Using $time ./4curl.sh where there are 4 curl GETS.
Using time on my httpserver that is single threaded
Real .252s
User .071s
Sys .097s
Multi threaded gave
Real:0.288S
User .052s
Sys .135s

There wasn't that much difference between the multi threaded and the non multithreaded. On average the multi threaded finishes faster in “user” but has a bigger number in “sys”. I think that the most likely bottleneck of this system is that there is only one dispatcher. That means there is only one thread listening and accepting requests. This is almost the same as single threaded and the only difference is that the dispatcher doesn't have to do the work. In my implementation I used semaphores to lock the memory the threads share so no more than 1 thread accesses it. The dispatcher has to wait for that same mutex so he may not be able to quickly add incoming requests because a worker is currently reading from it. I think to make this faster I need a better way of sharing the client’s fd from the dispatcher to the workers that are smoother and probably allow more than 1 dispatcher to really increase the throughput by being able to get multiple tasks added to the work queue.