ISSN 0389-4010
UDC 681.31

航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-1277

並列計算機システム用ジョブスケジューラ

末松和代

1995年9月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

目　　次

並列計算機システム用ジョブスケジューラ*

末松和代*1

Investigation on the Job Scheduler for Parallel Computer Systems

Kazuyo SUEMATSU

ABSTRACT

The Numerical Wind Tunnel, NWT, is a CFD-oriented vector parallel computer system with distributed memory. The development of a job scheduler is mandatory in order to operate the system effectively. Proposed herein is a scheduling algorithm which enhances the utilization ratio of processor elements, execution of jobs on the optimum number of processor elements, guarantees the turn-around time of jobs and operation of privileged jobs. The major point of the algorithm is to control job initiation and execution according to waiting time and demand for system resources of jobs. The effectiveness of the proposed scheduling algorithm is verified by various numerical experiments employing a scheduler simulator.

Key Word ; job scheduler, parallel computer system, waiting time, demand for system resources, scheduler simulator

概　　要

数値風洞（NWTシステム）は，計算空気力学プログラムの超高速処理を目的とした分散メモリ型の並列計算機システムである。このような並列計算機システムを効率的に運用するためにはジョブスケジューラの開発が必要である。本稿では，①要素計算機(PE)の高効率利用，②要求台数のPEの割り当て，③適切なターンアラウンドタイムの保証，および④特権ジョブの優先処理を実現することの可能なスケジューリングアルゴリズムについて提案する。本アルゴリズムの特徴は，ジョブの待ち時間およびシステム資源要求量に応じてジョブの起動や実行を制御している点にある。なお，スケジューリングアルゴリズムの有効性については，スケジューリングシミュレータを使用した種々の実験によって実証した。

1. はじめに

1980年代末，航技研では，近年急速に発展している計算流体力学（以下CFDと略記する）が今後の我が国の航空宇宙技術の研究開発の基盤技術における中核としての役割を果たすためには，メモリ量が32GB以上，CFDプログラムの実効処理速度がFujitsu-VP400の100倍以上の計算機が必須であり，その実現においてはMIMD（複数命令列，複数データ列）方式のメモリ分散型の並列計算機システムを採用すべきであると指摘されていた[1),2)]。これに基づき，上記性能をもつ並列計算機システムの実現を目指して，富士通(株)との間で共同研究を開始し，ハードウェアの構築，ハードウェア性能を十二分に発揮させるためのオペレーティングシステム機能，プログラム開発環境をより良くするための言語処理ソフトウェア機能等の検討・設計・開発を行った[3)-9)]。さらに，1993年2月には共同研究の成果とも言うべき数値風洞（以下NWTシステムと略記する）を導入し，新計算機システムとしての運用を開始した。本稿では，新旧の計算機システムを総称して数値シミュレータシステム（以下NSシステムと略記する）と呼び，区別が必

* 平成7年2月23日受付（received 23 February 1995）
*1 数理解析部（Computational Sciences Division）

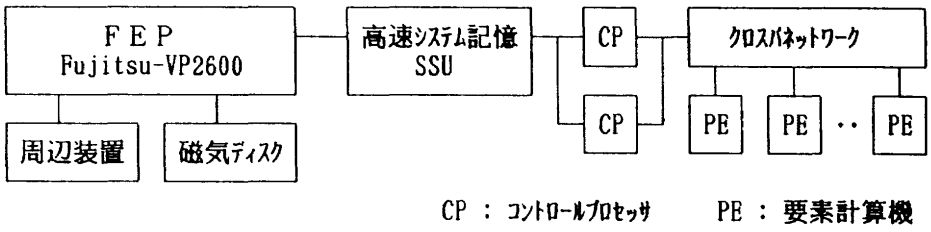


図1.1 NWTシステムのハードウェア構成図

要な場合には，従来の計算機システムを第一期NSシステム，新計算機システムを第二期NSシステムと呼称する。

図1.1にNWTシステムのハードウェア構成図を示す。NWTシステムは，演算処理用の140台の要素計算機（以下PEと略記する），ジョブおよび入出力制御用の2台のコントロールプロセッサ（以下CPと略記する），PE-CP間およびPE-PE間接続用のクロスバネットワーク，ジョブ実行に必要なファイル転送用の高速システム記憶（以下SSUと略記する）から構成されており，第一期NSシステムで使用していたFujitsu-VP2600をフロントエンドプロセッサ（以下FEPと略記する）として使用している。

NWTシステムは，複数ジョブの同時実行が可能であるため，PEの有効利用を図れるスケジューラが必要であったが，並列計算機用に開発されたそのようなスケジューラはなかった。また，並列計算機であるにも関わらず一般の汎用計算機用のスケジューラを使用した場合にはPE稼働率の低下が明白であった（第2.1(6)項に詳述する）ため，新たなアルゴリズムの検討を開始した。その結果，下記特徴を持つジョブスケジューリングアルゴリズムを考案することができた。

①任意のPE台数でのジョブ実行
②PEの高効率利用
③システム資源要求量に応じた適切なジョブターンアラウンドタイムの保証
④緊急ジョブ，特権ジョブの優先処理

なお，本アルゴリズムでは，ジョブ実行時のPE占有時間をCPU要求時間から予測する方式を採っているが，これにより，③項を実現する際のシステム資源利用率の大幅低下防止ばかりでなく，実行中ジョブの終了を待つためにシステム停止を予定時刻に行えないような事態を回避することも可能となった。

これらの機能は，航技研の計算機システムだけでなく，一般の並列計算機システムの運用においても十分有効な機能である。

本稿では，上記スケジューリングアルゴリズムの機能およびスケジューリングシミュレータを使用したアルゴリズムの有効性実証のための実験結果について述べる。なお，NWTシステムでは，1994年10月から本アルゴリズムを組み込み，新方式での運用を開始したところである。

＜略号表＞

CFD ：計算流体力学
CP ：数値風洞のコントロールプロセッサ
FEP ：フロントエンドプロセッサ
NS ：航技研計算機システム (Numerical Simulater system)
NWT ：数値風洞 (Numerical Wind Tunnel)
PE ：数値風洞の要素計算機
SHP ：NSシステムのベクトルプロセッサ
SSU ：数値風洞の高速システム記憶

2. NWTシステムのジョブスケジューラに関する予備的考察

本章では，NWTシステムの現状について簡単に説明を行った後，NWTシステムのジョブスケジューラに要求される機能およびジョブの投入から実行終了までの状態遷移について述べる。

2.1 NWTシステムの現状

本項では，NWTシステムの現状として，スケジューリングに関係のあるSSUの設置目的，PEの負荷分散，PEの割り当て方法，ジョブのシステム資源制限項目，大規模ジョブのスワップアウト[注1)]および現在使用中のジョブスケジューリング方式について説明する。

(1) SSUの設置目的

NWTシステムでは，ジョブ実行時の入出力データおよびスワップアウトデータを一時的に保存するために，FEPとCPの間にSSUを設置している（図1.1参照）。SSUを有効に使用することができれば，以下に示すように，ジョブ実行時の入出力待ち時間の軽減，スワップアウト，スワップイン[注2)]処理時間の短縮化が図れる。

最初に入出力待ち時間について説明する。NWTで実行

(注1) 実行中のジョブの情報を継続実行可能な形でディスク等に退避すること。NWTシステムの場合の退避先はSSUであり，SSU容量が不足した場合にはFEP配下の磁気ディスクを使用する。
(注2) スワップアウト中のジョブの情報をPEに復元すること。

する実行形式プログラムおよびデータ等はすべてFEP配下の磁気ディスクにあるため，ジョブ実行時には使用するすべてのデータをFEPからPEへ転送し，ジョブ出力時には出力データをPEからFEPに転送しなければならない。しかし，図2.1に示すように，PE-CP間の転送速度が最大421MB/Sであるのに対し，磁気ディスクは最大4.5MB/S（並列入出力方式により4ファイルに並列に入出力を行った場合でも18MB/S）であり，FEP-PE間のデータ転送では転送速度の最も低い磁気ディスクの速度で抑えられる。しかし，SSUからCPへの転送速度は最大1250MB/Sであるため，ジョブ実行に先立ってあらかじめSSUにデータを転送しておく（この機能をプレステージ機能と呼ぶ。）ことができれば，ジョブ開始時のデータ転送は最大421MB/Sとなり，ジョブの入力待ち時間を大幅に短縮することが可能となる。また，出力処理ではPEからSSUへのデータ転送とSSUからFEPへのデータ転送は独立に処理可能であり，SSUへの書き込みが終了した時点でプログラムは次の処理に移ることが可能となるため，SSU容量が十分にあれば出力も最大421MB/Sとなり，ジョブの出力待ち時間も大幅に軽減できる。

続いて，スワップアウト・スワップイン処理時間であるが，SSUが設置されていない場合にはスワップアウトデータはFEP配下の磁気ディスクに格納することになるため，処理速度は磁気ディスクの転送速度となる。しかし，SSUが使用できる場合にはスワップアウト時はPEの情報をSSUに格納し，スワップイン時はSSUの情報をPEに復元するため，入出力と同様に高速に処理が行える。

しかし，これらの処理でSSU容量が不足した場合にはすべて磁気ディスクとPE間の転送となるため所期の目的が果たせなくなり，入出力処理等に長時間かかることになる。したがって，SSU容量はこれらの目的を達成できる程度に十分にあることが前提となる。また，スワップアウトやプレステージの頻発によりSSUが占有されると同様の結果となるため，NWTシステムでは，スワップアウト量やプレステージ量を制限すべきである。

(2) PEの負荷分散

NWTシステムでは，複数PEを使用するジョブの場合，PEの使用効率を高め，ジョブの処理時間を最短にするためにPEの負荷をできるだけ均等に分散させることがユーザに義務づけられている。ここで，4種類の処理A，B，C，Dから構成されるジョブの負荷分散の例を図2.2に示す。なお，処理Aには8時間，処理B，Cにはそれぞれ2

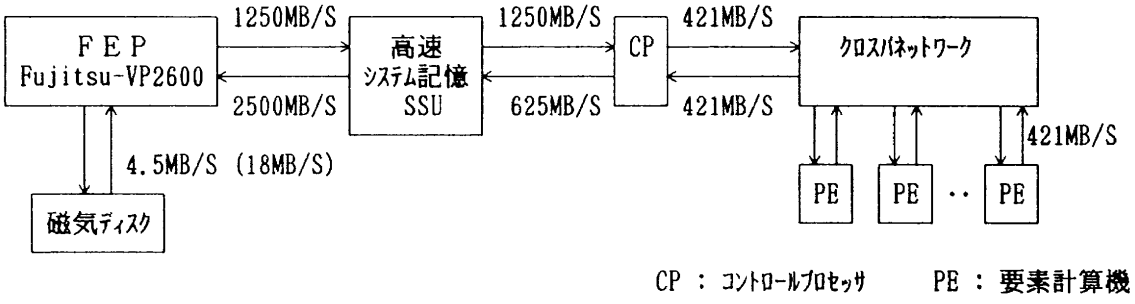
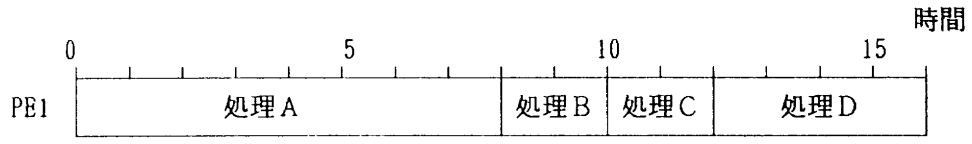
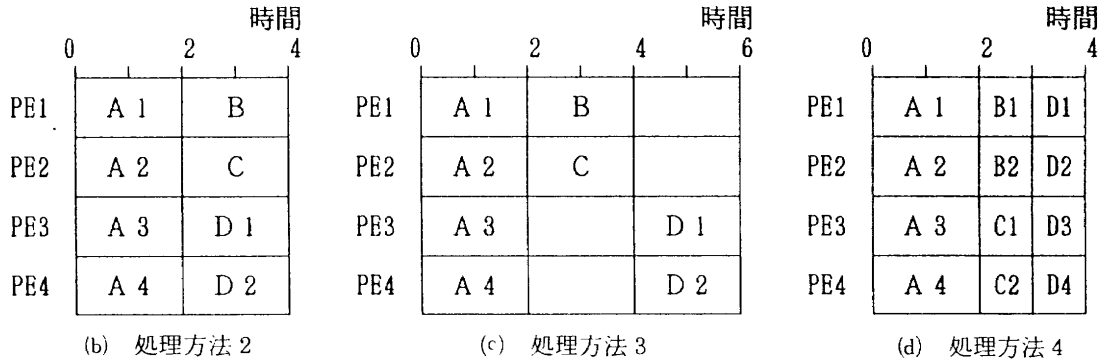


図2.1　データ転送能力（理論性能）



(a)　処理方法1



(b)　処理方法2　　(c)　処理方法3　　(d)　処理方法4

図2.2　並列計算機使用時のプログラムの負荷分散の例

時間，処理Dには4時間かかるものとする。

(a)はこのジョブを1PEで実行した場合を示している。当然のことながら，全処理を行うのに16時間かかる。(b)はこのジョブを4PEで実行した場合を示している。ここで処理A1，A2，A3，A4は処理Aを，処理D1，D2は処理Dを，それぞれ並列実行可能な処理に均等分割したものである。その結果，ジョブの実行時間は4時間となる。ただし，この場合には，処理B，C，D1，D2は処理Aの結果を使用して並列に実行可能でなければならない。もし，処理Dが処理B，Cの結果を使用するような場合には，処理Dの開始は処理B，Cの終了を待つため，(c)のように処理しなければならず実行時間は6時間となってしまう。この場合は，処理BはB1，B2に，処理CはC1，C2に，処理DはD1，D2，D3，D4に均等分割すれば(d)のような処理となり実行時間は4時間となる。一般的に，入出力処理は1PEで行うことが多いため，同図の例のように完全な負荷分散は不可能であるが，できるだけ(b)または(d)のように各PEの処理を分割することが望ましい。このような処理をPEの負荷分散処理と呼ぶ。

(3) PEの割り当て方法

NWTシステムでは，ジョブ実行時のPEの割り当て方法は，「マルチジョブ割り当て」と「シングルジョブ割り当て」というふたつの方法が選択可能であった。前者は，PEの有効利用の観点からメモリに余裕のあるPEを複数のジョブに割り当てる方法であり，後者は単一ジョブに割り当てる方法である。NWTシステムの検討開始に当たり，どちらの方式を採るかを検討した結果，以下の結論を得た。

①「シングルジョブ割り当て」の場合は，「マルチジョブ割り当て」に比べて入出力によるPEのアイドルの確率が高いと考えられるが，どちらの場合でも入出力によるアイドルは極めて少ないと考えられる。

②「マルチジョブ割り当て」の場合には，ジョブの実行は同一PEを使用している他のジョブの実行に影響されるため，ジョブ終了の見通しが立ちにくい。

③複数PEを使用するジョブでは，PEの有効利用を図るためにPEの負荷分散が重要な要素となる[6)-8)]が，「マルチジョブ割り当て」によりジョブの実行が他のジョブに影響される場合には，PEの負荷分散効果が期待できない。

本システムはCFDプログラムの高速実行を目的として開発するという前提に立っているため，上記3つの理由により「シングルジョブ割り当て方式」を採用することに決定した。なお，この決定は，NWTシステムの開発方針となった。

(4) ジョブのシステム資源制限項目

一般的に，どの計算機システムでも，システム資源使用量の公平化とプログラム処理異常時の対処のためにシステム資源を制限している。NWTシステムの場合もシステム資源の制限は必要であり，投入されるジョブに対してPE使用台数，CPU使用時間，経過時間（NWTシステム内でジョブが実行を開始してから実行を終了するまでの時間）に関する制限値を設定し，ジョブ実行中にCPU使用時間や経過時間の制限値を越えた場合にジョブを打ち切る機能がある。ただし，NWTシステムで複数PEを使用する場合には各PEで異なる処理を行わせることも可能であるため，PE毎のCPU使用時間が異なることが予想される。すなわち，CPU使用時間は，全PEが使用した総CPU使用時間，平均CPU使用時間，最大CPU使用時間の3種類が考えられる。このうち，総CPU使用時間，平均CPU使用時間は，個々のPEの情報だけでは求めることができないため，制限値を越えた時点でジョブを打ち切るためには常に各PEの情報を収集するためのPE間通信等が頻発する。したがって，PE間通信等のオーバヘッドの増加防止を図るためには総CPU使用時間や平均CPU使用時間は採用すべきでない。

また，ジョブは終了直前に計算結果を出力することが多いが，この場合には全ての出力データがPEからSSUを経由してFEPに転送された時点でジョブが終了する。すなわち，CPUの処理が全て終了し，出力結果がSSUまで出力された場合でも，FEPへの転送中に経過時間切れとなった時にはジョブ処理は中断され出力結果の保証はできない。

以上に述べた理由により，NWTシステムの運用ではシステム資源の主たる制限項目として，PE使用台数，最大CPU時間を使用し，プログラム処理異常時の対処を可能にするために大きめに設定した経過時間を補助的に使用するべきであると結論した。したがって，スケジューラにおけるジョブスケジュールの際にもジョブで設定した経過時間は使用できない。

(5) 大規模ジョブのスワップアウト

複数PEを使用するジョブの場合，処理時間を最短にするために，ユーザは(2)項で述べたPEの負荷分散が義務づけられている。したがって，このように複数PEを使用するジョブのPEが1台でもスワップアウトされるとその分だけ全PEの処理が遅れることになる。言い換えると，スワップアウトに伴うPE利用率の低下の割合は規模が大きいジョブをスワップアウトした場合ほど顕著となるため，大規模ジョブのスワップアウトを規制する必要がある。

(6) 現在のスケジューリング方式

現在，並列計算機用に考案されたスケジューリングアルゴリズムの実用例がないため，NWTシステムでは，通常の汎用計算機用のスケジューリングアルゴリズム[10)]を使用している。このアルゴリズムは，CPU時間，メモリ量等の制限が異なる数個のジョブクラスを定義し，それぞれのジ

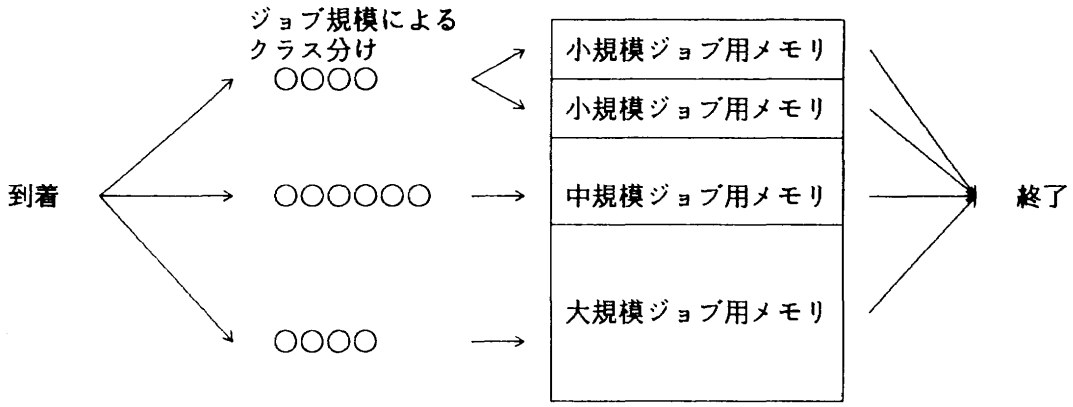


図2.3　ジョブクラス別多重度制御方式の例

ョブクラスの多重度（同時に実行可能なジョブ数）の指定に基づいてジョブクラス内ジョブの実行を制御する方式である。本稿では，この方式をジョブクラス別多重度制御方式と呼んでいる。以下に，本方式の特徴と NWT で使用した場合の問題点を説明する。

(a)　ジョブクラス別多重度制御方式の特徴

一般的な汎用計算機の運用では，CPU の高効率利用と公平なジョブのターンアラウンドタイムが重要課題である。航技研では，以下に述べるジョブクラス別多重度制御方式を使用することにより，汎用計算機の効率的運用を実現している。

図2.3 に汎用計算機でのジョブクラス別多重度制御方式例を示す。例ではジョブの規模により，大規模，中規模，小規模という3つのジョブクラスを定義し，多重度をそれぞれ1，1，2としている。この方式は，同一ジョブクラスに投入されたジョブは先着順に処理することを原則としているため，大規模ジョブクラスおよび中規模ジョブクラスのジョブはそれぞれ1本，小規模ジョブのみ2本まで先着順に処理される。当然のことであるが，各ジョブクラスのジョブを指定多重度で実行するためには，各ジョブクラスのメモリ制限値と多重度の積を加算した値はシステムで使用可能なメモリ量以下でなければならない。この方式の特徴を下記に示す。

①同一ジョブクラス内では先着順に処理されるため投入されているジョブのターンアラウンドタイムの予測が比較的簡単に行える。

②大規模クラスのジョブでも一定時間待てば処理される。

③ジョブクラス制限値に満たないジョブを実行すると未使用のメモリ領域が発生するが，システム内で複数のジョブが実行されている限り CPU の高効率利用が可能である。

(b)　NWT での問題点

表2.1 に，NWT システムでのジョブクラス別多重度制御方式例を示す。ここでは，システムの総 PE 台数が 140

表2.1　多重度設定例

	クラス別設定値		クラス別使用PE台数
	PE台数	多重度	
クラス1	1	6	6
クラス2	2	3	6
クラス3	4	2	8
クラス4	10	1	10
クラス5	30	1	30
クラス6	80	1	80
合　計	——	14	140

台であるにも関わらず，各ジョブクラスの PE 制限台数と多重度の積を加算した値はシステムで使用可能な PE 台数以下という制限からジョブクラス数は6しかない。この場合以下の問題点が発生する。

①プログラムにはそれぞれ効率実行するための最適な実行 PE 台数があるにもかかわらず，使用可能な PE 台数が限定される。

②プログラムをジョブクラスの制限台数に合わせて実行すると，各 PE のロードバランスの調整を最適にできないために PE の使用効率が低下する。

③最適 PE 台数でないジョブクラスでプログラムを最適 PE 台数で実行すると，実行中常に空き PE が発生し，PE の使用効率が大幅に低下する。

④混雑時でもジョブのないジョブクラスがあると未使用の PE が発生する。

したがって，PE の有効利用を図る観点からは，ジョブクラス別多重度制御方式は採用できない。

2.2　NWT システムのジョブスケジューラに要求される機能

ジョブスケジューラに要求される機能は，システム資源の効率的運用機能，適切なジョブターンアラウンドタイム

保証機能，システム運用管理機能に大別することができる。本項では，第一期NSシステムのジョブスケジューラ[12]の特徴をこれらの要求機能毎に示し，それに対応させてNWTシステムのジョブスケジューラに必要となる機能について述べる。

昭和62年2月に導入された第一期NSシステムはフロントエンドプロセッサにFujitsu-M780を，バックエンドプロセッサにベクトル計算機であるFujitsu-VP400とVP200（以下これらのベクトル計算機をSHPと略記する）を配置した疎結合の複合計算機システムである。なお，VP200は平成3年1月にシステム内の位置づけは同じままでVP2600に代替された。このシステムでは，ベクトル計算機であるSHPの有効利用を図るために，実行ステップのみSHPで処理し，実行ステップ以外のSHPジョブ，会話型処理等はすべてFEPで処理した。また，ジョブに対するシステム資源の制限は主としてメモリ量とCPU時間で行った。

(1) システム資源の効率的運用機能

汎用計算機システムである第一期NSシステムでは，システム資源，特にCPUの効率的運用が最重要課題であり，この課題を達成するために実行可能ジョブの確保機能およびジョブクラス別多重度制御機能を使用した。

実行可能ジョブの確保機能は，FEP内のSHPジョブを優先的に実行してSHPの実行可能ジョブを一定数確保する機能であり，SHPジョブのサービスステップ処理待ちによるSHPの空き状態の回避に有効である。ジョブクラス別多重度制御機能の実現方式については第2.1(6)項で述べたが，CPUの高効率利用に有効な機能である。

これに対し，NWTシステムでは各PEの有効利用を図ることが最重要課題である。しかし，汎用計算機用のスケジューリングアルゴリズムであるジョブクラス別多重度制御機能ではPEの有効利用が図れないため，システムの有効利用の観点から以下の機能が必要となる。

①使用可能なPE使用台数を限定しない。

②未使用のPEを極力発生しない割り当て方式を採用する。

③ジョブの流れを適切に制御してPEの利用効率をできるだけ向上させる。

なお，PEの利用効率向上のための機能として，NWTシステムでは第2.1(1)項に述べたプレステージ機能が提供されている。PEの有効利用を図るためにはこの機能を積極的に利用する必要がある。

(2) 適切なジョブターンアラウンドタイム保証機能

適切なジョブターンアラウンドタイムを保証することは，研究者が研究計画を遂行するために重要である。ジョブクラス別多重度制御機能は，ジョブクラス多重度分のジョブの実行が保証されるため，ジョブの適切なターンアラウンドタイムの保証にも有効な機能である。また，ジョブクラス毎にすべてのジョブに起動[注3]優先度，メモリ優先度，実行優先度を設定（運用時間帯ごとに設定可能）し，これらの優先度によりジョブの実行を制御しているため，システム資源が競合した場合にもジョブクラス毎のジョブのターンアラウンドタイムを相対的に保証することが可能である。

NWTシステムでも適切なジョブターンアラウンドタイムの保証は重要な課題である。その実現のためには，ジョブの属性やシステム資源要求量に応じたジョブ起動制御機能だけでなく，起動できずに不当に長時間待たされているジョブに対しては実行中ジョブのスワップアウト処理による要求PE台数の確保や他ジョブの起動抑止による早期起動を可能とする実行制御機能等が必要となる。

(3) システム運用管理機能

第一期NSシステムの特権ジョブは，システムが原因となる事故ジョブの処理が大半を占めており，発生件数が限られること，個別対応が必要なことからマニュアル操作での高優先度設定機能により対処していた。

これに対しNWTシステムでは，HOPE[注4]シミュレーションのようなプロジェクトジョブを緊急に処理する必要性が考えられていた。これらのジョブは，必ずしも実行中のジョブを中断させてまで即刻起動する必要があるとは考えられないが，使用PE台数が30台程度であり他のシステムで実行することが不可能に近いこと，比較的高頻度で投入されると予想されることから，システム管理者によるマニュアル操作なしで早い時期に起動できるようにする必要があった。そのためには，特権ジョブの概念を取り入れ，特権ジョブを非特権ジョブより優先して起動，実行するためのジョブ起動方式や実行制御方式を組み込むことが必要となる。

2.3 ジョブの状態と遷移

図2.4にジョブ状態遷移図を示す。同図において，スケジューラが意識すべきジョブの状態を①～⑦，起動待ちジョブのデータの状態をⅰ～ⅱに示す。また，a～eの状態遷移はスケジューラが決定することを示す。本項では，同図を基に，それぞれのジョブ状態と遷移について述べる。

① 起動待ち状態

システムに到着し，起動を待っている状態を起動待ち状態と呼ぶ。また，起動待ち状態のデータは，プレステージが完了しているか否かによりプレステージデータまたはプレステージ待ちデータと呼ぶ。なお，プレステージおよび

(注3) 実行待ちのジョブの実行を決定すること。また，そのために必要となる処理を行うこと。

(注4) H-Ⅱ orbiting planeの略語。航技研と宇宙開発事業団の共同開発プロジェクト

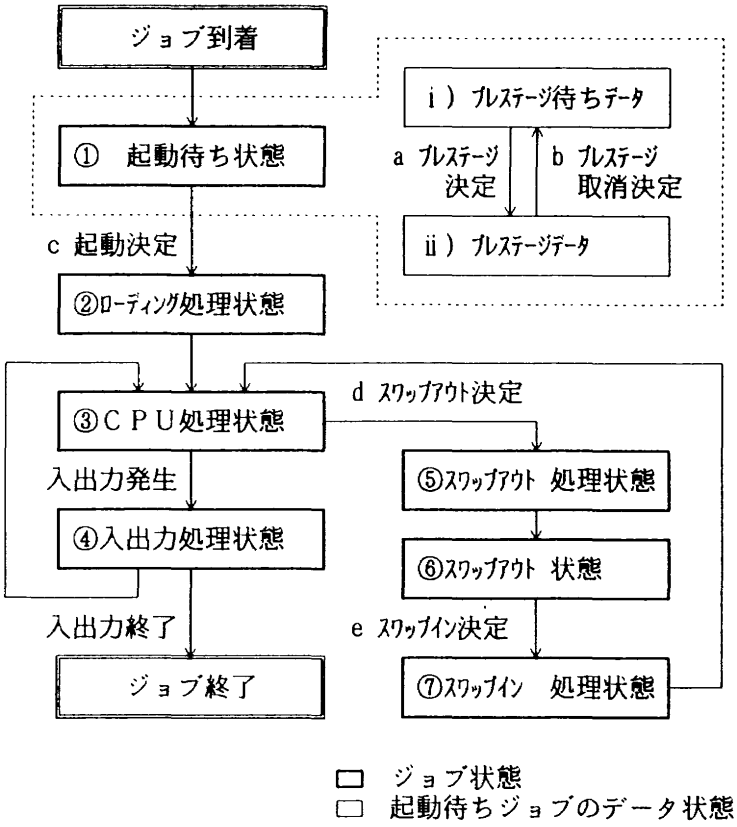


図 2.4　ジョブ状態遷移図

プレステージ取消の決定はスケジューラが行う。

② ローディング処理状態

スケジューラにより起動が決定され，実行開始に必要なデータを PE に転送中の状態をローディング処理状態と呼ぶ。転送データの内，プレステージデータは SSU から PE に，プレステージ待ちデータは FEP から SSU を経由して PE に転送される。

③ CPU 処理状態

ローディング処理状態，入出力処理状態，あるいはスワップイン処理状態でのそれぞれの処理が終了し，CPU処理中の状態を CPU 処理状態と呼ぶ。なお，CPU 処理によりすべての処理が終わったジョブ，あるいは予定CPU時間を使い切ったジョブは終了する。

④ 入出力処理状態

CPU 処理中に入出力要求が発生し，入出力処理中の状態を入出力処理状態と呼ぶ。なお，入出力によりすべての処理が終わったジョブは終了する。

⑤ スワップアウト処理状態

ジョブ実行中にスケジューラによりスワップアウトが決定され，スワップアウト処理中の状態をスワップアウト処理状態と呼ぶ。ジョブが CPU 処理状態の場合にはスワップアウトの決定と同時にスワップアウト処理状態となるが，入力処理状態の場合には入出力処理が終わり CPU 処理状態となった時点でスワップアウト処理が開始される。

⑥ スワップアウト状態

スワップアウト処理が終了し，スワップイン待ちの状態をスワップアウト状態と呼ぶ。

⑦ スワップイン処理状態

スワップアウト状態のジョブのスワップインが決定され，スワップイン処理中の状態をスワップイン処理状態と呼ぶ。この状態のジョブは，スワップイン処理が終了した時点で CPU 処理状態となる。

3. NWT システムのジョブスケジューリング方式

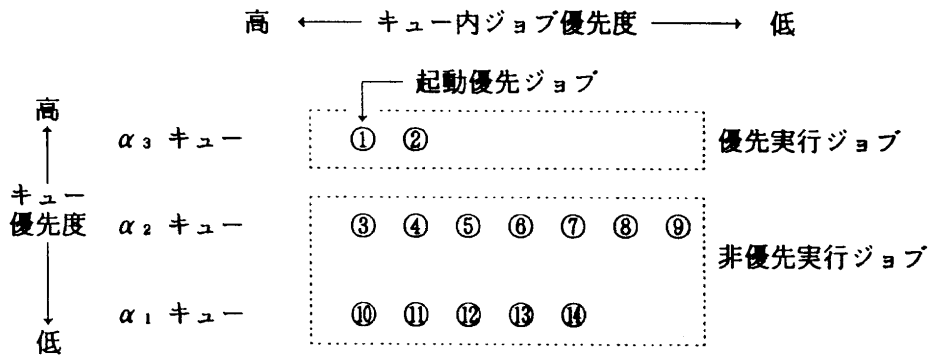
前章で，NWT システムのスケジューラに要求される機能について述べた。本章では，これらの要求事項を満たし，NWT システムを効率的に運用するために考案したスケジューリング方式について述べる。

3.1 ジョブスケジューラの機能

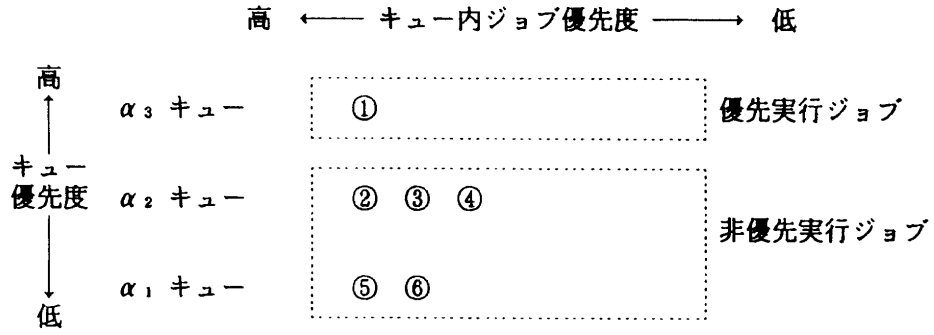
(1) ジョブの分類とジョブのキュー管理機能

システム内のジョブを異なる優先度レベルで処理するために，優先度の低い方から順に α_1 ジョブ，α_2 ジョブ，α_3 ジョブに分類する。これらのジョブは，実行待ちジョブ，実行中ジョブごとに図3.1 に示すジョブ管理キューで管理する。

実行待ちジョブ管理キューおよび実行中ジョブ管理キュ



(1) 実行待ちジョブ管理キュー



(2) 実行中ジョブ管理キュー

図3.1 ジョブ管理キュー

ーにはα_1キュー，α_2キュー，α_3キューと呼ぶ3本のキューを設け，それぞれα_1ジョブ，α_2ジョブ，α_3ジョブを優先度の高い順に管理する。図中の丸付き数字は各キューにつながれているジョブを示す。また，中の数字は実行待ちジョブ管理キューまたは実行中ジョブ管理キュー内の相対的ジョブ優先順位を示す。

なお，待ち時間が長くなったジョブを最優先に処理するためにはα_3ジョブの処理方式を変える必要がある。そこで，α_3ジョブを「優先実行ジョブ」，α_2ジョブおよびα_1ジョブを「非優先実行ジョブ」と呼び，ジョブ起動時，PEの割り当て時等において優先的な処理方式を設定する。ただし，α_3ジョブの起動時には強硬な手段を用いる必要があるため，すべてのα_3ジョブを対象とすると混乱を生じる恐れがある。したがって，特にその先頭ジョブを「起動優先ジョブ」と呼び，優先的な起動方式を設定する。

(2) ジョブ優先度の動的制御機能

ジョブ属性，PE要求台数，CPU要求時間，待ち時間からジョブの所属キューとキュー内優先度を決定するための関数をあらかじめ定義しておく。本稿では，システム資源要求量と待ち時間によって優先度が決定するように第3.2項(1)，(2)に定義する関数を使用しているが，関数を変えることにより，到着順の処理や特定規模のジョブの優先処理等，任意の設定が可能である。

起動ジョブを選択する際には，すべてのジョブがその時点での優先度で処理されるように，所属キューおよび優先度を見直し，必要に応じてジョブを適当なキューへつなぎ直す。

(3) ジョブの起動機能

割り当て可能な空きPEがある場合には，実行待ちジョブ管理キュー内のジョブの起動条件（第3.3項に述べる）を相対的ジョブ優先順位の高いジョブから順に調べ，起動条件を満たしているジョブをすべて起動する。処理中に起動条件を満たさないジョブがあっても，割り当て可能な空きPEがある限り後続のジョブの起動を試みることによりPEの使用効率の向上を図る。

(4) 優先実行ジョブに対するPEの優先割り当て機能

PEの割り当ては，実行中の優先実行ジョブ，実行待ちの優先実行ジョブ，実行中の非優先実行ジョブ，実行待ちの非優先実行ジョブの順に優先する。したがって，スワップイン，スワップアウト，起動のためのPEの割り当てはこの優先度にしたがって処理する。しかし，PE要求台数が極端に多い実行待ちジョブは優先度を高くするだけでは実行不可能な場合も考えられるため，α_3キューの先頭ジョブである起動優先ジョブは以下に述べる特別なPEの割り当て方式を採る。

起動優先ジョブ起動時の要求PE台数の割り当ては，空

き PE だけでなく，実行中の非優先実行ジョブのスワップアウトによる空き PE の割り当ても可能とする。それでも起動できない場合には，他のすべてのジョブの起動を抑止した場合のジョブ開始時刻と終了時刻を予測し，その時間帯に確実に実行するために要求台数の PE の割り当てを予約する。起動優先ジョブに対するこの処理を「PE リザーブ」と呼ぶ。

(5) スワップアウト機能

スワップアウト処理では，実行中の非優先実行ジョブの中のジョブ優先度の低いジョブの PE から順に取り上げる。ただし，下記の3項目の上限をあらかじめ設定しておき，これらの範囲内でのみスワップアウトを行うものとする。

①スワップアウト可能なジョブの規模（要求 PE 台数）

②1つのジョブを起動する際のスワップアウト PE 台数

③システム全体での総スワップアウト PE 台数

ここで，①は PE の利用率の大幅低下を引き起こす恐れのある大規模ジョブをスワップアウト対象から外すためのパラメータであり，②，③は，スワップアウトの頻発による SSU の専有やシステムオーバヘッドの増大を防止するためのパラメータである。

(6) プレステージ機能

プレステージ処理では SSU の無駄な専有を防止するために，次に起動されるジョブとその起動時間，プレステージに要する時間を予測し，起動直前にプレステージを終了させることが最も望ましい。しかし，ジョブの打ち切りは CPU 時間で行う方針であり，ジョブは終了予定時刻以前に異常終了する場合もあることから，次に起動されるジョブの正確な予測は不可能である。そこで起動待ちジョブの中の優先度の高い順に一定数のジョブをプレステージする方式とし，システム状況の変化によりプレステージすべきジョブが変更になった場合には，それに伴ってプレステージジョブも入れ換えるものとする。

3.2 ジョブとキューの定義

(1) キュー遷移条件

本アルゴリズムでは，待ち時間が一定値に達したジョブの優先処理を行うために，以下の遷移条件を満たしたジョブをα_2キューおよびα_3キューへ遷移させる。

α_2キューへの遷移条件：$WTj \geqq WTj_1$

α_3キューへの遷移条件：$WTj \geqq WTj_2$

ここで，WTj はジョブが投入されてから現在に至るまでの待ち時間であり，WTj_1，WTj_2はジョブのシステム資源要求量や待ち時間から任意に決定することができるが，NWT システムでは以下のように定義する。

a. 一般ジョブの場合

$$WTj_1 = [CPUj \times \sqrt{PEj}] \times FAC_1$$
$$WTj_2 = [CPUj \times \sqrt{PEj}] \times FAC_2$$

ここで，PEj はジョブ j の要求 PE 台数を，CPUj は要求 CPU 時間を表す。また，FAC_1，FAC_2 はシステムの混雑度に対処するためのファクタであり，

$0 < FAC_1 \leqq FAC_2 < \infty$

である。NWT システムでは，「大規模 CFD プログラムの高速実行」という設置目的により，PE 台数の多いジョブを優遇している。

本稿では，キュー遷移条件を決定する要素の中の FAC_1，FAC_2 をキュー遷移パラメータと呼ぶ。

b. 特権ジョブの場合

$$WTj_1 = -WTP$$
$$WTj_2 = -WTP$$

ここで，WTP は運用パラメータであり，$-WTP < 0$とする。したがって，先に述べた遷移条件から，投入された特権ジョブは，ただちにα_3キューにつながれる。

(2) ジョブのキュー内優先度

α_1 ジョブ，α_2 ジョブ，α_3 ジョブのキュー内優先度PR_1，PR_2，PR_3 はそれぞれ以下の関数に従って決定する。なお，関数の値の小さいもの程優先度は高いものとする。

$$PR_1 = WTj_1 - WTj$$
$$PR_2 = WTj_2 - WTj$$
$$PR_3 = WTj_3 - WTj$$

ここで，前項の遷移条件から，α_1 ジョブおよびα_2 ジョブの優先度 PR_1，PR_2 は常に正となり，α_3 ジョブの優先度 PR_3 は常に負となる。また，$PR_1 \neq PR_2$ であることから，α_2キューへ遷移したジョブの優先度はすでに存在するα_2ジョブの優先度より高くなることがあるのに対し，$PR_2 = PR_3$であるため，α_3キューへ遷移したジョブの優先度はすでに存在するα_3 ジョブの優先度より高くなることはない。

また，α_3キューにつながれた特権ジョブの優先度PR_3は，

$PR_3 = -WTP - WTj$

となることから，WTP を一般ジョブで予想される最大経過時間より十分大きく設定することにより，一般ジョブより常に優先的に処理することが可能となる。

3.3 ジョブ起動条件

ジョブを起動するためには，要求台数の PE の割り当て

が可能であることが必須であるが，本スケジューラでは効率的，効果的なスケジュールを実現するために，以下の起動条件を使用する。

a. 起動優先ジョブの起動条件

> 条件1 : PEj≦free
> 条件2 : (PEj−free)≦AVSWAP,
> ただし，AVSWAP=MAX(SWAP1, SWAP2, SWAP3−SWAP)

b. 非起動優先ジョブの起動条件

> 条件1 : PEj≦free

ここで使用されている記号の定義は，以下の通りである。

PEj : 起動対象ジョブの要求 PE 台数
free : 起動対象ジョブが使用可能な空き PE 台数
SWAP : スワップアウト中の PE 台数
SWMAX : スワップアウト可能なジョブの規模（単位は PE 台数）の上限
SWAP1 : 現在スワップアウト可能なジョブ（実行中の α_1 ジョブ，α_2 ジョブのうち要求 PE 台数が SWMAX 以下のジョブ）の総 PE 使用台数
SWAP2 : 一つのジョブを起動する際のスワップアウト PE 台数の上限
SWAP3 : システム全体での総スワップアウト PE 台数の上限

条件1は，空き PE だけでジョブを実行するための条件である。この場合には，要求台数の PE を割り当ててジョブを実行する。

条件2は，空き状態の PE が要求台数に満たないときに，不足分の PE をスワップアウトにより生成するための条件であり，起動優先ジョブの場合のみの条件である。この条件を満足した場合には，スワップアウト可能なジョブの中の実行優先度の低いジョブから順に (PEj−free) 台数分の PE をスワップアウトし，既存の空き PE とスワップアウトによって生じた空き PE を割り当ててジョブを実行する。

なお，起動条件で使用した「起動対象ジョブが使用可能な空き PE 台数 (free)」の算出方法について以下に説明する。

起動優先ジョブの場合，他の実行待ちジョブより優先的に起動する。したがって，空き PE 台数をすべて使用することができるが，使用可能 PE 台数が変化する場合には，その時刻と変化量を考慮しないとシステム停止等の処理が遅延する恐れがある。さらに，非起動優先ジョブの場合には，起動優先ジョブのための PE リザーブ台数をも考慮しなければならない。

ジョブを新たに起動しないと仮定すると，システムで使用可能な空き PE 台数は，①ジョブの終了，②使用可能最大 PE 台数の変更，③PE リザーブの開始，④PE リザーブの終了により変化する。①および②を考慮した場合の例を図3.2に示す。図では縦軸に PE 台数を，横軸に時刻をとっている。

説明に先立ち，以下の記号を定義する。

jmax : 現在実行中のジョブ数
PEj : j番目に終了予定のジョブの使用 PE 台数

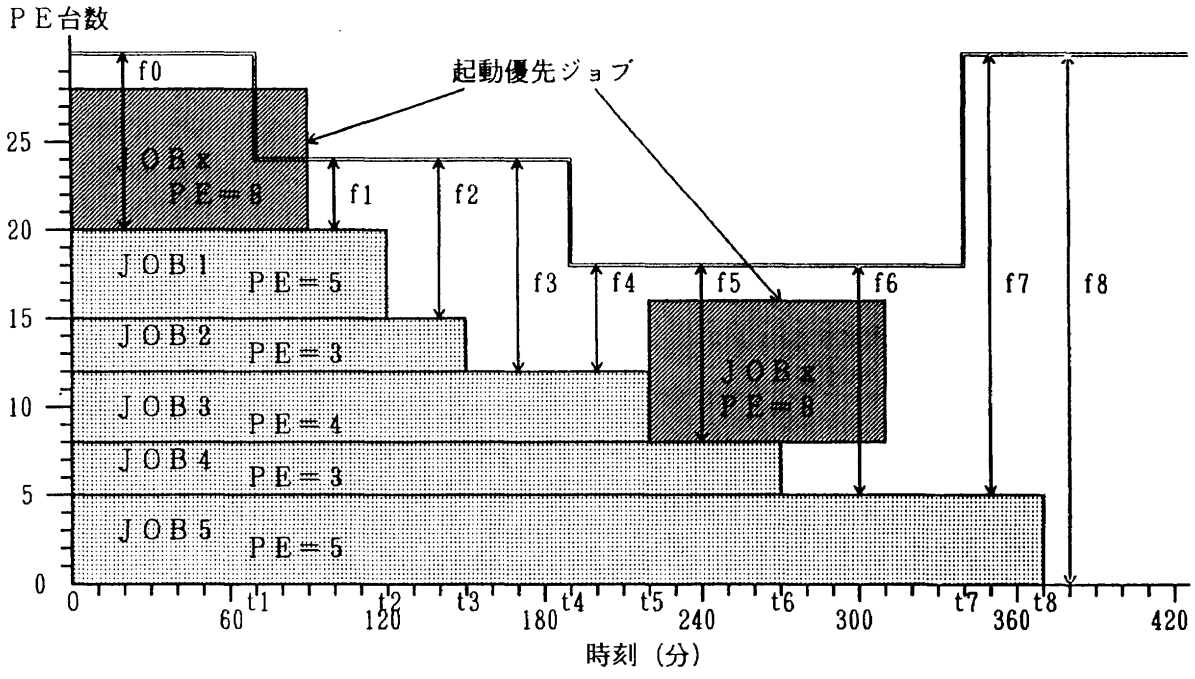


図3.2 ジョブ起動時の使用可能 PE 台数説明図

(j＝1～jmax)

nmax　：空き PE 台数変更事象の数

t0　：現在の時刻

tn　：n番目に発生する空き PE 台数変更事象の発生時刻 (n＝1～nmax)

fn　：時刻 tn における空き PE 台数(n=0～nmax)

FPEn　：時刻 tn における空き PE 台数の変化量 (n＝1～nmax)

PEmax　：時刻 t0 におけるシステムで使用可能な最大 PE 台数

この場合，空き PE 台数の変化事象はジョブの終了および図中の時刻 t1, t4, t7 で示されるような利用可能な PE 台数の変更となる。なお，時刻 tn における空き PE 台数 fn は次式で表される。

$$fn = \begin{cases} PEmax - \sum_{i=1}^{jmax} PEn \quad (n=0 \text{ のとき}) \\ fn_{-1} + \sum_{i=1}^{n} FPEn \quad (n=1 \sim nmax \text{ のとき}) \end{cases}$$

同図では，起動優先ジョブを時刻 t0 から開始した場合と時刻 t5 から開始した場合の例が示されているが，free=「現在の空き PE 台数」と考えて時刻 t1 までに終了しないジョブを t0 に起動した場合には，時刻 t1 に行う予定の使用可能 PE 台数の削減処理が行えないことを示している。すなわち，使用可能最大 PE 台数が変化する場合には，free＝「時刻 t0 から時刻 t0＋ELAPSj までの間の最小空き PE 台数」としなければならない。ここで，ELAPSj は「起動対象ジョブの PE 占有時間」を表す。なお，この場合の各時刻における空き PE 台数は，表 3.1 に示す表を使用して算出することができる。例では，時刻 t5 に起動可能と判定できるので，t5 を PE リザーブ開始時刻，t5+90分を PE リザーブ終了時刻として PE をリザーブする。この場合，PE リザーブ開始およびリザーブ終了をシステムの使用可能 PE 台数の変更事象と同様に考えて，表 3.1 を表 3.2 のように変更することにより，PE リザーブ有りの場合にも対応可能となる。

(4)　補足

(1), (2)項で述べた起動の可否や起動予定時刻の算出には起動対象ジョブの PE 占有時間および実行中ジョブの残り PE 占有時間が必要となる。しかし，NWT システムのジョブ打切りのための経過時間は，第 2.1(4)項に述べたように，ジョブの出力を保証するために大きめにセットせざるを得ないため，PE 占有時間としては利用できない。仮にユーザの予想する経過時間を知る方法があったとしてもジョブが途中で異常終了する可能性もあり，スケジューラが各ジョブの実 PE 占有時間を正確に知ることは不可能である。したがって，ジョブの PE 占有時間は要求CPU時間か

表 3.1　空き PE 台数算出表(1)

n	時刻（分）	FPEn	fn	備考
0	0	—	10	ジョブ終了
1	70	−6	4	PE台数削減
2	120	5	9	ジョブ終了
3	150	3	12	ジョブ終了
4	190	−6	6	PE台数削減
5	220	4	10	ジョブ終了
6	270	3	13	ジョブ終了
7	340	12	25	PE台数増加
8	370	5	30	ジョブ終了

表 3.2　空き PE 台数算出表(2)

n	時刻（分）	FPEn	fn	備考
0	0	—	10	ジョブ終了
1	70	−6	4	PE台数削減
2	120	5	9	ジョブ終了
3	150	3	12	ジョブ終了
4	190	−6	6	PE台数削減
5	220	4	10	ジョブ終了
6	220	−8	2	PEリザーブ開始
7	270	3	5	ジョブ終了
8	310	8	13	PEリザーブ終了
9	340	12	25	PE台数増加
10	370	5	30	ジョブ終了

ら予想するものとし，実 PE 占有時間と予想 PE 占有時間の差の程度によりどの程度スケジューリングに影響を及ぼすのかも合わせて検討する。

3.4 ジョブスケジューリング手順

スケジューラの行うべき処理を，キュー登録処理，キュー登録削除処理，キュー遷移処理，起動処理，プレステージ処理に大別し，以下に各々の処理内容を説明する。なお，ジョブ到着時にはキュー登録削除処理を除くすべての処理を，ジョブ終了時およびジョブキャンセル時にはキュー登録処理を除くすべての処理を行う。

(1) ジョブ登録処理

到着したジョブの所属キューおよびキュー内優先度を決定し，決定されたキューに優先度順につなぐ。

(2) ジョブ登録削除処理

ジョブ終了時には実行中ジョブの中から，ジョブキャンセル時には実行待ちジョブまたは実行中ジョブの中から該当するジョブを探し，削除する。

(3) ジョブのキュー遷移処理

a. α_1 キューの実行待ちジョブを優先度順に調べ，α_2 キューへの遷移条件を満たしているジョブがあればつなぎ替える。

b. α_2 キューの実行待ちジョブを優先度順に調べ，α_3 キューへの遷移条件を満たしているジョブがあればつなぎ替える。

c. α_1 キューの実行中ジョブを優先度順に調べ，α_2 キューへの遷移条件を満たしているジョブがあればつなぎ替える。

d. α_2 キューの実行中ジョブを優先度順に調べ，α_3 キューへの遷移条件を満たしているジョブがあればつなぎ替える。

(4) 起動処理

本処理は，以下の手順で行う。なお，システムに割り当て可能な空き PE がなくなった時点で本処理を中止する。

a. スワップアウト中の α_3 ジョブを実行優先順位の高いものから順に調べ，スワップイン可能なものをスワップインする。α_3 ジョブに対するスワップアウトは行わないが，スワップアウト後に α_3 ジョブとなる場合があるため，この処理が必要となる。

b. α_3 キューの実行待ちジョブを実行優先順位の高いものから順に調べ，第3.3項aに述べた「起動優先ジョブの起動条件」を満たすジョブが続く限り順次起動する。「起動優先ジョブの起動条件」を満たさないジョブがあった場合には，そのジョブのための PE リザーブ処理を行い，以下の処理へ移る。

c. α_3 キューの後続のジョブを調べ，第3.3項bに述べた「非起動優先ジョブの起動条件」を満たすジョブがあれば順次起動する。

d. スワップアウトされている α_2 ジョブ，α_1 ジョブがあれば実行優先順位の高いものから順次スワップイン可能なものをスワップインする。

e. α_2 キュー，α_1 キューの実行待ちジョブを実行優先順位の高いものから順次調べ，「非起動優先ジョブの起動条件」を満たすジョブがあれば順次起動する。

(5) プレステージ処理

実行待ちジョブの中の優先度の高いジョブから一定数のジョブをプレステージする。そのために，プレステージすべきジョブ名と現在プレステージされているジョブ名を調べて比較し，新たにプレステージすべきジョブ，プレステージを取り消すべきジョブを決定する。決定に従い，プレステージ処理，プレステージの取消し処理を行う。

4. シミュレーション実験におけるワークロード

本章では，第3章で述べたスケジューリングアルゴリズムの効果を検証するために行うシミュレーション実験で使用するワークロードについて説明する。なお，ワークロードは，ジョブの属性とジョブの到着率から決定する。

4.1 ジョブの属性

NWT システムは，CFDプログラムの高速実行を目的としている。そのため，シミュレーションでも，航技研の CFD プログラムの実行を前提とする。

以下に，CFD プログラムの特徴とシミュレーションで使用するジョブ属性の決定方法について述べる。

(1) 航技研における CFD プログラムの特徴

第一期 NS システムで処理されていた一般的な CFD プログラムの特徴[注5)]を以下に列挙する。

- 格子点数10万点当たりの計算に3600秒程度[注6)]の CPU 時間を必要とする。
- 格子点当たりのメモリ量は200Bから400B程度のジョブが多い。
- 定常状態の計算を行うジョブ（定常ジョブと呼ぶ）はジョブ実行開始直後の入力と終了直前の出力があり，計算途中でのデータ入出力はほとんどない。これに対し，非定常の計算を行うジョブ（非定常ジョブと呼ぶ）は一定間隔で出力があり，CPU使用時間は出力回数に応じて回

(注5) 本スケジューリングアルゴリズムの検討および実験を行った時点では NWT システムはまだ導入されていなかったため，第一期 NS システムでのデータを使用した。

(注6) これは VP-400を使用した場合の値である。PE の処理能力は VP-400の処理能力より勝っていると予測されたが，本実験を行った時点では明確な実測値が得られていなかったためこの値をそのまま使用した。

数倍に増える。

- 運用上の統計データによれば，計算結果の出力量はメモリ量の10%程度であり，入力量はメモリ量の18%程度である。入力量は出力量より格子データの分増える。
- 計算プログラムのディバッグ処理は本計算の約1/20程度のCPU時間を要する。
- 実在気体の計算は，一般的な完全気体の計算に比べてメモリ量，CPU時間ともに2倍程度必要である。

また，CFDプログラムをNWTシステムで処理する場合には，

- メモリ量の増加率（メモリオーバヘッド率と呼ぶ）は1.2から2程度である。

と予想されていた。並列化に伴うメモリ量増加の主な原因は，データ領域の重複である。NWTシステムはメモリ分散型の並列計算機であり，各PE内のメモリの情報はクロスバーネットワークを介して他のPEで参照することが可能であるが，他のPEの情報を参照するためには双方のPEに領域を重複してとる必要があるため，メモリ量が増加する。

なお，非定常ジョブは定常ジョブを連続して行うこととほぼ同等であり，入力が無い分だけSSUの負荷が軽くなるため，スケジューリングアルゴリズムを検証するためのシミュレーションでは非定常ジョブを考慮する必要がないと考える。

以上の結果，シミュレーション用ジョブ属性は，メモリ量，格子点数当たりのデータ量，メモリオーバヘッド率，ジョブファクタ（メモリ量，CPU時間に対するファクタで，実在気体を扱うジョブと完全気体を扱うジョブを発生させるために使用する），CPUファクタ（CPU時間に対するファクタで，ディバッグジョブを発生させるために使用する）の5つの要素を基にして決定することとした。この5つの要素をジョブ属性の基本要素と呼ぶ。

(2)　ジョブ属性の基本要素決定方法

NWTシステムではジョブ属性に未知の要素が多かったため，ジョブ属性の基本要素はそれぞれ独立に決定した。また，属性を決定する際の乱数の発生には，発生データの最小値および最大値が規定でき，パラメータの指定により平均値の異なるデータを発生することができるβ分布[13]を使用した。

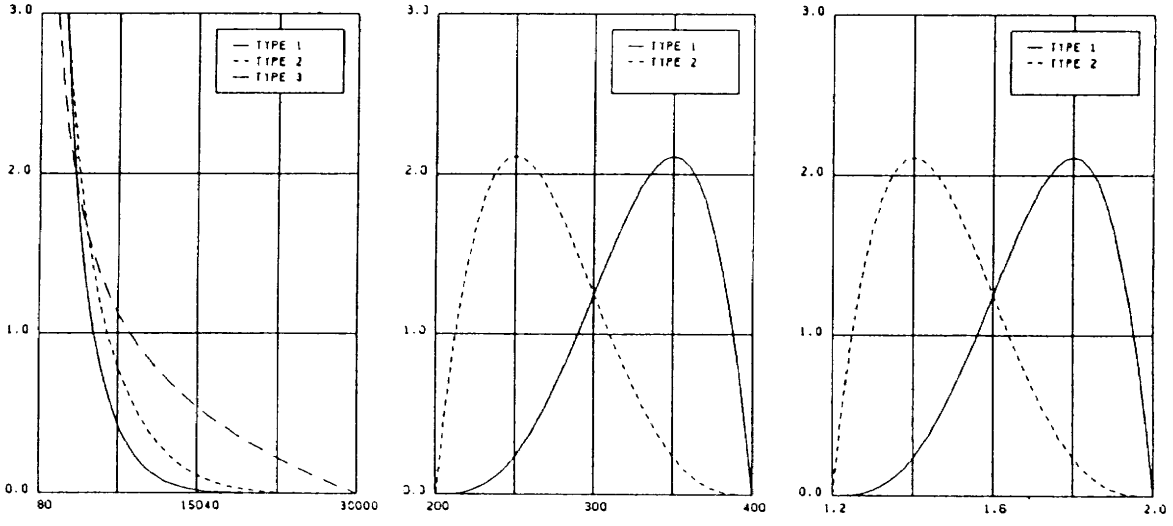
実験ではジョブファクタ，CPUファクタは固定し，ディバッグジョブの発生率は実験の目的に応じて設定する。また，メモリ量，格子点数当たりのデータ量，メモリオーバヘッド率は，β分布のパラメータをそれぞれ3種類，2種類，2種類用意した。したがって，1つの実験に対して12通りのワークロードを使用したシミュレーションが可能となる。そのため，実験結果の比較においては，メモリ量のタイプをi (i=1～3)，格子点数当たりのデータ量のタイプをj (j=1～2)，メモリオーバヘッドのタイプをk (k=1～2)とした場合，使用したワークロードをi/j/kのように表す。また，12通りの結果をメモリ量のタイプごとに集計した場合には，i/*/* (i=1～3)のように表す。

以下に，基本要素の決定方法を示す。ここで，各項目の後ろの［ ］内の記号は，各項目のデータの略号として使用する。

①　メモリ量 [RMEM]（単位 MB）

ジョブのメモリ量は，80 MBから30000 MBの範囲のβ分布で決定する。最大値の30000 MBはPE 120台を使用するジョブを想定したものである。なお，β分布のパラメータはタイプ1 (0.5, 8.0)，タイプ2 (0.5, 5.0)，タイプ3(0.5, 2.0)の3種類使用し（図4.1(a)参照），メモリ量の平均値がそれぞれ1840，2800，6064の場合の実験を行う。

②　格子点当たりのデータ量 [DGRID]（単位 B）



(a)　メモリ使用量　　(b)　格子点当たりのデータ量　　(c)　メモ　オーバヘッド率

図4.1　ジョブ属性決定に使用するβ分布

ジョブの格子点当たりのデータ量は，200Bから400Bの範囲のβ分布で決定する。なお，β分布のパラメータはタイプ1 (4.0, 2.0)，タイプ2 (2.0, 4.0) の2種類使用し（図4.1(b)参照），格子点当たりのデータ量の平均値がそれぞれ333.3, 266.6の場合の実験を行う。

③ メモリオーバヘッド率 [OH]

ジョブのメモリオーバヘッド率は，1.2から2.0の範囲のβ分布で決定する。なお，β分布のパラメータはタイプ1 (4.0, 2.0)，タイプ2 (2.0, 4.0) の2種類使用し（図4.1(c)参照），メモリオーバヘッド率の平均値がそれぞれ1.734, 1.466の場合の実験を行う。

④ ジョブファクタ [FACT]

完全気体のジョブと実在気体のジョブを1:1の割合で発生させ，以下の値を設定する。

$$FACT = \begin{cases} 1.0 & （完全気体の場合） \\ 2.0 & （実在気体の場合） \end{cases}$$

⑤ CPU ファクタ [CFACT]

ディバッグジョブを設定された割合で発生させ，以下の値を設定する。

$$CFACT = \begin{cases} 1 & （非ディバッグジョブの場合） \\ 1/20 & （ディバッグジョブの場合） \end{cases}$$

(3) 基本要素以外のジョブ属性の決定方法

以下に，基本要素以外のジョブ属性の決定方法について述べる。

①使用PE台数 [PE]

PE = RMEM/250.

②PE当たりのメモリ量（単位MB）

PEMEM = RMEM/PE

③基本メモリ量 [BMEM]（単位MB）

BMEM = RMEM/OH

④計算格子点数 [GRID]

GRID = BMEM * 1000000/(DGRID * FACT)

⑤PE当たりの計算格子点数 [PEGRID]

PEGRID = GRID/PE

⑥予定CPU使用時間 [ACPU]（単位ms）

ACPU = (PEGRID/100000.) * 3600. * FACT

⑦CPU使用時間 [CPU]（単位ms）

アベンドジョブを1割の割合で発生させ，以下の値を設定する。

$$CPUT = \begin{cases} ACPU（非アベンドジョブの場合） \\ ACPU * AVRATE \\ \quad （アベンドジョブの場合） \end{cases}$$

ここで，AVRATEは0.0～1.0の一様乱数により決定する。

⑧入力データ量 [INDATA]（単位MB）

INDATA = RMEM * 0.18

⑨出力データ量 [OUTDATA]（単位MB）

OUTDATA = RMEM * 0.10

表4.1は，12種類のワークロードを使用してそれぞれ1000件のジョブを発生し，メモリ量の発生タイプ別にそれぞれの平均値を求めた結果である。

4.2 ジョブ到着時間間隔

システムの負荷は，システムに到着するジョブの平均システム資源要求量とジョブの平均到着時間間隔により決定される。したがって，システムに到着するジョブの属性が決定している場合には，ジョブの到着間隔によりシステムの負荷状態が決まる。

いま，システムの処理能力相当の負荷状態を負荷率1.0，システムの処理能力のx倍の負荷状態を負荷率xとする。ジョブの平均CPU使用時間をcpu，平均使用PE台数をpe，使用可能PE台数をmaxpeとしたとき，負荷率x場合の到着時間間隔 ARRIV (x) は以下のようにして求めることができる。

$$ARRIV(x) = \frac{cpu \times pe}{maxpe \times x}$$

一般的に，負荷状態の異なるシミュレーション実験は負荷率xを変えることにより行うことができる。しかし，本アルゴリズムの場合には，ジョブ優先度の決定に待ち時間を使用しているため，処理能力以上の負荷を想定した場合には，単に負荷率を上げただけでは除々にジョブが蓄積し優先度設定の意味がなくなってしまうこととなる。実際のシステムの場合，混雑時にはジョブの投入を控えるユーザがいるためシステム内ジョブ数が一定値を越えると到着間隔

表4.1 ワークロード別ジョブ属性

	単位	最小値	最大値	1/*/*	2/*/*	3/*/*
PE台数	台	1	120	7.9	11.8	24.8
主記憶量	MB	80	29760	1846	2821	6050
予定CPU時間	秒	3920	35184	15930	16637	17620
実CPU時間	秒	8	35184	15090	15712	16821
入力データ量	MB	7	3856	212	323	692
出力データ量	MB	4	2142	118	179	384
格子点数	千点	54	90672	3012	4654	10020

は長くなるか到着しないようになるのが普通である。そこで，システム内の全ジョブを処理するのに要する時間の最大値（最大積算時間と呼ぶ）と最小値（最小積算時間と呼ぶ）を定義し，最大積算時間に達した場合にはジョブの到着をストップし，最小積算時間以下になった場合にジョブの到着を開始する方式を採用して，より現実的な状況を作ることにした。ジョブの最大積算時間をDMAX，最小積算時間をDMINとした場合，システム内のジョブの最大件数JMAX，最小件数JMINは以下のようにして求める。

$$JMIN = \frac{DMIN}{ARRIV\ (1.0)}$$

$$JMAX = \frac{DMAX}{ARRIV\ (1.0)}$$

なお，シミュレーション実験では，最小積算時間を72時間（3日），最大積算時間を120時間（5日）としている。

ここで，ディバッグジョブ発生率の変化に伴い，発生するジョブ属性と到着率がどのように変化するかを簡単に説明する。第4.1(1)項に述べた理由により，ディバッグジョブのCPU時間は本来のジョブのCPU時間の1/20とする。したがって，ディバッグジョブの発生率0.0の場合の平均CPU時間を1とすると，発生率0.5のときの平均CPU時間は以下に示すように0.525，発生率0.8のときの平均CPU時間は0.240となる。したがって，ディバッグジョブ発生率0.0の場合と同等の負荷状態にするためには，到着時間間隔をそれぞれ0.525倍，0.240倍とする必要がある。

$1.0\times0.5+0.05\times0.5=0.525$

$1.0\times0.2+0.05\times0.8=0.240$

シミュレーション実験では，上記のようにディバッグジョブ発生率を考慮して到着時間間隔を決定する。

5. シミュレーション結果の評価・検討

本章では，スケジューリングアルゴリズムの有効性を実証するために，以下に示す6種類のシミュレーション実験を行い，実験結果の評価・検討を行う。

(1) 基本実験

本スケジューリングアルゴリズムを使用した場合のシステム資源使用状況およびジョブ処理状況の傾向を調べる。また，ジョブクラス別多重度制御方式の場合と比較してシステム資源使用状況がどの程度改善されるかを予測する。

(2) 負荷実験

負荷率が変化した場合のシステム効率，ジョブ処理状況への影響を調べ，特定の条件のジョブにのみ影響を及ぼすことがないことを確認する。

(3) スワップアウト効果実験

起動優先ジョブを起動させる際のスワップアウトジョブ数に対するシステム効率，ジョブ処理状況への影響を調べる。また，SSUの有効利用を図るため，スワップアウトの適正条件を予測する。

(4) PEリザーブ効果実験

起動優先ジョブを最優先で起動するためのPEリザーブは，すべてのジョブの起動を抑止した場合と比較してどの程度システム効率に効果があるかを調べる。

(5) プレステージ効果実験

ジョブ開始時の入力待ち時間を削減するための手段であるプレステージ処理のシステム効率，ジョブ処理状況への影響を調べる。また，SSUの有効利用を図るため，プレステージの適正条件を予測する。

(6) PE占有時間精度実験

ジョブの予想PE占有時間は，要求CPU時間に一定のファクタを掛けて求めるが，ファクタの設定値がシステム効率にどの程度の影響を及ぼすかを調べる。また，ファクタの適正値を予測する。

なお，本稿では，システム結果はデータの最終値で比較するため，システムが定常状態にまで達しており，結果が振動していないことが必要条件となる。このような状態を安定状態と呼び，以下の安定条件で判定する。

本スケジューラは，使用PE台数，CPU要求時間および待ち時間によりジョブの起動からジョブ終了までの流れを制御しているため，システムが安定状態に達した時には経過時間に対する実CPU時間の割合（これをジョブ実行率と定義する）の平均値も安定することが期待できる。そこで，ジョブを使用PE台数が10台以下のジョブ（JOB1と呼ぶ），11～40台のジョブ（JOB2と呼ぶ），41台以上のジョブ（JOB3と呼ぶ）という3つのグループに分類し，JOB1とJOB2の平均ジョブ実行率は少数点以下2桁以上の値が連続10回以上変動がなく，発生件数の少ないJOB3の平均ジョブ実行率は少数点以下1桁以上の値が連続10回以上変動がないことをシミュレーションの安定条件とする。なお，実験結果の比較の都合上，シミュレーション時間はすべて50日分とし，安定条件を満たしているか否かは別に確認する。

5.1 基本実験

基本実験では，本スケジューリングアルゴリズムを使用した場合のシステム資源使用状況およびジョブ処理状況の傾向を見るために，通常の状態を想定した負荷率0.8の実験，混雑時を想定した負荷率1.0の実験，システムの極限状態を想定した負荷率2.0の実験を行った。

(1) 実験条件

基本実験の実験条件として以下のようにパラメータを設定する。

- 負荷率：0.8, 1.0および2.0
- ディバッグ率：0.0, 0.5および0.8
- α_2キューへの遷移パラメータ (FAC1)：0.25

- α_3 キューへの遷移パラメータ (FAC2) : 3.0
- プレステージジョブ数 : 3
- スワップアウトパラメータ : (4, 10)
- FEP 配下の磁気ディスクの入出力速度 : 18MB/S
- PE 占有時間ファクタ : 1.0

ここで，上記パラメータについて，簡単に説明する。

①FAC1，FAC2 の設定により，α_2 キュー，α_3 キューへの遷移条件である WTj_1，WTj_2 は，以下のようになる。

$$WTj_1 = [\text{予定CPU時間} \times \sqrt{\text{PE台数}}] \times 0.25$$

$$WTj_2 = [\text{予定CPU時間} \times \sqrt{\text{PE台数}}] \times 3.0$$

予定 CPU 時間を 1 時間と仮定すると，α_2 キューへの遷移時間は PE 台数が 4 台のジョブは 0.5 時間，PE 台数が36台のジョブは 1.5 時間，PE数が100台のジョブは2.5時間となる。同様に，α_3 キューへの遷移時間は PE 台数が 4 台のジョブは 6 時間，PE 台数が36台のジョブは18時間，PE数が100台のジョブは30時間となる。

②スワップアウトパラメータの (4, 10) は，1 つのジョブを起動する際の最大スワップアウト可能 PE 台数が 4 であり，システムの最大スワップアウト可能 PE 台数が10であることを表す。また，プレステージ処理では，起動待ちジョブの中の優先度の最も高いジョブから指定された数のジョブを常時プレステージする。

③PE 占有時間ファクタは，スケジューラがジョブの要求 CPU 時間からジョブの PE 占有時間を予測する際に使用する。この場合には，予定 PE 占有時間=CPU時間×1.0となる。

(2) 実験結果の評価・検討

本実験では，ディバッグ率が0.0, 0.5, 0.8 の場合についてそれぞれ12のワークロードを使用した実験を行った。ここで，SSU容量は十分にあることを前提とし，どの程度のSSU容量が必要であるかを調べるという観点で実験した。なお，割り当ては8MB 単位で行った。また，ジョブ起動後のプログラムおよび入力データの転送時間は PE の割り当て後に行われるため，PE 使用時間の中に含まれるべき種類のものである。しかし，これらの値は PE の効率的使用の観点からはできるだけ短いことが望まれること，これらの時間を PE 使用時間に含めた場合には PE が有効に使用されている時間の評価が難しくなることから，ジョブ起動後のプログラムおよび入力データの転送時間は PE 使用時間に含めないことにした。したがって，実システムの平均 PE 使用率に比べシミュレーションの平均 PE 使用率は実システムの計算方法に比べて低い値を示すことになる。なお，平均 PE 使用台数および平均SSU使用量は，平均使用量を 1 時間単位に集計して求めている。

(a) PE 使用状況

図5.1 は，基本実験結果の平均 PE 使用台数をワークロードのメモリ量発生タイプごとに平均し，棒グラフで表したものである。

この図から，一般的な傾向として，負荷率と平均 PE 使用台数の関係，ジョブ規模と平均 PE 使用台数の関係を挙げることができる。

低負荷を想定した負荷率 0.8 の場合の平均 PE 使用台数は最も高い場合でも120台程度であるのに対し，負荷率が1.0の場合には125台以上，負荷率が 2.0 の場合には大半が130台以上となっており，負荷率が高くなるにつれて，平均 PE 使用台数も確実に高くなっている。しかし，例外として，ディバッグ率0.0 の場合には，負荷率 1.0 の時より負荷率 2.0 の時の平均 PE 使用台数が若干減少している。これは，ディバッグ率が低いために，他の場合に比べてシステム内のジョブは大規模，長時間ジョブが多くなり，最優先ジョブのための PE リザーブにより起動できないジョブが多かったものと考えられる。

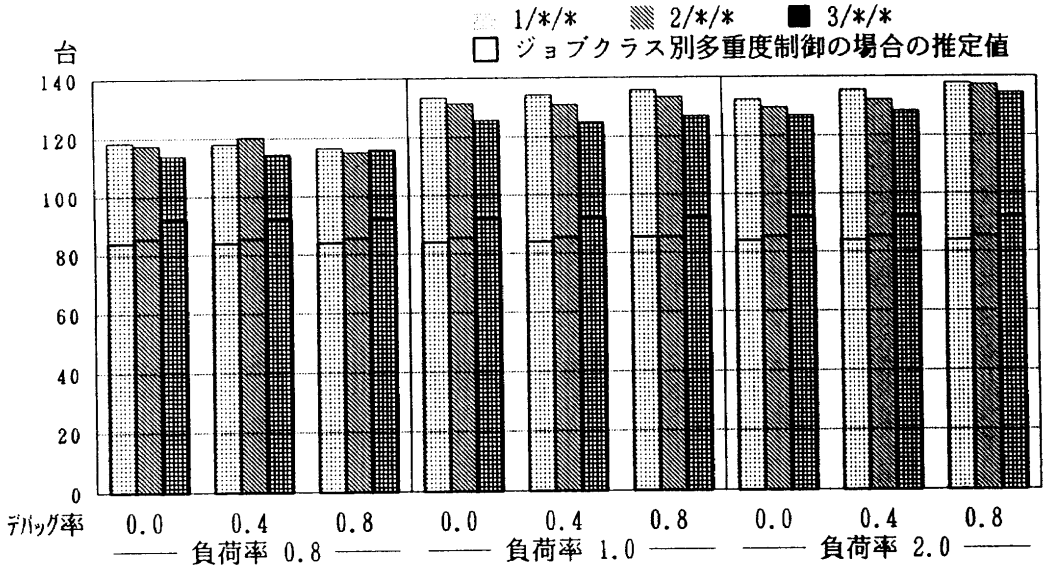


図5.1 基本実験における平均PE使用台数

また，ジョブの規模が大きい場合程平均 PE 使用台数に若干の低下が見られるが，これは，実行待ちジョブが多い場合，規模の小さいジョブの方が起動し易いことに起因しており，負荷率0.8の場合のように実行待ちとなるジョブが少ない場合にはその影響がほとんど現れていない。

なお，図5.1の太枠で囲まれたグラフは，ジョブクラス別多重制御方式でジョブを処理した場合の推定平均 PE 使用台数であり，以下のように算出した。

ジョブクラス多重度制御方式では，多重度分のジョブが実行されていた時には運用PE台数がすべて使用されるように設定することにより混雑時には PE の使用率を100%にすることができるが，ジョブクラスの設定 PE 台数に満たないジョブを実行した時には空き PE が発生する。そのため，推定平均 PE 使用台数 U_{pe} は，運用 PE 台数と PE 有効使用率 E_{rate}（割り当てられた PE が有効に使用されている割合）の積から算出することができる。以下に具体的な算出式を示す。

$$U_{pe} = \text{運用 PE 台数} \times E_{rate}$$

ここで，

$$E_{rate} = \frac{\Sigma_j\ (\text{使用 PE 台数} \times \text{経過時間})}{\Sigma_j\ (\text{占有 PE 台数} \times \text{経過時間})}$$

である。なお，算出式の中の Σ_j (...) は，すべてのジョブ情報の加算を意味しており，基本実験で得られた課金情報を使用して求めた。

この結果，推定平均 PE使用台数を求める際の条件を満たしている負荷率1.0，負荷率2.0の場合には，実験結果における平均 PE 使用台数は，ジョブクラス別多重度制御方式の場合の平均 PE 使用台数の推定値より平均43.90台分勝っていた。平均 PE 使用台数はシステムのジョブ処理量に直接影響を及ぼすためシステム効率の重要な指標となるが，上記結果を見る限りジョブクラス別多重度制御方式に比べて本アルゴリズムの優位性は明瞭である。

(b)　SSU 使用状況

SSUは主にプレステージ，入出力，スワップアウトに使用されるため，実験結果もこれらの使用量について評価する。

基本実験の平均プレステージ使用量を図5.2に示す。プレステージ使用量は，プレステージジョブ数を3としていることから負荷率よりもワークロードによる平均 PE 台数の違いの影響が大きく表れている。プレステージ量を負荷率で比較すると，負荷率1.0の場合が最も多く，ついで負荷率2.0，負荷率0.8の順になっているが，これはプレステージされたジョブの規模の違いと考えられる。負荷率0.8ではジョブの競合が少ないためプレステージされるジョブは大規模のものに限定されないが，負荷率1.0では処理されにくい大規模ジョブがプレステージされ，負荷率2.0では全体的に処理が滞ったため比較的小規模なジョブもプレステージされたものと考えられる。図5.3に平均スワップアウト使用量を示す。負荷が低い場合には α_3 キューまで進むジョブが少ないためスワップアウト使用量は極めて少いが，負荷率が高くなるに従って単純に増加している訳でもない。これは，ひとつのジョブを起動する際のスワップアウト PE 台数の制限を4としたために，比較的小規模なジョブが起動優先ジョブとならない限りスワップアウトの条件が満たされないためと考えられる。このことは，小規模ジョブの発生率が高い 1/*/* のスワップアウト使用量が相対的に多いことからも裏付けられる。

図5.4に平均入出力使用量を示す。入出力使用量はプレステージ使用量に比べて極端に少ない。これは，ジョブの CPU 時間がジョブクラス制限値という形で制限されていないために，CPU時間に対する相対的な入出力量が減少したためと考えられる。このことは，ディバッグ率が高くなるのに伴い入出力量が相対的に増加していることからも明らかである。したがって，適切なSSU 容量を見積もる際には，入出力量はプレステージ量やスワップアウト量程厳密に行う必要はないが，実運用において CPU 時間を制限する場合には増加することを考慮すべきである。

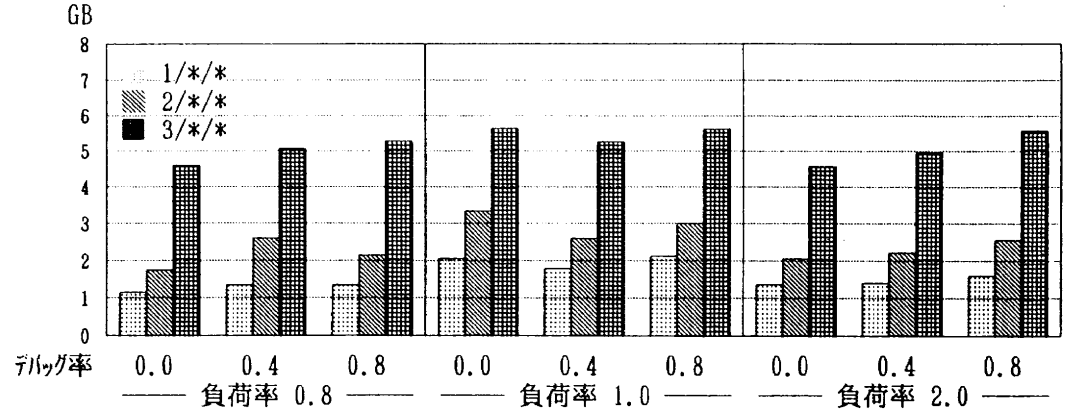


図5.2　基本実験における平均プレステージ使用量

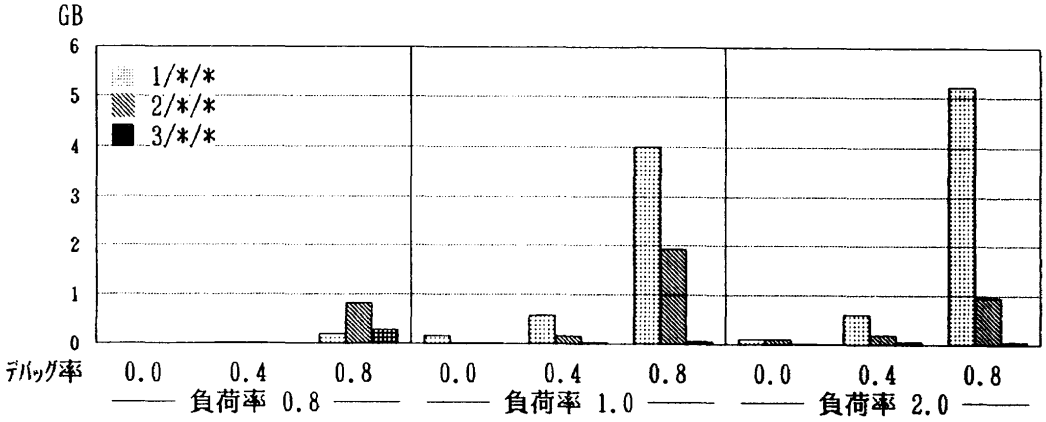


図5.3 基本実験における平均スワップアウト使用量

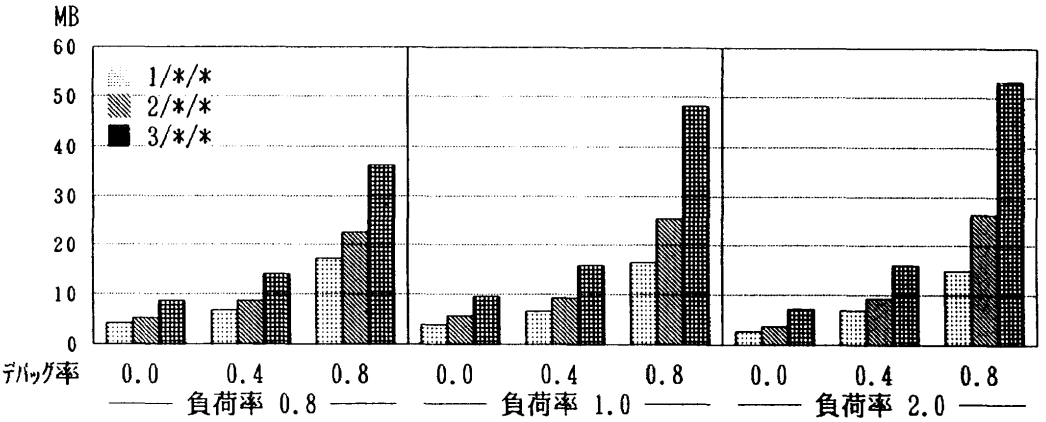


図5.4 基本実験における平均入出力使用量

(c) ジョブ処理状況

図5.5に基本実験結果のジョブ実行率を，図5.6にα_3キューまで遷移したジョブの割合（以後，α_3キューへの遷移率と呼ぶ）を示す。

本章の冒頭で述べたように，ジョブ実行率は経過時間に対する実CPUの割合であるため，値が1に近い程待ち時間が少なくジョブは良好に処理されていると考えることができる。負荷率0.8の場合のジョブ実行率は，JOB1は0.89以上，JOB2は0.62以上，JOB3は0.25以上を示しており，単純計算すると，PE要求台数が41台以上であるJOB3でさえCPU時間の4倍程度で処理されていることになる。これに対し，負荷率1.0の場合のジョブ実行率は，JOB1は0.53以上，JOB2は0.17以上，JOB3は0.05以上を示しており，JOB1のジョブはCPU時間の2倍程度，JOB2のジョブはCPU時間の5倍程度，JOB3の処理時間はCPU時間の20倍程度かかっていることになる。また，負荷率2.0の場合には，JOB1は0.14以上，JOB2は0.03以上，JOB3は0.01以上となり，ジョブの処理状況は全般的に悪化している。

また，α_3キューへの遷移率は，負荷の上昇に伴って高くなる傾向があり，ジョブの規模が小さいワークロードの場合程この傾向が顕著に現れている。この率が極端に高くなった場合には，優先方式の異なる3本のキューを用いている意味がなくなる上，起動優先ジョブの処理が増加するためシステムオーバヘッドも増加する。したがって，α_3キューへの遷移率は一定値以下に抑えることが必要となる。

一般的に，ジョブ実行率は負荷が低い場合には高く，負荷が高い場合には低くなるのは当然の結果であるため，これらの数値から単純にスケジューラの良否を判定することはできない。そこで，負荷率の変化に伴いジョブ処理状況がどの程度変化するかを，第5.2項の実験により詳細に調べる必要がある。

(d) 安定条件

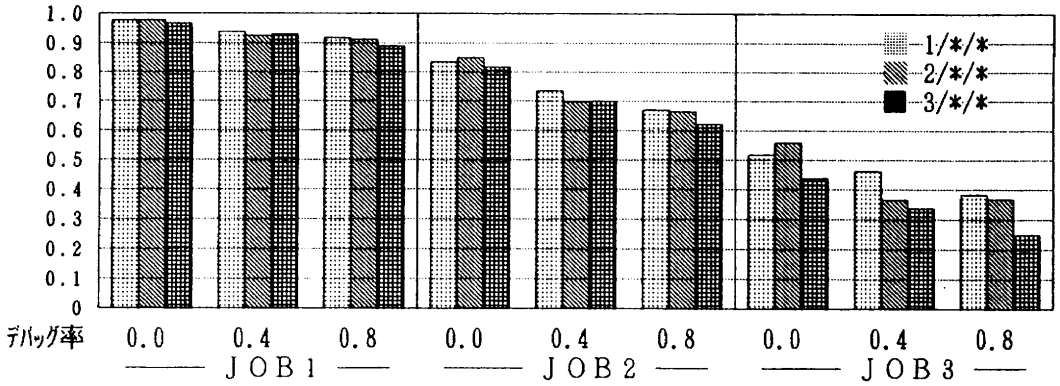
シミュレーションの安定条件に関しては，負荷率0.8の場合には1日から41日程度，負荷率1.0の場合には2日から22日程度，負荷率2.0の場合には1日から44日程度ですべて条件を満たしたことを確認した。

(3) 基本実験から得られた結論

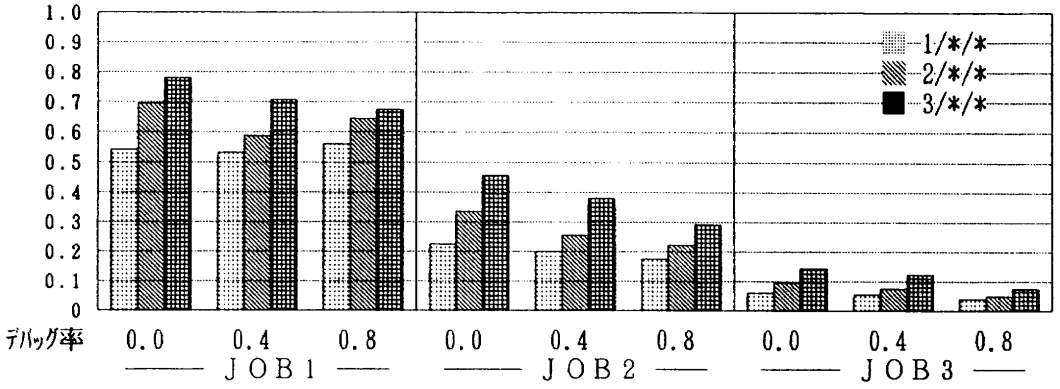
本実験結果から以下の事柄が確認された。

①PEの有効利用に関して，本方式はジョブクラス別多重度制御方式に比べて顕著な優位性がある。

②負荷が十分にあれば，ワークロードの規模に関係なく



(1) 負荷率0.8



(2) 負荷率1.0

(3) 負荷率2.0

図5.5　基本実験におけるジョブ実行率

PEの高利用が可能であると考えられる。

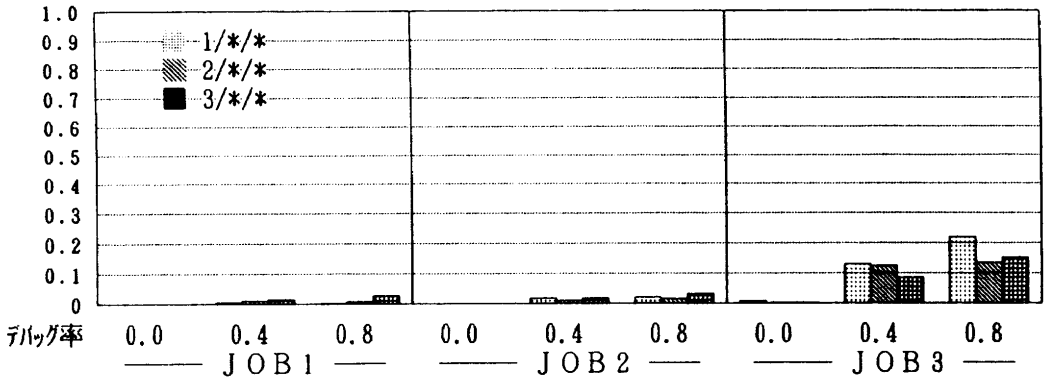
③α_3キューへの遷移率を一定に抑えるために，混雑時にはα_2キューやα_3キューへの遷移パラメータを大きめに設定する必要がある。

5.2　負荷実験

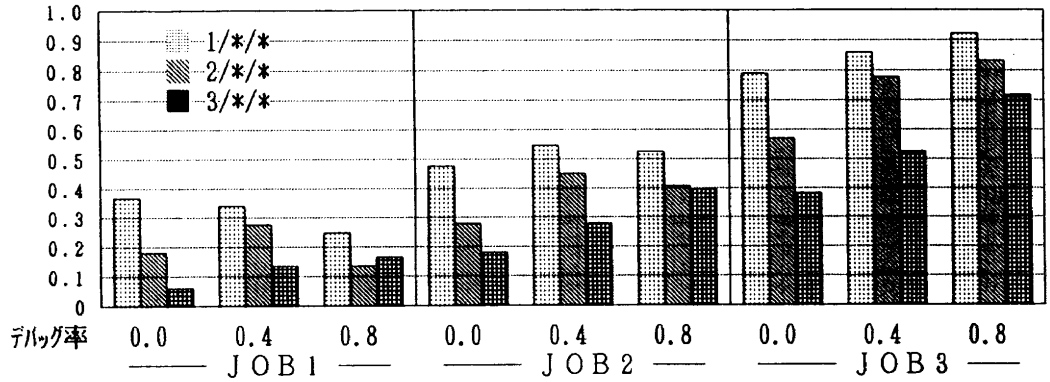
ほとんど待つこと無しにジョブが処理されていたシステムでも，投入ジョブ数の急激な増加や大規模ジョブ，長時間ジョブの増加に伴う負荷率の上昇により，経過時間が大幅に増加することがある。負荷が増加すればジョブの平均経過時間が長くなるのは当然の事であるが，特定の条件を満たすジョブの経過時間のみが大幅に増加するようなスケジューラは好ましいものではない。このような観点から，考案したスケジューリングアルゴリズムが負荷率の変化にどのように影響されるのか，その影響はワークロードによりどの程度異なるのかを確認する。

(1) 実験条件

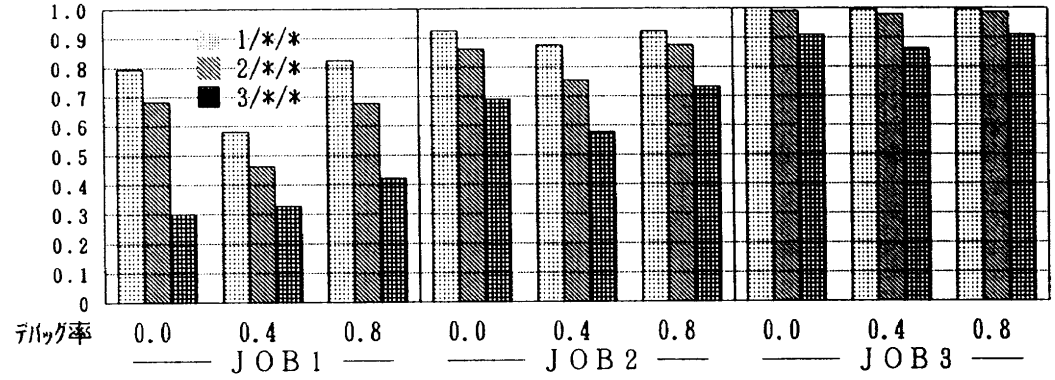
基本実験は負荷率0.8, 1.0, 2.0で行った。これに対し本実験では，2種類のワークロードを使用して0.8から0.2刻みで2.0までの負荷率での実験を行い，その影響を比較



(1) 負荷率0.8



(2) 負荷率1.0



(3) 負荷率2.0

図5.6 基本実験におけるα_3キュー遷移率

する。なお，ワークロードは平均PE要求台数および平均CPU要求時間が最も小さい1/1/1と最も大きい3/2/2を使用し，その効果をさらに顕著な形で実行するためにワークロード1/1/1はディバッグ率0.8で，ワークロード3/2/2はディバッグ率0.0で行うことにした。したがって，負荷実験の実行条件として以下のようにパラメータを設定した。

- 負荷率：0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0
- ディバッグ率：0.8（1/1/1の場合）または0.0（3/2/2の場合）
- α_2キューへの遷移パラメータ (FAC1)：0.25
- α_3キューへの遷移パラメータ (FAC2)：3.0
- プレステージジョブ数：3
- スワップアウトパラメータ：(4, 10)
- FEP配下の磁気ディスクの入出力速度：18MB/S
- PE占有時間ファクタ：1.0

(2) 実験結果の評価・検討

(a) PE使用状況

図5.7(1)に負荷実験結果の平均PE使用台数を示す。ワークロード1/1/1を使用した実験では，負荷率が高くなるに従って僅かづつではあるが平均PE使用台数が増加して

いる。これに対し，ワークロード3/2/2を使用した実験では，低下する場合もある。これは，起動優先ジョブのためのPEリザーブの影響であり，ジョブの規模が比較的小さい場合には問題がないが，規模が大きい場合にはジョブのターンアラウンドタイムの保証と引き換えにPE稼働率が若干低下することを示している。

(b)　ジョブ処理状況

図5.7(2)，(3)にワークロードが1/1/1，3/2/2の場合のジョブ実行率を示す。ワークロードが1/1/1の場合も3/2/2の場合も，負荷率の上昇に伴いJOB1，JOB2，JOB3のジョブ実行率は同様に低下していることが判る。

図5.8(1)，(2)にワークロードが1/1/1，3/2/2の場合のα_3キュー遷移率を示す。負荷の上昇に伴いα_3キューに達したジョブは予想通り増加しているが，ワークロード3/2/2の場合にのみJOB1の値が負荷率1.4を境に減少している。これは，ジョブの規模が大きく，負荷が高いためPE要求台数の大きいジョブが起動されずに大量に残っているため，規模の小さいジョブが起動されやすくなったためと考えられる。また，ワークロードが1/1/1の場合にはワークロード3/2/2の場合に比べて平均PE要求台数，平均CPU要求時間が小さいため，発生するジョブ数は相対的に多い。そのため，JOB1でα_3キューに達しているジョブの割合は，ワークロードが1/1/1の場合には負荷率が1.4程度でも80%以上であるのに対し，ワークロード3/2/2の場合には負荷率2.0でも20%以下となっている。

(c)　安定条件

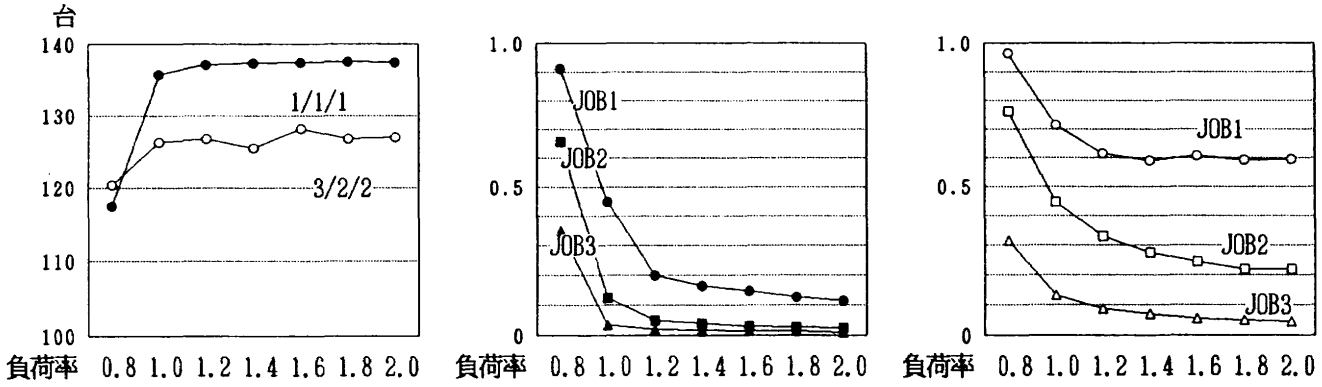
シミュレーションの安定条件に関しては，ワークロードが1/1/1の場合には1日から6日程度で，ワークロードが3/2/2の場合には6日から47日程度ですべて条件を満たしたことを確認した。この時間から見てもワークロードが1/1/1の場合のジョブ処理状況は安定しているが，3/2/2の場合には起動優先ジョブのためのPEリザーブの影響によってジョブ処理状況が少し不安定になっていることが判る。

(3)　負荷実験から得られた結論

負荷実験結果の検討から以下の事柄が確認された。

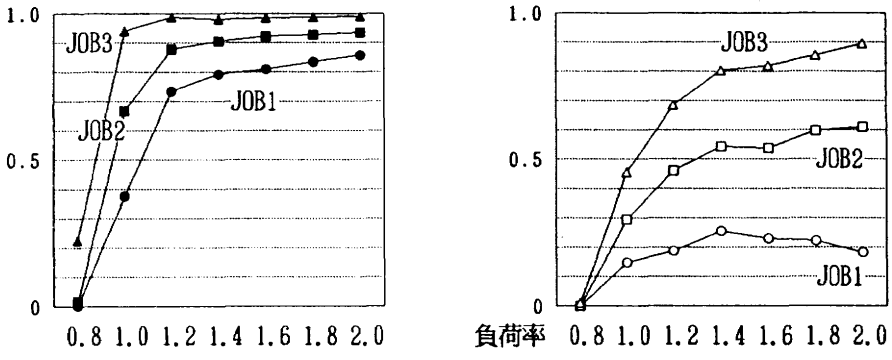
①ジョブの規模が小さいワークロードの場合には負荷率に応じてPEの利用率も高い値を示している。一方，ジョブの規模が大きいワークロードの場合には，起動優先ジョブのためのPEリザーブによりジョブのターンアラウンドタイムと引き換えに平均PE使用台数の低下が見られるが，低下の程度は小さく，許容範囲内であると考えられる。

②負荷率の上昇によりジョブ実行率は低下するが，ジョブの規模等に係わらず全体的に低下しており，本アルゴリ



(1)　平均PE使用台数　　(2)　ジョブ実行率（1/1/1の場合）　　(3)　ジョブ実行率（3/2/2の場合）

図5.7　負荷実験結果



(1)　α_3キュー遷移率（1/1/1の場合）　　(2)　α_3キュー遷移率（3/2/2の場合）

図5.8　負荷実験結果

ズムは負荷率の変化にも柔軟に対処可能であると判断できる。

③α_3キューのジョブは優先実行ジョブであるにも係わらず，大半のジョブが優先実行ジョブとなった場合には優先実行の効果が期待できない。そのため，α_3キューに達するジョブ数があまり増加しないように，ジョブのキュー遷移パラメータを調節することが望まれる。しかし，負荷率が同じでもジョブ数が多い場合にはα_3キューに達するジョブ数が増加するため，キュー遷移パラメータは，負荷率，ジョブ数等を考慮して決定しなければならない。

5.3 スワップアウト効果実験

本スケジューラでは，適切なジョブターンアラウンドタイムを保証するために，起動優先ジョブの起動時にスワップアウトによる起動を試みている。しかし，スワップアウトによる起動はシステム資源の有効利用の観点からは好ましいものではない。そこで，スワップアウトによりシステム状況が悪化することはないのか，ジョブのターンアラウンドタイムの目的は達成されているのかをスワップアウト効果実験により確認する。

(1) 実験条件

基本実験はスワップアウトパラメータ＝(4, 10)で行った。これに対し，本実験ではスワップアウトパラメータ＝(0, 0)，(4, 10)，(8, 20) の場合について行い，その効果を比較する。なお，負荷が極端に高い場合には実行ジョブの大半もα_3ジョブとなるためにスワップアウト可能なジョブが少なく，効果が期待できない。ディバッグジョブが多い場合にもジョブ発生件数が多くなり，ジョブがα_3ジョブとなる確率が高くなる。そこで，本実験は負荷率1.0，ディバッグ率0.0の条件で行うことにした。したがって，スワップアウト効果実験の実行条件として以下のようにパラメータを設定した。なお，ワークロードは全12種類を使用する。

- 負荷率：1.0
- ディバッグ率：0.0
- α_2キューへの遷移パラメータ (FAC1)：0.25
- α_3キューへの遷移パラメータ (FAC2)：3.0
- プレステージジョブ数：3
- スワップアウトパラメータ：(0, 0), (4, 10) および(8, 20)
- FEP配下の磁気ディスクの入出力速度：18MB/S
- PE占有時間ファクタ：1.0

(2) 実験結果の評価・検討

(a) PE使用状況

図5.9(1)にスワップアウト効果実験結果の平均PE使用台数を示す。平均PE使用台数は，最大スワップアウトPE台数が大きくなるに従い，若干ではあるが向上している。この結果からは，スワップアウトによる顕著な悪影響を見ることはできない。

(b) SSU使用状況

図5.9(2)に本実験結果の平均スワップアウト使用量を示す。使用量は最大スワップアウトPE台数が0の場合には当然のことながら0であり，最大値が大きくなるに従って増加している。この結果からも，スワップアウトPE台数の制限がSSU使用量を制限するために有効であることが判る。

(c) ジョブ処理状況

図5.10に本実験結果のジョブ実行率を示す。どのジョブの実行率も最大スワップアウトPE台数が大きくなるに従って僅かではあるが向上している。

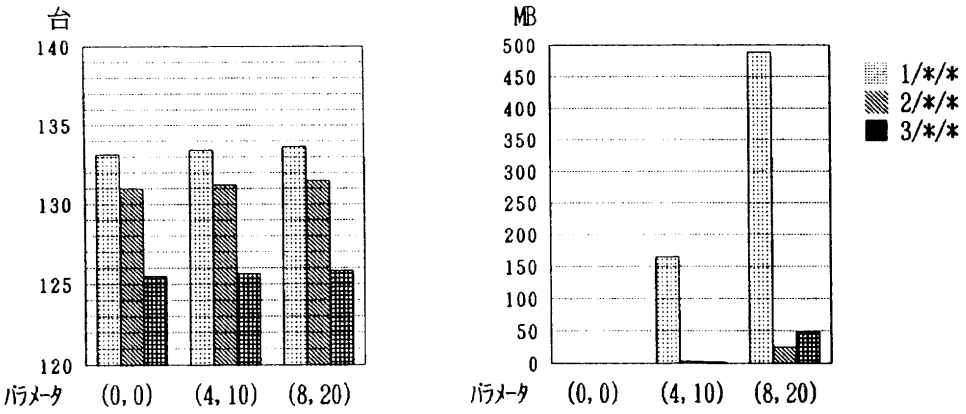
(d) 安定条件

シミュレーションの安定条件に関しては，6日から21日程度ですべて条件を満たしたことを確認した。

(3) スワップアウト効果実験から得られた結論

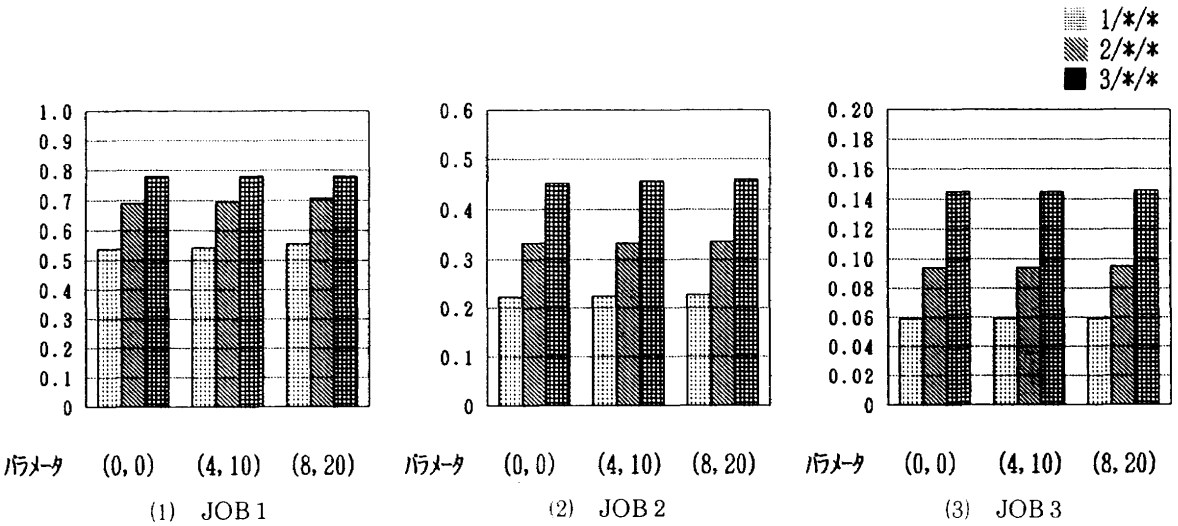
スワップアウト効果実験結果の検討から，以下の事柄が確認された。

①最大スワップアウトPE台数を大きく設定しても，ジョ



(1) 平均PE使用台数　(2) スワップアウト使用量

図5.9 スワップアウト効果実験のシステム資源使用量



(1) JOB 1　(2) JOB 2　(3) JOB 3

図5.10　スワップアウト効果実験のジョブ実行率

ブ実行率は僅かしか向上しないため，劇的な効果は期待できない。

②スワップアウトに一定の制限を設けているため，スワップアウトによるPEの利用率，ジョブ実行率の顕著な低下は見られない。

5.4　起動優先ジョブのためのPEリザーブ効果実験

起動優先ジョブが起動できない場合に他のすべてのジョブの起動を抑止する方式と起動優先ジョブの実行に必要なPEのみをリザーブする方式の実験を行い，PEリザーブ方式の優位性を確認する。

(1) 実験条件

PEのリザーブ効果を見るためには，システム状況はある程度混雑している必要があり，ジョブの規模は大きい方が影響が出易いと考えられる。そのため，負荷率は1.0および2.0とし，ディバッグ率は0.0として実験する。以下に示すように，PEリザーブ効果実験の実行条件のパラメータを設定する。なお，ワークロードは全12種類を使用する。

- 負荷率：1.0および2.0
- ディバッグ率：0.0
- α_2 キューへの遷移パラメータ (FAC1)：0.25
- α_3 キューへの遷移パラメータ (FAC2)：3.0
- プレステージジョブ数：3
- スワップアウトパラメータ：(4, 10)
- FEP配下の磁気ディスクの入出力速度：18 MB/S
- PE占有時間ファクタ：1.0

(2) 実験結果の評価・検討

(a) PE使用状況

図5.11(1)に負荷率1.0の場合の平均PE使用台数を，図5.12(1)に負荷率2.0の場合の平均PE使用台数を示す。負荷率1.0の場合，ワークロードパラメータが1/*/*および2/*/*の場合にはPEリザーブ効果は少ないが，発生ジョブの規模が大きい3/*/*のときには5台分以上の効果を示している。負荷率2.0の場合には負荷率1.0の場合より効果は顕著になっており，特に3/*/*のときには11台分以上の効果を示している。

(b) ジョブ処理状況

図5.11(2)～(4)に負荷率1.0の場合のジョブ実行率を，図5.12(2)～(4)に負荷率2.0の場合のジョブ実行率を示す。

負荷率1.0の実験では，ワークロードパラメータが1/*/*，2/*/*の場合には，全体的なジョブ実行率の効果が僅かながら見られる。また，3/*/*の場合には待たされている小規模ジョブの起動の機会がPEリザーブにより増え，その結果JOB1の実行率が大きく向上した反面，JOB3の実行率が僅かに低下している。

負荷率2.0の実験でも，負荷率1.0と同様の結果となっているが，ワークロードパラメータが3/*/*の場合には，JOB3だけでなくJOB2も僅かながら実行率が低下している。

(c) 安定条件

シミュレーションの安定条件に関しては，6日から42日程度ですべて条件を満たしたことを確認した。

(3) PEリザーブ効果実験から得られた結論

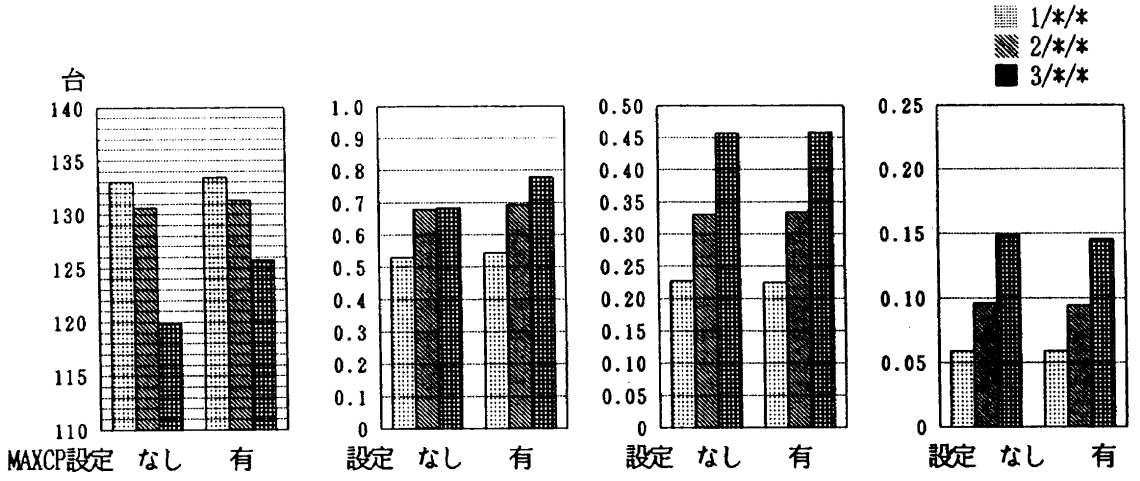
PEリザーブ効果実験結果の検討から，以下の事柄が確認された。

①PEリザーブの平均PE使用台数に対する効果は，発生するジョブの規模が大きく，負荷率が高い場合ほど顕著になる。

②PEリザーブのジョブ実行率に対する効果は，小規模ジョブに有効である。

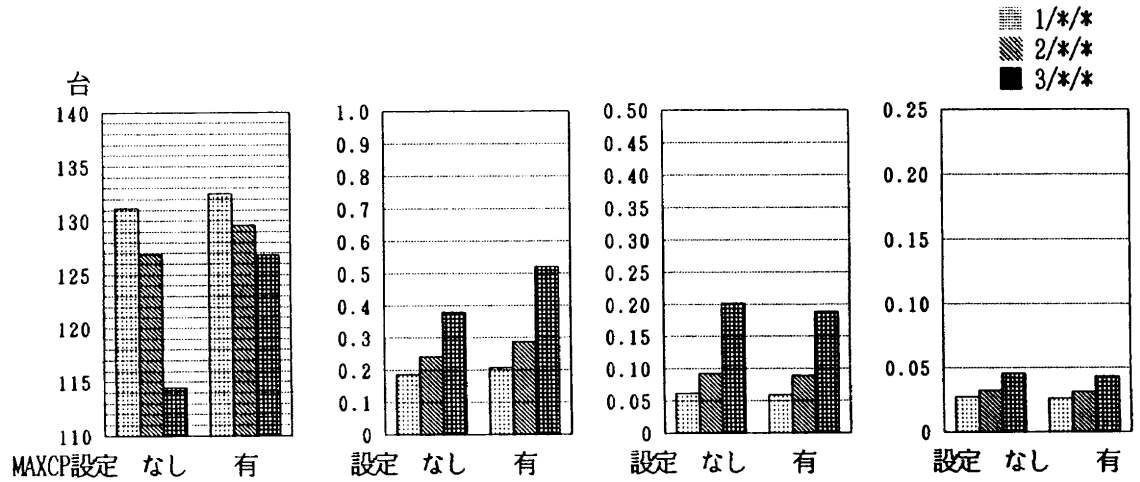
5.5　プレステージ効果実験

プレステージは，PEを有効に使用するための手段とし



(1) 平均PE使用台数 (2) JOB1の実行率 (3) JOB2の実行率 (4) JOB3の実行率

図5.11 PEリザーブ効果実験結果（負荷率1.0の場合）



(1) 平均PE使用台数 (2) JOB1の実行率 (3) JOB2の実行率 (4) JOB3の実行率

図5.12 PEリザーブ効果実験結果（負荷率2.0の場合）

て期待されている。本項では，最大プレステージジョブ数を変化させた実験を行い，プレステージの効果がどの程度期待できるものかを予測する。

(1) 実験条件

FEPのディスク－SSU間，SSU－PE間のファイル転送速度の差が大きい場合，ジョブの入出力量が相対的に高い場合（ディバッグジョブが多い場合，ジョブのCPU時間が制限されている場合などがこれに相当する）にはプレステージの効果が期待できる。ジョブの入出力量が少ない場合には，ファイルの転送速度が多少低くてもPEの利用率に悪影響を及ぼすことはないが，ジョブの入出力が多くファイルの転送速度の差が大きい場合には，転送速度の差はジョブが起動されてから実際に開始されるまでに要する時間を決定するため，PEの実質的な利用率に直接の影響を及ぼす。そこで，FEPのディスク－SSU間のファイル転送速度を3MB/Sと仮定して実験を行うことにした。この値は並列入出力を行わない場合の4.5MB/Sから決定した推定実効速度であり，今までの経験に基づき推定した。また，ジョブ入出力を相対的に多い条件で行うためにディバッグ率は0.8とする。

以上の結果，実験条件として以下のようにパラメータを設定する。なお，本実験はプレステージジョブ数を変化させるためワークロード当たりの実験数が多くなる。そこで，代表的なワークロードとして1/1/1の1種類を使用する。

- 負荷率：1.0
- ディバッグ率：0.8
- α_2 キューへの遷移パラメータ (FAC1)：0.25
- α_3 キューへの遷移パラメータ (FAC2)：3.0
- プレステージジョブ数：0, 1, 2, 3, 4, および5
- スワップアウトパラメータ：(4, 10)
- FEP配下の磁気ディスクの入出力速度：3MB/S
- PE占有時間ファクタ：1.0

(2) 実験結果の評価・検討

(a) PE 使用状況

図5.13(1)にプレステージ効果実験結果の平均 PE 使用台数を示す。プレステージジョブ数の増加に伴って PE 使用台数は高くなっているが，ジョブ数 3 の近辺からほぼ横這い状態となっている。この場合プレステージは PE 4 台分程度の効果となっているが，先に述べたようにジョブの入出力量が相対的に高い場合にはそれに伴う効果が期待できる。

(b) SSU 使用状況

図5.13(2)にプレステージ効果実験結果の平均プレステージ使用量を示す。プレステージ使用量はプレステージジョブ数にほぼ比例している。

(c) ジョブ処理状況

図5.11(3)にプレステージ効果実験結果のジョブ実行率を示す。プレステージジョブ数が高くなるに従いジョブ実効率も高くなっているが，効果の最も顕著な JOB1 の場合では，プレステージジョブ数 3 の近辺からほぼ横這い状態となっており平均 PE 使用台数の結果と一致している。

(d) 安定条件

シミュレーションの安定条件に関しては，2 日～4 日半程度ですべて条件を満たしたことが確認された。

(3) プレステージ効果実験から得られた結論

プレステージ効果実験結果の検討から，以下の事柄が確認された。

①本実験のように入出力量がかなり少ない場合でも，プレステージにより 4 台分程度の PE 利用率の向上が期待できる。したがって，ジョブの CPU 時間制限を行った場合やジョブの平均 CPU 時間が短い場合には，プレステージの効果は極めて高いと予想される。

②SSU の容量に余裕があれば，常時 3 ジョブ分程度のプレステージを行うことが望ましい。

5.6 PE 占有時間精度実験

本アルゴリズムでは，要求 CPU 時間に一定のファクタを掛けて PE 占有時間を予測するが，ファクタの設定値によりシステム効率がどの程度影響を及ぼされるのか，また適切な設定値はある程度予測できるのかを調べる。

(1) 実験条件

本実験では，ワークロードはジョブ規模およびジョブ時間の平均が最も小さい 1/1/1 を使用し，ディバッグ率も 0.8 で行うことにした。そして，ファクタ値を 0.8～1.4 まで 0.1 刻みに変化させた実験を行うことにした。

以上の結果，実験条件として以下のようにパラメータを設定した。

- 負荷率：2.0
- ディバッグ率：0.8
- α キューへの遷移パラメータ (FAC1)：0.25
- α_3 キューへの遷移パラメータ (FAC2)：3.0
- プレステージジョブ数：3
- スワップアウトパラメータ：(4, 10)
- FEP 配下の磁気ディスクの入出力速度：3 MB/S
- PE 占有時間ファクタ：0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4

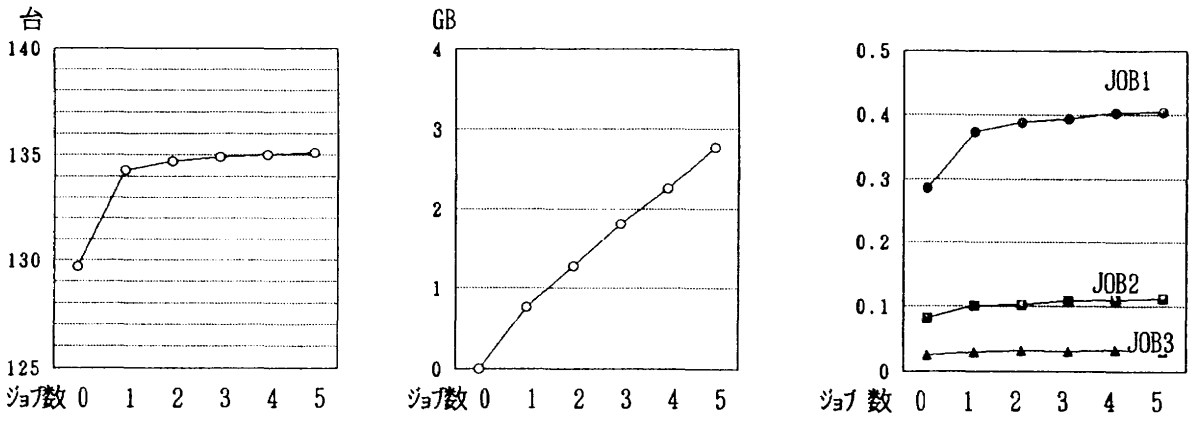
(2) 実験結果の評価・検討

(a) PE 使用状況

図5.14(1)に PE 占有時間精度実験結果の平均 PE 使用台数を，同図(3)に予想 PE 占有時間内に終了したジョブの割合を示す。同図(3)から，本実験での PE 占有時間ファクタの適正値は 1.0 から 1.1 の間にあったと考えられるが，PE 占有時間ファクタが 1.0 に近づくにつれて PE 使用台数は高くなっており，それ以上ではほぼ横這い状態となっている。

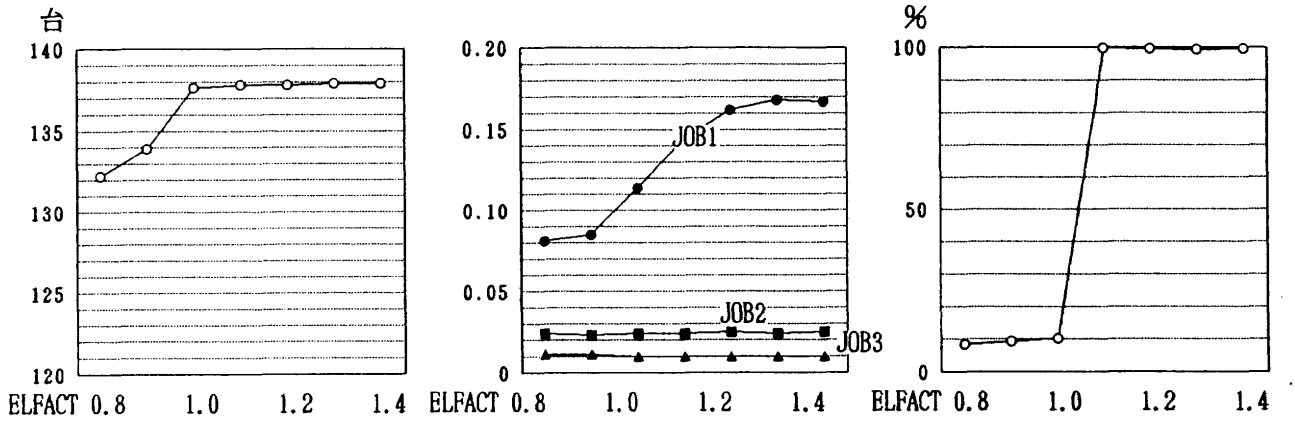
(b) ジョブ処理状況

図5.14(2)に PE 占有時間精度実験結果のジョブ実行率を示す。PE 占有時間ファクタの適切な設定は，小規模ジョブの実行状況の改善に有効であることが判る。



(1) 平均 PE 使用台数　(2) 平均プレステージ量　(3) JOB 実行率

図5.13 プレステージ効果実験結果



(1) 平均PE使用台数　(2) JOB実行率　(3) 予想経過時間内終了率

図5.14 経過時間精度実験結果

(c) 安定条件

シミュレーションの安定条件に関しては，1日～1日半程度ですべて条件を満たしたことが確認された。

(3) PE占有時間から得られた結論

PE占有時間精度実験結果の検討から，以下の事柄が確認された。

①PE占有時間ファクタを適切に設定することによりPE6台分程度のシステム効率の向上が期待できる。

②PE占有時間ファクタの設定はあまり厳密に考える必要はなく，適切な設定値が判らない場合には大きめに設定する方が効果的である。

5.7 シミュレーションプログラム

本章で使用した実験結果は，スケジューリングアルゴリズムを評価するために作成したシミュレーションプログラムを使用している。このプログラムはFORTRANで記述した約2000ステップのプログラムであり，事象を制御するためのメインプログラムと12のサブルーチンから構成されている。代表的なサブルーチンには，ジョブスケジューリングプログラム，ジョブ属性の決定および発生ジョブや実行終了ジョブの統計情報の収集を行うジョブ属性決定プログラム，入出力所要時間の計算およびSSU使用統計情報の収集を行うSSU管理プログラムがある。

本プログラムを使用して行ったシミュレーション実験の所要CPU時間は処理するイベント数に依存するためジョブの処理件数が増えるに従って増加する傾向にある。また，シミュレーションはVP-2600を使用したTSS処理で実行したが，基本実験のCPU使用時間は負荷率0.8の場合で0.459秒～4.412秒，負荷率1.0の場合で0.529秒～6.635秒，負荷率2.0の場合で0.629秒～60.139秒であり，50日分のシミュレーション実験であるにも係わらず極めて短時間で処理していることから，本シミュレーションプログラムの制御方式がシステム評価手法として極めて有効であったことが判る。なお，ジョブ実行率等の計算はシミュレーション結果から出力したジョブ情報（ジョブごとのシステム資源要求量，ジョブ投入時刻，起動時刻，終了時刻等の情報）を使用して別ジョブで処理しており，シミュレーションの所要時間には含まれていない。

6. おわりに

NWTシステムは，140台のPEから構成されている並列計算機システムである。各PEはピーク性能1.7 GFLOPSという高性能のベクトル計算機であるため，運用を行う上ではPEの有効利用を図ることのできるスケジューラが必須となる。しかし，このような規模の並列計算機システムは世界でも実運用の例がないこと，汎用計算機のスケジューリングアルゴリズムをそのまま適用できないことから，新たなスケジューラを開発する必要があった。

並列計算機システム用スケジューラの開発では，PEの有効利用および適切なジョブターンアラウンドタイムの保証を第一優先で考えた。その結果，無駄な空きPEを発生しないPE割り当て方式と待ち時間に基づくジョブ実行制御方式を組み込んだスケジューラを考案した。また，NWTシステムでの検証実験を最小限度に抑えるために，検証実験に先立って，スケジューリングパラメータの効果，スケジューラの有効性に関する実験をソフトウェアシミレータを使用して行った。その結果，以下の事柄が確認できた。

①本スケジューラの処理能力はジョブクラス別多重度制御方式の場合と比較してPE40台分程度向上する。

②負荷率の変化にも対応可能であり，ジョブの規模等に関係なく公平に処理することができる。

③負荷が高くなると平均PE使用台数も上昇する。まれに起動優先ジョブのために低下する場合もあるが，低下の度合いは極めて少ない。

④スワップアウトはターンアラウンドタイムの適正化に有効に作用する。また，スワップアウトに一定の制限を設

けることにより，システム資源利用率やジョブ実行率の低下は阻止できる。

⑤起動優先ジョブのためのPEリザーブは，他のジョブの起動を完全に抑止する場合に比べ最大PE10台分程度の処理能力の向上に匹敵する。

⑥常時3ジョブ分程度をプレステージしておくことにより，PE4台分程度処理能力が向上する。

上記の事柄を総合した結果，本スケジューラの有効性は極めて顕著であり，NWT用のスケジューラとして組みむべきであると結論した。

＜**謝辞**＞

本スケジューリングアルゴリズムの検討・開発にあたり，財団法人高度情報科学技術研究機構副理事長の三好甫氏から有益なる助言をいただいた。この場を借りて感謝の意を表する。

参　考　文　献

1) 三好　甫；CFDの推進に必要な計算機性能，NAL SP-13, pp.1～26, 1990年9月
2) 三好　甫；航技研超高速数値風洞(UHSNWT)の構想，NAL TR-1108, 1991年5月
3) 三好　甫；数値風洞；要求要件と概略，NAL SP-16, pp.91～98, 1991年12月
4) 三好　甫，吉岡義朗，池田正幸，高村守幸；数値風洞のハードウェア，NAL SP-16, pp.99～106，1991年12月
5) 福田正大，末松和代，土屋雅子，大空　瞭，工内　隆，坂本喜則；数値風洞のオペレーティングシステム，NAL SP-16, pp.107～114, 1991年12月
6) 福田正大，中村　孝，吉田正廣，岡田　信，中村修一；数値風洞の言語ソフトウェア，NAL SP-16, pp.115～122, 1991年12月
7) 吉田正廣，中村　孝，福田正大，中村修一，岡田　信；NWT向け並列Fortranプログラミングについて，NAL SP-19, pp.247～252, 1992年12月
8) 中村　孝，吉田正廣，福田正大，中村修一，村瀬丈夫，松崎達哉；NWT並列Fortranに基づく並列評価，NAL SP-19, pp.253～258, 1992年12月
9) 土屋雅子，末松和代，末松俊二，畠間晴夫，森重博司，山口　靖，軽部行洋；NWTのユーザインタフェース実現方式，NAL SP-19, pp.259～264, 1992年12月
10) 情報処理ハンドブック，情報処理学会，1989年11月
11) 中村　孝，吉田正廣，福田正大，村瀬丈夫，松崎達哉；CFDプログラムによるNWTの性能評価，NAL SP-22, pp.179～184, 1994年3月
12) 土屋雅子；NSシステム用ジョブ・ジョブステップ・スケジューラの開発，NAL TR-977, 1988年6月
13) JUHNSON KGIZ ; Distributions in Statistics ; Continuous Univariate Distribution-2

航空宇宙技術研究所報告1277号

平成7年9月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺東町7丁目44番地1
電話三鷹(0422)47-5911(大代表)〒182

印刷所 株式会社三興印刷
東京都新宿区西早稲田2-1-18

Printed in Japan