CICS−UDB高性能基幹OLTPシステム構築事例

CICS® TXシリーズをトランザクション・マネジャーとして、DB2 UDB™ for AIX®をデータベース・マネジャーとする、物理3層のサーバーに複数台（10ノード以上）のRS/6000® SP™とS80モデルを構成した、基幹OLTP（オンライン・トランザクション処理）システムの開発が行われ、成功裏にサービス・インしました。
本論文では、高可用性と高セキュリティーの要件にも触れながら、特に高性能要件の達成に焦点を当て、これらの要件を実現するために行った基盤設計において、どうして物理3層のシステム構成に至ったのか、その検討過程を説明していきます。さらには、決定したシステム構成の性能測定した結果もご紹介します。

日本アイ・ビー・エム株式会社 テクニカル・サポート
IBMディスティングイッシュト・エンジニア
ICPシニア・コンサルティングITスペシャリスト
ICP Senior Consulting IT Specialist
IBM Distinguished Engineer
Technical Support, IBM Japan, Ltd.

室住 正晴 Masaharu Murozumi

［プロフィール］
1976年、日本アイ・ビー・エム入社。本社SE技術部門にて主にDB2などのデータベース製品の技術支援を行う。現在、データベースを専門分野とするITスペシャリストとしてさまざまな業種のお客様で、高性能かつ高可用性が求められる基幹業務および情報系システムにおいて、メインフレームからUNIX、PCにわたるオープンかつ先進的なクライアント/サーバー・システムに構築する開発を、長年のデータベース基盤技術の知識・経験を生かしてご支援している。

Examples of the Construction of CICS-UDB High-Performance Mainstay OLTP Systems

A mainstay OLTP (On-Line Transaction Processing) system consisting of a physical three-layer server with the CICS® TX series as the transaction manager and DB2 UDB™ for AIX® as the database manager and with several RS/6000® SP™ units (10 nodes or more) and S80 models has been developed and has now successfully gone into service.
In this paper I focus especially on the achievement of high-performance conditions while alluding to the conditions for high usability and high security. I explain the study process as regards why we arrived at a physical three-layer system configuration in the basic design conducted to realize these conditions. I also present the results of measurement of the performance of the system configuration that was decided on.

1. はじめに

本論文で述べる基幹OLTP（Online Transaction Processing：オンライン・トランザクション処理）システムの大量トランザクションを処理するセンター・システムに、高性能で費用対効果の高いRS/6000®のSP™とS80モデルによるクライアント/サーバー・システムを導入しました。システムを構成する上で、ハードウェアで可能な個所は徹底して2重化・冗長化が図られたので、高い可用性を実現することができましたが、UDB（DB2 Universal Database™）をデータベース・サーバーとする基幹オンライン・システムで、500TPS（Transaction Per Second）に達する大量のトランザクションを処理している事例は、これまで国内には見当たりませんでした。本論文では、高可用性を維持しつつ、大量トランザクション処理を可能にした高性能OLTPシステムの基盤設計を、どのような観点から検討したのかを述べていきます。

2. 基本的なシステム要件と課題

2.1. オンライン・トランザクションの概要

センター・システムが提供する端末は、当センター・システムと契約した営業店の店頭に置かれています。サービス時間帯は常時、センター・システムと公衆電話回線で接続されているので、NTTのINS-PまたはDDX-Pパケット・サービスを利用することができます。来店したお客様は、注文したい商品をマーク・シートに記入します。店の販売員が、そのマーク・シートを端末に読み取らせると、端末は受注処理要求の電文に固有の符号化を行ってセンター・システムに送信します。センター・システムは電文を復号化した後、受注情報をUDBデータベースの受注表に挿入します。お客様が会員の場合は、会員表のその人の購買累積額を更新します。その処理結果は、端末に再度、符号化された応答電文として返信されます。端末は復号後、受注確認伝票（後日、商品受け取りの控えに使用する）を発行することによって、1個のトランザクション処理を完了させます（図1）。

2.2. オンライン・トランザクションに求められた基本的な要件

当システムのトランザクションの特徴と、センター・システム側の処理機能に与えられた課題は、(1)「低い通信費用」、(2)「高いセキュリティー」、(3)「高い可用性（耐故障性）」、(4)「高い信頼度」、(5)「高い性能と安定したピーク時端末応答時間」です。

2.3. 電文到達保証のためのトランザクション設計

上記(1)の低い通信費用のトランザクションを実現するために、端末側プログラムとセンター・システム側プログラムとの間の通信プロトコルは、送受信確認の合計4パケットが最少限必要なTCP/IP（ Transmission Control Protocol / Internet Protocol）ではなくて、送受信が最少2パケットで済むUDP（User Datagram Protocol）を採用しました。UDPは、TCP/IPとは異なり、受信側が重複電文を受け取る可能性があるので、電文到達の保証は、ユーザー・プログラムの責任になります。送信電文の未着信は、決められた時間内に、返りの電文が送られてこないことによって検知することができます。しかし、重複電文でないかどうかの検査は別途、ユーザーによる仕組みが必要です。そうしなければ、受注確認伝票を2重発行してしまうことがあるからです。そこで、重複電文が発生した場合、正しくトランザクションを処理する解決策として、次のような方法を採用しました。

(1) UDBのデータベースに電文ID表を用意し、一意な電文IDを索引キーとするユニーク索引を定義します。
(2) 端末は電文ごとに一意な電文IDと、受注ごとに一意な受注No.を採番して電文を送信します（電文IDと受注No.は同じではありません）。
(3) 受注トランザクションは、最初に電文ID表に挿入（SQL INSERT文）を試みます。もし、電文IDが重複キー違反ならば、重複電文と判断して、その電文を破棄します。挿入が成功すれば、コミットします。
(4) 端末が、採番した受注No.をキーにして、受注情報を受注表に挿入（SQL INSERT文）します。さらに、店別売上表の更新と、必要であれば会員表の更新を行ってコミットします。

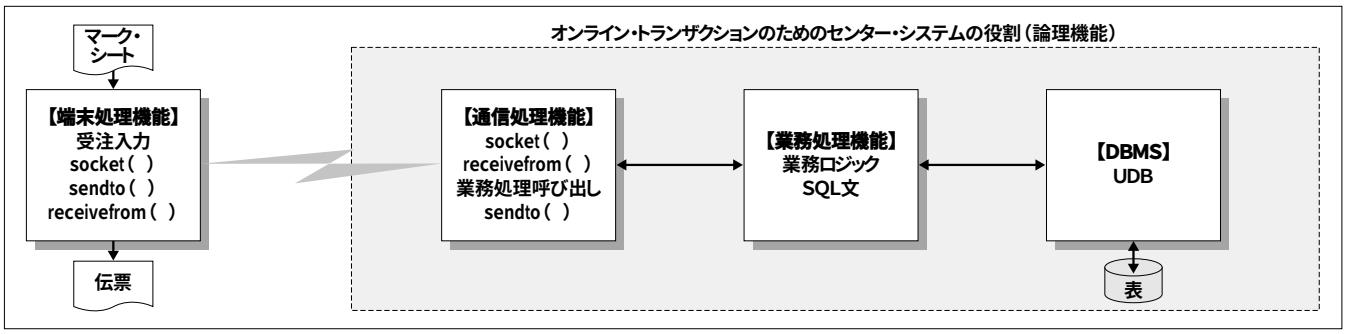


図1. オンライン受注トランザクションの処理機能とセンター・システムの役割

(5)もし、アプリケーション・プログラムが(4)の処理中に異常を検知したならば、端末にその旨を返り電文で知らせます。端末は、電文IDを新規採番するが受注No.は変えずに、受注要求電文を再送信するので受注を再実行できます。

(6)問題は、(4)のコミット後に成功を知らせる返り電文が端末に届かなかった場合です。端末は、受信待ちタイムアウトで異常を検知し、電文IDを新規採番しますが、受注No.を変えずに受注要求電文を再送信します。この場合、(4)の受注表処理で受注No.が重複違反となり、重複受注を防ぐことができます。

通信プロトコルにUDPを使用してもデータ整合性上の大きな問題はなくなりました、しかし上記のトランザクション設計では、1件当たり2回のコミット文を発行するので、目標500TPSの場合、毎秒1,000回のコミット処理が必要になります。コミット処理は、UDBの活動ログに強制物理I/O(Input/Output:入出力)を発生させるため、ログI/Oがボトルネックになるのではないかと懸念されました。この問題は、6.3節で考察します。

3. 通信サーバーのノード構成検討

システム基盤設計として、まず、通信処理機能を行うサーバーのノード(または物理ノード)構成を検討します。このノードとは、RS/6000のSPフレームの中の1台のプロセッサー、あるいは1台のモデルS80プロセッサーを意味します。店端末と通信サーバーとの間には、可用性と負荷分散を目的として実績のあるネットワーク・ディスパッチャー(以下、ND)を配置しました。通信サーバーのノード構成案を図2に示します。ここで問題になるのは、通信サーバーをHACMP(High Availability Cluster Multi-processing)構成にするかどうかです。図中のNDと各ノードとの間の数字は、負荷分散して振り分けるトランザクション量の平均の割合を示しています。その値を例に検討を進めます。

A:ND+HACMP構成

SPが4ノード与えられているとき、そのうち1ノードを待機ノードとすることで、そのほかの3ノードのいずれが障害を起こしても、処理は継続できます。障害発生時から待機ノードに引き継ぐまでの時間はNDの働きにより、電文が正常ノードに自動的に迂回(うかい)されるので、端末側は障害を意識することなく、負荷分散かつ耐故障性により連続稼働を実現できます。

B:ND+[N-1]構成

SPが4ノードを与えられていれば、正常稼働時、全ノードを使用できます。もし、いずれかのノードに障害が発生した場合は、そのノードの修復を待ちますが、NDは障害ノードを自動的に検知して正常ノードに電文を迂回(うかい)するので、負荷分散と耐故障性により連続稼働が実現します。通信サーバーの各ノードは、データベースのような回復が必要になる資源を持たないように設計すれば、HACMP構成でなくても、NDによる電文振り分けだけで、連続稼働が可能です。つまり、NDのおかげで、正常時にノードを待機させておく意味が薄れたのです。正常稼働時は、与えられた資源である全ノードの処理能力をフルに利用できるので、通信サーバーはHACMP構成をせず、1個のノードが障害のときは、N-1個のノードによる縮退運転を行います。これは今日、Webサーバーの構築によく見られるノード構成と同じです。基本的な違いは、通信プロトコルがHTTP(Hypertext Transfer Protocol)でなくUDPなので通信処理機能を独自に開発しなければならない点です。

なお、ノード数は、ベンチマーク測定によって得られたトランザクション1件当たりに必要なCPU(Central Processing Unit:中央演算処理装置)負荷と、どの程度までの N-M縮退運転を想定するかに基づいて、最終的に決めました。この後に述べるアプリケーション・サーバーについても同じ考え方でノード数を決めました。

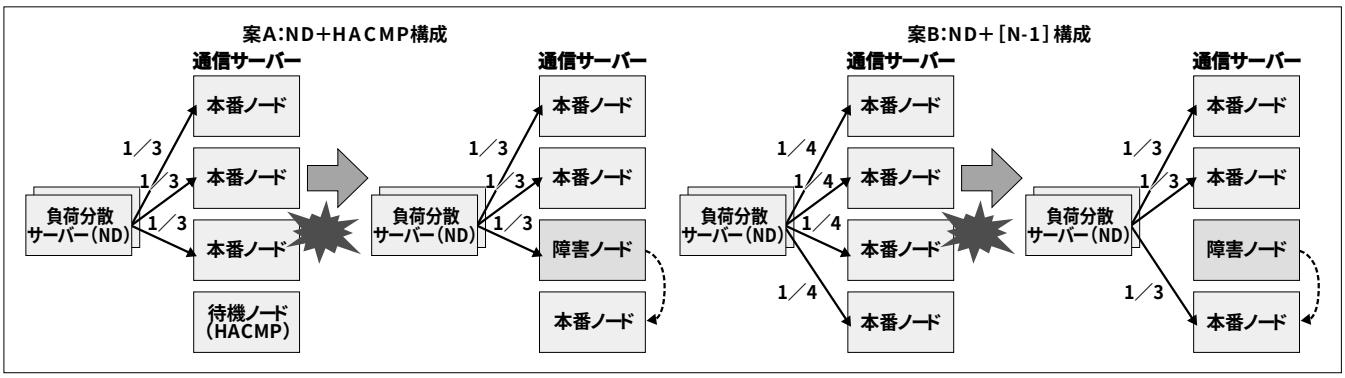


図2. 通信サーバーのノード構成案

4. トランザクション・マネジャーの選択とデータベース・アクセス方式

トランザクションのたびにSQL CONNECT文を発行してしまうと、オンライン・トランザクションを大量に処理する際に、性能上のボトルネックになってしまいます。実際、CONNECT文のCPU負荷は、オンライン・トランザクションでよく使われる静的SELECT文のそれに比べて52倍も重かったのです。そこで、CONNECT文の負荷がボトルネックにならないための設計と、トランザクション・マネジャー利用の案を検討しました。

また、データベース・アクセスに関しては、静的SQL(Structured Query Language：構造化照会言語)、CLI(Command Line Interface)、JDBC(Java™ Database Conectivity)やSQLJなどいろいろな方法が可能ですが[参考文献1]、高性能要件から、できるだけCPU負荷の小さいアクセス方式を採用しました。

(A)AIX®常駐プロセス－静的SQL

通信サーバーにDB2® CAE(Client Application Enabler)を導入しました。UDP送受信プログラムであるAIX常駐プロセスがUDBに接続したまま、受注要求電文を受けるたびにアプリケーションとして静的SQL文を発行します。

(B)CICS® TX-静的SQL

通信サーバーのAIXプロセスがCICS TXSeries®の下で稼働するアプリケーションをECI(Extend Call Interface)によって呼び出し、呼び出されたCICSアプリケーションが静的SQL文を発行します。

(C)WAS-JDBC

通信サーバーのUDP送受信プログラムとIIOP(Internet Inter-ORB Protocol)で通信するWAS(WebSphere® Application Server)をアプリケーション・サーバーで稼働させ、WASのEJB(セッション・ビーンまたはエンティティー・ビーン)がJDBC経由でUDBにアクセスします。

今回の高性能OLTP開発では、Java言語による利点を積極的に得られる状況にないと判断しました。そこで、数多くの実績と安定した技術を重視して、本プロジェクトでは(B)のように、トランザクション・サーバーとしてCICS TXを、開発言語としてC言語を選択しました。

5. アプリケーション・サーバーのノード構成検討

ここでは本来の受注処理という業務ロジックを実行するアプリケーション・プログラムを、どこで稼働するのかについて検討します。

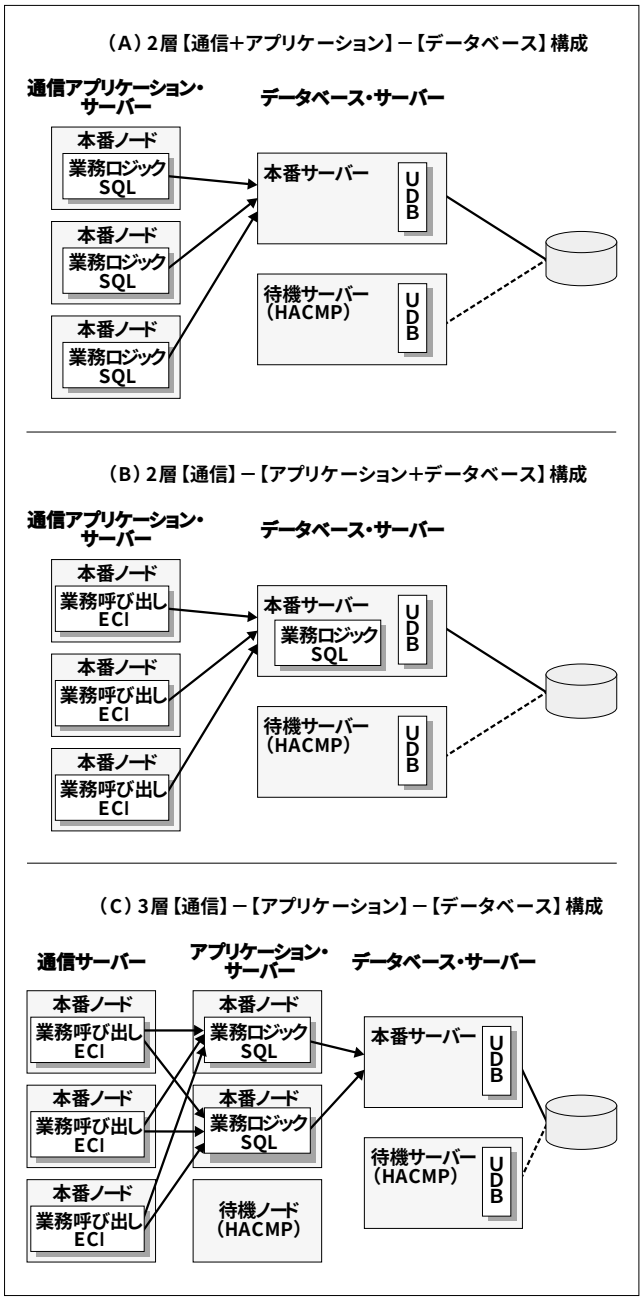


図3. アプリケーション・プログラム(業務ロジック)の実行場所検討のためのノード構成案

アプリケーション・サーバーを独立させるか否か、センター・システムのサーバーを2層にするか3層にするかの分かれ目です(図3)。

図3の(B)は、可用性(耐故障性)の観点では、アプリケーション・プログラムやトランザクション・マネジャーの不具合に影響されやすいという弱点があります。性能の観点からは、図3の(A)が望ましいと考えられました。しかし、開発体制が通信処理とアプリケーション処理とデータベース・サーバーのそれぞれに主管チームが分かれたので、開発テストと問題判別が最も容易な(C)が、採用されました。ただし、図3の(C)は、通過するノード数が多く性能は不利なので、性能測定によって許容できるかどうかを検証することにしました。その結果は、8章で述べていきます。

アプリケーション・サーバーでは、CICSが稼働します。しかし、CICS資源とUDB資源との間で2相コミットが必要になるような設計をしないことを前提に、複数ノードを与えました。よって、ノード障害時に単純なN-1縮退運転が可能です。

6. データベース・サーバーのノード構成検討

6.1. UDB EEE版を応用する妥当性

最新のRS/6000のS80モデルなどのマルチプロセッサー・モデルは信頼性が高いだけでなく、障害が発生しても障害CPUを切り離して処理を継続できるので、高可用性に優れています。しかし、RS/6000の全体障害の可能性は、完全にゼロではありません。そこでHACMP構成によって、さらに耐故障性を高めました。同時に、連続稼働を可能にする構成の妥当性を追加検討しました(表1)。

(A)UDB EEE版+HACMP構成

障害時に受注オンライン業務を部分的にも継続が可能な構成です。

- RS/6000 SPのノードが4個ある場合、3個のノードにUDB EEE(Extended Enterprise Edition)版を構成し、1個のノードを、いずれのノードが障害を起こしても引き継げるHACMPの待機ノードとします。
- トランザクションを1個のノードに閉じて処理可能となるように、表を営業店別に物理分割します。分割した表をそれぞれのノードに専用の表として配置します。例えば、A店の売り上げ情報はノード1の受注表に、B店のそれはノード2の受注表にそれぞれ記録します。
- 会員表は、全国に共通なのでノードをまたがる1個の表として作成します(EEE版では物理ノード間で論理的に1個の表として扱えます)。
- アプリケーション・サーバーのプログラムは、店コードを判別してトランザクションを送るべきデータベース・サーバーのノードを選択します。
- 例えば、A店に来られた非会員のお客様の受注は、データベース・サーバーのノード1に送ります。トランザクション処理を、ノード1だけで1相コミットで完結することができます。
- しかし、当受注サービスは、お客様がどの店に行ってもできるようにする必要があります。会員がB店に行った場合は、ノード2の受注表とノード間をまたがっている会員表を更新アクセスしなければならず、2相コミットが必要になります。

(B)UDB EE版+HACMP構成

「RS/6000高可用性パッケージ」が提供する構成と同じく、標準的な構成です。

- RS/6000 S80の2台をHACMP構成として、1台を待機ノード(待機サーバー)とします。
- UDB EE(Enterprise Edition)版により1個のデータベースを構築します。
- 本番ノードに障害が発生した場合、待機ノードに引き継がれるまで全体停止となります。

データベース・サーバーを複数のノードに分割することは、非会員の受注トランザクション処理であれば、負荷が分散されるので、性能の観点からは好都合のように見えます。しかし、店別受注表

表1. データベース・サーバーのノード構成案比較

比較項目	(A)		(B)	
	UDB EEE+HACMP構成	評価	UDB EE+HACMP構成	評価
可用性	•非会員による受注は全体停止が避けられる。 •カタログ・ノード障害は全体停止になる。 •いずれかのノードに障害が発生すると、会員からの受注が停止する。	◎ × ×	•全体停止になる。	×
性　能	•表容量を物理的に均等に分割できたとしてもトランザクション量は、各ノードに均等に負荷分散することができない。 •ノードをまたがるトランザクションは案Bに比べてCPU負荷が大きい。 •ログへの物理I/Oが分散される。 •ノード当たりのデータベース・バックアップ・回復時間は短い。	× △ ○ ○	•特に、考慮は不要。 •単一データベース・サーバーなので案Aに比べてCPU負荷が少ない。 •ログへの物理I/Oが集中してボトルネックになる可能性あり。 •全体バックアップ・回復時間が長い。	○ ○ △ ×
運　用	•表容量を均等に分割して維持することが難しい。 •表容量を均等に分割できたとしてもトランザクション量は、各ノードに均等に負荷分散することができない。 •複数ノード運用は単純でない。	× × ×	•考慮点は少ない。	○
開　発 生産性	•複数ノード管理の運用手順が必要。 •2相コミット・トランザクションがあるので障害テスト・ケースが多い。	× ×	•運用手順テスト・ケースは少ない。 •1相コミット・トランザクションなので障害テスト・ケースが少ない。	○
平均サービス停止時間	4.4分＝22分×会員比率(当初20%)＋0分×非会員比率(80%)		22分	△

を均等にノードに分割することと、流れるトランザクション量は別物です。よく売れる店とそうでない店との間で、ばらつくことは十分にあります。従って、実際のところ表をノードごとに、どのような基準で分割するのかを決めるのは、非常に難しいことが分かりました。

6.2. HACMP構成による、データベース・サーバーのシステム障害時の引き継ぎ時間

表1の(B)を採用する場合、全体停止は避けられません。そこで、業務の停止時間はどれくらいなのか、停止時間を業務として許容できるかどうかの評価を行いました。ここでは、HACMP構成での引き継ぎ時間を基に検討を行いました。

・データベース・サーバーの引き継ぎ時間の測定環境

- ハードウェア RS/6000 S7A ×2台(64bit−PowerPC 262MHz)。
- IPアドレス数＝3個(100BASE Ethernet+SP Switch)。
- ディスク・パーテーション数＝2個、合計20Gバイト(10個のファイル・システム)
- DB2バッファー・プール＝1Gバイト、DB2構成パラメーターSOFTMAX＝1、I/O CLEANERS＝1。
- DB2のバッチ・プログラムは、5回の単一行UPDATE文と1回の単一行INSERT文ごとに、COMMIT文を発行します。これを、1件のトランザクションと見なします。このプログラムは、1Gバイト以上の大きな表の全体を更新します。

・観測結果

- SPスイッチの障害検知のために30秒〜1分程度必要。そのほかの検知に30秒、合計90秒。
- ディスクの引き継ぎ(ファイル・システム・チェックおよびマウント処理) 10Gバイト／分。
- IPの引き継ぎ処理は、1アドレス当たり25秒程度。
- トランザクション量が85〜194TPSのときにCPU障害が発生したあと、待機サーバーでのDB2の始動処理(再接続を含む)には103〜199秒を要し、障害時のトランザクション量が多いほど長くなりました。なお、DB2構成パラメーターのSOFTMAX値は小さいほど、I/O CLEANERS値は大きいほど始動時間を短縮する効果があることが分かりました。上記の観測結果を当システムのデータベース・サーバーに適用すると、約22分と見積もられました。

DB2始動時間は、システム障害時に残っていたディスクに未書き込みのデータページ数と仕掛かり中だったトランザクション数に比例すると想定できます。そして、当システムのDB2バッファー・プールを測定環境の2倍の大きさである2Gバイトと仮定し、目標トランザクション量を500TPSとして計算しました。また、表1の(A)のように、UDB EEE版を利用して、部分稼働が可能になるように工夫したとしても、カタログ・ノードに障害が発生した場合は、全体停止になります。また、たとえカタログ・ノードが正常に稼働している場合でも、いずれかのノードに障害が発生すると、会員表がアクセスできなくなり、会員からの受注処理は停止されます。また、表1の(A)は平均サービス停止時間が4.4分と短いように見えますが、会員比率を上げたいという将来的な戦略に反して、会員比率が上がるほど、サービス・レベルが低下してしまいます(障害時、店側は会員の受注入力を待たせて、非会員の受注入力を優先せざるを得ない)。結局、全体停止を許容するということで、表1の(B)を採用することにしました。

6.3. コミット性能―活動ログのボトルネックの可能性

目標500TPSのトランザクション量のために、毎秒1,000回のコミット処理が必要になりました。実際、CICS-UDBにとっては、1,000TPSの内部トランザクション量に相当します。ユーザーの表や索引の部分は、複数のDASDに分散して配置すれば、物理I/O競合を意図的に減らすことができます。しかし、活動ログは常に1個所に集中します。メインフレームの基幹OLTPの開発経験によると、ログI/O性能は、スループットに影響します。実際には、DB2 for OS/390®版が提供するシスプレックス環境におけるデータ共用機能は、複数のDB2を稼働させることによって、各DB2の活動ログに物理I/Oが分散されるので、ログへのI/Oがボトルネックにならないように設計されています。では、データベース・サーバーが1個の当環境では、活動ログがボトルネックにならないのでしょうか。

そこで、RS/6000モデルS80をデータベース・サーバーとして、ESS(Enterprise Storage Server)に、ユーザー表と活動ログを配置した場合のコミット処理性能を測定しました。検証用バッチ・プログラムは、表の2列を更新して、約2Kバイトのログを出力するUPDATE文を、1回実行するごとにCOMMIT文を発行します(コミット1回分をトランザクションと見なします)。そして、このバッチ・プログラムを5〜50多重で実行させ(もちろんロック待ちにならないように異なる行を更新する)、かつ最大20Gバイトのログ量を出力させてキャッシュから確実にディスクに書き込ませるように考慮しました。S80とESSの組み合わせのグラフは、1,000TPSを超えてもCPU負荷は許容できる程度であり、また、CPU使用率が線形に増加しているので、活動ログへの物理I/Oがボトルネックになっていないことを示しました。その結果、データベース・サーバーが1個でも十分に目標のトランザクション量を処理できると判断しました。本番環境のESSはAIXミラーリングされましたが、PARALELL指定によってミラーリングによる性能低下の影響はありませんでした。それでは、これまでシステム構成を検討した結果を図4に示し、次章で説明していきます。

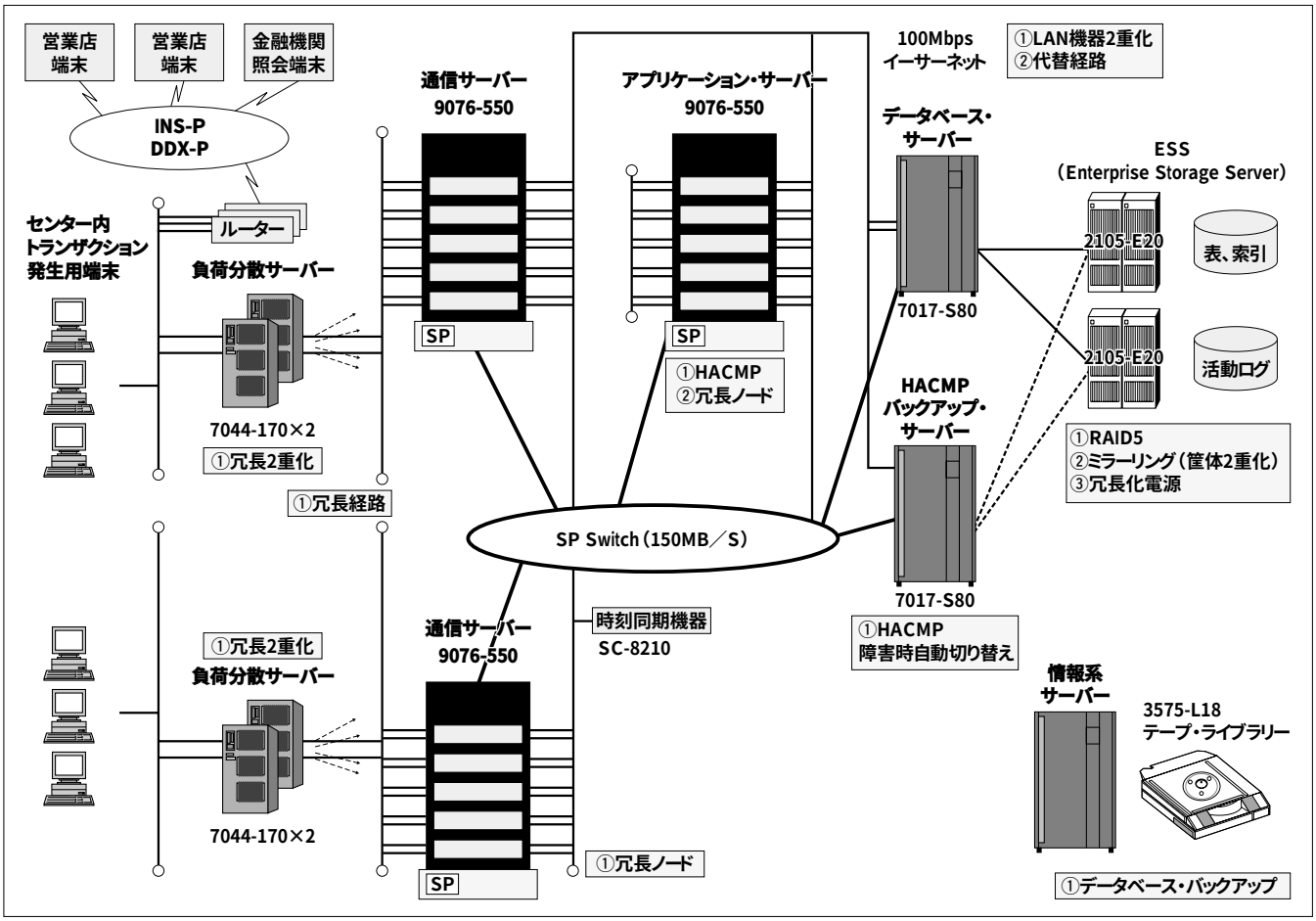


図4. センター・システムの最終構成の概要

7. センター・システムの最終構成

7.1. 各サーバーの役割

• 負荷分散サーバー

― 端末からの受注要求電文(パケット)を、複数ノードの通信サーバーに振り分けて、通信サーバーへの負荷を均等に分散させます。…負荷分散

― 負荷分散サーバーは、送信先サーバーから応答がない(ノード障害)、応答が遅い(高負荷)などの場合には、別のノードへ電文を送信します。…耐故障性

• 通信サーバー

― 負荷分散サーバーから電文を、まずはキューに保管した後、アプリケーション・サーバーの複数のノードのうち、送信先をラウンド・ロビンで選択します。…負荷分散

― アプリケーション・サーバーのノードが停止している場合には、そのノードへの送信を止めて、次の正常なノードに振り分けます。…耐故障性

― トランザクションの種類を解析し、CICS ECIサービスを利用して、適切なアプリケーション・サーバーのCICSアプリケーションを呼び出します。

• アプリケーション・サーバー

― CICS TXSeriesが稼働します。

― 通信サーバーからのトランザクション処理要求を実行するサーバーで、CICSアプリケーション・プログラムがデータベース・アクセス要求をデータベース・サーバーに行います。トランザクションの同期点処理(1相コミット)は、ここで行います。

• データベース・サーバー

― 受注情報をUDBのデータベースに格納し、アプリケーション・サーバーから要求されたデータベース・アクセス要求を処理します。

7.2. 高可用性のための対策方法

(1)ネットワークの2重化

LAN(Local Area Network)機器の2重化や、SPスイッチの代替経路として100MbpsイーサネットLANを用意して、障害時にバックアップLANへの自動切り替えを行います。

(2)ディスク構成におけるデータ整合性の維持

• 各サーバーの内蔵ディスクは、すべてAIXによるミラーリング構成を採用。

• UDBのデータベースおよびログ、データベース・バックアッ

プやログ・アーカイブのように、万が一の損失も許されない重要なデータはすべて、RAID5構成を行った信頼度の高いIBM ESSディスク・システムのディスク中に格納します。ESSは、2重化クラスター制御機構、SSAデバイス・インターフェース、冗長化電源系統、冗長化冷却機構により高可用性を実現しています。

- ESS単体の高可用性に加えて、当システムでは、それぞれ異なるESS筐体でディスク・ペアのAIXの機能によるミラーリングを施しました。万が一の筐体障害からもデータは保護されています。

7.3. 非定型照会に対するデータ機密保護の対策

これまでに述べてきたアプリケーションは、受注という定型的な処理でした。ところが、受注売り上げ実績などを分析する情報系オンライン照会業務がありました。さらには短期間に開発することが要求されたために、Visual Basic（以下、VB）を利用して開発することにもなりました。そこには次のような問題がありました。

（1）さまざまな照会の条件があり、それらをすべて固定のSQL文であらかじめプログラム中に用意するのが望ましいのですが、その数が幾何級数的に多くなる可能性がありました。そこで、実行時に動的にSELECT文を組み立てる方法が求められました。

（2）VBであるため、UDBにはODBC（Open Database Connectivity）経由でアクセスします。本番のVBを実行するPC（Personal Computer）端末は、物理的に隔離され、かつ厳しい運用管理の中で保護されています。ランタイム・モジュールは導入されていますが、通常、決められたプログラム以外に実行することはできません。しかし、悪意のある人が厳しい管理をかいくぐれば秘密裏にVBの開発環境を導入することは技術的に可能でしょうし、発行するSQL文を書き換えることが可能になります。また、このような事態が将来起きないとも限りません。

そこで、VBアプリケーションが直接、SQL文を発行するのではなく、あらかじめサーバーのUDBに格納されている「定型」ストアード・プロシージャーを呼び出すことにしました。ストアード・プロシージャーには、SQL SELECT文の選択リスト句からFROM句およびWHERE句の先頭までが、動的SQL文の初期値として「半完成品」のSQL文が埋め込まれています。VBアプリケーションからストアード・プロシージャーに渡せるパラメーターは、WHERE句の後部に連結したい条件（述語）だけであって、動的SQL文ではあっても、決められた表以外のアクセスや、SELECT文をUPDATE文などに書き換えたりすることはできません。

この結果、端末からの非定型なさまざまな条件の照会を可能にするだけでなく、アプリケーションの開発生産性を上げつつ、悪意のあるアクセスを防ぐ機密保護の両方の要件を両立させることができました。

表2. 受注オンライン・トランザクションがアクセスする表の特徴

表	行長（バイト）	列数	索引数	表容量（Gバイト）
表A	81	10	1	0.35
表B	206	24	5	4.1
表C	115	16	3	0.03

8. オンライン・トランザクション性能の検証

8.1. 表とトランザクションの特徴

前章までに、決定したシステム構成の上で本当に期待するトランザクション性能が得られるかどうかを検証しました。本章では、その結果を紹介します。受注トランザクションが、アクセスする表は、次の3種類です（表2）。

ここでは、CICSの同期点が2回あるために、実際は、受注1件当たり2個のCICS-UDBトランザクションから構成されるということに注意してください。2章で、UDPの重複電文を検査する仕組みにUDBのユニーク索引を利用したことを述べましたが、1回目のCICSの同期点までのINSERT文がそれに当たり、後半のINSERT文とUPDATE文が、本来の受注業務に必要な表の更新です（表3、図5）。

8.2. 測定環境の概要と仕様

（1）通信サーバー

- RS/6000 SP（POWER3™ 375MHz、4way、実記憶域 4Gバイト）
- AIX 4.3.3、 CICSクライアント 3.1.1

（2）アプリケーション・サーバー

- RS/6000 SP（POWER3 375MHz、4way、実記憶域 4Gバイト）
- AIX 4.3.3、CICS TXSeries サーバー4.3、DB2 UDB CAE 6.1

（3）データベース・サーバー

- RS/6000 S80（RS64III 450MHz、12way、実記憶域 4Gバイト）
- AIX 4.3.3、DB2 UDB 6.1 FixPak 6

表3. トランザクションが発行するSQL文とCICSコマンド

SQL INSERT INTO 表A ……1行 SQL SELECT FROM 表A ……1行（ユニーク索引アクセス） CICS SYNCPOINT
SQL INSERT INTO 表B ……1行 SQL UPDATE 表C ……1行（ユニーク索引アクセス） CICS SYNCPOINT CICS RETURN

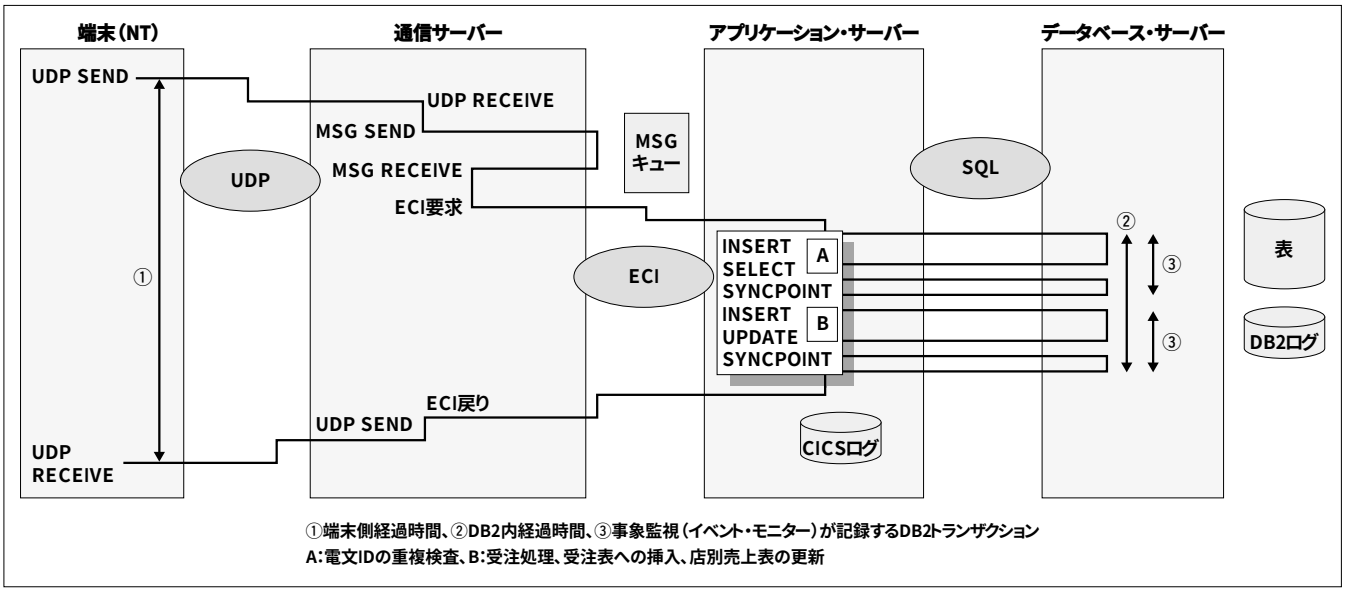


図5. 受注オンライン・トランザクションの処理フロー概要

(4)ディスク・IBM ESS 1台当たり404Gバイト、NVS 384Mバイト

- UDBのデータベースと活動ログはすべて、AIXミラーリングによって2重化されています。

(5)DB2の主な性能関連構成パラメーターなど

- 表スペースと活動ログは別のESS筐体に配置しました。
- 表と索引は異なる表スペースに格納し、表スペースはすべて同じESS筐体にあります。
- データ・バッファーの大きさはバッファー・プール=0.6Gバイト、拡張記憶域キャッシュ=1Gバイト。
- ログ・バッファー=2Mバイト(4Kバイト×512個)。
- 性能データ収集と問題判別のために事象監視(イベント・モニター)機能のデッドロック、接続、トランザクションを活動化しました。監査(AUDIT)機能も活動化しました。本番環境でも同様に活動化しています。

(6)CICSの主な性能関連パラメーター

- MaxServer=MinServer=50 アプリケーション・サーバーとUDBとの間の接続数。アプリケーション・サーバーが3個なので最大150個まで接続が可能としました。
- ECI_timeout=2秒 通信サーバーが要求したECIの経過時間の上限値。言い換えれば、2秒以内にアプリケーション・サーバーからデータベース・サーバーをアクセスして通信サーバーに戻らなければなりません。
- MaxRegionPool=20Mバイト。

8.3. ピーク時CPU使用率

ピーク時を想定して、400TPS～500TPSの大量トランザクションを、センター・システム内の端末から自作のトランザクション・ジェネレーターから発行したときの各サーバーにおける各ノードの平均CPU使用率と端末経過時間などを測定しました。

(1)各サーバーの各ノードのCPU使用率は問題ありません。

(2)通信サーバーのノードが10個のうち9個まで万が一、障害を起こして、1個のノードだけが稼働する縮退運転を行ったとしても、全トランザクション量におけるCPU負荷の観点からは処理可能です(410.6TPSのとき、8.3%×10=83%<100%)。

(3)アプリケーション・サーバーのノードがHACMPの待機ノードを含めた4個のうち、3個まで万が一、障害を起こしたとしても、同様に1個のノードだけで縮退運転が可能です(410.6TPSのとき、31.8%×3=95.4%<100%)。正常時だけでなく、ノード障害中の縮退運転時も、ピーク時トランザクション量が処理可能であることを確認しました。

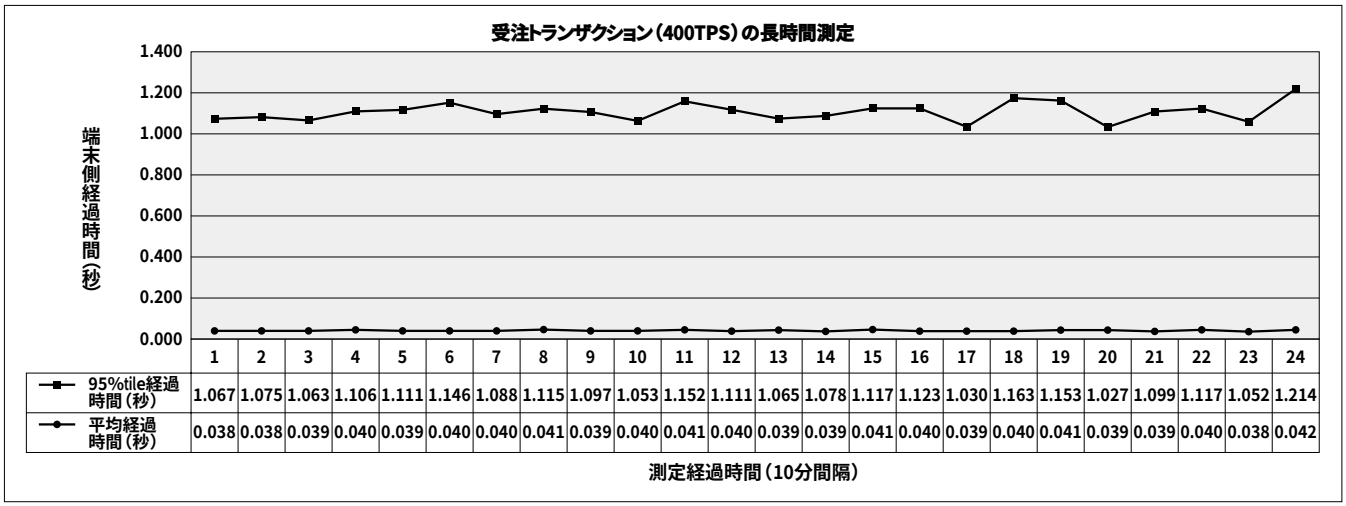
(4)トランザクション量に対して、各サーバーの各ノードのCPU使用率は線形に増加しているので、少なくとも506TPSまでシステムはボトルネックの現象は見られませんでした。

(5)図には示していないが、大量トランザクション量にもかかわらず、ディスクへの物理I/Oはボトルネックになっていませんでした。それはESSの大きなNVS(Nonvolatile Storage:不揮発ストレージ)キャッシュの効果と考えられます。

8.4. オンライン・トランザクションの端末側経過時間の観測結果(応答時間)

(1)端末側の平均経過時間の性能要件である5秒以内を問題なく達成しました。95%tile経過時間でも、5秒以内を達成しています。

(2)DB2内経過時間は、0.029秒以下でした。これは通信サーバーが設定したECIサービスのタイムアウト値の2秒よりも、



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
95%tile経過時間（秒）	1.067	1.075	1.063	1.106	1.111	1.146	1.088	1.115	1.097	1.053	1.152	1.111	1.065	1.078	1.117	1.123	1.030	1.163	1.153	1.027	1.099	1.117	1.052	1.214
平均経過時間（秒）	0.038	0.038	0.039	0.040	0.039	0.040	0.040	0.041	0.039	0.040	0.041	0.040	0.039	0.039	0.041	0.040	0.039	0.040	0.041	0.039	0.039	0.040	0.038	0.042

図6. オンライン・トランザクションの長時間測定における経過時間推移

はるかに短い値です。

(3)端末経過時間の性能要件を満たしていますが、細かく観察するとトランザクション量が、410.6TPSから506.3TPSに増えたとき、端末側平均経過時間と95%tile経過時間が伸びています。一方、DB2内経過時間はほぼ一定であり、安定していました。これは端末からデータベース・サーバーに来るまでのところに、何らかのボトルネックの兆候が現れていることを示しています。実際のところ、通信サーバーに装着されている電文の固有な符号化を支援するハードウェア・カードの処理能力が影響していることが分かりました。この測定時は、ノード当たりのカード装着枚数に制約がありましたが、増やせば解消される見込みです。

8.5. 長時間ピーク時における端末側経過時間（応答時間）

ピーク時の目標トランザクション量の80%の400TPSが長時間（約4時間）継続しても安定した経過時間が得られるかどうか、また、各サーバーのメモリー管理などに問題が発生しないかどうかの検証測定結果を図6に示しました。なお、図の横軸は10分間隔です。

9. おわりに

本論文で説明したシステム構成を検討する手順と過程は、WebSphereを中核のアプリケーション・サーバーとするe-ビジネスのためのサーバー・システムの基盤設計にも十分参考になるでしょう。UDBに関しての大きな違いは、Javaアプリケーションであるが故に、JDBCに置き換わることくらいです。JDBCドライバーやSQLJドライバーの性能が、課題として与えられた基幹業務に使用できるまで改善されれば、当システムに匹敵する高性能OLTPシステムが可能になるでしょう。

本論文が、高性能な基幹オンライン・トランザクション・システムを設計し、開発をする際に、少しでもお役に立てれば幸いです。

（ページ数および表記上の観点から、著者の了解を得て編集部にて手を入れてあります）

[参考文献]
[1] 菅原 香代子「ネットワーク・コンピューティングにおけるデータベース管理システムの位置付け」『PROVISION』No.18、日本アイ・ビー・エム、1998年