オペレーティング・システム　39－2
（１９８８．６．１７）

# 仮想計算機システムの運転方式について

加藤　保夫　　川崎　隆二
ＮＴＴソフトウェア研究所

仮想計算機システムの運転性を改善する方法として、限定優先方式による運転方式について提案する。
仮想計算機システム上で並走する多数の仮想計算機の中で特定の仮想計算機、例えばオンラインシステムを搭載した仮想計算機にＣＰＵ時間を優先して割り付けることにより、他仮想計算機に比べスループットを向上することができる。既存のＦＩＦＯ制御を用いて優先制御を行う場合、ＣＰＵ負荷の大きい仮想計算機が優先指定されると優先度の低い仮想計算機の沈み込みという問題が発生する。これらの問題点を改善するため、限定優先方式ではＣＰＵ時間を取り上げる機能と優先度を制限する機能を設けた。
本報告では限定優先制御方式について紹介するとともに、本方式の評価結果を示す。

## An Operating Method for the Virtual Machine System

Yasuo KATO　and　Ryuji KAWASAKI
NTT Software Laboratories, NTT
1-9-1 Kohnan Minato-ku Tokyo 108 Japan

In this paper a new virtual machine system operating methodology is proposed. The methode is named "Limited Priority Scheduling Methode(LPSM)".

When various operating systems run parallely on a virtual machine system, previously it was difficult to get high throughput for individual operating system. Traditional FIFO control mothode causes "under schedule problem", in which low priority operating system is suppressed to get CPU resource by high priority operating system. The LPSM resoleved this problem by controlling each operating system priority.

This paper presents the mechanism of LPSM and its evaluation.

## １．はじめに

従来仮想計算機システム（ＶＭシステム）は、ＶＭシステムが実現する仮想計算機（ＶＭ）の性能的な問題から主に試験システムを搭載していた。しかしＶＭを高速に動作させるためのハードウェアアシスト機構を備えた計算機の開発により、ＶＭの性能は飛躍的に向上してきている。一方ハードウェアの性能も毎年向上してきており、ＶＭシステムを搭載しても性能的な問題はすくなくなっている。このような状況から従来プログラムの試験のみに利用していたＶＭシステムをバッチサービス用更にはオンラインサービス用のシステムを搭載し利用できるようになった。

ＶＭシステム上に試験システム、バッチシステム、及びオンラインシステムを混在して搭載し運転する場合、搭載システムの特性上の問題の１つとして試験システムのプログラムバグによるループでＣＰＵ時間が寡占され、他のＶＭの沈み込みが発生することが考えられる。一方ＶＭシステムの運転方式として、搭載するシステムのサービス条件から特定ＶＭに優先的にＣＰＵ時間を割り当て、当該ＶＭのスループットを向上させる方式が考えられる。これにより例えばオンラインシステムの端末での応答時間を改善することができる。この場合当該ＶＭの負荷が高くなった時他の優先されていないＶＭが沈み込まないように、各ＶＭにＣＰＵ時間を割り付ける方法が必要となる。

本論文では、上記のような各種システムが混在して搭載したＶＭシステムの運転性向上のためのスケジュール方式として、ＶＭの沈み込み等の問題を解決した限定優先制御方式について一案を紹介し、その評価結果を示す。本限定優先方式を用いることにより特定の仮想計算機のスループットの向上を図ると共に、仮想計算機の沈み込みも防止することができ、仮想計算機システムの運転性向上を図ることができる。

## ２．ＶＭの利用形態モデルについて

### ２．１　搭載システムの特性とサービス条件

検討を進めるにあたりＶＭシステムに搭載するシステムを試験システム、バッチシステム、オンラインシステムに分類し、各システムの特性とサービス条件をＣＰＵ使用率の観点から表１のように整理した。

表１　搭載システムの特性とサービス条件

| 搭載システム | 搭載システムの特性とサービス条件 |
|---|---|
| 試験システム | 作成したプログラムの試験を行うためのシステムである。実行するジョブとしては、試験環境の作成、試験プログラムの入替え、試験プログラムの走行、試験結果の出力がある。通常状態ではＣＰＵ要求率は低いが、試験中であることからトレース等のＣＰＵ時間を必要とするデバグ機能の利用、プログラムバグによるループ状態があり一時的にＣＰＵ時間を寡占することがある。ＣＰＵ時間の割当の観点からは優先してＣＰＵ時間を割り当てる必要性はない。また他のシステムがＣＰＵ時間を必要としている場合は当該システムのＣＰＵ時間を犠牲にして良い。 |
| バッチシステム | バッチ処理サービスを提供するシステムである。長時間のジョブ、ＣＰＵ使用率の高いジョブを走行させることから、ＣＰＵ使用率は中程度から高程度となる。ＣＰＵ時間の割当の観点からは優先してＣＰＵ時間を割り当てる必要性は少ない。しかし他の優先度の高いシステムがＣＰＵ時間を必要としていても、当該システムにはある一定のＣＰＵ時間を確保する必要がある。 |
| オンライン<br>システム | オンラインサービスを提供するシステムである。同時接続の端末台数とそのトラフィックによりＣＰＵ使用率は変わるが、ＣＰＵ使用率は中程度と考えられる。オンラインサービスであることから端末に対する応答時間が短い必要がある。即ち当該システに対して優先的にＣＰＵ時間を割り当てる必要がある。 |

### ２．２　ＶＭスケジュール方式の考え方

ＶＭスケジュール方式の検討に当たり、前述の搭載システムの特性とサービス条件から以下の設計条件を設定した。

① オンラインシステムへの適用が可能なこと。
② 各搭載システムのＣＰＵ使用率を容易に制御可能なこと。
③ 搭載システムの沈み込みが回避可能なこと。

ＶＭシステムでは搭載システム対応にＶＭを割り当てている。搭載システムの沈み込みとは、ＶＭが要求するＣＰＵ時間に対して、ＶＭシステムのスケジュール機能が当該ＶＭに対してＣＰＵ時間を割り当てないために搭載システムの実行が非常に遅くなる現象である。例えば特定ＶＭが優先的にスケジュールされる運転方式の場合、当該ＶＭのＣＰＵ使用率が非常に高くなると優先度の低いＶＭの実行が遅くなる。

当該問題はＣＰＵ負荷の制御の問題として捉えることができることから、オペレーティングシステム（ＯＳ）のジョブスケジュールについても類似の問題がある。ＯＳのジョブは一定の時間で生成／消滅を繰返しており、ジョブ内でのＣＰＵ使用率の変動が大きくないことから、ＯＳはジョブの開始を抑止することにより負荷制御することができる。しかしＶＭシステムでは、以下の理由からＶＭの起動を抑止し動的に負荷制御することは困難である。

① オンラインシステム／バッチシステム用のＶＭではＶＭシステムの運転開始／終了と同期して起動／停止を行う。
② 試験システム用のＶＭでは利用者が利用するかなり長い時間を単位として起動／停止を行う。
③ 搭載システムによるが、ＣＰＵ使用率の変動が大きい。

## ３．ＶＭスケジュール方式案

ＶＭに対するＣＰＵ時間のスケジュール方式として以下の方式案が考えられる。

(1)平等方式（ＦＩＦＯ方式）

レディ状態の全ＶＭの実行をＦＩＦＯにより制御する方式であり、各ＶＭの動作特性が同一の場合各ＶＭのＣＰＵ使用率は平等になる。本方式の適用方法としては、全ＶＭに試験システムを搭載している場合のように各ＶＭへ平等にＣＰＵ時間を割り当てる場合が考えられる。ＶＭの沈み込みという問題は発生しないが特定ＶＭを優先して制御することはできない。

(2)クラス優先方式（多段ラウンドロビン方式）

ＯＳのジョブ管理等で広く実施されている方法であり、各ＶＭをクラスで分類し、レディ状態のＶＭのうち優先度の高いクラスのＶＭを実行し、同一クラス内のＶＭ間はＦＩＦＯにより制御する方式である。特定ＶＭを優先して実行する方法として有効であるが、優先度の高いクラスのＶＭのＣＰＵ使用率が高いとＶＭの沈み込みの問題が発生する。また優先度の高いクラスのＶＭがウェイト状態になると優先度の低いクラスのＶＭが走行するので、当優先度の低いクラスのＶＭがタイムスライスまでＣＰＵ時間を使用する場合、優先度の高いクラスのＶＭはＣＰＵ時間を沢山使用することはできない。従って端末への応答時間の条件がきびしいオンラインシステムを搭載するＶＭシステムには不適当である。

(3)限定優先方式

本論文において提案する方式であり、クラス優先方式に対して応答時間を改善するためＣＰＵ時間を取り上げる機能と、ＶＭの沈み込みを回避するため優先度を限定する機能を付加した方式である。優先度の限定はクラス毎設けた優先カウンタにより制御する。優先カウンタの最大値は優先回数（Ｎ）と非優先回数（Ｍ）を合計した値である。なお優先回数の割合（＝Ｎ／（Ｎ＋Ｍ））を優先率という。スケジュールはクラス優先方式と同様に優先度の高いクラスのＶＭから検索し、当該クラスの優先カウンタが優先範囲の場合スケジュールする。非優先範囲の場合はスケジュールしない。さらに優先度の低いクラスのＶＭが走行中に優先度の高いクラスのＶＭがレディになった時、優先度の高いクラスの優先カウンタが優先範囲であるなら走行中の優先度の低いクラスのＶＭを停止させ優先度の高いクラスのＶＭを走行させる。これをＣＰＵ時間の取り上げという。優先範囲とは優先カウンタの値が１～Ｎのときであり、非優先範囲とはＮ＋１～Ｎ＋Ｍのときである。優先カウンタは、①当該クラスのＶＭがスケジュールされた時、②下位のクラスのＶＭがスケジュールされた時、更新される。優先度の高いクラスの優先率を１にすることにより当該クラスのＶＭはレディになった時に即時に走行することになり、応答時間を改善することができる。ＶＭの沈み込みは優先率を０から１の範囲で設定することにより防ぐことができる。各クラスのＶＭが常にレディ状態の場合の限定優先方式のスケジューリング例を表２に示す。

各スケジュール方式の比較結果を表３に示す。

表２　限定優先方式のスケジューリング例

| フェーズ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| クラス１ | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| クラス２ | | | 1 | | | 2 | | | 3 | | | 1 | | | 2 | | | 3 |
| クラス３ | | | | | | | | | 1 | | | | | | | | | 2 |

| フェーズ | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| クラス１ | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| クラス２ | | | 1 | | | 2 | | | 3 | | | 1 | | | 2 | | | 3 |
| クラス３ | | | | | | | | | 3 | | | | | | | | | 4 |

（凡例）数字：優先カウンタ　　網掛け有：該クラス走行　　網掛け無：該クラス非走行

（条件）

| クラス | 優先回数 | 非優先回数 |
|---|---|---|
| クラス１ | 2 | 1 |
| クラス２ | 2 | 1 |
| クラス３ | 3 | 1 |

表３　スケジュール方式の比較結果

| 要求条件 | スケジュール方式 | | |
|---|---|---|---|
| | (1)平等 | (2)クラス優先 | (3)限定優先 |
| オンラインシステムへの適用 | × | △ | ○ |
| ＣＰＵ使用率の制御の容易性 | × | △ | ○ |
| ＶＭの沈み込みの発生 | ○ | × | △ |

## ４．評価方法と結果

各方式についてシミュレーションにより評価した。

(1)評価モデル

各ＶＭはスケジューリング時に指数分布に従ってＣＰＵ時間を使用し、ＶＭモニタによりタイムスライス切れとなったときには直ちにレディ状態となり、次にスケジューリング時はまた指数分布に従ったＣＰＵ時間を使用する。タイムスライス切れとなる前にＣＰＵ時間を放棄した場合は、指数分布に従った時間経過後にレディ状態になる。ＣＰＵ時間を放棄するのは周辺装置との入出力処理のためであるが、ここでは周辺装置との入出力時の競合は無いと仮定した。

(2)ＶＭシステムの動作特性の測定法

各ＶＭが動作するときには、ＶＭの走行時間に応じて①ＶＭ走行に伴うＶＭモニタの処理分、②ＶＭモニタのプロセス処理分、③割込処理の非タスク処理分の各オーバヘッドが発生する。①については当該ＶＭのＣＰＵ時間としてカウントした。②については全てのＶＭより高い優先クラス（クラス０）で動作する独立した処理としてカウントした。③についてはＶＭ走行と同期して余分に時間が経過したとして処理した。

(3)シミュレーション条件

ＶＭの優先クラスは、オンライン用、バッチ用及びデバグ用の３クラスとし、表４のシミュレーション条件を仮定した。ここでＣＰＵ要求率とは当該ＶＭが走行する時に必要とするＣＰＵ時間の割合であり、競合がない時はＣＰＵ要求率がＣＰＵ使用率となる。クラス１はオンライン用であることから優先率を大きくし、クラス２とクラス３はケース１のＣＰＵ要求率を考慮して決定した。ケース１は定常時である。ケース２はトラブル等によりデバグ用ＶＭのＶＭ０５が高負荷時であり、ＣＰＵ要求率が１００％となり、ＶＭシステムが過負荷となった状態である。

(4)シミュレーション結果

以下の場合についてシミュレーションを行い応答時間とＣＰＵ取上率を求めた。

図１＝＝＝ケース１、クラス１のＣＰＵ要求率を変化
図２＝＝＝ケース２、クラス１のＣＰＵ要求率を変化
図３＝＝＝ケース１、クラス１の優先率を変化、クラス１のＣＰＵ要求率＝２０％
図４＝＝＝ケース１、クラス１の優先率を変化、クラス１のＣＰＵ要求率＝８０％
図５＝＝＝ケース２、クラス１の優先率を変化、クラス１のＣＰＵ要求率＝２０％
図６＝＝＝ケース２、クラス１の優先率を変化、クラス１のＣＰＵ要求率＝８０％

応答時間はＣＰＵ使用率をもとに算出した。表３のＣＰＵ要求率はＣＰＵ使用率の期待値であり競合のない状態でのＣＰＵ使用率である。ＶＭシステム上で走行した場合、他ＶＭとの競合及びＶＭモニタとの競合のためＣＰＵ使用率はＣＰＵ要求率より小さくなる。すなわち競合のない状態での応答時間を１とすれば競合時の応答時間は（ＣＰＵ要求率／ＣＰＵ使用率）となる。

ＣＰＵ取上率とは限定優先方式において発生する状態であり、全スケジューリング回数に対するＣＰＵ時間の取り上げによりスケジューリングされた割合である。ＣＰＵ取上率が高いということは、当該クラスが優先されて実行している指標になる。

表４　シミュレーション条件

| 優先クラス | ＶＭ識別詞 | 優先制御 | | ＣＰＵ要求率（％） | | 記　事 |
|---|---|---|---|---|---|---|
| | | 優先回数（Ｎ） | 非優先回数（Ｍ） | ケース１（定常時） | ケース２（高負荷時） | |
| クラス０（ＶＭモニタ用） | － | １ | １ | － | － | |
| クラス１（オンライン用） | ＶＭ００ | ９ | １ | 変化 | 変化 | 図１、図２ |
| | | 変化 | | ２０ | ２０ | 図３、図５ |
| | | 変化 | | ８０ | ８０ | 図４、図６ |
| クラス２（バッチ用） | ＶＭ０１ | ５ | ２ | ３５ | ３５ | |
| クラス３（デバグ用） | ＶＭ０２<br>ＶＭ０３<br>ＶＭ０４<br>ＶＭ０５ | １０ | ０ | ５<br>５<br>５<br>５ ］２０ | ５<br>５<br>５<br>１００ ］１１５ | |

５．考察

(１)応答時間の改善効果について

図１、図２の結果から限定優先方式では他の方式に比べクラス１の応答時間が期待値に近いことから、オンラインシステムの応答時間の改善に限定優先方式は有効である。特にクラス３のＣＰＵ要求率が高くなった時には有効である。

(２)ＣＰＵ使用率の制御について

平等方式と比較した場合、限定優先方式ではクラス１のＣＰＵ要求率が増加するに従ってクラス２と３の応答時間が非常に大きくなる。これはクラス１がＣＰＵ時間を取り上げたためである。またクラス優先方式と限定優先方式では、クラス２の応答時間が大きく異なる。これは限定優先方式では優先率によりクラス２の走行が制約されているためである。従って各クラスのＣＰＵ使用率が優先率により制御できることが分かる。

(３)ＶＭの沈み込みについて

図１と図２において、限定優先方式のクラス３の応答時間はクラス優先方式より僅かに大きい。すなわち限定優先方式がクラス３の沈み込みが大きいことになる。原因としては、①クラス１の優先率が大きすぎる、②ＣＰＵ取上機能が影響しクラス２と３のＣＰＵ時間がクラス１に取り上げられたためと考えられる。

(４)ＣＰＵ取上率について

図１と図２からクラス１のＣＰＵ要求率が増加するに従って、ＣＰＵ取上率は減少している。これはクラス１のＣＰＵ要求率の増加に連れてクラス１のＣＰＵ使用率が増加すると他のクラスのＣＰＵ使用率が減少し、

取り上げ対象の優先クラスが走行しなくなるためである。従ってＣＰＵ要求率が１００％近くでは急に低下することになる。

図３から図６においては優先率の増加とともにＣＰＵ取上率が増加する。ＣＰＵ取上率の増加は応答時間の減少と対応しており、応答時間の改善にＣＰＵ取上機能が有効であることが分かる。優先率が１の場合のＣＰＵ取上率は、自分より下位のクラスが走行している確率になる。例えば図３では確率０．５５に対し取上率は０．５６となっている。また図５では確率０．８８（＝（３５＋１１５）／（２０＋３５＋１１５））に対し取上率は０．８７となっている。

(5)優先率の設定について

図３から図６の優先率を変化させた場合の応答時間から、優先率はＣＰＵ要求率を考慮して決定すれば良いことが分かる。図３と図４のケース１の定常時は優先率を変えてもクラス１の応答時間の変化は少ない。これはクラス１以外のＣＰＵ負荷が少ないためである。しかし図５と図６のケース２ではＶＭ０５の影響により優先率に依存して応答時間が大きく変化する。応答時間が１．３となるには、図５のクラス１のＣＰＵ要求率が２０％の場合は優先率０．４、図６のクラス１のＣＰＵ要求率が８０％の場合は優先率０．７である。優先率を大きくすればクラス１の応答時間は改善できるが、図４と図６からクラス１のＣＰＵ要求率が高いと他のクラスの応答時間がかなり大きくなることより、クラス１の優先率を適当な値にすべきである。

ＶＭシステムに各種のサービスを搭載した場合、オンラインシステムの応答時間を最も速くするには、オンラインシステムの優先率を１にすればよい。しかし他のシステムの応答時間は犠牲になることから、オンラインシステムの応答時間が例えば１．３倍程度になることが許容されるなら、優先率は０．７程度でよい（図６）。これにより他のシステムの応答時間もある程度保証できると考えられる。

また優先すべきシステムがなく全システムを同程度の応答時間にするには、各システムのＣＰＵ要求率の割合で優先率を決定する。

## ６．おわりに

今回の評価により限定優先方式についてある程度の解析結果が得られた。即ち優先率を設定することにより、①ＶＭシステムのオンラインシステムへの適応、②ＣＰＵ使用率の制御、が可能であることが確認できた。しかしＶＭの沈み込みの制御については十分な評価が得られなかった。この点についてはＣＰＵ取上機能を除いた限定優先方式について測定し評価する必要があると考えられる。

またシミュレーション結果として、①変化が異様な部分及び②ＣＰＵ要求率に対して僅かにＣＰＵ使用率が大きいという箇所が発見された。この点について問題の解析が進んでいないが、より精度の高いシミュレーションを行い評価結果を検証する予定である。

さらに優先率／ＣＰＵ要求率に対するＣＰＵ使用率は、ＶＭモニタのタイムスライス値、ＶＭ上のプログラムのラン時間／ウェイト時間とその分布特性により変化する。今後はこれらの影響についても含めて評価することとしたい。

【参考文献】

[1]加藤、川崎、「仮想計算機システムにおける優先制御方式について」、
情処第３６回全国大会（1988）、ＰＰ３５３－３５４
[2]池ヶ谷、他、「Ｍシリーズ仮想計算機機構（ＶＭ／ＥＸ機構）の開発－処理方式」、
情処第３６回全国大会（1988）、ＰＰ２２９－２３０
[3]池ヶ谷、他、「Ｍシリーズ仮想計算機機構（ＶＭ／ＥＸ機構）の開発－評価」、
情処第３６回全国大会（1988）、ＰＰ２３１－２３２
[4]小柳津、他、「ＤＩＰＳ－１１／５Ｅシリーズの実用化」、
ＮＴＴ研究実用化報告（1987）、Ｖｏｌ．３６、Ｎｏ．１、ＰＰ４９－５６
[5]塩川、他、「ＤＩＰＳ－１１／５Ｅシリーズ論理装置」、
ＮＴＴ研究実用化報告（1987）、Ｖｏｌ．３６、Ｎｏ．１、ＰＰ５７－６５

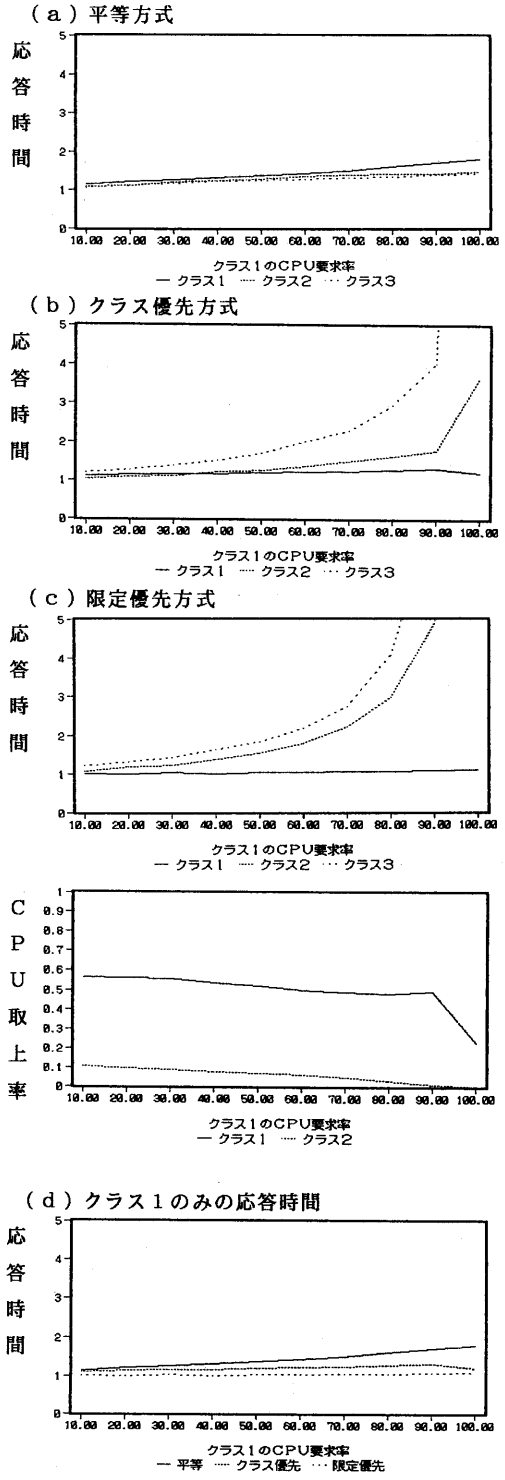


図１　ケース１、クラス１のＣＰＵ要求率変化

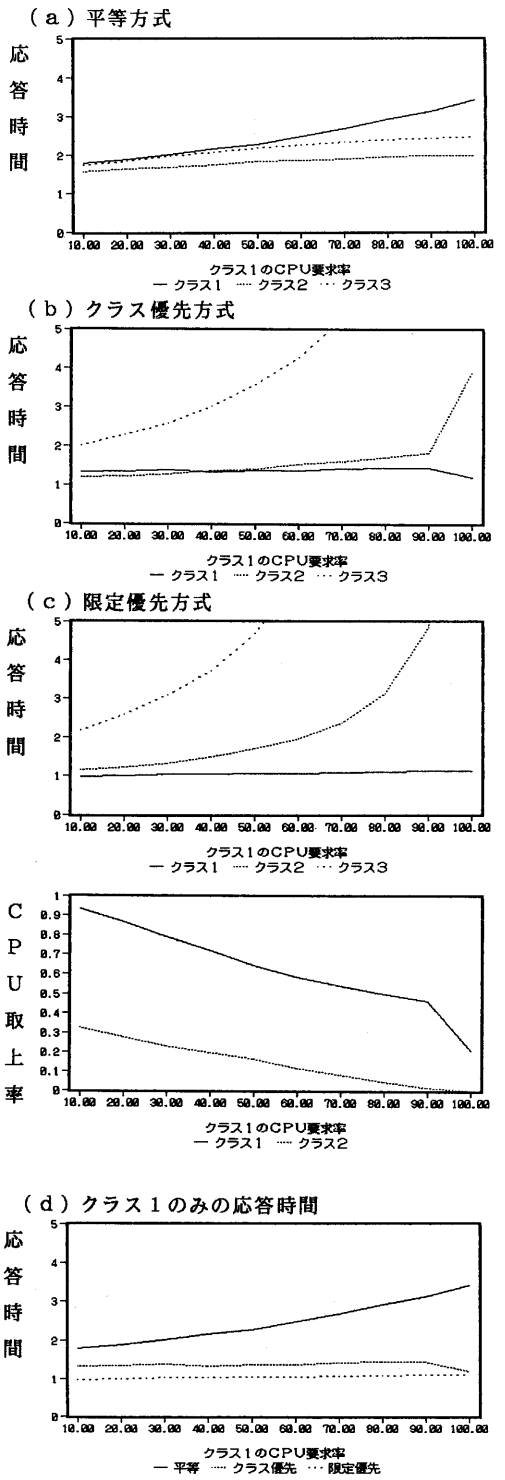


図２　ケース２、クラス１のＣＰＵ要求率変化

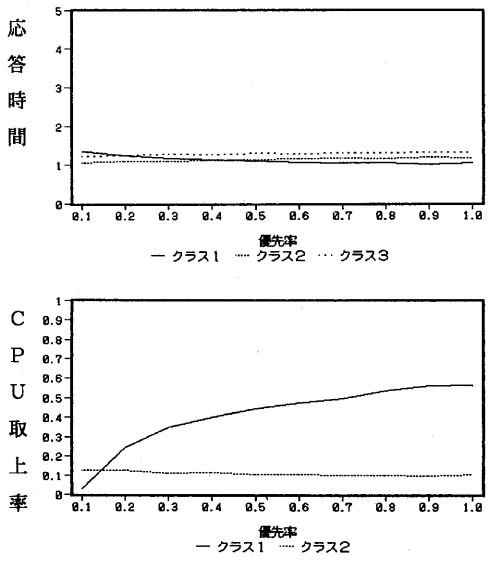


図3　ケース1、クラス1の優先率を変化、
クラス1のCPU要求率＝20％

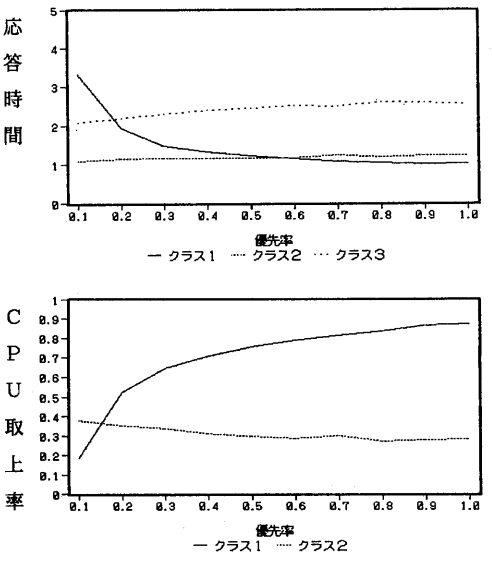


図5　ケース2、クラス1の優先率を変化、
クラス1のCPU要求率＝20％

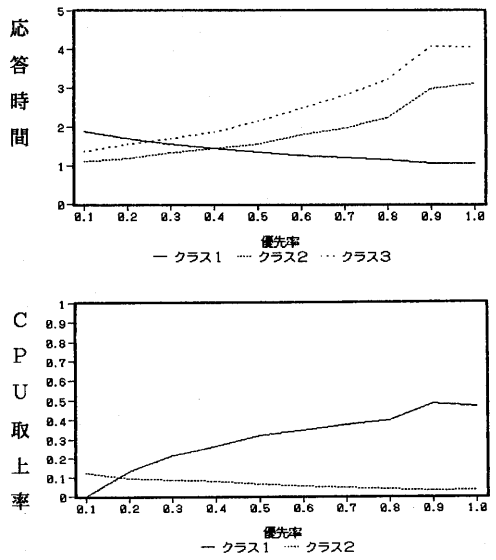


図4　ケース1、クラス1の優先率を変化、
クラス1のCPU要求率＝80％

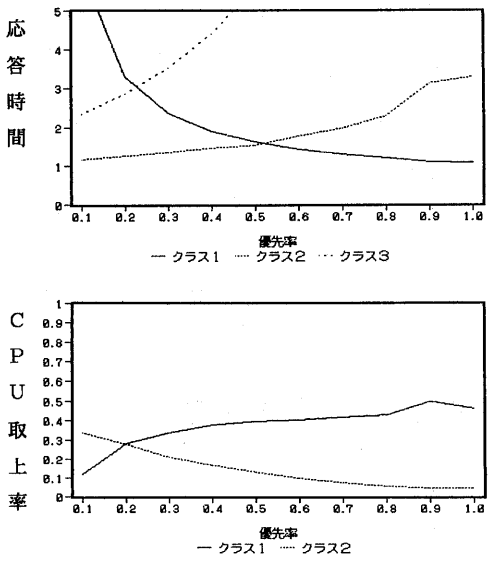


図6　ケース2、クラス1の優先率を変化、
クラス1のCPU要求率＝80％