

## ネットワークアーキテクチャの性能解析手法について

田中光一 柳生和男 関高明 樫尾次郎

(日立製作所システム開発研究所)

## 1. 性能が問題になる背景

近年、オンラインシステムはハードウェアのコストの低下にあいまって、オンラインシステムにのせる業務の多様化および、処理量の増加などにより次の傾向があらわれた。

- (1) ホスト計算機の分散 …… 計算機システムの統合または負荷分散
- (2) ホスト計算機の処理の分散 …… 通信管理の一部を前置プロセッサへ分散
- (3) 端末への処理の分散
- (4) 同一通信管理プログラムによる複数業務のサポート

これらの要求は各社でネットワーク・アーキテクチャとして一般化された。

このネットワーク・アーキテクチャにもとずいて構築される計算機ネットワークシステムは、従来の一般的なオンライン・システムに比較し次の点に特徴があり、性能評価は以前よりも難しくなっている。

## (1) Store and Forward

前置プロセッサ(FEP)でメッセージのバッファリング、経路制御、フロー制御などを行なう。このため、前置プロセッサ内にメッセージがたまり、これらの制御のためのパラメータの互い方、負荷の具合などが、メッセージの前置プロセッサ内の流れに影響する。

## (2) HDLCの採用

回線の効率的利用のため、両方向同時通信を行なえ、応答を待たずに情報を連続して先送りできる。

## (3) 回線の論理多重化

1つの回線を複数のオンライン業務で共用する。このため、同一回線内に異なる性質のデータの流れができ、それぞれに業務に適した論理リンクのフロー制御パラメータを手えねばならない。

## (4) ホスト計算機は仮想メモリ方式

実メモリネットワークによるページ例外の発生により処理時間の遅延を生ずる。

## 2. 性能評価手法と評価項目

計算機ネットワークシステムの性能評価手法は、その時期および評価に使用するツールの開発期間、評価ツールの使用目的とその範囲によりどの手法を採用するか決まる。表1にそれぞれの手法の比較を示した。

我々は性能評価手法として、人手をかかけずに、大規模構成の性能測定ができ、ホスト計算機内の通信管理プログラムおよびサポート・プログラムのデバグも可能な実システムの一部をシミュレートする方式を採用することとした。

表2の他の手法は次の理由で取り上げなかった。

## (1) 実機の場合

大規模構成の性能測定が人手、ハードウェアなどのため困難である。また、

表1 性能評価手法の比較

比較項目 方法	使用時期	作成手間	性能評価				
			ホスト プログラム	通信管理	前置 プロセス	ネットワーク ハードウェア	大規模 構成
机上	処理方式の 検討期。	○	△ 目安取得	△ 目安取得	△ 目安取得	△	△
解析モデル	処理方式の 検討期。 見積り期。	○ 評価対象が 多ければ手間	△	△	△	△	△
システムのモデル 化による シミュレーション		△	△	△	△	△	△
実システムの一部(端末 など)のシミュレート	システム開発 時。 見積り期	△	○	○	○ (方式による)	○	○ (方式による)
実機	システム開発 終期以降	—	○	○	○	○	○ (入出力 数量による)

使用できる時期が遅くなる。

(2) 机上、解析モデルの場合

処理方式の検討、概算見積りには有効である。一方、性能評価の結果は安全サイド寄りであり、実測またはシミュレーションによる裏付けが必要である。

(3) システムのモデル化によるシミュレーションの場合

構成要素の多いシステムの性能評価に用いる。モデル化がどこまで正しいのか実証するのが難しい。

### 3. 実システムの一部を擬似する方法による性能解析

シミュレータを性能解析に使用するためには次の機能を擬似する必要がある。

(1) トラフィック制御機能

実際に近いデータの流束、応答時間を求めるために必要である。このため回線速度、回線の伝送制御手順、論理リンクのフロー制御、ホスト-TEP間のチャネルのフロー制御などを擬似する必要がある。

(2) 多端末の擬似機能

多端末の場合のホスト通信管理プログラムの性能評価、前置プロセスのバッチ制御の性能評価などのために必要である。また、実機での大規模なネットワーク構成の測定は人手、多数の端末などのハードが必要、左の場所の確保などのため困難である。

表2 実システムの一部を擬似する方式の比較

方式 比較 項目		方式1 ホスト内でFEP, 端末などを擬 似	方式2 FEP内で回線 端末を擬似	方式3 ミニコンを使用し 端末をハード エミュレーション 作成	方式4(注) 中小計算機内 で複数端末を 擬似
シ ミュ レー ション 機 能	ホスト入出力 スーパバイザ	○ (必要)	- (省略可能)	- (省略可能)	- (省略可能)
	前置プロセッサ (FEP)	○ ( )	- ( )	- ( )	- ( )
	回線リンク	○ ( )	△ (一部改造)	○ (必要)	○ (必要)
	端末	○ ( )	○ ( )	○ ( )	○ ( )
	ネットワーク定数の 実システムの採用	○ シミュレータでネット ワーク定数と同じ使用	○	-	△
	処理結果、内 部状態の記録	○	△ 限られた情報しか 記録できない。	△ 全端末のみ情報	○
測定の手間	○	△	△	△	
大規模、高トラフィック の測定	○	△ (高トラフィックのみ)	× (1台では不可)	○	
ネットワーク性能要因 の測定	○ 擬似の精 度次第で必要あり	△ (一部可)	△ (一部可)	○	
測定時のシステムの処 理結果、内部状態の記録	○	△ (入出力装置負荷)	△ (端末のみ)	○	
テ レ バ グ 管 理	ホスト 前置プロセッサ の入出力インタ フェイス	○ (擬似的にシ ミュレーション)	○	○	○
	プロセッサ (NCP)	○	○	○	○
	プロセッサ (端末)	○	○	○	○
	プロセッサ (ホスト-ホスト)	○	△	△	△
	ソフト ウェア 正常処理	○	○	△ (一部可)	○
	異常処理	○ (どこまでシミュレーション 可能かによる)	△	△ (全体的に発生しない)	△ (全体的に発生しない)
開発工数	○ 小人数で可	△ 小人数でできる が、ハードウェアに よって時間がかかる	× 人手と日数がかかる	× 人手と日数がかかる	
使用上の制限	CPUネックにならない ような範囲で有効。	メモリが小さいと 大規模なものは不可	特になし	組込されている計 算機の能力まで	

(3) 測定結果の処理機能

性能測定した結果を解析するために必要である。

また、その他に、シミュレータの開発に当たっては、ホスト計算機内の通信管理プログラムおよびサポート・プログラムのデバッグもできるような考慮する。  
表2に実システムの一部を擬似する方式に付いて上述の事項およびテレバグに使用することも含めて、次の具体案についての比較を示した。表2と関連して図2には具体的な構成例を示した。

(1) 方式1

ホスト計算機内の通信管理プログラムが前置プロセッサを接続して行うチャネ  
ルに対し入出力要求を発行する部分でシミュレータを接続する。擬似の対象は

ホスト計算機  
内の入出カス  
ーハバインガ、  
チャンネル、前  
置プロセッサ、  
回線、端末で  
ある。この  
方式でネット  
ワーク・シス  
テムの性能評  
価を行なうに  
は上述の事項  
を考慮してシ  
ミュレータを  
開発する必要  
がある。

(2) 方式2  
前置プロセ  
ッサ内で多端  
末を擬似する。  
この方式の前  
提条件として

(a) 前置プロ  
セッサには補  
助記憶装置は  
接続できない。

(b) 前置プロ  
セッサのアド  
レスはホス  
ト計算機内で  
作成し、ロー  
ドする。

(3) 方式3  
ミニコン内  
で端末を擬似  
する。当該  
ネットワーク  
アーキテク  
チャ用の端末  
の試作に相当  
する。

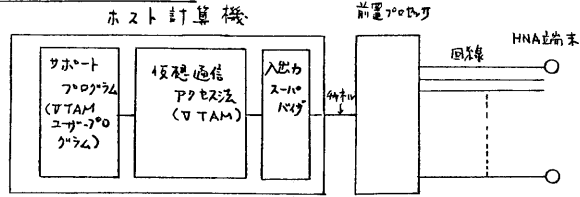
(4) 方式4

中心形計算機内で複数端末を擬似する。図1に示す構成とする。

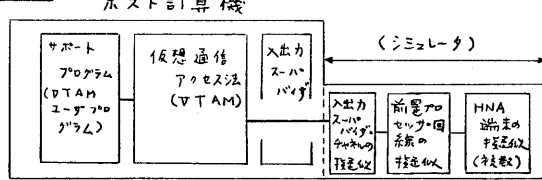
図1で点線で囲まれた部分をCとTとA(4チャンネル、2チャンネル、1チャンネル)に替える場合には

(a) 通信管理プログラムの入出力処理の一部を変更する必要がある

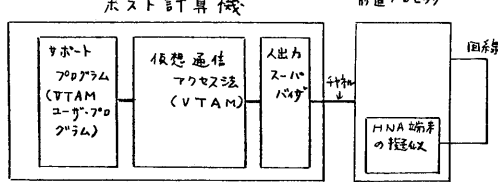
実システムの構成例



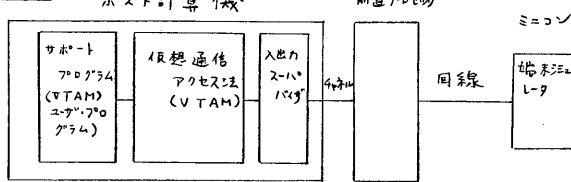
方式1



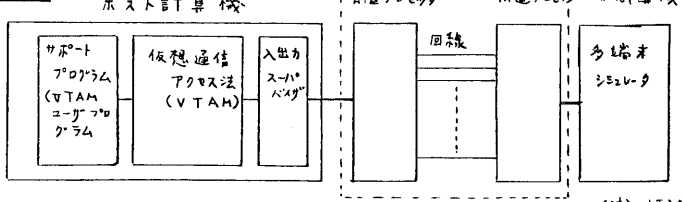
方式2



方式3



方式4



(注) CTA(4チャンネル、2チャンネル、1チャンネル)に置き換える。

図1 システムの一部を擬似する場合の構成例

(b) 中小形計算機内で前置プロセッサを擬似する必要がある。

上記の方式から、次の事項を重視して方式1を採用した。

- (1) 試作期間が短い。
- (2) 測定時の操作性が高い。
- (3) 大規模構成システムで連続した高トラフィックの負荷を容易に受ける。
- (4) ホスト計算機の通信管理プログラムおよびサポートプログラムのデバグに使用し易い。

方式1でシミュレータを開発する上で考慮すべき事項を次に述べる。

- (1) システム内の機能をどこまで擬似するか見定めぬこと。
  - (a) 伝達制御手順は回線の遅れという観点から擬似する。
  - (b) 端末の擬似は性能測定に影響のない範囲で機能を省略し、シミュレータ内のインタフェースを簡単にする。
  - (c) 擬似するシステム内のキューは全て擬似する方が、キュー処理のオーバーヘッドは増えるが、一般に擬似の方法は簡単に存在する。どこまでキューを擬似するかは、論理リンクのフロー制御および網内の論理リンクに存在している指令を消去する指令をどのプロトコルを考慮して決めることより。
- (2) 通信管理プログラムおよびサポート・プログラムのデバグの範囲を明確にする。性能測定のみ考慮する場合には、ハードウェア・エラーの発生、プロトコル・エラー検出機能は最小限擬似すればよいが、デバグを考慮するときにはどこまでシミュレータを使用してデバグするか考慮せねばならない。エラー発生時にシミュレータの内部状態を変え易いプログラム構造にしておく必要がある。

#### 4. HVAによるネットワーク・シミュレータの適用例

端末数が多く、高トラフィックの計算機ネットワークシステムの効率より性能測定およびその構成主要素であるホスト計算機内の通信管理プログラムVTA Mのヒートランまで含めたデバグのためには次の事項を満足する必要がある。

- (1) ネットワーク定義は、実システムと同じ定義がそのまま使用でき、定義を変更するだけで別構成の測定が簡単にできる。
- (2) 長時間連続して各種のラストデータを簡単に入力できる。
- (3) 端末、前置プロセッサ、回線などのハードウェアは不要。
- (4) 測定して得るシステムの処理結果および内部状態が詳細に分かる。
- (5) 測定中はホスト計算機の操作のみ行なえばよい。

ネットワーク・シミュレータは上記の事項を考慮して、第3章で述べた方式1を具体化したものである。

#### 4.1 ネットワーク・シミュレータの擬似機能

ネットワーク・シミュレータの擬似機能について述べる。

- (1) 入出力スーパーバイザ、チャネルの擬似

ネットワーク・シミュレータはVTA Mの前置プロセスに対する入出力インタフェースでVTA Mと接続されており、入出力サブパイプ、チャンネルおよび前置プロセスのチャンネル・インタフェースを擬似することにより、ネットワーク・シミュレータとVTA Mとのインタフェース整合を行なっている。

#### (2) 前置プロセスの擬似

(a) ホストからの前置プロセス宛の指令の処理と内部状態の管理および前置プロセスの内部状態の変化をホストへ報告する。前置プロセス宛の指令については順序性のチェック、フォーキャストチェックを行なっている。

(b) 論理図線のフロー制御を擬似する。

(c) 端末、他ホスト宛の指令のバツファリングを擬似する。

#### (3) 図線の擬似

図線のネットワーク定義と与えられたパラメータに従って伝送制御手順を擬似する。端末と前置プロセスとの間のデータの移動を制御する。伝送遅れはデータ長と図線速度も考慮して決められる。

#### (4) 多端末の擬似

MVAプロトコルに従う端末は1つの物理ユニットと複数個の論理ユニットからなる。物理ユニットに関しては、ホストからの物理ユニット宛指令の擬似のみを行ない、そのとき、物理ユニットの内部状態の管理および下位の論理ユニットの内部状態のリセットも行なう。論理ユニットに関しては、ホストからのセッション指令はネットワークシミュレータ内で擬似し、ホストのVTA Mコア・プログラムとの間の指令/応答の擬似はユーザ・コーディングで行なう。このときの内部状態の管理はネットワーク・シミュレータ内部で行なう。全二重モードの送受信を擬似する。多端末を擬似するために、物理ユニットおよび論理ユニット対応にそれぞれ、前者は1つ、後者は2つの擬似タスクとして動作する。

### 4.2 測定およびデバッグ用データの入力機能

測定およびデバッグ用データの入力機能としては次のものがある。

#### (1) 入出力エラー・パラメータ

VTA Mの前置プロセスに対する入出力要求の完了報告に指定したエラー状態をセットできる。

#### (2) 前置プロセス宛の指令およびMCPからのホストへの内部状態の報告のためのユーザ・コーディング

このユーザ・コーディングにより、左範囲のホスト宛のエラー状態を報告する応答の発生可能となる。図線のエラー発生を含む。

#### (3) 端末の論理ユニットのユーザ・コーディング

端末の論理ユニットとホストのVTA Mユーザ・プログラムとの間の任意の指令/応答を発生できる。データの発生間隔を変更することも可能である。1つのサブタスクでn個の論理ユニットの処理が可能である。

### 4.3 測定結果の処理機能

測定またはデバッグ中にシステム内で発生した指令/応答の発生時刻およびその内容、さらにチャンネルに対する入出力要求回数などのシミュレータの内部状態をロギングする。このロギングデータは使用目的に従って編集される。このデータは次のように使用できる。

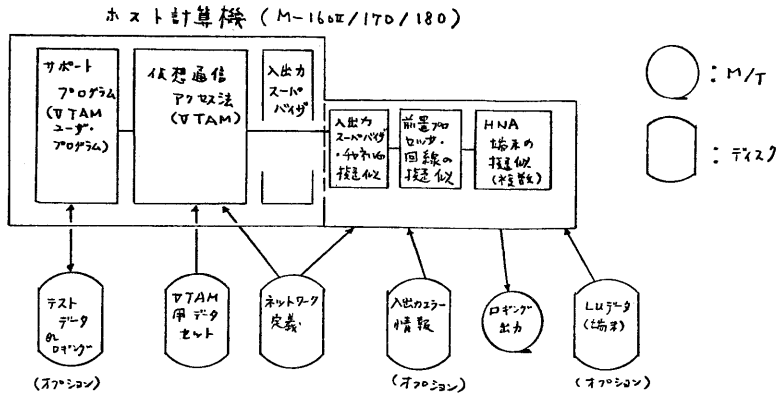


図2 ネットワーク シミュレータ 使用時のシステム構成例

- (1) ネットワークのフロー制御パラメータ，チャネルの制御パラメータなどの有効性および適値の範囲を求めらる。
- (2) システムの負荷量と応答時間の関係
- (3) ノード起動時間などの処理時間の測定
- (4) 指令に対する応答の正常性のチェック
- (5) 通信管理プログラム内での同時並列処理数

また他のモニタ・プログラムを同時に走行させることにより，VTAM内のバッファ使用状況，メモリ使用状況などとの関連を調べることにもできる。

#### 4.4 システム構成例

図2にネットワーク・シミュレータのシステム構成例を示した。使用できる計算機はMシリーズである。また仮想計算機でも使用できる。なお，処理能力については，初期の目標である，大規模構成で高トラフィックをかかての測定が行なえた。

#### 4.5 適用例

おもな適用例を以下に示す。

- (1) 通信管理プログラムのデバッグ
- (2) オンライン・サポート・プログラムのデバッグと性能テスト
- (3) 通信管理プログラムの性能テスト
  - (a) 実システムの構成で長時間にわたり高トラフィックの負荷をかける。
  - (b) 他のモニタ・プログラムを同時に動かすことにより各種の詳細な性能解析用情報が得られる。
- (4) ネットワーク・パラメータのテスト

ネットワーク・シミュレータで測定するネットワークの性能要因には次のものがある。

- (a) 問合せ応答，一斉送信などのデータの流入方の種類
- (b) 回線当りの端末数，端末の論理装置の数

(c) 回線のパラメータ, 論理リンクのフロー制御パラメータ

(d) チャネルパラメータ, NCPのバブル制御のパラメータ

上記のパラメータのうち机上解析で済むものは適値を与え, 他のものについてはそれぞれ値をかえて測定する。上記(a)(b)(c)については, 各組み合わせごとにネットワーク構成を作り, 一まとめのネットワーク構成とすれば, 一度に多数の場合の測定ができる。

#### 4.6 通用効果

ネットワーク・シミュレータを使用することにより, システムの性能測定およびデバッグの効率化を図ることができた。主な通用効果としては次のものがある。

(1) 安定した高トラヒックの負荷をかけることができる。

(2) 計算機使用時間の短縮。

(3) 前置プロセッサ, 端末などのハードウェアが不要であるため, これらと操作する人手がかかかず, 場所も不要である。

(4) 机上検討の裏付けができる。

#### 5. おわりに

HNAの性能評価に際し, 操作性のよさ, 大規模ネットワーク構成で安定した高トラヒックをかけられ, 性能測定にミテバッグにも使用できるという観点から多端末, 前置プロセッサなどの動作を擬似するネットワーク・シミュレータは有効である。

シミュレータ開発上の留意点は次のとおりである。

(1) 実際に近いデータの流しを発生させる。

(2) 多端末の動作を擬似する。

(3) 性能測定した結果を解析するための情報を出力する。

#### 文献

- (1) 田中,他 ; 多端末擬似試験システム(MTS-1), 通研実報, VOL 24, NO1, P252, 1975
- (2) 田中,他 ; VTA M用一端末シミュレータの試作, 情報処理学会18回全国大会(予稿), P713, 1972
- (3) W.R. Deniston ; S I P E : A TSS/360 Software measurement technique, PROC ACM, P229, 1968
- (4) R.E. Schwemm ; Experience gained in development and use of TSS, PROC, SJCC, P559, '72