高性能コンピューティング特論 講義メモ(3)

「並列数値計算の基礎」

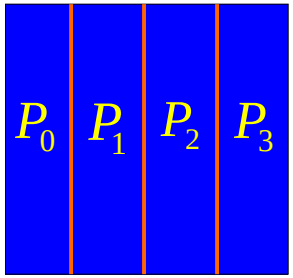
高橋大介
daisuke@cs.tsukuba.ac.jp
筑波大学大学院システム情報工学研究科
計算科学研究センター

講義内容

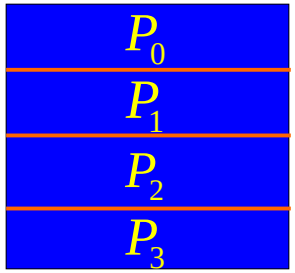
- データの分散方法
- BLAS(Basic Linear Algebra Subprograms)
- 連立一次方程式の解法
  - 共役勾配法(Conjugate Gradient Method, CG法)の並列アルゴリズム

配列の分割方法(1/4)

- MPIで並列化を行う際には,各ノードで配列を分割して持つようにすると,メモリを節約することができる.
- ブロック分割
  - 連続する領域をノード数で分割



列方向に分割したブロック分割

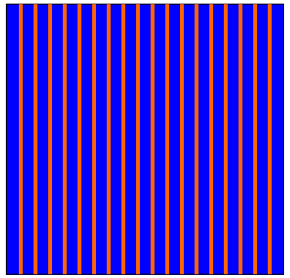


行方向に分割したブロック分割

配列の分割方法(2/4)

- サイクリック分割
  - 1列(または1行)ごとに分割
  - ブロック分割に比べてロードバランスが取りやすい

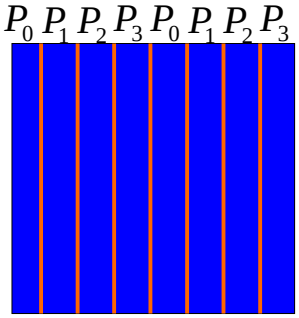
0123012301230123…



列方向に分割したサイクリック分割

配列の分割方法(3/4)

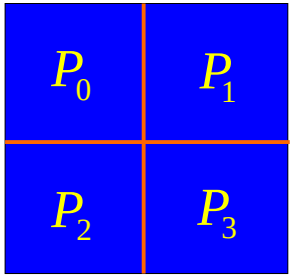
- ブロックサイクリック分割
  - 複数列(または複数行)ごとに分割
  - ブロック分割とサイクリック分割の中間



列方向に分割したブロックサイクリック分割

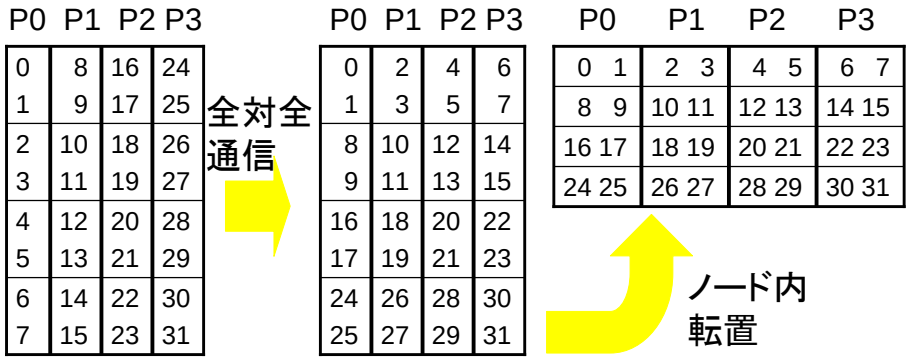
配列の分割方法(4/4)

- 二次元分割
  - 行方向と列方向の両方を分割
  - 一次元分割よりも通信量が減ることがある
  - 二次元のブロック分割，サイクリック分割，ブロックサイクリック分割が考えられる.



二次元ブロック分割

全対全通信(MPI_Alltoall)を用いた行列の転置



BLASとは

- BLAS(Basic Linear Algebra Subprograms)は，ベクトルと行列の基本演算を行うサブルーチン群.
- 呼び出し方法が統一されているので，異機種間での移植性に優れている.
- 各プロセッサに高度に最適化されており，高い性能を発揮する.
  - Intel MKL，AMD ACML，IBM ESSL，GOTO BLAS

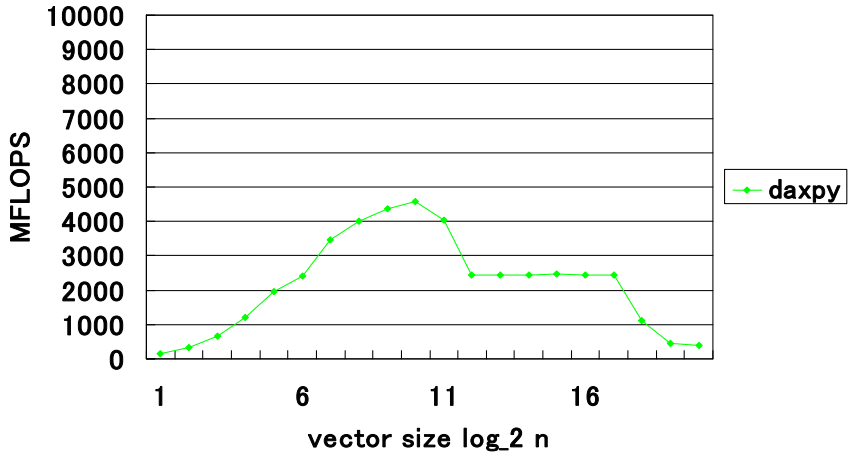
BLASの種類

- BLASには，大きく分けてLevel 1，Level 2，Level 3の3種類がある．
- Level 1（ベクトルーベクトル演算）
  - dot（内積），axpy（ベクトルの定数倍とベクトルの和）など
- Level 2（行列ーベクトル演算）
  - gemv（行列ベクトル積）など
- Level3（行列ー行列演算）
  - gemm（行列積）など
- サブルーチンの名前の前にs（単精度実数），d（倍精度実数），c（単精度複素数），z（倍精度複素数）が付く．

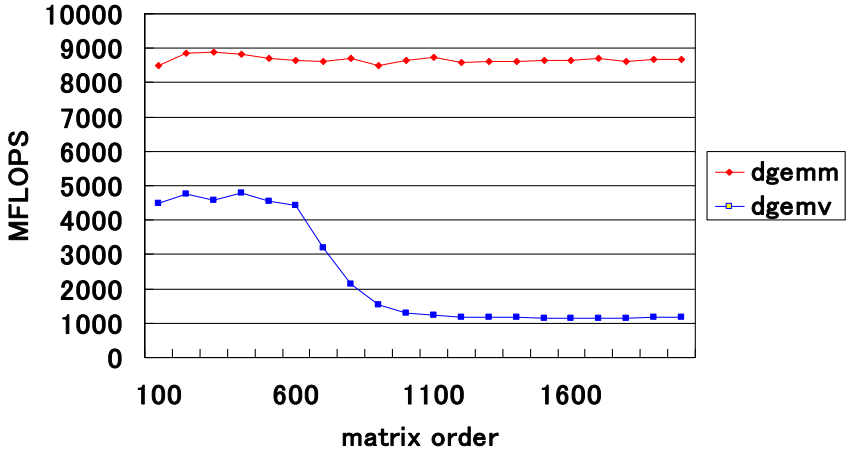
BLASの演算回数

BLAS	ロード回数＋ストア回数	浮動小数点演算回数	比 $n=m=k$
Level 1 DAXPY $y=y+\alpha x$	$3n$	$2n$	$3:2$
Level 2 DGEMV $y=\beta y+\alpha Ax$	$mn+n+2m$	$2mn$	$1:2$
Level 3 DGEMM $C=\beta C+\alpha AB$	$2mn+mk+kn$	$2mnk$	$2:n$

BLASの性能（Woodcrest 2.4GHz 4MB L2 cache，Intel MKL 9.1）



BLASの性能（Woodcrest 2.4GHz 4MB L2 cache，Intel MKL 9.1）



連立一次方程式の解法

- 科学技術計算の多くは連立一次方程式

$$A\mathbf{x} = \mathbf{b}$$

を解くことに帰着される.

- 大規模な連立一次方程式を解く際には，多大な時間を要することから，これまでにさまざまな解法が提案されている.
- 並列処理は，実行時間を短縮する上で必要不可欠.

連立一次方程式の解法

- 連立一次方程式の解法には大きく分けて，
  - 直接法（ガウスの消去法，LU分解など）
  - 反復法（ガウス・ザイデル法，ヤコビ法，共役勾配法など）

  がある.
- 今回は反復法である共役勾配法を取り上げる.

共役勾配法（CG法）

- 連立一次方程式の係数行列が対称正定値な行列 A（任意のベクトル$\mathbf{x}$ に対して $\mathbf{x}^T A\mathbf{x} > 0$ ）に対しては

$$A\mathbf{x} = \mathbf{b}$$

の解が

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{x}, A\mathbf{x}) - (\mathbf{x}, \mathbf{b})$$

という関数を最小にするという性質を利用したのが共役勾配法（Conjugate Gradient Method，CG法）である.

共役勾配法（CG法）のアルゴリズム

初期ベクトル$\mathbf{x_0}$を用意する

$\mathbf{r_0} = \mathbf{b} - A\mathbf{x_0}$，　$\mathbf{p_0} = \mathbf{r_0}$

for k = 0, 1, …

$\alpha_k = \dfrac{(\mathbf{p_k}, \mathbf{r_k})}{(\mathbf{p_k}, A\mathbf{p_k})}$ ・内積（ddot），行列ベクトル積（dgemv）

$\mathbf{x_{k+1}} = \mathbf{x_k} + \alpha_k \mathbf{p_k}$ ・ベクトルの定数倍とベクトルの和（daxpy）

$\mathbf{r_{k+1}} = \mathbf{r_k} - \alpha_k A\mathbf{p_k}$ ・行列ベクトル積（dgemv），ベクトルの定数倍とベクトルの和（daxpy）

収束判定

$\beta_k = \dfrac{(\mathbf{r_{k+1}}, A\mathbf{p_k})}{(\mathbf{p_k}, A\mathbf{p_k})}$ ・行列ベクトル積（dgemv）と内積（ddot）

$\mathbf{p_{k+1}} = \mathbf{r_{k+1}} + \beta_k \mathbf{p_k}$ ・ベクトルの定数倍（dscal）とベクトルの和

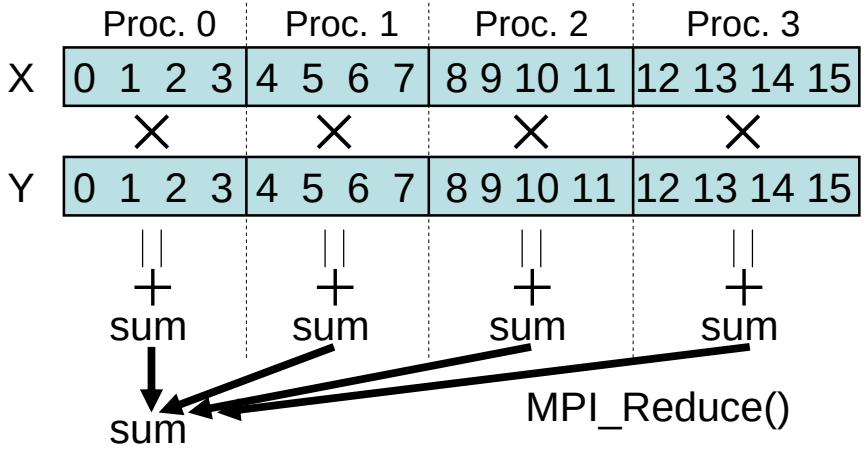
end

共役勾配法（CG法）に必要な計算

- ベクトルの内積（ddot）
- 行列ベクトル積（dgemv）
- ベクトルの定数倍とベクトルの和（daxpy）
- ベクトルの定数倍（dscal）
- 上記のルーチンは自分で書いてもよいが，よく使われるので，ライブラリになっている．

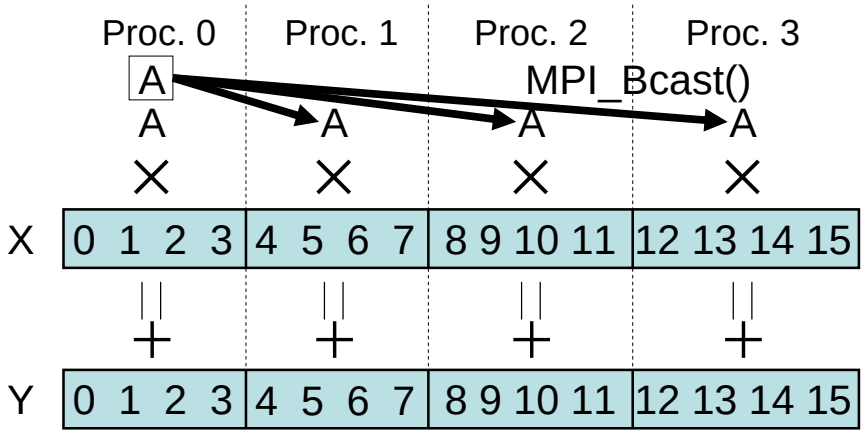
内積の並列化

```
for (i = 0; i < N; i++) sum += X[i] * Y[i];
```



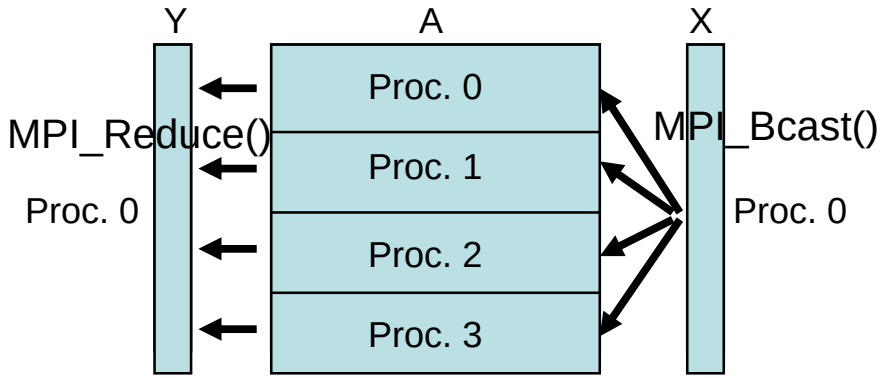
daxpyの並列化

```
for (i = 0; i < N; i++) Y[i] += A * X[i];
```



行列ベクトル積の並列化

```
for (j = 0; j < N; j++)
  for (i = 0; i < N; i++) Y[i] = A[j][i] * X[j];
```

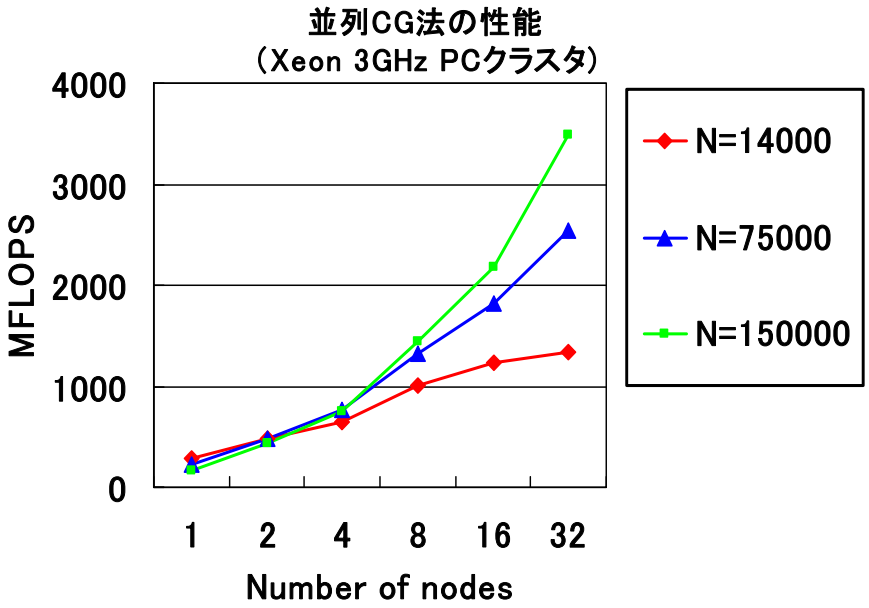


並列CG法のプログラム(一部)

```
do j=1,lastrow-firstrow+1
   sum = 0.0d0
   do k=rowstr(j),rowstr(j+1)-1
      sum = sum + a(k)*p(colidx(k))
   enddo
   w(j) = sum
end do

do i = l2npcols, 1, -1
   call mpi_irecv(…)
   call mpi_send(…)
   call mpi_wait(…)
   do j=send_start,send_start + reduce_recv_lengths(i) - 1
      w(j) = w(j) + q(j)
   end do
 end do
```

疎行列の非ゼロ要素だけを一次元配列に格納



まとめ

- 並列数値計算アルゴリズムとして,
  - CG法による連立一次方程式の解法

  を取り上げた.
- 問題の領域をどのように分割するかが鍵となる.
  - ブロック分割, サイクリック分割, ブロックサイクリック分割
- 科学技術計算には並列性を含んだものが多いので, 並列化はそれほど難しくない(ものもある).
  - 逐次ルーチンとMPIの集団通信ルーチンを使えば, かなりの部分は記述できる.