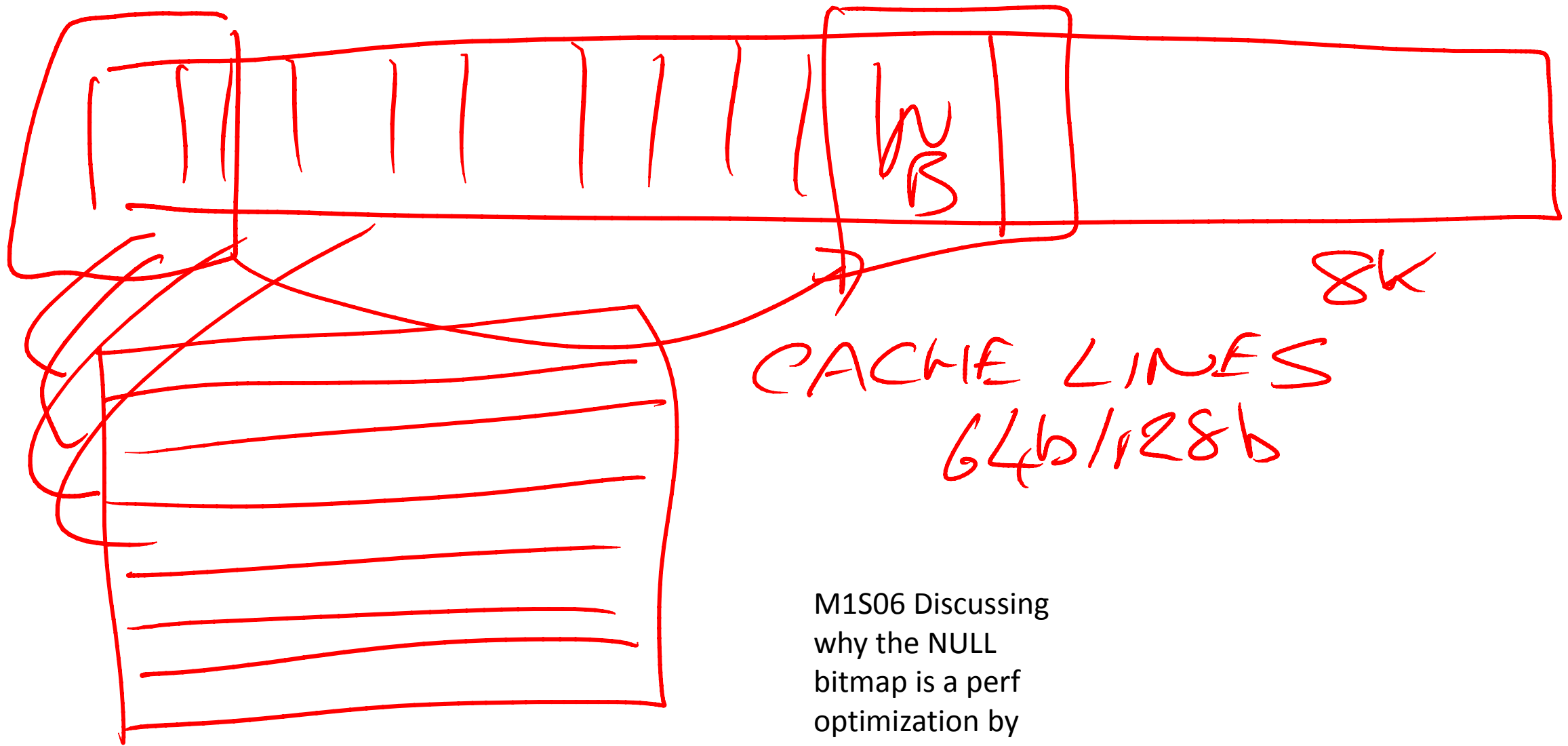


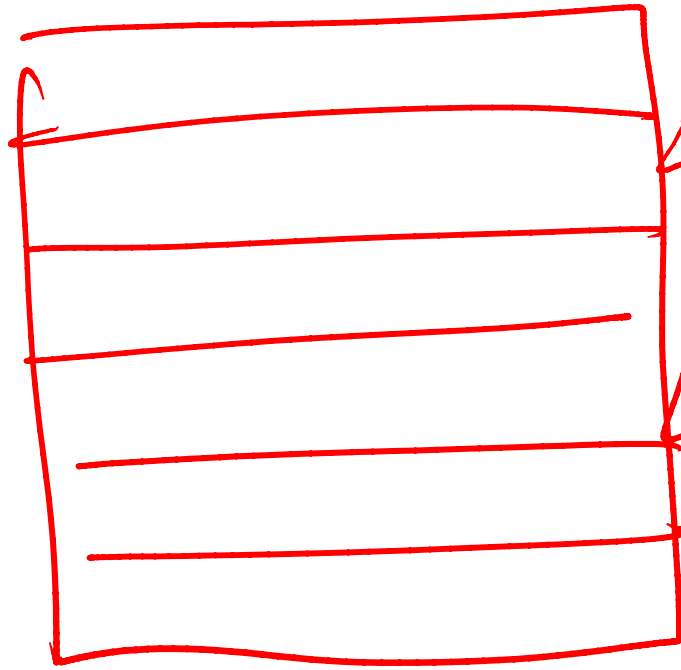


CACHE LINES  
64b/128b

M1S06 Discussing  
why the NULL  
bitmap is a perf  
optimization by  
avoiding having to  
read record parts  
into the CPU's L1  
cache

$c1, c2, \dots, c10$

NCI  $c2$



PTR

sel  $c1, c2, c3$

where  $c2 = X$   
(File, Page, Slot)

HEAP

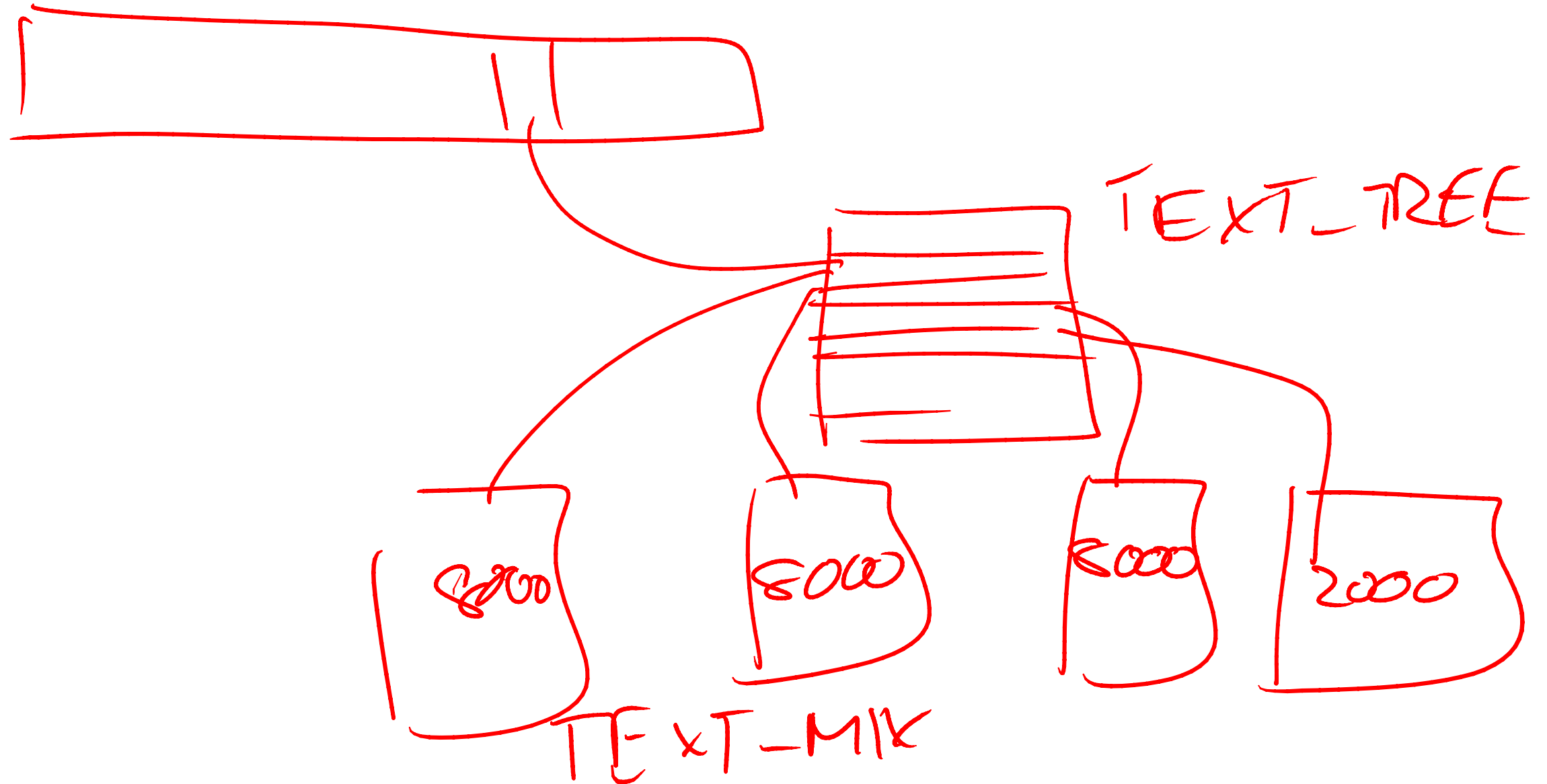
PTR = PHYSICAL RID

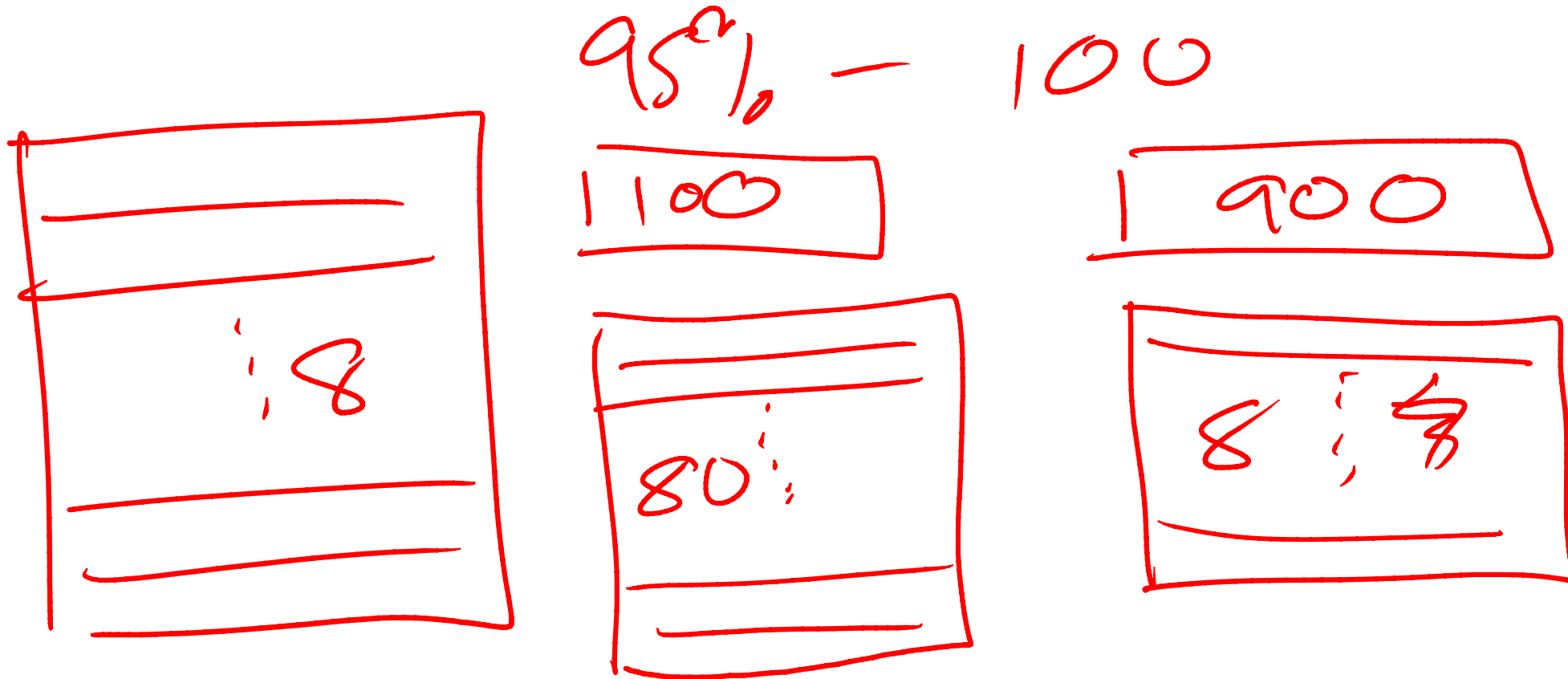
CL

PTR = LOGICAL RID (cluster keys)

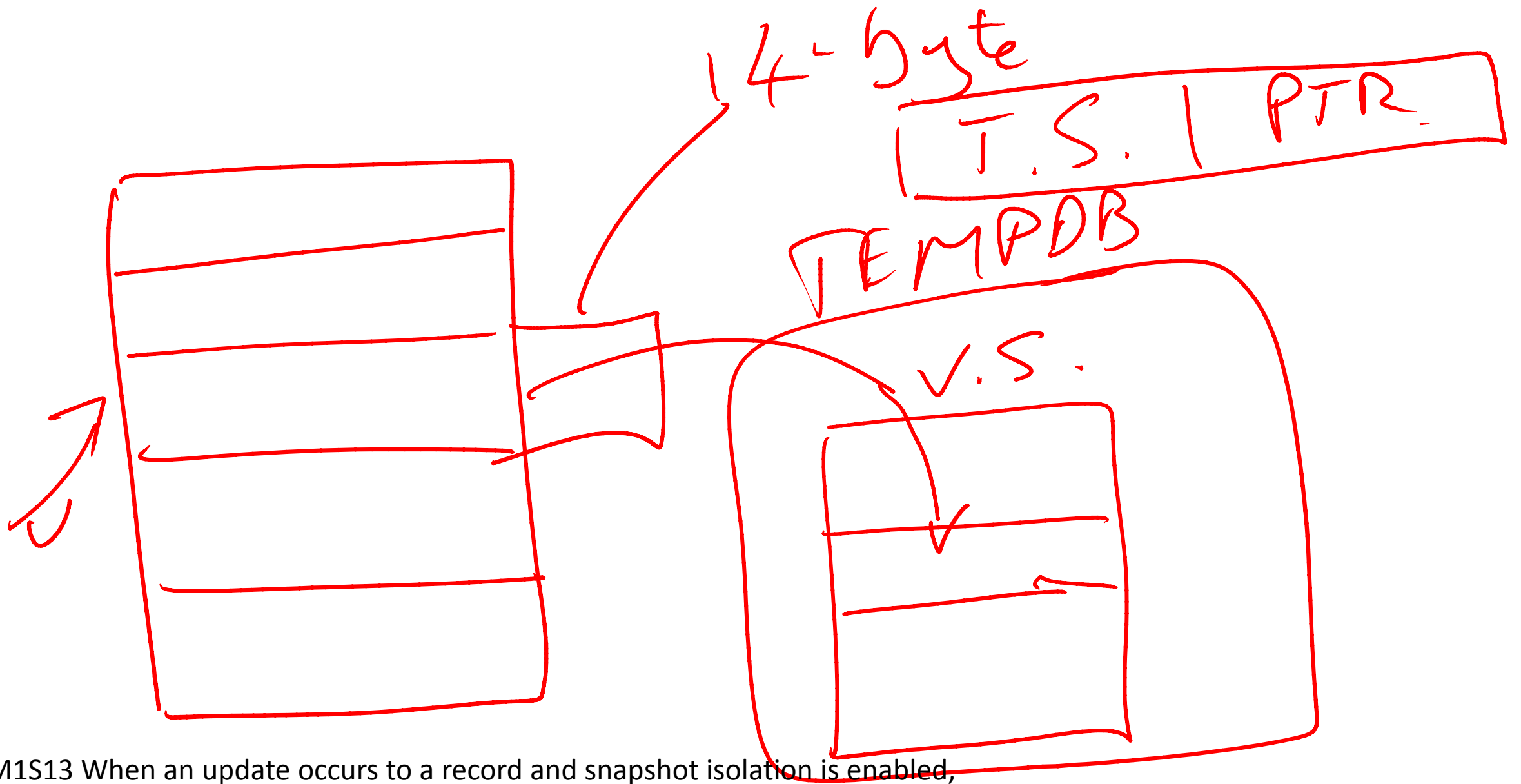
M1S09 Explaining NC index back-links to the data records as part of describing why forwarding/forwarded records exist. Without them, moving a heap row would entail changing all NC index records with a back-link to that row.

M1S11 Google Marc Rasmussen for his blog. He's blogged text tree structures.

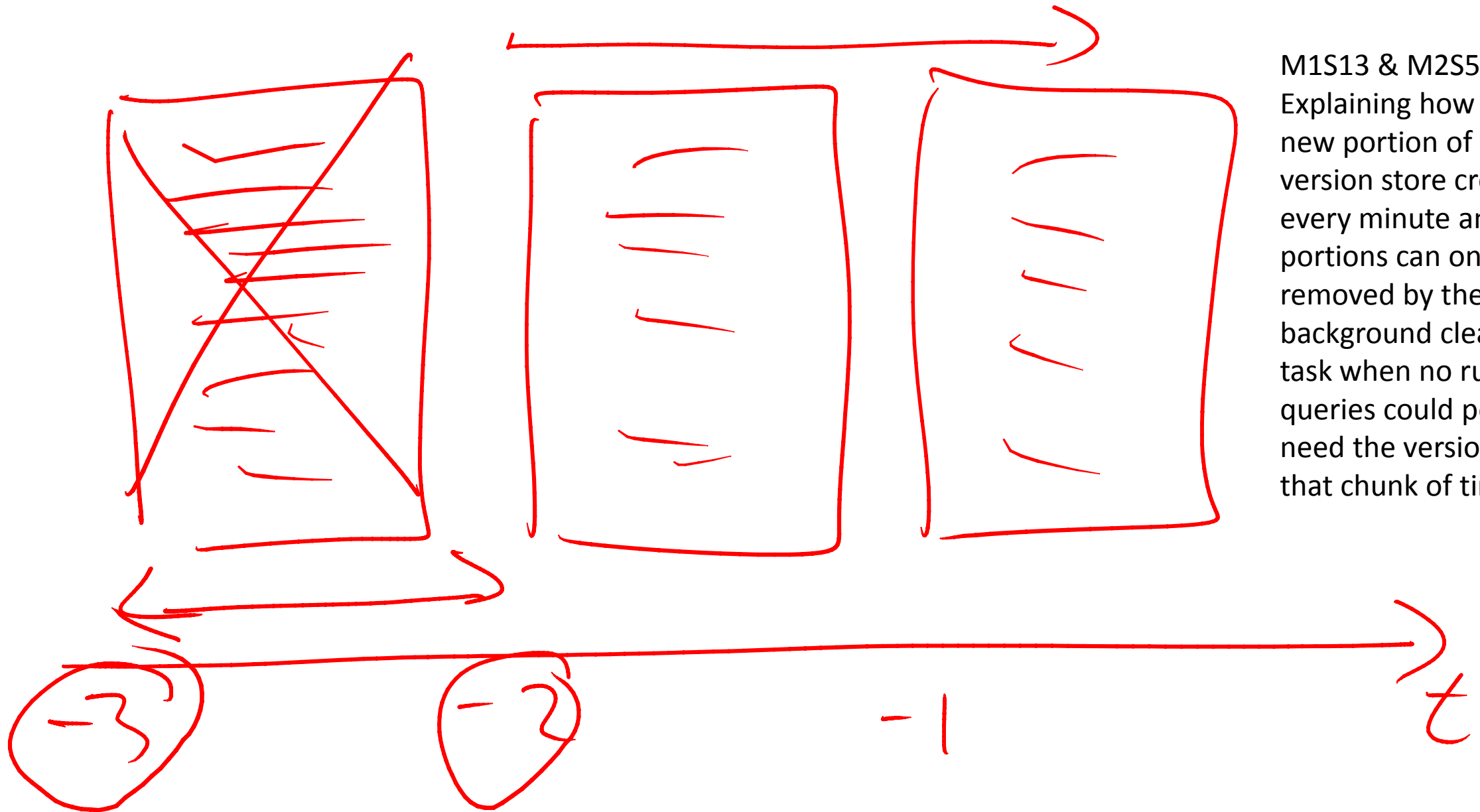




M1S12 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.

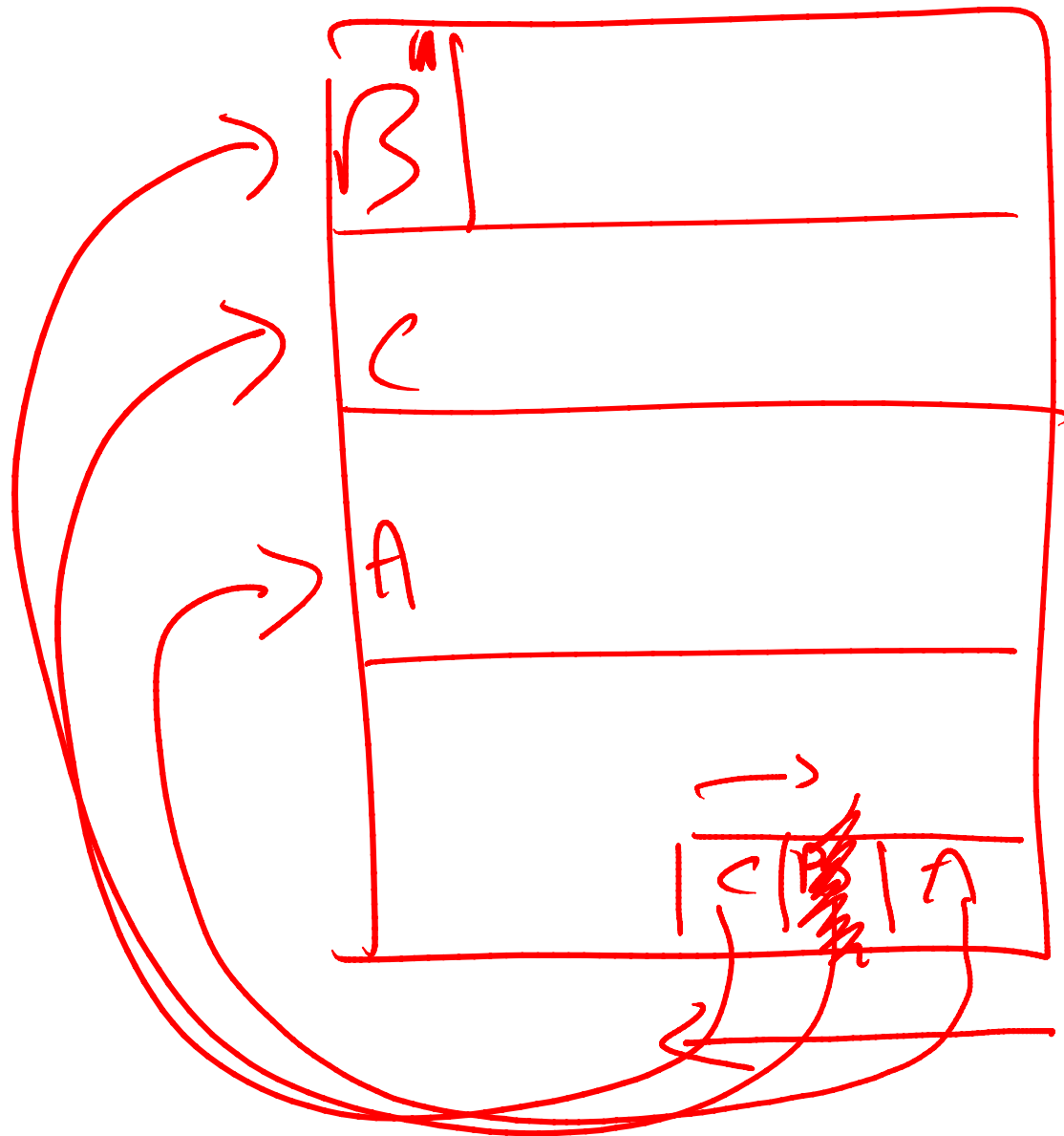


M1S13 When an update occurs to a record and snapshot isolation is enabled, the pre-change record is copied into the version store in tempdb. The changed record has a 14-byte tag on the end with the timestamp and pointer to previous record version. A chain of previous versions is possible.



M1S13 & M2S54

Explaining how there's a new portion of the version store created every minute and these portions can only be removed by the background cleanup task when no running queries could possibly need the version from that chunk of time.



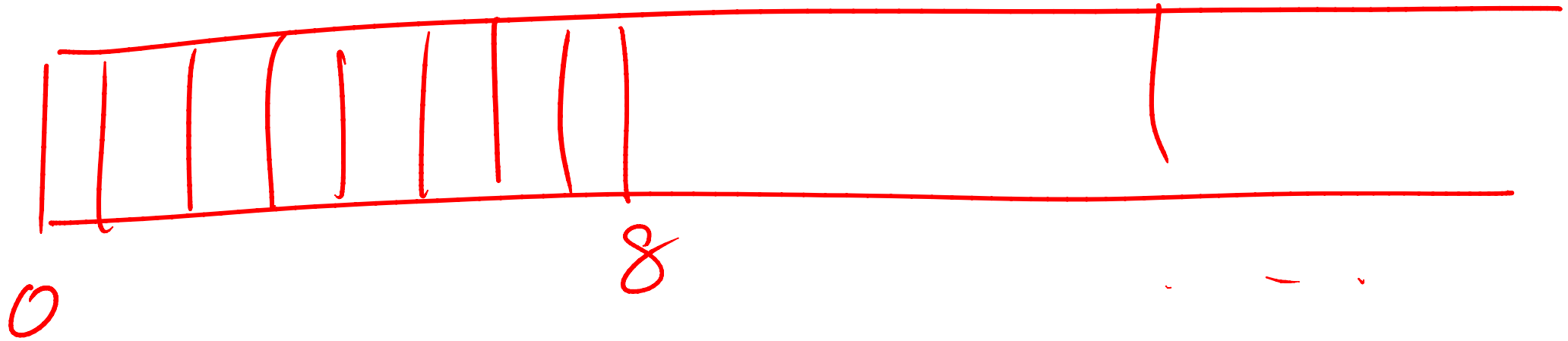
LOP\_EXPUNCE -  
GHOST

SLOT ARRAY

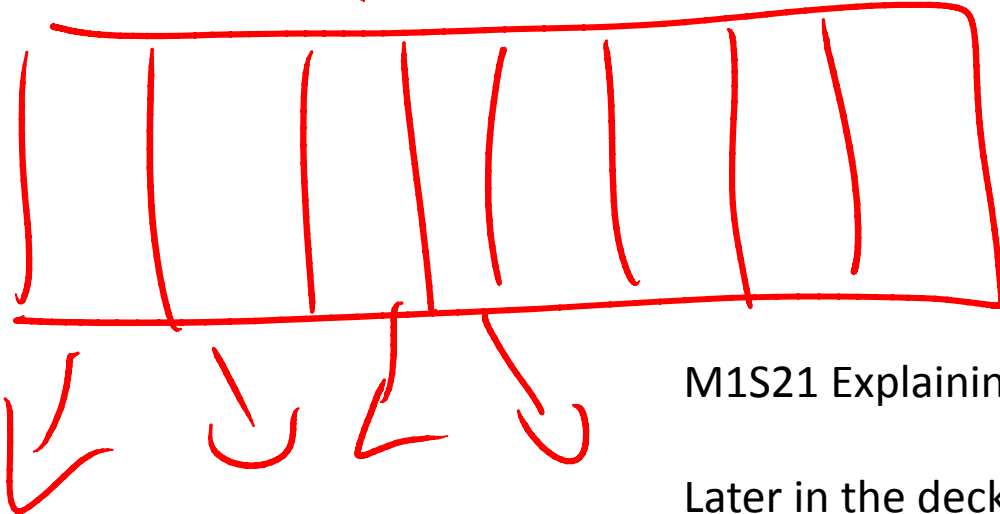
M1S14 When a record is deleted, it's marked as a ghost record. The ghost cleanup task does not physically delete the records – it removes the slot array entry that points to the record on the page. I.e. the record is unmarked as being a record and the area on the page is now free space – that just happens to have bits and bytes that used to be a valid record.

DBCC  
TRACEON (662, -1)  
            
3605

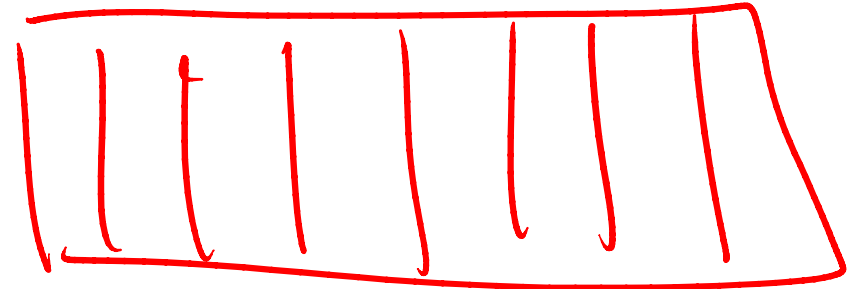
M1S14 How to enable the trace flag globally to watch the ghost cleanup task. You may also need to enable 3605 as well. That one allows debug traceprints to go to the error log.



MIXED



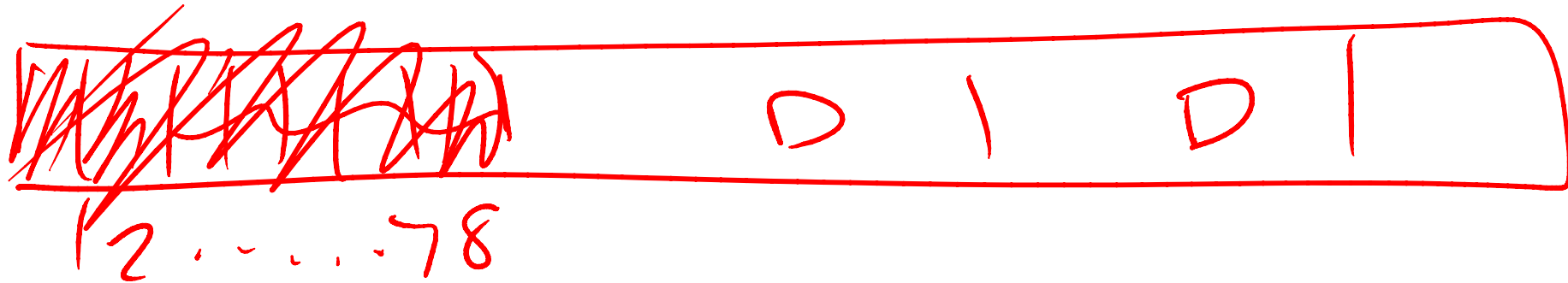
DEDICATED



M1S21 Explaining the difference between mixed and dedicated extents.

Later in the deck I refine 'object' to really be 'allocation unit'.

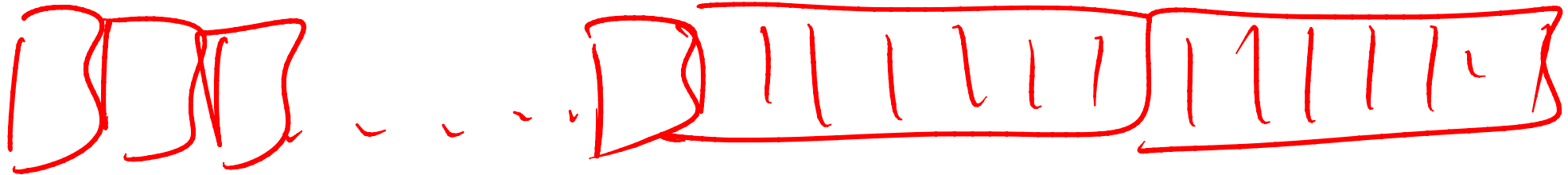
DO ALL SINGLE PAGE ALLOCS  
REMAIN - NO



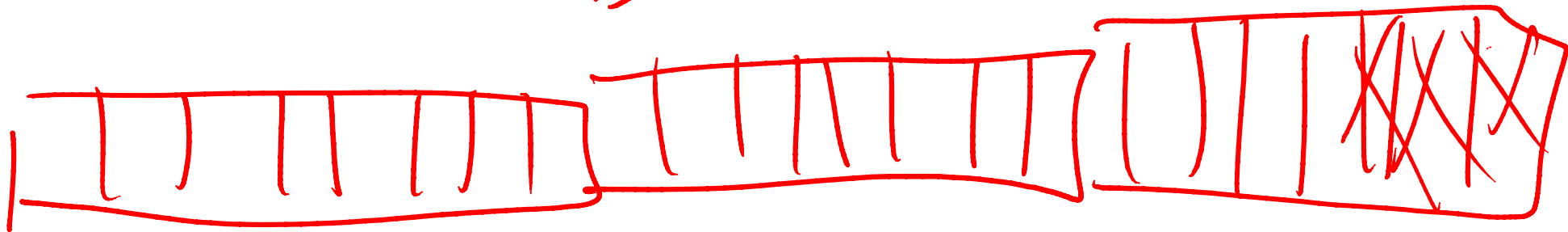
M1 Question on whether all single page allocations remain in a table. I.e. if one of the first 8 single-pages is deallocated, is another one allocated in its place? No.

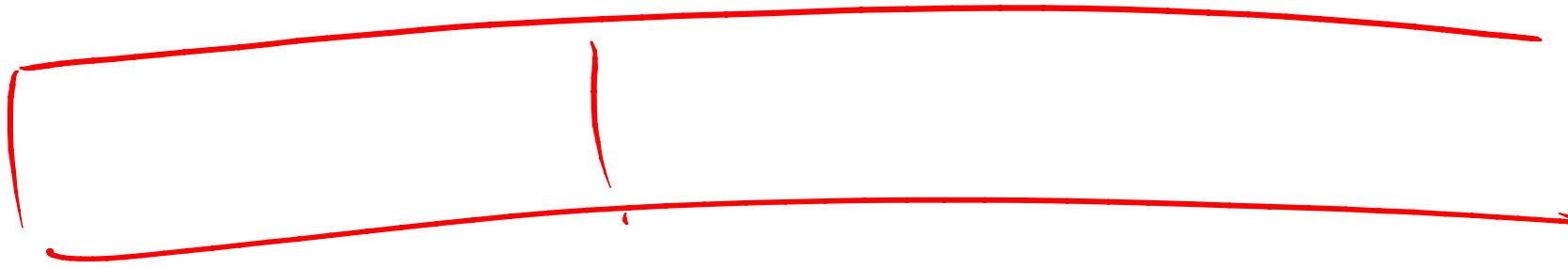
M1: Question on allocations during bulk load. Special case: dedicated extents are allocated and all pages in the extent are marked allocated at once in the PFS. At the end of the load, the final extent is examined and any unused pages are marked deallocated in the PFS.

REGULAR INSERT



BULK INSERT

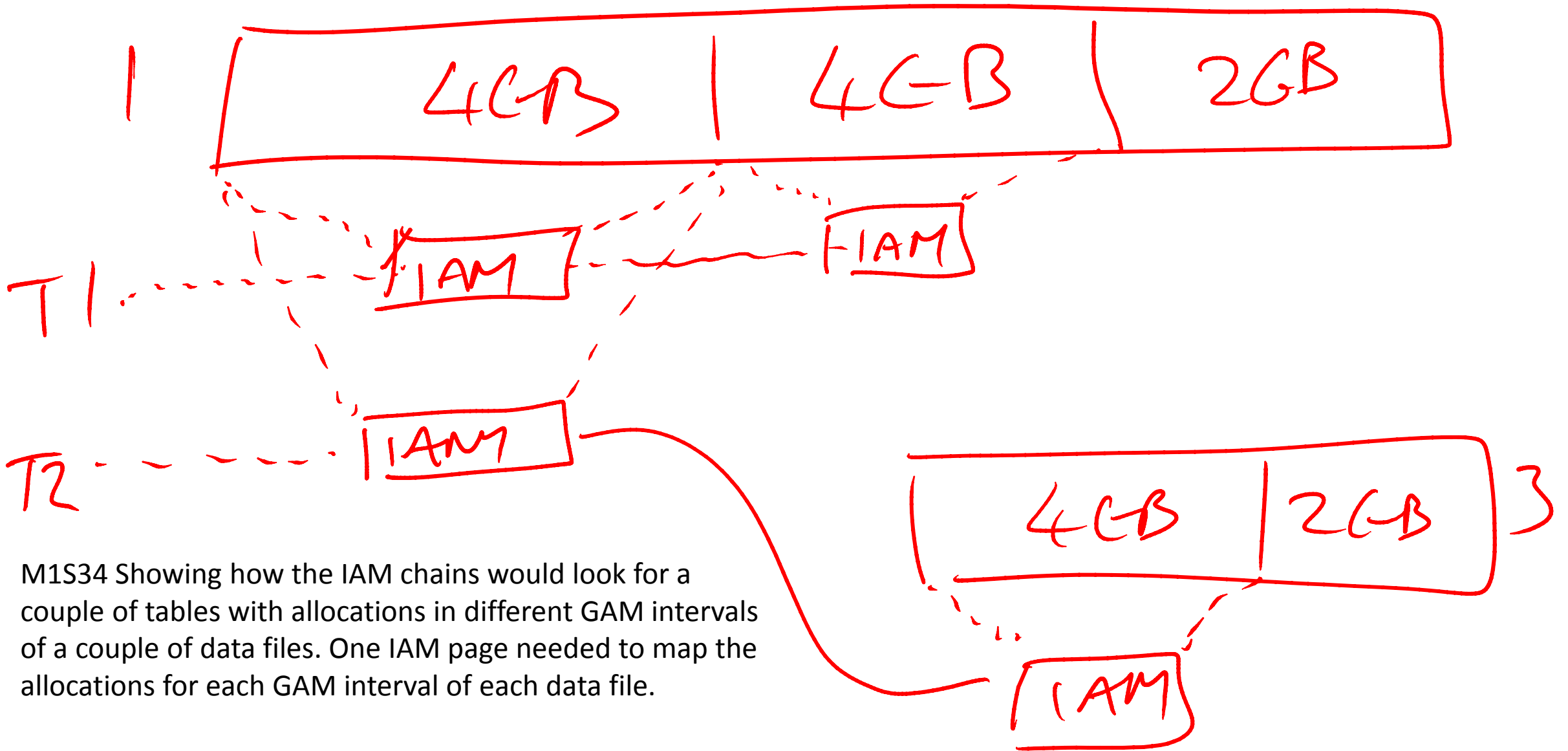




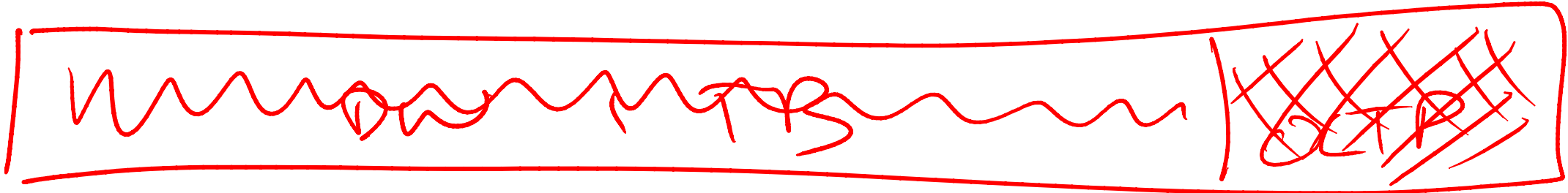
0 — FH  
1 — PFS  
2 — GAM  
3 — SGAM  
4 — UNUSED

5 — UNUSED  
6 — DIFF  
7 — ML

M1 The page types in the first extent in a data file



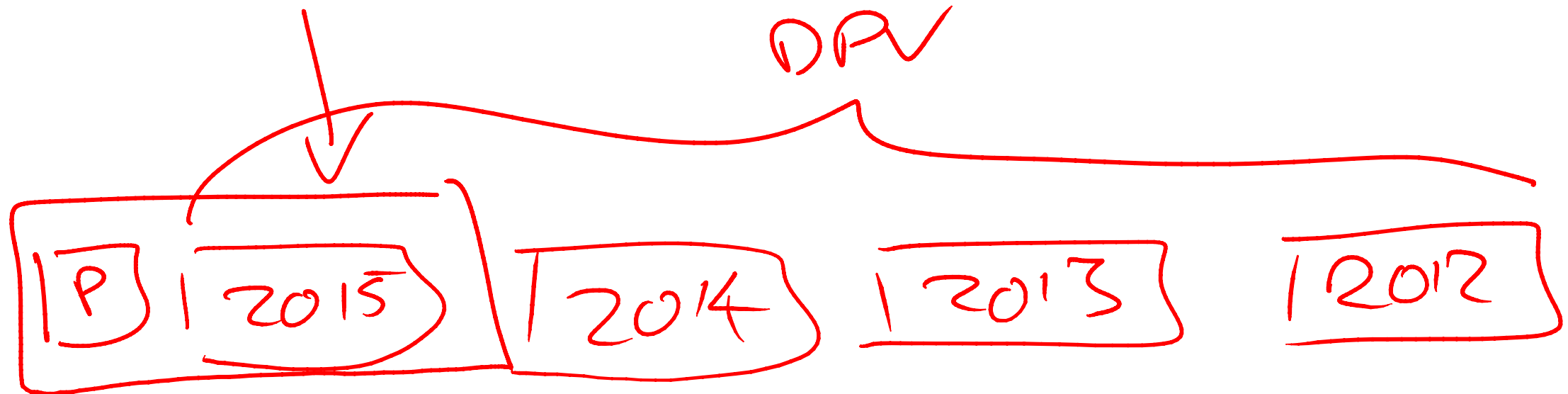
M1S34 Showing how the IAM chains would look for a couple of tables with allocations in different GAM intervals of a couple of data files. One IAM page needed to map the allocations for each GAM interval of each data file.

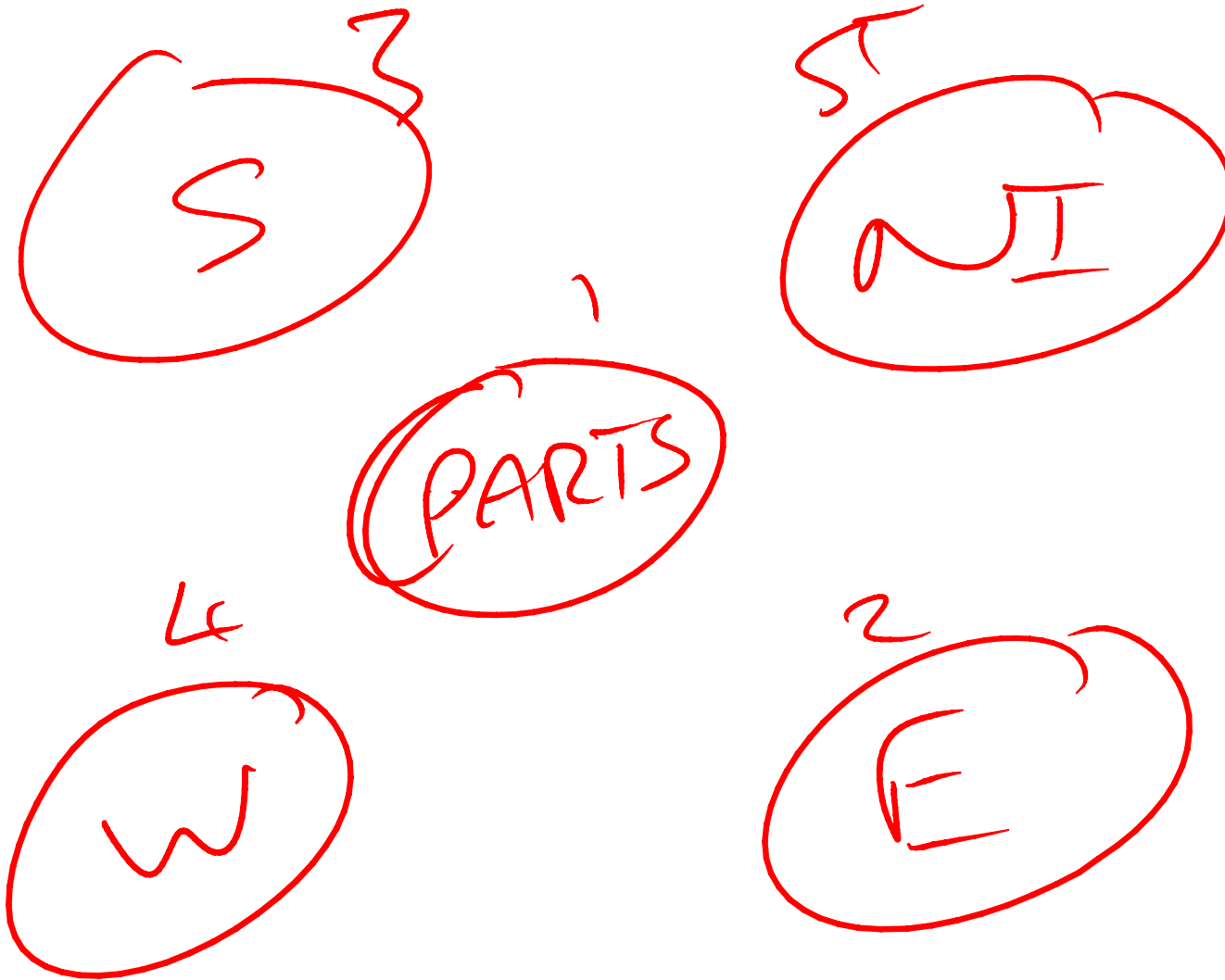


2012

2015

M2S6 Splitting a large database up into multiple filegroups can enable you to do small, targeted restores or even a partial restore from scratch – in the event that the database is damaged or destroyed.

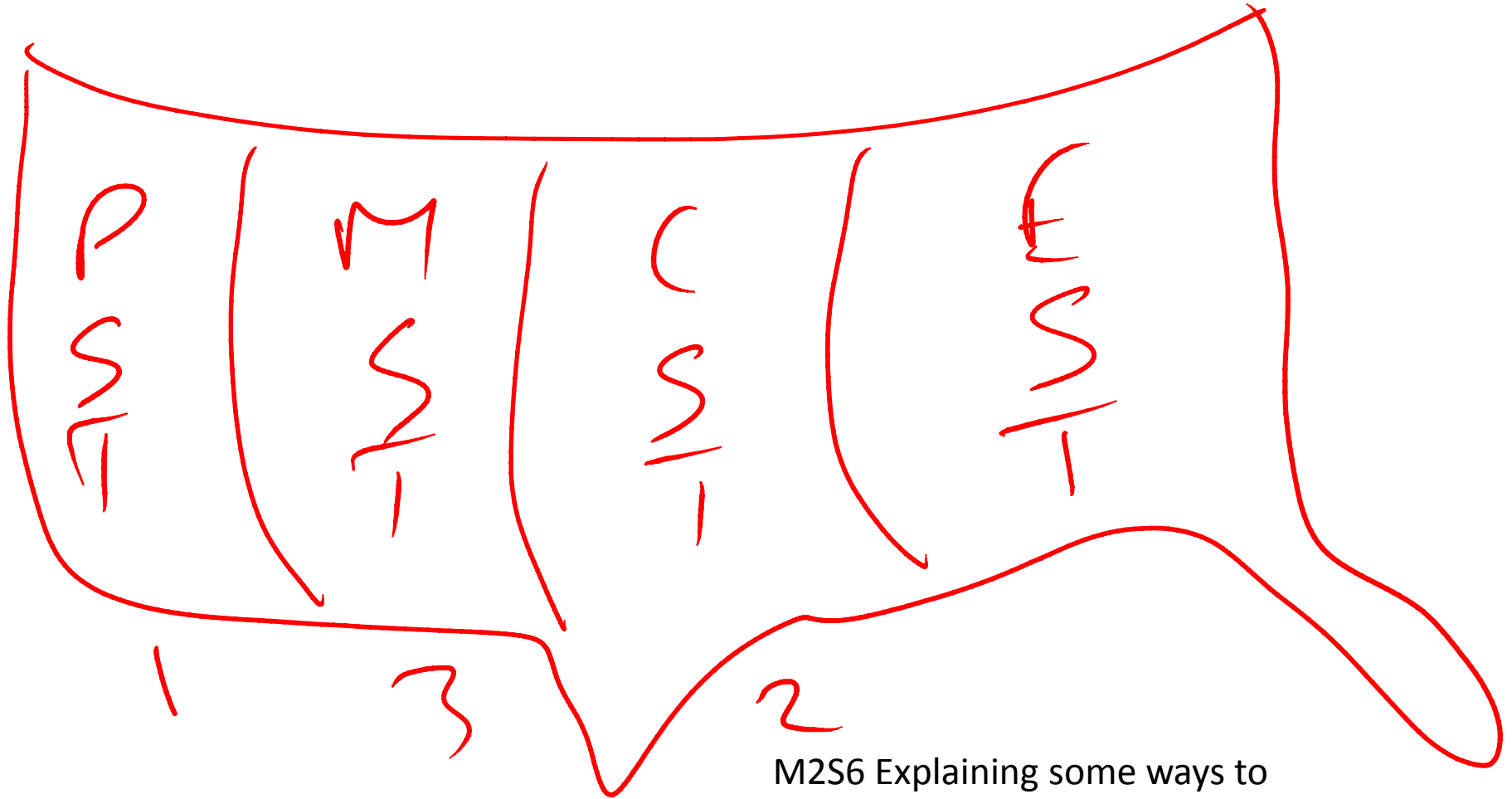




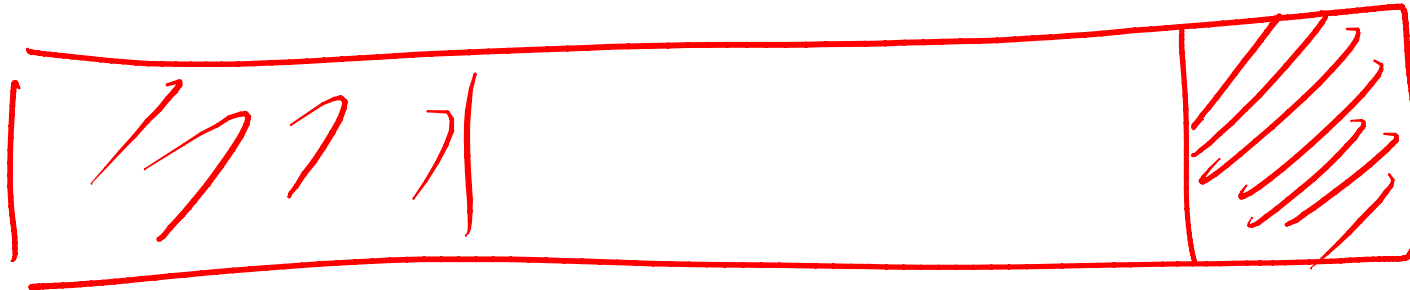
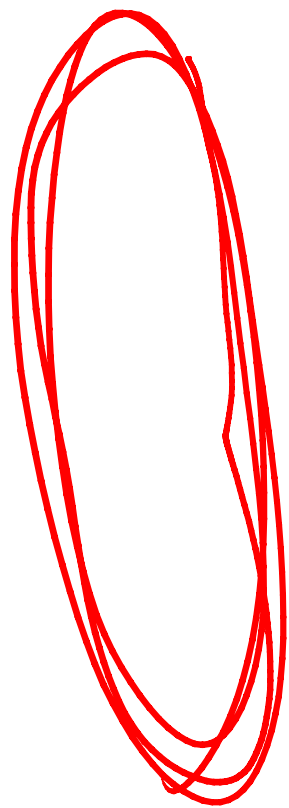
M2S6 Explaining some ways to split up a database to allow targeted restore. Consider time zone, population density, or other demographics.

D

8-6



M2S6 Explaining some ways to split up a database to allow targeted restore. Consider time zone, population density, or other demographics.

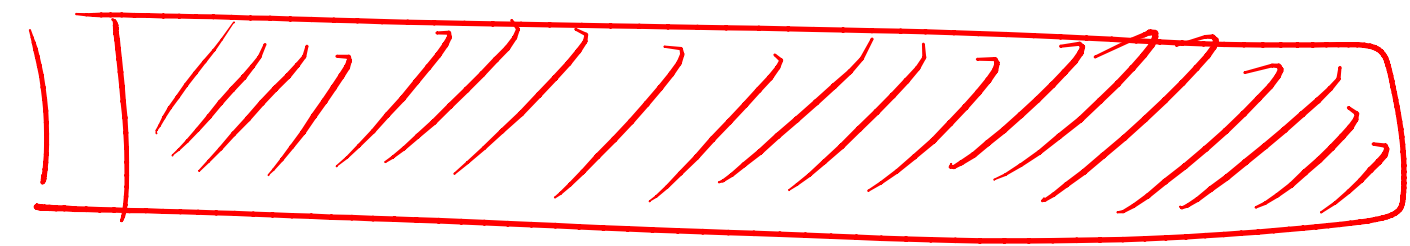


WEIGHTING

~~8148~~ 8147

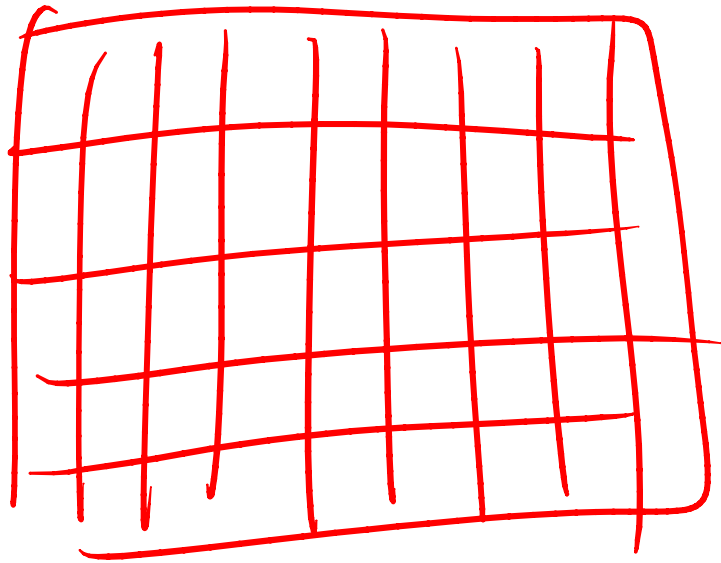


~~8148~~ 8147

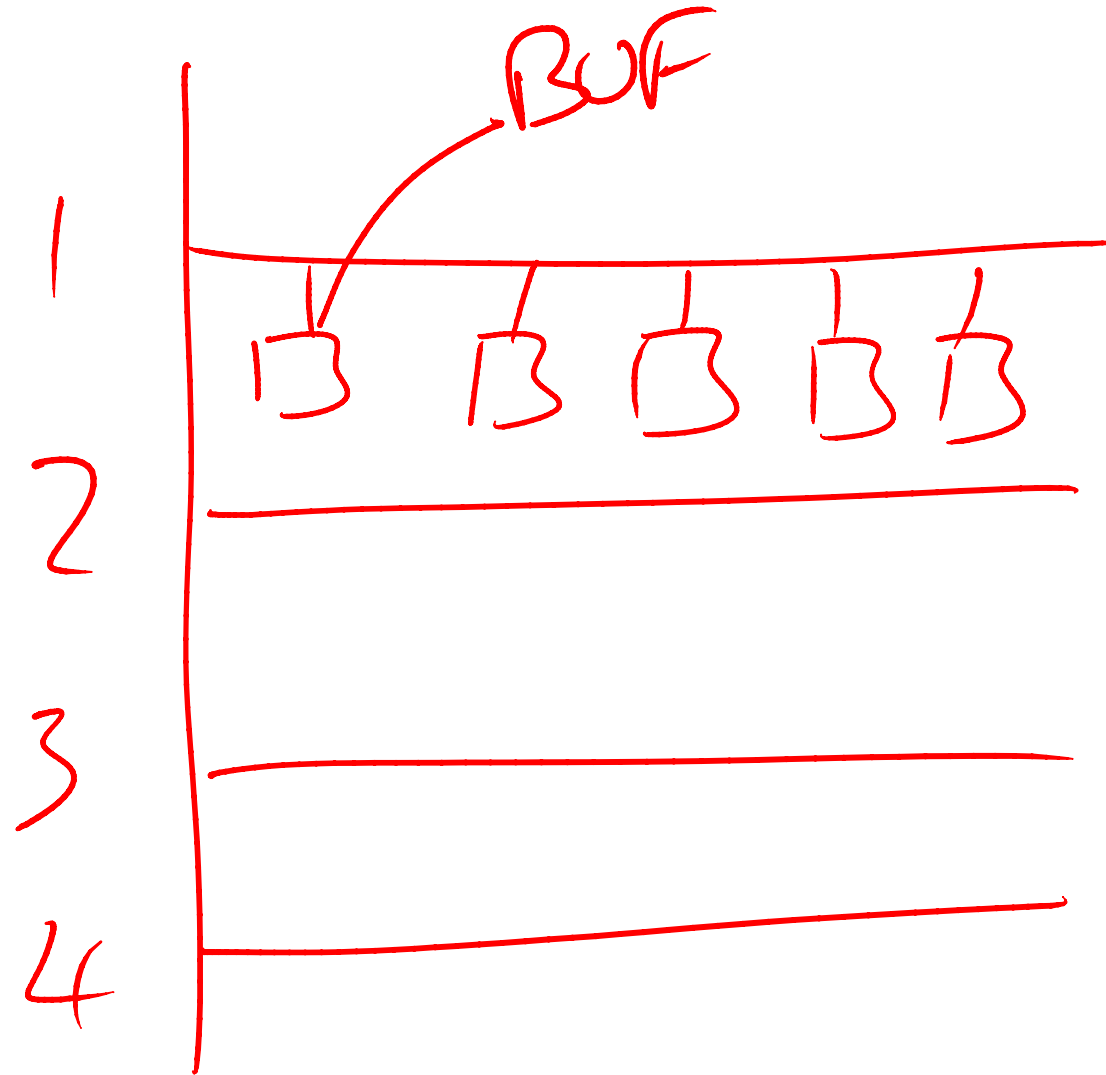


↓     ✓

M2S18 Explaining round-robin allocation and proportional fill. Each file has a proportional fill weighting. Allocations happen proportionally more frequently from files with more free space.

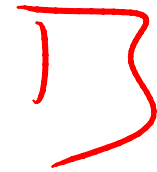
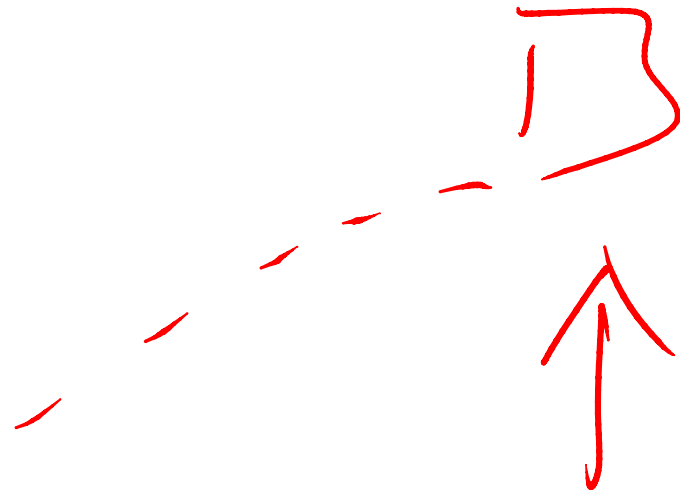


M2S19 Explaining the hash lists per database that allow the buffer pool to easily determine whether a page is already in memory or not



P. L. E

~~D. C. R~~



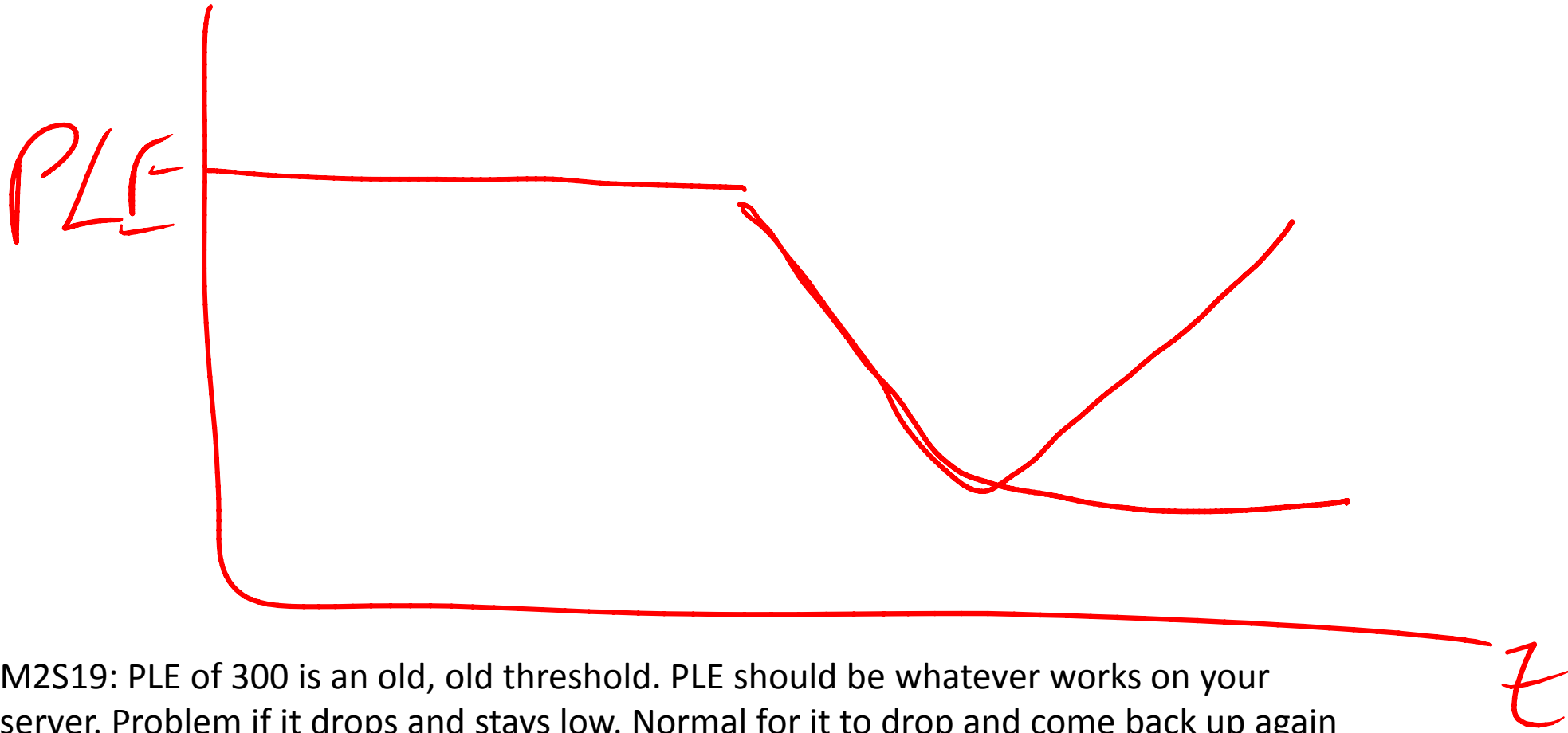
L. R. U. - K,  
K=2



LAZY WRITER

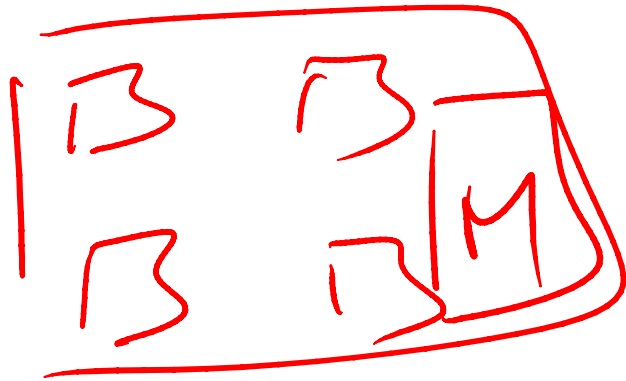
M2S19 Explaining that each page is a point on a big clock face and the lazywriter background process sweeps the clock face implement a Least-Recently-Used algorithm that helps it maintain free space by discarding unused pages.

$$PLE = 300 \approx S_{min}$$

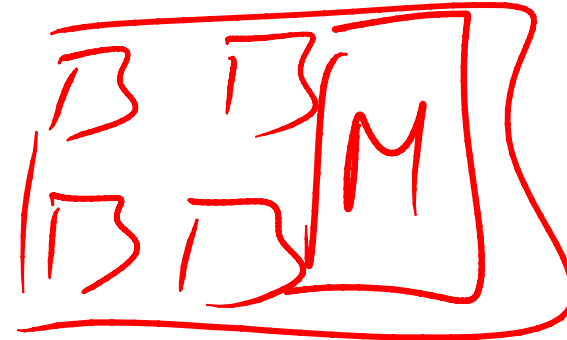
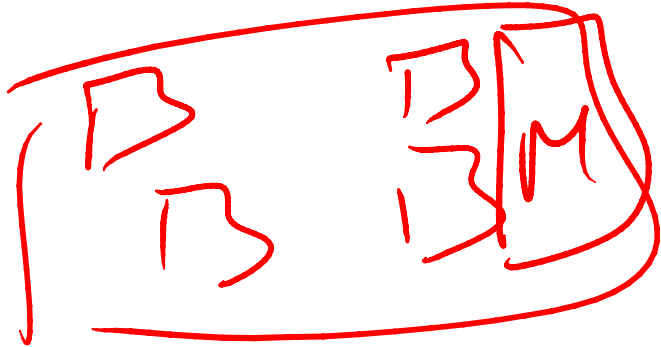


M2S19: PLE of 300 is an old, old threshold. PLE should be whatever works on your server. Problem if it drops and stays low. Normal for it to drop and come back up again after some operations.

# NUMA



M2S19: NUMA – Non Uniform Memory Access. Each processor (or group) has local memory, containing a partition of the buffer pool. 4 NUMA nodes = 4 equal-size partitions of the buffer pool.

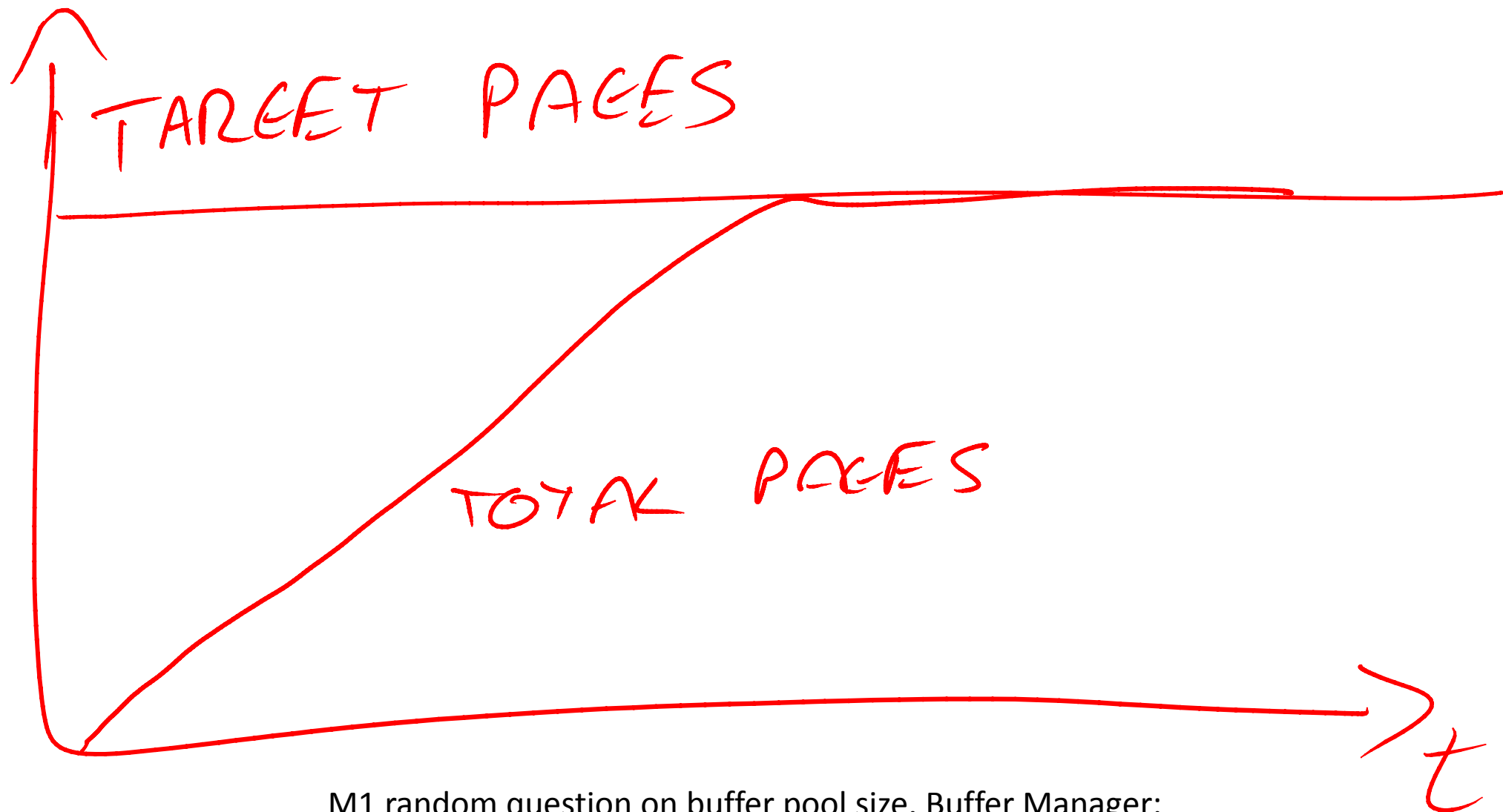


# 4 NUMA NODES

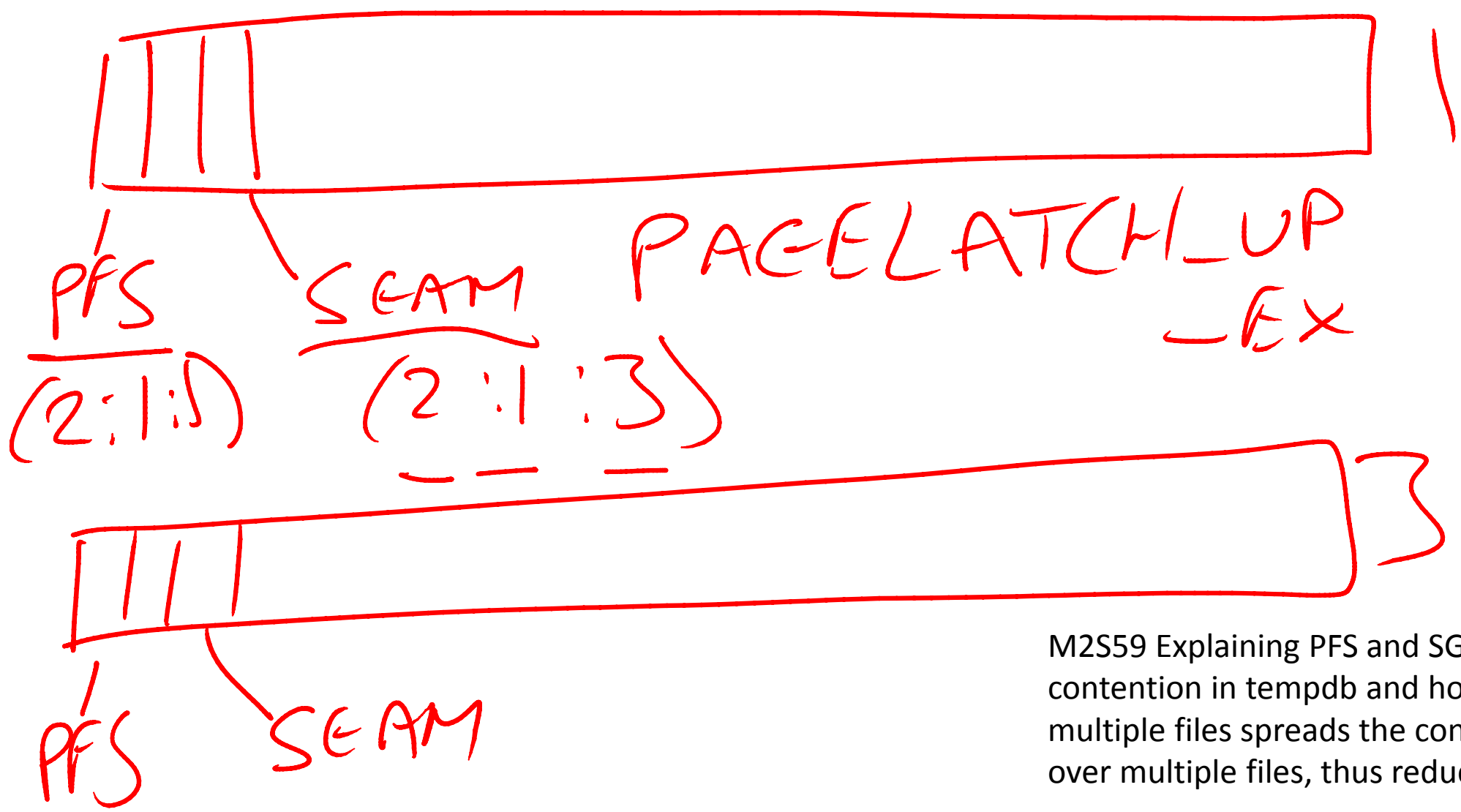
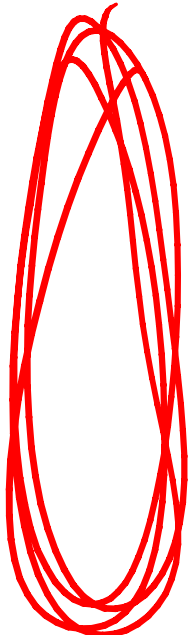
Buffer Mgr: PLE = harmonic mean  
of all B.Pool. nodes

Buffer Node: PLE 1  
2  
3  
4

M2S19: If you have NUMA, you need to monitor the PLE in each Buffer Node perfmon object, not the Buffer Manager PLE.

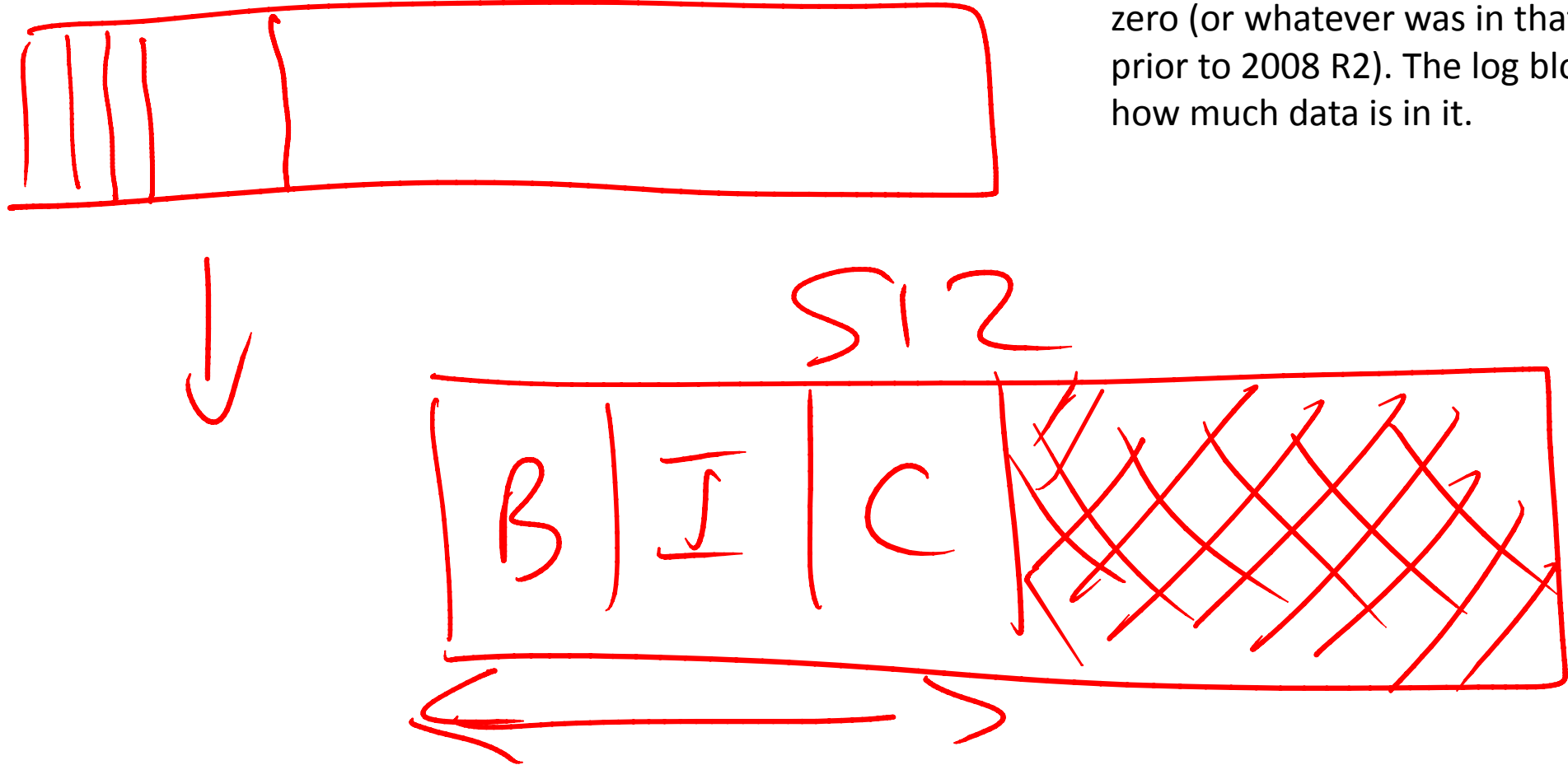


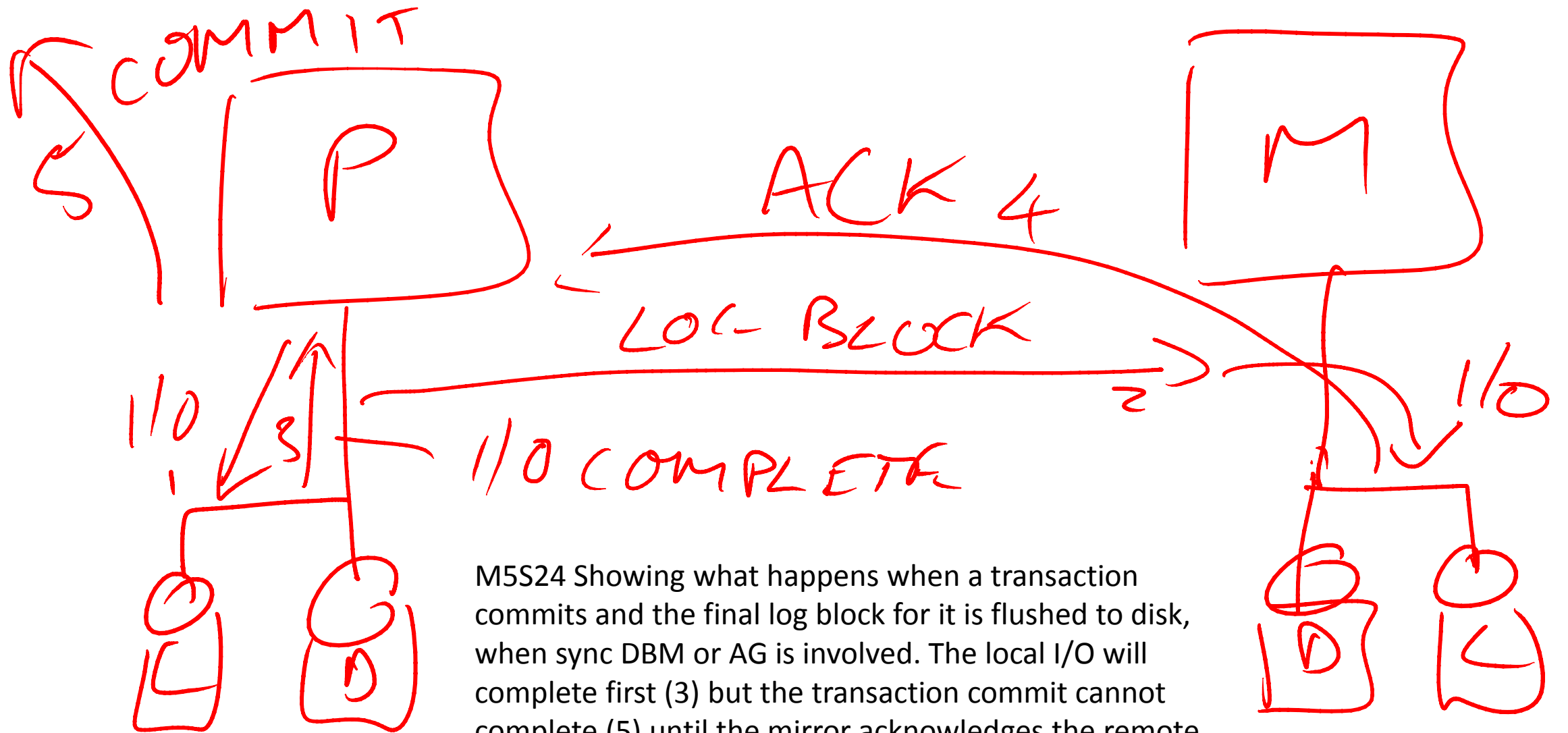
M1 random question on buffer pool size. Buffer Manager:  
Target Pages is the target size, Total Pages is current size. Graph  
above shows instance start up.



M2S59 Explaining PFS and SGAM contention in tempdb and how adding multiple files spreads the contention over multiple files, thus reducing it.

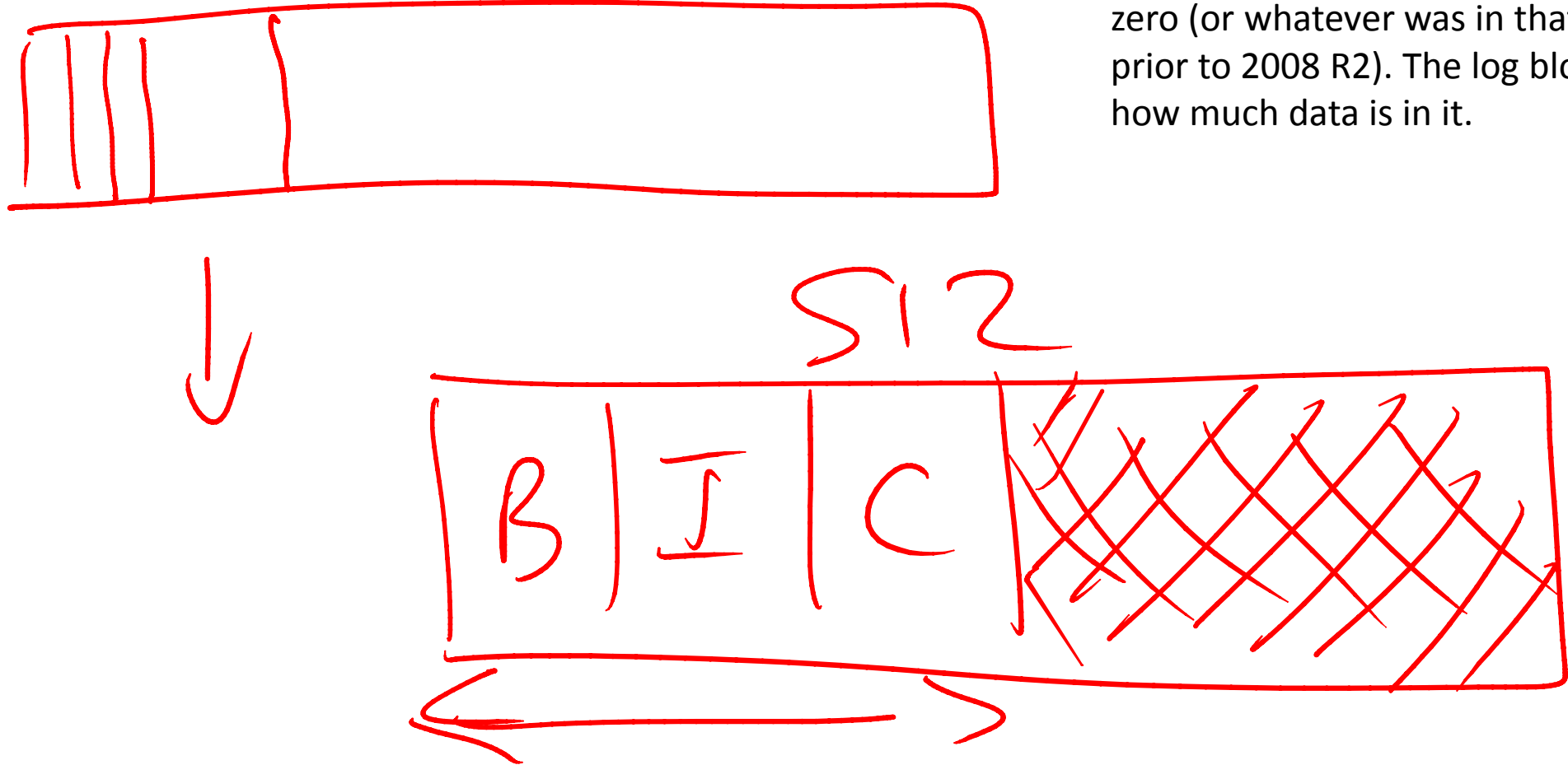
M5S24 Discussing the contents of a log block if they don't fill up right to a 512-byte boundary. Unused bytes may be zero (or whatever was in that memory prior to 2008 R2). The log block knows how much data is in it.

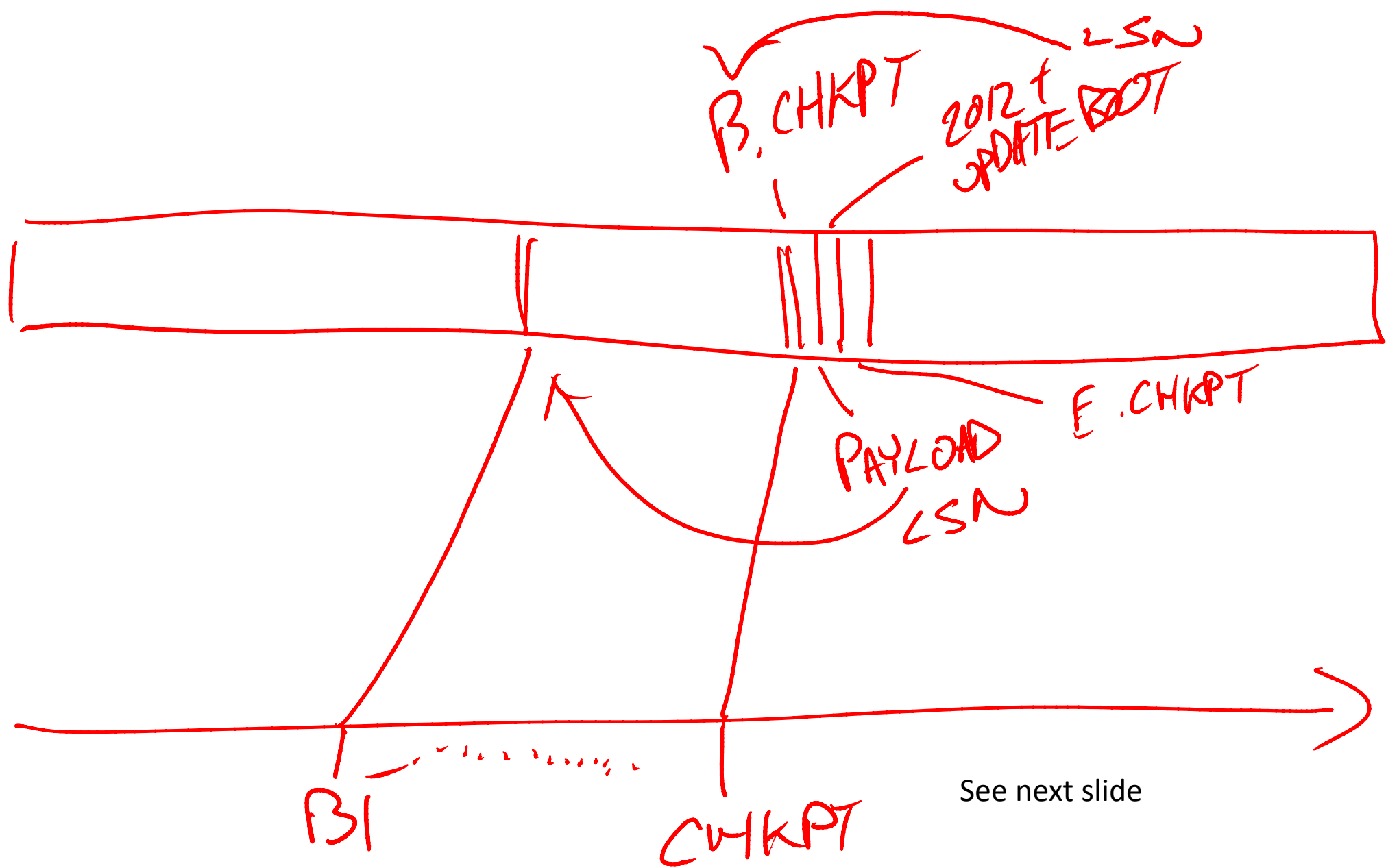




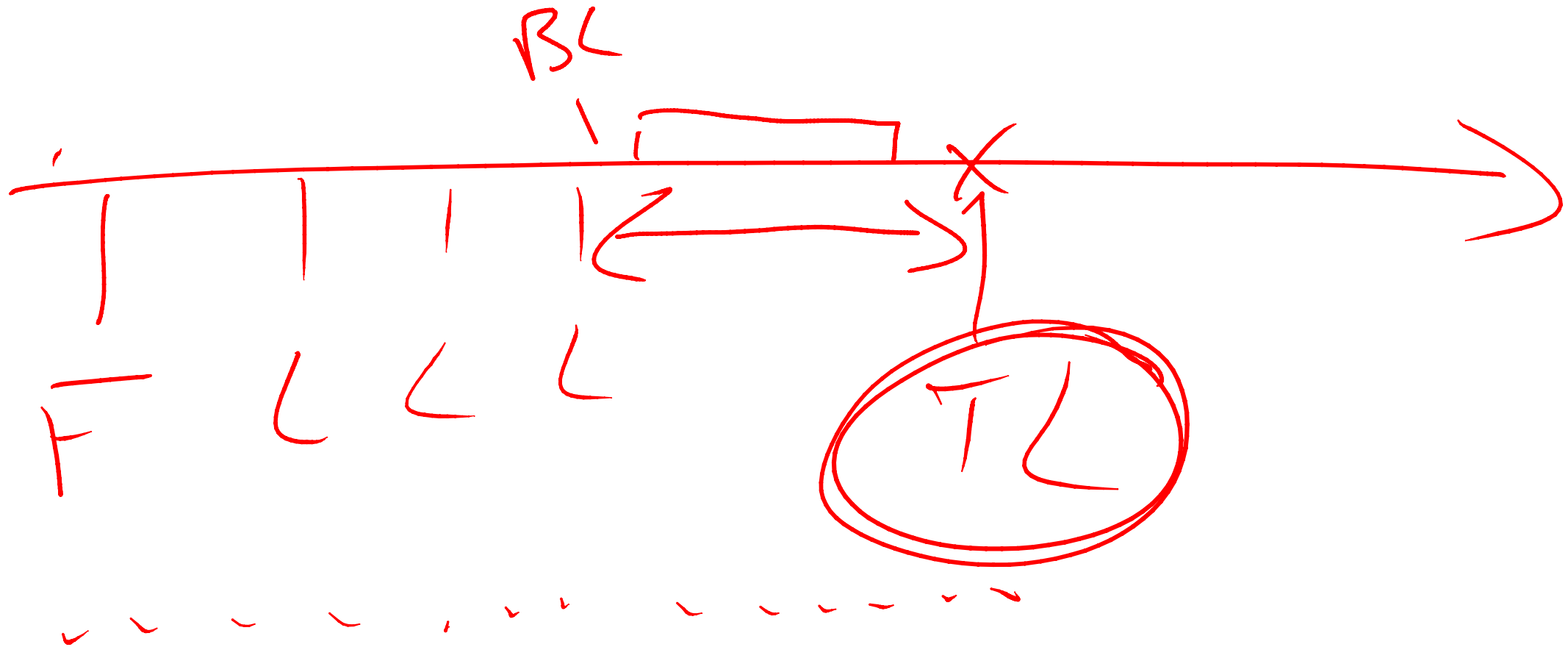
M5S24 Showing what happens when a transaction commits and the final log block for it is flushed to disk, when sync DBM or AG is involved. The local I/O will complete first (3) but the transaction commit cannot complete (5) until the mirror acknowledges the remote I/O (4).

M5S24 Discussing the contents of a log block if they don't fill up right to a 512-byte boundary. Unused bytes may be zero (or whatever was in that memory prior to 2008 R2). The log block knows how much data is in it.

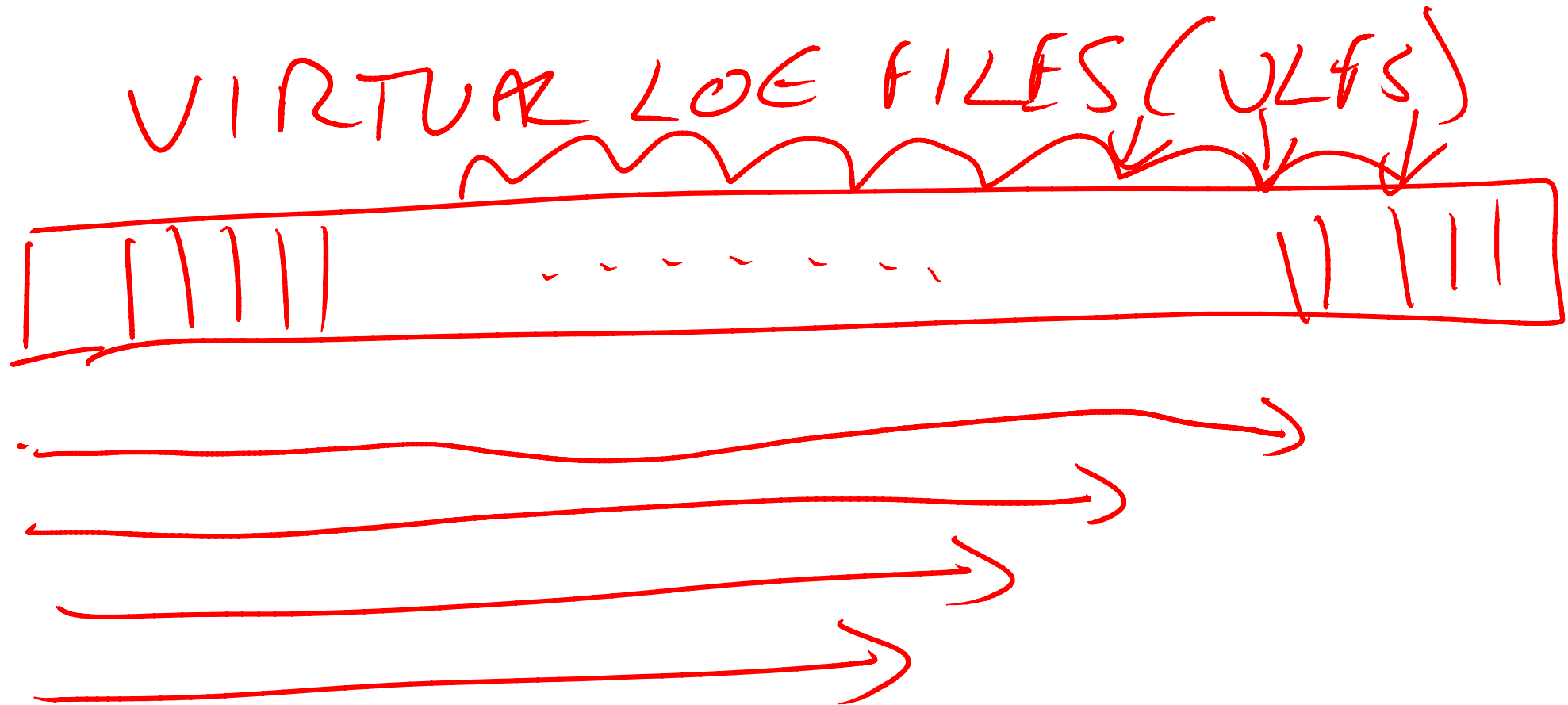




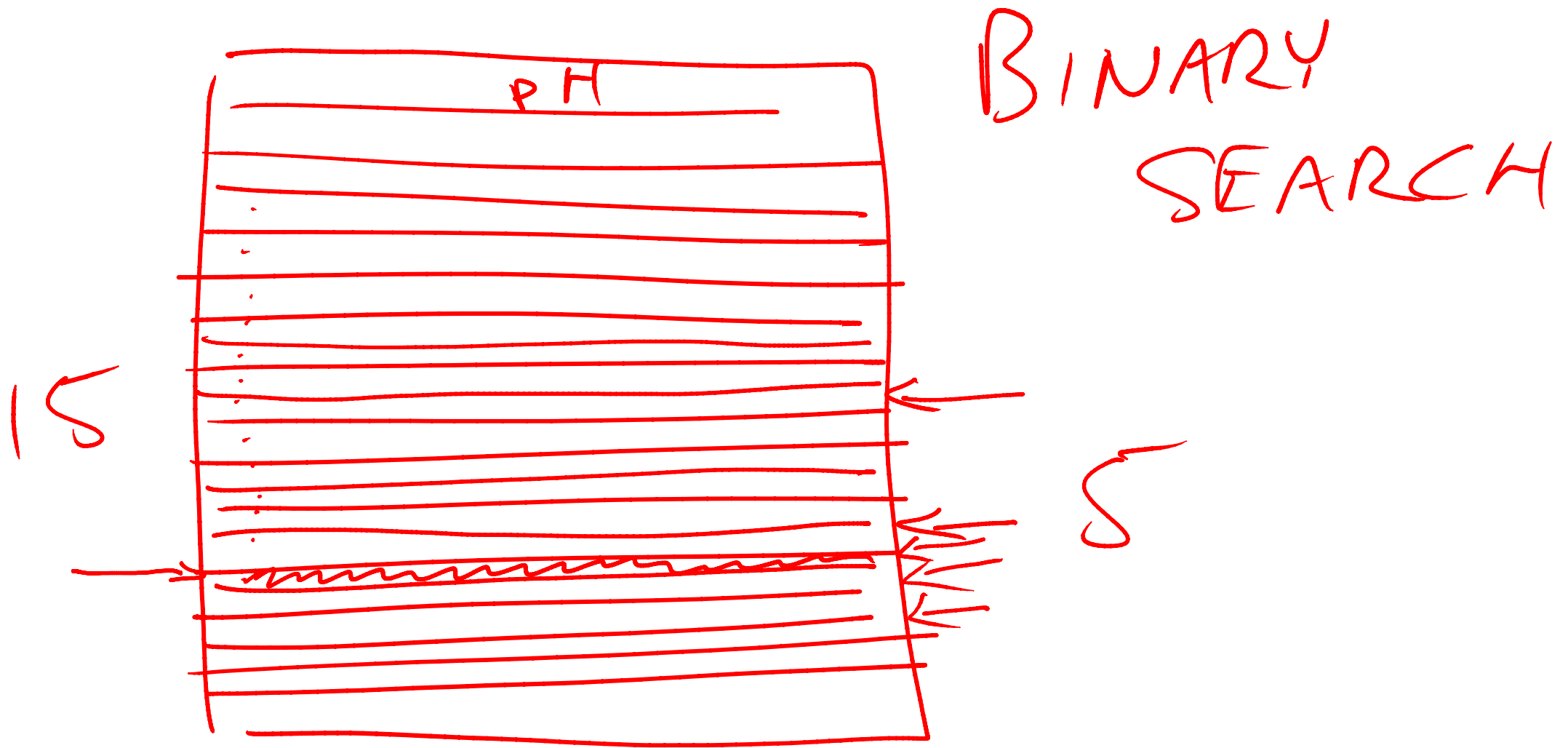
- The log records for a checkpoint:
  - Begin checkpoint
  - Payload (containing LSN of LOP\_BEGIN\_XACT log records for active transactions at that instant)
  - 2012+: updating boot page (happens on all versions, only logged in 2012+)
  - End checkpoint



M5S61 Explaining that if a minimally-logged operation is present in a log backup, you cannot do a point-in-time restore to any time covered by that backup (except the start or end of that time period). You also cannot do a tail-of-the-log backup after a crash that destroys the data files if there has been a minimally-logged operation since the last log backup. Well in 2008 R2/2012, you can, but the log backup is invalid.

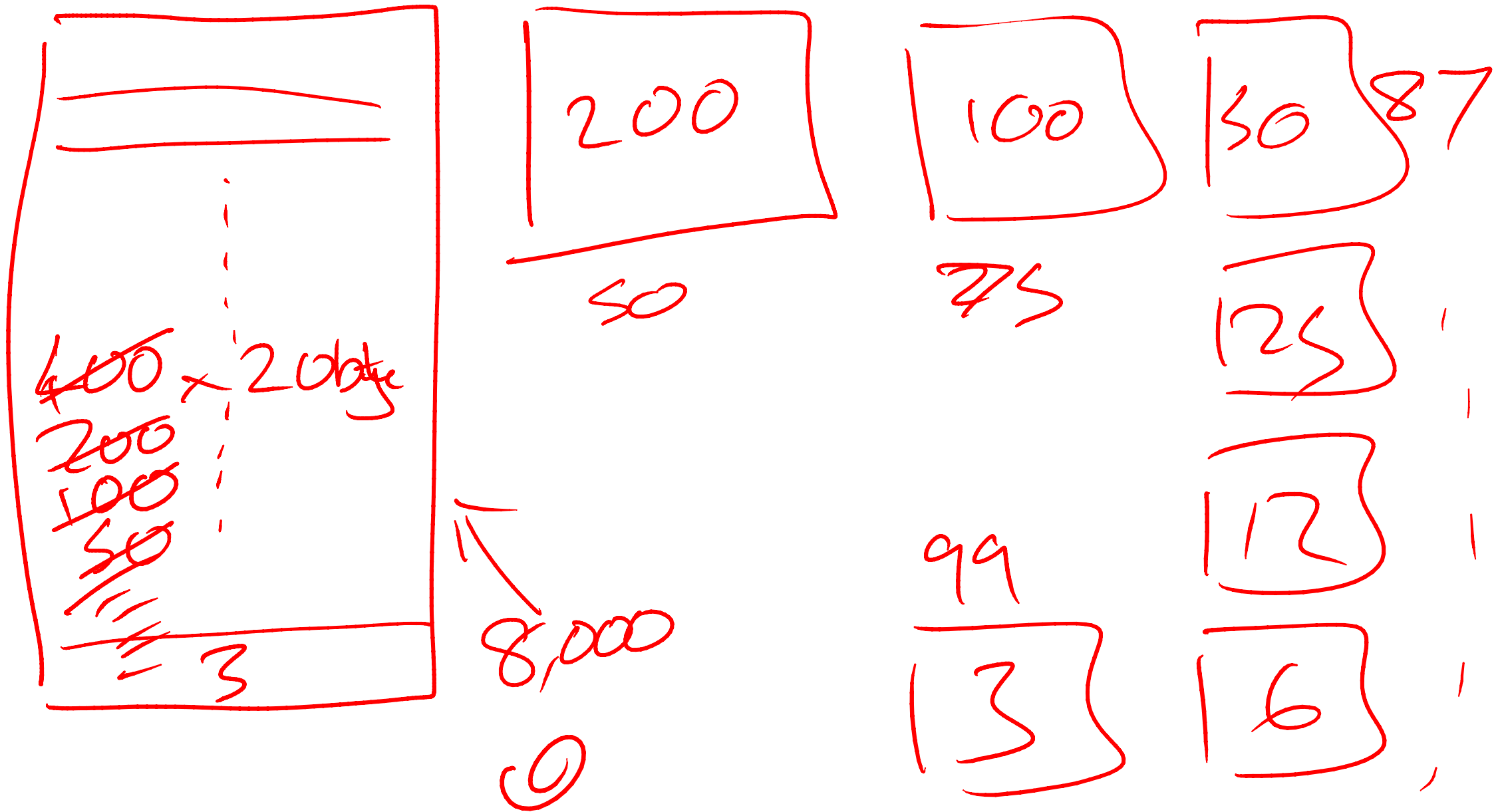


M5S64 VLF fragmentation is bad because of having to search through the VLFs for a particular VLF sequence number. See the KB article link for more info.



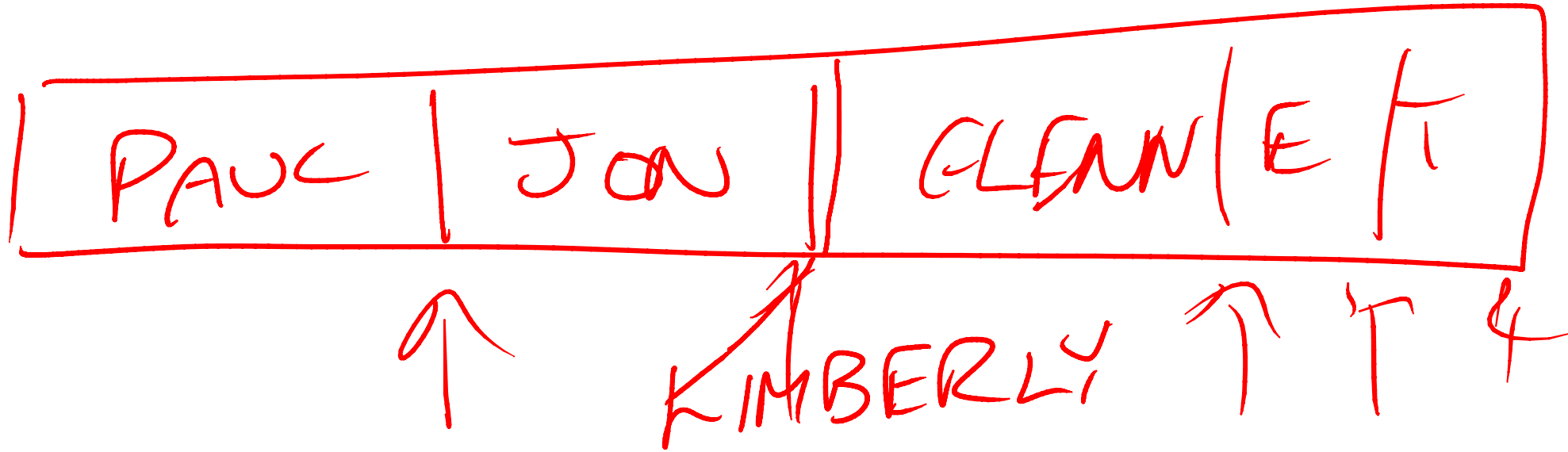
M8S18 Append-only insert pattern fills pages on right-hand side of index. New page allocations in this case are called page splits too – making all methods of tracking page splits largely useless (until 2012 with Xevents).

You can look for LOP\_DELETE\_SPLIT log records to see splits occurring.



M7S28 Discussing multiple splits. With a page with 400 x 20-byte records, if an 8,000-byte record is inserted, it will split multiple times to make room. This can create a page with 90+% free space.

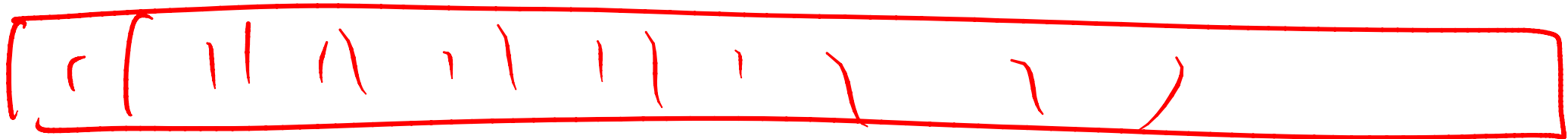
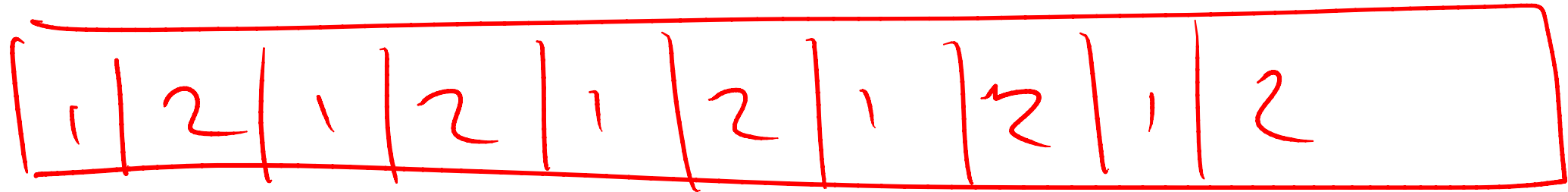
MEMBERID



M7S19 The MySpace story. The 'K' is the tiny portion of comments on Kimberly's page because she wasn't very popular back then... 😊

|      | NO<br>SPLIT | SPLIT | ?   |
|------|-------------|-------|-----|
| 1000 | 1256        | 7928  | x5  |
| 100  | 356         | 7120  | x18 |
| 10   | 268         | 11520 | x45 |

M7S28 The numbers from the page split logging cost demo.



M7S38 Discussing degree of parallelism for index rebuilds. DOP 2 will have two threads, each building a section of b-tree and allocating extents, resulting in the 1,2,1,2 pattern of extents above. DOP 1 gets the best contiguity for if you need to do very large scans (e.g. a big data warehouse)