

PlayStation®3によるCellプロセッサの効率化

日大生産工(院) ○峯 和也
日大生産工 角田 和彦

1. はじめに

2006年、SCE (Sony Computer Entertainment) 社から特殊なアーキテクチャを持つ独自プロセッサCell/B. E. (Cell Broadband Engine:以下Cell)が搭載された据え置き型ゲーム機器PlayStation®3(以下PS3)が発売された事は記憶に新しい。

Cellの処理能力は内部のコアを協調させることで最大218GFLOPS(浮動小数点演算能力)に達すると言われており、国内ではCell向けのプログラミングコンテスト“Cellチャレ”(Cellスピードチャレンジ)[1]が3年間に渡り開催され、海外ではPS3にLinuxをインストールして並列処理をさせる研究室や並列プログラミングの授業にPS3を利用する大学が出てくるなど、国内外問わず注目を浴びている。

本研究ではPS3にLinuxをインストール[2]し、SIMD演算利用の有無によって計算処理時間にどの程度の差異が生じるかという事と、各計算箇所でのどの程度処理時間がかかっているかという事を絶対値計算プログラム[3]を用いて数通りの条件で比較し、最終的にどの程度までCellの効率化ができたかを検証することを目的としている。

2. 性能比較について

SIMD演算を利用することによって処理効率がどの程度向上するか、絶対値計算プログラムを用いて処理時間の性能比較を行う。

本研究では特に絶対値計算部分に注目し、PPE1基のみ、PPE1基+SPE6基の場合それぞれについてSIMD演算の有無、最適化オプション“-O3”の有無の場合も含めて計算処理時間をそれぞれ10回計測[4][5]し、その平均値を用いて処理性能の比較検討を行う。

プログラムの流れを以下に示す。

- ①. -5～+5の値をランダムに配列に代入する。
- ②. 配列を分割し、PPEからSPEに転送する。
- ③. 各SPEが絶対値の計算をする。
- ④. 計算済みの配列をSPEからPPEに転送する。
- ⑤. SPEプログラムを閉じ、処理を終了する。

3. DMA転送

DMA(Direct Memory Access)転送とは、プロセッサを介することなく周辺装置とメモリ間のデータ転送を高速に行うための手段である。

PS3の場合、SPE内部に搭載されているMFC(Memory Flow Controller)によって、SPEプログラムの実行とは独立して行われる[6]。

4. SIMD演算

SIMD(Single Instruction Multiple Data)演算は、スカラ演算に比べて計算処理を少なくすることができる演算手法である。

例えば、図1に示す様に配列の足し算を行う場合、スカラ演算では配列のそれぞれの要素に対し演算をするので4回の演算処理が必要だが、SIMD演算を使用すると4回の足し算を1回の演算処理で実行できるので、全体の処理時間を削減することができる[7]。

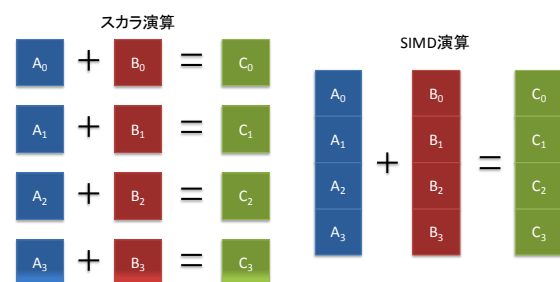


図1. スカラ演算とSIMD演算

Efficiency improvement of Cell processor by PlayStation®3

Kazuya MINE and Kazuhiko KAKUDA

5. システム環境及び計測条件

本研究で用いたPS3のシステム環境を表1に示す。計測条件は配列数24576、SPE基数6、ランダム値±5、メモリアラインメント値128である。

表1. PS3のシステム環境

PS3	
OS	Linux Fedora10
メモリ	256MB
CPU	Cell/B.E. 3.2GHz
Swap	1024MB

6. 結果及び比較検証

6-1. 絶対値計算のみの時間比較

絶対値計算のみの処理時間についての時間比較を図2-1～図2-3に示す。

図2-1はPPE1基のみと、PPE1基とSPE6基を協調させた処理も含む5つの計測条件の時間比較である。

図2-2、図2-3についてはPPE1基のみ以外のデータを抜き出してそれぞれSIMD演算の有無について詳細に比較した図である。

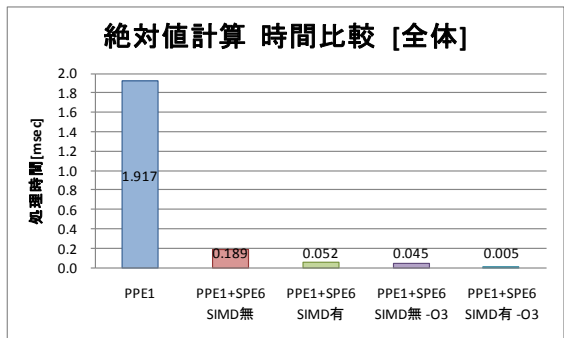


図2-1. 絶対値計算の時間比較[全体]

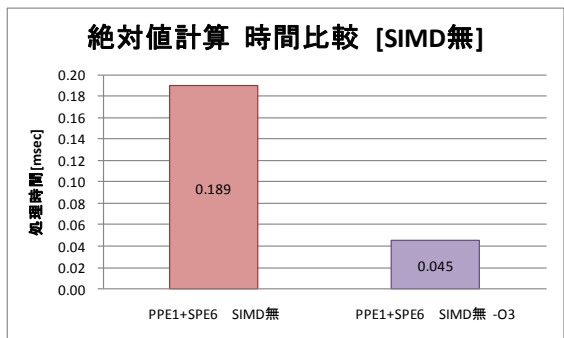


図2-2. 絶対値計算の時間比較[SIMD無]

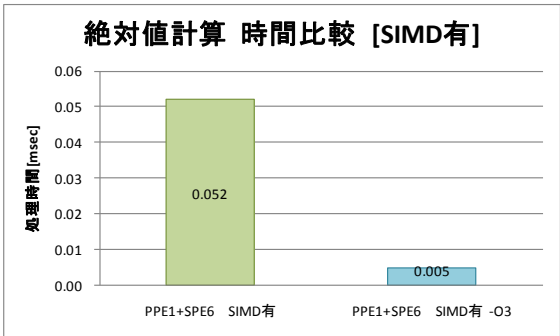


図2-3. 絶対値計算の時間比較[SIMD有]

表2. 絶対値計算のみの処理効率比較

計測条件	処理効率[倍]
PPE1基のみ	1
PPE1基+SPE6基 SIMD無	10.12
PPE1基+SPE6基 SIMD無 -O3有	42.6
PPE1基+SPE6基 SIMD有	36.86
PPE1基+SPE6基 SIMD有 -O3有	383.42

PPE1基のみで計算をした時間を1とし、何倍効率化できたかを表2、表3にまとめた。

表2から、絶対値計算においてPPEとSPEを協調させることでPPEのみで処理を行うよりも大幅に処理効率が向上することが分かる。

6-2. DMA転送を含む絶対値計算の時間比較

先ほどの結果にDMA転送時間を含めたものを図3-1～図3-3に示す。

図3-1はPPE1基のみとPPE1基とSPE6基を協調させた処理も含む5つの計測条件の時間比較である。

図3-2、図3-3についてはPPE1基のみ以外のデータを抜き出してそれぞれSIMD演算の有無について詳細に比較した図である。

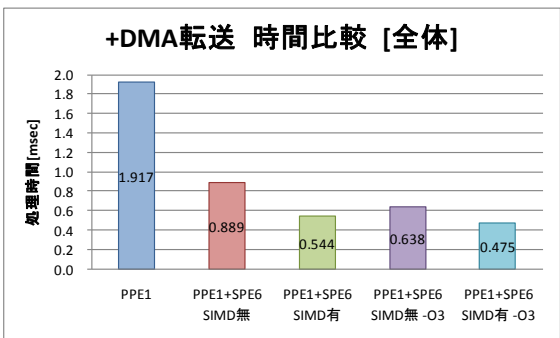


図3-1. DMA転送を含めた比較[全体]

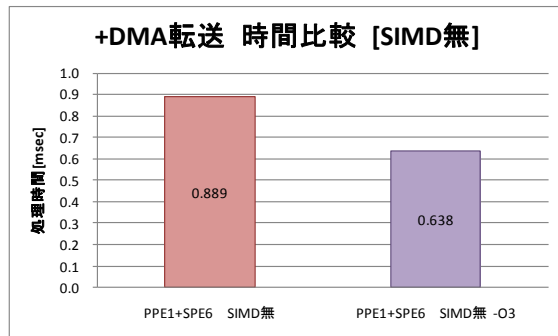


図3-2. DMA転送を含めた比較[SIMD無]

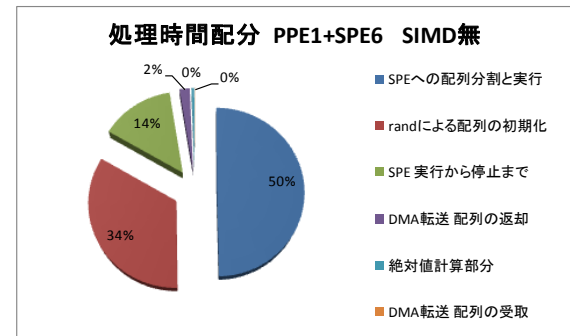


図4-1. 処理時間配分の比較[SIMD無]

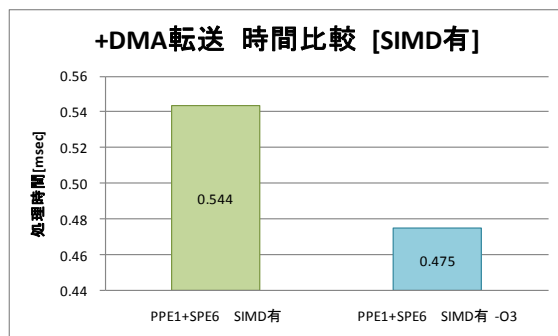


図3-3. DMA転送を含めた比較[SIMD有]

表4-2. 処理時間配分の比較[SIMD無 -O3]

計測項目	処理時間[msec]
SPEへの配列分割と実行	18.964
randによる配列の初期化	8.827
SPE 実行から停止まで	4.898
DMA転送 配列の返却	0.572
絶対値計算部分	0.045
DMA転送 配列の受取	0.022
合計値	33.326

表3. DMA転送時間を含めた処理効率比較

計測条件	処理効率[倍]
PPE1基のみ	1
PPE1基+SPE6基 SIMD無	2.15
PPE1基+SPE6基 SIMD無 -O3有	3
PPE1基+SPE6基 SIMD有	3.52
PPE1基+SPE6基 SIMD有 -O3有	4.03

表3から、処理効率が絶対値計算のみの場合よりも緩やかに上昇していることが分かる。

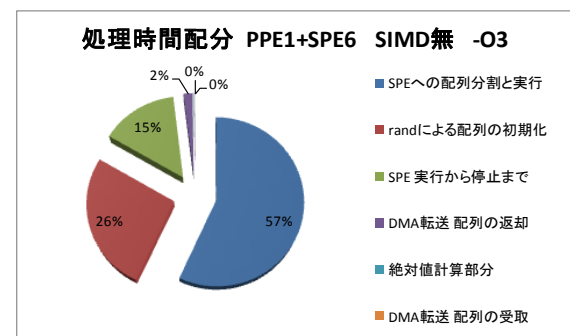


図4-2. 処理時間配分の比較[SIMD無 -O3]

6-3. 処理時間配分比較 SIMD無

処理時間配分の比較を表4-1、表4-2、図4-1、図4-2に示す。これはSIMD無のプログラムの各箇所で処理時間を計測、比較したものである。

表4-1. 処理時間配分の比較[SIMD無]

計測項目	処理時間[msec]
SPEへの配列分割と実行	17.351
randによる配列の初期化	11.734
SPE 実行から停止まで	4.915
DMA転送 配列の返却	0.679
絶対値計算部分	0.189
DMA転送 配列の受取	0.021
合計値	34.889

6-4. 処理時間配分比較 SIMD有

処理時間配分の比較を表4-3、表4-4、図4-3、図4-4に示す。これはSIMD有のプログラムの各箇所で処理時間を計測、比較したものである。

表4-3. 処理時間配分の比較[SIMD有]

計測項目	処理時間[msec]
SPEへの配列分割と実行	18.293
randによる配列の初期化	12.183
SPE 実行から停止まで	5.136
DMA転送 配列の返却	0.472
絶対値計算部分	0.052
DMA転送 配列の受取	0.020
合計値	36.155

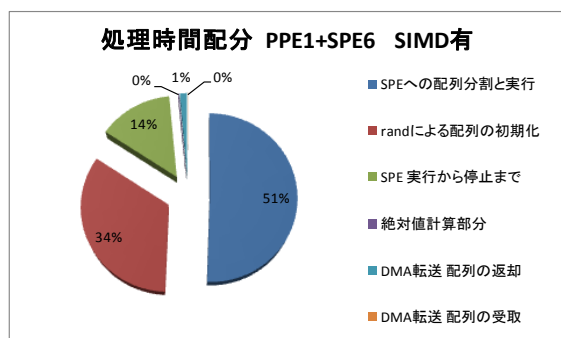


図4-3. 処理時間配分の比較[SIMD有]

表4-4. 処理時間配分の比較[SIMD有 -03]

計測項目	処理時間[msec]
SPEへの配列分割と実行	18.147
randによる配列の初期化	9.648
SPE 実行から停止まで	4.873
DMA転送 配列の返却	0.449
DMA転送 配列の受取	0.022
絶対値計算部分	0.005
合計値	33.143

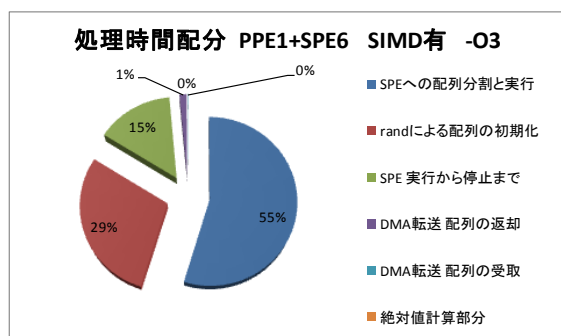


図4-4. 処理時間配分の比較[SIMD有 -03]

表4-1～表4-4、図4-1～図4-4より、処理時間の大部分を占めているのは、どの計測条件の場合も“SPEへの配列分割と実行”や“randによる配列の初期化”など、直接的に計算に関係のない部分であることが分かる。

絶対値計算のみの比較(表2)について全体的に約10倍～約30倍近くの効率化ができ、“PPE1+SPE6 SIMD有 -03”の条件で、PPEのみの時より約380倍の高速化ができた。

また、DMA転送の時間を含めた比較(表3)についてもPPE1基のみで実行させるより全体的に約2～4倍の高速化ができた。

7. おわりに

以上の結果から、取り扱う計算の演算回数が多い場合や時間のかかる計算部分はSIMD化する事で飛躍的な性能向上が期待できるが、DMA転送など直接的に計算に関係のない部分ではSIMD演算の恩恵を受けられないことが分かった。

また、計算処理自体が大規模なものでない、せっかく計算部分を高速化しても結局のところ直接的に計算に関係のない部分の処理時間がプログラム全体の時間の大部分を占めてしまうため、全体の処理効率向上に直接的に結びつきにくいということが言える。

今後の課題としては、計算処理に膨大な時間のかかる大規模な問題の設定や、PS3の台数を増やすことでより計算効率を底上げできないかどうかを検討することである。

参考文献

- [1] Cell Speed Challenge 2009
<http://www.hpcc.jp/sacsis/2009/cell/>
- [2] PS3 LinuxでCellプログラミング！
<http://plaza.rakuten.co.jp/miyazblog/>
- [3] Sλの日記 --うつろな日々--
<http://d.hatena.ne.jp/S-ili/>
- [4] 研究日誌。
http://blog.goo.ne.jp/ulow_cheap/e/cb9acafffb4ab1a4cfb05eae7dad5456
- [5] Young risk taker
<http://rakuto.blogspot.com/2006/11/cell-spe.html>
- [6] FIXSTARS PLAYSTATION®3
<http://cell.fixstars.com/ps3linux/>
- [7] 塩田紳二, 安田絹子, 國司光宣, 平初, 川井義治, 古坂大地, 青柳信吾, 八重樫剛史, PLAYSTATION3 Linux 完全攻略ガイド, インプレスジャパン(2007)