



## キャンドル コンピュータ レポート

**Vol.14 2004年7月号**

[http://www.candle.co.jp/news\\_event/report/index.html](http://www.candle.co.jp/news_event/report/index.html)

**日本キャンドル株式会社**

# 目次

---

## ■ Feature

ワークロード マネージャー - 群れを監視する .....	3
ゴール モードでの考慮点 .....	3
サービス クラスの管理 .....	4
1つのアドレス空間に異種のワークロード .....	5
移行支援としてのペロシティ .....	6
迷子の羊を探す .....	6

## ■ Case Study

アプリケーションのインフラストラクチャーに投資して、順調に利益を上げた National Australia Group .....	7
Candle のツールとサービスでチャネル展開を加速化 .....	7
複雑な状況 .....	8
賢明な投資 .....	9

本誌は米国 Candle 社発行の「Candle Computer Report 2」を日本キャンドル株式会社にて翻訳・編集したものです。

## ワークロード マネージャー - 群れを監視する

……だが、いったい誰が羊番をするのか？

Don Zeunert  
(キャンドルコーポレーション/ コンサルティング システムエンジニア)

仮に特効薬ではないとしても、ワークロード マネージャー (以下、WLM) は、互換モードよりもはるかに優れており、専門家でなくても簡単に使うことができるものである。しかし、互換モードと同様、監視と修正はまだまだ必要だ。たとえ期待どおりの結果が得られない場合でも、WLM がどのような判断を下しているのかを理解する必要がある。ゴールモード環境だからといって目標値が間違っていたり、もしくは単純なゴール設定の場合、必ずしもパフォーマンスがよくなるというわけではない。運よく期待どおりの結果が得られたとしても、逆に誤った安心感を抱く恐れもあるのだ。

第一に WLM は全体の動き、例えば、大切なプログラムがうまく機能しているかどうかを監視し、していない場合はその対策を講じる。重要なプログラムを WLM にサービス クラスとして認識するアドレス空間グループ、またはエンクレーブとして定義すると、他の小さなワークロードよりも重要なことが WLM に伝わる。WLM は、サービス クラス内のすべてのアドレス空間の平均的なパフォーマンスを管理する。そのため、1つのサービス クラスが目標を達成しても、その一方で達成しないアドレス空間が生じてしまう可能性が高い。

WLM ではこうした状況を修正できず、またユーザーにも通知することができないということから、例えにとってもいえば羊が迷ってしまうことになる。そこで、最良の友である番犬の登場となるのである。つまり、パフォーマンス モニターに個々のアドレス空間を監視させて、問題が起きていないか調べさせるのだ。何を監視するのか、また、どのアドレス空間に問題が潜んでいるのか把握していないと、この作業は非常に退屈なものになる。この記事を読めば、ワークロードが見逃される理由や、WLM にそれらの動きが厳密に監視できる方法についてより理解いただけるであろう。

### ゴール モードでの考慮点

ユーザーの中には、今でも WLM のゴール モードは重要なワークロードを厳重に監視しないと考えている人がいる。一般にこれは、サービス定義が適切に定義されず、また、非現実的な目標やベロシティ目標及び異なるワークロードが同種のものとして管理されていることから、誤った管理がなされていることに起因しているのである。

顧客の中には、それまでサービス レベルを作成したことがないにもかかわらず、自社の SLA を WLM に適用しているところもある。また、WLM の互換モードのベロシティ レポートを使用して、単にこれらと同じ数値を当てはめるところもある。これではよい管理結果は得られず、多くの場合は悪い結果を招くことになる。WLM は 10 秒毎に、サービス クラスを優先順位の最も高いものから低いものまで監視して、助けの必要なワークロードがないかを探す。非現実的な期待値や目標の場合、オーバーヘッドが発生したり反応時間が遅延したりと、WLM の機能を制限することになってしまう。ベロシティによる WLM 管理は、レスポンス タイムによる目標よりも効果が低い。だが、この記事では主に、異なる ワークロードを同種のものとして WLM

に管理させることの弊害について論じることにする。

## サービス クラスの管理

### ～ 定義するサービス クラスを少なくすることのメリット～

定義するサービス クラスを少なくするには、いくつかの理由がある。主として、サービス クラスの定義が増えるほど WLM の分析も増え、修正のための遅延や、オーバーヘッドも増加する。別の理由としては、どちらかと言えばユーザーがペロシティによる目標を使用すること、また、ディスパッチング優先順位と同じだけの一意のペロシティを作成する理由がないことも挙げられる。ディスパッチング優先順位とは異なり、10 未満の差異があっても WLM にはほとんど意味がないため、そうした細分性からいえば、サービス クラスは 9 つでよいことになる。

### ～ 定義するサービス クラスを少なくすることのデメリット～

一方、サービス クラスが少ないほど、さらに多くのアドレス空間を同じサービス クラスに分類しなければならなくなる。そうすると、異種のワークロードが結合されることになる。しかし残念ながら、WLM ではこれらを同種のワークロードとして管理してしまうのである。

もうひとつのデメリットとして、通常、多数のアドレス空間を持つサービス クラスは、アドレス空間の少ないものより管理が難しい。当然ながら、サービス クラス期間内のすべてのアドレス空間は、同じディスパッチング優先順位を持つ。WLM がサービス クラスの相対的な優先順位付けを行う必要がある場合、同じ優先順位のサービス クラスに悪影響を与えずに、1 つのサービス クラス期間内のすべてのアドレス空間を他よりも上位に移動しなければならない。TSO の最初の期間に 1,000 名のユーザーがいる場合、こうした優先順位付けを調整することは不可能に近い。同じことが 100 の CICS 領域を持つサービス クラスにも言える。このサービス クラスによる合計サービス利用度が高く設定されている場合、他のサービス クラスの利用に影響を及ぼすことになる。

ただし、すべての s 100 CICS 領域が同じディスパッチング優先順位でも、処理するトランザクション ボリュームが異なること、またアプリケーションも CPU またはメモリにより制限を受ける可能性がある。そのため、領域によっては、WLM がリソースを過剰に割り当て、単純なトランザクションを却って複雑に処理する恐れがある。反対に、WLM がリソースを少なく割り当てて、レスポンス タイムが遅くなる可能性もある。すべての CICS 領域の平均パフォーマンスでサービス クラスが目標を達成できている限り、WLM がそれ以上にリソースを割り当てることはない。これを、平均、またはパーセンタイルによるレスポンス タイム目標と混同してはならないのだ。サービス クラスのパフォーマンス指数計算は、すべての待ち件数対すべての使用件数となる。サービス クラスに 100 の領域があり、その内の 1～2 個の低ボリューム領域が目標を達成していなくても、サービス クラスのパフォーマンス指数には影響しない場合もある。

CICS の相対的なディスパッチング優先順位付けでもうひとつ問題になるのは、DB2 スレッド アタッチ優先順位だ。DB2 スレッドを高く設定した場合、こうした領域の CICS メイン TCB (DFHSIP) は、DB2 を使用していないか、または CICS と同等もしくは低く設定されている DB2 を使用している CICS 領域よりも優先順位が低くなる。このため、DB2 が高くアタッチされているものはパフォーマンスが低下する可能性がある。DB2

を高く、低く、あるいは同等にアタッチすることについては多くの議論が交わされているが、ここでその全てを紹介することはできない。同じサービス クラス内に他の CICS 領域と同じように DB2 の高い領域があれば、メインの TCB は他の領域よりも低くなると言えば十分だろう。DB2 が高く付加された CICS 領域に個別のサービス クラスを作成すれば、問題は多少とも緩和される。

これで、サービス クラスが多いほど、WLM がワークロードをさらに厳密に管理でき、より柔軟性を持たせて変更しやすくなることがお分かりいただけたであろう。だが、オーバーヘッドと反応時間の問題はどうか。目標が適切に設定されていれば、優先順位の最も高いサービス クラスが目標達成に失敗することはない。目標達成に失敗していなければ、WLM は何の調整も行わない。そのため、ドナー/レシーバーロジックのオーバーヘッドや、WLM ではどうにもできないすべてのワークロードの決定プロセスに要する経過時間の問題は生じない。

## 1 つのアドレス空間に異種のワークロード

異なるリソース要件のアドレス空間を 1 つのサービス クラスに混在させることに加え、異種のワークロードもアドレス空間に含めることができる。これらのアドレス空間にエンクレーブが含まれていても問題にはならない。エンクレーブは、別々のディスパッチ可能なワークの単位で、WLM により別個に異なるサービス クラスに分類することができる。IBM の DB2 分散スレッドと WebSphere アプリケーション サーバーはエンクレーブをサポートしている。

通常、IMS Message Processing Regions (MPR) と CICS Application Owning Regions (AOR) は、長期実行トランザクションと単純なトランザクションを処理する。これらサブシステムのどちらもエンクレーブはサポートしないが、それぞれ独自に内部ディスパッチング優先順位をサポートしている。IMS がうまく分類されれば、処理される MPR トランザクション クラスの変更が容易になり、より多くの同種のワークロードが可能になる。それぞれの MPR に単純、または長期実行トランザクションしかなく、それらが異なるサービス クラスに分類されている場合、WLM はワークロードをさらにうまく管理できる。ところが、CICS では単純および長期実行トランザクションを同時に処理するため、現実的とは言えない。

トランザクションのレスポンス タイムを測定するゴールの場合、WLM はトランザクションではなく、アドレス空間を管理し、そのアドレス空間がトランザクションを別々に管理しようとする。たとえば、IMS がクラス (1、2、5) を処理する場合、クラス 1 またはクラス 2 のトランザクションが待ち行列に入っていなければ、クラス 5 を処理することになる。WLM がクラス 5 のレスポンス タイムを管理しようとするれば、そのアドレス空間が利用できるリソースを増やすしかなく、そうすると、クラス 1 とクラス 2 を速く処理できるが、必ずしも待ち行列がなくなるわけではなく、クラス 5 が処理されるわけでもない。

内部の優先順位のディスパッチングを使用しなくても、同じことが CICS にも言える。CICS のストレージ管理は長期実行トランザクションよりも単純トランザクションの方に有利に働くため、IMS のクラス 5 のワークロードと同じ処理になる。ところが、CICS 領域は異系のものから同系のものに変更するのが非常に難しく、とりわけ、VSAM を伴う場合は困難だ。そのため、すべての CICS 領域を 1 つのサービス クラスに集中させて問題を悪化させないほうがいい。

## 移行支援としてのベロシティ

---

RMF 互換ベロシティは、制約を受けない環境の優れた移行支援だ。ストレージに制約を受けている場合、WLM は互換ストレージの分離パラメーターと同じようには積極的にメモリを管理しない。CPU で大きく制約を受けている場合、WLM は、それまでに割り当てたのとは別の方法で同じ優先順位のワークロードにリソースを割り当てる。ベロシティによるゴールを誤解しているユーザーは、ディスパッチング優先順位と同じにする可能性がある。普通、ベロシティによるゴールを非常に高く設定すると、これらのワークロードに対して高いパフォーマンス指数 (P/I) が計算されてしまう。そうすると、WLM がこれらのゴールを達成する方法を決定しようとしてサイクルを無駄にすることになる。

ベロシティの場合、WLM はレスポンス目標のように積極的でも効果的でもない。たとえば、ベロシティにより管理される場合、CICS はただのアドレス空間に過ぎず、ストレージによる遅延が発生するまでストレージ保護は受けない。CICS にレスポンス タイムによる目標を設定している場合、ストレージ分離が高く設定され、レスポンス目標を達成すれば下げられる。

ベロシティによる目標のもうひとつのマイナス効果は、IMS 制御領域の VTAM 汎用リソースのロード バランスに見ることができる。ベロシティによる目標の下では、WLM は CEC の利用効率を見る。この率は、たとえ IMS 制御領域が CP による制約を受けているとしても、低くなる可能性がある。しかし、WLM は CP の制約を受けているこの領域にワークを送る。IMS 領域がレスポンス タイムによる目標で管理されている場合、WLM は制御領域の P/I を見て、最も P/I が低い領域にワークを送る。VTAM 汎用リソースの経路指定での同様のマイナス効果は、CICS など、ベロシティ目標を設定した、CP の制約を受けるワークロードすべてに現れる。

## 迷子の羊を探す

---

~どのようにすれば、オンライン ワークロードのサブセットが目標達成に失敗していることが分かるだろうか。

OMEGAMON<sup>®</sup> XE for OS/390 がない場合、目標達成に失敗したことを見つける最も簡単な方法は、アドレス空間順にまとめられたサブシステムのトランザクション レベルの詳細記録を使用する方法だ。OMEGAMON XE for CICS または IMS のようなオンライン モニターは、平均レスポンス タイムが設定目標に達していない場合、リアル タイムのアラートを生成することができる。

サブシステム グループのメンバーがいくつかのアドレス空間に問題があると報告してきた場合、たとえば、サービス クラスのパフォーマンス指数が 8 であれば、z/OS には問題がないと断言することができる。OMEGAMON XE for OS/390 または、IBM の RMF Monitor III のアドレス空間の遅延統計 (障害分析) を使用して、それをサービス クラスの遅延と比較することができる。このアドレス空間の CPU と他の遅延はサービス クラスと比べてどうか。多くの場合、サービス クラスは CPU で 10% の遅延が発生するが、選択したアドレス空間ではさらに遅延する可能性がある。アドレス空間の遅延とサービス クラスの遅延を同じものと考えてはならない。

## アプリケーションのインフラストラクチャーに投資して、 順調に利益を上げた National Australia Group



### グローバルな金融機関がキャンドルの開発ツールとサービスを活用して顧客サービスを向上

いつでもどこでも預金情報にアクセスしたいという顧客の要望を受け、1970年代にATMが登場してからというもの、銀行業界は次々とデリバリチャンネルを追加してきた。今日では銀行が、民間窓口やコンビニ、ATM、インターネット、コールセンター、さらにはワイヤレス機器などといった、6つ以上のチャンネルを介してサービスを提供するのが当然のことのようになりつつある。

1990年代半ばまでに多くの銀行が、急速に、時には場あたりのIT発展による影響を受け始めた。つまり、銀行業務が非常に閉鎖的で統合されていないバックオフィスのレガシーシステムに縛られており、しかも顧客データが往々にして相互に連結されていない複数のシステムに格納されることが分かったのだ。その結果として、金融サービス機関は顧客サービスの向上と、多種の追加サービスや、より高価な商品の販売へと結びつくような大局的な展望を持つことができなかった。

米国、ヨーロッパ、アジア、オーストラリア、および太平洋諸島で事業を展開する国際金融サービス機関、National Australia Groupはまさにこの問題に直面した。急速に拡大するサービスデリバリチャンネルに対応するため、National Australia Groupは、アプリケーションのインフラ戦略を緊急に見直す必要に迫られた。

最初の難題は、新しいフロントエンドのアプリケーションを、口座と顧客データが格納されている既存のバックエンドシステムと一体化するという点だった。この工程を進めていく中、時間やコストおよび人材確保の点において、IT開発とサポートにはますます負担がかかっていった。「世界各地の営業活動情報の大量複製に加え、膨大な量のハードコーディングが必要な状況だったのです」とNational Australia Groupの開発ラインマネージャー、Colin Hume氏は語る。「そこで、当社では大量のコーディングを行うことなく、フロントエンドとバックエンドを分離し、世界中の営業拠点に容易に複製できる新しい統合方法を探すことにしました」

### Candle のツールとサービスでチャンネル展開を加速化

National Australia Groupは、今ではPathWAI™ Developer for WebSphere Business Integrationソリューションの不可欠な要素となっているキャンドル社の技術とコンサルティングサービスを選択し、増大するアプリケーションインフラの統合化の難題に取り組むことにした。PathWAIソリューション製品群を使用することによって、組織では自社のアプリケーションインフラ投資を最大限に活かすことができ、クリティカルで複

雑な企業環境にはよくあるパフォーマンス問題を克服して、投資収益率を上げることができる。

同社では、キャンドルの専門家によるコンサルティング サービス、業界のベスト プラクティスおよびそのクラスで最高のツールを活用して、最新のアプリケーションのインフラを実現した。このプロセスには、フロントエンドのコンポーネントとミドルウェア層を一体化するためのコネクタの作成も含まれていた。ツールとサービスは、Web ベースのアプリケーションの開発、管理に要する時間、労力および費用を削減するように設計されている。こうしたツールとサービスを使用することにより、National Australia Group では内部の人材を補強し、プロジェクト完了までの期間を短縮し、開発費とミドルウェア層導入の全体的なリスクを下げる事ができたのである。

キャンドルのコンサルティング サービスが、National Australia Group がミドルウェア プロジェクトの開発段階で発生したいくつかの問題を解決するのを支援してくれたと Hume 氏は述べている「新しいアプリケーションとサーバーのハードウェアをよく知らない、などの理由で、当社では開発プロセス完了までの期間を実際よりも短く見積もっていました。キャンドルからは、専門家によるコンサルティングとベスト プラクティスで、常に適切な指導を受けました。キャンドルのコンサルタントは、世界中で何千にもおよび業務統合を実現し、ミドルウェアを結合しています。その企業全体の経験を結集して我々のために協力してくれました」

## 複雑な状況

---

National Australia Group のイギリス支店は、業務がとりわけ困難な IT 要件に直面した。イギリスのチャネル統合プロジェクトには、以前は独立銀行であった 4 行が含まれ、それぞれが異なる 3 台のメインフレームを共有してバックオフィスのアプリケーションを稼働させていた。世界各地にある National Australia Group の拠点では、専用のメインフレーム環境を保有している。

また、この統合化問題をより一層複雑にしたものは、既存のフロントエンドの PC だった。PC の多くは最新のクライアント ソフトウェアを実行することができず、時代遅れのものになっていた。よって、新しいハードウェアが統合プロジェクト項目に盛り込まれたのである。

National Australia Group のイギリス支店では、新しいミドルウェア戦略を展開する前に電話による銀行取引を始めていた。その結果、IT チームはカスタマイズしたソフトウェアをプログラミングしながら、残高照会や支払いなど、ルーチン トランザクションを実行するのに必要なメッセージ フローと変換をサポートしていた。このシステムでは、トランザクションが共有の中間の孤立した層を介さずに、直接バックエンド システムに送られていた。こうした方法により、バックエンド システムに格納された重要なデータへのチャネル間アクセスが制限されたのである。

2001 年初頭、キャンドルは National Australia Group のイギリス支店と協働し、より整合性の取れた新しいチャネルの実装方法、即ち適応性があり、ハードコーディングをほとんど必要とせず、実装費用のかからない、しかも開発とサポートの重複を回避できる方法を開発した。National Australia Group では、まずこの方法を使用して、電話による銀行取引をイギリスの営業拠点のバックエンド システムにリンクさせた。

キャンドルのツールとサービスは、National Australia Group の IT 環境が複雑だったということもあり、中でもイギリス支店には有益なものとなった。キャンドルのミドルウェア ツールは、アプリケーションの外部のメッセージ変換が管理できるため、同グループではイギリス内の複数の銀行のフロントエンドを 1 回のキー操作でアップグレードすることができ、時間とコストが節減できた。「たった 1 組のソフトウェアでアップグレードを完了することができたのです」と Hume 氏は語る。「我がグループ全体が、フロントエンド システムを更新するのに 4~5 年に 200 百万ポンドを支出することを想定すれば、世界的規模で見たこの能力は莫大なものと言えるでしょう」

キャンドルのツールとサービスは、同社がオンライン バンキングを導入した際にもその価値を発揮した。「当社ではすでにレガシー システムを撤去していたので、新しいシステムを直接ミドルウェア層に繋ぐことができ、コーディングの必要もありませんでした」と Hume 氏は述べている。「当社では同じトランザクション セットを使用して、オンライン バンキングを直接ミドルウェア層に組み入れたため、すぐさま恩恵を受けることができました」

## 賢明な投資

---

多くの場合、問題が発生するとミドルウェアが都合よく槍玉に上げられるが、それはアプリケーションの担当者とヘルプ デスク要員の多くがミドルウェアと直接関係を持たないからである。ところが National Australia Group の場合は違っていた。「ミドルウェア層をインストールして以来、当社ではミドルウェア環境が原因で問題が発生したことは 1 度もありません。ほとんどの場合、問題はフロントエンドで発生し、アプリケーション パッケージが正しいメッセージを送信しないことに起因しています」と Hume 氏は語る。「こうした結果は、まさにキャンドルの専門技術の証しです。キャンドルのおかげで当社ではミドルウェア環境を開発し展開することができました」

発行:2003 年 11 月

このケーススタディは、キャンドル社の製品を実際に使用している顧客を紹介するものです。実際のシステム、使用状況、およびニーズは顧客によって異なるため、キャンドルはこれと同様の結果を保証いたしません。キャンドルは、顧客および キャンドルのビジネスパートナーから提供された情報の正確さについては、保証いたしません。キャンドルと取引する一般のお客様がここで言及されている製品を利用できない場合でも、キャンドルは当該製品を開発する義務を負いません。また、当該製品を市販したり、将来市販するいずれかの製品に当該製品のいずれかの機能・性能を組み込む義務も負いません。この文書に記載されたその他の製品名、語句などは各社の商標であることがあります。