Scaling

Introduction

Scaling is increasing resources and performance with increasing load and traffic over the existing system without affecting the complexity.

For example- we designed a system for 2000 requests per minute, and our system is running fine, but we need to expand the business and need to handle more requests in the same or less time. Now we would require more resources to handle all requests and computing power. Now let's say we can achieve 10k requests per minute.

Scaling is essential to increase the system and solve the current problem on a larger scale, so we should design our system to scale our system to a larger-scale whenever required.

- The load will be increased at any point in time, so we need enough resources to handle the increasing load.
- The system shouldn't be highly complex so that it's easy to scale at any point in time.
- Performance should always be increased with scalability.

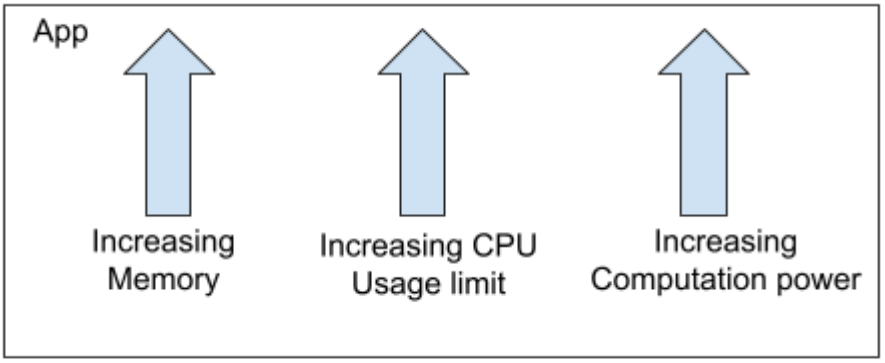
Types of Scaling

There are two types of scaling there horizontal and vertical scaling.

Vertical Scaling(Scaling up)- Increasing the same resource's capacity to fulfill the need and maximize throughput is called Vertical Scaling.

For example- let's say our current architecture was handling 1000 request per second, but load increased on the website, and our system is able to handle the increasing load

by affecting the configuration or, say, increasing storage capacity, adding fast methodologies to increase computation power and performance increases, this is an example of vertical scaling.



Able to server more requests by Vertical Scaling

Horizontal Scaling (Scaling out)- Increasing the number of comprehensive resources to serve the system's scaling need w.r.t. increased traffic/load.

For Example- let's say our system can handle the increased load, but the database is not able to handle all these requests at a single server. So even if our system is increasing throughput but we can't able to serve more requests because of constraints. In such a case, we apply horizontal scaling and make replicas or employ master-slave architecture to add more resources into the existing server to increase throughput.

App Instances are increasing ie Horizontal Scaling