

Cell BEによる粒子法の並列化

赤木正慶^{†1} 今村俊幸^{†1}

粒子ベースシミュレーションの一つである SPH(Smoothed Particle Hydrodynamics) 法は、気体や液体だけでなく剛体の表現に関する研究もなされている手法である。しかし、大規模なシミュレーションでは粒子の相互作用の計算に膨大な時間がかかる。Cell Broadband Engine(Cell BE) プロセッサはマルチメディアデータに対する高速な演算を目的として設計されたプロセッサである。その高い演算性能は科学技術計算への利用も期待されている。本論文では Cell BE プロセッサで動作する SPH 法プログラムを作成し、プログラムの多くを占める近傍探索処理の並列化を行った。その結果、6SPE で最大で約 2 倍の速度向上を示した。また、Cell BE プロセッサ上での倍精度の除算を含む計算の高速化が困難である点を示した。

Parallelization of a particle method on a Cell BE

MASAYOSHI AKAGI^{†1} and TOSHIYUKI IMAMURA^{†1}

SPH(Smoothed Particle Hydrodynamics) is one of particle methods to investigate not only a gas and liquid but a solid model numerically. It requires huge cost in a calculation of all the interaction among particles in a large scale simulation. The Cell Broadband Engine(Cell BE) is designed for high-performance processing of multimedia data. We implement an SPH program and parallelize it on a Cell Broadband Engine(Cell BE). The results show that the program performs about 2 times as fast with 6SPEs. We confirm that speed-up of the calculation including the division of double precision on the processor was difficult.

1. はじめに

数値流体シミュレーションの手法の一つである SPH 法は、天体のシミュレーションの手法として開発された¹⁾が、その後の研究により流体だけでなく剛体の表現も可能になり、広い範囲で用いることが出来る手法である。粒子ベースシミュレーションは、大規模なシミュレーションでは粒子の相互作用の計算に膨大な時間がかかる。これを解決するためにハードウェアの能力を利用するというアプローチがある。その一つとして、GRAPE のような多体問題専用の計算機が用いられる。原田らはグラフィック処理用の演算ユニットである GPU を用いて粒子ベースシミュレーションプログラムを並列化し、流体と剛体の相互作用を計算する手法を提案した²⁾。Hamada らは GPU を用いて、少ないメモリでも多体問題を高速に計算できることを示し、重力シミュレーションライブラリである“CUNBODY-1”を開発した³⁾。

Cell BE プロセッサは、マルチメディアやゲームの高速な演算処理を目的として設計されたプロセッサであり、その高い演算能力は科学計算の分野においても期待されている。Cell BE プロセッサは C 及び C++ によるプログラミングをサポートしており、C のライブラリとして API が提供されているため、他環境からのスムーズなプログラムの移行が可能である。また、家庭用ゲーム機である Play Station 3 にも搭載されており比較的入手が容易である。本論文では粒子ベースシミュレーションの一つである SPH 法を、Cell BE プロセッサを搭載した Play Station 3 を用いて実装し、並列化による性能向上を目指す。

2. Cell BE プロセッサの概要

Cell BE プロセッサは、汎用プロセッサコアである PowerPC Processor Element(PPE) と 8 つの Synergistic Processor Element(SPE) を Element Interconnect Bus(EIB) で接続した、非対称なマルチコア・プロセッサである⁴⁾。図 1 に Cell BE プロセッサの構成を示す。

2.1 PPE

PPE は 64bit の PowerPC Processor Unit(PPU)

^{†1} 電気通信大学大学院電気通信学研究科情報工学専攻
Department of Computer Science, Graduate School of
Electro-Communications, The University of Electro-
Communications

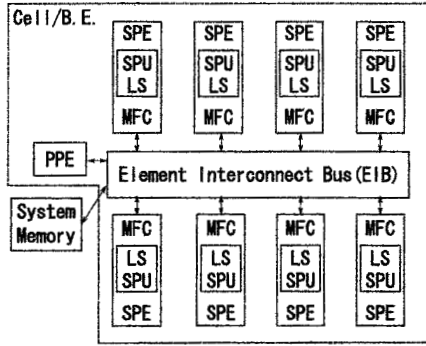


図 1 Cell BE プロセッサの構成図

にキャッシュを加えたものであり、汎用の処理ユニットとして用いられる。PPE の主な仕事はプログラムの流れの管理と SPE へのタスクの割り当てである。

2.2 SPE

SPE は Synergistic Processor Unit (SPU) と Memory Flow Controller (MFC) で構成される。SPU は SIMD 演算能力を持った新設計プロセッサであり、高密度な演算を行うことを目的としている。SPU は 128bit のレジスタを 128 本持ち、128bit の SIMD 演算オペランドは、4 個の 32bit データ、2 個の 64bit データにまとめられてレジスタに格納される。

SPU はプログラムとデータを格納するための Local Storage (LS) と呼ばれる 256KB のメモリを持つ。システムメモリとのデータ転送には MFC の Direct Memory Access (DMA) 転送が用いられる。DMA 転送は MFC によって、演算と非同期に行われるため、ダブルバッファリングを行うことで転送時間を隠すことができる。DMA 転送が可能なデータのサイズは、1, 2, 4, 8 バイト及び 16 の倍数バイトで、最大で 16KB までである。データサイズが 128 の倍数バイトで、LS 及びシステムメモリでデータが 128 バイト境界にアラインされている時にベストパフォーマンスを発揮する。このため、DMA 転送を行う必要のあるデータのアラインとサイズには注意を払う必要がある。

2.3 コミュニケーション

PPE と SPE の間でのコミュニケーションには、メールボックス機構とシグナル通知機構が利用できる。メールボックス機構は、SPU と他のプロセッサやデバイスとの間で 32bit の短いデータを送受信することを意図したものであり、メールボックス・キューへの書き込み・読み出しによって行われる。

シグナル通知機構はプログラム制御フラグなどのシグナルを他のプロセッサに送り、同期を取る時に用い

られる機能である。1 つの SPU は 2 つのシグナル通知レジスタを備える。シグナル通知機構には 1 対 1 のシグナル通知環境で便利な上書きモードと、多対 1 のシグナル通知環境で便利な OR モードのどちらかを選ぶことが出来る。

本論文では、主にメールボックス機構を使って PPE から SPE へのタスクの割り当てを行った。

2.4 実行環境

本論文では実行環境として Cell BE プロセッサを搭載している PlayStation 3 を用いた。Play Station 3 に搭載されている Cell BE プロセッサの SPE は 8 つのうち 1 つは歩留まり向上のため、1 つはシステムが利用するために、ユーザーが利用できる SPE の数は 6 つとなっている。ソフトウェア実行環境は、kernel 2.6.21, gcc 4.1.4, Cell SDK 2.1 を利用した。

3. SPH 法の概要

SPH 法は粒子法の一つで、流体を有限個の代表粒子で近似し、粒子を計算点として流体の数値的計算を行う手法である。計算点は離散的になるが、各粒子に物理量の空間的広がりを持たせる分布情報を与え、この広がりを足し合わせることで流体を表現する。

3.1 物理量の表現

SPH 法では空間に粒子を配置し、それぞれの粒子に物理量の分布を持たせ、それらを重ね合わせることで流体を表現する。物理量の分布は、粒子の持つ物理量とカーネルと呼ばれる関数の掛け算で表現される。粒子 i の座標 \mathbf{x}_i での物理量の近似 $\langle f(\mathbf{x}_i) \rangle$ は、他の粒子の持つ物理量分布との足し合わせを用いて式 (1) のように表される。

$$\langle f(\mathbf{x}_i) \rangle = \sum_{j=1}^n \frac{m_j}{\rho_j} f(\mathbf{x}_j) W_{ij} \quad (1)$$

$$W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h) \quad (2)$$

ここで、 n は粒子数、 m は質量、 ρ は密度、 $W(\mathbf{x}_i - \mathbf{x}_j, h)$ はカーネル関数である。 h はカーネルの大きさを表すパラメータで、平滑化距離 (smoothing length) などと呼ばれる。また、物理量の勾配はカーネルの勾配の足し合わせを用いて、式 (3) のように表される。

$$\langle \nabla \cdot f(\mathbf{x}_i) \rangle = - \sum_{j=1}^n \frac{m_j}{\rho_j} f(\mathbf{x}_j) \cdot \nabla W_{ij} \quad (3)$$

$$\nabla W_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}} = \frac{\mathbf{x}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}} \quad (4)$$

ただし $r_{ij} = |\mathbf{x}_i - \mathbf{x}_j|$ 、 $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ である。

3.2 カーネル関数

カーネル関数は原点でピークをとる、つりがね型の距離関数であり、平滑化距離 h を越えるとゼロとなるような特徴を持つものである。このような特徴といくつかの条件を満たすような関数であれば、様々な関数をカーネル関数として用いることができる。本論文では、3次スプライン関数を用いた。距離 R を用いて、3次のスプライン関数 (cubic spline, B-spline) は区分的に

$$f(R) = \alpha_d \times \begin{cases} \frac{2}{3} - R^2 + \frac{1}{2}R^3 & 0 \leq R < 1 \\ \frac{1}{6}(2-R)^3 & 1 \leq R < 2 \\ 0 & 2 \leq R \end{cases} \quad (5)$$

と定義される。 α_d は、1,2,3次元問題それぞれで、 $1/h$ 、 $15/7\pi h^2$ 、 $3/2\pi h^3$ である。

3.3 支配方程式

SPH 法では連続の式、運動方程式、エネルギー方程式を粒子を用いて近似して用いる。密度計算には、式 (1) をそのまま適用した式 (6) を用いる。

$$\rho_i = \sum_{j=1}^n m_j W_{ij} \quad (6)$$

$$\frac{D\mathbf{v}_i^\alpha}{Dt} = \sum_{j=1}^n m_j \left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_j^{\alpha\beta}}{\rho_j^2} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} \quad (7)$$

$$\sigma^{\alpha\beta} = -p\delta^{\alpha\beta} + \tau^{\alpha\beta} \quad (8)$$

$$\tau^{\alpha\beta} = \mu\epsilon^{\alpha\beta} \quad (9)$$

$$\epsilon^{\alpha\beta} = \frac{\partial \mathbf{v}^\beta}{\partial \mathbf{x}^\alpha} + \frac{\partial \mathbf{v}^\alpha}{\partial \mathbf{x}^\beta} - \frac{2}{3}(\nabla \cdot \mathbf{v})\delta^{\alpha\beta} \quad (10)$$

上付き添え字の α と β は軸の方向を表し、その項は軸の数だけ展開して足し合わせることを表す。 σ は応力テンソルであり、これは圧力成分 p と粘性応力 τ から成る。 $\delta^{\alpha\beta}$ はデルタ関数である。

3.4 人工粘性

SPH 法に衝撃波を扱うために Monaghan らによって人工粘性項が導入された⁵⁾。これにより、近づき合う2つの粒子の運動量を交換し、粒子のすり抜け現象を防ぐ効果を実現する。

$$\Pi_{ij} = \begin{cases} \frac{-\alpha\bar{c}_{ij}\phi_{ij} + \beta\phi_{ij}^2}{\bar{\rho}_{ij}} & \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} < 0 \\ 0 & \mathbf{v}_{ij} \cdot \mathbf{x}_{ij} \geq 0 \end{cases} \quad (11)$$

$$\phi_{ij} = \frac{h_{ij}\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2 + \varphi^2} \quad (12)$$

$$\bar{c}_{ij} = (c_i + c_j)/2 \quad (13)$$

$$\bar{\rho}_{ij} = (\rho_i + \rho_j)/2 \quad (14)$$

$$h_{ij} = (h_i + h_j)/2 \quad (15)$$

```
for() { // timestep
  neighbour search()
  density calculations()
  pressure calculations()
  force calculations()
  time integration()
}
```

図2 プログラムの流れ

α, β は定数で、1.0程度が用いられる。 c は音速である。この項を式 (7) に加え、実際のシミュレーションでは次式 (16) を用いる。

$$\frac{D\mathbf{v}_i^\alpha}{Dt} = \sum_{j=1}^n m_j \left(\frac{\sigma_i^{\alpha\beta}}{\rho_i^2} + \frac{\sigma_j^{\alpha\beta}}{\rho_j^2} + \Pi_{ij} \right) \frac{\partial W_{ij}}{\partial \mathbf{x}_i^\beta} \quad (16)$$

4. 実装

実装の手順として単一CPU (PPU) で動作するSPH法プログラムを作成し、次に並列化を行う。

4.1 プログラムの概要

SPH法プログラムの流れを図2に示す。プログラムの実装には倍精度浮動小数点を用いた。時間積分には速度ベレル法を用いた。領域は3次元立方体領域とし、領域から出る粒子に対しては面に対して反対方向の速度を与える完全弾性衝突とした。

速度ベレル法は式 (17)(18) で与えられる。

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t) + \mathbf{a}(t + \Delta t)}{2} \Delta t \quad (17)$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t) \Delta t + \frac{\mathbf{a}(t)}{2} \Delta t^2 \quad (18)$$

4.2 データ構造

Cell BE プロセッサでの並列化を考え、DMA 転送を行いやすいデータ構造を採用した。SPH法プログラムでは、計算の大部分を占めると予想される近傍探索や加速度計算を行う際に、複数粒子のデータにアクセスする。そこで、物理量ごとに配列として保持することで、まとめてDMA転送が行えるようにした。また、DMA転送を最適にするようにそれぞれの配列の先頭を128バイト境界にアラインした。近傍探索で計算したデータをペア情報の構造体のリストとして保持し、加速度の計算時に使いまわすことで余計な計算量を減らすことにした。構造体のサイズは64バイトになるように調整し、構造体のリストが128バイト境界に乗りやすいようにした。

4.3 近傍探索の高速化

SPH法では流体は粒子の持つ物理量の足し合わせ

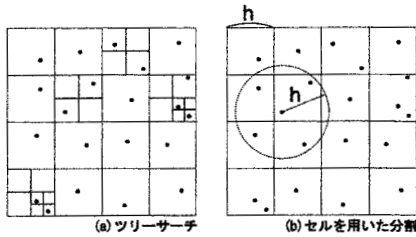


図3 近傍探索

で表現され、各粒子の運動は流体の持つ物理量による影響として表現される。見方を変えると、粒子の運動は他の粒子の物理量の足し合わせに影響していると考えることができる。単純な SPH 法の実装では、全粒子同士の相互作用として計算するが、これは粒子数 n に対して $O(n^2)$ のコストがかかる。

SPH 法の物理量の足し合わせはカーネルを用いて表現されるが、カーネルは中心から一定の距離遠ざかるとゼロとなる特徴を持っている。このため、カーネルの距離以上離れた粒子同士の相互作用は無視することができ、相互作用の計算は、カーネルの距離内の近傍にある粒子同士の計算のみに限ることができる。この特徴を考慮したツリーサーチ法と、領域をセルと呼ばれる格子状に区切る方法がある。どちらも、粒子の空間的な隣接関係を、木またはリストなどのデータ構造を用いて効率的に表現する方法である (図3参照)。

ツリーサーチ法では領域を2次元では4分割、3次元では8分割し、分割された小領域を子を持つツリーを作る。この小領域内に粒子が0個か1個のみある状態になるまで、領域分割を全領域に対して行っていく。これにより、各葉に粒子を0個あるいは1個持つツリーが作成できる。このツリーを用いることで、粒子同士の空間的な隣接をツリーの隣接情報によって調べることが出来る (図3(a))。この方法では $O(n \log(n))$ のコストで近傍探索を行える。

本論文では後者の領域をセルと呼ばれる格子状に区切る方法を採用する。この方法は、1辺を粒子の影響距離 h とした格子で領域を区切る。粒子は座標によりいずれかのセルに対応付けられ、セルの隣接情報を用いることで、ある粒子の近傍粒子をセル空間で隣接している粒子に限定することができる (図3(b))。この方法では、粒子同士の平均隣接数 m に対して $O(mn)$ のコストで近傍探索を行える。

4.4 セルの実装

一つのセルに対応付けられる粒子の数は一定に決まらない。考えられる実装として、各セルに対して粒子数よりも多めの配列を用意し、セルに含まれる粒子の

個数の変動をある程度許容する方法がある。この場合、実際にセルに含まれる粒子数に対して、無駄なメモリが非常に多くなる。Cell BE の SPE の LS の容量は極めて小さいため、できるだけメモリ消費を避けたい。

一方、粒子をセルに対応付ける度にメモリを確保するリスト構造を用いる方法では、メモリの容量は必要最低限で済むが、毎回メモリ確保を行うためのコストがかかってしまう。

これらを解決する方法として座標をハッシュに変換する方法がある⁶⁾。これは、粒子とセルの座標からハッシュ値を生成し、いくつかの1次元配列上にセルと粒子の関係を表現するものである。ハッシュ値の衝突が起こった時に近傍粒子のペアが重複して生成されるのを防ぐために実装は多少複雑になるが、必要なメモリ量が少なく、途中でのメモリ確保も不要なため効率が良い。

4.5 Cell BE プロセッサでの実装

PPE は PowerPC と互換性を持つため、作成したプログラムは特に問題なく PPE 上で動作した。しかし、Cell BE プロセッサによる並列化に当たっては、ハードウェアの特性に注意する必要がある。

4.5.1 倍精度演算

SPE では倍精度の除算と平方根演算はハードウェアでサポートしていない⁷⁾。倍精度の除算と平方根演算はソフトウェアで行われ、実行速度が低下する。除算ではニュートン法を用いて反復計算が行われ、収束時間によって計算時間が増加する。そこで、1億回の繰り返し計算を行い処理時間を比較した。表1にその結果を示す。倍精度演算では加算・乗算に比べ、平方根演算では約6.6倍、除算では約33.8倍のコストがかかっている。特に除算のコストが非常に高いことが確認された。

4.6 並列化

作成した逐次プログラムを用いて、1000粒子・1000ステップの計算を行い、プロファイリングしたところ表2、近傍探索の処理が約72.33%、次いで加速度の計算が約17.70%を占めることがわかった。そこで、プログラムの多くの部分を占める近傍探索の処理を並列化する。

演算	倍精度 (msec)	単精度
加算	918	697
乗算	918	697
平方根	6083	6622
除算	31020	1869

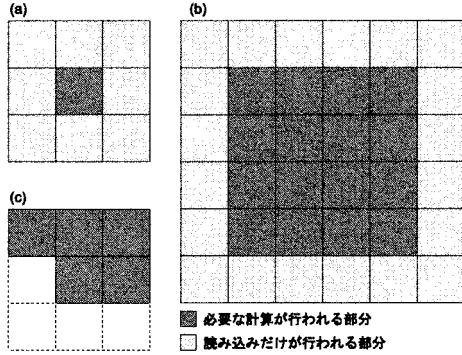


図 4 処理の分割方法

4.7 処理の分割単位

SPE は 256KB の LS を持つが、SPE バイナリ自体も LS に格納されるため、実際に利用できる領域は更に小さい。そこで本論文では、問題全体を細かく分割して SPE に割り当てることでこれを解決する。近傍探索の処理は隣接したセル同士に計算が限られるので、隣接したセルのかたまりを一つの処理単位とした。

かたまりの選び方によって計算密度が異なる図 4。中心のセルとその周囲のセルをかたまりとする場合(図 4(a))、周囲のセルは中心のセルの計算のために読み込まれるが、周囲のセルの計算自体はこの時行われず、周囲のセルの読み込み回数が重複し無駄が多い。かたまりを大きく取り、重複した読み込みを減らすことが可能であるが、計算に必要なデータサイズが大きくなり、LS の容量の制限を解決できない(図 4(b))。

近傍粒子のペアに対する計算は対称に行われることを考慮すると、計算結果を両方の粒子に適用することができる(図 4(c))。計算に必要なセルの数は 2 次元では周囲 8 マスのうち 4 マス、3 次元では 13/26 マスとなり、周囲のセルを半分だけ選択すればよい。計算密度が高く無駄が少ないことから、本論文ではこの方法を採用した。

4.8 ワークブロックの作成

中心となるセルと、これに隣接するセルをまとめて 1 つのワークブロックとした(図 5(a))。近傍探索の処理は全セルに対してワークブロックを作成し、各 SPE にはワークブロックのリストを分割して割り当てた。

処理	処理時間の比率 (%)
neighbour search	72.33
density calculations	3.26
pressure calculations	0.06
force calculations	17.70
time integration	0.60

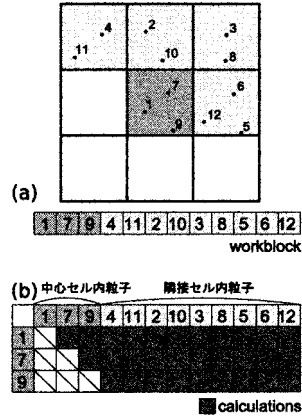


図 5 処理単位となるワークブロックと計算量の見積もり

1 つのワークブロック内の近傍探索にかかる計算コストは、図 5(b) のように、中心セル内の粒子と隣接セル内の粒子の個数から見積もることができるので、各 SPE には計算コストが均等になるようにワークブロックを割り当てた。SPE はワークブロック内での近傍探索を行い、近傍粒子のペアのリストを作成する。実装では、まずワークブロックの情報を転送し、各ワークブロックに含まれる粒子の情報を適時転送した。

ワークブロックに含まれる粒子のデータはそれぞれ離れており、個別に DMA 転送を行うと通信のレイテンシの影響が大きくなる。そこで、近いデータ同士をまとめて転送し、転送回数を減らした。

4.9 結果

図 6 は、SPE の利用数を変えたときの実行時間である (SPE0 は PPE のみでの実行)。並列化を行った近傍探索部分では、PPE での処理に比べて約 5.5 倍の性能向上が見られた。しかし、他の処理部分が支配的になってくるためプログラム全体での性能向上は約 2 倍にとどまっている。

近傍探索の並列化により、加速度計算がプログラムの多くを占めるようになったので、これを並列化することを試みるが、ここで、近傍探索処理で毎ステップに 1 回行われる倍精度による除算部分を、単精度演算に変更してみたところ、約 20% の速度向上が見られた。これは、もとのプログラムで倍精度の除算が性能に大きく影響していることを示している。加速度計算では SPE の苦手とする除算が多く現れるため、除算が必要な部分を PPE 側で計算することにした。また、加速度計算に必要なデータが散らばっているため、これを PPE がまとめて SPE に割り当てることにした。しかし、この方法では PPE の処理が多く、並列性が

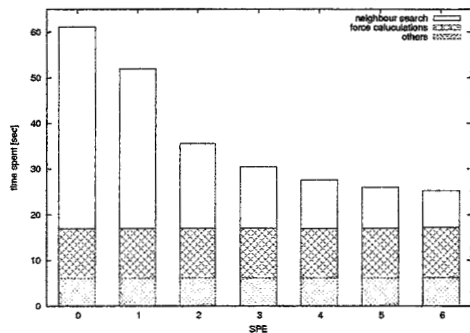


図 6 並列化による効果

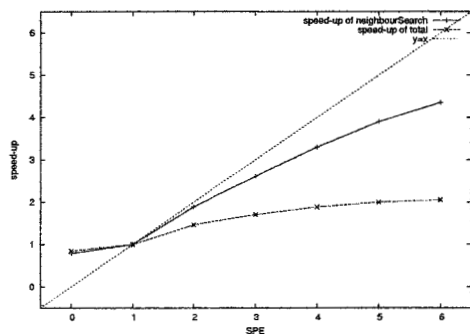


図 7 並列化によるスピードアップ

低いために良い性能向上は得られなかった。除算のコストが高くても並列性の高いアルゴリズムを選択する必要があると考えられる。

5. 考 察

4.7節で述べたように、ワークブロックとして周囲のセルのうち半分だけを選ぶ方法を採用した。全周囲を探索する方法では、中心セル内の粒子の計算が1回のワークブロックの処理で完了しているのに対し、半分を探索する方法では、1回のワークブロックの処理だけでは完了しない。このため、半周囲の探索では各ワークブロックでの計算結果を粒子についてまとめる作業が必要となる。これは、並列化においては同期をとることを意味する。

全周囲の探索では近傍探索の処理から、密度、圧力計算までまとめて処理できるのに対し、半周囲の探索では各処理のたびに同期をとらなくてはならない。したがって、計算密度において多少不利となるが、プログラム全体の並列性を考えると全周囲を探索する方法の方が良いパフォーマンスを示すと考えられ、アルゴ

リズムに改善の余地があると思われる。

近傍探索の並列化において、SPEの数が増すに連れてスピードアップの変化が小さくなっている。SPEの処理単位であるワークブロックには、含む粒子数にばらつきがあるため、SPEの数が増えるに連れてこの影響が大きくなり、SPEへのタスクの割り当てに不均等が起こっていることが分かった。これも、全周囲の探索を行う方法を用いることでワークブロックの均等化が見込まれ解決できると考えられる。

6. おわりに

Cell BE プロセッサで動作するSPH法プログラムを作成し、近傍探索部分の並列化を行った。並列化によって近傍探索部分における計算速度の向上が得られたが、プログラム全体での速度向上は不十分であった。並列性の高いアルゴリズムを選択することで改善する余地がある。

SPEは倍精度浮動小数点の除算と平方根演算は他の演算に比べて非常に計算コストが高い。これらの演算を多用する科学計算においては、Cell BEプロセッサの性能引き出すのが困難である。

今後、並列性の高いアルゴリズムによるプログラム全体の並列化、SIMDを用いた高速化を行うとともに、SPH法プログラムでの除算・平方根演算の回避の方法を探っていく。

参 考 文 献

- 1) G.R.Liu and M.B.Liu, Smoothed Particle Hydrodynamics a meshfree particle method, World Scientific, 2003.
- 2) 原田 隆宏, 田中 正幸, 越塚 誠一, 河口 洋一郎: 粒子ベースシミュレーションの並列化, 情報処理学会論文誌 Vol.48 No.11, pp.3557-3567, 2007.
- 3) Tsuyoshi Hamada, T.Oshiaki Iitaka: The Chamomile Scheme: An Optimized Algorithm for N-body simulations on Programmable Graphics Processing Units, arXiv:astro-ph/0703100, 2007.
- 4) IBM: Cell Broadband Engine Architecture. <http://www-128.ibm.com/developerworks/power/cell/>.
- 5) Monaghan J. J. and Gingold R. A., Shock simulation by the particle method SPH, Journal of Computational Physics, 53:374-389, 1983.
- 6) Nils Hjelte: Smoothed Particle Hydrodynamics on the Cell Broadband Engine, Master's Thesis in Computing Science, 2006.
- 7) IBM: Synergistic Processor Unit Instruction Set Architecture. <http://www-128.ibm.com/developerworks/power/cell/>.