

7T-6

並列プログラミングの実験支援システム COOP/VM

金井達徳、藤井啓明、柴山 潔、萩原 宏
(京都大学)

1. はじめに

多数の要素プロセッサを結合した並列計算機のアーキテクチャや、その上で実行する並列アルゴリズムの研究開発が盛んに行なわれるようになってきた。これらの研究開発においては、その対象となる並列計算機の性能や、仮想的な並列計算機における並列アルゴリズムの実行性能を評価するために、シミュレーションが、重要な技術になっている。

現在我々が開発中の COOP/VMは、種々のトポロジを持つメッセージ交換型の疎結合マルチプロセッサ上での、各種並列処理プログラムの実行特性の評価実験を支援するシステムである。システムは、メッセージ交換型の汎用プログラミング言語 COOPと、仮想計算機上に実現した実行環境 COOP/VMから構成される。COOP/VMによって、仮想計算機上に論理的に構築した並列計算機において、COOP言語で記述したプログラムの実行をエミュレートすることが可能になる。

2. プログラミング言語 COOP

プログラミング言語 COOP は、C言語をベースに、並列処理のための機構を付加した言語である。メッセージ交換型のマルチプロセッサ上への実装を前提としており、実行時のプロセス・マイグレーションや、ガーベジ・コレクションといった、動的な機能は持たない。

COOPを用いたプログラミングは、図1に示すように、並列計算機を構成する各要素プロセッサ上に存在するモジュールが、互いにメッセージをやりとりしながら、メッ

セージによって起動されたメソッドと呼ぶ処理手順に従って、並列に処理を進めて行く、並列オブジェクト指向モデルに基づいている。メッセージに対応して起動されるメソッドの実行をスレッドと呼び、一つのモジュールに対して複数個のスレッドが同時に存在することができる。他のモジュールによるメソッド呼び出しは、実行結果の待ちを伴う呼び出しと、待ちを伴わない呼び出しの2種類が可能である。

新しいモジュールは、構造的にその親となるモジュールによって同じ要素プロセッサ内に生成される。モジュールやスレッドの要素プロセッサへの割り当ては静的である。従って、各種のネットワーク・トポロジを持つ並列計算機上で実行されるプログラムを記述するためには、プログラム言語のレベルでトポロジを意識したプログラミングを可能にする機能を提供できなければならない。また、各要素プロセッサ上のスレッド実行のスケジューリングは、プログラム全体の挙動に影響を及ぼすため、スケジューラの戦略自身が研究対象となり、それがプログラム可能であることが望まれる。そこでCOOPでは、論理プロセッサと呼ぶ特殊なモジュールを導入し、そのプログラミング機能をプログラマに解放する。

論理プロセッサは、要素プロセッサを言語機能レベルで抽象化したものである。論理プロセッサは、その子にあたるモジュールのメッセージ送受信をトラップし、言語機能レベルのメッセージ送信を対象並列計算機ネットワークを介する通信へ変換する。さらに、自分の子モジュールのスレッド実行順序のスケジューリングも行う。この機

COOP/VM: Programming Environment for Experimental Parallel Processings
Tatsunori KANAI, Hiroaki FUJII, Kiyoshi SHIBAYAMA, Hiroshi HAGIWARA
Kyoto University

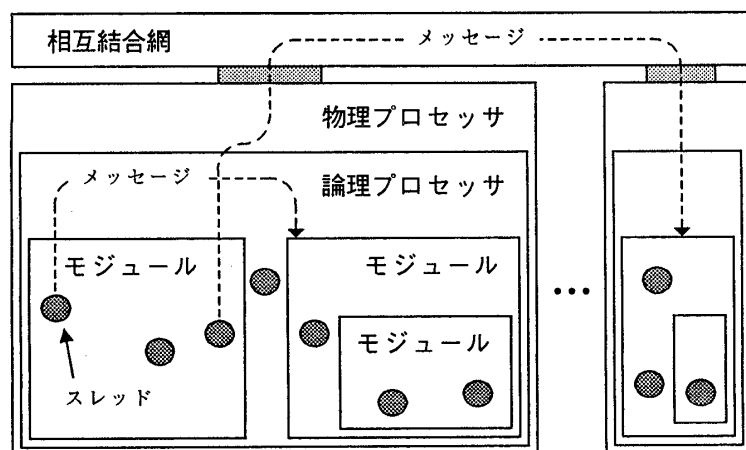


図1 COOPの実行モデル

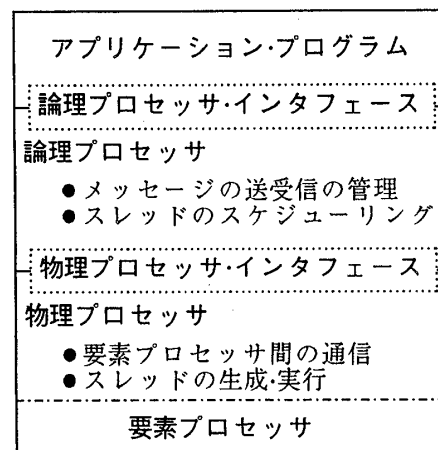


図2 COOPの階層構造

能をプログラマに解放するために論理プロセッサ・モジュールには、次のような論理プロセッサ・インタフェースと呼ぶメソッドの集合が規定されている。

```
sendMessage(callee, method, args, caller)
sendValue(caller, result)
nextThread()
```

論理プロセッサのプログラマは、この仕様を満たすメソッドを定義することによって、論理プロセッサの動作を変更できる。

論理プロセッサのメソッドは、図2に示すように物理プロセッサ・インタフェースを用いて、COOP自身で記述する。物理プロセッサ・インタフェースは、COOPの処理系が提供する最も基本的な関数と変数の集合であり、処理系に依存する。次節で述べるCOOP/VMの場合、実プロセッサ間通信のためのプリミティブやスレッドの操作といった物理プロセッサ・インタフェースを提供している。

論理プロセッサ・モジュールも、他のモジュールの子として生成でき、階層的な論理プロセッサの構成が可能である。これを利用して、逐次的なメッセージ処理を行うような同期機構を持つモジュールを、論理プロセッサとして記述できる。

3. 実験システム COOP/VM

COOP/VMでは、エミュレートする並列計算機の要素プロセッサは、汎用計算機上に定義した一台の仮想計算機によって実現す

る。要素プロセッサ間の通信は、仮想計算機間通信機構によって実現しており、任意の要素プロセッサ間の通信が可能である。実験対象のネットワーク・トポロジに依存する通信時間等のパラメータや、要素プロセッサ内のスケジューリング戦略は、論理プロセッサのプログラムによって実現する。

一般に並列計算機のエミュレーションでは、対象となる計算機上での仮想的な時間の管理と、それに対する同期が必要となる。COOP/VMでは、対象要素プロセッサ上の時間は、各仮想計算機の持つ時計で計る。また、仮想計算機間で送受信されるメッセージには時刻印を付加し、それを用いて仮想時間に対する同期を実現する。各仮想計算機のエミュレートする要素プロセッサの時刻は、仮想計算機間の共有メモリに置くことによって、時間の分散型管理におけるブロードキャストのオーバーヘッドを軽減する。これらの同期の処理は、論理プロセッサのメソッドとしてCOOP自身で記述する。その記述に必要な機能は、物理プロセッサ・インタフェースとして提供される。

4. おわりに

現在、IBM3081KのVM/SP上に、COOP/VMの実装を進めている。この処理系は、COOP言語のコンパイラ、各種のネットワーク・トポロジを実現する論理プロセッサ・ライブラリ、および実行トレースの解析支援系から構成される。