

THE INTERVIEW HACKER

Small hacks for big interviews

Those pesky design interviews

Posted on January 23, 2016February 18, 2016 by interviewhacker

For many engineers, the design interview is their Kryptonite.

These interviews are important, since you can demonstrate to prospective employers your ability to architect/design large systems and this might be the difference between a Senior and a SDE/SWE2 role.

Right at the onset of this article I will say that when it comes to design, there is no real substitute for mock interviews. Get a friend or colleague to help you out. If that's not possible, the best thing to do is to pick a question, time yourself and tackle a design questions on your own. Once the allocated time runs out you can step back and analyze the solution you came up with. Often you will be able to pin-point presumptions you made that weren't necessary valid or obvious flaws in your design.

Often there is no "right" and "wrong" answer and the interviewer really wants to see how you can work through ambiguities.

For these type of questions, make sure that you do not make any presumptions, seek clarifications, and gather specifications before jumping to your design.

For class design questions, use interfaces, and possibly design patterns. For system design questions, these are the main pillars you should be mindful of:

- Is it possible to make your design asynchronous and/or parallelize it?
- Can your system handle millions of requests and are there any bottlenecks?
- Can you cache?

If you are anything like me, then these questions much seem pretty scary to you, and if you aren't really in the habit of solving system design questions they actually are. However, there are plenty of useful resources out there and with practice you can definitely get better at this.

These are some awesome resources that you should definitely check out:

1. I would definitely highly recommend that you watch the following video when preparing for system design questions (it's two hours long but will give you a really solid understanding of how large scale scalable systems are designed): Scalability in Web Development (https://www.youtube.com/watch?v=W9F_D3oY4)
2. This link contains a pretty awesome example of how you can work through a design problem: Design a URL shortening service (<http://www.hiredintech.com/system-design/the-system-design-process/>)
3. I also found this four part series on scalability by Sebastian Kreutzberger to be quite pertinent:

- [Clones \(http://www.lecloud.net/post/7295452622/scalability-for-dummies-part-1-clones\)](http://www.lecloud.net/post/7295452622/scalability-for-dummies-part-1-clones)
- [Databases \(http://www.lecloud.net/post/7994751381/scalability-for-dummies-part-2-database\)](http://www.lecloud.net/post/7994751381/scalability-for-dummies-part-2-database)
- [Caching \(http://www.lecloud.net/post/9246290032/scalability-for-dummies-part-3-cache\)](http://www.lecloud.net/post/9246290032/scalability-for-dummies-part-3-cache)
- [Asynchronization \(http://www.lecloud.net/post/9699762917/scalability-for-dummies-part-4-asynchronism\)](http://www.lecloud.net/post/9699762917/scalability-for-dummies-part-4-asynchronism)

4. Finally, there are also a few example of large scalable real world systems on the [high scalability blog \(http://highscalability.com/\)](http://highscalability.com/).

While you are at it, you should also get familiar with Database Sharding, Database Normalization, Indexing, Database Denormalization (including 1 NF, 2 NF, 3 NF) and the difference between relational databases and no-sql.

Click [here for my post on a few good programming interviews questions \(https://theinterviewhacker.com/2016/01/13/a-few-good-programming-questions/#more-210\)](https://theinterviewhacker.com/2016/01/13/a-few-good-programming-questions/#more-210) , [here for my post on some stuff you positively definitely absolutely need to know \(https://theinterviewhacker.com/2016/01/10/stuff-you-positively-definitely-need-to-know/#more-63\)](https://theinterviewhacker.com/2016/01/10/stuff-you-positively-definitely-need-to-know/#more-63), [here for my post on tacking those pesky design questions \(https://theinterviewhacker.com/2016/01/23/those-pesky-design-interviews/\)](https://theinterviewhacker.com/2016/01/23/those-pesky-design-interviews/) and [here for my post on 10 phone screen questions. \(https://theinterviewhacker.com/2016/02/11/10-phone-screen-questions/\)](https://theinterviewhacker.com/2016/02/11/10-phone-screen-questions/)

If you have feedback/questions/concerns please leave a reply below.

Please consider following me on twitter – <https://twitter.com/theinterviewhak> (<https://twitter.com/theinterviewhak>)

This blog is a side project that I have undertaken to help others prepare for their interviews. I am not charging for any of my content or services. It would mean a lot if you could share the blog entries that you like on fb, twitter, google+ etc (sharing links provided on the blog page at the bottom of the content) and help spread the word as the traffic keeps me motivated to continuously serve up good content.

To receive an email every time I publish new content please “Follow” this blog by hitting the “Follow” button on the bottom right corner.

If you'd like to make a request for an invitation to be coached by me then please click [here \(http://eepurl.com/bMklgX\)](http://eepurl.com/bMklgX) (currently only corresponding via email).

5 thoughts on “Those pesky design interviews”

1. **Pulak** [February 8, 2016](#) [Reply](#)

Your website is just the right stuff that guides good engineers to the right kind of jobs. This is great work. Thank you.

[_interviewhacker_](#) [February 8, 2016](#) [Reply](#)