

## P2-2 分散型SILCの設計: MPIベースの行列計算ライブラリを使いやすくするインタフェース

梶山民人(JST) 額田彰(JST) 須田礼仁(東京大学)  
長谷川秀彦(筑波大学) 西田晃(東京大学)

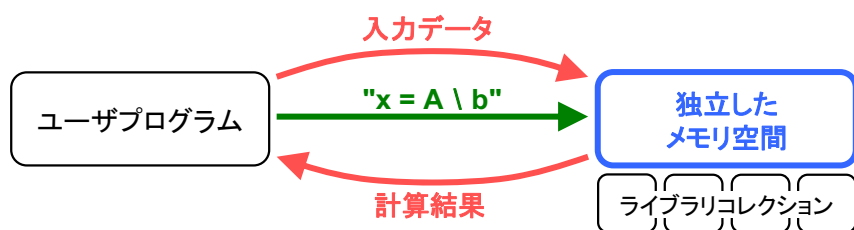
- ◆ SILC: Simple Interface for Library Collections
  - 行列計算ライブラリに対する計算環境やプログラミング言語に依存しないインタフェースの提案
  - SILC version 1.1 公開中(Unix, Linux, Mac OS X 対応)

<http://ssi.is.s.u-tokyo.ac.jp/silc/>

- ◆ 従来法の問題点
  - ライブラリ固有のデータ構造と関数呼出しに基づくライブラリ利用法ではユーザプログラムが特定のライブラリに依存
  - 別のライブラリ(解法, 行列の格納形式など)を利用するにはユーザプログラムの大幅な修正が必要

- ◆ 提案手法の特徴
  - SILC のユーザプログラムは使用する行列計算ライブラリに特化したコードを含まない
  - 容易に別の解法や行列の格納形式を利用できる
  - ライブラリの実行環境を逐次環境から並列環境に移すことで並列計算のメリットを自動的に享受できる

- ◆ SILC の基本アイデア
  1. ライブラリ呼出しを**データ転送**と**計算の指示**に分離
  2. 計算を**文字列(数式)**で指示
  3. ユーザプログラムから**独立したメモリ空間**で計算を実行



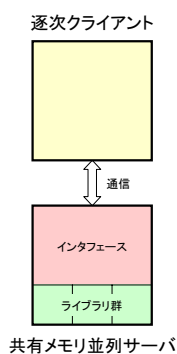
- ◆ 従来法と提案手法の比較
  - LAPACK による連立一次方程式  $Ax = b$  の求解例

```
double *A, *b;
int kl, ku, lda, ldb, nrhs, info, *ipiv;
/* 行列 A およびベクトル b の作成 */
dgbtrf (N, N, kl, ku, A, lda, ipiv, &info); /* LU 分解 */
if (info == 0)
    dgbtrs ('N', N, kl, ku, nrhs, A, lda, ipiv, b, ldb, &info); /* 求解 */
```

- 同じ計算を実現する SILC のユーザプログラムの例

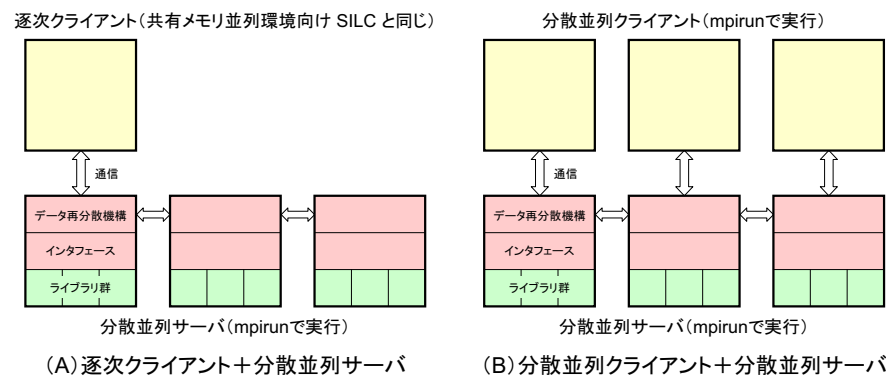
```
silc_envelope_t A, b, x;
/* 行列 A およびベクトル b の作成 */
SILC_PUT ("A", &A);
SILC_PUT ("b", &b);
SILC_EXEC ("x = A \ b"); /* 適切な求解ルーチンが呼び出される */
SILC_GET (&x, "x");
```

- ◆ 共有メモリ並列環境向け SILC
  - クライアント・サーバ方式で実現(右図)
  - 逐次クライアント(C, Fortran, Python)
  - OpenMP に基づく共有メモリ並列サーバ
  - モジュールによるライブラリ群の統合
    - 演算モジュール
      - ライブラリ関数を呼ぶためのラッパー集
      - prefer 文による選択(下図)
    - 格納形式モジュール
      - 行列の格納形式を定義するモジュール
    - 動的リンクによりプラグイン可能

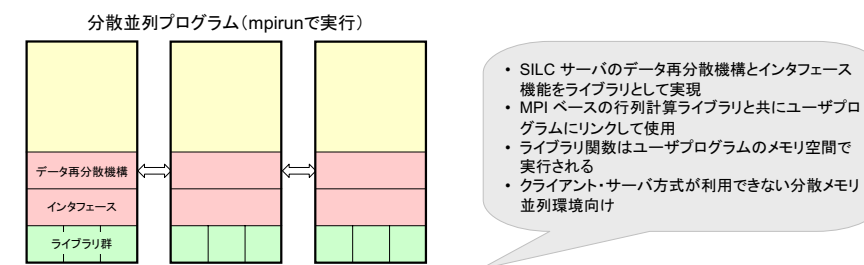


```
SILC_EXEC ("prefer leq_lu");
SILC_EXEC ("x1 = A \ b"); /* LU 分解法で求解 */
SILC_EXEC ("prefer leq_cg");
SILC_EXEC ("x2 = A \ b"); /* CG 法で求解 */
SILC_EXEC ("d = b - A * x1; norm1 = sqrt(d * d);"); /* ||b - Ax1|| */
SILC_EXEC ("d = b - A * x2; norm2 = sqrt(d * d);"); /* ||b - Ax2|| */
```

- ◆ 分散メモリ並列環境向け SILC の設計
  - MPI ベースの行列計算ライブラリを計算環境やプログラミング言語に依らない方法で手軽に利用可能に
  - 検討しているシステム構成
    - (A) 逐次クライアント+分散並列サーバ
    - (B) 分散並列クライアント+分散並列サーバ
    - (C) サーバの機能を内蔵した分散並列ライブラリ



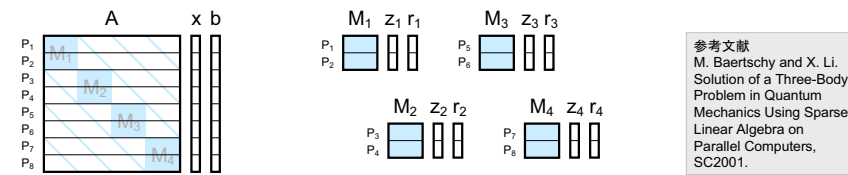
- クライアントはサーバプロセスの1つに接続してデータ転送と計算指示を行なう
- サーバは受信したデータをデータ再分散機構によりライブラリの要求するデータ分散方式に変換し、複数のプロセスに分散して保持する
- 計算結果はクライアントに送り返される前にクライアントの要求するデータ分散方式に戻される



(C) サーバの機能を内蔵した分散並列ライブラリ

- ◆ 分散版 SILC に要求される機能は?
  - 行列の分散方式を変えるためのデータ再分散機構
    - ユーザプログラムとライブラリが異なるデータ分散方式を採用している場合に相互変換を行なう

- プラグイン可能なモジュールとして実現
- 主なデータ分散方式
  - 2次元ブロックサイクリック分割(例: ScaLAPACK)
  - 1次元ブロック分割(例: Lis, SuperLU\_DIST)
  - データ分散なし(逐次クライアントの場合)
- サーバプロセスを部分集合に分けて並列動作させる仕組み
  - 例: 反復解法の記述とライブラリ呼出しの組合せ
    - (1) 連立一次方程式  $Ax = b$  に対するブロック Jacobi 前処理付き反復解法を SILC の数式で記述
    - (2) “\” 演算子を用いたライブラリ呼出しにより対角ブロックを前処理行列とする  $M_i z_i = r_i$  を求解



(1) 反復解法により  $Ax = b$  を (2) 2プロセスずつ4グループに分かれて8プロセス全体で求解  $M_i z_i = r_i$  を並列かつ独立に求解

- プロセス数の異なるクライアント・サーバ間接続
  - M: クライアントプロセス数, N: サーバプロセス数
- 主として想定する利用形態は  $M < N$ 
  - ユーザプログラムは PC や比較的小さなクラスタ環境で
  - SILC サーバは大規模クラスタ環境で実行
- 各クライアントプロセスの接続先の数は常に1つとする
  - データ再分散はすべてサーバ側で行なう
  - クライアント側の通信処理をシンプルにする
- $M > N$  のサポートは検討中
  - ライブラリの提供するデータ再分散ルーチンが利用できないことがあるため