



Backup-Verfahren in der Praxis – optimiert und intelligent

Roland Stirnimann, Trivadis AG

Ein zuverlässiges Backup-Verfahren ist für jedes Unternehmen ein Muss. Dieser Artikel erläutert aktuelle Herausforderungen im Backup-Kontext und beschreibt einen bewährten Ansatz aus der Praxis, um diese zu meistern.

Oracle selbst bietet von Haus aus verschiedene technische Möglichkeiten zur Ressourcen-Optimierung, etwa in Bezug auf die Volumen-Reduktion. Dieses Potenzial nutzen bereits viele Unternehmen, doch darüber hinaus ist viel mehr möglich. Durch intelligente Automatisierung und Standardisierung bietet ein modernes Backup-Verfahren massive Mehrwerte bezüglich Transparenz, Risiko-Minimierung und Effizienz beim täglichen Betrieb.

Hintergrund

IT-Ausfälle und Datenverlust sind für Unternehmen ein Albtraum. Häufig sind Software- oder Netzwerk-Probleme, aber auch menschliche Fehler mögliche Ursachen. Die Kosten können – je nach Kritikalität und Dauer – in die Millionen-

höhe gehen. Entsprechende Risiken gilt es durch ein verlässliches „Backup & Recovery“-Verfahren der Datenbanken zu minimieren. Trotz angemessener Sicherheit soll sich allerdings die Effizienz nicht verschlechtern, da mehr Sicherheit oft in Verbindung mit mehr Daten und Ressourcen gesehen wird und somit letztendlich Mehrkosten verursacht. Es stellt sich also die Frage: Wie können Risiken minimiert werden, ohne die Effizienz zu vernachlässigen? Ein optimales Backup-Verfahren ist erforderlich, das so wenig sichert wie möglich, aber so viel wie nötig.

Aktuelle Backup-Situation im Unternehmen

Die Sicherung von Datenbanken wird in den Unternehmen oft sehr pauschal

durchgeführt und generiert deshalb unverhältnismäßige Betriebskosten. Alle sind sich einig, dass es ein Backup braucht, jedoch ist es in der Praxis häufig schwierig, Argumente zu finden, um Investitionen in Verbesserungen/Erneuerungen tätigen zu können. Erst im Fehlerfall werden Entscheider und Business wachgerüttelt und sich der Wichtigkeit einer zuverlässigen, performanten Sicherung und Wiederherstellung bewusst. Die nachfolgende Zusammenstellung nennt aktuelle Problemzonen aus der Praxis, denen es zu begegnen gilt:

- *Backup-Performance*

Jede Sicherung generiert spürbare Systemlast, primär I/O-Operationen, und Datenvolumen im entsprechenden Speichersystem. Da bei einigen Storage-Herstellern volumenbasiert zu lizenzieren ist, lohnt es sich, beim Back-

up-Volumen genauer hinzuschauen. Das Volumen ist jedoch nur eine Seite der Medaille, denn die Last durch das Lesen der Daten ist in Bezug auf die Datenbank-Performance und letztendlich für die Applikation entscheidend. Backup-bedingte Lese-Operationen belasten direkt das Speichersystem der Datenbank, während die eigentlichen Sicherungsdateien meistens in ein dediziertes System geschrieben werden. In der Praxis ist die Backup-Last zudem oftmals schlecht verteilt, sodass massive Engpässe entstehen, die sich negativ auf die Performance auswirken.

- **Backup-Steuerung**

Die Steuerung und somit die Verteilung der Backup-Jobs ist bei mehreren Hundert Datenbanken nicht ganz einfach. Häufig fehlt der Überblick darüber, was wann startet, und so kommt es zu Lastspitzen aufgrund zu vieler paralleler Backup-Operationen. Ein weiteres Problem der verschiedenen Job-Steuerungen sind die statischen Sicherungs-Intervalle. Die Steuerung kann nicht auf veränderte Situationen reagieren, etwa wenn unerwartet viele Transaktionslog-Daten geschrieben werden und der Plattenplatz ausgeht. Dies führt unweigerlich zum Stillstand der Datenbank. Ein weiteres Beispiel sind wachsende Datenbanken, deren Backup-Dauer zunimmt, wodurch die Sicherungen langsam aber sicher die dafür vorgesehenen Zeitfenster überschreiten und den Geschäftszeiten in die Quere kommen.

- **Backup-Verwaltung**

Selbst große Umgebungen mit Hunderten von Datenbanken verwenden häufig lokale Scheduler wie Cron oder den Windows Task Scheduler. Dies erschwert die Backup-Verwaltung, wenn der Administrator beispielsweise fünf Datenbank-Server in ein Wartungsfenster versetzen möchte und dazu auf jeden Server einzeln verbinden muss. Generell finden viele Aufgaben direkt auf jedem einzelnen Datenbank-Server statt und kosten den Administrator Zeit und Nerven. Auch bei der Konfiguration unterschiedlicher Backup-Anforderungen (wie seitens SLA) in großen Umgebungen ist es eine Herausforderung, den Überblick jederzeit zu behalten.

- **Überwachung und Reporting**

In irgendeiner Form existiert meistens eine Backup-Überwachung. Doch was kann diese beziehungsweise was kann sie nicht? Häufig basiert die Überwachung auf Mail-Nachrichten, die im Fehlerfall versendet werden. Übergreifende Ad-hoc-Auswertungen zum Backup-Volumen, zu Backup-Laufzeiten, zu laufenden und fehlgeschlagenen Sicherungen müssen manuell erstellt werden und sind somit zeitaufwändig. Auch den unterschiedlichen Zielgruppen als Report-Empfänger wird wenig Beachtung geschenkt. Interessante Auswertungen für Management und Entscheider fehlen gänzlich oder lassen sich nur mühsam in manueller Arbeit erstellen. Umfassende Transparenz auf Knopfdruck ist vielerorts ein Wunschdenken.

- **Wiederherstellung**

Weil ein Backup seit Jahren funktioniert, ist dies keine Garantie, dass auch die Wiederherstellung einwandfrei klappt. Die Erfahrung zeigt, dass solche Vorgänge länger dauern als angenommen oder nicht so reibungslos funktionieren wie gedacht. Dies führt zu längeren Ausfallzeiten im Fehlerfall und letztendlich zu mehr Kosten.

Anwendungszwecke einer Sicherung

Bevor die Frage geklärt wird, wie eine optimale Sicherung aussehen könnte, sollte sich ein Unternehmen darüber im Klaren sein, wozu die Backups überhaupt verwendet werden. Natürlich kommt den meisten Leuten der Disaster-Restore als erster Gedanken in den Sinn. Jedoch kann ein Backup in der Praxis für weit mehr verwendet werden, wodurch je nach Anwendungsfall unterschiedliche Anforderungen bestehen.

Insbesondere für den Disaster-Fall glauben teilweise Administratoren, dass eine Hochverfügbarkeitslösung wie Data Guard ein Backup ersetzt. Diese Aussage ist natürlich nicht allgemein gültig. Klar ist die Hochverfügbarkeitslösung die erste Wahl, wenn beispielsweise das primäre Rechenzentrum ausfällt. Reelle Beispiele aus der Praxis lehrten aber den Autor allerdings, dass Murphy's Law existiert, im Fehlerfall auch die Hochverfügbarkeitslösung versagen

kann und somit ein Unternehmen einen Plan B braucht, das Backup.

Ein weiterer Anwendungsfall für das Backup ist die Wiederherstellung der Datenbank nach logischen Korruptionen, etwa nach einem fehlerhaften Applikations-Update. Die Datenbank ist eigentlich noch funktionsfähig, jedoch ist der Inhalt (Daten) fehlerhaft. Hier kommt die klassische Wiederherstellung auf einen bestimmten Zeitpunkt zur Anwendung (Point-in-time-Recovery). Das Klonen von Datenbanken kann ebenfalls auf RMAN-Sicherungen basieren.

Für das Duplizieren von Datenbanken können als Beispiele zwei Gründe ausgemacht werden: Erstens werden produktive Daten in die Entwicklungsumgebungen geklont oder zweitens wird ein bestimmter Datenstand der Produktion geklont zwecks Fehler-Analyse. Letzteres kann sehr hilfreich sein, indem ein Klon vor einer bestimmten Fehler-Situation erstellt wird und die Datenstände verglichen werden können.

Möglichkeiten zur Backup-Optimierung mit Oracle-Bordmitteln

Oracle bietet von Haus aus schon seit Längerem diverse Funktionalitäten, um Backup- und Restore-Vorgänge zu optimieren. Inkrementelle Sicherungskonzepte helfen bei der Reduktion des Backup-Volumens (Output) und belasten somit den Backup-Storage weniger. Die gesamte Datenbank wird somit nicht mehr täglich gesichert, sondern nur alle paar Tage.

In der Oracle Enterprise Edition steht zusätzlich die Funktionalität «Block Change Tracking» zur Verfügung. Damit lässt sich auch das Input-Volumen, also die Lese-Operationen, massiv verringern, indem bei inkrementellen Backup-Operationen nur die effektiv geänderten Blöcke gelesen werden anstatt der gesamten Datenbank.

Weitere Optimierungen beim Backup-Output sind durch Komprimierung möglich, die direkt in RMAN eingebaut ist. Jedoch ist beim Aktivieren der Komprimierung darauf zu achten, dass je nach verwendetem Algorithmus die Lizenz für die Advanced-Compression-Option zu erwerben ist. Lediglich die Komprimierungsstufe „BASIC“ ist in Standard Edition und Enterprise Edition ohne Folgekosten verfügbar. Die Stufen „LOW“, „MEDIUM“ und

HIGH“ können ausschließlich für die Enterprise Edition als Option erworben werden.

Für die zentrale Steuerung, Konfiguration, Überwachung und Verwaltung der Backups bietet sich Oracle Enterprise Manager an. Daneben gibt es auf dem Markt diverse Tools zur zentralen Steuerung, jedoch sind diese bei der Verwaltung der Backups häufig limitiert und dem DBA eher fremd.

Letztendlich ist bei zentralisierten Lösungen die Verfügbarkeit sehr wichtig, denn bei den meisten Lösungen muss der zentrale Management-Server verfügbar sein, damit die Agenten die Sicherungsaufgaben ausführen. Ein längerer Stillstand des Management-Servers kann somit zum Datenbank-Stillstand führen (wie Archiver Stuck) und stellt einen Single-Point-of-Failure dar.

Im Bereich der Restore- und Volumen-Optimierung bietet Oracle die Zero Data Loss Recovery Appliance (ZDLRA) an. Dieses Engineered System verringert das Backup-Volumen, indem lediglich ein initiales Komplett-Backup erforderlich ist. Danach

reichen ausschließlich inkrementelle Sicherungen aus. Basierend auf der initialen Komplettsicherung und den nachfolgenden Änderungen kann sich die Appliance über die logische Verknüpfung der Datenbank-Blöcke auf jeden Zeitpunkt innerhalb der Aufbewahrungsdauer ein „Incremental Level 0“-Backup zusammenbauen. Dieses logische Gebilde wird gegenüber RMAN präsentiert, wodurch jede Wiederherstellung ohne inkrementelle Sicherungen auskommt. Insbesondere ist auch die Recovery-Dauer mit Archive-Log-Dateien sehr kurz, weil die logische Komplettsicherung der Appliance sehr nahe am gewünschten Recovery-Zeitpunkt liegt. Dieser Umstand schlägt sich in einer kürzeren Wiederherstellungsdauer spürbar nieder. Zudem validiert die Appliance im Hintergrund kontinuierlich die Konsistenz der Blöcke und entdeckt so Korruptionen frühzeitig. Natürlich lässt sich eine ZDLRA äußerst performant via InfiniBand anbinden und wirkt somit auch auf Netzwerk-Ebene potenziellen Engpässen entgegen.

Die intelligente und zentrale Backup-Steuerung

Trotz all der hilfreichen technischen Funktionen wurde bisher eine Komponente zur Optimierung nicht betrachtet, nämlich die Steuerung der Backup-Vorgänge. Fragen Sie sich einmal: Wann tanke ich mein Auto? Jeden Montag exakt um sieben Uhr oder wenn der Tankinhalt zur Neige geht. Wohl eher das Letztere oder mit anderen Worten: Sie tanken bedarfsgerecht, wenn es nötig wird. Genauso verhält sich eine moderne Backup-Steuerung, die auf starre Sicherungs-Intervalle verzichtet und stattdessen anhand von Richtlinien entscheidet, wann ein Backup erfolgen muss. Für diese auf Richtlinien basierende Entscheidung braucht die zentrale Steuerung einen intelligenten Algorithmus. *Abbildung 1* zeigt grafisch die Architektur einer möglichen und bewährten Implementation, die spürbar die Anzahl der Backup-Vorgänge und somit die Gesamtlast reduziert.

Cloud oder On-Premise? Mit Expertise ans Ziel.



Wann ist es Zeit für Cloud? Was bedeutet das für Ihre IT? Profitieren auch Sie vom umfassenden Know-how unserer Experten bei dieser Entscheidung. Erreichen Sie Ihre Ziele mit dbi services.

Phone +41 32 422 96 00 · Basel · Nyon · Zürich · dbi-services.com



Infrastructure at your Service.



Auf jedem Datenbank-Server läuft ein Agent, der kontinuierlich die Situation zur System- und Datenbank-Auslastung ermittelt und an das zentrale Repository sendet. Darunter fallen beispielsweise der CPU-Verbrauch, die I/O-Belastung, das Transaktions-Volumen, der Füllgrad der Archive-Log-Destination, die Rolle der Datenbank (Primary oder Standby) etc.

Im Repository selbst läuft in kurzen Abständen ein Algorithmus in Form eines Datenbank-Jobs, der aufgrund der aktuellen Situation und der definierten Richtlinien für eine bestimmte Datenbank entscheidet, ob ein Backup erfolgen soll oder nicht. Entsprechende Backup-Jobs werden erstellt, vom Agenten abgeholt und ausgeführt. Abschließend meldet der Agent den Status über das Backup an das Repository zurück und der Kreislauf schließt sich.

Jede Datenbank ist einer bestimmten Richtlinie zugewiesen und somit gilt es, initial zu überlegen, wie die Datenbanken kategorisiert werden sollen (Produktion, Test, DWH etc.). Mit anderen Worten widerspiegelt eine Richtlinie letztendlich die Anforderungen an eine bestimmte Kategorie. Nachfolgend werden abschließend ein paar Anforderungen genannt inklusive eines Beispiels, wie sich diese auf die Backup-Steuerung auswirken:

- **Zeitfenster für die Sicherung**
„Incremental Level 0“-Backup-Operationen sollen nicht zu Geschäftszeiten laufen. Deshalb definiert die Richtlinie ein erlaubtes Backup-Zeitfenster über die Nacht.
- **Transaktionsvolumen**
Es werden Schwellwerte dafür definiert, dass beispielsweise nach 30 GB Transaktionsvolumen beziehungsweise bei einer Änderungsrate von 40 Prozent ein komplettes oder ein inkrementelles Backup erfolgen soll.
- **Füllgrad der Archive-Log-Destination**
Beim Überschreiten eines Schwellwerts in Prozent wird eine Archive-Log-Sicherung angestoßen. Somit kann diese Lösung dynamisch auf ein verändertes Lastverhalten reagieren, indem die Sicherungs-Intervalle temporär verkürzt werden und dadurch einen potenziellen Stillstand durch einen Archiver-Stück verhindern.

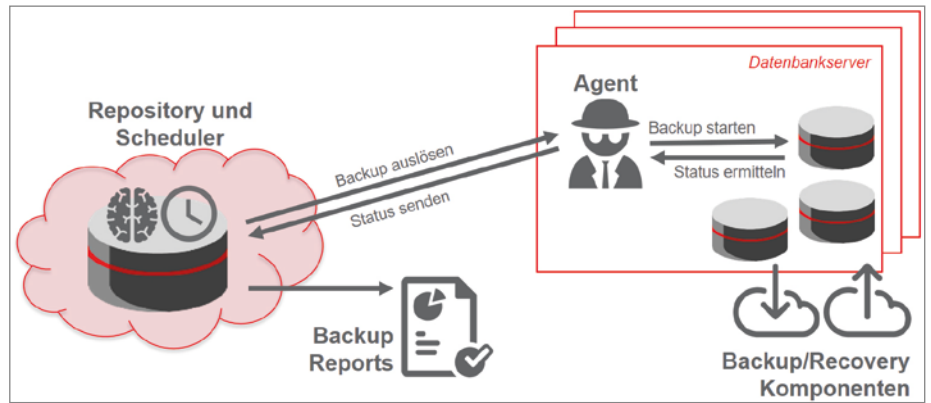


Abbildung 1: Architektur einer intelligenten Backup-Steuerung

- **CPU- und I/O-Last auf RAC-Nodes**
Durch eine maximale Obergrenze der CPU-Auslastung kann ein Backup komplett unterbunden werden, um ein System nicht vollends in die Knie zu zwingen. Zudem sollte im RAC-Umfeld dynamisch jener Knoten mit der geringsten Last für das Backup gewählt werden.
- **Recovery Point Objective (RPO)**
Die Vorgabe des maximal akzeptierten Datenverlusts definiert für den Scheduler implizit die maximale Intervalllänge zwischen zwei Archive-Log-Backups.

Eine intelligente Backup-Steuerung entscheidet aber nicht nur aufgrund der Anforderungen einer Richtlinie, sondern plant die Sicherungen innerhalb des konfigurierten Zeitfensters optimal ein. Dank dieser Verteilung der Backup-Operationen werden Performance-Engpässe vermieden und die Backup-bedingte Last ist ausgeglichener ohne massive Schwankungen.

Neben dem Herzstück der intelligenten Steuerung sollten bei der Implementierung der Architektur in *Abbildung 1* weitere Punkte beachtet werden. Der Agent muss in der Lage sein, bei einem temporären Ausfall der zentralen Steuerung eigenständig weiter zu sichern, indem er sich lokal die Sicherungs-Intervalle der letzten Tage merkt und gleich weiterfährt. Dadurch wird einem Datenbank-Stillstand durch einen Archiver-Stück vorgebeugt und das Risiko des Single Point of Failure eliminiert.

Ebenfalls sollten neue Datenbanken durch den Agenten automatisch erkannt, zumindest einer Standard-Richtlinie zugewiesen und somit gesichert werden. Die Praxis zeigt, dass es noch viele Erweiterungen gibt wie beispielsweise eine Schnitt-

stelle zu Enterprise Manager für die Alarmierung oder das Ausführen alternativer Aktionen im Fehlerfall („Corrective Actions“).

Der zentrale Ansatz bietet insbesondere im Bereich der Backup-Verwaltung riesige Vorteile, indem der DBA mit einem einzelnen Kommando beliebige Server/Datenbanken in den Wartungsmodus setzen kann. Auch im Sinne der Transparenz lässt die Lösung ihre Muskeln spielen, denn sämtliche Informationen rund um das Backup sind im Repository zentral gespeichert und können für Auswertungszwecke weiterverwendet werden.

Fazit

Die Praxis und die Erfahrung zeigen, dass die Nutzung von Oracle-Funktionalität bereits einiges an Optimierung ermöglicht. Doch erst die bedarfsgerechte und intelligente Steuerung der Backup-Operationen bringt die entscheidenden Mehrwerte in den Bereichen „Effizienz“, „Risiko-Minimierung“ und „Transparenz“ zum Vorschein. Trivadis bietet mit dem Produkt „TVD-Backup“ eine umfassende Backup-Lösung an, die mit der Komponente „Trivadis Intelligent Backup“ (TIB) genau den erwähnten Mehrwert bringt.



Roland Stirnimann
roland.stirnimann@trivadis.com