応用数理工学特論　2006/06/01 講義ノート

担当：計算理工学専攻　学籍番号 280667264　渡邉　航
計算理工学専攻　学籍番号 280667205　松野 善幸

3.5　分散メモリ型計算機上での並列化

ブロックサイクリック列分割による並列化

行列にブロックサイクリック列分割を施した図を **Fig.1** に示す.

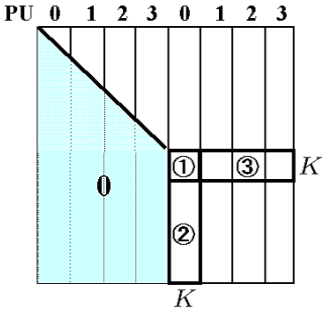

Fig. 1 ブロックサイクリック列分割 (PU 数 4 の例)

また，このとき
n:行列サイズ
b:ブロックの幅
N:ブロック単位で見たサイズ ($=\frac{n}{b}$)
P:プロセッサ数
Fig.1 の 1(○で囲まれた数字の 1):A_{KK}
Fig.1 の 2(○で囲まれた数字の 2):L_{IK}(ピボット列)
Fig.1 の 3(○で囲まれた数字の 3):U_{KJ}(ピボット行)
である．ここで，第 K ブロック段で行われる処理を以下に列挙する.

第 K ブロック段での処理

1. 第 k 列を持つプロセッサ (MOD$(K-1,P)$) は
   $A_{KK}=L_{KK}U_{KK}$ を計算 (LU 分解).
2. 第 k 列を持つプロセッサは
   $L_{IK}=A_{IK}U_{KK}^{-1}(I=K+1,\cdots,N)$ を計算.
3. 第 K 列を持つプロセッサは $L_{IK}(I=K,\cdots,N)$ を全プロセッサに送信 (ブロードキャスト).
4. 各プロセッサは L_{IK} を受け取る.
5. 各プロセッサは自分の担当する列について
   $U_{KJ}=L_{KK}^{-1}A_{KJ}$ を計算.
6. 各プロセッサは自分の担当する列について
   $A_{IJ}:=A_{IJ}-L_{IK}U_{KJ}$ を計算.

並列化の効果

- **演算量に関して**

$$\frac{N}{P}\gg 1 \quad (\frac{n}{bP}\gg 1)$$

上記の式を満たすならば負荷は均等になる．負荷が均等な条件下において，1 ブロック段・1 ブロックあたりの演算量は **Fig.2** より

$$\frac{O(n^2\cdot b)}{P}=O(\frac{n^2\cdot b}{P})$$

となる.

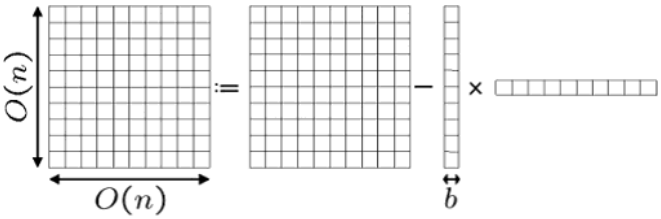

Fig. 2 1 ブロック段・1 プロセッサあたりの演算量

- **通信量に関して**

1 ブロック段で送るデータ＝ブロックピボット列であるため，1 回の通信量は

$$O(n)\times b=O(nb)$$

また，P 個のプロセッサの場合には $\log_2 P$ 段の通信が必要となる (**Fig.3**).

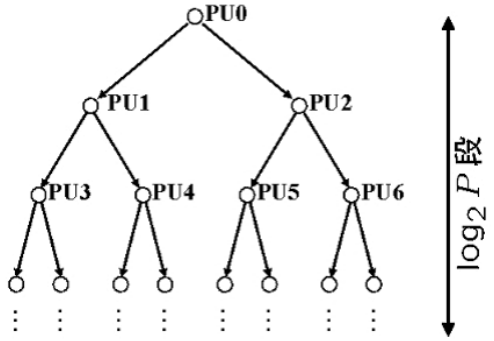

Fig. 3 ブロードキャストの通信パターン

これより，通信量は

$$O(nb)\times\log_2 P=O(nb\cdot\log_2 P)$$

となる．

これら演算時間，通信時間およびその合計である全実行時間をグラフに示すと **Fig.4** のようになる．**Fig.4** を見る

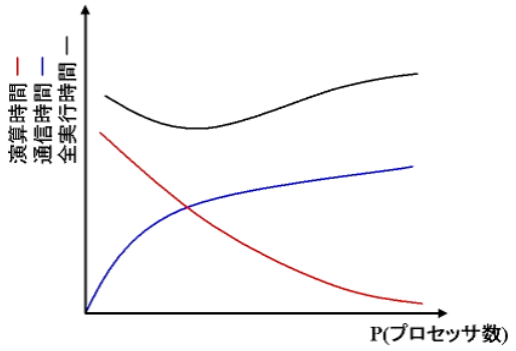

Fig. 4 演算時間，通信時間および全実行時間

と，あるプロセッサ数 P で全実行時間が最小値をとり，その数を超えると P が大きくなるごとに全実行時間は増大する (＝性能が低下する) ことがわかる．また P が大きくなりすぎると，負荷が均等になる条件 $\frac{N}{P} \gg 1$ が成り立たないといった問題も浮上する．

このような問題を解決するための方法として，分割方式を列分割から双方向分割に変える方法が提案された．

双方向のブロックサイクリック分割による並列化

行列に双方向のブロックサイクリック分割を施した図を **Fig.5** に示す．

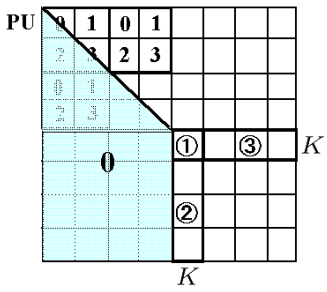

Fig. 5 双方向のブロックサイクリック分割 (PU 数 4 の例)

このとき
Fig.1 の 1(○で囲まれた数字の 1):A_{KK}
Fig.1 の 2(○で囲まれた数字の 2):L_{IK}(ピボット列)
Fig.1 の 3(○で囲まれた数字の 3):U_{KJ}(ピボット行)
である．ここで，第 K ブロック段で行われる処理を以下に列挙する．

第 K ブロック段での処理

1. A_{KK} を持つ PU(プロセッサ) は $A_{KK} = L_{KK}U_{KK}$ と LU 分解.
2. A_{KK} を持つ PU は U_{KK} を縦方向にブロードキャスト.
3. 第 K 列担当の PU は自分の担当するブロックについて $L_{IK} = A_{IK}U_{KK}^{-1}$ を計算.
4. A_{KK} を持つ PU は L_{KK} を横方向にブロードキャスト.
5. 第 K 行担当の PU は自分の担当するブロックについて $U_{KJ} = L_{KK}^{-1}A_{KJ}$ を計算.
6. 第 K 列担当の PU は自分の持つ L_{IK} を横方向にブロードキャスト.
7. 第 K 行担当の PU は自分の持つ U_{KJ} を縦方向にブロードキャスト.
8. 各 PU は自分の担当するブロックについて $A_{IJ} := A_{IJ} - L_{IK}U_{KJ}$ を計算.

並列化の効果

■演算量 (1 ブロックあたり)
負荷が十分均等 ($\frac{n}{b\sqrt{P}} \gg 1$) ならば

$$o(\frac{n^2 b}{P})$$

上記の演算量はブロックサイクリック列分割の演算量と等しい．

■通信量

ブロードキャストする回数は 4 回あるが，$L_{\mathrm{KK}}, U_{\mathrm{KK}}$ はブロックに関する情報であり $L_{\mathrm{IK}}, U_{\mathrm{KJ}}$ のブロック列・行に関する情報に比べ小さいため無視する．

- データ量（1 プロセッサあたり）

$$o(\frac{nb}{\sqrt{P}})$$

- 送り先の数

$$\sqrt{P}\text{個}$$

  ∗ 送り回数は $\log_2 \sqrt{P}$ となる．

- 通信時間

$$o(\frac{nb}{\sqrt{P}}) \times \log_2 \sqrt{P} = o(nb\frac{\log_2 P}{\sqrt{P}}) \to 0 \quad (P \to \infty)$$

  cf) ブロックサイクリック分割における通信時間は $o(nb\log_2 P)$ である．

Fig.6 に演算時間，通信時間および全実行時間の関係を示す．プロセッサ数が大きくなると全実行時間は減少することがわかる．

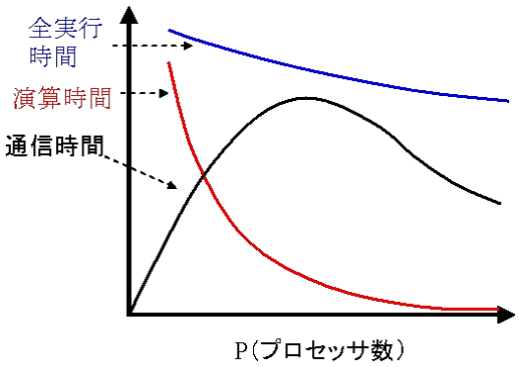

Fig. 6 演算時間，通信時間および全実行時間の関係

通信の隠蔽による高速化 (演算と通信のオーバーラップ)

- 第 K 列を持つ PU は
  $A_{\mathrm{KK}} = L_{\mathrm{KK}}U_{\mathrm{KK}}$ と LU 分解.
- 第 K 列を持つ PU は
  $L_{\mathrm{IK}} = A_{\mathrm{IK}}U_{\mathrm{KK}}^{-1}(I = K+1, ..., N)$ を計算.
- 第 K 列を持つ PU は
  $U_{\mathrm{K,K+1}} = L_{\mathrm{KK}}^{-1}A_{\mathrm{K,K+1}} \quad (I = K+1, ..., N)$ を計算.
  - $A_{\mathrm{K+1,K+1}} = L_{\mathrm{K+1,K+1}}, U_{\mathrm{K+1,K+1}}$ と LU 分解 (次の対角ブロックの LU 分解).
  - $L_{\mathrm{I,K+1}} = A_{\mathrm{I,K+1}}U_{\mathrm{K+1,K+1}}^{-1}(I = K+2, ..., N)$ (次のブロックピポッド列の計算) .
  - $L_{\mathrm{I,K+1}}(I = K+1, ..., N)$ をブロードキャスト (次のブロックピポッド列のブロードキャスト) .
  - 第 $K+1$ 列以外に対する処理を行う.

Fig.7 に演算と通信のオーバーラップの関係を示す．**Fig.8**

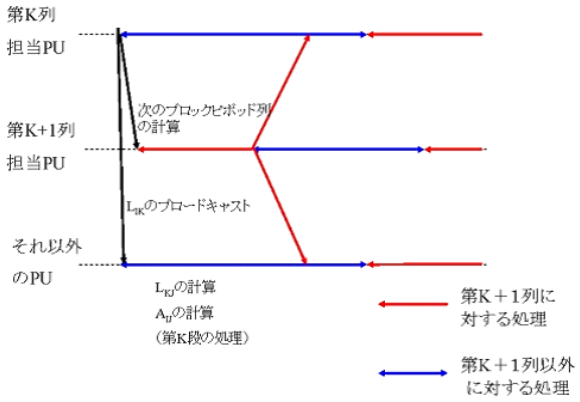

Fig. 7 演算と通信のオーバーラップの関係

に通信の隠蔽後のプロセッサ数と演算量の関係を示す．通信の隠蔽による高速化によりピーク性能に近い性能が出ていることが分かる.

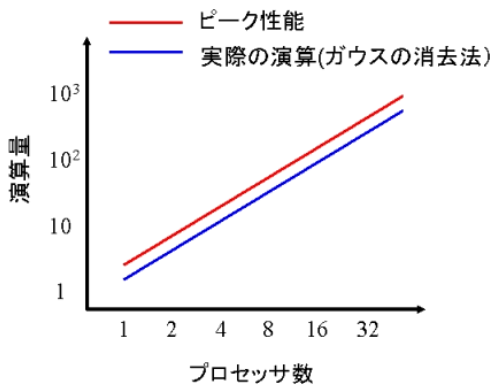

Fig. 8 プロセッサ数と演算量の関係