

Web インタフェースを備えた異機種並列分散プログラミング環境

松浦 健一郎† 荒木 拓也†
蒲池 恒彦† 妹尾 義樹†

近年、大規模化/複雑化するアプリケーションを効率良く処理するために、異なる並列計算機を接続した異機種並列分散システムへの期待が高まっている。しかし計算機間のアーキテクチャの違いや、データ転送の問題により、異機種並列分散システム向けの並列化/最適化は容易ではない。そこで我々は、異機種並列分散システムを容易に活用するためのプログラミング環境を開発した。本環境は Web ベースのグラフィカルユーザインタフェースを採用しており、遠隔地においても直観的な操作によって異機種並列分散システム向けの並列化/最適化作業を行なえることが特徴である。

A Heterogeneous Parallel and Distributed Programming Environment with Web Interface

KEN-ICHIRO MATSUURA,† TAKUYA ARAKI,† TSUNEHICO KAMACHI†
and YOSHIKI SEO†

In recent years, heterogeneous parallel and distributed systems which connect various kinds of computers are considered promising for managing the applications which are getting larger and more complicated. However, it is not easy to parallelize or optimize programs for heterogeneous systems because of architectural difference among computers and data transfer problems. In our study, we developed a programming environment to allow programmers to utilize such heterogeneous systems with minimal effort. This environment is equipped with a Web-based graphical user interface, which enables intuitive operation on parallelization and optimization for heterogeneous systems, even at remote sites.

1. はじめに

数値計算アプリケーションの大規模化/複雑化に対応するため、ベクトル計算機/スカラ並列機/PC クラスタといった、異なる性質を持つ計算機をネットワークで接続した、異機種並列分散システムに対するニーズが高まっている。異機種並列分散システムでは、システムを構成する各ノード計算機の性質を活用することにより、複合的の巨大計算を効率良く処理することができると期待される。

しかし異機種並列分散システム向けのプログラムを開発することや、既存の逐次プログラムを異機種並列分散システム向けに並列分散化/最適化することは容易ではない。なぜなら異機種並列分散システムには、以下のようなノード計算機間の「継ぎ目(シーム)」とも呼ぶべきものが存在し、これらのシームを意識しつつシステムの性能を引き出すようにプログラミングすることは困難だからである。

- 各ノード計算機向けのアーキテクチャの違いや性能差
- ノード内とノード間に渡る複数レベルの並列性やデータ転送
- システムの状態の動的な変化

異機種並列分散プログラミングにおけるシームを取り除くことは、プログラマにとって非常に負担が大きい。従って、プログラマがシームを意識せずにプログラム開発ができる、「シームレス」な異機種並列分散プログラミング環境を提供することが重要である。

そこで我々は、ベクトルスーパーコンピュータ SX-4、分散メモリ型スカラ並列計算機 Cenju-4、および PC クラスタの結合システムをターゲットに、シームレスな異機種並列分散システム向けプログラミング環境の構築を進めている。

また異機種並列プログラミング環境は、プログラムの並列分散化/最適化に関する情報や、性能分析情報/実行時モニタリング情報などの多様な情報を利用者に対して適切に提示するとともに、利用者がシステムを容易に運用するための手段を備えなければならない。グラフィカルユーザインタフェースは、このような目的を達成す

† RWCP 並列分散システム NEC 研究室
RWCP Parallel and Distributed Systems NEC Laboratory

るための有効な方法の一つである。

さらに、一般に異機種並列分散システムは複数の並列計算機を組み合わせた大規模なシステムとなるため、設置場所が限られる。このため遠隔地の利用者也考慮に入れる必要がある。遠隔地でも利用が可能で、かつ導入が簡単なグラフィカルユーザインタフェースとしては、Webインタフェースが有力な候補の一つとして挙げられる。

我々のプログラミング環境はWebベースのグラフィカルユーザインタフェースを備えており、遠隔地から異機種並列分散システム向けの並列分散化/最適化作業が行えるほか、システム上でのプログラムの実行や、実行時モニタリング情報の表示なども可能である。

本稿では2章でプログラミング環境全体の概要について述べ、3章でWebインタフェースについて述べる。続く4章では並列分散化支援ツールの主要機能とユーザインタフェースに関して説明する。最後の5章ではまとめと今後の予定について述べる。

2. 異機種並列分散プログラミング環境

我々のプログラミング環境は、並列分散化支援ツール、並列分散化コンパイラ、異機種間通信ライブラリから構成される(図1)。これらのコンポーネントが連携して、異機種並列分散システムにおけるシームレスなプログラミングを可能にする。以下、各コンポーネントの概要を説明する。

2.1 並列分散化支援ツール

グラフィカルなWebインタフェースを備えたプログラミング支援ツールであり、Fortranで記述された逐次プログラムの並列分散化を支援する。本ツールは並列分散化/最適化に必要な種々の情報をグラフィカルに表示するとともに、簡単な操作によるプログラムの並列分散化/最適化作業を可能にする。主な機能は次の通りである：

- プログラムの論理構造の表示
- 配列の分散状態の表示と、配列分散指示の支援
- 配列分散の自動決定
- プログラム実行時間の静的予測
- 実機におけるプログラムの実行
- 実行時モニタリング情報の表示
- 実行時再コンパイルの制御と、進捗状況の表示

本ツールの機能の詳細およびWebインタフェースの構成に関しては4章で述べる。

2.2 並列分散化コンパイラ

HPF言語をベースに、異機種並列分散処理向けの拡張機能を加えたコンパイラである。本コンパイラはHPFプログラムを入力し、異機種並列分散システム上で動作する並列分散プログラムを出力する。通信プリミティブとしては後述の異機種間通信ライブラリを使用する。本コンパイラの主な機能は次の通りである：

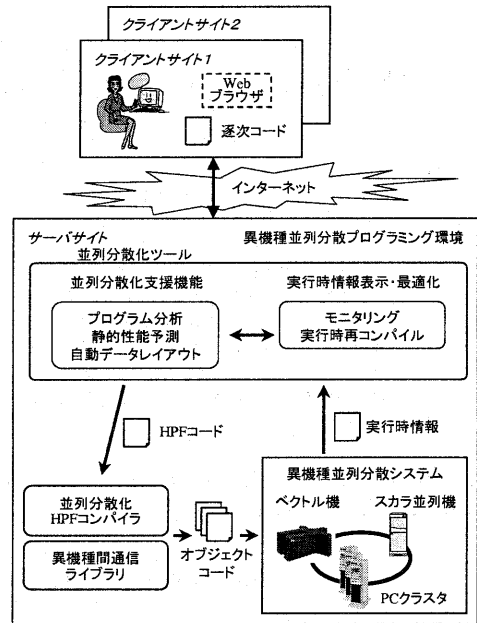


図1 異機種並列分散プログラミング環境の全体構成

- ユーザの配列分散指示に基づく、データや処理のプロセッサに対する割り付け
 - ループや並列実行可能性の判定とスケジューリング
 - 並列実行に必要な通信コードの挿入
- また拡張機能としては、プログラムの実行時モニタリング情報採取用のコードを挿入する機能や、実行時コンパイラ向けの拡張指示文を解釈してコードを変形する機能などがある。

2.3 異機種間通信ライブラリ

複数の異機種ノード計算機間と、各ノード計算機内における並列プロセッサ間において、シームレスな通信を実現するMPI(Message Passing Interface)ライブラリである。本ライブラリはノード計算機間のアーキテクチャの違いに起因する数値データのバイトオーダーや浮動小数点形式の相違なども吸収する。

3. Webインタフェース

本章では、我々が開発した並列分散化支援ツールにWebインタフェースを採用したことの背景、Webインタフェースの欠点を補填するためのJavaによるユーザインタフェースに関して述べる。

3.1 Webインタフェースを採用した背景

並列分散化支援ツールのユーザインタフェースを開発する上で、我々が目的としたのは次の事柄である：

導入が簡単であること ツールの利用者に対して専用の

ソフトウェアを配布したり、インストールさせたりする手間は省くことが必要である。

開発が容易であること ユーザインタフェースの開発、拡張が容易かつ迅速に行なえることが必要である。
遠隔地から利用できること 異機種並列分散システムは企業内などに設置される場合が多いので、セキュリティの問題を解決して、遠隔地からの利用を可能にする必要がある。

Web インタフェースは、以上のような条件を満たすグラフィカルユーザインタフェースの実現に大変適している。以下目的別に、Web インタフェースの利点に関して述べる：

導入の簡単さの実現 一般的な Web ブラウザによる表示/操作が可能であるため、利用者は専用のクライアントソフトウェアをインストールする必要がない。また Web インタフェースは広く普及しているため、利用者は特殊な操作を新たに学習する必要がない。

開発の容易さの実現 専用のクライアントソフトウェアを利用者のプラットフォーム毎に用意する必要がないため、移植や動作検証の労力を大幅に削減できる。また Web ブラウザの機能向上に伴って、比較的容易にツールの表現力や操作性を向上させることが可能である。

セキュリティ問題の解決 大規模な異機種並列分散システムは企業などの組織の内部に設置する場合が多いと考えられるが、一般にこのような組織ではセキュリティ上の理由からネットワークを通過可能なプロトコルに関して制限が加わる。その点、Web インタフェースに用いる HTTP はネットワークの通過を許可されていることが多い。

以上のような理由から、本ツールは Web インタフェースを採用した。Web インタフェースの採用により、異機種並列分散システムの設置場所における利用者だけでなく、遠隔地の利用者に対しても、グラフィカルなプログラミング支援機能が提供できる。

一方、Web インタフェースは遠隔地の利用者に対してグラフィカルな環境を提供する上で優れているが、以下のような欠点も併せ持つ：

対話的な操作に関する制限 Web インタフェースにおける利用者の操作手段としては、ハイパーリンクや、フォームを用いた入力/選択機能がある。しかしプログラムの並列分散化/最適化作業においては、これらの機能だけでは十分な操作性を確保することができない。

リアルタイム性の欠如 本ツールが扱う実行時モニタリング情報は、プログラムの実行に伴って時々刻々と変化するという点において、リアルタイム性を持つ情報である。しかし Web インタフェースは、利用者の操作に対して応答することが基本であるため、リアルタイムに更新される情報をサーバ側が能動的

に提示する用途には向かない。

ファイルアップロードの煩雑さ 利用者が手持ちの逐次プログラムを並列分散化する場合、並列分散化支援ツールが稼動するサイトにプログラムを転送する必要がある。Web インタフェースでこの転送を実現するためにはファイルアップロード機能を用いるが、セキュリティ上の配慮から、やや面倒な確認プロセスを伴う。

3.2 Web UI と Java UI の併用

Web インタフェースの弱点を補うために、本ツールでは Web インタフェースを用いたユーザインタフェース (以下、Web UI) に加えて、Java によるユーザインタフェース (以下 Java UI) も開発した。ネットワークの帯域/遅延や通過可能なプロトコルの条件を満たした利用環境ならば、利用者は Web UI に加えて Java UI を併用することができる。Java UI の使用が不可能な利用者に対しては Web UI のみの提供となる。Java UI の併用により、Web UI に不足する対話的な操作やリアルタイム性が大幅に強化される。

Web UI を通じて提供する主要な機能は以下の通りである：

- ソースプログラムのグラフィカルな表示
- 種々の実行環境におけるプログラム実行時間の静的な予測
- 実行時モニタリング情報のサマリー表示

Java UI を通じて提供する主要な機能は次の通りである：

- プログラムの論理構造およびループの並列化状態の表示
- 配列の分散状態およびループにおける配列アクセスパターンの表示
- ドラッグ&ドロップによる配列分散の指示
- 静的性能予測結果のグラフ表示
- 実行時モニタリング情報のリアルタイム表示

Web UI / Java UI とツールの各コンポーネントとの構成は図 2 の通りである。現在の Java UI は、サーバサイトにて Java アプリケーションを動作させ、クライアントサイトからは X プロトコルを通じて利用する構成である。

4. 並列分散化支援ツールの主な機能

以下の節では並列分散化支援ツールの主要な機能についてより詳しく説明する。

4.1 ソースコード表示

図 3 は Web UI によるプログラムのソースコード表示である。ソースコードに含まれる Fortran および HPF の構文要素 (予約語/コメント/文字列など) を色分けして表示する機能を持つ。色分け表示のルールはユーザーごとにカスタマイズすることができる。また、ループのネスト構造を表示する機能もある。図 3 では、「[*]」と

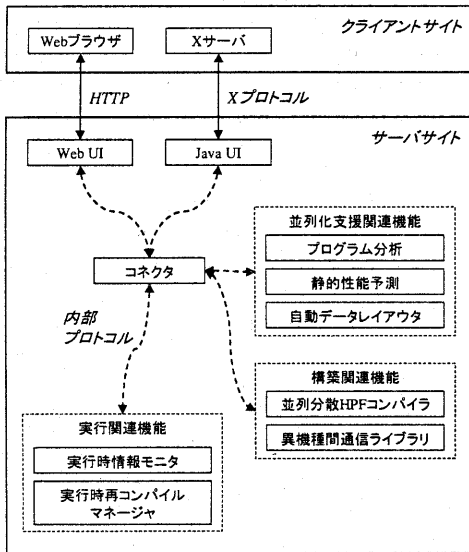


図2 Web UI / Java UIの構成

いう記号でループのネスト構造を图示するとともに、その表示色によってループの並列化状態を示している。そのほか、後述する実行時情報モニタリングに関連して、ループ毎に実行時進捗情報を表示するか否かを指定する機能も備えている。

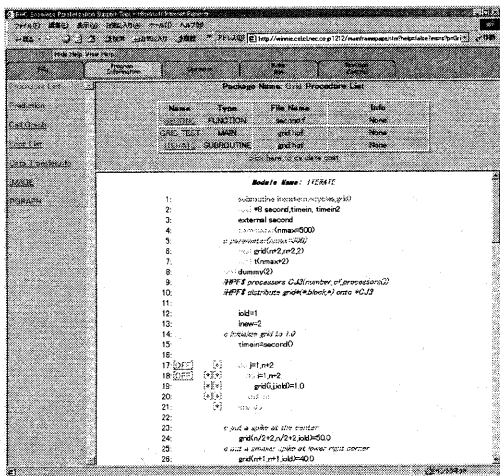


図3 プログラムソース表示

4.2 プログラム論理構造表示

図4はJava UIによるプログラム論理構造の表示で

ある。プログラムに含まれるループ/制御構文/手続き呼び出しなどの構造を、複数手続き間に渡ってツリー状に表示することができる。ツリーの各階層はマウス操作により開閉するので、必要な階層の深さまでを表示することが可能である。

また、各ループにおいてアクセスされる配列を、立方体(2次元以下の場合には矩形)のアイコンによって図示する。そして、アクセスの方向や範囲をグラデーションによって表現する。グラデーションの色が薄くなる方向に、該当ループにおいて配列がアクセスされる。

HPFをベースにした並列化では、配列のアクセスパターンを手続き間に渡って把握することが肝要である。本機能は、手続き間に渡って配列のアクセスパターンを容易にブラウズする手段を提供することにより、ユーザによるプログラム並列化/最適化のための分析作業を支援する。

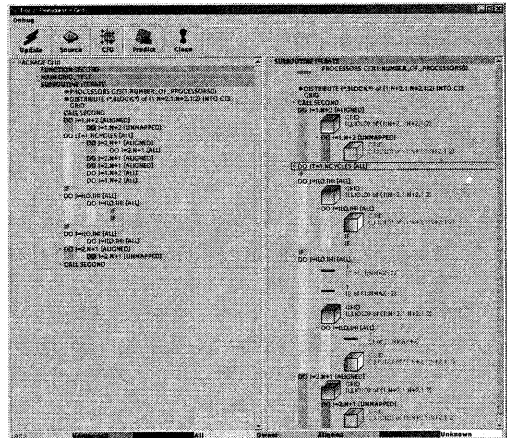


図4 プログラム論理構造の表示

4.3 ドラッグ&ドロップによる配列データのレイアウト

前節の図4に示したJava UIによるプログラム論理構造の表示画面においては、配列を表す立方体のアイコンをドラッグ&ドロップすることにより、配列データのレイアウトを指定することができる。また、本ツールは自動データレイアウト¹⁾も備えており、手動によるデータレイアウトとの併用が可能である。例えば、ユーザが必要に応じて、プログラムを並列化する上で重要な配列のデータレイアウトを手動で行ない、残りの配列に関しては自動データレイアウトを適用する、といった使い方がある。

4.4 静的性能予測

本ツールの異機種並列分散システムを構成する各計算機ノードや、計算機間を結合するネットワークの性能パラメータを基に、ワークステーション上でシミュレーションを行ない、プログラムの実行時間を静的に予測

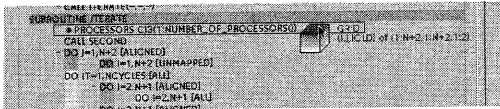


図5 ドラッグ&ドロップによる配列分散指示

する機能である。予測結果は Web UI によりテキストベースのテーブル形式で表示できるほか、Java UI では図6のような3次元グラフ表示が可能である。グラフ表示は、実行時間表示のほか、各計算機ノードの単一プロセッサエレメントによる逐次実行時間との比（速度向上率）を表示する機能も持つ。

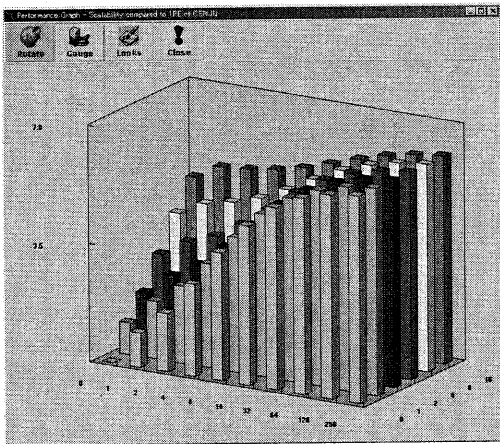


図6 静的性能予測結果グラフ

4.5 実行時モニタリング情報の表示

本支援ツールは、プログラムの実行時に各種の情報をモニタリングし、これをユーザに対して提示したり、あるいは最適化に活用したりするための仕組みを持つ。また、プログラムの一部分に関して、プログラムの実行中に実行時情報を用いた最適化を行ない、元のオブジェクトコードを最適化されたものに入れ換える実行時再コンパイル機構²⁾も備えている。

図7は、モニタリングした実行時情報を Java UI でリアルタイムに表示する機能である。棒グラフの各バーは計算に参加しているプロセッサエレメントの進捗度を表しており、数値表示は全体としての進捗度を表す。

図8は、実行時再コンパイルの進捗状況をグラフ表示する機能である。本機能は Web UI / Java UI のいずれでも利用可能であり、Web UI では実行終了後の一括表示、Java UI ではリアルタイム表示となる。

図8の上半分のグラフは、縦軸に実行時再コンパイルの対象となる部分プログラムの進捗度、横軸に経過時間をとったダイアグラムである。グラフは、再コンパイル前/再コンパイル中/再コンパイル後の3部分に色分け表示される。

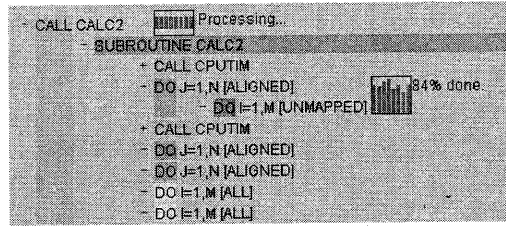


図7 実行進捗状況表示

そして図8の下半分は、再コンパイル前を基準とした各部分における相対実行速度を表示する速度計である。相対速度はダイアグラムの傾斜によっても知ることができるが、より直観的に速度向上率を把握できるように速度計を用意した。

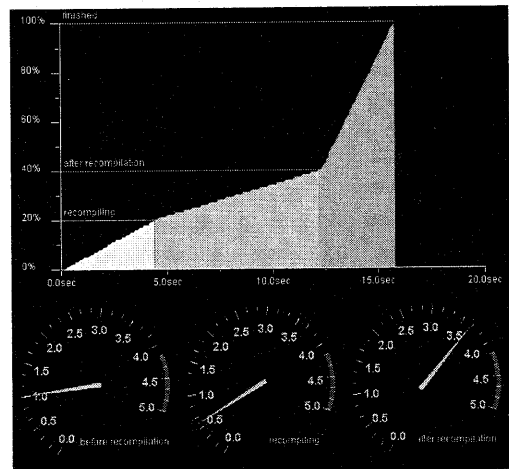


図8 実行時再コンパイル進捗状況表示

5. 関連研究

異機種並列分散システム向けのプログラミング支援環境としては、武宮らによる STA³⁾がある。STAは大まかに3つの要素から構成される。その要素とは、遠隔関数呼び出しとメッセージパッシングをサポートする通信基盤 SCE、各種並列計算機に対して共通のユーザインタフェースを提供するプログラム開発環境 PPDE、およびタスクマッピング/メタスケジューリング/資源管理を行なう実行環境 PPEXe である。

異機種並列分散システムを構成する各計算機には、各々のメーカーから提供されたツールが存在する。プログラム開発環境 PPDE は、共通のユーザインタフェースとなる専用のクライアントソフトウェアを用意し、各種ツールとの間にアダプタを介在させることにより、統一されたユーザインタフェースにおける各種ツールの運

用を可能にする。具体的に PPDE は、コンパイラ/デバッガ/性能解析ツールなど各種ツールからの情報をエディタ上のプログラムに対応づけて表示することにより、プログラム開発/最適化作業を支援する。

異機種並列分散プログラミング環境という点において、3) は我々の研究に近い。我々の研究の特徴は、Web インタフェースを採用しているために、専用のクライアントソフトウェアを導入することなく Web ブラウザが動く任意の環境で使用できることや、HTTP が利用可能であれば遠隔地からの使用も容易であることである。また、自動データ分割/静的性能予測/実行時コンパイルなどの機能を持つこと、グラフィカルな並列化支援ユーザインタフェースを備えることも特徴である。

一方、Web ベースのユーザインタフェースを備えたプログラミング支援環境としては、Insung らによる OpenMP 用の並列プログラミング支援ツール⁴⁾がある。バックエンドの並列化コンパイラ Polaris は、逐次 Fortran コードを並列化して Fortran+OpenMP コードに変換する。ユーザは Polaris のコントロールを Web ページ (Form ベース) で行なえるほか、実行性能測定結果をスプレッドシートで閲覧できる。

Web インタフェースは静的に情報を表示するのには向くが、リアルタイム情報を提供したり、グラフィカルかつインタラクティブな操作性を提供するのは難しい。そこで 4) では、AT&T のオープンソフトウェアである VNC を併用する。VNC は、HTTP により遠隔ワークステーションのデスクトップのスクリーンショットを受信するとともに、手元の端末のマウス入力とキーボード入力を送信する仮想端末ソフトである。ただしネットワークによる遅延のため、必ずしも実用的な速度で動作するわけではない。

Web インタフェースを備えた並列分散プログラミング支援ツールという点において、4) は我々の研究に近い。我々の研究の特徴は、異機種並列分散システム向けであること、自動データ分割/静的性能予測/実行時コンパイル/グラフィカルな並列化支援ユーザインタフェースなどの機能を備えることなどである。

6. まとめと今後の課題

我々が開発した異機種並列分散プログラミング環境のユーザインタフェース構成と主要機能に関して述べた。本プログラミング環境は Web インタフェースを備えており、遠隔地においても異機種並列分散システム向けの並列化/最適化作業を行なえることが特徴である。また Web インタフェースが不得意とする対話的な操作やリアルタイム性を補うために、Java を用いたユーザインタフェースも併用する。

本プログラミング環境の今後の主な課題は、異機種並列分散システムをより効率良く運用するための実行時負荷分散機構の実現や、実行時通信モニタリング情報を活

用した最適化などである。一方、ユーザインタフェースに関しては、次のような機能拡張を予定している。

ファイルアップロード機能 現時点では、ユーザが所有する逐次コードをクライアントサイトからサーバサイトに対してアップロードする際に、ftp などを用いる必要がある。今後の拡張では Web UI にファイルアップロード機能を実装して、本プログラミング環境の枠内でファイルのアップロードを可能にする。この際、セキュリティ上、Web ブラウザがファイルをアップロードするたびに確認作業が必要となるため、複数のファイルをアップロードする手順はやや複雑となってしまう。そこで Java アプレットなどを用いたアップロード支援機能も併せて実現する予定である。

ファイルダウンロード機能 アップロードとは逆に、プログラミング環境を用いて並列化/最適化したコードを、ユーザの手元にダウンロードする機能も必要である。現時点でも、並列化/最適化済みのコードをサーバサイトに出力する機能は実装済みであるため、ダウンロード機能の実現は比較的容易である。

外部ツールとの連携 ソースコードを編集するためのテキストエディタなど、他のツールとの連携機能も必要である。HTTP のみが利用可能な場合には、ソースファイルを一旦クライアントサイトにダウンロードして、ツールを用いた変更の後、再びサーバサイトにアップロードする必要がある。この手順を補助する仕組みを実現する予定である。

ユーザアカウント管理機能 複数ユーザが本プログラミング環境を同時に利用するためには、各ユーザのアカウントを管理する仕組みが必要である。具体的にはサーバサイトの接続口にアカウントサーバを用意しておき、ここでユーザのアカウントや所有ファイル、利用時間などを管理する予定である。

参考文献

- 1) 松浦 健一郎, 他: データ並列プログラムに対する高速な自動データ分割手法, 情報処理学会論文誌, Vol. 41, No. 5, pp. 1420-1429 (2000).
- 2) 荒木 拓也, 他: 実行時再コンパイルを用いた HPF プログラムの最適化, 並列処理シンポジウム JSPP2001, pp. 343-350 (2001).
- 3) 武宮 博, 他: 並列分散科学技術計算を支援するソフトウェア/システム (STA) の構築, 情報処理, Vol. 40, No. 11, pp. 1104-1109 (1999).
- 4) Insung, P., et al.: Towards an Integrated, Web-executable Parallel Programming Tool Environment, *Proc. ACM/IEEE SC2000 Conf.*, Dallas (2000).