

Query Processing Bottlenecks

by Sophia



WHAT'S COVERED

This tutorial explores some of the query processing bottlenecks in two parts:

1. Introduction
2. Types of Bottlenecks
 - a. Memory Bottlenecks
 - b. Input/Output Bottlenecks
 - c. Processing Bottlenecks
 - d. Other Bottleneck Sources

1. Introduction

Query processing bottlenecks can occur in a variety of scenarios. It depends on the purpose of the database (whether it is a transactional or analytical database), the number of concurrent users, and the database selection itself. The most common type of bottleneck generally comes from different transactions competing for the same data rows. The issue here comes from shared buffers and locks on the data that can generate a variety of issues that would cause bottlenecks.

There can be many other bottlenecks that may be specific to the hardware, like memory (RAM), the central processing unit (CPU), or input/output (I/O) constraints.

2. Types of Bottlenecks

2a. Memory Bottlenecks

Memory bottlenecks occur when there is insufficient physical memory on the system available to the database. They can also be due to the database activities taking up the available memory. Typically, if we are seeing longer query execution times, excessive input/output on the system, or even out-of-memory messages in the logs, these are good indicators that there is an issue. To avoid memory-related bottlenecks, typically we would want to run query optimizers to improve the query efficiency. Potentially adding more physical memory could also resolve a bottleneck. Without enough memory (RAM), all the running processes on the system would have to compete for this resource, as it is shared among them.

2b. Input/Output Bottlenecks

We can see input/output bottlenecks if there isn't enough storage on the system to support the database. In those situations, we will see longer response times from the database or time-outs on queries. To help avoid these types of issues, it would be useful to set up monitoring on the database to identify which activities use a lot of storage and figure out the best ways to fix those problems. In addition, the storage/hard disks will also depend on the speed and transfer rates of the disk. The storage is not only used for the end-user data but also for the rest of the system data as well. If there is not enough physical memory, the OS can use the hard disk for virtual memory.

2c. Processing Bottlenecks

With CPU bottlenecks, the most common issue is insufficient hardware resources. We can generally identify these issues by looking at the logs to see if the database is heavily using the CPU. Generally, to help avoid CPU-related issues, it can be beneficial to have only the database running on the server and run all the other software on a separate machine. It can help to identify the CPU hogs and then tune the issues accordingly. If the CPU has high utilization, the database would be very slow for work that is performed and would affect all of the other applications and processes on the system.

2d. Other Bottleneck Sources

Network issues can also be a bottleneck, although the issues are typically external to the database itself. However, this component helps to connect the database server to the clients. If we have many network nodes accessing the database at the same time, there can be a limited amount of bandwidth that has to be shared among all clients.

Application code is also another common issue that generally isn't caused by hardware. The limitation of the application code can be improved with code optimization and query optimization as long as the underlying database is well-designed.



SUMMARY

Query processing bottlenecks can be caused by issues internal to the database as well as external. Internal issues can result from limitations in the available memory or computer processing speed, while network bandwidth and poorly optimized application code can affect performance from outside the database.

Source: Authored by Vincent Tran