

OS 動作検証用シミュレータ

7J-1

青木 久幸[†], 柴田 隆秀^{††}

[†]NEC C&C システム研究所, ^{††}日本電気ソフトウェア(株)

1 はじめに

計算機システムの大規模化や複雑化にともない、ハードウェアを占有しないと検査できないOSの開発や保守作業のコストが増大している。特に、スーパーコンピュータに代表される大型計算機では、マシン自体が高価なため、開発や保守のコストが一層高くなる傾向にある。一方、従来より、ハードウェア完成前から新しい計算機のソフトウェアを検査するための様々なシミュレータ [1] が提案されている。我々も、同じような考え方で、スーパーコンピュータ用OSの開発・保守コスト削減を目標に、OSの検査を行うためのシミュレータを開発した。このシミュレータは、ターゲットシステムとは全く違うアーキテクチャのシステム上で動作し、プロセッサ動作と入出力動作との並列動作のシミュレーションを実現した点や、様々な構成のターゲットシステムをシミュレーション可能にした点に特徴がある。

本稿では、このシミュレータの構成と、基本的な性能について報告する。

2 ターゲットシステム

シミュレーションの対象とするシステムはSX-3 [2] である。図1にSX-3の装置構成の概略を示す。APが演算プロセッサであり、内部にスカラユニットとベクトルユニットを持っている。CPは周辺装置制御用のプロセッサであり、APとは全く違うアーキテクチャのものが使われている。APは最大4台まで接続可能であり、主記憶は全プロセッサから共有されている。

OSのSUPER-UX [3] は、ほとんどAP上で動作し、CP上ではIOMと呼ばれる小さな常駐モジュールだけが動作する。ユーザプログラムも全てAPで動作する。

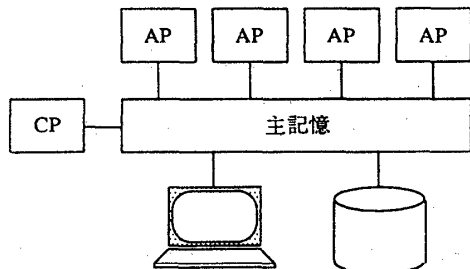


図1: SX-3の構成

3 シミュレータの動作環境

シミュレータはNEC製のEWS4800上で動作する。また、性能上は若干不利と考えられるが、シミュレーション作業の操作性やシミュレータ開発の容易性を考慮して、EWSのOS(EWS-UX [4])上で動作するアプリケーションとして実現した。特に、LANを経由してどこからでもシミュレーションができる点は重要である。

4 シミュレータの構成

図2にシミュレータの全体構成を示す。シミュレータ自体をマルチプロセスで構成することにより、プロセッサ動作と入出力動作の同時進行をシミュレーション可能にした。また、シミュレータ起動時に、構成定義ファイルの指定に沿って必要なシミュレータプロセスを起動することで、様々なシステム構成のシミュレーションを可能にした。シミュレータが別プロセスになる周辺装置の構成以外に、AP台数・ベクトルパイプライン数・メモリ容量なども変更できる。さらに、常に何もしないで正常終了のステータスを返すだけのダミー装置シミュレータを導入し、シミュレータの無い周辺装置を含む構成のシミュレーションもある程度可能にした。これは、該当装置ドライバ自体の検査は期待しない場合でも、検査条件のシステム構成と検査対象のOSを変更せずに、ドライバ以外の部分の検査を可能にするためである。

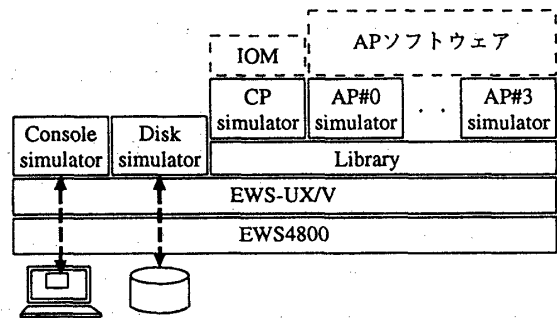


図2: シミュレータの構成

4.1 APシミュレータ

スカラ命令とベクトル命令の動作に加えて、OSの動作には必要不可欠なアドレス変換機構や割り込み・例外のハンドリングもシミュレーションしている。さらに、プロセッサ間通信などの制御命令もシミュレーションし、マルチプロセッサにも対応している。ただし、EWS自体はシングルプロセッサなので、時分割による

Simulator for debugging the Operating System
Hisayuki AOKI¹, Takahide SHIBATA²
¹NEC Corporation, ²NEC Software Co.Ltd

擬似的なマルチプロセッサである。このように、ソフトウェアに見える動作はほとんどシミュレーションしており、APソフトウェアのオブジェクトはそのまま動作可能である。なお、このシミュレータは性能情報入手が目的ではないので、実施するとかえって遅くなるキャッシュ機構はシミュレーションしていない。

4.2 CP シミュレータ

命令レベルのシミュレータは開発コストが高くなる点、CP上ではOS内部の小さなモジュールが動くのみである点を考慮し、IOM内のマシン依存関数群をシミュレータに置き換える方法を採用した。この部分では、シミュレータ全体の初期化の他に、IOMのブートローダ・プロセス管理・セマフォ管理・AP制御機能・周辺装置管理機能などをシミュレーションする。

4.3 主記憶のシミュレーション

ターゲットシステムの主記憶には、APソフトウェアのコードやデータ以外に、入出力のチャンネルプログラムや入出力データなども置かれる。従って、周辺装置のシミュレータプロセスからも参照可能でなければならず、EWS-UXの共用メモリを割り当てた。

4.4 入出力のシミュレーション

チャンネルプログラムレベルでシミュレーションしている。さらに、周辺装置の機能を含むOSの検査も可能にするために、ターゲットシステムの周辺装置の動作も、EWS中の同一性格の周辺装置を使って、シミュレーションしている。現時点では、最低限必要なコンソールとディスク装置のみをサポートしている。

チャンネルプログラムのシミュレーション

CPシミュレータ中で、チャンネルプログラム発行ルーチンが呼び出されると、該当する装置シミュレータへ入出力要求が通知される。この通知を受け取った装置シミュレータは、チャンネルプログラムを解読し、EWS-UXを呼び出して、該当する装置へ入出力を実施する。入出力が完了すると、CPシミュレータに入出力完了通知を送る。この通知を受けたCPシミュレータは、それまで実行していたプログラムを保留し、あらかじめ定義されたルーチンを動かす。キー入力などの周辺装置からのアテンションも周辺装置シミュレータが動きだす契機になり、入出力完了通知と同じ要領でCPシミュレータに伝えられる。

特に、入出力動作中のCPシミュレータと周辺装置シミュレータとの並行動作は重要である。なぜなら、実際のハードウェアも各装置は並列動作可能になっており、これを利用してシステムスループット改善を計るのが、OSのひとつの役目にもなっているからである。一方、EWS-UXの上では、入出力のためにOSを呼び出すと、完了するまでプロセスがブロックされてしまう。

そこで、プロセッサシミュレータとは別プロセスで入出力を行うことで、この並行動作を可能にしている。

コンソール

ひとつのX window(コンソールウィンドウ)をコンソールの画面に見立てている。即ち、コンソールへの表示要求はコンソールウィンドウへの表示に、コンソールウィンドウからのキー入力をコンソールからのキー入力としてハンドリングしている。

ディスク装置

EWSのディスク装置(raw device)を記憶領域として使用している。即ち、指定されたシリンダ番号・トラック番号から該当ディスク装置上のブロック番号を算出し、この領域への入出力を行う。

5 シミュレーション性能

このシミュレータは毎秒約14K個のAP命令を実行する。これはシミュレータマシンの約1/2700程度の速度に相当する。また、OS検査で頻繁に実施されるシステム立ち上げの時間は、最小構成のSUPER-UXで7~8分程度である。システム立ち上げ処理は入出力バウンドの傾向が強く、APシミュレーションの遅さはあまり目立たない。

6 おわりに

このシミュレータによって、低速ではあるが、SX-3全体がEWSの中に入っているような感覚で、OSの試し実行ができるようになった。システムのセットアップから検査まで、かなり実際のマシンに近い操作で作業ができる。現在の構成では、正確なタイミングが必要な機能やネットワーク機能などの検査は不可能である。しかし、プロセス管理やファイル管理などかなりの部分の試し実行が可能であり、OS検査のためのSX-3占有時間の削減は期待できる。今後は、シミュレーション範囲拡大や性能改善とともに、デバッグ機能や各種実行履歴収集機能などを追加し、より有効なツールに育てていく予定である。

最後に、本研究の機会を下さいましたNEC C&Cシステム研究所の山本所長、小池部長、中崎課長、また、有益な助言を下さいました皆様方に感謝致します。

参考文献

- [1] 武内他, 「OS検証用の新コンピュータ・アーキテクチャ・シミュレータ」, 情処第45回全大7P-5
- [2] 「SX-3シリーズシステム概説書」, May. 1989.
- [3] 浜地他, 「オペレーティングシステムSUPER-UX」, NEC技報, Vol.45, No.2, 1992.
- [4] 「EWS-UX/V(Rel 4.0)UNIX基本機能プログラミングリファレンス」, Jun. 1992.