
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 －

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 ■

大型メインフレーム GS21 600 で動く 10 時間の夜間バッチ処理を 3 時間以上の短縮に成功。オンラインサービスの延長，リカバリ時間の確保等に加え，CPU 削減により年間 2,000 万円以上のコストメリットも生み出した。地道なチューニングとともに，性能管理・改善の課題を以下の 3 点と設定しプロジェクトを推進した。

- ① 性能管理者自身の常識のブレークスルー
- ② 管理者視点からお客様視点でのブレークスルー
- ③ 個人プレイからチームプレイへ

従来の常識を破ることが最大のポイントであり，ブレークスルーしていく過程①・②は事例を使って解説する。技術的に見ても富士通社内でも例の無いものが多い。③は継続中の課題として，ハイパフォーマンスな性能管理チーム作りの興味深い取り組みを紹介する。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 当社の概要	
1. 2 背景	
2. 性能管理・改善の課題	《 4》
3. 性能管理者自身の常識のブレークスルー（課題①）	《 4》
3. 1 少ない工数で最大の効果が得られる改善から始める	
3. 2 誰にでも効くチューニング項目	
4. お客様視点でブレークスルーしたチューニング事例（課題②）	《 6》
4. 1 SORT が CPU を多量に使うのは仕様である	
4. 2 磁気テープへのバックアップが遅いのは当然である	
4. 3 ユーザアプリケーションのチューニングはむずかしい	
5. チームプレイの性能管理を目指して（課題③：進行中）	《 8》
6. 評価・効果	《 9》
6. 1 コスト効果	
6. 2 サービスレベル向上	
7. 今後の課題	《 11》
8. おわりに	《 12》

■ 図表一覧 ■

図 1 性能管理の課題の推移	《 4》
図 2 夜間バッチ処理時間削減へのアプローチ	《 4》
図 3 性能改善の効果と時間	《 5》
図 4 VSAM バッファサイズと CPU 時間の増減	《 5》
図 5 SORT の処理時間と CPU 時間の推移	《 6》
図 6 研修の目的	《 8》
図 7 ペアワークの Exercise	《 8》
図 8 参加者の評価	《 9》
図 9 チューニング前 夜間バッチジョブ稼動状況	《 10》
図 10 チューニング後 夜間バッチジョブ稼動状況	《 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/>）

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、お客様が富士通メインフレームをより快適に、そして適切なコストで使って頂くためのノウハウを提供している。

1. 2 背景

(1) メインフレームを取り巻く環境

メインフレームの現況について簡単に紹介する。

基幹システムのオープン化は徐々に進められているが、様子見の企業が多いのも事実である。2004 年度の出荷状況は前年度比数%減に留まっている。

- | |
|---|
| <ul style="list-style-type: none">・(社)電子情報技術産業協会調査 日本でのメインフレーム出荷状況 (2004 年度)
出荷台数 1,212 台 (前年度比 98%), 金額 2,467 億円 (前年度比 94%)・富士通メインフレーム稼動ユーザ数 約 4,000 社 (2004 年前半) |
|---|

プラットフォームは、IBM が zSeries をオープンメインフレームサーバと位置づけ、富士通は基幹 IA サーバ PRIMEQUEST を出荷した。マスコミの論調も「メインフレーム vs. オープン」から「次世代メインフレーム vs. 基幹系サーバ」にシフトしつつある。

(2) 今までの常識が通用しない

性能コンサルティングの最終報告の場で、お客様のトップから次のような話を頂戴した。「恥ずかしい話したが、最近のシステムは、エンドユーザの目線で性能が考えられておらず問題が起きている。担当者はマニュアルにある目先の対策しか打てない。今までの常識が通用しなくなっている。今回の改善事例をメインフレームだけでなくオープンの担当者にも伝えて欲しい。」

後日、このお客様では設計や性能に関する教育が実施されていることがわかった。では常識とはいったい何なのか。そもそも私たち自身の常識も壊さないといけない時期にきているのかもしれないと考えた。

(3) パフォーマンス改革 2004 からの進化

2004 年下期の論文「メインフレームのパフォーマンス改革 2004」では、「適切な性能管理と改善をすることにより、オープン化よりもコストを削減し、性能・信頼性の高いシステムをお客様主導で構築して頂ける」ことを説明した。

本論文では、性能管理の現場が直面している問題、これをブレイクスルーしていく過程を紹介したい。

2. 性能管理・改善の課題

性能管理の重要性は20年以上前から言われており、最近ではITIL (IT Infrastructure Library)でも定義されている。性能(キャパシティ)はお客様に提供する重要なサービスでありコストを決定づけるものである。Webシステムを中心に、性能測定ツールもたくさん提供されてきた。将来、性能管理者がこれらのツールに**使われている姿**が想像できる。

性能管理の重要性に気づきながらも、問題は長年先送りされてきた。属人的、知識が継承されない、現場で使える人が育たない。トラブルが発生すれば、結局10年前と同じ人が対応していることもある。性能管理の課題を図1のように捕らえている。

- 課題① 性能管理者自身の常識のブレークスルー
 - 課題② 管理者視点からお客様視点でのブレークスルー
 - 課題③ 個人プレイからチームプレイへ

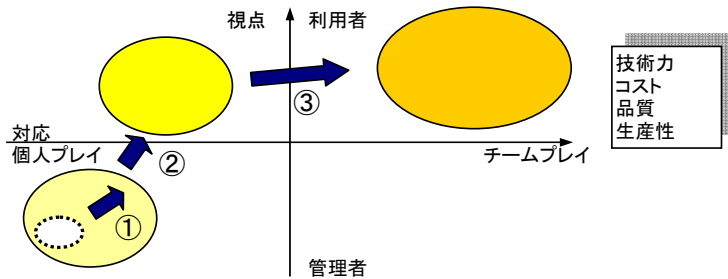


図1 性能管理の課題

3. 性能管理者自身の常識のブレークスルー (課題①)

3.1 少ない工数で最大の効果が得られる改善から始める

夜間バッチ処理時間を削減するには図2に示すような多岐のアプローチがある。私たちは改善目標と改善効果を予測しながら短期的・中期的な施策を立案していく。

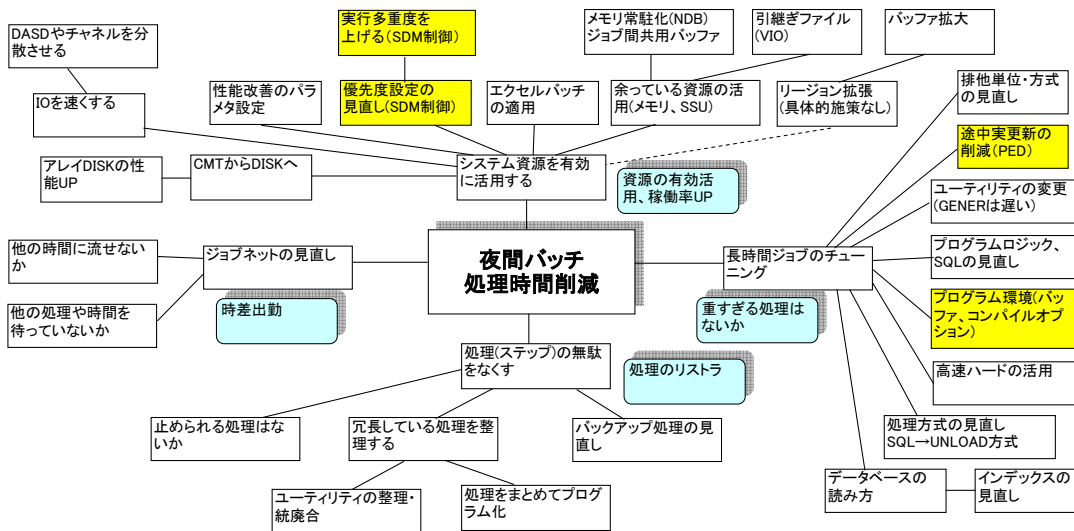


図2 夜間バッチ処理時間削減へのアプローチ

性能改善（チューニングという）は、図3の目標曲線のような「少ない工数で最大の効果」が得られる項目から行うことが多い。

今回、関連部署間（複数の業務開発、運用統括、アウトソーシング）の連携がうまく取れず、特定部署にクローズした「少ない工数で最大の効果」を目指したため、一ヶ月経過しても実線に示す効果しか得られなかった。

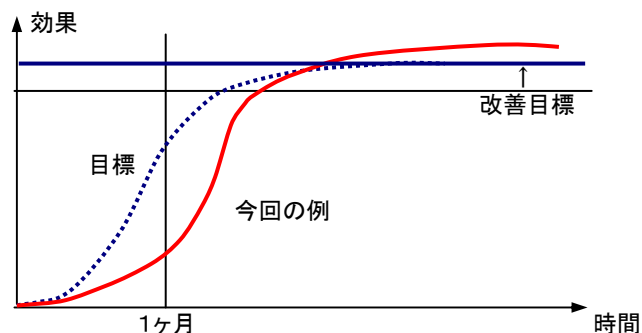


図3 性能改善の効果と時間

本件は、関係者が改善目標を再認識し、連携を強化し、4.1以降述べるブレークスルーにより目標はクリアすることができた。「少ない工数」を「組織間の壁を壊さない」と解釈すると今後も陥りやすいケースである。

3.2 誰にでも効くチューニング項目

メインフレームでは、チューニングの王道のようなものがある。

- ・ページバッファ（VSAM バッファ）を拡張し、途中実更新を防止する

図4はVSAM バッファを拡張（省略→1.4MB→6.4MB）したときの、CPU時間の増加の推移である。ジョブEは、数MB バッファを拡張したとこで、CPU時間が最大67%も増加してしまった。

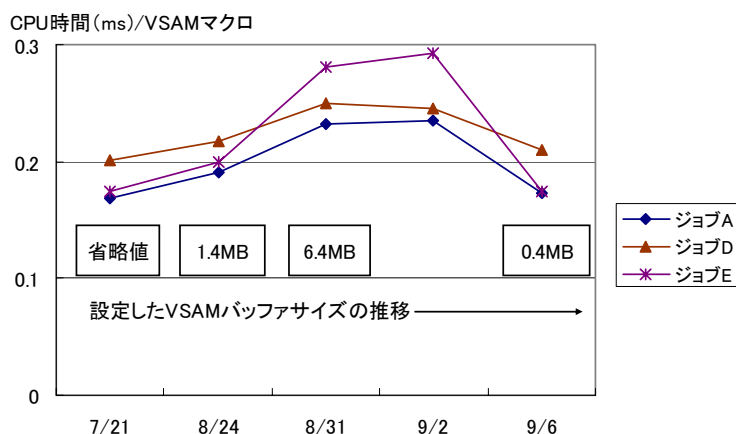


図4 VSAMバッファサイズとCPU時間の増減

現在は、省略値では管理しにくいいため0.4MBが最適値と判断し問題なく運用している。

搭載メモリの増加により、数MBのバッファ拡張はデメリットがほとんど無いため推奨してきたが、CPU使用量の増減は留意する必要がある。

以上、2つの事例だが、私たち自身の常識も見直す必要がある。

4. お客様視点でブレークスルーしたチューニング事例（課題②）

専門的な内容になるが、常識として扱われていた事象をお客様の目線で疑問を持ち、性能改善した事例を3つ紹介する。 ※重要度を最大★3つで表す

4.1 SORTがCPUを多量に使うのは仕様である（重要度★★★）

(1) 従来の常識

ソート処理（SORT）は3つのポイントで処理速度の向上を図ってきた。

- ①アルゴリズムの改善
- ②入出力処理の高速化：読書きのI/O処理をブロック単位からシリンダ単位へ
- ③ワーク処理の高速化：並び替えに使うワークをDASDからメモリへ

特に、③は遅いDASDアクセスの変わりに大きなテーブルを持ち、速いメモリアクセスで処理を行うためCPU時間が延びるのは当然である。

(2) お客様の視点

CPU：GS21 600/10（富士通最高速CPU） 処理：600MBの単純なソート
処理時間 150～200秒（処理能力3～4MB/S）、CPU時間 77～86秒（占有率41～51%）

CPU占有率40%は限りなくループである。なぜなら、CPU性能が1/10の中型機だとCPU時間は10倍超になり、占有率は90%を超えてしまう。こんな処理性能で基幹業務はできない。

(3) 仮説とチューニング結果

仮説：③のメモリアクセスでCPUを異常に使い過ぎている。

SORTで使うメモリを少なくする。具体的には拡張REGIONにテーブルを作らない。実行JCLでPARM='AESIZE(OK)'と設定する。

検証結果：図5の通り、処理時間はほとんど変わらずCPU時間は約60%も削減した。

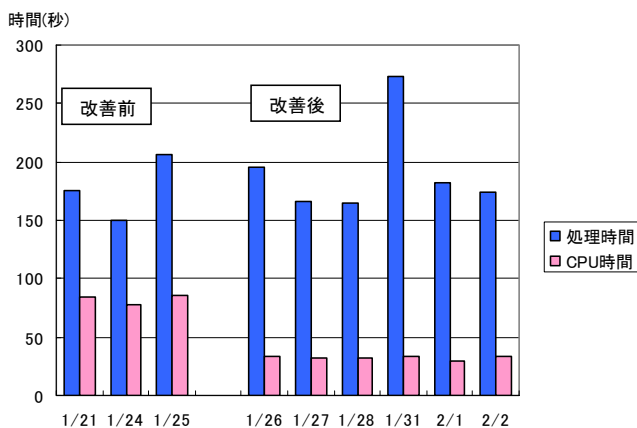


図5 SORTの処理時間とCPU時間の推移

考察：前述3.2で紹介したVSAMバッファの件も同様だが、DASD性能の飛躍的な向上により、メモリアクセスよりDASDアクセスの方が効率よく処理できるケースが出てきている。

CPU占有率の高いプログラム、ユーティリティ、サブシステムはその原因を調査し、改善を進めるべきである。

4. 2 磁気テープへのバックアップが遅いのは当然である (重要度★★★)

(1) 従来の常識

DBのバックアップ処理(TDUMP)を速くしたいなら、DISKに一度バックアップを取り、非同期で磁気テープに退避するのが常套手段である。

(2) お客様の視点

DISKに余裕があれば苦勞していない。何かアイデアはないのか。

(3) 仮説とチューニング結果

仮説 : バックアップの出力量を削減するため、TDUMPの圧縮処理を利用する。実行JCLでDATA=COMPと指定する。

注) 昔、磁気テープの本数を削減するために作られた機能だが、カートリッジテープ(CMT)装置の容量が大きくなったため使われなくなった。

検証結果 : 実際のデータで実測したところ平均圧縮率 15%と良好。(85%削減)

CPU時間は圧縮処理のため2秒→10秒に増加したが、処理時間は3分前後でほとんど変わらない。バックアップ時間はテープと比べると 1/10 以下 (1時間以上) に短縮する見込みがたった。

考察 : バックアップ中はバッチジョブの負荷が高くないため、CPUが若干増加してもシステム上問題なかった。

4. 3 ユーザアプリケーションのチューニングはむずかしい (重要度★★)

(1) 従来の常識

ユーザアプリケーションで性能上の問題を特定することは難しく、改修してもレベルダウンを起こさないためのテストが大変である。極力行わないことが望ましい。

(2) お客様の視点

ユーザアプリケーションでも、CPU占有率が40%を超えればほとんどループである。こんなプログラムが動いたらCPUがすぐに足りなくなってしまう。

(3) 仮説とチューニング結果

仮説 : 処理ループ、多量のテーブルサーチ、無駄なDBマクロ発行(SQL含む)、デバックオプション付加などが考えられる。

検証結果 : 日次夜間バッチプログラムでキー重複を防ぐため、時間取得のACCEPT命令のループを発見 [日次バッチ1本、週～月次バッチ数本]

CPU時間 17分37秒 ⇒ 5秒に改善

ボリューム圧縮のプログラムが常時に実行されている [毎日]

1時間当たり200回実行を半減 CPU時間 3分30秒/H ⇒ 50%削減
プログラム内のテーブルサーチをVSAMに変更 [日次バッチ数本]

処理時間 1060秒 → 1642秒, CPU時間 437秒(41%) ⇒ 273秒(17%)

考察 : GTFトレースを解析しプログラムの動きを外部から把握する。これを基にプログラムの動きがわかる人にヒアリングをする。答えは開発者が知っていることが多い(気づいていない)。

業務上最重要でないプログラムほどチェックされないケースが多い。

5. 個人プレイからチームプレイの性能管理（課題③：進行中）

「今回の改善事例をメインフレームだけでなくオープン担当者にも伝えて欲しい。」というお客様からの依頼を、第1ステップとして半日の研修を開催した。

実施回数：3回、出席者延べ：50名、SE経験2年～部長クラスまで

研修の目的を図6に示す。従来の知識供与型研修だけでなく、参加者が「性能」にどっぷり触れられる場を提供したいと考えた。

会社では教えてくれないシステムチューニング

性能に関する教育コースは既に用意されている。
この研修は、何が違うの？

技術、ノウハウを伝える(知識供与型)

暗黙知

問題意識
危機意識
当事者意識

・性能に関する知識
・プラットフォームに関する知識
・システム構築(設計～テスト)に関する知識
・ナレッジデータベース、Web

想像・創造
智恵
マインド
コミュニケーション

基本から最先端情報まで！

×
やってくれなきゃ意味がない
価値がなければ意味がない

図6 研修の目的

(1) 研修で工夫した点

- ・ペアワークを使い、参加者自らが考え、話し、聴き、そして気づきを共有する。
- ・経験の浅いSEからベテランまで、ホストもオープンも前提条件なしに一緒に行う。

ペアワークは次のように進める。

- ① 座席の隣の人とペアになる。できるだけ知らない人とペアになるようにする。
- ② 質問を決められた時間内で考え、回答を用紙に記入(メモ)する。
- ③ ペアで話し役と聞き役を決め、話し役は決められた時間いっぱい回答を話す。聞き役は意見を言うてはいけない。話し続けてくれるよう働きかけをする。
- ④ 話し役と聞き役をチェンジして行う。
- ⑤ 全員で気づきを共有する。講師は気づきを掘下げ、技術的なフォローをする。

図7にペアワークの例題を2つ紹介する。皆さんにも考えて頂きたい。

Exercise 1

◆状況

あなたはAシステムの責任者です。
今、Aシステムの「CPU使用率が100%になった」
との報告を部下から受けました。

◆質問

あなたはどうか判断し、どんなアクションをとりますか。
(複数アクション歓迎)

◆時間

用紙記入(3分)、ペアワーク(2分ずつ)

Exercise 2

◆質問

あなたが携わっているシステムで、3年前と比べて
性能が良くなっている順に並べてください。

- | | |
|-----------------|---------|
| ①CPU | ②メモリ |
| ③DISK | ④ネットワーク |
| ⑤磁気テープ | |
| ⑥システム記憶SSU(ホスト) | |

◆時間 用紙記入 2分 ペアワーク 1分

図7 ペアワークのExercise

Exercise 1 からは以下のような気づき生まれる。

- CPU 使用率の本質は何なのか。
- 視点は6つ以上ある。
- 1つの視点からも 180 度違う見方もできる。 例) CPU100%は善か悪か

Exercise 2 からは以下のような気づき生まれる。

- ハード性能の本質は何なのか。
- システムを取り巻く環境はみんな違うし、多くは変化し続けている。
- 1番2番が同じペアは1割くらいしかいない、予想以上にみんなバラバラ。

共通して伝えたいことは、

- 知らない人とも簡単に意見交換ができる。
- 一人で長時間考えるより、短時間で考え、人に聞いてもらうとことで自ら新しい気づきを生むことができる。

(2) 参加者の評価

研修終了後にとったアンケート結果の一部を図8に示す。(n=50)

参加者にとって予想外のエクササイズだったと思うが、8割以上の方は自分なりの気づきを見つけてくれたようだ。

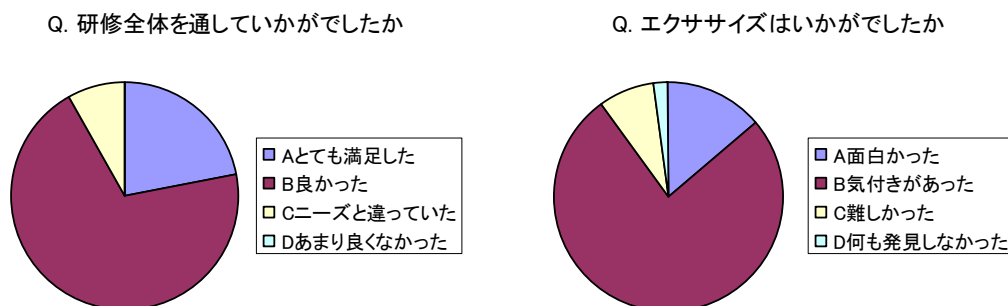


図8 参加者の評価

6. 評価・効果

6. 1 コスト効果

実際のコスト削減には至っていないが、チューニングによる CPU 時間の削減により、現在開発中業務のリリース時、必要となる CPU コストを抑止することができた。

机上計算だが、抑止できたコストは以下のようなになる。

前提条件：夜間バッチ時間7時間（21:00～4:00；図10）を遅延させない

- CPU 使用率 21:00～4:00 まで常時 100% 即ち CPU 時間は7時間
 - 削減できた CPU 時間 30分
 - CPU 本体の概算レンタル費用 3,000 万/月
- 年当たり $12 \times 3,000 \text{ 万} \times (0.5H / 7H) = 2,571 \text{ 万/年}$ (ソフト・保守費用含まず)

6. 2 サービスレベル向上

- (エンドユーザ) オンラインサービス終了時間 21時 ⇒ 23時 に2時間延長

- ・(運用部門) 夜間のリカバリ時間 1～2時間 ⇒ 3～4時間 (+2時間)
- ・(運用部門) 夜間バッチ時間を更に1～2時間短縮項目を保有
- ・(運用・開発部門) 性能に関する意識向上 (まだまだ入り口だが)
- ・(アウトソーシング部門) 複数システム間で競合している MTL 装置の負荷軽減

夜間バッチ時間の短縮について簡単に説明する.

(1) チューニング前の状況・・・図9

21:00 からおよそ 7:00 まで稼動. 2時間短縮を目標にチューニングを開始.

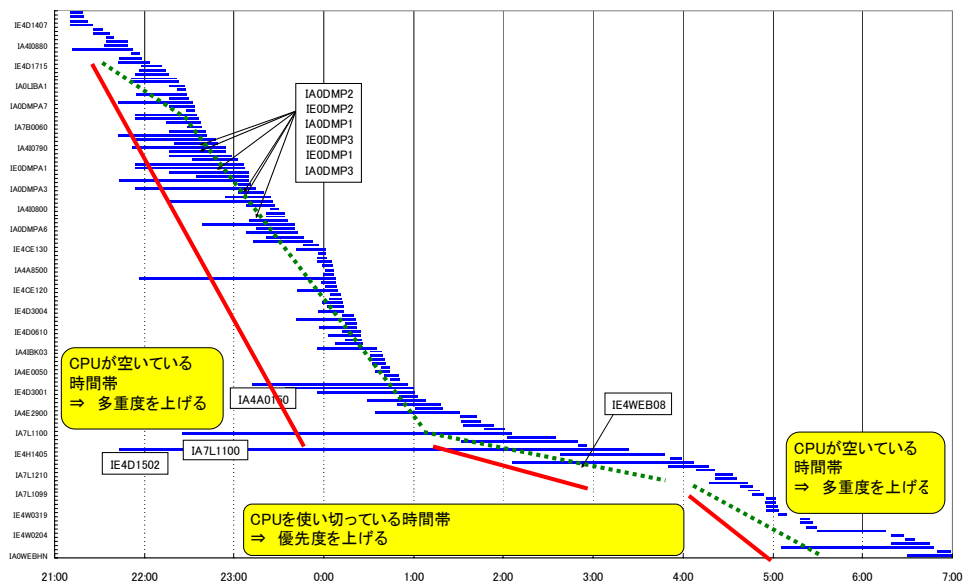


図9 チューニング前 夜間バッチジョブ稼動状況 (2004年11月1日)

(2) チューニング後の状況・・・図10

終了時間が4:00まで短縮した状態. この時間帯のCPU使用率は常時100%.

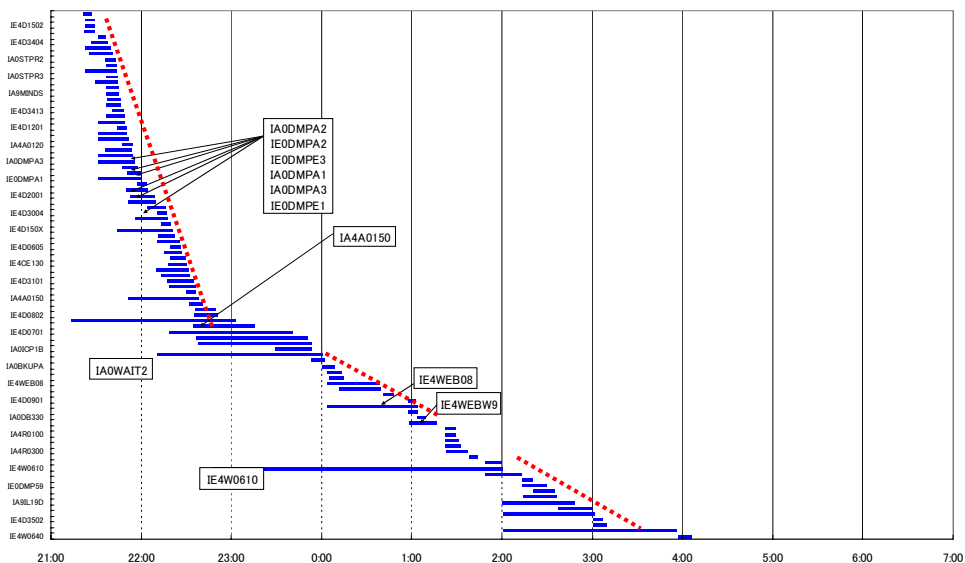


図10 チューニング後 夜間バッチジョブ稼動状況 (2005年2月10日)

前述した内容以外に実施したチューニング項目

- ・エクセルバッチを導入し、ワークボリュームとしての MTL の運用を無くす
- ・ファイルコピーを GENER から REPRO に変更
- ・システムパラメタ(SDM)の最適化
- ・CPU 占有率の高いプログラムのロジック見直し
- ・急ぎでない処理を、負荷の高い時間から別の時間帯へ移動 等

結果、夜間バッチ処理時間を3時間以上短縮し、オンラインサービス時間を延長した。

2月10日時点では、バックアップの DASD 化、23～0時の待合わせ時間の改善はまだ行っておらず、CPUに余裕があれば更に1～2時間の短縮は実現できる。

7. 今後の課題

7. 1 メインフレームを戦略的 IT として活用するために

私の PC (3年目)は最近異常に遅く、動作も不安定、作業の生産性が低下し、私自身のストレスも溜まってきた。どうすればよいのか的確なアドバイスをくれる人がいない。たかが1台 PCでも大問題である。

企業の基幹システムを支えてきたメインフレーム。実は多くのお客様が満足のいかないサービスレベルに慣らされ、高すぎるコストを支払っていると私は認識している。

1. 2背景で述べたように、今後もメインフレームは使われていく。メインフレームであれレガシーであれ、お客様にとってより戦略的 IT として活用されるべきである。今後の課題を以下のように捉えている。

- ・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

現状： なし

対応策：FUJITSU ファミリ会だからこそ実現できるのではないのでしょうか。

弊社は、富士通様共通技術部門と意見交換中。

- ・コスト削減のため、OS やサブシステムのオーバヘッド削減

現状： SymfoWARE, FTP, 本論で紹介した SORT, 前の論文で紹介した排他制御等、業務ロジック以外で CPU を使いすぎる。

対応策：説明は割愛するが IBM zSeries で提供している zAAP(zSeries Application Assist Processor), On/Off CoD(キャパシティ・オン・デマンド)に相当する機能を富士通様には提供して頂きたい。

弊社は、FUJITSU ファミリ会の論文等で積極的に情報を公開していく。

7. 2 チームでの性能管理を実現するために

5で紹介した研修も、現場に持ち帰って使われなければ意味がない。

- ・プロジェクトで実践できるよう、リーダに継続的なフォローをする

その他、チームのパフォーマンスを上げるために以下のようなものが効果的である。

- ・ベテラン技術者のノウハウ継承 → ワークショップなどを工夫する
- ・通常業務でペアワークの導入

8. おわりに

2004 年下期と今回，2 回にわたりメインフレームの性能改善事例を紹介した。

10 数年前，アレイディスクが出荷されると I/O 性能が 1/10 (30ms→3～5ms) になり性能の考え方が著しく変化した。速くなりすぎて性能問題も発生した。

最近，DISK が更に速くなり 1ms を切るようになると，また性能の考え方が変わってきた。メインフレームであっても簡単に CPU 使用率が 100%になってしまう。今までの性能管理だと問題が起きる可能性が高くなっている。

メインフレームの 3 年前の常識は通用しなくなってしまった。

お客様が余裕のある CPU を導入して頂ければ何ら問題はない。しかし，IT コスト削減は目的でなく前提条件になっており，ギリギリの CPU を入れようとする。そして問題が起き，重大な機会損失や社会的問題を誘発してしまう。

こんな状況下で企業が競合他社よりも優位に立つには，性能管理のブレークスルーが必要と考えた。当然メインフレームだけの話しではない。お客様の視点で，チームとしてどれだけパフォーマンスを上げられるかが継続中の課題である。

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

参考文献

- (1) 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 年下期論文