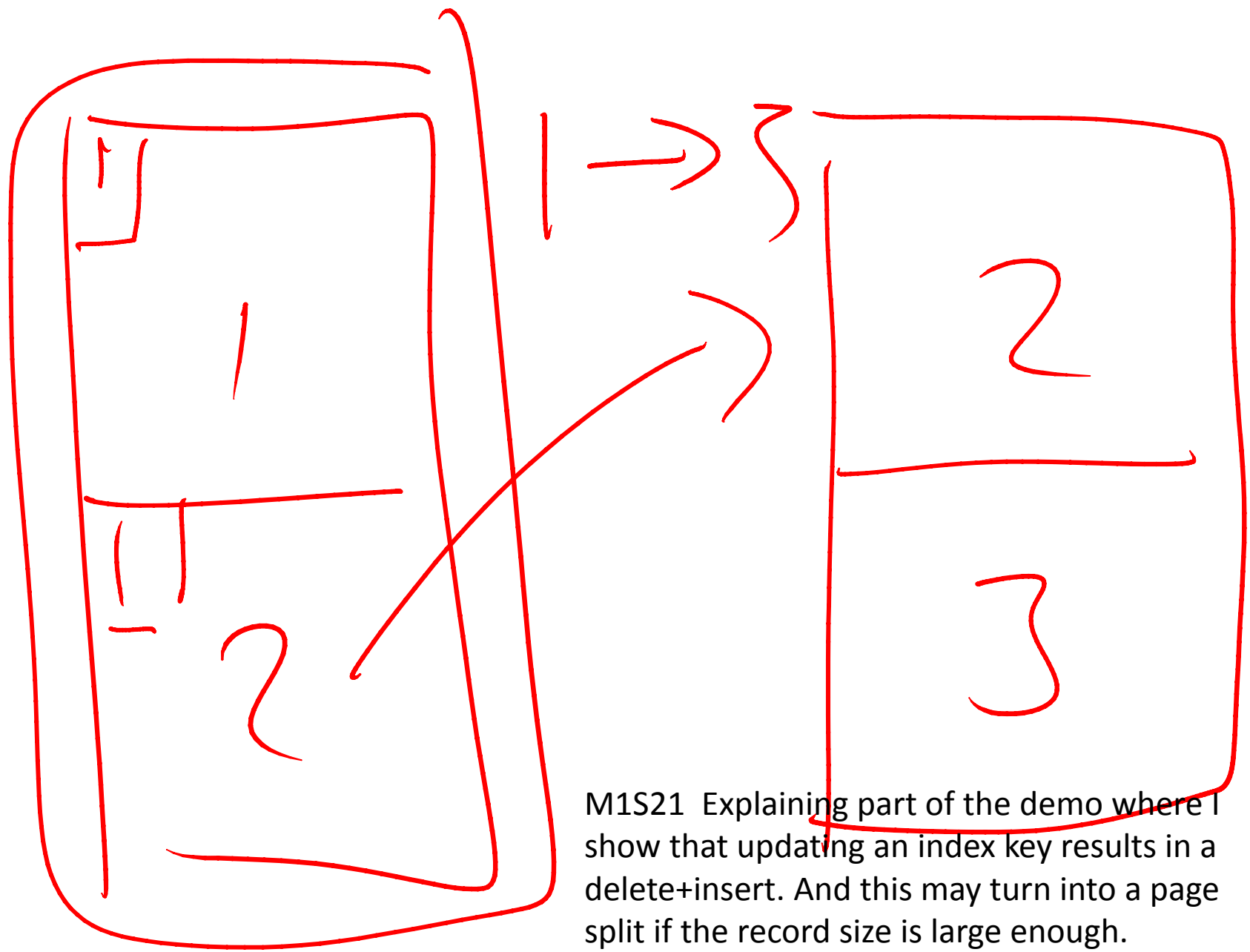


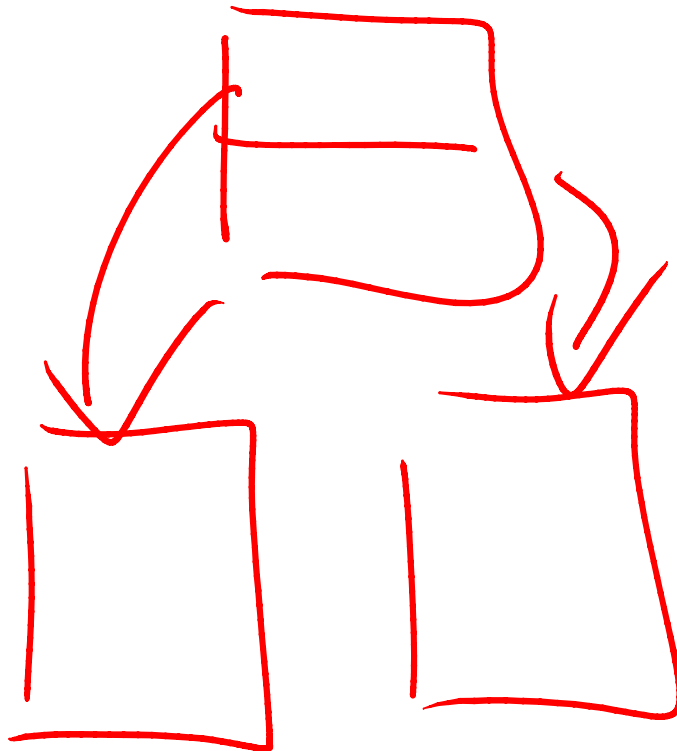


M1S18-19 Discussing how log records are flushed to disk by first being copied into a log buffer and then that buffer is flushed to disk.



M1S21 Explaining part of the demo where I show that updating an index key results in a delete+insert. And this may turn into a page split if the record size is large enough.

Hmm – not sure what this was.
Guessing it was maybe making the
point that a page split at the leaf level
may result in a cascading page split
right up to the root page (like M2S14
shows)



WEIGHTINGS

~~8100~~ - 8099

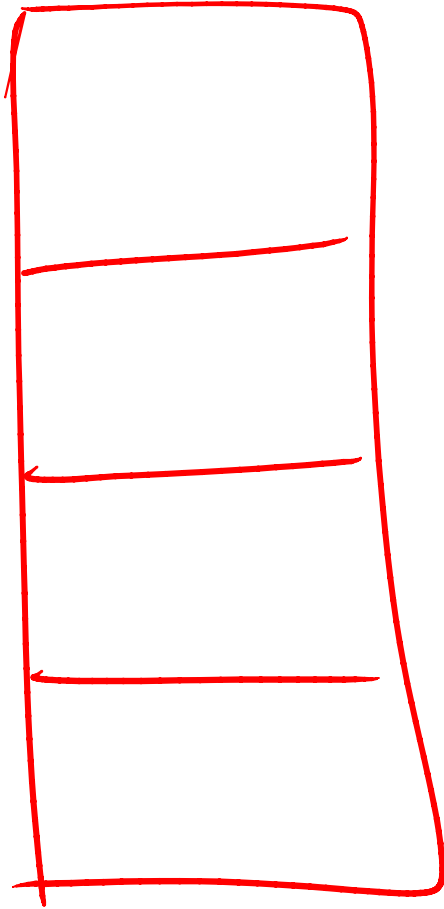
~~8100~~ 8099

~~8100~~ 8099

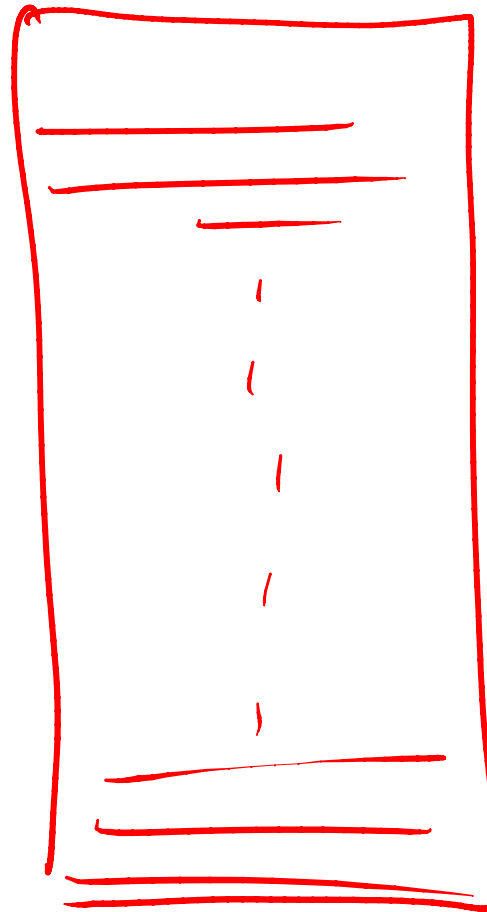
M1S22 Showing round-robin
allocation and proportional fill
weightings.



2000/4



250/32



M1S24 Discussing how record size can lead to different data densities on data pages and so lead to inefficiencies during query processing. Consider storing rarely-used LOB data off-row to increase data density.

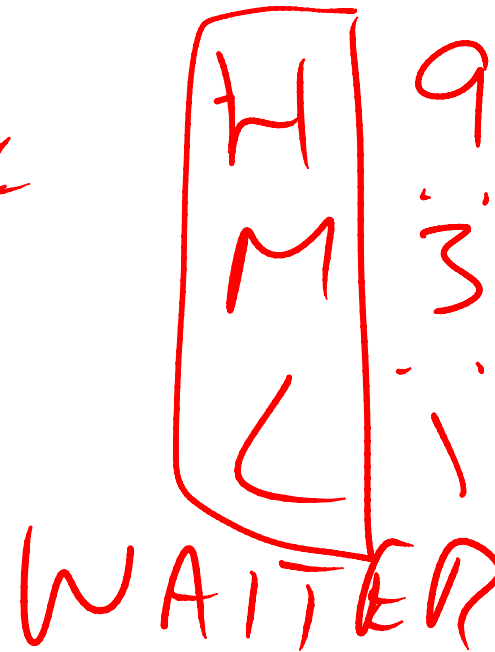


M1S32-34 Explaining PFS and SGAM contention in tempdb. Adding another data file reduces contention because of round-robin allocation.

WORKLOAD GROUPS

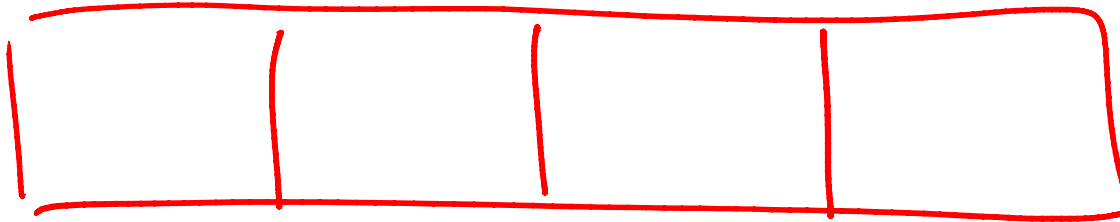
Running

Runnable

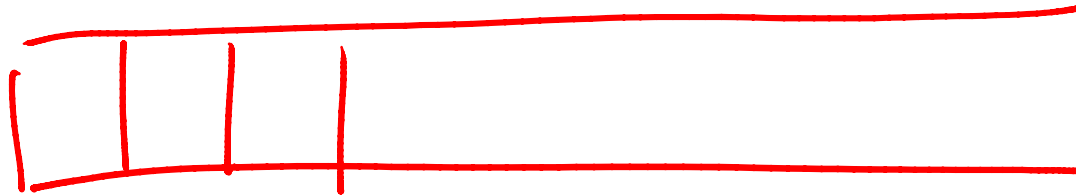


M2S9 Explaining how the Runnable Queue is a true FIFO (First-In, First-Out) queue unless Resource Governor workloads groups and group priorities are being used. High-Medium-Low priority equals 9-3-1 threads through the Runnable Queue.





$$DOP = 4 \quad 4 + 1$$



M2S24 Explaining how CXPACKET waits occur. In a parallel operation, the coordinator thread always incurs a CXPACKET wait. If the threads are not given an equal amount of work to do (the bottom illustration), some will finish early and incur CXPACKET waits until the longest-running thread(s) complete.