Essential Guide for Application-Centric Workload Migrations to Hybrid Cloud

Nick Howell, DatacenterDude Services
James Green, ActualTech Media

CONTENTS

INTRODUCTION

Ask any admin who has managed a data center or IT department over the last 30 years what their biggest obstacle has been, and 9 out of 10 will tell you that it has something to do with moving data and applications around seamlessly. Looking at some of the most successful entry points IT vendors use to sell new products to businesses shows that some of the most-requested features involve simplifying workload mobility and the agility to migrate quickly and without downtime.

Remember scheduled downtime? Gone are the days of being able to schedule 8- to 12-hour outage and maintenance windows. IT departments today have a pretty standard requirement of never going down, never deleting data, and seamless failover in the event of a problem. Their primary goal is to keep users online and conducting business. The ideal scenario involves the end user never experiencing delays or outages in any way, shape, or form.

To provide this level of availability, IT professionals may utilize infrastructure that spans multiple racks and systems, or even more broadly, multiple geographically separated locations. They may also use infrastructure from multiple cloud providers.

Data and Application Availability

Redundancy comes in multiple flavors, the first of which has to do with availability. In order to prevent interruption to the user's ability to access key systems and workloads during a failure scenario, a redundant infrastructure can be in place. This infrastructure can reside either onsite or in a different location, and it exists to allow users to continue to work despite an outage. This requires data and applications to be kept synchronized on as tight a schedule as possible, usually limited only by the network capacity to do so. Network capacity demands increase as the amount of data being synchronized increases.

LOAD BALANCING A WORKLOAD

Many business-critical applications require multiple instantiations of the same workload kept in lockstep in an active-active configuration to balance load at peak times. This, again, is usually limited by network capacity. The key difference between load balancing for performance and synchronizing for availability is that the network performance required to run an active-active configuration is significant, and it's not feasible to loosen the constraints. Network requirements for availability-focused synchronization can be relaxed by adjusting the service level agreements; a higher RPO (recovery point objective) will require less network capacity.

In both of these traditional scenarios of workload migrations—redundancy and load balancing—massive amounts of data are constantly flying across the wire, and huge up-front CAPEX outlay is necessary to provide the underlying infrastructure to support either scenario (moving workloads for availability or for performance).

As more and more cloud services began to come online, businesses started to see opportunities to "rent" this infrastructure in an on-demand fashion, removing the need to spend millions on infrastructure they might only need in a contingency, and that they may realistically never even use.

Availability Service Level Agreements

IT organizations often make commitments to the rest of the business to deliver certain levels of availability for applications and data. When it comes to recovering from a failure, two key metrics often define the commitment (or service level agreement) to the business:

- **Recovery Point Objective.** This metric is a measure of minutes, hours, or days backward in time from the failure. It tells the business how much data will be lost in the event of a failure. Will we be able to recover everything that happened up until 15 minutes prior to the failure? Four hours prior? Will we be able to recover to the data we had at close of business yesterday but lose everything that happened so far today? The gold standard for many organizations is 15 minutes, but it's not cheap to keep this promise for your users.
- **Recovery Time Objective.** This metric is a measure of minutes, hours, or days forward in time from the failure. It tells the business how long until the recovery will be completed in the event of failure. From the time that incident response begins, will the users be back online in 15 minutes? Will they need to leave for lunch and come back when the problem is resolved? Do we need to send everyone home for the day? Again, the gold standard is 15 minutes, but this requires serious organizational organization and advanced technology.

As architectures matured, so did the migration methods of these workloads. Let's take a moment to go over some of the more popular use cases, the problems they solve, and some of the new challenges they present.

On-Premises to Public Cloud Migrations

There are many reasons a company may wish to have a presence in one or more clouds outside of their on-premises data center. One of the most popular reasons is the long-term storage of "cold data," either in the form of offline backups, or simply data that has not been accessed in a specified period of time.

Popular object storage systems and applications have built-in algorithms to determine data age, and they have capabilities to manage the migration of cold data to a particular cloud service.

What is Object Storage?

Object storage is unfamiliar to many enterprise IT admins who are accustomed to dealing with file- and block-based storage. Object-based storage stores data as unique objects (with a globally unique identifier) and generally attaches some metadata to each object. Dealing with large amounts of unstructured data is a common reason for implementing object-based storage systems.

In addition, more and more companies are leveraging cloud storage solutions for offline and offsite long-term backup. As the cost-per-gigabyte of storage has plummeted, the sheer volume of storage available at an extremely discounted rate has driven IT leaders to consider cloud storage as a viable alternative; companies like Iron Mountain have created entirely new lines of business around cloud offerings to replace more traditional tape vault archives. This has also forced backup orchestration software vendors to build in functionality that includes backing up data to cloud-based destinations. Amazon S3 and Glacier, as well as Microsoft Azure Blob storage (Hot and Archive), are examples of some of these offerings.

At the end of the day, however, someone or something still needs to keep track of the location of these backups and cold data, to facilitate future access should the need arise from a failure, audit, or user error.

MOVING ENTIRE APPLICATIONS TO THE CLOUD

While cold data storage may be a no-brainer, many companies are interested in tackling even bigger challenges. Often, IT leaders turn an eye toward the cloud when they're looking to solve elasticity and scalability issues. One of the key differentiators of the cloud as compared to traditional infrastructure is that it's inherently flexible. When it comes to applications such as eCommerce and big data analytics, the ability to scale up and down as demand changes can be a powerful capability, and architects stand to improve performance and reduce costs.

Although the gains to be made are very real, moving a legacy application to the cloud isn't generally as easy as just picking it up and moving it. Due to the completely new architecture that cloud services are predicated on, a rewrite of the application is often needed, and at a bare minimum, configuration changes will be required to ensure the application transitions smoothly. Of course, all this is very difficult if not impossible without a proper understanding of how the application works and its dependencies.

Data Center to Data Center Migrations

Moving data between data centers is a long-standing tradition in IT, whether it's replication for passive backups, secondary workloads, or geo-located users. However, as data became entire workloads inclusive of VMs and multiple volumes, these migrations became more and more complex. Moving entire workloads between data centers today requires not only the movement of data, but the servers and applications that consume and leverage that data. This requires an unprecedented amount of monitoring, if only to keep track of where everything is at any given time.

This has a compounding effect over time from a cost perspective, and is one of the principal reasons companies began looking at third-party infrastructure (clouds) years ago, as it was much more cost-effective to use a service provider than to invest the capital expense required to stand up a duplicate data center.

Offsite Migration for Disaster Recovery

Prepping for disaster recovery, or what is commonly referred to as a smoking-hole scenario, has always been -- and likely always will be -- one of the most common and pervasive use cases for moving workloads offsite. As data became workloads and failover orchestration became more automated, this encouraged companies to leverage offsite tools more and more.

To be clear, this is different than storing cold data and backups, as discussed previously. Rather, this is the ability to completely stand up your entire workload in an offsite scenario in the event of a catastrophic natural disaster or other situation where the onsite infrastructure is no longer available. This involves an incredible amount of planning, and often involves complex, detailed runbooks with step-by-step procedures to follow in the event of a disaster.

Better to Be Prepared!

For many fortunate companies, a catastrophic disaster is something they'll never have to face. However, more common types of failures like a hardware malfuction or a power outage can still bring down entire data centers. Even if your data center is relatively safe and secure, you still need to be doing proper disaster recovery planning!

Goals of a successful migration include:

- Cost-effective and fast to deploy
- Application agnostic
- Rollback capability
- Zero downtime

When performing a migration, it's also good to take stock of your technical debt, eliminating leftover and unnecessary resources, VMs, and instances. Think of it as a good housecleaning practice to free up infrastructure or save money on next month's cloud bill!

Workload Migration Challenges

Migrating workloads, especially complex, multi-tiered applications, can be a harrowing process. Undertaking the migration without a solid understanding of the interdependencies of applications and the exact resource requirements means almost certain failure. In today's world, it's understandable that failures can, do, and will happen; what's not acceptable today is downtime. Cutovers of migrations need to be instant, or an hour at most.

A failure of this migration endeavor can often be attributed to a bottom-up infrastructure focus: simply trying to mirror hardware available onsite. The downfall of most workload migrations is lack of visibility into the configurations of the actual application(s), and how it consumes resources from that infrastructure. With a proper understanding of how applications interact, the secondary site's requirements become much clearer.

MAPPING APPLICATION DEPENDENCIES

Understanding and documenting the relationship between all components of an application in extensive detail is vital to the success of a migration, as well as to managing and maintaining said application. Most organizations lack this level of insight, as documentation is almost instantly out of date the minute it is written, and deep understanding of the application often exists only as tribal knowledge in the heads of the administrators maintaining the application.

This raises a crucial question: what happens to a critical application if there's a problem and the administrators with that tribal knowledge are unavailable? Perhaps those individuals are busy, on vacation, hospitalized, or retired; there's an infinite number of reasons that tribal knowledge goes offline. Manually creating the workflows and processes

necessary to avoid downtime or repair failures is still not enough in today's IT world. These types of recoveries can be managed and monitored via automation, and workflows can be triggered automatically in the event of a failure, eliminating this vulnerability.

No one person should hold the keys to your digital kingdom, but let's also not assume that there is even a single person that has the detailed knowledge necessary to migrate or restore application services. In a modern application-centric IT organization, you can use a tool to provide this insight into the interdependencies of applications.

UNDERSTANDING RESOURCE REQUIREMENTS

When breaking down an application, you need to consider all of the services that application provides and connects with to properly recreate the resources required to run it. Oftentimes, an application can be run in a degraded state temporarily while a migration is in process, or during a failover scenario. Said another way, you don't always need the exact tier one hardware to stand up another copy of the application, but you definitely need in-sync data and access availability. Users are more forgiving of a lack of speed than a complete lack of access.

Knowledge Check

So how do we go about measuring the sheer breadth of an application?

- Services running within the app
- External providers of data to the app
- External consumers of data from the app
- Servers and OS versions required to run the app
- Storage systems required to host the app data

All of these components are baseline requirements when it comes to understanding the needs of your application and determining its portability.

Migrate Successfully with Application-Centric Workload Visibility

For too long, we have placed focus on the underlying infrastructure of a data center and paid less attention to which applications would be affected in a migration scenario. Having a complete understanding of the workload and all application dependencies can prevent a lot of patch work and cleanup after the migration is complete.

When you use the application itself to define the underlying resources it is consuming, it is much easier to track supporting infrastructure and resources that need to migrate or travel with the application when it moves.

This top-down approach leads to more successful migrations, and to only replicating and moving the data and supporting systems required to support the app or apps being migrated.

It is critical to the success of a migration that you understand the resource requirements of any and all applications that need to have the ability to move between data centers. This will empower IT departments to be prepared for just about any given situation.

Working backwards can lead to the best results. Start at the top by outlining end goals, and backtrack down the stack to define any infrastructure, software, licensing, and other requirements necessary to allow you to migrate your workloads around seamlessly, whether that's between data centers or clouds, or a healthy mix of both.

These types of best practices can lead to a much smoother experience when fully migrating an application or workload to the public cloud, or doing iterative migrations between servers across myriad public and private cloud instances.