I did whole-system testing with curl commands. I sent multiple GET and PUT from one or two terminals. I used & to send parallel requests from one terminal.

I tried four put requests on httpserver.cpp and it took about 1.6-2.2 milliseconds each time. Although the filesize wasn't 4MB, it was way less than that. When I tried it with multi-threaded-server it took about 0.6-1.0 milliseconds each time. I was using 4 threads. I tried using fewer threads and it didn't make a difference, but when I tried 100 threads, it took a lot more time since it takes time to create different threads. So I guess more is not always good. It depends on the number of requests.

The bottleneck that I have is writing and reading files. It doesn't how fast my threads are, It won't change the throughput of reading and writing. There is no concurrency in dispatch. In the worker part, reading from the socket and parsing the header is happening concurrently. In logging, logging to file is happening concurrently. We can increase concurrency if we check different not accessing the same file, then reading and writing to file can happen concurrently.