Get the data from office and update the study material

--------------------------------------------------------------------------------------------------------------------------

SYSTEM DESIGN

--------------------------------------------------------------------------------------------------------------------------

1. Sharding
2. Micro-services Architecture and basics
3. Service discovery in Microservices (<https://dzone.com/articles/service-discovery-in-a-microservices-architecture>)
4. Client side

The client is responsible for determining the network locations of available service instances and load balancing requests across them. The client queries a service registry, which is a database of available service instances. The client then uses a load-balancing algorithm to select one of the available service instances and makes a request. The network location of a service instance is registered with the service registry when it starts up. It is removed from the service registry when the instance terminates. The service instance’s registration is typically refreshed periodically using a heartbeat mechanism.

1. Server side

The client makes a request to a service via a load balancer. The load balancer queries the service registry and routes each request to an available service instance. As with client-side discovery, service instances are registered and deregistered with the service registry.

1. Replication is for scaling.

Replication -> All writes to one DB(Master) and all reads from another DB(Slave).

This model can be applied when there is more need for reads than writes.

How to synchronise between master and slave?

How to handle consistency in case of Replication in any DB

--------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------------

DS & ALGO

--------------------------------------------------------------------------------------------------------------------------

<http://www.geeksforgeeks.org/amazon-interview-experience-set-280-on-campus-for-internship/>

<http://www.practice.geeksforgeeks.org/tag-page.php?tag=Amazon&isCmp=1>

<http://www.geeksforgeeks.org/amazon-interview-experience-set-268-experienced/>

<http://www.geeksforgeeks.org/amazon-interview-experience-set-262-for-sde1/>

<http://www.geeksforgeeks.org/dynamic-programming-set-27-max-sum-rectangle-in-a-2d-matrix/>

<http://www.geeksforgeeks.org/dynamic-programming-set-7-coin-change/>

Longest Increasing Subsequence

Longest Common Increasing Subsequence

<http://www.geeksforgeeks.org/largest-subarray-with-equal-number-of-0s-and-1s/>

<http://www.geeksforgeeks.org/cpp-implementation-minesweeper-game/>

<http://introcs.cs.princeton.edu/java/42sort/>

Back-Tracking :-

<http://www.geeksforgeeks.org/find-paths-from-corner-cell-to-middle-cell-in-maze/>

Binary Tree :-

Right View

<http://www.geeksforgeeks.org/print-right-view-binary-tree-2/>

Cousins

<http://www.geeksforgeeks.org/print-cousins-of-a-given-node-in-binary-tree/>

DESIGN DATASTRUCTURE :-

DESIGN A DICTIONARY :-

<http://www.geeksforgeeks.org/data-structure-dictionary-spell-checker/>

<http://www.geeksforgeeks.org/find-last-unique-url-long-list-urls-single-traversal/>

<http://www.geeksforgeeks.org/how-to-check-if-a-given-array-represents-a-binary-heap/>

HASHING :-

Birthday Paradox:-

<http://www.geeksforgeeks.org/birthday-paradox/>

Seperate Chaining :-

<http://www.geeksforgeeks.org/implementing-our-own-hash-table-with-separate-chaining-in-java/>

Cukoo Hashing :-

<http://www.geeksforgeeks.org/cuckoo-hashing/>

<http://www.geeksforgeeks.org/count-distinct-elements-in-every-window-of-size-k/>

<http://www.geeksforgeeks.org/find-number-of-employees-under-every-manager/>

Print Paths:-

<http://www.geeksforgeeks.org/print-all-possible-paths-from-top-left-to-bottom-right-of-a-mxn-matrix/>

<http://www.geeksforgeeks.org/longest-possible-route-in-a-matrix-with-hurdles/>

TO UNDERSTAND :-

1) Suffix tree

2) Subset pbm

3) LIS

4) Singly Linked List -> Palindrome

-> Reverse without stack

5) Bin Tree :: Boundary Traversal -> Single Traversal

--------------------------------------------------------------------------------------------------------------------------

———————————————————-

KAFKA

———————————————————-

Custom partitioning for Kafka

www.javaworld.com/article/3066873/big-data/big-data-messaging-with-Kafka-part-2.html

harishshan.blogspot.in/2017/03/how-to-create-dynamic-Kafka-topic.html

———————————————————

IPB

———————————————————

Crawling - Indexing - Ranking

1. Filtering based on channel
2. Property on top of Filtering

Sybase version : Adaptive Server Enterprise - 15.0.3

Size of a row in doc table -

———————————————————-

Scope <parent - child>

MVC - CRUD

Row Mapper & Result Set Extractor

Join/wait/sleep/yield

Runtime in multithreading

Custom Exeception handling

Implicit Objects in JSP

doGet in multithreading

Odd-Even Number printing in Multithreading

10 pages to 1 in Angular JS

Cursor/Trigger/View/Index/Join/GroupBy

SOLID principles

———————————————————

AMAZON

———————————————————

1. Design a Logging System to archive the logs(Application & Server logs) emitted by large number of cluster nodes

How will the decide the message size ?

How will u split the file and convert it to the message?

Wat kinda storage

1. Messaging System like FB / WhatsApp
2. Push Notifications – Cricket scores

Akka cluster

Neo4j

———————————————————