Scaling

Scalability is the measure of the ability of a system to increase or decrease in performance and cost, in response to changes in application or with increasing load and traffic on an existing system.

In order to build a scalable system we need to focus on these 3 areas:

1) System is able to handle increased load.
2) Not increase the complexity of the system.
3) Performance of the overall system should not take a hit.

It is of 2 types: -

1) Vertical Scaling(Buy Bigger Machine) -
   Vertical scaling, also called “Scaling Up” refers to the process of adding more resources to a single node in a system, such as increasing the amount of memory, storage, or processing power. This is done to improve the performance and capacity of the system.
   Advantages:-
   - Interprocess Communication - All the processing occurs in a single node, hence processing is faster.
   - Data remains Consistent

   It has some limitations, such as:-
   - Finite limit to the amount of resources that can be added to a single system can handle increased load without a decrease in performance.

   Disadvantages:-
   - Increased costs as the node becomes more powerful.
   - Single Point of Failure

2) Horizontal Scaling(Buy more machines) -
   Horizontal scaling, also called "Scaling Out" refers to the process of adding more nodes to distribute workloads and improve capacity of the given system. The goal is to spread the workload across multiple nodes, so that the
   - Uses Network calls / Remote Procedure Calls(RPCs) - Nodes communicate with each other using RPCs (RPC is a function call made between two nodes when they are physically separated).
   - Data Inconsistency.

   It uses a Load Balancer to distribute incoming traffic across multiple nodes.
   Advantages:-
   - Resilient - If one machine fails, The requests are transported to other nodes, hence no single point of failure occurs.
   - Scales well as the number of users increase.

Difference between Horizontal and Vertical Scaling:-

Horizontal Scaling	**Vertical Scaling**
1) Process of adding more nodes to distribute workloads.	1) Process of adding more resources to a single node in a system.
2) Load Balancer is required	2) No load balancing required.
3) Resilient System	3) Single Point of failure can occur
4) Uses Remote Procedure Calls (RPCs).	4) Interprocess Communication occurs.
5) Data Inconsistency	5) Data is Consistent
6) It scales well as the number of users increases.	6) It can reach a hardware limit (Cannot add more resources).

Hybrid Solution is to use Horizontal Scaling together with vertical scaling.

Pizza shop example:

1. Vertical Scaling: Optimise processes and increase throughput using the same resource.
2. Pre-Processing and Cron job: Preparing beforehand at non peak hours.
3. Backups: Keep backups and avoid single points of failure.
4. Horizontal Scaling: Hire more resources.
5. Micro-Service Architecture: we will define all the responsibilities to a chef and there’s nothing outside our business use case that they handle.
6. Distributed System (Partitioning) : open a new shop as a backup, route all your requests to this so that any order that is local or very close range to this shop can be handled by it.
7. Load Balancer: It can make intelligent business design and route requests in the smart way.
8. Decoupling: separating out concerns so that we can handle separate systems more efficiently.
9. Logging and metrics calculation : log every event and everything for analysis, auditing, reporting and machine learning algorithms to find sense out of those events.
10. Extensible : we don’t want to re-write the code again and again to serve a different purpose. Example: delivery agent doesn't need to know he is delivering a pizza, it could be a burger tomorrow.