

共同利用(産業利用トライアルユース:先端研究施設共用促進事業『みんなのスパコン』TSUBAMEによるペタスケールへの飛翔) 成果報告書 平成21年度 新規利用拡大 i09nc

利用課題名 新概念による大規模並列電磁界解析技術研究  
英文:A study of new large scale parallel electromagnetic simulation concept

利用課題責任者  
鈴木 誠  
Makoto Suzuki

所属  
ソニー株式会社

電子機器の高速高集積化に伴い、設計の難易度が飛躍的に高まっている。そのため、複雑な現象を定量的に把握するために数値電磁界解析が注目され、様々な研究と実用化が進められている。近年、並列計算環境の普及が進んでおり、大規模な数値電磁界解析にも応用が期待される。そこで、大規模電磁界解析の実現可能性を検討するため、複数 CPU/GPU を用いた並列差分解析プログラムを試作し、課題の把握およびその解決方法を検討した。今回の検討では、未知数( $E/H$ )の更新手法やデータ転送(デバイス内、デバイス間、ネットワーク間)のボトルネックの確認およびその解消策を試行することで数十億メッシュ規模の解析が実現できることを確認できた。

Fast and memory efficient parallel implementation of finite difference time domain electromagnetic scheme is prototyped on multiple CPU/GPU's. The key to the high performance of parallel executions is how to decrease the number of load and save operations between memory units and CPU/GPU's for effective use of limited memory bandwidth. Our implementation utilizes a coefficient hash table and optimal ordering of  $E$  and  $H$  field updates on a GPU Device, which co-simulate with a CPU's circuit simulation. Experimental results show the efficiency of our implementation for a benchmark model of 1 billion meshes.

*Keywords:* GPGPU, Finite Difference Time Domain, Hash Table, CPU/GPU Co-Simulation

## 背景と目的

電子機器の高速高集積化に伴い、設計の難易度が飛躍的に高まっている。特に動作速度の高速化に伴い、副次的に発生する環境電磁ノイズが問題となっている。そういった背景から、数値電磁界解析が注目され、様々な研究と実用化が進められている。しかし、解析対象が高密度になり複雑なモデリングが必要になるにつれ、解くべき支配方程式の規模が莫大なものとなり、そのための解析処理時間が問題となっている。

これに対し、近年並列処理による高速計算環境の普及が進んでおり、大規模な数値電磁界解析にも応用が期待されている。特に、異なる処理を並列して行うことが可能な CPU の並列化と、同一の処理を多数のスレッドで処理することが有効な GPU を利用することで、実

設計で求められる複雑で大規模な数値電磁界解析を実現できると期待される。そこで本プロジェクトでは、現実的な大規模電磁界解析の実現可能性を検討するため、複数 CPU/GPU を用いた数値解析プログラムを試作し、有効性の検討と問題点の把握およびその解決方法を検討した。検討に際し、電子機器全体を解析できるような大規模数値電磁界解析システムのプロトタイプを作成し、最終的に数十億メッシュ程度の解析対象物を数時間で計算できることを確認した。

## 概要

大規模電子機器における環境電磁ノイズの発生、伝導、放射現象を解析し、最適なノイズ低減設計を実現するために、複雑な電子機器を簡略化することなく詳細に数値モデル化することで環境電磁ノイズの発生メカ

ニズムを CPU/GPU 連成高速計算手法により現象を把握して可視化し、電子機器からのノイズを効果的に低減させる最適設計を数値シミュレーションにより行う。

電子機器を解析するための手法としては様々なものが提案され、応用されている。その中でも差分方程式 [1][2] のように本質的に並列処理に適合したアプリケーションでは GPU を活用することでこれまでに無いパフォーマンスを得ることが期待できる [3]。そこで有限積分時間領域アルゴリズムを例題とし、GPU による並列処理プログラムを実装した。さらに、大規模解析対象に対応するため、ネットワーク並列 (MPI) を用いたハイブリッド並列化を行い、パフォーマンス向上を検討した。また、単純並列化にあまり適さないと考えられる処理系 (回路シミュレーションなどの行列処理を伴う陰的アルゴリズム) の連成解析を実現するために、CPU と GPU とを協調させ、Co-Simulation の検討を行った。

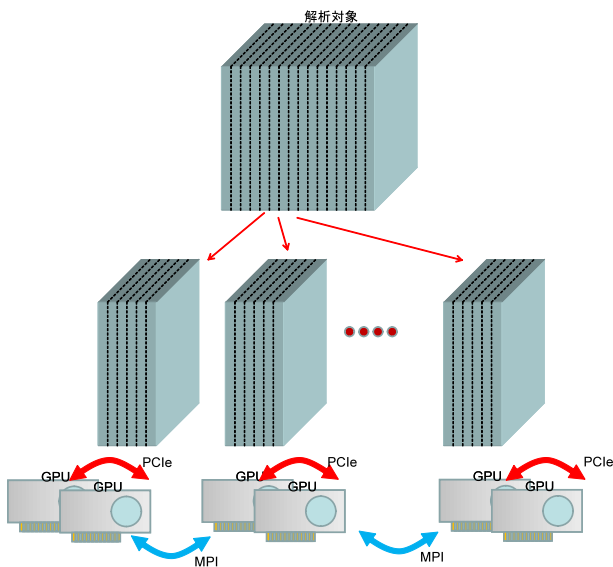


図 1 解析領域分割

GPU による並列化効率を最大限に生かすため、解析領域を図1に示す通り一方向にスライスし、それぞれのホストマシンに割り当てる。各ホストマシンには複数 GPU が装着されているため、各ホストマシン内でも同様にスライスし、解析領域を各 GPU に割り当てる。複数の GPU 間でデータ同期を取りながら計算を行うために、領域分割された境界 YZ 面を同一ホストマシン上の GPU 間では GPU1⇒Host⇒GPU2 という手順でデータ転送を行う。一方で異なるホスト上の GPU 間では、

GPU1⇒Host1⇒ネットワーク⇒Host2⇒GPU2 という経路で転送を行う。

GPU での並列計算は、多数のスレッドにて同時処理される。これらのスレッドはひとまとまりのグリッドを単位として実行され、複数のブロックで構成される。この処理を差分計算に適用するため、図2に示すとおり、解析領域を直交メッシュにより分割する。XYZ 軸に対し、X 軸方向にスライスした各 YZ 平面をひとつのブロックとして GPU で処理する。Z 軸方向にスレッドを配置し、各スレッドは Y 軸方向の処理を担当する。デバイスからレジスタとのメモリ転送を高速化するために、各状態変数はアラインメントされた形でメモリ上に確保する必要がある。電界変数を例とすると  $E_x[i][j][k]$  は  $k$  が連続する形でメモリ上にアラインメントして保存し、コアレスアクセスを実現する。

一連の手順として、GPU 上のグローバルメモリから実行のためにレジスタ内にデータを読み込み (load)、計算し、結果を保存 (store) する。近年の並列処理系では計算パフォーマンスが非常に高いため、load/store に要するメモリバンド幅がボトルネックとなる。このメモリウォールを低減するために、電磁界方程式を差分化した際の各メッシュ係数をハッシュテーブル化して保持し、メモリ使用量を削減してデータ転送量を減らすことを検討し、実装した。ハッシュテーブルには異なる種類の係数だけを分類し、short int 型でハッシュキーを保存する仕様となっている。これにより、本来保持すべき float の係数複数個をひとつの short int に置き換えることが可能となり、半分以下のメモリ使用量及び転送量を実現した。

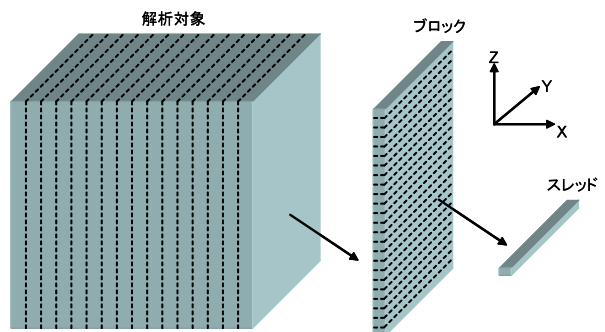


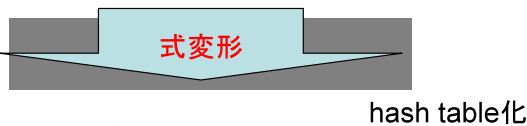
図 2 スレッドおよびブロック

これは、本トライアルで解析対象の特殊性に着目して

(様式第 20)

いることに基づいている。通常の電磁界解析では多くのメッシュはサイズが異なるだけの同一媒質の要素で占められている。従って、係数をあらかじめ計算する際に、同じ値が数多く発生する。そこで、必要となる係数が少なくなるように式変形を行った後、係数部分をハッシュテーブルを用いて圧縮することでメモリ使用量およびデータ転送量を削減することでメモリウォール問題を回避し、高パフォーマンスを得ることが可能となる(図 3)。

$$E_x^n(i, j, k) = \frac{1 - \frac{\sigma\Delta t}{2\varepsilon}}{1 + \frac{\sigma\Delta t}{2\varepsilon}} E_x^{n-1}(i, j, k) + \frac{\Delta t}{\Delta z \left(1 + \frac{\sigma\Delta t}{2\varepsilon}\right)} \left[ H_y^{n-\frac{1}{2}}(i, j, k-1) - H_y^{n-\frac{1}{2}}(i, j, k) \right] + \frac{\Delta t}{\Delta y \left(1 + \frac{\sigma\Delta t}{2\varepsilon}\right)} \left[ H_z^{n-\frac{1}{2}}(i, j, k) - H_z^{n-\frac{1}{2}}(i, j-1, k) \right]$$



$$\Delta x E_x^n(i, j, k) = \frac{1 - \frac{\sigma\Delta t}{2\varepsilon}}{1 + \frac{\sigma\Delta t}{2\varepsilon}} \Delta x E_x^{n-1}(i, j, k) + \frac{\Delta x \Delta t}{\Delta y \Delta z \left(1 + \frac{\sigma\Delta t}{2\varepsilon}\right)} \left[ \Delta y H_y^{n-\frac{1}{2}}(i, j, k-1) - \Delta y H_y^{n-\frac{1}{2}}(i, j, k) + \Delta z H_z^{n-\frac{1}{2}}(i, j, k) - \Delta z H_z^{n-\frac{1}{2}}(i, j-1, k) \right]$$

図 3 式変形およびハッシュ化

これら係数計算は事前に CPU 側で計算してハッシュテーブルに納めてから時間発展計算前に GPU 側にコピーする。

実際の電子機器には、単純に要素分割するだけでは表現できないデバイス等が実装され、特性を決めている。しかしこれらのデバイスに関しては GPU 上で有効な同一オペレーションによる並列実行が適さないため、この部分の処理は CPU 側で並列して実行し、1ステップ解析終了時に CPU/GPU 間で同期を取り、情報の交換を行う(図 4)。

GPU プログラムの実装では、メモリ転送を効率的に

行うことが重要であり、その際にコアレスアクセスを行う必要がある。三次元差分方程式では、各変数は空間的に前後左右上下のメッシュの値を参照し、計算を行う(図 5)。これらの変数は前述のようにアラインメントしてグローバルメモリに保持しているため、前後左右方向(X,Y 方向)へのアクセスはコアレスアクセスにならない。そこで

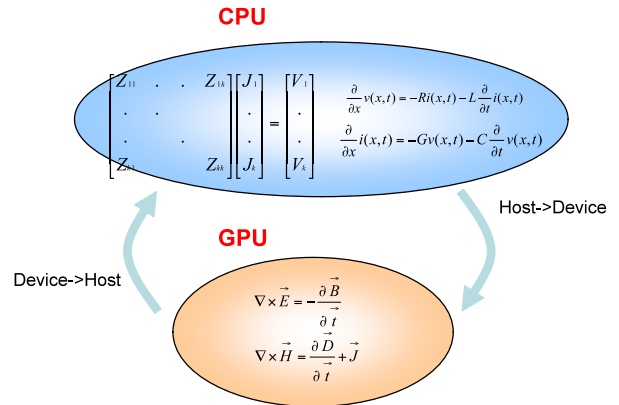


図 4 CPU/GPU 連成解析

Z 方向のデータの転送を行い、shared memory に格納して同一ブロック内の他のスレッドが、上下左右のデータに高速にアクセスすることを可能とした。

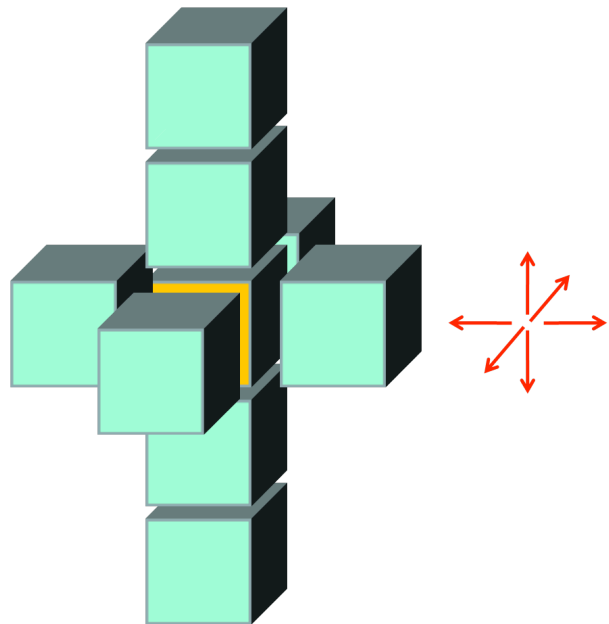


図 5 データアクセス

### 結果および考察

今回の実装では、CPU 側と GPU 側双方に解析変数を保持し、指定された数百ステップ程度の間隔で GPU⇒CPU 全変数コピーを行い、データ同期を取っている。これは、実際の解析では過渡解析の途

(様式第 20)

中の情報を取得、保存する必要があるためである。メモリ使用量及び転送量を削減することにより解析規模はメモリ 1GB 当たり 1000~1500 万メッシュ程度を扱うことが可能となった。これにより s1070 一台あたり 1 億メッシュを処理することが可能であることを確認した。解析に要する時間は、時間刻み  $\Delta t$  に依存する[1][2]ため解析対象により変動するが、弊社の従来プログラムに対して数倍~十倍程度の速度向上が確認できた(吸収境界条件を除く)。

#### **まとめ、今後の課題**

今回、複数 GPU を用いた CPU/GPU 連成解析のプロトタイプを実現した。GPU を用いることで、従来の CPU 単独でのパフォーマンスに比べ、数倍~十倍程度の速度向上が得られることが確認できた。その際、GPU 内のデバイス⇄レジスタ間データ転送、GPU 間データ転送がパフォーマンスに大きく影響することを確認した。

今後、PML(Perfectly Matched Layers)等の吸収境界条件を効率的に実装するための実装方法の検討や、非線形および分散性媒質等、単純な並列化が困難なアプリケーションに対して CPU/GPU 連成による高パフォーマンスの実現を検討する。

#### REFERENCES

- [1] 宇野 亨 「FDTD 法による電磁界およびアンテナ解析」 コロナ社
- [2] A. Taflove and S.C. Hagness, 「Computational Electrodynamics」 Third Edition, Artech House
- [3] 青木 尊之、額田 彰 「はじめての CUDA プログラミング」 工学社