storfirst EAS

seven10 STORAGE SOFTWARE

MANAGE DATA SMARTER™

StorFirst Enterprise Archival Storage
Whitepaper: File System Archiving for Persistent Data

Introduction to Persistent Data

Since the advent of low cost/high performance communication infrastructure many of today's enterprises have become highly leveraged on immediate access to their historical knowledge. Examples of this expectation of "infinite institutional memory" range from medicine, where new digital imaging modalities create gigabytes of clinical data per patient; all of which is expect to be available to clinicians with minimal delay virtually anywhere in the world, to the financial sector where transactions and email are retained based on policies closely tied to litigation and regulatory requirements. And now we see the entertainment industry discovering new value in digitizing its assets to provide on demand access to increasing smaller and specialized audience segments.

This new class of data: fixed content accessed infrequently is commonly referred to as persistent data. Persistent data presents some unique challenges to today's common infrastructure. The requirement that the data be available quickly for access precludes the application of traditional backup and "shelf management" approaches to old information. In fact most IT professionals realize that archived tapes represent a "Write Once Read Never" approach to data management which no longer satisfies user's requirements for data availability. The size and relative infrequency of access also challenges traditional SAN fabrics which have been designed to support online transactional processes. Worse still is the lack of file system support for multi-petabyte archives and imbedded retention policies required by persistent data.

While significant efforts have been made to address the hardware cost of persistent storage through new low cost innovative storage systems, little has been offered by traditional file systems to meet the functional requirements of the problem domain. Users are quickly discovering that traditional on-disk file systems such as NTFS have been developed for general purpose computing and lack the scalability, data security, and performance to cope with the infinite influx of fixed content. At Seven10 we applied our collective experience with file systems & persistent data to develop StorFirst EAS™ (Enterprise Archival Storage). This paper describes the challenges we perceived during the design process and provides detailed information about our implementation of EAS™.

Characteristics of Persistent Data

Persistent data and its source applications have the following characteristics:

Immutability – Persistent data typically does not change, and is not usually modified or edited by users.

Example: Radiologists cannot alter a patient's x-ray images.
Example: Trader's can't alter transaction data.
Example: Corporate Email and Documents are discoverable assets and can't be modified.

Policy driven long-term retention – Needs to be retained for extended periods, perhaps even indefinitely.

Example: Under HIPAA guidelines, healthcare records are maintained for the life of the patient & beyond.
Example: Unenforced corporate document retention policies make all documents discoverable.

Infrequent access – Users only need to access a small portion of the data at any one time.

Example: Medical images infrequently need to be loaded for viewing.
Example: Email and documents for litigation support.
Example: Historical transaction data for analyst or fund manager access.

High throughput and continuous streams – Streaming access to bulk data is required and read/write access is mostly sequential, rather than random.

Example: Multimedia recordings of a doctor's notes, such as audio or video files.
Example: Surveillance video and remote sensing data are all stream intensive.

Event-driven – Usually created by a specific event that initiates the retention of that data.

Example: A patients annual health exam, all records can be archived for storage.
Example: A criminal investigation requires selective recall of surveillance images.

Business vital – Not mission-critical in terms of needing instantaneous access, but data must be accessible in seconds or minutes.

Example: Historical patient records do not have to be immediately accessible in the event of a disaster, unlike mission-critical transactional data.

Traditional file systems have focused on high performance with large numbers of small transactions. They also rely on access control and permission mechanisms to prevent data modification and have no intrinsic support for immutable file types.

EAS Design Philosophy

The driving philosophy behind StorFirst EAS is simplicity. We actively avoided features which add complexity and are only peripherally associated with the problem domain. The following core principals have been strictly adhered to:

No Client Software Required – While persistent data can be managed by relatively few servers it may be accessed by hundreds or thousands (perhaps millions) of users in a given application area. Installation of file system extension software on clients is an unacceptable and unnecessary hurdle for end users.

No API Required – Early attempts at persistent data storage have required the use of proprietary API's to store and retrieve data. These API's add little other than vendor lock-in to specific hardware implementation. Ideally access to persistent data should be transparent and hardware agnostic.

Ease of Scale – Other efforts have preserved the underlying file system and have tried to virtualize the assets under management to achieve scalability. While storage virtualization has many important applications it adds unnecessary complexity to merely achieve high density.

Our approach has been to provide the user with a familiar environment and hide the details of the underlying implementation whenever possible. For example, EAS provides a standard CIFS interface and relegates device management to a simple explorer like data administration UI.

"Entia non sunt multiplicanda praeter necessitatem" William of Ockham

EAS Implementation

EAS is a plug-in file system designed for Microsoft Windows 2003 server. Several features distinguish from other file systems on Windows server and provide unique facilities for persistent data:

Dynamic LUN Aggregation – StorFirst EAS treats hardware or virtual LUNs as the atomic unit of aggregation. In other words any number of LUNs can be combined into a single virtual file system (VFS). LUNs are dynamic and can be added, deleted or moved offline using the administrator UI. This dynamic LUN management allows removable and fixed media to be mixed in the same VFS. This feature allows EAS to mix magnetic disk, tape, and future storage media (such as holographic disks) into a single logical volume.

N-Tier Virtualization of Heterogeneous On-Disk Formats – EAS optimizes the on disk formats for the media in question. For example EMC's unique C-Clip format is used for Centera while a proprietary WORM on disk format is used for FC Disk, MAID, or DAS. This allows all volumes to be self describing. This unique ability allows volumes to be moved between EAS installations. Most other efforts at virtualization have introduced assembly specific limitations which prevent LUNs from being moved to other server environments. For example, this fragility greatly limits the usefulness of NTFS dynamic disks.

Support for N-Tier data replication allows EAS to retain data on access-appropriate technology. The integrated retention engine (see below) use policy to preserve data on the lowest cost media possible.

High Performance External Metadata Services – EAS maintains all necessary metadata to make LUNS self describing. The same metadata is also externalized by the EAS metadata service. This has significant implications for performance. Tradition file system require that metadata on the volume be scanned for queries and searches. This imposes significant performance challenges in very large or power-managed storage systems.

Searching such systems with a traditional file system can result in poor performance (imagine a full meta-data search on a 5 PB volume) or pathological behavior (disk thrashing in a full power down MAID implementation). The highly indexed metadata service provides very response query operation with less than 0.1% storage overhead.

True Worm – EAS was designed as a true worm file system. Files can not be altered or removed by any end user actions. The retention policy manager is responsible for purging files. The software methods it uses are not exposed in any user level code or API. File systems which rely on permissions to simulate WORM can easily be circumvented.

Massive Single Sever Scalability – EAS currently imposes a limit on a single server of about 32Billion files. The practical limit at the time of this writing using high density disk storage is about 10 PB/server. This provides 50 times the scalability of NTFS.

High Availability – EAS offers two methods of high availability support. The feature driven model is to leverage EAS' asynchronous multi-target writes which provides built-in replication by dynamically writing to all available tiers of storage. This represents a new approach to lifecycle management and provides immediate access to data should and one of the tiered storage subsystems fail. The baseline model supports a fast cold-start re-inventory process. This capability allows a replacement server to quickly remount some or all of an EAS assembly rebuilding the meta-data storage directly from the on-disk file system(s).

Integrated Retention Policy – EAS supports simple user programmable retention based on file longevity. Data can not be purged from a storage tier except through the deliberation action of the policy manager.

Application Driven Meta Data Extensions – EAS supports an extensible meta-data schema. This schema is exposed to the application layer in the form of extended file system attributes. Advanced versions of the retention manager use these attributes (managed by the meta-data service) to provide fine-grained retention policy down to the file level.

MS Active Directory Support – EAS uses AD and authorization manager to provide user and role based security through publish Microsoft APIs.

Conclusion

StorFirst EAS provides the essential elements to support long-term compliant storage without blurring the line between OS services and application level responsibilities. Simple to implement and hardware agnostic, the Seven10 file system eliminates complexity and many of the limitations of API driven fixed content management systems. These attributes bring you all the virtues of the cradle-to-grave lifecycle concept, including lower storage cost, policy driven data retention, lower administrative burden and data immutability and security.