5K－7

# 光バスクラスタ計算機Euphoriaの開発 (3) システム・ソフトウェア

鈴木茂夫　数藤義明　井上淳　佐々木章友　吉本雅彦　柴山茂樹
キヤノン（株）情報メディア研究所

## 1. はじめに

光バスクラスタ計算機Euphoriaは，従来のイーサネットなどのネットワークとは異なり，計算機の内部バスに匹敵する帯域幅を持つ高速ネットワーク（光ファイバ）を活用することで，より高度な分散処理を目指している．本報告では，このEuphoria上で動作するシステムソフトウェアの開発について報告する．

## 2. Euphoriaにおけるシステムソフトウェア

Euphoriaにおいて，光ファイバで結ばれた個々のノードは，それぞれが独立して動作可能なワークステーション（WS）クラスの計算機であり，これらのノードが備える物理メモリは，光バスクラスタ全体で一つの物理メモリ空間を構成している．

このような構成からなるEuphoriaは，そのハードウェア特性を活用するシステムソフトウェアにより，高度な負荷分散処理を実現することで，(1) 中～大規模な並列計算機の様にも稼働し，また，(2) 距離をおいたノード間で，大量のデータ転送を伴って協調利用するようなアプリケーションシステムも実現可能とすることを目的とする．

Euphoriaは，これら多側面での利用を目指しているが，その特徴をフルに発揮するためには，既存のOSの機構だけでは不十分である．そこで，OSを中心としたシステムソフトウェアに対して，Euphoriaに対応した新しい機構を実現し，その特徴を最大限に発揮することを目指している．

本システムシフトウェアの実現により，上位のソフトウェアに対して，クラスタ内の各資源（プロセッサ，メモリなど）が，ノードの枠に捕らわれることなく公平に，システムの負荷に応じて，各ユーザプログラムに割り当てられるような環境を提供する．こうして実現されるプログラム実行環境を，我々はRIDE（Resource Impartially-Distributing Environment）と呼んでいる．

## 3. システムソフトウェアの実現

Euphoriaでは，各ノードごとに個々のシステムソフトウェアが稼働しそれらが協調動作する方式を採用した．個々のシステムソフトウェアは，拡張したMach3.0（CMUが開発）のマイクロカーネル（MK）とシステムサーバ（RIDEサーバ）から構成される．また，これらの開発をハードウェア開発と並行して行なうため，以前我々が開発したマルチプロセッサシステムStonehigh [1] を利用して，仮想的に，光ファイバにより結合された2CPU構成の2台のノードとして扱い，その上でMKの拡張およびRIDEサーバの試作を行なった．

## 4. プロセッサ・メモリ資源管理方式

### 4.1 基本アイデア

Euphoriaのハードウェアの特徴は，他のノード上のメモリをダイレクトにアクセス可能な構成になっているという点にある．この特徴を活かすためには，従来のネットワーク環境上で研究が進められているソフトウェアによる分散共有メモリまたはプロセスマイグレーションのような間接的なリモート資源利用ではなく，より直接的な利用形態が望ましい．

そこで，それを具体化する方法として，メモリ資源だけでなくプロセッサ資源をも，ノード間で受渡し可能とし，他のノード上のプロセッサ・メモリ資源をダイレクトに利用できるようにする手法を考案した（図1）．

このプロセッサ資源受渡し機構の実現は，簡単には，受渡し元ノード上のMK内で，受け渡すプロセッサを停止状態と同様の状態にしてから，そのプロセッサ自身が受渡し先ノード上のMKのコードを実

---

Design and Implementation of an Optical Bus Cluster Computer "Euphoria"
(3) System Software
S. SUZUKI, Y. SUDO, S. INOUE, A. SASAKI, M. YOSHIMOTO and S. SHIBAYAMA
Media Technology Laboratory, Canon Inc.

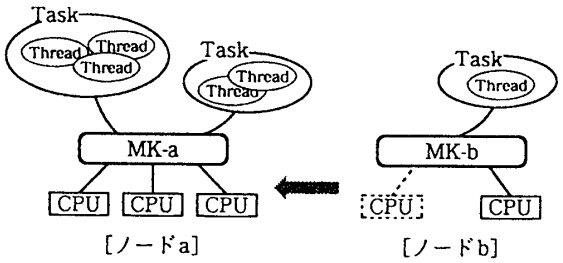


図1　プロセッサ資源の移動

行して，その管理下に入り，通常のプロセッサと同様に，動作可能状態のスレッドを割り当て実行していくようにすればよい．

この機構の実現により，例えば1CPU構成のノードであっても，他のノードから多数のプロセッサの管理を移動してくることによって，大規模な並列処理が，容易にかつ効率よく実行可能となる．

### 4.2 メカニズムとポリシの実現

我々は図2に示す通り，まず前述のプロセッサ・メモリ資源の受渡しメカニズムをMKとRIDEサーバの下位層により実現し，資源割当てのスケジューリングポリシをRIDEサーバの上位層により実現した．このような構成により，様々なポリシをサーバレベルで容易に実現可能となり，いくつかの方式を実現，評価，比較するといった研究をよりスムーズに行なえるようになった．

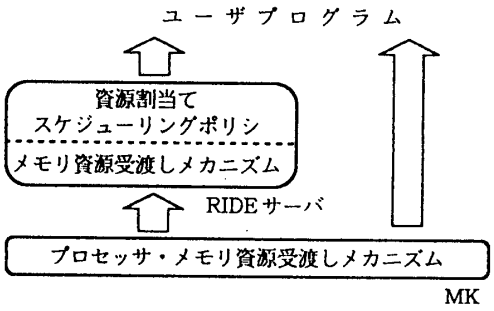


図2　システムソフトウェアの構成

プロセッサ資源受渡しメカニズムは，Mach3.0のプロセッサセット機構の拡張の形で，MKの拡張により実現した．例えば，拡張インタフェースprocessor_borrow()は，指定されたノードから任意個のプロセッサの管理を移動してきてデフォルトプロセッサセットに入れる．その後利用者は，自ノード上にあるプロセッサとまったく同様に，任意のプロセッサセットに割り当て利用することができる．また，メモリ資源受渡しメカニズムは，MKの拡張とRIDEサーバの下位層のメモリマネージャ機構により実現した．これにより，利用者は指定したノードからメモリ資源（ページ単位）を借りて直接利用することが可能となる．

上記のメカニズムの上に構築する資源割当てスケジューリングポリシにより，各ノードのプロセッサの負荷バランス，メモリ資源の使用バランスが自動的にとられることになる．簡単には，例えば負荷の集中しているノードがあると，そこに他の負荷の軽いノードから適当な数のプロセッサの管理を移動して負荷バランスを保つ．また，異なるノード上のプロセッサ，メモリの組み合わせの場合，メモリアクセスにネットワークが介在するため低速になる，という性質を考慮して，タスク実行のためには，できるだけ同一ノード上のプロセッサ・メモリ資源を組にして割り当てるようにした．

### 4.3 ユーザインタフェース

RIDEの利用者は，前述のポリシを利用すると，従来通りのWSの利用方法と同様であるにもかかわらず，実際には資源受渡しメカニズムが自動的に起動され，理想的なプログラム実行環境で目的のプログラムが実行されることになる．また，こうした利用方法とは別に，前述の各メカニズムを利用者が直接呼び出して，陽に他ノードのプロセッサ・メモリ資源を利用して，よりlow-levelなプログラミングも可能となる．このように，RIDEでは利用者の利用レベルに応じた柔軟なインタフェースを提供する．

## 5. おわりに

Euphoriaにおけるシステムソフトウェアは，拡張したMach3.0のMKとRIDEサーバから成る．これらにより，クラスタ内のプロセッサ・メモリ資源が直接的にノード間で受け渡され，各ノードのプロセッサ負荷バランス，メモリ使用バランスが保たれ，高効率のプログラム実行環境を提供できる．ハードウェアの完成に引き続き，このシステムソフトウェアを実機に実装し，評価を行なう予定である．

## 参考文献

[1] 伊達 他,「マルチプロセッサワークステーション“Stonehigh”—コンセプトとハードウェア概要—」，第45回情報処理学会全国大会6L-02，1992.