<https://github.com/lee2018jian/airbnb/blob/master/SystemDesign>

(1). RSS

<<(1a). EMC 发表于 昨天 07:38 >>

Design: RSS。 先算load， 然后design db schema 和 build index。 写sql query to fetch posts order by time and need to spuuort pagionation as well. 最后scale 和 讲讲 如何 sharding db。

"sql query to fetch posts order by time and need to support pagination as well"  
  
pagination 楼主是 ...

用 sql 的offset 和limit就好了啊。 然后再restful API里传参去set offset 和 limit的值。

<<(1b). grace19 发表于 >>3. RSS reader design。先说 DB，再说 server，再说全局 scale。DB 就是需要哪些 table，有哪些 column，怎么 index，DB 大小怎么估计。QPS 是多少。

<<(1c). wegnahz 发表于 2018-11-30 >> design一轮问的是一个RSA订阅的问题，主要考数据存储方面的设计，比如用几个table，每个table都有哪些column，每个操作怎么实现，主要就是弄一个Subscription表，里面有userId和siteId这样对应起来就好。后来又有一种操作，是一键把当前所有的推送全都标为已读，问怎么实现。这里注意不能直接给user标一个timestamp，而是一定要对每个(userId, siteId)标timestamp，因为将来可以订阅新的site。后来还扯了个CDN，其他的细节记不清了。

<<(1d). pf22099 发表于 2018-9-18 >>

- RSS reader，之前的qps和storage 计算很重要，决定你设计一个什么样的系统。

**补充内容 (2018-9-22 15:17):**  
补充RSS：刚开始有1000个用户，1000个provider，设计这个Rss reader，让每个用户每次刷新开始看到最近刚更新的的news，后来用户增长到10K, 100K, 系统怎么scale。其实就是设计Twitter/facebook news feed。  
  
**补充内容 (2018-9-22 15:20):**. check   
RSS的重点在于你需要计算QPS和需要多大的storage，这样你就知道你需要用SQL还是NOSQL，需不需要sharding，需不需要memcache。面试官当时问我用户增长到10M的时候，我设计的那个table会最先出问题，问得很仔细。。

<<(1e). flyman3046 发表于 2018-10-1>>

news feed design，主要就是设计database table，怎么index之类的

<<(f). r1unningMajia 发表于 2018-10-30>>

求问楼主， 能否讲下 RSS 阅读器的 问题呢？ 是设计Schema 么 难点有哪些哇？ 拜托拜托，多谢

. check 1point3acres for more.  
rss就是设计schema为主，怎么建index，怎么优化，scalability的也问了一些

<<(1g). 131415926 发表于 2018-5-12>>

3. System design， news feed 靠，纯是mysql各个table存什么数据，怎么index，B tree trade off blabla，这轮三哥笑嘻嘻还挺help full的

<<(1h). >>

<https://github.com/lee2018jian/airbnb/blob/master/SystemDesign>Top of Form

Bottom of Form

Top of Form

Bottom of Form

|  |
| --- |
|  |
| design. RSS reader 设计database schema, 具体到怎么index，follow up: 需要纪录post read/unread; 需要记录feed read/unread; 怎么scale..本文原创自1point3acres论坛 | |
|  | |

|  |
| --- |
| 我的答案仅供参考： |
|  |

|  |
| --- |
| user table: {user\_id, user\_name, user\_info etc}. visit 1point3acres for more. |
|  |

|  |
| --- |
| posts table: {post\_id, feed\_id, publish\_time, post\_title, post\_content} |
|  |

|  |
| --- |
| feeds table: {feed\_id, feed\_url, create\_time} |
|  |

|  |
| --- |
| subscription table: {subcription\_id, user\_id, feed\_id, last\_read\_time} last\_read\_time 用于纪录feed read/unread,我一开始加了一个新的table，经面试官提示合并到这个table里面。 |
|  |

|  |
| --- |
| read status table: {post\_id, user\_id} 只有看过的post会放到这个table里面 |
|  |

|  |
| --- |
|  |
|  |

(1i),

|  |
| --- |
|  |
|  |

|  |
| --- |
| 设计：订阅系统 需要考虑如果停止订阅要从当前Feed中删除对应条目 先设计存储schema，不考虑具体数据库类型假设单机，然后问如何加速查询如何对每个table进行index，需不需要建多个，是否需要复合。 然后就是典型的NewsFeed套路了。特别问到了如果某个发布者有海量订阅者怎么办。 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 这轮主要关注数据库的优化和系统的规模拓展性 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Google reader design. 问了如何提高database读取的效率，详细问了denormalization, 以及瓶颈在哪，怎么解决。 |
|  |

|  |
| --- |
| 这个人明显对index 很感兴趣。顺着他意思说就好，有多少说多少。他很感兴趣。还引进他所遇到的问题，甚至都没有说push 或者pull model。 |
|  |

Design news feed system like Google Reader，貌似是面经，就是用户订阅了好几个news provider比如人民日报、新浪财经，然后用户登录后需要把他订阅的provider的最新news按照时间逆序显示出来。这一轮问的比较细，从最开始的data table schema design，到怎么实现top k算法把news merge；之后问了一个小的feature就是mark news as read，需不需要再加一个table；最后问了一下master slave，类似于怎么sharding

(1j).

3. Google reader design. 问了如何提高database读取的效率，详细问了denormalization, 以及瓶颈在哪，怎么解决。

感觉他们家design问的好细，lz也很牛！lz方便说一下瓶颈和怎么解决是怎么回答吗？  
不牛不牛。瓶颈一般其实就是design里面的最后一步，即evolution. 比如说数据增多时，前面的web server就变成瓶颈了，需要load balancer来分流到多个web server上。另外后面的database也变成瓶颈，因为一台机器存不了这么多数据，就需要sharding来分到多个机器上。

<<(2). key value Store>>

<<(2a). [windmaple](https://www.1point3acres.com/bbs/home.php?mod=space&uid=174084) 发表于 2018-11-3 >>

第三轮， design，华人大哥，问了怎么设计一个 distributed object storage system。主要focus 在后端，写，读，delete，是怎么弄的。Durability，availability是怎么保障的。host down了 怎么解决。一路谈，这个轮比较顺利，大哥频频点头，比较满意。

<< (2b). AllRightThen 发表于 2018-11-15 >>

architecture：设计一个系统返回用户最近访问的5个Home的页面。我回答的是后台有一个service负责读写，前端调用service获取最近访问的列表和写入新访问的home页面，service的后端是key value store。细节可以按每个人喜好来。

<<(2c). >>

<https://github.com/lee2018jian/airbnb/blob/master/SystemDesign>

|  |
| --- |
| Kv store |
|  |

|  |
| --- |
| 问了好多好多最后都超时了……写一下还记得的吧，也不知道自己答的是不是靠谱 也是先说不要想多机器的情况先设计单机版本，内存快速存取且排序（说的跳表或treeset好像不太靠谱），内存满了dump到硬盘，dump的时候创建新表不block新进来的请求，内存存储对文件的index快速定位需要搜索的文件（忘了说boom filter了……），硬盘二分搜索，定期合并避免文件太多，更改删除元素直接append新元素，查询时先内存文件再倒时序查，多次更改删除后浪费空间-同一个key出现的文件数达到某一阈值后在下一次的非高峰时段进行文件合并每个key保留最近值。拓展到多机器，数据迁移一致性哈希，一致性哈希的查询表可以缓存在client端减轻master压力，上传文件时需要返回client完成信息-选择captain上传一由captain复制备份有一个备份传输完成则返回成功信息避免client等太久，保存不同replica的访问顺序避免复制完全结束前收到请求。 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| design就是有点天马行空的给定一个主题聊。好像那个小哥特别特别看database joining/indexing performance，反复针对几个专有名词让我define，比如说什么叫indexing, database reverse indexing, 怎么样的sql是有好的performance的sql language。问的都是我不太会的，最后感觉都快哭了他也没有要停下来让我说其他的意思。。。最后还是我把所有想说的说了，然后接了一句that's all I know.... 这个小哥才开始往其他方向问问题。整个过程非常不互动，就是他一个问题接一个问题，而且我回答了以后他也没有给我反馈，也没告诉我这么说对还是不对。。好几次说完我就停在那里等他下一个要问的问题。 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 考察db design多一些。由于没学过数据库的课，只在工作中用到才去查；因此这一轮回答的不好，很多点没答上来，看来后续还得加强这一点才是，比如MySQL的深入了解，NoSQL的理解等。         现在反思，也怪自己对system desgin的准备时间不够。只按照地里的system design如果明天面试该怎么准备的一个热帖 来准备 |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| 设计一个KV store。我答得中规中矩，中间面试官问如果node挂了怎么办，于是跟他扯到了consensus algorithm，这是我最失误的地方。给他讲了十几分钟joint consensus 和 2 phase commit，他一直觉得我说的有问题。他确实没懂，最后从hr的feedback来看，他觉得我有knowledge gap，不懂如何处理这种情况，我也是无语。早知道直接跟他说用zookeeper就好了。 |
|  |

|  |
| --- |
|  |
|  |

设计key-val database，你可以顺着cassandra也可以顺着hbase答

<<2d. >> design就是有点天马行空的给定一个主题聊。好像那个小哥特别特别看database joining/indexing performance，反复针对几个专有名词让我define，比如说什么叫indexing, database reverse indexing, 怎么样的sql是有好的performance的sql language。问的都是我不太会的，最后感觉都快哭了他也没有要停下来让我说其他的意思。。。最后还是我把所有想说的说了，然后接了一句that's all I know.... 这个小哥才开始往其他方向问问题。

<<(3). 翻译系统>>

<<(3a). windmaple 发表于 2018-11-3>>

第四轮，design，烙印，地里有的设计题，设计online 翻译系统。我就大概按 Google的 翻译系统的 设计讲。几个service之间如何交互，数据如何存，schema 如何，API 如何，问得比较细。

<< (3b). billywxz 发表于 2018-11-14 >>

第五轮design，翻译系统，亚裔小哥全程无笑容，不过lz第一轮都经历过了这还算什么！感觉这一轮虽然看了地里的面经还是面的一般，主要是这题和一般的设计题不太一样有很多前端知识，which lz并没有经验。。

<<(3c). >>

3.系统设计翻译系统，我看了所有气床的面经包括他公司blog上关于翻译系统的设计，感觉都没讲很清楚，所以虽然准备了但是感觉还是有点迷糊，当然我也没在气床工作过不具体了解他们的翻译系统，只是刚好我曾经做过app的localization（就是翻译各种语言给app store这样不同国家的人下载的版本就是那个语言的版本），所以面试的时候我是结合自己工作经验做的介绍，以及一些如何做网站翻译的扩展设计（气床的产品主要还是网站，当然现在也有手机端app，手机端的方法跟我做过的应该是一样的），以下只是我个人的理解，大部分也得到面试官的认可所以应该比较接近他们的系统。面试之后还跟他聊了一些他们翻译系统平时的用法和我给的方法的差别（下面会说）

翻译系统大体分为三个部分: 1. UI (front end)， 2. translation service (middle layer), 3. translator portal，这三部分在一起最后会做出完整的version (web/app)发给client，要么是给app store的app package，要么是转换成网页发给用户（或者CDN for scalability reason)  
  
1) UI部分: front-end developer和designer通常不用去理会如何翻译，他们主要做的是修改ui，这个过程中可能会增加，修改，删除一些文字（比如一个按键的label，比如增加新的网页和目录的标题etc)，他们的任务是确定网页和app这些UI用主语言（比如英语）是没问题的，通常这些修改之后，会提交给translation service...提交的东西你可以理解成是一整套resource文件，里面包含所有的ui相关的文字资料，标注了这是哪个Version，有哪些ui (key-value, where key represent ui variable name like "agreeButtonText", value represents actual text in English "Agree"），通常每个修改还会带一条介绍帮助翻译更好的理解语境，比如叫description里面会写这句话是当用户同意某个条款的时候点击同意的，这样翻译的人就会翻译成（同意，认同之类的），有些语言会有多义词所以这条解释很重要  
2) translation service: 当文字修改发给这个service，他负责三个部分，一个从resource文件中找出新版本中被改变的文字信息，比如有几行条款文字改了，或者一些button的文字改了，第二部分是把这些信息汇总（tokenlize)然后发给 Translator portal，可以理解成一个接受各种key-value pair的工具或者翻译平台，实际翻译发生在那里， 第三部分是当信息被翻译之后，汇总并声称新的resource文件给每个需要的语言，比如一开始只有en-us.resource，翻译之后这个Service会生成zh-cn.resource (中文），es-es.resource （西班牙语）。汇总后会连同这些语言一起发布这个版本的网页或者app (之后就是发布端了这里就不多讲）  
3) translator portal: 每个公司做多语言产品都必然会雇佣一些翻译来翻译它的产品，所以会有一个翻译平台，翻译的人不需要真实了解或者使用这个产品，他们需要的只是拿到一些词汇和语境介绍，然后给出相关语言的翻译，当translation service给了一系列需要翻译的词汇之后，这个平台会收录整理排序这些词汇（有些词汇可能是一样的，有些词汇可能更重要更急需比如一些legal文件更新），然后翻译的人从这个平台拿词，根据介绍翻译好然后发回去，通常可能会有一些PM或者designer会尝试把翻译好的版本自己测试一遍确定读起来通顺没问题，这要看公司的要求和产品重要性了。

上面这三部分基本构成了一套翻译系统，其实没太多系统设计的概念，关于手机端，最后会生成一个主的package 已经所有语言的Resource file发给store然后store负责转换语言生成新的package给不同语言的用户。网络段会有区别，可以理解成这个网页会根据不同的域名比如airbnb.jp跟Server索取日语版的网页，这时候server会有自己的一套生成系统自动提取翻译好的日语词汇安在对应的ui上 (eg.一个button叫agreButton, text = "$agreeButtonText"，他会自己找jp resource里agreeButtonText对应的值，日语的“同意”），生成网页后发给Client；Scale的考虑可以有cache这些网页到CDN, browser cache or even server cache，来缓解read-load...这里就用基本系统设计概念扩展就好了。

<<(3d). >>

|  |
| --- |
| 自己想问这四个大块（frontend engineer, translation service, user experience and deployment). 然后说好一个个来。感觉，哎，不错。还满organized的。然后开讲。然后问题来了。几乎不让你讲，要按他的思路。不是他要的关键词，一律不听。摇头。关键问题是，他用自己蹩脚的英文，没法问清楚自己想问的问题。搞个半天人家才知道他要干嘛。这个其实之前有面经也反映这个情况我当时看面经没在意。应该是不是那个candidate的问题，没听清人家说什么。果不期然，人家估计说的就是这个家伙。然后意外的又来了，我竟然音乐问到大蒜洋葱的味道。OMG! no excuse. seriously，这么小的房间，身上味道我忍了， 你打个这么味道大的嗝， 中午吃的是什么东西啊！我继续忍。他本来想问说http request里的 field有什么信息。花了很久时间， 问的问题，我不记得了，他问的问题是其他的。I'am sure. 然后他好不容易，问，user web brower type 这个link, 发生了什么？我就问，are we still talking about the translation system or you want me to talk more about anything happened during the request process? 继续摇头。不听。好吧。然后还有个地方，说API怎么设计，前一分钟还在讲前端的哦。后来莫名其妙地要说 translation service那里去。我说我们frontend 端讲完了吗？因为之前还在给他讲 JS template system 怎么用library call to decouple from the translation service. 我继续澄清，你是说这个API? 设计还是这个library API底层具体的Restful API？还有几处 都这样的情况。overall, 不让我讲大概框架，说好一个个来， 然后又莫名其妙的乱跳，一些workflow 莫名其妙地跳出来，我几次想跟他澄清，他很固执，自大。I seriously doubt his capability to explain his questions well. 看起来很organize, 但是脑子里其实就是一个浆糊。而且非常bossy. 关键是，你最终答出他想要的，他根本没有任何positive yes, 或者反应。非常奇葩。这个面经反应真的不是一例了。不知道什么样的人才和他对得上。很怀疑这种style和personality的人，很可能在google带不下去才去twitter, airbnb. |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

设计： 翻译系统 翻译系统，这个地里提到几次，我讲一下我的理解吧（个人理解，可能全错），主要考的是怎么管理这个流程， 包括UI developer， translator，怎么提速. 页面上应该有要I18N的词或句子 2. translator怎么能方便的去翻译 3. 什么样的管理流程

<<(3e). >>

第三轮Design: 面经上的翻译系统。主要问的是整体的System Design, 我的解法是先估算出所有翻译的容量，估算出来才几百兆，所以都可以存到Cache里面。然后Frontend server每个instance用memcache, Backend Server用shared Redis，Frontend 如果cache里找不到再call backend。面试官问的最tricky的是如果翻译更新了，如何迅速更新cache，我说用pubsub先broadcast给Backend，Backend更新好了Redis再broadcast给Frontend。面试官感觉对我的design非常满意，半个小时不到就结束了，然后开始愉快的吹水。

<<(4). Group Chat>>

<<jwl2006 发表于 2018-12-20>>

3. Archetecture: group chat. 问的很细，设计整体架构，schema设计，traffic estimation, 问了如何区分已读未读信息，如何获取前500个group的数据，每页显示5个，这里要用index排序，并且用timestamp composite index.这轮感觉还行

<< [cicilila](https://www.1point3acres.com/bbs/home.php?mod=space&uid=61573) 发表于 2018-11-16 07:20:24 >>

onsite system design： 类似fb messager（要你support group chat)

基本上就是个微信？上面显示所有你参与的对话， 有一对一的， 也有group chat。 然后每一个对话可以被点开， 看到具体的对话内容。

<< dddxhh 发表于 2018-11-29 12:14:04 >>

round 6: system design : airbnb group chat

<<>>

第四轮Design: 巨坑的一道题目，是面试官最近刚解决的问题。题目背景：有很多data scientists，每天要运行很多的SQL Query来Calculate metrics, 很多metrics都是重复的，或者和别人的很接近，如何整合所有的计算。我毫无头绪，乱扯先规范所有的SQL QUERY，然后把SQL Query转换成Map Reduce Jobs, 每天全体运行一次，这样Input tables可以重用之类的。感觉完全没有答道面试管想要的方向。

<https://www.quora.com/What-is-Airbnbs-mission-vision-statement>

"四个core value" 指的是？

下面提到六个啊？

https://www.quora.com/What-is-Airbnbs-mission-vision-statement

Brian Chesky talks about six core/cultural values:

1. Champion the mission (by living the mission)

The mission is to live in this world where one day you can feel like you're home anywhere and not in a home, but truly home, where you belong.

2. Be a Host

To live in this world where you can be home, you have to provide hospitality and hosts provide hospitality. Airbnb is a hospitality company.

3. Simplify

4. Every frame matters

Look at the end to end experience of everything, irrationally paying attention to details.

5. Be a Cereal Entrepreneur

See: What's the story behind the Obama O's and Captain McCain's cereals that the Airbnb founders sold before Airbnb was really successful?

6. Embrace the adventure

Having that youthful spirit regardless of age, being open minded and adaptable.