7P-7

可変構造型並列計算機の分散オペレーティング・システムの構想

福田晃　村上和彰　末吉敏則　富田眞治
(九州大学)

1. はじめに

我々は、多様な論理構造をもつ並列処理問題に柔軟に対処できる可変構造型並列計算機の開発を進めている(1)。我々の最終目的は、図1に示すような本計算機を1つのノードとした分散型システムを構築し、分散透明な計算機の利用環境を提供することにある。本稿では、可変構造型並列計算機の分散オペレーティング・システム(OS)の構想について述べる。

2. ハードウェア構成の概要

本計算機は、128台のプロセッシング・エレメント(PE)と相互結合網(クロスバー網)からなる。各PEはプロセッサ・ユニット(PU)、ローカル・メモリ、通信制御装置からなる。クロスバー網はブロードキャストの機能を提供している。

3. 設計方針

本計算機のOSを設計するにあたり、以下の方針をとる。

(1)システム資源を効率的に利用する。

システムの性能を最大限に引き出すためには、システムが提供する資源を効率よく利用する必要がある。その中でも特にPUを効率よく利用することが重要であると考える。

(2)既存のアプリケーション・ソフトの互換性を重視する。

本計算機の開発目的の1つは、研究者に並列処理研究の環境を与えることである。従って、既存の多くのアプリケーション・ソフトの互換性をもたせる必要がある。これを実現するため、UNIX†のシステム・コールの仕様を含んだシステム・コールをユーザに提供する。

(3)OSの実現、変更が容易な構造にする。

OSの開発段階で種々の変更が生じる可能性がある。そこで、これらに柔軟に対処できる構造にする必要がある。

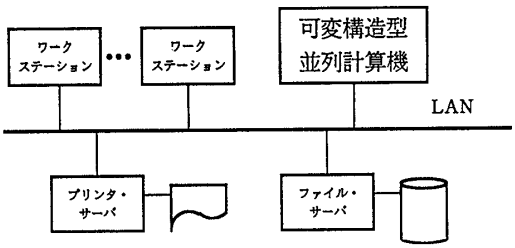

図1 将来システム構想図

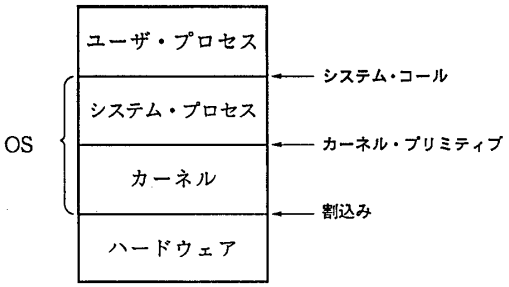

図2 システム構成

4. システム構成

本システム構成を図2に示す。OSをカーネル層とシステム・プロセス層の2つに分ける(2)。システム・プロセス層は、OSの種々の機能をプロセス・レベルで実現する層である。カーネル層は、OSの核となる部分で、システム・プロセス間の通信機能、プロセスのディスパッチなどの基本的な機能を提供する。

5. プロセスとプロセス間通信

(1)プロセスとアドレス空間

通常ユーザ・プロセスは、お互に協調しながら動作している。このとき、協調の度合が強いプロセス同士を密に結合した方が得である。そこで、これらのプロセスを1つのグループとし、アドレス空間を共用させ、アドレス空間の切り換えに伴うオーバヘッドをなくす(3)。グループの指定はユーザが行う。本計算機は各PEに分散しているローカ

Distributed Operating System Philosophy for a Reconfigurable Parallel-Processor

Akira FUKUDA, Kazuaki MURAKAMI, Toshinori SUEYOSHI, and
Shinji TOMITA
Kyushu University

†UNIXは米国ベル研の登録商標である.

ル・メモリを共有メモリとして使用できるので、各プロセスが異なるPE上で実行されてもよい。

(2)プロセス間通信

同一グループ内のプロセス間通信には共有メモリを介した通信を行い、グループの異なるプロセス間ではメッセージ・パッシングを用いる。

6. OSの機能

必要とされるOSの機能の概要を述べる。これには大きく分けて、各PE内の管理・制御を行うローカル管理・制御機構と、システム全体の管理・制御を行うグローバル管理・制御機構がある。

6.1 ローカル管理・制御機構

(1)ローカル・プロセス管理・制御機構

自PEで実行されるプロセスを管理・制御する。これには、スケジューリングやプロセス・テーブルの管理などが含まれる。

(2)ローカル・メモリ管理機構

ローカル・メモリの割当て、解放、空き領域の管理などを行う。

(3)通信管理機構

プロセスおよびPE間の通信管理を行う。

(4)タイマ管理機構

タイマを管理し、タイムスライス、タイムアウト、プロセスの呼び起こしなどに利用される。

6.2 グローバル管理・制御機構

プロセスが協調して動作したり、システム資源を有効に利用するためには、システム全体の管理をどのように実現するかが問題となる。この管理方法に次の2つがある。

1)集中管理方式: グローバル情報を1つの場所で集中して管理する方式である。

2)分散管理方式: グローバル情報を分散させて管理する方式である。

集中管理方式は、分散管理方式に比べて一般的に実現、制御が容易であるが、管理機構がダウンすると致命的な影響を与えるという欠点をもつ。一方、分散管理方式は、通信オーバヘッドや制御方法など難しい問題を抱えているが、フォールト・トレランスなどの面から魅力ある方式である。そこで我々は、分散管理方式を採用する。以下、動的負荷分散方式とグローバル・プロセスの管理方法について述べる。

(1)動的負荷分散方式

システム資源特にPUを有効に利用するためには、プロセスをPE間で移動させる動的負荷分散が重要である。この実現方法を以下に述べる。

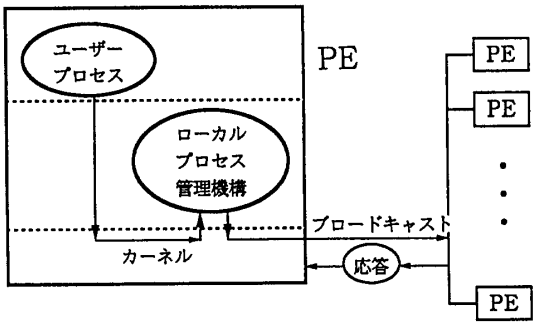

図3 グローバルプロセスの管理方法

各カーネルに全PEの負荷情報をもたせる。これは、各PEが自PEの負荷情報を一定周期毎にブロードキャストすることによって行う。移動の対象となるプロセスは実行待ちでかつ移動許可フラグがオンのプロセスとする。移動許可フラグの設定はユーザ・レベルでも指定できるようにする。自PE上で実行されるまでの時間と、他PE上へ移動させて実行するまでの時間を大まかに推定して、両者の大小比較によって、移動の決定を行う。このとき、負荷情報は周期的に集められたものであるので、負荷の軽いPEに負荷が集中する可能性がある。これを防止するために、各PEは他PEからの移動を受付ける時間間隔をある値(T)以上になるようにする。すなわち、移動を1つ受付けたらそれ以後T時間は受付けを拒否する。これをハードで行うことによって高速化を図る。

(2)グローバル・プロセスの管理

プロセスの移動を行うので、グループの異なるプロセス間の通信を行うためには、通信したいプロセスがどのPEにあるかを知る必要がある。この実現方法を以下に述べる(図3)。

カーネルはあるプロセスから通信相手のプロセス名を受け取ると、ローカル・プロセス管理機構にそのプロセスが自PEにあるかどうかを問合せる。なければ、カーネルはプロセッサ間通信管理機構を介してプロセス名を全PEにブロードキャストすることによって、PE名を知ることができる。

7. おわりに

可変構造型並列計算機の分散OSの構想について述べた。今後、これらの詳細化を進めていく。

参考文献: (1)村上ほか:"可変構造型並列計算機の構想", 情報処理学会マイクロコンピュータ研究会資料, 87-MC-47-2(1987). (2)田胡ほか:"オペレーティング・システムの構造記述に関する一試み", 情報処理学会論文誌, Vol.25, No.4, pp.524-534(1984). (3)Cheriton, D.R.: "The V Distributed System", Commun. ACM, Vol.31, No.3, pp.314-333(1988).