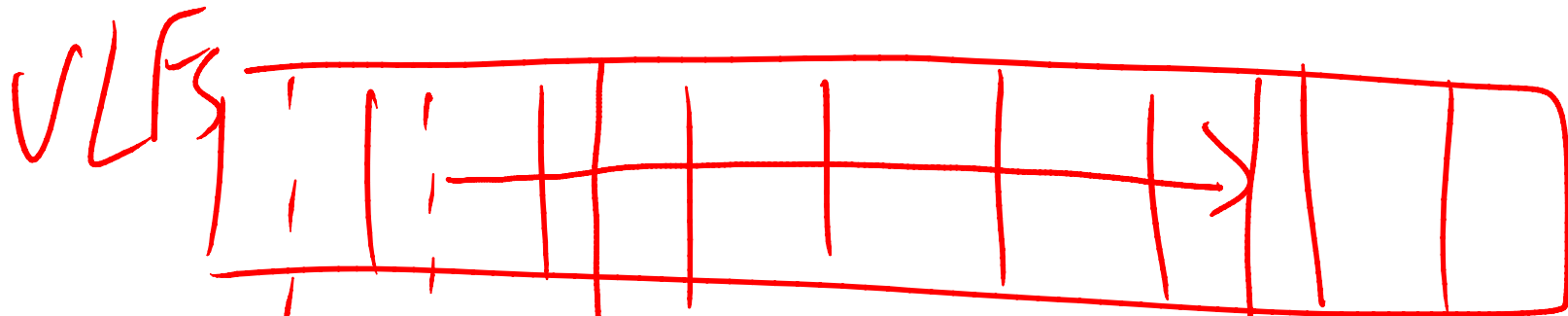


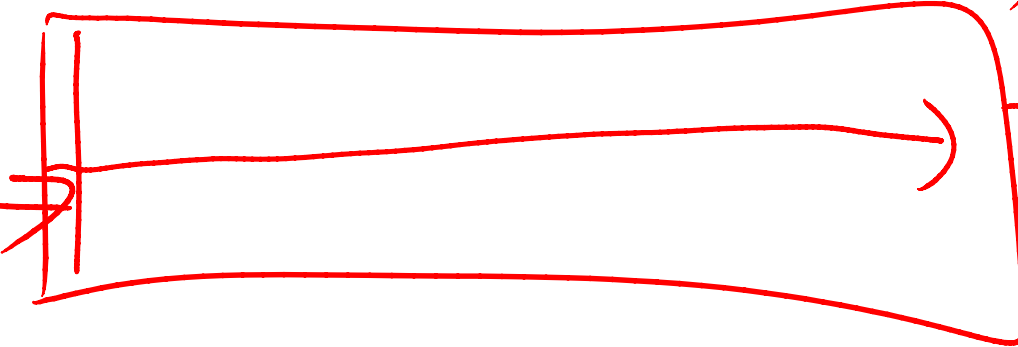


~~LSN~~
 LRAJ
 LSN
 CHKPT
 LSN
 BACKUP
 verbose logging

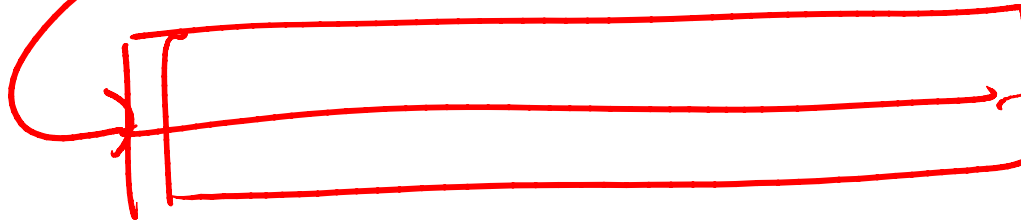
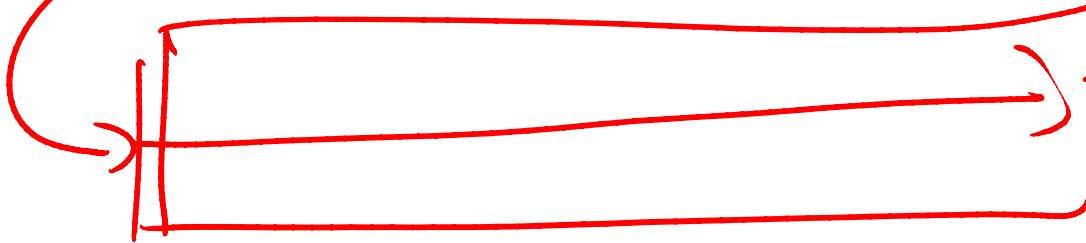
M1 Explaining
 how the tran
 repl log reader
 agent job (LRAJ)
 notifies the log
 mgmt system of
 the LSN to which
 is has read the
 log.

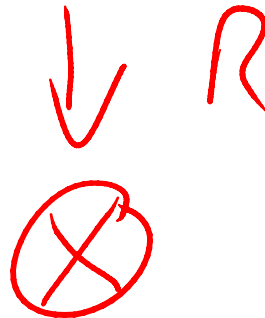
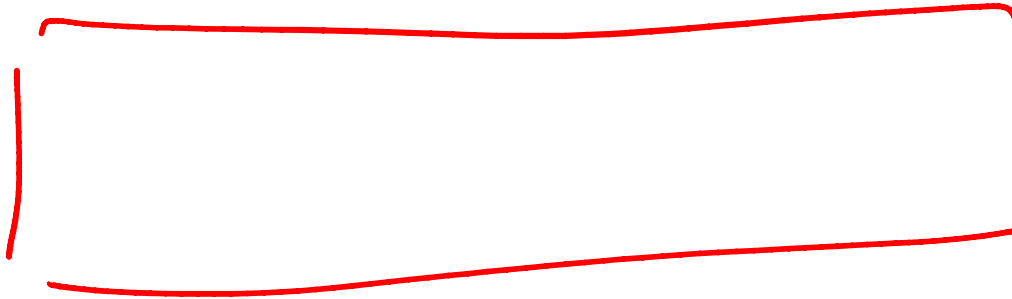
LSN
 / MAX
 LSN
 LRAJ / BACKUP

FILEHEADER (PAGE 0)

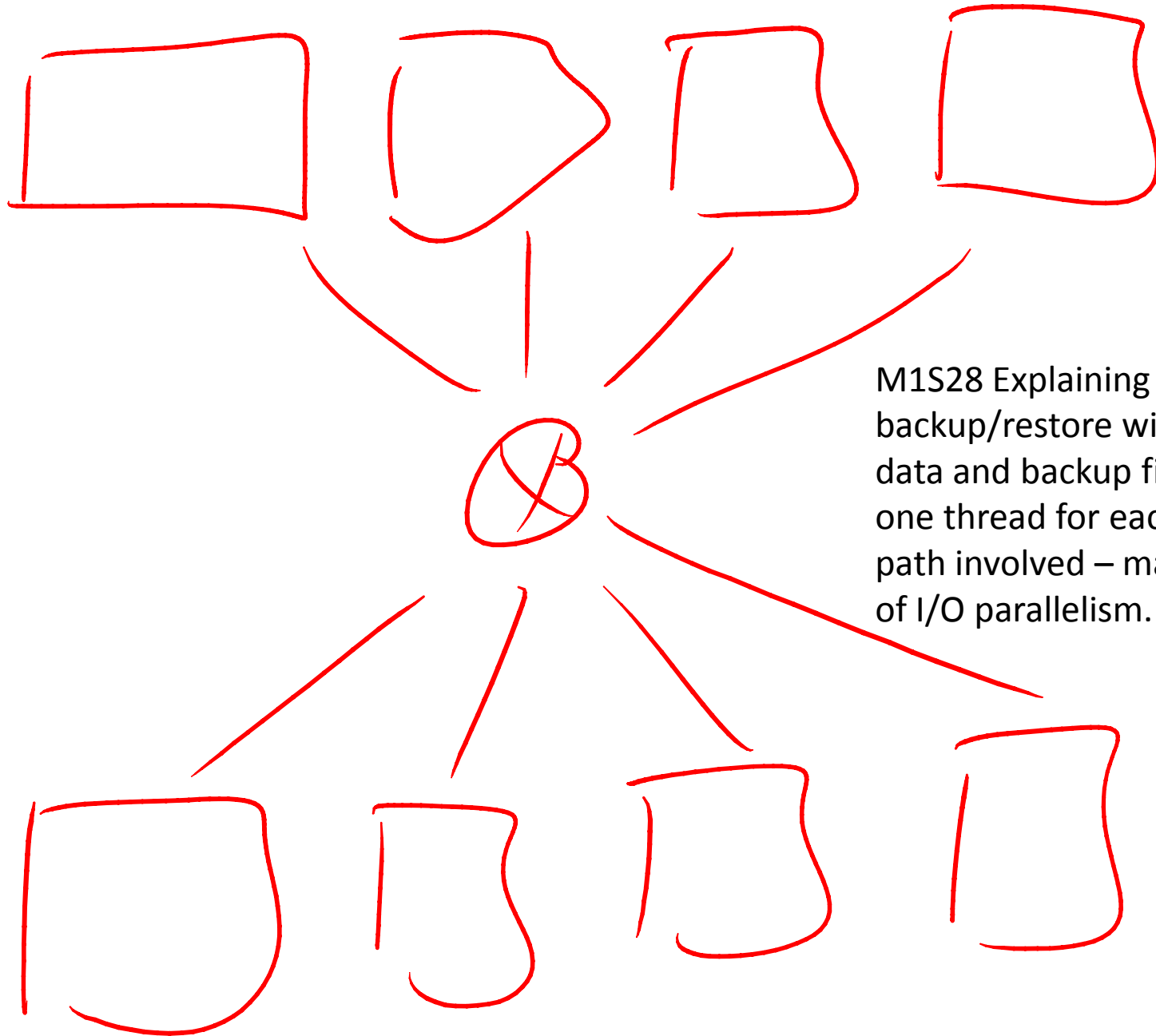


M1 Log files are never used in parallel. You will see writes to the fileheader pages in all log files though.





M1S28 Explaining the backup/restore with a single data file and a single backup file has one reader thread and one writer thread.



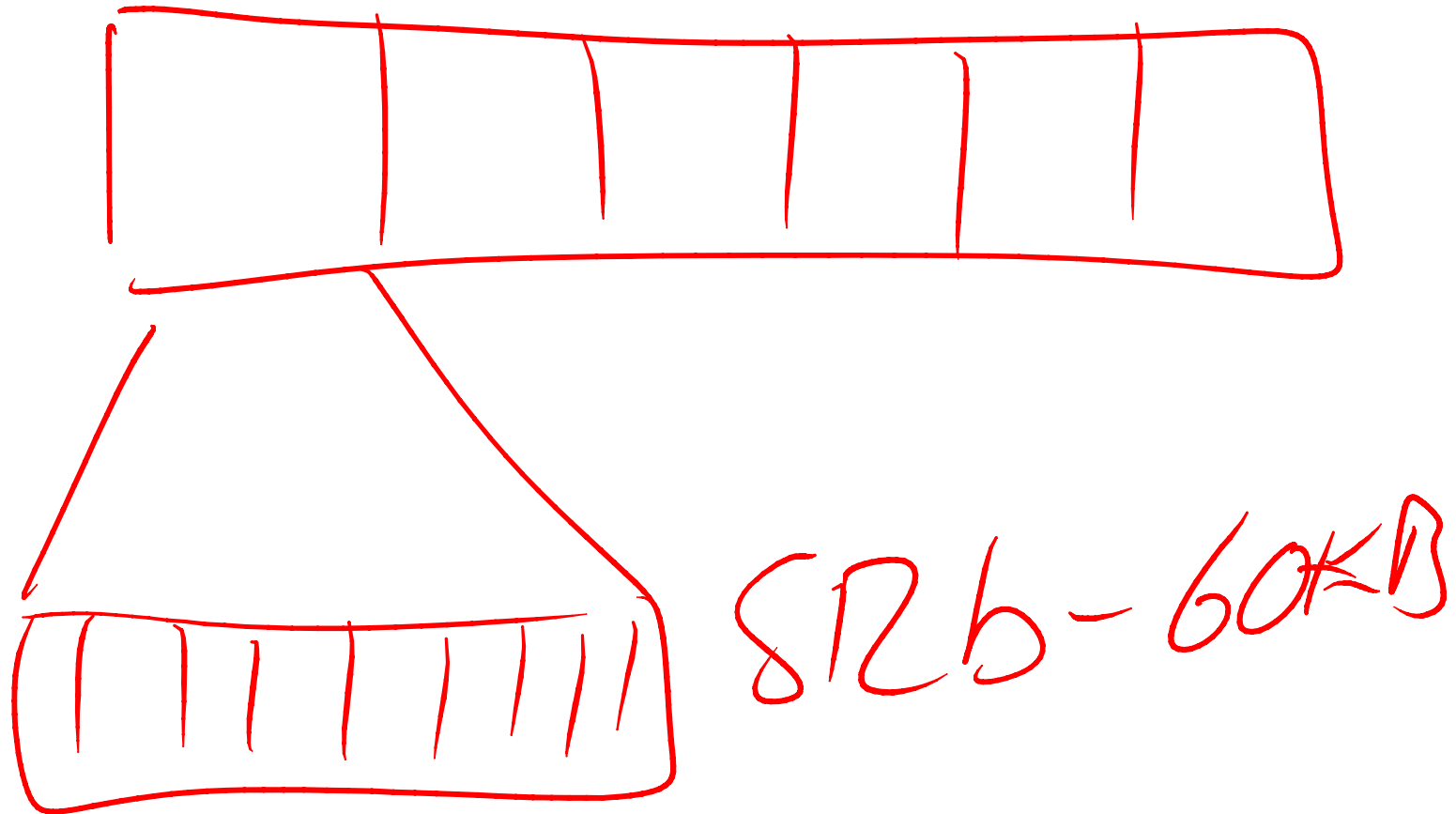
M1S28 Explaining the backup/restore with multiple data and backup files has one thread for each physical path involved – making use of I/O parallelism.

TAB: 100K - CLUS

7000 PAGES

FULL + REBUILD

M1 Discussing full logging doesn't mean one log record per operation. Index rebuild of a 100,000 row index in full recovery model, where 14 rows fit on a page will actually log a full page image rather than 14 inserts, per page. More efficient and still fully logged.



M1 Explaining that a VLF is split into variable-sized log blocks. A log block 'ends' when it fills to 60KB or when a transaction commits or rolls back. Having lots of tiny transactions leads to more log flushes of smaller log blocks. Consider making transactions bigger if log throughput is a bottleneck.

2012

fs1 | ☐ In_row

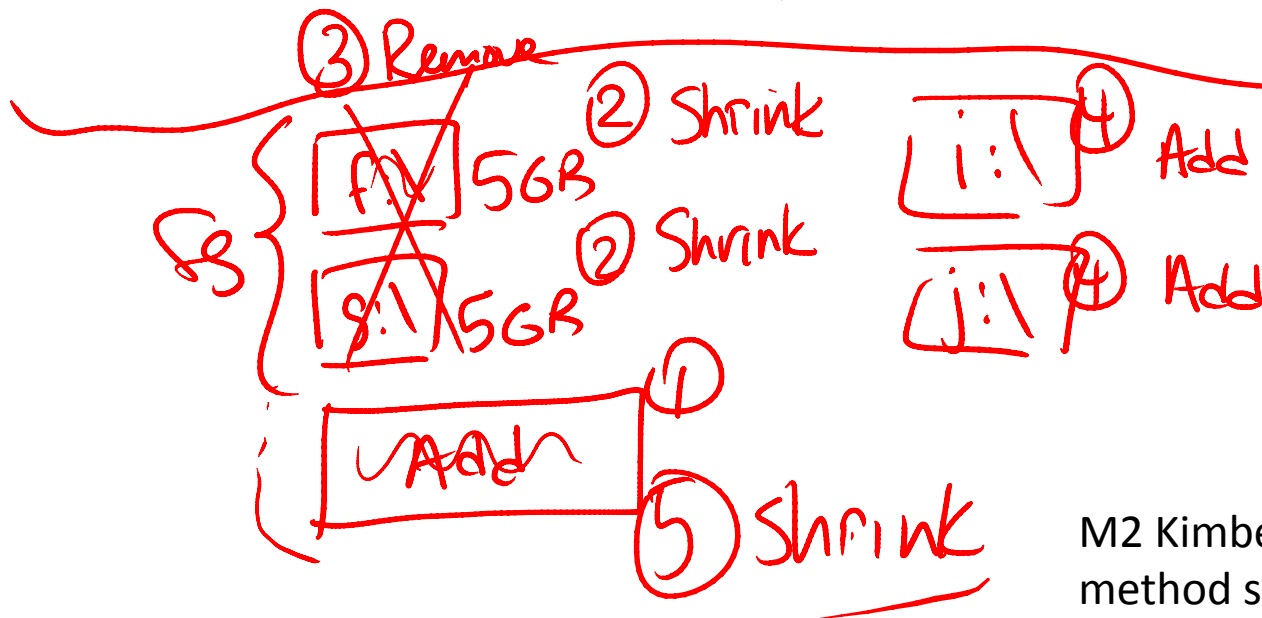
fs2 | ☐ LOB

to text image-on

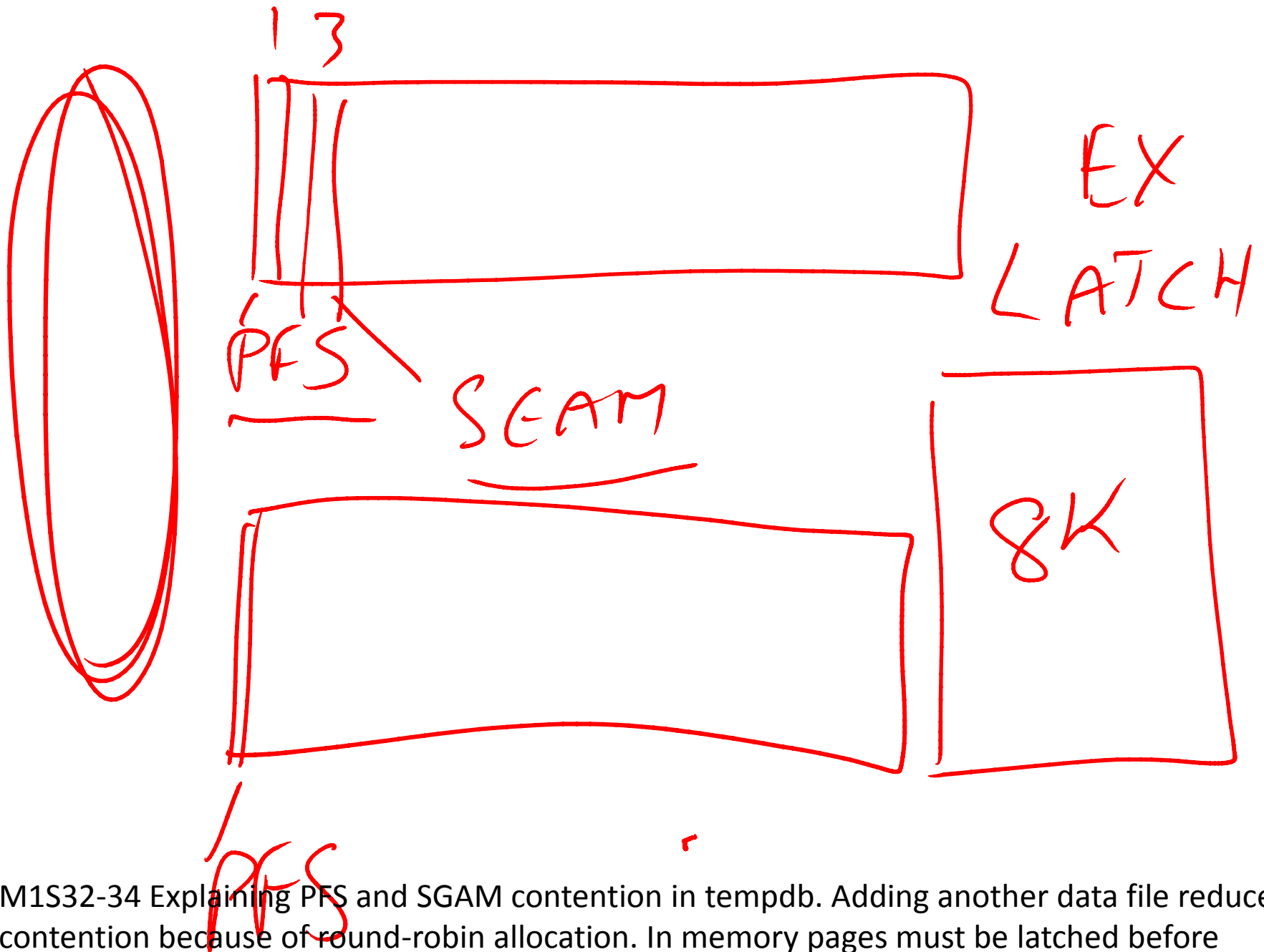
npt → pt

pt → pt
Sch1 Sch2

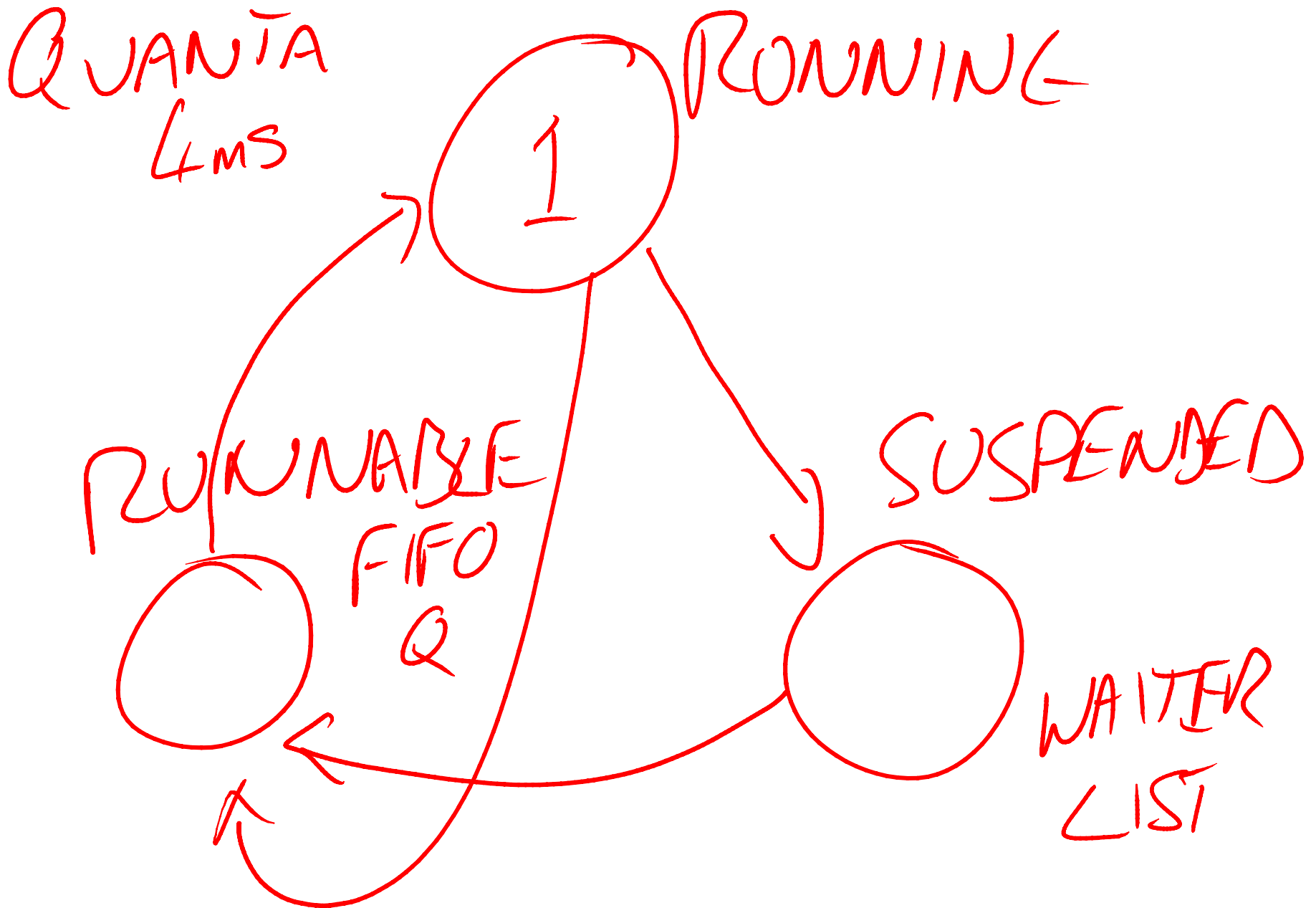
} move
LOB
data



M2 Kimberly explaining the method she blogged about (on our SQL Mag blog) to allow LOB to be moved.



M1S32-34 Explaining PFS and SGAM contention in tempdb. Adding another data file reduces contention because of round-robin allocation. In memory pages must be latched before being changed.



NES.COV

HIGH 9

MED 3

LOW 1

Run.Q

SPID61-M

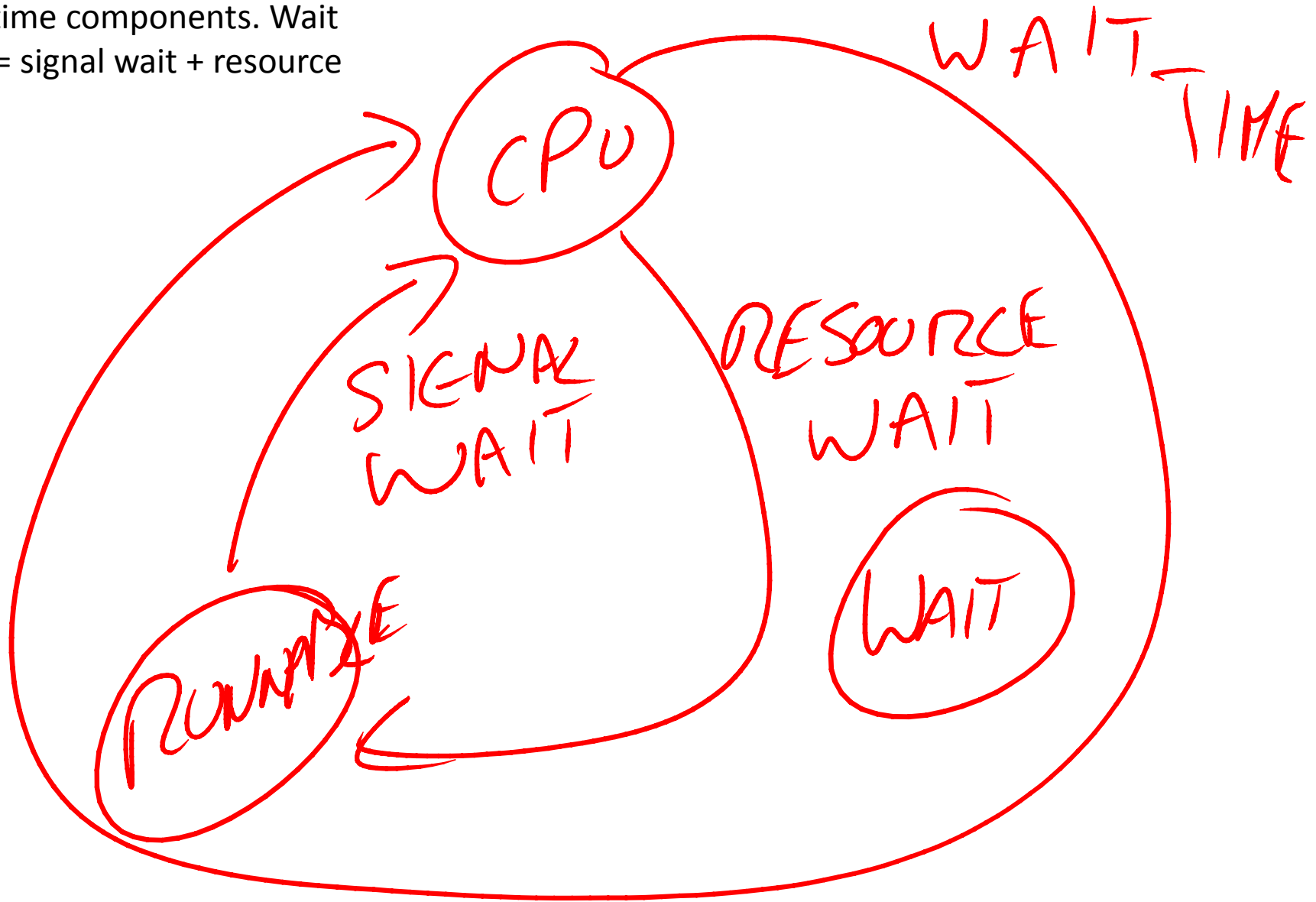
SPID57-L

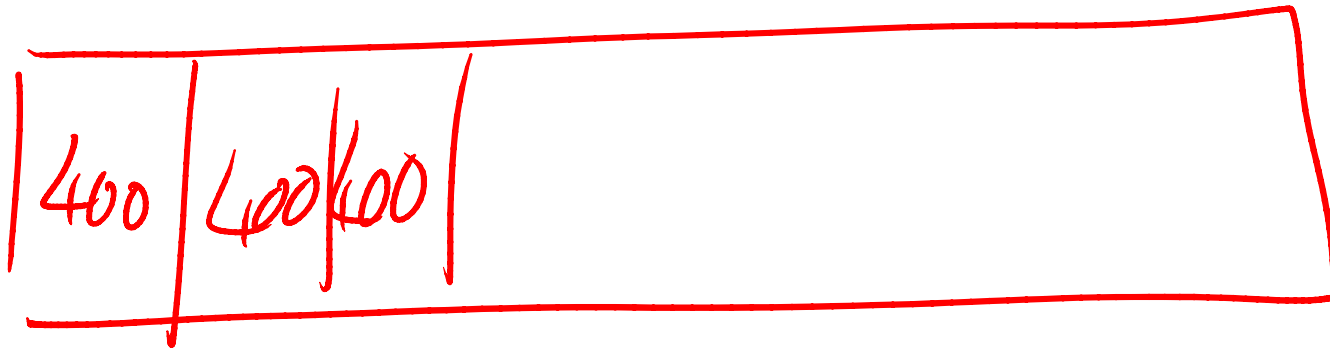
SPID63-L

SPID64

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.

M1S11 Explaining the various wait time components. Wait time = signal wait + resource wait





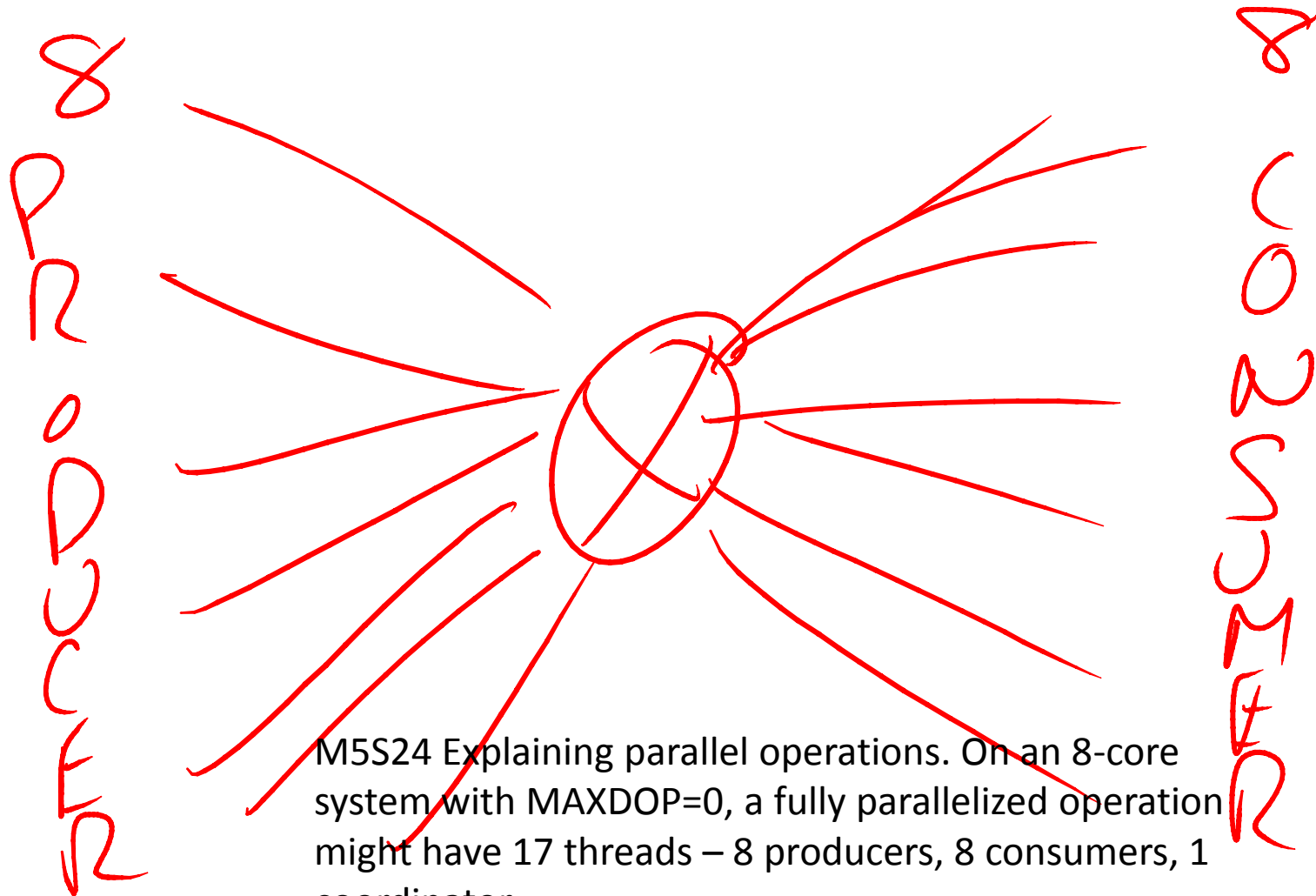
M5 Explaining how a cluster index with an int identity column can lead to hotspots. With several hundred concurrent inserters, each leaf page has contention for the EX latch to allow it to be altered.

See the hotspot demo.

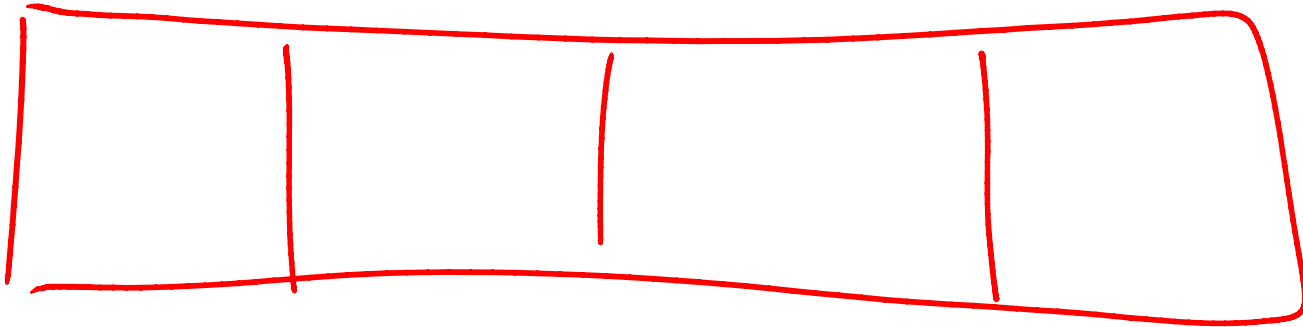
B1 C-INT CC KEY
20 bytes
400 rows/page

DOP=8, 17 THREADS

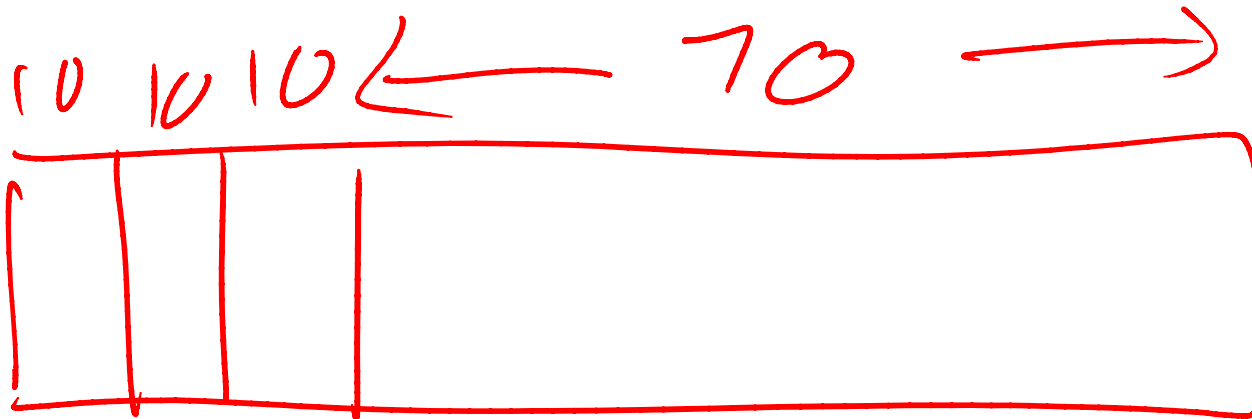
1 x COORDINATOR



M5S24 Explaining parallel operations. On an 8-core system with MAXDOP=0, a fully parallelized operation might have 17 threads – 8 producers, 8 consumers, 1 coordinator.



M5S24 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.



MORE RESTRICTIVE



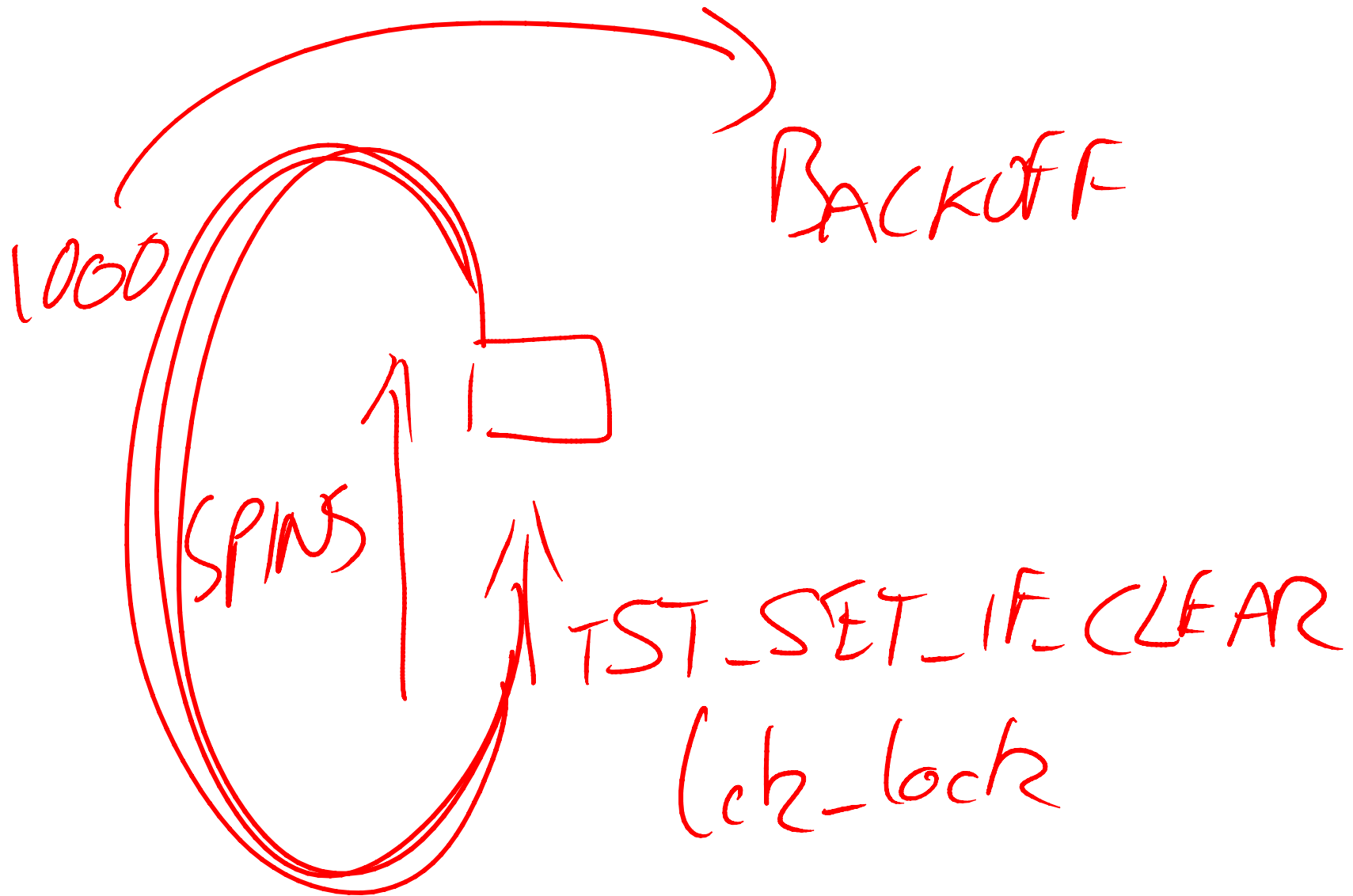
M6 Discussing how to limit MAXDOP.

Instance MAXDOP is least restrictive as it can be overridden by a query hint, but only up to the limit imposed by Resource Governor.

RG WORKLOAD GRP
MAXDOP

QUERY HINT MAXDOP

INSTANCE MAXDOP



M6 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.

100

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 100 elements, filling the array and then writing the 101st element will cause element 1 to be overwritten.

