

計算機システムにおける性能管理の一方式と それを用いた実験†

吉住 誠一†† 堀越 彌††

計算機システムの複雑化とともにその性能を決める要因が多岐にわたり、システム性能を常に最適に保とうとする性能管理の仕事が非常にむずかしくなっている。とくに大規模システムにおいては多様な処理形態の共存が一般的であり、負荷のピーク時などに発生する性能上の問題の正確な把握と解決には、専門家による多大のマン・パワーを必要としているのが現状である。性能管理の容易化、自動化に対する要求は強い。計算機システムの性能評価の手法には、モニタリング、ベンチマーク・テスト、モデリング、シミュレーション等があり、それぞれに対して多くの研究がなされてきているが、上記の問題を全面的に解決するものはいまだ登場していないといえよう。本論文ではシステム性能管理の最大の難所は、システム性能上のボトルネックの正確な抽出にあると考え、それを定量的に表現できる性能指標を新たに考案し、実験システムでのモニタリング・システムおよびその解析システムを開発した。これらの両システムの利用により、システム性能およびボトルネックの所在が具体的、定量的に示されるため、その利用者はシステムの性能状況を常に把握しておくことが可能であり、性能向上のために取るべき手段も明らかになる。本論文では実験システムにおいて上記手法の有効性を確認しており、性能管理の自動化への方向を示している。

1. ま え が き

計算機システムの利用形態の拡大、多様化、複雑化に伴い、計算機システムの処理能力の問題を扱う性能管理の仕事がしだいにむずかしくなっている。実システムで性能低下が生じた場合、その原因の追究と対策の実施に多数の専門技術者を必要とする現状では、性能管理の充実に要する費用が、性能向上による利得を上回ることが多いため、性能管理を怠ることにより浮く費用をハードウェアの拡充に充てることによってシステム性能の低下を防ぐほうが得策である¹⁾という認識が一般的になりつつあるといえよう。それだけに、性能管理の容易化、さらには自動化への要求には非常に根強いものがある。

性能管理には、問題の原因究明、対策の実施の2部分があり、前者がとくにむずかしく数多くの研究がなされている。

原因究明のためには、システムの現状を十分理解しておくことが重要であり²⁾、そのため各種のモニタリング手法³⁾およびツール類⁴⁾の研究、システムのモデリング技法⁵⁾の研究が活発に行われてきた。これらはおもにシステム・リソースの利用率を求め、問題点を抽出する方式に関するものであるが、実システムでの

性能管理においては、システム・リソース利用率の値が適正かどうかの判断がむずかしく、利用率だけでは真の原因究明につながらないことも多い。

これに対し、システムの処理の効率性を定量的に把握し、それをシステム性能上のボトルネックの抽出に利用することを目的として、本論文では新たに「実効的サービス率 ESR (effective service rate)」なる性能指標を導入し(2章)、それを用いた実験結果を示す(3章)。

2. 性能指標 ESR (effective service rate) について

2.1 考え方の背景

計算機システムは中央処理装置(CPU)、入出力装置(I/O)から構成されており、処理の単位であるタスクが、それらのリソースを順次要求し、使用しながら所定の処理を実行する。一つのプログラムを単独で実行させた場合、各時点において、CPUあるいはI/Oのいずれか一方が使用されているが、複数プログラムを同時に実行した場合*には一般に、CPU、I/Oが並行して使用されており、多重度が1の場合に比べてシステムの処理量が多く、よい高い性能が発揮されると考えられる。以下において、これらの事項の定量的表現を考える。

† A Methodology of Computer Performance Management with an Experimental Result by SEIICHI YOSHIZUMI and HISASHI HORIKOSHI (The 8th Dept., Central Research Laboratory, Hitachi, Ltd.).

†† (株)日立製作所中央研究所第8部

* N個のプログラムが同時に実行されている状態を指して、多重度Nでの実行と称する。より正確には、システム内で起動されているパーティションあるいは空間の数を指す。一つのプログラムを単独で実行した場合には多重度が1である。

多重度1の状態においては、CPU、I/O 使用中のいずれの場合にも、そのタスクは「1の速度」で処理を進めていると定義することにする。複数多重度の状態においては、複数個のタスクが CPU、I/O を使用しながら処理を進めている。ある時点で使用中の CPU、I/O はいずれかのタスクから使用されている訳で、そのタスクから見れば、自分は「1の速度」で処理をしていることになる。したがってある時点で使用中の CPU、I/O の総数を M とすれば、 M 個のタスクがそれぞれ「1の速度」で処理を実行しているので、システム性能は多重度1の場合の M 倍が達成されることになる。これはその時点におけるシステムのサービスの効率性を表現しており、その意味でこの値のことをシステムのサービス率 (service rate—SR と略す) と呼ぶことにする。システムのサービスのなかには実質上ユーザ・プログラムの処理に寄与しない部分 (ページング処理、スワッピング処理など) があり、SR からそれを取り除いたものを ESR (effective service rate) として新たに定義し、これによりシステムの実効的性能が表現されると考えることにする。

次にシステムにおける性能上のボトルネックについて考える。ジョブ実行の多重度を N とすれば、上記の定義によりある時点において $(N-SR)$ は処理を行っていないタスクの数であり、それらのタスクは CPU、I/O、主記憶 (ページ・イン待ち、スワップ・イン待ち)、あるいは逐次的使用を必要とするソフトウェア・リソースでの使用要求待ち行列などにおいて待っていることになる。各リソースでの待ちの長さをその時点におけるそのリソースの BD (bottleneck degree) と称すれば、その合計値が $(N-SR)$ に等しくなる。このように考えればシステムの SR 向上を阻害しているものが各リソースの BD であり、その値の大きなリソースが性能上のボトルネックになっていると考えられる。

2.2 性能指標 ESR の定義

前節でその考え方の説明を行った諸指標 (SR, ESR, BD) の正確な定義を以下に与える。

ある時間帯 (t_1, t_2) におけるシステムのサービス率 SR は、前節において定義した各時刻 $t \in (t_1, t_2)$ におけるサービス率 $SR(t)$ の時間平均で与えられる。

$$\begin{aligned} SR &= \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} SR(t) dt \\ &= \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \sum_{\rho \in R} U_{\rho}(t) dt \end{aligned}$$

$$\begin{aligned} &= \frac{1}{t_2 - t_1} \sum_{\rho \in R} \int_{t_1}^{t_2} U_{\rho}(t) dt \\ &= \sum_{\rho \in R} U_{\rho} \end{aligned} \quad (1)$$

ただし、

R : システム・リソース (CPU, I/O) の全体

$U_{\rho}(t)$: 時刻 t においてリソース ρ が使用中の場合 1, それ以外は 0

$U_{\rho} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} U_{\rho}(t) dt$ であり、リソース ρ の (t_1, t_2) での平均利用率

したがって SR とはシステム・リソースの利用率の合計にほかならない。次に実効的サービス率 ESR を定義する。

$$ESR = \sum_{\tau \in T} ESR_{\tau} \quad (2)$$

$$= \sum_{\tau \in T} \sum_{\rho \in R} \int_{t_1}^{t_2} C_{\tau, \rho}(t) \cdot E_{\rho}(t) dt$$

ただし、

T : その時点でシステム内に存在するタスクの集合

ESR_{τ} : タスク $\tau \in T$ の ESR

$C_{\tau, \rho}(t)$: 時刻 t において、システム・リソース ρ がタスク τ によって使用されている時 1, 他の場合 0

$E_{\rho}(t)$: 時刻 t におけるリソース ρ の有効度を表す。たとえば、リソース ρ がページング処理に使用されているとき $E_{\rho}(t) = 0$ とする

最後に BD を定義する。

$$BD = \sum_{\rho \in R'} BD_{\rho}$$

$$= \sum_{\rho \in R'} \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} BD_{\rho}(t) dt \quad (3)$$

ただし、

$BD_{\rho}(t)$: 時点 t でのリソース ρ における待行列の長さ

BD_{ρ} : 区間 (t_1, t_2) における $BD_{\rho}(t)$ の時間平均

R' : 前記 R (CPU, I/O) に逐次使用リソースを追加したシステム・リソースの全体集合

以上性能指標 SR, ESR, BD の正確な定義を与え、これらが以下の諸量から算出できることを明らかにした。

- (1) 各システム・リソースの利用率
- (2) 各システム・リソースの利用率のうちページング処理およびスワッピング処理に使用されている部分
- (3) 各システム・リソースおよび逐次使用リソースでの待行列の長さ

これらの諸量を実験システムから得るために開発した性能評価システムについては次節で説明し、以下において、ここで定義した性能指標 ESR の意味に触れることにする。

(1) 各タスクの ESR は 1 を越えない

多重度 1 の場合、ページングなどのムダ処理のない理想的な状態においては、そのタスクの ESR は SR に等しく、その値は 1 となる。複数多重度では複数タスクがリソースを競合し合って使用するため各タスクの ESR は低下し、1 以下になる。ESR は、したがって多重度 1 の場合の理想的状態の処理に比べてどの程度タスクの処理が進んでいるのかを表している。

(2) ESR はシステムの応答性能とともにスループットを表現する

各タスクの ESR は上記のようにタスクの進捗度を表しており、ジョブのターン・アラウンド・タイムや応答時間は ESR に反比例の関係をもっている。たとえば ESR が 0.5 ということは、同じ処理をするのに単独で実行した場合の 2 倍の時間を要することを意味する。一方、各タスクの ESR の合計値であるシステムの ESR は多重度 1 の場合に比べてシステム・スループットが何倍になっているのかを示す指標となる。このように ESR は応答性能、スループットを同時に表現できるという特性をもっている。

2.3 ESR 計測評価システム ETAS

これまでに定義した各種の性能指標を実システムから得るための性能評価システム ETAS (event tracer and analyzer system) を開発した。ETAS は実システムの稼動中に性能データを採取する部分、そのデータを解析し性能指標を算出する部分の 2 部分から構成されている。

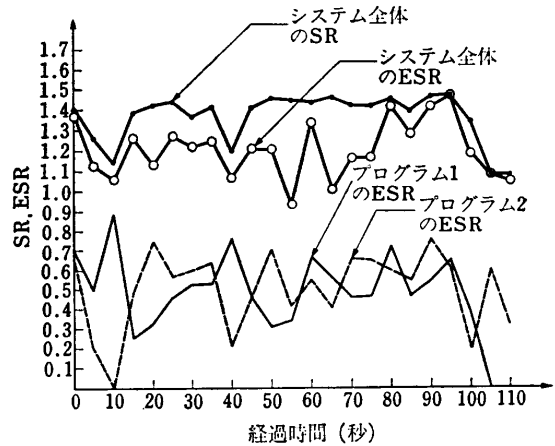
前者による実験システムからのデータ採取は以下の方法で行われる。オペレーティング・システム内の所定のルーチン群にデータ採取のためのフック*を埋蔵し、実行時に発生する MC 命令割込みを契機としてデータ採取ルーチンを起動し、そこで所定のデータ採取を行い外部のファイルに時系列に出力する。

ETAS の後者の部分は、前者の出力データを入力とし、前記の各種性能指標を算出する。これにより実システムにおける SR, ESR, BD などの性能指標を得ることができる。

* ハードウェア側の提供する MC (Monitor Call) 命令を利用する。実行時に割込みを発生する。

3. 実験結果

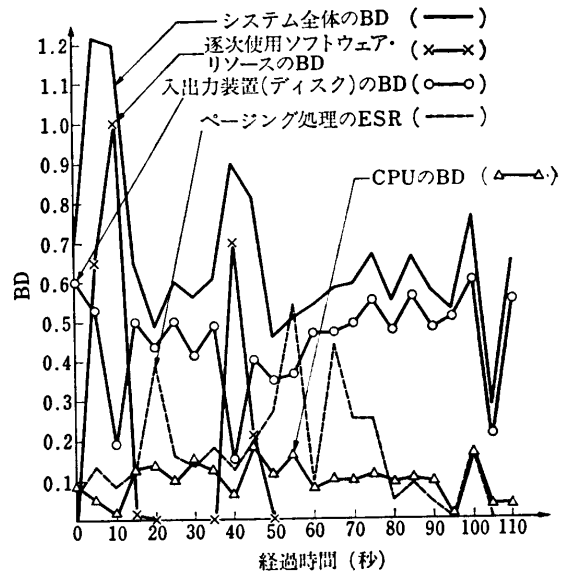
本章では、まず結果の概要について述べ、前章で定義した各種の性能指標 SR, ESR, BD の具体的な適用例を示す。次に他の諸量との関係を含めて詳細な検討を行い、最後にそれらの結果に基づいて、システム上のボトルネック摘出および解除のための検討を行う。



多重度 2 でプログラムが実行されている。
SR: service rate,
ESR: effective service rate

図 1 SR, ESR の変化

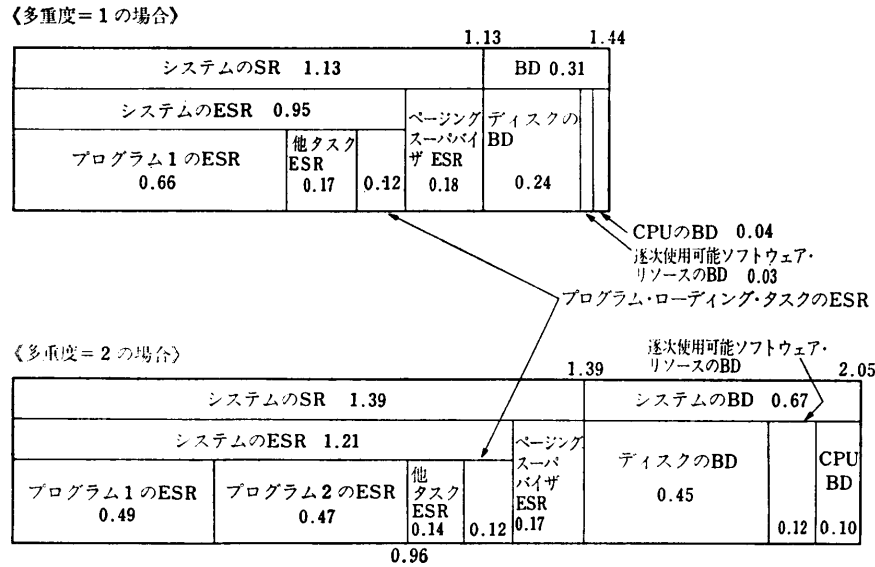
Fig. 1 Graph of monitored SR and ESR.



多重度 2 でプログラムが実行されている。

図 2 BD (bottleneck degree) の変化

Fig. 2 Graph of monitored BD.



プログラム 1, 2 は同じプログラムである.
SR: service rate, ESR: Effective service rate, BD: bottleneck degree

図 3 システム性能ダイヤグラム

Fig. 3 Diagram of system performance.

3.1 結果の概要

バッチ・システムにおいて性能評価システム ETAS を用い、小型のバッチ・ジョブを対象として実験を行った。その結果を図 1, 2 に示す。

図 1 には、システム全体のサービス率 (SR), 実効的サービス率 (ESR), および各ジョブの ESR の変化が示されている。5 秒ごとに SR, ESR の平均値がプロットされている。各ジョブの ESR の変動は大きく、ジョブ 1 では 0.26~0.88, 平均 0.49, ジョブ 2 では 0.0~0.75, 平均 0.47 となっている。2 個のジョブが、互いに、システム・リソースで競合しながら実行されてゆく状況が推測できる。

図 2 に各種システムリソースにおける BD (bottleneck degree) の変化を、図 1 と同一の時間軸に沿ってプロットした。システムリソースとしては、CPU, I/O (ディスク), 逐次使用のソフトウェア・リソースをとりあげ、さらにシステム性能にとってマイナスの要因となるページング処理の ESR* もあわせ示した。上記 3 種のシステム・リソースにおける BD の総和がシステム全体の BD である。時間の経過とともに、各種 BD が大きく変化している様子がよくうかがえる。CPU における BD は比較的安定しているが、逐次使用ソフトウェア・リソースにおける BD の変化

がとくに大きく、ジョブの負荷に大きく依存していることが推察される。

以上、図 1, 2 のいずれにおいても ESR, BD などの諸量が大きく変動していることが示されている。これらの諸量の変動要因および相互関係の詳細な解析については次節 3.2 に譲り、本節ではそれらの平均的な姿を眺めることにする。

図 3 は性能ダイヤグラムと称する図で、SR, ESR, BD の平均値とその内訳を示しており、多重度が 1 の場合が上に、2 の場合が下に図示されている。システム性能がシステムの ESR で表現されるので、多重度 1, 2 でのシステムの ESR がそれぞれ 0.95, 1.21 であることは、多重度を 1 から 2 に高めることによって、システム性能が 27.5% 向上していることを意味する。また BD の内訳により、ディスク A がシステム性能上の最大のボトルネックになっていることがわかる。

3.2 結果の検討

実験的に得られている性能指標 SR, ESR, BD の実測結果に対して以下の 2 点から検討を加える。

- ジョブ処理時間と ESR の関係: 従来より用いられてきたジョブ処理経過時間による直観的評価法と、ESR による新しい評価法を比較し、両者の利害得失を明らかにする。
- 性能指標 SR, ESR, BD の変動要因の分析
 - (1) ジョブ処理時間と ESR の関係

* ページング処理を行うシステム・タスクの ESR を意味する。

表 1 ESR と諸指標の関係

Table 1 Relationship between ESR and conventional performance indices.

比較項目	多重度=1		多重度=2	
	パーティション4	パーティション4	パーティション4	パーティション6
経過時間	51秒(1)		94秒(1.84)	95秒(1.86)
E S R	86.18(1)		52.15(1/1.65)	53.95(1/1.60)
システム・リソースの占有時間	C P U	24.31秒(1)	24.83秒(1.02)	24.30秒(0.99)
	ディスクA	18.54秒(1)	24.48秒(1.32)	26.80秒(1.45)
	ディスクB	0.23秒(1)	0.23秒(1.00)	0.28秒(1.17)
	合計	43.08秒(1)	49.54秒(1.15)	51.37秒(1.19)

ESR: effective service rate

() 内はシングル・ランの場合を1とした比率を表す。

通常システム性能というものは、ジョブ処理の開始から終了までの経過時間をベースとして考える。ESRと経過時間の関係を調べるために、多重度1,2でジョブを実行した場合のそれぞれの実験データを表1に示す。表1によれば、多重度2で実行した場合、多重度1のときに比べて、経過時間が1.84~1.86倍に伸びていることがわかる。すなわちジョブの処理速度が多重度1の外乱のない状態に比べて、それぞれ1/1.84, 1/1.86に低下していることを意味する。一方、ESRは第2章で定義したように、各ジョブの実質的なシステム・リソースの使用量であり、処理速度を表しているが、表1によれば、ESRで表現したジョブの処理速度は、多重度が2の場合、多重度が1のときに比べて1/1.60~1/1.65に低下しており、前記の経過時間による処理速度の評価結果に比べ、約1割程度の差異が生じている。この差の原因は、多重度1と2でのディスク入出力時間の差に由来する。表1におけるシステム・リソース占有時間の項を見れば、ディスクAにおいて多重度2の場合には多重度1の場合に比べて1.32~1.45倍の入出力時間の増加を招いている。多重度の増加とともにディスクAでのシークのためのアームの移動が激しくなり、平均シーク時間が伸びたためである。入出力時間の増加はそのままESRの増加につながり、見掛け上処理速度が向上しているように見える原因となる。このような状況は2章ESRの定義式一式(2)における係数 E_p を1.0としたために生じたものであり、多重度の増加とともに E_p を低減させることにより対処できるが、 E_p の値の具体的な設定については、システムの構成やジョブの特性に依存するところが多く、多種多様なデータ、知見が必要であり、今後の課題として考えたい。

(2) 性能指標 SR, ESR, BD の変動要因の分析
SR の変化が図1に示されている。SR の変動をもたらし要因としては、一般的に以下の2件が挙げられる。

- (i) システム内に生ずる待ち。待ち行列が生ずれば、そこでタスクが待ち状態になり、それだけシステム・リソースを活用する機会が減る。
- (ii) プログラム特性のばらつき。1タスクであってもCPUとI/Oを同時に使用することが可能である。CPUとI/Oのオーバラップの程度によって、SRの値が変動する。

上記(ii)のCPUとI/Oのオーバラップは、多重度の増大ともなってその割合が高まる。また、多重度の増大は、個々のジョブの特性のSRに及ぼす影響を減少させるので、ここでは(i)の要因に焦点を合わせ、その詳細解析を行う。

システム内に発生する待行列としては以下のものが考えられる。

- (i) CPU 空き待ち
- (ii) I/O 空き待ち
- (iii) 逐次使用可能なソフトウェア・リソースの空き待ち

これらのシステム・リソースにおける待ちの長さの時間平均が、各リソースのBDである。図2の結果によれば以下の事項が指摘される。

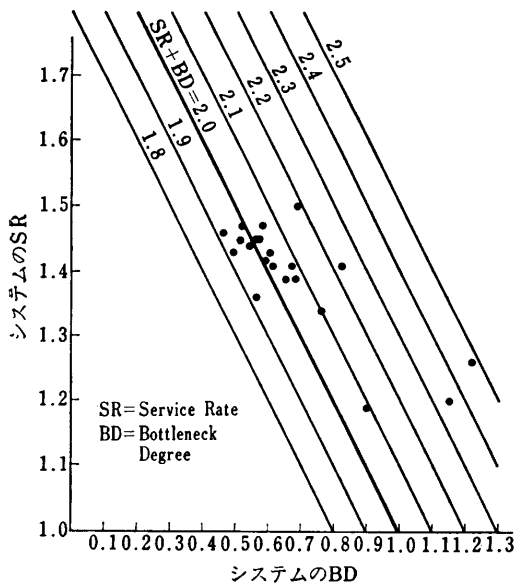
- (i) CPUのBDは、この実験のケースでは比較的変動が小さく、0.1前後を小刻みに動いている。平均値も0.102と小さな値である。これは、ジョブ処理多重度が2と低いこと、取り上げたジョブの特性として、CPU利用率が低く安定していること、の2点が主要な原因であるといえよう。
- (ii) ディスクAのBDは変動が激しく、その値も大きい。その平均値が0.447である。ディスクAへの入出力要求は待たされる確率が非常に高いことを表している。
- (iii) 逐次使用リソースでのBDは特定のインターバルにおいて集中的に高くなる。BDの平均値は、0.12と比較的小さいが、待ちが生じたときのBDの個々の値は大きくなっている。表2に関連するソフトウェア・リソースの名称を記す。この表に登場するリソースはすべてジョブ管理の使用するものである。

さて、新たに設定した指標であるBDがその名称の

表 2 逐次使用可能ソフトウェア・リソース

Table 2 List of serially usable software resources.

ソフトウェア・リソース名	項 目		
	役 割	BD≠0 の区間 (図 1, 2 の経過 時間に対応) 秒 秒	BD の値
SYSJDJSD. Q1	SYS1. SYSJOBQE の QCR	0~5	45
		5~10	64
SYSJDJSD. Q2	SYS1. SYSJOBQE の マスタ QCR	10~15	2
SYSJDJSD. Q4	Allocator のロック	20~25	64
		25~30	100
		30~35	2
SYSJDJSD. Q5	UCB ロック (ターミ ネータとの同期)	55~60	70
		60~65	22



斜線は $SR+BD=一定$ の線。線上の数字は $SR+BD$ の値。

図 4 SR と BD の相関関係

Fig. 4 Relationship between SR and BD.

示すとおり、SR の向上をどのように阻害しているかを調べる。各システム・リソースでの BD の総計(システムの BD と称す)を横軸に、システムの SR を縦軸にとって図 1, 2 の結果をプロットし、図 4 に示す。

これによれば SR と BD の間に強い負の相関が見られ、SR の低下は BD の増加によりもたらされているといえよう。2.1 節において SR, BD の基本的考え方の説明の中で、ジョブ実行の多重度を N とすれば、 $N=SR+BD$ という関係が成り立つことを述べた。図 4 の中の斜線は $SR+BD=一定$ の線を表しており、多重度を 2 としたこの実験において原理的に成り立つ $SR+BD=2.0$ の線と実験値との比較ができ

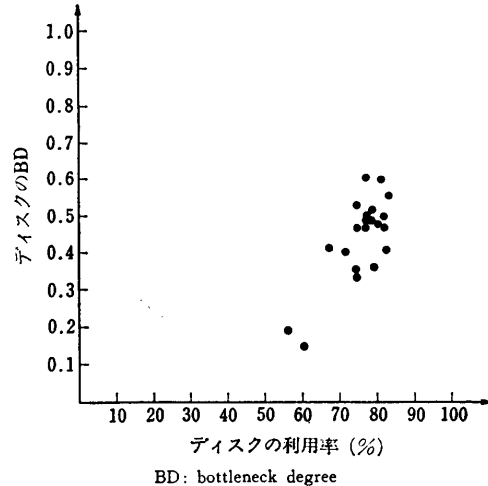


図 5 利用率と BD の関係

Fig. 5 Relationship between disk utilization and its BD.

る。実験値の多くは、 $SR+BD=1.9\sim 2.1$ の間に分布しており、 $N=SR+BD$ の原理式がほぼ成立する場合の多いことがわかる。しかし、原理式から大きくはずれた実験値も散見される。その原因としては、多重度は 2 であるが、実はその他にシステム・タスク(ロギング/アカウンティング、システム監視、コンソール入出力、プログラム・ローディングなど)が動作していること、ジョブ処理において CPU, I/O 処理のオーバラップがあること、ETAS の計測誤差、などが挙げられる。

BD はそのシステム・リソースにおける待ち行列の長さの時間平均として定義されている。一般にシステム・リソースにおける待ち行列長とその利用率との間には強い相関関係のあることが予想されるが、ここではその一例として、ディスク A における両者の関係を図示した(図 5)。利用率が 60% を超えると待ち行列長、すなわちディスク A の BD が急激に増大する様子が観察できる。

3.3 ボトルネックの抽出と解除について

ジョブの実行開始から終了までの各種性能指標の平均的な姿を図 3 の性能ダイヤグラムに示した。図には SR, ESR, BD とその内訳が示されているが、これらのうち、純粋にユーザ・ジョブの処理に対して有効に働いている部分は、ユーザ・タスクの ESR である。ユーザにとっては各 ESR がシステムの応答性能を表現し、その総和がシステム・スループットを表していることになる。したがって、システムの SR の構成要素のうち、上記ユーザ・タスクの ESR を除いたも

の,すなわち,システム・タスクの ESR, およびページング・オーバヘッドの部分はシステム・リソースが無駄に消費されていると見なすことができる。それらが解消されれば,それに応じてユーザ・タスクの ESR が増加し,システム・スループットの向上が期待できる。

BD の部分はシステム・リソースを消費しているわけではなく,むしろ,システム・リソースを有効に活用することのできるタスクをどの程度殺しているかを表している。たとえば図3の(B)において,ディスク A の BD が 0.447 であることは,ESR に対し最大 0.447 の増加をもたらすはずのタスク活動を殺していることを意味する。

さて以上の議論に基づき,システム性能の阻害要因を BD の高いものから以下に列挙する。

- | | |
|----------------------|--------|
| 1) ディスク A | 0.447 |
| 2) ページング処理 | 0.173* |
| 3) プログラム・ローディング | 0.119* |
| 4) 逐次使用可能ソフトウェア・リソース | 0.118 |
| 5) CPU | 0.102 |

これによればディスク A が,性能上最大のボトルネックになっていることがわかる。したがって,この場合,システム性能を向上させるためには,ディスク A の BD を低下させること,すなわちディスク A 使用要求の待ち行列の解消を第一に考えるべきである。以下,その具体的な手段を検討する。

ディスク A の BD が高くなっている原因は以下の 2 点にあると考えられる。

- (1) ディスク A に格納されているファイルへのアクセス頻度が高いこと。
 - (2) ディスク A への平均入出力時間が長いこと。
- これに対する対策としては以下のものがある。

- (1) ファイルごとのアクセス頻度情報に基づき,ファイルの再配置を行い,各ディスクにおけるアクセス頻度のバランスをとる。
- (2) ディスク内でのファイル配置については,アクセス頻度の高いファイルをディスクの中央シリンドラの近傍に順次配置していくことにより,ディスク・アームの移動にともなうシーク時間の減少を実現し,入出力時間を短くする。

これらの手段によるファイル配置の最適化によっ

て,システム性能上のボトルネックであったディスク A での待ちは大幅に解消されることが予想される。ディスク A での待ちを解消されたタスクは,他のシステム・リソースの利用に回り,システムの性能向上に寄与することが予想されるが,一方,そのタスクは他のシステム・リソースでの待行列の増大をもたらす,あるいはページング・オーバヘッドの増加をもたらす可能性があり,ディスク A の $BD=0.447$ のうちの程度がシステムの ESR の向上に向けられるかについては予測がむずかしい。仮にその半分が ESR の向上に向かうとすれば,システムの ESR は

$$1.213 + 0.447 \times 1/2 = 1.436$$

となり,約 18% の性能向上が得られる。

次に BD の大きなページング処理については,主記憶装置の増設,ページング方式の改良,多重度の調整など考えられる。プログラム・ローディング処理については,該当プログラムの常駐化により対処できるが一方ではページング・オーバヘッドの増大をもたらす可能性があり,両者をにらんだ決定が必要である。

逐次使用可能ソフトウェア・リソースおよび CPU の BD についてはその値が小さいので,この場合性能チューニングの効果はほとんど期待できないが,一般的には以下に述べるチューニング手段が考えられる。

逐次使用可能ソフトウェア・リソースについては,それを細分化し競合を減らすこと,あるいは,そのリソースを占有しているタスクの実行優先順位を高め,リソースの占有時間を短縮させるようなスケジューリング方式を採用することが有力な手段となる。

CPU については高性能 CPU に置換することが根本的であるが,ジョブ実行の多重度やジョブ・クラスの設定法の見直しなど運用面の改善も有効である。

4. む す び

計算機システムの処理の効率性を表現する指標「実効的サービス率 ESR」を新たに導入し,実験を通してその有効性の一端を示した。今後は,実際の場における性能管理に適用し,性能管理の容易化への効果を確認してゆくとともに,さらに性能管理の自動化を推進してゆく。

謝辞 なお,本研究の内容について貴重なご討論をいただいた電気通信大学 亀田寿夫助教ならびに(株)日立製作所中央研究所高橋栄主任研究員に感謝の意を表する。

* これらは,図3において BD の形では表現されておらず,該当するシステム・タスクの ESR として表されている。ここでは,しかし,性能阻害要因だと考え,BD と同等に取り扱うことにする。ESR, BD が互いに同一土俵で論じうる点が大きな特徴。

参 考 文 献

- 1) White, C. H. : System Tuning, p. 510, Infotech. State. of the Art Report (1977).
- 2) Bell, T. E., et al. : Frame-Work and Initial Phases for Computer Performance Improvement, Proc. AFIPS 1972 FJCC, pp. 1141-1154 (1972).
- 3) Lucas, H. C., Jr. : Performance Evaluation and Monitoring, *Comput. Surv.*, Vol. 3, No. 3, pp. 79-91 (1971).
- 4) Carlson, G. : How to Save Money with Computer Monitoring, Proc. ACM Conf., pp. 1018-1023 (1972).
- 5) Rees, R. L. D. : Performance Modelling and Prediction, Vol. 1, p. 254, Vol. 2, p. 437, Infotech. State. of the Art Report (1977).
- 6) 吉住 : 計算機システムの性能管理手法に関する考察, 情報処理学会第20回全国大会予稿集, pp. 977-978 (1979).

(昭和57年1月29日受付)

(昭和57年4月19日採録)