



論文

多重プログラミング・システムにおけるフィードバック 概念にもとづく一般資源管理方式*

西 垣 通**

Abstract

A new resource scheduling algorithm for balanced throughput/response optimization is presented. The algorithm, called the General Resources Manager, employs priority control of a central processing unit and channels, as well as swapping. The scheduling scheme is based on a feedback concept, which keeps the utilization of each resource and the service rate of each process within prespecified ranges.

Experimental simulation reveals high performance in various interactive/batch application environments. Performance is compared with the traditional algorithm which employs only swapping.

The General Resources Manager is expected to increase the efficiency of multiprogrammed computer systems.

1. ま え が き

複数のプロセスが計算機資源を競合する多重プログラミング・システムにおいて、各プロセスへの資源配分は処理能力向上と応答性保証の2目標をみたすよう実行されねばならない。このため、多くの方式が検討されてきた¹⁻⁷⁾。近年、フィードバック概念を用いて総合的に資源を管理する方式が注目されている¹⁻³⁾。同方式では、各資源の利用率や各プロセスに供給される資源サービス量を定期的に測定し、値があらかじめ設定した目標値域やサービス目標函数から外れた場合、外れた度合に応じた制御行動をとる。

サービス目標函数を用いてプロセスへの資源サービス量を制御し、応答性を保証する方式は、A. J. Bernstein, J. C. Sharp 等の提案した Policy Driven Schedulerにより実現された^{2,3)}。しかし上記2目標をバランスよく達成するフィードバック制御は、IBM 社の OS/VS2 Release 2 System Resources Manager

で最初に試みられた¹⁾。System Resources Managerは制御手段としてプロセスへの実メモリの割り当て（スワッピング***）を用いる。しかし、実メモリが性能上のネックとならない場合、スワッピングのみでは効果が上がらない。たとえば実メモリに余裕があり、中央処理装置やチャネルに関する競合が生ずるとき、スワッピングにより上記2目標を達成することはできない。

本論文では、上記2目標を多様な状況下で達成するために、制御手段を一般資源の割り当てに拡張した方式を提案する。従来個別に行われていた各資源の割り当て優先度の決定を相互に関連づけたことが、本方式の特徴である。このとき資源割り当てにおける拘束関係に着目する。実メモリに加え、中央処理装置およびチャネルの割り当てを制御手段とするアルゴリズムを具体的に提示する。さらに会話、バッチの混在するアプリケーションにおける本方式の制御効果を、スワッピングのみを制御手段とする方式と、シミュレーションにより比較する。

2. 処理能力と応答性の均衡向上

2.1 問題および用語の定義

本論文では、トランザクションに対する各種資源の

* The General Resources Manager based on Feedback Concept in Multiprogrammed Computer Systems by Tohru NISHIGAKI (Systems Development Laboratory, Hitachi, Ltd.).

** (株)日立製作所システム開発研究所

*** ページではなくプロセスの入れ替えを意味する。

割り当て優先度決定アルゴリズムの問題を扱う。まず用語を定義する。

(1) トランザクション：システムに到着する、資源割り当て要求単位。資源として実メモリ、中央処理装置、チャンネルに着目すると、会話型ジョブにおけるコマンドの処理、バッチ・ジョブにおけるジョブ・ステップの処理。

(2) 資源：トランザクションから割り当てを要求される論理的単位。チャンネルは論理チャンネルを意味する。マルチ、シングルによらず中央処理装置は1つの資源とする。実メモリは、個々のページでなくワーキングセット⁶⁾に対応するページ群を割り当ての単位とする。

(3) 資源利用率：資源の全サービス供給能力のうち、いずれかのトランザクションに供給された量の占める割合(%)。

(4) 資源サービス累計：あるトランザクションに、到着以来供給された資源サービス量の累計。各種資源のサービスを service unit (su) とよぶ一定単位に換算して求める。

(5) 資源サービス率：あるトランザクションに、1秒あたり供給される資源サービス量。単位は su/sec。

2.2 状態と制御目標

システムの状態に対応して、処理能力と応答性の均衡向上を達成するための制御方策が存在する。これを求めるための準備として、「状態」と「制御目標」につき述べる。状態は、各資源利用率と、システム内の各トランザクションの資源サービス率とから与えられる。前者は処理能力、後者は応答性の向上に関連する。

(1) 処理能力向上のための資源利用率制御目標

資源 j の利用率を u_j 、目標値域下限値を ρ_j とする。 $u_j \geq \rho_j$ のときビジー、 $u_j < \rho_j$ のときアイドルと2値で表現する。 n 個の資源が存在するとき、 2^n 個の状態が存在する。制御目標は次式で与えられる。

$$u_j \geq \rho_j \quad (j=1, 2, \dots, n). \quad (1)$$

上記目標と関連して、トランザクション i の資源 j 使用特性値 v_{ij} を定義する。 v_{ij} は、 i と他のトランザクションとの競合がないとき、資源 j の i への供給量が j の全供給能力のうちで占める割合(%)を表わす。

v_{ij} が大きいとき、 i を j のヘビー・ユーザとよぶ。

(2) 応答性向上のためのサービス分配制御目標

トランザクション i に供給された資源サービス累計 $R_i(\tau)$ は式 (2) で与えられる。

$$R_i(\tau) = \sum_j e_j \cdot r_{ij}(\tau) \quad (2)$$

τ は到着以来の経過時間を表わす。 $r_{ij}(\tau)$ は τ の間に i に与えられた資源 j のサービス量であり、 e_j は重み係数である。 $R_i(\tau)$ を Policy Curve $f(\tau)$ 以上に保つよう制御する方式が Policy Driven Scheduler^{2,3)}である。本論文では、負荷変動に対する柔軟性の点から、System Resources Manager¹⁾における Performance Objective g をサービス目標函数として採用する。 g は、 $R_i(\tau)$ の傾きである資源サービス率の目標値を、負荷レベルと関連づけたものである*。 i に供給されている資源サービス率を S_i 、 g の逆函数を g^{-1} とするとき、次式で得られる値を i の Normalized Workload Level (NWL) とよぶ (Fig. 1 参照)。 S_i は $R_i(\tau)$ の差分として求まる。

$$NWLi = g^{-1}(S_i). \quad (3)$$

状態は各トランザクションの NWL により定められる。制御目標は NWL のバラツキの最小化であり、次式により表わされる。

$$\sum_i (NWLi - SWL)^2 \rightarrow \min \quad (4)$$

ただし System Workload Level (SWL) はシステムに存在するトランザクションの NWL の平均値である。式 (4) は、資源サービス率の絶対値でなく、 g で規定される相対値を保証するものである。

3. 資源割り当ての拘束関係

制御方策は、具体的には資源割り当て優先度の変更により実現される。資源 j に対するトランザクション i の割り当て優先度を $P_i(j)$ とするとき、各状態で $P_i(j)$ を定めることにより式 (1)(4) をみたすこと

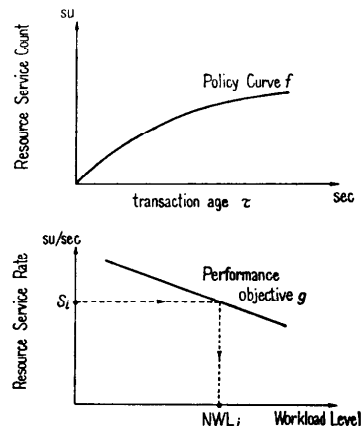
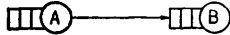


Fig. 1 Service objective function: Policy Curve, Performance Objective.

* g は負荷レベルの単調減少函数とする。

が、解決すべき問題である。このためには、資源割り当てにおける拘束関係に着目することが有効である。

トランザクションには、資源Aを割り当てられることが資源Bに対する割り当て要求発生条件である、という割り当ての拘束関係が存在する。いずれかのトランザクションがこの関係をみたすとき、AをBの predecessor, BをAの successor とよび、次のように表わす。



いま、Aがビジー、Bがアイドルであるとき、Bのヘビー・ユーザに対するAの割り当て優先度を増大させればBの利用率は上がる。また、ビジーな資源の割り当てはサービス分配のポイントであるから、Aの割り当て優先度を変更することにより、NWLのパラツキを制御できる。

中央処理装置、チャネル、実メモリ、仮想メモリについて割り当ての拘束関係を Fig. 2 に示す。中央処理装置と実メモリはあるトランザクションに同時に割り当てられるが、中央処理装置とチャネルとは通常同時には割り当てられない。しかしスケジューリングの観点からは、両者を同一に扱うことができる。

4. 一般資源管理方式 (The General Resources Manager)

4.1 資源利用率/サービス分配の制御方策

中央処理装置、チャネル #1、チャネル #2、実メモリの4種の資源 (Fig. 2 の鎖線内) について、具体的に割り当て優先度 $P_i(j)$ を提示する。

資源がビジーのとき1、アイドルのとき0とし、 $2^3, 2^2, 2^1, 2^0$ を各々実メモリ、中央処理装置、チャネル #1、チャネル #2 に対応させる。システムの状態は 0000~1111 の16通りである。処理能力向上のためには、中央処理装置、チャネル #1、チャネル #2 がビジーであればよいから、式(1)は次のようになる。

$$u_j \geq \rho_j \quad (j = \text{CPU, CH1, CH2}) \quad (5)$$

各状態*において、式(4)、(5)で与えられる制御目標を達成するために、次の制御方策をとる。

(1) 資源利用率制御方策

アイドルな資源の predecessor であるビジーな資源に関し、アイドルな資源のヘビー・ユーザの割り当て

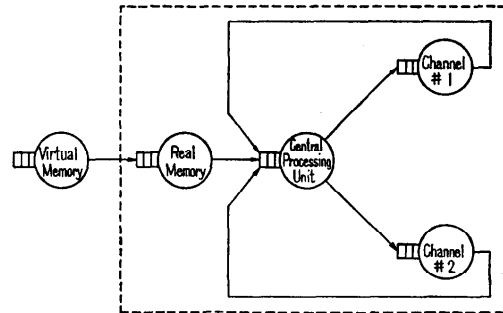


Fig. 2 Graphical expression of resource allocation constraints.

優先度を上げる。実メモリの割り当てについてこれを実行する状態は 1000~1110 の7通り、中央処理装置については 0100~0110, 1100~1110 の6通り、チャネル #1 については 0010, 0011, 1010, 1011 の4通り、チャネル #2 については 0001, 0011, 1001, 1011 の4通りである**。

(2) サービス分配制御方策

ビジーな資源に関し、NWL が相対的に大なるトランザクションの優先度を上げ、小なるトランザクションの優先度を下げる。これを実行するのは、いずれの資源についてもそれがビジーな場合であるから、各々8通りの状態においてである。

4.2 資源割り当て優先度決定方式

前節の制御方策を実現するため、定期的に資源割り当て優先度を算定する。

(1) 実メモリ割り当て優先度 $P_i(\text{RM})$

トランザクション i に対する実メモリ割り当て優先度 $P_i(\text{RM})$ は式(6)により与えられる。ただしシステム内のトランザクションのワーキングセット・サイズの総和が実メモリ総量以下であれば、全トランザクションに実メモリが割り当てられ、スワッピングは行われない。

$$P_i(\text{RM}) = \alpha_M \cdot \text{CP}_i(\text{RM}) + \beta_M \cdot \text{IO}_i(\text{RM}) + \gamma_M \cdot \text{WL}_i(\text{RM}) \quad (6)$$

$\alpha_M, \beta_M, \gamma_M$ は重み係数であり、処理能力、応答性のいずれを重視するか均衡をとることができる。

$$\begin{cases} \text{CP}_i(\text{RM}) = v_{i,\text{CPU}} \cdot \delta_{\text{CPU}} \cdot (\rho_{\text{CPU}} - u_{\text{CPU}})^2 \\ \text{IO}_i(\text{RM}) = v_{i,\text{CH1}} \cdot \delta_{\text{CH1}} \cdot (\rho_{\text{CH1}} - u_{\text{CH1}})^2 \\ \quad + v_{i,\text{CH2}} \cdot \delta_{\text{CH2}} \cdot (\rho_{\text{CH2}} - u_{\text{CH2}})^2 \\ \text{WL}_i(\text{RM}) = (\text{NWL}_i - \text{SWL}) \cdot |\text{NWL}_i - \text{SWL}| \end{cases} \quad (7)$$

$$\delta_j = \begin{cases} 1 & : u_j < \rho_j \\ 0 & : u_j \geq \rho_j \end{cases} \quad (j = \text{CPU, CH1, CH2}). \quad (8)$$

$u_j, v_{ij}, \text{NWL}_i$ は定期的に測定する ($j = \text{CPU, CH1, CH2}$).

* 各状態は概念的には NWL によりさらに細分されるが、方策はそれらの中では共通である。

** トランザクションはチャネル #1 か #2 のいずれか一方のみを用いると仮定する。

CH2). v_{ij} は、たとえば i が Δt 演算処理を行うごとに入出力命令を発行してチャンネル #1 を $\Delta t'$ 用いるとき、次のようにして求める。

$$\begin{cases} v_{i,CPU} = 100 \cdot \Delta t / (\Delta t + \Delta t') \\ v_{i,CH1} = 100 \cdot \Delta t' / (\Delta t + \Delta t') \end{cases} \quad (9)$$

$P_i(RM)$ が大なるトランザクションから実メモリが与えられる。ただし、過度のスワッピング・オーバーヘッドを防ぐため、スワッピングが実行されるのは次式がなりたつ場合のみである*。

$$P_i(RM) - \sum_{k \in X} P_k(RM) \geq SIT + |X| \cdot SOT \quad (10)$$

k' は実メモリを新たに割り当てられるトランザクション、 X はかわりに実メモリを剝奪されるトランザクションの集合、 $|X|$ はその要素数を表わす。SIT, SOT は、各々スワップ・イン閾値、スワップ・アウト閾値である。上記方式は、基本的に System Resources Manager の方式にもとづいている。

(2) 中央処理装置割り当て優先度 $P_i(CPU)$

トランザクション i に対する中央処理装置割り当て優先度 $P_i(CPU)$ は、式 (11) により与えられる。ただし、変更が行われるのは、中央処理装置がビジー ($u_{CPU} \geq \rho_{CPU}$) の場合に限る。

$$P_i(CPU) = \alpha_P \cdot IO1_i(CPU) + \alpha_P \cdot IO2_i(CPU) + \beta_P \cdot WL_i(CPU) \quad (11)$$

α_P, β_P は重み係数である。

$$\begin{cases} IO1_i(CPU) = (1 - \delta_{CH1}) \cdot OLDCH1 \\ \quad + \delta_{CH1} \cdot \xi \cdot v_{i,CH1} \cdot (\rho_{CH1} - u_{CH1}) \\ IO2_i(CPU) = (1 - \delta_{CH2}) \cdot OLDCH2 \\ \quad + \delta_{CH2} \cdot \xi \cdot v_{i,CH2} \cdot (\rho_{CH2} - u_{CH2}) \\ WL_i(CPU) = \eta \cdot (NWL_i - SWL) \end{cases} \quad (12)$$

ξ, η は正定数である。OLDCH1, OLDCH2 は、各々前回に算出された $IO1_i(CPU), IO2_i(CPU)$ を表わす。チャンネル #1 の利用率が目標値域内にあれば ($u_{CH1} \geq \rho_{CH1}$)、これを向上させるために $P_i(CPU)$ を変更する必要がなく、現状を保てばよいことから、式 (12) 第1式の第1項がえられる。第2式の第1項についても同様である。

中央処理装置は、 $P_i(CPU)$ が大なるトランザクションから割り当てられる。トランザクション k に中央処理装置が割り当てられているとき、 $P_k(CPU)$

$> P_{k'}(CPU)$ なる k' が割り当て要求を発生すると、その時点で中央処理装置は k' に割り当て変更される。

上記方式は、従来のダイナミック・ディスパッチング制御⁹⁾と、チャンネル利用率向上という点で共通しているが、個々のチャンネルを識別している点、及びサービス分配制御も行っている点が異なる。また、従来のタイム・スライス制御は、上記方式のサービス分配制御に概念的に含まれる。

(3) チャンネル # m の割り当て優先度 $P_i(CHm)$ ($m=1, 2$)

トランザクションに対するチャンネル # m の割り当て優先度 $P_i(CHm)$ は、式 (13) により与えられる。ただし、変更が行われるのは、チャンネル # m がビジー ($u_{CHm} \geq \rho_{CHm}$) の場合に限る。

$$P_i(CHm) = \alpha_c \cdot CP_i(CHm) + \beta_c \cdot WL_i(CHm) \quad (13)$$

α_c, β_c は重み係数である ($m=1, 2$)。

$$\begin{cases} CP_i(CHm) = (1 - \delta_{CPU}) \cdot OLDSCP \\ \quad + \delta_{CPU} \cdot \nu \cdot v_{i,CPU} \cdot (\rho_{CPU} - u_{CPU}) \\ WL_i(CHm) = \xi \cdot (NWL_i - SWL) \end{cases} \quad (14)$$

ν, ξ は正定数である。OLDSCP は、前回算出された $CP_i(CHm)$ を表わす。

$P_i(CHm)$ が大なるトランザクションの発生した入出力処理から実行される。ただし、トランザクション k の発生した入出力処理実行中に、 $P_{k'}(CHm) > P_k(CHm)$ なる k' が入出力処理を発生しても、 k の入出力処理完了まで待たされる。

5. シミュレーションによる評価

一般資源管理方式 (General Resources Manager) の効果を、System Resources Manager で用いられる方式との比較において考察する (以下、前者を GRM、後者を SRM と略記する)。SRM において、実メモリのスケジューリングは GRM と同一**とするが、チャンネルに関しては First Come First Serve と仮定する。中央処理装置についてはバッチ・トランザクションに対しダイナミック・ディスパッチング制御を行い、会話トランザクションにはバッチ・トランザクションより大きい一定の割り当て優先度を仮定する***。

両方式の比較は、CSS (Computer System Simulator) 言語の約 2k ステップのモデルを開発し、5 分間の状況をシミュレートすることにより行った。

5.1 周囲条件

シミュレーション評価に際し設定した周囲条件をまとめる。

* ただし、会話トランザクションの到着や完了、未使用ページ不足等の場合は、必要に応じてスワッピングを行う。

** したがって、以下述べる SRM は、OS/VS2 Release 2 の SRM と全く同一ではない。

*** バッチ・トランザクションによる会話トランザクションの応答性低下防止のためである。

Table 1 The characteristics of the workload model simulated.

Characteristics	Transaction		Terminal Initiated Transaction	
	CPU bound	I/O bound	Interactive (I)	Executive (E)
Execution Requirement (steps)	12M	6.6M	100k	12M
Working-set Size (pages, 1 page= 4 k Bytes)	40~60	40~60	20	40~60
I/O Operation Frequency (/100 k steps)	1.0	10.0	1.0	1.0
Page Fault Rate (/100 k steps)	0.5	0.5	4.0	0.5
Think Time (sec)	—	—	20	20

(1) ハードウェア・モデル

想定した機器構成を Fig. 3 に示す。中央処理装置の平均処理速度は 0.5 μ s/step とする。実メモリ量はシステム用領域を除いたユーザ用領域を表わす。ページング用デバイスはドラム(転送速度 1,450 k Bytes/sec), ユーザ入出力用デバイスは 8 台のディスク(転送速度 806 k Bytes/sec) とする。

(2) ワークロード・モデル

バッチと会話の混在した, Table 1 に示すワークロードを想定する。Table 1 において, 入出力処理はディスクに対する 8 k バイトの入出力とし, 演算処理はデータ管理部分を含む。

バッチ・トランザクションは, シミュレーション開始時点で既にジョブ・キューに並んでおり, あるトランザクションが実行完了するとただちに同一の新たなトランザクションが再開される。システムに存在するバッチ・トランザクションの数(多重度)は 8 とする。CPU bound と I/O bound の 2 種を設定したのは, 資源利用率制御効果を確認するためである。

会話用端末は, シミュレーション実行中常にアクティブと仮定する。I 型(インタラクティブ)と E 型(エグゼキューティブ)の 2 種を設定したのは, 後者が前者の応答性¹⁰⁾に与える影響を観察するためである。前者はテキスト編集などの処理を想定しており, モデルは H. A. Anderson¹¹⁾, A. Scherr¹²⁾等の実測結果を参考にして定めた。後者は FORTRAN プログラムなどの実行処理を想定している。

なお, システムへの到着時点で, トランザクションの資源使用特性値 v_{ij} は 0 とし, 実行中に学習していくものとする。また, 資源サービス率 S_i の初期値も 0 とする。

(3) 制御パラメータ設定値

シミュレーション実行に際して設定した, スケジューラ内部の制御パラメータの値を示す。

(a) 資源サービス量の単位

中央処理装置に関しては 10,000 命令実行を 1 su, チャネルに関しては入出力処理 1 回を 1 su とする。

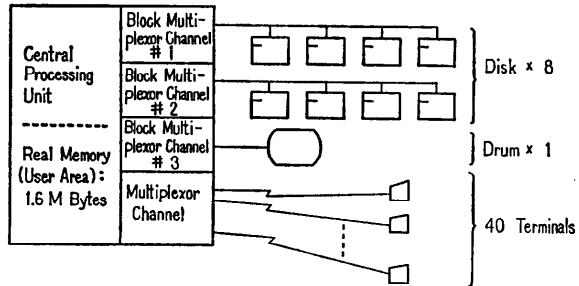


Fig. 3 The hardware configuration simulated.

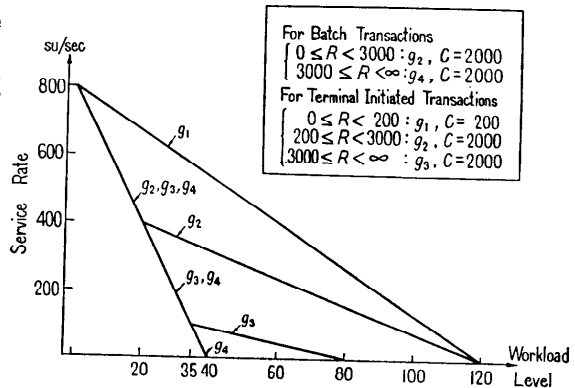


Fig. 4 The specified service objective functions to control relative service rates.

式(2)において $e_{CPU} = e_{CH1} = e_{CH2} = 10$ とする。このとき, I 型会話トランザクションを除くすべてのトランザクションについて, 実行完了までに供給される資源サービス量は 13,200 su である (Table 1 参照)。

(b) サービス目標函数

設定したサービス目標函数を Fig. 4 に示す。

Fig. 4 において, R はトランザクションの資源サービス累計である。 C は過度のスワッピング・オーバーヘッド防止用パラメータであり, いったんスワップ・インされたトランザクションは C 以上サービスを受けるまでは, 原則としてスワップ・アウトされない。 Fig. 4 は, 会話をバッチよりやや優先させ, 短

Table 2 Performance comparison of the two schedulers (GRM: upper row, SRM: lower row) with respect to the processing time of terminal initiated interactive (I) transactions.

CH1/CH2		4/4			6/2
w	I/E	36/4	32/8	28/12	36/4
Processing Time, Mean (sec)	60	{0.47 0.55}	{0.51 0.83}	{0.48 1.98}	{0.43 0.62}
	40	{0.42 0.70}	{0.63 1.35}	{0.61 2.48}	{0.42 0.77}
Processing Time, Standard Deviation (sec)	60	{0.30 0.31}	{0.41 0.64}	{0.34 1.35}	{0.27 0.37}
	40	{0.20 0.60}	{0.59 1.07}	{0.64 2.17}	{0.21 0.52}
Processing Time, 90% Accumulation (sec)	60	{1.00 1.25}	{1.00 2.00}	{1.00 4.00}	{0.75 1.25}
	40	{0.75 1.50}	{1.25 3.00}	{1.25 5.50}	{0.75 1.75}
Number of Completion (/300 sec)	60	{523 524}	{466 463}	{413 379}	{528 525}
	40	{532 519}	{462 446}	{403 374}	{530 519}

いトランザクションを長いトランザクションより優先させたもので、標準的な設定である。

(c) その他の制御パラメータ

$P_i(RM)$, $P_i(CPU)$, $P_i(CHM)$ の算出周期は、各各約 5 sec, 2 sec, 5 sec とした。制御間隔を余りにも短くすることは、オーバヘッドの増大を招く。したがって、中規模以上のトランザクションを主な制御対象とする。トランザクションの到着直後には高い資源割り当て優先度を与えることにより、制御間隔の間で完了してしまうような短小なトランザクションに対してもサービスを保証することができる。

その他のパラメータ設定値は以下のとおりである。

$$\rho_{CPU}=70, \rho_{CH1}=\rho_{CH2}=30, \alpha_M=\beta_M=\gamma_M=1$$

$$SIT=SOT=100, \alpha_F=1, \beta_F=2, \zeta=0.02$$

$$\eta=0.13, \alpha_e=\beta_e=1, \nu=0.01, \xi=0.13$$

5.2 性能比較結果

3 種のパラメータを変えて測定を行った。第 1 はトランザクション*のワーキングセット・サイズ w である。第 2 は端末特性であり、I/E は 40 端末のうちで I 型, E 型会話トランザクションを発生する端末数の比を示す。第 3 はバッチ特性であり、CH1/CH2 はチャンネル #1 を使用するトランザクション数とチャンネル #2 を使用するトランザクション数との比である。この値が 6/2 のとき前者の 83%, 後者の 50% を I/O bound とし、4/4 のとき両者とも 50% を I/O bound

* I 型会話トランザクションは除く。

Table 3 Performance comparison of the two schedulers (GRM: upper row, SRM: lower row) with respect to the processing time of terminal initiated executive (E) transactions.

CH1/CH2		4/4			6/2
w	I/E	36/4	32/8	28/12	36/4
Processing Time, Mean (sec)	60	{94.3 57.8}	{94.6 86.7}	{121.7 90.9}	{60.9 62.4}
	40	{86.4 39.7}	{87.7 63.1}	{112.2 74.9}	{60.4 37.9}
Processing Time, Standard Deviation (sec)	60	{30.6 34.2}	{30.0 35.3}	{32.2 48.8}	{25.6 37.0}
	40	{34.4 29.7}	{35.2 34.0}	{32.6 51.9}	{27.8 28.6}
Processing Time, 90% Accumulation (sec)	60	{140.0 110.0}	{150.0 130.0}	{160.0 150.0}	{100.0 130.0}
	40	{130.0 80.0}	{160.0 100.0}	{160.0 150.0}	{80.0 80.0}
Number of Completion (/300 sec)	60	{8+2.1 13+1.6}	{17+3.6 20+2.4}	{21+4.5 27+4.6}	{12+1.5 14+0.9}
	40	{10+1.5 17+2.6}	{18+4.1 25+3.3}	{22+4.5 29+5.4}	{12+2.4 20+0.7}

Table 4 Performance comparison of the two schedulers (GRM: upper row, SRM: lower row) with respect to the processing time of batch transactions.

CH1/CH2		4/4			6/2
w	I/E	36/4	32/8	28/12	36/4
Processing Time, Mean (sec)	60	{79.9 100.9}	{122.7 137.6}	{140.9 214.4}	{90.7 109.6}
	40	{68.3 90.0}	{109.6 177.5}	{158.4 —}	{76.5 102.2}
Processing Time, Standard Deviation (sec)	60	{28.3 52.0}	{39.8 75.9}	{27.0 12.9}	{40.2 57.1}
	40	{31.7 55.8}	{44.6 49.8}	{50.8 —}	{52.8 56.4}
Processing Time, 90% Accumulation (sec)	60	{120.0 190.0}	{170.0 270.0}	{170.0 230.0}	{150.0 190.0}
	40	{100.0 140.0}	{200.0 260.0}	{200.0 —}	{190.0 210.0}
Number of Completion (/300 sec)	60	{25+4.4 19+4.5}	{14+5.2 11+4.6}	{11+4.3 2+5.0}	{20+5.3 15+5.0}
	40	{30+3.2 18+4.0}	{17+4.0 9+3.2}	{12+3.8 0+5.9}	{25+4.8 14+4.6}
NWL, Standard Deviation	60	{19.7 19.6}	{21.6 24.4}	{19.2 26.4}	{18.3 23.1}
	40	{18.6 22.8}	{23.9 30.4}	{22.8 29.1}	{17.0 26.1}

とする。残りは CPU bound である。

(1) サービス分配能力 (応答性)

I 型会話, E 型会話, バッチの各トランザクションの processing time をそれぞれ Table 2, 3, 4 に示す。processing time は、トランザクションが到着してから実行完了までの経過時間を表す。90% accu-

Table 5 Performance comparison of the two schedulers (GRM: upper row, SRM: lower row) with respect to throughput.

	w	CH1/CH2		4/4			6/2
		I/E		36/4	32/8	28/12	36/4
Throughput (su/300 sec)	60			{582,006 561,972}	{579,429 556,117}	{586,136 551,134}	{572,415 520,586}
	40			{650,774 609,289}	{623,433 585,321}	{605,879 576,812}	{643,425 578,694}
CPU Utilization, in the parenthesis: by batch (%)	60			{73.9 (44.8) 75.3 (36.9)}	{77.6 (27.6) 77.2 (24.1)}	{81.1 (23.1) 81.0 (10.7)}	{71.1 (34.8) 66.5 (28.1)}
	40			{81.9 (49.8) 82.5 (34.0)}	{82.9 (30.1) 83.6 (19.7)}	{84.6 (24.0) 85.3 (9.0)}	{79.9 (41.9) 76.5 (25.9)}
Channel #1 Utilization, by user I/O (%)	60			{24.9 19.1}	{19.1 17.3}	{17.1 11.5}	{33.8 27.4}
	40			{28.5 18.8}	{22.6 15.4}	{17.9 12.4}	{37.6 28.9}
Channel #2 Utilization, by user I/O (%)	60			{23.9 20.3}	{21.9 16.4}	{18.8 12.2}	{18.2 15.0}
	40			{29.0 22.8}	{23.6 14.4}	{18.5 12.1}	{21.0 14.3}

mulation は, processing time がそれ以下のトランザクション数が完了した総数の 90% を占める点である。number of completion において, 第 1 項は 300 秒間に実行完了したトランザクション数である。第 2 項* は 300 秒経過時点でまだ実行中のすべてのトランザクションについての資源サービス累計の和を 13,200 su で除算し, 完了した数に換算した値である。Table 4 には上記に加え定期的にシステム内の全トランザクションについて測定した NWL の標準偏差をも示す。

Table 2 に示すように, GRM においては測定パラメータが変化しても応答性はほぼ安定している。SRM では, I/E において E が增大すると応答性が低下する。低下の程度は, $w=40$ すなわち実メモリがネックでないとき著しい。この理由は, GRM では中央処理装置割り当てに関しサービス分配制御を行っているのに対し, SRM ではこれを行っていないためである。GRM では, E 型会話トランザクションの中央処理装置割り当て優先度は, 供給された資源サービス率にしたがって動的に調整される。しかし SRM では, 常に I 型会話トランザクションと同一であり, したがって後から到着する I 型会話トランザクションにはただちに中央処理装置が割り当てられず, その応答性が低下する。

なお, バッチ特性 (CH1/CH2) の変化が応答性に与える影響は, GRM, SRM いずれにおいても小さい。

Table 3, 4 に示すように, $w=40$ の場合, SRM で

* Table 2 の I 型会話トランザクションについては, 第 2 項は無視する。

は E 型会話トランザクションにサービスが偏る。I/E = 28/12, $w=40$ のとき, バッチ・トランザクションの沈みこみが生じている。実メモリがネックでないとき, トランザクションの大半はスワップ・イン状態にあるので, スワッピングのみでサービス分配を制御することはできない。GRM と SRM とのサービス分配制御能力の差異は, Table 4 に示した NWL の標準偏差からも明らかである。

(2) 処理能力

処理能力測定結果を Table 5 に示す。Table 5 において throughput は 300 秒間にいずれかのトランザクションに与えられた資源サービス量の総計を表わす。CPU utilization は全トランザクションによる利用率であり, かっこ内は特にバッチ・トランザクションによる値である。

GRM における throughput は, SRM より 5~10% 向上する。向上効果は, チャネル利用がアンバランスな場合 (CH1/CH2=6/2) に顕著である。このとき, CPU bound の E 型会話トランザクションが SRM において GRM より多く実行されているにもかかわらず, CPU utilization は GRM が SRM より大きい点は注目すべきである。これはチャネル #1 の割り当てにおける資源利用率制御効果を示している。

(3) オーバヘッド

各種のオーバヘッドを Table 6 (次頁参照) に示す。Total CPU overhead は, GRM/SRM 実行に加えスワッピング, ページ・フォールト等によるオーバヘッドである。この中で GRM/SRM 実行のみによるオー

Table 6 Performance comparison of the two schedulers (GRM: upper row, SRM: lower row) with respect to overhead.

CH1/CH2 T/R	w	4/4			6/2
		36/4	32/8	28/12	36/4
Total CPU Overhead (%)	60	{16.9 16.5}	{15.7 15.5}	{14.9 13.7}	{16.8 16.4}
	40	{17.0 16.3}	{15.8 15.1}	{14.7 13.6}	{17.0 16.1}
CPU Overhead, by GRM/SRM (%)	60	{0.21 0.17}	{0.22 0.18}	{0.23 0.18}	{0.21 0.17}
	40	{0.21 0.16}	{0.22 0.18}	{0.24 0.18}	{0.21 0.16}
CPU Overhead, by swapping (%)	60	{10.0 9.8}	{9.2 9.3}	{8.7 8.1}	{9.8 9.9}
	40	{9.6 9.3}	{9.1 8.9}	{8.5 7.9}	{9.6 9.4}
Swap-out Count (/300 sec)	60	{596 589}	{552 555}	{521 483}	{585 594}
	40	{572 558}	{542 530}	{510 471}	{572 562}
Channel #3 Utilization, by paging (%)	60	{26.5 25.8}	{25.8 26.2}	{26.3 24.6}	{25.1 25.8}
	40	{22.9 22.4}	{22.9 22.6}	{22.4 21.0}	{22.9 22.3}

バヘッド、スワッピング実行にともなうオーバーヘッドについては各々別に示した。Total CPU オーバヘッドに関しては、GRM は SRM より、全体で 1 % 程度、GRM/SRM 自体については 0.04~0.06% の増加にとどまる。

Table 6 には、300 秒間に実行されたスワップ・アウト回数も示した。ページング、スワッピングのページ転送のオーバーヘッドは、チャンネル #3 の utilization から明らかなように、両方式ともほぼ同程度である。

6. む す び

一般資源の割り当てを制御手段として、資源利用率および各プロセスへの資源サービス量を目標値域内に保つフィードバック制御を行うことにより、処理能力と応答性を向上させる方式を提案した。実メモリ、中央処理装置、チャンネルの割り当て優先度を動的に決定するアルゴリズムを提示した。本方式は、スワッピングのみを制御手段とする System Resources Manager および従来のダイナミック・ディスパッチング制御、タイム・スライス制御を包含している。

スワッピングのみを制御手段とする方式とシミュレーションにより性能を比較した。その結果、処理能力、応答性ともに明らかな向上がみられた。

本方式により、会話とバッチの混在した多重プログラミング・システムの効率向上を期待しうる。今後は、オーバーヘッドの詳細な評価、パラメータ適正設定など、実現上の諸問題についてさらに検討を続ける予定である。

終りに、本研究について御指導いただいた東京大学大須賀節雄助教授に深く感謝いたします。

参 考 文 献

- 1) H. W. Lynch & J. B. Page: The OS/VS 2 Release 2 System Resources Manager, IBM Systems Journal, Vol. 13, No. 4, pp. 274~291 (1974).
- 2) A. J. Bernstein & J. C. Sharp: A Policy Driven Scheduler for a Time-Sharing System, Comm. ACM, Vol. 14, No. 2, pp. 74~78 (1971).
- 3) J. C. Sharp & J. N. Roberts: An Adaptive Policy Driven Scheduler, Proc. ACM SIGMETRICS SYMP. on Measurement & Evaluation, pp. 199~208 (1974).
- 4) L. Kleinrock: A Continuum of Time-Sharing Scheduling Algorithms, Proc. SJCC, pp. 453~458 (1970).
- 5) 亀田寿夫: 多重プログラミングシステムにおける資源配分の適応制御方式について、情報処理学会システム性能評価研究会資料 SE 13-3 (1976).
- 6) W. J. Doherty: Scheduling TSS 360 for Responsiveness, Proc. FJCC, pp. 97~111 (1970).
- 7) H. Hellerman: Some Principles of Time-Sharing Scheduler Strategies, IBM Systems Journal, Vol. 8, No. 2, pp. 94~117 (1969).
- 8) P. J. Denning: The Working-Set Model for Program Behavior, Comm. ACM, Vol. 11, No. 5, pp. 323~333 (1968).
- 9) K. D. Ryder: A Heuristic Approach to Task Dispatching, IBM Systems Journal, Vol. 9, No. 3, pp. 187~198 (1970).
- 10) S. Stimuler: Some Criteria for Time-Sharing System Performance, Comm. ACM, Vol. 12, No. 1, pp. 47~53 (1969).
- 11) H. A. Anderson, et al.: Tuning a Virtual Storage System, IBM Systems Journal, Vol. 14, No. 3, pp. 246~263 (1975).
- 12) A. Scherr: An Analysis of Time Shared Computer Systems, Cambridge, Mass. MIT Press (1967).

(昭和 52 年 8 月 24 日受付)

(昭和 52 年 12 月 27 日再受付)