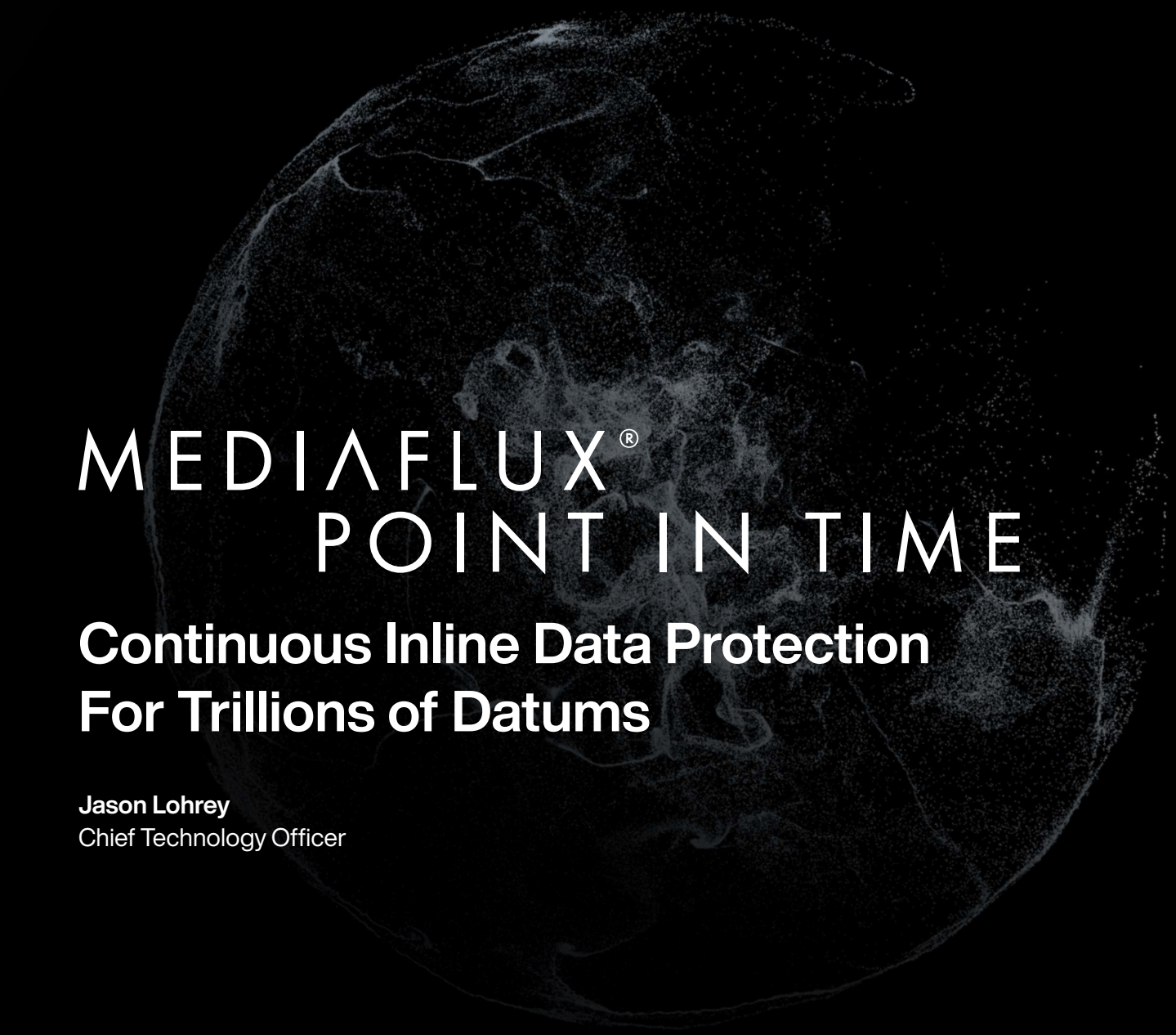
WHITE PAPER
VERSION 1.0 DEC 2022



MEDIAFLUX® POINT IN TIME

Continuous Inline Data Protection For Trillions of Datums

Jason Lohrey
Chief Technology Officer

Contents

The Problem With Backup As We Know It

We've always required ways of protecting our data because sometimes people make mistakes – in 1986 I managed to inadvertently delete 80,000 words of my Aunt's manuscript with a simple, but highly memorable: `'%DEL *.*'` - and sometimes there is corruption, malicious activity and increasingly there are external people trying to seek financial advantage through crypto-locking ransomware or other means.

Keeping duplicate copies has always been a way protect valuable data. If lost, damaged, or destroyed, then hopefully it can be recovered from one of the other copies. It is important that the duplicates are kept somewhere that won't suffer from the same failure *at the same time* as the primary copy. That typically means using "genetically diverse" storage – if the primary copy is on disk, then the duplicate copy should be on tape or immutable storage such as optical disk. If the data is *very* important, there should be more than one additional copy, and each copy should be stored on different types of storage. They should be geographically separated to account for natural disasters such as earthquakes, floods, etc. With each additional copy, the probability of failure to recover decreases significantly. The cost of storage also increases with each additional copy, and that can be an issue when the amount of data is large, both in number and in size.

For decades, protecting data from loss has been the job of backup systems. These are specialist applications that scan storage, such as file systems, at scheduled intervals and create copies of new and changed data at some other location. Typically, duplicate copies were stored on tape. Increasingly, cloudbased storage is replacing tape. Regardless, the mechanism is the same: scan the primary storage and keep a copy elsewhere. When backup systems scan the storage, they do so at discreet points in time. That may be once or twice a day, sometimes more regularly depending on the amount of storage to be processed. This means any changes that happened *between* backups, or that have occurred since the last backup will not be recorded and cannot be recovered.

The Recovery Point Objective Is Not What It Could Be

The length of time between any point in time and the last backup is known as the Recovery Point Objective (RPO). Ideally, that should be zero – every point in time can be reconstructed – and no data will be lost. In practice, traditional backup systems have a relatively large RPO (hours, days and sometimes longer), which significantly increase the probability of lost data.

Axiom: A Recovery Point Objective (RPO) that is greater than zero is a compromise.

Scanning backups will often (more often in larger systems) make copies of files that are in the middle of being written to – it may take two or more backup cycles to create backup when the files are not being written to, and that can significantly *increase* the RPO. The more active the filesystem, the greater the probability that incoherent files will backed up.

Some backup applications will avoid continually scanning the storage by processing the journals in the filesystem. An initial scan is required to establish the state of the system, but thereafter, the journals can be used to pick up changes without needing to scan the entire filesystem again. This requires the backup application to understand the specific file system and the format of the associated journals. Not all filesystems allow this form of integration, and it does not work for other forms of storage, such as tape or object storage – it's a specialised form of backup. If the two systems become disconnected for an extended period, a full scan is once again required to bring the systems into synchrony.

Traditional backup systems will generally perform two types of backups. A "full" backup is a complete copy of the data. That can take a long time to generate and will consume as much space as there is original data. To reduce this overhead, the backup applications will generate "incremental" backups between the full backups - these store just those files that have changed. This reduces the total amount of data that needs to be stored by the backup application but does mean reconstruction is harder because restoring to a given state will require restoration of the last full backup prior to the recovery point in time, followed by restoration of every incremental backup up to the time of restoration. Incremental backups only use the space required to store the changes, which is good. The full backups are problematic because of the amount of additional data they generate. For example, attempting to create a full backup every month of 100 petabytes of data – that would generate 1.2 exabytes of data every year!

Some filesystems have a form of backup known as snapshots. These are checkpoints where the state of files (all the blocks associated with every file) are known. Subsequent changes to each file will generate new blocks through a process known as "copy on write". Snapshots are lightweight and quick to create.

They may be generated hourly to reduce the RPO. Like backups, they don't record every point in time – if there are significant changes to files between the discreet snapshots, then those will be lost. Snapshots will not replace the need for traditional backups which copy data to an external location. Not every storage technology supports snapshots.

In recognition of the shortcomings of discrete backups and snapshots that allow changes to be missed, there is a set of products that are classed as providing *continuous data protection* (CDP) – that is, every change in the filesystem is recorded, allowing recovery to every point in time. That concept has been around for several decades – it is a technique was patented in 1989 by British entrepreneur, Pete Malcolm.

True continuous data protection achieves an RPO of zero. It's hard to assert through the obfuscation of marketing which products, if any, achieve an RPO of zero. Many refer to CDP and then go on to say they are "near real-time". Quite a few products are integrated with virtual machine interfaces to intercept and journal every file write – that is, they are in the data path only within the context of a virtual machine. Virtual machines are popular, but they represent only a fraction of the global data generated – estimated to exceed 180 zettabytes by the end of 2025. Even if the existing systems *could* intercept every possible write (in real-time), it's simply not possible to store every write for all time because systems with high-data churn rates can generate orders of magnitude more data I/O that is stored. That will only be possible when there is infinite storage that is infinitely fast, and we generate orders of magnitude less data than infinity. That is not plausible in the current universe.

For a backup solution to offer sustained continuous data protection it would need to provide IOPS (for write) and data rates equal to or better than the storage systems they are protecting. That is, CDP might work under certain conditions – notably when paired with virtual machine infrastructure and relatively small amounts of data – but unlikely to be able to sustainably handle very large amounts of data. Most of the world's data is unstructured and sits outside of the realm of most backup applications.

A complete data resilience strategy will need to focus on both a) creating sufficient copies of changed data to avoid a loss of continuity and significant business or personal loss, and b) how quickly a someone can recover from loss. The benefit of recording every change is undermined if it takes a significant time to recover lost data.

The Recovery Time Objective Is Not What It Should Be

The time that it takes to locate the right point in time and restore to that point is known as the Recovery Time Objective (RTO). Continuous data protection does not mean that data can be immediately recovered with zero downtime and disruption and many (most, if not all) products that are classed as providing continuous data protection do not state an RTO of zero.

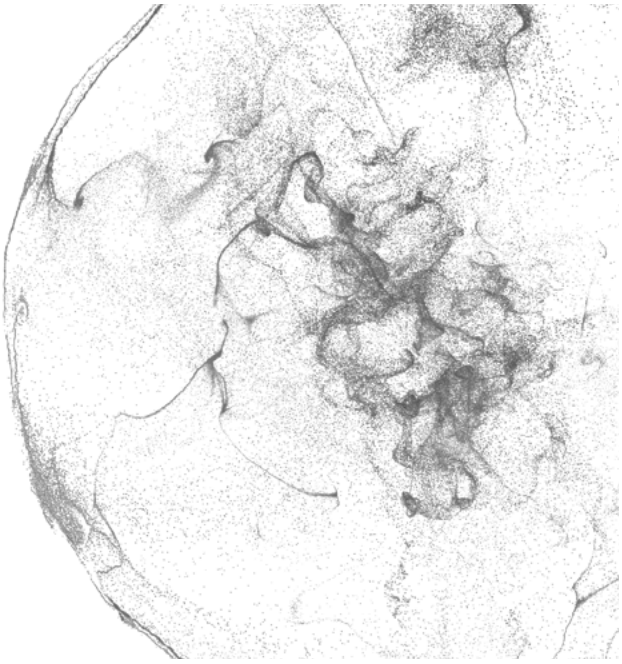
Axiom: A Recovery Time Objective (RTO) that is greater than zero is a compromise.

Let's examine a typical recovery process. The process is not necessarily straight forward. The end user will need to ask the IT department to recover their data. The first questions the IT department will ask are likely to be, "what was the name and path of your data, and when did you have it?". Often, these are not easy questions to answer. The backup system will be interrogated to locate candidate copies which can be restored into the primary storage. The end user will examine the restored copy to confirm it is (or not) what they seek to recover. If it is not, other backup sets will be restored and so on until the target data is properly located. This might be a process that takes days to find (and recover) the correct data. Finding data in a large system is problematic. I once had a customer say, "we spent €5 million on optical scanning of documents, but now we cannot find them" – that was because they have billions of files, and finding files in amongst billions is not easy. Finding data across all time is a significantly more difficult task.

Systems with snapshots allow end users to locate their data. In a filesystem, each snapshot will present as a hidden directory that will contain subdirectories for each snapshot point in time. The user will traverse as many as they need to find their files. If they are not sure when those files existed, they may need to spend a lot of time in those snapshot directories.

When we think of storage systems, we traditionally mean "file systems". Those are directly attached to the compute or attached via network protocols to storage that can be shared by many people in an enterprise. File systems provide random access storage at high rates of I/O operations. File systems are not inherently resilient to failure, including human errors such as accidental file deletion – which file systems can manage tens of billions of files or more and at the same time achieve high levels of resilience, where resilience includes genetically diverse storage that is geographically separated? Very few, if any.

Object storage is increasing popular for entitles with a lot of data and who see object storage (on-premises and in the cloud) as a way to reduce storage costs. Unlike file systems, object storage is not suited to random access, but rather for data that can be sequentially read and/or written. Dealing with failure and recovery also needs to be considered for object storage. Does your cloud provider guarantee they will not lose data? Unlikely. Even though some object storage systems provide snapshots like file systems, those suffer the same issue because they are at discreet time intervals. Data resilience is often overlooked when dealing with object storage, particularly when that storage is in the cloud. Object storage is no different – it also needs continuous data protection.



The Scales Are Changing

We are now in the *Data Age* where almost every human endeavour is data-driven. Millions of files are being replaced by billions to trillions, and terabytes of data are being overtaken by petabytes to exabytes and beyond. At those scales, the traditional methods of backing up data are increasingly unviable, and, in many cases, organisations will simply forego backing up at all because it takes too long and/or the cost of those backups is too expensive. That is a precarious place to be, particularly when the probability of loss is amplified due to external factors such as ransomware and malicious actors.

Every backup solution we are aware of is external (an add-on) to the primary storage system. They are not *part* of the storage system itself – rather, they are processes and applications that operate outside of the data path and as such they will always be at a disadvantage as the scales increase.

In our view, they only way to achieve resilience at scale it to combine data protection with the storage fabric so they become one and the same thing. Only when that happens will it be possible to achieve resilience for hundreds of petabytes or more and hundreds of billions of files.

MEDIAFLUX POINT IN TIME

Mediaflux Point in Time

Mediaflux Point in Time solves the inherent problems with backup, by eliminating backup entirely. Instead, the focus is on *resilience*, driven by the following objectives:

- Every significant change is recorded and immediately accessible,
- Deletion is non-destructive, unless it passes through some secure and gated process,
- The minimum number of copies will be maintained for any file or object – the number required to meet the objective for storage diversity (and no more).

The objectives are achieved by creating something unique. Mediaflux Point in Time is a virtual storage layer that sits in front of existing storage systems (flash, disk, object, tape, cloud, or any combination of these) and presents a single global namespace that provides continuous data protection for scales of billions to trillions of files and petabytes to exabytes of data and beyond.

Mediaflux is software that can be installed on dedicated hardware, virtual machines, containers, and cloud infrastructure or any combination of these including hybrid deployments. In future, it will be available as a bundled hardware appliance. Multiple nodes can be configured in a cluster for scale-out I/O performance.

The Recovery Point Objective Is What It Should Be

Mediaflux Point in Time is a temporal data fabric that records metadata and significant data changes and performs continuous backup forever in real-time. It enables immediate navigation to any point in time – rewinding the data space with the click of a button or API call. The initial objective was to address the problem of resilience for *big data* – scales in the tens to hundreds of petabytes and beyond. However, along the way it became clear that the elimination of backup as we know it is equally good for much smaller holdings of data.

Mediaflux Point in Time supports standard file system protocols including NFS and SMB, allowing concurrent access to the same data through different protocols at the same time. It supports many other protocols such as S3, sFTP, DICOM and more as well as API access – it's a system for data, that presents through many different protocols.

Mediaflux Point in Time will leverage existing storage infrastructure and allow the virtualisation of storage from different vendors and vintages – presented through a single mount point. End-users no longer need to access different storage – simply add additional storage when you need it, replace storage without affecting users and manage cost optimisation with intelligent, metadata-driven data placement and lifecycle management.

Recording the structure of the file system throughout time requires some amazing technology – it would simply not be possible without the right database engine. To achieve this, we created XODB® – a database that combines object, spatial and time series – trillions of datums, high performance and an incredibly small footprint. XODB has been deployed and refined since 2007 and is at the core of Mediaflux Point in Time.

Mediaflux Point in Time *is* the data path. Whenever data changes, additional copies are immediately requested without delay. The number of copies, and the type of storage to which those copies are made is automatic, driven by policies and metadata. For example, out of 100PB, perhaps only 20PB is critical so it's important to keep 3 distinct copies on genetically diverse storage, whereas the remaining 80PB is less critical – requiring only one additional copy, and in some cases no additional copies at all. The selection process, and the policy decisions are driven by the context encapsulated in metadata associated with each file. That metadata may be contextually applied, automatically extracted from the content of the files, machine generated through analysis of the files, or added by people.

With Mediaflux Point in Time, there is no longer a need for backup. Redundant copies are made when the data changes, and only the minimum number of copies are created and maintained – a function of the configured policies for each file. Those policies can apply to all files or uniquely to individual files or any combination. This results in a significant reduction in the total amount of storage required to achieve a given level of resilience when compared to traditional backup.

Mediaflux Point in Time eliminates the issue of backup processes failing to keep up because there is no external backup process.

The recovery point objective (RPO) – the time between any given point in time and the last backup – is almost zero. It's not precisely zero, because the laws of physics apply – it takes time to transmit the additional copies to redundant storage. The RPO for metadata changes is zero. For many files, the time to create a redundant copy will be in the order of milliseconds to seconds.

MEDIAFLUX POINT IN TIME

The Recovery Time Objective Is What We Want It To Be

The process of recovery is equally as important as the process of ensuring there are sufficient and timely copies of data. It should not take hours, days, or weeks to recover data and the process should be self-service.

Mediaflux Point In Time provides end-users with the ability to traverse the entire time continuum to find missing or misplaced data. This can be done through an API, or through a browser-based GUI. One of the difficulties with recovery is knowing where files are, and when they existed. The Mediaflux Point in Time GUI solves this issue through two significant features:

1. Leveraging the power of XODB, any authorised person can perform an arbitrary wildcard search for file names in milliseconds for billions of files. These searches apply across all points in time, allowing people to find files at a point in time before they changed name or locations, and
2. The Mediaflux Point in Time GUI enables a person to see files at all points in time at the same time, so you don't have to remember exactly when a file existed to see it.

Once the correct point in time is found, it can be applied to existing file system shares and mounts. The file system will then present as it was at that point in time and accessed as a regular file system using standard tools. It's easy to jump between different points in time. There is no need to "recover" data – it's simply all there at that point in time.

The time it takes to recover is up to you – log in, and immediately access files at any timepoint – a recovery time objective (RTO) of zero.

The objective was to create a system for making backup and recovery continuous and immediate. There are contemporary benefits for first line of defense in protecting against crypto-locking ransomware, a process that encrypts files, deletes the original and then demands payment to decrypt your data. With Mediaflux Point in Time, crypto-locking ransom is ephemeral – simply rewind the file system to the point in time immediately prior to the attack.

Preventing Damage in the First Place

Mediaflux Point in Time allows reconstruction of the data space (or file system) to any point in time. It enables the unwinding of accidental or malicious attempts to delete or modify data simply by reversing time. However, much of this can be prevented in the first place by enabling multi-factor in the data path.

Mediaflux Point in Time provides multi-factor in the file system. It is used to confirm identity during authentication and is also used when authorising data operations. For example, deletion may not be allowed unless the person attempting to delete is authorised and acknowledges the deletion using their phone as a second factor. Imagine hard destruction, or modification of data requiring a quorum of people to agree to destruction before that can occur in the first place, and that is only possible with multifactor.

Mediaflux multi-factor authentication and authorisation (MFA&A) is part of the Mediaflux data fabric. It's available to every protocol including file system protocols such as SMB and NFS.

Mediaflux MFA&A is provided by Mediaflux Pocket – a mobile application for on iOS and Android.