## Design Interviews at Facebook

One of the main goals of this interview is to design a Facebook-scale system

## What we ask

Some example questions are:

- Design a key-value store
- Design Google search
- Architect a world-wide video distribution system
- Build Facebook chat

We may focus on some machine learning, networking system, iOS or JavaScript if you have that domain expertise on your resume. We don't expect you to know crazy algorithms that you likely wouldn’t know off the top of your head (like quad trees or Paxos). The emphasis is on how you perceive the system and problem space.

## What we look for

We’re looking for:

- Can you arrive at an answer despite unusual constraints?
- Can you visualize the entire problem and solution space and identify its various components and the relationships between them?
- Can you point to the main use-cases that the system will address?
- Can you identify and address scale bottlenecks? How will your design make trade-offs like consistency, availability, partitioning, or performance?
- Can you give ballpark numbers on throughput/capacity for RAM, hard drive, network, etc. using modern hardware?
- How much have you thought about Facebook and some of the unique problems we face?

A good design shows that you:

- Clearly understand the problem and can describe it with simplicity.
- Are able to come up with a holistic system design, and can also break it down into smaller components and sub-systems.

- Identify the bottlenecks as the system scales and understand the limitations in your design.
- Understand how to adapt a solution when requirements change.
- Draw diagrams that clearly describe the relationship between the different components in the system
- Calculate (back-of-the-envelope) the physical resources necessary to make a system work

## How to study

Our design interviews are tricky to study for.

To practice, take any well-known app and imagine you work for a competitor. Your job is to figure out 1) the general purpose and goals of their system. 2) what key resources and costs are required (hardware? people? bandwidth? storage?) and 3) the fundamental bottlenecks and growth hurdles of their system. Answering those questions will necessarily force you to think about how a system is actually implemented. Answering only cost and bottlenecks forces you to focus on important areas and not nerdy details of the design. Focus more on the system rather than the nitty-gritty details of the tech.

For example, YouTube spends a *ton* of money on bandwidth, and secondarily on storage and compute. On the other hand, their long-tail traffic pattern means that their fundamental bottleneck is random disk seeks. Netflix also is a bandwidth hog, but most of their traffic is at night (when it's cheap) and their library of videos is much much smaller, so disk-seeks are probably not an issue at all.

Work out the above problems on a paper and just think about the ways to break them down. It also helps to read up on common large-scale systems, like watch the public videos about memcached and learn how search engines work. But during the interview, don’t parrot back what you read; make sure your solution actually answers the question being asked.