

Cell/B.E. による倍精度粒子法の高速度化

赤木 正慶 今村 俊幸

電気通信大学 電気通信学研究科

粒子法は粒子数の増加により膨大な計算時間を要するため、ハードウェア利用による高速化も行われている。Cell/B.E. は高い浮動小数点演算能力を持つが、倍精度数値計算利用はあまり行われていない。そこで SONY PLAYSTATION3 に搭載されている Cell/B.E. に対して、粒子法の一つである SPH 法プログラムの一部を最適化した。データ分割による SPE 並列化などにより、SPE の台数によるリニアな高速化を実現した。また要求帯域とメモリアクセス集中によってボトルネックが生じることを示し、特にデータサイズの大きくなる倍精度数値計算では、その傾向が高いことを示した。倍精度数値計算では、転送と処理のバランスを検討し、帯域を有効に活用するような設計が必要であることを示した。

1 背景と目的

粒子法プログラムでは粒子数の増加につれ、膨大な計算時間を要する。これは、粒子の相互作用の計算や、相互作用する粒子ペアを見つける近傍探索による影響が大きい。単純な方法では、粒子数 N に対して $O(N^2)$ の計算時間を要するが、良いアルゴリズムを用いて高速化が可能である。また、複数コアや GPU を利用するなど、ハードウェアによる高速化のアプローチも可能である。Cell/B.E. による高速化 [4] も行われているが、倍精度浮動小数点での利用は多くない。

本研究では粒子法の一つである SPH(Smoothed Particle Hydrodynamics) 法を対象とし、近傍探索アルゴリズムとして Incremental sort-and-update algorithm(ISU) を適用する。倍精度浮動小数点で実装した逐次 SPH 法プログラムの一部に対し、Cell/B.E.(Cell Broadband Engine) に対して最適化を図る。これらを通じて、倍精度粒子法プログラムにおける Cell/B.E. の効率的利用について調査する。

2 SPH 法

SPH 法 [1] は粒子法の一つであり、ガス星雲の数値計算に用いられた手法であるが、近年の研究によって非圧縮流体や剛体の計算への適用も可能となっている方法である。

カーネルと呼ばれる関数を用いて粒子の持つ物理量を平滑化し、その足し合わせで流体の物理量を表現する。一定距離以上離れた粒子同士は、カーネルの影響が無いいため、相互作用の計算が省略できる。そのため、プログラムでは、あらかじめ相互作用する粒子のペアを探索しておくことで効率が良い。

2.1 カーネル

カーネル W は原点でピークを取る釣鐘形の関数が用いられ、粒子 i, j に対するカーネルとカーネルの勾配は式 (1),(2) のように表現される。

$$W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, h_{ij}) \quad (1)$$

$$\nabla W_{ij} = \frac{\mathbf{x}_{ij}}{r_{ij}} \frac{\partial W_{ij}}{\partial r_{ij}} \quad (2)$$

ここで $h_{ij} = \frac{h_i + h_j}{2}$ は影響半径の平均、 $r_{ij} = |\mathbf{x}_{ij}|$, $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ である。

2.2 支配方程式

カーネルを用いて質量保存則と運動方程式は式 (3),(4) のように表現される。

$$\frac{D\rho_i}{Dt} = \sum_{j=1}^n m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij} \quad (3)$$

$$\frac{D\mathbf{v}_i}{Dt} = \sum_{j=1}^n m_j \left(\frac{p_i^{\alpha\beta}}{\rho_i^2} + \frac{p_j^{\alpha\beta}}{\rho_j^2} + \Pi_{ij} \right) \nabla W_{ij} \quad (4)$$

ここで、 m, ρ, p は、それぞれ質量、密度、圧力である。 Π_{ij} は人工粘性項と呼ばれ、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。

3 近傍探索アルゴリズム

Incremental sort-and-update algorithm (ISU)[2] は、多面体の衝突判定に用いられている手法であり、sort-and-prune と呼ばれる。衝突ペアのリストの変化分のみを探索し、逐次更新することで、毎ステップリストを作り直す必要がない。衝突判定を多面体を囲む BoundingBox の衝突に近似し、さらに、次元ごとの探索に問題を分割する。

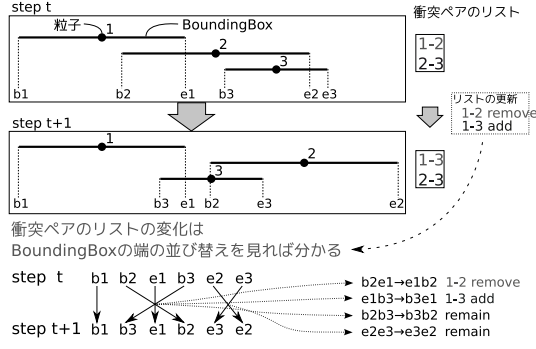


図 1: 具体例：衝突ペアの追加と削除 (1 次元)

表 1: 端点の並び替えパターンに対する衝突ペアリストへの操作

パターン	操作
$b_i e_j$ $e_j b_i$	ペア i, j を削除
$e_i b_j$ $b_j e_i$	ペア i, j を追加
$b_i b_j$ $b_j b_i$	なにもしない
$e_i e_j$ $e_j e_i$	なにもしない

3.1 アルゴリズムの概要

最初のステップでは初期の衝突ペアリストを作成しておく。次のステップからは変化分のみを探索し、リストを更新していく。

図 1 に、1 次元での探索の例を示す。各粒子を囲む BoundingBox の両端点 $b, e (b < e)$ に注目して変化分の探索を行う。全粒子の端点を毎ステップ座標順にソートし、前のステップから並び順が変化した 2 つの端点に注目する。表 1 に示すように、4 つの並び替えパターンに対して、衝突ペアへの操作を行う。多次元では、1 つ低い次元の衝突情報を用いて変化分の探索が可能である。

3.2 SPH 法への適用

SPH 法では、ペアの粒子の中心距離が、両粒子の影響半径の平均以下であれば、2 つの粒子は近傍と判定される。これに対して ISU では、BoundingBox の衝突判定によって近傍を判定する。

本研究では、BoundingBox の大きさを、粒子の影響半径に合わせて設定し、ISU による近傍が、SPH 法の近傍を包含する関係になるようにした。ここでは、ISU によるペアを衝突ペア、SPH 法によるペアを近傍ペアと呼び区別する。

表 2: 逐次プログラムの処理の内訳

section	time[sec]	比率
プログラム全体	16.542 sec	100.00%
ISU	1.091 sec	6.60%
ISU の後処理	3.563 sec	21.54%
カーネル計算	3.053 sec	18.46%
密度	2.411 sec	14.58%
人工粘性	2.426 sec	14.67%
加速度	2.474 sec	14.96%
その他	1.524 sec	9.21%

3.2.1 ISU の後処理

相互作用計算ではここで得られた、相互作用ペアリストを用いて計算するが、衝突ペアリストは、SPH 法の近傍ペアリストよりもサイズが大きく、無駄がある。そこで、衝突ペアリストの中から、SPH 法の近傍を満たすものを取り出すことで近傍ペアリストを作成する。この処理によって、リストのサイズが、3 次元問題では、約 4 割から 5 割程に削減される。

4 プログラムの概要

CellSDK2.1 を利用してプログラムを開発した。コンパイラは gcc4.1.1 を利用した。Fedora 6(kernel 2.6.21) を導入した PS3 で実行する。PPE, SPE 共有のタスクキューを用いて、SPE へのタスク割り当てを行う。

4.1 並列化対象

逐次のプログラムの一部を 6 基の SPE を用いて並列処理する。表 2 に ISU を適用した SPH 法プログラムの実行時間の内訳を示す。ISU の後処理に最も時間を費やしており、この部分を並列化する。

4.2 ISU の後処理の並列化

PPE は、衝突ペアリストをデータ分割し、これを処理するタスクを SPE に割り当てる。SPE はタスクを処理して結果を PPE に返す。

ISU の後処理では、以下のように実行する。

- タスクの指定する衝突ペアリストを LS に転送

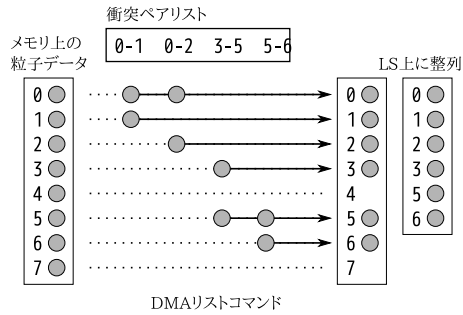


図 2: 分散した粒子データの転送

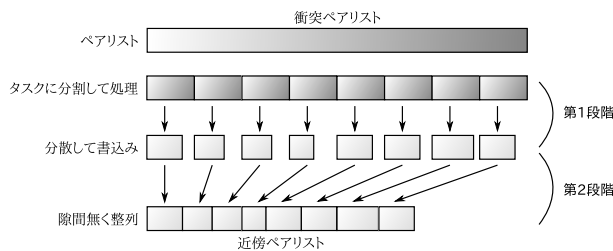


図 3: 近傍ペアリストの2段階書き込み

- 衝突ペアリストの含む粒子データを LS に転送
- 衝突ペアリストから近傍ペアリストを作成
- 作成した近傍ペアリストをメモリに転送

4.2.1 粒子データの受信

衝突ペアリストに含まれる粒子は、順番のばらばらな重複を含むものである。このリストから、重複せずに必要なデータのみを LS に転送する。図 2 に示すように、DMA リストコマンドを用いてこれを実装した。

4.2.2 近傍ペアリストの送信

作成した近傍ペアリストをメモリに転送する際、近傍ペアリストはメモリ上に1ヶ所に整列した方が、後の計算のために都合が良い。しかし、近傍ペアリストのサイズは処理の前には決定せず、1回の処理で整列させることは難しい。そこで、メモリへの転送を2段階に分け、1段階目の転送ではメモリの分散した箇所にリストを転送し、2段階目でそれらの分散したデータを1ヶ所に集める(図3)。

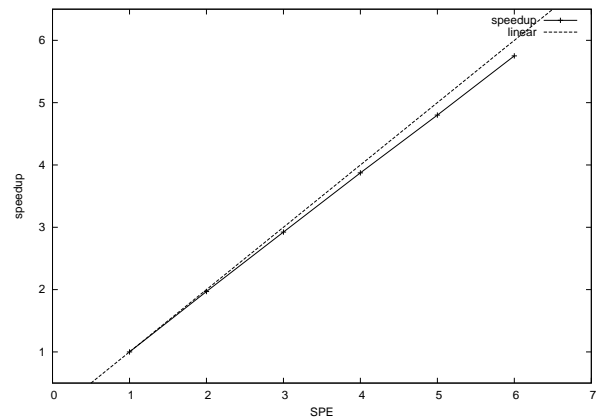


図 4: SPE1 基の実行時間に対するスピードアップ

4.3 メモリへのアクセス集中

2段階目の処理では、メモリ上の位置の移動の他には、データ整形の小さな処理しか行わないが、ダブルバッファリングによって十分に通信時間を隠蔽できるものである。しかし、頻繁に転送要求が出されるため、SPE1 基当たり約 10.4GB/s の帯域を必要とした。これに対して、メモリのバスの帯域は 25.6GB/s であり、複数の SPE が同時にメモリに集中アクセスすると、ボトルネックが生じることが分かった。そこで、2段階目の処理に加えてカーネル計算を行うことで、計算量を人為的に増やし、転送要求の頻度を下げることでこれに対応した。カーネル計算を追加することで、SPE1 基当たりに必要な帯域は約 2.68GB/s となり、ボトルネックを解消した。

5 結果と考察

立方体状に配置した 8000 個の非圧縮粒子の SPH 法計算を行い、並列化の効果を確認した。図 4 に並列化部分の、台数効果によるスピードアップを示す。SPE6 基では SPE1 基と比較して約 5.77 倍の高速化を達成した。

また PPE1 基に対しては約 17.2 倍、Intel Core i7 920(2.66GHz, 1 コア使用) に対しては約 1.33 倍で動作した。

飛び飛びの粒子データへのランダムアクセスは、DMA リストコマンドを利用することで実現した。また、ランダムアクセスの転送も、通常の転送と同様に、通信時間を隠蔽して実行できることを確認した。

5.1 メモリバスのボトルネック

最適化の途中でメモリバスのボトルネックが生じた。この原因は2つある。1つは処理と転送のバランスである。転送量に対する計算量が少ないと、転送頻度が増加し要求帯域が増加する。

1度に多くのデータの転送を必要とする計算では、転送頻度が増加しボトルネックを生じやすい。特に倍精度による実装では、扱うデータサイズが大きくなるため注意が必要である。

もう1つの原因はメモリへのアクセス集中である。EIBとメモリ間のバスの帯域は25.6GB/sであり、EIBの帯域(204.8GB/s)と比較して小さい。メモリ-LS間の転送を頻繁に行うと、1本のバスに対する集中アクセスがおき、帯域不足により律速されてしまう。これはEIBとLS間のバスでも同様であり、1ヶ所へのアクセスが集中には注意が必要である。この問題に対しては文献[3]が参考になる。

これらの原因が重なることでメモリバスのボトルネックが生じた。これを解消するためには、処理と転送のバランスの改善か、アクセス集中を回避する設計が必要である。

6 議論

Cell/B.E.の倍精度数値計算利用について議論する。

SPEでは一部の倍精度演算が遅いという問題があり、倍精度数値計算での利用には注意が必要である。

一方倍精度浮動小数点演算の強化された第2世代のCell/B.E.が登場しており、処理の高速化が期待できる。

6.1 逆数と平方根演算

SPH法のカーネル計算では、逆数と平方根演算に多くの処理時間を費やしている。これは倍精度の逆数、平方根演算がハードウェアで実装されていないためである。そのためソフトウェアで演算する必要があり、乗算などと比較して10倍程度のレイテンシを要する。

しかし逆数、平方根演算が高速化されても単純にプログラムが高速化するとは言えない。処理が高速化されれば、要求帯域が増加するためである。

ここで逆数や平方根演算が乗算と同じレイテンシで実行できると仮定する。この仮定により、本研究のプログラムのカーネル計算では約54%の処理時間の削減が見積もられる。ところがSPE1基当たりの要求帯域は5.78GB/sとなるため、4基か5基の使用で、ボトルネックが生じる

と予想される。このように処理の高速化は要求帯域の増加によるボトルネックを生じやすくと考えられる。

6.2 第2世代のCell/B.E.

第2世代のCell/B.E.(PowerXCell 8i)では、倍精度浮動小数点演算の強化されたeDP SPE(enhanced double precision SPE)を搭載しており、乗算などの基本的な倍精度浮動小数点演算が、高速化されている。しかしeDP SPEの利用時は、要求帯域の増加に更なる注意が必要である。

7 まとめと今後の展望

倍精度SPH法プログラムのISUの後処理とカーネル計算部分を、Cell/B.E.に最適化した。1SPEに対して6SPEで約5.77倍の高速化を達成した。

また、転送量に対する計算量が少ない処理では、転送頻度の増加により、1本のバスによるボトルネックが生じやすいことを示した。さらに倍精度数値計算では、扱うデータサイズが大きくなるため、転送帯域のボトルネックが生じやすい。特に倍精度浮動小数点演算の強化されたSPEを利用する場合には、計算時間が短くなるためバランスに注意する必要がある。

今後の展望としてPowerXCell 8i上での倍精度数値計算の評価と、他の部分の並列化が挙げられる。その際には十分に計算量と転送量のバランスの検討と、EIBを活用するプログラム設計が必要である。

参考文献

- [1] G.R.Liu and M.B.Liu: Smoothed Particle Hydrodynamics a meshfree particle method, World Scientific, 2003.
- [2] Alexander Shinner: Fast algorithms for the simulation of polygonal particles, Granular Matter, Vol.2, No.1, 35-43, 1999.
- [3] Michael Kistler, Michael Perrone, Fabrizio Petrini: Cell Multiprocessor Communication Network: Built for Speed. IEEE Micro, 2006.
- [4] Nils Hjelte : Smoothed Particle Hydrodynamics on the Cell Broadband Engine , Master 's Thesis in Computing Science, 2006.