
How to
Approach

* Generally open. ended problems are given.

* So, we will dig into the problem, like

SCOPING
- (i) whether we wanted to high level design of the full web-site
- (ii) Or, how a particular functionality like (liking a tweet) will work.

Requirements

functional

* How we want the application to function? What should happen when someone likes a tweet? How to generate the user's feed?
* Discussed so that we will design up only what's required, neither what I wanted
* Used to check how careful we are? or we just make assumptions without clarifying the statement

non-functional

* Mainly about Scalability
  - ↳ like we have 100 M daily users. So, how we should handle these many requests?
* Throughput
* Storage Capacity of our D.B.
* Performance
  - * Latency
  - * D.B partitions.
    - ↳ Like read < 1 sec
      write < 2 second
* Availability.

* we are not required to over-engineer our solution. Because after a particular threshold (5 9's), the improvement is of 5 mins, but cost is going to be really high.

* Trade off :- To let non-important things go, than loosing the Super-important functionality..

Back of the envelope calculations

* In the interview we do not have time for the calculation of exact no. of

* Let us understand it with an example :-

Suppose we have a user Base of 100 M out of which every user on an average reads about 100 tweets daily and 10% of these actually write tweets.

$\Rightarrow$ Total reads $\Rightarrow$ 100 M x 100 reads = 10 billion.

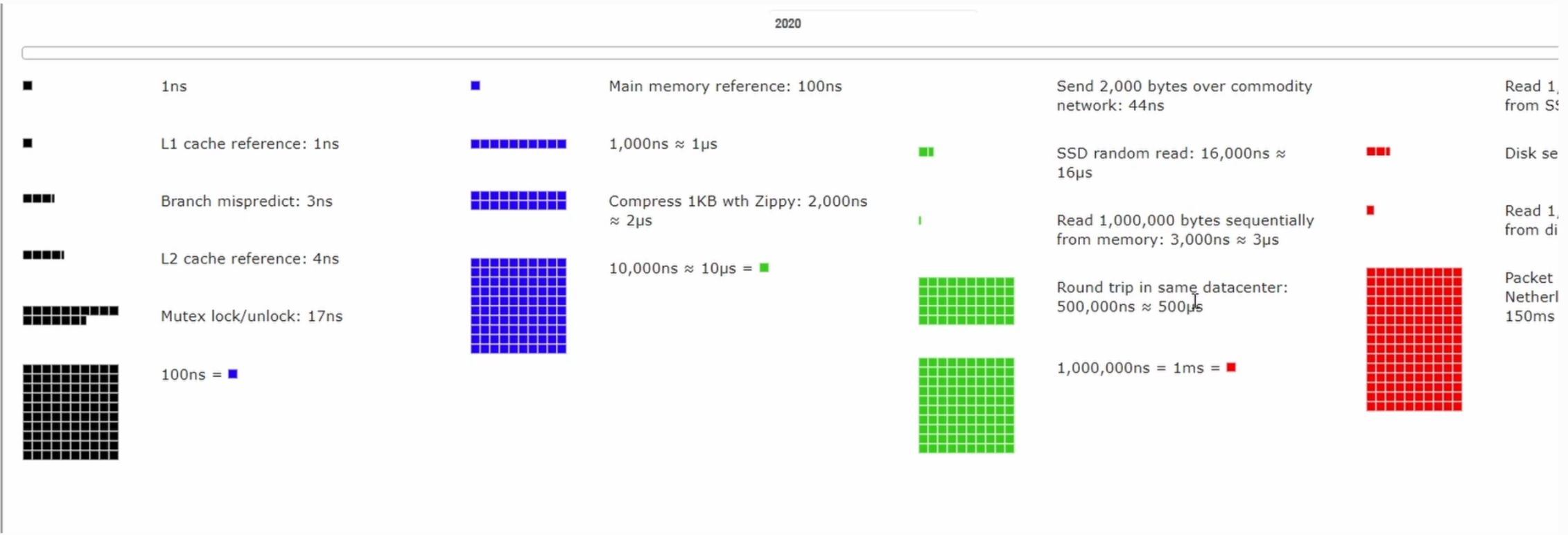
Total writes $\Rightarrow$ 10% of 100 M = 10 million

$$\text{Throughput}_{\text{read}} = \frac{10b}{\text{Total second in a day}} = \frac{10 \times 10^9}{60 \times 60 \times 24}$$

$$\simeq \frac{10 \text{ billion}}{100 \text{ k.}} = 10^5 \text{ reads / second.}$$

$$\text{Throughput}_{\text{write}} = 10^2 \text{ write / seconds.}$$

* for simplicity, I will ignore the requests of authentication & payments.


2020
1ns
L1 cache reference: 1ns
Branch mispredict: 3ns
L2 cache reference: 4ns
Mutex lock/unlock: 17ns
100ns =
Main memory reference: 100ns
1,000ns ≈ 1µs
Compress 1KB wth Zippy: 2,000ns ≈ 2µs
10,000ns ≈ 10µs =
Send 2,000 bytes over commodity network: 44ns
SSD random read: 16,000ns ≈ 16µs
Read 1,000,000 bytes sequentially from memory: 3,000ns ≈ 3µs
Round trip in same datacenter: 500,000ns ≈ 500µs
1,000,000ns = 1ms =
Read 1,
from S
Disk se
Read 1,
from di
Packet
Netherl
150ms