THE EXPERT'S VOICE® IN DATA MANAGEMENT

Pro Data Backup and Recovery

Securing your information in the terabyte age

Steven Nelson

Apress®

Pro Data Backup and Recovery

Steven Nelson

Apress®

Pro Data Backup and Recovery

ISBN-13 (pbk): 978-1-4302-2662-8

ISBN-13 (electronic): 978-1-4302-2663-5

Printed and bound in the United States of America (POD)

Publisher and President: Paul Manning
Lead Editors: Frank Pohlmann and Michelle Lowman
Technical Reviewer: Russell Brown
Editorial Board: Steve Anglin, Mark Beckner, Ewan Buckingham, Gary Cornell, Jonathan Gennick, Jonathan Hassell, Michelle Lowman, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Frank Pohlmann, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Coordinating Editor: Mary Tobin
Copy Editor: Nancy Sixsmith
Compositor: MacPS, LLC
Indexer: Carol Burbo
Artist: April Milne
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media, LLC., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail `orders-ny@springer-sbm.com`, or visit `www.springeronline.com`.

For information on translations, please e-mail `rights@apress.com`, or visit `www.apress.com`.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at `www.apress.com/info/bulksales`.

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

To Elena, Christopher, and Tommy

Contents at a Glance

Contents

About the Author

■ **Steven Nelson** has almost 20 years of experience in all areas of System Administration, spanning both *NIX and Microsoft Windows operating systems, but he has spent the last 10 years focusing on backup, recovery, and storage architecture and administration. He has worked for major manufacturing, online retail, and financial services companies, as well as working as a consultant to leading cellular telecom and industry-leading storage manufacturers. Steven is currently heading up the design and deployment of global backup, recovery, and storage services for the Research and Development division of a large software developer.

About the Technical Reviewer

■ **Russell Brown** is a graduate of Texas A&M University, where he earned a B.S. degree in Agricultural Engineering, and the University of Colorado, where he earned a Masters degree in Business Administration. His work experience includes time as a Backup & Recovery Systems Administrator for a computer system manufacturer in Austin, Texas and a major entertainment studio in Burbank, California. Between systems administration roles, Russell spent four years flying around the United States while working as a Denver, Colorado-based professional services consultant specializing in backup and recovery. During the past five years, Russell has worked in a backup and recovery-focused systems engineering role in Los Angeles, California and Sydney, Australia. When Russell isn't working on backup and recovery, he enjoys travel, live music, automotive maintenance, and walking his dog along Sydney's northern beaches.

Acknowledgments

There is a myth about being an author: you sit at a computer, write an entire book, send it in, and *voila!* a book is published. Nothing could be further from the truth—writing a book, especially a technical book, is a team effort. There are many people involved in this process and all of them have played a vital role in getting the book completed.

To the team at Apress: Frank, you took a chance on an unproven author with an offbeat idea for a book, and who was working in environments where time was limited. Thank you for giving me the opportunity and pushing me to get things done. Mary and Michelle, my editors, you guys were great about giving me feedback and helping me along where you could. I appreciate all the support in writing this book.

A technical book is not possible without a great technical reviewer. Russell Brown, I could not have chosen a better TR. I cannot tell you how I appreciate all the time and effort you put into someone else's project and helped it to be so much better than it started. Thank you is not enough to say. When you decide to embark on this journey, I hope I can be as much help to you as you were to me.

Finally, I could not have done this without the support of my family. Chris and Tommy, I know that there have been lots of times when you guys wanted to do stuff with your dad and I was "working on the book." You guys have been really patient and understanding, and I love you more than you can know. Thank you for being the best sons a dad could ever ask for.

Elena, my lovely wife, you pushed me when I thought that I would never complete this, you supported me when I was down and celebrated with me on my accomplishments. You once told me that I would be very proud of myself when I finished this project. You were right. I love you with all my heart and cannot tell you how much I appreciate all the sacrifices you made for me over the years that allowed me to reach this point in my career and in my life.

CHAPTER 1

Introduction to Backup and Recovery

Who Should Read this Book?

Pro Data Backup and Recovery has come from the many views of people that I have interacted with during my career as a systems administrator, systems engineer, and consultant. This book is primarily geared toward the systems engineers and architects within an organization, but it will also be useful for the day-to-day functions of systems administrators and line managers of both teams. System administrators will find it useful for understanding the issues that are involved in protecting the data that they are responsible for providing on their systems, as well as helping to fashion systems and methods that will help protect that data against inevitable systems failures. Systems administrators can also use the information in this book to influence users of their systems to protect their data, create data structures that are easy to back up, and identify data that is most critical to be backed up and how it should be protected.

Line managers will find this book useful for understanding some of the technical trade-offs in data protection, helping them make better decisions regarding the recommendations that their systems engineers are making and system administrators are implementing for backup and recovery. The book will also help these line managers interact with the various vendors of backup products, giving the manager help to ask the hard questions and be able to answer them when their team asks them.

Backup systems have several characteristics:

- They are not inexpensive over the life cycle of the backup solution
- Large amounts of resources can be consumed in terms of:
    - People time
    - Operational and capital expenses

Additionally, the value of backup systems is difficult to express in terms of direct organizational mission contribution. This book will help the line manager show that the organization’s data is being protected based on the criticality of the data, the cost of the backup platform, and the availability of the data for recovery.

Pro Data Backup and Recovery is primarily for systems engineers and architects (and administrators, in many cases) who are responsible for the design, implementation, and operation of backup and recovery systems. Backup is the one of those “invisible” jobs in systems—if people know who you are it is because bad things have happened. The main goal of this book is to help those who are in this sometimes thankless role to design, modify, and optimize your backup systems; to make those times where you are visible as short as possible; and to give you the tools to help make the recoveries successful. Within the pages of this book, you will find various configurations of both hardware and software that will allow you to build or upgrade backup systems to grow and meet the changing needs of your organization. Although these configurations can be applied to many different brands of backup software, this book focuses only on the two major backup vendors: Symantec NetBackup and CommVault Simpana. These two vendors represent similar approaches to performing backups, but for different customer organizational sizes.

What this book is not is a tutorial on the specific commands and day-to-day operational functions that are executed directly by system administrators. I make some assumptions about the familiarity of engineers and/or architects with the backup software being used with regard to commands and options. This book is more concerned with the "why" of using various components as well as the "how" of putting them together, but not with specific command sets used to do it. There are command examples within this book as necessary to illustrate particular use cases, but there is an assumption that the commands used will already be familiar to the reader.

Backup and Recovery Concepts

Backup and recovery is a topic that might seem basic at first glance, but it seems to be a little confusing to many people. Backups and archives tend to be used interchangeably, representing some type of data protection that spans a period of time. Adding to the confusion is the fact that many organizations group the functions together in a single group, with the emphasis more on the data backup side, thus giving the illusion of being a single function. Let's look at this in a little more detail to get a common language and understanding of the functions and roles of both backups and archives.

■ **Note** Where the difference between backups and archives gets particularly confusing is when backups are stored for long periods of time, on the order of years. Such backups can be mistakenly referred to as *archives* because the data the backup contains might indeed be the only copy of the data in existence at any particular point in time. This is particularly common in organizations that contain both open systems (UNIX/Linux and Windows) and mainframe environments because of terminology differences between the two platforms.

Backups

Backups are snapshot copies of data taken at a particular point in time, stored in a globally common format, and tracked over some period of usefulness, with each subsequent copy of the data being maintained independently of the first. Multiple levels of backups can be created. *Full backups* represent a complete snapshot of the data that is intended to be protected. Full backups provide the baseline for all other levels of backup.

In addition, two different levels of backups capture changes relative to the full backup. The *differential backup*, also known as the *cumulative incremental backup*, captures backups that have occurred since the last full backup. This type of backup is typically used in environments that do not have a lot of change.

The differential backup (see Figure 1–1) must be used with care because it can grow quickly to match or exceed the size of the original full backup. Consider the following: An environment has 20 TB of data to back up. Each day 5 percent or 1 TB of data changes in the environment. Assuming that this is a traditional backup environment, if a differential backup methodology is used, the first day 1TB of data is backed up (the first day's change rate against the previous full backup). The second day, 2 TB is backed up, and so on. By the end of 5 days, 5 TB of data is being backed up; by the end of 10 days, 10 TB might be being backed up; in 20 days, it could be a backup of 20 TB.

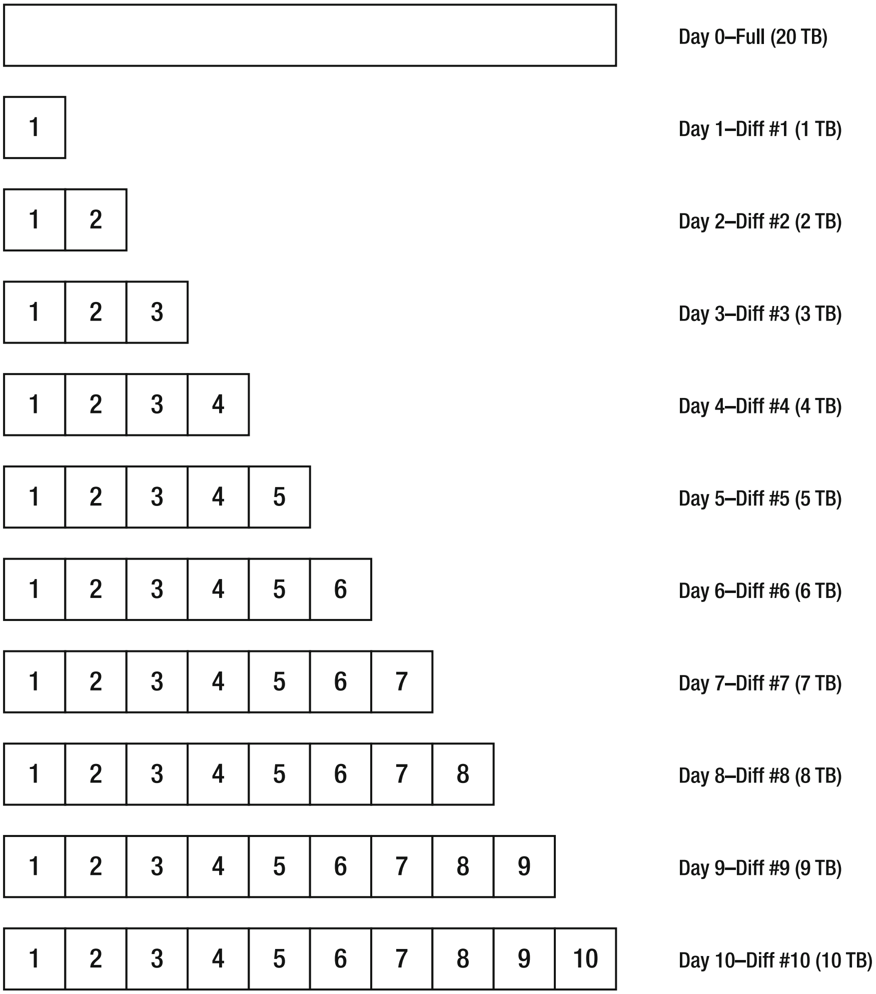


Figure 1–1. *Differential backups*

However, this 20 TB is not necessarily representative of a full backup. If the change rate represents a mixture of new files added to the respective clients as well as changes to existing data, the cumulative data backed up will not capture data that has not changed, even though the total amount of a full backup would still only incrementally change in size.

The advantage of using differential backups, especially during a restore, is the number of backup images required to perform the restore. A differential restore requires only the full backup plus the latest differential backup to complete the restore of any file in the backup image. Because there are only a limited number of images required, the probability of both images being lost, corrupted, or offsite is decreased significantly.

The second level of backup that captures images relative to the full backup is the *incremental backup* (see Figure 1–2). This backup captures changes that have occurred since the last backup of any type. Traditionally this is the most utilized form of backup in combination with a full backup. The incremental backup will contain the smallest amount of data required during a backup cycle, reducing the amount of data moved, and in general, the time required for a backup. Take the previous example of the 20 TB full backup. The first day after the full backup, the backup size would be 1 TB, the second day, 1 TB, and so on. The quantity of data backed up is only dependent on the difference between the previous backup and the current one, so the size (and proportionally the time required) for the backup is relatively consistent.

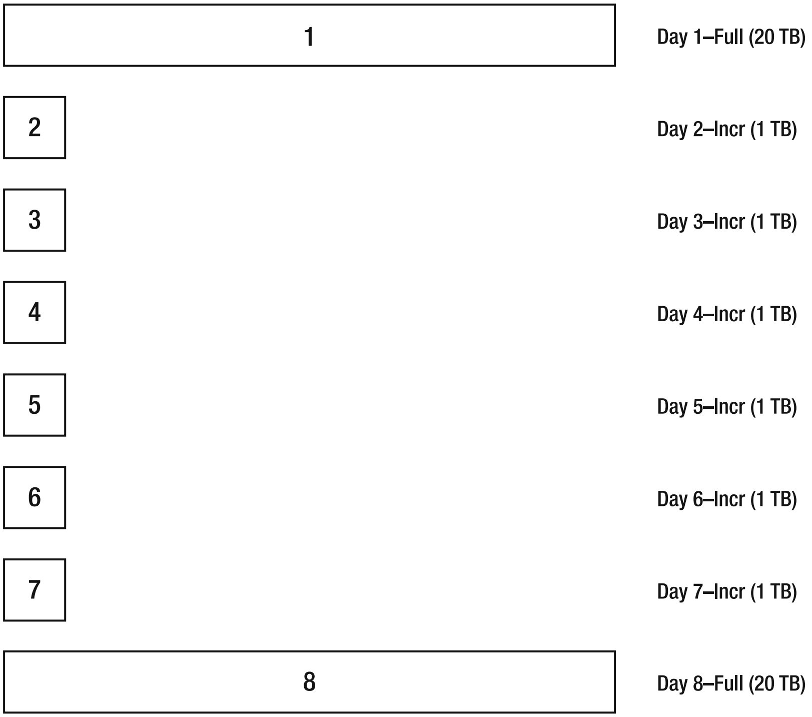


Figure 1–2. *Incremental backups*

There are some downsides to incremental backups. If you are recovering a set of files from a set of full and incremental backups, you will likely require more than two different backup images to complete the restore.

Suppose that four files need to be recovered: one has not changed since the full backup, one changed the first day, one changed the second day, and one changed the third day. To complete this restore, four images—the full backup and three different incremental images—are required to retrieve all the files needed (see Figure 1–3). In addition, because the images are relative to each other, some pieces of backup software will not allow parallel restores from the related sets of backup images, so the file recovers have to occur in serial fashion, each loading and unloading its own set of images for the restore. This can have a large impact on the time required for a restore.

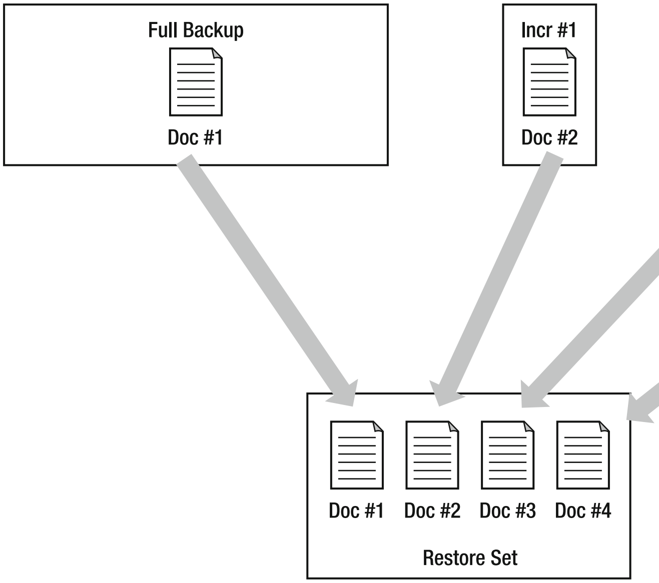


***Figure 1–3.** Multiple file incremental restore*

Also, because incremental backups capture the changes since the last full backup, the full set of backups can contain multiple copies of individual files. This can present problems during restore because it is too easy to select a copy of a file that shouldn't be restored.

Let's look at an example to illustrate this situation. Suppose that you have a file that contains the inventory for a particular division. This file gets updated on a daily basis, so it gets backed up daily. Three weeks later, a copy of the file needs to be restored, but now there are 21 copies to choose from. Which file is the one you want? Why 21 copies? Because the file gets updated each day, a new copy gets added to the backup index each day for 3 weeks or 21 days—thus 21 copies.

Another potential issue is the probability of loss or corruption. In many types of less sophisticated backup software, if a single incremental is corrupted or not available for recovery, the entire recovery will fail, even if the file is not located on that particular piece of media.

An alternate type of incremental backup is called the *level backup*. Each backup session is assigned a number, usually 0–9, with level 0 representing a full backup. Backups are executed relative to the number assigned, and a difference is taken against the next highest level of backup that was previously executed. Sound confusing? It can be, but it is a very powerful method of managing backups.

Consider the 20 TB example once again (see Figure 1–4). This time, a level 0 backup is executed against the 20 TB data set, capturing all 20 TB of data. The next backup executed is a level 1, which captures 1 TB of backup, representing the change since level 0 (the full backup). Now the next backup is a level 2 backup, which captures only the differences between the level 1 and the current backup—likely much smaller than a comparable level 1 backup. If another level 2 backup is executed as the next backup, the differences between the current data set and the last level 1 backup are captured. This type of backup can be much smaller than a comparable set of traditional incremental backups, resulting in short backup windows. However, the weaknesses of the incremental backup are also present in the level backup and need to be taken into consideration.

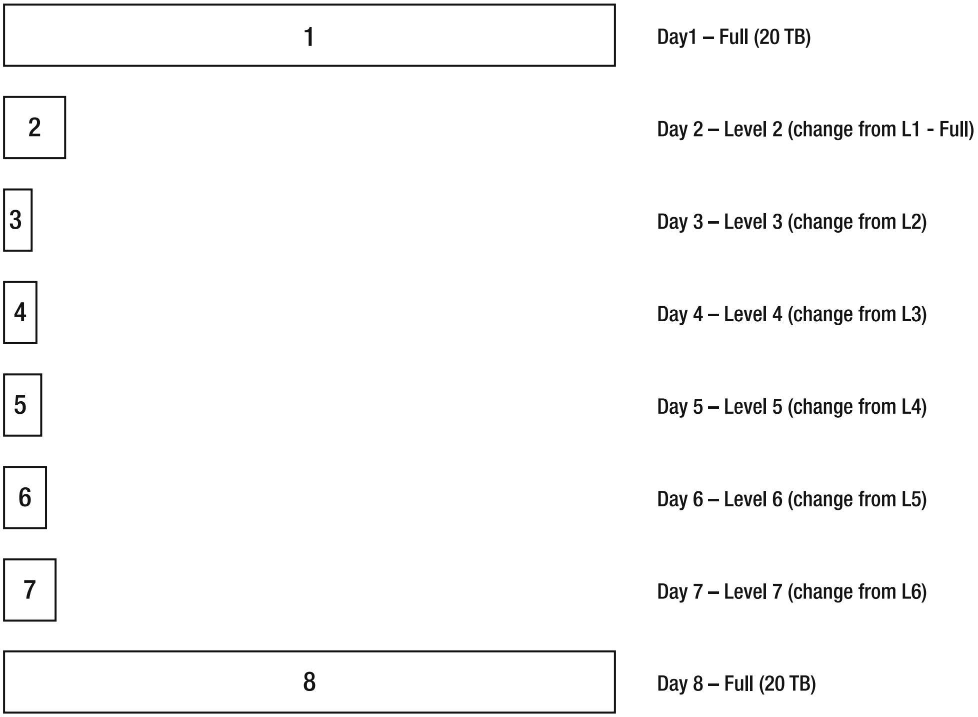

1
Day1 – Full (20 TB)
2
Day 2 – Level 2 (change from L1 - Full)
3
Day 3 – Level 3 (change from L2)
4
Day 4 – Level 4 (change from L3)
5
Day 5 – Level 5 (change from L4)
6
Day 6 – Level 6 (change from L5)
7
Day 7 – Level 7 (change from L6)
8
Day 8 – Full (20 TB)

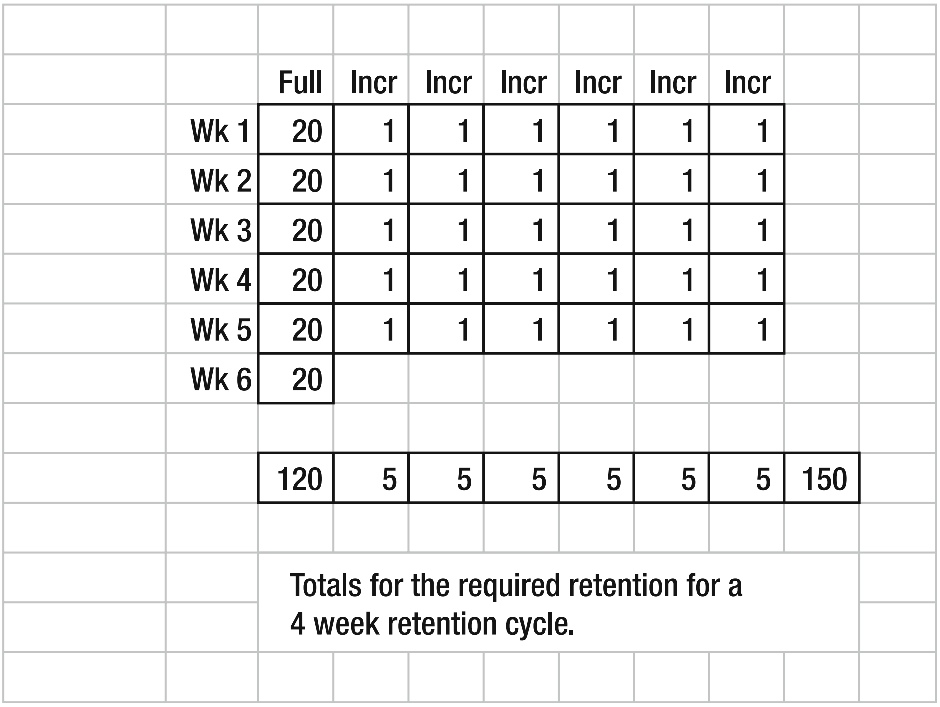
Figure 1–4. *Level-based backups*

Given the variety of backup levels, it is easy to see that multiple versions of individual data sets can exist within a particular set of backups. This feature allows for recovery of data to a particular point in time, complete with any changes in the specific data being recovered. This function exists because the backup software takes a complete copy of the data at the point of backup, regardless of the contents, usually based upon static information such as file names.

■ **Note** Many disk array vendors claim that the simple act of creating either a snapshot or clone of data using a piece of secondary storage constitutes a backup. However, this is not the case. Although single or even multiple copies of data can be made using these types of technologies, backups require tracking individual data sets. Technically, if the array- or file-based snapshots are identified by date of creation, and a manual record maintained all data sets and contents of the data sets that are protected, the set of snapshots could be considered backups. Pieces of backup software do maintain snapshot tracking, which then can be used as the recovery source—effectively making the snapshot a backup. EMC NetWorker is one example of backup software that has this capability.

Backups also quickly grow in total storage required. Take the 20 TB example you have been using and assume that there is a requirement to hold on to the backups for some period of time so that the data can be recovered at some point in the future. The period of time that the backups are required to be available is called the *backup retention period*. As the 20 TB data is repeatedly backed up with a combination of full and incremental backups, the total amount of data retained grows very quickly. This is not because of the addition of data to the system being backed up overall; it is strictly due to the number of copies of the same data that is stored repeatedly by both full and incremental backups.

To illustrate this, we will say that the organization has to keep a copy of this 20 TB of files and be able to retrieve a file that is 4 weeks old, relative to the current date—the backups must have 4 week retention. Also, assuming that weekly full and daily incremental backups are taken of this data, a minimum of 150 TB of backup storage media must be available to meet the four week requirement (see Figure 1–5).

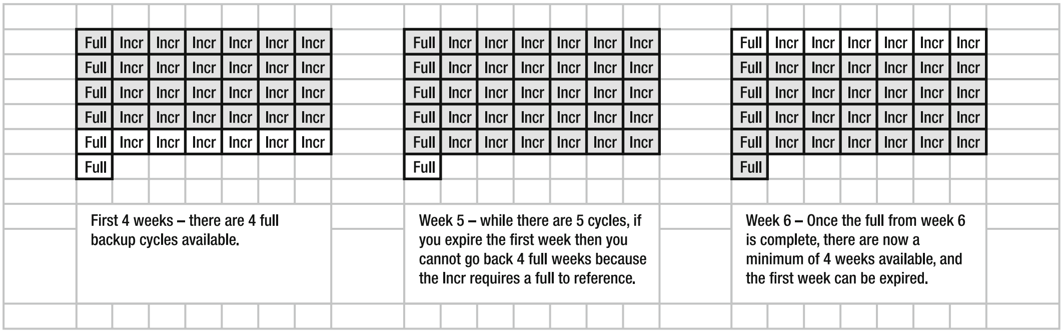


	Full	Incr	Incr	Incr	Incr	Incr	Incr	
Wk 1	20	1	1	1	1	1	1	
Wk 2	20	1	1	1	1	1	1	
Wk 3	20	1	1	1	1	1	1	
Wk 4	20	1	1	1	1	1	1	
Wk 5	20	1	1	1	1	1	1	
Wk 6	20							
	120	5	5	5	5	5	5	150

Totals for the required retention for a
4 week retention cycle.

Figure 1–5. *Required storage for a 20 TB backup*

■ **Note** A common mistake is to simply count the number of weeks to meet a retention requirement. For instance, retaining four weeks of full backups to meet a four-week retention requirement does not satisfy the requirement. Why not? Because the fourth week's backup has expired on week four, making it ineligible (or potentially unavailable) for recovery by the backup software. The fifth week is required to ensure that if the requirement is to go back four weeks, the oldest week will still be available, with the sixth week's full backup needed to fully release the first week and provide a full four weeks (see Figure 1–6).


Full Incr Incr Incr Incr Incr Incr
Full Incr Incr Incr Incr Incr Incr
Full Incr Incr Incr Incr Incr Incr
Full Incr Incr Incr Incr Incr Incr
Full Incr Incr Incr Incr Incr Incr
Full

First 4 weeks – there are 4 full backup cycles available.

Week 5 – while there are 5 cycles, if you expire the first week then you cannot go back 4 full weeks because the Incr requires a full to reference.

Week 6 – Once the full from week 6 is complete, there are now a minimum of 4 weeks available, and the first week can be expired.

Figure 1–6. *Backup pattern for four week retention*

Note that the driving force of the total size of the backup is not the backup size itself, but the number of copies of the same data, made repeatedly. This is particularly marked in backups of relatively static data (for example, government records such as U.S. Census statistics). These files are static once they are completed, but are backed up each time there is a full backup. So if the particular record is 10 TB in size, even if there are only 10 full backups retained, the single record represents 100 TB, not including any other records that are also in the set of records. The vast majority of data stored has not changed in these types of environments and has been copied in exactly the same form repeatedly, taking up unnecessary space and backup resources. Because of this tendency to grow quickly and without bounds, backups should be intended for short- or medium-term storage and used for long-term storage only when absolutely required—backups are intended to be restored.

A backup that has been stored for a very long time, on the scale of years, is probably not usable. Why not? Changes in hardware, operating systems, application software, database systems, and even backup software can make recovery exceedingly difficult, if not impossible in some cases. Backups should truly be considered for storage options only for no longer than three years. Beyond this point, changes in one or all of the variables discussed previously will most likely render the backup useless and/or unrecoverable. In order for data to be recovered over longer periods of time, the data must be put into a format that is universal and onto media that has a higher probability of accessibility in the future. This is the purpose and design of archives.

Archives

Let's postulate that an organization needs to maintain a copy of data for ten years and selects a backup to store the data. Let's also assume that the data is static—it will not change over the lifetime of the data—and that traditional full/weekly incremental backups are taken on the data. What has happened: there are now 520 copies of exactly the same data stored in a very large set of backups.

Archives, on the other hand, are not copies of the data; they are the actual original data that has moved from one location to another. This movement is based on criteria assigned to the data such as age, value, criticality, and so on. Archived data is tracked over time, much like backup data, but the data is unchanging and therefore only a single copy of the data exists at any one point in time. If data within the archive is modified, the archive can either treat the new data as a new entry in the archive or replace the existing entry, depending on the capabilities of the archive system and the data-protection policies assigned to the archive. Archives are intended to allow for long-term storage of data to media that are resistant to degradation over time.

Examples of the type of data that might have this type of general protection requirement are criminal records. These types of records need to be maintained indefinitely because they contain information that might assist law enforcement with solving additional crimes, provide clues to individual behavior, and so on. As such, these records are important to protect. However, simply backing them up, over and over, is a waste of resources and ultimately does not provide a long-term method of storage and recovery. An archive would remove the criminal data from primary storage and place it on storage specifically designed for static data. The archive could be either offline or nearline, depending on the requirements.

Although the focus of this book is not on archiving, it does warrant mention. Why? Archives are a powerful tool in reducing the amount of data that is backed up, thus an effective tool in reducing the backup windows required to perform a backup of a particular environment.

To understand this, let's expand the 20 TB example (see Figure 1–7). Within the 20 TB, assume that 5 TB of data is static and a candidate for archiving. An initial archive copy of the data is created on some piece of archive media, (disk storage, in this case). Once the 5 TB of data has been removed from the original 20 TB, the next time the backup occurs it will have to move only 15 TB of data in the same period of time. Assuming that the overall data transfer performance of the backup does not change, the backup will take only 75 percent of the time previously required to accomplish the original 20 TB backup.

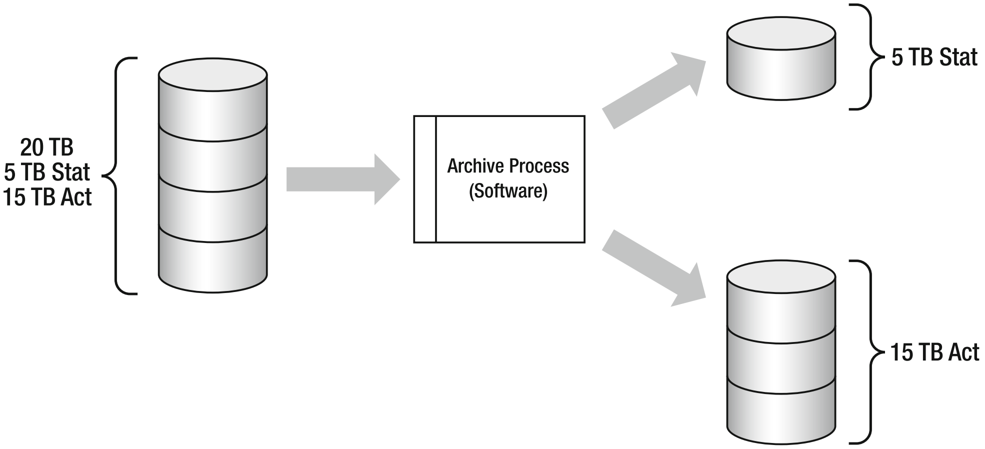


Figure 1–7. *Data reduction in an archive*

The original data can be referenced in one of two ways, depending on the method of creating the archive. The first method creates a copy of the data on static media and removes the original copy. Recalls of data are made through specific requests to the software that performed the archive, which is then recalled from the storage media and placed back on primary storage. This is what is known as an *offline archive* (the data is not directly accessible and is said to be offline), as shown in Figure 1–8. The offline archive typically removes the data completely from the original storage, requiring operational intervention to restore the data. Offline archives usually operate strictly on a file system basis, by archiving a particular file, set of files, or entire directory structures, regardless of other factors. Data is almost exclusively migrated to static media types, such as tape or optical platters. In order to access the archived data, end users must specifically request the data to be retrieved from the static media and replaced back on the original file system or into an alternate location.

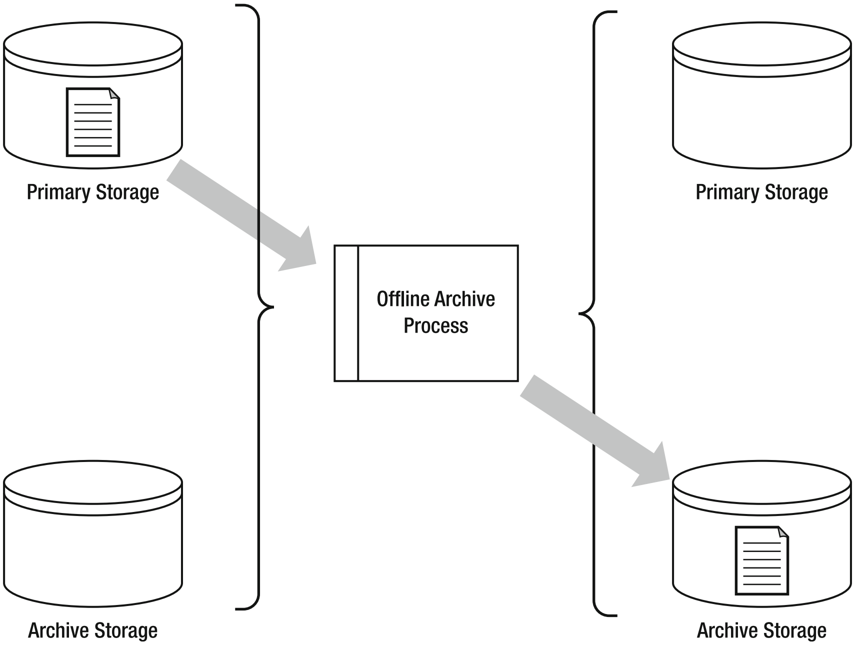


Figure 1–8. *Offline archives*

The offline archive system is most often included with backup software products and is part of both CommVault Simpana and Symantic NetBackup. As such, offline archives have a direct impact on both the size and performance of backups that are made of file systems that have data archived. Because the data has been physically removed and placed onto a separate piece of media, it no longer exists in the context of the backup. Thus, the amount of data backed up as well as the number of files per file system required to be processed are reduced. Although this might appear to solve the problem briefly stated previously, the removal of the data requires that all access requests be operationally satisfied, thus placing a great burden on backup operations staff to constantly retrieve data files.

The *active archive* (or *nearline archive*) differs from the offline archive by providing an archive of data, with the fact of the archive's existence transparent to the end user of the data. Active archives typically interact dynamically with the file systems or data structures that are being managed, typically will migrate data to various types of media, both static and dynamic (for example, disk-based), and leave markers, or *stubs*, behind on the file systems that represent the actual data migrated. Data is migrated based on metadata, such as the age, size, access time, owner of the data, or some combination thereof. The active archive product, typically a piece of software installed on individual file servers, migrates data on an automated, policy basis, using these metadata markers, leaving the stubs behind in its wake.

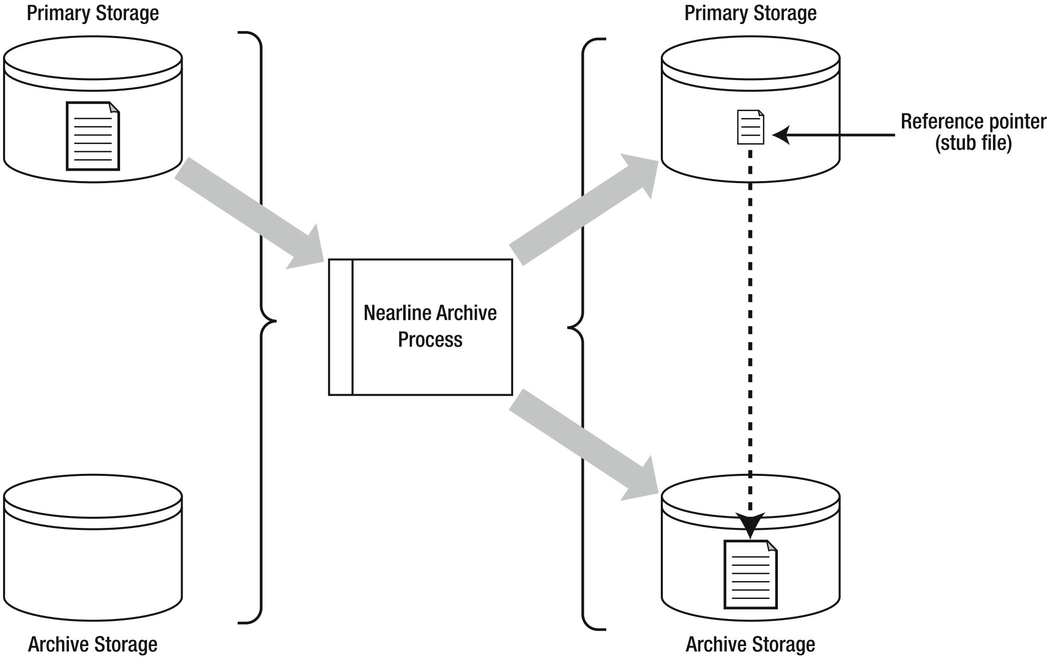


Figure 1–9. *Active (nearline) archive*

When the end user accesses the data in an active archive, the archiving software or hardware intercepts the request (typically an open(2) call to a particular file), preventing the operating system from immediately and directly satisfying the request. The archiving product then identifies whether the file in question is currently managed and whether the data has been archived to a secondary piece of media. If so, the archiving product recalls the data back to the original source and then allows the operating system to satisfy the request for the data to the end user. Note that the end user does not have to be a human user; it can be an automated application that is performing some type of access on data to satisfy a higher-level request. This could be anything from a virus scanner to a process that puts the contents of the file into a database.

Active archives are typically separate from backup software and have an indirect impact on the overall size and performance of the backup of the data source. Although the active archive removes data from the backup stream, it still leaves some data behind in the form of the stub. The stub can vary from 4 K to 8 K in size, depending on the original data type, and (in the case of a large number of files) can still represent a significant amount of data. One of the issues that will be addressed in Chapter 8 is the high density file system (HDFS). An HDFS is a file system with a very large number of files out of proportion to the size of a normal file system—typically one with more than 2 million files per terabyte. Although an active archive might reduce the amount of data backed up, the backup software will still need to process the stubs left behind on the file system. In the case of an HDFS, process can represent a substantial amount of time in the backup, outside of the data transfer.

Does the creation of an archive, whether active or offline, mean the data placed in the archive is never backed up again? No, to the contrary—it is doubly important that the archive is protected because the archive copy represents the only valid copy of the data. Depending on the type of media on which the archive is stored, it might mean that the archive will still require a backup of some type. This backup will not be as frequent, however, and there will be far fewer copies of the archive retained.

If the data were archived to unprotected static media such as a disk, an additional backup of the 5 TB of data in the archive would be required to ensure the survival of the archive in the event of a hardware or logical failure. The backup of the archive would be required only as frequently as new data is added to the archive, or as frequently as required to satisfy organizational or regulatory requirements that ensure the validity of data that is backed up.

Using our 5 TB archive example, suppose that the organization requires that backups older than one month need to be refreshed to ensure that the backup is valid and readable. For simplicity, also assume that no new data is added to the archive, making it static over the lifetime of the archive. To ensure that the 5 TB of data is recoverable, a backup is taken every month of the archive, with the backup having one month retention. How many copies of the data are required to be maintained at any time to ensure the recoverability of the archive? Two: the retention of any one backup copy will not exceed twice the period in which the archive is backed up—in the same way backups required additional weeks to achieve desired retention periods. To satisfy the rules of retention, both the current month and the previous month must be retained to have one month retention. On the third month, the first month's backup copy can be retired, leaving only the two most recent copies of the archive. Thus, only two copies are required at any one time to ensure the survival of the archive, as long as the retention period of the complete set of copies meets the requirements of the business or organization.

Although both backups and archives are copies of data, there is a fundamental difference in what they do with the copies. Backups simply make a copy of the existing data, place it into a specified format, and store the result on some type of media. Archives, on the other hand, make a copy of the data on a separate storage media and then remove the original copy, leaving only the copy as the representative of the original data. Even though the archive location is tracked, there will always only be a single piece of data within an archive situation, regardless of age.

Backups are typically intended to protect against an immediate threat: accidental deletion, system failure, disaster recovery, and so on. Archives are generally created for two reasons:

- To move inactive data from primary storage to lower-cost, longer-term storage
- To provide storage of data required to be stored for long periods of time in a static format

Backups and archives are not mutually exclusive. As discussed previously, the use of archives prior to executing backups can significantly enhance the performance of the backups by reducing the amount of data required to be backed up at any particular point in time.

Unfortunately, backups in many organizations tend to be used as long-term, "archive-like" storage of data and are typically used to satisfy internal or regulatory requirements. They are typically held for periods of more than 5 years and are stored in offsite locations under controlled conditions. As noted previously, backups should not be considered as long-term archives for a number of reasons. First of all is the problem of recoverability. Although the media itself might still be readable (some media has rated static life spans of more than 30 years under controlled conditions), the devices that are needed to actually read the tapes will most likely be gone well before that time.

One example is the videocassette recorder (VCR). When VCRs were first introduced to consumers, there were two competing formats: VHS and BetaMAX. For various reasons, VHS won. Because the two formats were fundamentally incompatible, that meant that anyone with a Beta videocassette was out of luck using it because the players disappeared. Now the same thing is happening to VHS because of DVDs—it is increasingly difficult to find a VHS player to read those tapes. Even the media is degrading—most original VHS tapes are virtually unreadable only 5–10 years after they are created.

Even if the devices are available and the backup software used to create the backup to the media can still read the backup, the application that was originally used to create the data will almost certainly not exist or function on existing hardware or operating systems even short time spans removed from the date of the backup creation. This typically becomes an issue with backups of database systems, but can also affect all types of software applications.

When architecting backup systems, it is important to consider data to be backed up as well as data that will be archived or stored for long periods. Although backups and archives are related, they are distinctly different in character. Backups should be used to provide short- and medium-term protection of data for purposes of restoration in the event of data loss, whereas archives provide long-term storage of data in immutable formats, on static or protected media. The data classification is critical for the proper design of backup systems needed to provide the level of protection required by the organization.

Service and Recovery Objectives: Definitions

When designing a backup solution, there are three key measures that will be the primary governors of the design with regard to any particular set of data:

- Recovery Time Objective (RTO)
- Recovery Point Objective (RPO)
- Service Level Agreement (SLA) associated with the data set

As such, these measures deserve a substantial review of their meaning and impact on design.

There are many different definitions of the SLA that are available. It can refer to the quality of service provided to a customer, the responsiveness of operational personnel to requests, and/or many other factors, but the measure that will be the focus of this discussion is the window in which backups of a particular data set are accomplished. The identification of what constitutes a backup window can be particularly difficult because different stakeholders in the completion of the backup will have differing views of when the window should start and end, and the length of the window. This definition of the SLA must be well-documented and agreed-upon by all parties so that there is no confusion regarding how the SLA is to be interpreted. The proper performance expectations of all parties should be set well before the SLA is in force.

The RTO represents the maximum amount of time that can elapse between the arbitrary start of the recovery and the release of the recovered data to the end user. Although this seems like a simple definition, there can be a great many vagaries embedded into this measure if you look closely (see Figure 1–10). The first is the definition of when the recovery starts. Depending on who you are in relation to the data being recovered, it can mean different things. If you are the end user of the data, this window might start at the point of failure: “I have lost data and I need to access it again within the next ‘X’ hours.” If you are the systems administrator responsible for where the data resides, it might start at the point at which the system is ready to receive the restoration: “The system is up and I need the data back on the system in ‘X’ hours.” Finally, as the backup administrator, you are concerned with the amount of time that it takes from the initiation of the restore to the end of the restore, including identification of data to be restored—“I need to find data ‘ABC’, start the restore, and have the restore finish in ‘X’ hours.”

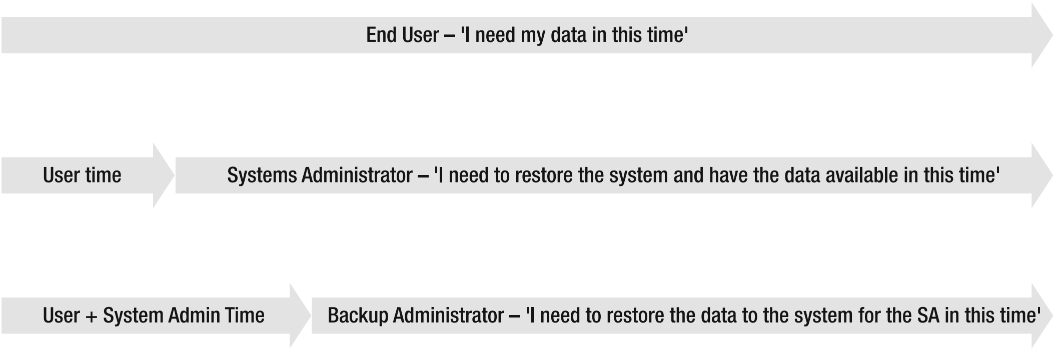

End User – 'I need my data in this time'
User time
Systems Administrator – 'I need to restore the system and have the data available in this time'
User + System Admin Time
Backup Administrator – 'I need to restore the data to the system for the SA in this time'

Figure 1–10. *Relative views of RTO*

Note that each of these viewpoints also contains an implied definition of the end time, or the time at which the data is released to the end user. For the purposes of this book, the RTO will refer strictly to the point at which the data has been identified and the restore initiated and measured to the point at which the restore has completed—Backup Administrator time in Figure 1–10. Although this is a narrow view of the RTO, it provides a point at which the design can be coherent against the variables that can be controlled by the backup infrastructure designer. However, any design must take into account all considerations, and might require significant modification to meet the other stakeholder needs.

The RPO on the other hand, represents the maximum amount of data that can be lost from the point of the last data protection event (see Figure 1–11). Note that a "data protection event" need not be a backup, *per se*, but can represent other types of transient data protection, such as snapshots, log dumps, or replications. Although these events can be controlled by many methods, selecting backup software that integrates the control of these events will simplify any design. The RPO, like the RTO, can be measured from a number of different perspectives. For the backup administrator, it will represent the largest amount of time that can elapse between backups (or controlled snapshots) in order to ensure that the data age is appropriately protected: "I need to complete a backup every 4 hours to ensure that only 3 hours of data is lost".

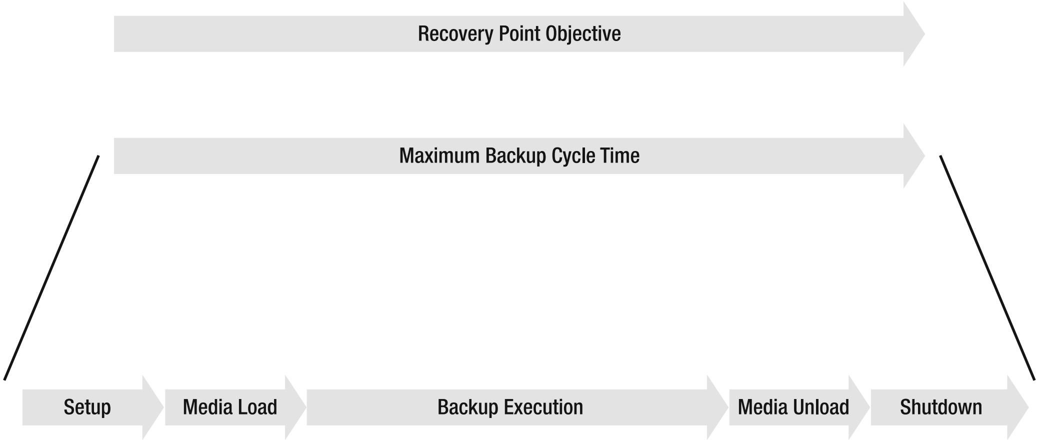


Figure 1–11. *RPO activities*

From the perspective of the data owner, this might represent a number of transactions, an amount of data that can be lost, or a particular age of data that can be regenerated: "The organization can afford to lose only the last 30 transactions".

The primary issue with establishing the RPO is the translation between time and data. A good way to illustrate this is to look at the two requirement statements in the previous paragraph. The first one, from the backup administrator, talks in terms of time between backups. For the backup administrator, the only way to measure RPO is in terms of time—it is the only variable into which any backup software has visibility. However, the requirement statement from the organization does not have a direct temporal component; it deals in transactions. The amount of time that a number of transactions represent depends on any number of factors, including the type of application receiving/generating the transactions. Online transaction processing (OLTP) database applications might measure this in committed record/row changes; data warehouse applications might measure this in the time between extract/transform/load (ETL) executions; graphical applications might measure this in the number of graphic files imported. The key factors in determining an estimated time-based RPO using data transactions are the time bound transaction rate and the number of transactions. The resulting time between required data protection events is simply the number of transactions required to be protected, divided by the number of transactions per unit time. For instance, if a particular database generates an average of 100 transactions per minute, and the required RPO is to protect the last 10,000 transactions, the data needs to be protected, at a minimum, every 100 minutes.

The other issue with RPO is that when designing solutions to meet particular RPO requirements, not only does the data rate need to be taken into account but the time for the backup setup and data writing also needs to be taken. In the previous example, if there is a requirement to protect the data every 8 hours, but it takes 8.5 hours to back up the data, including media loads and other overhead, the RPO has not been met because there would be 30 minutes of data in the overlap that would not necessarily be protected. This actually accelerates as time progresses. Again with the example, if on the first backup, it takes 110 minutes to perform the backup, the backup cycle is 30 minutes out of sync; the next time it will be 1 hour, and so on. If the extra time is not accounted for, within a week the backup process will be 8 hours out of sync, resulting in an actual recovery point of 16 hours.

If the cause of the offset is simply setup time, the frequency of the backups would simply need to be adjusted to meet the RPO requirement. So, let's say that it takes 30 minutes to set up and 8 hours to back up the data. In order to meet the stated RPO, backups would need to happen every 7.5 hours (at a minimum) to ensure that the right number of transactions are performed.

However, if simply changing the backup schedule does not solve the problem, there are other methods that can be used to help mitigate the overlap, creating array-based snapshots or clones. Then performing the backups might be able to help increase the backup speed by offloading the backups from the primary storage. Other techniques such as using data replication, either application- or array-based, can also provide ways to provide data protection within specified RTO windows. The point is to ensure that the data that is the focus of the RTO specification is at least provided initial protection within the RTO window, including any setup/breakdown processes that are necessary to complete the protection process.

■ **Note** So are the RTO and RPO related? Technically, they are not coupled—you can have a set of transactions that must be protected within a certain period (RPO), but are not required to be immediately or even quickly recovered (RTO). In practice, this tends not to be the case—RTOs tend to be proportionally as short as RPOs. Put another way, if the data is important enough to define an RPO, the RTO will tend to be as short as or shorter than the RPO:

```
RPO <= RTO
```

Although this is not always the case, it is a generalization to keep in mind if an RPO is specified, but an RTO is not.