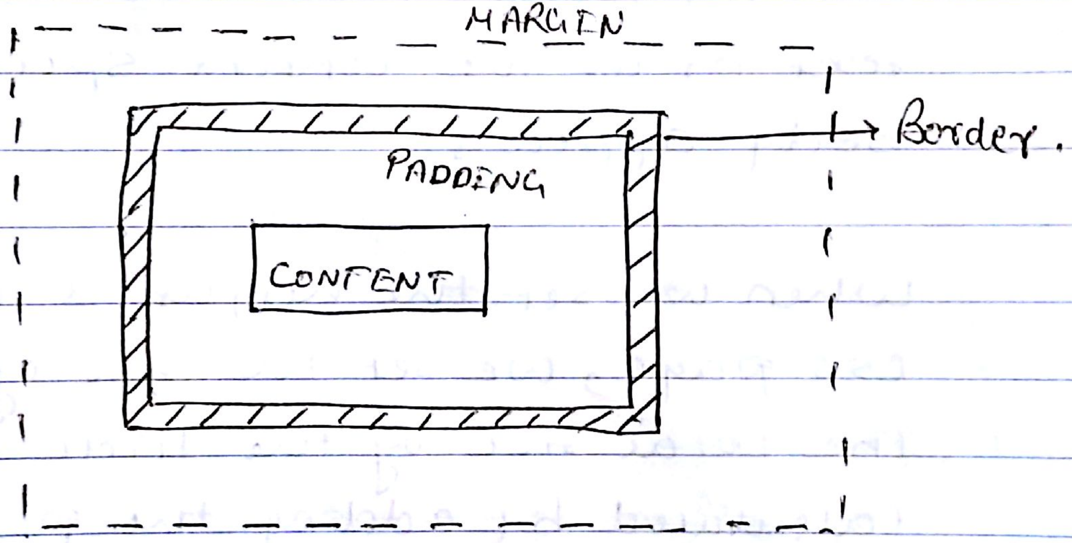
WPL

2.a) CSS Box Model

The box model applies to all HTML elements. It is basically used for the design & layout of HTML web pages.
The components of box model are:

- ○ Margin
- ○ Border
- ○ padding
- ○ content



The above diagram shows the CSS Box model.

i) The margin is transparent, it is the clear area (white space) present outside the Border.

2) The border can be either opaque or transparent (we can specify border style & types with thickness).
we can specify the border color which will inherit the box color if unspecified

3) Padding is the area around the content. This is usually transparent.

4) Content is the main area where the text or all the content specified in the body appears.

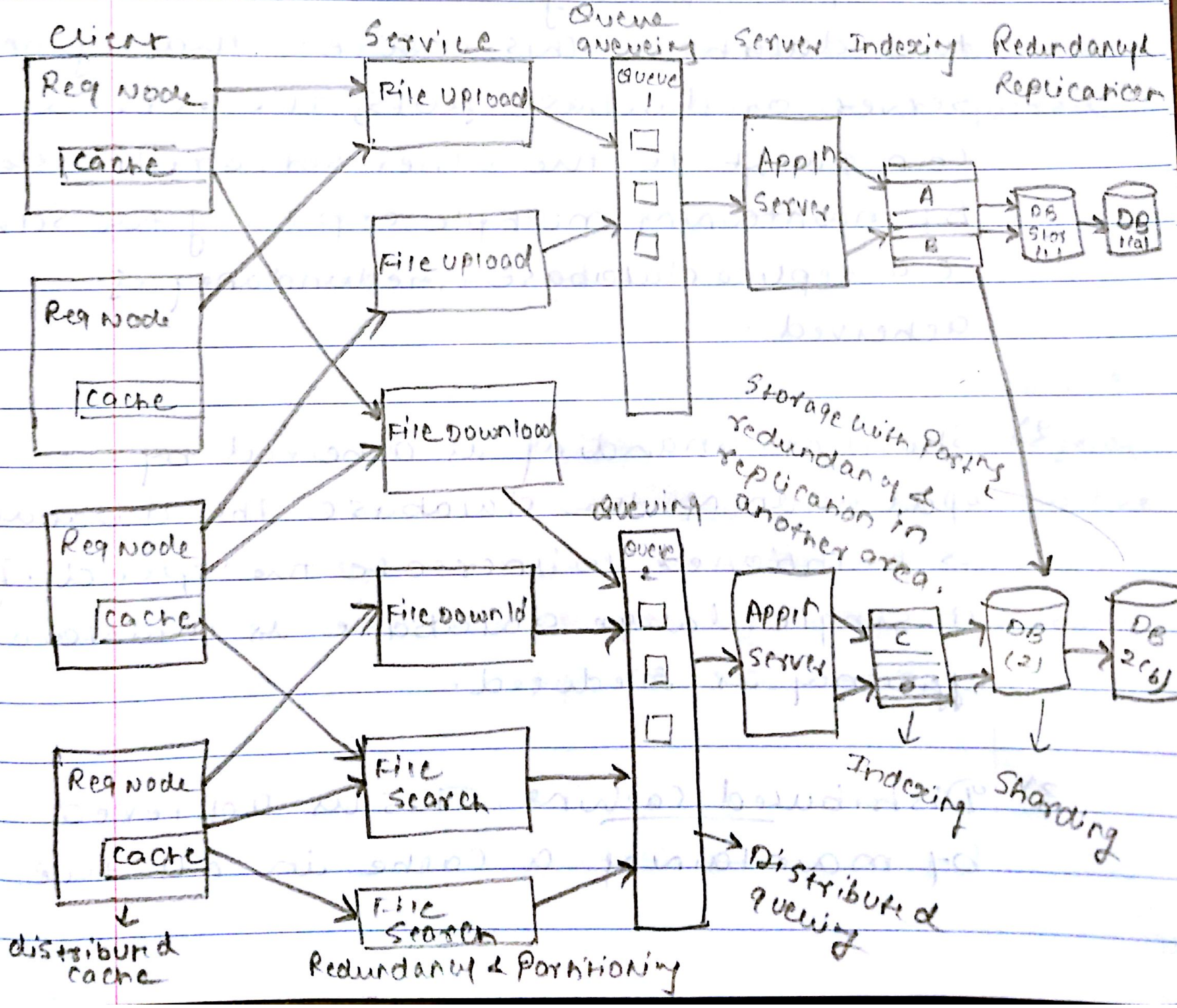
When we set the height & width in a CSS page, we set the size of the intent. The total size of the HTML element is calculated by adding the padding, borders, margin & the content area

For example:

```
div {
      width: 200px;
      Padding: 10px;
border: 5px;
margin: 5px;
}
```

The total width will be' 200+20(padding left + padding right) + 10(border left + border right) + 10(margin left + margin right)

= 240px



The diagram shows the scalable architecture for file upload, download & file browsing.

The following concepts are covered in diagram:

1) Redundancy: This is achieved by storing multiple copies of the server as well of the database. This ensures that if one server or database fails, the service can point to the other server. Therefore by maintaining multiple copies of the server & a replica database, redundancy is acheived.

2) Sharding: Sharding is acheived by partitioning the database. The database is partitioned to increase the speed. If a single large database is maintained efficiency is reduced.

3) Distributed caching: This is achieved by maintaining a cache in all the

requesting nodes. If data is present in the cache, it is directly retrieved rather than requesting the server everytime.

4) Indexing: Indexing technique is appued for the database storage & retrieved. The index maintains key values & a hashing technique to store and retrieve data in the database into the required location.

5) Distributed Queuing:

It is achieved by maintaining 2 queues for the servers. The request first goes to the queue, which then sends it to the server based on its availability & based on the request priority. This maintains load on the application server.