4B−8　ブロッキングLU分解法のVP2000シリーズ向けチューニングについて

○中西　誠，三上　次郎
富士通　(株)

【1】はじめに

連立1次方程式の解法の一つである外積型のガウス消去法を，ブロック化したブロッキングLU分解法を，VP2000シリーズの単一プロセッサ向けに開発した．富士通のスーパコンピュータVP2000シリーズのハードウェアの特性を活かしたチューニングを行い，高性能を実現したのでその方法を報告する．

【2】外積型のガウス消去法

n×nの正則な実行列は通常，部分ピボッティングによる行の入れ替えを行って，下三角行列Lと単位上三角行列の積に分解することができる．

$$PA=LU$$

ただし，Pは部分ピボッティングによる行の入れ替えを行う置換行列であり．以下の操作で$A=(a_{ij})$を変形して$L=(l_{ij})$と$U=(u_{ij})$に分解する．

$$A=A^{(1)}\rightarrow,\cdots,\rightarrow A^{(k)}\rightarrow,\cdots,\rightarrow A^{(n)}$$

$$u_{kj}=a_{kj}^{(k)}/a_{kk}^{(k)},\quad j=k,\cdots,n \qquad (1)$$

$$l_{ik}=a_{ik}^{(k)},\quad i=k,\cdots,n \qquad (2)$$

$$a_{ij}^{(k+1)}=a_{ij}^{(k)}-l_{ik}\times u_{kj},\quad j=k+1,\cdots,n,\ i=k+1,\cdots,n \qquad (3)$$

実際には，行を均衡化した部分ピボッティングによる行の入れ替えを行う．

(3) は(2) の列ベクトルと(1) の行ベクトルの外積を作り残りの部分を更新している．

【3】ブロッキングLU分解法

以上の外積型のガウス消去法は，列ベクトル，行ベクトル毎に更新を行い，ベクトルの外積を作り更新している．ブロッキングLU分解法では，等ブロック幅b1で列と行方向に分解して，分解した行列の積を作り，更新部分を更新する．

k番目の処理で，ブロック化した外積型のガウス消去法を行うときに，列マトリックスを$L_2^{(k)}$，行マトリックスを$U_2^{(k)}$，更新部分を$M^{(k)}$とする．各マトリックスの配置は図−1を参照．

(3) に対応する更新を(4) で行う．

$$M^{(k)}=M^{(k)}-L_2^{(k)}\cdot U_2^{(k)} \qquad (4)$$

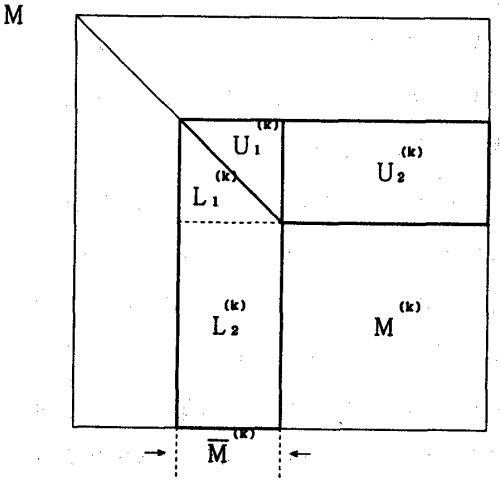
これに先立ち$L_2^{(k)}$，$U_2^{(k)}$は以下の式で更新する．

まず，$\overline{M}^{(k)}$を$(L_1^{(k)T},\ L_2^{(k)T})^T$と$U_1^{(k)}$に分解して，つぎに$U_2^{(k)}$を更新する．

$$\overline{M}^{(k)}=(L_1^{(k)T},\ L_2^{(k)T})^T\cdot U_1^{(k)} \qquad (5)$$

$$U_2^{(k)}=(L_1^{(k)-1})\cdot U_2^{(k)} \qquad (6)$$

結果として，外積型のガウス消去法と同じことを行っている．



図−1　ブロック化した配列Mの各要素の配置関係

この方法はグラニュラリティが大きくなるため，I.S. DUFF達により，マルチプロセッサへの適用の結果も報告されている．

【4】チューニングの課題

On Tuning of Blocking LU decomposition for VP2000 series
Makoto Nakanishi , Jiro Mikami
FUJITSU,Ltd.

ハードの性能を引き出すには以下のことが要件である.

① 演算密度とデータ供給量のバランス

② ベクトル長をなるべく長くする

③ データのアクセス時のバンクコンフリクトの回避

【5】ブロック化とチューニングの方法

富士通のベクトル計算機ＶＰ２０００シリーズには,スカラとベクトルの積とその中間結果ベクトルとベクトルの加算を１命令で行うＭｕｌｔ　＆　Ａｄｄ命令があり,高速化を行っている.

①の問題の演算密度とデータ供給量のバランスをとるために,共通に使われるデータをベクトルレジスタに保持して,演算を行うことを考える.演算量が増えるほど有効であるため,ブロック化して,演算量を増すことを考えた.つまりはブロック幅が大きいほど効果がでる.

ただし,②の観点からは,ブロック幅が大きくなれば,外積型のガウス消去法で式(3)で更新を行うよりも式(4)で更新するときのほうがベクトル長が短くなってしまう.

このため,この二つの要因が釣り合う最適なブロック幅がある.この値は,測定によって求めることができる.

【6】2本のロードパイプによるバンクコンフリクト

③に関しては,１命令でメモリをベクトルレジスタにロードするときロード内で発生するバンクコンフリクトが問題として扱われてきた.

最もコストの大きい式(4)の行列積を実行するとき,２本のロードパイプが密に詰ることにより,ロード命令の切り替わり部分で,バンクコンフリクトが発生する.

更新する行列は,ｋが変化するたびに,ブロック幅に対応するオフセットが変化し,２次元配列宣言で１次元目の値を調整しても,ロードが切り替わる部分で先行する２本のロードの最後の部分と,後続する２本のロードの先頭部分のメモリが同一のバンクになることを完全に避けれない.

【7】行列積のチューニング

式(4)による１×ｍとｍ×ｎの行列（ｍはブロック幅ｂ１）の積で,更新する列ベクトルをレジスタに保持し,更に同時に複数列（２本ないしは４本）更新を行う.

同時に更新をかけると,更新のために参照するデータを共通に使え,演算に対する参照する列ベクトルのロードの回数が減り,バンクコンフリクトを避けることができる.

【8】測定結果

行列積の部分の計算を１列更新と複数列同時更新したもによる性能を行列の次元数が１０００のものを解いたときの性能で示す.演算数は$2/3n^3+2n^2$で計算した.

また最適なブロック幅は,測定から最適値を決めた.

計算機	最大性能 (MFLOPS)	１列更新 (MFLOPS)	複数列同時更新 (MFLOPS)	ブロック幅
ＶＰ２６００ モデル１０	５０００	３６１１	４００１	１２４
ＶＰ２４００ モデル１０	２０００	１５２４	１６８０	１３６

【9】考察

データのレジスタ保持の考えから外積型のガウス消去法をブロック化したが,チューニングの結果高性能を達成した.ブロック化方式は,マルチプロセッサばかりでなく,単一プロセッサでも有効であると考える.

【参考文献】

(1) P.AMESTOY, M.DAYDE and I.DUFF, "Use of computational kernels in the solution of full and sparse linear equations", M, COSNARD, Y.ROBERT, P.QUINTON and M.RAYNAL, PARALLEL & DISTRIBUTED ALGORITHMS, North-Holland, 1989, pp13-19

(2) 島崎眞昭, スーパーコンピュータとプログラミング, 共立出版株式会社, 1989