Leetcode

Hanhee Lee

June 9, 2025

Contents

1 How to approach a system design interview question? 2

1.1 Back-of-the-envelope calculations 2

1.1.1 Powers of 2 Table 2

1.1.2 Latency numbers every programmer should know 3

1 How to approach a system design interview question?

Process: The system design interview is an open-ended conversation. You are expected to lead it.

1. **Outline use cases, constraints, and assumptions**
    - Gather requirements and scope the problem. Ask questions to clarify use cases and constraints. Discuss assumptions.
        - Who is going to use it?
        - How are they going to use it?
        - How many users are there?
        - What does the system do?
        - What are the inputs and outputs of the system?
        - How much data do we expect to handle?
        - How many requests per second do we expect?
        - What is the expected read to write ratio?
2. **Create a high level design**
    - Outline a high level design with all important components.
    - Sketch the main components and connections.
    - Justify your ideas.
3. **Design core components**
    - Dive into details for each core component.
    - For example, if you were asked to design a URL shortening service, discuss:
        - Generating and storing a hash of the full URL
            - MD5 and Base62
            - Hash collisions
            - SQL or NoSQL
            - Database schema
        - Translating a hashed URL to the full URL
            - Database lookup
        - API and object-oriented design
4. **Scale the design**
    - Identify and address bottlenecks, given the constraints.
    - For example, do you need the following to address scalability issues?
        - Load balancer
        - Horizontal scaling
        - Caching
        - Database sharding
    - Discuss potential solutions and trade-offs. Everything is a trade-off. Address bottlenecks using principles of scalable system design.
5. **Back-of-the-envelope calculations**
    -

1.1 Back-of-the-envelope calculations

1.1.1 Powers of 2 Table

Summary:

Power	Exact Value	Approx Value	Bytes
7	128		
8	256		
10	1,024	1 thousand	1 KB
16	65,536		64 KB
20	1,048,576	1 million	1 MB
30	1,073,741,824	1 billion	1 GB
32	4,294,967,296		4 GB
40	1,099,511,627,776	1 trillion	1 TB

1.1.2 Latency numbers every programmer should know

Summary:

Latency Comparison Numbers

Operation	Time (ns)	Time (us)	Time (ms)	Notes
L1 cache reference	0.5			
Branch mispredict	5			
L2 cache reference	7			14x L1 cache
Mutex lock/unlock	25			
Main memory reference	100			20x L2 cache, 200x L1 cache
Compress 1 KB with Zippy	10,000	10		
Send 1 KB over 1 Gbps network	10,000	10		
Read 4 KB randomly from SSD*	150,000	150		~1 GB/sec SSD
Read 1 MB sequentially from memory	250,000	250		
Round trip within same datacenter	500,000	500		
Read 1 MB sequentially from SSD*	1,000,000	1,000	1	~1 GB/sec SSD, 4x memory
HDD seek	10,000,000	10,000	10	20x datacenter roundtrip
Read 1 MB sequentially from 1 Gbps	10,000,000	10,000	10	40x memory, 10x SSD
Read 1 MB sequentially from HDD	30,000,000	30,000	30	120x memory, 30x SSD
Send packet CA → NL → CA	150,000,000	150,000	150	

- 1 ns = 10^{-9} seconds
- 1 μs = 10^{-6} seconds = 1,000 ns
- 1 ms = 10^{-3} seconds = 1,000 μs = 1,000,000 ns

- Read sequentially from HDD at 30 MB/s
- Read sequentially from 1 Gbps Ethernet at 100 MB/s
- Read sequentially from SSD at 1 GB/s
- Read sequentially from main memory at 4 GB/s
- 6–7 world-wide round trips per second
- 2,000 round trips per second within a data center