

Michael Pietroforte

Backup-Software

Konzepte-Bewertung-Vergleich
Version 1.0

Kapitel Offene Dateien

<http://www.pietroforte.de>

Offene Dateien

Die größte Herausforderung jeder Backup-Software ist die Sicherung von Dateien, die bereits von anderen Applikationen für den schreibenden Zugriff geöffnet sind. Wird eine Datei gesichert, während eine Anwendung noch in diese Datei schreibt, kann es zu Inkonsistenzen kommen, da eventuell einige neue Datenblöcke nicht mehr mitgesichert werden. In den meisten Fällen ist die gesicherte Datei dann vollkommen unbrauchbar.

Um dies zu vermeiden, gibt es verschiedene Strategien. Die einfachste ist sicher, alle Applikationen vor dem Start eines Backups zu beenden. Da es für viele Anwendungen aber erforderlich ist, rund um die Uhr im Einsatz zu bleiben, kommt diese Lösung meist nicht in Frage. Hinzu kommt, dass das Betriebssystem selbst ständig eigene Dateien geöffnet hält, die ebenfalls gesichert werden sollen.

Im Prinzip gibt es zwei grundsätzlich zu unterscheidende technische Lösungsansätze für das Problem. Bei der zuverlässigeren der beiden Methoden stellt die entsprechende Applikation eine spezielle Schnittstelle für Backup-Programme zur Verfügung. Die Anwendung wird von der Backup-Software informiert, dass jetzt eine Sicherung stattfindet, woraufhin diese keine Schreibzugriffe währenddessen auf die entsprechende Datei durchführt.

Einige Datenbanksysteme und Groupware-Anwendungen stellen solche Schnittstellen für den externen Zugriff zur Verfügung. Diese sind aber als Option gesondert zu erwerben, falls die Backup-Software sie unterstützt. Näheres hierzu erfahren Sie im Abschnitt über die Systemunterstützung.

Für alle anderen Anwendungen kommt die zweite Herangehensweise zum Einsatz. Die Backup-Software versucht, geöffnete Dateien in einem konsistenten Zustand zu sichern, ohne dass die entsprechende Applikation davon etwas mitbekommt. Hierfür wurden wiederum eine ganze Reihe von Strategien entwickelt, die je nach Anwendung mehr oder weniger effektiv sind. Die einfacheren Lösungen sind meist im Standardpaket der Backup-Software enthalten, für die ausgefeilteren ist eine zusätzliche Option zu erwerben.

Timeout

Eine sehr zuverlässige Methode, um Inkonsistenzen bei der Sicherung in einer Datei zu vermeiden, besteht darin, einfach eine vorgegebene Zeitspanne zu warten, ob die Applikation die Datei wieder schließt. Freilich ist das nicht besonders effektiv, denn manche Anwendungen wie beispielsweise Datenbanksysteme schließen zur Laufzeit einige ihrer Dateien überhaupt nicht. Außerdem wird der Backup-Prozess verlangsamt, da die Backup-Software immer eine Ruhepause einlegt, wenn sie auf eine offene Datei stößt.

Neuversuch

Etwas effektiver ist es da, wenn der Backup-Job zunächst weiterläuft und zu einem späteren Zeitpunkt, zum Beispiel wenn alle anderen Dateien gesichert sind, einen erneuten Versuch startet.

Schreibschutz

Bei Anwendungen, die hauptsächlich lesend auf ihre Dateien zugreifen ist unter Umständen das Verhängen eines Schreibschutzes von Seiten der Backup-Software die Lösung. Voraussetzung für die Anwendung dieser Methode ist allerdings, dass die Datei zum Zeitpunkt der Sicherung nicht schon durch die Anwendung geöffnet ist.

Sperrung

Sollte die Datei bereits geöffnet sein, besteht die Möglichkeit, die Datei für den Zugriff komplett zu sperren. Sobald die Sicherung abgeschlossen ist, wird die Sperre wieder entfernt. Unproblematisch ist dieses Vorgehen freilich nicht, denn so manche Anwendung reagiert mitunter recht unwirsch, wenn ihr der Zugriff auf die eigenen Dateien verwehrt wird.

Ohne Sperrung

Falls die Applikation nicht ihrerseits den Fremdzugriff gesperrt hat, kann die Backup-Software parallel zur Anwendung eine Sicherung starten. Findet währenddessen ein Schreibzugriff statt, kommt es zu besagten Inkonsistenzen, und das Backup ist unbrauchbar.

Schattenkopie

Die oben beschriebenen Methoden kranken entweder daran, dass nicht sichergestellt ist, ob jede offene Datei tatsächlich in einer in sich konsistenten Form gesichert wird oder gefährden unter Umständen sogar die Systemsicherheit. Das Problem ist offenbar, dass zwei Anwendungen gleichzeitigen Zugriff auf eine Datei benötigen. Eine etwas raffiniertere Methode soll hier Abhilfe schaffen. Anstatt sich mit der jeweiligen Applikation um Dateien zu zanken, wird eine Kopie der geöffneten Datei auf der Festplatte angelegt, die die Backup-Software dann problemlos sichert. Scheinbar entsteht dabei aber das gleiche Problem, schließlich muss das Backup-Programm wiederum auf eine bereits geöffnete Datei zugreifen.

Der Trick bei diesem Verfahren ist, dass die Backup-Software vor dem Anlegen ein gewisses Zeitintervall abwartet, ob Schreibvorgänge stattgefunden haben. Falls nicht, kann davon ausgegangen werden, dass sich die Datei, obwohl sie weiterhin geöffnet ist, in einem konsistenten Zustand befindet. Das Anlegen der Schattenkopie geht schnell vonstatten, da es auf der Festplatte des gleichen Systems stattfindet. Während die Backup-Software anschließend in aller Ruhe die Kopie sichert, kann die Applikation bereits wieder in das Original schreiben. Nach erfolgreicher Sicherung löscht das Backup-Programm die Kopie und der entsprechende Platz auf der Festplatte wird wieder freigegeben.

Gruppenkopie

Mit diesem Verfahren bekommt man noch ein weiteres Problem bezüglich offener Dateien in den Griff. Manche Applikationen halten gleichzeitig mehrere Dateien geöffnet, deren Inhalte logisch miteinander verknüpft sind. Selbst wenn jede einzelne Datei in sich konsistent gesichert wurde, ist damit noch nicht gewährleistet, dass die Dateien untereinander konsistent sind. Beim Anlegen der Schattenkopien muss die Backup-Software darauf achten, dass dies für zusammengehörige Dateien gemeinsam erfolgt.

Feature \ Software	ARCserve v9	Backup Exec v9	Retrospect v6.5	TapeWare v7
Timeout	✓	✓	-	-
Neuversuch	✓	-	-	✓
Schreibschutz	✓	-	-	-
Sperrung	✓	✓	-	-
Ohne Sperrung	✓	✓	-	-
Schattenkopie/Gruppenkopie	<i>Backup Agent for Open Files</i>	<i>Advanced Open File Option</i>	<i>Open File Backup</i>	<i>Open File Manager</i>

Tabelle 1: Offene Dateien