μ-VIS Multiscale, Multidisciplinary, Microtomographic Volume Imaging at Southampton
www.southampton.ac.uk/muvis

μ-VIS - DLS pilot

Richard Boardman
μ-VIS X-ray Imaging Centre
University of Southampton, UK

μ-VIS X-ray Imaging Centre

- A number of X-ray Computed Tomography systems
  - Resolutions down to <1μm
  - Energies up to 450keV
  - Sample size ranging from millimetre to metre scale
- Computing resource for image analysis
  - Around a dozen high performance workstations
  - Up to 512GB RAM
  - 48 CPU cores
  - ~300TB storage
  - Large range of 3D analysis and reconstruction software

Dataset sizes for a single scan

- μ-VIS
  - 20TB projection data
  - 30TB reconstructed volume
  - plus metadata, derivative datasets, analyses *&c.*
  - Data generation rate: up to 10GB/minute/machine, 24/7
- DLS
  - Datasets of the order of 50-100GB
  - Several datasets per sample
  - ~100 samples per visit (~2-3 days)
  - Phase retrieval inflates the size
  - Typically 10-50TB per visit

The problem

- Synchrotron beamtime sessions are hard work!
  - Typically 2-3 days long (or more)
  - 24 hours a day (time is expensive!)
  - Postdocs, postgrads work in shifts around the clock performing scans generating data - busy and tired
- Generated datasets (~50TB) copied on to hard disks and taken home
- Copied to local high-performance datastores
- Analysis commences

The problem, compounded

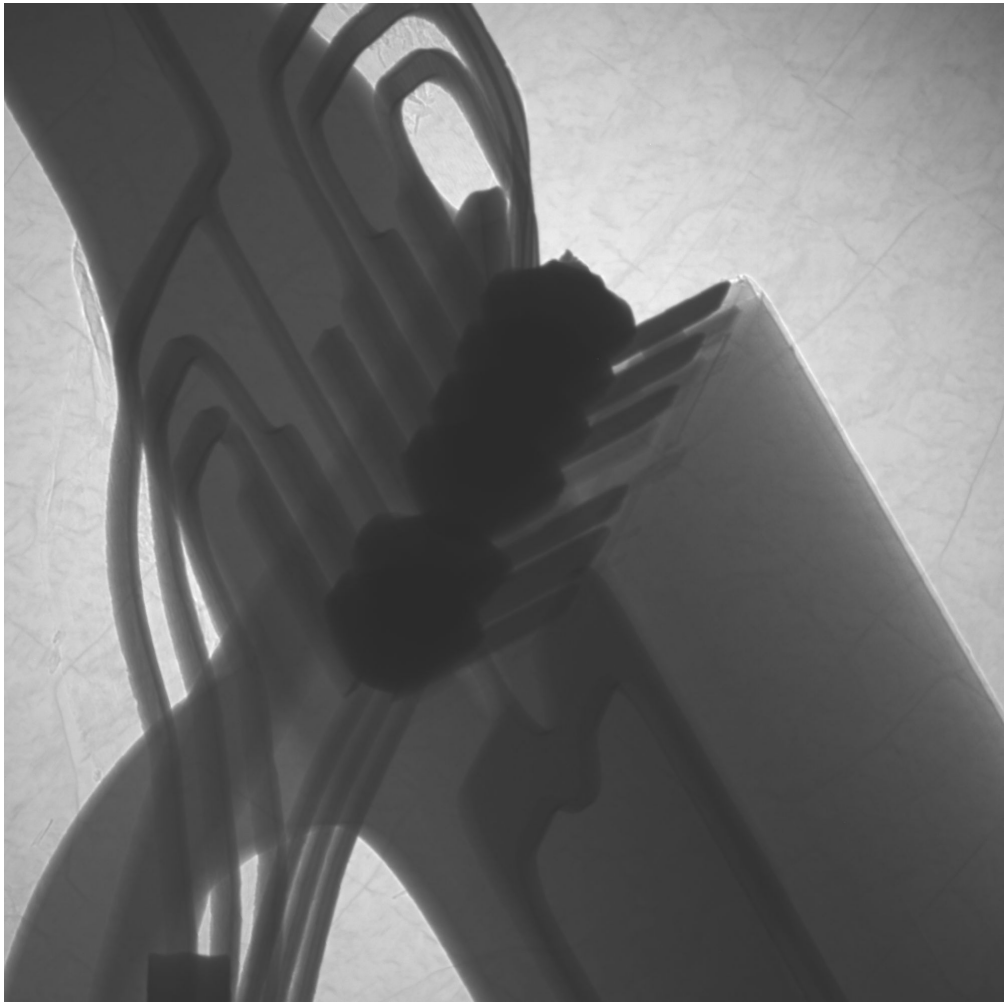
What if … ?

- Reconstructions were invalid
  - Re-reconstructing is hard work - time, effort, data transfer
- Beamline issues
  - Scintillator sub-standard
  - Images noisier than expected
- Inexperienced beamline team
  - Most contain people new to working on a beamline
  - Might not know the images aren’t optimal
- This can “write off” the whole trip
  - Time, effort, money
  - Samples may be destroyed or otherwise not reusable

Example problem 1

- Hard disk head interface
- 381nm resolution
- 60kVp @5W

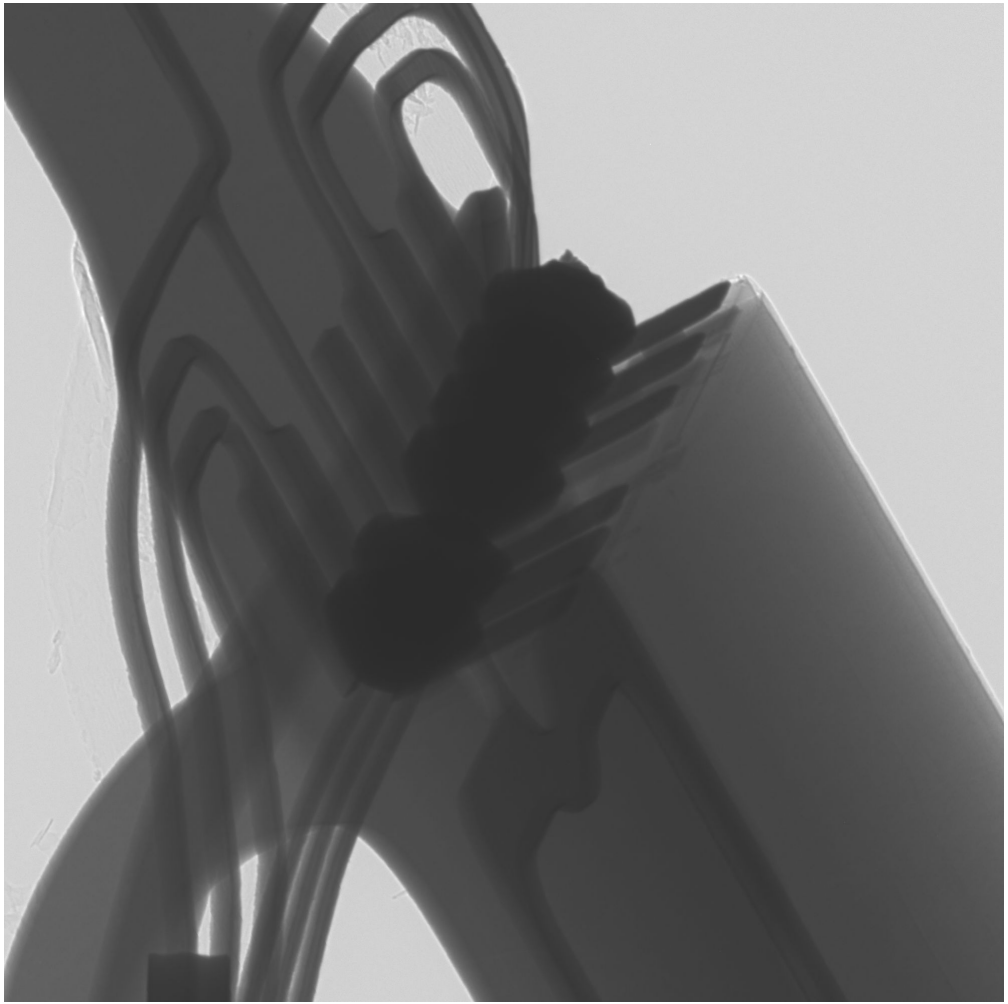
What’s the problem?



Solution 1

- Hard disk head interface
- 381nm resolution
- 60kVp @5W

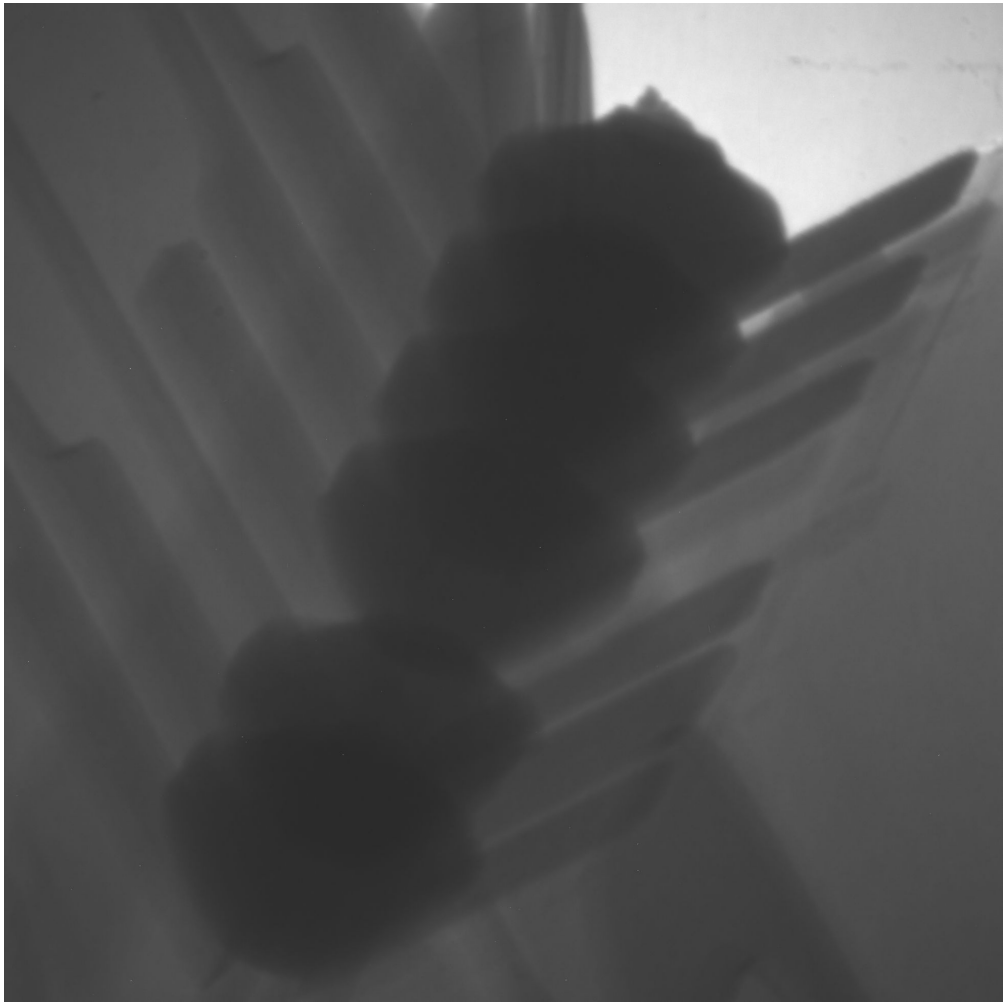
Scintillator “crazing”



Example problem 2

- Hard disk head interface
- 193nm resolution
- 160kVp @ 5W

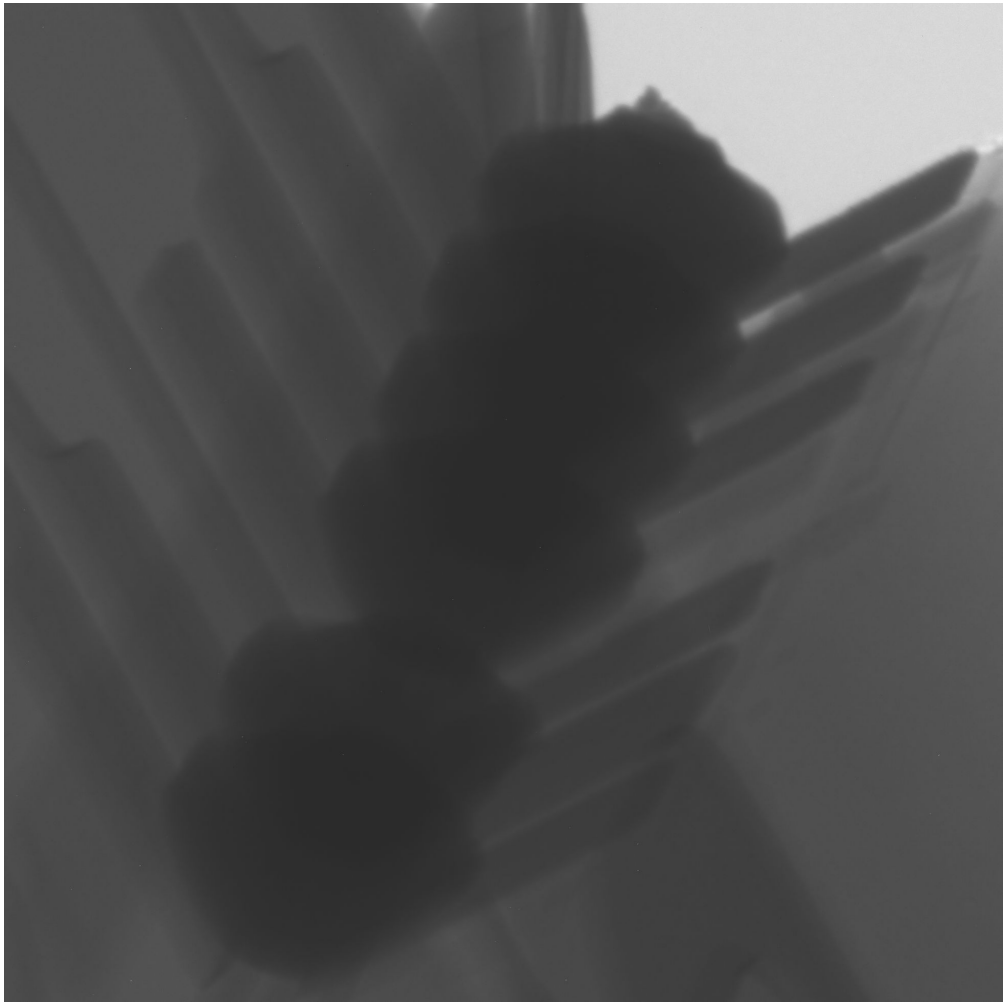
What’s the problem?



Solution 2

- Hard disk head interface
- 193nm resolution
- 160kVp @ 5W

Scintillator damage



Priorities

Get all the data!

1. Get it back **correctly** (data we do get, should be the data we acquired)
2. Get it back **reliably** (data we do get, should be all the data we acquired)
3. Get it back **quickly** (as soon as possible…)
4. Get it back **during** (…no, sooner!)

Current “gold standard” - **`rsync`** with hard disks. Slow but correct and reliable

Can we improve on this?

Current protocol

- For each experiment, a unique local ID is generated on campus
- Researchers take a “bag of hard disks” with them to the synchrotron
- During the experiment, as data are generated, they are copied to the disks organised by the local ID
  - Sinograms
  - Metadata
  - Reconstructions
  - Derivative datasets (*e.g.* phase retrieved data)
- Disks are then returned to campus
- Experimental data then ingested into μ-VIS’ data storage, using the unique local ID to put this into an accessible place for the researcher

Problems with the current protocol

- We can verify that the datasets are good on the disk (`rsync`, other checksums)
- However, if one of these datasets is corrupted, we have to transfer this again
  - Remote copy over the network (with varying success rates depending on available tools)
  - Or, post a hard disk
- Same applies if a disk is lost or damaged in transit
  - Rare, but possible
- Lost disk would be more serious
  - Not yet happened but it is an issue; a third party could access and read the data - probability is low but non-zero
  - Encryption isn't a viable option - experimental conditions, temporal pressures and system unknowns, incompatibilities makes planning and execution difficult
- But, overall it is **correct**, and over time it is **reliable**

Ingesting data from hard disk

- “Copy” tools from most file browser programs (e.g. MS Explorer, OSX Finder) aren’t great for this sort of thing
  - Some have a “merge” option, “skip”, but are not strictly reliable if the copy is interrupted
  - Don’t checksum data after copy
  - If a copy is cancelled partway through and restarted, inconsistent state is possible
- `rsync` alleviates this (https://rsync.samba.org/) - fast, incremental file transfer
- Data ingestion tasks are often run via scripts to do other things in the middle (renaming, logging, emailing and so on)

rsync command-line view

```
[rpb@sprawl /mnt/tmp] lsblk /dev/sde
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sde      8:64   0  5.5T  0 disk
├─sde1   8:65   0  200M  0 part
└─sde2   8:66   0  5.5T  0 part /mnt/tmp
[rpb@sprawl /mnt/tmp] df -h /mnt/tmp
Filesystem      Size  Used Avail Use% Mounted on
/dev/sde2       5.5T  4.9T  579G  90% /mnt/tmp
[rpb@sprawl /mnt/tmp] ls
20180301_ESRF_1799_3900_090_3_NS1_   20180301_ESRF_1799_3900_090_3_RBZ_   20180301_ESRF_1799_3900_090_4_22_   20180301_ESRF_1799_3900_090_4_NS1b_   20180301_ESRF_1799_example_images
20180301_ESRF_1799_3900_090_3_NS2_   20180301_ESRF_1799_3900_090_4_10_   20180301_ESRF_1799_3900_090_4_23_   20180301_ESRF_1799_3900_090_4_NS2_   20180301_ESRF_1799_misc
20180301_ESRF_1799_3900_090_3_RBX_   20180301_ESRF_1799_3900_090_4_12_   20180301_ESRF_1799_3900_090_4_24_   20180301_ESRF_1799_3900_090_4_RBX_   20180301_ESRF_1799_volfloat
20180301_ESRF_1799_3900_090_3_RBY_   20180301_ESRF_1799_3900_090_4_15_   20180301_ESRF_1799_3900_090_4_24b_   20180301_ESRF_1799_3900_090_4_RBY_   20180301_ESRF_1799_volraw
20180301_ESRF_1799_3900_090_3_RBYb_  20180301_ESRF_1799_3900_090_4_20_   20180301_ESRF_1799_3900_090_4_NS1_   20180301_ESRF_1799_3900_090_4_RBZ_
[rpb@sprawl /mnt/tmp] rsync -a --progress --stats * /mnt/data3/CTData/sr4g15/
```

```
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0004.edf
     11,061,248 100%   16.06MB/s    0:00:00 (xfr#12, ir-chk=6138/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0005.edf
     11,061,248 100%   13.58MB/s    0:00:00 (xfr#13, ir-chk=6137/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0006.edf
     11,061,248 100%   11.71MB/s    0:00:00 (xfr#14, ir-chk=6136/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0007.edf
     11,061,248 100%   10.41MB/s    0:00:01 (xfr#15, ir-chk=6135/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0008.edf
     11,061,248 100%   81.77MB/s    0:00:00 (xfr#16, ir-chk=6134/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0009.edf
     11,061,248 100%   42.71MB/s    0:00:00 (xfr#17, ir-chk=6133/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0010.edf
     11,061,248 100%   29.14MB/s    0:00:00 (xfr#18, ir-chk=6132/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0011.edf
     11,061,248 100%   21.75MB/s    0:00:00 (xfr#19, ir-chk=6131/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0012.edf
     11,061,248 100%   17.49MB/s    0:00:00 (xfr#20, ir-chk=6130/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0013.edf
     11,061,248 100%   14.63MB/s    0:00:00 (xfr#21, ir-chk=6129/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0014.edf
     11,061,248 100%   12.65MB/s    0:00:00 (xfr#22, ir-chk=6128/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0015.edf
     11,061,248 100%   11.07MB/s    0:00:00 (xfr#23, ir-chk=6127/6174)
20180301_ESRF_1799_3900_090_3_NS1_/1799_3900_090_3_NS1_0016.edf
      1,605,632  14%    1.53MB/s    0:00:06
```

Transfer over the network?

Perception: “the internet is too slow for this”

Reality: ?

- Currently, occasional datasets are transferred across (the ones that weren’t ready in time for the disk copy, or those that failed checksums)
- People will use their campus-connected, possibly congested personal workstations for this
- Very best case scenario: 1 gigabit/second

Globus

- Efficient, reliable transfer of files
- Transfer occurs between two “endpoints”
- DLS already has an endpoint…

… set up an endpoint at µ-VIS to test this out

Bandwidth contention

Problem: µ-VIS has a standard campus 1GbE connection.

- Best case? *circa* ⅔ hard disk speed

Answer: 10GbE connection

Pilot

- μ-VIS set up a Globus endpoint on a development workstation
- iSolutions provided a 10GbE connection for testing
- Trialled with high-speed data store (sustained >2GB/sec write speed)
- Test datasets sent to μ-VIS from:
  - DLS
  - CERN
- Real datasets sent from DLS

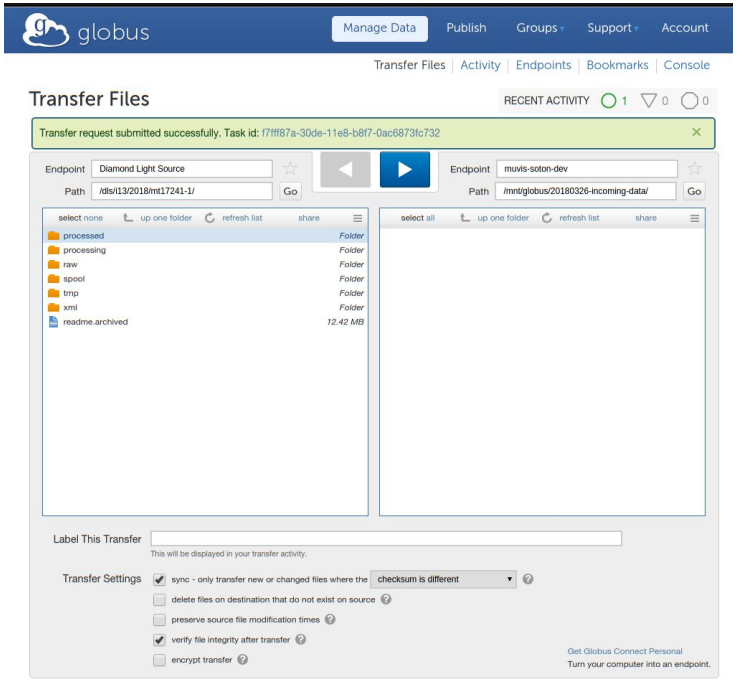
Using Globus: the user experience - web

- Log in via https://www.globus.org/
- Select source and destination endpoints
- Log in to each
- Select source dataset
- Optionally name the transfer and set other options (*e.g.* sync, no verify)
- Hit the arrow to start
- Close the browser

Can be done anywhere, even over a slow connection - just talking to a service that will manage everything for you

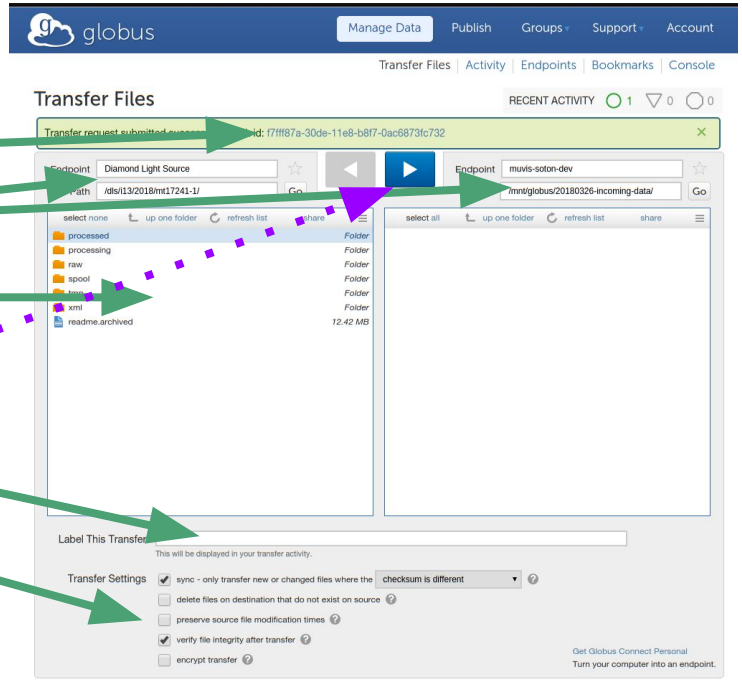
Globus file transfer view

- Transfer ID
- Endpoint names and paths
- File/directory browser
- Label field
- Transfer settings
- Transfer start button (directional)



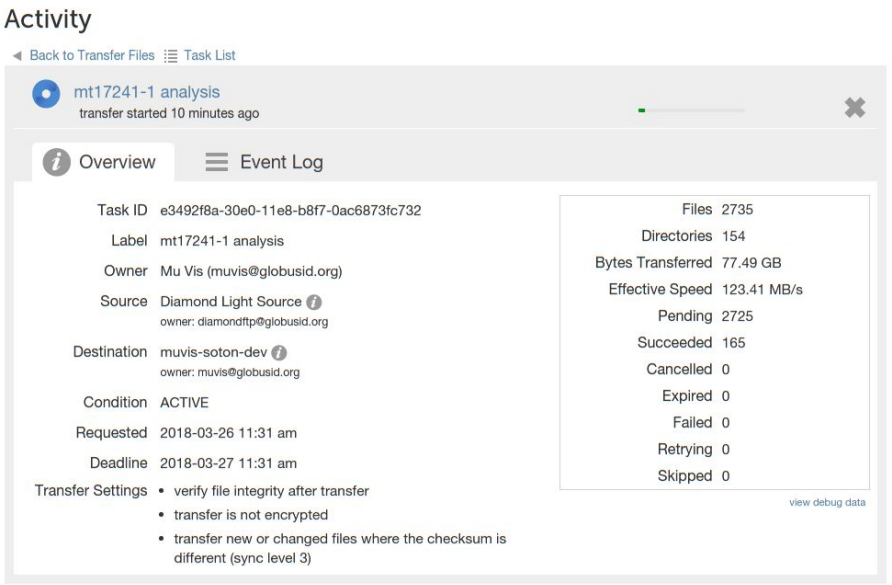
Globus file transfer view

- Transfer ID
- Endpoint names and paths
- File/directory browser
- Label field
- Transfer settings
- Transfer start button (directional)



Globus transfer details view

- Picking the task ID from one of the views will show the activity on that transfer
- Can see
  - how much has been transferred
  - data transfer rates
  - labels, endpoints
  - transfer settings
  - timestamps and conditions (e.g. failed, pending, succeeded)

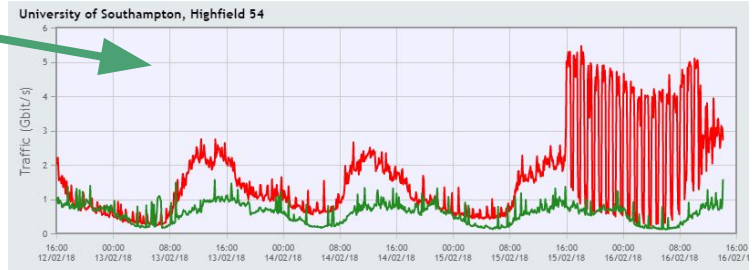


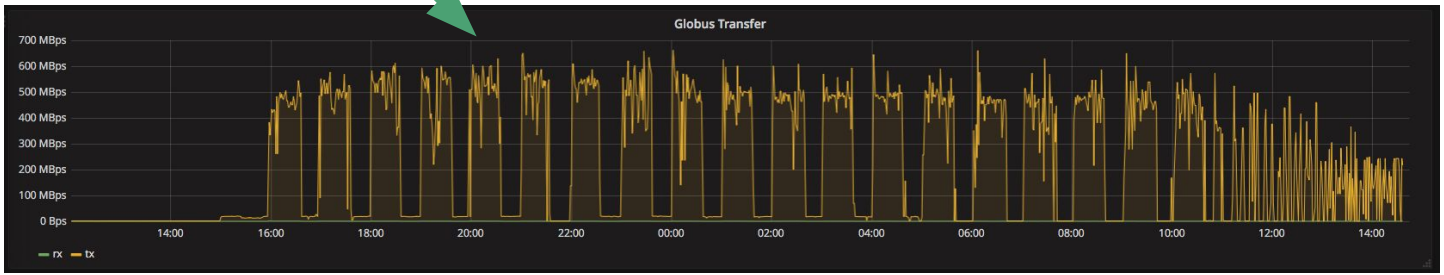
What’s happening on an endpoint?

- Unsurprisingly, lots of network activity!
- Here we see spot rates of over 800 megabytes / second
- Underlying datastore (SMB mount) in this case ~400 megabytes / second

What's happening on the 10GbE link?

- This particular test was transferring single, large files every 30 minutes
- Traffic at Southampton
  (courtesy of Simon Lane)
- Traffic at Diamond
  (courtesy of Alex White)





After the transfer...

```
20180120034700_99055  20180120143543_99087  20180121003221_99116  20180121084111_99145  20180121174641_99174
20180120040323_99056  20180120150608_99088  20180121004825_99117  20180121085522_99146  20180121193855_99175
[rpb@sprawl /mnt/globus/20180326-incoming-data/processed/auto_processed] cd 20180121050404_99132
[rpb@sprawl /mnt/globus/20180326-incoming-data/processed/auto_processed/20180121050404_99132] ls -lha
total 185M
drwxr-xr-x 2 rpb root    0 Mar 26 11:26 .
drwxr-xr-x 2 rpb root    0 Mar 26 11:18 ..
-rwxr-xr-x 1 rpb root 214K Mar 26 11:26 99132_processed.nxs
-rwxr-xr-x 1 rpb root 184M Mar 26 11:25 tomo_p3_tomopy_recon.h5
-rwxr-xr-x 1 rpb root 1.2K Mar 26 11:26 user.log
[rpb@sprawl /mnt/globus/20180326-incoming-data/processed/auto_processed/20180121050404_99132] cat user.log
2018-01-21 05:04:10,762 - User Log Started
2018-01-21 05:04:10,763 - User Log location is '/dls/i13/data/2018/mt17241-1/processed/auto_processed/20180121050404_99132/user.log'
2018-01-21 05:04:13,646 - Plugin list check complete!
2018-01-21 05:04:13,936 - *Running the DarkFlatFieldCorrection plugin*
2018-01-21 05:04:14,969 - DarkFlatFieldCorrection -   0% complete
2018-01-21 05:04:25,033 - DarkFlatFieldCorrection - 100% complete
2018-01-21 05:04:25,133 - DarkFlatFieldCorrection : 12 processes report : Nothing to Report
2018-01-21 05:04:25,243 - *Running the VoCentering plugin*
2018-01-21 05:04:25,253 - VoCentering -   0% complete
2018-01-21 05:05:43,703 - VoCentering - 100% complete
2018-01-21 05:05:48,288 - VoCentering : 12 processes report : Nothing to Report
2018-01-21 05:05:48,318 - *Running the TomopyRecon plugin*
2018-01-21 05:05:48,326 - TomopyRecon -   0% complete
2018-01-21 05:05:50,865 - TomopyRecon - 100% complete
2018-01-21 05:05:52,182 - TomopyRecon : 12 processes report : Nothing to Report
2018-01-21 05:05:52,251 - **********
2018-01-21 05:05:52,252 - * Processing Complete *
2018-01-21 05:05:52,253 - **********
[rpb@sprawl /mnt/globus/20180326-incoming-data/processed/auto_processed/20180121050404_99132]
```

After the transfer...

One of many datasets - a quick inspection (this dataset is in HDF5 format)

```
[rpb@sprawl /mnt/globus/20180326-incoming-data/processed/auto_processed/20180121050404_99132] h5ls -v -r tomo_p3_tomopy_recon.h5
Opened "tomo_p3_tomopy_recon.h5" with sec2 driver.
/                        Group
    Location:  1:96
    Links:     1
/3-TomopyRecon-tomo      Group
    Location:  1:800
    Links:     1
/3-TomopyRecon-tomo/data Dataset {2004/2004, 12/12, 2004/2004}
    Location:  1:1832
    Links:     1
    Chunks:    {501, 1, 501} 1004004 bytes
    Storage:   192768768 logical bytes, 192768768 allocated bytes, 100.00% utilization
    Type:      native float
[rpb@sprawl /mnt/globus/20180326-incoming-data/processed/auto_processed/20180121050404_99132]
```

...we could start working with the data whilst the rest are still transferring

Globus - pros and cons - the user perspective

Advantages:

- Relatively simple process to transfer complex datasets
- Persistent connection by the user not required
- Can be initiated anywhere you have a web connection
- Lots of (relatively) small files transfer much more quickly than `rsync`

Disadvantages:

- Not 100% reliable: it can't get past permission errors, and has a funny "retry" loop (rsync would just skip over the file and continue with the rest)
- Does not check available disk space before transferring (!)
- Although straightforward, it is "yet another system" to learn

Discussion

- Transfer rates with Globus are great - indicates infrastructure between Soton and DLS is more than capable
- Can be vague when it errors
- “Feels” less robust than rsync (and Globus’ command line interface is “clunkier” than rsync - more sensible to script rather than “one line” it)

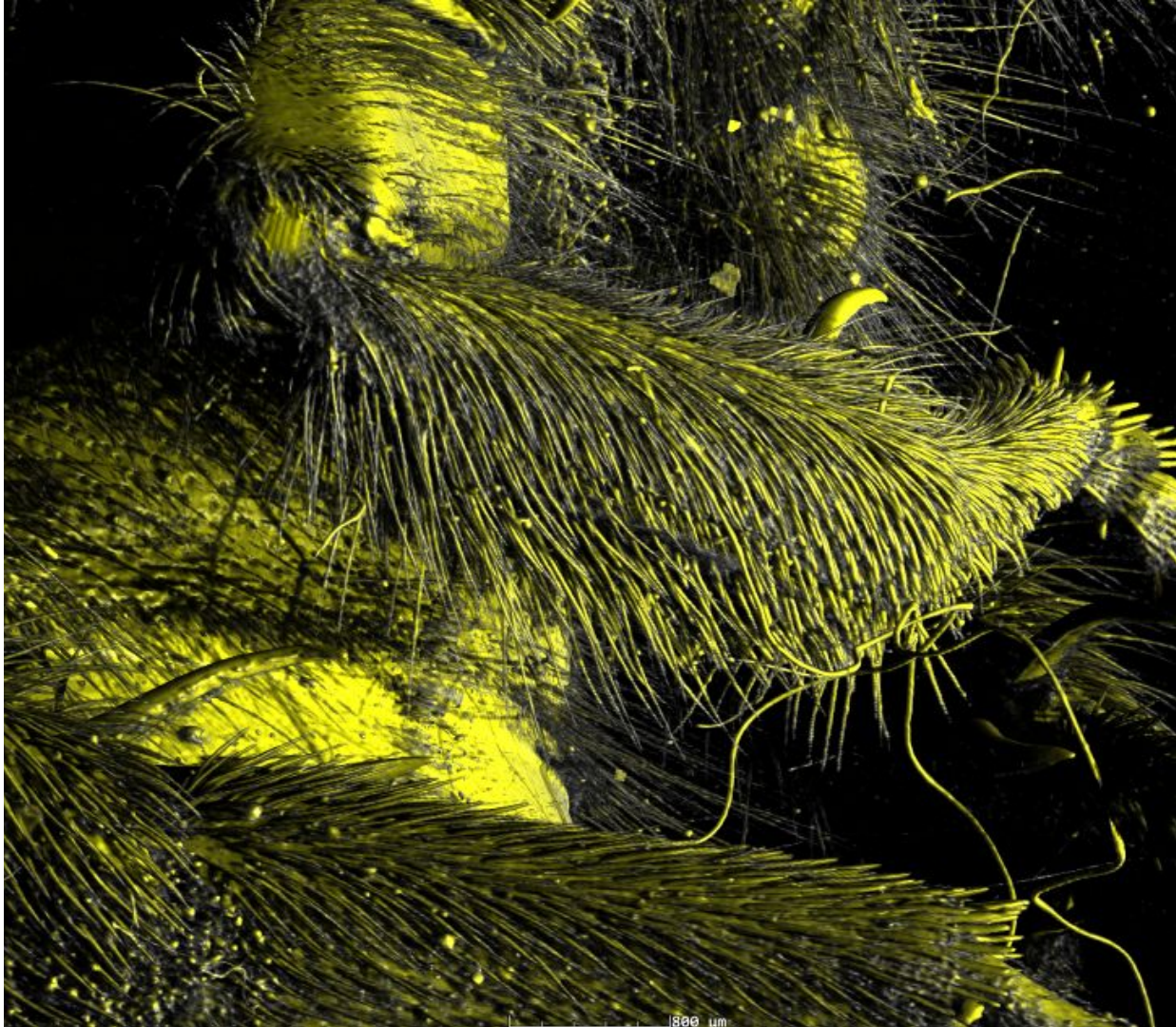
Quandary!

- rsync may be slower, but it is more robust
- Network is very snappy and transfer times are excellent
- If a site offers network-only transfers with Globus, or “bag of disks”, which would I choose?
    - For examining selected data at another site during an experiment, only one choice (Globus)
    - For reliably transferring the data between sites? Hard to choose!
    - In the current state, I would err on the side of reliability over performance and opt for “bag of disks”
    - If rsync/ssh option available over the (fast) network, this would be preferable to both **to me**, as I am comfortable with using the command line tools
- If we had someone less *au fait* with rsync over ssh, we would have to fall back to “bag of disks”: if Globus is “fire and forget”, chance of missing files (permissions)

µ-VIS X-ray Imaging Centre
Room 1015, Building 5 (Eustice)
Highfield Campus
University of Southampton
SO17 1BJ

muvis@soton.ac.uk

http://www.muvis.org



Close up of buff-tailed bumblebee (*bombus terrestris*), Nikon HMX 225