

**Sample Chapter**

Jason Buffington

# Data Protection for Virtual Data Centers



## Chapter 2

# Data Protection by the Numbers

Numbers make everything equal.

That applies to the wide range of data protection technologies, though it does not imply that synchronous storage arrays are equal to nightly tape backup. What we should think about is that comparative metrics like RPO allow us to look at that range of availability and protection alternatives objectively, without the bias of vendor preference, past experience, or preconceptions. We'll explore those kinds of comparative metrics in this chapter.

In this chapter, we will look at several metrics. We'll define each one and apply it to the discussion of what kinds of data protection and availability you need for different scenarios.

## The Technical Metrics: RPO and RTO

When comparing the wide range of data protection technologies and methodologies, the two technical metrics that provide us with a standard for comparison are the recovery point objective (RPO) and the recovery time objective (RTO). As an introduction to these terms, consider a traditional tape backup scenario, where a full backup is done every weekend and an incremental backup is done every evening after the users go home.

### Recovery Point Objective

In Chapter 1, we looked at the range of data protection solutions as categories that could be effectively delineated by the questions "How much data can you afford to lose?" and "How frequent is the data protection event?"

The proper term for this metric is recovery point objective (RPO). Where RPO really matters is as a method of objectively comparing the diverse range of data protection and availability technologies.

RPO is often thought of as the amount of data that could be lost. That's not the whole story, but we'll start there. If you are backing up every evening and we assume nothing goes wrong during the backups or the recovery, then the most you could lose is one business day's worth of data. If your data is made up solely of documents from Word or Excel, then you have lost only those documents that were updated during that day. If your data consists of transactions such as financial records, then the consequences could be worse. Imagine that you work in a bank and in one day, most if not all of your accounts have some kind of activity. If you lose a day's worth of those transactions, your entire data set is no longer valid. The key point here is that you must assess your potential for data loss in two ways: time spent re-creating lost data and the scope of data that will be lost or affected.

To expand on that, let's assume that a reliable backup takes place every evening and the restores will always work. I'll explain later why that doesn't usually apply; but for now, that supposition helps for the example. With that in mind, there are two extreme scenarios:

- ◆ If the server were to fail at the beginning of the business day, almost no data would be lost since the last backup. The actual data loss would be measured at near zero because nearly nothing would have changed since the last *recovery point* or backup event.
- ◆ If the server were to fail at the end of the business day, that entire day's worth of data would be lost, because no backups (recovery points) would have been created since the midnight before. We would measure the data loss as a full day's worth.

If we take the average of these two extremes, we can presume that the server will always fail at noon—halfway into the business day. Statistically speaking, this means that companies that use tape backup will lose half of a day's worth of data on average.

To learn the whole story when looking at RPO, it is the “O” that is most important. RPO is an objective (or goal). It specifies how much data you are willing to lose. In nightly tape backup, the statistical probability is that you will lose a half day of data. But if you establish your RPO at “half of a day” and then your server fails in the afternoon, you have actually lost more data than you planned, and you fall short of your goal or objective. So, most would set an RPO as “one day,” meaning that it is an acceptable business loss to lose an entire day of data because of the recognition that backups are only occurring nightly and the server could fail in the afternoon.

In the case of tape backup, we measure RPO in days (for example, half day or full day) because we normally do backup operations on a daily, or more specifically, on a nightly basis. To have an RPO, or goal, for how much data we can afford to lose (which can be measured in less than days), we have to protect our data more often than nightly. That usually takes tape out of the equation. When we look at the wide range of disk-based protection, we can see that disk-based solutions often replicate hourly, or every few minutes or seconds, or real time. RPO essentially becomes the measurement of that data protection frequency.

### DEFINING YOUR RPO

If your business goals say that you don't want to lose more than two hours of data, that is your RPO—your objective or goal for how frequently you need a reliable recovery point.

In the case of a two-hour RPO, you have to look at data protection technologies that operate at least every two hours. As we look at the sampling of protection technologies in this book, we'll consider how often they protect and relate that back as the respective RPO of each solution.

## Recovery Time Objective

From a technology perspective, almost all data protection and availability solutions can be compared with one another by charting them on a graph, with RPO as one axis and RTO as the other. That's how we'll assess the technologies later in the book—by looking at them in part by how they compare in RTO and RPO. In real terms, RTO asks the question “How long can you afford to be without your data?” which could also be asked as “How long can your services be out of operation?”

Using the same scenario as the RPO discussion (where you are doing a nightly tape backup), the RTO is the goal (objective) for how long it takes to conduct a recovery. The question is, how long will the restore take?

In the example of nightly tape backup, RPO for tape backup is measured in days or partial days, because that is how often a data protection (backup) event is actually occurring—every night. But for this same example, RTO is measured in hours, because it is a performance measure of the components in your solution itself. If your backup software and tape hardware can restore up to 2 TB per hour and the server has 6 TB of data, time to recover the data is effectively 3 hours—or at least 3 hours from when the restore actually begins.

If your largest server holds 10 TB of data and your tape hardware can restore 2 TB per hour, and you are confident that you could immediately locate the right tapes and restoration could begin soon after, then you might specify an RTO of 5 hours—or 6 to be cautious. But because you may need to prepare for the restore and locate the tapes, you will likely round up and specify an RTO of one day.

## Putting RPO and RTO Together

Let's combine the examples that we have used so far. We'll assume that the server we have been protecting in this chapter's scenario failed on Wednesday at 4:00 p.m.

It will take us most of the next day to recover the server. If we have IT personnel in the same office, we can optimistically identify what has failed and, if necessary, arrange for replacement parts (for example, new hard drives) to arrive early Thursday morning. On Thursday, we'll repair the server and restore the data. By Thursday evening, the server will be rebuilt and recovered. The recovery time will be one business day and that hopefully was within our RTO.

The unfortunate part of this scenario is twofold for the users:

- ◆ Thursday is a wasted day for the users, because they cannot get to their server or its data while things are being repaired, replaced, and restored.
- ◆ Wednesday's data is likely lost, because when the server is restored, it will be restored to the latest successful backup (Tuesday night). Everything that was created during Wednesday (after the Tuesday backup) will likely have been lost when the server storage failed. The recovery point was within one day of lost data (Tuesday midnight through Wednesday at point of failure), which again is hopefully within the set RPO.

**NOTE** To improve the RTO, we need a faster restore medium, which usually points us to disk instead of tape for routine restores.

To improve the RPO (frequency of backups), we need to perform data protection more often than nightly. For this we must turn to replication technologies, including the range from sub-hourly replication, to database mirroring within seconds, to synchronous disk.

But just doing the technical measurements of RPO and RTO isn't enough. We have to describe the RPO and RTO characteristics of our technology as something predictable that can be understood and agreed to by the business and operational stakeholders of the company. We have to set a service level agreement (SLA).

## Making RPO and RTO Real with SLAs

When we recognize that the "O" in both RPO and RTO is *objective*, we run into one of the key problems in most data protection and availability plans. An objective is a goal, not a promise. The promise comes when we describe our capabilities to the stakeholders in the business, when we tell the management of the people who rely on the server that they will be "running again within one business day and will lose an average of half a day of data but potentially a full day of data." We might tell the management team that with nightly tape backup, we could have an RPO of half

a day of lost data and that the RTO might be one business day to repair the server and restore the data. But those are goals based on ideal circumstances.

What happens when the circumstances are not ideal? In our scenario of tape backup of a failed server, we made a few assumptions:

- ◆ We assumed that we are able to react quickly to the server outage.
  - ◆ If we have IT staff on-site, they can identify the issue almost immediately.
  - ◆ If we don't have IT on site, our entire restore time window will be longer because nothing can happen until we get there (or remotely connect in).
- ◆ We assumed that the server is readily repairable. In our example, the server failed at 4 p.m. on Wednesday.
  - ◆ If parts are available, we can begin repairs immediately.
  - ◆ If we happen to be on the US East Coast, we can expedite parts from a West Coast provider, where they can overnight them and we can begin repairs the next morning.
  - ◆ If we happen to be on the US West Coast, we may not be able to get parts for another whole business day—and everything else will be prolonged accordingly.
- ◆ We assumed that every tape is readable.
  - ◆ If the latest Tuesday evening tape is unreadable, we will only be able to restore up through Monday's tape. We will have lost another day of data (RPO), and we will likely lose time trying to restore Tuesday's data before we can identify the failure (longer RTO).
  - ◆ If you are doing incrementals (only nightly changes) instead of differentials, then if Monday's tape is bad, Tuesday's tape is mostly irrelevant. Tuesday's incremental contains the differences between Monday and Tuesday, but without a successful restore of Monday's data, Tuesday's changes may not be substantive. This will vary by the production workload (as well as the backup software's tolerance for failed tapes in a recovery set).
  - ◆ If one of the weekend full tapes is bad, then Monday's and Tuesday's are irrelevant, because everything is in the context of the last full backup (which is unusable). Instead, we have two last-resort recovery scenarios:
    - ◆ If the daily incremental tapes are not overwritten each week (that is, Tuesday overwrites last Tuesday), we can restore the full backup from a week prior, and then the incrementals or differentials from the previous week. In short, when the server is repaired on Thursday afternoon (accessed Friday morning), the data will be as it was the Thursday of a week before—the last good tape.
    - ◆ If the daily tapes are overwritten, our data will only be back to the full backup from a week ago. All data for the previous week, as well as the beginning of this week, is lost (10 days of data in our example).

These aren't calamity-of-errors or niche cases. They are just examples of how easy it is for our reality to fail to match the ideal RPO and RTO that is defined by the hardware and software of

our data protection solution. It is for these reasons (where reality doesn't match our ideal RPO and RTO) that our SLA—our commitment to the business units as to what our recovery capabilities are—needs to be broader than just stating the RPO and RTO of the technologies themselves. We need to consider the processes and potential pitfalls as well:

**Time to React** Sites without IT staff should have longer RTO SLAs than sites with IT staff, because it will take time to get there, depending on the arrangement. Perhaps IT staff can drive or fly from their primary location to the remote office. Perhaps a local integrator or channel reseller can be dispatched; in that case, a pre-negotiated contract may have to be put into place, including an SLA from them to you on their committed response rate to your issue.

**Time to Repair the Server** Should spare parts or even complete cold-standby servers be acquired? Where can parts or servers be expedited from? Does a pre-negotiated agreement need to be signed between you and a vendor or distributor?

**Technical RPO and RTO** This relates to issues involving the RPO and RTO, as well as the perceived failure rate of the media.

Notice that technology wasn't mentioned until the last item. The first aspects of a server recovery SLA relate to people and process, followed by materials and access. Once we get to the technology, we are likely more in the comfort zone of the IT professional, but there are still unknowns concerning the technology.

Recall from Chapter 1 that an estimated 12 percent of modern server failures are caused by hardware rather than software (at 88 percent). In that case, the hardware-failed scenario that we've been using may only affect you one out of eight times, and that doesn't sound like too much. But in a datacenter with 100 production servers, the statistical probably is that 12 of them will have hardware issues. This means that you will have to enact this recovery scenario once per month.

After the server is ready to be restored, RTO will still vary based on whether you are restoring from Monday or Friday:

- ◆ With *differentials*, you might need the weekend full plus the Thursday night differential. But that differential will have appreciably more data to restore than the Monday differential. Each tape will have everything (the *difference*) since the last full backup.
- ◆ With *incrementals*, we see a similarly linear increase in restore time, as each subsequent incremental is layered on top of the one before it. Unfortunately, in our imperfect world, if a tape fails, and depending on the kind of interdependence data that is being restored, you may have to wipe the production volume and repeat the recovery exercise only up through the previous day's tape.

All of these overly dire and pessimistic examples are designed simply to prompt you to compare your data protection technology's presumed RPO and RTO to what you as the backup administrator can assure your management of being able to deliver.

This is the secret to a successful SLA. Salespeople call this sandbagging, where what you forecast that you'll sell is less than what you believe is likely. Personal life coaches might call this underpromising and overdelivering. As a consultant and IT implementer, I call it planning for Murphy's law.



## Real World Scenario

### MAKING “SENIOR” PREDICTIONS

Several years ago, I was working for a systems integrator, and my team and I were on site doing a deployment. The company had recently gone through a formalized standardization of job titles, and mine at the time was senior systems engineer. One of the other folks on my team was a systems engineer. During a break, we were chatting and he asked, “What makes you ‘senior’?”

At that time, I was younger than most of the other people working on my team. My response, though thoughtful, probably sounded arrogant. I replied, “I’ve done more things wrong and have the scars to prove it.” In theory, that meant that I would not do those things again and could help others avoid them.

Perhaps they weren’t all things that I actually did wrong, but I was there when the bad things did go wrong. That includes dealing with tapes that were unreadable, mirroring the clean drive over the copy with the data, doing something unsupported with an application and then not being able to get support, and more.

The lesson learned here is that SLAs can sometimes be more art than science, because to have good ones that both the IT staff and business management can be satisfied with takes creative planning, usually by folks who have suffered through the things that can go wrong. Balance must be achieved:

- ◆ If the IT team is overly conservative and cautious, they may set the SLA performance bar so low that the business management team believes that the IT staff is unknowledgeable or low-performing.
- ◆ If the IT team is overly optimistic or unrealistic, the SLA performance bar may be so high that even well-executed recoveries may fail to meet the measure established in the SLA.

When negotiating your IT SLA with the business leaders, consider first doing a brainstorming session with your senior IT folks to map out the recovery plans, identifying the likely failure points and your mitigating actions when the plan does break down. Once you have that workflow, only then should you talk to the business managers about SLAs.

When you are setting your own SLAs, don’t believe the RPO and RTO on the outside of the box of whatever technology you are looking at. And certainly don’t repeat the RPO and RTO to the business managers. Test it. Assume something will go wrong and think about how you would address such issues. Heck, you can even go to the extreme of thinking everything will go wrong and then negotiate with the business managers back to a point of reality.

## Business Metrics: RA and BIA

Now that we have some metrics to assess what our alternative technologies are by understanding RPO and RTO and setting an SLA for how well we can act upon them, let’s see how to apply them to the business, as well as how to pay for the technologies that we believe we need.

### Risk Analysis (RA): The Science of Worrying

How likely is it that your particular tape solution will have a problem?

Perhaps more important, how likely is it that your production resource will suffer failure? Consider my house in Dallas, Texas. How likely is it that I will suffer a flood? Or a monsoon? Or an earthquake? Or a tornado?

Let's focus on this last question to take technology out of the process. As I mentioned, I live in Dallas, Texas, which is approximately 400 miles from the nearest large body of water, the Gulf of Mexico. Because of this, I have no fear of a monsoon. Statistically speaking, the likelihood of Dallas being hit by a monsoon is effectively zero. Speaking of water, my home is in a 100-year flood plain, meaning that, statistically, my land will be flooded once per century. I could buy flood insurance for my home, but the probability is low enough that I choose not to. If I lived in Houston, Texas, which is much closer to the coast, flooding is more likely and I might want flood insurance. But, because the probability is so much higher, I probably could not afford it, as the insurance actuaries will have already calculated it. Case in point: hail damage is frequent enough in Dallas that I can't afford to buy insurance for it.

This has meaning for our discussion. Insurance is an entire industry built upon a consumer's presumption that they pay a little every month to avoid a potentially significant and perhaps life-altering financial impact later. The amount of insurance that you pay is based predominantly on two factors:

- ◆ How likely is the crisis that you are anticipating?
- ◆ What is the financial impact that you are mitigating?

**NOTE** Concepts such as data protection and data availability are very similar to the idea to buying insurance. First, you assess what could go wrong and consider how much it will cost if it does, and then you purchase something that costs appreciably less than that to mitigate the crisis.

## WHAT COULD POSSIBLY GO WRONG?

The first step in planning your data protection and availability strategy is to look at each of the servers and applications in your environment and think about what could go wrong.

Go crazy. Think about *everything* that could possibly go wrong. The most important rule of this exercise is to simply list everything that could go wrong. Do not (yet) think about the probability of something occurring, but just the potential of its occurring. And let yourself think small and think big.

In the case of a core application, don't just consider the application itself. An end user does not care if the reason they can't get to their data is because the application crashed, the OS hung, the hard drive failed, the DNS server isn't resolving correctly, Active Directory won't let you log on, or your PC browser isn't reading the page correctly. They don't care, because their data and their productivity is impacted, regardless of the cause.

That's from the user's perspective. Now think about the big problems. Is your company in a flood zone? If you are in Southern California, are you near a forest that can catch on fire? If you are in Northern California, what would an earthquake do? If you're in the Midwest, are you in tornado country? In the North, what would a blizzard do? On the East Coast, how likely is a hurricane? If you live in Florida, hurricanes are a *when*, not an *if*.

**NOTE** Several years ago, I was conducting a disaster recovery seminar, in a town in Florida. My opening remark to them was, "According to the National Weather Service, this city is in the eye of a hurricane every 2.83 years. It has been 3 years since you have actually been hit. You are due."

## How LIKELY Is It?

I am not suggesting that every IT professional should turn into an actuary, someone who lives with the statistics of risk all day long. What I *am* suggesting is that when you are first imagining the entire realm of bad things that could happen to your data, servers, infrastructure, and even people, first just list them. Having done that, put your practical hat back on and consider the reasonable probability of each one. The reason I don't buy flood insurance is because a flood, while possible, is not probable for me. I cannot buy hail insurance at a reasonable price, because it is almost certain it will happen to me.

In technology, there are some calamities that are certain to happen:

- ◆ You will lose a hard drive.
- ◆ The system board will fail.
- ◆ An application will crash.
- ◆ A database will become corrupted.
- ◆ A user will accidentally overwrite last month's file with this month's data and then regret it.

In business, there are some crises that are very likely:

- ◆ Someone may steal something, perhaps a laptop, from your company.
- ◆ Someone may maliciously delete data on their last day at work.
- ◆ Your server room may catch fire or might be flooded from the bathroom immediately above you.

In life, natural disasters could affect your company facilities.

So what is the likelihood of each thing that you listed? You may not have exact figures, but stack the kinds of things you're protecting against in relative probability to each other. This exercise is half of what you need to start planning your data protection and availability plan.

## Business Impact Analysis (BIA): How Much Will It Cost?

Data protection and availability is not just about technology—it is about reducing financial impact. To do that, not only do we need to look at the technologies that we could use and the calamities that we fear, but we also need to turn all of them into financial ramifications.

Let's look at the potential technology and business crises listed in the previous section. They are ordered approximately from most likely to least likely, with the exception of the end user who accidentally overwrites precious data (it is guaranteed that a user will overwrite data). Let's look at the two extremes of the list. Everything else will fall in between from a likelihood perspective as well as a financial impact.

- ◆ If I were to lose one hard drive within a production server, the physical costs are likely a few hundred dollars or less. Whether it was for a data drive or the operating system will determine the level of lost productivity. And how long since my last backup will determine the amount of lost data that I may or may not have to re-create.
- ◆ On the other end of the list, if my production facility were to be flooded, the entire server room as well as many other production resources, desktops, infrastructure, even copy machines and coffee makers would be destroyed. My business could be down for days and in fact might never reopen.

The goal of a business impact analysis (BIA) is to financially quantify what the cost of any crisis might be. Say we calculate that the total cost of a hard drive failure, lost productivity, and replacement is \$1,000. We believe that it is a highly likely event, so we need to aggressively seek a data protection or availability solution that mitigates that \$1,000 of impact to our business by finding a mitigation solution that costs less than the \$1,000—hopefully, a lot less. This may be RAID or backup or replication, all of which are discussed in this book. Similarly, while we may believe that a flood would cost \$3 million, it is admittedly far less likely than a hard drive failure. That statistical probability factors into determining what we might spend to mitigate that risk.

### ALWAYS TURN TECHNOLOGIES INTO DOLLARS

Most often, the person who writes the checks, particularly the checks for buying new assets like software and hardware, doesn't care about RPO and RTO, or DLT versus DAT, or disk versus tape. To move business-driven decision makers forward on data protection projects, we always want to quantify the risk or the reward in dollars, not gigabytes, minutes, or subjective assessments.

Data protection and availability projects are among the easiest in financial terms. Availability, or said another way productivity, can be calculated by looking at the cost of downtime. Protection and recoverability can be quantified based on the impact of lost data, as it relates not only to lost productivity but also to lack of compliance.

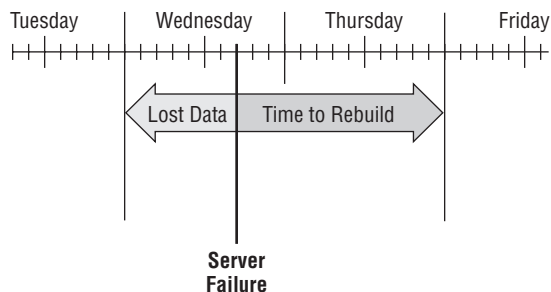
In short, if we can objectively state that 1 hour of downtime equates to \$10,000 and the solution to resolve it cost \$800, it is easy to justify new data protection or availability solutions.

### CALCULATING THE COST OF DOWNTIME

The key idea that we need to take away from this chapter is how to turn technology problems in financial problems. If you can fiscally quantify the impact something has on the business, then you can have a different kind of discussion with business (and budget) leaders on why you need to fix it. To start that conversation, we need to understand the cost of downtime. In other words, when a server breaks, how much does it cost the company?

If a server fails, you actually have to look backward as well as forward. Figure 2.1 shows a server failing at 2 p.m. on Wednesday.

**FIGURE 2.1**  
Downtime, forward  
and backward



For this first example, we will make three assumptions:

- ◆ The business day is exactly 8 a.m. to 5 p.m.
- ◆ We have a successful backup from Tuesday night that is restorable.
- ◆ The server will be fixed by the end of the next business day (Thursday).

With this in mind, we can quantify two kinds of time:

**Lost Data** If a server fails, the new data since the last backup is potentially lost. Since our server failed at 2 p.m. on Wednesday and we are assuming a reliable restore from a successful backup on Tuesday night, we can presume that we could lose whatever data was changed between 8 a.m. to 2 p.m. on Wednesday.

In our diagram, this arrow starts at the time of the server failure and points backward to the left to whenever the last successful backup can be reliably restored from—in this case, the previous evening. If the last backup had failed or was not able to be restored, the arrow would point further to the left until a successful backup could be restored. But for now, data loss is quantified at 6 business hours.

$T_d$  = Time of lost data, which in this case is 6 hours

We are quantifying lost data in a measurement of time because for our basic example, we are assuming that if the end users took 6 hours to originally create the data, then they will likely consume another 6 hours of business time to create the data again.

**Outage Time** Because the server failed in the afternoon, we are assuming that the end users may be idle for the remainder of the day and (in our example) idle for the whole next business day.

In our diagram, this arrow starts at the time of the server failure and points forward to the right until the server is completely back online, which in this case is the end of the next business day. If this is true, the outage time is the remaining 3 hours of Wednesday afternoon plus all 9 business hours of Thursday.

$T_o$  = Time of outage or lost productivity, which is 12 hours in our example

Added together, we can say that  $T_o + T_d = 12 + 6$ . So the total time impact of the server failure is 18 hours. Now, we need to decide how much those 18 hours are worth in dollars. Again, there are two kinds of \$/hour costs to consider:

**Human Costs** If we presume that an end user is completely idle while the IT resources are offline, then the company is essentially paying the salary or hourly wage of that person for no benefit.

$H_r$  = Hourly cost of impacted personnel (\$ per hour)

Consider the following:

In a restaurant that is not able to do any business, you might reduce your losses by sending the waiters and cooks home for the day. But if 15 hourly staff who each costs the company \$8 per hour (along with two salaried managers paid \$40,000 annually or \$20/hour) were to sit idle, then  $(15 \times \$8) + (2 \times \$20) = \$160$  per hour for idle time.

In an office, perhaps there are other activities that the people can do, so they are not completely idle but simply impacted. In that case, you might choose to take the salaried costs and divide them in half to show that they are half-impacted as opposed to idle and nonproductive.

Every business is different, but you should be able to assess a \$/hour number for some percentage of your hard costs of paying people who are unable to do their primary role due to an IT outage.

**Profitability** When a team that creates revenue is affected, revenue is affected. So, if you know what the weekly or monthly profitability of a team is, you can quantify how much profit that they are generating or not generating during an outage.

$P_r$  = Hourly profitability or loss (\$ per hour)

Consider the following:

A team may produce \$9,000 per day in profit, so their hourly profitability between 8 a.m. and 5 p.m. is \$1,000 per hour.

In a team that is subject to service contracts, you may be liable for fines or recouped losses if you are not offering your service.

A shipping department may not lose any money for a few hours of downtime, but if an entire day is lost, expedited shipping charges may be incurred in order to make timely deliveries the next day.

Every business is different, but you should be able to assess a \$/hour for the business value that a team creates per day or per hour. Some of that productivity is lost or penalties incurred when the team is unable to do their primary role due to an IT outage.

Adding  $H_r$  and  $P_r$  together gives us the total dollars per hour impact that an IT outage has on our team. Using the first example from each description, a team may cost \$160/hour by sitting idle ( $H_r$ ) and also not create revenue ( $P_r$ ) at \$1,000/hour. Thus, every hour is worth \$1,160 to the company.

This brings us to a basic formula for measuring systems availability in financial terms. We can take the total time for data loss plus outage time, and multiply that by how much an hour is worth to our business or team in consideration of human costs, as well as profitability or losses:

$$\text{Cost of Downtime} = (T_o + T_d) \times (H_r + P_r)$$

$T_o$  = Time, length of outage

$T_d$  = Time, length of data loss

$H_r$  = Human cost \$/hr (per person)

$P_r$  = Profitability \$/hr

In our examples, this would be:

$T_o$  = 12 hour outage

$T_d$  = 6 hours of lost data

$H_r$  = \$160/hour for the team to sit idle

$P_r$  = \$1,000/hour in lost revenue

$$\text{Cost of Downtime} = (12 \text{ hours} + 6 \text{ hours}) \times (\$160/\text{hour} + \$1,000/\text{hour})$$

$$\text{Cost of Downtime} = 18 \text{ hours} \times \$1,160/\text{hour}$$

$$\text{Cost of Downtime} = \$20,880$$

This particular company will lose nearly \$21,000 if a server fails and is recoverable by the end of the next day.

**YOUR MATH MAY VARY FROM THIS, AND THAT IS OKAY**

I guarantee that a good percentage of people who read that formula will find something that does not quite align with their business.

Perhaps team profitability should be reduced to half ( $\frac{1}{2}P_r$ ).

Maybe data is never lost—in the sense that it does not have to be re-created—because of the nature of the business ( $T_d = 0$ ).

The point is to take the formula as a starting place and adapt each of the four variables to correctly reflect your business:

- ◆ Hours of data loss or repeated work
- ◆ Hours of downtime or reduced-productivity
- ◆ Cost of sitting idle
- ◆ Lost profitability

So, although the basic formula works for me, it may not be your final formula. In fact, the best possible outcome for you taking this formula to your management would be the shreds of doubts that immediately follow, because then you can work together on adapting it to your business model. We will take a closer look at that throughout this chapter.

That is the formula, but it is still not the answer. The idea here is simply to help identify the variables that we'll need in order to quantify the cost of downtime.

**THE COST OF DOWNTIME FOR NIGHTLY BACKUP FOR A SMALL OFFICE**

We're still considering the same outage scenario of an environment that is using nightly tape backup that includes a full backup every weekend and incrementals each night. Consistent with the scenario we've used for this chapter, the production server fails at 2 p.m. on Wednesday. As we discussed earlier in the chapter, the users are affected for the rest of Wednesday and the recovery takes a good part of Thursday. By the end of Thursday, the server is running, the users are mostly happy, and business resumes. Two months from now, that incident will be thought of from long-term memory as a minor blip. Yes, the server went down, but everything resumed within a day. Pretty good, right?

$$\text{Cost of Downtime} = (T_o + T_d) \times (H_r + P_r)$$

$T_o$  = Time, length of outage

$T_d$  = Time, length of data loss

$H_r$  = Human cost \$/hour (for team)

$P_r$  = Profitability \$/hour

$$\text{Nightly Backup} = (1d + \frac{1}{2}d) \times (H_r + P_r) \times \text{hrs/day}$$

where

$T_o$  = RTO = 1 day recovery, including parts, shipping, and installation

$T_d$  = RPO = average ½ day (could fail early morning or late afternoon)

$H_r$  = Human cost \$/hour (per person)

$P_r$  = Profitability \$/hour

$T_o$  (time of outage) or RTO will likely be one business day, which includes identifying why the server failed, repairing those components, and restoring the data. If everything goes well, this should typically be one business day. If things do not go well, this might measure two or three days of downtime. In a perfect world, additional parts are already standing by, technicians are ready to go, and perhaps the server is up in just a few hours.

$T_d$  (time of data loss) or RPO is statistically probable as one half of a business day. As discussed earlier in the chapter, the server could fail at the beginning of a business day, resulting in near zero data loss since the last nightly backup, or could fail at the end of the business day, resulting in a complete day of data loss. Splitting the difference, we will assume data is lost from a half of the day.

If we take a closer look at this particular office, perhaps a retail store, we will assume a 10-hour workday ( $d = 10$  hours).

Small Store Using Nightly Backup =  $(1d + \frac{1}{2}d) \times (H_r + P_r)$  @ 10hours/day

For this particular store, managerial costs ( $H_r$ ) are \$24.00 per hour, while the five employees cost \$8.00 per hour. We will assume that the manager does not directly create profitability but will suffer from lost data:

Administrative Productivity Loss =  $(10hr + 5hr) \times (\$24 + 0) = \$360$

The five employees might not lose data, but they lose the ability to impact productivity and will have a hard cost of sitting idle:

Retail Employees =  $(10hr + 0) \times (\$8 \times 5 \text{ employees} + \$\$/\text{day}) = \$400$

And, in retail, of course, profitability is everything. Presume this small store generates \$100,000 in revenue over the course of one year. That would mean that within a six-day sales week, each business day generates \$320.

Resulting cost of a “minor” server outage:

Cost per Outage = \$360 (manager) + \$400 (employees) + \$320 (lost profit) = \$1080

Every time the server has an outage that must be recovered, the immediate cost to this small 6-person storefront is \$1,080, not including replacement parts/shipping, lost customer loyalty, and services from either headquarters or a local reseller to resolve the issue. This last penalty, of the additional expense for technology professional to be dispatched to the office and resolve the issue, exacerbates everything else. For a small business, while employees are down, the last thing that their budget can handle is an expensive emergency service call. They might spend \$250/hour for a technician to come out for a day to repair the process. At that point, they will have spent \$2,000 in labor to fix something that costs \$1,080. The total business impact is now \$3,080.

### THE COST OF DOWNTIME FOR NIGHTLY BACKUP FOR A BIGGER OFFICE

We don't have to lay out the entire scenario again, but it is worth recognizing how quickly this scales. Assume that this takes place within a larger organization, perhaps one division within a good-sized company, such as the inside sales team of a medium-sized company. Various surveys presume that the average white-collar worker in the United States costs \$36/hour. The reality of a statistic this broad is that it is guaranteed to be wrong for your workforce, but it serves as a placeholder for now. We will continue to presume a 10-hour workday, and that this team generates \$10,000,000 in revenue annually.

$$\text{Cost of Downtime} = (T_o + T_d) \times (H_r + P_r)$$

$T_o$  = Time, length of outage

$T_d$  = Time, length of data loss

$H_r$  = Human cost \$/hour (for team)

$P_r$  = Profitability \$/hour

$$\text{Inside Sales Team Relying on Nightly Backup} = (T_o + T_d) \times (H_r + P_r) \times \text{hours/day}$$

$T_o$  = RTO = 1 day recovery, including parts, shipping, and install

$T_d$  = RPO = average ½ day (could fail early a.m. vs. late p.m.)

$H_r$  = Human cost = \$36/hour/person  $\times$  50 employees = \$1,800/hour

$P_r$  = Profitability = \$10M annually = \$3,850/hour (10-hour workday, 5-day workweek)

$$\text{Inside Sales Team relying on Nightly Backup} = (1d + \frac{1}{2}d) \times (1800 + 3850) \times 10 \text{ hrs/day}$$

$$\text{Business Impact per Server Outage} = \$874,000$$

This is not a typographical error. If the primary file server fails for an average group of 50 office employees that creates revenue (all of our earlier presumptions), then the business impact to that group is \$874,000.

### ADAPTING THE FORMULA TO YOUR BUSINESS

To be fair, this isn't the whole story. The likely outcome of first evaluating this formula is that management will disagree with its validity. And that is fine, because they are probably correct.

- ◆ If your users cannot get to their file server, they might catch up on e-mail or they might have some documents on their local workstation or laptop. In this case, let's assume that the employees are not completely idle but are simply affected. If that were the case, we might add a multiplier to the formula to imply that the user base is operating at 2/3 efficiency. If so, a 1/3 multiplier against the formula results in a business impact of only \$291,000. That is still over a quarter-million dollars per server incident.
- ◆ In today's information worker world, we might presume that the users have a variety of activities that they could do. Between e-mail, database applications including contact management, as well as traditional office applications from a file server, perhaps we could presume only a minor inconvenience to a percentage of the users. Perhaps this results in a 10% impact to 10% of the employees. Literally, this would mean only 5 of the 50 employees had any impact and therefore the cost would only be \$8,710.

- ◆ But some departments don't have multiple functions that they can balance between. In the example of inside sales, what if all of the data was within a single SQL Server database, or the sales folks fundamentally could not operate without access to the database. In that case, they would suffer the whole (and what may have originally seemed extreme) business impact of \$871,000.

The business impact may be even higher when we recognize that a server doesn't usually go down just once. While these minor inconveniences may fade in the memory of users, they typically don't fade from your system's event log. You might be surprised to find that a particular server fails twice per year, in which case we would double all the previous numbers (which still don't include hardware or services costs). But even at one failure per year, if we presume that a typical server asset is expected to have a 3-year lifespan, then we should multiply the per outage cost times the number of outages per year times the number of years the resource will be in service.

$$\text{Total Cost per Server} = C_o \times O_{\text{per year}} \times LS$$

$C_o$  = Cost per Outage = the result of our earlier formula)

$O_{\text{per year}}$  = Number of Outages per year (let's presume only 1 per year)

$LS$  = Expected Lifespan of Server (typically 3 years)

$$\text{Total Cost per Server} = \$8,710 \times 1/\text{yr} \times 3 = \$26,000$$

Here is the punch line: the file server that has been recently purchased and deployed to service the inside sales team of our company is considered reliable and well managed; so it is presumed to only suffer one outage per year. With that in mind, the company should plan to lose over \$26,000 over its lifetime of service.

That is the BIA for this one server. It took a while in this chapter to break this down, but in real life, this goes quicker than you might expect. Essentially, as you are looking at what kind of protection or availability solutions that you might consider per server or application platform, you first need to understand what kind of risks you are protecting against as well as the financial impact if one were to occur.

## Risk Mitigation: Fixing It in Advance

You might be thinking, This won't happen to me because I have <fill in the blank>.

You might be right. Risk mitigation (RM) is the set of steps you do to avoid the more common or anticipated types of crises. In data protection and availability terms, this might be as simple as mirroring your hard drives or as complex as deploying replication software with failover capabilities between multiple geographic sites.

### RISK MITIGATION IS A CORE GOAL FOR THIS BOOK

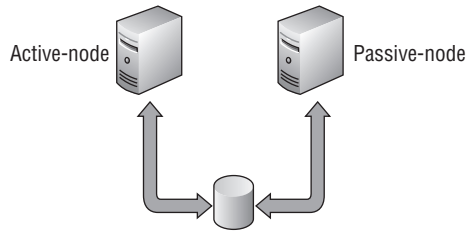
The technology chapters of this book are intended to make you successful in deploying the technologies and methods necessary to avoid productivity outages and data losses. To do that, Chapter 1 describes the landscape of what kinds of options are available, while this chapter focuses on the financial impact that we are trying to avoid. In this section of the chapter, we will discuss determining the appropriate level of protection or availability, based on the business impact, in order to mitigate our risks.

## Protection or Productivity?

Usually, you cannot solve for both protection and productivity equally. There are exceptions, and most people desire both. But for most technology approaches, the solution is optimized for either addressing robust protection and recovery scenarios or ensuring higher availability. Certainly, it can be as fine a line as 60:40 or completely exclusive to one goal or the other.

One example is Microsoft Cluster Services (MSCS), as shown in Figure 2.2. MSCS is exclusively a high-availability solution and makes no pretenses about data protection. In fact, MSCS typically has a single point of failure in its shared storage architecture.

**FIGURE 2.2**  
A Microsoft MSCS cluster for higher availability (productivity)



In Figure 2.2, we see two physical nodes, each running Windows Server and attached to a shared storage array. MSCS creates a logical layer between the independent operating systems and forges an identity as a single *cluster*. Applications requiring high availability are installed onto the cluster, whereby the application (as well as a virtualized identity of the server) may be running on either one clustered node or the other. If anything happens, whether at the application, OS, or node–hardware layer, the virtualized application server moves the entire instance to the surviving clustered node for assured uptime.

The single point of failure for a cluster, its shared storage solution, is made up almost entirely of disks, which we have previously discussed as the computer component most likely to fail. If the storage array fails, the cluster is left as two heads with no body. To mitigate this risk, clustered nodes often use mirrored storage arrays behind the scenes. We'll discuss this more in Chapter 6, but for now, consider it an example of a solution focused exclusively on productivity, whereas traditional backup focuses on protection.

## Availability

Availability is best handled in the platform that you are trying to make highly available.

The original method of ensuring higher availability was deploying more reliable storage, in the form of synchronously mirrored disk arrays. In those early days of LAN servers, this approach made a great deal of sense, since the majority of server outages were due to hardware and storage often had the components most likely to fail. Times have changed. Today, only a minority of crises are due to hardware, and not all of those are related to storage. Because of this evolution, most availability goals are now met by software-based solutions instead of expensive, redundant hardware. Even in those cases where redundant hardware is leveraged, such as mirrored storage, it is usually done within the larger implementation of clustering or other application or OS availability configuration.

The next leap forward in higher availability was generic cluster services or application-agnostic failover. In the case of MSCS, the idea was to create a highly available and a virtualized server that could be run in any of the nodes of the physical cluster. The model then became to install your

production server applications, such as SQL Server or Microsoft Exchange, within the clustered server instead of physical hardware. In those early clustering days, this approach was hindered by two key factors:

- ◆ Many production server applications were not originally designed to be “clusterable.” This often meant additional engineering and complexity for implementing the application into the cluster. You might install the entire application from node one and then install part of the same application repeatedly on nodes two through *N*. In other cases, you might have to run a customized script to force the application into a cluster. The latter approach was especially common in third-party alternatives to MSCS such as Veritas Cluster Server (VCS).
- ◆ The original releases of MSCS also had some limitations and complexities that often made Windows Server experts feel like novices. This was especially true in Windows NT 4.0 and Windows 2000.

In Chapter 1, I shared the anecdote that an application or an OS vendor often looks externally to partner developers to fill feature holes that it cannot initially deliver, but if the feature is requested enough by customers, it will usually be filled by the original developer. In this case, we see that availability for today’s applications is not relegated to hardware, nor is it delivered by changing the OS environments on which the application was originally intended. Instead, primary applications are now delivering higher availability within their own technology. Examples of this include SQL Server database mirroring, Exchange 2007 CCR and 2010 DAG, and Distributed File System (DFS) namespaces and replication. We will cover each of these specific technologies in future chapters on availability, as well as the application-agnostic approaches to availability, including MSCS and third-party software.

**NOTE** Whenever possible, application availability is (usually) best achieved by the application itself.

## Protection

Data protection and recovery is exactly the opposite of data availability with respect to where it should be delivered from. The reason that availability is best achieved through the application itself is because availability is still about the delivery of the original service. Similar to software debugging and error handling, availability mechanisms are part of ensuring the delivery of the application, whether that be servicing mailboxes, offering databases, or sharing files.

For example, consider Microsoft Exchange Server. In much the same way that the Exchange development team has made significant investments to improve the integrity of its own database within Exchange Server, their next development in Exchange 2007 was additional availability solutions such as LCR, CCR, and SCR, as well as DAG in Exchange 2010 (all of which are covered in Chapter 7). But in principle, these new technologies are all additional layers of availability, as they all are focused on providing you with current or near-current data. To achieve any other (previous) recovery point, you must stop looking at availability technologies and start looking toward protection mechanisms. But protection, in the sense of data backup and recovery, is not built into Exchange as availability solutions are. Exchange now enables its own high-availability scenarios but requires outside mechanisms to gain protection and recovery.

Similarly, SQL Server has replication for the purposes of availability but does not include the concept of built-in backup and recovery in the traditional sense. Instead, both of these applications have made an appreciable investment to facilitate data protection via external means.

The key point in this protection and availability section is to recognize that you may already have deployed some level of risk mitigation in the form of secondary availability or backup and recovery technology. That’s what a good part of this book is about—identifying and successfully deploying those technologies that are appropriate for your environment.

**NOTE** In contrast to application availability, application protection and recovery is (usually) best achieved outside the platform itself.

## Total Cost of Ownership

So far, in this chapter, we have discussed data protection and availability technologies in terms of cost, meaning the business cost of not doing something. There is, of course, the factor of price, which is different depending on to whom you are talking.

In this section, we want to recognize that the price is always more than the sticker or invoice. In fact, in many backup and recovery scenarios, the greatest contributing cost has nothing to do with the product at all but is labor. Let’s consider a traditional nightly tape backup solution.

The initial acquisition costs might include:

Backup server (software)	\$2,500	
Backup agents (software)	\$995	per production server
Backup server (hardware)	\$2,500	
Tape backup drive (hardware)	\$2,000	

Assume a traditional mid-sized company network with 25 servers.

Collectively, then, to purchase a nightly tape backup solution for this environment, you might be requesting \$37,000, not including deployment services.

That is our first mistake, because you will pay for deployment, even if you do it yourself. If you contract a local reseller or backup specialist, there is likely a fixed cost for the deployment, which hopefully also results in a fast and reliable solution, because presumably the reseller has previous experience and close ties with your backup software vendor. But as anyone who has ever done a significant home improvement project will tell you, while you might choose to save the additional labor, you will pay for it in time—literally. Your own IT staff, who would otherwise be doing other projects, will be deploying this instead. The project will likely take longer if your staff has not deployed this particular technology before, and their not following best practices may result in additional labor at a later date. But presuming that everything is equal, let’s assume 8 hours for the server deployment plus 30 minutes per production server for agent installation and backup scripts configuration. Splitting the difference between an in-house IT professional at \$75/hour and a local reseller, which might charge \$250 per hour, this results in 20 hours, which we could equate at approximately \$150/hour, or an additional \$3,000 total labor.

But we aren’t done yet. We should also calculate the cost of media. If we assume that each of the servers has 5 TB of storage, then we would have 125 TB of active storage across the environment. At an average 60 percent utilization rate, we would need approximately 75 TB of data to be protected. With an aggregate daily change rate of 5 percent (more for applications, less on file shares), you’ll be writing about 4 TB of new data per day—but with most tape backup software, you’ll use a different tape for each daily job, plus 4 weekly tapes and 12 annuals. Conservatively,

this puts you at 20 tapes at \$100 each for an additional \$2,000 in tape media (not including additional costs like offsite storage or services, which we discuss in Chapter 12).

There will also be ongoing costs such as power, space, and cooling. Space would be associated with your facilities costs, but simply running the new backup server in standard form factor might use a 500 W platform (plus the tape drive's 200 W). The monthly power costs for this server alone is:

$700\text{ W} \times 24\text{ hours per day} = 16,800\text{ WH or }16.8\text{ KWH per day}$

$16.8\text{ KWH} \times 30\text{ days in a month} = 520.8\text{ KWH per month}$

At \$0.06 per KWH, this server will cost \$31.20 per month or \$375/year.

We also need to add in the ongoing labor costs for:

- ◆ Rotating the tapes on a daily basis, which isn't a lot, but perhaps 10 minutes per business day
- ◆ Checking the backup jobs, 10 minutes per business day, plus one one-hour error resolution every 2 weeks

Those aren't significant numbers when looked at that way, but when added up, we see 8,220 minutes, or 137 hours, or 3.4 working weeks per year, just managing backups (and assuming most things go right most of the time) and not including restores. The labor for managing backups in this environment will consume at least a month of every year with no productivity benefit and will cost \$10,300.

This gives us the bigger picture, the total cost of ownership (TCO):

The initial purchase price of our backup solution might be \$37,000, plus \$3,000 to install it.

But the operational costs in the first year will be an additional \$12,800.

Assuming that most hardware and software assets have a presumed lifespan of three years, we can add software maintenance (15 percent), upgrade labor (half of deployment), and new annual plus daily tapes (5 annually) for years two and three. The ongoing costs for the second and third years are \$6,100 annually.

Thus, the TCO for this backup solution would be \$65,000, which is nearly double what the initial purchase price was and does not include any restores at all.

## Return on Investment

If TCO is thought of as the bad number to consider in any financial assessment, then return on investment (ROI) would be the good one. Dig way back to the beginning of the chapter to BIA: how much does the problem cost?

If a problem costs \$150,000, we can assume that that is lost money. But if you solve the problem, the company gets \$150,000 back. Think of it like an ante in poker or a coin dropped into a slot machine; that money is gone. If you make any money from poker or slots, then that is positive—winnings. Of course, if you bet \$5 and then later won \$5, you haven't actually won, you've broken even. Similarly, if your technology problem or vulnerability costs \$150,000 and you got it back by spending \$150,000 on a protection solution, then you haven't actually solved the problem of losing the money for the business—you've just chosen to spend it in a different way. That may be okay to your CFO, based on tax rules, but that's outside the scope of this book.

If you spent \$65,000 (TCO) to solve a problem that will cost the company \$150,000 (BIA), then you have solved the problem. You literally added \$85,000 to the company's bottom-line profitability because they otherwise would have lost those dollars due to the outages that you mitigated. This is where ROI comes into the picture: how much you saved or gained for the company, in comparison to what you had to spend to accomplish it.

## Calculating ROI

There are different ways to quantify ROI. You might prefer to think about it as we discussed earlier, where you saved the company \$85,000. Taking servers completely out of the discussion, if you could show your accounting manager that they are used to spending \$150,000 per year on something but you could save them \$85,000 by doing it a different way, that is usually an easy business decision.

Some measure it as the percentage of BIA/TCO. In this case, \$150,000 divided by \$65,000 yields 2.3—or a 230 percent yield. Others invert the percentage (TCO/BIA) as the percentage of the problem that you are spending to solve it. In this case, you can spend 43 percent of the problem to resolve it. That also means that we save 57 percent of our projected losses.

Alternatively, you might think in terms of payback windows. If a problem costs \$150,000 over the three-year lifespan of the asset, then consider how long into that window before the solution pays for itself. In this case, with an average of \$50,000 costs annually, the first-year cost of \$52,800 is basically breaking even, but years two and three go from \$50,000 to \$6,000 annually, saving almost everything.

### TIME TO VALUE

Somewhat related to the ROI of a solution is how quickly you will start to see the benefits of the solution you are deploying. When considering that you will see  $x$  dollars over the lifespan of the project; look also at when you will see those dollars.

Compare when the costs are to be incurred to when the savings will start to be realized. Will you just break even for the first year and then see gains in the second and third years (such as when you deploy a new component that will solve an ongoing problem)? Or will you see gains the first year but fewer gains in later years, as you postpone a problem or take on incremental costs throughout the project?

How else you might use (and grow) the earlier money can also affect the overall costs for the project.

The actual calculation for ROI is to take the net gain (\$150,000 minus the costs of \$65,000) of \$85,000 and then divide it by the costs, after which you can multiply it by 100 to arrive at a percentage:

$$(\text{Total Gain} - \text{Costs}) \div \text{Costs}$$

$$(\$150,000 - \$65,000) \div \$65,000$$

$$\$85,000 \div \$65,000 = 1.31, \text{ which is } 131 \text{ percent ROI}$$

Any positive ROI is a relatively good decision and any negative ROI is a relatively poor decision. Consider a \$10 problem:

- ◆ Spending \$6 to solve a \$10 dollar problem is good because  $(\$10 - \$6) \div \$6 = \$4 \div \$6 = 0.66$ , or 66 percent ROI. Said another way, for every \$1 that you spent in this way, you would get it back as well as an additional 66 cents.
- ◆ Spending \$9 to solve a \$10 dollar problem is not as good because  $(\$10 - \$9) \div \$9 = \$1 \div \$9 = 0.11$ , or 11 percent ROI. Said another way, for every \$1 that you spent in this way, you would only gain an additional 11 cents. There are likely other ways that the business could invest that dollar and gain more than 11 cents in return.
- ◆ Spending \$12 to solve a \$10 dollar problem is obviously not a good idea:  $(\$10 - \$12) \div \$12 =$  a negative 12 percent ROI. Said another way, for every \$1 that you spent, you lose 12 more cents than what the original problem was already costing. It would (obviously) be cheaper to live with the \$10 problem than to solve it for \$12.

The third example may have been overly obvious, but sometimes IT administrators do solve \$10 backup or availability problems with \$12 solutions because they do not understand the BIA or TCO well enough, or because they are not aware of the \$6 alternative solutions.

### Which ROI Method Is Most Accurate?

This chapter has been about converting technology issues into quantitative, and specifically financial, assessments. Once you have converted your protection or availability problem and potential solution(s) into this financial language, you can easily convert it from one denomination (ROI metric) to another, as easily as converting the denominator of a fraction by multiplying or dividing it by a common number.

But there is a rule of thumb worth noting:

25 percent ROI may be better than 60 percent ROI

One of the reasons that I prefer to deal in actual dollars is because CFOs and other accounting types can often crunch the numbers to their own liking, once you present two key numeric facts (though they must be defensible facts and not subjective opinions):

- ◆ The problem is currently costing the company \$XX,XXX. (BIA)
- ◆ I can solve the problem by spending \$YY,YYY. (TCO)

From there, some will subtract one from the other for savings, whereas others will find a ratio that helps them appreciate it. However, based on some anecdotal findings from surveys and the experience of many years in supporting sales efforts, there is a credibility concern to be aware of.

### The Credibility Challenge of ROI

Notwithstanding the recognition that every technology vendor (or other sales organization) always preaches how wonderful their widget is and how amazing their ROI (often unfounded) could be, ROI does have a credibility challenge.

Using the percentage ROI method, let's assume that the ROI of a solution is 43 percent, meaning that we are spending \$70 to solve a problem that costs \$100. The challenge is that the solution is costing over half of what the problem costs. That means that if your assessment of the cost of

the problem is perceived as too high (qualitatively, not necessarily quantitatively) or you may have underestimated something in your TCO, then your ROI goes down from 43 percent as your costs start getting closer to what the problem itself costs. If your CFO is willing to wager that a problem won't happen as often as you project, she might actually save money (or at least break even) by just allowing the problem to happen on a (hopefully) less frequent or less impactful nature than you have predicted. The project does not have enough ROI to warrant the initial expenditure.

On the other hand, what if you only needed to spend \$5 to save \$100—1,900 percent ROI? This has the opposite challenge: it sounds too good to be true. If you have a good amount of credibility with the financial decision maker, then you will be seen as a hero and your project will be approved (although with that much credibility between you and your CFO, you may not have calculate a specific ROI to begin with). For the rest of us in reality-land, if it sounds too good to be true, some financial decision makers will assume that it is not true (or viable as a “real” solution). There must be some significant cost factor that is either drastically inflated in the problem or underestimated in the solution. Either way, the solution is not perceived as credible. After all, how likely is it that you can purchase a mere toy to solve a real problem?

Based on anecdotes, experience, and a few old surveys, it appears that 20–25 percent ROI is the best way to justify a solution. The gain is enough that the solution is likely worth pursuing, though the investment is substantive enough that the solution can be considered reasonable for addressing the issue. Using this approach, we might consider the following ROI boundaries:

- ◆ Over 33 percent may lack credibility.
- ◆ Under 15 percent may not offer enough potential gain.

One the most interesting pieces of advice that I ever heard related to ROI was from someone at a Gartner CFO conference who attended a session on ROI. They heard that if a significant proposal was submitted for review and it had a TCO projection and ROI analysis on its first submission, it would be approved over 40 percent more often than those that did not have those calculations. If the same type of proposal was pushed back down to get the TCO/ROI analysis and it was resubmitted, it only had a 15 percent greater likelihood of approval over similar projects without one. The first ROI success tip: present the TCO and ROI assessment with the initial proposal, as it not only clarifies the legitimacy of the project to you, but also proactively clears a big hurdle for you with those who guard the dollars.



## Real World Scenario

### YOU SHOULD HOPE THEY ARGUE WITH YOUR FORMULA

This is my own advice, and I have never had someone challenge it. The best thing that can happen when you present your methodology and resulting BIA/TCO/ROI justifications for a project is that the business/operational/financial stakeholder challenges your formula (in a constructive way). When working with your business leaders and establishing the formula that you will use in your process, here are a few key ideas to frame the conversation:

- ◆ Working backward, ROI is just a comparison of BIA to TCO.
- ◆ TCO is simply a prospective invoice, along with some simple assumptions of fixed costs. Likely, challenges here will be minor tweaks to the fixed values, not wholesale changes to the math.

- ◆ BIA is where challenges occur—your business stakeholder doesn't agree with how you calculated the cost of downtime (one example from earlier in this chapter). This is great news because then you two get to decide why the formula doesn't apply to a particular business unit or technology resource.

If your discussion circle can collectively agree that when the database server is down for up to a day, employees can catch up on email, or vice versa (and thereby reduces some variable by half), then the collective team has turned your formula into their formula.

If the HR person can provide more specific hourly dollar values across a large department (though you are unlikely to get a list of individual salaries), your team now has much more accurate fixed values that both the IT management and the operational management will agree on.

In short, every pushback that can be discussed or refined brings buy-in and agreement by the other parties. When you have five variables to work with, the formula may seem academic. But if you get more accurate modifiers, and the dollar variables are filled in with real numbers, you are only left with the technology numbers, such as:

- ◆ How often does the server go down?
- ◆ What is the cost of replacement hardware?
- ◆ How much do tapes cost?

These numbers are usually easily accessible by IT management and complete the equation. From there, you now have a new BIA that is even more defensible and that now has credibility in the eyes of the other stakeholders. TCO comes from the invoice and projections. ROI is simply the mathematical comparison of the BIA and the TCO.

But now, because everyone has weighed in on the financial values and the relational impact of the formula, everyone believes the ROI, no matter how big or small. Going back to the concern we had around presumed credibility of the ROI formula:

- ◆ If the ROI is less appealing (for example, TCO is 50 percent of BIA), at least everyone was involved in understanding the legitimacy of the numbers, and you have a greater likelihood of them agreeing to the project.
- ◆ If the ROI is too appealing (not emotionally credible), you have the simpler problem of working with the vendor through side meetings to educate your stakeholding peers as to the legitimacy of the solution and the higher potential of being that hero by spending \$10 to save \$100.

Either way, having the initial formulas and variables challenged turns the project from yours to theirs and will help you pay for what you already know you want.

## Turning IT Needs into Corporate Initiatives

High availability (HA) and backup and recovery (B&R) are technology terms. In many companies, they are considered similar to taxes to the budget. No one likes the time or money spent on backup until they need a restore. Most folks think availability solutions cost too much until they are in the middle of an outage. But as logical as these tactical initiatives are, they often are among the first to suffer during budgetary sacrifices.

Business continuity (BC) and disaster recovery (DR) are usually considered strategic, not tactical. More importantly, they are often funded by higher-level organizations and typically have VP-level or C-level executive sponsorship. Your company may even have a Chief Risk Management Officer (CRMO). Although these initiatives are often included in the early budget-chopping process, it is usually for different reasons. BC and DR initiatives are often unwieldy, especially in their first year or two of delivery. They are often considered too expensive or too complex to deploy and maintain. As such, they are often put off until “next year.”

One key to success is to look at how your HA or B&R solution contributes to your company’s BC and DR needs.

If your BC goal has a guaranteed system uptime requirement, shape your HA deployment within that context, even to the point of calling it Risk Management, which is definitely part of most BC technology plans.

If your DR goal requires data to be offsite, how are you going to get it there? Would it come from your backup tapes? Would implementing a disk-to-disk replication solution give you your offsite capability without courier services for the tapes?

The key is twofold:

- ◆ Frame your HA or B&R project within the company’s BC and DR goals so that you get higher executive sponsorship, which will result in friendlier financial and operational stakeholders when you are calculating BIA, TCO, and ROI.
- ◆ You may be able to pay for your new backup solution from the company’s DR budget (instead of your IT budget), if you can show that the solution facilitates a desired capability for disaster recovery.

## Summary

That’s it for this chapter; the big takeaways are to:

- ◆ Look at RPO and RTO as a way to distill different data protection and availability technologies into consistent and comparable performance metrics.
- ◆ Understand what kinds of crises that you are solving using RA and BIA.
- ◆ Most importantly, convert your technology issues into dollars (BIA), and get everyone on the same page for how much you need better availability or protection.

Then, you can create a fair assessment to decide what you need. From there, understand the real costs (TCO) including acquisition, and be proactive in communicating the benefit (ROI), not just the needs.

Assuming that you’ve done all that, the rest of this book is intended to help you be successful in selecting and deploying different technologies in protection and availability.