

## 仮想並列計算機 ADENA4 システムの形式的実装

杉山 和徳

2D-7

京都大学工学部応用システム科学教室

## 1 はじめに

現在、仮想並列計算機 ADENA4 システムを開発中で、その核となる部分を実装したところである。このシステムの開発においては、微妙に異なる数々のアーキテクチャの ADENA4 システムの性能評価をこのシステムにて容易にできることを狙って C++ などのプログラミング言語より抽象度がより高い仕様記述言語 VDM を用いて簡潔かつ正確に仕様記述し、IFAD VDM-SL Toolbox と C++ Code Generator というツールを用いて VDM 言語で仕様記述し、そしてその VDM 仕様記述を C++ プログラムに自動変換してインプリメントする開発手法を取った [2] [3]。これにより、仕様変更が容易かつ迅速に対応できる ADENA4 シミュレータを効率的に実現することが可能となった。

## 2 並列計算機 ADENA4

## 2.1 並列計算機 ADENA4 アーキテクチャ

並列計算機 ADENA4 [1] は ADEPS と呼ぶ並列計算スキームに基づいて計算を進める並列計算機である。並列計算スキーム ADEPS とは多次元配列データを様々な次元方向から 1 次元配列データに切り分け、その 1 次元配列データごとに並列計算を繰り返し実行する並列計算スキームである [1]。

並列計算機 ADENA4 は、 $L$  をシステム・サイズを決定するある整数として、1 台のホスト・プロセッサ HP とそのホスト・メモリ  $hm$ 、2 次元配列構成のローカル・プロセッサ群  $\{LP[i, j] : i, j = 1, \dots, L\}$  と各々に付随するローカル・メモリ  $\{lm[i, j] : i, j = 1, \dots, L\}$  及び各プロセッサ間でデータ交換するためのグローバル・メモリとして機能する 3 次元配列構成のメモリ・バンク配列  $\{mba[i, j, k] : i, j, k = 1, \dots, L\}$  から構成される。ホスト・プロセッサ HP はホスト・メモリ  $hm$  にアクセスでき、各ローカル・プロセッサ  $LP[i, j]$  は自分のローカル・メモリ  $lm[i, j]$  にアクセスでき、更に、3 つの計算状態 ( $LP[/, /i, /j] : x$  方向計算、 $LP[i, /, /j] : y$  方向計算、 $LP[i, j, /] : z$  方向計算) に応じて、 $LP[/, /i, /j]$  はメモリ・バンク配列要素  $\{mba[l, i, j] : l = 1, \dots, L\}$

に、 $LP[i, /, /j]$  はメモリ・バンク配列要素  $\{mba[i, l, j] : l = 1, \dots, L\}$  に、 $LP[i, j, /]$  はメモリ・バンク配列要素  $\{mba[i, j, l] : l = 1, \dots, L\}$  に各々アクセスできる。また、ホスト・メモリ  $hm$ 、ローカル・メモリ  $\{lm[i, j] : i, j = 1, \dots, L\}$  及びメモリ・バンク配列  $\{mba[i, j, k] : i, j, k = 1, \dots, L\}$  間でのデータ交換も可能となっている。

ADENA4 上での 3 次元偏微分方程式の数値計算においては、その 3 次元計算領域内のある格子点  $(i, j, k)$  上のデータはメモリ・バンク配列要素  $mba[i, j, k]$  にデータを置いて計算を進めていくことになるが、その 3 次元計算領域が ADENA4 のシステム・サイズ  $L \times L \times L$  を越える場合は各メモリ・バンク配列要素  $mba[i, j, k]$  は複数の格子点上のデータを多重して持つことにより計算を進める。計算領域内  $\{(1, \dots, R_x) \times (1, \dots, R_y) \times (1, \dots, R_z)\}$  のある格子点  $(x, y, z)$  上のデータはメモリ・バンク配列要素  $mba[(x-1) \bmod L + 1, (y-1) \bmod L + 1, (z-1) \bmod L + 1]$  に置かれ、各ローカル・プロセッサ  $LP[i, j]$  は 1 次元配列をなすメモリ・バンク配列要素をアクセスしながら多重度分だけ繰り返して計算を進める。この処理を多重処理と呼ぶ。

## 2.2 並列言語 ADETRAN4

並列言語 ADETRAN4 は ADENA4 上の並列計算を記述するためのプログラミング言語であり、FORTRAN77 プログラミング言語に ADENA4 固有のプログラム単位、データ構造宣言文、並列動作実行文、並列データ転送文などが追加されている。

## 2.3 並列中間言語 ADEINTER

仮想並列計算機 ADENA4 システムは ADEINTER と呼ぶ並列中間言語で記述されたプログラムを 1 命令ずつ逐次解釈しながら並列計算機 ADENA4 でのプログラム実行をシミュレートする。並列中間言語 ADEINTER の各命令は 1 つの命令コードと 0 個以上のオペランドの組から成る。

### 3 仮想並列計算機 ADENA4 システム

#### 3.1 仮想並列計算機 ADENA4 システムの構成

仮想並列計算機 ADENA4 システムとは、ADETRAN4 コンパイラにより ADETRAN4 プログラムから ADEINTER プログラムに変換されたものをインタープリッタ形式で実行することにより、ADENA4 の動作をシミュレートしながら ADEINTER 中間言語の各命令の実行回数などの統計情報を収集する並列計算機シミュレータである。

#### 3.2 仮想並列計算機 ADENA4 システムの VDM 記述

仮想並列計算機 ADENA4 システムは 1 台のホスト・プロセッサ HP とそのホスト・メモリ  $hm$  と  $L \times L$  台のローカル・プロセッサ群  $\{LP[i, j] : i, j = 1, \dots, L\}$  とそのローカル・メモリ  $\{lm[i, j] : i, j = 1, \dots, L\}$  及び  $L \times L \times L$  台のメモリ・バンク配列  $\{mba[i, j, k] : i, j, k = 1, \dots, L\}$  から構成される。各メモリ・バンク配列要素  $\{mba[i, j, k] : i, j, k = 1, \dots, L\}$  は、計算領域が  $\{(1, \dots, R_x) \times (1, \dots, R_y) \times (1, \dots, R_z)\}$  である時、多重処理のため、 $M_x$  (x 方向多重度)、 $M_y$  (y 方向多重度)、 $M_z$  (z 方向多重度) を、それぞれ、 $M_x = (R_x - 1) \text{div} L + 1$ ,  $M_y = (R_y - 1) \text{div} L + 1$ ,  $M_z = (R_z - 1) \text{div} L + 1$  と置くと、 $M_x \times M_y \times M_z$  個のエリアに分割される。ホスト・メモリ  $hm$  の場合は HM 変数番号集合と添字集合の直積からその値への写像として、ローカル・メモリ  $\{lm[i, j] : i, j = 1, \dots, L\}$  はプロセッサ番号集合  $(1, \dots, L) \times (1, \dots, L)$ , LM 変数番号集合及び添字集合の直積からその値への写像として、メモリ・バンク配列  $\{mba[i, j, k] : i, j, k = 1, \dots, L\}$  はメモリ・バンク番号集合  $(1, \dots, L) \times (1, \dots, L) \times (1, \dots, L)$ , 多重処理エリア番号集合  $(1, \dots, M_x) \times (1, \dots, M_y) \times (1, \dots, M_z)$ , MBA 変数番号集合及び添字集合の直積からその値への写像として定義する。ホスト・プロセッサもローカル・プロセッサも  $n$  個のレジスタを持ち、レジスタ番号からその値への写像として定義する。変数、レジスタが取り得る値は、論理値、整数又は実数のいずれかである。仮想並列計算機 ADENA4 システムには記号表があり、ここに各変数 / 配列に関する情報を格納しておくことにより実行時に動的に変数・配列が HM 変数か、LM 変数か又は MBA 変数かを判断できるので、各種変数 / 配列へのアクセス命令 (Load/Save 関連命令) を共通化できた。

VDM 言語による仮想並列計算機 ADENA4 システムの仕様記述では、まず最初にシステム内部の状態を表すレジスタ  $reg$ , ホスト・メモリ  $hm$ , ローカル・メモリ  $lm$ , メモリ・バンク配列  $mba$  などの ADENA4 state の定義から行なった。

ADEINTER 中間言語の各命令に相当する各 VDM operation の実行により ADENA4 state が変化する。仮想並列計算機 ADENA4 システムは ADEINTER プログラムを実行することにより、仮想並列計算機 ADENA4 のレジスタ  $reg$ , ホスト・メモリ  $hm$ , ローカル・メモリ  $lm$ , メモリ・バンク配列  $mba$  などの ADENA4 state の内容がどのように変化するかをプログラムしたものである。

#### 3.3 IFAD VDM-SL Toolbox/C++ Code Generator によるインプリメント

仮想並列計算機 ADENA4 システムの開発では、最初、IFAD VDM-SL Toolbox にて仮想並列計算機 ADENA4 システムを VDM 言語で記述し、その仕様記述の VDM 静的構文検査を行なった。この構文検査でいくつかの簡単な構文上の誤りを発見できた。次に、VDM インタープリッタにて、システムの動的検査を VDM 言語レベルで行なった。システムの誤りはこの段階ではほぼ捕捉された。そして、C++ Code Generator にて VDM 言語で記述されたシステムを C++ プログラムに自動変換した。さらに、VDM 言語がサポートしないシステムのファイル I/O 部分を中心に残りを直接 C++ 言語で記述して、全プログラムをコンパイル / リンクしてシステム全体を完成させた。この時、VDM 言語によるシステムの仕様記述は全部で 4,016 行、これから自動生成された C++ プログラムは全部で 27,047 行、そして、直接 C++ 言語で記述したプログラムは 231 行であった。

仕様決定の作業量を除けば、実質約 2 人月の作業量で仮想並列計算機 ADENA4 システムを開発できた。

### 4 今後の予定及び課題

ローカル・プロセッサがベクトル・プロセッサであるタイプの ADENA4 アーキテクチャも考察中である。現在、このタイプの仮想並列計算機システムを実装中である。今回開発したシステムと多くの共通点を有しているため短期間で実装できると見通している。

#### 参考文献

- [1] T. Nogi: Promising Data Parallel Environment — ADEPS, ADETRAN, ADENA —, *Proc. of the First Aizu Int. Symposium on Parallel Algorithms/Architecture Synthesis*, pp.45-53, IEEE Computer Society, 1995.
- [2] The VDM-SL Tool Group: Users Manual for the IFAD VDM-SL Toolbox, Technical Report, IFAD, 1994.
- [3] The VDM-SL Tool Group: User Manual for the IFAD VDM-SL to C++ Code Generator, Technical Report, IFAD, 1994.