

## 並列 / 分散環境上のプログラミング言語処理系 ParaDisE

6 G - 8

縄田 毅史\* 岡村 耕二\* 平原 正樹\*\* 荒木 啓二郎\*

\*九州大学 工学部 \*\*九州大学 中央計数施設

## 1. はじめに

我々は、密結合・疎結合の分散環境上での並列アプリケーションに関する研究をするに当たり、それを記述し、実行するための言語処理系が必要であった。しかし、密結合の分散環境が手元に無いので、複数のワークステーションを Ethernet で結合した疎結合の分散環境上で、密結合の分散環境と同様のプログラミングができるようにシミュレートする並列プログラミング言語処理系 ParaDisE(並列 / 分散環境 Parallel/Distributed Environment)を開発した。

## 2. ParaDisE の開発方針

現在、ParaDisE の実行環境は疎結合の分散環境であるが、我々の最終的な目標は密結合と疎結合の混在した分散環境上での並列アプリケーションの記述である。そこで、疎結合分散環境上での効率は特に問題とせず、密結合の分散環境へ移行し易くすることを考慮して、ParaDisE の開発を行った。

ParaDisE の言語は Ada を基にした言語で、共有メモリを持った密結合の分散環境で良い効率が得られることを意図している。そこで、疎結合の分散環境から、密結合の分散環境への移行を行い易くするために、単一アドレス空間モデル<sup>[1]</sup>(5章)を用いた。

また、Ada に依存した機能で、疎結合・密結合に依らず、良い効率が得易くするために、いくつかの最適化と拡張(6章)を行った。

## 3. ParaDisE の言語仕様

ParaDisE の言語仕様は Ada を基にしている。既存の Ada 処理系をプログラムの開発、テスト、デバッグ環境として利用するため、Ada の縮小モデルとした。Ada の縮小モデルにするに当たっては、特に科学技術計算等の並列アプリケーションの記述を行うのに必要な仕様を採用した。また、Ada はリアルタイム処理を意識した言語であるが、これに関する仕様は削除した。

## 3.1 基本的な言語仕様

Ada から採用した主な言語仕様は

- パッケージ
- 配列
- 副プログラムの多重定義
- アクセス型

などである。

また、削除した仕様は

- 汎用体
- 例外処理
- 密封型
- レコード型
- 分割コンパイル

などである。

## 3.2 タスク処理とランデブ

タスク処理、ランデブ関係で採用した仕様は

- タスク(型)宣言
- タスク配列
- 動的タスク生成
- entry 呼び出し
- accept 文

• select 文

- accept 選択肢
- 条件付き accept 選択肢
- else 部
- terminate 選択肢

などである。

また、次のような仕様の削除あるいは簡略化を行った。

- タスクの従属関係に関するセマンティクスの簡略化
- タスクの終了・消滅に関するセマンティクスの簡略化
- delay 文、abort 文の削除
- 条件付き entry 呼び出し、時限 entry 呼び出し、entry 族の削除

## 4. ParaDisE 処理系の構成

ParaDisE の処理系の構成は図1の様になっている。

## 4.1 ParaDisE/Compiler

ParaDisE のソースファイルから C のコードと分散配置ファイルを生産するコンパイラである。分散単位となるモジュールごとに C のコードを出力し、それぞれのモジュールの配置に関する情報を出す。配置の種類には全てのノードに配置と、ある番号のノードに配置などがある。

## 4.2 疎結合マクロ定義ファイル

ParaDisE/Compiler が生成する C のコードでは遠隔メモリアクセスや遠隔手続き呼び出しの部分をマクロを呼び出すようにしている。疎結合分散環境用のマクロを定義してあるファイルがこれである。このマクロ定義では、単一アドレス空間モデル支援ライブラリの関数を呼ぶようになっている。密結合のマクロ定義では共有変数への直接アクセスとなる。

## 4.3 単一アドレス空間モデル支援ライブラリ UAS-Lib

分散アドレスをノード番号と実アドレスに分離して、次の疎結合 ParaDisE/Kernel を呼び出すのがこのライブラリである。メモリの読み書き、遠隔手続き呼びだし、メモリ確保などの関数がある。

## 4.4 疎結合 ParaDisE/Kernel ライブラリ DaOS-Lib

疎結合の分散環境上で、遠隔メモリアクセス、遠隔手続き呼びだし、ランデブ、マルチタスキングなどの支援を行うライブラリである<sup>[2]</sup>。

## 4.5 ParaDisE/Linker

ParaDisE/Compiler の生成する分散配置ファイルに基づいて、オブジェクトの名前の結合を行う。これにより、他のノードの関数アドレスは分散アドレスが結合される。

分散配置ファイルと分割されたモジュールのオブジェクトを一つのファイルにまとめる。

## 4.6 ParaDisE/Loader

ParaDisE/Loader により Link されたファイルを分散環境の各ノードに配置し、実行を開始する。

## 5. 単一アドレス空間モデル

単一アドレス空間モデルにより、ParaDisE/Compiler の生成する C のコードの論理的なアドレス空間のイメージと実際の分散環境の物理的なアドレス空間のイメージを分離しているので、密結合・疎結合の分散環境に依存しないコードを生成できる。

単一アドレス空間モデルは分散環境のアドレス空間を共有メモリのようなところからでも見えるアドレス空間と考えるもので、このアドレスを分散アドレスと呼ぶ。密結合の分散環境上では分散アドレスとして共有メモリのアドレスをそのまま使い、疎結合の分散環境上では分散アドレスへのアクセスをソフ

An Implementation of The Programming Language ParaDisE for Parallel/Distributed Environment

Takeshi NAWATA\*, Koji OKAMURA\*, Masaki HIRABARU\*\* and Keijiro ARAKI\*

\*Faculty of Engineering, Kyushu University

\*\*Computation Center, Kyushu University

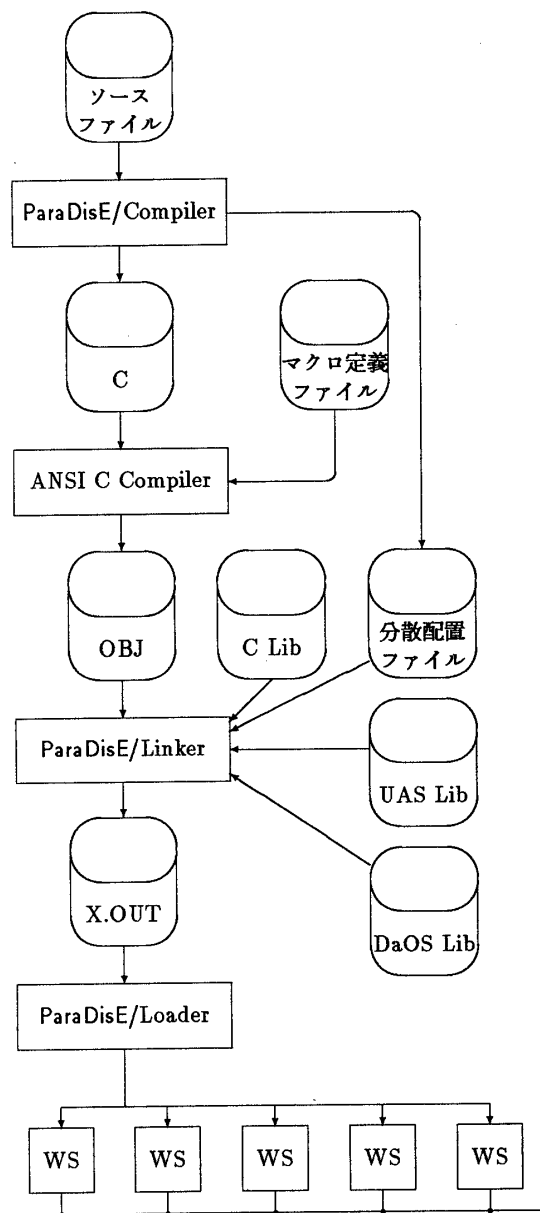


図 1: ParaDisE 処理系の構成

トウェアでシミュレートする。マクロ定義と UAS Lib は分散アドレスのイメージと物理的なノードとその上の実アドレスとのインタフェースをする。

疎結合分散環境では、分散アドレスを次のように扱う。

- i) コンパイラの生成する C のコードで、遠隔のメモリアクセスや遠隔の手続き呼び出しとなる所は、マクロを用いて記述しておく。
- ii) C コンパイラでコンパイルする時に疎結合用のマクロ定義を読み込み、マクロの部分で UAS Lib の関数呼び出しに展開する。
- iii) ParaDisE/Linker で手続き名に分散アドレスを結合する。
- iv) ParaDisE/Linker で単一アドレス空間モデル支援ライブラリを結合する。そして実行中は、遠隔手続き呼び出しや遠隔メモリアクセスをする時に分散アドレスをノード番号とノード上の実アドレスに分離し、実行時カーネル DaOS に渡す。

また密結合分散環境では、疎結合分散環境の場合の C コンパイラでコンパイルを行う部分で密結合用のマクロ定義を読み込み、共有メモリに対する直接アクセスにする。

## 6. 言語仕様に対する最適化と拡張

ParaDisE では、疎結合・密結合での効率を考え、次のような最適化を行う。

### 6.1 タスク配列の配置

各タスク配列は実行時のノード数により各ノードに均一に配置する。

ここで、実行時のノード数は実行開始時の DaOS のシステムコールにより得る。

また、ノード数を得るための拡張機能として、次のような副型を暗黙的に定義する。

subtype NUM\_NODE is INTEGER range 1..ノード数;

これを用いて次のようにタスク配列を宣言できる。

TASKS : array (NUM\_NODE) of T;

また、配列タスクのインデックスを得るための変数を用意する。多次元のタスク配列の場合は一次元に直した時のインデックスを与える。Ada では配列となっているタスクが自分のインデックスを知ることができず、親タスクとのランデブを行わなければならない<sup>[3]</sup>。

TASK\_INDEX : constant INTERGER := インデックス番号;

### 6.2 タスク配列に対する放送型ランデブ

次のように FOR ループでタスク配列全体に同じランデブを行う場合について、放送型のランデブを行う。タスク配列に対するランデブは良く使われるのだが、ランデブは待機するので多くのタスク配列に対してランデブをすると時間がかかってしまう。

```
for I in TASKS'RANGE loop
  TASKS(I).E(IN_ARGUMENTS);
end loop;
```

このような例は同じランデブを各タスクに行うので、放送と考えて良い。そこで、DaOS の放送型ランデブを用いる。

### 6.3 パッケージ

パッケージには独立に動作可能なライブラリ的な関数や手続きを集めたものと、共有資源を持ち、それにアクセスするための入口となる関数や手続きを集めたものがある。

独立に動作可能なライブラリ的なパッケージは各ノードに配置することができる。

また疎結合の場合、共有資源を持つパッケージは共有資源の存在するノードで処理を行う方が効率が上がることが多いので、手続きや関数を共有資源のあるノード上に配置し、他のノードからは遠隔手続き呼び出しにより起動するようにした。

## 7. おわりに

ParaDisE の処理系は密結合の分散環境上において並列アプリケーションを記述するためのものである。九州大学大学院総合理工学専攻 情報システム学専攻の富田研究室で開発中の可変構造型並列計算機<sup>[4]</sup>上での処理系の実現を考えている。

### 謝辞

日頃から御教授下さる九州大学工学部情報工学科の牛島和夫教授、吉田紀彦助手、およびいつもお世話になっている計算機研究室の皆さんに感謝の意を表します。

### 参考文献

- [1] 岡村, 綱田, 平原, 荒木: “単一アドレス空間モデルに基づいた分散環境上での並列プログラミング言語処理系の実現”, 信学技報 Vol.89 No.166, pp.33-38, 1989年8月.
- [2] 岡村, 綱田, 平原, 荒木: “分散環境上での並列プログラミング言語処理系の OS DaOS の概要”, 情報処理学会 第40回(平成2年前期)全国大会, 1990年3月.
- [3] A.Burns, A.M.Lister & A.J.Wellings: “A Review of Ada Tasking.” *Lecture Notes in Computer Science Vol.262*, Springer-Verlag, 1987.
- [4] 村上ほか: “可変構造型並列計算機のシステム・アーキテクチャ”, 情報処理学会「コンピュータアーキテクチャ」シンポジウム論文集, Vol.88, No.3, pp.165-174, 1988年5月.