

Informe

El objetivo de este informe es mostrar algunas estadísticas de experimentos haciendo peticiones del cliente al servidor de forma concurrente y de forma no concurrente cuando éste (el servidor) no es concurrente y cuando sí lo es.

Para empezar, vamos a hacer los experimentos con el cliente haciendo peticiones de forma NO concurrente a un servidor NO concurrente.

- Haciendo 10 peticiones.

```
Recibiendo 10 solicitudes, el servidor se tarda 4338.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 0
```

- Haciendo 20 peticiones.

```
Recibiendo 20 solicitudes, el servidor se tarda 43094.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 1
```

- Haciendo 30 peticiones.

```
Recibiendo 30 solicitudes, el servidor se tarda 10330.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 0
```

- Haciendo 40 peticiones.

```
Recibiendo 40 solicitudes, el servidor se tarda 69673.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 2
```

- Haciendo 50 peticiones.

```
Recibiendo 50 solicitudes, el servidor se tarda 73417.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 2
```

- Haciendo 60 peticiones.

```
Recibiendo 60 solicitudes, el servidor se tarda 78505.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 2
```

- Haciendo 70 peticiones.

```
Recibiendo 70 solicitudes, el servidor se tarda 73367.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 1
```

- Haciendo 80 peticiones.

```
Recibiendo 80 solicitudes, el servidor se tarda 76476.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 1
```

- Haciendo 90 peticiones.

```
Recibiendo 90 solicitudes, el servidor se tarda 96920.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 1
```

- Haciendo 100 peticiones.

```
Recibiendo 100 solicitudes, el servidor se tarda 95832.0 milisegundos en responder.  
El número de solicitudes que NO fueron exitosas es: 0
```

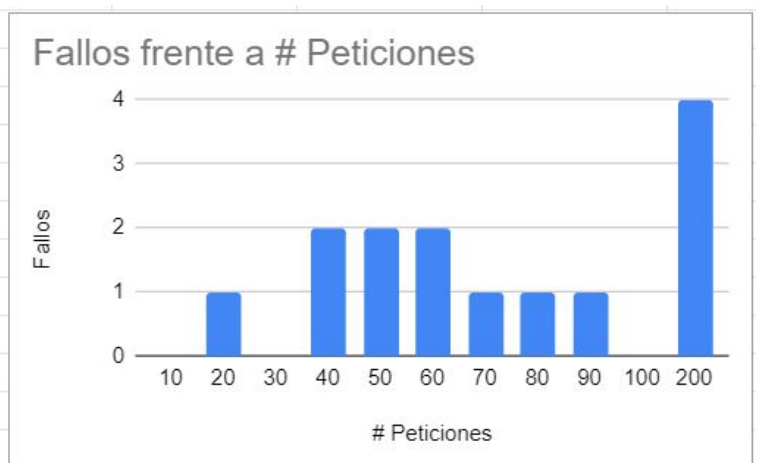
- Haciendo 200 peticiones.

Recibiendo 200 solicitudes, el servidor se tarda 416843.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 4

# Peticiones	Tiempo (ms)
10	4338
20	43094
30	10330
40	69673
50	73417
60	78505
70	73367
80	76476
90	96920
100	95832
200	416843



# Peticiones	Fallos
10	0
20	1
30	0
40	2
50	2
60	2
70	1
80	1
90	1
100	0
200	4



Ahora vamos a hacer los experimentos con el cliente haciendo peticiones de forma concurrente a un servidor NO concurrente.

- Haciendo 10 peticiones.

Recibiendo 10 solicitudes, el servidor se tarda 3460.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 4

- Haciendo 20 peticiones.

Recibiendo 20 solicitudes, el servidor se tarda 5752.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 12

- Haciendo 30 peticiones.

Recibiendo 30 solicitudes, el servidor se tarda 17453.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 18

- Haciendo 40 peticiones.

Recibiendo 40 solicitudes, el servidor se tarda 1599.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 29

- Haciendo 50 peticiones.

Recibiendo 50 solicitudes, el servidor se tarda 21744.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 30

- Haciendo 60 peticiones.

Recibiendo 60 solicitudes, el servidor se tarda 6585.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 47

- Haciendo 70 peticiones.

Recibiendo 70 solicitudes, el servidor se tarda 21745.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 53

- Haciendo 80 peticiones.

Recibiendo 80 solicitudes, el servidor se tarda 6866.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 63

- Haciendo 90 peticiones.

Recibiendo 90 solicitudes, el servidor se tarda 5059.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 57

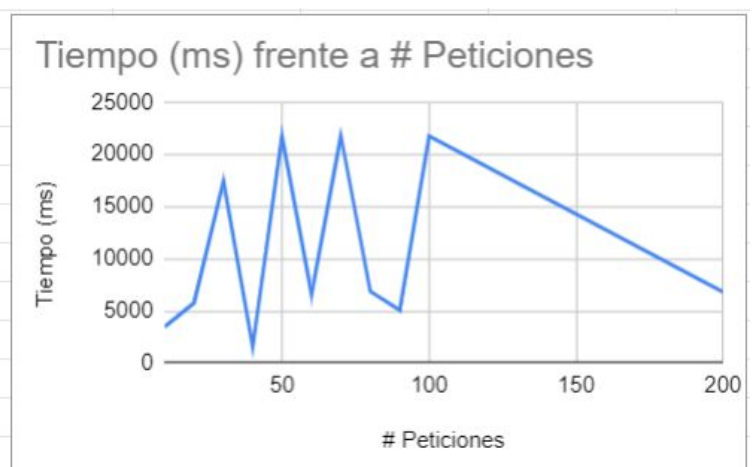
- Haciendo 100 peticiones.

Recibiendo 100 solicitudes, el servidor se tarda 21772.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 75

- Haciendo 200 peticiones.

Recibiendo 200 solicitudes, el servidor se tarda 6821.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 106

# Peticiones	Tiempo (ms)
10	3460
20	5752
30	17453
40	1599
50	21744
60	6585
70	21745
80	6866
90	5059
100	21772
200	6821



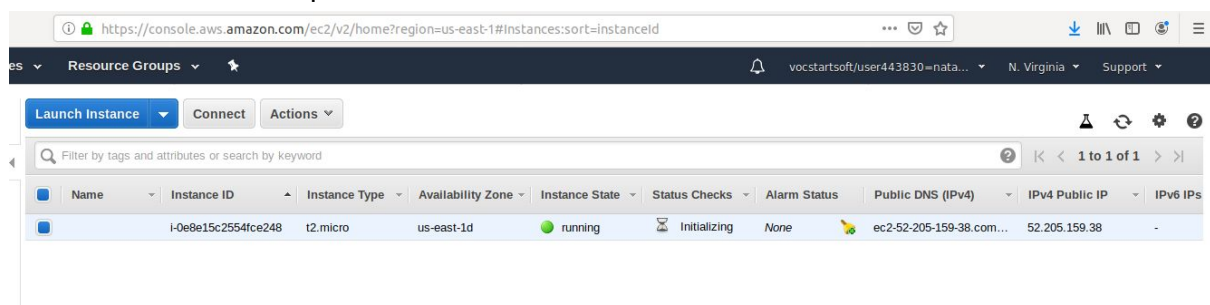
# Peticiones	Fallos
10	4
20	12
30	18
40	29
50	30
60	47
70	53
80	63
90	57
100	75
200	106



Ahora vamos a hacer pruebas nuevamente variando entre el cliente concurrente y el NO concurrente, pero ahora haremos las peticiones a un servidor concurrente.

Para poder hacer pruebas con el servidor concurrente hicimos lo siguiente:

1. Creamos la máquina virtual en AWS.



2. Nos conectamos a ella por ssh.

```
estudiante@linux-73:~/Downloads/AREP-TallerAWSCliente/cliente$ ls
Natalia.pem pom.xml src target
estudiante@linux-73:~/Downloads/AREP-TallerAWSCliente/cliente$ chmod 400 Natalia.pem
estudiante@linux-73:~/Downloads/AREP-TallerAWSCliente/cliente$ ssh -i "Natalia.pem" ec2-user@ec2-52-205-159-38.compute-1.amazonaws.com
The authenticity of host 'ec2-52-205-159-38.compute-1.amazonaws.com (52.205.159.38)' can't be established.
ECDSA key fingerprint is SHA256:Jq6/SgryOLExC+fqyzqLNdBi9x1JnBLpX5HN+9h6W2Q.
Are you sure you want to continue connecting (yes/no)? Y
Please type 'yes' or 'no': yes
Warning: Permanently added 'ec2-52-205-159-38.compute-1.amazonaws.com,52.205.159.38' (ECDSA) to the list of known hosts.

 _ _ _ _ _
| | ( _ | |
|_| \_|_|_| Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
2 package(s) needed for security, out of 6 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-47-186 ~]$
```

3. Con el comando 'java -version' podemos ver que originalmente está la versión 1.7.0 de java y debemos cambiarla a la versión 1.8.0. Para instalar la versión 1.8.0 usamos el comando 'sudo yum install java-1.8.0'.

```
[ec2-user@ip-172-31-47-186 ~]$ java -version
java version "1.7.0_231"
OpenJDK Runtime Environment (amzn-2.6.19.1.80.amzn1-x86_64 u231-b01)
OpenJDK 64-Bit Server VM (build 24.231-b01, mixed mode)
[ec2-user@ip-172-31-47-186 ~]$ sudo yum install java-1.8.0
```

- Luego de instalar la versión 1.8.0 de java, debemos remover la versión 1.7.0. Para eso usamos el comando 'sudo yum remove java-1.7.0-openjdk'. Luego, verificamos que sí se haya cambiado la versión.

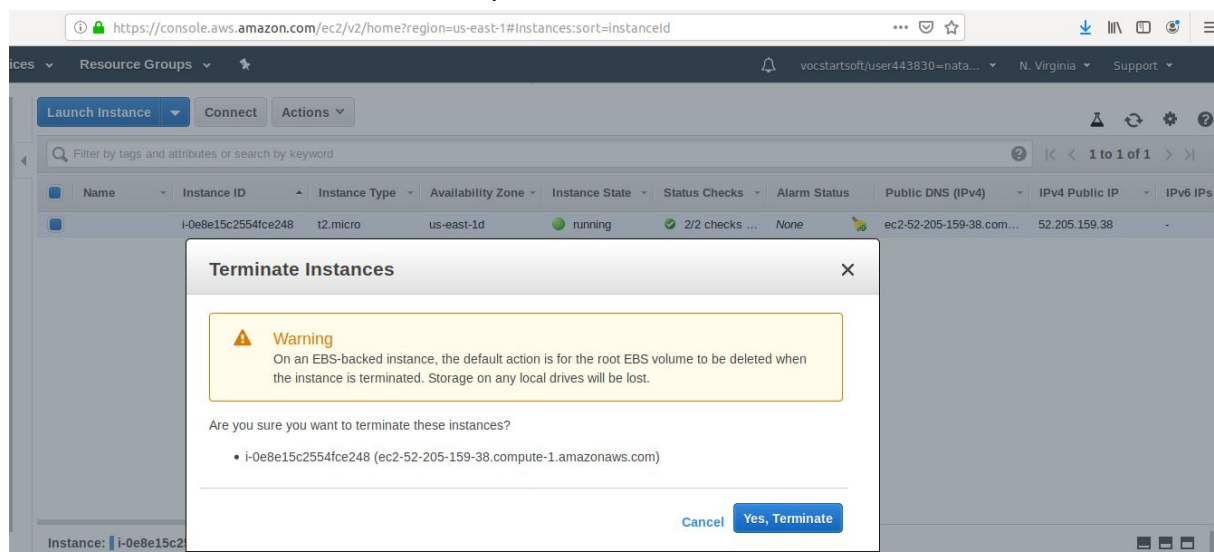
```
Removed:
  java-1.7.0-openjdk.x86_64 1:1.7.0.231-2.6.19.1.80.amzn1

Complete!
[ec2-user@ip-172-31-47-186 ~]$ java -version
openjdk version "1.8.0_222"
OpenJDK Runtime Environment (build 1.8.0_222-b10)
OpenJDK 64-Bit Server VM (build 25.222-b10, mixed mode)
[ec2-user@ip-172-31-47-186 ~]$
```

- Luego, nos conectamos por sftp para poder subir el jar del servidor concurrente a la máquina virtual.

```
estudiante@linux-73:~/Downloads/AREP-TallerAMSCliente/cliente$ sftp -l "Natalia.pem" ec2-user@ec2-52-205-159-38.compute-1.amazonaws.com
Connected to ec2-52-205-159-38.compute-1.amazonaws.com.
sftp> lcd target/
sftp> put c
classes/                               cliente-1.0-SNAPSHOT-jar-with-dependencies.jar      cliente-1.0-SNAPSHOT.jar
sftp> put cl
classes/                               cliente-1.0-SNAPSHOT-jar-with-dependencies.jar      cliente-1.0-SNAPSHOT.jar
sftp> put cliente
cliente-1.0-SNAPSHOT-jar-with-dependencies.jar      cliente-1.0-SNAPSHOT.jar
sftp> put cliente-1.0-SNAPSHOT-jar-with-dependencies.jar
Uploading cliente-1.0-SNAPSHOT-jar-with-dependencies.jar to /home/ec2-user/cliente-1.0-SNAPSHOT-jar-with-dependencies.jar
cliente-1.0-SNAPSHOT-jar-with-dependencies.jar      100% 3322    41.3KB/s   00:00
sftp>
```

- Al finalizar, eliminamos la máquina virtual.



Ahora mostramos los experimentos hechos con el servidor concurrente y el cliente NO concurrente:

- Haciendo 10 peticiones.

```
Recibiendo 10 solicitudes, el servidor se tarda 4764.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
```

- Haciendo 20 peticiones.

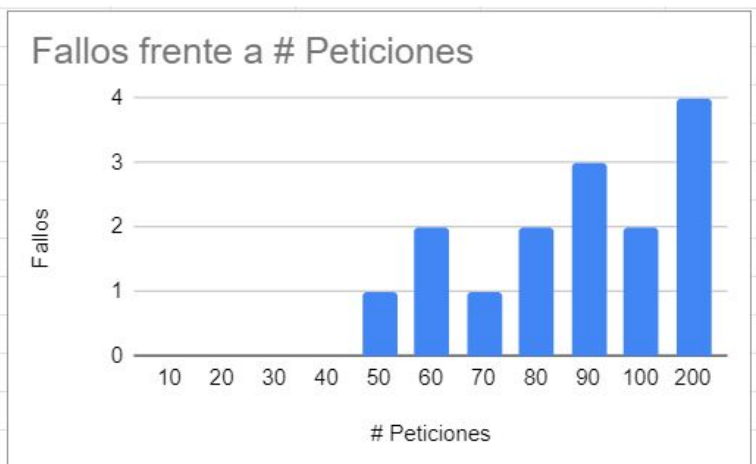
```
Recibiendo 20 solicitudes, el servidor se tarda 9298.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
```

- Haciendo 30 peticiones.
Recibiendo 30 solicitudes, el servidor se tarda 10966.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
- Haciendo 40 peticiones.
Recibiendo 40 solicitudes, el servidor se tarda 45257.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
- Haciendo 50 peticiones.
Recibiendo 50 solicitudes, el servidor se tarda 68343.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 1
- Haciendo 60 peticiones.
Recibiendo 60 solicitudes, el servidor se tarda 93903.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 2
- Haciendo 70 peticiones.
Recibiendo 70 solicitudes, el servidor se tarda 107801.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 1
- Haciendo 80 peticiones.
Recibiendo 80 solicitudes, el servidor se tarda 82472.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 2
- Haciendo 90 peticiones.
Recibiendo 90 solicitudes, el servidor se tarda 121498.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 3
- Haciendo 100 peticiones.
Recibiendo 100 solicitudes, el servidor se tarda 122168.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 2
- Haciendo 200 peticiones.
Recibiendo 200 solicitudes, el servidor se tarda 330084.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 4

# Peticiones	Tiempo (ms)
10	4764
20	9298
30	10966
40	45257
50	68343
60	93903
70	107801
80	82472
90	121498
100	122168
200	330084



# Peticiones	Fallos
10	0
20	0
30	0
40	0
50	1
60	2
70	1
80	2
90	3
100	2
200	4



Ahora mostramos los experimentos hechos con el servidor concurrente y el cliente concurrente:

- Haciendo 10 peticiones.
Recibiendo 10 solicitudes, el servidor se tarda 1524.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
- Haciendo 20 peticiones.
Recibiendo 20 solicitudes, el servidor se tarda 16200.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
- Haciendo 30 peticiones.
Recibiendo 30 solicitudes, el servidor se tarda 21821.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 2
- Haciendo 40 peticiones.
Recibiendo 40 solicitudes, el servidor se tarda 1718.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
- Haciendo 50 peticiones.
Recibiendo 50 solicitudes, el servidor se tarda 21872.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 10
- Haciendo 60 peticiones.
Recibiendo 60 solicitudes, el servidor se tarda 2169.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
- Haciendo 70 peticiones.
Recibiendo 70 solicitudes, el servidor se tarda 21818.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 10
- Haciendo 80 peticiones.
Recibiendo 80 solicitudes, el servidor se tarda 1897.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0
- Haciendo 90 peticiones.

Recibiendo 90 solicitudes, el servidor se tarda 21907.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 10

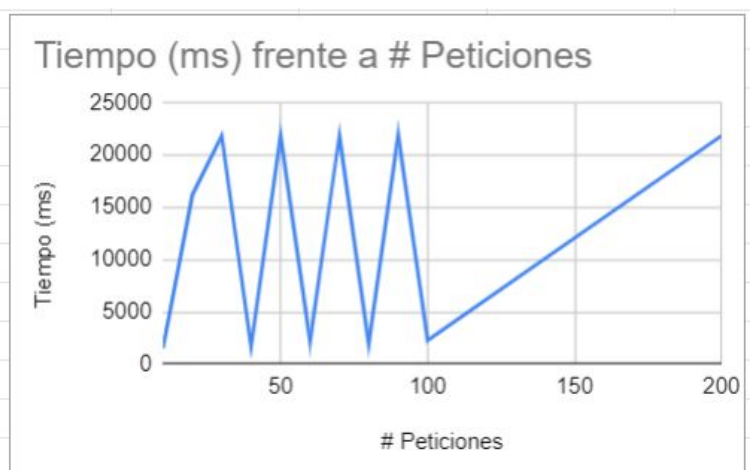
- Haciendo 100 peticiones.

Recibiendo 100 solicitudes, el servidor se tarda 2287.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 0

- Haciendo 200 peticiones.

Recibiendo 200 solicitudes, el servidor se tarda 21824.0 milisegundos en responder.
El número de solicitudes que NO fueron exitosas es: 24

# Peticiones	Tiempo (ms)
10	1524
20	16200
30	21821
40	1718
50	21872
60	2169
70	21818
80	1897
90	21907
100	2287
200	21824



Conclusiones

Como pudimos observar en las gráficas, hay una gran diferencia entre las peticiones concurrentes y las no concurrentes. Con la gráfica de tiempo vs. peticiones se puede observar más la diferencia, ya que cuando se hace concurrentemente varía mucho el tiempo que se demora en hacer todas las peticiones.

También hay una diferencia entre las peticiones al servidor concurrente y al no concurrente. Claramente notamos que cuando el servidor no era concurrente, había un fallo en las peticiones más o menos del 80% y esto se debe a que el servidor solo recibe de a una petición y no puede atender a muchas otras.