Vol. 36 No. 8　　情　報　処　理　　Aug. 1995

解　説

日本におけるオペレーティングシステム研究の動向

2.3 並列計算機 TOP-1 の OS†

河内谷 清久仁††

1. はじめに

TOP-1（TOkyo research Parallel processor-1）プロジェクトは，日本アイ・ビー・エム(株)東京基礎研究所で，マルチプロセッサ・アーキテクチャの設計項目の評価および，実際の並列ハードウェアを用いての各種並列処理ソフトウェアの研究を目的として行われたプロジェクトである[1]．当プロジェクトは，ハードウェアの開発からオペレーティングシステム，言語サポートまでを含めたものであるが，本稿ではTOP-1上で動作するOSを中心に解説する．

図-1にTOP-1ハードウェアの外観を示す．そのシステム構成は**図-2**のようになっている．TOP-1は，10台のプロセッシング・ユニット(PU)をスヌープ・キャッシュを介して共有バスで結合した，いわゆるUMA（Uniform Memory Access）型の並列計算機である[2)~4)]．各PUにはIntel i386DXと128Kバイトのキャッシュが装備されており，キャッシュ間の一貫性はハードウェアによって保たれる．PU間の同期には，バス・ロックとメッセージ割込みの機能が利用できる．TOP-1から制御できるI/Oデバイスとして4台のSCSI HDが用意されているが，このHDに直接アクセスできるのは，HD制御(HDC)カードに接続された特定のPU（図のPU 0）のみである．その他のI/Oや外界へのアクセスは，フロントエンド・プロセッサとして接続されているIBM PS/55を介して行われる．このPS/55からは，TOP-1上の共有メモリにアクセスしたり，各PUとの間でメッセージ割込みをかけることができる．

TOP-1プロジェクトでは，このハードウェア上で様々な並列処理の実験を行うためのベースとしてOSの開発を行った．開発にあたっては，以下の点を特に考慮している．

- 既存OSとの互換性から，Unix系のOSを採用
- TOP-1ハードウェアの評価機能のサポート
- OS自身の評価機能を設け，OSの構成法を検討
- 並列プログラミングのためのサポート

次章以降では，OS開発の際の問題点と，TOP-1上で稼働しているOSの構成について述べる．

2. OS並列化の問題点

Unix系のOSを並列マシンで動かす場合の最大の問題は，本来リエントラントになっていないOSカーネルをどうやって並列対応にするかという点にある．一般にUnixのカーネルは実行途中でプリエンプトされないことを前提として書かれているため，複数のPU上で同時にカーネル・コードが実行されると，内部のデータ構造が破壊さ

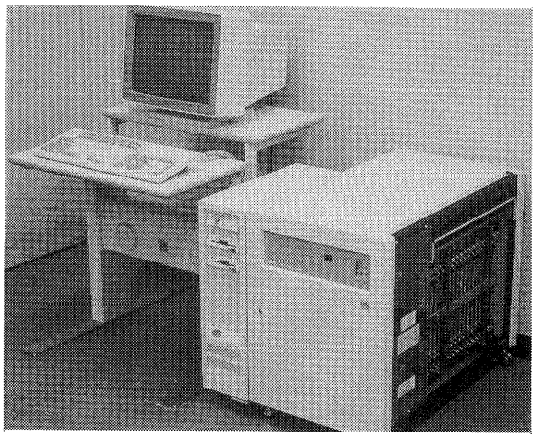


図-1

† Operating Systems on the TOP-1 Multiprocessor by Kiyokuni KAWACHIYA (IBM Research, Tokyo Research Laboratory).

†† 日本アイ・ビー・エム(株)東京基礎研究所

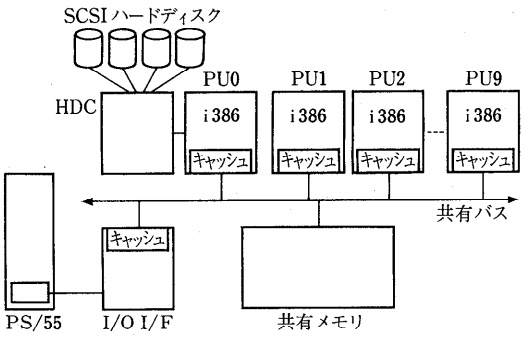


図-2

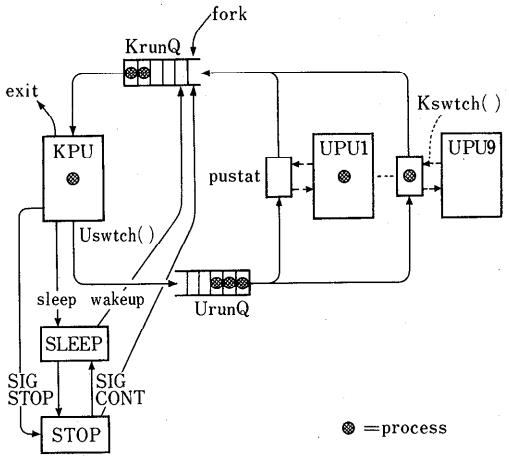


図-3

れてしまう．この問題を解決するには，各データへのアクセスに対して排他制御を行う必要がある．

Unix カーネルの並列化は，この排他制御のレベルによって，対称型とマスタ・スレーブ（非対称）型に大別できる．対称型では，カーネル内のデータ構造それぞれにロックを設け排他制御を行う．これに対し，マスタ・スレーブ型ではカーネル全体に対して排他制御を行う．複数のプロセスがカーネル・コードを実行しようとしたとき，そのうちの１つだけがカーネルに入ることを許可され，残りはカーネルの入口で待たされる．

マスタ・スレーブ型は実現が比較的容易である反面，カーネルの実行がシリアライズされるため，それがシステム全体のボトルネックになる危険性がある．対称型ではその心配はないが，複雑なカーネル・コード全体からロックすべきデータ構造を切り出し，しかもデッドロックなどが生じないように排他制御を行うには，かなり大がかりな変更が必要と考えられる．TOP-1 上では，まずマスタ・スレーブ方式の TOP-1 OS が開発された．続いて対称型マルチプロセッシングをサポートした OSF/1 TOP-1 が開発されている．

3. TOP-1 OS

TOP-1 上の OS 開発にあたってはまず，確実に動作する並列 OS を短期間で作成する必要から，マスタ・スレーブ方式で既存 Unix の並列化を行った．これが TOP 1 OS である[5]．TOP-1 OS は，IBM の i386 版 Unix である AIX PS/2 をベースにしており，OS のカーネル・コードは特定の PU（KPU: Kernel PU，図-2 の PU 0）でのみ実行される．残りの PU（UPU: User PU，図-2 の PU 1～PU 9）ではプロセスのユーザ・コード部分が並列に実行される．

3.1 TOP-1 OS の開発

TOP-1 OS の最大の特徴はそのプロセス管理機構である．Unix プロセスは通常，ユーザ・コードを実行している状態（ユーザ・モード）とカーネル・コードを実行している状態（カーネル・モード）の二状態を行き来しながら動作している．TOP-1 OS では，プロセスのカーネル・モード部分は KPU で実行され，ユーザ・モード部分は複数ある UPU の一つで実行される．そのため，モードが切り替わるときに同時に PU も切り替える必要がある．このモード切替え用に，KrunQ，UrunQ という 2 つのプロセス・キューを用意してプロセス管理を行っている．これは一般の Unix の runQ（runnable process queue）を拡張したもので，KrunQ には KPU での実行を待っているプロセスが，UrunQ には UPU での実行を待っているプロセスが入れられる．**図-3** に，これらのキューの扱いも含めた TOP-1 OS におけるプロセスの状態遷移の様子を示す．PU 間のプロセスの受渡しは，図中の pustat（PU status）構造体とメッセージ割込みを用いて行われる．

マスタ・スレーブ型の並列化では，大部分のカーネル・データに関しては排他制御は不要であるが，UPU でのプロセス実行時に使用される一部のデータに関しては排他制御などの対応が必要になる．そのため，TOP-1 OS ではメモリ管理機構，クロック処理，シグナル処理などの部分にも

変更が行われている．クロック処理の部分にはさらに，TOP-1キャッシュの統計情報収集のための機能が追加されている．

3.2 並列プログラミングのサポート

TOP-1 OSにおける最もプリミティブな並列実行単位はプロセスである．1つのアプリケーションが多数のプロセスをforkし，それらがUPU上で同時実行されることで並列性が生まれる．プロセス間のデータ共有にはSystem V系のUnixで用意されている共用メモリ・セグメントが利用できるが，その他に新しく追加したsharedforkというシステムコールを利用することもできる．sharedforkでは，プロセスのデータ空間を共有したままforkが行われる．スタック空間は共有されず，通常のforkと同様プロセスごとに別のものが用意される．プロセス間の同期に関しては，System V系Unixのセマフォの他，共有メモリ上に用意した同期変数に対してBusy Wait方式の同期をとる方法も利用できる．

これらの機能は非常に低レベルのものであり，アプリケーションを書く際に使いやすいものとはいえない．そのためTOP-1プロジェクトでは，並列プログラミング環境の研究も同時に行われている．C言語でのプログラミングには，前述のプリミティブを直接利用することもできるが，より軽い同期や排他制御をめざして，ユーザレベルで動作する並列LWP（Light Weight Process）パッケージも用意されている[6]．このパッケージでは，まずsharedforkを使ってメモリを共有した複数のプロセスを生成し，その上でユーザレベルのコンテクストスイッチングを行っている．並列サポート言語処理系としては，TOP-1 Common LispとParallel Fortran処理系の研究が行われている．TOP-1 Common Lispは，Kyoto Common Lispをベースにfutureなどの並列プリミティブの追加と並列ガーベージコレクション等の拡張を行ったもので，実装には前述の並列LWPパッケージが用いられている[7),8)]．TOP-1上のParallel Fortranは，IBM Parallel Fortranのサブセットで，PARALLEL LOOP文やPARALLEL CASES文などが利用できる．実装は，プリプロセッサと並列サポートライブラリにより行われている[9),10)]．

3.3 ハードウェアとOSの評価

TOP-1プロジェクトの目標の1つは，並列ハードウェアおよびOS自身の構成法の評価にある．そのため，TOP-1 OS上には各種データのモニタ機能が用意されている．代表的なものとして，TOP-1 OSモニタ，TOP-1トレーサ，キャッシュ情報収集ツールなどがある．

TOP-1 OSモニタは主にOSのデバッグのために作られたもので，共有メモリの内容表示などを，OSの走行中にリアルタイムで行うことができる[5]．バスロックをかけてメモリ内容をフリーズしたり，メモリ内容の変更をすることも可能である．このモニタは，TOP-1に接続されたPS/55から共有メモリにアクセスすることで実現されている．さらに，共有メモリ上に用意されたPUごとの「仮想端末エリア」をブラウズすることで，各PUにおけるOSコードのトレースやプロセスの実行状況の表示も行える．

実際の並列アプリケーション実行時のモニタ機能としては，TOP-1トレーサとキャッシュ情報収集ツールが利用できる[11]．TOP-1トレーサは，アプリケーションのメモリアクセス状況をトレースするためのツールで，i386のSingle-Step Trap機能を使って実現されている．キャッシュ情報収集ツールは，TOP-1キャッシュに内蔵された統計情報収集機能を利用したもので，キャッシュのヒット率やデータ共有の状況などをリアルタイムで知ることができる．TOP-1プロジェクトでは，これらの機能を利用して，ハードウェア，OS，アプリケーションの評価が行われている[12),13)]．

4. OSF/1 TOP-1

TOP-1 OSは，マスタ・スレーブ型の構成をとったため，カーネル処理の頻度や比率が高いプログラムにおいては，KPUでの処理がボトルネックになり，システムの並列性が十分に生かされない場合があった．ユーザ・モードでの処理がほとんどの，数値計算のようなアプリケーションでは，並列部分について8プロセッサで7倍程度の性能向上が得られる[10]が，スケジューラやI/Oの比率が高い処理では，2倍程度の性能向上で頭打ちになるものもみられた[14]．また，構造が固定的でOS自身の構成法の研究には使用しにくい面

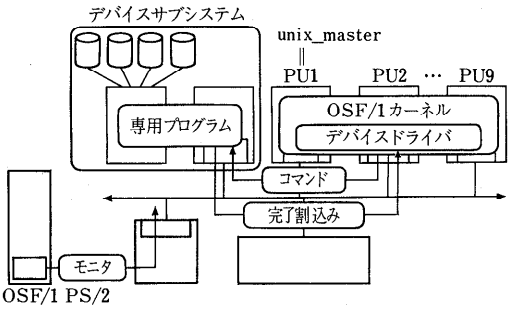


図-4

もあった．そのため，第二ステップとして，OSF/1 TOP-1 の開発を行った．これは，OSF/1 1.0 を TOP-1 に移植したものである．OSF/1 1.0 は，CMU で開発された Mach 2.5[17]をベースにした OS で，対称型のマルチプロセッシングをサポートしている．並列化は，カーネル内の各データ構造にロックを追加することで実現されており，ロックプリミティブとして，spin, read/write, mutex の 3 種類が使用されている[18]．

TOP-1 への移植にあたっては，これらの並列化用のプリミティブの実装と，機種依存のデバイス制御部などの開発を行った．図-4 に OSF/1 TOP-1 の構成を示す．OSF/1 TOP-1 では，PU 0 でデバイス制御のための専用プログラムを走らせ，「プログラマブルなデバイスサブシステム」と見なせるようにしている．これにより，残りの 9 台の PU からは同等にデバイスにアクセスできるようになり，OS 構成の自由度が大きくなっている．また，OSF/1 TOP-1 上にも **3.3** で述べたものと同様の評価ツールが実現されている．

TOP-1 プロジェクトにおいて，OSF/1 TOP-1 は主に並列 OS 自身の構成法の研究のために用いられた．主なものとしては，デバイスサブシステムを利用したデバイスアクセス手法の比較[15]や，ページ管理機構やキャッシュプロトコルの違いによる性能差の測定[16]などがあげられる．前者では，デバイスアクセスの並列化により 5〜20% の性能向上が得られることが示されている．また後者では，マルチユーザ環境には update 型の，シングルユーザ環境には invalidate 型のキャッシュ・プロトコルがより適していることなどが示されている．

5. おわりに

本稿では，並列計算機 TOP-1 について，OS を中心に解説した．一般に OS の研究には，「結果が見えにくい」「いじりにくい」という問題があるように思う☆．前者の問題を解決するには，OS カーネルだけでなくその上のサーバや言語，ミドルウェアも含めたトータルな研究を行っていく必要があるだろう．この点では，TOP-1 プロジェクトは一定の成果をあげられたと考えている．後者の問題点に関しては，TOP-1 プロジェクトでは根本的な解決は図られなかったように思うが，今後の方向としては，マイクロカーネルに代表される新しい OS 構成法を考えていく必要があると感じている．

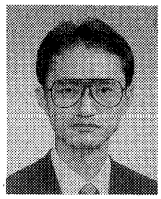
参考文献

1) 鈴木他：共有記憶型並列システムの実際，コロナ社並列処理シリーズ 16 (1993)．
2) 清水他：高性能マルチプロセッサ・ワークステーション TOP-1，並列処理シンポジウム JSPP' 89，pp. 155-162 (1989)．
3) Oba, N. et al.: TOP-1: A Snoop-Cache-Based Multiprocessor, Proc. 9th. Annual IEEE Int. Phoenix Conf. on Computers and Communications, pp. 101-108 (1990).
4) Shimizu, S. et al.: Design and Evaluation of Snoop-Cache-Based Multiprocessor, TOP-1, Proc. Int. Symp. on Shared Memory Multiprocessing, pp. 209-217 (1991).
5) 河内谷他：TOP-1 オペレーティング・システムの構造，情報処理学会研究報告 90-OS-48, pp. 25-34 (1990)．
6) 渦原，森山：マルチプロセッサシステムにおけるライトウエイトプロセス機構，日本ソフトウェア科学会第 6 回大会，pp. 353-356 (1989)．
7) Tanaka, T. and Uzuhara, S.: Multiprocessor Common Lisp on TOP-1, Proc. 2nd. IEEE Symp. on Parallel and Distributed Processing, pp. 617-622 (1990).
8) 渦原：共有メモリ型マルチプロセッサにおける並列ガーベージコレクション，情報処理学会研究報告 90-ARC-83 (SWoPP '90), pp. 205-210 (1990)．
9) 大澤：TOP-1 における流体問題解析，情報処理学会研究報告 90-ARC-83 (SWoPP '90), pp. 31-36 (1990)．
10) 大澤：TOP-1 における流体問題解析，電子情報通信学会論文誌 D-I, Vol. J75-D-I, No. 8, pp. 757-764 (1992)．

☆ 商用化という点で考えるとさらに，「従来 OS との互換性」や「はやらないとだめ」などの問題もある．

11) 大庭他：並列処理ワークステーション TOP-1 の性能評価環境，情報処理学会研究報告 90-ARC-83 (SWoPP '90), pp. 139-144 (1990).
12) Yamanouchi, N.: Performance Effects of Program Structures on a Snoop-Cached Multiprocessor System, Proc. InfoJapan '90, IPSJ, pp. 339-346 (1990).
13) Horiguchi, S. and Nakada, T.: Performance Evaluation of Parallel Fast Fourier Transform on a Multiprocessor Workstation, J. Parallel and Distributed Computing, Vol. 12, No. 2, pp. 158-163 (1991).
14) Shiratori, T. et al.: OSF/1 on the TOP-1 MP Workstation, IBM/TRL Tech. Rep. RT5026 (1992).
15) 河内谷他：MP UNIX におけるデバイスアクセス手法の比較，情報処理学会研究報告 92-OS-56 (SWoPP '92), pp. 89-96 (1992).
16) 山崎：マルチユーザ・オペレーティングシステムの基でのスヌープキャッシュの動作効率の評価，並列処理シンポジウム JSPP '93, pp. 371-378 (1993).
17) Accetta, M. et al.: Mach: A New Kernel Foundation for UNIX Development, Proc. USENIX 1986 Summer Conf., pp. 93-112 (1986).
18) OSF: Symmetrical Multiprocessing in the OSF/1 Operating System, OSF/1 White Paper, OSF-OSMP-WP13-1190-1 (1990).

(平成6年7月1日受付)



河内谷清久仁（正会員）

1963年生．1985年東京大学理学部情報科学科卒業．1987年同大学大学院理学系研究科情報科学専門課程修士課程修了．同年日本アイ・ビー・エム(株)入社．以来，同社東京基礎研究所にてオペレーティングシステムやマルチメディアシステムをはじめとするシステムソフトウェアの研究に従事．現在，同研究所副主任研究員．1994年本会全国大会奨励賞受賞．