

ES/1 NEO

MFシリーズ

MF-z/VM

パフォーマンス・チューニング作業

第3版 2024年 2月

©版權所有者 株式会社 アイ・アイ・エム 2024年

© COPYRIGHT IIM CORPORATION, 2024

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

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

目次

第 1 章 パフォーマンス・チューニング作業	1
1.1 本章の使用方法.....	1
PROC01n	2
PROC02n	3
STOR01n	4
IOSS01n	5
IOSS02n	7
IOSS03n	8
PAGE10n	10
WRKL10n	11
WRKL11n	12
WRKL12n	13
CONF01n & CONF02n	14
OSP04n	15

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

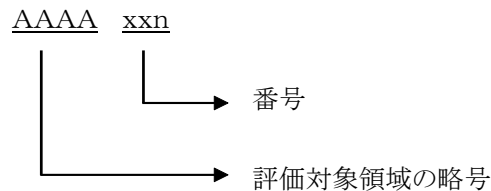
MF-z/VMでは、各種のチューニング・ヒントを出力します。しかし、MF-z/VMが出力するメッセージだけでは、その全てを説明することが困難です。この章では、MF-z/VMが出力するチューニング・ヒントに対応したチューニング作業について解説します。なお、この章は“ZVMPT00の使用方法”の章で説明されている事項を理解していることを前提で記述されています。また、チューニング作業の具体的な実施方法で疑問などがあれば、弊社担当者へお問い合わせ下さい。各種の方法でチューニング作業の支援を行います。

MF-z/VM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。

■プロセッサPROC
■主記憶STOR
■ページ・データセットPAGE
■実行効率WRKL
■拡張記憶CONF
■オペレーティング・システム・パラメータOSP

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-z/VM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>



PROC01n**【説明】**

プロセッサの使用率が高いため、重要業務の運用に遅延が発生することが危惧されます。ゲストの実行優先順位(Share)が正しく設定されていることを確認してください。

【解説】

仮想マシンがプロセッサを使用する場合、その仮想マシンに割り当てられた実行優先順位(Share)に従ってプロセッサ実行権が割り振られます。各仮想マシンの実行優先順位は、SETSHAREコマンド、もしくはユーザディレクトリでのSHAREステートメントで指定します。実行優先順位には絶対(Absolute) Shareと相対(Relative) Shareの2種類があります。絶対Shareは、使用可能なCPU能力の比率を明示します。例えば絶対Shareで3が指定されていれば、仮想マシンは3%のCPU時間(能力)を使用することができます。この際、他に幾つの仮想マシンが動作しているかを問いません。この絶対Shareは、稼働形態が判っている仮想マシンの運用に最適です。仮想マシン群に割り当てられた絶対Shareの合計値が99以上となった場合、z/VMはその合計値が99(%)になるように按分処理を行います。相対Shareは、並列稼働する他の仮想マシンと比較した相対重要度を示します。100の相対Shareを持つ2つの仮想マシンが並列稼働していれば、それぞれの仮想マシンは均等にCPU使用を行います。また、相対Shareが200の仮想マシンは、相対Shareが100の仮想マシンに比べ、倍のCPU時間を使用できます。この相対Shareは稼働状態が明確でない仮想マシンで効果を発揮します。絶対Shareの仮想マシンと相対Shareの仮想マシンが並列稼働している場合、まず絶対Shareの仮想マシンが指定量のCPU能力を使用します。そして残りのCPU能力を相対Shareに従い、相対Share指定の仮想マシンに割り振ります。指定されたShareは目標最小Shareとも呼ばれるものです。システムが疎負荷である場合、仮想マシンは指定されたShare以上にCPU能力を使用することができます。この状態のことをNOLIMITと呼びます。NOLIMITの代わりにLIMITHARDもしくはLIMITSOFTを指定することができます。LIMITHARDが指定されている仮想マシンでは、指定されたShareを超えてCPU能力を使用することができません。LIMITSOFTが指定されている仮想マシンは、残りのCPU能力を使用する仮想マシンが全くない場合に限り、指定されたShare以上にCPU能力を使用することができます。NOLIMIT、LIMITHARD、LIMITSOFTの何れも指定されていない場合、省略値としてNOLIMITが指定されたものと見なされます。

プロセッサ使用率が高い場合、重要業務が稼働している仮想マシンにおいて適切な実行優先順位が与えられていない場合、重要業務に遅延が発生する可能性があります。実行効率を調査してください。

PROC02n**【説明】**

システム(z/VM)によるプロセッサ使用率が高くなっています。このシステムによるプロセッサ使用率とは、通常、システム・オーバーヘッドと呼ばれているものです。また、実行効率も悪くなっていることが考えられます。実行効率を悪化させる要因について調査してください。

【解説】

プロセッサの使用率は、ユーザによるプロセッサの使用率とシステム(z/VM)での使用率に分けることができます。システムによりプロセッサが使用されていた部分をシステム・オーバーヘッドと呼びます。システム・オーバーヘッドが大きくなった原因として、主記憶と拡張記憶間のページムーブ数が増大していることが考えられます。ストレージの使用状況を調査してください。システム・オーバーヘッドが増大すると、実行効率の悪化が懸念されます。実行効率についても併せて調査が必要です。

STOR01n

【説明】

主記憶の競合が高くなっています。

重要業務の遅延が発生していないかを、ユーザ毎の実行効率で確認してください。

【解説】

z/VMでのメモリの仮想化はページングにより実現され、主記憶と仮想記憶、それにページングボリュームの3階層構造となっています。プログラムを実行するためには、プログラムやデータを主記憶に常駐させる必要があります。主記憶に常駐させることが必要なプログラムやデータ部のことをワーキングセットと呼び、ワーキングセット外のページ群は拡張記憶もしくはページングボリュームに格納されます。もし、ワーキングセット外のページを参照する必要が発生した場合、拡張記憶もしくはページングボリュームから必要なページをページインするという動作が発生します。この動作のためにプログラムは一旦動作を停止します。主記憶の競合が高くなるとワーキングセットが圧迫され、結果として主記憶に存在ページが増えることになり、ページングが多発する可能性があります。この場合、実行効率が悪化しますので、ユーザ毎の実行効率を調査し、重要業務の遅延が発生していないかを確認してください。尚、拡張記憶が割り当てられている場合、このチューニングヒントは出力されません。

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】

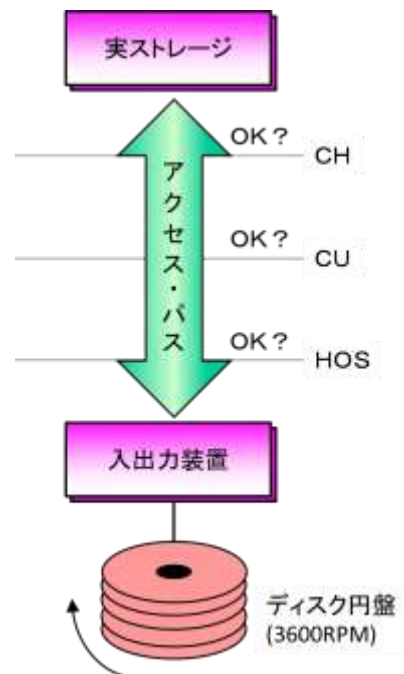
ディスク装置の応答時間は次のように分類できます。



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。

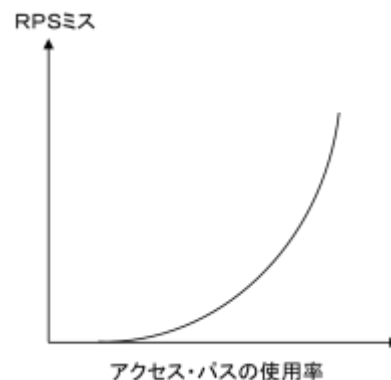


$$\begin{aligned} \text{RPSミス時間} &= RV \times (\text{1回ミスする確率}) + RV \times (\text{2回ミスする確率}) \\ &+ RV \times (\text{3回ミスする確率}) + \dots \end{aligned}$$

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

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

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。

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

【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。

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。



QUEUE	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
PEND	アクセス・パス待ち時間	チャン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
DISC	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
		$\frac{60}{\text{回転数} \times 2} \times 1000 \text{ (ミリ秒)}$
		(例) 3600RPM の時は約 8.3 ミリ秒
CONN	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



(注)
回転数については
IOSS01nを
参照して下さい。

【対応策】

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。

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。

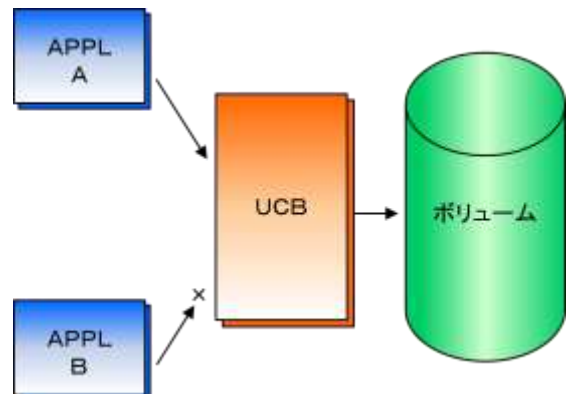
【解説】

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。

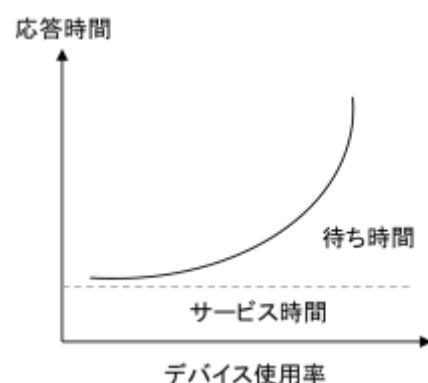


アクセス待ち時間とは、先行した入出力動作が完了するまで入出力装置の制御テーブルであるUCB(ユニット・コントロール・ブロック)で新たなアクセス要求が待たされる時間のことです。各ソフトウェア・モニタは、このアクセス待ち時間を各ディスク装置毎に時間 かもしくはキュー長でレポートします。複数の業務プログラムが同一ボリュームを全くランダムにアクセスする際の待ち時間は待ち行列技法のM/M/1の公式で求めることができます。

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



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。

PAGE10n

【説明】

ページングスペースの使用率が高くなっています。ページングスペースの拡張を検討してください。

【解説】

主記憶からページアウトされたページは、外部記憶に用意されたページングスペース(ページングボリューム)に格納されます。このページングスペースの使用率が高くなると、ページング時の競合が発生したり、スペースを確保するための時間がかかったりし、ページングのサービス時間が長くなる原因となります。そのため、ページングスペースは余裕をもって確保することをお勧めします。

ページングスペースの定義には次のようなことに気をつけてください。z/VMのページングボリュームには、ページング以外のファイルを割り当てないようにしてください。z/VMはページングの効率(ページングのスループット)を高めるために、ページングボリュームに対して連続したコマンド(I/O操作)を実行しています。ページングボリュームにミニディスクやスプールスペースを割り当てた場合、ページングによる連続したコマンド処理を一旦中断させて、それらのファイルをアクセスすることになります。このために、ページングスループットが下がってしまいます。また、ページングスペースは大きく一つのボリュームに確保するのではなく、小さくとも複数のボリュームに割り当てるのが理想です。一台のディスク装置へのアクセスは、その全てが逐次化されてしまいます。しかし、同じディスク装置が3台あれば、3つのI/O操作を平行して行うことが可能となり、そのスループットは3倍となります。複数のページングボリュームを割り当てる際、それぞれのディスク装置は、違ったアクセスパス(違ったチャネルの違った制御装置)を経由していることが望ましいです。IBMのESS(Enterprise Storage SubSystem)であれば、一つの論理ボリュームは、複数の物理ボリュームに対応します。ESSにページングボリュームを割り当てる際には、違った制御装置やアレーにボリュームが割り当てられるように留意し、競合を避けるべきです。

WRKL10n

【説明】

システム全体レベルでの実行効率が悪いと考えられます。実行効率を悪化させている要因を調査するとともに、その影響を受けているであろうと考えられるユーザを調査してください。

【解説】

実行効率は、z/VMで動作するユーザの実行状況を示すものです。該当ユーザが何ら阻害されることなく(待つことなく)プロセッサを使用できていれば、実行効率は100となります。何らかの待ち要因が発生すると、実行効率は小さな値となります。この実行効率は、次式で求めることができます。

$$\text{実行効率} = \frac{\text{プロセッサ時間} \times 100}{\text{プロセッサ時間} + \text{待ち時間}}$$

$$\begin{aligned} \text{待ち時間} = & \text{プロセッサ待ち時間} \\ & + \text{メモリ待ち時間} \\ & + \text{入出力待ち時間} \\ & + \text{その他待ち時間} \end{aligned}$$

プロセッサ待ち時間には3種類のものがあります。実行優先順位(プロセッサシェア)が低いため、より上位実行優先順位を持つユーザがプロセッサ使用を完了するのを待っている時間。この待ち時間は、CPモニタでは「CPU Wait」に報告されます。また、ユーザが特権命令などを実行した際、z/VMが命令実行のエミュレーションを行います。このエミュレーション機能の処理待ち時間も、このプロセッサ待ち時間に分類されます。このエミュレーション待ち時間は

「Simulation Wait」に報告されます。ユーザに割り当てられたシェア(Share)で許されたプロセッサ能力を使い切った際には、ユーザはLimitリストに登録されプロセッサ使用を抑制されます。このLimitリストに登録されている時間も、このプロセッサ待ち時間に分類されます。この待ち時間は「Limit List」に報告されます。メモリ待ち時間はユーザのプロセッサ使用がページ不在割り込みで中断され、目的のページが外部記憶から主記憶に読み込まれるまでの時間です。このメモリ待ち時間はCPモニタの「PageWait」に報告されます。尚、ページ不在割り込みが発生した目的ページが拡張記憶にある場合、拡張記憶から主記憶へのページ転送が行われますが、このページ転送の時間はプロセッサ待ち時間(CPUWait)に分類されるので注意が必要です。入出力待ち時間はユーザが入出力操作を実行し、その完了を待たなければならない時間です。この入出力待ち時間は「I/OWait」に報告されます。その他の待ち時間には、LINKやATTACHなどのユーザの状態を変えるCP機能を使用した場合などに生じるコンソール待ち時間などがあります。その他の待ち時間は「Console Function Wait」と「Other Wait」に報告されます。

重要ユーザの運用状況を監視し、問題が発生した時点での実行効率を調査します。実行効率が悪い場合、どのリソースでの待ち時間によるものかを確認してください。

WRKL11n**【説明】**

ローディングユーザを検出しました。このユーザが稼働する時間帯に他ユーザの実行効率が悪くなっているようであれば、運用時間帯を再考する必要もあります。

【解説】

ローディングユーザとは、ページングを多発させているユーザのことです。このユーザは単位タイムスライスあたりに5回以上のページ不在割り込みを引き起こしており、ページング資源を大量に消費していると判定されたユーザです。単位タイムスライスあたり5回とは、一台のページングデバイスをほぼ専有している状態に相当するため、このユーザが稼働している時間帯において他のユーザに影響を及ぼしている可能性があります。ローディングユーザが稼働している時間帯を確認し、その時間帯の他のユーザの実行効率を調査してください。実行効率に影響があるようであればローディングユーザの運用時間帯の再考も検討してください。実行状態にできるローディングユーザの数は、SRM(System Resource Manager)のSETSRMLDUBUFコマンドで指定できます。ここで仮想マシンが使用できるページングデバイスの割当率を指定することでローディングユーザ数を調整できます。

ローディングユーザはINDICATE QUEUES EXPコマンドで確認することができます。

WRKL12n

【説明】

リミットリストに登録されたユーザを検出しました。このユーザはしばらくの間プロセッサを使用することができなくなります。

【解説】

仮想マシンがプロセッサを使用する場合、その仮想マシンに割り当てられた実行優先順位 (Share) に従ってプロセッサ実行権が割り振られます。各仮想マシンの実行優先順位は、SET SHAREコマンド、もしくはユーザディレクトリでのSHAREステートメントで指定します。実行優先順位には絶対 (Absolute) Shareと相対 (Relative) Shareの2種類があります。絶対Shareは、使用可能なCPU能力の比率を明示します。例えば絶対Shareで3が指定されていれば、仮想マシンは3%のCPU時間 (能力) を使用することができます。この際、他に幾つの仮想マシンが動作しているかを問いません。この絶対Shareは、稼働形態が判っている仮想マシンの運用に最適です。仮想マシン群に割り当てられた絶対Shareの合計値が99以上となった場合、z/VMはその合計値が99(%)になるように按分処理を行います。相対Shareは、並列稼働する他の仮想マシンと比較した相対重要度を示します。100の相対Shareを持つ2つの仮想マシンが並列稼働していれば、それぞれの仮想マシンは均等にCPU使用を行います。また、相対Shareが200の仮想マシンは、相対Shareが100の仮想マシンに比べ、倍のCPU時間を使用できます。この相対Shareは稼働状態が明確でない仮想マシンで効果を発揮します。絶対Shareの仮想マシンと相対Shareの仮想マシンが並列稼働している場合、まず絶対Shareの仮想マシンが指定量のCPU能力を使用します。そして残りのCPU能力を相対Shareに従い、相対Share指定の仮想マシンに割り振ります。指定されたShareは目標最小Shareとも呼ばれるものです。システムが疎負荷である場合、仮想マシンは指定されたShare以上にCPU能力を使用することができます。この状態のことをNOLIMITと呼びます。NOLIMITの代わりにLIMITHARDもしくはLIMITSOFTを指定することができます。LIMITHARDが指定されている仮想マシンでは、指定されたShareを超えてCPU能力を使用することができません。LIMITSOFTが指定されている仮想マシンは、残りのCPU能力を使用する仮想マシンが全くない場合に限り、指定されたShare以上にCPU能力を使用することができます。NOLIMIT、LIMITHARD、LIMITSOFTの何れも指定されていない場合、省略値としてNOLIMITが指定されたものと見なされます。LIMITHARDが指定されているユーザで、割り当てられたプロセッサ能力を使い切った場合、ユーザはリミットリストに登録されプロセッサ使用を抑制されます。リミットリストに登録されている時間は、ユーザの実行効率を調査することで確認できます。重要な業務がリミットリストに登録されると業務に影響が出る可能性がありますので注意してください。ユーザのリミットリストに登録されている時間が長い場合、そのユーザの実行優先順位 (Share) を再考してください。

CONF01n & CONF02n

【説明】

拡張記憶が割り当てられていません。あるいは拡張記憶の大きさが主記憶に比べて適切でないと考えられます。z/VMは拡張記憶を利用したページングを行うように、チューニングされています。利用可能な物理メモリの4分の1は、拡張記憶として定義してください。

【解説】

z/VMでのメモリの仮想化はページングにより実現され、主記憶と仮想記憶、それにページングボリュームの3階層構造となっています。プログラムを実行するためには、プログラムやデータを主記憶に常駐させる必要があります。主記憶に常駐させることが必要なプログラムやデータ部のことをワーキングセットと呼び、ワーキングセット外のページ群は拡張記憶もしくはページングボリュームに格納されます。拡張記憶は高速なページングボリュームとして取り扱われ、拡張記憶が満杯になるとアクセス頻度の低いページ群がページングボリュームに移されます。尚、拡張記憶はオプションであり、そのサイズも任意に決定できます。拡張記憶が構成されていない場合、z/VMは主記憶とページングボリュームの間でページング操作を行うことになります。

仮想メモリは、多くのプログラムやデータを実記憶に常駐させようとするものです。ある一瞬を考えると、プログラムは限定されたメモリ域をアクセスするだけです。数秒間を考えると、プログラムは割り当てられたメモリ域のほんの一部をアクセスしているに過ぎません。仮想メモリシステムはページングにより実現されており、その仕組みは、頻繁にアクセスされているワーキングセットを実メモリに配置し、それ以外のページ群を外部記憶に保管するというものです。このような制御を行うことにより、実メモリは他のプログラムが頻繁にアクセスするページ群のために空けておくようにします。ユーザプログラムが直接アクセスできるのは、実メモリに常駐しているページのみです。また、I/O操作を行うためのバッファ域も実メモリに常駐していなければなりません。ここで言う実メモリとは、z/VMではPR/SMが提供する連続した物理メモリ(主記憶)のことです。

主記憶と拡張記憶はいずれも物理メモリの一部です。z/VMでは、この拡張記憶をページング処理にのみ利用しています。拡張記憶を割り当てると物理メモリが減少し、主記憶に割り当てるメモリ量も減少します。また、主記憶サイズに比べメモリ使用要求が増えれば、ページングは発生します。つまり拡張記憶を構成すると、ページングが発生する確率は高くなるといえます。しかしz/VMでメモリをオーバーコミットするのは通常の設定であり、z/VM自体がページングを許容するシステムです。ページングをなくすのではなく、最適なページング処理を許すシステム構成を採用すべきであるといえます。またz/VMは、拡張記憶からページングボリュームへ効率的なページ転送が行えるようにチューニングされています。つまり、主記憶とページングボリュームだけを利用したページング操作よりも、拡張記憶を使用するほうが効率的なページング処理が実現されます。まずはz/VMで使用可能な物理メモリの4分の1(25%)を拡張記憶として定義し、システム運用状況を監視しつつチューニングしてください。

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)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、ソフトウェアモニタは、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。