PERCONA

Top 5 Causes of Poor Database Performance

Your company relies on its database to do business. Whether you provide gaming services, SaaS or ecommerce services, you need a functioning database to achieve optimal application performance.

Identifying and understanding the true cause of database performance problems allows for a quicker and more efficient resolution. And contrary to popular belief, the problem is not always the database itself!

Below are the top five reasons for poor database performance:

Cause #1: Poor Database Design

Design-based performance issues are a worst-case scenario, because they are very expensive to change. Is your environment workload high-read, high-write, or both? Is your workload steady, or does it change based on specific events? What type of database engine are you using? Does it match your workload type? These are just some of the questions that should be asking (and answering) when designing a system for your data.

Cause #2: Poor Deployment

Even a well-designed and implemented system must be deployed correctly. A lot of deployment issues are trivial: using the wrong hardware, the wrong instance type in the cloud, an inappropriate amount of memory or spinning disks instead of flash-based storage. These types of issues are all typically caught quickly and easy to resolve.

Network-related issues are usually the biggest cause of most deployment related issues. Even in the same data center, adding multiple switches or slow firewalls between the application server and the database can multiply latency. If you place the database server and application/web server in different availability zones (or different data centers) for example, the latency increase gets even more dramatic.

Cause #3: Poor Software Configuration

Poor software configuration can cause problems ranging from application performance to downtime, and can include data corruption and data loss.

The configuration of software components is important to database performance. Most components are usually correctly configured, but just one misconfiguration might cause huge problems.

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 cause connection errors, poor performance or even crashes. Or, 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.

PERCONA

Cause #4: Resource Saturation

Another common reason for poor performance is a lack of resources. The system has only so much memory, CPU, disk IO and network resources, and can't do more than these constraints allow. By optimizing your software configuration, schema, or queries, often the same actions will require less of these types of resources.

Cause #5: The Database

Finally, there is database configuration. When users start complaining about application performance, how do you know when it is the database and not something else (like the issues above) causing the performance degradation? Sometimes the answer is simple: you find the application's queries running for more than 30 seconds in the database process list, for example. But in other cases, it might not be so obvious.

Percona can help you determine if your database issues are one of the items above, or something else entirely. Our Database Performance Audit, performed by one of our database experts, is a great first step. Contact us online or at sales@percona.com.