

---

---

# メインフレームのメタボ化とその診断手法

株式会社アイビスインターナショナル

---

## ■ 執筆者 Profile ■



有賀 光浩

1985年 富士通株式会社入社  
2003年 富士通株式会社退職  
2004年 株式会社アイビスインターナショナル  
設立 代表取締役

## ■ 論文要旨 ■

富士通メインフレーム GS の「メタボ化」が進行し、長年築いてきたお客様の信頼を損なう危機に直面している。

GS のメタボとは、外のメタボ（過剰投資）と内のメタボ（リスク）によりシステムが膨張している状態と定義する。外のメタボとは必要以上に大きなハードウェアを導入している状態、内のメタボとはソフトウェアが必要以上のリソースを使用、またはシステムの信頼性、安定稼動を阻害する可能性がある状態をいう。

従来の性能評価では GS のメタボは検出できず、新しくメタボ診断の手法を開発した。特長は CPU と I/O のバランス（CPU/I/O 頻度）の診断、及びメモリの使用状況によるシステム特性の診断である。これを 6 つのシステム例を使い考察した。診断はあくまで手段であり、お客様によりよい価値を提供できるよう改善するのが目的である。

メタボに関するリテラシーを高め、GS の満足度向上に取り組んでいる。

## ■ 論文目次 ■

<b>1. はじめに</b> .....	《 3》
1. 1 当社の概要	
1. 2 背景	
<b>2. GSのメタボ化</b> .....	《 5》
2. 1 GSのメタボの定義	
2. 2 GSのメタボの例	
<b>3. 性能評価の限界</b> .....	《 7》
3. 1 従来の性能評価例	
3. 2 右肩下がりの性能評価	
<b>4. CPUとI/Oのバランスと診断手法</b> .....	《 9》
4. 1 CPUバウンド・I/Oバウンドとは	
4. 2 CPU/I/O頻度という考え方	
4. 3 CPU/I/O頻度による診断	
4. 4 CPU使用率との関係	
<b>5. メモリによる診断手法</b> .....	《 13》
5. 1 従来の性能評価	
5. 2 仮想メモリによる診断	
5. 3 実メモリによる診断	
<b>6. サービスレベルの把握</b> .....	《 16》
<b>7. 考察</b> .....	《 17》
7. 1 メタボ診断の効果	
7. 2 改善へのアプローチ	
<b>8. 評価と課題</b> .....	《 19》

## ■ 図表一覧 ■

<b>図1</b> 国内メインフレーム出荷実績と予測 .....	《 3》
<b>図2</b> GSのメタボ化 .....	《 5》
<b>図3</b> CPU使用率とPAGE IN回数 .....	《 7》
<b>図4</b> CPU/I/O頻度の例 .....	《10-11》
<b>図5</b> CPU/I/O頻度とCPU使用率の例 .....	《11-12》
<b>図6</b> GSの仮想空間 .....	《 13》
<b>図7</b> 仮想メモリによる診断 (VS指数1) .....	《13-14》
<b>図8</b> 仮想メモリによる診断 (VS指数2) .....	《 14》
<b>図9</b> 実メモリによる診断 .....	《 15》
<b>図10</b> 実メモリによる診断 (RS指数) .....	《 15》
<b>図11</b> オンライン業務の稼動状況 .....	《 16》
<b>図12</b> GSのメタボ診断 (参考) .....	《 17》
<b>表1</b> 性能関連のトラブル例 .....	《 5》
<b>表2</b> CPU/I/O頻度 .....	《 9》
<b>表3</b> メタボの判定結果 .....	《 17》

# 1. はじめに

## 1. 1 当社の概要

株式会社アイビスインターナショナル（所在地：東京都）は、富士通メインフレーム（GS21, PRIMEFORCE）の性能コンサルティングを事業としている（ホームページ：<http://www.ibisinc.co.jp/>）。

代表の有賀光浩(Ariga Mitsuhiro)は富士通株式会社で18年間SEとして活躍、1992年からの11年間は共通技術部門でメインフレームの性能に関する技術支援、顧客システムの性能トラブル対応を担当した。対応システム数は国内外合わせて1,000以上に及ぶ。

富士通退職後、2004年に株式会社アイビスインターナショナルを設立。お客様が満足感を持ち、安心してメインフレームを使い続けて頂くためのコンサルティングを行っている。

## 1. 2 背景

### (1) メインフレームの国内市場

(社)電子情報技術産業協会(JEITA)の調査によると、2007年度のメインフレームの出荷台数は713台(前年度比82%)、金額は1,658億円(前年度比92%)であった(図1)。10年前の実績と比較すると出荷台数は25%、出荷金額は20%まで減少している。

稼働台数は発表されていないが、出荷実績から算出すると2007年度は約6,200台と予想する。毎年約10%減少しており、2000年以降この流れは大きく変わっていない。

メーカーではIBMがメインフレームの機能強化に最も力を入れている。富士通も新機種の出荷は2010年度と2014年度頃と発表している。

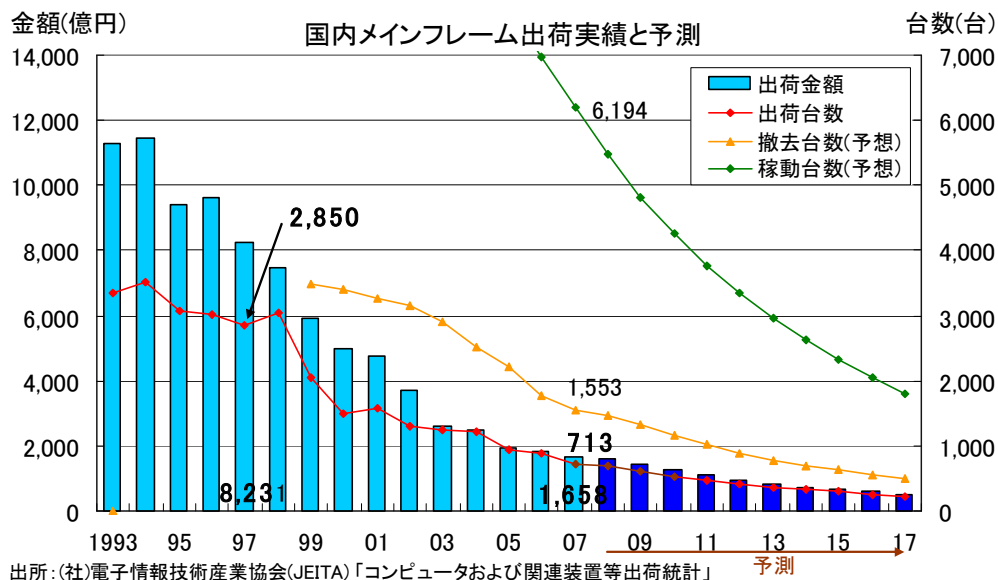


図1 国内メインフレーム出荷実績と予測

### (2) 直面したトラブル例

図1から撤去されるメインフレームの半数近くは再びメインフレームにリプレースされていることがわかる。富士通メインフレーム(以降GSと略す)を使い続ける理由としては

以下のものが考えられる。

- ・ハードウェアの高信頼，高品質
- ・安定した性能
- ・アプリケーション資産の継承

GS 本体が壊れたという話しは最近聞かないが，業務アプリケーションの暴走や運用ミスによるトラブルはどうだろうか．表 1 は私自身が性能コンサルティングの現場で直面したトラブル例である（2006 年度後期論文で紹介）．いずれも，社会システムや基幹システムであり，数ヶ月間適切な対策がとられていないものもあった．この例を基盤 SE 向けのセミナーで紹介したところ特に違和感はないようだった．

表 1 性能関連のトラブル例

顧客	機種	現象	状況，原因など
A	GS21 400	システム停止	オンラインプログラムの異常処理でループし，ログファイル（HLF）がパンクした
B	PRIMEFORCE	一部業務の処理遅延	オンラインプログラムがループし CPU 負荷 100%が 1 時間以上，1 ヶ月に 3 回発生した
		データベースのリカバリ不可	ログ環境を毎朝初期化してリカバリ不能状態だった（運よく問題は発生しなかった）
D	PRIMEFORCE	無駄なメモリ増設	メモリに十分余裕があるのに，性能改善のためにメモリを増設した
E	GS21 400	オンライン業務の一時停止	16MB 以下の実メモリが枯渇し REGION 不足でジョブが異常終了した
F	GS21 600	一部業務の処理遅延	オンラインプログラムがループし CPU 負荷 100%，1 時間後にキャンセル，1 ヶ月に 2 回発生した
G	GS8900	オンライン業務の一時停止	ECSA 領域が不足しジョブが異常終了した，適切な対応がとれず運用制限中

注) 顧客 C は意図的にとばしている

### (3) GS 特有の問題点

GS を取り巻く環境は他のサーバとは異なっている．

- ・業務量が減っている
- ・10～20 年前の業務プログラムが動いていることがある
- ・ISAM, NDB から RDB まで様々な（広義の）データベースを使っている
- ・他の GS システムと比較されない，情報が共有されない

これらを意識せずキャパシティ管理を行うと以下のような問題が発生する．

- ・一般的なキャパシティ管理は業務量の増加が前提である．業務量が年々減少するような環境は考慮されておらず適切な判断や対応がとれない．
- ・少量のリソースしか使わない 20 年前のプログラムと，CPU もメモリも使いたい放題の最近のプログラムが同じマシン上で動作している．これらを意識せずに評価するのは極めて乱暴である．
- ・評価の基準となる指標値が長年見直されていない．

表 1 のようなトラブルの予兆として「GS のメタボ化」が進んでおり，この状況を適切にキャッチアップできればトラブルの未然防止やコスト削減に有効であると考えた．本論文では，GS のメタボ化の定義とその診断手法や課題について論述する．

なお，GS の専門用語は最終ページにまとめて説明しているので活用して頂きたい．

## 2. GSのメタボ化

### 2.1 GSのメタボの定義

GSのメタボとは、図2中央「メタボな状態」に示すように、外のメタボと内のメタボによりシステムが膨張している状態と定義した。各々について説明をする。

#### (1) 外のメタボ（過剰投資）

必要以上に大きなハードウェアを導入している状態を「外のメタボ」と言う。

図2左「理想形」に示すように、システム導入時には機能実現に必要なリソース量( $r$ とする)を見積もり、そこからハードウェアの大きさ( $x$ とする)を決定する。リソースにはCPU、メモリ、I/O装置等があるが以下はCPUを例にして説明する。

GSはWebシステムと異なり業務量を予測しやすいので、CPUの大きさ $x$ は余裕をみても $r < x < 2r$ くらいになる。(2 $r$ とは余裕を見て2倍の大きさにする意味)

4～5年ごとにCPUがリプレースされると、次期システムの能力は業務量の増加がなくても1.3 $x$ 前後となる。リプレースを2度行えば1.7 $x$ 、3度行えば2.2 $x$ となる。丁度よい大きさのCPUモデルがないときには1.3倍以上になることもある。

業務が追加されると、その分のレベルアップは検討されるが、業務が削除されてもCPUをダウングレードすることはほとんどない。

業務量が増加しているときはこのやり方でも良かったが、減少傾向になると必要なCPU量 $r'$  ( $< r$ )が減り、 $r'$ と1.3 $x$ 、1.7 $x$ 、2.2 $x$ 、…との差が拡大する。

実際、CPUを30%程度しか使っていないシステムも珍しくない。GSはオープン系のサーバとは違い、CPU使用率が100%に近づいても安定して稼動することが真の強みである。

#### (2) 内のメタボ（リスク）

ソフトウェアが必要以上のリソースを使用する、またはシステムの信頼性、安定稼動を阻害する可能性がある状態を「内のメタボ」と言う。

品質の悪い業務プログラムを例にすると、機能実現に必要なリソース量 $r$ に対し、実際には $R$  ( $2r < R$ , 2倍以上)のリソースを使っていることがある。リソースの無駄遣いかつ性能の悪化である。これは、単体レベルの性能検証が不十分であることが根本的な原因である。稼動後にこのような問題を検出することは困難だったが、4章で述べるCPU/I/O頻度によりその検出が可能となった。信頼性の根幹であるAIMのログ設計に不備のあるシステムも多いのが実態である。

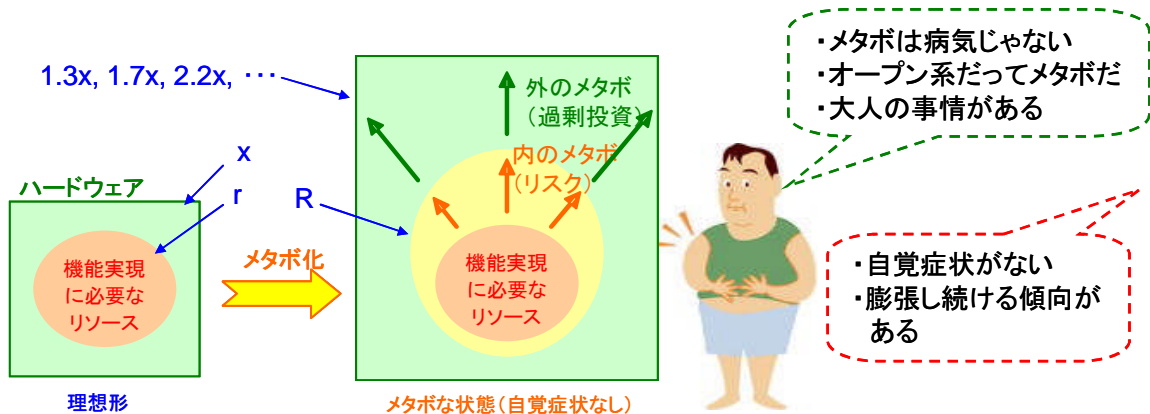


図2 GSのメタボ化

## 2.2 GSのメタボの例

表1のトラブル例を使い、GSのメタボと思われる症例とその背景を紹介する。

### 【外のメタボの症例】

- ・性能上、運用上効果のないハードウェアを増強した【D】  
⇒ 評価や見積りをせずに、遅いからCPUやメモリを増強する
- ・有効に使われていないリソースがある【D】  
⇒ メモリ、システム記憶、ディスクキャッシュなどを搭載しすぎている
- ・CPUがオーバスペックである【表1では見えず、後述】  
⇒ 見積りをせずに、機械的にCPUをレベルアップする

### 【内のメタボの症例】

- ・重い業務プログラムが動くと、リソースを使い切ってしまう【B, F】  
⇒ CPUループに気づかない、問題プログラムを特定できない
- ・大きなリスクに気づかないままシステムを運用している【E, G】  
⇒ システムやAIMの設計が不十分、リカバリテストを最近実施していない
- ・遅い処理に慣れてしまっている【表1では見えず、後述】  
⇒ 常識的に遅すぎるオンライン業務、バッチ処理を使い続けている

GSのメタボは人のメタボと似ていて「病気」ではない。即ち、直ぐにトラブルを誘発するものではないが、時間とともに発病する可能性が低くなることもない。

オープン系サーバの平均CPU使用率は20~30%と言われており、GSよりもはるかにメタボ（稼働率が低い）である。これも食文化や生活習慣の違う日本人と欧米人との差に似ている。CPU使用率が90~100%でも基幹業務を運用できる優れたハードウェアとOSを使っているのに、30%程度のパワーしか使っていないのは宝の持ち腐れであり、無駄なIT投資と言える。

また、業務プログラムは、OSやAIMの規約に準拠し、無駄にCPUやメモリを使わないこと（性能品質がよいこと）が基本である。性能品質の悪化については、2006年前期の論文から警告し続けている。

GSのメタボも自覚症状は薄く、トラブルが起きて初めて気づくものである。人と同じように、毎日体重や体脂肪率を測定する、定期的に検査をし、専門家に評価してもらう心がけが大切である。

メタボを放置しておくと膨張はするがスリムになることはない。

GSがこれからも価値を生み続け、進化していくためにはシステムが健康であることが重要だと考える。

従来の性能評価でGSのメタボは検出できるだろうか。

### 3. 性能評価の限界

#### 3.1 従来の性能評価例

図3は、6つのシステム（A～F）のCPU使用率（棒グラフ）とPAGE IN回数（折れ線グラフ）を表している。このグラフから何が読み取れるだろうか？

- ・ AとCは継続してCPU 負荷が高い
- ・ BとFは夜間のCPU 負荷が高い
- ・ DとEは突発的にCPU 負荷が高い
- ・ CとEはページングが発生している

いずれも「現在 CPU やメモリが不足しているとは言えないが継続監視が必要」くらいの結論になるのではないだろうか。これでは、前章で述べたメタボであるか否かの判断がむずかしい。

【補足】・仮想空間 [システム共通域]（図6 参照）の定義値は19%以上の余裕がある

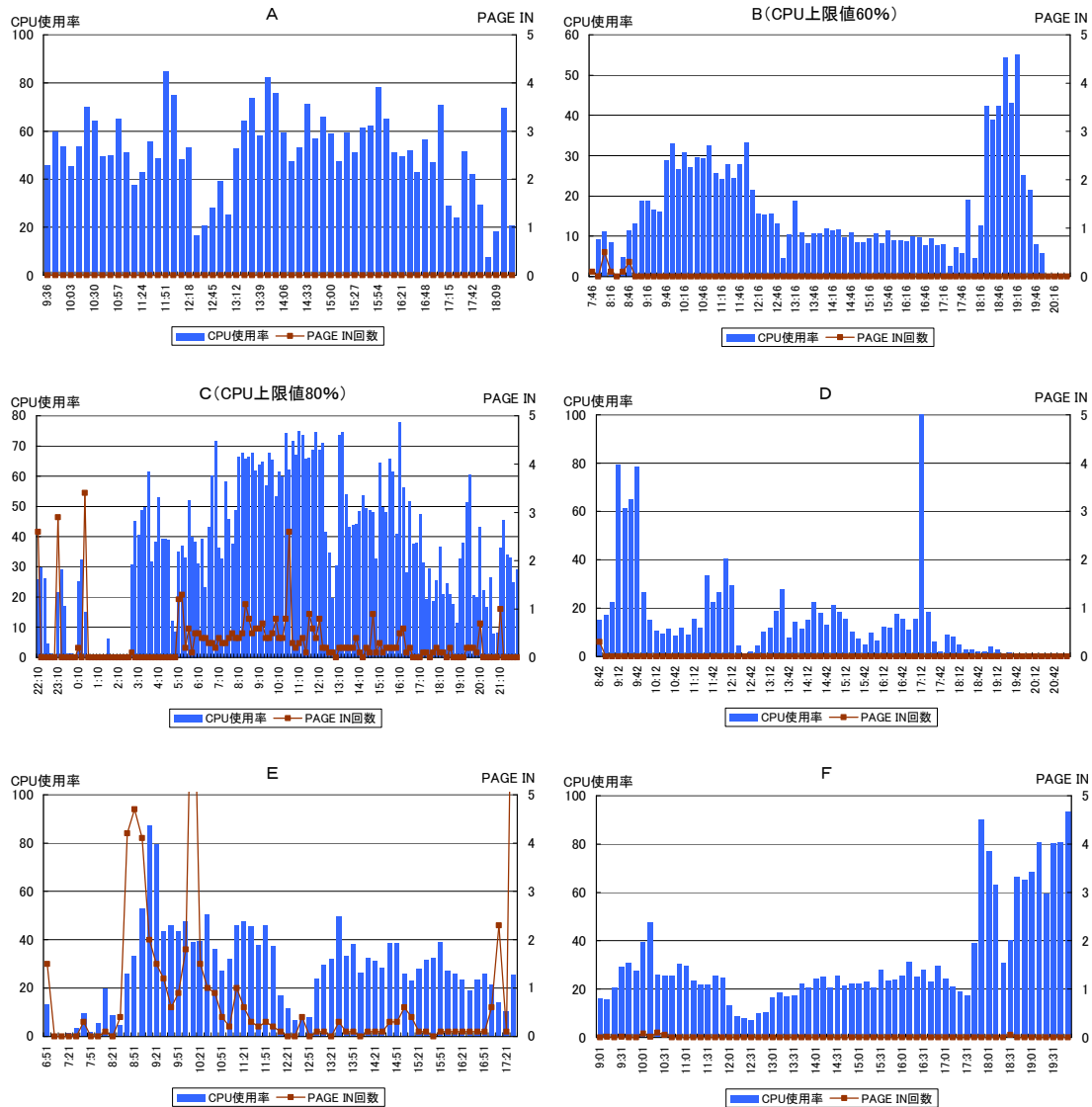


図3 CPU使用率とPAGE IN回数

### 3. 2 右肩下がりの性能評価

一般的なキャパシティ管理は業務量が増加することを前提としている。人でたとえると日々右肩上がり成長している（消費カロリーが上昇傾向）青年期で、食事でも質より量を重視する傾向がある。

一方、多くのGSはオープン化などにより業務量は減少傾向である。たとえると右肩下がり（消費カロリーが下降傾向）の中年期で、食事は量より質を重視すべきである。

若いときと同じものと量を食べていれば「摂取カロリー>消費カロリー」となり、余りが内臓脂肪として蓄積されメタボに近づいていく。

人のメタボ云々は個人の自由だが、企業や公共の資産であるGSがメタボであることは望ましくなく、その防止のためにも「右肩下がりの性能評価」への変革を提言する。

何を換えればよいのか。例えば、今までの性能評価では評価項目の実測値と指標値を比較をすることが多いが、指標値の根拠が明確でなかったり、昔から変わっていないことはないだろうか。

従来の指標値は右肩上がり前提で、リソース不足やトラブルを起こさないための限界値でありメーカ主導の視点であった。

これを、右肩下がり前提として、過剰投資の防止や性能品質維持・向上を目的としたお客様の視点に変える必要がある。（後述のメタボ診断では当然考慮している。）

具体例として以下の2つの状況をどのように評価するか考える。

- |  |
|--|
| ①平均 CPU 使用率が 90%である。<br>②平均 CPU 使用率が 20%である。 |
|--|

右肩下がりの性能評価では次のようなコメントが想定される。

①について

- CPU の稼働率が高く（90%）良好である。
- 無駄に CPU を使っていないか調査する必要がある。
- 業務に影響が出ていないことを確認する必要がある。

②について

- CPU の稼働率が低く（20%）無駄な投資をしている可能性がある。
- CPU ダウングレードの検討をすべきである。
- CPU に余裕があるのに業務の性能に不満がないか確認する必要がある。

簡単に CPU ダウングレードと書いたが、どんなに稼働率が低くても CPU をダウングレードすれば CPU 時間が延びるため処理時間は必ず遅くなる。「あなたが使っているノート PC やサーバのスペックを下げてもいいですか？」と言われれば私はダメと答えるだろう。理由は、今も遅いと感じることがあるし、更にもっと遅くなるのか想像がつかないためである。

GS では、今遅い処理は改善できるし、CPU をダウングレードしたときの性能も精度よく予測することができる。自信を持って CPU ダウングレードを提案できるサーバは世界中で GS だけと言えるかもしれない。

具体的にどのように診断をすればよいのか、次章以降で紹介する。



## 4. CPU と I/O のバランスと診断手法

まず、CPU 使用率の妥当性をできる限り客観的に評価する必要がある。

GS では DISK 上のデータをメモリに貼り付ける（常駐させる）ことは基本的に行わず、チャンネル経由で DISK の I/O が発生する。今までの経験から「GS 上で動作する業務プログラムは CPU と I/O のバランスに規則性がある」という仮説を立てた。

### 4. 1 CPU バウンド・I/O バウンドとは

一般的に CPU の依存度が高い処理を CPU バウンド、I/O の依存度が高い処理を I/O バウンドと言う。なんとなくイメージはできるが具体的にはどのように定義されているのか、MSP システムの例を使って説明する。

#### 【例 1】

CPU : GS21 600/40, GS21 600/20S, GS8800/30A (超大型機)  
OS の設定値 : MINOR=100, WLOW=2, WHIGH=2000

ある空間が CPU を 100ms (=MINOR 値)使い切ったとき入出力要求の待ちが 2 回 (=WLOW 値)以下のとき CPU バウンド、2000 回 (=WHIGH 値)以上のとき I/O バウンドとなる。超大型機で待ちが 2 回しか発生しないのは CPU ループであり 2000 回は少し多すぎる。実際にはほとんどが CPU バウンドでも I/O バウンドでもない準 CPU バウンドと判定されてしまう。

#### 【例 2】

CPU : GS8300/10M (中型機)  
OS の設定値 : MINOR=100, WLOW=2, WHIGH=7 または 200

中型機であっても待ちが数 10 回起きるのが普通であり、WHIGH 値が 7 のときはほとんどが I/O バウンド、200 のときは準 CPU バウンドと判定されてしまう。

OS の設定値は CPU 移行後に見直すことはほとんどなく、結果的にも技術的な根拠は消滅している。CPU バウンド、I/O バウンドとは気分で口にしてるのが実態であろう。

### 4. 2 CPU/I/O 頻度という考え方

性能評価の見える化ツールとして 2006 年度前期論文に「CPU/I/O 頻度分析」を紹介した。その後 100 を超えるデータを使ってブラシュアップし、CPU と I/O のバランスを表す新しい指標値としての CPU/I/O 頻度を紹介する。これは、性能データである PDL/PDA を使い、CPU と I/O のバランスの標準が 20~40 になるように独自の手法で算出している(表 2)。20 以下だと I/O 頻度が高い (I/O バウンド)、40 以上だと CPU 頻度が高い (CPU バウンド)、80 以上だと CPU ループである。

表 2 CPU/I/O 頻度

CPU/I/O 頻度	0~20	20~40	40~80	80 超
傾 向	I/O 頻度が高い (I/O バウンド) 瘦せぎみ	標 準	CPU 頻度が高い (CPU バウンド) メタボぎみ	CPU ループ (CPU バウンド) メタボ

注) CPU と I/O のバランスを表す一つの指標値として CPU/I/O 頻度を紹介している。他の指標値を使っても何ら問題ない。

### 4.3 CPU/I/O 頻度による診断

図4は横軸がCPU/I/O頻度、縦軸がIOPSの分布図である。一つの点は約10分間の測定値で、午前・午後・夜間などで分類している。この図を使って解説する。

(1) A (標準) ~ バランスの良いシステム

- ・平均 I/O 頻度 : 29, 0~20(8%), 20~40(85%), 40~80(7%), 80 超(0%)
- ・平均 IOPS : 687, 0~100(2%), 100~200(4%), 200~1000(78%), 1000 超(11%)

(2) B (バラバラ) ~ バランスの悪いシステム

- ・平均 I/O 頻度 : 23, 0~20(31%), 20~40(42%), 40~80(27%), 80 超(0%)
- ・平均 IOPS : 391, 0~100(19%), 100~200(36%), 200~1000(42%), 1000 超(2%)

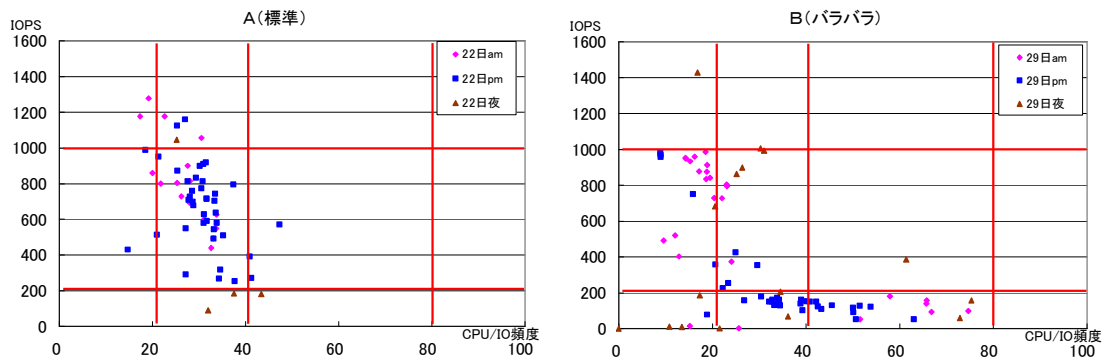


図 4 - 1 CPU/I/O 頻度の例 ( 1 )

(3) C (I/O 頻度が高い) ~ 全体的に I/O 頻度の高いシステム

- ・平均 I/O 頻度 : 11, 0~20(99%), 20~40(1%), 40~80(0%), 80 超(0%)
- ・平均 IOPS : 511, 0~100(14%), 100~200(6%), 200~1000(79%), 1000 超(1%)

(4) D (CPU 頻度が高い) ~ 全体的に CPU 頻度の高いシステム, IOPS が小さい

- ・平均 I/O 頻度 : 56, 0~20(0%), 20~40(15%), 40~80(65%), 80 超(20%)
- ・平均 IOPS : 28, 0~100(99%), 100~200(1%), 200~1000(0%), 1000 超(0%)

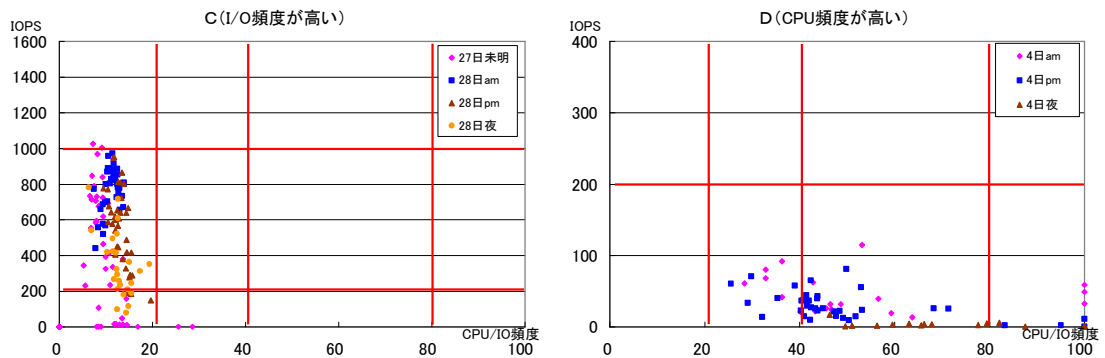


図 4 - 2 CPU/I/O 頻度の例 ( 2 )

(5) E (バラバラ) ~ バランスの悪いシステム

- ・平均 I/O 頻度 : 45, 0~20(11%), 20~40(16%), 40~80(73%), 80 超(0%)
- ・平均 IOPS : 334, 0~100(8%), 100~200(9%), 200~1000(82%), 1000 超(2%)

(6) F (夜間 CPU 頻度が高い) ~夜間に CPU 頻度が高くなるシステム

- 平均 I/O 頻度 : 45, 0~20(0%), 20~40(58%), 40~80(39%), 80 超(4%)
- 平均 IOPS : 881, 0~100(0%), 100~200(1%), 200~1000(71%), 1000 超(28%)

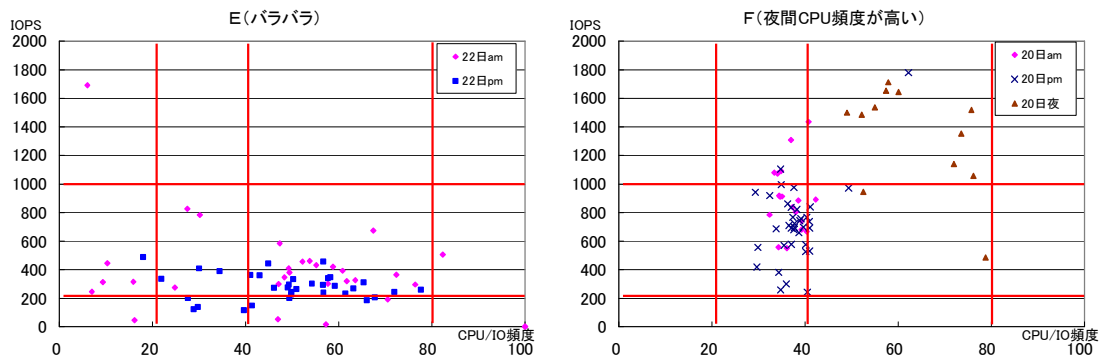


図 4 - 3 CPU/I/O 頻度の例 (3)

#### 4. 4 CPU 使用率との関係 (CPU の適正性)

図 5 は, 図 4 の縦軸を IOPS から CPU 使用率に変えたものである. CPU/I/O 頻度と一緒に  
 見ることで, CPU 使用率のバラツキと妥当性を評価することができる. 図 3 のグラフと合  
 わせて見て頂きたい.

(1) A (標準) ~平均的な CPU 使用率のシステム

- 平均 CPU 使用率 : 52%, 0~20(5%), 20~40(15%), 40~80(77%), 80~100(3%)

(2) B (バラバラ), AVM 上限値 60%~CPU の使用率が低いが高負荷もある

- 平均 CPU 使用率 : 16%, 0~12(55%), 12~24(18%), 24~48(24%), 48~60(3%)

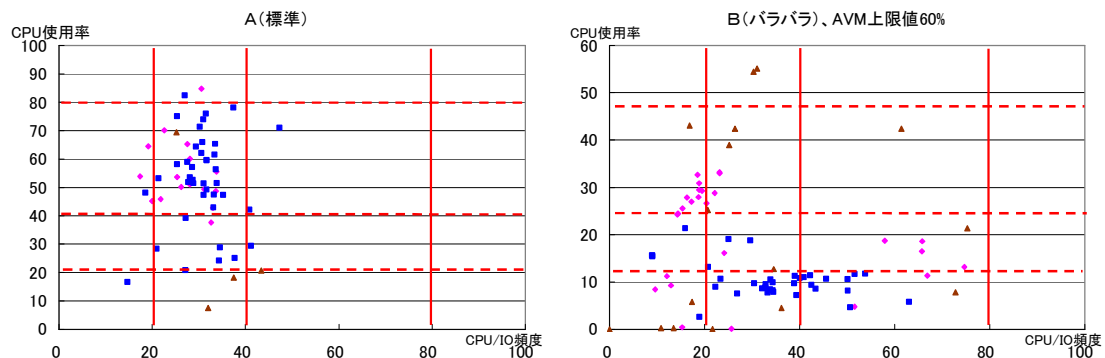


図 5 - 1 CPU/I/O 頻度と CPU 使用率の例 (1)

(3) C (I/O 頻度が高い), AVM 上限値 80%~CPU 使用率が少し高めのシステム

- 平均 CPU 使用率 : 38%, 0~16(17%), 16~32(21%), 32~64(44%), 64~80(17%)
- ジョブの多重度により CPU 使用率が変動する

(4) D (CPU 頻度が高い) ~CPU の使用率が低いが高いときもある

- 平均 CPU 使用率 : 15%, 0~20(80%), 20~40(12%), 40~80(7%), 80~100(1%)
- CPU ループするプログラムが動くと CPU が高負荷となっている

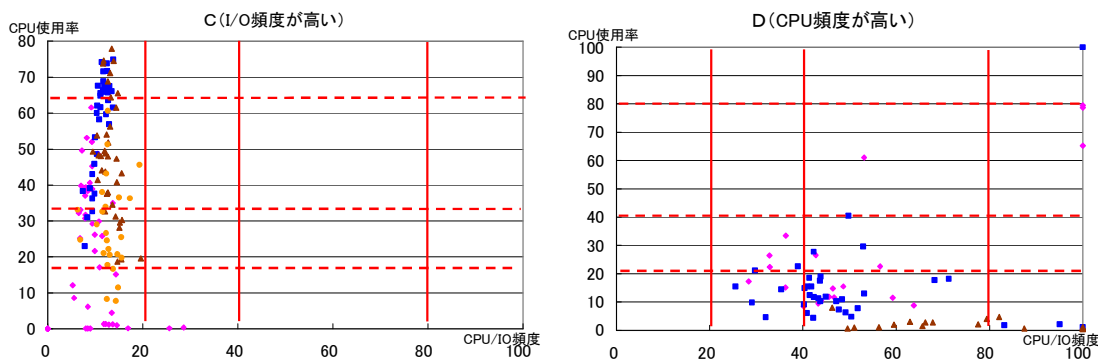


図 5 - 2 CPU/I/O 頻度と CPU 使用率の例 ( 2 )

( 5 ) E ( バラバラ ) ～平均的な CPU 使用率のシステム

- ・ 平均 CPU 使用率 : 29%, 0～20 (29%), 20～40 (51%), 40～80 (18%), 80～100 (2%)

( 6 ) F ( 夜間 CPU 頻度が高い ) ～夜間に CPU 使用率が高くなるシステム

- ・ 平均 CPU 使用率 : 32%, 0～20 (28%), 20～40 (52%), 40～80 (14%), 80～100 (7%)

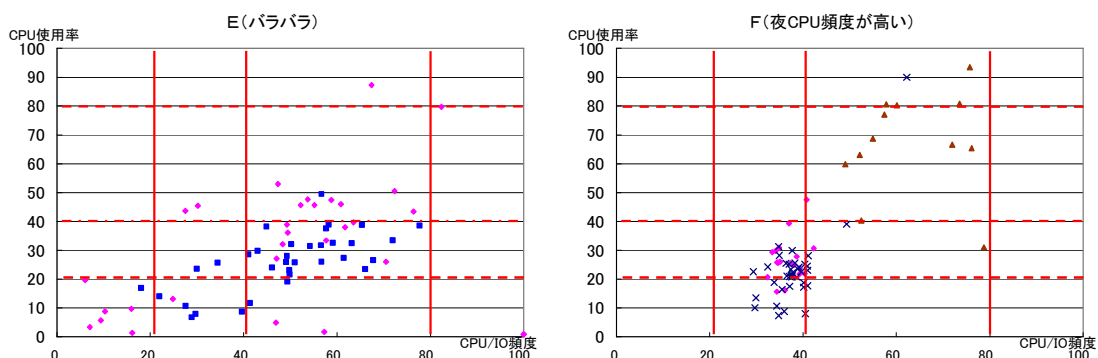


図 5 - 3 CPU/I/O 頻度と CPU 使用率の例 ( 3 )

CPU と I/O には興味深い関係があることがわかった。A のように CPU/I/O 頻度が 20～40 の間に 70%以上あればバランスがよいといえる。

個々の調査結果だが、B、E、F は複数の業務が全く異なった環境で開発され動いているため、各業務の特徴が出ていることがわかった。I/O 頻度の高い業務と CPU 頻度の高い業務が同時に動くと標準に見えてしまうので注意が必要である。

C は ISAM を多用しているため I/O の頻度が高かった。または、性能が出すぎている可能性もある。D は SymfoWARE をメモリ常駐しているため CPU 頻度が高くなった。こちらは本来の性能が出ていない可能性もある。

CPU 使用率が 80%を超えたとき、平常時より CPU 頻度が高くないか確認する。D、E、F などは CPU 使用率を下げられる可能性もある。

なお、CPU 使用率が 10%以下だとシステムの動く比率が高くなるので CPU 頻度が高くなる傾向がある。

AVM を上限値付きで使用しているときには、上記の CPU 使用率×上限値で考えればよい。

## 5. メモリによる診断手法

GS は図 6 に示すような仮想空間を持ち、システム共通域に OS の制御表やモジュールを、ユーザ固有域にユーザプログラムを配置する。そこで「仮想メモリの大小によりシステムの特性を予想することができる」と仮説をたてた。

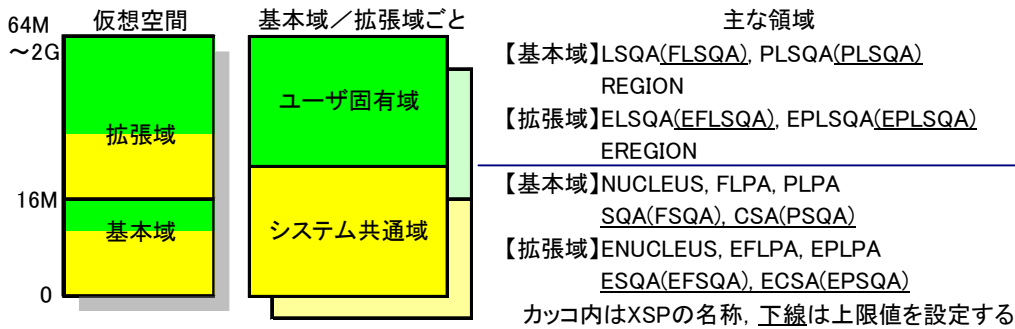


図 6 GS の仮想空間

### 5. 1 従来の性能評価

従来のキャパシティ管理でも、仮想メモリや実メモリに関する評価は行っている。

#### 【仮想メモリ】

- ・領域ごとに、使用量が上限値（システムパラメタに設定）に余裕があるか評価

#### 【実メモリ】

- ・ページング回数、スワップ回数の評価
- ・領域ごとに、V/R 比（＝実メモリ量／仮想メモリ量）の評価

実メモリは標準搭載量が増えているため、XSP システムではページングが起きることすら珍しく、時代遅れの評価項目と言える。

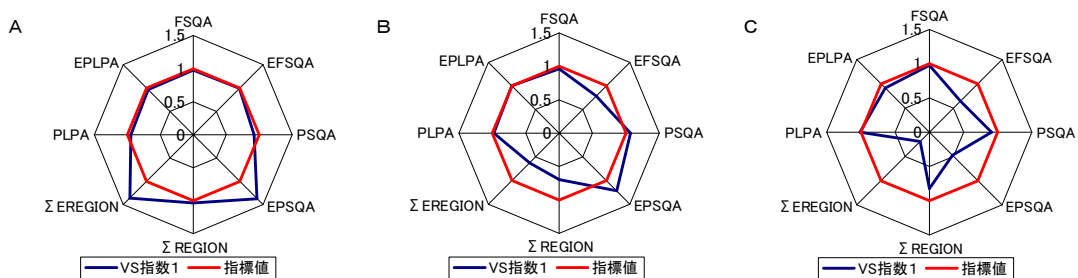
### 5. 2 仮想メモリによる診断

VS 指数を使った診断結果を図 7, 図 8 に示す。仮想メモリは苦手な人が多いため、問題がありそうか一目で見てわかるように工夫をした。簡単に言うと歪なものは問題ありと言える。厳密には、VS 指数 1 は指標値を 0.2 ポイント以上超えている項目は調査が必要である。VS 指数 2 は定義と 0.5 ポイント以上開きがある項目、定義が 2 を超えているものについては調査することが望ましい。

$$\text{VS 指数 } N = F \text{ vs } N (\text{仮想空間の最大総使用量, 指標値}) \quad N = 1 \text{ or } 2$$

$$\text{定義} = F \text{ vs } 2 (\text{仮想記憶の定義値等, 指標値})$$

$F \text{ vs } N(x, y)$  は関数, 指標値は経験値から設定している



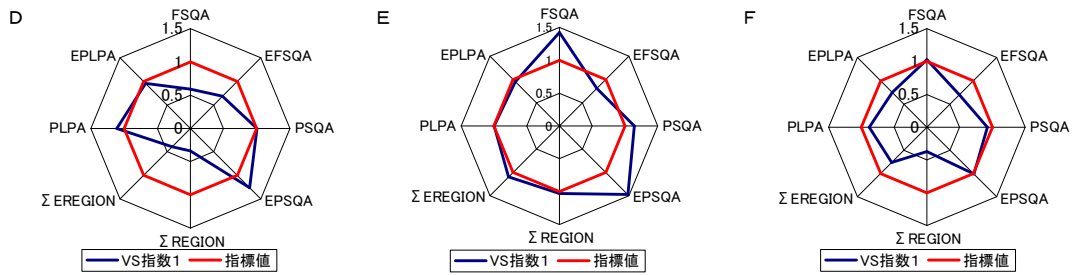


図7 仮想メモリによる診断 (VS 指数1)

【診断結果】

- ①EPSQA かつ  $\Sigma$  EREGION が高いシステム【A】 注)  $\Sigma$ は空間の総和の意
  - ・ SymfoWARE を使った巨大なプログラムが動作している
- ②PSQA や EPSQA が高いシステム～【B, D, E】
  - ・ DC バッファや排他制御表が大きい, データベースをメモリ常駐している
- ③FSQA, EFSQA が高いシステム～【E】
  - ・ 搭載メモリが大きい, VSAM が多い
- ④PLPA が高いシステム～【D】
  - ・ 古い P.P が導入されている

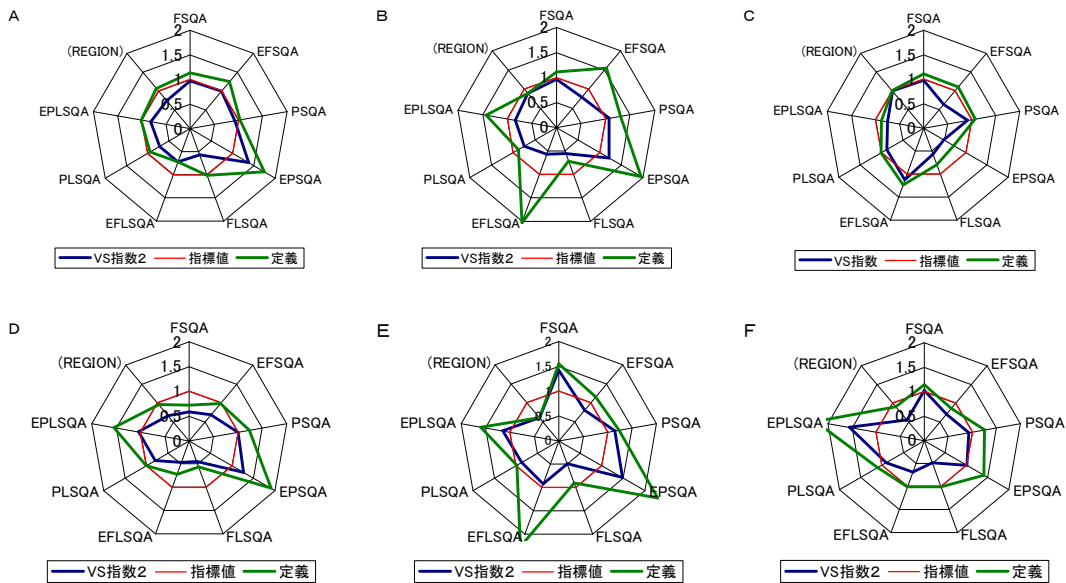


図8 仮想メモリによる診断 (VS 指数2)

【診断結果】

- ①定義と VS 指数2 のバランスが悪い (デコボコしている)【B, E】
  - ・ 考えられる原因～測定日の使用率が低い, 業務が削減された, 定義値が不適切
- ②定義が2を超えている (はみ出している)【B, D, E, F】
  - ・ EFLSQA が大きい … EPLSQA の定義値と同じにした可能性あり
  - ・ EPLSQA が大きい … 1ジョブで使っているデータベースが多い

### 5.3 実メモリによる診断

実メモリについては、まず使用率と 16MB 以下の空き率を評価する (図 9)。

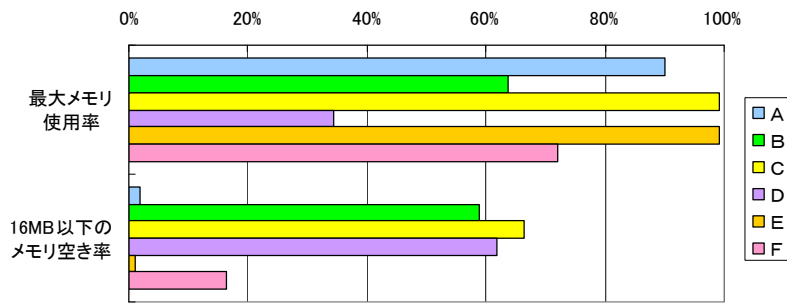


図 9 実メモリによる診断

#### 【診断結果】

##### ①最大メモリ使用率

- ・ 70%以下・・・大きすぎるメモリが搭載されている可能性あり【B, D】
- ・ 95%以上・・・ページング回数の評価が必要【C, E】⇒ 図 3 の PAGE IN を調査

##### ②16MB 以下のメモリ空き率

- ・ 10~20%で要注意【F】
- ・ 10%以下だとジョブやサブシステムが異常終了する可能性がある【A, E】

RS 指数を使った診断結果を図 10 に示す。

RS 指数 =  $Frs(R/V \text{ 値}, \text{指標値})$ ,  $R/V \text{ 値} = \text{実メモリ量} / \text{仮想メモリ量}$

$Frs(x, y)$  は関数, 指標値 は経験値から設定している

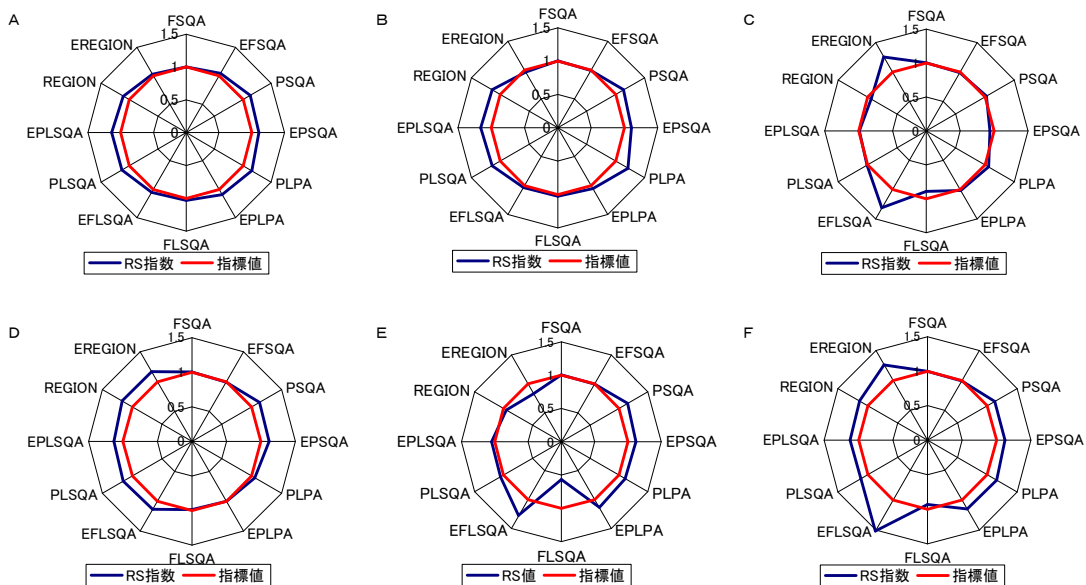


図 10 実メモリによる診断 (RS 指数)

#### 【診断結果】

- ・ EFLSQA ... セグメントテーブルが大きい【C, F】
  - ・ EREGION ... 拡張域を使っているジョブが少ない (よって R/V 値が大きい)
- 以上のように、メモリダンプを解析しなくてもシステムの特徴はある程度見えてくる。

## 6. サービスレベルの把握

内のメタボの例で紹介したが、遅すぎるオンライン業務やバッチ処理に不満感を持ちながら使っているケースがある。

図11はオンライン業務の稼動状況を一目でわかるように表した図である。

- ・円（点）が一つのオンラインプログラム
- ・赤い円は非常に遅い（最大処理時間が30秒以上）プログラム
- ・大きい丸は稼動率（＝総処理時間／測定時間）の高いプログラム

即ち、赤くて大きい丸は「よく使われていて遅い」オンラインプログラムが存在することを表している。【A, F】

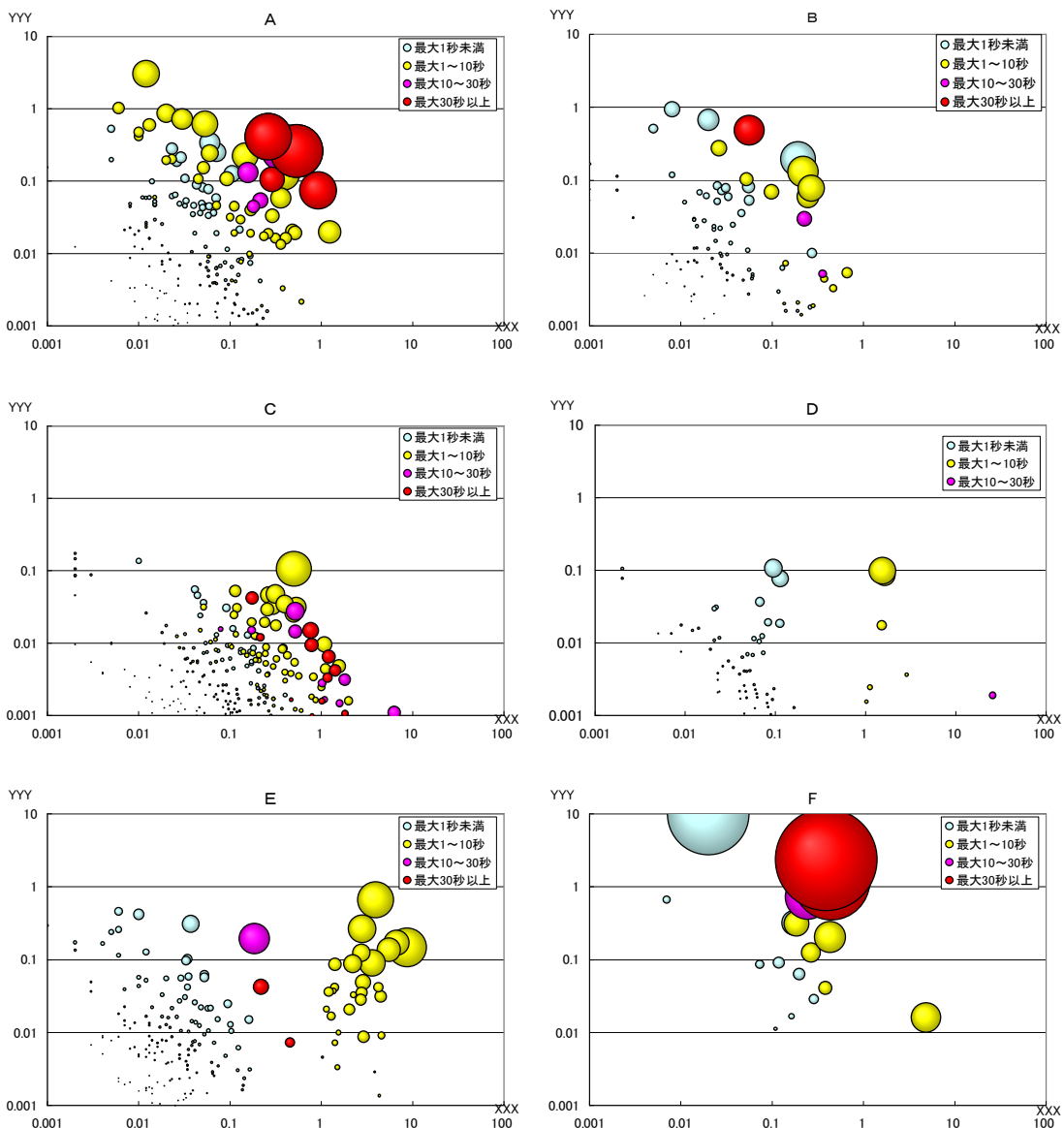


図11 オンライン業務の稼動状況



## 7. 考察

### 7.1 メタボ診断の効果

図3のグラフを見ての従来の性能評価は以下の通りだった。(再掲)

- ・ AとCは継続してCPU負荷が高い
- ・ BとFは夜間のCPU負荷が高い
- ・ DとEは突発的にCPU負荷が高い
- ・ CとEはページングが発生する
- ・ 仮想空間 [システム共通域] には余裕がある

同じデータを使った4章～6章の診断結果を表3にまとめる。C以外のシステムはメタボの兆候があると判定することができた。

表3 メタボの判定結果

	4.3 CPU/I/O 頻度	4.4 CPU使用率との関係	5.2 仮想メモリによる診断	5.3 実メモリによる診断	参考6. SL	メタボ判定
A	◎ (29) バランスよい	◎	○EPSQA, Σ EREGION	◎メモリ量	△	△1, ×1 メタボ
			◎	×16MB 以下		
B	△ (23) バラバラ	○平均してバラバラ	○PSQA, EPSQA	△メモリ量	△	△4 メタボ
			△EPSQA, EPLSQA			
C	○ (11) I/O 頻度高い	◎	◎	○メモリ量	○	メタボでない
			◎			
D	△ (56) CPU 頻度高い	△高負荷時はCPUループ	○PSQA, EPSQA	△メモリ量	○	△3 メタボ
			○EPSQA			
E	△ (45) バラバラ	◎	○FSQA, EPSQA	○メモリ量	○	△2, ×1 メタボ
			△EPSQA, EFLSQA	×16MB 以下		
F	○ (45) 夜間CPU頻度高い	△高負荷時はCPU頻度高	◎		△	△3 メタボ
			○EPLSQA	△16MB 以下		

表1に示したGを除く6件のトラブル例 (Bは2件) について、この診断によって認知できたかを考察した結果、認知可能○が3件、不可×が2件、中間△が1件となった。

(1) A : ログのパンクについて

×ログのパンクは認知できなかった ⇒ AIMのログ環境の診断が必要

【新】16MB以下の実メモリ不足を認知した

(2) B : CPUループとリカバリ不可について

○CPU/I/O頻度がバラバラであることからCPUループは認知できた

×リカバリ不可は認知できなかった ⇒ システム起動時のログの診断が必要

(3) D : メモリ増設について

○増設前からメモリに余裕があることは認知できた

(4) E : 16MB以下の実メモリ不足について

○認知できた

(5) F : CPUループについて

△実際のトラブルは日中だったが、夜間のCPU頻度が高い原因を調査し横展開すれば

認知できたかもしれない

注) GのみMSPシステムだったため、4～5章の説明を簡単にするために除外していた。

なお、GのECSA領域不足は5.2の診断で認知することは可能である。

以上の結果をふまえて、現在のGSのメタボ診断は図12のように行っている。

## ○ 必要なデータ

- ・PDL/PDA 3～5日分(必須)
- ・システムパラメタ(必須)
- ・AIMのパラメタ(必須)
- ・システム起動時のメッセージログ(任意)

## ○ メタボの判定項目と評価

- ・CPU/IO頻度による判定
- ・システム規模(IOPS)による判定
- ・仮想メモリによる判定
- ・DASD I/Oによる判定
- ・システム環境による判定
- ・CPU/実メモリ/IO環境の適正性
- ・オンライン処理による判定
- ・CPUダウングレードの実現性評価

### ステップ1 CPU/IO頻度とIOPSで性能リスクを判定

- IOPS<100が8割以上 または <200が9割以上 …… A
- IOPS≥100 かつ 20≤CPU/IO頻度≤40が7割未満 …… B

### ステップ2 診断結果により追加リスクをカウント

①CPU	②仮想/実メモリ	③DASD I/O	④その他
CPU使用率 >80% または <20% が 8割以上	VS指数 定義値の正当性 実メモリの正当性 が1つ以上check	IOPS>200 または >100 で全体 の1割以上	システム環境 オンライン の判定が 1つ以上check

### ステップ3 ステップ1、2から内のメタボ状態を判定

ステップ1	ステップ2	メタボレベル
A【小規模】	2つ以上	△: 診断をお勧めします
Aでない かつ B【性能品質】	3つ以上 2つ 1つ	×: 早急な対応が必要です ▲: 診断を強くお勧めします △: 診断をお勧めします
A、B以外 【標準】	3つ以上 2つ	▲: 診断を強くお勧めします △: 診断をお勧めします

### ステップ4 ステップ1、2から外のメタボ状態を判定

Aである または CPU/実メモリ/IO環境の適正性  
⇒ CPUダウングレードの実現性を評価

図 1 2 GS のメタボ診断 (参考)

## 7. 2 改善へのアプローチ

GS のメタボ診断は、システムの安定稼働や適正なコストを実現するための手段である。実際に問題が検出されたとき、何らかの改善ができなければ意味がない。

CPU/IO 頻度における問題は以下のように進めていく。

### (1) CPU 頻度が高い、CPU ループと診断されたとき

- ①該当するプログラムを SMF などから調査し、数本を抽出
- ②プログラムの基礎値 (CPU 時間、ファイルごとの I/O 回数、SQL 発行回数) を調査
- ③CPU 頻度が高い原因の調査

メモリ常駐の確認、OS トレース解析、開発者へのヒアリング、ソースプログラムのチェック、コンパイルオプションの確認など

### ④改善策立案と検討

CPU 頻度が高くてよいという選択肢もある

### ⑤改善 (プログラム改修など) , 評価, 横展開

### (2) I/O 頻度が高いと診断されたとき

- ①システム全体での I/O 発行状況の調査 (I/O トレース解析)  
カタログアクセス、ファイル編成 (ISAM, DAM の利用など)
- ②該当するプログラムを数本抽出
- ③プログラムの基礎値 (CPU 時間、ファイルごとの I/O 回数、SQL 発行回数) を調査
- ④I/O 頻度が高い原因の調査 (開発者へのヒアリング中心)
- ⑤検討～I/O 頻度が高いことのリスクは小さいので急いで改善する必要はない

仮想メモリ、実メモリ等のリスクは、通常のシステムチューニングを適用する。

メタボ診断の結果は、お客様が GS をいつまで使い続けるのか、自ら賞味期限をコントロールしたり、オープン化の検討材料としても活用することができる。

## **8. 評価と課題**

### (1) 評価

2006年度から「性能品質の悪化」として問題意識を持ち論文投稿をしていたものを、2008年初めにGSのメタボ化としてまとめることができた。次の3点について評価している。

- ①GSのメタボ化の発見と定義づけ
  - ・外のメタボ（過剰投資）と内のメタボ（リスク）
- ②GSのメタボ診断手法の開発
  - ・従来の性能評価とは全く異なる独自の手法
- ③普及
  - ・お客様向けセミナーでの講演
  - ・富士通SE向けの研修会（関東，関西）での講演
  - ・FUJITSUファミリ会論文などの投稿，執筆
  - ・弊社WEBサイトでの情報発信

### (2) 課題

今後の課題は次の3点である。

- ①GSのメタボ化に関するリテラシーの向上
  - ・顧客，SE向けセミナーでの講演
  - ・活用事例のまとめ
- ②GSのメタボ診断の実施とブラッシュアップ
  - ・富士通の関連部門との連携
  - ・富士通製品，パッケージへの適用
  - ・大規模システム（MSP）への対応
- ③お客様の満足度を向上させる取組み
  - ・PDCAサイクルをまわす仕掛け
  - ・CPUダウングレードとの連携

メタボ診断はまだ生まれたばかりで更に進化し続けます。論文を読んで頂いたあなたと意見交換できることを楽しみにしています。

以上

### **参考文献**

- [1] 有賀光浩，「メインフレーム変革！ お客様視点・現場重視の性能評価 － 3割のコスト削減・性能向上・満足度向上運動－」 FUJITSUファミリ会2006年前期論文
- [2] 有賀光浩，「メインフレーム変革！ 性能品質向上への挑戦 －CPU統合の見える化と改善－」 FUJITSUファミリ会2006年後期論文

## **用語説明**

GS21, PRIMEFORCE	富士通メインフレームの機種名
HLF	AIM配下の資源（データベース等）の更新後ログファイル
REGION	ユーザプログラムが使用する仮想空間
ECSA	サブシステムのテーブルやバッファが使用する仮想空間
ISAM	索引編成ファイル
NDB	ネットワーク型データベース
RDB	リレーショナル型データベース, 製品はSymfoware/RDB
DAM	直接編成ファイル
AIM	オンラインシステムを支援するサブシステム
PAGE IN	ページング処理において, 外部ページファイルからメモリにステージングする処理
MSP, XSP	富士通メインフレームのOS
LSQA, PLSQA, REGION, ELSQA, EPLSQA, EREGION, NUCLEUS, FLPA, PLPA, SQA, CSA, ENUCLEUS, EFLPA, EPLPA, ESQA, ECSA	MSPの仮想空間の領域名
FLSQA, PLSQA, REGION, EFLSQA, EPLSQA, EREGION, NUCLEUS, FLPA, PLPA, FSQA, PSQA, ENUCLEUS, EFLPA, EPLPA, EFSQA, EPSQA	XSPの仮想空間の領域名
AVM	仮想化を実現する製品
PDL/PDA	性能評価のための性能データ
SMF	課金データ
IOPS	1秒当たりのI/O回数