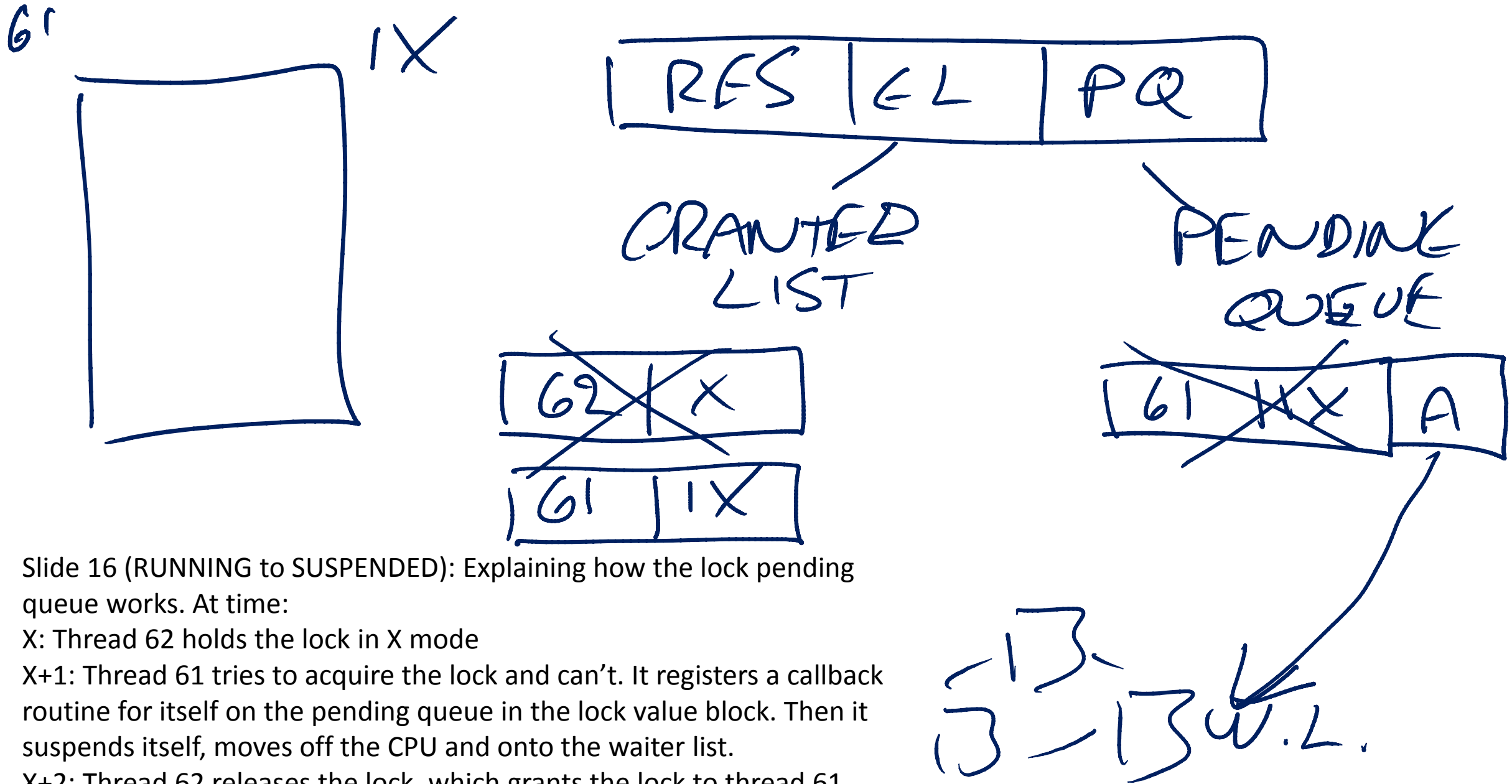
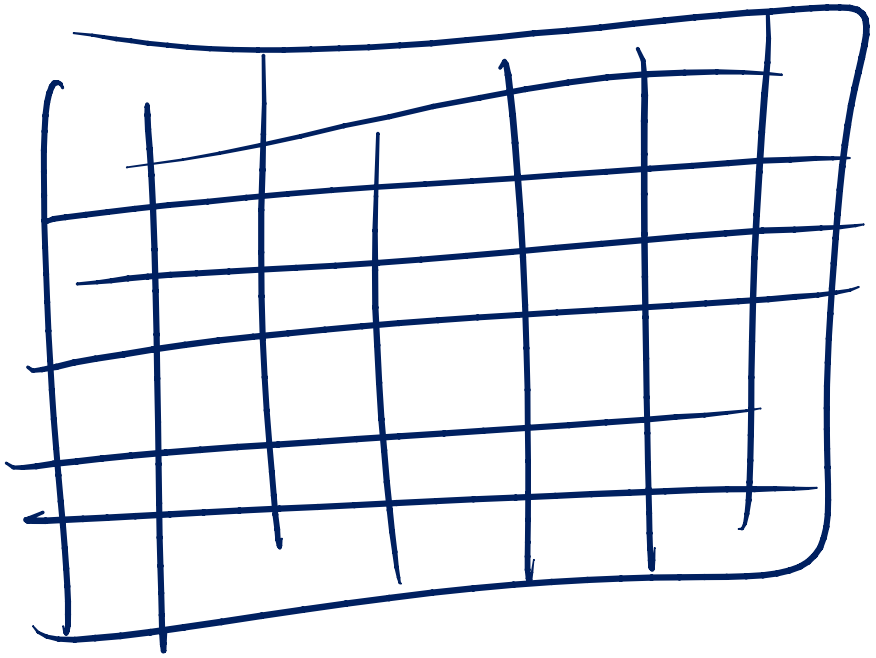
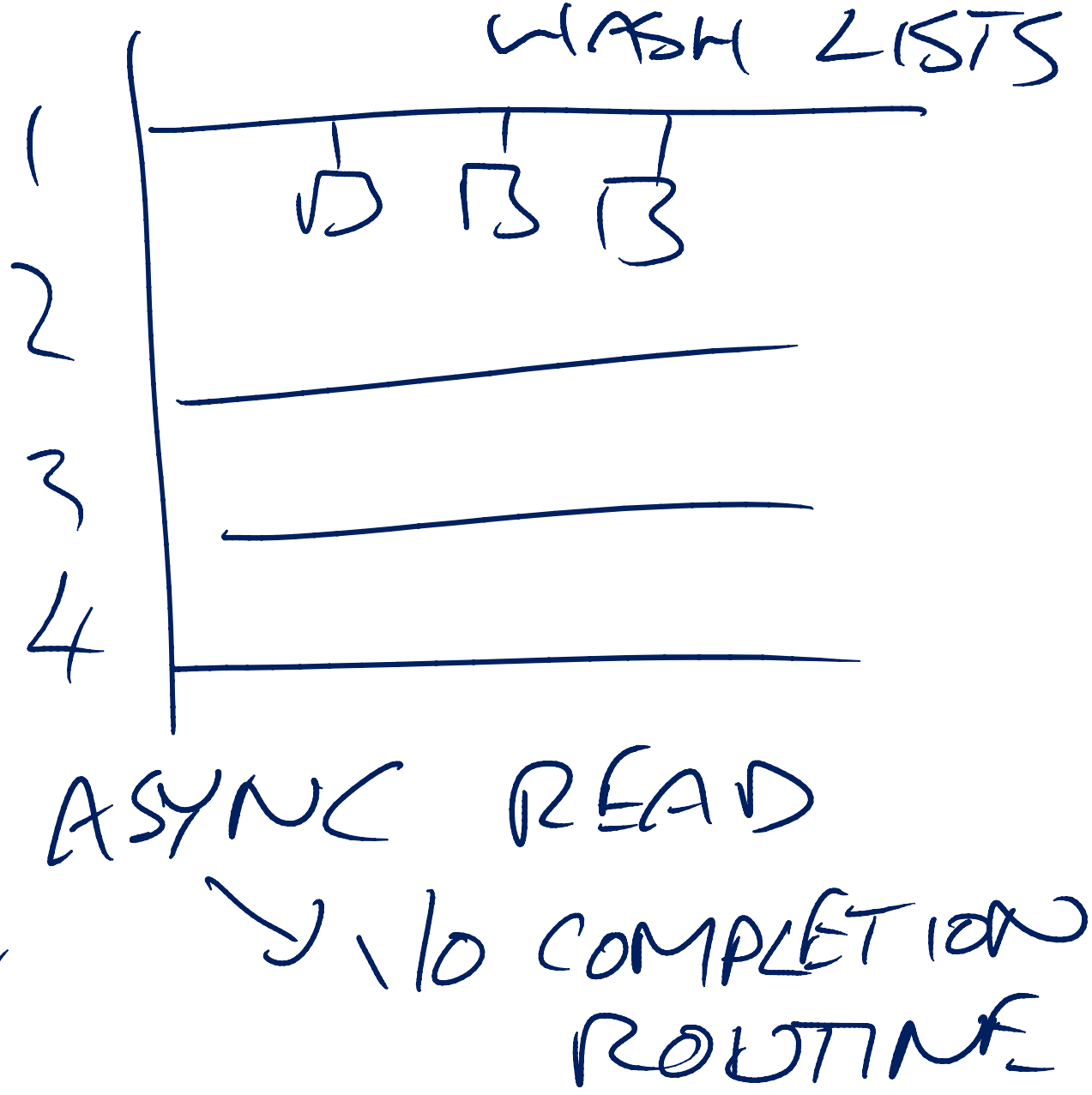


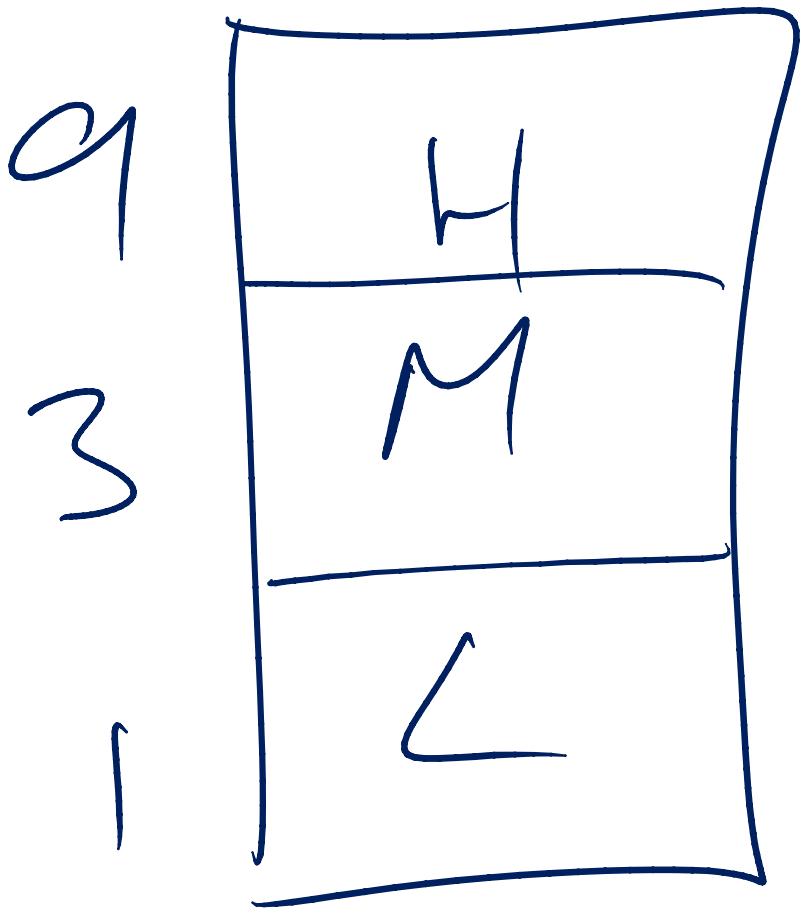
Slide 16 (RUNNING to SUSPENDED): Explaining how the lock pending queue works. At time:
X: Thread 62 holds the lock in X mode
X+1: Thread 61 tries to acquire the lock and can't. It registers a callback routine for itself on the pending queue in the lock value block. Then it suspends itself, moves off the CPU and onto the waiter list.
X+2: Thread 62 releases the lock, which grants the lock to thread 61. Thread 61's callback routine is called, signaling it and allowing it to move to the runnable queue on it's scheduler.



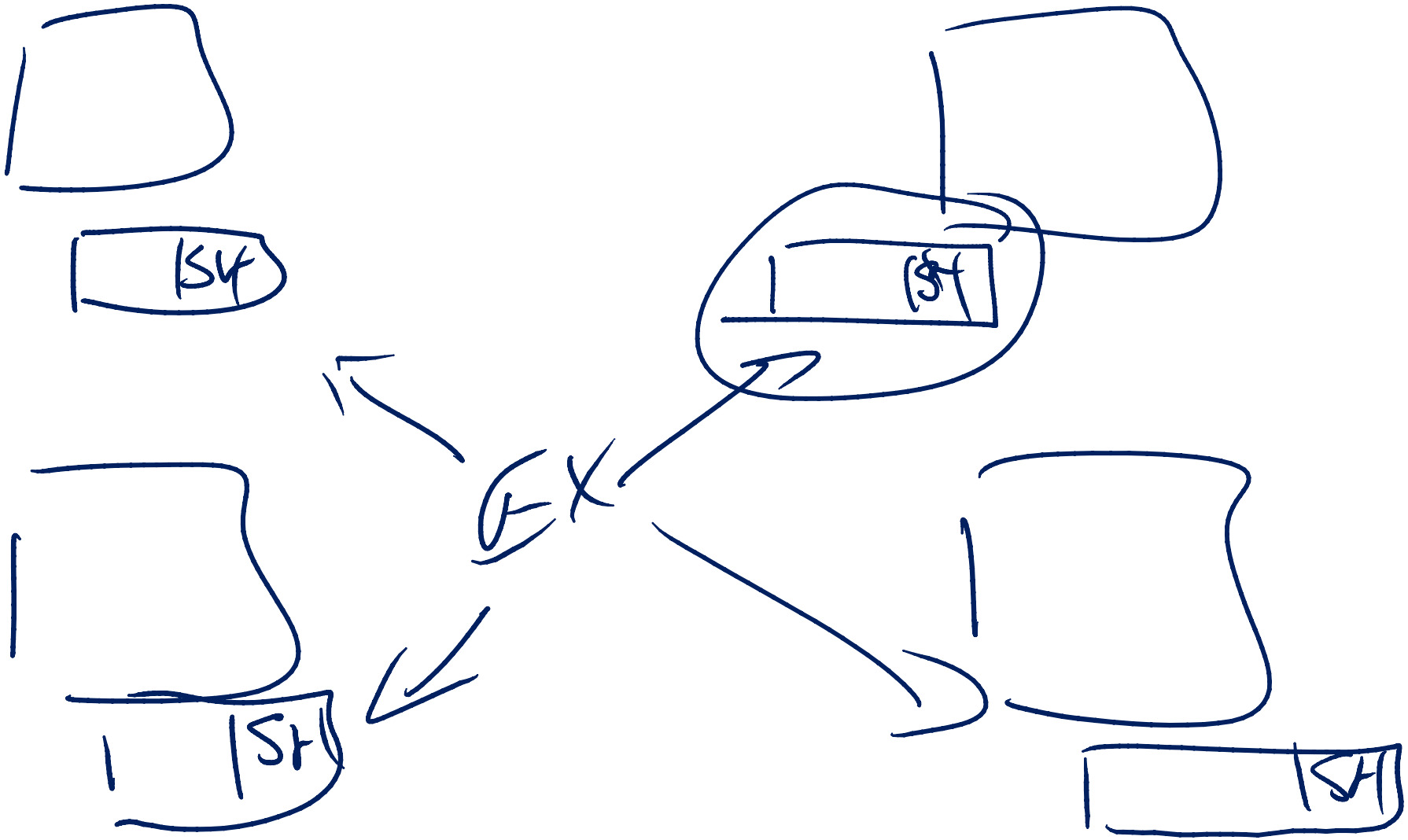


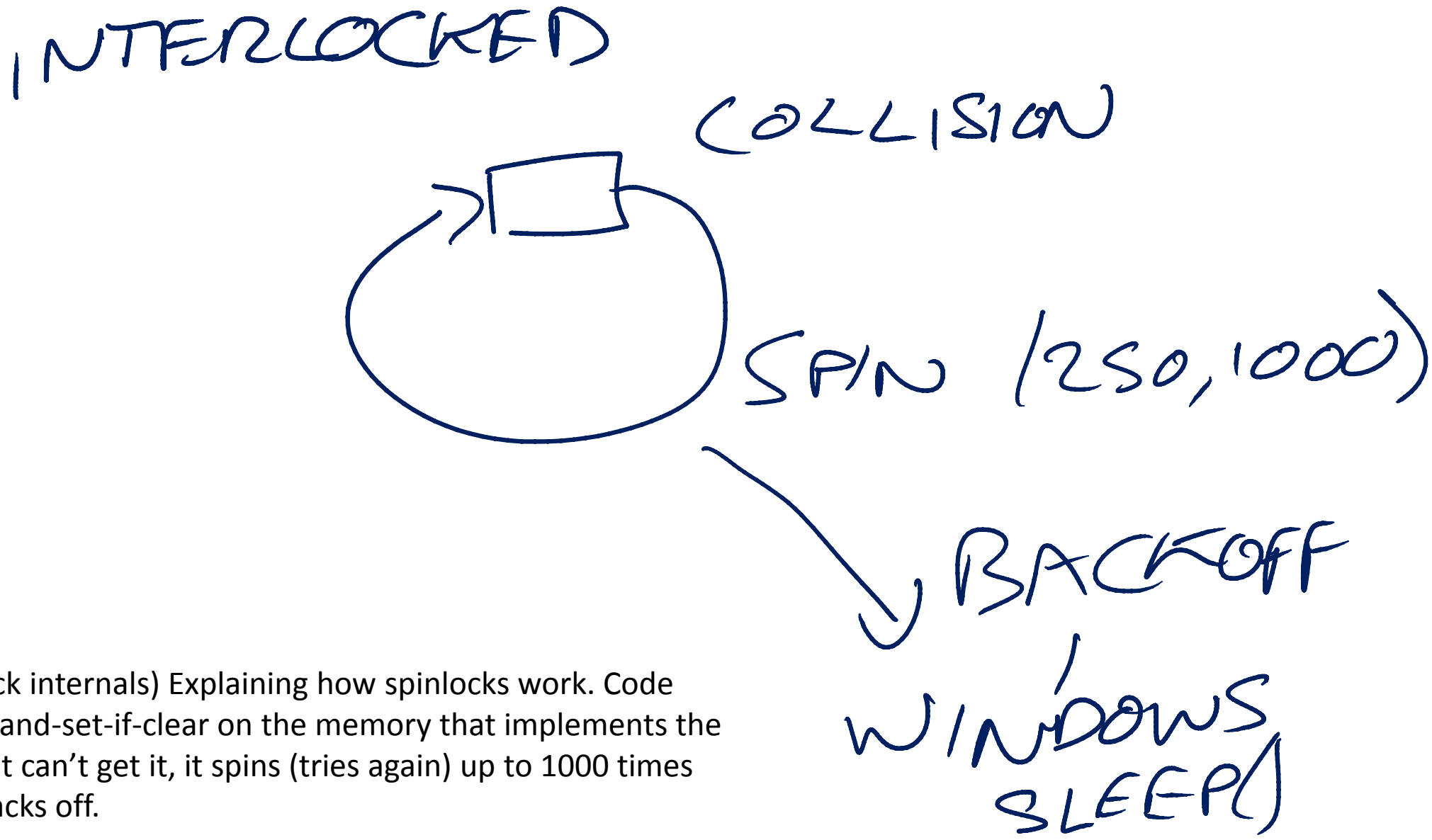
S16 Explaining the hash lists per database that allow the buffer pool to easily determine whether a page is already in memory or not. The thread starts the async read and another thread signals it.

S20 (Runnable Queue)
Explaining how the (unusable) Resource Governor changes to the runnable queue work. High/medium/low priority workload groups can be assigned to a resource pool. The runnable queue then allows 9 high-priority threads for every 3 medium threads for every 1 low thread.

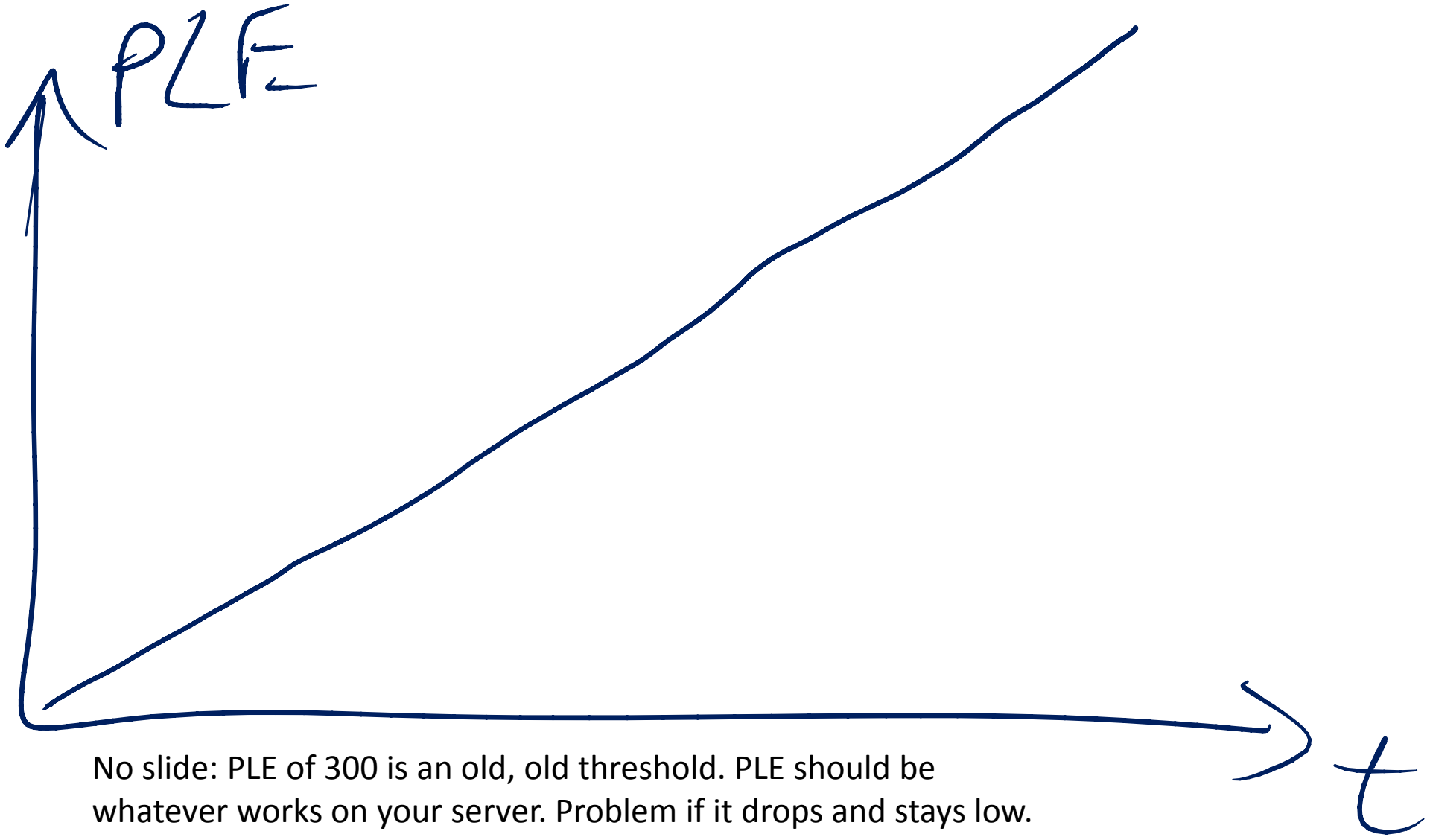


S39 (Superlatches) Explaining how superlatches work. SH latch gets promoted to one copy per scheduler. Collapses back down to regular latch if EX latch occurs.

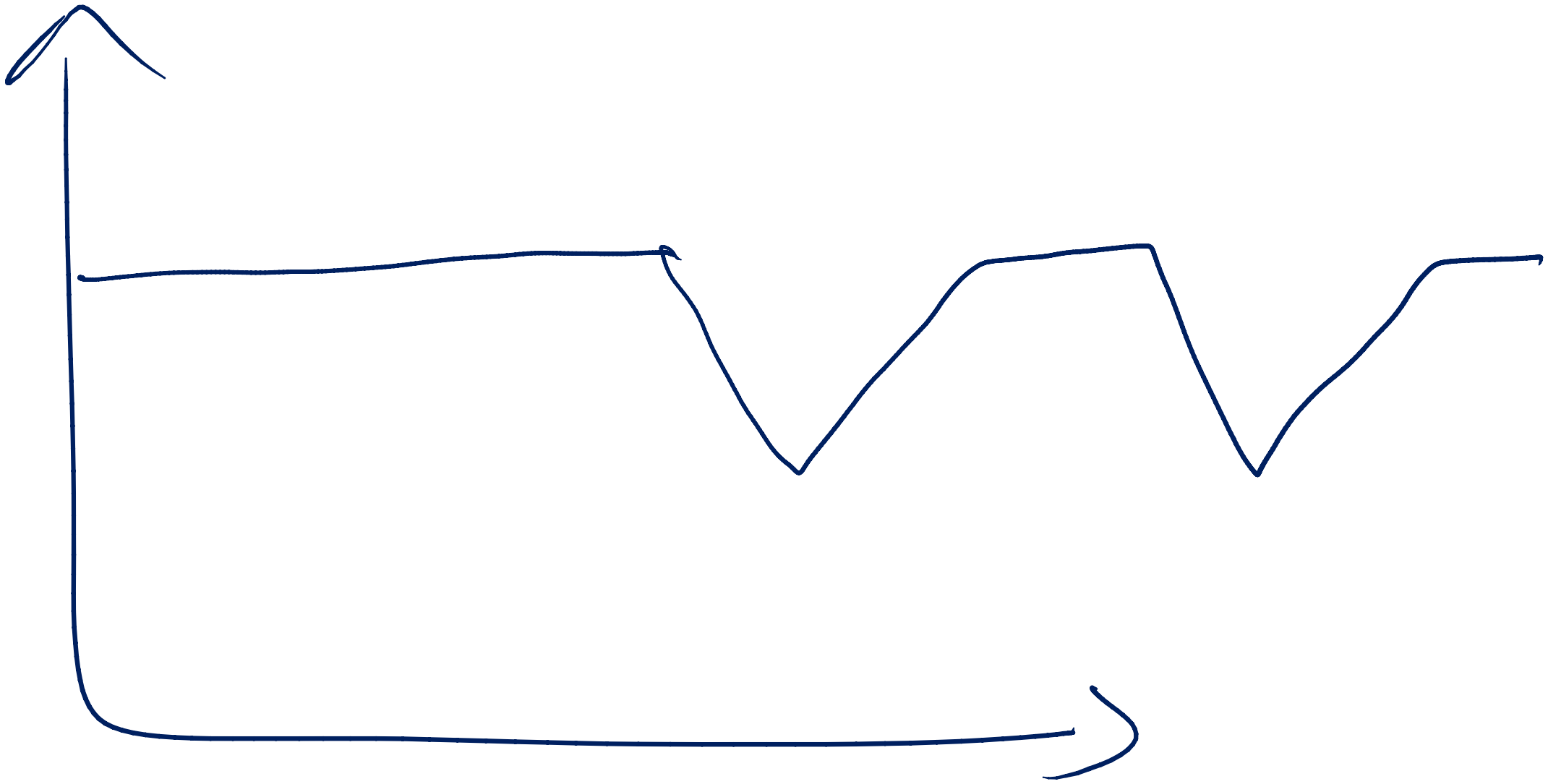




S44 (Spinlock internals) Explaining how spinlocks work. Code does a test-and-set-if-clear on the memory that implements the spinlock. If it can't get it, it spins (tries again) up to 1000 times and then backs off.

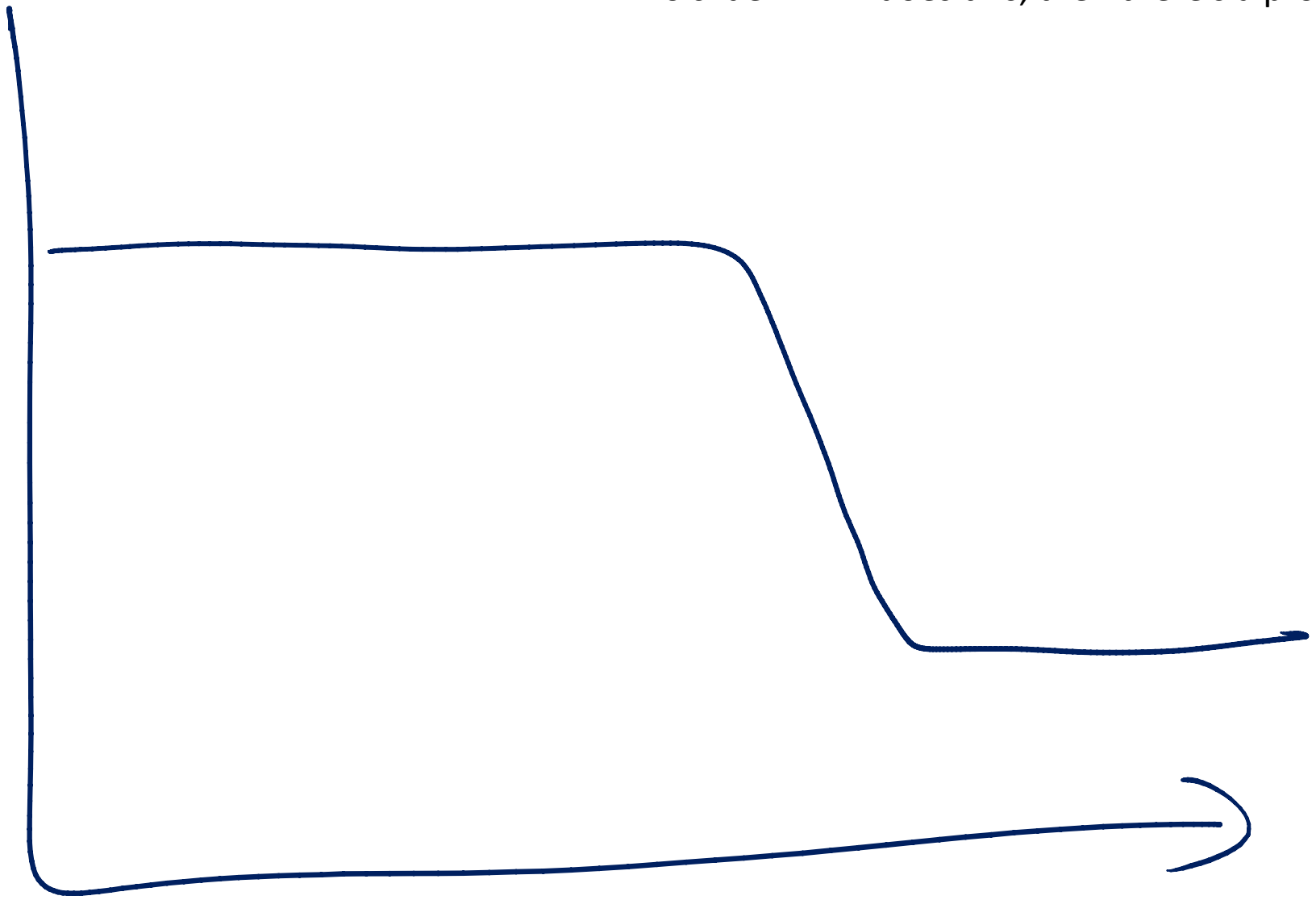


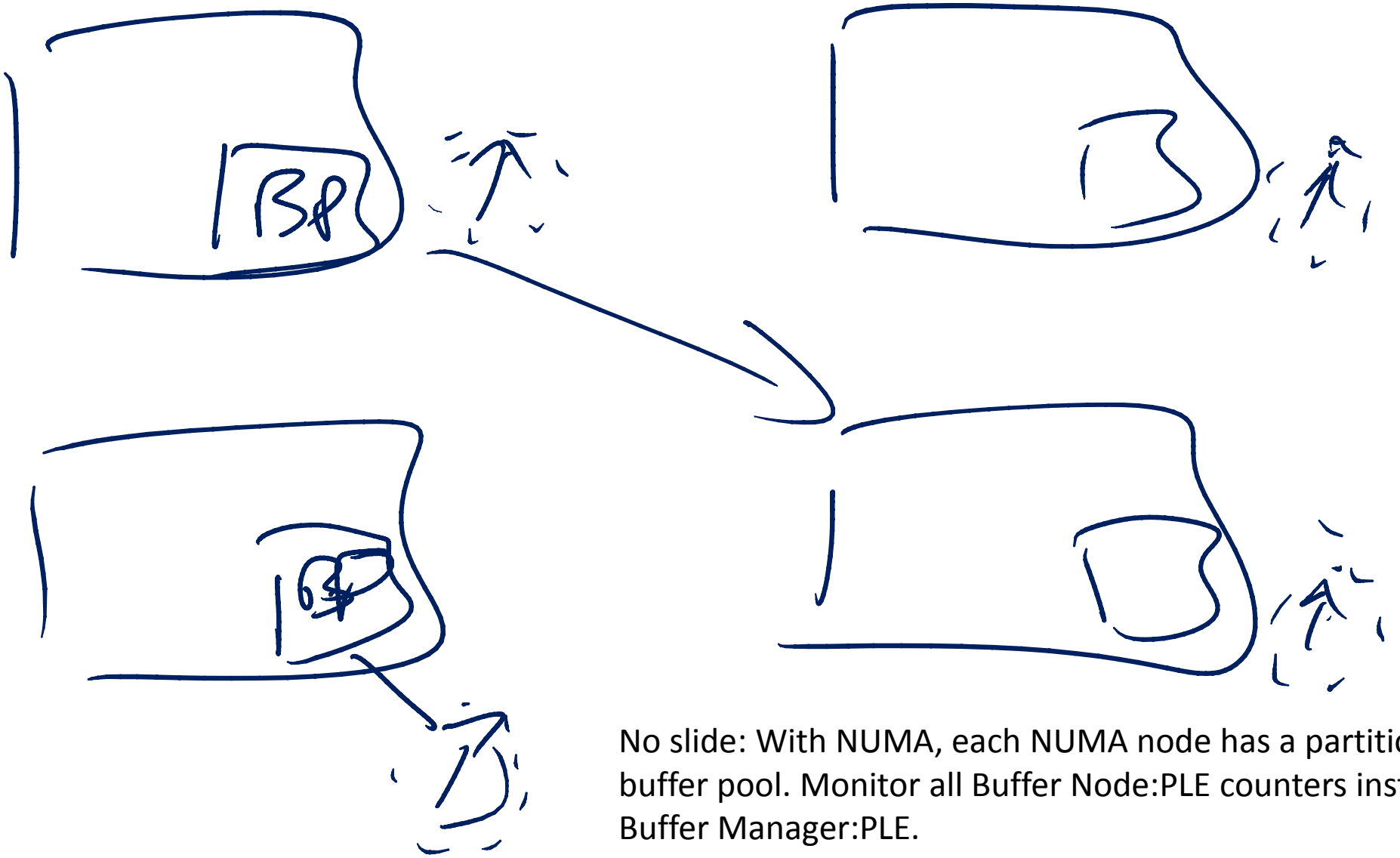
No slide: PLE of 300 is an old, old threshold. PLE should be whatever works on your server. Problem if it drops and stays low. Normal for it to drop and come back up again after some operations. It will tick up by one per second without memory pressure



No slide: If PLE does this, that’s normal

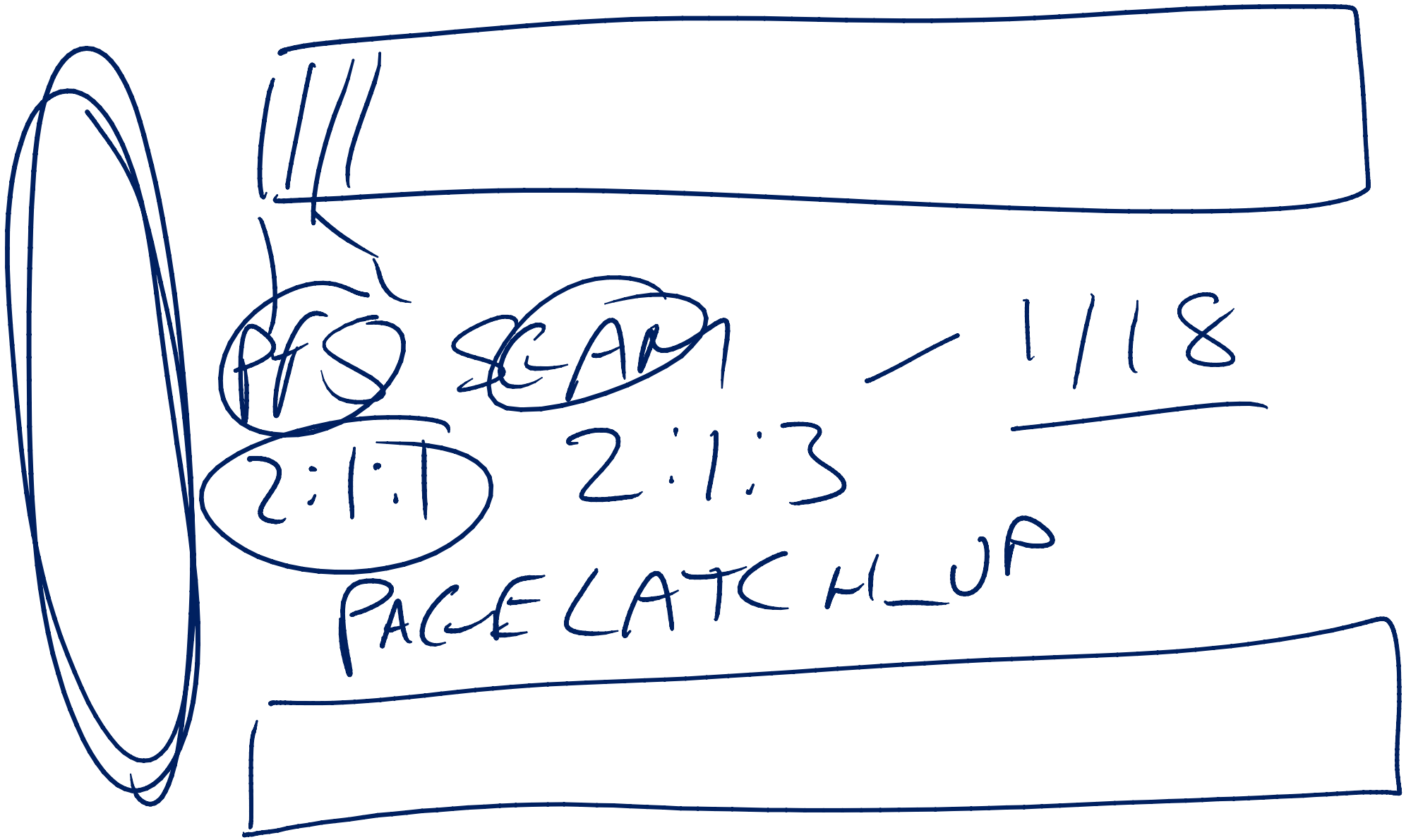
No slide: If PLE does this, then there’s a problem

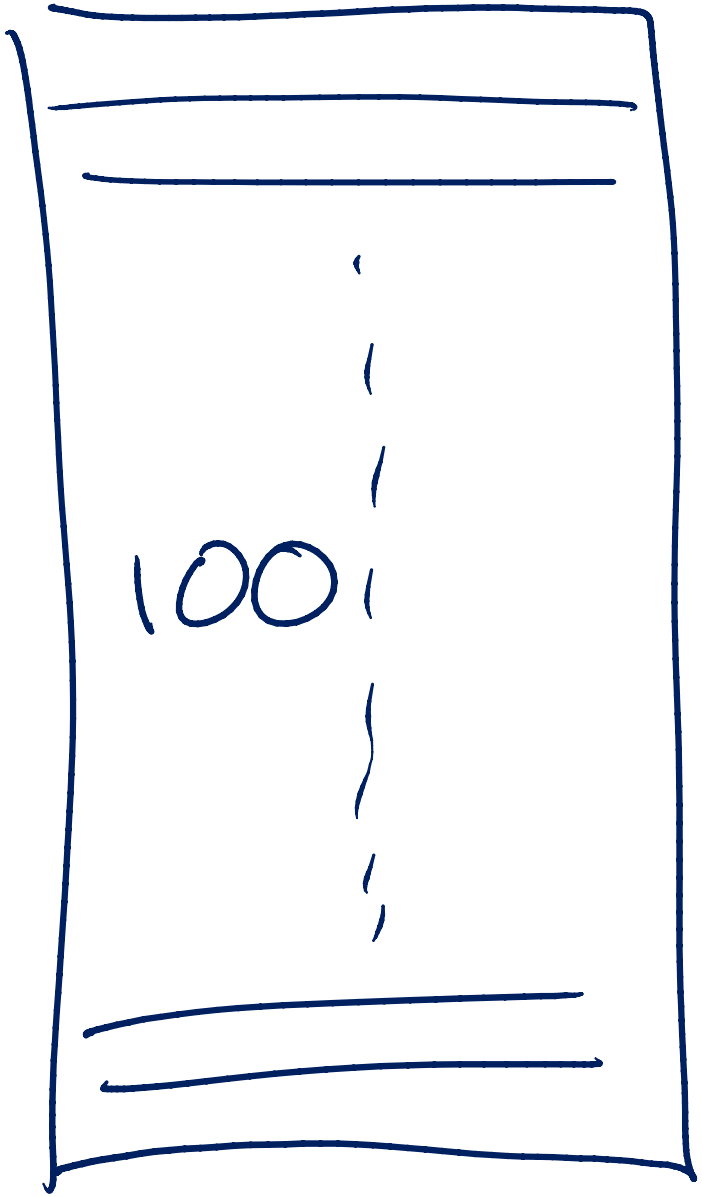




No slide: With NUMA, each NUMA node has a partition of the buffer pool. Monitor all Buffer Node:PLE counters instead of Buffer Manager:PLE.

S52 (PAGELATCH_XX Wait) Explaining tempdb allocation bitmap contention from the demo. See http://www.sqlskills.com/blogs/paul/correctly-adding-data-files-tempdb/





INT , IDENTITY

80

S52 again. Explaining insert hotspot. Lots of small rows on a page with an identity column and concurrent inserters. Only one thread can be modifying the page.