TSM intern

Ein vertiefender Einblick in das Backup-System

Die Lektüre dieses Artikels ist für die erfolgreiche Anwendung der Datensicherungssoftware TSM nicht notwendig. Er soll aber dem interessierten Anwender einen ersten Einblick in die Arbeitsweise des Systems geben und Verständnis wecken für ein paar Details in TSM, die vielen Anwendern sonst unverständlich erscheinen mögen.

Der IBM Tivoli Storage Manager TSM ist eine Client-Server-Software, die das Backup (Sicherung vor Datenverlust) und die Archivierung (langfristiges Aufheben von Daten ohne lokalen Plattenplatz zu belegen) ermöglicht. Mit TSM-Client und TSM-Server wird häufig auch die Hardware bzw. die Kombination von Hard- und Software bezeichnet, die die genannten Funktionen bereitstellt.

Die Client-Software muss auf jedem Rechner installiert sein, dessen Daten gesichert werden sollen. Sie stellt eine graphische Benutzeroberfläche und ein Kommando-zeilen-Interface zur Bedienung zur Verfügung, und sie vermittelt bei der Sicherung und beim Zurückladen die Daten zwischen dem zu sichernden Rechner und dem TSM -Server. Sie kann die zu sichernden Daten vor der Übertragung zum Server verschlüsseln. Sie ist aber insbesondere an der Verwaltung der gesicherten Daten nicht beteiligt.

Die Trennung in Client und Server vereinfacht auch die Bedienung und Administration des Systems: Der Anwender oder Klienten-Administrator muss sich z.B. nicht darum kümmern, dass der Server stabil und genügend performant läuft oder dass genügend freie Bänder vorhanden sind. Und der Server-Administrator muss sich nicht um die Sicherung und das ggf. notwendige Zurückladen der Daten von vielen Klienten kümmern, er kann sich auf den stabilen Betrieb des Servers konzentrieren.

Der TSM-Server hat zwei wesentliche Komponenten: ein sicheres Speichermedium zum Aufbewahren der Daten und eine Datenbank zur Verwaltung der gesicherten Daten.

Der TSM-Server des HRZ speichert die Daten in einer Tape-Library. Das ist ein großer Schrank mit vielen Regalfächern für Magnetbandkassetten, acht Magnetband-Laufwerken und einem vom TSM-Server gesteuerten Roboter-Greifarm, der die Kassetten zwischen den Laufwerken und den Lagerplätzen bewegt. Um die Datensicherheit zu erhöhen, kann TSM jede Datei mehrfach, ggf. auch an unterschiedlichen Orten speichern. Das HRZ der TU Darmstadt sichert die Zweitkopie der Daten auf dem TSM-Server der Uni Frankfurt. Umgekehrt liegt die Frankfurter Zweitkopie in Darmstadt.

Abb. 1:
Blick in die geöffnete Tape-Library des HRZ.
Der Robotergreifarm befindet sich in Ruhestellung. Knapp 10 Prozent der Bandkassetten sind sichtbar.

Damit mehr Klienten gleichzeitig Daten sichern können als der Server Bandlaufwerke hat, werden die auf dem Server eintreffenden Daten zunächst in einem großen Platten-Cache - wegen der Daten-Sicherheit sind das RAID-Systeme - zwischengespeichert. Falls eine Datei kurz nach der Sicherung zurückgeladen werden muss, kann das ebenfalls aus dem Cache geschehen und ist somit deutlich schneller.

Die wichtigste Komponente des TSM-Servers ist aber seine Datenbank. Sie enthält neben diversen Verwaltungsinformationen vor allem die Informationen über die gesicherten Dateien: welche Datei, von welchen Klienten, auf welcher Bandkassette, wo auf der Kassette, Zweitkopie, wann gesichert, ggf. wann gelöscht, Eigentümer, Zugriffsrechte und anderes mehr. Da die TSM-Datenbank extrem wichtig ist, wird sie häufig gesichert und an verschiedenen Orten bereitgehalten.

Die Sicherung der Daten kann vom Klienten oder vom Server aus angestoßen werden. Zunächst schickt der Server dem Klienten alles, was er über dessen bereits gesicherte Dateien weiß. Der Klient vergleicht diese Information mit den im Augenblick auf der Festplatte vorhandenen Dateien. Das Ergebnis des Vergleichs wird zum Server geschickt. Dabei können unterschiedliche Fälle auftreten:

• Eine Datei ist neu, sie wurde noch nicht gesichert.

• Die Datei wurde schon gesichert und liegt in identischer Version auf Klient und Server.

• Die Datei ist auf Klient und Server vorhanden, aber sie wurde seit der letzten Sicherung verändert.

• eine Datei wurde schon gesichert, inzwischen aber auf dem Klienten wieder gelöscht (absichtlich oder unbeabsichtigt)

Abhängig vom Ergebnis des Vergleichs kann nun Verschiedenes passieren:

• Die Datei selbst wird zum Server übertragen.

• Alte Versionen der Datei auf dem Server werden gelöscht.

• Der zur Datei gehörende Datenbank-Eintrag auf dem Server wird gelöscht.

• Es passiert nichts weiter.

Was davon tatsächlich abläuft, hängt von Sicherungsregeln ab. Diese Regeln sind der vielleicht am schwierigsten zu verstehende Teil des TSM. Sie geben an, wie lange wie viele unterschiedliche Versionen einer Datei aufbewahrt werden sollen. Die Regeln sind in unterschiedliche Policy Domains gruppiert. Jeder Klient ist einer Policy Domain zugeordnet, und kann nur deren Regeln verwenden.

Innerhalb einer Policy Domain gibt es einen oder mehrere Regel-Sätze, Management Classes genannt, Auf dem Klienten kann (prinzipiell) für jede einzelne Datei festgelegt werden, ob sie überhaupt gesichert wird und in welcher Management Class sie verwaltet wird, d.h. wie viele (alte) Versionen der Datei auf dem Server vorgehalten werden und für wie lange. Normaler-weise wird man aber alle Dateien in einer Management Class verwalten, und nur Sonder-fälle anders behandeln.

Die Kopie einer aktuellen Datei des Klienten auf dem TSM-Server heißt aktive Version, alle älteren Versionen heißen inaktive Version. Die aktive Version einer Datei wird auf dem Server nie gelöscht. Wenn eine Datei auf dem Clienten gelöscht wird, dann wird deren Kopie im TSM-Server ebenfalls inaktiv. Dies geschieht allerdings erst bei der nächsten Sicherung, vorher weiß der Server nichts davon.

Die Regeln einer Management Class sind durch zwei Werte-Paare charakterisiert: verexists und retextra, sowie verdeleted und retonly. Wenn eine aktive Version einer gesicherten Datei existiert, dann gelten verexists und retextra. Verexists gibt an, wie viele Versionen einer Datei mit aktiver Version auf dem TSM-Server aufgehoben werden. Retextra gibt an, wie viele Tage die inaktiven Versionen einer Datei auf dem Server höchstens aufgehoben werden, von denen eine aktive Version existiert. Wenn von einer Datei keine aktive Version mehr auf dem TSM-Server existiert, dann gelten verdeleted und retonly: verdeleted bestimmt wie viel alte Versionen aufgehoben werden, retonly sagt, wie viele Tage die letzte Version einer inaktiven (d.h. gelöschten) Datei auf dem Server maximal aufbewahrt wird. Vergleichbare Regeln gibt es auch für die Archiv-Funktion des TSM-Servers.

Ein Beispiel: Die Policy Domain CAMPUS auf dem TSM-Server des HRZ wird normalerweise für zu sichernde Arbeitsplatzrechner in den Instituten verwendet. Dort gibt es unter anderem eine Management Class STANDARD mit den Werten verexists=3, retextra=30 und verdeleted=1, retonly=60. Solange eine Datei regelmäßig bearbeitet und gesichert wird, liegt die durch verexists angegebene Anzahl (hier also: 3) an letzten Versionen auf dem Server. Nachdem die Datei gelöscht wurde, nur noch verdeleted Versionen. Nach retonly Tagen wird auch die letzte Version auf dem TSM-Server gelöscht.

Alle derzeit verfügbaren Policy Domains des TSM-Servers des HRZ mit ihren Management-Klassen sind unter http://www.hrz.tu-darmstadt.de/tsm/poldom.tud dokumentiert.

Die genaue Abfolge der Ereignisse dieses Beispiels ist in folgender Tabelle dargestellt:

Tag	Aktion auf dem Client	Aktion auf dem TSM-Server	Kopien im TSM-Server
1. Tag	Datei wird erzeugt (Version A) und gesichert		Version A ist active
2. Tag	Datei wird geändert und gesichert (Version B)		Version A ist inactive Version B ist active
3.Tag	Datei wird geändert und gesichert (Version C)		Version A ist inactive Version B ist inactive Version C ist active
4. Tag	Datei wird geändert und gesichert (Version D)	Version A wird auf dem Server gelöscht wg. verexists=3	Version B ist inactive Version C ist inactive Version D ist active
5. Tag	Datei wird gelöscht, danach Sicherung	Version B und C wird gelöscht wg. verdeleted=1	Version D ist inactive
6. Tag	Sicherung. Datei existiert nicht mehr auf dem Klienten		Version D ist inactive
...	Sicherung. Datei existiert nicht mehr auf dem Klienten		Version D ist inactive
64. Tag	Sicherung. Datei existiert nicht mehr auf dem Klienten	Version D wird gelöscht wg. retonly=60	–

Das beschriebene Vorgehen wird als „incremental forever“ oder „progressive“ bezeichnet. Es unterscheidet sich deutlich vom Generationen-Backup, das viele andere Datensicherungs-Programme verwenden. Diese führen regelmäßig (z.B. wöchentlich oder monatlich) full backups durch, d.h. es werden grundsätzlich alle vorhandenen Dateien gesichert. Dazwischen werden nur die Änderungen in differential backups gesichert. Sehr häufig haben solche Systeme keine Datenbank (Inhaltsverzeichnis), sondern der Umfang dieser Datensicherung ist nur durch den Inhalt der Bandkassette definiert.

Das TSM-Verfahren ist zwar ungewöhnlich, hat aber Vorteile: Jede Datei wird nur einmal gesichert. Das spart die Zeit und Netz-Bandbreite des regelmäßigen full backups und Platz auf dem TSM-Server (und damit Kosten). Bei einem ggf. nötigen Zurückladen muss jede Datei nur einmal übertragen werden. Bei einem einfachen Generationen-Backup ohne Datenbank werden dagegen alle Versionen einer Datei zurückgeladen: Zunächst der full backup, und dann die differentials in ihrer zeitlichen Reihenfolge. Dadurch werden auch beim Restore Dateien mehrfach übertragen und auf dem Klienten durch neuere Versionen wieder überschrieben. Auch wenn es auf den ersten Blick komplizierter erscheint, ist das Incremental-forever-Verfahren des TSM also ressourcenschonender.

Dr. Norbert Conrad