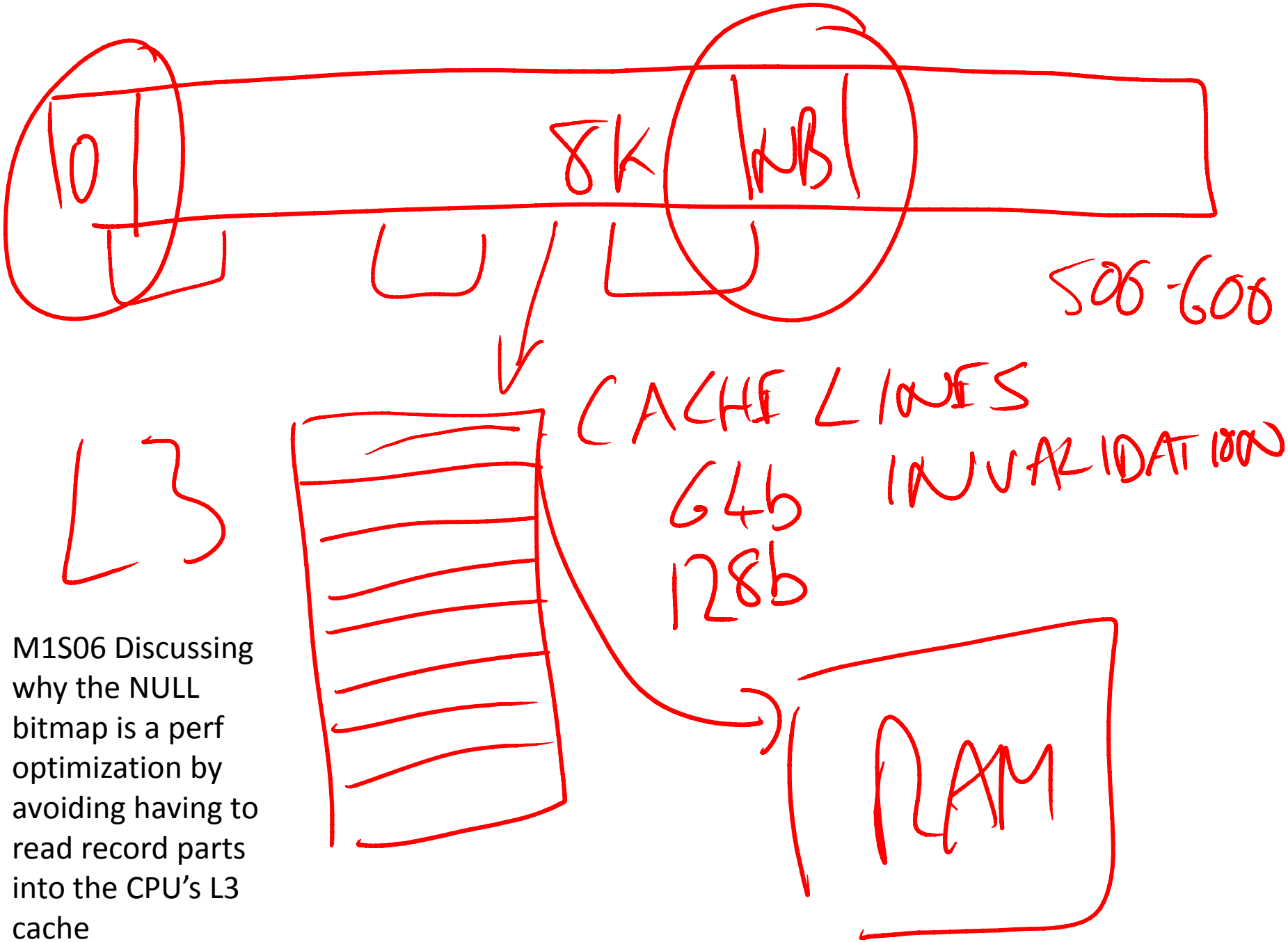
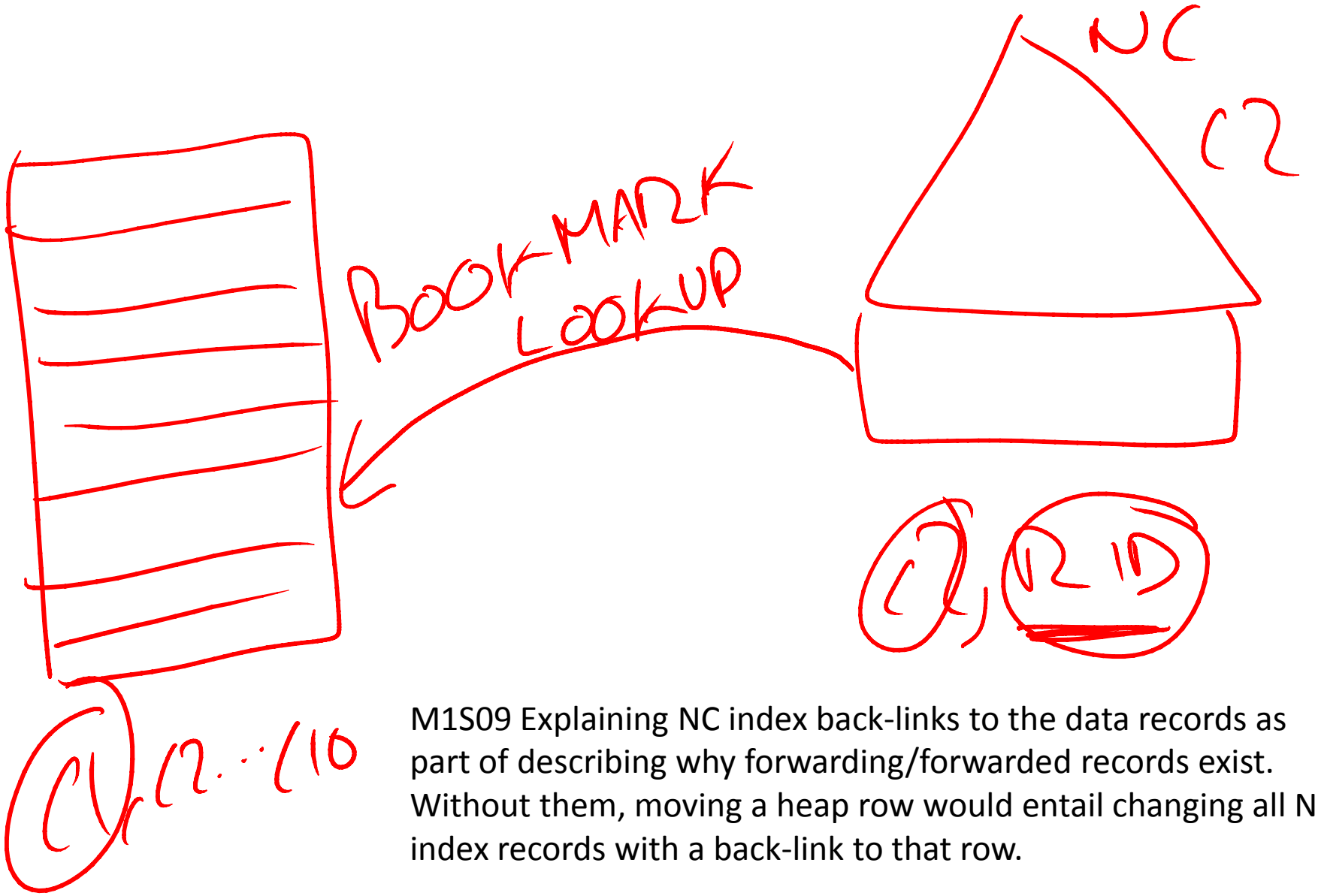
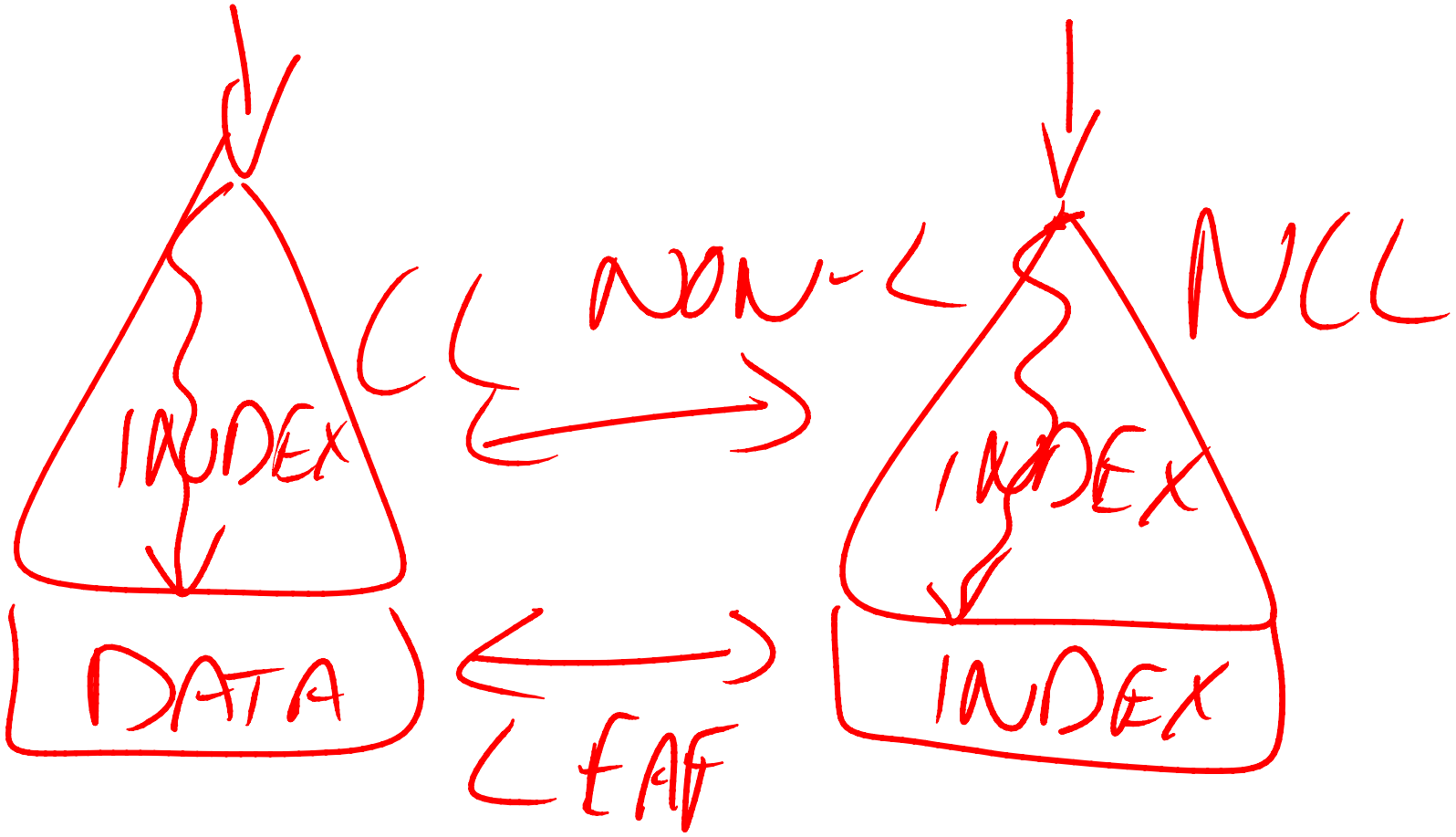


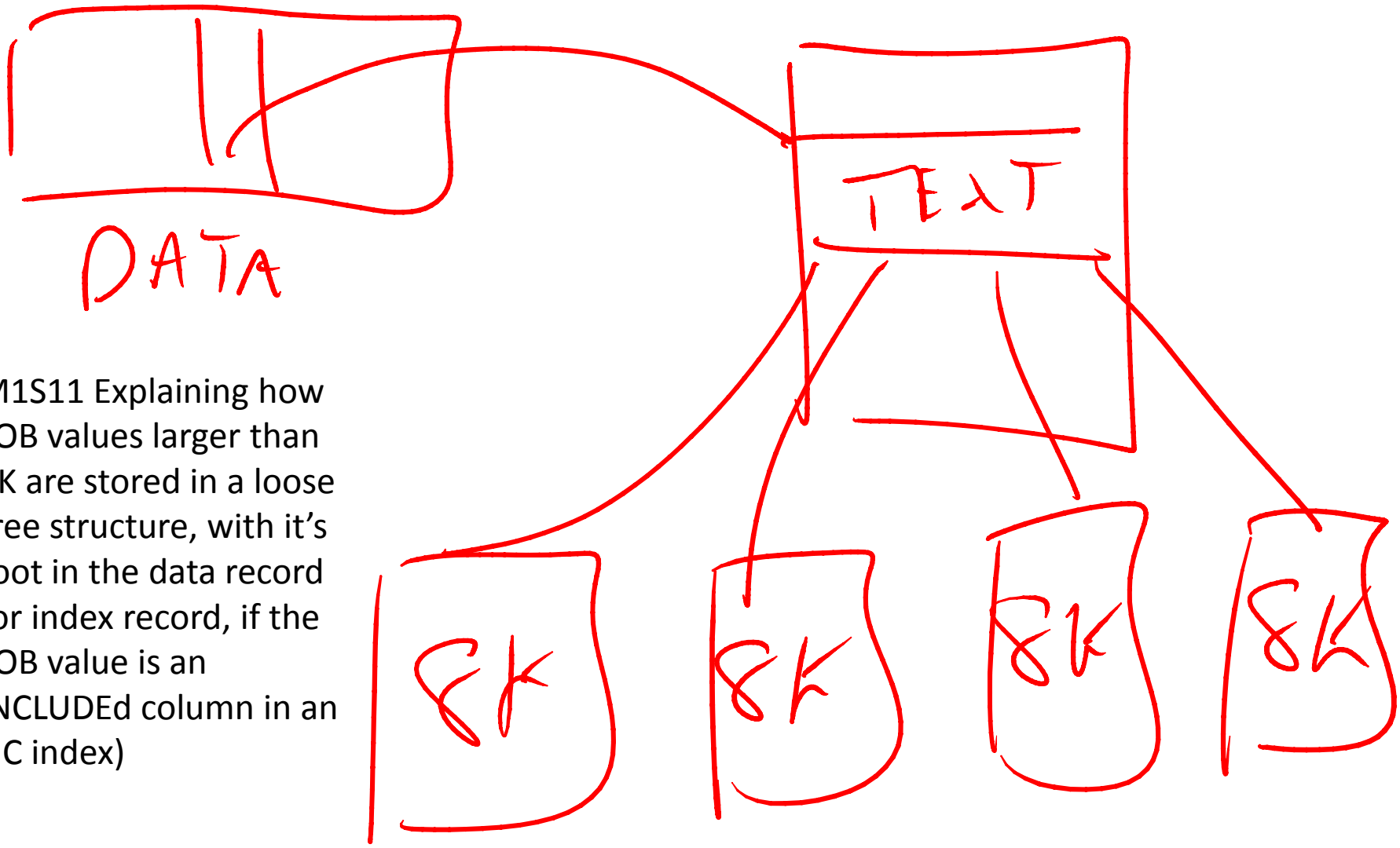
M1S06 Discussing why the NULL bitmap is a perf optimization by avoiding having to read record parts into the CPU's L3 cache



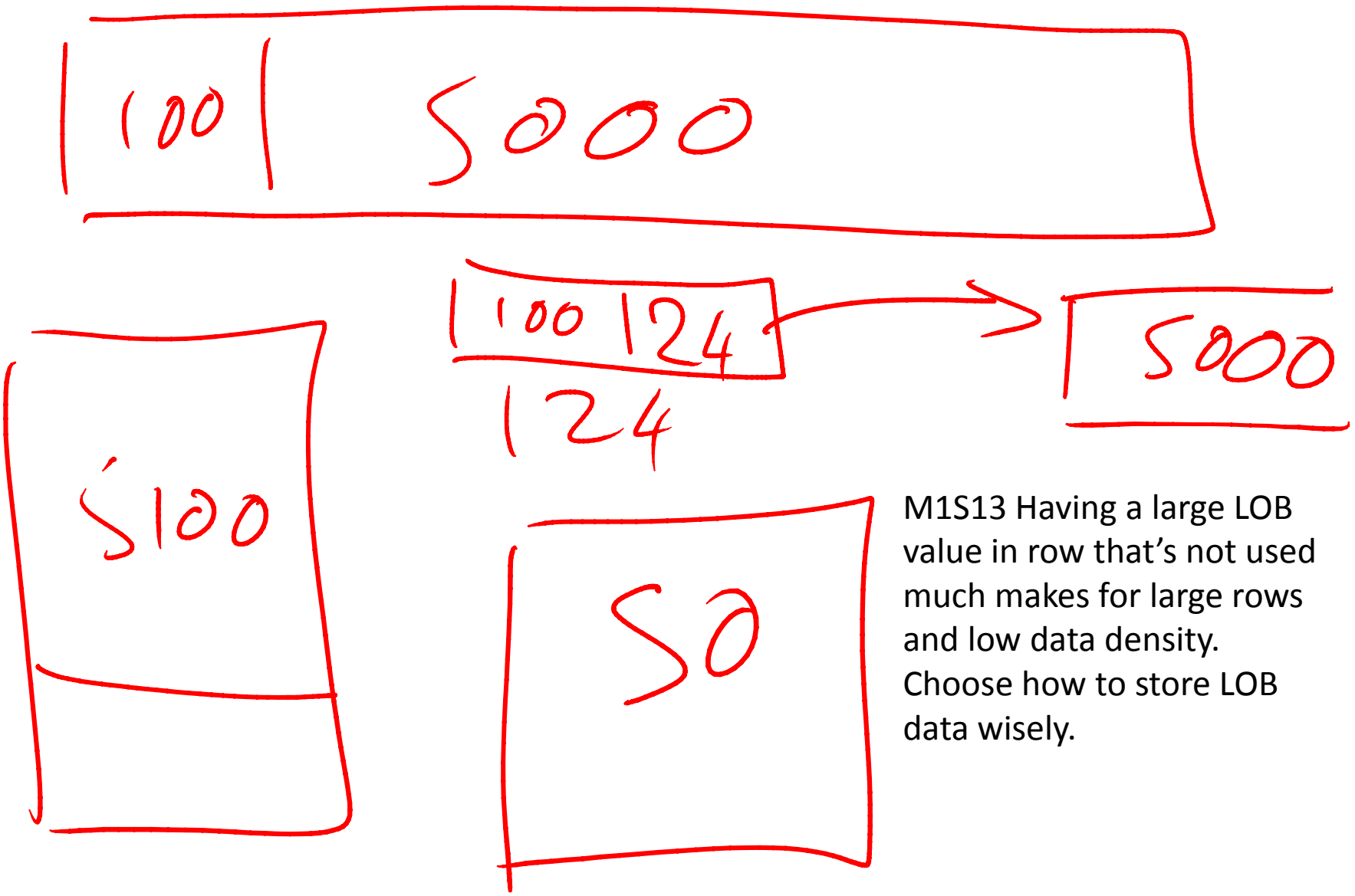
M1S09 Explaining NC index back-links to the data records as part of describing why forwarding/forwarded records exist. Without them, moving a heap row would entail changing all NC index records with a back-link to that row.



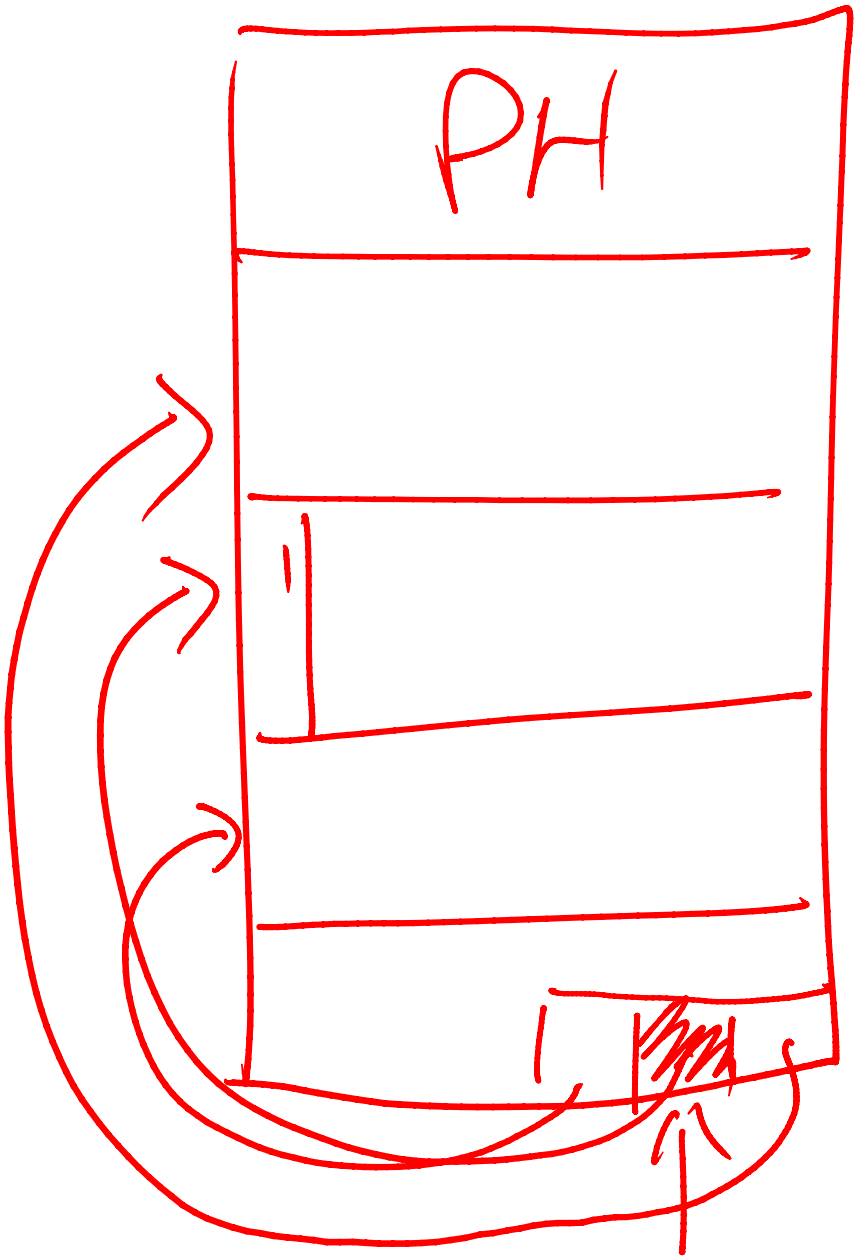
M1S10 Explaining the different record types at the leaf and non-leaf levels of clustered (on left) and nonclustered (on right) indexes. See Kimberly's index internals module for more details.



M1S11 Explaining how LOB values larger than 8K are stored in a loose tree structure, with it's root in the data record (or index record, if the LOB value is an INCLUDEd column in an NC index)

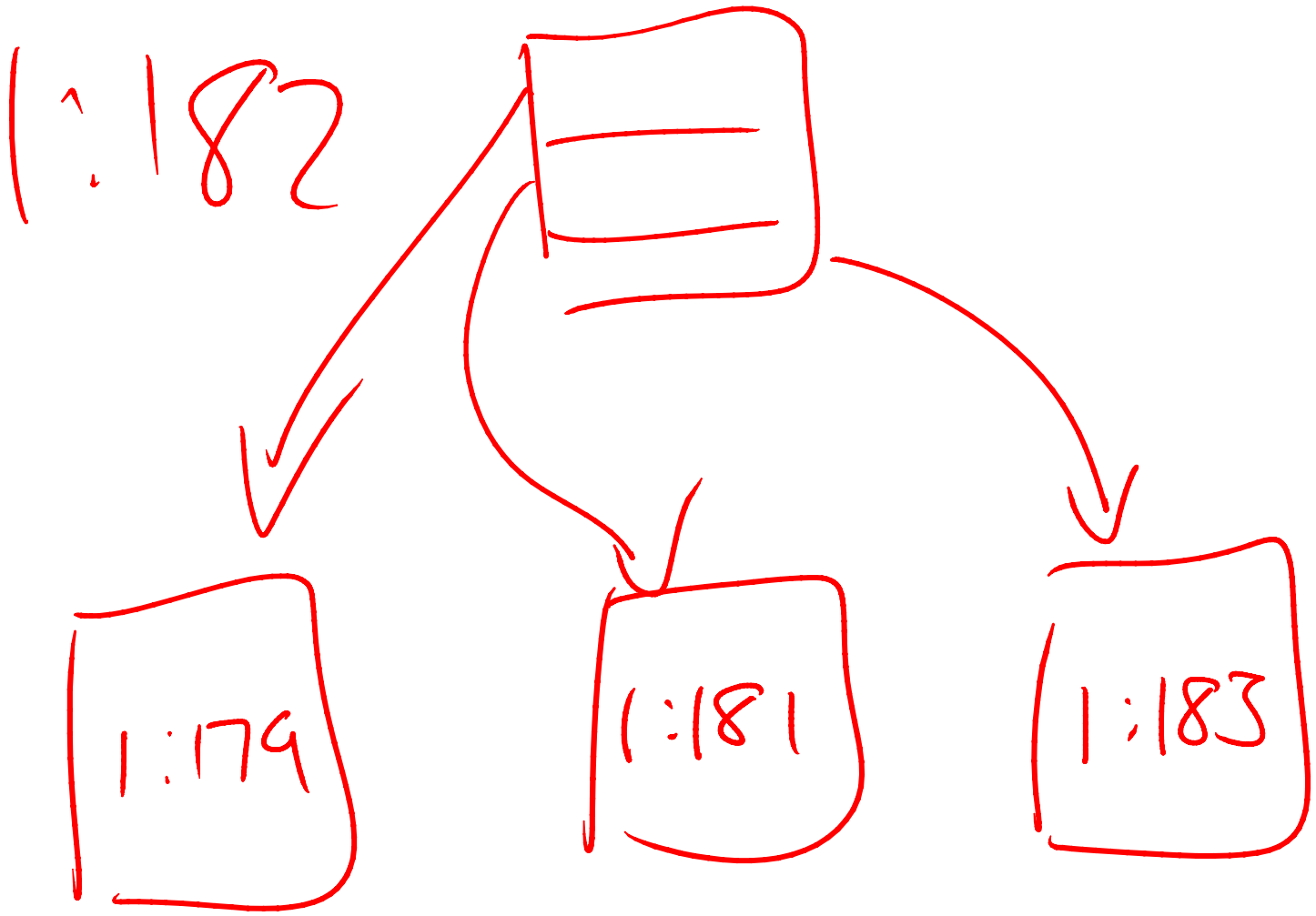


M1S13 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.

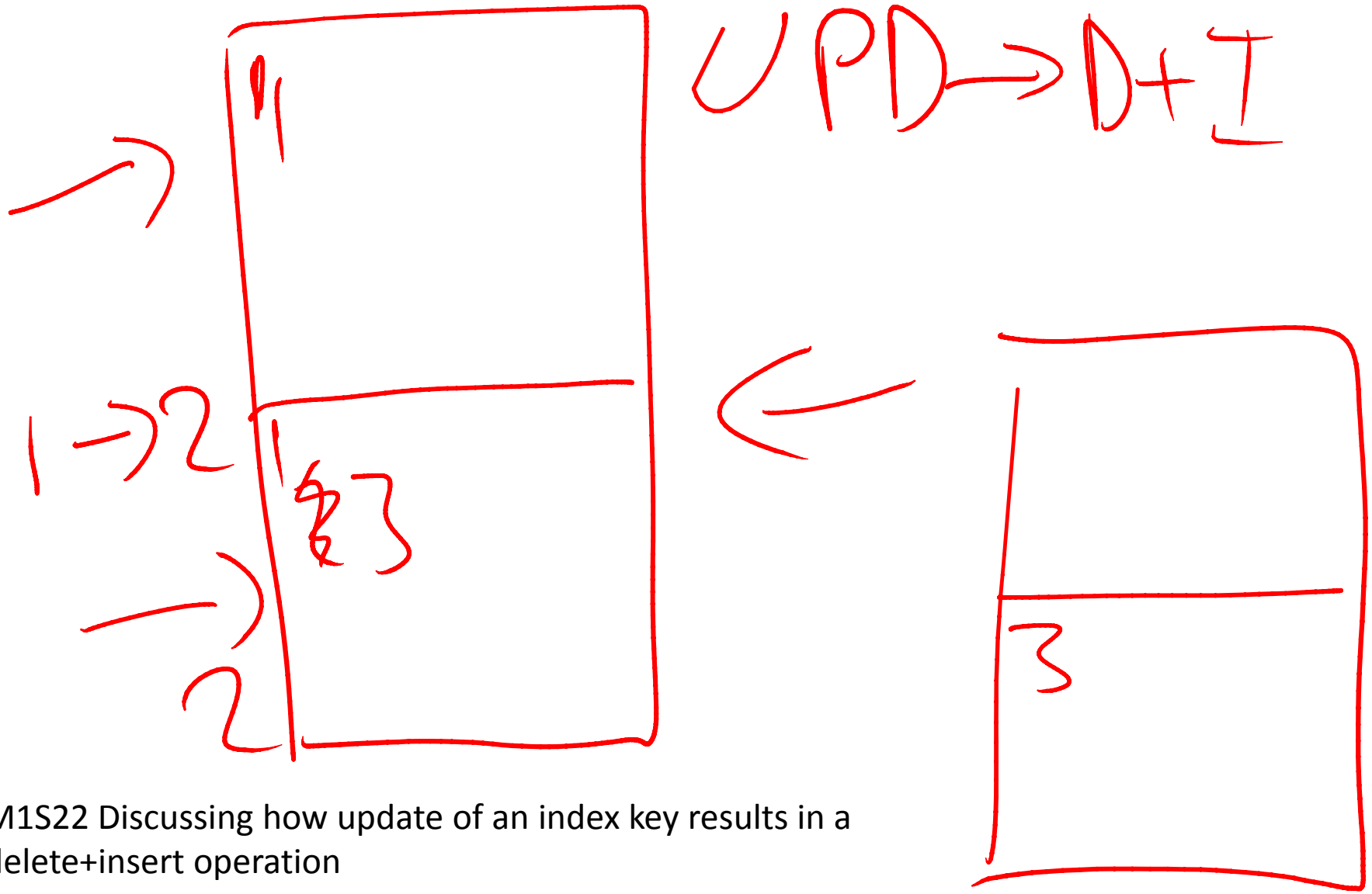


M1S15 When a record is deleted, it's marked as a ghost record. The ghost cleanup task does not physically delete the records – it removes the slot array entry that points to the record on the page. I.e. the record is unmarked as being a record and the area on the page is now free space – that just happens to have bits and bytes that used to be a valid record.

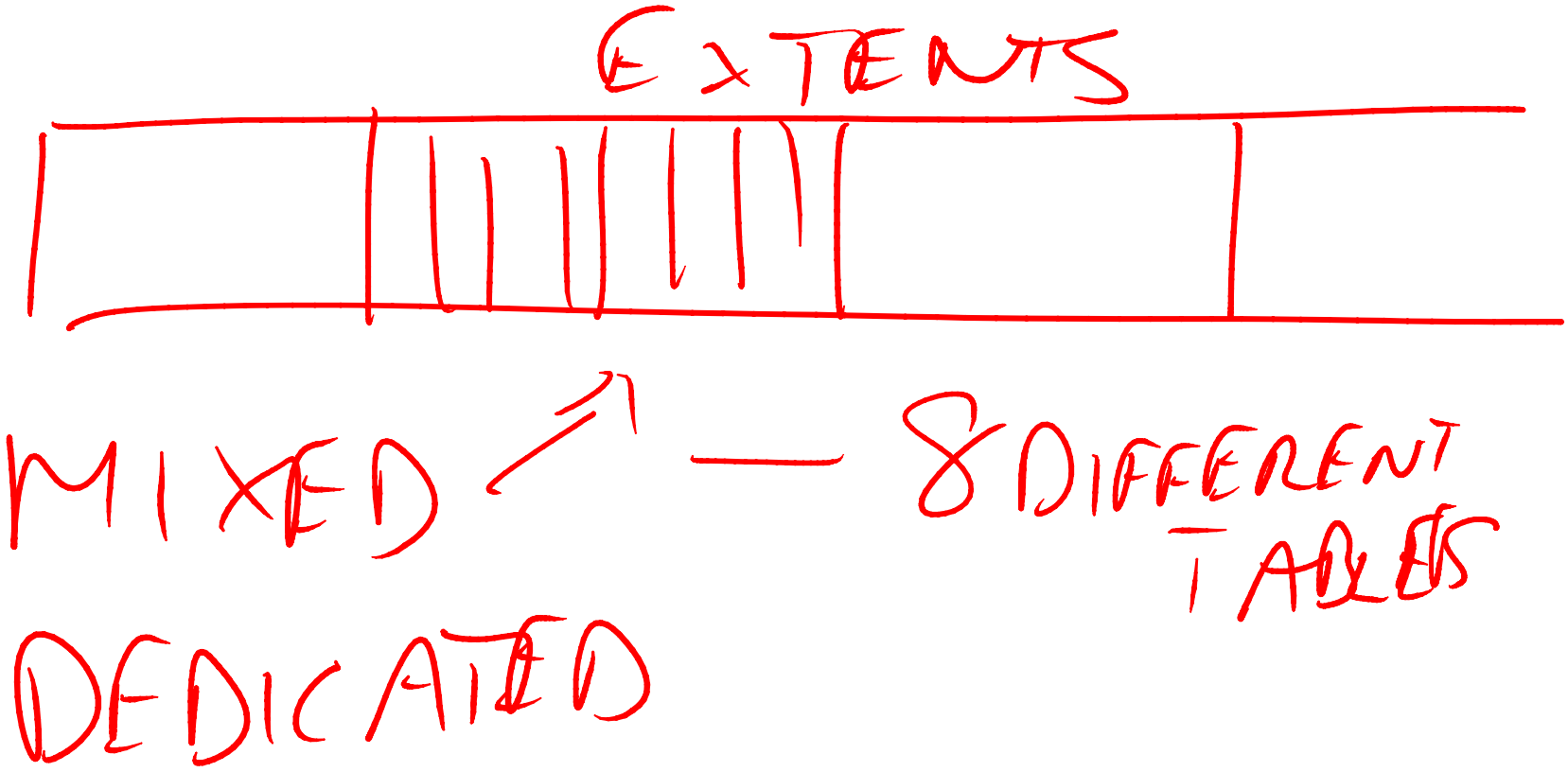
Hmm – don't remember what I was explaining here. Maybe is was about how an append-only insert pattern won't cause searches for free space in heaps to insert the new record…



M1S22 Showing the clustered index structure generated by the page/record demo.



M1S22 Discussing how update of an index key results in a delete+insert operation



M1S24 Explaining the difference between mixed and dedicated extents. Mixed extent is one in which the 8 pages could be allocated to 8 different objects. Dedicated extent is one where all 8 pages are reserved for exclusive use of a single object.

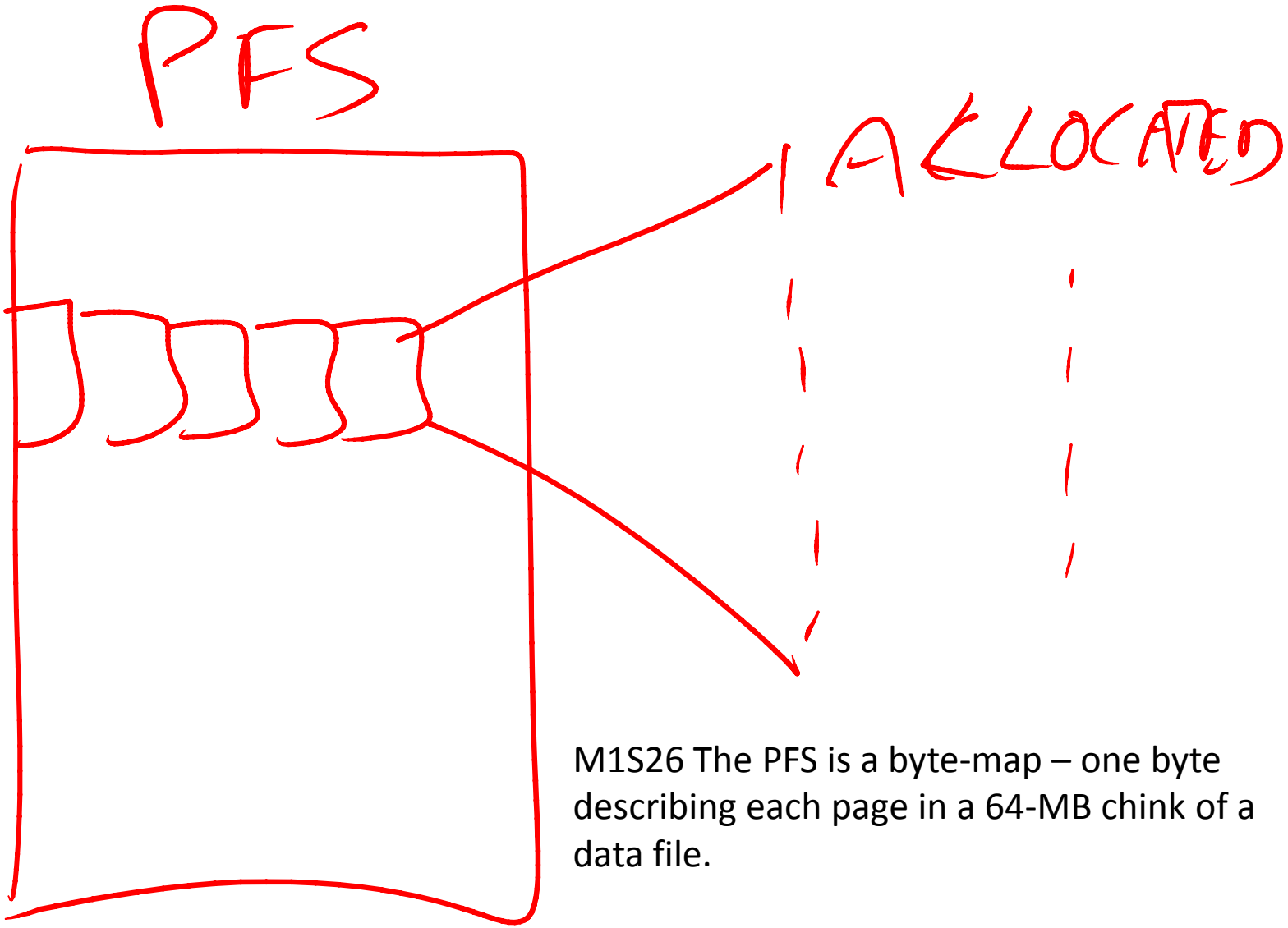
Later in the deck I refine ‘object’ to really be ‘allocation unit’.

IGNACIO PORTILLO

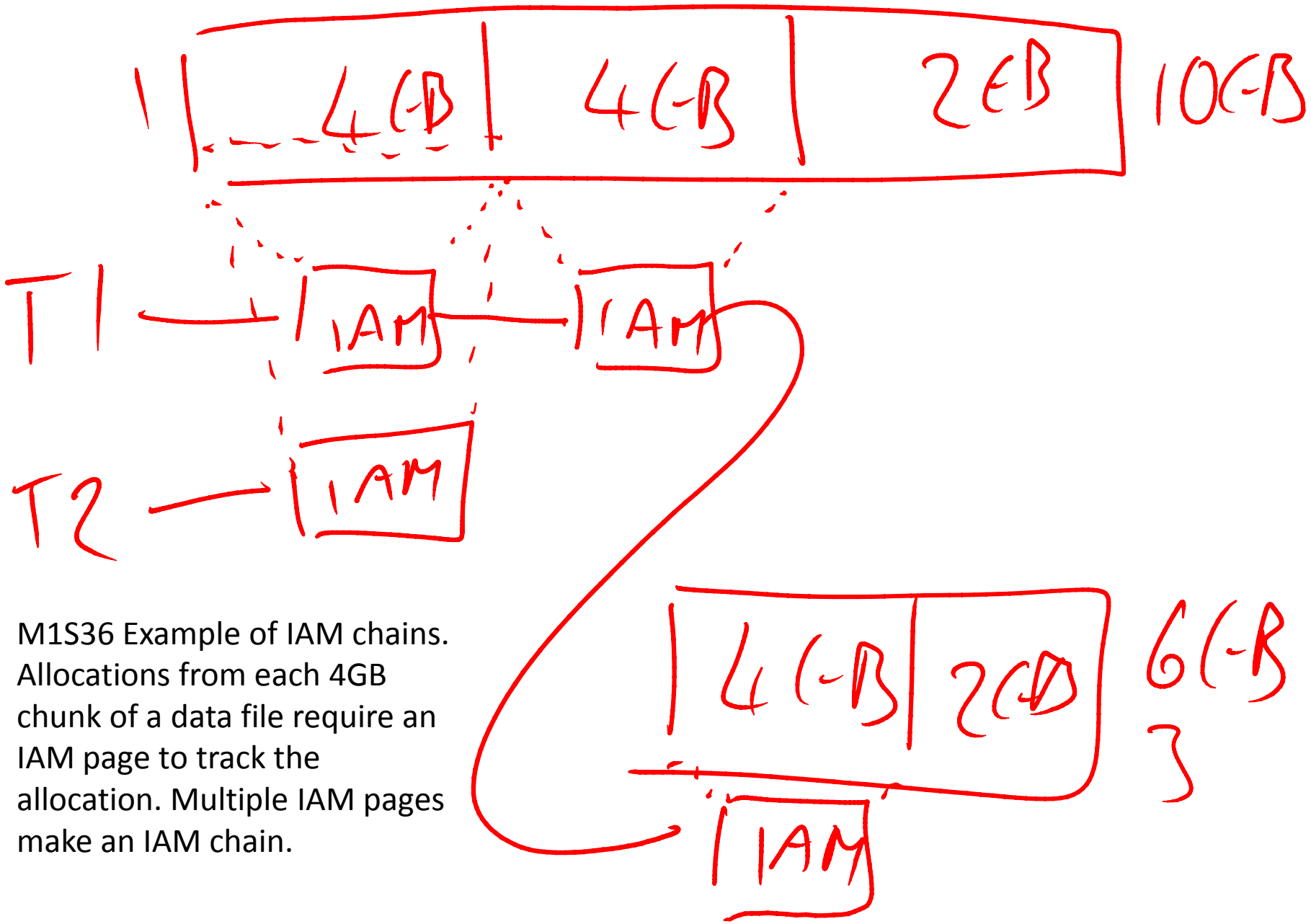
READAHEAD

Blog post on buffer pool ramp-up after a server start and how these don't show as readahead I/Os in output from SET STATISTICS IO ON.

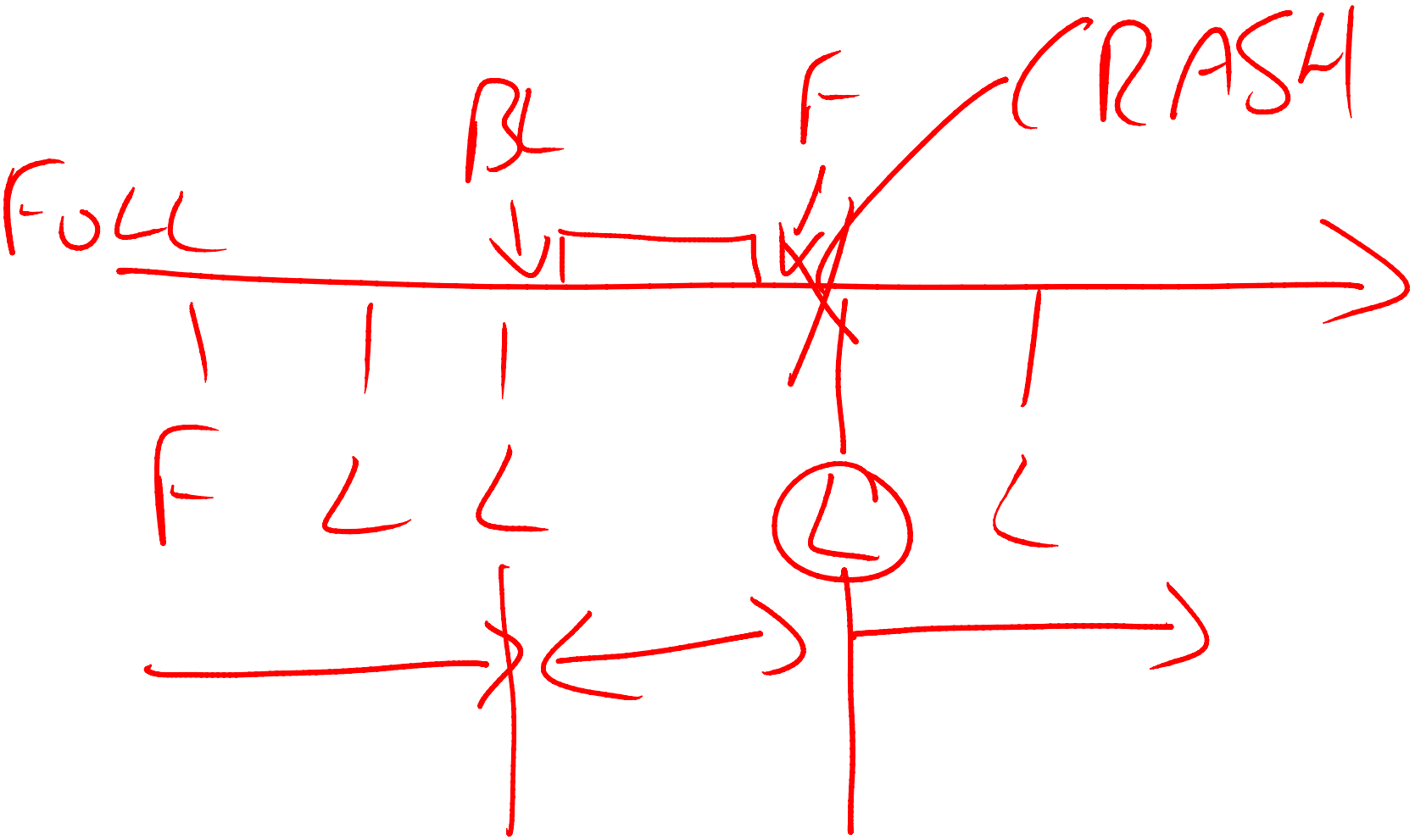
http://blogs.msdn.com/b/ialonso/archive/2011/12/09/the-read-ahead-that-doesn-t-count-as-read-ahead.aspx



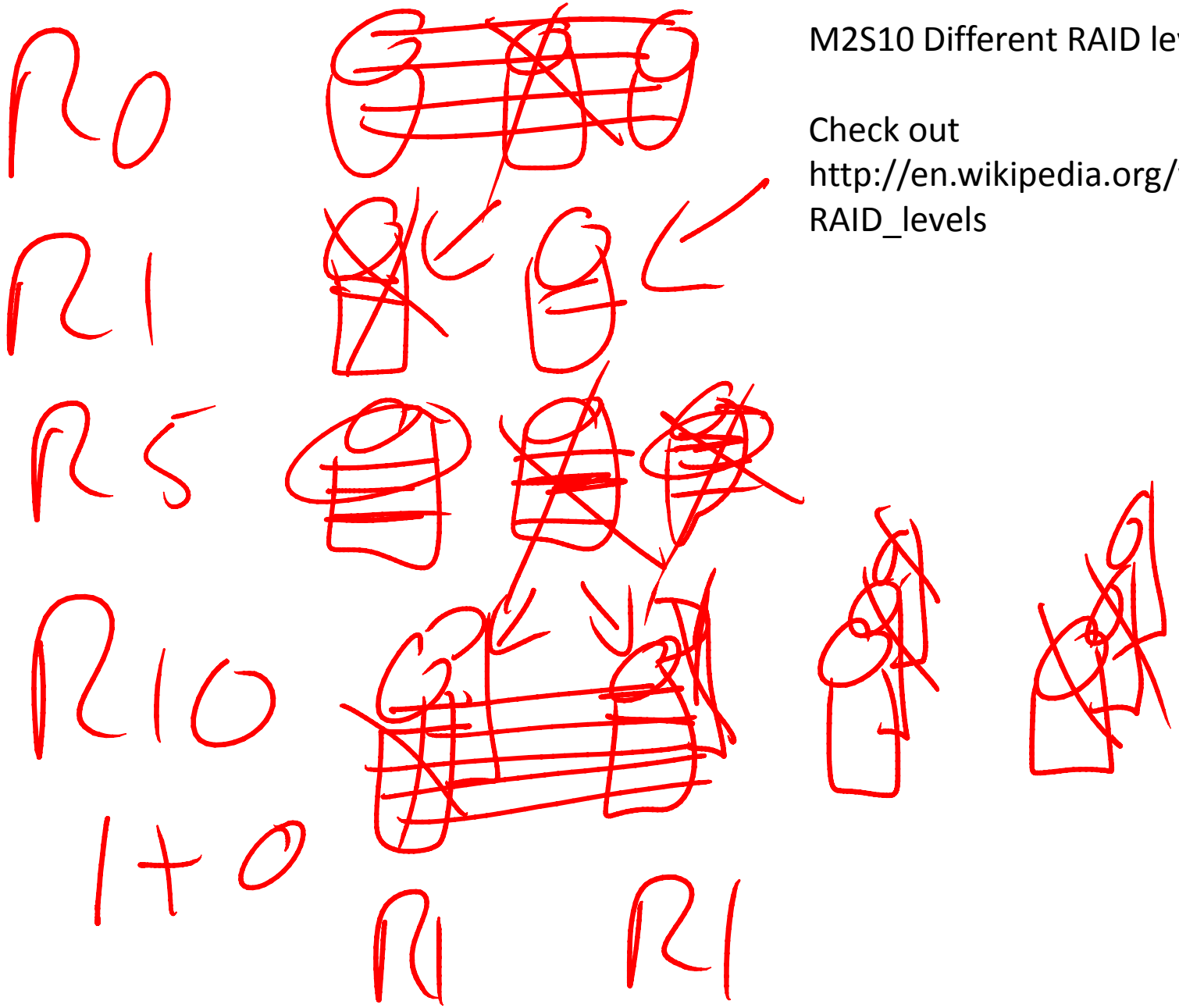
M1S26 The PFS is a byte-map – one byte describing each page in a 64-MB chink of a data file.



M1S36 Example of IAM chains. Allocations from each 4GB chunk of a data file require an IAM page to track the allocation. Multiple IAM pages make an IAM chain.

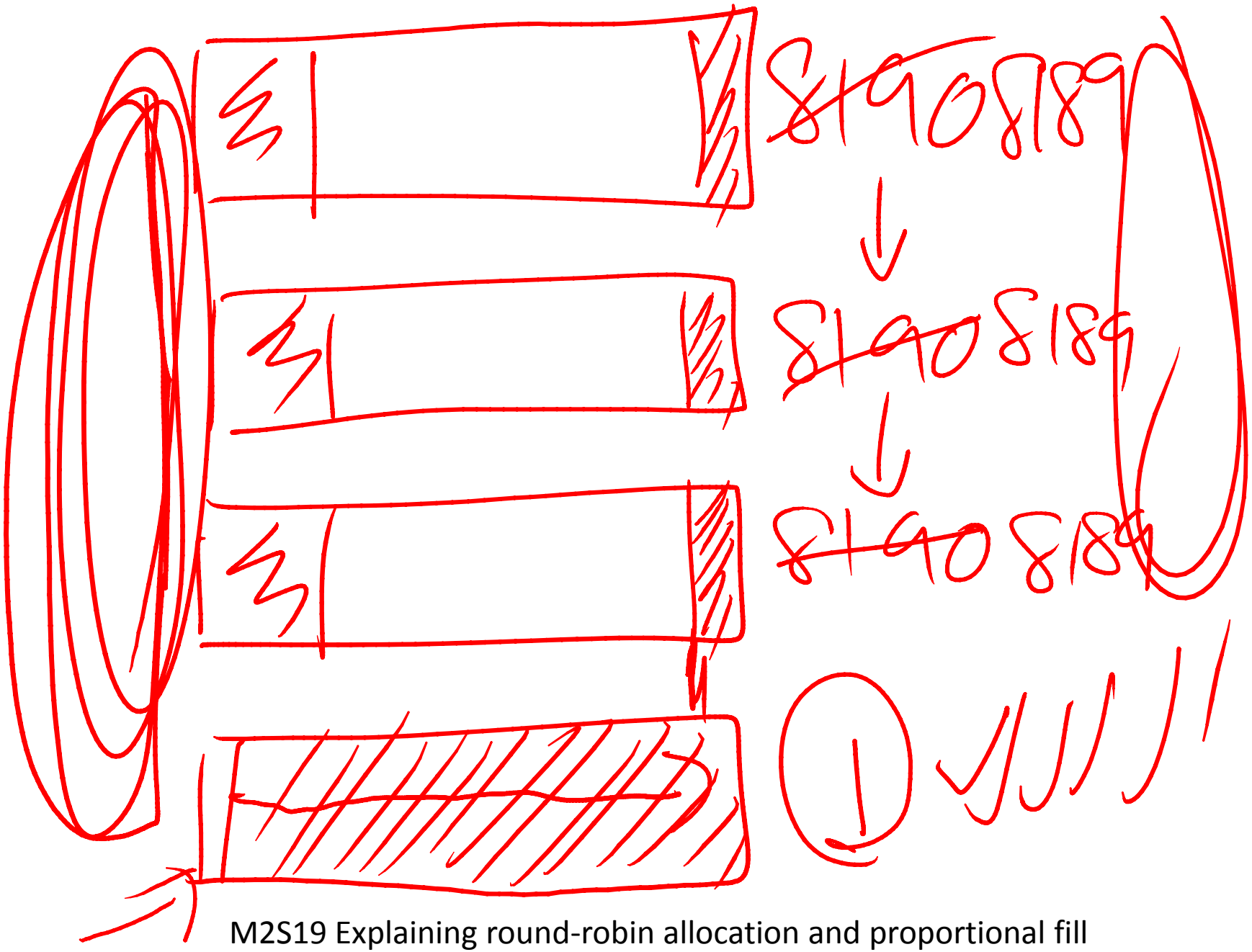


M1 (and M3S37) Explaining that if a minimally-logged operation is present in a log backup, you cannot do a point-in-time restore to any time covered by that backup (except the start or end of that time period). You also cannot do a tail-of-the-log backup after a crash that destroys the data files if there has been a minimally-logged operation since the last log backup. Well, you can, but the log backup is invalid.

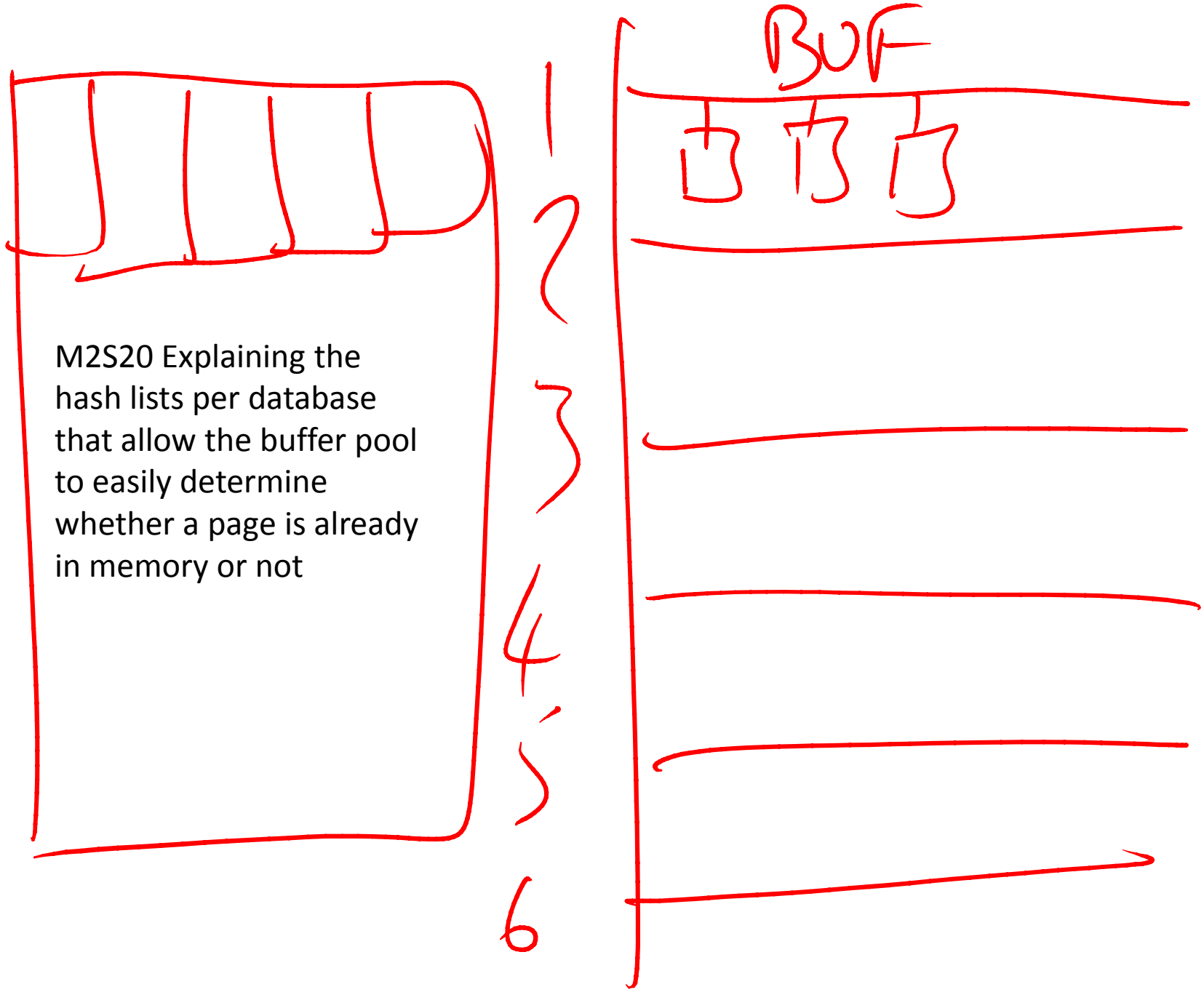


M2S10 Different RAID levels.

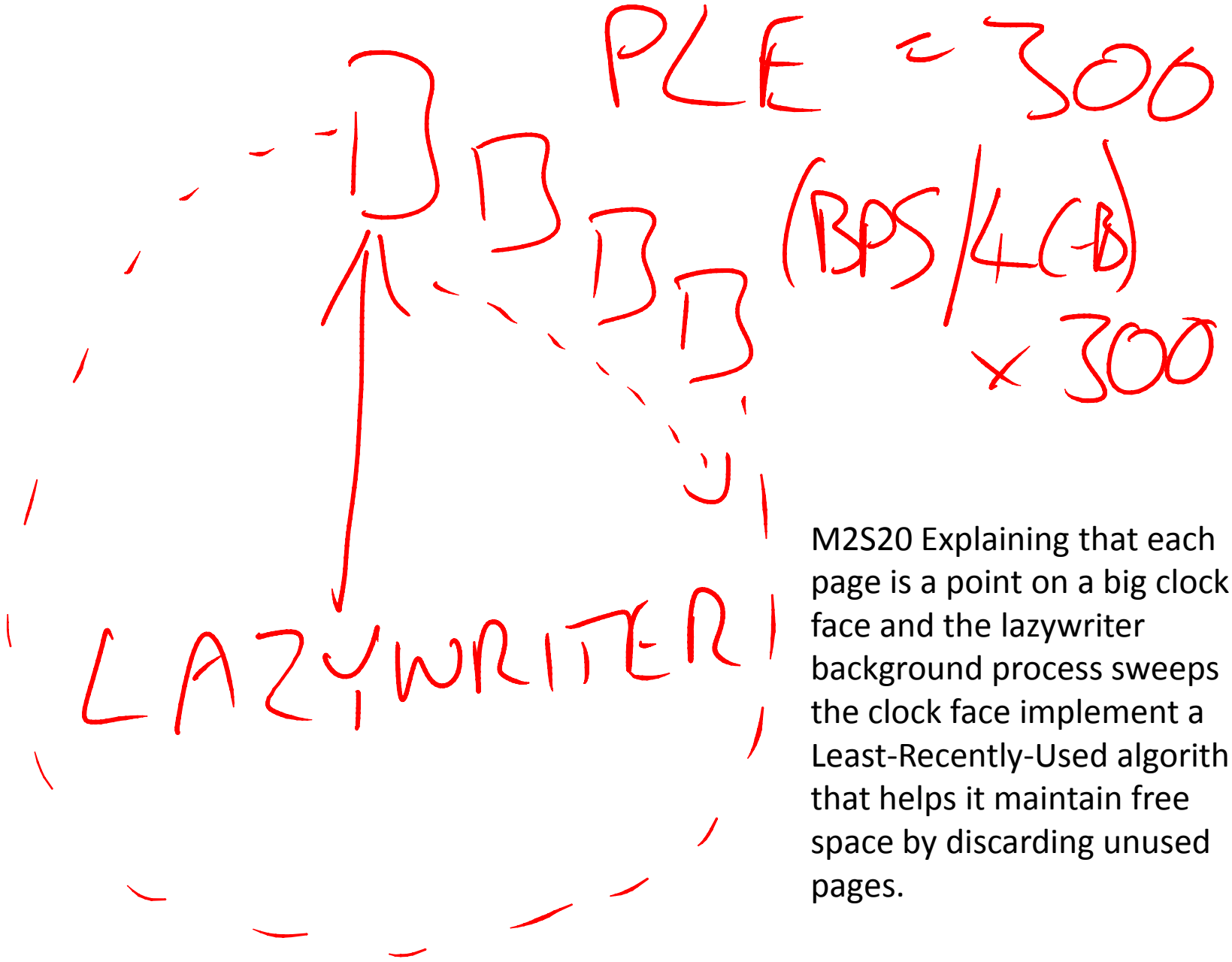
Check out
http://en.wikipedia.org/wiki/Standard_RAID_levels



M2S19 Explaining round-robin allocation and proportional fill

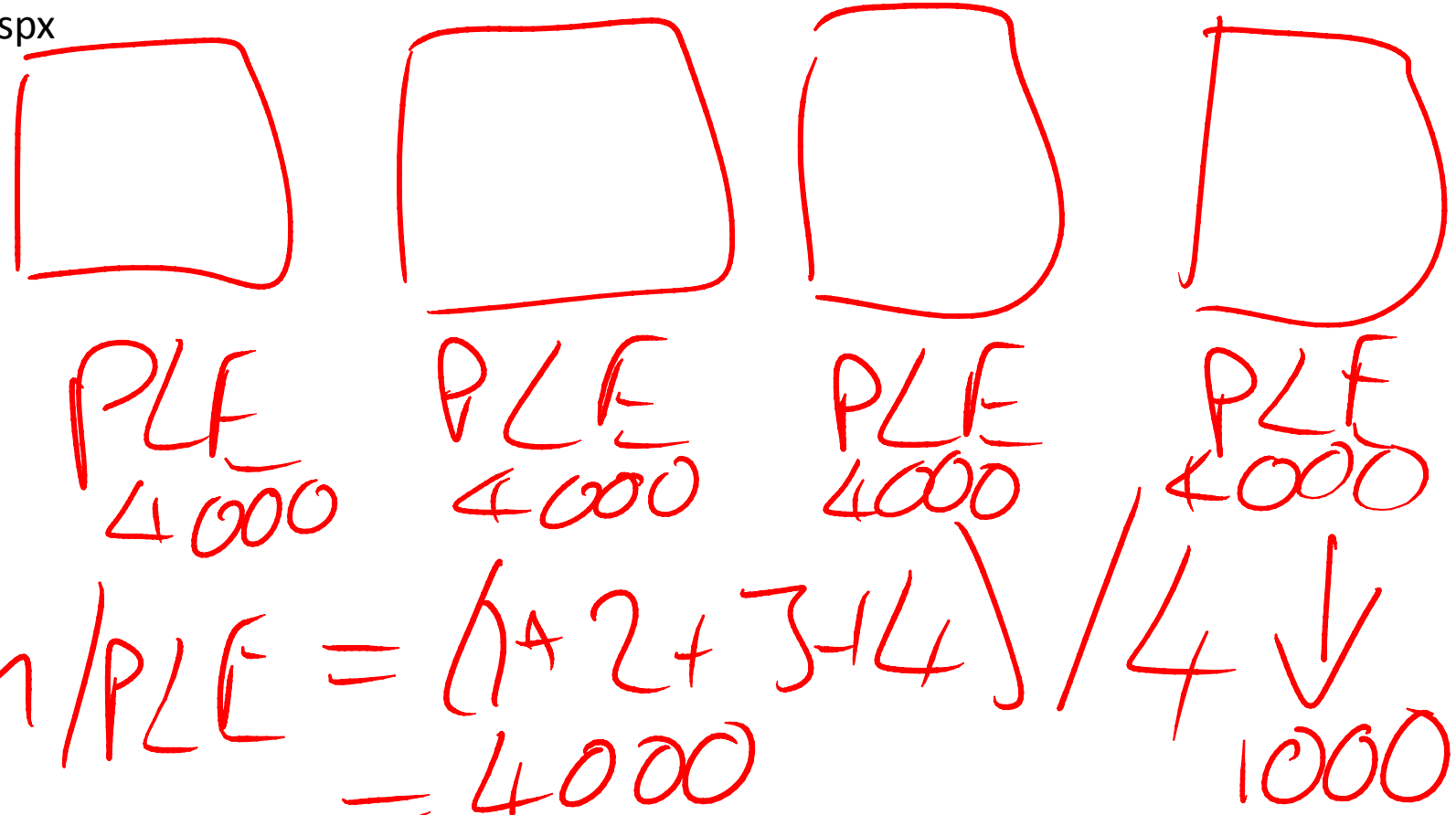


M2S20 Explaining the hash lists per database that allow the buffer pool to easily determine whether a page is already in memory or not



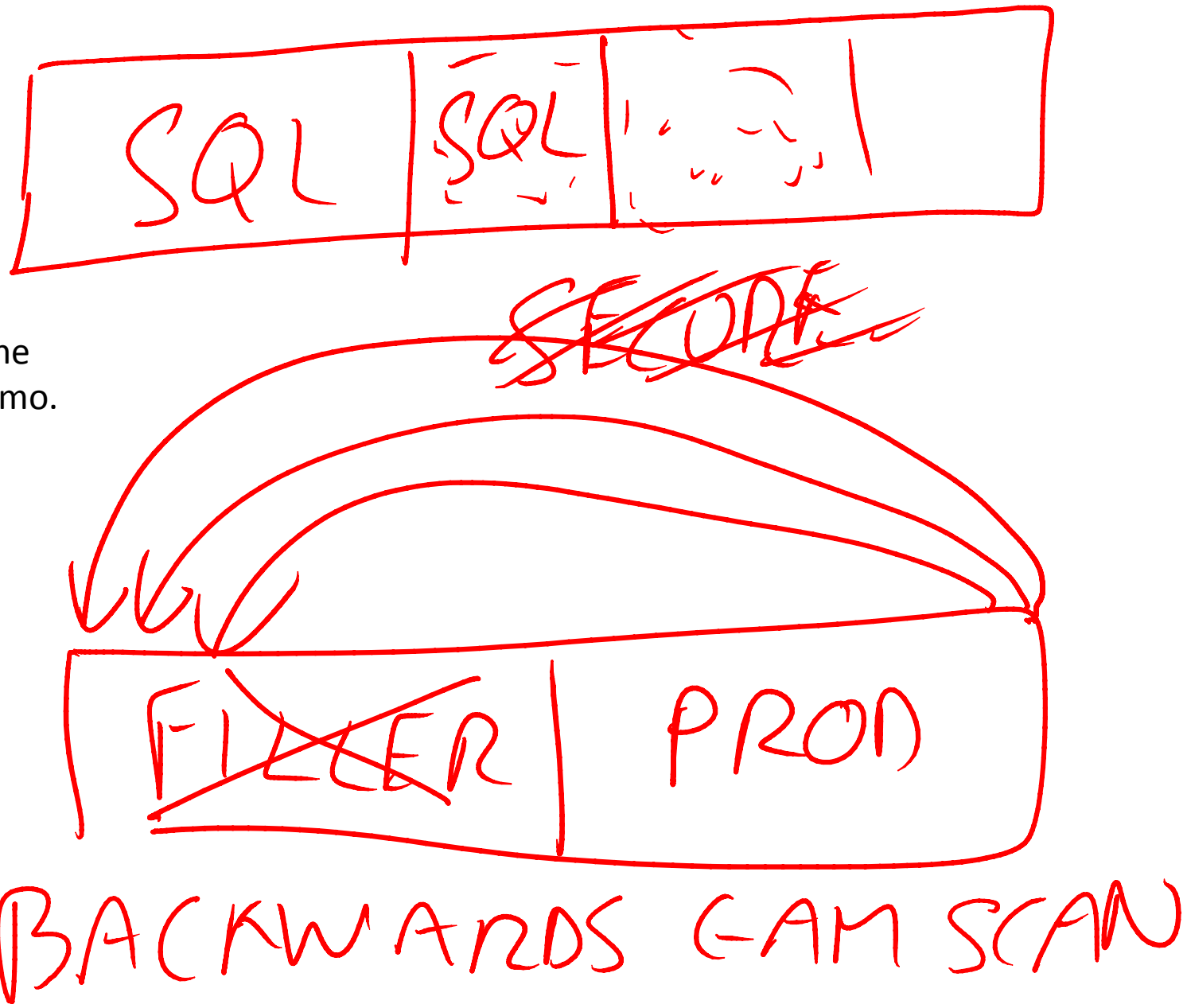
M2S20 Explaining that each page is a point on a big clock face and the lazywriter background process sweeps the clock face implement a Least-Recently-Used algorithm that helps it maintain free space by discarding unused pages.

See http://www.sqlskills.com/BLOGS/PAUL/post/Page-Life-Expectancy-isnt-what-you-think.aspx

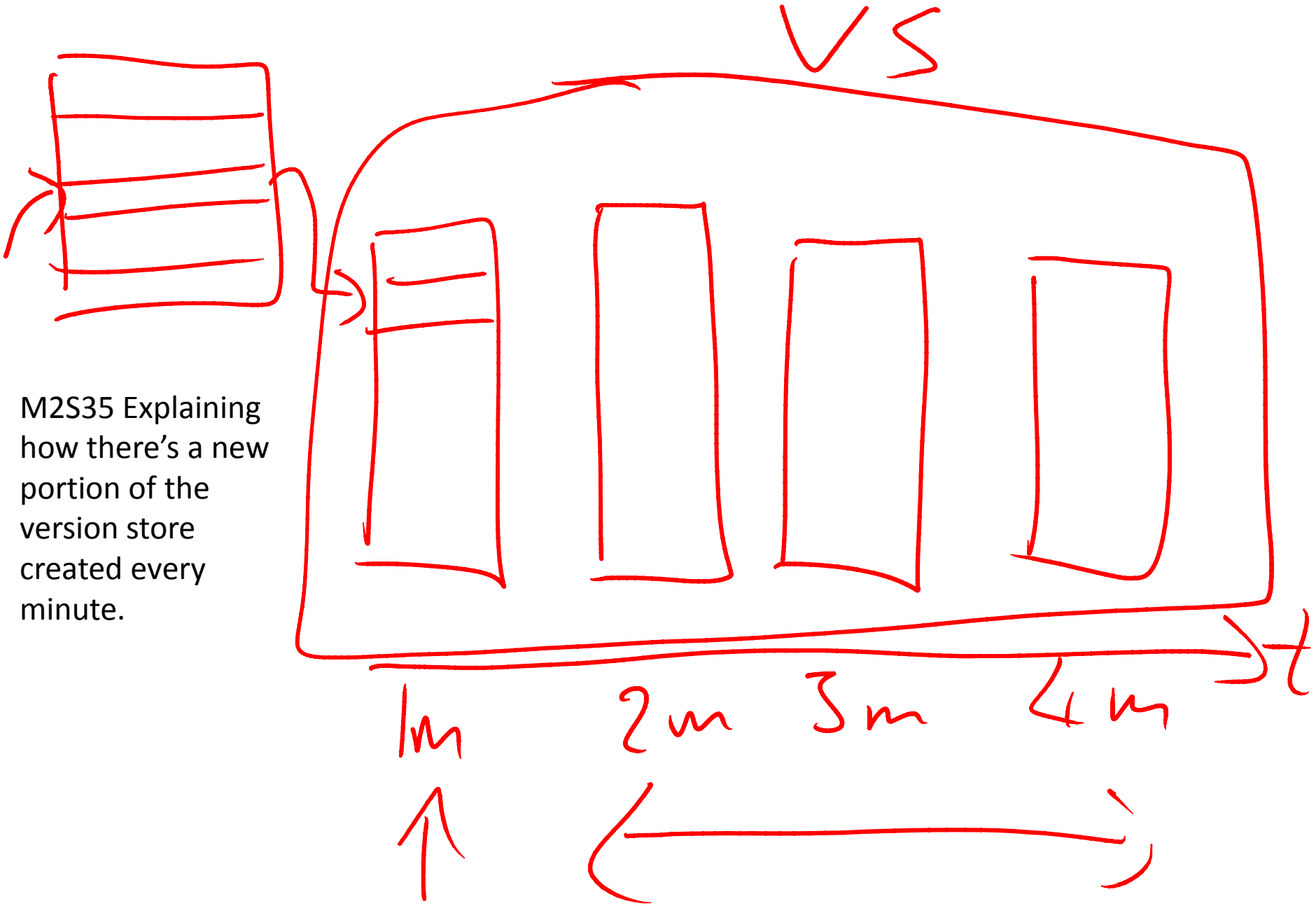


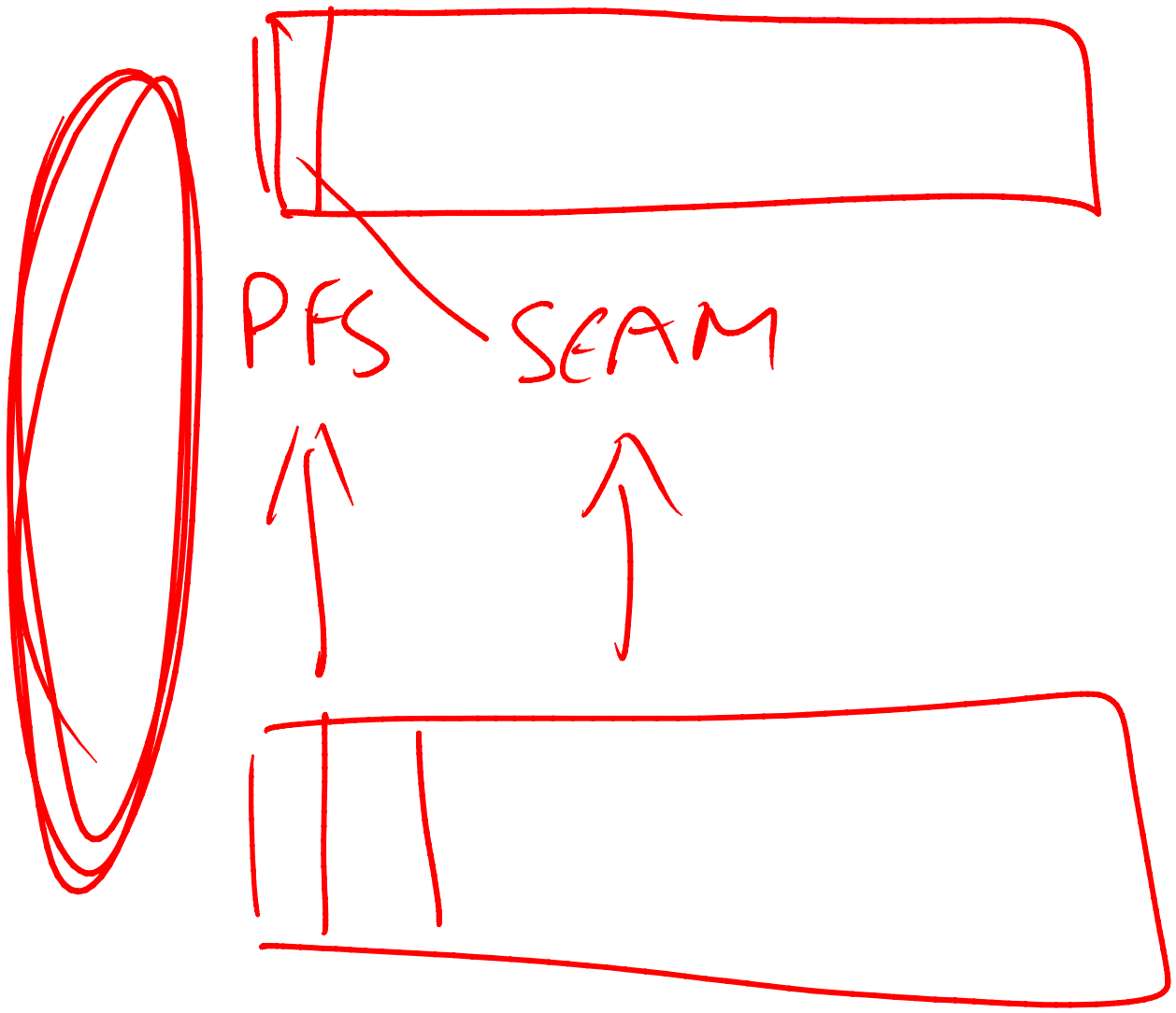
This should be Buffer Node instead of Buffer Partition – my mistake.

M2S28 The shrink demo.

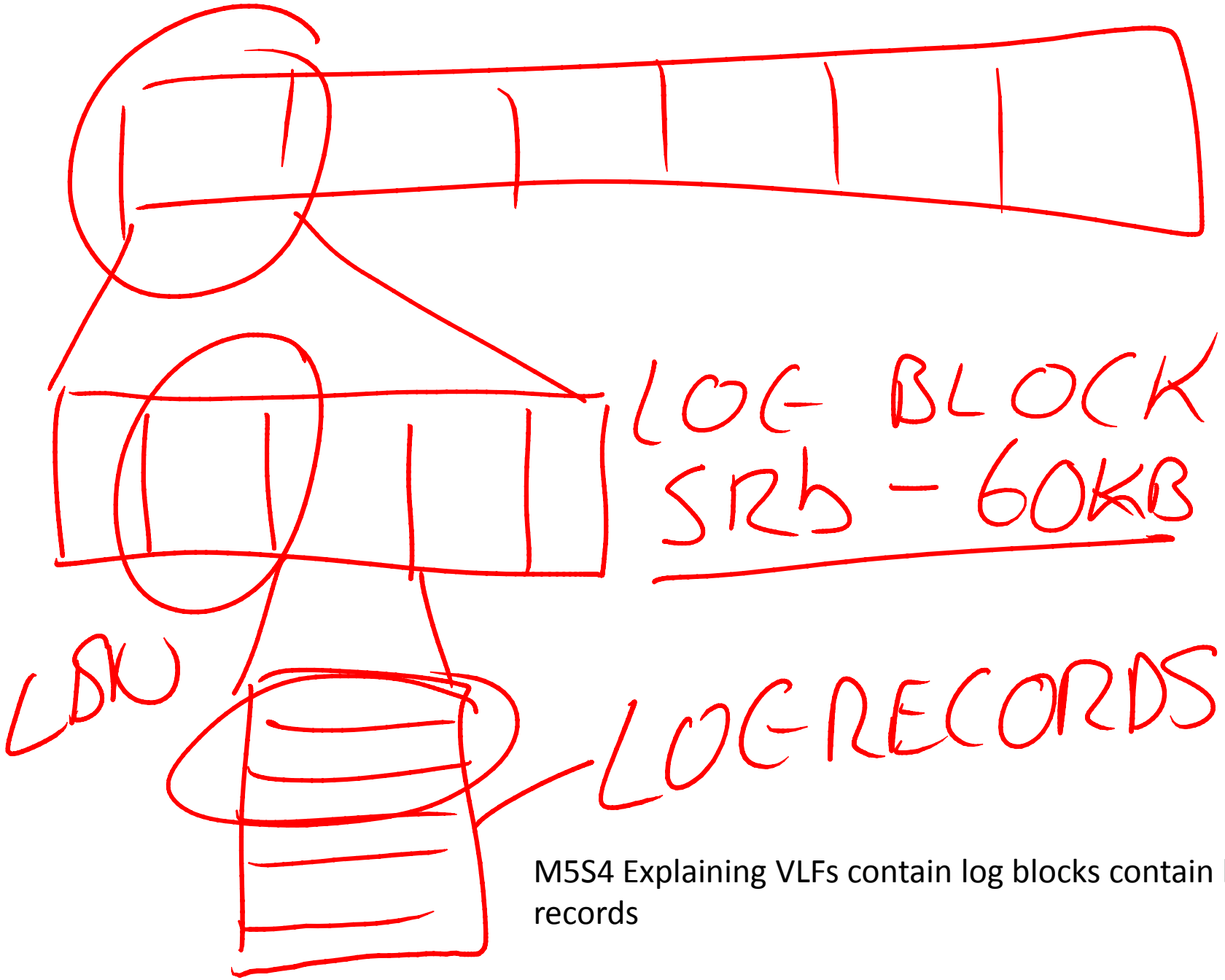


M2S35 Explaining how there's a new portion of the version store created every minute.

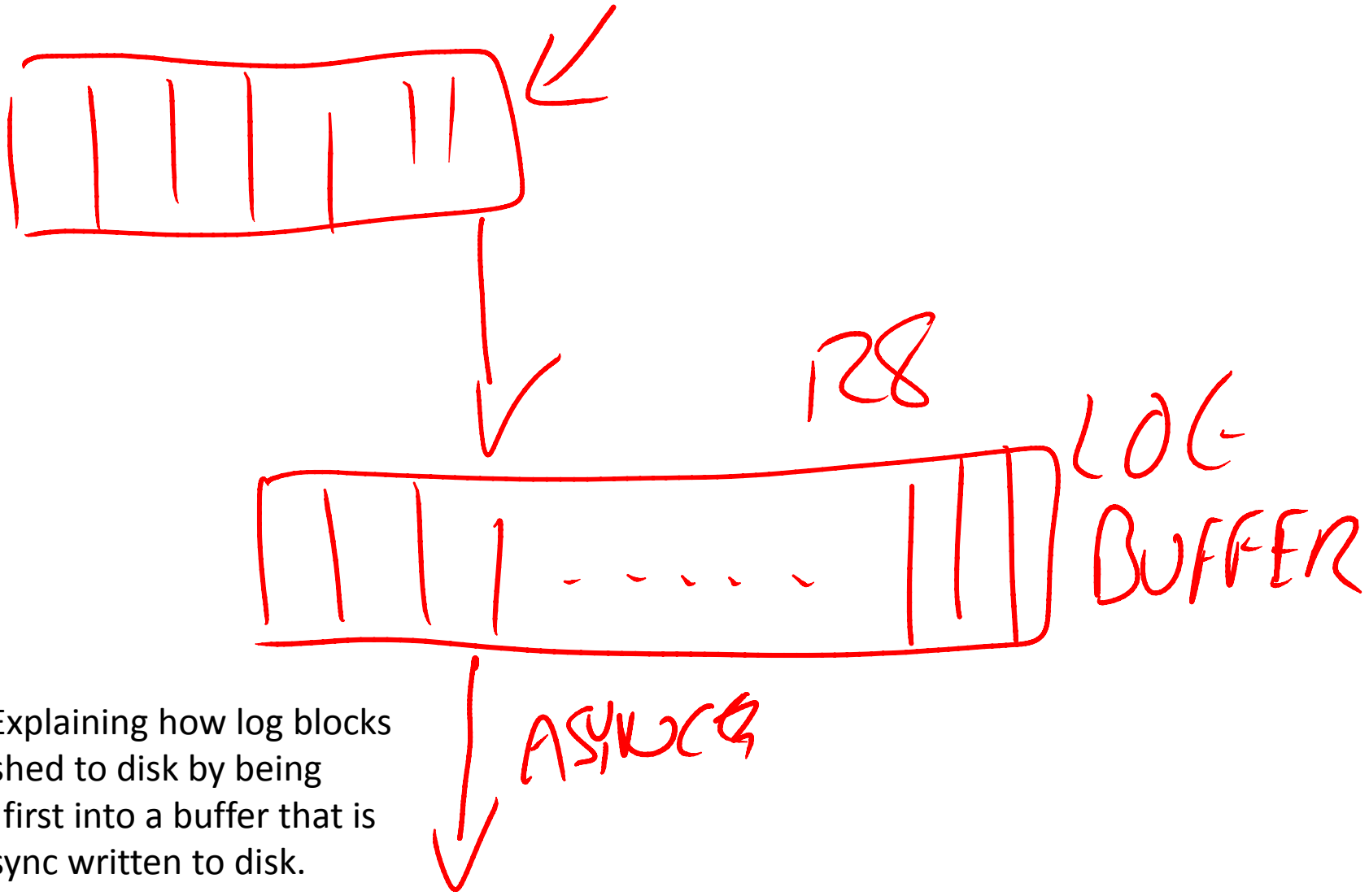




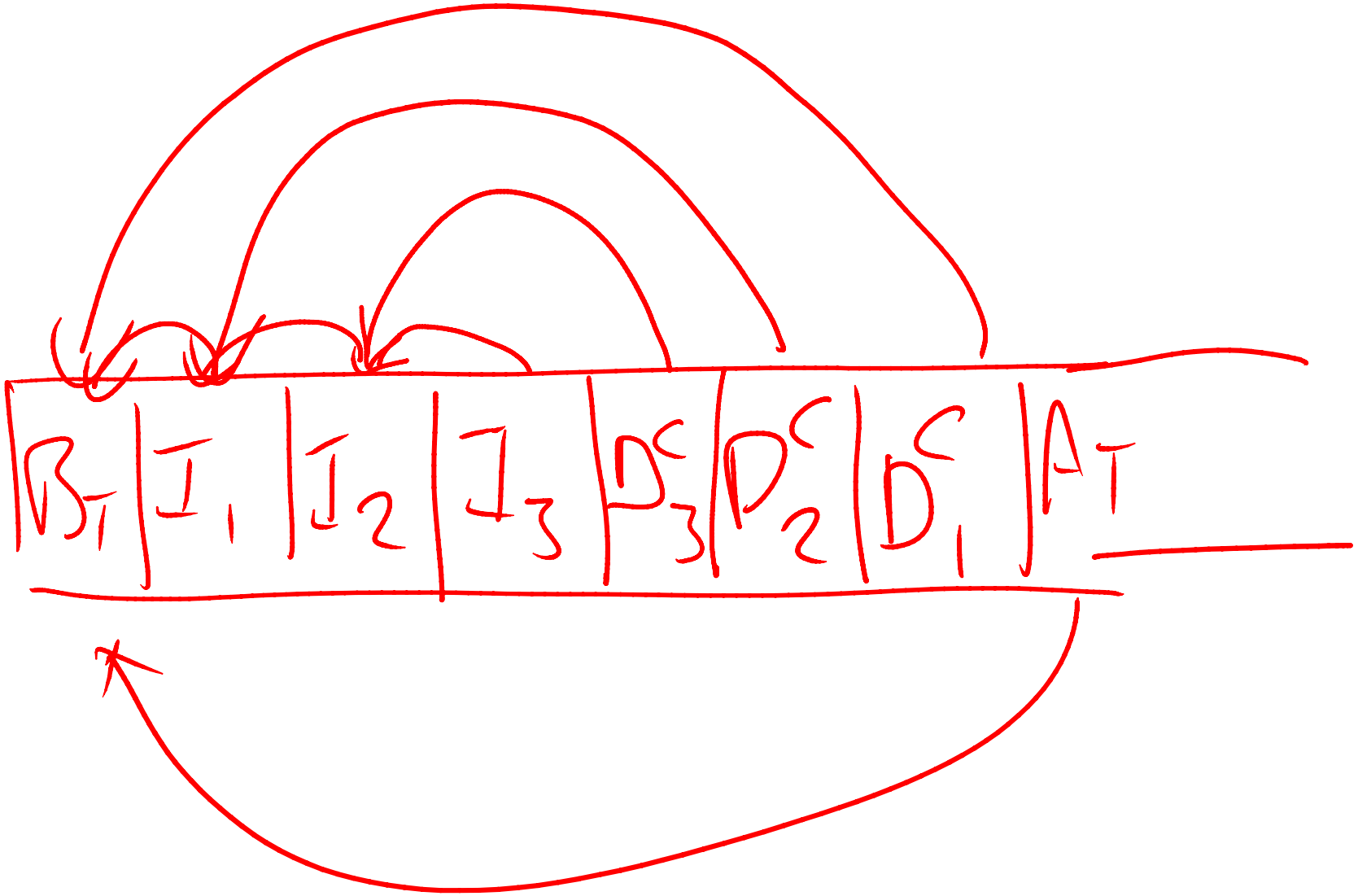
M2S40 Explaining PFS and SGAM contention in tempdb and how adding multiple files spreads the contention over multiple files, thus reducing it.



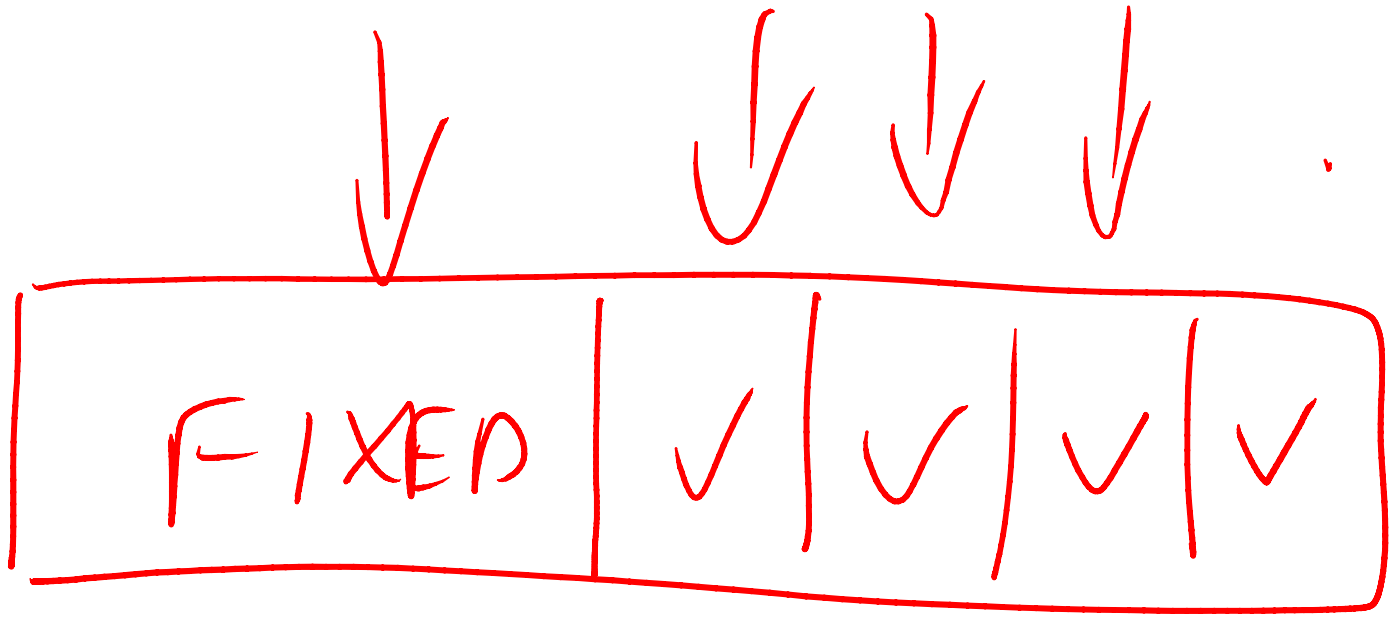
M5S4 Explaining VLFs contain log blocks contain log records



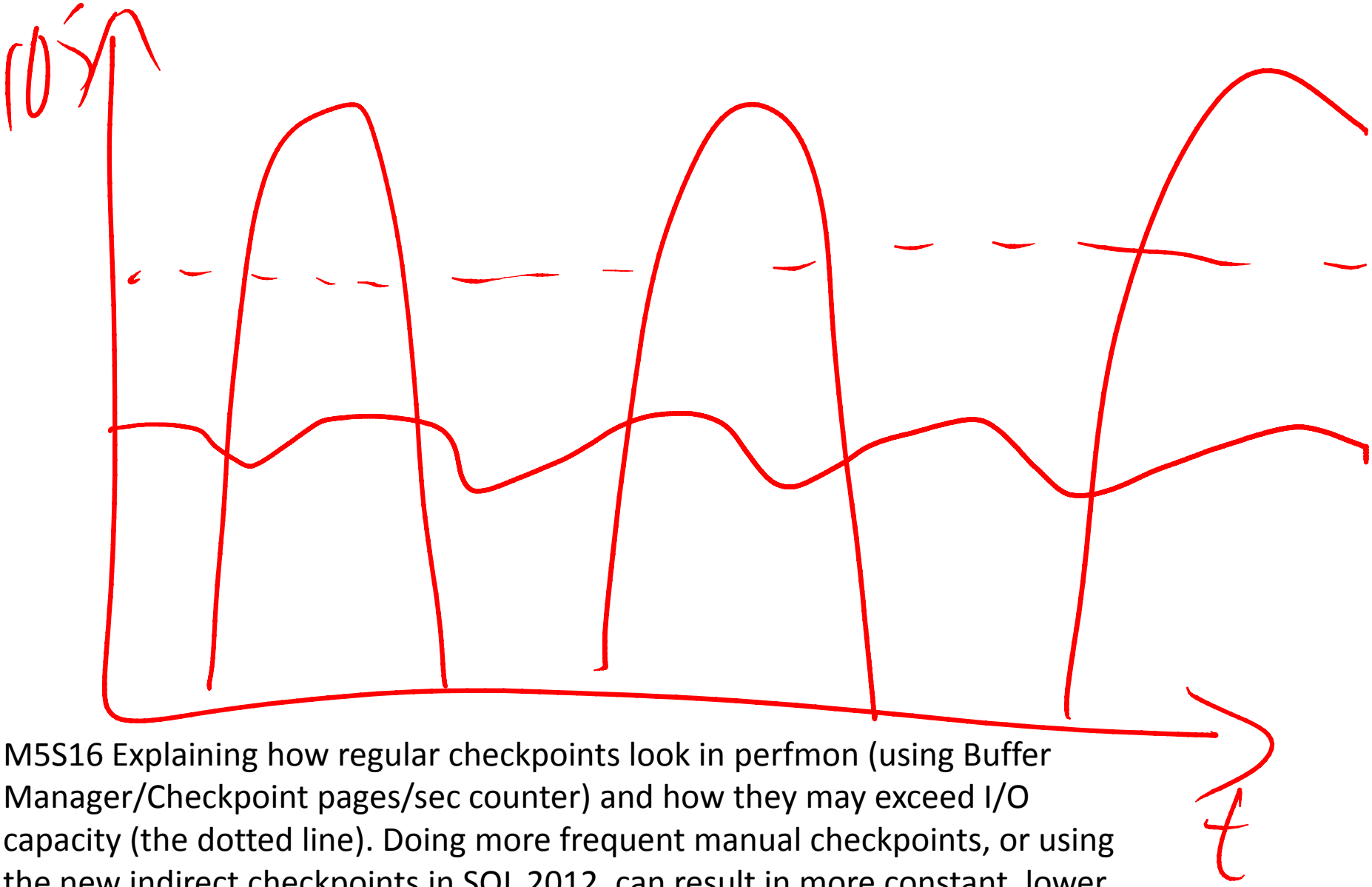
M5S4 Explaining how log blocks are flushed to disk by being copied first into a buffer that is then async written to disk.



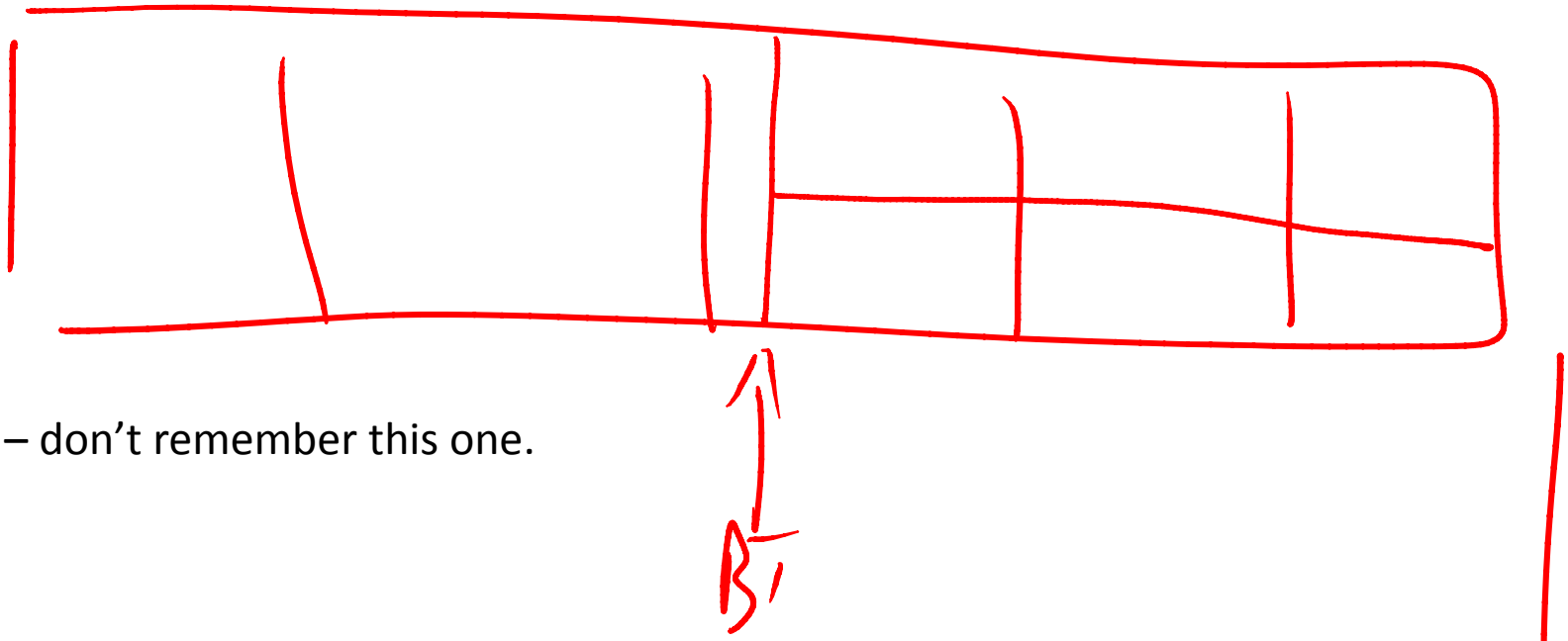
M5S11 Explaining how log records link back to the previous log record in the transaction to allow the transaction to be rolled back by generating compensating actions in reverse order. A compensation log record will link over the log record it is compensating for.



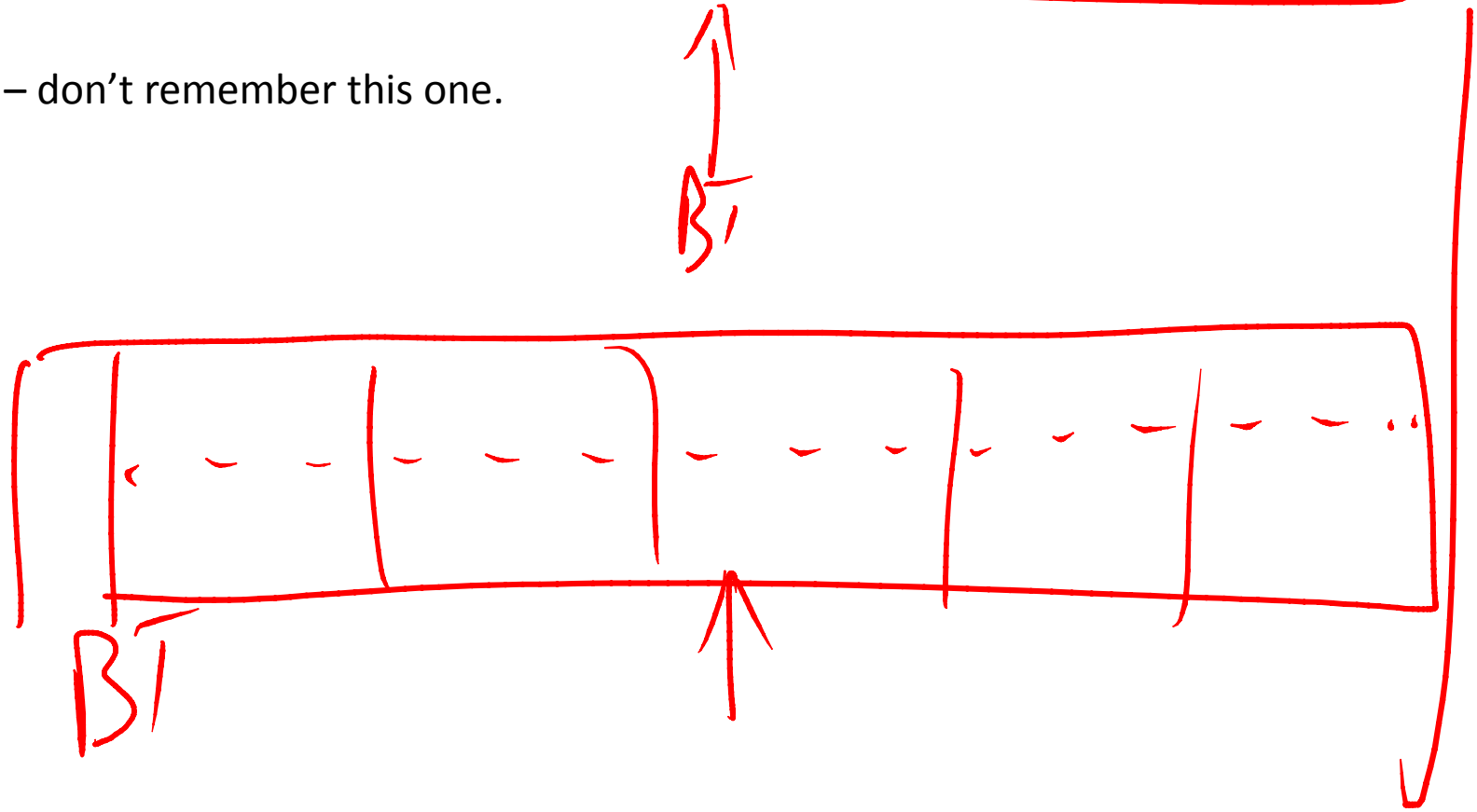
M5S11 Explaining how the Access Methods determines how to log UPDATEs efficiently by splitting the logging by portion of the record. Each variable length column is a portion, as is the fixed-width portion (up to 16 bytes in each log record)

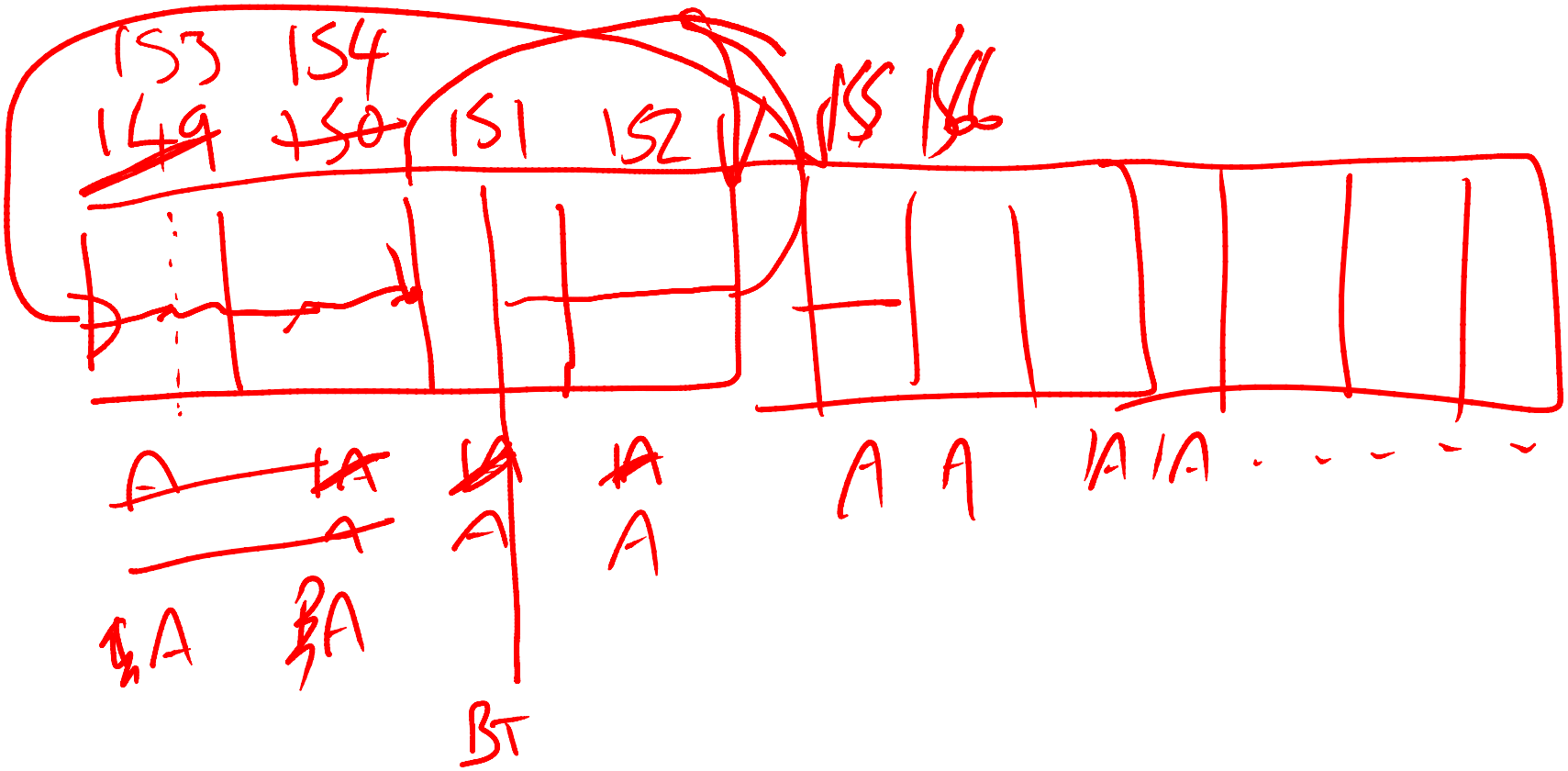


M5S16 Explaining how regular checkpoints look in perfmon (using Buffer Manager/Checkpoint pages/sec counter) and how they may exceed I/O capacity (the dotted line). Doing more frequent manual checkpoints, or using the new indirect checkpoints in SQL 2012, can result in more constant, lower IO load.

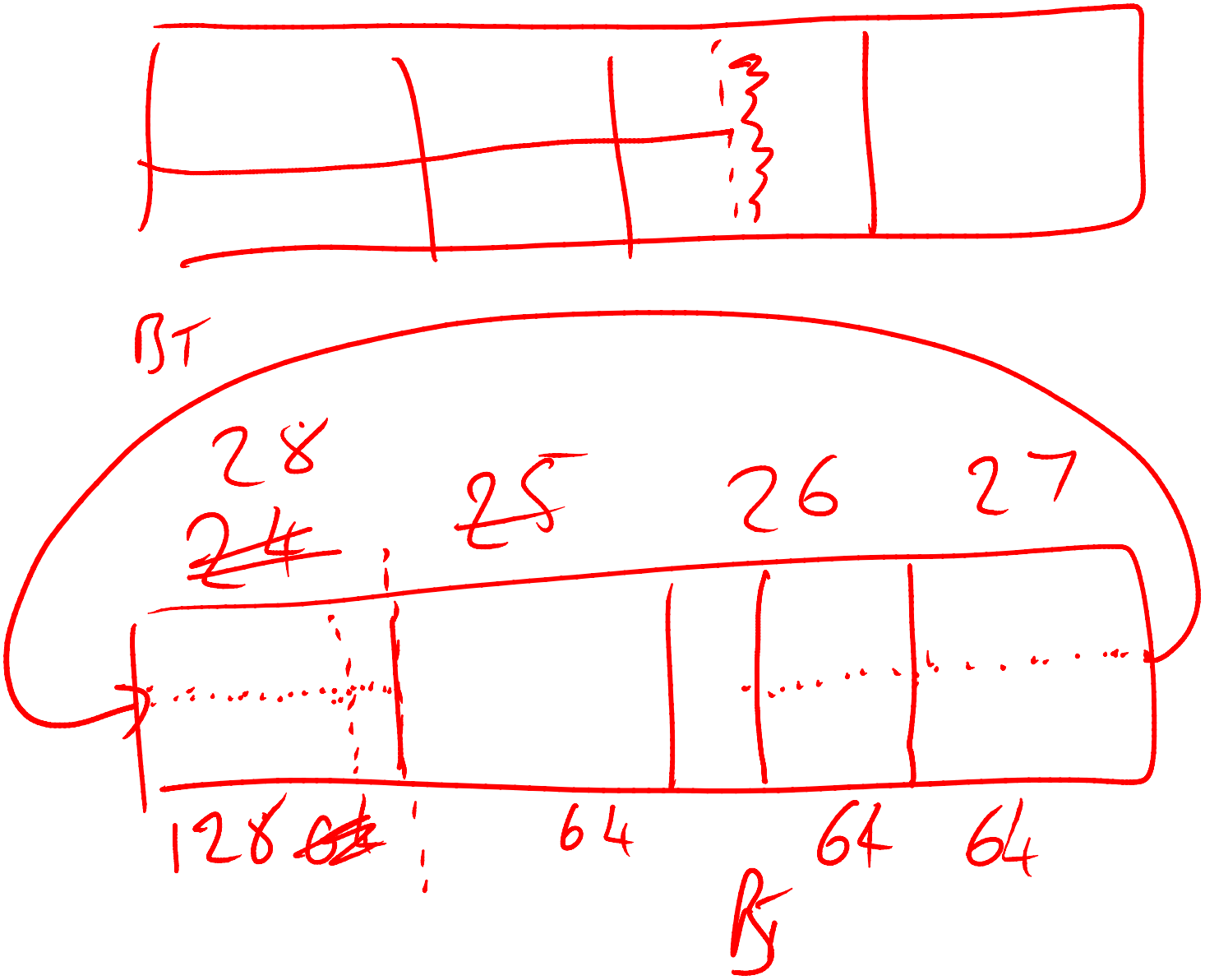


Hmm – don't remember this one.

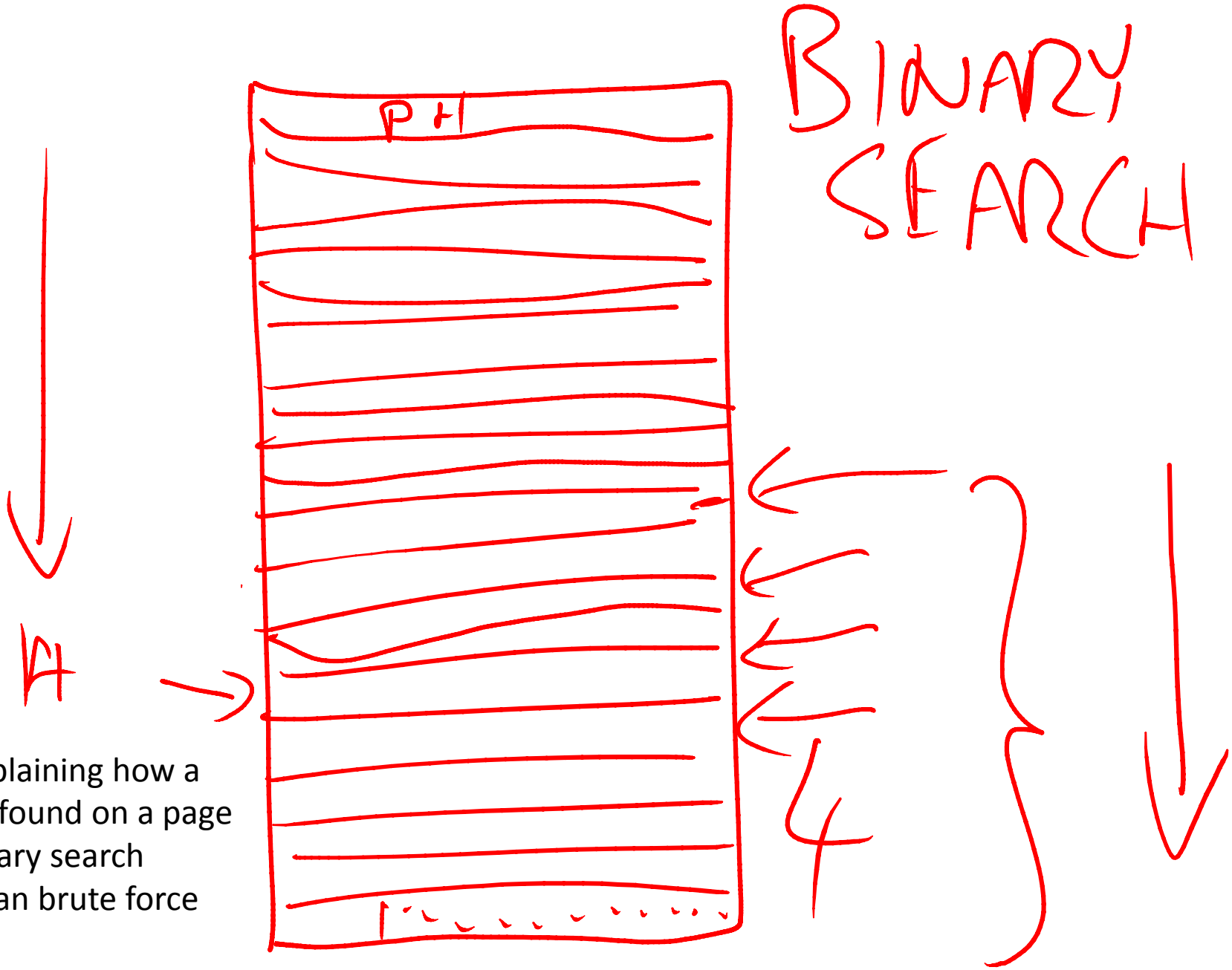




M5S28 Explaining the circular log demo. We discussed log space reservation that happens to ensure the active transactions can roll back without having to grow the log – this accounts for extra log growths. Space reserved in the log does not make a VLF become active – as there are no log records written out, just empty space reserved.

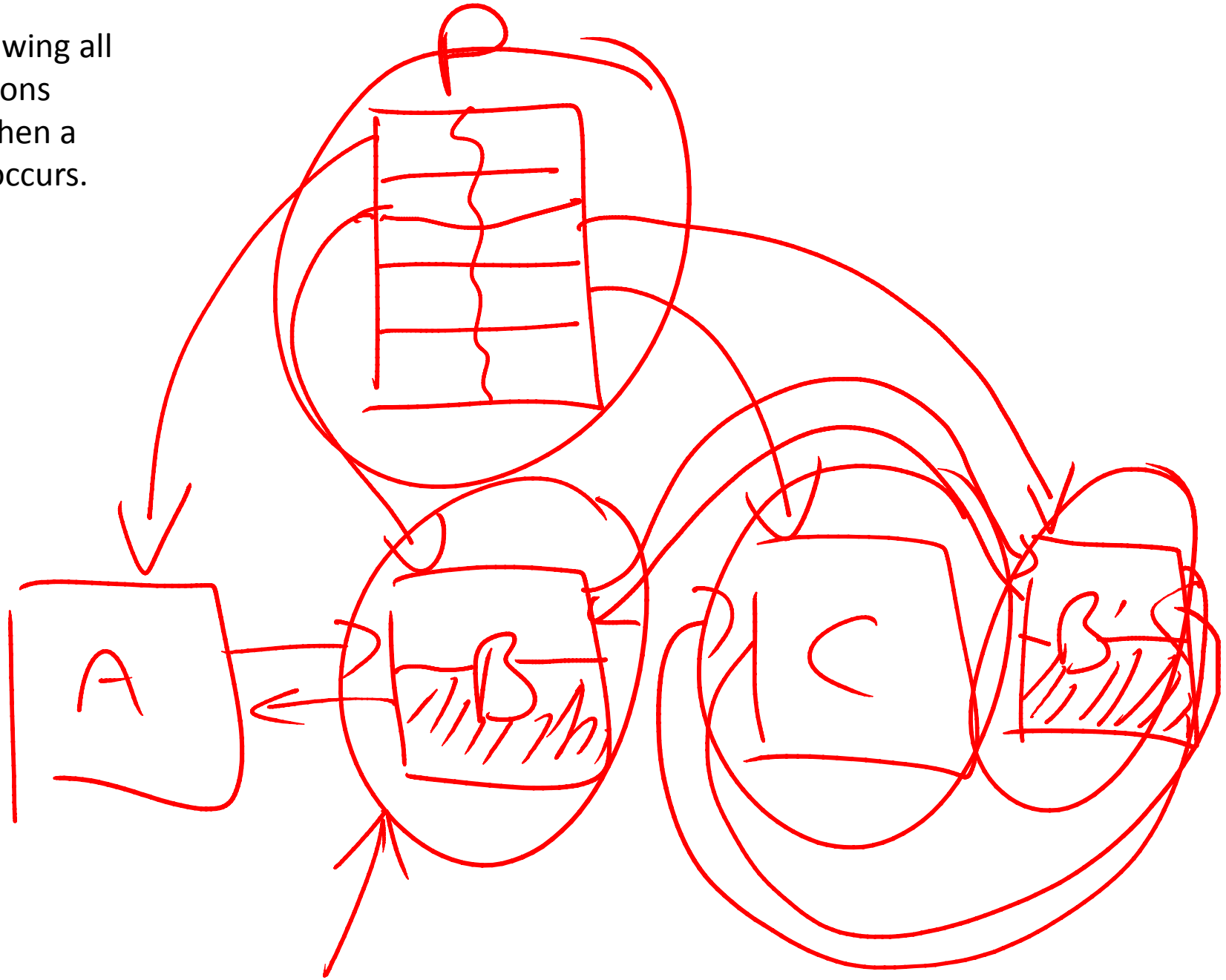


M5S29 Explaining the use of parity bits to help SQL Server determine where the end of the log is during crash recovery.



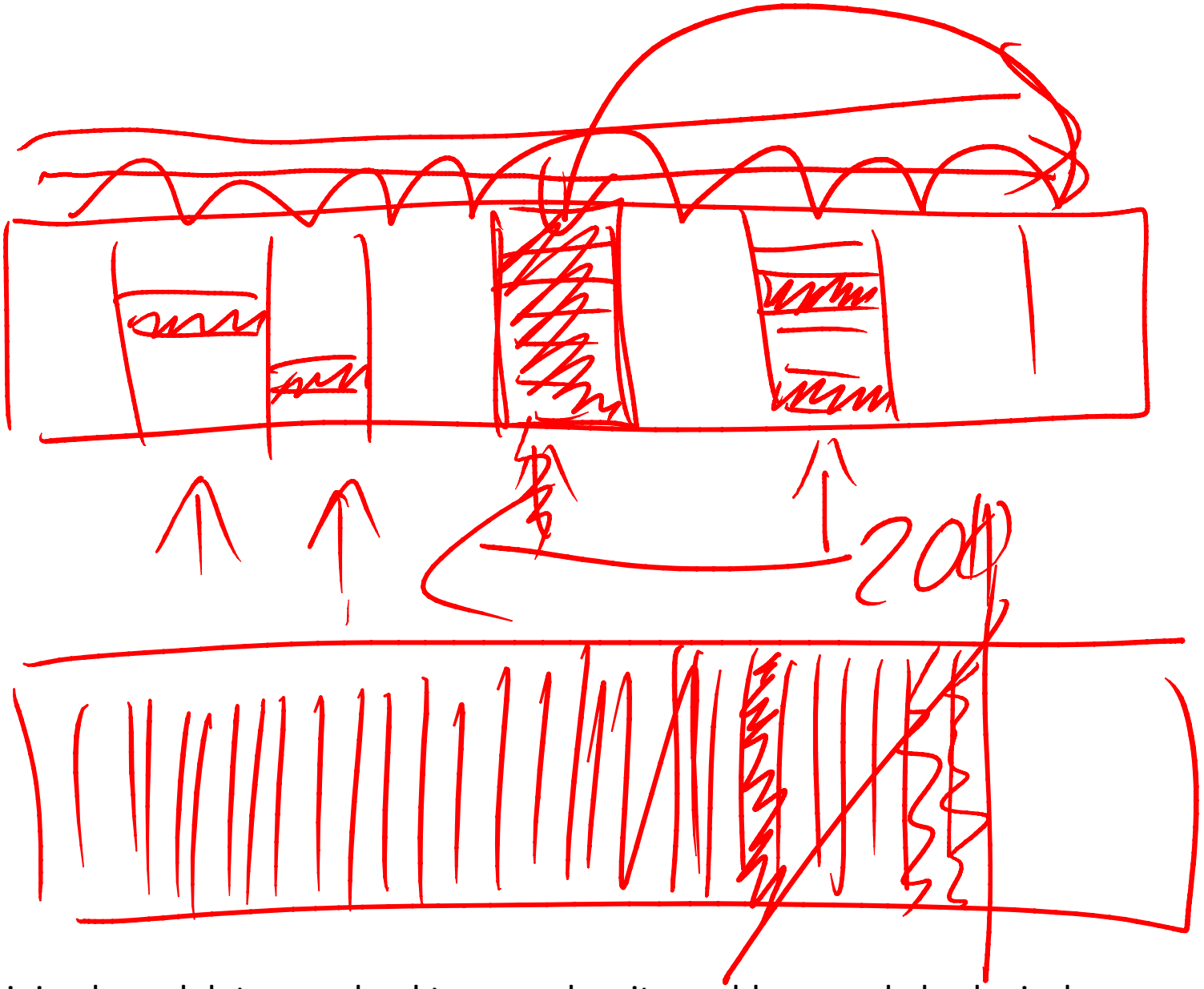
M8S5 Explaining how a record is found on a page using binary search rather than brute force search.

M8S15 Showing all the operations required when a page split occurs.

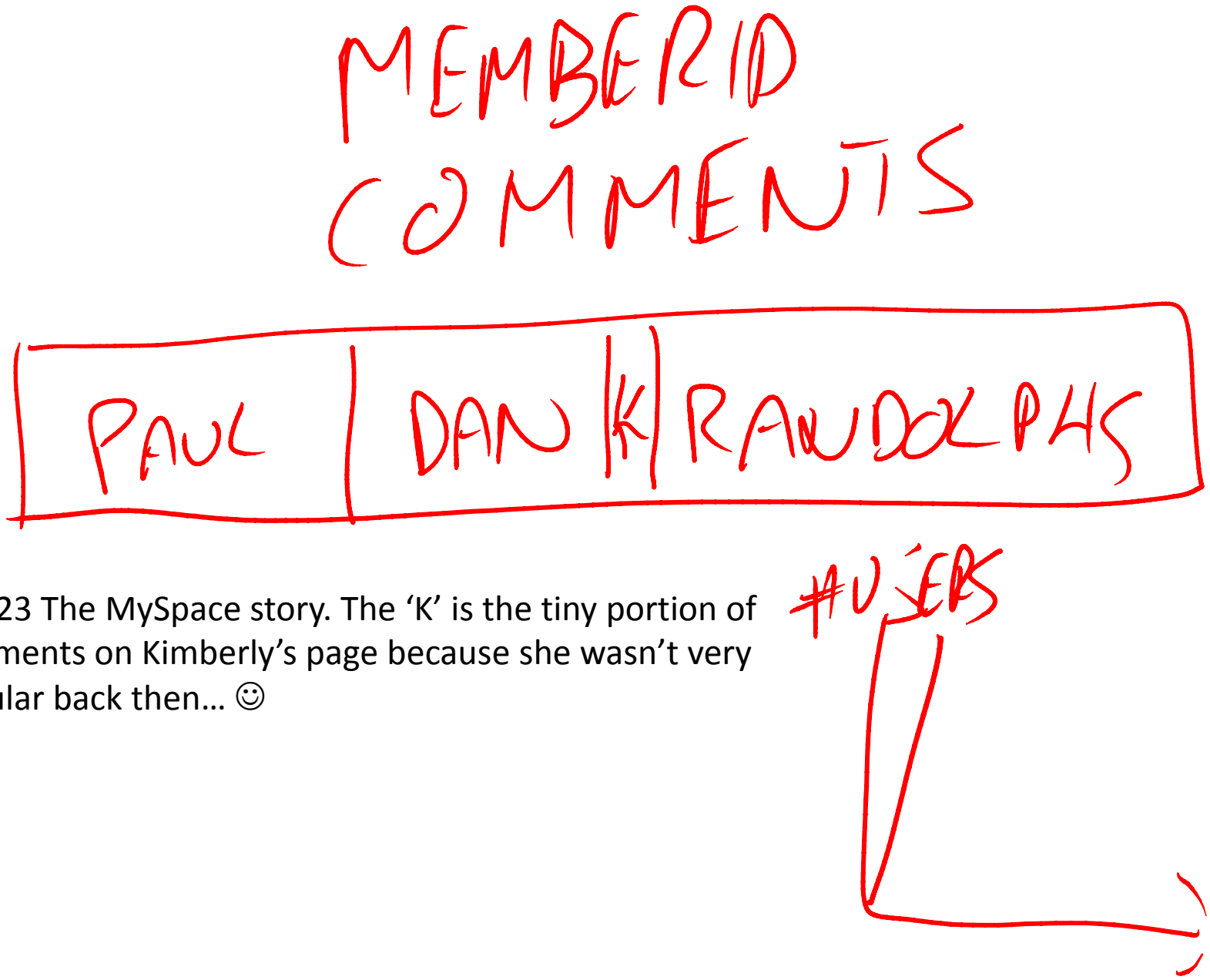


Row Size	No Split	Split	How Bad
1060	1244	6772	x5
100	344	5964	x18
10	256	10364	x45

M8S22 The numbers from the page split logging cost demo.



M8 Explaining how deletes can lead to page density problems and also logical fragmentation if an entire page in the middle of an extent is emptied and deallocated. If all pages in an extent are deallocated, the extent itself will be deallocated as well.



M8S23 The MySpace story. The ‘K’ is the tiny portion of comments on Kimberly’s page because she wasn’t very popular back then… ☺