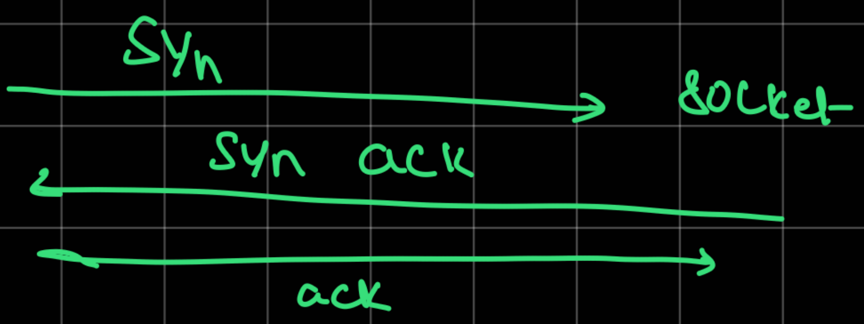
Java multithreaded web server

TCP protocol

Client 3 way handshake Server

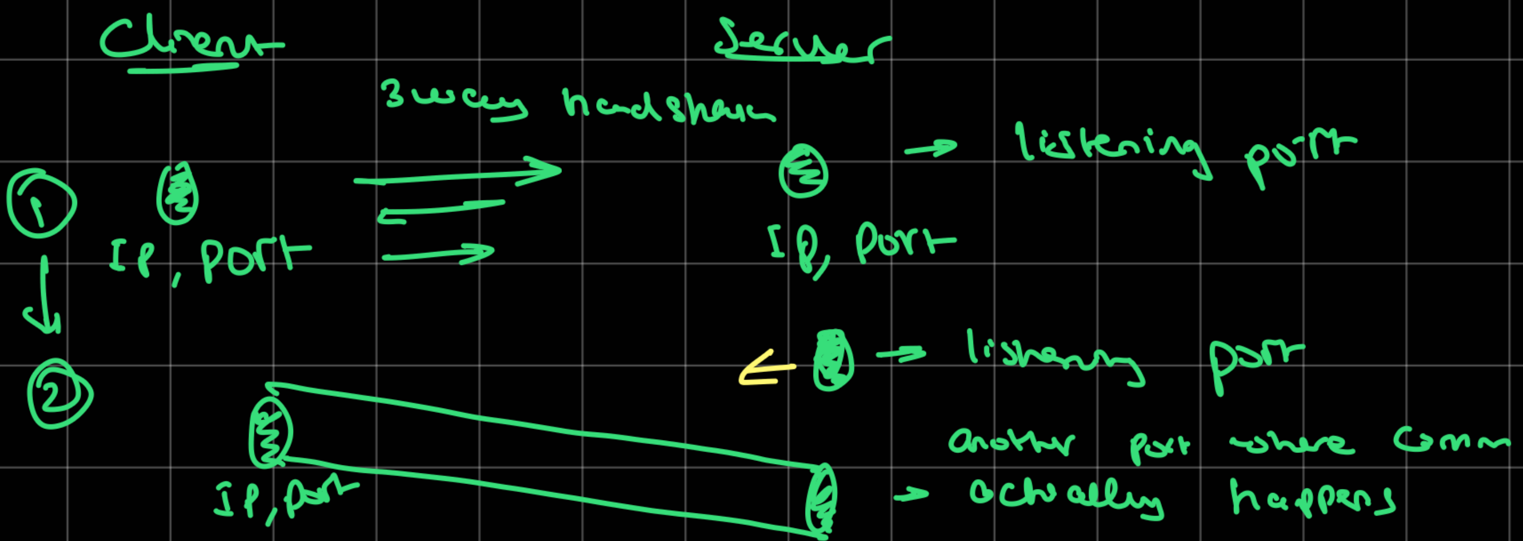


HTTP : → 1.0 : non persistent

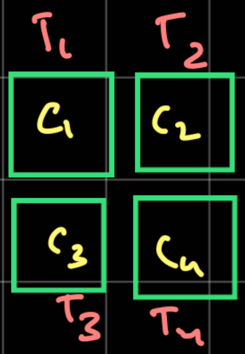
Client breaks contact with server after server fulfills request of data of one request

→ 1.1 : does not break contact until a timeout

↳ To close connection manually client sends Fin flag and server sends Ack flag



Context Switching: assume 4 CPU's in a multithreaded System



runtimes of threads : $T_1 = 1s$

$T_2 = 15ms$

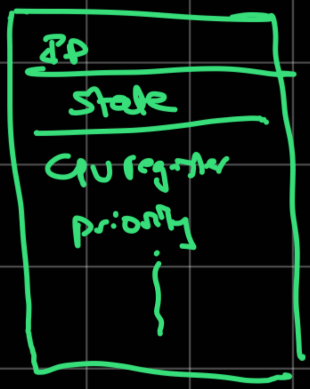
lets say T_1 is idle now and waiting for I/O Ops

as T_1 will run for 1 sec and T_2 for only 15 sec and now T_1 is idle. we make T_2 run on C_1. This is called Context Switching

Disadvantages of Multithreaded Servers

assume 10k requests per second 10K RPS

Each thread has a thread block



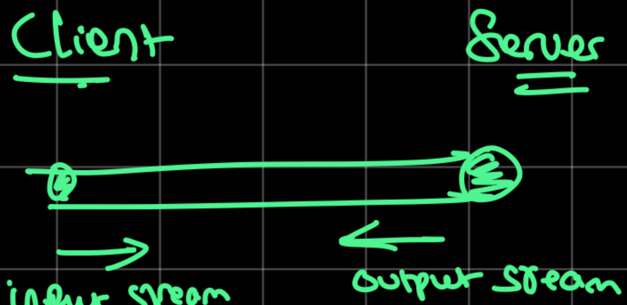
like this we have 10K TBS ↙ Thread blocks

but if we create 10K threads it will eat the hardware

Thread pool :

we make many threadpools of with each pool containing many threads as and when a new client requests anything the task is put in the task Queue and any thread free in the pool is given the task

Printwriter and Buffered reader



↳ in bytes 1011--
so use buffred reader

↳ need to Send in bytes
So will use printwriter format