# Para el servidor:

1° Correr el comando “top -d 0.1 | grep nghttpd >> server\_top\_no-push\_m100-w16.log” para guardar el consumo de cpu

Si no funciona, cambiar “0.1” por “0,1”(dependiendo del sistema podría usar puntos o comas)

Cambiar el nombre del archivo .log en función del tamaño de ventana, cantidad de streams y uso de push.

2° Correr el comando “nghttpd --no-tls -v -a 192.168.1.175 8084” para iniciar el servidor

Cambiar la ip y puerto del servidor a usar!

# Para el cliente:

1° Correr el comando “top -d 0.1 | grep nghttp >> client\_top\_no-push\_m100-w16.log”

Para guardar el consumo de cpu

Si no funciona, cambiar “0.1” por “0,1”(dependiendo del sistema podría usar puntos o comas)

Cambiar el nombre del archivo .log en función del tamaño de ventana, cantidad de streams y uso de push.

2° Correr el comando “python Client.py <tamañoventana> <cantidaddestreams>” para ejecutar las consultas del cliente al servidor

Cambiar en el archivo la ip y puerto del servidor a usar!

## Client.py

---------------------------------------------------------------------

from \_\_future\_\_ import print\_function  
import psutil  
import subprocess  
import re  
import sys  
  
  
#python client.py "<w>" "<m>"  
  
server=False  
ip = "192.168.1.175:8084"  
#w="16"  
w=sys.argv[1]  
print("w:",w)  
#m="01"  
m=sys.argv[2]  
print("m:",m)  
filename = "client/nghttp\_push\_w"+w+"\_m"+m+".log"  
with open(filename,'w') as filewriter:  
 for i in range(1,68):  
 print("running "+str(i))  
 filewriter.write("running "+str(i)+"\n")  
   
 nghttp = subprocess.Popen('nghttp http://'+ip+'/index.html -nas --no-dep -w '+w+' -W '+w, stdout=subprocess.PIPE, shell=True)  
 npid = nghttp.pid  
 pnghttp = psutil.Process(npid)  
 p=False  
 q = True  
 while nghttp.poll() is None:  
 for line in nghttp.stdout:  
 print(line)  
 filewriter.write(line)  
exit()

---------------------------------------------------------------------