

# *ES/1 NEO*

## *MFシリーズ*

MF-WebSphere  
パフォーマンス・チューニング作業

第3版 2024年 2月

©版權所有者 株式会社 アイ・アイ・エム 2024年

© COPYRIGHT IIM CORPORATION, 2024

ALL RIGHT RESERVED. NO PART OF THIS PUBLICATION MAY  
REPRODUCED OR TRANSMITTED IN ANY FORM BY ANY MEANS,  
ELECTRONIC OR MECHANICAL, INCLUDING PHOTOCOPY RECORDING,  
OR ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM WITHOUT  
PERMISSION IN WRITING FROM THE PUBLISHER.

“RESTRICTED MATERIAL OF IIM “LICENSED MATERIALS – PROPERTY OF IIM

# 目次

---

第 1 章 パフォーマンス・チューニング作業 .....	1
1.1 本章の使用方法.....	1
OSP02n .....	2
OSP03n .....	3
OSP04n .....	5
PROC01n .....	6
PROC02n .....	8
PROC03n .....	9
STOR01n & STOR02n .....	10
STOR05n .....	11
WKLD02n .....	12

# 第1章 パフォーマンス・チューニング作業

## 1.1 本章の使用方法

CPEWAS00プロセッサでは各種のチューニング・ヒントを出力します。しかし、CPEWAS00プロセッサが出力するメッセージだけでは、その全てを説明することは困難です。この章では、CPEWAS00プロセッサが出力するチューニング・ヒントに対応したチューニング作業について解説します。

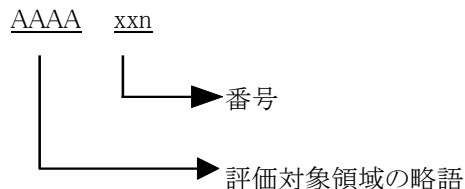
尚、この章は“MF-WebSphere 使用者の手引き 第1章 CPEWAS00の使用方法”で説明されている事項を理解していることが前提で記述されています。また、チューニング作業の具体的な実施方法で疑問などがあれば、弊社担当者へお問い合わせ下さい。各種の方法で、チューニング作業の支援を行います。

CPEWAS00プロセッサでは、評価対象のオペレーティング・システムを下記の領域に分割し、それぞれの評価結果をチューニング・ヒントで表示します。

■オペレーティング・システム・パラメータ	.....OSP
■プロセッサ	.....PROC
■主記憶、拡張記憶	.....STOR
■業務プログラム	.....WKLD

それぞれの領域には、その領域を示す略号が決められています。CPEWAS00プロセッサがチューニング・ヒントを出力する際には、参照コードと重要度が付加されます。参照コードは、評価対象領域の略号と3桁の番号により構成されています。この参照コードは、本章の各ページの上段に示されたページ識別名に対応付けられています。もし、同一の領域で複数のチューニング・ヒントが出力された際には、重要度の番号が小さい(重要な)ものから調査されることをお勧めいたします。

<参照コード/ページ識別名の形式>

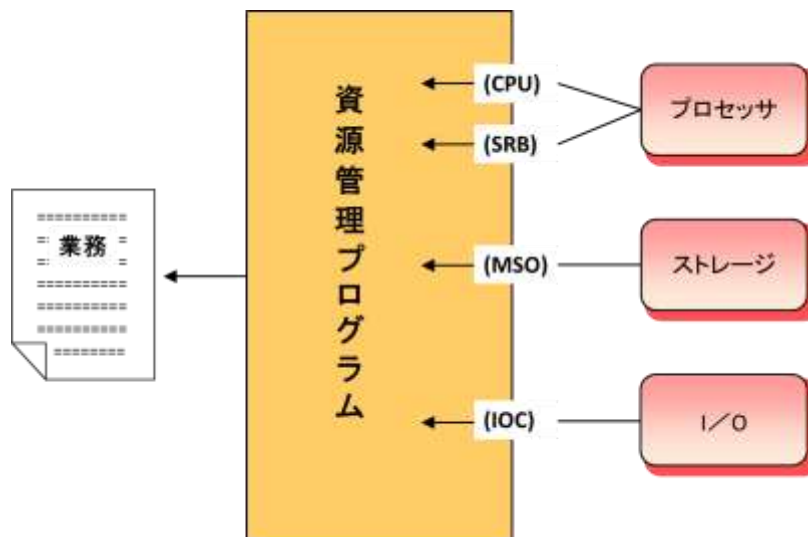


## OSP02n

## 【説明】

使用中のオペレーティング・システムにおける資源管理プログラムのIPSパラメータで、トランザクションのサービス量を算出する際に使用するリソース係数の値にゼロが設定されています。ゼロに設定されているリソース係数を1に変更されることをお勧めいたします。

## 【解説】



オペレーティング・システムの資源管理プログラムは、処理されるオンライン・トランザクションやバッチ業務のためにシステム資源をどの程度使用したかを示すためにサービス量と呼ばれる値を計算しています。このサービス量は、プロセッサやストレージおよび入出力サブシステムの各システム資源の使用率やアクセス回数などに基づいて算出します。資源管理プログラムは、トランザクションやジョブ毎に求めたサービス量により、それらの処理優先順位を決定します。この際、重要な資源(能力が相対的に小さいと考えられる資源)を多く使用するトランザクションなどの処理優先度を低くします。しかし、プロセッサ能力やストレージ容量および入出力サブシステムへの最大アクセス回数などの比率は、全てのシステムで同じではありません。このため、資源管理プログラムではリソース係数をシステム資源毎に設定するようにしております。リソース係数には、CPUとSRB、MSOおよびIOCの4種類があります。これらの内、CPUとSRBはプロセッサ使用に関するリソース係数です。MSOとIOCはそれぞれがストレージと入出力サブシステムに対応するリソース係数です。トランザクションやジョブ毎のサービス量の算出とリソース係数の関係は次のようになっています。

$$\begin{aligned} \text{サービス量} &= \text{プロセッサ使用時間} \times \text{プロセッサ・リソース係数 (CPUとSRB)} \\ &+ \text{使用ストレージ・フレーム量} \times \text{ストレージ・リソース係数 (MSO)} \\ &+ \text{入出力要求回数 (又は時間)} \times \text{入出力リソース係数 (IOC)} \end{aligned}$$

前式から判りますように、システム内で重要と考えられるシステム資源のリソース係数の値は、相対的に大きくする必要があります。下記に省略値を示します。

CPU	SRB	MSO	IOC
10	10	3	5

非常に稀ではありますが、このリソース係数をゼロにしておられるユーザがあります。その理由としては、対応するシステム資源の使用率を無視したい場合が多いようです。しかし、パフォーマンス管理の観点から、各パフォーマンス・グループ毎のシステム資源の使用状況を把握することが必須であると言えます。このため、リソース係数はゼロに設定しないようにして下さい。ゼロの代わりに“1”を設定して下さい。“1”を設定されても他のリソース係数が大きければ、そのシステム資源の使用率を無視したのと同じ事となります。

## 【対応策】

- ゼロに設定されているリソース係数の値を“1”に変更する。

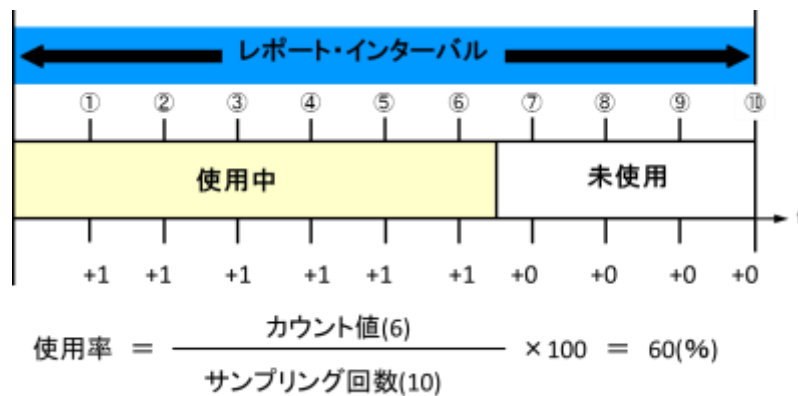
## OSP03n

## 【説明】

使用中のソフトウェア・モニタのインターバルが15分から30分以上に設定されています。評価精度を向上させるためには、インターバルを15分から30分に設定して下さい。

## 【解説】

ソフトウェア・モニタのインターバルは、感覚的に10分の方が15分よりもデータ量が多く、評価精度も向上すると思われます。しかし、RMF等のソフトウェア・モニタのインターバルは、それらのモニタが出力するデータ群の信頼性を決定するため、あまり短くされないようにして下さい。つまり、インターバルを短くすると、データの信頼度が低下し、評価精度を保証できなくなります。例えば、入出力装置の使用率などはサンプリング技法で計測されます。このサンプリング技法では、一定時間間隔(サイクル)毎に割り込みを発生させ、そのタイミング毎に装置の状態を記録します。そして、1回のインターバルの終了時に装置が使用中であった回数を全サンプリング回数で割り算することにより、使用率を算出しています。このため、10回のサンプリングで使用率を求めると、その結果は0, 10, 20, …となり、信用できるのは上位1桁のみとなります。数学的には、百分率をもとめるには1000回以上のサンプリングが必要であることが判ります。一般的には、サンプリングの時間間隔としては1秒が使用されています。この場合にインターバルを10分としますとサンプリング回数は600回となり、使用率等の小数部は信用できなくなります。



データの信頼性を向上させるためには、サンプリング回数を増やすため、サンプリング間隔を短くするか、インターバルを長くする方策が考えられます。サンプリング間隔を短くするとモニタのオーバーヘッドが増加します。このため、弊社ではインターバルを15分(サンプリング回数が900)にされることをお勧めいたします。また、30分以上のインターバルでは、全てが平均化されてしまうため、評価精度の向上は望むことができません。

## 【対応策】

- ソフトウェア・モニタのインターバルを15分から30分以内に設定することをお勧め致します。

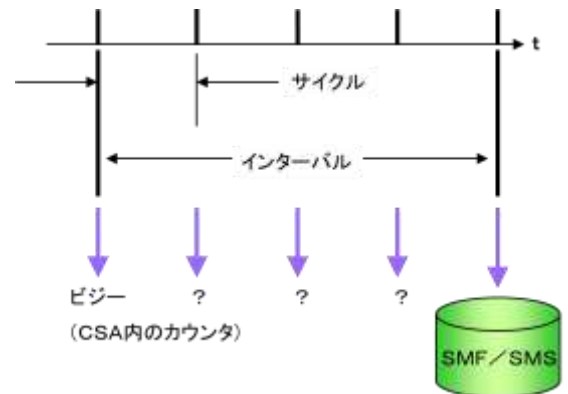
【注意点】

- RMFの注意点RMFの場合、メーカ提供のパラメータで動作させる事により問題が発生することは少ないようですが、データの信頼性を考察し、CYCLEとINTERVALの関係は次のようにする必要があります。

$$\text{INTERVAL} \div \text{CYCLE} \geq 2000$$

∴500msのサイクル指定の場合、インターバルは15分以下にできません。

上式中のCYCLEは、タイマ・ドリブン技法のデータ収集プログラムが起動される時間間隔です。INTERVALとは、サンプラが更新するサンプリング値(CSA等に記憶)を基に使用率などを算出し、SMF/SMSファイルなどに書き出す時間間隔です。



## OSP04n

## 【説明】

入力レコードを読み込んでいる際、おかしい順番のレコードを検出しました。プロセッサは強制的にレコード読み込み処理を終了します。入力レコードを日付と時刻で、正しくソートして下さい。でなければ、正しく評価対象の時間帯を選択することができません。

## 【解説】

ES/1 NEOでは、入力されたパフォーマンスデータを利用して、各種の解析機能を提供します。その際、システム内の単一リソースの使用状況を判定するのではなく、プロセッサやストレージ(主記憶)および入出力装置などの稼働状況を総合的に判定しています。しかし、大半のソフトウェアモニタ(RMFなど)は、一つ一つのリソースの稼働状況を報告するために、数多くの種類のレコードを出力します。これらのレコードを一对の時間帯のものであると判断するには、ソフトウェアモニタが出力した順に処理する必要があります。

ES/1 NEOでは、入力されたパフォーマンスデータが、ソフトウェアモニタが出力した順に並んでいるか否かを判定するために、レコード記録時間を検査しています。このチューニングヒントは、このパフォーマンスデータの並びに異常を検出した際に出力されます。もし、このチューニングヒントが出力された場合には、パフォーマンスレコードの作成日付と時刻、レコード番号などでソートし直す必要があります。場合によっては、ソートするだけで問題が解決できない場合もあります。そのような場合、パフォーマンスデータ収集の形態を変更することが必要となります。是非、当社の担当者までご連絡下さい。



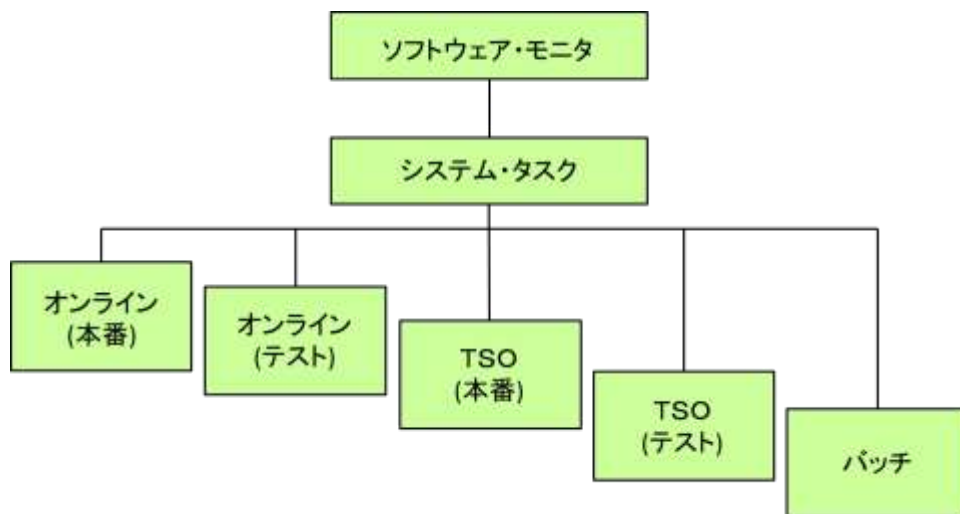
## PR0C01n

## 【説明】

プロセッサの使用率が高いため、プロセッサ使用のための待ち時間が長くなっています。重要な業務プログラムのディスパッチング・プライオリティに注意して下さい。

## 【解説】

プロセッサは、複数の業務プログラムで共用使用されているリソースです。このようなリソースにおいて使用率が高くなると、優先順位の低い業務プログラムがそのリソースをアクセスする際に待たされる時間(キュー時間)が長くなります。このため、プロセッサの使用率が高いシステムにおいては、プロセッサ・アクセスの優先順位であるディスパッチング・プライオリティが適正に設定されていることを確認しなければなりません。

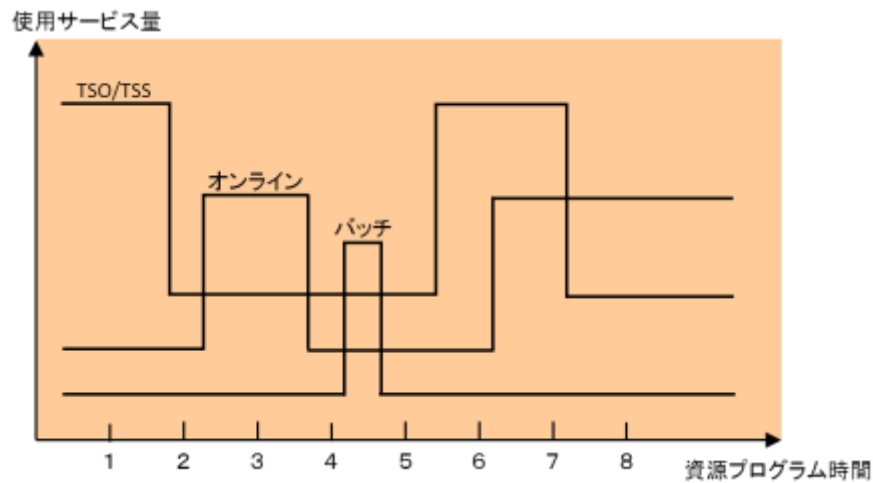


ディスパッチング・プライオリティは、運用中の業務プログラム・グループのバランスを考察して設定します。ここでは、一般的なガイド・ラインを示します。

- ① ディスパッチング・プライオリティには、固定と自動制御(MEANTIMETOWAIT)があります。固定はシステム・タスク用に用意されているものと理解して下さい。一般的な業務プログラムのディスパッチング・プライオリティは自動制御で指定して下さい。
- ② ソフトウェア・モニタ(RMF等)には、システムで最高のディスパッチング・プライオリティを割り当てて下さい。
- ③ JESやネットワーク制御プログラム(VTAM等)等のシステム・タスクには、ソフトウェア・モニタの次のディスパッチング・プライオリティを割り当てて下さい。
- ④ TSOおよびバッチは全て自動制御で指定して下さい。通常システムでは、オンライン>TSO>バッチの順番になります。

運用中のシステムにおいて、最高優先順位の業務がプロセッサを100%使用する場合は、他の業務でのプロセッサ使用を可能にするために、タイム・スライシングを指定して下さい。

<次頁につづく>



## 【対応策】

- ディスパッチング・プライオリティの適正化を図る。
- 最大プログラム多重度を制限する。
- タイム・スライシングを指定する。
- プロセッサを増強する。
- TSA (トータル・システム・アナライザ) や PPE (プロブレム・プログラム・エバリュエータ) 等で高負荷モジュールをチューニングする。
- RMFのモニタⅡでプロセッサ使用の待ち時間を調べる。また、その原因となるプロセッサ使用の多いプログラムなども調査し、チューニング対象とする。



RMFはIBMのソフトウェア・モニタです。また、TSA やPPEはブール&バページ社の 商品名で、IBMシステムでのみ稼働します。

## PR0C02n

## 【説明】

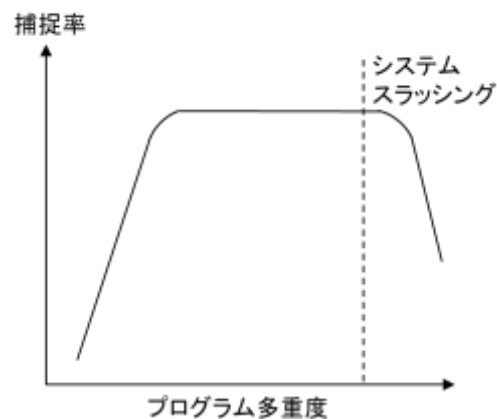
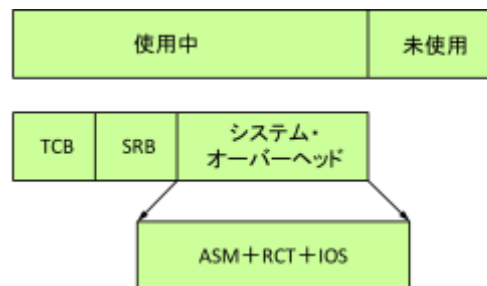
プロセッサ使用率の内に占めるシステム・オーバーヘッドが高すぎます。ストレージの競合（ページング）が原因と考えられますが、その詳細を調査して下さい。

## 【解説】

オペレーティング・システムにおけるプロセッサの使用率の内訳は右図のようになっています。ソフトウェア・モニタがプロセッサ使用率として求めているのが、プロセッサが使用中であった時間の割合です。また、各パフォーマンス・グループで実行される業務プログラムごとに計算されているCPUやSRBのサービス・ユニット値により、プロセッサがTCBやSRBモードで使用されていた時間も求めることができます。この2つの時間の差がシステム・オーバーヘッドです。

このシステム・オーバーヘッドには、ASM（ページング）やRCT（スワップ）およびIOS（入出力 動作）モジュールの動作時間が含まれています。ページングやスワップおよび入出力動作の回数は、プログラム多重度に比例して増加します。一方、TCBやSRBモードでのプロセッサの使用時間もプログラム多重度に比例して増加します。このため、非常にプロセッサ使用率が低い場合を除き、システム・スラッシングが発生するまでプロセッサ捕捉率は安定しています。

一般的に、システム・スラッシングの最大の原因は、ページングであると言われております。このため、システム・オーバーヘッド量が問題となる際には、ストレージ競合について考察する必要があります。ストレージ競合に関しては、STOR01nのチューニング・ヒントなどの項を参照して下さい。



## 【対応策】

- ストレージ・バウンドのプログラムをチューニングする。
- 最大プログラム多重度を制限する。
- 実ストレージを増強する。

**PR0C03n****【説明】**

この共有区画が使用したプロセッサ能力(プロセッサ使用率)が、共有区画の重み値で決定されるプロセッサ能力の配分量を越えています。

**【解説】**

論理分割を使用しているユーザは、複数の区画を使用できます。これらの区画では、プロセッサをどのような形態で使用するかを指定しなければなりません。プロセッサを一つの区画で専有する場合、その区画の事を専有区画と呼びます。この専有区画では、使用できるプロセッサの数だけのプロセッサ能力を全て使用できます。

一方、複数の区画でプロセッサを共有する場合、これらの区画の事を共有区画と呼びます。これらの共有区画では、各々が重み値を持っており、その重み値に対応した割合だけプロセッサ能力を使用するように制御されています。しかし、CP使用制限が指定されていない区画の場合、この重み値で決定されたプロセッサ能力以上にプロセッサ能力を使用することが可能です。このような状況は、正常でも異常でもあると言えます。

**STOR01n & STOR02n****【説明】**

主記憶の競合が高まっております。主記憶の使用状況を評価すると同時に、必要であれば主記憶の増強を検討して下さい。

**【解説】**

主記憶は複数の業務プログラムで共用使用されるリソースです。このため、プログラム多重度が増加するにつれ、主記憶の競合が激化し個々の業務プログラムが使用可能な主記憶フレーム数が業務プログラム毎の特性である最小ワーキング・セット・サイズ(WSS)以下になると、ページングの影響が顕著に現れ応答時間が急激に悪化します。オペレーティング・システムで運用される業務プログラムの多くが前述のような状態になりますと、主記憶内のフレームが再使用されるまでには、必ず一旦、実記憶から追い出されることとなります。このため、主記憶のフレームを管理するUIC(非参照時間)の値は急激に減少します。このような理由によりUICの値が低い値になる際には、重要な業務プログラムに十分なワーキング・セット・サイズが保証されるように配慮し、応答時間の安定化を図らなければなりません。この目的のために、資源管理プログラムのストレージ分離機能を使用して下さい。ストレージ分離機能を使用する際に、オンライン・サブシステムのように最小ワーキング・セット・サイズの判定が困難なものには、ページング・レートを指定して下さい。また、TSOやバッチには、最小ワーキング・セット・サイズを指定して下さい。但し、いずれも重要な業務プログラムにのみ指定されることをお勧めいたします。

**【対応策】**

- IPSパラメータでストレージ分離機能を使用してワーキング・セット・サイズをチューニングする。
- 最大プログラム多重度を制限する。
- 主記憶を増強する。

## STOR05n

## 【説明】

主記憶でページ固定されている割合が高過ぎます。各領域でのページ固定の割合を調査し、必要であれば主記憶を増設して下さい。

## 【解説】

オペレーティング・システムは、仮想記憶を使用目的に応じて、業務プログラムが使用する私有域やオペレーティング・システムが使用する共通域などに分割します。これらの私有域や共通域などに分割されたページ群の内、頻繁に使用されるものが主記憶内に存在できます。また、パフォーマンス上実記憶に常駐することが必要であるページ群はページ固定され、ページング対象外となっています。このように主記憶内に存在するページは、ページ固定されたページ、もしくはページング可能なページの中で頻繁に使用されるページです。これらのページが記憶されたフレームのことを割当て済フレームと呼びます。一方、主記憶のフレームの中には、どのページにも専有されていないものがあります。この状態にあるフレームのことを未使用フレームと呼び、AFQ(アベイラブル・フレーム・キュー)と呼ばれる待ち行列を構成しています。この未使用フレームは、ページ不在割り込みが発生した時に使用されます。資源管理プログラムは、ページ・イン処理時間を短縮するために、この未使用フレーム数を常時監視し、その数が一定基準値以下にならないようにしています。

また、資源管理プログラムは、主記憶でページ固定されている割合も常時監視し、その割合が一定基準値以上になると、“ページング可能域不足”と判断し、アドレス空間のスワップ・アウト、プログラム多重度の減少や新たなアドレス空間の生成を禁止します。この場合、ページングの増加も考えられますので、ページ固定の詳細を調査し、対処して下さい。

## 【対応策】

- ページ固定の割合を減少させる。
- 主記憶を増設する。

**WKLD02n****【説明】**

業務グループごとに設定された性能目標が達成されていません。重要な業務プログラムが影響を受けていないことを確認して下さい。問題のあった業務グループと、目標達成率を示します。

**【解説】**

MVS/ESAバージョン5のゴールモードでは、システムで運用される業務群にサービス目標を設定することができます。WLM(ワークロードマネージャ)は設定されたサービス目標が達成できるように、業務間の優先順位を動的に制御します。

ES/1 NEOでは入力されたパフォーマンスデータの中から、システム管理者が設定したサービス目標と、実測されたサービス状況に関する情報を抽出しています。そして、指定されたサービス目標が達成されていないサービスグループを検出すると、そのサービスグループの重要度などに応じた警告を発するようになっております。このチューニングヒントが出力される場合には、サービス目標を達成していないサービスクラスを見つけたことを意味します。同時に、それらのサービスクラスに関する情報も表示されていますので、それらのサービスクラスで重要な業務が動作していないことを確認して下さい。

ゴールモードでは、サービスクラスごとにサービス目標を設定しますが、達成することが困難な目標を設定すると、システム運用に各種の問題を引き起こすことがあります。WLMは、サービス目標が達成されていない業務に多くのリソース使用権を与えます。例えば、あまり稼働していないサービスクラスであってもサービス目標が達成されていなければ、WLMはそのサービスクラスに大量の主記憶域を専有させることがあります。このため、他のサービスクラスのプログラムでページングが頻発し、システム全体の運用効率が低下することもあります。このような事態をさけるために、サービス目標を達成していない業務はどれか、またその業務は重要か否かを常時監視しておく必要があります。