# WEEBCHAT

A dynamically distributed telnet chat server

First of all, **Dynamic**, the word itself possibly hints “unlimited.” Second, **Distributed,** which is one, if not the best way to write a future ready server. Put them together and you have a server that can scale horizontally in *N* number of servers. Why horizontal? Well, this helps you accommodate more users and can save you more money because the marginal cost of adding one more core or a hard drive that does a few more I/O opts per second grows exponentially. In the long run, the cost of adding one more node to the system becomes far cheaper than the cost of additional hardware.

With those in mind, I decided to write a load balancer called “Lobby” to distribute workloads across multiple servers I call “School”. A school is a server that hosts a number of chatrooms. Each time a user creates a new chatroom the lobby asks each active schools if it isn’t full yet. The first school to answer the lobby that it still has space gets the privileges to host the new chatroom. Now when a user tries to join a chatroom the lobby should pass the clients connection to the specific school hosting the room freeing itself from the duty of handling the client.

One of the things to consider when developing a server is the client. A lot of the server architecture is complimented by the clients design. But given this task that restrains me to use **Telnet** as a client, I bump into my first major problem: “How do I properly distribute load between servers when I cannot pass the client connection to another server?”