Scalability

What is scalability in system design?

Scalability measures a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

Scalability can be achieved in two ways: Vertical Scaling and Horizontal Scaling.

1. Vertical Scaling

Vertical scaling increases the scale of a system by adding more configuration or hardware for better computation power or storage. In practice, this would mean adding better processors, increasing RAM, or other power-increasing adjustments. Scaling here is done through multi-core by spreading the load between the CPU and RAM resources.

Example: MySQL

Pros of Scaling-Up

- It consumes less power as compared to maintaining multiple servers
- Administrative efforts are reduced as a single machine is to be managed
- Cooling costs are lesser
- Reduced software costs
- Implementation becomes easy
- The application compatibility is retained

Cons of Scaling up

- There is a high risk of hardware failure which can cause bigger problems.
- There is less scope for upgrading the system.
- It can become a Single point of failure.
- There is a limit to increase resources like memory storage, RAM as one single machine might not be able to handle it.

2. Horizontal Scaling

Horizontal scaling is a process of increasing the scale of a system by adding more machines. This entails collecting and connecting multiple devices to take on more system requests.

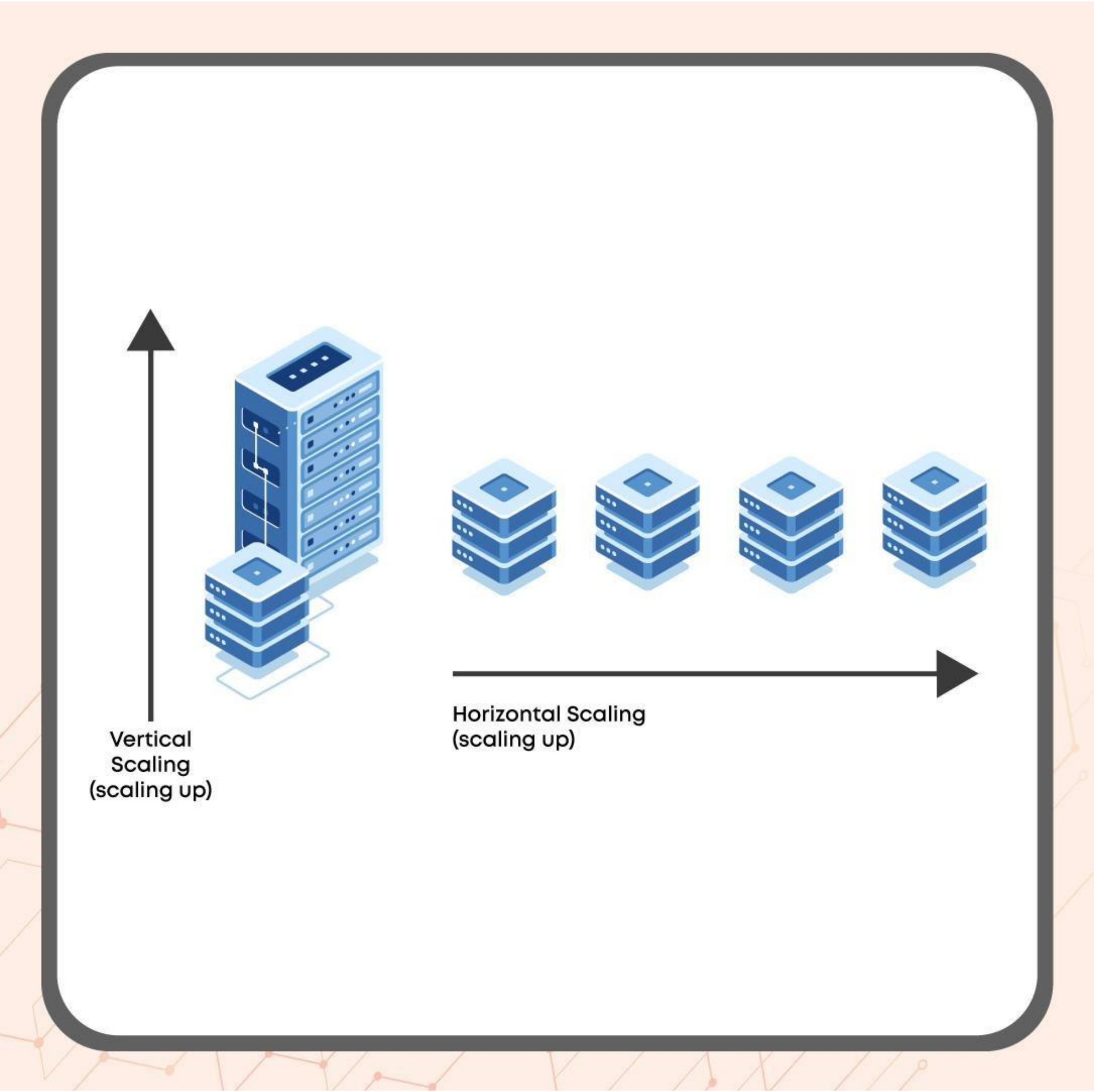
Example: Cassandra, MongoDB

Pros of Scaling-out

- Cost-effective compared to scaling-up.
- Takes advantage of smaller systems.
- Easily upgradable.
- Resilience is improved due to the presence of discrete, multiple systems.
- Fault tolerance can be handled easily.
- Supporting linear increases capacity.

Cons of Scaling-out

- The licensing fees are more.
- Utility costs such as cooling and electricity are high.
- It has a bigger footprint in the Data Center.
- More networking equipment is needed.



Limitations —

There is no limit to horizontal scaling, but it would not be cost or space-efficient compared to vertical scaling. Also, the performance of a single machine using horizontal scaling cannot continuously be increased in proportion to the hardware added because the performance curve comes to a standstill at the maximum limit.

Another limitation of vertical scaling is that it might be impossible to scale vertically after a limit due to limitations in hardware technology.

Scaling	Pros	Cons
Vertical Scaling	● Comparatively more areas of application ● Lower power consumption as compared to running multiple servers ● Easy to install and manage the hardware in a single machine	● Expensive, so more financial investments are required. ● Bigger outage due to hardware failure ● Low availability ● Limited upgradability in future
Horizontal Scaling	● Cheaper than vertical scaling ● High availability ● Easier to run integration testing and fault tolerance	● Limited applications ● Higher cost on utility like electricity ● Extra load since software has to handle data distribution and parallel processing

Vertical Scaling Vs Horizontal Scaling

Parameter	Horizontal Scaling	Vertical Scaling
Databases	It is based on the partitioning of data.	In this, the data lives on a single machine and scaling is done through multi-core. That is the load is divided between the CPU and RAM of the machine.
Downtime	Adding machines to the existing pool means making it possible to scale with less downtime.	Having a capacity of a single machine means scaling beyond its limit can lead to high downtime.
Data Sharing	Data sharing is complex in horizontal scaling as it consists of many machines.	Data sharing is easy in vertical scaling as single machine message passing can be done by just passing the reference.
Examples	MongoDB,Cassandra	MySQL, Amazon RDS