

定性／定量推論による 同期型待ち行列ネットワークの ボトルネック診断と改善

志田圭介 (上智大学大学院 理工学研究科 機械工学専攻 博士前期課程)

伊藤 潔 (上智大学 理工学部 一般科学研究室 情報科学部門)

本位田真一 ((株)東芝 システム・ソフトウェア技術研究所)

早瀬健夫 (上智大学 理工学部 機械工学科)

同期型待ち行列ネットワークの性能パラメータの測定平均値に対して、定性推論と定量推論を相補的に組み合わせてチューニングする手法を述べる。この中で2つのエキスパートシステム、すなわち、ボトルネックサーバを同定しその要因を解明して、択一的に選択可能な定性的な改善プランを提示する“Bottleneck Diagnosis Expert System for open Synchronized queueing network: BDES-S”と、択一的に選択可能な定量的な改善プランを提示する“Bottleneck Improvement Expert System for open Synchronized queueing network: BIES-S”について述べる。BDES-Sの定性的な改善プランはパラメータ間の増減関係のみを定めるものであり、BIES-Sの定量改善プランはパラメータの具体的な増減値を定めるものである。

Qualitative / Quantitative Reasoning Based Diagnosis and Improvement
for Bottleneck of Synchronized Queueing Network

Keisuke Shida*, Kiyoshi Itoh**, Shinichi Honiden***, Takeo Hayase*,

*: Department of Mechanical Engineering,
Faculty of Science and Technology, Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan

** : Laboratory of Information Science, Natural Sciences Center,
Faculty of Science and Technology, Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan

***: Systems and Software Engineering Laboratory, Toshiba Corporation
Yanagi-cho 70, Saiwai-ku, Kawasaki 210, Japan

This paper describes Bottleneck Diagnosis Expert System and Bottleneck Improvement Expert System for Synchronized queueing network (BDES-S and BIES-S). BDES-S and BIES-S carry out parameter tuning for performance measurement data, with the combination of qualitative reasoning and quantitative reasoning, in order to diagnose and improve bottlenecks of synchronized queueing networks. BDES can produce several qualitative improvement plans for one bottleneck server. According to one of these qualitative plans, BIES can carry out automatic parameter tuning to remove the bottleneck by quantitative reasoning.

1 はじめに

本稿では、同期型待ち行列ネットワークの形態にモデル化できる並行システムを対象とし、その性能パラメータに対して、定性推論と定量推論を用いて、並行システムの性能に悪影響を及ぼすものを検出し要因を解明し改善プランを提示する手法について述べる。

並行システムは、多くの構成要素から成り、これらの構成要素が自律的に移動し、さらに必要に応じてある構成要素から他の構成要素へ同期信号やメッセージ等の情報を送るシステムである。このような並行システムの性能を分析するためのモデルとして、通常の特待ち行列ネットワーク([KLE75])に同期型のサーバを導入した同期型待ち行列ネットワークを用いる。

対象の並行システムには過大な負荷がかかっている、すなわち、そのモデルである同期型待ち行列ネットワークが非定常(過負荷)状態にある、あるいはその可能性をもつ状況を想定する。この対象システムにはボトルネックがある。通常の特待ち行列ネットワークのサーバの場合、稼働率あるいは待ち行列長が過大なサーバがボトルネックとなる。同期型待ち行列ネットワークでは、このようなボトルネックサーバに加えて、2節で示す通り、その2つの入力量の間にバランスがとれていない同期型サーバもボトルネックとなる。

ボトルネックサーバを効果的に解消するために、同期型待ち行列ネットワークの性能パラメータの測定平均値に対して、定性推論と定量推論を相補的に組み合わせてチューニングする手法を述べる。この中で2つのエキスパートシステム、すなわち、ボトルネックサーバを同定しその要因を解明して、択一的に選択可能な定性的な改善プランを提示する“Bottleneck Diagnosis Expert System for open Synchronized queueing network: BDES-S”と、択一的に選択可能な定量的な改善プランを提示する“Bottleneck Improvement Expert System for open Synchronized queueing network: BIES-S”について述べる。BDES-Sの定性的な改善プランはパラメータ間の増減関係のみを定めるものであり、BIES-Sの定量改善プランはパラメータの具体的な増減値を定めるものである。

定性推論([BOB85], [MIZ89], [NIS88, 89])は、経済システム([APT86])や電気回路([DEK85])などの定性的な挙動を表すモデルとして使われていることが多い。パラメータチューニングへの定性推論の主な適用例として、Rajagopalan, R.のターボジェットエンジンの対気速度の増加と絞弁の設定の関係を表す定性モデル([RAJ84])がある。

通常の特待ち行列ネットワークのボトルネック改善を行うBDESとBIESについて既に著者らの論文([SAW89a], [SAW89b], [ITO89, 90])で述べた。BDESは、定性推論を待ち行列ネットワークへ応用した最初のものであった。BIESは、BDESの定性推論を受けて定量推論を行う。

すなわち、BDESとBIESの特長は、パラメータチューニングに対して、単に定性推論を用いるばかりでなく、相補的にパラメータを定量推論できることである。本稿のBDES-SとBIES-Sは、同期型待ち行列ネットワークのパラメータチューニングに適用するものである。

BDES-SとBIES-Sによる診断と改善の流れを示す。ネットワークの形状は固定パラメータで他は可変パラメータ(診断後に変更可能)とする。すなわち、たとえばサービス率(μ)は各サーバの能力として測定前に与えられるが、チューニング後に可変である。

性能パラメータを測定した後、稼働率(ρ)や待ち行列長(q)が過大であるサーバ群、および入力量のバランスの悪い同期型サーバ群をBDES-Sが列挙する。評価者はこのうち1つのサーバを選択する。BDES-Sは、ネットワークの部分形状とその部分形状内の性能パラメータ値に関する定性式で表現できる知識と、この定性式の適用順序の制御に関する知識をもつ。BDES-Sが分析する性能パラメータは、ばらつきなどの値を用いない平均値であり測定上簡単なデータである。

BDES-Sは、1つのボトルネックサーバの複数の定性的な改善プランを列挙する。評価者はこのうち1つの定性的な改善プランを選択する。BIES-Sは、改善後のパラメータの値について定量推論を行なって複数の定量的な改善プランを提示する。評価者はこのうち1つの定量的な改善プランを選択する。

BIES-Sは、1つの定量改善プランに従って、ボトルネックサーバとこれに関係するサーバを改善する。この後、BDES-Sによるボトルネックサーバ群の列挙に戻るフィードバックを繰り返すと、ネットワーク内の全てのボトルネックサーバとこれに影響を受けていた全ての性能パラメータの自動チューニングが行なわれる。

このフィードバックにより、 unnecessaryな測定を繰り返すことが極力防止される。評価者は、すべてのボトルネックを改善した後の、新しいサービス率や分岐確率(r)などのパラメータを使うだけでよい。

2 定義

2.1 同期型待ち行列ネットワークの定義

本稿で対象とする待ち行列ネットワークは、処理されるエンティティが全て外部から到着し、それらは処理済みになると外部へ出る、いわゆるオープン型の待ち行列ネットワークである。また、エンティティは外部から複数のサーバに到着することが可能である。待ち行列ネットワークの内部では、それらは区別して扱わないため、種類に応じて途中の分岐確率やサービス率が異なることはない。すなわち、待ち行列ネットワークの内部では、エンティティは1種類と考える。また、この同期型待ち行列ネットワーク中には、通常のFCFSのサーバのほかに、2.3節で示すような3種類の同期型サーバが混在している。

2.2 性能パラメータの定義

ある1つのサーバに対して、 λ , μ , t , ρ , q , r を以下のように定義する。

λ : 単位時間当りに待ち行列に到着するエンティティの平均個数(到着率: arrival rate)

μ : 単位時間当りにサーバで処理できるエンティティの平均個数(組数)(サービス率: servicing rate)

t : 単位時間当りにサーバから出て来るエンティティの平均個数(スループット: throughput)

ρ : 1つのサーバが実際にエンティティを処理している時間の割合(稼働率: utilization rate)

q : 待ち行列内に待機しているエンティティの平均個数(待ち行列長: queue length)

r : エンティティがあるサーバを出た後、後続の複数のサーバのいずれに分岐するかを定める確率

(分岐確率: branching probability)

2.3 サーバの種類

本稿で用いる同期型待ち行列ネットワークは、次の4種類のサーバが複数個結合したものである。それぞれのサーバの定義を現実の例とともに述べる。

(1) ノーマル型サーバ (Table 1 項目1)

1つの入力路に対して、1つの出力路を持つサーバ。待ち行列に到着したエンティティは、FCFSの規則にしたがってサーバに入り、ある有限時間を経過した後サーバを出て行く。

このサーバの入力は、上流の複数のサーバの出力の合流であってよい。しかし、これらの複数のサーバから到着したエンティティは、ただ一つのFCFSの待ち行列に入る。また、このサーバの出力は、下流の複数のサーバに分岐してその入力となってよい。一般に、この分岐は確率的に行われる。

タスクの実行などをこのサーバによって表現することが可能である。このとき、このサーバのサービス時間(サービス率の逆数)は、そのタスクの実行時間であると考えられる。

以下の(2)~(4)は同期型サーバと総称される。

(2) スプリット(split)型サーバ (Table 1 項目2)

1つの入力路に対して、2つの出力路を持つ同期型サーバ。待ち行列に到着した1つのエンティティは、FCFSの規則にしたがってサーバに入り、自分の複製である1つの子エンティティを生成する。この複製には、ある有限時間の処理が必要である。この後、親エンティティと子エンティティは同時に、各々別の出力路から出て行く。すなわち、2つの出力量は常に等しい。

タスクの生成やメッセージの送信などをこのサーバによって表現することが可能である。

(3) マージ(merge)型サーバ (Table 1 項目3)

2つの入力路に対して、1つの出力路を持つ同期型サーバ。別の入力路から到着するエンティティは、それぞれ別の待ち行列を形成する。一方の待ち行列に到着したエンティティは、もう一方の待ち行列のエンティ

ティを探索する。もし、相手エンティティが複数個存在する場合には、FCFSの規則にしたがうものとする。相手エンティティが存在すると、2つのエンティティは同時にサーバに入り、1つに融合、すなわち、どちらか一方のエンティティが消滅する。この消滅には、ある有限時間の処理が必要である。この後、もう一方のエンティティは、サーバを出て行く。相手エンティティが存在しない場合には、相手エンティティが到着するまで、そのまま待機し続ける。

タスクの消滅やメッセージの受信などをこのサーバによって表現することが可能である。

(4) マッチ(match)型サーバ (Table 1 項目4)

2つの入力路に対して、2つの出力路を持つ同期型サーバ。別の入力路から到着するエンティティは、それぞれ別の待ち行列を形成する。一方の待ち行列に到着したエンティティは、もう一方の待ち行列の中のエンティティを探索する。もし、相手エンティティが複数個存在する場合には、FCFSの規則にしたがうものとする。相手エンティティが存在すると、2つのエンティティは歩調を揃えて同時にサーバに入り、ある有限時間を経過した後、2つのエンティティは同時に、各々別の出力路からサーバを出て行く。すなわち、2つの出力量は常に等しい。相手エンティティが存在しない場合には、相手の待ち行列にエンティティが到着するまで、そのまま待機し続ける。

タスク間の同期、通信などをこのサーバによって表現することが可能である。

2.4 各サーバの性能パラメータ

各々のサーバにおける各性能パラメータ間の関係をTable 1に示す。

2.5 ボトルネックの定義

待ち行列ネットワークの中で稼働率が1に極めて近いサーバにボトルネック(bottleneck)がある。そのボトルネックサーバの待ち行列長は無限に成長する恐れがある。このようなボトルネックサーバが存在する場合、待ち行列ネットワークは非定常(過負荷)状態になっている。

稼働率が1に近くなくても過大なサーバはボトルネックとなる可能性がある。サーバへの到着率やサーバのサービス率が大きくなればつきをもつ場合、稼働率が1に近くなくても待ち行列長が時には急激に増加する危険性がある。これを考慮して、フェールセーフの考え方で、たとえば、稼働率が0.7以上のサーバにボトルネックの可能性があるとして診断する。

稼働率が過大ではないが待ち行列長が過大であるサーバには、測定終了以降にエンティティが過大に到着してその稼働率が更に上昇する危険性がある。

マージ型およびマッチ型サーバについては、さらに次のような点に注意しなければならない。これらのサーバにおいて2つの到着率が異なる場合、サーバで実

際に処理される量は小さい方の到着率に等しくなる。もし仮に、平均の到着率の差が極僅かであったとしても、システムが稼働している限りこの差が積算されていく。すなわち到着率が大きい方のエンティティの形成する待ち行列は無限に成長する危険性がある。

過大な待ち行列長は後に徐々に解消されるかもしれない。しかし、ボトルネックの発生を事前に防止するというフェールセーフの考え方で、測定終了時に平均待ち行列長が1以上のサーバ、また、マージ型・マッチ型サーバで、2つの到着率が異なるサーバ全てについて、ボトルネックの可能性があると診断する。

以上のような基準でボトルネックの可能性があると診断されたサーバが存在する場合、待ち行列ネットワークは、非定常(過負荷)状態となる危険性がある。

3 待ち行列理論、およびペトリネット理論との関係

3.1 待ち行列理論との関係

一般のオープン型待ち行列ネットワークでエンティティが1種類の場合、定常状態ではトラフィック方程式が成り立つ(例えば文献[GEI89])。しかし、BDESの論文([SAW89a], [SAW89b], [ITO89, 90])で示したように、本稿で用いているノーマル型サーバだけで構成されている場合でも、ボトルネックが存在する時には、ネットワークは定常状態ではなくこの方程式は成立しない。さらに、同期型サーバを導入した場合には、ボトルネックが存在していない時でさえも、簡単な方程式で表すことは容易ではない。すなわち、単純な解析の式はボトルネックの改善プランの列挙に直接使うことができないため、エキスパートシステムの導入が有効である。

3.2 ペトリネット理論との関係

ペトリネット理論に新しく時間や確率変数の概念を導入した時間ペトリネットや確率ペトリネット([RAZ85])が考案された。そこでは、トランジションの発火時間間隔、トランジションの発火に要する時間やプレースの滞在時間などにある確率分布に従う有限時間の間隔を持たせるなど様々な形で時間の概念が導入されている。

本稿で導入した同期型待ち行列ネットワークは、一般の待ち行列理論で用いられる待ち行列ネットワークに対してペトリネット理論による同期の概念を導入したものであり、時間ペトリネットや確率ペトリネットに変換できる。同期型待ち行列ネットワークに言及したものととして Florin, G.らの論文がある([FL089])。彼の論文では、確率ペトリネットを同期型待ち行列ネットワークとみなしている。本稿では、一般の待ち行列ネットワークに2.3節の同期型サーバを導入したものを同期型待ち行列ネットワークと呼ぶ。

4 単一サーバの定性挙動式と定性改善式

エキスパートのヒューリスティクスとして稼働率0.7をボトルネックランドマーク(BL)とし、 $\rho \geq BL$ なら

ば、ボトルネックの可能性があるとする。

また、待ち行列長1を待ち行列長に関するボトルネックランドマーク(QBL)とし、 $q \geq QBL$ ならば、ボトルネックの可能性があるとする。

これを定性推論記法で表現する。以下、 $[\rho]$ と $[q]$ は、各々、BLとQBLを原点として評価した定性値である。

・ $[\rho] = +$: ボトルネックの可能性がある。

このときフェールセーフの考え方で

$d\rho = +$ と考える。

・ $[q] = +$: ボトルネックの可能性がある。

このとき同様に $dq = +$ と考える。

定性推論の考え方に基いて、個々のサーバの挙動を表す定性式を構成する。ボトルネックの可能性がない場合とある場合で用いる式の種類を変えることに特長がある。以下の式を個々のサーバの定性挙動式(qualitative behavior expression: QLBE)と呼ぶ。ボトルネックの可能性のある場合を特に定性改善式(qualitative bottleneck improvement expression: QL-BIE)と呼ぶ。

各々のサーバにおける定性挙動式、定性改善式を簡略化してTable 2に示す。 $[\rho] = -$ の項目はQLBE、 $+$ の項目はQL-BIEである。例えば、Table 2-1の1段目の項目は、ノーマル型サーバにおいて $[\rho] = -$ のときには、入の増減に対して ρ 、 t がともに追従することを示す。同様に、Table 2-3の3段目の項目は、マージ型サーバにおいて、 $[\rho] = -$ 、 $[q_A] = -$ 、 $[q_B] = +$ であるときには、 λ_B を減少すると q_B が減少するが、 ρ 、 t 、 q_A には変化がないことが示されている。

あるサーバの入を変化させるためには、それより上流のサーバからの t を変化させなければならない。あるサーバの t を増加すると、下流のサーバへの入の増加につながるので、下流のサーバがボトルネックとなる可能性があり、そうならないようにしなければならない。すなわち、ボトルネック改善のためには、全てのサーバに対して立式して、それらを組み合わせる解かなければならない。

Fig. 1に本稿で扱う例題待ち行列ネットワーク("PQN1"とよぶ)を示す。この性能パラメータはシミュレーションパッケージによって測定されたものである。

PQN1に対する定性挙動式・定性改善式の個数は、約350本である。ボトルネックの改善を行なう時には、これらの式以外にサーバ間の結合情報や分岐確率などの情報を与えなければならない。しかし、これらを全て組み合わせると定性挙動推論により解くと、状態数の爆発が起きて効率的ではない。

5 部分形状とその性能パラメータに基づくボトルネック改善のための定性改善式

全てのサーバの立式を組み合わせる解く方法ではなく、サーバ間の結合情報を取り込んだ部分形状を踏ま

えて、部分形状毎に内部の性能パラメータ間の増減関係を表した式で立式して全体の式の個数を減らす。

5.1 ボトルネック改善のための浅い知識

5.1.1 ボトルネック箇所の初期診断のための知識

先に述べた基準でボトルネックの可能性があると診断されたサーバが存在する場合、待ち行列ネットワークは、非定常（過負荷）状態となる危険性がある。

ボトルネックサーバが存在したり、その可能性がある場合、それらのサーバの能力（サービス率）の増加やそのサーバへの到着量の減少等の改善プランが必要となる。ボトルネック箇所を初期診断するためのヒューリスティックな知識を以下に挙げる。

- 1) 稼働率が一番高いサーバにボトルネックの可能性はある。
- 2) 稼働率が0.9以上のサーバにボトルネックがある。
- 3) 稼働率が0.7以上のサーバにボトルネックの可能性はある。
- 4) 稼働率が0.7より小さいが、待ち行列長が1.0以上のサーバにボトルネックの可能性はある。
- 5) マージ型・マッチ型サーバにおいて、2つの入力路への到着率が異なるサーバには、その稼働率や待ち行列長の値にかかわらずボトルネックの可能性はある。

5.1.2 ネットワーク変形のための知識

本稿で導入している同期型待ち行列ネットワークには、前述の通り4種類のサーバが混在している。4節末尾で述べたように全てのサーバの立式を組み合わせることで解く方法を採用すると、式の数が多くなり解くのに時間がかかってしまう。これに対して、サーバ間の結合情報を取り込んだ部分形状を踏まえて、部分形状毎の主要な性能パラメータ間の増減関係についての知識を表した式で立式すると式の個数は減少する。しかし、このとき4種類のサーバを組み合わせた構造全てを考えると式の個数は増加してしまうことになる。実際のシステムにおいては、ノーマル型サーバの占める割合が他のサーバに比べて大きいと考えられるので、数多くの種類の式を羅列しても効率的でない。

したがって、同期型待ち行列ネットワークを一般の待ち行列ネットワークに変形することを考え、その変形された待ち行列ネットワークに対してボトルネックの診断を行う。この変形は、診断前に行うのではなく、診断の過程で行う。診断の過程で同期型サーバを訪れた時点で4節に示した定性式の情報を保持したまま、このサーバをノーマル型サーバに変形する。はじめの同期型待ち行列ネットワークに対するボトルネック改善の方策と、変形によって生成された待ち行列ネットワークに対するボトルネック改善の方策は、等価でなければならない。

この変形を行いながら、ノーマル型サーバのみから成る一般の待ち行列ネットワークとしてボトルネック診断の知識を使う。

ネットワーク変形のための知識をTable 3に示す。例えば、上流への診断の過程でスプリット型サーバを訪れ、下流診断が必要となった場合（そのもの変化によってその下流の窓口群に何らかの悪影響を与えるかを調査する場合）に適用される知識がTable 3-1に示されている。スプリット型サーバの2つの下流窓口群をそれぞれ下流窓口群とする、2つのノーマル型サーバに変形し、それぞれ別のネットワークと見なす。また、上流への診断の過程でマージ型サーバを訪れた場合に適用される知識をTable 3-2に示す。ここで、このマージ型サーバの2つの入に差がある場合（Table 3-2上段に対応する）には、その入のどちらか一方が原因であると診断する。（ここでは入_aが原因になっている。）次に、このマージ型サーバs3を、原因とされた入_aがもとなっているサーバs1のみを上流に持つノーマル型サーバに変形し、上流への診断を継続することが示されている。この際、入_bがもとなっているサーバs2については、この入_bが外部に出力されているものと見なす。

5.1.3 ボトルネックの要因診断のための知識

Table 4には、部分形状とその内部のパラメータ値によるボトルネック改善のための14個の知識が示されている。これらは対象とする同期型待ち行列ネットワークがノーマル型サーバのみからなる待ち行列ネットワークに変形された後に適用されるものである。Table 4の第3項の場合の直接的な改善法は、直列型(tandem)のサーバs2のボトルネックを改善するために、 μ_1 と μ_2 のバランスを変えることである。しかし、その根本的な改善法は、接続する他のサーバのパラメータの大きさを変えることである。たとえば第3項の知識3に示すことは、1つ上流のサーバの μ_{1a} を減少させることである。また、第2項の知識2に示す通り、 μ を増加する時には、下流の過大な ρ を減少させる必要がある。

Table 4の知識と定性改善式の示すことは、待ち行列ネットワークの部分形状に着目すれば、ボトルネックを改善するために着目する性能パラメータの個数が、サーバのパラメータにすべて着目する場合よりかなり少なくなることである。例えば、第7項の知識7に示す通り、ループに関係する形状で、ループからのスループットを受け取るサーバs3がボトルネックとなる可能性がある場合、このボトルネックを改善するためには、ループ内の性能パラメータではなくループへの到着量のみを減少すればよい。すなわち、このボトルネックを改善するためには、

$$d\rho_3 = - \leftarrow dt_{1a} = -$$

という定性改善式のみを使用すればよい。

また、変形前にマージ型・マッチ型であるサーバについては、過大な待ち行列を解消するために、変形後のノーマル型サーバの入を増加しなければならないこ

ともある。あるサーバの入を増加するためには、それより上流のサーバからの入を増加させなければならない。すなわち、あるサーバの入の増加は、その上流のサーバの ρ の増加につながり、これらのサーバがボトルネックとなる可能性を生じる。したがってこのようなボトルネックの改善のためには、知識1～9とは別の新たな知識によって立式して、それらを組み合わせで解かなければならない。このような場合には、Table 4の第10項以降の知識10～14を適用する。

5.2 あるサーバにボトルネックの可能性があると診断された場合の知識の適用順序

あるサーバにボトルネックの可能性があると診断された場合に、Table 4の第2項の知識2でその直接的な要因と下流を調べる。(スプリット型・マッチ型サーバについては、これらが二つの下流の窓口群を持つため、同じ知識を二度適用する必要がある。)この知識2の適用後、そのサーバとその直上流のサーバとの接続関係によりTable 4内の第3～14項の知識3～14を適用する。上述のネットワーク変形のためのTable 3の知識は、上流への遡行の過程で直上流のサーバが同期型サーバとなった時点で適用する。

5.3 BDES-Sによる定性改善のプロセス

BDES-Sは、Table 3のネットワーク変形のための知識と、Table 4の定性改善式を用いて定性推論により改善プランを列挙するようにProlog言語を用いてインプリメントされた。ユーザはこのBDESを用いて、対象システムのボトルネック診断を対話的に行うことができる。

Fig. 1のPNQ1のボトルネック窓口s6(マージ型サーバ)に対する定性的な診断プロセス(定性挙動推論プロセス)をFig. 2に示す。Fig. 2の最上部のブロック1は、s6のボトルネックを改善したいという改善のゴール($d\rho_s = -$)である。

この診断プロセスの結果、改善のゴールを実現するための9つの択一的な定性改善プランが示されている。

6 ボトルネック改善のための性能パラメータの定量推論

ユーザが選択した1つのボトルネック定性的な改善プランに対して、BIESは改善後のパラメータの定量値を推論する。すなわち低いサービス率を定量的に増加させたり、高い到着率や高い分岐確率を定量的に減少させる。また、マージ型・マッチ型サーバの2つの到着率を定量的に等しくする。さらに、このような1つの窓口のパラメータチューニングの後、他の窓口の新しい到着率や新しいスループットを自動的に算出する。

BIES-Sが用いる定量改善式(Quantitative bottleneck improvement equation: QT-BIE)については、すでに著者らの論文([SAW89a], [ITO89, 90])で述べたものとはほぼ同様なので本稿では省略する。

Table 5では、Fig. 1のPNQ1のボトルネック窓口s6に

対する定性改善プランのうちplan1を選択したときの、BIES-Sによる定量改善プランおよび、改善前と改善後のパラメータを使って測定を行った際の ρ と q の比較を行った。 ρ_{16} と ρ_{27} の値が減少し、ボトルネックが解消された。s6, s38の各々の2つの q は、必ずしも小さくなっているわけではないが、その差はかなり軽減された。測定平均値のみを用いた改善が有効であることが示されている。

7 おわりに

本稿では、定性推論と定量推論を組み合わせて同期型待ち行列ネットワークのボトルネック改善方法を考察し、改善のための知識や定性/定量改善式、BDES-Sによる定性推論とBIES-Sによる定量推論の相補的な組合せ、実行例を示した。

BDES-Sは改善すべきパラメータ増減という形式で定性的な診断を行って複数の定性的な改善プランを提示する。個々の定性的な改善プランに対してBIES-Sは各パラメータの値の増減量を定量的に提示する。このため測定を行なう回数を減らして効率の向上が図れる。

参考文献

- [APT86]Apte, C. et al.: Using qualitative reasoning to understand financial arithmetic, Proc. AAAI'86, (1986).
- [BOB85]Bobrow, D.G., et al. ed.: Qualitative reasoning about physical systems, MIT Press, (1985).
- [DEK85]De Kleer, J.: How circuits work, in [BOB85], (1985).
- [FLO89]Gerard Florin and Stephane Natkin: Necessary and Sufficient Ergodicity Condition for Open Synchronized Queueing Networks, Proc. IEEE'89.
- [GEL89]Gelenbe, E. et al.: Analysis and Synthesis of Computer Systems, Academic Press, (1989).
- [ITO89]Itoh, K. et al.: Knowledge-Based Parameter Tuning for Queueing Network Type System - A New Application of Qualitative Reasoning, CAPE'89, (October, 1989).
- [ITO90]伊藤, 本位田, 沢村, 志田: 定性推論と定量推論を導入した待ち行列ネットワークのボトルネック診断と改善法, 人工知能学会誌, Vol. 5, No. 1, (1990).
- [KLE75]Kleinrock, L.: Queueing Systems, John Wiley & Sons, Inc., (1975).
- [MIZ89]溝口, 古川, 安西編: 定性推論, 共立出版, (1989).
- [NIS88]西田: 定性推論に関する最近の研究動向,

Table 1 Four types of servers and their performance parameters

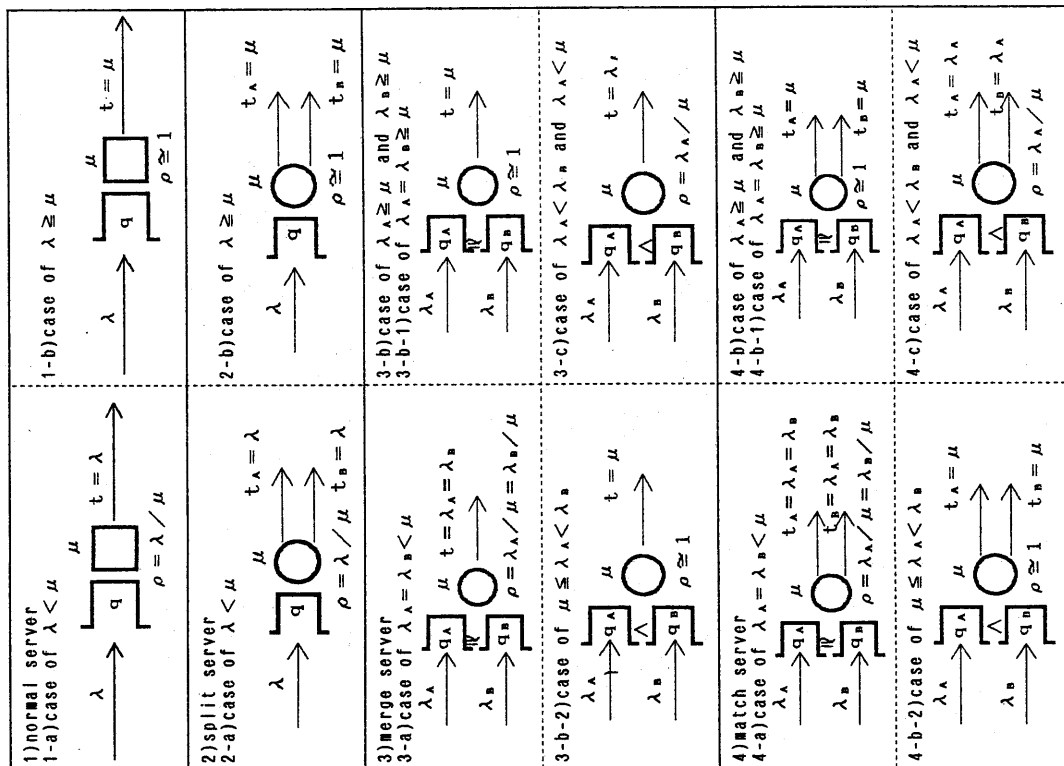


Table 2 QLBE and QL-BIE for each type of servers

Table 2-1 normal server

	$d\rho$	dt	$d\lambda$	$d\mu$
$[\rho] = -$	$\pm$	$\pm$	$\pm$	
	$\mp$	0		$\pm$
$[\rho] = +$	-	0	-	
	-	+		+

	$d\rho$	dq	$d\rho$	dq
$[\rho] = +$	-	-	-	
				-

Table 2-2 split server

omitted

Table 2-3 merge server

	$d\rho$	dt	dq_a	dq_b	$d\lambda_a$	$d\lambda_b$	$d\mu$
$[\rho] = -$	$\pm$	$\pm$	0	0	$\pm$	$\pm$	
$[q_a] = -$	0	0	+	0	+	0	
$[q_b] = -$	-	-	0	+	-	0	
	$\mp$	0	0	0			$\pm$
$[\rho] = +$	-	0	-	-	-	-	
$[q_a] = +$	-	+	-	-		+	
$[q_b] = +$	-	+	-	-	-	+	
*							
$[\rho] = -$	0	0	0	-		-	
$[q_a] = -$	+	+	0	0	+	+	
$[q_b] = -$							

* denotes the expression to be used in case of $q_a < q_b$

Table 2-4 match server

omitted

Table 3 Knowledge for network transformation
Table 3-1 Knowledge for downstreams (Knowledge m1)

before transformation	after transformation	Knowledge m1
		Apply to match or split servers. Duplicate match or split server. Consider it as normal server.

Table 3-2 Knowledge for upstreams (Knowledge m2, m3)

before transformation	after transformation	Knowledge m2
		Apply to merge or match server in case that either λ_A or λ_B is source of bottleneck.
QL-BIE $dq_A = - \leftarrow dt_1 = -$	QL-BIE $dq_A = - \leftarrow dt_1 = -$	
before transformation	after transformation	Knowledge m3
		Apply to merge or match server in case that both λ_A and λ_B are sources of bottlenecks.
QL-BIE $dq_A = - \leftarrow dt_1 = -$, $dq_A = - \leftarrow dt_2 = -$ $\leftarrow dt_2 = -$ need for matching	QL-BIE $dq_A = - \leftarrow dt_1 = -$ and $dq_A = - \leftarrow dt_2 = -$ $\leftarrow dt_2 = -$ need for matching	

Table 4 Substructure-based knowledge and QL-BIE adopted by BDES-S

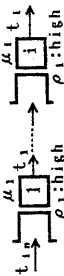
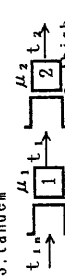
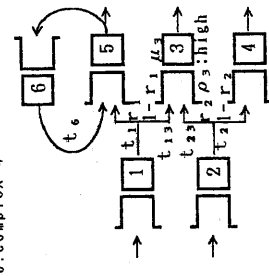
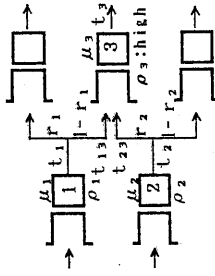
substructure	knowledge	QL-BIE
1 The servers whose " ρ "s $\cong 0.7$ or whose " q "s > 1.0 are may be bottleneck.		
2 	On improving a bottleneck server by increasing it's " μ 's", decrease " ρ 's" of it's downstream servers whose " ρ "s $\cong 0.7$ or whose " q 's" > 1.0 .	$d\rho_1 = -$ $\leftarrow d\mu_1 = -$ $d\rho_2 = -$ $\leftarrow d\rho_1 = -$
3:tandem 	Decrease t_{in} for decreasing ρ_2 .	$d\rho_2 = -$ $\leftarrow dt_{in} = -$
9:complex-4 	(1)Case of $t_2 * r_2 > t_1^*$ (1-r), decrease $t_{2,3}$ for decreasing ρ_3 . (2)Case of $t_2 * r_2 < t_1^*$ (1-r), decrease $t_{1,3}$ for decreasing ρ_3 . (3)Case of $t_2 * r_2 \leq t_1^*$ (1-r), decrease $t_{2,3}$ (or $t_{1,3}$) decreasing ρ_3 .	$d\rho_2 = +$ $\leftarrow dt_{2,3} = -$ $d\rho_2 = +$ $\leftarrow dt_{1,3} = -$ $d\rho_2 = +$ $\leftarrow dt_{2,3} = -$ $\leftarrow dt_{1,3} = -$
9:complex-5 	(1)Case of $t_2 * r_2 > t_1^*$ (1-r), decrease $t_{2,3}$ for decreasing ρ_3 . (2)Case of $t_2 * r_2 < t_1^*$ (1-r), decrease $t_{1,3}$ for decreasing ρ_3 . (3)Case of $t_2 * r_2 \leq t_1^*$ (1-r), decrease $t_{1,3}$ for decreasing ρ_3 .	$d\rho_2 = +$ $\leftarrow dt_{2,3} = -$ $d\rho_2 = +$ $\leftarrow dt_{1,3} = -$ $d\rho_2 = +$ $\leftarrow dt_{1,3} = -$ $\leftarrow dt_{2,3} = -$

Table 4 (cont.)

substructure	knowledge	QL-BIE
10: tandem-1 	Increase λ_1 for increasing λ_2 . (max: $\lambda_1 = \mu_1 * BL$)	$d\lambda_2 = +$ $\leftarrow d\lambda_1 = +$
10: tandem-2 	When $q_1 \geq QL$, increase μ_1 for increasing λ_2 .	$d\lambda_2 = +$ $\leftarrow d\mu_1 = +$
11: joint 	Increase λ_1 or λ_2 for increasing λ_3 . (max: $\lambda_1 = \mu_1 * BL$) (max: $\lambda_2 = \mu_2 * BL$)	$d\lambda_3 = +$ $\leftarrow d\lambda_1 = +$ or $d\lambda_3 = +$ $\leftarrow d\lambda_2 = +$
12: branch 	a) When r is modifiable, increase r for increasing λ_2 . b) When r isn't modifiable, increase λ_1 for increasing λ_2 . (max: $\lambda_1 = \mu_1 * BL$)	$d\lambda_2 = +$ $\leftarrow dr = +$ $d\lambda_2 = +$ $\leftarrow d\lambda_1 = +$
13: loop 	Increase t_{in} (input of a loop) for increasing λ_3 (output of loop).	$d\lambda_3 = +$ $\leftarrow dt_{in} = +$
14: external input 	a) When g_b is modifiable, increase g_b for increasing λ_2 . b) When g_b is not modifiable, increase $t_{1,2}$ for increasing λ_2 .	$d\lambda_2 = +$ $\leftarrow dg_b = +$ $d\lambda_2 = +$ $\leftarrow dt_{1,2} = +$

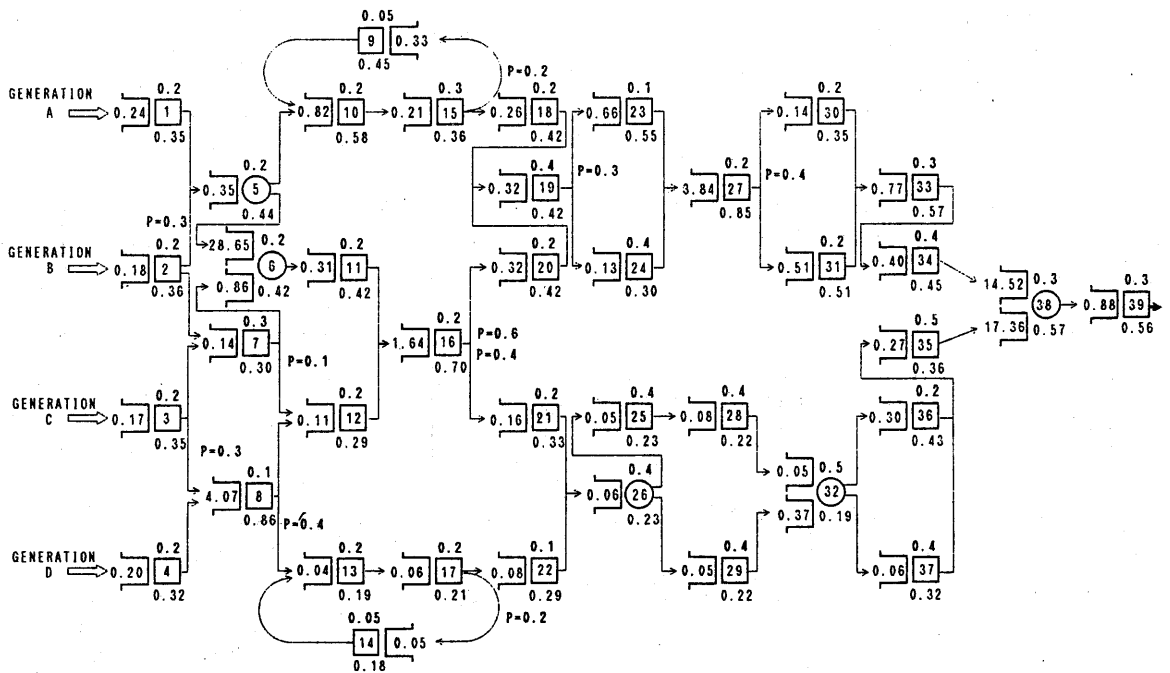


Fig.1 An example of Synchronized queueing network (PQN1)

情報処理, Vol.29, NO.9, No.11, (1988).

[NIS89]西田: 定性推論の基礎, 人工知能学会誌, Vol.4, NO.5, 522-527 (1988).

[PET84]J. L. Peterson, 市川, 小林(訳): ペトリネット入門, 共立出版(1984).

[RAJ84]Rajagopalan, R.: Qualitative modeling in the turbojet engine domain, Proc. AAAI'84, (1984).

[RAZ85]R. R. Razouk and C. V. Phelps: Performance Analysis Using Timed Petri Nets, in Protocol

Specification, Testing and Verification IV, (1985).

[SAW89a]沢村, 本位田, 伊藤: 定性推論を導入した待ち行列ネットワークのボトルネック診断, 情報処理学会知識工学と人工知能研究会, (January, 1989).

[SAW89b]沢村, 本位田, 志田, 伊藤: 知識工学的手法を用いた待ち行列ネットワークのボトルネック診断, 情報処理学会論文誌, Vol.30, No.8, 990-1002 (August, 1989).

Table 5 Comparison of ρ s and q s between two measurements

parameter	measurement-1	improved parameter	measurement-2
q_{6A} q_{6B}	28.65 0.86	g_a 0.069 $\rightarrow$ 0.064	0.02 18.42
q_{38A} q_{38B}	14.52 17.36	$r_{16,21}$ 0.40 $\rightarrow$ 0.39 $r_{16,20}$ 0.60 $\rightarrow$ 0.61	11.54 14.52
ρ_{16}	0.70	μ_{16} 0.20 $\rightarrow$ 0.265	0.62
ρ_{27}	0.85	μ_{27} 0.20 $\rightarrow$ 0.220	0.66

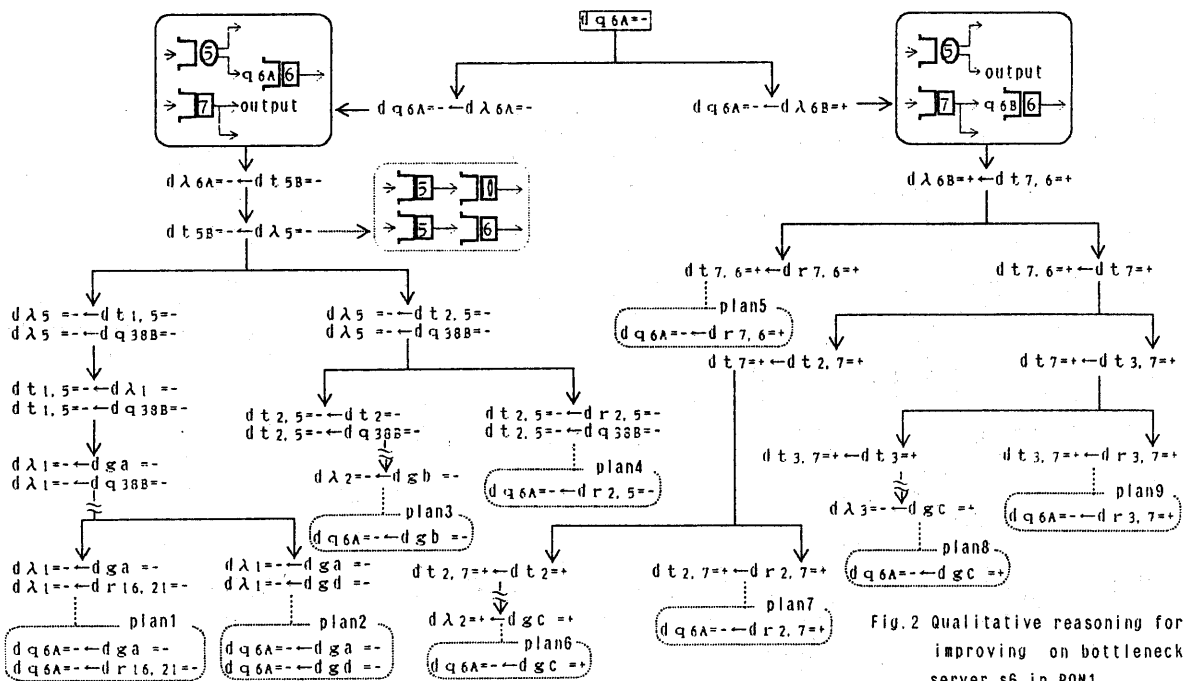


Fig. 2 Qualitative reasoning for improving on bottleneck server s6 in PQN1