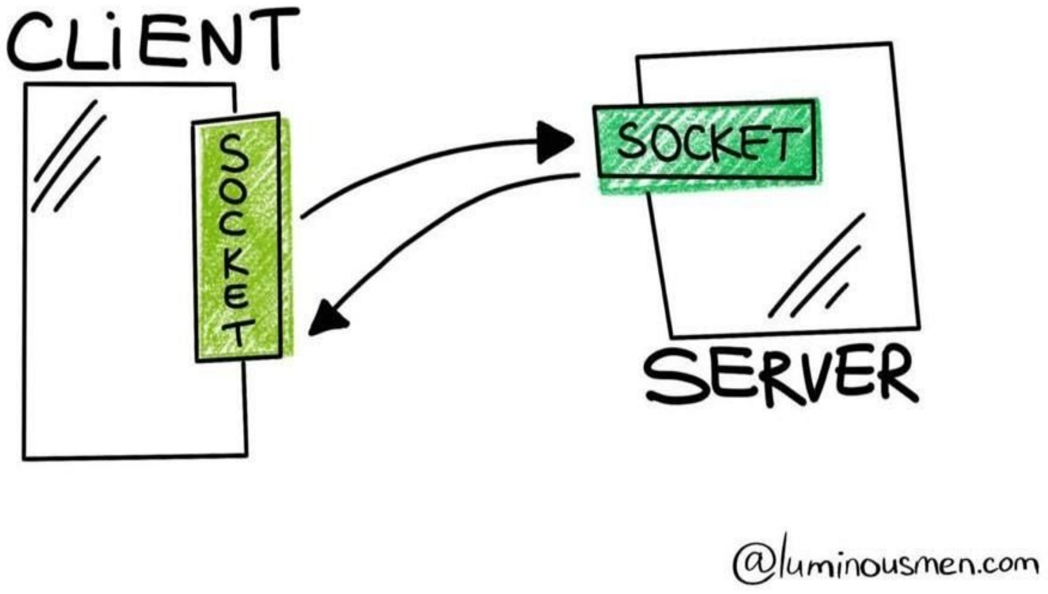
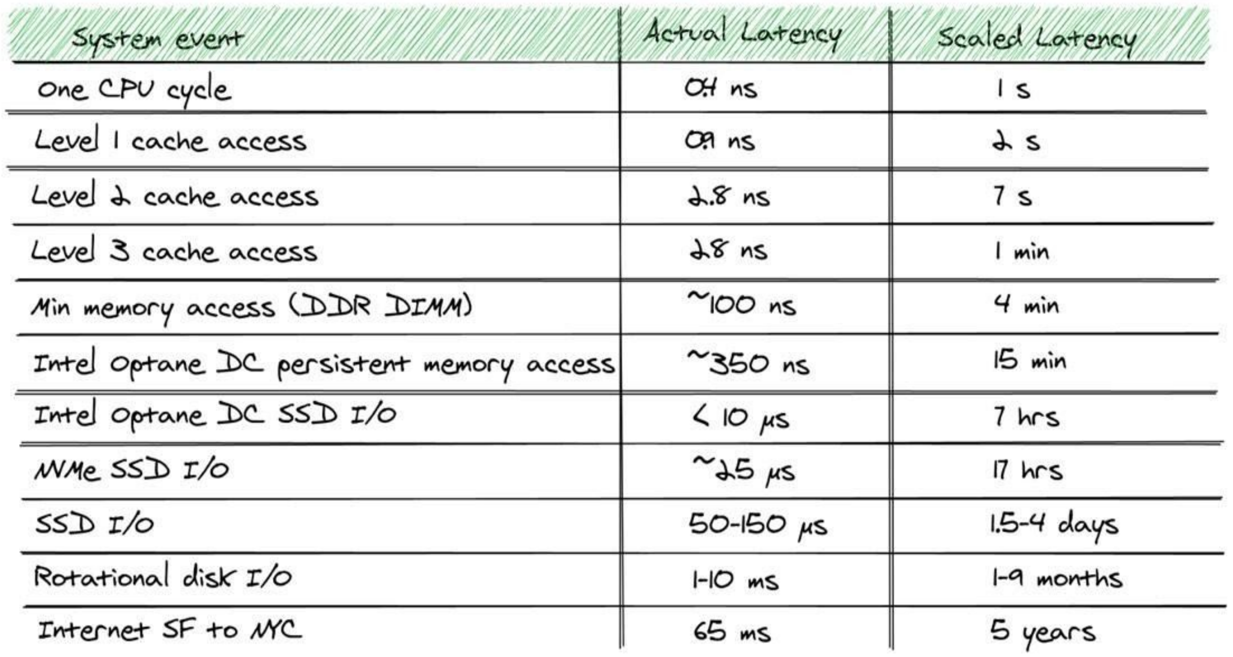
CHAPTER 1
BLOCKING I/O AND NON-BLOCKING I/O

One way or another, when you have a question about blocking or non-blocking calls, most commonly it means dealing with I/O. The most frequent case in our age of information, microservices, and FaaS will be processing requests. We can now imagine that you, dear reader, are a user of a web site, while your browser (or the application in which you're reading these lines) is a client. Somewhere in the depths of the amazon.com, there is a server that handles your requests to generate the same lines that you're reading.

In order to start an interaction in such client-server communications, the client and the server must first establish a connection with each other. We will not go into the depths of the 7-layer model and the protocol stack that is involved in this interaction, as I think it all can be easily found on the Internet. What we need to understand is that on both sides (client and server) there are special connection points known as sockets. Both the client and server must be bound to each other's sockets, and listen to them to understand what the other says on the opposite side of the wire.



In this communication, the server is doing something — either processes the request, converts markdown to HTML or looks where the images are, it performs some kind of processing.



System event	Actual Latency	Scaled Latency
One CPU cycle	0.4 ns	1 s
Level 1 cache access	0.9 ns	2 s
Level 2 cache access	2.8 ns	7 s
Level 3 cache access	28 ns	1 min
Min memory access (DDR DIMM)	~100 ns	4 min
Intel Optane DC persistent memory access	~350 ns	15 min
Intel Optane DC SSD I/O	< 10 μs	7 hrs
NVMe SSD I/O	~25 μs	17 hrs
SSD I/O	50-150 μs	1.5-4 days
Rotational disk I/O	1-10 ms	1-9 months
Internet SF to NYC	65 ms	5 years

If you look at the ratio between CPU speed and network speed, the difference is a couple of orders of magnitude. And if the application uses I/O most of the time, in most cases the processor simply does nothing — I/O becomes a bottleneck. This type of application is called I/O-bound.

There are two ways to organize I/O: blocking and non-blocking.

Also, there are two types of I/O operations: synchronous and asynchronous.

All together they represent possible I/O models.