



TESTS DE RENDIMIENTO

Grupo 21












Contenido

1.	Descripción del ordenador	2
1.1.	Descripción de máquina virtual.....	2
1.2.	Descripción de máquina física.....	3
2.	Casos de uso e informes.....	3
2.1.1.	Casos de uso	3
2.1.2.	Pruebas realizadas.....	3
2.1.3.	Conclusiones.....	5
2.1.4.	Casos de uso	5
2.1.5.	Pruebas realizadas.....	6
2.1.6.	Conclusiones.....	8
2.1.7.	Casos de uso	10
2.1.8.	Pruebas realizadas.....	11
2.1.9.	Conclusiones.....	13
2.1.10.	Casos de uso	13
2.1.11.	Pruebas realizadas.....	14
2.1.12.	Conclusiones.....	16
2.1.13.	Casos de uso	17
2.1.14.	Pruebas realizadas.....	17
2.1.15.	Conclusiones.....	19
2.1.16.	Casos de uso	20
2.1.17.	Pruebas realizadas.....	20
2.1.18.	Conclusiones.....	22
2.1.19.	Casos de uso	22
2.1.20.	Pruebas realizadas.....	23
2.1.21.	Conclusiones.....	25

1. Descripción del ordenador

Máquina virtual ejecutada con Oracle VM VirtualBox.

1.1. Descripción de máquina virtual

 General	
Nombre:	Pre-production-1.18.2
Sistema operativo:	Windows XP (64-bit)
Ubicación de archivo de preferencias:	/Users/elenamolina/Downloads/Pre-production-1.18.2
 Sistema	
Memoria base:	2048 MB
Procesadores:	2
Orden de arranque:	Disco duro, Óptica
Aceleración:	VT-x/AMD-V, Paginación anidada, PAE/NX
 Pantalla	
Memoria de vídeo:	16 MB
Controlador gráfico:	VBoxVGA
Aceleración:	Vídeo 2D, 3D
Servidor de escritorio remoto:	Inhabilitado
Grabando:	Inhabilitado
 Almacenamiento	
Controlador:	IDE
IDE primario maestro:	Pre-production-1.18-2.vdi (Normal, 30,69 GB)
IDE secundario maestro:	[Unidad óptica] Vacío
 Audio	
Controlador de anfitrión:	CoreAudio
Controlador:	ICH AC97
 Red	
Adaptador 1:	Intel PRO/1000 MT Desktop (NAT)
 USB	
Controlador USB:	OHCI
Filtros de dispositivos:	0 (0 activo)
 Carpetas compartidas	
Carpetas compartidas:	1
 Descripción	
Ninguno	

1.2. Descripción de máquina física

MacBook Pro

Información del hardware:

Nombre del modelo: MacBook Pro
Identificador del modelo: MacBookPro15,2
Nombre del procesador: Intel Core i5
Velocidad del procesador: 2,3 GHz
Cantidad de procesadores: 1
Cantidad total de núcleos: 4
Caché de nivel 2 (por núcleo): 256 KB
Caché de nivel 3: 6 MB
Memoria: 8 GB
Versión de la ROM de arranque: 220.250.366.0.0 (iBridge: 16.16.4507.0.0,0)
Número de serie (sistema): C02XX0YXJHCD
UUID de hardware: 75267765-05EE-5CCD-8B82-E26B73F0FD84

Cabe destacar que se realizan todas las peticiones con el protocolo HTTPS.

2. Casos de uso e informes

2.1 An actor who is not authenticated must be able to register to the system as a manager (RF 11.1)

El archivo correspondiente a este test es *RegisterManagerTestPlan.jmx*.

2.1.1. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Acceder al menú Register y pulsar en el botón “Register a Manager”

2.1.2. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	2000	18	15	31	5	270	0.00%	39.4/sec	283.6
/welcome/...	1000	10	8	19	5	89	0.00%	21.5/sec	78.3
TOTAL	3000	15	12	28	5	270	0.00%	58.4/sec	351.7

No hay errores HTTP y el tiempo medio de espera total es de 50ms = 0,05s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 31ms = 0,03s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

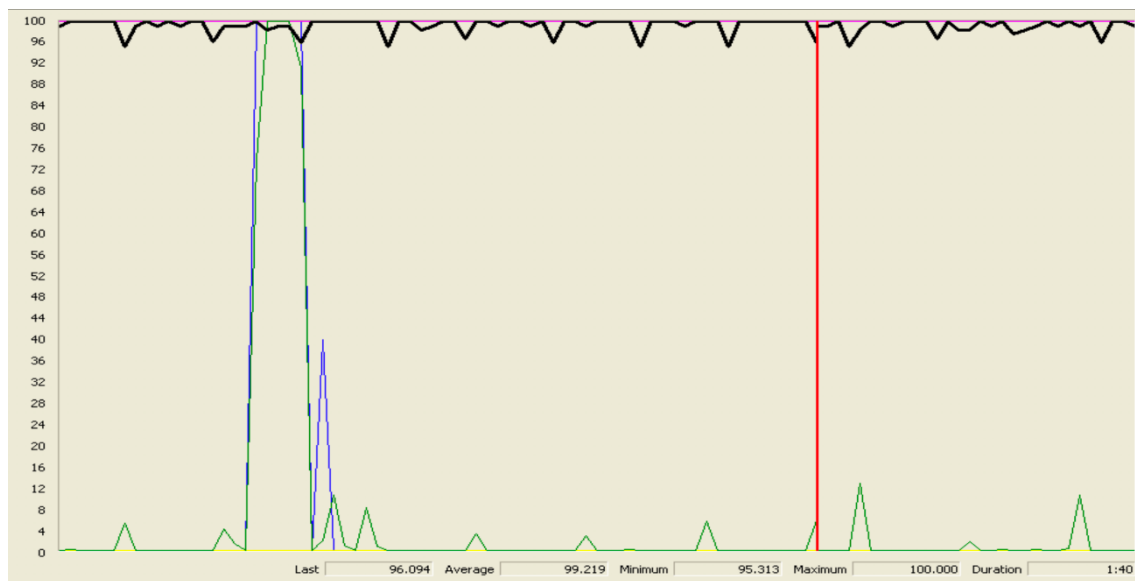
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	4000	29	17	52	5	1451	0.00%	86.1/sec	621.4
/welcome/...	2000	15	11	27	5	216	0.00%	44.6/sec	163.1
TOTAL	6000	24	15	44	5	1451	0.00%	125.7/sec	757.7

No hay errores HTTP y el tiempo medio de espera total es de 79ms = 0,07s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 52ms = 0,052s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	2465	64	27	145	5	2981	0.00%	100.4/sec	724.5
/welcome/...	1115	52	18	147	5	1058	0.00%	51.7/sec	188.8
TOTAL	3580	60	25	147	5	2981	0.00%	145.8/sec	890.4

No hay errores HTTP y el tiempo medio de espera total es de 292ms = 0,29.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de $147 = 0.14s$.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	2931	6588	26	239	1	116161	15.90%	17.1/sec	108.4
/welcomef...	1199	8129	20	292	5	116201	7.01%	8.7/sec	30.9
TOTAL	4130	7035	24	255	1	116201	13.32%	24.1/sec	133.2

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.3. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.2. An actor who is not authenticated must be able to register to the system as a client (RF 11.1)

El archivo correspondiente a este test es *RegisterClientTestPlan.jmx*.

2.1.4. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

2. Acceder al menú Register y pulsar el botón “Register a Client”

2.1.5. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	2000	157	32	276	7	3715	0.00%	37.8/sec	276.5
/welcome/...	1000	39	17	79	7	1231	0.00%	19.9/sec	72.5
TOTAL	3000	117	27	176	7	3715	0.00%	55.4/sec	337.5

No hay errores HTTP y el tiempo medio de espera total es de 355ms = 0,355s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 276ms = 0,027s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

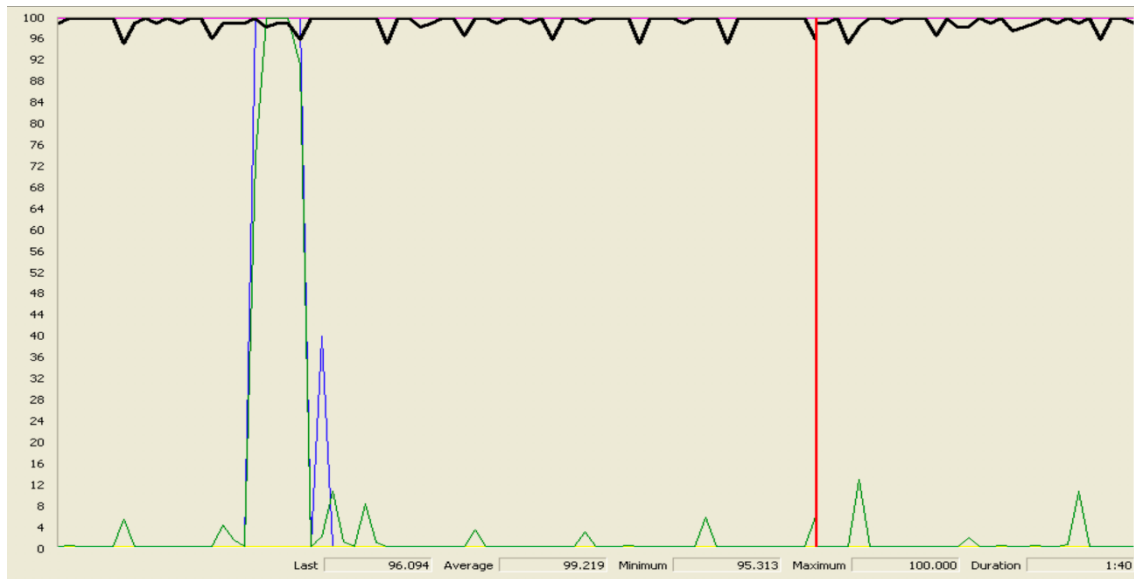
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	4000	63	27	120	7	2347	0.00%	81.5/sec	597.3
/welcome/...	2000	36	18	75	6	682	0.00%	41.9/sec	153.0
TOTAL	6000	54	24	104	6	2347	0.00%	118.6/sec	723.8

No hay errores HTTP y el tiempo medio de espera total es de 195 = 0,19s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 120 = 0,12s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	4000	59	31	129	6	1714	0.00%	102.8/sec	753.5
/welcome/...	2000	46	19	95	5	1224	0.00%	54.3/sec	198.4
TOTAL	6000	55	27	118	5	1714	0.00%	151.1/sec	922.3

No hay errores HTTP y el tiempo medio de espera total es de $224 = 0,22$.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de $129\text{ms} = 0,12\text{s}$.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/register/a...	469	1379	1025	3270	1	6750	2.13%	56.6/sec	409.0
/welcome/...	74	1149	943	2427	14	3705	0.00%	14.2/sec	52.0
TOTAL	543	1347	988	3186	1	6750	1.84%	65.5/sec	441.4

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.6. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

2.1 A actor who is not authenticated must be able to list all the accepted clubs that are in the system, see their information, and navigate to his manager or to the list of their events, clubs and manager (RF 11.2, RF 11.3, RF 11.4)

El archivo correspondiente a este test es *List-event-club-managerTestPlan.jmx*.

2.1.7. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

3. Acceder al menú club/managers/events y dar click a List clubs

2.1.8. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/	600	10	10	13	4	27	0.00%	11.6/sec	43.2
/club/list.do	600	11	11	16	7	33	0.00%	11.6/sec	61.7
/actor/manager/list.do	581	14	11	20	5	133	0.00%	11.4/sec	55.9
/event/list.do	500	23	20	35	11	107	0.00%	11.9/sec	76.2
TOTAL	2281	14	12	23	4	133	0.00%	40.7/sec	204.8

No hay errores HTTP y el tiempo medio de espera total es de 231ms = 0,23s.

El mayor tiempo de espera se da al realizarse la operación de logueo siendo de 0,78ms = 0,07s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

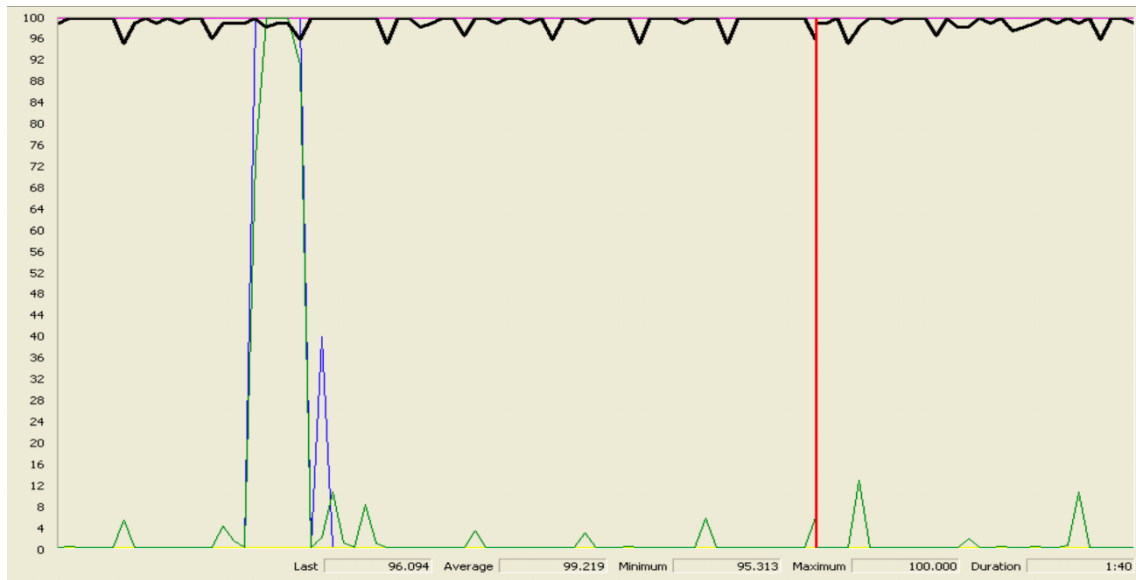
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1055	17	12	34	4	226	0.00%	19.8/sec	73.7
/clublist.do	1000	22	13	37	6	334	0.00%	22.8/sec	120.5
/actormanagerlist.do	1000	26	12	36	6	518	0.00%	21.8/sec	107.0
/eventlist.do	1000	107	32	210	11	2035	0.00%	21.8/sec	138.2
TOTAL	4055	43	15	73	4	2035	0.00%	76.0/sec	386.1

No hay errores HTTP y el tiempo medio de espera total es de 423ms = 0,423s.

El mayor tiempo de espera se da al realizarse la operación asignación siendo de 210ms = 0,2s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	2511	110	17	278	5	2892	0.00%	29.7/sec	110.8
/clublist.do	2403	68	16	151	6	2065	0.00%	30.1/sec	160.5
/actormanagerlist.do	2400	41	15	71	5	2199	0.00%	31.3/sec	154.0
/eventlist.do	2368	255	69	704	11	4284	0.00%	30.4/sec	195.3
TOTAL	9682	118	23	268	5	4284	0.00%	114.5/sec	582.1

No hay errores HTTP y el tiempo medio de espera total es de 1693ms = 1,693s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 704 = 0,7s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 450 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/	4500	257	75	715	4	4699	0.00%	42.9/sec	160.3
/clublist.do	4500	145	46	308	5	5056	0.00%	42.7/sec	227.9
/actor/managerlist.do	4500	251	50	614	4	6957	0.00%	41.9/sec	206.2
/eventlist.do	4500	513	208	1460	8	6117	0.00%	41.7/sec	267.6
TOTAL	18000	292	73	798	4	6957	0.00%	154.6/sec	789.1

No hay errores HTTP y el tiempo medio de espera total es de 5578ms = 5,5s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1460ms = 1,4.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 550 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/	4500	257	75	715	4	4699	0.00%	42.9/sec	160.3
/clublist.do	4500	145	46	308	5	5056	0.00%	42.7/sec	227.9
/actor/managerlist.do	4500	251	50	614	4	6957	0.00%	41.9/sec	206.2
/eventlist.do	4500	513	208	1460	8	6117	0.00%	41.7/sec	267.6
TOTAL	18000	292	73	798	4	6957	0.00%	154.6/sec	789.1

No hay errores HTTP y el tiempo medio de espera total es de 15578ms = 15,5s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 6957ms = 6,9.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

4.1. An actor who is authenticated must be able to edit his or her personal data. (RF 12.2)

El archivo correspondiente a este test es *EditPersonalDataTestPlan.jmx*.

2.1.9. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú de el perfil y pulsar “Edit Personal Data”

2.1.10. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	1000	17	8	30	4	315	0.00%	16.6/sec	85.4
/j_spring...	1000	49	30	88	11	714	0.00%	16.4/sec	86.3
/	1000	28	22	50	5	600	0.00%	16.4/sec	82.7
/actor/edi...	2000	22	16	37	6	429	0.00%	31.8/sec	246.0
TOTAL	5000	28	18	46	4	714	0.00%	75.8/sec	467.8

No hay errores HTTP y el tiempo medio de espera total es de 205ms = 0,20.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 88 ms = 0,08 segs

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

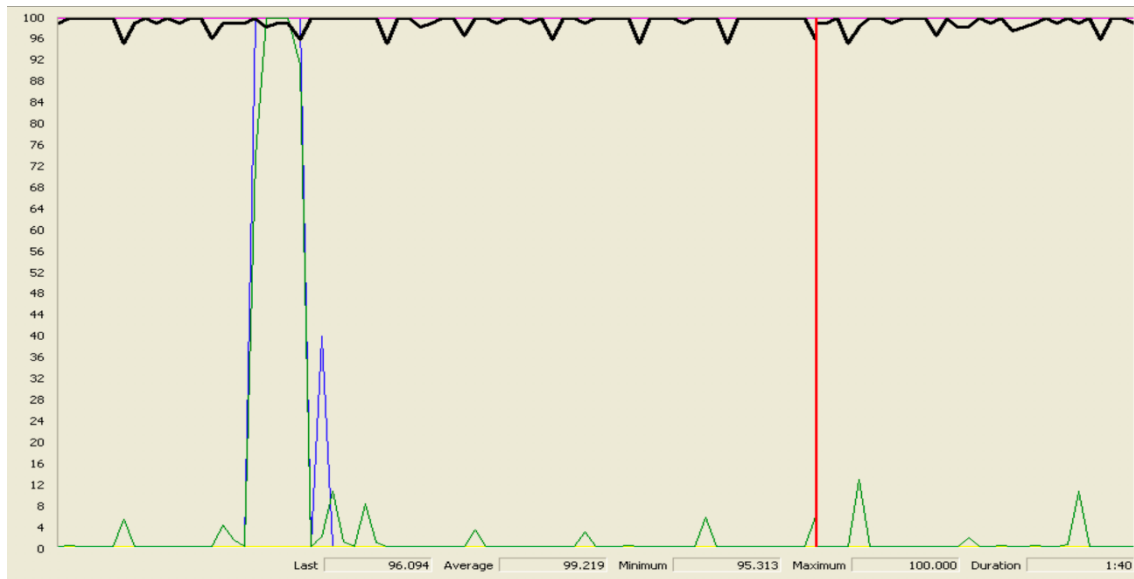
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	3000	109	26	206	4	4741	0.00%	15.6/sec	80.3
/j_spring...	3000	169	69	383	9	3745	0.00%	15.6/sec	82.2
/	3000	84	35	175	5	2771	0.00%	15.6/sec	78.7
/actor/edi...	6000	94	40	195	6	4044	0.00%	31.0/sec	239.6
TOTAL	15000	110	39	235	4	4741	0.00%	76.2/sec	471.3

No hay errores HTTP y el tiempo medio de espera total es de 959ms = 0,95 s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 383 ms = 0,38 segs.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	5500	105	24	198	4	7761	0.00%	16.8/sec	86.6
/j_spring...	5500	198	68	433	9	7734	0.00%	16.8/sec	88.7
/	5500	96	32	191	5	6618	0.00%	16.8/sec	84.8
/factor/edi...	11000	102	36	203	6	6973	0.00%	33.5/sec	259.1
TOTAL	27500	121	36	246	4	7761	0.00%	83.0/sec	513.1

No hay errores HTTP y el tiempo medio de espera total es de 1025ms = 1,025.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 433 ms = 0,43 segs

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	8500	93	23	184	4	7761	0.00%	19.5/sec	100.5
/j_spring...	8500	189	69	450	8	7734	0.00%	19.5/sec	102.7
/	8500	88	31	178	5	6618	0.00%	19.5/sec	98.2
/factor/edi...	17000	98	36	204	6	6973	0.00%	38.8/sec	299.9
TOTAL	42500	113	36	244	4	7761	0.00%	96.2/sec	595.4

No hay errores HTTP y el tiempo medio de espera total es de 1016ms = 1,016s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 450ms = 0,4.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 5:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	15307	271	39	665	4	16504	0.31%	19.4/sec	99.8
/j_spring...	15260	553	122	1484	6	20981	0.07%	19.4/sec	102.3
/	15250	268	47	638	4	19376	0.02%	19.4/sec	97.7
/actor/edi...	30480	276	55	667	6	30611	0.06%	38.6/sec	298.7
TOTAL	76297	329	60	812	4	30611	0.10%	96.3/sec	595.4

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

"Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated"

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *"Constant Delay Offset"* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.11. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated must be able to manage his or her social profiles, who include creating ,updating and deleting them (RF 12.3)

El archivo correspondiente a este test es *ManageSocialProfileTestPlan.jmx*.

2.1.12. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse
2. Acceder al menú del perfil y pulsar “List Social Profiles”
3. Pulsar el botón de créate, rellnar los campos y pulsar save
4. Pulsar el botón edit y save al escribir un cambio
5. Pulsar el botón display del social profile creado
6. Pulsar el botón de eliminar en la vista edit

2.1.13. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	1000	104	14	162	6	1969	0.00%	6.3/sec	32.4
/j_spring...	1000	121	40	261	12	2587	0.00%	6.3/sec	33.0
/	1000	68	30	124	6	1662	0.00%	6.3/sec	31.7
/socialPr...	2000	52	18	84	7	1735	0.00%	12.2/sec	66.9
/socialPr...	1000	39	20	68	7	744	0.00%	6.2/sec	37.0
/socialPr...	1000	82	38	153	12	1418	0.00%	6.2/sec	35.8
/socialPr...	2000	43	27	86	9	722	0.00%	12.1/sec	75.4
/socialPr...	1000	73	41	150	9	1200	0.00%	6.2/sec	35.5
TOTAL	10000	68	29	120	6	2587	0.00%	57.0/sec	320.8

No hay errores HTTP y el tiempo medio de espera total es de 1088ms = 1,08s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 261ms = 0,261 s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

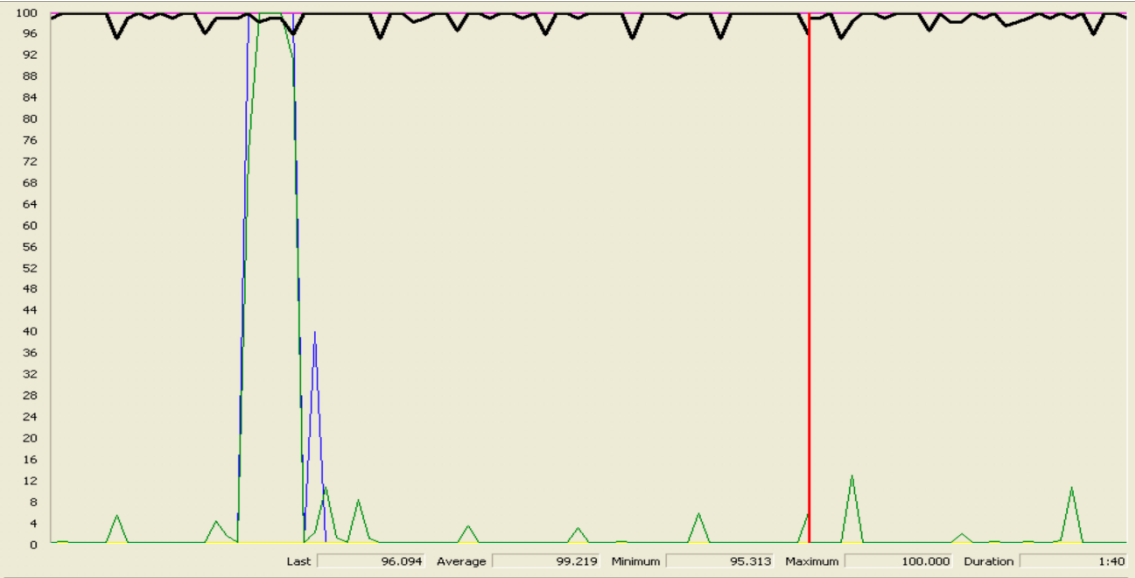
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	3000	240	47	629	5	6585	0.00%	6.1/sec	31.3
/j_spring...	3000	363	129	989	12	6676	0.00%	6.1/sec	32.0
/	3000	164	58	430	6	6247	0.00%	6.1/sec	30.6
/socialPr...	6000	170	47	444	6	9027	0.00%	12.1/sec	66.3
/socialPr...	3000	163	49	429	7	4261	0.00%	6.1/sec	36.0
/socialPr...	3000	374	113	979	12	11499	0.00%	6.1/sec	34.9
/socialPr...	6000	205	64	495	9	6574	0.00%	12.0/sec	75.0
/socialPr...	3000	376	122	1008	9	6781	0.00%	6.1/sec	34.9
TOTAL	30000	243	68	636	5	11499	0.00%	58.9/sec	332.3

No hay errores HTTP y el tiempo medio de espera total es de 5403ms = 5,4s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 989ms = 0,9s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	2500	337	111	929	4	6832	0.00%	13.2/sec	67.8
/j_spring...	2500	655	363	1744	9	6740	0.00%	13.1/sec	69.4
/	2500	324	125	846	5	5356	0.00%	13.1/sec	66.3
/socialPr...	5000	281	84	749	6	6941	0.00%	25.9/sec	142.6
/socialPr...	2500	258	75	741	6	5857	0.00%	13.2/sec	78.2
/socialPr...	2500	611	339	1648	12	5821	0.00%	13.2/sec	76.2
/socialPr...	5000	340	128	880	8	7923	0.00%	26.0/sec	162.6
/socialPr...	2500	648	339	1724	9	6514	0.00%	13.2/sec	76.3
TOTAL	25000	408	154	1125	4	7923	0.00%	124.1/sec	701.0

No hay errores HTTP y el tiempo medio de espera total es de 9261ms = 9,2s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1744ms = 1,7s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	5500	188	28	550	3	6832	0.00%	10.4/sec	53.5
/j_spring...	5500	359	92	1081	7	6740	0.00%	10.4/sec	54.8
/	5500	178	38	506	4	5356	0.00%	10.4/sec	52.4
/socialPr...	11000	151	28	401	5	6941	0.00%	20.6/sec	113.6
/socialPr...	5500	137	27	365	5	5857	0.00%	10.4/sec	61.8
/socialPr...	5500	342	98	955	10	5821	0.00%	10.4/sec	59.9
/socialPr...	11000	196	49	518	7	7923	0.00%	20.7/sec	129.0
/socialPr...	5500	378	107	1074	9	6514	0.00%	10.4/sec	60.0
TOTAL	55000	228	48	634	3	7923	0.00%	101.5/sec	573.3

No hay errores HTTP y el tiempo medio de espera total es de 10450ms = 10,4s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1081ms = 1,081s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 5:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/l...	8625	327	57	816	0	20277	0.07%	9.8/sec	50.2
/j_spring...	8619	652	175	1461	1	47163	0.17%	9.8/sec	51.6
/	8604	311	64	656	1	31169	0.07%	9.8/sec	49.2
/socialPr...	17183	293	52	608	0	29589	0.05%	19.4/sec	106.9
/socialPr...	8591	292	49	597	1	40534	0.07%	9.7/sec	57.9
/socialPr...	8584	617	179	1310	9	42675	0.01%	9.8/sec	56.4
/socialPr...	17163	331	76	720	1	31829	0.05%	19.5/sec	121.5
/socialPr...	8583	664	201	1533	3	28862	0.03%	9.8/sec	56.3
TOTAL	85952	411	85	915	0	47163	0.06%	96.4/sec	544.3

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.14. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated must be able to exchange messages with other actors and manage them. Manage his or her messages boxes, except for the system boxes. . An actor who is authenticated as an administrator must be able to broadcast a message to all of the actors of the system (RF 12.4, RF 12.5 and RF 15.4)

El archivo correspondiente a este test es *ManageBoxTestPlan.jmx*.

2.1.15. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

7. Loguearse
8. Acceder al menú de “My boxes” y pulsar “Exchange message”
9. Rellenar formulario y pulsar botón save
10. Pulsar “Create box”
11. Rellenar el formulario y pulsar el botón save
12. Acceder al menú de “My Boxes” y pulsar “Exchange broadcast message”
13. Rellenar el formulario y pulsar el botón save.

2.1.16. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/logi...	1000	32	11	54	5	1137	0.00%	5.3/sec	27.1
/j_spring_se...	1000	60	23	128	5	1262	0.00%	5.3/sec	27.7
/	1000	28	16	52	5	718	0.00%	5.3/sec	26.6
/message/a...	1000	54	36	95	13	1230	0.00%	5.2/sec	34.8
/message/a...	1000	126	75	240	26	1495	0.00%	5.2/sec	33.5
/box/actor/list...	3000	50	25	72	10	3973	0.00%	14.7/sec	91.9
/box/actor/cr...	1000	23	18	40	8	229	0.00%	5.2/sec	30.9
/box/actor/ed...	1000	33	29	52	13	135	0.00%	5.2/sec	31.4
/message/a...	1000	26	16	41	7	928	0.00%	5.2/sec	26.1
/message/a...	2000	220	95	534	12	2975	0.00%	10.3/sec	65.7
TOTAL	13000	75	28	147	5	3973	0.00%	62.2/sec	368.0

No hay errores HTTP y el tiempo medio de espera total es de 1308ms = 1,3s.

El mayor tiempo de espera se da al realizarse la operación de mensaje broadcast siendo de 534ms = 0,5s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

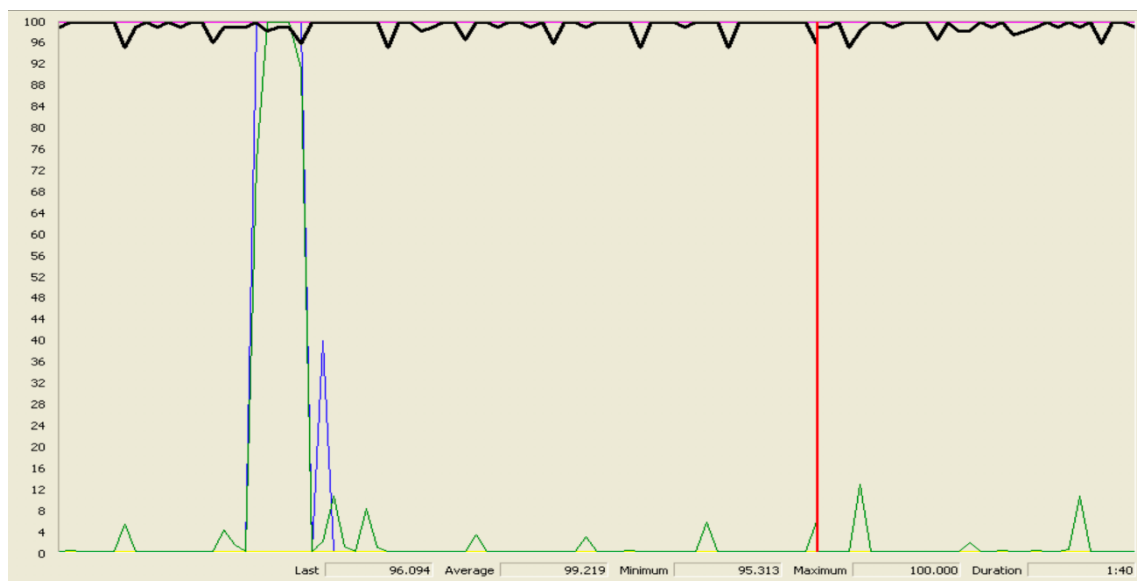
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/loge...	3000	79	17	166	4	5855	0.00%	2.4/sec	12.2
/j_spring_se...	3000	140	41	339	5	8228	0.00%	2.4/sec	12.5
/	3000	67	21	140	5	4540	0.00%	2.4/sec	11.9
/message/a...	3000	110	42	192	10	6642	0.00%	2.4/sec	15.7
/message/a...	3000	286	89	593	23	12730	0.00%	2.4/sec	15.2
/boxfactor/list...	9000	124	31	199	8	14425	0.00%	7.0/sec	43.9
/boxfactor/cr...	3000	103	24	112	8	6616	0.00%	2.4/sec	14.0
/boxfactor/d...	3000	123	37	139	12	7447	0.00%	2.4/sec	14.3
/message/a...	3000	129	22	178	7	11248	0.00%	2.4/sec	11.9
/message/a...	6000	302	101	708	12	11533	0.00%	4.7/sec	30.1
TOTAL	39000	155	38	294	4	14425	0.00%	30.4/sec	179.9

No hay errores HTTP y el tiempo medio de espera total es de 2766ms = 2,766s.

El mayor tiempo de espera se da al realizarse la operación de mensaje bvroadcast siendo de 708ms = 0,7.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/loge...	5500	528	43	1482	4	21000	0.00%	1.7/sec	8.8
/j_spring_se...	5500	1007	164	3042	5	21389	0.00%	1.7/sec	9.0
/	5500	504	50	1412	5	17724	0.00%	1.7/sec	8.6
/message/a...	5500	550	79	1466	10	18065	0.00%	1.7/sec	11.4
/message/a...	5500	1088	237	3192	23	26944	0.00%	1.7/sec	11.0
/boxfactor/list...	16500	540	59	1530	8	22376	0.00%	5.1/sec	31.8
/boxfactor/cr...	5500	518	46	1420	8	21115	0.00%	1.7/sec	10.1
/boxfactor/d...	5500	561	60	1571	12	22585	0.00%	1.7/sec	10.3
/message/a...	5500	535	45	1591	7	16262	0.00%	1.7/sec	8.6
/message/a...	11000	979	218	2888	12	21861	0.00%	3.4/sec	21.7
TOTAL	71500	682	89	2006	4	26944	0.00%	22.0/sec	130.6

No hay errores HTTP y el tiempo medio de espera total es de 33412ms = 33,4.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3042ms = 3,042s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/loge...	1245	1342	656	3118	0	20675	0.32%	7.8/sec	38.8
/j_spring_se...	1212	2554	1705	6082	0	21599	0.41%	7.7/sec	40.6
/	1194	1216	596	3038	5	17398	0.00%	7.6/sec	38.6
/message/a...	1181	1455	757	3419	0	23960	0.25%	7.6/sec	50.4
/message/a...	1154	2779	1782	6070	0	22434	0.26%	7.5/sec	48.2
/boxfactor/list...	3189	1352	675	3299	1	22301	0.38%	21.0/sec	130.8
/boxfactor/cr...	1120	1287	589	3335	0	17481	0.45%	7.5/sec	44.2
/boxfactor/d...	1099	1338	623	3343	0	18500	0.18%	7.4/sec	44.8
/message/a...	1065	1361	593	3375	0	19972	0.38%	7.4/sec	37.0
/message/a...	2040	2277	1343	5358	0	31565	0.59%	14.2/sec	90.5
TOTAL	14499	1687	832	4152	0	31565	0.34%	91.0/sec	536.1

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.17. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

MANAGER

3.1. An actor who is authenticated as manager must be able to manage their clubs, which includes listing, showing, creating, updating, and deleting them.

A club in final mode cannot be update or deleted. When a club is displayed, the manager can see the score of this club(RF 13.1)

El archivo correspondiente a este test es *ManageClubManagerTestPlan.jmx*.

2.1.18. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

14. Loguearse como manager
15. Pulsar en el menú “My clubs” , “create club”
16. Rellenar formulario y pulsar botón save
17. Pulsar el botón Edit del club creado
18. Rellenar formulario y pulsar botón save
19. Pulsar el botón Edit display y volver
20. Pulsar el botón Edit del club
21. Pulsar botón Delete

2.1.19. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1000	43	21	90	5	949	0.00%	5.0/sec	18.5
/security/login.do	1000	31	11	55	5	939	0.00%	5.0/sec	19.5
/j_spring_security_check	1000	65	24	110	10	2302	0.00%	5.0/sec	23.4
/clubmanager/create.do	1980	102	37	239	6	2471	0.00%	9.8/sec	66.2
/clubmanager/edit.do	1815	110	64	200	20	2433	0.00%	9.3/sec	73.1
/clubmanager/edit.do?clubId=36557	1801	30	20	50	9	845	0.00%	9.4/sec	57.5
/factormanager/listClubs.do	900	79	55	142	12	1099	0.00%	5.2/sec	39.4
TOTAL	9496	70	31	153	5	2471	0.00%	45.2/sec	275.4

No hay errores HTTP y el tiempo medio de espera total es de 886ms = 0,86s.

El mayor tiempo de espera se da al realizarse la operación de registro siendo de 110ms = 0,11s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios

- Loop de 10 iteraciones

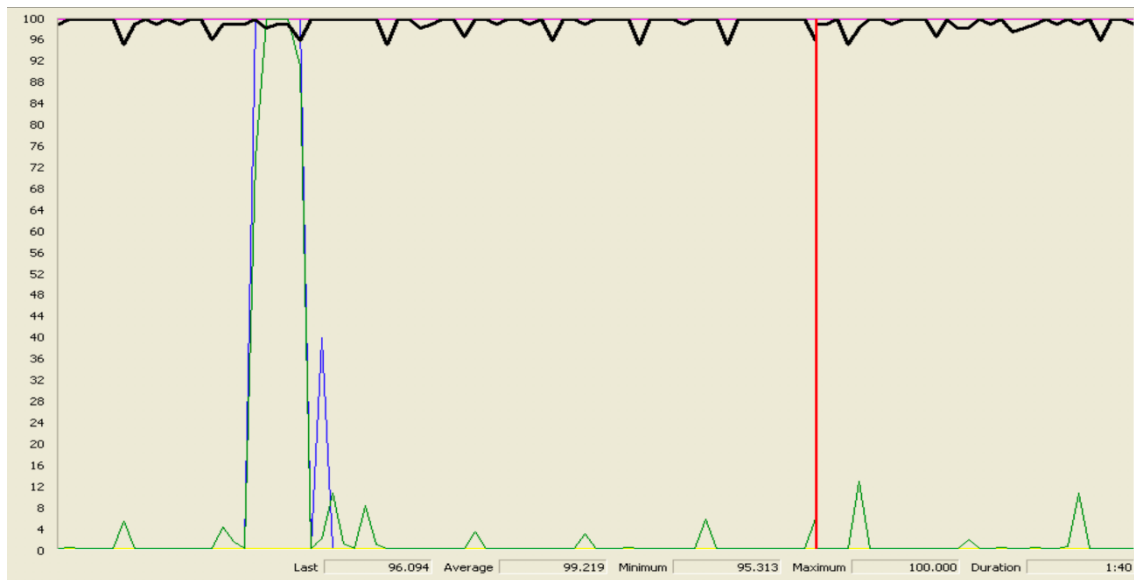
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	647	1020	226	3057	6	15925	0.00%	6.0/sec	22.2
/security/login.do	624	1062	281	3204	6	17718	0.00%	5.8/sec	22.7
/j_spring_security_check	610	2069	1003	5348	14	27999	0.00%	5.7/sec	26.5
/club/manager/create.do	1166	2301	898	6366	7	23226	0.00%	11.1/sec	75.2
/club/manager/edit.do	1045	3549	2181	9015	45	27377	0.00%	10.3/sec	81.5
/club/manager/edit.do?clubid=36557	1006	1690	647	4767	19	18018	0.00%	10.2/sec	62.6
/actor/manager/listClubs.do	463	1527	629	4429	73	14779	0.00%	5.0/sec	38.1
TOTAL	5561	2047	831	5609	6	27999	0.00%	51.1/sec	308.0

No hay errores HTTP y el tiempo medio de espera total es de 36203ms = 36,203s.

El mayor tiempo de espera se da al realizarse la operación de edición de 9015ms = 9,015s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	549	1635	233	5095	5	32146	0.00%	5.4/sec	20.0
/security/login.do	520	1663	582	3045	7	28867	0.00%	5.3/sec	20.6
/j_spring_security_check	495	2950	1557	6946	13	32511	0.00%	5.0/sec	23.4
/club/manager/create.do	888	6305	2351	18305	19	70404	0.00%	9.3/sec	62.5
/club/manager/edit.do	691	9867	5333	23740	313	74125	0.00%	7.7/sec	60.6
/club/manager/edit.do?clubid=36557	637	4752	1749	12957	212	47461	0.00%	7.3/sec	44.9
/actor/manager/listClubs.do	274	3543	1539	8472	300	35316	0.00%	3.8/sec	28.8
TOTAL	4054	4844	1759	13973	5	74125	0.00%	39.5/sec	232.5

No hay errores HTTP y el tiempo medio de espera total es de 78569ms = 78,596s.

El mayor tiempo de espera se da al realizarse la operación de creación siendo de 18305ms = 18,305s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 500 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	450	307	58	671	5	5440	0.00%	49.8/sec	186.1
/securitylogin.do	439	1264	746	3180	9	6379	0.23%	58.1/sec	226.1
/j_spring_security_check	252	1935	1578	3983	25	6613	0.40%	33.6/sec	157.2
/clubmanager/create.do	99	889	625	2523	16	4138	0.00%	21.7/sec	125.4
TOTAL	1240	1023	504	3062	5	6613	0.16%	127.7/sec	529.3

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.20. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as manager must be able to manage the events of their clubs, which includes listing, creating, updating and deleting them. A manager may update or delete an event only if it's saved in draft mode. When an event is saved in final mode, a notification must be sent at all the clients who follows the club where the event is organized. (RF 13.2)

El archivo correspondiente a este test es *ManageEventsManager.jmx*.

2.1.21. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

1. Loguearse como manager
2. Pulsar en el menú *“My events”*, *“Create event”*

3. Rellenar formulario y pulsar botón save
4. Pulsar el botón Edit del event creado
5. Rellenar formulario y pulsar botón save
6. Pulsar el botón Edit display y volver
7. Pulsar el botón Edit del club
8. Pulsar botón Delete

2.1.22. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	400	108	10	351	5	1724	0.00%	5.1/sec	20.3
/j_spring_security_check	400	157	21	606	10	1801	0.00%	5.1/sec	23.9
/	400	60	13	125	6	1218	0.00%	5.1/sec	22.8
/eventmanager/create.do	800	135	65	311	16	2028	0.00%	9.6/sec	88.3
/eventmanager/edit.do	698	239	105	562	28	2908	0.00%	8.8/sec	94.3
/eventmanager/edit.do?eve...	633	75	45	155	16	862	0.00%	8.2/sec	79.6
/eventmanager/myList.do	300	67	57	120	22	205	0.00%	5.6/sec	58.7
TOTAL	3631	130	47	300	5	2908	0.00%	40.9/sec	330.2

No hay errores HTTP y el tiempo medio de espera total es de 2230ms = 2,230s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 606ms = 0,6s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

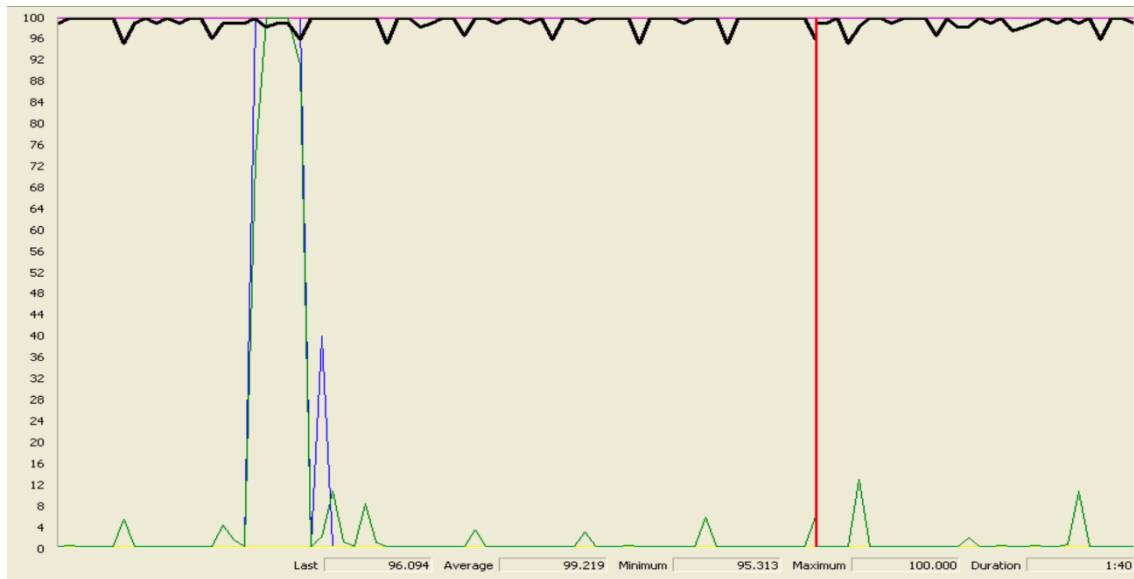
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	408	1274	171	2393	6	43671	0.00%	4.6/sec	18.1
/j_spring_security_check	392	2015	705	4904	13	32597	0.00%	4.4/sec	20.5
/	379	1108	175	2731	7	33392	0.00%	4.3/sec	19.0
/eventmanager/create.do	695	3965	1279	10249	28	70340	0.00%	8.1/sec	73.6
/eventmanager/edit.do	570	4814	2302	12095	160	54834	0.00%	7.1/sec	76.5
/eventmanager/edit.do?eve...	540	2265	1078	4670	46	50555	0.00%	7.1/sec	69.5
/eventmanager/myList.do	235	2707	895	5700	139	47564	0.00%	3.7/sec	38.5
TOTAL	3219	2823	995	6265	6	70340	0.00%	35.8/sec	282.5

No hay errores HTTP y el tiempo medio de espera total es de 42632ms = 42,6s.

El mayor tiempo de espera se da al realizarse la operación de edición siendo de 12095ms = 1,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	327	1010	166	2135	5	32052	0.00%	4.1/sec	16.2
/j_spring_security_check	304	2849	278	7740	12	71940	0.00%	3.8/sec	17.8
/	297	2366	110	5559	7	56310	0.00%	3.9/sec	17.2
/event/manager/create.do	553	8581	3930	22882	27	73180	0.00%	7.2/sec	65.8
/event/manager/edit.do	366	13302	9391	26899	2093	53102	0.00%	5.2/sec	56.2
/event/manager/edit.do?eve...	283	6711	3966	14595	1512	54952	0.00%	4.4/sec	44.4
/event/manager/myList.do	57	6192	4643	11804	1685	19483	0.00%	1.6/sec	17.4
TOTAL	2187	6294	2480	17761	5	73180	0.00%	27.3/sec	206.0

No hay errores HTTP y el tiempo medio de espera total es de 91714ms = 91,714s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 26899ms = 26,899s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	318	313	100	812	5	3136	4.40%	29.6/sec	114.4
/j_spring_security_check	259	726	565	1596	23	3152	0.00%	27.9/sec	130.6
/	246	307	159	756	11	2650	0.00%	24.5/sec	109.3
/event/manager/create.do	215	728	386	1748	1	4548	9.77%	30.8/sec	227.3
/event/manager/edit.do	1	5	5	5	5	5	100.00%	200.0/sec	373.6
TOTAL	1039	500	264	1410	1	4548	3.46%	96.3/sec	475.5

2.1.23. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as manager must be able to cancel an event. When an event is cancelled a notification must be send at all the clients who participle in this events. Only events saved in final mode can be cancelled. (RF 13.3)

El archivo correspondiente a este test es *cancelEventManagerTest.jmx*.

2.1.24. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

9. Loguearse como manager
10. Listar mis clubs
11. Pulsar botón “cancel”

2.1.25. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/security/login.do	144	106	7	253	3	2299	0.00%	4.7/sec	18.6
/j_spring_security_check	129	129	14	127	9	4196	0.00%	4.6/sec	21.3
/	129	81	9	56	5	2661	0.00%	4.6/sec	20.3
/eventmanager/myList.do	107	1431	671	3632	176	8467	0.00%	4.2/sec	43.6
/eventmanager/cancel.do	99	1657	1251	3413	184	5424	0.00%	4.5/sec	48.1
TOTAL	608	591	26	1884	3	8467	0.00%	19.7/sec	127.3

No hay errores HTTP y el tiempo medio de espera total es de 7481ms = 7,481s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3632ms = 3,632s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

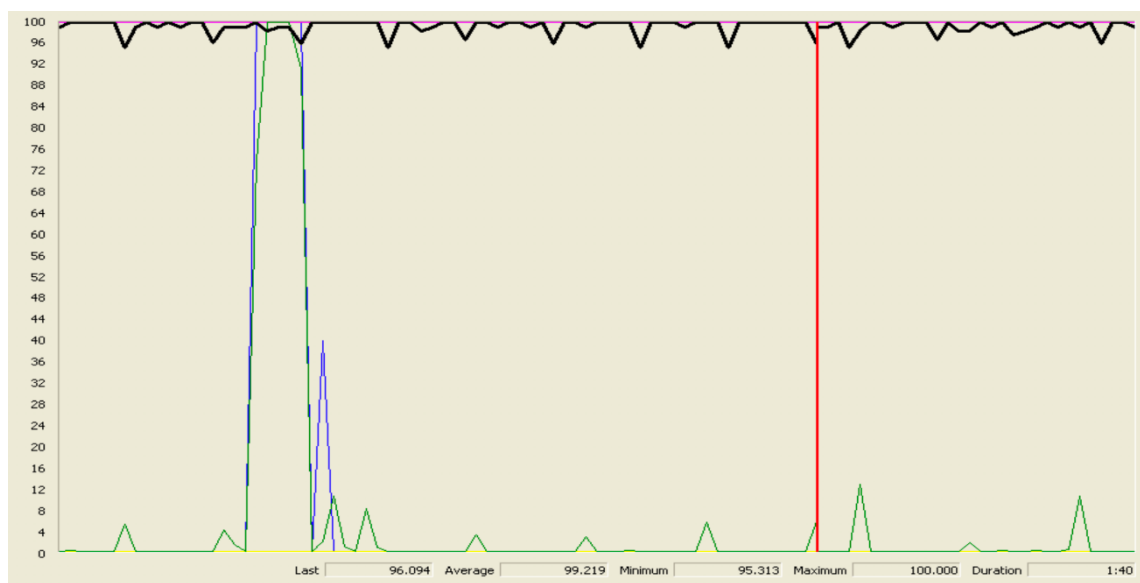
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	274	1031	140	2096	4	21021	0.00%	5.6/sec	22.1
/j_spring_security_check	241	1562	513	4983	10	13718	0.00%	5.2/sec	24.2
/	233	655	95	1704	6	7596	0.00%	5.0/sec	22.4
/event/manager/myList.do	217	3795	1636	12386	234	31711	0.00%	4.9/sec	51.6
/event/manager/cancel.do	192	4070	2299	10469	395	25471	0.00%	4.7/sec	49.9
TOTAL	1157	2089	700	6063	4	31711	0.00%	23.6/sec	154.1

No hay errores HTTP y el tiempo medio de espera total es de 31618ms = 31,6s.

El mayor tiempo de espera se da al realizarse el listado siendo de 12386ms = 12,386s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	355	755	9	3367	4	10408	0.00%	8.9/sec	35.2
/j_spring_security_check	285	2827	170	11088	12	35562	0.00%	7.6/sec	35.4
/	265	2688	107	9273	6	32892	0.00%	7.2/sec	32.1
/event/manager/myList.do	210	9426	7641	19827	256	33504	0.00%	6.0/sec	62.8
/event/manager/cancel.do	121	14007	13153	22245	4784	29090	0.00%	3.7/sec	39.5
TOTAL	1236	4418	209	14973	4	35562	0.00%	30.4/sec	182.2

No hay errores HTTP y el tiempo medio de espera total es de 65789ms = 65,789s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 22245ms = 22,245s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	398	25	10	41	4	678	0.00%	111.1/sec	440.8
/j_spring_security_check	183	494	397	1112	17	2246	0.00%	67.8/sec	317.0
/	142	254	192	524	9	1289	0.00%	51.7/sec	230.8
/event/manager/myList.do	1	14	14	14	14	14	100.00%	71.4/sec	133.4
TOTAL	724	188	28	566	4	2246	0.14%	134.7/sec	571.4

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.26. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. .An actor who is authenticated as manager must be able to list the opinions of his events. (RF 13.4)

El archivo correspondiente a este test es *listOpinionManager.jmx*.

2.1.27. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

12. Loguearse como manager
13. Listar my opinions

2.1.28. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	100	5	6	7	3	9	0.00%	7.4/sec	29.2
/j_spring_security_check	100	39	13	26	11	570	0.00%	7.1/sec	32.9
/	100	35	8	17	5	1131	0.00%	6.7/sec	29.9
/opinion/manager/list.do	97	826	486	2097	125	2871	0.00%	6.7/sec	42.0
/event/opinions.do	72	49	11	166	9	401	0.00%	6.0/sec	31.6
TOTAL	469	195	11	545	3	2871	0.00%	24.0/sec	117.7

No hay errores HTTP y el tiempo medio de espera total es de 2313ms = 2,312s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2097ms = 2,097s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	212	62	6	25	3	4535	0.00%	8.6/sec	34.2
/j_spring_security_check	198	502	59	2180	10	4523	0.00%	9.4/sec	43.9
/	198	154	27	430	6	3821	0.00%	9.4/sec	42.1
/opinion/manager/list.do	170	2244	1243	5763	134	7524	0.00%	8.7/sec	54.6
/event/opinions.do	153	523	102	1686	9	4786	0.00%	9.2/sec	48.6
TOTAL	931	649	46	2537	3	7524	0.00%	37.5/sec	182.3

No hay errores HTTP y el tiempo medio de espera total es de 10084ms = 10,084s.

El mayor tiempo de espera se da al realizarse la operación de listado siendo de 5763ms = 5,7s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	582	910	20	2793	5	20863	0.00%	13.9/sec	55.3
/j_spring_security_check	488	3524	848	10938	10	37297	0.00%	12.4/sec	58.1
/	457	2123	379	6469	6	36424	0.00%	11.6/sec	51.9
/opinion/manager/list.do	394	4516	2402	11749	172	32605	0.00%	10.6/sec	67.3
/event/opinions.do	336	3135	1554	7400	19	28698	0.00%	9.8/sec	52.0
TOTAL	2257	2682	793	7526	5	37297	0.00%	54.1/sec	261.3

No hay errores HTTP y el tiempo medio de espera total es de 31949ms = 31,94s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 11749ms = 11,7s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	400	12	9	22	5	89	0.00%	133.8/sec	530.7
/j_spring_security_check	255	600	386	1396	13	3627	0.39%	57.3/sec	267.8
/	196	328	209	843	10	2439	0.00%	47.9/sec	213.9
/opinion/manager/list.do	19	325	260	796	1	979	26.32%	12.1/sec	62.3
TOTAL	870	262	35	808	1	3627	0.69%	117.2/sec	505.1

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

.

2.1.29. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as manager must be able to list the participations of his events and the total number of participations for each event and navigate to the client who participate in the event and show his personal data and social profiles (RF 13.5)

El archivo correspondiente a este test es *ShowClientManagerTestPlan.jmx*.

2.1.30. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

14. Loguearse como manager
15. Listar my participations
16. Pulsar el boton Show event

2.1.31. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	1000	13	9	23	4	197	0.00%	5.7/sec	22.7
/j_spring_security_check	1000	29	19	53	10	403	0.00%	5.7/sec	26.8
/	1000	11	10	16	5	33	0.00%	5.7/sec	25.3
/eventmanager/myList.do	1000	35	27	62	14	217	0.00%	5.7/sec	48.1
/participationEventmanager/list.do	1000	19	17	30	10	78	0.00%	5.7/sec	31.8
/client/show.do	1000	11	10	17	6	33	0.00%	5.7/sec	35.9
TOTAL	6000	19	15	35	4	403	0.00%	31.7/sec	176.2

No hay errores HTTP y el tiempo medio de espera total es de 201ms = 0,201s.

El mayor tiempo de espera se da al realizarse la operación de listado siendo de 62ms = 0,062s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

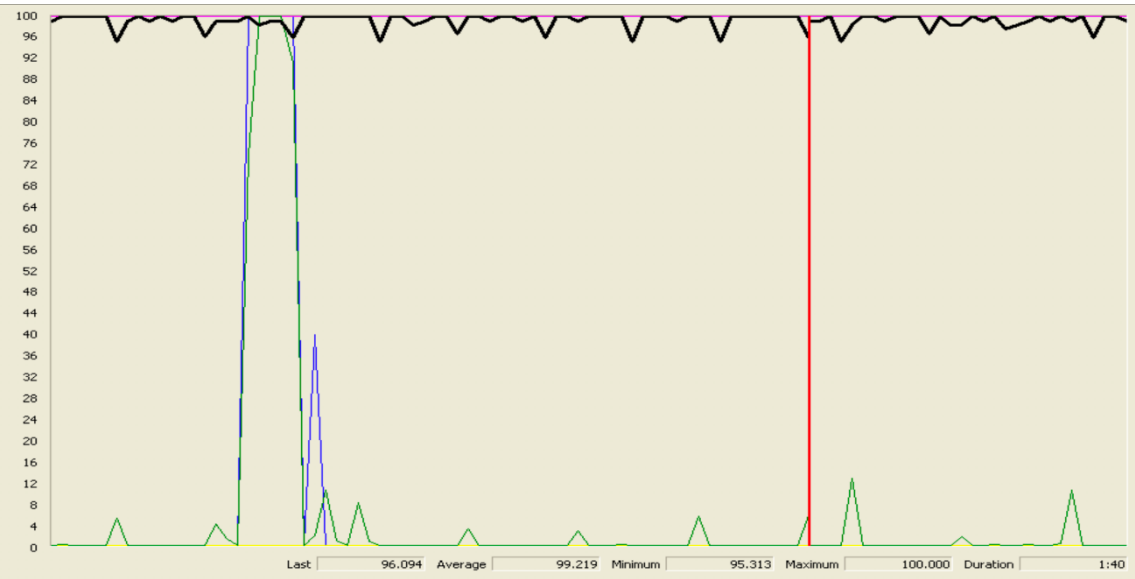
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	1600	22	9	40	3	499	0.00%	11.5/sec	45.5
/j_spring_security_check	1600	54	22	81	8	2493	0.00%	11.5/sec	53.7
/	1600	11	9	17	5	122	0.00%	11.7/sec	52.2
/event/manager/myList.do	1540	55	35	95	13	814	0.00%	11.4/sec	95.5
/participationEvent/manager/list.do	1404	21	17	33	8	238	0.00%	10.8/sec	60.4
/client/show.do	1400	11	10	16	5	87	0.00%	11.7/sec	73.7
TOTAL	9144	29	15	54	3	2493	0.00%	63.9/sec	353.4

No hay errores HTTP y el tiempo medio de espera total es de 282ms = 0,28s.

El mayor tiempo de espera se da al realizarse la operación de listado siendo de 95ms = 0,095s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	1920	38	15	78	3	1072	0.00%	15.9/sec	62.8
/j_spring_security_check	1920	133	42	310	8	2936	0.00%	15.9/sec	74.2
/	1968	60	22	119	4	1698	0.00%	16.3/sec	72.6
/event/manager/myList.do	1844	207	80	587	13	2501	0.00%	15.9/sec	133.5
/participationEvent/manager/list.do	1600	149	47	400	7	2651	0.00%	16.8/sec	94.4
/client/show.do	1600	37	16	68	5	1969	0.00%	16.8/sec	105.7
TOTAL	11252	103	31	240	3	2936	0.00%	92.8/sec	513.7

No hay errores HTTP y el tiempo medio de espera total es de 1552ms = 1,5s.

El mayor tiempo de espera se da al realizarse la operación de listado siendo de 587ms = 0,58s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 450 usuarios
- Loop de 10 iteraciones

/securitylogin.do	450	353	100	1074	1	2323	12.22%	54.6/sec	202.7
/j_spring_security_check	292	479	80	1688	2	2630	3.77%	35.5/sec	163.3
/	224	361	171	945	5	2106	0.00%	37.9/sec	168.8
/event/manager/myList.do	37	38	34	69	16	86	0.00%	12.1/sec	101.9
TOTAL	1003	380	89	1176	1	2630	6.58%	91.4/sec	393.6

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.32. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

CLIENT

3.1. An actor who is authenticated as a client must be able to follow or unfollow a club (RF 14.1)

El archivo correspondiente a este test es *followUnfollowClubClient.jmx*.

2.1.33. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

17. Loguearse como client
18. Listar my clubs
19. Pulsar botón follow/unfollow

2.1.34. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	300	21	9	34	4	374	0.00%	13.4/sec	52.9
/j_spring_security...	300	42	22	91	10	412	0.00%	13.4/sec	64.6
/	300	20	13	36	5	321	0.00%	13.4/sec	61.8
/club/client/follow.do	300	25	22	41	7	87	0.00%	13.3/sec	92.5
/club/client/unfollo...	228	28	23	48	16	101	0.00%	10.9/sec	78.3
/club/client/myList.do	200	12	12	15	8	21	0.00%	16.0/sec	110.7
TOTAL	1628	25	18	43	4	412	0.00%	63.0/sec	353.2

No hay errores HTTP y el tiempo medio de espera total es de 265ms = 0,265s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 91ms = 0,091s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 300 usuarios
- Loop de 10 iteraciones

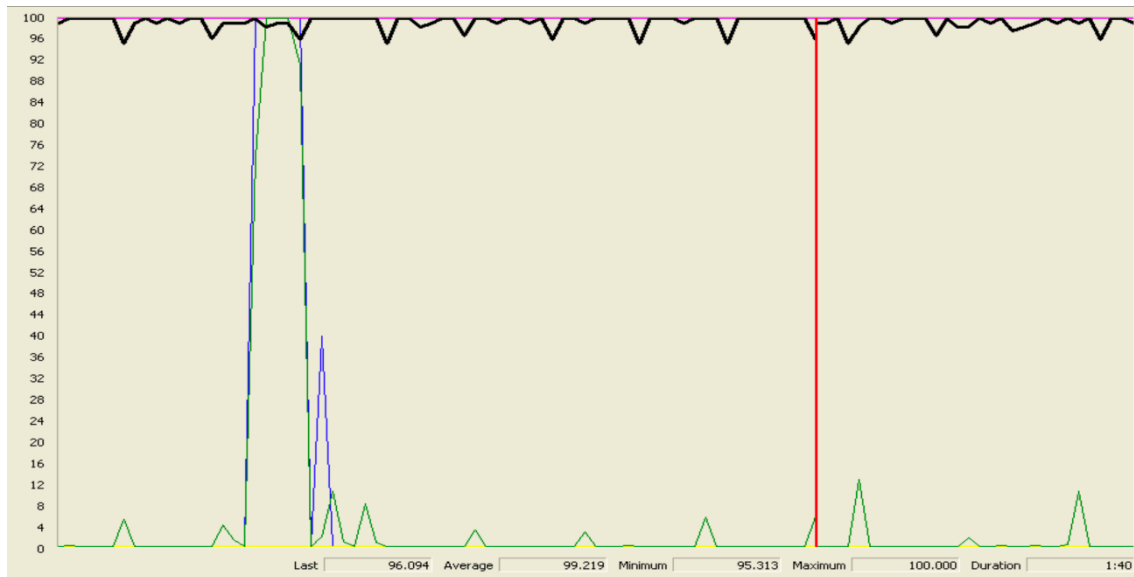
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	476	279	80	694	4	6985	0.00%	26.7/sec	106.0
/j_spring_security...	445	605	225	1629	12	6612	0.00%	25.0/sec	121.0
/	421	280	73	701	6	3906	0.00%	23.7/sec	109.7
/club/client/follow.do	337	972	471	2438	33	6317	0.00%	22.0/sec	155.0
/club/client/unfollo...	280	941	555	2220	50	6409	0.00%	21.7/sec	154.2
/club/client/myList.do	247	486	211	1142	16	5988	0.00%	24.3/sec	166.8
TOTAL	2206	558	214	1454	4	6985	0.00%	123.5/sec	674.8

No hay errores HTTP y el tiempo medio de espera total es de 8824ms = 8,8s.

El mayor tiempo de espera se da al realizarse la operación de follow siendo de 2438ms = 2,4s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/security/login.do	398	475	185	1531	6	3278	12.81%	39.7/sec	146.7
/j_spring_security...	298	690	432	1806	13	3926	7.05%	32.1/sec	149.9
/	255	267	89	810	8	2045	2.35%	31.9/sec	145.6
/club/client/follow.do	225	1043	723	2429	38	3691	8.89%	32.4/sec	213.6
/club/client/unfollo...	116	867	742	1815	80	3379	43.10%	27.0/sec	133.7
/club/client/myList.do	4	139	90	220	36	220	0.00%	4.3/sec	29.7
TOTAL	1296	616	313	1772	6	3926	11.42%	125.5/sec	591.3

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

2.1.35. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to list the clubs he follows and navigate to the events that are organized in that clubs ordered by date and divided in three lists (AVALAIBLE, CANCELLED and FINISHED)(RF 14.2)

El archivo correspondiente a este test es *ListClubNavigateEventClient.jmx*.

2.1.36. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

20. Loguearse como client
21. Acceder al menú “My follow clubs”

2.1.37. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/security/login.do	264	485	260	1144	6	4194	0.00%	26.3/sec	103.8
/j_spring_security_...	235	1218	864	2784	25	6047	0.00%	24.1/sec	116.2
/	216	691	383	1692	20	5250	0.00%	22.5/sec	103.8
/club/client/myList.do	199	514	295	1252	31	4812	0.00%	21.2/sec	147.2
/event/client/myList...	182	610	339	1525	42	5273	0.00%	20.9/sec	158.1
TOTAL	1096	709	384	1695	6	6047	0.00%	107.4/sec	581.5

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

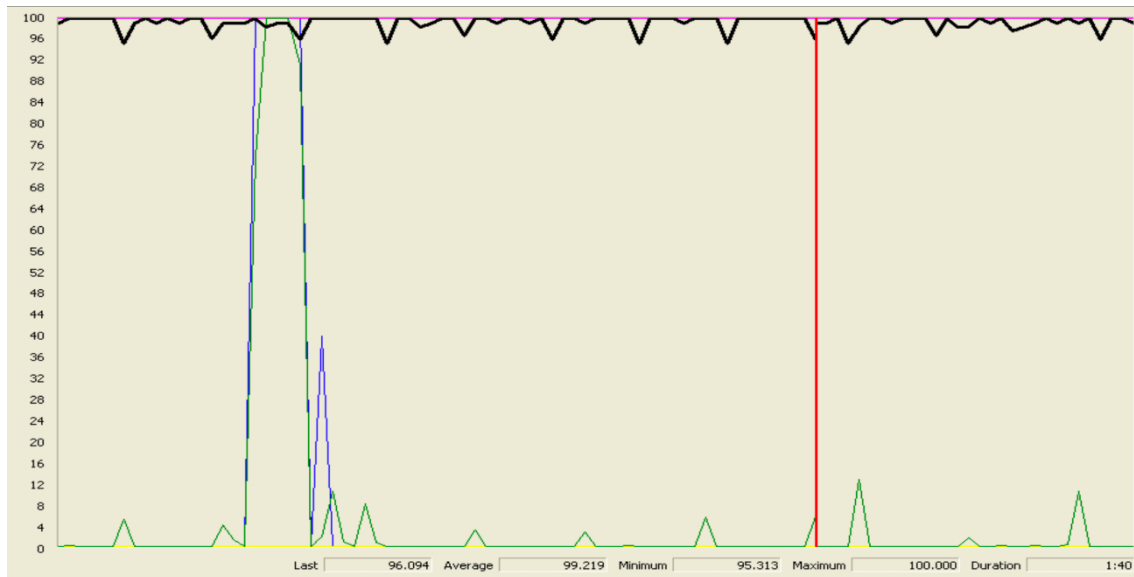
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/security/login.do	762	896	507	2243	6	12609	0.00%	30.0/sec	119.1
/j_spring_security_...	703	2425	1429	5490	17	21251	0.00%	28.0/sec	136.1
/	667	952	423	2354	11	12370	0.00%	26.7/sec	123.9
/club/client/myList.do	637	1055	496	2698	15	14101	0.00%	25.5/sec	177.3
/event/client/myList...	601	993	505	2393	22	10834	0.00%	24.1/sec	182.8
TOTAL	3370	1273	592	3058	6	21251	0.00%	132.6/sec	728.8

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/security/login.do	301	426	255	976	5	3107	0.00%	52.0/sec	206.3
/j_spring_security...	287	1455	1300	2911	32	3480	0.35%	50.2/sec	243.2
/	137	842	767	1501	7	3340	0.00%	24.3/sec	112.5
/club/client/myList.do	22	825	832	1282	101	1523	0.00%	4.2/sec	29.1
/event/client/myList...	3	950	1143	1638	71	1638	0.00%	52.0/min	6.6
TOTAL	750	909	724	1916	5	3480	0.13%	129.0/sec	583.3

No hay errores HTTP y el tiempo medio de espera total es de 7026ms = 7,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1263ms = 1,263s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

2.1.38. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to filter the catalogue of events using the following filters: a single key word that must appear

somewhere in its club, ticker, description, a category to which the event must belong, a range of prices, or a range of dates (RF 14.3).

El archivo correspondiente a este test es *FilterEventsClient.jmx*.

2.1.39. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

22. Loguearse como client
23. Listar mis events
24. Rellenar filtro y pulsar botón

2.1.40. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	200	48	19	128	3	539	0.00%	19.9/sec	79.8
/j_spring_security_...	200	91	60	215	11	593	0.00%	19.9/sec	96.1
/	200	43	27	97	6	301	0.00%	20.0/sec	92.3
/event/list.do	200	15	14	21	10	46	0.00%	19.5/sec	173.1
/event/client/search...	125	48	29	79	19	277	0.00%	14.7/sec	97.6
TOTAL	925	49	23	119	3	593	0.00%	69.8/sec	398.8

No hay errores HTTP y el tiempo medio de espera total es de 236ms = 0,236s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 43ms = 0,043s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

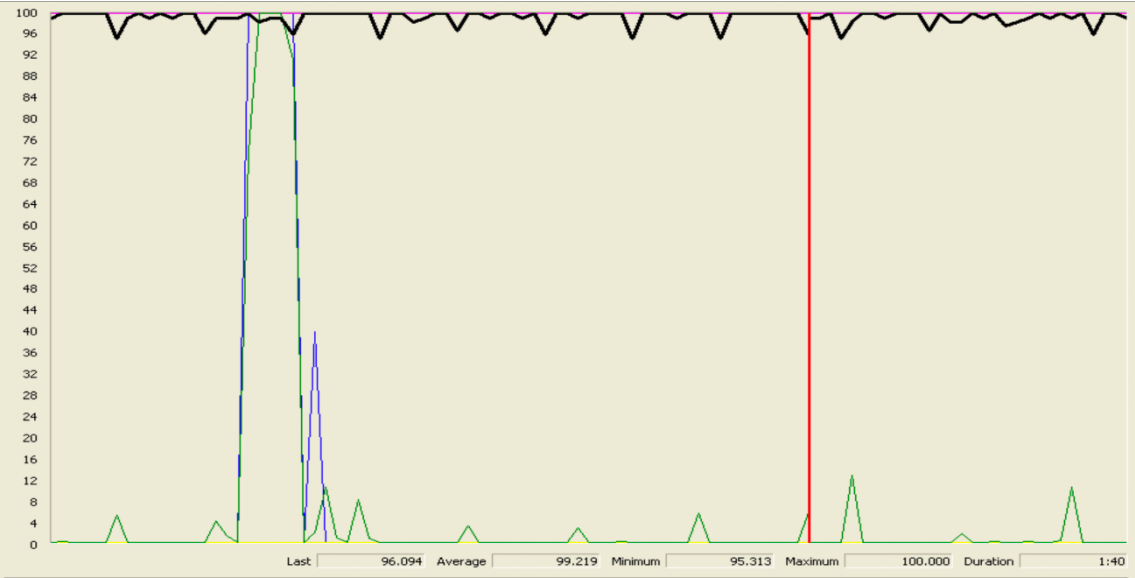
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	452	391	141	1114	5	3106	0.00%	24.3/sec	96.5
/j_spring_security_...	443	533	311	1503	10	3033	0.00%	23.8/sec	115.4
/	437	183	69	472	6	1759	0.00%	23.6/sec	109.4
/event/list.do	400	141	44	436	8	1087	0.00%	28.0/sec	249.4
/event/client/search...	376	138	74	299	20	1457	0.00%	29.3/sec	194.7
TOTAL	2108	285	95	805	5	3106	0.00%	113.4/sec	646.2

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 209ms = 0,209s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	943	564	150	1377	6	10364	0.00%	28.7/sec	113.7
/j_spring_security...	905	990	387	2631	15	12426	0.00%	27.6/sec	134.1
/j	883	444	129	967	7	12660	0.00%	27.0/sec	125.2
/event/list.do	799	594	225	1518	19	7658	0.00%	26.8/sec	236.4
/event/client/search...	737	711	265	1886	29	8372	0.00%	26.8/sec	176.8
TOTAL	4267	661	217	1781	6	12660	0.00%	129.5/sec	735.8

No No hay errores HTTP y el tiempo medio de espera total es de 8026ms = 8,026s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2631ms = 2,631s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	400	33	11	99	4	575	0.00%	133.5/sec	529.4
/j_spring_security...	400	1079	1194	1946	1	2189	24.00%	182.5/sec	771.3
TOTAL	800	556	96	1715	1	2189	12.00%	138.9/sec	568.9

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “Constant Delay Offset” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.41. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to participate in an event of one of the clubs he follows. The client must have a valid credit card linked. Each client can only have one participation by event. Clients can not participate in events which have already finished (RF 14.4)

El archivo correspondiente a este test es *ParticipationEventClientTest.jmx*.

2.1.42. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

25. Loguearse como client
26. Listar mis clubs

2.1.43. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/security/login.do	200	14	10	25	4	164	0.00%	15.4/sec	61.0
/j_spring_security_...	200	32	25	62	10	142	0.00%	15.4/sec	74.5
/	200	17	15	30	6	81	0.00%	15.4/sec	71.3
/event/client/myList...	200	21	17	35	13	88	0.00%	15.1/sec	124.3
/event/client/partici...	127	64	58	102	24	277	0.00%	10.8/sec	91.9
/participationEvent/...	100	41	35	69	22	100	0.00%	42.5/sec	223.3
TOTAL	1027	28	20	61	4	277	0.00%	61.5/sec	355.0

No hay errores HTTP y el tiempo medio de espera total es de 852ms = 0,852s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 102ms = 0,01s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

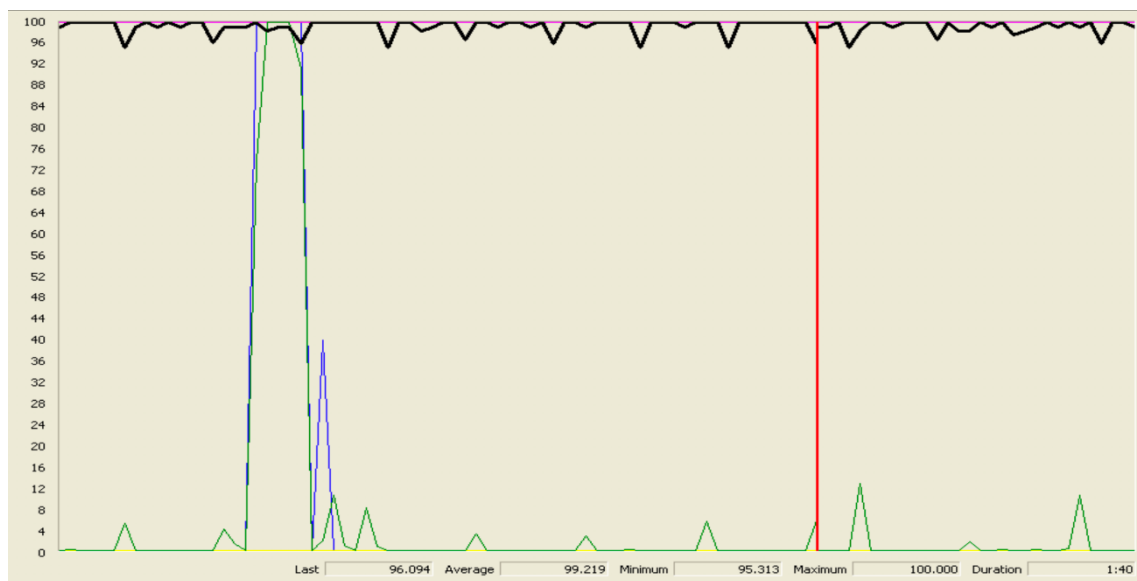
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	557	171	68	453	4	2306	0.00%	21.2/sec	83.9
/j_spring_security_c...	556	311	182	754	12	2297	0.00%	21.2/sec	102.3
/	555	149	64	398	6	1512	0.00%	21.1/sec	97.6
/event/client/myList.do	436	139	62	299	12	1708	0.00%	18.3/sec	150.8
/event/client/participa...	400	563	344	1407	25	3327	0.00%	21.7/sec	183.9
/participationEvent/cil...	399	308	170	741	42	2521	0.00%	21.4/sec	112.8
TOTAL	2903	262	118	679	4	3327	0.00%	110.3/sec	628.6

No hay errores HTTP y el tiempo medio de espera total es de 4052ms = 4,052s.

El mayor tiempo de espera se da al realizarse la operación de participación siendo de 1407ms =1,4s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	260	124	18	390	5	1092	5.00%	56.3/sec	217.0
/j_spring_security_c...	199	152	39	399	15	1371	1.01%	44.9/sec	216.2
/	180	75	18	124	7	1043	0.00%	41.8/sec	193.2
/event/client/myList.do	93	227	181	480	16	707	0.00%	45.1/sec	371.3
TOTAL	732	132	29	393	5	1371	2.05%	158.4/sec	769.7

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.44. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to list the events he participate(RF 14.5)

El archivo correspondiente a este test es *ListEventHeParticipateClient.jmx*.

2.1.45. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

27. Loguearse como client
28. Acceder a My events en el menu

2.1.46. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/securitylogin.do	1000	16	8	25	4	480	0.00%	9.1/sec	35.9
/j_spring_security_c...	1000	18	18	67	11	1788	0.00%	9.1/sec	44.0
/	1000	12	9	15	6	175	0.00%	9.0/sec	41.6
/participationEvent/c...	1000	15	14	20	9	49	0.00%	8.9/sec	44.7
/event/show.do	1000	12	12	17	8	40	0.00%	8.8/sec	58.2
TOTAL	5000	19	12	25	4	1788	0.00%	42.0/sec	210.2

No hay errores HTTP y el tiempo medio de espera total es de 144ms = 0,144s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 67ms = 0,067s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

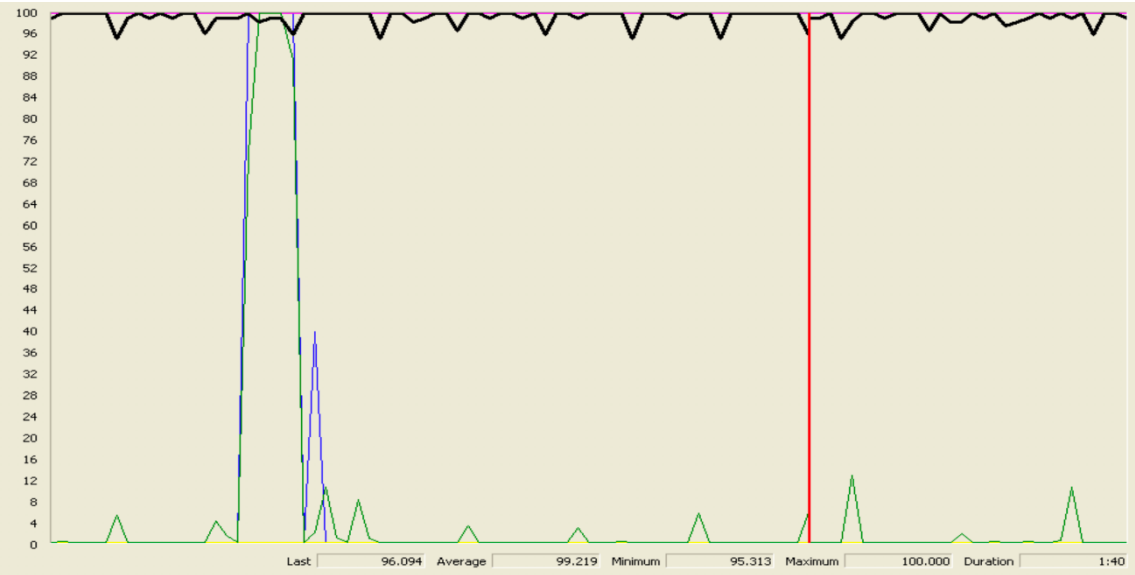
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/security/login.do	2000	33	16	71	4	822	0.00%	22.2/sec	88.0
/j_spring_security_c...	2000	75	42	162	10	1667	0.00%	22.2/sec	107.5
/	2000	12	10	20	5	148	0.00%	22.1/sec	102.4
/participationEvent/c...	2000	19	16	33	7	131	0.00%	21.3/sec	107.2
/event/show.do	2000	20	14	36	7	420	0.00%	21.3/sec	140.6
TOTAL	10000	32	16	64	4	1667	0.00%	100.2/sec	501.8

No hay errores HTTP y el tiempo medio de espera total es de 322ms = 0,322s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 162ms = 0,162s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/j_spring_security_c...	2579	137	57	298	8	2447	0.00%	33.3/sec	161.5
/security/login.do	2573	64	21	143	3	1969	0.00%	33.4/sec	132.6
/	2696	36	16	67	4	1137	0.00%	34.5/sec	159.7
/participationEventic...	2551	52	21	64	7	1872	0.00%	33.8/sec	169.7
/event/show.do	2422	63	26	105	6	2101	0.00%	33.2/sec	219.7
TOTAL	12821	70	25	130	3	2447	0.00%	161.2/sec	805.2

No hay errores HTTP y el tiempo medio de espera total es de 677ms = 6,77s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 298ms = 2,98s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 5:

- 450 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	307	167	62	441	4	1549	0.98%	69.1/sec	272.7
/j_spring_security_c...	240	263	87	769	18	1310	1.25%	54.2/sec	260.7
/	49	203	115	476	8	1187	0.00%	29.9/sec	138.2
TOTAL	596	208	79	580	4	1549	1.01%	134.2/sec	583.7

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.47. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to publish an opinion about an event he participated who is already finished (RF 14.6)

El archivo correspondiente a este test es *publishOpinionClientTest.jmx*.

2.1.48. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

29. Loguearse como client
30. Listar my opinions
31. Pulsar botón de "Create opinión"
32. Rellenar formulario y pulsar save

2.1.49. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	200	23	7	26	3	681	0.00%	15.7/sec	61.9
/j_spring_security_...	200	131	24	441	9	725	0.00%	15.6/sec	75.5
/	200	41	15	114	5	434	0.00%	15.7/sec	72.3
/opinion/client/crea...	262	133	63	376	23	767	0.00%	21.8/sec	121.4
/opinion/client/list.do	100	35	33	53	21	70	0.00%	39.2/sec	212.9
TOTAL	962	80	28	241	3	767	0.00%	66.3/sec	322.8

No hay errores HTTP y el tiempo medio de espera total es de 1010ms = 1,01s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 441ms = 0,441s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	383	164	34	463	4	3034	0.00%	19.7/sec	77.9
/j_spring_security_...	376	384	122	882	11	8162	0.00%	19.1/sec	92.5
/	370	171	35	443	7	6063	0.00%	19.0/sec	87.9
/opinion/client/crea...	589	673	233	1589	40	9727	0.00%	33.8/sec	188.4
/opinion/client/list.do	198	417	195	955	43	3184	0.00%	15.9/sec	86.2
TOTAL	1916	391	124	902	4	9727	0.00%	96.5/sec	473.8

No hay errores HTTP y el tiempo medio de espera total es de 4332ms = 4,33s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1589ms = 1,589s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	562	915	222	2180	5	20246	0.00%	19.4/sec	77.0
/j_spring_security...	519	1765	794	4359	14	22480	0.00%	18.0/sec	87.3
/	492	750	207	1791	8	8986	0.00%	17.1/sec	79.1
/opinioni/client/crea...	831	1363	687	3155	44	22666	0.00%	31.5/sec	176.2
/opinioni/client/list.do	321	1148	637	2569	43	13845	0.00%	15.3/sec	83.0
TOTAL	2725	1217	505	3061	5	22666	0.00%	94.0/sec	463.0

No hay errores HTTP y el tiempo medio de espera total es de 14054ms = 14,054s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 4359ms = 4,359s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	640	1282	605	3011	6	21641	0.00%	20.9/sec	83.1
/j_spring_security...	567	3242	2153	7142	23	22271	0.00%	19.3/sec	93.7
/	536	1492	857	3317	8	19237	0.00%	18.3/sec	84.9
/opinioni/client/crea...	814	2177	1300	4777	43	24127	0.00%	30.3/sec	169.4
/opinioni/client/list.do	297	2016	1203	4120	95	12770	0.00%	13.9/sec	75.6
TOTAL	2854	2043	1145	4613	6	24127	0.00%	93.2/sec	455.5

No hay errores HTTP y el tiempo medio de espera total es de 6118ms = 6,118s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 7142ms = 7,142s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado. El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 5:

- 500 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/securitylogin.do	281	63	16	196	4	331	0.00%	117.2/sec	464.5
/j_spring_security...	19	352	378	458	133	506	10.53%	33.9/sec	156.3
TOTAL	300	81	19	238	4	506	0.67%	104.9/sec	419.9

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.50. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as a client must be able to manage his credit card which includes creating, editing and deleting it. (RF 14.7)

El archivo correspondiente a este test es *ManagerCreditCardTestPlan.jmx*.

2.1.51. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

33. Loguearse como client
34. En el menú del perfil, pulsar *“Edit personal data”*
35. Cambiar tarjeta de crédito y pulsar save

2.1.52. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	981	9	8	12	5	49	0.00%	14.6/sec	60.9
/security/login.do	500	12	8	24	5	123	0.00%	8.0/sec	31.0
/j_spring_security_c...	500	28	19	57	11	255	0.00%	8.0/sec	38.6
/creditCard/client/ed...	1200	14	12	19	7	120	0.00%	22.5/sec	143.7
TOTAL	3181	14	11	23	5	255	0.00%	47.5/sec	240.3

No hay errores HTTP y el tiempo medio de espera total es de 112ms = 0,112s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 24ms = 0,024s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

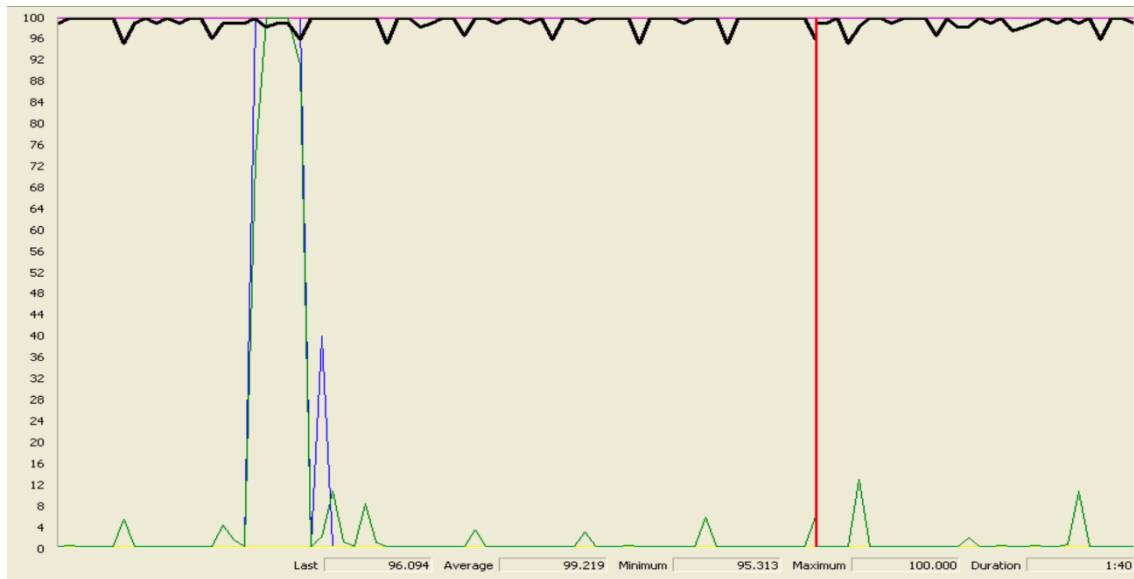
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	2400	12	9	18	4	453	0.00%	28.0/sec	116.8
/security/login.do	1200	53	9	74	4	1424	0.00%	14.8/sec	57.4
/j_spring_security_c...	1200	106	22	268	8	1804	0.00%	14.8/sec	71.5
/creditCard/client/ed...	3295	17	13	25	5	339	0.00%	40.0/sec	255.8
TOTAL	8095	34	12	36	4	1804	0.00%	90.7/sec	466.1

No hay errores HTTP y el tiempo medio de espera total es de 385ms = 0,385s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 268ms = 0,268s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	6000	29	14	59	4	1125	0.00%	39.6/sec	165.6
/security/login.do	3000	84	19	175	3	2878	0.00%	20.5/sec	79.8
/j_spring_security_c...	3000	170	53	412	8	2912	0.00%	20.5/sec	99.4
/creditCard/cliented...	8564	22	15	42	4	423	0.00%	58.3/sec	373.8
TOTAL	20564	55	17	88	3	2912	0.00%	133.8/sec	690.6

No hay errores HTTP y el tiempo medio de espera total es de 688ms = 0,688s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 412ms = 0,412s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 450 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	574	214	42	664	1	5150	0.35%	71.2/sec	279.0
/security/login.do	448	910	468	2581	1	4459	4.91%	74.0/sec	280.8
/j_spring_security_c...	262	1676	1275	3719	5	5669	1.91%	43.3/sec	208.4
TOTAL	1284	755	222	2486	1	5669	2.26%	159.0/sec	645.1

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.53. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

ADMINISTRATOR

3.1. An actor who is authenticated as an administrator must be able to create new user accounts for new administrators(RF 15.1)

El archivo correspondiente a este test es *RegisterAdminTestPlan.jmx*.

2.1.54. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

36. Loguearse como admin
37. Acceder al menú Register
38. Pulsar botón “Register a administrator”
39. Rellenar formulario y pulsar save

2.1.55. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	1000	100	15	170	5	1772	0.00%	11.5/sec	59.0
/j_spring...	1000	154	38	236	12	4717	0.00%	11.5/sec	60.2
/	1000	99	27	110	6	3040	0.00%	11.6/sec	58.2
/registerf...	2000	68	31	123	7	2163	0.00%	22.7/sec	195.4
/welcom...	1000	29	16	60	6	562	0.00%	11.5/sec	57.9
TOTAL	6000	86	27	123	5	4717	0.00%	63.5/sec	399.2

No hay errores HTTP y el tiempo medio de espera total es de 699ms = 0,699s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 236ms = 0,236s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

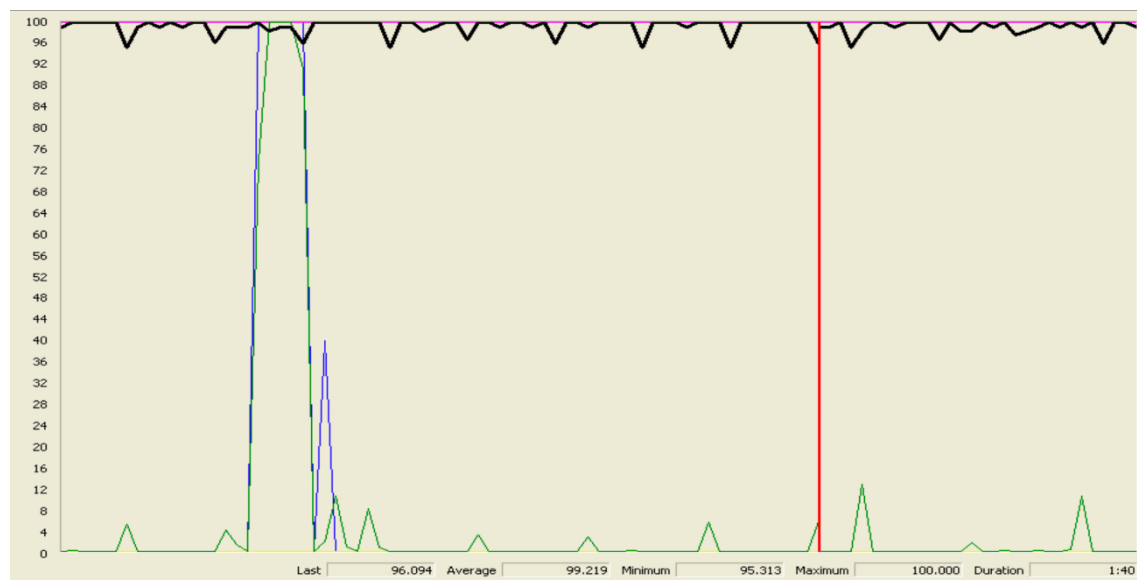
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	3000	236	54	633	5	5463	0.00%	10.8/sec	55.8
/j_spring...	3000	472	165	1318	11	7223	0.00%	10.8/sec	56.7
/	3000	229	66	556	5	6244	0.00%	10.8/sec	54.4
/register/...	6000	248	82	630	6	6515	0.00%	21.5/sec	185.3
/welcom...	3000	210	58	528	6	5663	0.00%	10.8/sec	54.6
TOTAL	18000	274	77	709	5	7223	0.00%	63.2/sec	397.7

No hay errores HTTP y el tiempo medio de espera total es de 3665ms = 3,6s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1318ms = 1,3s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	5500	265	57	729	4	6523	0.00%	11.3/sec	58.4
/j_spring...	5500	557	193	1566	5	7816	0.00%	11.3/sec	59.6
/	5500	266	68	723	5	6244	0.00%	11.3/sec	57.1
/register/...	11000	279	83	765	5	6515	0.00%	22.6/sec	194.7
/welcom...	5500	240	55	639	5	5663	0.00%	11.3/sec	56.9
TOTAL	33000	314	79	879	4	7816	0.00%	66.6/sec	419.3

No hay errores HTTP y el tiempo medio de espera total es de 3884ms = 3,88s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1566ms = 1,5s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	8500	463	116	1336	4	10970	0.00%	12.4/sec	63.8
/j_spring...	8500	924	422	2534	5	10525	0.00%	12.4/sec	65.4
/	8500	461	135	1300	5	9508	0.00%	12.4/sec	62.6
/register/...	17000	477	147	1305	5	10767	0.00%	24.7/sec	213.7
/welcom...	8500	434	108	1247	5	13252	0.00%	12.4/sec	62.6
TOTAL	51000	539	160	1526	4	13252	0.00%	73.7/sec	464.1

No hay errores HTTP y el tiempo medio de espera total es de 7722ms = 7,7s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 2534ms = 2,5s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 5:

- 350 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	9586	459	127	1315	0	10970	0.31%	12.2/sec	62.1
/j_spring...	9511	922	449	2475	0	10525	0.15%	12.1/sec	63.8
/	9485	453	141	1270	5	9508	0.00%	12.1/sec	61.0
/register/...	18829	474	158	1291	0	10767	0.22%	24.0/sec	207.0
/welcom...	9310	433	121	1229	0	13252	0.16%	11.9/sec	60.1
TOTAL	56721	537	172	1504	0	13252	0.18%	72.0/sec	451.9

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “Constant Delay Offset” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.56. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as an administrator must be able to manage the catalogue of categories, which includes listing, showing, creating, updating and deleting them. Note that categories evolve independently from events, which means that they are referenced from an event or not. (RF 15.2)

El archivo correspondiente a este test es *manageCategoryAdmin.jmx*.

2.1.57. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

40. Loguearse como admin
41. Acceder a categories a través del menú
42. Pulsar el botón “Create category”
43. Rellenar el formulario y pulsar save
44. Pulsar el botón edit de la categoría creada
45. Rellenar el formulario y pulsar el botón save
46. Pulsar el botón de display y volver
47. Pulsar el botón de edit y de delete

2.1.58. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	50	5	5	7	3	12	0.00%	38.9/sec	153.7
/j_spring_security_check	50	12	13	15	10	22	0.00%	39.3/sec	209.9
/category/administrator/create.do	92	2506	1483	6398	31	8126	0.00%	9.1/sec	195.3
/category/administrator/edit.do	29	1083	172	4183	23	5060	0.00%	3.2/sec	45.3
TOTAL	221	1189	35	4372	3	8126	0.00%	10.6/sec	137.4

No hay errores HTTP y el tiempo medio de espera total es de 10573ms = 10,573s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 6398ms = 6,3s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

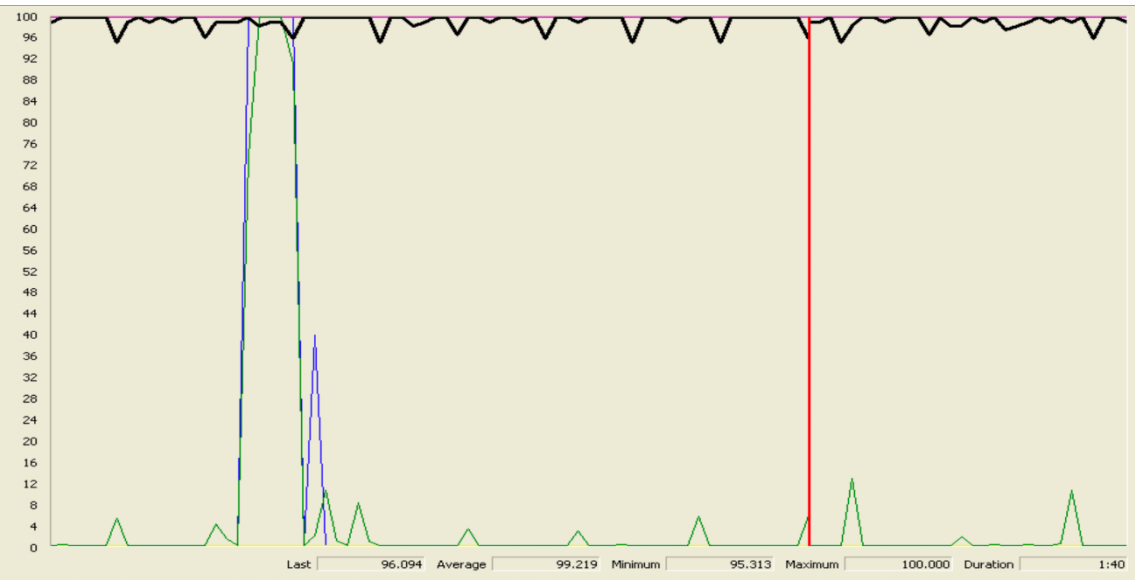
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	100	5	6	7	3	11	0.00%	69.5/sec	275.0
/j_spring_security_check	100	17	16	26	11	39	0.00%	67.6/sec	360.7
/category/administrator/create.do	151	6665	5703	13976	316	20076	0.00%	5.1/sec	122.5
/category/administrator/edit.do	46	5728	4997	9565	79	11619	0.00%	2.0/sec	31.3
TOTAL	397	3205	33	10213	3	20076	0.00%	10.9/sec	145.3

No hay errores HTTP y el tiempo medio de espera total es de 23574ms = 23,574s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 13976ms = 13,976s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	300	10	8	16	4	197	0.00%	151.1/sec	599.1
/j_spring_security_check	300	139	71	364	8	706	0.00%	128.7/sec	689.0
/category/administrator/create.do	455	28736	28527	53585	899	72073	0.00%	6.0/sec	173.5
/category/administrator/edit.do	46	14994	13889	22405	6770	31254	0.00%	57.5/min	9.2
TOTAL	1101	12956	292	42482	4	72073	0.00%	13.6/sec	203.0

No hay errores HTTP y el tiempo medio de espera total es de 76370ms = 76,3s.

El mayor tiempo de espera se da al realizarse la operación de creación siendo de 53585ms = 53,5s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	400	3363	3313	7340	1	12676	24.50%	24.5/sec	84.6
/j_spring_security_check	188	1210	1344	1728	73	3780	0.00%	30.5/sec	163.1
TOTAL	588	2675	1681	6463	1	12676	16.67%	32.5/sec	132.1

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el “Constant Delay Offset” de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.59. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as an administrator must be able to list the clubs that are not accepted yet and accept or refuse them. If the club is refused, a reason must be given by the administrator (RF 15.3)

El archivo correspondiente a este test es *acceptRejectClubAdmin.jmx*.

2.1.60. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

48. Loguearse como admin
49. Listar mis clubs en el menú
50. Pulsar botón Accept
51. Pulsar botón Reject

2.1.61. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	700	20	10	40	4	411	0.00%	5.6/sec	22.0
/j_spring_security_check	700	48	21	107	5	882	0.00%	5.6/sec	29.7
/	699	12	11	19	6	131	0.00%	5.6/sec	28.5
/club/administrator/list.do	1822	17	15	26	8	120	0.00%	14.8/sec	123.3
/club/administrator/accept.do	600	43	22	53	13	627	0.00%	5.8/sec	49.6
/club/administrator/reject.do	600	25	22	37	13	91	0.00%	5.8/sec	49.7
/club/administrator/reject.do?clubid...	600	13	11	19	6	107	0.00%	5.8/sec	33.0
TOTAL	5721	24	16	32	4	882	0.00%	44.6/sec	304.4

No hay errores HTTP y el tiempo medio de espera total es de 301ms = 0,3s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 107ms = 0,1s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

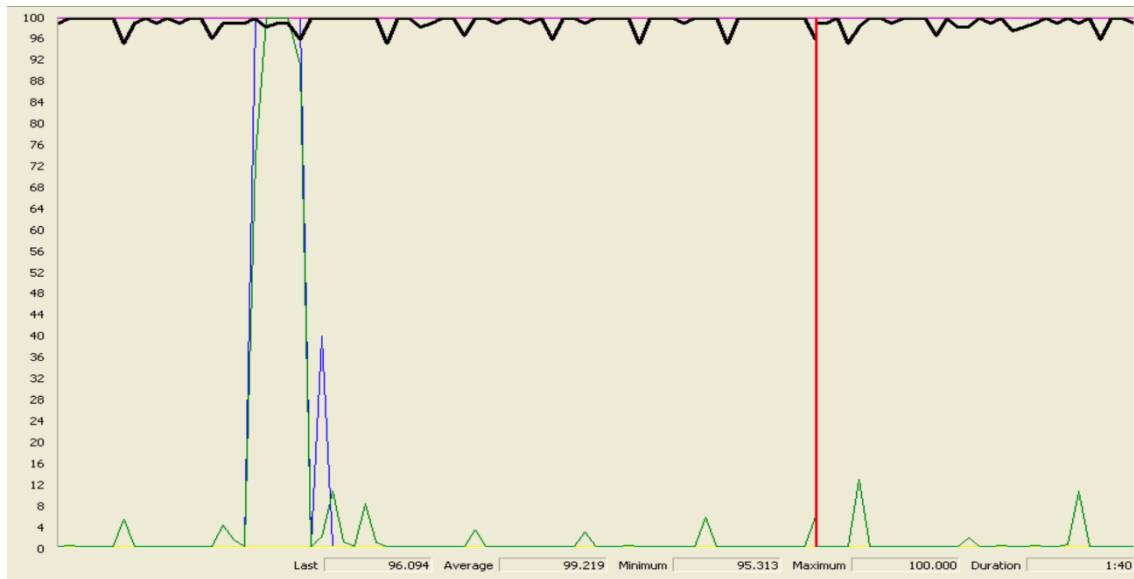
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	1000	106	30	265	3	2172	0.00%	11.1/sec	44.0
/j_spring_security_check	1000	229	70	686	10	2586	0.00%	11.1/sec	59.4
/	1000	60	12	61	5	1721	0.00%	11.2/sec	57.2
/club/administrator/list.do	2786	36	18	55	7	1716	0.00%	29.6/sec	247.8
/club/administrator/accept.do	1000	69	33	151	10	994	0.00%	11.1/sec	95.7
/club/administrator/reject.do	851	53	29	98	10	707	0.00%	9.9/sec	85.1
/club/administrator/reject.do?clubid...	801	16	12	26	5	240	0.00%	9.6/sec	54.8
TOTAL	8438	74	21	134	3	2586	0.00%	84.7/sec	585.3

No hay errores HTTP y el tiempo medio de espera total es de 1342ms = 1,342s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 686ms = 0,68s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	2400	199	60	492	4	3622	0.00%	14.4/sec	57.1
/j_spring_security_check	2399	392	178	1011	9	3753	0.00%	14.4/sec	77.1
/	2297	95	34	252	4	1649	0.00%	14.0/sec	71.9
/club/administratorlist.do	6394	73	42	147	7	2313	0.00%	39.5/sec	330.7
/club/administrator/accept.do	2120	144	91	327	11	1458	0.00%	13.3/sec	114.9
/club/administrator/reject.do	2100	106	69	227	9	1483	0.00%	14.3/sec	123.8
/club/administrator/reject.do?clubid...	2100	35	21	70	5	744	0.00%	14.4/sec	82.4
TOTAL	19810	135	51	317	4	3753	0.00%	118.7/sec	815.9

No hay errores HTTP y el tiempo medio de espera total es de 2526ms = 2,526s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 1011ms = 1,011s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 450 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	450	1022	742	2375	5	5583	4.44%	50.6/sec	195.9
/j_spring_security_check	243	1369	452	3854	16	6545	5.35%	25.5/sec	132.4
/	165	947	626	2281	3	4979	3.03%	28.7/sec	144.5
/club/administratorlist.do	60	443	314	979	25	1583	0.00%	22.8/sec	191.1
TOTAL	918	1063	566	2755	3	6545	4.14%	96.2/sec	454.5

No hay errores HTTP y el tiempo medio de espera total es de 9489ms = 9,489s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 3854ms = 3,8s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado. El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 5:

- 400 usuarios
- Loop de 10 iteraciones

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.62. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. An actor who is authenticated as an administrator must be able to display a dashboard (RF 15.5)

El archivo correspondiente a este test es *DashboardAdminTestPlan.jmx*.

2.1.63. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

52. Loguearse como admin
53. Acceder a la dashboar a través del menu

2.1.64. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	1000	22	11	36	6	451	0.00%	14.8/sec	76.2
/j_spring...	1000	32	30	56	10	208	0.00%	14.8/sec	77.9
/	1000	20	14	37	6	223	0.00%	14.8/sec	74.4
/dashbo...	1000	31	25	59	9	215	0.00%	14.8/sec	91.6
TOTAL	4000	26	19	48	6	451	0.00%	55.5/sec	300.2

No hay errores HTTP y el tiempo medio de espera total es de 158ms = 0,158s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 30ms = 0,03s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

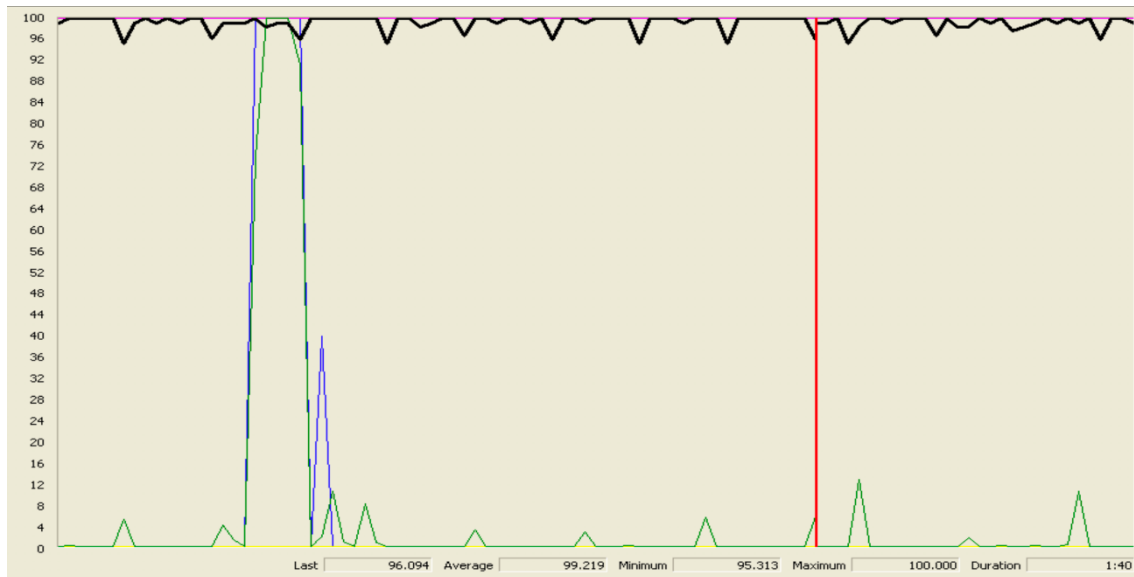
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	3000	31	13	60	5	1532	0.00%	11.4/sec	58.9
/j_spring...	3000	62	32	129	6	1011	0.00%	11.5/sec	60.4
/	3000	46	17	84	6	1535	0.00%	11.5/sec	57.8
/dashbo...	3000	52	26	89	8	1523	0.00%	11.5/sec	71.3
TOTAL	12000	48	22	89	5	1535	0.00%	45.2/sec	244.8

No hay errores HTTP y el tiempo medio de espera total es de 336ms = 0,36s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 109ms = 0,109s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...	KB/sec
/security/l...	5500	51	16	111	5	2372	0.00%	8.3/sec	42.5
/j_spring...	5500	102	40	243	6	3085	0.00%	8.3/sec	43.6
/r	5500	64	23	149	5	2181	0.00%	8.3/sec	41.8
/dashbo...	5500	73	32	151	8	1968	0.00%	8.3/sec	51.4
TOTAL	22000	73	27	164	5	3085	0.00%	32.9/sec	178.2

No hay errores HTTP y el tiempo medio de espera total es de 752ms = 0,76.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 243ms = 0,243s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/logi...	8500	111	18	227	4	6462	0.00%	9.5/sec	48.9
/j_spring_se...	8500	230	46	597	6	7527	0.00%	9.5/sec	50.0
/r	8500	130	24	276	5	7469	0.00%	9.5/sec	47.9
/dashboard/...	8500	134	35	275	8	4918	0.00%	9.5/sec	58.9
TOTAL	34000	151	30	333	4	7527	0.00%	37.7/sec	204.7

No hay errores HTTP y el tiempo medio de espera total es de 982ms = 0,98.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 592ms = 0,59s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 5:

- 400 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/logi...	469	454	64	1509	0	6376	1.49%	51.3/sec	211.1
/j_spring_se...	350	590	70	2136	0	3419	3.43%	46.1/sec	237.6
/	231	446	36	1554	0	3483	0.87%	37.6/sec	188.8
/dashboard/...	170	893	605	2181	0	3537	5.88%	39.5/sec	234.4
TOTAL	1220	553	74	1929	0	6376	2.54%	133.3/sec	644.8

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.65. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.

3.1. The system must be easy to customize at run time (RNF 17)

El archivo correspondiente a este test es *itemProviderCasePlan.jmx*.

2.1.66. Casos de uso

En este caso tenemos un caso de uso con los siguientes pasos:

54. Loguearse como admin
55. Acceder al menú del perfil y pulsar Configuration
56. Pulsar edit
57. Rellenar el formulario y pulsar save

2.1.67. Pruebas realizadas

Prueba 1:

- 100 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...
/security/l...	1000	17	8	31	5	518	0.00%	9.4/sec
/j_spring...	1000	60	31	90	11	2022	0.00%	9.4/sec
/	1000	27	24	42	5	414	0.00%	9.4/sec
/configur...	1000	15	9	28	5	315	0.00%	9.5/sec
/configur...	2000	25	19	44	6	225	0.00%	18.7/sec
/configur...	1000	18	14	32	7	82	0.00%	9.5/sec
TOTAL	7000	27	17	42	5	2022	0.00%	61.3/sec

No hay errores HTTP y el tiempo medio de espera total es de 56ms = 0,56s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 90ms = 0,09s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta aceptable.

Prueba 2:

- 200 usuarios
- Loop de 10 iteraciones

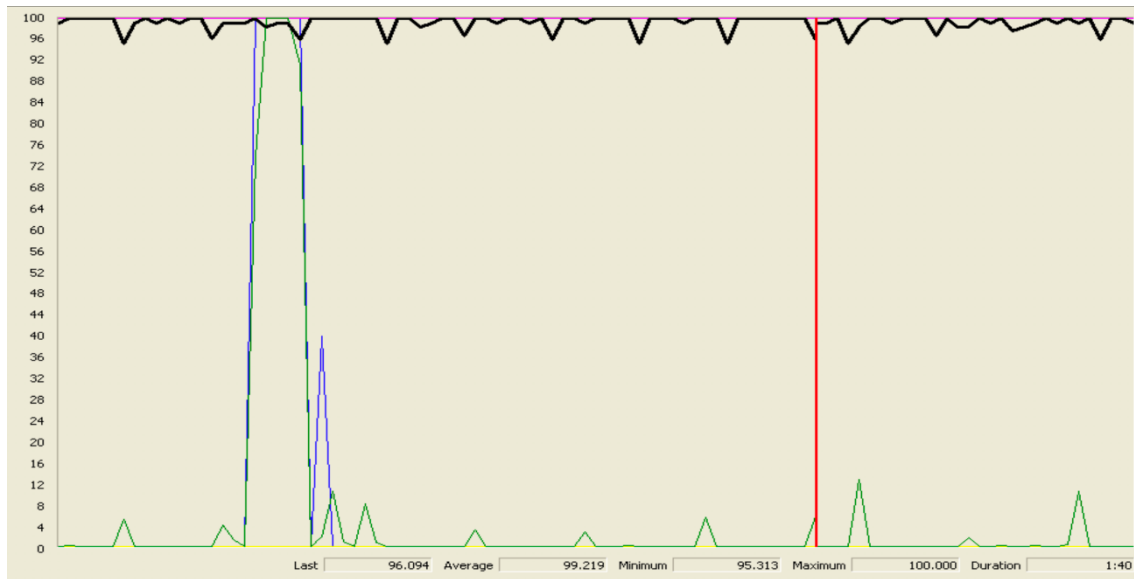
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...
/security/l...	3000	62	13	121	4	4361	0.00%	11.2/sec
/j_spring...	3000	131	35	275	9	10764	0.00%	11.1/sec
/	3000	60	23	108	5	7907	0.00%	11.2/sec
/configur...	3000	64	14	74	5	10165	0.00%	11.2/sec
/configur...	6000	125	24	136	6	10077	0.00%	22.3/sec
/configur...	3000	106	21	164	7	8091	0.00%	11.2/sec
TOTAL	21000	96	22	140	4	10764	0.00%	76.0/sec

No hay errores HTTP y el tiempo medio de espera total es de 881ms = 0,81s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 275ms = 0,275s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta ya empieza a ser demasiado elevado.

El cuello de botella se produce en la CPU, como se puede ver en la ventana de rendimiento.



Prueba 3:

- 250 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...
/security/l...	5500	58	13	98	4	4361	0.00%	11.6/sec
/j_spring...	5500	106	34	206	4	10764	0.00%	11.6/sec
/	5500	49	20	85	4	7907	0.00%	11.6/sec
/configur...	5500	55	14	75	4	10165	0.00%	11.6/sec
/configur...	11000	91	23	117	5	10077	0.00%	23.2/sec
/configur...	5500	75	21	105	7	8091	0.00%	11.6/sec
TOTAL	38500	75	21	115	4	10764	0.00%	79.7/sec

No hay errores HTTP y el tiempo medio de espera total es de 902ms = 0,092s.

El mayor tiempo de espera se da al realizarse la operación de seguridad al loguearse siendo de 206ms = 0,206s.

Por lo tanto, la ejecución ha sido satisfactoria y el tiempo de respuesta es demasiado elevado.

El cuello de botella se sigue produciendo en la CPU al igual que en la anterior prueba.

Prueba 4:

- 300 usuarios
- Loop de 10 iteraciones

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughp...
/security/l...	8437	100	16	164	3	6246	0.00%	13.1/sec
/j_spring...	8437	182	44	410	0	10764	0.06%	13.1/sec
/	8432	84	24	148	4	8234	0.00%	13.1/sec
/configur...	8432	92	18	163	1	10165	0.02%	13.2/sec
/configur...	16860	113	27	198	5	10077	0.00%	26.3/sec
/configur...	8430	88	25	153	7	8091	0.00%	13.2/sec
TOTAL	59028	110	25	201	0	10764	0.01%	91.1/sec

En esta prueba ya existen errores HTTP, lo que significa que no podemos asegurar el servicio al 100% de los usuarios simultáneos.

Mirando el código de error devuelto vemos que se trata del siguiente:

“Non HTTP response code: javax.net.ssl.SSLPeerUnverifiedException, Non HTTP response message: peer not authenticated”

Mirando en internet vemos que este error es debido a un número demasiado elevado de peticiones simultáneas. Si elevamos el *“Constant Delay Offset”* de la prueba podemos realizar esta misma prueba con éxito, pero ello no resultaría muy realista (el tiempo recomendado por la asignatura es de 1500ms).

2.1.68. Conclusiones

El punto óptimo respecto a la velocidad de respuesta está entre 150 y 300 usuarios simultáneos y respecto a la disponibilidad del servicio al 100% de los usuarios está entre 350 y 400 usuarios simultáneos.