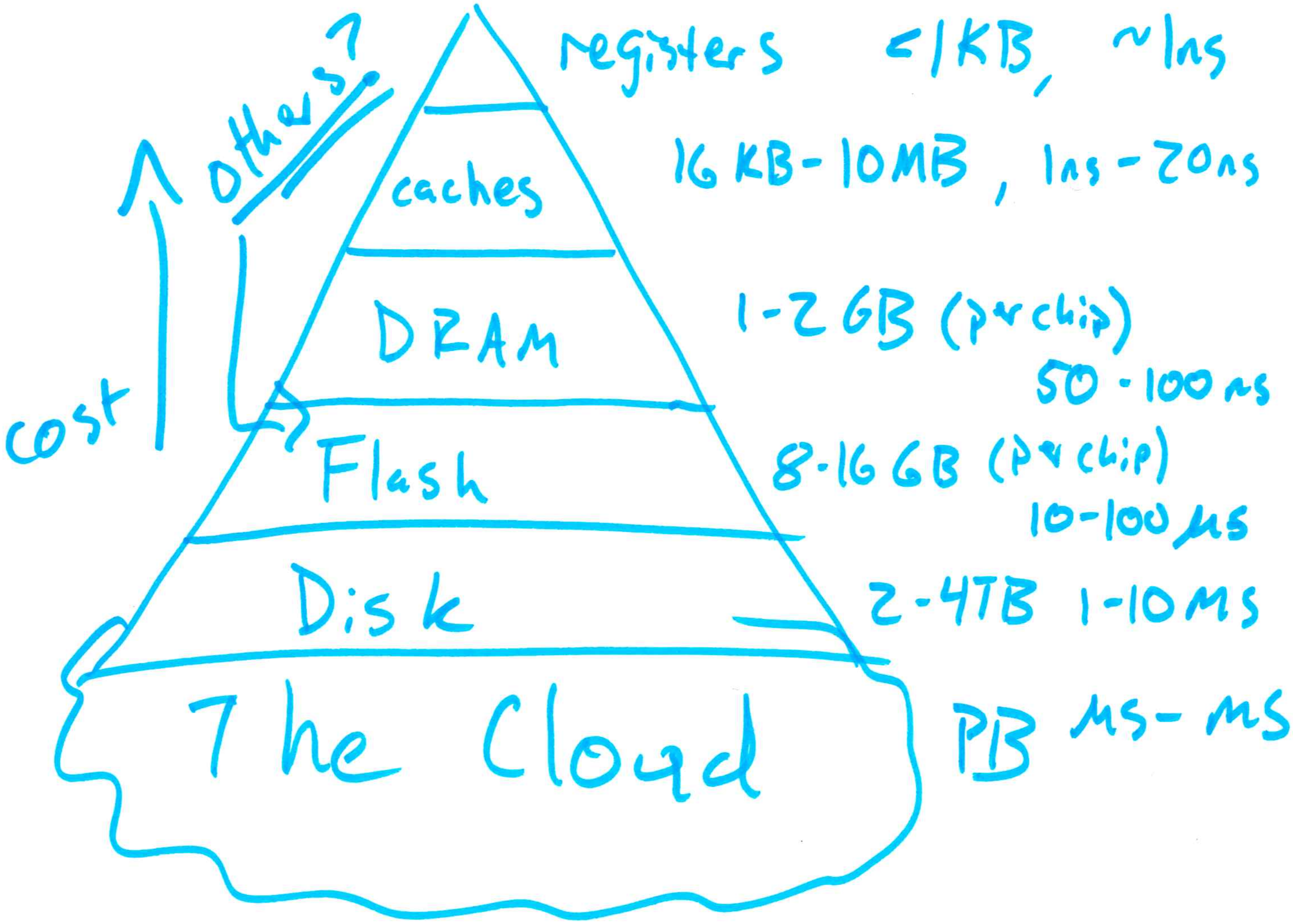


This works because of <u>locality</u>

2 types of locality

<u>Spatial locality</u>
↳ if I'm accessed it's likely my neighbor will be

- → instruction stream
- → stack accesses
- → reading an array

<u>Temporal locality</u>
↳ I'm likely to be reused

- → access data repeatedly
- → loops of instructions
- → stack accesses

```
sum=0
for (int i=0; i<n; i++){
    sum += a[i];
}
return sum;
```

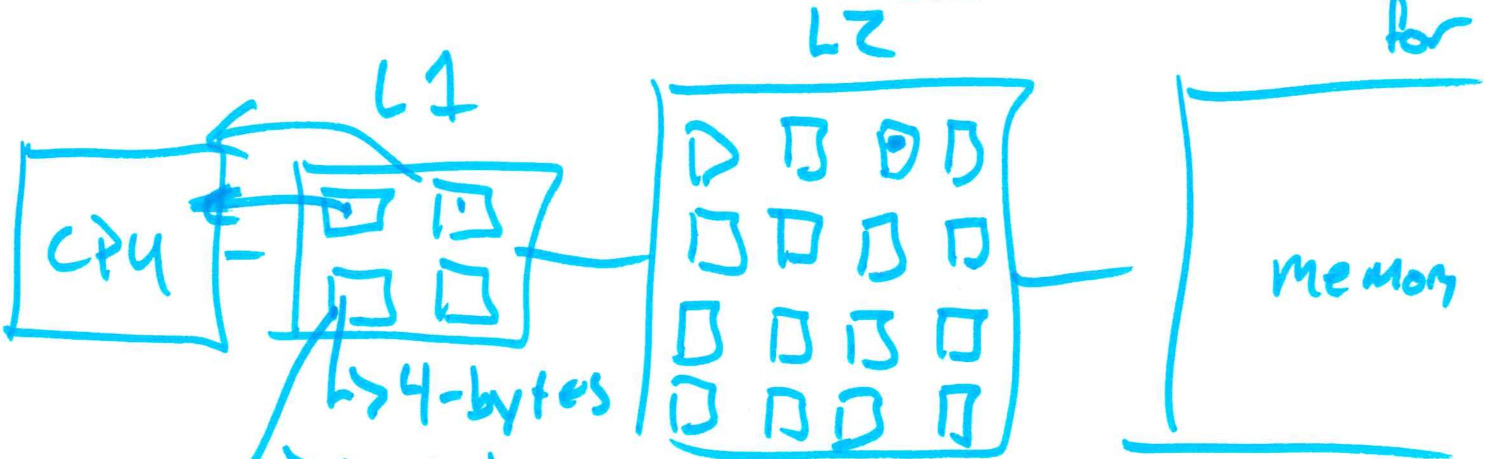
a[i], i++ → spatial

sum → temporal

spatial locality in instruction stream + temporal locality in the loop

Caches take advantage of locality

- Temporal locality → keep recently used things close
- Spatial locality → load more than was asked for



→ much bigger to take advantage of spatial locality
64-bytes

CPU checks cache → if it's there: **HIT**

if not there: **MISS**

3 different kinds of cache misses (3 C's)

- Cold/compulsary misses
  - ↳ ~~brin~~ miss because cache is empty
  - ↳ "warm up" caches
- Conflict misses
  - cache could hold the data, but because of limited "slots" it's a miss
- Capacity miss
  - cache is full
  - Working set is too big for cache
    - ↳ current data program is using

Caches are managed by hardware

↳ no software/OS/etc involved

↳ logically transparent to the programmer

in contrast to registers which are managed by the programmer

Average memory access time

$$(T_{hit}) + (T_{miss}) \cdot \begin{pmatrix} \text{miss} \\ \text{rate} \end{pmatrix}$$ → or miss ratio

Cache hit time 1ns

miss time 100ns

Hit rate → 99% → Miss rate = 1%

↳ $\frac{\#\text{ hits}}{\text{total }\#\text{accesses}}$ $\frac{\#\text{ misses}}{\text{total }\#\text{ of accesses}}$

$$AMAT = 1ns + 100_{ns} \cdot (.01)$$
$$= 2ns$$

$$T_{miss} = AMAT_{L2} = T_{hitL2} + T_{missL2} \cdot \begin{pmatrix} \text{L2 miss} \\ \text{rate} \end{pmatrix}$$

L2 miss rate = 25%

L2 hit time = 10ns

L2 miss time = 100ns

$$AMAT = 1ns + (.01) \cdot [10ns + .25 \cdot 100]$$
$$= 1 + .01 \cdot (35)$$
$$= 1.35ns$$

$$AMAT = T_{hit} + T_{miss} \cdot \begin{pmatrix} miss \\ rate \end{pmatrix}$$

$$T_{miss} = AMAT_{L2} = T_{hitL2} + T_{missL2} \cdot \begin{pmatrix} L2\ miss \\ rate \end{pmatrix}$$

$$AMAT = T_{hitL1} + \begin{pmatrix} L1\ miss \\ rate \end{pmatrix} \left[T_{hitL2} + T_{missL2} \cdot \begin{pmatrix} L2\ miss \\ rate \end{pmatrix} \right]$$