開発技術と開発プロセス

# 高信頼性と高速バックアップを実現するSANの活用

生産品質強化本部
設計・インフラ監理センター
田中　薫

## 1．はじめに

サーバー（特にDBサーバー）に使用するハードディスクには、高い信頼性と高速性が要求される。ハードディスクの改良が進み、100GBを超える容量で高速の製品も珍しくなくなったが、単体のハードディスクでは信頼性に限界がある。一方、SAN（Storage Area Network）は、近年一般化し、製品も多く販売されるようになった技術だ。一言で言えば、外部記憶装置間および記憶装置とコンピュータの間を結ぶストレージ専用の高速なネットワークであり、本プロジェクトでは、このSANを使用してDBサーバーの構築を行った（使用した製品は、NECのiStorage S2300）。本稿では、その際の構築の状況や、設計時に考慮すべき点について述べる。

## 2．DB用の物理媒体としてのSAN

SANは、RAID（Redundant Arrays of Inexpensive Disks）技術の発展形と考えることもできる。RAIDは、ディスクを冗長構成にして、二次記憶を構成する複数のディスクのうち１本が故障しても、システムの動作に影響しないようにする技術である。サーバーの処理を実行中にも、故障した１本を取り替えることができるようになり、24時間365日稼動するサーバーの二次記憶として使われている。

SANに基づく製品は、サーバーとの接続チャネルの複数化・高速化と、筐体内に自律的な管理機能を持たせることにより、高信頼性とともに、高度のデータ保全性を実現しようとするものだ。

SAN製品は、通常複数のサーバーに対して大容量の二次記憶を提供し、次のような特徴を備えている。

・多くの製品は複数の OS に対応し、メインフレームからWindows OSまで幅広いサーバー類の二次記憶を統合するソリューションとして注目されている。
・サーバーとの接続は、高速のFibre Channelであり、SCSIより速い。二次記憶へのバスに専用のネットワーク（SAN）を使うことで、広帯域･高パフォーマンスを実現する。
・内部のディスク構成はRAID にすることができ、信頼性が高い。これにより、個々のディスクの故障があっても、サーバーの動作を止めない。
・容量は、数百GBから数百TBが可能で、用途により分割して使用できる。用途と信頼性の程度に応じて、多くの製品が販売されている。
・内部的な管理機能を使用して、ディスク内容のコピーを作成できる。

マスターボリューム（MV）に対して、複数のレプリケーションボリューム（RV）を設定することが可能で、MV←→RV間のコピー動作は内部の管理機能により行うことができる。

つまり、サーバーから見た場合、SAN上のボリュームは、高速・大容量の通常のディスクと同様だ。特殊な使い方をしなければ、サーバー側で考慮する必要がないため、気軽に導入することが可能である（図１）。

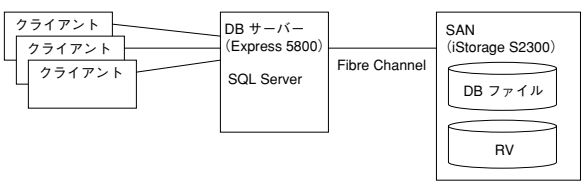


**図1　DB用の物理媒体としてのSAN**

なお、SANと似たものにLANに接続して使用するNAS（Network Attached Storage）というものがあるが、これは、ネットワークドライブを供給する機能に特化したサーバーと考えれば良い。

さて、このSANをDB サーバーの二次記憶として使用することには、次のような意味がある。

・二次記憶の高信頼性と高速化を享受できる。
・SANのスナップショットバックアップとスナップショットリストア （NEC製品での機能名） を使うことで、DBのバックアップ・リストアを高速に実行可能である。
・SANの内部に作成したレプリケーションボリュームを他の用途に使用できる。

DBは、その技術確立の当初から、データの安全性と可用性が重要な課題であった。そのために、ログによる更新途中のデータの管理、トランザクション機能の実装、本体データやログデータファイルの二重化など、さまざまな工夫が行われてきた。しかし、これらの努力もDBを格納する物理領域の信頼性が低ければ意味をなさない。DBサーバーにRAID構成のディスクを使用するのはいまや常識であるし、今後は、SANを活用する機会が増えていくだろう。

## 3．DBバックアップ・リストアの高速化

DBのサイズが大きいとき、SQL Serverなどの普通のバックアップコマンドでDBのバックアップを取得すると長い時間がかかり（20GBで30 分～ 1 時間）、サーバーへの負荷も大きい。

これは、DBのデータファイルに格納されたデータをバックアップファイルに転送する時間が必要なためで避けようがない。テープドライブを複数台設置して、バックアップのデータストリームをパラレル化するなどの工夫もできるが、DBのサイズが1TB に近くなると、バックアップの設計が最重要課題になることはおわかりだろう。

SANを使うとDBのバックアップ・リストアがなぜ高速化されるかは、少し技術的な説明が必要になる。SANはDBサーバーにボリュームを提供する以外に下記のような内部管理機能を持っている。

・設定したボリューム間にマスター/レプリケーションの関係付けを行い、マスターボリュームの内容をサーバーの動作とは無関係にレプリケーションボリュームにコピーすることができる （MV→RVの同期）
・その逆に、レプリケーションボリュームの内容をマスターボリュームにコピーすることもできる （RV→MVのリストア）

これらの機能を使用すれば、DBのデータ本体およびログ用のファイルを格納するボリュームをマスターボリューム（MV）とし、SAN内部でレプリケーションボリューム（RV）を定義しておき （DBサーバーからはMVしか見えない）、MV→RVの同期が完了した時点でMV、RV間の関係を切り離せば、RV上にその時点でのデータのコピーが残る。逆に 2 番目の機能を使用して、RV→MVにデータをリストアしてやれば、以前にRVにコピーした状態をMV上に復元することができる（図2）。

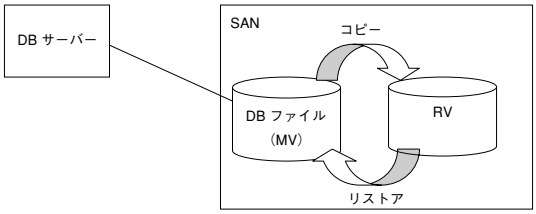


**図2　SANの内部管理機能**

これらの操作は、MV、RV間の接続・切り離しなので、普通にバックアップ・リストアを行うのに比べれば一瞬にして終了すると言ってよいだろう（実際のデータのコピーは、その前後にSAN内部でバックグラウンドで行われている）。

MV、RVを用意するためには、相当のディスク容量を確保する必要があるが、DBのバックアップ・リストアを高速化するためのコストと考えれば、それほど高いものではない。

実際の動きはもう少し複雑なので、以下に補足する。

・SQL Serverにバックアップを認識させる

SQL ServerのDBファイルを別なサーバーにコピーしてDBをアタッチすると、DBの内容をそのサーバー上で参照できる。SANの機能で、DBが格納されたボリュームをコピー（してアタッチ）すれば、元に戻した際にコピーを取った時点でのDBとして使用可能にはなる。しかし、通常のDBのバックアップのように、バックアップ以降のログを適用（ロールフォワード）して、ログのバックアップ時点までのDBを回復してやるためには、SQL ServerにSAN でコピーを取った時点でバックアップを行ったと認識させなければならない。

SQL Serverは、DBのバックアップ（フル・差分）、ログのバックアップなどを行うたびにその記録を取っている。そこで、SANのユーティリティとして提供される（NECの製品の場合、iSMsql_snapshotbkup.exeやiSMsql_snapshotrst.exeなど）機能により、SQL Serverに、SAN上でのコピーを取得したときにDBのバックアップを取得したことを教え、後にそのデータを使用してDBのリストアを行ったときに通常のDBバックアップから戻した状態になったものとして、そのDB ファイルを認識させることで、その後にログのロールフォワードなどを行うことができるようにすることが可能だ （ただし、このユー

ティリティを使用するためには、OSとして、Windows2000 Advanced ServerとSQL Serverにも、2000 Enterprise Editionを使用する必要がある)。

・ボリュームリストア時にMVがすぐに使用可能になる仕組み

RV→MVのボリュームリストアを行った場合、RVからMVへのコピーはその時点から開始される。このコピーが終了しなければ、MVをそのままDBとして認識させることはできないが、SANの機能として、コピー処理そのものはバックグラウンドで開始して、RVの内容をMVの代わりにサーバーに見せてやることが可能な仕組みが備わっている。この機能のおかげで、ボリュームのリストアを開始した直後から、MVの内容がすでにコピーされたものと同じように使用することができるようになる。バックグラウンドのコピーが終了した時点でRV→MVの関係付けは解消されるが、コピーを行っている間にも、MVへの更新は可能であり（更新情報は別なところに取っておき、最終的にMVへの更新として反映される)、サーバーから見ると、リストアが一瞬にして終了するように見える。

## 4．レプリケーションボリュームの使い道

レプリケーションボリューム（RV）は、MVから切り離された時点、または、スナップショットバックアップが行われた時点のコピーを保持するので、このデータを下記のように別な用途に使用することが可能である。

・業務用DBの正確なコピーをテープ等にバックアップすることで、DBサーバーに負荷をかけることなくバックアップを取得できる。

・DBの内容を、EUCなど別な用途に提供する際に、業務用のDBサーバーとは別なDBサーバーを立てて、データの参照を行うことができる。

・SQL Serverの場合、DBの整合性チェックを定期的に行うことが推奨されているが、コピーされたDBに対してDBCCチェックをかけることで、本体DBの整合性チェックを行うことができる。

本プロジェクトにおいても、業務用のDBに対してRVを3つ（RV1、RV2、RV3）持ち、これを下記のような用途に使用している。

・RV1

バッチ処理のうち、処理時間が長大で、翌日のオンライン中に行ってもよいもの（帳票作成など）について、前日のバッチ終了時点でのコピーを別なDBサーバーで参照できるようにしておき、その処理をこのDBを参照するように実行させる（DBはリードオンリーとして使用)。

・RV2

統合バックアップシステムとして、前日のバッチ終了後のバックアップを取得するために使用する。

・RV3

前日のバッチ終了後のDBをエンドユーザーに生のままで参照させる目的で別途DBサーバーを立て、クライアントからこのDBを参照させる（DBはリードオンリーとして使用)。

なお、RV1とRV3は、異なる時点でのコピーとなっており、現在は別々に参照されている。

DBファイルの構成は、図3～図5のように設定した。DB本体データとログファイルを分離するためにボリュームの構成は本体データ用とログファイル用に分割し、本体データ部分はさらに複数の物理ファイルに分割している。

## 5．DBバックアップ関連ジョブの構成

DBのバックアップ関連ジョブは、前述のようなSANの機能を使用するために、図6のようなスケジュールで実行している（バッチジョブの制御はJP1を使用して起動している)。ジョブの構成は次のようなものだ。

・オンライン実行中は、MVだけで処理を行い、バッチ実行開始に向けて、RVを接続する。

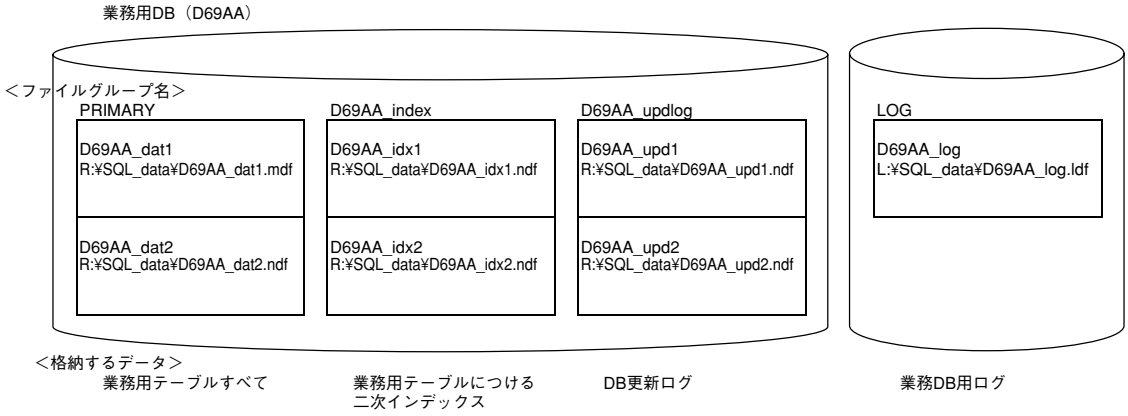


**図3　業務用DBのファイル構成**

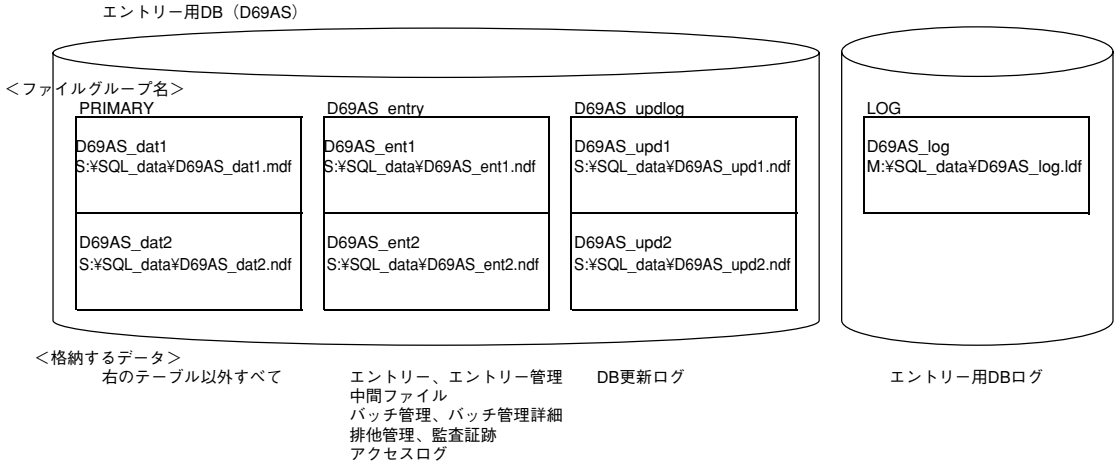


**図4　エントリー用DBのファイル構成**

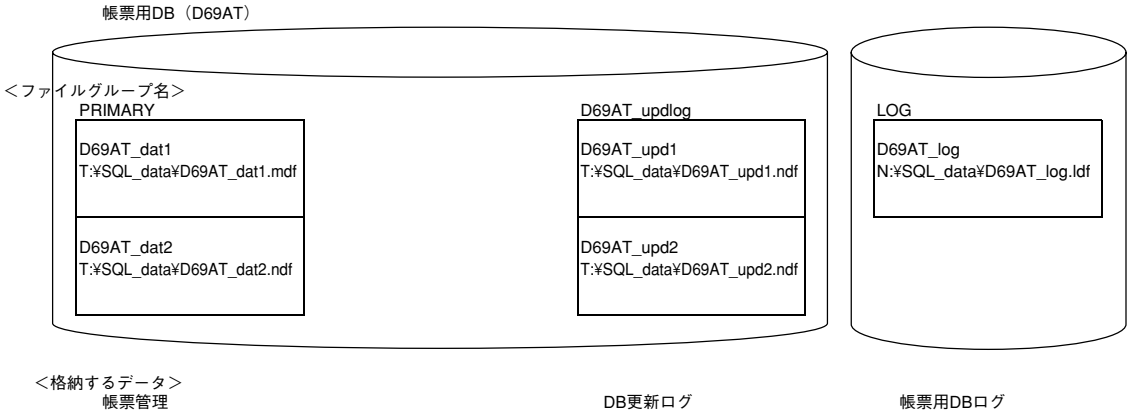


**図5　帳票用DBのファイル構成**

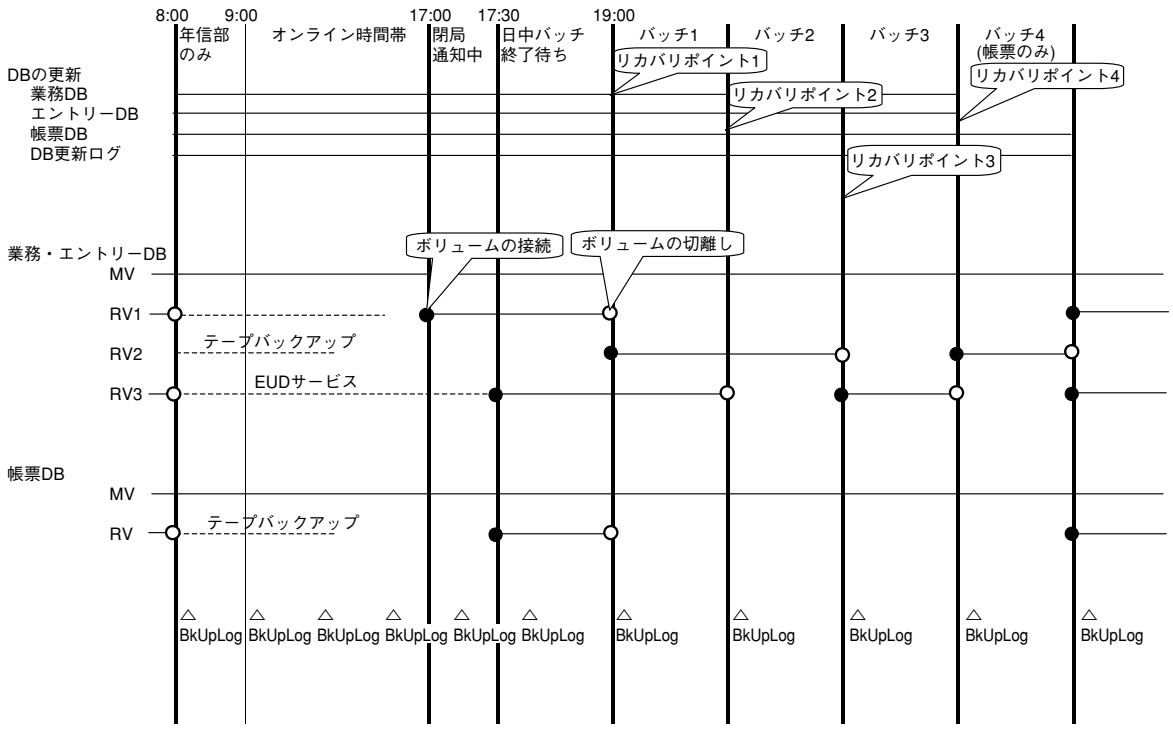


**図6　バックアップスケジュール**

・オンライン実行中の各RV は、5.3 で述べたように他の用途にそれぞれのRVを使用する。
・バッチ実行中は、バッチの処理時間を4つの区間に分割し、そのそれぞれで、区間の先頭に戻すことを可能にするように、最低1つのRVがその区間の先頭時点でのデータを保持する。バッチ処理中に異常発生などがあった場合、該当のRVを使用して、その区間の先頭の状態にDBを戻し、バッチを再実行する。
・オンライン実行中は、30分おきにログバックアップを取得して、オンライン実行中の異常発生に備える。
・バッチ実行中にも、各区間の先頭でログバックアップを取得しているが、これは、ログファイルの連続性を保つためと、ログファイルが増大するのを防ぐためである。

また、バックアップ関連ジョブを組み立てる際に、いくつかの問題点・留意点が存在する。

・SANのユーティリティがDBサーバー上でしか動作しない

ユーティリティプログラムは、DBサーバー上で稼動させなければならず、かつ、ターミナルサービス等でも動作しない。JP1を稼動させるのはバッチ管理サーバー上であり、JP1のエージェントをDBサーバーには入れたくなかったので、バッチ管理サーバーからDBサーバーにジョブをつなぐ必要があった。この連携には、WindowsのRSH機能を使用して実行させている。

・DBデータの静止点を確保する必要がある

SANのスナップショットバックアップを行って、RVを切り離す操作中に、DBのバッファがすべてフラッシュされるので、データの静止点を確保する必要がある、この時点でDBに対するアクセスがあると、切り離されたRVは正確なデータを持たなくなるため、DBに対するアクセスを禁止する必要がある。バッチ処理については、この時点でジョブをスケジュールしないようにすることで対処できるが、オンライン処理については、いつアクセスがあるかわからないため、Web処理のアプリケーション変数に「処理禁止フラグ」を持たせ、この値がONの場合、アプリケーション処理が実行されないようにコントロールする。

・ジョブそのものの制御は、JP1により行う

ジョブの実行スケジュールを管理する方法は、Windowsのタスクとして登録する、SQL Serverのタスクとして登録するなど、いくつかの方法があるが、本プロジェクトでは、全体のジョブ管理ツールとしてJP1を使用しているので、この機能を使用して行う。

SANのバックアップ・リストアは、前後の同期処理がバックグラウンドで行われるので、その時間を含めてバックアップ・リストアのジョブを構成する必要がある。

同期開始後、切り離しを実行するまでに下記の時間が必要となる。

・ボリュームのフル同期の場合、ボリューム全体をコピーする時間
・差分同期の場合、ボリュームの変更された部分をコピーする時間

コピーは、SAN内部のバスを通して行われるので確かに速いが、ボリュームのサイズ（同期を必要とするデータサイズ）が大きければそれだけの時間は必要になる（100GB当たり20分位）。

## 6．SAN活用の評価と設計上の留意点

SANを使って得られたメリットは、

・データ格納領域としての信頼性が向上する。
・SANで使用するディスクは、アクセス速度が速い。
・DBのバックアップ、リストア時間が短縮される。

などである。また設計上の留意点は、

・DBの各ファイルをできるだけ独立したボリュームに配置するように設計が必要。これはSANに特有の話ではないが、SANのボリューム構成を検討する際に十分な配慮が必要である。
・SANのユーティリティプログラムの動作条件・環境を調べ、事前にテストしておく。
・複数のサーバーが連携して処理を実行しなければならない。この連携が問題になる。単純な連携はRSHなどで行えば可能だが、相手側のサーバーが止まっていたとき、連携に失敗したとき、連携は成功しても処理に失敗したときなどには、これらの状況を把握して、必要なら元の状況に戻すことを考えておく。
・SANのオペレーションでは、待ち時間が意外に大きなウェートを占める。DBのスナップショットバックアップを取る際、SQL Serverに対して、メモリー上のデータをディスクに書き込み、ディスク上に残されたデータに不整合が生じないようにしてから切り離す必要がある。この指示を行ってから、実際にボリュームの切り離しを行うまでに一定の待ち時間 （今回の製品では、メーカーの推奨待ち時間は1分） を確保する必要がある。この時間帯にDBに対するアクセスは一切ないようにするが、バッチ処理のスケジュールを組む上でこの待ち時間が大きいので注意する必要がある。

本プロジェクトで、唯一やり残したと考えるのは、DBサーバーにクラスタ（MSCS）を組んだ関係もあり、DBのリストア処理を完全に自動化することができなかった点だ。これは、メーカーの推奨手順として、クラスタの待機系のサーバーをシャットダウンする必要があり、この手順をオペレータが手作業で行うために完全に自動化する手順を確立することができなかったことによる。

今後、同様の環境でシステム構築する機会があれば、ぜひチャレンジしてみたい課題である。