



## Práctica

# Apache HTTP server benchmarking tool



Elaborada por: M. en C. Ukranio Coronilla

En esta clase nos interesa aprender a realizar un banco de pruebas para los servidores HTTP, esto mediante la simulación de múltiples usuarios que realizan solicitudes concurrentes.

Para obtener el desempeño del servidor utilizaremos la herramienta **ab** (Apache HTTP server benchmarking tool) cuya página web es:

<https://httpd.apache.org/docs/trunk/programs/ab.html>

Proceda a instalarla en su sistema LINUX con el comando:

```
sudo apt install apache2-utils
```

Y pruébela ejecutando en una terminal el servidor HTTP en el puerto 8080 y en otra terminal ejecute la herramienta ab pasándole como parámetro la URL del servidor:

```
ab http://localhost:8080/status
```

Obtendrá una salida del programa como la siguiente:

```
Server Software:
Server Hostname:      localhost
Server Port:          8080

Document Path:        /status
Document Length:      23 bytes

Concurrency Level:    1
Time taken for tests:  0.070 seconds
Complete requests:    1
Failed requests:       0
Total transferred:    118 bytes
HTML transferred:     23 bytes
Requests per second:  14.31 [#/sec] (mean)
Time per request:     69.862 [ms] (mean)
Time per request:     69.862 [ms] (mean, across all concurrent requests)
Transfer rate:        1.65 [Kbytes/sec] received
```

Las respuestas a los siguientes ejercicios se deberán adjuntar **todos** en un único archivo PDF que será enviado al canal de su equipo en teams.

## Ejercicio 1

Revise con ayuda del manual de la aplicación (véase man) o en la página web de **ab**, el significado de cada uno de los términos compilados en la siguiente tabla y descríbalos con sus propias palabras:

Item	Descripción
Complete requests	
Failed requests	
Total transferred	
HTML transferred	
Requests per second	
Time per request	
Transfer rate	

El programa ab puede recibir diversas opciones. Con ayuda del manual de la aplicación consense con su equipo para llenar la siguiente tabla:

Opción	Descripción
-n	
-c	
-k	
-v	
-p	
-T	

## Ejercicio 2

Levante el servidor al que se le agregó el endpoint `/searchtoken` y utilice las opciones adecuadas para que la aplicación ab pueda enviar al servidor una solicitud exitosa al endpoint **searchtoken** verificando con el parámetro **-v** que la solicitud se contesta de manera correcta por el servidor. Para enviar los datos desde el cliente tendrá que usar el parámetro **-p** y agregar un archivo con los datos para que se considere una solicitud POST. Adjuntar captura de pantalla probando que se resuelve correctamente la solicitud.

**Importante:** Es posible que la solicitud no envíe correctamente los datos y parezca que no hay ocurrencias en la búsqueda de la cadena. Esto se puede deber a que su editor está insertando un carácter salto de línea en el archivo con los datos. En estos casos se recomienda usar otro editor y

verificar que el tamaño del archivo en bytes corresponde exactamente con el número de caracteres insertados.

### Ejercicio 3

Como administradores del sistema nos interesa saber la cantidad de peticiones que puede manejar nuestro servidor en un intervalo de tiempo y el tiempo que tarda el servidor en generar y enviar la respuesta. En la herramienta ab el primero se obtiene de **Requests per second**, y el segundo de **Time per request** pero en el modo **mean, across all concurrent requests**, porque este representa el tiempo que cada cliente tuvo que esperar cuando había varios haciendo peticiones al mismo tiempo.

Para obtener estos dos parámetros de rendimiento supondremos un caso específico donde todos los clientes en internet acceden al endpoint **searchtoken** con búsquedas de una palabra en una cadena con 17576 tokens.

Dado que tenemos que someter al servidor a la máxima carga posible con la herramienta **ab** haremos mil solicitudes concurrentes (usar la opción -c) variando desde 1 hasta n de manera que el número de solicitudes por segundo haya llegado a su máximo posible. Envíe una tabla tabulando **n**, **Requests per second** y **Time per request**.

*Recomendación: Usar la LAP con las mejores características de hardware como servidor para estas pruebas.*

### Ejercicio 4

En el ejercicio anterior, cliente y servidor se encuentran en la misma computadora, lo cual hace que los tiempos de retraso en los mensajes sea casi cero, lo cual no sucede en la realidad debido a que existen tiempos de transferencia en las redes LAN o aún latencias mayores en Internet. Y también debido a las características de hardware de las tarjetas de red.

Para obtener datos más realistas repita el ejercicio anterior, pero ubicando al servidor y a la herramienta ab en computadoras distintas. Se recomienda conectar las computadoras en una LAN por medio de un celular utilizado como punto de acceso, debido a que los endpoints de la escuela filtran algunos puertos y para que no afecte a la medición el tráfico en la red por otras conexiones.

Después de llenar la tabla para este ejercicio, remarque cual es el número de solicitudes por segundo máximo que atiende el servidor, y cuánto tiempo en promedio debe esperar un cliente para obtener una respuesta. Por último, conteste la pregunta de cómo se verían afectados estos valores si el servidor ejecutándose en su LAP estuviera brindando el servicio no en una LAN, sino en Internet.

**Importante:** Probablemente sea necesario deshabilitar el firewall en las computadoras con Windows. En el caso que no se pueda realizar la conexión entre dos computadoras si se utiliza, se puede utilizar la terminal de Windows MS-DOS para correr el servidor instalando javac.