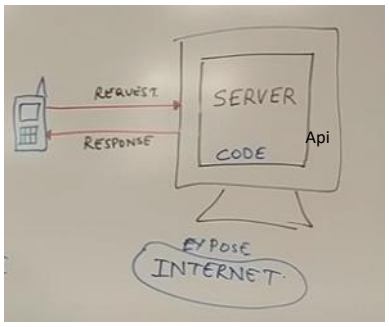
System Design Basics: Horizontal vs. Vertical Scaling

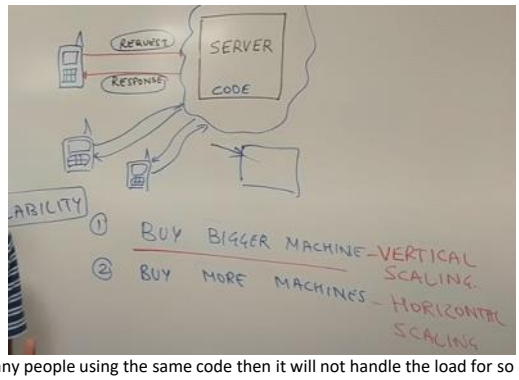
Monday, February 15, 2021 4:03 AM

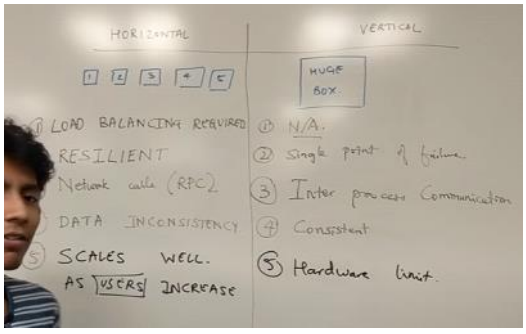


Api

1 . So I cant host it on to my machine it may be power cut. And lots of people paying money. So will host my code on cloud

Desktop vs Cloud-
1.cloud-> A set of computer somebody provides you. Remote logging. Configuration Relibility can be taken by cloud providers

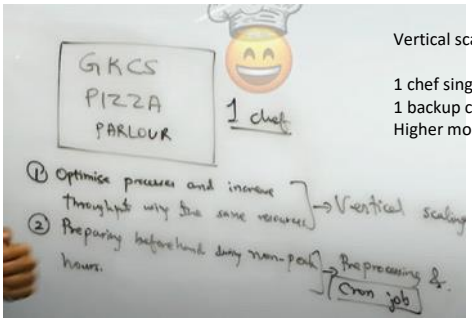




If many people using the same code then it will not handle the load for so many request
1.Bigger Machine //Scalibility vertical
2.Buy more machine // Horizontal scalibility

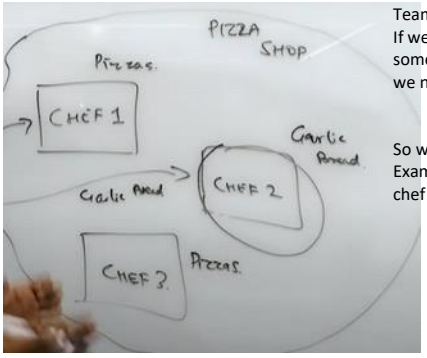
****

Distirbuted system



Vertical scaling

1 chef single point failure
1 backup chef (Master & slaves architecure)
Higher more chef horizontal scaling

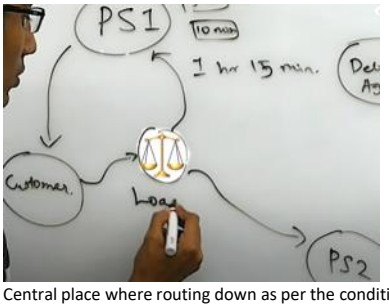


Team of garlic bread
If we want to change something in garlic bread then we need to refer chef 2

So we will scale at different rate
Example pizza more so pizza chef more higher

If electricty goes out->
We take backup
Local people will be served with second shop

Example if we want to get response then local server should be there everywhere in region



Central place where routing down as per the condition of serevr

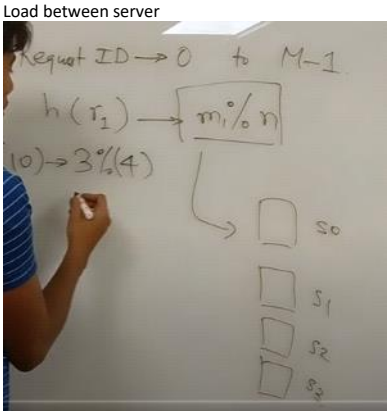
Decoupling the service

Logging & metrics-> Analyis
Extensible extneds->

****

Load balancing

Load between server



If there are n server. Then mobile will generate one id then we will hash it and mod it then the mod value will put and process in that server

Garbage collector algorithim->

How java knows its is garabage-> First it see that what is the object which cannot be reached by program. If you cant use them then it is as good as garbage

Tricolour algorithim

****

LoadBalancing-> Request id when the mobile send request it randomly generate the number 0 to m-1

Take this request id and hash it. Get particular number. This number mapped on some server.

Problem is adding and removing the server

LoadBalncer principal: We don't have duplicate on same server