順次アクセス入力処理におけるディスク・キャッシュ装置の効果解析†

西垣 通†† 山本 彰††

ディスク・キャッシュの効果解析モデルを提案する．ディスク・キャッシュとは主メモリとディスク間の緩衝メモリ装置であり，その目的はデータ・アクセス時間短縮によるシステム性能の向上にある．本モデルは，順次アクセス入力処理を対象とし，CPU やチャネルの処理速度，ディスクのシーク，サーチ時間，ジョブ多重度，ディスク・キャッシュ容量，ディスク・キャッシュへの先読み単位などをパラメータとして，システムの処理能力を与える．本モデルの特徴は，待行列理論を用いてデータ・アクセス時間の短縮が処理能力に与える影響を解析する点にある．具体的な数値例について，ディスク・キャッシュ導入による処理能力向上効果，最適なデータ先読み単位などを検討する．

1. まえがき

近年の集積回路技術の進歩は，主メモリと補助メモリのアクセス速度の間に顕著な差異をもたらした．補助メモリ媒体として多用されるディスクへのアクセスは機械的動作をともなうため，主メモリに比較して高速化が困難である．したがってディスク入出力がボトルネックとなりシステム性能が低下するおそれがある．

ひとつの解決策としてディスク・キャッシュの導入があげられる．ディスク・キャッシュとは，主メモリとディスクの中間に在って両者間のデータ転送のバッファリングを行うメモリ装置[4),12)]である．通常，容量，アクセス速度ともに両者の中間に位置し，アクセスは電気的に行われる．ディスク・キャッシュの目的は実効的なディスク入出力実行時間の短縮によるシステム処理能力の向上にある．しかし，その効果はディスク上のデータの参照特性にはなはだしく依存する．参照データを予測可能な場合には顕著な効果が期待できるが，データ参照特性がランダムでこれを予測不能な場合には効果はない．したがってディスク・キャッシュの効果解析においては，データ参照特性をもあわせて評価する必要がある．

従来，ディスク・キャッシュの効果解析に関していくつかの報告がある．Smith[9)]，菅[11)]，宮地[6)]，根岸[7)]らはディスク・アクセスの実測トレース結果をシミュレートし，データ・アクセス時間の短縮効果を示した．この方法は個別システムにおける効果解析には有効であるが，結果自体は，データ参照特性に依存するため一般的なものとはいいがたい．一般的な解析のためにはアプリケーション形態を限定する必要がある．いま，順次アクセス処理に着目すると，データ参照特性が定型的なので一般的解析が可能である．また多くのシステムでこの種の処理が負荷の大半を占めるので，解析は実用的意味がある．Smith[10)]，Welch[13)]は，順次アクセス処理のもとでのディスク・キャッシュの効果について報告し，ディスク・キャッシュ容量やバッファリング単位の最適化などについて述べている．しかし，これらを含め従来の報告の共通点は，ヒット率（目的のデータがディスク・キャッシュ内に存在する確率）の算定をもとにデータ・アクセス時間のみにより評価を実行していることである．実際には，処理能力は CPU 処理速度，ジョブ多重度等の諸要因によっても影響されるので，データ・アクセス時間の短縮がただちに処理能力向上をもたらすとは限らない．

本論文では，ジョブ多重度，CPU 処理速度，チャネル処理速度，ディスク特性値（シーク時間，回転時間），ディスク・キャッシュ容量，ディスク・キャッシュへのデータ先読み（バッファリング）単位などをパラメータとして，システムの処理能力を算定するモデルを提案する．アプリケーションは順次アクセス入力処理を想定する（電源事故時の情報保護のため，出力処理は直接ディスクに行う場合が多いので，ここでは入力処理のみに着目する）．待行列ネットワーク・モデルによりデータ入力要求の発生時間間隔を算定し，システム処理能力を評価する点が本モデルの特長である．まず2章で解析の基本的前提条件を述べ，3章でモデルを提示する．さらに4章では数値例によりディスク・キャッシュの効果を例示する．

† Performance Analysis of Disk Cache Systems for Sequential Data Access by TOHRU NISHIGAKI and AKIRA YAMAMOTO (Systems Development Laboratory, Hitachi, Ltd.).

†† (株)日立製作所システム開発研究所

2. 解析の前提条件

はじめにディスク・キャッシュを有するシステムの動作概要を述べ，ついでそのモデルを導入するための基本的な前提条件および用語の定義をまとめる．

システムのハードウェア構成は**図1**に示したとおりであり，一般に複数のディスクからなるディスク・システムと主メモリの間にディスク・キャッシュが介在する．主メモリのなかには複数のジョブが存在し，これらが多重プログラミングで交互にCPUを割り当てられて実行される．ジョブが順次アクセス入力処理を実行し，ディスク・キャッシュが先読みを行う場合，ディスク上の情報はジョブの入力命令発行を待たず次々とディスク・キャッシュ内のバッファに読みこまれる．先読みは各ジョブについて順番に行われる．

あるジョブがディスクに入力命令を発行したとき，目的の情報がすでにディスク・キャッシュに到着していれば当該情報はただちに主メモリに読みこまれて処理され，その占有していたバッファは開放される．未到着ならば到着するまでジョブは待ち状態となる．CPUの処理速度に比べて先読みの速度が遅い場合，ジョブは頻繁に待ち状態となる．逆の場合にはしばしばバッファが一杯になり先読みを中断せざるをえなくなる．

上記システムをモデル化するための基本的な前提条件ならびに用語を以下にまとめる．システム内には n 個のジョブが多重プログラミングで実行されているとし，n ジョブ中の個々のジョブは区別せず，ジョブ群の平均的特性に着目した単一ジョブ・クラス・モデルとする．ジョブは順次アクセス入力処理，すなわちディスク上に連続して蓄積されたデータを順次読みこんで処理するという動作を継続中とする．ここで「データ」とは1回の入力命令で読める情報単位であり物理的なレコードを意味する．また「処理能力」とは単位時間あたりにディスク・システムから読みこまれ処理されるデータ件数とし，θ で表す．

あるジョブがデータ入力命令を発行してから次のデータの入力命令を発行するまでの平均CPU実行時間を c とする．またディスク・キャッシュ内にあるデータを主メモリに読みこむときの平均チャネル占有時間を e とする．ディスク・キャッシュにはディスク・システムからのデータがジョブの入力命令発行を待たず次々に先読みされてくるとし，先読みは n ジョブのおのおのについて k 個のデータの一括読みこみを順番に実行する．k を「先読み単位」とよぶ．入力命令が発行されたとき，目的のデータがディスク・キャッシュ内にあればただちに主メモリに読みこむ．なければ目的のデータがディスク・キャッシュに到着するまで待ち，到着時点で主メモリに読みこむ．なお，ディスク・システムからディスク・キャッシュへのデータ転送とディスク・キャッシュから主メモリへのデータ転送は並行動作可能と仮定する．

ディスク・キャッシュ容量については，ひとつのジョブあたりのディスク・キャッシュの平均割当て量（平均バッファ・サイズ）を M_0 とする．ディスク・システムからディスク・キャッシュへのデータ先読みは，ディスク・システムの最大転送能力で実行してもジョブあたりの平均ディスク・キャッシュ使用量（平均バッファ使用量）M が M_0 以下の場合は最大転送能力で実行する．これ以外の場合は，M が M_0 に一致する速度でデータ先読みを実行する．

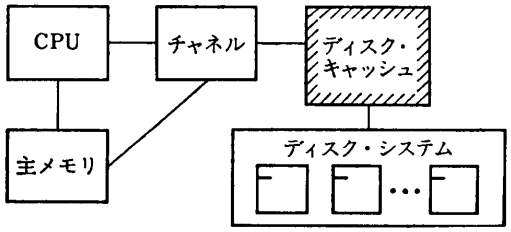

図1 システムのハードウェア構成
Fig. 1 The hardware configuration of the system.

3. 性能解析モデル

3.1 一般モデル

n ジョブ中のあるひとつのジョブに着目する．当ジョブの用いるデータは，図1のディスク・システムからディスク・キャッシュに先読み単位 k 個ずつまとめて先読みされてくる．これらのデータ群はディスク・キャッシュ内のバッファを占有するが，やがて主メモリに読みこまれCPUで当データの処理が完了するとバッファは開放される．したがって，系はデータ(呼)が先読み単位 k 個ずつ集団で到着する窓口数1の集団到着待行列モデルに帰着される．**図2**を参照されたい．ここで，データ集団の到着率を λ，ひとつのデー

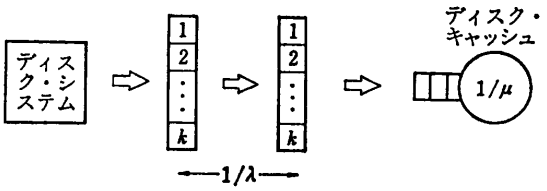

図2 集団到着待行列モデル
Fig. 2 Bulk arrival queueing model.

タの平均サービス時間を $1/\mu$ とする．λ は当該ジョブの用いるデータ集団のディスク・キャッシュへの入力率であり，データ k 個の先読みの平均実行間隔の逆数にほかならない．n と k が与えられたとき，λ にはディスク・システムの物理的特性から定まる最大値 λx が存在する．λx はディスク・システムが最大転送能力でデータをディスク・キャッシュに入力する際の到着率である．また $1/\mu$ はデータ1個の平均処理時間である．いいかえると $1/\mu$ は，ディスク・キャッシュ内にあるデータ1個をチャネル経由で主メモリに読みこみ，CPU でこれを処理し，次のデータの入力命令を発行するまでの平均時間である．後述するように $1/\mu$ は n, k, λ の変数である．

いま，当該ジョブの平均ディスク・キャッシュ使用量 M について考える．M は，この集団到着待行列モデルにおけるデータの平均滞在数と平均データ長の積にほかならない．なお，本モデルの利用率 ρ は，

$$\rho=k\lambda/\mu \tag{1}$$

で与えられる．以下，M と ρ の関係について述べる．データ集団をひとつの呼とみなし，平均到着時間間隔 $1/\lambda$，平均サービス時間 k/μ の待行列モデルをあらためて考える．一般に $G/G/1$ モデルにおいて，呼の平均待ち時間 w_1 は次式で与えられる[5]．ただし σ_a^2, σ_b^2, σ_c^2 はそれぞれ到着時間間隔，サービス時間，遊休時間の分散を表す．

$$w_1=\frac{\lambda(\sigma_a^2+\sigma_b^2)}{2(1-\rho)}-\frac{\lambda\sigma_c^2}{2(1-\rho)} \tag{2}$$

平均待行列長は Little の公式より $w_1\lambda$ となるから，サービスを待っている集団に含まれるデータの平均個数は $kw_1\lambda$ で与えられる．次に，現在サービスを受けつつある集団のなかで，サービスを受けているかサービスを待っているデータの平均個数を求める．このため，サービスの平均残余時間 w_2 に着目する．$G/G/1$ モデルで w_2 は次式で与えられる[5]．

$$w_2=\frac{k}{2\mu}+\frac{\mu\sigma_b^2}{2k} \tag{3}$$

したがって Little の公式より求めるデータ平均個数は $k\lambda w_2$ となる．平均データ長を D とすれば M は，

$$M=k\lambda D(w_1+w_2) \tag{4}$$

で与えられる．ただし式(2)において，σ_c^2 は不明な場合が多く，ρ が1に近いとき第2項を近似的に無視できることが知られている[5]．したがって本論文ではこれを無視する．さらに到着時間間隔，サービス時間の変動係数 $v_a(=\lambda\sigma_a)$, $v_b(=\mu\sigma_b/k)$ を用いて書き直すと，M は結局次式で与えられる．

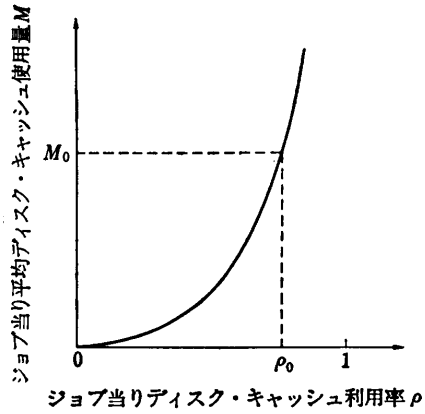

図3　ジョブあたりの，平均ディスク・キャッシュ割当て量 M_0 とディスク・キャッシュ利用率上限値 ρ_0
Fig. 3　The mean disk cache allotment M_0 and the disk cache utilization upper limit ρ_0 per job.

$$M=\frac{D}{2}\left\{\frac{k}{1-\rho}v_a^2+\frac{k\rho}{1-\rho}v_b^2+k\rho\right\} \tag{5}$$

M は ρ とともに増大し，$\rho\to1$ で無限大となる．

いま，ジョブあたりの平均ディスク・キャッシュ割当て量 M_0 に対応する ρ を ρ_0 とする（図3参照）．後述のように n と k が与えられたとき λ を定めると μ は求まるので，ρ, λ, μ のうちひとつを定めれば他の2変数は定まる．いま，ρ_0 に対応する λ, μ をそれぞれ λ_0, μ_0 とかく．また，λ の最大値 λx に対応する ρ, μ をそれぞれ ρx, μx とかく．さらに ρx に対応する M を Mx とする．

次に処理能力 θ を求める．ディスク・システムからディスク・キャッシュへの先読みは，2章で述べたように $M_0\geq Mx$ のとき λx, $M_0<Mx$ のとき λ_0 の転送速度で実行する．ここで一般に M が λ の増加関数であることに注意すれば，処理能力 θ は次式で与えられる．

$$\begin{aligned}\theta&=\min(nk\lambda x,\ nk\lambda_0)\\&=\min(nk\lambda x,\ n\rho_0\mu_0)\end{aligned} \tag{6}$$

式(6)で，$nk\lambda x$ はディスク・システムの物理的特性より定まるが，複数のディスク群が複数のパスで連結されている場合には，回転待ちの後のパス再結合成功確率などの評価が必要となり，一般に算定がむずかしい．とくに単一のディスクのみが使用される場合については次節で検討する．また，式(6)の $n\rho_0\mu_0$ は図4の閉待行列ネットワークを解析することにより求まる．図4において，$1/\mu_0$ はジョブがⒶ点からⒷ点にいたるまでの平均時間である．p はミス率（$=1-$ヒ

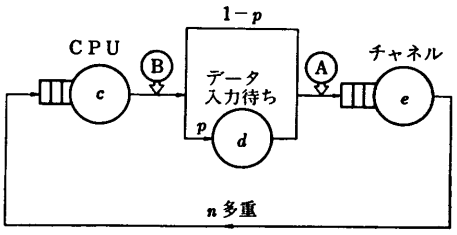

図 4 ディスク・キャッシュの平均使用量 M が平均割当て量 M_0 に等しいときのデータ処理率 $n\rho_0\mu_0$ を求めるモデル
Fig. 4 The system model to evaluate the data processing rate $n\rho_0\mu_0$, where the mean disk cache use amount M is equal to the mean disk cache allotment M_0.

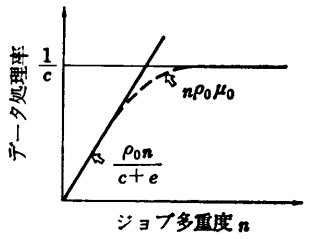

図 5 データ処理率 $n\rho_0\mu_0$ の漸近線
Fig. 5 The asymptotes of the data processing rate $n\rho_0\mu_0$.

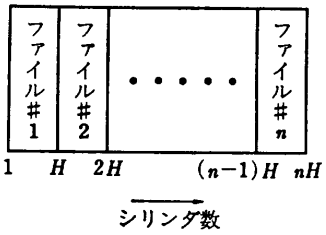

図 6 ディスク上のファイル構成
Fig. 6 The layout of the files on the disk.

ット率．CPU から入力命令が発行されたデータがディスク・キャッシュ内に存在しない確率）であり，d はミスの場合に目的のデータがディスク・キャッシュに読みこまれるまでの平均入力待ち時間である．平均サイクル時間（ジョブが図4のネットワークを一周する平均時間）を T とすると，

$$T=1/(k\lambda_0)=1/(\rho_0\mu_0) \quad (7)$$

であるから，

$$pd=1/(k\lambda_0)-1/\mu_0$$
$$=(1-\rho_0)/(\rho_0\mu_0) \quad (8)$$

が成り立つ．一般に CPU やチャネルでジョブのサービスが先着順に行われるとき，サービス時間が指数分布にしたがわないと求解は困難となる[1]．ただし，近似として $n\rho_0\mu_0$ の漸近解[8]は以下のように簡単に求まる．n が小さいとき，

$$1/\mu_0\cong c+e \quad (9)$$

である．n が増すにつれ CPU またはチャネルがボトルネックになる．両者は対称なので，以下本論文では $c>e$ と仮定する．CPU の利用率は cn/T であるから，n が大のとき

$$cn/T\cong 1 \quad (10)$$

となる．よって式(7)より，このとき

$$n\rho_0\mu_0\cong 1/c \quad (11)$$

となる．これらを図示すると**図5**のようになる（なお，次節で指数サービスの仮定のもとに厳密解の求解法について述べる）．

3.2 単一ディスク・モデル

処理能力 θ の算定には λx, ρ_0, μ_0 を求める必要があるが，前述のようにこれはつねに可能とは限らない．本節では次の三つの仮定のもとにモデルを簡単化し，θ を導出する．

〔仮定1〕 単一ディスク：n 個のジョブの用いるファイルはすべて単一のディスク上に蓄積されており，それぞれ**図6**のようにHシリンダを占める．

〔仮定2〕 BCMP 型待行列ネットワーク[1]：図4で，CPU，チャネルは先着順サービスとし，サービス時間は指数分布にしたがう（または processor sharing サービスとし，一般分布にしたがう）．

〔仮定3〕 $D/E_k/1$ モデル：図2で，データ集団の到着時間間隔，サービス時間はそれぞれ一点分布，k 次のアーラン分布にしたがう．

仮定1のもとでは，後述のようにディスク・ヘッドの動作が規則的なので，到着時間間隔が一点分布にしたがうという仮定3の前半はほぼ成立する．ただし仮定3の後半は，仮定2と厳密には矛盾する．すなわち，仮定3のもとでひとつのデータのサービス時間は指数分布にしたがうが，このとき図4の CPU とチャネルのサービス時間は指数分布にはしたがわない．しかし，この種の待行列ネットワークでは分布形の差異が結果に与える影響は比較的小さいことが知られているので[15]，ここでは解析を容易にするため近似的に仮定2，3が並立するとみなす．

仮定3のもとでは，式(5)において $v_a=0$, $v_b=1/\sqrt{k}$ であるから，ジョブあたりの平均ディスク・キャッシュ使用量 M は，

$$M=(D/2)\{\rho/(1-\rho)+k\rho\} \quad (12)$$

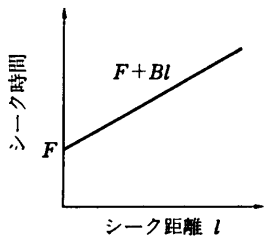

図7 ディスク・シーク時間の一次関数による近似
Fig. 7 A continuous approximation for disk seek time.

となる．M_0 が与えられたとき，式(12)から ρ_0 を求める．ついで次のように式(6)により θ を算定する．まず仮定1のもとで，$nk\lambda x$ は以下のように求まる．図6において，あるファイルからのデータ入力を完了して次に隣りのファイルからデータ入力を行うための平均シーク距離はHシリンダである．ディスク・シーク時間は図7のような一次関数 $F+Bl$（l：シーク距離）で近似できることが知られているので，このときの平均シーク時間は $(F+BH)$ である．n 個のファイルを順次サービスし，再びもとのファイルに戻るまでに $(n-1)$ 回このようなシーク動作が行われる．さらに，ファイル #n からファイル #1 に戻るときの平均シーク時間は $\{F+(n-1)BH\}$ である．したがって n ファイル全部を一巡サービスするときの平均シーク時間は $(n-1)(F+BH)+\{F+(n-1)BH\}$ となる．また，ディスクの回転時間を R，転送速度（≒トラック容量/回転時間）を S とすると，平均サーチ時間はランダム・サーチの仮定のもとで $nR/2$，平均データ転送時間は平均データ長を D として nkD/S で与えられる．以上より，$1/\lambda x$ は，

$$1/\lambda x = n(F+2BH+R/2+kD/S)-2BH \quad (13)$$

となり，結局 $nk\lambda x$ は式(14)で与えられる．

$$nk\lambda x = [D/S+\{F+2BH(1-1/n)+R/2\}/k]^{-1} \quad (14)$$

次に，$n\rho_0\mu_0$ について考える．仮定2のもとで図4はいわゆる BCMP 型の待行列ネットワークとなり，比較的簡単に解が得られることが知られている[1]．とくにこのとき解はネットワーク・ノードへの負荷のみに依存するので，p や d を個別に求める必要はなく，式(8)で求まる両者の積を用いればよい．なお，求解の際，入力パラメータは変数 μ_0 を含むので，次のような繰返し演算が必要となる．

(1) μ_0 の初期値を仮定し，入力パラメータ pd を式(8)で計算する．

(2) 図4の待行列ネットワークを解き μ_0 を求める．

(3) 仮定した μ_0 と求めた μ_0 との差を計算する．差が十分小なら求めた μ_0 を解とする．差が大なら新たな μ_0 を仮定し，pd を求め直し，ステップ(2)に戻る．

この繰返し演算は，ステップ(3)で新たな μ_0 を仮定する際，2分探索法を用いると必ず収束する．理由は次のとおりである．いま，ステップ(2)において求めた関数値が仮定した μ_0 に等しいとき収束解となるが，この関数は仮定した μ_0 の単調減少関数である（これは，μ_0 を増すと式(8)よりデータ入力待ちが減り，図4で CPU やチャネルの待ちが増すことから明らかである）．また，解の存在範囲は既知である（ジョブが図4のⒶ点からⒷ点に至る平均時間は明らかに $(c+e)$ と $n(c+e)$ との間にある）．以上より，2分探索法によって必ず解を見いだすことができる．なお，収束の判定条件としては，求めた μ_0 と仮定した μ_0 との差異が μ_0 の絶対値の 1/1,000 程度のオーダであれば収束解とみなす．上記のようにして，$nk\lambda x$，$n\rho_0\mu_0$ を求めたのち，式(6)より θ を算定できる．

次にデータの先読み単位 k の最適化について考える．ジョブあたりの平均ディスク・キャッシュ使用量 M は式(12)に示したように k とともに増大するので，ρ_0 ならびに $n\rho_0\mu_0$ は k の減少関数である（図3参照）．一方，式(14)の大括弧内の第2項はデータ1個あたりのディスク・アクセス動作のオーバヘッドであり，k を増すとこれが減少するので $nk\lambda x$ は k の増加関数である．したがって式(6)より，処理能力 θ を最大にする k は $n\rho_0\mu_0$ と $nk\lambda x$ が等しくなるような k にほかならない．これを k_{opt} とかく．一般に k_{opt} は n の関数であり解析的に求めることは困難であるが，$n\rho_0\mu_0$ と $nk\lambda x$ の漸近的性質に着目すると以下の議論が成り立つ．いま，すべての n について，

$$n\rho_0\mu_0 \leq 1/c \quad (15)$$

$$nk\lambda x \geq \{D/S+(F+2BH+R/2)/k\}^{-1} \quad (16)$$

である（図5および式(14)参照）．したがって，式(15)，(16)の右辺を等しくする k を k_c とかくと，

$$k_c = \{F+2BH+R/2\}/(c-D/S) \quad (17)$$

である．前述のように $n\rho_0\mu_0$, $nk\lambda x$ はそれぞれ k の減少関数，増加関数なので，一般に $k_{opt} \leq k_c$ である．さらに n を増すにつれ k_{opt} は k_c に近づくので，k_c を k_{opt} の近似値として採用可能である（図8参照）．

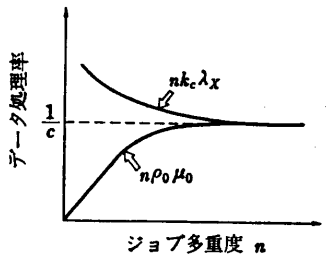

図 8 データ先読み単位 k の最適化
Fig. 8 The optimization of the data prefetching size k.

4. 解 析 例

前章で述べたモデルにより，ディスク・キャッシュ導入時の処理能力 θ を算定できる．本章では数値例により，ディスク・キャッシュ導入前後の処理能力を比較する．なお数値例においては，ディスクは HITAC H-8595 ディスク[3] を想定し，最小シーク時間 $F=7$ ms，シリンダあたりのシーク時間増分値 $B=0.068$ ms，回転時間 $R=16.7$ ms，転送速度 $S=1,198$ k バイト/秒とする．また図6において，ひとつのファイルの占める平均シリンダ数 $H=10$，平均データ長 $D=3$ k バイトとする．また，データ1個を処理するための平均 CPU ステップ数を 60 k ステップ，CPU 処理速度を 6 Mips，チャネルの転送速度を 3 M バイト/秒と想定すると，平均 CPU サービス時間 $c=10$ ms，平均チャネル・サービス時間 $e=1$ ms となる．さらにジョブ多重度 $n=1$〜10 程度とする．

(1) ディスク・キャッシュ導入後の処理能力

前述のように処理能力 θ はジョブあたりの平均ディスク・キャッシュ割当て量 M_0 とディスク・キャッシュへの先読み単位 k により影響される．いま，k として 1, 4, 16 の三つの場合を設定する．ジョブあたりの平均ディスク・キャッシュ使用量 M を式(12)により求め，これを図示すると**図9**のようになる．なお，3章では M_0 が与えられて ρ_0 を求めるという議論を展開したが，ジョブあたりのキャッシュ利用率 ρ_0 があまりにも小さいことは現実的ではない．したがって本例では逆に $\rho_0=0.9$ とし，90% の利用率達成のために必要な M_0 を求めることにする．このとき $n\rho_0\mu_0$ は k に依存しないが，M_0 が k に依存する．

ディスクが最大転送能力（利用率 100%）を発揮したときのデータ処理率 $nk\lambda_X$ を式(14)より求めた結果を**図10**に示す．また，ディスク・キャッシュの平均使用量 M が平均割当て量 M_0 に等しい場合のデータ処理率 $n\rho_0\mu_0$ を 3.2 節で述べた方法で求めた結果を**図11**に示す．式(6)より，処理能力 θ は両者の小さいほうで与えられるから，これを図示すると**図12**のようになる．$k=1$ の場合は θ はディスクの転送能力すなわち $nk\lambda_X$ でおさえられるが，$k=4$, 16 の場合は θ は実質的に CPU の処理能力に一致する．実際，式(17)を計算すると本例では $k_c=2.23$ ($\cong k_{opt}$) とな

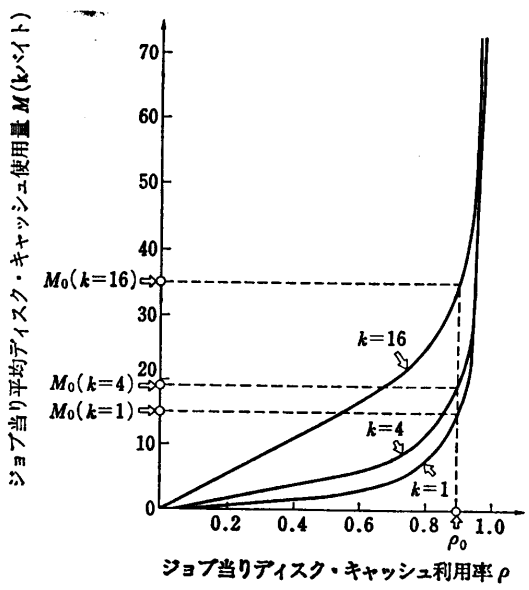

図 9 利用率上限値 $\rho_0=0.9$ のときの，ジョブあたり平均ディスク・キャッシュ割当て量 M_0
Fig. 9 The mean disk cache allotment M_0, where the utilization upper limit $\rho_0=0.9$.

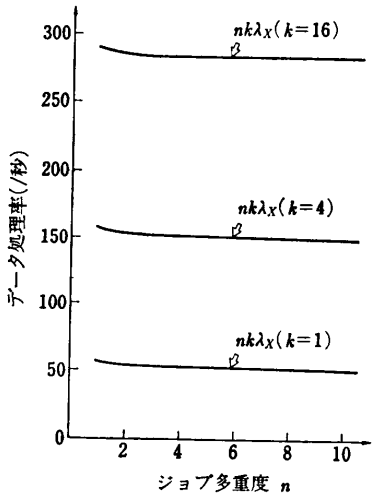

図 10 ディスクの最大データ転送能力に対応するデータ処理率 $nk\lambda_X$
Fig. 10 The data processing rate $nk\lambda_X$ corresponding to the maximum data transfer capability of the disk.

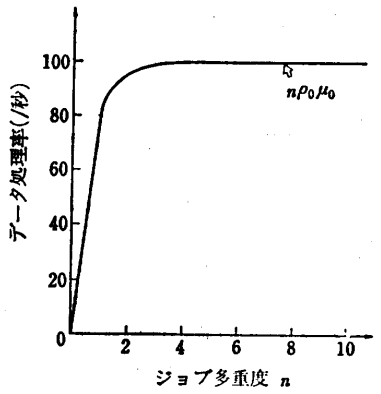

図 11 ディスク・キャッシュの平均使用量 M が平均割当て量 M_0 に等しいときのデータ処理率 $n\rho_0\mu_0$
Fig. 11 The data processing rate $n\rho_0\mu_0$, where the mean disk cache use amount M is equal to the mean disk cache allotment M_0.

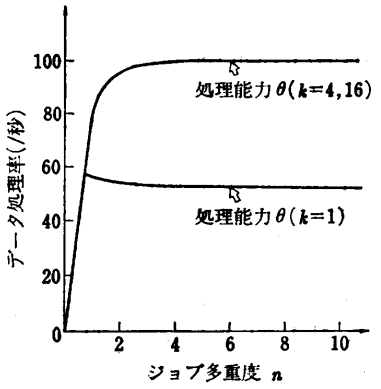

図 12 ディスク・キャッシュ導入後のシステムの処理能力
Fig. 12 The throughput of the system with disk cache.

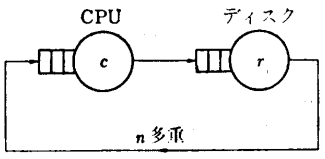

図 13 ディスク・キャッシュ導入以前のシステムの処理能力評価モデル
Fig. 13 The model to evaluate the throughput of the system without disk cache.

る．すなわち k を3より増加すると，θ は不変なのにもかかわらず必要なディスク・キャッシュ容量 M_0 が増加する．したがって $k=1$, 4, 16 のなかでは $k=4$ が最適である．

（2） ディスク・キャッシュ導入以前の処理能力

ディスク・キャッシュがない場合には，データはディスクから直接主メモリに読みこまれて処理が実行される．この場合，モデルは図4において，ディスク・キャッシュへのデータ入力待ちとチャネル待行列とをディスク待行列で置きかえたものになる（**図 13** 参照）．

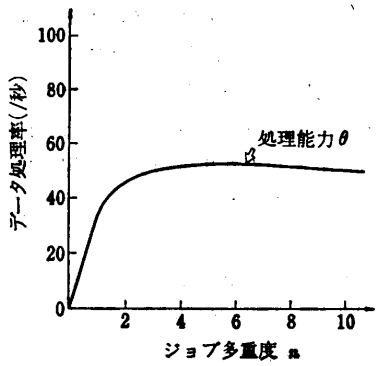

図 14 ディスク・キャッシュ導入以前のシステムの処理能力
Fig. 14 The throughput of the system without disk cache.

ディスクの平均サービス時間を r とすると，これは次のように求まる．図6に示した n 個のファイルはすべてシーケンシャルにアクセスされ，その処理の進度もほぼ等しいとみなせるので，アクセスするシリンダの存在範囲は $(n-1)H$ シリンダである．いま，ディスク・ヘッドの位置およびアクセスするシリンダの位置がともにこの範囲で一様分布にしたがうとすれば，平均シーク距離は $(n-1)H/3$ で与えられる[2]．したがってディスクの平均サービス時間 r は次式で求められる．

$$r=F+B(n-1)H/3+R/2+D/S \qquad (18)$$

ディスクのサービスが先着順に行われ，サービス時間が指数分布にしたがうと近似すると図13は BCMP 型[1]のモデルとなり，求解が可能である．**図 14** にその結果を示す．

ディスク・キャッシュ導入による処理能力向上効果は，図12と図14を比較すると明らかである．$k=1$ の場合は効果は小さいが，$k=4$, 16 の場合には処理能力が約2倍に向上する．これは，ディスク・キャッシュの導入によりディスクがボトルネックでなくなり，CPU の処理能力を最大限に発揮することが可能となったためである．

5. むすび

順次アクセス入力処理を実行するシステムにおけるディスク・キャッシュ導入時の処理能力向上効果を近似的に解析するモデルを提案した．解析の基本的な前

提条件は，(1)ジョブのディスク入出力はすべて順次アクセス入力処理とする，(2)ディスク・キャッシュにはジョブの用いるデータが先読みされてくる，(3)ディスクとディスク・キャッシュ間，およびディスク・キャッシュと主メモリ間のデータ転送は並行動作可能，の3点である．この条件のもとで以下の解析結果を得た．

まず一般に処理能力が，データ先読み単位をパラメータとして，ディスクの最大転送能力とディスク・キャッシュ容量の両者から決定されることを示した．次にジョブが単一ディスクのみを入出力実行する場合について，両者から定まるデータ処理率を具体的に求め，処理能力算定式を導出した．また，データ先読み単位の最適化についても考察を加えた．さらに，想定した計算機システムの諸元を使用した数値例について検討し，ディスク・キャッシュ導入前後の処理能力を比較した．

今後の課題としては，(1)ジョブが複数のディスクを入出力実行する場合の，データ入力の競合を考慮した処理能力の評価，(2)順次アクセス処理を実行するジョブとランダム・アクセス処理を実行するジョブが混在するシステムの性能解析，などがあげられる．

謝辞 本研究について，電気通信大学亀田壽夫助教授から貴重なご意見をいただいた．また，当社システム開発研究所川崎淳所長，同小田原工場小菅富士夫部長，宮崎道生副技師長，同ソフトウェア工場高須昭輔部長は研究の機会を与えて下さった．さらに同システム開発研究所大町一彦，北嶋弘行両主任研究員，木下俊之研究員より終始ご援助をいただいた．これらの方々に深く感謝いたします．

参考文献

1) Baskett, F., Chandy, K. M., Muntz, R. R. and Palacious, F. G.: Open, Closed and Mixed Networks of Queues with Different Classes of Customers, *J. ACM*, Vol. 22, No. 2, pp. 248-260 (1975).
2) Denning, P. J.: Effects of Scheduling on File Memory Operations, Proc. SJCC, pp. 9-21 (1967).
3) 日立製作所：HITAC H-8536-1 ディスク制御装置 H-8595 ディスク駆動装置，解説書 8080-2-022 (1980).
4) 金子，菊池，田中：ディスクキャッシュシステム，*FUJITSU*, Vol. 34, No. 2, pp. 253-261 (1983).
5) Kleinrock, L.: *Queueing Systems*, Vol. II, *Computer Application*, John Wiley & Sons, Inc., New York (1976).
6) 宮地，三石，溝口：階層型ディスクキャッシュ・サブシステムの方式と性能評価，情報処理学会計算機システムの制御と評価研究会資料，16-2 (1982).
7) 根岸，米田：バッファ付きディスク装置の性能評価，情報処理学会第23回全国大会予稿集，6E-3, pp. 131-132 (1981).
8) 西垣，山本：資源割当て優先度のある多重プログラミング・システムのボトルネック解析，情報処理学会論文誌，Vol. 23, No. 5, pp. 562-569 (1982).
9) Smith, A. J.: On the Effectiveness of Buffered and Multiple Arm Disks, Proc. 5th Annual Symp. Computer Architecture, pp. 242-248 (1978).
10) Smith, A. J.: Sequential Program Prefetching in Memory Hierarchies, *Computer*, Vol. 11, No. 12, pp. 7-21 (1978).
11) 菅，上田，田中：ディスク・キャッシュ装置のシミュレーションによる効果測定，情報処理学会論文誌，Vol. 22, No. 1, pp. 22-28 (1981).
12) Tokunaga, T., Hirai, Y. and Yamamoto, S.: Integrated Disk Cache System with File Adaptive Control, Proc. Compcon Fall 80, pp. 412-416 (1980).
13) Welch, T. A.: Analysis of Memory Hierarchies for Sequential Data Access, *Computer*, Vol. 12, No. 5, pp. 19-26 (1979).
14) 西垣，山本，木下，大町：順次アクセス読みこみ動作におけるディスク・キャッシュの効果解析，情報処理学会第26回全国大会予稿集，7N-4, pp. 181-182 (1983).
15) Brandwajn, A.: Models of DASD Subsystems: Basic Model of Reconnection, *Performance Eval.*, Vol. 1, No. 3, pp. 263-281 (1981).

(昭和58年6月29日受付)
(昭和58年9月13日採録)