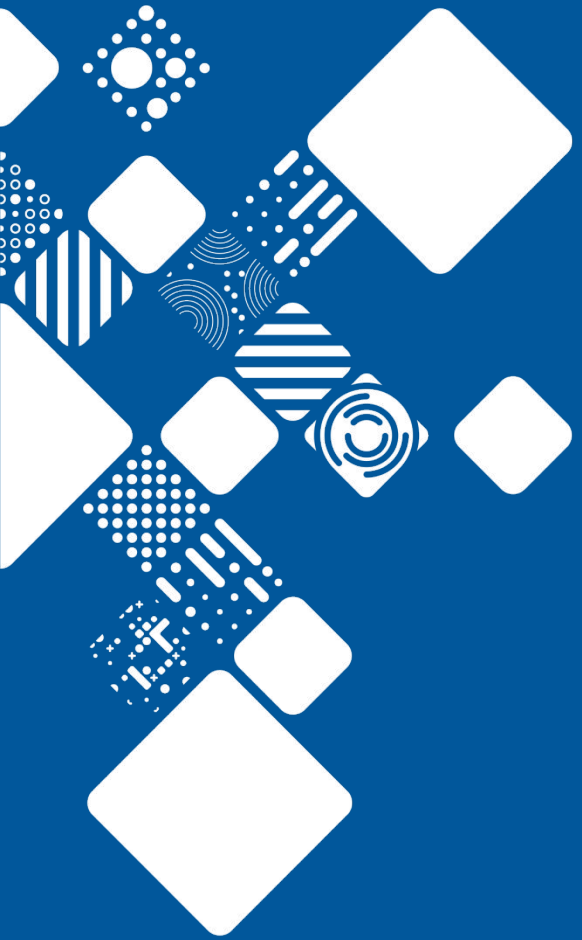


Migrating Data When Decommissioning PetaBytes of Storage

John Constable
Informatics Support Group, ICT
jc18@sanger.ac.uk
@kript

wellcome sanger institute

Background

- 19PB of genomic data in 399 Resources on 76 resource servers over six Zones
- 41 servers need decommissioning this year, another 10 next year; aka ~10 PB across three types of hardware.
- Generating 10TB/data week, expecting to go up to 760TB if the scientists turn on all the PacBio/Nanopore sequencers they might buy for upcoming programs like 'Tree of Life'

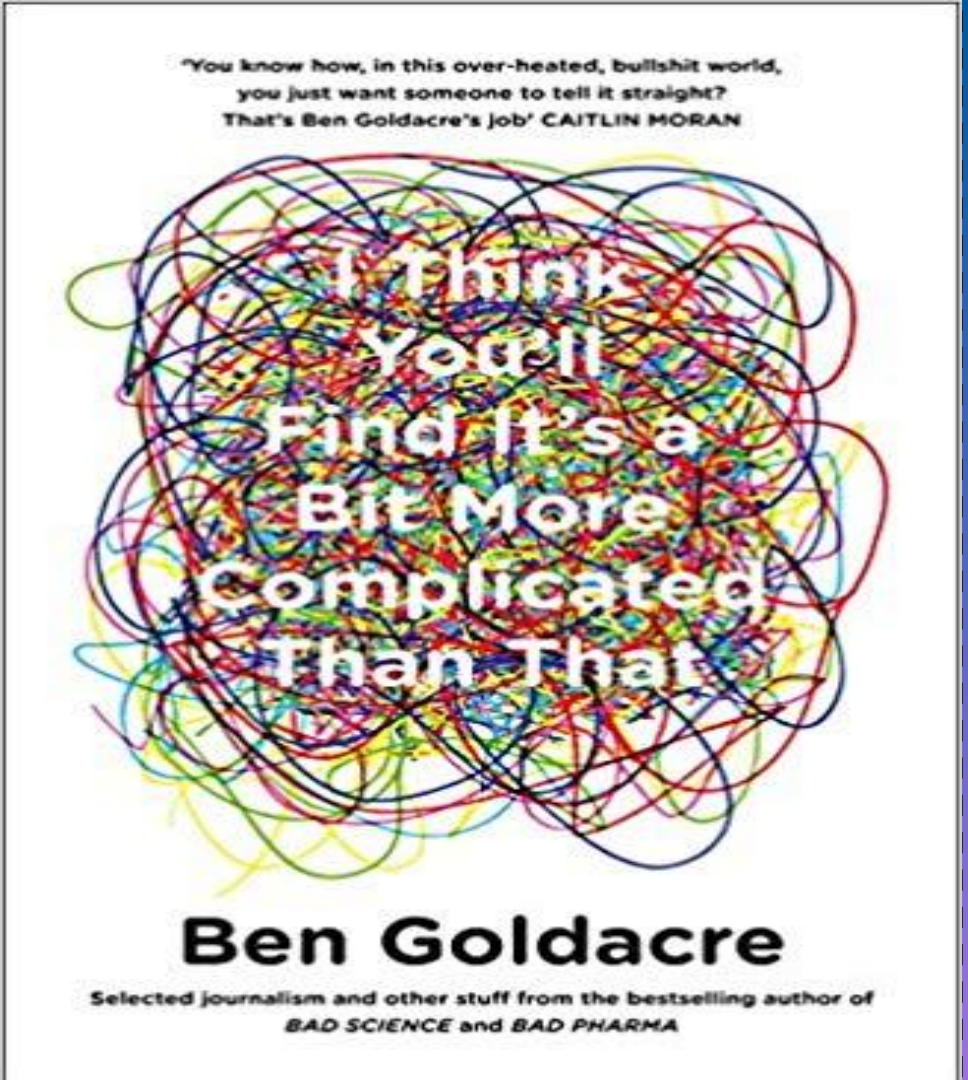
wellcome sanger institute

https://docs.irods.org/4.2.6/system_overview/tips_and_tricks/#decommissioning-a-storage-resource

Advice is;

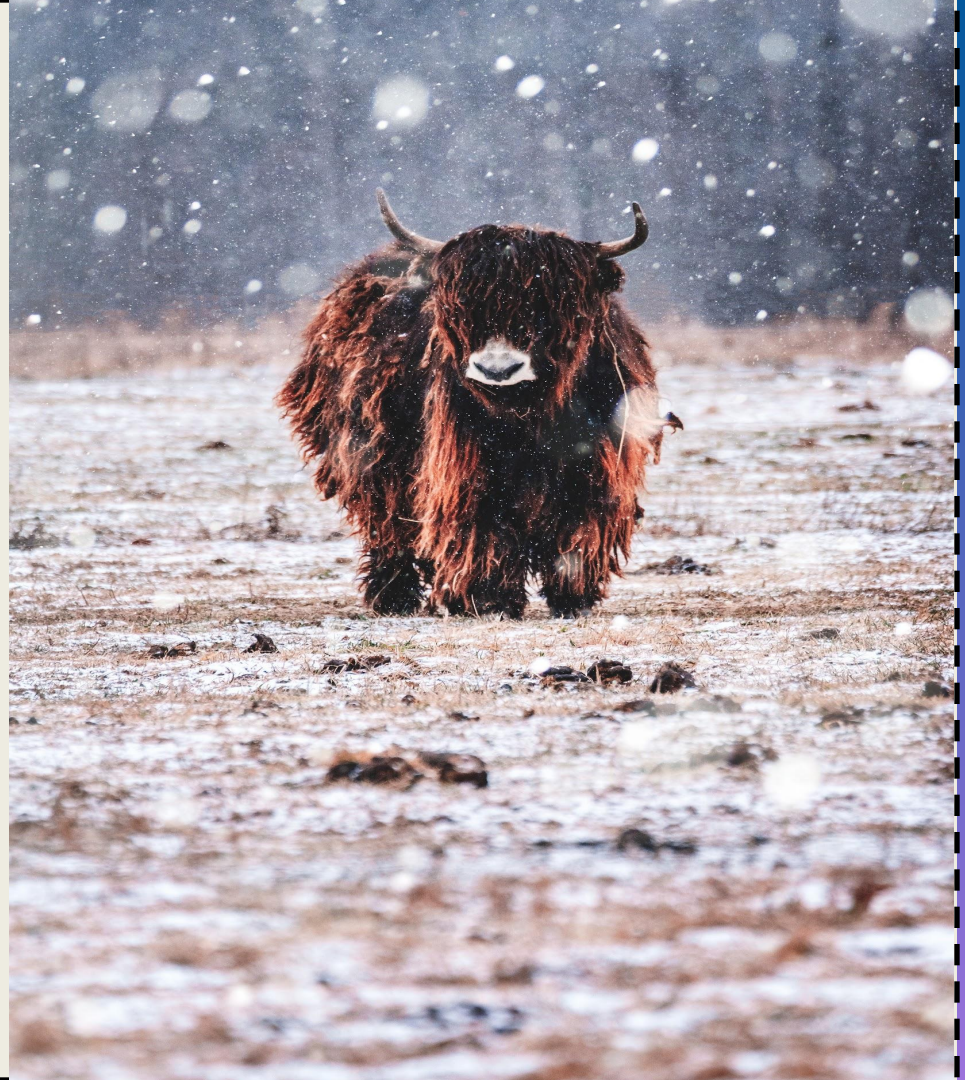
1. Determine which iRODS server will host the new device.
2. Create a new iRODS resource that uses the new device.
3. Add the new resource to the appropriate resource hierarchy (could be standalone).
4. Replicate data to the new resource.
5. Trim data from the to-be-retired resource.
6. Remove the to-be-retired resource.
7. Safely disconnect the to-be-retired device.

wellcome sanger institute



4. Replicate data to the new resource.

wellcome sanger institute



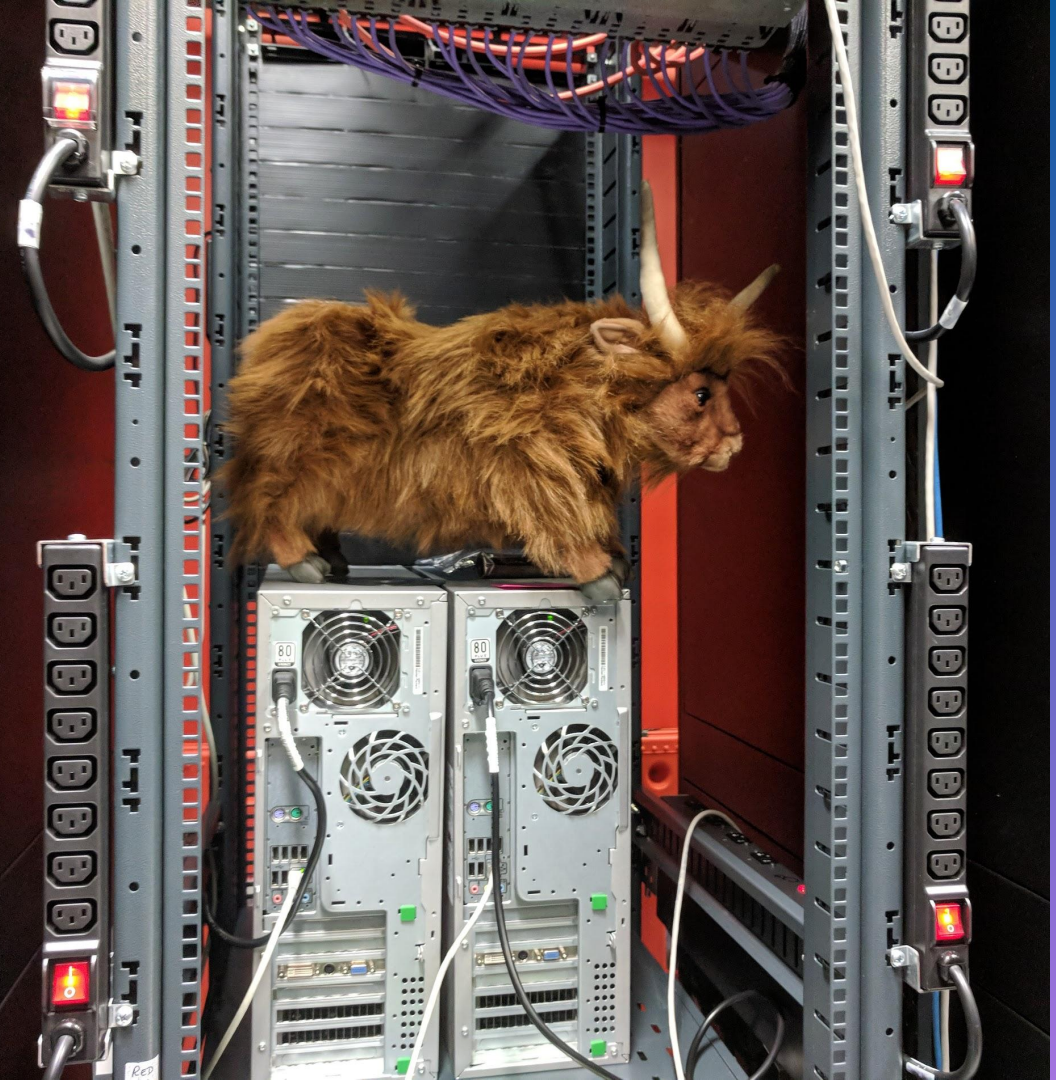
Yak Shaving

Any apparently useless activity which, by allowing you to overcome intermediate difficulties, allows you to solve a larger problem.

"I was doing a bit of yak shaving this morning, and it looks like it might have paid off."

Definition credit ghyston.com
Photo by Bryan Minear on Unsplash

wellcome sanger institute



Standing On The Shoulders Of Giants

This is mostly the work of my colleague Brett Hartley, with input from Terrell and the iRODS team

wellcome sanger institute

Solution One: iphymv within a single subtree

"*Physically move a file in iRODS to another storage resource.*

Note that if the source copy has a checksum value associated with it, a checksum will be computed for the replicated copy and compared with the source value for verification."

(from the man page)

Solution One: iphymv within a single subtree - REJECTED

Issue 4010 - “repl to resource with existing replica does nothing”

“Nothing happens. Repl logic short-circuits resource plugins by detecting the good replica and determining that there is nothing to do.”

Solution Two: move resource out of hierarchy and then iphymv.

As a bonus, this would also stop new files being written to the resource!

Solution Two: move resource out of hierarchy and then iphymv - REJECTED

In 4.1.x the resource location is stored as a string for each object, e.g.

```
ils -l jc18_2G_20170710

jc18              0 root;replicate;seq-red;red4;irods-seq-i21-de   1744830464 2018-04-18.15:11
& jc18_2G_20170710

jc18              1 root;replicate;seq-green;green1;irods-seq-sr01-ddn-ra08-33-34-35 1744830464
2018-04-18.15:11 & jc18_2G_20170710
```

So every object would need an SQL UPDATE operation. We have hundreds of thousands of objects in each resource *and* it's a one-off, non-resumable operation.

wellcome sanger institute

Solution Two: move resource out of hierarchy and then iphymv - REJECTED

Also, we were slightly spooked by

#4402 - “renaming resource with substring affects all similarly named resources”

wellcome sanger institute

Solution Three: itrim everything off the resource, mark as down, then rebalance

Solution Three: itrim everything off the resource, mark as down, then rebalance

REJECTED

This leaves us with a period of time where each object only has 1 replica, which involves more risk than we were willing to accept.

Oh, and itrim *cowardly and unreasonably* refuses to trim below two objects, especially in a compound tree with two leaves below a replication resource

wellcome sanger institute

Solution Four: iphymv out of the composite resource, then back in

wellcome sanger institute

Solution Four: iphymv out of the composite resource, then back in

ACCEPTED!

wellcome sanger institute

Solution Four: iphymv out of the composite resource, then back in

ACCEPTED!

BUT!

Issue: 4212 - “iphymv doesn't move file in composite resource tree”

NOW we have Three Copies!

This could be something about our rulebase but...

Created by ElRatus from Noun Project

Created by QI studio from Noun Project Created by QI studio from Noun Project Created by QI studio from Noun Project Created by QI studio from Noun Project Created by QI studio from Noun Project Created by QI studio from Noun Project

wellcome sanger institute

Solution Four: iphymv out of the composite resource, then back in

trel commented on 10 Dec 2018 — Member

Whoa, that *is* exciting.

So, you targeted `root;replicate;red;red3;irods-seq-i18-fg` and you got a bonus replica in `green9` with a new timestamp...

wellcome sanger institute

Solution Four: iphymv out of the composite resource, then back in

So we need a way to address the three replicas - Brett scripted a tool using the python API (including adding functionality as merge request #162!)

Created by Juan Pablo Bravo
from Noun Project

Created by Qi studio from Noun Project
Created by Qi studio from Noun Project
Created by Qi studio from Noun Project
Created by Qi studio from Noun Project
Created by Qi studio from Noun Project
Created by Qi studio from Noun Project
Created by Qi studio from Noun Project

wellcome sanger institute

Solution Four: iphymv out of the composite resource, then back in

However, files are still being written to the resource, while we drain it.

Solution:

Set `minimum_free_space_for_create_in_bytes` (See Using free space check on unixfilesystem resources in the manual) to be slightly larger than the filesystem backing the resource. This ensures that no files can be written to the resource, even once it is emptied.

wellcome sanger institute

Solution Four: iphymv out of the composite resource, then back in

If you don't already have one, find a resource outside of the composite resource which is large enough to hold the largest file in the retiring resource.

Fortunately, we can use the demoResc's on the IRES's, since even the largest files are *only* 600GB At the moment*, as long as we're careful about parallelisation...

Solution Four: iphymv out of the composite resource, then back in

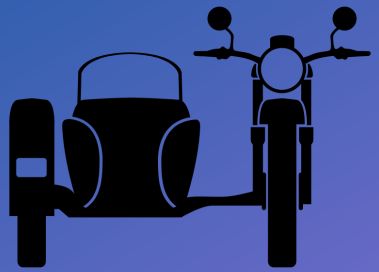
So for each file all we need to do is;

```
iphymv -M -S $retiringresourcehierarchy -R $outsideresource $file

iphymv -M -S $outsideresource -R root $file

irods-triple-replicas/triples.py $file

tee $file >> movedfiles.log
```

Solution Four: iphymv out of the composite resource, then back

Terrell came up with a one liner to do most all of this

(adjusted for an attempt at readability)



```
#!/bin/bash

SIDECAR="demoResc"

HIER_TO_BE_DRAINED="root;replicate;red;red3;irods-seq-i18-bc"

iquest "iphymv -M -S \"${HIER_TO_BE_DRAINED}\" -R \"${SIDECAR}\"
\"%s/%s\" && iphymv -M -S \"${SIDECAR}\" -R "root" \"%s/%s\"; echo
%s/%s > trimmedfile; irods-triple-replicas/triples.py -f trimmedfile;
cat trimmedfile >> movedfiles; rm trimmedfile" "select COLL_NAME,
DATA_NAME, COLL_NAME, DATA_NAME, COLL_NAME, DATA_NAME where
DATA_RESC_HIER = '${HIER_TO_BE_DRAINED}'"
```

wellcome sanger institute

Disclaimers:

1. We have tested this successfully on our development zone, but have yet to move production data.
2. No Yaks were harmed in the making of this talk

wellcome sanger institute

Thank you for ~~staying awake~~ listening!

Questions?

Credits!

Brett Hartley, ISG

Helen Cousins, ISG for the Yak Photo's *in-situ*

Terrell and the iRODS Team

[Baffalo by Qi studio from the Noun Project]

[Centaur by Eliricon from the Noun Project]

[Superhero by Juan Pablo Bravo from the Noun Project]

[Sidecar By DiabloTim, Oakland (from the Noun Project)]

[Two Yaks Photo by DDP on Unsplash]

wellcome sanger institute