21st February 2015

系统设计大全

Algorithms

- https://www.hackerrank.com/domains [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fwww.hackerrank.com%2Fdomains]
- https://oj.leetcode.com/problemset/algorithms/ [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Foj.leetcode.com%2Fproblemset%2Falgorithms%2F] LeetCode至少要刷三遍，付费部分的题建议花点钱看一下，舍不得孩子套不着狼
- http://lintcode.com/ [http://lintcode.com/]
- http://www.ninechapter.com/solutions/ [http://www.ninechapter.com/solutions/]
- http://www.geeksforgeeks.org/about/interview-corner/ [http://www.geeksforgeeks.org/about/interview-corner/]
- TopCoder Algorithm Tutorial: http://help.topcoder.com/data-science/competing-in-algorithm-challenges/algorithm-tutorials/ [http://help.topcoder.com/data-science/competing-in-algorithm-challenges/algorithm-tutorials/] 其中几何算法的教程要仔细看，Google特爱出几何题
- CC150: Cracking the Code Interview: http://www.valleytalk.org/wp-content/uploads/2012/10/CrackCode.pdf [http://www.valleytalk.org/wp-content/uploads/2012/10/CrackCode.pdf]

System Design

Here are some articles about system design related topics.

- How to Rock a Systems Design Interview [http://www.palantir.com/2011/10/how-to-rock-a-systems-design-interview/]
- System Interview [http://www.hiredintech.com/app#system-design]
- Scalability for Dummies [http://www.lecloud.net/tagged/scalability]
- Scalable Web Architecture and Distributed Systems [http://www.aosabook.org/en/distsys.html]
- Numbers Everyone Should Know [http://everythingisdata.wordpress.com/2009/10/17/numbers-everyone-should-know/]
- Scalable System Design Patterns [http://horicky.blogspot.com/2010/10/scalable-system-design-patterns.html]
- Introduction to Architecting Systems for Scale [http://lethain.com/introduction-to-architecting-systems-for-scale/]
- Transactions Across Datacenters [http://snarfed.org/transactions_across_datacenters_io.html]
- A Plain English Introduction to CAP Theorem [http://ksat.me/a-plain-english-introduction-to-cap-theorem/]
- The CAP FAQ [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fgithub.com%2Fhenryr%2Fcap-faq]
- Paxos Made Simple [http://research.microsoft.com/en-us/um/people/lamport/pubs/paxos-simple.pdf]
- Consistent Hashing [http://www.tom-e-white.com/2007/11/consistent-hashing.html]
- NOSQL Patterns [http://horicky.blogspot.com/2009/11/nosql-patterns.html]
- Scalability, Availability & Stability Patterns [http://www.slideshare.net/jboner/scalability-availability-stability-patterns]

[https://www.blogger.com/null] Hot Questions and Reference:

There are some good references for each question. The references here are slides and articles.

Design a CDN network
Reference:

- Globally Distributed Content Delivery [http://www.akamai.com/dl/technical_publications/GloballyDistributedContentDelivery.pdf] .

Design a Google document system
Reference:

- google-mobwrite [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fcode.google.com%2Fp%2Fgoogle-mobwrite%2F]
- Differential Synchronization [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fneil.fraser.name%2Fwriting%2Fsync%2F] .

Design a random ID generation system
Reference:

- Announcing Snowflake [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fblog.twitter.com%2F2010%2Fannouncing-snowflake]
- snowflake [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fgithub.com%2Ftwitter%2Fsnowflake%2F] .

Design a key-value database
Reference:

- Introduction to Redis [http://www.slideshare.net/dvirsky/introduction-to-redis] .

Design the Facebook news seed function
Reference:

- What are best practices for building something like a News Feed? [http://www.quora.com/What-are-best-practices-for-building-something-like-a-News-Feed]
- What are the scaling issues to keep in mind while developing a social network feed? [http://www.quora.com/Activity-Streams/What-are-the-scaling-issues-to-keep-in-mind-while-developing-a-social-network-feed]
- Activity Feeds Architecture [http://www.slideshare.net/danmckinley/etsy-activity-feeds-architecture]

Design the Facebook timeline function
Reference:

- Building Timeline [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fwww.facebook.com%2Fnote.php%3Fnote_id%3D10150468255628920]
- Facebook Timeline [http://highscalability.com/blog/2012/1/23/facebook-timeline-brought-to-you-by-the-power-of-denormaliza.html] .

Design a function to return the top k requests during past time interval
Reference:

- Efficient Computation of Frequent and Top-k Elements in Data Streams [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Ficmi.cs.ucsb.edu%2Fresearch%2Ftech_reports%2Freports%2F2005-23.pdf]
- An Optimal Strategy for Monitoring Top-k Queries in Streaming Windows [http://davis.wpi.edu/xmdv/docs/EDBT11-diyang.pdf]

Design an online multiplayer card game
Reference:

- How to Create an Asynchronous Multiplayer Game [http://www.indieflashblog.com/how-to-create-an-asynchronous-multiplayer-game.html]

- How to Create an Asynchronous Multiplayer Game Part 2: Saving the Game State to Online Database [http://www.indieflashblog.com/how-to-create-async-part2.html]
- How to Create an Asynchronous Multiplayer Game Part 3: Loading Games from the Database [http://www.indieflashblog.com/how-to-create-async-part3.html]
- How to Create an Asynchronous Multiplayer Game Part 4: Matchmaking [http://www.indieflashblog.com/how-to-create-async-part4-html.html#comment-4447]
- Real Time Multiplayer in HTML5 [http://buildnewgames.com/real-time-multiplayer/]

Design a graph search function
Reference:

- Building out the infrastructure for Graph Search [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fwww.facebook.com%2Fnotes%2Ffacebook-engineering%2Funder-the-hood-building-out-the-infrastructure-for-graph-search%2F10151347573598920]
- Indexing and ranking in Graph Search [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fwww.facebook.com%2Fnotes%2Ffacebook-engineering%2Funder-the-hood-indexing-and-ranking-in-graph-search%2F10151361720763920]
- The natural language interface of Graph Search [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fwww.facebook.com%2Fnotes%2Ffacebook-engineering%2Funder-the-hood-the-natural-language-interface-of-graph-search%2F10151432733048920] and Erlang at Facebook [http://www.erlang-factory.com/upload/presentations/31/EugeneLetuchy-ErlangatFacebook.pdf] .

Design a picture sharing system
Reference:

- Flickr Architecture [http://highscalability.com/flickr-architecture]
- Instagram Architecture [http://highscalability.com/blog/2011/12/6/instagram-architecture-14-million-users-terabytes-of-photos.html] .

Design a search engine
Reference:

- How would you implement Google Search? [http://programmers.stackexchange.com/questions/38324/interview-question-how-would-you-implement-google-search]
- Implementing Search Engines [http://www.ardendertat.com/2012/01/11/implementing-search-engines/]

Design a recommendition system
Reference:

- Hulu's Recommendation System [http://tech.hulu.com/blog/2011/09/19/recommendation-system/]
- Recommender Systems [http://ijcai13.org/files/tutorial_slides/td3.pdf]

Design a tinyurl system
Reference:

- System Design for Big Data-tinyurl [http://n00tc0d3r.blogspot.com/]
- URL Shortener API [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fdevelopers.google.com%2Furl-shortener%2F%3Fcsw%3D1] .

Design a garbage collection system

Reference:

- Baby's First Garbage Collector [http://journal.stuffwithstuff.com/2013/12/08/babys-first-garbage-collector/] .

Design a scalable web crawling system

Reference:

- Design and Implementation of a High-Performance Distributed Web Crawler [http://cis.poly.edu/suel/papers/crawl.pdf]

Design the Facebook chat function

Reference:

- Erlang at Facebook [http://www.erlang-factory.com/upload/presentations/31/EugeneLetuchy-ErlangatFacebook.pdf]
- Facebook Chat [https://www.evernote.com/OutboundRedirect.action?dest=https%3A%2F%2Fwww.facebook.com%2Fnote.php%3Fnote_id%3D14218138919%26id%3D9445547199%26index%3D0]

Design a trending topic system

Reference:

- Implementing Real-Time Trending Topics With a Distributed Rolling Count Algorithm in Storm [http://www.michael-noll.com/blog/2013/01/18/implementing-real-time-trending-topics-in-storm/]
- Early detection of Twitter trends explained [http://snikolov.wordpress.com/2012/11/14/early-detection-of-twitter-trends/]

Design a cache system

Reference:

- Introduction to Memcached [http://www.slideshare.net/oemebamo/introduction-to-memcached] .

由maxthon于 21st February 2015发布

标签: 设计题

0 添加评论

输入您的评论...

评论： Xincun Li (Goo ▼ 退出

发布 预览 通知我