1P−6

超並列計算機の実用並列計算モデルLogPQと数値計算の実行評価

當山 孝義　堀口 進
北陸先端科学技術大学院大学情報科学研究科

1. はじめに

並列計算機の登場以来，PRAMモデルに現実性を考慮した制約を付加した様々な並列計算モデルが提案されてきた．近年，Cullerらは，より実用的な並列計算モデルとしてLogPモデル(1)を提案した．

筆者らは，実用的観点からLogPモデルを検討し，その改良としてLogPQモデルを提案した．

そして，並列多倍長GCDアルゴリズムの検討と，LogPQエミュレータによる動作解析を行った．

2.LogPQモデル

LogPモデルは，RAMを通信路で接続したものであり，共有メモリを持たない．プロセッサ間通信はメッセージパシングにより行われる．通信コストは3パラメータで表わす．すなわち，通信命令の実行時間(overhead:o)，通信命令の実行可能な間隔(gap:g)，通信路における遅延(latency:L)である．また，プロセッサ数はPである．

LogPモデルの特長は，この4パラメータを実際の計算機構造に合わせて設定することにより，各種並列計算機の動作をモデル化できることである．

LogPQモデルは，LogPの通信路にキューを付加し詳細化したものである．図1にLogPQモデルを示す．プロセッサには送信キューおよび受信キュー，通信路には転送キューを導入している．各キュー長の上限として，$SendQ$，$RecQ$，$TransQ$の3パラメータを追加する．図のtransfer rateは通信路の転送レートを，transfer latencyおよびmacro capacityは通信の集中がない場合の入出力プロセッサ間の遅延および通信路容量を示す．Lは送信命令開始時から受信までの時間，oは通信命令の実行時間，gはデータ送信間隔を示す．

LogPQパラメータは，LogPモデルのパラメータと同様であるが，通信命令開始時が起点なのでアルゴリズムを検討しやすい．データ通信は1ワード通信を基準とし，複数ワードのデータ通信は，この組み合せで表わす．パラメータPはLogPモデルと同じくプロセッサ数である．

3. 並列多倍長GCD計算

Practical parallel computation model LogPQ and evaluation of a numerical computation, Takayoshi Touyama and Susumu Horiguchi, Graduate School of Information Science, Japan Advanced Institute of Science and Technology

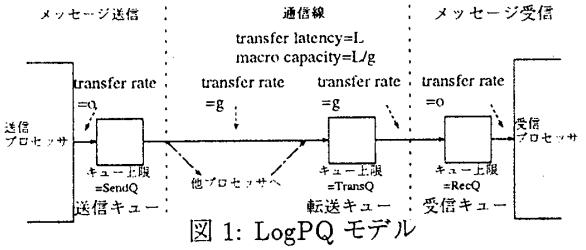

図1: LogPQモデル

1. $\delta := 0$
2. 以下の処理を$b = 0$になるまで繰り返し実行する。
   (1) b:evenの間、以下の保存変換を繰り返す。
   $gcd(a,b) := gcd(a,b/2), \delta := \delta + 1$
   (2) $\delta > 0$ならば、以下の変換を実行する。
   $gcd(a,b) := gcd(b,a), \delta := -\delta$
   (3) 以下の保存変換を実行する。
   ・ $(b+a)/2$: $even$ の場合
   $gcd(a,b) := gcd(a,(b+a)/2)$
   ・ $(b-a)/2$: $even$ の場合
   $gcd(a,b) := gcd(a,(b-a)/2)$
3. GCD:=$|a|$

図2: Brent-Kungアルゴリズム

図2に，実用性の高い逐次アルゴリズムであるBrent-Kung(BK)アルゴリズム(3)を示す。BKアルゴリズムは，保存変換の$O(n)$回繰り返しでGCDを求める．

パイプライン計算は，多倍長整数の各ワード各a_iをプロセッサP_iに割り当て，p個のプロセッサによる並列計算を行うものである．プロセッサ間同期を，プロセッサP_0からP_{p-1}に向けて線型にとることにより，パイプライン動作をさせる．P_iはP_{i-1}からの通信により動作を開始しP_{i+1}に通信を送る．キャリー処理は隣接プロセッサとの通信により行う．

パイプライン計算は，冗長性を付加した表記を用いることにより，キャリー伝搬による遅延の影響を避けることができる．本稿では，Signed Digit表記(4)を拡張しキャリー処理回数を削減した拡張SD法を用いる．多倍長整数は，$a = \Sigma a_i 2^{li}$で表わされる．ただし，$h > l$, $-2^h \leq a_i < 2^h$である．

並列多倍長GCDアルゴリズムは，保存変換フェーズ実行後，終了処理フェーズを実行する．保存変換フェーズは，保存変換ステップを$b = 0$になるまで繰り返す．終了処理フェーズは，aをバイナリ表記に変換しその絶対値を求める．保存変換ステップの実行時間を大きくするために，P_0でBK保存変換算出をk回繰り返し，この複合であるk変換(3)の，正負反転をk変換行列と

する．この正負反転は$b=0$比較の為である．$b=0$ならば，最大2回の(-1)乗算実行で$\wedge(b_i=0)$となる．また，多ワード複合として，$P_i(0<i<p-1)$にはmワード，P_{p-1}には$m+1$ワードを割り当てる．

図3に保存変換フェーズ動作を示す．保存変換ステップは，以下のように実行される．P_0は，k変換行列(c,d,e,f)を算出するとともにa_0,b_0の積和演算を行う．$P_j(0<j<p-1)$は，以下を実行する．P_{p-1}は同様であるが，最初に$b=0(\equiv Z_{p-1})$比較を行う．

最初に上方向通信を行う．すなわち，P_{j-1}よりデータ受信後，以下をP_{j+1}へ送信する．同期信号，P_j以下のプロセッサに割り当てられるb_iの全てが0であるかどうかを示すフラグZ_i，上キャリーn_a,n_bおよびk変換行列．次に，P_jに割当てられた各ワードa_i，b_iの積和演算を行う．図4に各ワードの積和演算を示す．最後に，P_jに割当てられたb_iに対する$u=\wedge(b_i=0)$を求める．次の保存変換ステップの開始時点において，P_{j+1}に伝達する$n_a,n_b(P_j$に割当てられた最左ワードの$n_{a_i},n_{b_i})$，$Z_i(=Z_{i-1}\wedge u)$は直ちに求まる．

パイプライン計算において，多倍長表記はプロセッサのワード長w_pおよび通信路のワード長w_cで制約される．本計算では，$w_p\geq l+k+3, w_c\geq\max(h-l,k)$，$2l\geq h\geq l+3, l\geq 2k+1$である．

遅延部分は，保存変換フェーズにおけるキャリー処理による遅延と，保存変換フェーズにおけるZ_i伝播遅延時間と終了処理フェーズ実行時間の和がある．前者は保存変換の実行回数に比例するので定率遅延，後者は一定なので定量遅延と呼ぶ．

保存変換フェーズにおいて，保存変換ステップ実行時間が大きい場合，下キャリー待ちは生じない．この条件は，多倍長計算の実行時間をt_m，保存変換算出の実行時間をt_tとし，データ通信が上方向はl_uワード，下方向はl_dワードとすると，P_1以降の隣接プロセッサ間の関係より$(L+(l_u-1)g)+l_u\cdot o+t_m+(L+(l_d-1)g)\leq l_u\cdot o+(m-1)t_m+l_d\cdot o$，$P_0$と$P_1$間の関係より$(L+(l_u-1)g)+l_u\cdot o+t_m+(L+(l_d-1)g)\leq l_u\cdot o+k\cdot t_t+l_d\cdot o$である．また，$P_i(i\geq 1)$の保存変換ステップ実行時間が$P_0$より大きいと，$P_0$に待ちが生じる．これは$m$の上限を規定する．これらは，定率遅延に影響している．以上より，所与のLogPQパラメータに対するm,kの適当な範囲が得られる．

拡張SD法を用いる利点は，キャリー伝播の削除による定量遅延の削減である．n_a,n_b,Z_iは，バイナリ表記では保存変換ステップが終了するまで得られなかったのが，拡張SD法では保存変換ステップ開始時に得

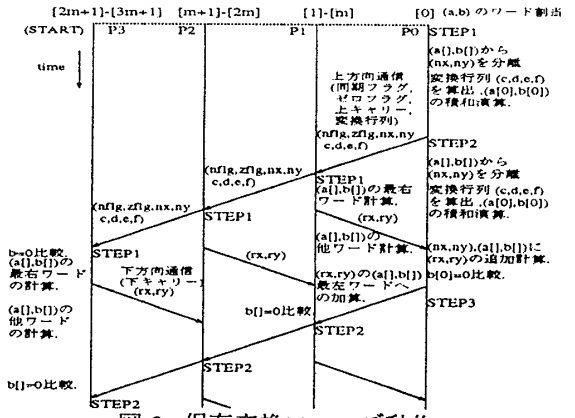

図 3: 保存変換フェーズ動作

1. $n_{a_i}=(a_i$の左$h-l$ビット$)$
2. $a_i=$(符号付き整数$n_{a_{i-1}}$)$+(a_i$の右lビット$)$
   $(b_i$も同様$)$
3. $a_i=a_i\cdot c+b_i\cdot e$
4. $r_{a_i}=(a_i$の右kビット$)$
5. $a_i=(a_i$をkビット右シフト$)$
   $+(r_{a_{i+1}}$を$l-k$ビット左シフト$)$
   $(b_i$も同様$)$

図 4: 多倍長計算

られる．従って，Z_iのP_0からP_{p-1}への伝播時間は保存変換ステップ実行時間の$p-1$倍だけ減少する．プロセッサ間同期を線型でなく木状にとることにより，さらに定量遅延時間の削減が見込まれる．

4. LogPQ エミュレータによる実行評価

Hämäläinen が開発し公開しているPRAMエミュレータ(2)を基に開発した，LogPQエミュレータを用いて，並列多倍長GCDプログラムの動作解析を行った．32ビットプロセッサを想定すると，たとえば多倍長表記は$h=23, l=20$，k変換複合は$k=8$となる．

5. まとめ

本稿ではLogPQモデルを示し，LogPQモデルを用いて並列多倍長GCDアルゴリズムの検討を行った．特に，遅延の影響について，詳細に検討した．

この検討は，他アルゴリズムにも適用可能と思われるが，今後の課題である．

参考文献

(1) D.Culler etc., 'LogP:Towards a Realistic Model of Parallel Computation", Proceeding of the 4th ACM SIGPLAN Symposium on Principles and Parallel Programming (1993).

(2) P.Hämäläinen, "PRAM EMULATOR", University of Joensuu, Department of Computer Science, Joensuu, Finland (1993).

(3) B.Chor etc., "An Improved Parallel Algorithm for Integer GCD", Algorithmica, 5, pp.1-10 (1990).

(4) 亀山充隆 他, "Signed-Digit 数系に基づく双方向電流モード多値基本演算回路とその評価", 信学論 D, Vol.J71-D No.7, pp.1189-1198 (1988).