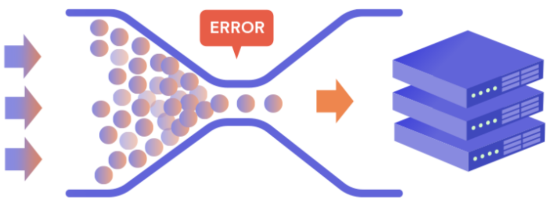
Performance Bottlenecks: Common Causes and How to Avoid Them

February 2, 2023 Tech

Hey there! Is there anything you need help with?

Are you experiencing downtime? Is your application taking too long to load, or maybe acting up, but you're not sure why? Unfortunately, these problems might drive end-users straight to your competitors' more-easily-accessible applications. We can help! To sustain optimal application performance and keep your users satisfied, understanding what bottlenecks are, how to identify them, and what steps you can take to avoid them is essential.

What is a performance bottleneck?



A bottleneck refers to the point where a single component limits the capacity of a computer system or an application. The component, whether in the software stack or the hardware infrastructure, restricts the flow of the process. It limits the system's overall capability, leading to problems such as increased downtime, long load times, and high levels of end-user frustration. Imagine the neck of a bottle; it tapers from a slightly wider base to a narrower opening at the top. A narrow neck might be optimal for your favorite beverage bottle but not for application functionality!

How do Bottlenecks Impact User Experience?

In today's world of quickly-advancing technology, we all know that customers can only become satisfied if an application performs as expected or better! User experience is critical in our field, and exploring the issues that affect user experience is vital for keeping customers satisfied. User experience is heavily impacted if the application does not run as expected or the load time is too long. In addition to negatively impacting a website's SEO, making it challenging to find on SERPs, bottlenecks prevent an application from running properly.

How to Identify a Bottleneck:

Symptoms of bottlenecks most commonly include system downtime, slow response times, long load times, and software breaks. When faced with an application bottleneck, you might think: "Is there an error in the code? Is it the CPU, the network, or a memory use issue?". There are many possible causes for the performance slow-down, but the tools at Gatling and Gatling Enterprise can help discover the culprit and get your application running smoothly.

Identifying bottlenecks is becoming increasingly more difficult as application software becomes more and more complex. Performance testing is the primary approach to identifying the cause of a performance bottleneck. It is valuable to test website traffic load (https://gatling.io/features/simulating-heavy-traffic/) to analyze the application's performance in various circumstances. The first step to identifying and resolving a bottleneck is gathering data and establishing a strategy to optimize speed and overall performance.

Common Causes of Application Performance Bottlenecks:

Understanding the common causes of bottlenecks is imperative for avoiding them. So let's explore common application performance bottlenecks and learn how to prevent them from becoming your next performance slow-down.

Software Limitations:

Although a massive amount of time, energy, communication, and organization goes into application software development, it could be the source of the constricting bottleneck. If you build your application using a framework that cannot utilize multiple CPU streams, you won't be able to use your multi-core processors or high RAM. Then, you would face a software bottleneck.

Good code can go a long way in optimizing application performance. Analyze the code's efficiency and use of system resources as the first step in troubleshooting a software bottleneck. Clean, efficient code is essential for an open process. When code executes quickly and efficiently, the application is reliable, and end-users are satisfied. Avoid extraneous functions or unnecessary loops! Follow development-based best practices, use group code reviews, and keep the communication among the development team open and transparent to help resolve or even avoid software bottlenecks. Code might also require rewriting or patching to get the application's performance back on track.

CPU Utilization:

The CPU is a likely culprit for most performance bottlenecks. This occurs when the processor cannot perform the requests asked of it. Too many processing requests can overload the system, causing slow-downs. CPUs run database inquiries, handle information requests, execute programs, and process data. A bottleneck arises when the data that can be processed are less than the data waiting to be processed. The processing capacity is lacking, causing a bottleneck. Unanticipated heavy traffic, inefficient server-side scripting, or insufficient caching can cause CPU utilization bottlenecks.

An example of a CPU bottleneck can be seen in game-based applications.

The bottleneck can happen if the CPU processing speed is slower than the GPU processing speed. The GPU or the graphics card demands a lot of power from the CPU. If the CPU cannot keep up with the GPU, the user experience is impacted by lower fps, stutters, and frustrating slow-downs. Since the CPU and the GPU rely on each other to render the final image, a bottleneck develops if one holds up the other.

The number of requests and the processor load must stay within the server's CPU power limits to run smoothly. Depending on the application's requirements, you may need to upgrade to a more powerful processor to ensure a larger cache, clock speeds, or supported threads.

Database Queries:

Too many database calls that are made for trivial tasks can cause bottlenecks. In addition, restricted access to back-end database servers can severely impact performance and frustrate end-users. To avoid a slow-down caused by database queries, locate the longest parts of the queries during development and explore how you might shorten the execution time frame.

Utilize database monitoring by using load-testing scripts to generate traffic to the database server. Analyze the data to see how the queries performed under load. Then, keep testing and fine-tuning to optimize performance.

Memory Utilization:

RAM holds data and makes it accessible to the processor. The RAM resources of the server system determine the number of processes your CPU can perform. Therefore, insufficient RAM would hinder the performance of an application. For example, a memory bottleneck would arise without enough RAM, even with a very powerful CPU for your servers.

To repair or avoid a memory utilization bottleneck, upgrade your RAM's speed and memory. Also, inspect your code, as issues such as memory leaks or memory bloat can result in bottlenecks. Memory leak refers to a slow and gradual increase in memory usage, while memory bloat refers to a sudden increase in memory usage.

Network Utilization:

If the amount of data transfer is limited due to the insufficient bandwidth of a network, a network bottleneck can surface. The bottleneck limits the flow of the network's data; this type of hot spot often results in frozen web pages or slow data transfer, leaving end-users frustrated and tempted to bring their business elsewhere. The network bottleneck could happen if the server or a network communication device is overloaded, if the hardware is mismatched, or if there is excessive fan-in. An example of a hardware mismatch would be if a team server were fitted with a GbE port, but the corresponding switch port connected to the server offers a legacy 10/100 Ethernet port. The slow switch would create a bottleneck

Testing network performance at various points along the data path would allow you to identify points performing more slowly than others. The network bottleneck can be avoided or resolved by upgrading or adding servers or upgrading network hardware such as routers and hubs.

Disk Usage:

Long-term storage, including HDDs and SSDs, can undoubtedly be the culprit for a hard-to-troubleshoot bottleneck. A disk usage bottleneck is most likely boiled down to an issue with the physical disk size or how quickly data is read from or written to the disk. When the server-side code includes multiple disk operations, the bottleneck can be resolved by upgrading to SSD disk storage. Other resolution strategies include increasing caching rates in RAM or reducing fragmentation.

Poor design of the disk storage system is often to blame for this type of bottleneck. Examples of design flaws could include having too few spindles in a RAID group, using a small array cache, or using underperforming SATA drives with low RPMs.

Understanding how your physical storage aligns with underlying LUNs, disks, and RAID groups can help you to avoid storage-related bottlenecks. Resolving this bottleneck is especially crucial for dynamic web applications requiring frequent data transfer among the client, server, and database.

Can Bottlenecks Change Locations?

There are various circumstances under which a bottleneck might change to another component. These might include the fluctuating demands of the workload or the relationship between the hardware and the software.

Workload:

The demands of the workload can drastically change in just seconds. I/O sizes or fluctuating application activity, for example, can influence the location of a bottleneck.

Hardware – Software Relationship:

Software is developed using assumptions about the capabilities of the hardware. If the hardware is upgraded, limitations in the software will become apparent. Therefore, software needs to be optimized for upgraded hardware when necessary.

The Importance of Load Testing to Analyze Application Performance

Ever-changing technology, customer expectations, and user behavior give developers challenging opportunities to push their skills and keep the application development field progressing. With these opportunities, challenges such as identifying and resolving bottlenecks are inevitable but manageable!

To meet these challenges, utilize a process such as load-testing (https://gatling.io/2020/02/what-is-load-testing/) to identify bottlenecks and a strategy to fix or avoid them altogether. In addition to identifying bottlenecks, load-testing reports (https://gatling.io/features/response-times-reports/) can help you test improvements and analyze the results. Gatling and Gatling Enterprise can help with client-side and server-side metrics, such as runs comparisons, TCP connections, DNS resolutions, and monitoring. Check out the resources (https://gatling.io/resources/) available at Gatling to see how we can help you optimize the performance of your application.

Related resources

You might also be interested in…

(https://gatling.io/2022/11/why-we-test-the-beginners-guide-to-why-load-testing-is-important/)

Peter - Customer Success Manager

Why we test: the beginners guide to why load testing is important

(https://gatling.io/2022/11/why-we-test-the-beginners-guide-to-why-load-testing-is-important/)
What are the challenges that can affect all web based enterprises and some of the advantages that can be gained through load testing.

2 November, 2022

(https://gatling.io/resourc

Tutorial #14 – Getting re

(https://gatling.io/resourc
Your website is expecting
don't know how to prepar
help you figure it out!

20 October, 2020

Share the news

n.com/shareArticle?
ling.io/2023/02/performa
-causes-and-how-to-
ce Bottlenecks: Common
m&summary=&source=)

(https://www.facebook.com/sharer/sharer.php?u=https://gatling.io/2023/02/performance-bottlenecks-common-causes-and-how-to-avoid-them/)

(http://twitter.c
Bottlenecks: C
Them&url=https://gat
bottlenecks-commor

‹ OLDER POST
Private Locations Update | Kubernetes
(HTTPS://GATLING.IO/2023/01/PRIVATE-LOCATIONS-UPDATE-KUBERNETES/)

NEWER POST ›
Welcome to Antoine!
(HTTPS://GATLING.IO/2023/02/WELCOME-TO-ANTOINE/)

Why Gatling?

Gatling (https://gatling.io/open-source/)

Gatling Enterprise (https://gatling.io/enterprise/)

Features (https://gatling.io/features/)

Integrations (https://gatling.io/integrations/)

Services
(https://gatling.io/services/)

Learn more

Blog
(https://gatling.io/blog/)

Gatling Academy
(https://gatling.io/academy/)

Documentation
(/docs/)

Resources
(https://gatling.io/resources/)

Gatling Community
(https://gatling.io/community/)

Gatling Corp

Our company
(https://gatling.io/company/)

Events
(https://gatling.io/events/)

Join us
(https://www.welcometothejungle.com/fr/companies/gatling-corp)

Contact
(https://gatling.io/contact/)

Follow us

Community

Facebook

Twitter

LinkedIn

YouTube

GitHub

Subscribe to our Newsletter (https://content.gatling.io/gatling-subscribe-to-the-newsletter)

Privacy Terms of service