

---

# 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 －

株式会社アイビスインターナショナル

---

### ■ 執筆者 Profile ■



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

有賀 光浩

### ■ 論文趣旨 ■

メインフレーム上で、システムの性能品質の悪化が進んでいる。2007年問題とは関係ない。やるべきことをやっていないだけである。問題の本質を見極め、基本に忠実に対応することが今求められている。新・性能評価プロセスの実践が重要となる。

CPUのダウングレードとともに、CPU統合のニーズが増えてきた。そこで、CPU統合プロセスを見える化し、CPU見積り手法「CPU能力積上げ方式」の考察と、3つの性能品質問題について事例を使って紹介する。

- ① AVM(仮想化)を使うと、ジョブの処理時間が不安定になることの分析と対応
- ② ブラックボックス化したシステム環境への対応
- ③ 性能品質の悪いアプリケーション（CPUループが頻発）への対応

前回の論文と合わせて、性能評価プロセスの見える化手法も一通り完成した。この取組みは性能品質の向上だけでなく、今後内部統制とも密接に関係してくるので一日でも早い着手をお勧めする。

■ 論文目次 ■

<b>1. はじめに</b> .....	《 3》
1. 1  当社の概要	
1. 2  背景	
1. 3  論点	
<b>2. CPU 統合の現状と問題点</b> .....	《 5》
2. 1  CPU 統合プロジェクトの現状	
2. 2  あいまいな CPU 選定	
<b>3. 性能評価事例</b> .....	《 8》
3. 1  不安定な処理時間	
3. 2  ブラックボックス化したシステム環境	
3. 3  性能品質の悪いアプリケーション	
<b>4. 評価と課題</b> .....	《 12》

■ 図表一覧 ■

図 1  メインフレームの動向 .....	《 3》
図 2  性能評価プロセス .....	《 5》
図 3  CPU 統合における問題 .....	《 5》
図 4  CPU 能力積上げ方式 .....	《 6》
図 5  CPU 時間の短縮率 .....	《 7》
図 6  AVM による待ち .....	《 7》
図 7  バッチジョブの処理時間の推移 .....	《 7》
図 8  旧システムからの処理時間の短縮率 .....	《 8》
図 9  ジョブ多重度と処理時間の遅延率 .....	《 8》
図 10 システムの稼動状況 (2006 年 10 月 3 日, 6 日) .....	《 11》
図 11 CPU/IO 頻度 (10 月 3 日, 6 日) .....	《 11》
表 1  性能関連の障害例 .....	《 4》
表 2  CPU 統合と CPU 単体性能との関係 .....	《 6》
表 3  バッチジョブ A, B の稼動状況 .....	《 9》
表 4  テスト用 PED 例 .....	《 12》

# 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。

2004年に株式会社アイビスインターナショナルを設立し、お客様がメインフレームをより快適に、適切なコストで使っ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、2005年度の出荷台数は949台(前年度比78%)、金額は1,933億円(前年度比78%)であった。

富士通はこの状態を「大幅な減少はなく、今後もコンスタントにメインフレーム需要が継続」とコメントし、2006年～2008年度末までの3年間で1,500台(GS21新モデルとPRIMEFORCE)を販売すると発表している。

メインフレームの動向を整理すると図1のようになる。今後リプレースを検討されているお客様の関心は以下の3点である。

- ① 複数のCPUを統合しCPU台数を削減する。
- ② オープン化を段階的に進めCPUをダウングレードする。
- ③ オープン化が完了したのでCPUを撤去する。

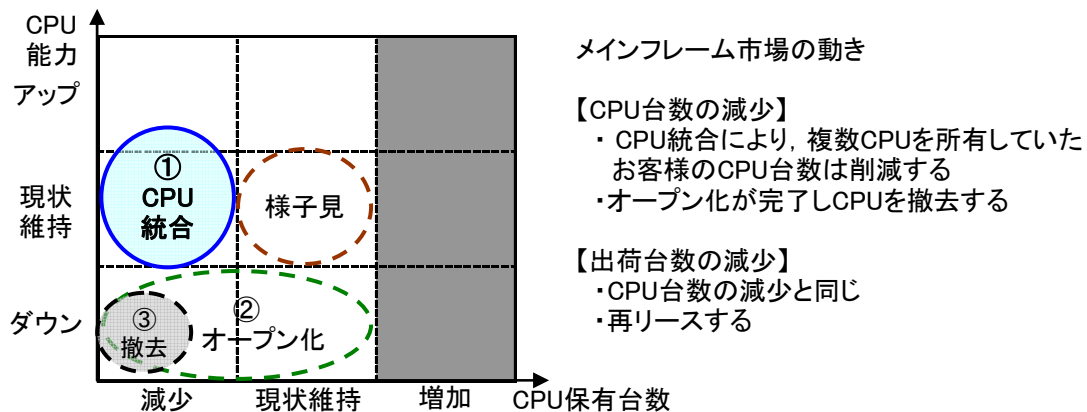


図1 メインフレームの動向

### (2) 止まらない性能品質の悪化

2005年11月以降発生した東京証券取引所でのトラブルから私たちは、運用ミス、オペレーションミス、性能問題が社会的な影響を及ぼすことを学んだ。富士通フォーラム2006の講演で黒川社長は「11月1日を富士通の再出発点と位置づけ、顧客システムの

安定運用に向けて全社一丸で取り組んでいく」と宣言している。

しかし、性能評価の現場で感じることは、システムの性能品質が年々悪化していることである。弊社が関係しているお客様で 2006 年 1 月以降に発生した性能関連のトラブル例を表 1 に示す。A～E の 5 社すべて社会システム、基幹システムの本番系である。

表 1 性能関連のトラブル例

項番	顧客	機種 発生月	現象	対応 状況	現状、原因など
1	A	GS21 400 3 月	システム停止 CPU ループ	対応済	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。
②	B	GS21 600 8 月	一部処理遅延 CPU ループ	対応中	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、9 月も 2 回発生。
③	C	PRIMEFORCE 7 月	一部処理遅延 CPU ループ	対応中	オンラインプログラムがループし CPU 負荷 100%。1 時間以上、9 月も 3 回発生。
4	D	GS21 400 1 月	オンライン業務 の一部停止	対応済	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。
⑤	E	GS8900 7 月	オンライン業務 の一部停止	保留	ECSA 領域不足し、ジョブが異常終了した。適切な対応がとれず運用制限中。
⑥	C	PRIMEFORCE ※未遂	データベースの リカバリ不能	対応済	ログ環境を毎朝初期化しリカバリ不能状態だった。 <b>※障害が無かったため実害はなかった</b>
7	F	PRIMEFORCE 5 月	無駄なメモリを 増設	対処 できず	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。

注) 項番○つきは AVM (仮想システム) を使っているシステム。

件数に統計的な意味はなく、これらを重大障害と見る人もいれば軽微な障害と見る人もいる。それよりも、高信頼性、運用・保守性を重視しているメインフレームにおいて、目前で起きている現象に気づかず、適切な対応をとらず、放置されていることが危機的な状況である。

### 1. 3 論点

性能品質の悪化は流行の 2007 年問題とは何の関係もない。やるべきことをやっていないだけであり、問題の本質を見極め基本に忠実に対応することが重要である。そのためには次ページ図 2 に示す「新・性能評価プロセス」を実践していけばよい (注)。

本論文では、CPU 統合における問題点とその対応について事例を通して紹介する。そして、性能品質向上のための提言を行う。メインフレームの CPU 統合に関する情報は少なく、他のお客様で発生した問題などを知って頂き、貴社のシステム改善や次期システム構想のヒントとして活用して頂けると大変ありがたい。

10 年後も安心してメインフレームを使い続けて頂くために「3 割のコスト削減、3 割の性能向上、3 割の満足度向上 (通称 Triple 3) 運動」を継続して実施している。私たちは、中核となる性能評価の透明化を推進する。

(注) 2006 年前期論文では性能評価プロセスの見える化ツール (CPU/I0 頻度分析等)、問題抽出手法 (性能指標値 21)、原因分析手法 (仮説検証型性能解析) について論述している。その後、⑩フィードバックを追加した。

## 2. CPU統合の現状と問題点

### 2.1 CPU統合プロジェクトの現状

複数のCPUを保有しているお客様では、仮想化技術(AVM)を使ったCPU統合を徐々に進めている。ホストコンピュータ上の仮想化技術は20年近い歴史があるはずだが、システム性能に関してはあまり議論されておらず様々な問題が発生している。問題を整理すると図3のようなる。

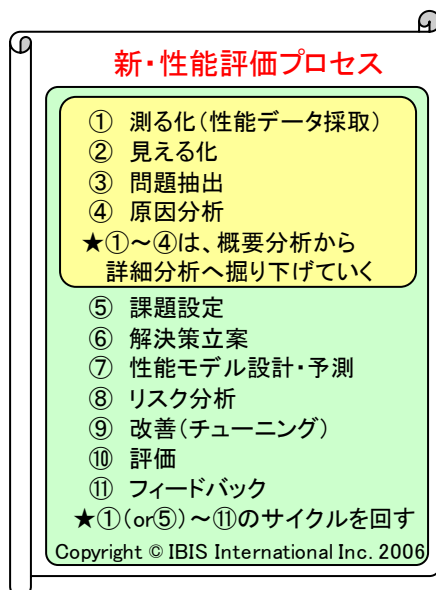
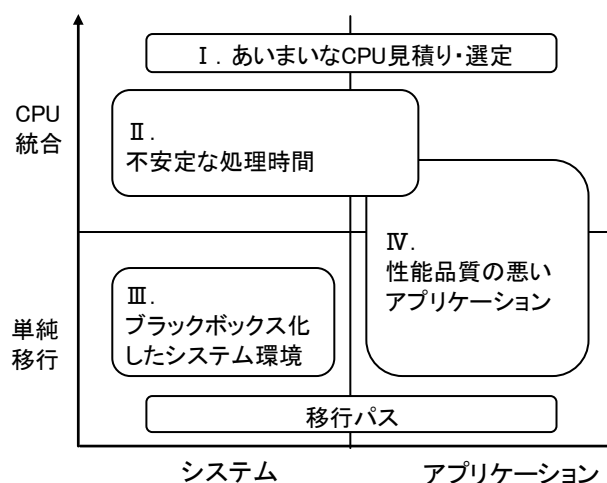


図2 新・性能評価プロセス



CPU統合: CPU統合が直接要因となっている問題  
単純移行: 単純移行でも起こりうる問題

図3 CPU統合における問題

最初の問題として、CPU統合を行うときの性能見積り(サイジング)手法が標準化されていない。既存のCPU能力をベースにした「CPU能力積上げ方式」でCPUを選定しているケースが多く見受けられるが、この方法について考察する。

第二にAVMを使うとバッチジョブの処理時間が予想以上にばらつく問題を発見した。不安定な処理時間を生み出すロジックについては今まで議論されたことが無かったので細かく解説をする。処理時間にAVM待ち時間という概念を取り入れ、処理時間のばらつきを説明している。

$$\text{処理時間} = \text{CPU時間} + \text{I/O時間} + \text{CPU待ち時間} + \text{AVM待ち時間}$$

第三にブラックボックス化したシステム環境が生み出す問題について解説する。CPU統合には直接関係しないが、システム環境の性能品質を向上させる絶好のタイミングなので是非とも参考にして頂きたい。

最後に前ページ表1に紹介したCPUループを代表とする性能品質の悪いアプリケーションについて考察する。見える化ツールCPU/I/O頻度分析ではCPUループを新しく定義した。メインフレームはCPU使用率が100%でもきれいに動いてしまうため、ループに対する問題意識が薄れている可能性がある。実践している品質向上施策についても簡単に紹介する。

## 2.2 あいまいな CPU 選定

システムの単純移行では見積りレスで CPU を選択できる移行パスという論理が定着している。CPU 統合でこれに相当するのが「CPU 能力積上げ方式」である。実際には、旧システムの稼動状況や問題を把握せずに CPU を選定しているため、統合後に問題が表面化するケースも見受けられる。

本節の内容はマニュアル等に記載されている考え方ではなく、筆者の経験からまとめたものである。

### (1) CPU 能力積上げ方式

図4の例を使って説明する。例えば、CPU が3台あり総 CPU 能力が 200 だとすると、新システムでは 230~276 (余裕値 0~20%) の能力の CPU が選択される。余裕値を調整すれば一見よさそうだが、様々な問題が潜在している。

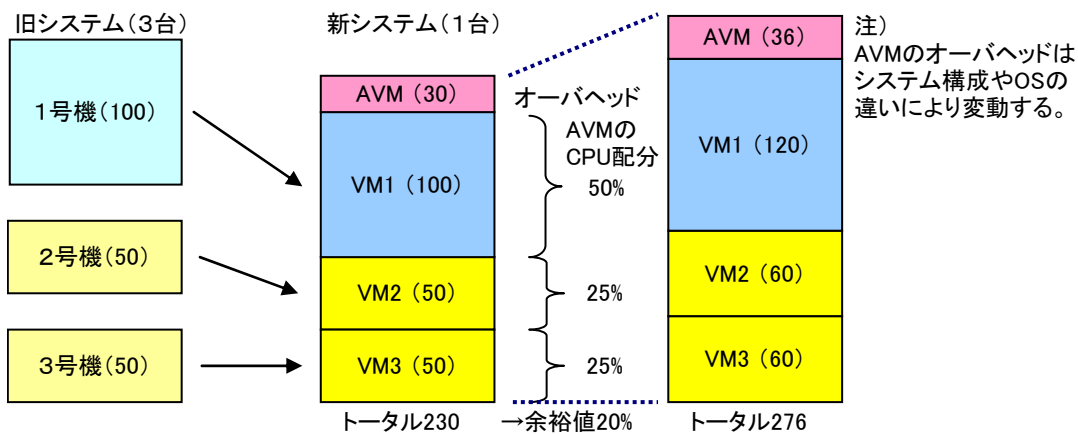


図4 CPU 能力積上げ方式

### (2) CPU の単体性能

CPU 統合の大きな特徴は、表2に示すようにほとんどのケースで単体 CPU 性能が数倍速くなることである。CPU 能力は面積、CPU 性能は速さである。仮想化は高速な CPU を時分割して使っている。

表2 CPU 統合と CPU の単体性能との関係

形態	CPU 能力	一つのCPUの単体性能		
		速い	遅い	同等
CPU統合 (AVM)	UP	○	▲シングル→マルチ	—
	同等	○CPU配分による	▲シングル→マルチ	—
	DOWN	○CPU配分による	○	—
単純移行	UP	○	▲シングル→マルチ	—
	同等	△マルチ→シングル	▲シングル→マルチ	○
	DOWN	△マルチ→シングル	○	—

シングル  
→ シングルプロセッサ(CPU1個)

マルチ  
→ マルチプロセッサ(CPU2個以上)

○: 普通の組合せ  
△: マルチプロセッサを使っていると、起こりうる組合せ  
▲: 理論上はありうるが、非常に稀な組合せ

### (3) CPU 統合前後の変化

図4のCPU能力230(図の中央)のマシンへの統合を考える。旧1号機, 2号機に比べて単体CPU性能が向上するため, CPU時間は図5のように短縮する。

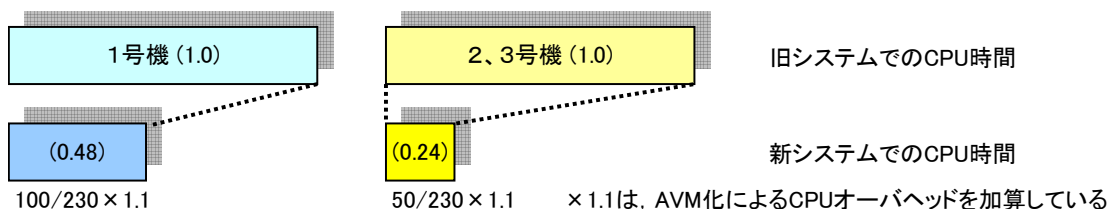


図5 CPU時間の短縮率

図6で他VM待ち時間が一定になるのがLOGICALモード, 他VMの動作状況により変動するのがAUTOMATICモードである。

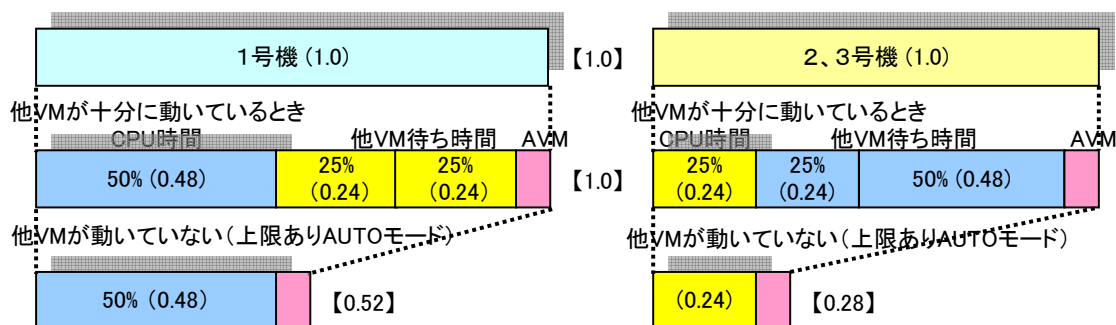


図6 AVMによる待ち時間の考え方

一般的にはCPUを有効に活用できる, 上限ありAUTOMATICモードを使用することが多い。一例だが旧システムで単体処理時間が15分(CPU時間10分, I/O時間5分)であるバッチジョブを新システムで実行すると, 図7のように10分弱~15分で変動する。

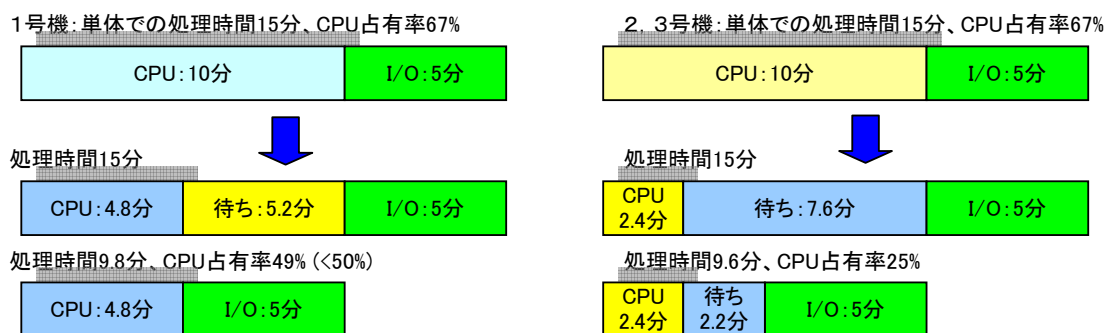


図7 バッチジョブの処理時間の推移

重要な点は自システムの単体処理性能が他VMの稼働状況に左右されることである。この原因はCPUの単体性能が速いことである。実際にあったCPU統合モデルで他VM未動作時のシミュレーション結果を図8, 図9に示す。

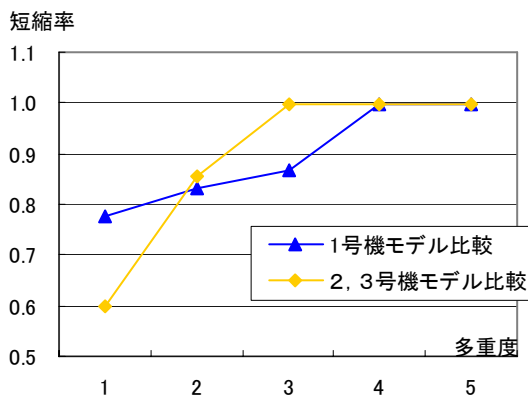


図8 旧システムからの処理時間の短縮率

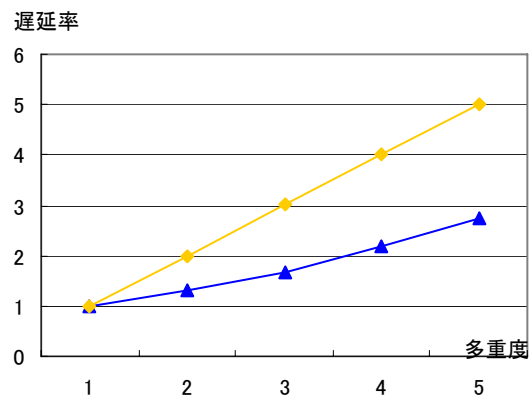


図9 ジョブ多重度と処理時間遅延率

AVM では旧システムと同等の CPU を割り当てている。

図8では多重度が小さいと処理時間が短くなってしまふことがわかる。自システムの CPU を使い切ると旧システムと同じ処理時間になる。

図9では CPU 割り当てが小さいと処理時間は多重度に比例して2倍3倍と延びていくことがわかる。多重走行に強いメインフレームの面影もない。

#### (4) リスクヘッジ

見積りレスの CPU 能力積上げ方式には次の3点のリスクがある。

- ① 過剰な CPU 投資をする可能性があること。
- ② 旧システムで潜在していた問題が表面化する可能性があること。
- ③ 多重性能が弱くなること。

リスクヘッジの方法について考える。

適切な CPU 能力を見積もるためにはシステムの目標サービスレベル（レスポンス時間、夜間バッチ処理の終了時間、単位時間内の処理件数など）の明確化と現状把握が必須条件である。これができないときは、積極的に過剰投資をして頂くか、運を天にまかせるしかない。（自分の山勘を自慢する人が多いのも事実である）

次に、現行のシステムに対し図2で示した新・性能評価プロセスを着実に実践することが重要である。性能トラブルを起こすシステムの半数以上は性能データの採取すらできていない。現行システムの重大問題（後述のブラックボックス化したシステム環境、性能品質の悪いアプリケーションなど）は早めに対処しておく必要がある。

多重性能については、AVM の LOGICAL モード（注）を使うことにより性能の安定化を図ることができる。

（注） LOGICAL モードが敬遠されてきた理由は以下の2つである。

- ・ 空いている CPU を使わないのはもったいない。
- ・ CPU 速度が遅い時代に、I/O 完了が遅れ、その影響で I/O チェーン切れを起こしバッチ処理の遅延、CPU 時間の増加を起こした。



### 3. 性能評価事例

#### 3.1 不安定な処理時間

原因とリスクについては前節で解説したのでここでは事例を紹介する。

##### (1) システムの概要

CPU : GS8300(移行前) → PRIMEFORCE3000 シリーズ(移行後)
メモリ : 512MB → 512MB    DASD : F6495 → ETERNUS6000
AVM/EX (上限あり AUTOMATIC モード)
バッチ処理 : オンライン稼動中に実行されるかつ処理時間の長い業務バッチ

##### (2) ジョブの稼動状況

表3にジョブの稼動状況を示す。バッチAは I/O 中心の処理、バッチBは CPU 中心の処理である。処理時間を下式で表すと、バッチBの方が CPU 待ち時間、AVM 待ち時間の影響を受けやすい。

$$\text{処理時間} = \text{CPU 時間} + \text{I/O 時間} + \text{CPU 待ち時間} + \text{AVM 待ち時間}$$

$$\text{AVM 待ち時間} = \text{他ゲスト待ち時間} + \text{AVM 走行時間}$$

表3 バッチジョブA, Bの稼動状況

ジョブ		処理時間	→短縮率	CPU時間	→短縮率	EXCP回数	→増減率	CPU占有率	CPU/EXCP
バッチA	移行前	10:40:44	1	0:57:36	1	8,956,193	1	9%	0.386
	移行後	3:45:03	0.35	0:20:06	0.35	8,611,340	0.96	9%	0.140
	移行後	4:17:30	0.40	0:24:25	0.42	8,618,679	0.96	9%	0.170
ジョブ		処理時間	→短縮率	CPU時間	→短縮率	EXCP回数	→増減率	CPU占有率	CPU/EXCP
バッチB	移行前	2:45:29	1	1:06:41	1	427,967	1	40%	9.350
	移行後	1:21:43	0.49	0:30:02	0.45	428,019	1.00	37%	4.211
	移行後	2:34:43	0.93	0:37:27	0.56	428,100	1.00	24%	5.248

EXCP回数: 論理I/O回数

CPU占有率=CPU時間/処理時間

CPU/EXCP: 1EXCP当たりのCPU時間(ms)、バッチAよりバッチBの方がCPUを使っていることがわかる

##### 【補足説明】

バッチAについて (I/O バウンド処理)

- ・処理時間の短縮率が 0.35~0.40 になったのは DASD 性能の向上による効果。
- ・移行後の処理時間に 32 分の差があるが CPU 待ちによる影響。
- ・今後 CPU 負荷が上がっても処理時間は影響を受けにくい。

バッチBについて (CPU バウンド処理)

- ・移行後の 1:21:43 が速すぎる。
- ・移行後の処理時間 1 時間 13 分の差は CPU 待ちと AVM 待ちによる影響。

##### (3) 課題

- ・処理時間の安定化 (処理を速くしすぎない)。
- ⇒ LOGICAL モードの適用を検討中。

### 3. 2 ブラックボックス化したシステム環境

システムの運用担当者は、決められた責任範囲内で正確に作業を行っている。しかし、運用管理全体から見るとモレがあり、予想外の重大トラブルを起こす可能性がある。

弊社が関係したお客様での事例を紹介する。「顧客システムの安定運用に向けて全社一丸で取組む」には最適な案件であると思う。また、お客様も自身の問題として点検して頂きたい。

#### チェック 1：システムの起動プロシジャを点検する。

昔の担当者が設定しているため、ブラックボックス化している。

実際にあった例

- ①システムパラメタを SET コマンドや SET PFG コマンドで変更している。  
特に COLD-IPL 時だけ動くロジックに注意
- ②AIM-SNAP を起動している。
- ③AIM のログ環境(SIS)を毎回 INIT モードで起動している。

#### チェック 2：AIM のリカバリ環境を再点検する。

何度も点検しているはずなのだが、運よく動いていたシステムは多い。

- ①一時的にリカバリ不能状態で運用している。
- ②バックアップがとれていない。
- ③AIM ログファイル、データベースのボリューム配置が不適切。
- ④AIM ログ環境が現行システムに合っていない。(バッファ個数, ブロック長, 容量)  
特に SymfoWARE を使うようになってログ量が増加している。

#### チェック 3：システムパラメタを点検する。

煩雑、解読不能なパラメタになっていないか。動的変更していないか。

#### チェック 4：仮想メモリ, 実メモリを点検する。

メモリに関する問題は、偶然表面化した事象であり、その裏に大きな問題が潜んでいることが多い。これを深堀し、根本的な対応をすることが重要である。

例えば、16MB 以下の実メモリ不足 (表 1 の項番 2) はシステムパラメタの変更だけでほぼ改善できるが徹底されていない。具体的に言うが、自治体システム全体の問題だと早く認識し、総点検かつ未然防止を図るべきである。

#### チェック 5：性能データを取得できることを確認する。

システム移行の前後には、性能データ(PDL)の取得を作業項目に必ず入れること。

性能データが正常に取得できているか、事前に必ず確認すること。

SMF データは保管すること。

CPU やメモリの使用状況はお客様にオープンにすること。

### 3. 3 性能品質の悪いアプリケーション

CPU ループ (表 1 の項番③) の事例を紹介する。CPU ループの定義を明確にし、見える化することが重要である。

#### (1) システムの概要

CPU:PRIMEFORCE3000 シリーズ   メモリ:512MB   DASD:ETERNUS6000  
AVM/EX (上限あり AUTOMATIC モード)

#### (2) 見える化

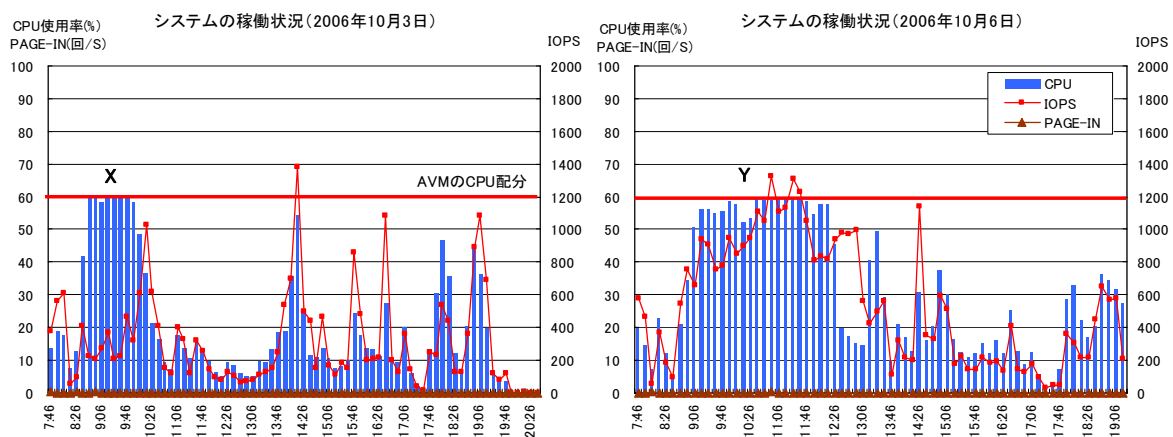


図 10 システムの稼働状況 (2006 年 10 月 3 日, 6 日)

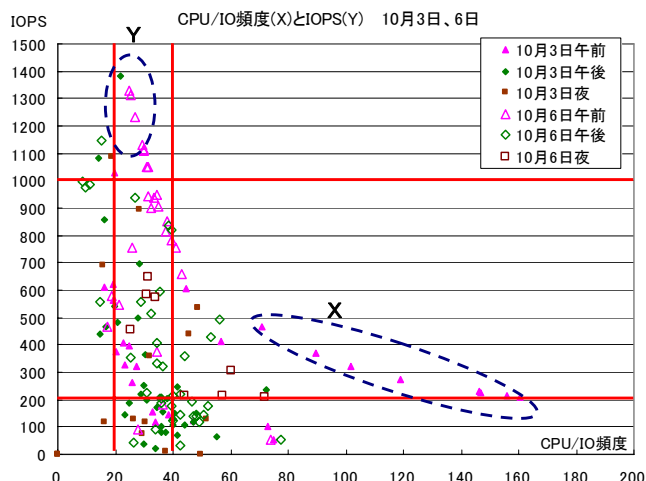


図 11 CPU/I/O 頻度 (10 月 3 日, 6 日)

前回論文で発表した CPU/I/O 頻度分析に CPU ループを新しく設けた。

CPU/I/O 頻度	20 以下 (左)	20~40 (中央)	40 以上 (右)	80 超 (右)
システムの傾向	I/O 頻度が高い	正常値	CPU 頻度が高い	CPU ループ

図 10 を見ると, X も Y も CPU 使用率は上限の 60% に達しているが, 図 11 から CPU ループ状態 (CPU/I/O 頻度が超), Y は CPU の過負荷状態 (20~40) と判断できる。

メインフレームは CPU が 100%（例では上限の 60%）になってもシステムは正常に動くので、ユーザジョブが CPU ループ状態になっても認識しにくい。

また、AVM 配下で動作していると、他 VM に CPU をディスパッチするために割り込みが入るので、他ジョブにも CPU が割り当てられやすくなっている可能性がある。

CPU ループへの問題意識が薄れているのはこういった背景も考えられる。

### (3) ジョブの特定

この状態のときに DISPLAY コマンドの投入またはリアルタイムモニタの操作ができればジョブの特定は容易にできる。

事後調査は SMF や PDL を使って、CPU 使用時間の多いジョブ、CPU 占有率の高いジョブを絞り込んでいく。MSP はジョブごとに CPU/IO 頻度を算出することができるので XSP よりも調査がしやすい。

実際には、プログラム名がわかってからの調査が大変であり、テスト工程で品質を高めることが重要である。

### (4) 課題：品質向上施策～テスト工程での改善

業務開発チームにできるだけ負担をかけず、知らないうちに性能データを拾う仕掛けが必要である。現在、プロジェクトで試行中のため概要を紹介する。

まず、プロジェクトとして性能品質の上限値（大き目でよい）を設定する。

例) オンライン処理であれば、1 トランザクションの処理時間は 20 秒以内、CPU 時間は 10 秒など。

IT テスト用に上限値を超えたら異常終了させる PED を用意する（表 4）。発生した事象については、その根本原因についても必ず考察する。

表 4 テスト用 PED 例

定義	内容	設定方法	処置
PED コマンド	CPU 時間監視 (ms)	TRANSACTION-TIME IS 10000	異常終了
	実時間監視 (s)	TRANSACTION REAL TIME IS 20	異常終了
	排他資源数監視	OCCUPATION LIMIT IS 100	警告メッセージ
	AIM 課金統計情報取得	ACCOUNT IS USED	HLF に取得
【参考】 AIM パラメタ	排他待ち時間取得 (s)	SMFWTIME 30000	SMF115 に取得
	ログデータ出力量監視 (KB)	LOGALARM 1000	警告メッセージ

以下の性能データは常時採取する。

PDL, SMF, AIM 課金統計情報（表 4 の PED コマンド）

SymfoWARE の SMF（システムチューニングマクロで設定）

特に、SMF, HLF ファイルは十分な容量を確保する必要がある。

### (5) 補足

CPU 統合後、CPU 負荷率が予想通り下がらないケースがある。移行前後で CPU 時間が余り変わらないようなジョブがないかチェックする必要がある。

## 4. 評価と課題

システムの性能品質を向上させる活動の成果が、お客様先で少しずつでてきた。

今後、全てのお客様が公平に安定したシステムの運用ができるよう、メーカーも巻き込んで展開していきたい。

### (1) CPU 統合の見える化

CPU 統合のプロセスやリスクを見える化し、性能トラブルの未然防止を実施中。

前回の論文と合わせて、性能評価プロセスの具体例も一通り完成した。

### (2) お客様，SE が理解できないシステム環境の改善

ブラックボックス化した環境を解説し、お客様，SE に説明し、改善の方向を検討。

### (3) アプリケーションの性能品質の改善

CPU/IO 頻度分析により、性能品質の悪いプログラムのあぶりだしは可能となった。

テスト工程での性能品質改善が課題である。

今後も論文やホームページを通して事例やノウハウを発信し続けるつもりだが、多くの人に活用して頂き、性能品質を向上して頂くことが課題である。近い将来、これらがすべて内部統制と密接に関係するので一日も早い着手をお勧めする。

以上

### 用語説明

AVM	メインフレーム上の CPU の仮想化を実現するソフトウェア
CPU/IO 頻度	CPU と I/O のバランスを意味し正常値は 20~40 になるように算出した係数 参考文献 C を参照
PDL, SMF	代表的な性能データの名称。
PED コマンド	オンラインジョブ（及び DB を使うバッチジョブ）の環境定義
PRIMEFORCE	メインフレームの名称

### 参考文献

- A. 有賀光浩, 「メインフレームのパフォーマンス改革 2004 - お客様が感動したチューニング効果とコスト削減のプロローグ」 FUJITSU ファミリー会 2004 年度後期論文
- B. 有賀光浩, 「性能管理・改善のブレークスルー - 自らの常識を破り, どのように性能改善を実現したか」 FUJITSU ファミリー会 2005 年度前期論文
- C. 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 ファミリー会 2006 年度前期論文