

ILM in the Datapath: Lack of a Standard Framework

**Henry Newman
6 August 2008**

Agenda

Why do we need ILM

What is broken and why

How we got here

Some Thoughts

WHY WE NEED ILM

The data management problem

What is Needed

First of all users and system

Administrators have different needs

Users need to be able know what they have and manage their data

Administrators need to be able to implement ILM policy based on business rules

Current frameworks have:

- No ILM support for archival systems***
- Only have limited ILM support that is proprietary***

Most support that is currently available is designed for Regulatory Compliance

Examples for User Needs

Tag a file with metadata

- HDF-5 or NETCDF-4 for example*

Retention policy

- Delete in 4 years*
- No DR copy needed*

Permissions

- What users can see the file*
- Who has seen the file*
- Who gets custody after the user leaves or dies*

Usage

- The file will be used again soon*
 - Try to cache the file*
- The file might be extended within 10 days*

ILM Benefits

Users can manage their data with a UNIX file system framework

- The correct data will be saved over time not just everything***
- Users will know how and why a file was created***
- Per file deletion policies can be attached at create rather than panic calls from the system staff when storage is reaching a limit***

System Administrators can implement system wide policy

- Policy will no longer be specific to a single vendor software***
- Very likely reducing the growth rate of data***

Examples for Admin Needs

Project/Account ID

- ❑ ***Mainframe and other systems in the 70s and 80s had project allocations and accounting by project***
 - ❑ ***The same is needed today for allocations***

Default policies

- ❑ ***Might want a policy to delete files after X number of years***
- ❑ ***Default DR policy***
- ❑ ***Default file reliability policy***

Information

- ❑ ***Access count***
- ❑ ***Accounting by user, group, project/account ID***
- ❑ ***% sent to DR***

What is wrong with the current framework

All ILM functions must be done in user space

Each vendor is doing something different

Many vendors are looking at the Sarbanes/Oxley problem

Way different requirements than archival ILM requirements

None of what is being done moves from system to system as there are no standards

WHAT IS BROKEN AND WHY

What the user can and can't do today and the history

What Is Broken

The open() system call has not changed in the last 20 years

What the user can do from an application in a standard way is limited by open()

- ❑ Many vendors have additions such as using POSIX extended attributes such as is used for ACLs***
- ❑ Scaling and reliability require more information down the data path***
 - ❑ True for all I/O not just HPC***

Per-file metadata does not exist in system space

ILM is not part of the standards process except to a limited degree by SNIA

Some Examples of Our Broken I/O World

T10 Protection Information (PI) (was DIF) allows a checksum to be transmitted from the application to the disk drive

- What standard framework exists to allow the application to do this?***
 - None is the answer***

T10 OSD allows a variety of I/O hints and security features per object

- What framework is available to access?***
 - Again none is the answer***

The Requirement is Across Agencies

Per-file metadata, reliability and policy is needed by many, and is implemented currently in user space by vendors

- ❑ *LOC, NARA, DoD HPCMP, NSF and others*

Industry needs the same thing

- ❑ *Pharmaceutical companies for drugs*
- ❑ *Medical devices*
- ❑ *Manufacturing*
 - ❑ *Planes, trains, and cars*
- ❑ *GIS applications for*
 - ❑ *Utilities*
 - ❑ *Oil and gas*

HOW WE GOT HERE

A trip down memory (I/O) lane

The I/O Stack

The USL/XOpen/OpenGroup
(POSIX), SNIA(no standards yet)

Application/User space

T10, T13, IETF (NFS), SNIA
(no standards yet) OpenFabrics,
T11 (FCOE)

Drivers, OS and File
System Interface

T11, IBTA, IEFT, Ethernet,
T10, T13, SNIA (no standards
yet)

Storage and Transport

The data path today is basically the same as the data path
of 20 years ago

Limited improvement for management (errors or system)

Standards Process

User space is controlled by the OpenGroup

- ❑ ***When did the open() last change***
 - ❑ ***Likely before 1990***

SCSI and ATA are controlled by T10 and T13

- ❑ ***New Standards for PI and OSD, but no changes above in the stack***

Channels are controlled by T11 (FC), IEEE for ethernet, IBTA for IB, IETF for IP and NFS

Seemingly not a lot of intergroup communication

The USG worked on standards in the 70s and 80s but no more

- ❑ ***DARPA, NSF, NIST, DoD and others***

Standards Process is Disjointed

No standards for

- File systems other than the interface***
- HSM policies***
- Data reliability***
- Per-file metadata***

Lots of different standards bodies as mentioned

- Some have competing goals***
- SNIA is industry controlled with little to no outside input and little to no agreement on standards***
 - Only one standard so far SMI-S in 10 years***

No standards for error correction or detection for each file

- In the file system nor for archives***
- PI addresses the packet issue but not the file***
 - PI is not supported on tape or SATA***

The Problem

I believe if applications cannot communicate down the data path then scaling, reliability, ILM are not possible

Agency after agency is reinventing the preservation archive wheel

- NSF, LOC, NARA, DoD HPCMP just to name a few***
 - User space applications without standard to manage collections***
 - These are impossible to port***
- Preservation is just one example of wheel re-invention***

SOME THOUGHTS

Here are some ideas that Gary and I have discussed that would improve things

Framework

Use POSIX extended attributes and define a common set of attributes that move with a file

- ❑ File systems will need to support the common set along with UNIX commands (ls, tar etc) and file transport (ftp, pNFS, rcp, etc) will need to support the framework***
- ❑ As a start, some ILM suggested attributes would address:***
 - ❑ Archive/backup***
 - ❑ OSD framework***
 - ❑ Data integrity (PI)***
 - ❑ Performance hints***

Current T10 OSD and PI need to be addressed as part of this

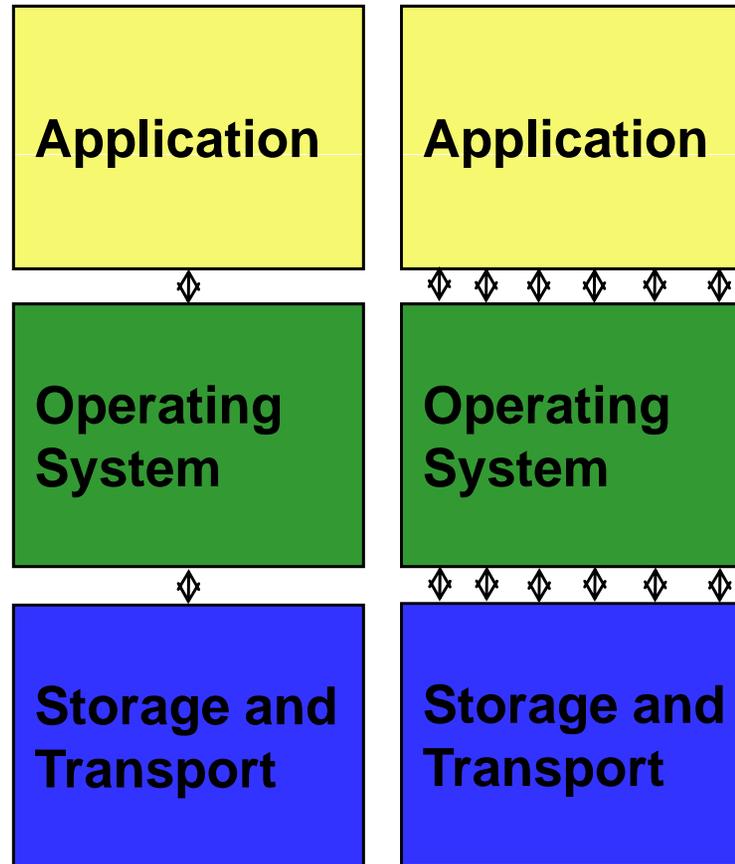
Datapath Changes are Needed

Current

Current POSIX system calls open/read/write/aio. Limited communication with OS layer

POSIX Atomic operations open/read/write/aio. No communication with physical layer

Block based storage and limitations of 30+ year old technology



Future Needs

ILM framework that applications can interface. Information is passed on through generations of systems. User searchable

ILM archival and backup policies based on per file information or defaults.

More intelligent storage to address issues such as fragmentation, encryption, OSD and reliability to name a few

Data path today is the same as the data path 20 years ago

Focus on end-to-end view of I/O in whole datapath. Moved from 3 separate boxes to 1. The benefits of this framework extend to the entire datapath, not just ILM.