

SILC: Simple Interface for Library Collections

概要

行列計算ライブラリを使いやすくするインタフェース SILC (Simple Interface for Library Collections)を提案し、逐次・共有メモリ並列環境向けおよび分散並列環境向けシステムをクライアント・サーバ方式で実現した。本システムにより、ユーザプログラムを行列計算ライブラリや計算環境に依らない方法で作成することが可能になる。

従来法の問題点

ライブラリ固有のインタフェース(API)に基づいてユーザプログラムを作成すると、ユーザプログラムが特定のライブラリに依存する。そのため、別のライブラリを利用するにはユーザプログラムの大幅な修正が必要である。

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

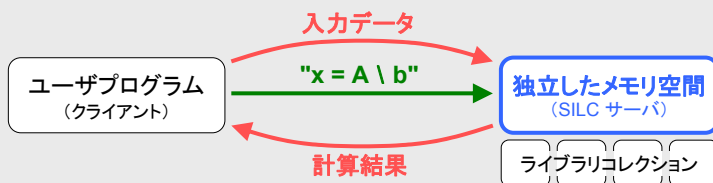


図1 SILC の基本アイデア

◆ 従来のユーザプログラム(図2)

ライブラリ固有のデータ構造で行列やベクトルなどの入力データを用意する。次に、所定の関数名と引数の順序でライブラリ関数を呼び出す。この方法は直感的で手軽な反面、ユーザプログラムが特定のライブラリに依存する。

```
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 */
```

図2 従来のユーザプログラム

(LAPACK による連立一次方程式 $Ax = b$ の求解例)

◆ SILC のユーザプログラム(図3)

SILC のサポートするデータ構造で入力データを用意する。各データに名前を付けて **SILC_PUT** ルーチンでサーバに預ける。必要なデータを預けたあと、**SILC_EXEC** により計算を文字列(数式)で指示する。計算終了後、**SILC_GET** で結果を受け取る。

```
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");
```

図3 SILC のユーザプログラム

(図2のユーザプログラムと同じ計算を実現する例)

SILC の特徴

- ◆ ユーザプログラムを特定の行列計算ライブラリや計算環境に依存しない方法で作成できる
 - ユーザプログラムを修正せずに別の計算環境を利用できる
 - さまざまな行列計算ライブラリを同じ枠組みで利用できる
 - 解法や行列の格納形式を容易に変更できる
- ◆ 最小限の入力データと記憶領域を用意するだけでよい
 - 計算途中で必要な記憶領域は自動的に確保・解放される
- ◆ さまざまなプログラミング言語で利用できる
 - C言語, Fortran, Java, Python(オブジェクト指向スクリプト言語)

SILC のシステム構成

- ◆ クライアント・サーバ方式に基づく2種の実装
 - OpenMP 版: 逐次・共有メモリ並列環境向け
 - MPI 版: 分散並列環境向け
- ◆ ユーザプログラムは逐次プログラムまたは分散並列(MPI)の並列プログラム
- ◆ SILC サーバは共有メモリ並列(OpenMP)または分散並列(MPI)の並列プログラム

表1 サポートする計算環境の組み合わせ

構成	ユーザプログラム (クライアント)	SILC サーバ	対応バージョン
(1)	逐次	逐次	OpenMP 版
(2)	逐次	共有メモリ並列	OpenMP 版
(3)	逐次	分散並列	MPI 版
(4)	分散並列	分散並列	MPI 版

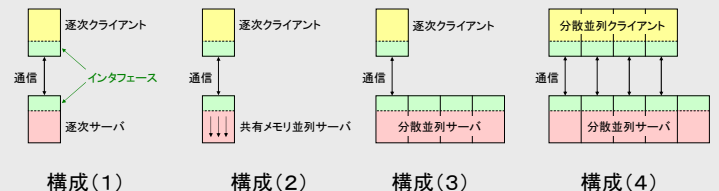


図4 SILC のシステム構成

行列計算ライブラリインタフェースの機能

- ◆ データ構造
 - データ型: スカラー, ベクトル, 行列, 3次元配列
 - 精度: 整数, 実数, 複素数(単精度および倍精度)
 - 行列の格納形式: 密, 帯, CRS(行圧縮)形式, JDS(Jagged Diagonal Storage)形式
- ◆ 命令記述言語
 - 文(statement)
 - 右辺が式である代入文(単純代入文および累算代入文)
 - 引数が式である手続き呼出し文
 - 式(expression)の構成要素
 - 変数名, 定数(i, e, pi など)
 - リテラル(すべてのデータ型のリテラルを記述可能)
 - 四則演算(+, -, *, /, %), 連立一次方程式の求解(A \ b)
 - 要素ごと乗算(*@)および要素ごと除算(/@)
 - 共役転置(A'), 複素共役(A~), 転置(A'~)
 - 関数(sqrt, diagvec, zeros, ones, sparse など)
 - 添字(subscript)による部分参照および制約代入
 - 例: "A[1:5,1:5]" は行列Aの部分行列(5行5列)を表す

SILC の利用例

◆ 偏微分方程式の初期値問題の求解

従来法および SILC の逐次ユーザプログラム(図5, 図6)と MPI 版の並列サーバを表2の計算環境の組み合わせで実行して2次元拡散方程式の初期値問題を差分法(クランク・ニコルソン法)で解いた。時間ステップ数 $K = 40$ までのユーザプログラムの実行時間を図7に示す。時間ステップ毎の連立一次方程式の求解には前処理なし CG 法(Lis 1.0.2)を用い、行列 A, C は CRS 形式で与えた。次数を N , 反復回数を α とすると通信量は $O(KN)$, 計算量は $O(\alpha KN)$ であるため、反復回数のかかる問題であれば通信コストを加味しても遠隔の高速な並列計算機を利用した方が速い。

1. 行列 A, C , および開始時刻 t_0 の解 x_0 を作成
2. 時刻 t_k ($k = 1, 2, 3, \dots$) について:
 - (a) 時刻 t_k の右辺ベクトル $b_k = Cx_{k-1}$ を計算(疎行列ベクトル積)
 - (b) 連立一次方程式 $Ax_k = b_k$ を求解

図5 従来法のユーザプログラム

1. 従来法のプログラム(図5)と同様に A, C , および x_0 を作成
2. SILC_PUT で A, C, x_0 をサーバに送信
3. 時刻 t_k ($k = 1, 2, 3, \dots$) について:
 - (a) SILC_EXEC で $b_k = Cx_{k-1}$ の計算(疎行列ベクトル積)を指示
 - (b) SILC_EXEC で連立一次方程式 $Ax_k = b_k$ の求解を指示
 - (c) SILC_GET で時刻 t_k の解 x_k をサーバから受信

図6 SILC のユーザプログラム

表2 ユーザプログラムと SILC サーバの実行環境

	ユーザプログラム	SILC サーバ
従来法	Xeon4 (1 PE)	—
SILC (local)	Xeon4 (1 PE)	Xeon4 (4 PEs)
SILC (remote #1)	Xeon4 (1 PE)	Xeon8 (8 PEs)
SILC (remote #2)	Xeon4 (1 PE)	Altix (16 PEs)

表3 実行環境の詳細(Gigabit Ethernet で接続)

ホスト名	仕様
Xeon4	IBM eServer xSeries 335 (dual Intel Xeon 2.8 GHz, 1.0 GB RAM) × 4, Red Hat Linux 8.0, LAM/MPI 7.0
Xeon8	Xeon4 と同じ PC クラスの別の8ノード
Altix	Intel Itanium2 1.3 GHz × 32基, メモリ 32 GB, Red Hat Linux Advanced Server 2.1, SGI MPI 4.4 (MPT 1.9.1)

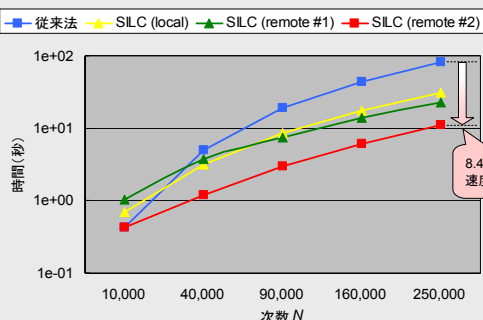


図7 実験結果

(時間ステップ数40までのユーザプログラムの実行時間)

◆ 複数の解法の利用行列

計算ライブラリはモジュールとしてサーバに組み込まれる。prefer 文を用いてモジュールの優先順位を変えることにより、図8に示すようにひとつの演算子に対応する複数のライブラリ関数(解法)を容易に切り替えて利用できる。

```
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|| */
```

図8 連立一次方程式の求解精度を比較するプログラム

まとめ

計算環境やプログラミング言語に依存しない行列計算ライブラリインタフェース SILC を提案し、逐次・共有メモリ並列環境向けおよび分散並列環境向けシステムをクライアント・サーバ方式で実現して提案手法の実現可能性を示した。また、種々の計算環境、解法、行列の格納形式を柔軟かつ容易に利用できること、高性能並列計算機の遠隔利用により本手法の主要な利用コストである通信時間を十分に補える速度向上が得られることを示した。

今後の課題

- ◆ 現実味のある問題への提案手法の適用
- ◆ 主な行列計算ライブラリに対応するモジュール(ラッパー)
- ◆ クライアント・サーバ方式に依らない SILC システムの実装
- ◆ 命令記述言語の単独利用可能なスクリプト言語への拡張
- ◆ 数式に基づく実行時最適化

ソフトウェアの公開

- ◆ OpenMP 版 SILC のソースコード、Win32 用バイナリパッケージ、ドキュメント、サンプルプログラムをフリーソフトウェアとして一般公開している
- ◆ ホームページ: <http://ssi.is.s.u-tokyo.ac.jp/silc/>
- ◆ 動作環境: Unix, GNU/Linux, Mac OS X, Windows
- ◆ 公開履歴
 - SILC バージョン1.0(2005年9月20日)
 - SILC バージョン1.1(2005年11月25日)
 - SILC バージョン1.2(2006年11月12日)

論文リスト

- [1] A. Nishida, H. Kotakemori, T. Kajiyama, A. Nukada. Scalable Software Infrastructure Project, In SC06, poster, 2006.
- [2] T. Kajiyama, A. Nukada, R. Suda, H. Hasegawa, A. Nishida. A Performance Evaluation Model for the SILC Matrix Computation Framework. In Proc. IFIP International Conference on Network and Parallel Computing (NPC 2006), pp.93-103, 2006.
- [3] 梶山, 額田, 須田, 長谷川, 西田. 行列計算ライブラリインタフェース SILC の分散並列環境への実装, 2006年並列/分散/協調処理に関する『高知』サマリー・ワークショップ (SWoPP 高知 2006), 情報処理学会研究報告 2006-HPC-107, pp.251-256, 2006.
- [4] T. Kajiyama, A. Nukada, R. Suda, H. Hasegawa, A. Nishida. Distributed SILC: An Easy-to-Use Interface for MPI-based Parallel Matrix Computation Libraries. In Proc. 2006 Workshop on State-of-the-Art in Scientific and Parallel Computing (PARA'06), LNCS, Springer, to appear.
- [5] 梶山, 額田, 須田, 長谷川, 西田. 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, ハイパフォーマンスコンピューティングと計算科学シンポジウム (HPCS2006), p.31, ポスター, 2006.
- [6] T. Kajiyama, A. Nukada, H. Hasegawa, R. Suda, A. Nishida. LAPACK in SILC: Use of a Flexible Application Framework for Matrix Computation Libraries. In Proc. 8th International Conference on High Performance Computing in Asia Pacific Region (HPC Asia 2005), 2005.
- [7] T. Kajiyama, A. Nukada, H. Hasegawa, R. Suda, A. Nishida. SILC: A Flexible and Environment Independent Interface to Matrix Computation Libraries. In Proc. 6th International Conference on Parallel Processing and Applied Mathematics (PPAM 2005), LNCS, to appear.
- [8] 梶山, 額田, 須田, 長谷川, 西田. 共有メモリ並列環境における SILC の実現と利用, 第34回数値解析シンポジウム講演予稿集, pp.49-52, 2005.
- [9] 梶山, 額田, 須田, 長谷川, 西田. SILC: 行列計算ライブラリの利用を簡単化するフレームワーク, 第10回日本計算工学会講演会論文集, pp.239-242, 2005.
- [10] 梶山, 額田, 須田, 長谷川, 西田. 行列計算ライブラリに対する計算環境に依存しないインタフェースの開発, ハイパフォーマンスコンピューティングと計算科学シンポジウム (HPCS2005), ポスター, 2005.
- [11] 長谷川, 須田, 額田, 梶山, 中島, 高橋, 小武守, 藤井, 西田. 計算環境に依存しない行列計算ライブラリインタフェース SILC, 情報処理学会研究報告 2004-HPC-100, pp.37-42, 2004.