# 知識ベースマシンＭｕ－Ｘ（１）

## －　並列制御方式　－

酒井　浩，仲瀬　明彦，柴山茂樹
（株）東芝　総合研究所

物井　秀俊，伊藤　英則
（財）新世代コンピュータ技術開発機構

### １　はじめに

通産省第５世代コンピュータプロジェクトの一環として，知識ベースマシンＭｕ－Ｘ[1] の試作が進められている。この「知識ベースマシン」という用語は，未だ明確な定義がなされていないが，Ｍｕ－Ｘでは第５世代コンピュータらしい特徴を備えたデータベースマシンととらえ，複数の推論マシンと結合した形態の知識情報処理システムを目標としている。

Ｍｕ－Ｘはマルチプロセッサシステムであり，並列処理によりデータベース演算を高速化するため，種々の工夫がなされている。本稿では，Ｍｕ－Ｘにおける並列処理制御方式について述べる。

### ２　Ｍｕ－Ｘの目標と設計方針

#### ２．１　Ｍｕ－Ｘの目標

Ｍｕ－Ｘの目標は，第５世代コンピュータらしい特徴を備えたデータベースマシンの実現である。そのため情報の表現能力を推論マシンに近づける目的で，データや問合わせ等は項で表現することにした。そして，それらに対する演算として項間の単一化やその他のPrologの評価可能述語を採用した。また，ホストマシンとしてＰＳＩを想定し，マルチトランザクション環境の実現を目指した。問合わせの傾向としては，一貫性のチェックなど処理負荷の重い演算と電子化辞書の見出し語検索など処理負荷の軽い演算が混在することを想定し，レスポンスタイムとともにスループットを重視することにした。

#### ２．２　Ｍｕ－Ｘの設計方針

Ｍｕ－Ｘの設計にあたっては，このように機能面で高い目標を実験機規模で実現するため，図１に示すように汎用マイクロプロセッサを処理要素(PE: Processing Element)とするマルチプロセッサ構成をとり，ソフトウェア指向のシステムとした。アーキテクチャの特徴は，各ＰＥにディスク装置をつけてデータベースシステムの課題とされる主記憶とディスク間のI/O ボトルネック解消を図ったこと，ＰＥ間の通信等に必要な共有記憶を２種類のメモリ（ワードアクセス可能な通常の共有メモリと高速ページアクセス

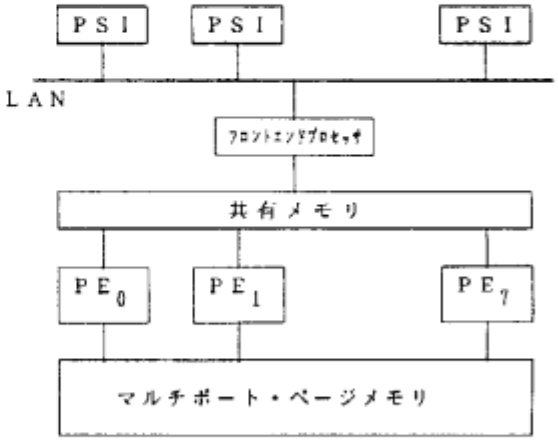


図１　Ｍｕ－Ｘのハードウェア構成

可能なマルチポート・ページメモリ）のハイブリッド構成とすることにより，データの参照特性に応じた使い分けを可能にしたことである。

### ３　ＰＥ間の機能分担と並列制御方式

#### ３．１　ＰＥ間の機能分担

一般にデータベースシステムで行なうべき処理機能は，ホストマシンから到着するキュエリの解析やデータ辞書の更新を行なう知識管理機能とリレーションに対する演算を処理する知識演算機能に分けることができる。従来のマルチプロセッサ構成のシステムでは，各々のＰＥにどちらか一方の機能を割当てることが多かった[2]。しかしながらＭｕ－Ｘでは，各々のＰＥに両方の機能を分担させることにした。これは，２．１で述べたように問合わせの傾向として，知識管理機能に比べ知識演算機能が重い演算と軽い演算が混在するとの想定に基づいて，その比率が変動した場合でも安定した性能が発揮できるようにするためである。

あるホストマシンから到着するキュエリの解析は，ある特定のＰＥで処理することにより，ホストマシンにローカルな情報は対応するＰＥの内部で保持することにした。これは，アクセスネックとなりやすい共有メモリに置くべき情報を減らすためである。一方，各ＰＥの知識演算機能は

---

Knowledge Base Machine Mu-X  Control Mechanisms for Parallel Processing
Hiroshi SAKAI[1], Akihiko NAKASE[1], Shigeki SHIBAYAMA[1], Hidetoshi MONOI[2], Hidenori ITOH[2]
[1] TOSHIBA Corporation.  [2] Institute for New Generation Compter Technology

すべてのホストマシンからのキュエリに応じた処理を担当させることにした。そのため，知識管理機能で並列処理用内部コマンドを共有メモリ上に作成し，各ＰＥの知識演算機能ではその指示に基づいた処理をする方式を採用した。

3．2　並列処理制御方式に関する設計方針

先に述べたように，Ｍｕ－Ｘでは内部コマンドを用いて演算の並列処理を実現することにしたが，細部の設計ではキュエリ到着からレスポンス作成までの総処理量を小さくすることを目標に，次のような点に留意した。

(1) ある演算を実現に要するコマンド数を減らす。
(2) 並列処理アルゴリズムはＰＥ間の同期処理のコスト等も考慮して取捨選択する。
(3) 複数ＰＥで同じ処理の重複が起こらないようにする。

## 4　並列処理用内部コマンドの概要

### 4．1　コマンドの意味

並列処理コマンドの意味として次の２方式を取り上げ，その得失を検討した。

(1) ひとつのコマンドで，ＰＥ１台がなすべき演算を規定する。
(2) ひとつのコマンドで，ＰＥのグループがなすべき演算を規定する。

その結果，(1) の方式を採用すると，少なくとも処理にかかるＰＥ台数だけコマンドを生成する必要があり，生成する側の処理負荷が大きくなること，各コマンドの処理の所要時間に違いがあるとＰＥ間の負荷分散がうまく図れないという短所があることがわかり，そのような欠点のない (2) の方式を採用することにした。一般に，メッセージ通信によりＰＥ間の同期をとるシステムでは (1) の方法をとらざるを得ないが，Ｍｕ－Ｘのように共有メモリを用いるシステムでは (2) の方法をとることができ，その点でＭｕ－Ｘは共有メモリ方式の利点を活かしているといえる。

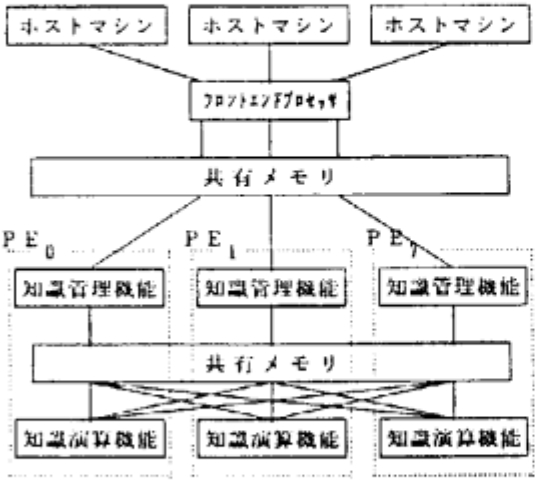


図２　Ｍｕ－ＸのＰＥ間の機能分担

### 4．2　ＰＥ指定コマンドとＰＥ無指定コマンド

Ｍｕ－Ｘでは並列処理の対象となる関係は，水平分割によりＩＥのディスクに分散格納されるか，マルチポート・ページメモリ上に格納されるかのいずれかとした。そのため，演算対象が前者の場合はディスク装置を有するＰＥに処理を委ねる必要があるが，後者の場合にはどのＰＥでも処理可能である。そこで，内部コマンドでは，処理すべきＰＥをビット列形式で指定するＰＥ指定コマンドと，同時実行可能なＰＥ台数の上限だけを指定するＰＥ無指定コマンドを用意した。

また，ひとつの内部コマンドを複数ＰＥで処理する場合，処理の開始／終了のタイミングが各々のＰＥでまちまちである。そこで内部コマンドに関わるすべての処理の終了を効果的に判定する手段が必要となる。そのためコマンドごとに各ＰＥによる処理の開始／終了を示すビット列を用意した。そして，ＰＥ指定コマンドでは要求ビット列と終了ビット列の一致により，またＰＥ無指定コマンドでは開始ビット列と終了ビット列の一致によりコマンドとしての終了を判定するようにした。

### 4．3　並列処理の粒度とオブジェクトカウンタ

ＰＥ無指定コマンドでは，4．1で述べたように各ＰＥがどのオブジェクトの処理をすべきか前もって決めないで，あるオブジェクトの処理が終了したものから順に次のオブジェクトの処理を開始するようにした。この処理の粒度は，選択演算ではページとした。また，結合演算ではネステッドループで行なう場合はページとリレーション全体の結合演算，動的クラスタリングを前処理として行なった場合は対応するバケット間の結合演算とした。

また，Ｍｕ－Ｘの内部コマンドでは各ＰＥが動的に処理対象を獲得する方式であるため，その処理を僅かなオーバヘッドで実現することが望まれる。そこでコマンドのパラメタとして次に処理すべき対象を示すカウンタを用意し，各ＰＥが処理対象を獲得するごとにカウンタの値をその分だけ更新する方式を採用した。

## 5　おわりに

Ｍｕ－Ｘでは，各ＰＥに知識管理機能と知識演算機能を持たせ，両者の負荷バランスが変動する場合でも安定した性能を目指している。また，両機能の間で行われるデータベース演算の並列処理の制御は，共有メモリ上に作成する内部コマンドを用いて実現している。今後，システム全体の評価を通じて方式の有効性を確認する予定である。

参考文献

[1] 酒井　他：知識ベースマシンＭｕ－Ｘ（３）　並列処理のための基本機能”，第３６回情処全国大会予稿

[2] Su. S. Y. W.: Database Computers Priciples. Architectures & Techniques". McGraw-Hill. 1988.