

大規模計算機システムの性能評価手法とその適用例

村田正幸	稲井三重	稲井寛	横平徳美	宮原秀夫
大阪大学	JR西日本	神戸大学	大阪大学	大阪大学

大規模計算機システムの性能評価を行なうためには、待行列網システムにモデル化するのが有効である。そのために、まず、計算機システムのための待行列網モデルの構築手法について述べ、実際に稼働している大阪大学大型計算機センターの計算機システムに適用することによってモデルの妥当性をシミュレーションにより検証する。しかしながら、シミュレーション手法を用いる場合、対象システムが大規模になるとシミュレーション時間は莫大なものとなる。そこでモデルの簡略化を行ない、簡略化によって得られる時間短縮効果と簡略化により失われる結果の信頼性との関係について考察する。最後に、構築したモデルを用いて大阪大学大型計算機センターの計算機システムの性能評価を行なう。

Performance Evaluation Methods for Large Scale Computer Systems and Its Application

Masayuki Murata† Mie Inai‡ Hiroshi Inai* Tokumi Yokohira† Hideo Miyahara†

† Osaka University ‡ West Japan Railway Company * Kobe University

Department of Information and Computer Sciences
 Faculty of Engineering Science, Osaka University
 Machikaneyama-cho 1-1, Toyonaka, Osaka 560

A queueing network model is a powerful tool to evaluate computer systems. We describe the technique to construct a queueing network model for large scale computer systems in a systematic manner and validate it by applying it to the running system at Computation Center, Osaka University, using a simulation technique. However, simulation takes a lot of time to obtain reliable performance measures in the case where the target system becomes large. Then, we next propose several simplified models to reduce the number of queues for saving simulation time. We discuss the trade-off relationship between the accuracy and the cut effect in time among those models. Last, we evaluate the computer system at Computation Center, Osaka University by using proposed models.

1. はじめに

計算機システムの性能評価を行なうには、待ち行列網モデルを用いた方法が有効であり、過去においても様々な計算機システムの性能評価が待ち行列網モデルを用いて行われてきた[1]-[3]。ところが、これらは性能評価手法に重点を置いたもので、現実の稼働システムに適用し、モデルの妥当性まで検証したものは少ない。また、性能評価をシミュレーションにより行なう場合、対象となるモデルが大規模になると、結果を得るまでに多大なシミュレーション時間を要し、限られた時間内に結果を得ることが困難になる。それにもかかわらず、構築されたモデルについて、結果の妥当性を損なわないことを保証した上でシミュレーション時間を短縮する手法に関してはあまり研究がなされていない。さらに、現実には多くの計算機システムが会話型のジョブ(以降 TSSジョブ)とバッチジョブの双方を処理しているが、それら双方を考慮して性能評価を行なったものは少ない。

そこで、計算機システムのモデル化に関して情報収集からそのモデルの構築手順、検証までを系統立てて示し、それを実際に大阪大学大型計算機センターの計算機システムに適用し、モデルの妥当性を検証する。さらに、シミュレーションによる性能評価を効率的に行なうために、シミュレーション時間の短縮を目的としたモデルの簡略化を行い、簡略化によって得られる時間短縮効果と結果の信頼性について考察する。最後に、適用例として、大阪大学大型計算機センターの計算機システムにおいてバッチジョブ、TSSジョブが互いに及ぼしあう影響に焦点をあてて性能評価を行なう。

2. 計算機システムの性能評価モデルの構築と検証

計算機システムに関する様々な情報に基づいた待ち行列網モデルの構築とモデルの妥当性の検証について、大阪大学大型計算機センターの計算機システム(日本電気、ACOS1000)を例にとり述べる。

2.1. システム情報の収集

実際のシステムをモデル化するにあたり、そのシステムについての様々な情報を収集する必要があるが、その情報は以下の4つに大別できる。

(1) システム構成に関する情報

各装置の接続形態、台数に関する情報。例えば、ディスクの台数、複数CPUによる並列処理形態など。

(2) OSの制御に関する情報

OSが行なうジョブの制御に関する情報。例えば、バッチジョブの処理系内最大数、マルチプログラミングレベル(以降 MPLと略す)など。

(3) 装置に関する情報

各装置の仕様に関する情報。例えば、ディスクアクセス時間、チャンネルアクセス時間など。

(4) 統計情報

システムの利用状況に関する情報で、現実に稼働しているシステムから収集可能なもの。統計情報は更にジョブのワークロード特性を示す情報とシステムの特性を示す情報に分類できる。前者は待ち行列網モデルへの入力値となるもので、詳細な情報が得られれば、モデルもより現実のシステムを反映したものになるが、

実際には、これらの情報収集はシステムへのオーバーヘッドとなるので必要な入力値の一部しか得られていない。そこで、得られた値から前者の値を決定しなければならない。また、後者は待ち行列モデルにより得られる性能指標と一致すべきもので、例えば各装置の利用率がこれにあたる。本稿ではこの利用率の比較により、構築したモデル及び用いたワークロード特性の妥当性を検証する。

大阪大学大型計算機センターの計算機システムに関する情報を整理すると以下ようになる。

(1) システム構成に関する情報

- 主記憶装置(12438kwords): 1台
 - CPU: 2台
 - チャンネル装置: 2台
 - ディスク入出力用(12本)、プリンタ出力用(6本)、磁気テープ入出力用(4本)、端末入出力用(6本)、カードリーダ入出力用(2本)
 - ディスク: 100台
 - 端末: 300台
- 他に磁気テープ装置、プリンタ装置などがある。

(2) OSの制御に関する情報

- バッチジョブの処理系内最大数: 18
- 優先順位: バッチジョブは TSSジョブより優先順位が低い
- ルーティング: 入力されたジョブは CPUからチャンネルを介して周辺装置にアクセスし、再び CPUに戻るかまたは出力される
- マルチプログラミングレベル: 120

(3) 装置に関する情報

- ディスク平均サービス時間: 0.053秒

(4) 統計情報

例えば、次のような情報が得られる。(大阪大学大型計算機センターの場合)

- 測定日時: 昭和62年2月21日14時12分2秒-16時10分3秒
- 測定時間: 7081秒
- ディスクアクセス回数: 1223034回
- ディスク平均利用率: 9.15%
- CPU利用率(バッチジョブ): 39.87%
- CPU利用率(TSSジョブ): 29.65%
- ディスク入出力チャンネル平均アクセス回数: 50959回
- ディスク入出力用チャンネル平均利用率: 7.80%
- TSSジョブ平均サービスデマンド: 0.448秒
- 平均TSS人数: 73人
- TSSジョブ処理件数: 17519件
- バッチジョブ処理件数: 240件

なお、CPUの利用率には OSの処理も含まれている。また、TSS人数には端末で思考中の人も含んでいる。

2.2. 待ち行列網モデルの構築

2.2.1. 処理装置のモデル化(図1)

CPU: CPUは、EPU(Execution Processing Unit)と呼ばれる演算処理装置2台とSCU(System Control Unit)と呼ばれるジョブをどちらのEPUに送るかを決定する制御装置1台から成る。EPUはPS(Processor Sharing)で処理を行うと仮定する。また、SCUは非常に高速であるためそのサービス時間は0と仮定する。CPUは2台

並列に設置されていて、ジョブは2台の内のいずれか一方で処理を受ける。2台のCPUは1台の主記憶装置を共有しており、また各々のCPUから、全てのディスクにアクセスすることが出来る。なお、ここではEPUの利用率をCPU利用率としている。

チャンネル装置: 各装置用に予約された32個のサーバ(チャンネル)を持っており、各々のチャンネルは予約された装置全てに接続されている。すなわち、例えば、ジョブはディスク入出力用に予約されたあるチャンネルを通して全てのディスクにアクセスすることが出来る。ジョブがチャンネルを通して各装置にアクセスする際のアクセス方法には次の2通りが考えられる。

占有型: アクセスした装置での処理が終了するまで次のジョブがサーバに入れない。

通過型: アクセスした装置にジョブが移動してしまうとただちに次のジョブがサーバに入る。

占有型の場合、装置内に入ることで出来るジョブ数は予約されたサーバ数で制限される。そこで、モデルでは、チャンネルの前にチャンネル数と同じ容量を持つバッシブキュー(付録参照)を置く。通過型の場合、ジョブは処理の前後に一度ずつチャンネルを通過するが、サービス時間がアクセスする装置のそれに比べて非常に短いため省略する。ディスク、磁気テープ用のチャンネルが占有型にあたり、端末、プリンタ用のチャンネルが通過型にあたる。サーバではFCFS(First-Come First-Service)でサービスが行われるとする。また、チャンネル装置は2台並列に設置されている。

ディスク: 100台が並列に接続されている。サーバではFCFSでサービスが行われる。

端末: 端末において利用者は前に投入したジョブを待っているか、次のジョブを投入するために思考を行なっているかのいずれかの状態にある。従って、端末は利用者の思考時間をサービス時間としたIS(Infinite Server)でモデル化する。

他の磁気テープ装置、プリンタ装置などは、上述の装置に比べ利用回数が非常に少ないのでここでは省略する。

2.2.2. 処理系内のジョブの流れ(図1)

処理系内ジョブ数は、MPLで制限されている。更に、TSSジョブには、バッチジョブよりも高い優先順位が割り当てられる。処理系内ジョブ数がMPLに達したときには、TSSジョブはバッチジョブを一時的に処理系内から追い出して処理系内に入ることが出来る。従って、処理系の入口にFCFS-PR(FCFS Preemptive-Priority)バッシブキューPQ1を設定する。また、バッチジョブは、処理系内最大数を制限されているので、これを表現するFCFSバッシブキューPQ2をバッチジョブ入力キュー(Source Queue for Batch Jobs)とPQ1の間におく。

図1において、TSS、バッチ両ジョブの処理の流れは以下ようになる。TSSジョブは、端末から投入されPQ1に並ぶ。このとき、トークンが余っていれば、ト

ークンを持って処理系内に入る。トークンがなく、処理系内にバッチジョブが存在するときは、処理系内にあるバッチジョブを処理系内から追い出して、TSSジョブが処理系内に入る。追い出されたバッチジョブはPQ1の先頭に並ぶ。処理系内で処理を受けたTSSジョブは端末へ戻り、端末での思考時間を経たのち、再びPQ1に並ぶ。

バッチジョブは、バッチジョブ入力キューからPQ2を経てPQ1に並ぶ。PQ1にTSSジョブが存在する間は処理系内には入れない。PQ1にTSSジョブが存在せず、トークンが余っている場合に限り、処理系内に入ることが出来る。処理系内では、バッチジョブは、TSSジョブと同じ優先順位でサービスを受ける。しかしながら、PQ1にTSSジョブの到着があり、MPLに余裕がない場合、前述のように処理系内のバッチジョブが追い出され、PQ1の先頭に並ぶ。処理を終えたバッチジョブは処理系内を出る。

2.2.3. モデル・パラメータの決定

上記モデルにより実際に性能評価するためには、各パラメータを決定しなければならない。2.1からそのまま利用できる値として以下のものが挙げられる。但し、*、**、***の分布はそれぞれ指数分布、ポアソン分布、幾何分布に従うものと仮定する。

- バッチジョブのバッシブキューの容量: 18
 - マルチプログラミングレベル: 120
 - ディスク平均サービス時間: 0.053秒*
 - TSSジョブ平均サービスデマンド: 0.448秒
 - 平均TSS人数: 73人
- 収集情報から直接得られない値については集めたデータから算出する。
- バッチジョブ平均到着率: 0.0339件/秒**
(バッチジョブ処理件数)/(測定時間)
 - CPU平均サービス時間: 0.0161秒*
(測定時間)・(CPU利用率)/(CPUアクセス回数)
- CPUアクセス回数は統計情報では得られていないが、CPUを通ったジョブは必ずディスクにアクセスするので、ディスクアクセス回数で代用する。
- ディスク入出力チャンネルサービス時間: 0.0108秒*
(測定時間)・(チャンネル利用率)
/(チャンネルアクセス回数)
 - 平均思考時間: 30秒*
(TSS人数)・(測定時間)/(TSSジョブ処理数)
 - バッチジョブ平均サービスデマンド: 47.05秒
(バッチジョブCPU利用率)・(測定時間)
/(バッチジョブ処理件数)
 - TSSジョブの系内ループ平均回数: 27.8回***
(TSSジョブのサービスデマンド)
/(CPUサービス時間)
 - バッチジョブの系内ループ平均回数: 2922.5回***
(バッチジョブのサービスデマンド)
/(CPUサービス時間)
 - 各CPUへの分岐確率: 0.5
- 統計情報ではCPU利用率は2台分まとめたものしか得られていないので、均等に分岐すると仮定する。
- 各ディスクへの分岐確率: 0.01
 - 各ディスクのアクセス回数がほぼ等しいので1/(ディ

スク台数)とする。

- 各チャンネル装置への分岐確率: 0.5
- 各チャンネル装置のアクセス回数がほぼ等しいので 1/(チャンネル台数)とする。

2.3. モデルの検証

モデルの検証を行うために、各装置の利用率について、実際に稼働システムから得られている統計情報値とシミュレーション結果を比較する。統計情報は、システムが比較的混んでいる日(昭和62年2月21日、TSS 73人、バッチジョブ平均到着率0.034件/秒)、及び、比較的空いている日(昭和62年4月7日、同23人、0.015件/秒)の2日分のデータを採用した。但し、バッチジョブの処理系内最大数、MPL、各装置のサービス時間は2日とも同じとする。尚、本論文を通してシミュレーションは RESQ ([4],[6]-[8]) でスペクトラル法を用い、信頼水準90%が TSSジョブ平均応答時間の10%以内に収まるように実行した。表1の比較結果より、構築した性能評価モデルはほぼ妥当なものであるといえる。

3. 待ち行列網モデルの簡略化とその妥当性

待ち行列網モデルを用いて、シミュレーションによりシステムの性能評価を行う場合、そのシミュレーションに要する時間が問題になる。本章ではシミュレーション時間を短縮する手法とそれによって得られる結果の信頼性について述べる。

シミュレーション時間を短縮する方法として、モデル内の待ち行列の数を減らすことが考えられる。そのためには、いくつかの待ち行列をまとめて、同様の振舞いをする1つの待ち行列に置き換えることが必要になる。モデルが BCMP型[5]の場合には、等価流量法[1]により置き換えが可能であるが、そうでない場合、例えば、図1のようにバッチキューがあるような場合は、モデルの置き換えは近似モデルを導入することになる。従って、モデルの簡略化により得られる性能評価指標を元のモデルのものと比較してその妥当性を検証する必要がある。ここでは、2.2で構築した大阪大学大型計算機センターのモデル(以降モデル0 (Model0)と呼ぶ)の簡略モデルとして以下の2つを考え、その精度を検討する。

モデル1 (Model1): ディスクを簡略化したモデル

モデル0において、ディスクのモデルは1サーバFCFS待ち行列が100個並列に接続されたもので表現されている。これらを等価流量法を用いて10個の待ち行列が並列結合されたものに置き換える。すなわち10個の並列結合された1サーバFCFS待ち行列を1個の Composite Queue [1]で置き換える。簡略化を行なった部分のみにおいて平均応答時間の比較を行なった結果を図2に示す。簡略化自体は妥当なものであるといえる。

モデル2 (Model2): CPUを簡略化したモデル

モデル0においてCPUのモデルは2サーバPS待ち行列が2台並列に接続されたもので実現されている。これらを処理速度が4倍の1サーバPS待ち行列で置き換える。簡略化を行なった部分のみにおいて平均応答時間の比較を行なった結果を図3に比で示す。モデル0に

比べてかなりの誤差がある。

これらの各モデルについて、シミュレーション実行時間を比べると、モデル1ではモデル0の場合の約20%とかなり短縮されているが、モデル2ではモデル0の場合の95%程度にしかなり短縮されなかった。これは削減した待ち行列数の差に原因があると思われる。

各モデルの TSS人数に対する TSSジョブ平均応答時間を図4に示す。図4より、モデル1はモデル0とほぼ同様の結果が得られるが、モデル2の結果はモデル0のそれと比べてかなり誤差がある。そこで、モデル0の場合における各装置の利用率を見ると(図5)、CPUは利用率が非常に高く、ボトルネックとなっているのに比べ、ディスク、チャンネル利用率は10%程度に収まっていることがわかる。すなわち、ボトルネックとなっている待ち行列を簡略化した場合、大きな誤差が生じると考えられる。このことを確認するために、ディスクがボトルネックとなるようパラメータを以下のように変更し、各モデルについて TSSジョブ平均応答時間の比較を行う。

- ・ディスク平均サービス時間: 0.848秒
- ・CPUサービス時間: 0.00183秒
- ・マルチプログラミングレベル: 400
- ・チャンネルサーバ数: 400
- ・バッチジョブ平均到着率: 0.001件/秒

他のパラメータに関しては、2.2で挙げたものをそのまま用いる。図6にモデル0における各装置の利用率を示す。図6から分かるようにディスクがボトルネックとなっている。このときの各モデルの TSS人数に対する TSSジョブ平均応答時間を図7に示す。モデル2はモデル0とほぼ同等の結果が得られる。一方、モデル1では、TSS人数が約110まではモデル0とほぼ同等の結果を得ることが出来るが、TSS人数が110以上になるとモデル0から大きく異なった値が得られる。ここで、TSS人数が110のときのディスクの利用率を見ると、図6より、約40%である。CPUがボトルネックとなっている場合は、TSS人数に関わらずモデル2の結果はモデル0のそれと大きく異なっているが(図4)、図5よりCPUの利用率は常に40%を超えていることが分かる。従って、ボトルネックになっている装置の利用率がある値(この場合約40%)以下であるときはその装置のモデルの簡略化を行ってもよいが、利用率がその値を超えているときはその装置を忠実にモデル化しなければならないと考えられる。

4. 大型計算機システムの性能評価

最後に、2.2で構築したモデルを用いて大阪大学大型計算機センターの計算機システムの性能評価を行い、バッチジョブが TSSジョブに及ぼす影響を中心に調べていく。なお、特に断らない限り、パラメータについては、2.2で得た値を用いる。

4.1. バッチジョブが TSSジョブに及ぼす影響

バッチジョブが TSSジョブに及ぼす影響を調べるために、バッチジョブがあるときとないときで TSS人数を変化させた場合の各性能指標を求める。

図8に TSSジョブ平均応答時間の変化を示す。図8

より、バッチジョブがない場合、TSSジョブ平均応答時間はTSS人数が125まではほとんど変化はないが、そこから徐々に増加を始め、135から急激な増加が起こることが分かる。一方、バッチジョブがある場合には、TSSジョブ平均応答時間はTSS人数が75まではあまり変化がないが、75から100までは急激に増加する。100から135までの間は、増加は頭打ちになるが、135を超えると再び急激な増加が見られる。そこで、バッチジョブ平均応答時間を見ると(図9)、TSS人数が75まではあまり変化がないが、それを超えると徐々に増加を始め、さらに100を超えると急激に増加する。これは、図8に於いてバッチジョブのある場合のTSSジョブ平均応答時間の増加が緩やかに始まり始めるときのTSS人数と一致する。

各装置の利用率を図10に示す。バッチジョブの有無にかかわらず、CPUの利用率が最も高く、これがボトルネックになっていることが分かる。CPU利用率に注目すると、バッチジョブがない場合はTSS人数が大きくなるに従って増加してゆき、TSS人数が200でほぼ飽和するまで滑らかな曲線を描くが、バッチジョブのある場合にはTSS人数100から135の間で増加が停滞する。この区間は図8においてTSSジョブ平均応答時間の増加が緩やかになる区間と一致する。更に、CPU利用率のTSSジョブ、バッチジョブの内訳(図11)を見ると、TSS人数135でバッチジョブがほぼ0となる。これはちょうど図8においてTSSジョブ平均応答時間が再び急激な増加を始めるときのTSS人数と一致する。

これらの結果より、バッチジョブがある場合、TSS人数が100から135の区間において処理系内でなんらかの現象が起こっているものと考えられる。以下、この現象について考察する。

4.2. MPLがシステムに及ぼす影響

処理系内ジョブ数がMPLに達するまでは、TSSジョブはバッチジョブを追い出すことなく処理系内に入る。バッチジョブはTSSジョブに比べ、1つのジョブを処理するのに必要なCPUサービス時間(サービスデマンド)が非常に大きく(今回の場合、約100倍)、従って、バッチジョブはTSSジョブに比べて処理系内に長く留まるため、1つのバッチジョブが処理系にかけられる負荷は非常に大きなものとなる。よって、処理系内ジョブ数がMPLに達するまでは、バッチジョブがある場合のTSSジョブ平均応答時間はTSSジョブのみの場合に比べて大きくなる。ところが、処理系内ジョブ数がMPLに達すると、優先順位の高いTSSジョブは優先順位の低いバッチジョブの追い出しを始める。バッチジョブの追い出しが行われている間は、負荷の大きいバッチジョブが処理系を退去し、代わりに負荷の小さいTSSジョブが投入されるので処理系への負荷は軽減される。一方、この間TSS人数の増加に伴い、TSSジョブによる処理系への負荷が増加している。これら処理系への負荷の軽減と増加が互いに打ち消しあって、TSS人数100から135の間でTSSジョブ平均応答時間の増加が緩やかになっていると考えられる。また、バッチジョブ平均応答時間はTSSジョブによるバッチジョブの追い出しが始まると急激に増加する。更にTSS人数が増加していくと追い出すべきバッチジョブが処理系内に存在し

なくなるので、処理系内はTSSジョブだけで占められることになり、TSSジョブ平均応答時間はバッチジョブが存在しないときとほぼ同じになる。

上述の考察から処理系内の状態は、表2のように分類することができる。従って、各状態におけるTSSジョブ平均応答時間は、

状態1：TSSジョブ平均応答時間の曲線はMPLが無限大のときのものとはほぼ一致する。

状態2：TSSジョブ平均応答時間の曲線は状態1と状態3との遷移状態を表わしている。

状態3：TSSジョブ平均応答時間の曲線はバッチジョブがないときのものとはほぼ一致する。

と考えられる。これを確認するために、以下の2通りの方法によりシミュレーションを行なった。

(a) MPLの変更

処理系内ジョブ数がMPLに達するとバッチジョブの追い出しが始まるので、MPLを変えると状態1から状態2へ移行するときのTSS人数が変わる。例えば、MPLが小さくなると、より少ないTSS人数で状態2への移行が始まるものと思われる。

(b) バッチジョブの処理系内最大数の変更

処理系内ジョブ数は、バッチジョブ数とTSSジョブ数の和に等しいので、バッチジョブの処理系内最大数を変えると状態1から状態2へ移行するときのTSS人数が変わる。例えば、バッチジョブの処理系内最大数が小さくなると、より大きなTSS人数で状態2への移行が始まるものと思われる。

(a)ではMPLが80、120、 ∞ の場合、(b)ではバッチジョブの処理系内最大数が0、12、18の場合についてのTSSジョブ平均応答時間を図12、13に示す。図12より、状態1から状態2へ移行するときのTSS人数は、MPLが80、120のとき、それぞれ約80、100である。図13より、状態2へ移行するときのTSS人数は、バッチジョブの処理系内最大数が12、18のとき、それぞれ約110、100である。これらの結果より、予想は正しかったといえる。また、図12より、MPLのある程度下げること(場合80以下)か、あるいは図13よりバッチジョブの処理系内最大数を抑える(この場合12以下)ことにより、TSSジョブ平均応答時間が急激に増加するときのTSS人数を約135とすることができ、従って、TSSジョブ平均応答時間の振舞いをバッチジョブが存在しないときとほぼ同じにすることができる。但し、いずれの場合もバッチジョブ平均応答時間は、当然のことながら増加する。

5. おわりに

大型計算機システムの待ち行列網によるモデル化について、情報の収集からモデルの構築法、検証までを系統立てて示し、具体例として大阪大学大型計算機センターの計算機システムにそのモデルを適用した。

更に、シミュレーション時間の短縮を目的として、待ち行列網モデル内の待ち行列数を減らした簡略化モデルを考え、簡略化の妥当性を検討した。この結果、ボトルネックとなっている装置の利用率がある値以上であるときはその装置のモデルの簡略化を行ってはい

らないという結論を得た。本論文ではボトルネックが移行しない場合についてのみ検討を行なったが、今後の課題として、パラメータの値を変化させることによってボトルネックが移行するモデルについても検討する必要があると思われる。

また、このモデルを利用して大阪大学大型計算機センターの計算機システムについてバッチジョブとTSSジョブが相互に及ぼしあう影響に焦点をあててシミュレーションにより性能評価を行い、定量的に結果の解析を行なった。これにより、TSSジョブによるバッチジョブの追い出しに処理系内のジョブの制限数が大きな影響を及ぼすことが示された。これを基に、多次元処理システム内のバッチジョブ許容数、MPLを定量的に示した。今後は、思考時間やバッチジョブのサービスデマンドなどの変化がバッチジョブ追い出しに及ぼす影響をについても調べる必要があると思われる。

文献

- [1] C. H. Sauer and K. M. Chandy: "Computer Systems Performance Modeling", Prentice-Hall (1981).
- [2] H. Kobayashi: "Modeling and Analysis: An Introduction to System Performance Evaluation Methodology", Addison-Wesley (1978).
- [3] D. Ferrari, G. Serazzi and A. Zeigner: "Measurement and Tuning of Computer Systems", Prentice-Hall (1983).
- [4] C. H. Sauer, E. A. MacNair and J. F. Kurose: "The Research Queueing Package Version 2: CMS User's Guide", IBM Res. Rep. RA-139(1982).
- [5] F. Baskett, K. M. Chandy, R. R. Muntz and J. Paracios: "Open, Closed and Mixed Networks with Different Classes of Customers", J. A. C. M., Vol. 22, No. 2, pp. 248-260 (1975).

- [6] C. H. Sauer, E. A. MacNair and J. F. Kurose: "Queueing Network Simulations of Computer Communication", IEEE J. Sel. Areas Commun., Vol. SAC-2, No. 1, pp. 203-220 (1984).
- [7] C. H. Sauer, E. A. MacNair and J. F. Kurose: "The Research Queueing Package Version 2: Introduction and Examples", IBM Res. Rep., RA-138 (1982).
- [8] 村田正幸: "待ち行列システムのモデリング言語 RESQ - RESEARCH Queueing package-", オペレーションズ・リサーチ, Vol. 30, No. 7, pp. 442-449 (1985).

パッシブキュー (Passive Queue) について

例えば、マルチプログラミングシステムでメモリが空くまでプログラムの実行が待たされる場合などを待ち行列モデルで表現するために、パッシブキューという概念が考えられている(図14)[4]。客は、様々な資源を表わすトークン(Token、前述の例ではメモリ)をアロケートノード(Allocate Node)で獲得した後、システム内に入り、リリースノード(ReleaseNode)でトークンを返してからシステムを去る。アロケートノードにおいてトークンが不足したときは、客はこのノードで待たされる。パッシブキューではトークンの渡し方によってスケジューリングの方法が決められる。

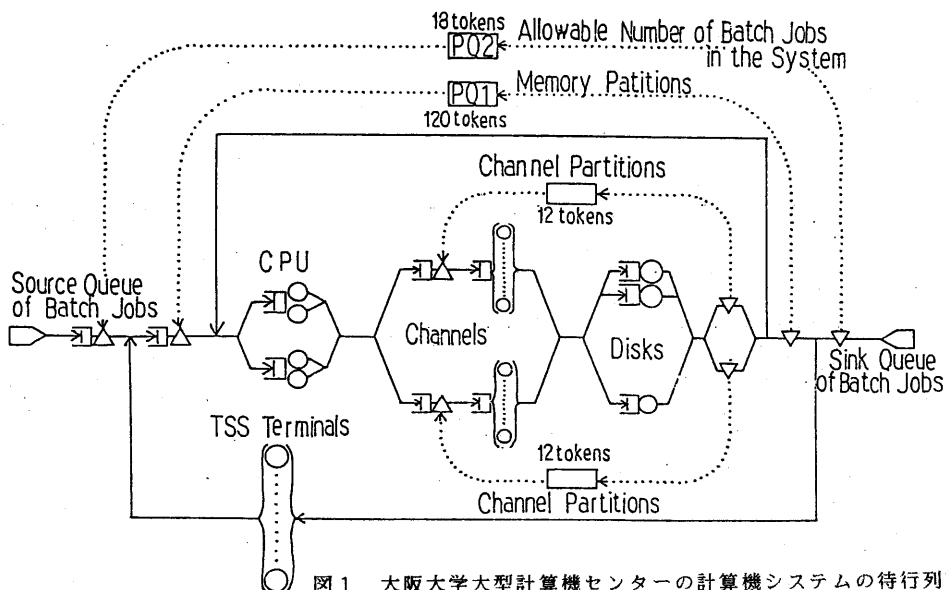


図1 大阪大学大型計算機センターの計算機システムの待行列網モデル

表 1: 統計情報値とシミュレーション結果の比較

	昭和62年2月21日		昭和62年4月7日	
	統計値	シミュレーション値	統計値	シミュレーション値
CPU利用率(%)	69.52	67.02	14.48	16.23
ディスク利用率(%)	9.15	9.69	3.58	3.01
チャンネル利用率(%)	7.80	8.08	3.04	3.12

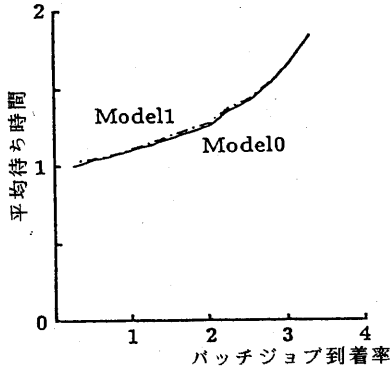


図2 ディスク簡略化によるディスク平均待ち時間の比較

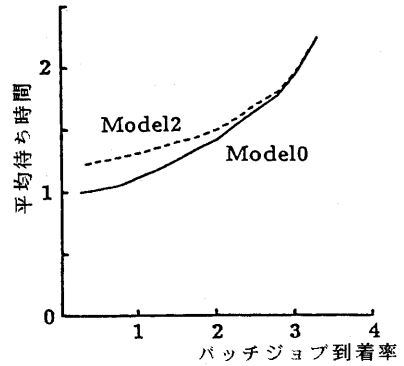


図3 CPU簡略化によるCPU平均待ち時間の比較

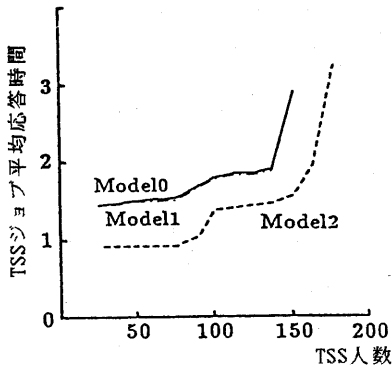


図4 TSSジョブ平均応答時間の比較

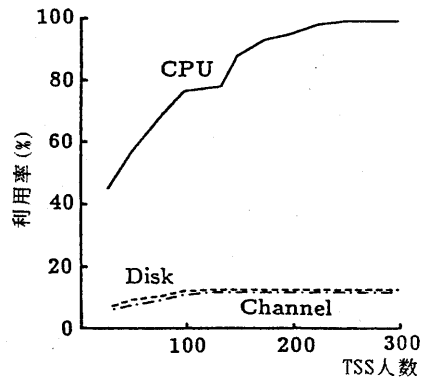


図5 各装置の利用率

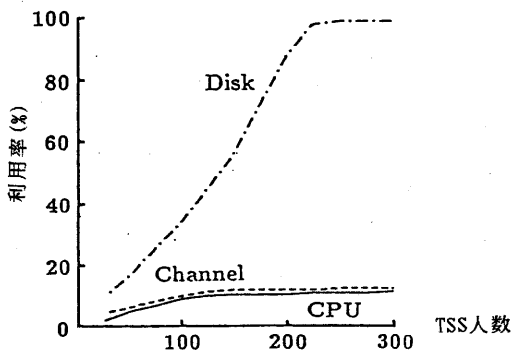


図6 ディスクボトルネックの場合の各装置の利用率

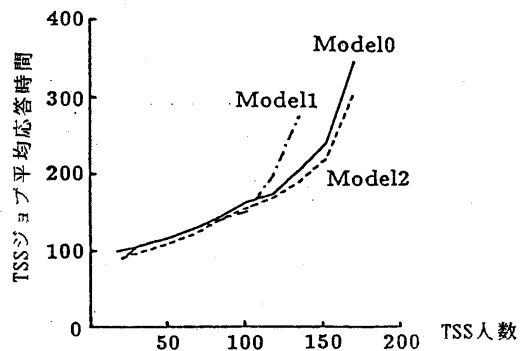


図7 ディスクボトルネックの場合のTSSジョブ平均応答時間の比較

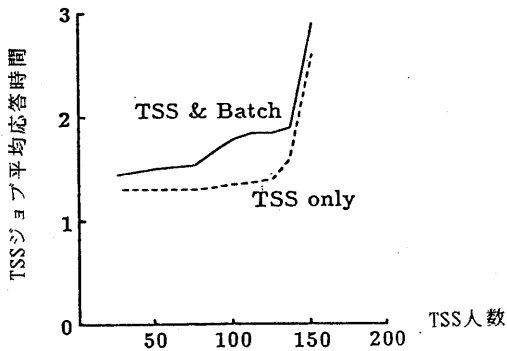


図8 バッチジョブがTSSジョブに与える影響

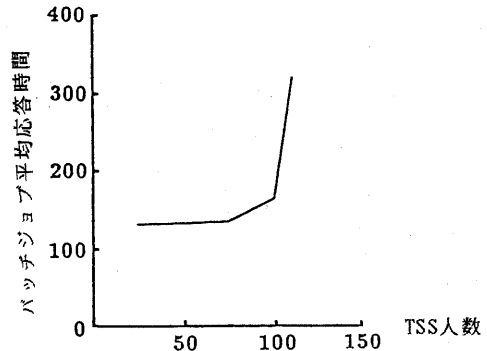


図9 バッチジョブ平均応答時間

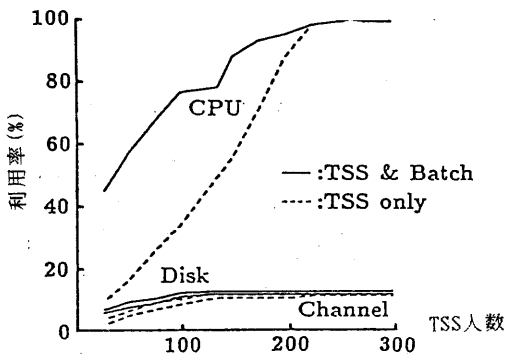


図10 各装置の利用率

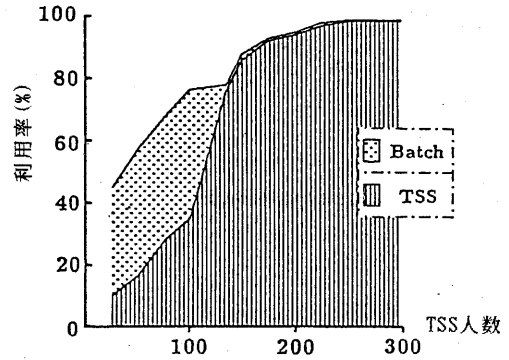


図11 CPU利用率の内訳

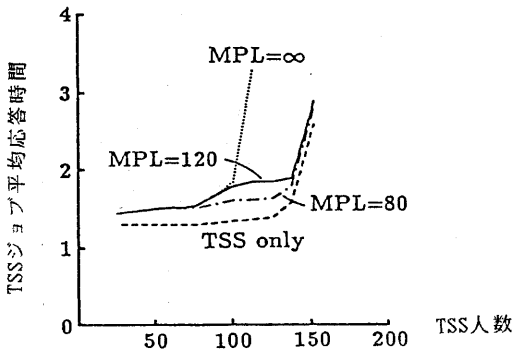


図12 MPLがTSSジョブに及ぼす影響

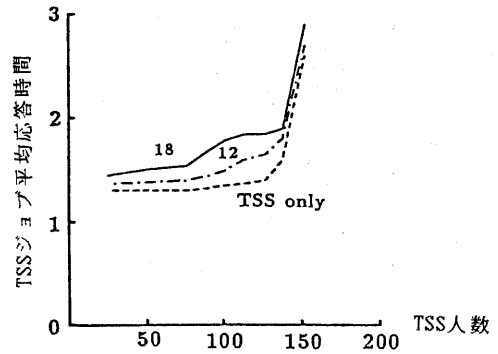


図13 バッチジョブの処理系内最大数がTSSジョブに及ぼす影響

表2: 処理系内の状態

状態	TSS 人数	TSS ジョブとバッチジョブの関係
1	0-100	TSS ジョブ, バッチジョブは互いに影響を及ぼさない.
2	100-135	TSS ジョブによってバッチジョブの追い出しが行われている.
3.	135-∞	バッチジョブの追い出しが完了し, 処理系内には TSS ジョブのみが存在する.

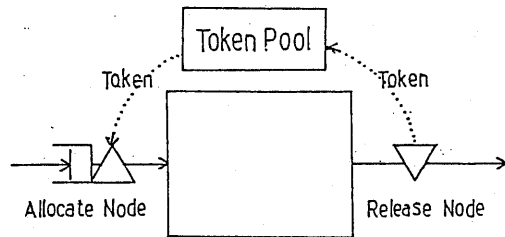


図14 パッシブキュー