

Cell Broadband Engine における SPEC CFP2000 の並列化

山口 武美[†] 大津 金光[†] 横田 隆史[†] 馬場 敬信[†]
[†]宇都宮大学工学部情報工学科

1 はじめに

近年、シングルコアプロセッサでの性能向上が難しくなり、マルチコアプロセッサが主流となってきている。その中でもソニー、IBM、東芝により次世代プロセッサとして開発されたマルチコアプロセッサ Cell Broadband Engine (以下、Cell プロセッサ) がある。

Cell プロセッサはメインプロセッサ (PPE) が 1 個、サブプロセッサ (SPE) が 8 個搭載されている非対称型マルチコアプロセッサである。PPE, SPE 共に複雑なスケジューリング機構を搭載しないことで単純化し、高クロック化を実現している。複数の SPE の並列処理により、その高い演算能力とデータ転送能力において非常に高いパフォーマンスを持つ [1]。

そこで、本研究では SPEC CFP2000 ベンチマークのプログラムを対象として Cell 向けに並列化し、並列化手法の検討及び評価を行う。プログラム中で最も呼び出されている関数のホットループを並列化の対象とし、処理データを各 SPE に分割して並列実行させる。本稿では、評価環境として Cell プロセッサを搭載した実機であるプレイステーション 3 (以下、PS3) を用いる。PS3 はアプリケーションに使用可能である SPE 数が 6 個となっており、この SPE を有効活用することが性能向上に繋がると考えられる。

以上の条件で使用 SPE 数、データ分割数を変えながら評価を行うことで、Cell プロセッサの演算能力とデータ転送能力を発揮できる並列化手法を検討する。

2 Cell ハードウェア構成

Cell プロセッサのハードウェア構成を図 1 に示す。Cell プロセッサはメインプロセッサコアの役割を持つ PowerPC Processor Element (PPE) が 1 個とサブプロセッサコアの役割を持つ Synergistic Processor Element (SPE) が 8 個、及び I/O がそれぞれ Element Interconnect Bus (EIB) で接続されている。

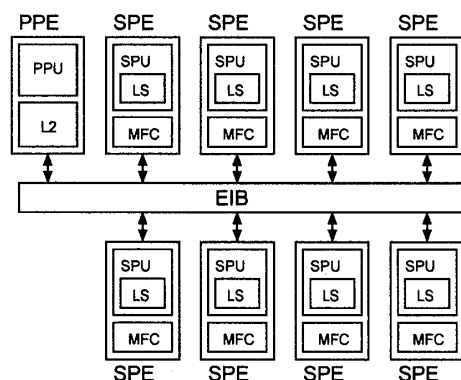


図 1: Cell ハードウェア構成

2.1 PPE

PPE は 64 ビット PowerPC アーキテクチャと同等の機能を有した汎用プロセッサであり、OS の駆動やアプリケーションの実行、及び SPE の制御を行う。PPE には、PPE の演算処理を行う核となるユニット PowerPC Processor Unit (PPU) が含まれる。

2.2 SPE

SPE はデータ演算処理を得意とするシンプルなプロセッサコアであり、SIMD 演算能力を持つ。SPE の演算処理を行う核となるユニット Synergistic Processor Unit (SPU) は 256KB の Local Storage (LS) と呼ばれる専用メモリを持ち、そこにプログラムとデータを格納し実行する。SPU は DMA 転送を用いてのみメインメモリと LS 間のデータ転送を行うことが可能で、Memory Flow Controller (MFC) がその役割を担う。

2.3 並列化の検討

図 1 から分かるように、Cell プロセッサの大部分は SPE で占められている。また、PPE と SPE は全く別個のコアなので完全に独立動作できる。そのため、PPE で処理していた仕事の一部を待機中 SPE に振り分けるようにすれば、PPE は振り分けた仕事の処理時間を短縮することができる。PPE は OS の駆動やリソース管理はもとより、SPE を使わない全てのプログラムが動いているので、PPE の CPU 時間は高価であり、極力 SPE を使うべきである。このことから SPE を使用した並列処理は Cell プロセッサの性能を引き出す上で欠かせない手法であると判断できる。

3 並列化手法

本章では並列化のための PPE でのデータ分割や DMA 転送、及び同期処理の手法等を検討する。また、並列化の説明図を図 2 に示す。図のようにデータを SPE 個数分に分割し、PPE が起動したそれぞれの SPE で演算させ、結果を戻すという手法になる。

開発環境としては、IBM が提供している CellSDK2.1[2] を用いる。並列化を適用する対象は SPEC CFP2000 ベンチマークの流体シミュレーション 171.swim 中のホットループとする。

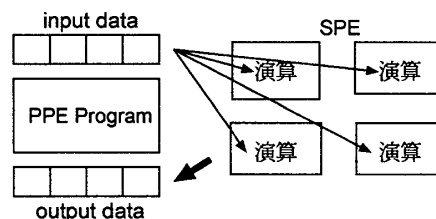


図 2: Cell プロセッサを用いた並列化手法

3.1 データ分割

SPE に仕事を割り振るに当たり、PPE で処理するデータを SPE 台数分に分割する必要があり、それを後述のように DMA 転送サイズを考慮して分割する。

171.swim ではホットループに使用するデータは配列データとなっており、その全てを一度に DMA 転送するのは不可能なので、その配列から 1SPE に処理させる分を取り出して各 SPE に DMA 転送させる。具

体的には配列の取り出したい領域の先頭アドレスをそれぞれ SPE に転送し、受け取った SPE がそのアドレスを用いてメインメモリにある配列の領域を DMA 転送で LS に持ってくるという手法である。DMA 転送により受け取った配列の要素 (分割数により数は異なる) は SPE で処理され、最初に受け取った先頭アドレスを用いて今度は LS からメインメモリへと結果を戻す。この作業を繰り返すことで SPE を用いた並列処理が実現できる。

3.2 DMA 転送

DMA 転送する上で気をつけなければならないのは転送量の制約である。SPE の LS の容量が 256KB であり、一度の転送で 16KB を超えるデータ量を送ることは出来ない。また、1 回あたりの転送サイズは 1,2,4,8,16Byte かそれ以上で 16Byte の倍数でなければならない。後者については、転送アドレスを 128Byte 境界に揃えることで転送速度が最大となる。この転送アドレスの整列が正確に成されていない場合、バスエラーとなり実行が終了する。

この厳しい制約下で、効率の良いデータ分割及び転送を行うことが処理速度の向上に繋がる。

3.3 同期

PPE と各 SPE との同期を取るためにフラグを用いる。具体的には以下の手順となる。まず PPE で SPE を台数分起動し、PPE から SPE 開始のフラグを SPE に送る。次に起動した各 SPE プログラムで開始フラグを受け取り演算を実行する。そして結果と共に SPE 終了のフラグを PPE に送り、PPE がフラグを検出し SPE を使用した演算のサイクルが終了する。

なお、PPE が開始フラグを出し SPE からの終了フラグを受け取るまでの間、PPE は待ち状態となる。

4 並列化と評価

171.swim のホットループを以下の手順で並列化し SPE 台数を 1, 2, 4 台、また DMA 転送サイズを 16, 32, 64, 128 と変化させながら評価を行う。入力データセットは train を用いる。

1. メインメモリの処理データを SPE 個数分に分割
2. 各 SPE スレッドを起動
3. メインメモリから LS への DMA 転送
4. SPE でデータを処理し、結果を PPE に転送
5. PPE で各 SPE での結果を受け取る

なお、評価中の時間計測は SPE スレッド起動前から SPE スレッド終了後の範囲で行う。

表 1: DMA 転送サイズによる処理時間の差

転送サイズ (Byte)	16	32	64	128
処理時間 (ms)	23.89	16.13	12.48	10.49

4.1 DMA 転送サイズによる評価結果

まず、DMA 転送サイズによる処理時間の違いを調べるために、SPE 台数を 4 台に固定して DMA 転送サイズを変化させた場合の処理時間の評価を行う。入力データセット train はホットループのイテレーション数が 512 回で、関数 Calc2 を 200 回呼び出している。その 200 回の平均を取り、表 1 に結果を示す。16Byte 転送では SPE4 台で計 64Byte 分の演算を 8 回行うことで総イテレーション回数の演算を実現している。同様に 32Byte 転送では 4 回、64Byte 転送では 2 回、128Byte 転送では 1 回となる。よって転送サイズが大きくなれば総オーバーヘッドは小さくなるが、その分 1SPE での計算量も大きくなると考えられる。しかし、表から DMA 転送サイズの方が 1SPE での計算量よりも、処理時間に影響を及ぼしていることが分かる。

このことから転送サイズの大きさを重視して 128Byte とし、SPE 台数を変えての評価を行う。

4.2 SPE 数による評価結果

先の結果から、次は転送サイズを 128Byte に固定して SPE 台数を 1, 2, 4 台と変化させて評価を行う。(SPE を使わずに)PPE プログラムのみで実行した時の処理時間を 1 としたときの速度向上率を表 2 に示す。SPE1 台実行では速度低下となり、2 台及び 4 台実行では速度向上となった。まず SPE1 台実行では速度が低下した。これは DMA 転送を 4 回も行う必要があったためであると考えられる。逆に、SPE2 台実行は DMA 転送回数が 2 回、SPE4 台実行は DMA 転送回数が 1 回で済むことになるので、転送のオーバーヘッドを SPE での並列実行による速度向上が上回ったことになる。

表 2: SPE 台数を変えた場合の速度向上率

SPE 台数	1	2	4
速度向上率	0.75	1.41	2.53

4.3 考察

今回のプログラムでは、ホットループの前で SPE を起動しプログラムをロードしていたが、この SPE の起動にかかる時間が処理時間に大きく影響を与えていることが判明した。表 3 は各 SPE 台数での時間計測の最初 3 つである。これを見れば分かるように、最初だけ 2 番目以降に比べて大きく時間が掛かっている。これは SPE を用いる上では仕方ない損失ではあるが、出来るだけこの SPE の起動時間を最小限に抑えることが全体の速度向上に繋がると言える。

表 3: 各 SPE での時間計測

SPE2 台 (ms)	111.77	18.59	18.51	18.52	...
SPE4 台 (ms)	135.99	9.79	9.67	9.89	...

5 おわりに

本稿では、SPEC CFP2000 ベンチマークの 171.swim に対して並列化を適用し、DMA 転送サイズと SPE 台数の変化による評価を行った。その結果、SPE 台数が同じならば DMA 転送サイズが大きく取られている場合の方が DMA 転送のオーバーヘッドが小さく済むために、処理時間の短縮へと繋がった。また、SPE 台数の違いによる評価も行った。結果、SPE 台数 4 台で実行した場合が最も速度向上し、Cell による並列実行が成功したと考えられる。

今後の課題としては今回の結果を踏まえた上で SPE へのデータ分割の手法を改善し出来る限り分割のオーバーヘッドを少なく、且つ一度に 128Byte 以上でデータを DMA 転送可能にすることが挙げられる。

謝辞 本研究は、一部日本学術振興会科学研究費補助金 (基盤研究 (B)18300014, 同 (C)19500037, 若手研究 (B)17700047) および宇都宮大学重点推進研究プロジェクトの援助による。

参考文献

- [1] 若林直樹, 大津金光, 横田隆史, 馬場隆信, “マルチコアプロセッサにおけるデータ転送能力から見た適切なコア数の検討”, 情報処理学会 第 69 回全国大会講演論文集, 2K-2, pp.1-67~1-68, 2007.
- [2] IBM, “Cell Broadband Engine resource center,” <http://www-128.ibm.com/developerworks/power/cell/>.