Chapter 15: Back-of-Envelope Estimations

1. Traffic Estimation

- **QPS (Queries Per Second)**: Estimate the number of requests your system will handle per second, often broken down into:
  - **Daily Active Users (DAU)** or **Monthly Active Users (MAU)**.
  - Average number of requests per user per day or hour.
  - Peak vs. off-peak traffic to understand the highest load.
- Example: If you expect 1 million DAU with an average of 10 requests per user per day, your QPS could be roughly (1,000,000×10)/(24×3600)≈115 QPS on average.

  (1,000,000×10)/(24×3600)≈115(1,000,000 \times 10) / (24 \times 3600) ≈ 115

2. Data Storage Requirements

- **Size of Each Record**: Calculate how much storage each data type will consume (e.g., profile data, posts, messages).
- **Total Data Volume**: Estimate total storage based on the number of records and their size.
  - Multiply the size of each record by the estimated number of records per user and total users.
- **Data Growth Rate**: Estimate how much new data will be added daily, monthly, or yearly.
- **Example**: If each user profile takes 2 KB and you expect 10 million users, total storage for user profiles alone would be 10,000,000×2KB=20GB.

  10,000,000×2 KB=20 GB10,000,000 \times 2 \, \text{KB} = 20 \, \text{GB}

3. Network Bandwidth

- **Data Transfer per Request**: Estimate the data sent and received for each request.

- **Total Bandwidth**: Calculate bandwidth requirements by multiplying data per request with QPS.
- Example: If each request transfers 5 KB and QPS is 100, total bandwidth is $5KB \times 100 = 500KB/s$ or about 4Mbps.

  5 KB×100=500 KB/s5 \, \text{KB} \times 100 = 500 \, \text{KB/s}

  4 Mbps4 \, \text{Mbps}

4. Latency Requirements

- **Latency Goals**: Define acceptable latencies for different operations (e.g., reads vs. writes).
- **Cache Requirements**: If low latency is essential, consider the proportion of requests that can be served by a cache and calculate cache hit ratios.
- Example: If the cache hit rate is 80% and latency for a cache hit is 5 ms while a database read takes 50 ms, average latency would be $0.8 \times 5 + 0.2 \times 50 = 14$ ms.

  0.8×5+0.2×50=140.8 \times 5 + 0.2 \times 50 = 14

5. Read/Write Patterns

- **Read/Write Ratio**: Calculate the proportion of reads to writes, which impacts database choice and caching strategies.
- Example: In a social media feed, the read/write ratio might be 10:1, with users reading more frequently than posting.

6. Cache Size Estimation

- **Hot Data**: Determine the percentage of data accessed most frequently and estimate the cache size based on it.
- **Cache Expiry**: Calculate how long data should stay in cache before expiring, balancing freshness with hit rate.
- Example: If 20% of data is “hot” and your total dataset is 100 GB, then a 20 GB cache might suffice.

7. Database Storage and Partitioning

- **Sharding Strategy**: Estimate the number of shards or partitions based on data volume and anticipated growth.

- **Indexing**: Calculate additional storage requirements for indexes based on key fields.
- **Replication**: Factor in storage overhead if replication is required for high availability.

8. Replication and Availability Requirements

- **Number of Replicas**: Estimate storage overhead and network traffic if data is replicated across multiple nodes.
- **Uptime Requirements**: Consider the necessary redundancy and failover capacity to meet SLAs for availability.

9. Estimating Compute Resources

- **CPU Requirements**: Based on request processing time and QPS, estimate the number of servers needed.
- **Memory Requirements**: Calculate memory needs, especially if caching or in-memory storage is used.
- Example: If each request takes 10 ms of CPU time and you have a QPS of 1000, total CPU time needed per second is $10ms \times 1000 = 10seconds$ of CPU per second, meaning at least 10 cores are needed to handle the load.

  10 ms×1000=10 seconds10 \, \text{ms} \times 1000 = 10 \, \text{seconds}

Example

Info and Assumptions

- Assume the application has 50 million signed up users and 10 million DAU.
- Users get 10 GB free space.
- Assume users upload 2 files per day. The average file size is 500 KB.
- 1:1 read to write ratio.

With the provided information, here are some back-of-the-envelope estimations that we can make to help design the system and evaluate its scalability requirements:

1. Storage Requirements

- **Total Storage Allocation**:
  - Each user has 10 GB of free storage space.
  - Total storage needed: $50\,million\ \ users \times 10\,GB = 500\,PB$ (Petabytes).
- **Daily Upload Storage Requirement**:
  - Given that daily active users (DAU) are 10 million, and each uploads 2 files of average 500 KB:
    - **Total daily storage for uploads**: $10\ million\ \ users \times 2\ files \times 500KB = 10TB/day$ (Terabytes per day).
- **Yearly Growth in Storage Due to Uploads**:
  - Assuming uploads occur every day, the yearly storage requirement would be:
    - **Yearly growth**: $10TB/day \times 365 \approx 3.65PB/year.$
  - This yearly increase suggests that the system's storage should be designed to handle an annual growth of around 3.65 PB in addition to the initial allocated space.

2. Traffic Estimation (QPS)

- **Average QPS for Uploads**:
  - **QPS for upload API**: $10\ million\ \ users \times 2\ uploads/day/86,400seconds \approx 240QPS.$
- **Peak QPS**:
  - Assuming peak traffic is double the average, the peak QPS for the upload API would be:
    - **Peak QPS**: $240 \times 2 = 480.$
  - The system should be designed to handle a peak QPS of 480 for file uploads.
- **Download QPS**:
  - With a 1:1 read-to-write ratio, download requests will match upload requests, so **average download QPS** would also be **240** and **peak download QPS** would be **480**.

3. Bandwidth Requirements

- **Upload Bandwidth**:
  - Each file upload is approximately 500 KB.
  - With a peak QPS of 480 for uploads:
    - **Peak upload bandwidth**: $480\ QPS \times 500KB = 240,000KB/s$, which is approximately **234 MB/s**.
- **Download Bandwidth**:
  - Since the read-to-write ratio is 1:1, download bandwidth would mirror upload bandwidth.
    - **Peak download bandwidth**: **234 MB/s**.
- **Total Bandwidth**:
  - The system will need to support a combined peak bandwidth of **468 MB/s** (upload + download) during peak times.

4. Database Storage for Metadata

For every file, metadata like file ID, user ID, filename, file size, and upload timestamp are typically stored in a database.

- **Daily Metadata Storage**:
  - Assuming each metadata entry is about 1 KB, and with 20 million uploads per day:
    - **Daily metadata storage**: $20\ million\ files \times 1KB = 20GB$.
- **Yearly Metadata Growth**:
  - With daily uploads, the annual metadata storage requirement would be:
    - **Yearly metadata storage**: $20GB/day \times 365 \approx 7.3TB/year$.

5. Cache Size Estimation

To reduce load on storage and improve latency, frequently accessed files and metadata should be cached.

- **Assumed Hot Data**:
  - Let’s assume 20% of daily active users' data is accessed frequently and could benefit from caching.

- With 10 million DAU, this is **2 million users' data**.
- Assuming each user accesses 2 files, we would need to cache approximately **4 million files**.

- **Cache Storage Requirement**:
  - If each cached file is 500 KB, the total cache size needed for hot files is:
    - **Cache size**: $4\ million\ files \times 500KB = 2\ TB.$

Summary of Estimations

Metric	Value
Total Storage	500 PB
Daily Upload Storage	10 TB
Yearly Upload Growth	3.65 PB
Average QPS (Uploads)	240
Peak QPS (Uploads)	480
Average QPS (Downloads)	240
Peak QPS (Downloads)	480
Peak Upload Bandwidth	234 MB/s
Peak Download Bandwidth	234 MB/s
Total Peak Bandwidth	468 MB/s
Daily Metadata Storage	20 GB
Yearly Metadata Growth	7.3 TB
Cache Size	2 TB

These calculations provide a foundation for understanding the system's scalability requirements and help in designing appropriate storage, caching, bandwidth, and database solutions to support this scale.