Latency in System Design

Imagine a pipe with water flowing through it. The speed at which water flows through the pipe may vary i.e. sometimes flowing very fast and sometimes flowing very slow. This concept is similar to latency in system design. Latency determines the speed at which data can be transferred from one end of a system to another.

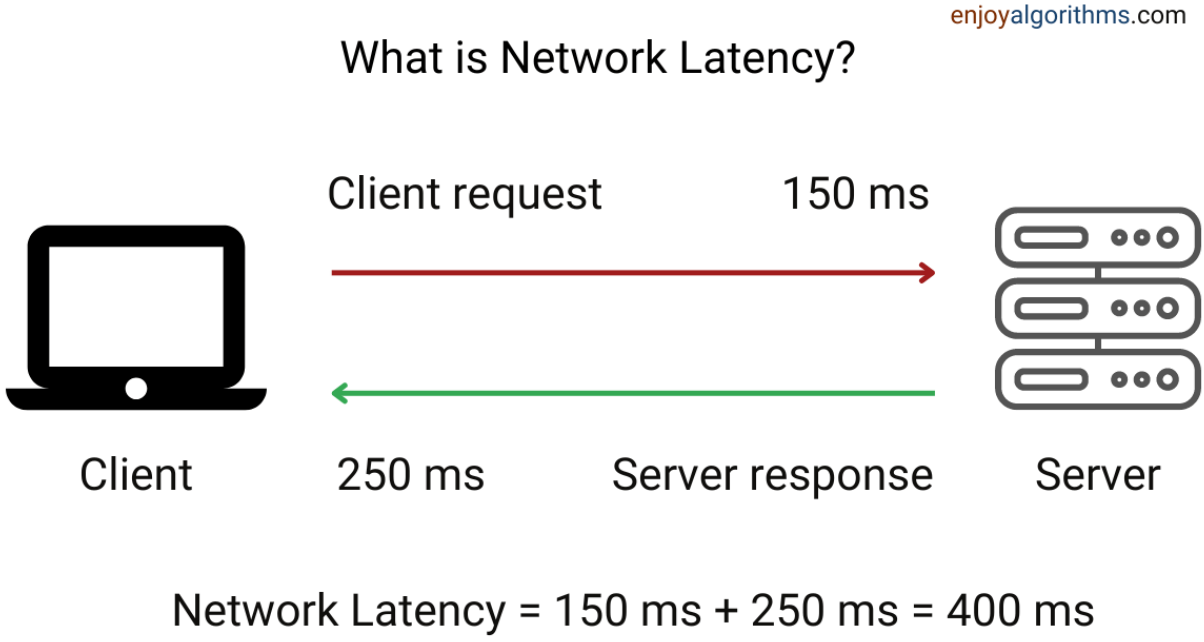
In this blog, we will focus on the concept of latency and how it affects the performance of a system. We will also discuss measures that can be taken to improve the latency.

What is Latency?

Latency is a measure of how quickly data can be transferred between a client and a server. It is directly related to performance of the system: Lower latency means better performance. Let's understand this from another perspective!

Browser (client) sends a signal to the server whenever a request is made. Servers process the request, retrieve the information and send it back to the client. So, latency is the time interval between start of a request from the client end to delivering the result back to the client by the server i.e. the round trip time between client and server.

Our ultimate goal would be to develop a latency-free system, but various bottlenecks prevent us in developing such an ideal system. The core idea is simple: We can try to minimize it. Lower the system latency, the less time it takes to process our requests.



How does latency work?

Latency is nothing other than the estimated time the client has to wait after starting the request to receive the result. Let's take an example and look into how it works.

Suppose you interact with an e-commerce website, you liked something, and added it to the cart. Now when you press the "Add to Cart" button, following events will happen:

- The instant "Add to Cart" button is pressed, the clock for latency starts and browser initiate a request to the server.
- Server acknowledges the request and processes it.
- Server response to the request, the response reaches your browser, and product gets added to your Cart.

You can start the stopwatch in the first step and stop the stopwatch in the last step. This difference in time would be the latency.

What causes Latency?

Latency in a network depends on various parameters. One of the major

factors contributing to latency is outbound calls. In the previous example of adding cart exercise, when you click the button on browser, the request goes to some server in the backend, which again calls multiple services internally for computation (in parallel or sequentially) and then waits for a response. All this adds to the latency.

Latency is mainly caused by following factors:

Transmission mediums: Transmission medium is the physical path between the start and endpoints. So system latency depends on the type of medium used to transmit requests. Transmission mediums like WAN and Fiber Optic Cables are widely used, but each medium has limitations.

Propagation: It refers to the amount of time required for a packet to travel from one source to another. So system latency highly depends on the distance between communicating nodes. Farther the nodes are located, more the latency.

Routers: Routers are an essential component in communication and take some time to analyze the header information of a packet. So latency depends on how efficiently router processes the request. Router to router hop increases the system latency.

Storage delays: System latency also depends on the type of storage system used, as it may take some time to process and return data. So accessing stored data can increase latency of the system.

How to measure Latency?

There are many methods used to quantify latency. Three most common methods are:

Ping: Ping is the most common utility used to measure latency. It sends packets to an address and sees how fast response is coming. It measures how long it takes for the data to travel from source to destination and back to the source. A faster ping corresponds to a more responsive connection.

Traceroute: Traceroute is another utility used to test latency. It also uses packets to calculate the time taken for each hop when routed to the destination.

MTR: MTR is a combination of both ping and Traceroute. MTR gives a list of reports on how each hop in a network is required for a packet to travel from one end to the other. The report generally includes details such as percentage Loss, Average Latency, etc.

Latency optimization

Latency restricts performance of the system, so it is necessary to optimize it. We can reduce latency by adopting following measures:

HTTP/2: We can reduce latency by the use of HTTP/2. It allows parallelized transfers and minimizes round trips from the sender to the receiver.

Less external HTTP requests: Latency increases because of third-party services. By reducing number of external HTTP requests, system latency gets optimized as third-party services affect speed of the application.

CDN: CDN stores resources in multiple locations worldwide and reduces the request and response travel time. So instead of going back to the origin server, request can be fetched using the cached resources closer to the clients.

Browser Caching: Browser caching can also help to reduce the latency by caching specific resources locally to decrease the number of requests made to the server.

Disk I/O: Instead of often writing to disk, we can use write-through caches or in-memory databases or combine writes where possible or use fast storage systems, such as SSDs.

Latency can also be optimized by making smarter choices regarding storage layer, data modelling, outbound call fanout, etc. Here are some

ways to optimize it at an application level:

- Inefficient algorithms are the most apparent sources of latency in code. It is necessary to avoid unnecessary loops or nested expensive operations.
- Use design patterns that avoid locking because multithreaded locks introduce latency.
- Use an asynchronous programming model for better usage of hardware resources because blocking operations can cause long wait times.

Conclusion

Latency is a concept associated with the design of every system. One can never make a fully latency-free system, but one can easily optimize it. With modern hardware and heavy computationally efficient machines, latency is no longer a bottleneck of the system.

Thanks to Chiranjeev and Suyash for their contribution in creating the first draft of this content. If you have any queries or feedback, please write us at contact@enjoyalgorithms.com. Enjoy learning, Enjoy system design, Enjoy algorithms!