

KLIC へのデータ並列処理機能の導入について

4N-2

藤瀬 哲朗
三菱総合研究所近山 隆
東京大学上田 和紀
早稲田大学稲村 雄†
NEC 情報システムズ関田 大吾
三菱総合研究所

1. はじめに

KLIC [1] は、並行並列論理型言語 KL1 [2] を汎用計算機上に実装した処理系である。並列プログラミングでは、一般に処理(プロセス)およびデータをどこに配置し実行するのかが記述することになる。KLIC の実装方式では処理系自身が必要に応じてデータを移し余計なデータの移動を防ぐ。そのためデータ移動記述が必要がなく、複雑で粒度が細かいタスク並列プログラムの容易な記述を実現している。

並列数値処理の場合では、各ワーカがアクセスし易い場所に配列データの必要な部分を配置することが重要であり、KLIC のようなタスク並列言語とは異なる戦略を必要とする。そこで KLIC に分散配置配列と呼ばれる並列数値計算向け配列データを導入する。分散配置配列はプログラマの指示に従って分割し各ワーカへの物理的に配置することを可能とし、さらに数値計算については一般的である Fortran 90 や C 等の逐次並列言語でその処理を記述することができるものとする。

本稿では並列動作する KL1 プログラムから逐次数値処理に適した仕様を持つ言語で記述したルーチン呼び出して、全体として並列数値計算を行うのに適した分散配置配列について、その基本方針とその仕様の概要について述べる。本配列を利用することで、例えばベクタ並列計算機上にタスク並列処理とベクトル処理を兼ね備えた計算を容易に実現する環境を構築できる。

2. 基本方針

分散配置配列の仕様は KL1 の並行論理型言語としての性質を損なわないこと(副作用を持たないこと、したがって実行順序が最終結果に影響を及ぼさないこと、など)、および、KLIC の構成の方針を損なわないこと(実装に標準的な言語機能や OS 機能のみを用い、高い移植性を保つこと、など)を前提とした設計とした。具体的には以下のような性質を満たす仕様となっている。

- KL1 プログラムから呼び出した逐次処理向きの他言語プログラムのルーチンが、特別な処理をする必要はない。すなわち、通常の逐次処理のために記述した逐次言語のサブルーチンを特段の変更を加えることな

くそのまま用いて、その上位レベルに KL1 によるプログラムを記述することによって、全体としては並列処理を実現することができる。

- 大量のデータに対して処理を行なう場合、十分な性能を得られる。KLIC は並列論理型言語システムとしてはかなり高性能であるが、数値処理に用いた場合の性能は高くない。しかし、数値処理の主要な部分は逐次言語で記述したルーチンが行なうため、高い性能が期待できる。また、大量のデータを扱う場合は多くの計算を一回の逐次言語ルーチン呼び出しで行なえるため、通信頻度を低くでき、通信のオーバヘッドも抑制できる。

(1) 他言語インタフェースの設計方針

逐次処理向きの他言語で記述したルーチンの中では、KL1 プログラムから呼ばれたということ、全体として並列処理を行なうプログラムの一部として動作するという点について、いっさい特別な配慮をせずに済むインタフェースをもつ。具体的には以下の条件を満足する。

- KL1 プログラムから呼び出されたことによって、同じ言語のプログラムから呼び出された場合と異なる処理を行なう必要はない。具体的には以下の通り。
 - KL1 プログラムから渡されるデータの表現は手続き型言語プログラムで通常用いられる表現である。
 - KL1 プログラムに計算結果を返す方法は手続き型言語で通常用いられる方法と同じである。
- 他言語ルーチンでは、全体としては並列処理を行なっているということ意識してコーディングする必要はない。

(2) 性能確保のための設計方針

性能上問題となる大量の計算は、主として他言語ルーチンで行なうものと想定し、その前提で高い性能が出るような設計とした。具体的には以下の方針を採った。

- データの配置や転送の時期は、ユーザプログラムから指示できるようになっている。これは KL1 の負荷分散の方針と合致する。指示はかなり高いレベルで行ない、具体的な転送の詳細などは処理系に任せられている。具体的には以下の通り。
 - データ分散方針はユーザプログラムから指示する。
 - データ転送の時期を、ユーザプログラムから指示できる。

A Distributed Array in KLIC
FUJISE, T., Mitsubishi Research Institute, Inc.
CHIKAYAMA, T., University of Tokyo
UEDA, K., Waseda University
INAMURA, Y., NEC Information Systems
SEKITA, D., Mitsubishi Research Institute, Inc.

† 現在、日本ペリサイン

- 配列データの表現は他言語ルーチンでの処理に都合の良い表現をとっている。これにともない KL1 からこれらを扱う際に若干のオーバーヘッドが生じるが、これは許容する。これは主たる処理を他言語ルーチンで行なうものと想定しているためである。
- データを分散したままで一括して扱うこともできる。すなわち、KL1 レベルでは単一のデータと見える配列が、物理的には分散して配置された要素を持てる。
- データを分散したままの配列に対し、まったく同じ分散を指示する場合には、データの (いったん集中してまた分散する、といった) 転送が起きない。

3. 分散配置配列の機能概略

(1) 分散配置配列の基本機能

- 分散配置配列データは KL1 からプロセスのように見える。すなわち、配列データのインタフェースはすべてストリームを介したメッセージのやりとりによって行なう。配列要素の更新は KL1 からはプロセスの内部状態変化となる。
- KL1 レベルで扱う配列は 1 次元のものだけであり、他言語プログラムで多次元配列を扱う場合も、KL1 ではその表現のデータ並びを 1 次元配列として扱う。他言語の 2 次元以上の配列の 1 次元配列への対応づけは、ユーザの責任である。ただし、配列の分解に関するプリミティブは、通常の多次元配列の密表現へのマッピングを用いる場合に簡単になるものを用意する。
- 配列要素の型として浮動小数点数のみのものと、整数のみのものがある。浮動小数点数は C, Fortran の倍精度に対応するものである。整数は C の long, Fortran の INTEGER*4 または *8 に対応するもの (機種に依存する) であり、KLIC の整数で表現できる範囲に制約されない。¹
- 配列データに対する操作は以下のものがある。
 - 基本的な要素の操作 (取り出し, 更新)
 - 単一の配列を複数の部分配列に分解する操作
 - 複数の配列から、それらの要素を集めて単一の配列を組み立てる操作

分解・組み立てには、多次元の行列を通常の方法で 1 次元配列によって表現した場合、多次元行列に対し数値計算に頻繁に利用される分解パターンを容易に実現する分解・組立てプリミティブがある。

これらの他に、プログラムのセマンティクス上は意味を持たない以下の操作がある。

- 配列要素をすべて 1 個所に集める操作

(2) 他言語インタフェース

- 他言語ルーチンの呼び出しは逐次的なサブルーチン呼び出しである。すなわち他言語ルーチン呼び出し

¹KLIC の整数データは C の long が 32 ビットのシステムでは 28 ビット、64 ビットのシステムでは 60 ビットの精度しかないが、分散配置配列の要素はそれぞれ 32, 64 ビットの精度をもつ。

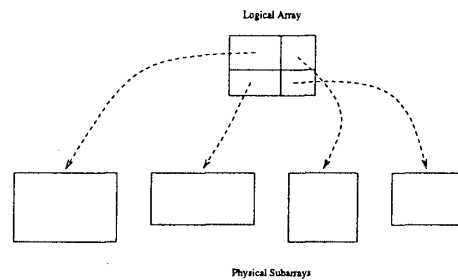


図 1: 配列の表現

た後、そのルーチンが終了して復帰するまで、当該のワーカ²は他の処理は行なわない。

- 他言語ルーチンの引数に配列データを渡す場合、他言語ルーチンから見たその表現形式は、その言語の通常の配列の表現と同じに見える。

(3) 配列の表現

- 分散配置配列はなんらかの単位ごとに分かれた部分配列を統括するものになっている (図 1)。個々の部分配列は単一のワーカ上にある場合もあれば、複数のワーカに分かれて分散配置されている場合もある。ひとつの分散配置配列全体が部分配列ひとつだけからなる場合もある。

(4) 想定する使用法

分散配置配列を操作するプリミティブは、通常部分配列の生成・初期化、全体配列の組立て、部分配列への分解、部分配列要素の集結、計算結果の組立て、の手順で用いられることを想定する。³

4. おわりに

本データ配列の実装はすでに終了し、近々リリースする予定であり、また評価を実施する予定である。

なお本研究の一部は、情報処理振興事業協会の創造的ソフトウェア育成事業の一端として実施された。

参考文献

- [1] Chikayama, T.: KLIC: A Portable Parallel Implementation of a Concurrent Logic Programming Language, Parallel Symbolic Languages and Systems, (Ito, Halstead and Queinnec Ed.), Springer, pp.286-294, 1995.
- [2] Ueda, K., Chikayama, T.: Design of the Kernel Language for the Parallel Inference Machine, *The Computer Journal*, Vol.33, No.6, pp.494-500 (1990).

²KLIC の並列計算の一部を行なうプロセス。多くの場合物理的なプロセッサに対応する。

³部分配列へ分解しての計算の後にいったん全体配列を組み立てるのは、部分配列に分けたままではアルゴリズムの見通しが悪くなるのを避けるためである。要素データの転送が本来不要なら、この組み立てを行なっても要素データの転送は生じず、配列の構成に関する情報が送られるだけなので、大きなオーバーヘッドにはならない。