IT PUTS THE CARD IN THE BOWL

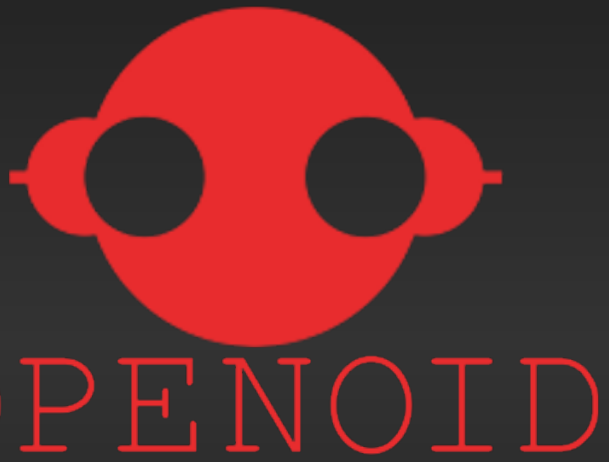
… if it wants a chance at a free copy of PoE!

Move over, rsync!

yet *another* reason you should be using ZFS

Jim Salter

Technomancer,
Mercenary Sysadmin,
Small Business Owner



Today's slides can be found at:

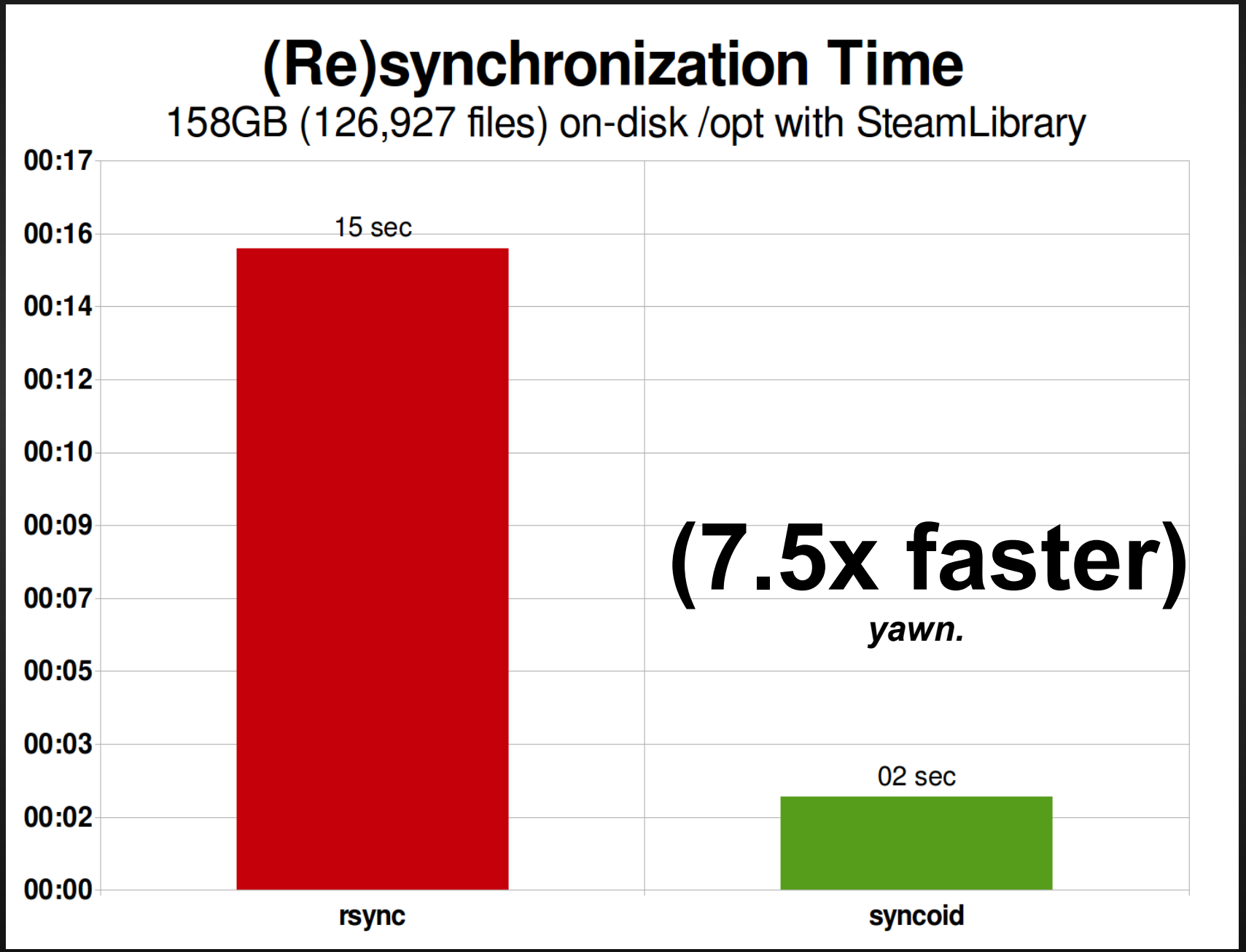
http://openoid.net/presentations/

i ♥ rsync!

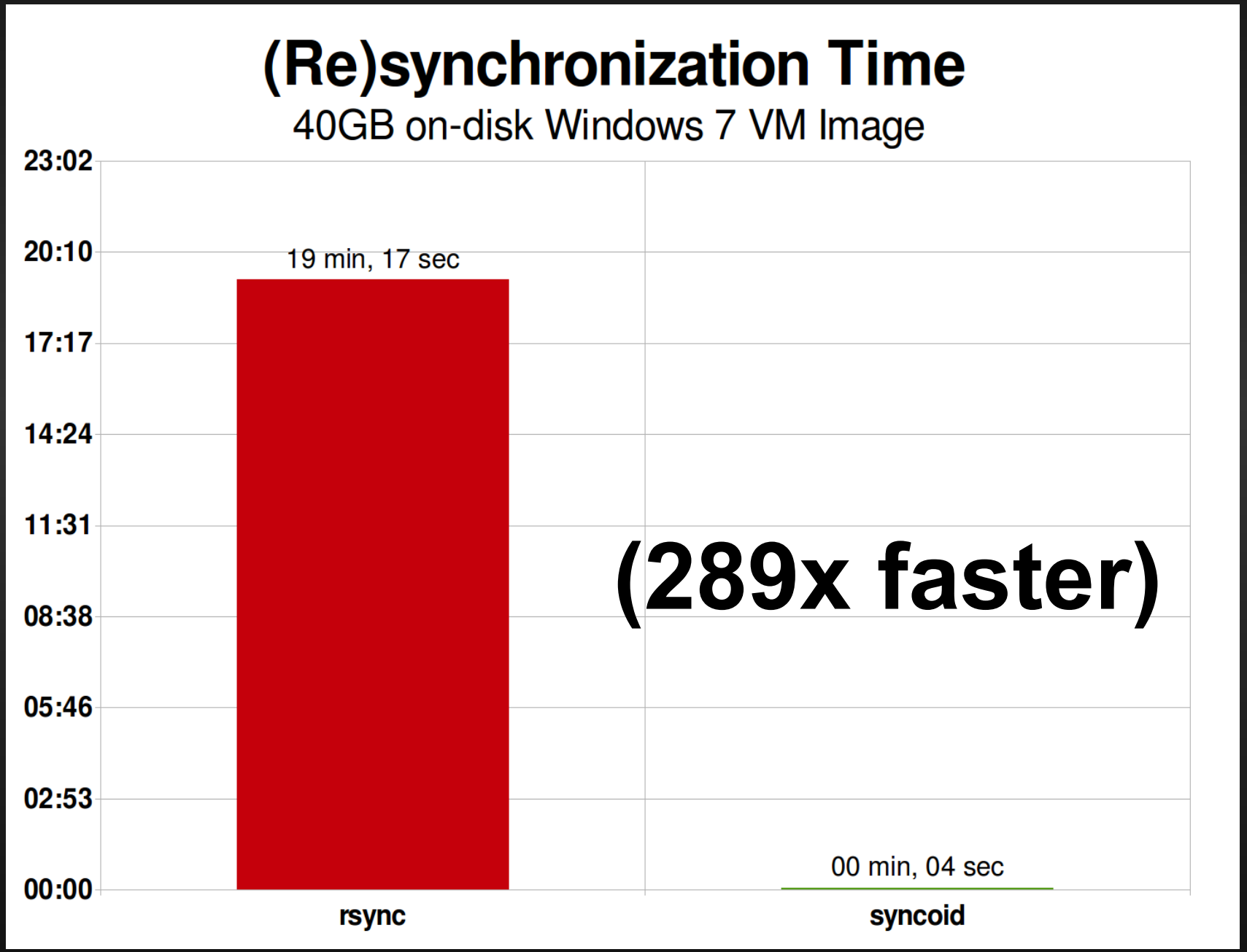
doodling its name in my Trapper Keeper since 1998

I come not to praise rsync...

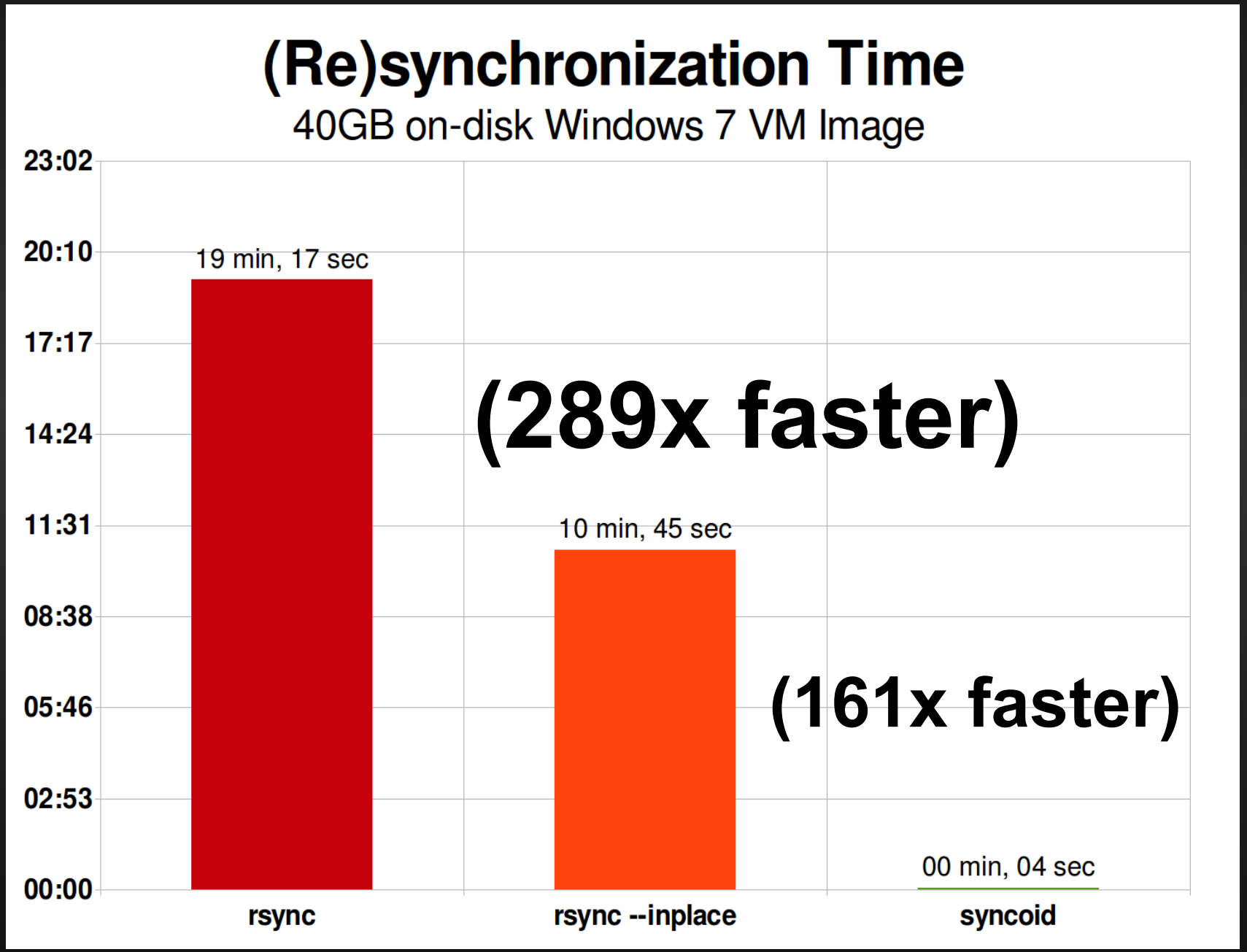
… but to bury it.



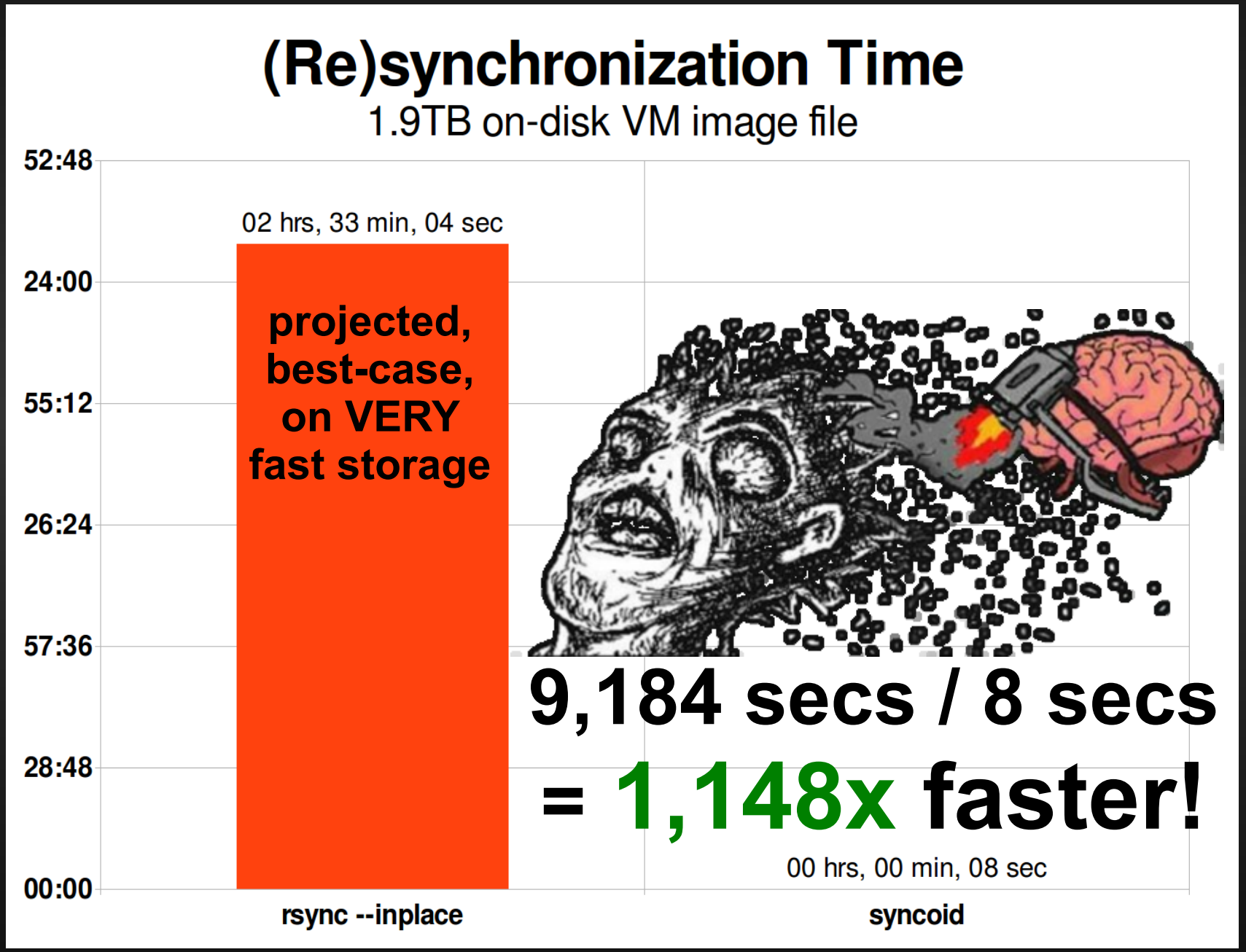
… bury it *deep*.


(Re)synchronization Time
40GB on-disk Windows 7 VM Image
23:02
20:10
17:17
14:24
11:31
08:38
05:46
02:53
00:00
19 min, 17 sec
00 min, 04 sec
rsync
syncoid
(289x faster)

… *really deep.*


(Re)synchronization Time
40GB on-disk Windows 7 VM Image
23:02
20:10
17:17
14:24
11:31
08:38
05:46
02:53
00:00
19 min, 17 sec
10 min, 45 sec
00 min, 04 sec
(289x faster)
(161x faster)
rsync
rsync --inplace
syncoid

3 *orders of magnitude* deep.



~~magnets~~ rsync: how does that work?

First Pass:

- Stat all files, comparing sizes and datestamps

Second Pass:

- target chunks changed files, hashes each chunk 2x

Third Pass:

- source compares simple hashes, compares MD5 hashes, then sends mismatched chunks to target

replication: how does *that* work?

First Pass:

- compare list of snapshots

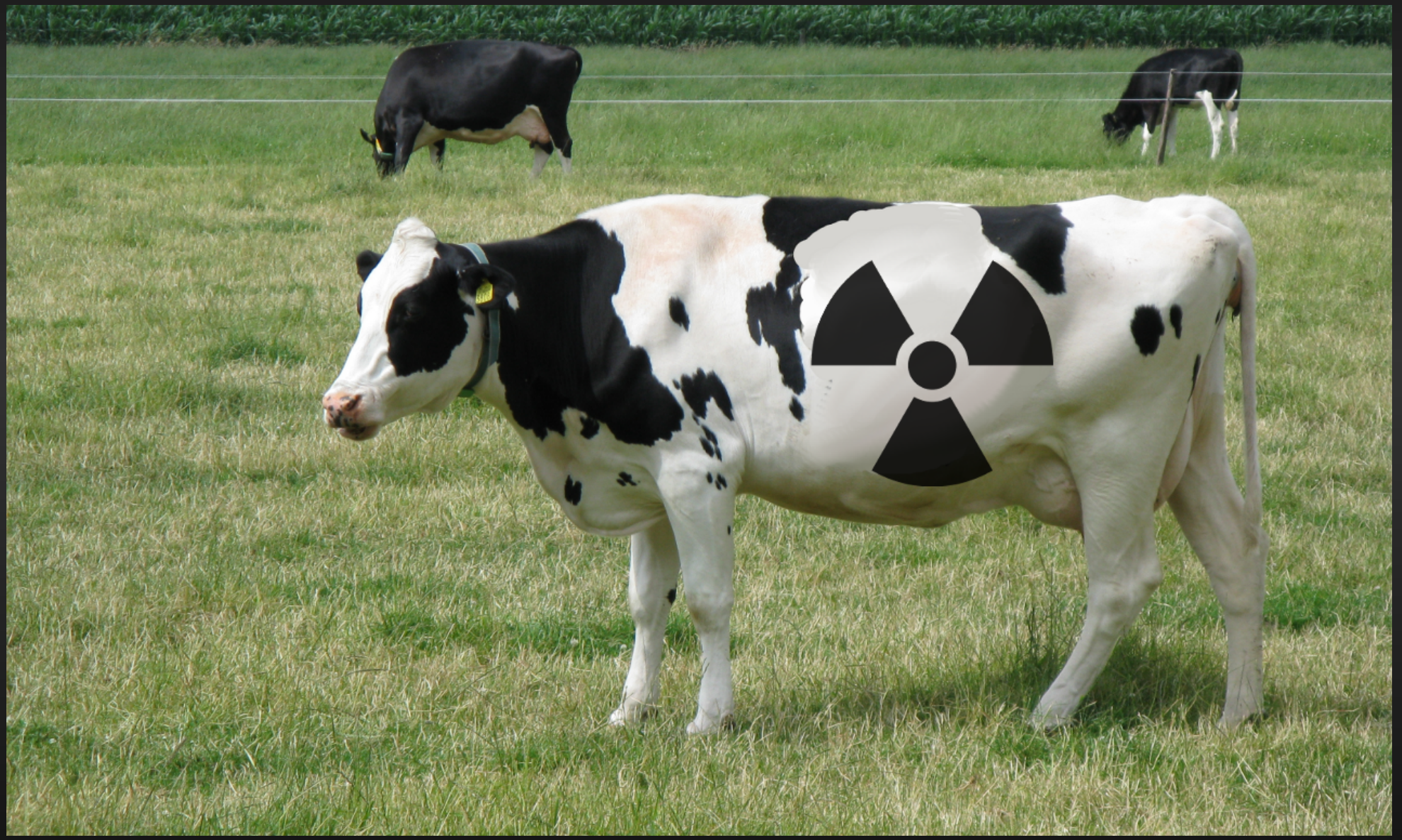
Second Pass:

- send blocks used only in missing snapshots

Third Pass:

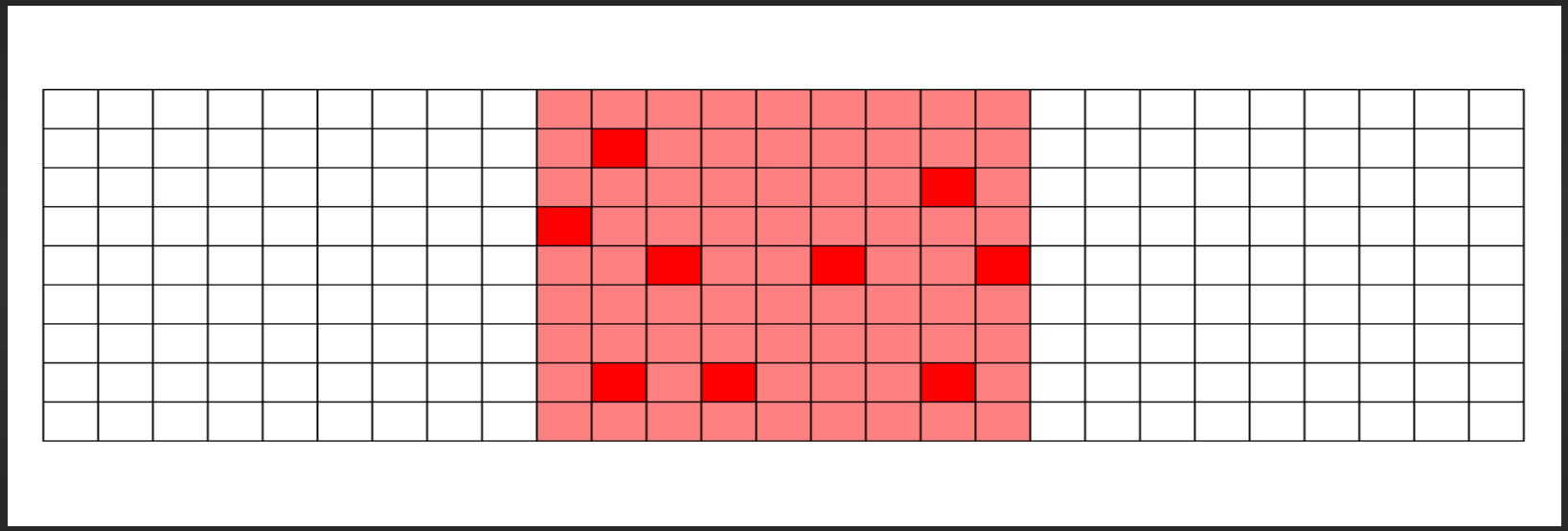
- drinking and laziness

learn the ways of the atomic CoW



Traditional FS

in-place modification of data is just what it sounds like



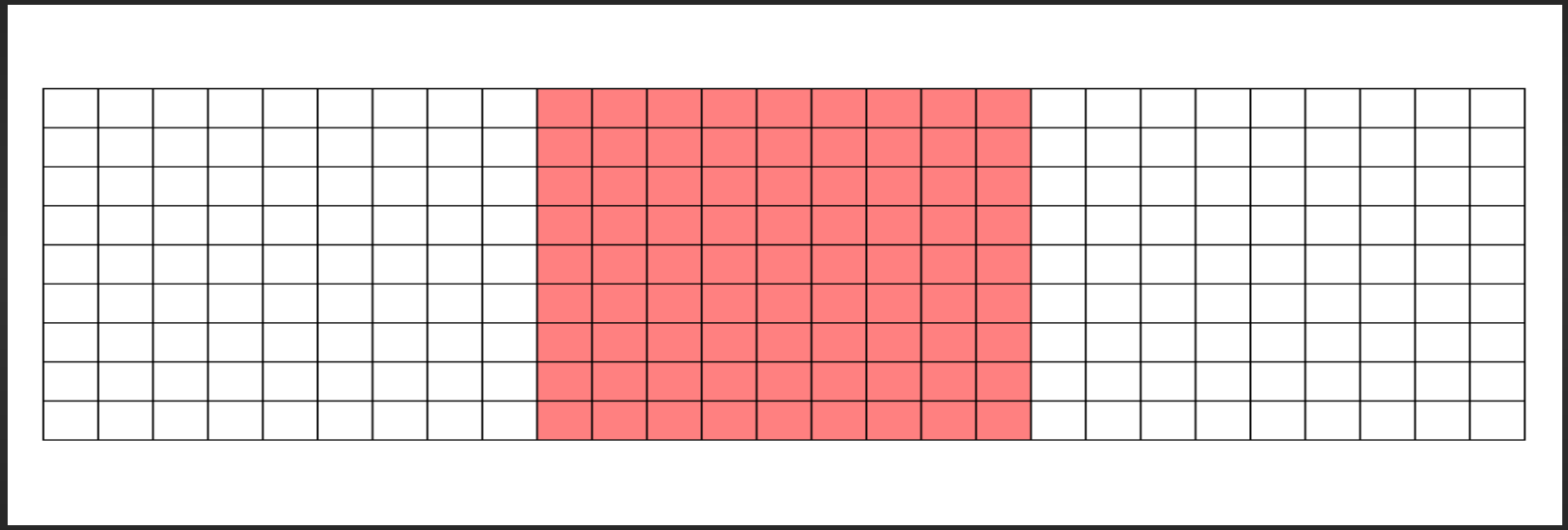
Dark red: newly (re)written data blocks

Pale red: existing data blocks

White: unlinked data blocks

Copy on Write FS

”the data comet” : write a new block, unlink the old block



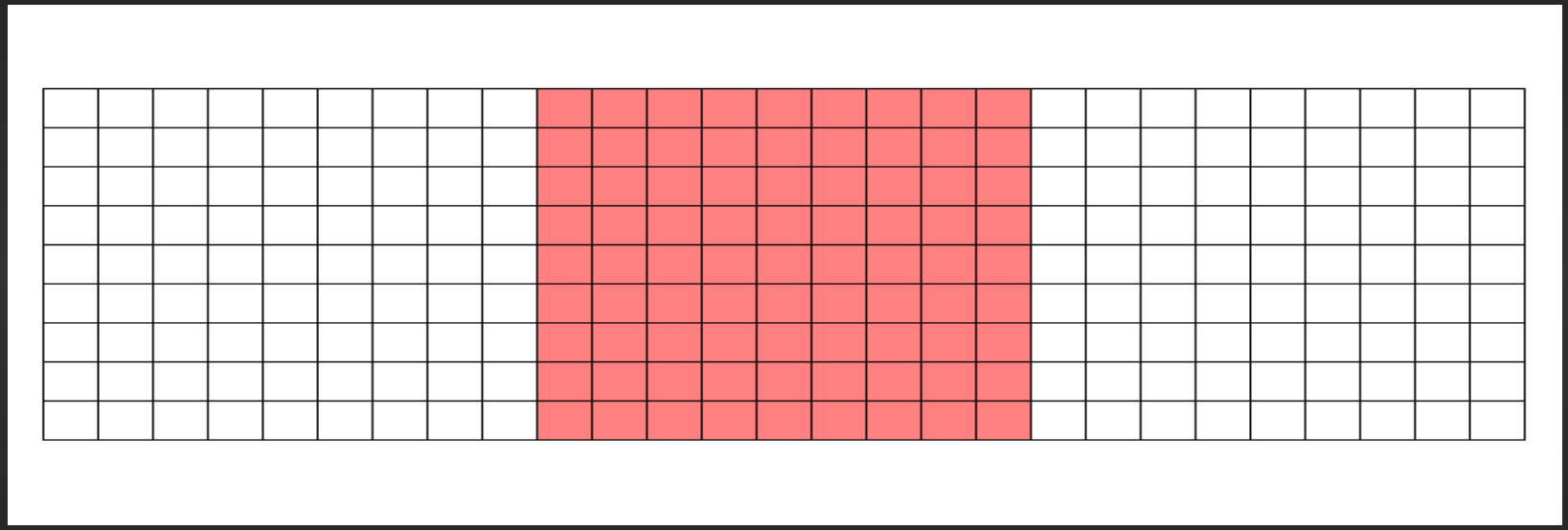
Dark red: newly (re)written data blocks

Pale red: existing data blocks

White: unlinked data blocks

Abstracting CoW

where the blocks are isn't important: “the data worm”



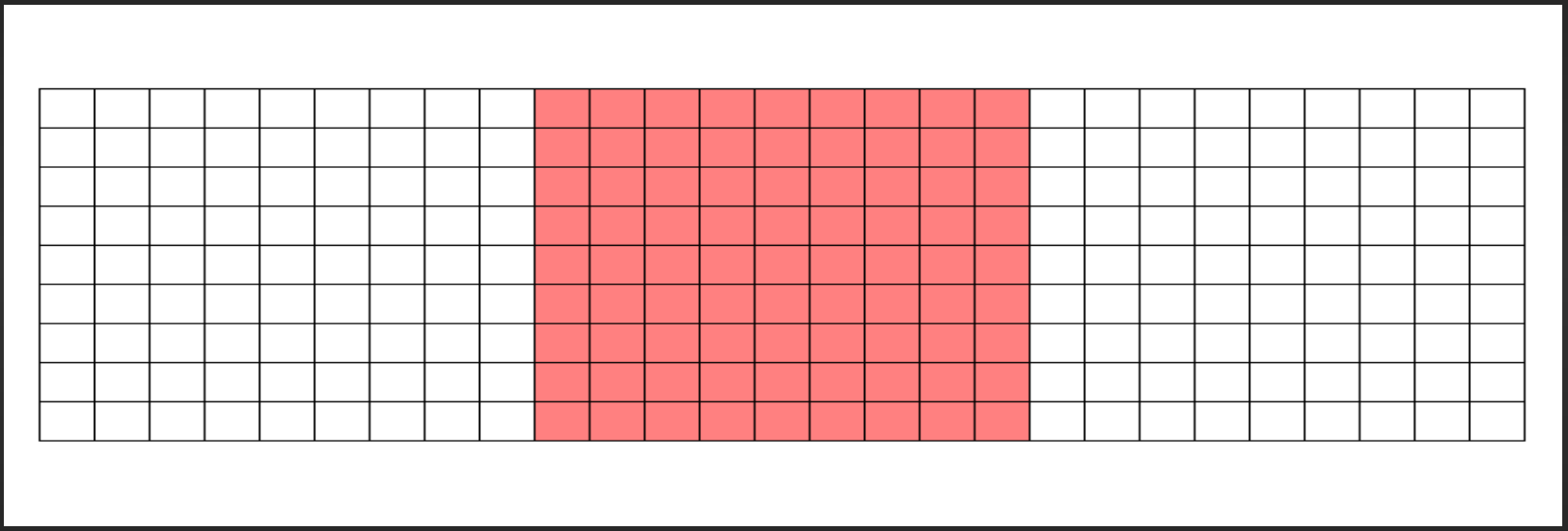
Dark red: newly (re)written data blocks

Pale red: existing data blocks

White: unlinked data blocks

Understanding CoW

visualizing “atomic CoW snapshots”



Dark red: newly (re)written data blocks

Pale red: existing data blocks

White: unlinked data blocks

Blue tint: snapshot @1

Yellow tint: snapshot @2

replication: the hard way

```
root@source: zfs snapshot dataset@1
root@source: zfs send dataset@1 \
             | ssh target 'zfs receive dataset'

root@source: zfs snapshot dataset@2
root@source: zfs send -i dataset@1 dataset@2 \
             | ssh target 'zfs receive dataset'
```

present but not shown:

donkeywork. *lots* of donkeywork.

replication: the easy way

```
root@source: syncoid dataset root@target:dataset
Sending incremental older ... newer (~ 276.1 MB):
219MB 00:04 [92.1MB/s] [========>  ] 79% ETA 00:01
```

http://sanoid.net/

present but not shown:

recursion, compression, network buffering, snapshot creation...

something something HIPAA SOX datacenter argle bargle flurg?

… but seriously: if you need that kind of thing, $60/TB/mo or less is pretty sweet.

DO YOU HUNGER FOR MORE?

Google: jim salter site:arstechnica.com

(Coming soon: a review of rsync.net's ZFS replication target offering)

Blog(s): http://jrs-s.net

http://openoid.net/blog

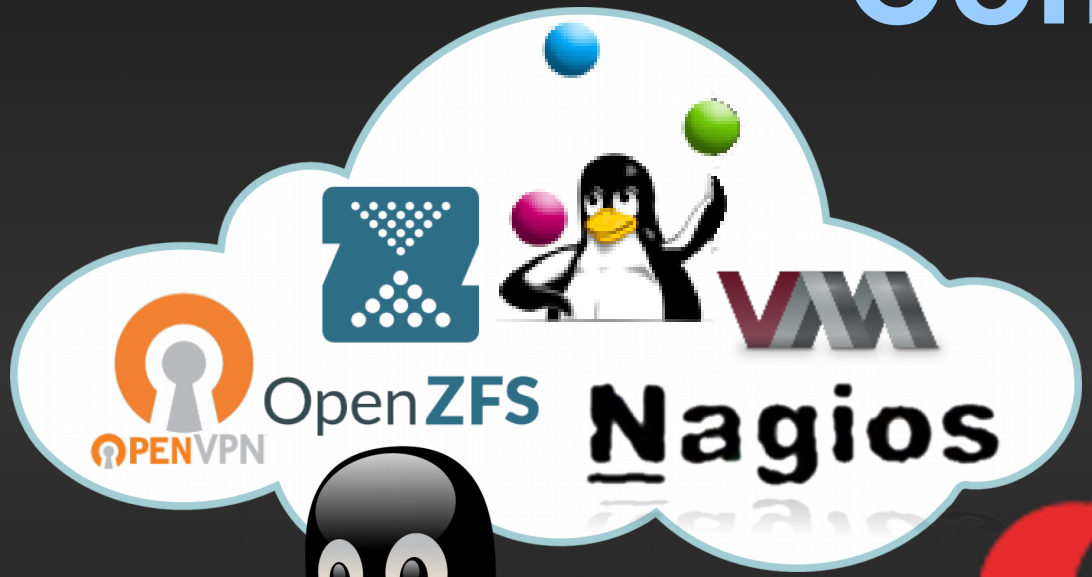
Twitter (lol): @jrssnet

(you're in luck, because I basically never shut up.)

Questions?

Comments?

Angry denunciations?



OPENOID

RAFFLE TIME!