

What is NoSQL?

NoSQL, popularly misunderstood as No-SQL but it actually means **Not only SQL database**. It's a mechanism for storing and retrieving data and is also claimed as the next generation database. They are used in real-time web applications and big data and their usage is peaking over time.

Earlier databases weren't compatible with the scale and agility of modern applications, nor were they able to optimally use commodity storage and processing power which is available today. They, with time, weren't able to catch up with the flow of information demanded by people.

Hence, to cope up with all the parameters, in mid 1990's a non-relational database named NoSql was introduced.

- These are schema free
- Data structure being used for it is different from what we learned in Relational databases, data structures used here are more flexible than the relation used in relational databases.
- It can handle huge amounts of data (big data) and it's performance can be tuned by adding more machines to our clusters.
- Most of the NoSQL are open source and have a capability of horizontal scalability which means that commodity kinds of machines could be added.

Why NoSQL?

The Relational Databases we use are not designed to manage all kinds of data efficiently, like structured, unstructured and semi-structured data whereas NoSQL databases are designed to manage these types of data efficiently.

NoSQL databases go against the conventional attitude of storing information at a single location, instead it distributes and stores information over a set of multiple servers. The data are stored in flexible and fluid data models. This distribution of data helps the NoSQL database server to distribute the load at the database tier therefore it also signifies that the system can scale out rather than just scale up.

What is Scaling?

It is the ability to expand or concise the capacity of the system by either adding or removing the resources from it, in order to alter the usage of our application.

There is mainly two types of scaling:

- Horizontal Scaling.
- Vertical Scaling.

Horizontal Scaling:

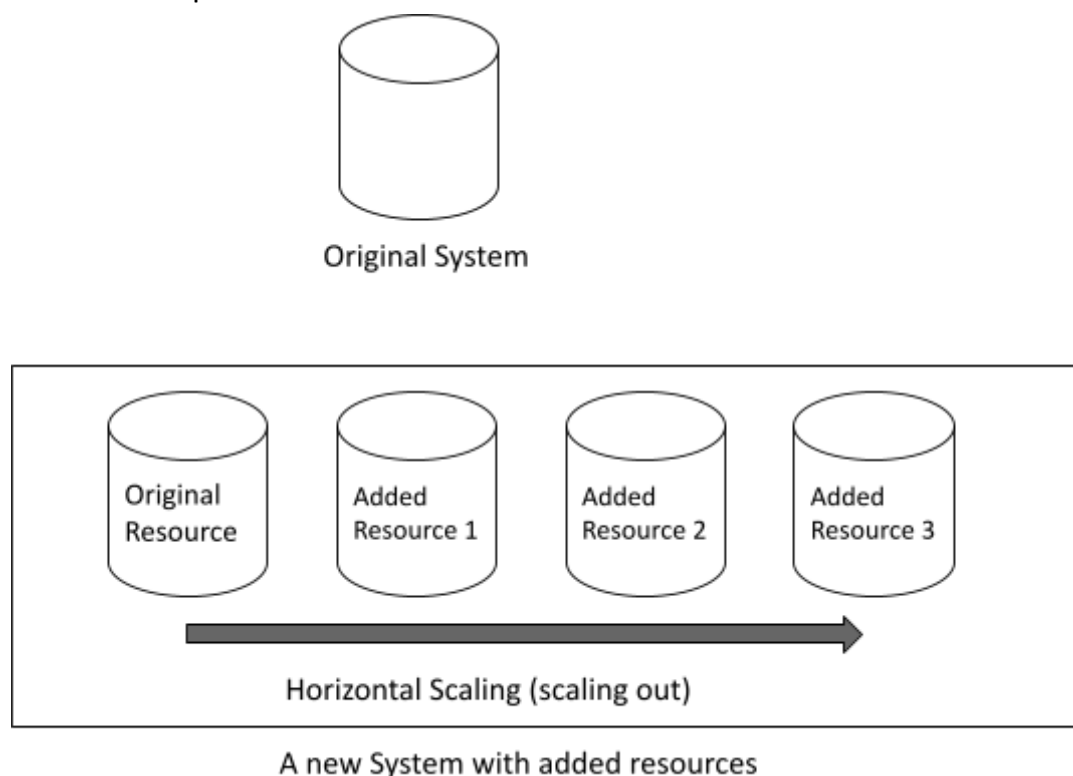
Horizontal Scaling, also known as **Scale out**, refers to addition of extra servers/nodes to distribute the load among them.

Now, this type of scaling puts up a little difficulties when dealing with relational databases because maintaining those relations while spreading the data out is tough to work out.

Although if we take into account the non-relational databases, then scaling out doesn't seem to be a bad option.

Therefore, to be little precise Horizontal Scaling refers to the addition of resources to the existing system to optimise the performance of the system accordingly.

A small example:



Three resources were added in the resources pool to build up a new system.

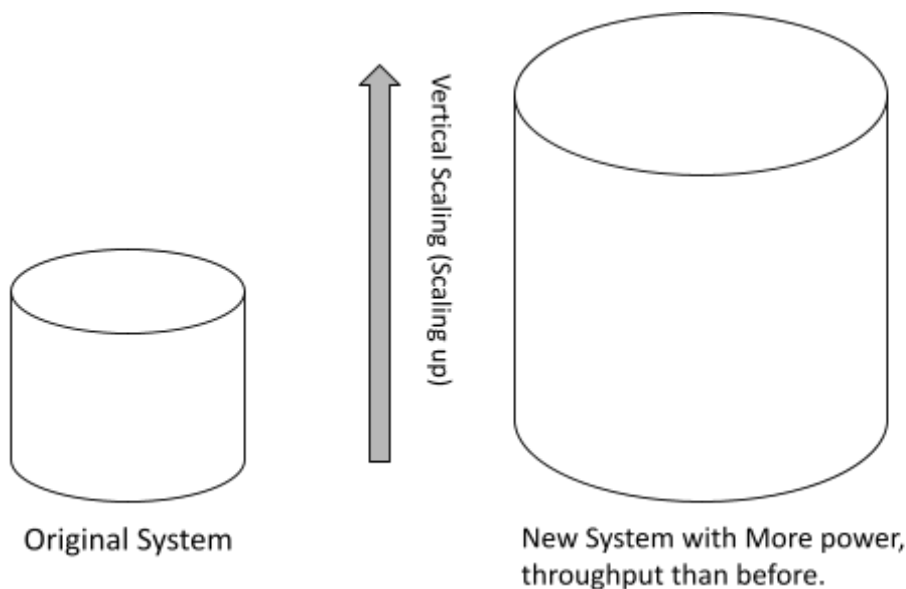
Vertical Scaling:

Vertical Scaling also known as **Scale up**, refers to increasing the power of the existing system/machine.

This type of scaling can be adapted by both relational as well as non-relational databases. But as access to everything is bad, scaling up also has a certain limit after which doing it further makes no sense.

Instead of combining multiple machines/servers we build one huge machine/server having much more power and throughput than the previous system.

Below is how scaling up is done.



Difference between Horizontal Scaling and Vertical Scaling:

Horizontal scaling	Vertical Scaling
Horizontal Scaling refers to the addition of resources to the existing system.	Vertical Scaling refers to increasing the computational power of the existing system/machine.
Difficult to implement.	Relatively, easy to implement.
Cost is expensive.	Not that expensive.