All-Pairs Shortest Path Problem on the PLAYSTATION 3

松 本 和 也 † スタニスラフ セドゥーキン†

グラフの全ての最短経路を探し出す問題は最も基本的なグラフ理論の問題の一つであり，実世界においてバイオインフォマティクス，ネットワーク・ルーティング，CAD などの多くの応用例が考えられる．全対最短経路問題 (All-Pairs Shortest Path Problem or APSP) は n 個のノードを持つ重みつきグラフの全てのペアの最短経路の長さを求める問題である．この問題はフロイド・ワーシャル (FW) のアルゴリズムを用いて解くことができる．しかし FW のアルゴリズムはデータ依存関係のあるアクセスパターンを含んでいるために，単純にアルゴリズムを移植しただけでは高い性能を発揮するのが難しい．それゆえにそのアルゴリズムをブロック化することで高速化を図ろうという試みがなされてきた [1] [2] [3]．プレイステーション 3 (PS3) に搭載されている Cell Broadband Engine (Cell) の Synergistic Processor Element (SPE) はそのコアエレメント同士がメモリ領域を共有しないという特徴から，問題をブロック化することで，その性能を最大限に発揮することができるようになる．そこで私たちはブロック化された全対最短経路問題のアルゴリズムを Cell 上に実装し，その性能を測定した．

全対最短経路問題

$G = (V, E, w)$ を重みつき有効グラフとする．ここに頂点 $v \in V = \{1, \cdots, n\}$，辺 $(i, j) \in E \subset V \times V$，そして重み関数 $w : E \to R_+$; $w(i, j)$ は辺 (i, j) の距離あるいは重みとする．グラフ G は以下の $d(i, j)$ による $n \times n$ 行列 $D = [d(i, j)]$ で表すことができる．

$$d(i,j) = \begin{cases} 0 & \text{if } i = j; \\ w(i,j) & \text{if } i \neq j \text{ and } (i,j) \in E; \\ \infty & \text{if } i \neq j \text{ and } (i,j) \notin E. \end{cases}$$

そこで点 i から点 j への最短距離を

$$d^*(i,j) = \min_{P \in \text{paths}(i,j)} \sum_{(u,v) \in P} w(u,v)$$

† 会津大学
The University of Aizu

としたとき，$n \times n$ 行列 $D^* = [d^*(i, j)]$ の計算を要求する問題を全対最短経路問題と呼ぶ．

全対最短経路問題は FW のアルゴリズムを用いて n^3 回の $(\min, +)$ 演算で解くことができるが，ここでは対角線上の要素の重みを 0 としているので，図 1 のアルゴリズムのように，$i = k, j = k$ の場合の不要な計算を省くことができる． このアルゴリズムの

for $k = 1 : n$
  for all $i = 1 : n$ $(i \neq k)$
    for all $j = 1 : n$ $(j \neq k)$
      $D_{ij}^{(k)} = \min\left(D_{ij}^{(k-1)}, D_{ik}^{(k-1)} + D_{kj}^{(k-1)}\right)$;

図 1 Scalar APSP algorithm

$(\min, +)$ 演算の合計数は式 (1) のようになる。

$$S_{(\min,+)}^{\text{scalar}} = n^3 - 2n^2 + n = n(n-1)^2 \qquad (1)$$

ブロック化された全対最短経路問題のアルゴリズム

図 2 にブロック化された全対最短経路問題のアルゴリズムを示す．このアルゴリズムは $n \times n$ 行列 D の問題を $N \times N$ ブロック $D_{IJ}^{(0)}$ の問題へと分割することを基としている．ここに N は b をブロックサイズとしたときの n/b の値であり，グラフのサイズ n は b の倍数のみと想定する．このことは問題の一般性を失われるせるものではない．

$K = 1, 2, \cdots, N$ の各々の場合において，図 2 のブロック化されたアルゴリズムは始めに図 1 のスカラーアルゴリズムを用いて black ブロックを更新する (D_{KK}，ライン 3)．その後，$(N-1)$ 個の red ブロック (D_{IK}，ライン 6) と $(N-1)$ 個の blue ブロック (D_{KJ}，ライン 9) をそれぞれ更新する．最後に $(N-1)^2$ 個の white ブロックを更新する (D_{IJ}，ライン 13)．重要な事実としては，すべての red，blue，white ブロックの更新は行列の乗加算演算 ($C = A \otimes B \oplus C$ の形をとる演算) に tropical semiring とも呼ばれる $(\min, +)$-algebra を適用したものだということである．この事

実はループ・インターチャンジ，ループ・アンローリングなどといった存在する全ての行列乗算の最適化手法をそれらのブロックの更新部を最適化するのに用いることができるということを意味する．

1 **for** $K = 1 : N$ **do**
2 *% Black block update*
3 $D_{KK}^{(K)} = \left(D_{KK}^{(K-1)}\right)^{*}$;
4 *% Red blocks update*
5 **for all** $I = 1 : N\ (I \neq K)$
6 $D_{IK}^{(K)} = \min\left(D_{IK}^{(K-1)}, D_{IK}^{(K-1)} + D_{KK}^{(K)}\right)$;
7 *% Blue blocks update*
8 **for all** $I = 1 : N\ (J \neq K)$
9 $D_{KJ}^{(K)} = \min\left(D_{KJ}^{(K-1)}, D_{KK}^{(K)} + D_{KJ}^{(K-1)}\right)$;
10 *% White blocks update*
11 **for all** $I = 1 : N\ (I \neq K)$
12 **for all** $J = 1 : N\ (J \neq K)$
13 $D_{IJ}^{(K)} = \min\left(D_{IJ}^{(K-1)}, D_{IK}^{(K)} + D_{KJ}^{(K)}\right)$;
14 **end**

図 2 Blocked APSP algorithm

図 2 からわかるように，それぞれのブロックの更新に必要な行列の数は異なる．black ブロックの更新には 1 つの $b \times b$ 行列しか必要としないが，red, blue ブロックの更新にはそれぞれ 2 つの行列，white ブロックの更新には 3 つの行列が必要とされる．このブロック化されたアルゴリズムの $(\min, +)$ 演算の合計数は，

$$S_{(\min,+)}^{\text{block}} = n(n^2 - 2b + 1) \qquad (2)$$

である．式 (1) と式 (2) を比べると，ブロック化されたアルゴリズムのほうがスカラーのものより多くの演算数を必要とすることがわかる．その演算数の差は

$$\Delta = S_{(\min,+)}^{\text{block}} - S_{(\min,+)}^{\text{scalar}} = 2n(n - b)$$

である．この差は毎回の red,blue ブロックを更新する際に b^3 回の演算を行うので，その 1 回のブロック更新ごとにスカラーアルゴリズムよりも b^2 回だけ多くの演算を行っているために表れる．N が大きい場合，このブロック化されたアルゴリズムはその計算時間のほとんどを red,blue,white ブロックの更新に費やす．

実 装 結 果

図 3 に 64 刻みのサイズのグラフにおいて，異なる数の Cell の SPE を用いた場合における PS3 でのブロック化された全対最短経路問題のアルゴリズムの性能測定結果を示す．そして図 4 には私たちの PS3 での全対最短経路問題の性能と他のいくつかのプラットフォームでの性能との比較を示す．図からもわかるように，PS3 の Cell プロセッサは全対最短経路問題において，最大で 50.5 Gflop/s という強力なパフォーマンスを発揮する．

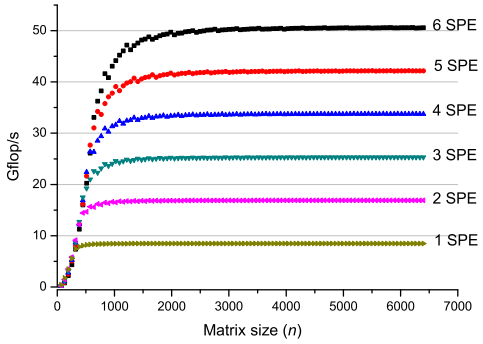

図 3 PS3 performance on the blocked APSP algorithm

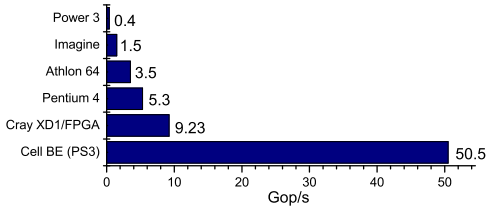

図 4 Performance comparison of the APSP algorithm. Sources: Cray XD1/FPGA [1]. Pentium 4, Athlon 64 [4]. Imagine, Power 3 [2].

参 考 文 献

1) Bondhugula, U. et al.: Hardware/Software Integration for FPGA-based All-Pairs Shortest-Paths on a Reconfigurable Supercomputer, *Technical Report OSU-CISRC-1/06–TR13* (2006).
2) Griem, G. and Oliker, L.: Transitive Closure on the Imagine Stream Processor, *Proc. of MPS-5* (2003).
3) Takahashi, A. and Sedukhin, S.: Parallel Blocked Algorithm for Solving the Algebraic Path Problem on a Matrix Processor, *Proc. of HPCC 2005, LNCS 3276*, pp. 789–795 (2005).
4) Han, S.-C., Franchetti, F. and Püschel, M.: Program Generation for the All-Pairs Shortest Path Problem, *Proc. of PACT-15* (2006).