

Cell Be による数値計算プログラムの並列化の一検討

A parallel programming for numerical calculation on Cell Broadband Engine

CS01 青木 宏之
指導教員 島川 陽一

1. はじめに

近年, CPU の単体での性能を向上させることにはコスト的, 発熱量的な面から限界がきており, 更なる性能の向上が期待できる複数の CPU コアを搭載するマルチコア CPU が開発されつつある. しかし, 既存のプログラムが並列処理を行うマルチコアに最適化されているとはいいがたく, 特に Cell Be のように特殊な構成の CPU ではその傾向が顕著である.

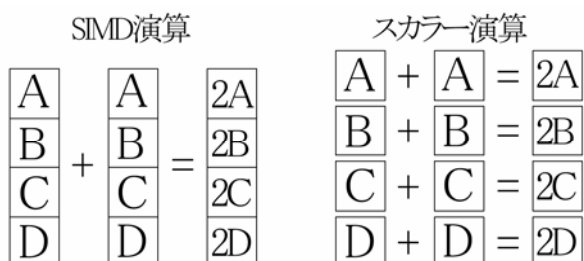
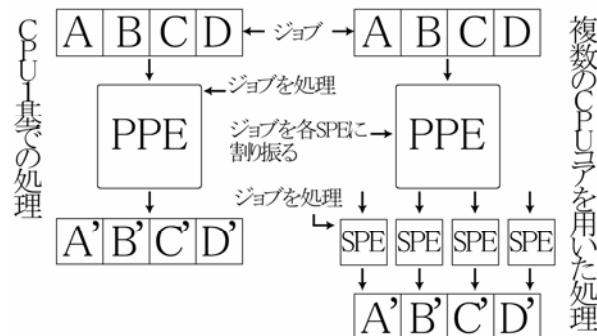
そこで本研究では, Cell Be で並列化が可能なプログラミングを行うことでプログラムの並列化についての知見を得る. 逐次的な処理と比べてどのように速度が向上するかを計測する.

開発環境に PlayStation3 を用いる. OS は Linux (Fedora)を用いる.

2. 並列化を検討するアルゴリズム

Cell プロセッサによるプログラムの並列化は図 1 に示すように Cell に搭載されている CPU コアである 1 基の PPE と 7 基の SPE を利用したハードウェアとしての並列化と, 図 2 に示す複数の命令を一度に行える SIMD 演算を利用した並列化がある.

本研究ではマージソートを適用対象として取り上げ, 並列化の問題点を整理する.



3. 数値実験による考察

図 3 に CPU の種類と数を変えて数値計算を行った場合の計算時間を示す.

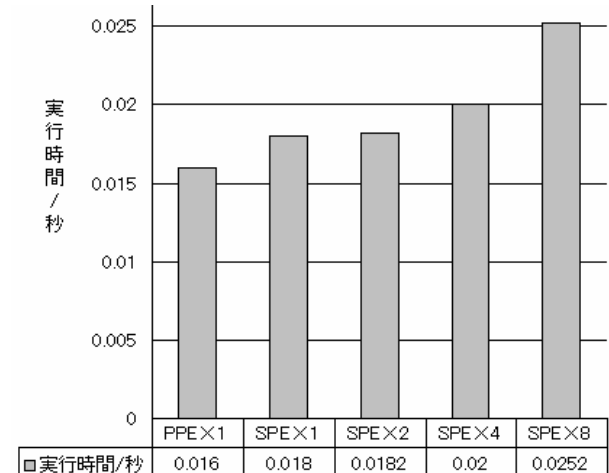


図 3. 実行結果

本来ならば SPE の数が増えるにつれて実行時間は早くなるはずであるが, 逆に実行時間が遅くなった. これは各 SPE へデータとプログラムを転送する処理が各 SPE 内で行われる処理よりも時間がかかったことと, ソートのアルゴリズムにマージソートを用いたためと考えられる. それぞれの SPE から PPE へデータを返すタイミングが重なってしまい, データ転送の際に待ち時間が発生した.

4. おわりに

本研究ではマージソートを適用対象とした並列化を試み, 並列プログラムの知見を得た. プログラムの並列化においては計算対象の規模とデータの転送にかかる手間を考慮しないと計算速度の向上は望めないことが分かった.

本研究の計算では各 SPE の処理結果が返るタイミングを意図的にずらし, また処理するデータが大規模で SPE 内で行う処理にもっと時間がかかるプログラムなら逐次的なプログラムと比べ高速になると考えられる. しかし上記の条件においても SPE へのデータとプログラムのロードにかかる時間は無視できない. データ転送のタイミングも重要な要因である.

文献

- [1] Cell Broadband Engine トータル・ソリューション・カンパニー - フィックスターズ <http://www.fixstars.com/>