Death of Data: workflows for users involving DMF

Robert C. Bell | CSIRO IMT Scientific Computing

DMFUG 30 August 2019

Outline

- Problem of shared filesystems
- Some issues
- Some solutions
  - Dataflows, not data stores
  - Use scratch more – scalable flushing
  - Use /data less
  - Collections
  - Provide utilities to manage data flow
  - Users ask, “Where is the best place to store my data?”

Problem of shared filesystems

- Users want
  - every file kept and backed-up
  - only one file system
  - globally visible across all the systems they use
  - high-performance everywhere
  - infinite capacity
  - zero cost
  - utterly reliable
  - always available
  - low latency
  - no HSM!
- There are many reasons why this is not possible: principally financial, but there are the constraints caused by worsening ratios of bandwidths to capacity, the trade-offs between wide accessibility and performance (e.g. caching decisions), and the rapidly growing demands for capacity.
- We provide a range of filesystems, each meeting some of the needs.

Problem of shared filesystems

- For users
  - Home: too small
  - Scratch: too dangerous
  - Data: too full
  - HSM: too hard
  - Cloud: too new
- For system managers
  - Home: backup requirements
  - Scratch: too many files to flush
  - Data: too stagnant
  - HSM: too hard? Takes careful management
  - Cloud: too hungry

Some example issues

- NCI
  - /short not flushed for 5 years
    - can't get enough campaign space (436 M files on 31 Jan 2019)
  - /g/data areas – allocations locked up
  - Massdata – too small, perceived to be hard to use:

    'mdss put ...' of two or more files will fail if the average file size is smaller than 1024Kb.
- Pawsey
  - Was struggling to flush /scratch
    - large number of files (787 M files on 22-05-2018)
  - Limited access to HSM storage without extensive project proposals
- CSIRO SC
  - overload on shared areas led to stalls
  - 2018 'incident' resulted in the large-scale removal of contents of /flush* areas
  - no longer giving access to /data area for new users
  - Need to keep copies in CSIRO of data at partner sites?

/data and project area issues

- Risk – little protection from loss – no backup or HSM
  - Leads to complacency
  - No user perception of consequences of loss
- Hard to manage usage and get space for new projects
  - used to do “name and shame” – peer pressure => bullying
- Hard to recover allocations, except project end or user cessation
  - difficult even then to get either action to copy the data to safe keeping, or someone to take responsibility to delete data in the era of “keep everything”

Some solutions

- Dataflows, not data stores
  - data is a journey, not a destination
  - Data Lifecycle Management
- Use scratch more, with big quotas
  - scalable flushing makes this feasible
- Use /data less: use scratch and/or HSM-enabled
  - no support for un-managed static data
- Collections
  - Enables scaling
  - CSIRO was adding 5 M files per day to science data stores!
- Utilities to manage data flow

Utilities to manage data flow

- Initial scope:
  - Allow data collections to be maintained on scratch storage
  - Allow protection from loss by maintaining reference copy in HSM or archive
    - Consolidation to get scalability
  - Allow optimised re-syncing in the face of flushing
  - Allow updating
  - Replace the use of /data and /g/data* filesystems
- Can also be used for dealing with departed users, or users who have filled Home (on SC clusters), or to move data from one flush area to another

Capabilities from prototype utility

- In a working area in /flush* or /short, commands are:

```
mirror.sh help
mirror.sh man
mirror.sh create
mirror.sh list
mirror.sh update
mirror.sh sync
mirror.sh moveto new_area ; cd new_area
mirror.sh flush
mirror.sh release
mirror.sh recall
…  17 more operations (delete, cleanse, check, status, kill,
explain, removetmp, dev_history, restore, auditw, auditm, verify,
config, getremote, putremote, discover, rebuild)
```

- Available on SC systems bracewell, pearcey and ruby, and NCI raijin
  - ~bel107/bin/mirror.sh ~rcb599/bin/mirror.sh
- Works with a mirror in $STOREDIR or /g/data* & mdss (at NCI) (configurable)

Examples from prototype utility

- 1) Create a mirror

```
cd a_working_directory
mirror.sh create      # W->M..T
```

- 2) List the location and contents of the created mirror

```
mirror.sh list
```

- 3) After creation, minimise the on-line storage held by the mirror

```
mirror.sh release     # M..T
```

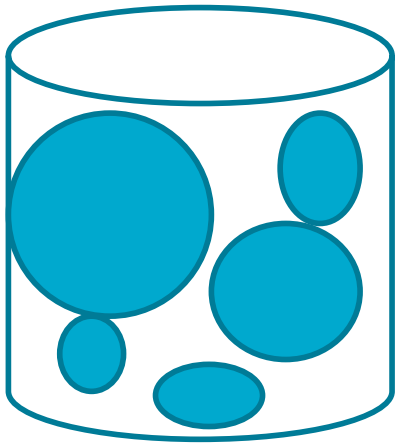
- 4) After creation, minimise the storage held in the working area

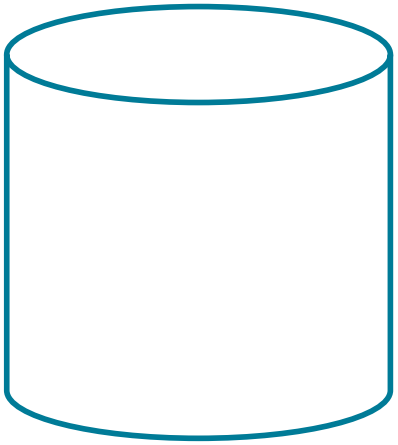
```
mirror.sh flush       # xW
```

Create a mirror

- 1) Create a mirror

```
cd a_working_directory
mirror.sh create      # W->M..T
```



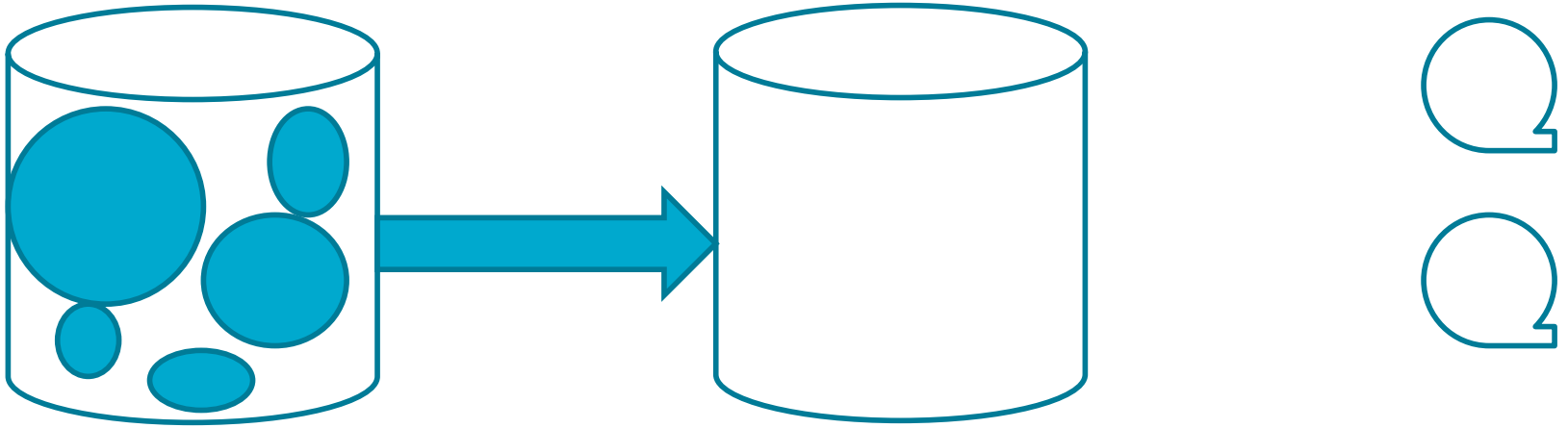


Working directory Mirror HSM tape

Create a mirror

- 1) Create a mirror

```
cd a_working_directory
mirror.sh create      # W->M..T
```

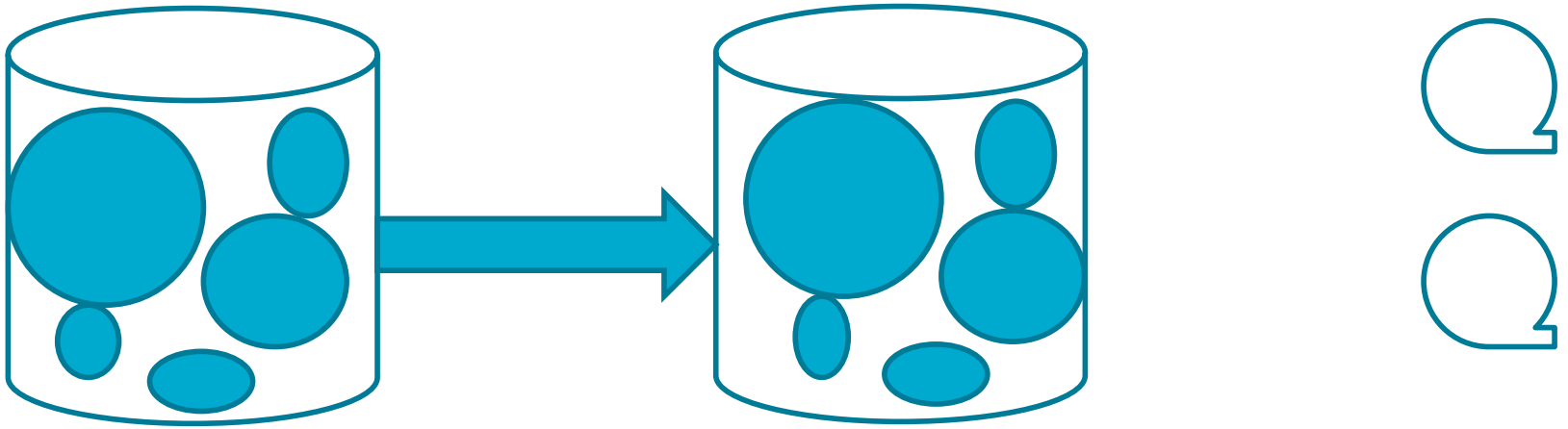


- rsync, or course

Create a mirror

- 1) Create a mirror

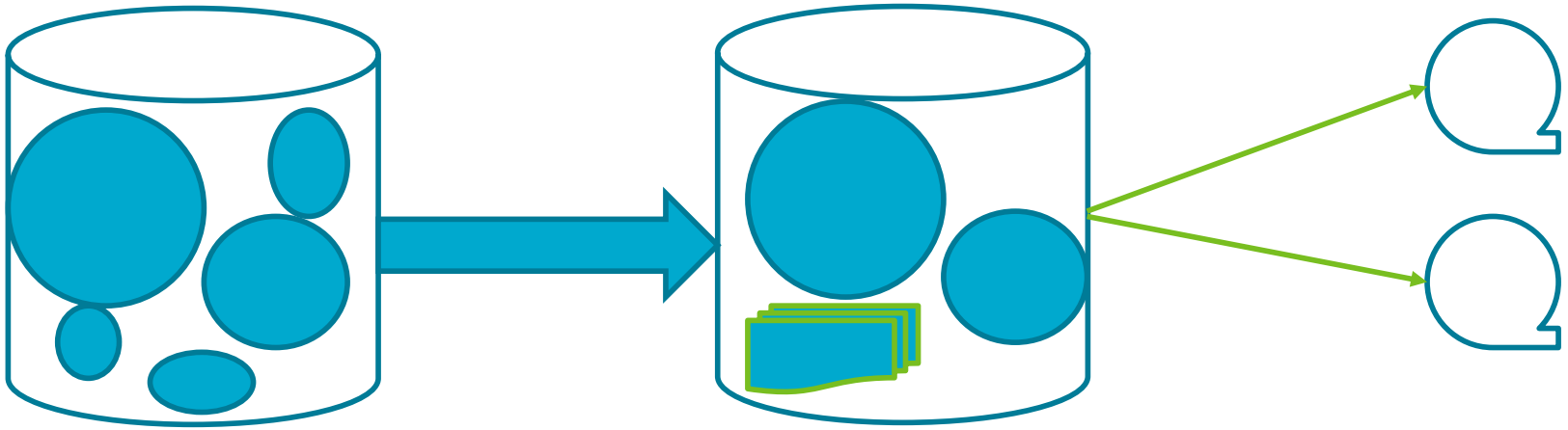
```
cd a_working_directory
mirror.sh create      # W->M..T
```



Create a mirror

- 1) Create a mirror

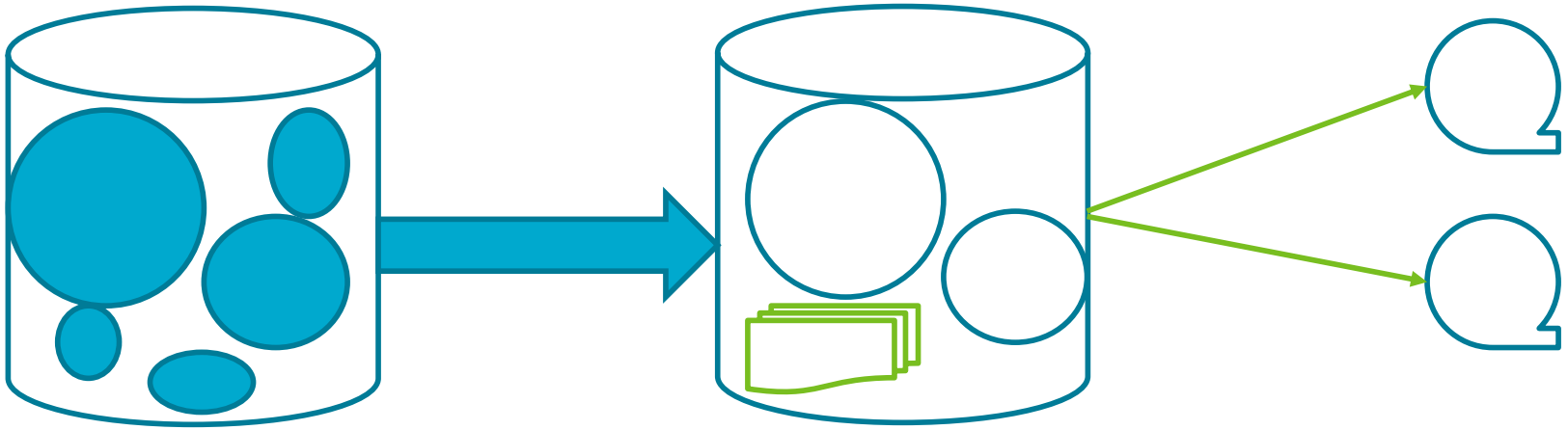
```
cd a_working_directory
mirror.sh create      # W->M..T
```



Create a mirror

- 1) Create a mirror

```
cd a_working_directory
mirror.sh create     # W->M..T
```

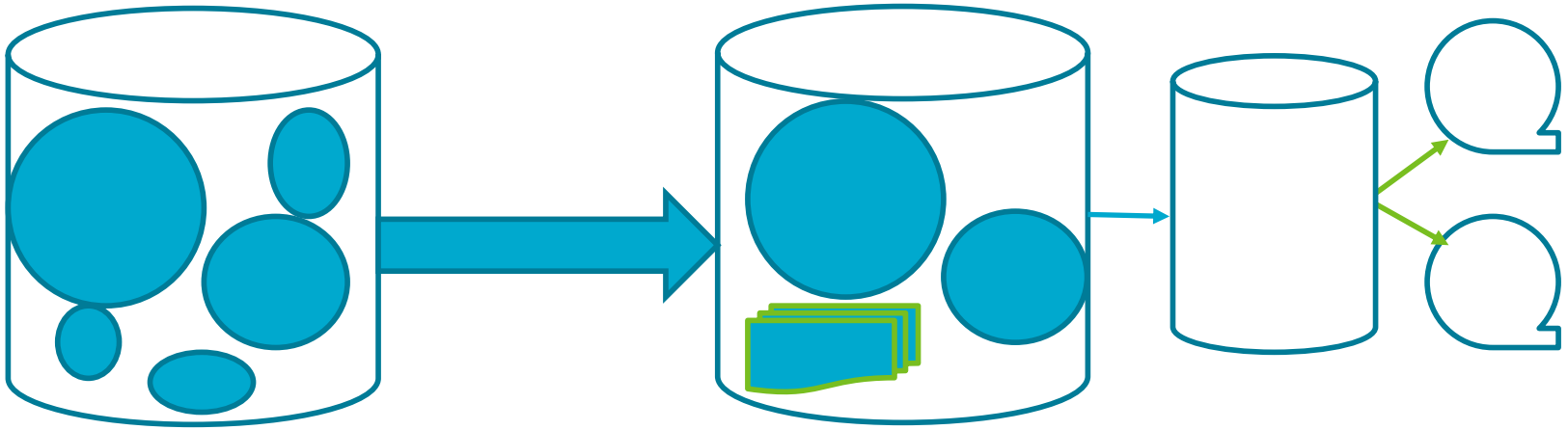


```
mirror.sh release – invokes dmput -r
```

Create a mirror: NCI

- 1) Create a mirror

```
cd a_working_directory
mirror.sh create      # W->M..T
```



Examples from prototype utility

5) After adding to the working area, or making other changes

```
mirror.sh update      # W->M..T
```

6) After doing a release, and knowing that files will need to be restored

```
mirror.sh recall      # M<-T
```

7) After the above flush, or system flush, restore the working area.

(This could be embedded into batch scripts).

```
yes | mirror.sh rebuild  # W<-M
cd a_working_directory
mirror.sh sync        # W<-M
```

Examples from prototype utility

- 11) To undertake a move of an existing area, say in NCI /g/data/mydata to a $FLUSH* area, but don't need to have the data ready now. This can be quite quick if a mirror already exists, since no user data is moved from the old to the new location.

```
cd /g/data/mydata
mirror.sh create (if not done already)        # W->M..T
mirror.sh sync (to check)                     # W<-M
mirror.sh flush (to remove data)              # xW
mirror.sh moveto $FLUSHDIR/mydata       # W->W
cd $FLUSHDIR/mydata
mirror.sh list
```

Examples from prototype utility

- 12) To squirrel away part of the $HOME area on pearcey, bracewell or NCI raijin because quota limits have been hit.

```
cd $HOME/dormant
mirror.sh create        # W->M..T
mirror.sh list
mirror.sh flush         # xW
mirror.sh release       # M..T
```

- 13) To restore the above files

```
cd $HOME/dormant
mirror.sh recall        # M<-T
mirror.sh sync          # W<-M
```

Examples from prototype utility

- 16) To create and update a mirror and keep all previous versions of files (Tim Erwin's suggestion)

```
cd $FLUSHDIR/mydata
mirror.sh create                  # W->M..T
# … make changes
mirror.sh update preserve       # W->M..T
# … make changes
mirror.sh update preserve       # W->M..T
# Get back all mirrored versions of file1
mirror.sh restore ./file1       # W<-M
```

Examples from prototype utility

- 19) On SC systems, get data from a remote system such as NCI, and then create mirror: keep on adding data, and remove data from NCI after copying.

```
cd $FLUSHDIR/from-nci-1
um=nci_user@r-dm.nci.org.au
d=/short/PROJ/USER/good
umd="${um}:$d"
mirror.sh getremote $umd            # R->W
mirror.sh create                    # W->M..T
mirror.sh getremote $umd            # R->W
mirror.sh update                    # W->M..T
# Optionally remove the data from NCI.
ssh $um "/bin/rm –rf $d"
# Add another collection.
d=/short/PROJ/USER/good2
umd="${um}:$d"
mirror.sh getremote $umd            # R->W
mirror.sh update                    # W->M..T
```

Examples from prototype utility

- 20) On SC systems, put data from the working area to a remote system such as NCI.

```
cd $FLUSHDIR/from-nci-1
mirror.sh putremote ./good/ \
     nci_user@r-dm.nci.org.au:/short/PROJ/USER/good # R<-W

# Add another collection.
mirror.sh putremote ./good2 \
     nci_user@r-dm.nci.org.au:/short/PROJ/USER/     # R<-W
```

DMF issues

- Relies on DMF to provide protected storage, ‘infinite’ capacity
- Cushions the use from dealing with DMF directly and having to make decisions about performance trade-offs
- Detection of DMF-managed areas
  - decisions about dmgets, rsync options
  - Uses local dmget wrapper (PSDE) to process files in order of retrieval
- Collections, to reduce inode count
  - Intelligent selection of archive sizes – small files tarred up, large files not.
- Users tardir utility: keeps lists (likely to stay on-line) of files that are in tar archives
- Copes with direct access to HSM (SC Data Store from ruby), slow mount access to HSM (SC Data Store from SC clusters), and indirect access (Massdata at NCI).

Conclusion

- With scratch under control:

  Death of /data and /g/data* !

- Hopelessly unmanaged and unmanageable storage!
- CSIRO users can use either scratch, or HSM-managed, or Bowen Research Cloud or DAP
- Prototype utility to mirror scratch areas into persistent storage, and have the ability to re-create after flushing
- Utilities for profiling holdings – age and waste ($)

Utility – file profiling – diu2

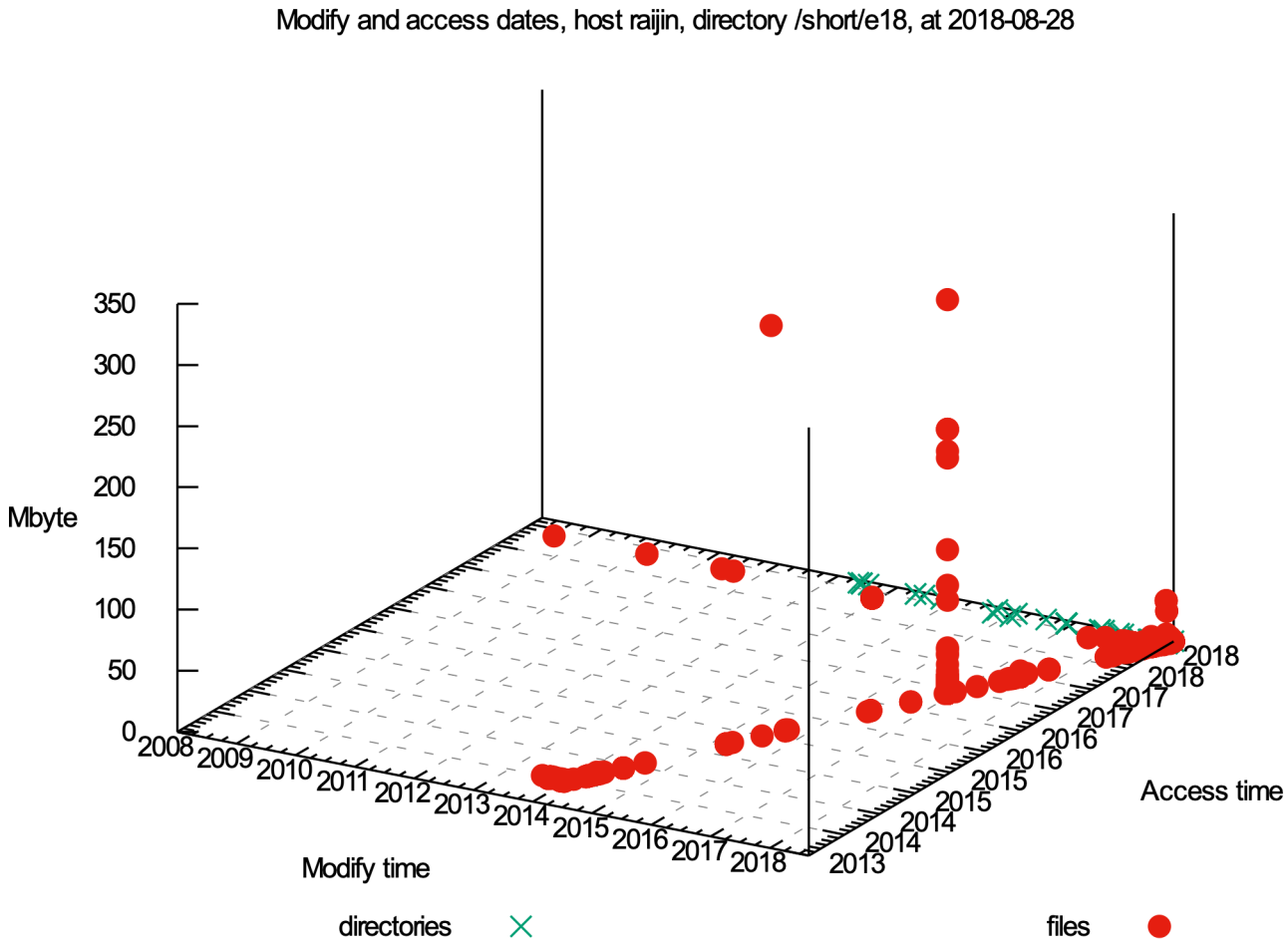
Size
File types, e.g.
7 d
215 f
Largest file: ….
Earliest access time: ….
Earliest modify time: ….
Average file age is 2018-09-06 which is 103 days old.
Average data age is 2018-09-25 which is 84 days old.

Most wasteful file wastes 30444 Gbyte-days:
Total waste is 119021647 Gbyte-days, cost is about $50835

Utility – file profile graphing – plot.diu2.sh

- Plots of file profiles by:
  - access time
  - modify time
  - number of files
  - amount of data
  - sizes

Example profile



Conclusion

- Data lifecycle management
- User education
- Assisting people in deleting or archiving data where appropriate
- Helping data not to be lost
- Helping users have good places to store data collections
- Maximising the value of our data:

  CSIRO's data principles project
- Utility provides: protected storage, working filesystem on several SC hosts, high performance, infinite capacity, zero-cost to end-users, reasonably reliable, on-line for direct access: high bandwidth, low latency, no migration (HSM) – most of desirables.

Solution

Org	Filesystem	Access	Protection / backup	Reach	Perf.	Capacity	Relia-bility	Management	Band-width	Latency
CSIRO SC	/datastore	Direct on ruby	Yes, dual copies (interstate): 92 days backup	SC systems	H	∞ no total space quotas	H	inode and on-line quotas, HSM	H	L—H
CSIRO SC	Cluster /home	NFS	Yes, years (selective)	Cluster	L	L	H	Quotas	L	M
CSIRO SC	/flush0	Direct on ruby	Nil	Local to ruby	H	M	H	Quotas, flushing	H	L
CSIRO SC	/flush[1-]	NFS	Nil	SC systems	L—M	M	M	Quotas, flushing	L	M
CSIRO SC	Working area + mIrror	Direct	By mirror (backup selectable)	User choice	H—M	∞	M	Quotas, flushing, HSM	H	L
NCI	/home	NAS	Yes	Single system	L	L	H	Quotas	L	M
NCI	/short	Lustre	Nil	Single system	H	H	M	Quotas	H	M
NCI	/g/data*	Lustre	Nil	NCI systems	H	H	M	Quotas	H	M
NCI	Massdata	Indirect	Yes, dual buildings	NCI systems	M	∞, but space quotas	H	Small file restrictions, quotas, HSM	H	H

Thank you

CSIRO IMT Scientific Computing
Robert C. Bell
CSIRO HPC National Partnerships
t +61 428 108 333
e Robert.Bell@csiro.au
w www.csiro.au