

データベース・システムの性能解析

梨 山 修, 村 井 正 和, 河 津 誠 一
(電 々 公 社 ・ 横 須 賀 通 所)

1. まえがき

企業又は国家活動等において、情報、データが有用な資源であるという認識は常識となってきた。その結果、既存のファイルを統合して、常に鮮度の高いデータを蓄積しておき、各部門がこれを共有して多目的に使用できるソフトウェア技術の発展が要望されている。又ハードウェア・コストの低下に比して、ソフトウェア・コストが相対的に高くなって行く傾向から、データとプログラムの独立性を保証することが要求されている。

今日、これらのニーズに応えるツールとして、データベース管理システムが一般に注目されるようになってきたが、まだ発展段階のソフトウェア技術であり、実用レベルのデータベース・システムを実現するとなると、いくつかの難しい問題を解決しなければならない状況である。現状のハードウェア技術の下で、実用レベルのデータベース・システムを構築する場合の大きな向題の1つは性能である。本論文は、こうした向題の解決の1方法として、データベース・システムに対する解析モデルを作成し、スループットと応答時間に関して解析を行なったものである。

2. データベース・システムの特徴分析

データベース・システムの性能を解析するに先立ち、データベースの技術的特徴と、それらの特徴がデータベース・システムの制御方式、処理方式にどのように関係しているかを考察する。図2.1はデータベースの技術的特徴と、それらの特徴を実現する手段、制御方式、処理方式の関係を概念的に示したものである。この中から、データベース・システムの性能と関係する制御方式、処理方式を抽出すると以下のようなものがある。

- (i) トランザクションのスケジューリング方式 (Resource limit scheduling / Resource free scheduling)
 - (ii) 排他制御方式 (排他制御単位: ページ/レコード)
 - (iii) ジャーナリング方式 (ジャーナリングの多重度: シングル/マルチ, ジャーナリングの単位: ページ/レコード, ジャーナリング用媒体: 磁気テープ/ドラム/パック)
 - (iv) バッファ管理方式 (バッファをタスク間で共有するかどうか: 共有バッファ/個別バッファ, スワッピング・アルゴリズム)
 - (v) 非同期処理方式 (非同期の度合: 完全非同期/部分非同期)
 - (vi) 負荷管理方式 (CPU, メモリ, エ/0回数を考慮した負荷管理)
- 今回報告する解析モデルには、上記方式中の(ii)~(vi)を要因として取り入れている。(i)~(vi)については今後の検討課題である。

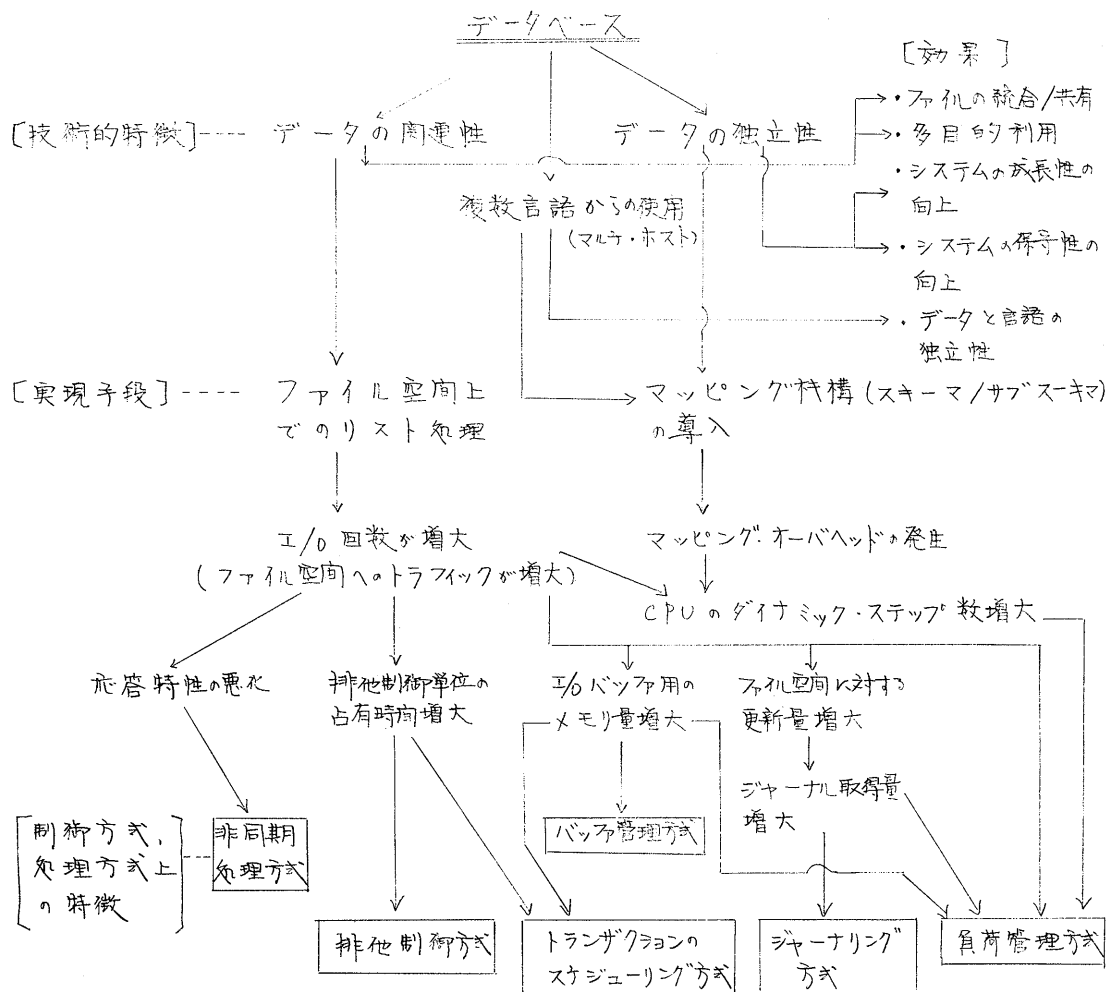


図 2. 1 データベース・システムの特徴分析

3. 性能解析

データベース・システムの性能を次の手順に従って解析を行なう。

- (1) オ1段階; 解析モデルの作成
- (2) オ2段階; 実効リソース使用時間によるスループットの算出
- (3) オ3段階; 応答時間の計算×スループット→応答時間曲線の作成

3. 1 トランザクションーリソース・モデル

トランザクションーリソース・モデルの概念は、図 3. 1 に示す通りである。環境（外部システム）よりシステム中へインプットされるトランザクション（例；メッセージ、ジョブ）は、システム中の各種リソースを使用したかゝる処理を行ない、環境にアウトプットされる。この時、各種リソースの利用率 ρ_i は次式で示される。

$$\rho_i = \frac{\sum \lambda_i \cdot T_i}{n \cdot N_i} \quad \text{----- (3. 1)}$$

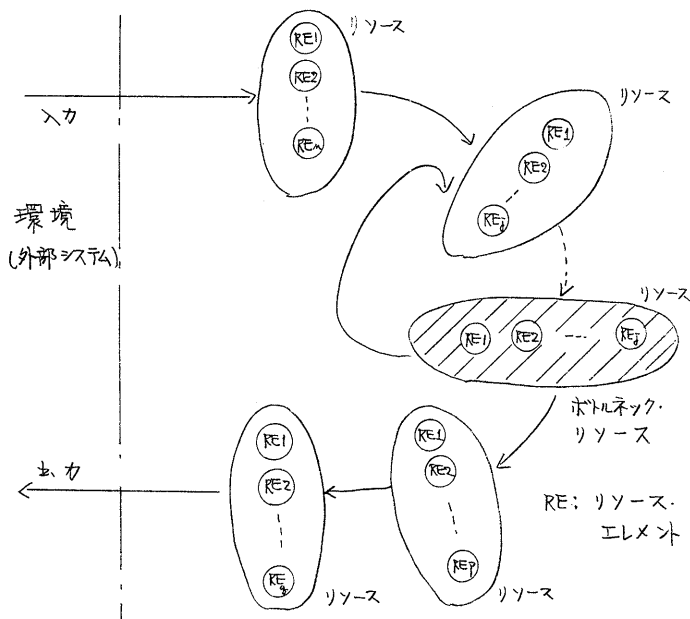


図 3. 1 トランザクション-リソース・モデルの概観

＜記号の意味＞

ρ_i : リソース i の利用率 (i はリソースを表わす)

S_i : スループット。
単位時間当りに、リソース i において処理されるトランザクションの量。平衡状態においては、すべての S_i は等しい。

T_i : 1 個のトランザクションがリソース i を使用する平均時間 (平均使用時間)。

ϕ : 単位時間。

N_i : リソース i におけるエレメント数 (リソース・エレメント数)。

このリソースの中で、利用率の最も高いリソースをボトルネック・リソース (BNR と略記) と呼ぶ。ここで N_i 個のリソース・エレメントが完全に独立に動作できる場合は問題ないが、動作できない場合には何らかの補正が必要になってくる。補正後の N_i を N_i^* であらわし、有効リソース・エレメント数と呼ぶ。 N_i^* を使えば、 ρ_i は

$$\rho_i = \frac{S_i \cdot T_i}{\phi \cdot N_i^*} \quad \text{---- (3.2)}$$

となる。これより

$$S_i = \frac{\rho_i \cdot \phi}{\left(\frac{T_i}{N_i^*} \right)} \quad \text{---- (3.3)}$$

となる。 $\rho_i = 1$ の時のスループットを最大スループットと呼ぶと、最大スループットは $\left(\frac{T_i}{N_i^*} \right)$ によって決まる。この $\left(\frac{T_i}{N_i^*} \right)$ を有効リソース使用時間と呼ぶ。 N_i から N_i^* を算出する方法はリソースの種類、使われ方によって異なるので、リソース毎に考える必要がある。例えば、マルチ・プロセッサの場合に使われているマルチプロセッサ数は、CPU が n 台の場合の N_i^* に相当すると考えられる。

3.2 解析モデルの作成

データベース・システムのアーキテクチャを決定するうえでの最も大きな外部要因は、スループットと応答時間の関係であると言える。そこでモデルとして、これまでいくつか開発されているデータベース・システムのアーキテクチャから次の二つの代表的タイプを挙げて、スループット、応答時間の関係を解析する。

- ・非スケジューリング・モデル
- ・スケジューリング・モデル

この二つのモデルを改良するために共通する基本的前提条件は次の通りである。
 <前提条件>

- ① システム資源中で、ボトルネックになる可能性のある資源を、タスク、CPU、排他制御単位、ジャーナル、バッファ（データベース用入出力バッファ）磁気ディスク・パックとする。
- ② その他の資源においては、待ち時間は無視できる程小さいものとする。
- ③ 1トランザクションの処理に関して、READ、又はWRITE操作とCPUの並行処理は行わないとする（同期処理）。

3.2.1 非スケジューリング・モデル

非スケジューリング・モデルはデータベース性能とデータ通信性能とをいわば直列に結合して、実時間の処理形態でデータベースに対する処理を行なおうとするものである。本モデルの特徴は、後述するスケジューリング・モデルで行なっているトランザクション・スケジューリングを行わないことである。本モデルの構成を図3.2に示す。

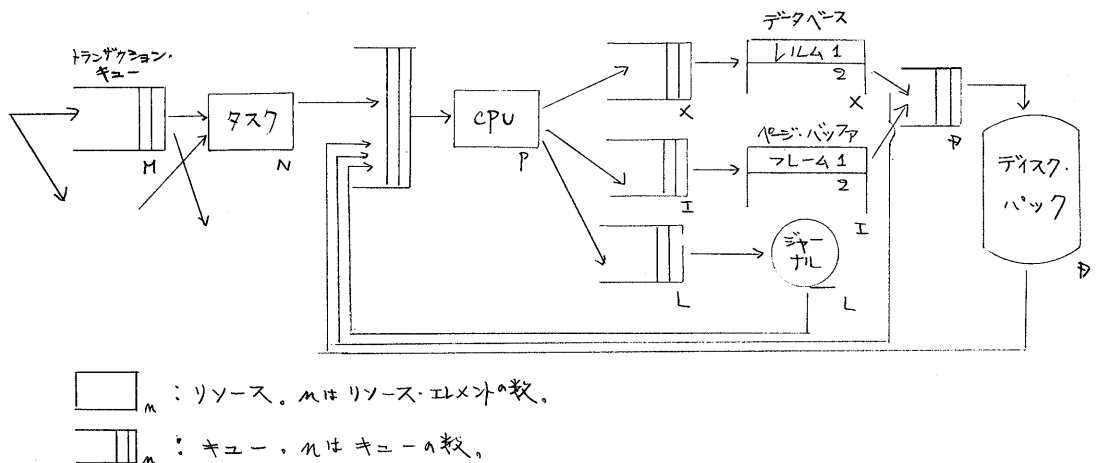


図3.2 非スケジューリング・モデル

3.2.2 スケジューリング・モデル

非スケジューリング・モデルでは、トランザクションがタスクをつかむことと、そのタスクが使用するレダムとの関係は、全く無関係である。従って、タスクは一般には処理の途中でレダム待ちの状態に遭遇する（レダム・レベルの排他制御時）。とくに、タスクのレダム使用時間が長く、かつレダム待ちの状態に遭遇する確率がきわめて大きくなるような状況下においては、トランザクションをタスクにFIFOで渡す単純な非スケジューリング方式では、タスク・ネックを発生する可能性がある。このため、このネックを解消する手段としてトランザクション・スケジューリングを行なう方式がある。すなわち、そのトランザクションの所要とするレダムが空き状態にあるものから、優先的にタスクでの処理を始めるようにスケジューリングする方式である。本モデルでは、トランザクションをトランザクション・キューに接続すると同時に、レダム・キューにも接続してしまおうとするものである。この方式では、タスクが処理の途中でレダム待ちになる

ことはないから、タスクにおける処理効率は一様に向上すると考えられる。しかし、タスクでの処理が開始され、データベースへアクセスしてみないと、一般にはどのページが使用されるかわからないので、トランザクション・スケジューリングでページ・レベルの排他制御を行なうことは不可能である。この結果、非スケジューリング・モデルに比して、常に有利な方法とはいえない。

3.3 スループット、応答時間の算出

各モデルにおけるスループット、応答時間を算出する手順は以下の通りである。

＜算出手順＞

① 各リソース毎の T_i の計算

② 各リソース毎の λ_i の計算

③ 有効リソース使用時間 (T/λ) 表の作成。

T/λ の最大なりソースが、ボトルネック・リソース (BNR) である。もしその BNR の ρ_i の上限値 (限界利用率) が応答時間とは無関係に定められている場合には、(3.3) 式より直ちに ρ_i を計算することができる。これが同時にシステム全体のスループットとなる。

④ スループット-応答時間曲線 (S-R 曲線) の作成。

BNR の限界利用率が定められていない場合は、 T/λ 表を利用して、S-R 曲線を描く。この曲線とシステムに要求されている平均応答時間との関係から、システム全体のスループットが定まる。この様子を図 3.3 に示す。

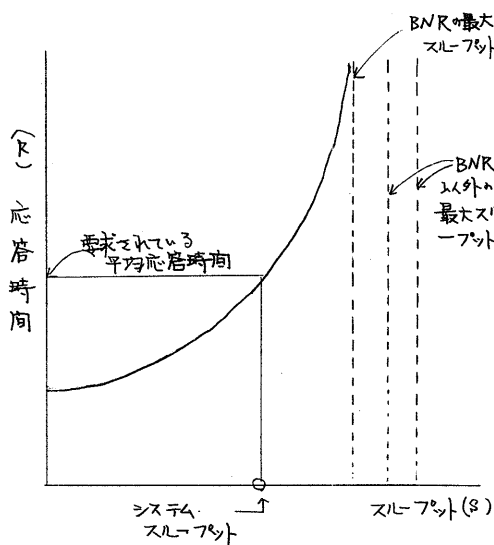


図 3.3 S-R 曲線による
スループットの推定

3.3.1 非スケジューリング・モデルのスループットと応答時間
上記の算出手順に従って、非スケジューリング・モデルのスループットを算出する。

(1) T_i の計算

本モデルにおけるリソースとしては、CPU、レルム、ページ、ジャーナル、磁気ディスク・バック、タスクがある。それぞれの T_i は以下のようなになる。

(i) CPU

1 トランザクションは CPU を時分割的に複数回使用するが、その使用時間の総和の平均値が $T_{c,p}$ である。また待ち時間の総和の平均値を $W_{c,p}$ とする。これらの値は排他制御のレベルによって異なる。ページ・レベルの排他制御のとき、

$T_{c,p}$, $W_{c,p}$, レルム・レベルの排他制御のとき $T_{c,R}$, $W_{c,R}$ と表わす。

(ii) ジャーナル

1 トランザクションは、一般に複数回ジャーナルを WRIT するが、その

使用時間の総和の平均値が T_f である。また待ち時間の総和の平均値を $\bar{W}_f$ とする。

(iii) ディスク・バック

1 トランザクションは、一般に複数回の READ、又は WRITE 操作をディスク・バックに対して行なう。この時の使用時間の総和の平均値が T_b である。また待ち時間の総和の平均値を $\bar{W}_b$ とする。

(iv) レルム

1 トランザクションは、一般に複数のレルムを同時に占有する、又処理の途中で、必要に応じて新しいレルムを占有したり、不要になったレルムの占有を解除したりする。1 トランザクションが 1 レルムを占有する時間の総和の平均値が T_R である。また待ち時間の総和の平均値を $\bar{W}_R$ とする。これはレルム・レベルの排他制御時のみ意味を持つ。

(v) ページ

1 トランザクションが 1 ページを占有する時間の総和の平均値が T_p である。また待ち時間の総和の平均値を $\bar{W}_p$ とする。

(vi) タスク

1 トランザクションが、1 個のタスクを使用する時間は、排他制御のレベルによって異なる。1 トランザクションのタスク使用時間の平均値 T_k は次式により与えられる。

$$T_k = H_c + H_J + H_b + \left\{ \frac{\bar{W}_k}{\bar{W}_p} \right\}$$

$\left\{ \frac{\bar{W}_k}{\bar{W}_p} \right\}$: レルム・レベル排他制御時には $\bar{W}_R$, ページ・レベル排他制御時には $\bar{W}_p$.

$H_c = T_c + \bar{W}_c$ ----- 1 トランザクションの CPU 保留時間。

保留時間 = 使用時間 + 待ち時間

$$H_J = T_J + \bar{W}_J$$

$$H_b = T_b + \bar{W}_b$$

CPU 使用時間およびタスク使用時間が排他制御レベルの影響を受ける。

(2) N_k^* の計算

各リソース毎の有効リソース・エレメント数 N_k^* は以下になる。

(i) CPU

$$N_c^* = (1 - l_c) \cdot P \quad \text{ここで } l_c: \text{CPU の競合係数, } P: \text{CPU の台数}$$

(ii) ジャーナル

$$N_J^* = d_J \cdot (1 - l_J) \cdot L$$

ここで d_J : ジャーナルのからり係数, l_J : ジャーナルの競合係数
 L : JNL の台数

(iii) ディスク・バック

$$N_b^* = d_b \cdot (1 - l_b) \cdot T$$

ここで d_b : ディスク・バックのからり係数, l_b : ディスク・バックの競合係数, T : ディスク・バックの台数

(iv) レルム

$$N_R^* = d_R \cdot K / m_R$$

ここで d_R : レルムのからり係数, m_R : 1 トランザクションが占有するレルムの平均数 (マルチ・リンクエスト係数), K : レルムの数

(v) ページ

$$N_p^* = d_p \cdot X \cdot K / m_p$$

ここで d_p : ページのアクセス係数, X : レルムのページ数, K : レルムの数, m_p : トランザクションが占有するページの平均数(マルチリクエスト係数)

(vi) タスク

$$N_T^* = \begin{cases} N & (\text{if } M \cdot d_{Ta} > N) \\ M \cdot d_{Ta} & (\text{if } M \cdot d_{Ta} \leq N) \end{cases}$$

ここで N : タスクの数(多重度), M : トランザクション・キューの数, d_{Ta} : トランザクション・キューのアクセス係数

(3) 実効リソース使用時間(T/N^*)のテーブル化

上で求めた T_i , N_i^* より T/N^* 表を作成すれば、表 3.1 のようになる。

表中で、 T_i/N_i^* が最大のリソースが、モデル中のボトルネック・リソース(BN R)となる。もし BN R が CPU, ジャーナル, ディスク・パックのうちのいずれかであることが仮定できる場合には、

$$S_{BNR} = \frac{\rho_{BNR}}{T_{BNR}/N_{BNR}^*} \quad (5)$$

直ちにシステムのスループットが求められる、但しこの時 ρ_{BNR} は、BN R の限界利用率である。一方上記の仮

定が成立しない場合には、レルム、ページ、タスクの T_i を求める必要がある。この時には、CPU, ジャーナル, ディスク・パックの滞留時間(=使用時間+待ち時間)を求める必要がある。これについては、次項(4)参照。

(4) スループット-応答時間($S-R$)曲線の作成

BN R の T_i が他リソースでの待ち時間に関連しているり、又は BN R の限界利用率を求められないような場合には、前項までの手順ではスループットは求まらない。このような場合には、次のように応答時間との関連で、パフォーマンスが規定されるのが妥当であろう。

「単位時間当りのスループットは A 件で、その時の平均応答時間は B 秒以内であること」。または、「平均応答時間 B 秒で、その時のスループットは単位時間当り A 件以上であること」。このような要求に応えるためには、スループットと応答時間の関係が把握される必要がある。本モデルにおける応答時間を、トランザクションがトランザクション・キューに到着してから、タスクによる処理が終了するまでと定義すると、応答時間 R は次式で表わされる。

$$R = W_T + T_T$$

トランザクション・キューに到着するトランザクションの到着間隔、および各リソースにおける使用時間については、実際のシステムに対応した確率分布を設定

表 3.1 非スケジューリングモデルの実効リソース使用時間

リソース	T_i	N_i^*	T_i/N_i^*
CPU	T_c	$(1-l_c) \cdot P$	$T_c/(1-l_c) \cdot P$
ジャーナル	T_j	$d_j \cdot (1-l_j) \cdot L$	$T_j/d_j(1-l_j) \cdot L$
ディスク・パック	T_d	$d_d \cdot (1-l_d) \cdot T$	$T_d/d_d(1-l_d) \cdot T$
レルム	$T_R = H_{c,R} + H_{j,R} + H_{d,R}$	$d_R \cdot K / m_R$	$T_R/(d_R \cdot K / m_R)$
ページ	$T_p = H_{c,p} + H_{j,p} + H_{d,p}$	$d_p \cdot K \cdot X / m_p$	$T_p/d_p \cdot X \cdot (K/m_p)$
タスク	レルム・レベル 排他制御	$N : \text{if } M \cdot d_{Ta} > N$	T_T/N_T^*
	ページ・レベル 排他制御	$M \cdot d_{Ta} : \text{if } M \cdot d_{Ta} \leq N$	

することが必要であるが、この場合かなり複雑な分布となり、計算のし易いものにならない。そこで大体の目安をつけることを目的として近似的な分布を考えてみる。即ち、トランザクション・キューに到着するトランザクションの到着間隔が指数分布、すべてのリソースにおける使用時間が指数分布であると仮定する。また複数のリソース・エレメントが存在するタスク、CPU等においてもポイソン分布として、 T/N^* を T とおき、単一サーバとみなして待ち時間を計算する。つまり全リソースに対して $M/M/1$ を適用する。以上の仮定のもと、 R は以下のようになる。

$$R = \frac{\rho_T}{1-\rho_T} \cdot \frac{T_T}{N_T^*} + T_T = \left\{ \frac{\rho_T}{(1-\rho_T)} \cdot \frac{1}{N_T^*} + 1 \right\} T_T = F_T \cdot T_T \quad \text{--- (3.4)}$$

但し $F_T = \frac{\rho_T}{1-\rho_T} \cdot \frac{1}{N_T^*} + 1$, 表3.1の結果をこれに代入すると、

(i) レルム・レベル排他制御時

$$R = F_T \cdot T_T = F_T \cdot (H_C + H_J + H_D + W_R)$$

$$\text{一方 } W_R = \frac{\rho_R}{1-\rho_R} \cdot T_R = \frac{\rho_R}{1-\rho_R} (H_{C,R} + H_{D,R} + H_{J,R}) \quad \text{--- (3.5)}$$

$$\text{であるから } R = F_T \cdot \left\{ H_C + H_J + H_D + \frac{\rho_R}{1-\rho_R} (H_{C,R} + H_{D,R} + H_{J,R}) \right\} \quad \text{--- (3.6)}$$

$$\text{また } H_C = \left\{ \frac{\rho_C}{1-\rho_C} \cdot \frac{1}{N_C^*} + 1 \right\} T_C = F_C \cdot T_C, \quad H_J = \frac{1}{1-\rho_J} \cdot T_J$$

$$H_D = \frac{1}{1-\rho_D} \cdot T_D, \quad H_{C,R} = F_C \cdot T_{C,R}, \quad H_{J,R} = \frac{1}{1-\rho_J} \cdot T_{J,R}, \quad H_{D,R} = \frac{1}{1-\rho_D} \cdot T_{D,R}$$

であるから、これを(3.6)式に代入して、

$$R = F_T \cdot \left\{ F_C \cdot T_C + \frac{1}{1-\rho_J} \cdot T_J + \frac{1}{1-\rho_D} \cdot T_D + \frac{\rho_R}{1-\rho_R} \left(F_C \cdot T_{C,R} + \frac{1}{1-\rho_D} \cdot T_{D,R} + \frac{1}{1-\rho_J} \cdot T_{J,R} \right) \right\} \quad \text{--- (3.7)}$$

(3.7)式は、 $\rho_i, N_i^*, T_i, T_{i,R}$ の関数であるから、

$$R = f(\rho_i, N_i^*, T_i, T_{i,R}) \quad \text{一方 } \rho_i = \frac{\beta \cdot T_i}{N_i^* \cdot \phi} \text{ より } R = f(\beta, N_i^*, T_i, T_{i,R})$$

CPU, ジャーナル, ディスク・バックの T_i はスループット(β)に無関係に定まるが、タスク, レルムの T_i は β の関数であるから、

$R = f(\beta, N_i^*, T_C, T_D, T_J, T_{C,R}, T_{D,R}, T_{J,R})$ $N_i^*, T_C, T_D, T_J, T_{C,R}, T_{D,R}, T_{J,R}$ を固定(定数)と考えるとよい時は、 R は β のみの関数、すなわち $R = f(\beta)$ となる。

(ii) ページ・レベル排他制御時

レルム・レベルの排他制御時と同様に

$$R = F_T \cdot \left\{ F_C \cdot T_C + \frac{1}{1-\rho_J} \cdot T_J + \frac{1}{1-\rho_D} \cdot T_D + \frac{\rho_R}{1-\rho_R} \left(F_C \cdot T_{C,P} + \frac{1}{1-\rho_D} \cdot T_{D,P} + \frac{1}{1-\rho_J} \cdot T_{J,P} \right) \right\} \quad \text{--- (3.8)}$$

(3.7)式と(3.8)式の F_c, T_c は一般にその値が異なることに注意する必要がある。この場合も同様に $N_i^*, T_c, T_D, T_J, T_{c,p}, T_{D,p}, T_{J,p}$ を恒長(変数)と考えるとよい時には、 $R=f(\rho)$ となる。式(3.7), (3.8)よりスループット-応答時間曲線(ρ - R 曲線)を描くことができる。曲線の形がどのようなになるかは、各パラメータの値に依存するが、その一例を図3.4に示す。また仮にA, Bが図に示すような位置にあるとすれば、このようなサービス・システムにおいてはレベル・レベルの排他制御の方が有利であると言える。

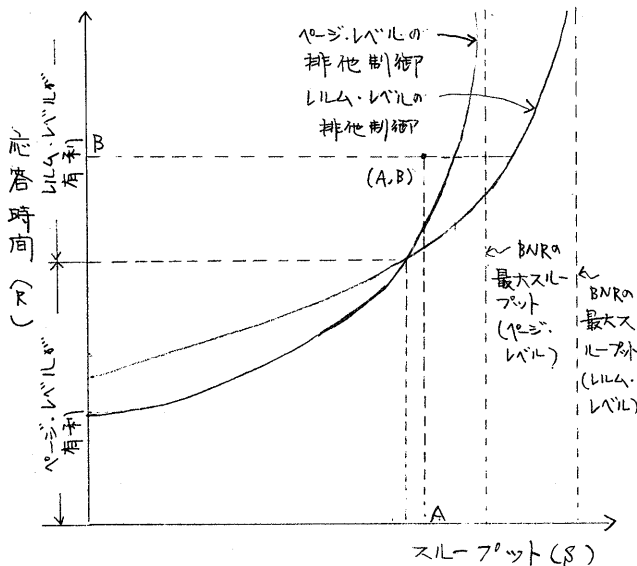


図3.4 ρ - R 曲線の一例

3.3.2 スケジューリング・モデルのスループットと応答時間

基本的な解析手法は前項と同じであるから、非スケジューリング・モデルとの相違点を中心に述べるにとどめる。スケジューリング・モデルでは、タスクの処理中にトランザクションがレベル待ちの状態に遭遇することはないから、 $T_T = H_c + H_J + H_D$ --- (3.9)
また本モデルでは、トランザクション・キューでタスクとレベルを待っていることとなるから、トランザクション・キューにおける滞留時間を T_{Ta}

とすると $T_{Ta} = W_T + W_R$ --- (3.10)

で近似的に表わされる。 N_i^* は非スケジューリング・モデルの場合と全く同じである。実効リソース使用時間を表3.2に示す。本モデルにおける応答時間は、

$$R = T_{Ta} + T_T \quad \text{--- (3.11)}$$

(3.10)式より

$$R = W_T + W_R + T_T = F_T \cdot T_T + W_R \quad \text{--- (3.12)}$$

一方

$$\left. \begin{aligned} H_c &= F_c \cdot T_c \\ H_J &= \frac{1}{1-\rho_J} \cdot T_J \\ H_D &= \frac{1}{1-\rho_D} \cdot T_D \end{aligned} \right\} \quad \text{--- (3.13)}$$

表3.2 スケジューリング・モデルの実効リソース使用時間

リソース	T_i	N_i^*	T_i / N_i^*
CPU	T_c	$(1-l_c) \cdot P$	$T_c / (1-l_c) \cdot P$
ジョナル	T_J	$d_J \cdot (1-l_J) \cdot L$	$T_J / d_J (1-l_J) \cdot L$
ディスク	T_D	$d_D \cdot (1-l_D) \cdot T$	$T_D / d_D (1-l_D) \cdot T$
レベル	$T_R = H_{c,R} + H_{D,R} + H_{J,R}$	$d_R \cdot K / m_R$	$T_R / (d_R \cdot K / m_R)$
タスク	$T_T = H_c + H_J + H_D$	$N: \text{if } N \cdot d_{Ta} > N$ $N \cdot d_{Ta} : \text{if } N \cdot d_{Ta} \leq N$	$\frac{T_T}{N_T^*}$

$$W_R = \frac{\rho_R}{1-\rho_R} \cdot T_R = \frac{\rho_R}{1-\rho_R} (H_{c,R} + H_{D,R} + H_{J,R}) = \frac{\rho_R}{1-\rho_R} (F_c \cdot T_{c,R} + \frac{1}{1-\rho_D} \cdot T_{D,R} + \frac{1}{1-\rho_J} \cdot T_{J,R}) \quad \text{--- (3.14)}$$

(3.9), (3.13), (3.14) を (3.12) に代入して,

$$R = F_T \left(F_c \cdot T_c + \frac{1}{1-\rho_j} \cdot T_j + \frac{1}{1-\rho_d} \cdot T_d \right) + \frac{\rho_R}{1-\rho_R} \left(F_c \cdot T_{c,R} + \frac{1}{1-\rho_d} \cdot T_{d,R} + \frac{1}{1-\rho_j} \cdot T_{j,R} \right) \quad \text{--- (3.14)}$$

4. モデルの活用

前章までは, 2つのデータベース・システム モデルを作り, それを実効リソース使用時間, S-R 曲線等を使って性能の解析を行なった。この結果は, 以下のような向題を差える際に有効なツールとなろう。

- (1) レルム・レベルの排他制御とページ・レベルの排他制御は, パフォーマンス上どちらが有利か。もしこの有利さが左右される条件があれば, それは何か。
- (2) レルム・レベルの排他制御時には, トランザクション・スケジューリングを行なう方式と, 行なわない方式が考えられる。パフォーマンス上どちらが有利か。もしこの有利さが左右される条件があれば, それは何か。
- (3) タスクの多重度, CPU の台数 (マルチ・プロセッサ, ポリプロセッサ) / 速度, ジャーナルの台数 / 性能, レルムの数, バッファの面数 (ページ・フレーム数), ページ・サイズ等はスループット, 応答時間とどのような影響を与えるか。

5. 今後の課題

今後に残された課題としては, 以下のようなものがある。

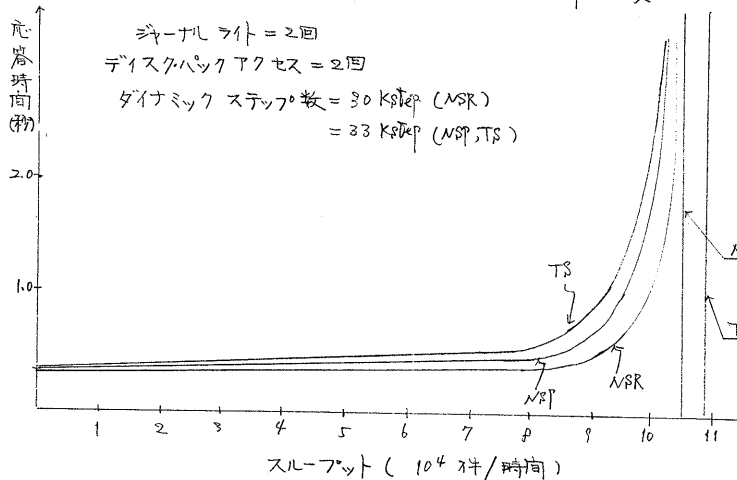
- (1) バッファ管理方式のモデルへの取り込みと, スループット, 応答時間と与える影響の解析。またオンデマンド・ページングとの関連。
- (2) 非同期処理方式のモデルへの取り込みと, スループット, 応答時間と与える影響の解析。
- (3) オンライン・トランザクションとバッチ・ジョブのように一般的に処理特性の異なるものが同一データベースへコンカレントにアクセスする時のスループット, 応答時間の関係。またこの時に有効な負荷管理方式。
- (4) TSS 環境下におけるデータベース・システムの性能解析。
- (5) 超大容量記憶装置 (MSS) が導入された時のステージング / デステージングの向題と性能と与える影響の解析。

6. 参考文献

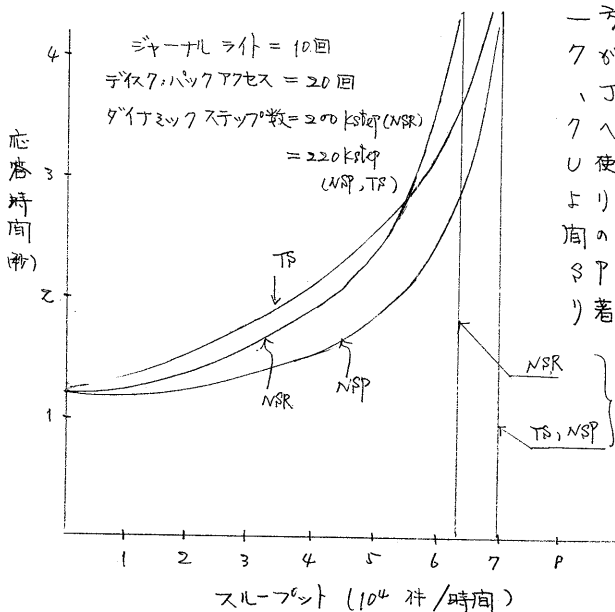
- (1) J.F.C. Kingman; The single Server Queue in Heavy Traffic, Proc. Camb. Phil. Soc 57 (1961), pp 902~904
- (2) Buzen, J.; Analysis of system bottlenecks using a queuing network model, ACM Workshop on System Performance Evaluation.
- (3) 森 雅夫; 待ち行列論の最近の動向, 経営科学, Vol 15, No 4, 1971年11月
- (4) 平塚, 林, 河津; オンラインシステムにおけるリエントラントエリアの最適構成について, 情報処理学会 昭和46年度第12回大会予稿集, 1971, pp 183~184
- (5) 河津; 処理能力測定報告書, データ通信本部内部資料, 1972

<付属資料 — 解析例>

電気会社が関与している、あるいは関与することを計画しているシステムの中から代表的な業務タイプとしてバンキング・タイプ（以下タイプAと呼ぶ）と登録業務タイプ（タイプB）を取り上げる。本文におけるスケジューリング・モデル（以下TSと略す）と、非スケジューリング・モデルとを使った解析結果を示すと付図1、付図2のようになる。非スケジューリングの場合には、S-R曲線を排他制御単位がレيلم（NSRと略記）、およびページ（NSPと略記）の2つのケースについて求めてある。図中に、特性パラメータの中の主なものを付記しておいた。各々についてスループットが一矢の時の応答時間を比較すると、タイプAではNSRが、タイプBではNSPが優れていることが分かる。タイプA



付図1. タイプAのS-R曲線



付図2. タイプBのS-R曲線

を定量的に把握し、その上で最適な方式を選択する配慮が必要となってこよう。

におけるBNRは、TS、NSPの時はCPUであり、NSRの時はタスクである。本タイプでは、CPU使用時間(T_c)の増加が応答時間に与える影響（悪化）の方がレيلم待ち時間の減少、タスク待ち時間の減少よりも大きいために、TS、NSPの応答特性の方がNSRの応答特性よりも悪化していると考えられる。

一方タイプBでは、いずれの場合もタスクがBNRになっている。本タイプでは、TSへの書き込み、ディスク・バックへのアクセス回数が多いために、CPU使用時間の増加による応答特性の悪化よりも、レيلم待ち時間、タスク待ち時間の減少の効果の方が大きいために、NSPの応答特性がNSR、TSのそれより著しくよくなっていることが分かる。

これらの例からも明かなように、各種の方式の有利さは、対象とするシステム環境と密接に関連している。データベース管理システムのインプリメンタ、あるいはデータベース・システムの設計者は、この事実を充分に認識し、システム環境を構成する各種リソースの性能、数量等を