A Guide to Horizontal vs Vertical Scaling

In 2023, it's estimated that 3.2 exabytes of data will be created every day on a global scale. In just two years (2025), the amount of data generated each day is expected to reach 463 exabytes. That is a 14,368.75% increase!

Needless to say, the amount of data organizations work with every day is increasing exponentially. As a result, data storage strategies that extend existing infrastructure, computing resources, and processing power are top-of-mind for IT and business professionals alike.

This guide to horizontal and vertical scaling will discuss the meaning of database scaling, scaling strategy, differences between horizontal vs vertical scaling, and considerations in choosing the right cloud scaling approach for your organization.

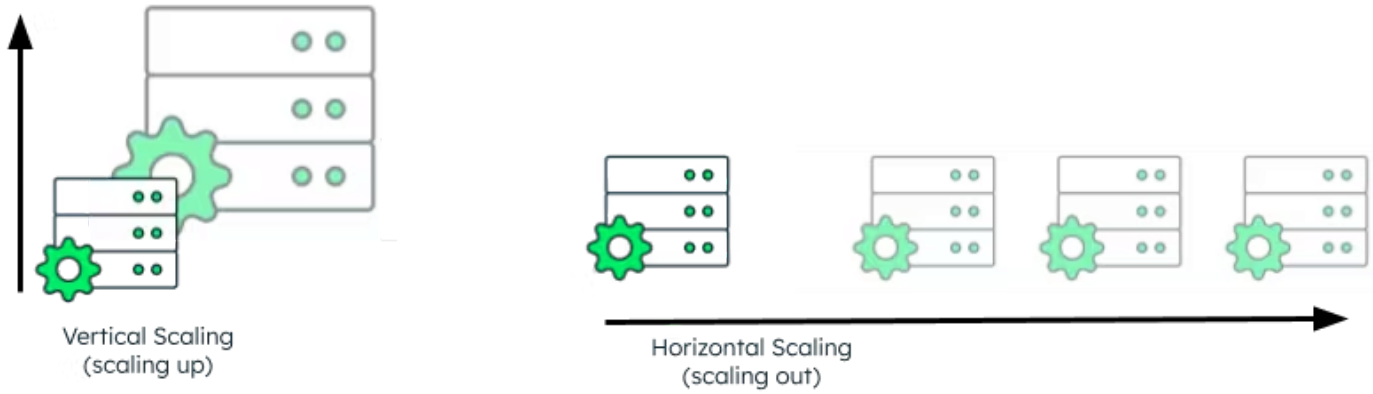
Table of Contents

- FAQs

What does database scaling mean?

The term database scaling refers to the process of changing database system capacity depending upon the amount of data or user activity occurring at any given time. There are two approaches to database scaling: horizontal and vertical scaling.

Learn more about database scalability.



What's the difference between horizontal and vertical scaling?

The vertical scaling approach

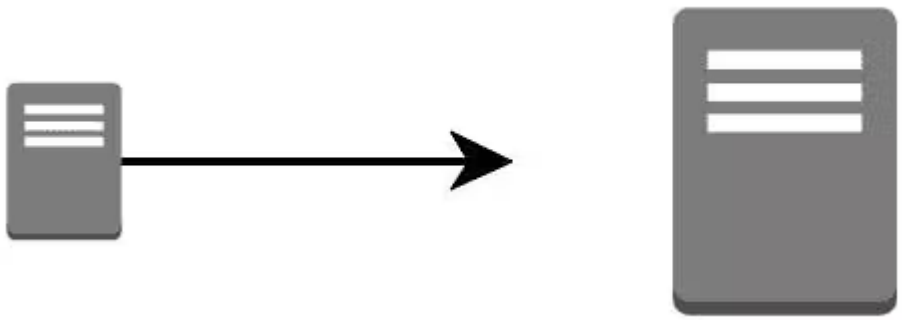
The vertical scaling approach, sometimes referred to as "scaling up," focuses on adding more resources or more processing power to a single machine. These additions may include CPU and RAM resources upgrades which will increase the processing speed of a single server or increase the storage capacity of a single machine to address increasing data requirements.

One of the advantages of scaling vertically is that it is easier than the alternative horizontal scaling approach. Specifically, since hardware resources and/or computing resources are only being added to one machine, there is less complexity involved. This extends to the initial vertical scaling of the existing machine or existing server, as well as less complicated maintenance going forward. Further, since only one machine is being upgraded, vertical

scaling is often a more economical choice in the short-term than the horizontal scaling approach.

With that said, there are some disadvantages to the vertical scaling approach as well. With one machine, there is a limit to the amount of upgrades or expansion that can occur. This means that an organization's scalability needs may not be met if their single machine doesn't have the necessary expansion capacity.

MongoDB Atlas makes it simple to vertically scale up or down as needed. You can even enable auto-scaling so your available resources always meet your needs.



Pros

- The main benefit of vertical scaling is that nothing changes about your database infrastructure other than the hardware specifications of the machine running the database.
- As such, it's transparent to the application. The only difference is that you have more CPUs, memory, and/or storage space.
- Vertical scaling is a good option to try first if massive storage and processing are not required.

Cons

- The downside of scaling up is that servers with more storage and processing power can be a lot more expensive.
- There is also a physical limit on the amount of CPUs, memory, network interfaces, and hard-drives that can be used on a single machine. For those scaling up using a cloud platform provider, you will eventually reach the highest tier of machine available.
- The available configuration options from cloud providers may be limited meaning that you may need only 10% more resources but the next available tier has 50% more resources for 50% more cost.
- If scaling vertically requires a migration between hardwares, it could result in downtime or service disruption.

When cost and/or machine limitations become an issue, be sure to consider **horizontal scaling**.

The horizontal scaling approach

The horizontal scaling approach, sometimes referred to as "scaling out," entails adding more machines to further distribute the load of the database and increase overall storage and/or processing power. There are two common ways to perform horizontal scaling – they include sharding, which increases the overall capacity of the system, and replication, which increases the availability and reliability of the system.

Pros

- Horizontal scaling is "infinitely scalable" as you can always add another machine while you may not be able to buy a larger machine if you are already using the largest machine available.
- There is more predictable increases in pricing. Adding additional machines is a predictable cost while upgrading to larger machines may have increasing costs for smaller gains in performance.
- Horizontal scaling can also deliver better performance and customer experience. An example of this is distributing with global clusters to deliver better performance in each region.

Cons

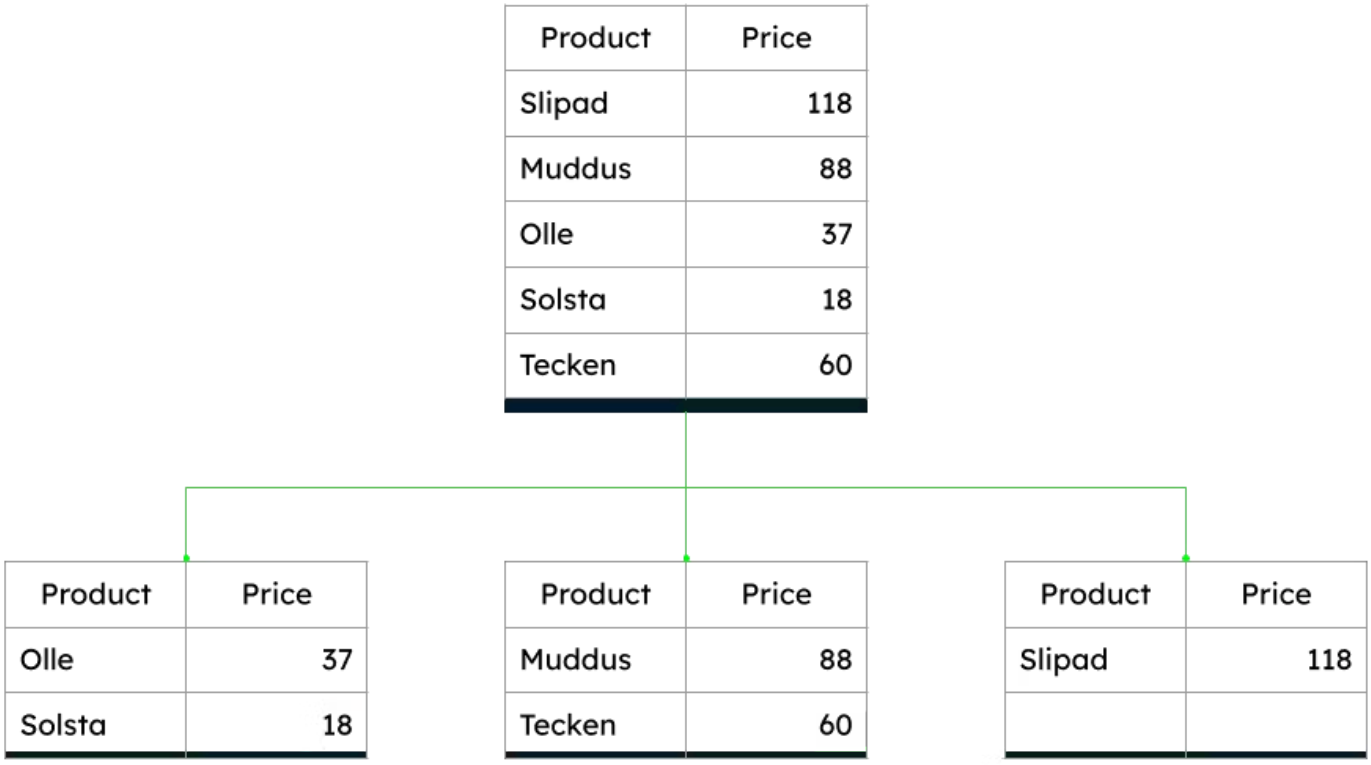
- Horizontal scaling may require application architecture and code changes due to the distributed nature of the data. How you store and query the data can significantly affect your application performance.
- Database systems that are scaled horizontally can be more complicated to manage and maintain, leading to more work for you and your team.

MongoDB implements horizontal scaling with a method called **sharding**.

Sharding

The sharding method of horizontal scaling involves dividing a large database into smaller, more manageable pieces (called shards) and then distributing the shards across multiple machines. Each shard contains a subset of the data, and each machine is responsible for storing and performing requests for a specific set of shards.

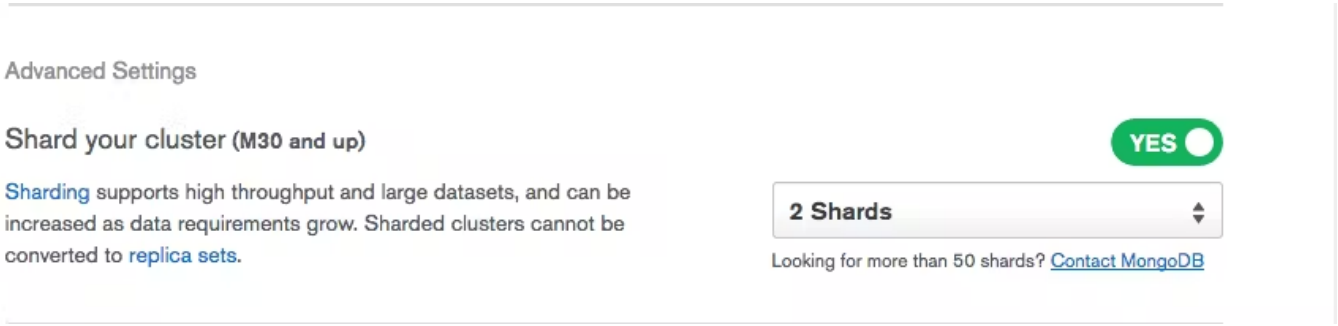
For example, in the illustration below, the data shard data subsets were divided by price range.



This approach to horizontal scaling improves the system's fault tolerance and availability, as a single point of failure in one machine does not affect the remaining machines. However, this approach also adds additional complexity as it's important to make sure data is evenly distributed across the shards and there is no duplicated or lost data.

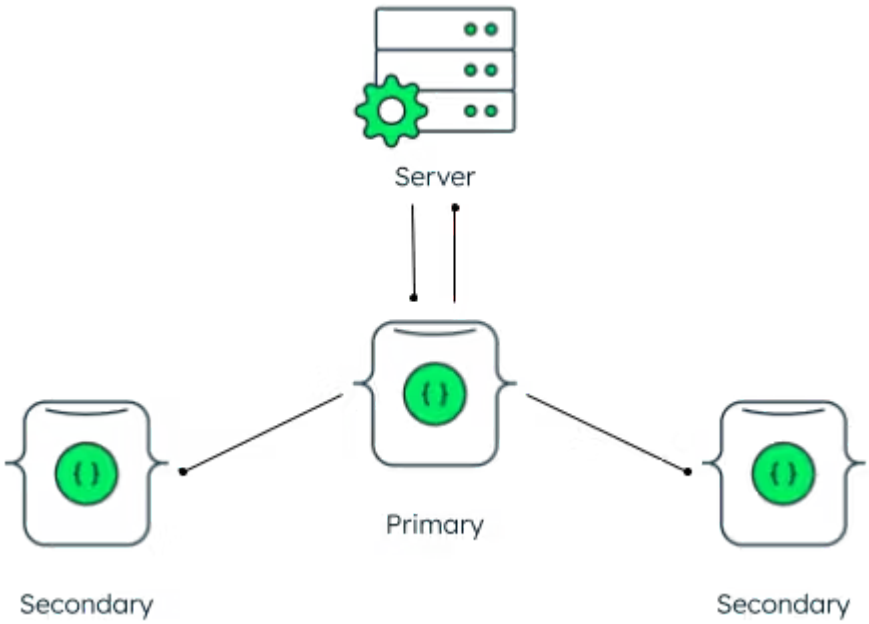
The easiest, most convenient, and most cost-effective way to deploy and manage a sharded cluster is via MongoDB Atlas, the Developer Data Platform that simplifies sharded cluster implementation.

As long as you are on a large enough sized dedicated cluster (M30 or larger), all you need to do is turn it on:



Replication

The replication method of scaling horizontally creates multiple copies of the same database on multiple machines. Usually, one machine is designated as the primary machine (e.g., first machine where database changes are made) and all database changes made to that database are propagated to all other database replicas (e.g., the other machines with the same database). This ensures that all instances of the database are up-to-date.



The advantage of this horizontal scaling method is that system availability and fault tolerance is greatly improved. Specifically, if the primary machine has an outage, one of the other existing machines can be promoted to the status of primary machine. And, since all machines have the same database with the same data stored, the system can continue to operate without interruption. In addition, due to the existence of more machines, improved performance can also occur as data requests can be distributed across multiple machines.

Some of the disadvantages to replication include the introduction of additional complexity and risk of duplicated or lost data (as with sharding). In addition, because replication

requires the use of multiple copies of the database across multiple machines, additional system traffic and storage requirements are also a concern. This can sometimes lead to additional system and personnel costs.

Replication is included by default in MongoDB Atlas providing higher availability and fault tolerance without any additional complexity or setup work.

Key differences between horizontal and vertical scaling

While both vertical scaling and horizontal scaling can increase the capacity and optimize the performance of a system, it's important to understand these differences before selecting the right approach for an organization.

Vertical scaling and horizontal scaling differences

Scalability differences

- **Vertical scaling:** Because vertical scaling focuses on enhancing the storage and processing capabilities of one machine, there is a limit to how far these enhancements can be taken. This limit is determined by the specifications of the machine being enhanced. There are also limitations in the options provided by cloud providers meaning that you may only need 10% more resources but the next available tier may have 50% more resources for 50% more cost.
- **Horizontal scaling:** Since horizontal scaling focuses on adding additional machines (e.g., scaling out), there is technically no limit to how far system enhancements can be taken. Of course, budget and system manageability are key factors to consider, but technically there is far more capacity within which to address organizational requirements.

System resiliency differences

- **Vertical scaling:** The use of one enhanced machine means that when necessary maintenance or upgrades are being completed, there will likely be a period of system downtime or some impact to user access and experience. In addition, if there is a hardware or software failure, system downtime is very likely given there is only one machine carrying organizational workloads. With that said, because there is only one

machine, downtime may be shorter in duration as only one machine needs to be examined, repaired, or restored.

- **Horizontal scaling:** Unlike vertical scaling, system maintenance or upgrades can often be completed in such a way that, while one machine or part of the system is unavailable, other machines can carry the workload of the machine in maintenance. This means that users are often unaware when certain machines are being worked on.

Complexity difference

- **Vertical scaling:** Since vertical scaling relates to the enhancement of a single machine, it makes sense that complexity in scaling up is limited.
- **Horizontal scaling:** To scale horizontally, the addition of more machines is required and, for this reason, increasing system complexity is a characteristic of this scaling strategy. With that said, the complexity involved is on a sliding scale, meaning that prudent expansion planning and internal skill set development can help manage the increasing complexity encountered when organizations do choose to horizontally scale.

Internal skill set requirements

When considering the internal skill set requirements associated with vertical scaling and horizontal scaling, there are some commonalities. For example, regardless of the approach chosen, internal skill sets required include:

- **System administration:** A solid understanding of system administration is essential for managing and configuring the infrastructure, whether it involves a single machine (vertical scaling) or a cluster of machines (horizontal scaling).
- **Automation and scripting:** Familiarity with automation tools and scripting languages (e.g., Bash, Python) is beneficial for streamlining administrative tasks, managing configurations, and orchestrating deployments in both scaling approaches.

However, there are some skill sets that are specifically required for vertical scaling or horizontal scaling.

<u>Vertical scaling skills:</u>

- Hardware familiarity and component comprehension specifically related to installing and upgrading single machines to extend capacity
- Individual server configuration and optimization

- Virtualization software and familiarity with microservices, containers etc.

Horizontal scaling skills:

- Networking and distributed systems management
- Distributed computing and load balancing
- Network traffic, bottleneck, and performance troubleshooting
- Containerization and orchestration, encompassing the management, scaling, and deployment of applications using containers.

Costs

Costs related to database scaling are somewhat complex in that both overall cost, as well as timing of costs must be considered. Specifically, vertical scaling may appeal from a cost perspective since only one machine is being enhanced and internal skill set enhancements related to management of multiple machines is not required. This means that, short term, costs are lower. However, when the single enhanced machine's capacity is met (e.g., scalability and storage), the organization will have to choose between machine replacement or scaling out (e.g., horizontal scaling) which may incur greater costs. For some, this may be preferable as capital may not be available at the time of scaling up (e.g., vertical scaling), but will be available in the future through budgeting or anticipated income.

Conversely, other organizations may find costs associated with horizontal scaling more aligned with their overall IT strategy and budget. This is because scaling out can occur one machine at a time allowing for planful expansion while avoiding an unplanned scalability limit event accompanied by a large expenditure and potential system outage.

Choosing the right scalability approach

When considering the right database scaling strategy for an organization, it's important to examine a number of factors.

Key considerations

Scalability

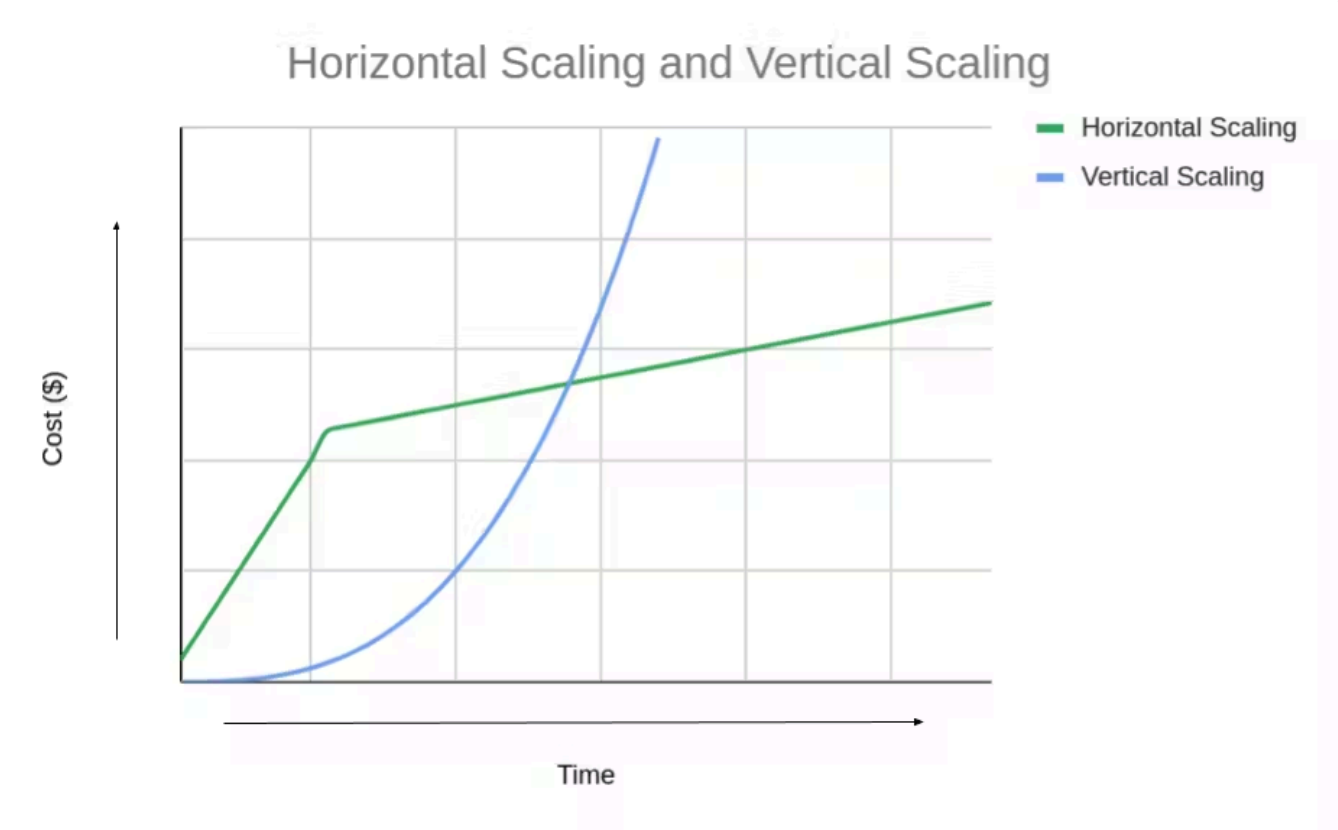
One of the first things to consider is the catalyst for database scaling. Is it due to a short-term situation expected to be resolved in the near term or is the increasing need for storage and processing expected to continue growing in the long-term? Along those same lines, is it anticipated that user demand will ebb and flow or there will be an increasing, steady-state workload? How important is planning for long-term capacity limits in relation to available budget?

Resiliency

The next item to consider is how business critical is access to the database being scaled. Are intermittent, scheduled outages acceptable or must this database be accessible at all times? Aside from taking the appropriate risk management steps to ensure restoration of the database if there is a critical hardware or software failure, how detrimental would that downtime be for users and the organization as a whole?

Costs and timing

From a budgetary standpoint, consider whether the funds available are being applied as a short-term fix or part of the organization's overall data management strategy. For example, if part of an overall expansion strategy, would it be better to pursue horizontal scaling with one additional machine with less enhancements vs scaling up an existing machine with significant enhancements for roughly the same amount? Or, is this an interim solution to bridge the organization to a broader database expansion in the future where the current equipment will become redundant?



Another consideration is budget availability over time. Is the organization more concerned with keeping costs low overall, keeping costs lower presently with the intention of spending more in the future, or perhaps a long-term, steady investment in scaling their database systems?

By considering these questions in relation to the differences between vertical vs horizontal scaling and detailed business requirements, organizations can determine the best go-forward strategy for their IT organization and business user community. To wrap up let's take a look at some recommended solutions for several priorities and budgets.

Recommendations

My application is in early stages and I need a risk-free environment for development and testing purposes.

MongoDB Atlas has a free database tier that is perfect for learning MongoDB and during your development and testing phases. While this option is not suitable for production applications due to resource limitations, it provides a risk-free way to test ideas rapidly during application development.

I don’t know how many resources I need for my application.

Creating a serverless cluster in MongoDB Atlas will seamlessly scale based on application demand so you never pay for unused resources. Serverless is ideal for apps with unpredictable demand.

My application needs high availability to ensure minimal downtime.

Database instances in MongoDB Atlas are replica sets by default. This means each database has at least two backups that can be relied upon in case of any issues with the primary node.

My database is growing over time and approaching limits of available storage.

Currently MongoDB Atlas can vertically scale to handle databases with 4TB of storage capacity. If additional storage is needed, then horizontal scaling will be needed with each shard handling a portion of the overall database.

My application has predictable highs and lows in usage due to seasonality.

Enabling cluster auto-scaling in MongoDB Atlas will ensure that available resources meet demand over time to account for seasonality, promotions, and other events that drive increased traffic. This gives you automated and reactive vertical scaling both up and down, without having to worry about setting up new servers, transferring data, or even downtime while scaling.

FAQs

What does database scaling mean?

The term *database scaling* refers to the process of changing database system capacity depending upon the amount of data or user activity occurring at any given time. Scaling can occur temporarily if you expect a sudden burst of traffic due to some ad placements or more permanently when you see a constant increase in the popularity of your services.

What are the two approaches to database scaling?

There are two approaches to database scaling: horizontal and vertical scaling. Scaling vertically means adding more hardware resources, computing power, or data storage to one machine. Meanwhile, horizontal scaling means adding more servers and/or engaging in distributed computing by adding machines and computing resources.

What are common types of horizontal scaling?

Common types of horizontal scaling include:

- **Sharding:** The sharding method of horizontal scaling involves dividing a large database into smaller, more manageable pieces called shards, and then distributing the shards across multiple machines.
- **Repetition:** The replication method of scaling horizontally creates multiple copies of the same database on multiple machines.
- **Clustering:** The clustering method of horizontal scaling involves the combination of multiple servers or machines so that they can behave as one united system.

What are the key differences between horizontal vs vertical scaling?

There are several key differences between vertical and horizontal scaling, including:

- Scalability
- System resiliency
- Complexity
- Internal skill set requirements
- Cost

Social

 GitHub

 Stack Overflow

 LinkedIn

 YouTube

X

 Twitch

 Facebook