

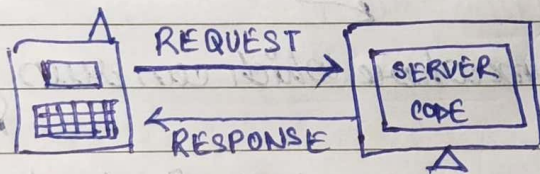
LOW LEVEL DESIGN

High level \rightarrow Deploying on servers \rightarrow figuring out how a system will interact with other

Low level design \rightarrow low level system has a lot more to do with how you are actually going to code

eg making classes, making objects, functions, signatures
* How to write efficient code and clean code

System Design Basic: Horizontal Vs Vertical Scaling 28/01/2025



Imagine you have a computer with you in which we have written an algorithm

Some code is running in the computer

Code (function) \rightarrow Takes Input \rightarrow Gives Output
(Request) (Response)

* compeople look to the code & find it useful for them

→ Pay you to use the code

↳ We cannot give our computer to other

↳ Expose our code using some protocol eg. Internet

↳ Expose the code using API (Application Programming Interface)

* Exposing the code, it will give an output. Rather than return the response
we send request and we get a response

Setting Up A computer

1) It requires a database connected to desktop

2) Configure the end points that people are connecting to

3) Consider if power loss → host the service to a cloud
we cannot afford if service goes down (People paying money for you)

Desktop Vs Cloud

↓
Your own 1 computer
↳ set of computer that somebody provides you for money
eg AWS → Amazon web Service

if we pay AWS they will provide you computation power

A desktop that they have somewhere which can run your Algo

* To store you algorithms → Do a remote login to their desktop

Cloud → Set of desktops/set of computers that can use to run your service

* Reason we would do this because the configuration, the settings, the reliability can be taken care by the solution providers

* Now we have server Hosted on a cloud, that is basically some computer that we don't know about.

Page No.:

Date:

YOUVA

Business Requirements :-

- 1) There are a lots of people who are using our algorithm now
- 2) It gets to a point where the code that you are running on the machine is not able to handle all of these connections.

Buy A Bigger Machine (Vertical Scalling)

Buying More Machines (Horizontal Scalling)

* The ability to handle more request by buying more machine or buying bigger machine is called Scalability.

- 1) Buy Bigger Machine → Computer will be larger
 - ↳ It can process request faster
 - ↳ This is called VERTICAL SCALING

- 2) Buying More Machines → It means request can fall on any one of the machine and it will be processed.
 - ↳ Because of them, the request can be randomly distributed amongst the machines that you have just bought.
 - ↳ This is called Horizontal Scalling

* With these 2 ways we can increase the scalability of the system.

* Scalability is being able to handle more requests.

Horizontal Scalling

Vertical Scalling

1) 2) 3) 4) 5) 6) 7) 8) 9) 10)

huge box

1) Load Balancer Required

1) NA

2) If one machine fails we can re-route the request to other one [RESILIENT]

2) Single point of failure

3) Slow → All the communication

3) Fast → Inter process communication

between the server will be over network

* Network calls are slow Its I/O

RPC → REMOTE PROCEDURE CALLS (Show)

HORIZONTAL

1 2 3 4 5 6 7

4/ Data consistency is low as I sends the data to other and so on

* Data is complicated to maintain.

* If there is a transaction where the operation has to be atomic, we have to lock all server → Impracticable

* Here we have loose transactional guarantee

4/ Data consistency is the Real Issue

5/ Scales well as user increase

In Real world → we take the good quality of both the scaling

eg horizontal → ① Resilient

② Scales well as user increase

Vertical → ① Fast interprocess communication

② Data is consistent

eg the cache is consistent

↳ No dirty read ↳ No dirty write

* Hybrid Solution → Horizontal Scaling with Big Box

Small firm → vertical scaling, mid firm → horizontal scaling

Big firm → Hybrid Solution

VERTICLE

1 Huge box

4/ There is just one system on which all the data resides. Hence it is consistent

4/ Data is very consistent.

5/ Sometimes we have hardware limitation that we cannot extent a computer after a certain limit.