



【参考資料】

MOCHAプロジェクトで実施されたシナリオの概要

MOCHAプロジェクトで行われた検証内容をもう少し具体的に理解いただくために、今回のプロジェクトで採用された検証シナリオの概要をここに掲載する。なお詳細については、前ページで紹介したマイクロソフトのホームページにアクセスして欲しい。

1. Time Finderによるオンラインスプリットと同期

TimeFinderを利用することで、SAPが使用しているデータベース内容を他のシステムで利用したり、ディスク to ディスクの迅速なリカバリなどが可能になる。ここでは複数のシナリオを用いることで、TimeFinderの基本動作やBCVを利用したデータリカバリ等の動作検証が行われた。

1-1.

TimeFinderの基本動作

まず最初に行われたのは、TimeFinderの基本動作の検証である。BCVの切り離しと他システムへのマウント、そして再びメインボリューム(STD)に対して同期させるという一連の動作が、下記の手順で検証された。

*

(1) まず最初にES7000のパーティション1で使用するデータベース(STD)と、そのスナップショットバックアップ(BCV)となるボリュームを作成し、TimeFinderのコマンドで両者を同期させる。

(2) パーティション1上でTimeFinderコマンドを発行し、Stateが Synchronized になっていること(STDとBCVが完全に同期していること)を確認する。

(3) パーティション1上でTSIM 1のコマンドを発行し、スナップショットバックアップを取得してBCVを切り離す。

(4) ES7000のパーティション2上でTSIMのコマンドを発行し、切り離されたBCVをマウント、取得したスナップショットバックアップをパーティション2のSQL Sever 2000にリストアする。

(5) パーティション2上のSQL Server 2000でデータベースをデタッチする。(MMC上で対象のデータベースを選択、右クリックで“ Detach database ”)

(6) BCVをパーティション2からアンマウント。

(7) パーティション1でTimeFinderの同期コマンドを発行し、STDとBCVを再び同期させる。

1-2.

STDに対するBCVからのリストア-1

スプリット後にSTD上のデータベースに障害が発生した場合、BCV上のスナップショットバックアップからリストアを行うことができる。ここではBCVから直接データベースをリカバリする方法を、下記の手順で検証した。

*

(1) 1-1の(1)~(4)と同様の手順で、BCVをパーティション2にマウントする。

(2) 1-1 (5)~(6)と同様の手順で、BCVをパーティション2からアンマウントし、デー

タベースをデタッチする。

(3) パーティション1上でTimeFinderコマンドを発行し、ディスクキャッシュを解放する。

(4) パーティション1上でTSIMコマンドを発行し、スナップショットバックアップをリストアする。

(5) リストアの完了を確認した上でデータベースを書き込み可能にし、SQL Server Enterprise Manager で対象データベース(MR3)をOfflineにしてQuery Analyzerでrestoreコマンドを実行する。

(6) SQL Server Enterprise Managerより対象データベース(MR3)をOnlineにする。

(7) SAPを起動する。(MMCで対象のインスタンスを選択して右クリック Start)

1-3.

STDに対するBCVからのリストア-2

1-2ではBCVからのリストアにTSIMの機能を利用したが、TSIMを利用しない方法もある。ここではスナップショットバックアップにログ適用を行い、その後にTimeFinderでBCVからのリストアを行う方法を、下記の手順で検証した。

*

(1) 1-1の(1)~(4)と同様の手順で、BCVをパーティション2にマウントする。

(2) パーティション1で取得したログバックアップファイルをパーティション2にコピーし、パーティション2上でこのランザク

MOCHA Project

ションログを適用する。

- (3) パーティション2上でデータベースをデタッチし、BCVをアンマウントする。
- (4) パーティション1上のデータベースをデタッチし、アンマウントする。
- (5) TimeFinderでリストアを実行する。
- (6) STDをパーティション1にマウント、データベースをアタッチ、SAPを起動。

なおここでは紙面の都合上、すべてのシナリオを掲載することはできなかった。TimeFinderの機能に関しては、リカバリをかけてからログ適用を行うケースや、SQL Server 2000へのアタッチを手動で行うケースなど、他にもいくつかのシナリオが検証されている。

TSIM : TimeFinder - SQL Server Integration Module。SQL Server 2000上でTimeFinderを利用可能にする、EMCが提供するモジュールのこと。

いずれかのインスタンスで負荷の高い処理が実行され、他のインスタンスの負荷が軽い場合、負荷の軽いインスタンスに割り当てられているCPUを負荷の高いインスタンスに振り向けることができる。この際にシステムを停止する必要はなく、割り当ての変更は即座にシステムに反映される。ここではIDES上で給与計算シミュレーションを行うことで高負荷状態を作り出し、CPU配分の動的な変更を行うという検証を、下記の手順で実施した。

＊

- (1) 「Payroll Program」をひとつのIDESインスタンス(ID1)上で、複数同時実行する。
- (2) プロセス制御でID1のプロセスエイリアス名の「プロセス実行の規則」プロパティを開き、プロセッサ関係のタブ上で、他のインスタンスに割り当てているCPUをID1に割り当てる。
- (3) ID1以外のプロセスの「プロセス実行の規則」プロパティを開き、先程ID1に割り当てた分のCPU割り当てを解除する。
- (4) タスクマネージャーで、新たに割り当てられたCPUに対してID1の負荷が分散されるのを確認する。

なおこのようなマルチインスタンス環境では、OSコレクタはすべてのインスタンスにおいて共通である。単一のSAPOSCOLサービスを起動することで、すべてのインスタンスからOSコレクタの収集する情報にアクセスできる。

2. SAPマルチインスタンス環境の構築とその利用

Windows 2000 Datacenter Server及びES7000は最大32CPUまでサポートしており、単一OS上に複数のSAPインスタンスを構築できる。ここでは下記の手順で単一OS/マルチインスタンス環境を実際に構築した上で、CPU資源の配分を動的に行えることを検証した。

2-1.

マルチインスタンス環境の構築

まず最初に下記の手順で、SAPマルチインスタンス環境の構築を行った。SAPインスタンスの構成としては、「mySAP.com BW 2.1C」×1、「IDES 4.6B」×2の、合計3インスタンスとした。

＊

- (1) マルチインスタンスのインストール。SAP標準手順に従い、複数のSAPインスタ

ンスのインストールを行う。対応するSQL Server 2000は、各SAPインスタンスに明示的に対応するよう、SAPインスタンス名に対応した名前つきインスタンスとしてインストールする。

- (2) プロセス制御の設定。Windows 2000 Datacenter Serverの管理ツール、プロセス制御を用いて、特定のプロセスに対して使用するCPUを割り当てる。割り当て方法としては、特定のディレクトリ以下にあるプログラム群を「プロセスエイリアス」としてグループ化して実行ルールを規定する方法と、各プログラムのインスタンス毎に実行ルールを規定する方法がある。今回は前者の方法を適用した。

2-2.

CPUの動的配分