# HighLoadServers

## Accept queue size

If you are creating a lot of connections in a short time (as is often the case with load testing), then you may need to increase the size of the connection accept queue. This can be done by setting the AcceptQueueSize field on the jetty connectors. You may also need to increase the somaxconn kernel attribute to match (see below).

## High TCP Slot Usage

We have seen that sometimes under certain loads test cases would fail when they are spinning up a large amount of client and server connections as a part of the test execution. The below setting seems to address some of the issues we have encountered. It stands to reason that under a heavy load it would benefit a server outside of the test case environment as well.

### Mac OSX

```
> sudo sysctl -w kern.ipc.somaxconn=256
```

kern.ipc.somaxconn controls the size of the connection listening queue and typically only needs to be adjusted in high-performance server environments. The default value of 128 is more than adequate for a home/work machine and most workgroup servers. If, however, you are running a high-volume server and connections are getting refused at a TCP level, then you want to increase this. This is a very tweakable setting in such a case. Too high and you'll get resource problems as it tries to notify a server of a large number of connections and many will remain pending, and too low and you'll get refused connections.

From http://www.macgeekery.com/tips/configuration/mac_os_x_network_tuning_guide_revisited

### Linux

```
> sudo /sbin/sysctl -w net.core.somaxconn=256
```

This is the same setting described above for Mac OSX.

```
> sudo /sbin/sysctl -w net.core.netdev_max_backlog=3000
```

The net.core.netdev_max_backlog controls the size of the incoming packet queue for upper-layer (java) processing. This setting has allowed the developers to properly address various performance testing aspects of jetty-client, however, this isn't the only mechanism for tweaking congestion on a Linux environment.

See http://fasterdata.es.net/TCP-tuning/linux.html for further discussion on tuning TCP under Linux and alternate congestion settings.