DENSITY

→ WCAB: Write congestion aware bypass

→ VHC: Virtual Hybrid cache

SRAM:
- → Low density
- → High leakage

Solution: NVM → STT RAM (Spin-Transfer-Torque)
→ ReRAM (Resistive RAM)

Issues:
- → Long write latency
- → This latency puts pressure on the queue ⇒ congestion

Issues with current solutions:
- → Reduces LLC density
- → Increased error rate ⇒ unreliable
- → Lowering write latency reduces the density advantages

Writes to LLC in inclusive Vs exclusive:

Inclusive: 1) LLC miss
2) Dirty victims from L2

Exclusive: 1) Hit in LLC → As they are made I and pushed to upper

2) Clean and dirty victims from L2.

(X) writes in exclusive ≫ writes in inclusive. But exclusion increase capacity.

WCAB: Write Congestion Aware Bypass:

→ Aim of existing bypass ⇒ Improve hit rates

→ As LLC capacity increases ⇒ The fraction of writes that won't be used, increases drastically.

Solution: More aggressive bypass

↓

But this would decrease hit rate.

Tradeoff
- → Reducing LLC congestion
- → Minimize hit rate loss

Goal
- → Reduce congestion
- → Minimize cost of bypass

Dynamic params
- → Liveness
- → Write fraction
- → Request bandwidth

Request Queue:

(Occupancy of request queue is high) + (fraction of writes is above threshold) = LLC congestion

Counters
- → Avg. number of writes
- → Avg. occupancy

→ For previous K windows

More congestion → More WCAB bypass

Less congestion → Less WCAB bypass

Running average → Small spikes are not treated as congestion

Liveness:

→ Simple bypass → Reduced hit rate
→ Increased traffic

→ Allocate observer sets in L2 & LLC ⇒ To track reuse behaviour
→ Hashed L2 access PC (Program counter): Instruction address that last accessed the cache line in L2.

↳ 5 buckets (0 - 100 %.) increments of 20.

(0 - 100 %.) → Liveness buckets

Liveness = How much of the evicted sets are recalled again

(Eg.) 20% = 20 sets were recalled out of the 100 evicted sets.

Only done for observer sets

↓

For 21% - 50% ⇒ → Eviction → Decrement by 2
→ Recall → Increment by 10

WCAB: Write Congestion Aware Bypass

1) Detect congestion:

→ Monitor LLC request queue.
→ If the no. of write requests is above a fixed threshold, it indicates severe congestion.
→ WCAB activates severe bypassing

2) Liveness score:

→ Dont bypass block with high liveness score

→ Observer set:

- Track how often these blocks are accessed after being written

→ Liveness score:

- Score is divided into bucket ⇒ 4 buckets

3) Bypass decision:

→ When congested is detected, writes with low liveness score is bypassed

4) This bypass is adapted overtime based on congestion to modulate the agressiveness.

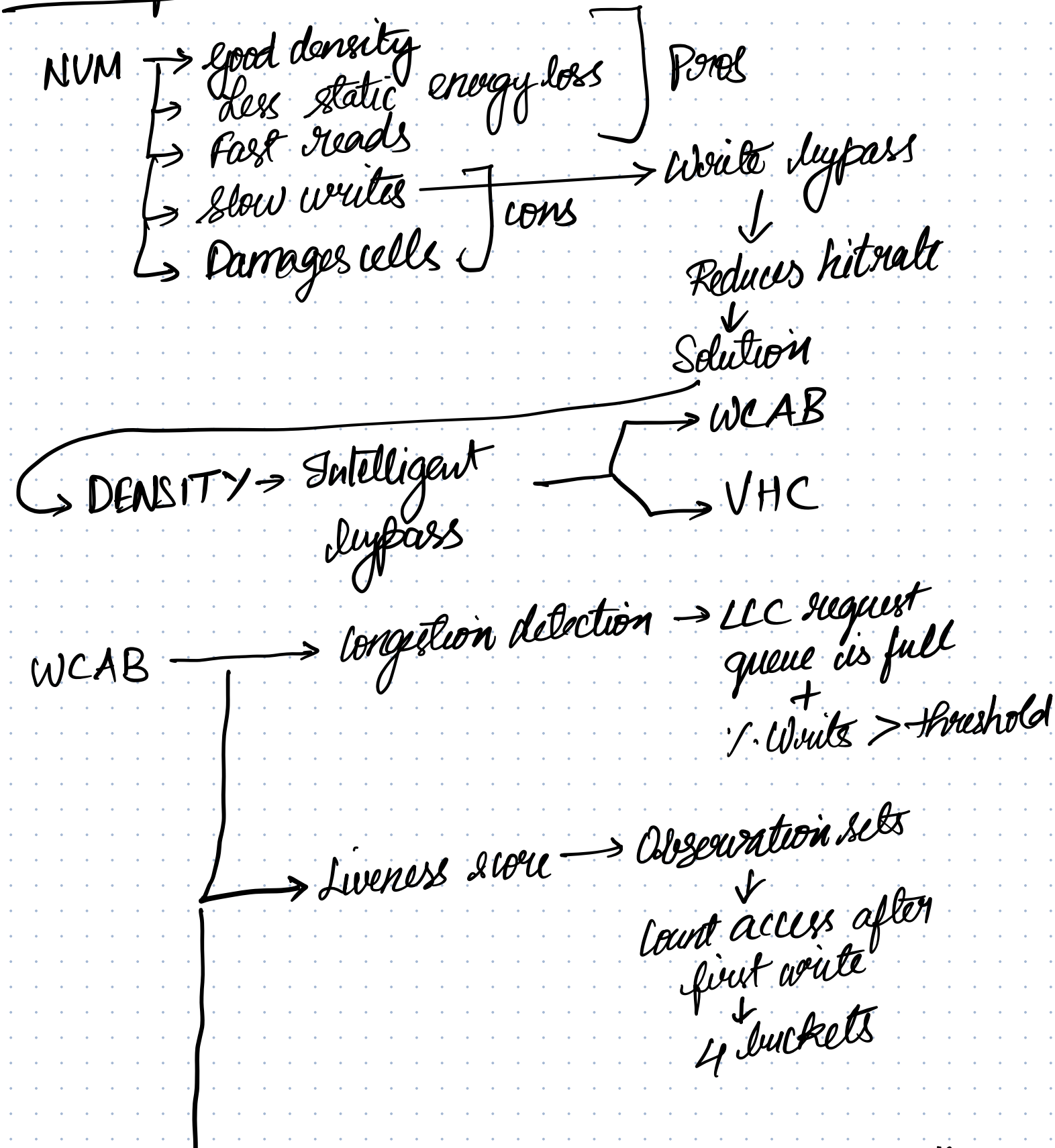
Victim Hybrid Cache (VHC):

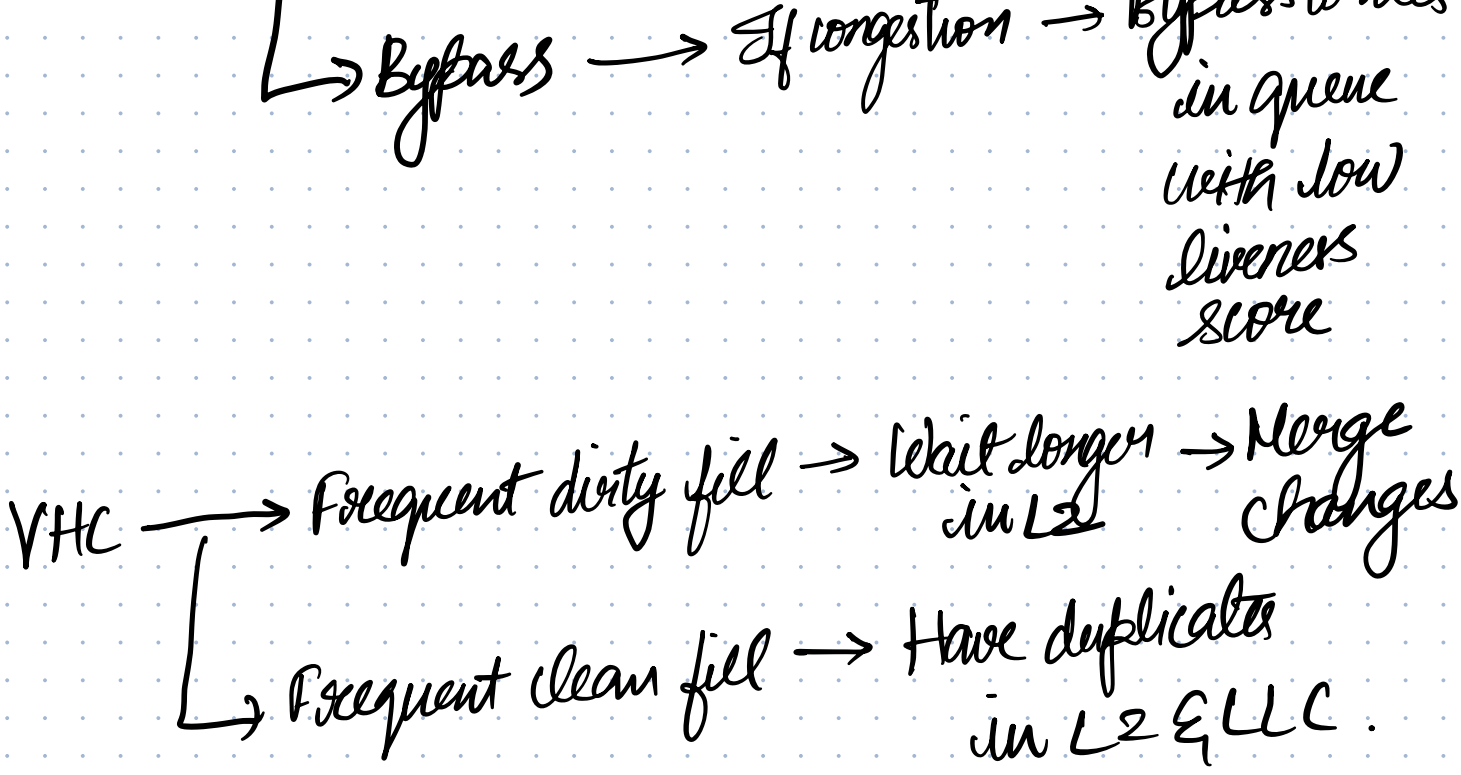
1) Frequently dirty fills: The writeback is not done to LLC everytime. Instead it is stored longer in L2 and these changes are merged & then written back to LLC.

2) Frequently clean fills: A copy of frequently clean

blocks are stored both in L2 & LLC to prevent unnecessary writebacks -

Mindmap :




Bypass
If congestion
Bypass to lines in queue with low liveness score
VHC
Frequent dirty fill
Wait longer in L2
Merge changes
Frequent clean fill
Have duplicates in L2 & LLC.