komprise

Stubs and Symbolic Links

Comparing two popular ways of representing file content and locations.

Overview

The "Information Big Bang" explosion happened a few years ago and we are now in a rapid digital expansion similar to the first few years after the original one that formed our universe. Thanks to lightning-fast adoption of applications like 4k (soon 8K) video, medical imaging, genomics, ADAS (Advanced Driver Automotive Systems), IoT and AI-based analytics, our data universe, much like the physical universe, is also rapidly expanding. Analysts estimate that 90% of all data has been created in the past two years, with 80% of that data being unstructured, not current, with its value reducing dramatically with time. Data management will create a worldwide need for 331EB (Exa Bytes: that's 1,024 petabytes!) of storage-based capacity by 2021.

Almost every enterprise has adopted or is considering secure, seamless and flexible data storage, data management, retention and archiving policies and tools. All data is not the same: some is more urgently and more frequently required than others. More frequently used or more quickly needed (lower latency) data would need to be kept on faster-responding, quicker access (and more expensive) storage media. Infrequently used, less important or less urgently needed data can be stored on less expensive media, offsite from a company's offices, and "in the cloud". This classification of data is called Data Tiering, and is extremely important to consider for cost and management factors.

Identifying hot and cold data enables the archiving or movement of cold data to cheaper storage and backup approaches, which ensures that the right data is consuming the right level of resources. In all cases, the data needs to be moved and accessed seamlessly, without impacting its users and applications. Typical data tiers include:

1. **Flash storage** – When high value and high performance is more important than cost.
2. **SAN (Storage Area Network) and NAS (Network Attached Storage) Storage arrays** – Medium value, medium performance, medium cost.
3. **Object Storage** – Less frequently accessed data that's great for storing unstructured data, lower cost than commonly used NAS/SAN.
4. **Cloud** – Long-term archival for data that is almost never accessed.

Most data generated today is unstructured – videos, audio, genomics data, seismic data, documents, IoT data, electronic design data are all examples of unstructured data. Unstructured data is growing rapidly, and needs to be managed efficiently across different tiers of storage.

Data is moved and managed by computer systems using several tools or protocols, ranging from the industry standard to proprietary. Moving, copying or archiving data requires a thoughtful approach, and most businesses have different needs and priorities for each. Regardless, there are some commonalities and salient differences to understand.

While archiving can save costs, if done in a brute-force manner of simply moving the data to a lower, more cost-effective tier, it can be disruptive to users and applications not aware that some data that was on Tier 1 has now been moved to Tier 3. Users are unable to find the data when they need it and applications can fail.

This document discusses two common methods of preserving information in a file system when data is moved or copied to a new location to provide seamless, transparent archival without such disrupton.

Links: Stubs vs. Symbolic

In computing, a symbolic link (also called symlink) is a term for any file that contains a reference to another file or directory. A symbolic link is a file-system object that points to another file system object, where the object being pointed to is called the target. A stub is a small file left in place of the moved file and which tells the accessing entity the file isn't where it was and has been moved to an identified alternate location. Both stubs and symbolic links intend to preserve access to moved files – but they work very differently. This document will discuss the relative merits and differences between these two common methods of preserving links to moved files, and a deeper dive into Komprise's method of using dynamic symbolic links.

Many traditional data management solutions use static stubs – they replace moved data with a pointer to the new location of the file. A stub exists to tell the accessing entity the file has been moved to another location.

There are several problems with this approach that stem from the fact that stubs are proprietary and static. The biggest problem is that if the stub file is corrupted or deleted, the moved data gets orphaned.

Stubs are proprietary, so you need either storage agents so each storage can understand the stub, or a proprietary interface to each storage. This means they are not portable, and managing revisions of stubs along with storage upgrades or migrations is time consuming, cumbersome, and error-prone. Since stubs are proprietary, adding storage agents impacts the data path of both hot and cold data which is also undesirable as it can cause

performance degradation of hot data and metadata.

Symbolic links are transparent to users; the links appear as normal files or directories and can be acted upon by the user or application in exactly the same manner. A symbolic link contains a text string that is automatically interpreted and followed by the file system as a path to another file or directory. This other file or directory is called the “target”. The symbolic link is a second file that exists independently of its target. Symbolic links are standard protocol constructs in NFS v1 and above and SMB 2.0 and above. So, they do not require any agents on each storage or any proprietary interfaces.

Komprise Dynamic Links

While stubs are proprietary by nature, symbolic links are based on industry standard protocols and are widely recognized. This precludes the need for any extra agents or overhead to manage them. Komprise took the native advantages of symbolic links and innovated further, dynamically binding them to the file at runtime, akin to a DNS router. This makes the links themselves disposable – if a link is accidentally deleted, Komprise can restore it.

When Komprise archives data from a file system, it replaces the original file with a dynamic link address: resilient, always available and flexible. There are several benefits to the dynamic link approach:

a. The link itself can be deleted and Komprise can repopulate it.
b. The file can be moved again through its lifecycle and the link is unchanged.
c. Allows Komprise to not sit in the hot data and metadata paths because it uses standard file system constructs.
d. The link is resilient when coupled with the high-availability architecture of Komprise and has no single point of failure.

Let us explore these advantages in more detail.

Once Komprise moves a file and replaces it with a dynamic link, if the file is moved again – say, for example, after the first archive a file to an object store and another later to the cloud – the dynamic link address does not need to be changed. This eliminates the headaches of managing links. Users and applications continue to access the moved data transparently from the original location even as the data is moved throughout its lifecycle, without any changes.

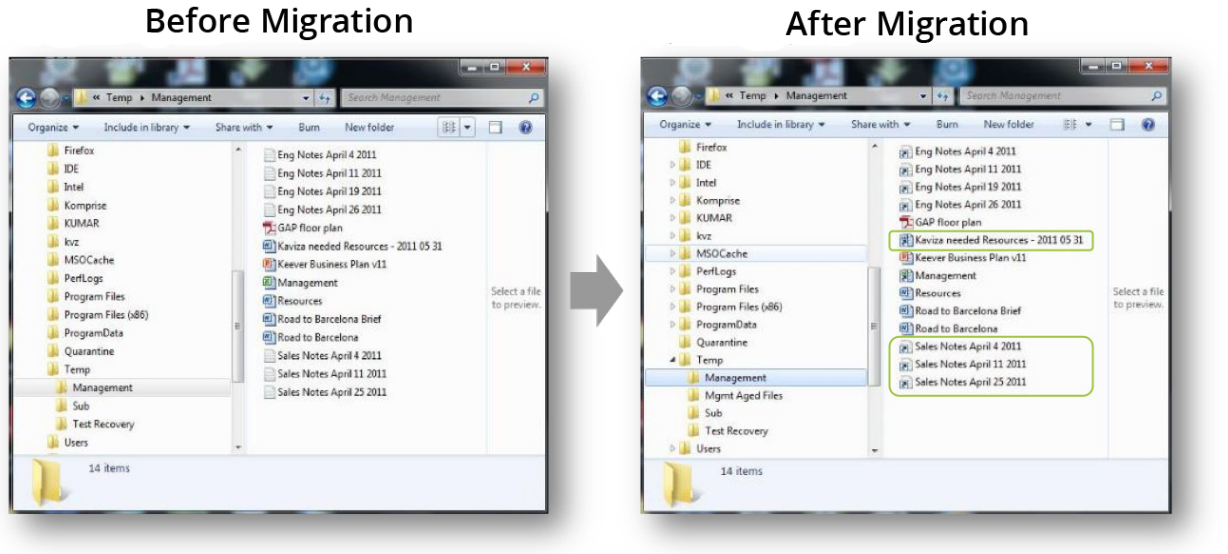

Figure 1: Before and After Migration: SMB (Windows) Systems

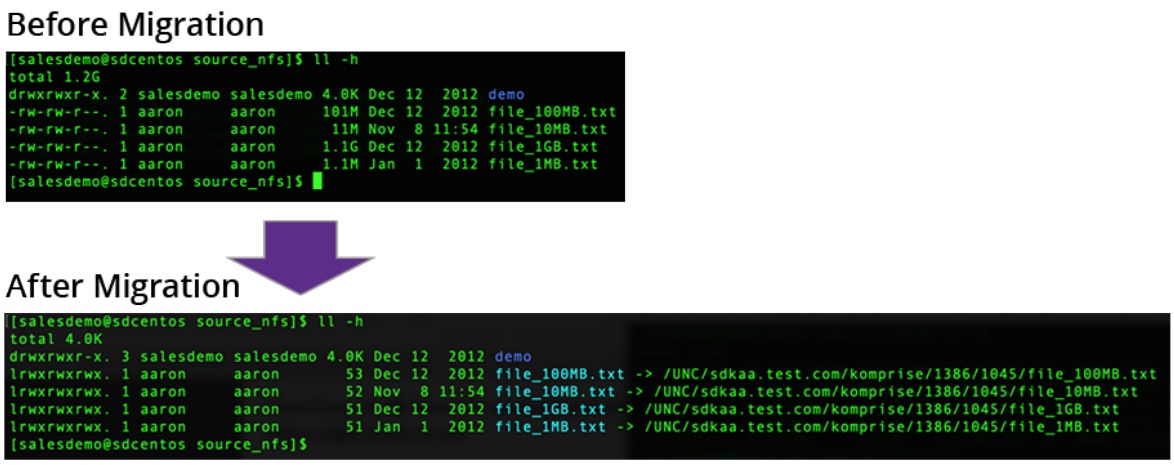

Figure 2: Before and After Migration: NFS (Linux) Systems

By leveraging a standard protocol construct whenever possible (in more than 95% of all cases), Komprise is able to deliver non-proprietary, transparent data access without getting in front of the hot data or metadata. If a user accidentally deletes the links on the source, Komprise can repopulate the links since the link itself does not contain the context of the moved file. Data can be moved from one destination to another (e.g. for ongoing data management) and there are no changes to the link.

To the user, this means no disruption. Users' storage teams won't get bothered with help desk tickets from employees unable to find their data, and their applications will be able to keep access to their data. Users and applications that rely on the data that has been moved by Komprise are unaffected.

At a glance:

Static.	Dynamic.
If the stub is deleted, the data is orphaned.	If the link is deleted or moved, data remains accessible and the link can be repopulated.
Proprietary: need either storage agents so each storage can understand the stub, or a proprietary interface to each storage.	Industry standard: Native to file servers and standards-based. Storage vendor agnostic.
Not portable: need to manage revisions of stubs along with storage upgrades. Migrations can be problematic.	Portable: based on standard SMB and NFS/CIFS symbolic links; no need for any agents on the storage. Works across multi-vendor storage.
Can cause major disruptions.	No disruptions. Users and applications that rely on archived data remain unaffected.
Can cause performance degradation to hot data and metadata access as they are in the data path.	No performance degradation of hot data and metadata as the link access is only for cold data outside the hot data paths.

Table 1: Comparing Stubs and Komprise Dynamic Links

While it's clear that symlinks offer superior resilience and flexibility than static stubs, it's not that stubs are never useful or never used by Komprise. While most file servers support symbolic links, there are a few situations where the file servers do not support symbolic links. For such file servers, Komprise uses stubs that are dynamic. Dynamic stubs point to Komprise, which redirects them to actual files in the target. This ensures that even if the stub is lost, the corresponding file on the target can be accessed via Komprise and the stub can be restored. Komprise's dynamic stubs can be made similar in size and appearance to the original file.

Summary

Komprise delivers Intelligent Data Management software that works across multi-vendor storage and clouds to help businesses manage unstructured data growth while cutting costs. Komprise analyzes file and object data to identify hot and cold data, and archives, migrates and moves data by policy without any changes to user and application access. With data accessible in its native format no matter in which tier it resides, Komprise users achieve maximum flexibility while driving down costs.

komprise

Komprise, Inc.
1901 S. Bascom Ave. Suite 400
Campbell, CA 95008
United States

For more information:
Call: 1-888-995-0290
Email: **info@komprise.com**
Visit: **komprise.com**

For media requests email:
marketing@komprise.com