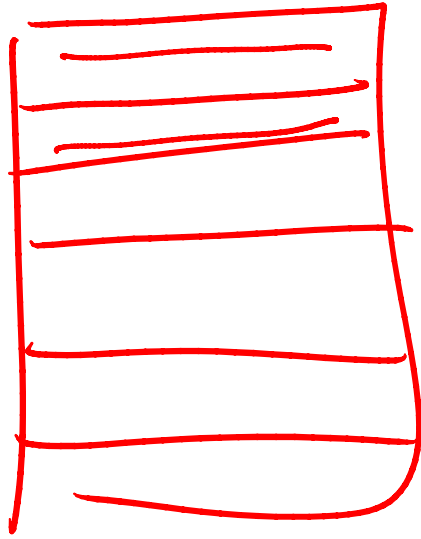
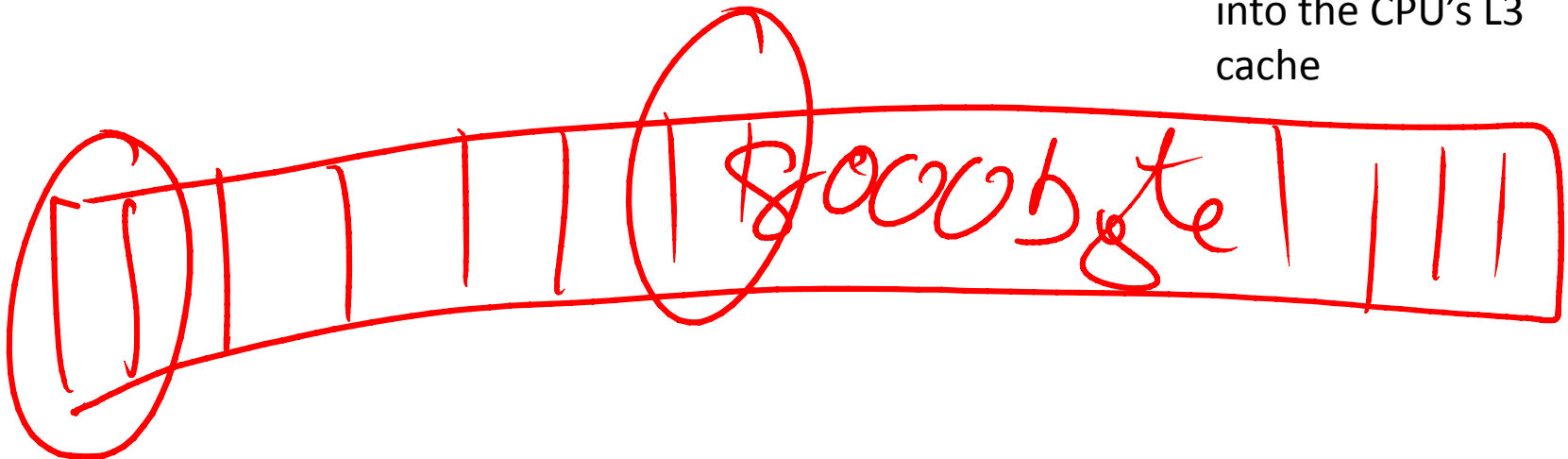


CACHE LINE INVALIDATION

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 L3
cache



c1, c2... c10



HEAP
CL

SEL c1, c6, 8

W. c2 = x

PTR

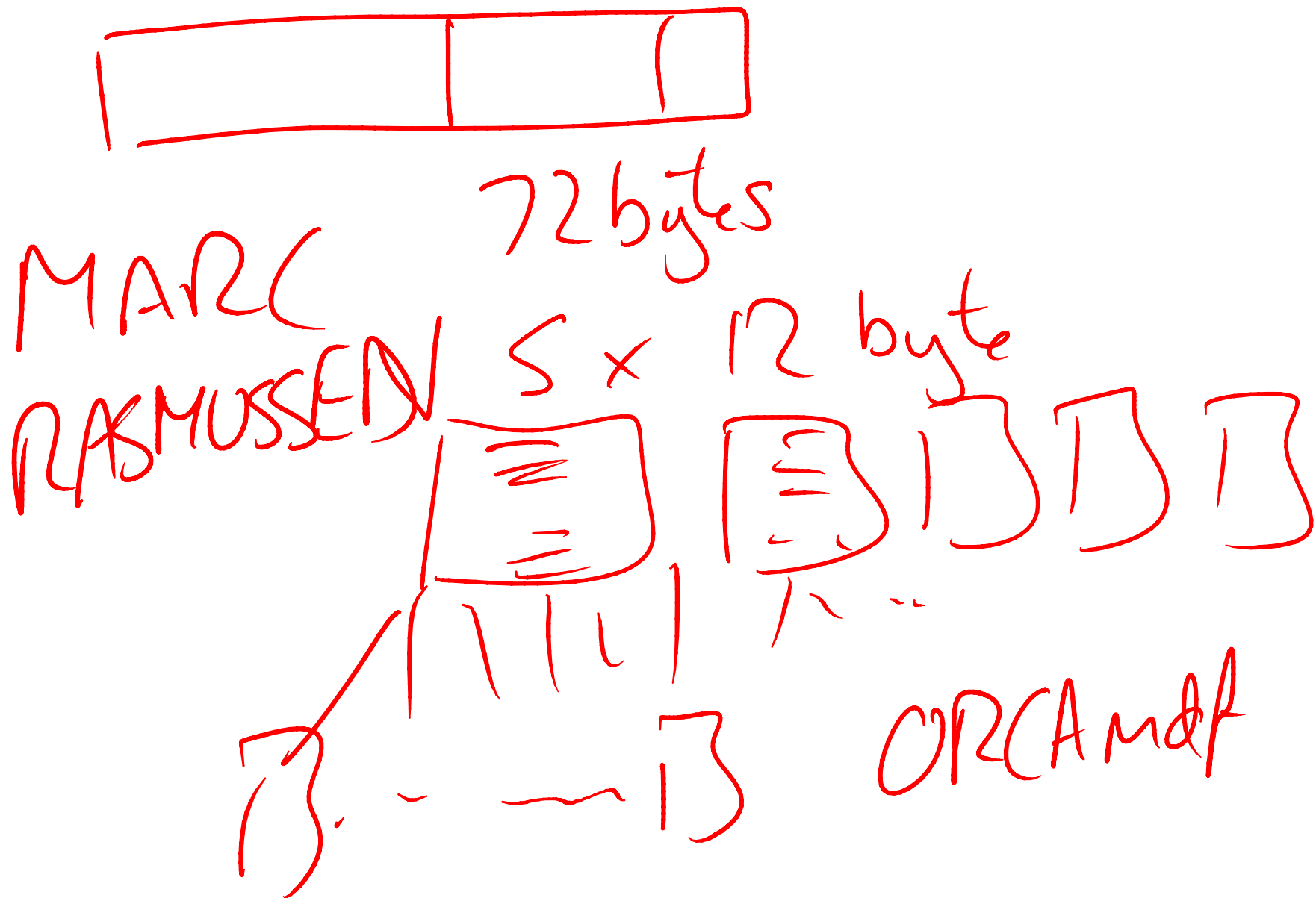
PHYSICAL RID

LOGICAL



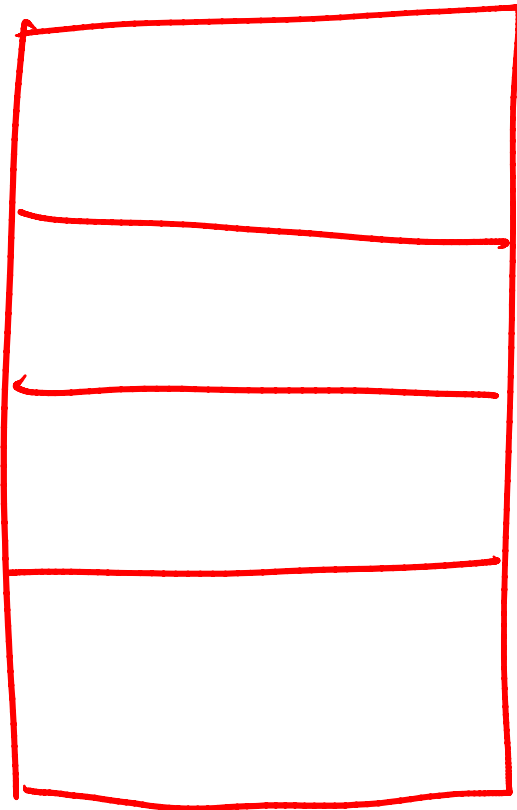
M1S09 Explaining NC index back-links to the data records as part of describing why forwarding/forwarded records exist. Without them, moving a heap row would entail changing all NC index records with a back-link to that row.

(FILE : PAGE : SLOT)
(CL KEY(S))



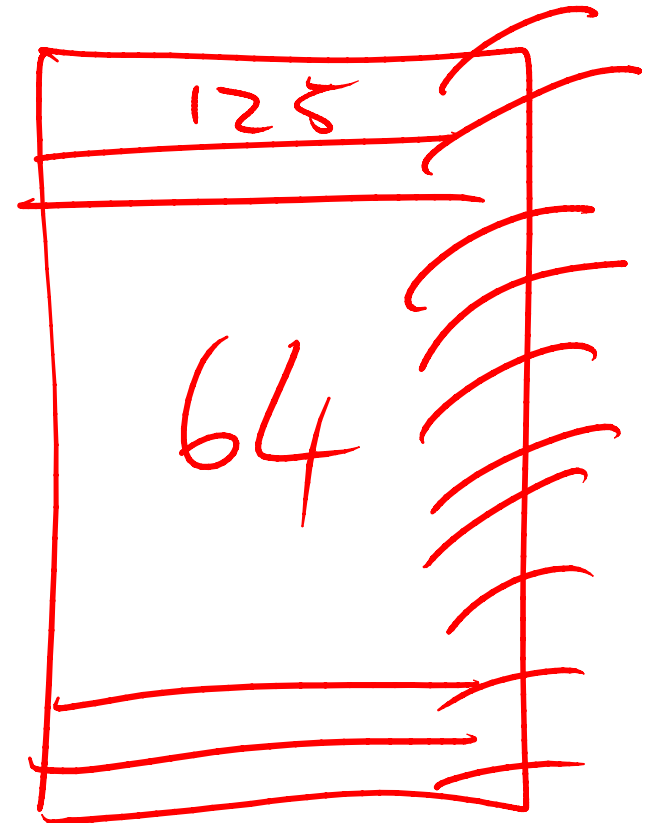
Google Marc Rasmussen for his blog. He's blogged text tree structures.

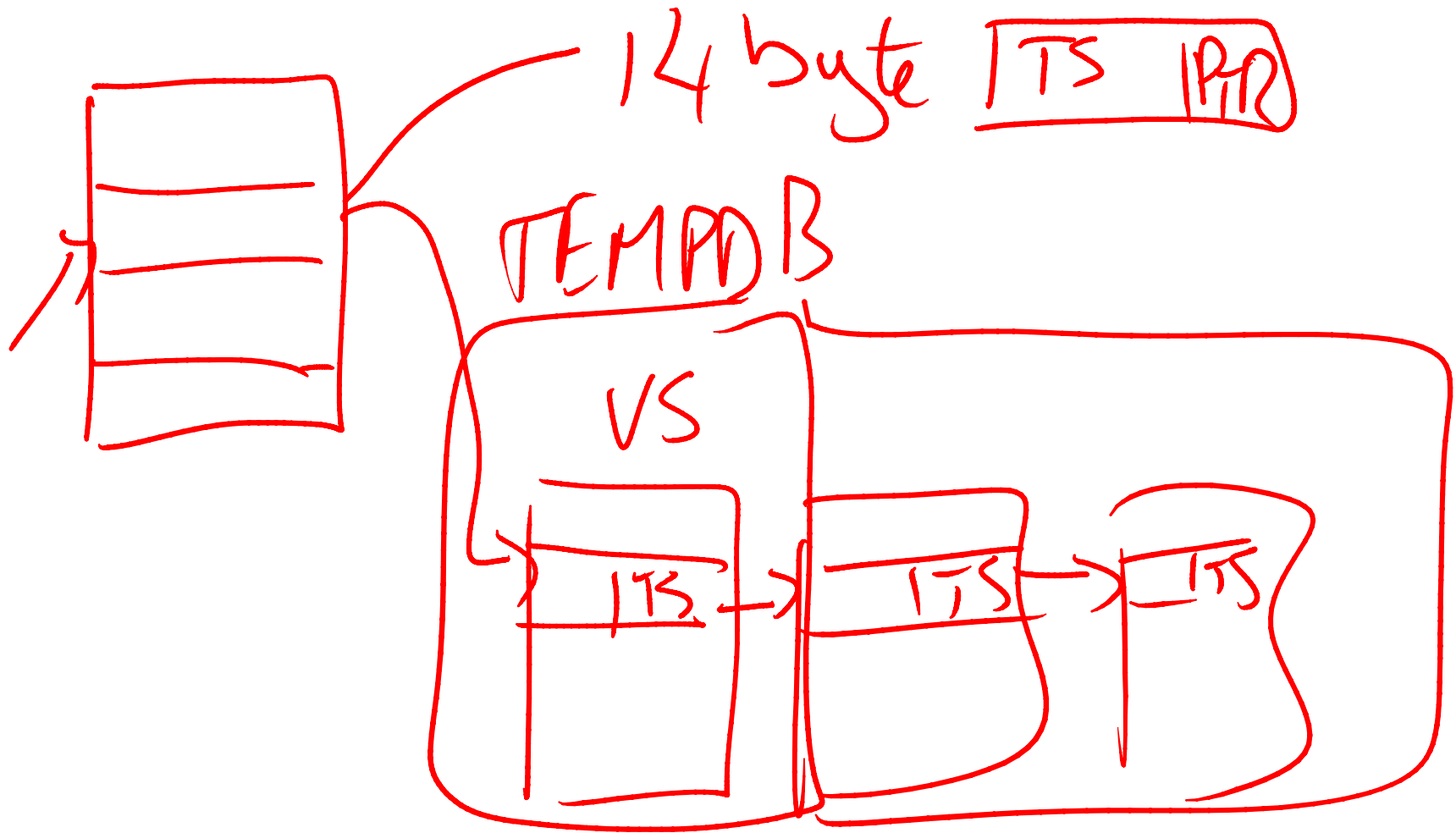
100	1900 byte
-----	-----------



4

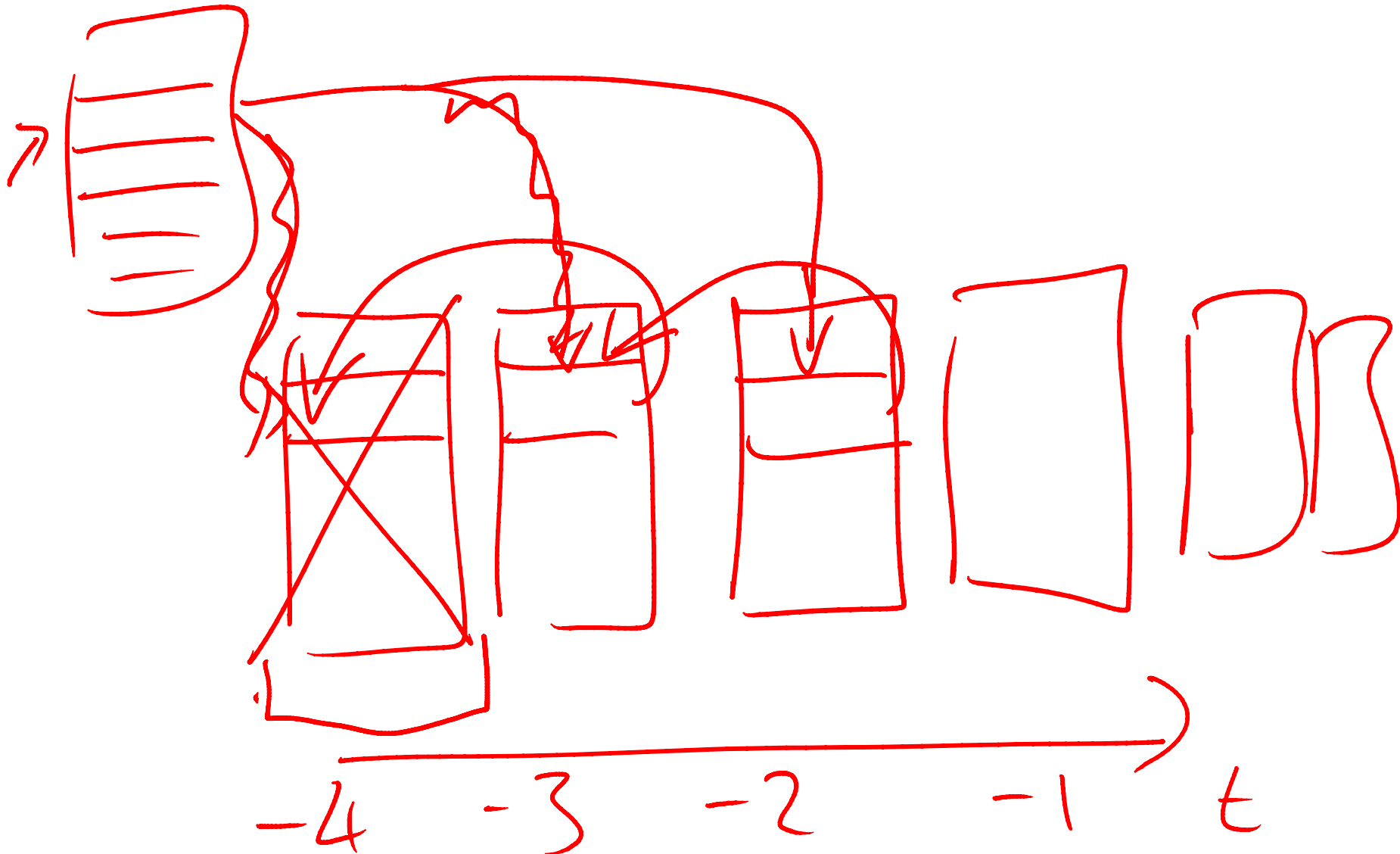
M1S12 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.

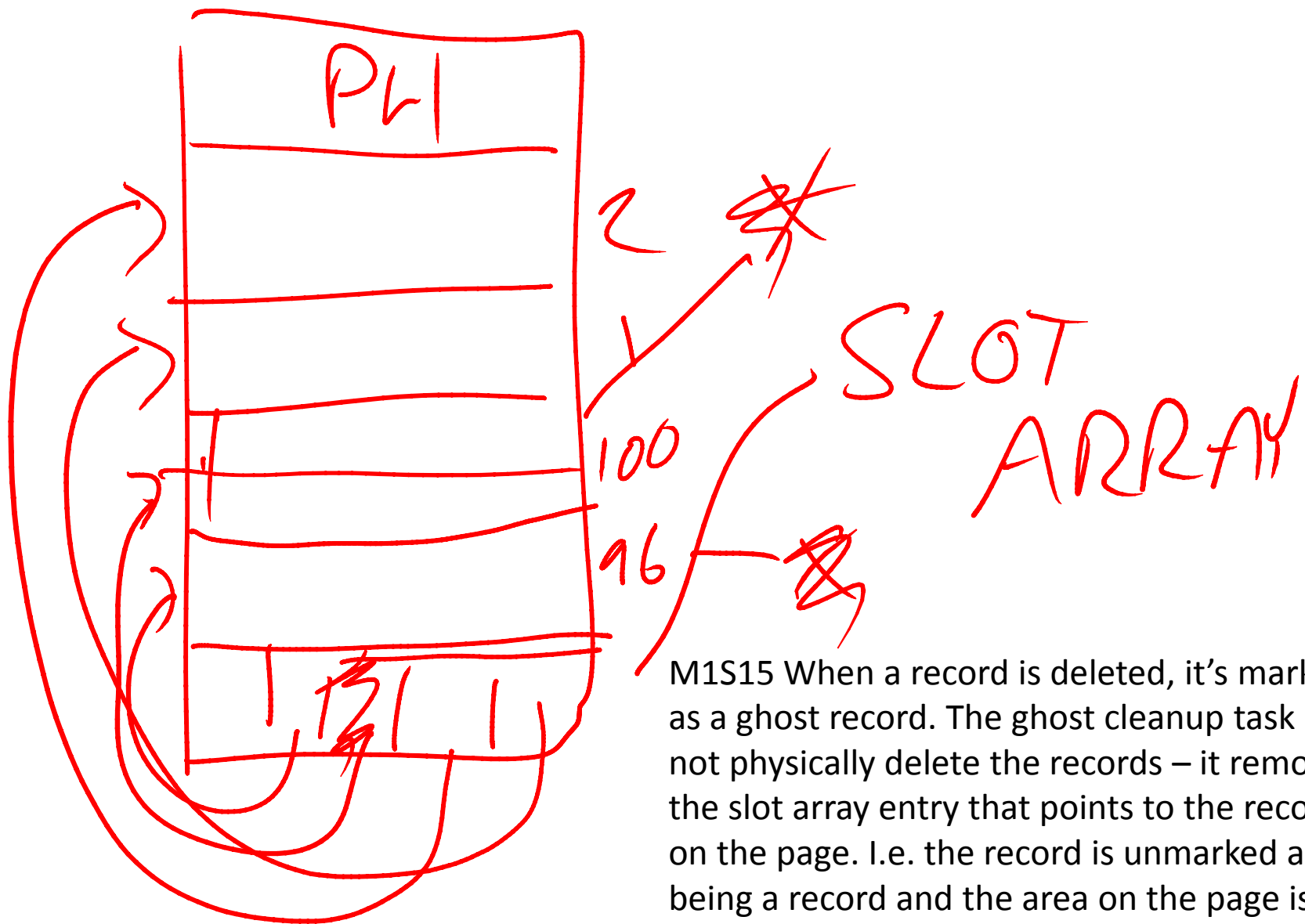




M1S14 When an update occurs to a record and snapshot isolation is enabled, the pre-change record is copied into the version store in tempdb. The changed record has a 14-byte tag on the end with the timestamp and pointer to previous record version. A chain of previous versions is possible.

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





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.

LOCK HASH

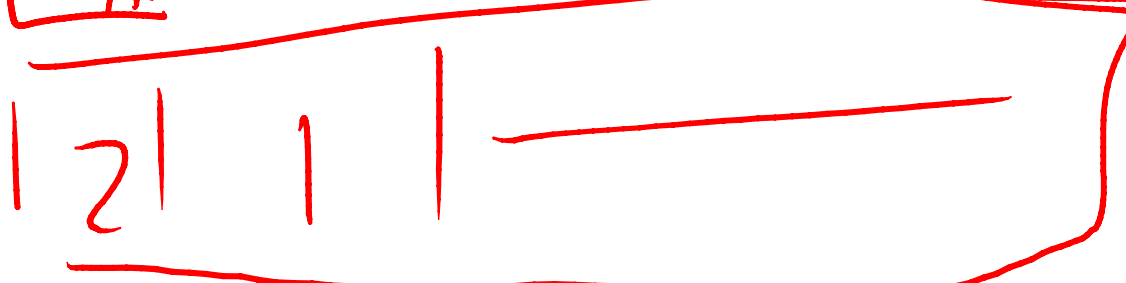
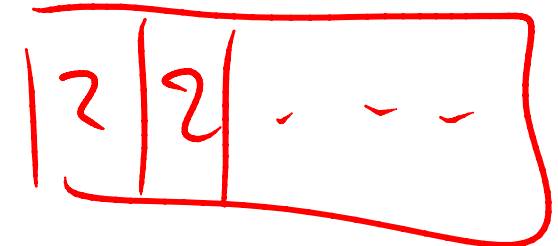
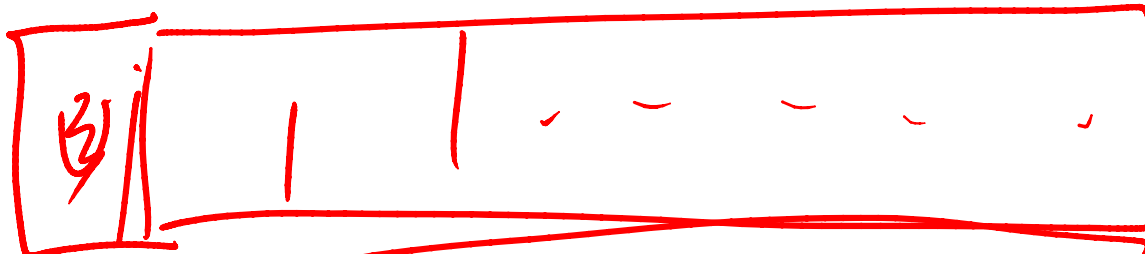
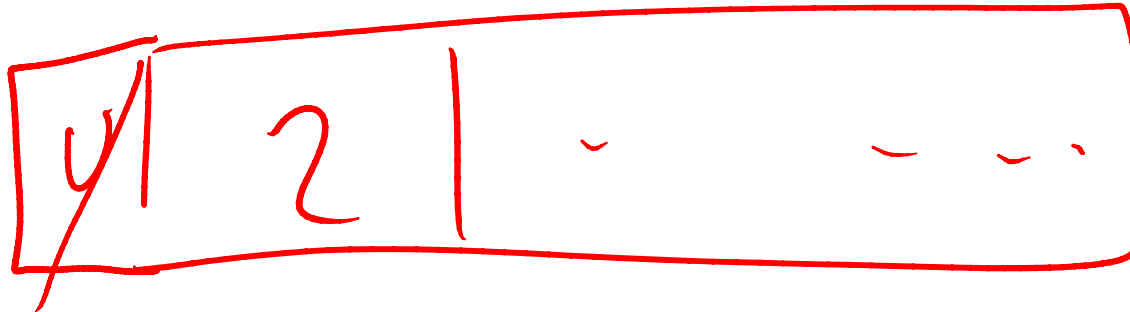
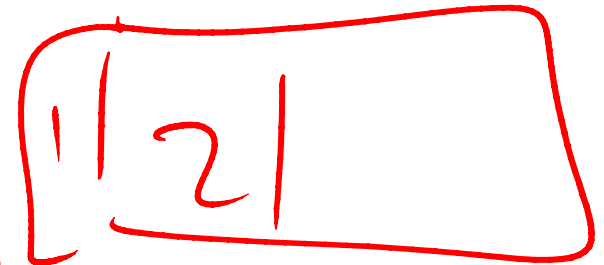
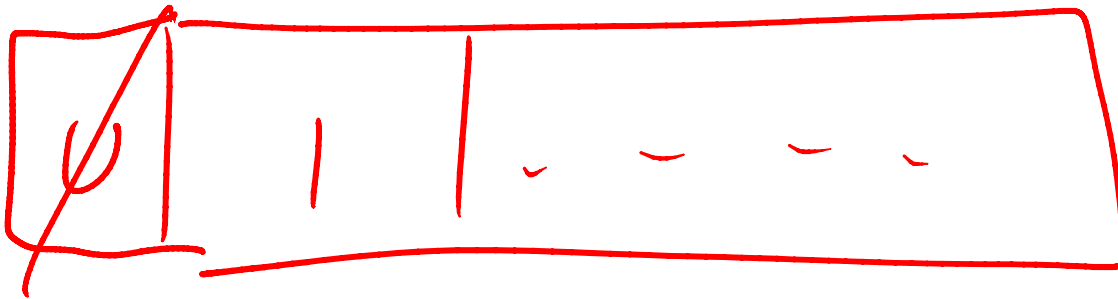
c1, c2, c3, c4

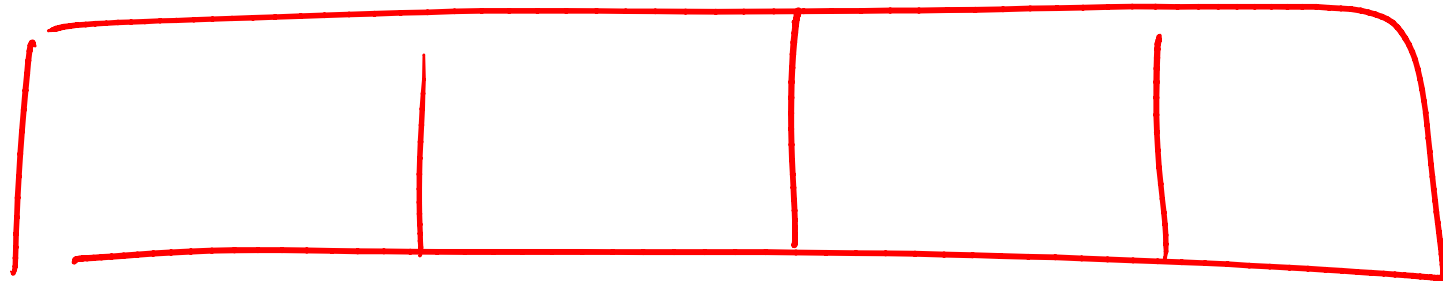
M1 Discussing lock hash collisions. See <http://bit.ly/YplGld>

A B C D → X

Z Y X W → X

M1 Explaining uniquifiers. It's a 4 byte column added as the leading key to non-unique clustered indexes and is used as a tie-breaker between non-unique keys.





0 — 7 8 — 15 16 — 23

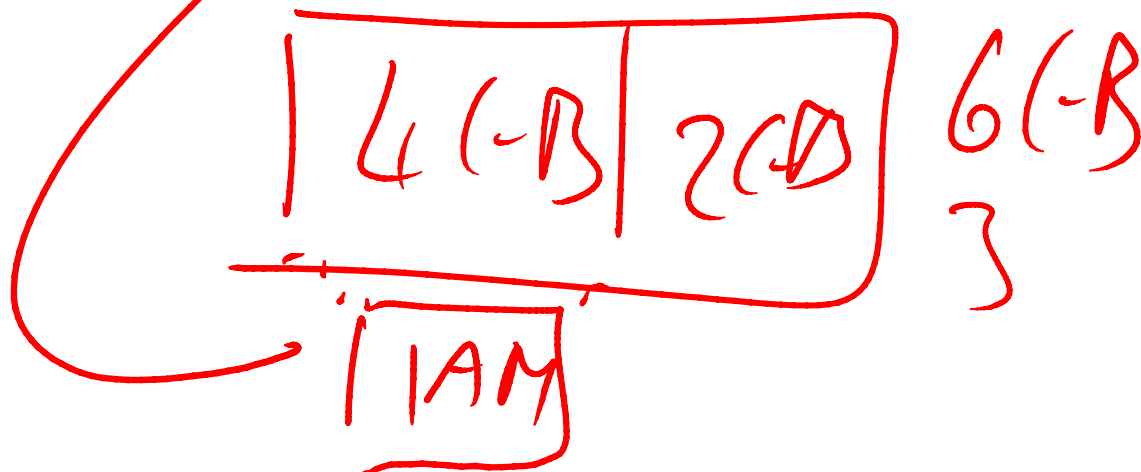


M1S24 Explaining the difference between mixed and dedicated extents. Mixed extent is one in which the 8 pages could be allocated to 8 different objects. Dedicated extent is one where all 8 pages are reserved for exclusive use of a single object.

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

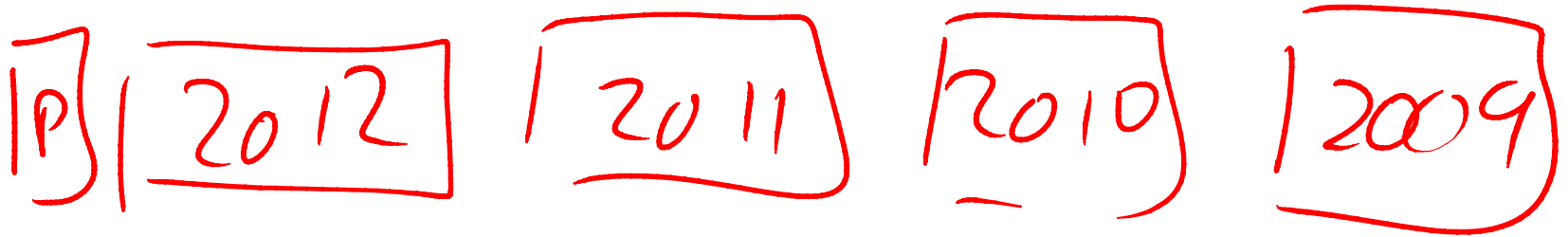
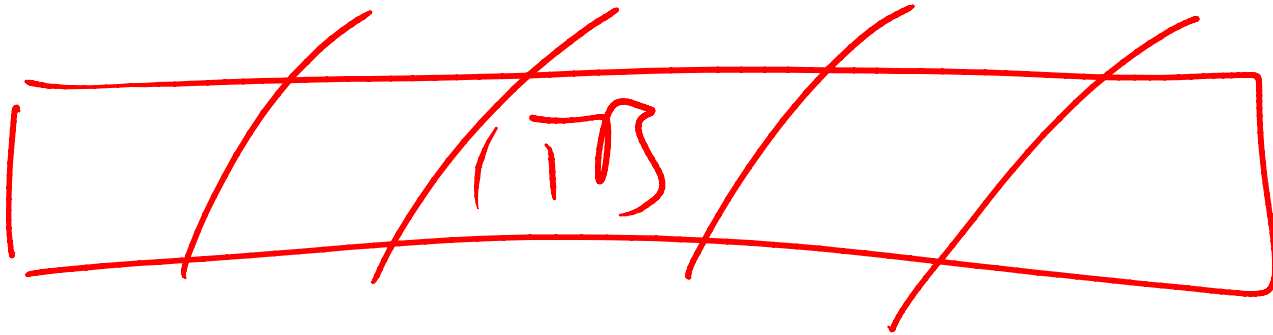


M1S36 Example of IAM chains.
 Allocations from each 4GB
 chunk of a data file require an
 IAM page to track the
 allocation. Multiple IAM pages
 make an IAM chain.



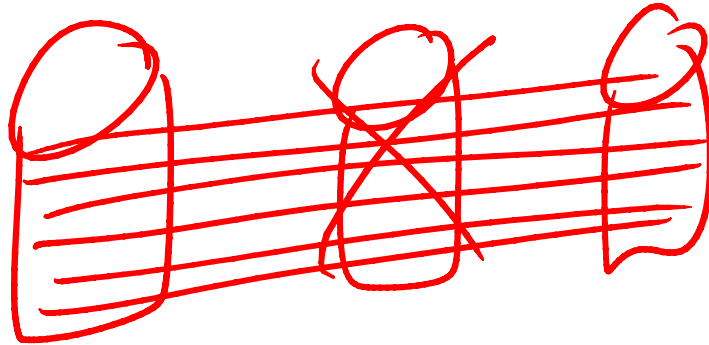
`%%LOCKRES%%`

M1 If you want to see what the lock hash resource for records will be inside the lock manager, use the undocumented `%%LOCKRES%%` function.



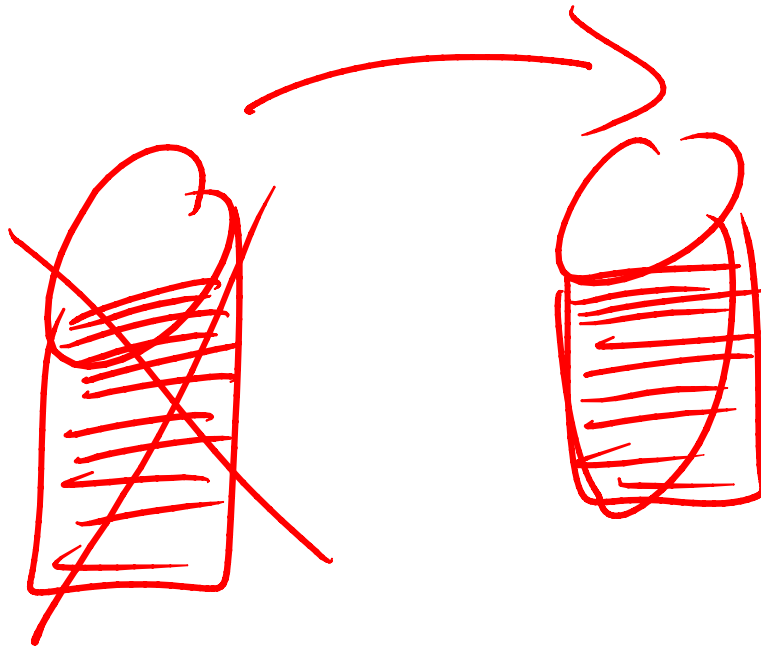
M2S7 Splitting a large database up into multiple filegroups can enable you to do small, targeted restores or even a partial restore from scratch – in the event that the database is damaged or destroyed.

$R \phi = \text{STRIPING}$



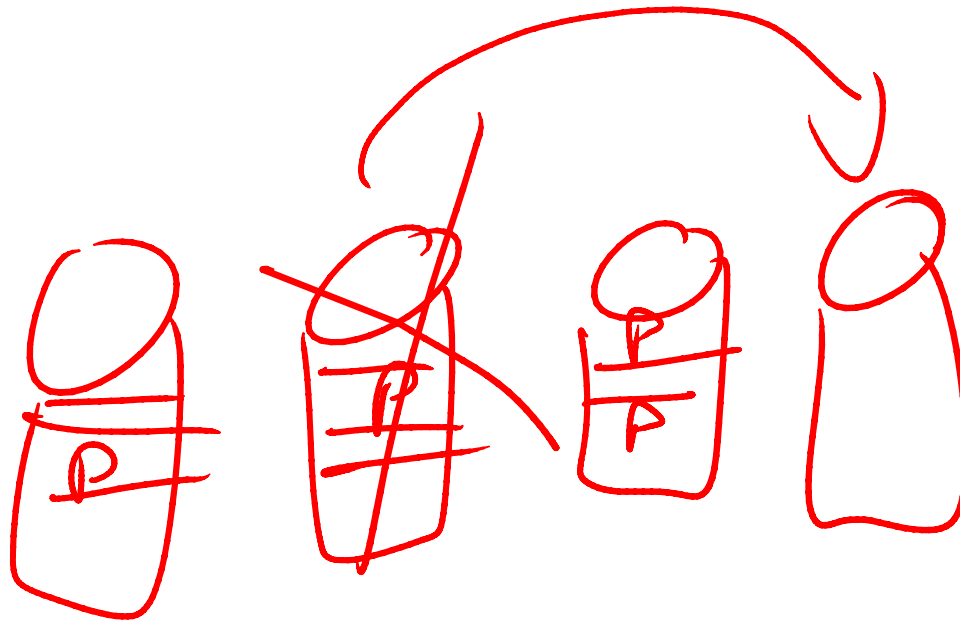
http://en.wikipedia.org/wiki/Standard_RAID_levels

RA1 = MIRRORING



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

RS = ROTATING
PARITY



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

$$21 + 0 = 10$$

$$M + S$$



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

sys.dm_io_virtual_file_stats
dm_io_pending_io_requests

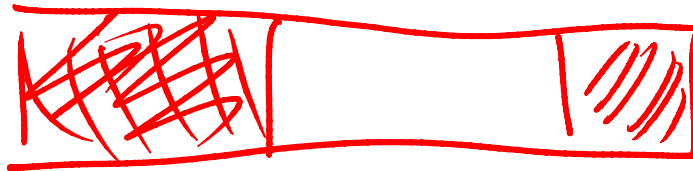
M2 Tracking I/O latencies: see <http://www.sqlskills.com/blogs/paul/how-to-examine-io-subsystem-latencies-from-within-sql-server/>

wmic

diskpart

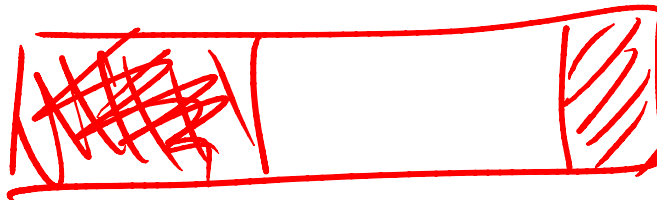
M2S8 The two commands you can use to check partition alignment. Don't use diskpart though as it rounds and can give false results.

RR

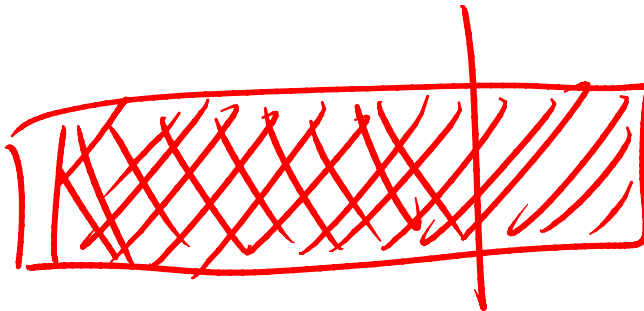


PROP FILL
WEIGHTING

~~8100~~
SO (8099)

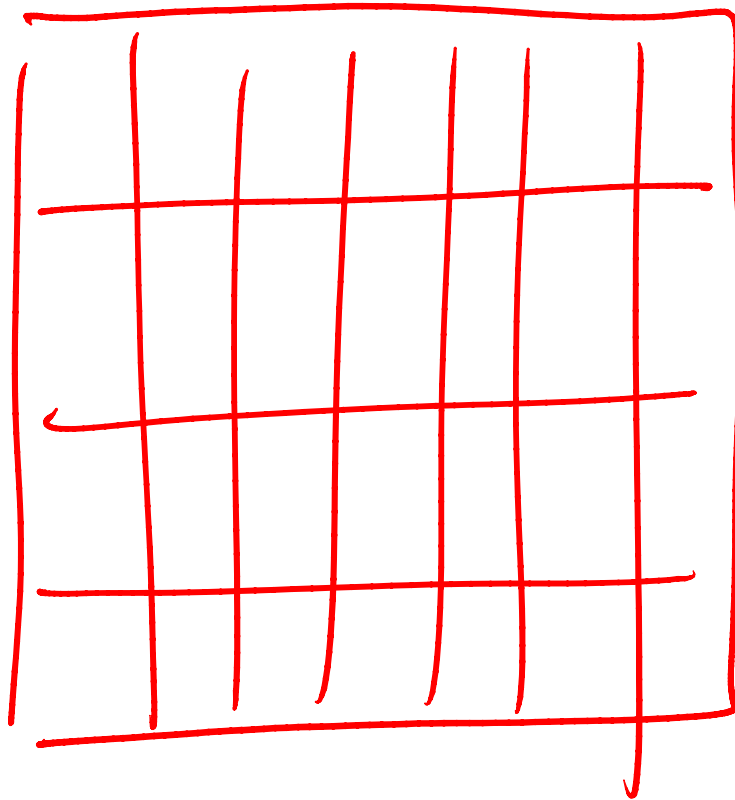


~~8100~~
SO (8099)

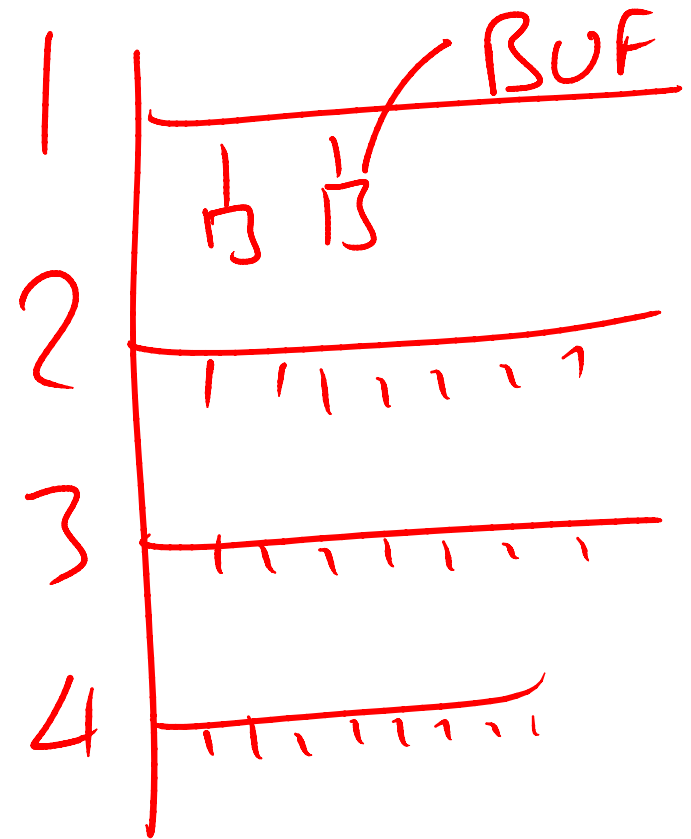


1
8100 ← ✓

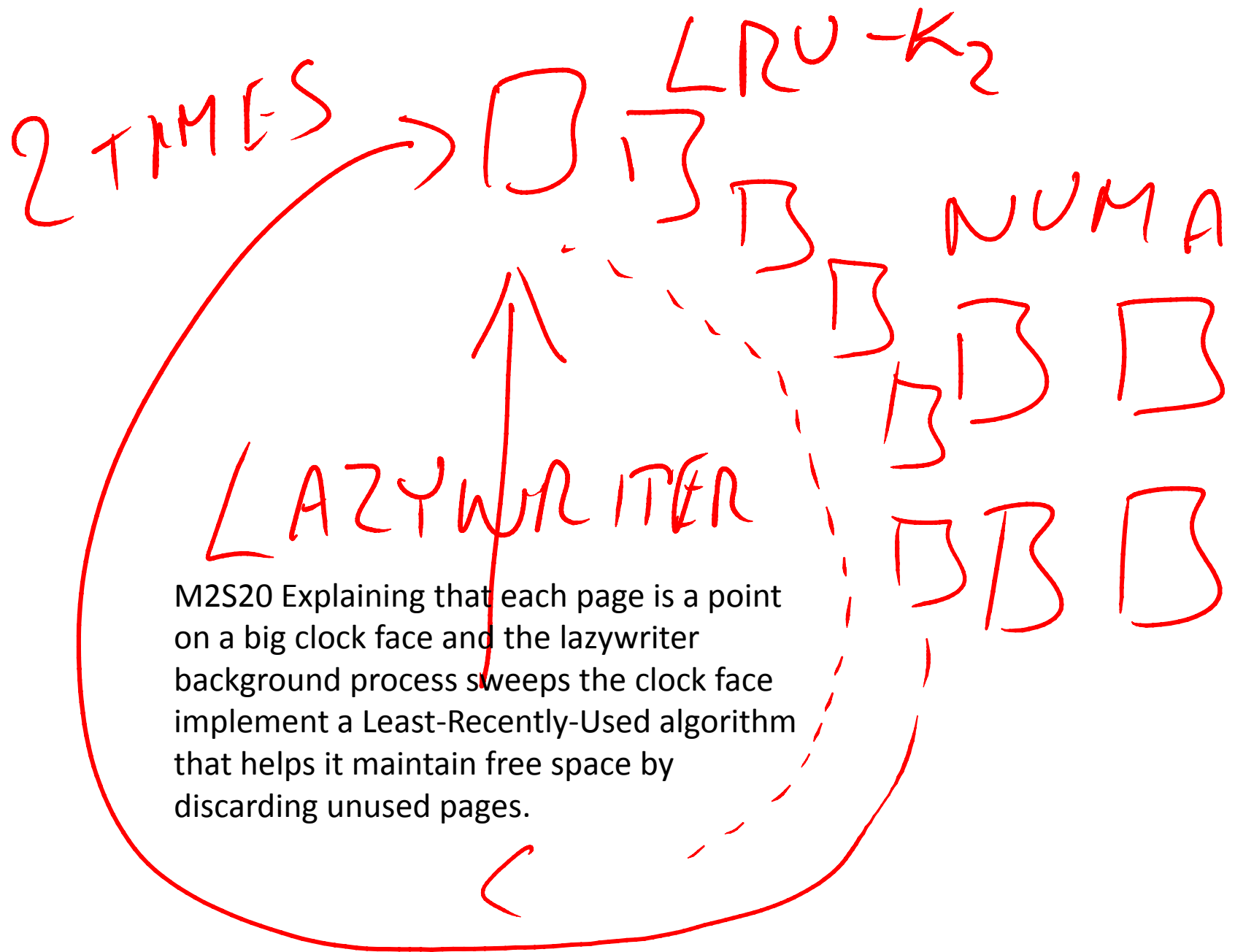
M2S19 Explaining round-robin
allocation and proportional fill



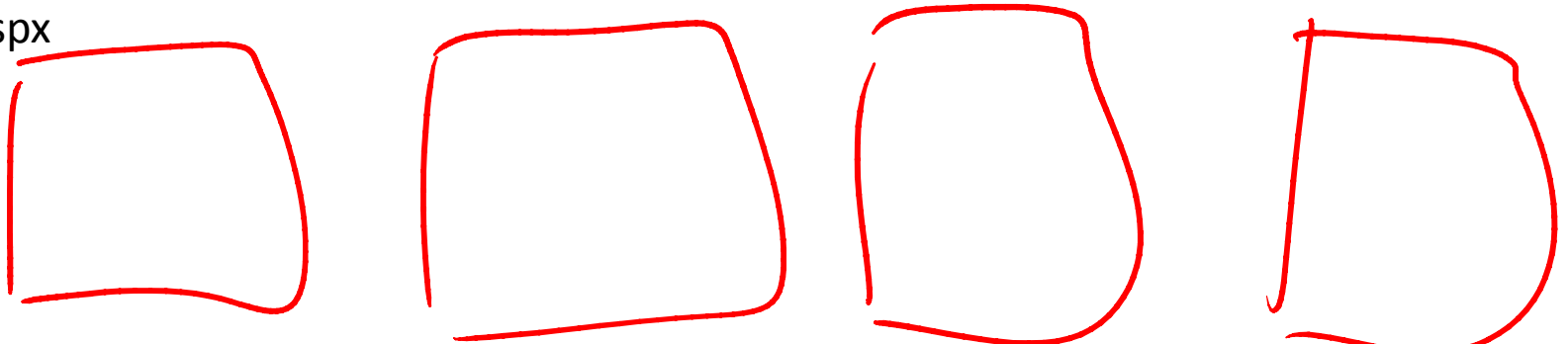
HASH LISTS



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



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



PLE
4000

PLE
4000

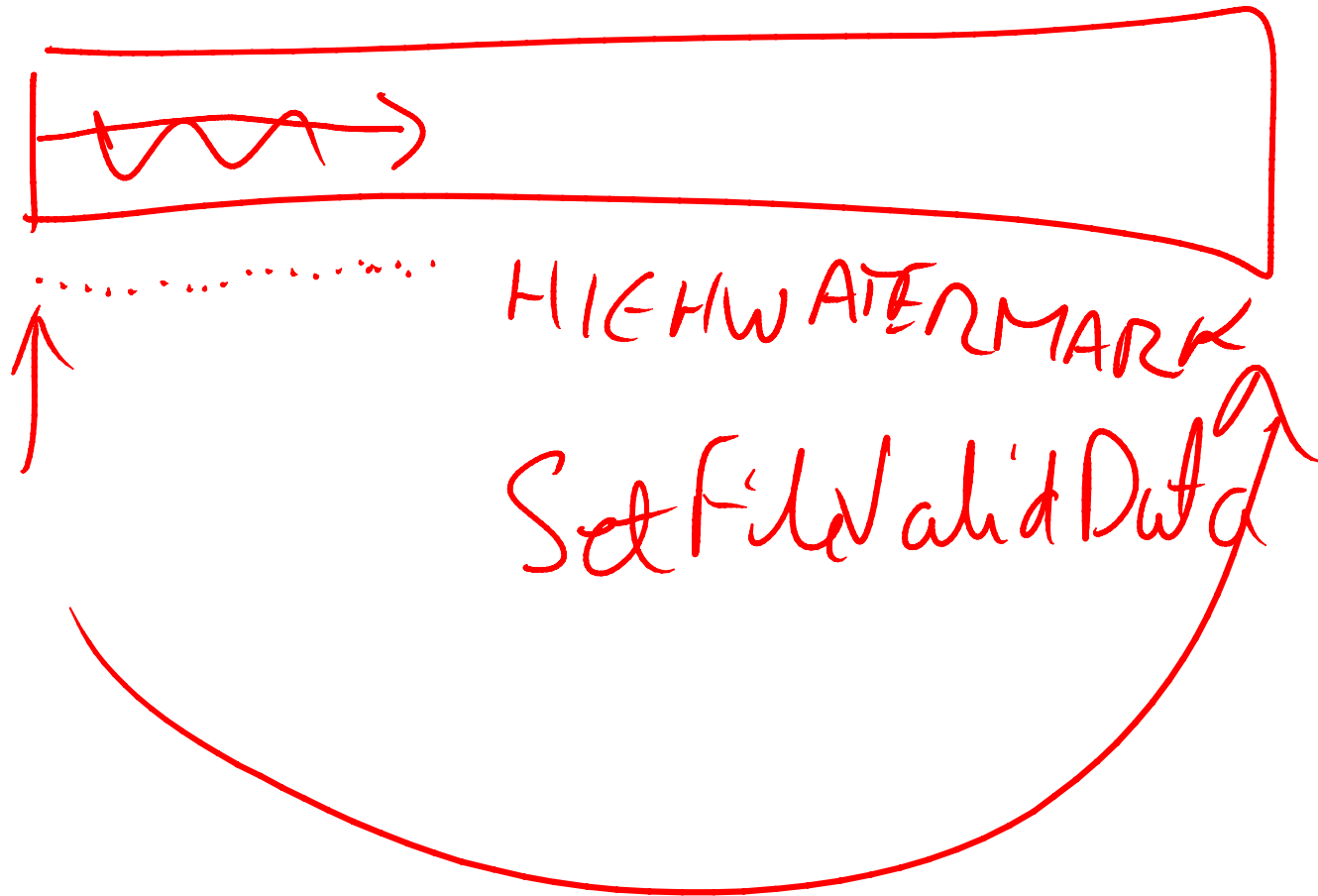
PLE
4000

PLE
4000

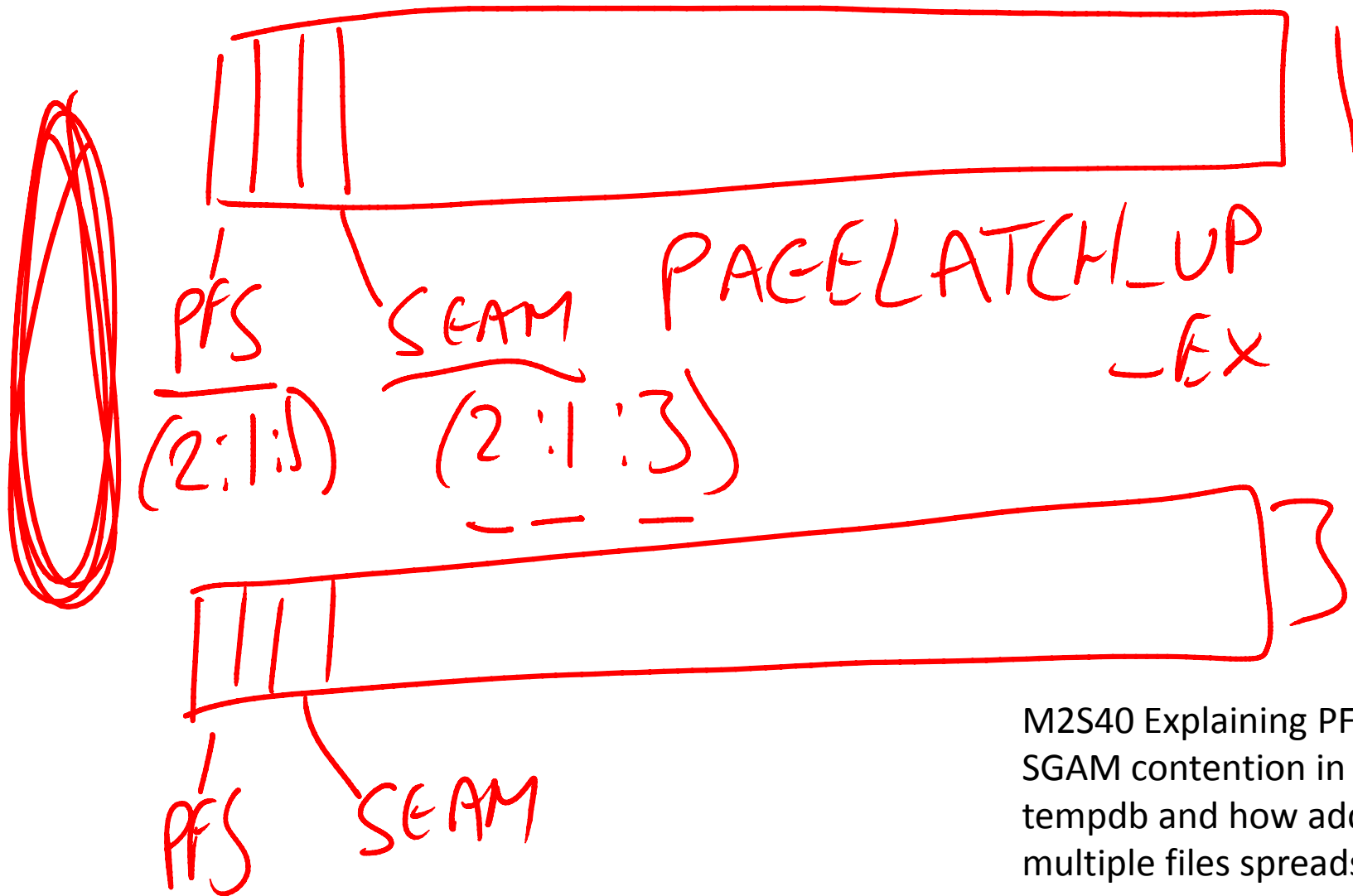
$$BM/PLE = (1+2+3+4) / 4 \downarrow$$
$$= 4000 \quad 1000$$

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

3625
BUFFER MANAGER $\Rightarrow$ BUFFER PARTITION

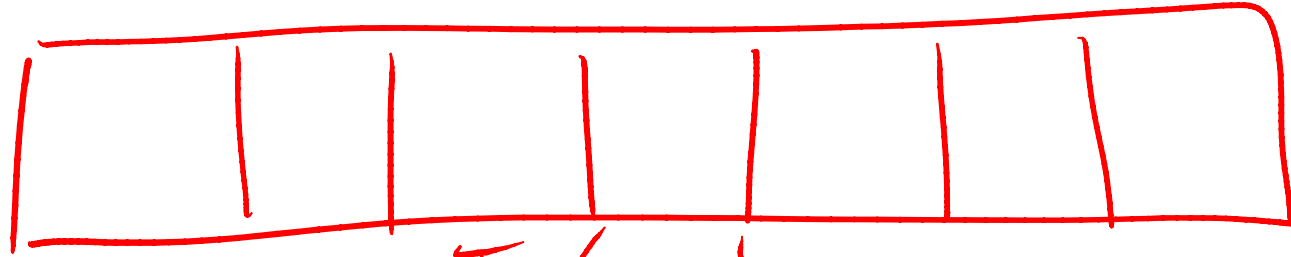


M2S22 Explaining zero initialization and how NTFS does not trust newly allocated space until the highwatermark of the file has increased, either through consecutive writes, or setting the HWM through a called to SetFileValidData.

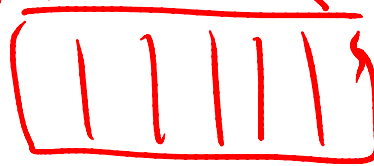


M2S40 Explaining PFS and SGAM contention in tempdb and how adding multiple files spreads the contention over multiple files, thus reducing it.

VLFs



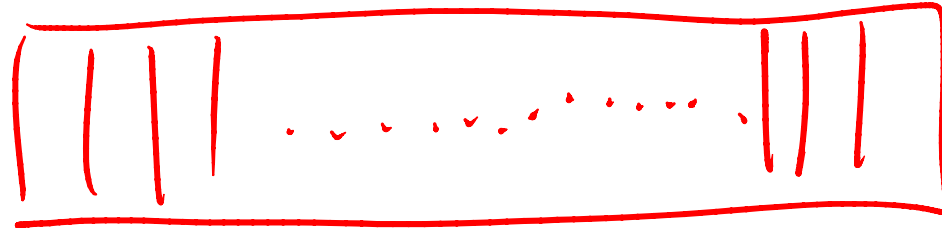
LOG_COMMIT
- TRAN
x ACT



SIZE - 60KB

LOG BUFFER

WRITELOG

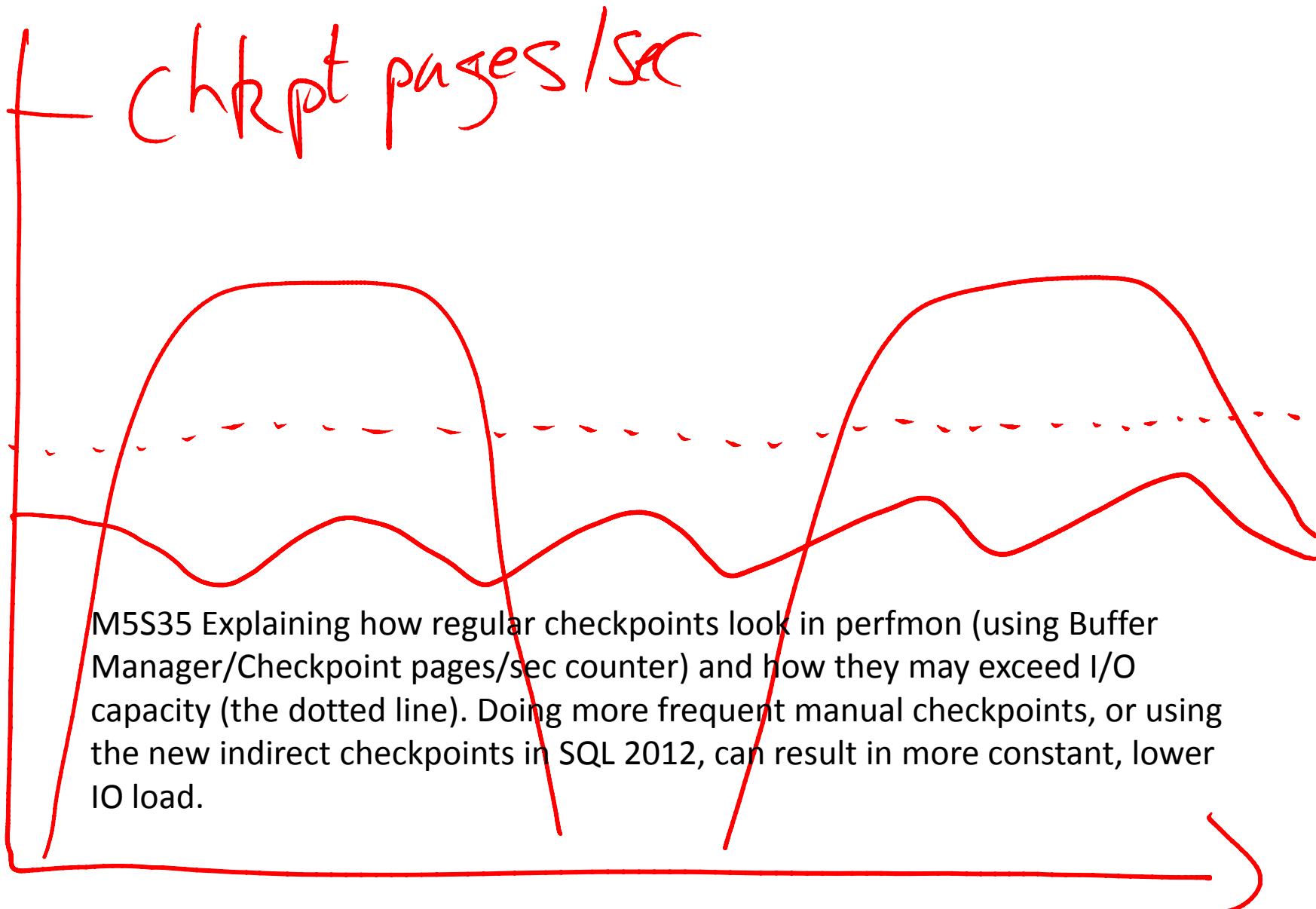


128

3840K

ASYNC I/O
32

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.

100,000

10

M5S60 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.

~~INS
INS
INS~~

F - PAGE
F - PAGE

10

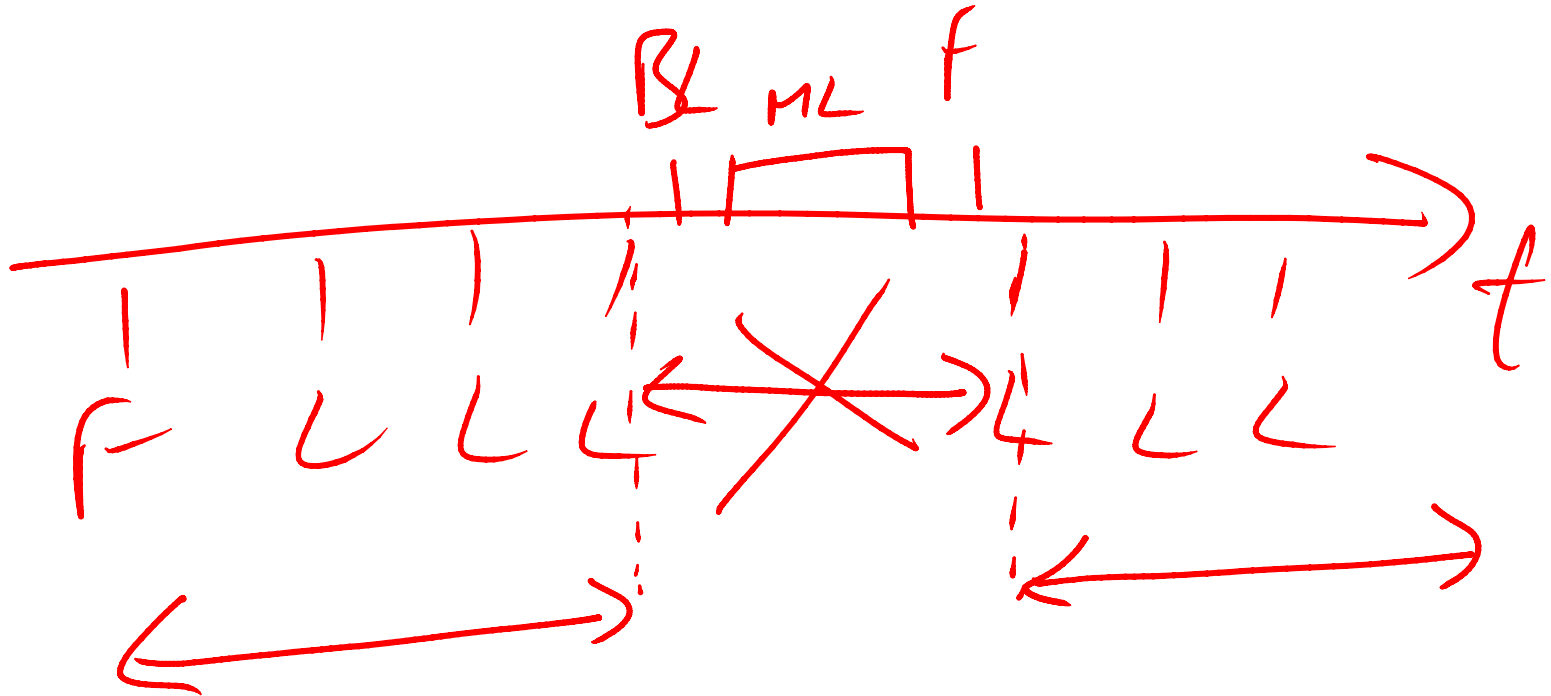
This is a slide that was lost when PPT crashed. It showed the timeline of a full backup:

1. Backup begins with a checkpoint at time t_0
2. Data-reading portion of the backup ends at time t_1
3. Log-reading portion of the backup ends at time t_2

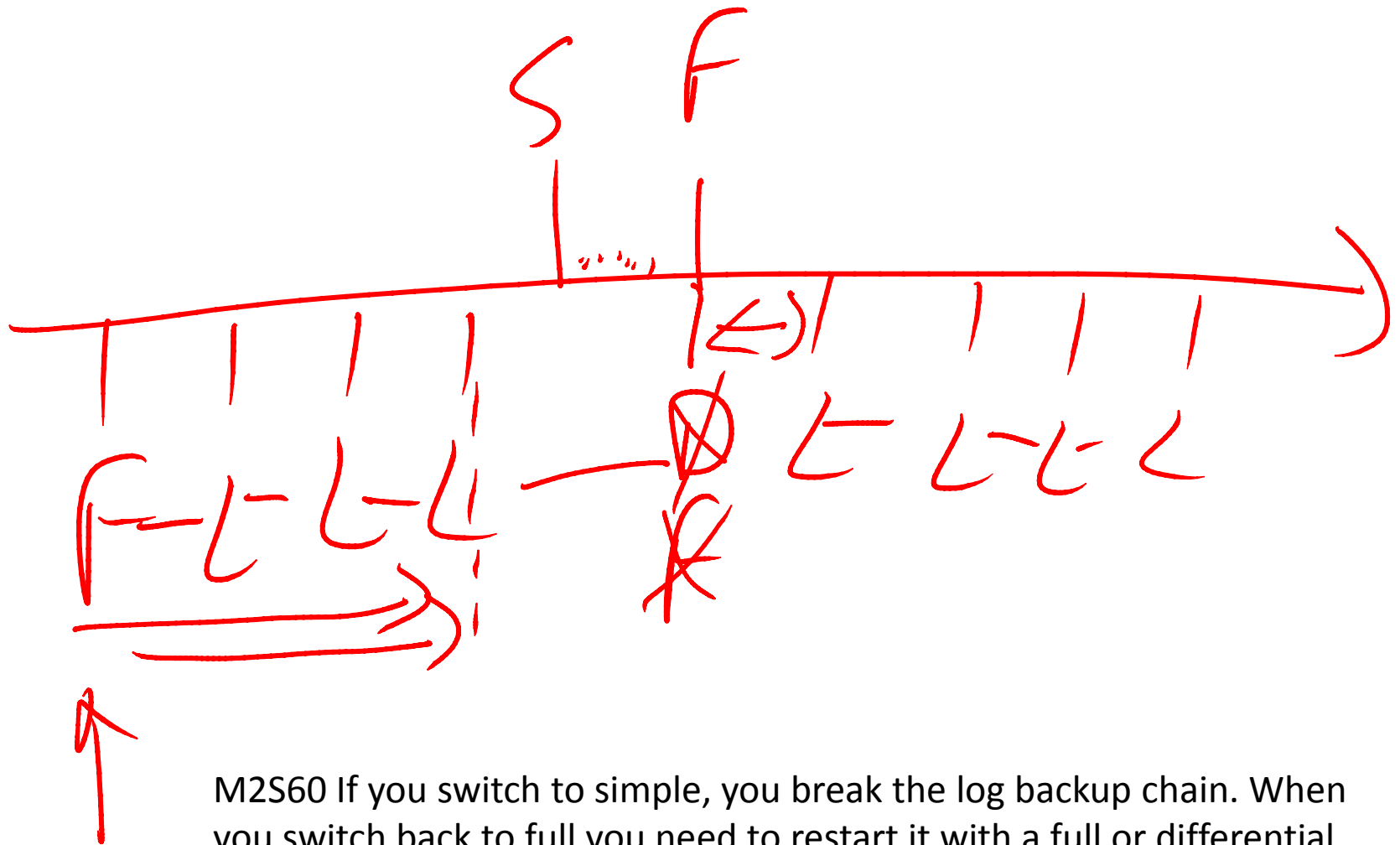
When you restore the backup, you get the database as of time t_1 .

Common misconception is that you get the database as of time t_0 or t_2 .

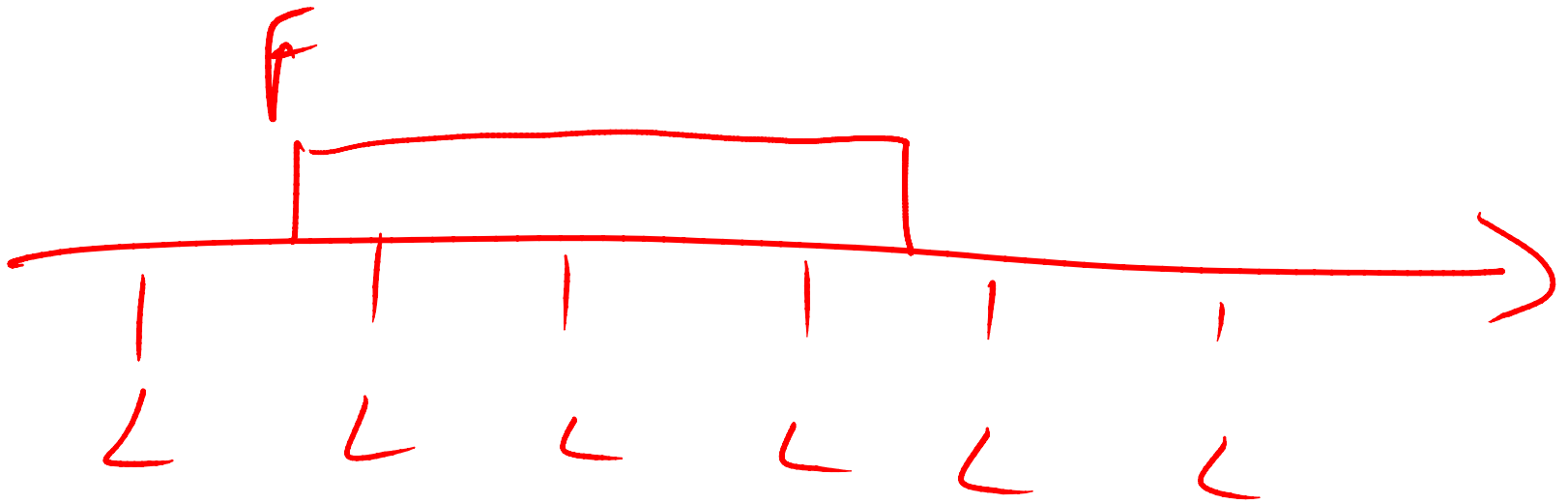
If you want to restore to a point in time between t_0 and t_1 , you cannot use that full backup – you need to use the previous one and then log backups after that.



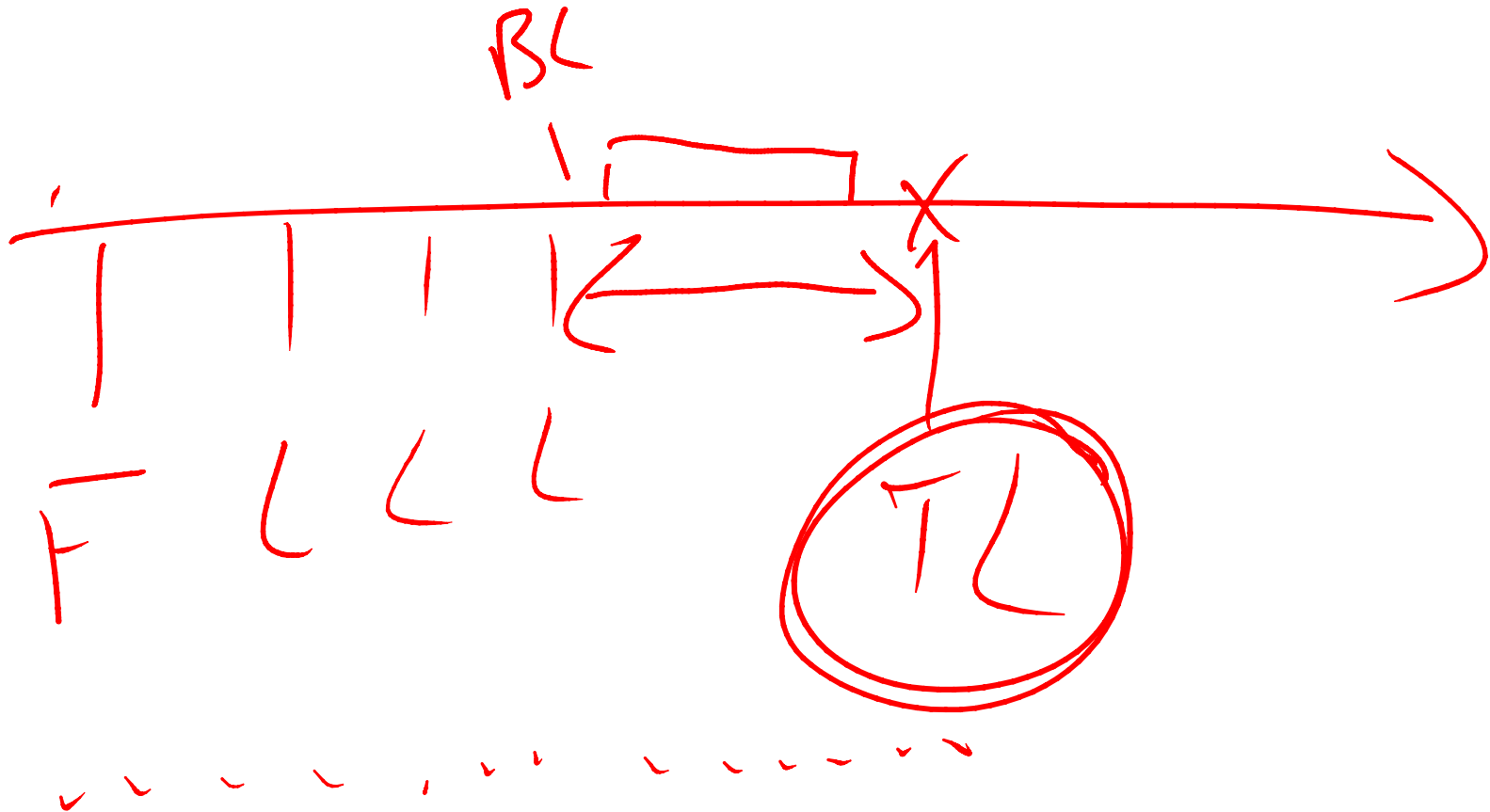
M5S60 If a log backup contains a minimally-logged operation, it cannot be used to do a point-in-time restore using STOPAT for any point in time between the beginning and end of the backup. It can be used as part of a restore sequence as normal though.



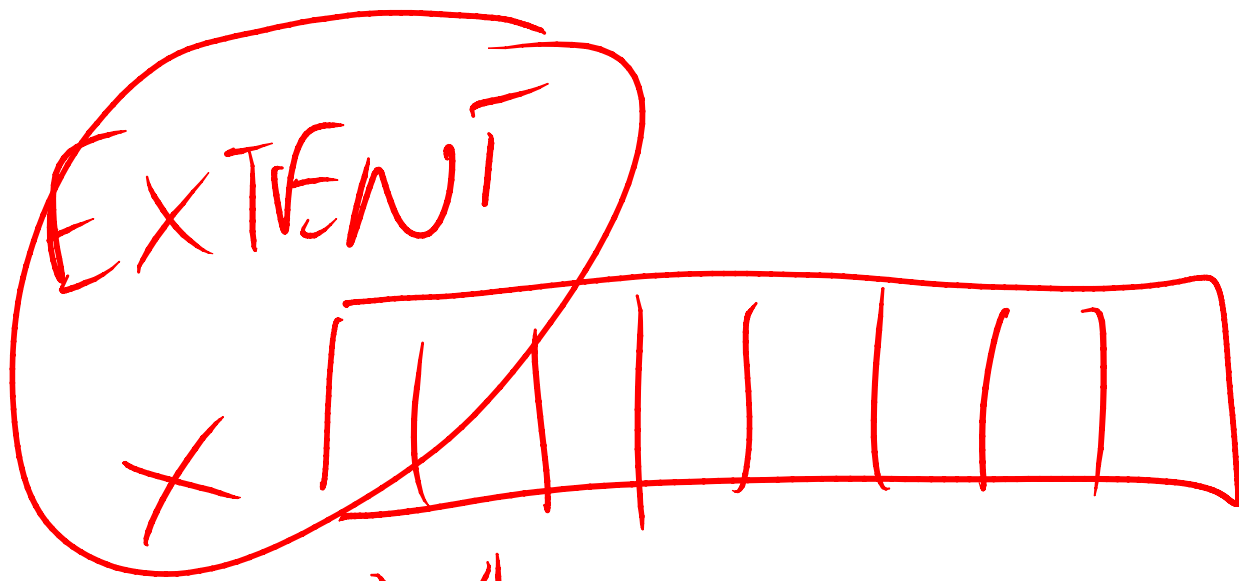
M2S60 If you switch to simple, you break the log backup chain. When you switch back to full you need to restart it with a full or differential backup.



M5S42 It's possible for concurrent log backups to occur with a full backup from 2005 onward. Log clearing cannot happen though. It's deferred until the full backup completes, giving the impression that a full backup clears the log.



M5S60 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.

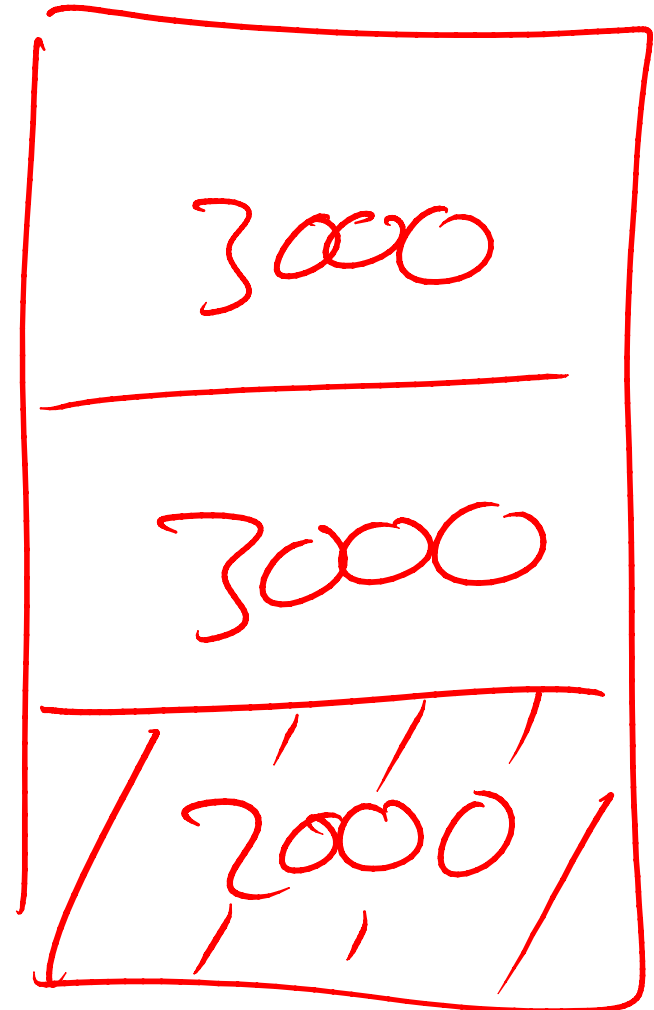
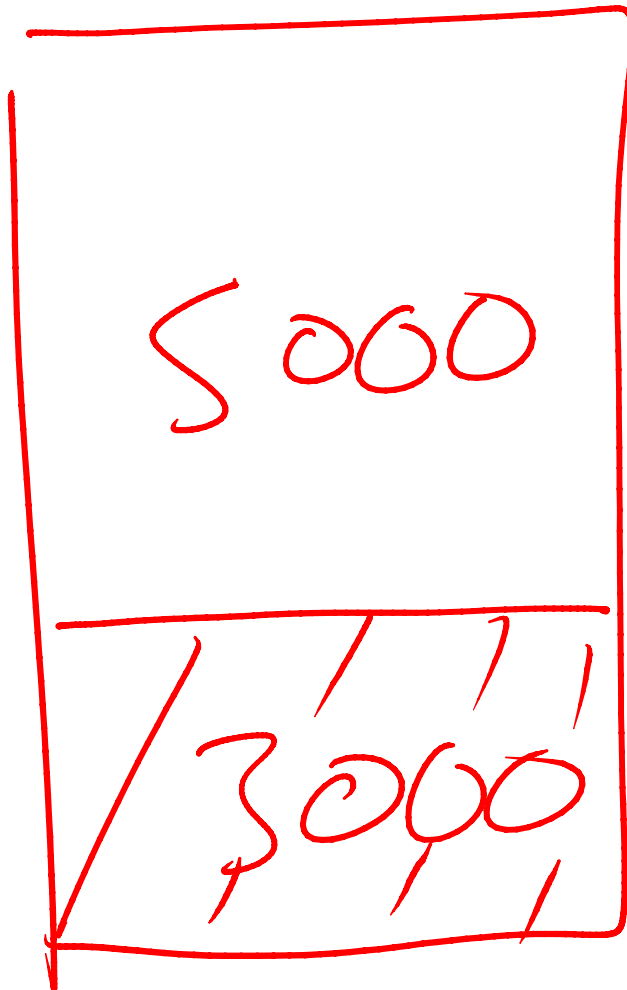


PROBE

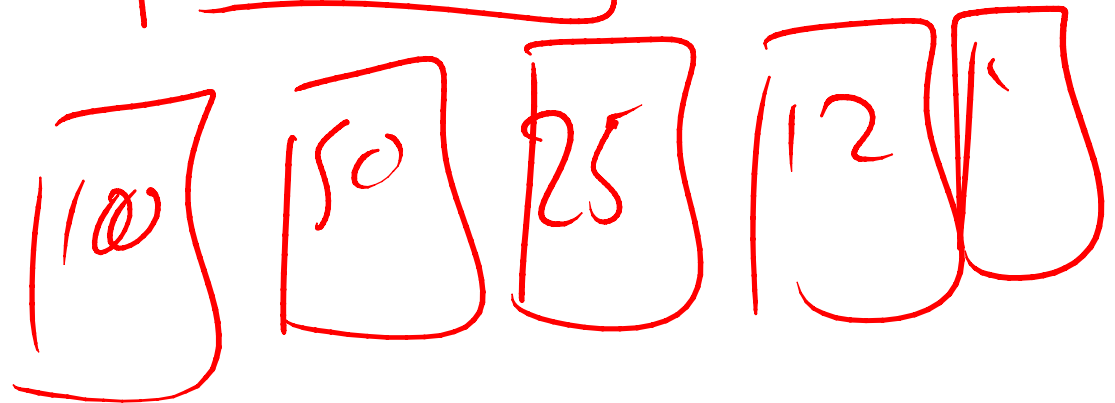
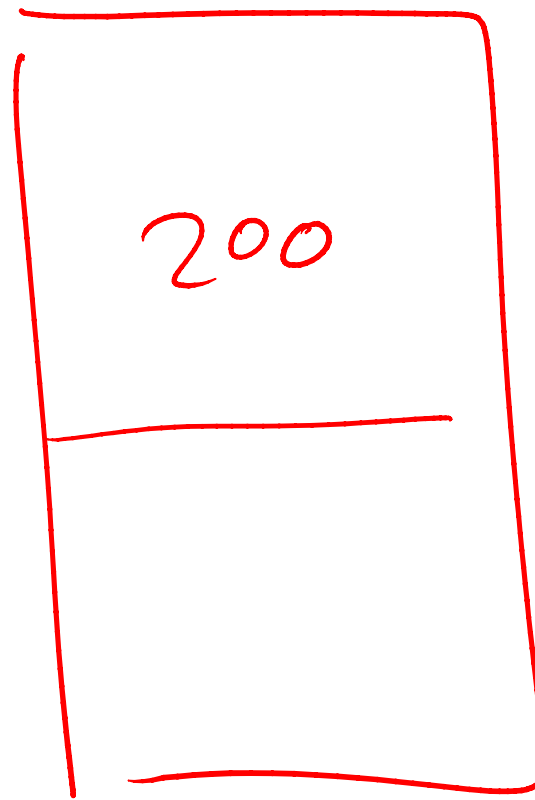
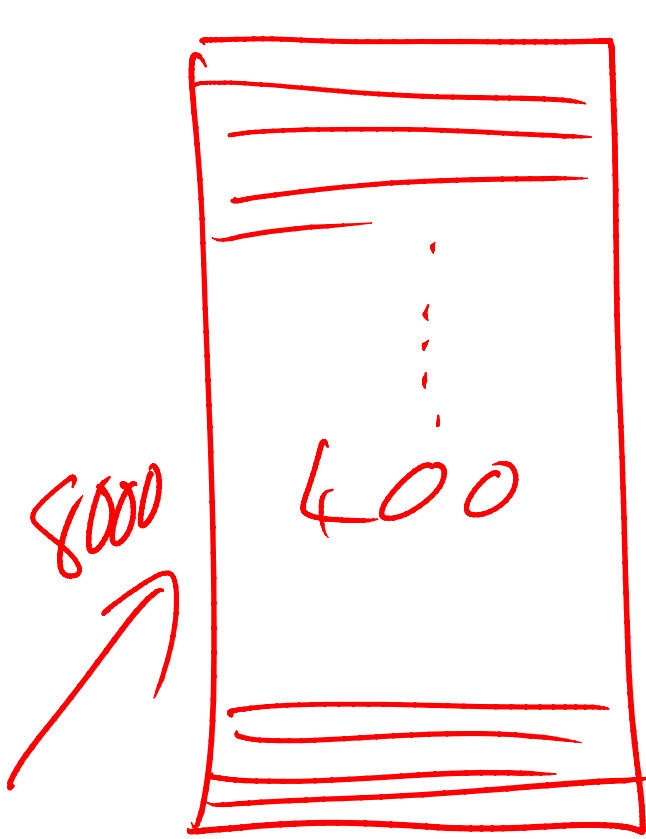
M5 Explaining part of the deferred-drop functionality. Before an extent can be deallocated, an X lock is acquired on it and all 8 page locks are probed (instant acquire X lock and release) to make sure nothing else has them locked.



M5S25 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.

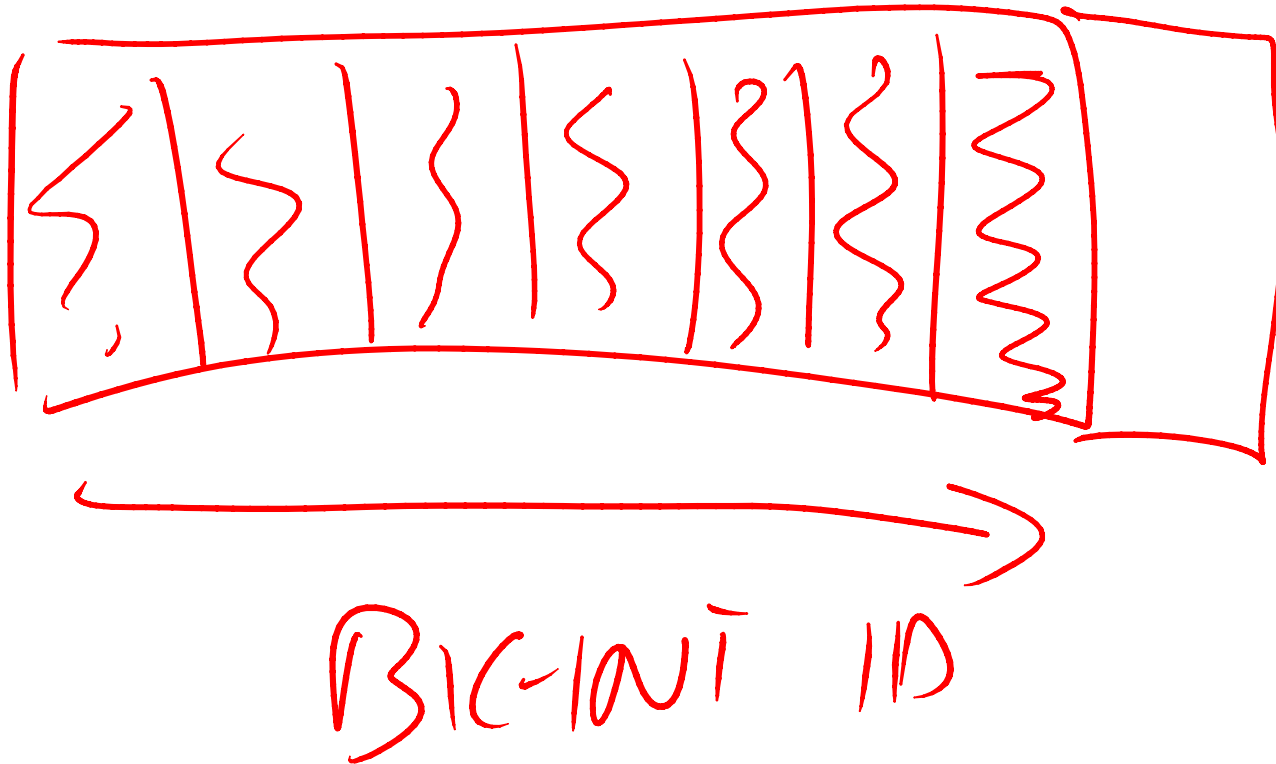


M8S13 Explaining how fixed-size large rows leads to low page density and wasted space.



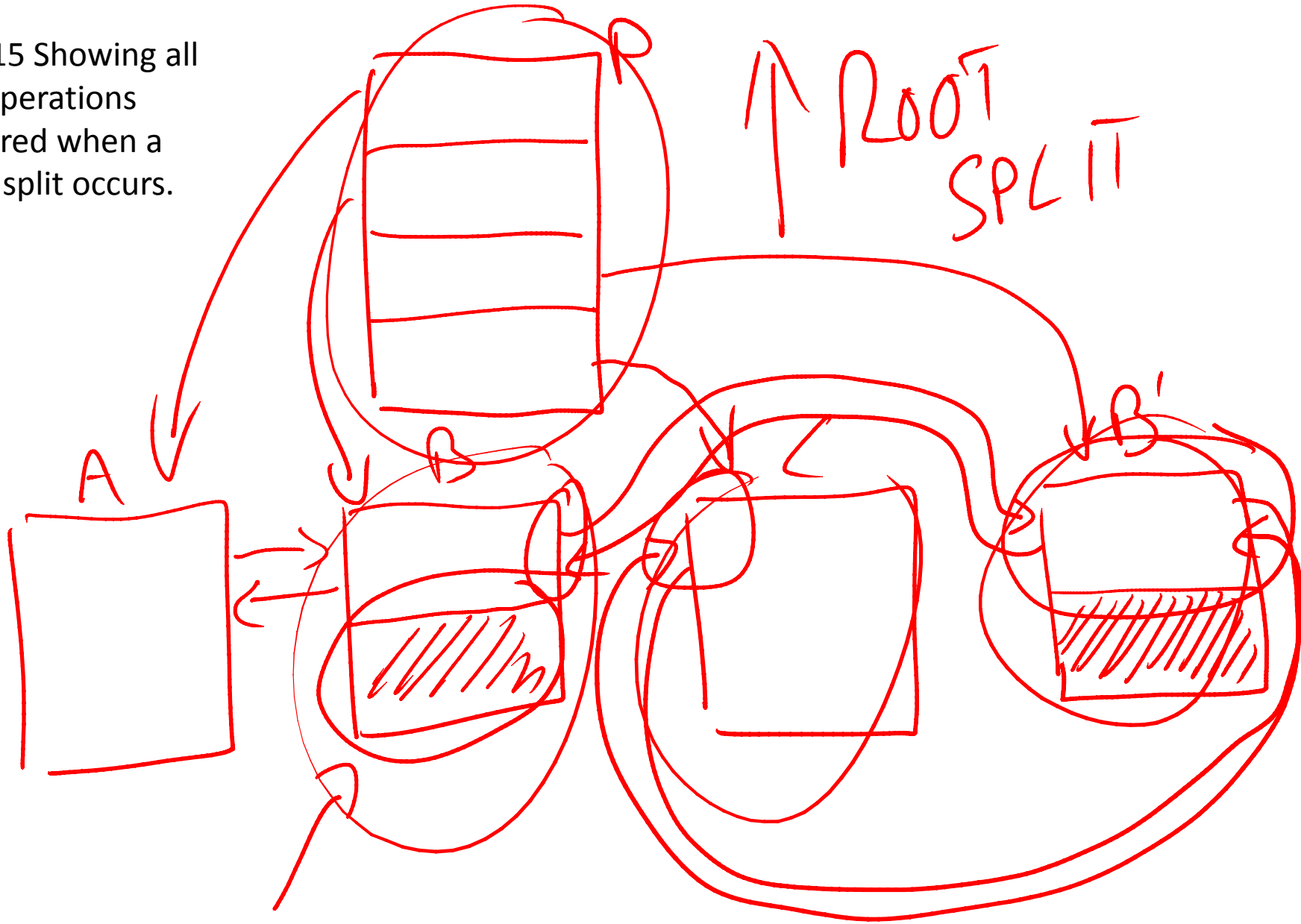
M8 Explaining how a page split might turn into multiple page splits. E.g. Insertion of 8000 byte record into page with 400×20 byte records.

LOP_DELETE_SPLIT



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.

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



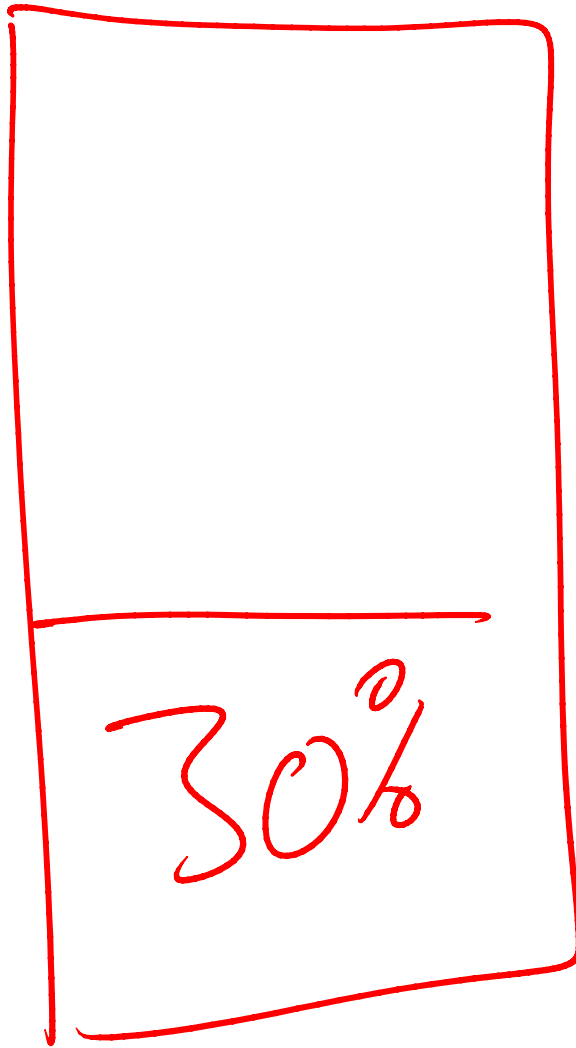
	NO SPLIT	SPLIT	X
1000	1256	7572	x5.5
100	356	6764	x18
10	268	11164	x47

M8S22 The numbers from the page split logging cost demo.

COMMENTS		MEMBERID	
1	PAUL	GREE	JOAN

KIMBERLY

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



$$FF = 70$$

M8S25 To leave 30% free space, set the fillfactor to 70.

ITB



M8S37 Explaining staggered index maintenance with database mirroring. See <http://bit.ly/IS04W5> for more explanation