

# *ES/1 NEO*

## *MFシリーズ*

MF-XSP

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
OSP03n (1/2)	2
OSP03n (2/2)	3
PROC01n (1/2)	4
PROC01n (2/2)	5
PROC02n	6
STOR01n & STOR02n	7
STOR05n	8
IOSS01n (1/2)	9
IOSS01n (2/2)	10
IOSS02n	11
IOSS03n (1/2)	12
IOSS03n (2/2)	13
IOSS04n	14
IOSS05n	15
IOSS06n (1/2)	16
IOSS06n (2/2)	17
IOSS07n	18
IOSS08n	19
IOSS10n	20
IOSS11n	21
IOSS12n & IOSS13n	22
PAGE01n & PAGE02n	23
PAGE03n	24
PAGE04n & PAGE05n	25
PAGE06n	26
PAGE07n	27
SWAP10n	28
VS03n & VS04n	29
VTAM01n	30
VTAM02n	31
WKLD01n	32
第 2 章 パフォーマンス・チューニング作業 (AIM)	33
B0F01x	34
B0F02x	35
DB01x	36
DC01x	37
HLF01x	38
HLF02x	39
LOG01x	40

LOG02x .....	41
RESP01x .....	42
TLF01x .....	43
TLF02x .....	44

# 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

ES/1 NEO MF-XSPでは各種のチューニング・ヒントを出力します。しかし、ES/1 NEO MF-XSP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、ES/1 NEO MF-XSP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“CPEPRT00の使用方法”と“評価手法の紹介”の章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。

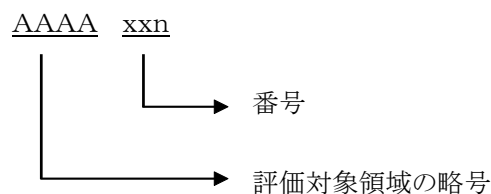
MF-XSPでは、評価対象のオペレーティング・システムを下記の領域に分割し、それぞれの評価結果をチューニング・ヒントで表示します。

■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、システム記憶	.....STOR
■入出力サブシステム	.....IOSS
■ページ・データセット	.....PAGE
■スワップ・データセット	.....SWAP
■仮想記憶	.....VS
■ネットワーク・サブシステム	.....VTAM
■業務プログラム	.....WKLD
■その他	.....NOTE

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。

ES/1 NEO MF-XSP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>



## OSP03n (1/2)

## 【説明】

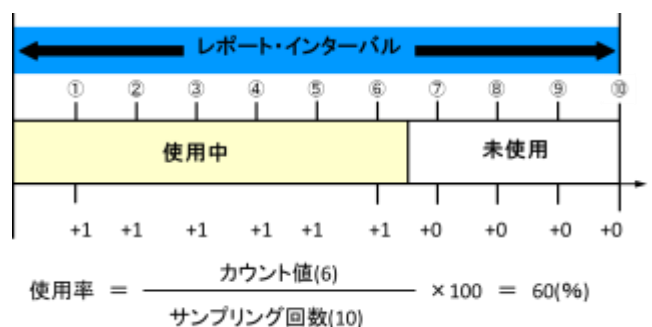
使用中のソフトウェア・モニタのインターバルが15分から30分以上に設定されています。評価精度を向上させるためには、インターバルを15分から30分に設定して下さい。

## 【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、PDL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】

- CPECNVRTプログラムで共通レコード形式に変換する際のインターバルを15分から30分以内に設定する。

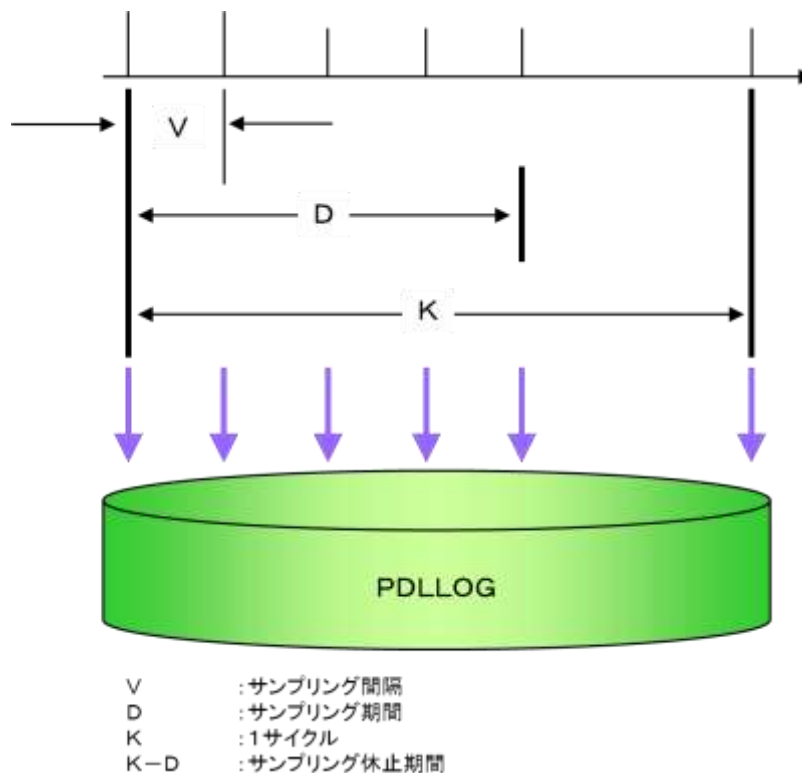
## OSP03n (2/2)

## 【注意点】

## ■ PDLの注意点

PDLの場合、メーカー提供のパラメータで動作させるとシステム負荷と出力データ量が伴に大きくなります。一般的に、一日中PDLを動作させるとフル・テープ1巻程度のデータが出力されます。この為弊社のカスタマ・サポートでは、PDLのデータ収集プログラムの特性などを吟味し標準パラメータを設定しております。詳細については『ES/1 NEO MF シリーズ ハンドブック』を参照して下さい。

PDLのデータ収集プログラムの内、VOLUME、SYSTEM、INIT、SVCの4つについてはシステム負荷と出力データ量が多いため、特に注意する必要があります。



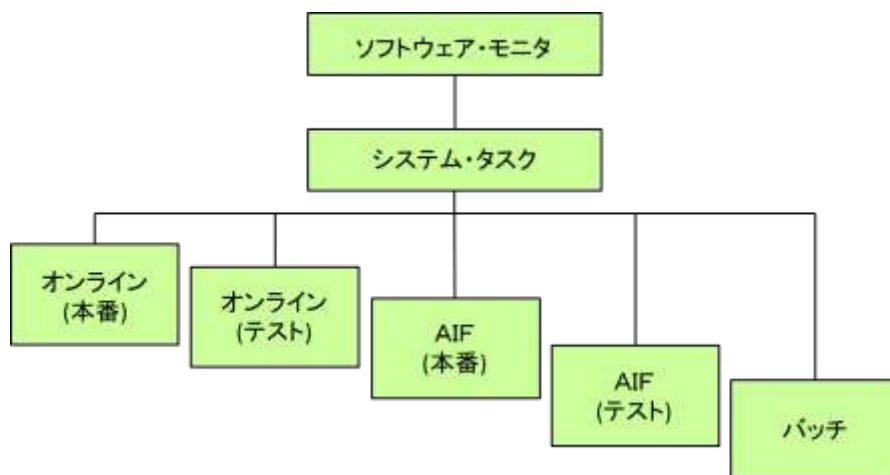
## PROC01n (1/2)

## 【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りますと、優先順位の低い業務プログラムがそのリソースに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。

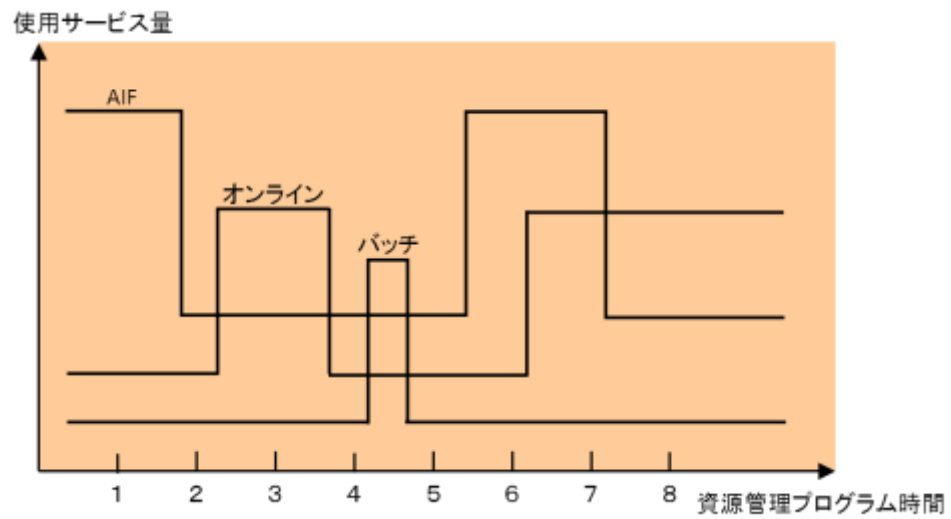
- ① ディスパッチング・プライオリティには、固定と自動制御(CPUバウンド／NON-CPUバウンド)の2種類があります。固定はシステム・タスクやオンラインで、AIFやバッチは自動制御で指定して下さい。
- ② ソフトウェア・モニタ(PDL)には、システムで最高のディスパッチング・プライオリティを割り当てて下さい。
- ③ ジョブ入力制御プログラムやネットワーク制御プログラム(VTAM等)等のシステム・タスクには、ソフトウェア・モニタの次のディスパッチング・プライオリティを割り当てて下さい。
- ④ AIFおよびバッチは全て自動制御で指定して下さい。通常のシステムでは、オンライン＞AIF＞バッ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、CPU最低保証率を指定してください。

＜次頁へ続く＞



## PROC01n (2/2)



## 【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。
- 最大実行多重度を制限する。
- CPU最低保証率を指定する。
- プロセッサを増強する。
- PDLのMODULEサンプラーを使用して、高負荷モジュールを洗い出す。

## 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】

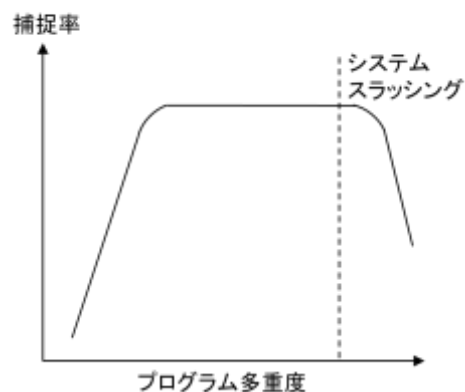
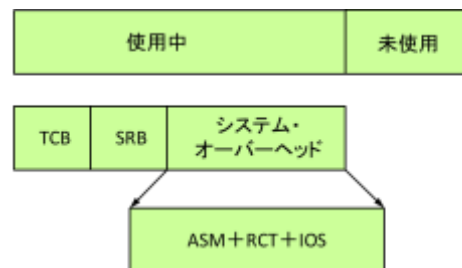
オペレーティング・システムにおけるプロセッサの使用率の内訳は右図のようになっています。ソフトウェア・モニタがプロセッサ使用率として求めているのが、プロセッサが使用中であった時間の割合です。また、各パフォーマンス・グループで実行される業務プログラムごとにタスク(TCB)やアクティビティ(SRB)で使用されたプロセッサ時間が報告されます。この2つの時間の差がシステム・オーバーヘッドです。

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。



## STOR01n &amp; STOR02n

## 【説明】

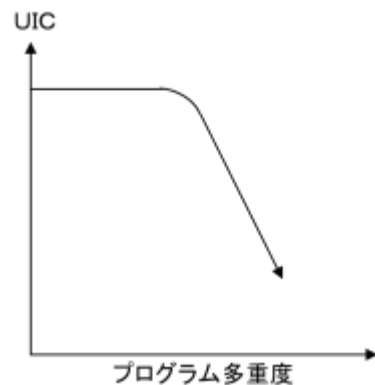
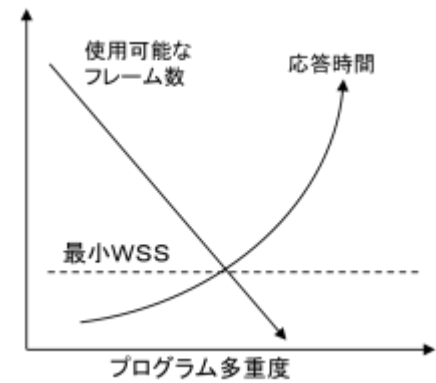
主記憶の競合が高まっております。主記憶の使用状況を評価すると同時に、必要であれば主記憶の増強を検討して下さい。

## 【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になります。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。

資源管理プログラムでは、主国の処理効率を向上させる目的でロールイン、ロールアウト機能を提供しています。その際、優先的に主記憶にロールインさせるために主記憶優先度をパフォーマンス・グループ単位に指定することができます。



## 【対応策】

- 主記憶優先度を見直す。
- 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。

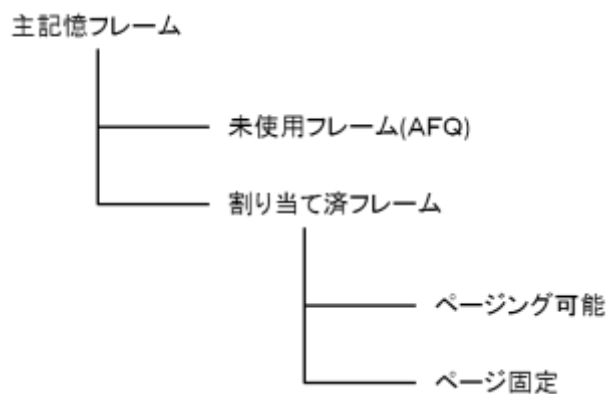
## 【解説】

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、アドレス空間のロール・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。



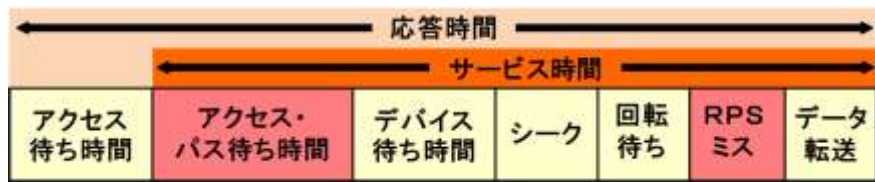
## IOSS01n (1/2)

## 【説明】

入出力サブシステムへのアクセス・パスを構成するチャンネルもしくはチャンネル・パスの使用率が高過ぎます。アクセス・パスのバランス化を考慮すると同時にディスク装置でのRPSミス時間に注意して下さい。

## 【解説】

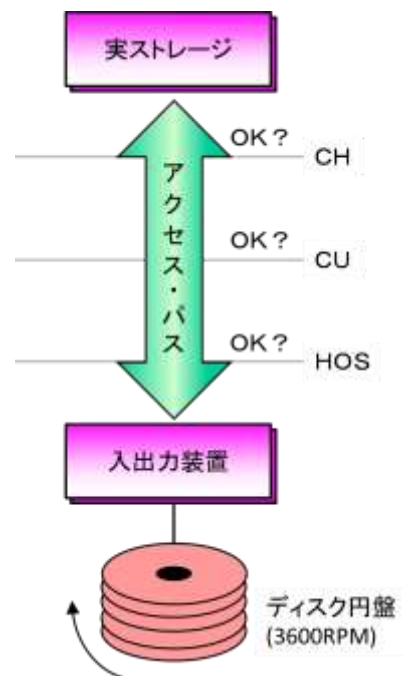
ディスク装置の応答時間は次のように分類できます。



これらの項目の内、アクセス・パスの使用率に関する項目は、RPSミス時間です。他の項目は、その他の要因によって決定されます。

RPSミス時間はシーク動作完了後、目的レコードがリード/ライト機構の直前に到着した時点で、入出力装置と実ストレージ間のアクセス・パスを専有できるか否かによって、その長さが決まります。もし、アクセス・パスを構成するチャンネル(CH)やコントロール・ユニット(CU)およびヘッド・オブ・ストリング(HOS)のいずれかでも、他の装置に専有されていれば、目的レコードがリード/ライト機構を通過してしまいます。この場合、データ転送を開始するためには目的レコードが再びリード/ライト機構の直前に到着するまで、1回転(16.7ミリ秒)の待ちが生じます。この時間のことをRPSミス時間と呼びます。

このRPSミスは、アクセス・パスを構成する3つの装置の全てが未使用状態で、そのアクセス・パスを専有できるまで繰り返されます。このためRPSミス時間は、次の確率計算で求めることが可能です。



$$\begin{aligned} \text{RPSミス時間} &= RV \times (1\text{回ミスする確率}) + RV \times (2\text{回ミスする確率}) \\ &+ RV \times (3\text{回ミスする確率}) + \dots \end{aligned}$$

$$\begin{aligned} \text{RPSミス時間} &= RV \times \rho + RV \times \rho^2 \\ &+ RV \times \rho^3 + \dots \end{aligned}$$

$$\text{RPSミス時間} = RV \times \frac{\rho}{(1-\rho)}$$

式中のRVはディスク円盤の回転時間であり、16.7ミリ秒です。また、 $\rho$ はアクセス・パスの使用率です。チャンネル使用率からアクセス・パスの使用率を求める場合には、次式を使用して下さい。



## IOSS01n (2/2)

## DPRの無いディスク装置のアクセス・パスの使用率

$$\frac{(\text{CHB1} \times \text{CHB2} \cdots \times \text{CHBn}) + (\text{CHB1} + \text{CHB2} \cdots + \text{CHBn})/n}{2}$$

## DPRの有るディスク装置へのアクセス・パスの使用率

$$\text{CHB1} \times \text{CHB2} \cdots \times \text{CHBn}$$

このように、RPSミス時間はチャネル・パスの使用率より理論的に求めることができます。また、チャネル・パスの使用率が増加すると、RPSミス時間は指数的に増加します。しかし、RPSミス時間を直接測定することができないため、無視されているのが現状です。

ディスク装置の性能が改善され、その処理速度が高速化されるにつれ、RPSミス時間の重要性が増しています。充分、RPSミスに注意し、ディスク装置のサービス時間を改善されることをお勧めいたします。

ディスク装置の回転数が3600RPM以外の場合は、回転時間が16.7ミリ秒ではなく、次のようになります。

$$1\text{回転時間 (ミリ秒)} = \frac{60}{\text{回転数}} \times 1000$$

メーカー名	製品名	回転数 (／分)
富士通	6425 (H以外)	3600
	6425H	4340

## 【対応策】

- ボリューム移動によるアクセス・パスの使用率をバランス化する。
- アクセス・パスを増強する。(例:ABBBの構成をABABの2系列にする。)

## IOSS02n

## 【説明】

ディスク装置へのアクセスを行う際の応答時間が長過ぎます。重要なディスク・ボリュームが指摘されている時は、その応答時間の内訳を調査の上対処して下さい。

## 【解説】

ディスク・ボリュームの評価を行う際には、応答時間の最悪値と平均値の両方を吟味する必要があります。応答時間の最悪値は、各インターバル毎の変動を示し、特に重要なディスク・ボリュームの時は注意しなければなりません。一方、応答時間の平均値は、ディスク・ボリュームの負荷バランス等を評価する場合に有効です。これはI/Oスキャン機能で評価できます。

このチューニング・ヒントは、ディスク・ボリュームの最悪応答時間を評価した結果です。この場合は、応答時間の内訳を調査する必要があります。ディスク・ボリュームの応答時間は次のように分類できます。



アクセス待ち時間

同一ディスク・ボリュームに複数の業務プログラムからの要求が同時に発生した時の待ち時間

アクセス・パス待ち時間

チャネルや制御装置が使用中のため、入出力要求が待たされた時間

デバイス待ち時間

共用ディスク・ボリュームによる待ち時間

シーク

ヘッドが目的シリンダへ移動する時間

回転待ち

回転数に依存した定数

60

$\frac{60}{\text{回転数} \times 2} \times 1000 \text{ (ミリ秒)}$

(例) 3600RPM の時は約 8.3 ミリ秒

RPS ミス

データ転送を行う際、アクセス・パスが使用中のために待たされた時間

データ転送

データ転送に要した時間

ディスク・ボリュームの応答時間の内訳を評価した結果が他のチューニング・ヒントに指摘されますので、そのチューニング・ヒントを参照して下さい。

## 【対応策】

- アクセス待ち時間の場合、IOSS03nを参照して下さい。
- アクセス・パス待ち時間の場合、IOSS07nを参照して下さい。
- デバイス待ち時間の場合、IOSS04nを参照して下さい。
- シーク時間の場合、IOSS06nを参照して下さい。
- RPS ミス時間の場合、IOSS01nを参照して下さい。

## IOSS03n (1/2)

## 【説明】

ディスク・ボリュームへのアクセスを行う際のアクセス待ち時間が長過ぎます。複数の業務プログラムからのアクセス要求が同時に発生しているため、アクセス対象のデータセットを調査し、データセットの分散/分割を検討して下さい。

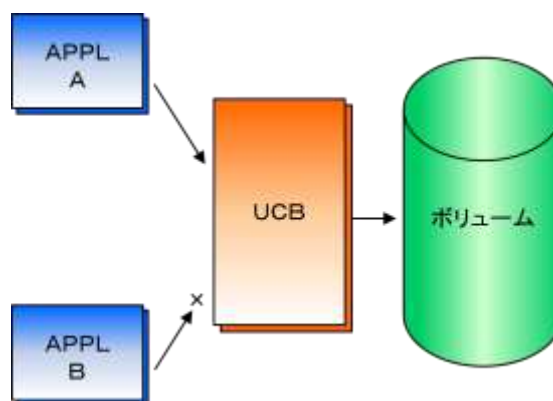
## 【解説】

ディスク装置の場合、1台の装置にマウントされた1つのディスク・ボリューム内に、複数のデータセットを配置し、同時に使用することが可能です。同一ディスク・ボリューム内の複数データセットを1つの業務で使用する際には、シーク時間に注意するだけで事足ります。しかし、複数の業務プログラムで同一ディスク・ボリューム内のデータセットを使用する際には、そのアクセス要求はシリアルライズ(逐次化)されるため、アクセス待ち時間にも注意する必要があります。

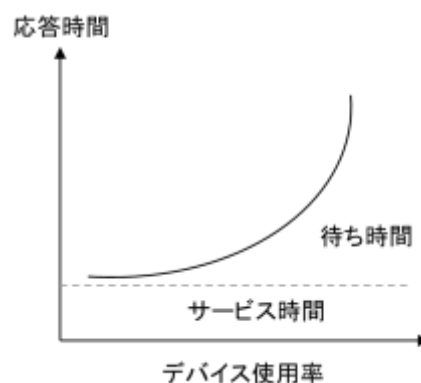


アクセス待ち時間とは、先行した入出力動作が完了するまで入出力装置の制御テーブルであるUCB(ユニット・コントロール・ブロック)で新たなアクセス要求が待たされる時間のことです。各ソフトウェア・モニタは、このアクセス待ち時間を各ディスク装置毎に時間 かもしくはキュー長でレポートします。複数の業務プログラムが同一ボリュームを全くランダムにアクセスする際の待ち時間は待ち行列技法のM/M/1の公式で求めることができます。

$$\text{アクセス待ち時間} = \text{サービス時間} \times \frac{\rho}{(1-\rho)}$$



式中の  $\rho$  はデバイス使用率であり、サービス時間は競合が無い状態でアクセスした時の応答時間です。この計算結果は右図のようになります。アクセス待ち時間は、サービス時間の半分程度に抑えるべきであると考えられます。前述の式にこの条件を代入すると デバイス使用率が33.33...%であるとの結論を得ることができます。



$$\text{アクセス待ち時間} \leq \text{サービス時間} \div 2 \rightarrow = \frac{1}{2} = \frac{\rho}{(1-\rho)} = \frac{0.333...}{0.666...}$$



**IOSS03n (2/2)**

この根拠で、長い間、ディスク装置のデバイス使用率は30%程度に保つようとのガイドラインが設定されていました。

アクセス待ち時間がサービス時間の半分以上になりますと、どのように高速のデバイスを導入しても、その効果を半減させていると言えます。充分、アクセス待ち時間に注意し、ディスク装置の応答時間を改善されることをお勧めいたします。

**【対応策】**

- 同時にアクセスされているデータセットを他のボリュームに分散する。
- 単一の区分データセットが同時にアクセスされている場合は、そのデータセットを分割し、他のボリュームに分散する。
- 資源管理プログラムへのパラメータで入出力優先順位制御を使用すると、アクセス待ち時間の長いボリュームへの優先制御が行えます。

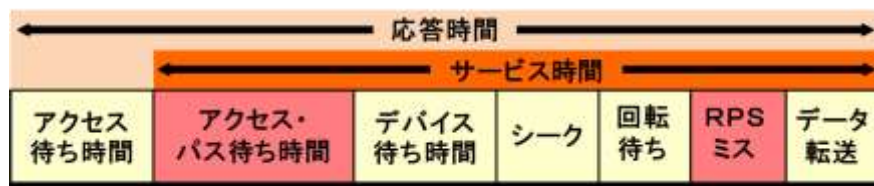
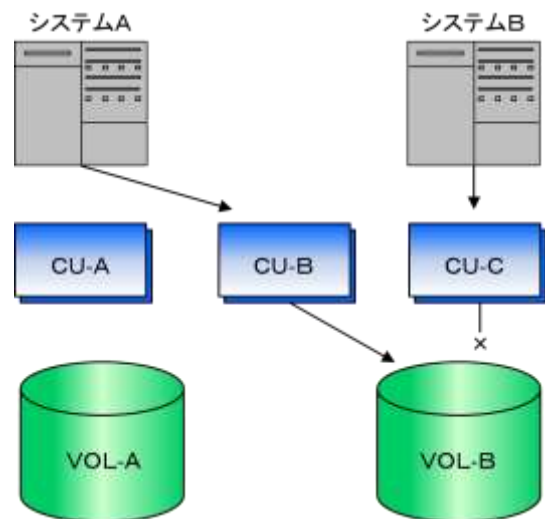
## IOSS04n

## 【説明】

複数のシステムで共用されているディスク・ボリュームのデバイス待ち時間が長過ぎます。各システムからのアクセスの正当性を確認し、データセットの分散等を検討して下さい。

## 【解説】

同一サイトにおいて、複数システムを運用している場合、ディスク・ボリュームを共用することがあります。このようなディスク・ボリュームをアクセスする場合、その実行結果に矛盾が発生しないようにするため、リザーブ指令を使用した排他制御を行います。例えば、システムAでVOL-BのVTOCを更新する場合、システムBからVOL-Bへのアクセスができないように、リザーブ指令を実行します。一度リザーブ指令が実行されると、次にリリース指令が出されるまで、システムBからのアクセス要求は実行を拒否されます。このために生じる待ち時間をデバイス待ち時間と言います。



このデバイス待ち時間が応答時間に占める割合が大きい場合は、各システムからのアクセス要求の正当性を調査します。そのアクセス要求が不当な場合は、その要求者を排除し、正当な場合は、該当ディスク・ボリューム内のデータセットの使用状況を調査し、データセットを他のディスク・ボリュームに分散させることを検討して下さい。

## 【対応策】

- 共用ディスク・ボリュームへのアクセス要求の正当性を確認する。
- データセットを他のディスク・ボリュームに分散する。
- 運用において各業務プログラムの実行をシフトする。

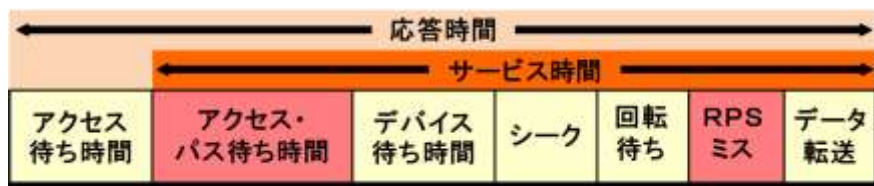
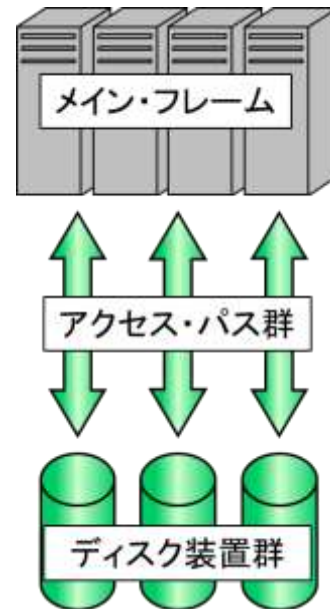
## IOSS05n

## 【説明】

オペレーティング・システムがディスク・ボリュームをアクセスするルート(アクセス・パス)毎のディスク・ボリュームの負荷が均等ではなく、大きな片寄りがあります。負荷を分散し、均等化を図って下さい。

## 【解説】

入出力サブシステムは複数のディスク装置とそれをアクセスするルートであるアクセス・パス群により構成されています。このようなサブシステムのパフォーマンスを良好に保つ唯一の手段が、アクセスの負荷を各ハードウェア資源に均等に分散させることです。もし、負荷の分布に片寄りが発生しますと、負荷が高いハードウェア資源がシステム・ボトルネックとなります。この負荷の分布状況は、アクセス・パス(論理チャネルもしくは論理制御装置)とディスク・ボリュームの単位で監視します。アクセス・パスの負荷分布に片寄りがある場合は、ディスク・ボリュームのRPSミス時間に注意して下さい。また、ディスク・ボリュームの負荷分布に片寄りがある場合は、アクセス待ち時間に注意して下さい。



## 【対応策】

- アクセス・パスのバランス化はボリューム移動(負荷の低いアクセス・パスのディスク・ボリュームと入替え)で対応する。この場合、チューニング・ヒント“IOSS08n”に示されるディスク・ボリューム群を参考にして下さい。
- ディスク・ボリュームのバランス化は、データセットの分散で対応する。

## 【注意点】

- このチューニング・ヒントは、I/Oスキャン機能(特定の時間帯)の評価結果です。選択された時間帯を確認して下さい。

## IOSS06n (1/2)

## 【説明】

ディスク・ボリュームの負荷が均等ではなく、大きな片寄りがあります。負荷の高いディスク・ボリュームで応答時間の長いディスク・ボリュームを原因コードで示された項目に対応したチューニング手法を選択して実施して下さい。

## 【解説】

入出力サブシステムの評価を行う場合、重要なディスク・ボリュームもしくは負荷の高いディスク・ボリュームに着目する必要があります。チューニングを実施する場合には、各ディスク・ボリュームの応答時間の内訳を調査し、チューニング手法を選択しなければなりません。ディスク・ボリュームの応答時間は次のように分類されます。



ディスク・ボリュームの応答時間の内訳で、アクセス・パス待ち時間はアクセス・パスの競合による待ち時間で1～5ミリ秒程度の時間であり、回転待ち時間は定数で、データ転送時間は、データブロック長とデータ転送速度により決定されるものです。このため、ディスク・ボリュームの応答時間を大きく変動させる要因には、アクセス待ち時間(Q)、デバイス待ち時間(C)、シーク時間(S)、RPS ミス時間(R)の時間要素があります。

$$\text{デバイス待ち時間} = \frac{60}{\text{回転数} \times 2} \times 1000 \text{ミリ秒}$$

(注)回転数についてはIOSS1nを参照してください。

ES/1 NEOでは、実測されたディスク・ボリュームのデータから各時間要素を計算し、これらの時間が、応答時間の3分の1以上を占めるようであれば、対応するチューニング・ヒントを出力します。

各時間要素が長くなる原因としては次のことが考えられます。

- アクセス待ち時間 (Q) : 複数の業務プログラムでの競合 (IOSS03nを参照して下さい。)
- デバイス待ち時間 (C) : 複数システムでの競合 (IOSS04nを参照して下さい。)
- シーク時間 (S) : アクセス機構の競合
- RPS ミス時間 (R) : アクセス・パスの競合 (IOSS01nを参照して下さい。)

<次頁へ続く>

## IOSS06n (2/2)

ここでは、シーク時間について説明します。シーク時間について考察する場合、次の3つの点に注意して下さい。

- ① 異なったデータセット間のアクセスの競合  
同一ボリューム内の複数データセットが同時にアクセスされると、そのデータセット間をリード/ライト機構が移動するためのシーク時間が増加します。
- ② 同一データセットの異なったエクステント間のアクセスの競合  
1つのデータセットが1つの連続した領域(エクステント)で構成されるとは限りません。複数のエクステントで構成されたデータセットをアクセスする際にも、シーク時間が増加します。
- ③ 巨大エクステント内でのアクセスの競合  
非常に大きな連続した領域で構成されたデータセットであっても、シーケンシャル以外での処理を行う場合に、シーク時間は増大します。

## 【対応策】

- アクセス待ち時間の場合は、IOSS03nを参照して下さい。
- デバイス待ち時間の場合は、IOSS04nを参照して下さい。
- RPSミス時間の場合は、IOSS01nを参照して下さい。
- シーク時間の場合は、次の事を行って下さい。
  - ・同時にアクセスされているデータセットを他のボリュームに分散する。
  - ・ボリューム内でのデータセットの配置を適正化する。
  - ・データセットのエクステント数を減少させる。
  - ・PDLのVOLUMEサンプラーでシーク時間を監視する。

## 【注意点】

- このチューニング・ヒントは、I/Oスキャン機能(特定の時間帯)の評価結果です。選択された時間帯を確認して下さい。

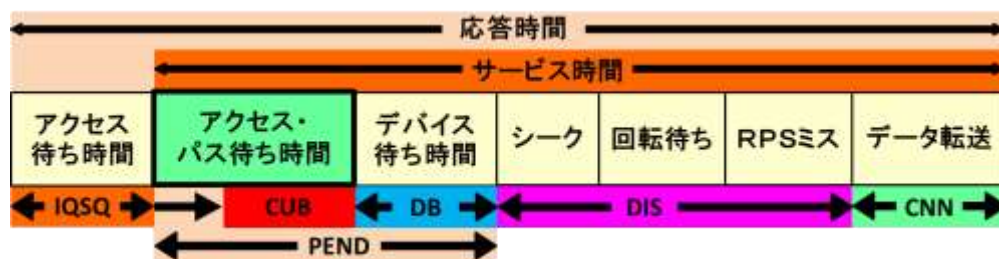
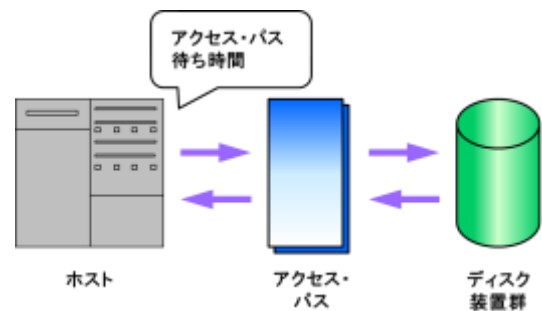
## IOSS07n

## 【説明】

論理制御装置の遅延時間が長過ぎます。この論理制御装置に接続されているディスク装置の応答時間に注意して下さい。

## 【解説】

オペレーティング・システムがディスク装置に入出力要求をハードウェアに実行させる際に、チャネルや制御装置が使用中のために待たされた時間をアクセス・パス待ち時間と呼びます。ソフトウェア・モニタでは、ディスク・ボリュームの応答時間を次のようにレポートします。



ES/1 NEOでは、ソフトウェア・モニタが計測したデータから次の計算式を使用してアクセス・パス待ち時間を求めています。

$$\text{アクセス・パス待ち時間} = \text{ペンディング時間 (PEND)} - \text{デバイス待ち時間 (DB)}$$

このディスク・ボリューム毎のアクセス・パス待ち時間をアクセスするルート単位である論理制御装置で合計した時間を評価しています。この時間が長い場合、この論理制御装置に接続されているディスク・ボリューム群へのアクセス要求が多く、ディスク・ボリュームの応答時間にも影響がでていると思われます。各ディスク・ボリュームの負荷を調査し、その負荷の分散を検討して下さい。

## 【対応策】

- ディスク・ボリュームの負荷を調査し、ディスク・ボリュームの移動やデータセットの移動を行う。
- チャネルを増設する。

**IOSS08n**

**【説明】**

ディスク・ボリュームの負荷を評価した際に、負荷の低いディスク・ボリューム群を示しています。負荷の分散を行う時の候補として下さい。

**【解説】**

入出力サブシステムは、複数のディスク装置とそれをアクセスするルートであるアクセス・パス群により構成されています。このようなサブシステムのパフォーマンスを良好に保つ唯一の手段が、アクセスの負荷を各ハードウェア資源に均等に分散させることです。もし負荷の分布に片寄りが発生しますと、負荷の高いハードウェア資源がシステム・ボトルネックとなります。

負荷を分散する場合には、このチューニング・ヒントで示されたディスク・ボリューム群を移行先の候補として下さい。

**【対応策】**

- 負荷分散時の移行先を候補リストから選択する。

**【注意点】**

- このチューニング・ヒントはI/Oスキャン機能(特定の時間帯)の評価結果です。選択された時間帯を確認して下さい。

## IOSS10n

### 【説明】

入出力サブシステムを評価する場合、ディスク・ボリュームの応答時間の最悪値と平均値を吟味する必要があります。平均応答時間を調査するI/Oスキャン機能が実行されませんでした。実行パラメータを設定して下さい。

### 【解説】

入出力サブシステムを評価するために、I/Oスキャン機能が用意されています。この機能では、特定の時間帯におけるディスク・ボリュームの平均応答時間とアクセス負荷を評価することができます。

各プロセッサの実行パラメータを確認し、設定して下さい。

### 【対応策】

- 各プロセッサの実行パラメータを確認し、設定する。



## IOSS11n

### 【説明】

論理チャンネルか、もしくは論理制御装置のデータがありません。

### 【解説】

入力されたパフォーマンス・データ内に入出力サブシステムの構成データが見つかりませんでした。これは、システム特有の制限事項です。このため、チャンネルの評価が出来ていません。

この現象は次の場合に発生します。

- 仮想マシン (AVM) 下で稼働しているオペレーティング・システム
- 国産システムの一部

### 【対応策】

- 特になし。

## IOSS12n &amp; IOSS13n

## 【説明】

入出力サブシステムにキャッシュ・コントローラがあるか否かを表しています。

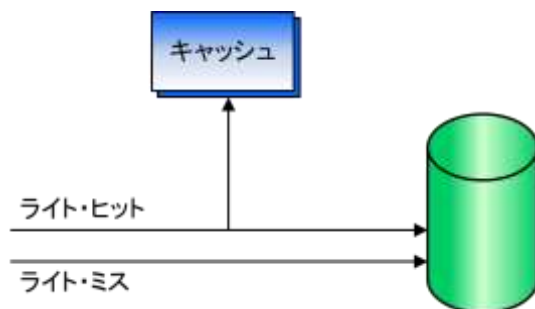
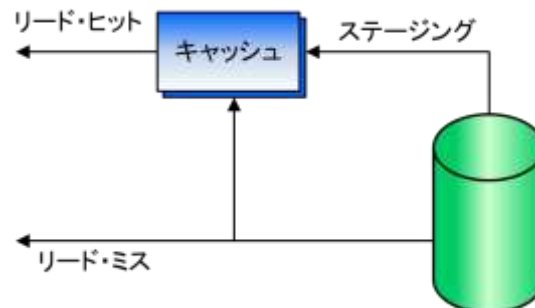
## 【解説】

キャッシュ・メモリーを内蔵したディスク制御装置には、そのキャッシュ・メモリーをリードとライトの両方で使用するものと、リードのみで使用するものの2種類があります。また、キャッシュ・メモリーの制御モードにも、ノーマル、シーケンシャル、インビット、バイパスの4種類のモードがあります。ノーマル・モードでのキャッシュ・コントローラの動作は次のようになっています。

リード・ヒットとは、目的データ・ブロックがキャッシュ内に記憶されていた場合のことです。この場合、ディスク装置へのアクセスを行う必要がなく、シークや回転待ちおよびRPSミスなどの時間が不要となります。また、データ転送はチャンネルの最大転送速度で行われます。リード・ミスは、目的のデータ・ブロックがキャッシュ内になかった場合のことです。この場合、通常のアクセスと同様にディスク装置からチャンネルとキャッシュへのデータ転送を行います。この際、後続のデータ・ブロックをキャッシュにロードする作業(ステージング)を単独で行う装置もあります。

ライト・ヒットとライト・ミスは書き出そうとする目的データ・ブロックがキャッシュ内にあるか否かによる分類です。いずれの場合も、ディスク装置をアクセスし、実際のライト動作を行います。ライト・ヒットの場合、キャッシュ内の目的データ・ブロックも同時に更新されます。

ES/1 NEOでは、ディスク装置の種類やキャッシュ・メモリーの制御モードに関係なく、総アクセス回数の内、何%のアクセスがキャッシュ・メモリーへのアクセスだけで完了したかを論理ヒット率として求めています。コンピュータ・メーカーの保守技術員がディスク制御装置のログを調査して求めるヒット率やキャッシュ・レポータ・プログラムの報告するヒット率は、総リード・アクセス回数の内、何%がキャッシュ・メモリーへのアクセスだけで完了したかを示すリード・ヒット率です。両者の数値の意味が異なっていますので注意して下さい。



## 【対応策】

- キャッシュ・コントロールを使用している場合、キャッシュ・ボリュームとして認識されていないキャッシュ・ボリュームがありましたら、そのパフォーマンスを調査して下さい。

## 【注意点】

- このチューニング・ヒントは、I/Oスキャン機能(特定の時間帯)の評価結果です。選択された時間帯を確認して下さい。

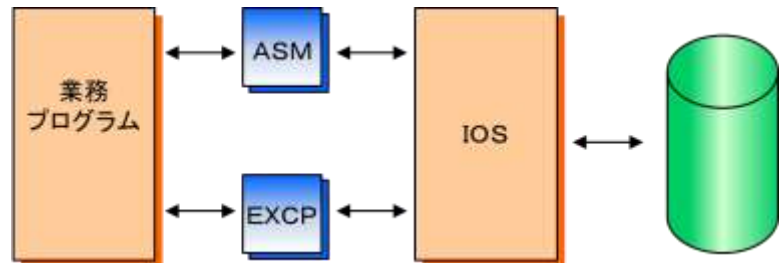
## PAGE01n &amp; PAGE02n

## 【説明】

JOBEPSが1つしか存在していないか、または、それらのデータセットのサービス時間が長過ぎます。新たなJOBEPSを割り当てて下さい。

## 【解説】

オペレーティング・システムで運用されている業務プログラムは2種類の入出力動作を行っていますと言えます。一方がEXCPを経由した本来の入出力動作です。もう一つが、ストレージ競合により発生するページング動作であり、ASMを経由して実行されます。



このページングによる入出力動作は業務プログラムが意識していないため、特別な配慮が必要です。例えば、オンラインの制御空間でページ不在によるページング動作が発生すると、そのページング動作が完了するまでオンラインの制御空間の実行は中断されます。つまり、ページング動作中、オンラインの制御空間が実行すべきオンライン処理は、全て中断されてしまいます。このページング動作を無くすることは不可能ですし、また、重要な業務プログラムでページ不在によるページング動作が発生しないようにすることも困難です。このため、ページ不在によるページング動作は出来るだけ高速処理されることが望まれます。

ページ・データセットとして使用されるディスク装置のサービス時間には、ボリュームやそのアクセス・パスの競合などにより変動するシーク時間やRPSミス時間が含まれています。このため、単一ボリュームにだけ頼ったページング動作を行うことは危険です。ASMは、このリスクを分散するために複数のJOBEPSをサポートします。また、複数のJOBEPSの内、高速処理が可能なデータセットを選択して、ページング動作を行うようにしています。

オペレーティング・システムの効果的な運用を考える場合、最低でも2つのJOBEPSを割り当てることが必要です。もし、複数のJOBEPSが割り当てられているにもかかわらずサービス時間が悪化する場合は、データセットが割り当てられているディスク装置のアクセス・パスの使用率を調査して下さい。特に、改善点がなければ、新たなJOBEPSを追加して下さい。

## 【対応策】

- 必ず複数のJOBEPSを割り当てる。
- JOBEPSのアクセス・パスの使用率は低く保つ。
- 半導体ディスク装置や拡張記憶機構の導入を検討する。

PAGE03n

【説明】

JOBEPSに割り当てられているスロットの大半を使用しております。JOBEPSの総容量を増やして下さい。

【解説】

JOBEPSは、仮想空間(アドレス・スペース)の私有域を格納するページ・データセットです。ISP機能を使用中のシステムではJOBEPS15が必須となります。このJOBEPSの総容量は、下記の理由により、大きめに設定されることをお勧めいたします。

- ① ASMは、1回のページング処理で複数のページ・アウト動作を行おうとします。このため、連続した空きスロットや同一シリンダ内の空きスロットを捜して使用します。十分な空きスロットが無い場合には、ページングのサービス時間が長くなります。
- ② JOBEPSが満杯に近くなりますと、新たなプログラムの起動が禁止されたり、スワップ可能なプログラムはスワップ・アウトされます。このため、システムのスループットが低下することもあります。

【対応策】

- 新たなJOBEPSを割り当てる。

## PAGE04n &amp; PAGE05n

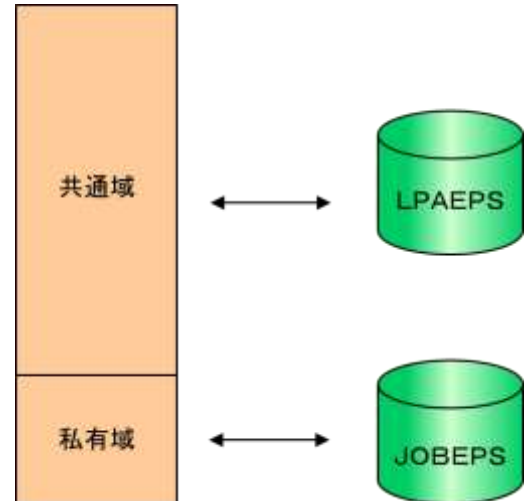
## 【説明】

LPAEPSのサービス時間が長過ぎます。より高速処理が可能なディスク・ボリュームに、それらのページ・データセットを移動させて下さい。

## 【解説】

オペレーティング・システムの仮想空間(アドレス・スペース)のページング域には、私有域と共通域の2種類の領域があります。私有域は、実行されるプログラムが固有に割り当てられるべきページング域のことです。このため、私有域の総容量は、並行に実行されるプログラム数に比例して増加します。一方、共通域は実行中の複数のプログラムが共用したり、オペレーティング・システムが使用するページング域です。このため、その容量は私有域に比べ極めて小さいと言えます。

これらのページング域の外部記憶データセットとして、LPAEPSならびにJOBEPSが用意されています。私有域のページ・データセットは、私有域の容量が大きいので、複数のデータセットで構成できるようになっています。



ページング処理を行うASMは、複数のJOBEPSの内、最も高速なページ・データセットを使用することにより、そのページ不在時間を最小にしようとしています。しかし、共通域用のLPAEPSは1つしか割り当てることができません。このため、ローカル・ページ・データセットのようなASMのロジックによる助けを期待することはできません。共通域のページング速度はシステム全体のスループットに大きく影響するため、パフォーマンス管理者は、LPAEPSを高速処理が可能なディスク・ボリュームに割り当てるように努力する必要があります。

## 【対応策】

- LPAEPSをより高速なディスク・ボリュームに移動させる。
- 同一ボリュームに複数ページ・データセットを割り当てない。
- 半導体ディスクや拡張記憶機構の導入を検討する。

## PAGE06n

## 【説明】

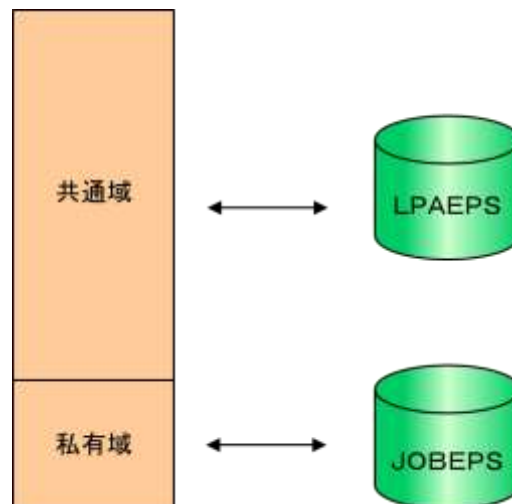
同一ディスク・ボリュームに複数のページやスワップのデータセットが割り当てられています。シーク時間によるサービス時間の悪化を防止するために、それらのデータセットを他のボリュームに分散して下さい。

## 【解説】

仮想記憶方式を使用するオペレーティング・システムでは、実記憶と仮想空間（アドレス・スペース）の差を埋め合わせるためのページング動作が必要となります。オペレーティング・システムでは、このページング処理を効果的に行うために、ページング対象となる仮想空間領域の特性に応じて2種類のデータセットを使用します。この2種類のデータセットは、共通域用のLPAEPSと私有域用のJOBEPSにグループ分けできます。私有域は共通域に比べ、その総容量が大きいいため、複数のデータセットでページングを処理させることができます。

一般的に、ページングは実記憶の競合が高まるにつれ、指数的に増加すると言えます。また、どの領域のページングが全体の何%程度であるかは、その領域が全仮想空間に占める割合に比例すると考えられます。このため、ページングが発生し始めると、全てのタイプのページ・データセットへのアクセスが同時に発生する確率が高まります。元来、このページング動作は高速処理されるべきものですので、これ

らのページ・データセットは専用のディスク・ボリュームに配置しなければなりません。しかし、専用化が困難なシステムでは、これらのページ・データセットと他のデータセットを同一ボリュームに配置する場合があります。しかし、前述のようにページ・データセットへのアクセスは並行処理されるため、同一ボリュームに複数のページ・データセットを割り当てますと、シーク動作による遅れ時間が大きな問題となります。



## 【対応策】

- 同一ディスク・ボリュームに配置されているページやスワップのデータセットを他のボリュームに分散して下さい。

## PAGE07n

## 【説明】

ページデータセットが他のシステムと共用可能なディスク・ボリュームに割り当てられています。他のシステムからのアクセスとの競合により、サービス時間が長くなる可能性がありますので留意して下さい。

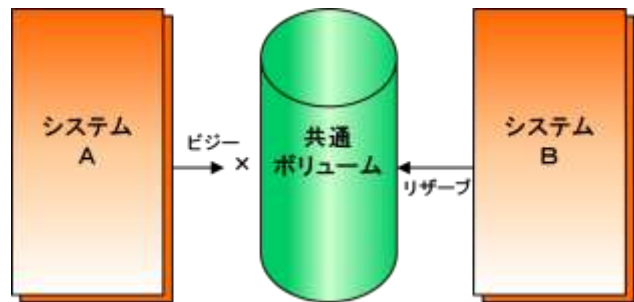
## 【解説】

複数のシステムを運用中のセンターにおいては、全てのシステムから1つのディスク・ボリュームを共用することができます。このようなボリュームのことを共用(シェアラブル)ボリュームと呼びます。

共用ボリュームをアクセスする際には、そのアクセスが他のシステムの動作と矛盾が生じないようにするため、各種の制御が介在します。

オペレーティング・システムでは、データセットのアロケーションに伴うVTOCデータの更新時などにその共用ボリュームをリザーブする場合があります。共用ボリュームが特定システムにリザーブされますと他のシステムからのアクセスはデバイス・ビジー(ディスク装置が使用中の状態)を理由にその実行が拒否されます。

ページデータセットを配置するボリュームで、他のシステムでリザーブされるような事態が発生することは回避する必要があります。このため、ページデータセットは共用ボリュームに配置しないようにして下さい。また、やむを得ず共用ボリュームを使用する際には、極力、他のシステムからそのボリュームをアクセスする回数を減少させて下さい。



## 【対応策】

- ページデータセットを非共用ボリュームへ移す。
- 他のシステムから、その共用ボリュームをアクセスする回数を減らす。

## SWAP10n

## 【説明】

主記憶のページ可能フレーム数が不足したことによるロールイン、ロールアウトが発生しています。このような事態は発生するべきではありません。原因を調査し、必要であれば、主記憶を増強して下さい。

## 【解説】

主記憶のフレームの状態を分類すると下図のようになります。この内、使用状態にあるフレーム数は、システムの業務負荷が上昇するに比例して、増加します。当然、ページ固定状態のフレーム数(ページ固定域)も同様に増加します。もし、主記憶容量以上の使用要求があるシステム負荷を与えますと、ページ固定域は増加しますが、ページ可能状態のフレーム数(ページ可能域)は逆に減少します。

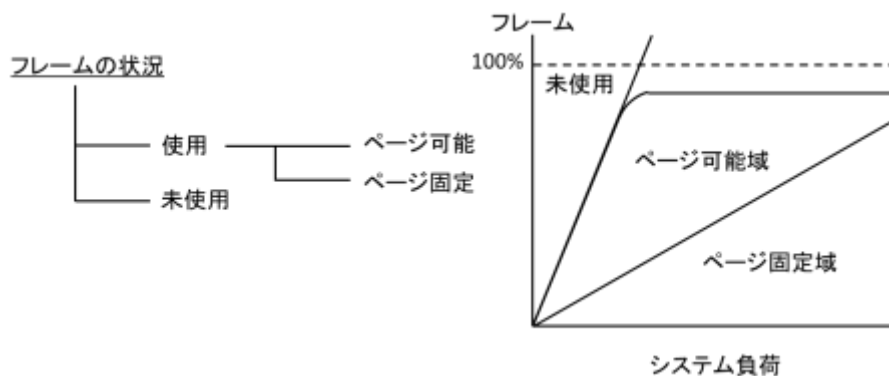
このような状態が発生し始めますと、適用業務プログラムが自由に使用できる主記憶域が減少するため、ページングによる影響を考慮する必要があります。ページングが発生しますと、ページングの入出力動作中はそのフレームは固定状態となります。つまり、ページングが発生しますとさらにページ固定域が大きくなります。

資源管理プログラムでは、このような主記憶のページ固定域が増加することによるシステムの性能低下を防止しようとします。つまり、ある一定以上の主記憶フレームが固定状態になると、新たなプログラムの開始を阻止すると同時に、その原因となった適用業務プログラムをロール・アウトします。新たなプログラムの開始を阻止するために資源管理プログラムは次のような処置を行います。

- ログオンやマウント・コマンドおよびスタート・コマンドの実行阻止
- イニシエータが新たなジョブの処理を開始することを阻止

## 【対応策】

- PDLでページ固定量の大きい適用業務プログラムを監視する。
- 同様のソフトウェア・モニタでストレージ使用量の大きい適用業務プログラムを監視する。
- 主記憶を増強する。





## VS03n &amp; VS04n

## 【説明】

16MBのアドレス境界以下のFSQA、もしくはPSQAが満杯になりつつあります。対応する領域を拡張して下さい。

## 【解説】

FSQAやPSQAは、複数のシステム・タスクで共用される共通域であるため、使用目的に応じたサブ・プール番号で管理されています。このサブ・プール単位にGETMAINされた領域は使用中として管理されます。

しかし、1つのページを複数のサブ・プールで共用しないようにするため、一部のページには大半が未使用のままとなっています。このため、実際に使用された領域の大きさ(バイト単位)と割り当てられた領域の大きさ(ページ単位)には差(フラグメンテーション)があります。FSQAやPSQAの使用率を監視する際には、領域サイズと割り当てられた領域の大きさに着目します。もし、この係数が85%を越えますと、その領域が満杯になりつつあると考えねばなりません。しかし、その使用率が安定していれば意図的に設定されているものと考えられます。

## 【対応策】

- FSQAもしくはPSQAを拡張する。



**VTAM01n****【説明】**

NCPのバッファ域が不足したため、NCPがスローダウン・モードとなっています。NCPのチューニングが必要です。

**【解説】**

NCPは、自分が持つリソース(主にワーキング・ストレージ)の使用率が高くなり過ぎると、スローダウン・モードとなり、VTAMや端末よりのデータ受信を拒否します。このような現象が頻発するとネットワーク時間が長くなり、業務プログラムのレスポンス時間が悪化します。このスローダウン・モードに関連するパラメータには、次の3つがあります。

**① DELAY**

VTAMとNCP間の通信負荷を減少させるためのパラメータです。この値により、端末より受けたメッセージをNCP内でホールドしておく時間を調整します。

**② MAXBFRU**

端末よりNCPがメッセージを受け取る際に、NCPが準備すべきバッファ数を指示します。

**③ VPACING**

VTAMが1回のライト動作でNCPに渡すデータ量を指示します。

これらのパラメータの値が大きすぎますと、NCPのストレージ域が不足するために、スローダウン・モードが発生します。

**【対応策】**

- 上記3種のパラメータのチューニング
- 接続回線数(または端末数)を減少させる。
- 通信制御装置のストレージを増強する。

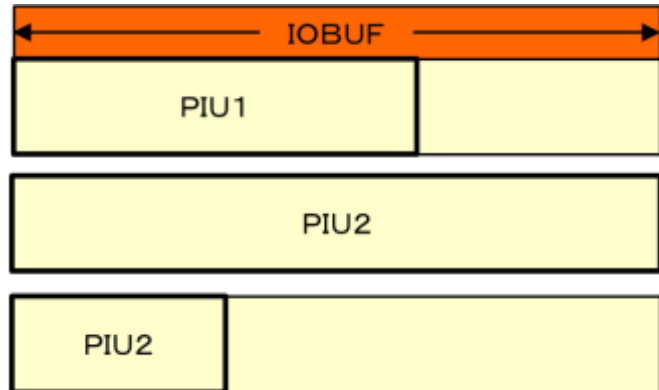
## VTAM02n

## 【説明】

VTAMのIOBUFが小さいため、経路情報単位 (PIU) が分割されています。IOBUFを大きくして下さい。

## 【解説】

VTAMがリード動作を実行する際には、入力されるべきPIUはIOBUFに格納される必要があります。このIOBUFには、1つのPIUしか格納されません。このため、IOBUFの大きさがPIUよりも大きいと、IOBUFの残った領域が無駄になります。しかし、IOBUFの大きさがPIUよりも小さいと、PIUが複数のIOBUFに分割されます。このPIUの分割処理には、プロセス負荷がかかるため、できるだけ回避して下さい。



## 【対応策】

- IOBUFの大きさを、平均のPIUの大きさより大きく設定する。

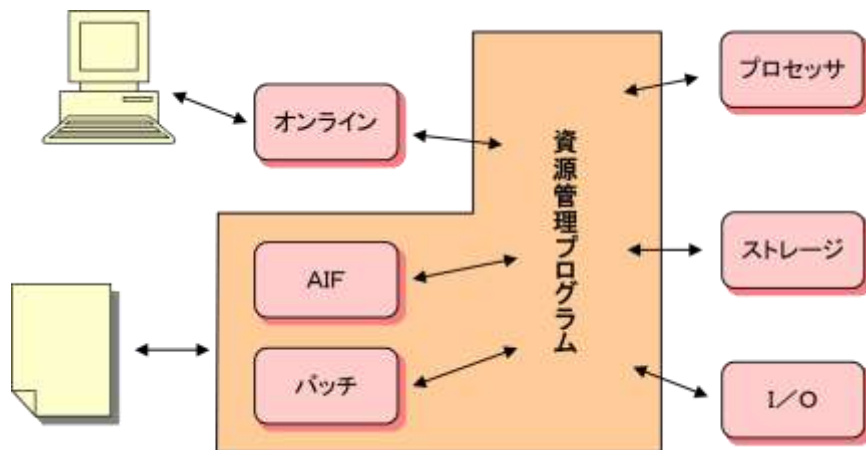
## WKLD01n

## 【説明】

業務プログラムが使用した各資源のバランス情報です。

## 【解説】

オペレーティング・システムには、そのシステムが持つ各資源(プロセッサ、ストレージ、I/O)と各業務とのバランスを考慮して、システムのスループットを最大限に引き出すためのコンポーネントとして『資源管理プログラム』があります。



この資源管理プログラムは、各システム資源を各業務に割り振る場合、パフォーマンス・グループ単位で制御しています。このため、全ての業務プログラムは、いずれかのパフォーマンス・グループに属し、各システム資源を使用します。資源管理プログラムは、各パフォーマンス・グループが資源を使用した量を管理/制御します。

このチューニング・ヒントでは、I/Oスキャン機能で選択された時間帯においてプロセッサ、ストレージ、I/Oを最も多く使用したパフォーマンス・グループ順にその割合と共に示しています。この情報から、各パフォーマンス・グループの特性(CPUバウンド、ストレージ・バウンド、I/Oバウンド)を知ることができます。

## 【対応策】

- 各パフォーマンス・グループでの資源使用状況を確認し、必要であれば、資源管理プログラムへのパラメータを変更して下さい。

## 【注意点】

- このチューニング・ヒントは、I/Oスキャン機能(特定の時間帯)の評価結果です。選択された時間帯を確認して下さい。

## 第2章 パフォーマンス・チューニング作業(AIM)

### 2.1 本章の使用方法

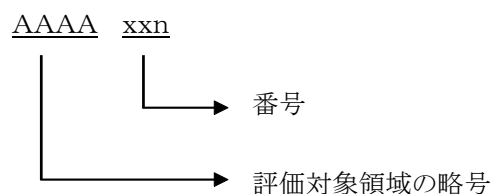
ES/1 NEO MF-XSPでは、各種のチューニング・ヒントを出力します。しかし、ES/1 NEO MF-XSPが出力するメッセージだけでは、その全てを説明することが困難です。この章では、富士通AIMオンラインサブシステムの評価結果として、ES/1 NEO MF-XSPが出力するチューニング・ヒントに対応したチューニング作業について解説します。なお、この章は“AIMPRT00の使用方法”と“添付資料C(ログデータ取得機能)”の章で説明されている事項を理解していることを前提で記述されています。また、チューニング作業の具体的な実施方法で疑問などがあれば、弊社担当者へお問い合わせ下さい。各種の方法でチューニング作業の支援を行います。

ES/1 NEO MF-XSPでは、評価対象の領域を下記のように分割し、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。

■BOFファイル	.....BOF
■データベース	.....DB
■データ通信	.....DC
■HLFファイル	.....HLF
■ロギング	.....LOG
■トランザクションの応答時間	.....RESP
■TLFファイル	.....TLF

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。ES/1 NEO MF-XSP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)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>



## BOF01x

## 【説明】

BOFファイルをアクセスする際のアクセス待ち時間が長過ぎます。アプリケーションリージョンに準備されるBOFバッファを拡張するか、もしくはBOFファイルをより高速化して下さい。

## 【解説】

トランザクションの処理中にアプリケーションプログラムが異常終了すると、データベースの更新内容を無効にしなければなりません。もし、この処理が正常に動作しなければデータベースの信頼性は著しく損なわれます。プログラムが異常終了した際に、データベースの内容を復元しAIMシステムの信頼性を保証する機能の事をダイナミックバックアウトと呼びます。

ダイナミックバックアウトでデータベースの復元を図る際には、プログラムがデータベースを更新したときのレコード位置と更新前のデータを使用します。つまり、プログラムが異常終了したときには、プログラムが更新したデータベースレコードに更新前のデータを書き込むだけです。このレコード位置と更新前のデータを記録しているのが、BOFファイルです。

このBOFファイルに記録されているデータが必要なのは、プログラムによるトランザクション処理が開始されてから、その処理が終了するまでです。つまり、BOFファイルの大きさは、同時に処理されるトランザクションの数とそれらのトランザクション処理中に更新されるレコード件数およびそれらのレコードの大きさにより決定されます。

BOFファイルに書き込むべきデータは、アプリケーションリージョンに確保されたBOFバッファに蓄積されます。このBOFバッファが充分大きければ、BOFファイルへのアクセスは発生しません。しかし、このBOFバッファの大きさが不足している場合、BOFファイルに対するログデータの書き込みが行われます。もし、BOFファイルへのログデータの書き込みが必要な場合、その書き込み動作が完了するまでアプリケーションプログラムの実行は中断されます。このため、BOFファイルは高速であることが必須となります。

BOFファイルへのアクセスに時間が掛かるか、もしくはBOFバッファが余りにも小さ過ぎてアクセス頻度が高い場合、BOFファイルをアクセスする際のアクセス待ち時間が増大します。このような事態は回避しなければなりません。まず最初に、BOFファイルへのアクセス頻度の減少を検討して下さい。つまり、アプリケーション毎に定義されたPEDコマンドでのBOFバッファ数の値を確認し、不足しているようであればその値を増加させて下さい。BOFバッファ数が充分であるならば、次にBOFファイルの高速化を検討して下さい。BOFファイルへのアクセスとデータベースへのアクセスは同期して発生します。このため、BOFファイルとデータベースが同一のディスクボリュームに割り当てられているとアームの競合などにより、BOFファイルへのアクセス時間が長くなります。

## 【対応策】

- PEDコマンドで指定されたBOFバッファ数を再度見直してください。
- BOFファイルとAIMデータベースが同一のディスクボリュームに割り当てられていないことを確認してください。
- 他の理由により、BOFファイルのアクセス時間が長くなっていないかを確認してください。
- 半導体ディスクがあれば、BOFファイルを半導体ディスクに割り当てて下さい。

## BOF02x

## 【説明】

BOFファイルのサービス時間が長過ぎます。BOFファイルをより高速のディスクボリュームに移動させるか、もしくはアプリケーションリージョンに準備されるBOFバッファを拡張して下さい。

## 【解説】

トランザクションの処理中にアプリケーションプログラムが異常終了すると、データベースの更新内容を無効にしなければなりません。もし、この処理が正常に動作しなければデータベースの信頼性は著しく損なわれます。プログラムが異常終了した際に、データベースの内容を復元しAIMシステムの信頼性を保証する機能の事をダイナミックバックアウトと呼びます。

ダイナミックバックアウトでデータベースの復元を図る際には、プログラムがデータベースを更新したときのレコード位置と更新前のデータを使用します。つまり、プログラムが異常終了したときには、プログラムが更新したデータベースレコードに更新前のデータを書き込むだけです。このレコード位置と更新前のデータを記録しているのが、BOFファイルです。

このBOFファイルに記録されているデータが必要なのは、プログラムによるトランザクション処理が開始されてから、その処理が終了するまでです。つまり、BOFファイルの大きさは、同時に処理されるトランザクションの数とそれらのトランザクション処理中に更新されるレコード件数およびそれらのレコードの大きさにより決定されます。

BOFファイルに書き込むべきデータは、アプリケーションリージョンに確保されたBOFバッファに蓄積されます。このBOFバッファが充分大きければ、BOFファイルへのアクセスは発生しません。しかし、このBOFバッファの大きさが不足している場合、BOFファイルに対するログデータの書き込みが行われます。もし、BOFファイルへのログデータの書き込みが必要な場合、その書き込み動作が完了するまでアプリケーションプログラムの実行は中断されます。このため、BOFファイルは高速であることが必須となります。

BOFファイルがAIMデータベースと同じディスクボリュームに割り当てられている場合、BOFファイルのサービス時間が長くなります。これは、ログデータの書き込みとデータベースの実更新が同期しているためです。このため、ディスクアクセスのアクセス待ち時間やアーム競合によるシーク時間が増大します。また、AIMを使用しているシステムでは、ページングが多いのが通常です。このため、BOFファイルとページング・データセットが同一のディスクボリュームに割り当てられている場合にも、同一の現象が発生します。

## 【対応策】

- BOFファイルとAIMデータベースが同一のディスクボリュームに割り当てられていないことを確認して下さい。
- 他の理由により、BOFファイルのアクセス時間が長くなっていないかを確認して下さい。
- 半導体ディスクがあれば、BOFファイルを半導体ディスクに割り当てて下さい。
- PEDコマンドで指定されたBOFバッファ数を再度見直して下さい。

## DB01x

## 【説明】

データベースのデッドロックが発生しています。デッドロックが発生しているデータベース・エクステントを調べ対応するアプリケーションプログラムを調査して下さい。デッドロックが発生することにより、何れかのトランザクションの応答時間が長くなっています。

## 【解説】

トランザクションの処理中にアプリケーションプログラムは多くのデータベースレコードを参照もしくは更新します。AIMシステムは複数のアプリケーションプログラムがアクセスするデータベースレコード群の排他制御を行うことにより、それらの実行結果に矛盾が発生しないようにしています。

たとえば、あるアプリケーションプログラムがデータベースレコードを更新した場合、そのアプリケーションプログラムが正常にトランザクション処理を完了するか、もしくはそのアプリケーションプログラムがチェックポイントコールを実行するまで、他のアプリケーションプログラムがそのレコードをアクセスできないようにします。

通常、アプリケーションプログラムを設計する際には、データベース更新に際してアクセスするデータベースレコードの順番を規定します。そのデータベースをアクセスする全てのアプリケーションプログラムはこの規定を遵守しなければなりません。もし、あるアプリケーションプログラムがその規定(アクセスするレコード順序)を守っていなければデッドロックが発生します。たとえば、Aのアプリケーションプログラムが $\alpha$ の次に $\beta$ のレコードを更新し、Bのアプリケーションプログラムが $\beta$ の次に $\alpha$ のレコードを更新する場合を想定して下さい。このとき、Aが $\alpha$ またBが $\beta$ を更新すると、次にAは $\beta$ またBは $\alpha$ をアクセスしようとして、しかし、その際には各々のレコードが他のアプリケーションプログラムに更新されているためにアクセスすることができません。この状態は、一度発生すると永遠に解除できません。

AIMはこの様なデッドロックを発見すると、その条件を解除するためにデッドロックに関与したアプリケーションプログラムの特性を調査し、一方のアプリケーションプログラムを疑似的に異常終了させます。すると、デッドロック条件がなくなり、異常終了させられなかったアプリケーションプログラムの処理を正常に実行させることが可能となります。

デッドロック条件を解除させるために、疑似的に異常終了させられたトランザクションはタスクSMQNにリンクされます。このタスクSMQNにリンクされたトランザクションはAIM端末から入力されたトランザクションと同様に、対応するタスクが動作可能状態になると直ちに処理されます。しかし、開始されたトランザクション処理が途中で中断され、再度実行されると言うことは、そのトランザクションの応答時間が長くなることを意味しています。このため、このデータベースのデッドロックは発生してはならないものであると言えます。

## 【対応策】

- デッドロックが発生したデータベース・エクステントから、問題が発生しているアプリケーションプログラムを調べて下さい。
- 問題のあるアプリケーションプログラムのロジックを調査し、必要であれば訂正して下さい。
- アプリケーションプログラムの訂正が困難である場合や、直ちに訂正することが困難である場合、対応するトランザクションの入力時間帯を制限するなどの方策も検討して下さい。



## DC01x

## 【説明】

端末とAIMシステム間のメッセージ転送に使用するDCMSバッファの枯渇が発生しています。DCMSコマンドで指定されたDCMSバッファの数を増加させて下さい。

## 【解説】

AIM端末からのメッセージは、VTAMにて受け取られます。VTAMは、その端末セッションが接続されたプログラムを調べ、AIMデータ通信の通信制御タスクに処理すべきメッセージが到着したことを通知します。AIMデータ通信の通信制御タスクは、VTAMバッファに読み取られている端末からのメッセージをAIMのリージョンに読み込むためにレシープマクロを実行します。この際、AIMリージョン内のバッファ領域として使用されるのがDCMSバッファです。また、DCMSバッファに読み込まれたメッセージは、アプリケーションプログラムでその処理が正常に完了するまでDCMSバッファ内に保管されます。

アプリケーションプログラムがトランザクションの処理を行った結果として端末へ送るレスポンスメッセージも、このDCMSバッファに転送されます。端末へのメッセージは、AIMの通信制御タスクがセンドマクロでVTAMに渡します。VTAMが端末へのデータ転送を完了すると、DCMSバッファに記録されていたメッセージが削除されます。

この様にDCMSバッファは、端末からもしくは端末へのメッセージ転送に使用されます。このDCMSバッファが不足すると、VTAMが端末からのメッセージを受け取った場合、AIMの通信制御プログラムがレシープマクロを実行することができません。このために、AIMシステムのトータルスループットが低下してしまいます。このような事態は回避する必要があります。

## 【対応策】

- PDLレポートで、どのDCMSバッファにおける枯渇が発生しているかを調査して下さい。
- 対応するDCMSコマンドのDCMSバッファ数を増加させて下さい。

## HLF01x

## 【説明】

HLFファイルをアクセスする際のアクセス待ち時間が長過ぎます。共通域に準備されるHLFバッファを拡張するか、もしくはHLFファイルをより高速化して下さい。また、TLFファイルを使用していないシステムでは、TLFファイルの使用も検討して下さい。

## 【解説】

AIMもシステムダウンすることがあります。しかし、システムダウンが発生したからといっても、データベースが破壊され、その回復が図れないという事態は回避しなければなりません。このために、AIMは実際のデータベース更新を行うたびに更新前のデータと更新後のデータをログファイルに書き出します。この場合、これらのデータが主記憶内のログバッファに記録されただけでは充分ではありません。何故なら、AIMがシステムダウンした場合、AIMが記憶している主記憶内のデータも全て回復できないからです。

このようなログファイルへの書き出しは、ページバッファやデータベースバッファが満杯となったときのデータベース更新やトランザクション処理が同期点に達したときに行われます。実更新が発生するまでのデータベースの更新内容はログバッファに記録されています。もし、このログバッファが満杯になると、その内容がログファイルに書き出されます。

AIMシステムの場合、TLFファイルを使用しているか否かにより、この動作が全く違ってきます。TLFファイルがないシステムの場合、ログバッファ(HLFバッファ)の内容は必ずHLFファイルに書き出されます。しかし、HLFファイルへのログデータの書き出しを必要とする際に、全てのHLFバッファが満杯であるとは限りません。満杯状態のHLFバッファはただ単にHLFファイルに書き出されます。しかし、満杯状態でないHLFバッファの内容は一旦HLFファイルに書き出されますが、その後そのバッファが満杯となった時に再度同一のレコード位置に同一のバッファの内容が書き出されます。この様な処理を行うことにより、HLFファイルには満杯状態のHLFバッファの内容が記録されたようになります。

TLFファイルのあるシステムでは、HLFファイルには満杯状態のHLFバッファの内容が書き出されます。中途半端(満杯状態でない)なHLFバッファの内容はすべてTLFファイルに書き出されます。この様な制御を行うことにより、HLFファイルに対する無駄なアクセスを回避しています。

HLFファイルをアクセスする際のアクセス待ち時間が長い場合、HLFファイルへのアクセス回数とそのアクセスに必要な時間の両方を考慮しなければなりません。HLFファイルのアクセスに必要な時間が長い場合、HLFファイルを割り当てているディスクボリュームを調査する必要があります。HLFファイルへのアクセス回数は、準備されたHLFバッファの大きさ(HLFファイルのブロック長)とHLFバッファの数により決定されます。また、TLFファイルを使用していないAIMシステムでは、中途半端なHLFバッファの内容もHLFファイルに書き出されます。是非とも、TLFファイルの使用を検討して下さい。

## 【対応策】

- 常時、アクセス待ち時間が長い場合、HLFバッファの大きさ(HLFファイルのブロック長)を拡張するか、もしくはHLFバッファ数を増加させて下さい。
- TLFファイルを使用していない場合、TLFファイルの使用を検討して下さい。
- HLFファイルがTLFファイルやAIMデータベースと同一のディスクボリュームに割り当てられていないことを確認して下さい。
- 他の理由により、HLFファイルのアクセス時間が長くなっていないことを確認して下さい。

## HLF02x

## 【説明】

HLFファイルのサービス時間が長過ぎます。HLFファイルをより高速のディスクボリュームに移動させるか、もしくは共通域に準備されるHLFバッファを拡張して下さい。

## 【解説】

AIMもシステムダウンすることがあります。しかし、システムダウンが発生したからといっても、データベースが破壊され、その回復が図れないという事態は回避しなければなりません。このために、AIMは実際のデータベース更新を行うたびに更新前のデータと更新後のデータをログファイルに書き出します。この場合、これらのデータが主記憶内のログバッファに記録されただけでは充分ではありません。何故なら、AIMがシステムダウンした場合、AIMが記憶している主記憶内のデータも全て回復できないからです。

このようなログファイルへの書き出しは、ページバッファやデータベースバッファが満杯となったときのデータベース更新やトランザクション処理が同期点に達したときに行われます。実更新が発生するまでのデータベースの更新内容はログバッファに記録されています。もし、このログバッファが満杯になると、その内容がログファイルに書き出されます。

AIMシステムの場合、TLFファイルを使用しているか否かにより、この動作が全く違ってきます。TLFファイルがないシステムの場合、ログバッファ(HLFバッファ)の内容は必ずHLFファイルに書き出されます。しかし、HLFファイルへのログデータの書き出しを必要とする際に、全てのHLFバッファが満杯であるとは限りません。満杯状態のHLFバッファはただ単にHLFファイルに書き出されます。しかし、満杯状態でないHLFバッファの内容は一旦HLFファイルに書き出されますが、その後そのバッファが満杯となった時に再度同一のレコード位置に同一のバッファの内容が書き出されます。この様な処理を行うことにより、HLFファイルには満杯状態のHLFバッファの内容が記録されたようになります。

TLFファイルのあるシステムでは、HLFファイルには満杯状態のHLFバッファの内容が書き出されます。中途半端(満杯状態でない)なHLFバッファの内容はすべてTLFファイルに書き出されます。この様な制御を行うことにより、HLFファイルに対する無駄なアクセスを回避しています。

HLFファイルへのアクセスはTLFファイルやデータベースと同時に発生します。このため、HLFファイルやこれらのファイルが同一のディスクボリュームに割り当てられていれば、HLFファイルのサービス時間が長くなります。また、通常、処理トランザクションの多いAIMシステムでは、ページングが多くなっています。ページング・データセットと同一のディスクボリュームにHLFファイルが割り当てられている場合にも、問題が発生します。

## 【対応策】

- HLFファイルがTLFファイルやAIMデータベースと同一のディスクボリュームに割り当てられていないことを確認して下さい。
- 他の理由により、HLFファイルのアクセス時間が長くなっていないことを確認して下さい。
- 常時、アクセス待ち時間が長い場合、HLFバッファの大きさ(HLFファイルのブロック長)を拡張するか、もしくはHLFバッファ数を増加させて下さい。
- TLFファイルを使用していない場合、TLFファイルの使用を検討して下さい。

**LOG01x****【説明】**

アプリケーションプログラムがデータベースの更新処理を行った際のデータを、ログバッファに転送するために使用するLRQBの枯渇が発生しています。LOGコマンドのLOGエントリーで定義されるLRQB数を増加して下さい。

**【解説】**

アプリケーションプログラムがデータベースを更新する度に、更新前のデータや更新後のデータをログファイルに出力しなければなりません。また、トランザクションが終了した際、そのトランザクション処理のために使用したリソース量などを、課金処理のためにログファイルに書き出しておく必要もあります。このようなログデータは一ヶ所(ログバッファ)に集められてから、ログファイルに書き出されます。AIMシステムの中の複数ヶ所で作成されるログデータをログバッファに転送する際には、各々のプログラムが勝手にデータ転送を行うと、ログファイルの信頼性が保証できません。このため、ログデータをログバッファに転送する際には、複数のプログラムが同時にデータ転送を行わないように制御しなければなりません。

AIMシステムでは、このログデータの転送制御のためにLRQBと呼ばれる制御表を使用しています。LRQBはログデータをログバッファに転送するプログラムが使用する制御表であり、この制御表を基にしてログデータの転送を逐次化しています。もし、このLRQBが確保できない際には、ログデータをログバッファに転送することは許されません。アプリケーションプログラムがデータベースの更新などを行う場合、データベースレコードの内容をデータベースバッファで更新すると同時に、このLRQBを確保し更新前と更新後のデータをログバッファに転送します。

AIMシステムがダウンしたような際に、データベースが破壊されたままにならないよう、アプリケーションプログラムはデータベースの更新を行う際には、必ず更新前のデータと更新後のデータをログファイルに書き出さなくてはなりません。このため、LRQBが確保できずログデータのログバッファに転送できない場合、データベースの更新要求は待たされます。つまり、LRQBが確保され、ログデータがログバッファに転送できるまで、データベース更新要求は完了しません。

LRQBの枯渇が発生すると、前述の様な理由により、データベースの更新要求などが短時間で完了しなくなります。つまり、AIMシステムのトータルスループットが低下します。このため、LRQBは充分な数だけ確保し、決して枯渇などが発生しないようにしなければなりません。

**【対応策】**

- PDLレポートで、枯渇が発生しているシステムを調査して下さい。
- 対応するシステムの、LRQBの数を増加させて下さい。LRQB数は、LOGコマンドのLOGエントリーで指定されます。

## LOG02x

## 【説明】

HLFバッファの枯渇が発生しています。LOGコマンドで指定されるHLFバッファの数やHLFバッファの大きさ(HLFファイルのブロック長)を拡張して下さい。

## 【解説】

アプリケーションプログラムがデータベースを更新する度に、更新前のデータや更新後のデータをログファイルに出力しなければなりません。また、トランザクションが終了した際、そのトランザクション処理のために使用したリソース量などを、課金処理のためにログファイルに書き出しておく必要もあります。このようなログデータは一ヶ所(ログバッファ)に集められてから、ログファイルに書き出されます。AIMシステムの中の複数ヶ所で作成されるログデータを効果的に記録するためログバッファ(HLFバッファ)は共通域に確保されます。ログデータをログファイルに記録するプログラムは、ログデータをHLFバッファに転送します。HLFバッファに転送されたログデータは必要に応じてHLFファイルもしくはTLFファイルに書き出されます。

AIMシステムでは、このログデータをHLFバッファに転送するためにLRQBと呼ばれる制御表を使用しています。LRQBはログデータをログバッファに転送するプログラムが使用する制御表であり、この制御表を基にしてログデータの転送を逐次化しています。もし、このLRQBが確保できない際には、ログデータをHLFバッファに転送することは許されません。また、ログデータを転送すべきHLFバッファに空きのエントリーがない場合にもログデータを転送することはできません。アプリケーションプログラムがデータベースの更新などを行う場合、データベースレコードの内容をデータベースバッファで更新すると同時に、このLRQBと空きのHLFバッファのエントリーを確保し更新前と更新後のデータをログバッファに転送します。

AIMシステムがダウンしたような際に、データベースが破壊されたままにならないよう、アプリケーションプログラムはデータベースの更新を行う際には、必ず更新前のデータと更新後のデータをログファイルに書き出さなくてはなりません。このため、LRQBやHLFバッファの空きエントリーが確保できずログデータのログバッファに転送できない場合、データベースの更新要求は待たされます。つまり、LRQBとHLFバッファの空きエントリーが確保され、ログデータがログバッファに転送できるまで、データベース更新要求は完了しません。

HLFバッファの枯渇が発生すると、前述の様な理由により、データベースの更新要求などが短時間で完了しなくなります。つまり、AIMシステムのトータルスループットが低下します。このため、HLFバッファは十分な数だけ確保し、決して枯渇などが発生しないようにしなければなりません。

## 【対応策】

- PDLレポートで、HLFバッファの枯渇が発生しているシステムを調査して下さい。
- 対応するシステムの、LOGコマンドで指定されたHLFバッファ数を増加させて下さい。
- HLFファイルのブロック長を再度検討して下さい。HLFファイルのブロック長がHLFバッファの大きさを決定します。
- HLFファイルの高速化を図って下さい。HLFファイルは、他のログファイルやデータベースと同一のディスクボリュームに割り当てられていないことを確認して下さい。
- HLFファイルのアクセス時間が長い場合、TLFファイルの使用を検討して下さい。

## RESP01x

## 【説明】

AIMシステム全体のトランザクションの応答時間とシステム内のチューニング監視指標との間に顕著な相関関係が確認されました。応答時間の安定化を図るため、対応するチューニングを行って下さい。

## 【解説】

AIMシステムなどのオンラインシステムでのチューニング項目には、数多くのものがあります。それらのチューニング項目の内、何れの項目が自社のオンラインシステムに有効であるかを判定するのは容易ではありません。また、評価する時間帯や日によっても、その評価結果は変わってきます。この様な問題を容易に解決するためには、オンラインシステム全体の平均応答時間とチューニング項目の必要性を示す指標群の値の相関判定を行います。つまり、平均応答時間の変動状況と多くの指標群の変動状況を判定するわけです。

もし、平均応答時間が長くなったときにある指標の値が必ず大きくなっていけば、その指標に関するチューニング項目がもっとも有効であるといえます。また、指標によっては、平均応答時間が長くなったときにその指標の値が小さくなるものもあります。いずれにしろ、平均応答時間との相関関係がもっとも高い指標のチューニングを実施するべきです。

平均応答時間と相関判定を行う指標の選択には、AIMシステムの知識やパフォーマンス管理に対する理解が必要です。つまり、むやみやたらに相関判定を行えば良いと言うものではありません。現在のリリースでは、次のような指標と平均応答時間の相関関係を判定しています。

- |                |                |
|----------------|----------------|
| ■ ページ・イン回数     | ■ LRQB枯渇回数     |
| ■ 処理トランザクション数  | ■ HLFバッファ枯渇回数  |
| ■ DCMSバッファ枯渇回数 | ■ HLFファイルの応答時間 |
| ■ DBの排他待ち回数    | ■ BOFファイルの応答時間 |
| ■ DBのデッドロック回数  | ■ TLFファイルの応答時間 |

平均応答時間と前述の指標群の相関判定を行う際には、どのような数学的判定を行うかということも重要になります。たとえば、同じ相関判定を行うにしても、単回帰や重回帰など、各種の方法があります。数多くのオンラインシステムの評価を行うことにより、重回帰などの複雑な相関判定であればあるほど、その相関判定の結果を評価することが困難になることが明白となっています。このため、このタイプの相関判定では、単回帰を利用します。

このチューニング・ヒントを出力する際には、前述の指標群ごとの相関係数が表示されます。この係数は、0.7以上であれば平均応答時間との間に強い相関関係がある事を示しています。このために、このチューニング・ヒントで複数の指標が同時に表示される場合は、必ずもっとも大きな相関関係が表示された指標に関するチューニングを実施して下さい。2番目や3番目に表示された指標に関するチューニングに専念しても無意味となる場合が多いですので注意して下さい。

## 【対応策】

- 相関係数のもっとも大きな指標に関するチューニングを行って下さい。
- 2番目や3番目に表示された指標に関するチューニングは、最初に表示された指標のチューニングが完了してから実施して下さい。
- 評価対象の日や時間帯に応じて、相関係数が変化する場合もあります。このため、その傾向を判定しながらチューニング項目を決定して下さい。



## TLF01x

## 【説明】

TLFファイルをアクセスする際のアクセス待ち時間が長過ぎます。TLFファイルのブロック長を拡張するか、もしくはTLFファイルをより高速化して下さい。

## 【解説】

AIMもシステムダウンすることがあります。しかし、システムダウンが発生したからといっても、データベースが破壊され、その回復が図れないという事態は回避しなければなりません。このために、AIMは実際のデータベース更新を行うたびに更新前のデータと更新後のデータをログファイルに書き出します。この場合、これらのデータが主記憶内のログバッファに記録されただけでは充分ではありません。何故なら、AIMがシステムダウンした場合、AIMが記憶している主記憶内のデータも全て回復できないからです。

このようなログファイルへの書き出しは、ページバッファやデータベースバッファが満杯となったときのデータベース更新やトランザクション処理が同期点に達したときに行われます。実更新が発生するまでのデータベースの更新内容はログバッファに記録されています。もし、このログバッファが満杯になると、その内容がログファイルに書き出されます。

AIMシステムの場合、TLFファイルを使用しているか否かにより、この動作が全く違ってきます。TLFファイルがないシステムの場合、ログバッファ(HLFバッファ)の内容は必ずHLFファイルに書き出されます。しかし、HLFファイルへのログデータの書き出しを必要とする際に、全てのHLFバッファが満杯であるとは限りません。満杯状態のHLFバッファはただ単にHLFファイルに書き出されます。しかし、満杯状態でないHLFバッファの内容は一旦HLFファイルに書き出されますが、その後そのバッファが満杯となった時に再度同一のレコード位置に同一のバッファの内容が書き出されます。この様な処理を行うことにより、HLFファイルには満杯状態のHLFバッファの内容が記録されたようになります。

TLFファイルのあるシステムでは、HLFファイルには満杯状態のHLFバッファの内容が書き出されます。中途半端(満杯状態でない)なHLFバッファの内容はすべてTLFファイルに書き出されます。この様な制御を行うことにより、HLFファイルに対する無駄なアクセスを回避しています。

TLFファイルには、特別のバッファは用意されていません。TLFファイルに書き出されるログデータはHLFバッファに格納されているものであるため、特別のバッファは必要ありません。このため、TLFファイルのアクセス待ち時間が長い際には、TLFファイルのブロック長を更に大きくするか、もしくはTLFファイル自体のスピードを速くする以外に方法がありません。しかし、TLFファイルのアクセス待ちは何れかの方法でなくするべきです。

## 【対応策】

- TLFファイルのブロック長を拡張して下さい。この際、TLFファイルのブロック長はHLFファイルのブロック長の正数分の1でなければならないことに注意して下さい。

## TLF02x

## 【説明】

TLFファイルのサービス時間が長過ぎます。TLFファイルをより高速のディスクボリュームに移動させるか、もしくはTLFファイルのブロック長を拡張して下さい。

## 【解説】

AIMもシステムダウンすることがあります。しかし、システムダウンが発生したからといっても、データベースが破壊され、その回復が図れないという事態は回避しなければなりません。このために、AIMは実際のデータベース更新を行うたびに更新前のデータと更新後のデータをログファイルに書き出します。この場合、これらのデータが主記憶内のログバッファに記録されただけでは充分ではありません。何故なら、AIMがシステムダウンした場合、AIMが記憶している主記憶内のデータも全て回復できないからです。

このようなログファイルへの書き出しは、ページバッファやデータベースバッファが満杯となったときのデータベース更新やトランザクション処理が同期点に達したときに行われます。実更新が発生するまでのデータベースの更新内容はログバッファに記録されています。もし、このログバッファが満杯になると、その内容がログファイルに書き出されます。

AIMシステムの場合、TLFファイルを使用しているか否かにより、この動作が全く違ってきます。TLFファイルがないシステムの場合、ログバッファ(HLFバッファ)の内容は必ずHLFファイルに書き出されます。しかし、HLFファイルへのログデータの書き出しを必要とする際に、全てのHLFバッファが満杯であるとは限りません。満杯状態のHLFバッファはただ単にHLFファイルに書き出されます。しかし、満杯状態でないHLFバッファの内容は一旦HLFファイルに書き出されますが、その後そのバッファが満杯となった時に再度同一のレコード位置に同一のバッファの内容が書き出されます。この様な処理を行うことにより、HLFファイルには満杯状態のHLFバッファの内容が記録されたようになります。

TLFファイルのあるシステムでは、HLFファイルには満杯状態のHLFバッファの内容が書き出されます。中途半端(満杯状態でない)なHLFバッファの内容はすべてTLFファイルに書き出されます。この様な制御を行うことにより、HLFファイルに対する無駄なアクセスを回避しています。

TLFファイルは、そのサービス時間を短くするために、特殊な形式を採用しています。TLFファイルとして使用されるデータセットの各トラックには複数のレコードがあります。しかし、それらのレコードの番号(主記憶のアドレスに相当します)は、全て同じ値になっています。AIMはTLFファイルにログデータを書き込む際に、一つのトラックには一つのレコードのみを書き込みます。つまり、シークした後、最初に検出したレコードに目的のデータを書き込んでいます。この様な特殊制御を行うことにより、ディスク装置特有の回転待ちやRPSミスなどの時間をなくしています。この結果、TLFファイルのアクセスの高速化は保証されますが、n個のTLFブロックが必要なシステムでは、n個のTLFトラックが必要となります。

## 【対応策】

- TLFファイルが他のログファイルやAIMデータベースと同一のディスクボリュームに割り当てられていないことを確認して下さい。
- 他の理由により、TLFファイルのアクセス時間が長くなっていないことを確認して下さい。
- TLFファイルのブロック長を拡張して下さい。この際、TLFファイルのブロック長はHLFファイルのブロック長の正数分の1でなければならないことに注意して下さい。