

UNIX システムのネットワーク性能評価

1 P-8

本田 亮\*1 関戸 一紀\*1 友田 正憲\*1 山本 賢司\*2

\*1(株) 東芝 情報処理・機器技術研究所 \*2 東芝システムクリエイタ (株)

1 はじめに

近年、ビジネス分野でも計算機によるネットワーク環境が浸透しつつあり、性能評価において従来の集中型システムにはなかった構成要素をモデル化する必要が出て来た。これらを正しくモデル化し、利用形態に応じた性能を把握することは、システムの構築や拡張時のチューニングを可能とし、投資効果の高いシステムを実現するために必須である。

我々は、UNIXの代表的なネットワークサービスであるNFS(ネットワークファイルシステム)サーバ処理を詳細にモデル化し、性能を評価するシミュレータの開発を進めている。本手法は、ソースコードを解析し、トレース結果を元にモデル化を行なうため、より正確な性能が把握可能なシミュレーションプログラムを効率よく生成できるようにする。また、ネットワーク以外の処理のモデル化も可能である。

本発表では、NFS処理を例にネットワーク処理をモデル化する手法とその要素について述べ、モデル化の例を示す。

2 モデル化の手法

一般に、ネットワークの処理は大部分がシステムモードで動作し、その処理内容をユーザレベルから把握することが困難である。そこで、ソースコードの解析、モジュール単位の処理時間の測定、その結果を使ってのモデル粒度の決定、シミュレーションプログラムの生成ステップから成る手法を提案する。その過程を図1に示す。

- ソースコード解析  
モデル化の対象となるソースコードの制御構造を解析する。実際には関数や呼び出し関係を自動的に調べ、データベースに格納するツールを用意した。
- トレース  
ソースコード解析の結果をもとにソースコードにモジュール単位でプローブを挿入し処理時間を計測する。計測結果はデータベースに格納し、モデルの粒度を決定する尺度として用いるとともにシミュレーションプログラムを生成する際のパラメータとしても使用する。
- モデル化  
トレースの結果を元に各モジュールの粒度を決定す

る。処理時間が長かったり、処理時間の変動が大きいモジュールは、制御構造に即してモデル化する単位を細かくしていく必要がある。

- シミュレーションプログラムの作成  
ソースコードの制御構造とトレース結果をもとにモデルを決定し、シミュレータ記述言語を使ってシミュレーションプログラムを作成する。このプログラムは待ち行列モデルに、割り込み処理などを付加した複合モデルとして記述される。

- 負荷生成  
従来、通信トラフィックの研究分野では、負荷をバケット単位で捉えており、その到着間隔はポアソン分布に従ってモデル化されることが多かった。しかし、ファイル転送時は一度に多量のバケットが連続して発生する性質(バースト性)があるため、特にファイルサーバなどの性能評価では、ポアソン分布に従った負荷発生モデルはあまり適していない[1]。

このためできる限り現実の系に近い負荷モデルを用意する必要がある。そこで実際に運用している環境から収集したNFSコールの発生履歴を用いて、それと等価なRPCを疑似的に発生するツールを開発した。疑似的負荷はトレース部がモジュールの処理時間を計測する際やシミュレータへの入力として使用する。またRPCコールの発生間隔を一様に変化させることにより、実際の利用環境に合わせて負荷を調節することが可能である。

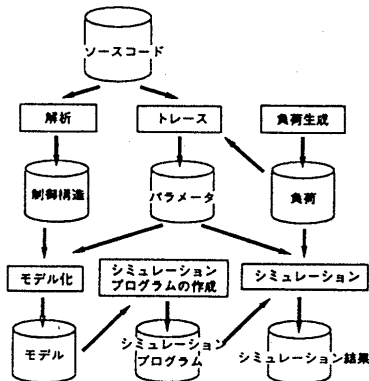


図1: シミュレーションプログラム生成の過程

3 ネットワーク処理のモデル化

OSやネットワークのプロトコル処理は挙動が複雑であり、単にCPUを消費するジョブとしてモデル化するだ

Performance Evaluation of Network Processing on UNIX.  
Makoto HONDA, Kazunori SEKIDO, Masanori TOMODA:  
Information Systems Engineering Lab. TOSHIBA Corp.  
Kenji YAMAMOTO: TOSHIBA System Creator Corp.

けでは不十分である。また、リソースの制限や割り込みなどによる変動が性能に著しく影響を与える部分については、詳細なモデル化が必要となる。

ここではストリーム [2] によってネットワーク機能が実装されている UNIX システムにおいて、NFS 処理をモデル化する際に着目すべき要素について述べる。

● OS の機構

OS の振舞いは割り込みハンドラによる割り込み処理と、プロセススケジューラによるプロセス切替えの動作が基本となっている。NFS 処理では予め代行プロセス (nfsd) がいくつか用意されており、このプロセス数だけ複数の NFS コールを同時に処理できる。また、ネットワーク処理を実装しているストリーム機構では、ストリームにいつ CPU が割り当てられるかが性能を大きく左右する。さらにページ処理やロック処理を伴うファイルシステム、割り込みレベルを考慮する必要がある。

● ネットワーク処理

今回モデル化するネットワーク処理は、各プロトコル処理がストリームモジュールとして階層化されて実装されている。プロトコル処理の内、データの 캡セル化、ヘッダの作成などはある程度一様な処理としてモデル化できるが、パケットの細分化 / 再構成は、他のモジュールやネットワークでのサービス数が変わるため、転送データサイズに従って詳細にモデル化しなければならない。

● ハードウェア

ハードウェアリソースのうち、NFS 性能に大きく影響するものとしてはディスク処理がある。特に書き込みは同期処理されているためディスクの書き込みが応答時間の大部分を占める。またサーバの処理性能が著しく高い場合にはネットワークのドライバでネックが発生することもある。CSMA/CD 方式ではパケットの衝突、バックオフ処理、ネットワーク媒体の性能をモデル化する必要がある。

4 モデルの粒度

モデルの粒度は、モジュール単位毎の平均処理時間、処理時間の変動、メモリの使用、割り込みレベルなどを尺

度として決定する必要がある。

表 1 に実際の NFS\_WRITE コールにおけるネットワーク処理にかかる CPU コストの例を示す。ディスクドライバ、ネットワークドライバ受信処理、UFS 処理、NFS 処理、IP 受信処理の順にコストがかかっていることがわかり、モデルの粒度を細かくしていく必要がある。

以上述べた手法をもとに、実際に構成したモデルの例を図 2 に示す。性能を大きく左右する OS の機構やハードウェアも含んでいる。現在この構成に従ってシミュレーションプログラムを構築中である。

表 1: ネットワーク処理の CPU コストとモデルの粒度 (NFS\_WRITE コールの場合)

処理単位	CPU 時間の内訳 (%)		モデルの粒度
	受信時	送信時	
ネットワークドライバ	20.8	2.3	細かい
IP 処理	8.3	3.7	細かい
UDP 処理	2.4	3.7	粗い
TIMOD 処理	1.6	1.3	粗い
TLI 処理	2.2	1.3	粗い
NFS 処理		12.5	細かい
UFS 処理		14.3	細かい
ディスクドライバ		29.1	細かい

(注) 割り込み処理は除く

5 おわりに

本発表では、UNIX におけるネットワーク処理の性能評価のためのモデル化について述べた。ソースコードの制御構造とトレース結果を用いたモデル化と、結果をデータベースで管理することにより、モデルの粒度の変更が容易になり、より実機の振舞いに近いシミュレーションプログラムを効率よく作成できるようになる。

今後は、マルチプロセッサシステム、NFS クライアントのモデル、NFS 以外のネットワーク処理のモデル等の検討を踏まえて、多様なシステムに対応できるモデル化手法へと発展させる予定である。

参考文献

[1] RICCARDO GUSELLA, A Measurement Study of Diskless Workstation Traffic on an Ethernet.  
 [2] UNIX SYSTEM V RELEASE 4 Programmer's Guid, STREAMS. Prentice Hall.

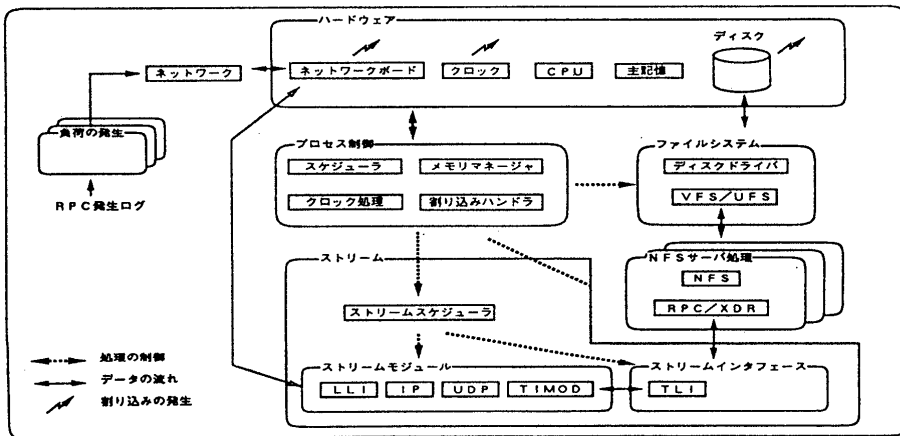


図 2: モデルの構成