**Scalability**

Means, Flexible with demand.

**X – Axis Scaling:**

Also called Horizontal Scaling, It duplicates the application in many servers and load balance traffic in high availability manner. Improves the capacity/availability, this option does not have development cost implication but does cost in higher hosting and maintenance expense.

**Z – Axis Scaling:**

You can scale the application into z – axis and the application code are duplicated into several servers, similar to x – axis, but in this case the server is responsible for only a fraction of the data. The big difference is that each server is responsible for only a subset of the data. Some component of the system is responsible for routing each request to the appropriate server. One commonly used routing criteria is an attribute of the request such as the primary key of the entity being accessed. Another common routing criteria is the customer type. For example, an application might provide paying customers with a higher SLA than free customers by routing their requests to a different set of servers with more capacity.

In Short Data is Split

**Drawbacks:**

* One drawback is increased application complexity.
* Another drawback of Z-axis scaling is that doesn’t solve the problems of increasing development and application complexity. To solve those problems we need to apply Y-axis scaling.

**Y – Axis Scaling:**

In Y – Axis the application is decomposed into functionality, service or resource. The way you do this is entirely your choice. Unlike X-axis and Z-axis, which consist of running multiple, identical copies of the application, Y-axis axis scaling splits the application into multiple, different services. Each service is responsible for one or more closely related functions. There are a couple of different ways of decomposing the application into services.