What Is a Distributed Database?

May 6, 2021

DATABASE

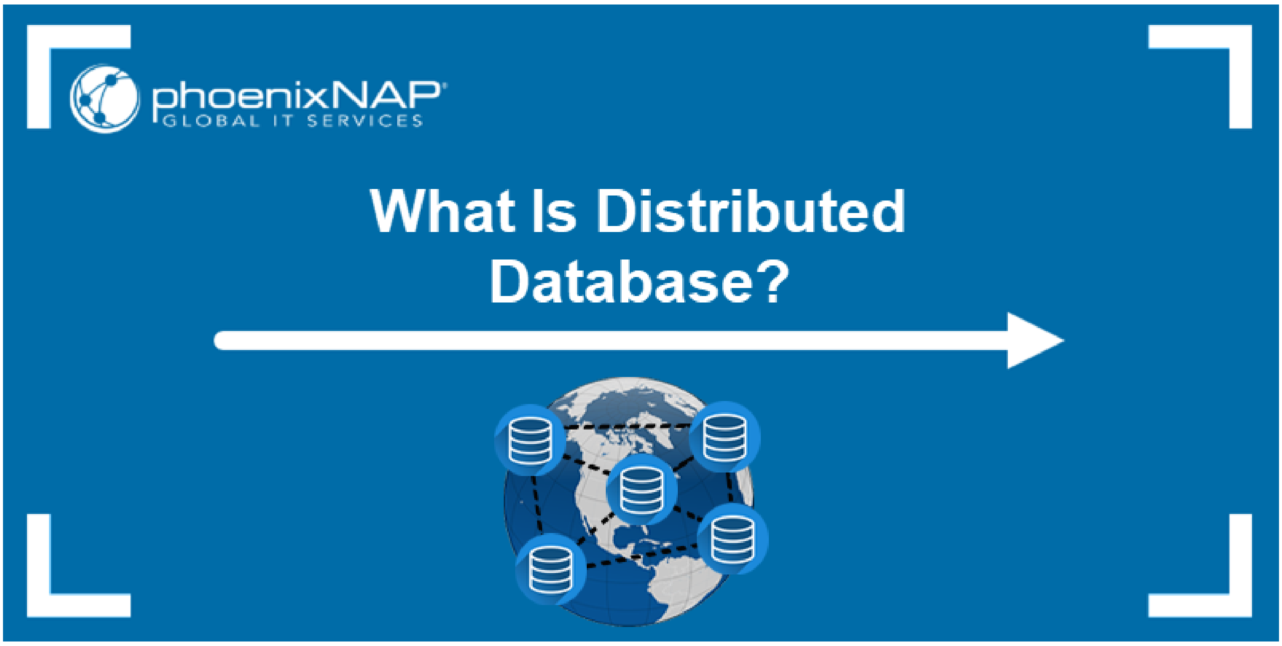
Home » Databases » What Is a Distributed Database?

Introduction

Distributed databases are used for horizontal scaling, and they are designed to meet the workload requirements without having to make changes in the database application or vertically scale a single machine.

Distributed databases **resolve various issues**, such as availability, fault tolerance, throughput, latency, scalability, and many other problems that can arise from using a single machine and a single database.

In this article, you'll learn what distributed databases are and their advantages and disadvantages.



Distributed Database Definition

A distributed database represents multiple interconnected databases spread out across several sites connected by a network. Since the databases are all connected, they appear as a single database to the users.

Distributed databases utilize multiple nodes. They scale horizontally and develop a distributed system. More nodes in the system provide more computing power, offer greater availability, and resolve the single point of failure issue.

Different parts of the distributed database are stored in **several physical locations**, and the processing requirements are distributed among processors on multiple database nodes.

A centralized distributed database management system (**DDBMS**) manages the distributed data as if it were stored in one physical location. DDBMS synchronizes all data operations among databases and ensures that the updates in one database automatically reflect on databases in other sites.

Distributed Database Features

Some general features of distributed databases are:

- **Location independency** - Data is physically stored at multiple sites and managed by an independent DDBMS.
- **Distributed query processing** - Distributed databases answer queries in a distributed environment that manages data at multiple sites. High-level queries are transformed into a query execution plan for simpler management.

- **Distributed transaction management** - Provides a consistent distributed database through commit protocols, distributed concurrency control techniques, and distributed recovery methods in case of many transactions and failures.
- **Seamless integration** - Databases in a collection usually represent a single logical database, and they are interconnected.
- **Network linking** - All databases in a collection are linked by a network and communicate with each other.
- **Transaction processing** - Distributed databases incorporate transaction processing, which is a program including a collection of one or more database operations. Transaction processing is an atomic process that is either entirely executed or not at all.

Distributed Database Types

There are two types of distributed databases:

- **Homogenous**
- **Heterogenous**

Homogeneous

A homogenous distributed database is a network of **identical databases** stored on multiple sites. The sites have the same operating system, DDBMS, and data structure, making them easily manageable.

Homogenous databases allow users to access data from each of the databases seamlessly.

The following diagram shows an example of a homogeneous database:

Heterogeneous

A heterogeneous distributed database uses **different** schemas, operating systems, DDBMS, and different data models.

In the case of a heterogeneous distributed database, a particular site can be completely unaware of other sites causing limited cooperation in processing user requests. The limitation is why translations are required to establish communication between sites.

The following diagram shows an example of a heterogeneous database:

Distributed Database Storage

Distributed database storage is managed in two ways:

- **Replication**
- Fragmentation

Replication

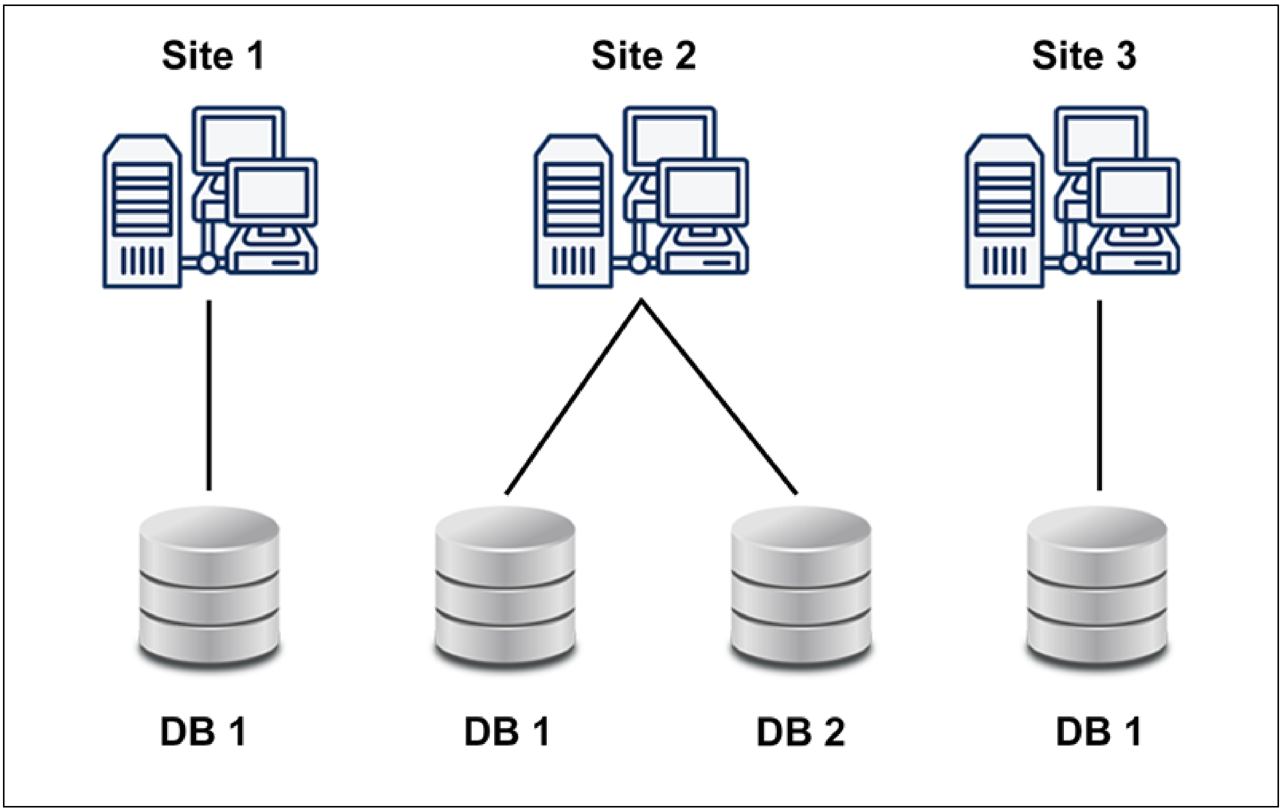
In database replication, the systems store **copies of data on different sites**. If an entire database is available on multiple sites, it is a fully redundant database.

The advantage of database replication is that it **increases data availability o**n different sites and allows for parallel query requests to be processed.

Note: Learn how replication works in our article that compares Replication and Backup.

However, database replication means that data requires constant updates and synchronization with other sites to maintain an exact database copy. Any changes made on one site must be recorded on other sites, or else inconsistencies occur.

Constant updates cause a lot of server overhead and complicate concurrency control, as a lot of concurrent queries must be checked in all available sites.



Note: Read our tutorial to learn how to set up MySQL Master Slave replication.

Fragmentation

When it comes to fragmentation of distributed database storage, the relations are fragmented, which means they are **split into smaller parts**. Each of the fragments is stored on a different site, where it is required.

The prerequisite for fragmentation is to make sure that the fragments can later be reconstructed into the original relation without losing data.

The advantage of fragmentation is that there are **no data copies**, which prevents data inconsistency.

There are two types of fragmentation:

- **Horizontal fragmentation** - The relation schema is fragmented into groups of rows, and each group (tuple) is assigned to one fragment.
- **Vertical fragmentation** - The relation schema is fragmented into smaller schemas, and each fragment contains a common candidate key to guarantee a lossless join.

Note: In some cases, a mix of fragmentation and replication is possible.

Distributed Database Advantages and Disadvantages

Below are some key advantages and disadvantages of distributed databases:

Advantages	Disadvantages
Modular development	Costly software
Reliability	Large overhead
Lower communication costs	Data integrity
Better response	Improper data distribution

The advantages and disadvantages are explained in detail in the following sections.

Advantages

- **Modular Development**. Modular development of a distributed database implies that a system can be expanded to new locations or units by adding new servers and data to the existing setup and connecting them to the distributed system without interruption. This type of expansion causes no interruptions in the functioning of distributed databases.

- **Reliability**. Distributed databases offer greater reliability in contrast to centralized databases. In case of a database failure in a centralized database, the system comes to a complete stop. In a distributed database, the system functions even when failures occur, only delivering reduced performance until the issue is resolved.

- **Lower Communication Cost**. Locally storing data reduces communication costs for data manipulation in distributed databases. Local data storage is not possible in centralized databases.

- **Better Response**. Efficient data distribution in a distributed database system provides a faster response when user requests are met locally. In centralized databases, user requests pass through the central machine, which processes all requests. The result is an increase in response time, especially with a lot of queries.

Disadvantages

- **Costly Software**. Ensuring data transparency and coordination across multiple sites often requires using expensive software in a distributed database system.

- **Large Overhead**. Many operations on multiple sites requires numerous calculations and constant synchronization when database replication is used, causing a lot of processing overhead.

- **Data Integrity**. A possible issue when using database replication is data integrity, which is compromised by updating data at multiple sites.

- **Improper Data Distribution**. Responsiveness to user requests largely depends on proper data distribution. That means responsiveness can be reduced if data is not correctly distributed across multiple sites.

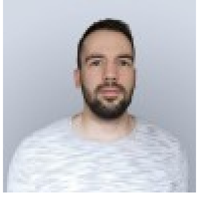
Note: Consider using a Multi-Model Database. Multi-model databases provide a singular engine for various database types.

Conclusion

You now know what distributed databases are and how they operate.

Distributed databases come with considerable benefits compared to centralized databases. After reading this article, you should be able to pick the right database type for you.

Was this article helpful? Yes No



Bosko Marijan

Having worked as an educator and content writer, combined with his lifelong passion for all things high-tech, Bosko strives to simplify intricate concepts and make them userfriendly. That has led him to technical writing at PhoenixNAP, where he continues his mission of spreading knowledge.

Next you should read

Databases

How to Create Database and Collection in MongoDB

April 29, 2020

Create new MongoDB databases and add data collections by using the commands presented in this article. The use, insert and find commands are crucial tools that allow you to perform basic administrative tasks on your database...

READ MORE

MySQL

How To Show a List of All Databases in MySQL

October 13, 2022

With Structured Query Language (SQL), you can easily access and manage content in all your databases. This guide will show you how to list all MySQL Databases.

READ MORE

MySQL, SysAdmin

How to Improve MySQL Performance With Tuning

January 15, 2020

The performance of MySQL databases is an essential factor in the optimal operation of your server. Make sure you are avoiding the common pitfalls concerning MySQL queries and system setup.

READ MORE

Backup and Recovery, MySQL, SysAdmin

How To Set Up MySQL Master Slave Replication

May 29, 2019

In cloud computing, master-slave data replication refers to storing the same information on multiple servers. One server controls the group, and the other devices handle the work within the same node. This guide will walk you...

READ MORE

Live Chat Get a Quote Support | 1-855-330-1509 Sales | 1-877-588-5918

Privacy Center Do not sell or share my personal information

Contact Us
Legal
Privacy Policy
Terms of Use
DMCA
GDPR
Sitemap