

# デッドライン制御をベースとした応答性管理方式とその評価

土井根 敏明, 大野 直哉, 久保 秀士  
(日本電気(株) 中央研究所)

## 1. はじめに

コンピュータ・システムの大規模化, 形態の複雑化, 処理内容の多様化に伴って, システム全体を管理するオペレーティング・システムの資源管理機能の重要性は増してきている。資源管理機能はシステム・リソースを個々のジョブに適切に配分し割当てることにより, システムとしての性能目標の達成を図る。

システム性能の指標としては次の3種類を考慮する必要がある。<sup>(1)(2)</sup>

- ① システム・スループットの向上
- ② 個々のジョブに対する応答性の保証
- ③ システム管理者の望むサービス量(処理能力)の配分

大規模化, 複雑化に伴って, リソース管理の困難さが増すと同時に適切な管理を行った場合の効果は非常に大きくなるため, 有効なスループット向上方式が望まれる。処理内容の多様化は応答性に対する要求の多様化をもたらしており, 個々の利用者の要求に応じ, かつ全体としてバランスのとれた応答性管理が望まれる。また, 大規模化, 複雑化に伴って, 一つのシステムを共用する組織間, 業務間, あるいは処理次元間におけるシステム使用要求の競合に解決を与え, 適正なサービス量の配分を実現する必要が生じてきている。これら3種類の性能目標は互いに競合する面があるため, 各々の設置システムごとのシステム管理者のポリシーを反映して, バランスよく達成しうるようなメカニズムが提供されるべきである。また, 時々刻々変動するシステムの負荷状況, ジョブの特性, に応じた制御を行うため, 資源管理は動的に行われる必要がある。

本報告では, ACOSシリーズ 77 システム 800 モデル 3 (以下 ACOS 800-3 と略称) 大型機向けの OS である ACOS-4 / MVP (Multi-Virtual Processor) の機能検討の一環として行ってきた動的資源管理 (DRM; Dynamic Resources Management) 方式の検討について, 応答性管理方式とそのシミュレーションによる評価結果を中心として述べる。評価結果は 3), 4) で報告した内容に, いくつかの新しい結果を追加したものである。

ここで, 本報告の背景となる動作環境として ACOS 800-3 及び ACOS 4 / MVP について簡単に紹介しておく。ACOS 800-3 は, 従来よりの ACOS 4 系アーキテクチャの特徴である,

- プロセスの概念のサポート (具体的には, 状態管理, ディスロッキング, セマフォによる同期機構など)
- セグメンテーションによるプロセス毎の多重仮想記憶のサポート
- リング・プロテクションのサポート
- スタックを用いたコール, リターン機構のサポート
- 入出力チャネル・スケジューリングのサポート

のハードウェア/ファームウェアによる実現<sup>5)</sup>, の他にさらに,

- ノーシング<sup>6)</sup>の導入
- 密結合マルチプロセッサのサポート
- チャネル DAT の導入

などを行った、主記憶容量最大16MB、チャネル転送能力最大40MB/秒の大規模コンピュータである<sup>6)</sup>。ACOS-4/MVPは従来のACOS-4の技術を生かし、OSの一部として統合されたTSS(ATSS; Advanced Time Sharing System)とDB/DCシステム(VIS; Versatile Information System)、JES(バッチ及びRJEジョブの入出力管理)をもち、大規模システムにとって特に重要な、信頼性向上機能、資源管理機能、を重視して設計されている<sup>6)</sup>。

## 2. 応答性管理方式の概要

DRMでは制御の階層化を行い、論理的な部分と物理的な部分との分離を図っている。具体的には、内部構造と独立なユーザ・インタフェースとしてポリシーの設定があり、これを受けてポリシーの論理的実行階層がシステムの実況、各ジョブの実況に応じて、内部制御パラメータ値に交換する。最下層では、この値に従って、「リソースの割当てをスワッピング」、「ディスクパッキング」という手段により制御する。以下、この制御階層に沿って、応答性管理方式の概要を述べる。

### 2.1 応答性目標と管理上の制約

#### (1) 管理の対象と応答クラス

応答時間管理の対象は、バッチについてはジョブステップ、TSSとオンラインについてはインタラクションとする。これらを総称して、トランザクション(Xactと略記)と呼ぶ。管理の容易さから、Xactは応答性要求の性質によってグループ分けされ(これを応答クラスと呼ぶ)、このクラスごとに応答性目標が設定されてその達成が図られる形をとる。Xactは発生時に、各々自分の属する応答クラスを指定する(エンドユーザの要求)。

#### (2) 応答性に対する要求の表現

応答クラスに対する応答性目標の表現方法としては次の3種類が考えられる。

- a) Xact処理過程のガイドラインを設ける(微分的/積分的) -- e.g. SRM<sup>7)</sup>
- b) 相対的重要度を与える(プライオリティの付与) -- 従来の一般的方法
- c) 目標応答時間を与える

a, bの方式は直接的な応答目標を与えるものではなく、エンドユーザの要求に答える保証はない。また、ユーザの欲する応答時間を達成するのに余裕がある状況にあっても、予め定められた働きききに従って働き過ぎのサービスを行う可能性もある。cの方式では目標を直接に示せばいいため指定し易く、それを達成するための制御も行い易くエンドユーザの要求に答える可能性が高まる。また、目標の達成に余裕のある状況であることを把握できるので、目標を下回らない範囲でスループットの向上にリソースを回しうる。このため、ここではcの方式を採用する。すなわち、応答クラス別に目標応答時間と達成率目標とを設定して貰う(例えば、95%のXactが5秒以内の応答、など)。

#### (3) サービス配分管理面からの制約

DRMにおいては、1.で述べたように、ジョブのグループ(ドメインと呼ぶ)に対するサービス量の配分を管理することが必要と考えている<sup>2)</sup>。サービス配分管理は、各ドメインとシステム全体としての負荷状況、これに応じて定まる各ドメインの優先度

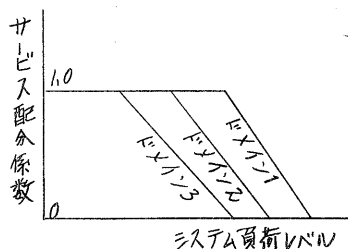


図1. 負荷レベルに応じたサービス配分

合(システム設置者のポリシーによる)に応じて、サービス量(処理能力)を各ドメインへ配分する(図1参照)。従って、応答性管理は各ドメインに割当てられたサービス量の制約の下で、各Xactへの配分を制御することになる。

#### (4) スループット向上とのバランス

ドメインに割当てられたサービス量は、そのドメインにおいて応答性優先の制御をすればする程、ロスとなる部分が大きくなる。逆にスループット優先の運用をすれば応答性は悪くなる。この両者のバランスのとり方は、システム管理者のポリシー(負荷状況の関数)として与えられ、これにも従う必要がある。

#### (5) 応答クラス間の遅れの配分

(3),(4)で述べた制約及びXact自身の走行時間(CPU+I/O時間)特性のために、設定された応答目標が必ず達成できるとは限らない。達成不可能な場合には応答目標を切下げることを考える。すなわち、レスポンスから見た負荷のレベルに応じて、各応答クラスの目標をどの程度落していかをポリシーとして与えておき(図2参照)、これに従って制御する。各応答クラスの応答性目標は、図2で示される関係まで含めて規定して貰うことになる。

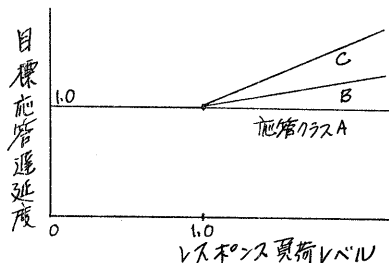


図2. 目標応答遅延度の決定

以下で述べる応答性管理方式は、以上(3),(4),(5)で述べた制約が具体的な外部条件として与えられた時に、その下で働く方式として考えられている。シミュレーション結果は、一つのドメイン内における結果として解釈しよう。

### 2.2 GPの導入とその制御<sup>1)</sup>

各時点における個々のXactの応答時間に関する要求の度合を示す指標として、GP(General Priority)を導入する。これにより、システムへの投入時刻、目標応答時間、達成率目標の異なる各Xactの応答性に関する要求を一元的に表現する。最下層の制御の場で各XactのもつGP値に対処した制御が行われることにより、応答性要求の度合いが反映されることになる。

応答目標が達成されるためには、各時点で各々のXactに対して適切なGP値が割当てられている必要がある。このために以下の3種の制御を導入する。

- a) デッドライン制御<sup>4)</sup> : Xact発生時には比較的低いGP値を与え、目標応答時間が迫るにつれてGP値を上昇させる。これにより、目標応答時間の達成率の向上と低負荷時におけるスループットの向上が期待できる。
- b) フィードバック制御 : 各時点におけるドメインの負荷状況、各応答クラスの日標達成状況に応じて、各クラスに与えるGP値あるいはGP値の枠を制御する。これにより、様々な負荷レベルに亘って一定の応答性をバランスよく実現できるようになると期待される。
- c) 個別制御 : Xactの目標応答時間の達成の難易が予測できる場合(コマンドにより処理時間がほぼ定まるもの等)、その情報をもとにXactの初期GP値を与える。これにより、無理なく無駄なく目標の達成を図りうる。

### 2.3 スワッピング・レベルにおける制御

スワッピング・レベルの制御とは、主記憶上に置くXactの集合を決定し、スワップイン、スワップアウトによりそれを実現する制御である。Xactの到着時、終了時における場合の他に、実行中における強制的なスワッピングも用

いる。ここではGPに基づく制御を基本とし、レスポンス多重度、メモリ・タイムスライスの概念も導入する。

α) GPによる制御： GPの値によりXactを3つのレベル(L1~L3)に分類し、各レベルに応じた制御を行う。L1は最も応答要求の厳しいレベルで、L1 Xactが発生すると原則として直ちにスワップインされる。このために、必要ならば他のレベルのXactをスワップアウトする。L2は応答時間を要求するレベルで、主記憶に空きがあればGPの高い順に優先的にスワップインされるが、他のジョブを追い出してまではスワップインされない。L3レベルは最も優先度が低く、L3 Xactは応答時間の観点からはスワップインの対象とならず、L1, L2のXactが無い時に、スルーフォットを長くするために最も適当なものか選択されてスワップインされる。

β) レスポンス多重度： L1, L2レベルのXactの主記憶多重度に上限、下限を設ける。これにより、スルーフォットとレスポンスのバランスにおけるホリシの反映を図り、また、高応答を要求するXact(TSSなど)の負荷変動が激しくてもスワッピング頻度の増大を抑える。

γ) メモリ・タイムスライス： 主記憶滞在時間(CPU時間)が一定以下のXactはスワップアウトの対象外とする。スワッピングの頻発を防ぎうる。

#### 2.4 ディスパッチング・レベルにおける制御

ディスパッチングでは、スルーフォットの観点から原則としてAPG(I/Oの多いプロセス優先度を高める)を採用するが、応答要求の特に厳しいXact(GP値≧DRB)にはAPG領域より高い優先度を与える。ディスク・スケジュールにおいても、原則としてシーク最適化を採用するが、応答要求の特に厳しいXactのI/O要求は、これよりも優先して処理する。

### 3. シミュレーションの方法<sup>1)</sup>

シミュレーション・モデルの基本的な構成を図3に示す。モデルにおけるシステム構成とハードウェアの性能を表1に示す(但し、性能はACOS 800-3の値のものではない)。システム負荷はバッチXactおよびTSS Xactとする。TSS Xactには4タイプがあり、各々は異なる応答クラスに属するとする。TSSのジョブ特性、応答性目標を表2に示す。バッチのジョブ特性はI/Oを多用するジョブの割合を高めている(事務計算指向)。バッチ用に3つの応答クラスを設けている。Xactの到着はTSSはポアソン到着、バッチは60個のstacked

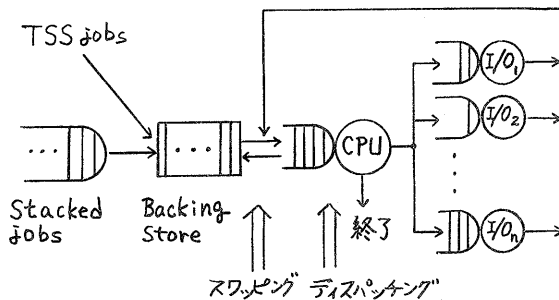


図3. モデルの基本的な構成

表1. システム構成及びハードの性能

	台数	性能*
CPU	1	1 MIPS
ディスク (ファイル用)	8 (MSC 2台)	Seek (平均) 20ms 回転待ち 8.3ms データ転送 1MB/s
ディスク (スラップ用)	1 (MSC 1台)	同上
CR	2	1050 枚/分
LP	3	1500 行/分

\* ACOS 800-3の値のものではない

jobs である。各 Xact のプログラム・サイズの違いは意識しない、すなわち、主記憶およびバッキングストアの多重度は固定とする。主記憶多重度は 8、バッチのバッキングストア多重度は 10、TSS のバッキングストア多重度は制限なしとする。スワップインまたはアウトのための CPU 時間は 50ms、I/O 時間は 133ms とする（但し、ページング機構なしとした場合の見積りである）。シミュレーションは TSS 頁荷（インタラクティブ数/秒）が 1.0, 1.5, 1.67, 2.0 の各場合について行う。シミュレーションは残りバッチ Xact 数が主記憶多重度 - 1 になるか、または 400 秒が経過すると終了するものとする。

表2. TSS の特性 \*分布は位相2のアラン分布

ジョブタイプ	ジョブ特性			応答性目標	
	平均総実行時間(ms)	平均I/O 間隔(ms)	発生率	目標応答時間(sec)	目標未達成率(%)
TSS1	520	16	0.53	5	5
TSS2	2000	5	0.15	10	10
TSS3	3500	17	0.07	15	10
TSS4	650	50	0.25	5	10

シミュレーションで比較評価したアルゴリズムはデッドライン(DL)方式と固定優先順序方式である。固定優先順序方式ではバッチジョブに対して CPU 時間 1000ms のメモリ・タイムスライティングを行う。TSS ジョブに関してはタイムスライスなしの方がよい応答性が得られたので、なしとした。優先度はバッチ Xact よりも TSS Xact を高くしている。DL 方式では GP1~3 を L3, 4~5 を L2, 6~8 を L1 とし、DRB は GP7 に設定した。デッドライン制御では目標応答時間の 1/4 経過するごとに GP を 1 つずつ上げていく。バッチ Xact には 1~5 の間の GP を許している。

なお、両方式とも、バッチ Xact の主記憶多重度に制限を設け、その制限を越えてはバッチ Xact はスワップインできないとする。また、今回のシミュレーションではフィードバック制御は、シミュレーション結果を見てパラメタ値を修正することにより、オフラインで行われている。

## 4. 評価結果

### 4.1 レスポンス多重度

レスポンス多重度の下限値、上限値を種々変えてみて、その効果を調べた<sup>3)</sup>。ここで、各 Xact に割当てた GP 値から、TSS はすべて L2 あるいは L1 レベルにあるが、バッチのほとんどは L3 にある。従って、レスポンス多重度 = TSS 多重度、としてよい。

#### (1) レスポンス多重度の下限値について

まず、TSS 頁荷が 1.67 の場合で、TSS 頁荷に下限値 (= 3) を設ける場合と、設けない場合 (下限値 = 0) を比較した。アルゴリズムは DL 方式である。表 3 の結果から、下限 = 3 の場合は各タイプとも応答性目標が達成されているが、下限 = 0 の場合は TSS 4 を除いて未達成である。これは、下限を設定することにより TSS の平均主記憶待ち時間を小さくできるためである (1.82 秒 → 1.4 秒に減少)。また、スループット (CPU, ファイル用ディスクの同時稼働数平均) も % 程度よくなっている。これは下限を設定することにより、バッチのスワップアウト回数が減少するためである。

次に、TSS 頁荷が比較的低い 1.0 の場合で、TSS 多重度の下限 = 3 の場合と下限 = 2 の場合とを比較した。アルゴリズムは DL 方式である（但し、DRB は動かさず、ディスクロッキングはすべて APG とする）。表 3 の結果から、下限

表3. レスポンス多重度に関するシミュレーション結果

評価項目	方式	TSSの4つの範囲	TSS負荷	TSS, Batchの多重度	平均応答時間(sec)				応答未達成割合(%)				平均MM多重度		TSSの平均MM待ち(sec)	スワップアウト数	スルーポイント
					TSS1	TSS2	TSS3	TSS4	TSS1	TSS2	TSS3	TSS4	バッチ	TSS			
下限	DL	4-7 5-7 5-7 4-7	1.67	T≥0, B≤8	3.1	6.2	9.9	3.5	6	15	17	10	3.9	3.7	1.8	84	3.74
					T≥3, B≤5	2.7	5.3	9.3	3.2	3	8	10	7	3.9	3.5	1.4	48
	DL (APGのみ)	4-6 4-7 4-7 6-8	1.0	T≥3, B≤5	1.6	4.5	8.0	2.5	1	7	4	7	4.7	2.0	0.6	61	3.99
					T≥2, B≤6	2.0	5.2	7.9	2.9	2	9	4	6	5.3	2.1	1.0	84
上限	DL	4-7 5-7 5-7 4-7	1.5	3 ≤ T ≤ 5	2.5	5.1	8.5	3.0	4	10	3	8	4.4	2.8	1.4	33	4.02
				3 ≤ T ≤ 6	2.4	4.9	7.8	2.9	2	6	6	4	4.4	2.9	1.2	38	3.99
				3 ≤ T ≤ 8	2.4	4.9	7.9	2.8	2	4	5	3	4.2	3.0	1.1	46	3.97

がえ、3いづれの場合も、応答性目標はすべて達成されている。一方、スルーポイントは下限=2の方が3の場合よりも5%程度向上している。これはバッチの平均主記憶占有ジョブ数が下限=2の時5.3で、下限=3の時の4.7より大きいためと思われる。下限値は応答性目標が達成可能な範囲で、できるだけ小さく設定するのが望ましいと言える。

以上から、レスポンス多重度の下限を設けることは、応答性、スルーポイントの両面にとって有効であることが明らかとなった。

(c) レスポンス多重度の上限値について

TSS負荷が1.5の場合に、TSS多重度の上限値を8, 6, 5と変化させた。いづれの場合も下限値=3で、アルゴリズムはDL方式である。

表3に示す結果から、上限を小さくするに従いTSSの平均主記憶待ち時間は延びるが応答性目標はすべて達成されている。これはGI値がDR3以上になるとディスクパッチングで優遇されるため、主記憶滞在時間が僅かながら減少するためと思われる。一方、上限を小さくしてもスルーポイントの向上は僅かである。これは上限値が小さい程TSSがディスクパッチングで優遇される割合が高くなるためと思われる。応答性の面からは、上限を小さくするには限界があると言える。

4.2 メモリ・タイムスライス

メモリ・タイムスライスに関するシミュレーション結果を表4に示す。アルゴリズムは固定優先順方式、TSS多重度の上限は8、下限は3である。

TSSはタイムスライシングなしとした場合を見ると、バッチのタイムスライスを短くするに従い、TSSの平均主記憶待ち時間は減少し応答性も向上する(ケース③~⑤)。すなわち、タイムスライスの長さにより優先度の異なるMacの間の優遇度合いの差を制御できると言える。一方、ケース⑥~⑨の結果から、タイ

表4. メモリ・タイムスライスに関するシミュレーション結果(固定優先順方式)

ケース	①	②	③	④	⑤	⑥	⑦	⑧	⑨
タイムスライス長 (TSS) (ms)	(300) (1000)	(600) (1000)	(∞) (1000)	(∞) (2000)	(∞) (∞)	(∞) (1000)	(∞) (600)	(∞) (1000)	(∞) (600)
優先度	TSS1 > TSS2 = TSS3 > TSS4 > バッチ					TSS4 > TSS1 > TSS2 = TSS3 > バッチ			
TSS負荷	1.5					1.5		2.0	
TSSの平均MM待ち(sec)	1.58	1.09	1.03	1.50	2.25	1.21	1.30	2.26	1.91
スワップアウト数	TSS	128	45	0	0	0	0	0	0
	バッチ	59	63	66	31	0	61	102	47
スルーポイント	3.77	3.94	3.99	3.92	4.14	3.92	3.84	3.73	3.69

ムスライス長はTSS負荷に応じて適切に決める必要があると思われる。ケース⑥～⑦はタイムスライスを短くしても主記憶待ち時間が短くならない場合を示す。これはタイムスライスが短い程スワッピング回数も増加するためと思われる。

次に、バッチのタイムスライスを固定(1000ms)し、TSSのタイムスライスを変化させた場合(ケース①～③)は、タイムスライスが短くなる程TSSのスワッピング回数が増え、TSSの主記憶待ち時間の増加、応答性の低下が見られる。応答要求の厳しいXacはなるべく1回のタイムスライス内に終了するようにタイムスライス長を決める必要があることを示している。

以上のことから、固定優先順方式では、メモリタイムスライスは応答性の制御に不可欠であり、その長さは種々の要因を考慮して適切に決める必要があると考える。一方、DL方式の場合、シミュレーションではメモリ・タイムスライスを導入する必要はなかったが、実際の場面においては、Xacの走行時間の分散が大きい応答クラスも存在すると考えられ、特に走行時間の長いXacが他に悪影響を及ぼさないようにメモリ・タイムスライスを導入する必要があるとされている。但し、この場合はタイムスライスの長さについては神経質になる必要はない。

#### 4.3 デッドライン制御の効果

デッドライン制御の導入により、次のような効果が認められた<sup>4)</sup>

(a) きめ細かな遅れの制御が可能

各応答クラスに割当てるGP値の範囲を制御することにより、その応答クラス自体のスワッピング、あるいはディスクバッチにおける優先のされ方と共に、他の応答クラスとの優先順位の相対関係も、きめ細く制御できる。

例えば、スワッピング・レベルについてみれば、図4において、固定GP方式では常にL2レベルにする(①)か、L1レベルにする(④)か、の選択しかできないのに対して、DL方式ではこれらの中道として、当初はL2レベルを割当て、目標応答時間の1/2あるいは1/4が経過するとL1レベルになるような割当て方(②, ③)もできる。これにより、主記憶待ち時間のきめ細い制御が可能となる。

応答クラス毎の遅れの配分に関しても、スワッピング・レベルについて見ると、固定GP方式の場合にはGPの割当てについて等しいか、大か、小かの3通りの選択しかできないので、例えば図5に示すように、TSS4に対してTSS3よりも高いGPを与えると、TSS4では応答性目標が達成されるが、TSS3では未達成となってしまい(ケース②)、これ以上遅れの配分を調整することはできない。これに対して、DL制御の場合には、図6に示すように、例えば一方のGPを一定にし(①)、他方のGP値を変えることにより、後者について目標応答時間の1/4が経過した場合は前者よりも高優先に(②)、1/2までは低く、3/4以後は高く(③)等、種々の割当てが可能になる。

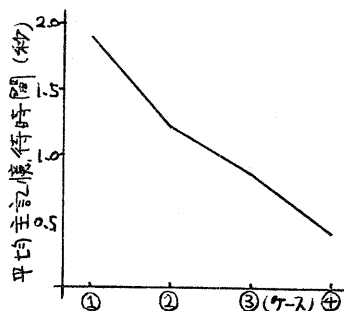


図4: GP値と平均主記憶待ち時間

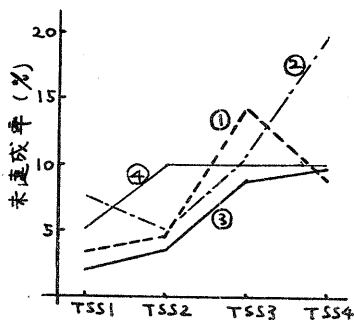
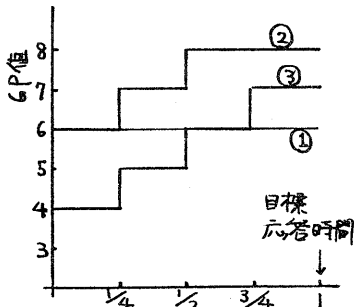
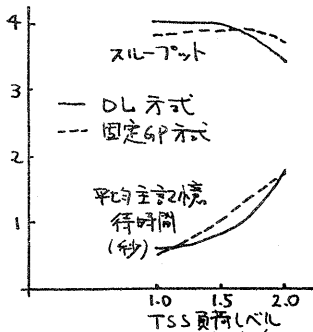


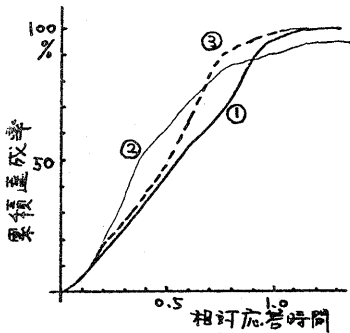
図5: 応答クラス毎の未達成率



オ6図: DL制御によるGP値の割合



オ7図: 負荷レベル変動の影響



オ8図: TSS4応答特性

これにより、応答クラス間での遅れの配分がきめ細く行えることになり、オ5図のケース③に示すように、TSS3とTSS4との適切な遅れの配分が実現できている。なお④は達成率目標を示している。

このように、各応答クラスにおける遅れ、応答クラス間の遅れの配分をきめ細く制御できることは、各負荷状況における目標応答の達成の度合いに応じてGPを制御するフィードバック制御が行い易いことを意味し、各応答クラスに対して、固定GP方式に比べて、応答性目標に近いレベルで過不足なしに目標を達成できることになる。

#### (b) 負荷変動に対する耐性

DL方式においては、応答要求の激しいジョブの負荷が低い状況においては、比較的低い優先順位にある間に処理が完了するのでスルーフットへの悪影響を小さくできる。また、負荷が高くなると処理を遅らせる $X_{act}$ が増え、これらは高い優先順位で処理されるので、スルーフットは低下するが応答時間はそれ程悪くならない、という作用が働くことが期待される。

例えば、オ7図に、固定GP方式とDL方式の場合で、制御パラメータは固定しておきTSS全体の負荷レベルを変動させた場合の平均主記憶待ち時間とスルーフットについて示してあるが、これにも、ある程度の負荷レベルまではスルーフットを犠牲にしても一定の目標応答時間を確保しようという傾向がみられる。ただし、負荷レベルがある程度以上高くなると、スワッピングの増大等のために急激に遅れが大きくなっている。

これからも、DL方式の場合には同一制御パラメータ値でカバーできる負荷範囲が広く、従って、フィードバック制御が行い易いということが言える。

#### (c) 応答特性の改善

DL制御においては、処理が遅延された $X_{act}$ はGP値が上昇され、投入されて間もない $X_{act}$ に比べて優先的に処理が行われることとなる。このために、不必要に早く完了する $X_{act}$ は少く、また極端に遅らされる $X_{act}$ も少くなることが期待される。

オ8図に、TSS4の応答特性を示す。相対応答時間とは目標応答時間に対する比を表わす。TSS4はCPU処理が中心であり、ケース①の場合には、目標応答時間の3/4を経過するとDL制御によりGP値がDRBを越え、優先的なディスクパッチング・プライオリティが割当てられて処理が加速され、以後終了する $X_{act}$ が急激に増加していることが示されている。

②は固定GP方式でディスクパッチング方式を常にAPGとした場合である。DL方式で、ディスクパッチング・レベルでは常にAPGを適用した場合にも、応答



特性は②の場合と同様で、①のような顕著な傾向は見られず、スワッピング・レベルの制御のみでは同一応答クラス内の応答特性改善の度合いは小さいと解る。

(d) スループットの向上

DL制御の場合には、(a)に示したように各応答クラスに対して遅れをきめ細く配分できるので、達成率目標に近い値に達成率を制御し得る。これに対して固定GP方式の場合には、同一応答クラスで応答目標を達成しようとする時、あるクラスでは過達成にせざるを得ないようなケースが多く、これがスループットに悪影響を与える。

また、(c)にも示したように、DL制御の下では、同一クラス内においても過応答の $X_{act}$ が少なくなっており、これもスループットの向上に貢献すると考えられる。

図9に、各TSS負荷レベルにおいて、目標応答を達成するように制御パラメータ値を選択した場合のスループットを示す。固定GP方式に比べてDL方式の方が10%程度高いスループットが得られている。

(e). DL制御回数を変えた時の影響

以上で得られた結果は、目標応答時間を4分割し、1/4の時間が経過する毎にGP値を1ずつ上昇させて得られている(制御回数3)が、目標応答が短い場合、あるいは負荷が大きい場合にはDL制御のためのタイム割込み回数が増大し、オーバヘッドの観点からはDL制御回数を少なくすることが望ましいと考えられる。一方、先に上げたDL制御の効果という観点からはきめ細かくGP値を上昇させる方が望ましいと考えられる。

図10に、DL制御回数を減らして目標応答の1/2の時間が経過するとGP値を上昇させた場合(制御回数1)におけるスループットを、固定GP方式(制御回数0)、制御回数3の場合と共に示してある。制御回数3の場合には固定GP方式の13%増であるのに対して、制御回数1の場合には10%増で、制御回数を減らすことによりスループットは僅かに減少するが、DL制御を行わない場合と比較してかなりよい値を示していることがわかる。

また、図8③に応答特性を示すように、目標応答の1/2経過後に完了する $X_{act}$ が急激に増大しており、固定GP方式に比べて良い応答特性を示している。これらより、制御回数を少なくしても、DL制御を導入する効果は大きいと言えるであろう。

#### 4.4 個別制御の効果

個別制御の導入により、処理時間が長いことがわかっている $X_{act}$ には予め高いGP値が割当てられて他の $X_{act}$ に比べて処理が優先され、また、処理時間の短い $X_{act}$ は他の $X_{act}$ よりも低いGP値が割当てられて処理が遅らされる。これによって、応答特性はさらに改善されると期待される。

図11に、負荷レベル2.0/秒において個別制御を導入した場合(ケース①)

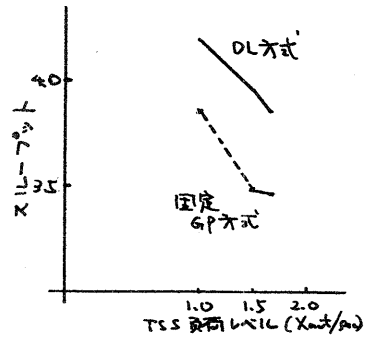


図9: 負荷レベルとスループット

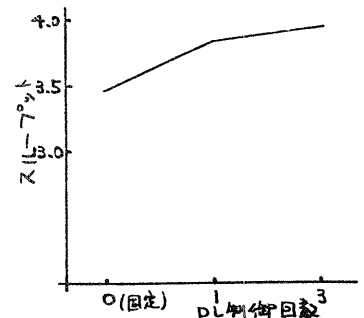


図10: DL制御回数とスループット

としない場合(ケース②)における応答未達成率を示す。個別制御の導入により改善されていることが示されている。

また、応答特性の改善は純走行時間(CPU+I/O時間)と目標応答時間の比が最も高いTSS3において最も顕著にあらわれており、その図に示すように目標応答時間の50%から100%までに集中してXacも完了している。

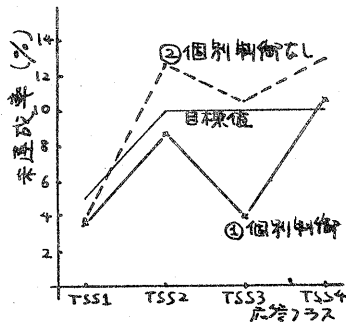


図1: 個別制御導入の効果

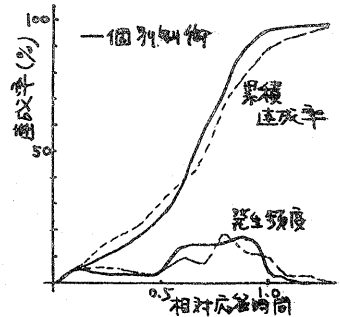


図2: 個別制御導入の効果

## 5. おわりに

大型コンピュータ・システムを対象とした動的資源管理方式の検討について、応答性管理を中心としてその方式と評価結果とを報告した。提案した応答性管理方式では管理の階層化を行っている。応答性に対する要求としては、直接的に目標応答時間を指定して貰う。これを各ジョブの各時点における応答要求を表現する内部パラメータ、GP、に変換しGPの制御によって目標の実現を図る。GPの制御にはデッドライン制御、フィードバック制御、個別制御を導入する。GPを反映した具体的な制御階層としてはスワッピング、ディスパッチングがあり、前者においてレスポンス多重度、メモリ・タイムスライスの特徴が導入される。

応答性管理の評価は、目標が多分に定性的なものであって困難を伴うが、ここでは上述の管理方式の個々の制御の効果について、応答性目標の充足性と、その条件下でのスループットの高さを指標としてシミュレーションにより評価した。特に、デッドライン制御、レスポンス多重度による制御は、ユーザの望む応答性をきめ細く実現する上で非常に有効であることが確認された。

ここで述べた応答性管理方式は、ACOS-4/MVPの動的資源管理の一部として、段階的に実装されていく予定である。

最後に、日頃御指導頂いている当社中央研究所藤津部長、箱崎課長、ACOS-4への適用につき御協力頂いている基本ソフトウェア開発本部藤野本部長、澄家部長、辻課長、鈴木主任、片岡氏、また、本方式の検討に協力して頂いた宇宙開発事業部小野氏、中研佐々木氏、NTS(株)増子氏、の各氏に感謝いたします。

## 参考文献

- [1] 久保, 他, オペレーティング・システムの動的資源管理方式の評価, 情報システム性能評価研究(1978.3).
- [2] 久保, 大野, 多次元システムにおけるCPU配分管理の検討, 情報システム全国大会, 99 (1979.7).
- [3] 土井根, 他, レスポンス多重度による主記憶ジョブミックス制御の評価, 同上, 983 (1979.7).
- [4] 大野, 他, 動的システム資源管理におけるデッドライン制御の効果について, 同上, 985 (1979.7).
- [5] 水野, 日井, コンピュータACOSシリーズの概観, 日経エレクトロニクス no.107, 82-135 (1975.5-5).
- [6] 日本電気, ACOSシリーズのNEACシステム800モジュールシステム概説書, PAZ 51-1 (1978.7).
- [7] H.W. Linch, J.B. Page, The OS/VS2 Release 2 System Resources Manager, IBM Syst. J., 13, 4 (1974), 294-291.