

性能評価ツール“蟻塚”とその利用環境

小島昌一 米田清 田中利一 香木和仁
(株)東芝 システム・ソフトウェア技術推進部

1. はじめに

分散処理システムは、機能、性能、価格の面からの有利さを認められ、その利用範囲は急速に拡大している。そのため、システム設計の経験豊かな技術者が不足することは避けられない状況にある。それを補うために、体系的な支援機構を構築して、熟練した技術者でなくともシステム設計が出来る環境を用意する必要がある。

システムの仕様は、機能と性能の2つの面から定義される。機能は定性的に議論されるのに対して、性能は定量的に議論されることが多い。したがって、システムの計画や設計の早い段階において、応答時間、処理能力、稼働率などを予測し、評価する作業が必要になる。

本論では、分散処理システムの計画段階での性能評価用ツール“蟻塚”^[1]について述べる。“蟻塚”は、待行列網としてモデル化された分散処理システムに対して解析計算によって性能指標を計算する。さらに、その計算部分の前にモデル化を支援するエキスパート・システム、後ろに計算結果の解釈を支援するエキスパート・システムを用意し、待行列網の専門家でなくても利用できるようにしている。

2. 計画段階における性能評価

システムの性能評価の方法として、解析計算、シミュレーション、および測定がある。それぞれの方法は、それに必要な手間と、得られる精度の間に図1のような関係があると考えられる。

システムの計画段階では、まだ測定をする対象が存在しないので、測定は不可能である。そこで解析計算とシミュレーション及びその両者の混合の利用が考えられる。

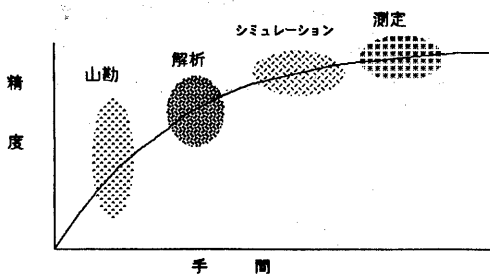


図1 各性能評価方法における手間と精度の関係

シミュレーションでは、原理的には、任意の対象に対して任意の精度まで、評価が可能である。しかし、その対価は、人的資源と計算機の使用時間の大量消費である。解析計算では、逆に、扱える対象範囲は限られ、精度もそれほど高くはない。しかし、モデルを作成し解くことが迅速にできる。

システムの計画段階では、もともと詳細なデータが揃っていることは少く、したがって、あまり精密に解析しても精度の高い評価結果は期待できない。また、種々のシステム代替案を評価し、それぞれを比較検討し、さらにそれらを修正し、繰り返し評価検討を行うことが普通である。このため、計画段階では、粗くても簡単迅速に性能評価ができる方法が望ましい。以上のことから、“蟻塚”では、解析計算による方法を採用している。

3. 性能評価の手順

“蟻塚”による性能評価は、システム計画作業の一部である。性能評価が必要になる段階では、すでに各種の視点から計画資料が作成され、以後の作業に必要なデータはデータベースに入力されている。したがって、性能評価に必要なデータは、そのデータベースから取り出し、対象システムのERA (Entity-Relationship-Attribute) モデルとして作成することができる。このとき、必要なデータが未だ揃っていない場合は、この時点で検討されなければならない。このように、システムの基本的な構成が既にできあがっているとき、次の手順に従って性能評価を行うことができる。

性能評価を行う手順は、次の4つの作業として考えることができる。

- (1) モデル化 計画中の対象システムを待行列網モデルの枠組に当てはめる。
- (2) パラメタ設定 対象システムの業務内容や機器の性能データから待行列網モデルのパラメタを設定する。
- (3) 解析計算 待行列網モデルの諸性能指標を算出する。
- (4) 解釈 性能指標の対象システムでの意味付けを行なう。

“蟻塚”では、図2に示すように、(1)と(2)の作業をモデル化エキスパート・システムが支援し、(3)の作

業を解析計算部が行い、(4)の作業を結果解釈エキスパート・システムが支援する。

この性能評価の結果をもとに、システムの計画を見直す。システムの評価は、性能だけでなく他の評価要素があり、総合的な評価の上で計画資料は修正され、必要があれば再度、性能評価が行われる。

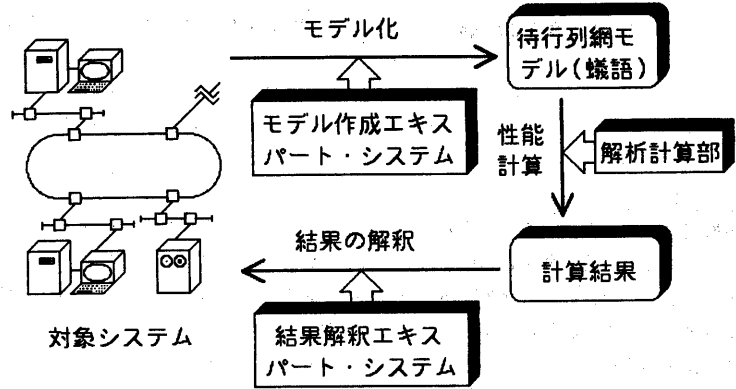


図2 “蟻塚”の全体構造

4. モデル化エキスパート・システム

“蟻塚”の解析計算を使って性能評価を行うには、対象システムの待行列網モデルを作る必要がある。そのためには、計画データの他に、4.2に分類して述べるような多くの知識が必要である。モデル化エキスパート・システム^[2]は、知識ベースに格納されたその知識を使い、不足している情報をユーザとの対話で得ながら待行列網モデルを構築する。

4.1 入力データと出力情報

モデル化エキスパート・システムは、あらかじめ用意された機器構成や業務処理などの計画データをERAモデルで表現したものを入力データとする。そして、待行列網モデルのERAモデル表現に変換し、それを待行列網モデルを記述する言語“蟻語”でモデルを記述し、出力する。この変換時に必要なデータ、例えば、注目している業務や解析計算を行うときの制御パラメタなどは、ユーザに質問し入力を促す。質問の方法としては、計画データから推論された候補をメニューとして表示し、1つを選択させるか、推論の結果として推奨されるデフォルト値を表示し、採用するか修正するかをユーザに判断させる。

4.2 知識ベース

次の5つに分類される知識が知識ベースに格納されている。

(1) 近似方法及び解析方法の知識

対象システムを待行列網の枠組に当てはめる際に、できるだけ簡単なモデルを使って、できるだけ高い精度の結果を得るための知識である。

(2) モデル作成手順の知識

モデルを作成していく手順には、ある程度の定石が

ある。定石に従って作業を行っていくと効率良く作業できる。経験的に得られた知識である。

(3) ハードウェア機器の知識

機器の処理速度や接続の可否などの知識である。

(4) 対象分野固有の知識

対象分野の業務内容やそこで使われるソフトウェアに関する知識である。

(5) “蟻語”の知識

待行列網モデルを“蟻語”で記述するために必要になる“蟻語”の文法についての知識である。

4.3 推論方式

モデル化エキスパート・システムでは、前向き推論を使ってモデル化を支援している。つまり、各ルールは、条件部と操作部からできている。与えられた機器構成と業務処理に対して、各ルールの条件部を評価して適用で

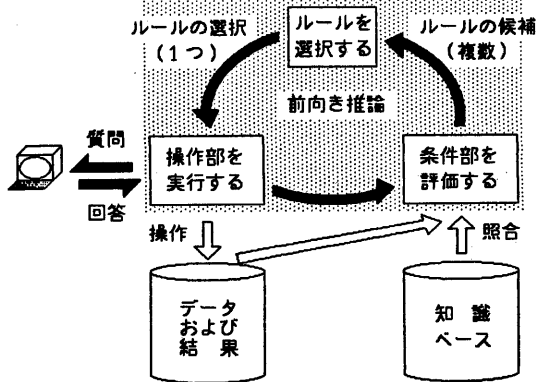


図3 モデル化エキスパート・システムの推論方式

きるルールを捜し、そのルールを適用する。複数のルールが適用可能な場合は、あらかじめ設定されている基準によって最も優先順位が高いものを適用する。ルールが適用されると、操作部の実行として、結果の一部または中間データが生成されたり、ユーザに対して質問が行われたりする。そのルールの操作部の実行が終ると、また各ルールの条件部を評価する。(図3)このように、前向き推論をモデルの完成まで繰返し行う。

5. 解析計算部

解析計算の理論的枠組には、伝統的な確率過程に基礎を置く待行列理論と、ここ数年の間に広く使われ始めた操作的解析 (operational analysis) に基礎を置くものがある。確率過程からは正規化定数法と呼ばれる算法が、また操作的解析法からは平均値解析法 (mean value analysis) [4] と呼ばれる算法が自然に導かれる。

平均値解析法は、モデルの設定が比較的単純で処理が速く、また柔軟性があり、適用範囲を拡張しやすいという特徴を持つ。したがって、この方法は、対象システムの性能を大局的に短時間に評価するという計画段階の性能評価の要求に一致する。そこで、“蟻塚”では平均値解析法と、それに基く近似を採用している。

平均値解析法では、その名の通り、扱うデータは全て平均値である。したがって、解析計算の結果として得られる指標も全て平均値である。

解析計算部は、分散処理システムには直接依存していないので、“蟻語”を入力とし5種類の性能指標を出力とする汎用の待行列網解析ツールとしても使用することができる。

5.1 入力モデル記述と計算結果

“蟻塚”の解析計算部の入力は、“蟻語”(図4)によって、次の5つの項目に分けて記述される。

(1) サービス・ステーション

サービス(処理)を行なうものを表す。各サービス・ステーションには待行列ができる。サービスを行う順番として次の4つが扱える。

①delay

任意の個数のジョブを同時に並行してサービスする。

②processor sharing

待行列に並んでいるジョブを微小なタイムスライスで時分割サービスする。

③last come first served with preemption

後入れ先出しでサービスする。

④first come first served

先入れ先出しでサービスする。

(2) ジョブ

①ジョブ

サービス・ステーションでサービスを受けるものがジョブである。ジョブは、サービス・ステーションを訪問してサービスを受けながら網の中を巡回している。ジョブには、開ジョブ(外部からシステムに到着し、システムの中でサービスを受けた後システム外に出て行く)と、閉ジョブ(システム内を常に一定の個数で巡回している)があり、混在が可能である。

②ジョブ・クラス

ジョブは、ジョブ・クラスによって特徴付けられる。即ち、ジョブ・クラスごとにサービス・ステーションでサービスを受ける時間やサービス・ステーションの訪問パターンが指定される。

③ジョブ・チェーン

ジョブが、ジョブ・クラスを交りながら待行列網の中を巡回する場合がある。そのような場合は、ジョブ・クラスを一括して考えなければならない。このジョブ・クラスの集りをジョブ・チェーンと呼び、それに対してジョブの量(開ジョブではシステムへの平均到着密度または平均到着間隔、閉ジョブではシステム内に存在する平均のジョブ数)を指定する。

```

name : [ 仙台支店 ];
goal : [ 仙台支店におけるバッチジョブ処理能力を知る。 ];
unit : [ s - 秒 ];

station:
    演算装置 is CIPS;
    記憶装置 is CIFF;
    端末 is CDIS;

chain:
    データ交換 pauses 1[s];
    会話 has 10[jobs];
    バッチ has 3[jobs];

class:
    データ交換; chains 入力 検索;
    会話;
    バッチ;

visit:
    データ交換 visits 演算装置 ratio 5;
    記憶装置 ratio 3;
    入力 visits 演算装置 ratio 4;
    記憶装置 ratio 1;
    端末 ratio 2;
    検索 visits 演算装置 ratio 0.6;
    記憶装置 ratio 0.1;
    バッチ visits 演算装置 ratio 5;
    記憶装置 ratio 2;

time:
    anyone spends 0.003 [ s ] at 演算装置;
    anyone spends 0.005 [ s ] at 記憶装置;
    入力 spends 5 [ s ] at 端末;
    検索 spends 8 [ s ] at 端末;

for バッチ 0.. 20 by 2;

response: 端末;

```

図4 “蟻語”記述の例

(3) 訪問比

各ジョブ・クラスのジョブが、どのようにサービス・ステーションを訪問するかを記述する。開ジョブの場合は、システムの外部からシステムに到着してからシステムの外部に行くまでに各サービス・ステーションを訪問する平均回数を記述し、閉ジョブの場合は、各サービス・ステーションを訪問する平均回数の比率を記述する。

(4) サービス時間

各ジョブ・クラスが、サービス・ステーションで受けるサービスに要する平均時間を記述する。

(5) 制御パラメタ

①ジョブの量及びサービス時間の変化の指定

同じシステムの構成でジョブの量（業務量に対応する）やサービス時間（機器の能力に対応する）を変化させて計算を行う場合に指定する。感度分析をしたり、いくつかのケースの比較検討する場合に利用される。

②応答時間の計算方法の指定

集中処理での性能評価が端末に視点を置いて見ることが多かったのに対して、分散処理の性能評価では、視点を移しながら考察することが多い。そのため、応答時間も種々の定義が考えられる。

5.2 計算結果

解析計算の結果として出力される指標は、以下の5種類である。これらは、各サービス・ステーションに対して各ジョブ・チェーンごとに得られる。

- (1) 平均スループット 単位時間当りに処理できるジョブの量である。
- (2) 平均滞在時間 待ち時間も含めて1回の処理に要する時間である。
- (3) 平均滞在数 処理中及び待っているジョブの量である。
- (4) 平均利用率 処理に利用されている割合である。
- (5) 平均応答時間 待ち時間も含めて、一連の処理に要する時間である。

計算結果の表示は、メニュー方式でガイドされ、表またはグラフの形式で行われる。

5.3 解法

“蟻塚”では、平均値解析法の中のSchweitzer-Zahorjan近似(SZ)^[5]ないしBard-Reiser近似と呼ばれるものが主に利用される。他に、厳密算法やlinearizerも利用できる。

ここで、以下のように記号を定義する。

- c : ジョブ・チェーン
- n_c : ジョブ・チェーンcに属するジョブの数
- M : ジョブ・チェーンの数
- s : サービス・ステーション
- q_{cs} : ジョブ・チェーンcのサービス・ステーションsにおける平均滞在数= $q_{cs}(n_1, \dots, n_M)$

平均値解析法の厳密算法では、計算の途中で、計算しようとする状態からジョブを1つ減らした状態の平均滞在数、即ち $q_{cs}(n_1, \dots, n_{j-1}, n_j-1, n_{j+1}, \dots, n_M)$ が必要になる。これを計算するためには、同様に、更にジョブの数を減さなければならない。結局 $q_{cs}(0, \dots, 0)$ までさかのぼることになる。よって、厳密算法では、 $q_{cs}(0, \dots, 0)=0$ から出発して順次ジョブ数を増やして $q_{cs}(n_1, \dots, n_M)$ まで計算を進めていく。この積立には、莫大な量の記憶領域を必要とするため、小さなモデルしか扱えず、実際上ほとんど使えないと言える。

SZは、次の近似によって、この積立を不要にしている。

$$q_{cs}(\dots, n_j-1, \dots) = \begin{cases} q_{cs}(\dots, n_j, \dots) & \dots j \neq c \\ q_{cs}(\dots, n_j, \dots) \cdot (1-1/n_j) & \dots j = c \end{cases}$$

つまり、 $j \neq c$ のときは、計算しようとしているジョブ・チェーン以外のジョブを減らしても q_{cs} には影響しな

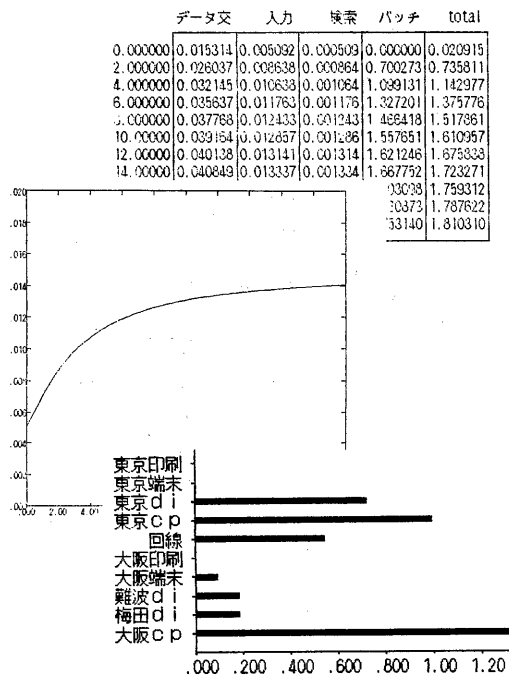


図5 解析計算部の結果の表示例

いことを示す。また、 $j=c$ のときは、計算しようとしているジョブ・チェーンのジョブを1つ減らすと、もとの q_{cs} に比例して影響が現われることを示している。

SZの変形としてlinearizer^[6]がある。これは、SZの近似式に修正項を付加している。(SZは、この修正項を0としたものである。)

$$q_{cs}(\dots, n_j - 1, \dots) = q_{cs}(\dots, n_j, \dots) + n_c \cdot d_{cjs}(n_1, \dots, n_M) \quad \dots j \neq c$$

$$q_{cs}(\dots, n_j, \dots) \cdot (1 - 1/n_j) + (n_c - 1) \cdot d_{cjs}(n_1, \dots, n_M) \quad \dots j = c$$

とする。ここで、 d_{cjs} は、

$$d_{cjs}(n_1, \dots, n_M) = q_{cs}(\dots, n_j - 1, \dots) / n_c - q_{cs}(\dots, n_j, \dots) / n_c$$

である。これは、ジョブ・チェーンのジョブ数が1つ減ったときの、そのサービス・ステーションにいるジョブの全体に対する割合の変化率である。linearizerは、SZ近似に比べ、高い精度の結果を得ることができる。しかし、記憶領域、計算時間を多く必要とする。ただし、厳密算法に比べればはるかに少なくすむ。

また、行列網を注目するジョブ・チェーンだけからなるものとし、他のジョブ・チェーンの影響はload concealmentと呼ばれるテクニックを使って補償する、Reiserの近似^[7]もある。しかし、手続きが、かなり面倒であるにもかかわらず、精度はSZに比べて必ずしも高くない。

SZの利点としては、

- ①ジョブ数は無制限に扱える
- ②必要な記憶容量が少なくてよい
- ③プログラムが簡単である
- ④ジョブ数が整数でなくてよい
- ⑤計算が速い

などが挙げられる。

6. 結果解釈エキスパート・システム

解析計算部で計算された結果としては、5. 2で述べた平均スループット、平均滞在時間等の指標が得られる。しかし、この結果を解釈して問題点を挙げ、その対策を立てるには、モデル化と同様に待行列網の知識が必要である。結果解釈エキスパート・システム^[8]は、その対象システムの診断を解析計算の結果から行う。このエキスパート・システムはエキスパート・システム構築ツールPROBE^[9]によって作成されている。

6. 1 入力データと出力情報

(1) 入力データ

入力データとして以下の3種類のデータを使用する。

①対象システムの計画データ

モデル化エキスパート・システムの入力として得られているERAモデルを用いる。

②待行列網モデル

モデル化エキスパート・システムによって作られた解析計算用モデルである。

③解析計算結果

解析計算部で計算された結果である。

(2) 出力情報

入力情報をもとに推論を行い、以下の情報を出力する。

①対象システムの診断の結果

対象システムでのボトルネックになっているサービス・ステーションの指摘を行う。また、そのボトルネックの主要原因になっているジョブを特定する。さらに、そのボトルネックを解消する方法の呈示を行う。

②解析モデルの診断結果

解析計算の計算誤差が大きくなっている可能性のある部分を指摘する。

③計算結果説明

①②の診断結果を顕著に表す性能指標の呈示やその見方の説明を行う。

6. 2 知識ベース

次の3つに分類される知識が知識ベースに格納されている。

(1) 平均値解析法の知識

計算誤差が出やすい条件などの平均値解析法に関する知識と計算結果の読み方に関する知識である。また、性能指標の値から物理的な意味付けを行う知識である。

(2) ハードウェア機器の知識

主に対策を推論する場合に用いる。機器の性能や接続の可否、問題となった機器に対して代替する機器の候補などの知識である。

(3) 対象分野固有の知識

モデル化エキスパート・システムの知識と同様に、対象分野の業務内容やそこで使われるソフトウェアの知識である。

6.3 推論方式

結果解釈エキスパート・システムは、エキスパート・システム構築ツールPROBEで作成されている。したがって、多段階の推論方式を採用している。モデル化エキスパート・システムの場合に、ルールが条件部と操作部から成っていたのとは異なり、条件部と結論部から成っている。それぞれは、共に事象またはその組合せで表され、条件部の事象が成立ったら、結論部の事象も成立つという形式をとる。

ボトルネックになっているサービス・ステーションの指摘やジョブの特定などの第一段推論は、後向き推論で行う。後向き推論は、結論となる可能性のあるものを仮定し、その仮定が正しいかどうかを条件部を調べて検証して行く方式である。条件部を調べるには、その中の各事象を結論部を持つルールを捜し、その条件部を検証する。これを繰り返すことにより推論が進められる。

また、ボトルネックの解消方法の呈示や説明の支援などの第二段推論では、第一段推論の結果を前提として前向き推論を行っている。これは、条件部の一部に今まで成立することが検証された事象を持つルールを捜し、残りの条件部を検証する。これを順次繰り返して行って成立する事象は何かを調べるという形式である。

7. おわりに

性能評価ツール“蟻塚”とその利用環境であるエキスパート・システムの概要について述べた。分散処理システムの計画段階の性能評価を目的としているので、おおまかな評価を簡単迅速に行えるように、待行列網でモデル化し、平均値解析法で解く方法を“蟻塚”では採用した。また、性能評価を行うシステム・エンジニアは計算機システムの専門家であっても待行列網モデルの専門家ではないことを考え、その利用が容易になるように、解析計算部の前後にエキスパート・システムを開発した。“蟻塚”の利用によって、性能評価が簡便に実施でき、分散処理システムの計画の労力と時間が軽減できるようになった。

参考文献

- [1] 春木他：分散処理システムの性能評価；東芝レビュー第40回，第5号，pp. 400-402，(1985)
- [2] 田中他：性能評価ツール“蟻塚”用モデル作成エキスパート・システムの開発；第30回情報処理学会全国大会，pp. 833-834，(1985)
- [3] 小島他：待行列網解析ツール“蟻塚”の開発（Ⅰ）（Ⅱ）；第30回情報処理学会全国大会，pp. 1099-1102，(1985)

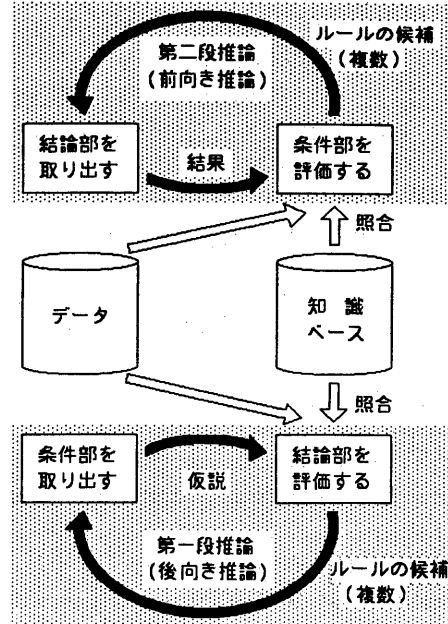


図6 結果解釈エキスパート・システムの推論方式

- [4] E. D. Lazowska, et al: Quantitative System Performance; Prentice-Hall, (1984)
- [5] S. C. Bruell and G. Balbo: Computational Algorithms for Closed Queueing Networks; Elsevier North Holland, (1980)
- [6] K. M. Chandy and D. Neuse: Linearizer: A Heuristic Algorithm for Queueing Network Models of Computing Systems; CACM, VOL-25, no. 2, pp. 126-134, (1982)
- [7] M. Reiser: Queueing Network Analysis of Computer Communication Networks with Window Flow Control; IEEE Trans. on Comm., COM-27, no. 8, pp. 1199-1209, (1979)
- [8] 田中他：エキスパート・システム構築ツールPROBE（Ⅱ）－性能評価計算結果の解釈への応用－；第32回情報処理学会全国大会，5L-2(1986)
- [9] 小島他：エキスパート・システム構築ツールPROBE（Ⅰ）－全体構成－；第32回情報処理学会全国大会，5L-1(1986)