System Design Back-of-the-Envelope Cheat Sheet

Numbers + formulas + tiny examples for fast interview estimation

Ready■Reckoner Numbers

Topic	Rule-of-Thumb Number	Use / Why
Latency (CPU→RAM)	100 ns – 1 µs	Memory access; tiny compared to network
LAN RTT	~0.5–1 ms	Same■DC service calls
Cross■Region RTT	60–120 ms	User noticeable; avoid chatty protocols
User-noticeable lag	~100 ms	Aim below this for perceived instant
Disk seek (HDD)	~10 ms	Random IO on spinning disks
SSD read	~100 µs–1 ms	Much faster than HDD; still slower than RAM
Throughput per app server	~1,000 QPS	Safe ballpark; varies by work
Redis/Memcached node	100k–1M QPS	Serve hot data from memory
Kafka/SQS consumer	5k–50k msg/s	Background work; smooth spikes
Postgres single node	Few k QPS	Heavily query■dependent
Cache hit rate	70–95%	Design for 80%+ for big wins
Read:Write mix	90:10 (reads heavy)	Most consumer apps
Peak ÷ Avg	≈ 5×	Size for peak traffic
CDN offload	60–95%	Static media & public GETs
1 Gbps link	≈ 125 MB/s	Bandwidth conversion
S3 GET p50	10–50 ms	Remote object fetch (order■of■mag)
Image size (web)	100–500 KB	Thumbnail/preview payloads
JSON API resp	1–50 KB	Typical REST payloads

Quick Conversions

- 1 Gbps = 125 MB/s
- 1 TB @ 100 MB/s ≈ 2.8–3 hours
- KB≈10^3, MB≈10■, GB≈10■, TB≈10^{12} (use decimal for estimates)

Formulas You’ll Use in 90% of Interviews

- QPS (avg) ≈ DAU × (reqs per user per day) ÷ 86,400
- Peak QPS ≈ Avg QPS × 5
- Bandwidth ≈ QPS × payload size
- Daily Storage ≈ events/day × event size
- Cache Size (hot set) ≈ 20% of total data (Pareto)

Tiny Worked Examples

Login API

- Assume 5M DAU, 10 requests/day/user → 50M req/day
- Avg QPS ≈ 50,000,000 ÷ 86,400 ≈ 579 QPS; Peak ≈ 2,900 QPS
- Servers: ~3 app servers (1k QPS each) + redundancy (N+1 → 4–5)
- Bandwidth (2 KB JSON): 2,900 × 2 KB ≈ 5.8 MB/s (easy)

News Feed Read

- Assume cache hit 90%, payload 200 KB, peak 10k QPS
- From cache: 9k QPS × 200 KB ≈ 1.8 GB/s → needs CDN/edge
- DB only sees 10% misses: 1k QPS; add read replicas or CQRS

Tweet Storage

- 100M tweets/day × 300 B ≈ 30 GB/day → ~11 TB/year
- Hotset (20%) ≈ 6 TB; keep in cache or fast tier

Image CDN

- 20M image views/day, avg 300 KB → 6,000,000,000 KB/day ≈ 6 TB/day
- With 80% CDN hit, origin sees 1.2 TB/day; size origin egress for peak

Golden Rules (Memorize)

- Design for peak, not average.
- Cache first; measure hit rate; aim 80%+.
- Push static media to CDN; keep APIs lean (≤10 KB when possible).
- Keep services chatty only within a LAN; cross■region calls are expensive.
- Budget N+1 capacity (one server can fail and you’re fine).