Blog

Vertical vs. horizontal scaling explained

Discover which scaling strategy suits your business for better performance, scalability, and cost-efficiency.

September 24, 2024 | 11 min read

Alexander Patino
Content Marketing Manager

On this page

Scaling means adjusting resources by adding more capacity to existing systems or distributing workloads across additional machines to accommodate growing demands. As businesses expand, so do their data processing needs. Scaling is fundamental to ensure that IT systems handle increasing workloads while maintaining performance, reliability, and cost-efficiency.

Why is scalability important for today's businesses?

Organizations create large amounts of data, and their applications need to handle fluctuating, often unpredictable workloads. Scalability lets businesses meet current demands while preparing for future growth. A scalable infrastructure improves business continuity, reduces downtime, and helps companies stay competitive by adapting to changing market conditions and customer needs.

The two types of scalability — vertical scaling and horizontal scaling — and how they're different

Vertical and horizontal scaling are the two primary methods used to help IT infrastructure handle growth.

- **Vertical scaling (scaling up)** means adding more capacity to an existing server, such as CPU, RAM, storage, or a combination. It's a straightforward approach that boosts the capacity of a single machine, but it has physical limitations.

- **Horizontal scaling (scaling out)**, on the other hand, means adding machines to distribute workloads across multiple systems. This can boost scalability significantly and provide backup in case a server goes down. This approach is more complex, but it's also more flexible, redundant, and resilient.

Choosing between vertical and horizontal scaling or using a combination of both depends on the business' specific needs, including performance requirements, cost constraints, and future growth projections.

Understanding vertical scaling

What does vertical scaling mean, specifically in database and application management?

Vertical scaling, or "scaling up," makes an existing system more powerful by adding CPU, RAM, and storage resources. It lets a server do more work by upgrading its internal components. Vertical scaling is particularly common in on-premises infrastructure or environments where organizations have more control over hardware upgrades. It can boost application performance by increasing the capacity of an existing machine for more processing (e.g. for compression or encryption), memory (for storing more data in-memory for access), or storage (for more I/O). In database management, vertical scaling helps handle larger datasets or more complex queries by adding memory or computational power to an existing server. It's easy to implement because it requires no major architectural changes.

How does vertical scaling address performance and scalability challenges?

Vertical scaling improves application performance by making an existing server more powerful.

What are the limitations and benefits of vertical scaling?

Benefits:

- **Simplicity:** Minimal system changes make implementation easier and quicker, with fewer operational disruptions.

- **Immediate performance boost:** Adding CPU, memory, or storage generally improves performance without needing more servers.

- **Cost-effective for smaller clusters:** Best suited for when workloads can be managed by few machines.

Limitations:

- **Hardware limits:** Physical constraints on hardware upgrades can lead to bottlenecks.
- **Requires downtime**: To vertically scale a server, it typically needs to be taken offline, so downtime would need to be planned accordingly.

While vertical scaling is a quick way to boost performance, it's often a short-to-medium-term strategy. Vertical scaling is enough for businesses with moderate growth, but horizontal scaling works better for larger enterprises with growing data needs.

Understanding horizontal scaling

How is horizontal scaling different from vertical scaling?

Horizontal scaling, or "scaling out," adds more machines to a system, distributing workloads across multiple servers or nodes. Unlike vertical scaling, which gives a machine more resources, horizontal scaling gives you a number of machines working together, which works better for larger or fluctuating workloads.

While vertical scaling is limited by the amount of hardware you can afford to stuff in a box, horizontal scaling lets systems grow much larger. It’s commonly used in cloud environments where multiple servers work together to balance loads and manage data, improving uptime and performance.

What advantages does horizontal scaling offer for data management and application performance?

Horizontal scaling provides several advantages:

- **Higher scalability:** By adding more machines, businesses handle growing or unpredictable workloads more easily.
- **Fault tolerance and redundancy:** Distributing workloads across multiple machines makes systems more reliable because it reduces the impact of a node failure.

- **Improved performance through load distribution:** Workloads are spread across multiple servers, reducing bottlenecks and improving performance, especially in data-intensive applications.
- **Cost efficiency at scale:** In cloud environments, horizontal scaling lets resources be allocated as they're needed, using auto-scaling.

What challenges does horizontal scaling introduce, and how do you fix them?

Horizontal scaling introduces several challenges:

- **Increased complexity:** Managing multiple servers can be difficult, but tools like load balancers and orchestration platforms like Kubernetes can help.
- **Data consistency:** Maintaining consistency across nodes is challenging, especially in distributed systems. Techniques such as distributed databases and replication help keep data synchronized.
- **Network latency:** More inter-machine communication can lead to network delays, especially across geographic locations. Solutions such as sharding and edge computing can reduce latency by processing data closer to where it's generated.
- **Cost management:** Scaling out, when done incorrectly, can become expensive, but autoscaling and monitoring tools can help businesses dynamically scale and optimize resource usage.

Horizontal scaling is often a more sustainable, long-term solution for organizations to deal with growth, large amounts of data, or availability requirements. It's flexible and resilient for today's complex jobs.

Vertical vs. horizontal scaling

What are the technical differences between vertical and horizontal scaling?

The main difference between vertical and horizontal scaling is how they make a system more powerful:

- **Vertical scaling** improves a server's performance by adding more resources (CPU, RAM, and storage).

- **Horizontal scaling** adds more machines or nodes to distribute workloads across systems, increasing computing power and redundancy. Consequently, it allows for load balancing.

In short, vertical scaling suits predictable workloads on a server, while horizontal scaling is better for dynamic, growing environments.

How do these scaling strategies affect application and database performance?

Application performance:

- **Vertical scaling:** Preferred for applications needing high computational power or memory, allowing for immediate resource boosts.
- **Horizontal scaling:** Works well for distributed workloads, such as spreading tasks across multiple nodes, which improves performance where it's needed.

Database performance:

- **Vertical scaling:** Upgrading storage, memory, or processing power improves performance for predictable read/write operations, but scaling limits may cause bottlenecks in larger datasets.
- **Horizontal scaling:** Distributes data across multiple nodes, using techniques such as sharding and replication to increase capacity and fault tolerance. This reduces potential downtime and improves performance with large datasets.

In what use cases is one scaling strategy preferred over the other?

Vertical scaling is preferred when:

- The application has predictable growth and resource needs
- Simplicity is crucial, and managing multiple servers is unnecessary
- The workload fits within a server's capacity, and hardware upgrades are enough
- High availability isn't an issue

Horizontal scaling is preferred when:

- The system must handle rapidly increasing or changing workloads

- High availability and fault tolerance are required (for example, in e-commerce, social media, and cloud applications)
- The system needs to handle a large number of concurrent users
- Large and growing datasets need distributed databases for efficiency
- Long-term scalability is a priority, as hardware ceilings don't limit horizontal scaling

Often, a hybrid approach is used, with businesses starting with vertical scaling for simplicity and transitioning to horizontal scaling as demands grow.

Examples: Vertical and horizontal scaling

How have real-world businesses implemented vertical scaling?

Vertical scaling, or "scaling up," is often used by businesses with predictable workloads. One example is financial services, which require low latency and high performance. Financial institutions often scale vertically to boost the computational power of existing infrastructure, improving transaction processing without major architectural changes.

For instance, a large bank might upgrade its database servers by adding CPU power and memory to handle more transactions during peak times. This lets the bank maintain performance without adding servers for operations such as stock trades and fund transfers.

How have real-world businesses implemented horizontal scaling?

Horizontal scaling, or "scaling out," is favored by businesses that need reliability. Criteo, an advertising content company, is one example. To manage its global user base, Criteo shifted to a microservices architecture and distributed workloads across thousands of servers. This prevents any single server from overloading, resulting in smoother streaming and less downtime, even when busy.

What lessons can be learned from these case studies in scaling strategies?

Takeaways from these examples include:

- **Match scaling strategy to business needs:** Evaluate workloads, traffic, and growth to choose the right strategy. Vertical scaling suits predictable demands, while horizontal scaling is better for rapid growth or dynamic traffic.

- **Plan for long-term scalability:** Vertical scaling offers immediate benefits but has limits. Businesses expecting growth, like Criteo, should plan for horizontal scaling early on to support future demand.

- **Invest in automation tools:** Managing horizontal scaling can be complex. Automation tools such as AWS Auto Scaling and tools like Aerospike Kubernetes Operator streamline the process and make resource management easier.

- **Consider redundancy and fault tolerance:** Horizontal scaling's redundancy helps prevent downtime, as seen with Criteo, where regional server outages are less likely to affect other areas.

Successful businesses often adopt a hybrid approach, starting with vertical scaling and gradually incorporating horizontal scaling as they grow.

Aerospike's approach to scaling

How does the Aerospike Database support vertical and horizontal scaling?

Aerospike Database supports both vertical and horizontal scaling.

- **Vertical scaling:** Aerospike Database optimizes server resources such as CPU, RAM, and SSDs to boost database performance. With off-the-shelf hardware, it takes advantage of vertical scaling for businesses needing immediate throughput improvements without restructuring infrastructure.

- **Horizontal scaling:** Aerospike's distributed architecture lets businesses add more nodes and automatically distributes data across the cluster. This makes performance more consistent and the cluster more available, which is important for real-time applications. As businesses grow, they can add nodes to handle increased workloads.

What makes Aerospike's data management solutions best suited for scalability challenges?

Aerospike handles both vertical and horizontal scalability needs:

- **Hybrid Memory Architecture:** By placing only indexes in DRAM and persisting data to SSDs but reading/writing at in-memory speeds, Aerospike optimizes performance while reducing costs. This approach more easily enables vertical scaling, offering storage efficiency without sacrificing speed.

- **Smart Client technology:** Aerospike's Smart Client balances loads in horizontally scaled systems, reducing bottlenecks and improving latency. This helps ensure that adding nodes enhances performance rather than slowing the system.

- **Strong consistency model:** Aerospike maintains real-time data consistency across all nodes, making it more reliable for applications that cannot tolerate discrepancies.

- **High availability and fault tolerance:** Aerospike uses replication and automatic data redistribution when nodes are added or removed, reducing downtime and providing uninterrupted service for large-scale, always-on applications.

- **Cloud and on-premises integration:** Aerospike supports both cloud and on-premises deployments, offering auto-scaling in environments such as AWS or Google Cloud. This lets businesses adjust resources based on demand.

How do Aerospike's technologies save money when scaling?

Aerospike maximizes performance and cost efficiency, helping businesses scale more effectively for less money:

- **Cost-effective scaling with SSDs:** Aerospike's Hybrid Memory Architecture uses SSDs instead of relying on expensive DRAM, reducing the cost of vertical scaling while maintaining high performance for real-time applications.

- **Efficient resource utilization:** The Smart Client uses cluster resources efficiently by balancing the load and directing traffic to the most appropriate nodes. This reduces waste and improves throughput, helping businesses control costs when scaling horizontally.

- **Elastic scaling in cloud environments:** Aerospike integrates with tools such as Kubernetes and AWS Auto Scaling to allow dynamic resource allocation based on demand, meaning businesses pay only for the resources they use.

Aerospike's approach is designed to meet the needs of today's data-intensive applications. It offers both vertical scaling for immediate performance and horizontal scaling for long-term growth, resulting in speed, reliability, and cost efficiency at scale.

Vertical vs. horizontal scaling takeaways

Choosing the right scaling strategy is vital for long-term business success. Whether using vertical, horizontal, or a combination of the two, the right approach lets businesses meet current demands and prepare for growth.

Without a proper scaling strategy, businesses risk performance bottlenecks and downtime. However, A well-executed scaling plan maintains service quality as workloads grow, helping businesses meet customer expectations and take advantage of market opportunities.

Aerospike provides the performance and flexibility businesses need to compete in a fast-paced, data-driven world. By understanding vertical and horizontal scaling and using Aerospike's solutions, companies can gauge their infrastructure for both present and future needs.

Takeaways:

- Vertical scaling offers quick performance boosts but is limited by hardware constraints.
- Horizontal scaling provides higher growth potential, fault tolerance, and flexibility but requires more complex management.
- Aerospike's solutions integrate both scaling strategies for efficient, fast, and cost-effective scaling.

The right scaling strategy improves business continuity, reduces resources, and prepares for growth.

Database Cloud Managed Service Cost efficiency Developer Architect Architecture Application development Database management