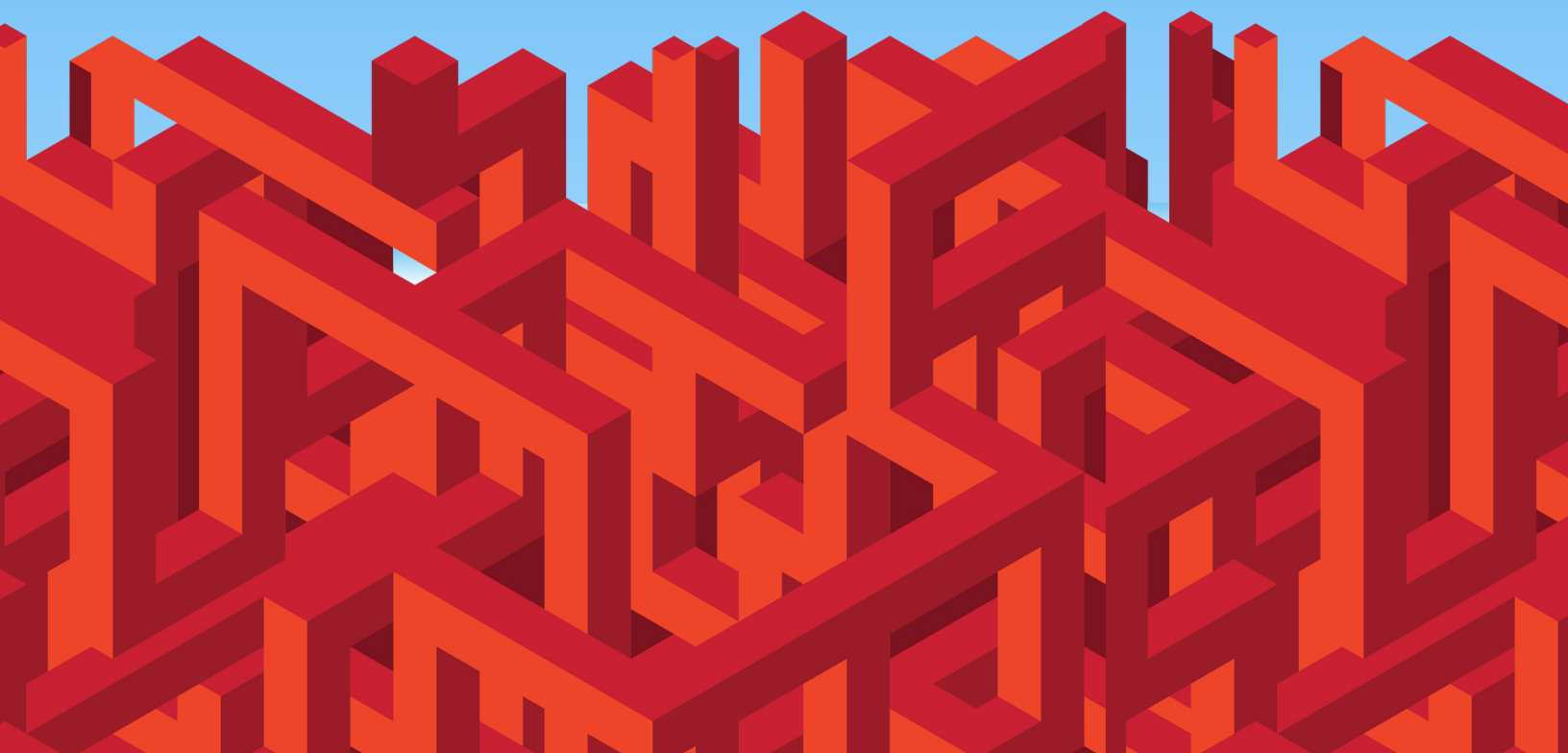
PERCONA

6 Common Causes of Poor Database Performance

. . . and what to do about them



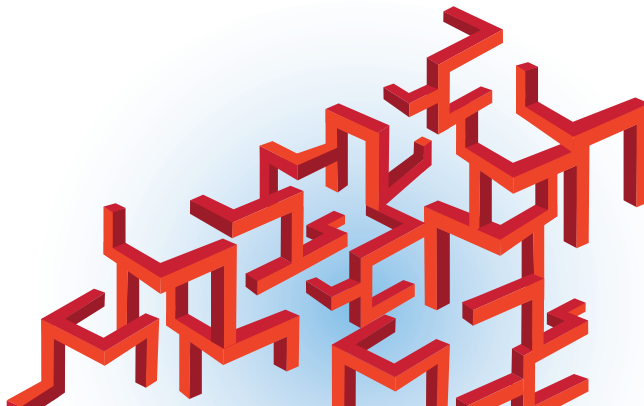
6 Common Causes of Poor Database Performance

. . . and what to do about them

Your company relies on databases to succeed in business. Whether you provide gaming services, software as a service, or an e-commerce site, you need a fully optimized, highly available, and superbly performant database to ensure your applications meet customer expectations.

When even the slightest slowdown can frustrate users, damage your brand, and lead to revenue loss, you must be able to quickly identify the true causes of database performance problems and fix them fast.

Below are the top six causes of poor database performance that you should look out for.



1 : Inefficient Database Design

How you design your database environment impacts everything you do down the line. Poor design decisions cascade throughout the life of your applications, potentially leading to serious problems with performance and scalability. The cost to you and your team will be high as the volume of complaints grows and you find yourself applying valuable resources to retroactively fix early flaws.

Why is good design so important? Because in complex database infrastructures, everything is connected. Applications, which once comprised a single stack, now encompass many diverse services. Databases work within large ecosystems of components. And infrastructures stretch across localities, countries, and continents, operating across different environment types, including cloud, on-premises, and hybrid.

With all these interdependencies, it's important to ask good questions as you consider the design of your database infrastructure. Here are a few:

Capacity. Understanding how data flows through your architecture helps you create the most effective environment for your applications. How much data will you store and where? How quickly will it need to be accessed? How many concurrent users do you anticipate for various applications?

Workload. To optimize your infrastructure, you need to forecast workload. Will it be high-read, high-write, or both? Will it be steady, or will it change based on specific events? Does your database engine match your workload type?

Data types. The more data you store, the less you can fit into memory, so your data types must be as efficient as possible. Will they increase the footprint of the data they're storing? Will they make processing and handling data more efficient?

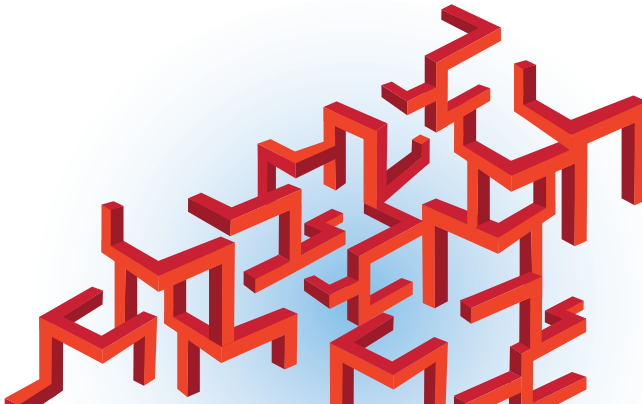
Software ecosystem. Databases don't exist in isolation. What other software will you need along with your database? How will you integrate with third-party plugins? Will data be fed from third-party applications?

Cloud. Although cloud vendors manage operational automation and day-to-day tasks, you still own ensuring the performance and availability of your database. How will you manage and monitor queries? How will you handle disaster and recovery? How will you ensure your database is secure?

Scalability. As data grows, your database slows down. How will you scale as data grows? Can you partition or "shard" your database? Can you create database clusters?

Limitations. Infrastructure teams support multiple application stacks. Do you have the power to pick optimal configurations, or is that dictated by your company or other teams?

Support. Think about what will be required of your team. Do you have the internal knowledge to manage your database? Will you need external help to ensure all your components work together now and in the future?



2 Errors in Deployment

Even a well-designed, well-implemented system must be deployed correctly. A lot of deployment issues are trivial: using the wrong hardware or the wrong instance type in the cloud, using spinning disks instead of flash-based storage, or using an inappropriate amount of memory. Typically, these types of issues are caught quickly and are easy to resolve. But some are not; for example, database change errors, which can cause serious problems in production and are time-consuming to diagnose.

Thorough testing helps prevent deployment issues, but problems can still be missed. Before you go to production, deploy your database in a test environment first and make sure you have a backup of your data. Also, implement version control of your database and schemas so you can roll back if something goes wrong and easily trace who did what and when. Traceability allows you to more quickly isolate and fix issues.

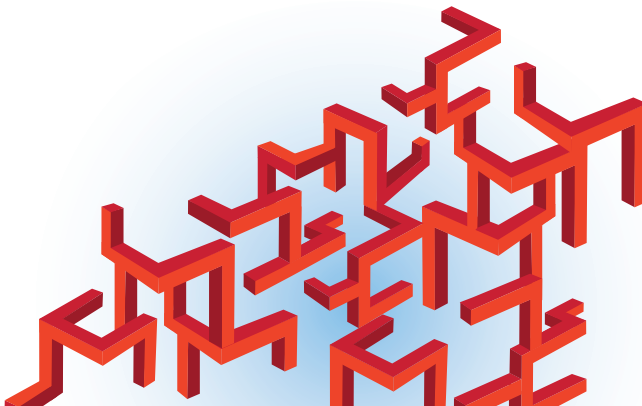
More than the database, network-related issues are usually the biggest cause of deployment related problems. Even in the same data center, adding multiple switches or slow firewalls between the application server and the database can multiply latency.

Finally, if you place the database server and application/web server in different availability zones or data centers, the latency increase gets even more dramatic. As you decide where your data will sit, keep in mind that data needs to be near running applications to minimize latency and provide a highly performant user experience.

3 Software Misconfigurations

The configuration of software components is important to database performance. Just one misconfiguration can not only slow your applications, but cause downtime, data corruption, data loss, and security breaches.

From the application standpoint, a misconfigured driver is a common database-related issue. Configuring a Java connection pool to be too small, for example, can cause low performance and application errors. Setting it too large can lead to connection errors, poor performance, or even crashes. Setting the wrong isolation mode in the connection settings without understanding the implications, can cause the application to work incorrectly and potentially cause data corruption.



4 Over Indexing

The difference between your database being fast, responsive, and scaling properly often depends on how effectively you use indexes. The big advantage of indexes is that they permit queries, updates, and deletes to run as fast as possible. However, over indexing can lead to massive slowdowns in overall performance.

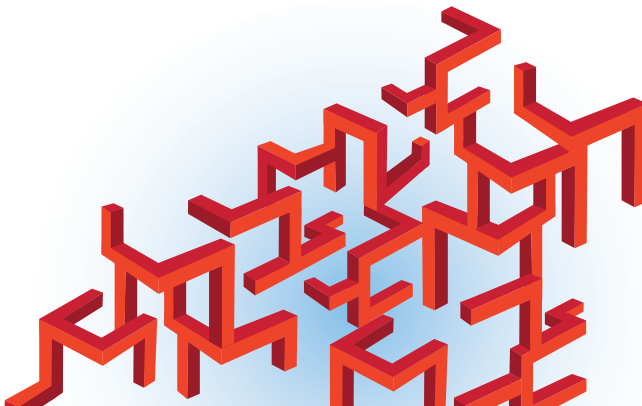
To be really effective at improving performance, indexes should be as small as possible. In addition, the number of indexes you need depends on your application workload. Some developers create a lot of indexes without a specific reason or to test a query, but then forget to drop them, which, in some cases, can make the size of the indexes larger than the data itself. The general rule is to create all the indexes your application really needs for solving the most frequent queries—not one more or one less.

Creating indexes for solving queries is a good habit, but you need to use them wisely. Consider maintaining your indexes from time to time, dropping all duplicates and unused indexes that may slow down performance.

5 Resource Saturation

Database systems have only so much memory, CPU, disk IO, and network resources, and can't do more than their constraints allow. Optimizing your software configuration, schema, and queries places less demand on these types of resources. For example, a web application developer may write code that involves several separate queries instead of batching them, which unnecessarily increases load on the CPU and memory.

Designing for efficient resource usage is important, but so is having a way to continually monitor that usage so you can address problems as they arise. Consider using a monitoring tool that provides features that allow you to analyze your database queries and identify and fix issues as quickly as possible. It should also provide well-designed dashboards that give you easy insight into system metrics, such as CPU, network, memory, and disk performance.



6 : Database Misconfigurations

When users start complaining about application performance, how do you know whether it's the database or something else causing performance degradation? Sometimes finding the answer is simple. For example, you check the database process list to see which application queries have been running for more than 30 seconds. But in other cases, it might not be so obvious, like a misconfigured buffer pool or log file size.

Here again having a tool that gives you deep insight into what's happening in your database system allows you to determine whether the problem is with the database, and if so, identify what it is. This should be a database-focused tool for in-depth troubleshooting and investigation so you can fully understand the health of your infrastructure.

It should include capabilities for monitoring the state of your systems based on predefined sets of metrics. It should also provide observability, which gives you deep visibility into the internal states of your systems, helping you actively discover the "unknown, unknowns."

Finally, a database brought down by a cybersecurity attack due to misconfigurations does not perform at all. Make sure you have a tool that helps you quickly find and fix common data security issues across all of your open source databases. It should provide daily security checks for common issues, alerts for when non-compliance is found, and the ability to audit your security check history.

How Percona can help

Percona can help you determine whether your database issues are due to one of the reasons above or something else entirely. We'll assess your current deployment and provide a detailed analysis and plan for optimal performance, which you can implement yourself or with our help.

LEARN MORE about Percona Database Consulting.

Want to get started right away? Contact us at **sales@percona.com.**

About Percona

Percona is widely recognized as a world-class open source database software, support, and services company. The organization is dedicated to helping businesses make databases and applications run better through a unique combination of expertise and open source software. Percona works with numerous global brands across many industries creating a unified experience for monitoring, managing, securing, and optimizing database environments on any infrastructure.

Percona equips organizations with the freedom to choose, the freedom to create, and the freedom to make a difference - helping them scale and innovate with speed as they grow.

For more information, visit **percona.com**.

Databases run better with Percona.

For more information please contact us at:
+1-888-316-9775 (USA), +44 203 608 6727 (Europe),
or via email at sales@percona.com

170123