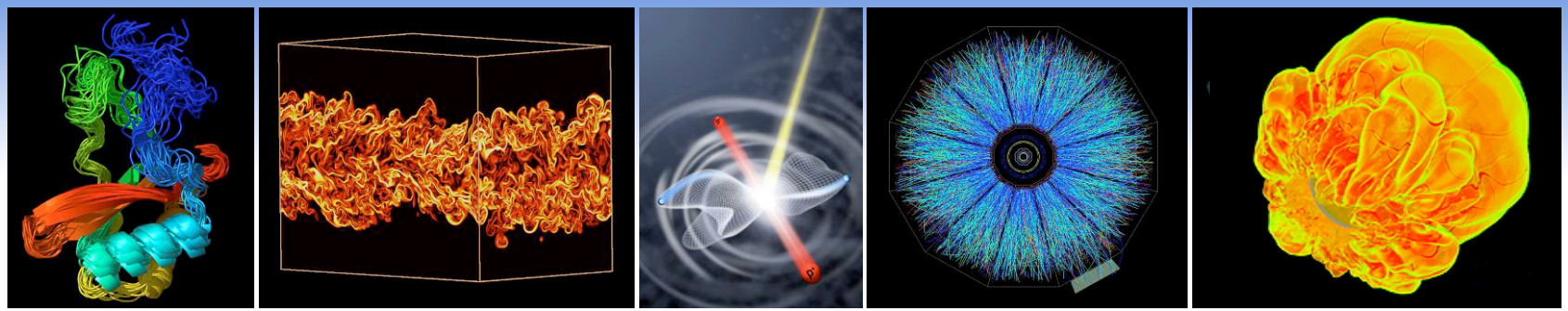


Reaping the Benefits of MetaData

Nicholas P. Cardo

cardo@nersc.gov

U.S. DEPARTMENT OF ENERGY

NERSC

BERKELEY LAB

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

"Strive for perfection in everything. Take the best that exists and make it better. If it does not exist, create it. Accept nothing nearly right or good enough."

Sir Henry Royce

co-founder of Rolls-Royce

It's a fact…

- **File Systems are getting bigger**
- **Data sets are larger**
- **File counts have increased**

	scratch	scratch2
Total Size	209 TB	209 TB
Space in Use	104 TB	62 TB
Inodes in Use	18 million	6 million

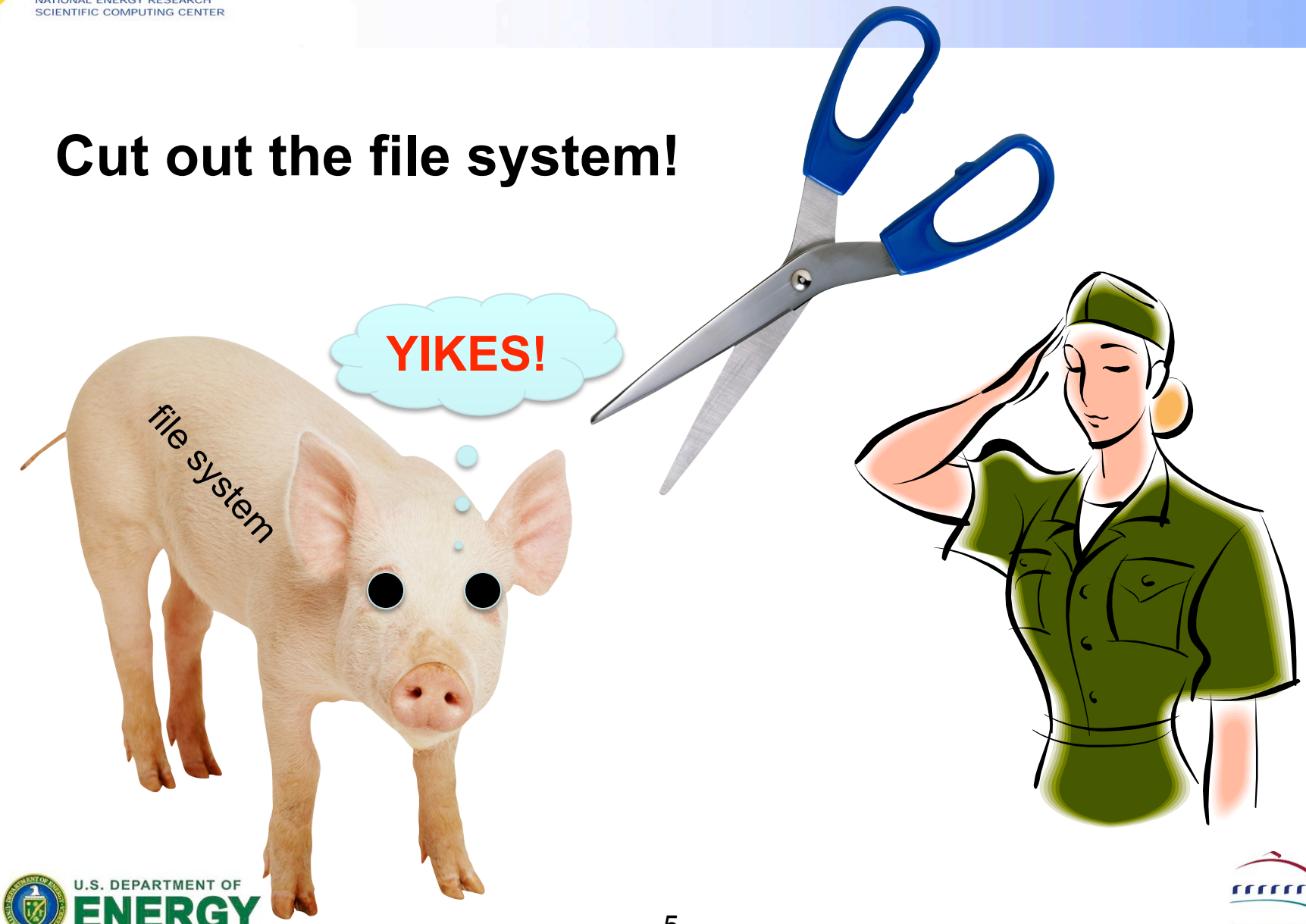
Snapshot taken on 4/9/10

The Problem…

NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

The Solution

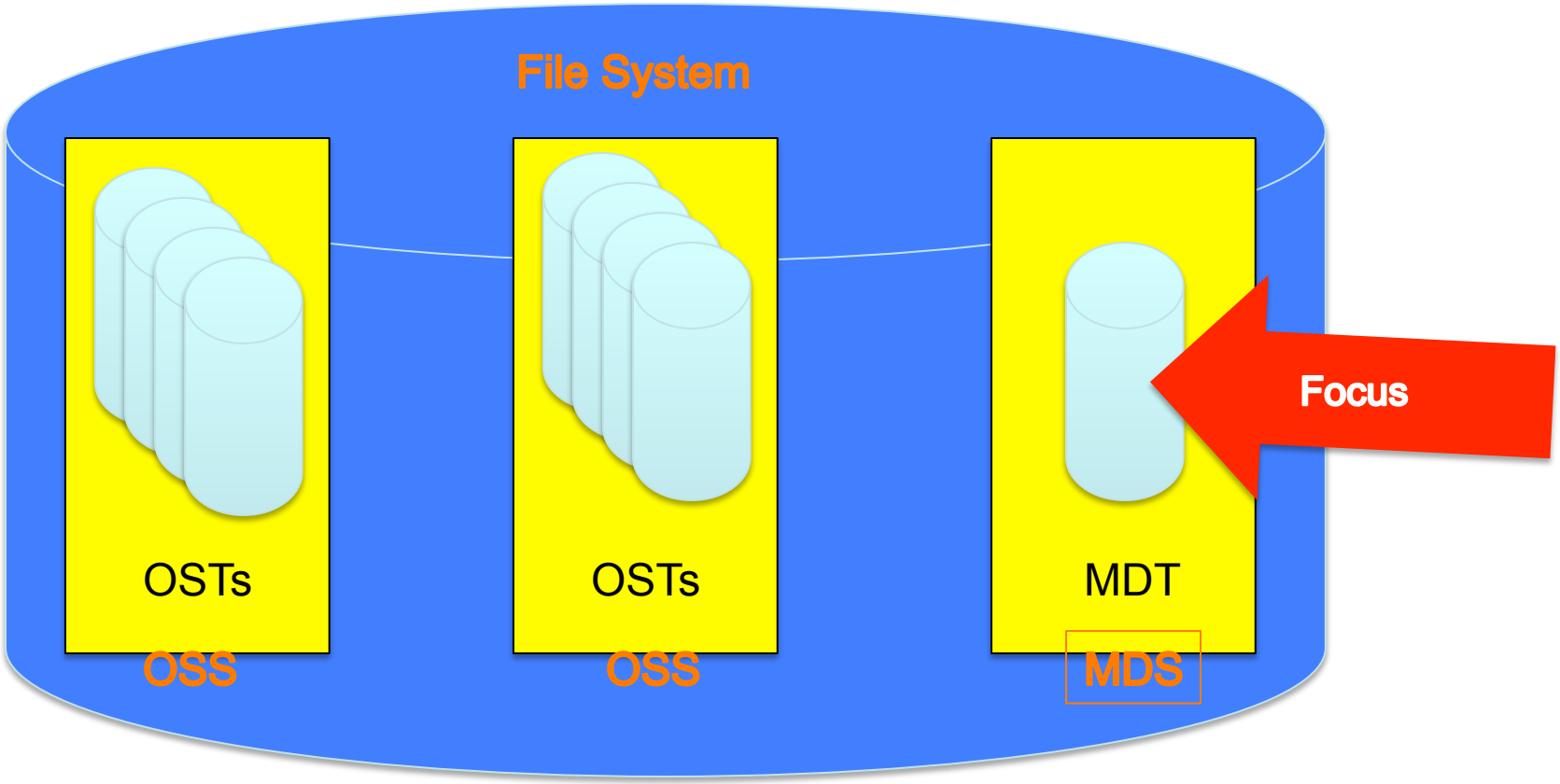
Cut out the file system!



U.S. DEPARTMENT OF ENERGY

BERKELEY LAB

Architecture



Metadata

Normal

- UID
- GID
- Atime
- Mtime
- Ctime
- Mode
- Size
- Inode number

Lustre Extended

- Object ID
- OST number

Well Really…

Its possible to access the metadata quickly and search it? ***Well, yes you can!***

How is this possible?
Modify e2scan to collect the data and scan routinely.

Here's What You Get!

header

```
#IDENT#|1|1.40.7.sun3-0suse|
1271070593|Mon Apr 12 11:09:53 GMT
2010|nid00336|/dev/sda|48|2|0|4194304|
scratch2|daily scan
```

data

```
1270054706|1269984003|1269984003|
18599|1018599|100600|0|159515275|
21:1a26ba2,17:19ebafe|./ROOT/
scratchdirs/cardo/on/.on.swp
```

footer

```
#complete#1271073186
```

Header

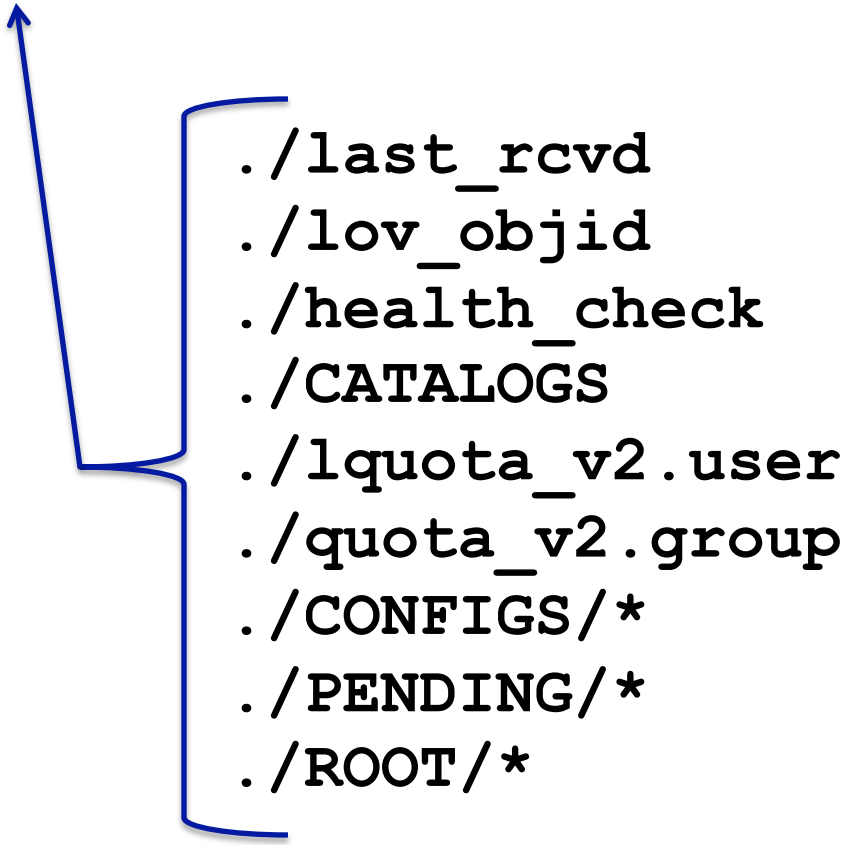
- #IDENT# ----------→ line label
- 1 ----------→ file format version
- 1.40.7.sun3 ----------→ e2fsprog version
- 1271070593 ----------→ timestamp
- Mon Apr 12 11:09:53 GMT 2010 -→ timestamp
- nid00336 ----------→ node name
- /dev/sda ----------→ scanned device
- 48 ----------→ total OSTs
- 2 ----------→ default stripe count
- 0 ----------→ default OST
- 4199304 ----------→ block size
- scratch2 ----------→ file system scanned
- daily scan ----------→ scan label

Data

- 1270054706 ----------→ atime
- 1269984003 ----------→ ctime
- 1269984003 ----------→ mtime
- 18599 ----------→ uid
- 1018599 ----------→ gid
- 100600 ----------→ mode (*octal*)
- 0 ----------→ size (*not really*)
- 159515275 ----------→ inode number
- 21:1a26ba2,17:19ebafe -→ OST:object

Pathname

```
./ROOT/scratchdirs/cardo/on/.on.swp
```



```
./last_rcvd
./lov_objid
./health_check
./CATALOGS
./lquota_v2.user
./quota_v2.group
./CONFIGS/*
./PENDING/*
./ROOT/*
```

Footer

- `#complete# ------→ line label`
- `1271073186 ------→ timestamp`

Wouldn't and API be nice?

- clearsearch -----→ wipe out previous search
- closefsdata -----→ close the file
- findrec ---------→ search
- getfield --------→ extract a field
- getfsident ------→ extract the header
- getnextrec ------→ get the next record
- getobject -------→ parse the objects
- getost ----------→ parse the ost numbers
- getrawrec -------→ return the raw unparsed record
- get-stripewidth -→ count the stripes
- openfsdata ------→ open the file
- printfsident ----→ human readable header
- printrec --------→ human readable record
- set-search ------→ add search criteria

What about a general utility?

```
fsfind  [-g  gid]  [-H] [-m modebits] [-o ost] [-p string]
        [-s stripes] [-u uid] [-P] [-V | -h] fsdatafile
```

`-g gid`	Search for files with group id gid ownership.
`-H`	Display results in human-readable format.
`-m modebits`	Search for files with specific mode bits, modebits, set.
`-o ost`	Search for files residing on the OST specified by ost.
`-p string`	Search for pathnames containing string.
`-s stripes`	Search for files with a stripe count of stripes.
`-u uid`	Search for files with user id uid ownership.
`-P`	Search for pathnames containing non-printable characters.

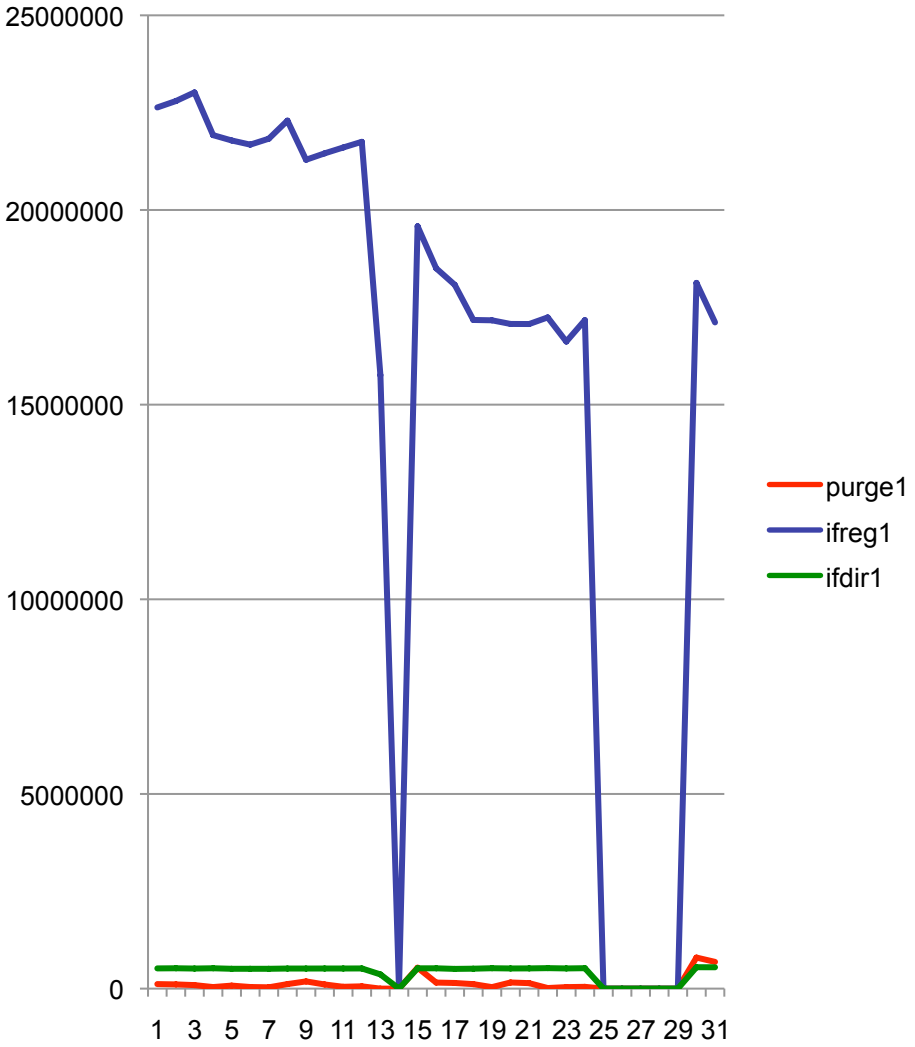
Enabling…

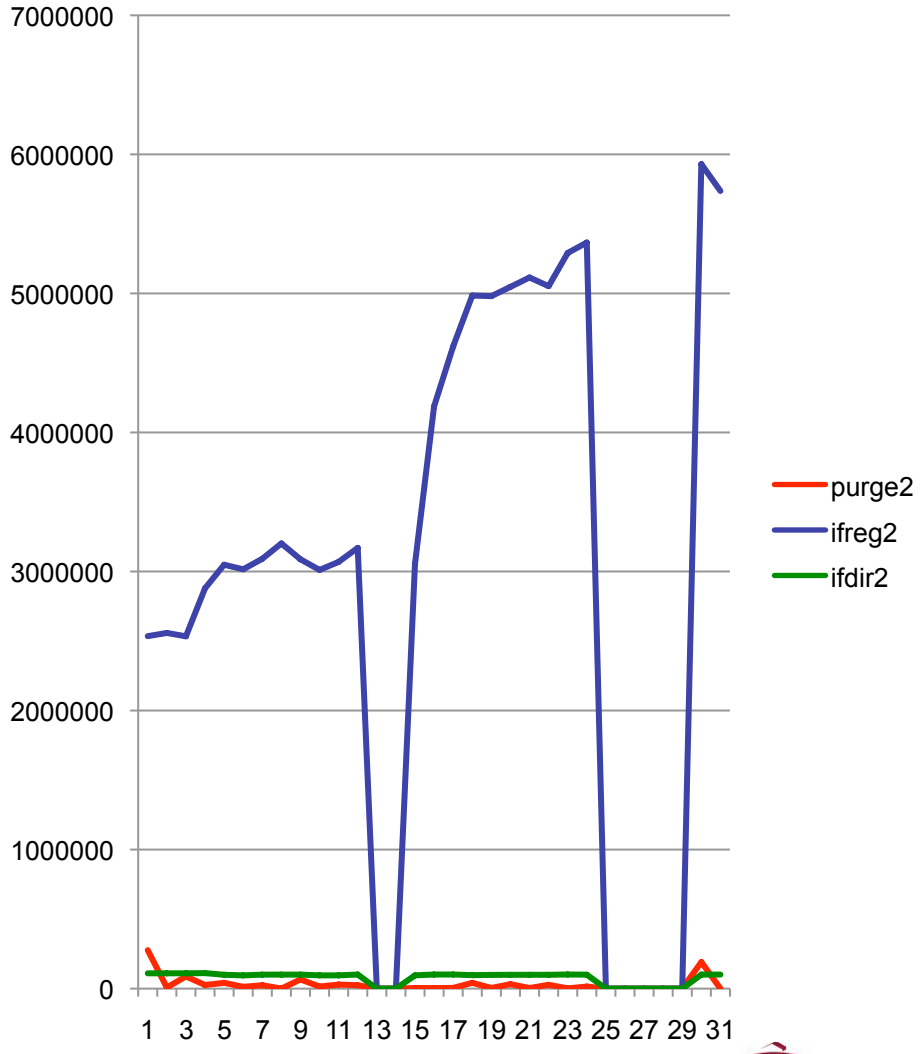
- Metrics
- Age based purging
- OST file identification
- Single stripe file identification
- Security checking
- File location

What We Do

- Scan daily (cron)
- Purge daily (cron)
- Routine OST diagnostic
- Routine security scanning

File System Metrics

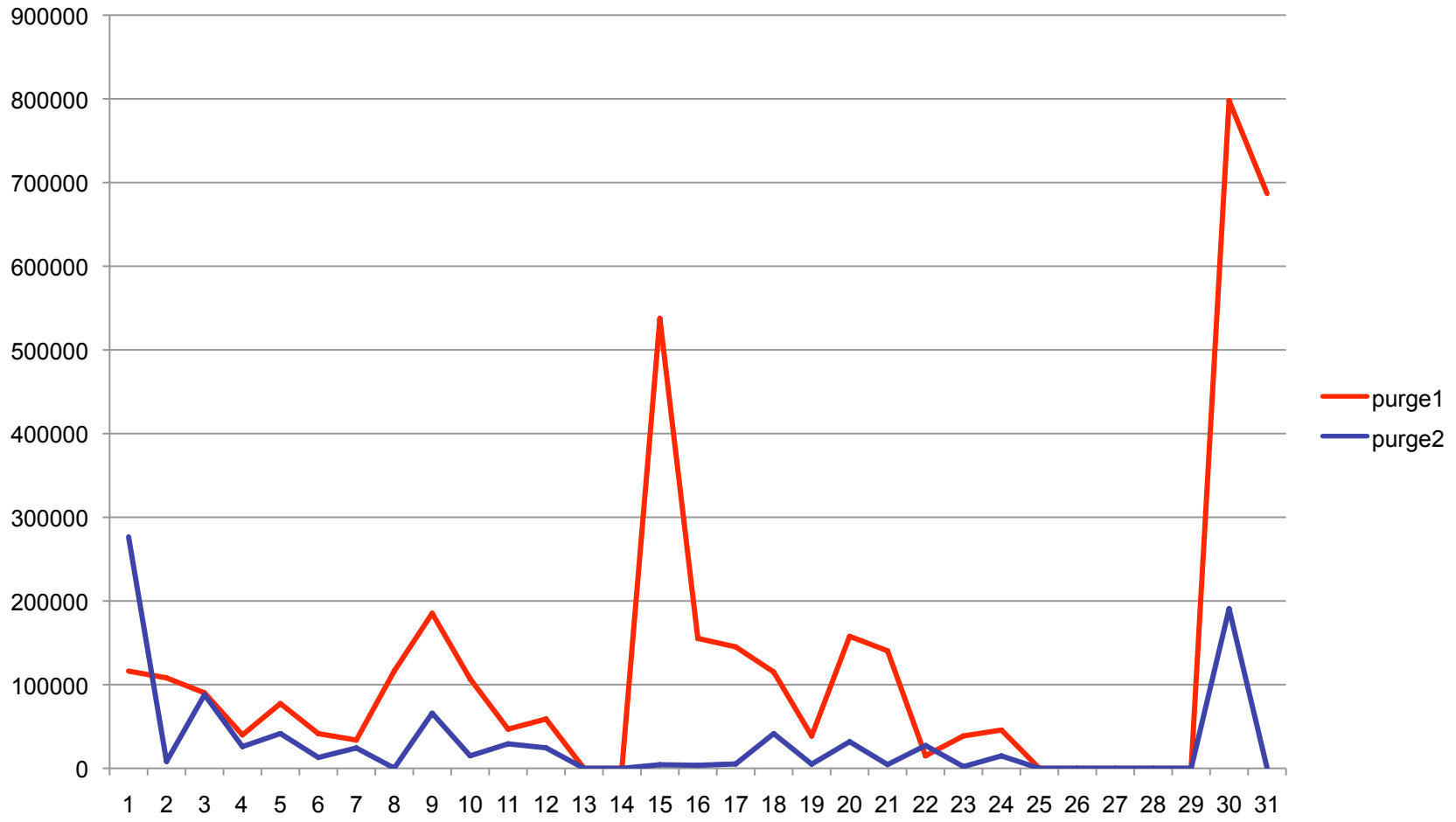




Hmmm, Purge anyone?

```
#  number of days old based on atime and mtime a file must be to be considered for purging.
PURGEDAYS       84

#  number of days old based on ctime a file must be in order to be considered for purging.
SAFEDAYS        7

#  Username to exclude.  Use only on nodes that can translate the username to UID properly.
#EXCLUDEUSER    username

#  UID to exclude.
EXCLUDEUID      18599 # cardo

#  Groupname to exclude. Use only on nodes that can translate the groupname to GID properly.
#EXCLUDEGROUP   username

#  GID to exclude.
#EXCLUDEGID     UID

#  Paths to exclude on scratch
EXCLUDEPATH     /scratch/scratchdirs/cardo
EXCLUDEPATH     /scratch/crayadm
EXCLUDEPATH     /scratch/tmp
EXCLUDEPATH     /scratch/backups
EXCLUDEPATH     /scratch/JobFiles
EXCLUDEPATH     /scratch2/scratchdirs/cardo
```

March Purge Statistics



How to Use the Data

```
fsfind -o 0 fsdata.scratch2.20100301
fsfind -s 1 fsdata.scratch2.20100301
fsfind -s 1 -o 0 fsdata.scratch2.20100301
fsfind -o 5 -o 6 -o 7 -o 8 fsdata.scratch2.20100301

fsfind -u 0 fsdata.scratch2.20100301
fsfind -m 4000 fsdata.scratch2.20100301
fsfind -p exploit fsdata.scratch2.20100301
```

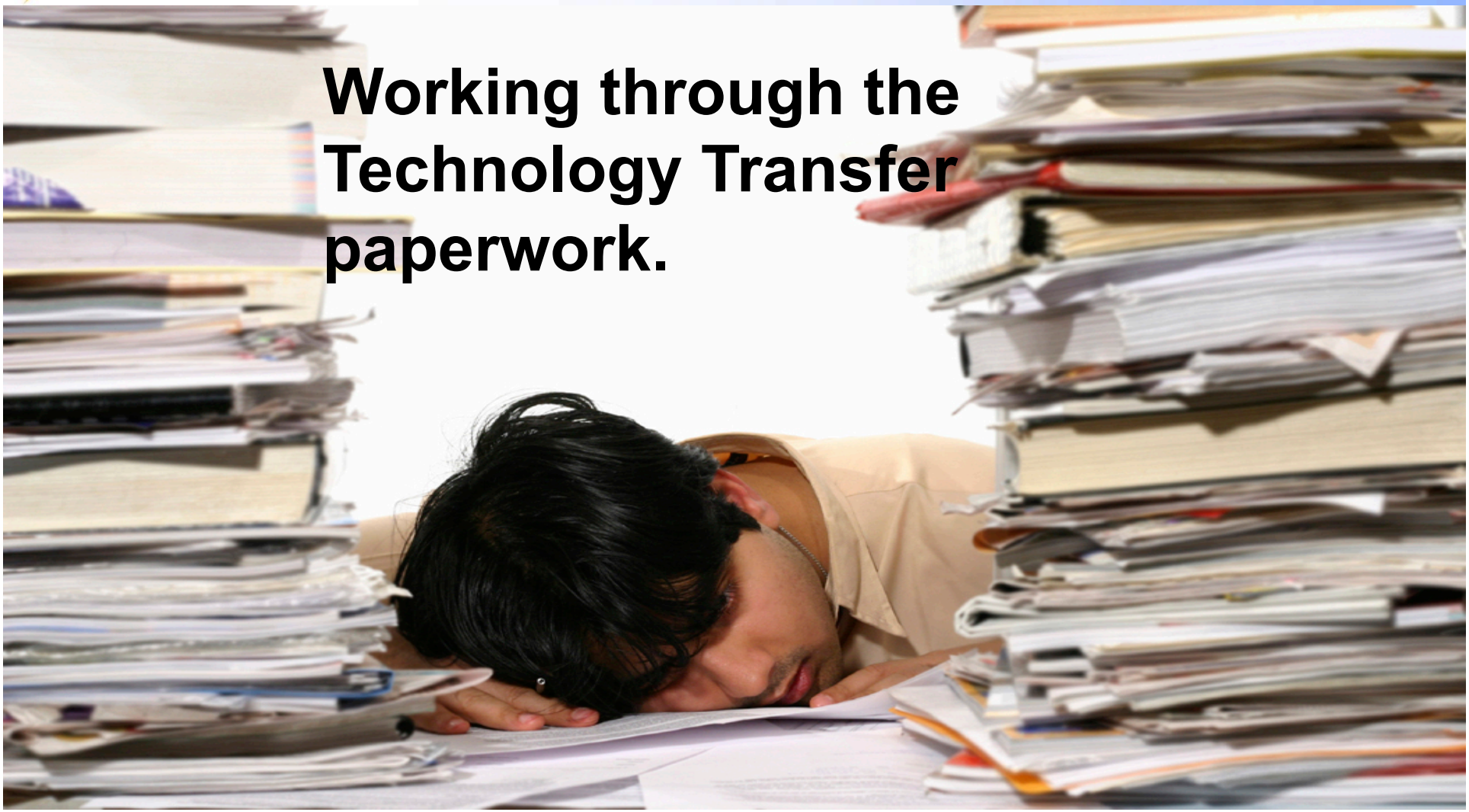
So what's next?

- **Provide options for generating and processing a binary file.**
- **Age based searching for fsfind.**

Where can I get ne2scan?

I welcome the opportunity to work with Lustre developers to get this modification into the mainstream Lustre e2scan.

What about the utilities?



Acknowledgements

- Cary Whitney @ NERSC
- Jason Hill @ ORNL
- _ _ _ _ K _ _ _ _ _ _ _ _ S @ _ _ B

NERSC
NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

Thank You!