に基づくパラメータ空間の制限、モンテカルロ法による粗い探索、および準最適解の周辺のパラメータスイープを組み合わせた、オフライン自動チューニングの枠組を開発して、最適パラメータの並列探索を行った。その結果、8 コアプロセッサを用いたスレッド並列実行では、立方体タイルでは 40^3 の空間タイルと $m=17$ の時間タイルが最適で 2.0 倍の加速効果となるのに対し、 $320 \times 10 \times 24$ の直方体タイルと $m=15$ が最適解として得られ、加速効果を 2.3 倍に向上させることができた (学会発表[3,5])。

また成果の発表には至っていないが、タイル内の計算のスレッド並列化のための 2 次元あるいは 3 次元分割、タイルを単位としたスレッド並列化、プロセス並列化の際のプロセス間通信の遅延隠蔽法などについても検討を行った。

(2) PIC 法

まず、PIC 法の空間ループに対する局所視点に基づくコード断片の記述方法について、その要件と有効性を明らかにした (学会発表[10,11])。続いて、空間全体を走査する複数のループを 1 個のループに統合するループ融合によって、宇宙プラズマコードでは 1.2 倍、また核融合プラズマコードでは 2.0 倍の性能向上が得られることを明らかにした。しかし宇宙プラズマコードを詳細に解析した結果、ループ融合によって反復あたりの演算・操作の数が増加し、かつ条件分岐などコンパイラによる最適化を阻害する要因も増加するため、これらが参照局所性の改善による性能向上を抑えることが明らかになった。そこで単純にループ融合を行うのではなく、個々の操作に対してタイリングを施した上でタイルへの操作を連結する変換が必要であることと、ローレンツ力や粒子移動に伴う電流成分の計算など元々のコードでも反復あたりの演算・操作の数が多いものに対しては、ループ分割を施した結果にタイリングを適用する必要があることが明らかになった。

また本研究に先行する研究成果である PIC 法の動的負荷分散ライブラリ OhHelp (Nakashima, et al., ICS2009) を用いたプロセス並列化では、プロセス単位で時間方向のタイリングを行い、プロセス間の粒子移送や

負荷均衡操作の起動頻度を減らすことが効果的であることが明らかになった。この時間的タイリングの結果、1.8 倍の性能向上が得られることを明らかにした (雑誌論文[1,3]、学会発表[2])。

(3) マルチグリッド法ポアソンソルバ

マルチグリッド法によるポアソン方程式の求解操作は、一種の時間方向ループである V サイクルの反復に大域的な逐次過程が含まれているため、空間タイルに局所的な時間発展操作を施すことはできない。しかし PIC 法のタイリングと同じように、空間的な状態変数に対する一連の操作を時空間タイリングの概念を援用してタイリングすることは有効であり、これにより 4 個の 4 コアプロセッサを有する SMP ノードでのスレッド並列実行で、1.2 倍の加速効果が得られることを見出した。

さらに、空間タイルに対して局所的な反復スムージングを行う乗法シュワルツスムージングを行うことで、ソルバの収束性を向上させつつ、本来的な時空間タイリングを適用可能な問題とできることを見出した。この新たなスムージング方式により、前述の SMP ノードでは操作間のタイリングとの組み合わせで 1.6 倍の加速効果を、また 2 個の 8 コアプロセッサからなるノードを 14 個用いた 216 プロセス並列実行では 1.5 倍の加速効果を、それぞれ得ることができた (雑誌論文[4]、学会発表[7,9])。また、これも PIC 法のタイリングで述べたループ分割とタイリングの組み合わせが、本質的に逐次過程であるガウスザイデル法スムージングに対する SIMD 演算の部分的活用にも有効であり、メニーコアプロセッサ Xeon Phi での 240 スレッド並列実行で 1.3 倍の加速効果が得られることも見出した (学会発表[1,4])。

5. 主な発表論文等

[雑誌論文] (計 6 件)

- [1] Yohei Miyake, Chris M. Cully, Hideyuki Usui, Hiroshi Nakashima. Plasma Particle Simulations of Wake Formation behind a Spacecraft with Thin Wire Booms. *J. Geophys. Res.*, 査読有, Vol. 118, No. 9, 2013, pp. 5681-5694, doi: 10.1002/jgra.50543.
- [2] 南武志, 岩下武史, 中島浩. 冗長な計算を伴わない 3 次元 FDTD 法の時空間タイリング. *情報処理学会論文誌: コンピューティングシステム*, 査読有, Vol. 6, No. 1, 2013, pp. 56-65, <http://id.nii.ac.jp/1001/00089920/>.
- [3] Yohei Miyake, Hideyuki Usui, Hirotsugu Kojima, and Hiroshi Nakashima. Plasma Particle Simulations on Stray Photoelectron Current Flows Around a Spacecraft. *J. Geophys. Res.*, 査読有, Vol. 117, No. A09210, 2012, pp. 1-13,

- doi: 10.1029/2012JA017673.
- [4] 河合直聡, 岩下武史, 中島浩. ブロック化赤-黒順序付け法による並列マルチグリッドポアソンソルバ. 情報処理学会論文誌: コンピューティングシステム, 査読有, Vol. 5, No. 3, 2012, pp. 1-10, <http://id.nii.ac.jp/1001/00082469/>.
- [5] Yasuhito Takahashi, Takeshi Iwashita, Hiroshi Nakashima, Tadashi Tokumasu, Masafumi Fujita, Shinj Wakao, Koji Fujiwara, Yoshiyuki Ishihara. Parallel Time-Periodic Finite-Element Method for Steady-State Analysis of Rotating Machines. IEEE Trans. Magn., 査読有, Vol. 48, No. 2, 2012, pp. 1019-1022, doi: 10.1109/TMAG.2011.2171923.
- [6] Yasuhito Takahashi, Takeshi Iwashita, Hiroshi Nakashima, Shinj Wakao, Koji Fujiwara, Yoshiyuki Ishihara. Micro-magnetic Simulations of Perpendicular Recording Head Using the Parallel Fast Multipole Method Specialized for Uniform Brick Elements. IEEE Trans. Magn., 査読有, Vol. 47, No. 10, 2011, pp. 3805-3808, doi: 10.1109/TMAG.2011.2154305.
- [学会発表] (計 1 1 件)
- [1] Masatoshi Kawai, Takeshi Iwashita, Hiroshi Nakashima. SIMD Implementation of a Multiplicative Schwarz Smoother for a Multigrid Poisson Solver on an Intel Xeon Phi Coprocessor. Intl. Mtng. High-Performance Computing for Computational Science, 2014 年 7 月 2 日, Eugene, OR, USA.
- [2] Yohei Miyake, Hiroshi Nakashima. Low-Cost Load Balancing for Parallel Particle-in-Cell Simulations with Thick Overlapping Layers, Intl. Symp. Parallel and Distributed Processing with Applications, 2013 年 7 月 18 日, Melbourne, Australia.
- [3] Takeshi Minami, Motoharu Hibino, Tasuku Hiraishi, Takeshi Iwashita, Hiroshi Nakashima. Performance Improvement of Three-Dimensional Tiled FDTD Kernel Based on Automatic Parameter Tuning, Intl. Conf. Computation on Electromagnetic Fields, 2013 年 7 月 3 日, Budapest, Hungary.
- [4] Takeshi Iwashita, Akihiro Ida, Masatoshi Kawai, Hiroshi Nakashima. Performance Evaluation of Multi-threaded Iterative Solver on Recent Processors. 2013 SIAM Conf. Computational Science and Engineering, 2013 年 2 月 18 日, Boston, MA, USA.
- [5] 日比野元春, 南武志, 平石拓, 岩下武史, 中島浩. Xcrypt を用いた 3 次元 FDTD 法プログラムの自動チューニング. 日本応用数学会 2012 年度年会, 2012 年 8 月 31 日, 稚内市.
- [6] Yohei Miyake, Hiroshi Nakashima, Hideyuki Usui. Development of a Scalable PIC Simulator for Spacecraft-Plasma Interaction Problems. Intl. Mtng. High-Performance Computing for Computational Science, 2012 年 7 月 19 日, Kobe, Japan.
- [7] Masatoshi Kawai, Takeshi Iwashita, Hiroshi Nakashima, Osni Marques. Parallel Smoother Based on Block Red-Black Ordering for Multigrid Poisson Solver. Intl. Mtng. High-Performance Computing for Computational Science, 2012 年 7 月 17 日, Kobe, Japan.
- [8] Takeshi Iwashita, Yasuhito Takahashi, Hiroshi Nakashima. Algebraic Block Multi-Color Ordering Method for Parallel Multi-Threaded Sparse Triangular Solver in ICCG Method. Intl. Parallel and Distributed Processing Symp., 2012 年 5 月 22 日, Shanghai, China.
- [9] 河合直聡, 岩下武史, 中島浩. ブロック化赤-黒順序付け法による並列マルチグリッドポアソンソルバ. ハイパフォーマンスコピューティングと計算科学シンポジウム, 2012 年 1 月 26 日, 名古屋市.
- [10] Hiroshi Nakashima. Local View Kernels: A New Programming Scheme for Plasma Simulation. Plasma Conf., 2011 年 11 月 23 日, Kanazawa, Japan.
- [11] Hiroshi Nakashima. Generator for Library and Application: Splitting What & How by Domain-Specific Local-View Programming. Intl. Cong. Industrial and Applied Mathematics, 2011 年 7 月 22 日, Vancouver, Canada.
- [その他]
ホームページ等
- [1] OhHelp ライブラリ公開ページ,
<http://www.para.media.kyoto-u.ac.jp/ohhelp/>
6. 研究組織
- (1) 研究代表者
中島 浩 (NAKASHIMA HIROSHI)
京都大学・学術情報メディアセンター・教授
研究者番号: 10243057
- (2) 研究分担者
岩下 武史 (IWASHITA TAKESHI)
京都大学・学術情報メディアセンター・准

教授

研究者番号：30324685

(3)連携研究者

平石 拓 (HIRAISHI TASUKU)

京都大学・学術情報メディアセンター・助教

研究者番号：60528222

大村 善治 (OMUA YOSHIHARU)

京都大学・生存圏研究所・教授

研究者番号：50177002

村田 澄彦 (MURATA SUMIHIKO)

京都大学・大学院工学研究科・准教授

研究者番号：30273478