



---

[download/](#)

hi there dear friends! i have this video..so please guys watching this video please comment below i am really thankful of your valuable time..i know if u are thinking about watching this video..ok give me your valuable comments and i will be thankful of that..so please for those guys i will update this video on daily basis..so please be with me for more updated video..i have uploaded other video too in my channel..so please check out those videos as well..so thanks a lot for watching this video..i will be eternally indebted of those valuable comments and opinions..hope you like this video and leave your valuable comments and experiences.please don't forget to subscribe my channel and share this with your friends.Q: java nio multithreaded performance I am implementing a service class. The service starts listening to a sockets, reading all the input and performing some computation on it. I am using thread pools to loadbalance incoming requests and also accepting incoming messages from socket. In some scenarios, I need to read messages multiple times (usually 10-20 times). As the service shouldn't block and still handle incoming socket messages, do you think reading 10-20 times is acceptable? Reading this much data from a socket at one time is expensive and can be done using a threadpool. I am wondering if reading 10-20 times will have some performance impact on the system. I am thinking of how my service reacts to a message or a request if I read it for only once. EDIT: The service can handle lots of request. The request is a series of strings or java objects. So, do you think 10-20 times will be ok? A: You should read up on blocking and non-blocking IO in the official Oracle doc. Basically, you should never keep the socket open if you want to handle more requests immediately. It may work for small requests, but certainly won't for large requests. Instead, you should read/write as needed (non-blocking). If there's nothing to read or write, wait until there is. In the end, it's a balancing act, and you have to choose what will give you the best performance. If that's a load-balanced set of threads, then you can do that with non-blocking IO. If it's a one-thread system, then I'd suggest you read/write until a better time comes. If you want to learn more about