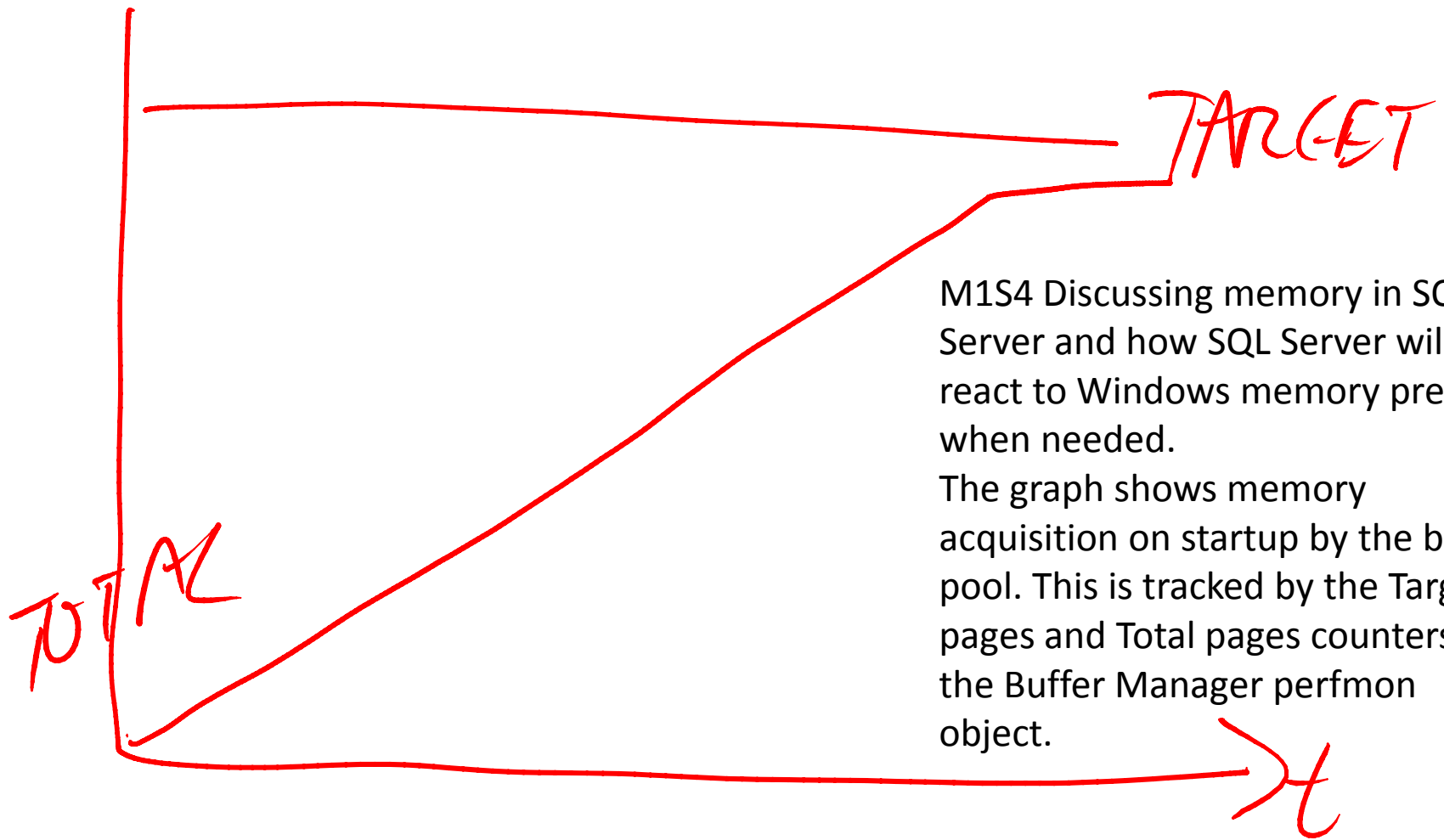
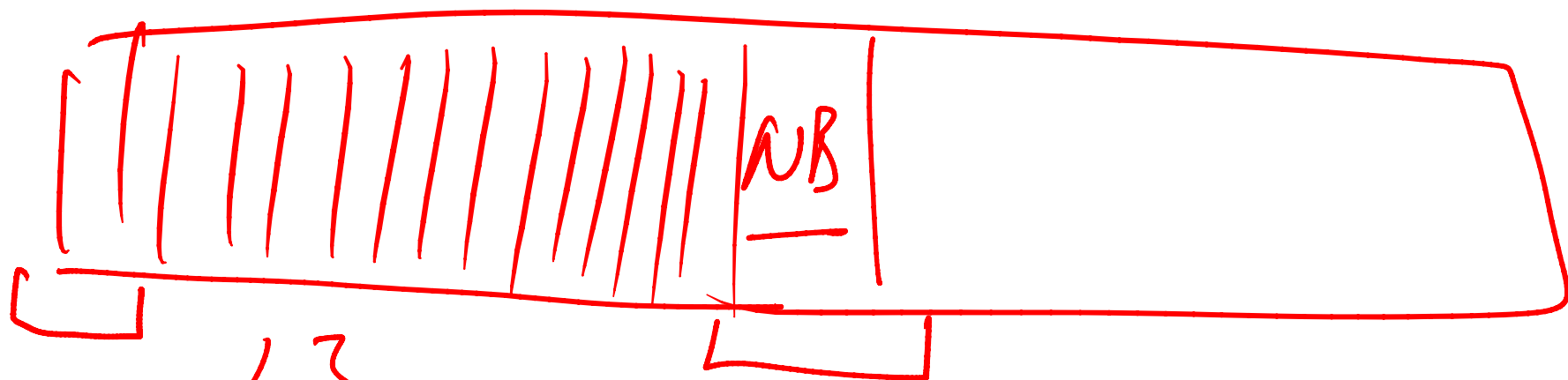




M1S4 Discussing memory in SQL Server and how SQL Server will react to Windows memory pressure when needed.

The graph shows memory acquisition on startup by the buffer pool. This is tracked by the Target pages and Total pages counters in the Buffer Manager perfmon object.



L3

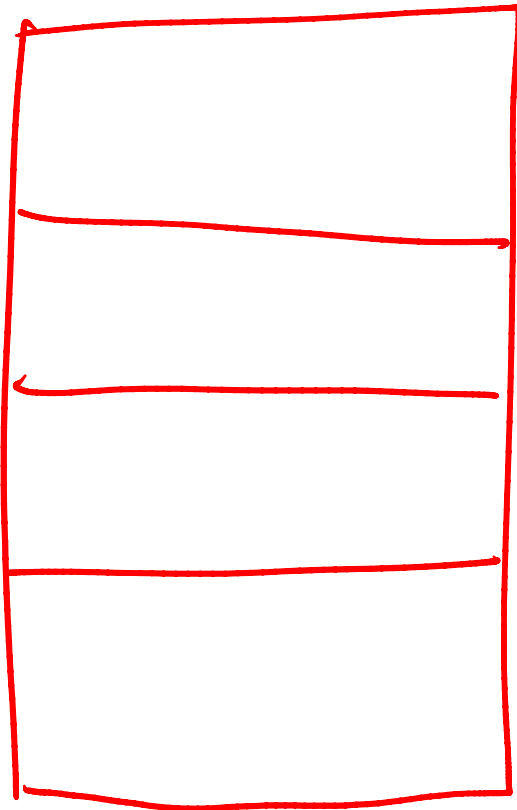
CACHE LINES

64b
128b

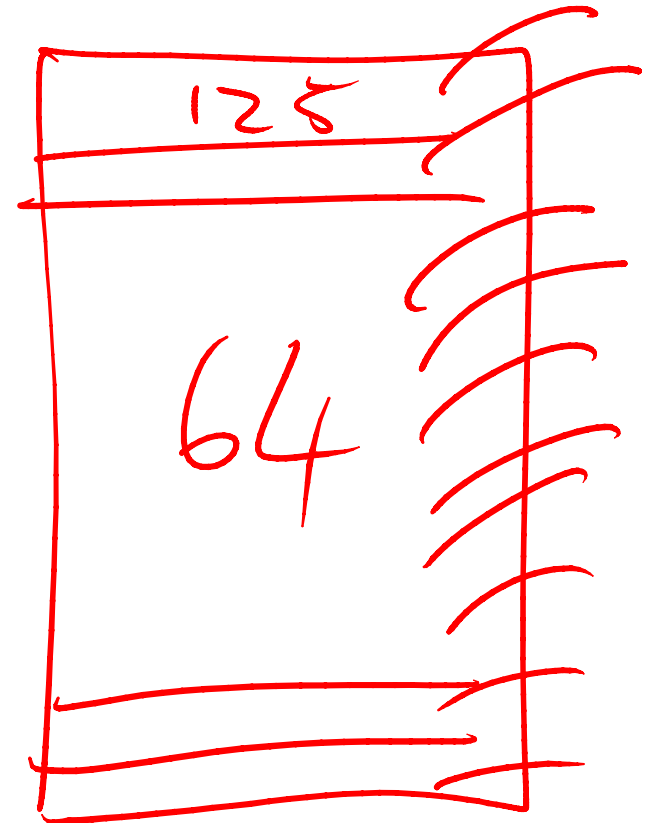
INVALIDATION

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

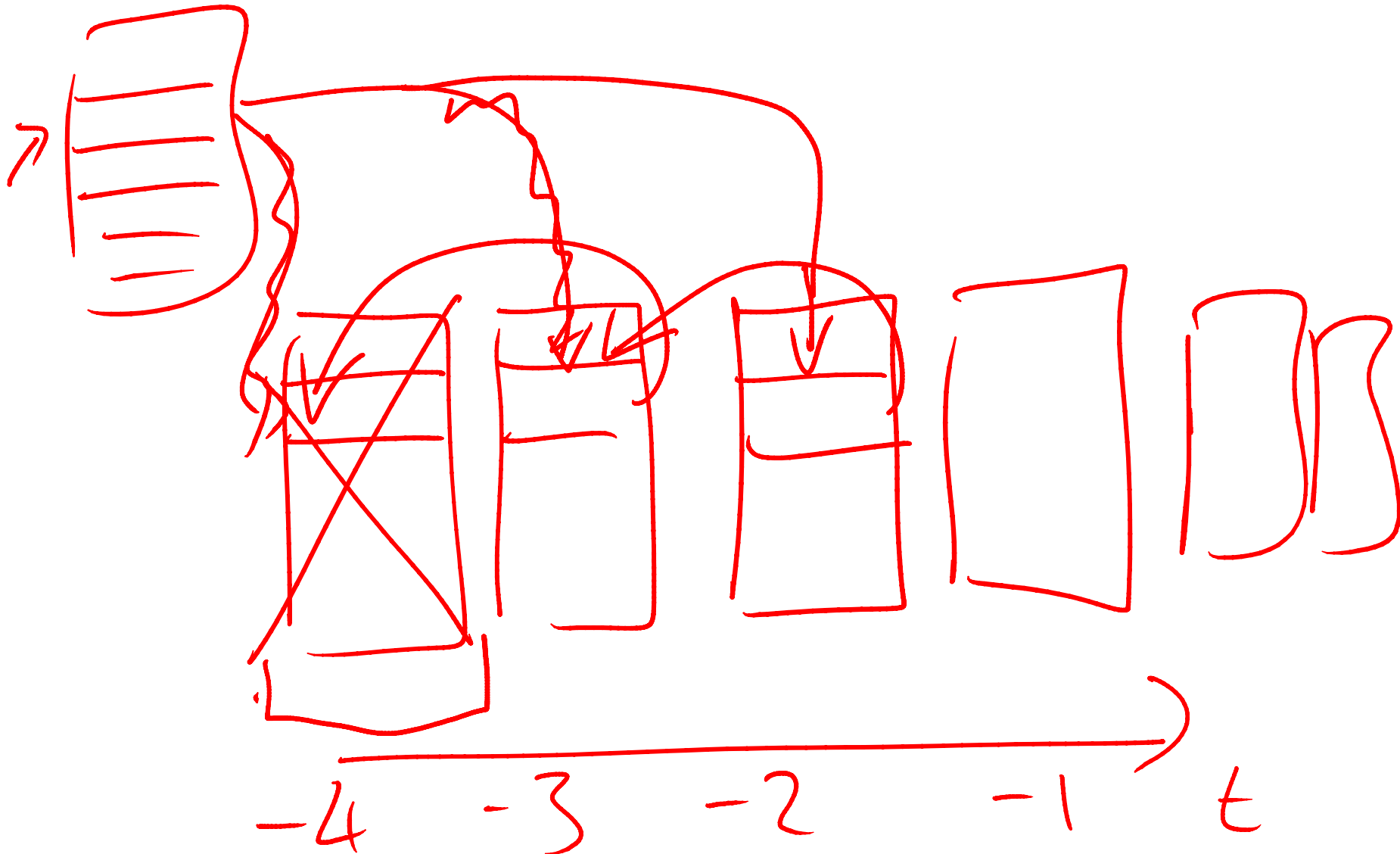
100	1900 byte
-----	-----------

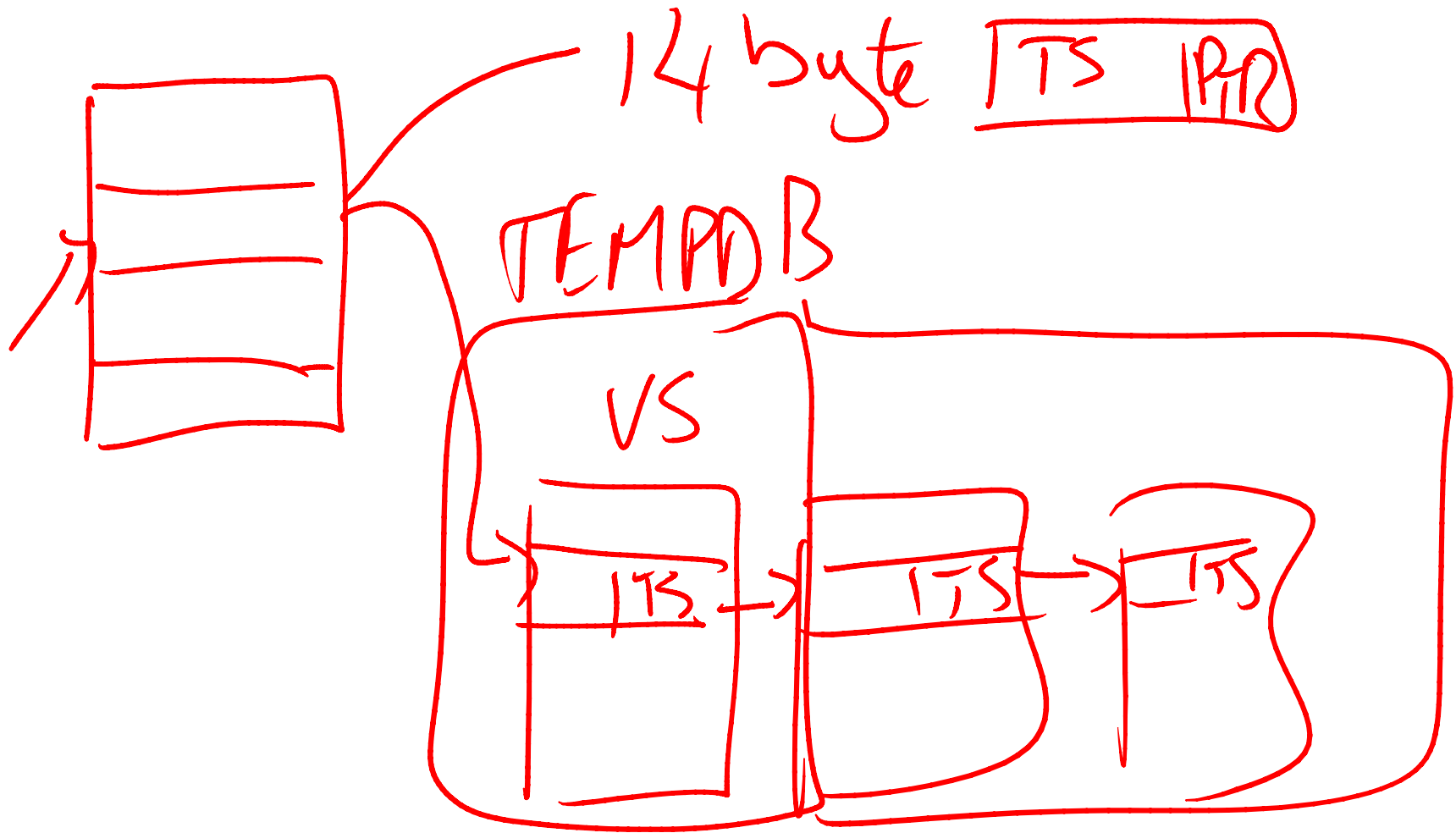


4 M1S12 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.

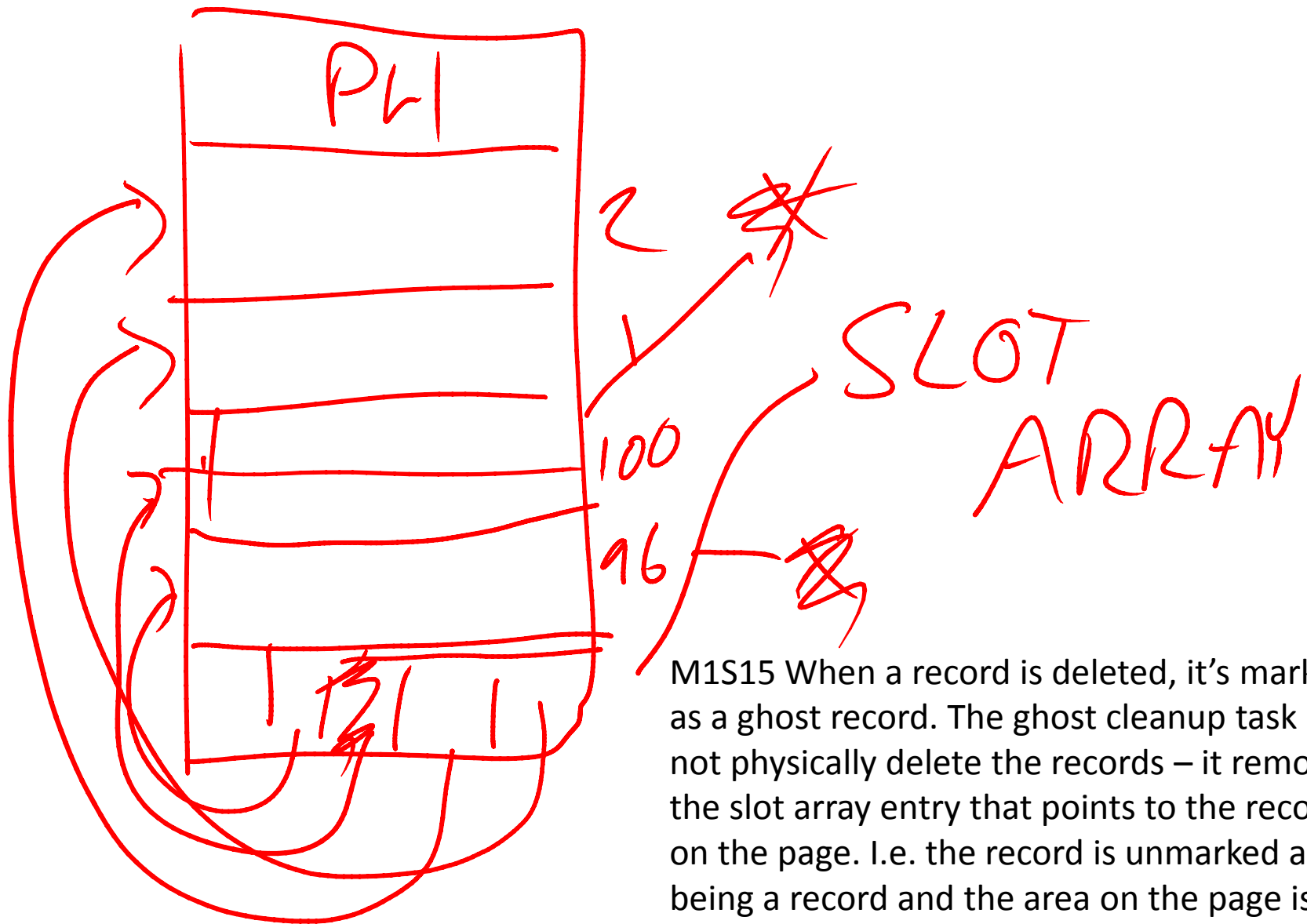


M1S14 & M2S35 Explaining how there's a new portion of the version store created every minute.

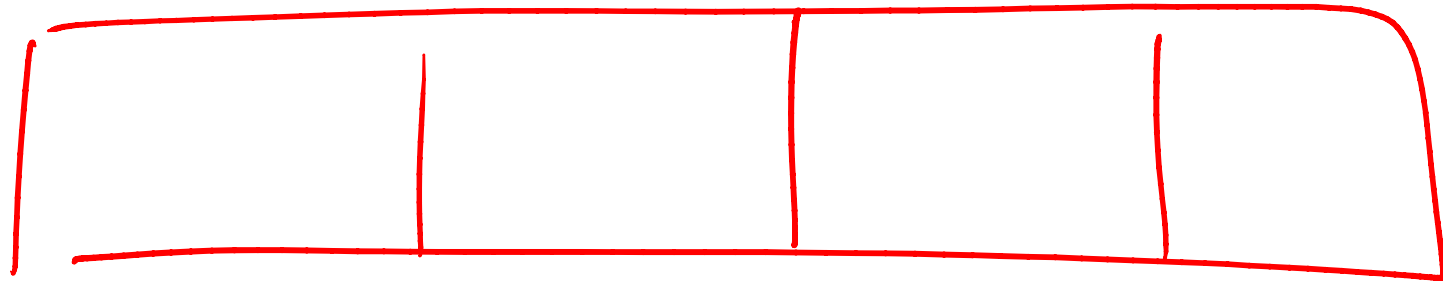




M1S14 When an update occurs to a record and snapshot isolation is enabled, the pre-change record is copied into the version store in tempdb. The changed record has a 14-byte tag on the end with the timestamp and pointer to previous record version. A chain of previous versions is possible.



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.



0 — 7 8 — 15 16 — 23

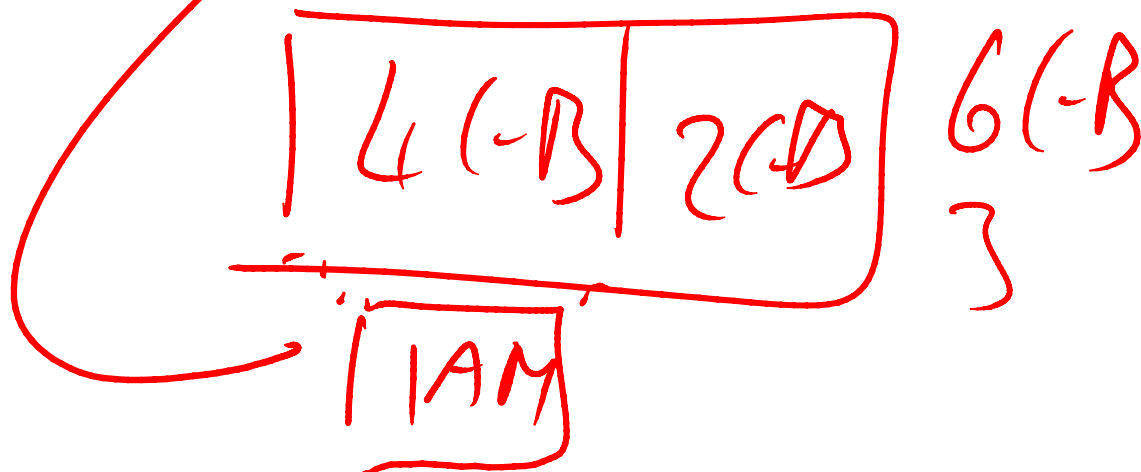


M1S24 Explaining the difference between mixed and dedicated extents. Mixed extent is one in which the 8 pages could be allocated to 8 different objects. Dedicated extent is one where all 8 pages are reserved for exclusive use of a single object.

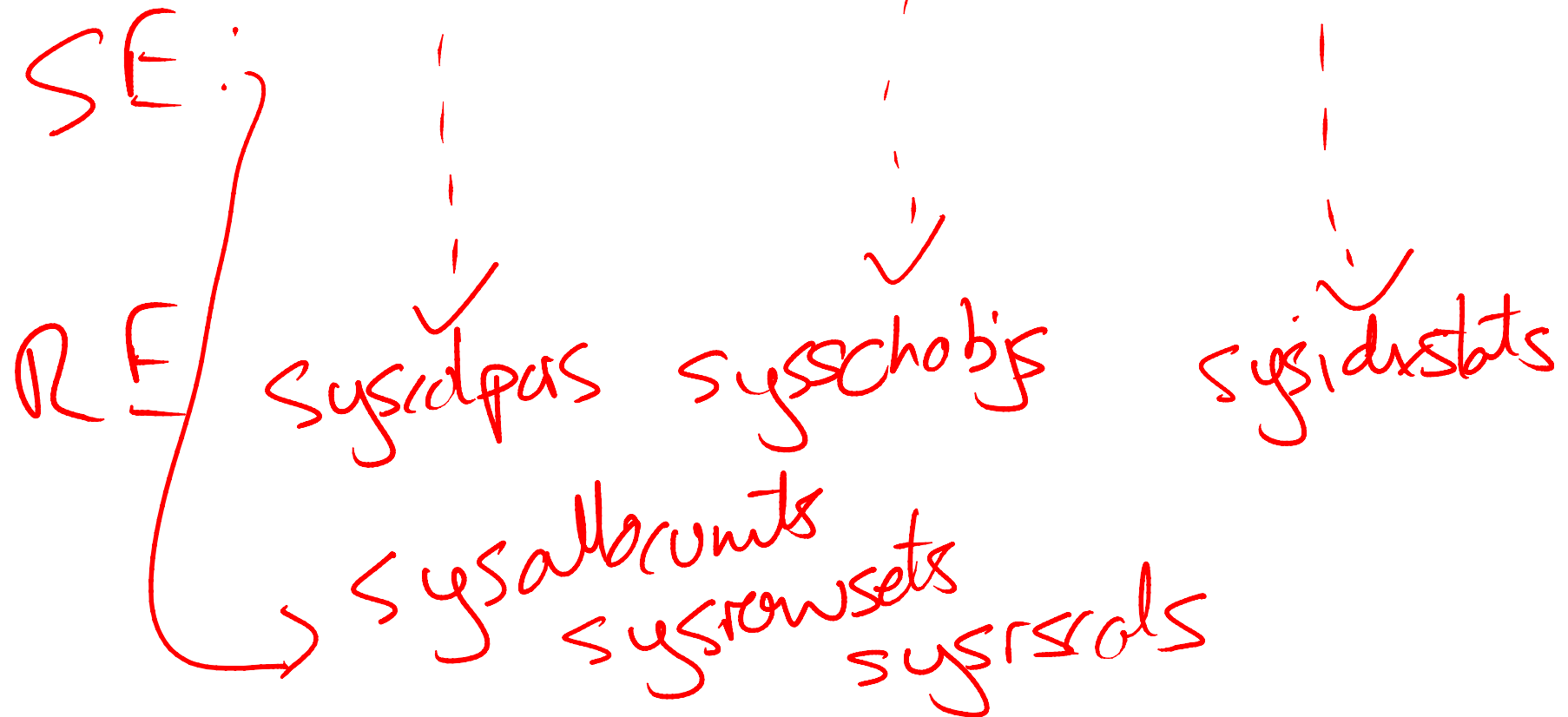
Later in the deck I refine 'object' to really be 'allocation unit'.



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.



2000: syscolumns, sysobjects, sysindexes

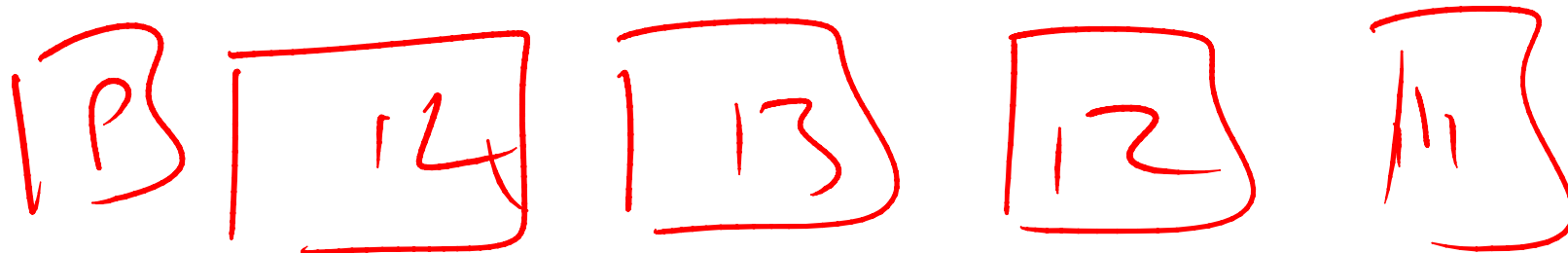


M1 Explaining the names of the hidden system tables that store column metadata.

SALES 1 TB

2014 2013 2012 2011

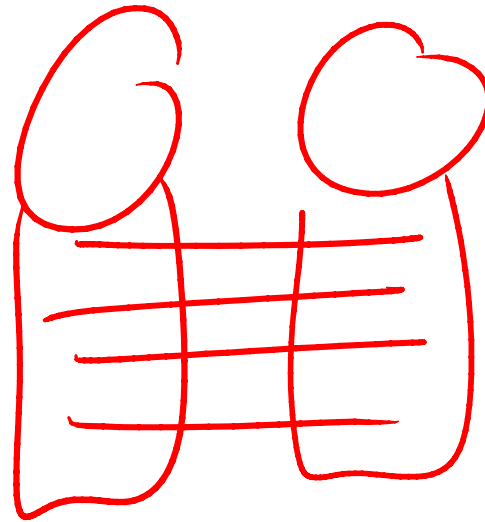
M2S7 Splitting a large database up into multiple filegroups can enable you to do small, targeted restores or even a partial restore from scratch – in the event that the database is damaged or destroyed.





M2S7 Explaining some ways to split up a database to allow targeted restore. On the left, auto parts sales system – makes sense to bring parts and NSW online first.

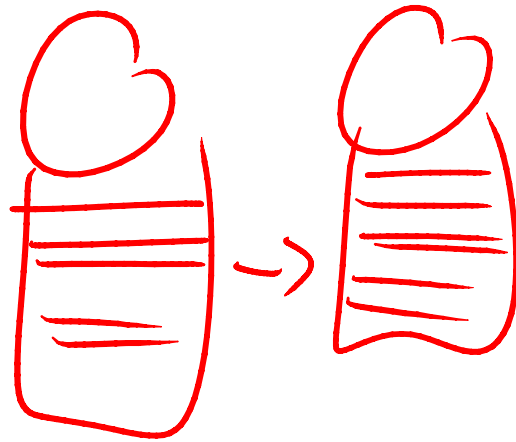
STRIPING



R0

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

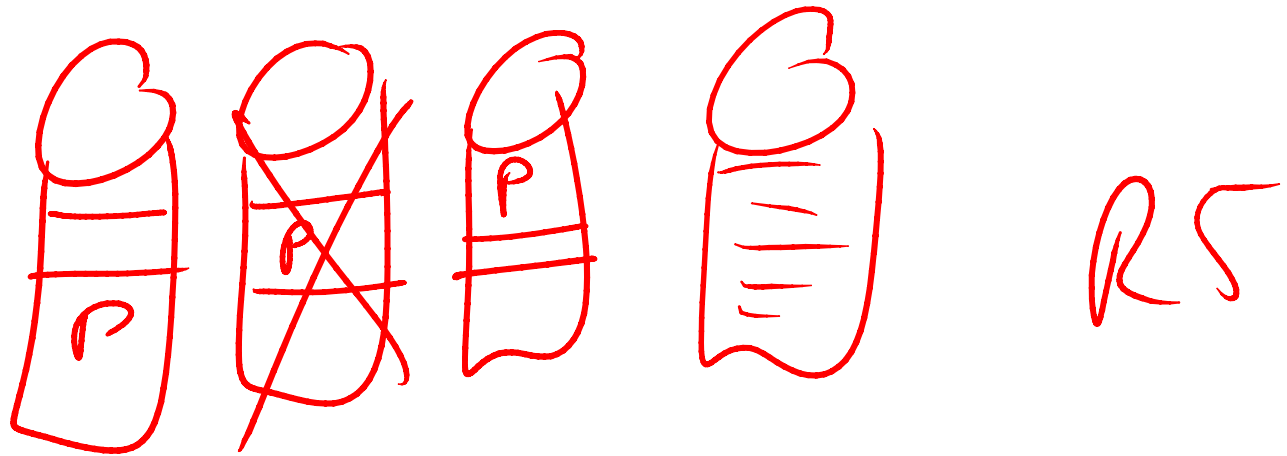
MIRRORING



RA1

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

NOTATING PARITY

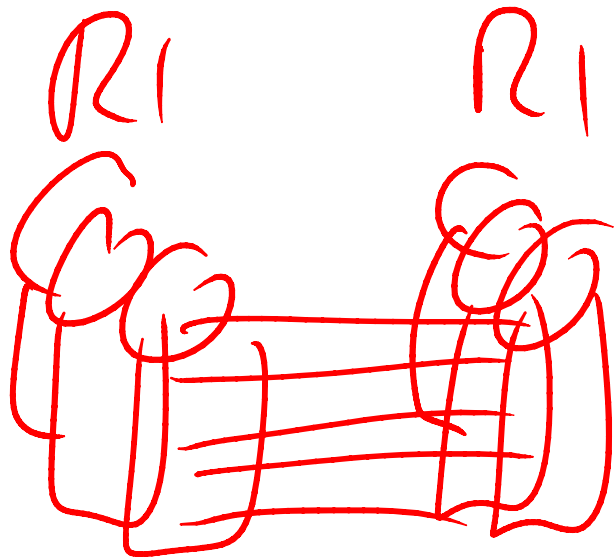


http://en.wikipedia.org/wiki/Standard_RAID_levels

$$R10 = R1 + G$$

= MIRRORING

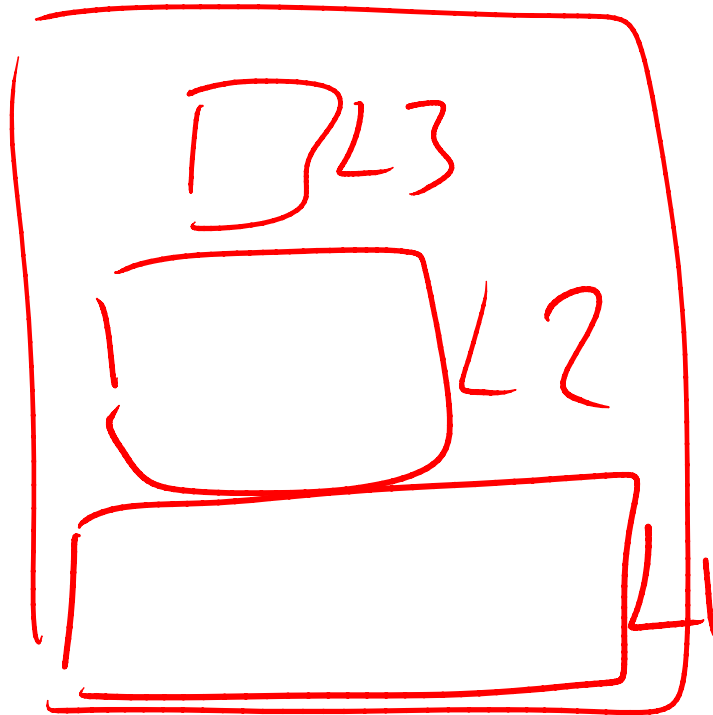
+ STRIPING



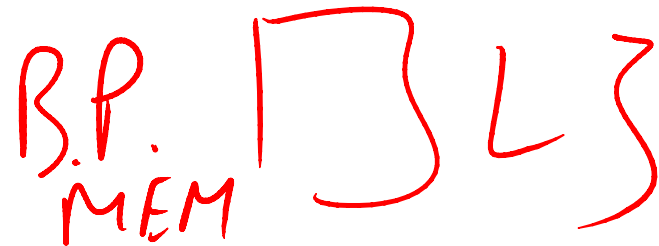
sys.dm_io_virtual_file_stats
dm_io_pending_io_requests

M2 Tracking I/O latencies: see <http://www.sqlskills.com/blogs/paul/how-to-examine-io-subsystem-latencies-from-within-sql-server/>

CPU

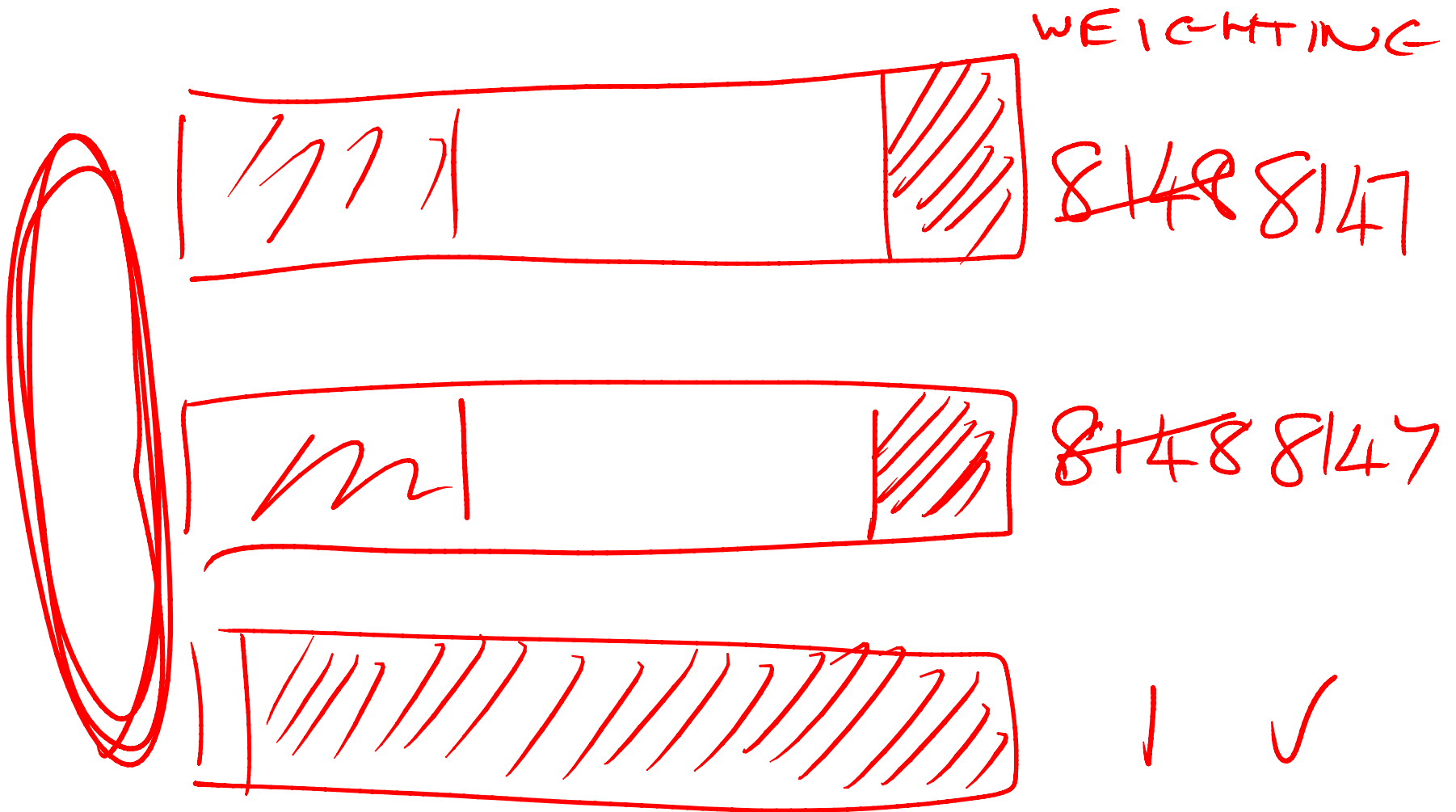


I/O LATENCY

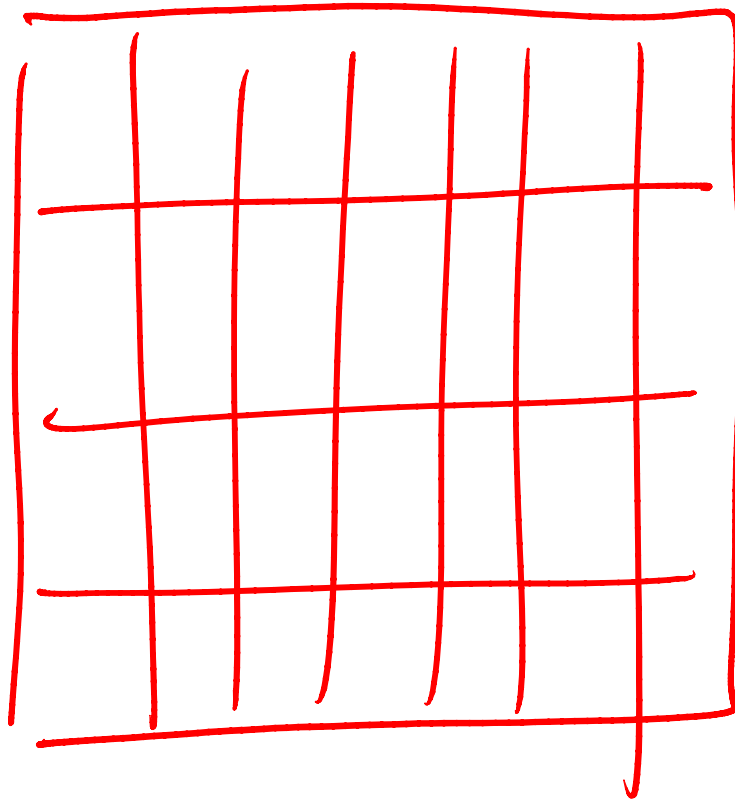


I/O SUBSYSTEM

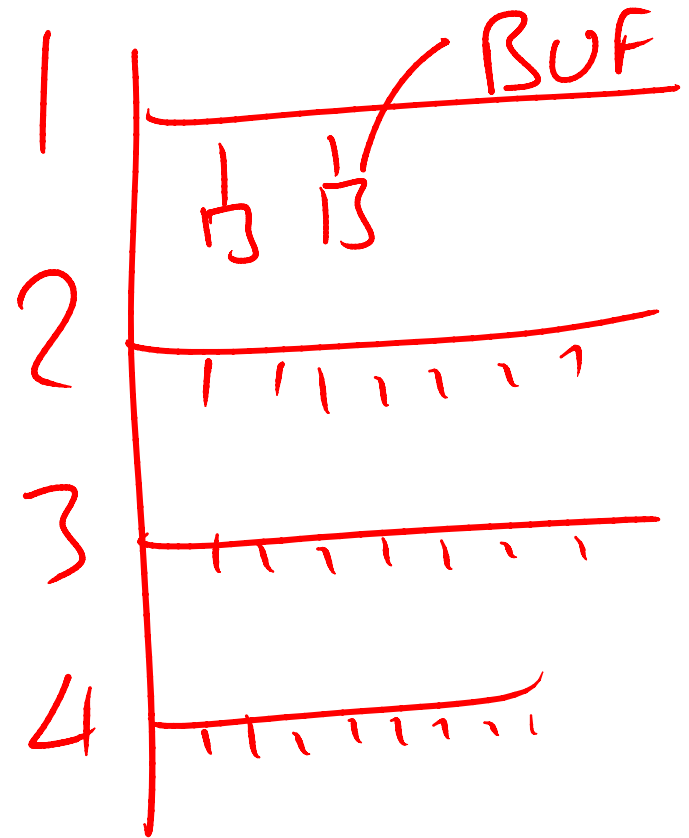
M2S17 Explaining 2004 Buffer Pool Extension. It's like have L1-L3 caches for your data in memory. BPE sits in between memory and the I/O subsystem.



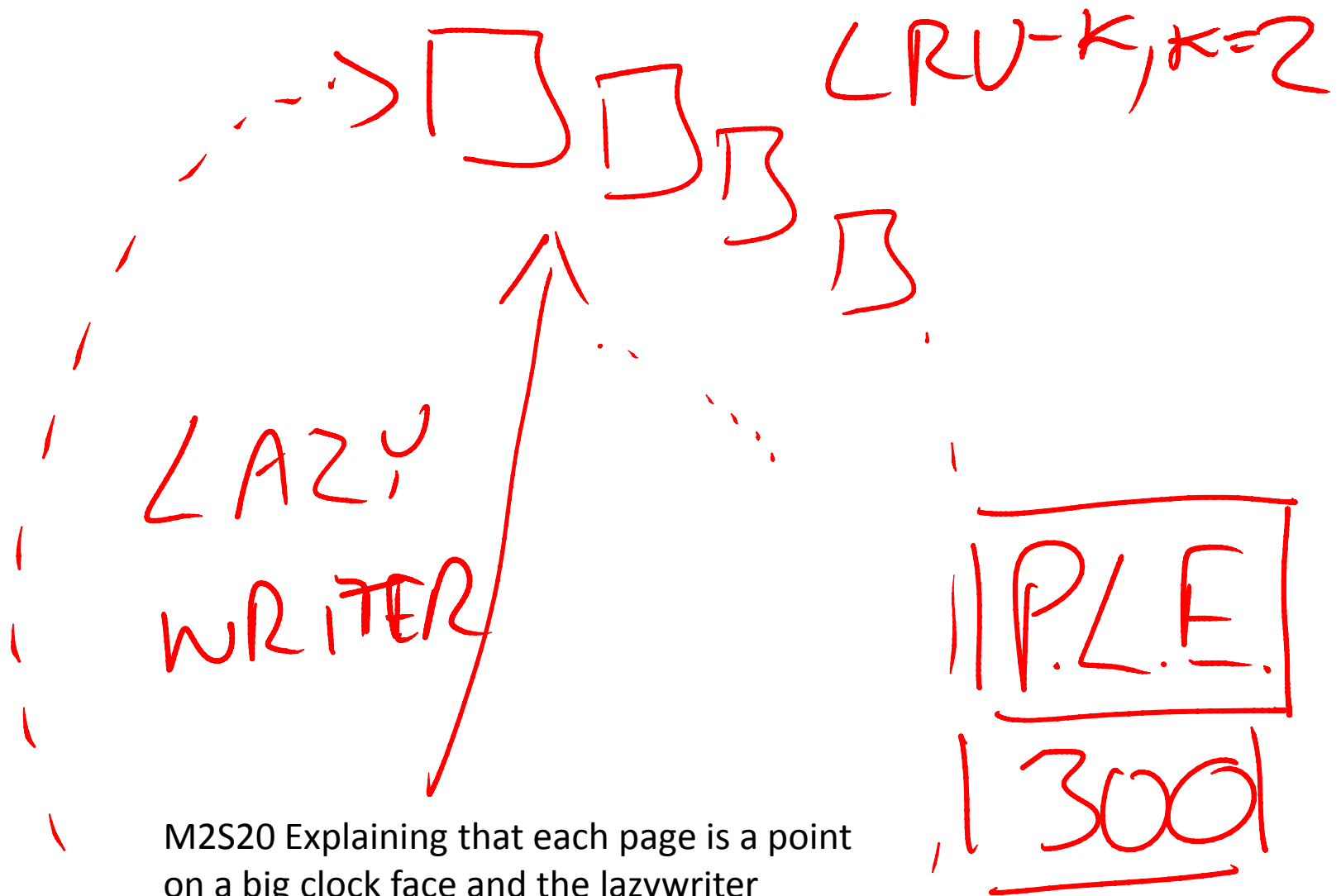
M2S19 Explaining round-robin allocation and proportional fill. Each file has a proportional fill weighting. Allocations happen proportionally more frequently from files with more free space.



HASH LISTS



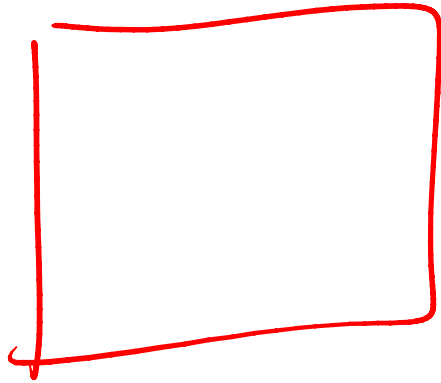
M2S20 Explaining the hash lists per database that allow the buffer pool to easily determine whether a page is already in memory or not



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.

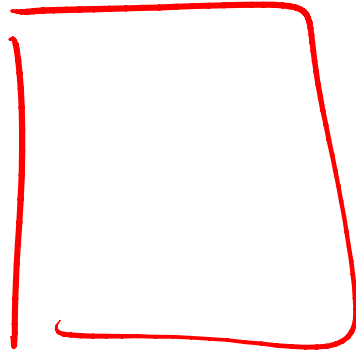
PLE = 300 is a nonsense threshold for tuning. See next slide.

See <http://www.sqlskills.com/blogs/paul/page-life-expectancy-isnt-what-you-think/>

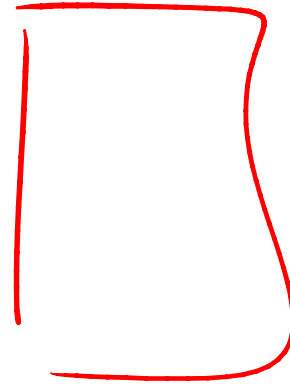


PLE
4000

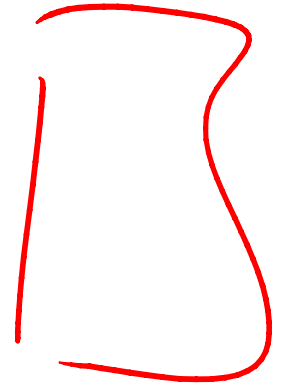
4000



PLE
4000

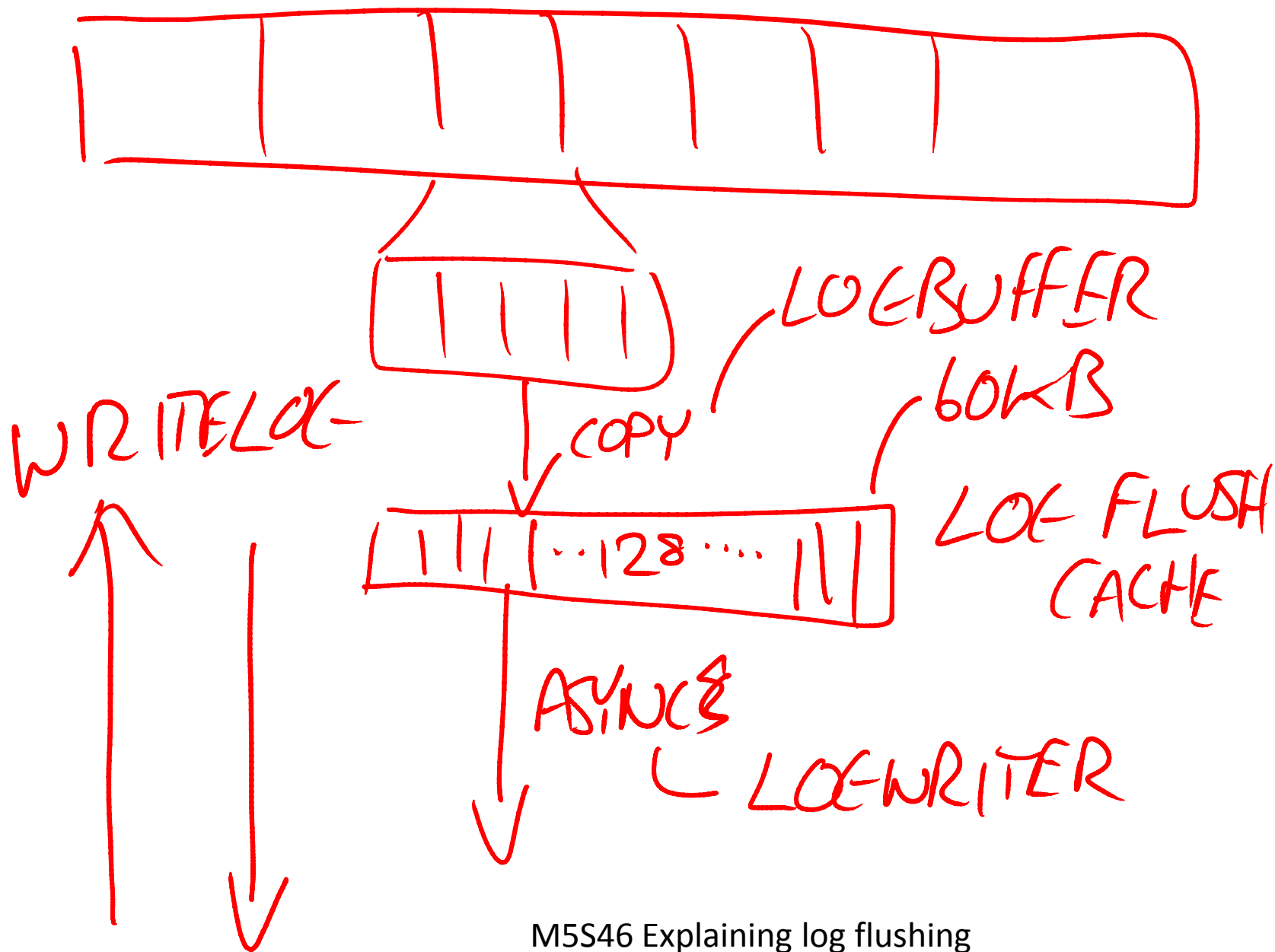


PLE
4000

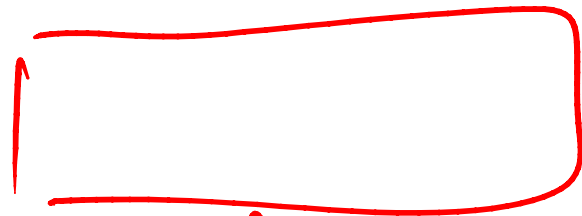


PLE
1000

3625



M5S46 Explaining log flushing



ASYNC

< 2012

32 logblocks

48-16

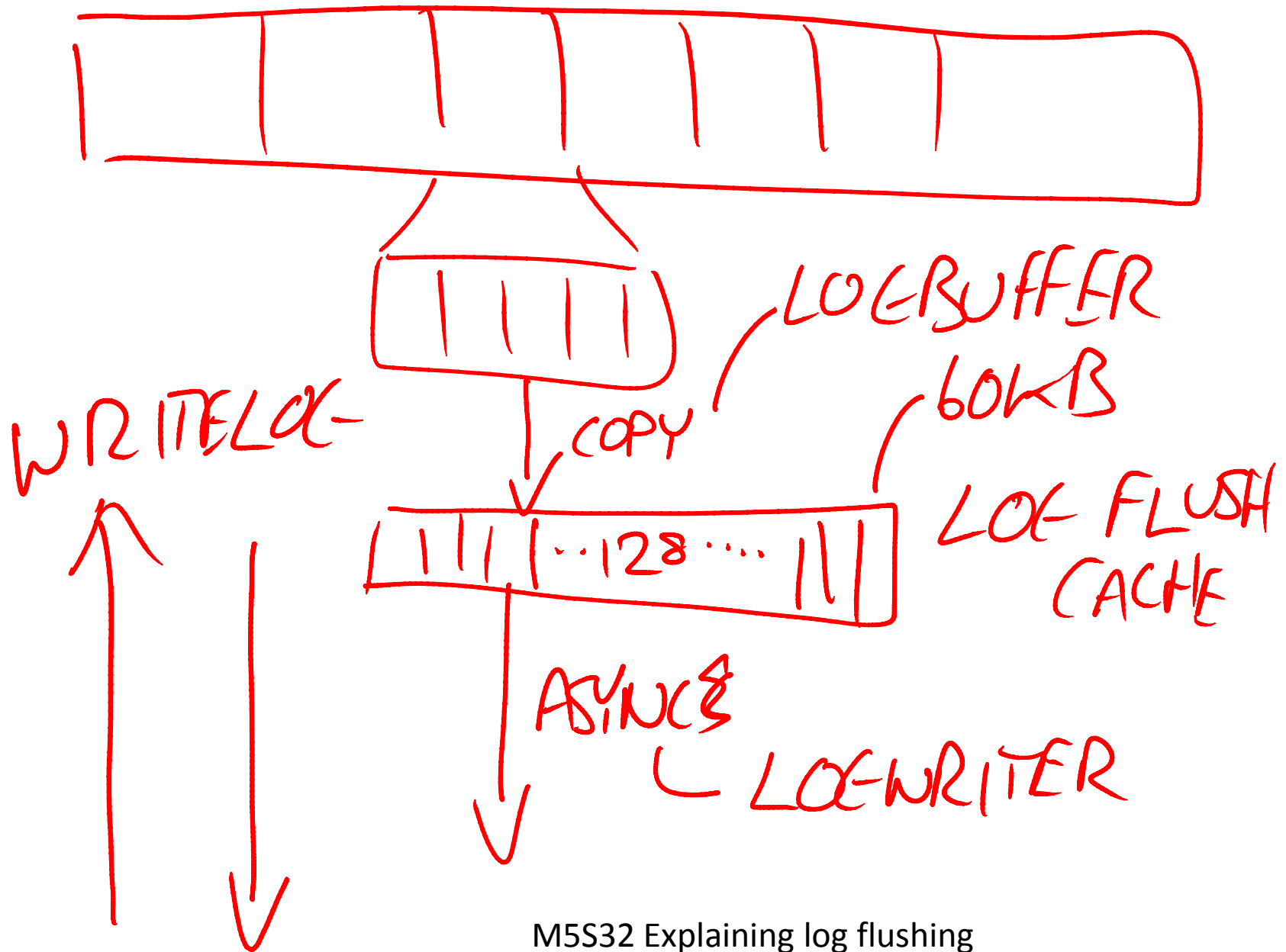
2012 >

128-16

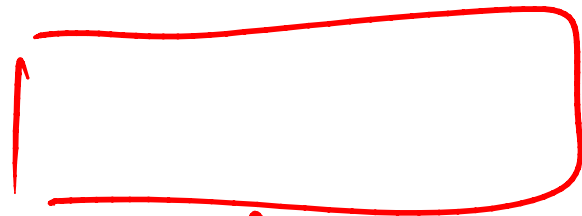
112 logblocks

sys.dm-io-pending-io-requests

M5S46 Explaining log flushing. There is a limit of 32 outstanding async log block I/Os pre-2012. The limit is 112 in 2012.



M5S32 Explaining log flushing



ASYNC



< 2012

32 logblocks

48-16

2012 >

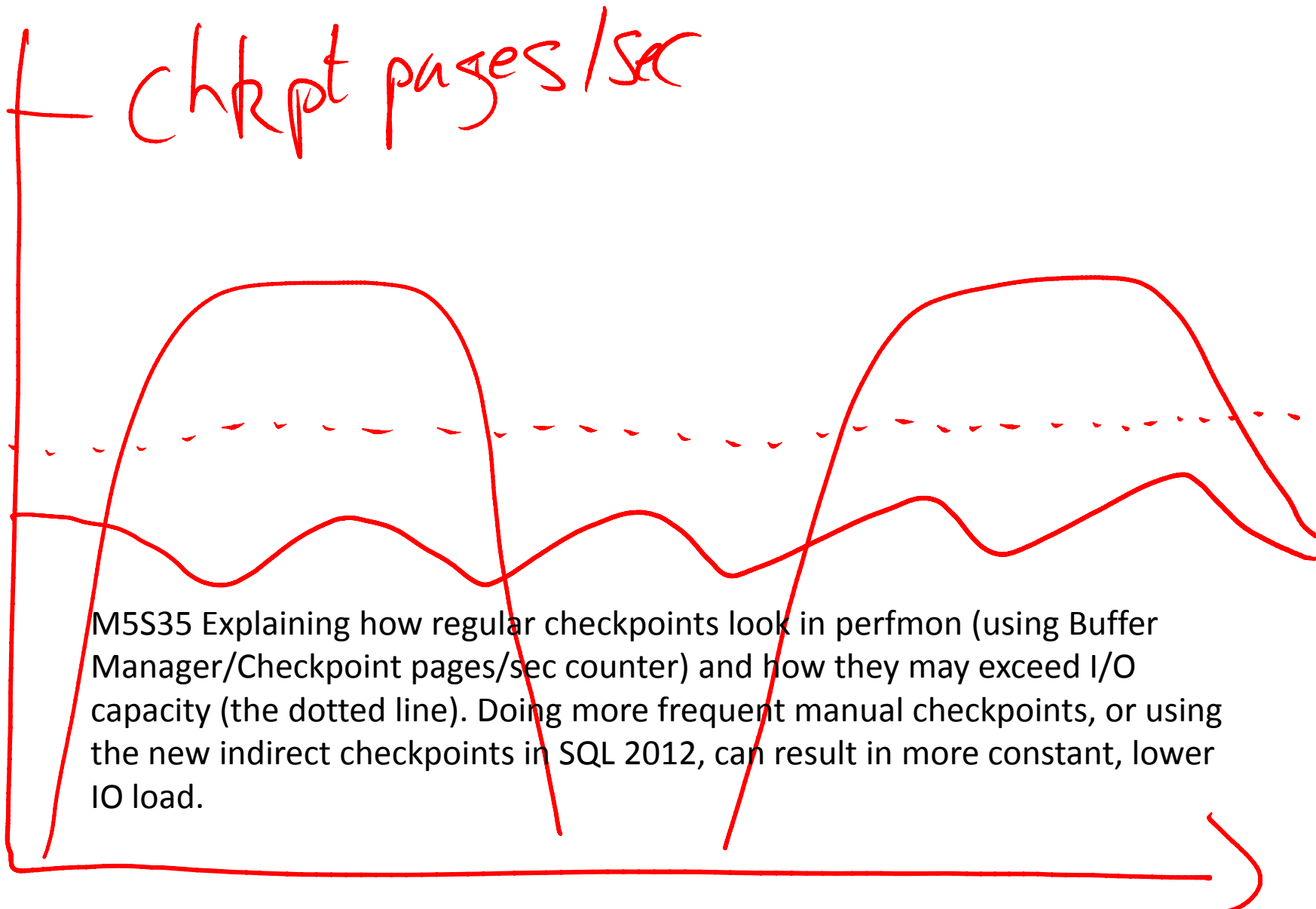
128-16

112 logblocks

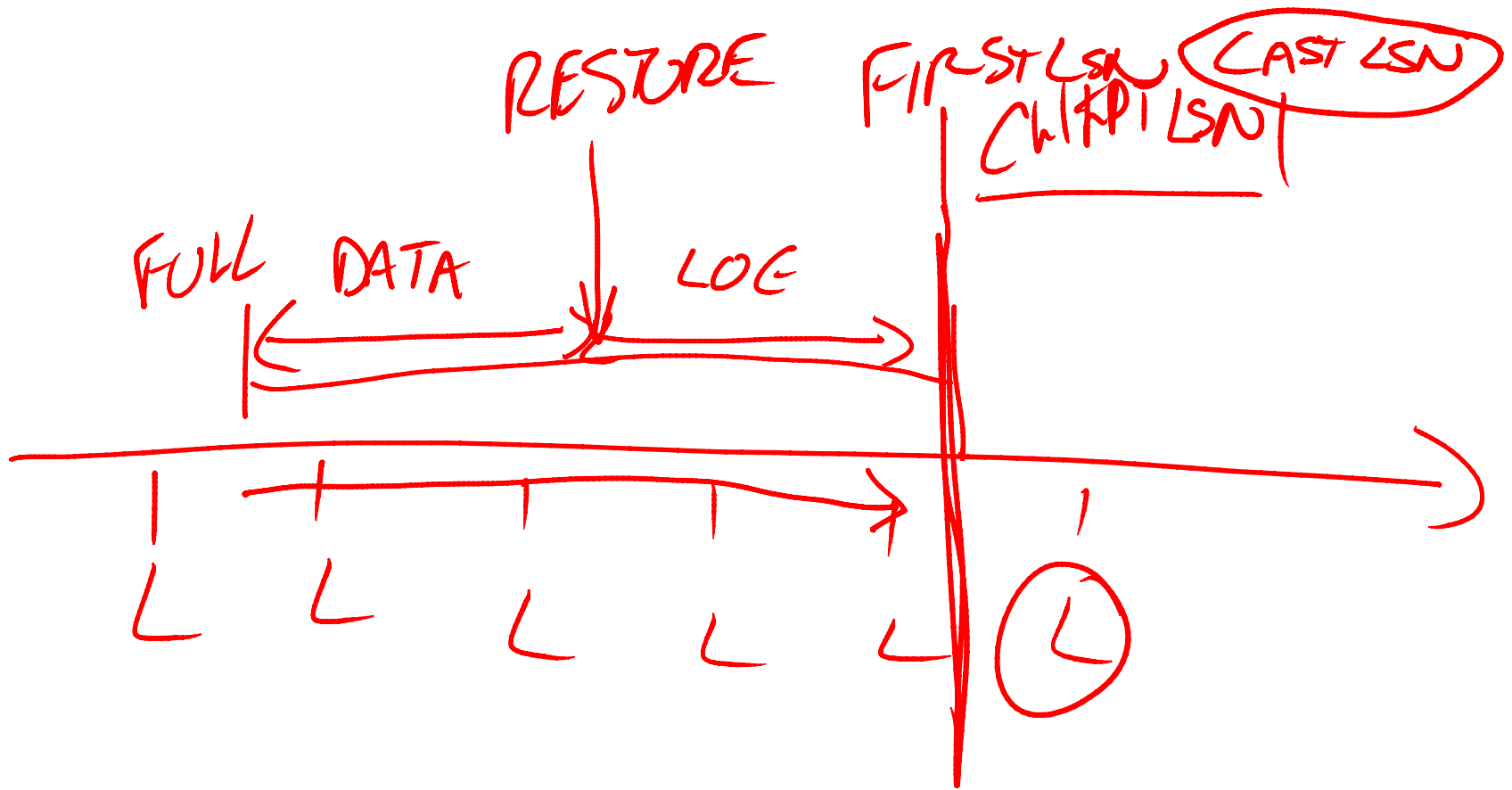
sys.dm-io-pending-io-requests

M5S32 Explaining log flushing. There is a limit of 32 outstanding async log block I/Os pre-2012. The limit is 112 in 2012.

chkpt pages/sec

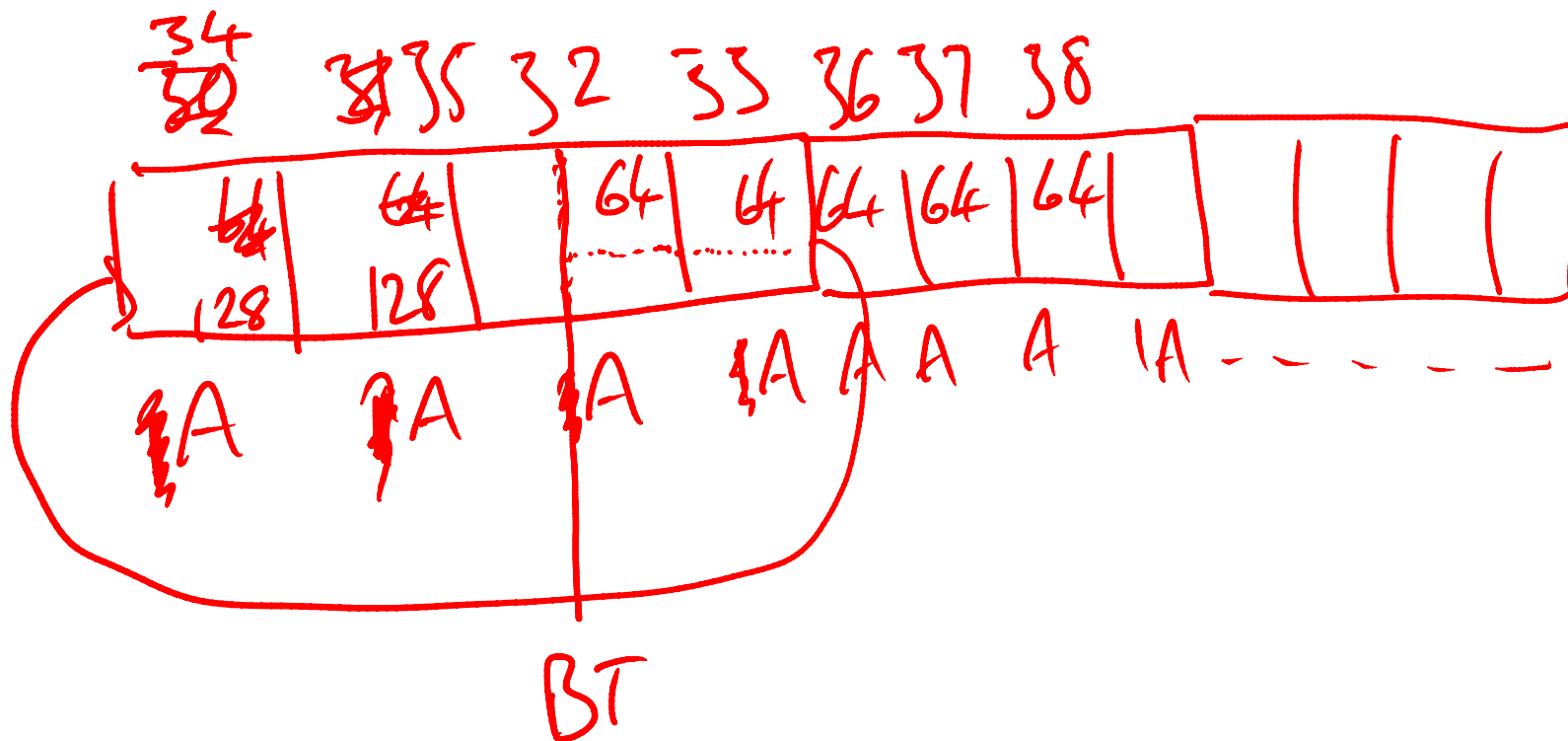


M5S35 Explaining how regular checkpoints look in perfmon (using Buffer Manager/Checkpoint pages/sec counter) and how they may exceed I/O capacity (the dotted line). Doing more frequent manual checkpoints, or using the new indirect checkpoints in SQL 2012, can result in more constant, lower IO load.



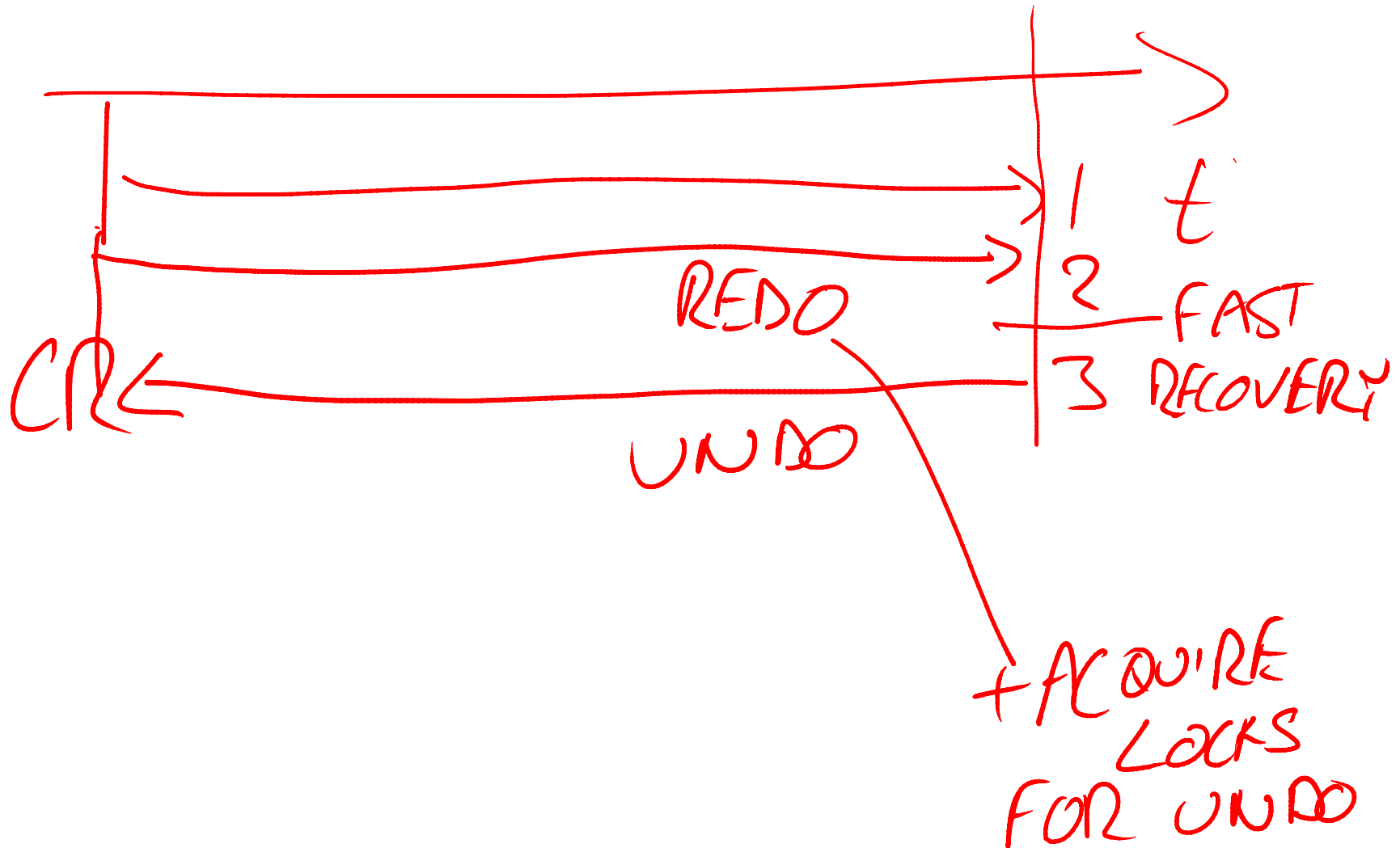
M5S43 It's possible for concurrent log backups to occur with a data backup (full or differential). Log clearing cannot occur though, as the data backup will back up the log. When the data backup finishes, deferred log clearing takes place.

The point in time you get when you restore a data backup (and don't restore any more) is the point when the data reading portion of the backup ended (marked RESTORE in the diagram).



M5S49 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.

M5S50 Explaining the three passes through the log during crash recovery. Locks for UNDO are acquired during pass #2. Fast recovery can then let you in to the database.



LOG REUSE WAIT DESC BLOG POST

M5S56 I did a blog post explaining the various wait_desc values. See http://www.sqlskills.com/blogs/paul/worrying-cause-log-growth-log_reuse_wait_desc/

100,000

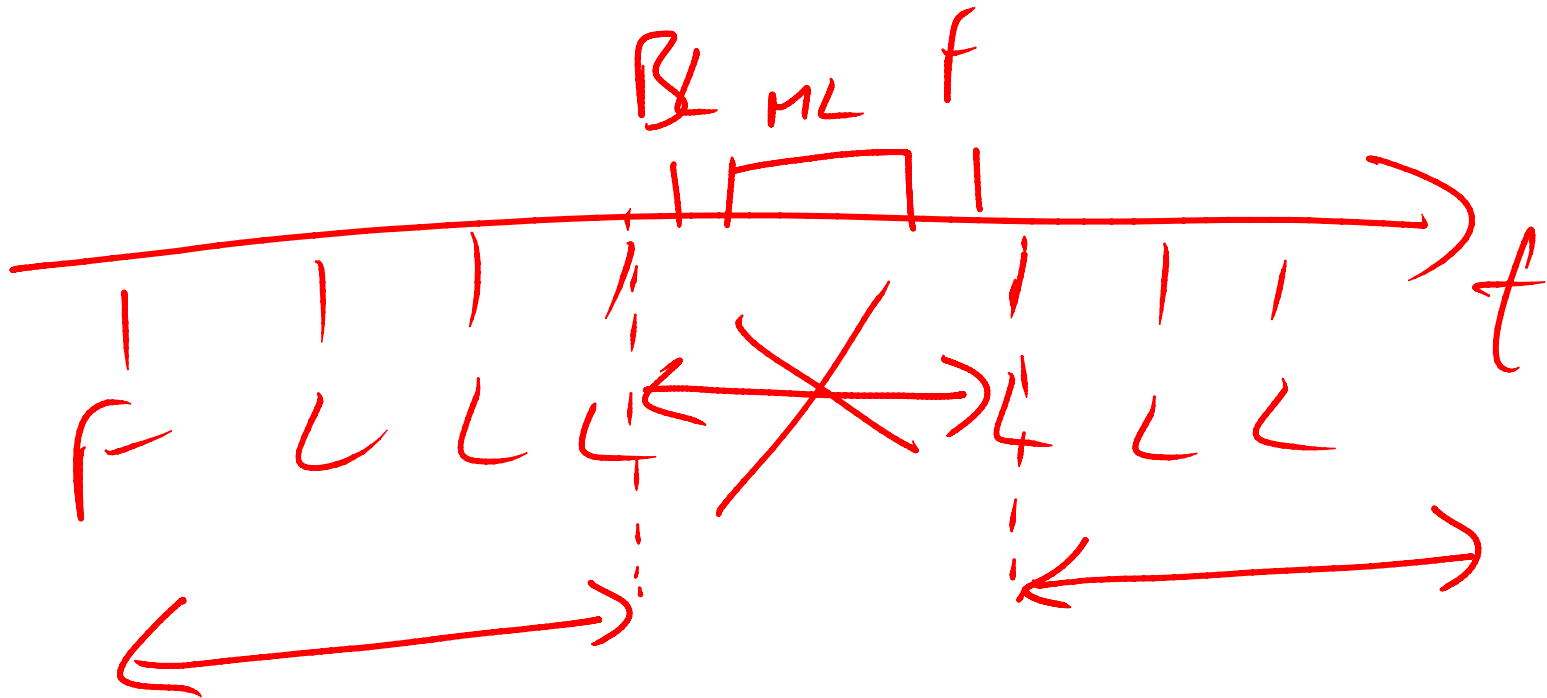
10

~~INS
INS
INS~~

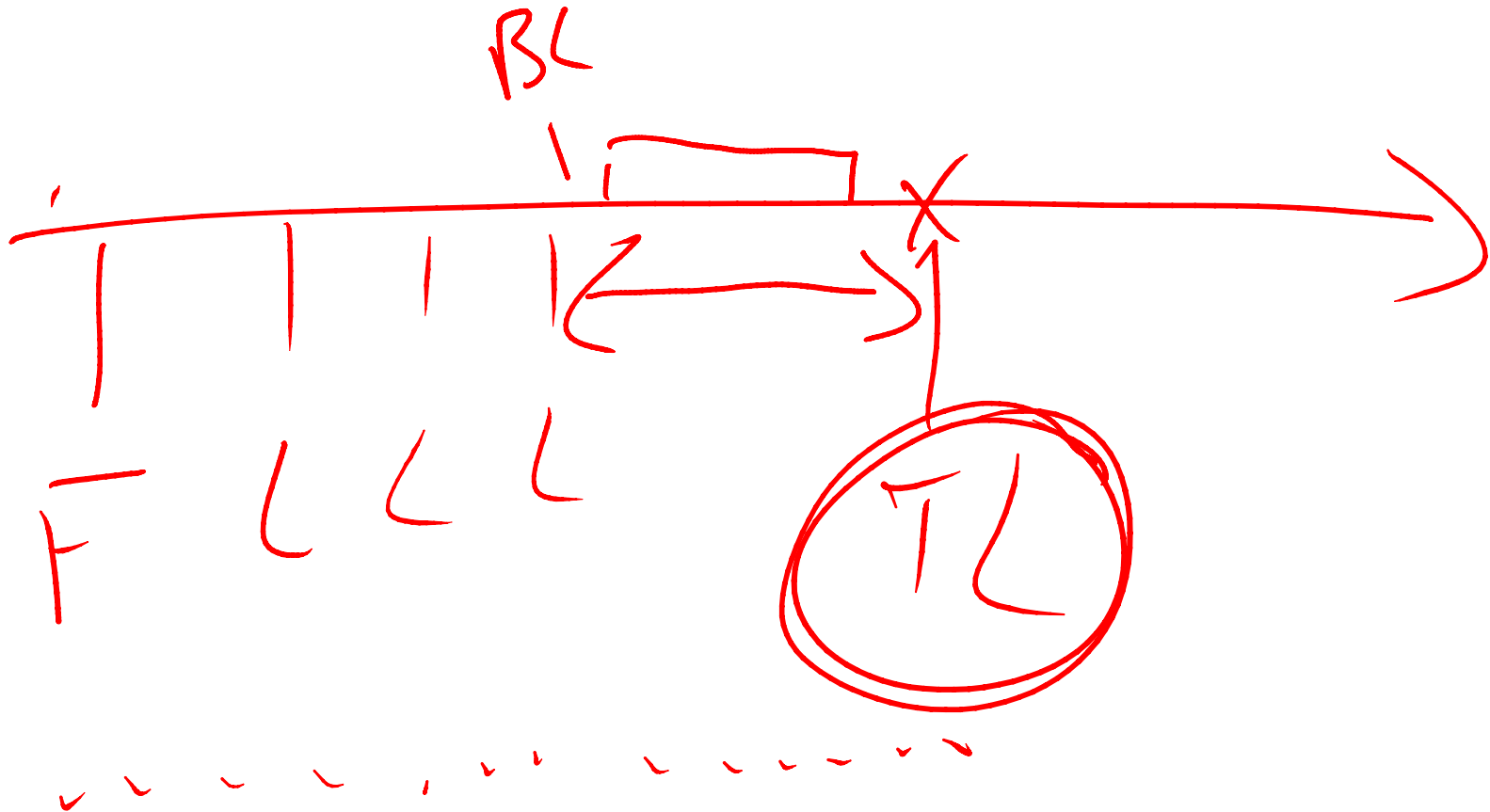
F - PAGE
F - PAGE

10

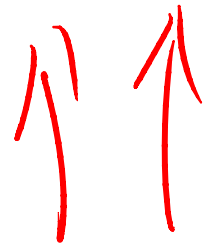
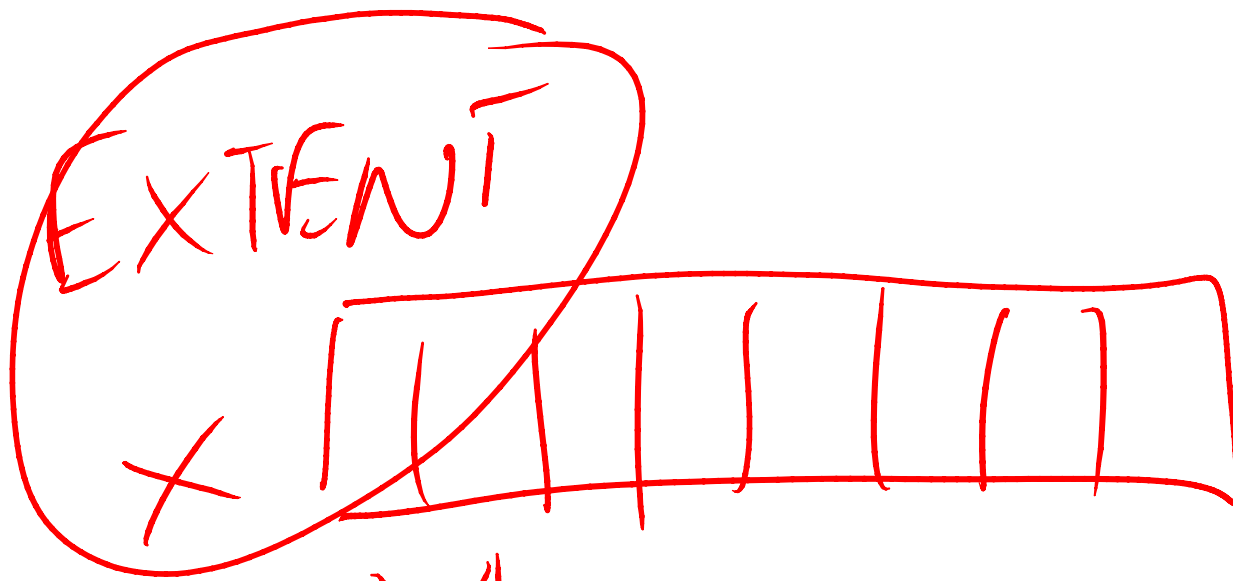
M5S59 Explaining that full recovery doesn't mean there's always one log record per operation. An index rebuild in full recovery will log full LOP_FORMAT_PAGE log records instead of multiple LOP_INSERT_ROWS log records. It's more efficient, but is still fully logged.



M5S61 If a log backup contains a minimally-logged operation, it cannot be used to do a point-in-time restore using STOPAT for any point in time between the beginning and end of the backup. It can be used as part of a restore sequence as normal though.



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.

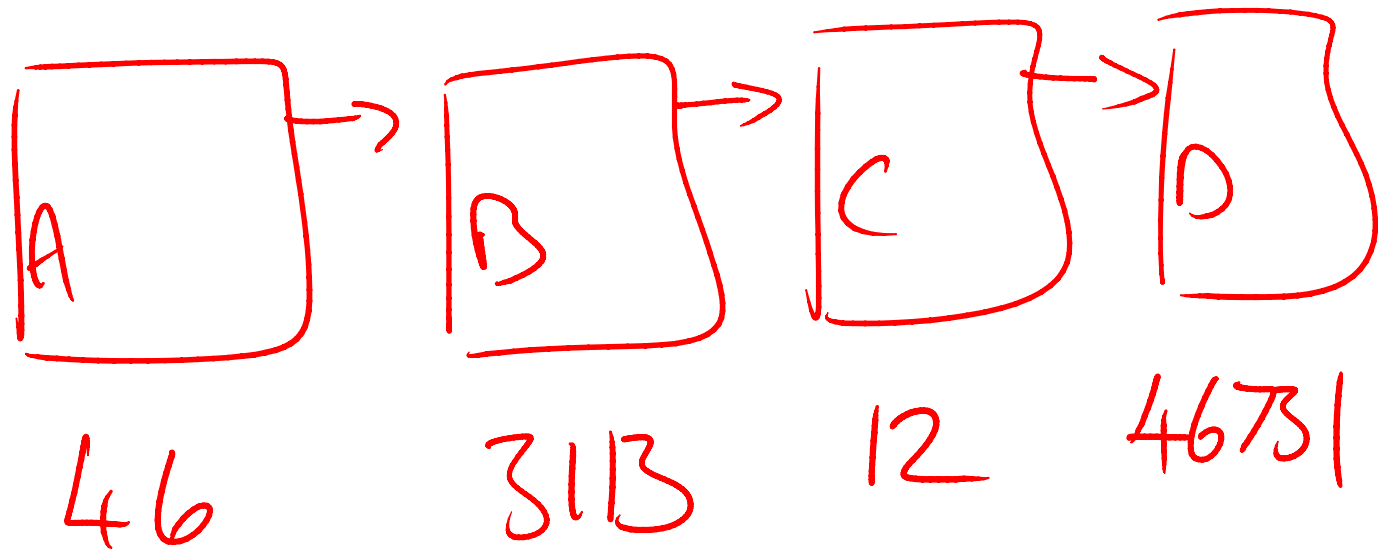


PROBE

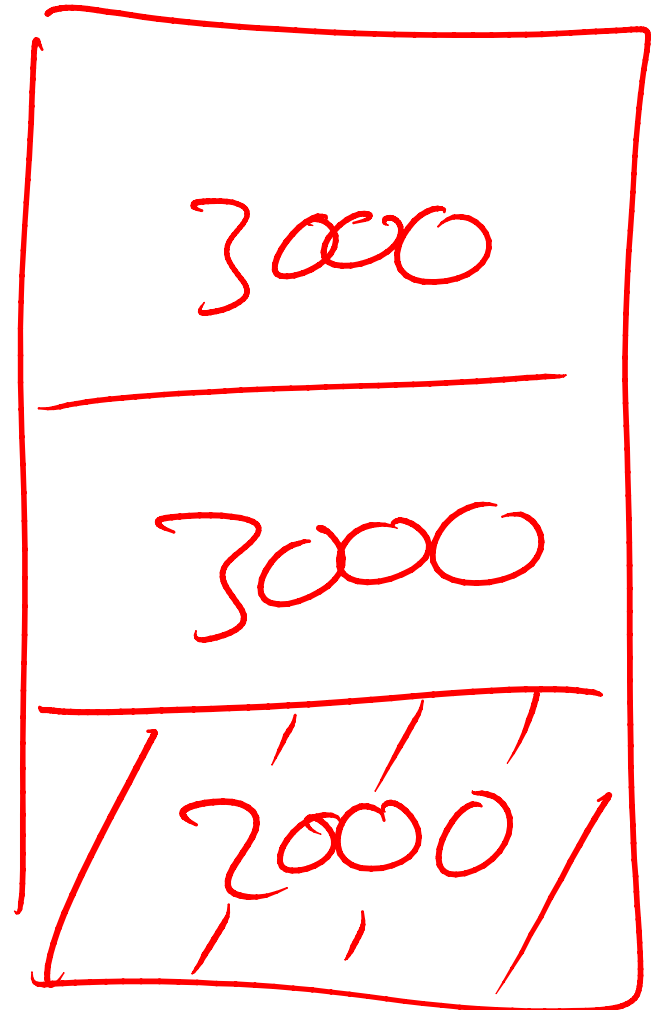
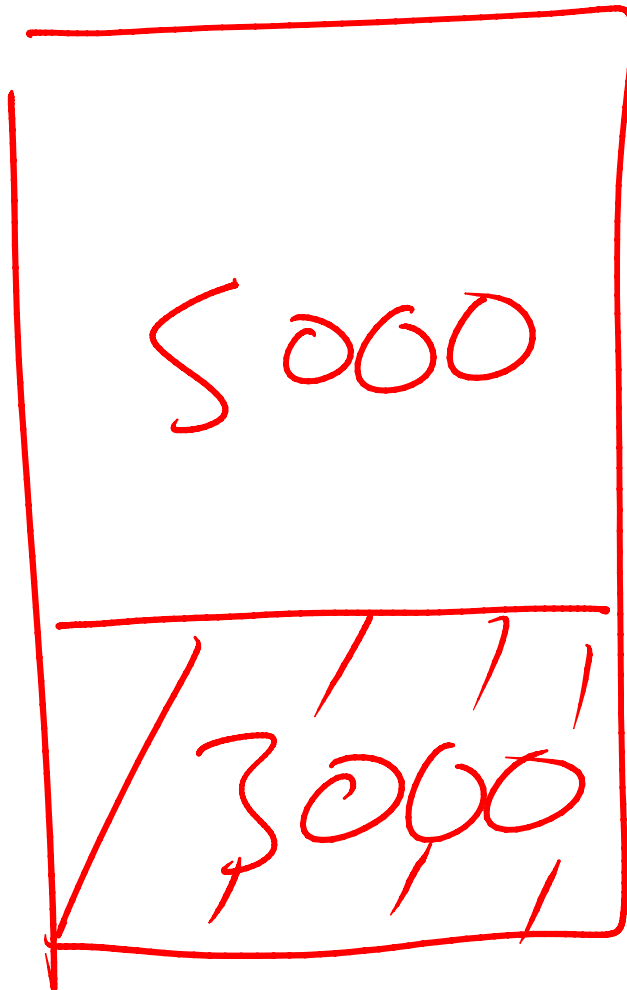
M5S63 Explaining part of the deferred-drop functionality. Before an extent can be deallocated, an X lock is acquired on it and all 8 page locks are probed (instant acquire X lock and release) to make sure nothing else has them locked.



MSS64 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.

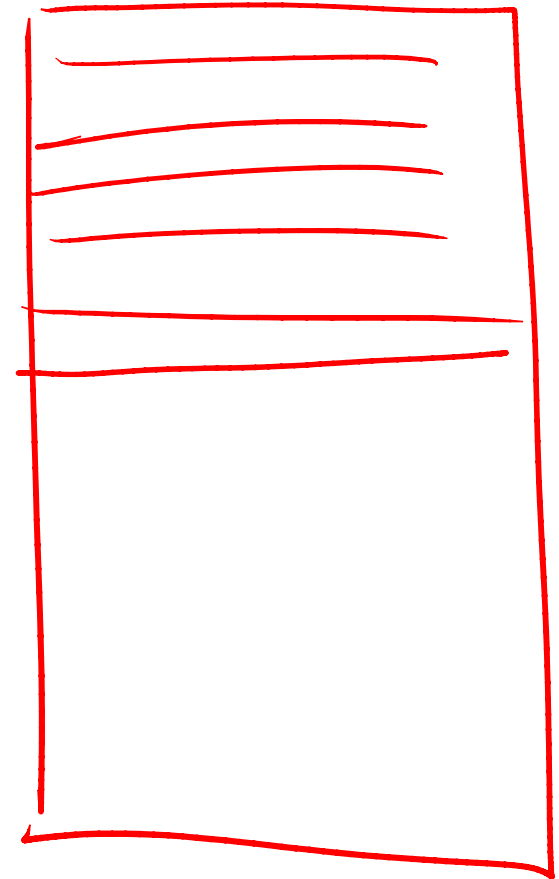
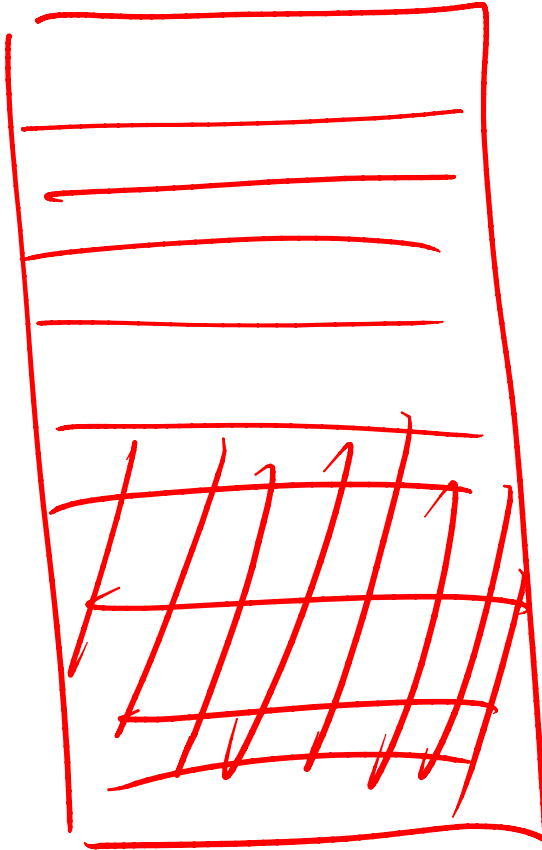


M7S11 An example of logical fragmentation. The pages above are in logical (key) order, but their page IDs are all over the place.

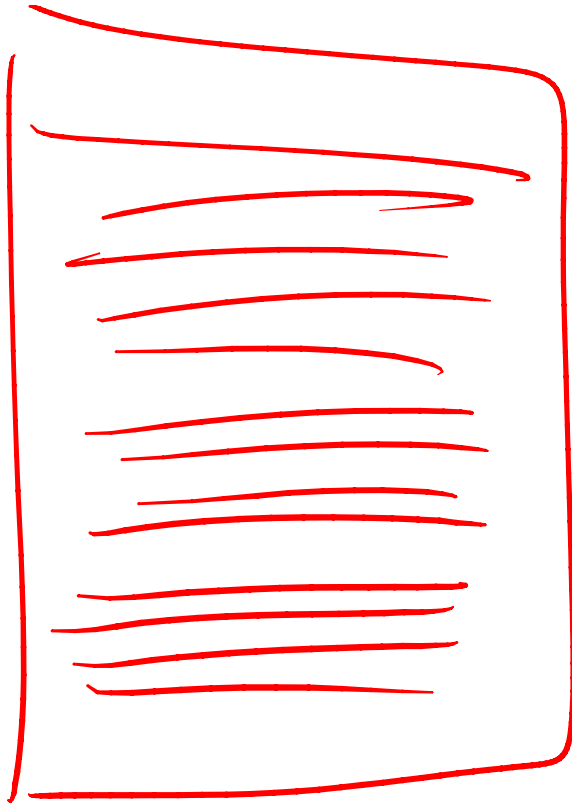


M7S14 Explaining how fixed-size large rows leads to low page density and wasted space.

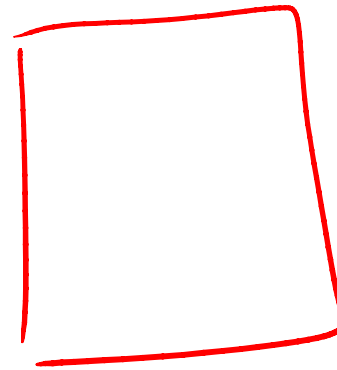
M7S16 When a page split occurs, it's usually 50/50 and creates 50% free space on each page.



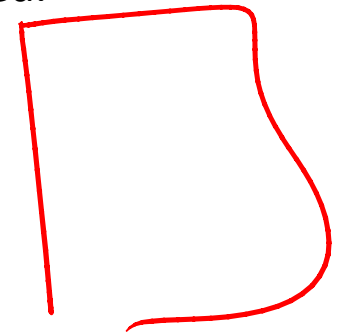
M7S16 Explaining how the Storage Engine is not smart sometimes. Imagine a page full of 20-byte rows. Inserting an 8000-byte row there will lead to 8 splits before enough space is created.



20 byte



4000



2000

8000
INS



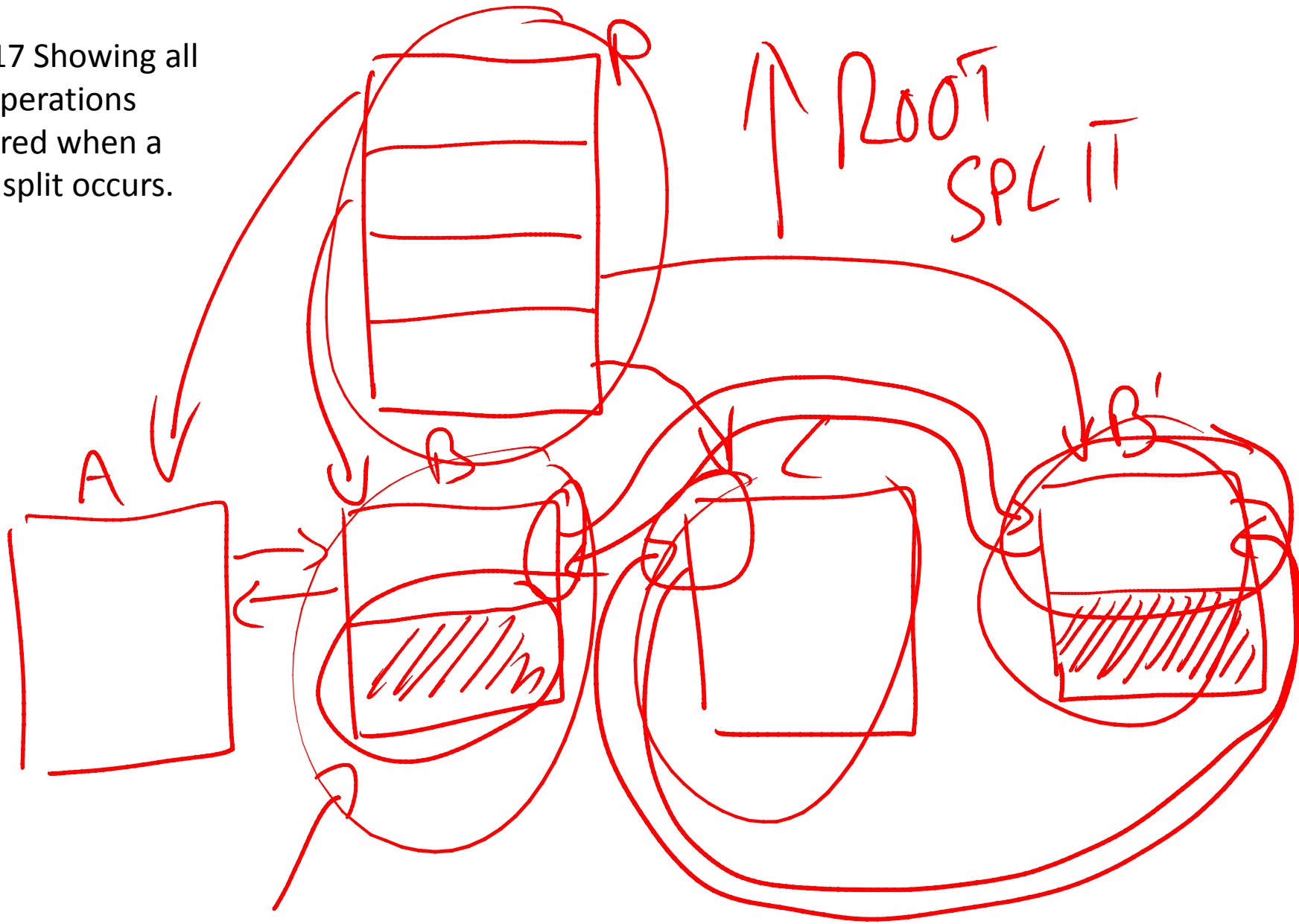
1000

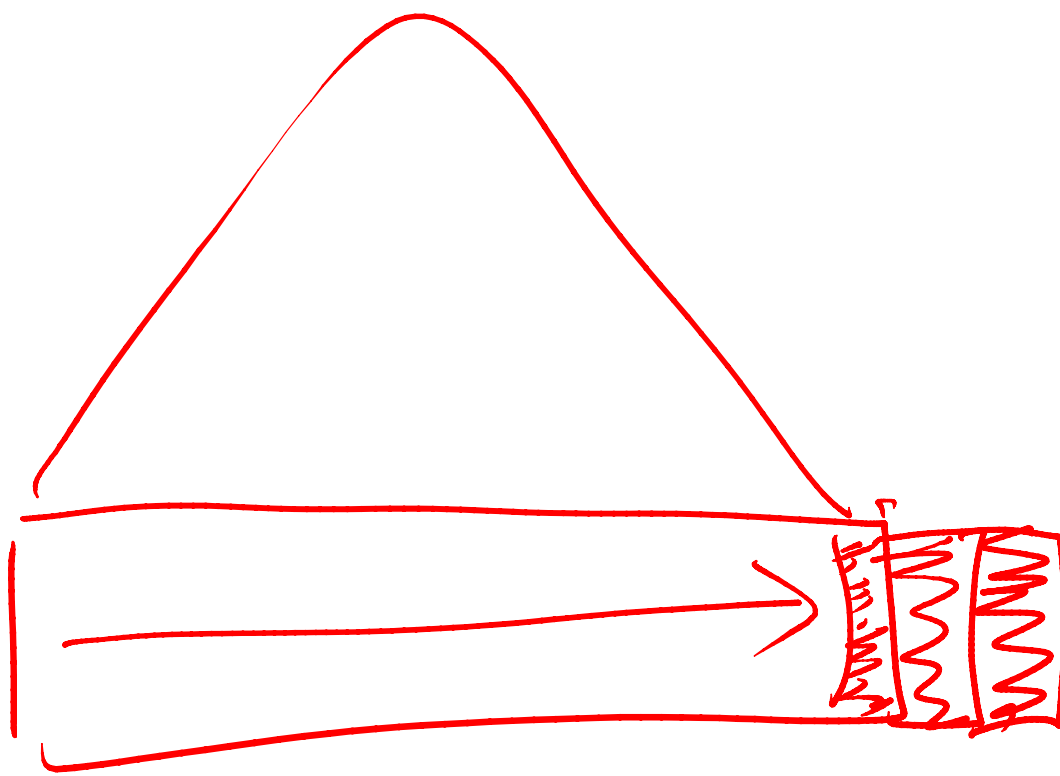


500

250 125
62 31

M7S17 Showing all
the operations
required when a
page split occurs.





LOP_DELETE_SPLIT

M7S19 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.

	NO SPLIT	SPLIT	
1000	1256	7572	x5
100	356	6764	x19
10	268	11164	x45

M7S23 The numbers from the page split logging cost demo.

COMMENTS

PAUL	JON	ERIN	ELENA
MEMBER_ID			K

M7S24 The MySpace story. The 'K' is the tiny portion of comments on Kimberly's page because she wasn't very popular back then...



ITB



M7S33 Explaining staggered index maintenance with database mirroring. See <http://bit.ly/IS04W5> for more explanation