Windows Server で構築するミッションクリティカルシステムの勘所

Critical Success Factors of Mission Critical System Based on Windows Server

三ツ井　淳一

要　約　S-BITS プロジェクトで開発されたフルバンキングシステムには，日本ユニシスが蓄積してきたオープンシステム構築のノウハウが結集されている．S-BITS プロジェクトでは，Windows OS を中心としたオープン基盤上で，銀行業務の中心である勘定系システム「BankVision」を安定稼働させるために，様々な視点から検討および検証を行った．特に，プロジェクト開発時には実績が少なかった Windows 64bitOS の採用や SQL Server 2005 のデータベースミラーリング機能を積極的に採用することにより，性能要件，拡張性要件，高可用性要件を実現している．また，マイクロソフト社と協力し，定常的に記録しておく必要のあるデータを徹底的に洗い出し，一般的に Windows システムの弱点と言われている障害追究能力を向上させた．

Abstract　All the accumulated knowledge of open system implementation done by our company have been concentrated to the full banking system developed by the S-BITS project. In the S-BITS project, a variety of investigations and verifications have been made from various viewpoints to allow the banking system 'BankVision', the core of banking business applications, to run stably on top of the open systems infrastructure of Windows operating system. Especially, we succeeded to satisfy high performance, scalability and high-availability requirements by positively adopting Windows 64bitOS and the 'Database Mirroring' in SQL Server 2005 in spite of rare expertise about them in all over the world at that time. Furthermore, the enhancement of the troubleshooting capability generally called as "Windows weakness", has been achieved by exhaustive digging up of data that need to be gathered routinely, keeping close cooperation with Microsoft Corporation.

1. は　じ　め　に

日本ユニシス株式会社（以下，日本ユニシス）は，株式会社百五銀行（以下，百五銀行）と共同で地銀向け勘定系システム「BankVision」を開発した．この BankVision を中心としたフルバンキングシステムを S-BITS[*1] 共同アウトソーシングセンタ（以下，S-BITS 共同 OSC）として，2007 年 5 月より本格運用を開始した．日本ユニシスは現在，この百五銀行の成功を踏まえて，他の地方銀行へ積極的にセールスを展開している．

S-BITS 共同 OSC を立ち上げるため，S-BITS プロジェクトは，標準提供するシステム（標準システム）として，勘定系システム，基幹系 DWH システム，営業店インタフェースシステム（営業店 I/F システム），対外系システム，連携システムを開発した．また，顧客が必要に応じて利用可能なシステム（推奨システム）として，国際系システム，融資稟議支援システムが開発された．更に，これらのシステム群を運用するための運用管理システムやセキュリティ管理システムも構築している（図 1）．

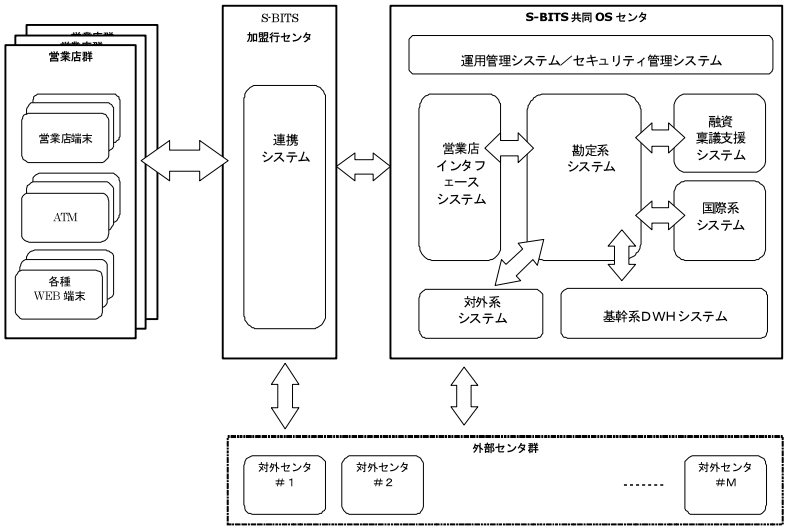

図 1　共同 OSC と提供するシステム概念図

2.　システム構成

S-BITS 共同 OSC で提供するフルバンキングシステムは，図 1 に示すように，勘定系システムを中心として稼働している．本章では，この勘定系システムの構成を中心としてシステム構成を紹介する．

2. 1　勘定系システムのサーバ構成

百五銀行の勘定系システムは，24 時間 365 日運用を実現するために，平日勘定系システムと休日勘定系システムを準備し，平日と休日で稼働するシステムを切り替えて運用を実施している．またこのシステムでは，サーバの二重障害が発生した場合でもオンラインが問題なく稼働できるような可用性要件から，平日系，休日系ともに，AP サーバ 4 台と DB サーバ 3 台のサーバ構成となっている（図 2）．AP サーバでは，後述する MIDMOST および ACAB をミ

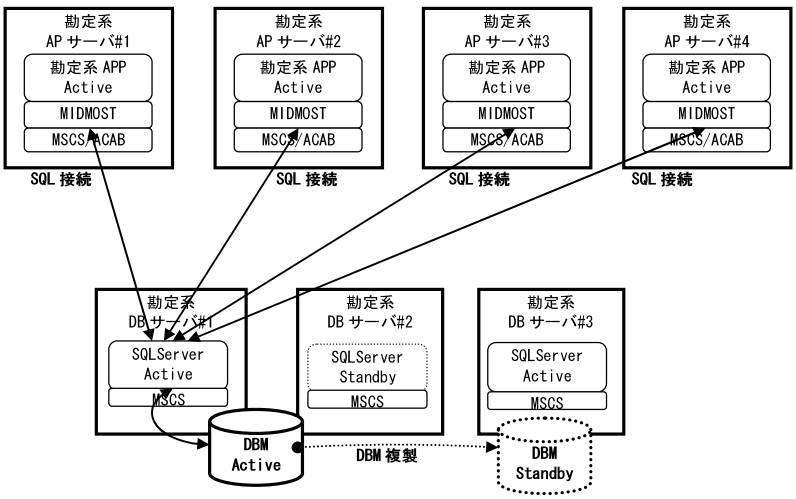

図 2　勘定系システム・サーバ構成図

ドルウェアとして，COBOL言語で開発された業務アプリケーションが稼働している．また勘定系DBサーバは，データベースマネージメントシステム（以下，DBMS）としてSQL Server® 2005を採用しており，更にSQL Server 2005のSP1（サービスパック1）にて正式リリースされたデータベースミラーリング（以下，DBM）機能を利用した構成を採用している．DBM機能の詳細および勘定系DBサーバのDBMの構成詳細については，2.3.3項で述べる．

2.2　ハードウェア構成の勘所

2.2.1　サーバ機器の選定

百五銀行勘定系システムにおいて，APサーバはES7000/One Xeon，DBサーバはES7000/One Itanium2を採用している．開発当時（2004年～2005年頃），Windows® オペレーティングシステム（以下，OS）は，64ビットアーキテクチャ版[*2]は主流ではなかったが，勘定系システムを稼働させるサーバとして，32ビットアーキテクチャのOSでは，メモリ空間の限界[*3]，拡張性，デスクトップヒープ問題[*4]などが不安視されていた．そのため，勘定系APサーバは，当時リリースが予定されていたIntel-64[*5]のCPU上で稼働するx64版OSの採用を決定した．x64アーキテクチャのOSは，32ビットアプリケーションと高い互換性をもち，性能劣化させることなく稼働させることが可能であることが主な採用理由であった．また，勘定系DBサーバは，EPICアーキテクチャ[*6]採用による高い性能評価結果が多数報告されており，SQL Server 2000での実績も高かったことから，Itanium2プロセッサの64bitOSを採用することを決定した．また，これらのサーバリソースやハードウェア構成（HBA構成，ラッキング構成等）については，実機検証，COBOLアプリケーションコード上の必要メモリの見積り，可用性や運用性の面から担当SE，担当技術，ハードウェア主管部署と検討し，最終的な構成を表1のとおり決定した．

表1　勘定系システムサーバスペック

	勘定系APサーバ（平日／休日）	勘定系DBサーバ（平日／休日）
サーバ機種	ES7000/600 (One Xeon)	ES7000/One Itanium2
OS	Windows Server 2003, Datacenter x64 Edition	Windows Server 2003, Datacenter Itanium Edition
CPU	64bit Xeon MP EM64T 平日系 Xeon 3.33 GHz x 4 休日系 Xeon 2.83 GHz x 4	64bit Itanium2 平日系 Itanium2 1.6 GHz x 16 休日系 Itanium2 1.6 GHz x 16
メモリ	8GByte Memory	16GByte Memory
内蔵ディスク	72GB x 3 (RAID1 + Hot Spare Disk)	72GB x 3 (RAID1 + Hot Spare Disk)
HBA Fiber Channel	FCH752323-PCX(Emulex LP10000DC-M2) x2 2Gigabit 2Ch Fiber HBA <LC> (2ch：ローカル x1 共有用 x1) x 2	FCH752313-PCX(Emulex LP1050-F2) x4 2Gigabit Fiber HBA <LC> ローカル用 x2，共有用 x2
HBA Network Interface	ETH33322-PCX (Intel PRO1000/MT Dual PWLA8492MTG3) x 2 2port Giga Copper <RJ45> (2ch：基幹用 x1 MSCS用 x1) x2	ETH33312-PCX (Intel PRO1000/MT PWLA8490MTG1) x 4 1port Giga Copper <RJ45> 基幹用 x2，MSCS用 x2

2.2.2　ストレージ機器の選定

ストレージ機器は，S-BITS共同OSC内の多数のサーバが接続する必要があることから，用途に応じて機種選定，接続構成を決定している．ディスクに対する大量のI/Oが発生する可

能性のあるバッチシステムについては，オンライン系システムとストレージ機器を別に準備し，性能に影響を与えないような配慮をしている．標準システムのオンライン系システム（勘定系，対外系，営業店 I/F システム）用として「SANARENA5800」，標準システムのバッチ系システム用として「SANARENA5200」，推奨系，運用およびセキュリティ系システムとして「SANARENA1890」を準備している．更に，勘定系システムのデータベースは，高い性能が要求されることを勘案し，SQL Server のデータファイルに対して I/O を分散させる物理ドライブ構成になっている（図 3）．

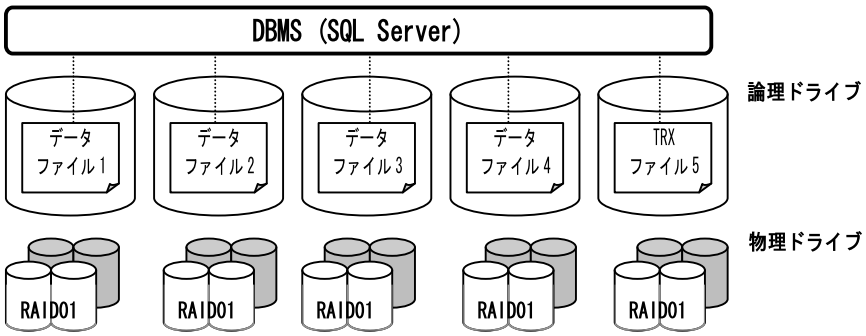

図 3　SQL Server データファイルの分散配置

2.2.3　サーバと周辺機器との接続構成について

ストレージサブシステムの SANARENA シリーズとのマルチパス I/O（以下，MPIO）を実現するために，日立製作所の HDLM（JP1/HiCommand Dynamic Link Manager）を利用している．このような構成は，勘定系システムに限らず，S-BITS 共同 OSC 内に存在するほとんどのサーバで採用しており，サーバとストレージ機器の間の接続が二重化されている（図 4）．この構成を採用することによって，ストレージサブシステム側のチャネル障害，ファイバケーブル障害，HBA 障害などが発生しても問題なく後続の IO 処理を行うことができる．

S-BITS では，ストレージサブシステムとサーバ間の接続障害をできる限り早く検知するための設定として，HBA（Emulex LP10000DC-M2，Emulex LP1050-F2）ドライバの設定を変更している．

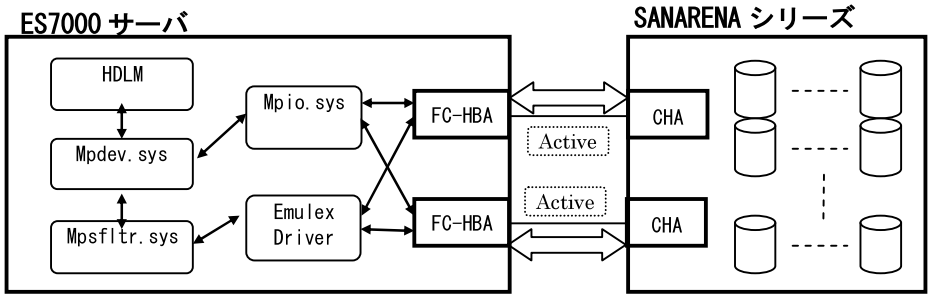

図 4　サーバとストレージ間の接続概要図[1]

また，サーバとネットワーク機器の接続においては，Intel 社製のネットワークドライバが提供する「SFT（Switch Fault Tolerance）」と「GEC（Gigabit Ether Channel）」を利用して

いる（図 5）．SFT はチーム化されている一つのアダプタをアクティブ，もう一方をスタンバイとして利用し，異なるスイッチに接続することができるのが特徴である．チーム内のアダプタ障害，アダプタとスイッチ間のケーブル障害，接続スイッチの障害を検出し，必要に応じてスタンバイ用アダプタへの切り替えを行う機能を備えている．一方 GEC は，サーバとスイッチ間のスループットを増加させるため Cisco Systems 社が開発したパフォーマンステクノロジで，フォルトトレランス（アダプタ障害，ケーブル障害）とロードバランシングの両方を提供している．

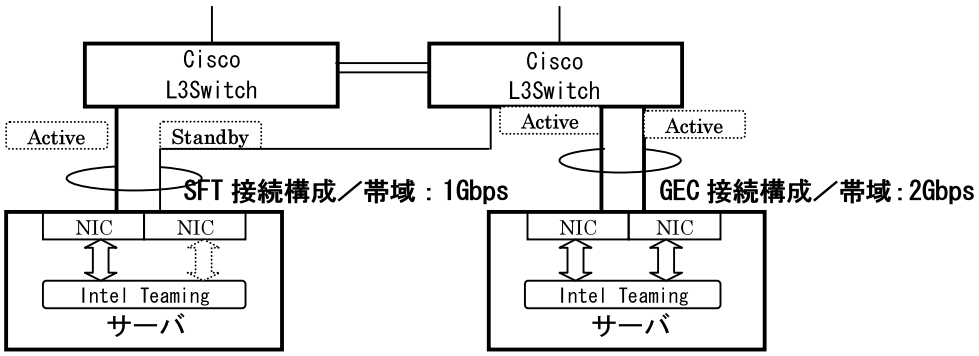

図 5　SFT 接続と GEC 接続概要図

勘定系システムにおいては，実機検証の結果を踏まえ，勘定系 AP サーバはネットワーク障害時の切り替えを優先して SFT 接続構成を，勘定系 DB サーバは AP サーバからの SQL の同時アクセス時の性能を優先して GEC 接続構成を採用した．

2.3　ソフトウェア構成の勘所

2.3.1　MIDMOST の利用

MIDMOST は，ミッションクリティカルシステム用に開発された日本ユニシスのミドルウェアであり，主に金融ユーザに利用されている．S-BITS 共同 OSC 内のほとんどのシステムのミドルウェアとして利用されており，勘定系システム（基本的に勘定系 AP サーバ）と他システム間のオンライン連携は，MIDMOST の通信機能である MIDMOST/CP[*7] によって行われている．百五銀行における ATM および営業店端末からの通信（図 6）を例に，MIDMOST がどのように利用されているか説明する．

顧客が営業店の窓口や ATM で取引を行うと，百五銀行センタ設置の通信サーバに取引電文が送信される．通信サーバは受信した取引電文を S-BITS 共同 OSC 内に設置されている営業店 I/F サーバへ送信を行い，営業店 I/F サーバが勘定系サーバへの振り分けを行う．勘定系サーバからの返信についても同様の経路で返信される．通信サーバと営業店 I/F システム間，営業店 I/F サーバと勘定系サーバ間で MIDMOST の通信機能により電文送受信が行われている．これらの通信は，ラウンドロビン送信機能により電文が振り分けられて送信されている．また，あるサーバに障害が発生した場合，MIDMOST/CP が持つ TCP セッションのヘルスチェック機能により，障害ノードを検知して，送信先リストから障害サーバを削除し，正常ノードのみに振り分け処理を行う．

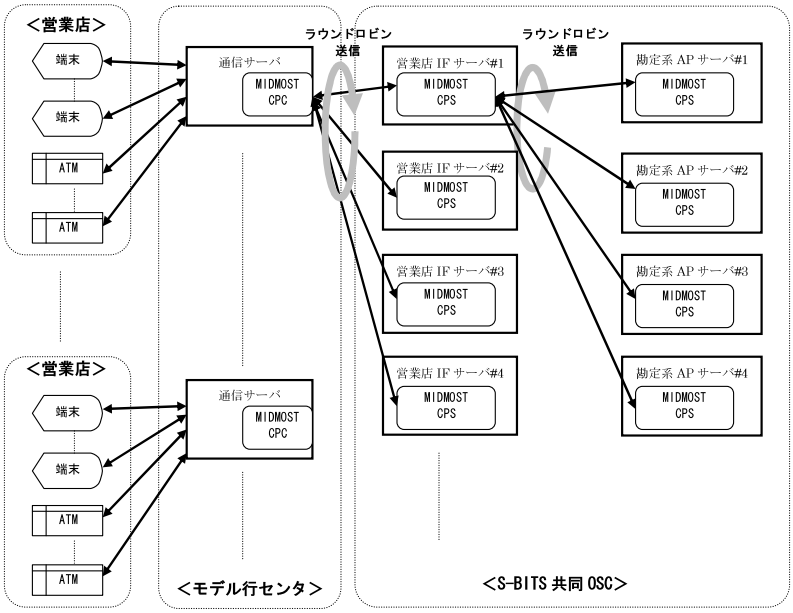

図 6　MIDMOST によるサーバ間通信

2. 3. 2　クラスタサービスと ACAB の利用

Windows Server® の可用性とスケーラビリティを提供するクラスタサービス（MSCS）は，一つのクラスタ内の複数サーバ間で常に通信が維持され，クラスタ内のいずれかのノードが障害の発生や保守のために利用できなくなると，別のノードがサービスの提供を開始する．登録可能なサービス（リソース）として Windows サービス，ユーザアプリケーション，ディスクリソース，印刷スプーラなど様々なリソースを監視対象とすることができる．また，日本ユニシスが開発したミドルウェアである ACAB は，クラスタサービスが提供していないきめ細かいプロセスの監視，複雑なフェイルオーバ条件の設定，リソース障害時のアクションなどの補完機能を提供している．MIDMOST で開発するプロセスの稼働環境は，基本的に ACAB を利用することが前提となっており，S-BITS においてもクラスタサービスと ACAB の組み合わせ環境上で MIDMOST のプロセスを稼働させている．また，S-BITS では ACAB のローカルディスク監視機能を利用して，一般的に早期発見が難しいとされる内蔵ディスクに対するアクセス障害（ディスクアレイコントローラ障害などによる）を実現している．

2. 3. 3　データベースミラーリングの利用

一般的に銀行の業務システムが停止するということは社会的な影響が大きい．日本ユニシスがフルバンキングシステムのアウトソーシングビジネスを展開していく中で，システムの障害復旧時間を極力短くすることは，重要な課題であり，顧客のニーズでもあると認識していた．このため，短時間での復旧が求められるシステムにおいては，SQL Server 2005 SP1 で提供された DBM 機能を利用している．マイクロソフト社が提供する DB サーバの障害に備えた高可用性ソリューションとして，前述したクラスタサービスを利用する選択肢もあったが，このソリューションでは，障害時の復旧時間が分オーダとなってしまう．これは，待機側のサーバが障害サーバよりサービスを引き継ぐ際に，「障害の検知～ DB 用ディスク領域の取得～ SQL

Server プロセスの起動～ DB のオープン」のような一連の処理を行うことに起因する．これに対し，DBM は二つのサーバに別々のディスク領域を準備し，両サーバ上で SQL Server のプロセス（インスタンス）を稼働させて DB を複製させながら動作するため，障害復旧時間を秒オーダにすることを実現している．

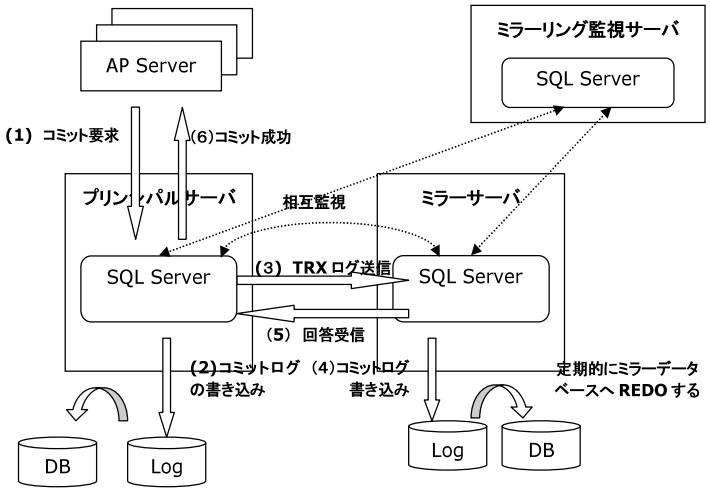

図 7　DBM 処理概要（同期転送・自動フェイルオーバ時）

DBM では，クライアントからの要求を処理するサーバをプリンシパルサーバ，複製を行っているサーバをミラーサーバと呼ぶ（図 7）．DBM では同期転送か非同期転送のどちらかを選択することができる．文字通り，非同期転送は二つのデータベースの複製が非同期で行われ，同期転送は二つのデータベースが完全に同期した状態でトランザクションが実行される．同期転送で，障害時に自動的に切り替え処理（自動フェイルオーバ）を行わせるためには，ミラーリング監視サーバ[*8] が必要となる．これは自動フェイルオーバが誤って行われないように，プリンシパルサーバの障害がミラーサーバとミラーリング監視サーバの両サーバで認識された場合のみ，自動フェイルオーバが行われる仕組みとなっている．

百五銀行の勘定系システムでは，DB サーバ障害時の切り替え時間が短縮されること，勘定

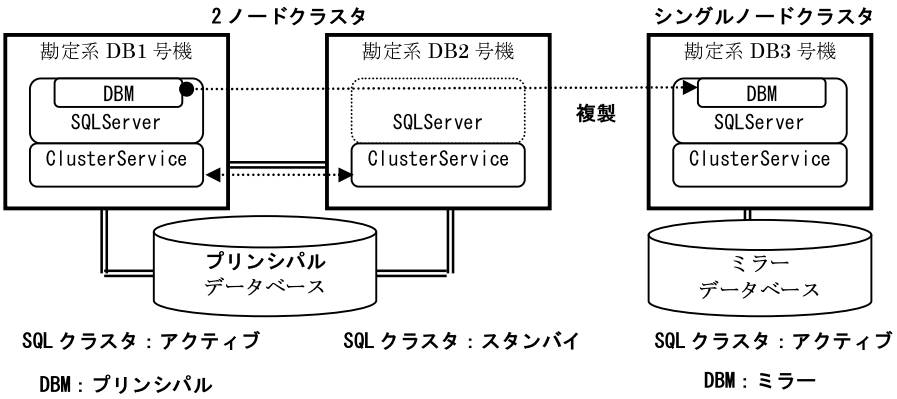

図 8　勘定系システムの DB サーバ構成および DBM 構成

系データベースの四重化（ハードウェア RAID による二重化と DBM による DB 複製で四重化）が可能なことから DBM を採用している．また，サーバの二重障害まで考慮し，プリンシパルサーバを 2 ノードクラスタ，ミラーサーバをシングルノードクラスタのサーバ構成（図 8）とした．

勘定系 DB サーバ 1 号機に障害が発生した場合，3 号機で稼働している SQL Server インスタンスが DBM の自動フェイルオーバにより，新プリンシパルサーバとして稼働する．その後，クラスタサービスによって，2 号機で SQL Server インスタンスが起動されるが，このとき 2 号機のインスタンスはミラーサーバとして起動される．DBM によって 3 号機が 1 号機の障害を検知し切り替わる時間と，クラスタサービスによって 2 号機が 1 号機の障害を検知し，フェイルオーバしてインスタンスを起動させるまでの時間を比較した場合，あきらかに DBM の切り替えが速く処理されるためこのような動作となる（図 9）．

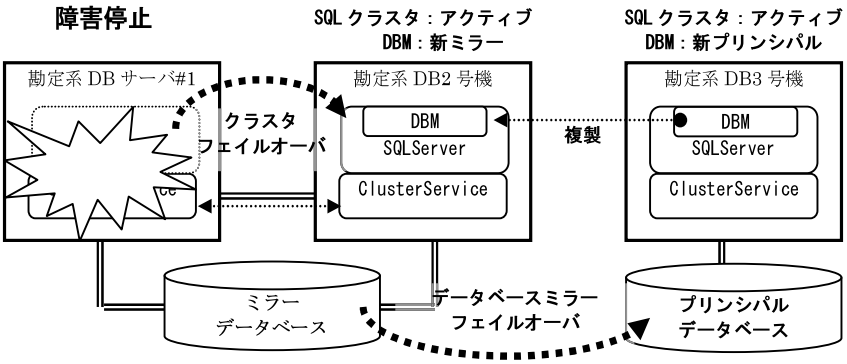

図 9　勘定系データベースサーバ障害時の動作

3.　障害に備えた取り組み

本章では，S-BITS プロジェクトにおいて実施した障害に備えた取り組みについて紹介する．まず，サーバ障害が発生した場合，他ノードからの障害検知方法，検知時間のチューニングについて紹介する．次に，日々の運用で発生する障害，課題に備えて，Windows Server や SQL Server において定常的に採取すべきデータ，障害発生時に取得すべきデータとしてどのようなものを取得しているか紹介する．

3. 1　障害検知のチューニング

図 6 で説明したように，ATM から取引要求があった場合，いくつかのシステムを経由して勘定系システムに到達する．勘定系システムで処理が完了すると処理結果を返信するが，その場合も同様である．S-BITS 共同 OSC 内のシステムでは，システム間（AP サーバ間）の連携は，MIDMOST の通信機能で行われ，AP サーバと DB サーバ間は，MIDMOST の DB インタフェースを利用し，ODBC[*9] ドライバ経由で SQL 通信が行われる．これらの通信中に障害が発生した場合，どのような仕組みで検知されるか，また，そのチューニングポイントについて紹介する．

3.1.1　AP サーバ間の障害検知について

勘定系システムは，他システムとの通信を，MIDMOST の通信機能を利用して実現している．MIDMOST は Winsock を利用して TCP/IP 上で通信を行う．また，MIDMOST の通信機能は，ノード間の通信が正しく行われているかどうかをチェックする機能を備えている．この機能では最終データの受信時間が一定時間以上経過すると，ヘルスチェック電文を送信し，接続が正常か確認している（図 10）．これらの動作に関連するパラメータは，MIDMOST の構成定義ファイルにて指定できる．

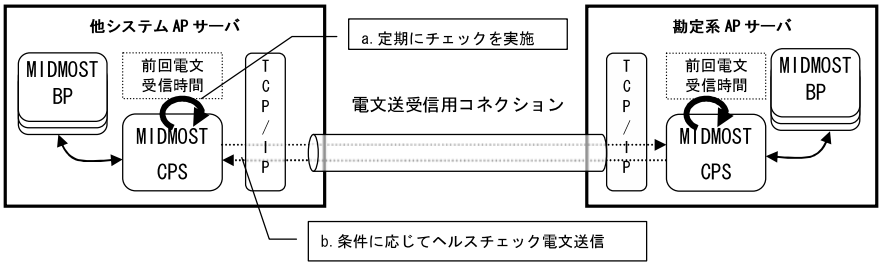

図 10　MIDMOST 通信機能のヘルスチェックの仕組み

3.1.2　AP サーバと DB サーバ間の障害検知について

MIDMOST 上で DB 接続を行う場合，開発者は MIDMOST のデータベースインタフェース機能を利用することができる．この機能を利用することにより，開発者は DB サーバとの接続処理やセッション管理に関する配慮が基本的に不要となる．

また，DB サーバの障害を検知する方法として，MIDMOST は DB との接続が良好かどうか確認する機能を持っており，1 秒間隔でデータベースに対してテスト用の SQL 文を発行している（図 11）．

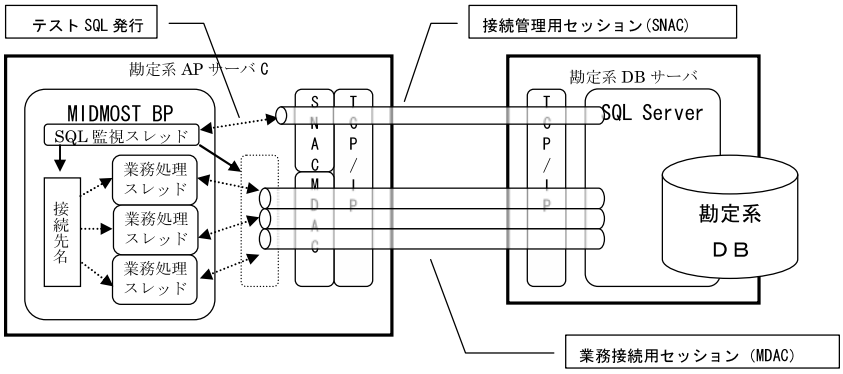

図 11　MIDMOST の DB 接続監視の仕組み

MIDMOST 内に存在する SQL 監視スレッドが定期的にテスト SQL 文を発行して回答があるかどうかを監視している．業務処理スレッド用に定義されているトランザクションタイムアウト値までにテスト SQL 処理が完了しない場合には，プロセス内の全接続を破棄して再接続処理を実施する．なお，トランザクションタイムアウト値より TCP 再送タイムアウト時間が小さい場合は，サーバ障害時はトランザクションタイムアウトまで待つことなく検知される．

3.1.3　DBM の障害検知と自動フェイルオーバについて

2.3.3 項で述べたように，DBM で自動フェイルオーバを実現するためには，プリンシパルサーバ，ミラーサーバ，ミラーリング監視サーバの 3 台（3 インスタンス）を準備する必要がある．これら 3 台の間は，お互いの状態を監視するために，DBM 用の接続を保持し，すべてのサーバインスタンスが接続の状態を監視する．パートナー（プリンシパルサーバとミラーサーバ）間の相互接続が失われると，各パートナーはミラーリング監視サーバを使用し，データベースとして機能しているサーバがないか確認する．ミラーサーバからプリンシパルサーバへの接続が失われても，ミラーリング監視サーバとの接続が失われていなければ，ミラーサーバはミラーリング監視サーバにアクセスし，ミラーリング監視サーバとプリンシパルサーバとの接続を確認してその挙動を決定する．以下にその挙動を記述する[2]．

- プリンシパルサーバとミラーリング監視サーバとの接続が失われていなければ，自動フェイルオーバは行われない．プリンシパルサーバは引き続きデータベースとして機能する．ミラーサーバと再接続された場合には，その間蓄積したログレコードをミラーサーバに送信する．
- プリンシパルサーバからミラーリング監視サーバへの接続も失われている場合，ミラーサーバはプリンシパルデーターベースが使用できないことを認識する．この場合，ミラーサーバはすぐに自動フェイルオーバを開始する．
- ミラーサーバからミラーリング監視サーバおよびプリンシパルサーバへの両接続が失われた場合，プリンシパルサーバがどのような状態であっても，自動フェイルオーバは行われない．

また，DBM は独自のタイムアウトメカニズムを持っており，一定間隔で各接続に対して PING[*10] を送信し，接続が保たれているかどうかを評価している．ミラーリングセッションの別のインスタンスへこの PING メッセージを送信し，指定した時間（タイムアウト時間）まで待機し，万が一返信がない場合には接続解除されたと判断する．このタイムアウト時間の決定にあたっては，他の障害対策とのバランスをとる必要がある．タイムアウト値を小さくし過ぎると，本来 DBM とは無関係にリカバリ可能な障害（例えば NIC 障害，ネットワーク機器障害など）を誤検知し，DBM の自動フェイルオーバを発生させてしまう可能性がある．したがって，サーバ機器構成や各機器のリカバリ時間などを勘案して，最終的なタイムアウト値を決定する必要があり，本番環境で想定どおりの動作をすることも確認すべきである．S-BITS では，ネットワークカードの障害やネットワーク機器の障害が発生した場合のリカバリが 5 秒以内に行われることを確認した上で，タイムアウト値を 5 秒に設定している．

3.2　障害時に備えた定常採取データと障害時の採取データ

一般的に，Windows Server のミッションクリティカルシステムでの利用を懸念する理由として，障害追究の難しさが挙げられる．これは，オープンシステムを基盤として利用する場合，複数ベンダが提供するソフトウェア，ハードウェアを組み合わせて利用していることもその要因の一つだが，そもそも，Windows Server 自体を考慮なく運用してしまうと障害時に解析するための資料がほとんど取得されず，障害発生個所の特定すらできない状況になってしまうことがある．S-BITS では，マイクロソフト社と協力し，発生した障害や問題を可能な限り追究できるようにするために，どのようなデータを定常採取しておくべきかを検討した．Win-

dows Server と SQL Server に分類して紹介する．

3. 2. 1　Windows Server 2003 の障害分析用データ

Windows Server で障害が発生した場合の追究情報の収集ツールとして，マイクロソフト社より提供されている Microsoft Product Support Reporting Tools（通称 MPSReport）がある．このツールにより，システム情報，ドライバ情報，ネットワーク設定，イベントログ，ディスク情報など様々な情報をまとめて取得することができる．一般的に，トラブルが発生した場合には，これらの情報が基本情報として取得され障害解析が行われる．しかし，これらの情報だけで全ての障害を解決することは困難である．特に，性能系の問題の解析に必要な情報は決定的に不足している．性能問題の解析には，Windows Server のシステムモニタ機能（標準機能）で，パフォーマンスデータの定期的な取得が必要である．特に，プロセッサ，メモリ，ディスク，ネットワークなどの基礎的なデータは，可能な限り短い周期で取得したほうがよい．S-BITS では取得するパフォーマンスカウンタごとに取得間隔（1 秒，10 秒，60 秒，600 秒）を決定している．但し，これらのデータ取得周期を短くする場合（特に 1 秒ごとに取得するデータの場合），取得データが膨大になるため，取得するデータファイルの切り替えやファイルのガベージを考慮する必要がある．

その他，S-BITS では障害時の追究情報として，意図的にメモリダンプを取得できるような仕掛けを組み込んでいる．MIDMOST のプロセス（業務系プロセスを含む）の障害時は，User Mode Process Dump を利用し，例外発生時のプロセスダンプを取得している．また，クラスタサービスや SQL Server インスタンスが障害となった場合，Windows Server 自体が正常に稼働していない可能性も考えられるため，サーバを強制終了させて，OS のフルダンプを取得している．更に，ネットワーク系の障害時や追究情報としてサーバの送受信データの確認が必要な場合に備え，Windows Server Support Tools が提供している「netcap」コマンドを利用し，ネットワークデータを定期的にファイルに出力するツールも準備している．

3. 2. 2　SQL Server 2005 の障害分析用データ

一般的にデータベースマネージメントシステムにおける問題のほとんどは，データベースのアクセス性能に関する問題である．S-BITS では，この種の問題をできる限り早く解決するために，発生前後のアプリケーションの稼働状況，SQL Server 内の各種リソース使用状況など，様々な面から分析が可能なように定期的にデータを取得している．マイクロソフト社の提案により，SQL プロファイラ（SQL トレース），トレースフラグ，システムモニタ（パフォーマンスログ），DMV 監視（Dynamic Management View）の各種情報を取得している．

SQL プロファイラは，デッドロックの数，重大なエラー，ストアドプロシージャと Transact-SQL ステートメントのトレース，ログインの利用状況など，サーバおよびデータベースの利用状況を監視する．また，SQL プロファイラのデータを SQL Server テーブルやファイルにキャプチャしておけば，後で分析することができる．しかし，これら全てのデータを取得する場合，性能に多大な影響を与えるため，取得するデータの取捨選択が必要である．トレースフラグは，インスタンス起動パラメータとして設定し，サーバ固有の特性設定や特定の動作を切り替えるときに使用される．また，パフォーマンス問題の診断や，ストアドプロシージャや複雑なシステムのデバッグにも使用される場合がある．SQL Server を導入すると SQL Server

固有のカウンタがシステムモニタに登録され，SQL Server のパフォーマンス分析に必要な情報が取得可能になる．DMV は SQL Server 2005 より新しく提供された動的管理ビューであり，インスタンスのヘルス状態の監視，問題の診断，パフォーマンスのチューニングなどに有用である．表 2 および表 3 に，S-BITS で利用しているトレースフラグと監視対象の DMV を示す．

表 2　トレースフラグの設定

	トレースフラグの説明
205	AutoStats の結果として統計依存のストアド プロシージャが再コンパイルされる時点をレポート
818	報告されない入出力の問題を検出するために追加された新しい SQL Server 診断機能
1204	デッドロックの解決を行うために使用
1451	データベースミラーリングの追加情報をエラーログへ書出し
1453	プリンシパルからミラーへのログ転送に際し，設定されている遅延を無効にする
1460	Redoスレッドのスケジューリング情報をERRORLOGへ出力
1485	既定で4プロセッサ毎に起動されるREDOスレッド数を，プロセッサ数へ増加させる
2544	SqldumperでSQLのダンプを取得するにあたり，「フルダンプを採取する」事を指定
2546	SqldumperでSQLのダンプを取得するにあたり，「全スレッドをダンプする」事を指定
3400	リカバリタイミングでのERRORLOGへの出力
3504	チェックポイントページフラッシュ情報をERRORLOGへ出力
3605	デッドロック検出時に収集した情報をERRORLOGへ出力

表 3　DMV 監視対象

DMV名/監視内容	DMV説明[2]
sys.dm_os_schedulers スケジューラ監視	このビューは，スケジューラの状況の監視やランナウェイタスクの特定に使用できる．
sys.dm_exec_requests クエリ監視 （リソース消費状況）	SQL Server 内で実行中の各要求に関する情報を返す．
sys.dm_exec_query_stats クエリ監視 （クエリプラン毎の処理時間） （データベース毎のメモリ消費）	キャッシュされたクエリプランの集計パフォーマンス統計を返す．このビューには，一つのクエリプランにつき 1 行のデータが含まれており，その行の有効期間は，クエリ プラン自体に関連付けられている．つまり，プランがキャッシュから削除されると，対応する行もこのビューから削除される．
sys.dm_os_buffer_descriptors メモリ監視 （オブジェクト毎のメモリ消費）	SQL Server インスタンスのデータベースで使用されているバッファプールのバッファ記述子を返す．
sys.dm_io_virtual_file_stats ペンディングI/O監視	データファイルおよびログファイルのI/Oの統計を返す．
sys.dm_db_index_operational_stats インデックス使用状況監視 （行ロック数）	インデックスに対するI/O，ロック，ラッチ等の情報をレポートする．ここでレポートされるデータはインデックスのメタデータがメモリに読み込まれた以降の統計になっている．
sys.dm_db_file_space_usage Tempdb空き容量監視	データベースファイル毎の使用状況をレポートする．SQL Server 2005ではtempdbのデータベースファイル情報のみレポートしている．
sys.dm_exec_connections カーソル監視	各種データベースに関連する SQL Server のインスタンスに対してローカルやリモートのさまざまなユーザが確立した接続に関する情報と，各接続の詳細を返す．
sys.dm_tran_locks ロック競合監視	現在アクティブなロックリソースの情報をレポートする．

4.　セキュリティ対策について

銀行の主業務である勘定系システムをアウトソーシングセンタで運用するには，セキュリティ対策にも非常に高いレベルが要求される．S-BITS プロジェクトでは，高いセキュリティレベルを実現させるために，S-BITS 共同 OSC セキュリティポリシを策定し，物理セキュリティ（入退室管理，監視カメラの設置など），セキュリティ運用（組織作成，セキュリティ監査の実

施など)，IT セキュリティ（ネットワークセキュリティ対策，サーバセキュリティ対策など）の観点から対策を実施している．本章では，IT セキュリティ対策の面から，主にネットワークセキュリティ対策のファイアウォールの実装，サーバの要塞化について，実装事例を紹介する．

4.1　ファイアウォールと利用ポートの固定化

S-BITS 共同 OSC では，不要なネットワーク通信を遮断するためにファイアウォールを設置してアクセス制御を行っている．ネットワーク上でアクセス制御を行うことにより，情報漏洩・不正アクセス，ウィルスといったセキュリティの脅威を防いでいる．S-BITS 共同 OSC のネットワークセキュリティ対策は，図 12 に示すような多層防御の考え方で構築されており，外部ネットワークからのアクセスにおいて重要な顧客データへのアクセスが最も厳しく制限されている．

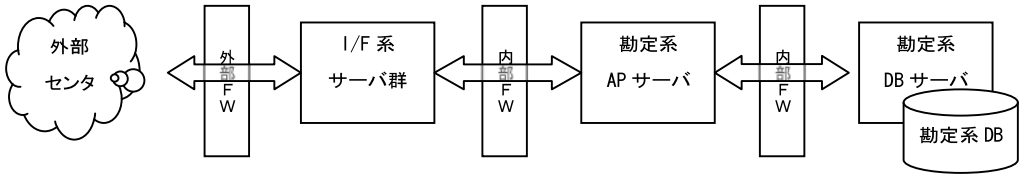

図 12　ネットワーク多層防御

図 12 に示すように，S-BITS 共同 OSC では外部センタとの接続用のファイアウォールだけでなく，センタ内においてもファイアウォールを設置し，利用する通信ポート以外のポートでは通信できないような防御をしている．更に，Windows Server でよく利用される RPC（Remote Procedure Call）の動的ポート割り当てを一定ポート範囲内で利用する設定，オンライン連携用の MIDMOST 用通信ポートの固定，AP サーバと DB サーバ間の SQL 通信ポートの固定化などを実施し，利用ポートを極小化している．

4.2　サーバの要塞化

サーバの要塞化を行う目的は，想定される多種多様なセキュリティリスクを最小化することにある．そのため，脆弱となる可能性のある部分をあらかじめ排除しておくことが重要となる．S-BITS において実施している主な要塞化対策として，「不要サービス無効化」，「ユーザ/パスワード管理」，「ウィルス監視」，「セキュリティ修正プログラム適用」が挙げられる．Windows Server では，インストールした状態のままだと使用用途に関わらず，様々なサービスが稼働してしまう．このため，サーバの用途に応じてサービスの停止/無効化の設定を行うことがセキュリティ対策となる．S-BITS 共同 OSC 内の全てのサーバは一つのドメインとして構成されており，要塞化を実現するために，Active Directory のグループポリシーオブジェクト（以下，GPO）を利用し，不要なサービスを無効化している．これは，不要なサービスを単に起動しないだけでなく，不正なサービスの起動を防止する効果もある．また，「ユーザ/パスワード管理」を一元的に行っている．更に，ローカルサーバ上の不要なローカルユーザアカウントの削除，パスワードの有効期限設定，複雑性の確保，パスワード履歴保持による繰り返し使用の回避なども行われており，これらの設定は Active Directory のドメインポリシーによって

実現している．

次に，ウィルス対策として，全てのサーバ上にウィルス防御ソフトウェアを導入し，常駐監視している．パターンファイルやエンジンも定期的にアップデートして最新に保っている．これは，主にサーバメンテナンスなど，外部から電子媒体を持ち込む作業でのウィルス侵入を許さないためである．

最後に，Windows Server を運用していく上で，考慮しなければならない点として，マイクロソフト社から月に一度リリースされるセキュリティ修正プログラムの適用がある．S-BITS 共同 OSC のネットワークは，インターネットに直接接続していない内部ネットワークであり，前述したように多層防御でファイアウォールを配置しており，高いセキュリティネットワーク環境が確保されている．ただし，このような環境であっても，未知のウィルスが混入する可能性は完全には否定できない．したがって，OS や DBMS のような基本ソフトウェアの脆弱性は，定期的に改善しておく必要がある．S-BITS 共同 OSC では，定期的にリリースされるセキュリティ修正プログラムを月一度のメンテナンス作業にて適用している．

5. お わ り に

S-BITS プロジェクトでのフルバンキングシステム開発において，Windows Server と SQL Server を中心としたオープンプロダクトの機能をどのように活用し，どのような点に留意したかについて紹介してきた．今回紹介できなかった他の分野の構築においても様々な検討の結果が活かされている．特に運用管理については，JP1 プロダクトを利用して大規模（100 台近いサーバ，大量のバッチ数，大量のバックアップ要件など）の運用要件を勘案したシステムが構築されている．

S-BITS プロジェクトは，2007 年 5 月に無事本番を迎えることができた．しかし，本番稼働後，本格的にアウトソーシングセンタとしての運用が開始され，新たな課題も見えてきている．その一つとして，今後，S-BITS 共同 OSC は，複数の銀行システムを同時に開発し，運用を実施していかなければならないということが挙げられる．限られた人員，コスト，期間の中で，安全かつ効率的に開発および運用ができなければ，顧客満足度の向上，新規顧客の獲得，収益の向上などに結びつかないだけでなく，アウトソーシングビジネスとして成功したとは言えないであろう．

S-BITS 共同 OSC は，日本ユニシスのオープンシステムを基盤としたアウトソーシングビジネスを成功させるための試金石であり，その成功には，直接携わる開発担当者，運用担当者だけに留まらず，S-BITS に関わる様々な担当者の協力が必要不可欠であり，今後も日本ユニシスグループ全体で協調して取り組むべきミッションと考える．

＊ 1　S-BITS（= Succeeding Banking Information Technology for Success consortium，エスビッツ）新たなバンキングシステムを研究し，その実現のため，対応策の策定と実証モデル（ひな型）開発を行い，参加行に提供することを目的とした，次期バンキングシステム検討・検証コンソーシアム．日本ユニシスと日本ユニシス地方銀行勘定系ユーザ 7 行で構成されている．2000 年 11 月 15 日に設立．また，コンソーシアムで策定したグランドデザインを実現するために発足した次期オープン基幹系システムの共同開発プロジェクトをいう．

＊ 2　当時，Windows の 64bit アーキテクチャ版は，Itanium 版のみ発売されており，x64 版については，AMD 版のみリリースされている状況だった．

＊ 3　32 ビット Windows で動作するプロセスの仮想メモリサイズは，デフォルト値で 2GB である．

一方，IA64 システムでは 7152GB，x64 システムでは 8192GB，と大幅に拡張されている．

＊ 4　デスクトップヒープ領域は，Windows2000 において，OS の仕様上システム全体で 48MB の固定領域であったために発生する同時稼働数の問題である．Windows Server 2003 では，32bitOS，64bitOS ともにこの領域が拡張可能となっており改善されている．

＊ 5　IA-32 の 64 ビット拡張であり，当初，EM64T（Intel Extended Memory 64 Technology）という名称だった．IA-64 とは異なる．

＊ 6　Explicitly Parallel Instruction Computing の略，複数命令を並列に実行するための情報を，あらかじめプログラム内に埋め込んでおくため，効率良くプログラムの実行が可能．

＊ 7　MIDMOST のコミュニケーションプロセスの略名．プロセスの種類として，サーバ用機能をもつ常駐プロセスの CPS と，クライアント機能をもつ常駐プロセスの CPC がある．

＊ 8　ミラーリング監視サーバは，別名ウィットネスサーバと呼ばれることがある．

＊ 9　Open Database Connectivity の略．Microsoft 社によって提唱されたデータベースにアクセスするためのソフトウェアの標準仕様．

＊10　DBM 独自の PING は，通常の PING コマンドとは異なる．DBM 独自の接続確認のための PING 処理をするプロトコル．

参考文献　［1］ David Solomon，Mark Russinnovich，豊田孝　監訳，インサイド Windows 第 4 版 下，日経 BP ソフトプレス，2005 年 8 月，第 9 章

［2］ SQL Server 2005 Books Online，http://msdn.microsoft.com/ja-jp/library/ms130214.aspx

執筆者紹介　三ツ井　淳一（Junichi Mitsui）

1993 年 学習院大学理学部数学科卒業．同年日本ユニシス(株)入社．社公向けオープン系システム開発，BANCS システム開発，MIDMOST 開発，S-BITS プロジェクト 等を経て現在，共通利用技術部 S-BITS 基盤技術室に所属