
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 －

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 ■

メインフレーム変革！ レガシーだ、コスト高だと言われ放題のメインフレームだが、信頼性、運用性をはじめ優れたサーバであると私は評価している。

10年後もメインフレームを安心して使い続けて頂くために「3割のコスト削減・性能向上・満足度向上（通称 Triple 3）運動」を開始した。中核となる性能評価はよくわからない、むずかしいと昔から指摘を受けており、お客様視点・現場重視に変えることが最優先課題であると考え、次の3点に取組んだ。

- ① 性能評価プロセスの透明化
- ② 徹底した見える化 — CPU/IO 頻度分析など見える化ツール群の開発
- ③ 原因分析サイクル、改善サイクルのスピードアップ

運動は始まったばかりだが、性能品質を向上させながら強いシステムに変革していく過程と成果を3つの事例を使って紹介する。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 当社の概要	
1. 2 背景	
1. 3 論点	
2. 性能評価プロセス	《 5》
2. 1 性能評価プロセスとは	
2. 2 見える化ツール～CPU/IO 頻度分析	
2. 3 見える化ツール～DASD の I/O 分布	
2. 4 問題抽出～性能指標値 21	
2. 5 原因分析～仮説検証型性能解析	
3. 性能評価事例	《 8》
3. 1 事例 1：夜間バッチ処理時間を短縮したい	
3. 2 事例 2：オンラインレスポンスを改善したい	
3. 3 事例 3：次期 CPU 選定のアプローチ	
4. 評価	《 13》
5. 今後の課題	《 13》
6. おわりに	《 13》

■ 図表一覧 ■

図 1 システムのライフサイクルと性能	《 3》
図 2 性能評価プロセス	《 5》
図 3 性能データのポジショニングマップ	《 5》
図 4 CPU/IO 頻度分析	《 6》
図 5 DASD の I/O 分析	《 6》
図 6 事例 2 の CPU/IO 頻度分析	《 9》
図 7 事例 2 の DASD の I/O 分析	《 9》
図 8 事例 2 のレスポンス分析	《 10》
図 9 バッチ処理の性能予測	《 12》
表 1 メインフレームの性能指標値（標準的な例）	《 4》
表 2 問題点とその原因（事例 1）	《 8》
表 3 問題点とその原因（事例 2）	《 10》

1. はじめに

1. 1 当社概要

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, GS シリーズ）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/>）。富士通プラットフォームソリューションセンターのメインフレーム唯一のアライアンスパートナーである。

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。

弊社はベンダーから独立した企業として、また富士通と協力し、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) 富士通メインフレームの動向

2006 年 5 月 15 日、富士通は 3 年ぶりに GS21 シリーズの新モデルを発表し、今後もメインフレームの開発を進めていく方針を示した。最上位機の性能は IBM に遠く及ばないようであり、メディアは「富士通は IBM との性能競争から降りた」と衝撃的に伝えている。

しかし、4,000 と言われるメインフレームユーザのほとんどは最上位機種種の性能など関係のないことである。今後もサポートを続けるとコミットしたこと、価格性能比が下がったことはお客様にとって最高のチャンスである。

(2) 性能品質の劣化

昨年の東証でのトラブルをはじめとして、性能問題が社会的に影響を及ぼすことが現実的に起きている。性能評価の現場で感じることは、システムの性能品質が想像以上に劣化していることである。性能品質が悪いとは、機能を実現するのに必要とする倍以上の CPU や I/O を使っていることを意味する。

システムのライフサイクルにおいて性能は重要な位置づけにあり（図 1）、SLA(Service Level Agreement)としても定義される。しかし、実際には性能検証していない（できていない）システムを見るのが非常に多い。

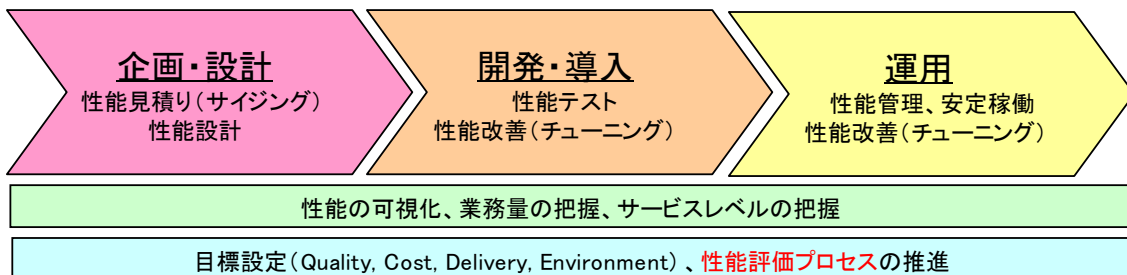


図 1 システムのライフサイクルと性能

背景には、工数削減・納期短縮による弊害、性能改善を推進するリーダ不在、また、ハードウェアの高速化・大容量化に伴いテスト時に性能の不具合に気がつかないことなども考えられる。

また、性能管理をしているシステムであっても、長年使っている「性能指標値」には大きな問題が潜在している。

(3) 性能指標値

性能管理を行うとき、管理（監視）項目に指標値を設定する。メインフレームの性能指標値は 20 年以上前からほとんど変わっていないことが多い。ハードウェアの処理能力や容量は大幅に向上し、システムの形態も変化しているが、1995 年頃に RAID ディスク装置の提供で若干の見直しが入っただけである（表 1）。

表 1 メインフレームの性能指標値（標準的な例）

性能評価の対象	管理項目	指標値の例	
		第 1 世代 1980 年台	第 2 世代 1995 年～
CPU	CPU 使用率	70～90%	←
チャンネル	チャンネル負荷率	30～40%	(MSP) 50～60%
DASD	DASD 負荷率	30～40%	←
	I/O レスポンス時間	30ms 以下	10ms 以下
RAID ディスク	キャッシュヒット率	—	0.8 以上
実メモリ	ページング回数	(MSP) 20～60 程度	←
	V/R 比	REGION は 0.5 以上	←
仮想メモリ	共通域の使用率	80%以下	
サブシステム	バッファ使用率	80%以下	←
オンライン業務	レスポンス時間	業務ごとに設定	←
	トランザクション件数	〃	←
バッチ業務	処理時間	業務ごとに設定	←
	処理件数	〃	←

この指標値には次に示す大きな問題を含んでいる。

- ・ハードウェアの過剰投資を評価できない。
- ・システムの性能品質が良いことが大前提である。

性能管理はメーカー任せであることが多く、お客様の視点で注目することは稀である。例えば「平均 CPU 使用率：20%」は、ほとんどの SE は「性能上問題ありません」と言うが、高価な CPU が 80%も遊んでいて本当に問題ないのだろうか。システム開発や運用をメーカーに委託していたとしても、お客様自らが性能をチェックし、メーカーと協力して改善していくことが重要だと考える。

1. 3 論点

レガシーシステムとは「変革のスピードに追いつけない、価値を創造しないシステム」を意味する。「メインフレーム＝レガシー」は正しくなく、オープンでもレガシーは存在する。富士通のメインフレームは優れたサーバであり、メーカーはもっと自信を持ち、自らが製品を使いこなし（重要）、お客様と未来を語る努力をすべきである。

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」を開始した。中核となる性能評価は昔から，よくわからない，むずかしいと指摘を受けており，お客様視点・現場視点への変革が最優先課題であると考え，性能評価プロセスの透明化に取り組んだ。

2. 性能評価プロセス

2.1 性能評価プロセスとは

性能評価プロセスを図2のように定義した。「①測る化」で使用する性能データは図3に独自のポジショニングマップとして整理した。これらはお客様のシステムで今すぐ活用できるものである。既にやっているという方も今一度ご確認頂ければと思う。

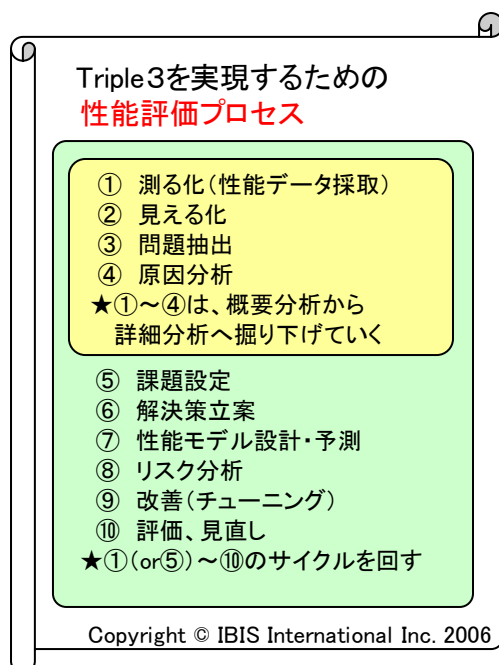


図2 性能評価プロセス

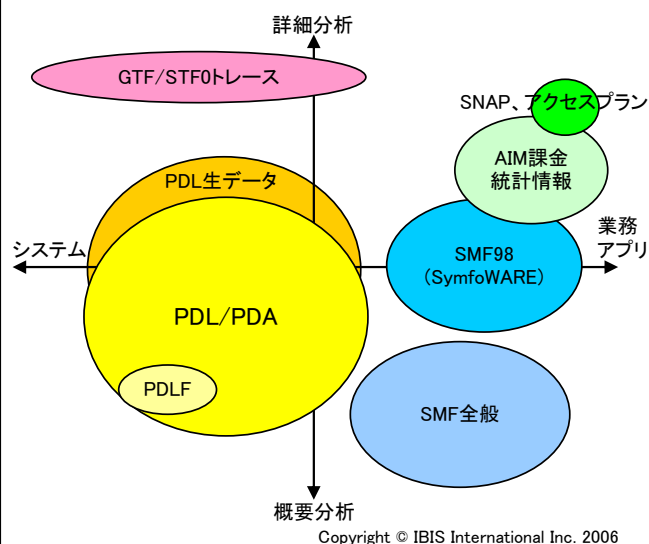


図3 性能データのポジショニングマップ

性能評価の代表的な失敗例とチェック方法を以下に示す。（プロセス①～⑤まで）

- ・使ったことのある性能データしか使わない。 → 図3と照合すると明確になる
- ・性能データを正確に採取できていない。 → 参考文献Aに記載
- ・指標が古いため真の問題抽出ができない。 → 前述+後述 2.4
- ・根本原因が究明できていない。 → ①～④サイクルが回っているか
- ・問題を整理しないため，適切な課題設定ができない。 → 場当たりに効果の小さいチューニング（改善作業）をしていないか

2.2 見える化ツール～CPU/IO 頻度分析

性能評価の見える化ツール群の柱となる「CPU/IO 頻度分析」を図4に示す。これは前述の「システムの性能品質はよいことが大前提である」を検証するリトマス試験紙のようなものであり，潜在している問題が一目でわかる画期的なツールである。

横軸の CPU/I/O 頻度は、CPU と I/O のバランスを意味し、正常値は 20~40 に入るように算出している。20 以下は I/O の頻度が多く、40 以上は CPU ループの傾向がある。

縦軸の IOPS (実 I/O 回数/秒) が 1,000 以上は XSP という OS の性能特性とチャンネル 4 パスの初期警戒ポイントの設定をしている。図 4 では縦横中央の領域が最も望ましい。

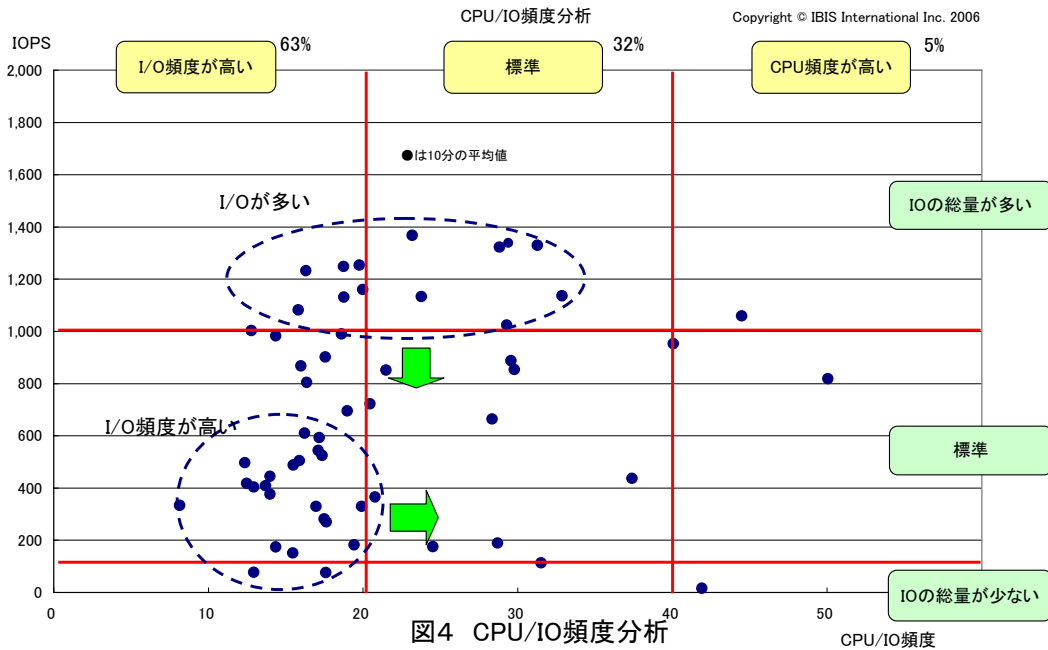


図4 CPU/I/O頻度分析

2. 3 見える化ツール～DASD の I/O 分布

次に、見える化ツール「DASD の I/O 分布」を図 5 に示す。

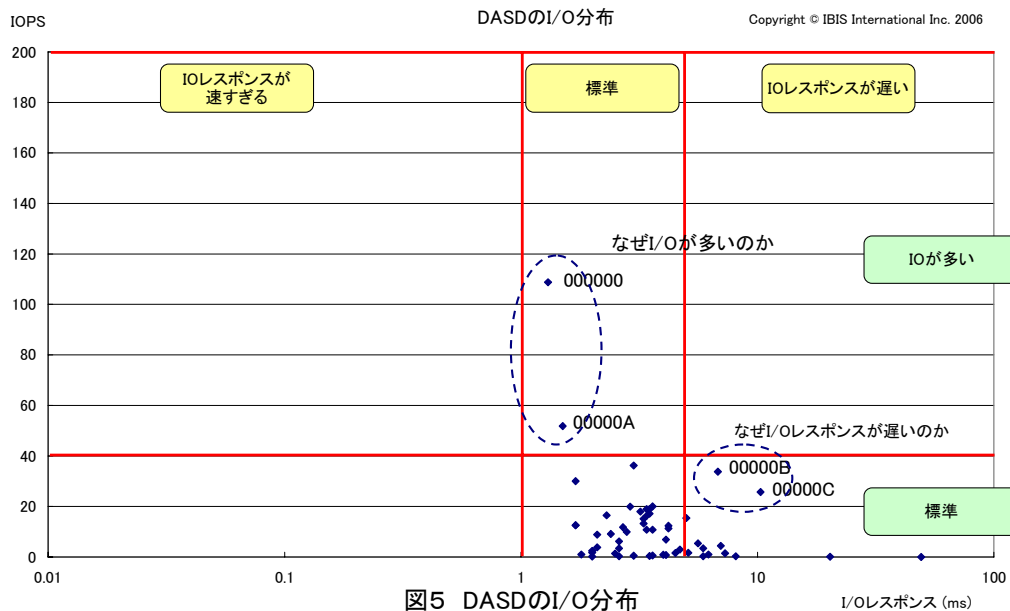


図5 DASDのI/O分布

横軸の I/O レスポンスは、現在では 1~5ms が標準的である。メーカーも認識していないが、I/O レスポンスが速すぎる (1ms 未満) のは運用上大きな問題となりうる。

縦軸は各ボリュームへの IOPS を示し、40~200 は I/O が多く、200 を超えるとシステ

ム上ボトルネックとなる。図5では中央下の領域が最も望ましい。

アレイディスクの登場でI/O性能は飛躍的に向上しDISKネックになることは稀になった。数100超の全ボリュームを性能管理することは非効率であり、DASDのI/O分布によってチェックすべきボリュームが見えるようになる。

(図4, 図5は後述する事例1で実際に使ったものであり, 個別の分析は3.1を参照)

2.4 問題抽出～性能指標値 21

従来の性能指標値に性能品質と最適投資の視点を追加し, 性能指標値 21 と命名する。

(1) CPU/I/O 頻度の 20 以下または 40 以上が 30%以上を占めている (性能品質が悪い)。

⇒ 性能品質を改善しない限り, 価値のある性能管理はむずかしい。

(2) 10 分平均の CPU 使用率が継続して 30%以下のとき, CPU の過剰投資である。

⇒ CPU 能力を落とせる可能性大である。

(3) オンライン稼働中の R/V 値が 1.0 以上のとき, メモリの過剰投資である。

⇒ $R/V \text{ 値} = \text{搭載メモリ量} / \text{全空間の最大仮想記憶サイズの和}$ と定義する。

ページング回数が 0 のとき, メモリを搭載しすぎているケースが多い。

参考) 従来の V/R 比: 領域ごとに $\text{実メモリ量} / \text{総仮想記憶量} (\leq 1)$

(4) 1 トランザクション (オンライン), 1 ジョブステップ (バッチ) が長い。

⇒ 処理が冗長している可能性がある。 ※指標値は現在検証中

メインフレームがオープン系サーバよりも優れている点を忘れてはいけない。

- ・ CPU 使用率が 100%になってもシステムは安定して運用できる。
- ・ 高性能なページングスーパーバイザが提供されている。
- ・ 強力な OS のトレース機能がある。
- ・ 多機能, 高品質なサブシステム AIM が提供されている。

2.5 原因分析～仮説検証型性能解析

抽出した問題については, 根本原因を究明することが重要である。システムの運用管理者は, アプリケーションの性能問題に消極的なケースが多い。そこで, トレース (MSP は GTF, XSP は STF0) を積極的に活用した仮説検証型性能解析を行うことで, プログラム (システム) 内の正確な動きと原因究明にスピードアップが実現できる。

仮説: ・性能問題の原因はプログラム (システム) 内に存在する。

・開発者は原因を知っている。(今は気づいていない, 忘れている)

前提: ・性能問題の状態を再現できること。

・プログラム開発者 (プログラムロジックのわかる人) と会話ができること。

手順: ①再現テスト～トレースの採取

↓ (再現できたとき) (再現できないとき) →他の手段を検討

②トレース解析

↓ (原因が見えたとき) (原因が見えないとき) →仮説の再設定

③データを持って, 開発者にヒアリング

↓ (何かに気づいたとき) (気づかないとき) →誰ならわかりそうか

④開発者に引継ぎ, 管理者に状況報告。

3. 性能評価事例

現場で性能評価プロセスを実践した3つの事例を紹介する。

3.1 事例1：夜間バッチ処理時間を短縮したい

(1) システムの概要

CPU : GS8500 メモリ : 384MB DASD : GR840 夜間バッチ処理 : DB の UNLOAD → 業務処理 → DB の RELOAD 課題 : RELOAD の処理時間 (約 2 時間) の短縮

(2) 見える化

CPU/I/O 頻度分析 … 図4, DASD の I/O 分布 … 図5 を参照

- I/O 頻度が高く, I/O も多い.
- 000000 と 00000A の I/O が多く, 00000B と 00000C の I/O レスポンスが遅い.

(3) 問題抽出と原因分析

仮説検証型性能解析により, 表2に示す問題点について根本原因の究明をした.

表2 問題点とその原因 (事例1)

問題点	主な調査データ	原因
① I/O 頻度が高い (図4)	トレース	OS カタログへの I/O が多い → 問題点②へ
② OS カタログへの I/O が多い	トレース 実機で検証	ファイル名のつけ方 カタログの並びがバラバラ
③ I/O が多い (図4)	トレース	OS カタログへの I/O が多い → 問題点②へ VSAM カタログへの I/O が多い → 問題点④へ
④ VSAM カタログへの I/O が多い	トレース ヒアリング	VSAM アクセスルーチンで毎回 OPEN/CLOSE をしている.
⑤ 000000 の I/O が多い (図5)	トレース	OS カタログへの I/O が多い → 問題点②へ
⑥ 00000A の I/O が多い (図5)	トレース	特定のファイルが原因でなく, 単純にボリュームへの I/O が多い.
⑦ RELOAD が遅い	PDL, SMF	DB の I/O レスポンスが遅い → 問題点⑨へ
⑧ 00000B, 00000C の I/O レスポンスが遅い (図5)	トレース	RELOAD している DB の I/O レスポンスが遅い → 問題点⑨へ
⑨ RELOAD している DB の I/O レスポンスが遅い		キャッシュヒットしていない (仮説) → 問題点⑩へ
⑩ RELOAD がキャッシュヒットしない (仮説)	実機で仮説検証	システムランダムのためヒットしにくい構造, かつ DB サイズに比べキャッシュ容量が小さい
⑪ SMF タイプ 110 の出力が異常に多い	SMF ヒアリング	ジョブを特定, 問題点④が原因だと判明 → 問題点④へ

(4) 主な改善項目と効果

- OS カタログの圧縮 (再創成) を行った. … 問題点②
 ⇒ 一つのファイルをアロケーションするときの OS カタログへの I/O 回数が 583 回 → 21 回 に削減した. ボリューム 000000 の IOPS は 100 以上 → 40 以下 に減少し, 夜間バッチ全体のスループットが向上した.
- RELOAD をキャッシュヒットするようにプログラムを改修した. … 問題点⑩
 ⇒ RELOAD 処理は 約 2 時間 → 約 40 分 に短縮し, 課題は解決した.

3. 2 事例2：オンラインレスポンスを改善したい

(1) システムの概要

CPU : GS8500 メモリ : 1GB DASD : ETERNUS 6000
 オンライン処理 : 10年以上前の基幹業務に新しい業務が逐次追加
 課題 : オンライン処理のレスポンス改善 (システム移行後にレスポンスが悪化)

(2) 見える化

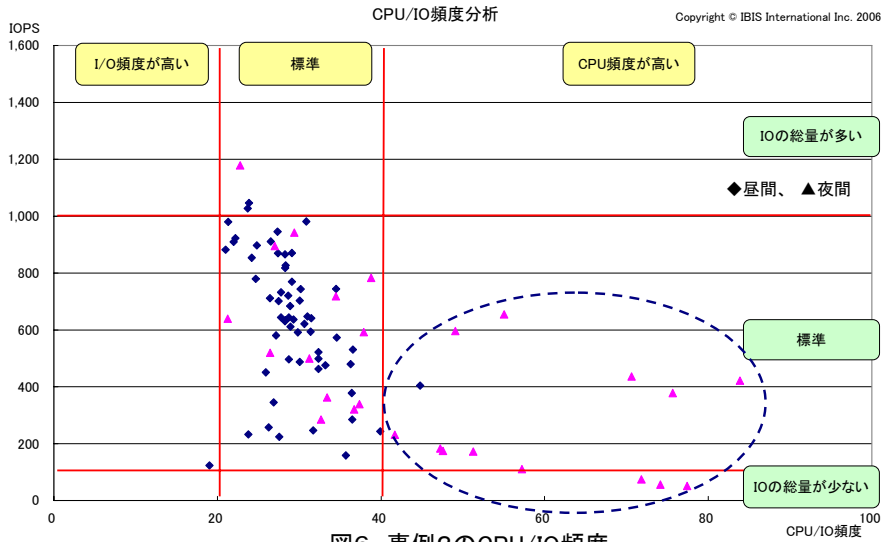


図6 事例2のCPU/I/O頻度

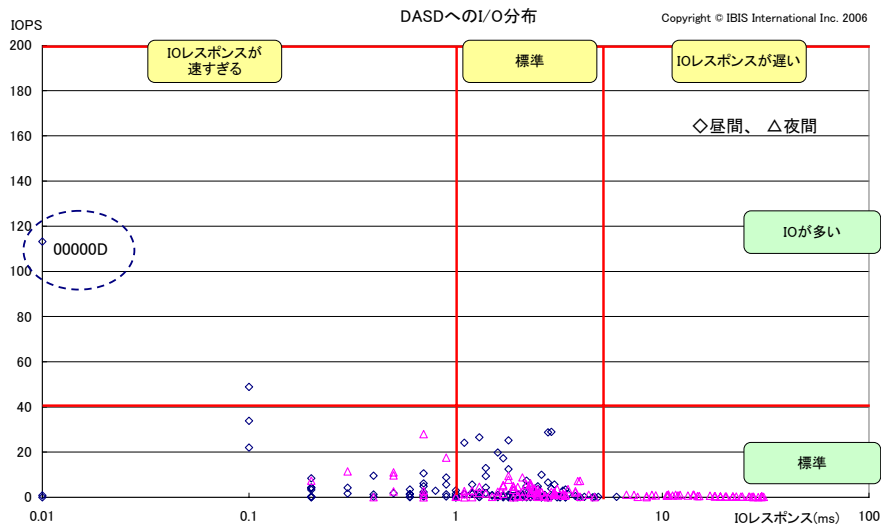


図7 事例2のDASDのI/O分析

- ・ 図6 : 昼間◆の CPU/I/O 頻度は 20~40 に入っており良好である。
- ・ 図6 : 夜間▲は 40 以上が多く CPU ループに近いジョブが存在する. ... A
- ・ 図7 : 昼間◇は 00000D に I/O が多く I/O レスポンスは 0.1ms 以下で非常に速い.
 ... B ⇒ A と B については原因究明が必要である。
- ・ 参考 : R/V 値=0.71, ページング 1 回未満 ⇒ 搭載メモリ量は妥当である

オンライン処理の代表的な見える化ツール「レスポンス分布」を図8に示す。

各プログラムの処理時間（平均，最大，総時間）と処理件数の4項目を使い，問題プログラムが一目でわかるように工夫している．図8では丸が大きく，赤く，右上に位置しているプログラム（点線内）に着目する．

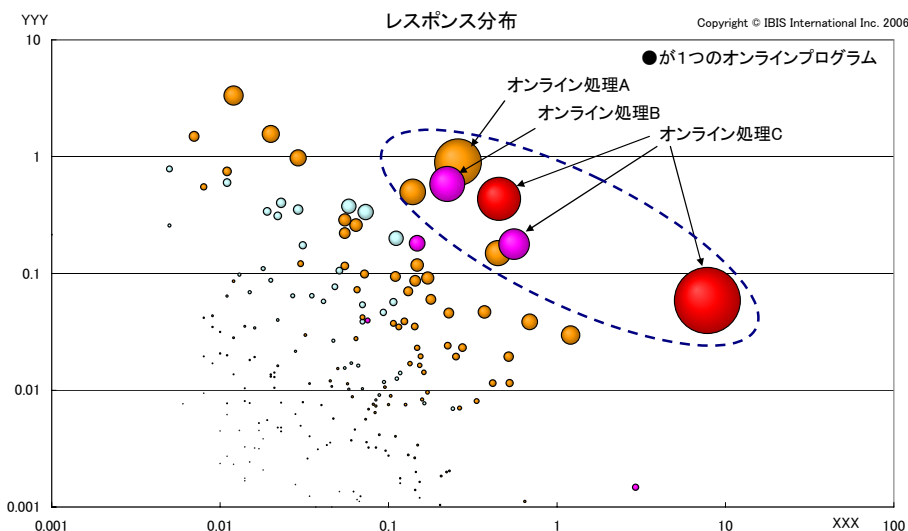


図8 事例2のレスポンス分布

オンライン処理も DASD と同様に，数 100 本の全プログラムを管理することは非効率である．レスポンス分布で全体を把握しターゲットを絞ることが重要である．

(3) 問題抽出と原因分析

仮説検証型性能解析により，根本原因の究明を行った（表3）．

表3 問題点とその原因（事例2）

問題点	主な調査データ	原因
①オンライン処理 A が遅くなった（ユーザの申告あり）	トレース	VSAM カタログへの I/O が多い→問題点②へ
②VSAM カタログへの I/O が多い	トレース ヒアリング	VSAM アクセスルーチンで毎回 OPEN/CLOSE をしている．
③00000D への I/O が多い	トレース	VSAM カタログへの I/O が多い→問題点②へ
④オンライン処理 B が遅い（ユーザの申告あり）	開発機で再現 トレース	データベース X への I/O が多い→問題点⑤へ
⑤データベース X の I/O が多い	開発部隊に調査依頼	不要なデータベースアクセスをしていた．
⑥CPU 頻度が高い	SMF AIM 課金統計	CPU 依存度が高い（CPU 時間が長く，CPU 占有率が高い）ジョブが複数存在．→問題点⑦へ
⑦特定のジョブで CPU 依存度が高い	ヒアリング	新規に追加した業務プログラムに問題ありそう．開発部隊預かり．
⑧オンライン処理 C, D の処理が遅い（図8でDの事象は発生していない）	AIM 課金統計 SMF	特定の処理でデータベースへアクセス（マクロ発行回数）が多い→問題点⑨へ データベース Y の排他待ち（仮説）→問題点⑩へ
⑨C で特定の処理でデータベースアクセスが多い	ヒアリング	多分，業務の仕様である．開発者が継続調査．
⑩D がデータベース Y で排他待ちが発生	ヒアリング	参照系のデータベースを定義ミスにより更新モードでアクセスしていた

(4) 主な改善項目と効果

- ・ VSAM アクセスルーチンのプログラム修正を実施。(オンライン処理 A)
⇒ レスポンス (実測) 2～3 秒→1 秒以下に改善
- ・ 不要なデータベースアクセスを無くすプログラム修正を実施。(オンライン処理 B)
⇒ レスポンス (実測) 10～20 秒→1 秒に改善
- ・ データベースのアクセス定義を修正。(オンライン処理 D)
⇒ レスポンス (実測) 最大 700 秒以上→1 秒に改善

3. 3 事例 3 : 次期 CPU 選定のアプローチ

継続中のプロジェクトであるが、お客様も SE も悩んでいる CPU 選定のプロセスが参考になるので紹介する。

(1) システムの概要

- | |
|---|
| ・ CPU : PRIMEFORCE メモリ : 512MB DASD : GR820 |
| ・ 業務 : 販売系システム (オンライン+バッチ) |
| ・ 課題 : 次期 CPU の機種選定 目標 : CPU 性能 30%以上削減 |

(2) 性能指標値 21 による評価

ホスト業務をサーバに切出しているため、資源にはかなり余裕が出ている。
従来だと「性能上問題なし」となるが、次世代性能指標を使うと以下のようなになる。

- ・ CPU/I/O 頻度 : 38 (20 以下 : 0%, 20～40 : 64%, 40 以上 : 36%)
⇒ CPU を使いすぎるプログラムが動作することがある。
- ・ 平均 CPU 使用率 : 16%, 10 分平均 CPU 使用率 : 30%未満が全体の 90%
⇒ CPU は過剰状態。
- ・ IOPS : 100 以下 平均 I/O レスポンス : 2.9ms
⇒ 小規模システム
- ・ R/V 比 : 搭載メモリ 508MB / 仮想記憶サイズ 244MB = 2.08, ページング 0 回
⇒ メモリは過剰状態。

(3) 次期 CPU の考え方

潤沢な資金があり、リスクを取りたくないお客様はメーカー推奨の移行パスに従って CPU を選定して頂きたい。CPU が余っているからといって CPU の能力を落とすと処理は必ず遅くなる。

同じ能力の CPU に移行するケースをよく見かける。ジョブの特性にもよるが、一般的に 2 割のジョブは CPU 時間が短縮し、2 割のジョブは CPU 時間が増加することをお客様も SE も認識して欲しい。最悪のケースは目に見えて遅くなり、移行後に問題となる。

CPU の能力を落とすことは、品質を下げることではない (手抜き建築とは違う)。今まで過剰だったものを最適化する当然のプロセスであり、信頼性の高いメインフレームだからこそできるものである。メモリについても全く同じことが言える。

最適な CPU を選定する上での留意点を以下に紹介する。

- ◆同等性能の CPU 移行であっても、
 - ・現状を分析し、基礎データを押さえておく。(PDL, SMF の解析は必須)
 - ・現行システムに大きなボトルネックがあれば必ず改善する。
 - 改善できないなら、移行パスに従うべき
 - ・2割の処理は遅くなることを覚悟する。
- ◆CPU 能力を下げたいなら、
 - ・性能予測を行うために、精度の高い性能モデルを複数作成する。
 - ・性能モデルを使ってデジタルに考える。(直感は当てにならない)
 - ・DASD の高速化、システムチューニングにより性能向上を図る。
 - ・遅くなることのリスクを分析する。
- ◆メーカー推奨の移行パスに従っても、
 - ・全ての処理に性能向上を期待してはいけない。
 - ・I/O が速くなりすぎると、一部の処理が遅延する可能性もある。

(4) 性能モデルと性能予測

現行機種を CPU 能力 70%の機種に単純移行することを考える。内蔵 DISK 装置を使っているため I/O 性能は向上する。

平均 CPU 使用率 16%が $22.9\% (= 16 \div 0.7)$ に増加するような簡単な話しではなく、CPU 時間の増加によりバッチジョブの処理時間が遅延する。

性能モデルは I/O 時間、CPU 時間、CPU 待ち時間から作ることができる。バッチ処理の多重度 1 ~ 3 で性能予測し、見える化した結果を図 9 に示す。

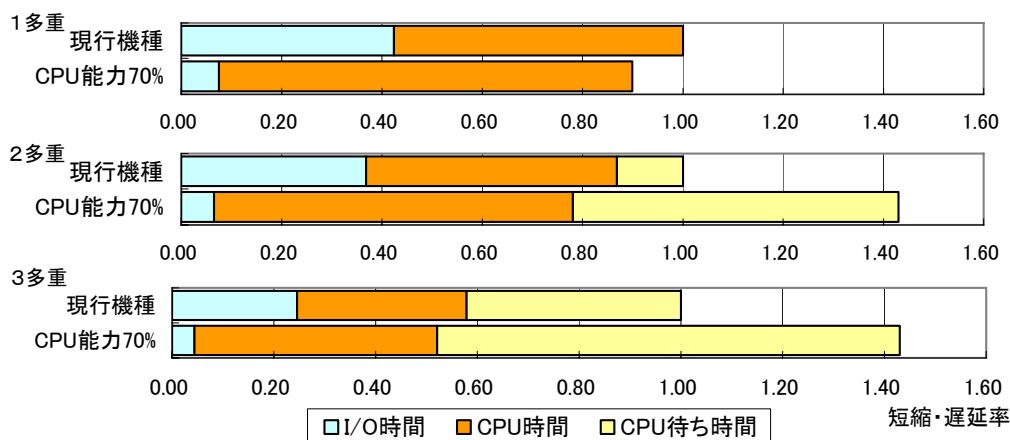


図9 バッチ処理の性能予測 Copyright © IBIS International Inc. 2006

- ・処理時間は 0.9 ~ 1.43 倍となる。オンライン処理は 1 多重モデルに近似する。
- ・現在の CPU 使用率のほとんどが 30%未満ということから、バッチ処理は常時 1 多重しか動いていないことがわかる。
- ・即ち、ほぼ同等性能 (最良 0.9 倍) が期待できるが、バッチジョブが重なると処理時間は最悪 1.43 倍になってしまう。

(5) 課題

- ・CPU の無駄をなくすチューニング → CPU/I/O 頻度の 40 以上が 36%の対応
- ・バッチ処理の重なり具合の把握と、処理時間遅延の業務への影響調査

4. 評価

(1) お客様視点・現場視点での性能評価

お客様は性能品質（例. あのジョブ動かすと他が動かなくなる，移行したら遅くなったなど）について敏感である．性能指標値 21 を使い，実際に数字で説明すると皆さん大変納得され，原因や改善方法も色々思い浮かぶようである．

資源の無駄については，利害関係や当事者か否かにより反応は様々である．

(2) 性能の見える化

お客様にわかって頂くために，性能評価プロセス（図 2）を透明化し，その中でも見える化に力を入れてきた．CPU/IO 頻度分析（図 4）は今までなかった手法であり，お客様だけでなく SE から何が問題なのかわかりやすいと好評である．

全体を俯瞰して細部を見る，森を見て枝葉を見るバランス感覚が自然と生まれてくる．

(3) スピードアップ

性能問題の調査は，メインフレームでも従来から時間のかかるものであった．

これ打破するために，仮説を設定し，トレースを使って検証していく仮説検証型性能解析を生み出した．事例 1 も事例 2 も，トレースの取得・解析～原因の究明までは数時間で完了し，かつ想像もしていなかった原因が出てきてお客様が驚かれるケースも多い．

5. 今後の課題

一年前の論文（参考文献 B）で課題を 3 点あげており，これを継続して進めていきたい．

- | |
|---|
| <ul style="list-style-type: none">①性能のサービスレベル向上のためのお手軽相談，お客様間での情報交換②コスト削減のため，OS やサブシステムのオーバーヘッド削減（富士通への依頼）③チームでの性能管理の実現 |
|---|

①については，ホームページ上に専用ページを作り，お客様や SE に技術情報の提供，性能無料相談の場を立ち上げ運用している．次ステップとして，富士通プラットフォームソリューションセンターでの研修会を半年ほど前から富士通関係者と検討をしている．本論文から具体的な内容も取り入れ実現したい．FUJITSU ファミリー会内での会員様との交流も期待していたが，チャンネルが見つからず残念ながら休止状態である．

②については，簡単に製品開発に届かないが，現場の声として言い続けいく．

③については，性能評価プロセスが明確になったので，チームとして推進する方法を継続して考えていきたい．

6. おわりに

先日，書斎を整理していたら私が 1992 年に執筆した性能評価に関する論文が出てきた．気づいたことが 2 点，

- ・今でもほとんど使える内容で，掲載している事例に最近も直面している．

・この14年間、性能評価手法は表面的にはほとんど進化していない。

このことは、冒頭に述べた性能指標値が変わっていないこと、性能評価はSEの仕事だという意識にも関係している。

性能評価は以前から特定の人に丸投げすることが多い。どんなに優れた性能評価ツールを導入しても、大量のレポートとコメントの作成ツールになっただけでは意味がない。

本論で記載した、性能評価プロセスをお客様とSEの真ん中に置き、見える化の智慧を出しあい、必要な場面で目的（コスト削減、性能改善など）にあった性能評価～改善に取り組んで頂くことを期待する。

今、一番気をつけなければいけないのは、性能品質の悪いアプリケーション群である。これらは管理できないし、私自身はウイルスと同類だと考えている。どうすればよいか、

- ・入ってこないように予防する（受入れ時に性能検証する）
- ・定期的に（ウイルス）チェックする
- ・不幸にも見つかったら駆除する（運用に入ったら大抵は駆除不能）
- ・駆除できないときは、開発者にパッチを作ってもらい適用する
（事実を認めさせるのに時間がかかり、パッチを作るのに更に時間がかかる）
- ・パッチがだめなときは、この事実を公表し、自分が加害者（他業務が遅くなる、追加のハード費用がかかるなど）にならないよう、ひっそりと運用する

論文を読んで頂いたあなたと意見交換ができることを楽しみにしております。

以上

用語説明

PDL/PDA	性能評価ツール, Performance Data Logger/Performance Data Analyzer
SMF	課金情報, System Management Facility
GTF, STF0	OSの内部トレース
GS21, GS, PRIMEFORCE	富士通メインフレームの機種名
MSP, XSP	富士通メインフレームのOS名
カタログ	ファイルの管理簿

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 年前期論文