Storage Lifecycle Management (SLM) End-user Training

NAVY August 2013

ARL September 2013

SLM End-user Training

- **Location: NAVY and ARL**
  - Tuesday, August 27 2013 through Thursday, August 29 at NAVY
  - Tuesday, September 10, 2013 through Thursday, September 12, 2013 at ARL
- **Presenters:**
  - Tino Scheder General Atomics
  - Lawrence A. Murakami HEUE SLM CEW
- **Training Materials:**
  - https://help.ccac.hpc.mil/heue
  - http://people.arsc.edu/~murakami/SLM

Not approved for public release

Agenda

- Introductions, Welcome, and Agenda Review (Tino)
- SLM overview: Why is it needed? What is it? Why are we here? (Lawrence)
- Scommand Introduction and Current vs. Future use of the Archive (Tino)
- Break
- Hands-on: The basics (Tino)
- Lunch
- Hands-on: Advanced (Lawrence)
- Break
- Introduction to workflow scripting solutions (Lawrence)
- Where we are today (Tino)
- Closing remarks and Questions (Lawrence)

SLM overview: Why is it needed? What is it? Why are we here?

Not approved for public release

SLM Overview: Why Is It Needed?

- **Old (current) archive paradigm**
  - Users archive with no limit
  - All files get a remote DR copy
  - We hope that they will delete stuff when no longer needed
  - The only expense for users is the time / effort to cleanup
  - The "good citizen" is penalized in time / effort
  - Net effect: archive has a 160% CAGR -> archive costs will swallow Program budget
- **New storage paradigm**
  - **Still no quota**
  - **Still no charge for storage used**
  - Remote DR copy attribute can be set
  - All users must spend some time / effort in setting retention attributes
  - We eliminate effort to delete files when it's time
- **We do require a modest effort in identifying how long objects should be retained**

Not approved for public release

SLM Overview: Why Is It Needed?

- **Archive is growing at an exponential rate**
- **Growth cannot be sustained within program funding constraints**
- **Users have a tendency to archive and forget**
  - Write once, read never

Not approved for public release

SLM Overview: Why Is It Needed?

- **As data ages it becomes difficult to remember what the data was and its importance**
  - Reluctance to delete old data because we don't know

- **As file count increases, it becomes more difficult for users to manage their data**
  - Often use convoluted file and directory naming schemes to identify data
  - When users leave, there is often nobody who can interpret the naming scheme
  - Only so much information can be encoded in file and directory names
  - Makes searching for data nearly impossible

SLM Overview: What Is It?

- **Storage Lifecycle Management (SLM) is a set of tools that enables researchers, developers, and other knowledge workers in collaborative groups to discover, share, and manage files through a unified interface**

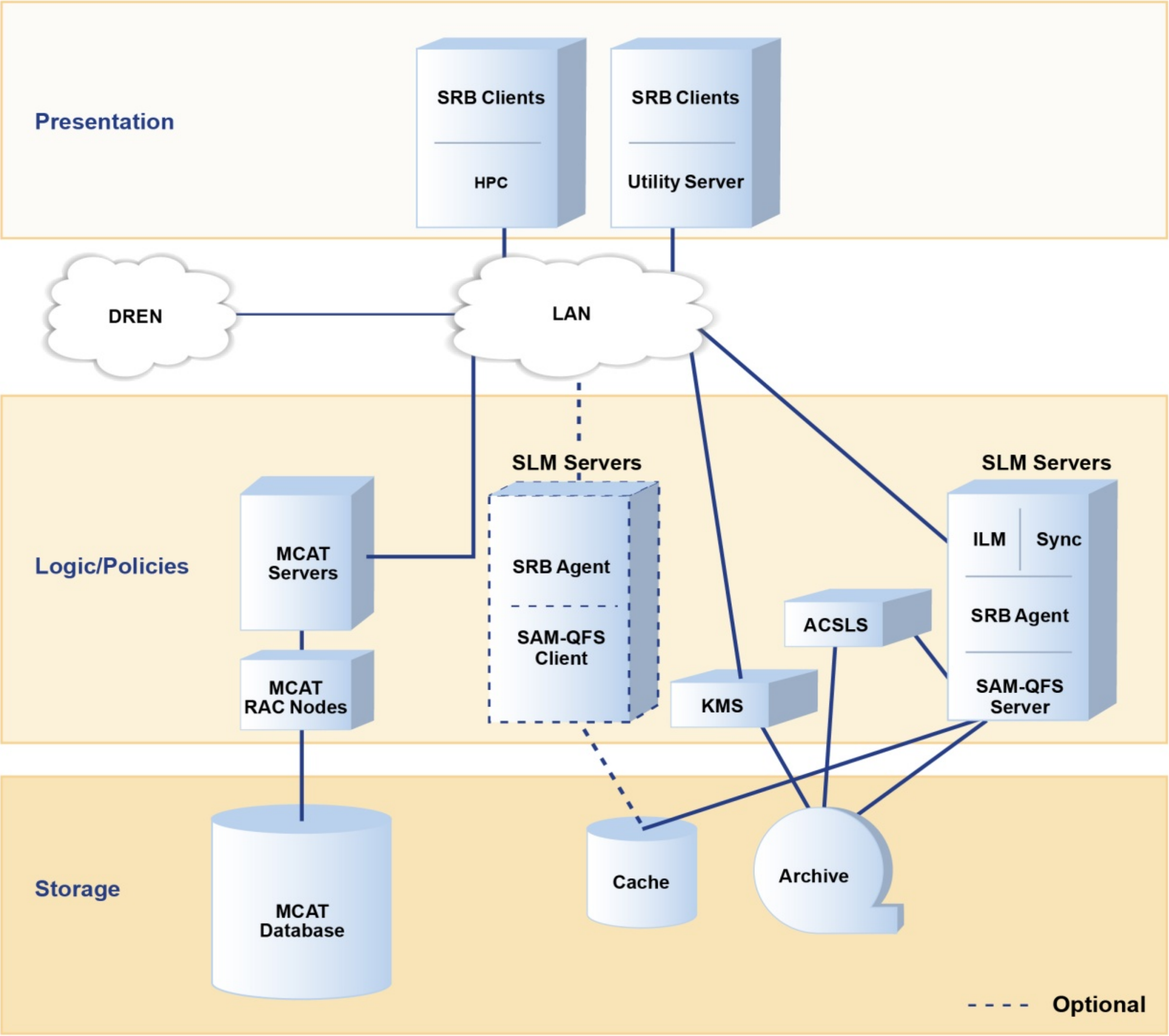
- **The unified interface makes use of a Metadata Catalog (MCAT) which enables the creation of a Global Namespace, allows for centralized administration, and provides discovery, security, auditing, and other services**

- **SLM uses Nirvana SRB and Oracle commercial software on a set of MCAT servers to form an Information Lifecycle Management (ILM) system interfacing with a Hierarchical Storage Management (HSM) back end. HSM is sometimes referred to as tiered storage**

SLM Overview: What Is It?

- **HPCMP DSRCs use the SAM-QFS software as the HSM back end.**
- **SLM provides the ability to associate user-supplied descriptive metadata with data files after the data has been registered into the application.**
  - Metadata tagging will help users effectively organize and search for archived data, and ensure that data is saved to the appropriate hardware resource for the appropriate amount of time
  - Critical data can be archived, while unnecessary or out-of-date data can be deleted instead of archived to tape and potentially forgotten, taking up costly long-term archive space

SLM Overview: What Is It?



Not approved for public release

SLM Overview: Why Are We Here?

- The goal of SLM - more intelligently manage data growth at the multi-petabyte scale to reduce the rate of storage cost growth
- At the end of this training you will be prepared to start using SLM.
- Once we have early adopters we can expand use of SLM to all users and meet our goal.

Not approved for public release

SLM Terminology

- **Files and directories**
  - Regular POSIX files and directories
  - What exists on the local file system
- **Data Objects**
  - SLM registered files
- **Collections**
  - Similar to directories, but within SLM, not the local file system
- **Retention time – HPCMP policy**
  - How long a data object will remain in SLM before being automatically deleted
- **Review period – HPCMP policy**
  - Amount of time a data object can be in SLM without being reviewed before it is automatically deleted
- **Notification period – HPCMP policy**
  - The time before a data object is to be deleted that the owner is notified, allowing the option of extending the life of the data object

What Are Scommands?

The command-line interface clients to SRB

- **"Sls” or “Sls -l“**
  - List the contents of your Collection
- **"Scd“**
  - Change the path in the SRB. Behaves similarly to the unix "cd" command
- **"Sput filename   ."**
  - Watch the period "." near the end of the Sput command
  - Copy the local filename to the current path IN SRB
- **"Scp dataname1 dataname2"**
  - This copy ONLY copies data objects from the SRB archive to the SRB archive
- **"Smv dataname1 dataname2"**
  - This moves a Data Object in Collection space (or renames it as shown here)
  - File bits do not move and files not re-named in underlying filesystems

What Are Scommands?

- **"Sget dataname   ."**
  - Copy a Data Object from SRB to the current unix path
- **"Smkcoll mycoll"**
  - Create a Collection named mycoll
- **"Srm"**
  - Remove a Data Object from SRB
- **“Schmod”**
  - Apply access control to a Data Object or Collection
- **“SgetD -a”**
  - Display access controls for a Data Object or Collection
- **“SgetD –x”**
  - Check archive status of a Data Object

What Are Scommands?

- **"Scat dataobject"**
  - List the contents of a text Data Object to stdout
- **“Spwd”**
  - Display current working Collection
- **"Senv"**
  - Displays information about SRB User's session or default session file
- **“Sscheme”**
  - Lists, adds, modifies, or deletes Metadata information about SRB Data Objects

What Are HPCMP Scommands?

- **Developed by Customer Experience Workgroup**
- **"Sretain“**
  - Set the date for a Data Object in the SLM archive to expire (be deleted)
- **“Sdata”**
  - Set/change/delete/display metadata for a Data Object in the SLM archive.
  - If Sdata above does not meet one's metadata needs, you can use Sscheme for more advanced metadata usage
- **“Sreview”**
  - Set the review date for a Data Object in the SLM archive.

SRB Documentation

- **User documentation for SRB is included with the client**
  - ${SRB_HOME}/doc {/opt/slm/current/doc on US)
  - Documentation is in pdf and html format
- **Every Scommand has man pages and the -help option**
- **The User Guide - User Tasks - Policy Management section includes tables that list available SRB Data Object and Collection Attributes, SRB Expressions, and available operators**
- **Documentation on the CCAC Web site at:**
  - https://help.ccac.hpc.mil/heue
- **Additional Documentation at:**
  - http://people.arsc.edu/~murakami/SLM

Not approved for public release

Using Metadata – SLM Schemes

- **SLM metadata will help you manage files for long-term use**
- **Admin Scheme**
  - Automatically created when the Data Object is placed into SLM
- **Dublin_Core Scheme**
  - Created when a user sets attribute values for a Data Object using either Sdata or Sscheme
- **Name_Value Scheme**
  - Created when a user sets attribute values for a Data Object using either Sdata or Sscheme
  - The "name" is arbitrary alphanumeric and the user can create multiple "name" = "value" pairs

Using Metadata – SLM Schemes

- **Users Scheme**
  - Created when a user sets attribute values for a Data Object using Sscheme
  - This scheme allows a high character count, arbitrary value designed by the user
  - The value can be data in XML format
- **HSM Scheme**
  - Created & maintained automatically by the Ssync daemon
  - Has attributes related to the disc cache copy including the inode
- **HSM_Copy Scheme**
  - Created & maintained automatically by the Ssync daemon
  - Has attributes related to the Tape and DR copies including Copy_Number and Media_Type
- **TOC Scheme**
  - Table of Contents attributes for individual members of container files

Not approved for public release

Admin Scheme Attributes

- **Archive_Behavior** **yes (default), no**
  - Is the Data Object to be archived? User modifiable.
- **DR_Behavior** **yes, no (default)**
  - Is a DR copy to be made for the Data Object? User modifiable.
- **Purge_Behavior** **yes, no (default)**
  - Is the Data Object immediately released after archival? User modifiable.
- **Retention_Period** **ordinal-valued days**
  - Identifies the number of days past the Data Object create date the Data Object is retained in the archive. User modifiable.
  - Default 30 days
- **Retain_Time**
  - = object creation time + Retention_Period
  - Date-time until which Data Object will be retained

Not approved for public release

Admin Scheme Attributes

- **Last_Review_Time**
  - = attribute value modification date
  - Current date-time automatically updated on scheme modify
- **Next_Review_Time**
  - = least(Last Review + Review Period, Retain_Time)
  - Next date-time when archive policy of the Data Object is to be reviewed by the user
- **Admin_Hold yes, no (default)**
  - Is the Data Object prevented from expiration and purge?
- **Warning_Note yes (default), no**
  - Does the user receive a notification at the Warning_Period_User before Data Object expiration? User modifiable.
- **HPCMP_Project_ID**
  - 13 character HPCMP project code
  - Inherited from parent Collection if available else null-valued
  - Future – may be required for > 30 day retention so best practice Fill It In!

Not approved for public release

Dublin_Core (Adapted) Scheme Attributes

- **Creator (string[64])**
  - Document creator
- **Create_Date (timestamp)**
  - Date document was created (default: current time)
- **Type (string[32])**
  - Type of document
- **Contributor (string[128])**
  - Entity providing contributions to the document
- **Document_ID (integer)**
  - Document Identifier – creator provided
- **Publisher (string[64])**
  - Person, organization, or service. Used to indicate the entity.

Dublin_Core (Adapted) Scheme Attributes

- **Description (string[128])**
  - Description of the document
- **Subject (string[64])**
  - Document subject (keywords, key phrases or classification codes)
- **Title (string[64])**
  - Document Title
- **Rights (string[128])**
  - Statement about property rights

Name_Value Scheme

- **The Name_Value Scheme allows for an alternate, flexible, metadata entry**
- **The Name_Value Scheme permits multiple rows of metadata attributes per Data Object or Collection**
  - Multi-row schemes in SRB automatically maintain a row_id sequence number to keep track of the rows
- **Name (string[32])**
  - User-specified attribute name
  - Best practice use only alphanumeric with leading alpha
  - Punctuation & special chars will create future aggravation for you when searching!
- **Value (string[256])**
  - User-specified attribute value

Metadata Entry

- Two native Scommands in SRB to manipulate metadata on files and directories – Sscheme and SmodD –S
- The HPCMP has provided its users an easier, more tailored mechanism to perform metadata entries and manipulations using the Sretain, Sreview and Sdata commands
- The Sretain, Sreview and Sdata commands developed by the HPCMP are described in detail by man pages and via command help

Not approved for public release

Metadata Entry - Sretain

- **Sretain is the command line interface to set a Retain_Time on Storage Resource Broker (SRB) objects**

- **The Sretain command updates the 'Admin' Scheme which is a System Scheme that is automatically applied during ingestion of objects into SRB**

- **The Sretain command requires a days or date argument for the Retain_Time date and Data Objects or Collections to act upon**

- **The Last_Review_Time is set automatically by the Sretain command**

- **Sretain can also set HPCMP_Project_ID, DR Copy and other 'Admin' Scheme attributes.**

Metadata Entry - Sreview

- **Sreview is the command line interface to set a Last_Review_Time on Storage Resource Broker (SRB) objects.**

- **The Sreview command updates the 'Admin' Scheme which is a System Scheme that is automatically applied during ingestion of objects into SRB.**

- **The Sreview command requires only Data Objects or Collections to act upon as an argument.**

- **The Last_Review_Time is set to the current date by the Sreview command.**

- **The -R option allows the Last_Review_Time to be set recursively for all Data Objects within a Collection or Collections.**

Metadata Entry - Sdata

- Sdata allows one to display, set, change, or delete keyword-value pairs or the project in the SRB metadata
- Sdata accepts a keyword=value syntax and the scheme name is not required
- For the Title, Creator, Subject, Description, Publisher, Contributor, Creation Date, Type, Document ID, and Rights keywords metadata values will be stored in the Dublin_Core scheme
- The values for all other keywords will be stored in the Name_Value scheme

How Do I Get Started?

- **For csh/tcsh: source $SAMPLES_HOME/slm/init.csh**
- **For sh/ksh/bash (watch the period, ".", at the beginning of the line!)**

  ```
  . $SAMPLES_HOME/slm/init.sh
  ```

- **You should see output including something similar to:**

```
Welcome to SRB
Your User Name is “yourname@HPCMP.HPC.MIL"
Your Default Classification is “SBU"
Your Default Resource is “arl.msas14.srb_test"
Your Home Collection is "/archive/username/arl/srb_test"
```

Hands-on Scommand Practice

Initialize into SLM,
Create a Collection,
Set retention period

Not approved for public release

File Renewal Notification

- **Today via Virtual Collections only!**
- **Users can receive notification for files that are subject to removal**
  - All Data Objects registered or ingested into the SLM system will automatically obtain "Admin" scheme attributes according to user defaults, administrator defaults, or parent Collection
- **Users are free to change many attributes to, for example, extend the Retention_Period of a set of files**
  - Users can change the Last_Review_Time without making changes to the Retention_Period or other attributes of the Admin scheme that affect archive and retention of the Data Object
  - **Changing of the Last_Review_Time represents a review**
  - Default Retention_Period is 30 days.
  - Warning_Period_User is 28 days.
  - Warning_Period_PI is 15 days.
  - Review_Period 3 years.
  - Warning_Note default is yes.

Not approved for public release

Query Conditions (Policy Queries)

- **One of the advantages of a database-resident file system such as SRB is the ability to perform queries against the database using all the metadata attributes associated with the Data Objects and Collections in the file system.**
- **There are generally two mechanisms to perform such queries in SRB**
  - SgetD or Sls (or other Scommand) –policy <query>
  - Virtual Collections
- **Both mechanisms utilize SRB's pseudo SQL query language**

Query Examples

- **List Data Objects that have a Retain_Time less than 31 days in the future**

  -
    ```
    Sls -R -policy \
    "(EXPRESSION.current_timestamp > Admin.Retain_Time - '31') \
     AND (DATA_OBJECT.data_type not like '*collection')” <collection>
    ```

- **List objects in order of Retain_Time with create time**

  -
    ```
    Sls -R -policy \
    "(DATA_OBJECT.data_type not like '*collection')” \
    -order "Admin.Retain_Time" -select \
    "DATA_OBJECT.data_name,DATA_OBJECT.create_timestamp,\
    Admin.Retain_Time,Admin.Retention_Period” <collection>
    ```

- **List objects for a specific project**

  -
    ```
    Sls -R -policy \
    "(Admin.HPCMP_Project_ID = <Project ID>) \
    and (DATA_OBJECT.data_type not like '*collection')” <collection>
    ```

SLM Global Namespace

- **The global namespace includes a fully qualified path for all Data Objects known to SLM**
  - The namespace conventions have been designed to allow for all Data Objects on the HPCMP archive filesystems to be ingested or registered into SLM with no name collisions
  - The user namespace provides a tree view of all of a user's Data Objects known to SRB on all resources
  - User Namespace
    - /archive/<srb_user>/<site>/<filesystem> an ‘Slink’ pointing to…
  - Resource Namespace
    - /<site>/<filesystem>/<user>

SLM Global Namespace

- **The Home Collection as known to SLM is not writable by the user**
  - Prevents future name collisions when the namespace of the individual DSRCs are joined in a federated system
  - Cannot Sput:
    - /archive/
    - /archive/<srb_user>/
    - /archive/<srb_user>/<site>/
    - /<site>/
    - /<site>/<filesystem>/
  - Can Sput:
    - /archive/<srb_user>/<site>/<filesystem>/ because pointer to…
    - **/<site>/<filesystem>/<user>**

Not approved for public release

SRB Users And SRB Groups

- SRB Users and SRB groups do not map directly to POSIX users and groups
- User identity in SRB is the HPCMP Kerberos principal
  - username@HPCMP.HPC.MIL
- SRB user name need not match the login name on a host
- SRB groups represent common roles or data access needs that tie multiple users together
- They can be used for projects, responsibilities or access groups

SRB Access Control

- SRB access control is done via Access Control Lists (ACLs)
- Default is all objects to inherit the ACL of the parent Collection
- SRB account provisioning makes the top level Collection where a user can write have an ACL for that user
- If a user does nothing to change access control all Data Objects within that Collection and any child Collections will be controlled by that single ACL
- A user can make any Collection be it's own ACL inheritance object or use any other object as it's ACL inheritance object

SRB Access Control

- **Best practice granularity is trees of access control**
  - Set the ACL inheritance object of a top level Collection of a tree of objects that will have the same access
  - ACLs affect performance so best practice is use only at Collection level and not on each Data Object
- **If Data Objects already exist in a tree this can be set recursively**
- **Any new Data Objects ingested anywhere in that tree will inherit the ACL object from the parent Collection**
- **Multiple trees of Data Objects can inherit the same ACL object**
- **Only one ACL object is required for each unique set of access requirements**

SRB Access Control

- **An ACL object can have access granted for individual users or for groups**
- **ACL constraints include read, write, all and deny**
  - The deny constraint can be used along with groups to allow access to all group members except those that have the deny constraint specified
- **If a user is the owner of a Data Object, then he/she automatically gets "all" access**
  - If the user is not the owner, then the access they get depends on the ACLs associated with the object that match the user
- **If a user is going to allow Data Objects of other users in a Collection they own, they should grant themselves explicit access to automatically have access to Data Objects other users put in that Collection**

Not approved for public release

SRB Access Control

- **Summary of the access in SRB:**
  - read - Read-only of the Data Object contents and metadata
  - write - Write, truncate, or modify the Data Object and metadata, Add objects
  - all - Delete, give access, change ownership the Data Object
  - deny - No access to he Data Object contents or metadata

- **Notable differences from POSIX**
  - You cannot Srm an object you have write access to in SRB.
  - 'write' access allows for overwriting of contents and metadata.
  - 'all' access is required to delete an SRB Data Object.

Mandatory Access Controls (MAC)

- **SRB provides for Mandatory Access Controls (MAC) that can be implanted with levels and flags**
- **The configuration for HPCMP unclassified systems uses only the SBU and ORS levels with SBU being the higher level**
- **A user and the SRB storage Resource must have the MAC level of the object or higher**
- **Use of the SRB flag MAC controls are not included in the initial configuration for HPCMP unclassified systems**
  - A possible flag would be ITAR
- **A user must have the same flag as any object marked with a flag**

Current vs. Future Use of the Archive

Future Use Of The Archive

- Current archive access methods go away and will be replaced by Scommands
- Changes at the command line should be self-evident

ARL & NAVY Transfer Use Cases

- ARL uses NFS mounts on HPC Login nodes for archive access.
- NAVY allows direct login to archive servers via HPCMP Kerberos-aware commands such as krcp, kftp, krsh, krlogin, and ssh
- NAVY allows use of the non-Kerberized 'rcp' and 'rsh'commands from HPC login nodes and transfer queues
- NAVY allows rcp access to archive servers from HPC resources including compute nodes on some systems
- ARL and NAVY systems do NOT support the ‘archive’ command
- NAVY systems do NOT mount archive file-systems on HPC login or compute nodes
- With SLM all DSRCs will have common archive access

Not approved for public release

Transfer To/From Archive Systems

- **Use SLM (Sput to Archive, Sget from Archive)**
  - HPC ${WORKDIR} <-> Archive
  - US ${WORKDIR} <-> Archive
  - CWFS <-> Archive

- **Batch**
  - Use HPC Transfer Queue for transfers involving HPC ${WORKDIR}
  - Use any US node for transfers involving US ${WORKDIR}
  - Use HPC or US Transfer Queue for transfers involving CWFS.
    - siauth used to get STORAGE realm credentials
    - SLM to access archive

Not approved for public release

Transfer To/From A Host Not An HPC Or US Login Node

- Current workflows may include direct transfer to and from archive from a machine that is not a HPC or US login node
- No initial SLM support planned on non-HPCMP assets

Transfer To/From Archive Systems Initiated By A cron Job

- **NAVY and ARL may approve cron access on a per case basis**
- **A cron job can submit a PBS transfer queue job that can authenticate to the STORAGE realm and use SLM commands (Sput to Archive, Sget from Archive)**
- **STORAGE realm credentials are obtained using the siauth command**
- **The siauth command will be configured to only allow it’s use within a PBS job**

Future Use In A Job Script

- **#PBS –l slmstagein**
  - List archived data to be retrieved and location
  - Creates a transfer queue job
  - PBS estimates when you job will start and how long the transfer will take, not starting the transfer until it needs to
- **#PBS –l slmstageout**
  - List files to be archived and where to archive them
  - Creates transfer queue job
- **#PBS –l stinscr=/u/work/…../scriptname.ksh**
  - The script must manage STORAGE realm
  - Can make any SLM calls
  - Can check for file availability to decide if data needs to be retrieved from archive
- **#PBS –l stoutscr=/u/work/……/scriptname.ksh**
  - The script must manage STORAGE realm
  - Can make any SLM calls

Not approved for public release

Use Case - Ingest and Set Retention

Problem: A Geologist wishes to automate archive of simulation run output artifacts to SLM and make DR copies of certain files

- Step 1: Create a to be PBS transfer queue script
- Step 2: Evaluate Target collection and source directory, error check
- Step 3: Run Sretain on target Collection to set default retention period, HPCMP Project, No DR copy
- Step 3: Run Sput from source directory to target Collection
- Step 4: Verify return status
- Step 5: Evaluate fileset existence before replication
- Step 6: Run Sscheme on subset of fileset to set DR attribute.
- Step 7: Exit and report success/failure

Not approved for public release

Use Case - Post Processing

Problem: A Mathematician modeling scientist wishes to run post processing computation using intermediate output file from his last three runs

- Step 1: Evaluate input Target collection and input argument
- Step 2: Gather three intermediate output file sets from SLM using Sget command
- Step 3: Construct a post-processing PBS run file against the three file sets
- Step 4: Submit PBS job
- Step 5: Print success/failure report

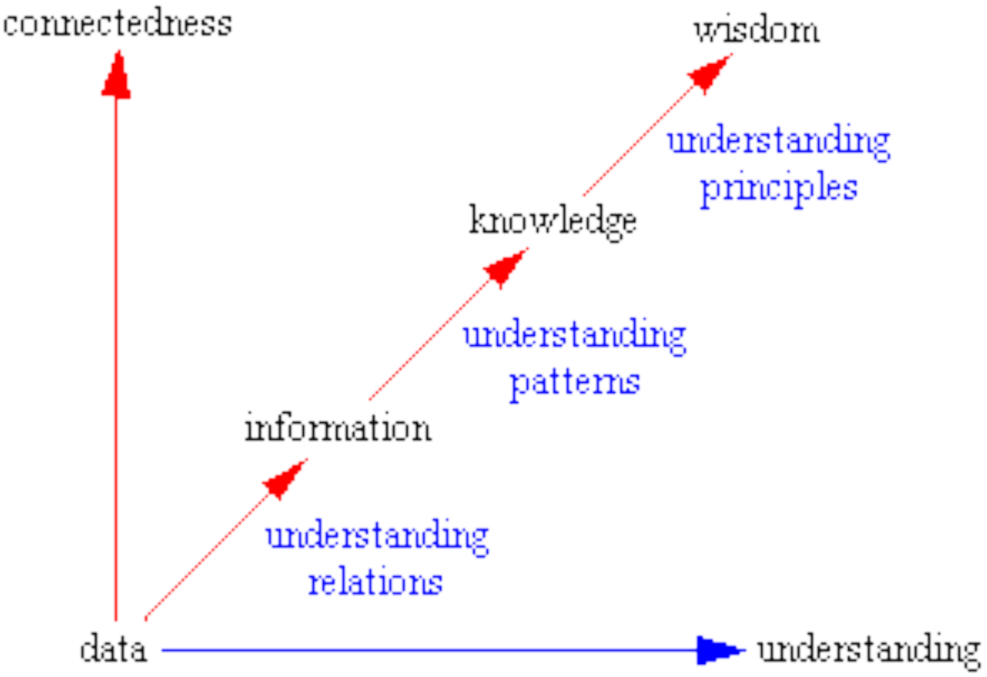
Not approved for public release

Hands-on Scommand Practice

Scientific Metadata

A Successful Information Management Should Facilitate the Transition to Wisdom

- **Data**
  - Raw, in and of itself
- **Information**
  - Processed data
  - Relational connections
  - Who, what, where, when
- **Knowledge**
  - Applying data and information
  - How
- **Understanding**
  - Appreciation of why
  - Cognitive and analytical
- **Wisdom**
  - Evaluated understanding
  - Deals with the future



G. Bellinger

R.L. Ackoff, "From Data to Wisdom,"
Journal of Applies Systems Analysis, Volume 16, 1989 p 3-9.

Evidence Suggests that Much of the Data From Experiments and Simulations Remains Untouched

- **For files: 66% re-opened once, 95% fewer than five times***
  - As file systems grow in size, just too much to sift through
- **Provenance**
  - Lineage of data products
  - Workflow: data objects, parameters of each process step
- **Metadata**
  - Information about each process step
  - A set of data that describes and gives information about other data
- **Ontology**
  - Structure that captures common terms to describe object properties
  - Controlled-vocabulary or classification structures

*Andrew Leung, et al., "Measurement and Analysis of Large-Scale Network File System Workloads," Proceedings of the 2008 USENIX Annual Technical Conference, Boston, Massachusetts, June 2008.

Scientific Metadata Can Accelerate Science Discovery

- **Metadata has accelerated understanding of large data repositories**
  - Rapid browsing, search, and discovery
  - Facilitates data comparison: both experiments and simulations
  - Creates a rich knowledge base of the science for newcomers
- **Metadata and provenance tracking should be embedded in workflows**
  - Automated as much as possible to not burden the scientist
- **Metadata can go beyond simple scalars**
  - Includes text for comments or an electronic logbook
  - Dates & times for historical logging
- **Able to be used in a variety of UIs**
  - Web browsers, scripts, interactive graphics, custom code, etc

SLM Provides For User Metadata

- **SLM MCAT is very powerful**
  - This power can be used for accelerating science
- **One goal of the Navy Pilot is to ID several "metadata interested" scientists**
  - Do one or two proof of concepts to show value
  - Requires scientist to work with us to create something of value
- **For your science, where can metadata add value?**
  - Are there workflow bottlenecks that can be alleviated?
  - Are there things desired to be done that can not be done now that metadata might help?
- **A variety of avenues can be investigated**
  - Web portals, custom GUIs (IDL, Python, etc.), scripts
  - Our thinking is to not require scientists to know SLM specific commands

Simultaneous POSIX and SLM Access Concerns

- GA strongly recommends against this
- POSIX "mv file new_file" could result in metadata loss
  - including Retention_Period / Retain_Time
  - could result in file deletion
- Smv is a metadata-only operation
- SLM appends characters to the POSIX object name it creates for Sput
- Early Adopters should avoid this
  - Separate trees for SLM and POSIX access

SLM dual path to production

- **Path 1 - SLM is moving to production behind the scenes.**
  - ARL is already there with files on the army, armya, armyb, armyc, navy, navya, navyb, navyc, quota, airforce, and others archive file systems being handled BY SLM behind the scenes. A NAVY rebuild of the SLM database and ingestion of all files in the b and g archive file systems is next and ERDC and AFRL following within months.
- This change is transparent to the users but allows some benefits of SLM to be implemented including having metadata for all files in the catalog.
- **Path 2 - Early adopters will use the SLM client and modify workflows until all archive access is through the SLM client.**
  - Client roll-out to all users will follow the early adopter phase.

Not approved for public release

Closing Remarks and Questions