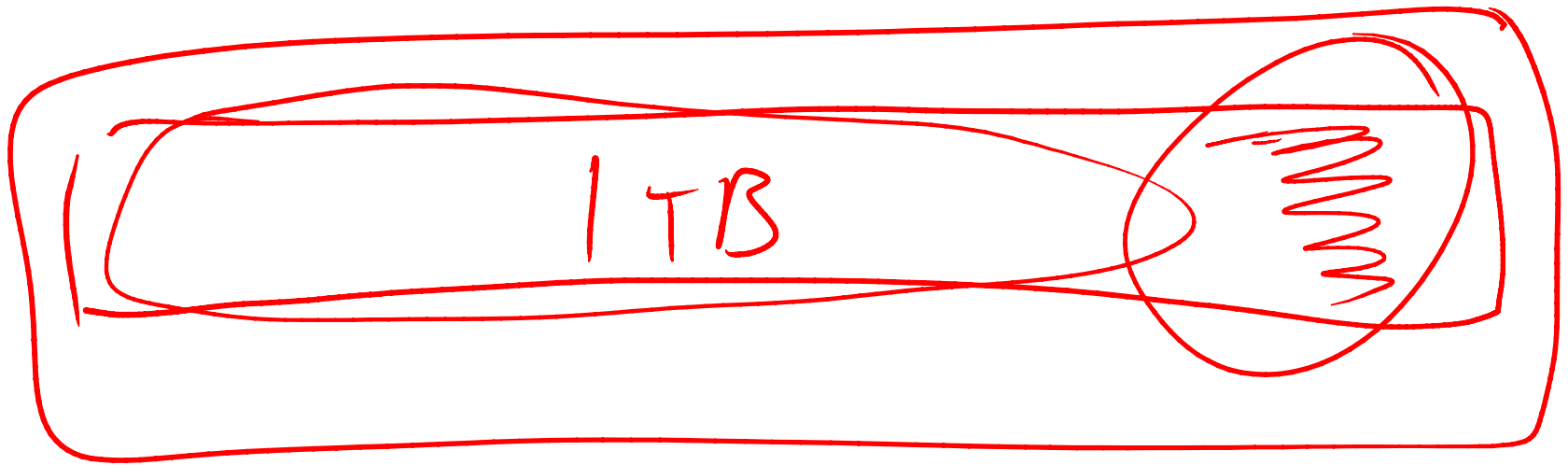
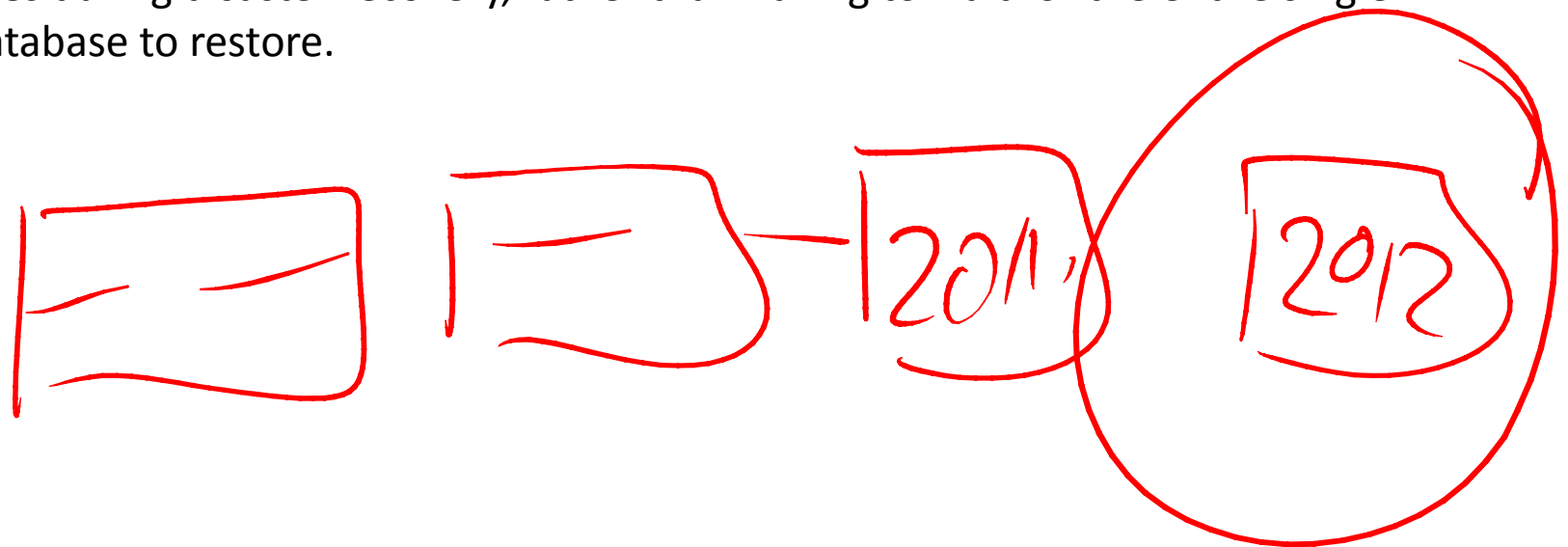
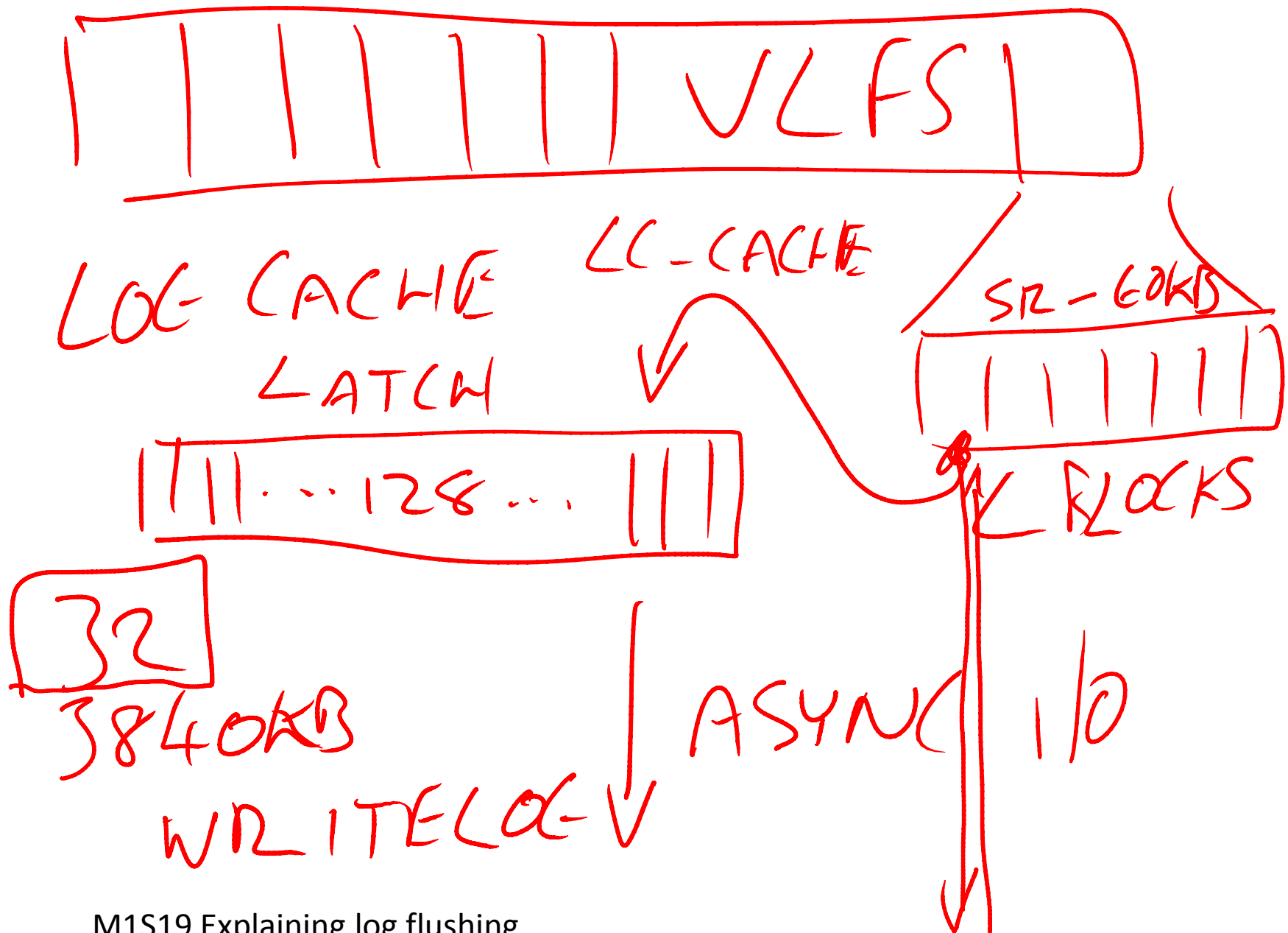
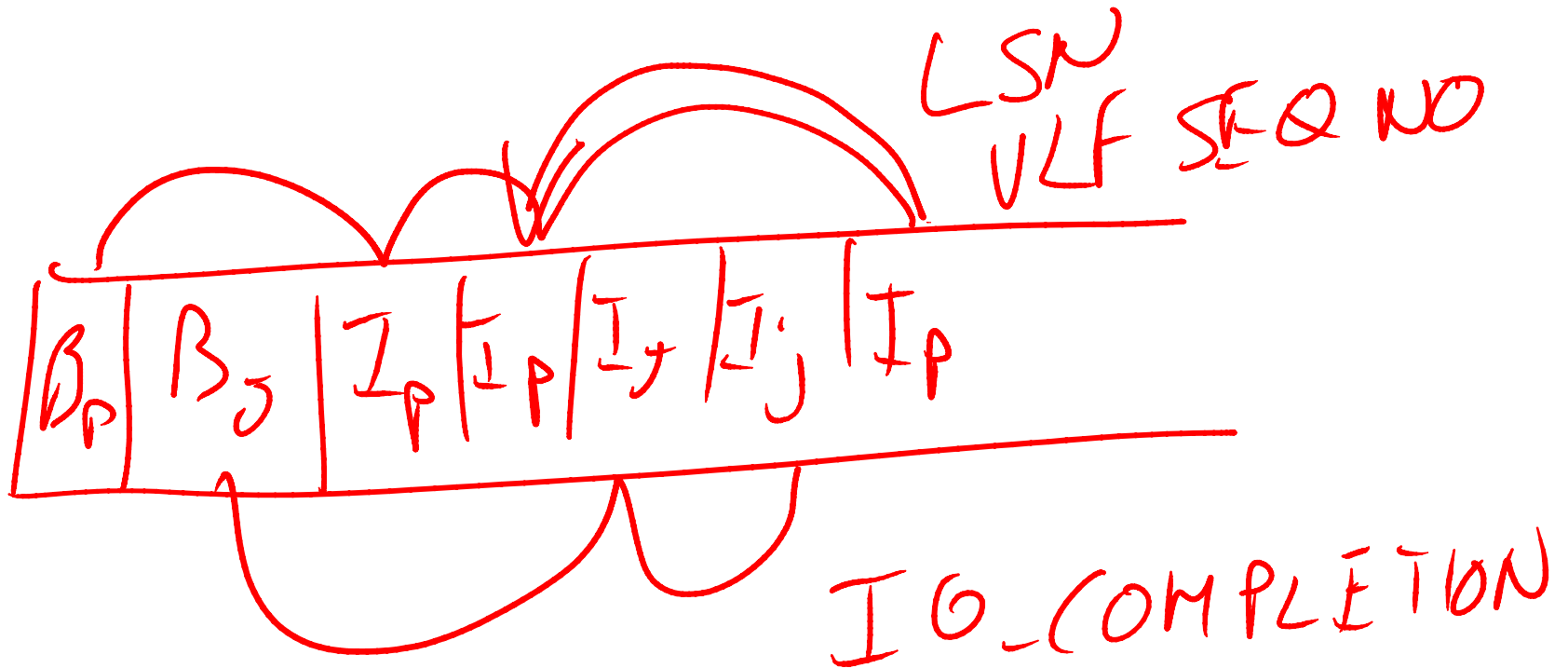
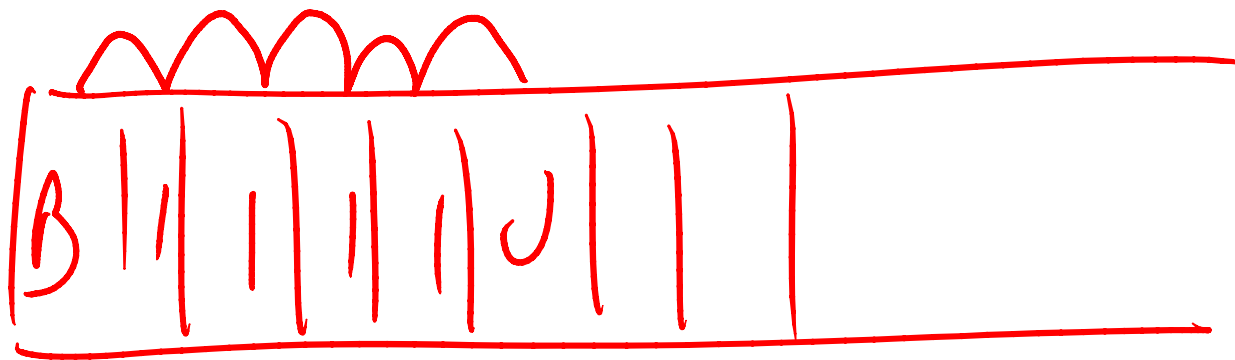




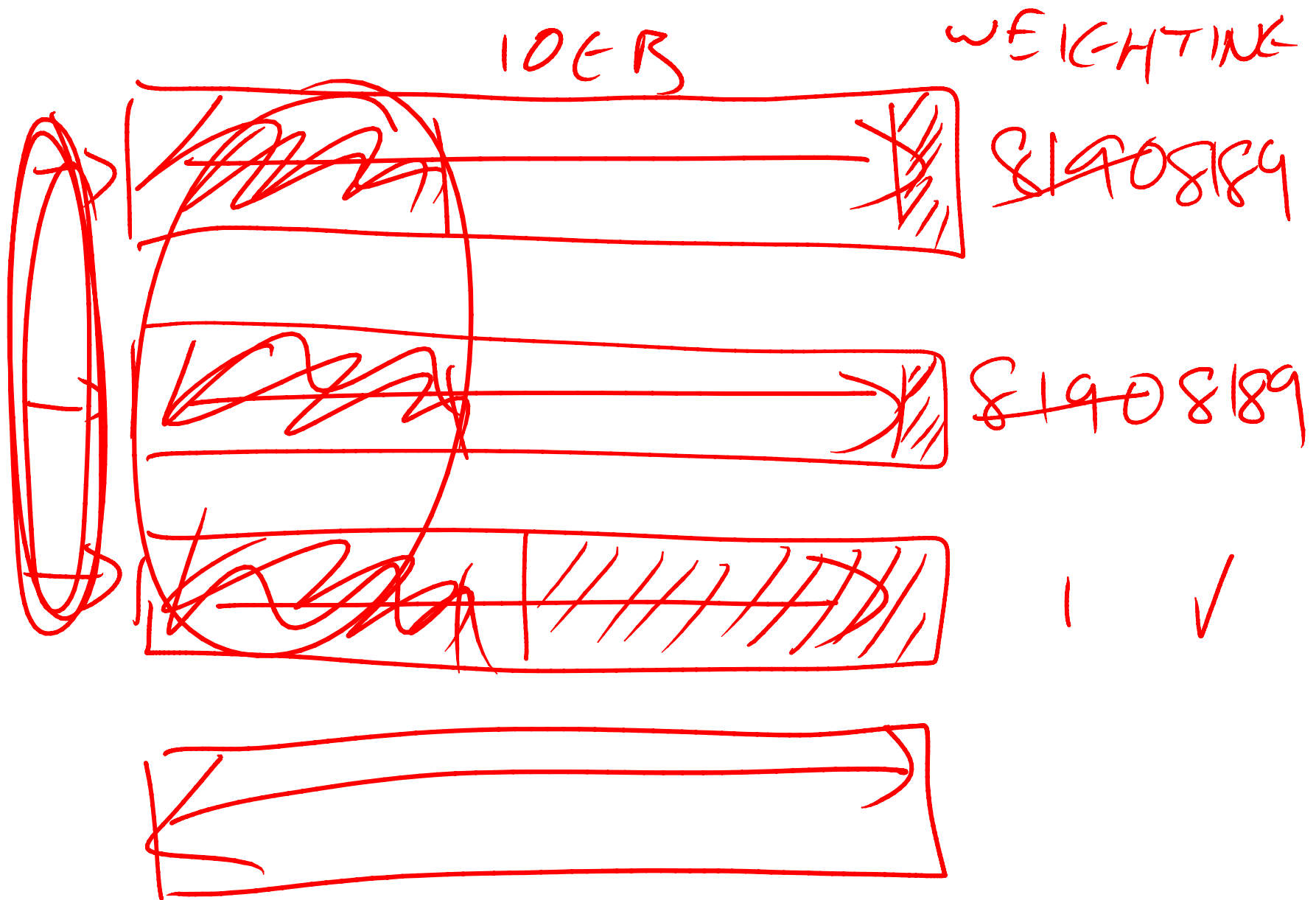
M1S7 Explaining that dividing a large database into filegroups allows targeted restores during disaster recovery, rather than having to wait for the entire single file database to restore.







M1 Explaining how log records are backwards linked and how having lots of
 , VLFs can cause the search for a log record to take time -> IO_COMPLETION
 waits



M1S22 Explaining proportional fill and round robin. Adding another file to a full filegroup does not allow for rebalancing of data.

1000	3000 LOB
------	----------

1%
1/4

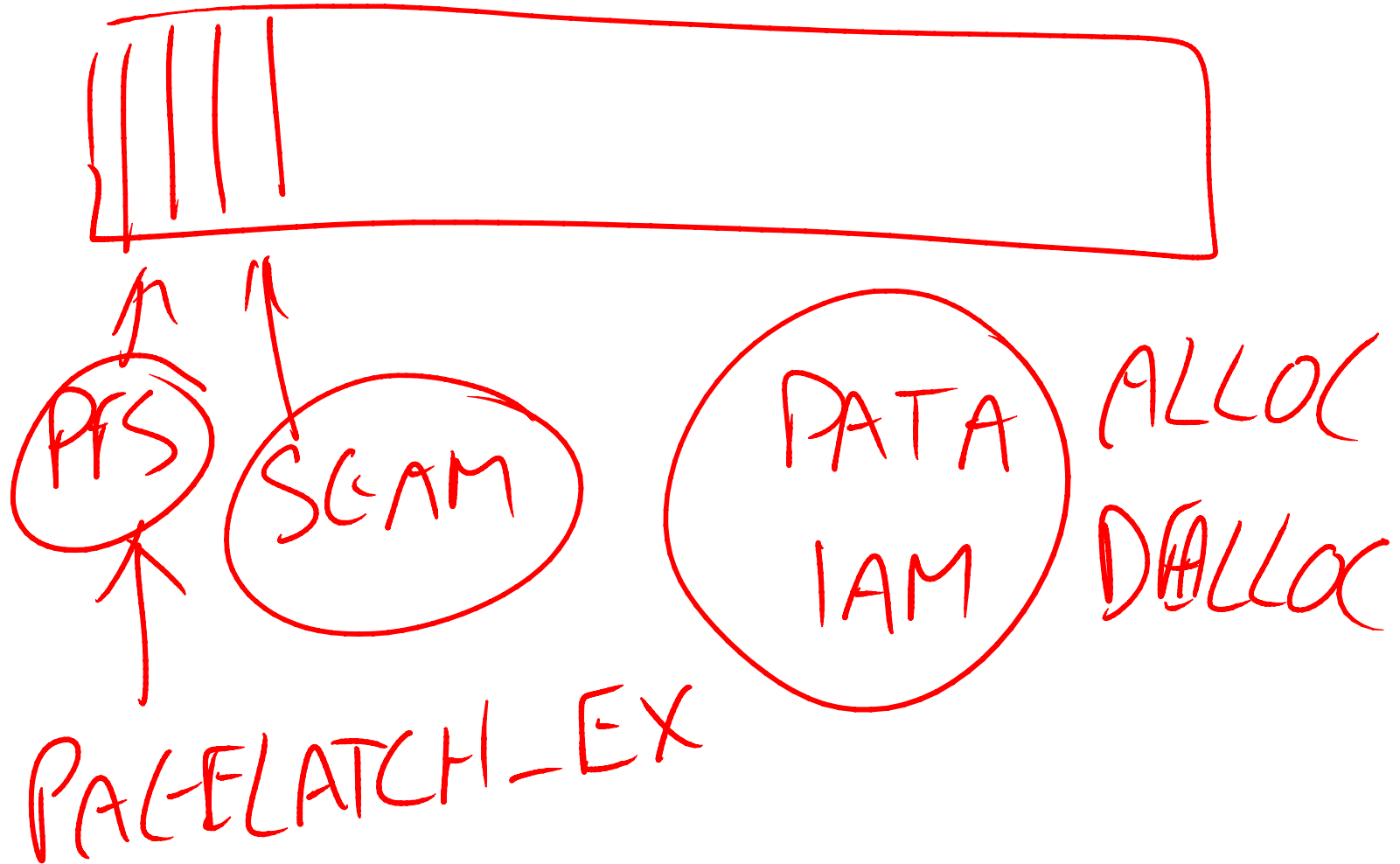
2

1000

3000

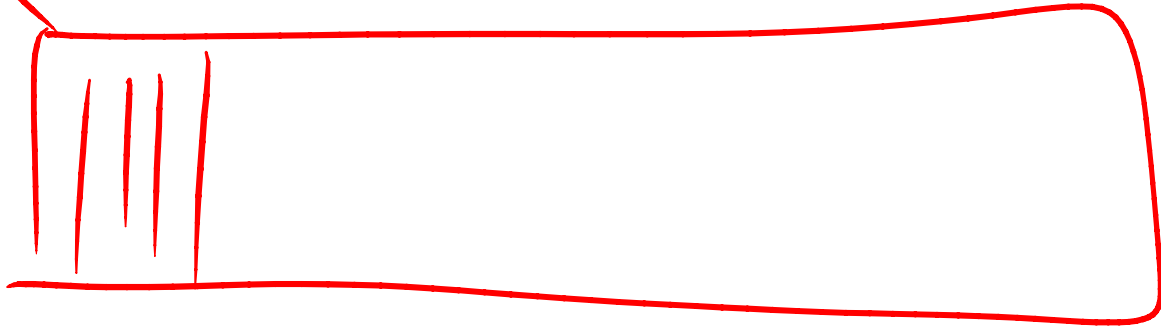
1
8

M1 Discussing LOB storage schemes to increase data density. If 3000 bytes are only used 1% of the time, store them separately so the rows are smaller and there are more per page. This leads to fewer page reads and better buffer pool data density.

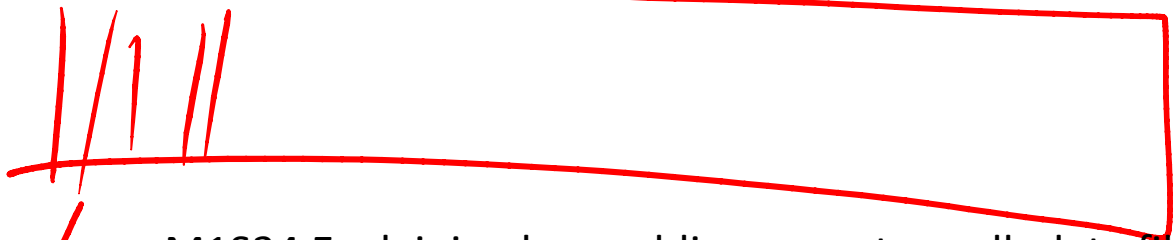


M1S33 Explaining how PFS and SGAM contention occur in tempdb

PFS



TF 1118 - 328551



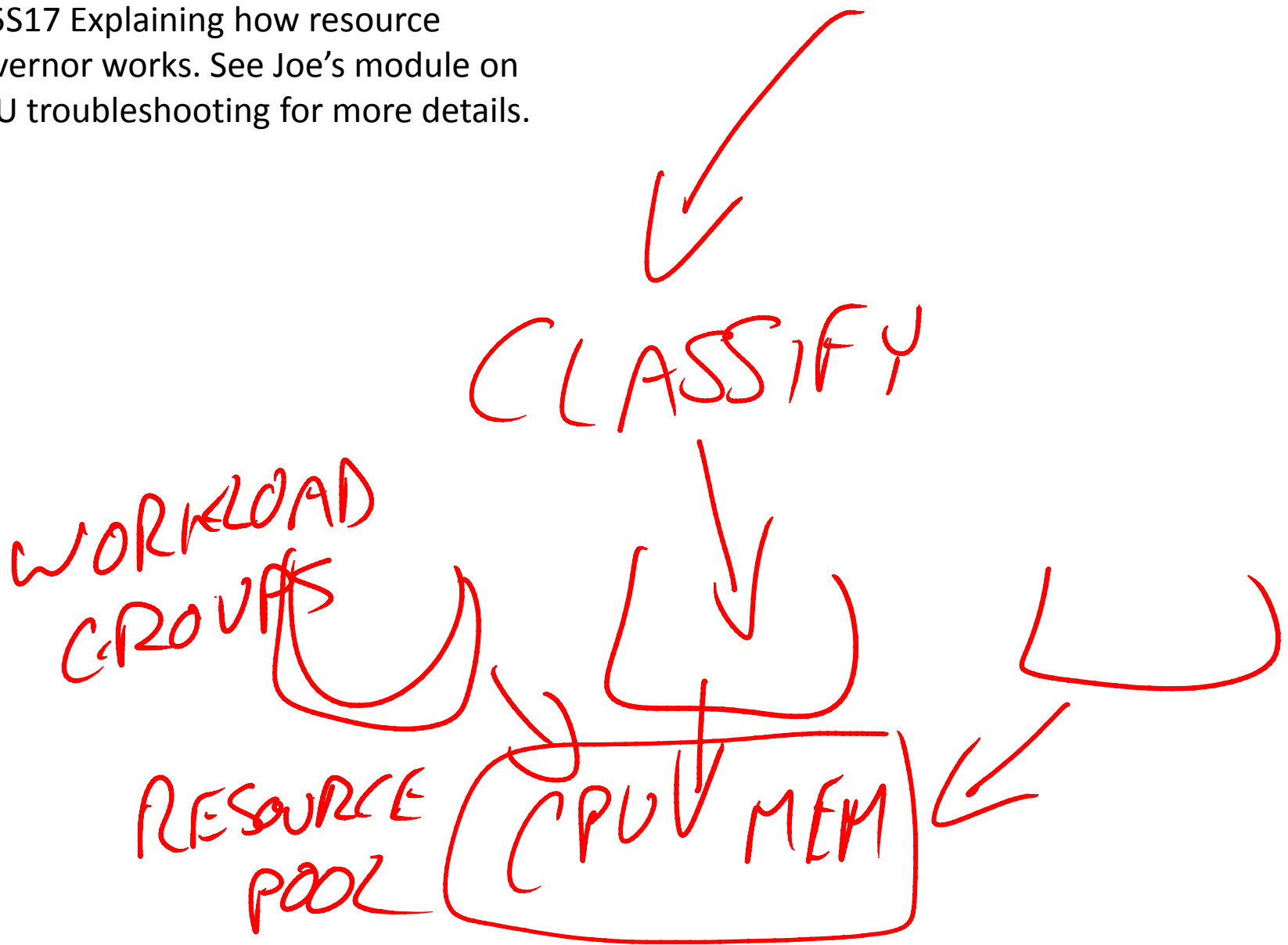
M1S34 Explaining how adding more tempdb data files leads to round-robin allocation and a spread of the contention to multiple files, leading to lower overall contention.

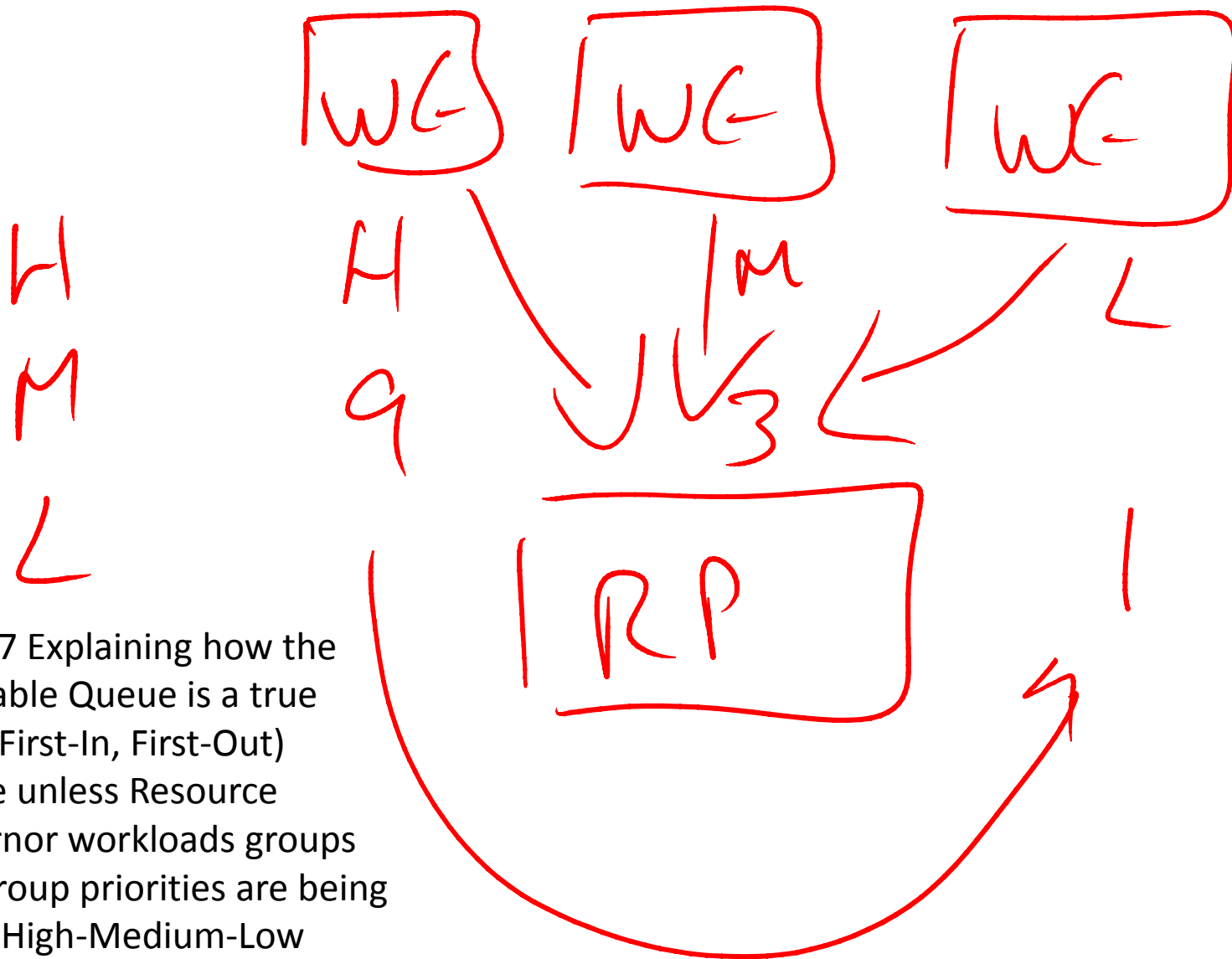
PFS



PFS

M5S17 Explaining how resource governor works. See Joe's module on CPU troubleshooting for more details.





M5S17 Explaining how the Runnable Queue is a true FIFO (First-In, First-Out) queue unless Resource Governor workloads groups and group priorities are being used. High-Medium-Low priority equals 9-3-1 threads through the Runnable Queue.

SPINLOCK

LOCKS

ACQUIRE

LATCHES

INTERLOCKED

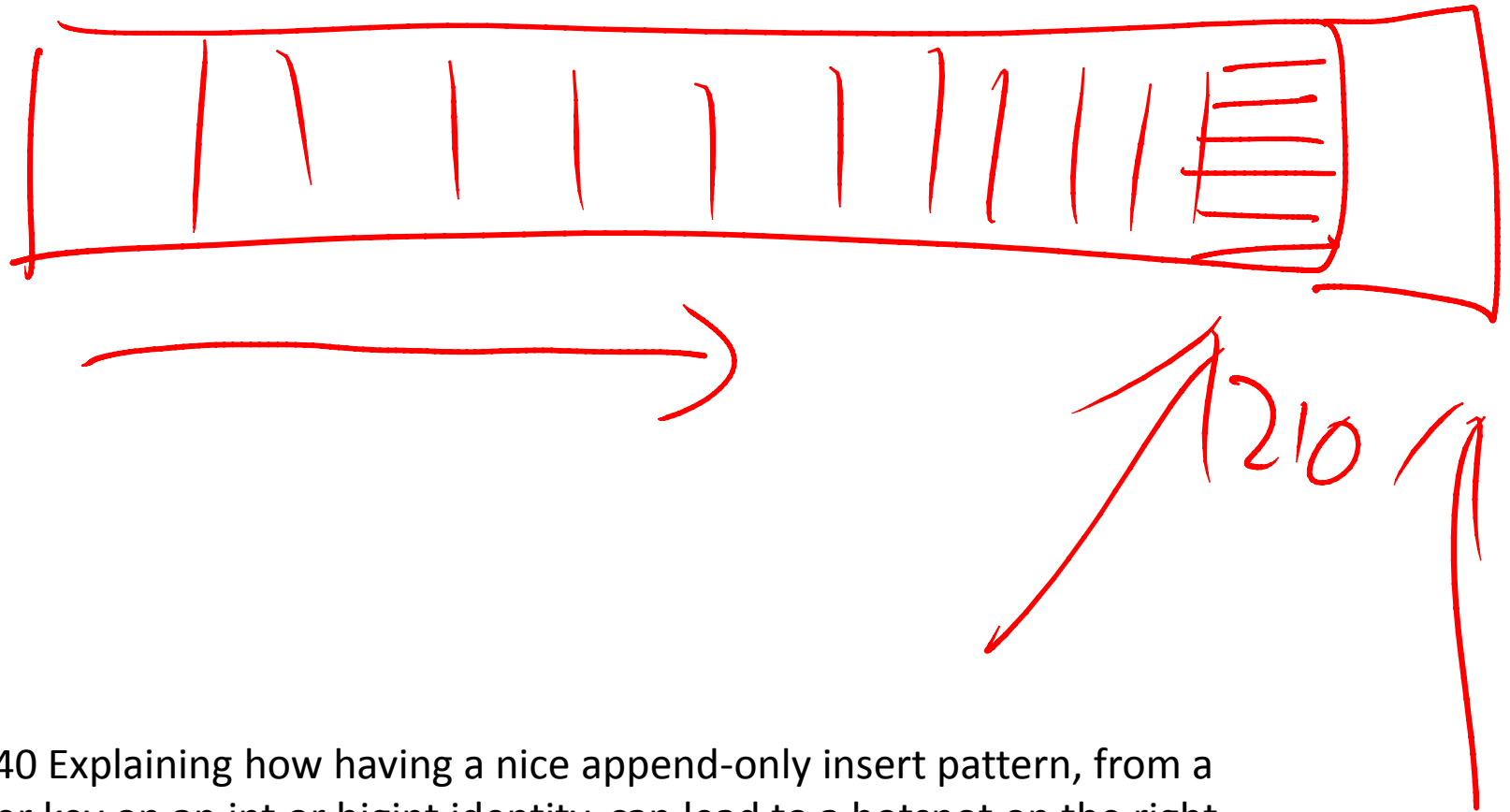
TEST-AND-SET-IF-CLEAR

4ms



M5 Explaining how spinlocks work. Code does a test-and-set-if-clear on the memory that implements the spinlock. If it can't get it, it spins (tries again) up to 1000 times and then backs off. SOS_SCHEDULER_YIELD is the wait type when a backoff occurs.

SPINLOCKS



M5S40 Explaining how having a nice append-only insert pattern, from a cluster key on an int or bigint identity, can lead to a hotspot on the rightmost data page as hundreds of threads all try to insert onto the same page – leading to PAGELATCH contention.