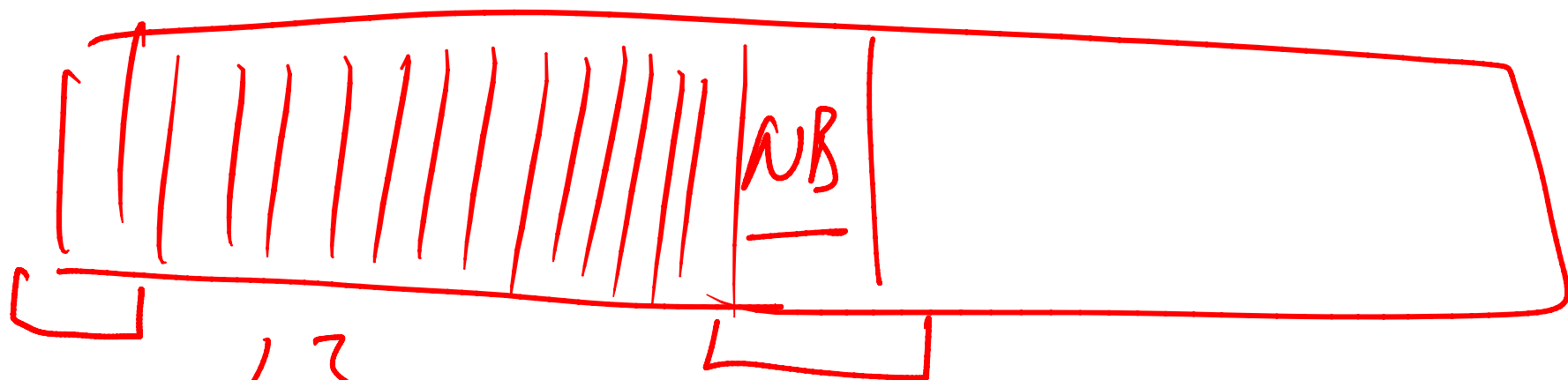




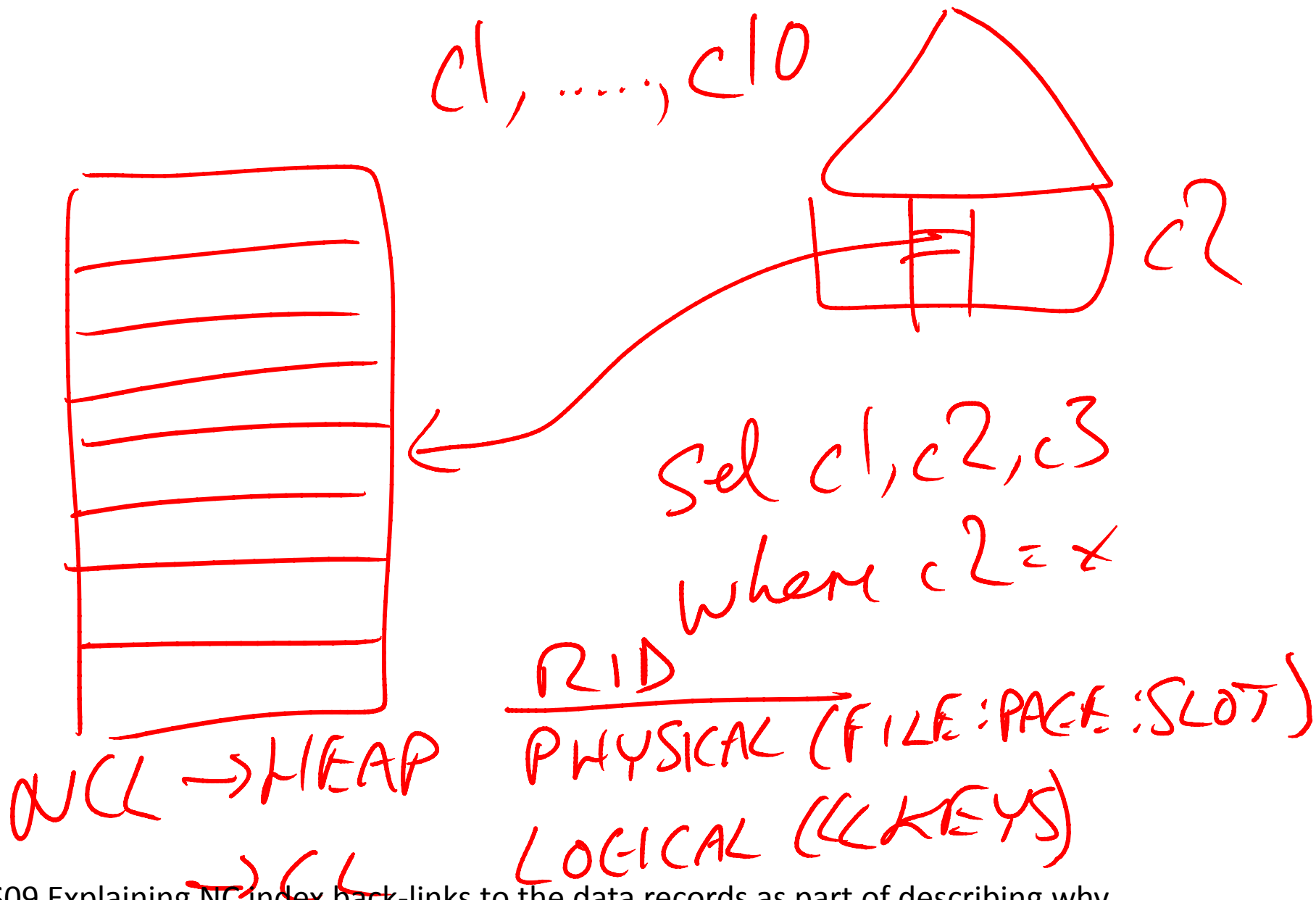
L3

CACHE LINES

64b
128b

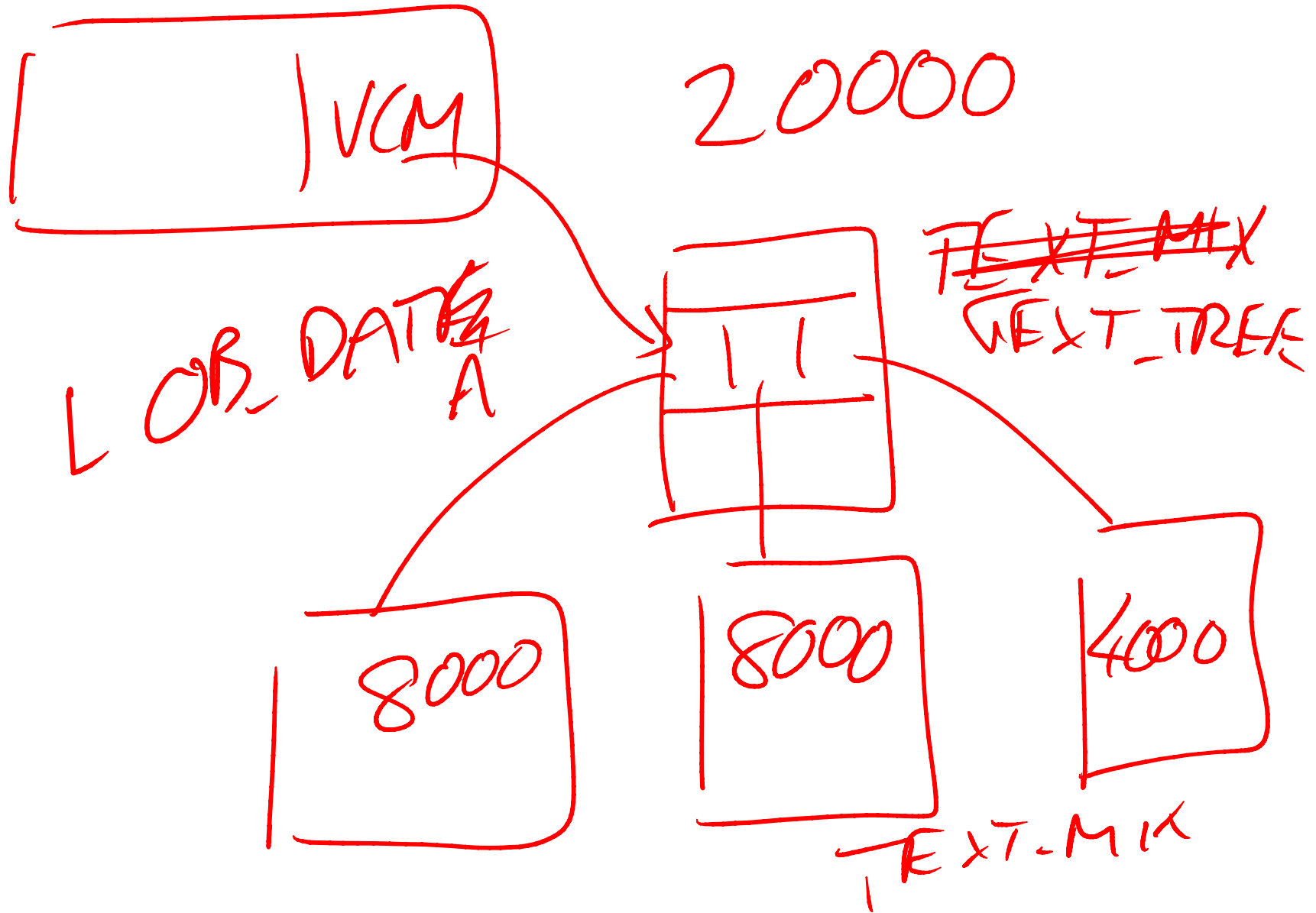
INVALIDATION

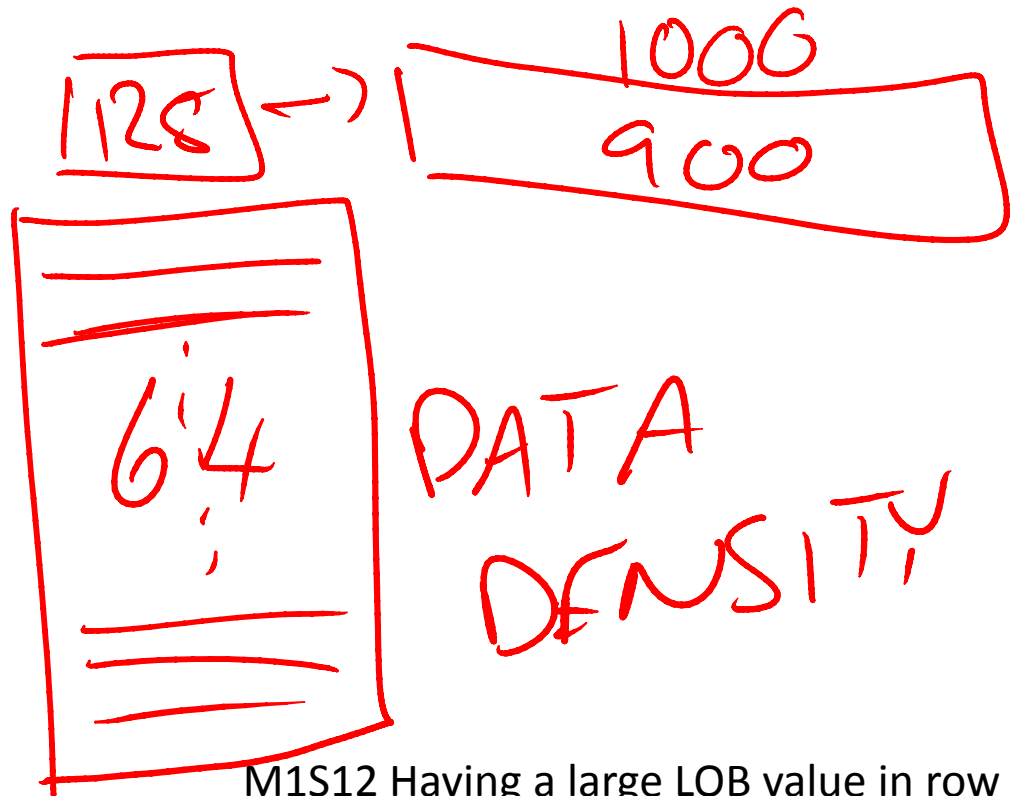
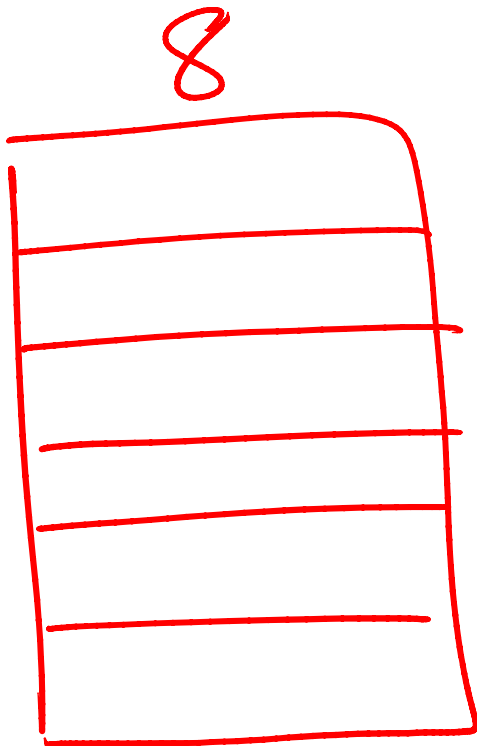
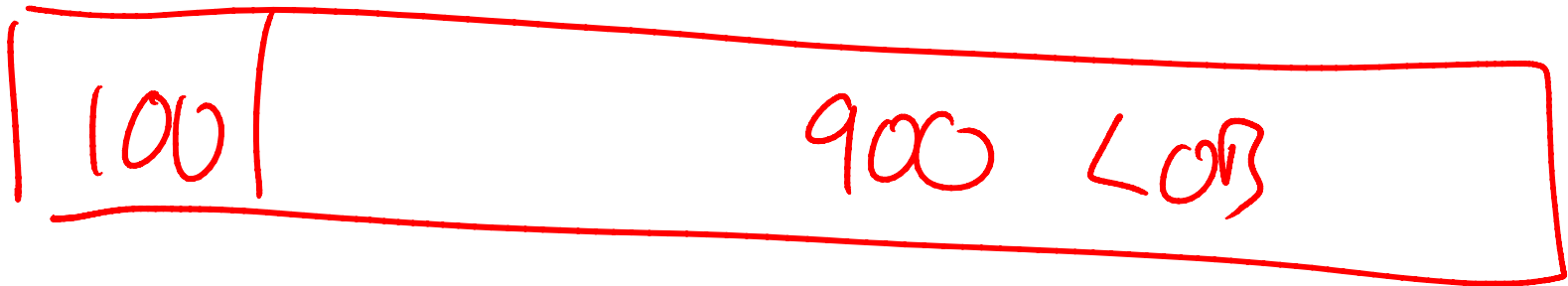
M1S06 Discussing
why the NULL
bitmap is a perf
optimization by
avoiding having to
read record parts
into the CPU's L3
cache



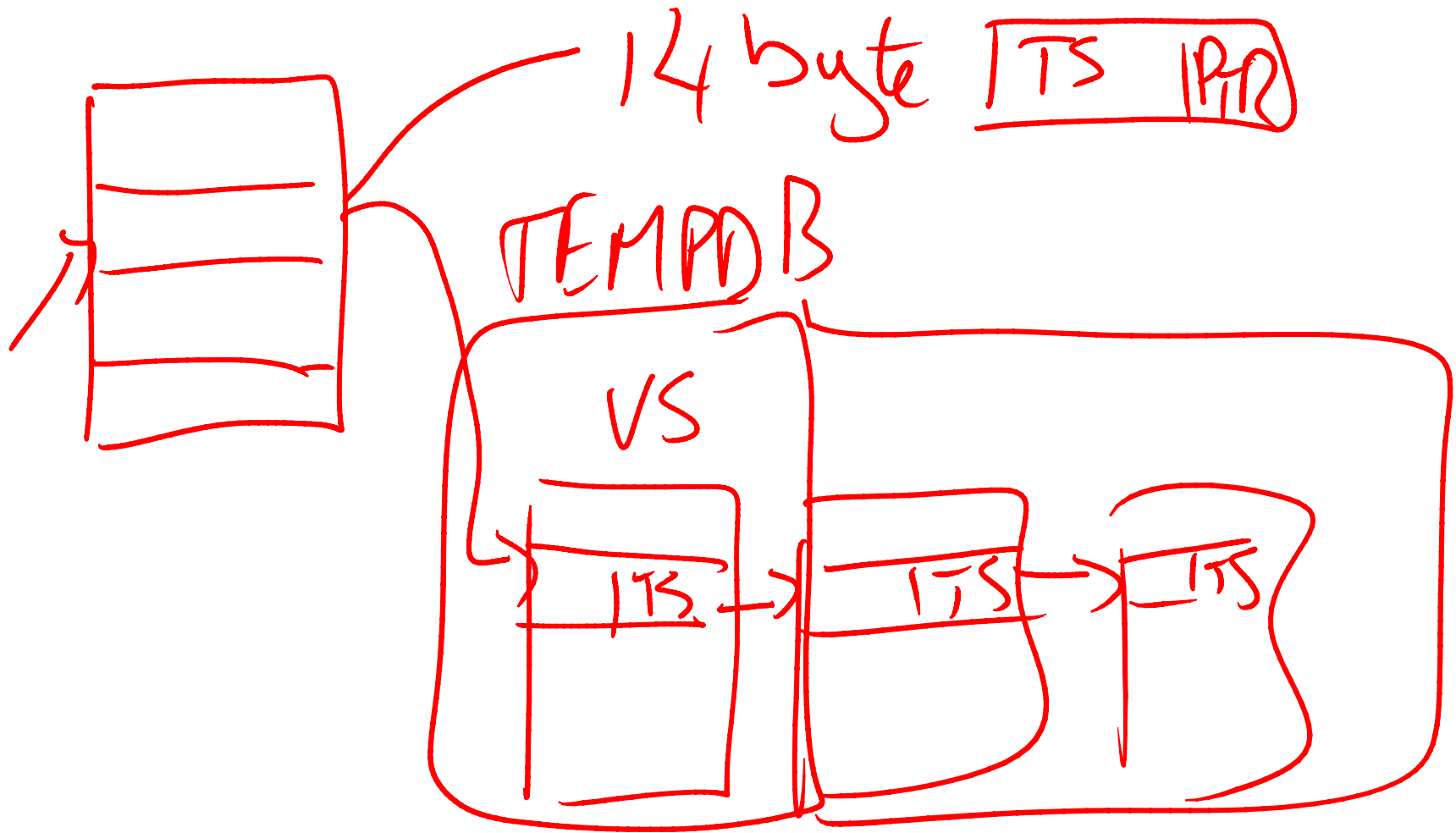
M1S09 Explaining NC index back-links to the data records as part of describing why forwarding/forwarded records exist. Without them, moving a heap row would entail changing all NC index records with a back-link to that row.

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



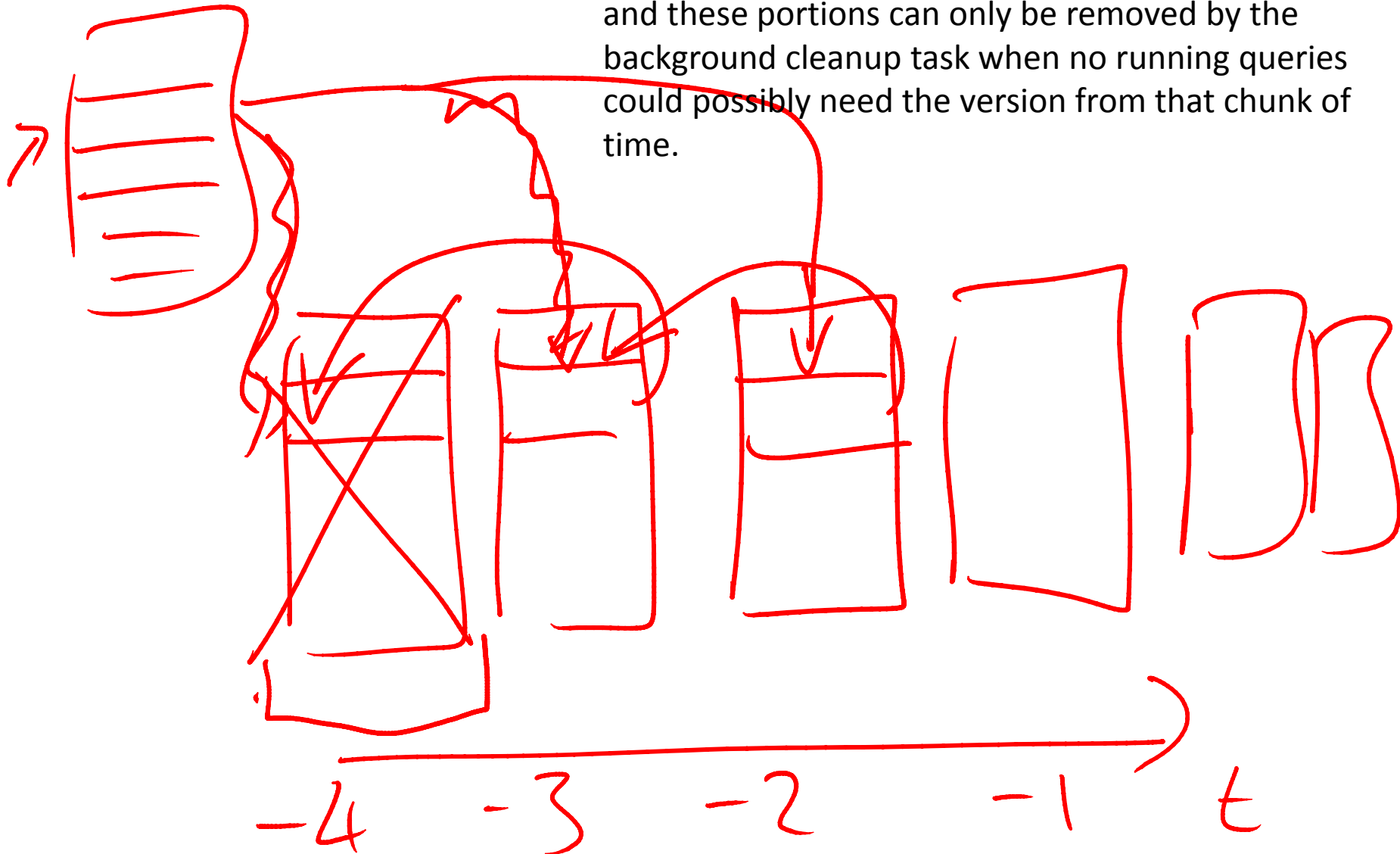


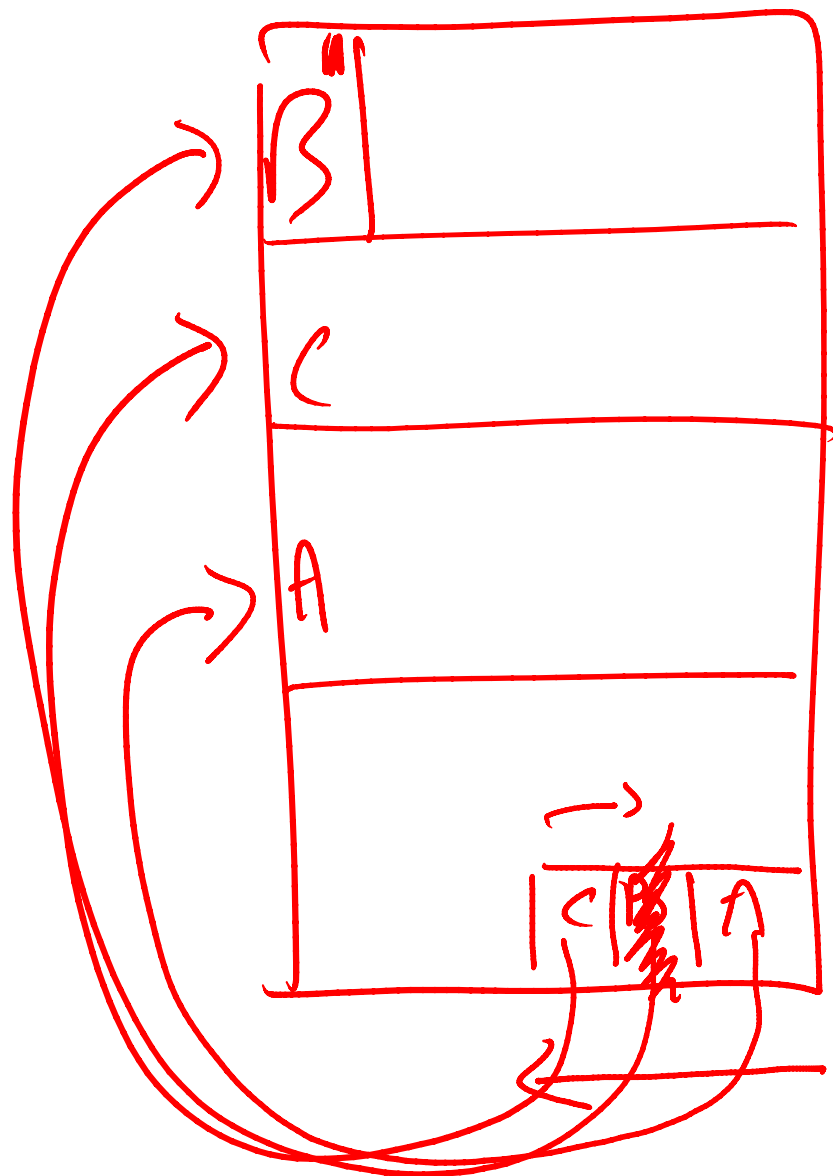
M1S12 Having a large LOB value in row that's not used much makes for large rows and low data density. Choose how to store LOB data wisely.



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 -
CHOST

SLOT ARRAY

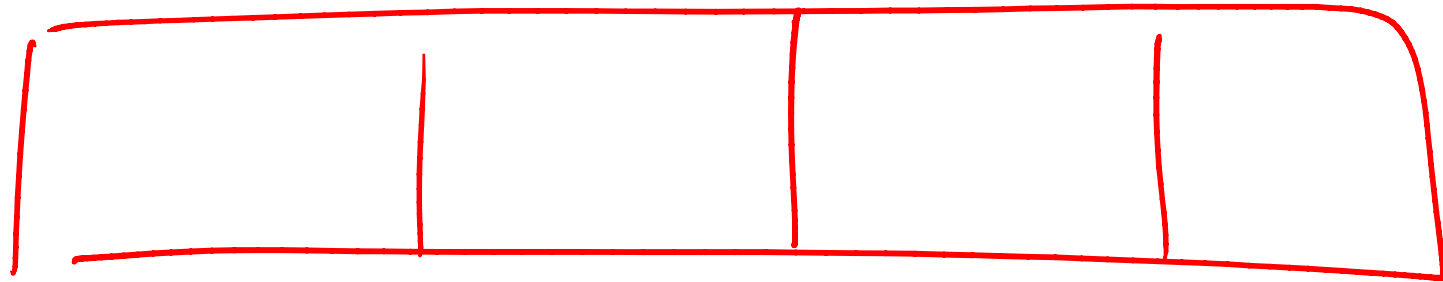
M1S14 When a record is deleted, it's marked as a ghost record. The ghost cleanup task does not physically delete the records – it removes the slot array entry that points to the record on the page. I.e. the record is unmarked as being a record and the area on the page is now free space – that just happens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

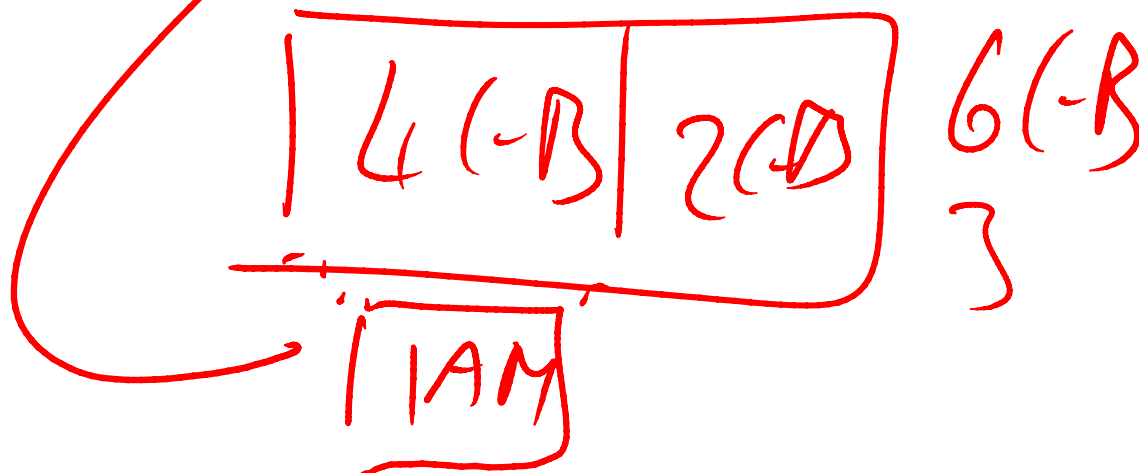


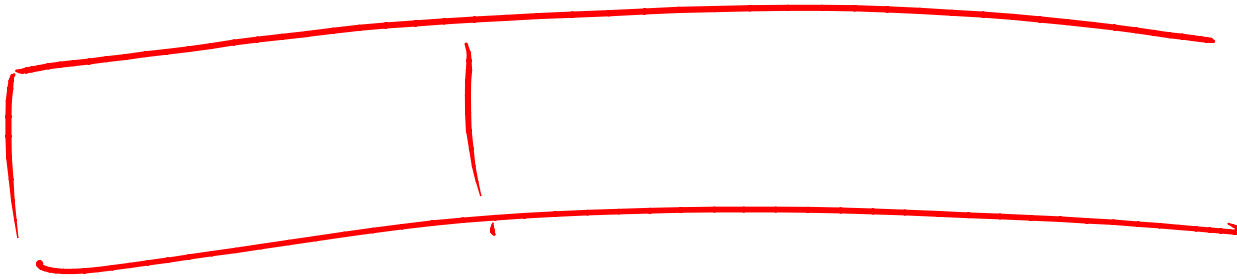
M1S21 Explaining the difference between mixed and dedicated extents. Mixed extent is one in which the 8 pages could be allocated to 8 different objects. Dedicated extent is one where all 8 pages are reserved for exclusive use of a single object.

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



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.





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

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

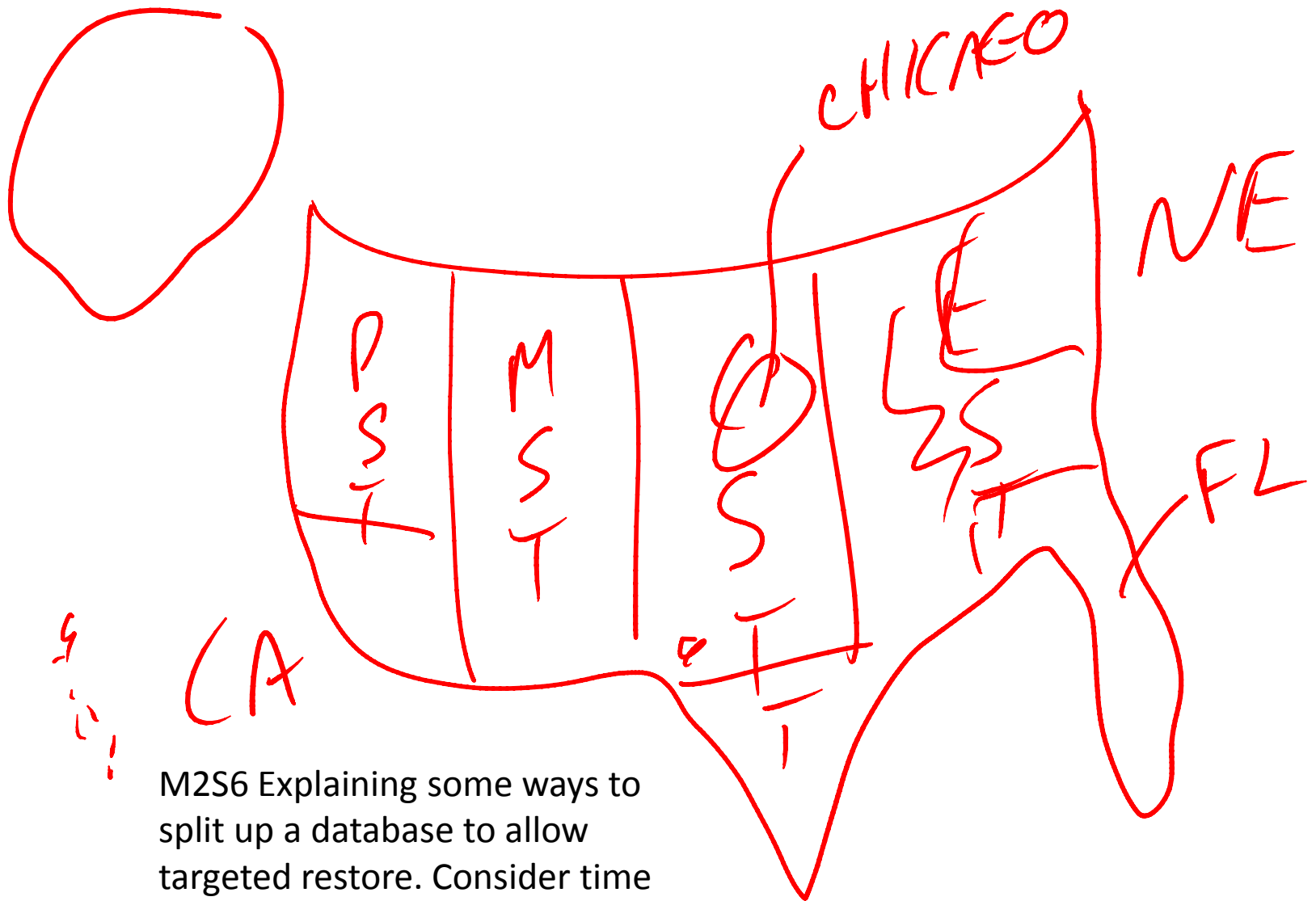
2014

2013

2012

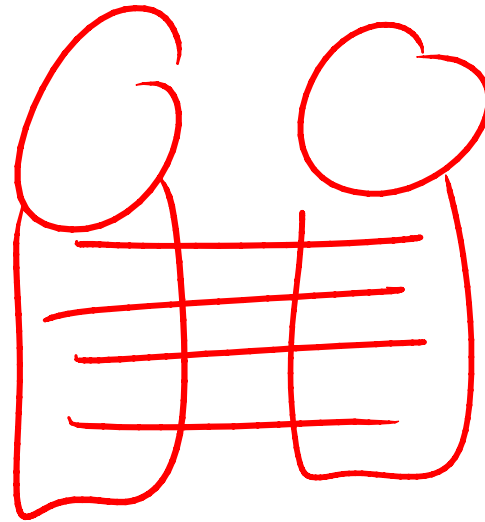


M2S6 Explaining some ways to split up a database to allow targeted restore. On the left, auto parts sales system – makes sense to bring parts and NSW online first.



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

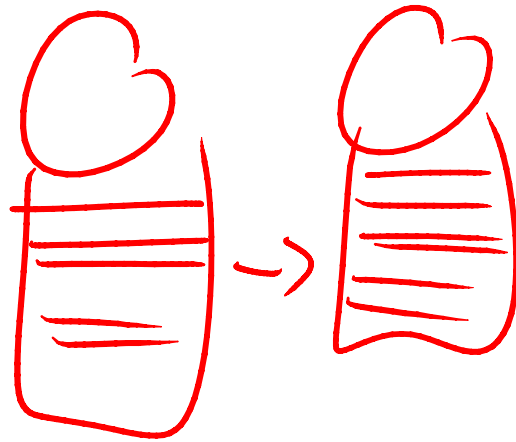
STRIPING



R0

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

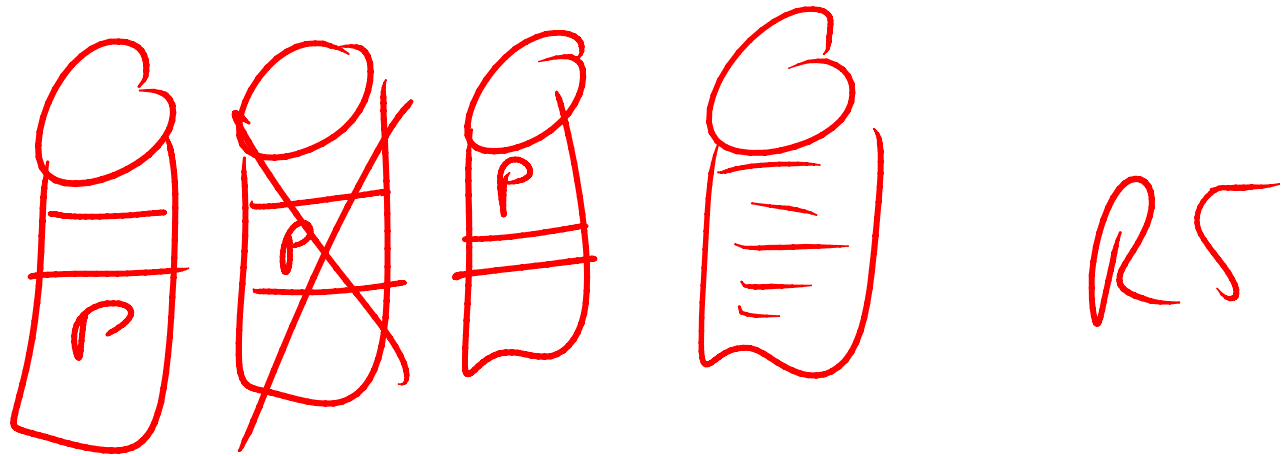
MIRRORING



RA1

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

NOTATING PARITY

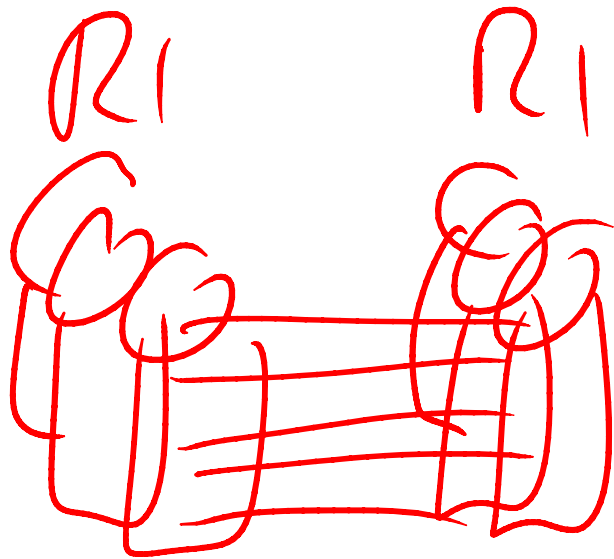


http://en.wikipedia.org/wiki/Standard_RAID_levels

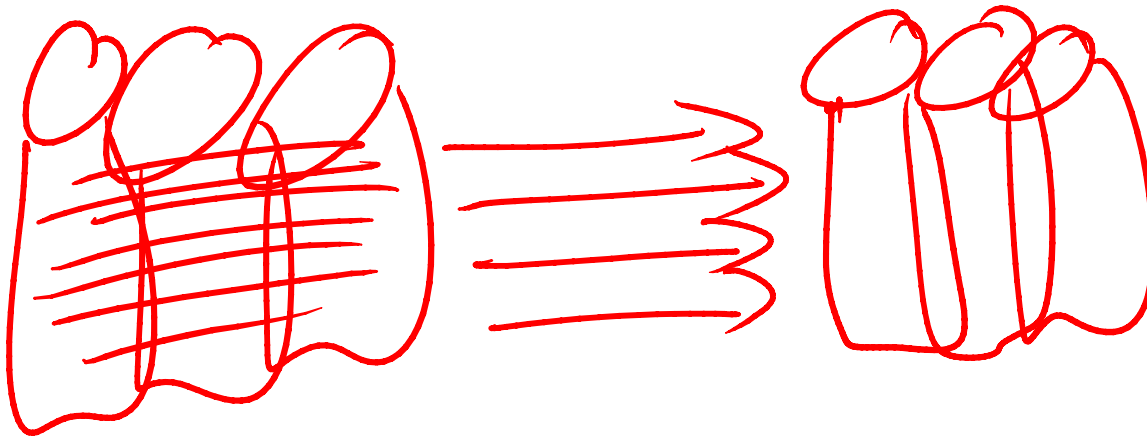
$$R10 = R1 + G$$

= MIRRORING

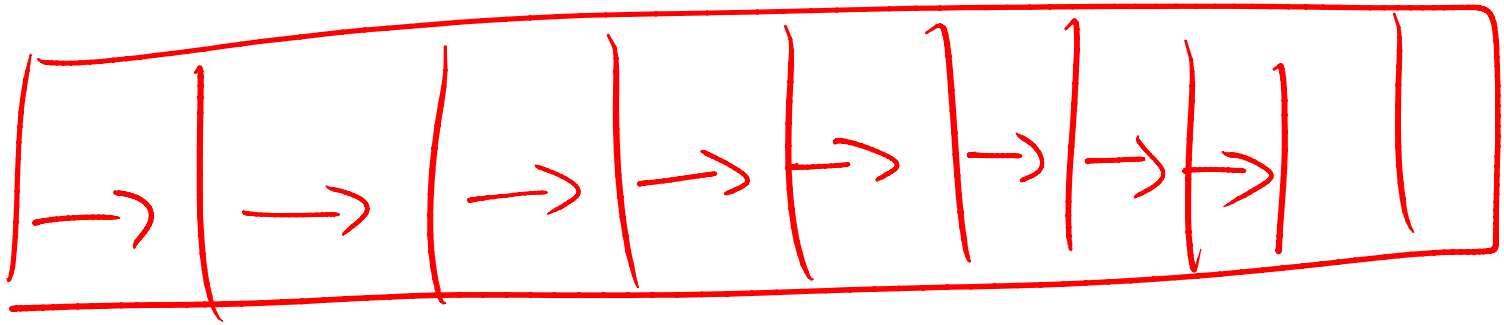
+ STRIPING



$R O + 1$

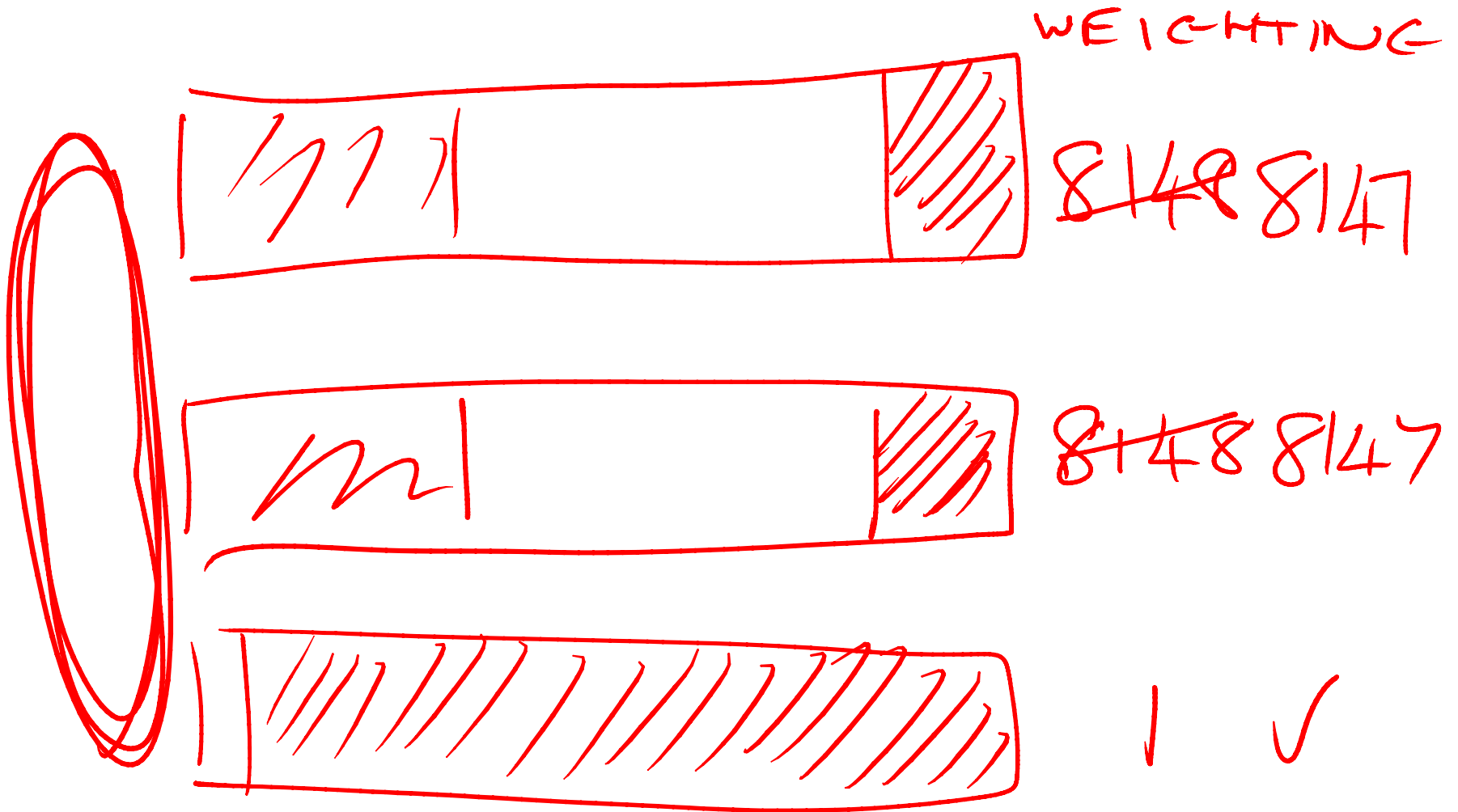


http://en.wikipedia.org/wiki/Standard_RAID_levels

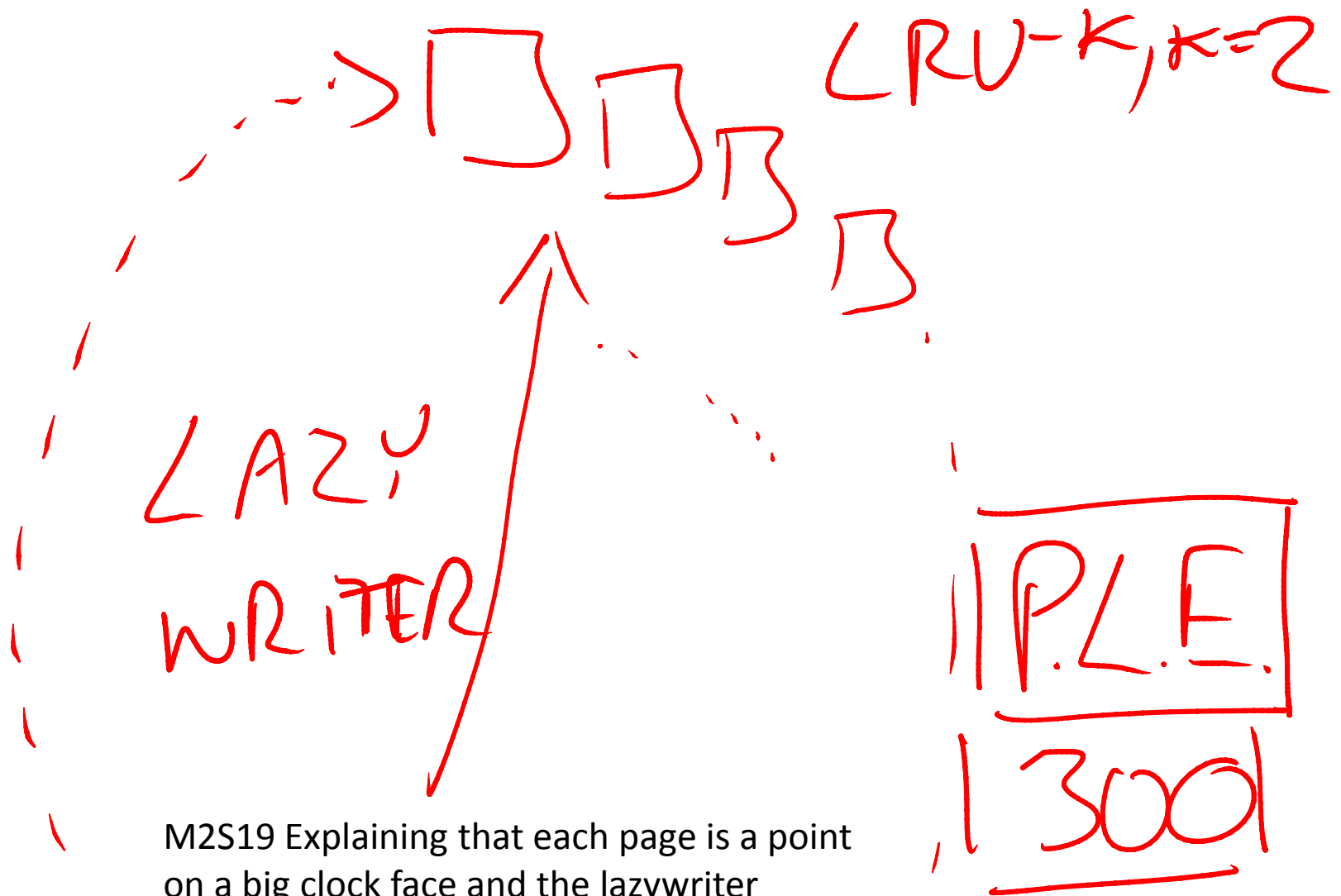


△:

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



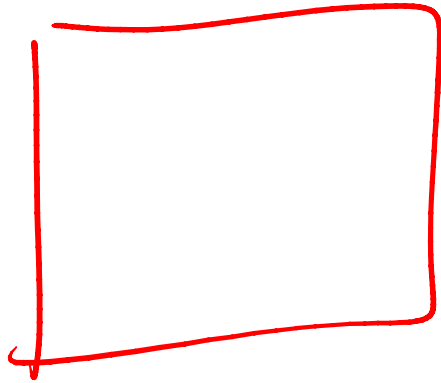
M2S18 Explaining round-robin allocation and proportional fill. Each file has a proportional fill weighting. Allocations happen proportionally more frequently from files with more free space.



M2S19 Explaining 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.

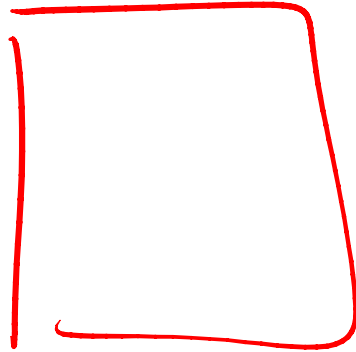
PLE = 300 is a nonsense threshold for tuning. See next slide.

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

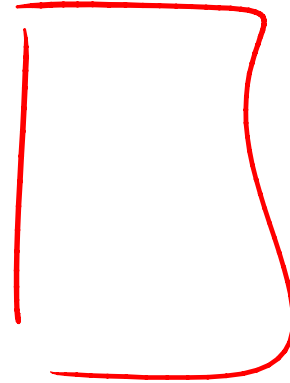


PLE
4000

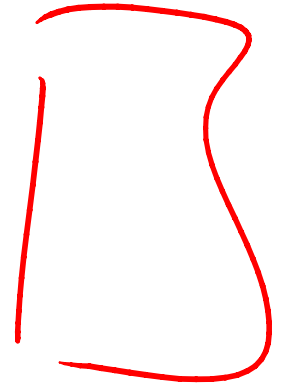
4000



PLE
4000

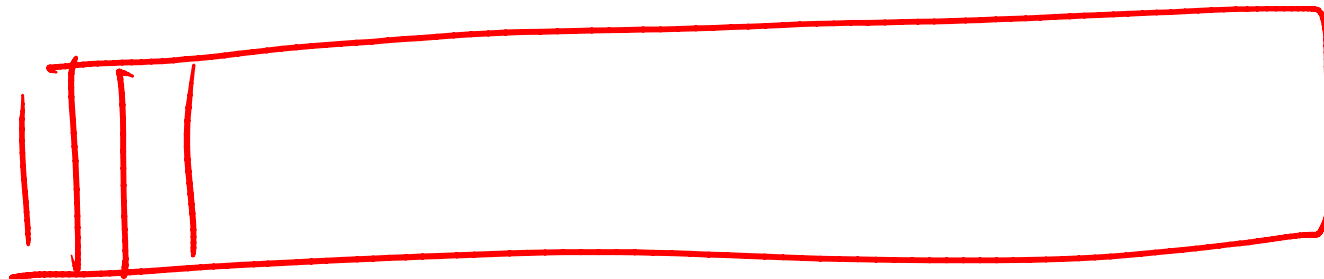
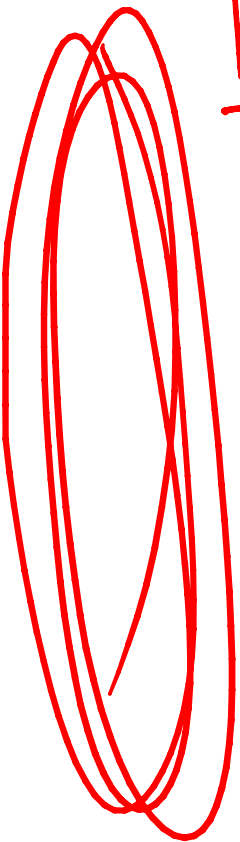


PLE
4000



PLE
1000

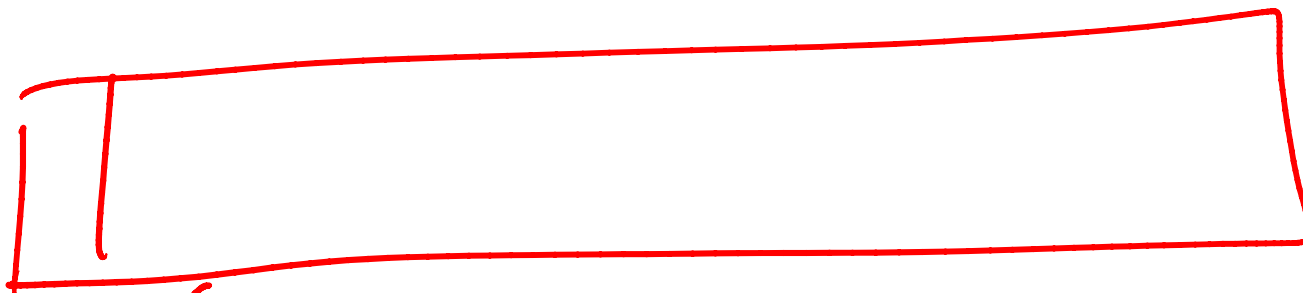
3625



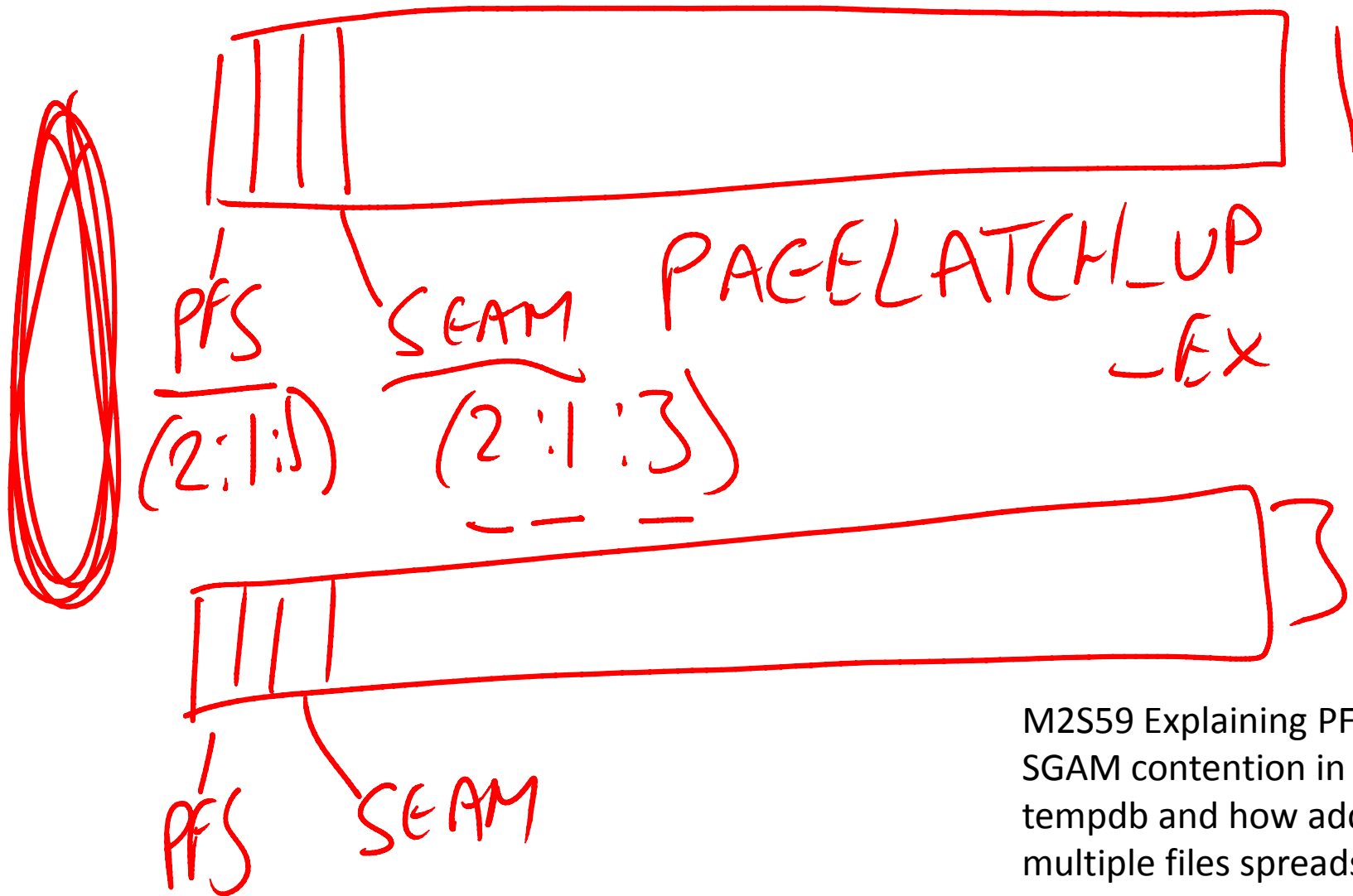
1118

PFS ~~SCAN~~

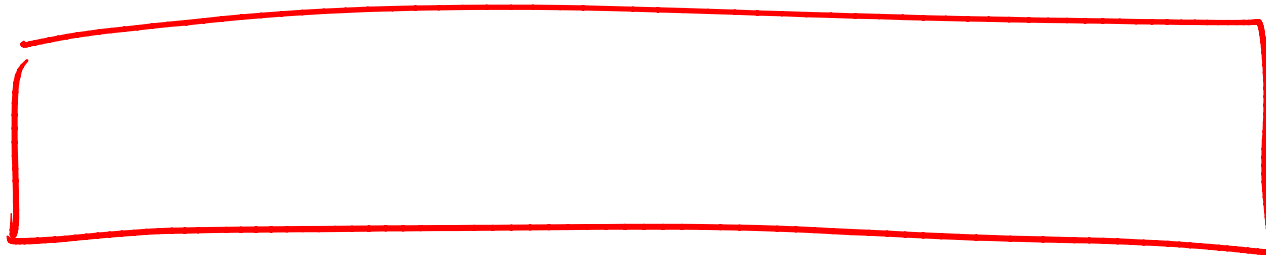
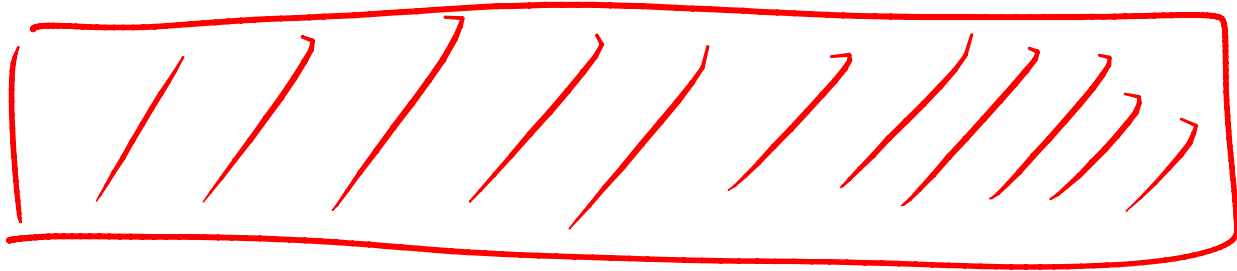
PAGE LATCH_EX
_UP



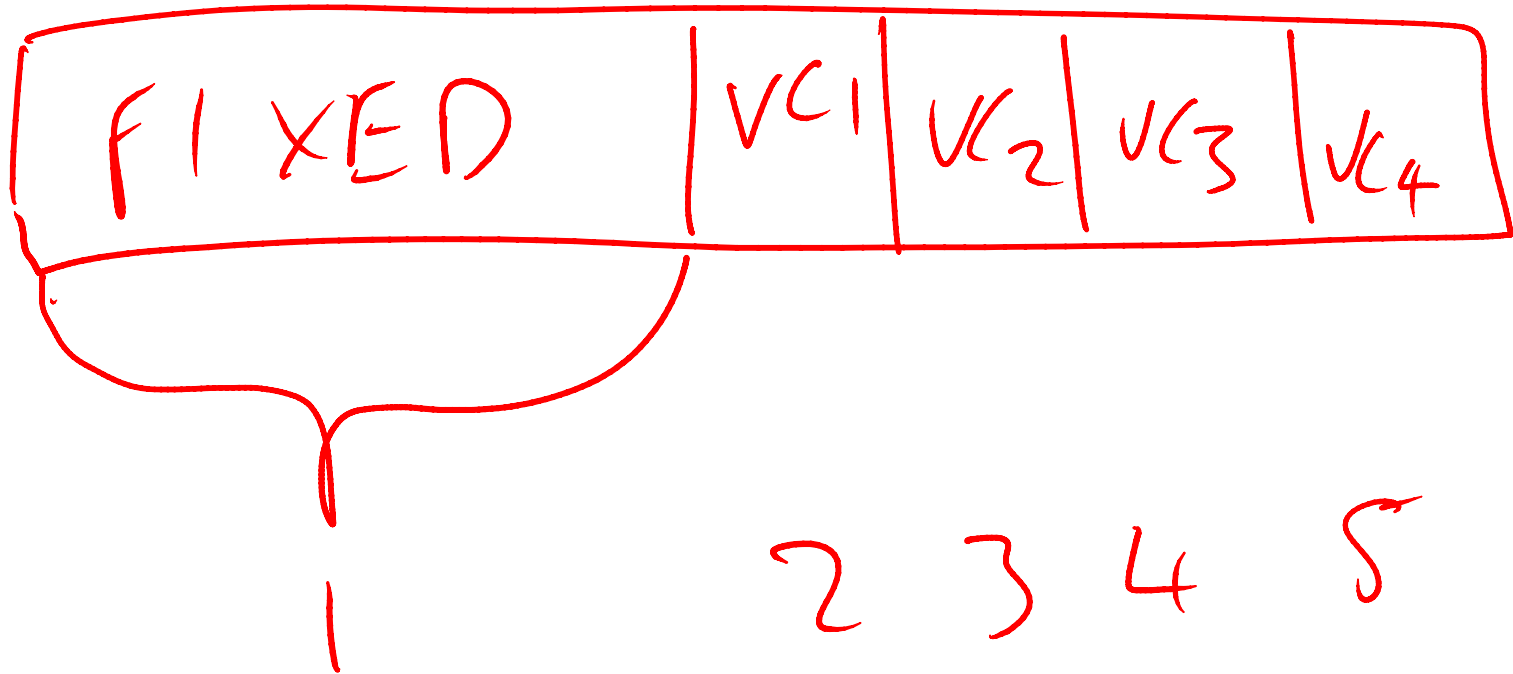
PFS

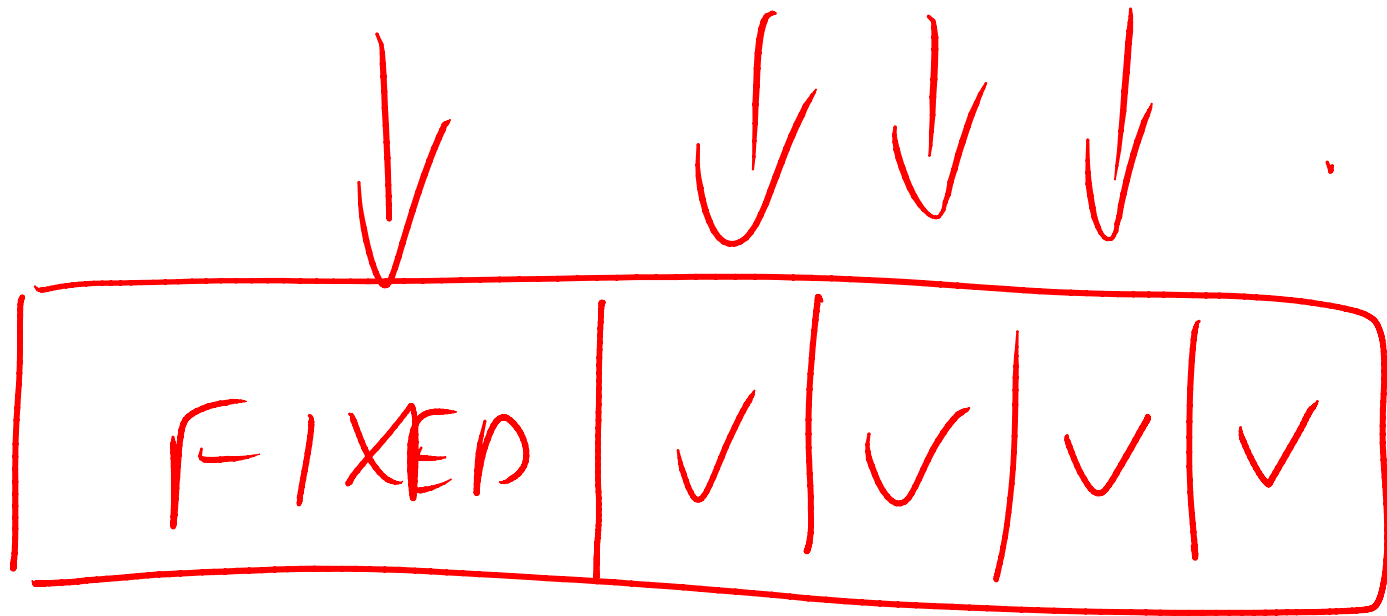


M2S59 Explaining PFS and SGAM contention in tempdb and how adding multiple files spreads the contention over multiple files, thus reducing it.



M2S61 You might add a second tempdb data file to alleviate contention and see no gain in perf. That could be because the first file is very full, and the new one very empty, so proportional fill makes all the allocations come from the new file -> no drop in allocation contention.

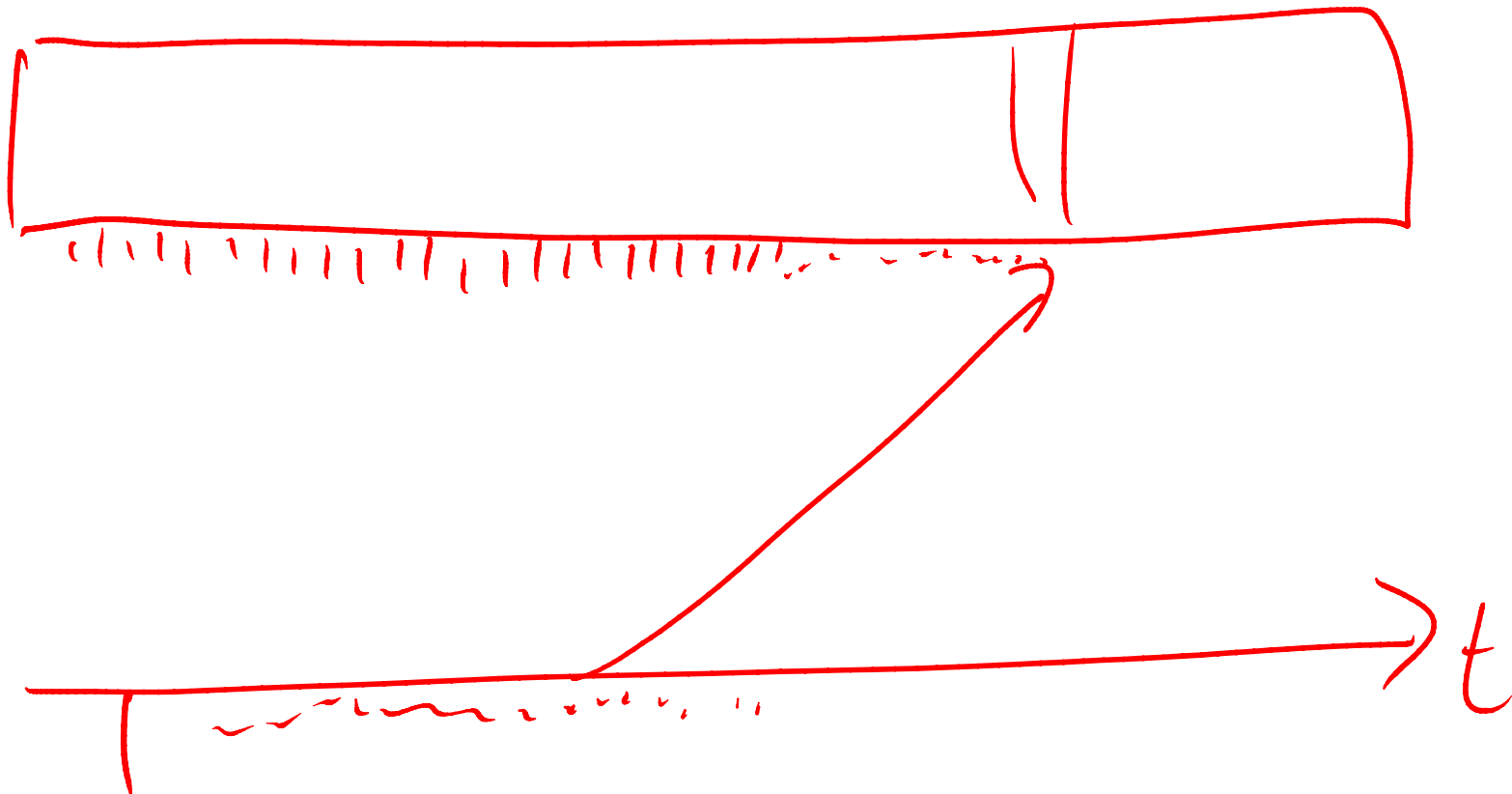




LOP_MODIFY_ROW

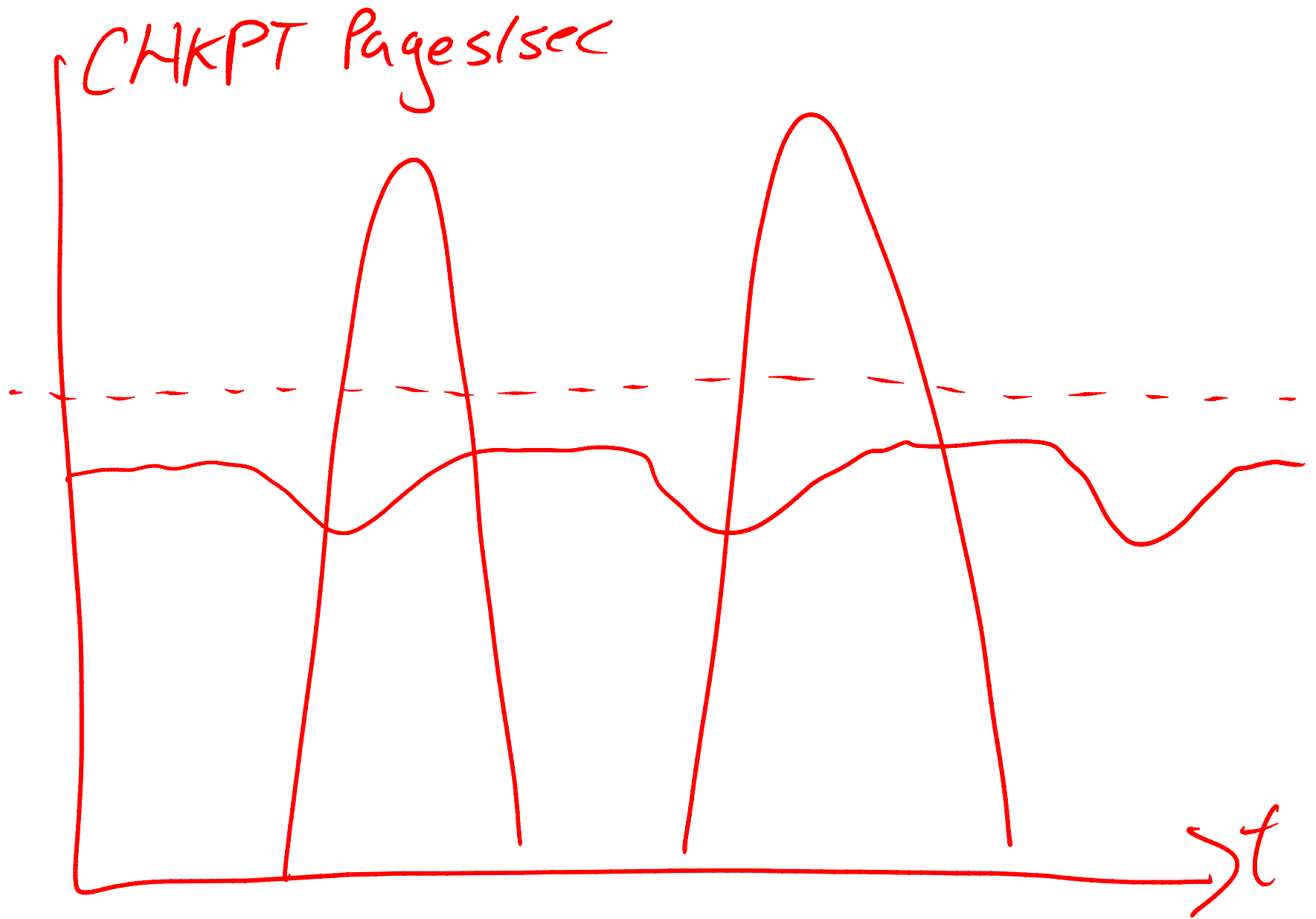
LOP_MODIFY_COLUMN

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)

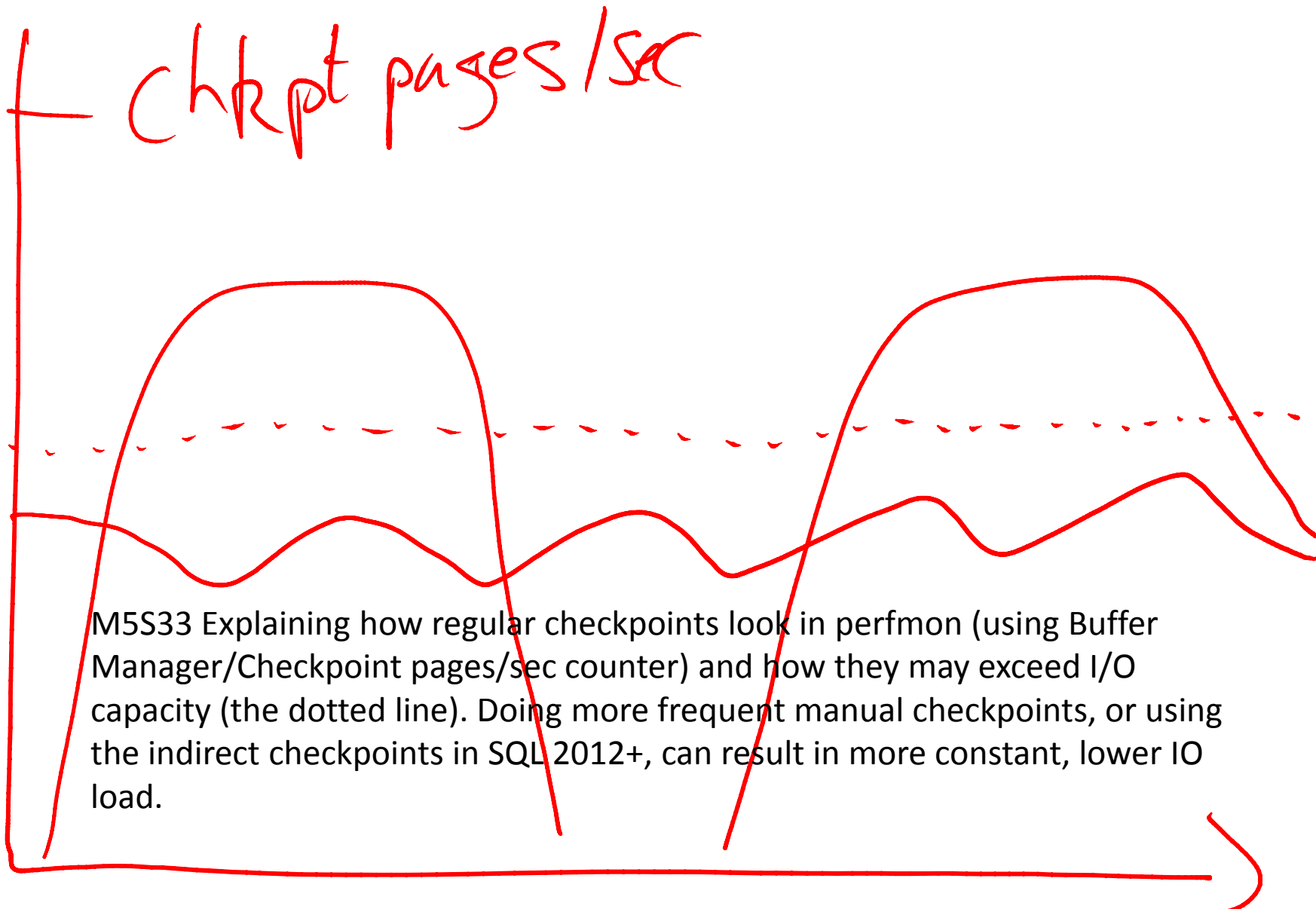


BC

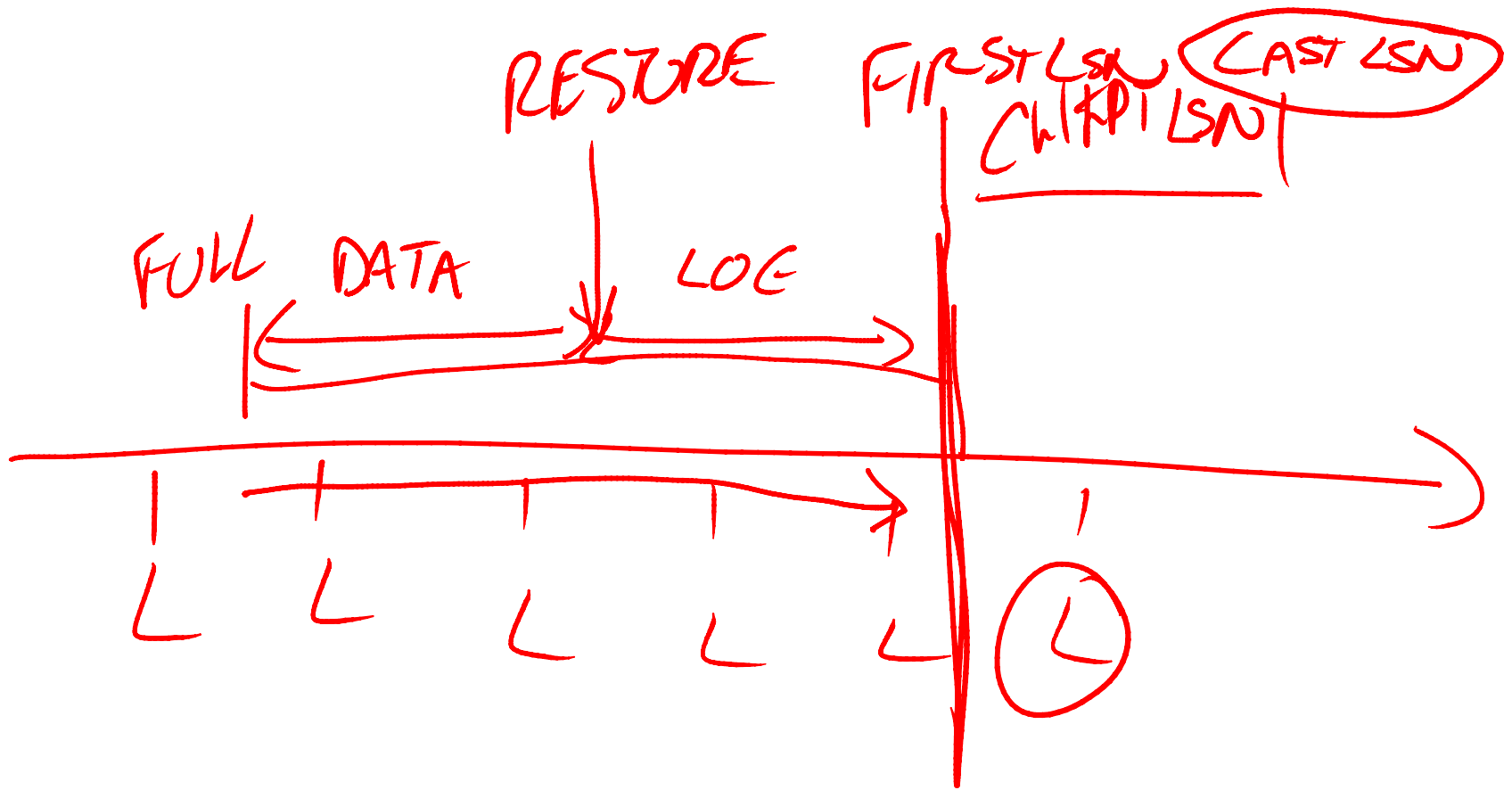
M5S30 Explaining why a checkpoint can't just flush the log at the start and be done. After the checkpoint starts, a data page in that database might become dirty and then needs to be written out by the checkpoint. The checkpoint has to make sure that any log records pertaining to that page are written to disk before the page: write-ahead logging.



chkpt pages/sec

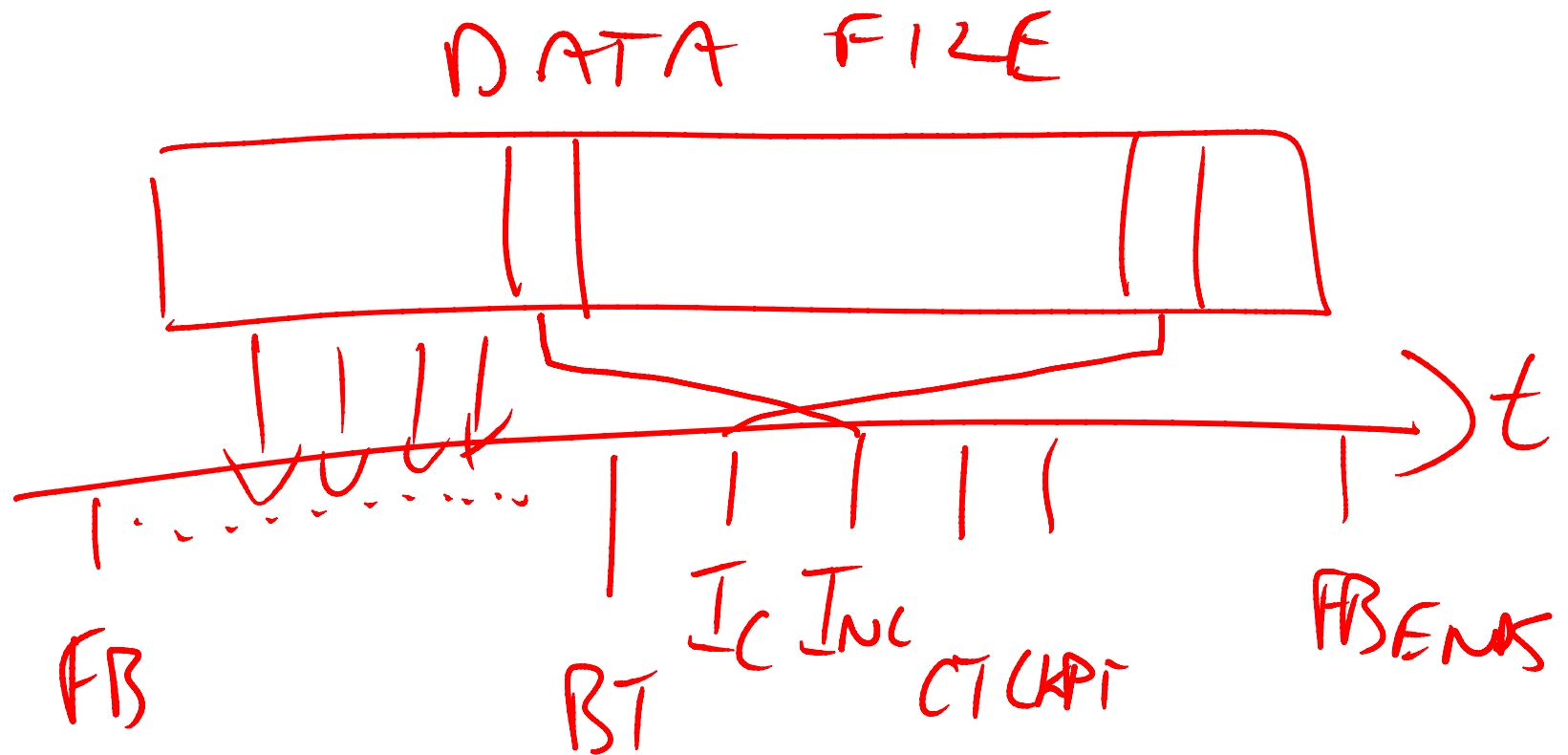


M5S33 Explaining how regular checkpoints look in perfmon (using Buffer Manager/Checkpoint pages/sec counter) and how they may exceed I/O capacity (the dotted line). Doing more frequent manual checkpoints, or using the indirect checkpoints in SQL 2012+, can result in more constant, lower IO load.



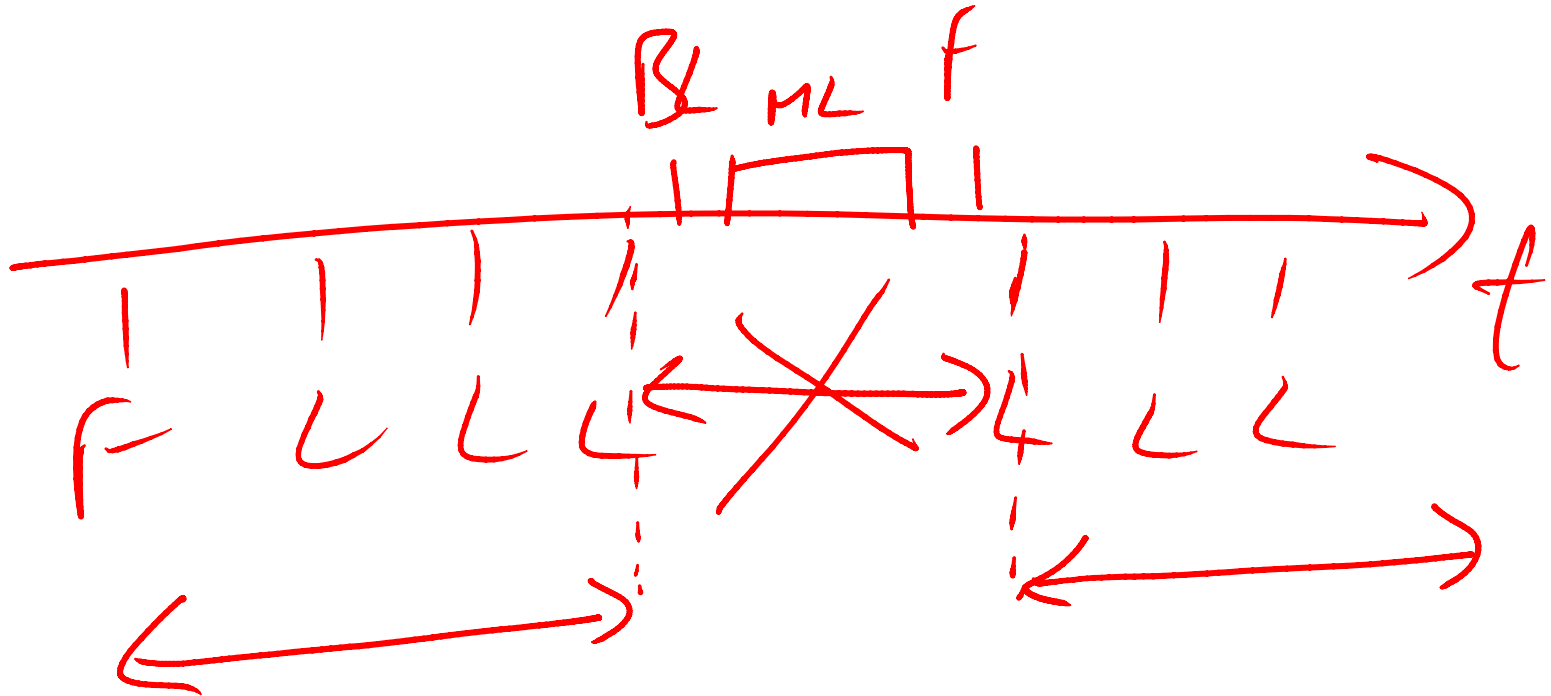
M5S45 It's possible for concurrent log backups to occur with a data backup (full or differential). Log clearing cannot occur though, as the data backup will back up the log. When the data backup finishes, deferred log clearing takes place.

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).

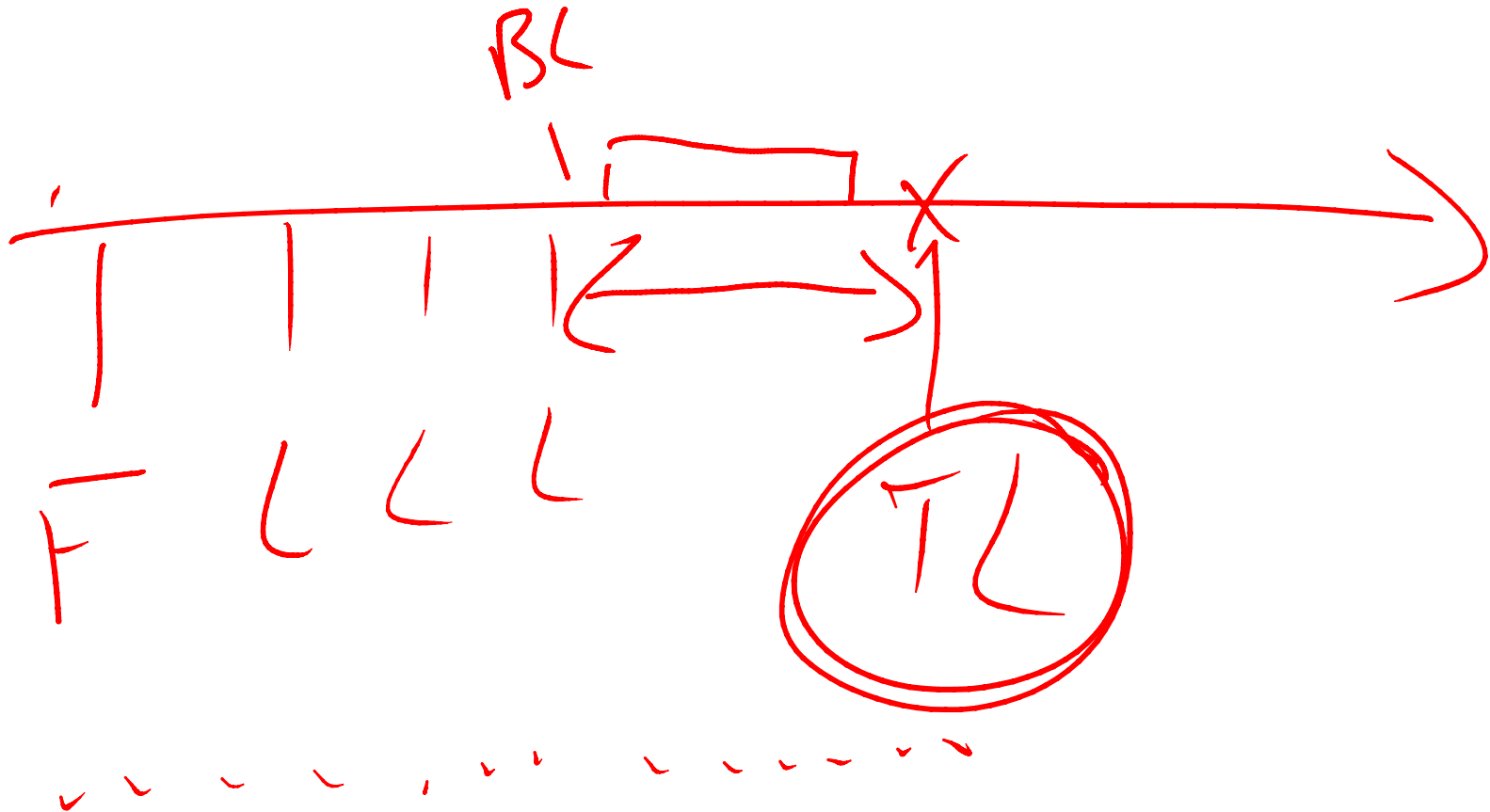


M5S45 Why does a full/diff backup contain log? It's to make sure that the restored database is transactionally consistent.

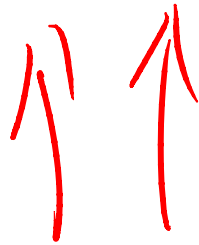
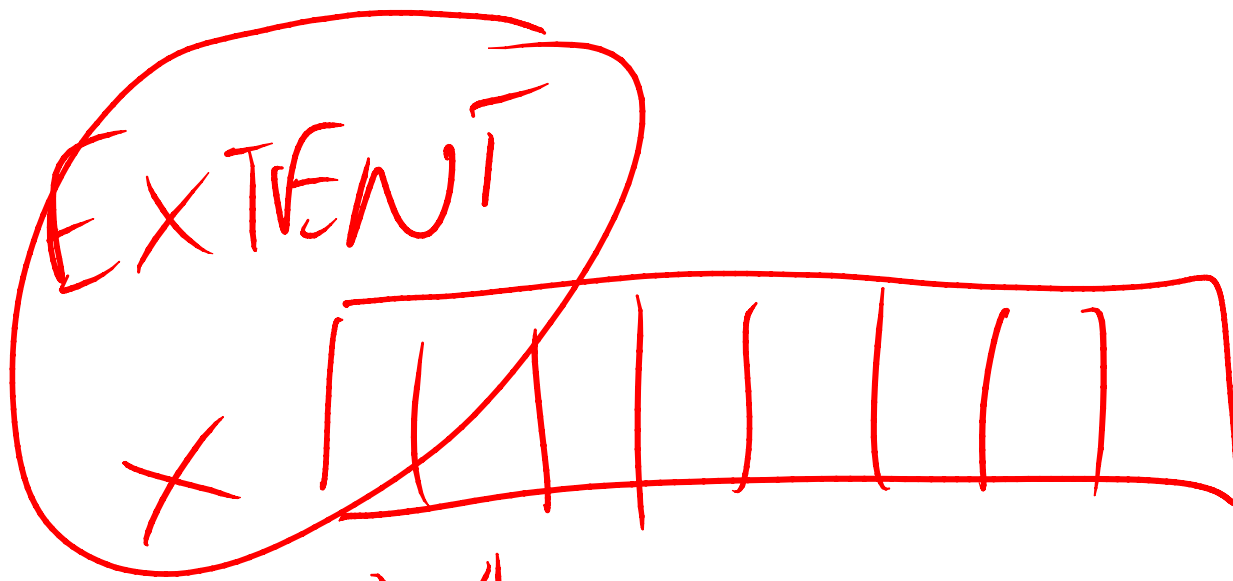
Scenario: full backup starts and reads nonclustered index page before insert. Transaction starts and inserts a row into clustered index page (Ic) and nonclustered index page (Inc). Checkpoint flushes out both pages. Backup then reads clustered index page with insert. If no log in the backup, restored database would be corrupt – missing nonclustered index record.



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.

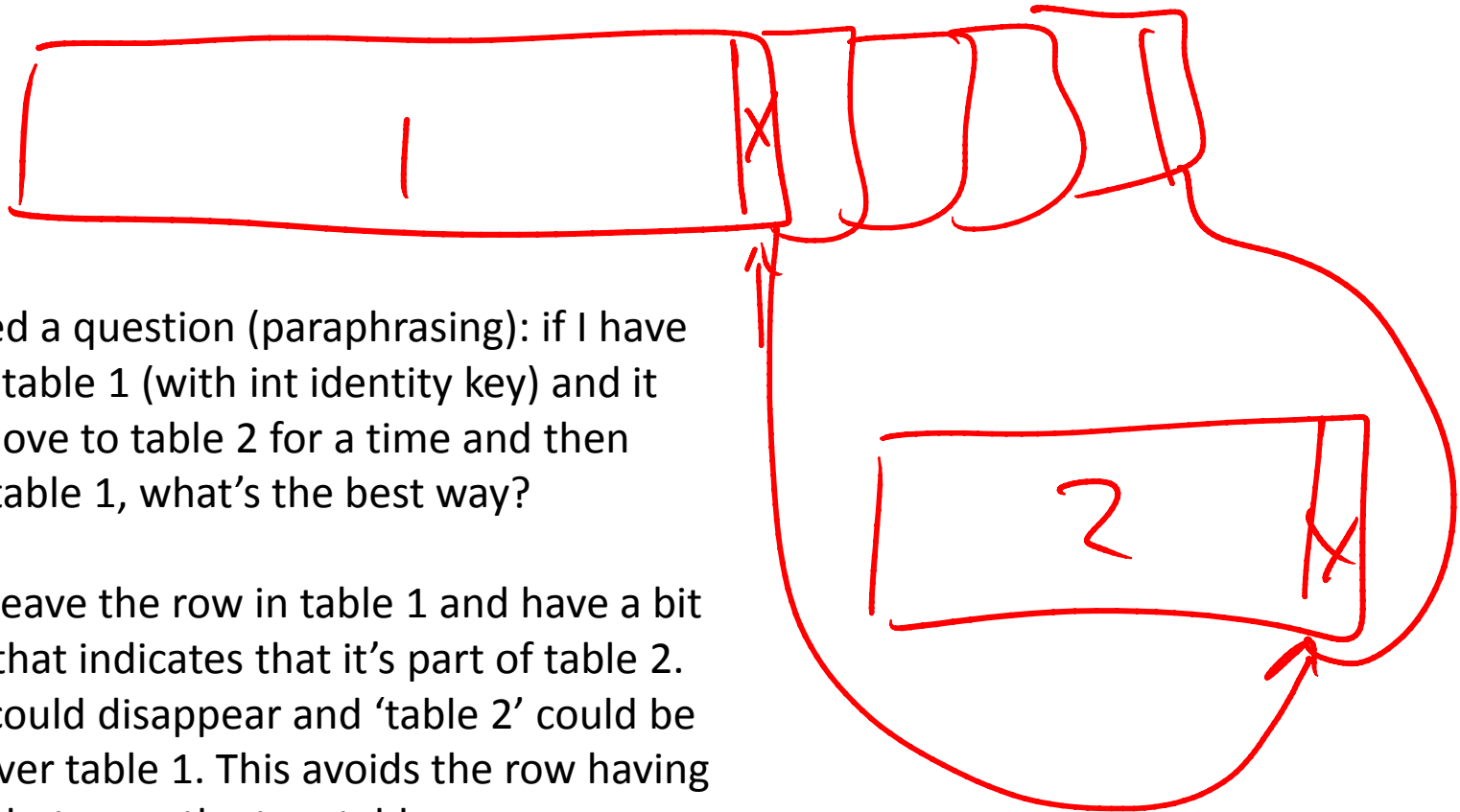
COMMENTS

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...



KIM'S CASE



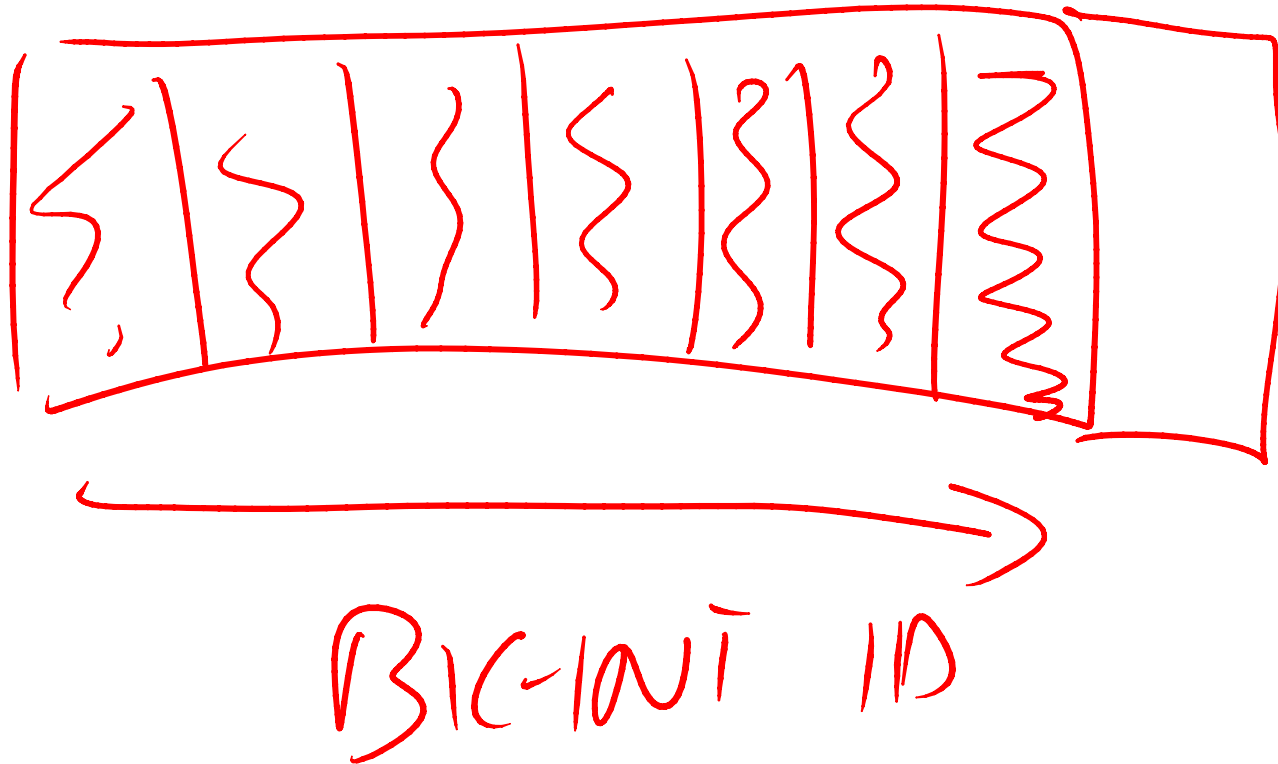
Kim asked a question (paraphrasing): if I have a row in table 1 (with int identity key) and it has to move to table 2 for a time and then back to table 1, what's the best way?

I would leave the row in table 1 and have a bit column that indicates that it's part of table 2. Table 2 could disappear and 'table 2' could be a view over table 1. This avoids the row having to move between the two tables.

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

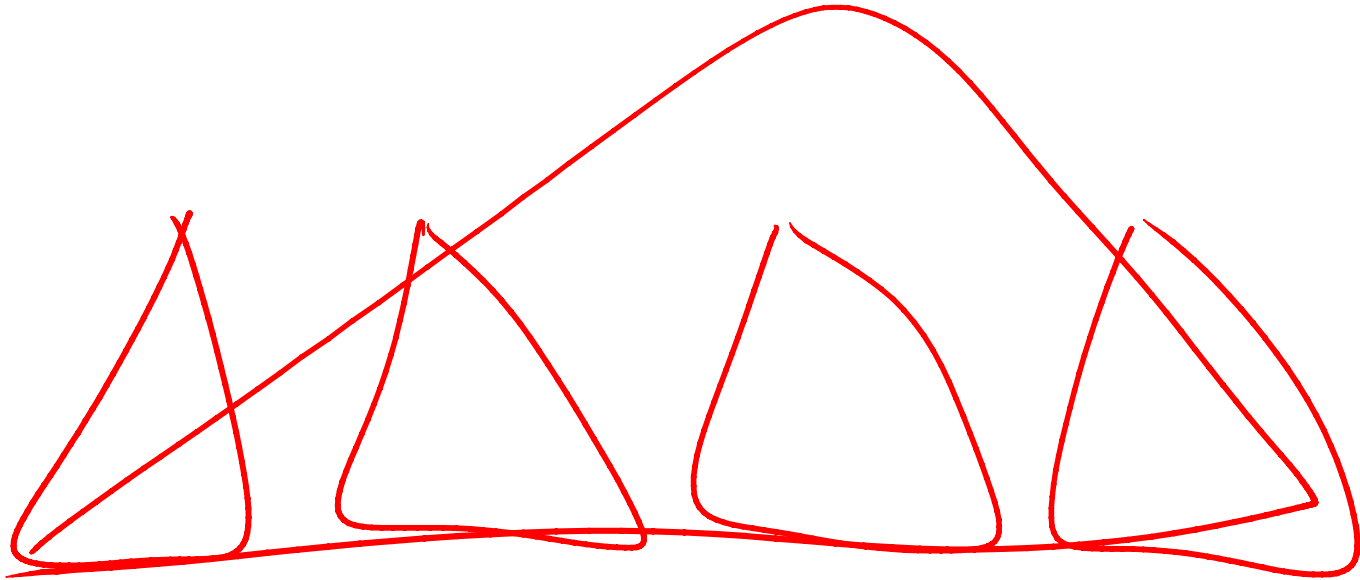
M7S28 The numbers from the page split logging cost demo.

LOP_DELETE_SPLIT



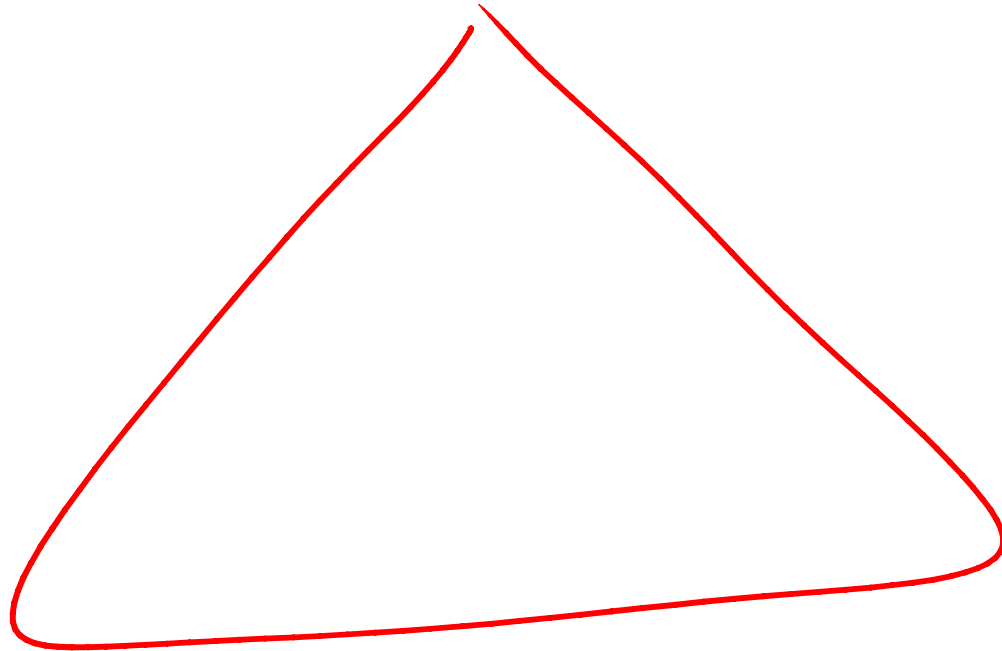
M7S30 Append-only insert pattern fills pages on right-hand side of index. New page allocations in this case are called page splits too – making all methods of tracking page splits largely useless (until 2012 with Xevents). You can look for LOP_DELETE_SPLIT log records to see splits occurring.

DOP 4



M7S38 Discussing how DOP affects contiguity. With DOP 4, 4 threads are building portions of the new index in parallel. They will all be allocating extents, leading to extent fragmentation (and hence logical fragmentation).

DOP 1



M7S38 Discussing how DOP affects contiguity. With DOP 4, 4 threads are building portions of the new index in parallel. They will all be allocating extents, leading to extent fragmentation (and hence logical fragmentation).

ITB



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