phoenixnap.com

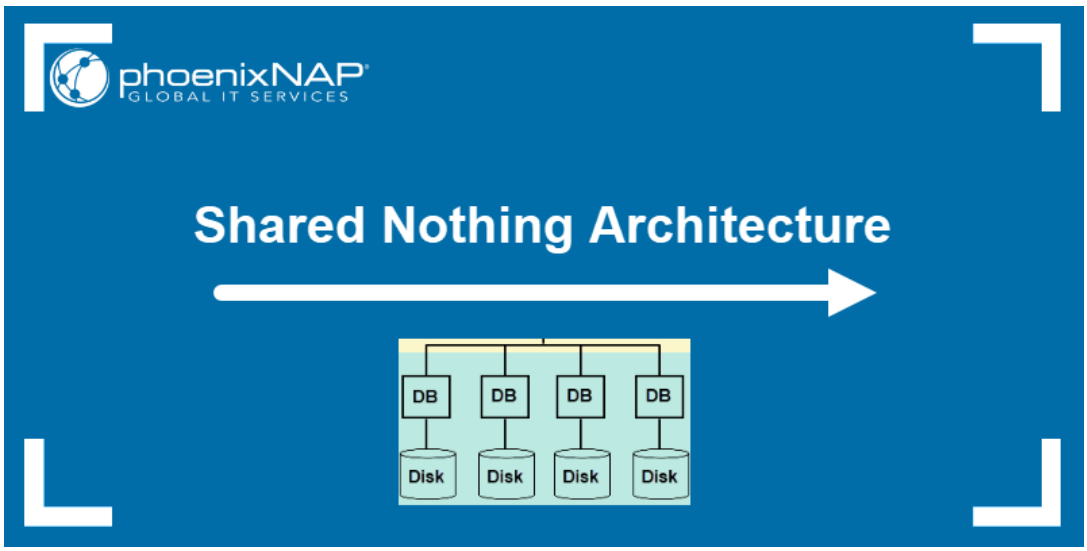
Shared Nothing Architecture Explained {Diagram, Pros & Cons}

6-7 minutes

Introduction

Why are companies such as Google and Facebook using the Shared Nothing Architecture, and how does it differ from other models?

Read on to learn what Shared Nothing is, how it compares to other architectures, and its advantages and disadvantages.



Shared Nothing Architecture (SNA) is a distributed computing architecture that consists of multiple separated nodes that don't share resources. The nodes are independent and self-sufficient as they have their own disk space and memory.

In such a system, the data set/workload is split into smaller sets (nodes) distributed into different parts of the system. Each node

has its own memory, storage, and independent input/output interfaces. It communicates and synchronizes with other nodes through a high-speed interconnect network. Such a connection ensures low latency, high bandwidth, as well as high availability (with a backup interconnect available in case the primary fails).

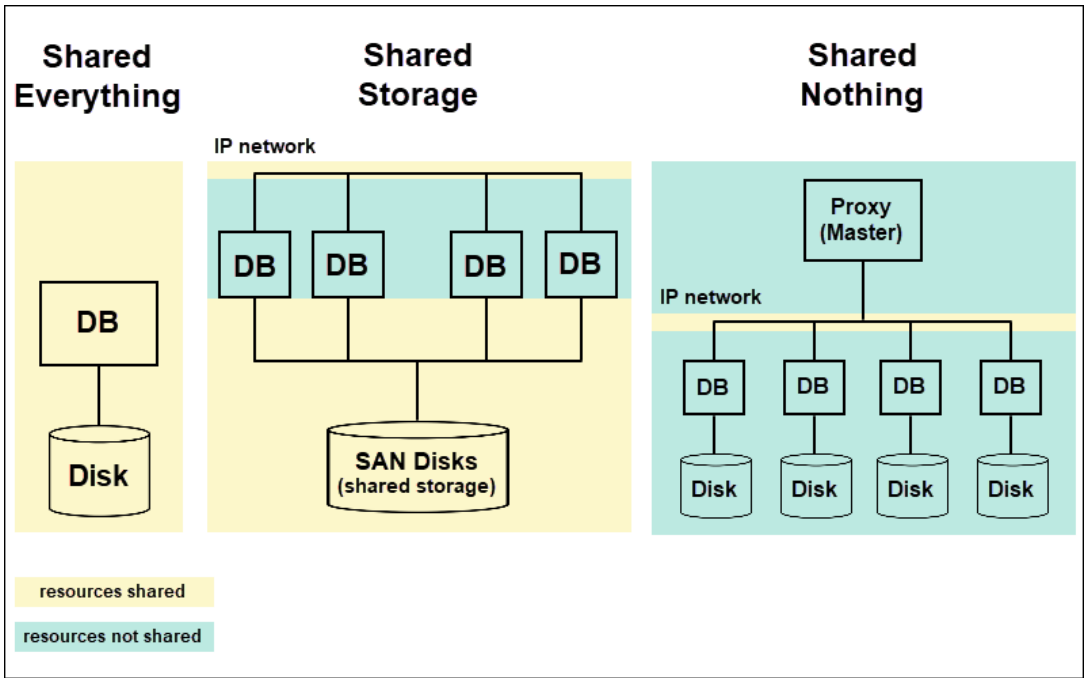
Since data is horizontally partitioned, the system supports incremental growth. You can add new nodes to scale the distributed system horizontally and increase the transmission capacity.

Shared Nothing Architecture Diagram

The best way to understand the architecture of the shared-nothing model is to see it side by side with other types of architectures.

Below you see the difference in shared vs. non-shared components in different models - **Shared Everything**, **Shared Storage**, and **Shared Nothing**.

Unlike the others, SNA has no shared resources. The only thing connecting the nodes is the network layer, which manages the system and communication among nodes.



Other Shared Architecture Types Explained

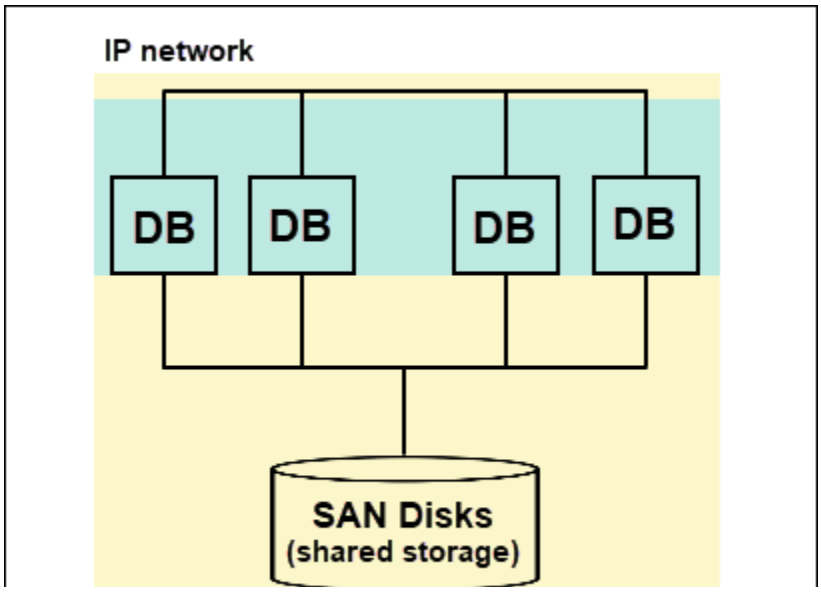
The concept of “shared nothing” was first introduced by Michael Stonebraker in his 1986 research paper, in which he contrasted shared disk and shared memory architecture. While comparing these two options, Stonebraker included the possibility of creating a system in which neither memory nor storage is shared.

When deciding whether SNA is the solution for your use case, it is best to compare it with other cluster types. Alternative options include:

- Shared Disk Architecture
- Shared Memory Architecture
- Shared Everything Architecture

Shared-Disk Architecture

Shared disk is a distributed computing architecture in which all the nodes in the system are linked to the same disk device but have their own private memory. The shared data is accessible from all cluster nodes and usually represents a shared disk (such as a database) or a shared filesystem (like a storage area network or network-attached storage). The shared disk architecture is best for use cases in which data partitioning isn’t an option. Compared to SNA, it is far less scalable.

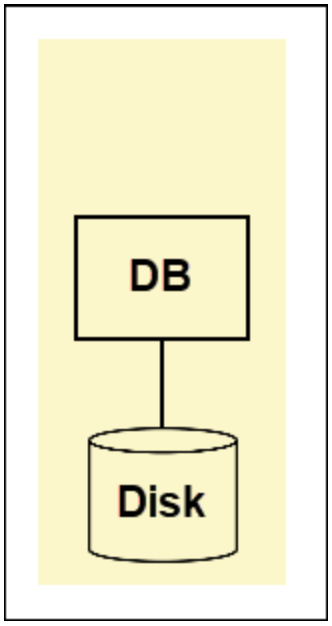


Shared-Memory Architecture

Shared memory is an architectural model in which nodes within the system use one shared memory resource. This setup offers simplicity and load balancing as it includes point-to-point connections between devices and the main memory. Fast and efficient communication among processors is key to ensure efficient transmission of data and to avoid redundancy. Such communication is carried out through an interconnection network and managed by a single operating system.

Shared-Everything Architecture

On the opposite side of the spectrum, there is the shared everything architecture. This architectural model consists of nodes that share all resources within the system. Each node has access to the same computing resources and shared storage. The main idea behind such a system is maximizing resource utilization. The disadvantage is that shared resources also lead to reduced performance due to contention.



Advantages and Disadvantages of Shared Nothing Architecture

When compared to different shared architectures mentioned above, it is clear the Shared Nothing Architecture comes with many benefits. Take a look at some of the advantages, as well as disadvantages of such a model.

Advantages

There are many advantages of SNA, the main ones being scalability, fault tolerance, and less downtime.

Easier to Scale

There is no limit when it comes to scaling in the shared-nothing model. Unlimited scalability is one of the best features of this type of architecture. Since nodes are independent and don't share resources, scaling up your application won't disrupt the entire system or lead to resource contention.

Eliminates Single Points of Failure

If one of the nodes in the application fails, it doesn't affect the functionality of others as each node is self-reliant. Although node failure can impact performance, it doesn't disrupt the overall behavior of the app as a whole.

Simplifies Upgrades and Prevents Downtime

There is no need to shut down the system while working on or upgrading individual nodes. Thanks to redundancy, upgrading one node at a time doesn't impact the effectiveness of others. What's more, having redundant copies of data on different nodes prevents unexpected downtime caused by disk failure or data loss.

Disadvantages

Once you considered the benefits of SNA, take a look at a couple of disadvantages that can help you decide whether it is the best option for you.

Cost

A node consists of its individual processor, memory, and disk. Having dedicated resources essentially means higher costs when it comes to setting up the system. Additionally, transmitting data that requires software interaction is more expensive compared to architectures with shared disk space and/or memory.

Decreased Performance

Scaling up your system can eventually affect the overall performance if the cross-communication layer isn’t set up correctly.

Conclusion

After reading this article, you should have a better understanding of Shared Nothing Architecture and how it works. Take into account all the advantages and disadvantages of the model before deciding on the architecture for your application.