The **Purge Threat**: Scientists' thoughts on peta-scale usability

Alexandra Holloway <fire@soe.ucsc.edu>
Storage Systems Research Center + Assistive Technology Lab
University of California, Santa Cruz

Introduction

- Usability problems, including mediating the threat of data loss when parallel file system fills up
  - The **Purge Threat**
- Discussion of a usability problem
  - Interview data
  - Not a solution

Research questions

- **RQ1**. How do participants interact with the file system currently?
- **RQ2**. What are the biggest usability problems concerning the peta-scale file system?
- **RQ3**. How do scientists address the major usability concerns?

Participants

- Los Alamos National Lab:
  13 participants (10 groups)
- Lawrence Livermore National Laboratory: 4 participants
- Developers: 2
  Users: 11
  Mixed roles: 2
  Other roles: 2
- Men: 16
  Women: 1

Pseudonym	Org.	Role
Aaron	LANL	Developer
Bruce	LANL	Developer
Charlie	LANL	User
Donald	LANL	User
Erin's team	LANL	User Team
Farhad	Affiliate	Researcher
Grisham	LANL	User
Harry	LANL	User
Ian	LANL	User
Jake	LANL	User, Developer
Kelsey	LLNL	Consultant
Leslie	LLNL	User
Mark	LLNL	User
Nate	LLNL	User, Developer

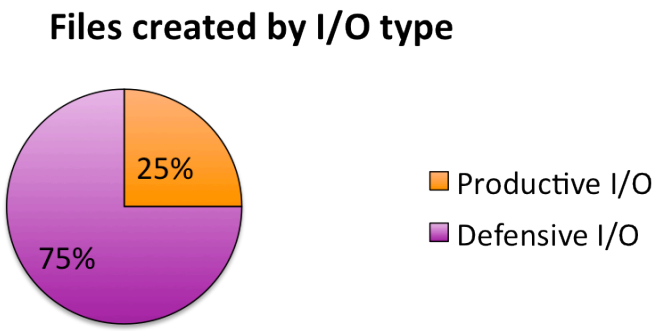
System

- Parallel system
- NFS
- Local machine
- Archival storage (tape)

The problem

- Scientists generate potentially thousands of files per job

Where do files come from?



Visualization dump size: 1—10% × restart files

- Productive I/O
  - Data the user needs to perform analyses and draw conclusions
  - E.g., Visualization dumps
- Defensive I/O
  - Data the user needs to show proof that results were obtained deterministically
  - E.g., Restart files, time histories, parallel output data

What happens to all these files?

- File system fills up

The Purge Threat

- Least recently accessed files scheduled for deletion
- List of affected files published
- Affected users must decide:
  - Archive
  - Delete (or allow deletion)
- **Purge threat** is the threat of data loss

Ideal file life cycle

1. Run simulation or job, creating 10000+ files.

1

Ideal file life cycle

2. Import select results for processing and visualization.

1

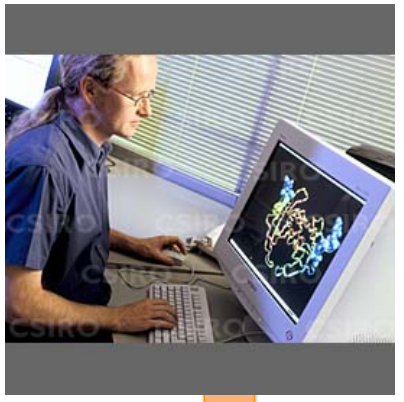
2

Ideal file life cycle

3. Think about which data are important to save.

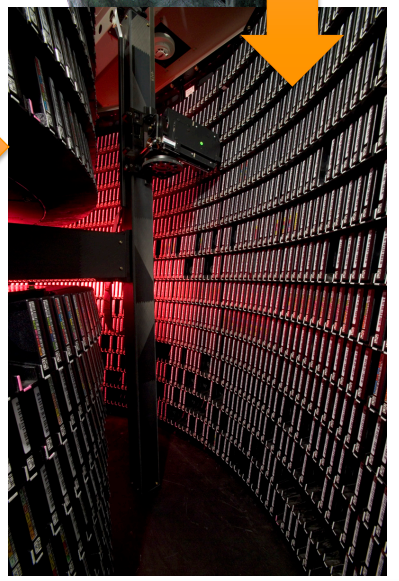
Ideal file life cycle

4. Archive important data.



4

4



Ideal file life cycle

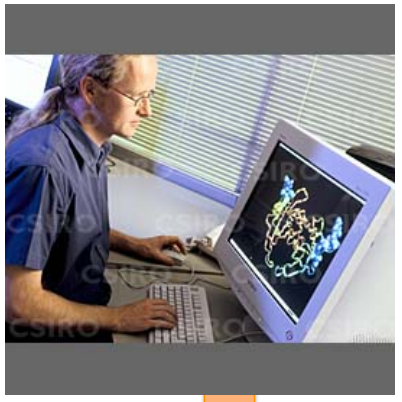
- Ideal file life cycle only happened 1 in 17 participants
- What did the other 16 do?

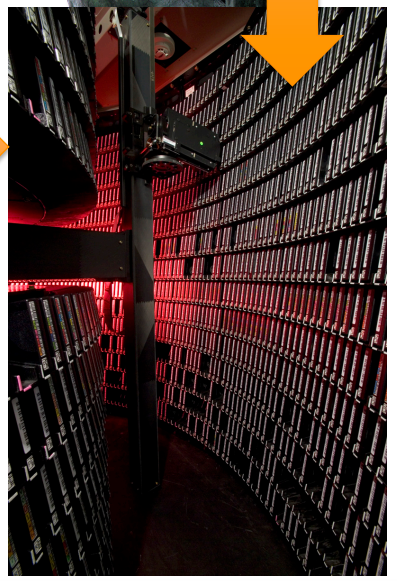
Addressing the purge threat

- Three ways to address the purge threat:
  1. Analysis
  2. Automation
  3. Subversion
- Interestingly, nobody named:
  4. Do nothing and let files perish

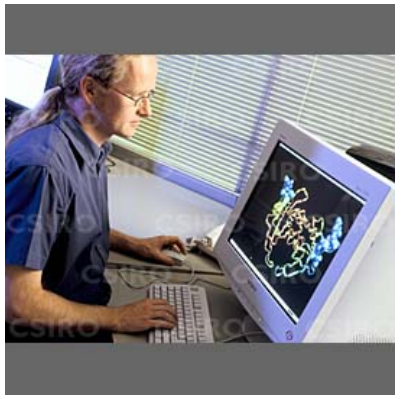
Analysis

- Think about affected files and move them to tape manually.
- (The ideal file life cycle)

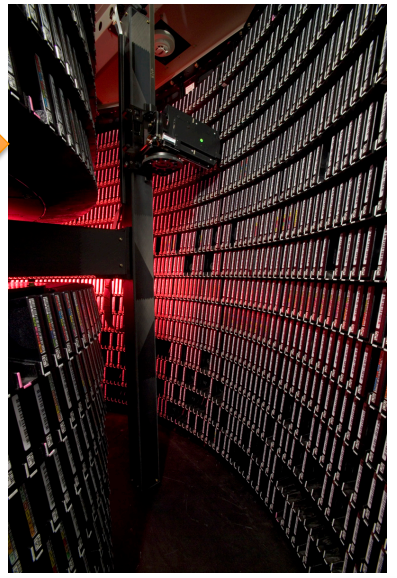




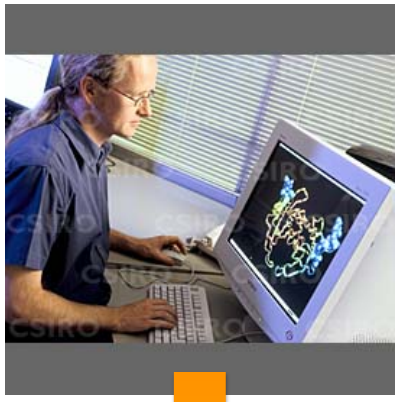
Automation



- Write a script to move all affected files automatically.



Subversion

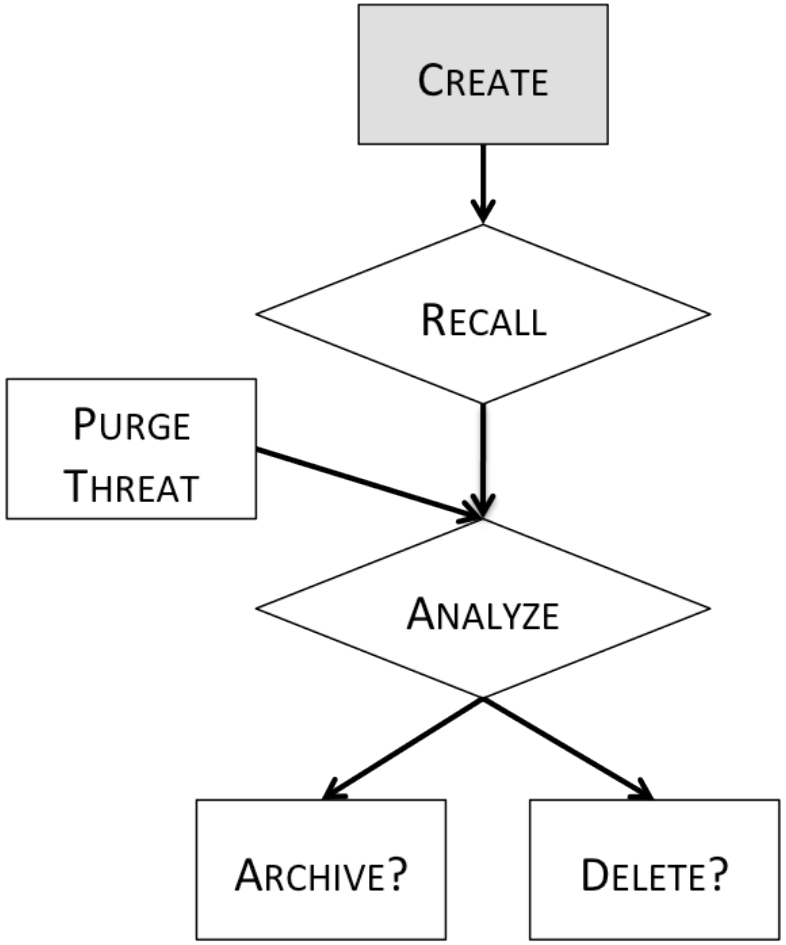


- Refresh the access date on files using `touch`.

Reasons to keep data

- Parallel file system is not backed up
  - Save data in case of a system crash
- Save all data that led to a decision
  - Reproduce deterministically even years later

Purge threat in the work flow



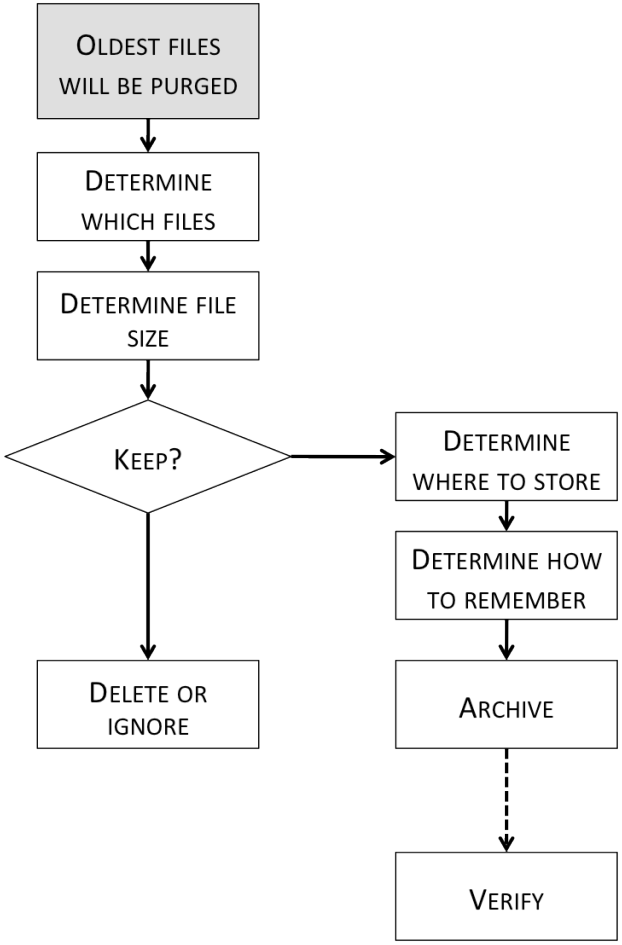
Two archiving methods

- Cautionary archiving
  - Protect against unanticipated data loss (e.g., crash)
- Reactionary archiving
  - Protect against purge threat and scheduled purge

Why not just archive everything?

- Archiving is “real money in tapes.”
- 90% of archive is never read – “Write Once, Read Never.”
- Retrieval is painstakingly slow.
- Archiving has huge cognitive load.

Deciding to archive



What happens next?

The next generation [of scale] may be the breaking point from "barely doable" to "what do we do next?"

Usability problems

- User must retrieve the list
- User may not understand seriousness
- User may not understand scope

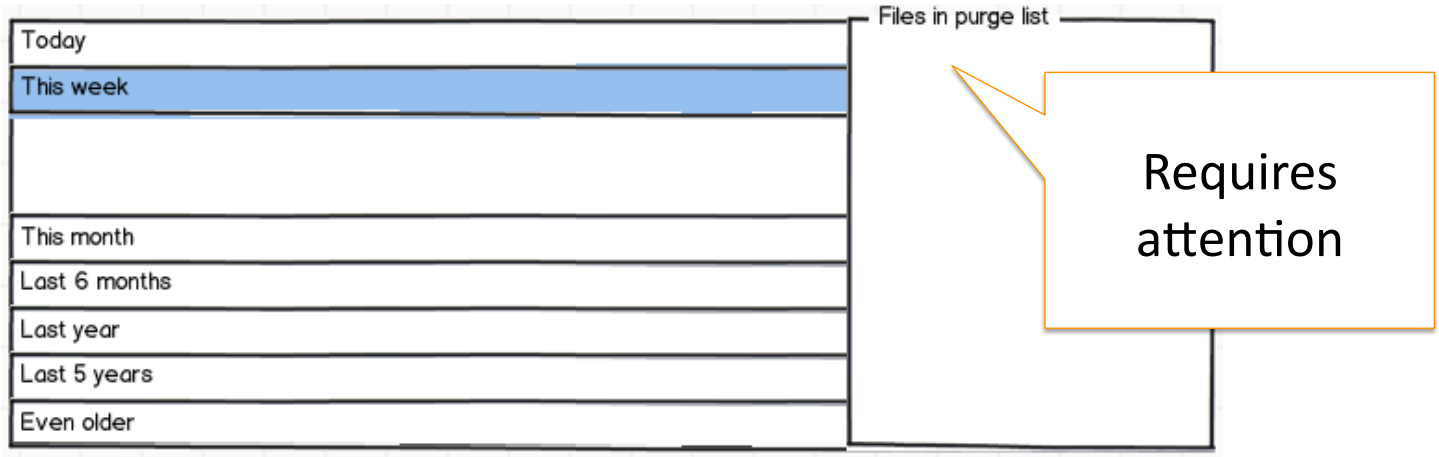
Proposed solutions

- Bottleneck is walking the directory structure
- Time-oriented file representation
- Space-oriented file representation

Time-oriented file representation

- Files in last-accessed chronological order
- Appropriate granularity
  - `dump.1`, `dump.2`, *etc*. represented as `dump.[1–256]`
- Threatened files listed

Time-oriented file representation




```
[user@sys %] lst --week
Accessed this week:
project1/vars/dump.[1-256]
project1/vars/restart.time[112988-98]
```

Space-oriented file representation

- Removing the largest size may mediate the purge threat
- How far down the directory structure is the first file of a particular size?

Research questions

- **RQ1**. How do participants interact with the file system currently?
  - Command line
- **RQ2**. What are the biggest usability problems concerning the peta-scale file system?
  - Decision-making and usability surrounding purge
- **RQ3**. How do scientists address the major usability concerns?
  - Analysis, automation, and subversion

Conclusions

- Purge threat
- Addressing the purge threat does not meet usability demands
- Decision-making paradigms surrounding archiving: reactionary and cautionary
- Three reasons for poor usability
- Proposed interfaces

Questions?

- The **Purge Threat**: Scientists' thoughts on peta-scale usability

Alexandra Holloway <fire@soe.ucsc.edu>
Storage Systems Research Center + Assistive Technology Lab
University of California, Santa Cruz