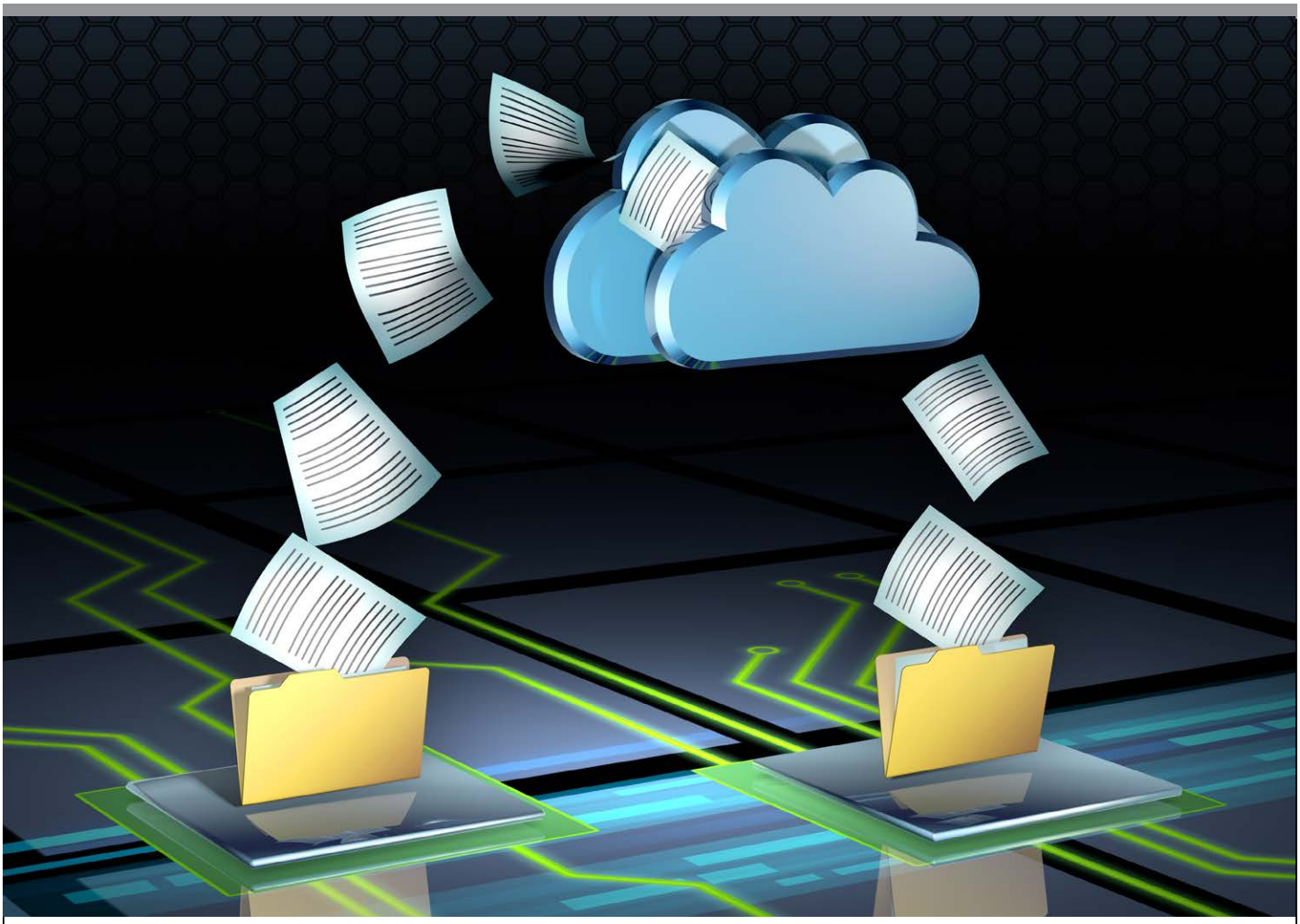
CONTENT PROVIDED BY

Redmond VIRTUALIZATION REVIEW

BACKUP & DISASTER RECOVERY SPOTLIGHT

The essential guide to preparing for the worst. Sponsored by *Virtualization Review* and *Redmond* magazines.

- Backup vs. Replication (and Why You Need Both) *Page 1*
- 3 Common Challenges with Cloud-Based DR *Page 7*
- Disaster Recovery and the Question of Balance *Page 14*
- Disaster Recovery Planning for Hyper-Converged Infrastructure *Page 20*
- Business Impact Analysis: The Key to Successful Continuity Planning *Page 27*
- The Quest for Guaranteed Recovery Assurance *Page 34*



Backup vs. Replication (and Why You Need Both)

Some see them as competing technologies. The reality is that they go together like peanut butter and jelly.

By James Green

Disaster recovery (DR) is one of the most critical insurance policies in which a business can invest. A high-quality disaster recovery strategy can be the difference between a minor speed bump and a business-ending tragedy. As such, it's important for business leaders to understand the components that make up a well-rounded DR plan and know how to leverage each to meet their business requirements.

Definitions

Before reviewing the technical methods for protecting a business from disaster, the foundational business measurements must be understood. In many businesses, a standard set of measurements is defined that makes certain promises to the rest of the business with regard to the availability of services. Although there are others, there are three types of measurements that commonly appear in a DR conversation.

Service-Level Agreements

Frequently abbreviated as SLA, a service-level agreement is a contract of sorts that establishes the scope and quality of a service that will be provided. Sometimes this sort of agreement is negotiated between a third party and the business, like a public cloud provider, for example. It can also define goals for services offered to a customer, like a help desk or call center. In the context of DR and this article, SLAs are also internal—and often informal—contracts between IT (or other departments) and the rest of the business. These agreements define things like:

The sole purpose of crafting a sound disaster recovery strategy is to ensure that SLAs are met.

- To what level business data will be protected from infrastructure failure
- How long business data will be retained in archives
- How "available" business services will be (the amount of downtime that will be tolerated)
- The granularity with which lost data can be recovered

SLAs are very important in the context of DR, because they define the targets for the entire DR plan. The sole purpose of crafting a sound DR strategy is to ensure that SLAs are met. If SLAs are fairly aggressive, a robust, multi-faceted DR strategy might be required in order to meet the demands of the business. But if SLAs are fairly lax, perhaps a simple nightly backup will suffice. This is why SLAs are so important to define.

Recovery Point Objective

Recovery point objective (RPO) and the next metric, recovery time objective (RTO), could actually be part of an SLA. But they're important to call out on their own because these two measurements are two of the most important in determining which technical measures to take to meet the requirements. RPO establishes the amount of data a company is willing to lose in the event of a disaster. This is a

delicate balance, but easily calculated with access to the right numbers. Simply, it comes down to finding the break-even between the cost of lost data and the expense to ensure against losing data. When it costs more to bolster the data protection mechanism than it costs to just lose the data, you've found the right point. Of course, this is never an exact science because it's effectively insurance and failures aren't predictable; theoretically, a business could have no failures and have "wasted" money on DR. But if a business finds itself on the other end of the spectrum and has *many* disasters that were all gracefully recovered from, the investment in the DR mechanisms more than paid for itself.

The break-even between the cost of lost staff productivity or customers served and the cost of the DR mechanism is the sweet spot.

RPO is measured in time. For example, we're willing to lose 15 minutes worth of data. This means that all data that is older than 15 minutes must be adequately protected and restorable within the RTO in the event that the primary copy of the data or the services becomes unavailable.

Recovery Time Objective

RTO balances the DR equation and specifies the amount of time after the failure that will be tolerated before the service or data is restored. This metric is measured in units of time just like RPO, so an RTO of 15 minutes for a specific service indicates that after a failure, the data or service must be available again within 15 minutes at the RPO (maximum amount of data loss) expected.

Similar to calculating a reasonable RPO, calculating a reasonable RTO just requires the correct inputs. The break-even between the cost of lost staff productivity or customers served and the cost of the DR mechanism is the sweet spot here. Combined, RPO and RTO will make up the total amount of loss in a DR scenario. Sometimes, they'll end up being symmetrical, like the example I've been using: an RPO and RTO of 15 minutes. In some business cases, like a business that processes a high volume of transactions every minute, both RPO and RTO should probably be high. In other cases, like a video production company, a short RPO but a longer RTO could possibly be tolerated.

With an understanding of the business targets and the way they're measured, it's possible to logically evaluate the technical methods for achieving these objectives. Now I'll explore the difference between backup and replication and see how they relate to RPO and RTO goals. While there are technically many more components of the full

DR strategy, these are two of the most widely leveraged tools in the DR mechanic's tool chest.

Start with Backup

Backups are the cornerstone of any DR solution, and have been around since the days of punch cards. For as long as you've been storing data, you've been wise enough to keep backup copies of that data. Today, backups can be considered the first or second line of defense against data loss, depending on how you look at them. From an infrastructure perspective, backups are really the second line of defense; the first is infrastructure resiliency and fault tolerance. Should that fail, restoring backups could happen next.

Backups and replication are complimentary, accomplish different goals and should be used together.

But failure isn't the only cause for data loss; you must also consider possibilities such as user error. In the case that a user manually but unintentionally deletes a file that they really wanted to keep, backups are the first line of defense in recovering that file.

One of the key points I want to make in this article is that backups and replication are complementary, accomplish different goals, and should be used together; it's not an "either/or" discussion where just one of them will solve the problem. As I mentioned, the only purpose for any technological approach is to meet the defined SLAs of the business. With that in mind, the question at hand is: How do backups help achieve RPO/RTO goals? And how are they configurable to meet different levels of RPTOs (an abbreviation used to discuss both goals at once)?

For the purposes of this article, a backup will be defined this way: "Backup is the activity of copying files or databases, so that their additional copies may be restored in case of a data loss accident." Backups take more of a "long term" approach to protecting the datacenter.

Backups store data in a holding area to meet the sort of SLA that says, "We will be able to recover data from a specified data set from any point within the last 30 days with a 15-minute RPO." This SLA states that unless the data was altered 10 minutes ago or 31 days ago and is outside the scope of the RPO, it will be recoverable.

Archival: A Subset of Backups

It's worth mentioning that backup data is treated in two different ways. Primarily, backup data is available to meet production

workload RPTO requirements and will be kept available as such. But many organizations also have regulatory mandates to retain data for a certain period of time, and some choose to retain data for a long time, even though they aren't required to, just in case they would ever need it.

This data that could be years old is commonly referred to as "archive" data, and might be treated as another tier outside of the production backup system. While archived data might take longer to retrieve and restore, it is still a part of meeting SLAs and a part of the overall backup strategy.

While archived data might take longer to retrieve and restore, it is still a part of meeting SLAs and a part of the overall backup strategy.

Complement with Replication

Replication is the act of synchronizing data between a primary site and a secondary/DR site for the purpose of resiliency. Oftentimes, replication is viewed as superior to (or a preferable alternative to) backups. This is not the case, however. As **Table 1** shows, while replication can help shorten RPTO targets, it becomes prohibitively expensive to have long retention periods where a recovery can take place within the RPTO target time frame and *backups* are the key for retention.

Recovery Metric	Backup	Replication
RPO	24 Hours	15 Minutes
RTO	2 Hours 0 Minutes	15 Minutes
Retention	Long	Short

Table 1. Comparing backup and replication

The trick, therefore, to creating a successful DR strategy, assuming RPTO goals are somewhat aggressive, is to leverage *both* backup and replication (among other tools), rather than one or the other. In fact, for comprehensive protection, they should even interact with each other. Take Figure 2 as an example. This is one way that backup and replication tools could be used together to create a DR strategy that covers more of the exposure to risk.

In the example, production data is being backed up on a regular basis to a local backup repository. These backups are restorable within a short period of time, and data is kept in this local repository. Any file needed is recoverable with no dramatic action taken. The primary site is also protected by replication such that in the event of a site-level disaster, the workloads could be failed over to the DR site

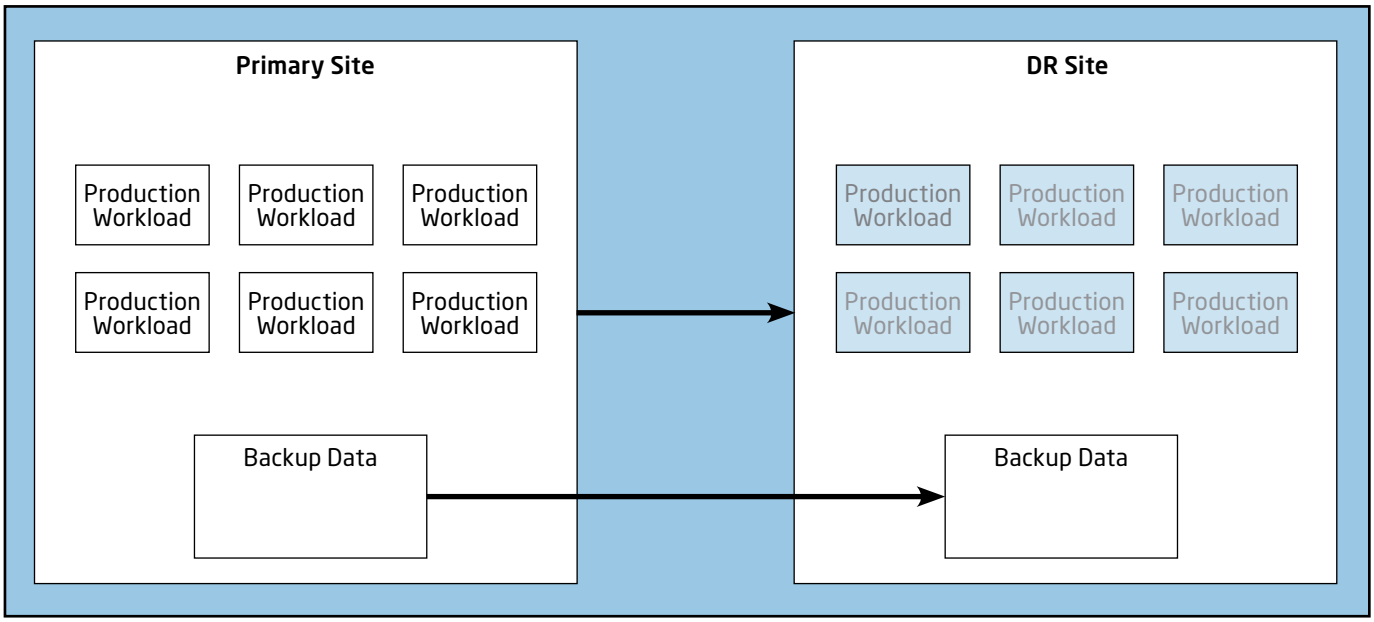


Figure 2. Backup and replication working in tandem provides the best solution in most cases.

and critical services could be back online in roughly 15 minutes. Although a failover is more dramatic, it has a low RPTO and can restore services in short order.

Working together is where backup and replication really shine.

Working together is where backup and replication really shine. Near the bottom of Figure 2, backup data is being replicated to the DR site. This means that not only is backup data with long retention available for restores, but it's highly available because it's replicated to the DR site. Imagine that a tornado wiped out the primary site, but there were no backups and only replication was being used for high availability. In this sad scenario, the business would be up and running with the data that was available at failover, but old data would not be available for restore as it was destroyed in the disaster. This illustrates why the use of both technologies together is commonly the best strategy. VR

James Green is a Partner at ActualTech Media and writes, speaks and consults on enterprise IT. He has worked in the IT industry as an administrator, architect and consultant, and has also published numerous articles, white papers and books. Green is a 2014-2016 vExpert and VCAP-DCD/DCA.

3 Common Challenges with Cloud-Based DR

When considering a move to the cloud for your precious data, keep these gotchas in mind. By James Green

Backup and disaster recovery (DR) has been a critical part of the datacenter for decades, and that won't be changing anytime soon. No matter what business you're in, there's a pretty good chance that a loss of either uptime or data would be costly to your business.

As a protection against natural disasters, human error, equipment failure and anything else that could go wrong, wise businesses invest in the "insurance" of a DR solution. Historically, that solution has been relatively primitive, yet effective. But in the post-Internet era, with high-speed connections to shuttle data anywhere on the globe, disaster recovery solutions have the potential to be quite extravagant and complex. Due to the cost of downtime in some organizations, the cost and complexity is sometimes justified.

Now, in a day and age where anyone can take a credit card and begin to build out a production-grade application in a world-class datacenter in mere minutes, some businesses are starting to leverage that same flexibility for their DR practices. For the purpose of this article, I'll refer to this practice as "cloud-based DR." Although that theoretically could mean something to do with availability within a private cloud platform, I'm using the term to refer to the use of *public* cloud resources from a provider like Amazon Web Services Inc. (AWS) or Microsoft Azure to be the "DR site" for a DR plan.

Traditional disaster recovery has been—in most cases—mundane but effective.

Why Cloud-Based DR?

Traditional DR has been—in most cases—mundane but effective. As long as it's maintained and the processes are followed to the letter, DR hasn't been too difficult. But there are a handful of common experiences among organizations that suggest that there's a motive for a new kind of DR:

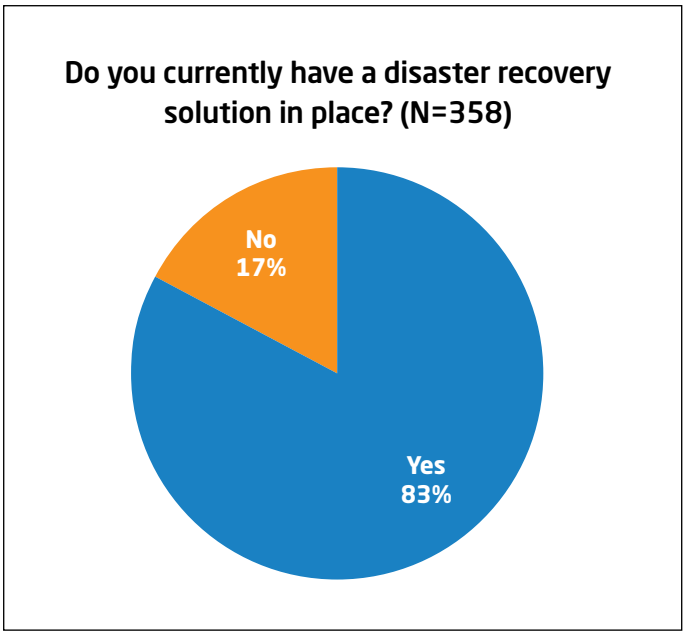


Figure 1. Survey results regarding disaster recovery solutions.

- Organizational growth and change causes changes to the infrastructure. The backup strategy can be overlooked until it's too late and data is lost.
- The sheer scale of certain organizations causes the "insurance" of disaster recovery to be very expensive.
- Refreshing DR infrastructure is a slow and laborious process.
- As seen in **Figure 1**, which is based on research results from a recent DR as a Service (DRaaS) market report sponsored by Infrascale Inc., too many businesses still don't even have a DR solution. This is likely due to cost and complexity concerns.

In light of these challenges, many businesses are looking at the agility that comes with leveraging public cloud resources and wondering, "Couldn't we use that for disaster recovery?" The answer is a resounding "Yes!" And it comes with a number of benefits. Some of the reasons an organization might choose to adopt a cloud-based DR strategy include:

- A reduced datacenter footprint means less expense from hardware, maintenance, utilities and operations staff. From the business's perspective, the environment is purely logical, so the overhead to maintain it is far less.
- Increase flexibility by shifting DR spend to an operational expense (OpEx) model. Because of the way cloud-based DR resources are purchased, there's little or no capital investment at the outset. The business is billed for usage on a monthly basis, just like a utility (power, water, gas and so on).
- Because the business is only billed for what's in use, the spend on DR resources can be *substantially* less than if the business purchased hardware, space and staff to run an entire second datacenter all the time. Because resources can largely be left powered off until disaster strikes, the majority of money is theoretically only spent in the event of a true disaster.

In the post-Internet era, with high-speed connections to shuttle data anywhere on the globe, disaster recovery solutions have the potential to be quite extravagant and complex.

Unfortunately, it doesn't always work out as simply and effectively as that. As a matter of fact, a poorly designed, poorly understood cloud-based DR strategy can actually end up costing *more* in the long run than a well-designed on-premises DR solution. So what's the rub? What causes some organizations to fail miserably at implementing a cloud-based DR strategy?

Well, the possibilities are endless, but here are three common mistakes that businesses make when heading down this path. Understand and avoid these pitfalls, and your chances of success with cloud-based DR are an order of magnitude higher.

1. Underestimating Required Network Bandwidth

Because building a DR strategy around workloads running in a datacenter not owned by the company inherently involves moving data off-site, most organizations see this challenge coming and plan fairly well for their replication traffic. In a number of cases, there are many years of historical data from site-to-site replication that the

business can comb through to find out exactly what sort of bandwidth is going to be required.

Many vendors who sell cloud-based DR tools even have handy tools that analyze the state of an environment and give a pretty good estimate of what kind of network bandwidth will be required, given the inputs. The good news is that implementers typically get this right the first time. The bad news is that the bandwidth needed to complete replication and meet service-level agreements (SLAs) with regard to the company's recovery point objective (RPO) isn't the only bandwidth in question.

What are the chances that the business-critical applications were written with non-local user traffic in mind?

Imagine for a moment that a water pipe servicing the bathrooms next to the datacenter bursts. The datacenter floods and has to be shut down completely. Fortunately, this business has a cloud-based DR solution, and all the high-priority workloads can be recovered in the cloud. Within a matter of minutes, business-critical systems are online and ready to accept user connections.

From a network perspective, what happens now that could've been overlooked during the planning phase? *Every user is now accessing the applications over the WAN or VPN.* And what are the chances that the business-critical applications were written with non-local user traffic in mind? Slim. Many of the applications assume the user is on the local network, and thus have no measures in place to reduce the amount of network traffic they produce or consume.

This is a really bad situation. Although replication has been working quite nicely ever since the solution was implemented and bandwidth has never been an issue, when the time comes to push the big red button and save the day, the system falls flat on its face because there isn't enough network capacity to support the user traffic.

To avoid this ugly situation, be sure the planning phase of a cloud-based DR implementation involves not only calculations with regard to keeping the off-site data up-to-date and within SLAs, but also with regard to user traffic when an actual recovery is needed.

2. High Data Transfer Costs

When it comes to billing for public cloud solutions, the best way to describe it is: "Death by 1,000 cuts." It's a penny here, a penny there,

and next thing you know the monthly bill is $92,000, when it would have cost $45,000 to run the same workloads on-site. Without watchful oversight, an organization's cloud spending can nickel and dime it into a tough position.

Pretty much all resources are billed this way, but one fee that seems to really cause problems is the fee for data transfer.

Public cloud services providers charge a fee, both for data being ingested (entering their datacenter) and for data upon egress (exiting their datacenter). **Figure 2** shows an example of a pricing chart—in this case, for AWS EC2—for data transferred both in and out of the system. It looks very minimal—a bunch of zeroes and $0.01 charges.

Data Transfer IN To Amazon EC2 From	**Pricing**
Internet	$0.00 per GB
Another AWS Region (from any AWS Service)	$0.00 per GB
Amazon S3, Amazon Glacier, Amazon DynamoDB, Amazon SES, Amazon SQS, or Amazon SimpleDB in the same AWS Region	$0.00 per GB
Amazon EC2, Amazon RDS, Amazon Redshift and Amazon ElastiCache instances or Elastic Network interfaces in the same Availability Zone	
Using a private IP address	$0.00 per GB
Using a public or Elastic IP address	$0.01 per GB
Amazon EC2, Amazon RDS, Amazon Redshift and Amazon ElastiCache instances or Elastic Network interfaces in another Availability Zone or peered VPC in the same AWS Region	$0.01 per GB
Data Transfer OUT From Amazon EC2 To	**Pricing**
Amazon S3, Amazon Glacier, Amazon DynamoDB, Amazon SQS, or Amazon SimpleDB in the same AWS Region	$0.00 per GB
Amazon EC2, Amazon RDS, Amazon Redshift or Amazon ElastiCache instances, Amazon Elastic Load Balancing, or Elastic Network interfaces in the same Availability Zone	
Using a private IP address	$0.00 per GB
Using a public or Elastic IP address	$0.01 per GB
Amazon EC2, Amazon RDS, Amazon Redshift or Amazon ElastiCache instances, Amazon Elastic Load Balancing, or Elastic Network interfaces in another Availability Zone or peered VPC in the same AWS Region	$0.01 per GB
Another AWS Region	$0.02 per GB

Figure 2. AWS EC2 data transfer pricing chart, April 2016.

The real problem in many situations is caused by the fees tucked away at the bottom of the chart: fees for data transferred out to the Internet (not pictured in **Figure 2**). While everything else looks to be some harmless charges from zero to one cent, data transferred out to the Internet can cost between 5 cents (a discount for large amounts of data) and 15 cents, depending on the AWS region from which the transfer is taking place.

Although ongoing operations where data is transferred tend to be free or minimally costly, it can get expensive if an actual failure takes place. This is where the software you use to control failover, failback and replication can really make or break the strategy.

Although ongoing operations where data is transferred intend to be free, it can get expensive if an actual failure takes place.

A poorly designed solution will require a full copy of the data from the failover site to be replicated back to the primary site before the failback process can occur. This means that if you've failed over a 40TB datacenter to AWS because of a disaster, getting your data back on-site (assuming you want to keep the data that has accumulated during the recovery window) will require a full replication of the entire dataset. Not only will this potentially take ages, but it's going to hit you in the pocketbook. Replicating 40TB of data at something like $0.09/GB might not be the end of the world for some organizations, but in most cases it's an unexpected fee at the very least. And a fee like that in the case of a small business could be crippling.

This unforeseen challenge can't really be avoided; it can only be planned for. There are two primary ways to handle this situation. The first is to acknowledge that it's a reality and set aside an estimate of what it would cost to transfer the entire dataset back out as a part of the disaster recovery plan. Then that money is available in the event that a disaster ever occurs and the data needs to be recovered.

A much more palatable option is to leverage DR software that doesn't require a full replication of data before a failback. As this market has matured, most software vendors selling a cloud-based DR solution have taken this cost into consideration and provide ways to minimize the amount of data transferred out of the public cloud platform.

3. Not Clarifying SLAs

The final challenge that can sometimes be overlooked when implementing a cloud-based DR solution is that of the cloud provider's SLAs. It's important to consider that when you're leveraging another organization's services to guarantee or ensure the availability of your own, you've inherently put yourself at the mercy of that other company. Their promises in the way of uptime, availability and so on have a direct impact on your ability to provide adequate DR protection.

This is especially important to consider when pricing out a product, because as one would expect, there tends to be a tradeoff between cost and the SLA. For example, EBS (block storage from AWS) has a steep cost when compared to archive storage like Glacier (object-based archive storage from AWS), but the SLA for data stored within Glacier indicates that data will be available for retrieval within roughly 3 to 5 hours from requesting it.

Make sure to carefully weigh the tradeoffs you're making when settling on the provider and tier of service you will procure for your own cloud-based DR strategy.

This is a dramatic example that most organizations would catch and easily understand, but it clearly illustrates the point. When planning for a cloud-based DR implementation, the cost of 99.9 percent uptime versus 99.9999 percent uptime can be the difference between the cloud provider's datacenter being affected by the same disaster, and a flawless recovery that makes IT look like heroes. Make sure to carefully weigh the tradeoffs you're making when settling on the provider and tier of service you will procure for your own cloud-based DR strategy. **VR**

James Green is a Partner at ActualTech Media and writes, speaks and consults on enterprise IT. He has worked in the IT industry as an administrator, architect and consultant, and has also published numerous articles, white papers, and books. Green is a 2014-2016 vExpert and VCAP-DCD/DCA.

Disaster Recovery and the Question of Balance

Because one size does not fit all, smart enterprises consider a litany of factors. **By Dan Kusnetzky**

Disaster Recovery (DR) is a hot topic today. Despite that fact, one of the main challenges to a successful DR implementation is knowing exactly what it is. Vendors use DR as a blanket term when speaking about many different types of products, even though each of these products is doing something different.

One reason for confusion is that DR is really a combination of processes—risk analysis, planning, implementation and ongoing operation—combined with both hardware and software. They're jointly designed to respond to some sort of disaster quickly, reliably and efficiently, allowing business operations to continue.

What Constitutes a "Disaster"?

Disasters include a wide range of events, including:

- Natural disasters such as a fire, flood or storm.
- Hardware failures like power, air conditioning, systems, system components, storage networks, storage devices, network connections and network devices.
- Software failures such as poorly implemented applications, database failures, loss of messaging from one software component to another, or application-framework failures.
- Security issues like malicious injection of SQL code, corruption or loss of files.
- A wide range of human errors that can bring down even the best-designed application environments.

Approaches are presented as if they're really the whole enchilada, rather than just a few tortilla chips.

It's easy to see that planning for disasters involves many levels of business, facilities and IT management, as well as experts in systems software, virtualization technology, application frameworks, application development, database management, storage and networking.

I'm going to focus on the IT hardware, software and services elements. A quick examination of those areas reveals that vendors offering products that touch on *any* aspect of "keeping the lights on" will claim the full mantle of DR, even though the use of its product or service isn't a complete answer. These vendors often use catch-phrases such as "continuous processing," "always on" or "nonstop."

The Right Tool for the Right Job

Consider many of the different ways vendors address DR. Although few actually address all of an enterprise's requirements, each of the following approaches are presented as if they're really the whole enchilada, rather than just a few tortilla chips:

- Hardware components that support continuous processing; that means nonstop, fault-tolerant computers, such as the ftServer from Stratus Technologies. This approach focuses on the

underlying host systems and assures that end users will never see a failure. These systems are designed with multiple layers of redundant hardware and special firmware that detect failures and move processing to surviving system components. Failover takes only a number of microseconds and is automatic.

- Clusters of systems designed to detect slowdowns or failures and move applications and data to maintain continuing operations. Suppliers such as Dell Inc., Hewlett Packard Enterprise (HPE), IBM Corp., Microsoft, Oracle Corp., Red Hat Inc., SUSE and many others offer this type of DR solution. Cluster-software managers monitor the health of systems, applications, and application components, and move functions to another system when a failure or slowdown is detected. Applications typically must be designed to work with this cluster-software manager. Failover can take hours, depending on the design of the cluster manager. Systems typically are supporting workloads, and are there as warm standbys for all other systems.

Vendors offering products that touch on any aspect of "keeping the lights on" will claim the full mantle of DR, even though the use of its product or service is not a complete answer.

- Systems that house backup software. These are appliance servers pre-loaded with backup software. In this setup, applications and data are constantly backed up to either storage in the datacenter or to a cloud-storage service. Upon failure, this data can be manually or automatically recovered. While some products can detect failures and start a recovery process, many require manual intervention. Full recovery can require a number of days, depending on the complexity of the environment.

- Storage systems that keep multiple copies of each data item, making it possible for applications to continue accessing and updating these data items even though a component has failed. Suppliers such as EMC Corp., Hitachi Data Systems (HDS), NetApp and many others include this capability in their storage servers. Replication software that can keep copies of data items in several places is available. Replication occurs either in the storage server itself or in host systems attached to the storage server. In the case of a failure, operations staff can point applications to data items in another location. Some products will automatically redirect storage requests, rather than requiring manual intervention.

- Storage software that, like storage hardware, keeps multiple copies of each data item. This approach, offered by suppliers such as DataCore Software, Citrix Systems Inc./Sanbolic and others, also make it possible for the data to be replicated to other datacenters or cloud services.

- Software that supports continuous processing. This can include special-purpose virtual machine (VM) software such as everRUN from Stratus Technologies. This approach is very similar to a combination of continuous-processing hardware combined with monitoring, management and application migration. If a slow-down or failure is detected, applications and their components are moved to a surviving system. The detection and migration process is automatic and simulates how continuous-processing systems work. This often requires multiple systems to be used as hot or warm standbys.

While some products can detect failures and start arecovery process, many require manual intervention.

- Software that monitors VM operations and initiates a migration from one host to another system. This category includes VM monitoring, management and migration tools offered by Citrix, Microsoft, Red Hat, SUSE and VMware Inc. When the monitoring software detects a slowdown or failure, applications or even entire workloads can be migrated from host to host, datacenter to datacenter, or even from datacenter to the cloud.

"One-Size-Fits-All" Is a Myth

It's clear from reviewing these different approaches that each has its benefits and its limitations. Some approaches provide an environment that never fails, but at a very high cost. Others are less costly, but the failover process can take some time, possibly resulting in data loss.

Three key questions must be considered for *each application and application component for the proper mix of technology to be selected:*

1. How much money does the enterprise want to invest in availability for specific applications and their data?
2. How quickly must each application become fully available and functioning?
3. How much overhead can be supported to provide availability?

Depending on the answers to those questions, a mix of solutions can be selected. Here are a few suggestions:

- Hardware-based solutions, such as continuous-processing systems combined with either storage servers or storage software, can be deployed for the most-critical applications. The type of storage, which includes rotating media, flash-based devices or cloud services, can then be selected based on the cost, performance and availability parameters that are best for the application.

- Software-based solutions, such as the use of VM migration combined with storage-software solutions, often cost less and are more flexible. Remember, however, that they're often slower than hardware-based solutions. Less-critical applications can be hosted on these types of solutions.

- Cloud-based solutions, such as those offered by nearly all cloud services providers, often appear to be the least complex and lowest cost. The failover process, however, can be lengthy, and access to data items can be slow when compared to local storage.

Rotating media-based technology offers solutions to any but the most strenuous application and business requirements.

Another important point to consider is storage performance and its impact on each of these potential solutions. Here are some general guidelines:

- In-memory storage is used for some extreme transaction, Big Data and Internet of Things (IoT) applications. This approach offers the highest performance for the highest cost. They also typically require backup software to be used to prevent data loss in the case of system failure.

- Flash-based storage solutions offer high levels of bandwidth and low levels of latency. While very fast, these solutions are costly compared to traditional media. Suppliers of all-flash storage systems, such as EMC, Kaminario, NetApp, HDS and others would point out that the cost per gigabyte has been dropping dramatically, and is now more comparable to other types of storage.

- Rotating media-based technology offers solutions to any but the most strenuous application and business requirements. The intelligent caching capabilities of storage-software products, such as those offered by DataCore or Citrix/Sanbolic, often provide similar levels of performance using a small amount of system memory or flash storage. When combined with other types of storage, they can offer lower overall cost.

How Long Is a Piece of String?

In the end, there is no single answer that addresses the needs of all enterprise applications. Each form of DR and storage technology fits the requirements of some applications and not others. This is a bit like being asked the question, "How long is a piece of string?" The only correct answer is to measure the piece of string in question.

There is no single answer that addresses the needs of all enterprise applications.

Enterprises must take the time to understand their own application portfolio, along with their own business and availability requirements. This must be done for each application. Only then can the proper mix of approaches be selected to address both the need for availability and disaster tolerance, balanced against budgetary limitations. VR

Dan Kusnetzky writes the Dan's Take column for Virtualization Review *magazine. A reformed software engineer and product manager, he founded Kusnetzky Group LLC in 2006. Kusnetzky's literally written the book on virtualization, and often comments on cloud computing, mobility and systems software. He has been a business unit manager at a hardware company, and head of corporate marketing and strategy at a software company.*

Disaster Recovery Planning for Hyper-Converged Infrastructure

Is "shelter in place" really the right answer? By Jon Toigo

Most of the chatter these days about Big Data analytics envisions a sprawl of inexpensive server/storage appliances arranged in highly scalable clustered-node configurations. This hyper-converged infrastructure (HCI) is considered well-suited to the challenge of delivering a repository for a large and growing "ocean" (or "lake" or "pool") of data that is overseen by a distributed network of intelligent server controllers, all operated by a cognitive intelligence application or analytics engine.

It all sounds very sci-fi. But, breaking it down, what are we really dealing with?

HCI has never been well-defined. From a design perspective, it's pretty straightforward: a commodity server is connected to some storage that's usually mounted inside the server chassis (for example, internal storage) or externally connected via a bus extension interface (for example, direct-attached storage over Fibre Channel, SAS, eSATA or some other serial SCSI) all glued together with a software-defined storage (SDS) stack implemented to provide control over the connected storage devices.

Hyper-converged infrastructure has never been well defined.

What's Old Is New Again

The SDS stack provides all of the "value-add" functionality that was traditionally delivered via value-add software operated on the controller of an expensive shared or SAN-attached storage array—functions such as de-duplication and compression, thin provisioning, incremental snapshots, snap-clones and disk-to-disk mirroring. Touted as "new," SDS is actually a retro-architecture, resembling System Managed Storage (SMS) software that was a fixture on mainframes from the 1970s forward.

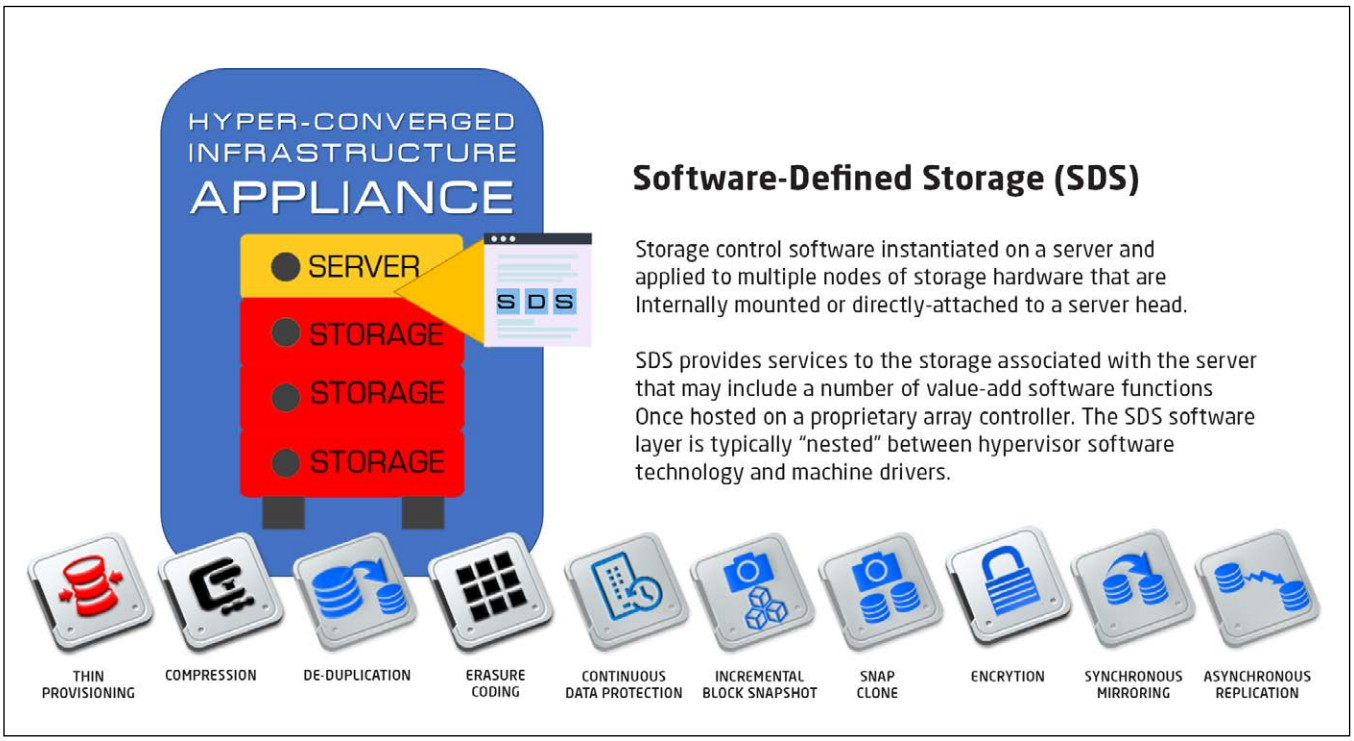


Figure 1. The kinds of functions usually provided on a hyper-converged infrastructure appliance as part of the software-defined storage stack.

SDS (re-)appeared a few years ago as the latest of many efforts by VMware Inc. to resolve performance problems with virtualized applications. Despite the fact that storage I/O couldn't be demonstrated in most cases to be the cause of slow virtual machines (VMs), and despite the preponderance of evidence that sequential I/O processing at the chip level was to blame for poor VM performance, VMware nonetheless cast blame on "proprietary, legacy storage" and proposed its wholesale replacement with SDS as the fix.

"Software-Defined" to the Rescue?

SDS was offered to solve a problem over which it had no influence or control.

So, SDS was offered to solve a problem over which it had no influence or control. While "legacy" storage vendors had, in fact, leveraged proprietary value-add software joined at the hip to proprietary array controllers as a means to differentiate their commodity hardware kits from their competitors' (and in many cases to justify obscene prices for gear), this fact had little or nothing to do with virtual application performance. However, VMware's (and later Microsoft's) embrace of SDS saw the model trending (see Figure 1).

SDS was subsequently leveraged by VMware (and Microsoft) to create a proprietary stack of software and hardware represented as an "open" hyper-converged architecture: hypervisor, SDS, software-defined network (coming soon), hypervisor-vendor-approved commodity hardware. The result was completely open—to anyone who built their infrastructure using only VMware or only Microsoft.

However, a number of third-party or independent software vendors also entered the market with their own take on SDS. Most improved in one way or another on the SDS stacks of the larger vendors; and, in order to claw market share, joined together with server vendors (most of whom were weary of being characterized as a "commodity kit" by VMware or Microsoft) to create HCI appliances.

Take a farm of these HCI appliances, each with its own storage nodes, cluster them together, overlay with a workload parsing (spreading parts of the overall analytic workload around to different nodes) and analytics engine (to collate and derive information from the ever-growing pool of data storage across the nodes), and, *voila*, Big Data happens.

Understood for what it is, this kind of infrastructure is reminiscent of superclusters (supercomputers built from distributed server nodes). Some of the inherent ideas for protecting the data in this infrastructure and for recovering the infrastructure from machine faults are also borrowed.

Multi-Nodal Storage

For one thing, each HCI appliance has two or three storage nodes at deployment. With three node kits, one node provides a quorum function—overseeing data mirroring between the other nodes and approving the synchronicity of mirrored volumes and data.

There is considerable debate over the value of data in a Big Data environment.

With two node kits, the quorum functionality doesn't require its own storage node. Vendors of the three-node-minimum storage configuration tend to issue separate software licenses for each node, which helps to account for why they prefer a standalone quorum node.

The data protection afforded by multi-nodal storage is simple: Data is replicated on two different sets of media at time of write—whether the copy is made to two targets at once or is made from target node A, then to target node B, with an acknowledgement made of the second write before the process concludes (so-called two-phase commit).

The end result of this configuration is that the same data has been written to two (or three) nodes, enhancing its survivability in the event of a hardware failure in any one node. Of course, this strategy is vulnerable to the failure of the node controller (the server), which will make the data on all nodes unavailable. It's also vulnerable to the propagation of erred data or corrupted data to all nodes.

The possibility of bad data being replicated is handled by some SDS vendors by providing a snapshot of block data in the primary target to a snap volume (another location on the node) prior to writing the new data to the primary target. In this way, an erred data write can be "backed out" of the primary volume if necessary.

High Availability

The possibility of nodal controller failure is handled by high-availability architecture, clustering the primary server with a mirrored server. If the two are operated in concert, with each supporting the identical workload, this is an active-active cluster that will survive a server

failure without stopping operations. If one server remains offline or near-line, serving only to mirror the data on the primary until it needs to be activated to support workload, then the configuration is active-passive. In such a case, the second server-node controller activates when the primary server "heartbeat" is lost (suggesting server failure). Some data loss may occur, but the shift of the workload to the secondary cluster member is expected to be swift.

These are the basics of disaster recovery (DR) and data protection that you're likely to encounter in any Big Data HCI farm. Not surprisingly, this is where most DR planning ends. "Not surprisingly," because of a couple of unfortunate ideas that have crept into the architecture.

Why protect a lot of data that has no value?

Big Data Concerns

First, there's considerable debate over the value of data in a Big Data environment. Some data scientists argue that the constant influx of data into the Big Data repository has a very limited shelf life. Imagine using Big Data to evaluate the validity of a credit-card purchase in Hawaii. The analytics engine might examine the last 10 purchases made with the card, evaluating where the credit card was presented at the time of each purchase. If minutes before the card is swiped in a reader in Honolulu, the same card is used in Kalamazoo, Mich., there might be a problem. If the analytics engine only examines the last 10 purchases, then what's the value of purchase 11 or 12?

If data only has a very limited useful period, a somewhat incomplete data-protection effort is understandable. Why protect a lot of data that has no value?

In many cases, firms believe that historical data may eventually have value, but they prefer not to incur the cost or "friction" created by moving a lot of data to an archive. Instead, many vendors prefer a strategy of "shelter in place."

Gimme Shelter

Shelter in place has different definitions, too. In some cases, especially among object-storage advocates, shelter in place might mean ending the mirroring of data and replacing it with a data-protection strategy based on erasure coding. Erasure coding involves the application of an algorithmic process to a piece of data that creates mathematically related objects that can be distributed across nodal volumes. If the original data is corrupted, it can be

recovered using a subset of the mathematically related objects. This technique is useful for very infrequently changing data and uses less storage than redundant mirroring of all files.

Another meaning of shelter in place is to spin down or de-energize drives in the pool that contains infrequently accessed data. The theory is that quiesced data on unpowered drives can be made available again "at the flick of a switch," if needed, by the analytics engine. While the industry does support different power modes on some hard disk drives, issues remain regarding the wisdom, and efficacy, of turning off drives that contain "archival" data.

The possibility of nodal controller failure is handled by high availability architecture, clustering the primary server with a mirrored server.

Both of these shelter-in-place strategies also run afoul of a bigger issue with HCI Big Data farms generally: the risk of a facility-level or milieu-level outage. If a facility burns down, or a pipe leak develops that requires a power down of hardware and evacuation of the facility, all of the shelter in place, CDP, incremental snapshotting, and intra- and inter-nodal mirroring in the world will not enable recovery.

Playing the Odds

Many hypervisor and Big Data vendors are quick to point to outage statistics suggesting that up to 95 percent of annual downtime results from logical and localized interruption events: application errors, human errors, component failures, malware and viruses. The largest portion of this 95 percent pie slice is scheduled downtime. The other 5 percent of annual downtime results from capital "D" disaster events such as building fires, weather events, geological events, nuclear or chemical disasters, and so on. They openly state that a sensible DR strategy is one that "plays the odds," that is, that high availability trumps DR.

This is the kind of thinking that can get an organization into trouble. The fact that only 5 percent of annual downtime results from disasters at the facility or milieu level is not to suggest that effective DR planning can safely ignore these potentials. At a minimum, if the Big Data operation is deemed critical to business operations, a copy of data supporting this critical set of business processes should be stored off-premises and at a distance sufficient to prevent it from being consumed by the same disaster that destroys the original data.

Given the huge amount of data that's being amassed in Big Data farms, replicating all data across a wire to another location or a

DR-as-a-Service provider (or cloud-based Big Data infrastructure) might seem impossible. Moving just 10TB across an OC-192 WAN or an MPLS MAN will take a couple of hours (much faster than the 400-plus days it would take to move the same quantity of data across a T-1 connection). The alternative is cloud seeding.

Cloud Seeding

Cloud seeding involves making a copy of data farm bits to a virtual tape library (another storage node behind an SDS controller such as StarWind Software). Then, in an operation that doesn't take processor capability from the working servers, data is copied over to a tape system operating under the Linear Tape File System (LTFS, created by IBM and now an ANSI standard). Given the huge and growing capacities of tape, the portability of the medium, its resiliency, and combining that with the no-hassle straight copy of files or objects to tape media with LTFS, the result is a means to move a large quantity of data at the speed of any popular transportation method.

Given the huge amount of data being amassed in Big Data farms, replicating all data across a wire to another location or a DRaaS provider might seem impossible.

With a copy of your data loaded to tape, the media can then be sent to an off-site or cloud services provider that can store—or load—your data from tape into compatible infrastructure so it's ready for use if and when a capital D disaster occurs. The cost is minimal and the recovery capability afforded is awesome.

In the final analysis, hyper-converged infrastructure is a work in progress. The foundation of the technology, SDS, is still in flux as vendors struggle to determine what functions should be included in the SDS stack; how best to support hardware flexibility; and how to deliver workload agnosticism. Moreover, much work needs to be done on the data-protection story of HCI. Shelter in place isn't really a full-blown data-protection strategy; it's more akin to the laissez faire strategies for DR from three decades ago, which amounted to: "Take a backup and cross your fingers." Shelter in place and hope for the best won't cut it in an always-on world. **VR**

Jon Toigo is a 30-year veteran of IT, and the managing partner of Toigo Partners International, an IT industry watchdog and consumer advocacy. He is also the chairman of the Data Management Institute, which focuses on the development of data management as a professional discipline. Toigo has written 15 books on business and IT and published more than 3,000 articles in the technology trade press.

Business Impact Analysis: The Key to Successful Continuity Planning

If your potential DRaaS provider doesn't do it, find another one. By Jon Toigo

Business impact analysis (BIA) can be conceived of as the "heavy lifting" of continuity planning.

Recently, an investment fund manager contacted my firm with questions about investment possibilities in the Disaster Recovery-as-a-Service (DRaaS) provider market. What, he asked, were the key attributes that he should consider before committing his client money in such ventures?

Was the size and location of the DRaaS cloud-hosting environment the big issue? Was the accessibility of the DRaaS services via WANs and MANs the key? Was it the target market: horizontal services aimed at all small and midsize firms, for example, versus a vertical focus on services for health care firms, or media and entertainment firms, or oil and gas?

The best answer I could come up with to the question was a simple one: Determine whether the provider offers business impact analysis, whether on a free or fee-paid basis, prior to selling backup, mirroring/

replication or re-hosting services. The rationale for this response: Without a solid investigatory and analytical exercise—often referred to as business impact analysis (BIA)—performed as a precursor to creating a continuity plan, firms are doing business continuity planning all wrong. A competent DRaaS provider would know this.

The phone went silent for a few seconds. Then the manager cleared his throat. He said he had spoken with a number of DR experts and none had suggested this notion. He had been told instead that, because everyone had adopted server virtualization, there was no longer any need for all of that traditional DR process, including up-front data collection and impact analysis. In an emergency, a company would simply slide its workload over to an adjacent server in a local high-availability cluster, or worst case, to a server host in a cloud somewhere, and continue operating until the non-cooperative node could be replaced.

Chances are pretty good that firms haven't actually "mapped" business process to application and application to data and infrastructure, because each of their applications were first purchased, or developed and deployed.

In short, the strategy recommended by the preponderance of DRaaS providers is to forego analytical steps to determine which business processes are "mission-critical" (and, therefore, must be continuous in their operation), and to jump straight to the strategy-selection process (deciding which services provider to use). On closer examination, this makes little sense, especially from the standpoint of strategy cost.

Heavy Lifting

BIA can be conceived of as the "heavy lifting" of continuity planning. Chances are pretty good that firms haven't actually "mapped" business process to application, and application to data and infrastructure, because each of their applications were first purchased or developed and deployed. At first glance, BIA is simply an effort to re-discover these relationships so you can get a handle on where the data, application and infrastructure assets you need to protect (or replace, in the event of a big disaster) are physically located.

The procedure usually begins by interviewing senior management to learn what they think are the most critical business processes: those whose interruption for any length of time would stop the business from operating (see **Figure 1**). Of the potentially hundreds or thousands of business processes in a large firm, this insight will usually winnow the field of "mission-critical" down to a handful. That's a starting point.

Q&A

Next, the investigator needs to talk to the line-of-business manager in whose realm the critical business process operates. Information needs to be collected to create a flow diagram depicting how the business process works. Ideally, this diagram will include the application software accessed to perform automated functions, or to obtain data necessary for business-decision making.

The strategy recommended by the preponderance of DRaaS providers is to forego analytical steps to determine which business processes are "mission-critical," and to jump straight to the strategy selection process.

Next, it's off to the IT department to learn more about the cited applications. What databases do they use? What are the interdependencies of each application on the operation of, or data from, other applications? Where and how are the applications hosted? Where is the data that's produced by the application stored? How much data is there, how often is it re-referenced, does it expire?

You're getting the idea: The purpose of the BIA exercise is to use the thing you're trying to protect or continue—the business process—as a starting point to identify applications, data and infrastructure that support it. The reason for this is simple; applications, data and infrastructure have no value except in relation to the business process

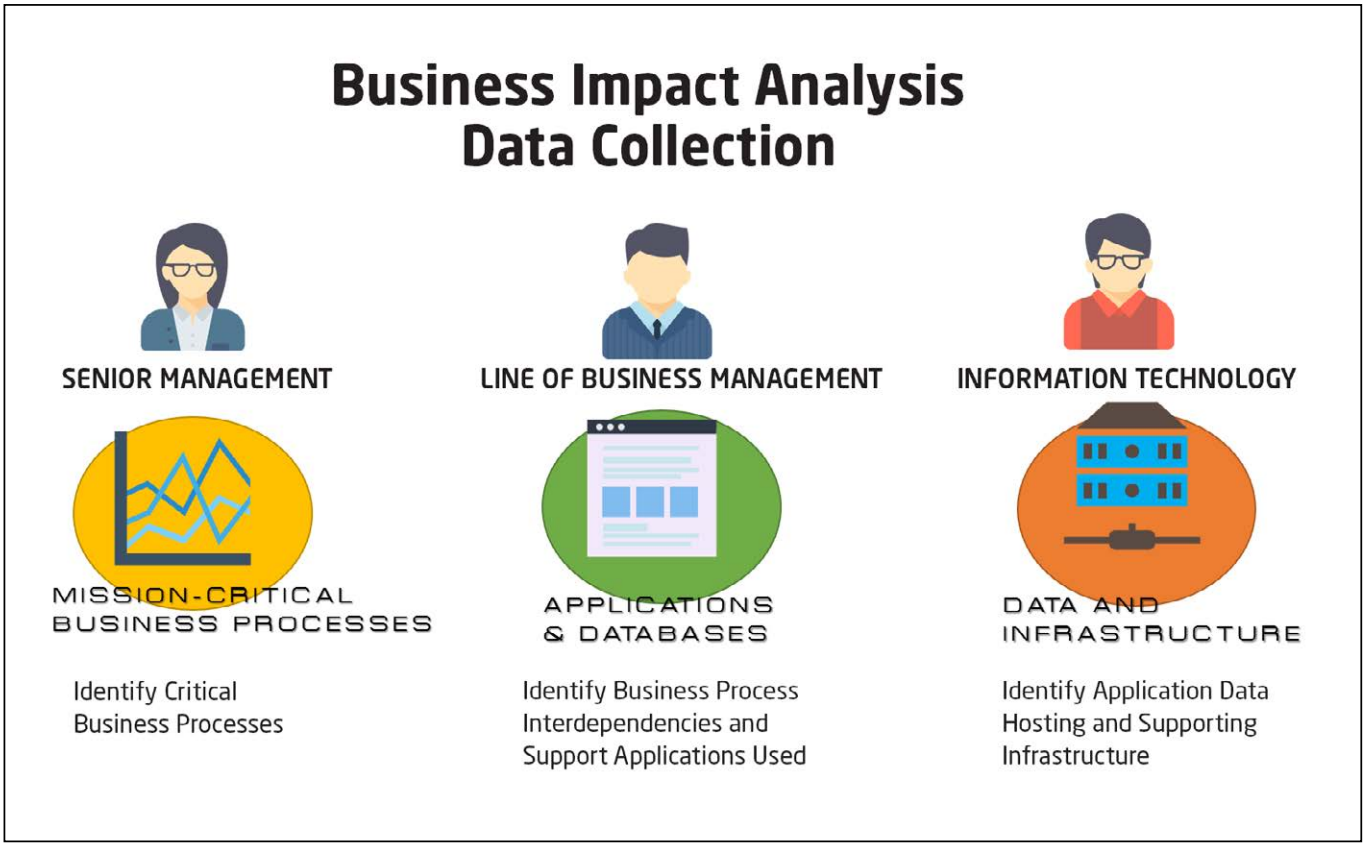


Figure 1. A thorough analysis, including key decision makers, is the first step in business impact analysis.

they serve. All SQL databases are the same; all data is anonymous ones and zeros; all hardware is commodity. The business process they serve determines whether they're mission-critical, important or merely archival.

The alternative to doing this sort of mapping exercise is to simply treat everything the same, whether the application will determine the solvency of the business or have no impact whatsoever on business continuity. The alternative to a business-process-centric analysis is to ask a business manager which of his applications is important for inclusion in a DR plan. You'll always receive the predictable answer: They all are.

One reason for doing a Business Impact Analysis is to show management that you care about how the company's money is being spent.

Why do we care about this difference? Simply put, because of cost. Mission-critical business processes impart this value, like so much DNA, to the applications, data and infrastructure that support them. These need the primo service level, the best protection and best continuity strategy available. They're prioritized in what is likely a constrained IT budget over the recovery requirements and strategies of less-critical apps, data and hardware. So one reason for doing a BIA is to show management that you care about how the company's money is being spent.

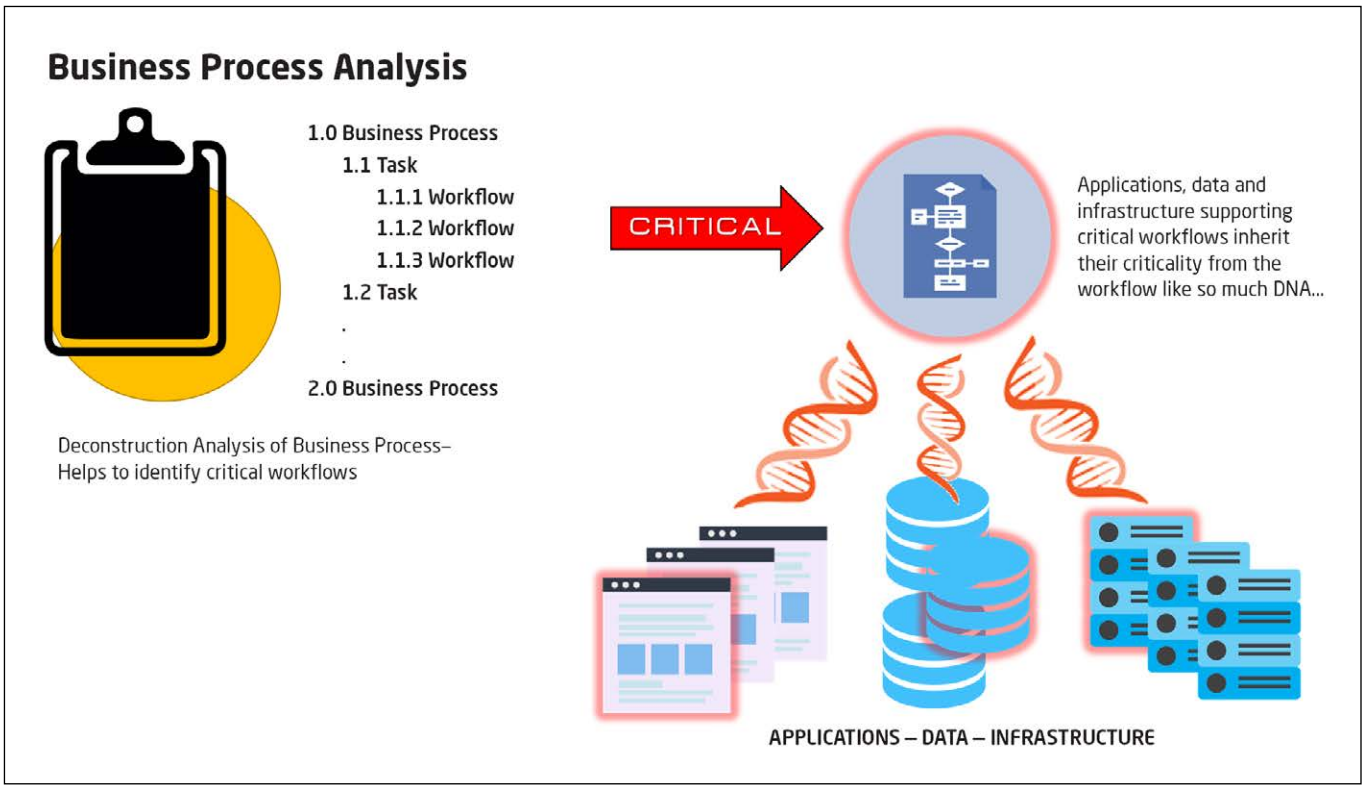


Figure 2. The analysis should identify critical workflows to the business.

Second, the only way to create a workable DR plan is to understand what you're trying to protect and recover. This requires detailed research into the application's hosting environment, network traffic, data-storage methods and so on. Assuming you want to make a mission-critical app highly available or always-on, continuity planning begins with a specification for hosting that application and its data in a highly resilient way to begin with. **Figure 2** shows how this process should work.

This might include active-active clustering with ongoing data mirroring across redundant networks, for example, to help contain nodal-level faults or logical errors that might impact one side of the cluster. Such hosting strategies are expensive, however, and shouldn't be applied to every application or virtual machine, but only to those that merit such strategies.

Without a solid investigatory and analytical exercise, often called business impact analysis, performed as a precursor to creating a continuity plan, firms are doing business continuity planning all wrong.

So, a BIA doesn't just serve as a basis for assigning appropriate data protection services, it also helps to determine the efficiency and appropriateness of the production hosting environment. In fact, BIA data can be used to review IT infrastructure efficiency, to better map data to resources to achieve regulatory compliance and to design an intelligent data security strategy. It serves many masters and pays for itself in terms of improved efficiency and economics.

The BIA is critical for another reason, as well. As interviews with business and technical personnel are conducted to map business process to app to data and infrastructure, the investigator can also gather other information to determine actual criticality (see **Figure 2**). Remember, you're proceeding based on input from senior management as to what the most-critical processes are. The folks in the trenches often have insights to offer that are not obvious from senior management input.

For example, management might see a particular business process as critical, but not understand that delivering the output of the process is contingent on several subordinate processes that, taken separately, might appear to be less-important activities. Only the folks in the trenches understand interdependencies that must also be recovered in short order for the critical process to be restored: They're also critical by association.

The folks in the lines of business are also in a better position to explain the costs—whether tangible (lost revenues per hour) or intangible (lost reputation by the minute)—of an outage in their business process for an hour, several hours, a day or several days. Their numbers might be bogus ("we make a billion dollars a year, so every day of downtime equals $1 billion divided by 365") but they are ‹‹ital››their ‹‹text››numbers.

When the time comes (usually right after the BIA) to justify the cost of a DR plan to management, investigators can tally up the outage costs per hour, per day, per week and so on, calculated using values supplied by lines of business managers (and not by the DR planning team) to show what the company stands to lose in the event of an unplanned interruption. The cost for building an effective recovery capability is typically a small fraction of this potential loss exposure. "Don't believe me," the BIA investigator can say to management, "believe your own management team!"

Different applications and data have very different protection service requirements.

Aside from these values, the BIA also provides the rationale for selecting different techniques and methods for protecting data, for preventing interruption events that can be prevented and for recovering from events that can't be prevented.

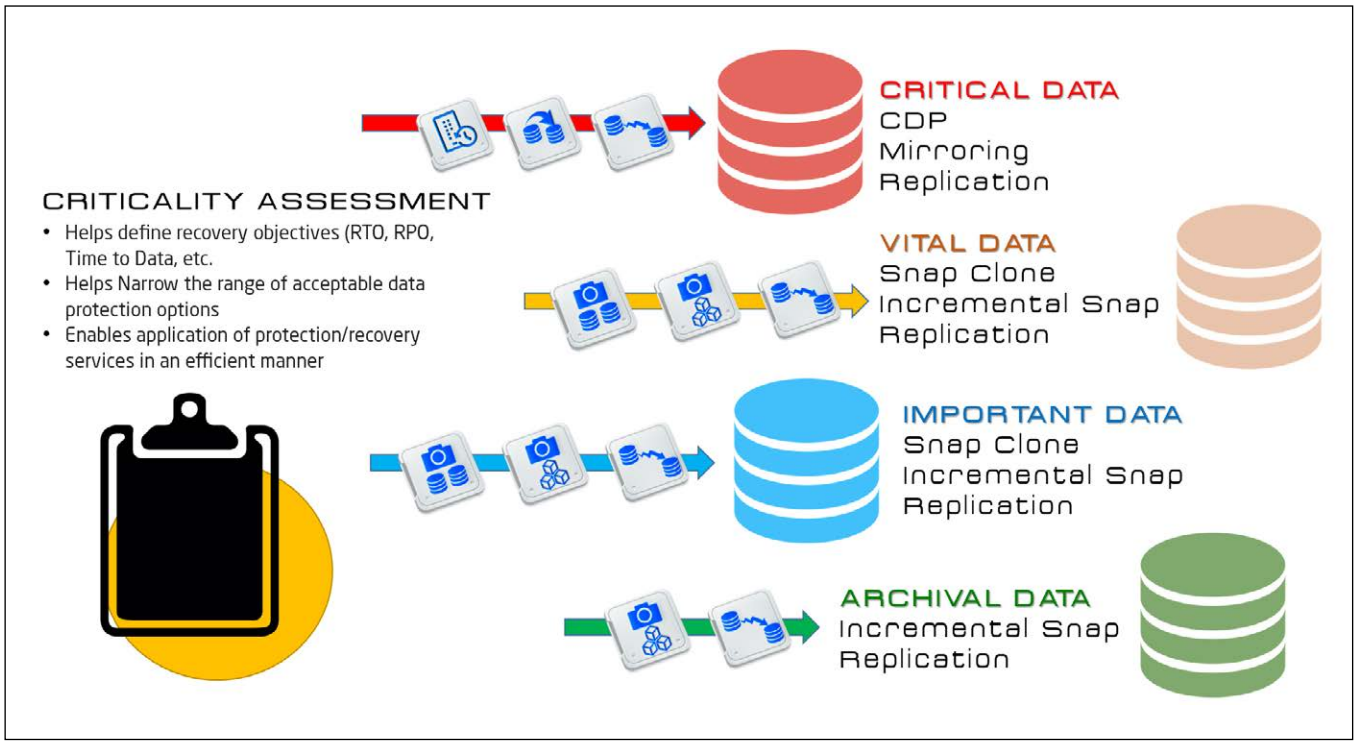


Figure 3. The criticality assessment helps prioritize data for disaster recovery scenarios.

In the area of data protection, for example, critical data may require continuous data protection (CDP) services that copy each I/O and its time stamp to a continuity medium so that, if a ransomware attack occurs, you can rewind I/O to a point in time immediately before the malware was inserted or activated. That kind of protection (shown in **Figure 3**) might be supplemented by semi-frequent snapshots of change data made to a separate snapshot volume, or that are applied to a clone volume on a routine basis. These are additional protections against logical- and component-level faults.

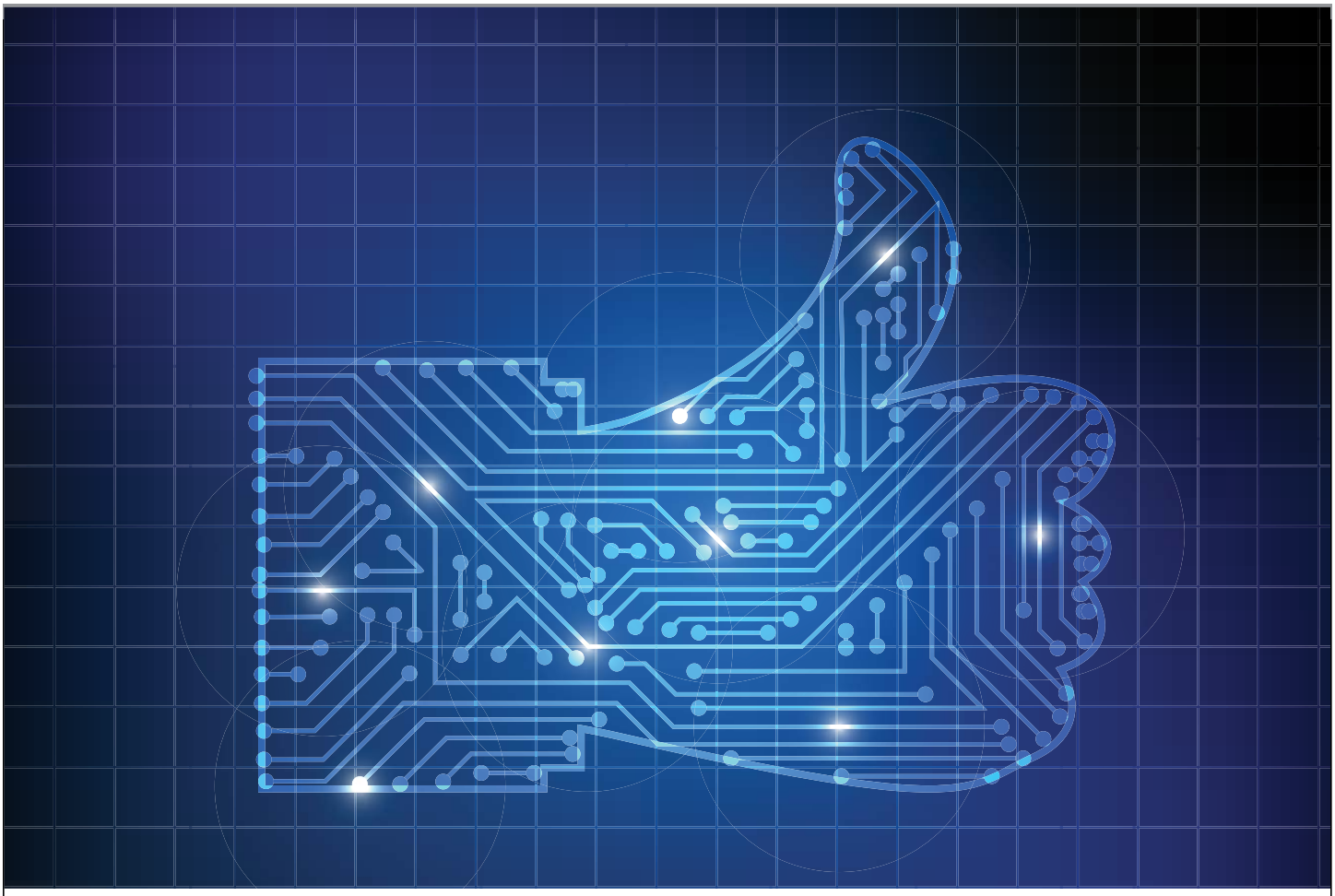
But because snaps and CDP are usually stored on the same nodal hardware as original data, you might wish to enhance the protection of critical data with mirroring between storage nodes as protection against a nodal failure, and also to replicate mirrored data off-premises to a DRaaS cloud or hot site or other business facility. Mirroring and asynchronous replication over distance are two additional services that might be layered on to the protection of mission-critical data. Certainly, not all these services are required for archival quality data whose change/re-reference rates have slowed to virtually nil.

Mirroring and asynchronous replication over distance are two additional services that may be layered on to the protection of mission critical data.

The point is that different applications and data have very different protection-services requirements. Only a BIA will provide the information needed to match protection and recovery services cost-effectively to assets that require protection.

A DRaaS provider who offers assistance to perform a BIA is definitely distinguishing itself from competitors by aiding its customers to use the service selectively and intelligently. Even companies that aren't using cloud-based DR services need to resist the urge to forego the impact analysis and to go straight to the acquisition and deployment of backup and recovery technologies. Simply put, the BIA is what imbues the continuity strategy with its efficacy and with its business-savvy. **VR**

Jon Toigo is a 30-year veteran of IT, and the managing partner of Toigo Partners International, an IT industry watchdog and consumer advocacy. He is also the chairman of the Data Management Institute, which focuses on the development of data management as a professional discipline. Toigo has written 15 books on business and IT and published more than 3,000 articles in the technology trade press.



The Quest for Guaranteed Recovery Assurance

You probably have a disaster recovery plan in place. But do you know that it's ready if your worst-case scenario occurs?

By Jim Whalen and Christine Taylor

Most businesses have a disaster recovery (DR) plan. But unbeknownst to those companies, many of those plans will never work. That's not a problem if the business never experiences data corruption, or if an employee never walks off with a server, or if no tornado, hurricane, or earthquake ever strikes. The question is: Do you want to take the chance of these things happening and turning into a nightmare scenario? Because having a DR plan isn't enough: You need to know that it's going to work when

you need it. The only way to do that is by periodically testing your DR plan with recovery assurance (RA) testing.

Business continuity (BC) allows the business to continue to generate revenue no matter what happens, and is the driver for data protection and DR. DR plans should counter a multitude of threats to applications and systems, ranging from relatively minor data loss or equipment failure, to a major natural disaster such as flooding or a hurricane.

Having an effective DR plan for these circumstances doesn't mean just a nicely written plan on which someone in IT spent a lot of time. It does mean that the DR plan is in place, that it covers applications by service levels and that it's ‹‹ital››guaranteed ‹‹text››to work. This is a matter of basic business survival; losing your compute capabilities for any significant length of time constitutes a disaster—and the definition of "significant length" gets shorter all the time.

Losing your compute capabilities for any significant length of time constitutes a disaster.

You've Got Choices

Virtualization allowed IT to consolidate workloads by running applications in multiple virtual machines (VMs) on top of shared server hardware, making better use of available resources. These same benefits apply to DR.

In the "old days," IT would have to set up physical servers running dedicated applications at a DR site, mirroring what was running on the production floor. Now, they can keep applications running as VMs on generic hardware, with much more flexibility and at a lower cost, making comprehensive DR more feasible than back in the pre-virtualization, pre-cloud days.

However, while DR is now more feasible and more important than ever, it's also more complicated. Companies now demand tighter recovery point objectives (RPOs) and recovery time objectives (RTOs), applications running over multiple VMs, load balancing, boot order and dependencies. Other big complicating factors include the sheer size of data and backing up from multiple sites to multiple locations, including remote sites and the cloud.

Ultimately, RPO and RTO must rule the disaster recovery roost: RPO for the maximum amount of data that can be lost without significant business loss, and RTO for the maximum amount of time that an

application can be down without significant business loss. Let's look at how well different solutions ensure acceptable RPO and RTO:

Ultimately, RPO and RTO must rule the DR roost.

- **Do nothing:** This "solution" is more common than you might think. It's generally a combination of the unwillingness or inability to do serious DR testing, the fervent hope that nothing really bad will happen, and the over-optimistic belief that even if there is a disaster, the environment can recover before any serious damage is done.

- **Restore from off-site backups.** This is the most elementary of DR plans. It's workable in smaller environments with generous RPO and RTO, and a tested restore plan such as a contract with an off-site company to deliver data on removable media within 12 hours of an outage. However, the process is highly manual and error-prone, takes hours to restore from tape or optical drives, and may require rebuilding servers and storage from bare metal.

- **Self-managed DR:** These companies manage their own DR programs and are usually heavily integrated with the cloud. IT chooses hot, warm, and cold options depending on application priority and RPO and RTO. Hot options include immediate automated failover to the secondary site upon a threshold event. Warm options enable a failover site that IT manually launches as needed. Cold options present an environment that IT can prepare and launch when needed. The same IT group might invest in all three services, according to budgets and differing application priority. Whatever combination IT chooses, it's absolutely critical that they periodically test all three options.

- **Cloud-based Disaster Recovery as a Service (DRaaS):** Instead of internally managing DR, IT works with a cloud-based DR services provider to develop a custom plan. The provider is responsible for building and maintaining infrastructure and verification. It's not a hands-off process; IT works closely with the provider to communicate service levels and DR priorities, and to expand as needed. The DRaaS provider will do the heavy lifting of deploying and managing the recovery infrastructure and verifying recoverability.

What Is Recovery Assurance?

Let's talk more about verifying recoverability, or recovery assurance (RA); it's also called guaranteed DR, reliable DR, DR assurance or DR testing. At the simplest level, RA simply means doing enough testing on your backups and replications so you know you can recover systems in the event of a failure.

However, as the old saying goes, the devil is in the details. RA can be complicated because IT needs to pay ongoing, consistent attention to keeping applications continuously available. This is not a trivial undertaking.

Business continuity (BC) allows the business to continue to generate revenue no matter what happens, and is the driver for data protection and disaster recovery.

First of all, the DR environment is subject to entropy. IT needs to regularly carry out DR testing to keep production and recovery environments in sync. Once a year, or even once a quarter, is probably not enough. Data grows, OSes are updated, patches pile up, applications are upgraded to new versions and so on.

RA also needs to be non-disruptive. As critical as it is, testing cannot compromise production. In order to test DR, some IT organizations try to take evenings and weekends to verify recovery operations. Finally, RA must be robust enough to detect and flag issues so IT can correct them. Typical examples include application-inconsistent backups across multiple interdependent VMs, failed backups and corrupted backups (see **Figure 1**).

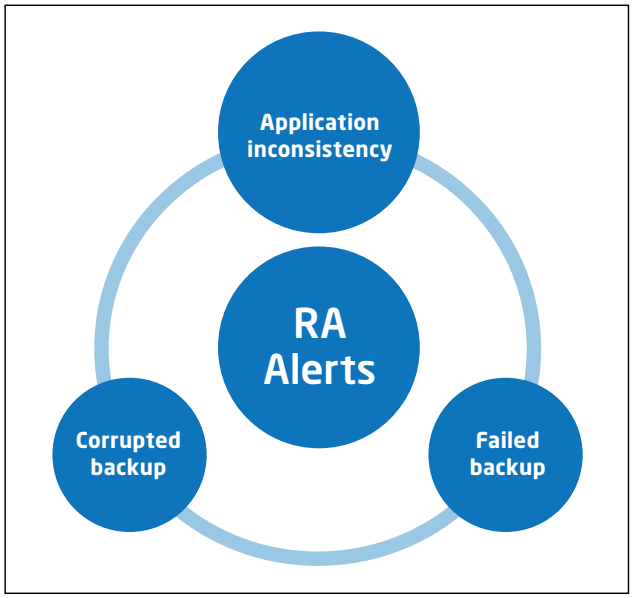


Figure 1. Recovery assurance should detect multiple points of failure.

What Do You Need in an RA Solution?

- **Automated testing.** Manual testing has its place, but only automated testing can sufficiently test DR across a variety of service levels and applications. For example, critical applications might require a near-continuous process of backup and time-to-failover testing. Less-critical applications won't have to be tested as often, because IT will have the time to rebuild application servers in-house and restore data.

- **The ability to verify application-consistent backups.** You'll likely be dealing with multiple VM applications and boot order orchestration, and possibly OS and software upgrades. Your RA solution needs to ensure that all of those things are checked out and in order.

- **The ability to account for failed backups.** RA needs to note that a backup has failed and flag it, so IT can fix any problems and reissue a backup command. It also needs to verify completion and check for corruption on finished backups, to avoid being dependent on an unrecoverable backup set.

A sufficient RA process is long and complex.

- **The ability to do sandbox testing.** Verifying recovery in a sandbox environment lets you test and tweak your DR plan without disrupting production.

How To Deliver RA

You can do RA in-house or as a purchased service, or a combination of the two. There are two do-it-yourself approaches to RA: manually and using a third-party tool. The first method is usually inadequate, because a sufficient RA process is long and complex. If you're going to do RA in-house, it's far better to go with third-party tools and use one of the automated toolsets available.

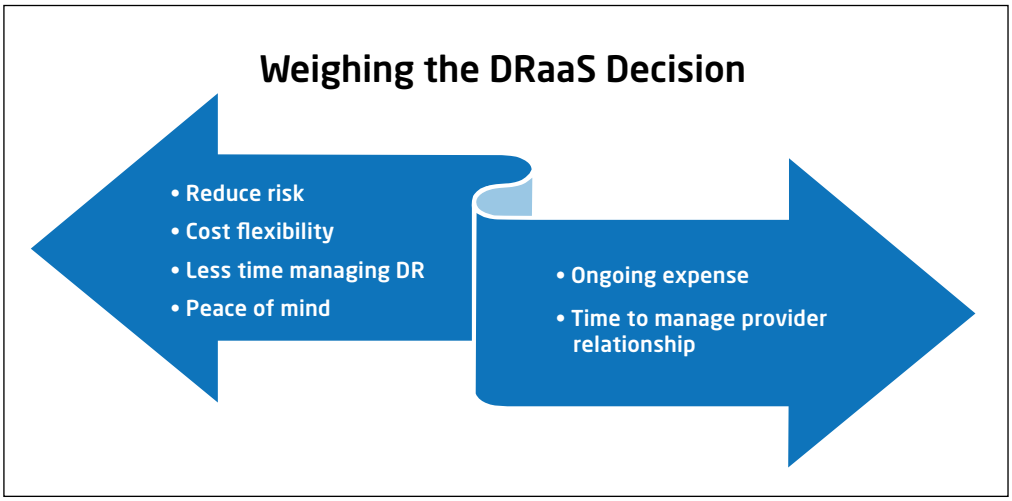


Figure 2. Comparing the advantages and disadvantages of Disaster Recovery as a Service.

These toolsets will enable you to non-disruptively test multiple applications for recovery. Although highly automated toolsets will probably cost more, they'll come with two big benefits: they'll work more thoroughly, and they'll let you test more frequently. When it comes to RA, don't practice a false economy.

The second major RA option is to turn it over to a cloud-based DR-as-a-Service (DRaaS) provider. DRaaS in the cloud is worth looking into, with more and more established vendors entering the market every day. Benefits include offloading RA expertise from your staff to a provider who already specializes in it, and reducing risk by entrusting your RA process-to-recovery experts. If cost is a consideration (and when is it not?), you can selectively move critical applications to the service and take care of the rest in-house (see **Figure 2**).

Recovery Assurance also needs to be non-disruptive. As critical as it is, testing cannot compromise production.

The Long and Painful Road

There's good news and bad news about RA. The bad news is that there is no safe escape from the complexity of providing effective DR. The good news is that DR and RA technology and services are getting better all the time, and are much more accessible across the board to enterprise, small to midsize enterprise and small to midsize businesses. Strongly consider taking advantage of these tools and services. The time and money you might save by not doing DR well is negligible compared to the cost of a long and painful recovery process. VR

Jim Whalen and Christine Taylor are analysts specializing in data protection at the Taneja Group (tanejagroup.com).