

Using the IBM Red Brick Warehouse Backup and Restore System

イントロダクション

Red Brick Warehouse の 6.20 では Table Management Utility (TMU)でフルバックアップとリストアシステムをインプリメントしました。TMU はオンラインバックアップとチェックポイント、フルバックアップと差分バックアップ、そして自動データリストアをサポートしました。

バックアップデータは、ディスクファイルから、あるいは X/Open の XBSA API をサポートしているストレージ管理システムから、直接バックアップすることが可能となります。

このデモンストレーションでは、バックアップとリストアの機能を確認した後に、そのコンフィグレーションの方法とケーススタディーで、バックアップとリストアシステムを実際使用します。そして、幾つかの問題点とその解決方法について記述します。また最後に、ストレージマネージャを介してのバックアップとリストアの方法についても紹介します。ストレージマネージャを介してのバックアップとリストアに関しては、2つのプロダクト(TivoliR Storage Manager、Legato Networker.)を使用します。

概要

バックアップレベル

TMU は3つのバックアップレベルをサポートします。

- ・Level 0:フルバックアップ:全てのデータベースオブジェクトのバックアップ
- ・Level 1:差分バックアップ:レベル0からの差分バックアップ.
- ・Level 2:差分バックアップ:レベル0, 1, 2を含む前回からのバックアップ

TMU は外部フルバックアップもサポートします。フルバックアップは Red Brick server の外で実行し、外部のバックアップツールまたはOSのユーティリティプログラムを使用します。

バックアップモード

TMU にはオンラインとチェックポイントという2つのバックアップモードがあります。オンラインバックアップはデータベースへの読み込み時、あるいは書き込み処理中にもバックアップが可能です。チェックポイントバックアップはデータベースの読み込み時、バックアップが可能です。書き込み時のバックアップはサポートされていません。

オンラインバックアップは、バックアップスケジュールを自由に設定でき、データベースのダウンタイムを必要としません。チェックポイントバックアップはデータをリストアする場合に要求される、整合性のあるバックアップイメージの取得を可能とします。

リストアオペレーション

TMU は、幾つかのリストアタイプをサポートします。

- ・データベースリストアはデータベースの全てのオブジェクトをリストアします。
- ・部分リストアは特定のセグメントまたは PSU 単位でリストアします。
- ・コールドリストアはデータベース回復のためのフルリストアを行い、オンラインでは使用できません。

・外部回復リストアは外部バックアップからリストアをします。

「IBM Red Brick Warehouse Table Management Utility Reference Guide」は TMU によるバックアップやリストアの詳細について記述をしています。その内容は、特定のケーススタディーにスコープしており、幾つかのパフォーマンスデータ、成功の為のポイントとヒント、大規模ウェアハウスへのリストアについて記述しています。

スタートの前に

TMU バックアップを始める前に、まず最初に行うのが、バックアップ戦略を確定させることです。データベース・コンフィグレーションとバックアップ/リストア用のソリューションから、バックアップを行うメディアを選択します。

バックアップ戦略の確定

問題を理解するにあたり、サンプルのケーススタディーを見てみましょう。

データベースコンフィグレーションの例:

- ・データベースのトータルサイズは約 530GB (198 セグメントと 566PSU) です。
- ・データベースは、デイリー(月曜～日曜)で AM5:00 から PM11:00 までの時間で、バージョニングをオンにしてデータ更新をします。
- ・日次のデータベース更新処理は、AM12:00 から AM4:00 までの間に行いません。
- ・多くの日次更新処理は同じセグメントグループに影響を与えます。

毎日に 15%以上のデータ(~80GB)が更新され、1週間を通して 20%以上のデータ(~106GB)が更新されません。

バックアップ/リストアソリューションの要求:

- ・データ更新処理の時間中(日次の 5:00 から 11:00)は、データベースをダウンさせない。
- ・深夜の(12:00 から 4:00)忙しくない時間帯は、データベースのクエリパフォーマンスは考慮しない。
- ・データベースのバックアップはデイリーで実行され、そのバックアップイメージは障害時に復旧される。データベースの復旧が必要な場合、リストア時間として、24 時間を超えない。

バックアップ戦略:

データベースのコンフィグレーション、バックアップリストアの要求から、以下のバックアップ戦略が確定されます:

- ・レベル 0 のバックアップスケジュールは、毎週月曜日午前 0 時にスタートします。530GB のデータを、レベル 0 でバックアップするには、想定した以上の時間がかかるためと、データベースのダウンタイムを避けるため、オンラインバックアップを用いることにします。
- ・差分バックアップは日次で実行します。差分バックアップはデータベースへの修正のみを捕らえたものです。時間の短縮とディスクの消費を考慮しました。ここでは日次のバックアップスケジュールも含めています。

レベル2のバックアップはデータベースの更新がない時間に、毎夜(月～金を除く;以下を参照)AM12:00 からスタートします。チェックポイントバックアップはリストアポイントを保証します。この日次バックアップ処理は 1 日分のデータベースの変更を取得します。

レベル1のバックアップは毎週金曜日のAM12:00からスタートします。このバックアップは月曜日のレベル0バックアップからの変更を取得します。レベル1バックアップの目的は、データベースリストア時間の短縮です。

バックアップメディアの選択

ビジネスニーズにもよりますが、ディスクまたはテープデバイスへ直接バックアップが可能です。例えば Informix StorageManager, Legato, 又は Tivoli の Storage Manager などを使用して行います。

ディスク又はテープデバイスへのバックアップは、オペレーションが簡易で、コストも小さく、小中規模のデータベースでの使用が可能です。大規模なデータベースに関しては、Storage Manager の利用を考えます。Storage Manager は、バックアップメディアを管理し、データ圧縮等の特徴を提供します。ただフェイルオーバー機能については、TMU のサポート対象ではありません。

この例では、容易なディスクへのバックアップを選択しています。

コンフィグレーション:

バックアップ戦略を立てた後は、バックアップメディアを選択します。TMU バックアップノリストシステムのコンフィグレーション定義は簡単です。以下のステップを参照してください。

・バックアップセグメントの作成:

最初は以下の計算式によりバックアップセグメントの容量を計算します。

(データベースは 198 のセグメントと 566 の PSU から成り立っています)

$$\text{Size} = ((2 * 198) + (566 * 10) + 1) * 8 = 48456 \text{ KB} = 48\text{MB}$$

バックアップセグメントは以下のステートメントを使用します:

```
RISQL>create segment backup_seg
storage 'bkseg.dat'
maxsize 48M initsize 1024;
RISQL>alter database create backup data in backup_seg;
```

・ディレクトリ \$RB_CONFIG/bar_metadata が存在するかを確認します。このディレクトリはサーバインストール時に作成されます。バックアップメディアとして、ディスクファイルを使用します。更なるコンフィグレーションは必要ありません。

・もし、テープデバイスまたは Storage Manager を使うのであれば、rbw.config ファイルにコンフィグレーション情報を追加します。(この後に記述)

この状態で、バックアップノリストのコンフィグレーションは完了しており、バックアップオペレーションを行う準備ができています。

バックアップデータ

TMU コマンドを使ったバックアップ:

バックアップ/リストアの操作は TMU のコントロールファイル中に記述したコマンドを用いて実行されます。例えば、レベル 0 のオンラインバックアップをディレクトリ/backup_dir0 に実行する場合、コントロールファイル backup_10.tmu の中に以下のラインを記述します。

backup to directory '/backup0'

online level 0;

例えば、(>rb_tmu backup_10.tmu admin adminpwd)のようなコマンドを実行することで、レベル 0 バックアップを行います。(admin と adminpwd は、DBA のユーザ名とパスワードです。)

差分バックアップは、適正なコマンドを TMU コントロールファイルに記述します。例えば、レベル 1 のチェックポイントバックアップは、

backup to directory '/backup1' checkpoint level 1;

レベル 2 のチェックポイントバックアップでは:

backup to directory '/backup2' checkpoint level 2;

パフォーマンス:

バックアップ(リストア)オペレーションのパフォーマンスのポイントは、I/O の集中です。バックアップ/リストアのオペレーションは、シリアルに実行されます。そのため、TMU は複数の同時バックアップ/リストアセッションをサポートしていません。これは、Storage Manager による同時バックアップ/リストアをサポートしないことを意味します。

この例の場合は、ローカルマシンのディスクファイルへのバックアップを行いました。バックアップ操作は 64-bit Solaris Sparc9 (750 MHz)で実施しました。平均のバックアップ/リストアのレートは約 1GB/min ほどです。

- ・約 7.5 時間でレベル 0 のバックアップが完了しました。(530GB、以下に rb_tmu の出力)
- ・レベル 2 バックアップは約 1.1 時間で完了(80 GB data)
- ・レベル 1 バックアップは約 1.5 時間で完了(106 GB data)

以下の出力は TMU のレベル 0 バックアップ処理で表示されます:

(C) Copyright IBM Corp. 1991-2002.

All rights reserved.

Version 06.20.0000(0)TST

** INFORMATION ** (523) Backup of database

'/perf/local/33/huiliao/toucan_db'

with backup level 0,

backup type ONLINE, and

backup media DIRECTORY started.

...

```

** INFORMATION ** (7051) Backup to
'/backup_dir0/rb_bar_liao_toucan_db.20030413.
043719.00022549.0547'
started on Sunday, April 13, 2003 4:37:19 AM.
** INFORMATION ** (7061) Backup to
'/backup_dir0/rb_bar_liao_toucan_db.20030413.
043719.00022549.0547'
completed on Sunday, April 13, 2003 4:37:19 AM.
** INFORMATION ** (7087) Backup of the database
/perf/local/33/huiliao/toucan_db
completed successfully
on Sunday, April 13, 2003 4:37:20 AM.
** STATISTICS ** (500) Time = 04:56:14.36
cp time,
07:27:03.36 time, Logical IO count=69645752,
Blk Reads=2640848, Blk Writes=2474915

```

time コーティリティーにより上記コマンドの出力を示します。

```

real 7:27:05.4
user 18:49.8
sys 4:37:24.6

```

Storage Manager を使用した場合、Storage Manager によって提供されるデータの圧縮は、パフォーマンスの向上に役立ちます。(以下のバックアップデータ圧縮を参照) またパフォーマンス向上の幅に関しては、データの圧縮度に依存します。

例えば、8.5GB のデータベースを AIX 4.3 64-bit platform (CPU: ~370MHz) プラットフォーム Tivoli Storage Manage を使用して、フルバックアップでのテストを行いました。

データを圧縮した場合のパフォーマンスを検証するため、圧縮の有、無を指定してテストを行いました。TSM クライアントとサーバは同一サーバ上にあり、バックアップ用ストレージは、ローカルにある環境で行いました。テストした結果を見ると、データを圧縮した場合のフルバックアップは、圧縮しない場合に比べて、約 1/3 の時間のパフォーマンスが期待できます。以下にテスト結果を示します。

(表)

	Directly to local disk files	TSM, local disk files, no data compression	TSM, local disk files, data compression
Full backup time	43 minutes	48 minutes	17 minutes

データ圧縮:

TMU 自身はバックアップデータの圧縮をサポートしていません。そのため、バックアップをディスクディレクトリ、またはテープに行く場合、バックアップ容量よりもわずかに多くの領域を必要とします。その拡張領域は、バックアップ用の PSU メタデータ情報のために使用されます。

このオーバーヘッドは小さく、PSU あたり 16KB から 40KB 程度です。(530GB のデータベースでは、バックアップスペースのオーバーヘッドは、およそ 4MB 程度です)バックアップデータの圧縮を TMU のバックアップノリストアでは、外部機器での圧縮指定で達成できます。以下にアプローチを示します：

・外部フルバックアップ:

もしレベル0バックアップで外部バックアップツールを使用する場合、それは適切な選択となります。例えば、tar ユーティリティを使用する場合は、以下の様に記述します。

```
tar cf - * | gzip > archive.gz
```

・ストレージマネージャによって得られる圧縮:

多くのストレージマネージャは、バックアップデータに対して、データの圧縮を提供しています。たとえば、Tivoli Storage Manager の場合、バックアップの前に Tivoli Server Administration Web Interface の Client Compression をチェックします。バックアップデータは TMU から送られ、Storage Manager に送付する前に圧縮されます。テストでは、8.3GB のデータベースに圧縮指定をしましたが、その場合のバックアップスペースは 1.02GB で、圧縮する前に比べて約 1/8 のサイズになりました。

データのリストア:

データベースリストアはデータベース内の全てのオブジェクトを回復します。デフォルトの TMU リストアでは、最後のチェックポイントまでのリストアを行います。例えば、530GB のデータベースを日曜 AM8:00 に実行する場合、まずデータベースの処理をストップします。次に以下の記述のある TMU コントロールファイルを実行します。

```
restore;
```

リストアパスは以下の通りです。最初のリストアは、月曜の AM12:00 レベル0バックアップから行います。次に、金曜日の 12:00AM レベル1のバックアップから行い、最後に土曜日の AM12:00 レベル2のバックアップを行います。合計のリストア時間は、 $(7 + 1 + 1.4) = 9.4$ 時間となり、これは、要求されたリストア時間に当てはまります。(24 時間を超えない)

(C) Copyright IBM Corp. 1991-2002. All rights reserved.

Version 06.20.0000(0)TST

```
** INFORMATION ** (7054) Starting database restore  
of database /perf/local/33/huiliao/toucan_db.
```

```
...
```

```
** INFORMATION ** (7044) Completed restore from  
/Backup_dir2/rb_bar_liao_toucan_db.20030413.
```

091358.00004299.0001

on Sunday, April 13, 2003 6:22:05 PM.

**** INFORMATION ** (560) Restore process will
re-start the database**

/perf/local/33/huiliao/toucan_db now.

**** INFORMATION ** (7088) Restore of the database**

/perf/local/33/huiliao/toucan_db completed

successfully on Sunday, April 13,

2003 6:22:45 PM.

**** STATISTICS ** (500) Time = 05:41:04.68**

cp time, 09:24:08.91 time,

Logical IO count =94087412,

Blk Reads=3330116, Blk Writes=3443960

Here is the output from the "time" utility for the above rb_tmu command:

これは、rb_tmu コマンドの Time ユーティリティから出力されます。

real 9:24:08.9

user 19:49.1

sys 5:23:15.6

このリストア時間はバックアップスピードとほぼ同一です。リストア時間は、リストアするデータ量とリストアを実行するマシン上のディスク又はテープの I/O スピードに依存します。

いくつかの管理者用のヒント

バックアップメディア履歴ファイルの操作:

テキストベースのメディア履歴ファイル(\$RB_CONFIG/bar_metadata/<db_name>/rbw_media_history)は全ての PSU バックアップのためのバックアップ情報を含んでいます。このファイルは、バックアップオペレーションによって増加します。OS のファイルサイズに限度がある場合、このサイズをチェックした方がよいでしょう。ファイルサイズが上限に到達した場合、バックアップの処理が出来なくなります。その場合には、以前のバックアップコマンドで作成された不要のレコードを削除します。

この場合には、vi や Emacs の様なエディタを使用して直接編集します。編集する場合には、特に注意が必要です。注意事項については、TMU のリファレンスガイドを参照して下さい。そのほかの方法としては、TMU_LOAD ユーティリティを使って、テーブルにファイルをロードし、SQL 文で不要なレコードを削除します。SQL_EXPORT 文で修正されたレコードをメディア履歴ファイルへ戻します。

レコード削除の間は、削除されるレコードに該当するバックアップファイルのリストを(STORAGE MANAGER を

使用する場合 XBASA のコピーID)取得します。メディア履歴ファイルからレコードを削除した後、バックアップファイル(XBSA オブジェクト)をバックアップスペースを開放するために削除します。

以下のサンプルは、ロードユーティリティの使用方法和 EXPORT 文でメディア履歴レコードを削除する方法です。このケースでは、下記の処理により、04/25/2003 以前のレコードを全て削除します。

- 1.オリジナルの rb2_media_history ファイルを保存します。
- 2.メディア履歴レコードのフォーマットとマッチするようにカラム定義された表を作成します:(TMU Reference Guide を参照して下さい)

```
RISQL> create table media_history (  
    seqnum varchar(10),  
    version varchar(10),  
    bklevel char(1),  
    bktype char(1),  
    medtype char(1),  
    time char(19),  
    psutype char(1),  
    fullpsu char(1),  
    mediaid varchar(1024),  
    bksize varchar(10),  
    offset varchar(10),  
    psusize varchar(10),  
    psuname varchar(1024),  
    segname varchar(128)  
)
```

Load the media history file into the above table media_history,
using the following TMU control file:

- 3.メディア履歴ファイルを表 media_history へロードします。

TMU コントロールファイルは以下の通りです:

```
load data inputfile ('rbw_media_history')  
replace  
format separated by '  
into table media_history (  
    seqnum char,  
    version char,  
    bklevel char,  
    bktype char,
```

medtype char,
time char,
psutype char,
fullpsu char,
mediaid char,
bksize char,
offset char,
psusize char,
psuname char,
segname char

);

4.表 media_history へのクエリが、04/25/2003 以前のレコードを削除するかを確認します。削除の後、バックアップ/リストアの操作ができます。

例えばこのケースの場合、レコードを削除した後に、差分バックやデータベースのリストアをしたとします。その場合、レベル0のバックアップが 04/25/2003 以降に行われたものがあるかを確認します。メディア履歴レコードではバックアップオペレーションをユニークな番号で特定しますので、その番号をクエリの中で使用します。

-- check that there is a level 0 backup

-- performed after 04/25/03

```
RISQL> select distinct(int(seqnum))
```

```
from media_history
```

```
where time>'2003-04-26'
```

```
and bklevel='0';
```

```
SEQNUM
```

30

-- check that there are checkpoint

-- backups performed

-- after the level 0 backup

```
RISQL> select distinct(int(seqnum))
```

```
from media_history
```

```
where int(seqnum)>30 and bktype='C';
```

```
SEQNUM
```

32

34

40

5.クエリ結果からは、レベル0 によるバックアップでバックアップナンバー30 で04/25/2003以降に実施した内容を見ることができます。そして、その後3つのチェックポイントバックアップが実施されています。このことから、04/25/2003以前のレコードは削除しても安全です。最後にバックアップメディアIDのレコードリスト(mediaID_list)を保存し、削除します。

```
RISQL> create view mediaID_list
as select
mediaid from media_history
where int(seqnum)<30;
RISQL> set export_delimiter ' ';
RISQL> export to 'mediaID_list'
format delimited
(select * from hui_view);
```

6.メディア履歴表からレコードを削除します。30(有効となるレベル0のバックアップの番号)以下の番号のレコードを削除します。

```
RISQL> delete from media_history
where int(seqnum) < 30;
** INFORMATION ** (211) Deleted
803 rows from MEDIA_HISTORY.
```

7.media_history 表をメディア履歴ファイル rbw_media_history へエクスポートします。

```
RISQL> set export_delimiter ' ';
RISQL> export to
'rbw_media_history'
format delimited
(select * from media_history);
```

8.rbw_media_history ファイルがうまくアップデートされたかをチェックします:例えば、有効なレベル0バックアップとチェックポイントバックアップが存在することを確認します。アップデートファイルが有効かを確認して、レベル1,2のバックアップを実行します。

9.最後に mediaID_list file に格納されているバックアップファイルを削除し、バックアップスペースをフリーにします。

バックアップが失敗したとき:

バックアップオペレーションが失敗した場合、失敗したポイント直前のバックアップまでが有効となります。失敗し

たバックアップポイントがチェックポイントの場合、失敗したポイントの前で、既に成功している全ての PSU にバックアップモードをマークします。この方法により最適なエラーリカバリ処理が実施されます。特に中、長時間のバックアップ実行時には有効です。この場合、再度バックアップを実行をしますが、その場合、TMU は、失敗したチェックポイント以降のバックアップされていないデータだけをバックアップします。

データベースが破壊された時:

RedBrick はバックアップメタデータディレクトリ (\$RB_CONFIG/bar_metadata/database_name)に完全な形(バックアップコピーからリストアできる)でデータベースのバックアップデータを保持しており、その場合には、コールドリストアが可能です。例えば、データベースパスが/redbrick/DB で、このディレクトリ以下がアクシデントで削除された場合、以下のステップでデータベースをリストアできます。

1.環境変数 RB_PATH と RB_HOST を確認し、失われたデータベースと同じ値をセットします。ロケールも同一です。(このケースの場合、ロケールは English を指定)

2.データベースの再作成:

```
rb_creator /redbrick/DB
```

(C) Copyright IBM Corp. 1991-2002.

All rights reserved.

Version 06.20.0000(0)TST

** INFORMATION ** (1899) Database

created using locale

'English_UnitedStates.

US-ASCII@Binary'.

3.失ったデータベースのバックアップメタデータディレクトリの内容が完全な場合には、さらなるアクションは必要ありません。また、別の方法として、バックアップコピーからディレクトリを回復することも可能です。

4.以下のコマンドを使用した TMU コントロールファイルを使ったデータベース回復の開始:

```
restore;
```

データベースリストアの一部としてバックアップのセグメントは回復しました。データベースが正常に回復した後、すぐに差分バックアップを実行します。

```
RISQL> select BACKUP_SEGMENT
```

```
from dst_databases;
```

```
BACKUP_SEGMENT
```

```
BACKUPSEG
```

storage management system の構築

いくつかの storage management system は X/Open Backup Services API (XBSA)に準拠しています。例えば、IBM Tivoli Storage Manager や Legato Networker などが該当します。TMU backup/restore は、XBSA API を介して Storage Manage に管理されます。ユーザは TMU コマンドを使用して backup/restore 操作を行い、location やバックアップ・コンテンツの管理を storage management が行うこととなります。

TMU backup/restore を XBSA STORAGE システムと使用する場合、インストール、コンフィグレーション、動作に必要なリクエストを実行します。また Tivoli と Legato では、ベンダーにより、XBSA API に関して幾つかの違いがあります。TMU ユーザが異なる STORAGE SYSTEM のコンフィグレーションを設定する場合、多少の設定方法の違いが生じます。

IBM Red Brick の開発チームでは2つの STORAGE MANAGER SYSTEM でテストを行っています。(Tivoli Storage Manager と Legato Networker)storage management systems と TMU で XBSA に問題が発生した場合には、テクニカルサポートまで連絡して下さい。

このセクションでは、Tivoli と Legato を使用したコンフィグレーションの方法について説明していますが、ひとつひとつのプロセスに関しては、個人の経験がベースとなっているため、リファレンスの目的のみで利用ください。また完全なコンフィグレーションのガイドラインを知るには、storage manager のドキュメントを参照する必要があります。

Tivoli Storage Manager を使用: (Version 5, Release 1, on an AIX machine)

Tivoli SM software のインストール:要求されたソフトウェアに含むもの:

Tivoli Storage Manager Server, Tivoli Storage Management Device Support と Tivoli Storage Manager Backup-Archive Client.

1. Tivoli Storage Manager Server と Device Support をマシン host1 にインストールします。
2. TSM server (dsmserv)をスタートします。
3. Red Brick Warehouse server のインストールされているサーバ(マシン host1)に Backup-Archive client をインストールします。この時、XBSA ライブラリは、/usr/tivoli/tsm/client/Informix/bin64/bsahr10.0 以下にインストールされます。

Tivoli SM のコンフィグレーション:

インストールの後、Tivoli SM のデフォルトコンフィグレーションにより、TMU backup/restore が可能です。しかし要求する、ニーズにマッチしない場合には、以下のコンフィグレーションのカスタマイズを行います。

1. TSM Server Administration にログインします。Web インターフェイス <http://host1:1580>
2. backup storage を作成します(このケースでは、バックアップメディアとしてディスクファイルを使用します。)

(1) storage pool の作成

Object View -> Server Storage -> Storage Pools -> Disk storage pools を実行します。Disk Storage Pools のクリックにより disk storage pool "BAR_DISKPOOL"を作成します。そして、Operations "Define a

new disk storage pool"を選択します。

(2) storage pool のボリュームの作成

Object View -> Server Storage -> Storage Pools -> Disk storage pools -> BAR_DISKPOOL -> Volumes を実行します。

Operations "Define a disk storage pool volume"を選択しボリューム名を選択します。
"/BAR/storage/backup_dir/file1";このディスクは TMU バックアップに使用します。必要なボリュームを定義します。

3. バックアップポリシーと新しいクライアントノードを作成します。

(1) ポリシードメインの定義:

Object View -> Policy Domains: Options (Define Policy Domain).

"BARPolicy"と呼ばれるポリシードメインを定義します。

(2) ポリシーセットを定義しクラス管理:

Object View -> Policy Domains -> BARPolicy -> Policy Sets: Options (ポリシーセットの定義).ポリシーセットの"BARPolicySet."を定義します。同様のプロシージャを使用してポリシーセットの "Management Classes"を定義します。management class の"BARPolicyMagmt."を定義します。

(3) 定義されたポリシーセットに、上記の(BAR_DISKPOOL)定義された storage pool を割り付けます:

Object View -> Policy Domains -> BARPolicy -> BARPolicySet -> Management Classes -> BARPolicyMagmt -> Backup Groups -> Define Backup Group.

Choose Copy Destination を"BAR_DISKPOOL." にセットします。

(4) 上記で定義された management class "BARPolicyMagmt"を Default Management Class.として割り付けます。

(5) 定義された policy set "BARPolicySet." を Validate、Activate します。

(6) 定義されたポリシードメインのために新しいクライアントノードを登録します。Object View -> Policy Domains -> BARPolicy -> Client Nodes: Operations "Register a new node."Policy Domain Name を定義された"BARPolicy."にセットします。バックアップデータを圧縮する場合、Client Compression Setting を "YES."とします。TMU backup/restore のコンフィグオプションをセットするにあたり、BAR_SM_USER をセットしノード名を入力します。ノード名は "BAR_CLIENT."を指定します。

環境変数とエラーログファイルはクライアントマシンにあります:

・DSM_DIR: クライアントインストールディレクトリを指定する環境変数です。クライアントソフトウェアがデフォルト以外にインストールされている場合、この環境変数を使用します。(例えば、テストケースでのクライアントインストールディレクトリはデフォルトを指定しています。 /usr/Tivoli/tsm/client/api/bin64.)

・ DSMI_CONFIG:TSM クライアントから接続するサーバのコンフィグレーションファイル (dsm.opt)の場所を指定します。クライアントインストールディレクトリからサンプルファイル(dsm.opt.smp)を入手できます。DSM.OPTファイルで定義するサーバ名は、DSM.SYSファイル内で定義したサーバ名の位置を指定します。

・ dserror.log: エラーが発生した場合には、このファイルにロギングされます。デフォルトでは、XBSA

application (テストケースは **RB_TMU**) が動作しているディレクトリに作成されます。作成されるディレクトリを確認して、適切なパーティションを付加します。

TMU backup/restore のコンフィグレーション:

1. **Red Brick barxbsa** ユーティリティを使用して TSM サーバとのコネクティビティを確認します。TSM クライアントマシンから実行します。

TSM クライアント上のファイル **dsm.opt** に正確な TSM サーバー名が指定されたかを確認して下さい。例えばこのケースでは、TSM サーバはマシン **host1** で動作します。TSM クライアントはマシン **client1** にインストールされています。(Red Brick サーバはインストールされている)**dsm.opt** と **dsm.sys** ファイルはクライアントマシン上にあり、以下のラインを含んでいます。

contain the following lines:

cat **dsm.opt**:

```
SErvername host1
```

cat **dsm.sys**:

```
Servername      host1
COMMmethod      TCPip
TCPport         1500
TCPserveraddress host1
```

barxbsa command:

```
barxbsa /usr/tivoli/tsm/client/Informix/
bin64/bsahr10.o BAR_CLIENT
```

For a successful conneciton, the following messge will be displayed:

```
OPTION BAR_XBSA_LIB
/usr/tivoli/tsm/client/Informix/
bin64/bsahr10.o
OPTION BAR_SM_USER BAR_CLIENT
```

```
XBSA version: 4, release: 2, level: 0
** INFORMATION ** (7010) BAR library
operation was successful.
```

TSM を使用する場合の **rbw.config** ファイルのアップデート:

```
Option BAR_SM_USER BAR_CLIENT
Option BAR_XBSA_LIB
/usr/tivoli/tsm/client/Informix/bin64/bsahr10.o
```

"BAR_CLIENT"はクライアントノード名を上記のように指定しました。

以上で TMU backup/restore 実行の準備の完了です。例えば、レベル 0 のバックアップを実行しようとした場合には、以下のコマンドを実行します。

backup to XBSA online level 0;

Legato Networker の使用 (Version 6.0.1, on a Solaris machine)

Legato software のインストール

Legato software には、Legato Networker Server, Legato Networker Client, と Informix Networker モジュールを含んでいます。

1. Legato Networker Server ソフトウェアをマシン host 2 にインストールします。
2. Legato Networker Client ソフトウェアをマシン clinet2 にインストールします。
client 2 マシンには、Red Brick サーバがインストールされています。
3. Informix Networker は、client2 のディレクトリ/work/Legato/client/にあります。
XBSA library libxnmi.so.1 は TMU/バックアップノリストアで使用されます。

Legato Networker の初期化:

1. Legato Networker server を開始します。

```
/etc/init.d/networker start
```

通常、スクリプトは rc ディレクトリに存在し、ブート時に有効となります。

2. Legato NW GUI administrator tool を実行します。:nwadmin
3. "redbrick"ユーザに管理者権限を与えます。(ユーザ"RedBrick"は rb_tmu のユーザとして backup/restore を実行します)
server をクリックしサーバウィンドウを起動します。Administrator オプションで、ユーザ名 RedBrick を指定し、Add をクリックします。
4. バックアップメディアを作成します。Device ウィンドウの下に使用可能なテープデバイスが自動で表示されます。テープデバイスへのラベル付けとマウントを行い、バックアップメディアを選択します。このケースでは、バックアップメディアとしてディスクファイルを使用しました。
Media -> Devices -> Create: Name: バックアップディレクトリのパス名 (このケースの場合 /BAR/Legato/Backup_dir in this case).
メディアタイプ: ファイル
5. バックアップメディアプールを作成します。
Media -> Pools -> Create: Name: プール名の入力
例えば、ここでは TMU バックアップ用プールとして、"BARpool"を定義します。新しいプールを必要としない場合、"Default"プールを使用します。
6. デバイス名を付けます。
Label ウィンドウを表示し、ボリューム名を入力して、プール名を選択します。
例えば、ここでは"BARpool"をチェックしました。

7. デバイスをマウントします。

デバイス名を明示化し、**Mount** ボタンをクリックします。

環境変数の値をセット:

- ・ **NSR_DATA_VOLUME_POOL:**

この環境変数に使用するボリュームプールを指定し、バックアップ時に使用します。デフォルトでは、"Default"が使用されます。特別なボリュームをターゲットとする場合に、この環境にプール名を指定します。このケースでは、バックアップ実行のために、メディアプール"BARpool"を指定しました。

```
export NSR_DATA_VOLUME_POOL=BARpool
```

- ・ **NSR_DEBUG_LEVEL:**

XBSA トラッキングをオンとします。例えば、

```
export NSG_DEBUG_LEVEL=6
```

- ・ **NSR_DEBUG_FILE:**

トレース情報を格納するファイルです

- ・ **NSR_SERVER:**

クライアントと接続をするサーバを指定します。

- ・ **NSR_NO_BUSY_ERRORS:NSR**

サーバが実行されていない場合、接続のタイムアウトエラーとなります。この場合、代わりに **Storage Manager** デーモンへの接続のリトライを行います。サーバーからの答えがなければ、NSR クライアントは接続を再び試みません、値は **TRUE** を指定します。

サーバからの応答がない場合には、**rb_tmu** を実行できません。

TMU backup/restore のコンフィグレーション:

1.Red Brick barxbsa ユーティリティから NSR サーバとの接続性を確認します。

```
barxbsa /work/Legato/client/libxnmi.so.1
```

```
INFORMIX
```

正確には、"INFORMIX" の文字列が使用されます。接続が成功した場合には以下のメッセージが表示されます。

```
OPTION BAR_XBSA_LIB /work/Legato/client/libxnmi.so.1
```

```
OPTION BAR_SM_USER INFORMIX
```

```
XBSA version: 1, release: 1, level: 1
```

```
** INFORMATION ** (7010) BAR library
```

```
operation was successful.
```

2.Legato Networker を使用した rbw.config の更新

```
Option BAR_SM_USER INFORMIX
```

Option BAR_XBSA_LIB /work/Legato/client/libxnmi.so.1

文字列/work/Legato/client/libxnmi.so.1を actual XBSA library path として、実際のマシン環境に合わせて変更します。

Note : BAR_SM_USER は INFORMIX を指定します。

以上で、TMU backup/Restore オペレーション実行の準備ができました。

最後に IBM Red Brick Warehouse の backup/Restore 機能は、フルと差分のバックアップをサポートしており、自動的なデータのリストアが可能です。またチェックポイントバックアップモードと同様にオンラインバックアップもサポートしており、オンラインバックアップを使用する場合には、システムのダウンタイムを必要としません。TMU は、ディスクとテープデバイスへの直接バックアップを可能とし、同様に、XBAS I/F をサポートする storage management systems をサポートします。TMU によるバックアップとリストアのコンフィグレーションの設定と管理は容易で、ケーススタディーとして大型の DWH に対して実証しています。さらにバックアップパフォーマンスを追及し、潜在的な管理問題の解決策や、Tivoli、Legato の storage management system のためのコンフィグレーション要件をカバーしました。

Disclaimer

The configuration information contained in this document has not been submitted to any formal IBM test. Anyone attempting to adapt these techniques to their own environments does so at his or her own risk. The use of this information or the implementation of any of these techniques is derived under specific operating and environmental conditions. While the information has been reviewed for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.