2R-3

分散処理交換システムのシミュレーション環境の実現

田中 聡 久保田 稔
NTT 交換システム研究所

1 はじめに

電話サービスへの多様な要求に対応するため、交換システムには機能追加や処理能力の向上が容易に行えることが求められている。このため、交換機能をハードウェアも含めていくつかのモジュールに分割し、これらを組み合わせて構成する分散処理交換システム[1]についての検討を進めている。このような分散処理交換システムの実現に先だって、

(1) 複数のシステム構成の比較検討

(2) プロセッサ間通信など分散処理に必要な実現技術の確認

などを行う必要がある。

これらは従来机上で行われることが多く、手間がかかるとともに、定量的評価データをだすことが難しかった。そこで、分散処理交換システム構成の評価、機能確認を早期に効率良く行うために、UNIX 上にシミュレーション環境を作成した。本稿では、このシミュレーション環境について報告する。

2 分散処理交換システムモデル

分散処理交換システムのモデルとして、交換機の機能(信号方式、中継方式など)ごとにモジュール化し、それぞれのモジュール間を結合したシステムを想定している。具体的には、アナログ加入者線モジュール、Iインターフェース加入者線モジュール、中継線モジュール、共通線モジュール、ATM モジュールなどである。

ここでいうモジュールは、交換装置(トランク、加入者回路、通話路など)とそれらを制御するためのプロセッサ(群)から構成されるものであり、プロセッサ上では、交換装置を制御し交換処理を実現する呼処理プログラムが動作する。(以降、ソフトウェアモジュールと区別するために、**システムモジュール**と呼ぶ。)そして、それぞれのシステムモジュールはシステムモジュール間結合機構により結ばれていて、それを通じて、呼処理の情報や信号が通信される。モデルを図1に示す。

このような分散処理交換システムを擬似するためには、

(a) 分散性 … 複数システムモジュールの同時実行

(b) 多重性 … システムモジュール内での処理の多重性

(c) 実時間性 … 割り込みに基づく時間機能

(d) 入出力処理 … 交換装置との入出力処理

をシミュレーション環境の機能として実現する必要がある。

3 分散処理交換システムのシミュレーション環境

3.1 UNIX 上の CHILL 処理系

各システムモジュールでの呼処理プログラムの記述には、交換向きのプログラミング言語である CHILL[2] を用いる。CHILL は並行プロセスとプロセス間通信をサポートしている。

UNIX 上で動作する CHILL 処理系[3]が開発されており、この処理系では、UNIX の1プロセス内で複数の CHILL プロセスが動作し、CHILL プロセス間での通信を行うことができる。CHILL プロセスの実行を制御するモニタプログラム(以下 CHILL モニタと呼ぶ)はライブラリとして提供され、アプリケーションプログラムと結合して実行される。

この CHILL モニタは、

(1) CHILL プロセスの並行実行制御

(2) プロセス間通信制御

(3) UNIX からの割り込みによる時間制御

の機能を実現している。現在、実行制御としては、non-preemptive な制御をおこなっている。これは、C 言語で記述し、全体で約 4K である。

3.2 シミュレーション環境の実現

本シミュレーション環境では、分散処理の1システムモジュールまた交換装置を UNIX の1プロセスに対応させる。CHILL モニタと UNIX プロセス間通信機能をサポートすることによって、分散処理交換システムのシミュレーション環境を実現した。(図2)

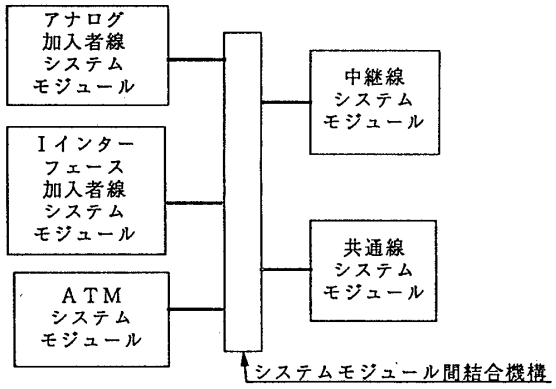

図1 分散処理交換システムモデル

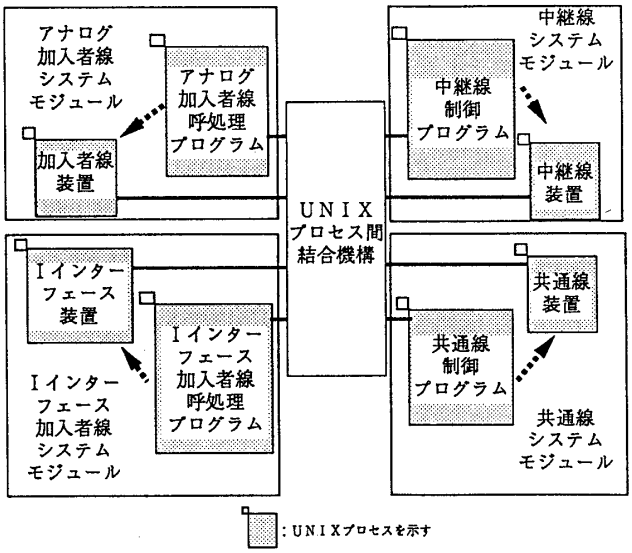

図2 分散処理交換システムシミュレーション環境

Development of a Simulation Environment for a Distributed Switching System
Satoshi TANAKA,Minoru KUBOTA
NTT Communication Switching Laboratories

分散処理交換システムシミュレーションに必要な機能の実現法を示す。

(a) UNIX の 1 プロセスを 1 システムモジュールに対応させる。UNIX プロセスは並行に処理されるので、システムモジュールの分散性 を擬似できる。 システムモジュール間通信は、UNIX のプロセス間通信機能である socket[4] を使って、データの転送を行う。

(b) CHILL モニタが UNIX プロセス内で複数の CHILL プロセスを並行処理 することによって、システムモジュール内での多重性を実現する。

(c) CHILL モニタの時間機能によって実時間性を実現する。

(d) システムモジュールが制御する交換装置を UNIX プロセスで擬似する。 ハードウェアを擬似する UNIX プロセスとの通信も socket によって行う。

CHILL モニタに加えて、socket 入出力機能部、CHILL プロセス入出力機能部を実現した。(以下カーネルと呼ぶ。図 3)

なお、本シミュレーション環境では、カーネルの機能のみを擬似していて、プロセッサの機能は擬似していない。現在、外部機器からの割り込みは CHILL モニタの時間制御機能を用いたルックイン方式で擬似している。

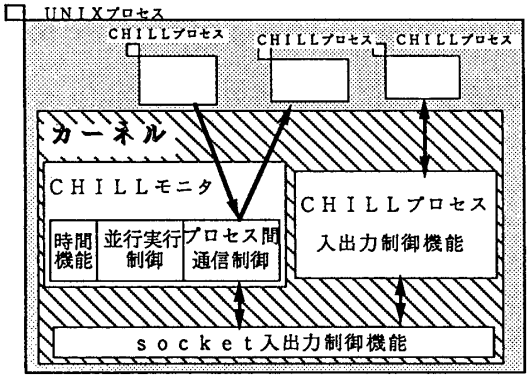

図 3 カーネルの実現

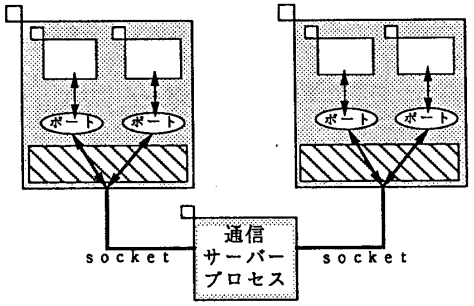

図 4 通信サーバープロセスによる通信機構

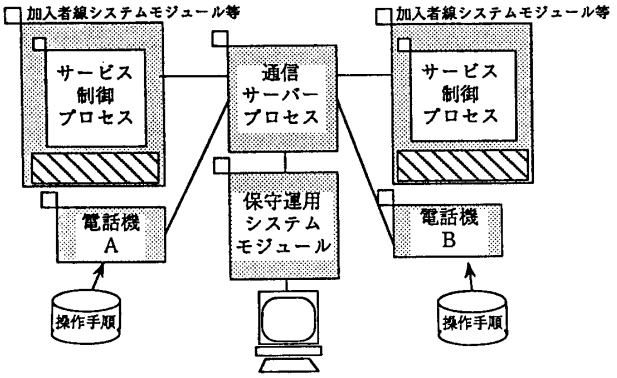

図 5 本シュミレーション環境の応用例

しかし、プロセッサシミュレータを用いて、カーネルの機能をその上で実現することによって、より実機に近い環境での擬似が可能である。現在、次バージョンとして、プロセッサシミュレータを用いた環境について検討を進めている。

3.3 socket によるシステムモジュール間通信機構

本シミュレーション環境の一つの目的は、複数の分散構成方式を比較し、評価することにある。そのため、システムモジュール間の結合形態を容易に変更できるような、柔軟な通信方式を実現する必要がある。そこで、本シミュレーション環境では、先の分散処理交換システムモデルで述べたシステムモジュール間結合機構にあたる機能を一つの UNIX プロセス (以後通信サーバプロセスと呼ぶ。図 4) として独立させ、これによってシステムモジュール間通信制御を行う方式をとっている。これにより、それぞれの UNIX プロセスは通信サーバプロセスとの間で socket をオープンする。UNIX プロセス間の通信は、この通信サーバプロセスを通じて行う。

さらに、システムモジュール間通信、入出力処理など UNIX プロセス間にまたがる通信は、システムモジュールに対応した UNIX プロセス上の一つの socket で多重化して行う方式をとっている。このため、CHILL プロセス間での通信を行うためには、同一の UNIX プロセス内での CHILL プロセスを識別する必要がある。そこで、通信用ポート (以下ポートと略す。) を導入し、次のような番号を導入する。

(1) システムモジュール番号 (UNIX プロセスの番号)
(2) ポート番号 (UNIX プロセス内でのポートの番号)

CHILL プロセスはポートをアロケーションし、ポートに対してデータの入出力を行う。そして、ポート間の対応をつけることによって、特定の CHILL プロセス間での通信が可能となる。また、交換装置をシミュレーションする UNIX プロセス内の特定の装置を指定するためにもポートを用いる。(図 4)

さらに、システムモジュール間をまたがった CHILL プロセス間でメッセージ通信を行う必要があるため、CHILL モニタのプロセス間通信機能部に、メッセージの送り先プロセスが同一システムモジュール内かどうかを判定する機能を付加した。これによって、同じシステムモジュール内でのメッセージであれば通常のメッセージ転送処理をおこない、そうでなければ、socket 入出力機能を使って、送り先プロセスのプロセス間通信機能部にメッセージを転送する。(図 3)

本方式によりシステムモジュール間の結合、プロセス間の結合が容易に変更できる。

4 シミュレーション環境の応用

本シミュレーション環境を用いて、細かな通話路制御の擬似が必要ないサービス制御処理の評価を行うことが可能である。(図 5) たとえば、電話機を一つの UNIX プロセスとして実現し、直接論理信号を通信することによって擬似する。また、電話機プロセスをプログラム可能とすることで、サービス方式の自動検証も可能である。

交換システムにおいて重要な機能の一つに保守・運用機能がある。本シミュレーション環境を用いることによって、保守・運用システムモジュールをプログラムすることが可能で、保守・運用機能の確認、評価に適用可能である。(図 5)

5 むすび

UNIX 上の CHILL 処理系を用いて、分散処理交換システムを擬似する手法を示した。本手法により、従来机上で行っていた評価が容易に行えるので、分散処理交換システム設計の工数削減が期待できる。

[参考文献]
[1] 荒木, 村上, 山田,"通信ノードシステム向き分散制御機能の一考察", 第 37 回情処全国大会,Sep,1988
[2]CCITT,"Recomendation Z.200",1989
[3] 佐藤, 大森,"マイクロコンピュータ用ソフトウェア開発に向けた CHILL 言語処理システムの実現", 情処研報,Vol.88,No.90,1988
[4]Sun Microsystems,"UNIX Interface Reference Manual"