

ES/1 NEO

MFシリーズ

MF-ADVISOR
パフォーマンス・チューニング作業

第16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
OSP01n	2
OSP02n	4
OSP03n	6
OSP04n	8
PROC01n	9
PROC02n	11
PROC03n, PROC05n & PROC06n	12
PROC04n (MVS のみ)	13
PROC07n & PROC08n	14
PROC091	16
PROC10n	17
PROC11n	18
STOR01n & STOR02n	20
STOR03n & STOR04n	21
STOR05n	22
IOSS01n	23
IOSS02n	25
IOSS03n	26
IOSS04n	28
IOSS05n	29
IOSS06n	30
IOSS07n	32
IOSS08n	33
IOSS10n	34
IOSS11n	35
IOSS15n	36
PAGE01n & PAGE02n	37
PAGE03n	38
PAGE04n & PAGE05n	39
PAGE06n	40
PAGE07n	41
SWAP01n & SWAP02n	42
SWAP03n	43
SWAP10n	44
SWAP11n	45
SWAP12n	46
SWAP13n (MVS のみ)	47
SWAP14n (MVS のみ)	48
SWAP15n (MVS のみ)	49

VS01n & VS02n	50
VS03n & VS04n	51
VTAM01n	52
VTAM02n	53
WKLD01n	54
WKLD02n	56
COUPLING01n	57
COUPLING02n	58
COUPLING03n	59
PRD10n	60
PRD11n	61
PRD201	63
PRD301	64
PRD401	65
NOTE01n	66
第2章 評価手法の紹介.....	67
プロセッサ捕捉率.....	68
システム・スラッシング.....	68
オンライン専用システムでのシステム負荷指標	69
ワーキング・セット	70
ページングとワーキング・セット・サイズ	70
最小ワーキング・セット・サイズ.....	71
主記憶フレームの状態	71
ページ不在割り込みとページング.....	72
フレームの未使用時間	72
拡張記憶の動作.....	73
アクセス待ち時間.....	74
アクセス・パス待ち時間とデバイス待ち時間	74
ディスク装置の動作	75
入出力サブシステムの負荷バランス.....	75
第3章 パフォーマンス評価データ算出式	76
実 CPU 時間	76
計測 CPU 時間	76
プロセッサ捕捉率(CAPTURE RATIO).....	76
アクティブ・フレーム	76
アクティブ・フレーム率	76
マルチ・プログラミング・レベル(MPL).....	77
ワーキング・セット・サイズ(WSS)	77
スワップ・アウト率	77
スワップ・アウト時間	77
レジデント時間(スワップ・イン時間)	77
CPU バースト時間	78

入出力動作回数	78
入出力待ち時間.....	78
装置のビジー率	78
装置のサービス時間	78
装置のアクセス待ち時間	78
装置のレスポンス時間	79

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

MF-ADVISORでは各種のチューニング・ヒントを出力します。しかし、MF-ADVISOR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、MF-ADVISOR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-ADVISOR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。

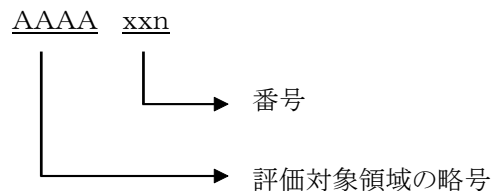
MF-ADVISORでは、評価対象のオペレーティング・システムを下記の領域に分割し、それぞれの評価結果をチューニング・ヒントで表示します。

■ 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
■ 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
■ 結合機構COUPLING
■ 相関傾向PRD
■ 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。

MF-ADVISOR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>

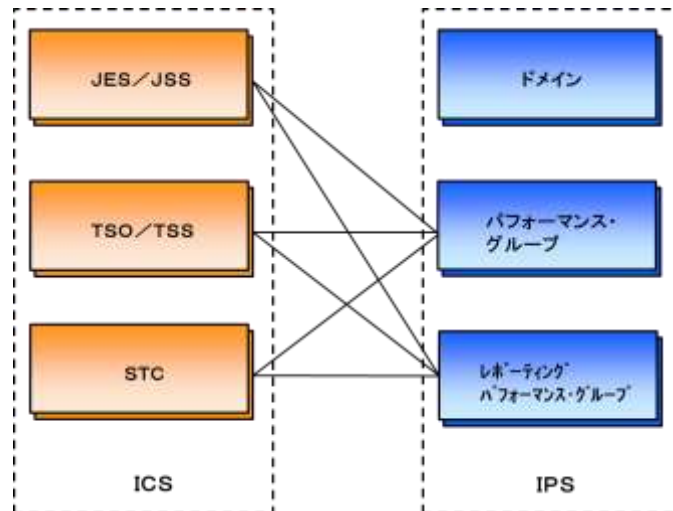


OSP01n

【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パラメータがメーカーが提供する初期システムのデフォルトのままであると思われます。自社の運用形態に合わせた形式で設定し直して下さい。

【解説】



オペレーティング・システムは、資源管理プログラムのパラメータ群が初期システムのままであっても動作します。しかし、そのシステムが単一業務システムでない限り、IPSパラメータなどは、自社の業務運用の形態に合わせてカスタマイズする必要があります。

ここでいう単一業務システムとは、24時間中オンラインのみを行うようなシステムを意味します。

資源管理プログラムのパラメータ群を自社用にカスタマイズされる際には、次の作業が必要です。

運用中の業務の分類

- オンライン、TSO/TSS、バッチなどのサブシステムによる分類
- 応答時間や処理経過時間などの制約による分類
- 業務プログラムの性格 (CPU バウンドなど) による分類

パフォーマンス・グループの設定

- 分類された業務グループ毎へのパフォーマンス・グループの割り当て
- 使用サービス量に応じたパフォーマンス・ペリオッドを設定
- プロセッサのディスパッチング・プライオリティの決定
- ストレージ分離機能の使用の有無の判定
- 入出力プライオリティの決定

業務プログラム毎のプログラム多重度の設定

- 類似した業務プログラムをグループ化し、ドメインを設定
- ドメイン毎の最小と最大のプログラム多重度を決定

ICSパラメータでは、業務プログラムのジョブ名やユーザIDおよびトランザクション名などにより、業務プログラムとパフォーマンス・グループを対応付けます。また、IPSパラメータでは、プロセッサやストレージおよび入出力サブシステムのアクセス優先度やプログラム多重度などを設定します。資源管理プログラムの制御定数の省略値を設定しているOPTパラメータは、特殊な事情がない限り変更しないで下さい。

<次頁へ続く>

以上の作業を実施することにより、最適なシステム運用と効果的なパフォーマンス管理が実施できるようになります。文中のPS、ICS、OPTパラメータはSYS1.PARMLIBのメンバーを意味します。また、関係するメンバーを下記に示します。

	IBMシステム	富士通システム	日立システム
SYSパラメータ	IEASYSxx	KAASYSxx	JAASYSxx
ICSパラメータ	IEAICSxx	KAAICSxx	-----
IPSパラメータ	IEAIPSxx	KAAIPSxx	JAAIPSxx
OPTパラメータ	IEAOPTxx	KAAOPTxx	JAAOPTxx
APGパラメータ	-----	KAAAPGxx	-----
IOPパラメータ	-----	-----	JJA IOPxx

【対応策】

■弊社へご連絡いただくか、下記のメーカ提供のマニュアルをご参照下さい。

IBMシステム

MVS/XA Initialization and Tuning

MVS/XA System Management Facilities

富士通システム

OSIV-F4 MSP 制御プログラム説明書

OSIV-F4 MSP システムパラメタ説明書

OSIV-F4 MSP SMF 説明書

VSP 運用手引書

日立システム

VOS3/ES センタ運営 - JSS3 編 -

VOS3/ES システム管理支援

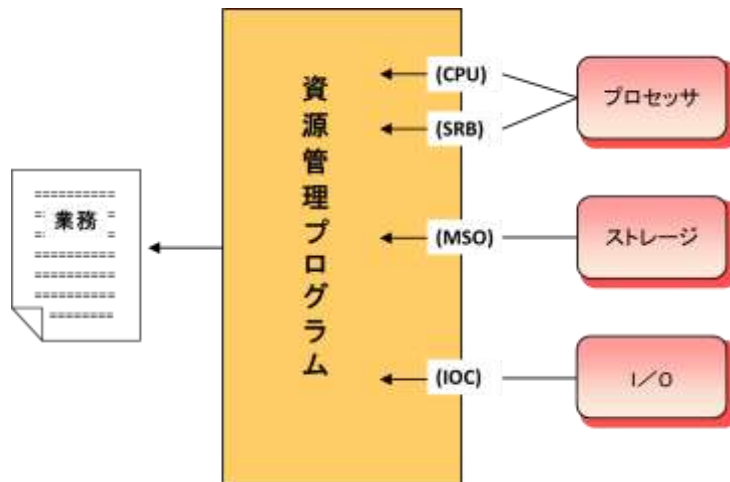
VOS3/ES モニタリング支援

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。(日立システムにはSRBリソース係数はありません。)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。下記に各メーカーが提供する省略値を示します。

<次頁へ続く>

	CPU	SRB	MSO	IOC
IBMシステム	10	10	3	5
富士通システム	3	3	0.1	1
日立システム	5.5		0	5.5

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】

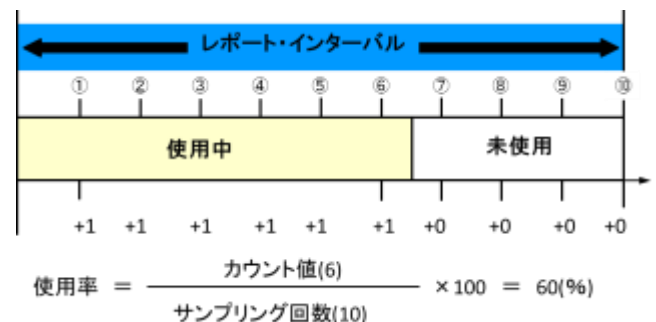
使用中のソフトウェア・モニタのインターバルが15分から30分以上に設定されています。評価精度を向上させるためには、インターバルを15分から30分に設定して下さい。

【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やPDLおよびSAR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とSARの注意点

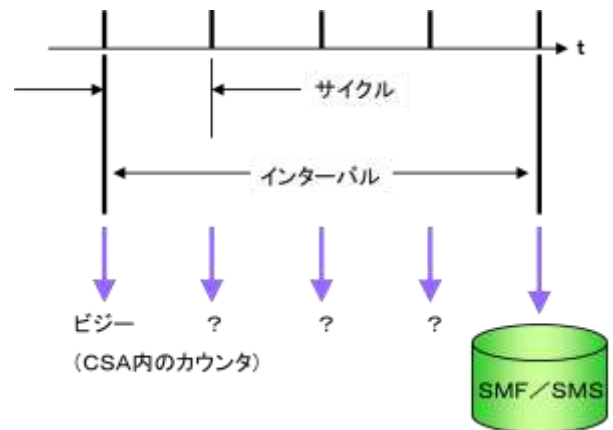
RMFとSARの場合、メーカー提供のパラメータで動作させる事により問題が発生することは少ないようですが、データの信頼性を考察し、CYCLEとINTERVALの関係は次のようにする必要があります。

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

∴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。

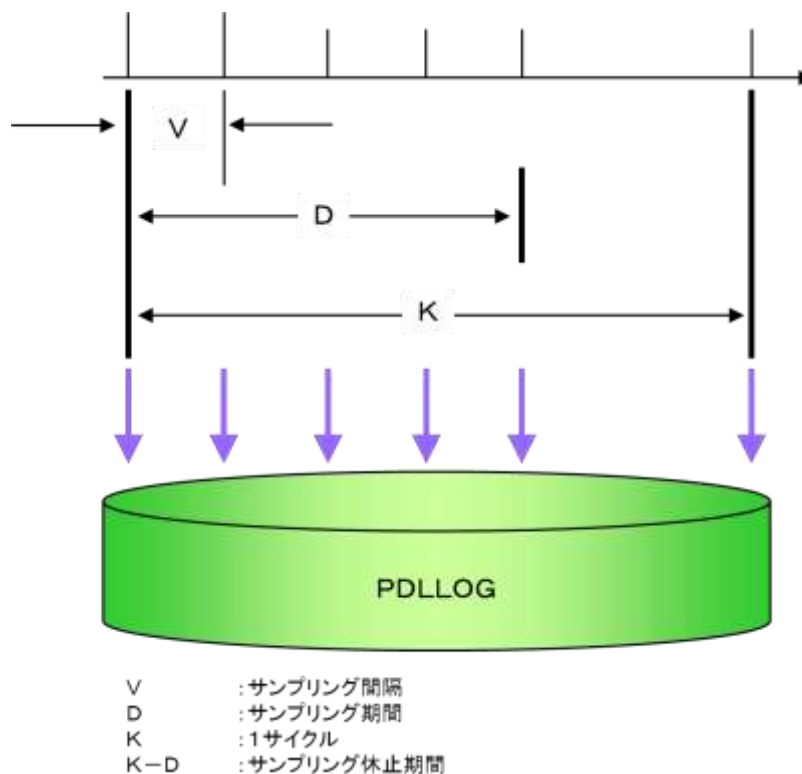
INTERVALとは、サンプラが更新するサンプリング(CSA等に記憶)を基に使用率などを算出し、SMF/SMSファイルなどに書き出す時間間隔です。



■ 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。



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(主記憶や拡張記憶もしくはシステム記憶)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、PDL、SAR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。

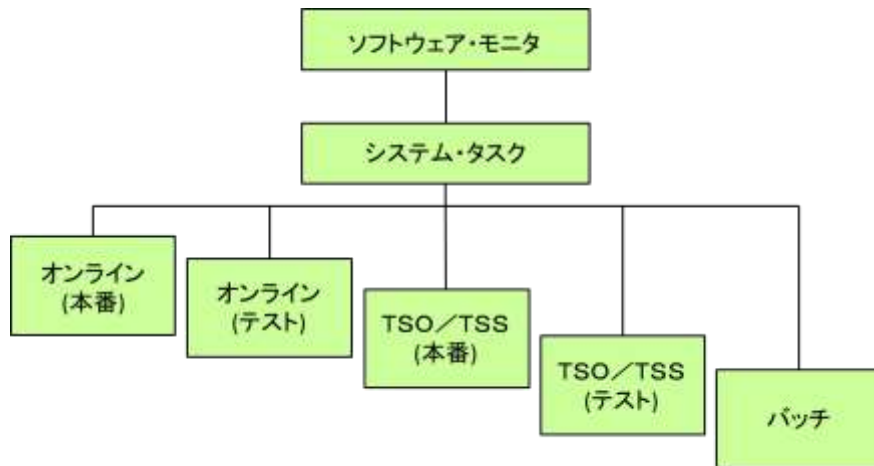
PROC01n

【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。

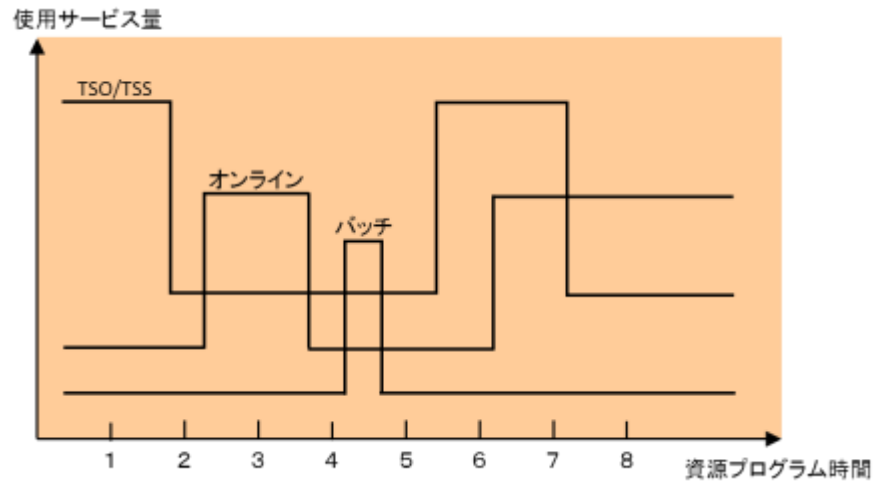


ディスパッチング・プライオリティは、運用中の業務プログラム・グループのバランスを考察して設定します。ここでは、一般的なガイド・ラインを示します。

- (注) IBM システムには、ローテーションがありません。
- 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 (MEAN TIME TO WAIT) およびローテーションの3種類があります。固定はシステム・タスク用に、ローテーションは科学技術計算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, PDL, SAR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/JSS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/TSS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 > TSO/TSS > 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。

< 次頁へ続く >



【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。
- 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。
- RMFのモニタⅡやSAR/D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, PDLは富士通, SAR/Dは日立システムのソフトウェア・モニタです。また、TSAやPPEはブール& パベージ社の商品名で、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】

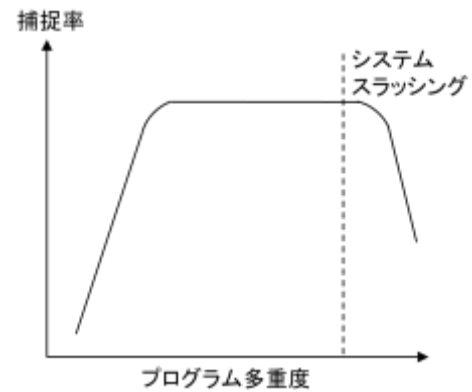
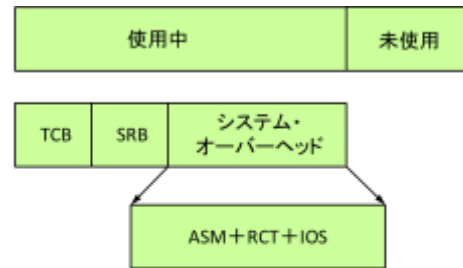
オペレーティング・システムにおけるプロセッサの使用率の内訳は右図のようになっています。ソフトウェア・モニタがプロセッサ使用率として求めているのが、プロセッサが使用中であった時間の割合です。また、各パフォーマンス・グループで実行される業務プログラムごとに計算されているCPUやSRBのサービス・ユニット値により、プロセッサがTCBやSRBモードで使用されていた時間も求めることができます。この2つの時間の差がシステム・オーバーヘッドです。

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。



PROC03n, PROC05n & PROC06n

【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の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。

【対応策】

- このシステムで実行されている業務のプロセッサ使用率を確認してください。
必要に応じて、業務の実行時間帯の変更や多重度を見直してください。
- 共有区画の重み値の見直しを検討してください。

PROC04n (MVS のみ)

【説明】

主記憶と拡張記憶の間におけるページ転送(ページムーブ)の数が増加したため、その処理によるシステムオーバーヘッドが急激に増加しています。注意深く、ページ転送量を監視して下さい。

【解説】

拡張記憶は主記憶と違い、システム内における高速な記憶装置として位置づけられています。従来は、この拡張記憶に高速ページデータセットのような役割を果たさせていました。最近ではページデータセットだけではなく、ハイパー空間の技法を利用したユーザデータセットの高速化を図るための記憶域としても使用されています。

拡張記憶は主記憶と同様にプロセッサに内蔵されたメモリーです。しかし、その使用効率を高めるために、拡張記憶ではハードウェア的またソフトウェア的にも主記憶とは違った制御方式が採用されています。また、プログラムが拡張記憶に記録されているデータを直接アクセスする方法は提供されていません。このため、プログラムが拡張記憶に記憶されたデータを使用するためには、一端そのデータを主記憶へ持ってくる必要があります。

主記憶は4KBのフレームを単位として管理されています。拡張記憶においてもその管理を容易にするために、主記憶と同様に4KBのフレームを単位として管理されています。拡張記憶に記憶されているデータをプログラムで使用する場合、そのデータが記憶されているフレームを検索し、主記憶の未使用フレームにコピーします。同じデータを主記憶と拡張記憶の両方に記録して置くと各種の矛盾が生ずるため、主記憶へデータが転送されると拡張記憶に記録されていたデータは削除されます。

OSはこの様な方法により、高速なアクセスが可能な拡張記憶を主記憶の一部として使用します。しかし、ここで問題となるのが、主記憶と拡張記憶の間におけるデータ(もしくはページ)の移動に関するオーバーヘッドです。

もし、十分な容量の主記憶が手当されている場合、一度主記憶に読み込まれたページは、そのページが使用されなくなるまで主記憶に常駐します。その間、そのページが主記憶内で移動させられることは(特殊な例を除き)ありません。

しかし、主記憶の容量が充分でない場合、拡張記憶や外部記憶との間でデータの転送を行う必要があります。もし、外部記憶とのやり取りが多くなるようであればページングが頻発し、顕著なパフォーマンス低下を引き起こします。一方、主記憶と拡張記憶の合計容量がプログラムが必要としている実記憶の容量を充分カバーしている場合、外部記憶とのデータのやり取りをする必要はありません。しかし、十分な主記憶容量を確保している場合と違い、主記憶と拡張記憶間のデータの移動が頻繁に行われている状態となります。

主記憶と拡張記憶の間におけるデータの移動は、主記憶内でのデータの移動(ムーブ)と同じです。例え、このデータ転送が高速に行われたとしても頻発するようであれば、その制御のためのオーバーヘッドを無視することは出来ません。事実、1秒間に何千ページも主記憶と拡張記憶間で転送すると、その制御のために、多くのCPU時間(CPU使用率で5%以上)を費やしてしまうケースもあります。

この主記憶と拡張記憶間のデータ(ページ)転送を制御しているプログラムのCPU使用時間はパフォーマンスグループ0で報告されます。ES/1 NEOでは、このパフォーマンスグループ0のCPU使用率と、主記憶と拡張記憶間のページ転送回数の相関判定を行います。もし、その相関関係が顕著であるとされると、このチューニングヒントは作成されます。より詳細には、「拡張記憶ページ転送グラフ」を参照して、調査の基礎資料を作成して下さい。

【対応策】

- 主記憶が過負荷状態になっています。ドメインごとの最小MPL値を再度検討し、システム負荷を減少させて下さい。
- モニタIIのASD画面などで、拡張記憶からのページ転送が多い業務プログラムを探し、その業務プログラムの実行時間帯を再検討して下さい。
- 主記憶の増設を検討する必要があります。

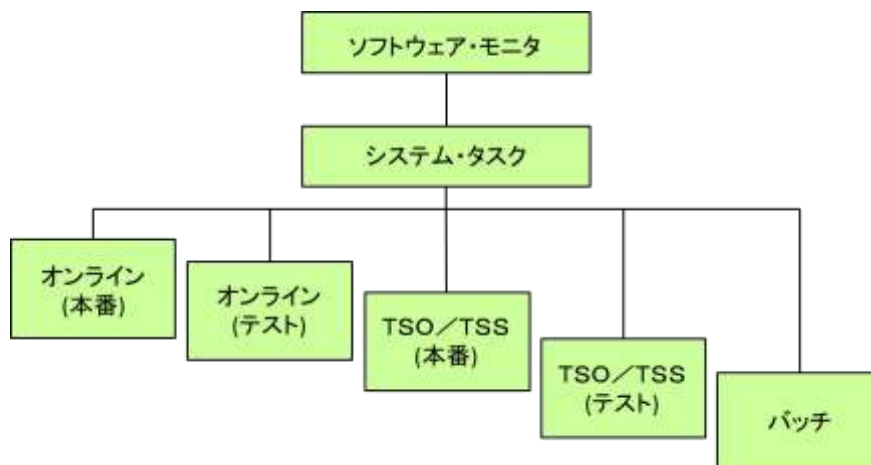
PROC07n & PROC08n

【説明】

特殊プロセッサ(zAAPまたはzIIP)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、特殊プロセッサ(zAAPまたはzIIP)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。

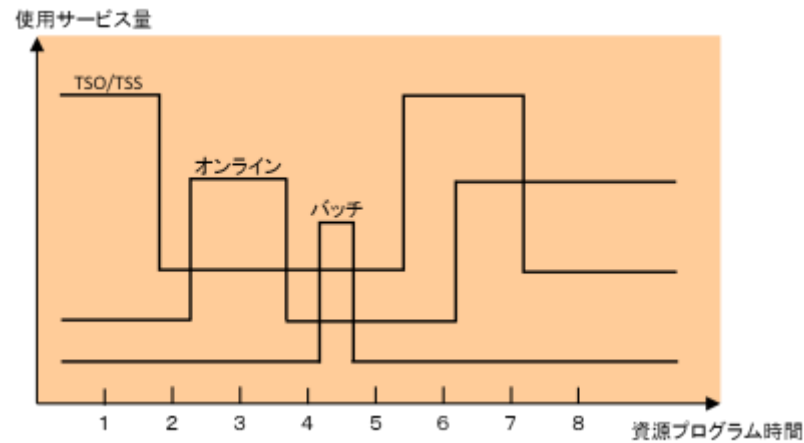


ディスパッチング・プライオリティは、運用中の業務プログラム・グループのバランスを考察して設定します。ここでは、一般的なガイド・ラインを示します。

- 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(MEAN TIME TO WAIT)の2種類があります。固定はシステム・タスク用に用意されています。一般的な業務プログラムのディスパッチング・プライオリティには自動制御を指定して下さい。
- ② ソフトウェア・モニタ(RMF)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。

<次頁へ続く>



【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やSAR/D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。

PROC091

【説明】

運用しているシステムのプロセッサ数が動的に変更されています。そのため、プロセッサ使用率を単純に比較することができません。

【解説】

プロセッサ使用率は、プロセッサ数に関係なく0～100%で出力されます。これは、プロセッサ側でプロセッサ数を考慮して算出している為です。これにより、プロセッサ使用率が70%であるとき、30%の余力があると簡単に判断できます。

CoD (Capacity on Demand)機能では、システムが動的にプロセッサ数を増減することがあります。これは、パフォーマンス計測ツールの収集インターバルで異なるプロセッサ数が記録されることを意味します。

例えば、10分インターバルで収集している場合、最初のインターバルは1台、次のインターバルで1台追加した場合を想定します。追加したインターバルで5分だけ2台構成で運用された際には、次のように記録されます。

プロセッサ数	
最初のインターバル	1.0 台
追加したインターバル	1.5 台
その次のインターバル	2.0 台

現状では、各インターバルでオンラインであったプロセッサ数を基にプロセッサ使用率を算出します。その為、上記の例ではプロセッサ使用率の意味がインターバル毎に異なることになります。

【対応策】

- 実行するプロセッサでCODCTLSW=1を指定して、プロセッサを再実行してください。この場合、報告されるプロセッサ使用率は、そのシステムが使用可能な最大プロセッサ数を基に算出します。

PROC10n

【説明】

この共有区画はプロセッサ使用制限機能が指定されていない区画です。この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おり、筐体全体の通常プロセッサ(CP)のプロセッサ使用率も高い状態です。

【解説】

PR/SMハイパーバイザーによるプロセッサ使用権の制御は、タイムスライス方式を採用しています。専用区画は常時プロセッサが割当られます。共有区画はウェイトモード(WC:WAIT COMPLETION)の指定に従います。ウェイトモードは、論理区画で実行するシステムがウェイト状態になった際の制御を指示します。WC=YESは、システムがウェイト状態になっても規定時間はプロセッサ使用権を継続して与えます。一方、WC=NOはシステムがウェイト状態になった際に、他の論理区画にプロセッサ使用権が移動します。

論理区画のプロセッサ使用量は、重み値やプロセッサ使用制限機能の有無で制御されます。論理区画の重み値は、論理区画で使用可能なプロセッサ使用量を決定します。プロセッサ使用制限機能は、その使用可能量以上のプロセッサ使用を許可するか制限するかを指示します。CAP=YESは使用可能量以上のプロセッサ使用を制限します。これがハードキャッピングです。一方、CAP=NOは、全体のプロセッサ使用に余裕がある際には使用可能量を超えてプロセッサ使用を許可します。当然、全体のプロセッサ使用が過負荷状態になると、各論理区画の使用可能量で制限されることになります。これらのことから、WC=NOやCAP=NOの論理区画は、使用可能量以上のプロセッサ使用が許可されていることになります。

【対応策】

- このシステムで実行されている業務のプロセッサ使用率を確認してください。
必要に応じて、業務の実行時間帯の変更や多重度を見直してください。
- 共有区画の重み値の見直しを検討してください。
必要に応じてプロセッサ使用制限機能の有無を検討してください。

PROC11n

【説明】

共有区画にはキャパシティ設定値(DEFINED CAPACITY)を指定することでその論理区画の使用を制限することができます。この際、共有区画が実際に使用した4時間移動平均値(long-term rolling 4-hour average)とキャパシティ設定値が比較され、超えている場合はその使用を制限するためにキャッピングされます。

【解説】

z/OSでは、ソフトウェア使用料金のコスト削減を目的としたWLC(Workload License Charges)を提供しています。単一論理区画を対象とする場合や、複数論理区画を対象とする機能があります。これらの機能では、ソフトウェアが実行するプロセッサモデルに応じた料金設定が適用されます。この際には、MSU(Million Service Units)値が基本となります。各プロセッサモデル毎に、基本となる1時間当たりのMSU値(MSU/H)がメーカーによって決定されています。

単一論理区画での制限は、PR/SMの論理区画の定義項目でキャパシティ設定値(DEFINED CAPACITY)を指定することで実現されます。このキャパシティ設定値もMSU/Hの数値を指定します。

論理区画の定義でキャパシティ設定値が指定されている際には、この値を超えないように制御されます。この制御はWLM(ワークロードマネージャ)とPR/SMが連携して実施することから、ソフトキャッピング(soft capping)とも呼ばれます。これに対して、PR/SMの設定で重み値やプロセッサ使用制限機能を指定してプロセッサの使用を制限することを、ハードキャッピング(hard capping)と呼びます。

このソフトキャッピングの制御は、4時間移動平均値を算出して、キャパシティ設定値と比較します。この4時間移動平均値は、WLMが5分単位に計測して、48個に分割されています。システムがIPLされた時点の初期値は1で、5分経過する毎に更新され4時間移動平均値も再算出されます。これにより、一時的に使用量が急激に増大するスパイクも吸収されることになります。

論理区画に指定されたキャパシティ設定値を4時間移動平均値が超えた際には、WLMがPR/SMにキャッピングするように指示を出します。その後、キャパシティ設定値を下回った時点でキャッピングが解除されることになります。

WLMによるソフトキャッピング制御は、これらの機能を基に論理区画の重み値とキャパシティ設定値の関係から、次の3種類の制御を行います。

①重み値＝キャパシティ設定値

WLMは4時間移動平均値がキャパシティ設定値を超えた時点で、PR/SMにキャッピングを指示します。これは重み値を基に連続したキャッピングを継続することになるため、ハードキャッピングを意味します。

②重み値<キャパシティ設定値

この場合、重み値を基にした使用制限を実施すると、キャパシティ設定値まで使用することができなくなります。このため、ある時間帯は重み値で制限し、別の時間帯は重み値による制限を外すなどの非連続的なキャッピング制御を行うことで、結果としてキャパシティ設定値まで使用していることになります。

③重み値>キャパシティ設定値

対象論理区画をキャパシティ設定値までに使用制限する目的で、仮の論理区画(PHANTOM LPAR)を定義します。この際の仮の論理区画の重み値は、下記の算出式を使用します。

仮の重み値(PHANTOM WEIGHT)

= 全体キャパシティ ÷ キャパシティ設定値 × 対象区画の重み値 - 全区画の重み値の合計

パフォーマンス計測ツール(RMFなど)では、キャッピング状況を計測して報告します。このキャッピング状況にはWLMがキャッピングが必要と判断した状況と実際にキャッピング処理が実施されている状況があります。例えば、WLMキャッピング状況の100%は実際にキャッピングされている場合や、ソフトキャッピングで仮の重み値制御が実施されている場合が含まれます。一方の実際のキャッピング状況は実際にキャッピングを実施した割合になります。

【対応策】

- システムで実行している重要業務に影響がないか確認してください。
必要に応じて、重要業務以外の実行時間帯の変更や多重度を見直してください。
- キャパシティ設定値の見直しを検討してください。

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】

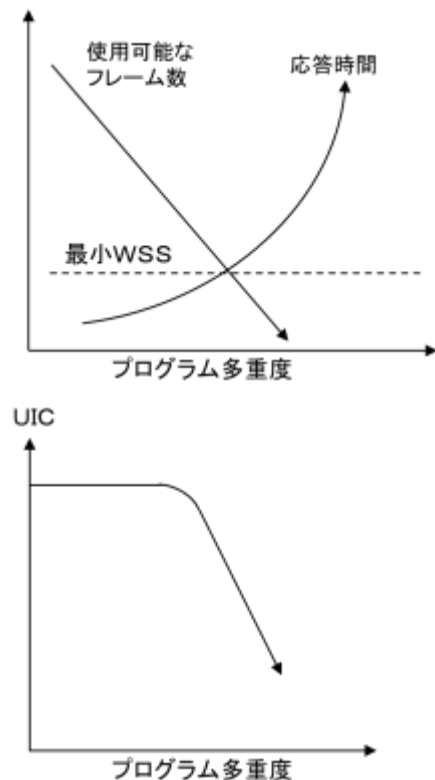
主記憶は複数の業務プログラムで共用使用されるリソースです。このため、プログラム多重度が増加するにつれ、主記憶の競合が激化し個々の業務プログラムが使用可能な主記憶フレーム数が業務プログラム毎の特性である最小ワーキング・セット・サイズ (WSS) 以下になると、ページングの影響が顕著に現れ応答時間が急激に悪化します。

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。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/TSS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。



STOR03n & STOR04n

【説明】

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

【解説】

IBMシステムにおいて、主記憶の使用率が高くなると、非参照時間の長い主記憶フレームを拡張記憶にページ・ムーブします。このため、主記憶の使用率に比例し、拡張記憶の使用率が高くなります。拡張記憶が満杯になると、今度は、拡張記憶から外部記憶へのマイグレーションが多くなります。このため各拡張記憶フレームのマイグレーション・エイジが急激に減少し、ページ・イン回数が指数的に増加します。このような現象を拡張記憶の過負荷状態といいます。

(拡張記憶の各フレームは、マイグレーションエイジ(非参照時間)を基にしたLRU方式で管理されています。)

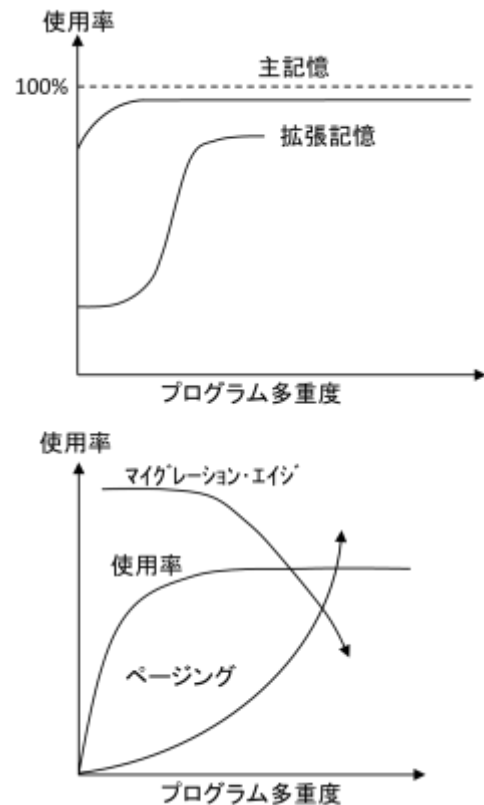
このため、重要な業務プログラムには、十分なワーキング・セット・サイズが保証されるように配慮し、応答時間の安定化を図って下さい。また、拡張記憶域の使用状況(用途)とサイズについても調査・検討して下さい。

重要な業務のワーキング・セット・サイズを保証するためには、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は、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/TSS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。
- 拡張記憶を増強する。この場合、主記憶容量と同等もしくは2倍以上の容量にされることをお勧めいたします。



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。

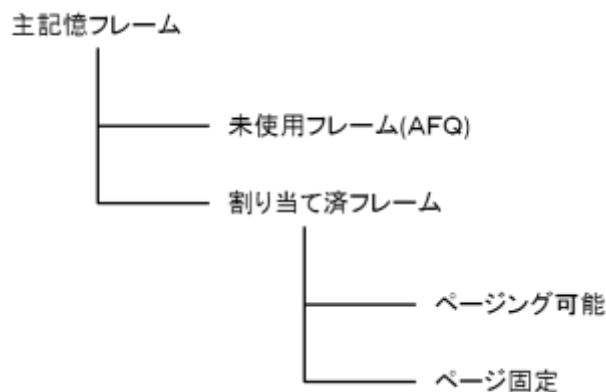
【解説】

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。



IOSS01n

【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】

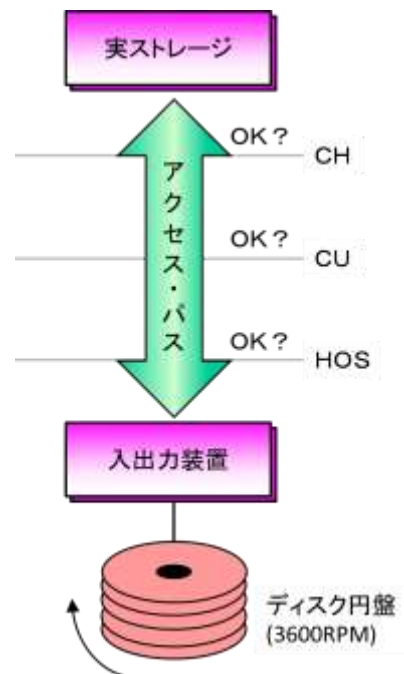
ディスク装置の応答時間は次のように分類できます。



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。

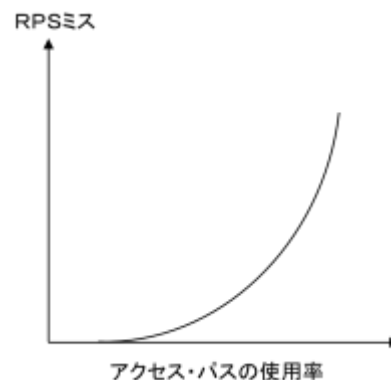


$$\begin{aligned} \text{RPSミス時間} &= RV \times (\text{1回ミスする確率}) + RV \times (\text{2回ミスする確率}) \\ &+ RV \times (\text{3回ミスする確率}) + \dots \end{aligned}$$

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

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

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、 ρ 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。



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ミス時間はチャンネル・パスの使用率より理論的に求めることができます。また、チャンネル・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）
I B M	3380	3600
	3390	4260
富士通	6425 (H以外)	3600
	6425H	4340
日立製作所	6586K	3600

【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。）
- DASD/ADVISORでRPSミス時間を監視する。



DASD/ADVISORはブル&バページ社の商品名でIBMシステムでのみ動作します。

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

(注)
回転数については
IOSS01nを
参照して下さい。

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。

I0SS03n

【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。

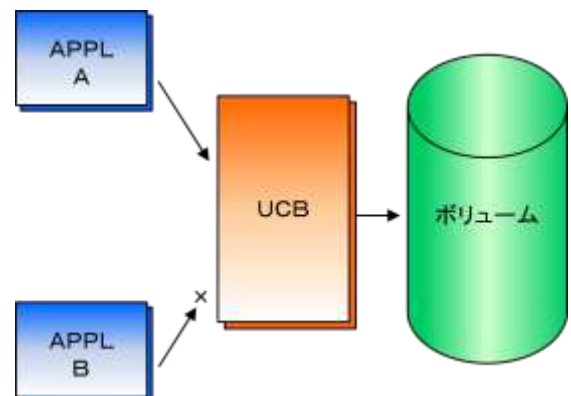
【解説】

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。

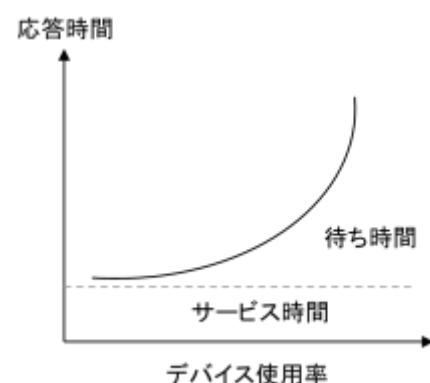


アクセス待ち時間とは、先行した入出力動作が完了するまで入出力装置の制御テーブルであるUCB(ユニット・コントロール・ブロック)で新たなアクセス要求が待たされる時間のことです。各ソフトウェア・モニタは、このアクセス待ち時間を各ディスク装置毎に時間 かもしくはキュー長でレポートします。複数の業務プログラムが同一ボリュームを全くランダムにアクセスする際の待ち時間は待ち行列技法のM/M/1の公式で求めることができます。

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



式中の ρ 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。



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

この根拠で、長い間、ディスク装置のデバイス使用率は30%程度に保つように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。

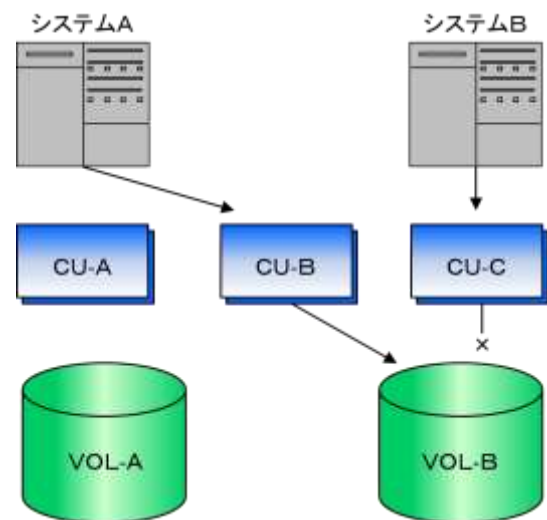
I0SS04n

【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。

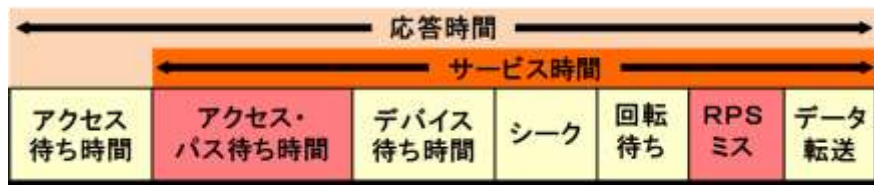
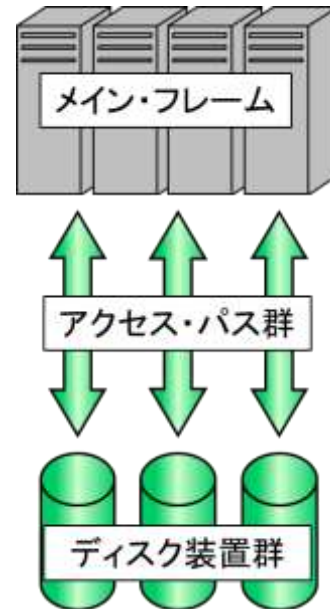
IOSS05n

【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

【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。

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>

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。
 - ・DSO(データセット・オブティマイザ)でシーク時間を監視する。
 - ・PDLのVOLUMEサンプラーでシーク時間を監視する。
 - ・RMFのモニタⅡやSAR/Dでプログラム単位のディスク・ボリューム競合状況を監視して下さい。



DSOはブル&バページ社の商品名でIBMシステムでのみ稼働します。
RMFはIBM, PDLは富士通, SAR/Dは日立システムのソフトウェア・モニタです。

【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。

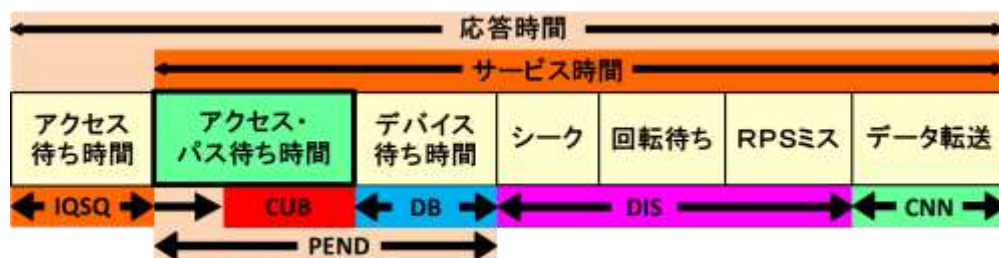
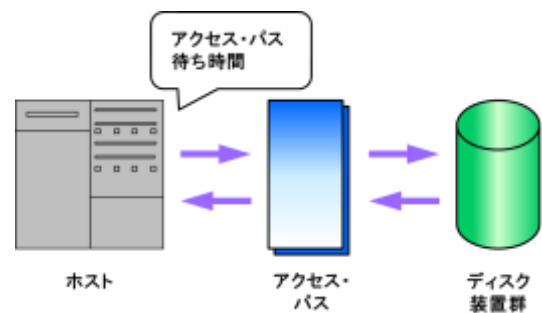
IOSS07n

【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、チャン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。
- チャンネルを増設する。

I0SS08n**【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。

I0SS10n

【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 (VM/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。

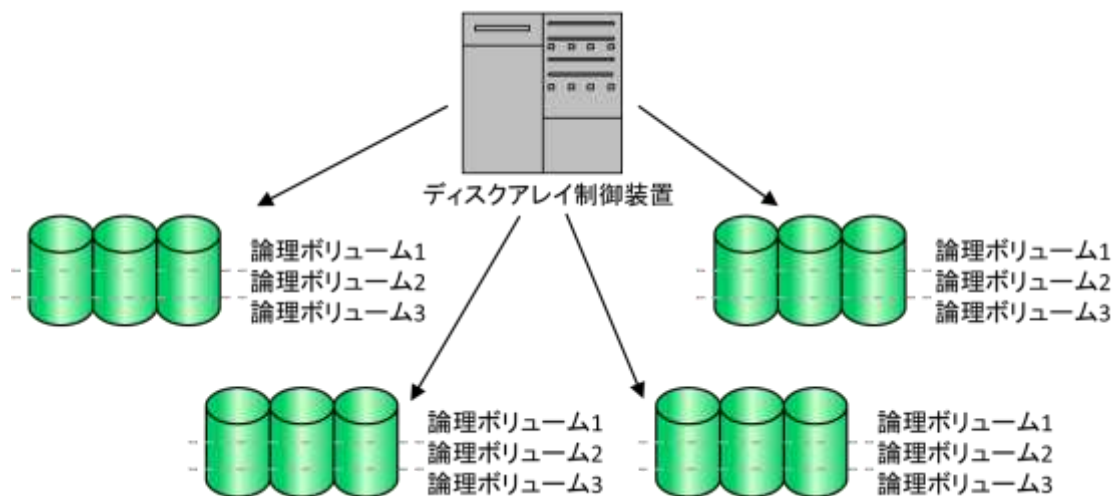
【説明】

ディスクアレイ装置でパリティグループ間の負荷が均等ではなく、大きな片寄りがあります。負荷を分散して均等化を図ってください。

【解説】

ディスクアレイ装置は、従来のディスクボリュームと異なり、複数の論理ボリューム群をパリティグループと呼ばれる複数のディスク装置群でエミュレートしています。このパリティグループ毎の論理ボリューム数は、ディスクアレイ装置のモデルにより異なります。論理ボリューム群はそのパリティグループを構成する全てのディスク装置群を共有します。オペレーティング・システムは論理ボリューム単位の逐次処理は行いますが、パリティグループ単位の逐次化は行いません。そのため、パリティグループ内の複数の論理ボリュームを同時にアクセスすることができますが、実際にアクセスするディスク装置群での競合により待ち状態が発生することがあります。

ディスクアレイ装置を管理するためには、パリティグループ単位や論理ボリューム単位での負荷バランスに注意する必要があります。



【対応策】

- 負荷の高い複数の論理ボリュームを同じパリティグループに配置しないようにしてください。
- 論理ボリュームのバランス化はデータセットの分散で対応してください。

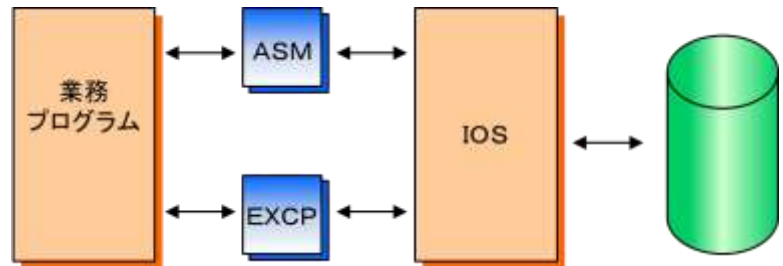
PAGE01n & PAGE02n

【説明】

ローカル・ページ・データセットが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。新たなローカル・ページ・データセット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ると言えます。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ローカル・ページ・データセットをサポートします。また、複数のローカル・ページ・データセットの内、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つのローカル・ページ・データセットを割り当てる必要があります。もし、複数のローカル・ページ・データセット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、新たなローカル・ページ・データセットを追加して下さい。

【対応策】

- 必ず複数のローカル・ページ・データセットを割り当てる。
- ローカル・ページ・データセットのアクセス・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】**

ローカル・ページ・データセット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。ローカル・ページ・データセットの総容量を増やして下さい。

【解説】

ローカル・ページ・データセットは、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。また、VIOを使用中のシステムでは、VIOデータセットもローカル・ページ・データセットに格納されます。このローカル・ページ・データセットの総容量は、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 VIOを使用中のシステムでは、VIOデータセットの大きさに注意しなければなりません。しかし、現在、VIOデータセットの大きさや数を効果的に制御する方法はありません。(ダイレクトVIOの機能の利用をお勧めします。)
- ② 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。
- ③ ローカル・ページ・データセット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】

- 新たなページ・データセットをローカル用に割り当てる。
- 事前にページ・データセットをフォーマットしておき、必要な時にPAGEADDコマンドでスペース追加を行える体制を検討する。

PAGE04n & PAGE05n

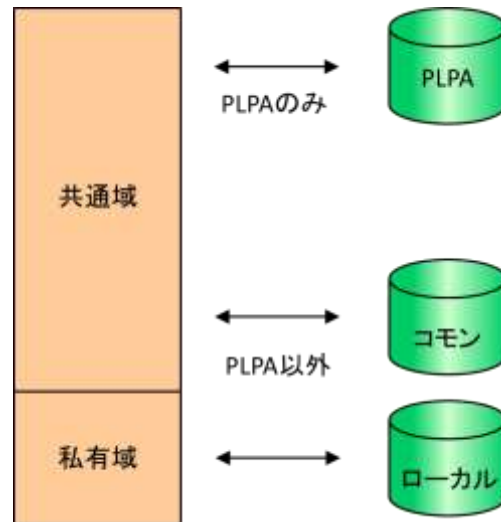
【説明】

PLPAとコモンのページ・データセット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、PLPAやコモンならびにローカル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、複数のローカル・ページ・データセットの内、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ます。しかし、共通域用のPLPAやコモンのページ・データセットは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、PLPAやコモンのページ・データセット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】

- PLPAやコモンのページ・データセット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(スワップ・データセットを含む)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】

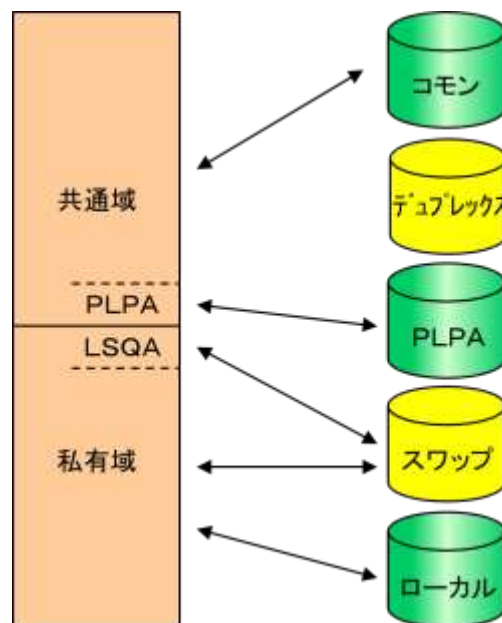
仮想記憶方式を使用するオペレーティング・システムでは、実記憶と仮想空間（アドレス・スペース）の差を埋め合わせるためのページング動作が必要となります。オペレーティング・システムでは、このページング処理を効果的に行うために、ページング対象となる仮想空間領域の特性に応じて5種類のデータセットを使用します。この5種類のデータセットは、共通域用のコモン、PLPA、と私有域用のローカル、スワップおよび二重化用のデュプレックスにグループ分けできます。私有域は共通域に比べ、その総容量が大きいため、複数のデータセットでローカルやスワップ・ページングを処理させることができます。

二重化用のデュプレックス・ページ・データセットは各オペレーティング・システムにより異なります。IBMシステムの場合は、コモンとPLPAについてデュプレックスを持つことができます。富士通システムの場合は、全ページ・データセットについてデュプレックスが持てます。日立システムの場合は、コモン、PLPAとローカルについてデュプレックスが持てます。また、スワップはスワップと拡張スワップ・データセットに分類されています。

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ますと、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】

ページやスワップの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。

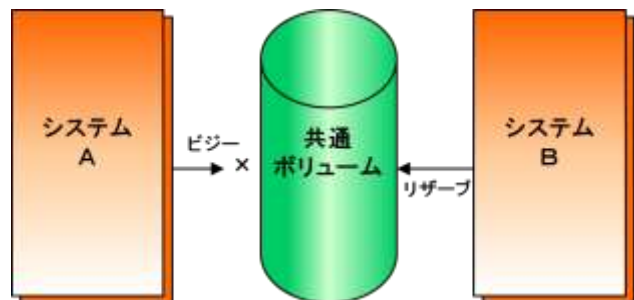
【解説】

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る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。

ページやスワップの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、ページやスワップの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】

- ページやスワップの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。

SWAP01n & SWAP02n

【説明】

スワップ・データセットが存在しないか、または、スワップ・データセットのサービス時間が長すぎます。新たに、スワップ・データセットを割り当てて下さい。

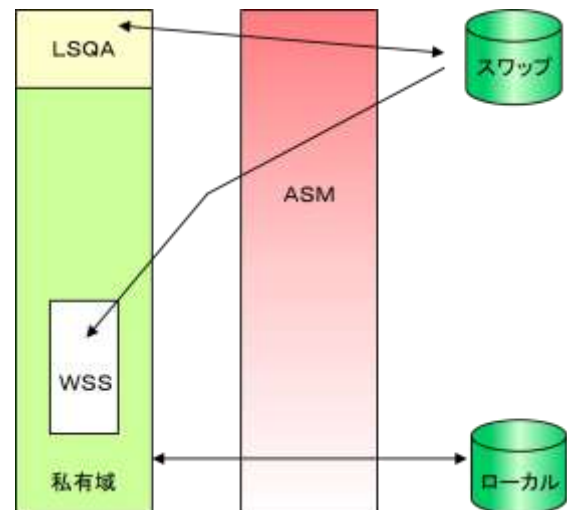
【解説】

TSO/TSSやバッチで運用される業務プログラムは、そのほとんどがスワップの対象となります。このような業務が運用されているシステムでは、そのスワップ動作を効果的に行わせるために、スワップ・データセットを準備する必要があります。

オペレーティング・システムでは、実行を一時中断させたい業務プログラムをスワップ・アウトしようとします。このスワップ・アウトや逆のスワップ・インの動作時には、LSQAやその業務プログラムのワーキング・セット・サイズ内のページを瞬時にページ・アウトやページ・インしようとします。

もし、この際にスワップ・データセットが割り当てられていなければ、それらの大量のページング動作は、ローカル・ページ・データセットに対して実行されます。このため、スワップ・データセットが割り当てられていなければ、通常のページング動作に悪影響が表れます。

ストレージ競合が発生しているシステムで、TSO/TSSの業務プログラムを実行しているような場合、1つのトランザクション処理のたびに、業務プログラムはスワップ・インとスワップ・アウトを繰り返しております。このため、スワップ・データセットもページ・データセットと同様に高速のサービス時間を保証することが必要です。このため、ASMは複数のスワップ・データセットをサポートしています。また、スワップ動作では大量のページ処理が必要であるため、1回の処理単位である スワップ・スロットは12ページの大きさを持っています。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1つのスワップ・データセットを割り当てることが必要です。もし、スワップ・データセットの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。特に改善点が無ければ、新たにスワップ・データセットを追加して下さい。



【対応策】

- 必ずスワップ・データセットを割り当てる。
- スワップ・データセットのアクセス・パスへの使用率を低く保つ。
- 半導体ディスク装置や拡張記憶機構の導入を検討する。

SWAP03n

【説明】

スワップ・データセット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。スワップ・データセットの総容量を増やして下さい。

【解説】

スワップ・データセットは、オペレーティング・システムにおけるスワッピング制御を高速処理するためのデータセットです。このスワッピング動作では1回に大量のページを転送するために、スワップ・データセットのスロットは12ページ分の大きさとなっています。(ページ・データセットのスロットは1ページ分の大きさです。)

このスワップ・データセットの総容量は、下記の理由により、大きめに設定されることをお勧めします。

- ① ASMは、1回のアクセスで複数のスワップ・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、スワッピングのサービス時間が長くなります。
- ② スワップ・データセットが満杯になると、スワップのためにローカル・ページ・データセットを使用します。このため、通常のページング時間に悪影響を及ぼす可能性があります。

【対応策】

- 新たなスワップ・データセットを追加する。

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スワップが発生しています。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スワップ・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

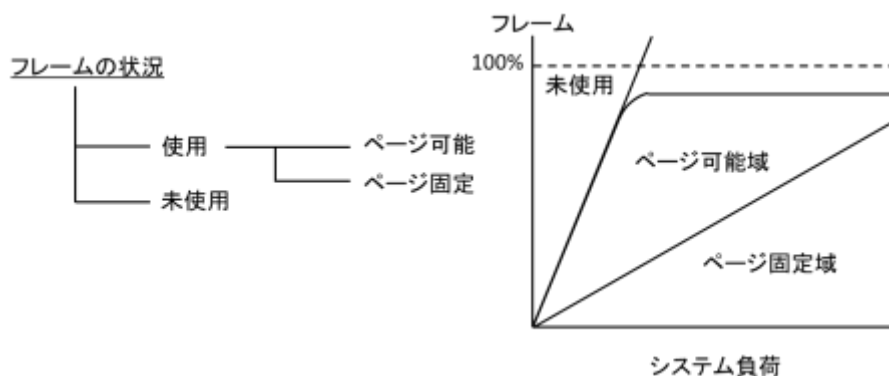
【対応策】

- RMFのモニタⅡやPDLおよびSAR/D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。



(注)

RMFはIBM、PDLは富士通、SAR/Dは日立システムのソフトウェア・モニタです。



SWAP11n

【説明】

外部記憶の未使用ページ・スロット数が不足したことによるスワップが発生しています。通常、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、外部記憶の容量を拡大して下さい。

【解説】

システムの業務負荷が上昇するにつれ、全ての仮想記憶データ(ページ)を主記憶(もしくは拡張記憶)に格納しておくことができなくなります。また、適用業務プログラムにおいては、資源管理プログラムにより瞬時であってもスワップ・アウトの対象となることがあります。このような事態が発生し、プログラムが主記憶や拡張記憶から追い出されるとプログラムは外部記憶に移されます。

以上の理由により、システムの業務負荷に比例して、外部記憶の使用スロット数が増加します。適用業務プログラムの中には、GETMAINやVIO処理を繰り返し行うことにより、そのプログラム自身が使用する仮想記憶域を増加させるものがあります。このような際、十分な大きさの外部記憶を準備していなければ、そのプログラムは異常終了することになります。また、OSのコントロール・プログラム自体が外部記憶を使用できなければ、システム全体に影響を与えます。

資源管理プログラムはこのような事態を回避するために、常時、外部記憶の使用率を監視しています。もし、ある一定以上の外部記憶スロットが使用状態となると、新たなプログラムの開始を阻止すると同時に急激に外部記憶を使用し始めた適用業務プログラムをスワップ・アウトします。新たなプログラムの開始を阻止、また現行のシステム負荷を軽減するために、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
- イニシェータが新たなジョブの処理を開始することを阻止
- ドメイン毎のプログラム多重度を最低値に減少させる。
- 新たなスワップ・インを阻止する。

【対応策】

- 新たなページ・データセットを追加する。
- 事前にページ・データセットをフォーマットしておき、必要な時に PAGEADDコマンドでスペース追加を行える体制を検討する。
- オペレータ・コンソールのログを分析し、外部記憶不足となった時のメッセージから原因と考えられる適用業務プログラムを知り調査する。

SWAP12n

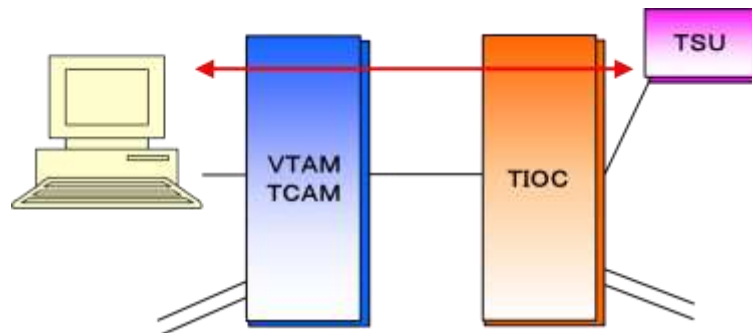
【説明】

TSOやTSSの端末制御プログラム(TIOC)のバッファ不足によるスワップ・アウトが発生しています。通常、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、必要であれば端末制御プログラムのバッファを拡張して下さい。

【解説】

TSOやTSSはVTAMやTCAMとのインターフェイスを簡素化するために端末制御プログラム(TIOC)を用意しています。TSOやTSSのユーザ空間で実行されるコマンド群やCLISTなどは全てこのTIOCを使用して端末との通信を行っています。

TIOCは全ての端末との通信に必要なバッファを準備し、その効率を高めています。



しかし、コマンドやCLISTからの出力メッセージが極端に多くなると、出力側(TSUから端末へのデータ転送)でバッファが満杯となり、一切の入力側処理が行えなくなります。このため、出力側で利用できるバッファ数を制限しています。もし、このTIOCのバッファ数が充分確保されていなければ、出力側のメッセージ転送を要求したユーザ空間は資源管理プログラムに強制的にスワップ・アウトされます。特定のユーザ空間の問題であったとしても、バッファ不足が発生しますと、TSOやTSSのレスポンスが全体的に悪化します。このため、このTIOCのバッファ不足などは発生するべきではありません。

【対応策】

■ IBMシステム

TSO/VTAMの場合はSYS1. PARMLIBのTSOKEYxxメンバーのHIBFREXT、TSO/TCAMの場合はIKJPRMxxメンバーのOWAITHIの値を再吟味して下さい。

■ 富士通システム

VTAM-Fの場合はSYS1. PARMLIBのKEQPRMxxのOWAITHI、VTAM-Gの場合はTSSKEYxxのHIBFREXTの値を再吟味して下さい。

■ 日立システム

SYS1. PARMLIBのJCTPRMxxのメンバーのOWAITHIの値を再吟味して下さい。

SWAP13n (MVSのみ)

【説明】

OSが主記憶の使用効率を高めるために、業務プログラムを強制的にスワップアウトしました。このスワップは主記憶の競合が非常に高い際に発生します。主記憶の競合状態を注意深く監視して下さい。

【解説】

拡張記憶がないシステムでは、主記憶が効率的に使用されていなければ外部記憶とのページングが増大し、システムパフォーマンスが低下します。もし十分な量の拡張記憶があれば、拡張記憶を使用して高速なページングの処理が可能であるため、顕著なシステムパフォーマンスの低下は発生しません。しかし、高速なページング処理を拡張記憶で行った場合でも、ページ不在割り込みなどによるシステムオーバヘッドは確実に増加しています。

OSは、このページング処理におけるオーバヘッド量を常に監視しています。そして、オーバヘッド量が一定(CPU時間で5%程度)以上となると、主記憶の使用効率が悪化したとして、各種の制御を開始します。最初に、スワップ可能で、ストレージ分離機能(ストレージ・アイソレーション)を使用していない業務プログラムの実行効率を監視し始めます。この監視では、各業務プログラムが占有している主記憶量とページ不在割り込みの発生頻度を計測します。そして、拡張記憶からのページ転送を一括して行うブロックページング機能を利用することにより、ページ不在割り込みの回数が改善されるかを実測します。

ブロックページングを利用してもページ不在割り込みの回数が改善されない業務プログラムでは、CPU使用時間の解析が行われます。そのプログラムが使用したCPU時間と、そのプログラムが引き起こしたページ不在割り込みの処理に必要としたCPU時間の比が求められます。その値により、オーバヘッド量の判定が行われ、オーバヘッドが高いと考えられる業務プログラムは強制的にスワップアウトされます。

この様にして、主記憶を効率的に使用していない業務プログラムの実行を一時中断し、スワップアウトさせることにより他の業務プログラムの実行効率を改善します。

ES/1 NEOでは、このスワップ要因によるスワップアウトを検出すると、このチューニングヒントを作成します。

【対応策】

- 主記憶を過度に使用しています。ドメインごとのMPL値を減少させて下さい。
- それでも改善できない場合は、主記憶の増設を検討する必要があります。

SWAP14n (MVS のみ)**【説明】**

OSがページ不在割り込みの発生回数を抑制するために、業務プログラムを強制的にスワップアウトしました。このスワップは主記憶の競合が非常に高い際に発生します。主記憶の競合状態を注意深く監視して下さい。

【解説】

拡張記憶がないシステムでは、主記憶が効率的に使用されていなければ外部記憶とのページングが増大し、システムパフォーマンスが低下します。もし十分な量の拡張記憶があれば、拡張記憶を使用して高速なページングの処理が可能であるため、顕著なシステムパフォーマンスの低下は発生しません。しかし、高速なページング処理を拡張記憶で行った場合でも、ページ不在割り込みなどによるシステムオーバーヘッドは確実に増加しています。

ページ不在割り込みが発生した場合、OSはページリクレーンや拡張記憶からのページ転送、もしくは外部記憶からのページイン処理を行う必要があります。このためページ不在割り込みが発生すると、ある一定量のシステムオーバーヘッドが必要となります。OSはシステムオーバーヘッド量を一定量に抑制するために、常時このページ不在割り込みの発生回数を監視しています。もし、システム管理者が指定した以上のページ不在割り込みが発生すると、ページ不在割り込みを頻発している業務プログラムを強制的にスワップアウトします。

ES/1 NEOでは、このスワップ要因によるスワップアウトを検出すると、このチューニングヒントを出力します。

【対応策】

- 主記憶を過度に使用しています。ドメインごとの MPL 値を減少させて下さい。
- それでも改善できない場合は、主記憶の増設を検討する必要があります。
- SYS1.PARMLIBのIESOPTxxメンバーで指定されたRCCPTRTパラメータの最大値が小さすぎることも考えられます。

SWAP15n (MVS のみ)**【説明】**

OSが、主記憶の使用効率を高めるために強制的にスワップアウトした業務プログラムをスワップインしようとしたところ、主記憶に十分な空き領域がありませんでした。このため、主記憶の空き領域を確保するために、実行中の業務プログラムを強制的にスワップアウトしました。このスワップは主記憶の競合が非常に高い際に発生します。主記憶の競合状態を注意深く監視して下さい。

【解説】

OSは主記憶の使用効率を調べるために、合計CPU使用時間とページングなどの処理に必要であったCPU時間の割り合いを常時監視しています。もし、ページングなどによるオーバヘッド量が大きくなると、オーバヘッド量を抑制するために問題と考えられる業務プログラムを強制的にスワップアウトします。すると、他の業務プログラムが占有することが出来る主記憶容量が増加するために、システム全体におけるページング処理などが減少し、オーバヘッド量も抑制することが出来ます。

通常のスワップ制御では、主記憶や拡張記憶の競合が低いと判定された時点で、次にスワップインすべき業務プログラムを選択します。しかし、主記憶の使用効率を改善するためにスワップアウトされた業務プログラムは、より素早くスワップインし、処理を続行する必要があります。このために、通常のスワップイン処理ではなく、特別な処理を準備しておく必要があります。

この様な優先処理のために、OSはその業務プログラムの種類ごとに定めた時間間隔ごとに、このスワップ要因でスワップアウトされた業務プログラムのスワップイン処理を行います。この時間間隔はTSOで30秒、その他の業務プログラムでは10分となっています。

一定時間間隔ごとに、業務プログラムの入れ替え(スワップ)が行われるわけですが、その業務プログラムは主記憶の使用効率が悪いためにスワップアウトされたものです。このためスワップインしようとした際、スワップインすべき十分な主記憶の空き領域が確保できないことが考えられます。また、強引にスワップインしても、また主記憶の使用効率が低下することが考えられます。この様な場合、新たな業務プログラムをスワップインする前に、主記憶の競合状況を低下させるために実行中の業務プログラムをスワップアウトする必要があります。

ES/1 NEOでは、このスワップ要因によるスワップアウトを検出した場合、このチューニングヒントを出力します。

【対応策】

- SWAP13nのチューニングヒントも参照して下さい。
- 主記憶を過度に使用しています。ドメインごとのMPL値を減少させて下さい。
- それでも改善できない場合は、主記憶の増設を検討する必要があります。

VS01n & VS02n

【説明】

システム待ち合わせ域(SQA やESQA)の大きさが不足したため、共通サービス域(CSAやECSA)の領域を流用した。システム待ち合わせ域を拡張して下さい。

【解説】

オペレーティング・システムでは、システム内で発生したソフトウェア事象を記憶するために、SQAやESQAに待ち行列を作成します。しかし、記憶するべきソフトウェア事象の数が増加し、SQAやESQAが不足しますと、CSAやECSAを待ち行列用の領域として使用し始めます。これは、システム・ダウンを回避する非常手段であり、通常、発生してはならない事態です。

【対応策】

- SQAもしくはESQAを拡張する。

VS03n & VS04n

【説明】

16MBのアドレス境界以下のシステム待ち合わせ域(SQA)、もしくは共通サービス域(CSA)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。

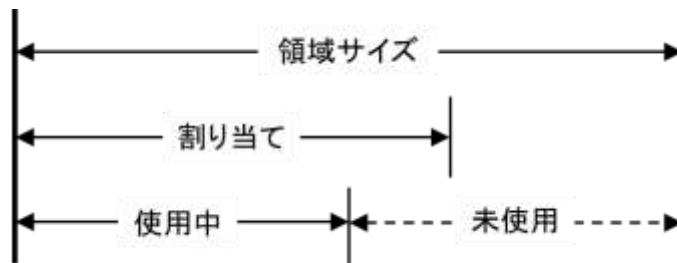
【解説】

SQAやCSA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。SQAやCSA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】

- SQAもしくはCSA 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。

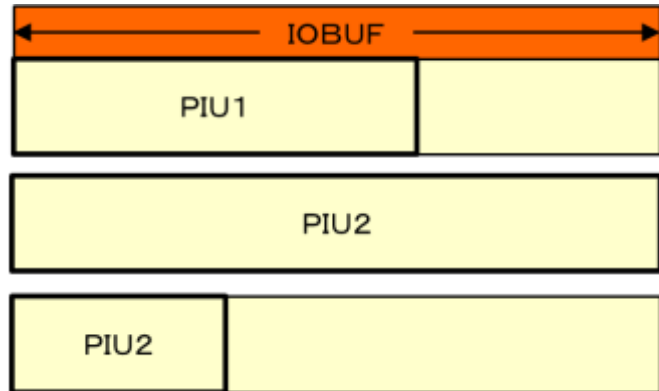
VTAM02n

【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。

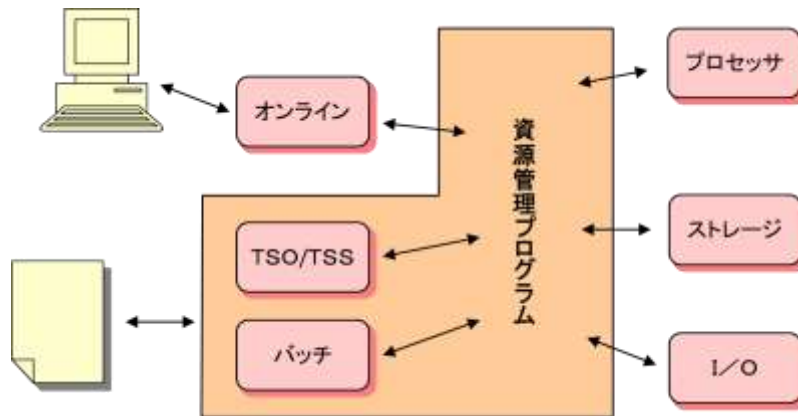
WKLD01n

【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総サービス量として管理/制御します。この総サービス量には、プロセッサ使用量、ストレージ使用量、I/O使用量が含まれており、次の計算式で求めることができます。

$$\begin{aligned} \text{プロセッサ使用量} &= \text{CPUサービスユニット量} \times \text{CPUサービス係数} \\ &+ \text{SRBサービスユニット量} \times \text{SRBサービス係数} \end{aligned}$$

$$\text{CPUサービスユニット量} = \frac{\text{TCB モードでを使用した CPU 時間}}{\text{プロセッサ速度定数}}$$

$$\text{SRBサービスユニット量} = \frac{\text{SRB モードでを使用した CPU 時間}}{\text{プロセッサ速度定数}}$$



プロセッサ速度定数とは、CPUに依存した値で、資源管理プログラムが持っています。
また、日立システムにはSRBデータはありません。

$$\text{ストレージ使用量} = \text{MSOサービスユニット量} \times \text{MSOサービス係数}$$

$$\text{MSOサービスユニット量} = \frac{\text{使用フレーム数}}{50} \times \text{MSOサービス係数}$$

【注意】

- 最近のIBMシステムではMSOサービスユニット量が非常に大きな値になるため、MSOサービス係数にゼロをセットしてMSOサービスユニット量を使用しないシステムが増えています。このような場合にはMSOサービスユニット量を取得することができず、ストレージ資源使用量のバランスを判定することができません。

その為、V3L18よりIBMシステムでは「MSOサービスユニット量」ではなく

$$\text{ストレージ使用量} = \text{フレーム専有時間}$$

としてストレージ資源使用量のバランスを判定しています。

$$\text{I/O使用量} = \text{I/Oサービスユニット量} \times \text{I/Oサービス係数}$$
$$\text{I/Oサービスユニット量} = \text{入出力要求回数} \text{ 又は } \text{デバイス・コネクト時間}$$


IBMシステムの場合は選択可能で、その他のシステムでは入出力要求回数のみが使用できます。

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。

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。

COUPLING01n

【説明】

結合機構のプロセッサの競合が高すぎますので、結合機構のプロセッサの処理能力を高めて下さい。結合機構のプロセッサ能力が不足しているとシステム間の通信速度が低下し、システム運用のパフォーマンス目標を達成することが困難になります。

【解説】

パラレルシステムでは、複数のOSで制御されるシステムを上手く稼働させるために、それらのOS間で通信を行っています。その通信経路を使用して転送される情報には、リソースの排他制御情報、データベースなどのバッファ情報などがあります。これらの情報はシステムの保全性を確保するために必要なものであり、アプリケーションプログラムの実行に欠くことができないものです。

IBMのパラレルSYSPLEXシステムでは、このシステム間通信を効率的に行うために結合機構と呼ばれる装置を使用しています。この結合機構はただ単なる通信パスであるだけでなく、結合機構自体もストレージを内蔵した一つのプロセッサとなっております。結合機構に接続されたMVSシステムからの通信要求がやってくると、結合機構のプロセッサがその要求を認識し、必要な処理を行います。通常、パラレルSYSPLEX全体の能力の20%程度の能力を結合機構に持たせなければならないとされています。

ES/1 NEOでは、常時結合機構のプロセッサ使用率を監視し、その使用率が異常に高くなるとこのチューニングヒントを出力します。その際には、何らかの異常が発生していないかなど、詳細な調査が必要となります。また、このチューニングヒントが出力された際には、各MVSシステムからの通信要求が激増していないかを調査して下さい。前述しましたように、結合機構が動作するのはMVSシステムから通信要求があった場合です。このため、MVSシステムに何らかの異常が発生し、結合機構に過度な通信要求を出した場合、結合機構のプロセッサ使用率も高まります。

COUPLING02n

COUPLING021**【説明】**

結合機構のストレージの使用率が高すぎます。結合機構の制御ストレージの容量を追加して下さい。結合機構のストレージ容量が不足していると、新たなストラクチャを使用できません。このため、パフォーマンス目標を達成することが困難になります。

【解説】

パラレルシステムでは、複数のOSで制御されるシステムを上手く稼働させるために、それらのOS間で通信を行っています。その通信経路を使用して転送される情報には、リソースの排他制御情報、データベースなどのバッファ情報などがあります。これらの情報はシステムの保全性を確保するために必要なものであり、アプリケーションプログラムの実行に欠くことができないものです。

IBMのパラレルSYSPLEXシステムでは、このシステム間通信を効率的に行うために、結合機構と呼ばれる装置を使用しています。この結合機構はただ単なる通信パスであるだけでなく、結合機構自体もストレージを内蔵した一つのプロセッサとなっております。結合機構に接続されたMVSシステムからの通信要求がやってくると、結合機構のプロセッサがその要求を認識し、その通信すべき情報を結合機構内のストレージに記録します。ストレージ内では、キャッシュやストラクチャーと言った情報の種類に応じた管理がなされています。

ES/1 NEOでは、常時結合機構のストレージの使用率を監視し、その使用率が異常に高くなるとこのチューニングヒントを出力します。その際には、何らかの異常が発生していないかなど、詳細な調査が必要となります。また、このチューニングヒントが出力された際には、各MVSシステムからの通信要求が激増していないかを調査して下さい。前述しましたように、結合機構が動作するのはMVSシステムから通信要求があった場合です。このため、MVSシステムに何らかの異常が発生し、結合機構に過度な通信要求を出した場合、結合機構のストレージ使用率が高まることも考えられます。

COUPLING022**【説明】**

結合機構のストレージ容量が大きすぎます。結合機構のストレージの使用率が50%にも達していません。ストレージを無駄にしていると考えられます。再度、結合機構に割り当てるべきストレージ容量の見積もりを行って、適正値を算出してください

【解説】

パラレルシステムでは、複数のOSで制御されるシステムを上手く稼働させるために、それらのOS間で通信を行っています。その通信経路を使用して転送される情報には、リソースの排他制御情報、データベースなどのバッファ情報などがあります。これらの情報はシステムの保全性を確保するために必要なものであり、アプリケーションプログラムの実行に欠くことができないものです。

IBMのパラレルSYSPLEXシステムでは、このシステム間通信を効率的に行うために、結合機構と呼ばれる装置を使用しています。この結合機構はただ単なる通信パスであるだけでなく、結合機構自体もストレージを内蔵した一つのプロセッサとなっております。結合機構に接続されたMVSシステムからの通信要求がやってくると、結合機構のプロセッサがその要求を認識し、その通信すべき情報を結合機構内のストレージに記録します。ストレージ内では、キャッシュやストラクチャーといった情報の種類に応じた管理がなされています。

ES/1 NEOでは、常時結合機構のストレージの使用率を監視し、その使用率が低い場合にはこのチューニングヒントを出力します。結合機構のストレージは十分な容量が割り当てられている必要はありますが、必要以上に割り当てても資源の無駄となります。

ストレージの使用量を継続的に監視し、その結合機構に必要な容量がどのくらいであるかを見積もってください。

COUPLING03n

【説明】

結合機構への要求の多くがパスビジーや全てのサブチャネルがビジーであるために待たされています。システム間の通信速度が低下し、システム運用のパフォーマンス目標を達成することが困難になると思われます。

【解説】

パラレルシステムでは、複数のOSで制御されるシステムを上手く稼働させるために、それらのOS間で通信を行っています。その通信経路を使用して転送される情報には、リソースの排他制御情報、データベースなどのバッファ情報などがあります。これらの情報はシステムの保全性を確保するために必要なものであり、アプリケーションプログラムの実行に欠くことができないものです。

IBMのパラレルSYSPLEXシステムでは、このシステム間通信を効率的に行うために、結合機構と呼ばれる装置を使用しています。この結合機構はただ単なる通信パスであるだけでなく、結合機構自体もストレージを内蔵した一つのプロセッサとなっております。また、結合機構とMVSシステムの間には、1秒間に100MBものデータを転送できる高速データバスが用意されています。この高速データバスを通じて、通信のためのメッセージがやり取りされます。

ES/1 NEOでは、常時結合機構との高速データバスの使用状況を監視し、異常を見つけるとこのチューニングヒントを出力します。その際には、何らかの異常が発生していないかなど、詳細な調査が必要となります。また、このチューニングヒントが出力された際には、各MVSシステムからの通信要求が激増していないかを調査して下さい。

PRD10n

【説明】

システム負荷指標の一つであるディスク・ボリューム群への総入出力回数とプロセッサ使用率の間に相関傾向があることが確認されました。この結果から入出力サブシステムに対するシステム負荷量の最大値を算出して評価・解析をしています。

【解説】

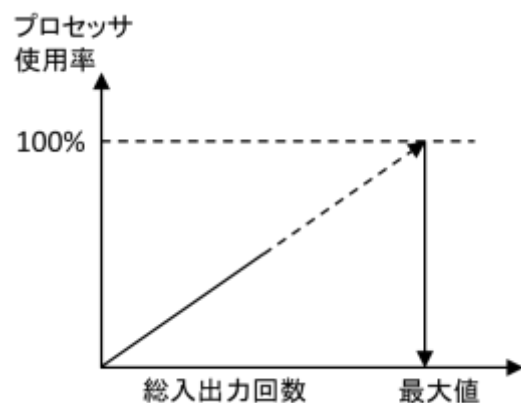
システムでは数多くの業務プログラムが同時に実行されており、それらの業務プログラムがプロセッサを共用しています。また、運用中の業務プログラムを制御するためのオペレーティング・システムもプロセッサを使用します。このオペレーティング・システムによるプロセッサ使用をシステム・オーバヘッドと呼び、ページングやスワップ動作および入出力割り込み処理に起因されています。このため、同時に実行される業務プログラムの本数(プログラム多重度)に比例してシステム・オーバヘッド量も増加します。また、業務プログラムによるプロセッサ使用もプログラム多重度に比例します。この際、プロセッサ使用率はプログラム多重度に比例し、線形で増加します。

運用されているシステムには、オンライン主体の場合があります。その際、プログラム多重度とプロセッサ使用率の間には、前述のような相関関係が確認できないことがあります。これは、プログラム多重度で表す「同時に実行されているプログラム本数」が、同時にスワップ・インされているアドレス空間数であることに起因します。

オンライン制御プログラムによっては、オンライン・システム起動時に一定数のアドレス空間を立ち上げ(スワップ・イン状態)で運用するものがあります。この場合、実際に処理されるトランザクションに関係なくアドレス空間が存在するため、プログラム多重度は業務負荷を表す指標にはなりません。

オンライン・システムでは、本来、同時に実行されているオンライン・トランザクション数をシステム負荷指標とすべきです。しかし、システム内ではオンライン制御プログラムだけではなくネットワーク制御プログラムなどの他のプログラムも数多く実行されています。このため、処理トランザクション数を統合的に表すディスク装置への総入出力回数をシステム負荷指標とすることが最適です。プログラム多重度をシステム負荷指標とすることが出来ないシステムでは、このディスク装置への総入出力回数をシステム負荷指標とすることができます。

総入出力回数とプロセッサ使用率について直線回帰分析を行ったところ、2つのデータ間に相関傾向を確認できました。この際の総入出力回数とプロセッサ使用率との相関度を表す相関係数が同時に表示されます。この相関係数が0.7以上で1に近い程、相関傾向が高いことを意味します。現行システムにおいて、総入出力回数が増加していった場合、プロセッサ使用率が100%に達した時点の総入出力回数を相関判定の結果から算出することができます。これは、相関判定で総入出力回数とプロセッサ使用率の関係を直線式で表すことで可能となります。



【対応策】

- 今後、業務負荷とシステム負荷指標としての総入出力回数との関連を調査することで、将来のプロセッサ使用率の変動を知ることができます。

【注意点】

- 評価対象の日や時間帯により、相関係数が変化する事があります。
- 相関係数が低い際には、入力とする日や時間帯を変更して再実行して下さい。

PRD11n

【説明】

システム負荷指標とシステム資源の間に相関傾向があることが確認されました。今後の業務負荷が増大した際のシステム資源の状況を把握するための基礎資料として下さい。

【解説】

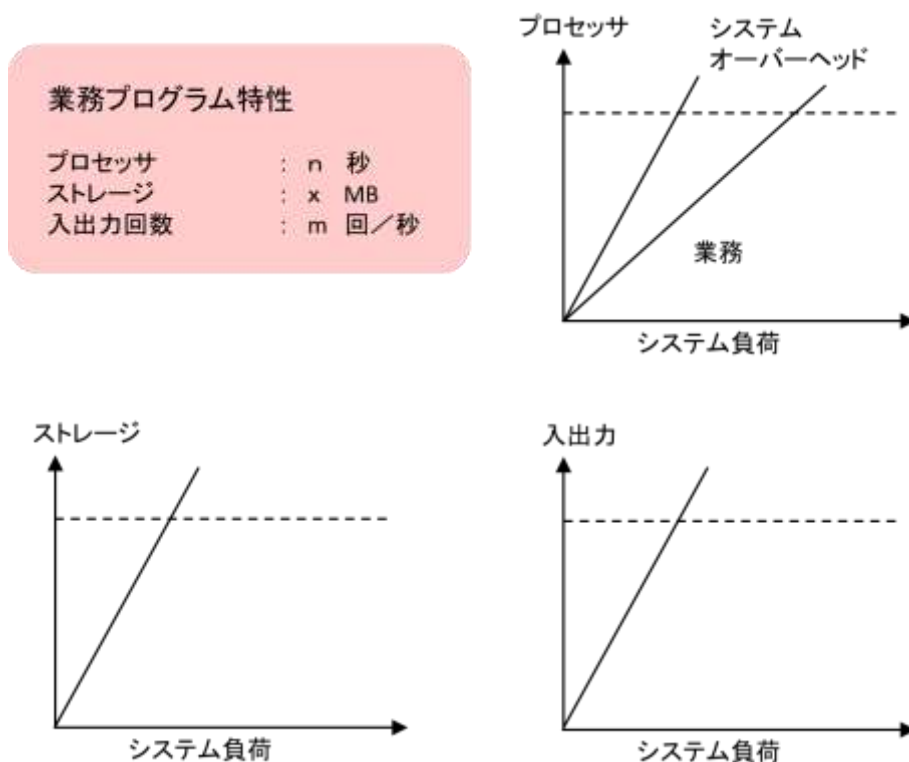
システム資源としては、プロセッサ、主記憶や拡張記憶(あるいはシステム記憶)と入出力サブシステムがあります。これらのシステム資源とシステム負荷量との特性を考える際、実際に実行している業務プログラムの特性を吟味する必要があります。

業務プログラムの特性とは、1つのトランザクション(バッチジョブやオンライン・トランザクションなど)を処理するために必要なシステム資源の使用量を意味します。これらの使用量はパフォーマンス計測ツールが収集したデータから求めることができます。

1つのトランザクションで使用するシステム資源の量が判明すると、システム負荷量が増加するごとに線形にシステム資源の負荷も増加することが判ります。例えば、プロセッサ使用率の場合、1本の業務プログラムで5%を使用する時には、10本では50%を使用すると予測できます。しかし、実際にはオペレーティング・システムのオーバーヘッドを考慮しなければなりません。このため、プロセッサ捕捉率で補正する必要があります。

ストレージの場合は、トランザクションが頻繁に使用するWSS(ワーキング・セット・サイズ)を使用します。この際、オペレーティング・システムが専有する容量やページ固定される容量などを考慮する必要があります。

プログラム多重度が増大すると入出力要求回数も同様に増加します。これは、データベースへのアクセスやログ・ファイルへのアクセスが発生するためです。



運用されているシステムにおいて、各システム資源の使用状況に影響を与える変動要素を見つけることは非常に重要です。各システム資源に対応する変動要素を見つけるには、そのリソースの使用率と変動要素との相関判定を行うことで可能となります。

この相関判定では、次に示すシステム資源と負荷指標についての相関関係を判定しています。

<システム負荷指標>

- ・プログラム多重度
- ・総入出力回数

<システム資源>

- ・プロセッサ使用率
- ・主記憶使用率
- ・拡張記憶あるいはシステム記憶の使用率

このチューニング・ヒントでは、システム負荷指標の名称と相関傾向にあるシステム資源の名称及びその相関係数を表示しています。この中で、相関係数は、0.7以上であれば相関があると判断しています。

システム負荷指標としてはプログラム多重度と全ディスク・ボリューム群への秒当たりの総入出力回数があります。これらのシステム負荷指標とシステム資源の間に相関傾向を確認することができました。このことから、システム負荷指標が大きくなるとシステム資源も線形で増加することを意味します。

【対応策】

- 今後の業務負荷の増大に伴うシステム資源の使用状況を監視するための基礎資料として下さい。

【注意点】

- 評価対象の日や時間帯に応じて、相関係数が変化する事があります。

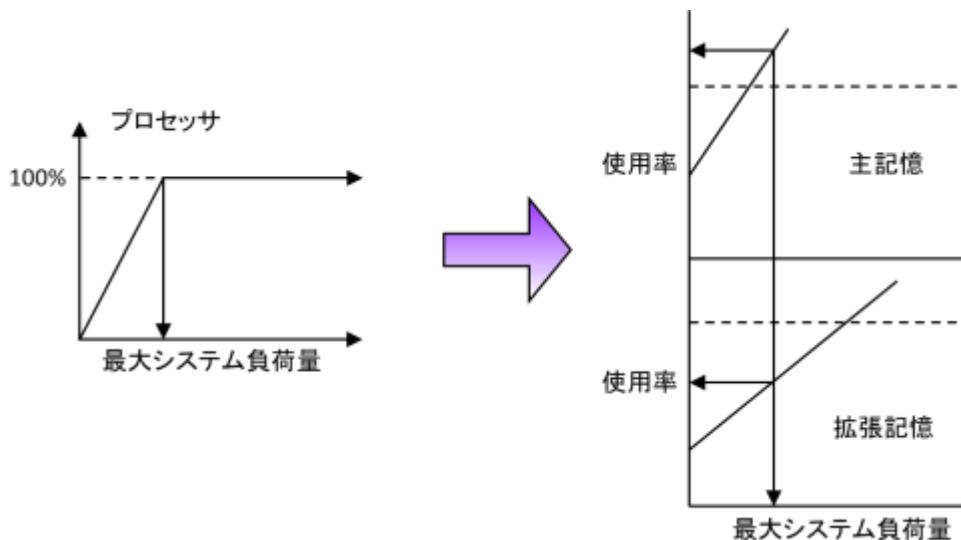
PRD201

【説明】

システム負荷指標と主記憶使用率との相関判定で相関傾向がある場合に、最大システム負荷量を与えた際に主記憶使用率が100%を越える結果がでました。

【解説】

最大システム負荷量を求める場合、まず、システム負荷指標とプロセッサ使用率の相関判定を行います。その結果からプロセッサ使用率が100%に達した時点のシステム負荷量を最大値としています。この際のシステム負荷指標としてはプログラム多重度と総入出力回数を対象としています。この結果からシステム負荷指標と主記憶使用率の相関判定を行い、相関傾向が確認されると最大システム負荷量に対応した主記憶使用率を算出しています。ここで算出された主記憶使用率が100%を越えた際にこのチューニング・ヒントが表示されます。



主記憶使用率(全体容量から未使用量を減じた値)は、オペレーティング・システムの制御がページング処理の高速化のために必ずある一定量だけを未使用域として維持するために100%になることはありません。これは、要求量が主記憶容量を越えるとその結果ページング処理が多発し業務プログラムに悪影響を与えます。これらのことから、もし、直線回帰で100%を越える結果が得られた際には、ページング処理が多発していることを示します。また、プロセッサの能力と主記憶容量のバランスを考えた際に、主記憶容量がプロセッサ能力に比べて小さいことを意味します。

【対応策】

- 今後、主記憶使用率の監視を継続して下さい。

【注意点】

- 評価対象の日や時間帯により、相関係数が変化する事があります。

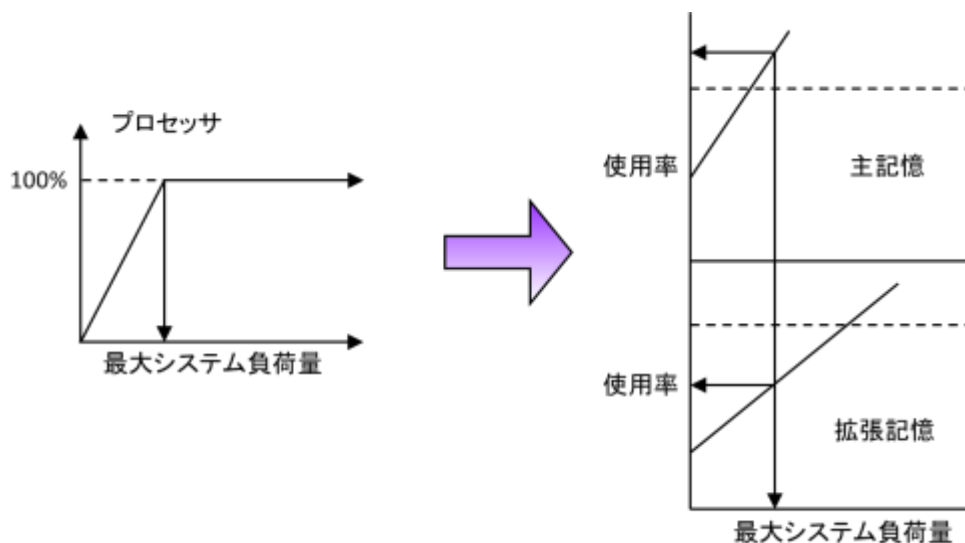
PRD301

【説明】

システム負荷指標と拡張記憶(システム記憶)使用率との相関判定で相関傾向がある場合に、最大システム負荷量を与えた際に拡張記憶(システム記憶)使用率が100%を越える結果がでました。

【解説】

最大システム負荷量を求める場合、まず、システム負荷指標とプロセッサ使用率の相関判定を行います。その結果からプロセッサ使用率が100%に達した時点のシステム負荷量を最大値としています。この際のシステム負荷指標としてはプログラム多重度と総入出力回数を対象としています。この結果からシステム負荷指標と拡張記憶(システム記憶)使用率の相関判定を行い、相関傾向が確認されると最大システム負荷量に対応した拡張記憶(システム記憶)使用率を算出しています。ここで算出された拡張記憶(システム記憶)使用率が100%を越えた際にこのチューニング・ヒントが表示されます。



拡張記憶(システム記憶)使用率(全体容量から未使用量を減じた値)は、オペレーティング・システムの制御がページング処理の高速化のために必ずある一定量だけを未使用域として維持するために100%になることはありません。これは、要求量が拡張記憶(システム記憶)容量を越えるとその結果ページング処理が多発し業務プログラムに悪影響を与えます。これらのことから、もし、直線回帰で100%を越える結果が得られた際には、ページング処理が多発していることを示します。また、プロセッサの能力と拡張記憶(システム記憶)容量のバランスを考えた際に、拡張記憶(システム記憶)容量がプロセッサ能力に比べて小さいことを意味します。

【対応策】

- 今後、拡張記憶(システム記憶)使用率の監視を継続して下さい。

【注意点】

- 評価対象の日や時間帯により、相関係数が変化する事があります。

PRD401

【説明】

現行の入出力サブシステム構成で最大システム負荷量进行处理しようとした場合に、特定のディスクボリュームのデバイス使用率が100%を越えてしまいます。この際の最大システム負荷量は、プロセッサ使用率が100%に達した際に予想される総入出力回数から算出しています。

【解説】

ディスク・ボリュームの応答時間を説明する際には、待ち行列理論を利用する必要があります。実際に待ち行列理論をディスク・ボリュームに適用した場合には、右図のようになります。母集団とは、そのディスク・ボリュームをアクセスする業務プログラムであり、待ち行列は同時にアクセス要求が発生した際に逐次制御でUCBの待ち行列につながれます。先行する入出力が完了すると次の要求がディスク・ボリューム（サーバ）で処理されます。

右図において、リトルの法則では、平均待ち個数は平均アクセス回数と平均待ち時間の積に等しいとされています。つまり、数（長さ）を時間に換算する際に利用され、次の式で表現されます。

$$\begin{aligned} R &= Q + S \\ Nq &= \lambda \times Q \quad \therefore Q = Nq / \lambda \\ S &= \rho / \lambda \quad \therefore \rho = S \times \lambda \end{aligned}$$

上記の式は、パフォーマンス計測ツールで収集されたデータから時間に換算する際に利用されています。また、この関係をプロットすると応答時間や使用率との関連は次のようになります。

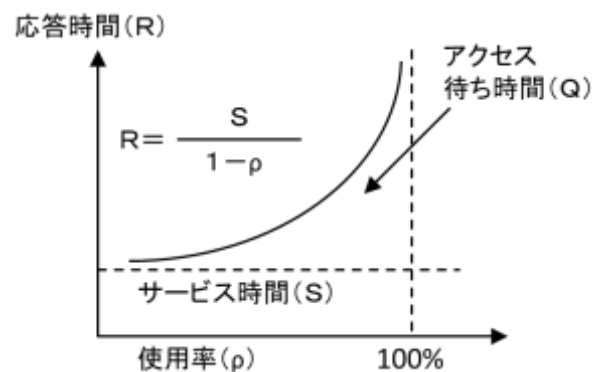
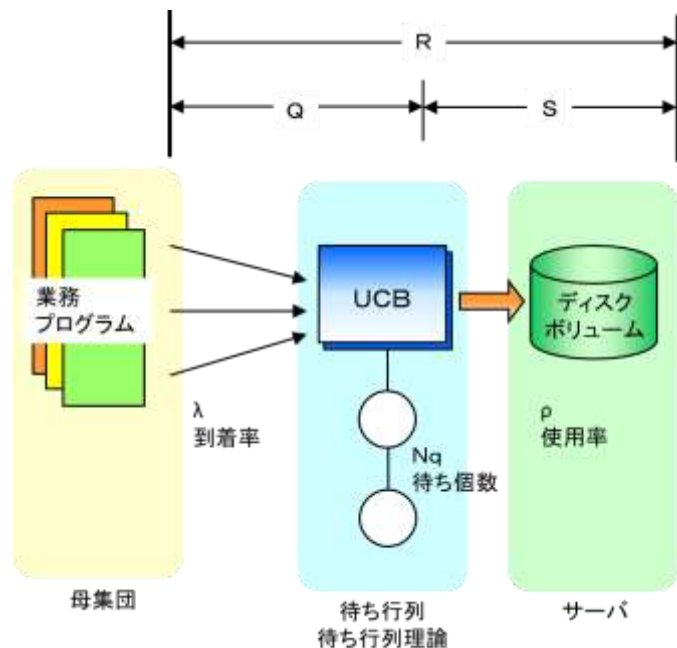
これらのことから、個々のディスク・ボリュームへのアクセス回数が增大した際のディスク装置の使用率を算出することができます。ディスク装置の使用率はアクセス回数（到着率）とサービス時間の積で求められます。その値は100%を越えることはありません。

【対応策】

- 今後の業務負荷の増大に注意して下さい。
- データセットの再配置や分割などでアクセス回数を減らすことを検討して下さい。
- 最大システム負荷量については、「PRD10n」を参照して下さい。

【注意点】

- 評価対象の日や時間帯により、予測された応答時間が変化する事があります。



NOTE01n

【説明】

複数システムの評価で、入力されたデータの範囲が各システムで異なっているかもしくは、同一システムのデータで、途中からプロセッサが変更されています。このため、評価結果が保証できません。

【解説】

複数システムの評価において、入力データに矛盾を検出した場合に出力されます。この原因としては、次の事が考えられます。

- ① 入力データの範囲が各システムで異なる場合複数システムの評価を行う場合、各システムの入力データは同じ時間帯でなければなりません。もし、異なった時間のデータが入力された場合、正確な評価ができません。
- ② 同一システムでプロセッサが変更された場合入力されたデータで、あるシステムのプロセッサが他のプロセッサにスイッチされているため、正確な評価ができません。

【対応策】

- 入力データを調査して、時間帯を選択して下さい。

第2章 評価手法の紹介

CPEPRT00プロセッサでは単一システムのパフォーマンス評価を行います。このパフォーマンス評価のために、各種の評価技法を集大成した手法でもってパフォーマンス・データの解析・評価を行う必要があります。ここでは、その評価手法の概要を紹介します。

2.1 プロセッサの評価

コンピュータ・システムでは数多くの業務プログラムが運用されており、それらの業務プログラムがプロセッサを共用しています。また、運用中の業務プログラムを制御するために、オペレーティング・システムもプロセッサを使用しています。このオペレーティング・システムによるプロセッサ使用をシステム・オーバヘッドと呼んでいます。使用中のシステムのプロセッサ能力が運用中の業務量に比べ余裕がある場合、システム・オーバヘッド量は問題になりません。しかし、プロセッサ能力の余裕が少ない場合、システム・オーバヘッド量を監視・制御する必要があります。

プロセッサ捕捉率

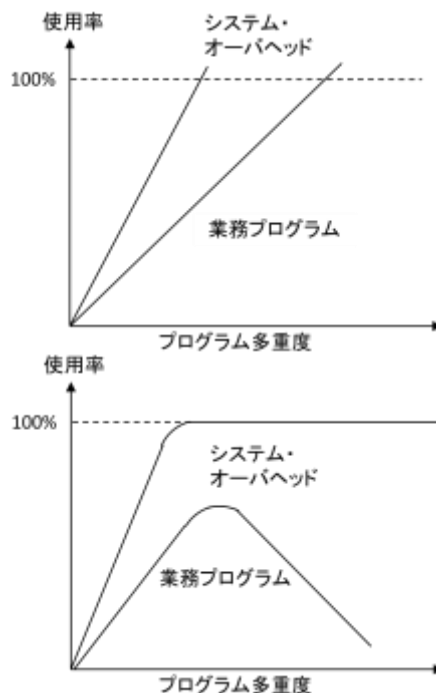
プロセッサ使用率(CPUビジー)は、プロセッサが使用可能であった時間からプロセッサが未使用であった時間を減じた時間で求めています。このため、このプロセッサ使用率には業務プログラムによるプロセッサ使用とシステム・オーバヘッドの両方が含まれています。プロセッサの評価を行うには、それぞれを別々に把握し管理する必要があります。プロセッサ使用率を業務プログラムによるプロセッサ使用とシステム・オーバヘッドに分割するだけで、プロセッサ評価がかなり容易になります。さらに、評価作業を標準化するためには、それら2種類のプロセッサ使用量の比率を係数で表すことが必要になります。このため、導入されたのがプロセッサ捕捉率(キャプチャ・レシオ)であり、次式で算出できます。

$$\text{プロセッサ捕捉率} = \frac{\text{業務プログラムによるプロセッサ時間}}{\text{プロセッサ使用率}} \times 100$$

このプロセッサ捕捉率は、プロセッサ使用率の内どの程度が業務プログラムによるプロセッサ使用であることを示しています。一般的に、プロセッサ捕捉率はオンライン・システムで低く、また科学技術計算システムで高くなる傾向があります。しかし、管理目標としては70%程度の値を設定されることをお勧め致します。

システム・スラッシング

オペレーティング・システムによるプロセッサ使用、つまりシステム・オーバヘッドはページングやスワップの動作および入出力割り込み処理に起因されています。このため、同時に実行されている業務プログラムのシステム・本数(プログラム多重度)に比例して、システム・オーバヘッド量も増加します。また、業務プログラムによるプロセッサ使用もプログラム多重度に比例します。プロセッサ能力に余裕があれば、プロセッサ使用率は、プログラム多重度に比例し、線形で増加します。プロセッサ能力に余裕がないシステムでは、プログラム多重度が増加して行く途中にプロセッサ使用率が100%(プロセッサ能力の限界)に達します。この場合、これ以上にプログラム多重度を増加させるとシステム・オーバヘッドは増加するが、既にプロセッサが飽和しているため、逆に業務プログラムによるプロセッサ使用率が減少します。このように、プログラム多重度が増加することにより業務プログラムによるプロセッサ使用量が少なくなる現象をシステム・スラッシングと呼びます。システム・スラッシングはプロセッサ使用率が100%に達した時に発生するとは限りません。例えば、プロセッサ能力に比べ極端にストレージ(主記憶)容量が小さい場合、プロセッサ使用率が100%に達する前にページング動作によるシステム・オーバヘッドが増加します。また、ページ不在が多発することにより業務プログラムの実行が中断されるため、業務プログラムによるプロセッサ使用量も減少します。つまり、プロセッサ使用率が100%に達する前にシステム・スラッシングが発生します。



オンライン専用システムでのシステム負荷指標

プロセッサなどのコンピュータ・リソースの評価を行う際、そのリソースの使用率とシステム負荷指標の相関判定を行います。前述のプロセッサの評価では、プログラム多重度をシステム負荷指標として使用しています。しかし、オンライン専用システムの場合、プログラム多重度をシステム負荷指標として使用すると、前述のプロセッサ使用率の特性が確認できない場合があります。これは、プログラム多重度で表す“同時に実行されているプログラム本数”が、同時にスワップ・インされているアドレス空間数であることに起因しています。オンライン制御プログラムによっては、オンライン・システム起動時に一定数のアドレス空間をスワップ・イン状態にし、使用し続けるものがあります。このようなオンライン・システムでは、いくつのトランザクションが同時に処理されようとも、このアドレス空間数に変化はありません。このため、プログラム多重度は真のシステム負荷を表す指標ではなくなり、前述のプロセッサ使用率の特性は確認できません。オンライン専用システムでは、本来、同時に処理されているオンライン・トランザクション数をシステム負荷指標とするべきです。しかし、システム内ではオンライン制御プログラムだけでなくネットワーク制御プログラムなどの他のプログラムも数多く実行されています。このため、処理トランザクション数などを統合的に表す磁気ディスク装置へのアクセス回数をシステム負荷指標とすることが最適です。プログラム多重度をシステム負荷指標とできないシステムでは、この磁気ディスク装置へのアクセス回数をシステム負荷指標として使用されることをお勧め致します。

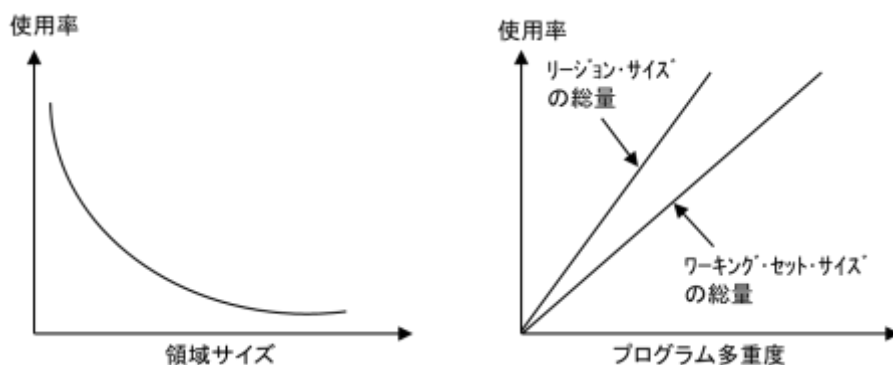
2.2 ストレージの評価

コンピュータ・システムでは、運用される業務プログラムを仮想記憶(仮想アドレス)で動作させています。この仮想記憶の総量が主記憶の容量よりも小さければページングは発生しません。しかし、通常のシステムでは、仮想記憶の総量が主記憶容量の数十倍から数百倍の大きさになっています。このため、運用中の業務プログラムで頻繁に使用されるページのみが主記憶内に存在することができるページであると言えます。

ワーキング・セット

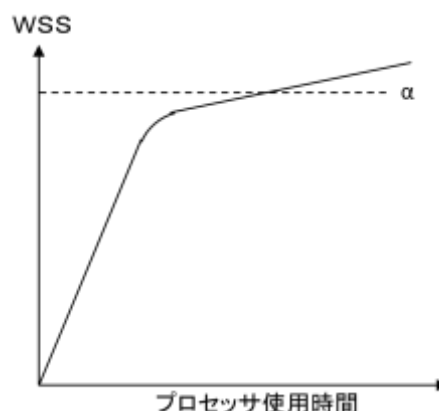
業務プログラムが実行のために必要とする仮想記憶の大きさはリージョン・サイズであると言えます。このリージョン・サイズの中には、滅多に使用されることのないエラー処理ルーチンや使用されることのないデータ域などが含まれています。一方、業務プログラムが頻繁に使用する領域には、メイン・ルーチンとそれにより参照・更新されるデータ域が含まれています。この頻繁に使用される領域のことをワーキング・セットと呼びます。また、その大きさのことをワーキング・セット・サイズ(WSS)と呼びます。

リージョン・サイズとワーキング・セット・サイズの総量は、同時に実行される業務プログラムの本数(プログラム多重度)に比例して増加します。もし、総リージョン・サイズが主記憶の容量を越えると、主記憶に割り当てられたページ群の内、最も長時間使用されなかったページが主記憶から追い出されます。プログラム多重度が増加して主記憶容量以上の総リージョン・サイズになると、その後は1本の業務プログラムが専有できる主記憶容量が減少し始めます。



ページングとワーキング・セット・サイズ

業務プログラムが専有できる主記憶容量が減少するとページングが発生します。ここで、ページングとワーキング・セット・サイズについて考察します。ワーキング・セット・サイズは、業務プログラムが頻繁に使用する領域の大きさです。オペレーティング・システムはこのワーキング・セット・サイズを実測するために、一定時間間隔ごとに業務プログラムが参照・変更したページ数を調査します。もし、ある業務プログラムがその時間間隔内で一度もプロセッサを使用しなければ、そのワーキング・セット・サイズはゼロとなります。その時間間隔内で、プロセッサを使用すればするほど、その業務プログラムのワーキング・セット・サイズは増加します。しかし、業務プログラムが本当に頻繁に使用する領域まで(α 点まで)は急激に増加しますが、以降、その増加率は減少します。

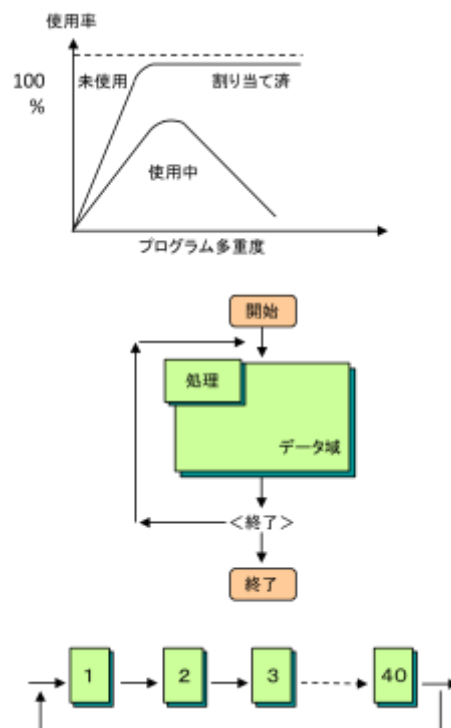


プログラム多重度が増加し、総リージョン・サイズが主記憶容量を超えるとページングが発生します。このページングにより業務プログラムのプロセッサ使用が中断されることも多くなります。すると、ワーキング・セット・サイズを実測するための時間間隔内におけるプロセッサ使用時間が減少するため、その業務プログラムのワーキング・セット・サイズも減少します。このようなワーキング・セット・サイズの減少が発生しても、本当に業務プログラムが頻繁に使用する領域サイズ(前図の α 点)が保証されている限り、その業務プログラムに与える影響は少ないと言えます。この際の α 点のことを、業務プログラムの最小ワーキング・セット・サイズと呼びます。

最小ワーキング・セット・サイズ

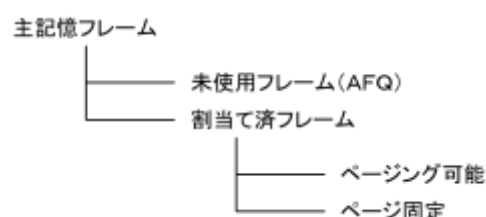
各業務プログラムには、確実に使用可能でなければ、その実行に悪影響を及ぼす主記憶域の大きさがあります。この主記憶域の大きさを、その業務プログラムの最小ワーキング・セット・サイズと呼びます。プログラム多重度が増加すると、業務プログラムの総ワーキング・セット・サイズ(図中では使用中と記述)が減少します。これは、各業務プログラムで使用可能な主記憶容量が業務プログラムの最小ワーキング・セット・サイズ以下になり始めていることを示しています。

ここで、業務プログラムの最小ワーキング・セット・サイズとそのプログラム動作について考察します。一般的に業務プログラムはループを形成しています。例えば、バッチ・プログラムは1つの入力レコードの処理を入力レコードの数だけ繰り返すループ・プログラムであると考えられます。この処理部で使用する領域が40ページであるとする、この業務プログラムの最小ワーキング・セット・サイズは160KB(=40×4KB)になります。この業務プログラムで50ページの主記憶域が使用可能であればページングは発生しません。しかし使用可能な主記憶域が40ページ以下になるとページングが多発します。ここで、この業務のプログラムは40ページの領域を順序正しく使用することを想定し、この業務プログラムが39ページの主記憶域を使用できる場合の動作を考察します。業務プログラムの実行が進み40番のページを使用している時、この業務プログラムは39ページ分の主記憶域を専有しています。それらのページは2番から40番の39ページです。40番のページの使用を完了すると、次は1番のページを使用します。1番のページは主記憶に存在しないため、1番のページはページ・インされます。しかし、この時点でこの業務プログラムが専有する主記憶域は40ページになるため、それらのページの内、最も古い2番のページが主記憶から追い出されます。1番の次は2番のページが使用されるため、同様の理由により2番のページがページ・インされると同時に、3番のページが主記憶から追い出されます。このように、犬が自分の尻尾を咬む時のように堂々巡りを繰り返します。その結果、ページングが増加すると同時にワーキング・セット・サイズも急激に減少し、業務プログラムにも悪影響を与えます。



主記憶フレームの状態

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。主記憶内に存在するページは、連続した4KBの主



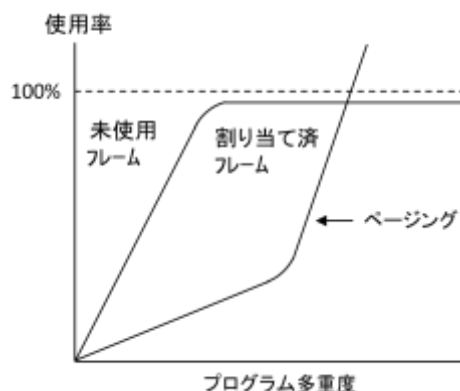
記憶域(フレーム)を専有します。ここで、ページとフレームを区別するのは、ページが4KBの有用なプログラムもしくはデータであるのに対し、フレームはページを記憶するための器であるためです。このフレームは、使用可能な主記憶域を4KB単位に区切った領域であると理解して下さい。前述のように、主記憶内に存在するページはページ固定されたページ、もしくはページング可能なページの内、頻繁に使用されるページです。これらのページは主記憶内で、1つのフレームを専有します。このように、いずれかの有効なページが記憶されたフレームのことを割当て済フレームと呼びます。一方、主記憶のフレームの中にはどのページにも専有されていないものがあります。このような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。

ページ不在割り込みとページング

オペレーティング・システムは主記憶をLRU(リスト・リセントリ・ユース)方式で管理しています。つまり、最も頻繁に使用しているページ群のみを主記憶内に残すようにしています。このため、オペレーティング・システムは最も頻繁に使用されなかったページを、順次、主記憶から追い出すようにしています。

業務プログラムが使用しようとしたページが主記憶内に存在しない場合、ページ不在の割り込みが発生し、オペレーティング・システムはそのページをページ・インしなければならないことを知ります。この際、オペレーティング・システムは目的ページをどのフレームにページ・インするべきかを決定する必要があります。ページ不在割り込みが発生した時に主記憶の全フレームを検査し、最も長時間使用されていないページが専有しているフレームを捜し出すのは容易なことではありません。このため、ページ・イン時にはAFQで管理されている未使用フレームを使用します。

プログラム多重度に比例して割当て済フレームも増加し、未使用フレームの数も減少します。しかし、同時にページング回数も増加するため、ページ・インに使用する未使用フレーム数をある一定量確保しておく必要があります。オペレーティング・システムはAFQ内にある未使用フレームの数を常時監視し、その数が一定基準値以下となると、ページング可能な割当て済フレームの内、最も長時間使用されていなかったフレームを強引に未使用フレームとします。



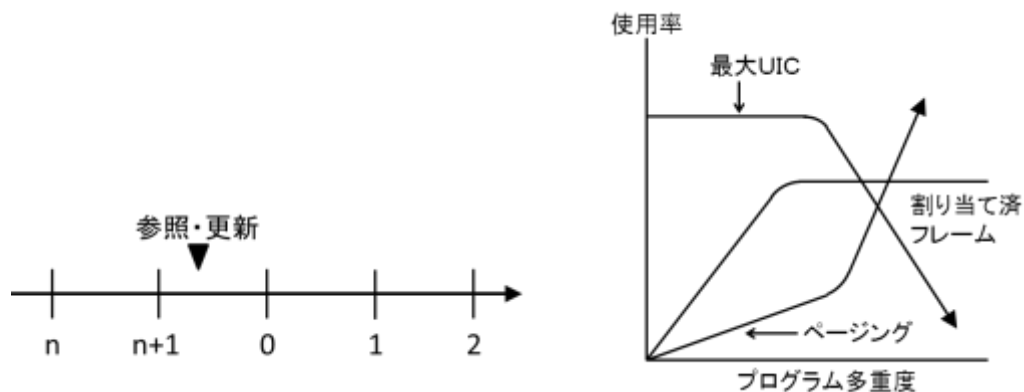
フレームの未使用时间

主記憶の未使用フレームを確保するためには、主記憶内に存在するページング可能なページ群の使用頻度を管理する必要があります。オペレーティング・システムは、主記憶のフレームごとに、この使用頻度を示すカウンタとしてUIC(アンリファレンスト・インターバル・カウント)を準備しています。

オペレーティング・システムは1秒ごとに主記憶の全フレームを検査します。もし、フレーム内に記憶されたページの内容がプログラムもしくはデータとして使用されていれば、そのフレームに対応するUICをゼロにします。もし、過去1秒間にそのフレームに記憶されたページが使用されなかった場合は、ICの値がプラス“1”されます。このUICの最大値は255です。

このように、UICはそのフレームの使用されなかった時間長(使用頻度)を示しています。

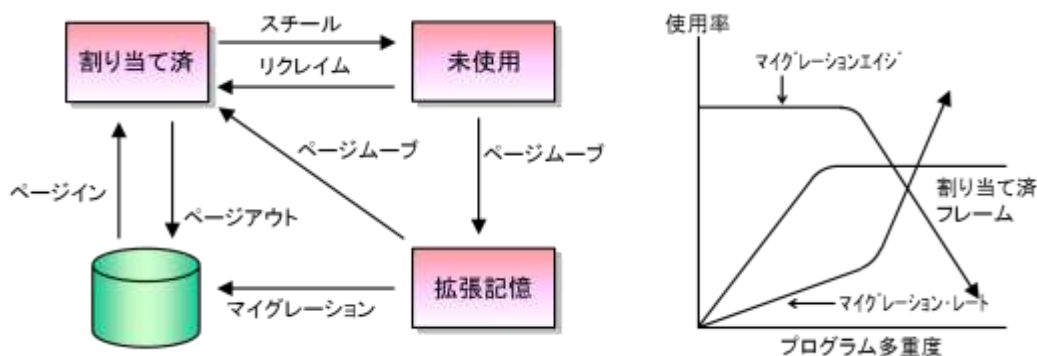
主記憶のフレームの内、最大のUIC値を持つフレームが未使用フレームに移されることはありません。このため、UICが255のフレームも存在します。しかし、未使用フレーム数が基準値以下になると、割当て済フレームが頻繁に未使用フレームへと移行させられるため、フレームの最大UIC値も減少します。つまり、主記憶の競合が増加すると、システムの最大UIC値も減少します。



拡張記憶の動作

IBMシステムには、主記憶と拡張記憶があります。ESA においては、拡張記憶と主記憶を違ったレベルで使用するためにハイパー・スペースなどの概念が導入されました。しかし、それらの機能を使用しない限り、拡張記憶は主記憶の延長であると言えます。いずれの場合においても、オペレーティング・システムは拡張記憶を主記憶と同様の手法で管理しています。

ここで、ページの状態推移について考察します。主記憶内で使用頻度の低いページが専有するフレームは強引に未使用フレームとされる場合があります。この際、そのフレームに記憶されたページの内容は拡張記憶にコピー（ページ・ムーブ）されます。もし、そのページが拡張記憶に存在している間に、ページ不在割り込みが発生し再び必要な場合、拡張記憶から実記憶へページ・ムーブされます。もし、拡張記憶の未使用フレーム数が一定基準以下になると、最も使用頻度の低いページが外部記憶へ追い出されます。この動作をマイグレーションと呼びます。このような、拡張記憶と主記憶間でのページ転送をページ・ムーブと呼びます。また、拡張記憶から外部記憶へのページ転送をマイグレーションと呼びます。（実際には、拡張記憶から外部記憶へのページ転送は主記憶を経由します。）オペレーティング・システムは拡張記憶の使用頻度を管理するためにマイグレーション・エイジと呼ばれるカウンタを使用します。このマイグレーション・エイジは主記憶のUICに相当します。

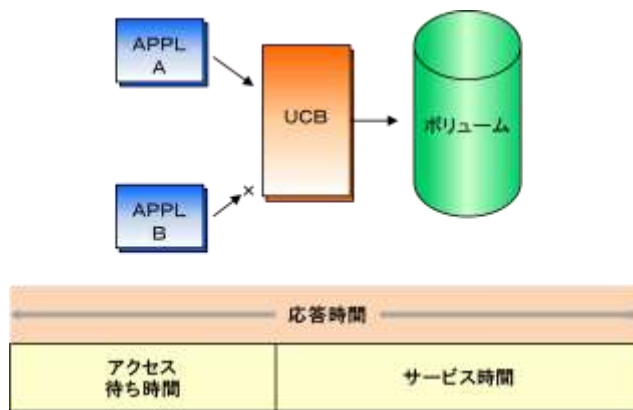


2.3 入出力サブシステムの評価

コンピュータ・システムは、プロセッサとストレージおよび入出力サブシステムにより構成されています。パフォーマンス評価を行う際、多くのシステムがオンライン指向になっているため、ディスク・ボリュームの応答時間（アクセス速度）に着目する必要があります。ディスク・ボリュームの応答時間を高速化するためには、その応答時間の内訳と負荷バランスについて理解する必要があります。

アクセス待ち時間

オペレーティング・システムはディスク・ボリューム内のスペースを効率的に使用するため、1つのディスク・ボリュームに複数のデータセットを割当てられるようにしています。業務プログラムはデータセットをアクセスする際、オペレーティング・システムの排他制御機能により、実行結果に矛盾が生じないように制御されます。例えば、2つの業務プログラムが同じデータセットを同時に更新できないようにしています。しかし、同一ディスク・ボリューム内であってもデータセットが違っていれば、そのボリュームをアクセスする業務群は同時に実行されます。



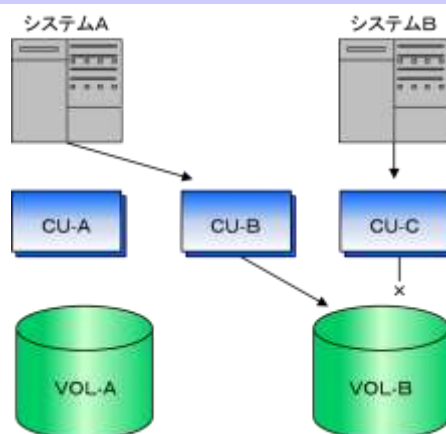
ディスク・ボリュームをアクセスするためのハードウェア機構（ヘッド群）は、ディスク・ボリュームに1つだけ用意されています。このため、同時に2つの業務プログラムが同一のディスク・ボリュームをアクセスしようとしても、その両方を同時に処理することはできません。オペレーティング・システムは、このハードウェアの制約を制御するために、入出力装置用の制御テーブルであるUCB（ユニット・コントロール・ブロック）を利用し、入出力要求の逐次化（シリアルライゼーション）制御を行います。

オペレーティング・システムは、同一ボリュームに対する複数の入出力要求を受け付けると、先に受け付けた入出力要求の処理を開始します。後に受け付けた入出力要求は、先行する入出力要求の処理が完了するまで、その処理開始が待たされます。このUCBにおける逐次化制御により生じる待ち時間のことをアクセス待ち時間と呼びます。ディスク・ボリュームの応答時間からアクセス待ち時間を除いた時間がサービス時間と呼ばれます。

アクセス・パス待ち時間とデバイス待ち時間

業務プログラムからの入出力要求が UCB での逐次化制御をパスし実行権が与えられると、オペレーティング・システムはハードウェアに指示を出し、その入出力要求を処理しようとします。しかし、入出力サブシステムはチャンネルと制御装置および入出力装置から構成されているため、それぞれの装置に起因される待ち時間が生じます。この内、チャンネルや制御装置は入出力装置とストレージ間のデータ転送を行うためのルート（アクセス・パス）を構成するものです。このため、チャンネルや制御装置が使用中であるために待たされる時間をアクセス・パス待ち時間と呼びます。

同一サイトで複数のシステムが運用される場合、それらのシステムでディスク・ボリュームを共用する場合があります。このようなディスク・ボリュームをアクセスする際、その実行結果に矛盾が生じないように、リザーブ指令を使用した排他制御を行います。例えば、一方のシステムでVTOCを更新する際、他方のシステムがそのディスク・ボリュームをアクセスできないようにリザーブ指令が実行されます。一度リザーブ指令を実行すると、次にリリース指令を実行するまで、他方のシステムからの入出力要求はその実行を拒否されます。このリザーブ指令が他方のシステムより実行されたことにより、ディスク・ボリュームがアクセスできなかった時間のことをデバイス待ち時間と呼びます。



ディスク装置の動作

オペレーティング・システムが入出力要求の起動に成功すると、ディスク装置の動作が開始されます。このディスク装置の動作には、シークと回転待ち、RPS ミスおよびデータ転送の4つの時間要素があります。シーク時間は、ディスク装置のアクセス機構（ヘッド）がディスク円盤の目的位置へ移動する時間です。また、回転待ち時間は、ディスク円盤が回転し目的レコードがヘッドの直前までやって来るまでの時間です。この回転待ち時間は、統計学上、ディスク円盤が2分の1回転する時間です。

シークと回転待ちの間、チャンネルや制御装置などのアクセス・パスは他のディスク・ボリュームが使用できる状態となっています。しかし、目的レコードがヘッドの直前に到着すると、データ転送のため再びアクセス・パスを専有しようとし、もし、その瞬間、アクセス・パスが他のディスク・ボリュームのために使用されていると、アクセス・パスが使用できないまま目的レコードがヘッドを通過することがあります。このような事態が発生すると、ディスク円盤がもう1回転するまで、データ転送が待たされます。このディスク円盤の空回りに必要な時間がRPSミス時間です。目的レコードがヘッドの直前に到着した時点でアクセス・パスが確保できれば、データ転送が開始されます。



入出力サブシステムの負荷バランス

入出力サブシステムには、数多くのディスク・ボリュームとアクセス・パスがあります。パフォーマンスを最大限に保つためには、これらのディスク・ボリュームやアクセス・パス群の負荷が均等に分散されていることが必要となります。もし、特定のディスク・ボリュームもしくはアクセス・パスに負荷が集中しますと、システム全体のパフォーマンスはそれらのディスク・ボリュームもしくはアクセス・パスの能力により左右されます。この負荷バランスは、2～3時間分のパフォーマンス・データを基に判定する必要があります。

ディスク・ボリュームの負荷バランスを判定する場合、その応答時間だけでなくアクセス回数にも着目します。このため、それぞれのビジー率ではなく、応答時間にアクセス回数を掛けた値をディスク・ボリュームの負荷指標とします。この負荷指標が高いディスク・ボリュームのチューニングを最初に行うべきです。負荷指標の小さいディスク・ボリュームをチューニングしても、それに費やす努力に比べその効果は少ないと言えます。

アクセス・パスについてもディスク・ボリュームと同じことが言えます。このアクセス・パスとは、1つのディスク系列をアクセスするルートのことです。このため、1つのディスク系列に2つのチャンネルが接続されている場合、それらのチャンネルは1つのアクセス・パスを構成します。このアクセス・パスごとの負荷がバランスしていないと、アクセス・パス待ち時間やRPSミス時間が増加します。

第3章 パフォーマンス評価データ算出式

ES/1 NEOでは、適用業務の評価を容易に実施できるように、各種の計算式を使用してワークロード・データなどを求めています。ここでは、使用する計算式を説明します。

実 CPU 時間

オペレーティング・システムを含む、全てのプログラムにおいて使用した総プロセッサの時間です。

$$\text{実CPU時間} = (\text{CPUビジー}) \times (\text{RMFインターバル}) \times (\text{CP数})$$

計測 CPU 時間

全てのパフォーマンス・グループ・ペリオッドにおける TCB モードと SRB モードにおけるプロセッサ使用時間の合計です。

$$\begin{aligned} \text{計測CPU時間} = & \sum \frac{(\text{CPUサービス・ユニット})}{(\text{CPUサービス定数}) \times (\text{プロセッサ定数})} \\ & + \frac{(\text{SRBサービス・ユニット})}{(\text{SRBサービス定数}) \times (\text{プロセッサ定数})} \end{aligned}$$

プロセッサ捕捉率 (CAPTURE RATIO)

オペレーティング・システムが行うシステム・オーバヘッドの量を把握するための計数です。

$$\text{プロセッサ捕捉率} = \frac{(\text{計測CPU時間})}{(\text{実CPU時間})}$$

アクティブ・フレーム

パフォーマンス・グループ・ペリオッドごとに実行されているプログラムが使用(参照)しているページ・フレーム数です。

$$\text{アクティブ・フレーム} = \frac{(\text{MSOサービス・ユニット}) \div (\text{MSOサービス定数})}{(\text{CPUサービス・ユニット}) \div (\text{CPUサービス定数})} \times 50$$

アクティブ・フレーム率

ストレージの実際の使用状況を管理するための係数です。

$$\text{アクティブ・フレーム率} = \frac{(\text{アクティブ・フレーム})}{(\text{ページ可能域})}$$

マルチ・プログラミング・レベル (MPL)

パフォーマンス・グループ・ペリオッドごとに実行されているプログラムの内、スワップ・イン状態にあるプログラム数です。システム全体のプログラム多重度を判定するためや、ドメインのパラメータ設定のために重要な意味を持ちます。

$$\begin{aligned} \text{MPL} &= \frac{(\text{総サービス量}) \div (\text{RMFインターバル})}{(\text{アブソープション率})} \quad \dots (\text{RMFリストよりの算出式}) \\ &= \frac{(\text{プログラム常駐時間})}{(\text{RMFインターバル})} \quad \dots (\text{SMFデータよりの算出式}) \end{aligned}$$

ワーキング・セット・サイズ (WSS)

パフォーマンス・グループ・ペリオッドごとに実行されているプログラムの1本が、常時使用(参照)していたページ・フレーム数です。

$$\text{WSS} = \frac{(\text{アクティブ・フレーム})}{\text{MPL}}$$

スワップ・アウト率

トランザクション処理時間(レスポンス時間)の内、何パーセントの時間がスワップ・アウト状態であったかを示します。レスポンス時間の評価時には、チューニング方法が変わりますので、注意して下さい。

$$\begin{aligned} \text{スワップ・アウト率} &= 1 - \frac{(\text{サービス率})}{(\text{アブソープション率})} \quad \dots (\text{RMFリストよりの算出式}) \\ &= 1 - \frac{(\text{プログラム常駐時間})}{(\text{プログラム稼働時間})} \quad \dots (\text{SMFデータよりの算出式}) \end{aligned}$$

スワップ・アウト時間

レスポンス時間内、プログラムがスワップ・アウトの状態であった時間です。

$$\text{スワップ・アウト時間} = (\text{レスポンス時間}) \times (\text{スワップ・アウト率})$$

レジデント時間 (スワップ・イン時間)

レスポンス時間内、プログラムがスワップ・インの状態であった時間です。

$$\text{レジデント時間} = (\text{レスポンス時間}) - (\text{スワップ・アウト時間})$$

CPU バースト時間

1回の入出力動作を行うために必要な、プログラムの平均プロセッサ連続使用時間です。

$$\text{CPUバースト時間} = \frac{(\text{パフォーマンス・グループの計測CPU時間})}{(\text{I/Oサービス・ユニット}) \div (\text{I/Oサービス定数})}$$

入出力動作回数

1つのトランザクション処理に必要な、プログラムの平均入出力動作回数です。

$$\text{入出力動作回数} = \frac{(\text{I/Oサービス・ユニット}) \div (\text{I/Oサービス定数})}{(\text{トランザクション数})}$$

入出力待ち時間

1回の入出力動作に必要な理論的な待ち時間です。この時間には、実際の入出力動作時間やページングによる待ち時間などを含みます。

$$\text{入出力待ち時間} = \frac{(\text{レジデント時間})}{(\text{入出力動作時間})} - (\text{CPUバースト時間})$$

装置のビジー率

チャネルや入出力装置およびページやスワップのデータセットのビジー率は、次式で求められます。

$$\text{ビジー率} = \frac{(\text{ビジー時のサンプル回数})}{(\text{総サンプル数})}$$

装置のサービス時間

入出力装置やページおよびスワップのデータセットのサービス時間は、次式で求められます。

$$\text{サービス時間} = \frac{(\text{ビジー率}) \times (\text{RMFインターバル})}{(\text{アクセス回数})}$$

装置のアクセス待ち時間

入出力装置のアクセス待ち時間は、次式で求められます。

$$\text{アクセス待ち時間} = \frac{(\text{平均アクセス待ち数})}{(\text{アクセス回数})}$$

装置のレスポンス時間

入出力装置のレスポンス時間は、次式で求められます。

$$\text{レスポンス時間} = \text{アクセス待ち時間} + \text{サービス時間}$$