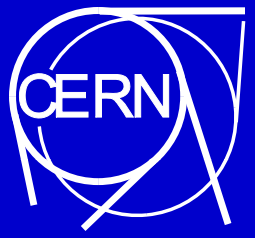


The Unbearable Slowness of Tape

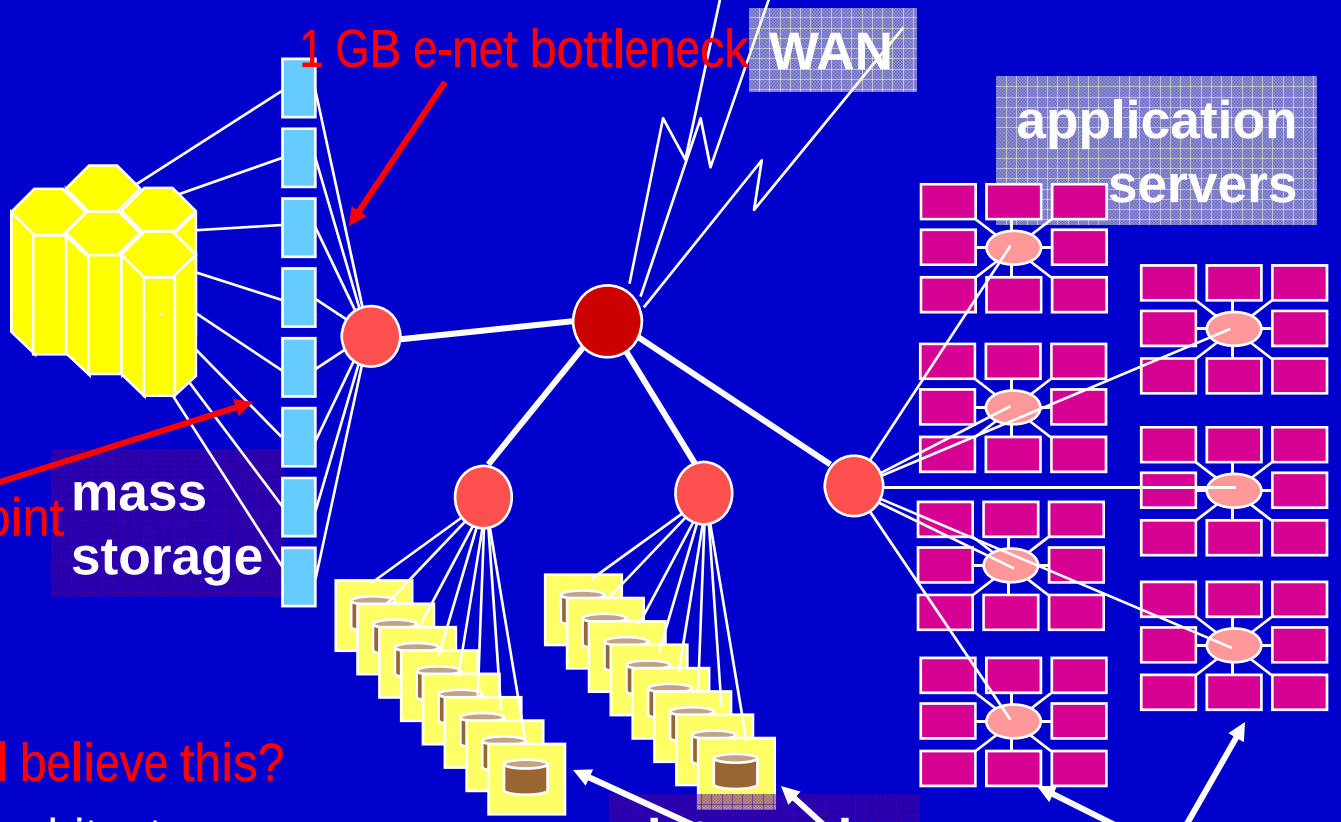
C. Curran, CERN

HEPIX

CERN, May 2008 (version 6.5.2008 10h00)

CERN Storage Model for ~15 PBs/yr

Three layers of 'separated functions' still exist in 2008



Does anyone still believe this?

simple, flexible architecture

- easy to integrate mass market components
- easy evolution to new technologies

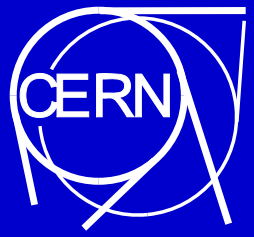
Storage: Sun, IBM libraries

513
16K

613
20K

513
12K

Lots of high quality equipment



The CERN 'tape layer' is still cheaper than disk

Can constrain media costs, robotics growth: Using IBM/Sun top-line drives..

Year	Slots	Media	Drives	PB	LHC	Robot	Media	Drives
2008	47 K	JB/T1	120	21	~30 PB	0	1.0	0
2009	57 K	JB/T1	120+	57	~45	0.5	1.0	2.5
2010	57 K	JB/T1	120	57	~60	0	0	0
2011	57 K	JC/T2	240	114	~75	0	6.0	5.0
2012	57 K	JC/T2	240	114	~90	0	0	0
						0.5	8.0	7.5

It looks 'a bit tight' in 2008, 2010

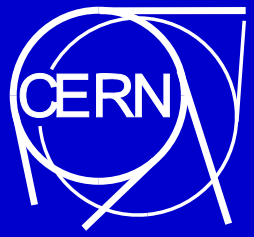
2009 Robots full-sized, add 1 3584

+ IBM, Sun drives upgraded (guess/hope!)

JC/T2 media change (guess/hope!)

Costs in MFS, total 16 MFS

'Notionally' media 100 FS, drive or upgrade 25 KFS, slot 50 FS



Fall back plan
LTO view
(IBM/Sun slow to GA, doesn't GA, get bought..)

Year	Slots	Media	Drives	PB	LHC	Robot	Media	Drives
2008	47 K	JB/T1	120	21	~30 PB	0	1.0	0
2009	57 K	LTO4	120	46	~45	0.5	5.7	1.8
2010	57 K	LTO5	120	92	~60	0	5.7	1.8
2011	57 K	LTO5	240	92	~75	0	0	0
2012	57 K	LTO6	240	184	~90	0	5.7	3.6
						0.5	18.1	7.2

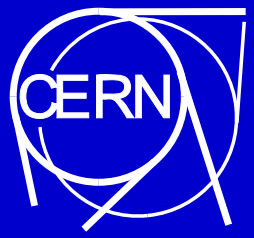
It also looks 'a bit tight' in 2008

2009 Robots full-sized, add 1 3584

Costs MFS, total 25.8 MFS (if LHC is late, is LTO4 needed? Saves 8 MFS)

'Notionally' media 100 FS, drive 15 KFS, slot 50 FS

T10000 / 3592 class drives still have the advantage



Drive is typically capable of single stream, 100 MB/s

We have ~120 top-grade drives already

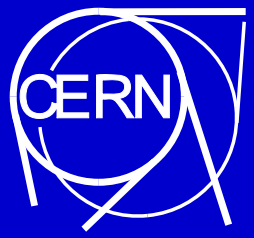
So why don't we see 10-12 GB/s?

So why do we still mount ~15,000 tapes per day?

Why is there so much incredibly ineffective READ activity?

CASTOR only 97 M files, 12 PB (TSM 1.5 B)

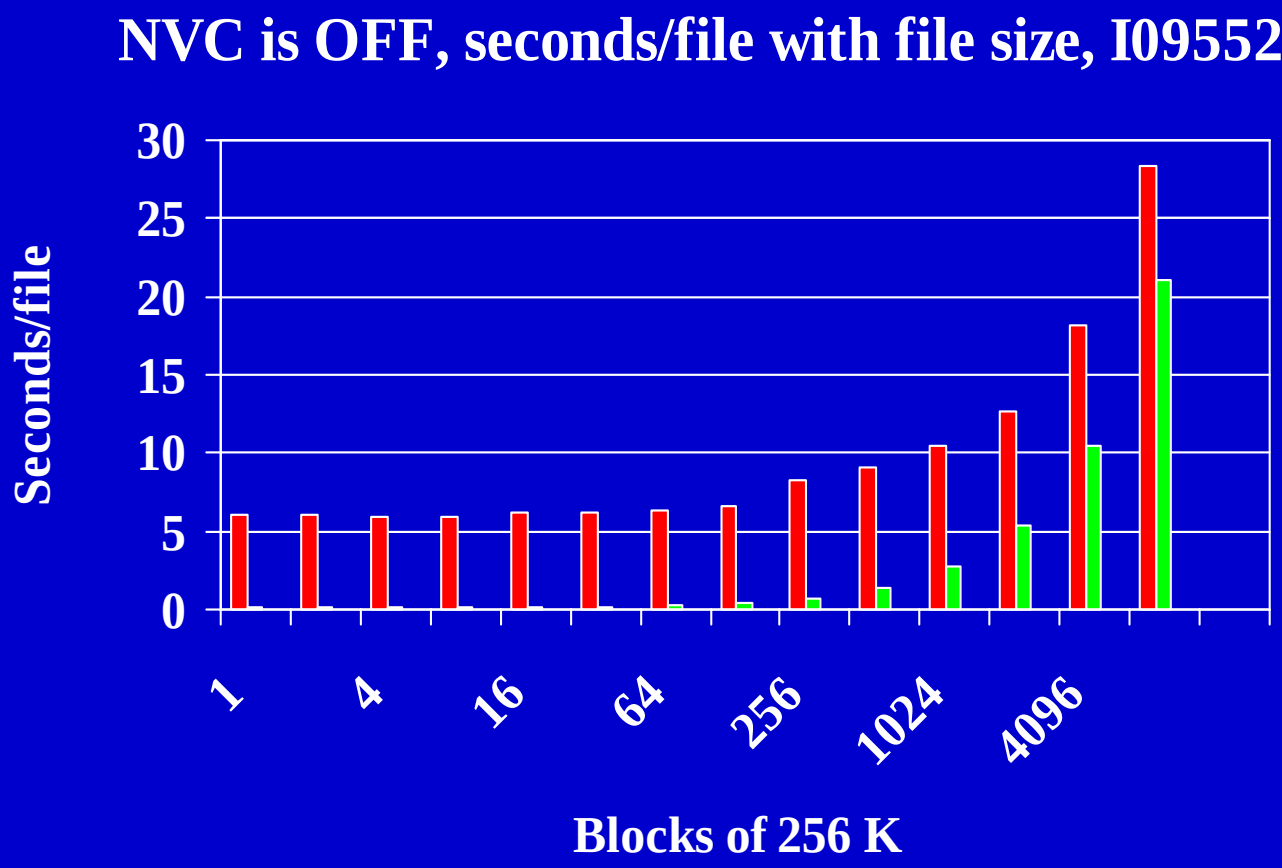
Why is it all so dreadful?

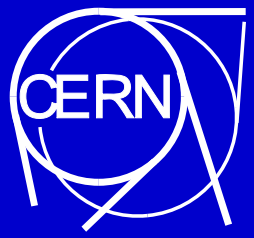


It's (y)our fault

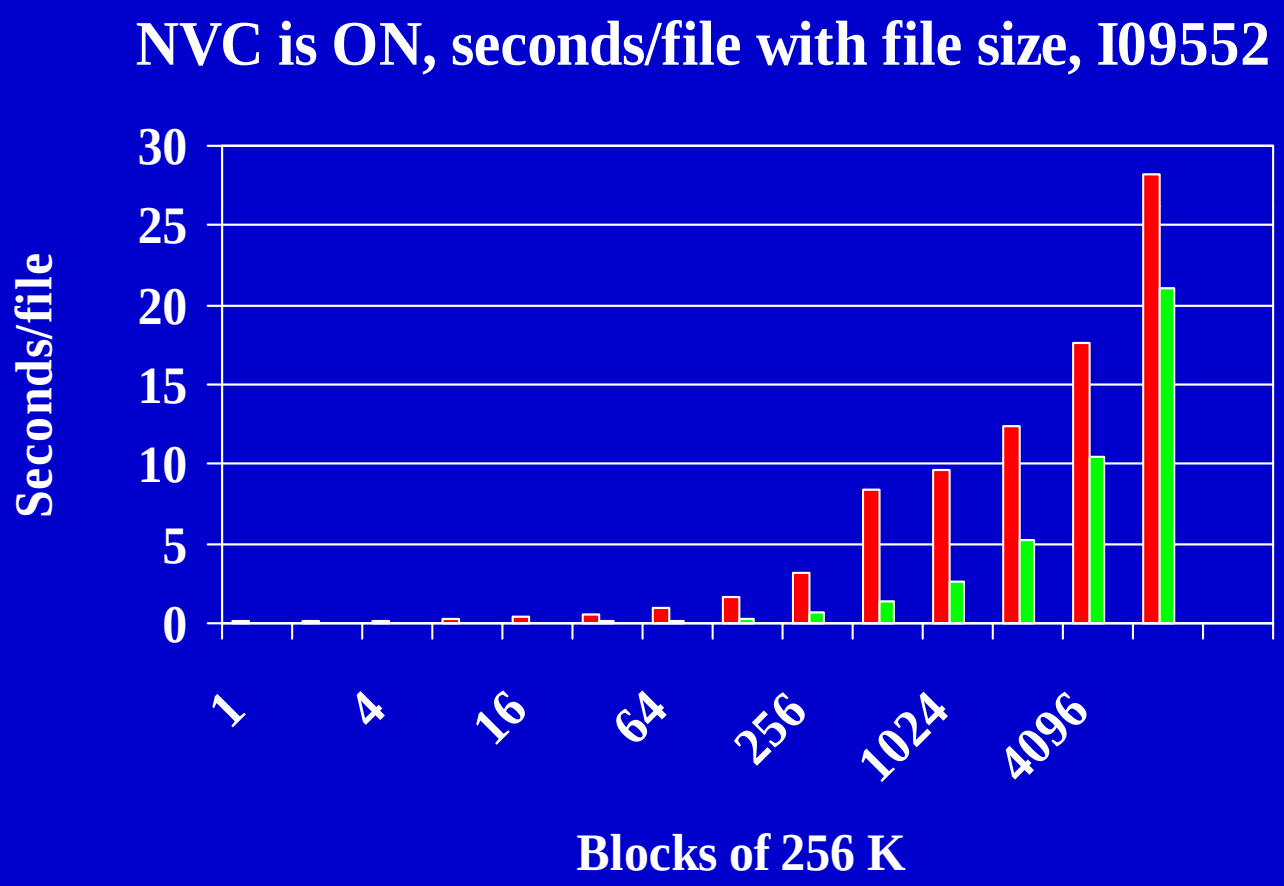
We need to look at some characteristics of this type of drive
CASTOR presently ignores most of them…

File process rate (ANSI file write/read)
Similar timings for T10000A, LTO4..
~ 5 s per ANSI labelled file



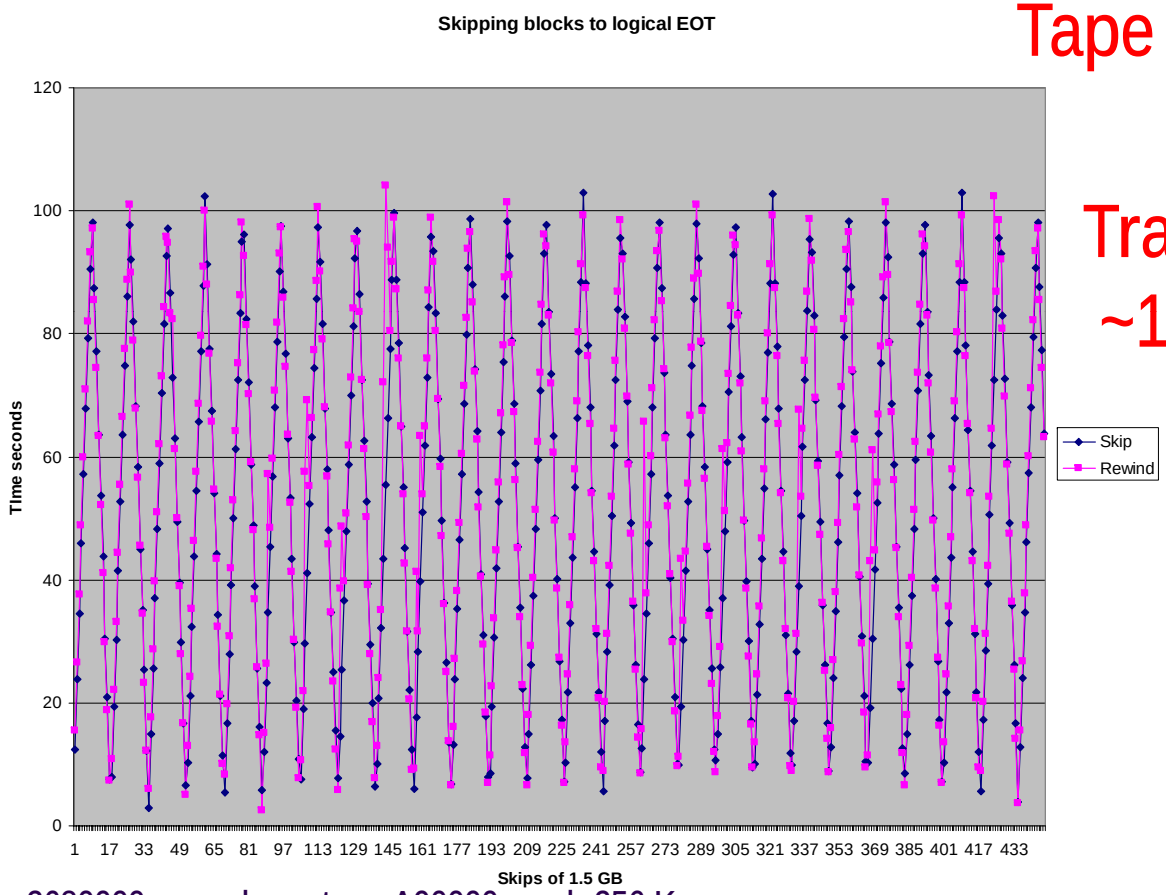


File process rate (ANSI file write/read)
IBM has a helpful trick for small files
Sun, LTO do not



Using all the tape in a cartridge: record position/rewind times, IBM 3592

Tape is ~90 s long



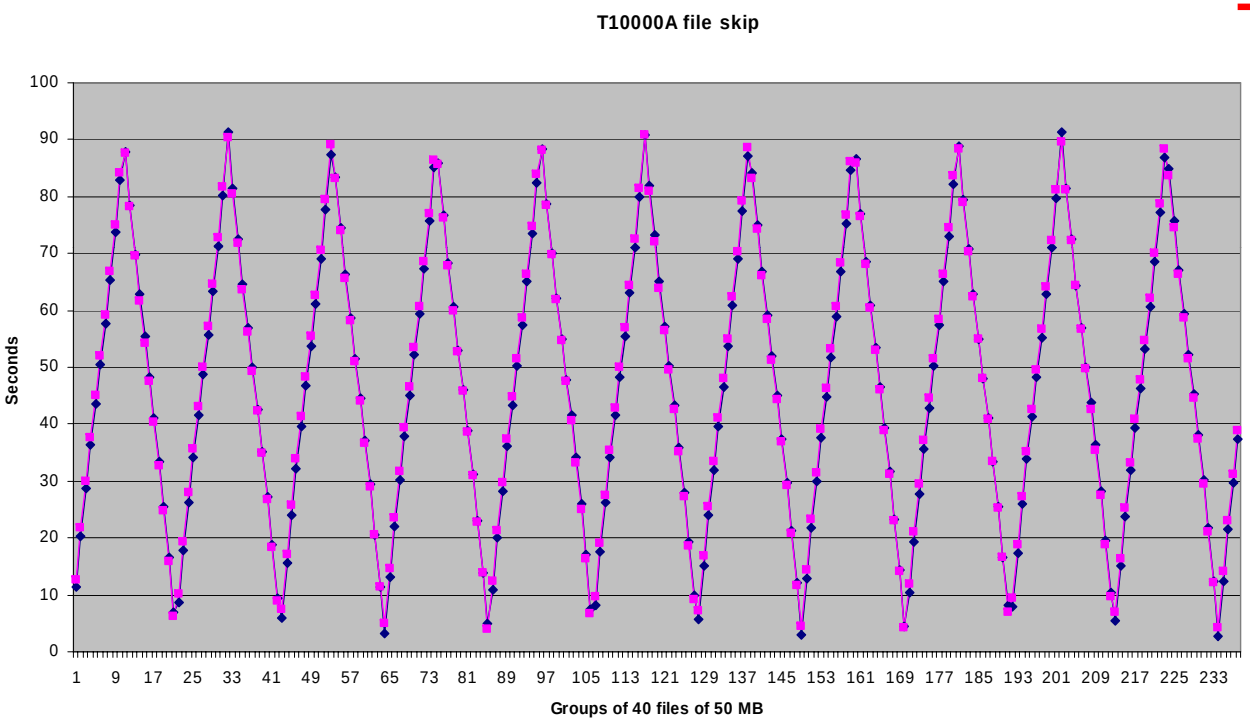
Track in/out is ~13 GB long

~2680000 records on tape A90000, each 256 K.

Here ~51 of 56 track sets are visible, in write to logical EOT mode, ~706 GB

But setting byte 5 of mode page 37 to ‘1’, so write is to physical EOT, ~764 GB

Using all the tape in a cartridge: ‘labelled file’ position/rewind times, T10000A



Tape is ~90s long

Track in/out is ~23 GB long

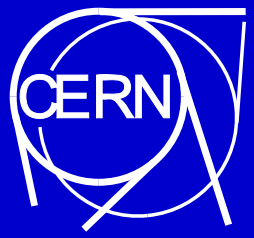
} This is effectively a mid-point tape load Region, bigger than 9840:
Good place to mount or dismount

‘user’ files on tape, each ~50 MB (200 blocks of 256K), simulated labels

Here only ~22.5 of 24 track sets are visible, so only 500 of potential 534 GB ‘allowed’

Command mt -f /dev/nst0 fsf n

Command mt -f /dev/nst0 rewind

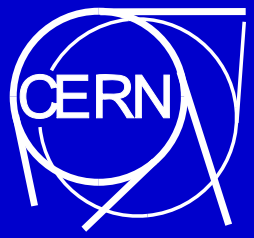


The case of WRITE badly… part 1

Starts well: accumulate ~100 GB for the CASTOR migrator
Current file size average in CASTOR is ~ 100 MB!
LHC, maybe 1 GB, so ~ 100 files?

Choose a tape (correlation to 'file style'?)
Choose a drive (correlation to tape location?)

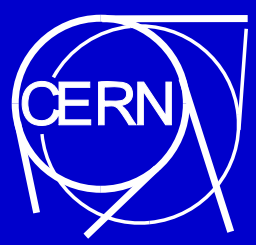
Mechanics start to lumber into life to do this random mechanical shuffle….
IBM: ~50 s dismount, ~50 s mount (dual accessor, but..)
Sun: ~150 s dismount, ~150 s mount (vertical, horizontal..)



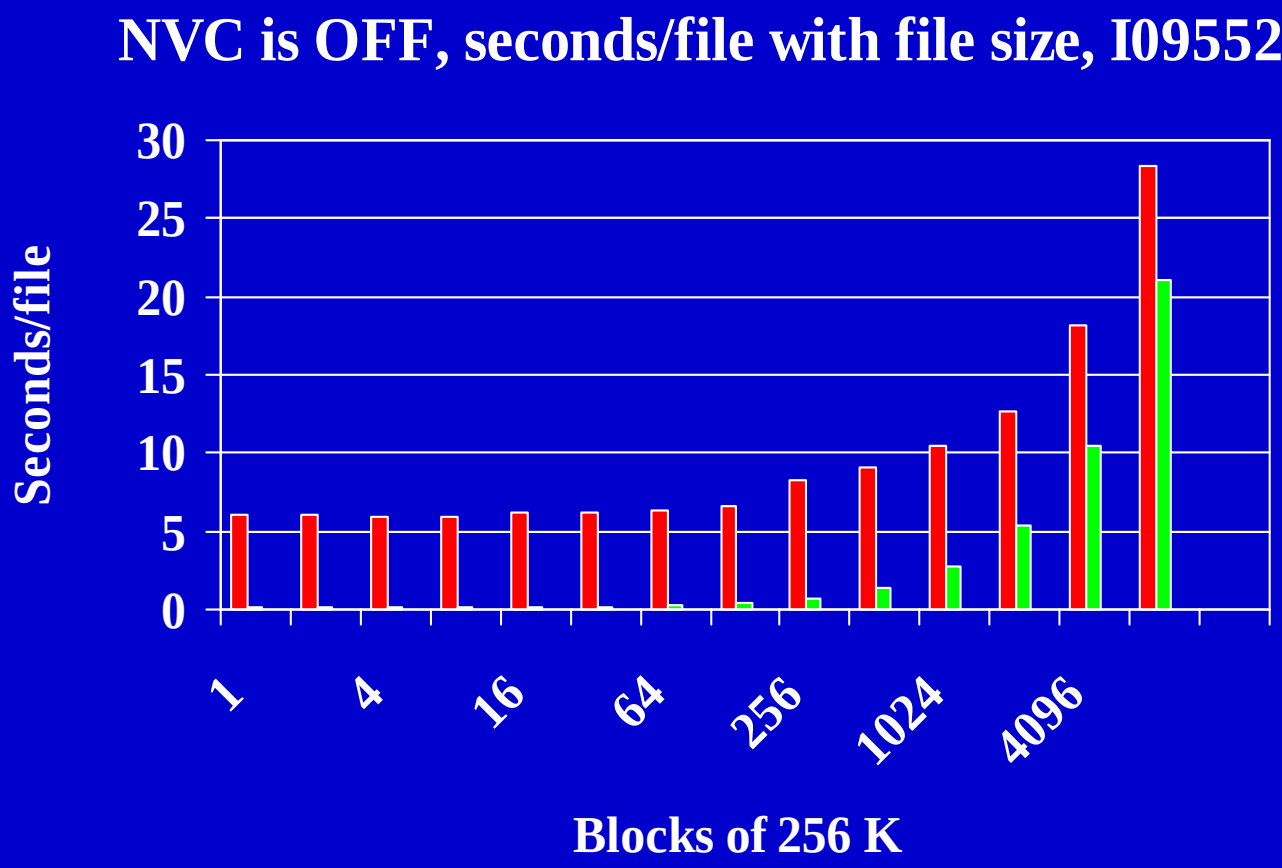
The case of WRITE badly… part 2

Drive starts to lumber into life to do this WRITE of 100 1GB files
It's connected by Gbit ethernet to a shared disk server
You'll struggle to get near 100 MB/s

IBM, Sun, LTO: ~ 20 s thread
Where's EOD? ~ 45 s winding forward
Write 100 x 1 GB files: every labelled ANSI file, ~ 5 s
Rewind? ~ 45 s rewinding to BOT
Unloading, ~20 s
Moves, thread, seek, ANSI labels, rewind, unload:
~ 730 s IBM, ~ 930 s Sun
Data writing: ~ 1000 s 'Useful': 58% IBM, 52% Sun

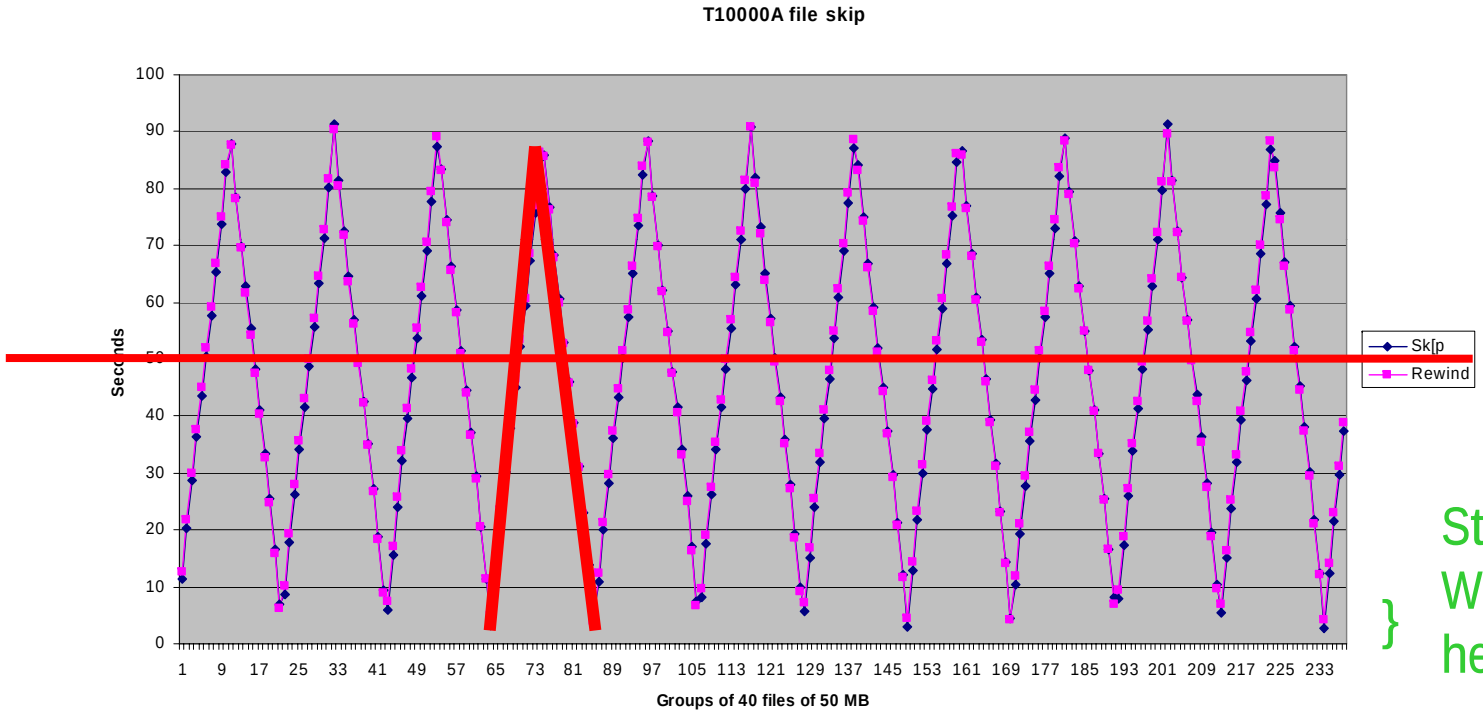


File process rate (ANSI file write/read)
Similar for T10000A, LTO4..
Doing better: don't do labels!



Using all the tape in a cartridge: ‘labelled file’ position/rewind times, T10000A

Doing better: don’t do positioning



Random
Start and end
~ 45 s

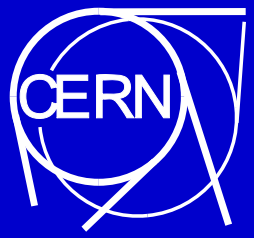
Start and end
WRITE, n x 46 GB
here! ~ 0 s

‘user’ files on tape, each ~50 MB (200 blocks of 256K), simulated labels

Here only ~22.5 of 24 track sets are visible, so only 500 of potential 534 GB ‘allowed’

Command mt -f /dev/nst0 fsf n

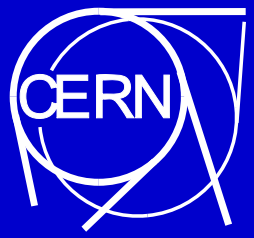
Command mt -f /dev/nst0 rewind



The case of WRITE badly… we CAN do better

Select drive and tape (many possible) with shortest move
Especially helpful in Sun case (approach IBM?)
Write the optimum amount of data (2n x 13 or 23 GB)

IBM, Sun, LTO: ~ 20 s thread no change
Where's EOD? ~ 0 s winding forward
Write: NL file, or 'super VBS' ~ 0 s
Rewind? ~ 0 s rewinding to BOT
Unloading, ~ 20 s no change
Moves, thread, seek, rewind, unload:
~ 130 s IBM, ~ 130 - 330 s Sun
Data writing: ~1000 s 'Useful': 88% IBM, 88 – 75% Sun

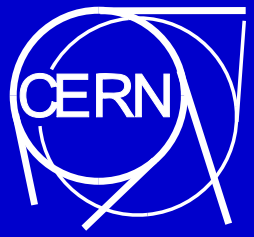


The case of READ badly… part 1

Starts VERY badly: ~1.5 files today for a CASTOR recall
Current file size average in CASTOR is ~ 100 MB!
LHC, maybe 1 GB, maybe ~ 100 files (maybe even a FULL tape?)

Choose a tape (correlation to 'file style'?)
Choose a drive (correlation to tape location?)

Mechanics start to lumber into life to do this random mechanical shuffle….
IBM: ~50 s dismount, ~50 s mount (dual accessor, but..)
Sun: ~150 s dismount, ~150 s mount (vertical, horizontal..)



The case of READ badly… part 2

Drive starts to lumber into life to do this READ

You'll struggle to get near 100 MB/s

IBM, Sun, LTO: ~ 20 s thread

Where's first data? ~ 45 s winding forward

READ: every 1 GB file (labelled ANSI or not), ~ 10 s

READ: every new random file, ~ 45 s spacing

Rewind? ~ 45 s rewinding to BOT

Unloading, ~20 s

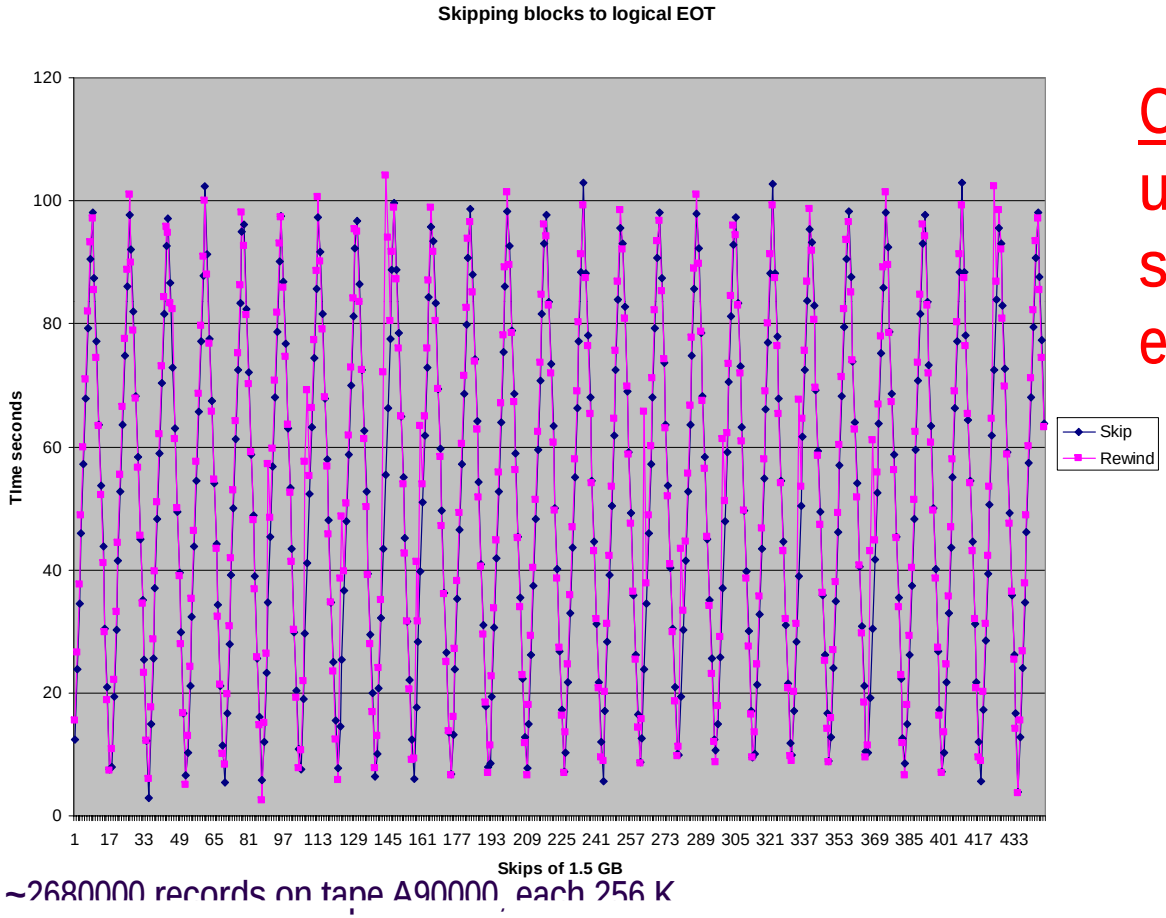
Moves, thread, seek/seeks, NO label write, rewind, unload:

~ 230 - 4680 s IBM, ~ 430 – 4880 s Sun (1 files, 100 files)

Data reading 1 – 100 files: ~ 10 - 1000 s

'Useful': 4 – 21 % IBM, 2 – 20 % Sun

Can do better: order of reading files matters: record position/rewind times, IBM 3592

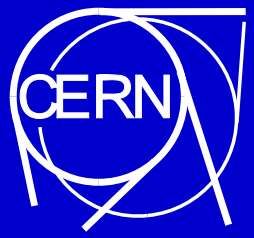


Can waste up to 90s spacing for each file

~2680000 records on tape A90000, each 256 K

Here ~51 of 56 track sets are visible, in write to logical EOT mode, ~706 GB

But setting byte 5 of mode page 37 to '1', so write is to physical EOT, ~764 GB



The case of READ badly… we CAN do better

Select drive and tape (many possible) with shortest move

Especially helpful in Sun case (approach IBM?)

Do not mount until (say) ~ 20 x 1 GB files to read

SORT by position, read ½ on 'way out ', ½ on 'way back'

IBM, Sun, LTO: ~ 20 s thread no change

Where's first data? ~ 0 s winding forward

READ: 'space to next file' ~ 0 s, total ~ 180 s by ordering

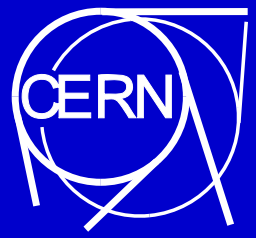
Rewind? ~ 0 s rewinding to BOT

Unloading, ~ 20 s no change

Moves, thread, seek, rewind, unload:

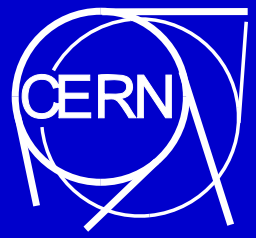
~ 310 s IBM, ~ 310 - 530 s Sun

Data reading: ~ 200 s 'Useful': 39 % IBM, 39 – 27 % Sun



Media or drive upgrade = 'repack' = #@!*&!

- Full tape read or write, at full drive speed, has consistently been '~2 hours'
  - Single upgraded drive might read & re-write ~5 tapes/day
- End 2008, ~35 PB, ~45000 cartridges, ~120 drives
- Conversion time, new media, higher density? Two views ....
  - Using half of 120 upgraded drives, '5 per day', ~150 days
  - At '~5s / file' write (ANSI labels, NVC cannot do it all), ~100 M files is 108 s, ~120 days
- In practice, we never reached these 'expected' rates
  - 9940B took 14 months, first with 16 then rising to 32 drives out of 44 in total
  - We need many drives to collect data for ~120 days / year
  - We need many drives to read back for ~120 days / year
- Repack 2 is ....
- Just exchanging ~45 K old cartridges for new is a long task for IBM 3584 / Sun SL8500



‘repack’: media or drive upgrade vs. physics

Year	Slots	Media	Drives	PB	LHC	Physics read	Physics write	repack read	repack write	
2007	30 K	JB/T1	120	20	~ 0 PB	~ 5	~ 5	~ 9	~ 9	(repack 1) Demonstrated
2008	47 K	JB/T1	120	2	~ 30	~ 9	~ 9	0	0	
2009	57 K	LTO4	120	46	45	45	15	45	45	
2010	57 K	LTO5	120	92	60	60	15	60	60	
2011	57 K	LTO5	240	92	75	75	15	0	0	
2012	57 K	LTO6	240	184	90	90	15	90	90	
2013	57 K	LTO6	240	184	105	105	15	0	0	
2014	57 K	LTO7	240	398	120	120	15	120	120	
2015	57 K	LTO7	240	398	135	135	15	0	0	
Total PBs moved						644	119	324	324	

LTO7 in 2014 is speculative, of course.

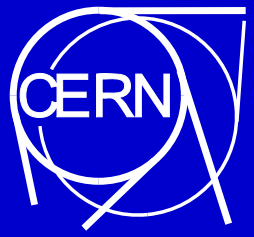
However, we seem to have enough robotics.

‘repack’ is roughly as important as ‘physics’…

Best to do drive upgrade/media change fast, to reduce maintenance / interference

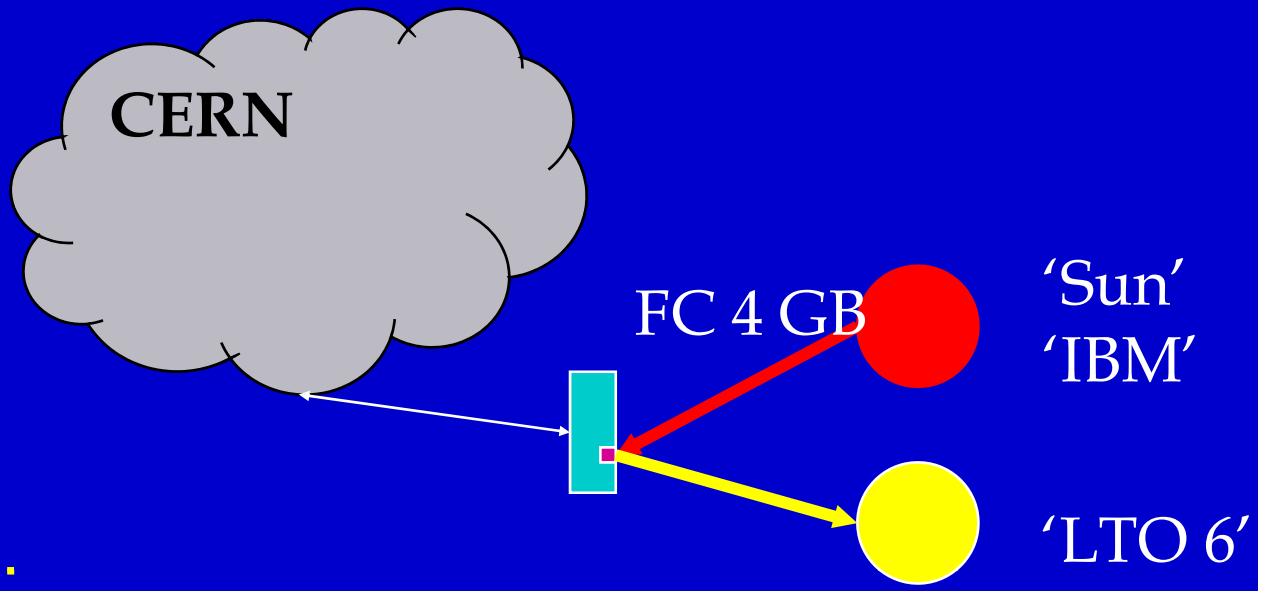
So why use ‘physics’ software?

This task does not need the power or flexibility of CASTOR



An example at random, Monday 5th May…
Just what can these jobs be doing?
An entire tape can be read or written in ~7200 s

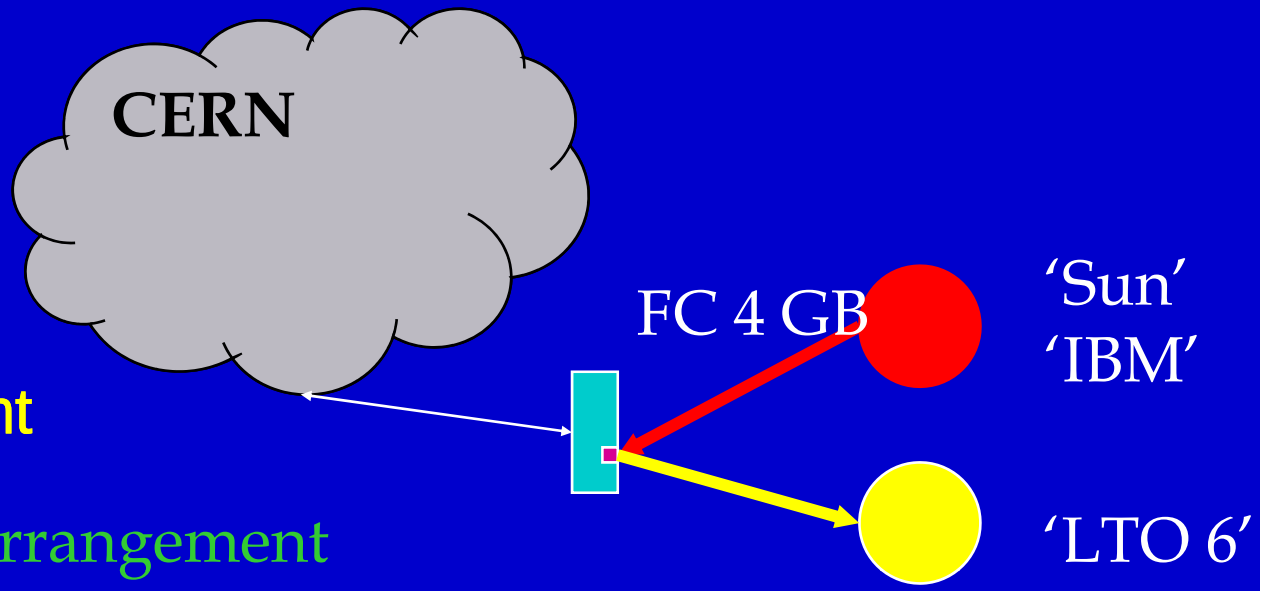
Which inquiry?....
Asked about RUN
Showqueues for RUN
DA T10KR1 T10K101A@tpsrv606 RUNNING 257781 (No_dedication) T12910 T12910 R 18254 (stage,st)@c2cmssrv102.cern.ch
DA 3592B1 35921003@tpsrv204 RUNNING 242041 (No_dedication) I06527 I06527 R 25358 (stage,st)@c2cmssrv102.cern.ch
DA T10K60 T1060711@tpsrv909 RUNNING 192729 (No_dedication) T12763 T12763 R 6033 (stage,st)@c2cmssrv102.cern.ch
DA T10KR1 T10K131C@tpsrv638 RUNNING 187549 (No_dedication) T13352 T13352 R 12411 (stage,st)@c2cmssrv102.cern.ch
DA T10KR1 T10K1314@tpsrv636 RUNNING 68909 (No_dedication) T12866 T12866 R 27475 (stage,st)@c2cmssrv102.cern.ch
DA T10KR1 T10K141F@tpsrv621 RUNNING 68845 (No_dedication) T12852 T12852 R 22873 (stage,st)@c2cmssrv102.cern.ch
DA T10K60 T1060218@tpsrv907 RUNNING 66429 (No_dedication) T13201 T13201 R 30938 (stage,st)@c2cmssrv102.cern.ch
DA T10KR1 T10K111C@tpsrv611 RUNNING 65647 (No_dedication) T12830 T12830 R 17751 (stage,st)@c2cmssrv102.cern.ch
DA T10K60 T1060018@tpsrv913 RUNNING 63711 (No_dedication) T13231 T13231 R 16851 (stage,st)@c2cmssrv102.cern.ch
…. 60 similar lines…..
DA 3592B1 35921024@tpsrv229 RUNNING 10956 (No_dedication) I07484 I07484 R 1880 (stage,st)@c2cmssrv102.cern.ch
DA 3592B1 35921007@tpsrv208 RUNNING 9180 (No_dedication) I08976 I08976 R 3358 (stage,st)@c2cmssrv102.cern.ch
DA 3592B2 35922014@tpsrv150 RUNNING 7312 (No_dedication) I03196 I03196 R 7359 (stage,st)@c2cmssrv102.cern.ch
DA T10K60 T1060415@tpsrv930 RUNNING 6211 (No_dedication) T08901 T08901 R 26433 (stage,st)@c2publicsrv102.cern.ch
DA 3592B2 35922008@tpsrv138 RUNNING 6152 (No_dedication) I10260 I10260 W 21128 (stage,st)@c2atlassrv102.cern.ch
DA T10K60 T1060219@tpsrv927 RUNNING 5933 (No_dedication) T15605 T15605 W 30144 (stage,st)@c2atlassrv102.cern.ch
DA T10K60 T106021D@tpsrv910 RUNNING 5738 (No_dedication) T15599 T15599 W 11182 (stage,st)@c2atlassrv102.cern.ch
DA 3592B1 35921009@tpsrv210 RUNNING 5584 (No_dedication) I10732 I10732 W 5171 (stage,st)@c2cmssrv102.cern.ch
DA 3592B1 35921020@tpsrv225 RUNNING 5322 (No_dedication) I02862 I02862 R 2812 (stag



'repack' is important

25 MB/s 250 MB files,
80 MB/s 1GB files with:

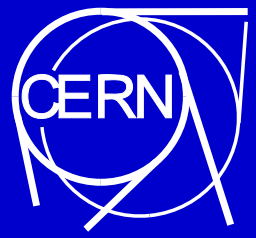
```
#!/bin/csh
@ OK = 0
while ( $OK == 0 )
dd if=/dev/nst0 ibs=80 of=/dev/nst1 obs=80
dd if=/dev/nst0 ibs=262144 of=/dev/nst1 obs=262144
dd if=/dev/nst0 ibs=80 of=/dev/nst1 obs=80
@ OK = $OK + $status
end
```



'repack' is important

Use a specialist arrangement
Trivial (almost) to set up, replicate..
Tape-to-tape, block-to-block, rather simple
If it fails, it's physics data, so user can recover via GRID
Can immediately verify by a full read-back before commit

- Demonstrably faster
- Easier to follow progress
- Not limited by general disk or network infrastructure
- With NL or 'super VBS', REACH native drive speeds



It's (y)our fault

but

We CAN do better

If we don't, then let's just pay for disk and forget it