

Schnelles Backup & Restore mit Multisection

Sinan Petrus Toma
Finanz Informatik GmbH & Co. KG
Hannover

Schlüsselworte:

Backup, Restore, Multisection, Section Size, Multiplexed Backup

Einleitung

Es werden verschiedene Konstellationen zur Verbesserung der Backup- und vor allem der Restore-Geschwindigkeit vorgestellt. Folgende Parameter werden diskutiert und deren Beeinflussung auf die Verteilung der Datafiles auf die Backup-Sets untersucht:

FILESERSET, MAXOPENFILES, SECTION SIZE, BACKUP_TAPE_IO_SLAVES, COMPRESSION, und die Anzahl der allokierten RMAN-Channels.

Zu jeder Parameterkonfiguration wird ein Beispiel und dessen Ergebnis vorgestellt. Weiterhin werden die Unterschiede zwischen Multisection-, Multiplexed- und Duplexed-Backups erläutert.

Multisection-Backup

Die Idee von Multisection-Backup ist, ein großes Datafile in mehreren Bereichen (Sections) zu teilen und diese mit mehreren RMAN-Channels parallel zu sichern. Jede Section präsentiert einen zusammenhängenden Bereich von Datenblöcken. Das Backup wird durch das RMAN-Schlüsselwort `SECTION SIZE` mit Angabe der Section-Größe gekennzeichnet:

```
RMAN> BACKUP DATABASE SECTION SIZE = 200M;
```

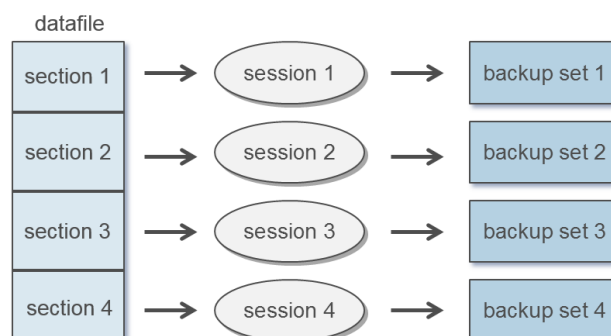


Abb. 1: Multisection-Backup

Multisection-Backup wurde das erste Mal in Oracle 11g vorgestellt. Es war für Full- und Incremental-Level-0-Backup-Sets verfügbar. In Oracle 12c können nun auch Full- und Incremental-Level-0-Image-Copy und Incremental-Level-1-Backup-Sets mit Multisection durchgeführt werden.

Backup-Konfiguration

Folgende Parameter beeinflussen die Verteilung der Datafiles auf die Backup-Sets und die Backup- und Restore-Geschwindigkeit:

PARALLELISM: bestimmt die Anzahl der RMAN-Prozesse, die gleichzeitig ausgeführt werden. Der Standardwert ist 1.

MAXOPENFILES: bestimmt die Anzahl der Datafiles, die von einem RMAN-Prozess (Channel) gleichzeitig gesichert werden können. Der Standardwert ist 8.

FILESERSET: bestimmt die maximale Anzahl der Datafiles in einem Backup-Set. Der Standardwert ist 64.

MAXPIECESIZE: bestimmt die maximale Größe eines Backup-Piece. Die maximale Größe ist nicht limitiert.

SECTION SIZE: bestimmt die maximale Größe eines Bereiches bei Multisection-Backups. Für diese Option gibt es keinen Standardwert und muss daher immer angegeben werden.

Beispiel 1: PARALLELISM = 4

Mit dem Allokieren von mehreren Channels wird die Backup-Geschwindigkeit erheblich verbessert. Die weiteren Parameter behalten Ihre Standardwerte. Somit sichert jede Channel bis zu 8 Datafiles in parallel und speichert deren Blöcke in einem Backup-Piece. Diese Art von Backups wird „Multiplexed Backup“ genannt, da ein Backup-Set mehrere Datafiles mit gemischten Blöcken enthält. Es sind nur komplette Datafiles und niemals nur Teile eines Datafiles in einem Backup-Piece enthalten. Die maximale Anzahl der Datafiles in einem Backup-Piece wird als „Level of Multiplexing“ bezeichnet und wird durch folgende Formel bestimmt:

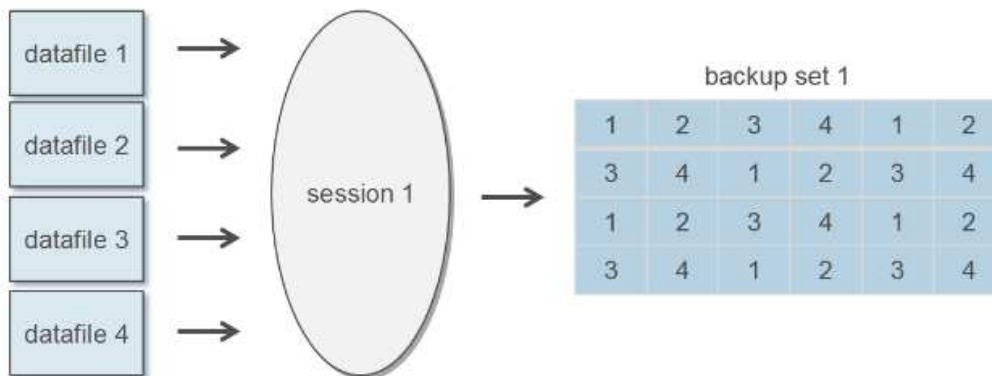
$$\text{Multiplexing Level} = \min(\text{FILESERSET}, \min(\text{MAXOPENFILES}, \text{files_per_channel}))$$


Abb. 2: Multiplexed-Backup

Die Nachteile dieser Art von Backups spiegeln sich bei der Restore-Geschwindigkeit wider. Das große Backup-Piece mit mehreren Datafiles kann nur mit einer Channel eingelesen werden. Sollte nur einer der Datafiles zurückgesichert werden, müssen trotzdem alle Blöcke im Backup-Piece durchsucht werden.

Beispiel 2: MAXOPENFILES = 1

Mit dieser Option verhindern wir nun, dass eine Channel mehrere Datafiles gleichzeitig einliest und in einem Backup-Set speichert. Da aber FILESPERSET immer noch den Standardwert von 64 hat, kann es vorkommen, dass eine Channel mehrere Datafiles im gleichen Backup-Piece speichert, nur nicht gleichzeitig, sondern hintereinander.

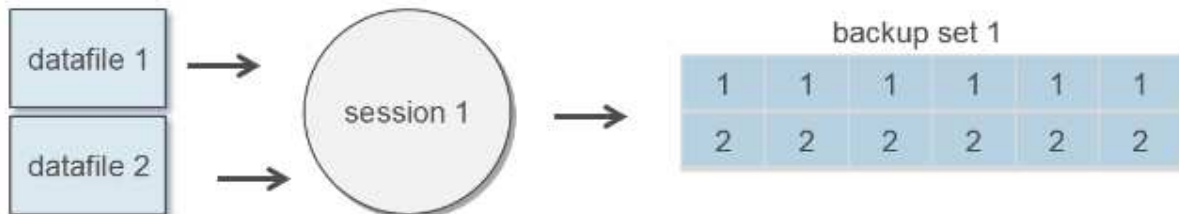


Abb. 3: MAXOPENFILES = 1

Laut Definition handelt es sich hier um kein Multiplexed-Backup, obwohl mehrere Datafiles im gleichen Backup-Piece gesichert sind. Die Datafile-Blöcke sind nicht gemischt, sondern hintereinander platziert.

Beispiel 3: FILESPERSET = 1

Um nun RMAN zu zwingen, die Blöcke eines jeden Datafiles separat in einem Backup-Piece zu speichern, setzen wir den Parameter FILESPERSET auf den Wert 1. Hierdurch entsteht folgende Situation:

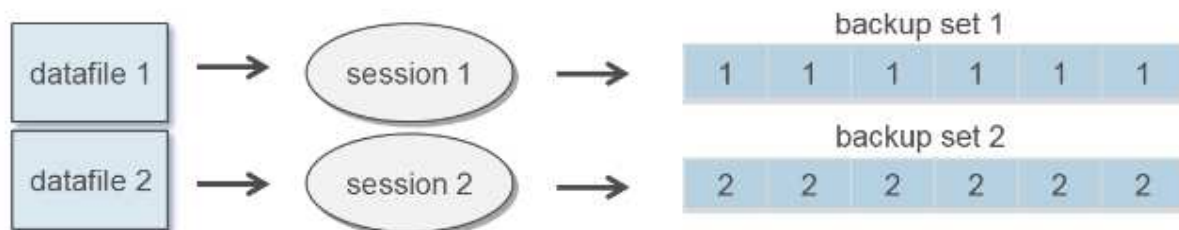


Abb. 4: FILESPERSET = 1

Der Nachteil hier ist weiterhin, dass die Zursicherung eines Datafiles immer noch mit maximal einer Channel durchgeführt werden kann.

Beispiel 4: MAXPIECESIZE setzen

Um die Blöcke eines Datafiles auf mehrere Backup-Pieces zu verteilen, setzen wir die maximale Backup-Piece-Größe auf z.B. ein Viertel der Datafile-Größe. Dadurch wird das Datafile auf vier Backup-Pieces verteilt.

Zu unserer Überraschung (oder auch keine Überraschung) wird die Zursicherung hier auch mit maximal einer Channel stattfinden, auch wenn mehrere Channels allokiert sind. Das Problem liegt hier nicht darin, dass RMAN nur aus einem Backup-Piece lesen kann, sondern darin, dass RMAN nur mit einer Channel schreiben kann.

Beispiel 5: SECTION SIZE setzen

Um endlich unser Ziel erreichen zu können, ein Datafile mit mehreren Channels gleichzeitig zurückzusichern, greifen wir auf die Multisection-Option zu. Hier wird RMAN auch beim Zurücksichern mehrere Channels verwenden können, da die Sections zusammenhängende Blockbereiche darstellen und diese gleichzeitig geschrieben werden können.

Multisection: Weitere Informationen

Die Optionen SECTION SIZE und MAXPIECESIZE schließen sich gegenseitig aus. D.h., man kann nur das eine oder das andere in einem Backup-Command angeben. Denn im Endeffekt bestimmen beide das Gleiche: die maximale Backup-Piece-Größe.

Sollte mit SECTION SIZE gleichzeitig FILESPERSET größer als 1 gesetzt sein, wird dieser auf 1 zurückgesetzt. Denn FILESPERSET ist eigentlich die Anzahl Datafiles pro Backup-Piece und diese kann bei Multisection nur eins sein. Wenn MAXPIECESIZE auf ihren Standardwert ist – nicht limitiert – dann besteht ein Backup-Set i.d.R. aus nur einem Backup-Piece.

Ein Datafile kann maximal in 256 Sections unterteilt werden. Ist SECTION SIZE so klein ausgewählt, dass sich für ein Datafile mehr als 256 Sections ergeben, erhöht RMAN die SECTION SIZE für dieses Datafile automatisch auf einen Wert, der genau 256 Sections erzeugt.

Ist SECTION SIZE größer als die Datafile-Größe, so wird das gesamte Datafile als eine Section behandelt und in einem Backup-Piece gesichert. D.h. RMAN verwendet für dieses Datafile kein Multisection.

Sollte ein Multisection-Backup abbrechen, bevor alle Sections eines Datafiles gesichert sind, kann diese in unvollständige Backup-Metadaten im RMAN Repository (Controlfile und/oder Catalog) resultieren.

Diese Teile kommen bei einem Restore durch RMAN nicht in Frage. Die Metadaten müssen manuell mit dem DELETE-Command entfernt werden. Beim Wiederholen des Backups werden alle Sections des Datafiles erneut gesichert.

Multisection: Überlegungen

Die Frage, die sich bei Multisection-Backup stellt ist die Section-Größe und wie sie auszuwählen ist. Die Größe der Sections sollte abhängig von der Datafile-Größe und dem Parallelisierungsgrad gewählt werden. Sind die meisten Datafiles in der Datenbank z.B. 32GB groß und man führt ein Backup mit 8 Channels aus, so ist die Section-Größe auf 4GB zu setzen.

Es ist ebenfalls möglich, unterschiedliche Datafiles je nach deren Größe mit unterschiedlicher Section-Größe zu sichern. Dies erfordert allerdings mehr Koordination, denn diese Datafiles müssen separat gesichert werden.

Wenn die Section-Größe zu kleine ausgewählt ist, resultiert dies in eine sehr große Menge an Backup-Pieces. Hier ist zu beachten, dass der Speicher-Manager dies nicht negativ beeinflusst.

Parameter BACKUP_TAPE_IO_SLAVES

Ist dieser Parameter auf seinen Standardwert (=FALSE) gesetzt, so allokiert RMAN die Tape-Buffers von der PGA. Daher sollte dieser Parameter auf TRUE gesetzt sein, damit RMAN die Tape-Buffers von der SGA allokiert. Oracle kann Speicher in der SGA effizienter verwalten als in der PGA.

Es ist ebenfalls zu empfehlen, den Parameter `LARGE_POOL_SIZE` zu setzen, ansonsten erfolgt das Allokieren des Speichers vom Shared-Pool.

Backup-Komprimierung

Komprimierung erfordert CPU-Ressourcen und dies bedeutet mehr Zeit beim Backup. Ist der Speicherplatz kein Problem, so könnte man auf die Komprimierung verzichten. Dies ist aber meistens nicht der Fall, da durch die Komprimierung große Mengen an Speicherplatz gespart werden können. Sollte die Advanced Compression Option vorhanden sein, so ist `MEDIUM` Compression zu wählen, da diese eine sehr gute Kombination aus Komprimierungsrate und Geschwindigkeit bietet. Ansonsten kann `BASIC` verwendet werden, wenn auf die Komprimierung nicht verzichtet werden kann.

Multisection VALIDATE

Es ist möglich, auch bei der Datenbankvalidierung mit dem `RMAN VALIDATE`-Command die Option `SECTION SIZE` zu nutzen. Dadurch werden große Datafiles in Sections unterteilt und parallel überprüft:

```
VALIDATE [CHECK LOGICAL] SECTION SIZE 2G DATABASE;
```

Multisection Performance

Bei den vielen Tests, die wir ausgeführt haben, war zu sehen, dass die Backup-Geschwindigkeit mit Multisection beim Sichern der gesamten Datenbank sich um 10 bis 20% gegenüber der Sicherung ohne Multisection verbessert hat.

Bei der Sicherung von großen Datafiles (Bigfile Tablespace mit 100GB) mit vier Channels war die Performance um mehr als 65% besser.

Vor allem beim Restore kommen die Vorteile von Multisection zum Vorschein. Die Restore-Geschwindigkeit für die gesamte Datenbank konnte um 30 bis 40% verbessert werden. Für einen einzigen Datafile (32GB) um bis zu 80%.

Duplexed Backup Sets

Duplexed bedeutet nur, dass mehrere Kopien vom erstellten Backup-Piece erzeugt werden. Diese Möglichkeit kann sowohl beim Multiplexed- als auch bei Multisection-Backups verwendet werden.

Kontaktadresse:

Sinan Petrus Toma
Finanz Informatik GmbH & Co. KG
Laatzener Str. 5
D-30539 Hannover

Telefon: +49 (0) 172 6183618
E-Mail: sinan.petrus.toma@f-i.de
Internet: www.f-i.de