LTO performance

- Make Tape Reading Great Again -

HEPiX Fall 2018, Barcelona

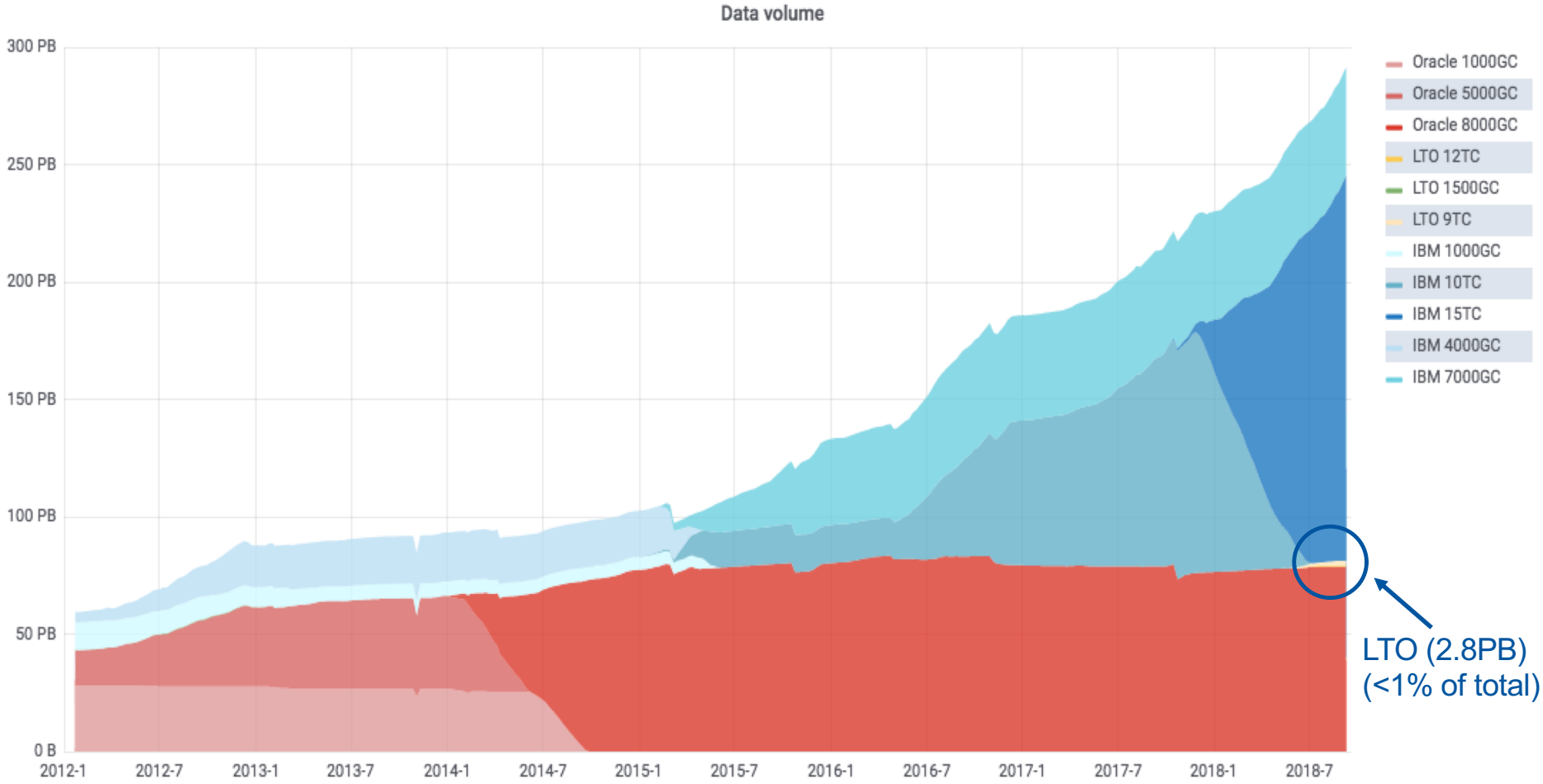
Germán Cancio Meliá – CERN IT/ST

german.cancio@cern.ch

LTO: status at CERN

- CERN continues multi-vendor, multi-platform strategy for tape
  - Keep flexibility in technology, avoid commercial lock-ins
- Shift from {IBM enterprise, Oracle} to {IBM enterprise, LTO} initiated
  - New LTO library purchased: IBM TS4500, 10K slots base, 20 LTO-8 drives, 6PB media (7M and 8)
  - Successfully validated and put in production (July 2018)
  - Open tendering of additional media with interesting price/TB
  - Another LTO library to be procured during LS2 (open tender)
  - Oracle enterprise progressively being decommissioned + media repacked by end LS2

LTO usage at CERN

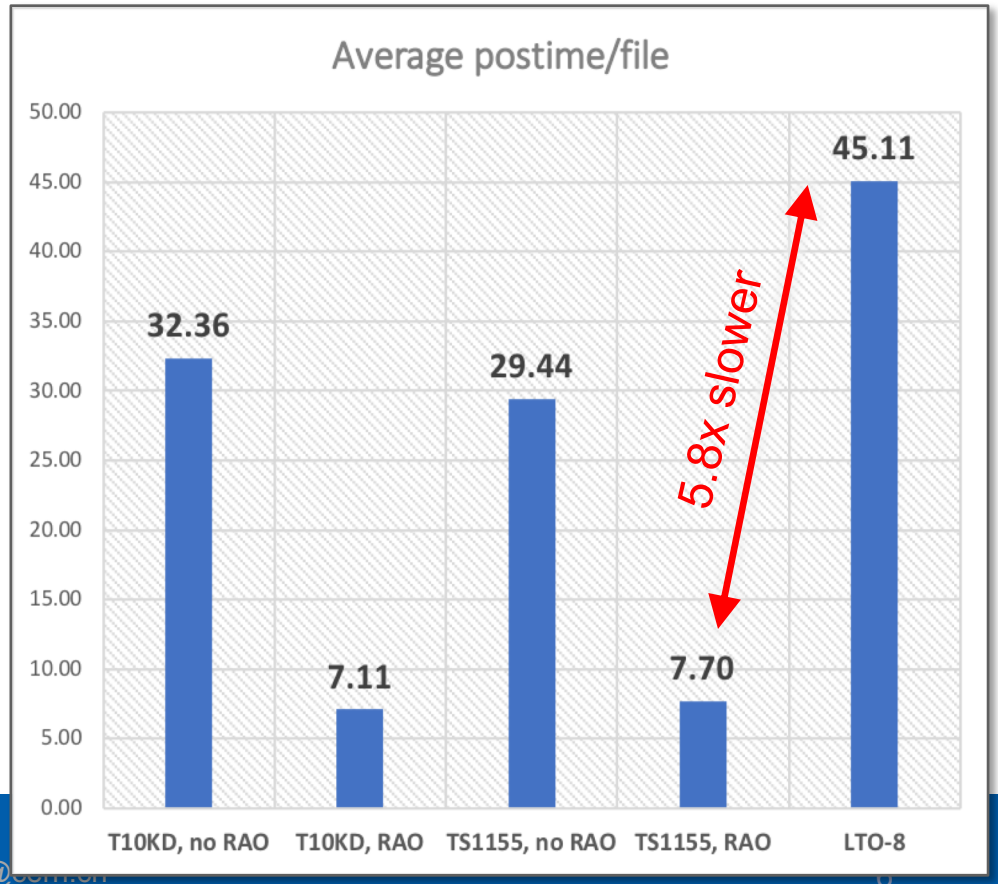


LTO vs Enterprise: key points

Characteristics	IBM TS1155	Oracle T10KD	LTO-8
Capacity	15TB(JD), 7TB(JC)	8TB(T2)	12TB(LTO-8), 9TB(LTO-7M)
Media layout	-- linear serpentine --		
Streaming R/W speed	360MB/s	250MB/s	360MB/s (FH,L8)
Data Buffer	2GB	2GB	1GB
Read access optimisation	RAO, HRTD (64 position areas)	RAO, ??	TD (2 position areas)
Avg file-to-file position time, no RAO, CERN measure	29.4s	32.3s	45.1s
Avg file-to-file position time, RAO, CERN measure	7.7s	7.11s	-
UBER (specs)	10^{-20}	10^{-19}	10^{-19}
Media lifetime (specs)	~26M motion meters	~25K loads/unloads	~22M motion meters, ~20K loads/unloads
Media reusability at higher density	Yes	(Yes)	Only uninitialised LTO7 media

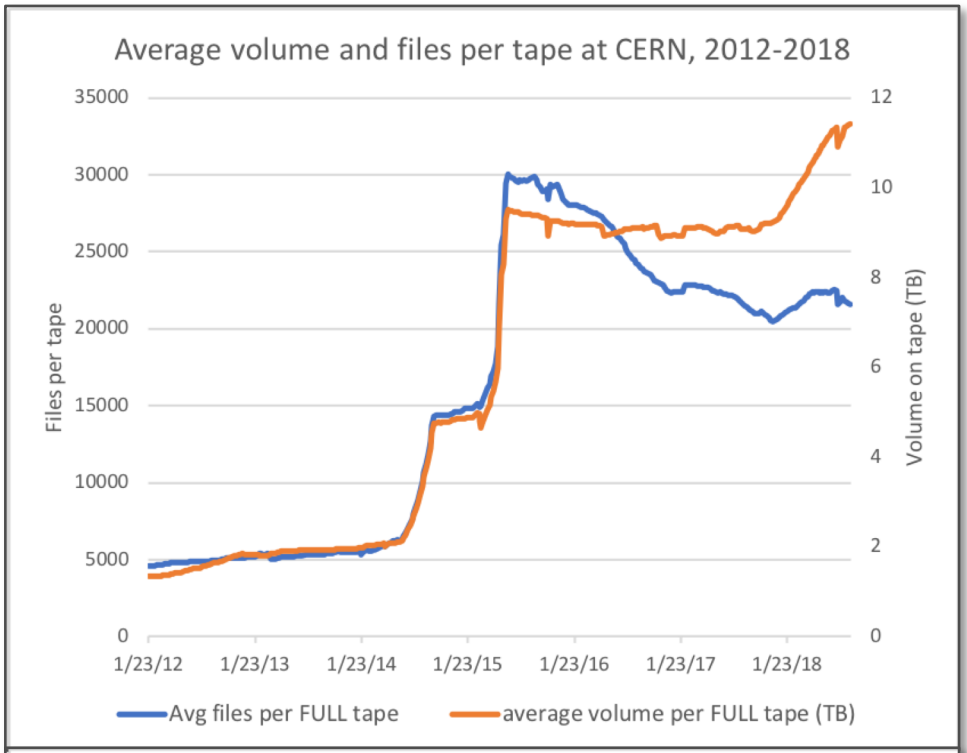
LTO reading efficiency

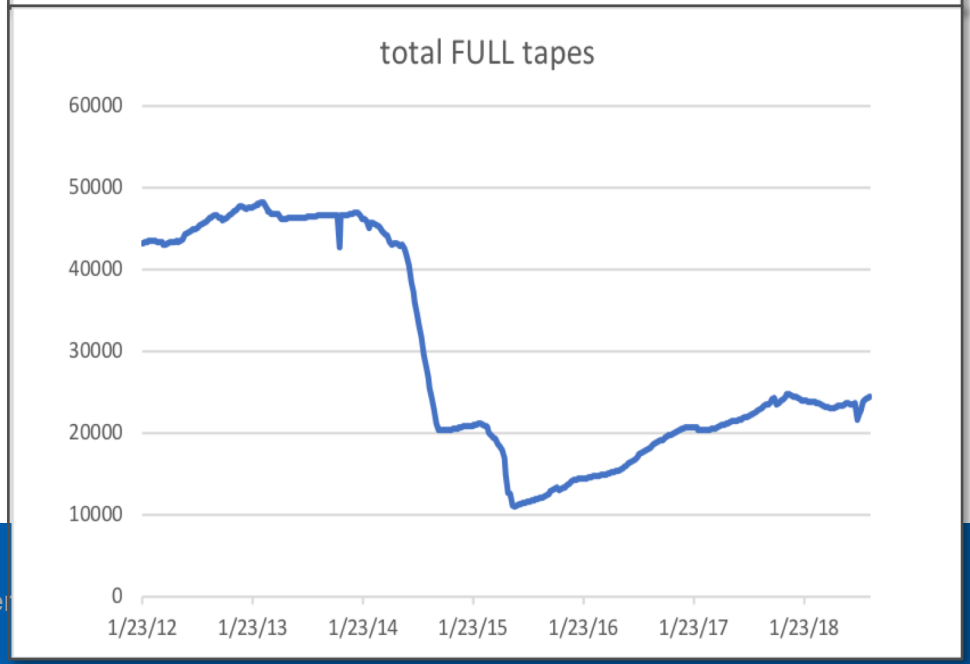
- LTO-8 R/W streaming performance == IBM enterprise
- However, positioning times are **significantly** worse -> much slower effective speed
- Reasons:
  - **Slower seek** speeds (~25%)
  - Lack of high-resolution tape directory -> **hi-speed positioning areas**
    - TS1155 has 64 across tape length
    - LTO-8 only has 2
  - **Lack of drive-assisted request ordering (RAO)** -> files are read in logical ID order, which due to serpentine tape layout is almost as bad as random ordering



Impact

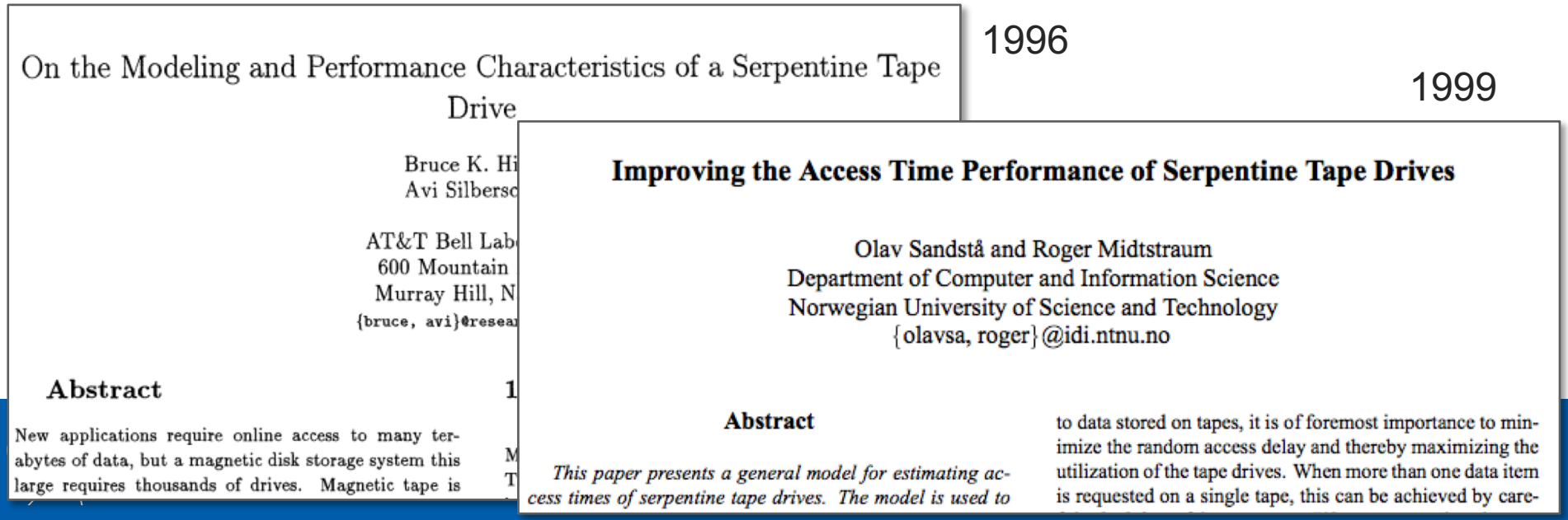
- File sizes are growing slowly, but capacity/tape is growing ~25%/year
  - From 5K files / tape to > 20K files / tape in ~6 years
- Performance bottlenecks moving from robotics/drive availability to head positioning within bigger tapes
  - Majority of read time is for positioning
  - Cannot access data within one tape concurrently!
- Real-life example: ATLAS tape recalls reading ~150 files/tape avg, ~2GB/file. Effective speed
  - IBM enterprise: ~150 MB/s
  - LTO: **~42 MB/s** – less than 1/3rd!
  - Efficiency: $t_{read}/(t_{pos} + t_{read})$ = 0.12
- Faster streaming drives won't help
  - Previous example with doubled streaming performance (720MB/s, +100%): **44MB/s** (+6%)
- Media wear-out increases significantly due to long head traversals





So, what to do?

- One possibility is just to live with it
  - use LTO for “old” and/or “unused” data, such as media migrations of previous run data, or RAW data that should be recalled rarely.
- Another possibility is to purchase a Spectralogic T-Finity
  - Their customised LTO drives offer a TAOS – Time-based Access Ordering System, using generic heuristics to emulate RAO at SCSI level
- Optimising read performance of serpentine tape drives is not a new problem, there was some research done 20 years ago (long before LTO), but seemingly got forgotten… let’s pick up from there…



On the Modeling and Performance Characteristics of a Serpentine Tape Drive

Bruce K. Hi
Avi Silbersc

AT&T Bell Lab
600 Mountain
Murray Hill, N
{bruce, avi}@resea

Abstract

New applications require online access to many terabytes of data, but a magnetic disk storage system this large requires thousands of drives. Magnetic tape is

1

1996

1999

Improving the Access Time Performance of Serpentine Tape Drives

Olav Sandstå and Roger Midtstraum
Department of Computer and Information Science
Norwegian University of Science and Technology
{olavsa, roger}@idi.ntnu.no

Abstract

This paper presents a general model for estimating access times of serpentine tape drives. The model is used to

to data stored on tapes, it is of foremost importance to minimize the random access delay and thereby maximizing the utilization of the tape drives. When more than one data item is requested on a single tape, this can be achieved by care-

How to improve positioning times?

1. You first need to obtain or estimate the **physical location** ("longitudinal position" and "wrap" aka x/y coordinates + direction) of each of the start and end blocks of the file segments on your tapes
   - I.e. $phys\_pos(start\_block(fseg_i)) \rightarrow \{longit = x, wrap = y, dir = z\}$

2. Then, **build a model** and establish a cost estimate function for getting from block i to block j: $cost(block_i, block_j)$, taking into account cost factors (distances, direction changes etc)

3. Finally, define a travelling salesman **algorithm** to **traverse** with minimal cost your file segments to be recalled:
   - $total\_cost = \min(\sum^{i,j} cost(end\_block_i, start\_block_j))$

☛ Read your segments in the resulting order, and you're done ☺

1. Physical location of segments (I)

There is no drive call providing a logical block->physical location mapping: you need to implement this yourself

- First approximation: Use general characteristics of each tape type (LTO-7, LTO-7M, LTO-8) and deduce physical layout of its segments based on each tape's occupancy

  assume tapes are completely filled -> e.g. 168 wraps for LTO-7M

  - Average blocks/wrap for a given tape: $bpt = \frac{\Sigma\, blocks}{\Sigma\, wraps}$
  - $wrap(block_i) = \left\lfloor \frac{block_i}{bpt} \right\rfloor$; $longit(block_i) = block_i \,\%\, bpt$
  - $dir(block_i) = wrap(block_i) \% 2$
- Advantages: trivial to implement
- Drawbacks: not very accurate as the number of filled wraps varies from tape to tape, and there might be reserved wraps. Blocks per wrap also may change depending on compression variance ("accordion tapes"), write rate adaptation, bad media areas, etc. => All errors accumulate over 160+ wraps!

1. Physical location of segments (II)

There is no drive call providing a logical block->physical location mapping: you need to implement this yourself

- Second approximation: RTFM… and use the REQUEST SENSE SCSI command to find out the current physical location of the tape head

Table 125 — REQUEST SENSE Sense Data Format

Byte	Bit							
	7 msb	6	5	4	3	2	1	0 lsb
0	VALID	RESPONSE CODE						
1	Obsolete (00h)							
2	FILEMARK	EOM	ILI	Reserved	SENSE KEY			
22	VOLUME LABEL							
28								
29	PHYSICAL WRAP							
30	(MSB)							
33	RELATIVE LPOS VALUE							(LSB)

Source: IBM Ultrium LTO Tape Drive SCSI Reference Manual

1. Physical location of segments (II)

There is no drive call providing a logical block->physical location mapping: you need to implement this yourself

- Second approximation: RTFM… and use the REQUEST SENSE SCSI command to find out the current physical location of the tape head

```
# sg_requests -H /dev/sg1
 00      70 00 00 00 00 00 00 58   00 00 00 00 00 00 30 00
 10      00 00 00 00 01 01 4c 37   30 30 36 33 4d 02 00 01
 20      1a 77 56 00 00 00 00 56   81 00 60 4d 38 00 07 b0
```

wrap: 02h
LPOS: 00011A77h

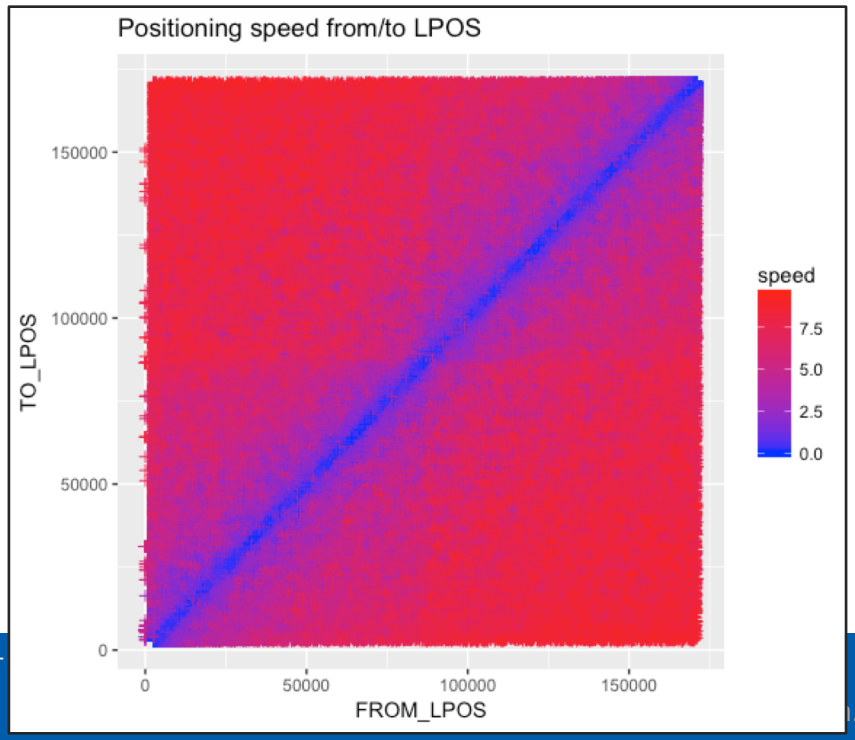
  - physical wrap: from 0 to 167 (LTO-7M) or 203 (LTO-8)
  - LPOS (longitudinal position) units: from 0 to 171144, counted from BOT
- This information can be obtained either during reading or writing - at no noticeable performance loss
  - Record wrap/LPOS of the beginning or end of each newly written segment and keep it along with fseq and blockID in your tape catalogue
  - Requires (of course) modifications to your MSS software!

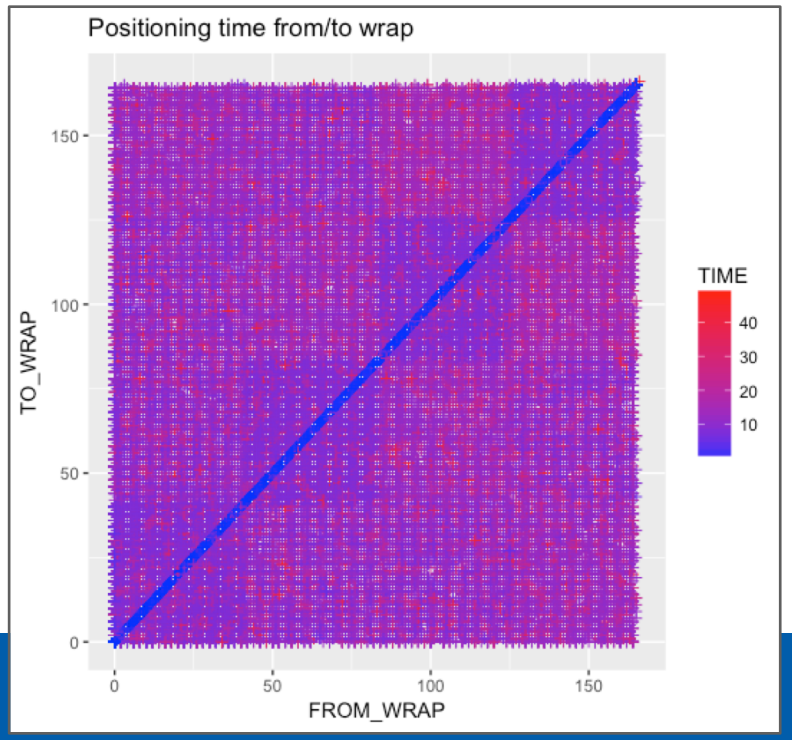
2. Modeling positioning cost (I)

- $cost\ (block_i, block_j)$. What are the cost coefficients?
  - Speed and longitudinal distance between $block_i$ and $block_j$
  - Cost of change of wrap
  - Cost of change of band (wraps are organized in 4 separate bands)
  - Cost of crossing mid-tape (change of hi-speed positioning area)
  - Cost of change of read direction
  - Cost of stepping back (rewind) within the same direction
  - etc..
- No documentation is available on these drive internal parameters, so need to measure/infer yourself
  - Across many positions within a tape
  - Across many tapes and tape types
  - Across many drives

2. Modeling positioning cost (II)

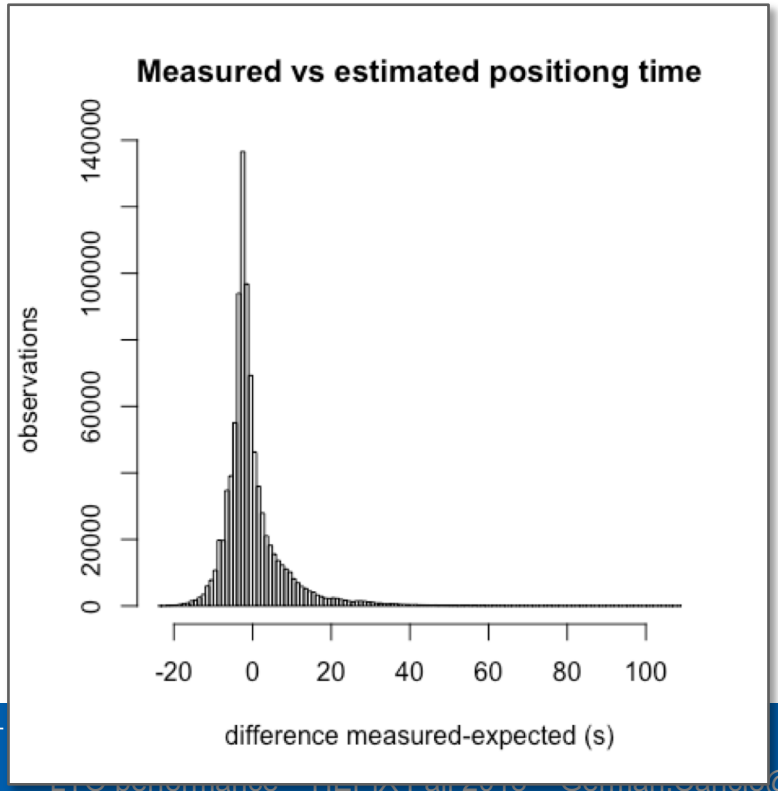
- Generate tape read workloads (on real data) and collect physical position information (from, to, time)
- Using multilinear regression to obtain values for cost coefficients
- Compare expectations vs measurements and repeat over several iterations
- In total, analysed 800K positions across 8400 mounts, 140 LTO-7M tapes and 20 drives

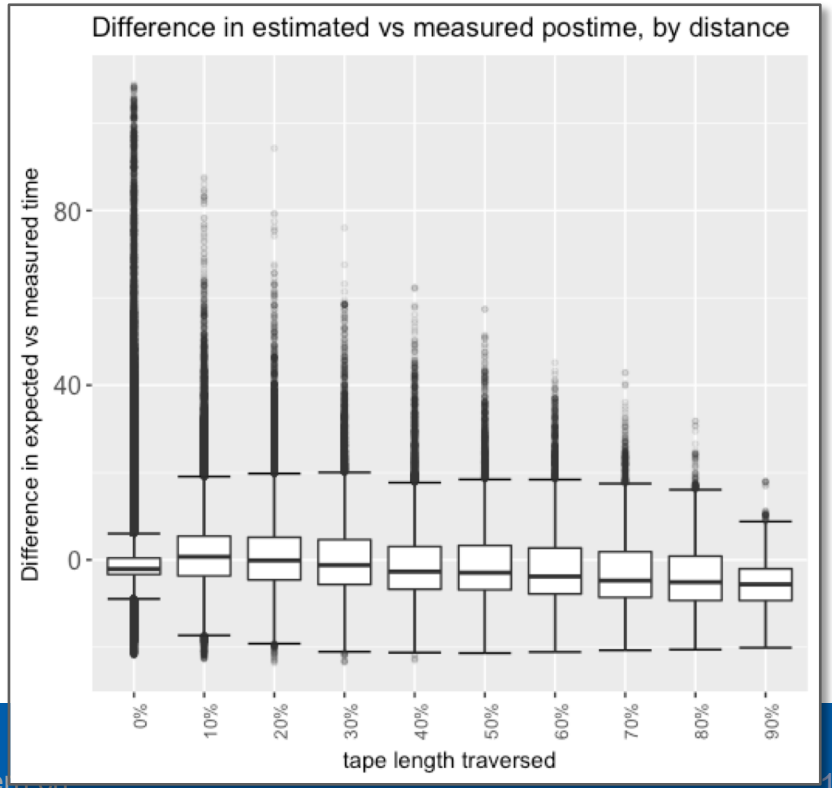




2. Modeling positioning cost (III)

- Good approximations overall… (see MLR coefficients in annex)
- .. but estimates in particular for short distance positionings can still be improved – and these are frequent when recalling $O(100)$ files / tape
  - Better modeling of drive behavior when drive switches from seek->read (cf. annex slides)





3. Traversal algorithm (I)

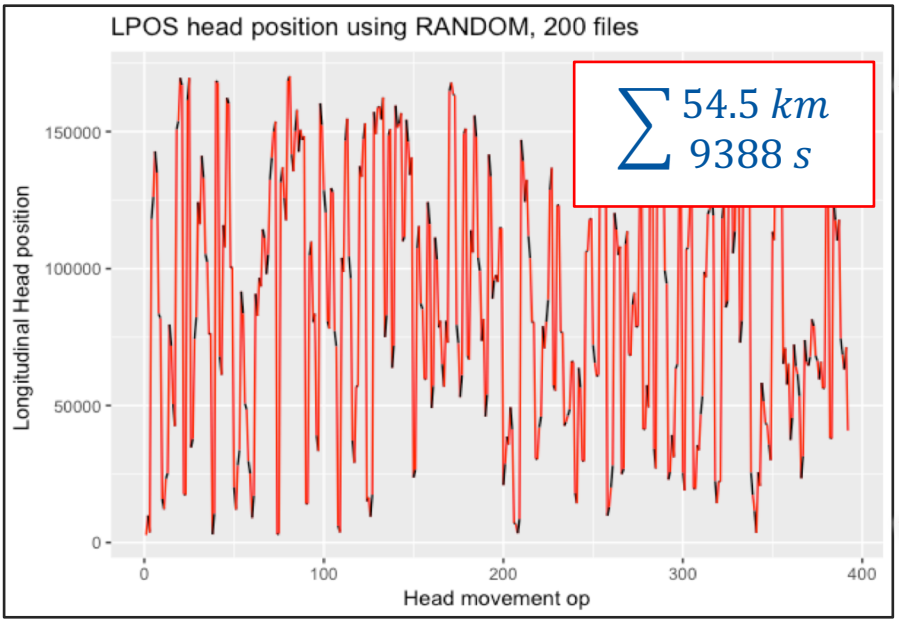
- With the cost function established, you now need to feed your list of n tape files to retrieve into a traversal algorithm to compute the optimal order
  - Build a DAG with segments as vertices and traversal costs as edges
- Optimal algorithm (all path permutations) is $O(n!)$ so not practical for $n \geq 10$

- Evaluated and compared the following algorithms running O(1000) grouped recalls of 10, 50, 100, 200, 500 files:
  - RANDOM: retrieve files in random order (ie. FIFO)
  - LINEAR: order by logical file ID (default in CASTOR, HPSS etc)
  - SCAN: two-pass elevator algorithm – partition segments by direction and order them by physical LPOS, and do a forth and back pass
  - SLTF: (Shortest Locate Time First) – traverse by always picking the nearest (lowest cost) neighbour

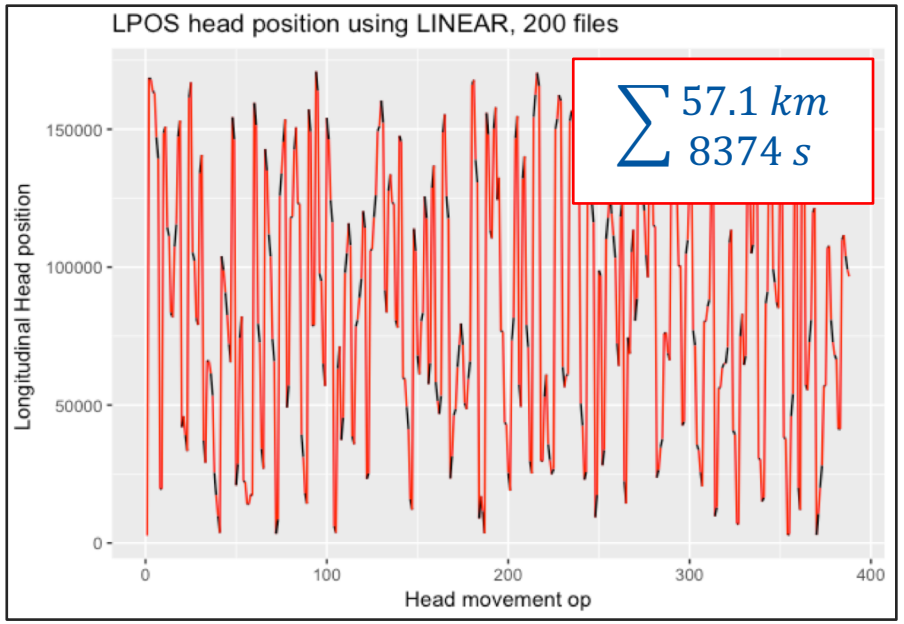
3. Traversal algorithm (II)

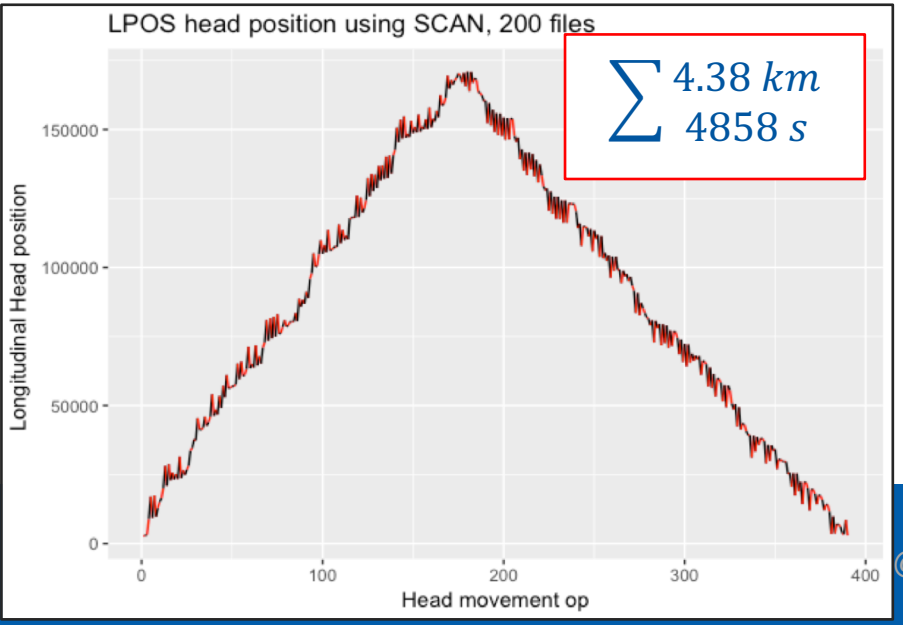
Example: retrieving same 200 files from same tape

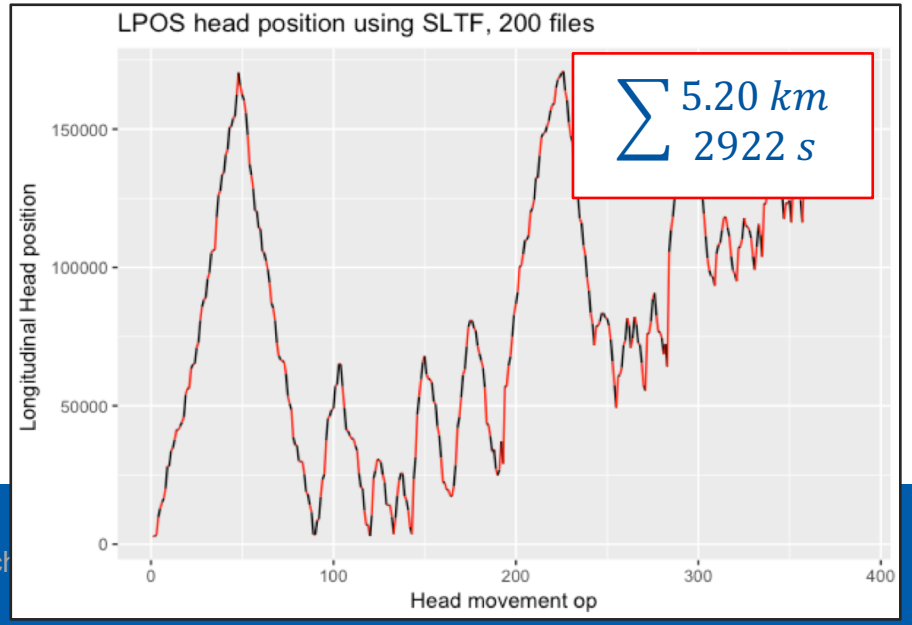
: positioning movement
: reading movement



Media length (~1km)







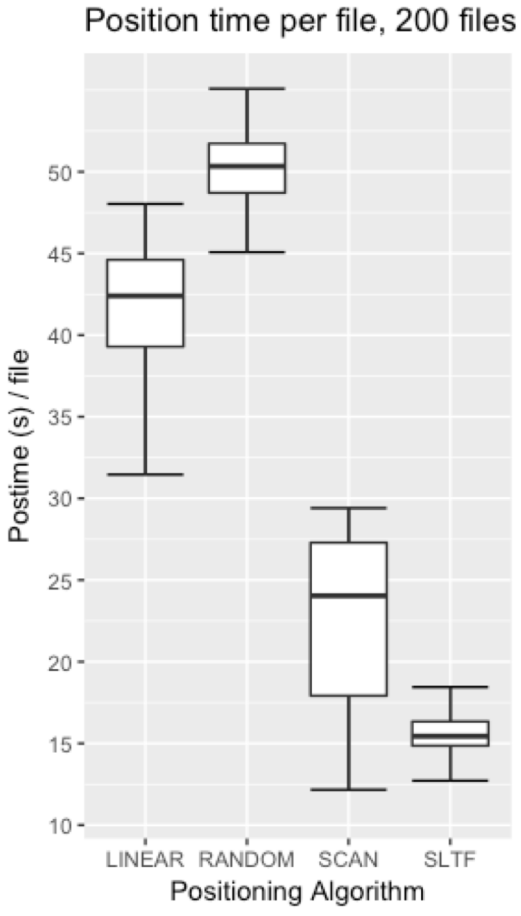
Results so far

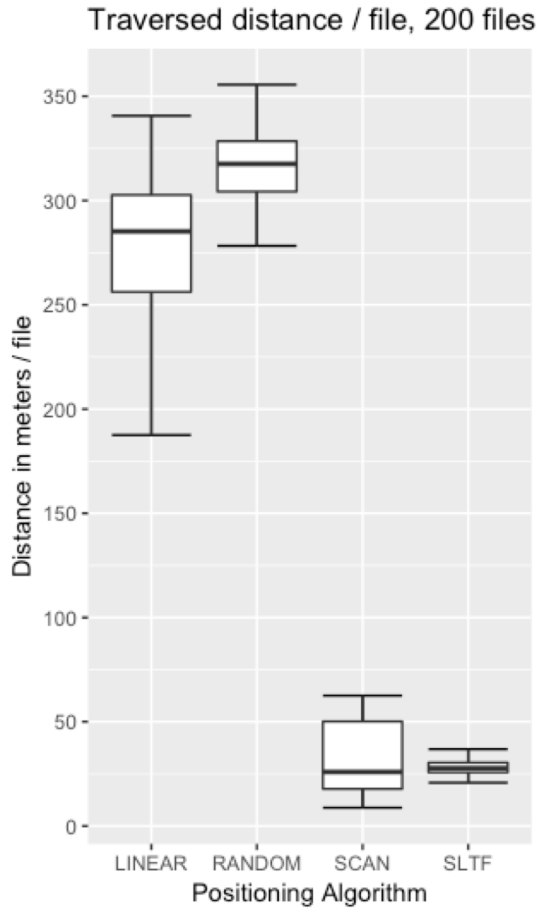
- Compared to LINEAR and RANDOM, SLTF and SCAN reduce media traversal distance (thus wear) by over 11x
- SLTF is the fastest positioning algorithm (~2.5x-3x faster than LINEAR)

Further gains in reach:

- Improved local movement modeling
- More performant algorithms using heuristics and approximations
- A research area for our PhD student starting this month

Support for optimised LTO access will be implemented in the upcoming CERN Tape Archive (CTA) SW during 2019





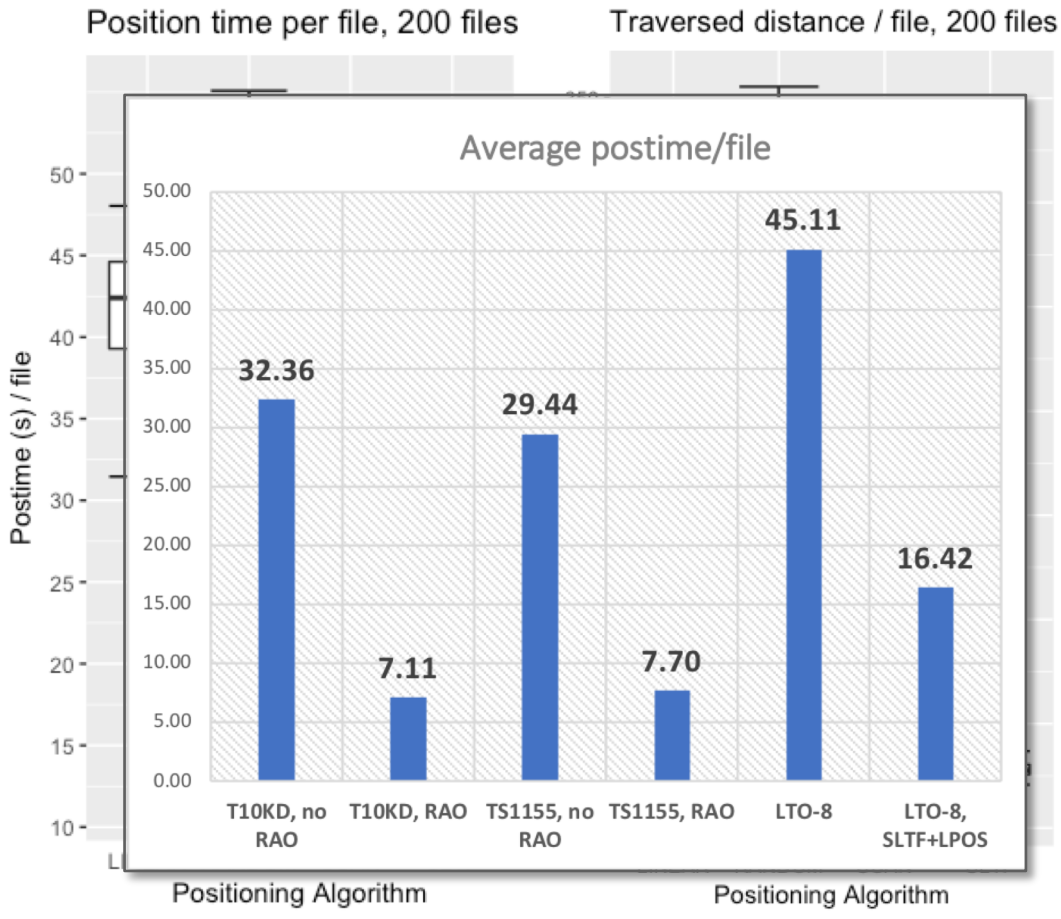
Results so far

- Compared to LINEAR and RANDOM, SLTF and SCAN reduce media traversal distance (thus wear) by over 11x
- SLTF is the fastest positioning algorithm (~2.5x-3x faster than LINEAR)

Further gains in reach:

- Improved local movement modeling
- More performant algorithms using heuristics and approximations
- A research area for our PhD student starting this month

Support for optimised LTO access will be implemented in the upcoming CERN Tape Archive (CTA) SW during 2019



Summary

- LTO-8 is an interesting alternative/complement to IBM enterprise offering a very attractive price/TB
- LTO lacks drive-assisted reordering of file requests, making reading much slower -> significant impact for “tape carousel” activities…
- … but this reordering can be implemented outside the drive!
- Internal (undocumented) drive parameters can be inferred via linear regression techniques
- Positioning times can be improved by an average factor of 2.5x-3x, and media traversal (thus wear) reduced by over an order of magnitude… and there is still room to improve further…
- … making LTO Tape Reading Great (…or better) again

Annex and References

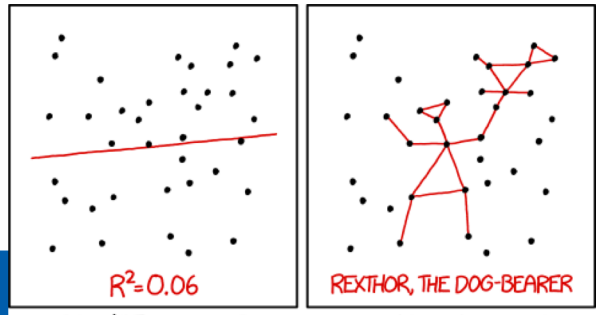
Linear Regression coefficients

```
Call:
lm(formula = TIME ~ WRAPCHANGE + BANDCHANGE + MID_CROSS + DIRCHANGE +
    STEPBACK + distance, data = all)

Coefficients:
(Intercept)   WRAPCHANGE   BANDCHANGE    MID_CROSS    DIRCHANGE     STEPBACK     distance
  4.2389745    6.6977300    3.1991467   -6.0424286    5.2179431   11.3157622    0.0006192

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.239e+00  2.791e-02   151.9   <2e-16 ***
WRAPCHANGE   6.698e+00  3.114e-02   215.1   <2e-16 ***
BANDCHANGE   3.199e+00  1.862e-02   171.8   <2e-16 ***
MID_CROSS   -6.042e+00  4.079e-02  -148.1   <2e-16 ***
DIRCHANGE    5.218e+00  2.396e-02   217.8   <2e-16 ***
STEPBACK     1.132e+01  2.325e-02   486.6   <2e-16 ***
distance     6.192e-04  4.894e-07  1265.2   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.193 on 886028 degrees of freedom
Multiple R-squared:  0.831,     Adjusted R-squared:  0.831
F-statistic: 7.262e+05 on 6 and 886028 DF,  p-value: < 2.2e-16
```

base repositioning overhead (in s)

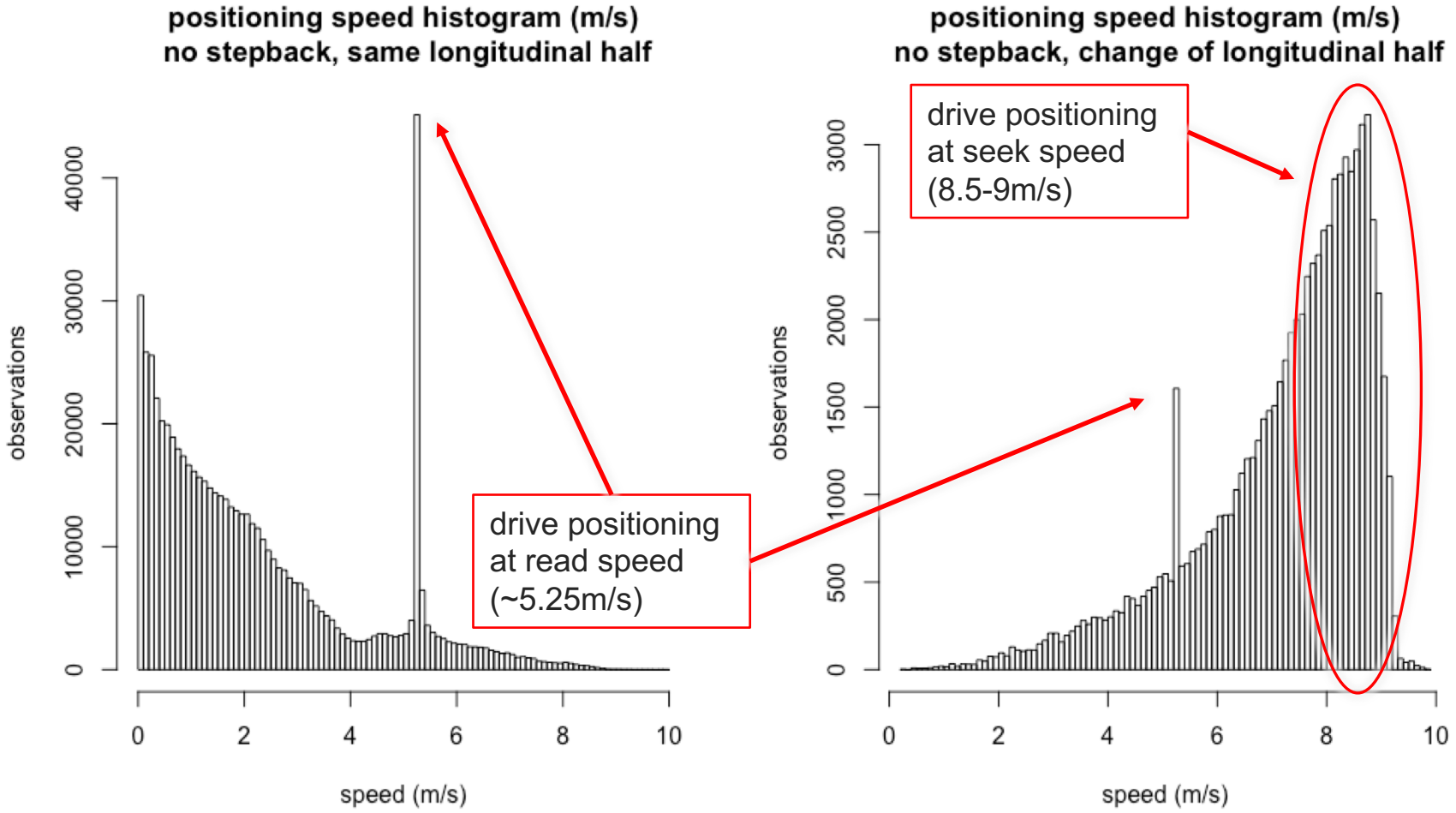
wrap change (in s)
band change (in s) [on top of wrap change]
Crossing mid-tape (in s) [negative: faster]
direction change (in s)
stepback (in s)

distance factor (seconds/LPOS unit)

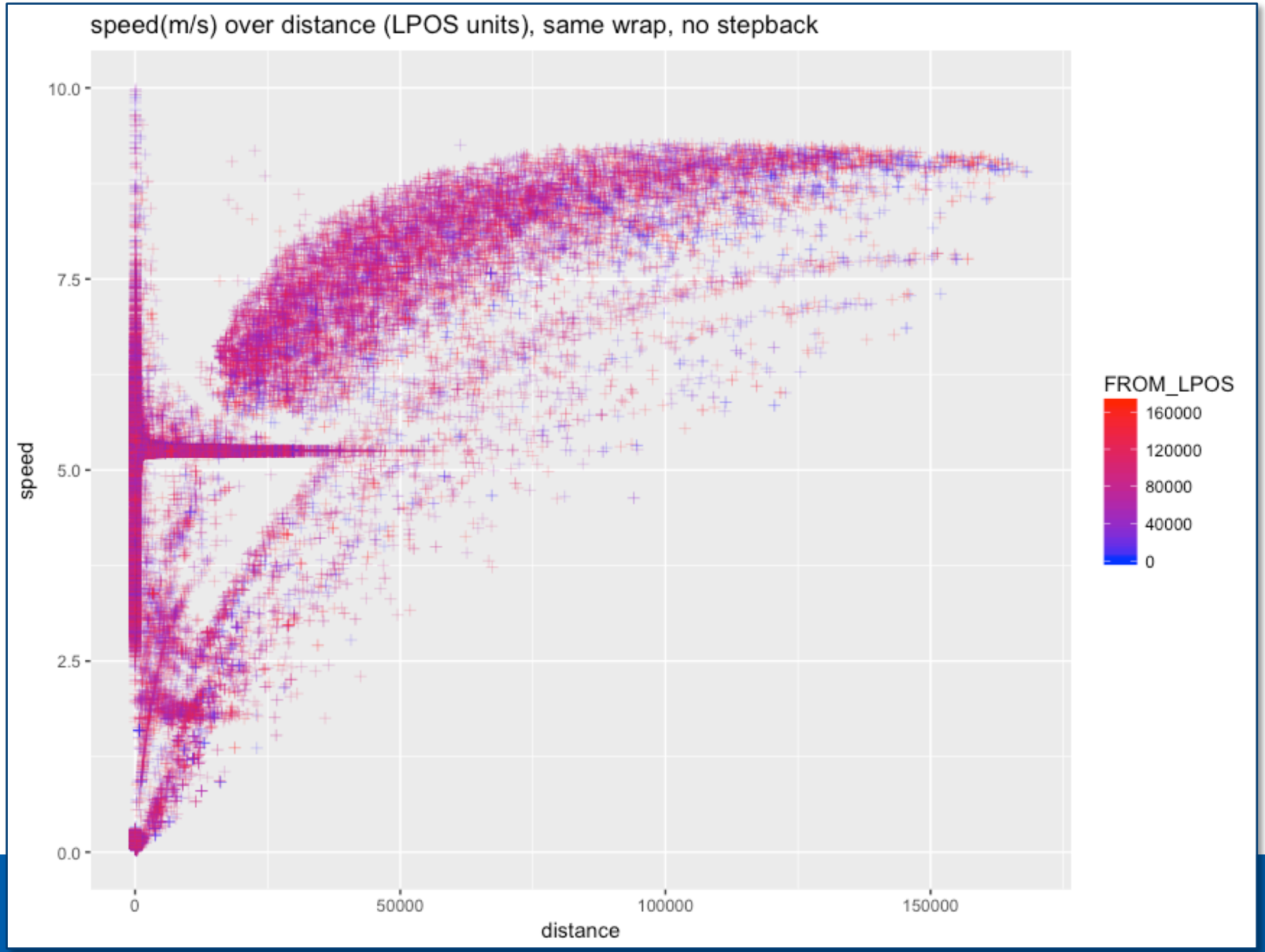


I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

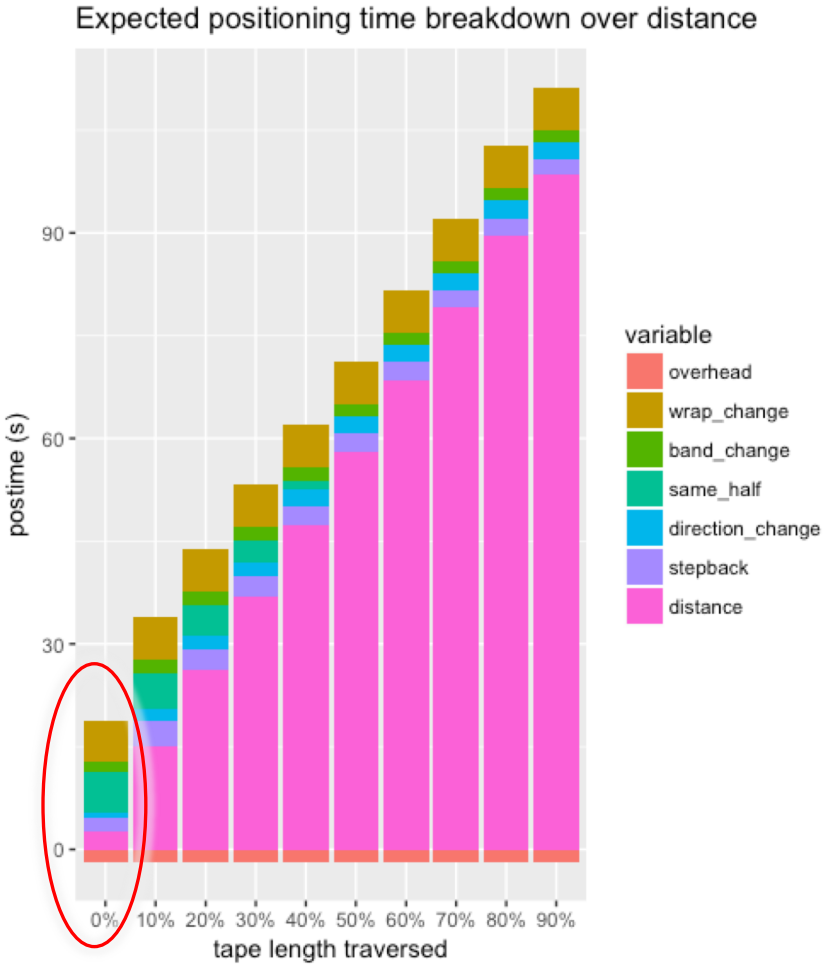
Positioning speed and crossing mid-tape (change of positioning area)

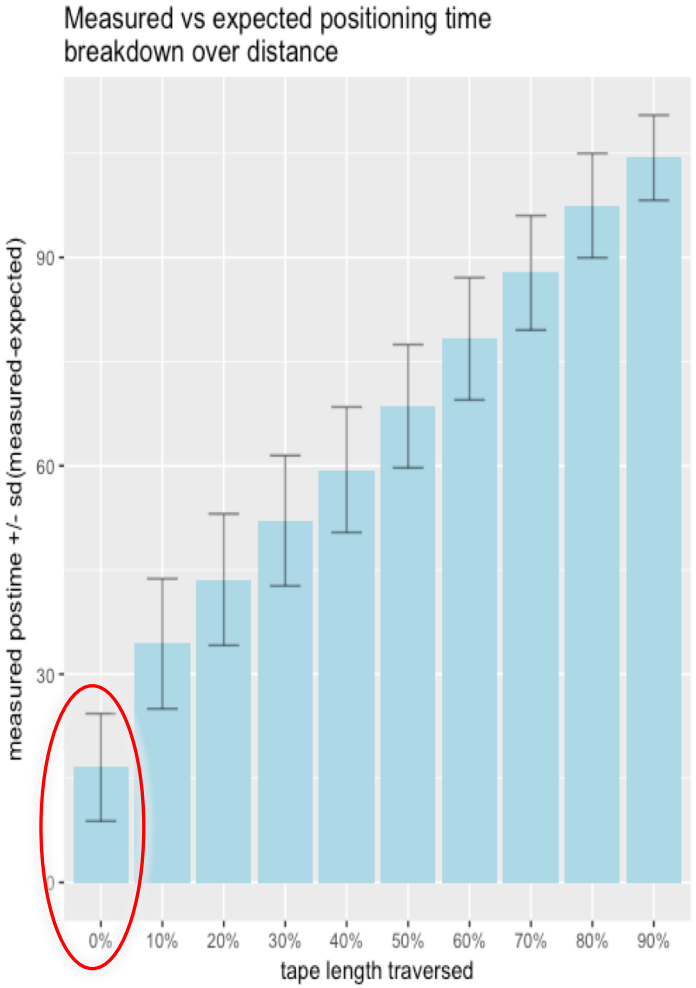


Positioning speed within same wrap

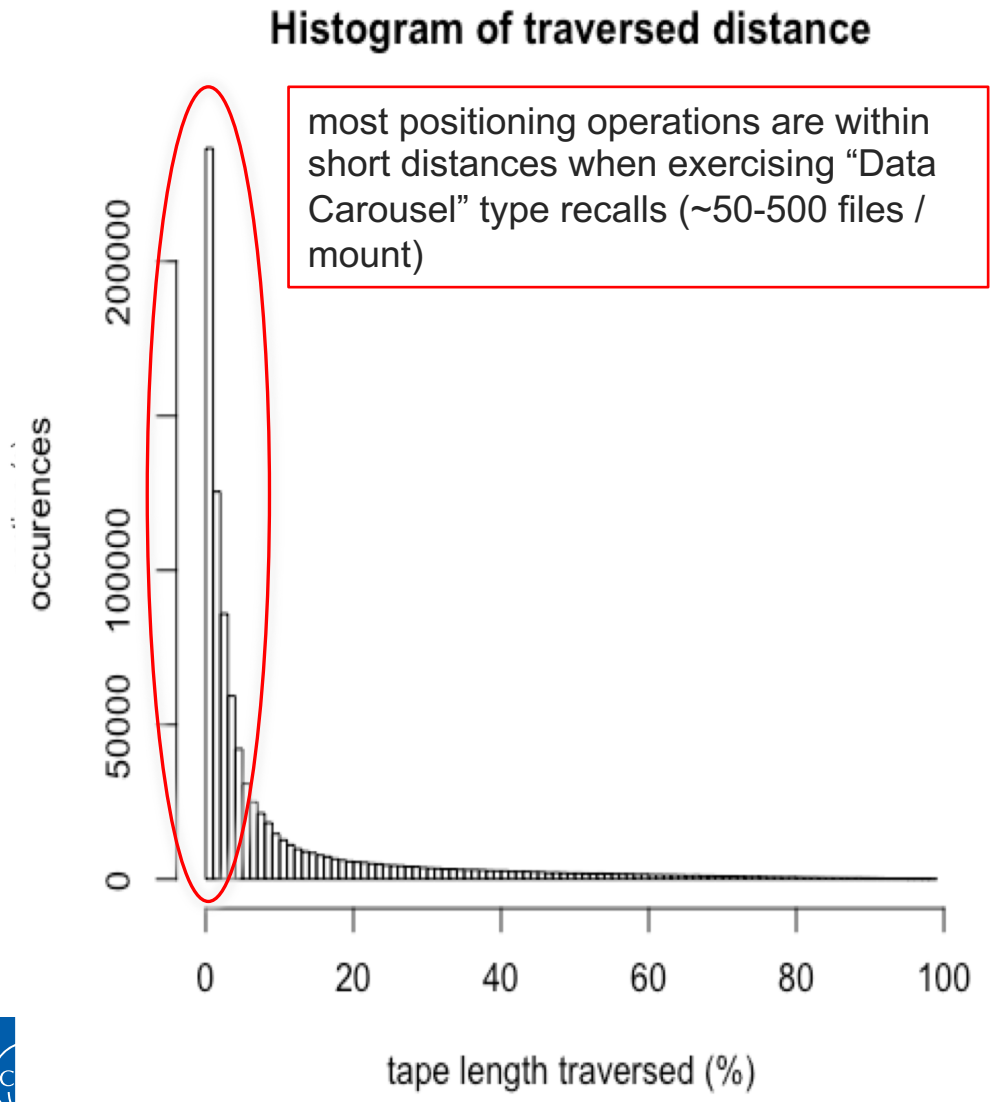


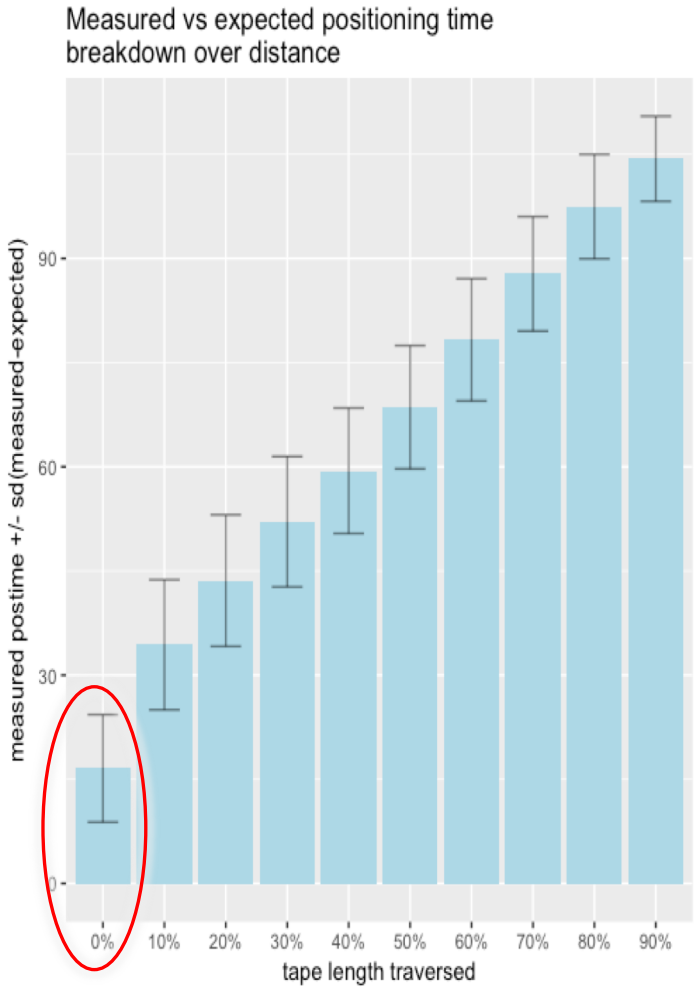
Expected vs observed positioning times





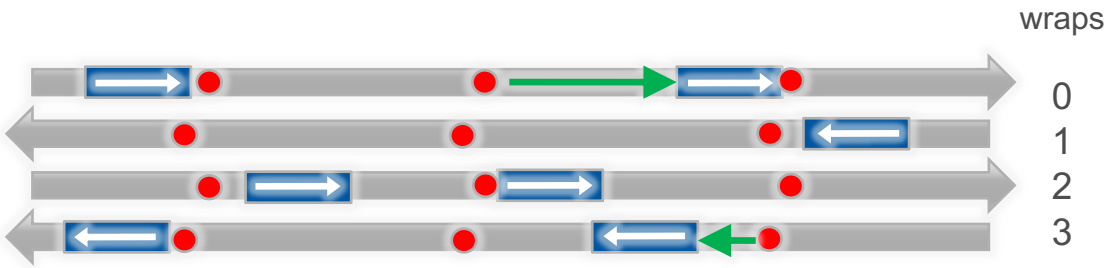
Expected vs observed positioning times





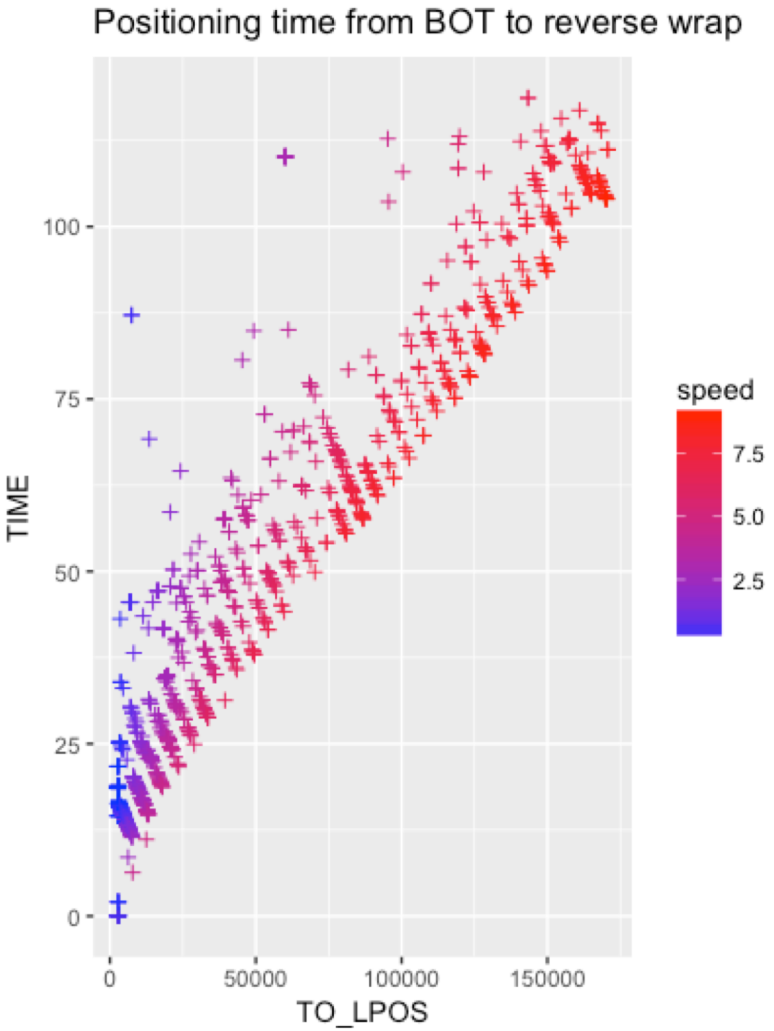
Impact of “landing” behavior on performance

To be investigated: landing behavior
local “key points” or “landing points”: index areas on media where drive can switch from seek->read
cf. papers by Hillyer+Sandsta

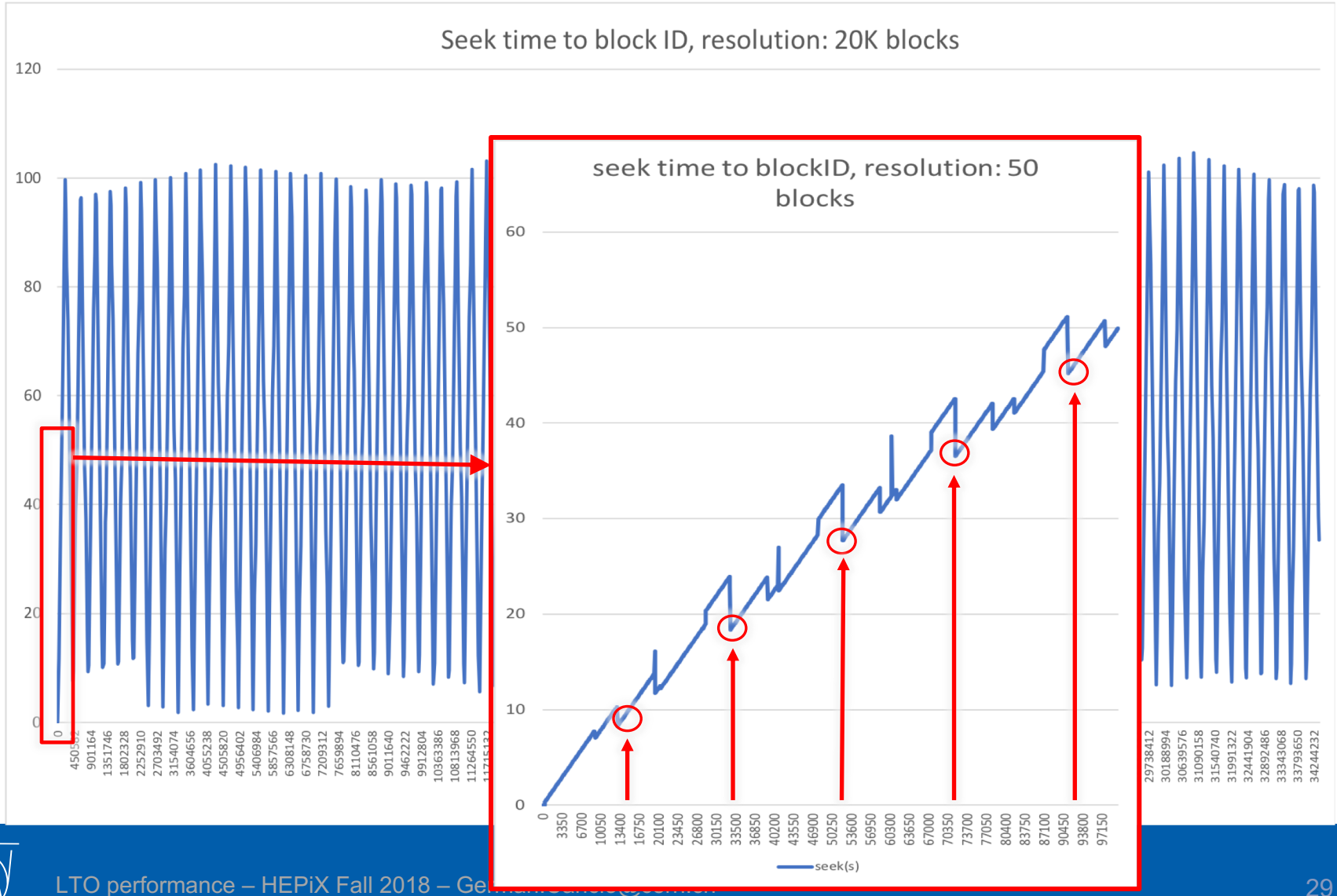


● : tape head local landing points (from seek to read)

→ : distance to beginning of file to be traversed at reading speed



Local landing points and positioning times



References

- IBM TotalStorage LTO Ultrium Tape Drive SCSI reference
- B. K. Hillyer and A. Silberschatz. *On the modeling and performance characteristics of a serpentine tape drive.* Proceedings of the 1996 ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, pages 170- 179, Philadelphia, Pennsylvania, May 1996.
- B. K. Hillyer and A. Silberschatz. *Random scheduling in online tertiary storage*. In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pages 195-204, Montreal, Canada, June 1996.
- O. Sandsta and R. Midstraum. *Low-Cost Access Time Model for Serpentine Tape Drives.* 16th IEEE Symposium on Mass Storage Systems and the 7th NASA Goddard Conference on Mass Storage Systems and Technologies, pages 116–127, San Diego, California, USA, March 1999.
- O. Sandsta and R. Midtstraum. *Improving the access time performance of serpentine tape drives*. 15th International Conference on Data Engineering, Sydney, Australia, March 1999.