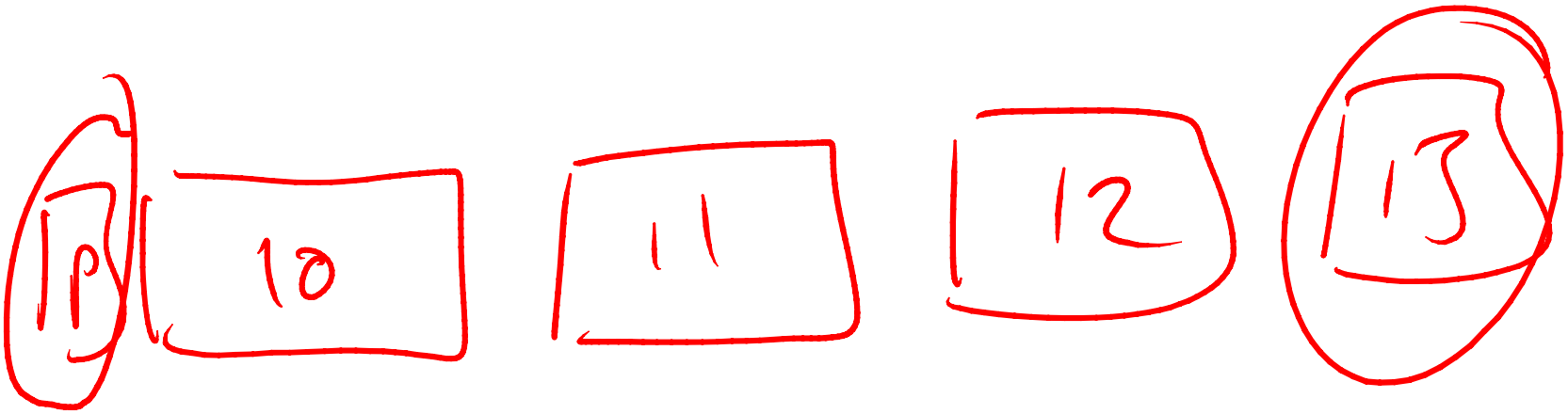
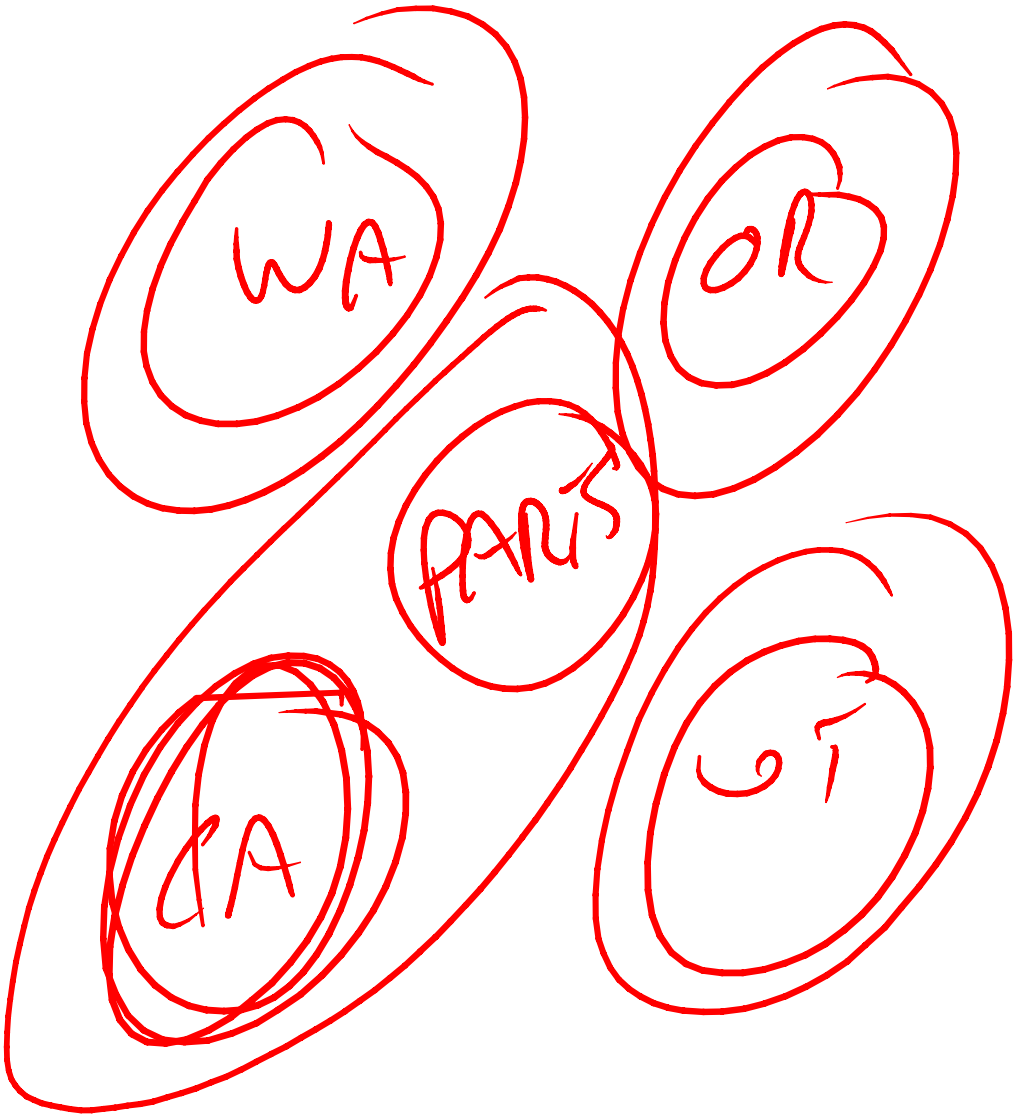
Your friend on twitter…

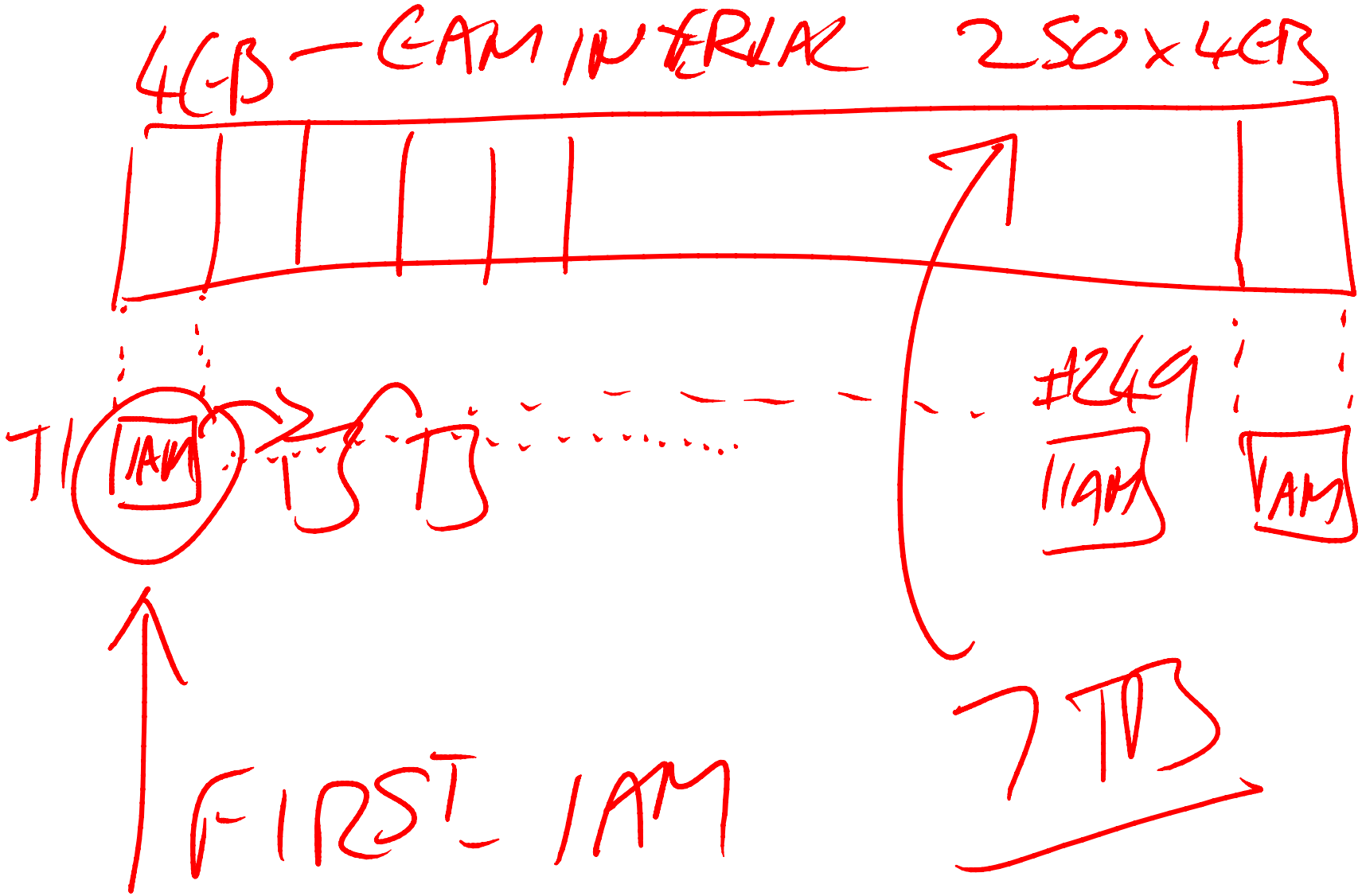
M1S7 Explaining that dividing a large database into filegroups allows targeted restores during disaster recovery, rather than having to wait for the entire single file database to restore.





M1S7 Explaining a different partitioning scenario – auto parts sales to WA/CA/OR/UT.

If things go down, it makes sense to restore the portion of the database that will bring in the most sales quickly – CA, as it has the most population.



M1 Explaining how if you have a very large (e.g. multi-TB), non-partitioned table, whenever allocations happen from a new 4GB portion of the filegroup, a new IAM page must be added at the end of the IAM chain. With 100s of IAM pages in the chain, this can become problematic.

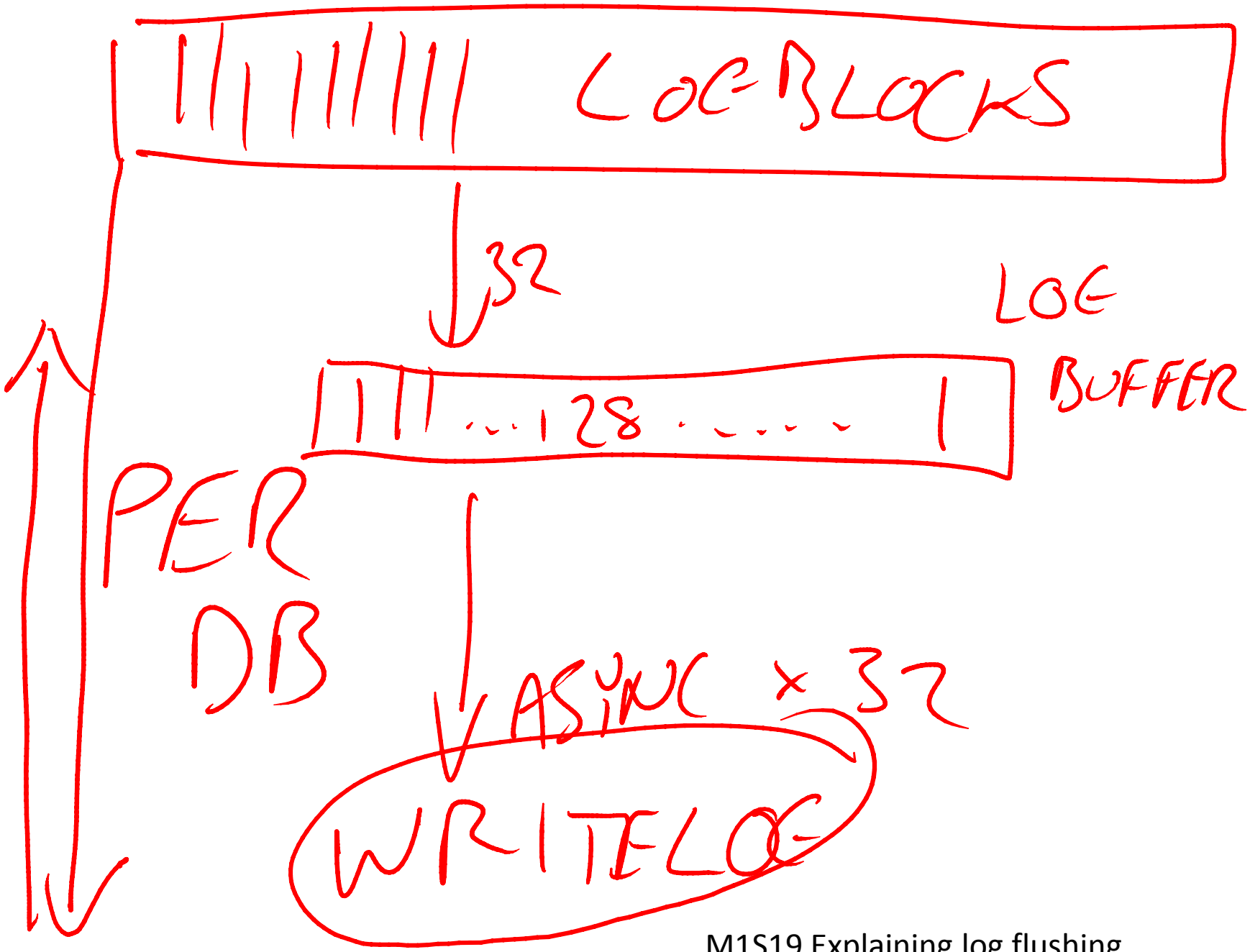
1 log flush = 1 log block

512b → 60KB

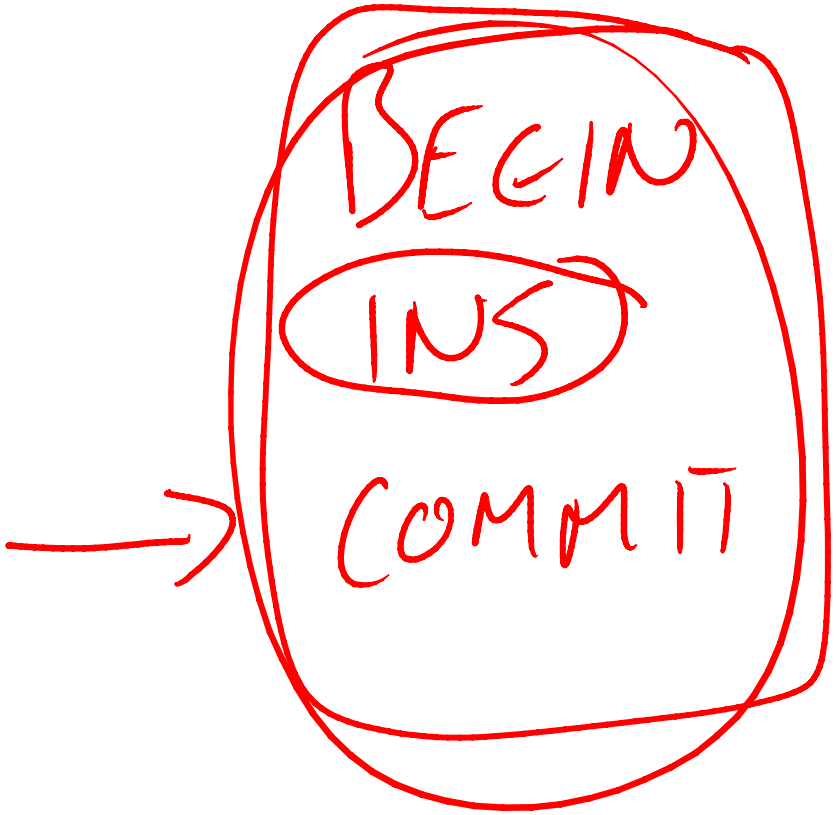
3840KB = 64 × 60KB

32 outstanding

32 waiting



M1S19 Explaining log flushing

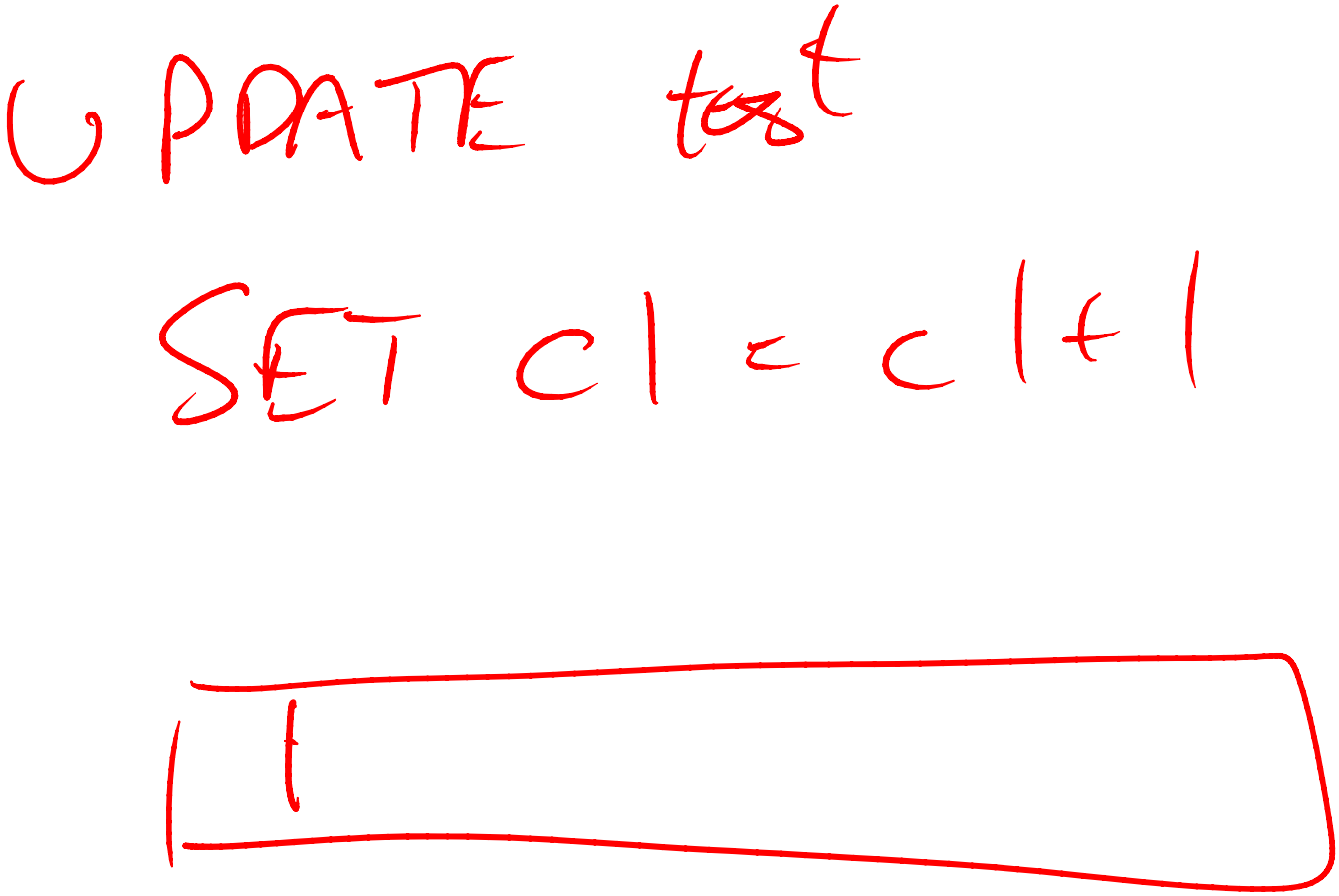


< 512

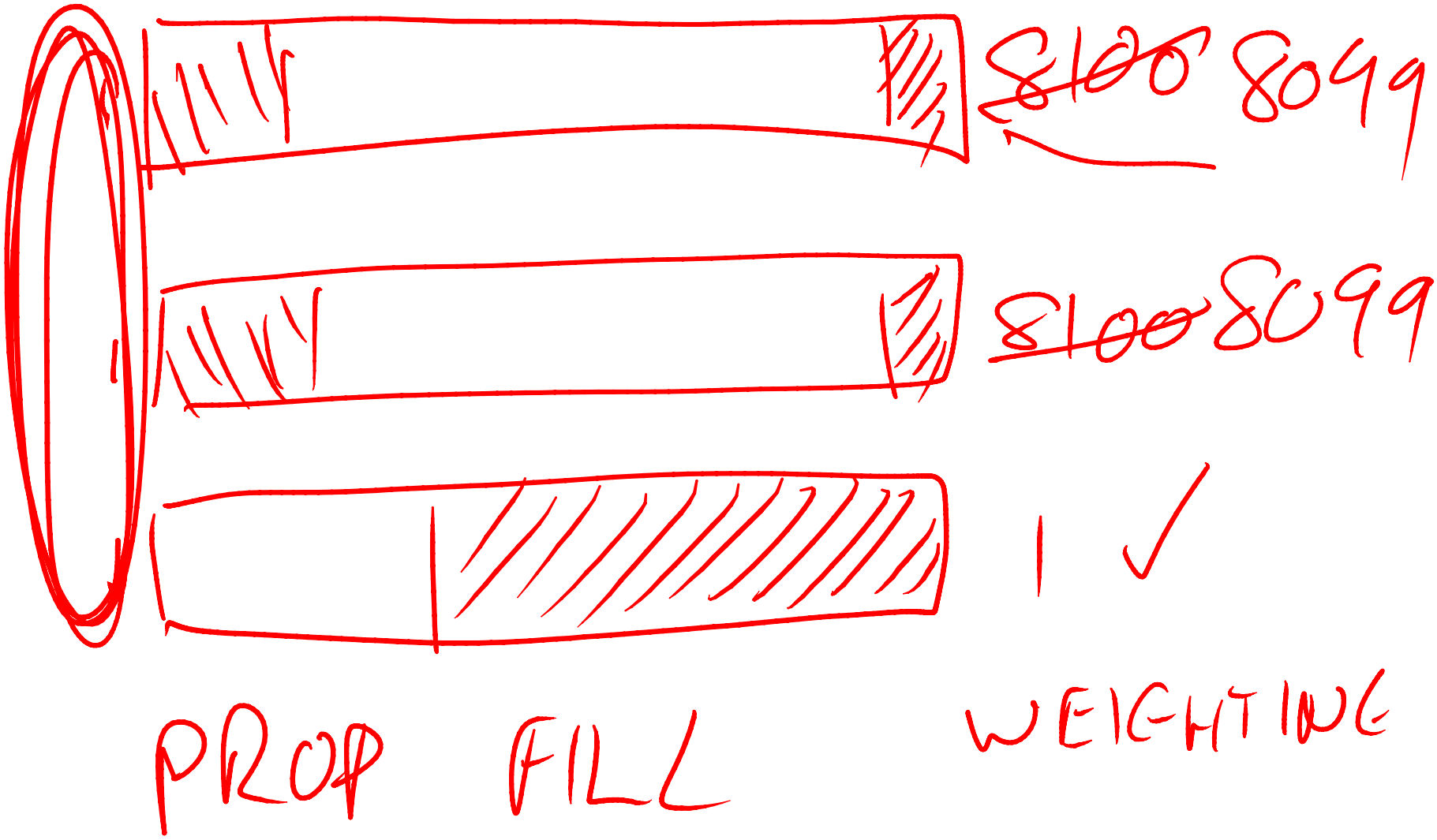
M1 Explaining that a tiny transaction results in a 512byte log flush.

KB2769701

KB about poor SAN performance on Win Svr 2008 R2

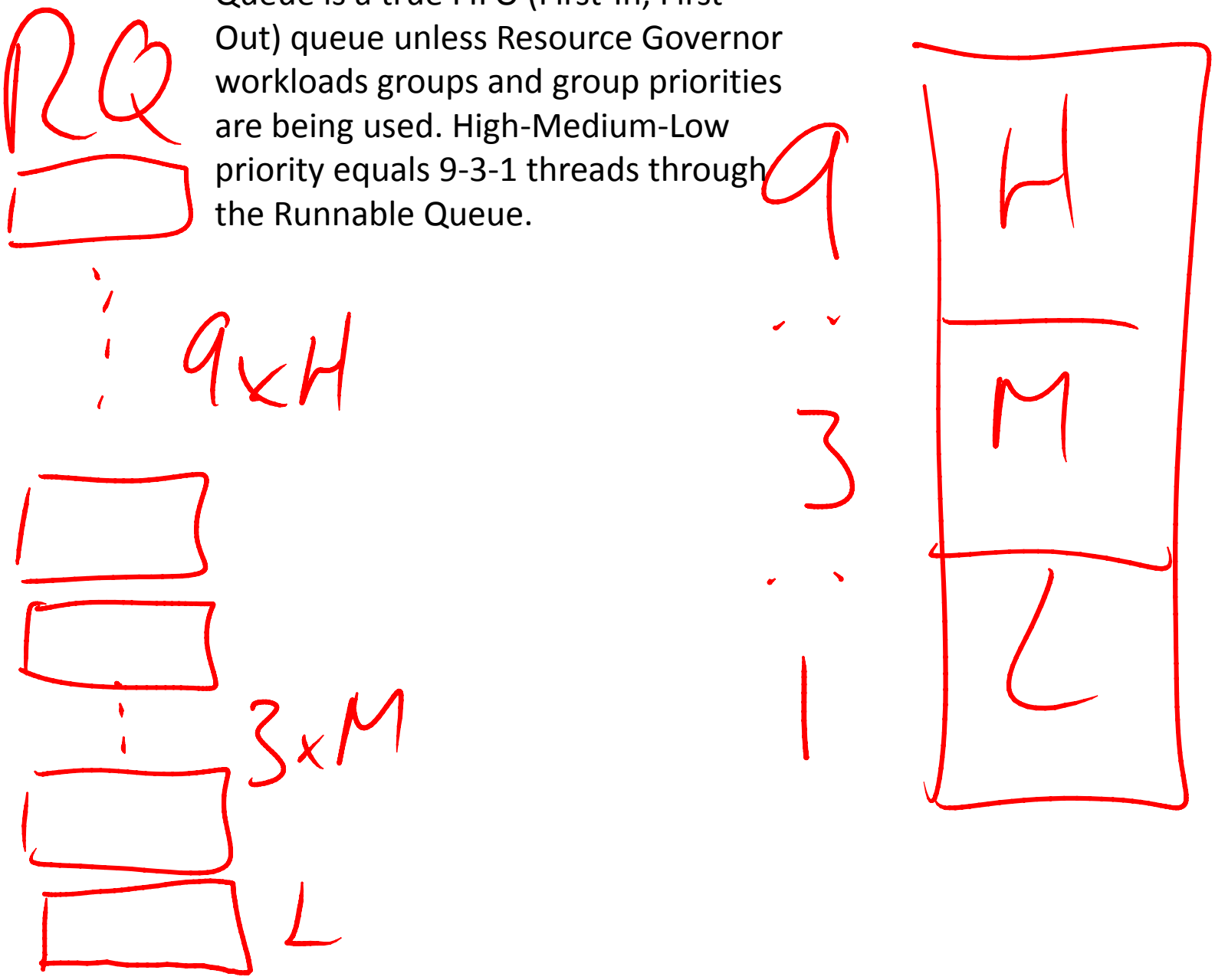


M1 Talking about ‘Halloween Protection’ – preventing an open-ended update seeing rows multiple times. Updating a key column becomes a delete+insert.

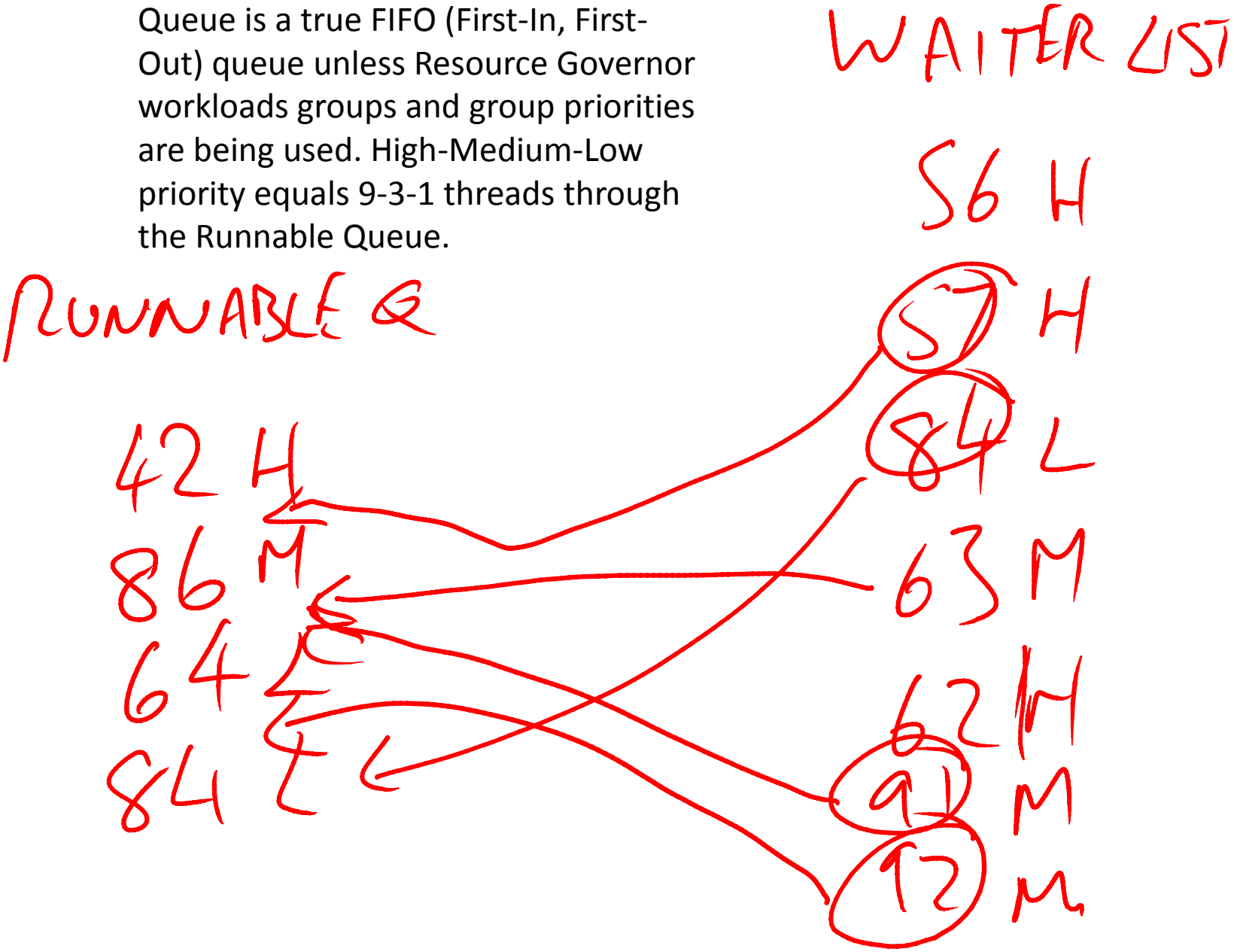


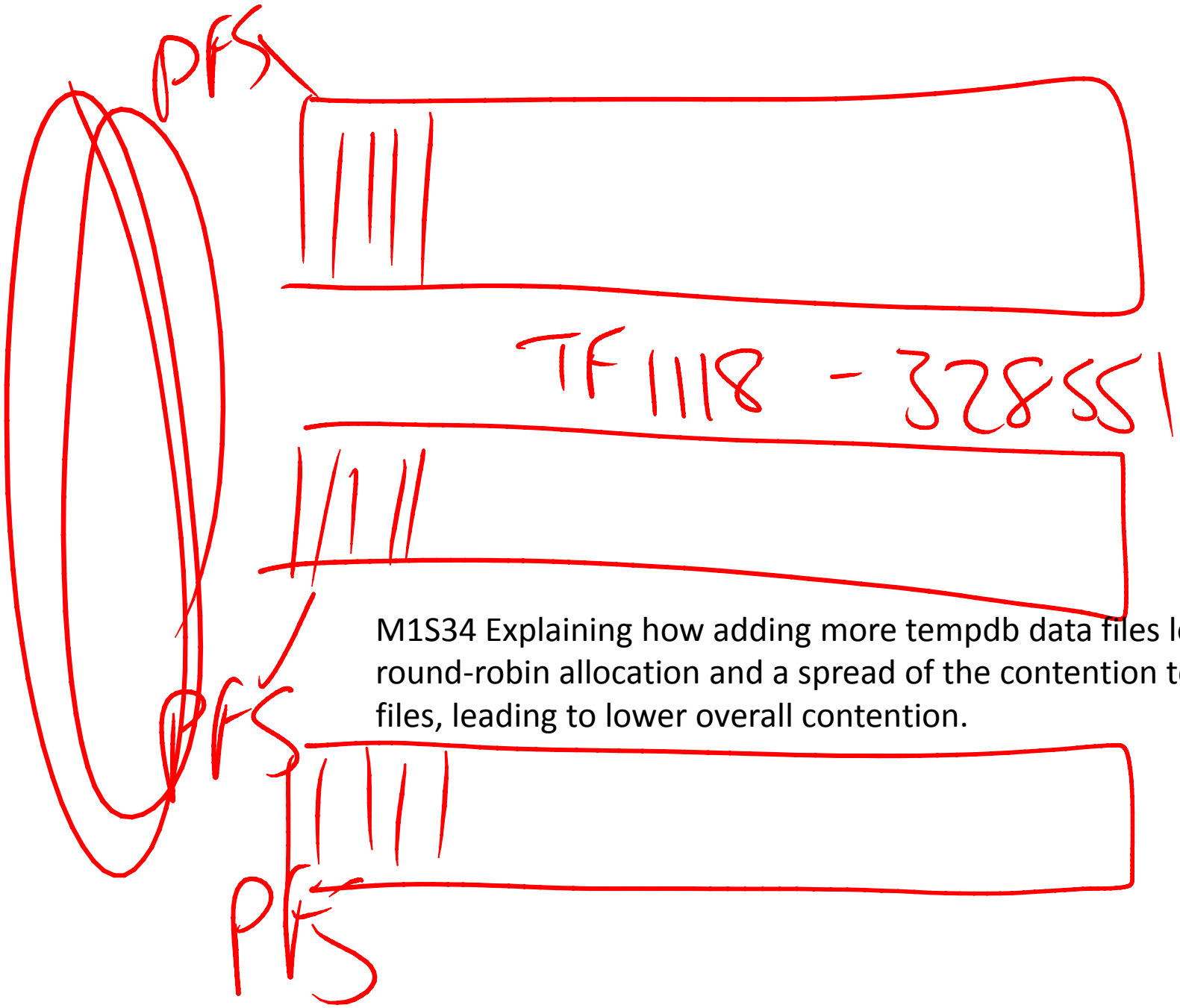
M1S22 Explaining proportional fill and round robin. Adding another file to a full filegroup does not allow for rebalancing of data.

M5S9 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.

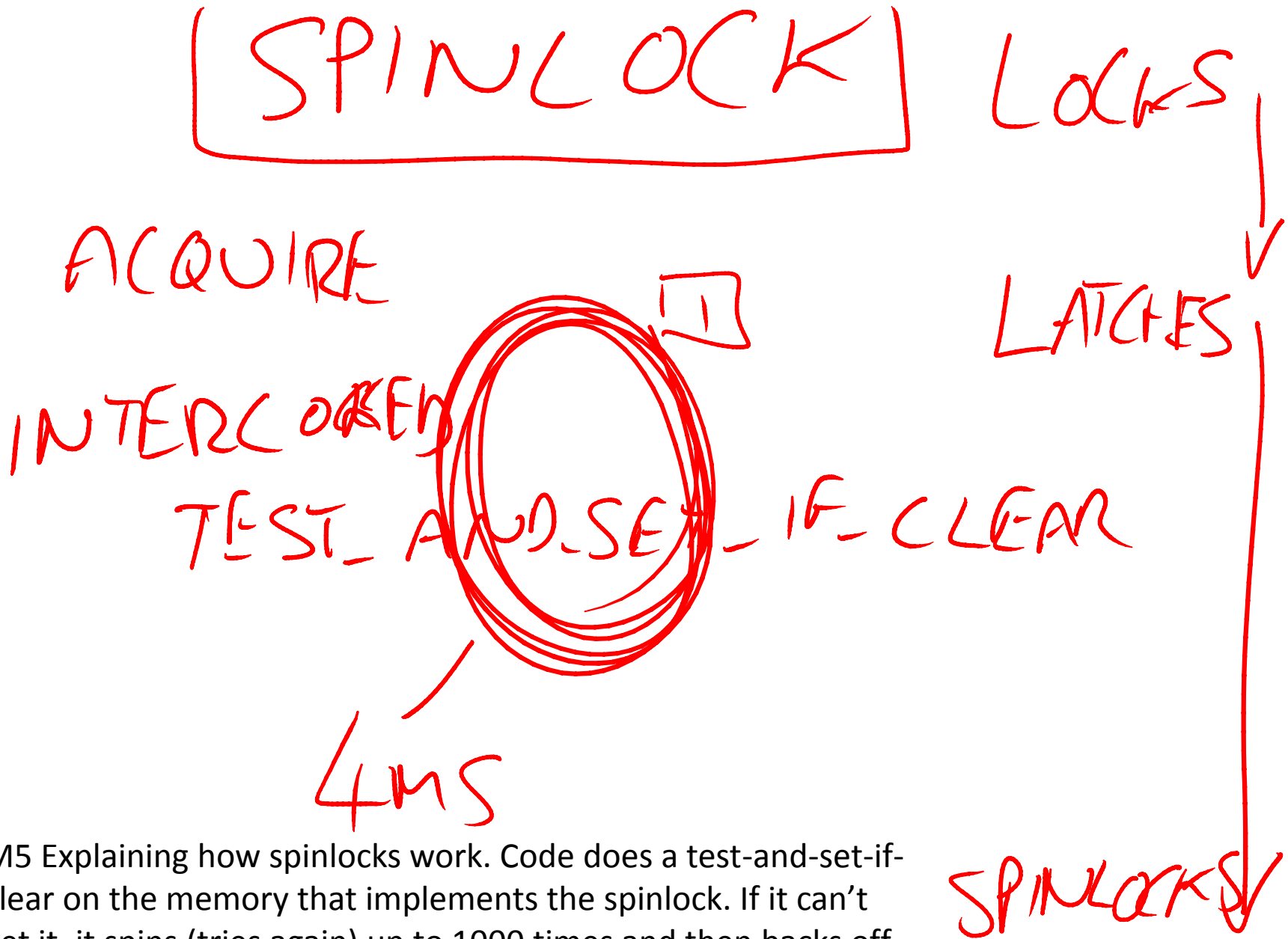


M5S9 Explaining how the Runnable Queue is a true FIFO (First-In, First-Out) queue unless Resource Governor workloads groups and group priorities are being used. High-Medium-Low priority equals 9-3-1 threads through the Runnable Queue.

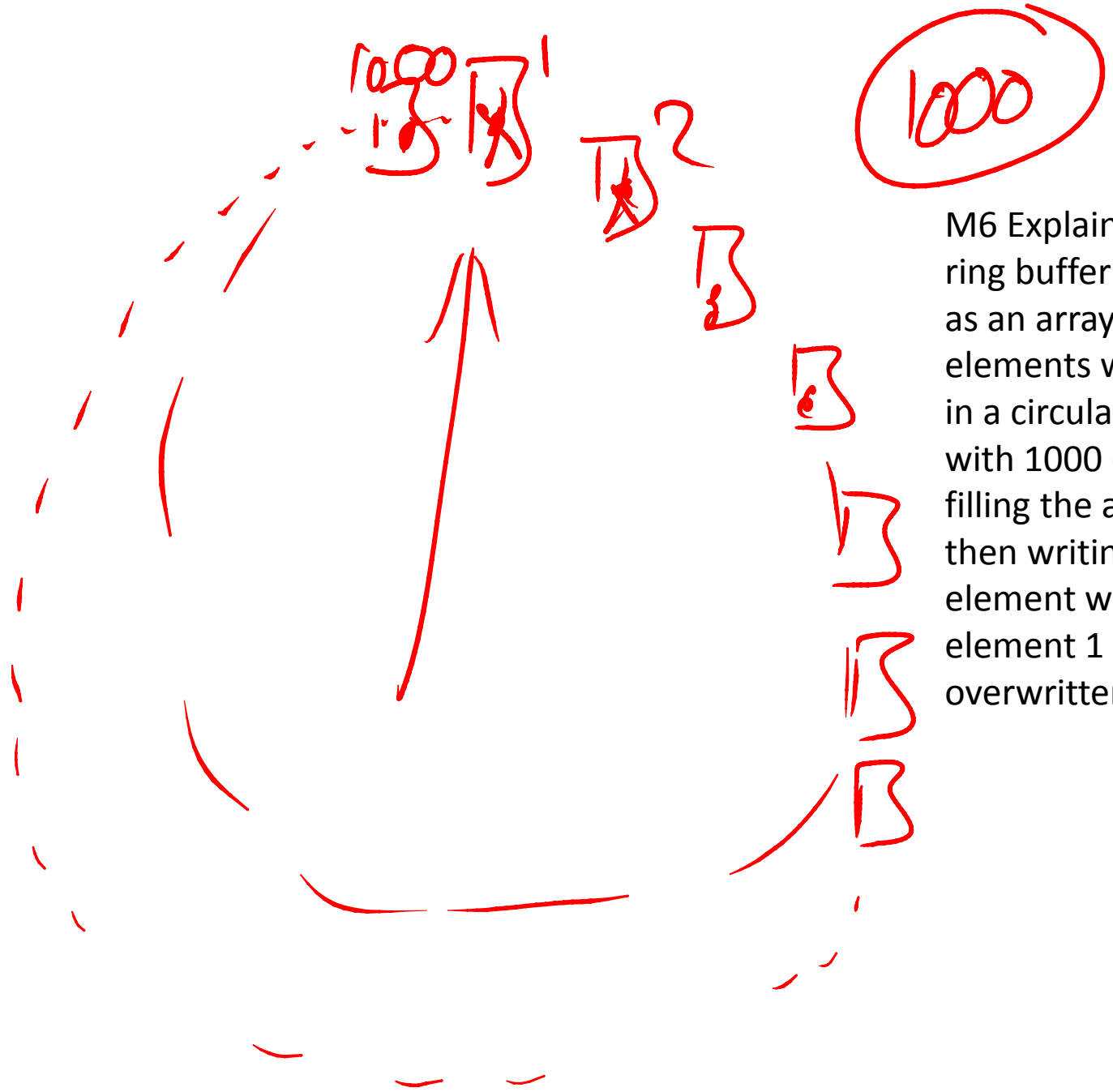




M1S34 Explaining how adding more tempdb data files leads to round-robin allocation and a spread of the contention to multiple files, leading to lower overall contention.



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.



M6 Explaining what a ring buffer is. Think of it as an array of data elements which is filled in a circular fashion. E.g. with 1000 elements, filling the array and then writing the 1001st element will cause element 1 to be overwritten.