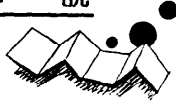


## 解説



## C.mmp の 評 価†

杉 本 正 勝††

## 1. はじめに

C.mmp は、米国のカーネギメロン大学が研究・開発したマルチプロセッサ・システムである。研究・開発は 1971 年に開始された。

このマルチプロセッサは次の特徴を持っている。

- (I) 16 台までのプロセッサが共有メモリにアクセスする密結合マルチプロセッサ (Tightly coupled multi-processor) である。
- (II) システムに課せられる負荷を複数のプロセッサで分散処理する負荷分散マシンである。
- (III) プロセッサとしてはミニコンピュータのプロセッサを使用している。

C.mmp の設計の目的は以下の通りである。

- (I) 複数のプロセッサが共有メモリにアクセスする形式のマルチプロセッサの技術を開発すること。
- (II) マルチプロセッサについて出来るだけ多くのことを学ぶこと。

1971 年当時は、マルチプロセッサの科学的性質、性能、問題点がほとんど知られていなかった。そこで、システムを実際に作ることで、これらのことを明確にしようと考えた訳である。

1977 年には、システムはほぼ完成の状態となった。現在までに判明したマルチプロセッサの利点は次の 4 点である。

- (I) システムの処理能力、特にスループットの向上
  - 一台のプロセッサの能力には限界があるが、これを多数台並列に動作させることで、処理能力、特にスループットを向上できる。
- (II) 性能・価格比の向上
  - 比較的安価なプロセッサを多数個使用して所望の性能を実現する可能が生まれる。

## (III) 信頼度の向上

同一種類のプロセッサが複数台あるので、どれかが故障したときでも他のプロセッサで代用ができる。

## (IV) 性能の増加が容易

システムに要求される処理能力がシステムの導入時点には不明であったり、システムの運用時に性能を増加したい場合に、プロセッサの台数を増加して漸次システムを拡大できる。

C.mmp とそのオペレーティングシステムである Hydra は、その運用経験にもとづく興味ある結果と重要な結論が集まる時点に達している。米国では C.mmp と Hydra プロジェクトを評価する専門委員会の報告書も公表されている<sup>4)</sup>。

ここではまず C.mmp を簡単に紹介し、次にこのシステムを評価する。

2. システムの構成と動作<sup>1),2)</sup>

C.mmp の構成は図-1 に示すように複数のプロセッサが共有メモリにアクセスする形式である。

C.mmp と図-1 との相違は、C.mmp では独立に動作するメモリ・モジュールが多数個あり、それらがスイッチを通してプロセッサに結合されている点である。

このようにした理由は、複数個のプロセッサが同時にメモリにアクセス出来る可能性を増加するためである。複数台のメモリ・モジュールを使用すれば、プロセッサからの要求が分散され、メモリに同時にアクセス出来る可能性が増加する。

図-2 は C.mmp の詳細なハードウェア構成図である。システムは共有メモリ群 (Mp (0)~Mp (15)) と

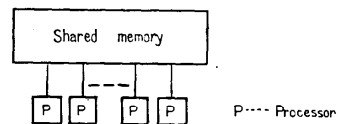


図-1 メモリ共有のマルチプロセッサの一般形式

† Evaluation of C.mmp by Masakatsu SUGIMOTO (System Design Department, System Development Division, Fujitsu Ltd.).

†† 富士通(株) 開発事業部第一方式部

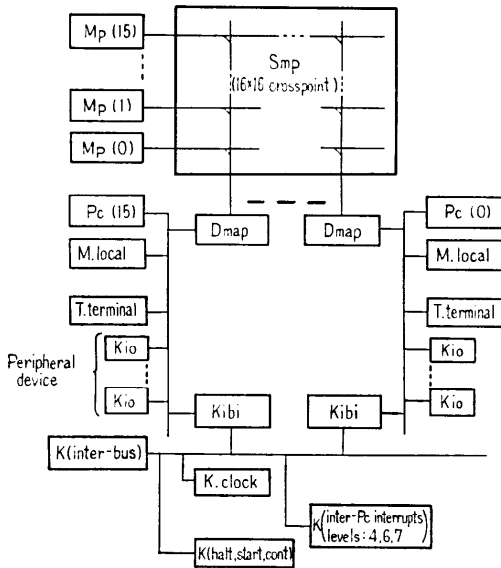


図-2 C.mmp の構成

プロセッサ群 (Pc (0)~Pc (15)) が Smp で結合されている。

プロセッサ間の通信の制御にはプロセッサ間バス (K(inter-bus)) が用いられている。

### 2.1 構成要素の説明

#### (1) プロセッサ (Pc)

プロセッサはミニコンピュータのプロセッサである。PDP-11 のモデル 20 とモデル 40 に若干の改造を加えたものを使用している。

#### (2) メモリモジュール (Mp)

メモリモジュールは最大 16 台取り付け可能である。1つのモジュールの容量は最大 2MB まで増加できる。システムとして最大 32MB までのメモリを実装できる。

サイクル・タイムは MOS メモリを使用した場合 200ns である。

#### (3) クロスポイント・スイッチ (Smp)

16台のプロセッサを16台のメモリモジュールに接続する機能を持つ。Pc(i)とMp(j)との関係は完全群である。クロスポイント・スイッチを通るバスは、メモリ要求があるごとに独立に確立される。全プロセッサが別個のメモリモジュールを参照すれば、16個のプロセッサは同時に働く。

クロスポイント・スイッチは優先度決定回路、メモリインタフェース等から構成されている。クロスポイント・スイッチにおけるオーバーヘッドは 200ns であ

る。

#### (4) Dmap

ユニバス上の 18 ビットのアドレスを 25 ビットのアドレスにマッピングする。この 25 ビットのアドレスが主メモリのアドレスとなる。

#### (5) プロセッサ間バス (K(inter-bus))

プロセッサ間の通信などの目的で用意されている。これについては次に詳細に述べる。

### 2.2 プロセッサ間バス (K(inter-bus))

プロセッサ間通信はマルチプロセッサ・システムの主要なポイントである。

C.mmp ではプロセッサ間バスを用いている。プロセッサ間バスの主要な機能を 3 つ以下に示す。

#### (1) 機能レジスタ

プロセッサ間のインタフェースとして 6 つのレジスタがある。これらのレジスタの幅は 16 ビットであり、各ビットは 16 台のプロセッサの 1 台に割り当てられる。

(i) HALT このレジスタのビットを 1 にセットすると、そのビットに対応したプロセッサが HALT 状態になる。

(ii) START プロセッサのスタート・ボタンに相当する。このレジスタがセットされると、対応するプロセッサのスタート・アップ・プログラムが起動される。

(iii) CONTINUE このレジスタがセットされると、HALT 状態にある対応プロセッサが動作を再開する。

#### (iv) プロセッサ間割り込みレジスタ (3 個)

C.mmp のプロセッサ間通信では、プロセッサがデータを直接送り合うことはしない。データを共有メモリに置いた状態で、プロセッサが他のプロセッサに割り込みをかけることで通信する。

#### (2) グローバル・クロック

#### (3) インターバル・クロック

### 2.3 システムの現状

1977 年 1 月の時点で、5 台の PDP 11/20 (5  $\mu$ s/命令) と 5 台の PDP 11/40 (2.5  $\mu$ s/命令) が結合されていた。

その後、PDP 11/20 5 台と PDP 11/40 11 台、計 16 台が結合された状態になった。主メモリの合計は約 3MB である。総合実行速度は約 6MIPS である。

C.mmp は ARPANET に接続されているので、カ

一ネギメロン大学以外からも広く使用されている。

システムの価格は約 60 万ドルである。プロセッサの価格が 30 万ドル、メモリの価格が 20 万ドル、残りの 10 万ドルはクロスポイント・スイッチ、プロセッサ間バス、その他の特殊装置の価格である。このシステム価格には I/O 装置、周辺装置関係の価格は含まれていない。

### 3. Hydra<sup>1),3)</sup>

C.mmp のオペレーティングシステムの核 (kernel) の部分は Hydra\* と呼ばれている。

Hydra に関して主要な研究項目は次の 3 つであった。

(1) マルチプロセッサ向きオペレーティング・システム

- (i) マルチプロセッサ・システムを使いこなすために、ユーザに提供すべき機能は？
- (ii) 分散型 OS (OS がどのプロセッサ上でも動き、また同時に複数個動ける) とマスタースレーブ型の OS とでいずれがよいか？
- (iii) マルチプロセッサ向きのスケジューリング・アルゴリズムやページング・アルゴリズムは？

(2) 拡張可能 OS

(1) の問題に取り組むために、OS を拡張可能な構造にして、核 (kernel) の部分を最初に作ることにした。

(3) ソフトウェア工学の実験

Hydra は分散型 OS を採用した。その理由は「OS の信頼度の向上のためには、各プロセッサを機能別にすべきではない。分散型 OS は、共通データへの相互排他的アクセス機構を導入する必要があるが、マスタースレーブ型の OS に比較して特に困難ではない。」という判断による。

図-3 は、Hydra の下でのユーザ環境を示す。中心に OS の核の部分があり、それに加えて多くのサブ・システム (スケジューラ、ファイル・システム、ディレクトリ・システム等) がある。

例えば、この図で SCHED 1 と SCHED 2 は異なるポリシーに基づくスケジューラである。ユーザはこれらのサブシステムのグループの内から自分に適したものを選択することが出来る。複数のユーザが同一の

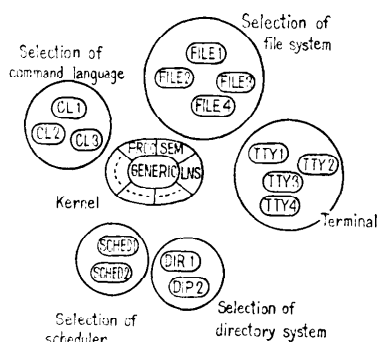


図-3 Hydra のユーザ環境

サブシステムを選択してもよいし、また自分の欲するものがなければ、新しいソフトウェアを追加することも出来る。

核には次に示すような OS の基本機能が用意されている。

- (i) マルチプロセッサにプロセスを分配する機能 (PROCESS)
- (ii) 同期機能 (SEMAPHORE)
- (iii) オブジェクトの保護機能 (Capability-based 形式の保護機構を採用)

以上に述べたように、Hydra では OS の基本機能のみを用意し、スケジューリング・アルゴリズム、ページング・アルゴリズム、ディレクトリ・システム等のポリシーに関する機能の実現は、ユーザ・レベルのプログラムとして任意に追加可能としている。ここでユーザとはマルチプロセッサの研究者、オペレーティング・システムの研究者である。

OS の核である Hydra に加えて以下のソフトウェアを作成済みである。

- (i) ファイル・システム
- (ii) ディレクトリ・システム
- (iii) スケジューラ
- (iv) 言語プロセッサ (ALGOL-68, L\*, コマンド言語)

特に、システムのユニークな特徴である ALGOL-68 を使用するユーザが多い。この ALGOL-68 では並列処理の記述が出来る。

### 4. C.mmp の性能<sup>1)~4)</sup>

C.mmp 形式のマルチプロセッサで性能が向上するか?」は興味のある問題である。この問題に直接関係する事項は次の 2 点である。

\* Hydra とはギリシャ神話の中に出てくる怪物の名前で、沢山の頭を持っており、どれか 1 つを切り落とすと、そこから 2 つの頭が生えて来るという怪物のことである。

(1) マルチプロセッサにおけるメモリ・コンテンション

(2) ソフトウェアのオーバーヘッド

メモリ・コンテンションについて C.mmp の研究者は多くの研究を行ってきた。その結論は「メモリ・コンテンションによる性能の低下は 10% 以下」であった。性能低下の最大の要因はメモリのバンド幅である。

C.mmp と Hydra の開発を開始した時点では、ソフトウェアのオーバーヘッドについてはよく分かっていなかった。

図-4 は PDE (偏微分方程式) の問題における Pc の数と実行速度の関係を示す (PDE の 18 プロセスでの解法を利用している)。この図はマルチプロセッサの適用の効果を示すものである。更に、C.mmp の運用を通じて次のことが分かっている。

(i) C.mmp で解こうとする問題がうまく並列プロセスに分解出来る場合、マルチプロセッサの利点を生かせる。

(ii) あるベンチマークの問題に対して、C.mmp のクロスポイント・スイッチによる遅れと Hydra のオーバーヘッドとの合計は、スタンダードの PDP-11 に対して 25% あったが、(i) による性能の向上でこのオーバーヘッドを十分補うことが出来た。

(iii) 分散型 OS でよく指摘される点は、共通データに対するロックの衝突による処理の遅れである。C.mmp と Hydra の下でハードウェア・モニタを使用して調査した結果、ロックの衝突による処理の遅れは 1% 以下であり特に問題にならないことが判明した。

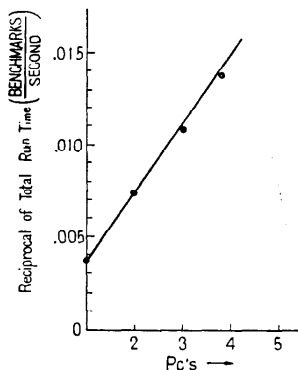


図-4 PDE のベンチマークの実行速度

カーネギメロン大学の研究者は、C.mmp と C.mmp に相当する性能を持った単一プロセッサである PDP-10 との性能・価格比を調査している。

PDP-10 については、次の 3 つのコンポーネントを考慮に入れている。

- (i) CPU
- (ii) 主記憶
- (iii) チャンネル

C.mmp については、次の 3 つのコンポーネントを考慮に入れた。

- (i) PDP-11/20 または PDP-11/40 のプロセッサ
- (ii) 主記憶
- (iii) C.mmp に特有なハードウェア
  - (a) クロスポイント・スイッチ
  - (b) PDP 11/20 または PDP 11/40 の改造費

上記のコンポーネントの市場価格を用いて性能・価格比を求めた。その結果、C.mmp は PDP-10 より 4 倍以上の性能・価格比であるとしている。

しかし、一方では「市場価格は販売上の政策で決定される性格が強いのので、上記の結論は真の値を表わしていない」という指摘がされている<sup>3)</sup>。

今後のマイクロプロセッサの性能・価格比の向上、LSI 化された主メモリの価格の減少の速度が大であることを考慮すると、C.mmp と同様のシステムを新たに設計するときには、システムを利用する時点に焦点を合せて性能・価格比を計算して評価する必要がある。

## 5. 評価

C.mmp と Hydra のプロジェクトを評価する専門委員会が米国で組織され、報告書が公表されている<sup>4)</sup>。この委員会は C.mmp と Hydra の設計者、製作者、利用者の各グループの代表者から構成された。ここでは、この報告書に沿って C.mmp と Hydra を考察する。

このプロジェクトで成功した主要な点は次の 5 つである。

- (1) 経済的な対称型マルチプロセッサを実現したこと
- (2) Hydra は capability-based 形式の保護機構を採用し、ユーザ・レベルのプログラムとして OS の機能を実現し得たこと
- (3) 分散型 OS を実現出来たこと
- (4) ソフトウェア、ハードウェアのエラーの検出と回復機構を準備したこと

(5) OS の核を作成するための効率的なソフトウェアの開発法

一方、明らかに失敗した点は次の5つである。

- (1) ハードウェアの信頼度が予想外に低く、MTBF が小であること
- (2) PDP-11 の小さなアドレス空間がプログラムの構造や性能に非常に大きなマイナスの要因を与えたこと
- (3) C.mmp のプロセッサとメモリ・モジュールを分割して、2つ以上の独立したシステムとする機能が実現されなかったこと
- (4) 研究プロジェクトの管理に十分な配慮がなされなかったこと

以上の項目から特に負荷分散型のマルチプロセッサ構成と関係が深いものについて次に考察する。

### 5.1 経済的なマルチプロセッサ

多くのユーザは各自の問題のアルゴリズムを複数のプロセッサで並行処理が出来るように再構成することで、大きな性能を得ている。複数プロセッサで並行処理が出来る問題を持ったユーザには C.mmp は魅力的なものに見える。

マルチプロセスで走行するジョブで同一コードが参照され、大きなメモリ・コンフリクトが発生する場合がある。

これに対しては、プロセッサごとにバッファ（キャッシュ）を用意することで十分解決された。このバッファは命令コードのようにリードオンリの情報のみを格納する方式を採用することで、マルチプロセッサ構成で通常大きな技術上の問題となる「バッファ合せ」を不要としている。

PDP-11 のアーキテクチャでは各 I/O 装置を特定の1個のプロセッサに対応づけられるようになる。これが対称型マルチプロセッサの実現上の問題点になると考えられていたが、この点はソフトウェアによる機能でユーザが意識しなくてもよいように出来た。

### 5.2 分散型 OS

I/O 装置のサポート・プログラムを除いて、すべてのシステム・タスクはどのプロセッサでも実行しうるようにしている。その結果、効率のよい同期化の手法が必要となったが、この点に関しては次の方式を採用して成功している。

- (1) コードに対してではなく、データ（共通データ）に対して同期をとる
- (2) 同期化のためのプリミティブとして、高速な

ロック機構と、低速なセマホア機構の両者を採用した。結果として単純な構造の分散型 OS が実現されている。

### 5.3 小さなアドレス空間

ミニコンのプロセッサ PDP-11 を使用しているので、ユーザ・プログラムは1時点には8ページにアクセス出来るだけである（1ページは8KB）。プログラムで8ページ以上をアクセスする場合にはリロケーション・レジスタのスワップが必要となりオーバーヘッドとなる。

OS のサブシステムは、ほとんど64KBに納まるのであるが、リスト処理言語である L\* を用いて大きなメモリ空間をアクセスするときには大きな支障となっている。

### 5.4 総合評価

C.mmp と Hydra では対称型でかつ汎用のシステムを実現している。

対称型とは、プロセッサ群の各プロセッサがそれぞれ対等に負荷を処理することである。

汎用システムとは、単一ジョブを並行処理して高速化することと、独立したジョブの集合に対する処理のスループットを改善することの両方が出来ることである。

単一ジョブの並行処理については次に示すような今後解決すべき問題がある。

- (i) 一般のユーザ・プログラムの大部分は、並行実行可能な多数のプロセスに分解するのが困難である。
- (ii) 並行実行可能な多数のプロセスに分解出来る問題でも、与えられたユーザ・プログラムをこれらのプロセスに機械的に分解する有効な手法がない。
- (iii) 並行実行されるユーザ・プログラムを効果的にデバッグする手法の研究が十分ではない。

以上の点から C.mmp 型のシステムは、次の2つの処理に適していると判断される。

- (1) 独立したジョブの集合に対する処理のスループットの向上
- (2) 並行実行可能なプロセスにうまく分解できる特別な問題の高速処理

また、高信頼度なシステムの実現の観点から C.mmp のハードウェア構成を見直す必要がある。C.mmp の場合は期待外れであったハードウェアの信頼度の低さの原因は使用した単体プロセッサの信頼度に

よるものである。現在のマイクロプロセッサをはじめとする LSI 化されたプロセッサでは、この信頼度の問題は既に解決されていると考えられる。

Hydra に見られる核構成の OS, すなわち核の部分とそれ以外のポリシーの部分とを分離するという思想は、分散型 OS の研究・開発には有効なものとして評価すべきであろう。また、capability-based 保護機構は計算機のセキュリティの強化の面で、大いに評価すべき点である。

C.mmp では大きな問題となった「小さなアドレス空間」の問題も、現在ではメガバイト級のアドレス空間を持つミニコンピュータが使用出来るので、今後は特に問題にはならなくなる。

## 6. おわりに

以上、C.mmp の紹介と評価を行った。なお、C.mmp の参考文献のリストは文献4)に詳細なものが記載されている。

C.mmp を開発したカーネギメロン大学は 1975 年に Cm\* と呼ぶシステムの具体的な設計を開始してい

る<sup>5)</sup>。Cm\* は C.mmp の研究成果に基づいた、より大規模な分散型システムであり、今後の展開が注目されている。

## 参考文献

- 1) Wulf, W. A. : コンピュータ・コンプレックス・システムとソフトウェア・エンジニアリングに関する動向, 日本電子工業振興協会, pp. 23-60 (1977).
- 2) Fuller, S. H. : Price/Performance Comparison of C.MMP and PDP-10, Proc. 3rd Annual Symposium on Computer Architecture, pp. 195-202 (1976).
- 3) 分散処理計算機システムの動向——新技術動向調査報告——, 日本電子工業振興協会, pp. 138-162 (1977).
- 4) Wulf, W. A. and Harbison, S. P. : Reflections in a Pool of Processors—An Experience Report on C.mmp/Hydra, Proc. NCC, pp. 939-951 (1978).
- 5) 計算機システム技術の最近の展開, 日本電子工業振興協会, pp. 166-175 (1978).

(昭和 54 年 1 月 12 日受付)