# Latency Numbers Every Programmer Should Know

| Operation | Time in ns | Time in ms (1ms = 1,000,000 ns) |
|---|---|---|
| L1 cache reference | 0.5 | |
| Branch misprediction | 5 | |
| L2 cache reference | 7 | |
| Mutex lock/unlock | 100 | |
| Main memory reference | 100 | |
| Compress 1KB with Zippy | 10,000 | |
| Send 2KB over 1 Gbps network | 20,000 | 0.02 |
| Read 1 MB sequentially from memory | 250,000 | 0.25 |
| Round trip within same datacenter | 500,000 | 0.5 |
| Disk seek | 10,000,000 | 10 |
| Read 1 MB sequentially from 1Gbps network | 10,000,000 | 10 |
| Read 1 MB sequentially from disk | 30,000,000 | 30 |
| Send packet CA->Netherlands->CA | 150,000,000 | 150 |

Therefore, it is possible to read:

- sequentially from disk at a rate of ~30MB per second
- sequentially from 1Gbps Ethernet at a rate of ~100MB per second
- sequentially from main memory at a rate of at a rate of ~4GB per second

No more than 6-7 round trips between Europe and the US per second are possible, but approximately 2000 per second can be achieved within a datacenter.

***A note on these latency numbers***

*These numbers are from a 2009 presentation entitled "Software Engineering Advice from Building Large-Scale Distributed Systems" given by Google engineer Jeff Dean. The numbers are now somewhat outdated and hardware is faster, but they are reasonable numbers to use for quick calculations and they allow a sanity check on a design without needing to worry about the statistical distribution of latencies, queuing issues and so on.*

# Back of the Envelope Calculations

*These are some general (and basic) examples of using latency numbers to do rough estimates - these are not tailored to the specific exercise in the SRE Classroom event.*

A system stores images 256KB in size, and thumbnails are 24KB in size. It serves search result pages with 30 thumbnails per page.
The servers used all have 24GB RAM, 8 cores, 2x2TB hard drives, 1Gbps ethernet

**How long does it take to generate image result pages?**
The naive design is to do all the work on one machine: this is dominated by disk seek time.
30 images / 2 disks per machine = 15 - reads per disk to generate a page.
(256KB / 1MB) * 30 ms + 10ms seek = 17.5 ms - time to read one image from disk.
15 reads * 17.5 ms = 362 ms - approximate time to generate one page.

In a parallel design, frontends issue reads in parallel across a number of image servers.
Time to read 30 * 24KB over network: (30 * 24KB * 10 ms) / 1MB = ~7 ms
17.5 ms + 0.5 ms (intra-DC roundtrip) + 7 ms = ~25 ms to generate one page.

**How many image servers are needed to serve 100,000 image results pages per second?**
100,000 * 30 = 3,000,000 thumbnail requests per second (i.e. 3 million).
(1000 ms / 17.5 ms) * 2 disks = ~120 thumbnails generated per second per server.
3,000,000 / 120 = 25,000: the number of machines needed.

**Dealing with failure: tolerate the loss of up to 20% of image servers**
Add 20% extra capacity - run image servers on 30,000 machines (25,000 * 1.2).

**How might caching thumbnails affect number of image servers required?**
For each 4TB of disk there is ~20GB of RAM to cache thumbnails (OS uses ~4GB at runtime).
20GB * (256KB / 24KB) / 4TB = ~0.2TB / 4TB = 0.05
We can cache thumbnails for 5% of the images that could be stored on disk.
However it often happens in practice, some images are far, far more popular than others. What is the impact if the cache hit rate is 50%?

Serving from RAM is very fast and we have plenty of CPU with 8 cores, so it happens entirely in parallel with serving from disk (without slowing down the latter).
The serving capacity per machine doubles (because we now serve 50% from RAM cache), so the number of image servers needed halves.
Is gigabit ethernet enough? 120 * 2 * 24KB = 5760KB per second, under 6MB per second - it is.