The fundamental knowledge of System Design — (part 3) — Throughput & Latency

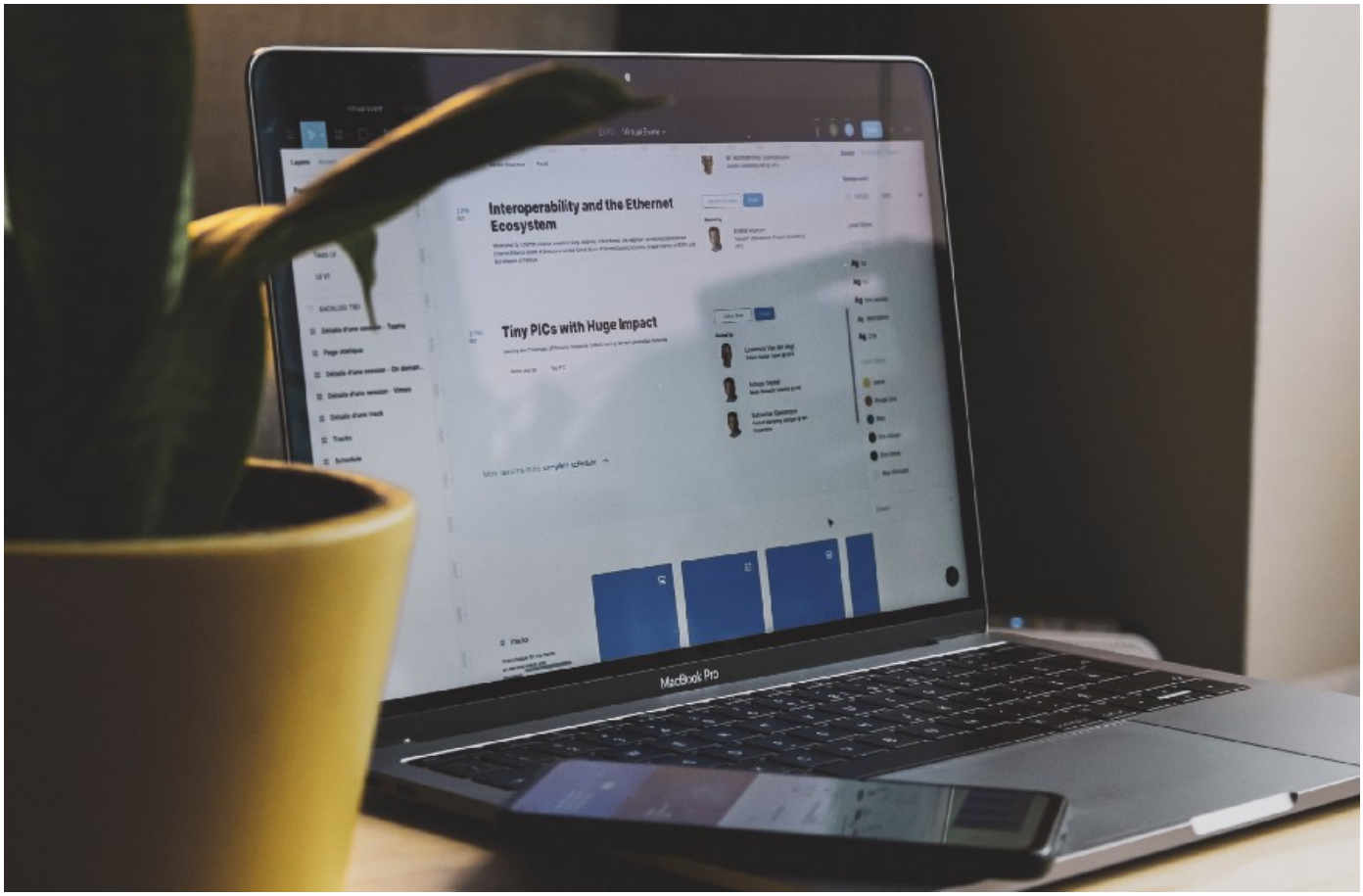


Photo by Sigmund on Unsplash

Servers are the core of today's computational world. Server performance depends on **throughput and latency**. Generally speaking, DDR-DIMMs (Double Data Rate Dual In-line Memory modules) are used as server memory. SSD/HDD is used as storage in the current trend. So, measuring the level of throughput or latency can help to identify performance issues on the network.

Throughput

Throughput usually refers to the number of queries or requests processed in a given period. There are 2 factors that determine the upper limit of throughput: 1) the number of

hardware resources available, and 2) the resource allocation and effective utilization in a system.

PS: The following are the main concepts and calculation formulas of the performance test.

The throughput of a system is closely related to the CPU consumption of requests, external interfaces, IO, and so on. The higher the CPU consumption of a single request, the slower the external system interface and IO impact speed, the lower the system throughput, and vice versa.

Parameters of system Throughput

1. **QPS/TPS** (Number of request/query/transactions per second): The number of requests/transactions per second
2. **Concurrency:** The number of requests/transactions processed by the system at the same time
3. **Response Time**: The time to complete a request (Generally take the average response time)

After understanding the meaning of the above 3 elements, the relationship between them can be deduced:

****QPS/TPS** = concurrency/average response time**

Determine the throughput

So, the throughput of a system is usually determined by 2 factors: the number of concurrencies and QPS/TPS. These 2 values for each system have a relative limit value. As long as a certain item reaches the highest value of the system, the system's throughput reaches its maximum. If the number of requests and concurrencies continues to increase, the system's throughput will decrease instead. It is because the system is overloaded, and

other consumptions such as context switching, memory, etc cause the system performance to decline.

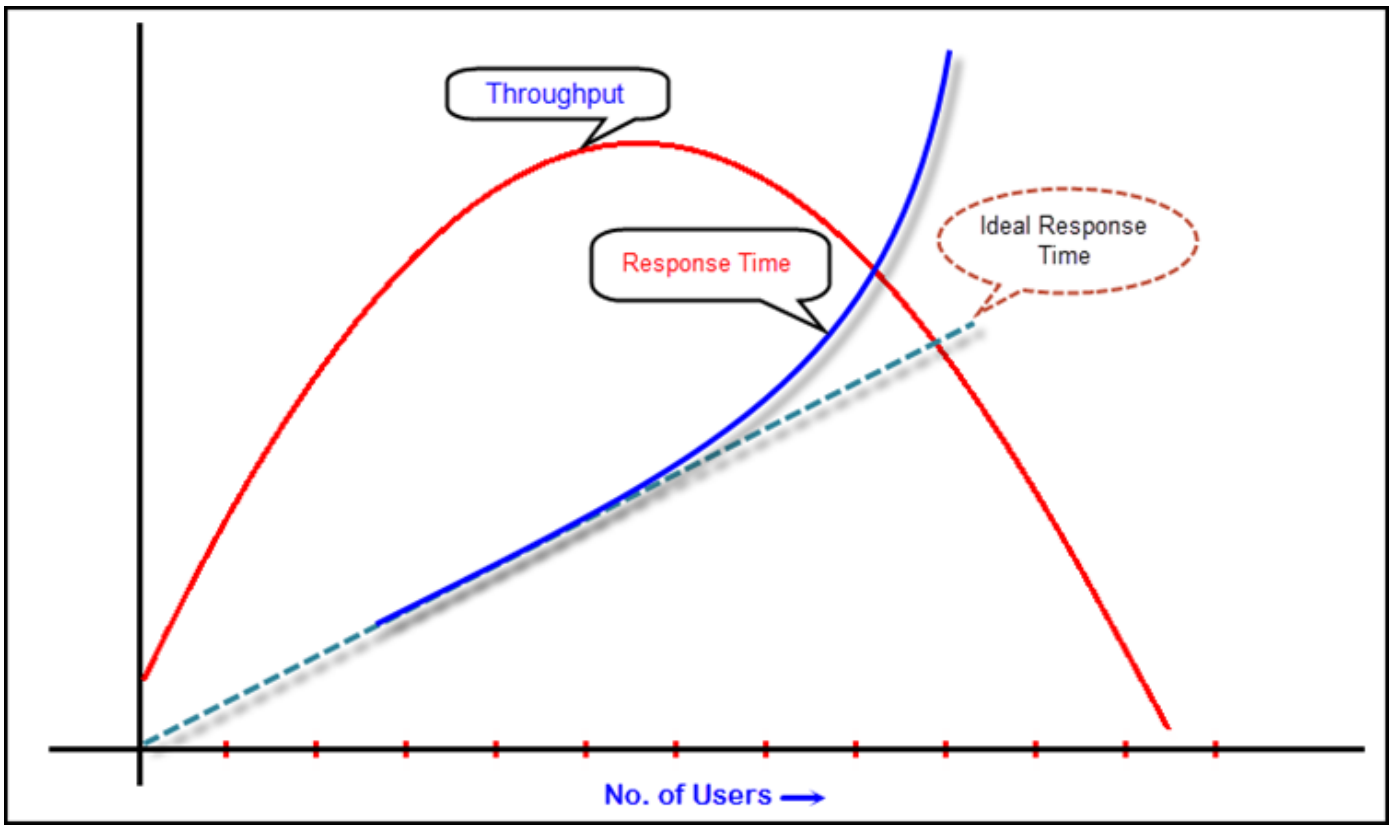
Determine the response time

The response time of a system is the system impact time in which the system will go through a critical path. The critical path is composed of CPU operations, IO, external system responses, and so on.

When we do system design, we need to consider the impact of CPU operations, IO, and external system response factors, as well as preliminary estimates of system performance.

Besides the QPS, concurrency, and response time, we also need to consider daily peak value. By observing the access log of the system, it is found that in the case of a large number of users or the highest TPS, the access traffic in the same time period in each day have a relatively stable relationship except for holidays and seasonal factors.

The relationship between the number of users, QPS, response time, and throughput.



https://sqa.stackexchange.com/questions/14302/how-should-i-intrepret-the-jmeter-result

Latency

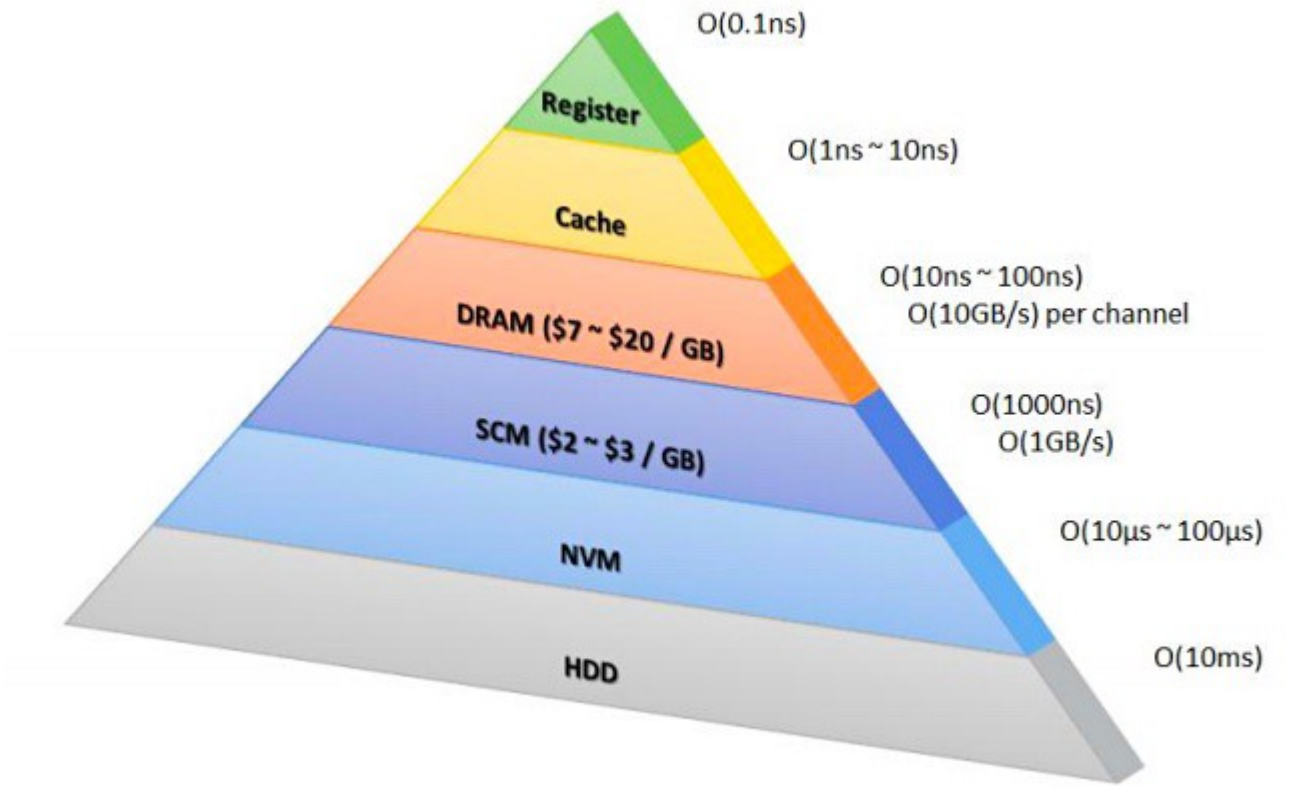


Figure 1: key attributes of memory and storage classes

https://blocksandfiles.com/2018/11/28/2019-the-year-of-storage-class-memory/

Latency generally includes one-way delay and round trip delay. Its unit is generally ms, s, min, hr, and so on.

End-to-end latency = client processing latency + Network latency + server processing latency

- **Server processing delay** (calculation delay): The time it takes for a task to enter processing and finish processing. Assuming that the available amount of computing resources, effective utilization, and energy conversion efficiency remain unchanged, the tasks scheduling algorithm will determine the processing delay of a task.

Network Delay = sending delay + propagation delay + processing delay + Queuing delay

- **Sending delay:** The time it takes for a host or router to send a data frame (first bit of the data frame to the last bit of the frame)
- **Transmission delay**: Data frame length (b)/channel bandwidth (b/s)
- **Propagation delay:** The time it takes for an electromagnetic wave to propagate a certain distance in the channel medium outside the machine (channel length (m)/electromagnetic wave propagation speed on the channel (m/s))
- **Processing delay:** When a host or router receives a packet, it takes a certain amount of time to process, such as analyzing the header of the packet, extracting the data from the packet, performing error correction, or finding the suitable route
- **Queuing delay:** When a packet is transmitted through the network, it must pass through many routers. After entering the router, the packet must be queued in the input queue for processing. After determining the forwarding interface, it must be queued in the output queue.

The user's point of view

When clicking a link, until the system displays the result in a user-perceived form, the time consumed in this process is response time. When the response time is short, the user experience is very good. The user experiences include personal subjective factors and objective response time. When designing an application or software, we have to consider these 2 factors to achieve the best user experience.

1. Personal Subjective
2. Response time

The administrator's point of view

1. The Corresponding time

2. Can the system be expanded?
3. Can the system support 7 X 24 hours of business access
4. Change the storage device to improve performance
5. The use of server resources is reasonable
6. The use of application server and database resources is reasonable
7. How much the system can handle TPS and support how many users access

The developer's point of view

1. Is the architecture design reasonable?
2. Is the database design reasonable?
3. Is there unreasonable memory usage in the system?
4. Is there any unreasonable thread synchronization method?
5. Is code performance good?

The performance test engineer's point of view

1. Network transmission time = N1+N2+N3+N4
2. Application server processing time =A1+A3
3. Database server processing time =A2
4. Response time = N1+N2+N3+N4+A1+A3+A2

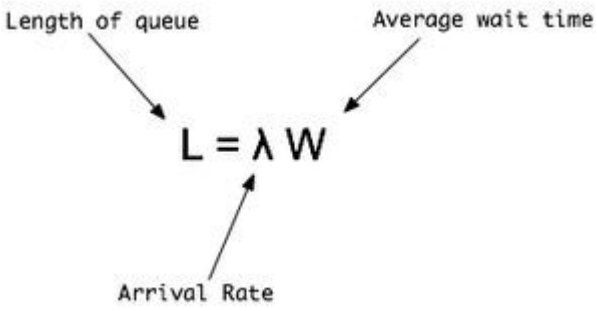
Throughput expressed in a different way can illustrate problems at different levels. For example, the number of bytes per second can be expressed as a bottleneck due to network infrastructure, server architecture, application server constraints, etc.; the number of requests per second indicates that the bottleneck is mainly reflected by the constraints of the application server and application code. Resource utilization: refers to the usage of

various resources in the system, such as CPU occupancy rate of 70% and memory occupancy rate of 35%.

Latency is affected by task scheduling, which is usually used to determine the time allocation rules for individuals to use computing and network resources. There is a relationship between latency and throughput, but there is no absolute correlation. We can only say that throughput does well to the overall energy efficiency status at the macro-level (how individuals participate in the allocation of energy efficiency resources and don't care about the overall situation). The time delay is for individuals.

When the two change in the same direction, a simultaneous increase can be achieved, usually 1 increase and 1 decrease or the same, depending on the factors affecting the delay or throughput change. For example, when we reduce the useless power, it can reduce latency and improve throughput; reducing task concurrency can reduce latency and throughput; a reasonable increase in concurrency can improve throughput and increase latency; excessively increasing concurrency can lead to both at the same time.

Little's law expresses the relationship between throughput and latency. When a system enters a steady-state, the flow of outflows does not fluctuate significantly.



https://www.shmula.com/cycle-time-reduction-littles-law/9023/

Optimization

- Find out the bottleneck in the system

- Determine the optimization method by evaluating the expected benefits of optimization which requires theoretical analysis support

Amdahl Law is an optimization that gives theoretical speedup in latency of the execution of a task at a fixed workload. It defines the performance improvement or speedup in execution time that can be achieved by speeding up the processing of a device.

$$\text{Overall Speedup} = \frac{\text{Old execution time}}{\text{New execution time}}$$

$$= \frac{1}{\left((1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}} \right)}$$

https://www.geeksforgeeks.org/computer-organization-amdahls-law-and-its-proof/