OpenVMS でのバックアップの理論と実践

John Gillings 著
ソフトウェア・システム・コンサルタント、OpenVMS アンバサダ

1.0 はじめに

コンピュータ・システムのバックアップが非常に重要であることは、誰でも知っています。しかし、システムのバックアップを作成するという行為は宗教的儀式のようになっていて、データに対して意味不明な呪文を唱えているようなものかも知れません。テープ・ドライブが回転して、何かが書き込まれていることは確かなのですが、役立つシステム・バックアップが作成されているとは限りません。残念なことに、バックアップ・ストラテジが不適切だったことがわかるのは通常、まさに最もまずいとき、つまり、障害発生の後、システムを復旧しようとするときなのです。

ビジネスがますますコンピュータ・システムに依存するようになるにつれて、消失したデータを復旧できないと、ビジネスに重大な支障をきたすようになります。テープ・システムを売り込む業者は、データ消失から 1 年以内に破産に追い込まれた企業の割合がいかに高いかを示すかも知れません。このようなディザスタ (災害) を防止するには、通常のシステム・メンテナンスにバックアップを統合することが重要です。さらに、バックアップは、データの完全な復元を保証できる方法で実行することも必要です。

2. バックアップとは?

2.1 定義

ここでは、バックアップとは、**将来**システムを**復元**できるように、**システムの状態**に関して**十分な情報**を収集するための**ストラテジ**であると定義します。太字で示した用語は、「**システムの状態を復元**」するという全体的な概念の中で、キーとなる重要な要素です。特に、「ディスクの内容」に関する定義が一切ないことに注意してください。バックアップに関して重大な誤解が生まれているのは、ディスクやテープだけに注目し、「**全体としてのシステム**」を見落とした結果です。ディスク・ドライブの内容は、確かにシステムの状態にとって重要な要素ですが、メイン・メモリの内容や、コントローラのキャッシュなどのキャッシュやバッファの内容も同じように重要です。さらに、メモリやディスクを変更することのできるアプリケーション・プログラムの状態も重要であり、言い換えれば、システムの現在のオペレーションに影響するすべてのものが重要なのです。さらに、この定義では、特定のコマンドやプロシージャよりも、**ストラテジ**を重視していることに注意してください。最後に、バックアップの目標は**十分な情報**の収集であり、必ずしも**すべての情報**が必要ではないことにも注意してください。

2.2 キー・ポイント

次の 3 つのキー・ポイントを考慮することで、信頼性が高く、有効なバックアップ・ストラテジを設定できます。

- システムの状態を復元する
- コマンドではなく、ストラテジを使用する
- 十分な情報を収集する

さらに、システムについてよりよく理解しておけば、バックアップの実行に必要なリソースと作業量を最低限に抑えることができます。

3. バックアップはなぜ必要か?

3.1 現実の世界では、問題発生の可能性があれば問題が発生する…

コンピュータ・システムがいつでも意図したとおりに動作する「理想の世界」では、バックアップは必要ありません。しかし、コンピュータ・システムには「理想の世界」といった概念はなく、マーフィーの法則がコンピュータ業界全体に影響を与えているかのように思えてしまいます。つまり、問題発生の可能性があるものは、そのとおりに発生するのです。コンピュータ・システムを「正常な状態」または「正常でない状態」になるマシンであると考えてみましょう。システムの状態がどうなるかについて考えると、その可能性は膨大な数になり、その大多数は「正常でない状態」です。また、驚くほどさまざまなイベントによって、システムは「正常な状態」から「正常でない状態」に変化します。状態変化の中には、簡単に元の状態に戻すことができるものと、元に戻せないものがあります。適切なバックアップとは、あらゆる状況でシステムを「正常な状態」に復元できるものです。

3.2 どのような問題が発生する?

発生する可能性のある困ったイベントとして、たとえばハードウェア障害、アプリケーション・エラー、オペレーティング・システムのバグ (たまに発生することがあります)、電源障害、火災 (他のビルから延焼するかも知れません)、水害などが考えられます。もちろん人間が引き起こすエラーもあります。たとえば、不適切なディレクトリで「**DELETE *.*;***」と入力してしまったり、「このボタンは何に使うの」と無邪気に質問しながら、押してはいけないボタンを押してしまうなどのエラーが発生しかねません。

3.3 バックアップはデータにかける保険

バックアップ・ストラテジは、一種の保険であると考えてみましょう。機器への投資 (追加のディスク・ドライブ、テープ・ドライブ、テープなど)、保管経費、人件費、CPU 時間などの形で保険の掛け金を支払います。災害が発生した後、システムを復元してもらうことで、保険金を回収することができます。その場合、保険の保障の対象となるイベントと、保障されないイベントを正確に理解しておくことが必要です。

さらに、基本的な要素として、「自己負担金」(賠償請求に必要なコスト) について考慮する必要があり、保険の掛け金と保障レベルの間の「トレードオフ」(得るものに対して過剰投資していないかどうか) についても考えなければなりません。

4. バックアップ・ストラテジの策定

何にでも有効な万能の保険が存在しないのと同様に、あらゆる人にとって役立つ万能のバックアップ・ストラテジもありません。もしも HP Universal Backup for OpenVMS/VAX/Alpha™/Itanium® といった万能のソリューションがあるとすれば、誰もがそのソリューションを実行するでしょうし、ここでこれ以上、バックアップについて語る必要はありません。しかし、バックアップ・ストラテジは個々のサイトやビジネス・モデルに適合するように設定する必要があります。それぞれの状況を把握することで、保障の範囲を最大限に拡大し、掛け金をできるだけ少なくすることのできるバックアップ・ストラテジを構築することができます。

まず、ストラテジを完全に記述することが必要です。日単位、週単位、月単位、年単位で実行している手順について、詳細な説明を記述し、さまざまなレベルの復元手順 (1 つのデータベース、1 つのディスク、システム全体、ディザスタからの完全な復元など) も記述します。バックアップ・ストラテジは、単にディスクからデータをバックアップするためにどのような操作を行うかということにとどまるのではなく、初期のシステム・プランニングから始まる一連のオペレーションの不可欠な部分であると考えなければなりません。

4.1 リスクについての理解

まず第一に、データの価値を現実的に見積もる必要があります。コンピュータ・スタッフはこの質問にほとんど答えることができません。経理担当者や経営者に「このデータが消失したら、データを復旧するのにどれくらいのコストがかかりますか、あるいはこのデータがなくてもビジネスを続行できますか」という質問をしなければなりません。保険の対象となるデータの価値がわかったら、必要な「掛け金」を現実的に判定することができます。また、障害が発生した後、システムの状態を復元するコストを評価するために、データが消失した場合に必要となる時間あたりのコストも把握する必要があります。「自己負担金」を多くすれば (つまり、復旧に必要な時間を長くすれば)、「掛け金」を削減することができます。

システム・オペレーションを担当するスタッフがこの種の保険に関する判断を下すべきではありません。このようなスタッフの役割は、提示された保険プランをもとに適切な判断を下すことができる「保険契約者」についての情報を提供することです。どのような保険を契約するかを判断したら、次にその保険を実現しなければなりません。また、コストを判断するにあたって、外見だけでは正確な判断ができないことにも注意しなければなりません。たとえば、DLT ドライブは導入コストが比較的安いのですが、数時間かけてテープをフィードするためにオペレータに支払う時間外手当は非常に高価になる可能性があります。無人操作が可能な大容量のテープ・ライブラリは、初期投資コストがかさみますが、長期的にみると、トータル・コストははるかに安く、信頼性も高くなります。

最後に、ある人にとってはゴミであるものが、他の人にとって財宝である可能性についても考慮しなければなりません。たとえば、ほとんどのサイトでは、ACCOUNTNG.DAT ファイルが消失しても、それほど深刻な問題にはならないでしょう。しかし、システムで課金処理サービスを実行していて、請求書を作成するためのデータがそのファイルに保存されていた場合は、ファイルの消失によって財務的に大きな痛手をこうむります。

4.2 データについての理解

概念的には、最もシンプルで信頼性の高いバックアップは、システムのあらゆるもののスタンドアロン・スナップショットをとることです。この方式では、すべてのディスクの完全なイメージ・コピーを保存します。ダウンタイムをある程度許容できる場合は、このバックアップ手法が最も高速かつ最も信頼性の高い復旧を可能にします。この方法では、すべてのディスクを保存して、システムをリブートするだけで、必要な操作は完了です。しかし、このストラテジを実際に採用できるケースはほとんどありません。この手法を実際に自動化することはできないので (一部のコンソール管理製品は自動化に近づいていますが)、あまりにも多くの労働力を必要とします。また、最速のテープ・ドライブ・テクノロジを利用したとしても、現在の大容量のディスク・ファームやディスク・ドライブでは、スタンドアロン・スナップショットの作成にはあまりにも長い時間がかかってしまいます。

データについて十分に理解していれば、システムを正常な状態に復元するために収集しなければならない情報量を削減することができます。50 GB のデータベースがあったとしても、毎日、すべてのデータが変更される可能性は非常に低いでしょう。したがって、最新のフル・バックアップをとった後で行われた変更の内容だけを保存するようにすれば、バックアップの実行に必要な時間を大幅に短縮できます。ただし、この方法では、最新のフル・バックアップを復元した後で、その後の変更内容をデータベースに反映させなければならないので、システムの状態を復元するために、多少余分に時間がかかるという問題があります。

その他のアプローチとして、データをクラス別に分類する方法があります。たとえば、すべての読み取り専用データを 1 つの物理ディスクに格納することができる場合は、そのディスクのバックアップは必要ありません。完全に読み取り専用のデータであれば、CD や DVD に保存することができ、念のために複数のコピーを作成しておくことも可能です。このようにして作成した CD は、日常の操作で直接読み取って使用することができ、通常はバックアップ用に保存しておき、緊急時に直接読み取ることも可能です。

4.3 ストラテジのテスト

バックアップ・ストラテジの中で、テストは最も見過ごされてしまいがちです。適切なテストを行っておかないと、緊急時にシステムを正しく復旧できるかどうかを確認することができません。障害が発生した後、バックアップ・ストラテジから致命的な欠陥が見つかったとしたら、それは最悪の状況です。しかし、実際には緊急時にこのような欠陥が見つかることが最も多いのです。

ストラテジをテストするには、システムをシャットダウンしても支障のない週末を利用するか、できるだけ運用システムと同じような構成のテスト用ハードウェアを用意します。復旧はできるだけ現実的なシナリオにそって行ってください。テストの目標は、適切に記述されたディザスタ復旧プランを使用して、運用システムを復旧することです。オペレーションに必要な時間を測定し、問題点を書き留め、そのテスト結果をもとに、プランを見直してください。

実際に障害が発生した場合は、復旧手順の事後検証を行い、実際に有効だった部分と改善が必要な部分を確認します。

5. システムのバックアップの方法

システムのバックアップを作成するには、「システム状態」と呼ばれる瞬間的な状態を記録することが必要です。ここで問題になるのは、どのシステム状態もマイクロ秒単位で変化してしまう可能性があるのに、状態を記録するには数時間かかることもあるということです。スタンドアロン・バックアップの場合のように、システムが変更されないような特殊な状況を実現できない限り、バックアップのターゲットである「既知の状態」の後に発生した状態の変化をバックアップ・メカニズムで無視するようにバックアップ手順を設定する必要があります。この手法を**チェックポインティング**と呼びます。アプリケーション・コードは概念上の線を引き、そのポイントの前に発生した変化だけをバックアップに含むようにします。

チェックポインティングはアプリケーション固有の機能であるため、アプリケーション・ロジックと連携しなければ、汎用システム・ツールでライブ・データのオンライン・バックアップを実行することはできません。しかし、バックアップはシステム関連の機能として認識されることが多く、アプリケーション設計者がバックアップ機能をアプリケーションに統合することはほとんどありません。ところが、実際にはまったくその逆であり、バックアップは基本的にはアプリケーション関連の機能なのです。オペレーティング・システムはバックアップのために情報を収集する支援ツールを提供しなければなりませんが、どの情報を収集するかはアプリケーションで判断しなければなりません。オフライン・バックアップを実行するのに必要なダウンタイムを許容できない場合は、ライブ・バックアップを作成する機能をアプリケーション・コードに組み込む必要があります。このような処理を自動的に実行してくれる「魔法の杖」はありません。

5.1 バックアップに役立つツール

ここでは、バックアップを実行するときに役立つツールを紹介します。

5.1.1 データベース管理システム

Oracle の Oracle/RDB や Ingres などのデータベース・システムにはチェックポイントをサポートする機能が組み込まれており、トランザクション・レベルでアプリケーションと連携します。アプリケーションでチェックポイントを宣言する必要がありますが、面倒な処理の大部分は DBMS が実行します。詳細については、本書の範囲をこえているので、ここでは説明しません。

5.1.2 RMS Journaling

アプリケーションで RMS ファイルを使用する場合は、バックアップの支援に RMS Journaling を利用できます。しかし、アプリケーション・コードで明示的または暗黙にチェックポイントを宣言する必要

があります。RMS Journaling の機能と使い方については、RMS Journaling のマニュアルを参照してください。

5.1.3 Backup ユーティリティ

BACKUP はバックアップを実行するのにとても便利なツールですが、実際にはある場所から別の場所にデータ・ビットを移動するだけのツールに過ぎません。魔法のような機能が組み込まれているわけではなく、単に指示に従ってファイルをコピーするだけです。適切なファイルをコピーしていれば、非常に役立つのですが、そうでないと、ゴミの山をコピーしてしまうことになりかねません。BACKUP ユーティリティに関するドキュメントを注意深く参照して、すべての機能を理解するようにしてください。

5.1.4 DIRECTORY コマンド

DCL の DIRECTORY コマンドは、バックアップを実行するためのツールではありませんが、データの流動性を分析するのに役立ちます。たとえば、次のような簡単なコマンドを実行するだけで、特定のディスクでデータがどの程度変更されているかを把握することができます。

DIRECTORY/MODIFIED/SINCE=date disk:[000000...]

5.1.5 スペア・ディスク

安くて大容量の SCSI ディスクが提供されるようになったため、スペア・ディスクをいくつか用意するのもバックアップ・ストラテジでコスト効果の優れた方法です。スペア・ディスクを用意しておけば、情報の保存でも復元でも、柔軟性を大幅に向上できます。36 GB の SCSI ディスクを導入することで、操作に必要な時間をどれだけ削減できるか計算してみてください。また、技術者が交換用ディスクを届けるのに必要な時間も削減できます。故障したディスクをただちに交換することができれば、障害から復旧するのに必要な時間を大幅に短縮でき、その効果はドライブの導入にかかるコストを大幅に上回ります。

5.1.6 適切なハードウェア

テープ・ドライブの小型化、高速化、大容量化は日進月歩で進んでいます。最新のテクノロジを導入することが常に必要なわけではないかも知れませんが、必要とされる負荷を処理できる能力をドライブが確実に備えていることは非常に重要です。

後からの思いつきでテープ・ドライブがシステムに追加されているケースをあまりに多く見かけます。システムを設計する場合、多くのディスク・ドライブが搭載されているストレージ・シェルフに膨大なデータが格納されていることを考えると、フル・バックアップに必要な時間は計り知れません。現在入手可能な最速のテープ・ドライブを使用したとしても、フル・バックアップには数日かかる可能性があります。システム・コストを計算する際には、バックアップのコストも忘れずに盛り込んでください。

5.1.7 Archive/Backup System for OpenVMS (ABS)

大規模なシステムの場合は、ABS を使用するとバックアップ・ストラテジの自動化、テープ・デバイスの制御、メディアの管理に役立ちます。ABS は各ファイルが格納されているテープを迅速に探すことができます。

5.2 役立たない手法

この記事を書こうと思った最大の動機は、ディスクの有効なオンライン・バックアップを作成できると一般に考えられている手法が、実は役に立たないことを明らかにすることでした。確かに、これらの手法はディスクの有効なコピーを作成できる場合もありますが、必ず作成できるわけではないのです。このような手法を採用することは、ちょうど掛け金の安い保険を契約するようなものです。このような保険契約では、保険金を請求しても、契約が履行されない可能性があります。そんな保険にビ

ジネスを託すことができるでしょうか。ここでは、一般的に使用されているバックアップ手法で、実際には役立たない手法について説明します。

5.2.1 /IGNORE=INTERLOCK 修飾子の使用

/IGNORE=INTERLOCK 修飾子の目的は、「ファイル・アクセスの競合」の重大度を ERROR から WARNING に変更することです。その名前が示すように、/IGNORE=INTERLOCK 修飾子を使用すると、システムは、ファイルの整合性を保護するファイル・システムのロックを回避します。ACCONFLICT 警告が発生した状態でバックアップされたファイルの場合、確実なことは、そのファイルが存在することと、そのファイルが、バックアップに含まれるファイルとほぼ同じサイズであるということだけです。使用可能なファイルのコピーは作成されますが、そのファイルにはゴミしか入っていないかも知れません。

/IGNORE=INTERLOCK の使用に関して皮肉なことは、警告が発生するファイルほどバックアップが必要なファイルであり、活発に変更されているファイルであるということです。

5.2.2 シャドウ・セットの解除

もう 1 つの役立たないバックアップ手法は、シャドウ・セットの物理的なメンバーをシャドウ・セットから切り離して、そのイメージ・バックアップを作成した後、再びシャドウ・セットに戻すという方法です。この手法は、次の 2 つの理由から非常にお粗末です。

- 第一に、/IGNORE=INTERLOCK コマンド修飾子がバックアップにとって信頼性のない機能であるのと同じ理由で、シャドウ・セットが解除された時点でオープンされていたファイルは、その整合性が維持されるという保証がありません。OpenVMS バージョン 7.3 (または最新のシャドウイング ECO を装備したバージョン 7.2) 以降では、シャドウ・セットのメンバーをディスマウントすると、ディスク構造レベルでは整合性が保証されますが、その時点でオープンされていたファイルの整合性は必ずしも保証されません。OpenVMS バージョン 7.3 より前のバージョンでは、ディスマウントされたシャドウ・セット・メンバーの内容が正確であるという保証はありません。
- 第二に、シャドウ・セットをシングル・メンバーに縮小すると、両方のディスクに潜在的な不良ブロックが含まれるという危険性が発生します。常にシャドウ・セットに 2 つのメンバーを保持するようにすれば、同じ LBN が同時に両方のメンバーで不良になったときにだけ、不良ブロックの影響が発生します。しかし、このような状況が発生する可能性はほとんどありません。不良ブロックは、そのブロックが読み取られるときに初めて検出されるので、実際に読み取られるまで、長期にわたって検出されない可能性があります。不良ブロックが読み取られ、検出されると、シャドウイング・ソフトウェアはもう一方のメンバーからデータを復元しようとします。しかし、シャドウ・セットが解除されていると、復元可能なバックアップ・コピーがもはや存在しないので、不良ブロックが「顕在化」します。同様に、BACKUP ユーティリティは削除されたメンバーの不良ブロックをコピーし、不良ブロックとしてマークします。バックアップを復元するときに、これらの不良ブロックは顕在化します。

5.2.3 シングル・メンバー・シャドウ・セットの設定

もう 1 つの役立たない手法は、すべてのディスクをシングル・メンバー・シャドウ・セットとして設定する方法です。この手法は、ディスクでエラーが発生し始めたら、第 2 のメンバーをシャドウ・セットに追加して、コピーすることができるようにすればよいという理論のもとに考えられた方法です。そして、コピーが終了したら、欠陥のあるディスクは取り除けばよいという考え方です。このロジックの問題点を指摘することは難しいのですが、この手法が機能する可能性はほとんど皆無です。まず、シャドウイング・ソフトウェアは、正常なデータをコピーするときと同じ方法で、壊れたデータもコピーします。さらに、故障しかけているディスクに対して大量の I/O を実行することは望ましくありません。ディスクがかなり深刻な状態になっている場合は、シャドウイング・ソフトウェアはシャドウ・

セットからそのディスクを除外する前に、完全なコピーを作成することができません。このようなシナリオの結末を想像すると、恐ろしいものがあります。

シングル・メンバーのシャドウ・セットは、メンバーを追加して、その内容をコピーした後、再び解除することにより、バックアップを作成する目的で使用されることがあります。しかし、すでに述べたように、この手順で作成された結果は信頼できません。

シングル・メンバーのシャドウ・セットを使用することは、ほとんど無意味です。多くの人が痛感しているように、シャドウイング・ソフトウェアは実に複雑で、さまざまな問題を引き起こす可能性があります。シングル・メンバーのシャドウ・セットを利用しても、シャドウイングの恩恵はまったく得られず、ソフトウェアの複雑さから発生するリスクにさらされるだけです。物理的なスピンドルはおそらくシャドウ・セットの中で最も安価なものであるため、2 つのメンバーで構成されるシャドウ・セットを構築する方が確実に優れています。

5.2.4 適切なバックアップの代用としての RAID

RAID コントローラを利用すると、ストレージ管理が単純になり、柔軟性も大幅に向上します。さまざまな種類の RAID セットによって、パフォーマンスとデータ冗長性のいずれか一方あるいは両方を向上することができますが、だからといって適切なバックアップが不要になるわけではありません。しかし、バックアップ・ストラテジの中で RAID セットの機能をうまく活用して、バックアップの高速化とコスト削減を目指すことは必要です。

5.3 役立つ手法

これまでは悲観的な話ばかりしてきました。しかし、ライブ・データ (あるいはほとんどライブに近いデータ) のバックアップを安全に実行する方法もあります。最悪でも、ダウンタイムを数分以内まで短縮することができます。ここでは、そのような役立つ手法を紹介します。

5.3.1 単純な索引付きファイルに対する CONVERT/SHARE の使用

単純な索引付きファイルでは、レコードは相互に依存せず、他のファイルのレコードにも依存しません。レコードの更新情報は、保存されるか、保存されないかのいずれかである (つまり、部分的な更新情報が保存されることはない) という制約を容認できるなら、CONVERT/SHARE コマンドを使用して、オープンされている RMS 索引付きファイルの正確なコピーを作成できます。この手法では、アプリケーション・コードが共有アクセスのためにファイルをオープンすることを前提にしています。SYSUAF.DAT ファイルについて考えてみましょう。CONVERT 操作の実行中に、システム・マネージャがユーザ・パスワードをリセットすると、CONVERT 操作が実行されるタイミングに応じて、バックアップ・コピーは特定の変更を反映するか、反映しないかのいずれかになります。RMS のルールでは、特定の変更情報がコピーされなかった場合、それ以降の変更情報もコピーされません。

次に、新規ユーザの追加という、もう少し複雑な場合について考えてみましょう。この場合、レコードを SYSUAF.DAT ファイルに追加し、「論理的にリンク」されたレコードを RIGHTSLIST.DAT ファイルに追加する処理が必要です。そのとき、ジョブで両方のファイルに対して CONVERT を実行すると、CONVERT 操作が実行される順序に応じて、両方のファイルが変更されるか、どちらのファイルも変更されないか、いずれか一方のファイルだけが変更されるかの、いずれかになります。この状況では、バックアップ・プロシージャが両方のファイルの変更をアトミックに取り扱うという保証がないため、全体としての「データベース」(2 つのファイルで構成) は矛盾した状態になる可能性があります。データベースのタイプによっては、この動作を許容できないこともあります。この特別なケースでは、最初に RIGHTSLIST.DAT ファイルのバックアップを作成し、次に SYSUAF.DAT ファイルのバックアップを作成することができます。そのようにすれば、両方のファイルの更新情報をバックアップに盛り込むか、どちらのファイルの更新情報も盛り込まないか、あるいは UAF エントリの更新情報だけを盛り込むことができます (RIGHTSLIST.DAT ファイルのエントリの更新情報だけをバックアップに盛り込むことはできません)。MCR AUTHORIZE ADD/IDENTIFIER/USER=* コマンドを使用す

ることで、ユーザ名ごとにライト識別子が有効であるかどうかを確認するプロシージャを簡単に自動化することができます。したがって、CONVERT/SHARE コマンドは、ユーザ登録ファイルをオンラインでバックアップするための許容できる方法として使用できます。

5.3.2 シャドウイングの正しい使い方

シャドウイングを安全に使用して、ほとんどオンラインでバックアップを行う方法もあります。この手法では、アプリケーションの連携はそれほど必要とされません。手順は次のとおりです。

1. 2 メンバーのシャドウ・セットにスペア・ディスクを追加します。
2. 完全なシャドウ・コピーの実行を許可します。
3. このディスクを使用するすべてのアプリケーションをシャットダウンします (すべてのファイルをクローズします)。
4. 3 番目のメンバーをディスマウントします。
5. アプリケーションを再起動します。
6. 都合のよいときにスペア・ディスクのバックアップを作成します。

この手法では、コピーが完了した直後に、新しい各メンバーから不良ブロックがなくなることが保証されるため、不良ブロックに関する問題を回避できます。また、もともとシャドウ・セットに含まれていた 2 つのメンバーのいずれかに、各ブロックの正常なコピーが少なくとも 1 つは存在することになります。アプリケーション・コードをシャットダウンしているので (すべてのファイルをクローズ)、ファイルはすべて整合性のある状態に維持されます。

アプリケーションによっては、トータル・ダウンタイムがわずか数分ですむこともあります。この手法では、スペア・ディスク (前述のとおり安価で便利) と、シャドウのコピーを実行する時間がコストとして必要です

5.4 特殊なケース -- システム・ディスクとシステム・ファイル

システム・ディスクは特殊なケースとして考えなければなりません。まず、システム・ディスクのフル・バックアップを作成するには、/IMAGE 修飾子を指定してスタンドアロン・バックアップを実行しなければなりません。この手法について、考慮の余地はありません。次に、ディスク上の情報の大部分は読み取り専用であり、配布メディアから簡単に復元できます。最後に、システム・ディスクに格納されているファイルのうち、変化する可能性のあるほとんどすべての重要なファイルは常にオープンされています。たとえば、SYSUAF.DAT、RIGHTSLIST.DAT、QMAN$*、その他の「クラスタ・パーソナリティ」ファイル、さまざまなログ・ファイルやジャーナル・ファイルなどです。つまり、夜間に定期的に $BACKUP/IMAGE/IGNORE=INTERLOCK SYS$SYSDEVICE: tape:SYSTEM.BCK/SAVE を実行しているとすれば、その処理は基本的に時間とテープの両方を無駄にしているだけです。バックアップできているファイルはバックアップの不要なファイルであり (HELPLIB.HLB のコピーをいくつも作成することはまったく無駄です)、バックアップの必要なファイルはバックアップされていないのです。クラスタ・ファイルについても、その格納場所とは無関係に、同じことが言えます。

この問題についても、万能の解決策はありません。しかし、システム・ディスクのバックアップ・ストラテジは次のように設定することができます。

- アップグレードの直前と直後に、イメージ・バックアップを実行します (アップグレードの後、イメージを復元して、システムの状態を整理することもできます)。
- 頻繁に変更されるファイルは、夜間に CONVERT/SHARE コマンドを使用して、別のディスクのディレクトリにコピーします。このような処理が必要なファイルとしては、SYSUAF.DAT、RIGHTSLIST.DAT、VMSMAIL_PROFILE.DATA、VMS$PASSWORD_HISTORY.DATA などがあります。

- データが格納されているディレクトリのバックアップを他のディスクに作成します。このディレクトリは、必要に応じてただちに使用できる「データのホット・スタンバイ」コピーであると考えることもできます。
- 毎週あるいは必要に応じて CONVERT/SHARE を使用して、それほど頻繁には変更されないファイルも同じ場所にコピーします。このような処理が必要なファイルとしては、SYSALF.DAT、NETPROXY.DAT 、 NETOBJECT.DAT 、 NETNODE*.DAT 、 LMF$LICENSE.LDB 、 VMS$AUDIT_SERVER.DAT などがあります。
- ACCOUNTNG.DAT、SECURITY_AUDIT.AUDIT$JOURNAL、OPERATOR.LOG などのログ・ファイルは、再起動して、毎週または 1 週おきにアーカイブします。
- 必要に応じて、SYS$MANAGER ディレクトリのコマンド・プロシージャなど、変更される可能性のある他のファイルをバックアップします。
- 最新のイメージ・バックアップを適用し、ミニマム・ブートを実行し、変更されたファイルや随時行われた変更内容を復元することで、システムを復元します。この手順は、ユーザ定義 SYSGEN パラメータ (たとえば USERD1) を使用して、スタートアップ・プロシージャで自動化することも可能です。

ここに示した一連の手順は、必要なすべての操作を網羅しているわけではありません。システム・ディスクを分析して、どのファイルが変更されるのかを特定し、どのファイルのコピーが必要かを判断する必要があります。

5.5 キューについて

キュー・マネージャ・データベースのバックアップは実に厄介です。まず、STOP/QUEUE/MANAGER/CLUSTER コマンドを使用して、キュー・マネージャ・プロセスを停止する必要があります (**警告**: このコマンドは実行中のバッチ・ジョブを停止します)。これで、QMAN$MASTER ディレクトリ内の SYS$QUEUE_MANAGER.* ファイルに対して、COPY コマンドや BACKUP コマンドを使用できるようになります (しかし、キュー・マネージャが実行されている間、これらのファイルをバックアップすることはできません)。ファイルを復元する場合は、キュー・マネージャを起動する前に、SYS$QUEUE_MANAGER.QMAN$JOURNAL の名前をバージョン 1 に変更してください。このように変更しておかないと、このファイルは無視されてしまいます。

また、BACKUP コマンドを使用してこれらのファイルのバックアップを作成することは、一般に推奨できません。では、キュー・マネージャ・データベースが消失しないように保護するには、どのような措置を講じればよいでしょうか。SHOW QUEUE/ALL/FULL コマンドの出力を使用すれば、何もない状態からキューを作成するためのコマンド・プロシージャを簡単に作成できます。この例で示したバックアップは、バックアップの対象になる情報の物理的なコピーではなく、むしろ論理的なコピーです。

同様に、LMF$LICENSE.LDB ファイルのコピーを作成し、以下のコマンドを使用して LICENSE REGISTER コマンドのリストを作成することで、ライセンス・データベースのバックアップを作成することもできます。

```
LICENSE ISSUE/PROCEDURE/DATABASE=COPY_OF_LICENSE.LDB
*/ALL/OUTPUT=file
```

(**警告**: このコマンドは、すべての PAK を ISSUED 状態にしてロードされないようにするので、ライセンス・データベースのコピーに対してだけ実行してください。)

この結果、キュー・エントリだけが残されます。キュー・エントリは 3 つのグループに分類できます。最初のグループは、バックアップを開始しようとしたときに、キューに登録されていた一時的なジョブです。このようなジョブは、将来の任意の時点ではおそらくすでに処理が完了しているため、再起動されることはありません。このようなジョブを再実行したり、再び印刷した結果がどのようになるかは予測できません。

第 2 のグループは、常にスケジューリングまたは実行しておかなければならない再帰的な、セルフ・リサブミット・ジョブです。このようなジョブは再起動が必要ですが、おそらくシステムまたはアプリケーションのスタートアップ・プロシージャで取り扱うことができます。アプリケーションを起動するときに、関連するジョブが正しくスケジューリングされているかどうかを確認します。スケジューリングされていない場合は、再びサブミットします。この手法では、システム・クラッシュや電源障害、データの消失などによって発生する問題を回避できます。この他にも、スケジューラ製品を利用して、このようなジョブを制御する方法も考えられます。

第 3 のグループは、スケジューリングされていたものの、キュー・マネージャ・データベースが消失した時点でまだ実行されていなかった臨時的なジョブです。このグループのジョブに対する単純な解決策はありませんが、このようなジョブが発生することは非常に稀です。キュー・マネージャ・データベースが消失する可能性も非常に低いことと合わせて考えると、このように発生することがきわめて稀なイベントに対応するために、バックアップ・ストラテジを複雑にすることは合理的ではありません。必要に応じて、SHOW QUEUE/ALL/FULL コマンドの出力を利用して、キューとジョブの両方を再作成することができます。

6. それでも 24 x 7 x 365 体制でのオペレーションが必要な場合は?

休むことなく 24 x 7 x 365 体制でオペレーションが実行されていて、ダウンタイムをまったく許容できない場合は、バックアップ・ストラテジが実際に機能するかどうか、特に注意を払って確認する必要があります。妥当なパフォーマンスを備えたオペレーティング・システムで、このような状況に対応できるバックアップ機能を提供するシステムはありません。したがって、通常の処理と並行して、バックアップと復元を実行できるように、アプリケーション・コードを作成することが必須です。魔法の杖などありません。しかし、アプリケーションを注意深く設計し、チェックポイントとジャーナリング機能を効果的に活用し、データベースを適切に設計しておけば、24 x 7 x 365 体制のオペレーションを実現できない理由はありません。ただ、このようなオペレーションが自動的に実現されるとか、コストを伴わずに実現されるなどといった期待は禁物です。

7. まとめ

バックアップの目標について考慮し、現在のバックアップ・ストラテジがこれらの目標を達成しているかどうかを検討するにあたって、これまでの説明が参考になれば幸いです。この他にも、次のような質問を自分自身に問いかけてください。

現実のリスクについて理解しているか。
データについて理解しているか。
有効な復旧プランを作成しているか。
プランをテストしているか。
適切なツールとハードウェアを使用しているか。
適切なファイルをバックアップしているか。
BACKUP コマンドは実際に機能しているか。
必要でないものまでバックアップしていないか。
システム・ディスクとシステム・ファイルをどのように取り扱っているか。
アプリケーションにバックアップ機能が組み込まれているか。

詳細については、最寄のカスタマ・サポート・センターにお問い合わせください。