2000 年 9 月 8 日

サイエンティフィック・システム研究会
科学技術計算分科会 第 1 回会合資料

宇宙科学研究所 VPP800/12 システムにおけるジョブ制御報告

宇宙科学研究所
宇宙科学企画情報解析センター
篠原 育
iku@stp.isas.ac.jp

宇宙科学研究所
宇宙輸送研究系
高速流体力学部門
藤井 孝蔵
fujii@flab.eng.isas.ac.jp

富士通株式会社
科学システム統括部
宇宙システム部
松井 秀司
matsui@ssd.se.fujitsu.co.jp

1. はじめに

宇宙科学研究所 計算機センタは、宇宙科学計画に参加する多数の人々から利用される共同利用研究施設であり、衛星管制・取得データ処理はもとより、軌道計算、数値シミュレーションなど、幅広い分野にわたり利用されている。

数値シミュレーションに代表される科学技術計算の需要は年々増加傾向にあり、大規模・高性能な計算機が要求される。この要求に応えるため、平成 11 年度に Fujitsu VPP500/7 システムを Fujitsu VPP800/12 システムにリプレースし運用を開始している。

科学技術計算の高速計算手法・規模は様々であるが、計算機資源の有効利用と高い利用効率の実現は、計算機センタとしての役割である。また、CPU 課金を実施しており、課金単価に応じたジョブの優先度制御の実現と、利用者に対する資源割り当ての公平性確保を両立させる必要がある。

当計算機センタでは、これらの課題を解決するため、独自にジョブ制御方法を設計／開発・運用し、高い利用効率を実現している。

本稿ではジョブ制御の効果について報告し、その具体的な制御方法を紹介する。

2. 設計更新の経緯と効果

計算機の利用形態は VPP500/7 の運用経験から把握できている。しかし、運用初年度はプログラムの高速計算手法の多様化、演算性能・メモリ容量の飛躍的な向上により新しい計算を模索する期間でもあり、時間の経過とともに利用形態が変化すると推測される。

計算機センタとしては、この変化を迅速に捉え、約 3 ヶ月に 1 度のサイクルで設計を見直し、現在に至るまで高い利用効率を実現している(図 1，図 2.)。

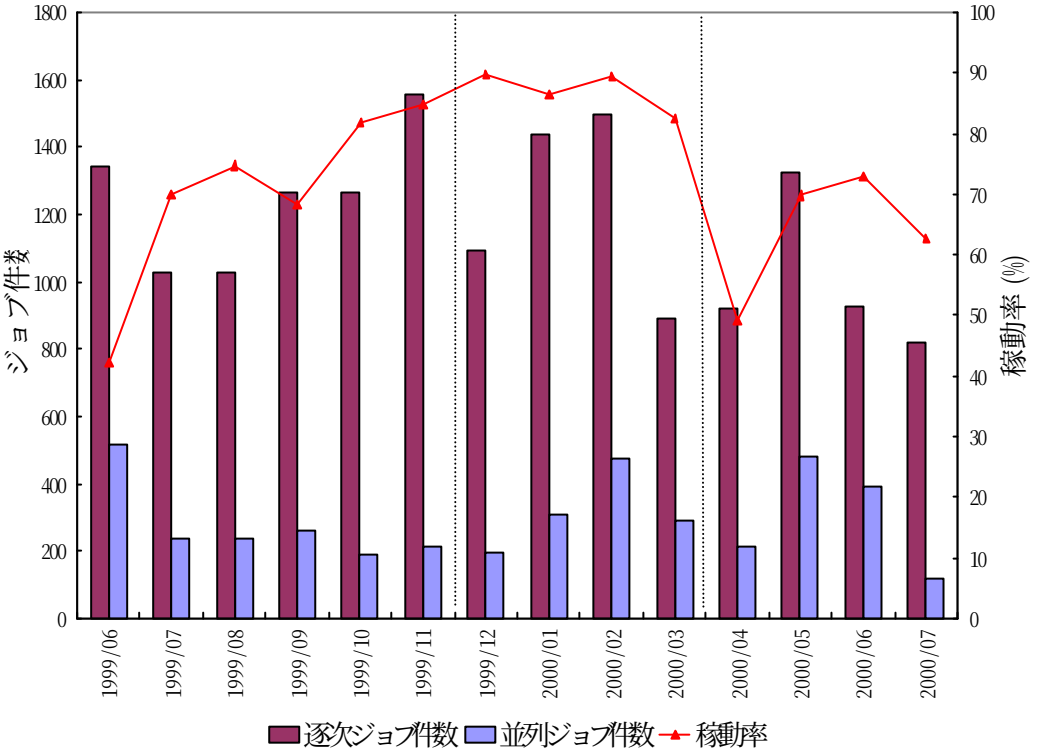

図 1. ジョブ件数と稼動率

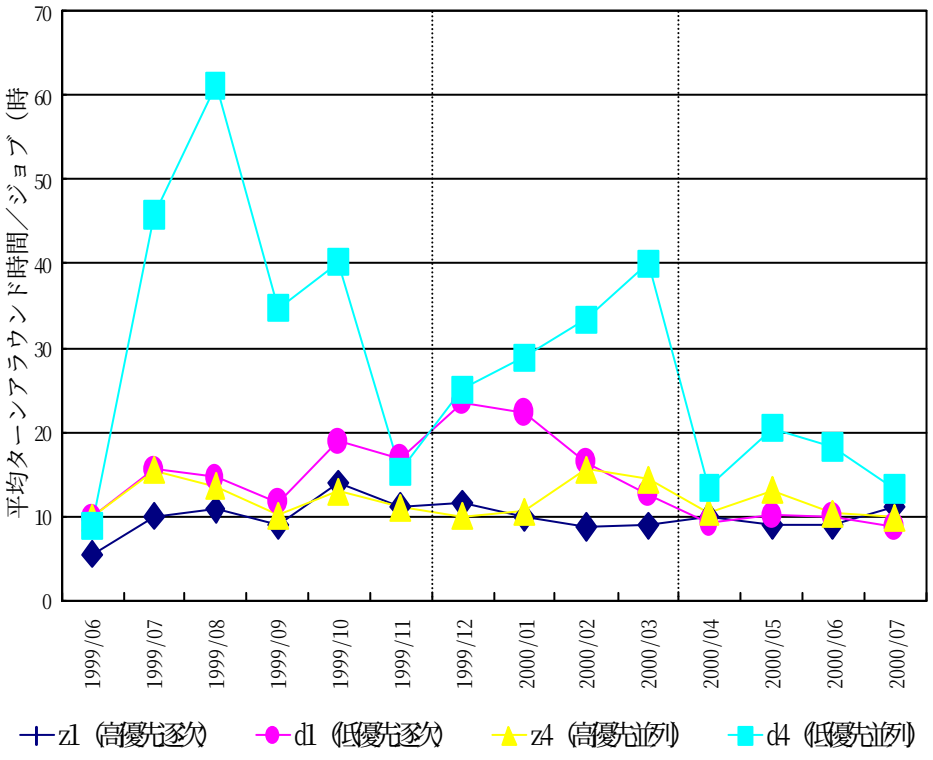

図 2.ジョブの平均ターンアラウンド時間

2.1. 導入設計（1999 年 3 月）

導入当初の設計では、「障害時 PE 割り当て変更機能」、「同一利用者連続実行禁止機能」、「シーケンシャル実行機能」を導入している。

VPP800/12 システムは 8GB メモリ搭載 PE×8、16GB メモリ搭載 PE×4 で構成されており、総メモリ容量は 128GB である。また、PE の単体性能も 5 倍に向上している。運用初期は VPP500/7(総メモリ容量 1.8GB)で実行できる規模の計算が多く実行されると推測されるため、メモリ容量には余裕がある。メモリを効率良く使用するため、各 PE では SHARE 実行としている。

VPP800/12 の PE 割り当てを表 1.に示す。 ■が各キューに割り当てられている PE であり、PE#1,2,4 は逐次ジョブ専用、PE#3,5,6,7 は並列ジョブ専用、PE#8,9,a,b は逐次ジョブと並列ジョブ兼用(16GB メモリ搭載)である。

表 1. VPP800/12 PE 割り当て

Queue	優先	PE	使用 PE											
			0	1	2	3	4	5	6	7	8	9	a	b
z1	高	1		■	■		■				■	■	■	■
d1	低	1		■	■		■				■	■	■	■
z4	高	4				■		■	■	■	■	■	■	■
d4	低	4				■		■	■	■	■	■	■	■

同一利用者によるジョブの連続実行を禁止する「同一利用者連続実行禁止機能」を導入し、特定利用者による計算機資源の独占を防止している。

また、複数のジョブをキューイングされた順番に実行する「シーケンシャル実行機能」を導入し、利用者の計算環境を支援している。本機能により実行されたジョブ数は、全体の約 10%と利用率は高く、使い勝手の良い計算環境を提供することができている(図 3.)。

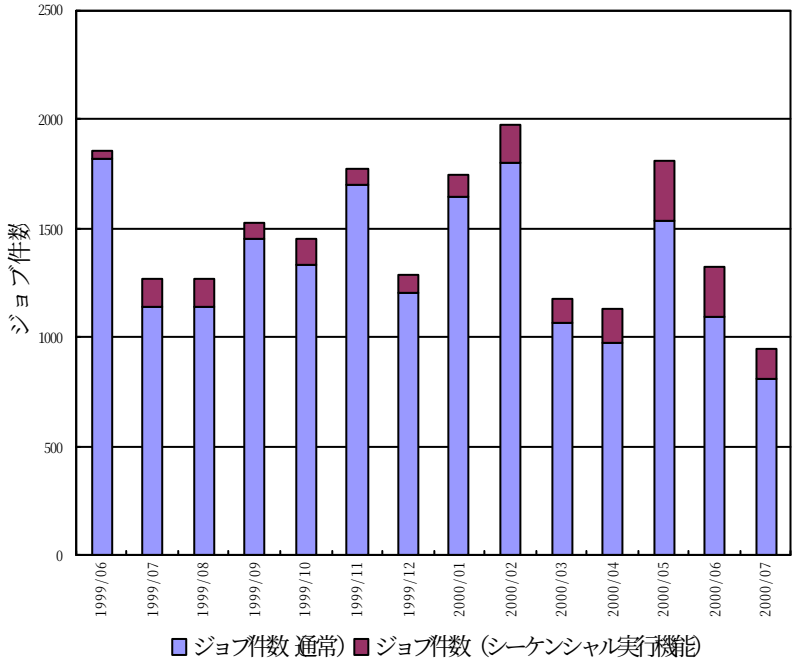

図 3.シーケンシャル実行ジョブ数

2.2. 第一回設計見直し（1999 年 11 月）

第一回設計見直しでは、「PE 括り付け緩和機能」、「残 CPU 時間制御機能」、「逐次パッキング機能」 を 1999 年 11 月末に導入している。

運用開始後、非常に高い利用効率を実現しているが、PE#8～PE#9 で並列ジョブと逐次ジョブが SHARE 実行されていることが原因により、優先順位の低い並列ジョブである d4 の実行待ち時間が長くなる傾向にある。並列ジョブが占有して実行できる PE 数を増やすことにより並列ジョブの待ち時間の短縮を実現している。

しかし、逐次ジョブに割り当てられた PE 数が少なくなるため、逐次ジョブを並列ジョブ用 PE で実行する「PE 括り付け緩和機能」、「残 CPU 時間制御機能」、「逐次パッキング機能」を導入し、並列ジョブ用 PE で実行できる逐次ジョブ数の増加を図っている。

設計見直し後の PE 割り当てを表 2.に示す。PE#0,4,a,b は逐次ジョブ専用であり 16GB メモリを搭載している。PE#1,2,3,5,6,7,8,9 は並列ジョブ専用である。ただし、 PE#1,2,3,5 は「PE 括り付け緩和機能」、「残 CPU 時間制御機能」、「逐次パッキング機能」により逐次ジョブでも使用される。

表 2. VPP800/12 PE 割り当て

Queue	優先	PE	使用 PE											
			0	1	2	3	4	5	6	7	8	9	a	b
z1	高	1	■	■	■	■	■	■					■	■
d1	低	1	■	■	■	■	■	■					■	■
z4	高	4		■	■	■		■	■	■	■	■		
d4	低	4		■	■	■		■	■	■	■	■		

図 4.に平均ターンアラウンド時間の推移を示す。問題となっていた並列ジョブ(d4)の実行待ち時間(1999/7～10)が短縮されている。導入後(2000/1～3)に並列ジョブ(d4)のターンアラウンド時間が長くなる傾向にあるが、これは優先順位の高い並列ジョブ(z4)の処理件数が多く(図 5.)、優先順位の低い並列ジョブ(d4)に CPU、メモリが配分されないことが原因である。この問題は後述する第二回設計見直しで改善している。

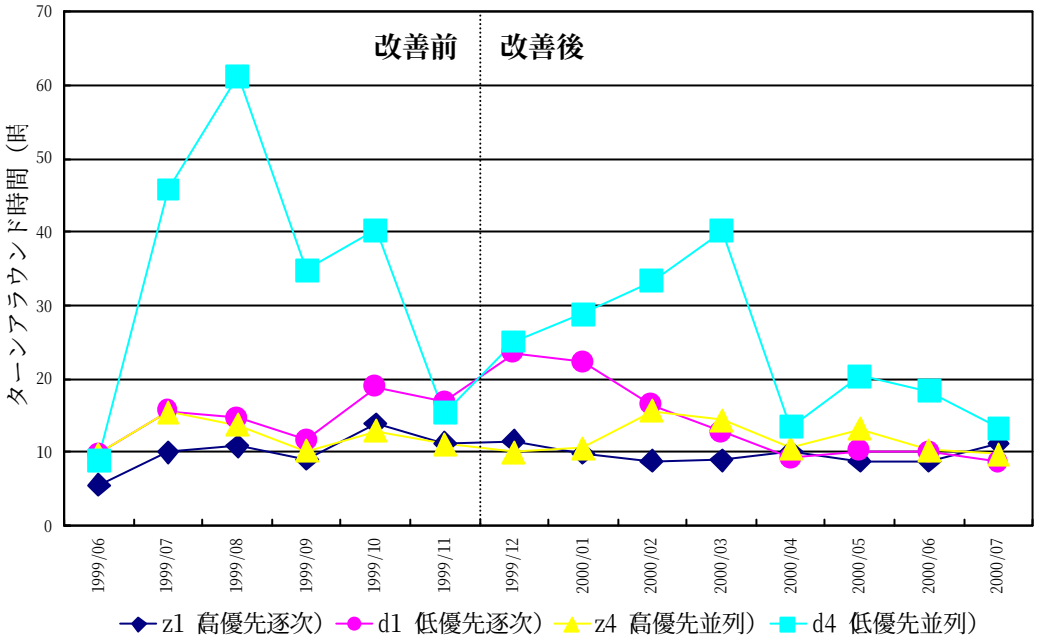

図 4. ジョブの平均ターンアラウンド時間

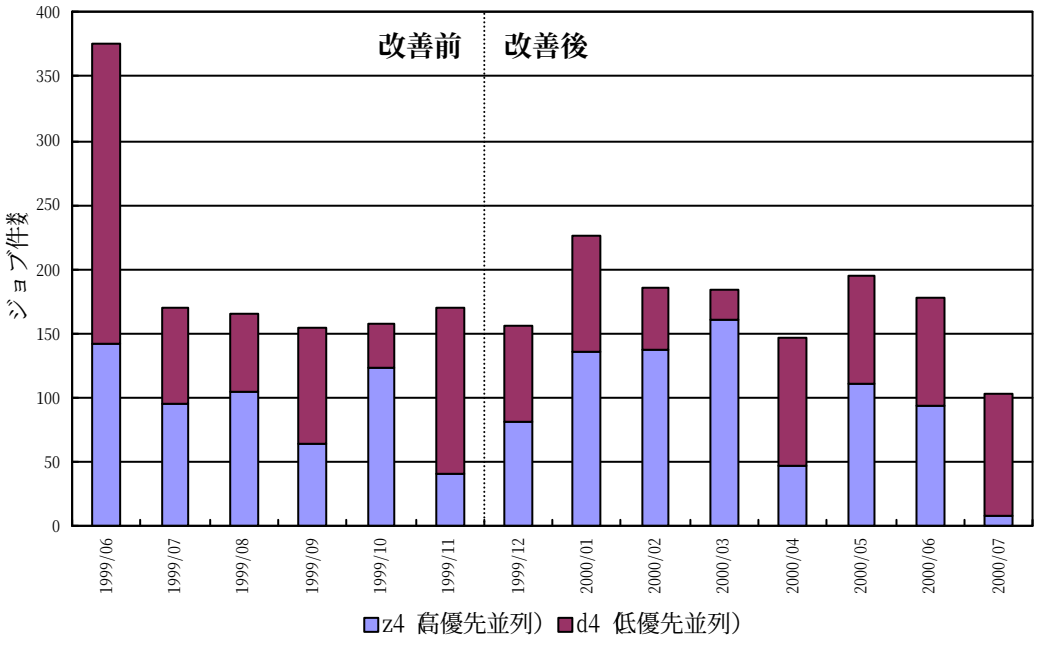

図 5. 並列ジョブ件数

「PE 括り付け緩和機能」により実行されたジョブ数は、全ジョブ数の約 10%である(図 6.)。また、「逐次パッキング機能」は逐次ジョブ件数が多い時(1999/12～2000/2)に機能している(図 7.)。1999/11 以降、逐次ジョブ用 PE が減っているが、「PE 括り付け緩和機能」、「残 CPU 時間制御機能」、「逐次パッキング機能」により、逐次ジョブへの PE 割り当て数を増やすことにより、変更前のターンアラウンド時間を維持することに成功している(図 4.)。

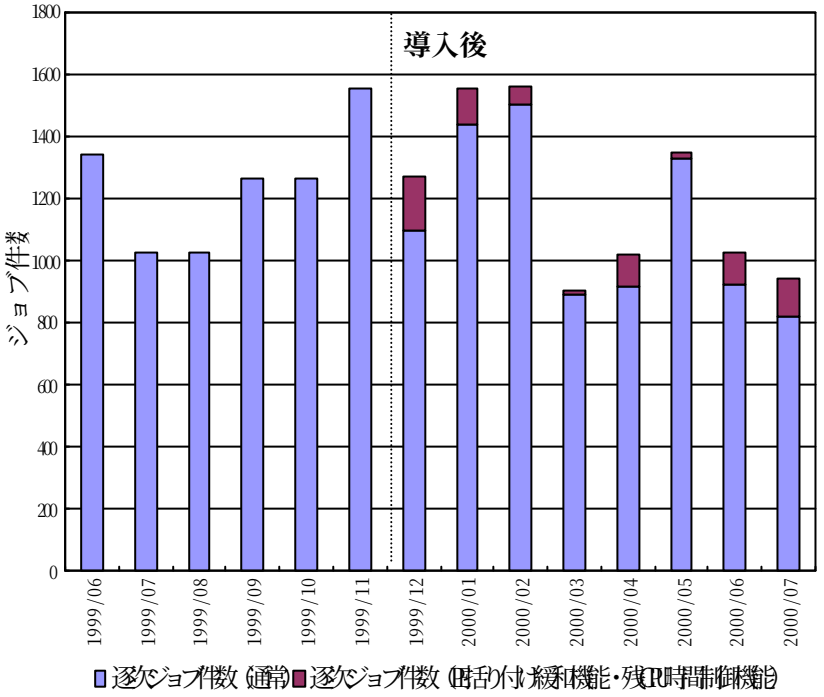

図 6.PE 括り付け緩和機能 実行ジョブ数

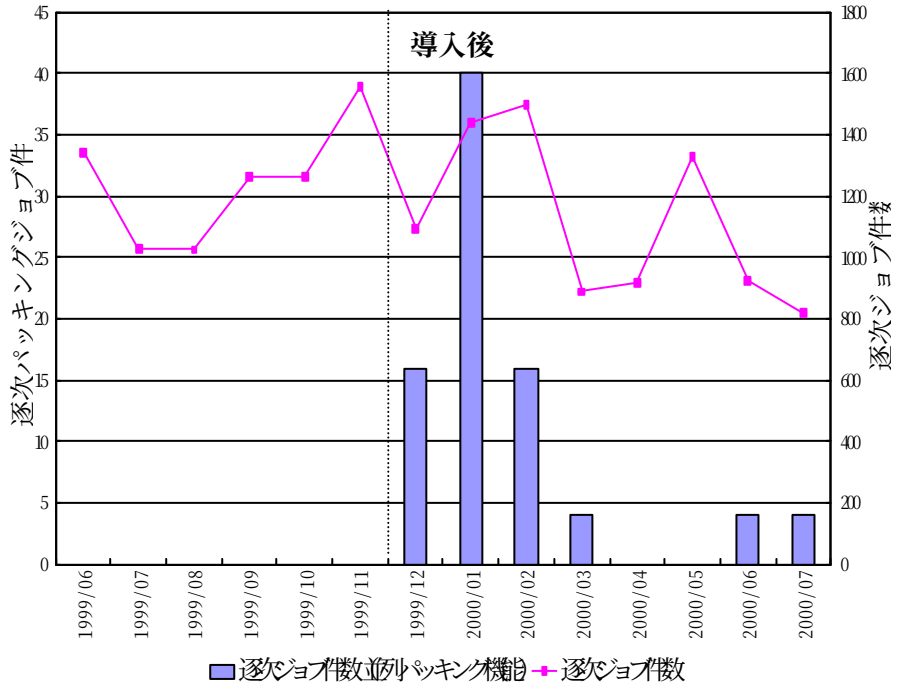

図 7.逐次パッキング機能 実行ジョブ数

2.3. 第二回設計見直し（2000 年 3 月）

第二回設計見直しでは「スワップアウト時間制限機能」、「SHARE 実行時間制限機能」を 2000 年 3 月末に導入している。

第一回の設計見直し後も高い利用効率を維持しており、並列ジョブの実行待ち時間も短縮されている。しかし、図 8.に示す通り、優先順位の低いジョブの経過時間が長くなる傾向にある(1999/12,1999/2～1999/3)。SHARE 実行された場合、優先順位の高い z1,z4 に対し、優先順位の低い d1,d4 は CPU が配分されず、CPU 使用時間の数倍の経過時間を必要としている。また、ジョブスワップも同様であり、優先順位の低い d1,d4 ではスワップアウト状態が数十時間に及ぶ場合がある。

優先順位の低いジョブの経過時間を短縮することを目的に、「SHARE 実行時間制限機能」及び「スワップアウト時間制限機能」を導入し、経過時間の短縮を実現している(図 8.)。

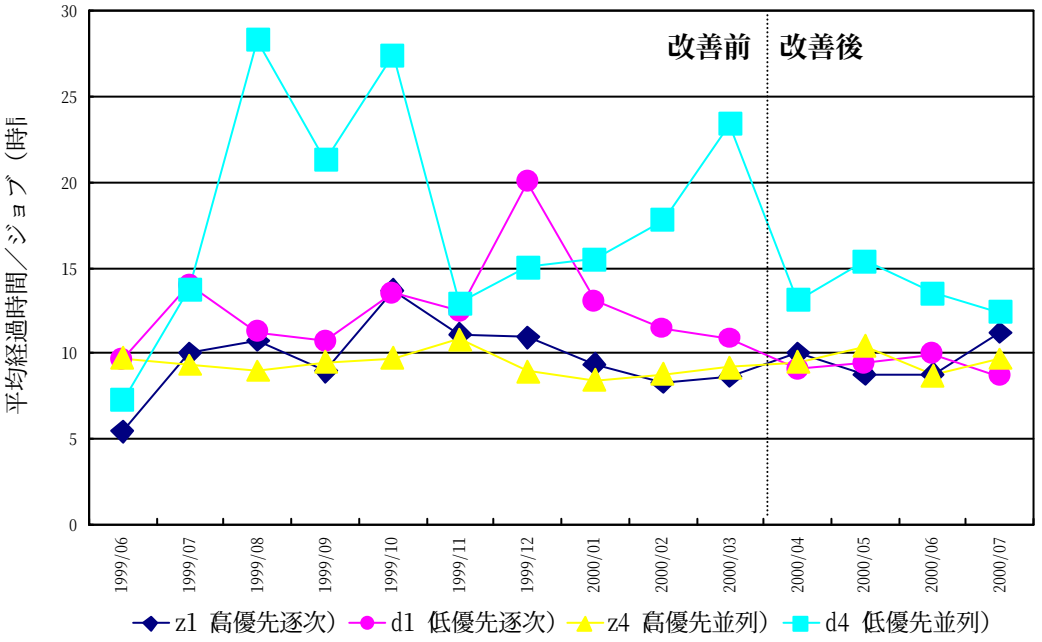

図 8. ジョブの平均経過時間

3. ジョブ制御機能紹介

計算機資源の有効活用、計算機利用における公平性確保、ターンアラウンド時間短縮を目的に、独自にジョブ制御機能を検討／開発し運用しており、その機能を紹介する。

ジョブの制御に当たっては、NQS(Network Queuing System) , PM(Partition Manager)の情報をジョブ制御プログラムの入力にし、資源の空き状況に応じてジョブを PM に渡すことにより実現している。

3.1. 計算機資源の有効利用

計算機資源の有効利用を目的に、「PE 括り付け緩和機能」、「残 CPU 時間制御機能」、「障害時 PE 割り当て変更機能」を導入している。

1) PE 括り付け緩和機能

VPP500/7 では逐次と並列ジョブが使用する PE(Processing Element)を完全に分けて運用してきたが、並列ジョブが少ない時期があり、PE に空きが発生する(図 9.)。VPP800/12 ではこの点を改善し利用効率の低下を抑止するとともに逐次ジョブのターンアラウンド時間を改善する必要がある。

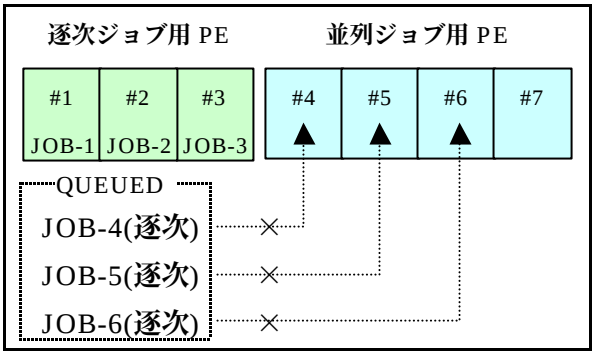

図 9.VPP500/7 PE 括り付け問題点

この問題を解決するため、PE 括り付け緩和機能を導入している。本機能は、実行待ちの並列ジョブが存在しない場合に限り、逐次ジョブを並列ジョブ用 PE で実行させる機能であり、並列ジョブが少ない時期に PE を有効に利用することができる。図 9.に対し本機能を使用した例を図 10.に示す。本機能を運用すると、実行待ちである 3 本のジョブ(JOB-4,5,6)は並列ジョブ用 PE(#4,5,6)に割り当てられ、PE の有効利用を実現している。

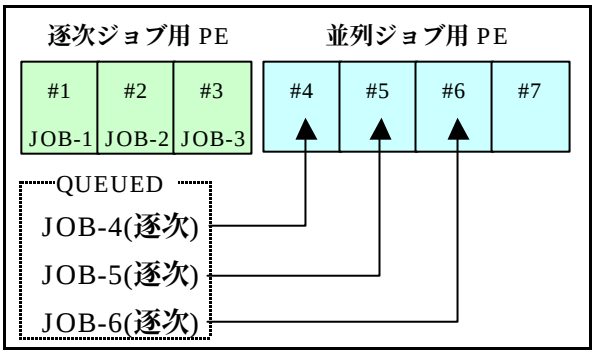

図 10.使用例 PE 括り付け緩和機能

ここで注意すべき点は、1PE(#7)が空いている状況で並列ジョブが投入された場合である。本来は並列ジョブ用 PE であるため、速やかに PE を明け渡す必要がある。実際の運用では、並列ジョブ用 PE で実行される逐次ジョブは、ジョブスワップの禁止、SIMPLEX 実行とすることで、明け渡す時間の短縮を図っている。逐次ジョブをスワップし、並列ジョブを優先的に実行することも検討したが、スワップの多発によるスループットの低下とディスク資源の枯渇を懸念し、本機能により実行された逐次ジョブに対しては採用していない。

また、図 10.の状況で並列ジョブが投入され、更に逐次ジョブが投入された場合の制御方法についても、後述する残 CPU 時間制御機能により、有効な PE 割り当てを実現している。

2) 残 CPU 時間制御機能

残 CPU 時間制御機能とは、並列ジョブが使用する PE が開放されるまでの時間内で終了する逐次ジョブを、空いている並列ジョブ用 PE で実行する機能である。

本機能について、図 11.を例に説明する。並列ジョブ(JOB-6)は、現在実行中の逐次ジョブ(JOB-4,5)が全て終了するまで実行することができない。並列ジョブ用 PE で実行されている逐次ジョブ(JOB-4,5)は、残り CPU 時間を計算すると、並列ジョブ用 PE(#4,5)を開放するまで 7 時間必要である。言い換えれば、7 時間は 2PE(#6,7)が空いている状態であり資源の無駄である。これを解決するため並列ジョブ用 PE が開放されるまでの 7 時間以内で終了する逐次ジョブ(JOB-8,10)を、空いている並列ジョブ用 PE(#6,7)に割り当てることにより、PE の有効利用を実現している。

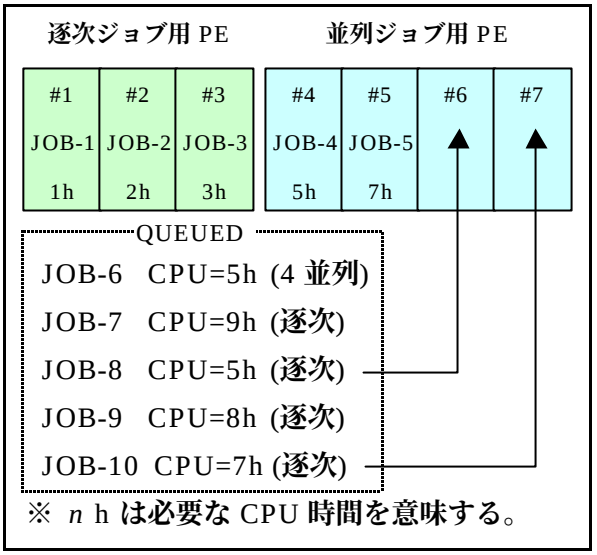

図 11.使用例 残 CPU 時間制御機能

3) 障害時 PE 割り当て変更機能

並列ジョブを実行するには、並列度と同数の PE を確保する必要がある。並列ジョブ用 PE で障害が発生すると、並列ジョブは必要な PE 数を確保することがでず、実行待ちとなる。図 12.では PE#4 で障害が発生した時の例を示している。PE#4 で障害が発生すると、並列ジョブは PE#4 が復旧するまで実行待ちとなる。更に、PE#5～7 の PE においては空き状態となる。

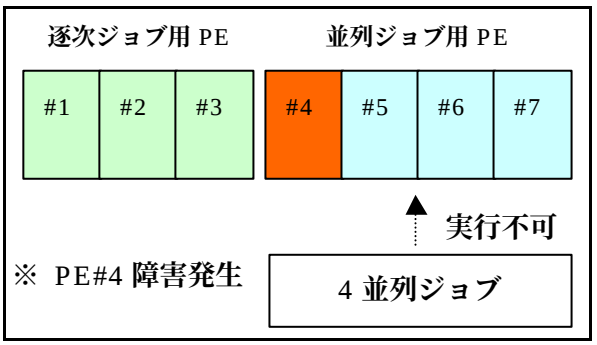

図 12.PE 障害発生時の問題点

障害発生時に空き PE をなくし、並列ジョブに PE を保証するため、障害時 PE 割り当て変更機能を導入している。図 12.の障害が発生した場合、逐次ジョブ用 PE#1～3 から 1PE を並列ジョブ用に振り替えることにより、並列ジョブに対して PE を保証している。

障害発生時、逐次ジョブ用 PE でジョブが実行されていない空き PE があれば、空き PE(#3)を並列ジョブ用に割り当てる (図 13)。並列ジョブ実行終了後も並列ジョブ用として使用される。

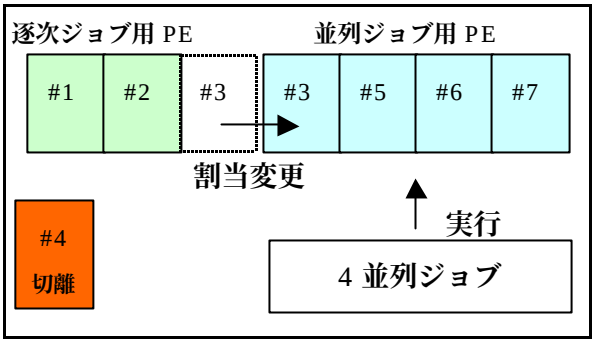

図 13.使用例 逐次ジョブ数 = 0 本

逐次ジョブ用 PE 全てでジョブが実行されていた場合は、残り CPU 時間より最初に開放される PE を求める。逐次ジョブ用 PE では SHARE 実行されるため、最初に開放される PE には逐次ジョブの割り当てを禁止し開放されるまでの時間短縮を図り、逐次ジョブ用 PE が 1PE でも開放された時点で、その PE を並列ジョブ用に割り当てる(図 14.)。

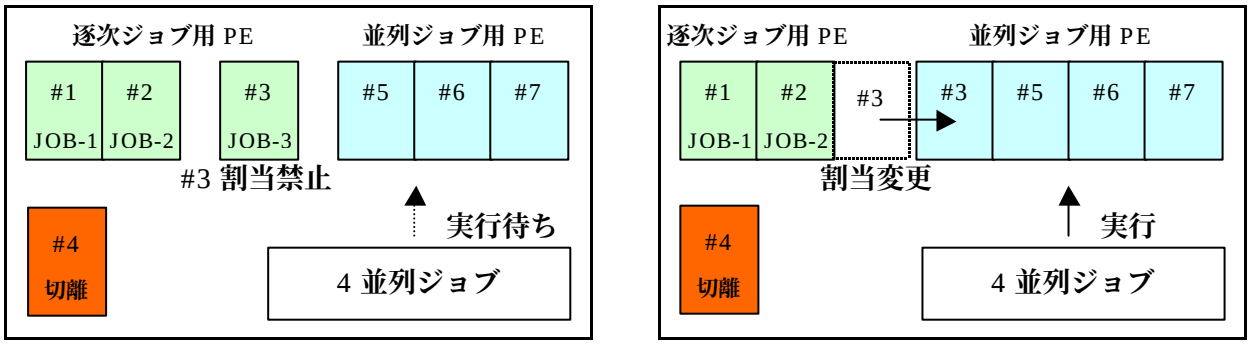

図 14.使用例 逐次ジョブ数 ≠ 0 本

3.2. 計算機利用における公平性確保

利用者に対し計算機資源を平等に割り当て、特定の利用者の資源占有を避ける必要がある。また、使いやすい計算機環境を提供するための機能を提供している。

1) 同一利用者連続実行禁止機能

同一利用者による資源の占有を防止するため、NQS のキュー内での連続実行を禁止している。ただし、他の利用者の実行待ちジョブが存在しない場合は、連続であっても実行される。

本機能は NQS-JM で提供される「キュー内連続実行制限機能」と同等であるため、本稿では詳細な説明を省略する。

2) シーケンシャル実行機能

利用者によっては複数のジョブを連続で順番に実行する場合があり、ジョブを実行した結果を元に次のジョブを実行する場合がこれにあたる。これを実現するには、複数のジョブをキューイングされた順番に実行する必要がある。連続実行を実現するためシーケンシャル実行機能を導入している。

シーケンシャル実行機能では、通常のジョブと見分けるため、NQS のバッチリクエスト名の先頭に "#" を付加している。"#"が付加されたジョブは、同一キュー内で投入された順番に実行される。

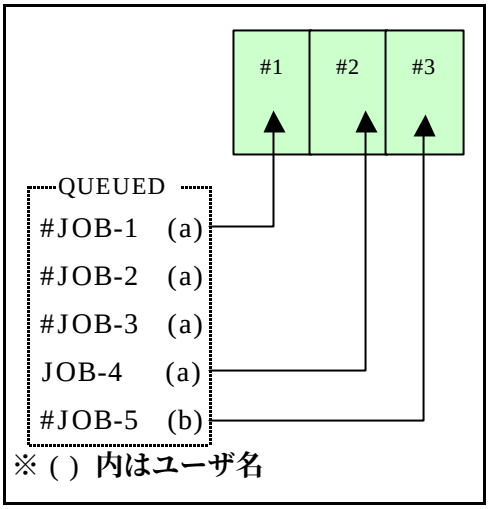

図 15.使用例 シーケンシャル実行機能

本機能を図 15.を例に説明する。最初に #JOB-1 が実行される。#JOB-2 , #JOB-3 は PE が空いていてもシーケンシャル実行が指示されているため、 #JOB-1 が終了するまで実行されない。JOB-4 はシーケンシャル実行が指示されていないため実行される。

3.3. ターンアラウンド時間改善

前述のジョブ制御機能により運用を開始し、非常に高い利用効率を実現している。しかし、優先順位の低いジョブにおいて、ターンアラウンド時間が長くなる傾向にある。ターンアラウンド時間を改善するため、「スワップアウト時間制限機能」、「SHARE 実行時間制限機能」、「逐次パッキング機能」を導入している。

1) スワップアウト時間制限機能

優先順位の高いジョブと低いジョブを設定しており、高優先ジョブに対して CPU 課金を実施している。高優先ジョブは低優先ジョブをスワップし実行することにより、課金の優位性を保っている。

しかし、高優先ジョブが多く実行されると、低優先ジョブはスワップされた状態が数十時間続くことがあり、ターンアラウンド時間が長くなる傾向にある。低優先ジョブのターンアラウンド時間を短縮するため、スワップアウト時間制限機能を導入している。

スワップアウト時間制限機能とは、ジョブスワップ対象のジョブがスワップアウト制限時間に達している場合、その時間を超えてスワップされることを防止する機能である。本機能により、低優先ジョブのターンアラウンド時間の短縮を図っている。

2) SHARE 実行時間制限機能

高優先ジョブと低優先ジョブ、または同じ優先順位のジョブ同士で SHARE 実行させることにより、メモリ資源の有効利用を図っている。また、高優先ジョブに CPU を独占させる CPU 配分率を設定することにより、高優先ジョブのターンアラウンド時間を保証し、課金の優位性を保っている。

しかし、低優先ジョブは数十時間にわたり、CPU が配分されない状態が発生し、ターンアラウンド時間が長くなる傾向にあるうえ、PE、メモリが長時間にわたって開放されない。この問題を解決するため、SHARE 実行時間制限機能を導入している。

本機能は、SHARE 実行される時間に制限を設け、その制限に達しているジョブとの SHARE 実行を抑止する機能であり、低優先ジョブのターンアラウンド時間の短縮を実現している。

3) 逐次パッキング機能

並列ジョブと比較すると逐次ジョブの割合が多く、逐次ジョブのターンアラウンド時間が長くなることが問題となる。

逐次パッキング機能とは、4 本の逐次ジョブをまとめて 4 並列ジョブとして見なし、並列ジョブ用 PE に割り当てる機能である。

図 16.では、逐次ジョブ(JOB-4,5,7,8)を並列ジョブと見なし、並列ジョブ用 PE(#4,5,6,7)に割り当てる例を示している。

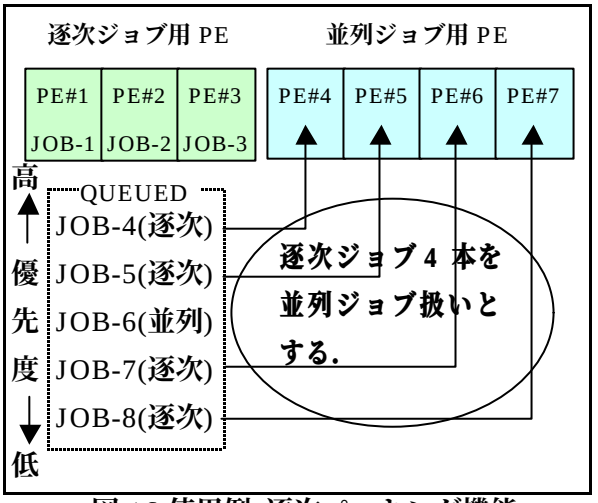

図 16.使用例 逐次パッキング機能

この機能で注意すべき点は、逐次ジョブと並列ジョブの優先順位の付け方である。図 16.を例にすると、並列ジョブ(JOB-6)よりも優先順位の低い逐次ジョブ(JOB-7,JOB-8)が先に実行されることになる。並列ジョブへの PE 割り当て遅延を防止するため、逐次パッキング機能により実行される逐次ジョブに対し次の条件を設けている。

- 逐次ジョブは同一キューのジョブとする。
- 逐次ジョブは、４本中に２利用者以上のジョブを含める。
- 並列ジョブを追い越す場合、並列ジョブより優先順位の高い逐次ジョブを 2 本以上含む必要がある。

4. 今後の課題

2 回の設計見直しにより、ジョブ制御は柔軟性を増し、使い勝手の良い計算機環境を利用者に提供することに成功している。

今後は、ジョブが使用しているメモリ容量、CPU 時間の統計情報を参考に、メモリ容量の少ないジョブ、CPU 時間の短いジョブに対する優先的な PE 割り当ての方法など、より柔軟なジョブ制御を検討したい。

添付資料

VPP800/12 システム構成図、現在の NQS 定義情報を添付する。

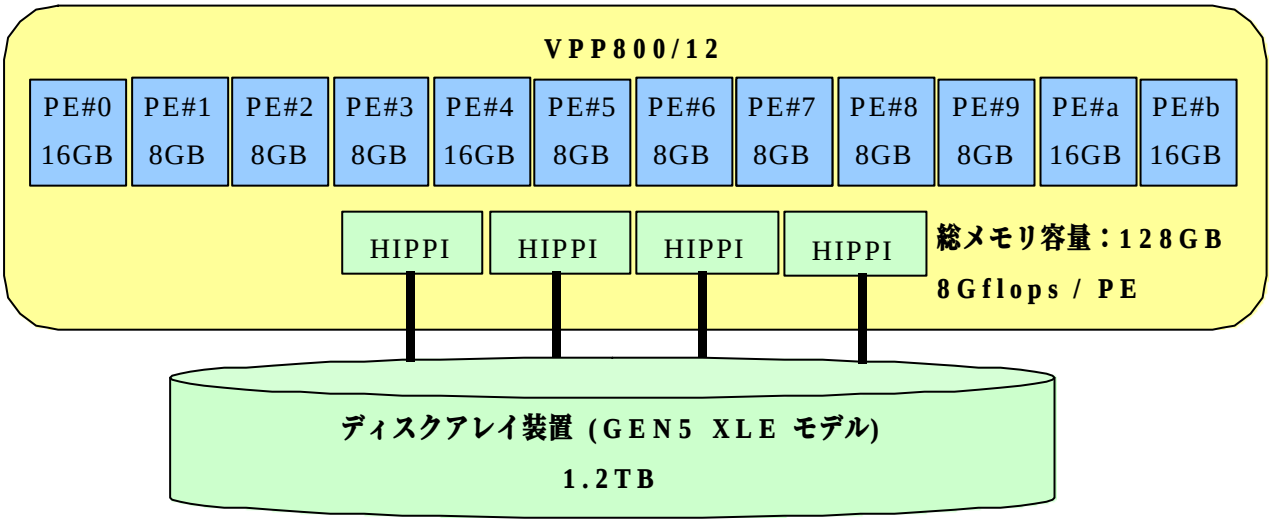

図 a. VPP800/12 システム構成図

表 a. NQS 定義情報

Queue	PE	CPU, Limit(min)	Memory Limit(MB)	CPU Ratio	使用 PE											
					0	1	2	3	4	5	6	7	8	9	a	b
v1	1	10	16000	100	①	⑤	⑥	⑦	②	⑧	⑨	⑩	⑪	⑫	③	④
s1	1	600	16000	100	①	⑤	⑥	⑦	②	⑧	⑨	⑩	⑪	⑫	③	④
z1	1	600	16000	10	①	⑤	⑥	⑦	②	⑧	⑨	⑩	⑪	⑫	③	④
d1	1	600	16000	1	①	⑤	⑥	⑦	②	⑧	⑨	⑩	⑪	⑫	③	④
v4	4	10	7808	100		①	①	①		①	②	②	②	②		
s4	4	600	7808	100		①	①	①		①	②	②	②	②		
z4	4	600	7808	100		①	①	①		①	②	②	②	②		
d4	4	600	7808	100		①	①	①		①	②	②	②	②		
z8	8	600	7808	100		①	①	①		①	①	①	①	①		
s8	8	600	7808	100		①	①	①		①	①	①	①	①		

※ ■ は割り当てられる PE であり、○内の数字は PE が割り当てられる順番を意味する。

参考文献

[1] 宇宙科学研究所 PLAIN センター(1999),「計算機利用の手引 操作編 スーパーコンピュータ(VPP800/12 システム)」
[2] 宇宙科学研究所 PLAIN センター(1996),「VPP500 利用の手引」
[3] UXP/V ネットワークキューイングシステム-JS V20 用
[4] UXP/V ネットワークキューイングシステム-JM V20 用