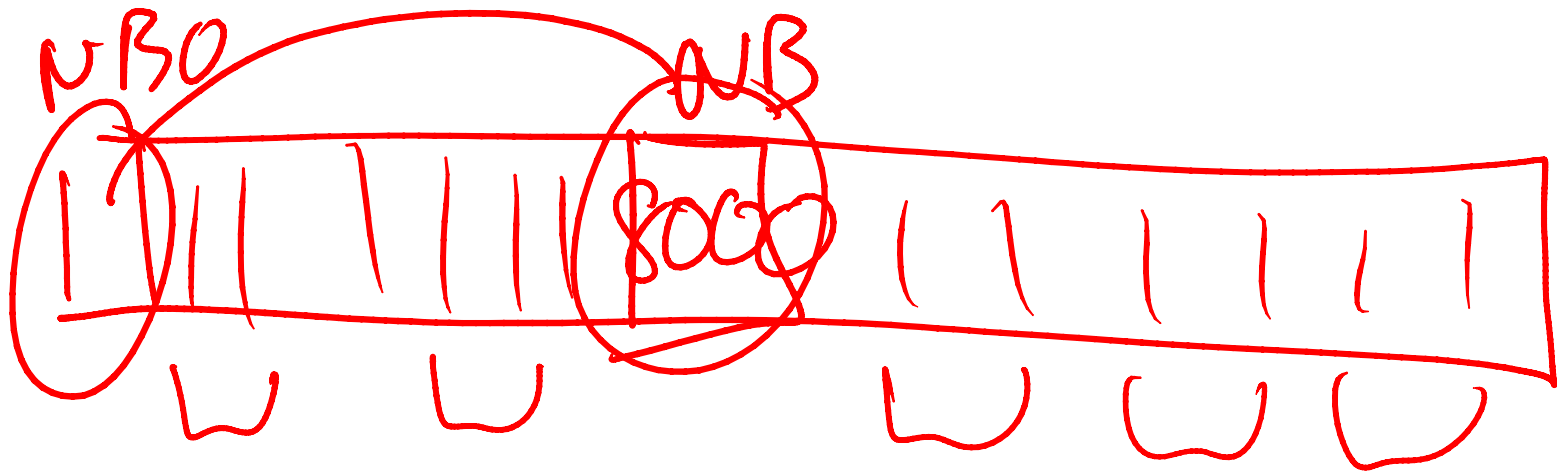
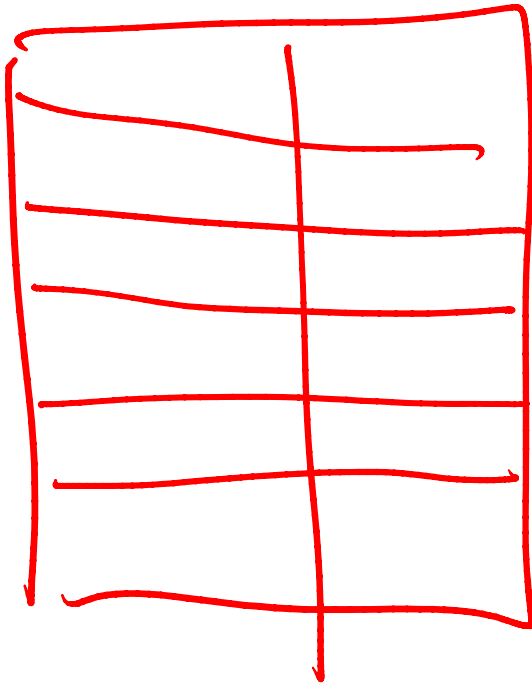
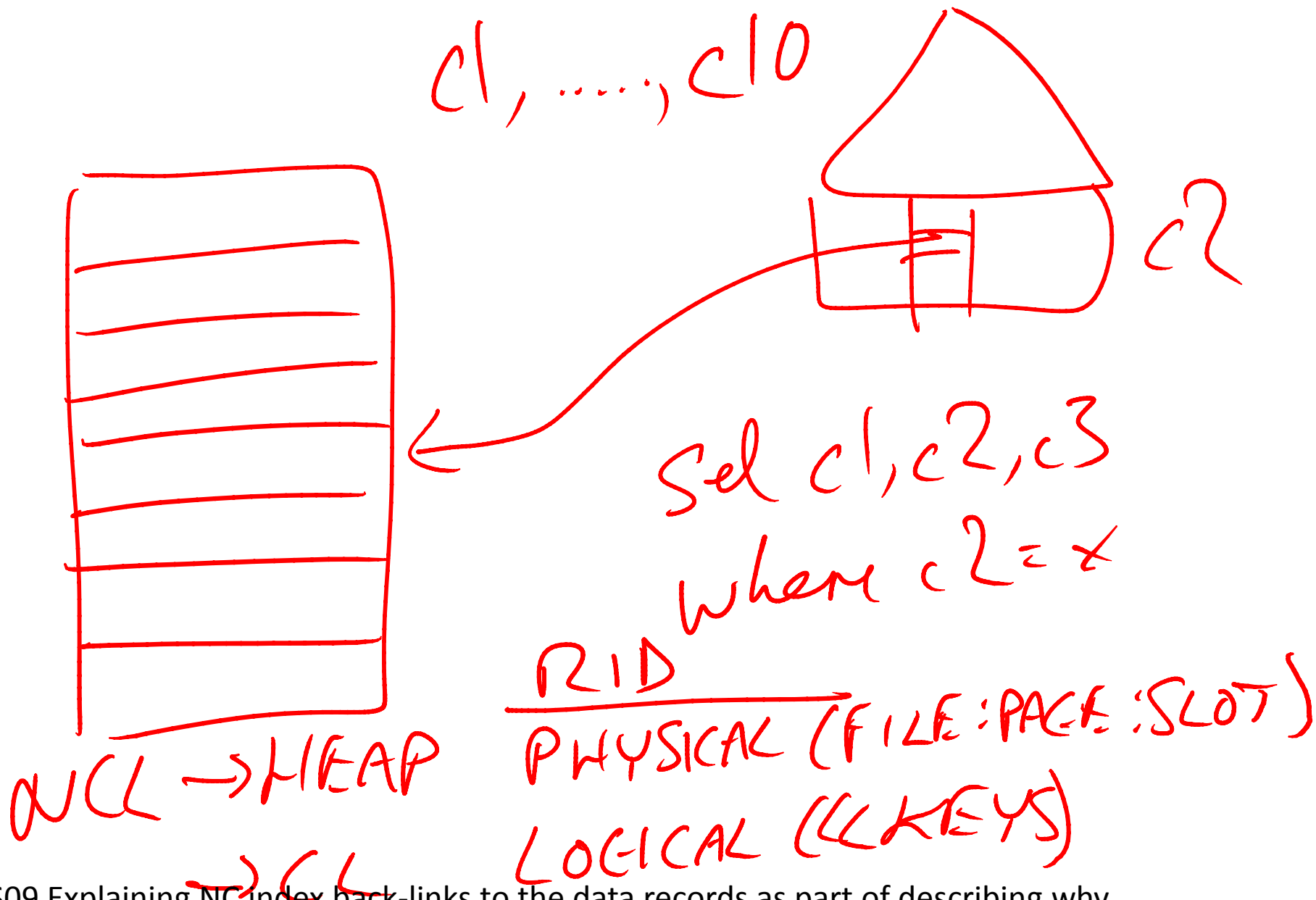




CACHE LINES
64b / 128b / 256b

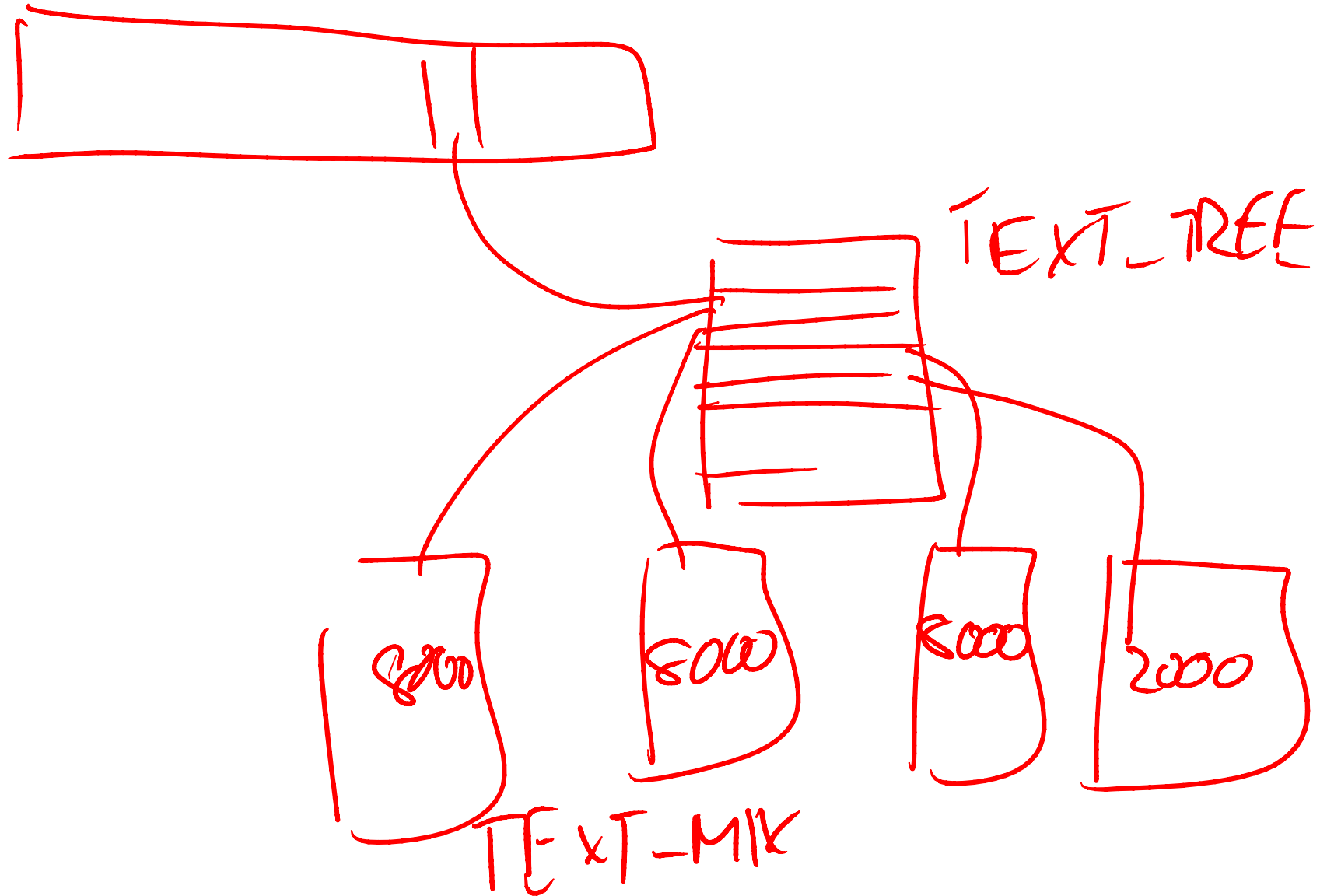


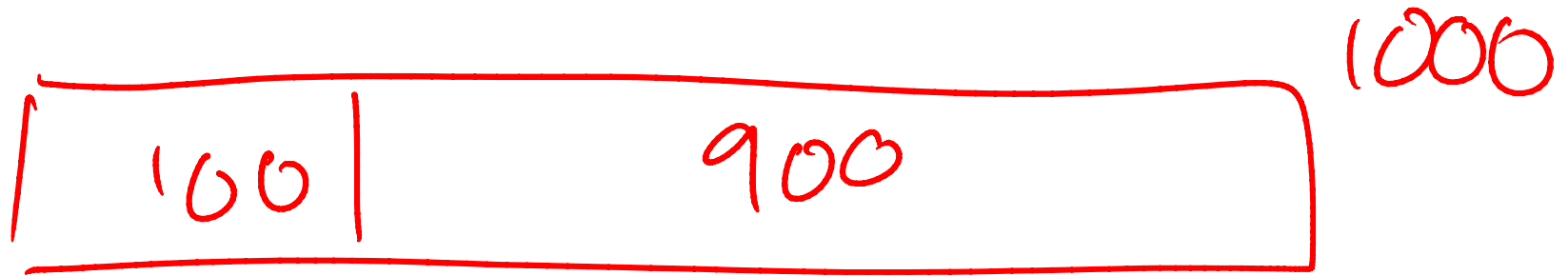
M1S06 Discussing
why the NULL
bitmap is a perf
optimization by
avoiding having to
read record parts
into the CPU's L3
cache



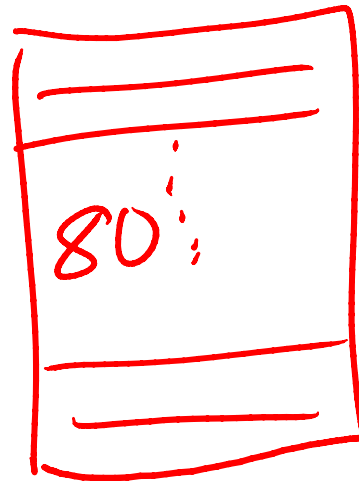
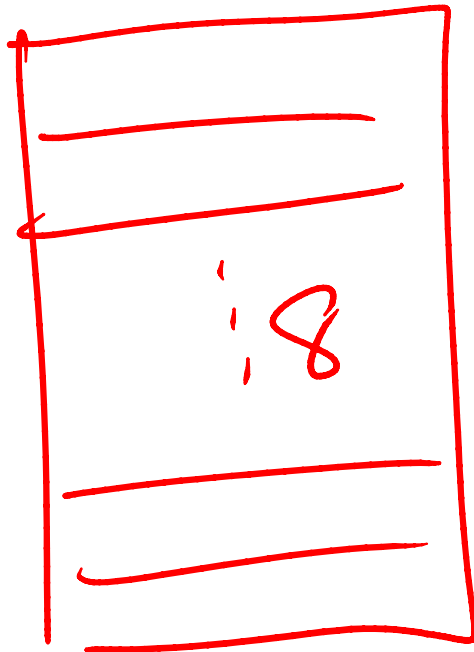
M1S09 Explaining NC index back-links to the data records as part of describing why forwarding/forwarded records exist. Without them, moving a heap row would entail changing all NC index records with a back-link to that row.

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





95% - 100



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.

SPARSE



M1S12: Question on sparse columns storage. Non-null sparse columns take 4-bytes of extra space on top of their data size. See

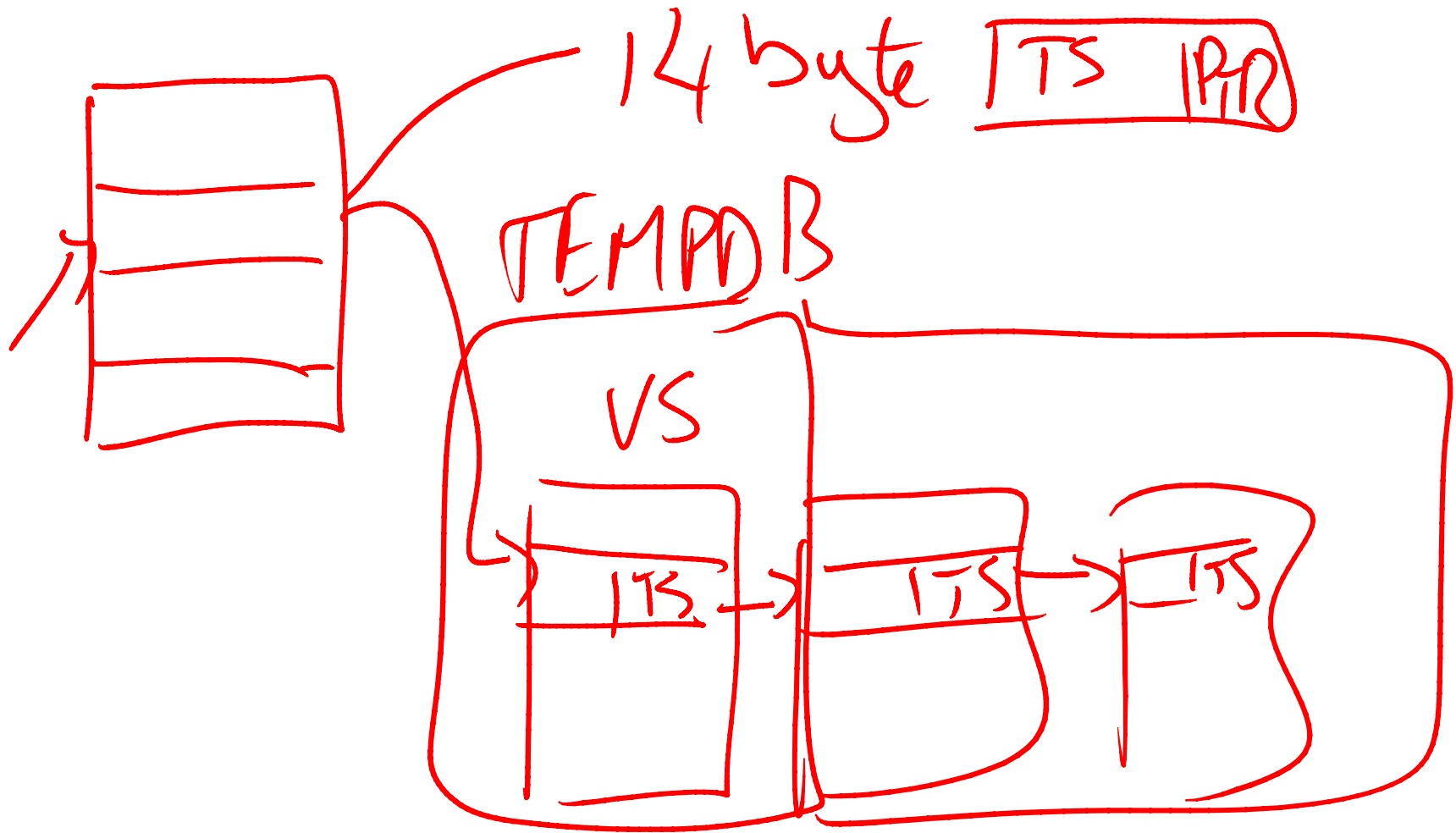
<http://www.sqlskills.com/blogs/paul/category/sparse-columns/>

Row OVERFLOW

VC(5000),

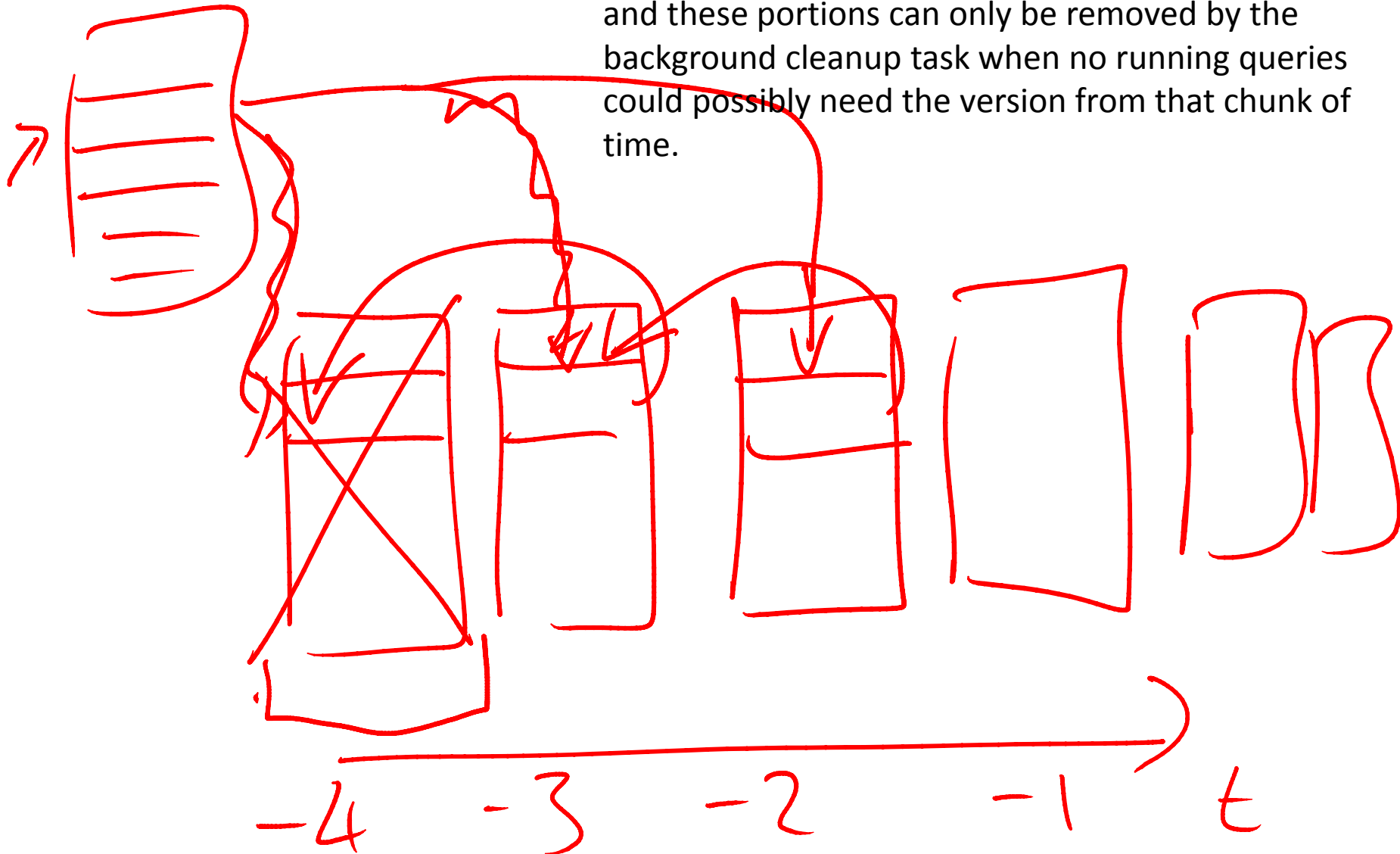
VC(5000)

M1S12: question about row overflow. In SQL Server 2000, a table with two varchar (5000)s could be defined, but you couldn't populate both columns to 5,000 bytes. You can from 2005 onwards – and one of them will be pushed off row and stored as if it's an off-row LOB column.



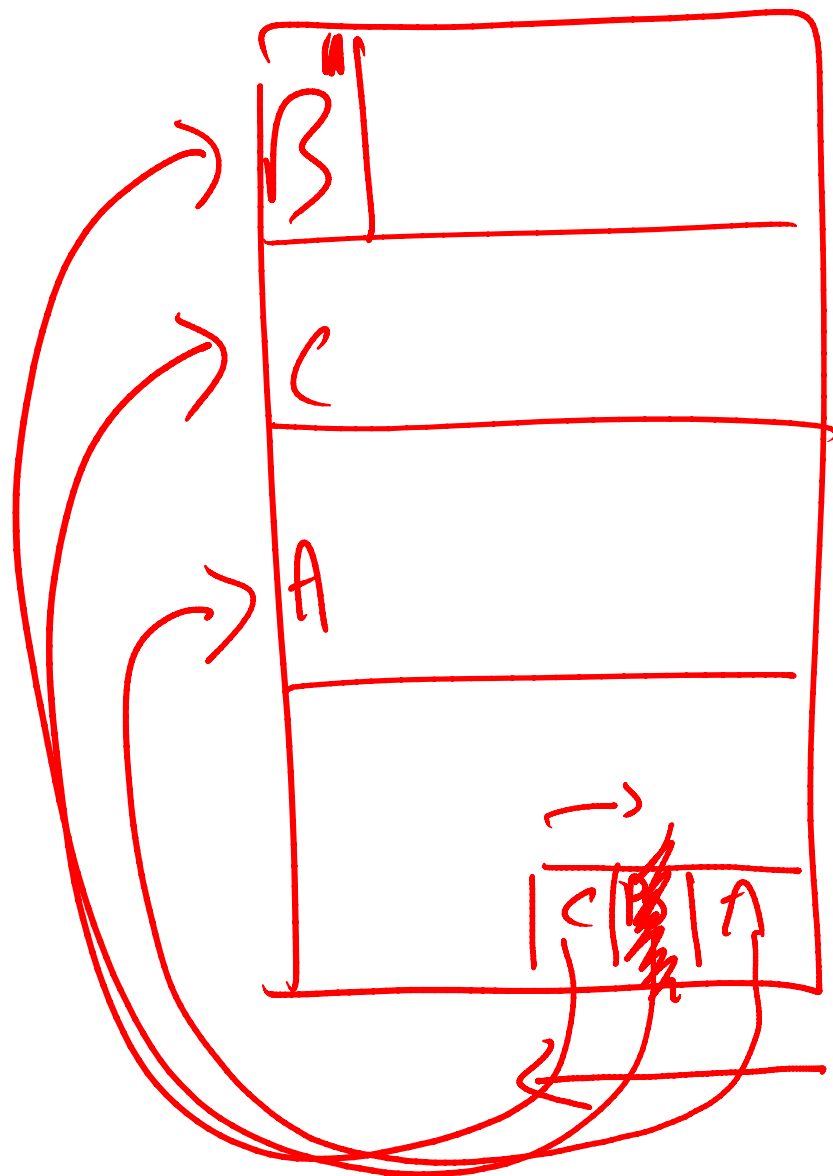
M1S13 When an update occurs to a record and snapshot isolation is enabled, the pre-change record is copied into the version store in tempdb. The changed record has a 14-byte tag on the end with the timestamp and pointer to previous record version. A chain of previous versions is possible.

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.



	SnapShot ISOLATION	PRIM tempdb	SEC tempdb
PRIM	Y	Y	-
	N	N	N
SEC	Y	N	Y
(readable)	N	N	Y

M1S13: Table showing when you'll see version store use when using an availability group.



LOP EXPUNCE -
CHOST

SLOT ARRAY

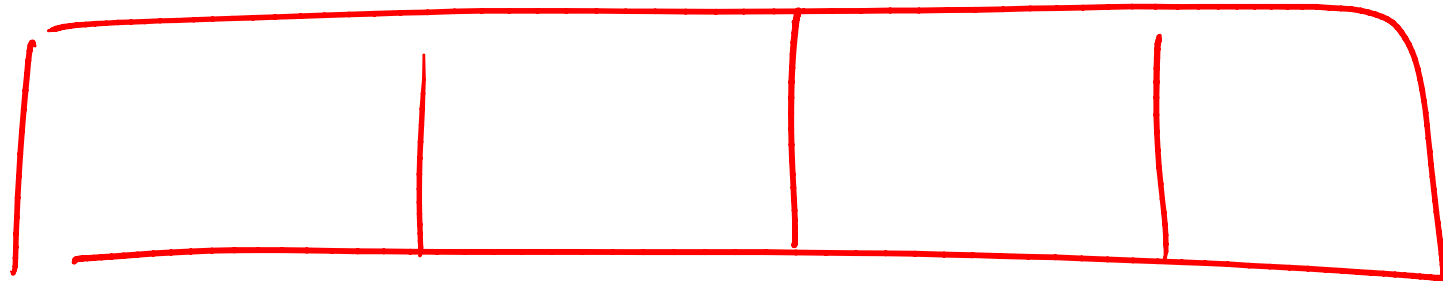
M1S14 When a record is deleted, it's marked as a ghost record. The ghost cleanup task does not physically delete the records – it removes the slot array entry that points to the record on the page. I.e. the record is unmarked as being a record and the area on the page is now free space – that just happens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

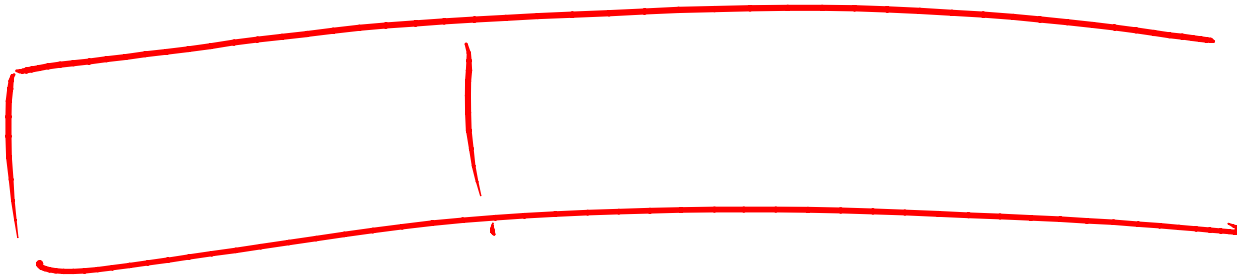


0 — 7 8 — 15 16 — 23



M1S21 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'.



0 - FH

1 - PFS

2 - GAM

3 - SGAM

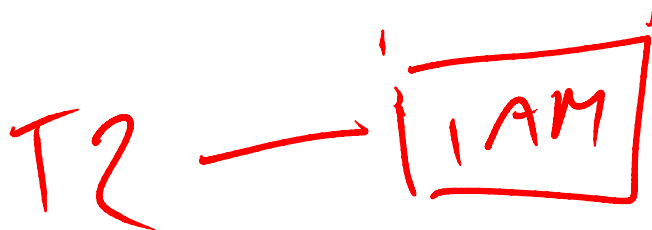
4 - UNUSED

5 - UNUSED

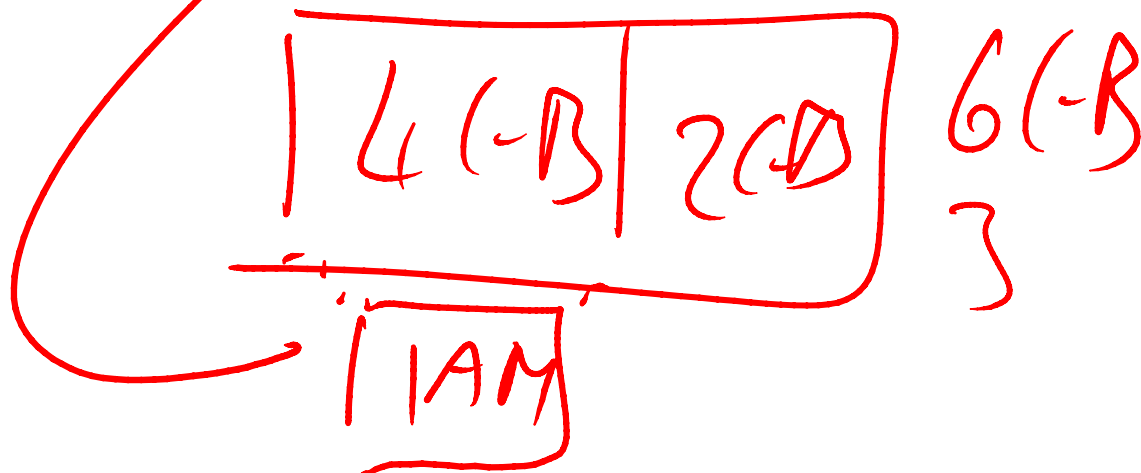
6 - DIFF

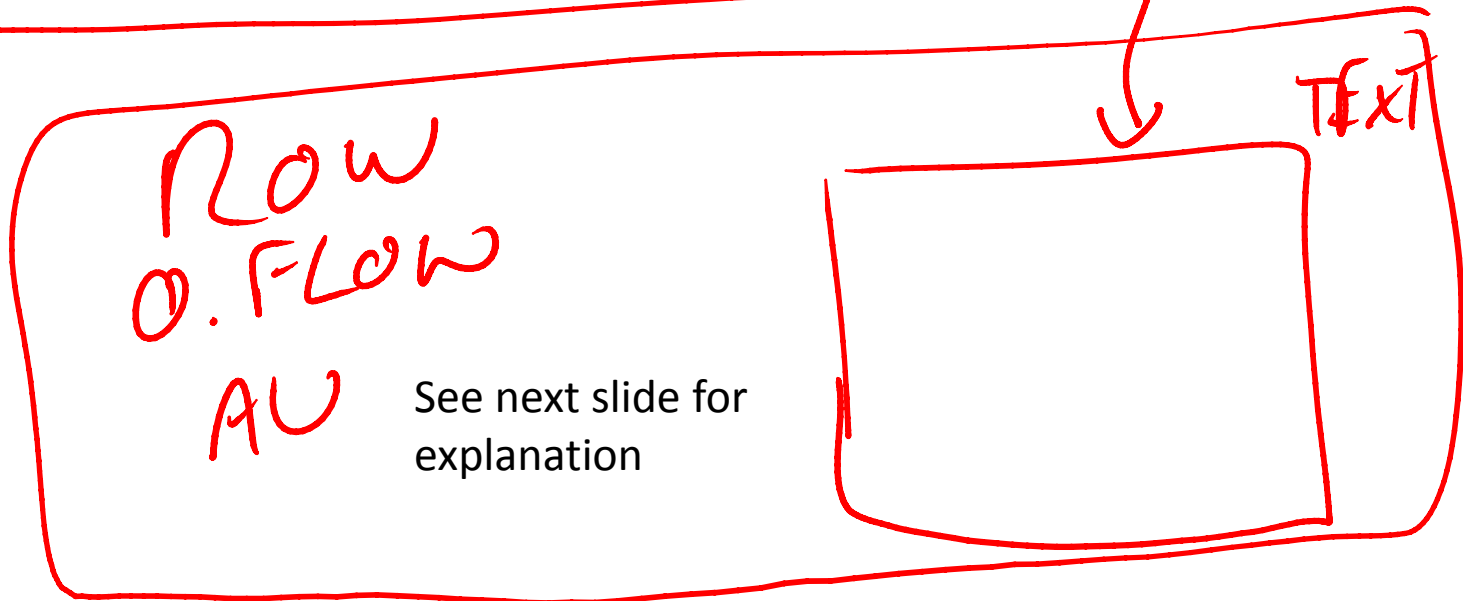
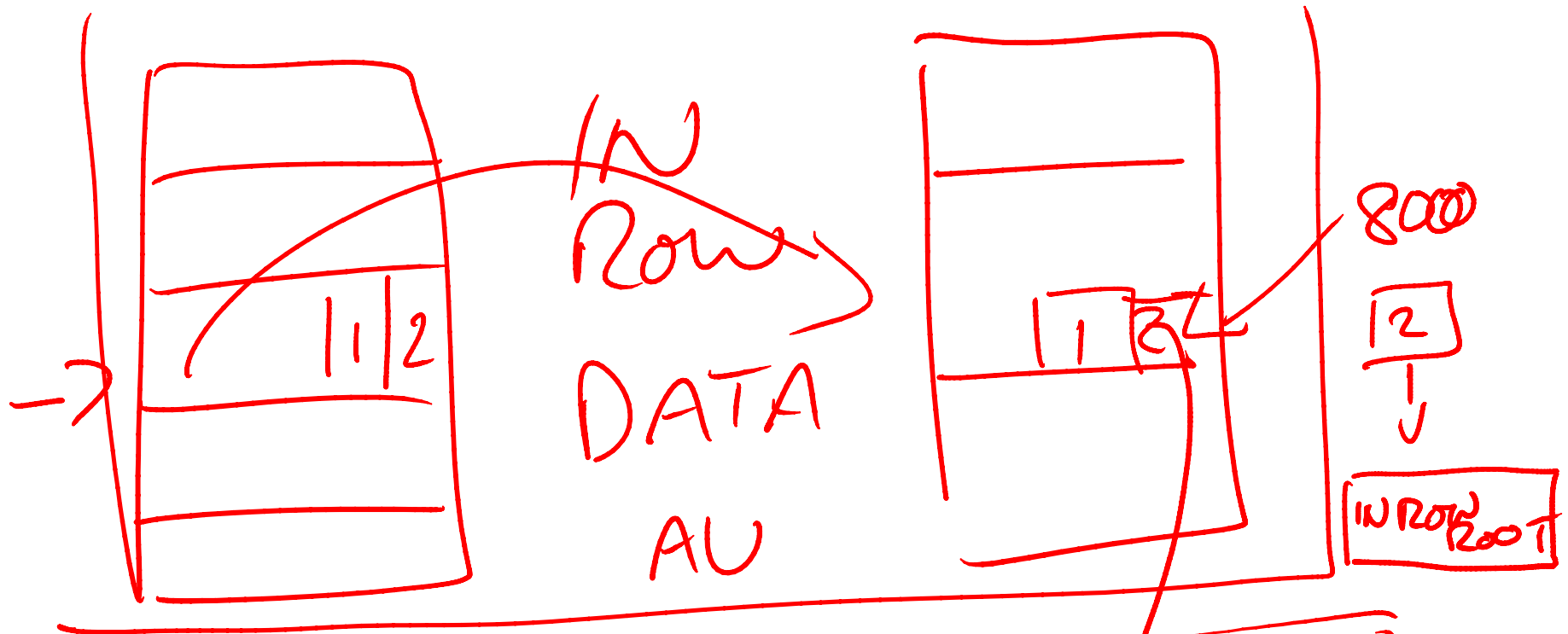
7 - ML

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



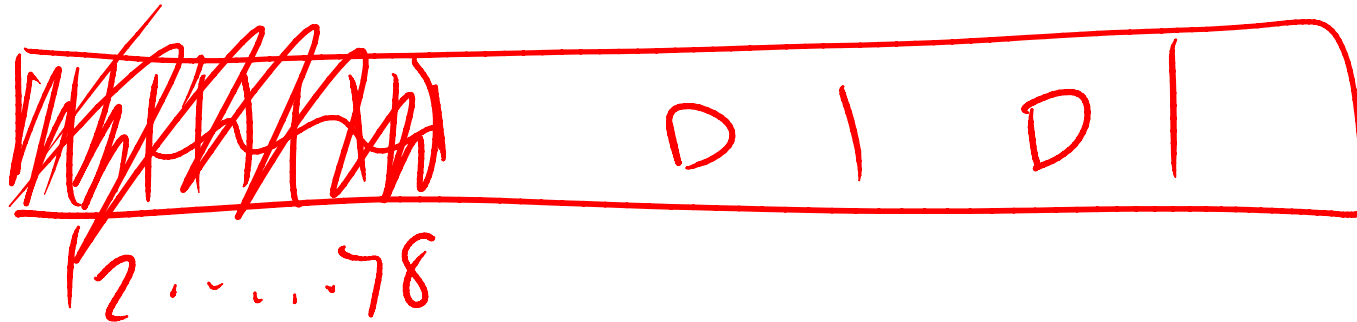
M1S34 Example of IAM chains.
Allocations from each 4GB
chunk of a data file require an
IAM page to track the
allocation. Multiple IAM pages
make an IAM chain.



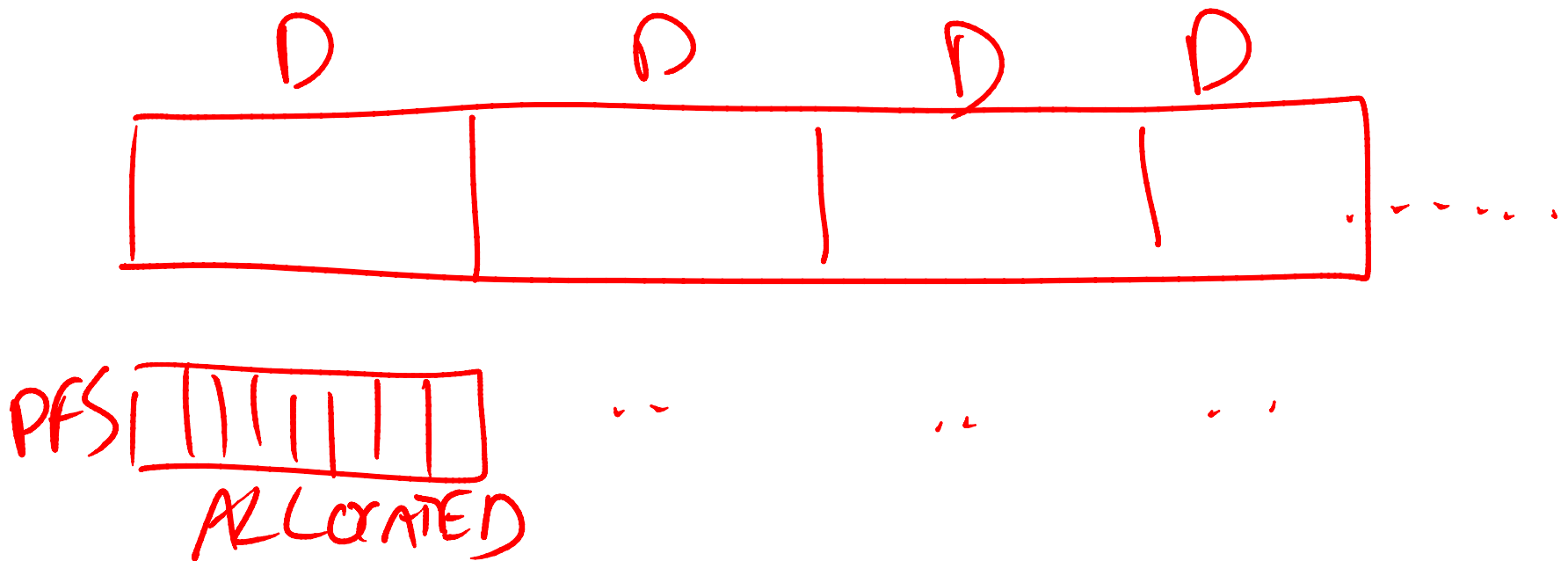


- The 1 and 2 on the left are columns and column 2 is a varchar (8000)
- If column 2 is updated to be, say, 8,000 bytes and no longer fits, it will be pushed off-row into a text page in the ROW_OVERFLOW allocation unit
- In its place in the row, is a 24-byte pointer called an in-row blob root

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.

SALES 1 TB

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.

TP4 2015

2014

2013

2012

SALES 1 TB

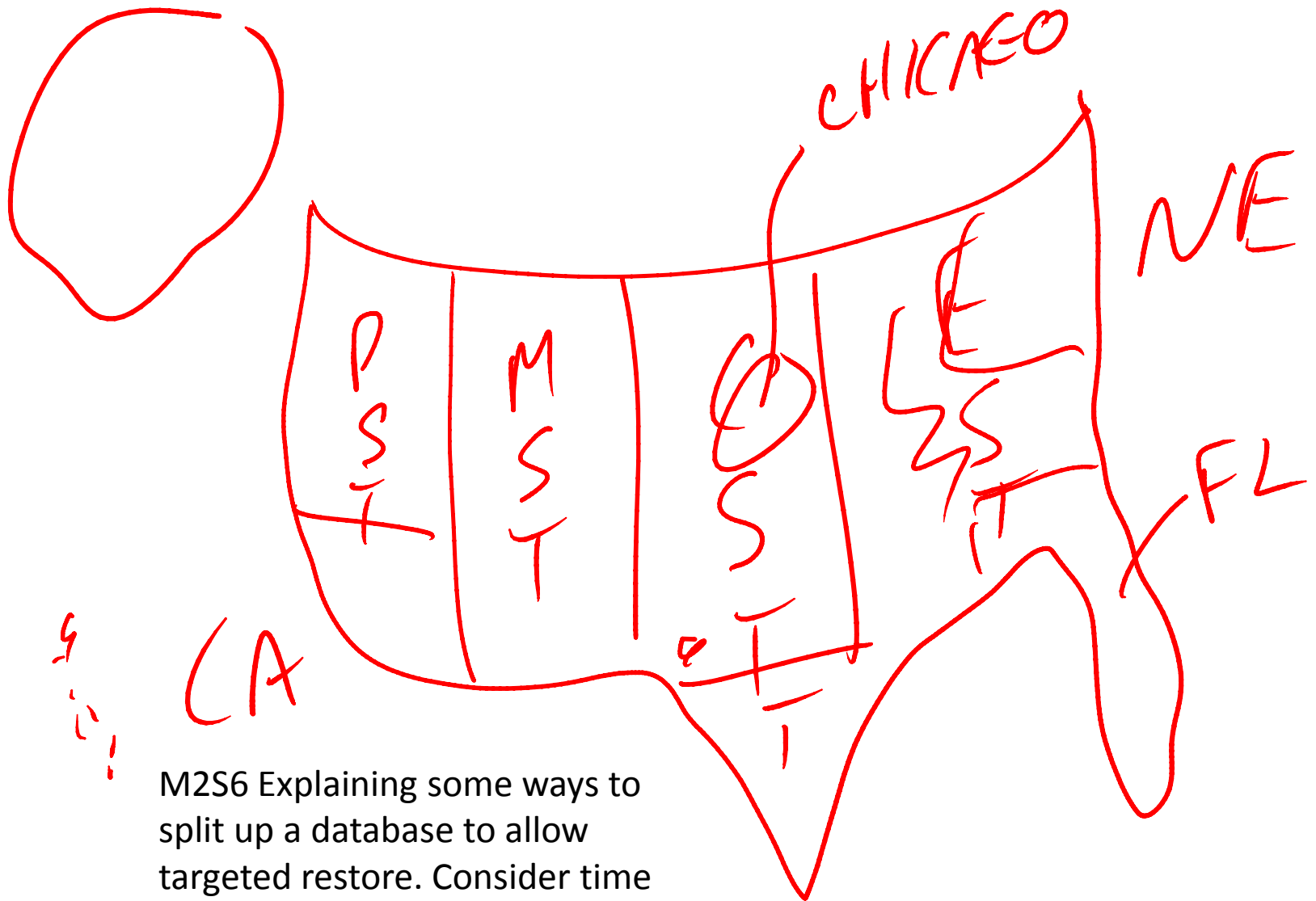
M2S6 Splitting a large database up into multiple filegroups can enable you to do small, targeted restores or even a partial restore from scratch – in the event that the database is damaged or destroyed.

TP4 2015

2014

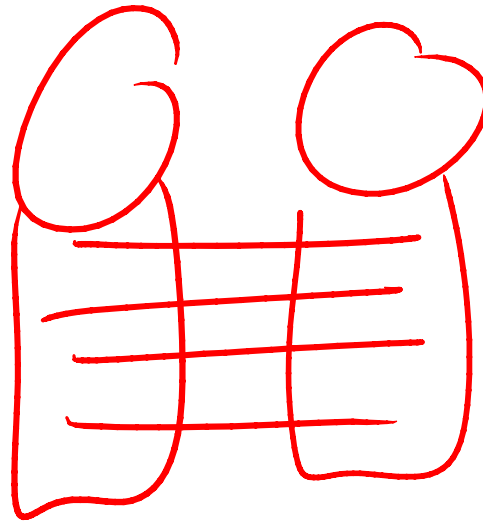
2013

2012



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

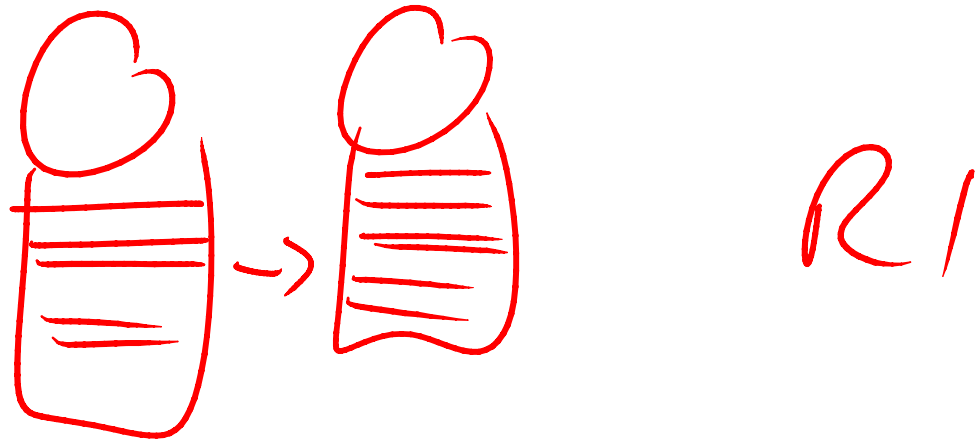
STRIPING



R0

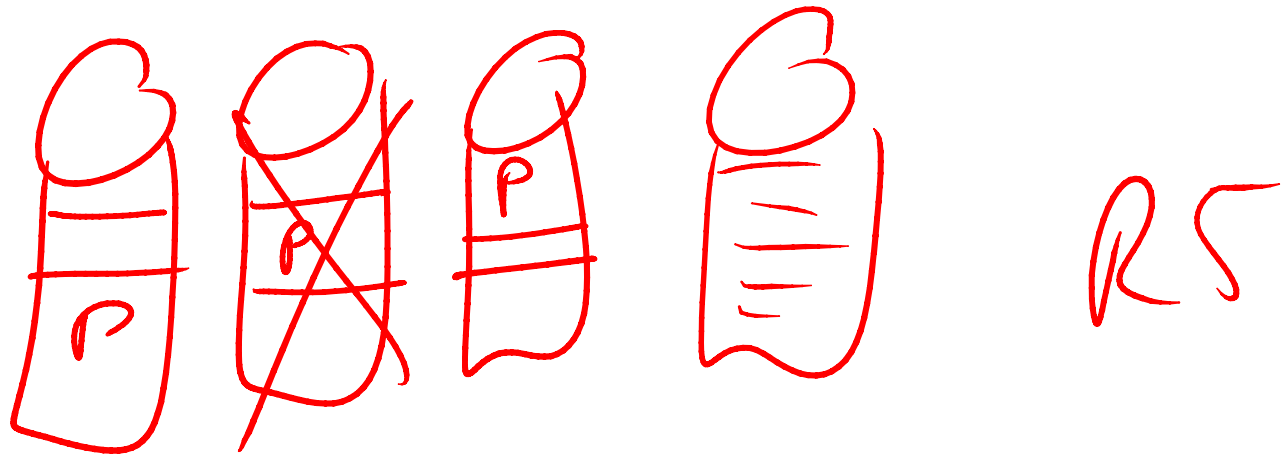
http://en.wikipedia.org/wiki/Standard_RAID_levels

MIRRORING



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

NOTATING PARITY

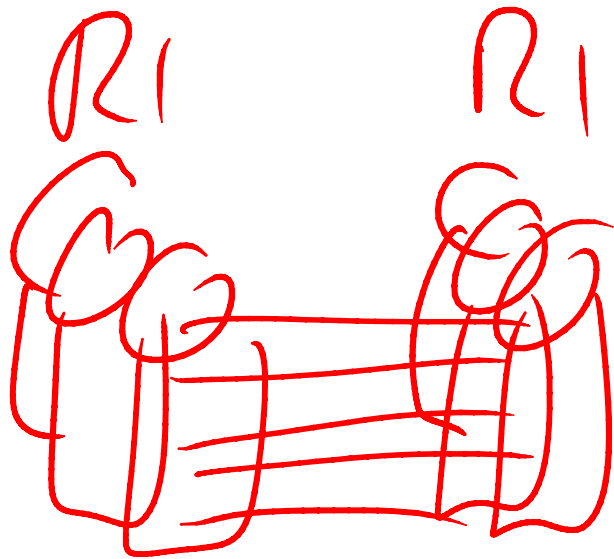


http://en.wikipedia.org/wiki/Standard_RAID_levels

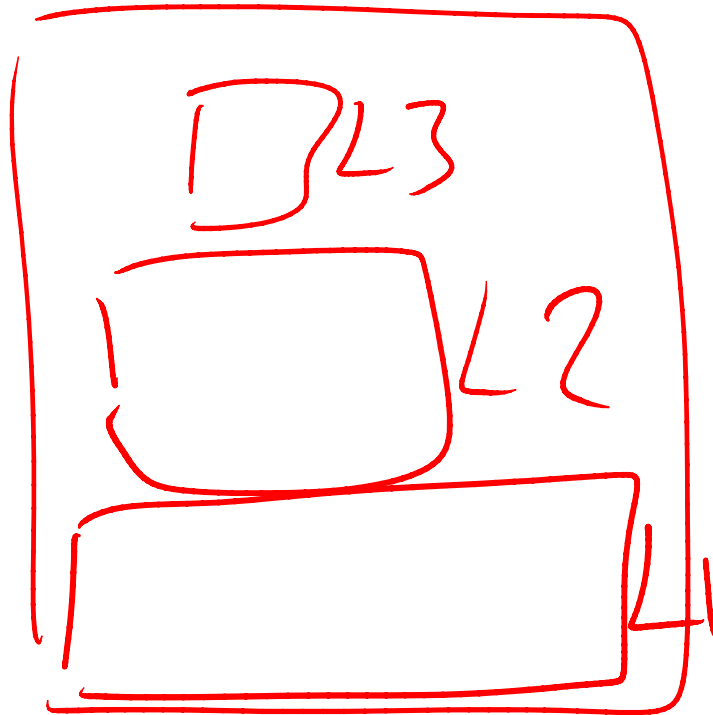
$$R10 = R1 + G$$

= MIRRORING

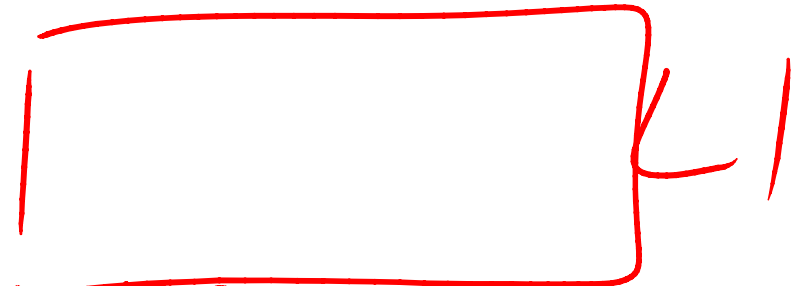
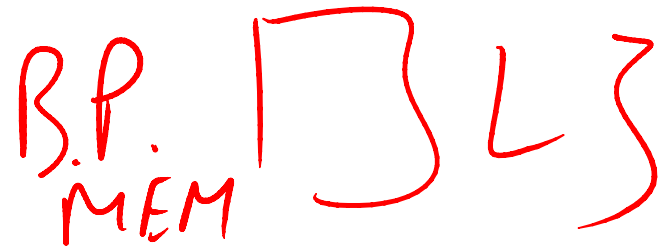
+ STRIPING



CPU

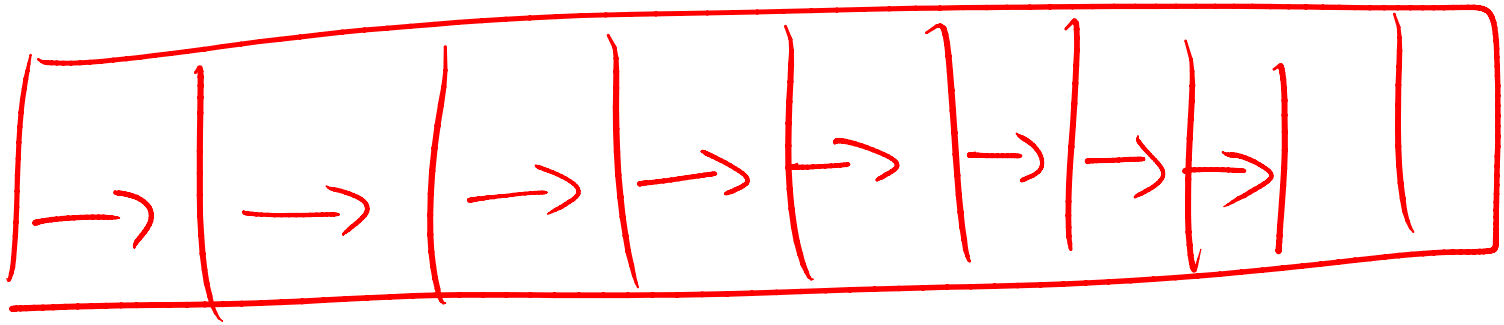


I/O LATENCY



I/O SUBSYSTEM

M2S11 Explaining 2014 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.

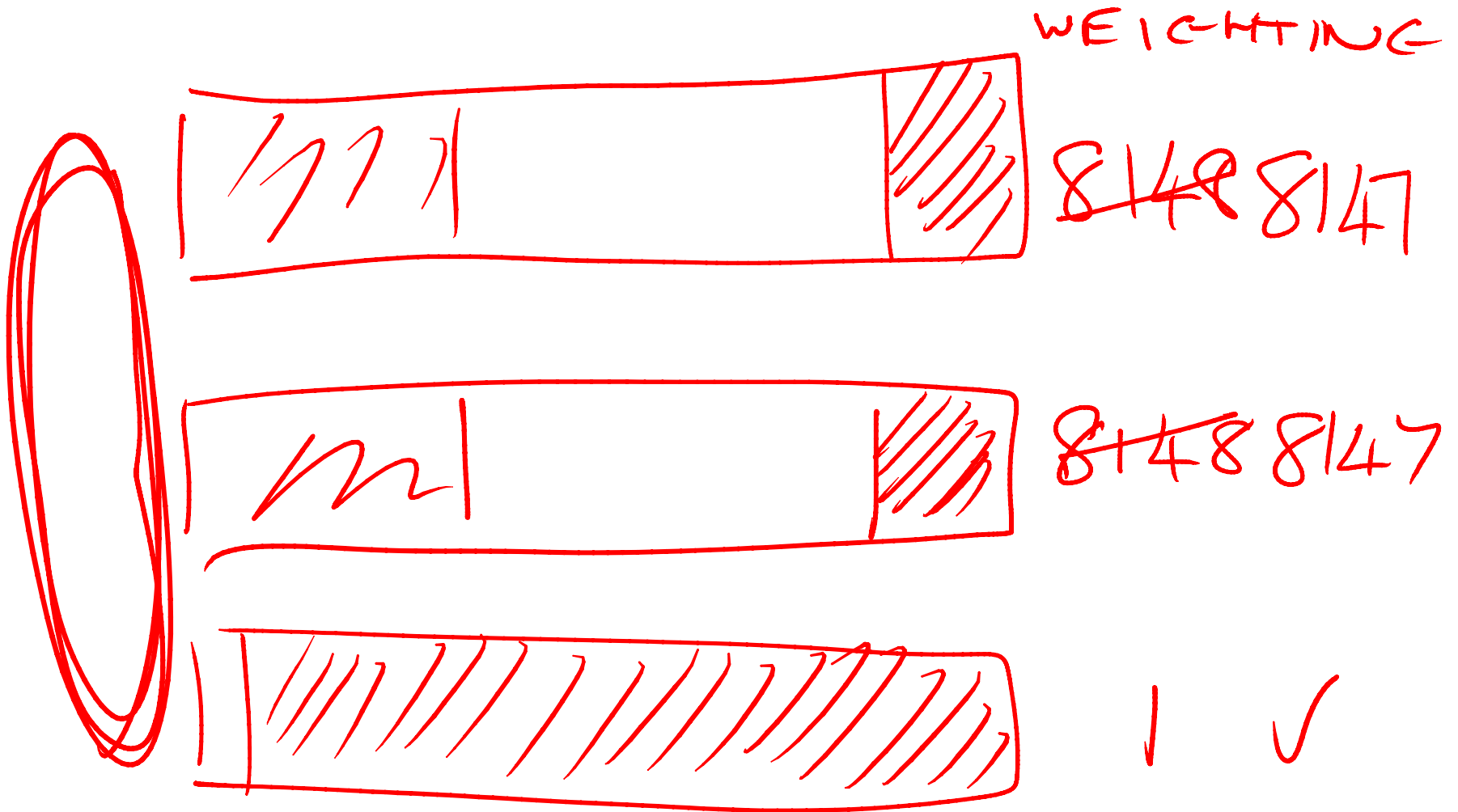


∠:

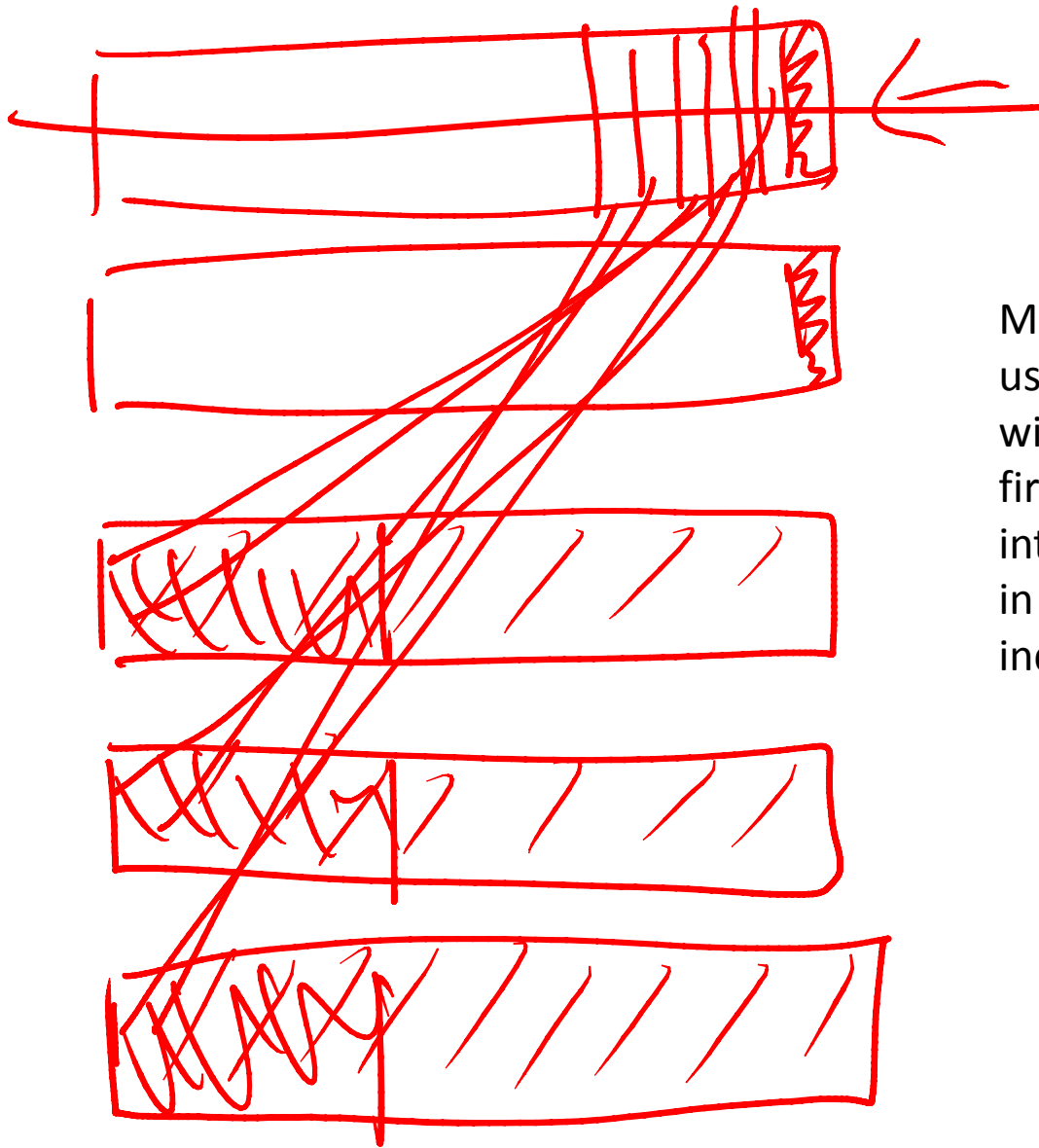
M2S12 If you put all your log files on one volume, its I/O characteristics become random

sys.dm_io_virtual_file_stats

M2: sys.dm_io_virtual_file_stats is the DMV to examine SQL Server's per-file view of I/O latencies.

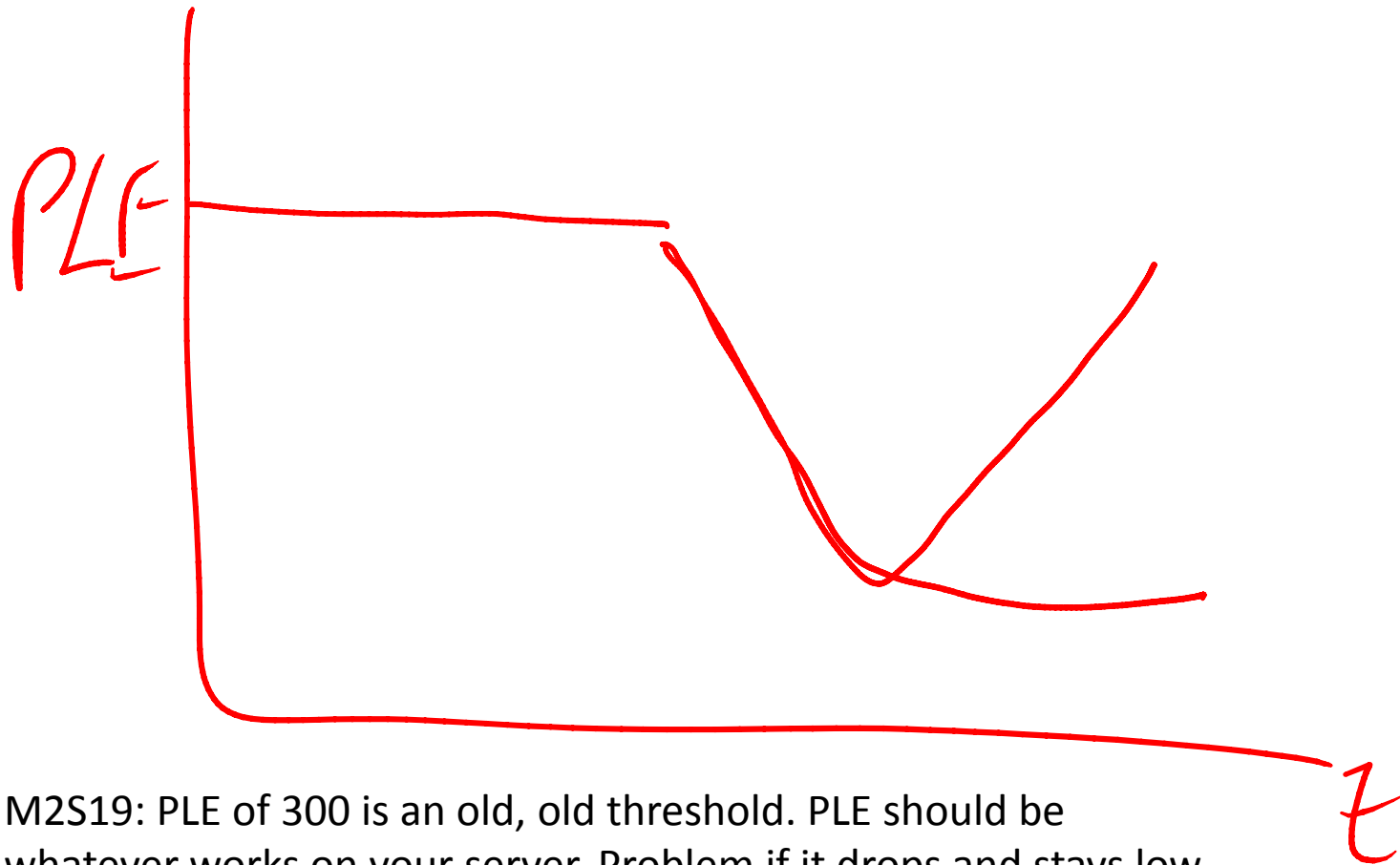


M2S18 Explaining round-robin allocation and proportional fill. Each file has a proportional fill weighting. Allocations happen proportionally more frequently from files with more free space.



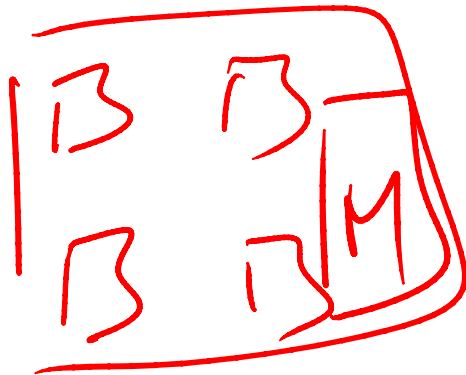
M2S18 Explaining how using DBCC SHRINKFILE with EMPTYFILE on the first file would move pages into the three added files in a filegroup, still causing index fragmentation.

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

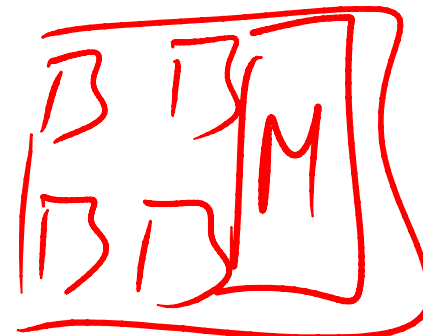


M2S19: PLE of 300 is an old, old threshold. PLE should be whatever works on your server. Problem if it drops and stays low. Normal for it to drop and come back up again after some operations.

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.

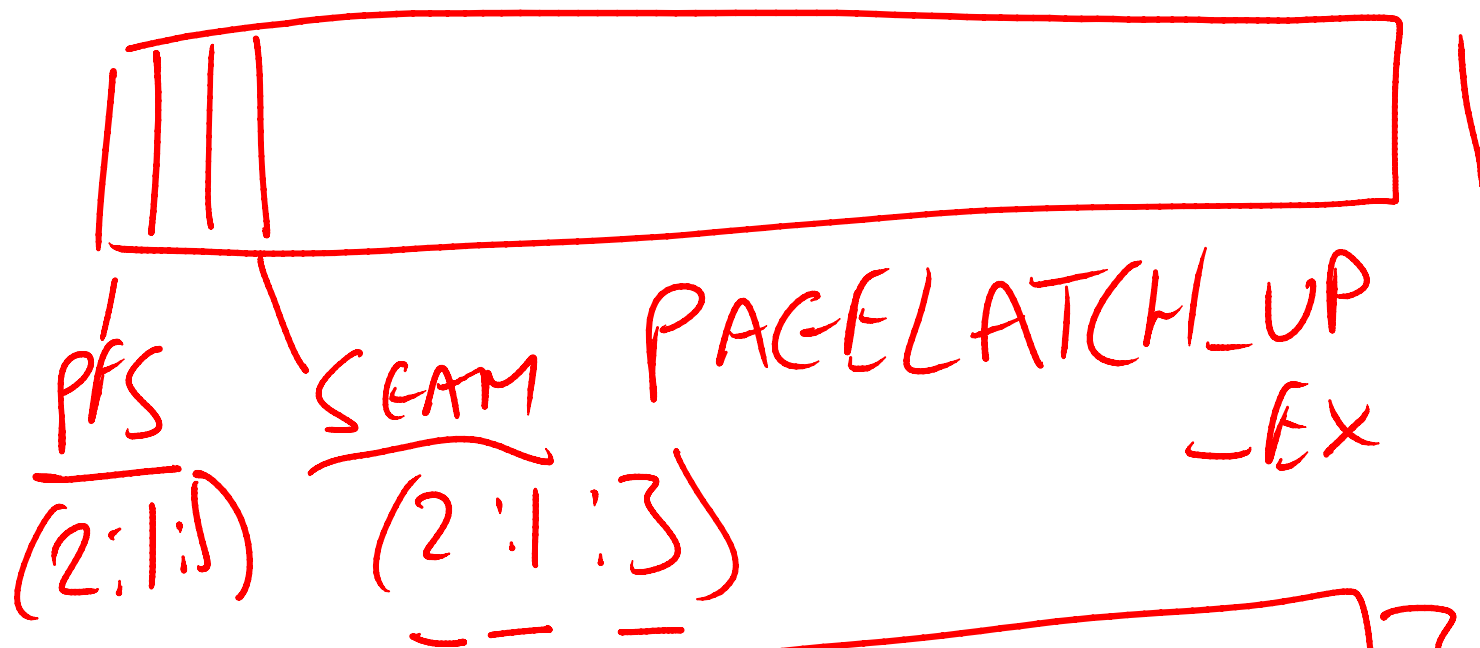
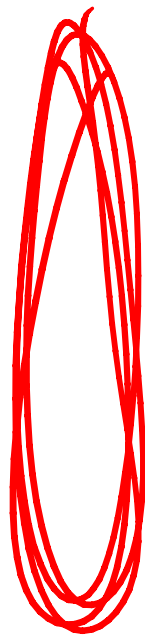


4 NUMA NODES

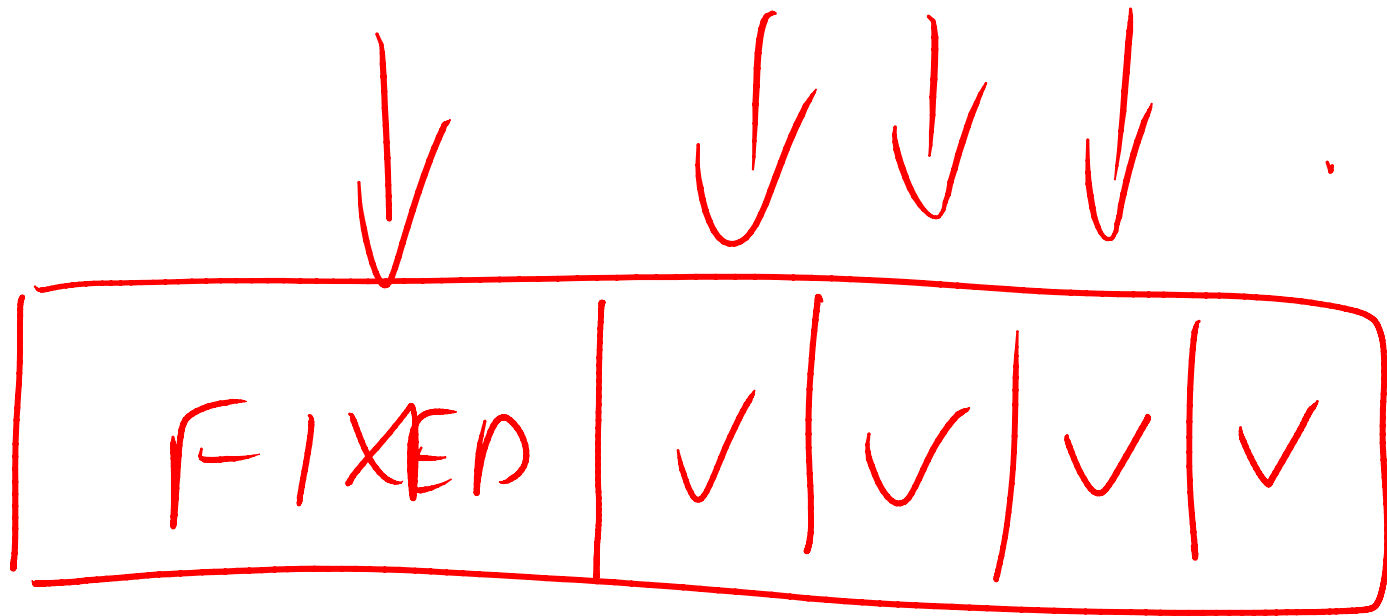
Buffer Mgr: PLE = harmonic mean
of all B.Pool. nodes

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.



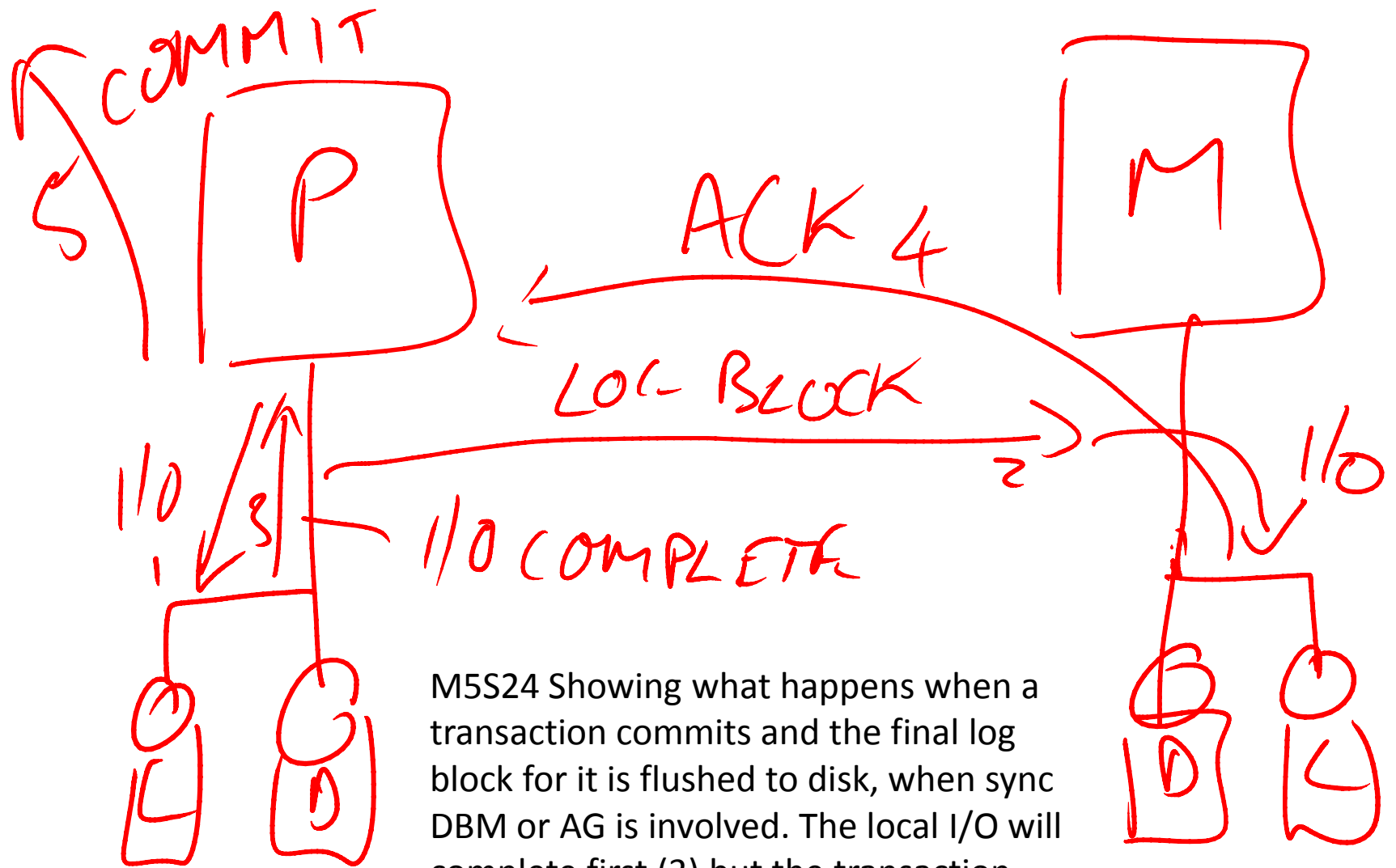
M2S59 Explaining PFS and SGAM contention in tempdb and how adding multiple files spreads the contention over multiple files, thus reducing it.



LOP_MODIFY_ROW

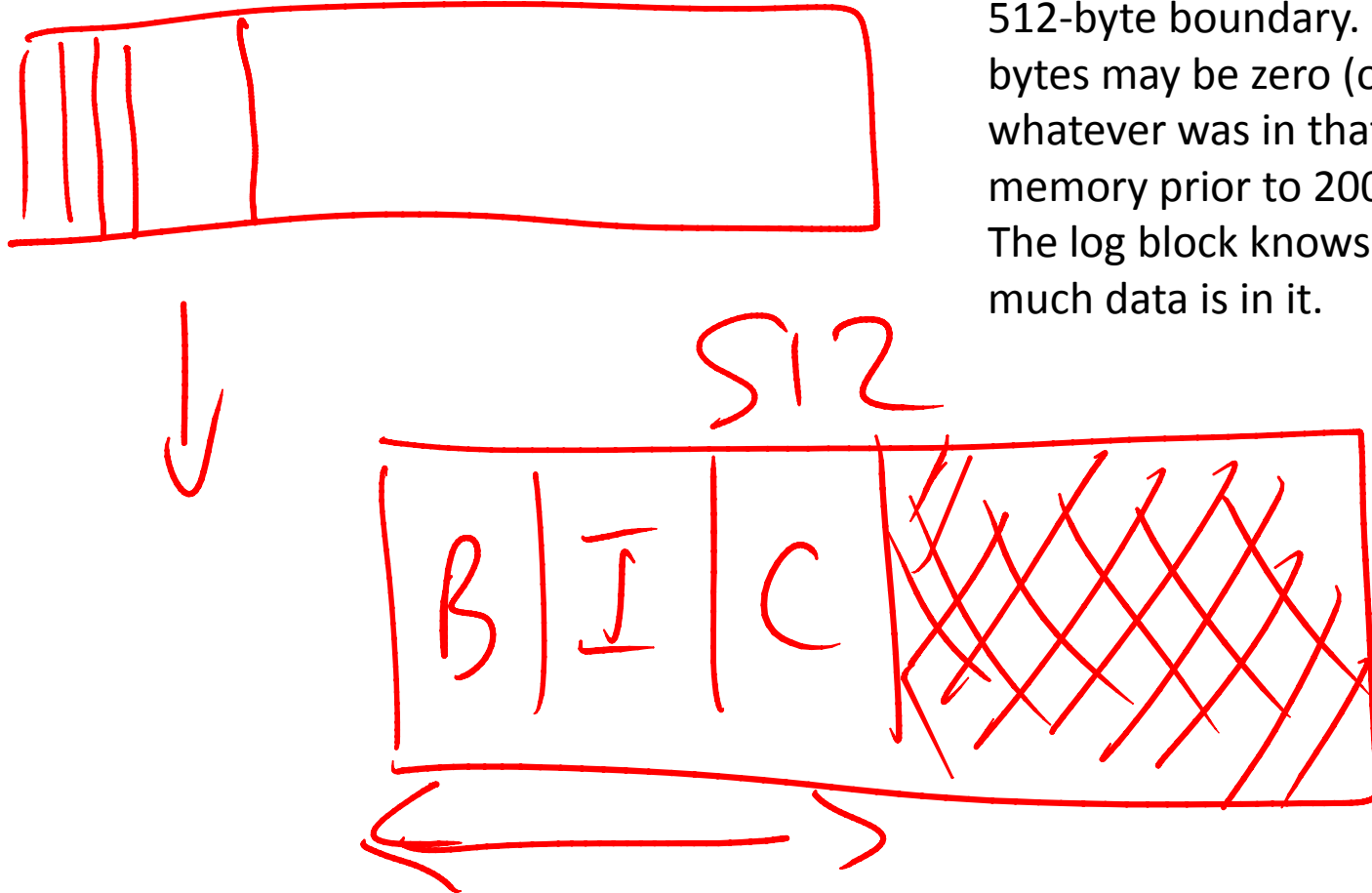
LOP_MODIFY_COLUMN

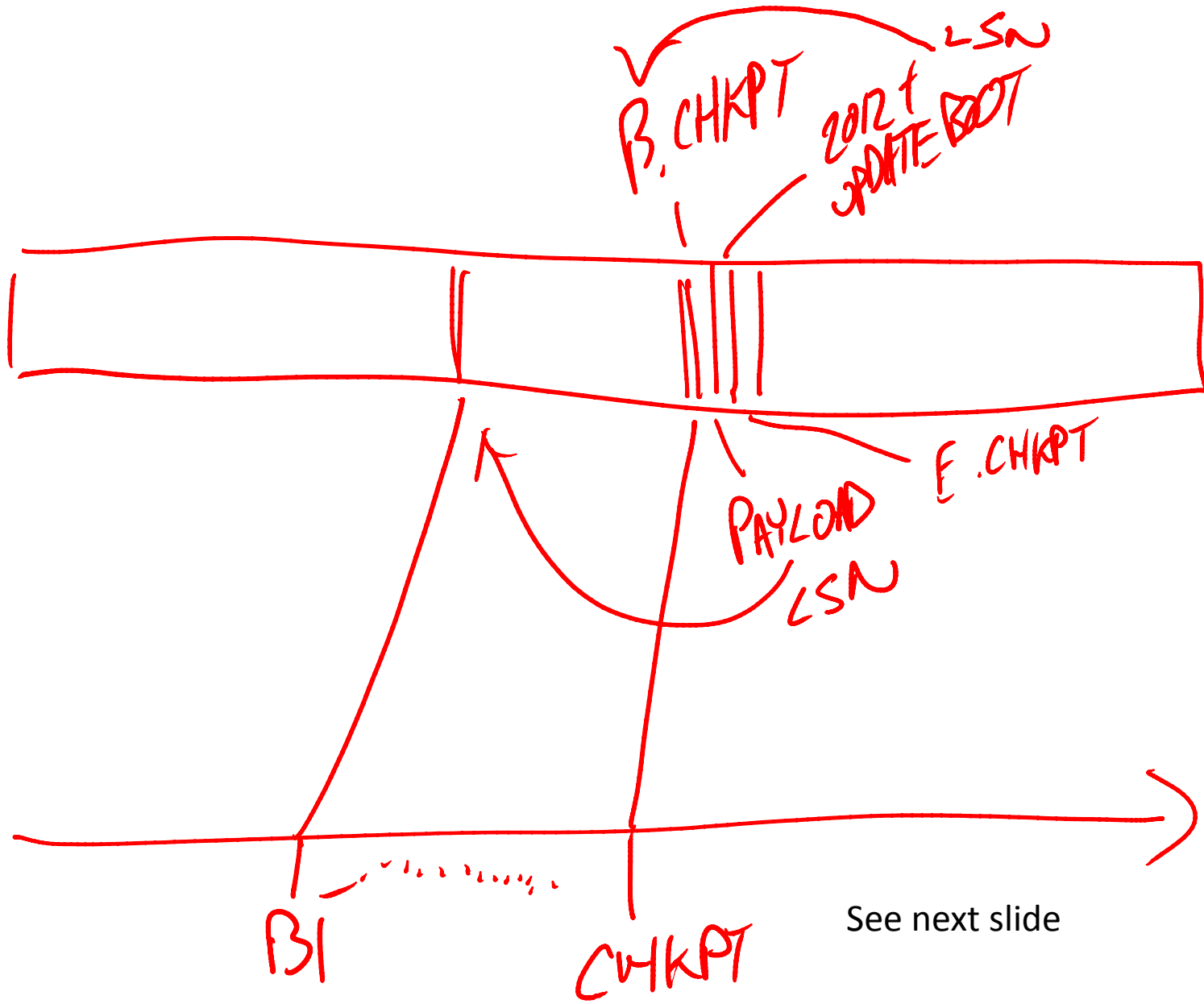
M5S19 Explaining how the Access Methods determines how to log UPDATES efficiently by splitting the logging by portion of the record. Each variable length column is a portion, as is the fixed-width portion (up to 16 bytes in each log record)



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).

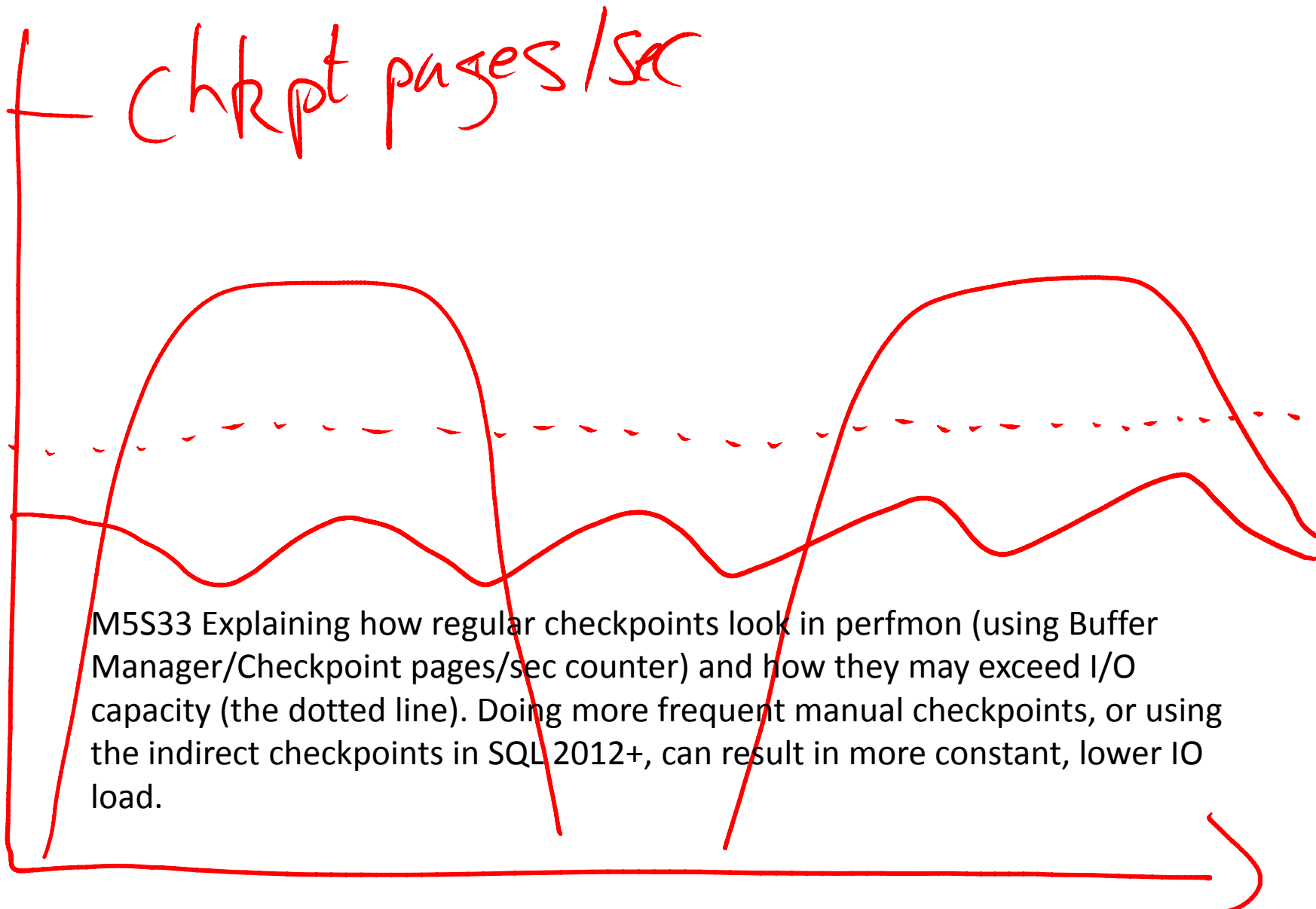
M5S24 Discussing the contents of a log block if they don't fill up right to a 512-byte boundary. Unused bytes may be zero (or whatever was in that memory prior to 2008 R2). The log block knows how much data is in it.



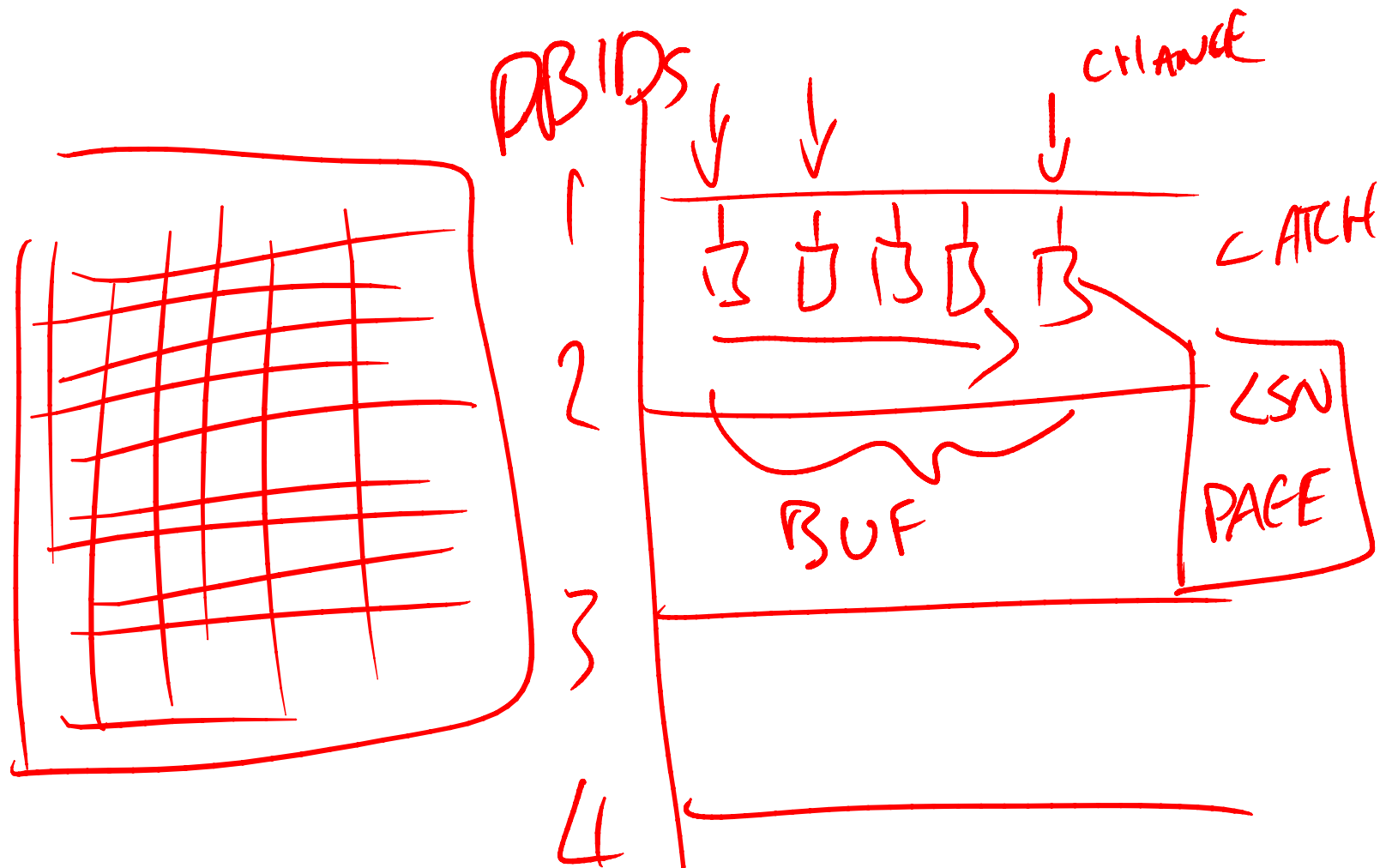


- The log records for a checkpoint:
 - Begin checkpoint
 - Payload (containing LSN of LOP_BEGIN_XACT log records for active transactions at that instant)
 - 2012+: updating boot page (happens on all versions, only logged in 2012+)
 - End checkpoint

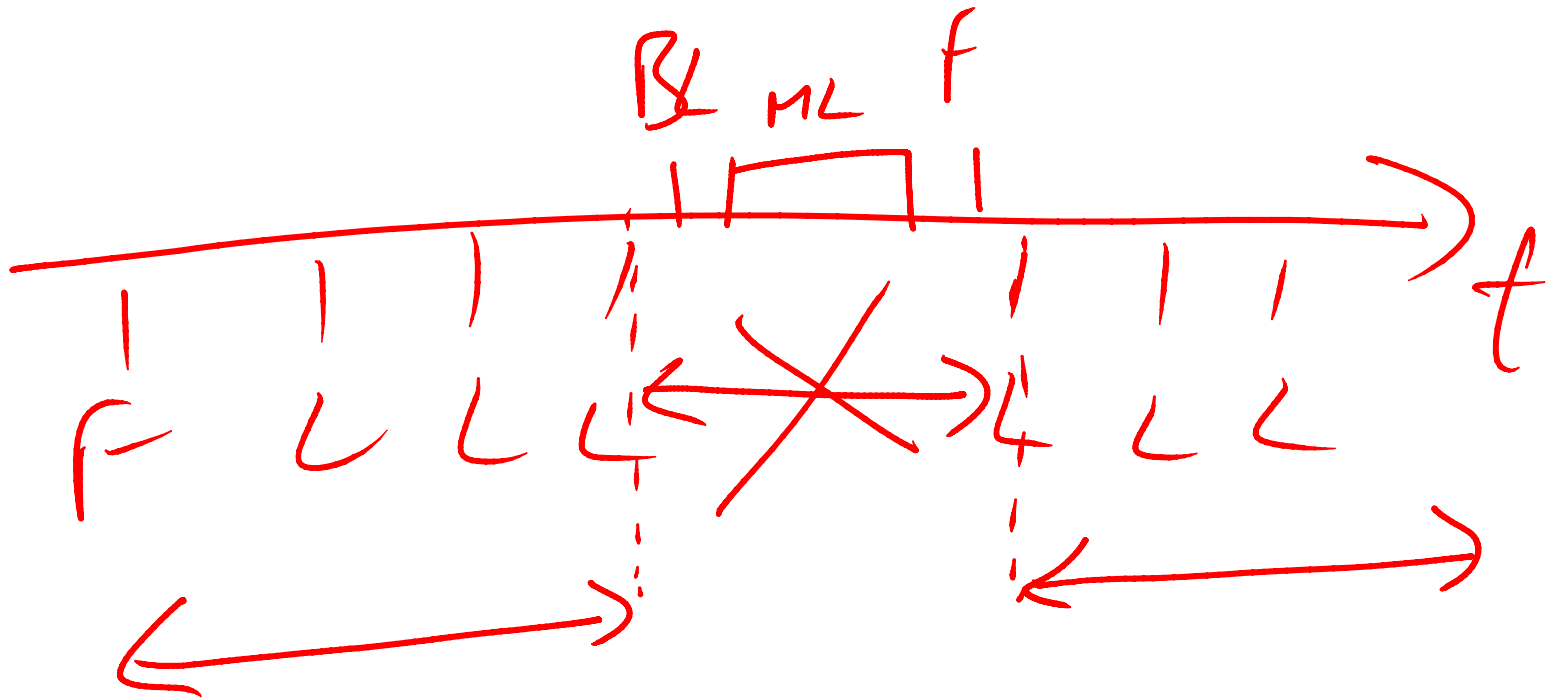
chkpt pages/sec



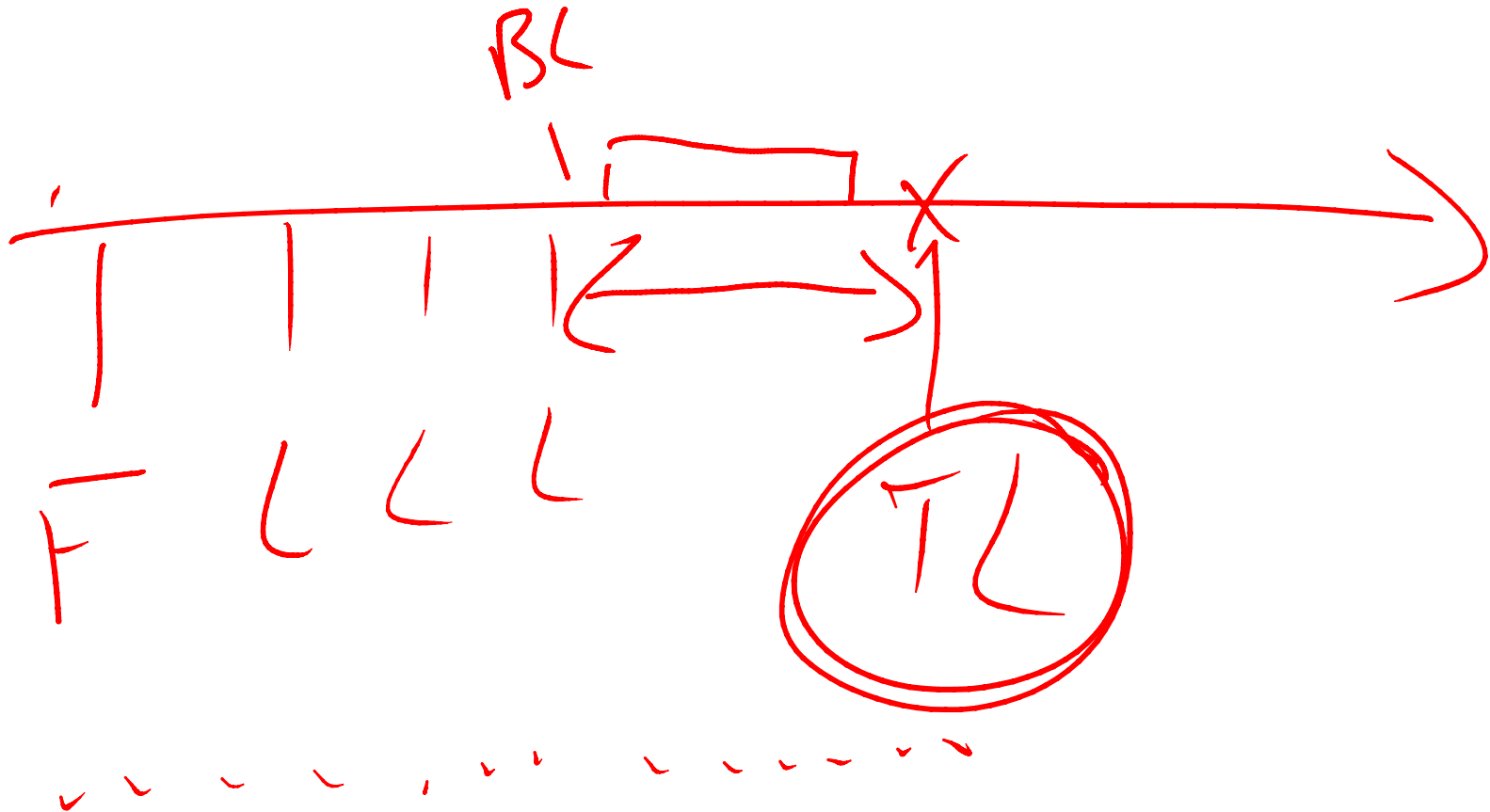
M5S33 Explaining how regular checkpoints look in perfmon (using Buffer Manager/Checkpoint pages/sec counter) and how they may exceed I/O capacity (the dotted line). Doing more frequent manual checkpoints, or using the indirect checkpoints in SQL 2012+, can result in more constant, lower IO load.



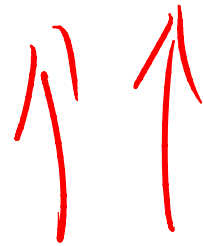
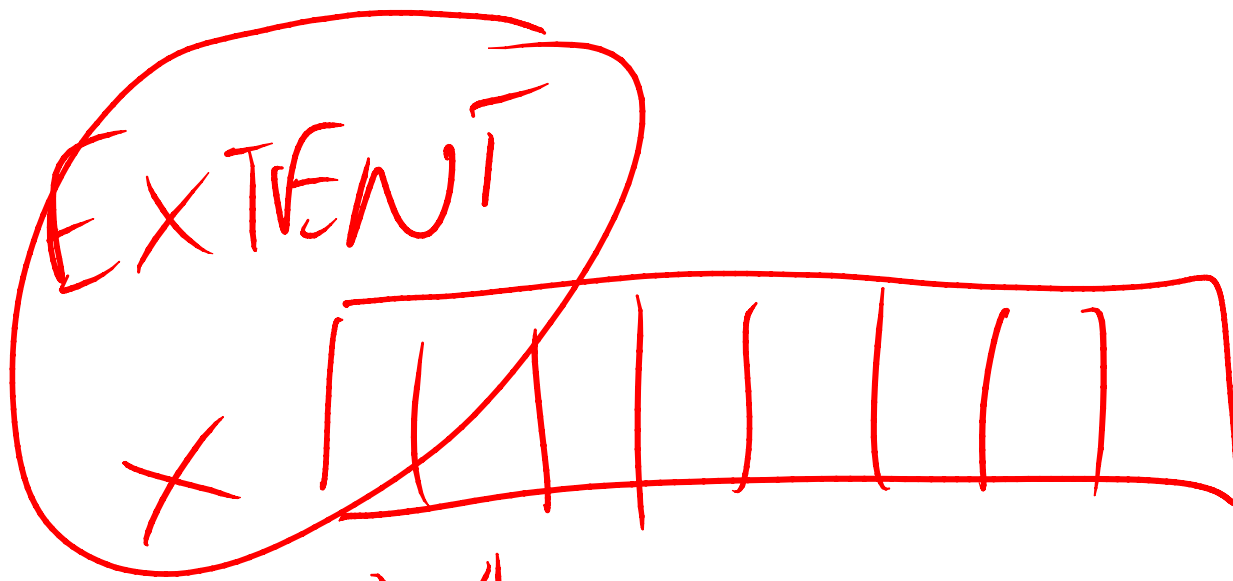
M5S31 Explaining how the buffer pool finds dirty pages to write out. It doesn't scan through memory, it looks through the list of BUF structures for the database (one per page, mapping that page into the buffer pool)



M5S63 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.

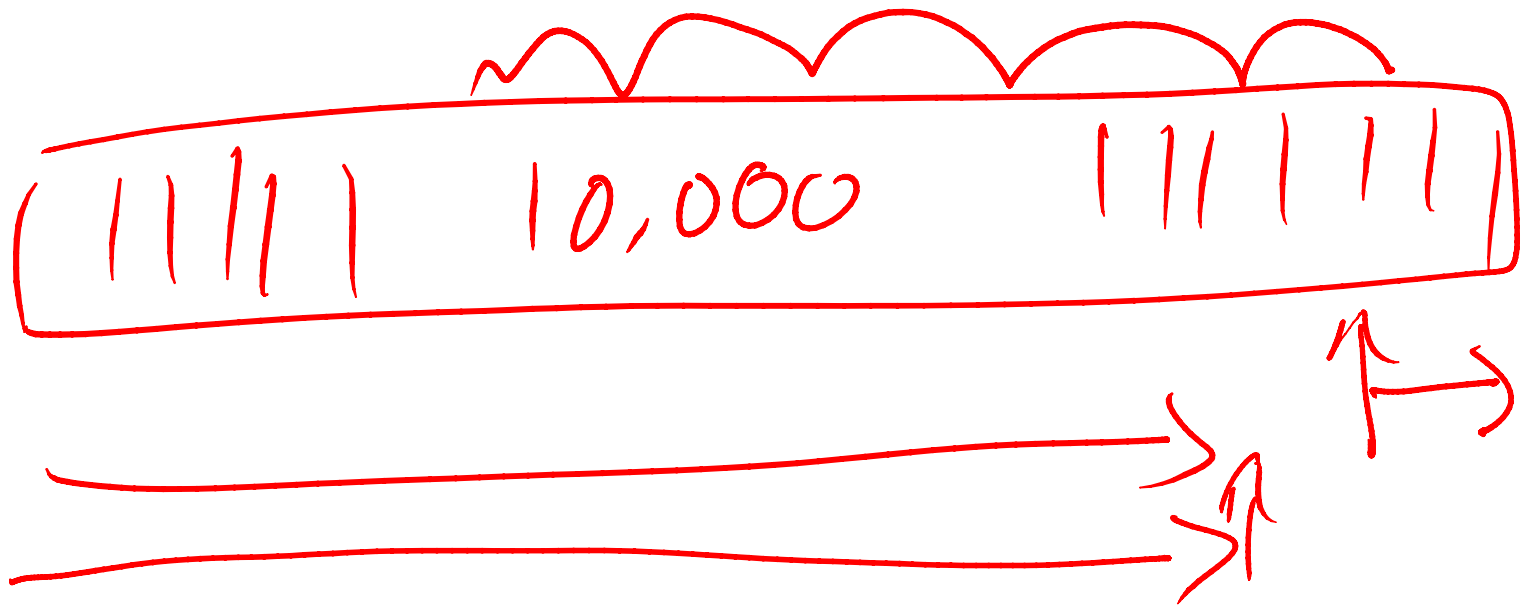


M5S63 Explaining that if a minimally-logged operation is present in a log backup, you cannot do a point-in-time restore to any time covered by that backup (except the start or end of that time period). You also cannot do a tail-of-the-log backup after a crash that destroys the data files if there has been a minimally-logged operation since the last log backup. Well in 2008 R2+, you can, but the log backup is invalid.

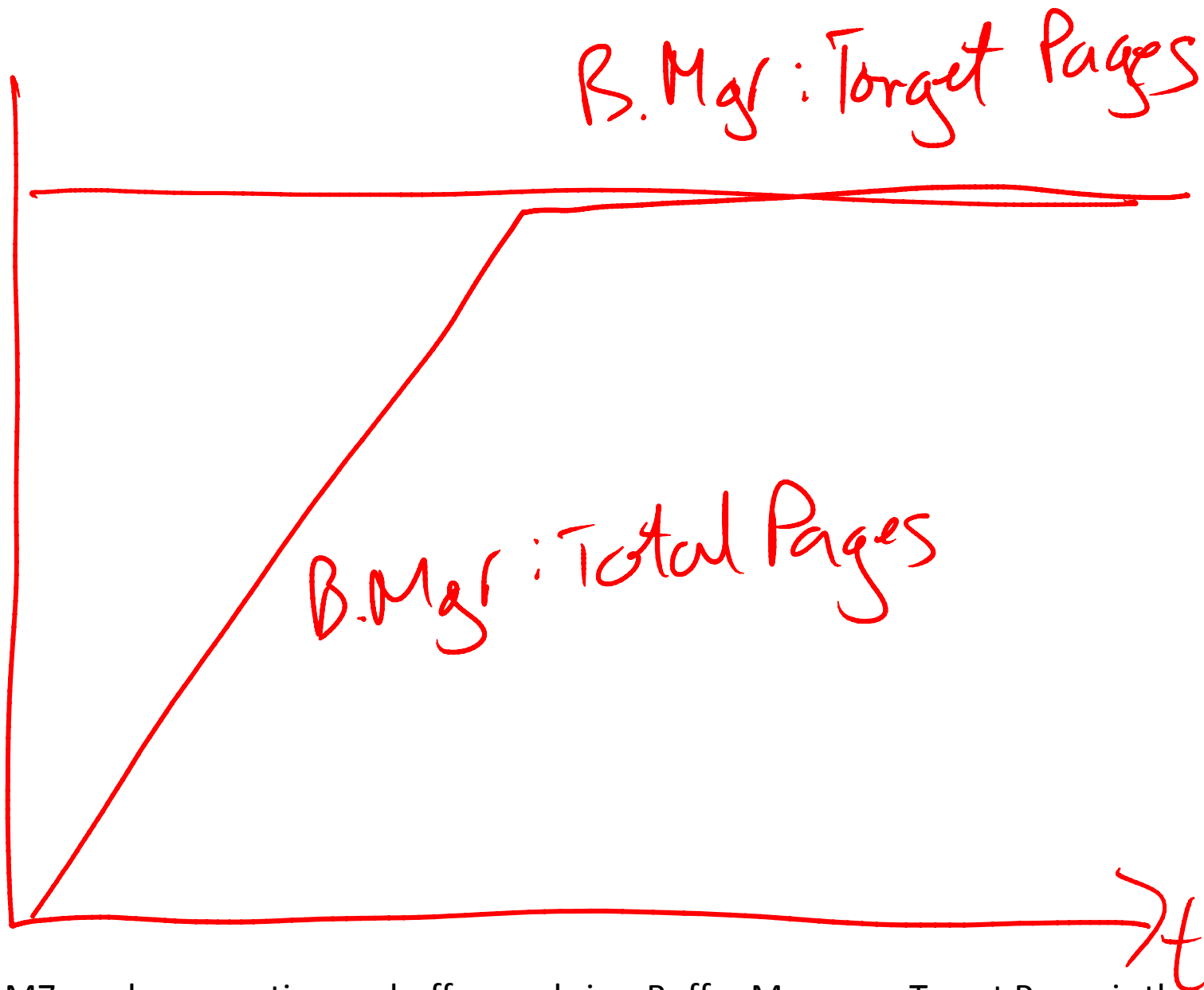


PROBE

M5S65 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.



M5S66 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.



M7 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.

COMMENTS

PAUL	JON	ERIN	ELENA
MEMBER_ID			K

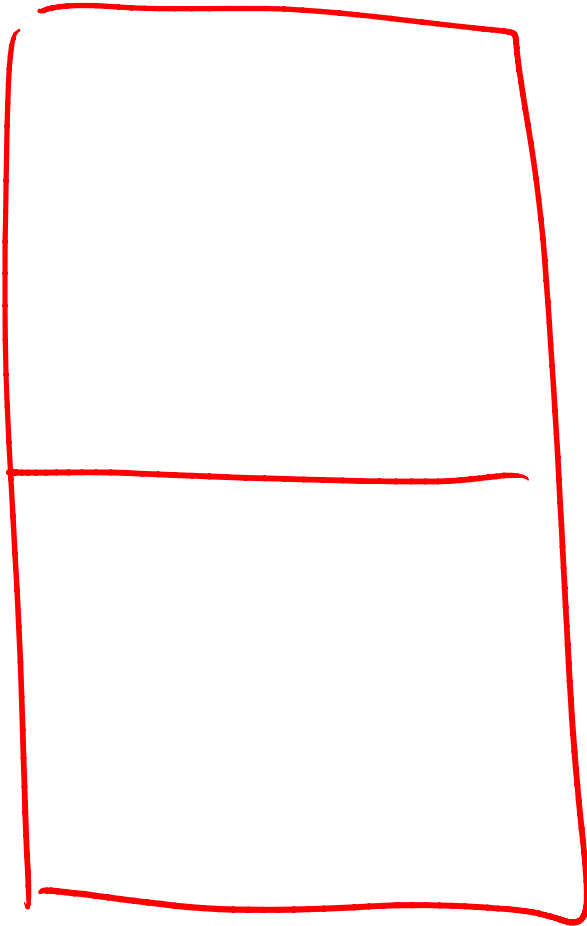
M7S19 The MySpace story. The 'K' is the tiny portion of comments on Kimberly's page because she wasn't very popular back then...



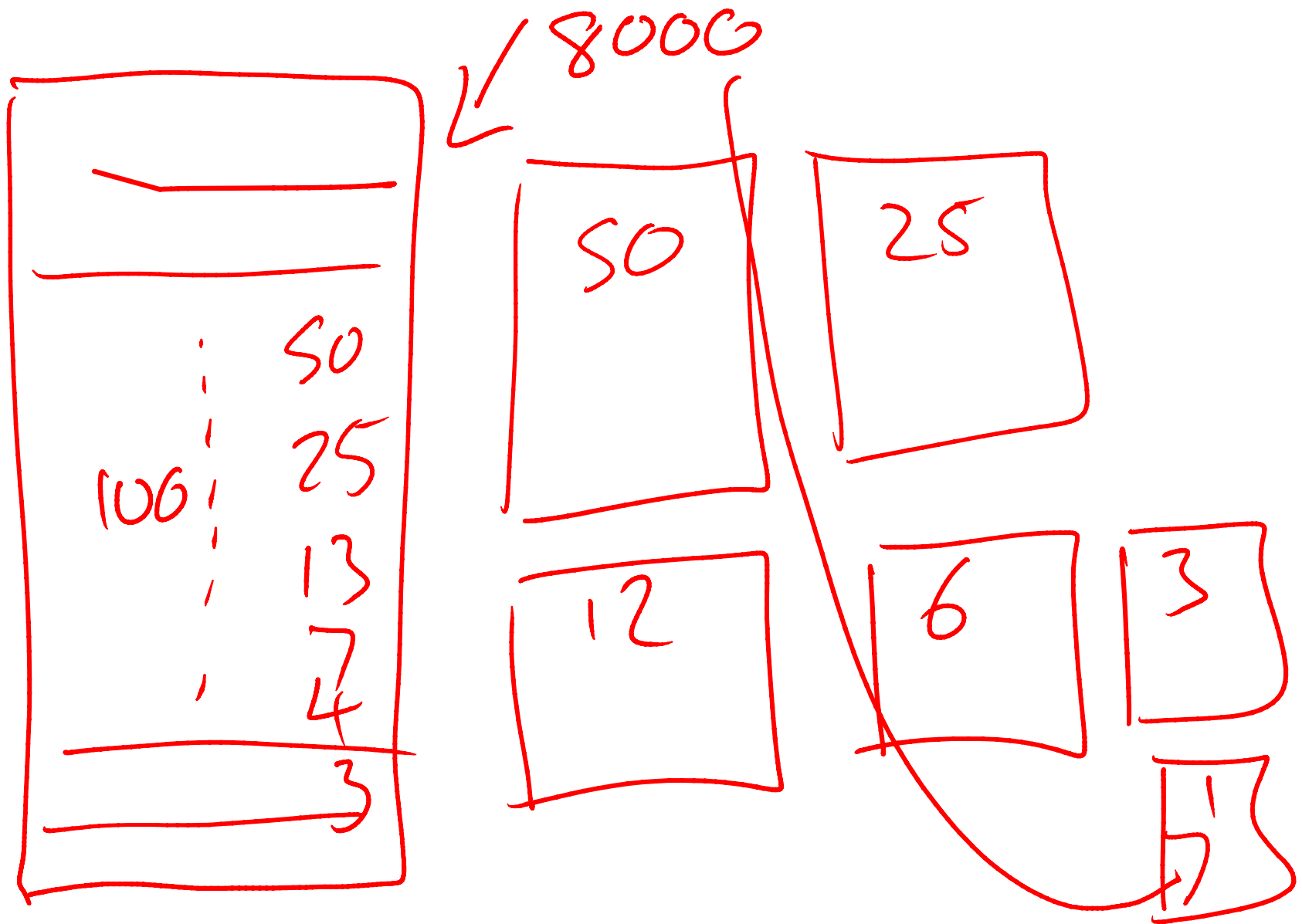
	NO SPLIT	SPLIT	?
1000	1256	7928	x5
100	356	7120	x18
10	268	11520	x45

M7S28 The numbers from the page split logging cost demo.

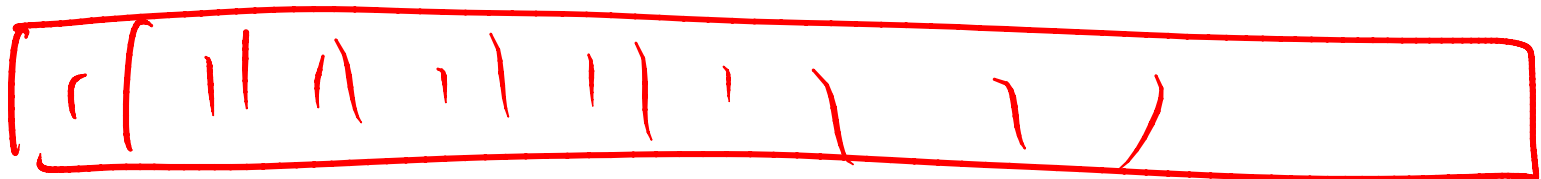
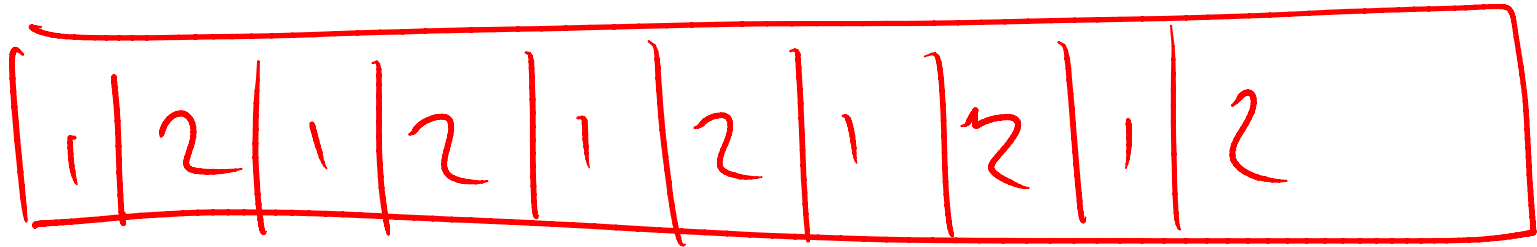
SKEW



M7S28 Discussing skewed page splits. Sometimes the Access Methods will notice a better boundary point than 50/50 to split at (e.g. two distinct sets of records, as above).



M7S28 Discussing multiple splits. With a page with 100 x 80-byte records, if an 8,000-byte record is inserted, it will split multiple times to make room.



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)

ITB



M7S44 Explaining staggered index maintenance with database mirroring. See <http://bit.ly/IS04W5> for more explanation