

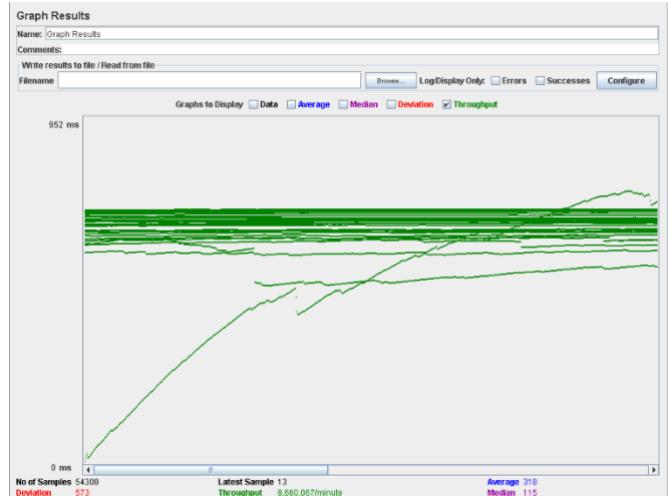
D12 – Tests de rendimiento

Hemos cogido el valor de 30 para las repeticiones de cada acción por parte de cada usuario para poder comparar el rendimiento de los usuarios entre los tests de forma adecuada, siendo este un valor que no crea un excesivo tiempo de espera y hace que la gráfica de Throughput se estabilice.

Caso de uso 001: Registrar actor

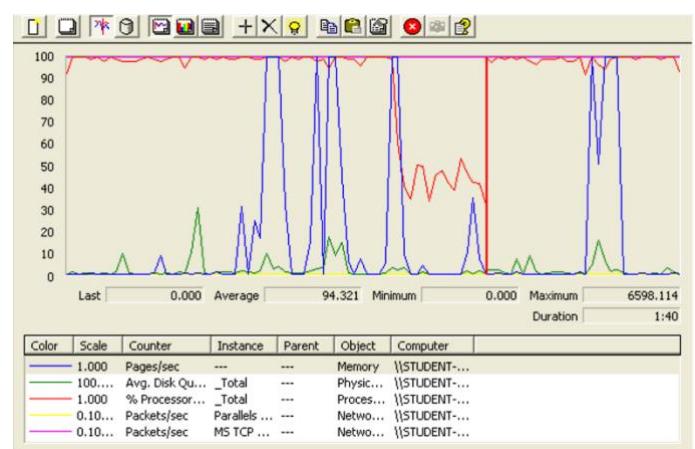
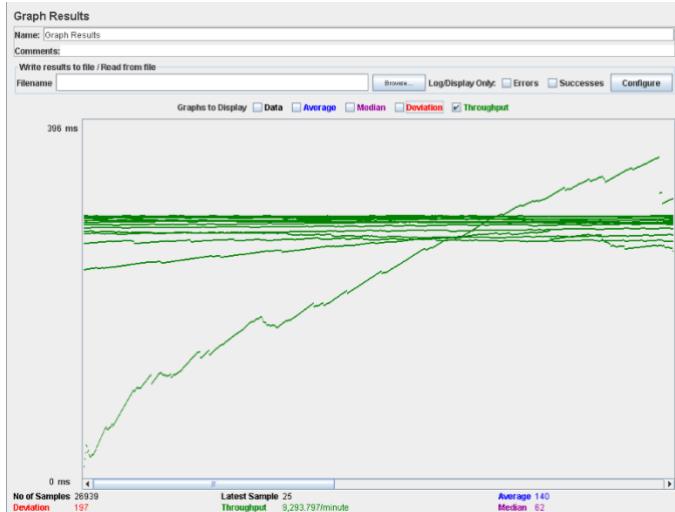
Procedemos a realizar el test del caso de uso de crear un actor en el sistema, en este caso crearemos un usuario. Comenzamos con la cifra de 100 usuarios concurrente y el usuario en principio puede sin problemas.

Aggregate Report								
Name: Aggregate Report								
Comments:								
Write results to file / Read from file								
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput
/	8909	854	615	1046	24	8912	0.00%	17.7/sec
scriptshelpers	35115	160	55	315	1	3915	0.00%	9.1/sec
scriptshelpers...	29517	132	54	373	2	3925	0.00%	8.4/sec
scriptshostst...	35111	143	64	307	3	3930	0.00%	9.0/sec
scriptsjquery...	3498	161	82	334	4	4077	0.00%	9.0/sec
rhymescircle.cs...	1496	142	53	267	2	3930	0.00%	9.0/sec
rhymescothdra...	3489	189	97	371	5	3999	0.00%	9.0/sec
rhymescothdra...	3489	135	55	256	2	3916	0.00%	9.0/sec
rhymescookies...	3477	133	55	274	2	3904	0.00%	9.0/sec
rhavon.ico	3472	122	53	263	2	3921	0.00%	8.9/sec
rhavon.ico	3458	235	143	492	9	4040	0.00%	8.9/sec
rhavonlink_br...	3453	389	243	816	25	4502	0.00%	8.9/sec
rhavonlink_br...	3453	120	22	200	2	3999	0.00%	8.9/sec
actorunregister...	3418	558	302	1239	12	6481	0.00%	8.9/sec
actorunregister...	3399	757	589	1411	27	6557	0.00%	8.9/sec
TOTAL	55527	316	115	802	1	8812	0.00%	142.0/sec



Aumentamos la cantidad hasta los 150 usuarios donde, como podemos observar en las imágenes de abajo, vemos unos tiempos que se sitúan en el límite que consideramos aceptable que sería un segundo. Por ende, podemos decir que nuestro sistema para este caso de uso soporta alrededor de 150 usuarios concurrentes.

Como vemos, al ser una acción de creación tiene un impacto mayor en el uso del disco, pero como es habitual la tarjeta de red y el procesador se encuentran en sus máximos.



Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

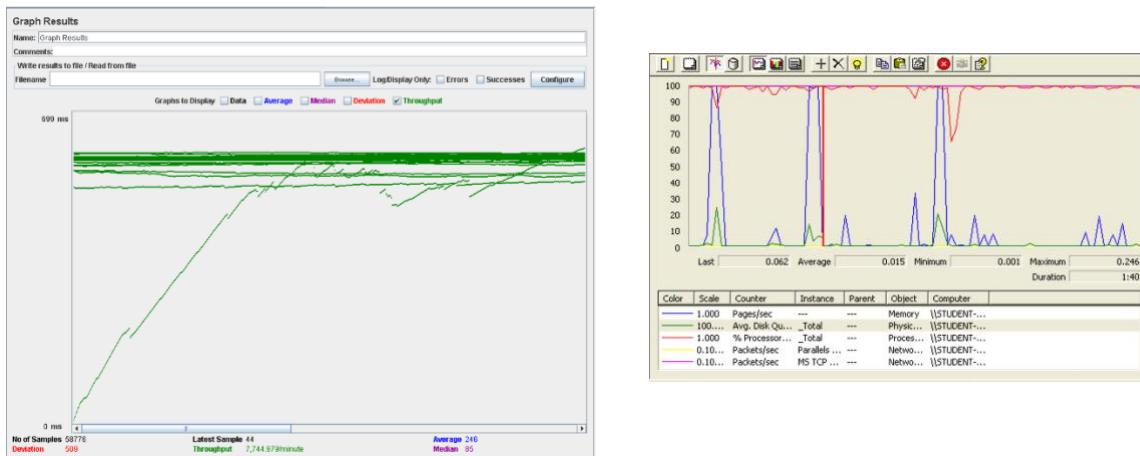
Filename Log/Display Only: Errors Successes

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3268	377	322	667	30	3009	0.00%	19.9/sec	213.0
/scripts/helpers...	1681	45	32	81	2	547	0.00%	10.3/sec	23.5
/scripts/polyfills...	1677	39	30	73	3	515	0.00%	10.3/sec	77.7
/scripts/bootstrap...	1671	55	41	104	3	523	0.00%	10.3/sec	376.9
/scripts/jquery....	1669	79	63	145	6	638	0.00%	10.3/sec	983.0
/styles/circle.css	1666	42	31	91	2	373	0.00%	10.3/sec	4.8
/styles/bootstrap...	1656	90	74	165	5	817	0.00%	10.3/sec	1220.6
/styles/style.css	1651	40	31	80	2	487	0.00%	10.3/sec	16.1
/styles/cookiec....	1646	42	32	83	2	538	0.00%	10.3/sec	42.4
/favicon.ico	1643	43	30	86	2	822	0.00%	10.3/sec	57.3
/images/link_br...	1640	176	135	329	10	1247	0.00%	10.2/sec	3337.5
/images/logo.jpg	1630	295	229	590	31	1862	0.00%	10.2/sec	11485.0
/scripts/cookiec...	1618	29	16	58	3	690	0.00%	10.3/sec	352.4
/actor/user/crea...	1606	173	125	357	12	3118	0.00%	10.3/sec	100.7
/actor/user/edit...	1594	354	305	615	30	2922	0.00%	10.3/sec	101.9
TOTAL	26316	140	62	376	2	3118	0.00%	160.2/sec	17938.7

Caso de uso 002: Editar actor

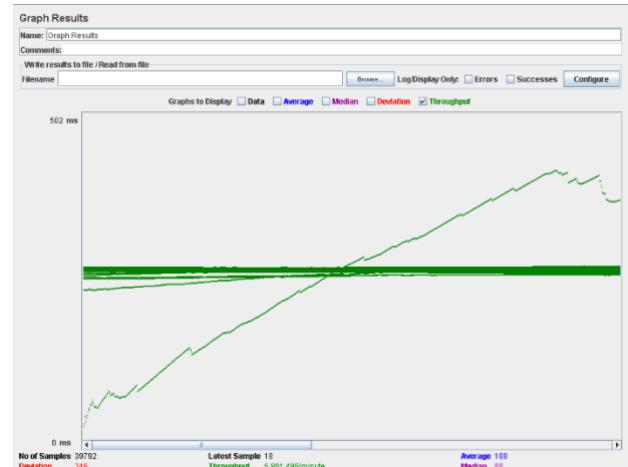
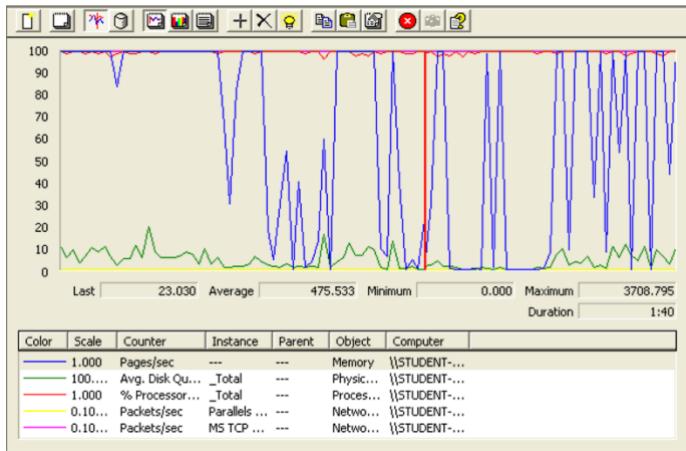
Procedemos a medir el rendimiento del sistema en el caso de uso en el que editamos un actor, en este caso iniciaremos sesión como una compañía y cambiaremos algunos datos. Comenzamos con 115 usuarios y vemos muy buenos resultados, pero vamos a probar algunos usuarios más para poner a prueba al sistema.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Filename										
Browse... Log/Display Only <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input checked="" type="checkbox"/> Configure										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	10608	476	328	951	20	15508	0.00%	23.3/sec	265.3	
/scripts/polyfills..	2710	49	28	95	2	2457	0.00%	6.0/sec	45.1	
/stylesheets..	2705	78	42	154	5	2534	0.00%	6.0/sec	710.5	
/scripts/bootstrap..	2702	61	33	118	3	2522	0.00%	6.0/sec	218.1	
/stylesheets/cookie..	2699	49	28	99	2	2480	0.00%	6.0/sec	24.7	
/scripts/bootstrap/	2694	69	44	155	4	1745	0.00%	6.0/sec	570.9	
/stylesheets.css	2694	47	29	90	2	2393	0.00%	6.0/sec	6.3	
/stylesheets-circle.css	2691	43	27	90	1	2526	0.00%	6.0/sec	2.8	
/favicon.ico	2684	50	28	104	2	2461	0.00%	5.9/sec	33.2	
/imageseslink_br..	5311	92	49	192	8	2586	0.00%	11.7/sec	3823.1	
/imagesesLogo.jpg	2681	197	136	395	28	2660	0.00%	6.0/sec	6713.7	
/scripts/cookies..	2680	58	37	119	3	2394	0.00%	5.9/sec	203.9	
/scripts/helpers..	2679	31	11	59	2	2040	0.00%	6.0/sec	13.6	
/sessions/login.db	2676	283	157	570	6	2135	0.00%	6.0/sec	37.3	
il_spring_secu..	2667	666	408	1275	43	18915	0.00%	6.0/sec	60.1	
/actor/company..	2647	308	214	734	14	14053	0.00%	6.0/sec	61.4	
/actor/company..	2638	781	598	1423	65	20677	0.00%	6.0/sec	69.1	
il_spring_secu..	2610	498	357	954	30	10985	0.00%	5.9/sec	64.6	
TOTAL..	58776	246	85	637	1	2135	0.00%	129.1/sec	12803.7	



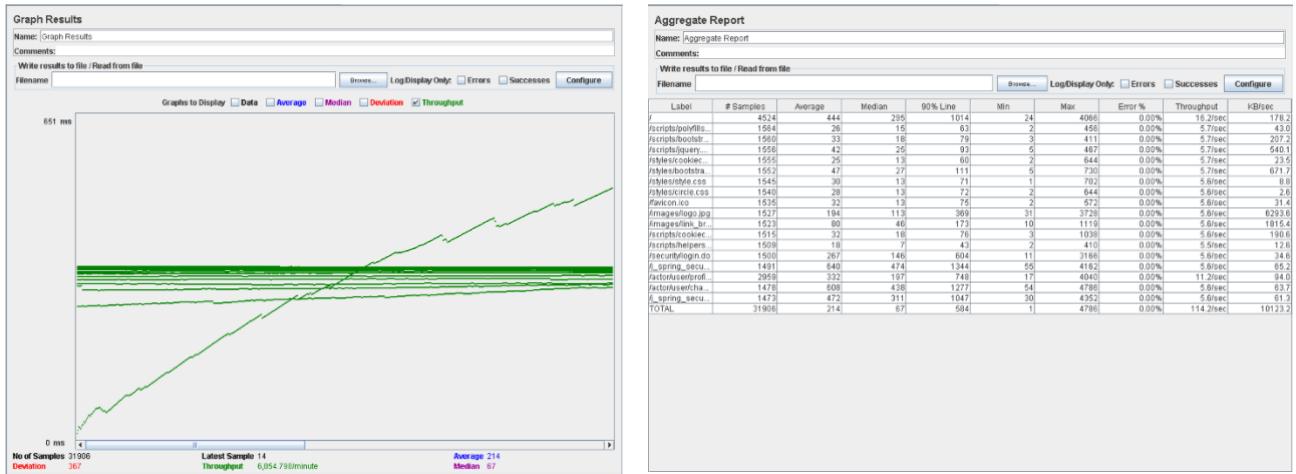
Subimos hasta los 140 usuarios, donde ya vemos que se supera con creces el límite que exigimos el rendimiento del sistema. Por ende, podemos concluir que 115 son los usuarios concurrentes que soporta el sistema para este caso de uso. El throughput sería 6564. En cuanto a la gráfica del perfmon.exe, sacamos las mismas conclusiones que en el anterior test: el uso del disco duro sube más que en otras ocasiones debido a que es una operación de escritura.

Aggregate Report										
Name: Aggregate Report Comments: Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	7148	358	296	683	23	2617	0.00%	18.0/sec	196.5	
/scripts/polyfills...	1841	47	38	90	3	937	0.00%	4.7/sec	35.2	
/styles/bootstrap...	1839	87	71	181	5	1159	0.00%	4.7/sec	553.5	
/scripts/bootstrap...	1838	57	45	108	5	1089	0.00%	4.7/sec	170.4	
/scripts/cookiec...	1835	42	32	74	2	940	0.00%	4.7/sec	19.3	
/scripts/jquery....	1835	80	62	142	6	1161	0.00%	4.7/sec	444.3	
/styles/style.css	1834	41	32	80	2	839	0.00%	4.7/sec	7.3	
/styles/circle.css	1830	38	29	71	2	681	0.00%	4.7/sec	2.2	
/favicon.ico	1829	39	30	76	2	1005	0.00%	4.7/sec	26.0	
/images/link_br...	3590	121	92	239	10	1652	0.00%	9.1/sec	2974.3	
/images/logo.jpg	1827	295	234	538	42	1781	0.00%	4.7/sec	5250.3	
/scripts/cookiec...	1822	50	41	96	4	908	0.00%	4.7/sec	160.4	
/scripts/helpers...	1819	19	11	41	2	888	0.00%	4.7/sec	10.6	
/security/login.do	1803	142	86	324	12	1613	0.00%	4.7/sec	28.1	
/j_spring_secu...	1797	366	289	719	51	2504	0.00%	4.6/sec	53.0	
/factor/company...	1777	230	172	477	18	2020	0.00%	4.6/sec	47.4	
/factor/company...	1771	580	515	1035	52	2827	0.00%	4.6/sec	53.5	
/j_spring_secu...	1757	398	343	734	49	2095	0.00%	4.6/sec	50.5	
TOTAL	39792	188	88	495	2	2827	0.00%	99.7/sec	9921.8	

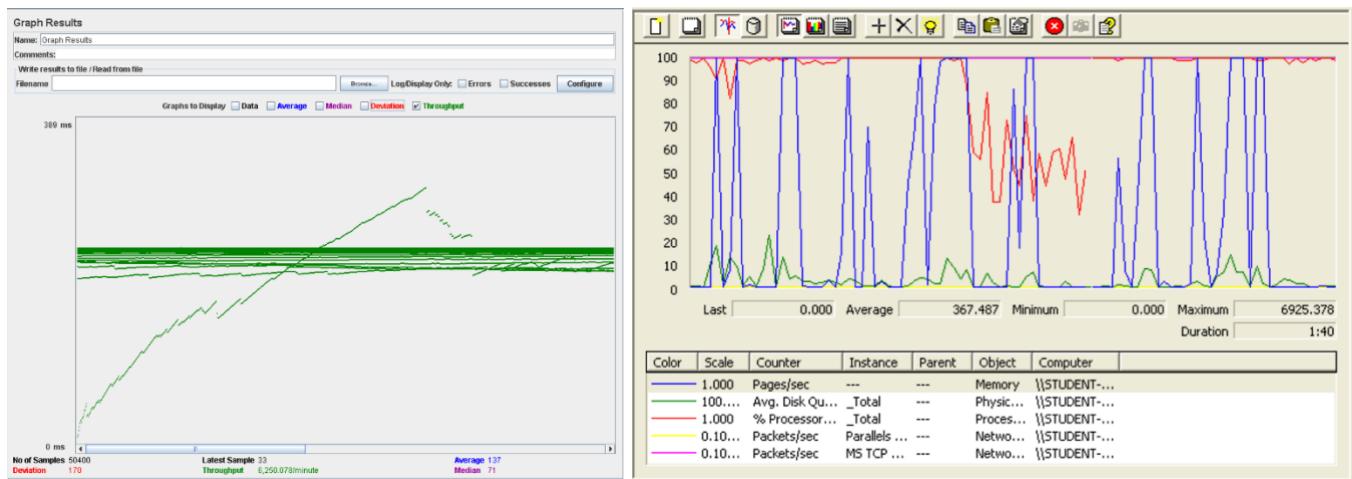


Caso de uso 003: Publicar/ocultar lista de deseos

Procedemos a medir el rendimiento en el caso de uso que consiste en cambiar el estado de la lista de deseos del usuario. Comenzamos con 100 usuarios y como vemos en el aggregate report los tiempos superan el segundo.



Bajamos el número de usuarios hasta los 80. El resultado nos muestra que los tiempos entran dentro del límite, por lo cual, podemos concluir que el sistema para este caso de uso soporta como mucho 80 usuarios. El throughput es 6250 y en el estado del sistema que arroja el perfmon nos muestra como el disco duro ha aumentado más que en otros casos de uso y que el procesador está la mayor parte del tiempo del 100%.



Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename

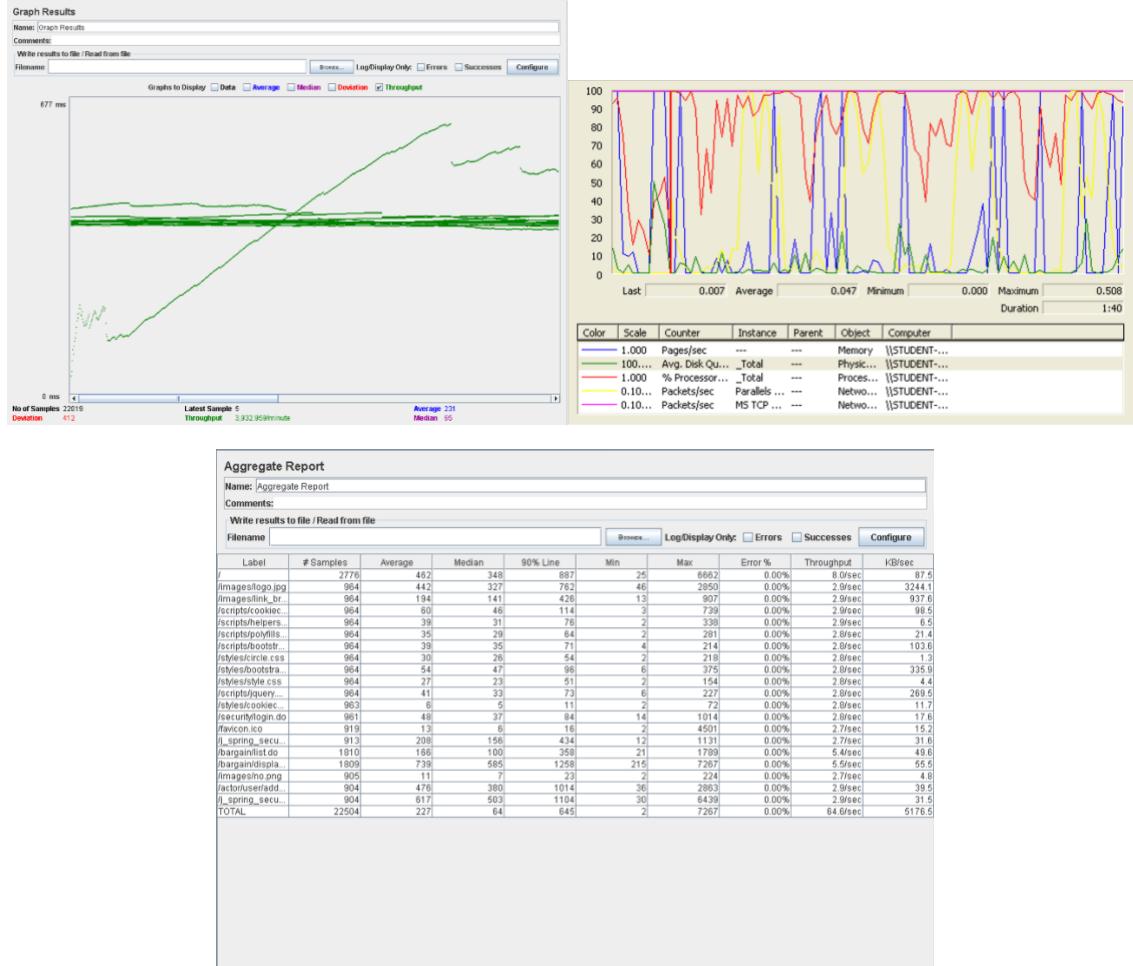
Browse...

Log/Display Only: Errors Successes Configure

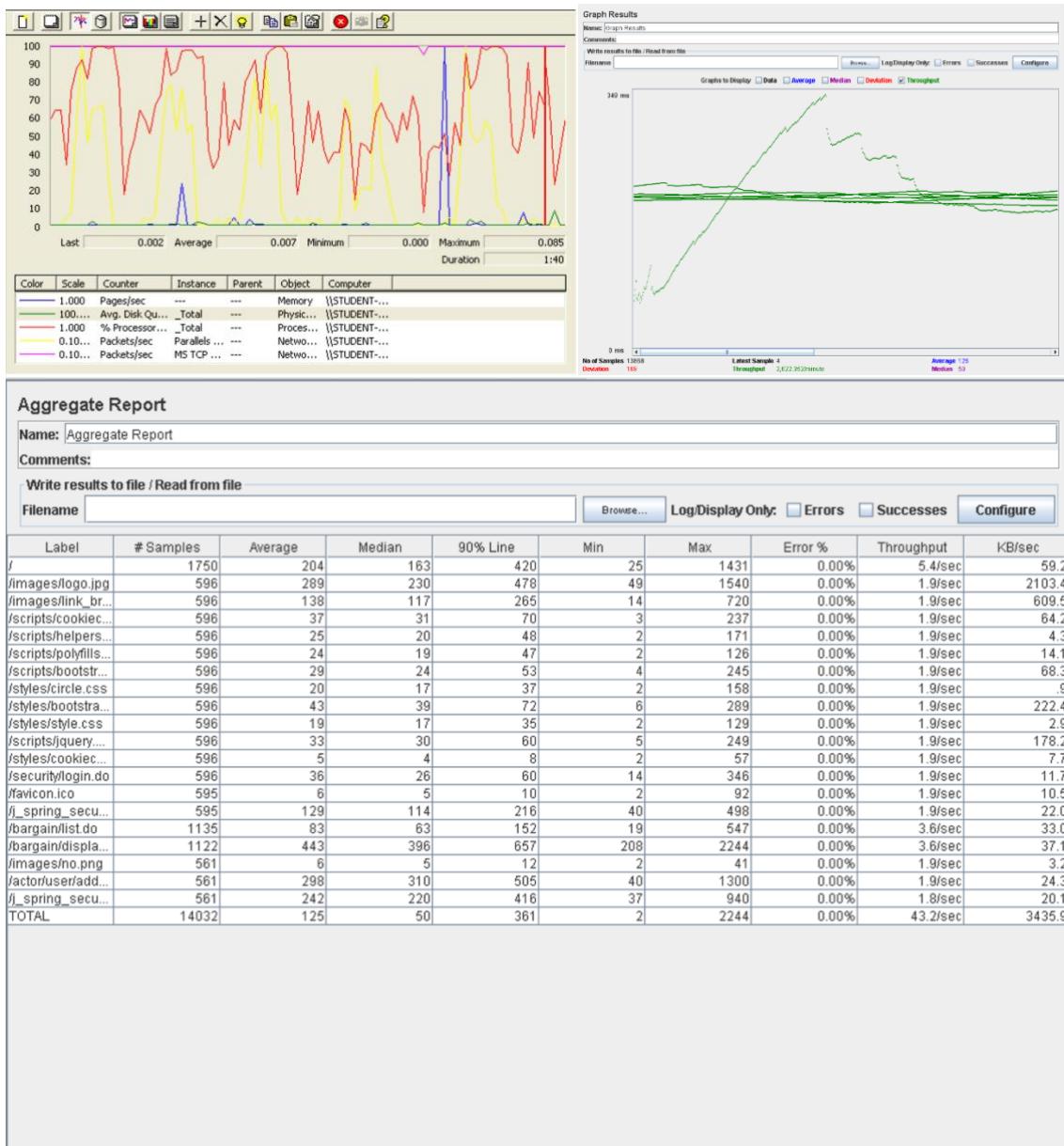
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	256	210	492	28	2525	0.00%	14.9/sec	163.3
/scripts/polyfills...	2400	35	27	69	2	388	0.00%	5.1/sec	38.2
/scripts/bootstr...	2400	41	32	79	3	389	0.00%	5.1/sec	185.2
/scripts/jquery....	2400	58	47	106	5	840	0.00%	5.1/sec	483.2
/styles/cookiec....	2400	32	23	62	2	494	0.00%	5.1/sec	21.0
/styles/bootstrap...	2400	69	53	128	5	1171	0.00%	5.1/sec	602.6
/styles/style.css	2400	33	23	68	2	648	0.00%	5.1/sec	7.9
/styles/circle.css	2400	32	22	66	2	796	0.00%	5.1/sec	2.4
/favicon.ico	2400	32	22	68	2	682	0.00%	5.1/sec	28.4
/images/logo.jpg	2400	249	184	469	36	2897	0.00%	5.1/sec	5734.7
/images/link_br...	2400	128	102	240	12	1707	0.00%	5.1/sec	1663.7
/scripts/cookiec...	2400	44	34	82	3	667	0.00%	5.1/sec	175.7
/scripts/helpers...	2400	16	10	37	2	318	0.00%	5.1/sec	11.7
/security/login.do	2400	109	73	251	10	1112	0.00%	5.1/sec	32.0
/j_spring_secu...	2400	324	282	584	9	1878	0.00%	5.1/sec	60.1
/actor/user/profil...	4800	165	126	344	15	1300	0.00%	10.2/sec	86.2
/actor/user/cha...	2400	300	248	549	31	1844	0.00%	5.1/sec	59.8
/j_spring_secu...	2400	284	246	523	31	1593	0.00%	5.1/sec	56.1
TOTAL	50400	137	71	352	2	2897	0.00%	104.2/sec	9177.2

Caso de uso 004: Añadir/quitar chollo de lista de deseo

Procedemos a comprobar el rendimiento que arroja el sistema para el caso de uso en el que se añade/quita un chollo de la lista de deseo de un usuario. Comenzamos con 60 usuarios puesto que este test tiene bastante contenido. El resultado nos muestra que a pesar de ser un nº de usuarios tan bajo, aún salen tiempos por encima del segundo.



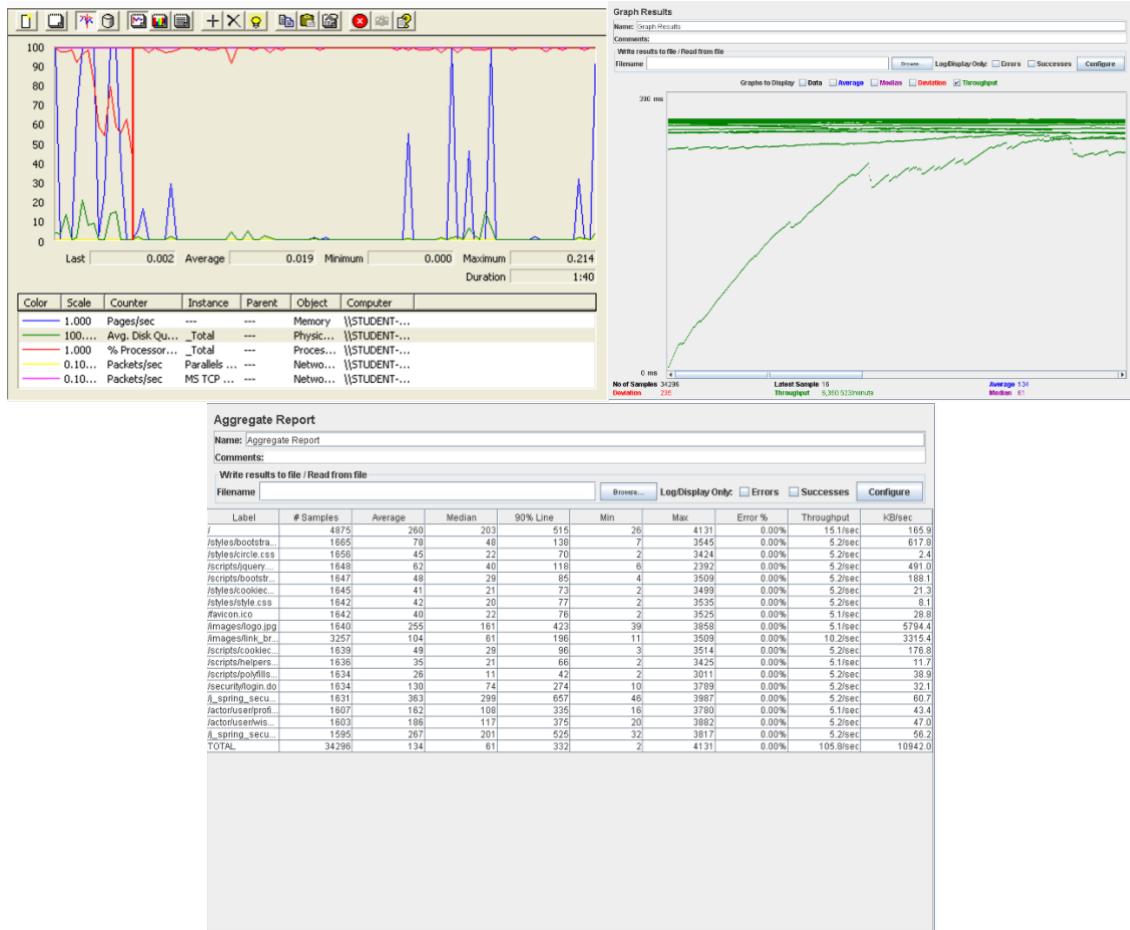
Bajamos hasta los 45 usuarios, y vemos que ahora sí los tiempos son aceptables. Por ende, concluimos que el sistema para este caso de uso soporta 45 usuarios concurrentes. Como vemos en las imágenes de abajo, el throughput será de 2622 y en perfmon vemos que de nuevo es la tarjeta de red la que se encuentra al 100% de su capacidad.



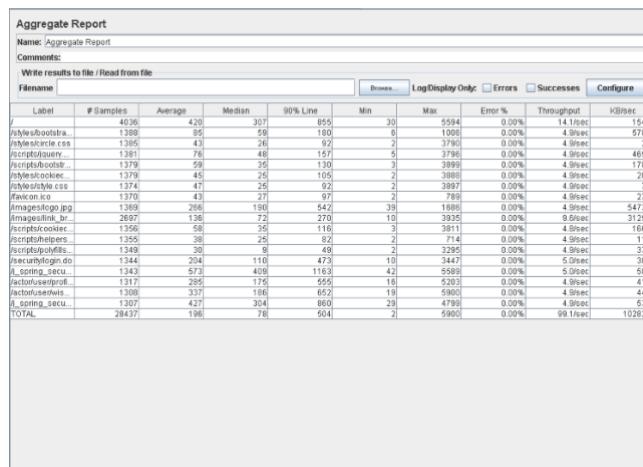
Caso de uso 005: Listar lista de deseo

Procedemos a realizar el test de rendimiento sobre listar la lista de deseo de un usuario.

Comenzamos el test con 80 usuarios. Esto nos arroja unos tiempos muy buenos dentro de los límites.

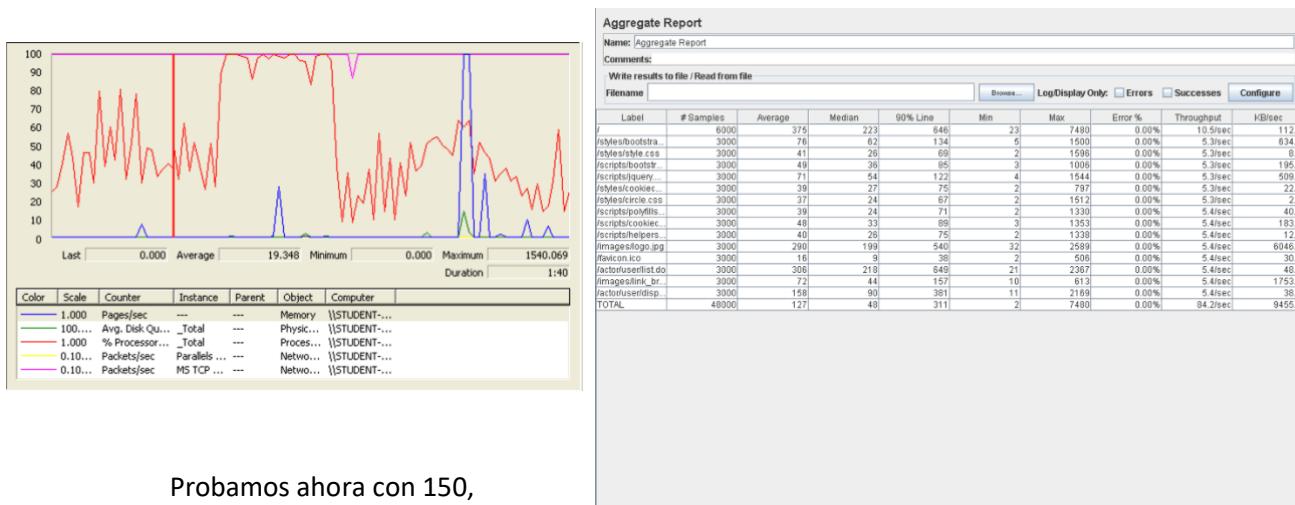
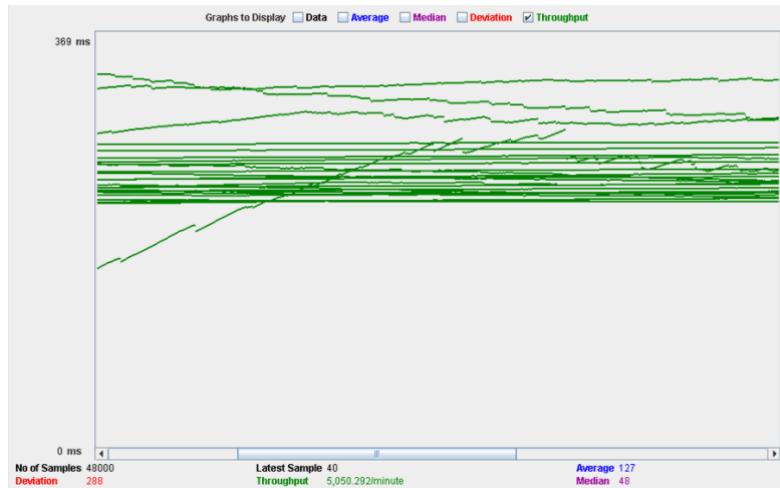


Aumentamos a 90 ese número de usuarios concurrentes y vemos como el tiempo ya se sale y se pasa del segundo máximo que buscamos. Es por ello que podemos concluir que el sistema para este caso se usó aguanta 80 usuarios concurrentes. El throughput sería 6350 y en cuanto al comportamiento del sistema vemos como la tarjeta de red y el procesador están al límite de su capacidad.

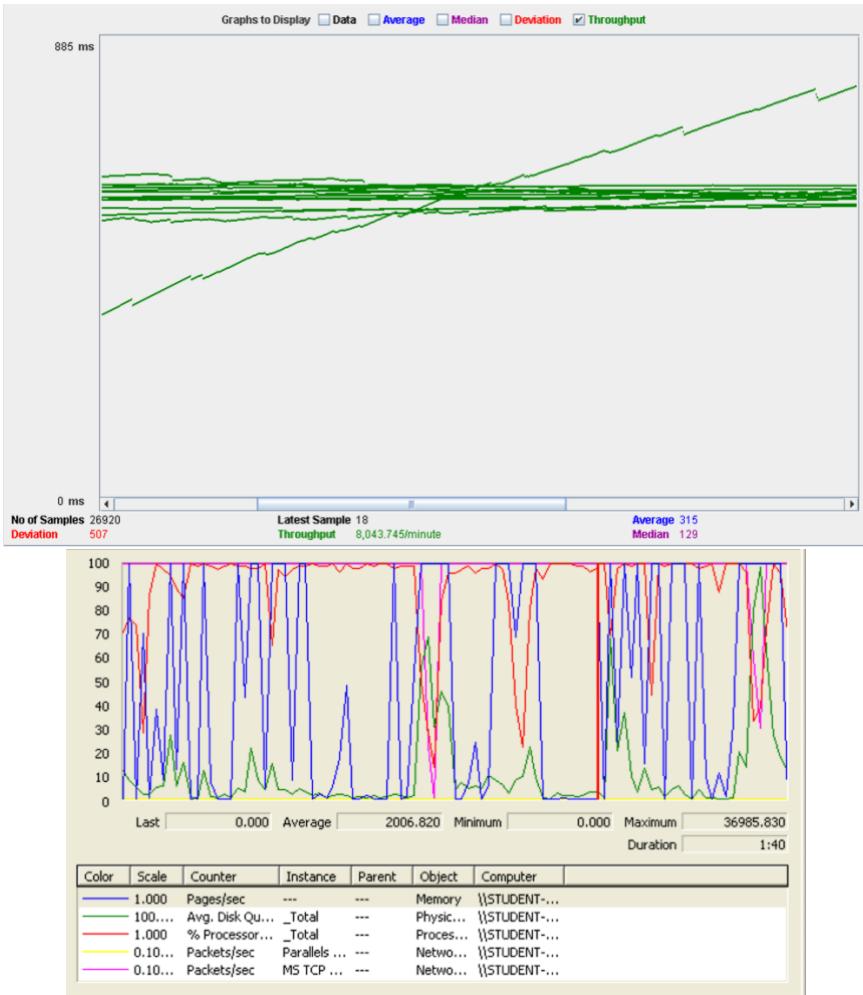


Caso de uso 006: Listar y desplegar usuarios

Procedemos a realizar el test de rendimiento de listar y desplegar un usuario del sistema. Comenzamos probando con 100 usuarios y los resultados que arrojan son bastante buenos.



Probamos ahora con 150, tratando de buscar el límite de la capacidad de nuestro servidor. En este caso vemos como el tiempo obtenido supera el segundo marcado como límite. Por ende, podemos concluir que el sistema para este caso de uso soporta 100 usuarios concurrentes. El throughput será 5050 y como vemos en la gráfica del estado del sistema el procesador se mantiene al 100% de su uso.



Aggregate Report

Name: Aggregate Report

Comments:

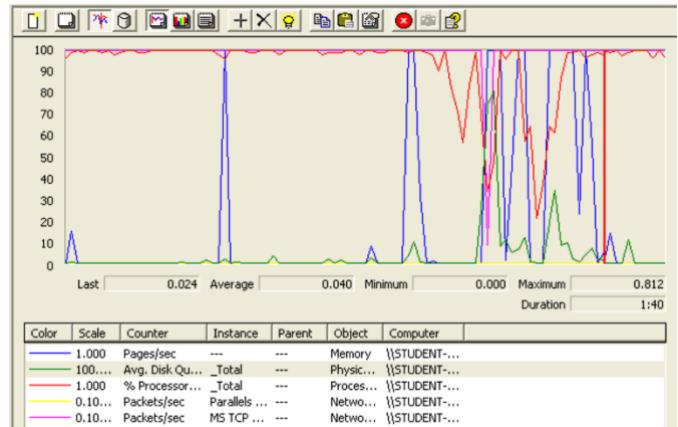
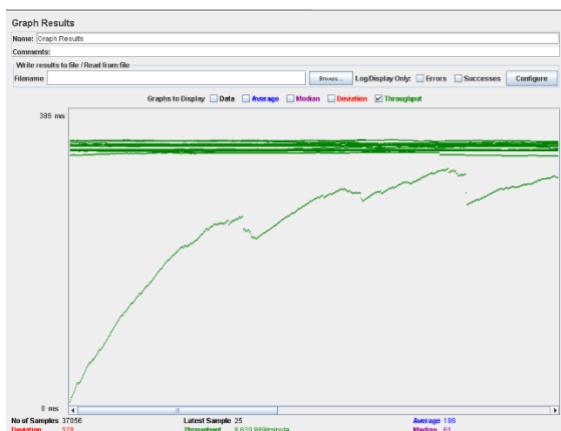
Write results to file / Read from file

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3542	700	554	1302	29	4726	0.00%	16.6/sec	177.4
/styles/bootstrap...	1835	174	129	340	6	2950	0.00%	8.7/sec	1034.8
/styles/style.css	1830	105	63	217	2	3304	0.00%	8.7/sec	13.7
/scripts/bootstrap...	1820	104	78	220	3	1101	0.00%	8.7/sec	317.1
/scripts/jquery....	1805	163	112	317	4	3308	0.00%	8.6/sec	821.6
/styles/cookiec...	1797	128	67	239	2	3333	0.00%	8.6/sec	35.7
/styles/circle.css	1793	142	71	259	2	3339	0.00%	8.6/sec	4.1
/scripts/polyfills...	1790	143	70	268	2	3322	0.00%	8.6/sec	64.8
/scripts/cookiec...	1787	171	87	314	3	3319	0.00%	8.6/sec	296.0
/scripts/helpers...	1778	141	77	309	2	3384	0.00%	8.6/sec	19.6
/images/logo.jpg	1770	530	348	1030	29	4004	0.00%	8.5/sec	9623.0
/favicon.ico	1738	141	27	276	2	3295	0.00%	8.5/sec	47.4
/actor/user/list.do	1722	823	598	1806	33	4455	0.00%	8.4/sec	75.5
/images/link_br...	1700	324	113	685	9	3422	0.00%	8.6/sec	2805.7
/actor/user/displ...	1700	541	353	1238	17	4219	0.00%	8.5/sec	61.4
TOTAL	28407	311	128	799	2	4726	0.00%	132.8/sec	14821.6

Caso de uso 007: Listar y desplegar compañías

Procedemos a realizar el test en el que veremos el rendimiento del sistema a la hora de listar y desplegar compañías. Comenzamos con 105 usuarios y los tiempos que nos muestra el aggregate report indican que aún el 90% de los usuarios aguantan más de 1 segundo esperando.

Aggregate Report										
Name:		Aggregate Report								
Comments:										
Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	4568	546	209	1069	18	25394	0.00%	10.0/sec	192.5	
/scripts/bootstrap...	2323	61	40	136	3	3164	0.00%	9.2/sec	336.3	
/scripts/query...	2326	81	51	161	5	1639	0.00%	9.4/sec	891.2	
/styles/bootstrap...	2326	98	56	179	6	3229	0.00%	9.2/sec	1095.3	
/styles/style.css	2320	58	29	108	2	3169	0.00%	9.2/sec	14.4	
/styles/circle.css	2317	56	28	105	2	3168	0.00%	9.2/sec	4.3	
/styles/cookiec...	2312	59	29	117	2	3174	0.00%	9.2/sec	37.9	
/favicon.ico	2303	58	29	117	2	3160	0.00%	9.2/sec	51.2	
/images/link.br...	2298	151	95	302	11	2813	0.00%	9.1/sec	2975.0	
/images/logo.jpg	2290	273	177	503	30	4631	0.00%	9.1/sec	10250.3	
/scripts/cookiec...	2282	59	37	122	3	1710	0.00%	9.1/sec	311.1	
/scripts/helpers...	2279	47	28	106	2	3020	0.00%	9.1/sec	20.7	
/scripts/polyfills...	2278	34	16	62	2	2526	0.00%	9.1/sec	68.8	
/actor/company...	2267	433	195	921	18	17738	0.00%	9.1/sec	82.8	
/actor/company...	2253	384	156	811	12	21887	0.00%	9.1/sec	63.7	
TOTAL	36747	183	61	397	2	25394	0.00%	144.6/sec	16241.9	



Bajamos hasta los 95 usuarios y ahora sí los tiempos no superan el segundo por lo que podemos concluir de que el sistema para este caso de uso aguanta como mucho 95 usuarios. El throughput es 8075 y el estado del sistema durante el desarrollo del test nos muestra que el cuello de botella se encuentra en el procesador y la tarjeta de red.



Aggregate Report

Name: Aggregate Report

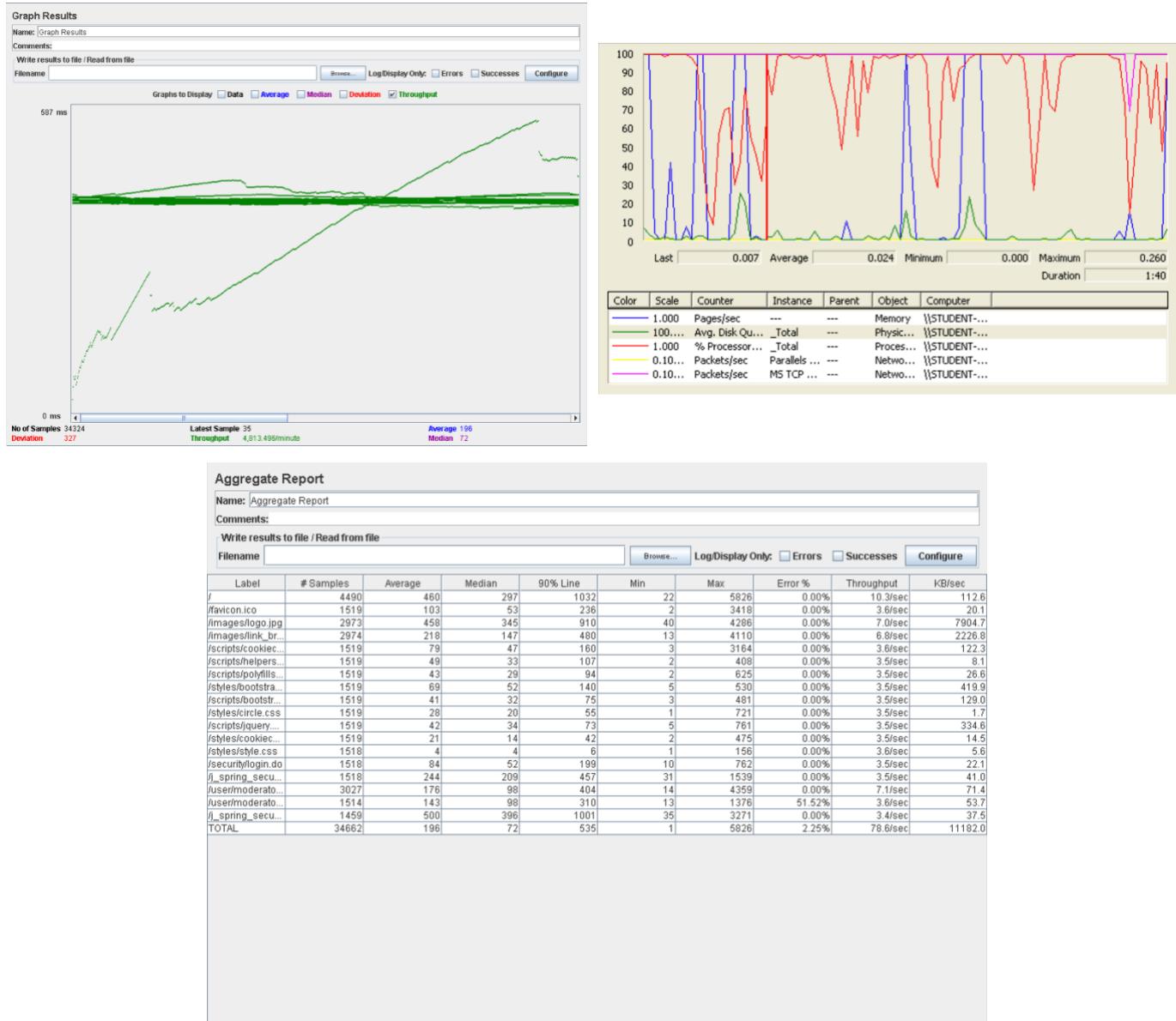
Comments:

Write results to file / Read from file

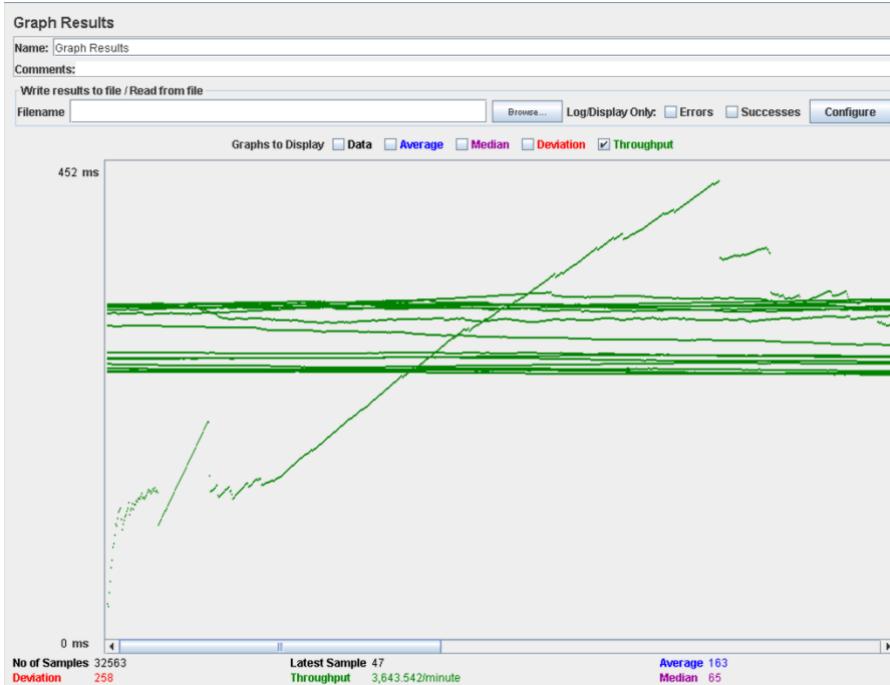
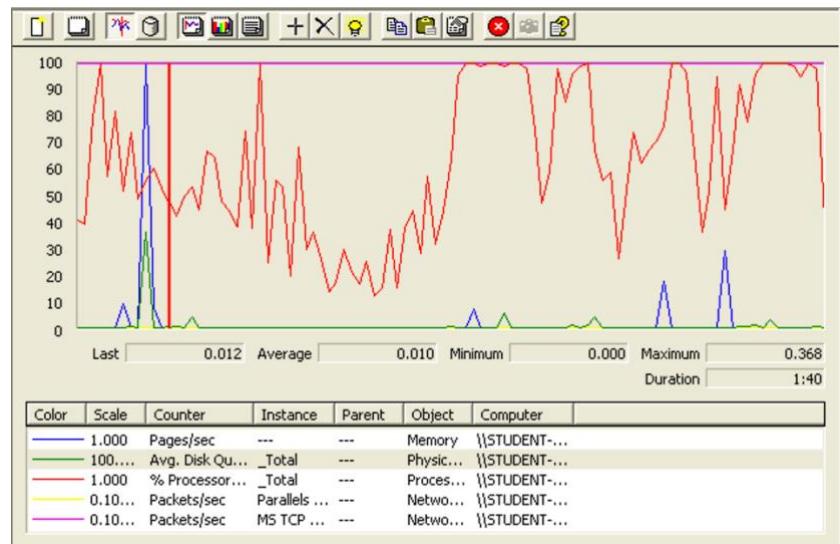
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	2834	327	248	642	32	2714	0.00%	17.0/sec	182.5
/scripts/bootstrap...	1440	63	43	122	3	2125	0.00%	8.8/sec	322.0
/scripts/jquery...	1439	80	56	179	5	2095	0.00%	8.8/sec	838.6
/styles/bootstrap...	1431	99	59	229	6	2126	0.00%	8.8/sec	1041.3
/styles/style.css	1429	51	32	98	2	2076	0.00%	8.8/sec	13.7
/styles/circle.css	1411	50	31	97	2	1826	0.00%	8.7/sec	4.1
/styles/cookiec...	1409	46	29	90	2	1916	0.00%	8.7/sec	35.9
/favicon.ico	1407	50	31	104	2	773	0.00%	8.7/sec	48.5
/images/link_b...	1406	197	102	416	11	1963	0.00%	8.7/sec	2826.3
/images/logo.jpg	1404	389	202	801	39	3107	0.00%	8.7/sec	9792.2
/scripts/cookiec...	1403	67	44	131	3	1926	0.00%	8.7/sec	298.5
/scripts/helpers...	1401	45	28	99	2	2092	0.00%	8.7/sec	19.9
/scripts/polyfills...	1401	28	12	60	2	1101	0.00%	8.7/sec	65.8
/actor/company...	1392	231	128	522	16	2762	0.00%	8.8/sec	80.0
/actor/company...	1387	213	110	466	11	2665	0.00%	8.9/sec	61.8
TOTAL	22594	141	59	343	2	3107	0.00%	134.6/sec	15082.1

Caso de uso 008: Banear/desbanear usuario

Procedemos a realizar el test de rendimiento del caso de uso en el que el moderador banea/desbanea un usuario. Comenzamos con 65 usuarios y podemos ver en las siguientes imágenes que el límite de un segundo que hemos marcado es superado en la página principal (/).



Bajamos a 55 usuarios y vemos ahora que los tiempos han bajado, pero aun así la petición de logo.jpg sigue cercano al segundo; vamos a ignorar este hecho porque en sí esto es un fallo ajeno a la programación puesto que la imagen no ha sido comprimida y de ahí el tiempo que tarda. Concluimos, por ende, que el sistema para este caso de uso soporta hasta 55 usuarios concurrentes. El throughput es 3643 y el perfmon muestra que el procesador suele estar en bastante uso durante la mayor parte del test y que la tarjeta de red está continuamente al 100% del uso.



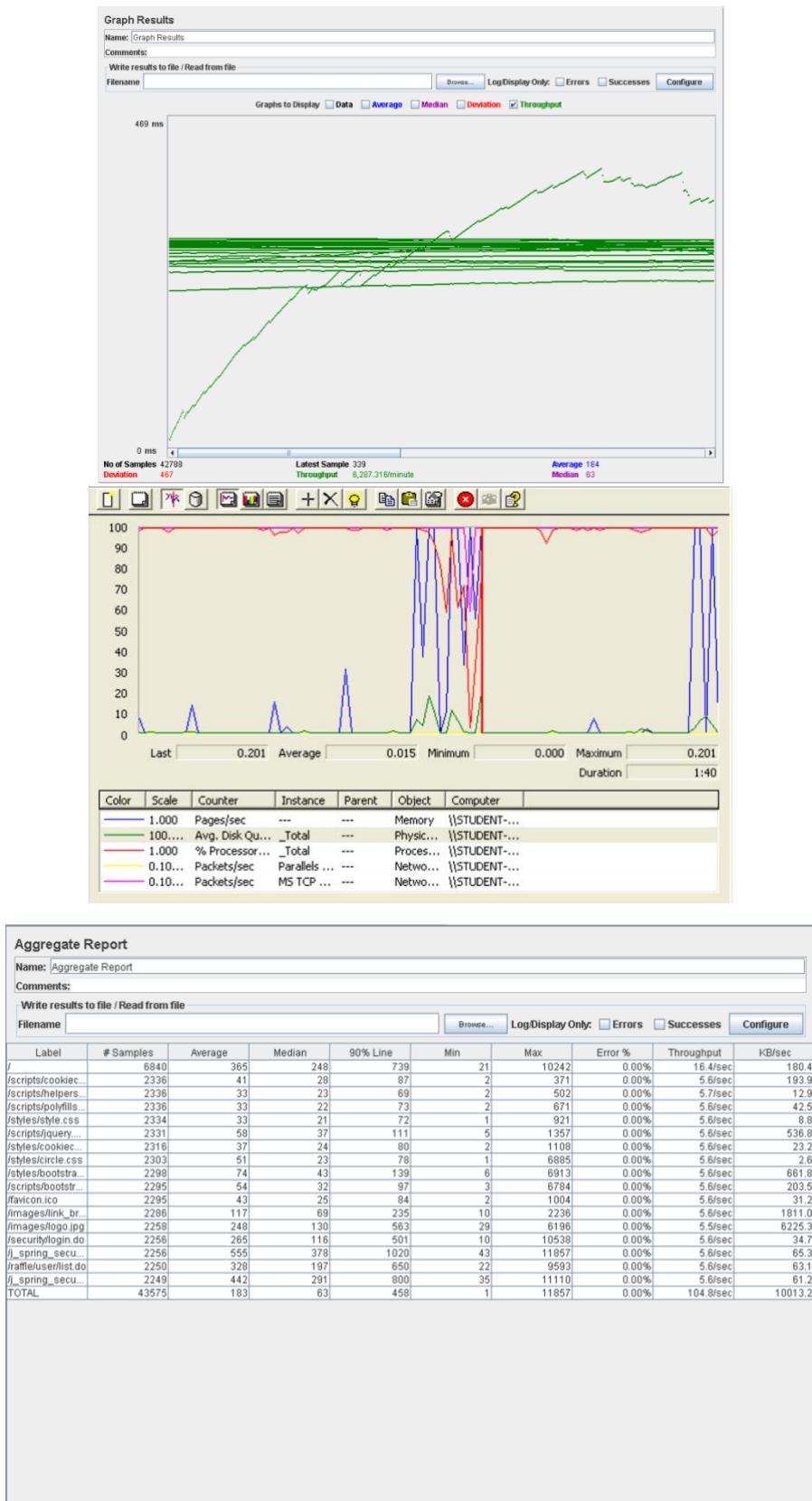
Aggregate Report

Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only Errors Successes Configure

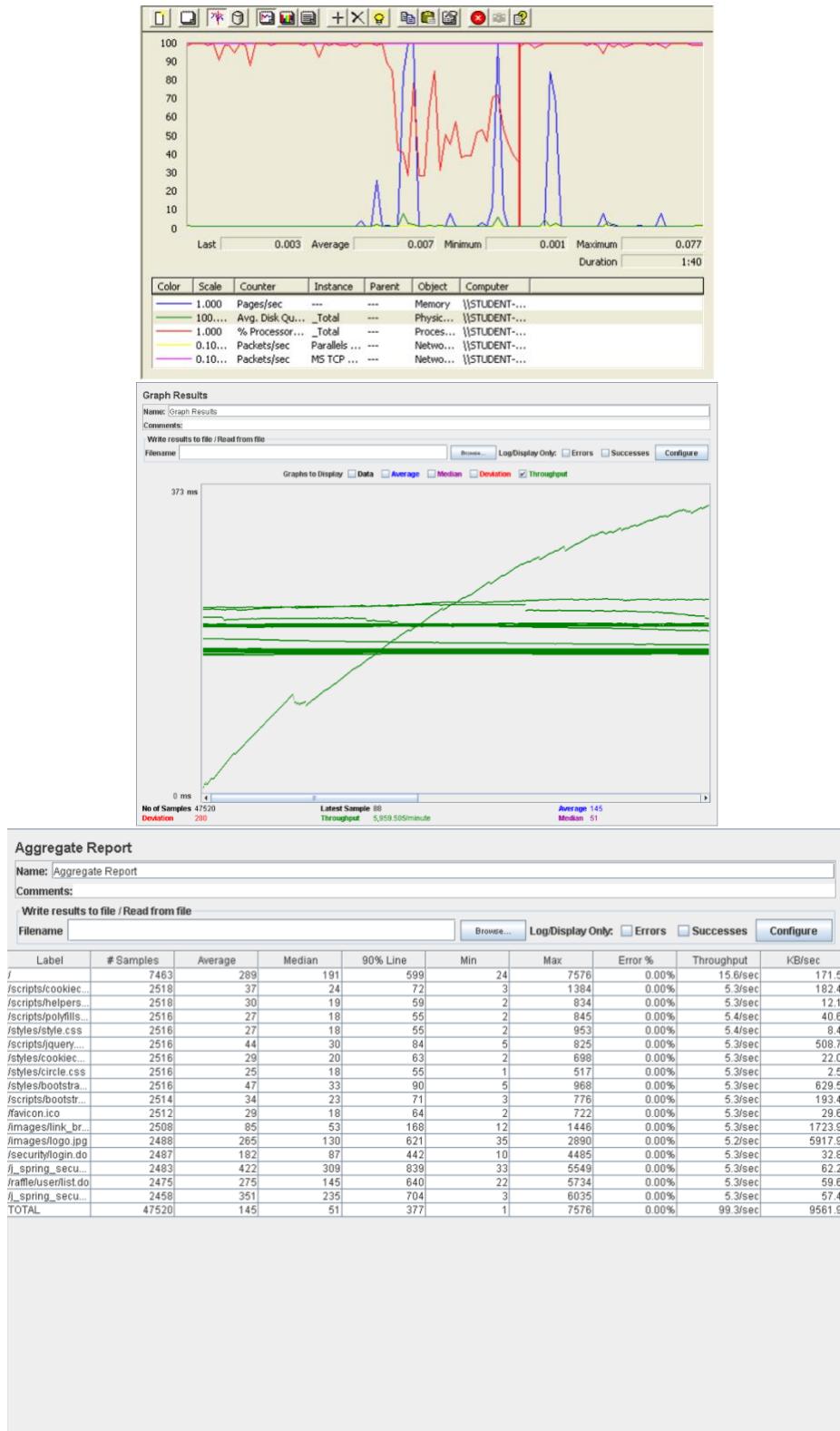
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4233	333	235	697	20	4344	0.00%	7.9/sec	86.8
/favicon.ico	1427	59	35	101	1	2083	0.00%	2.7/sec	15.0
/images/logo.jpg	2802	463	356	987	38	3632	0.00%	5.3/sec	5955.4
/images/link_br...	2825	193	160	391	14	2420	0.00%	5.3/sec	1743.2
/scripts/cookie...	1415	48	38	90	2	1694	0.00%	2.7/sec	92.1
/scripts/helpers...	1409	32	24	61	2	464	0.00%	2.7/sec	6.1
/scripts/polyfills...	1407	29	23	53	2	267	0.00%	2.7/sec	20.1
/styles/bootstrap...	1406	68	57	139	6	375	0.00%	2.7/sec	316.4
/scripts/bootstrap...	1405	40	32	67	3	3394	0.00%	2.7/sec	97.5
/styles/circle.css	1405	23	19	47	2	318	0.00%	2.7/sec	1.2
/scripts/jquery...	1405	50	41	98	5	876	0.00%	2.7/sec	253.9
/styles/cookiec...	1404	20	15	39	2	426	0.00%	2.7/sec	11.3
/styles/style.css	1404	5	4	8	2	152	0.00%	2.7/sec	4.3
/security/login.d...	1404	70	39	154	10	1843	0.00%	2.7/sec	17.0
/j_spring_secu...	1404	223	169	450	15	2074	0.00%	2.7/sec	31.6
/user/moderato...	2808	128	74	260	14	4441	0.00%	5.4/sec	54.7
/j_spring_secu...	1404	127	96	247	10	1596	50.21%	2.7/sec	40.7
/j_spring_secu...	1404	372	303	701	28	5529	0.00%	2.7/sec	29.6
TOTAL	32371	162	65	436	1	5529	2.18%	60.7/sec	8718.9

Caso de uso 009: Listar y desplegar rifas (usuario)

Procedemos a medir el rendimiento del caso de uso en el que se listan las rifas para las que el usuario tiene tiques. Comenzamos con 90 usuarios, pero vemos que supera ligeramente el segundo que hemos marcado como límite.



Por ello bajamos un poco la cantidad de usuarios hasta los 85 y vemos que el resultado ya entra dentro de lo aceptable. Podemos entonces afirmar que el sistema para este caso de uso aguanta 85 usuarios concurrentes. El throughput es 5959 y el cuello de botella en este caso de uso se encuentra en el procesador y la tarjeta de red.



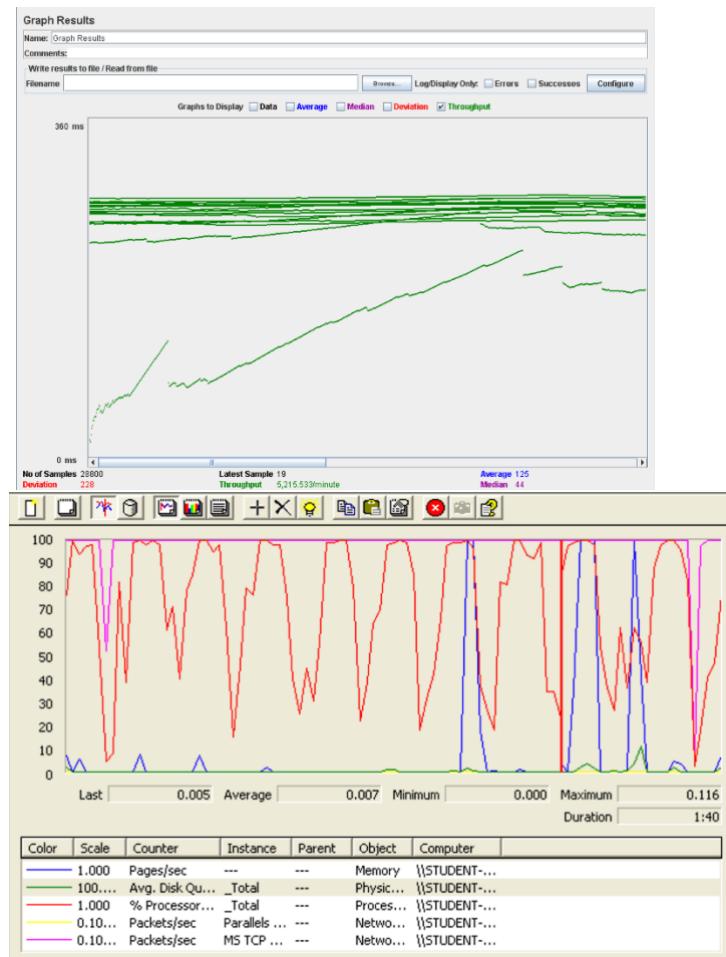
Caso de uso 009: Listar y desplegar rifas (sin autenticar)

Procedemos a medir el rendimiento del caso de uso en el que se listan y despliegan las rifas para los usuarios sin autenticar. Comenzamos con 75 diputados y vemos como los tiempos superan mínimamente el segundo límite.



Bajamos hasta los 60 usuarios y ahora sí vemos tiempos dentro del rango que consideramos aceptable, es por ello, que podemos concluir que para este caso de uso el sistema soporta hasta 60 usuarios concurrentes, el throughput es 5215 y por último el cuello de botella del

sistema se encuentra como es costumbre en la tarjeta de red y en este caso también el procesador está algo apurado.



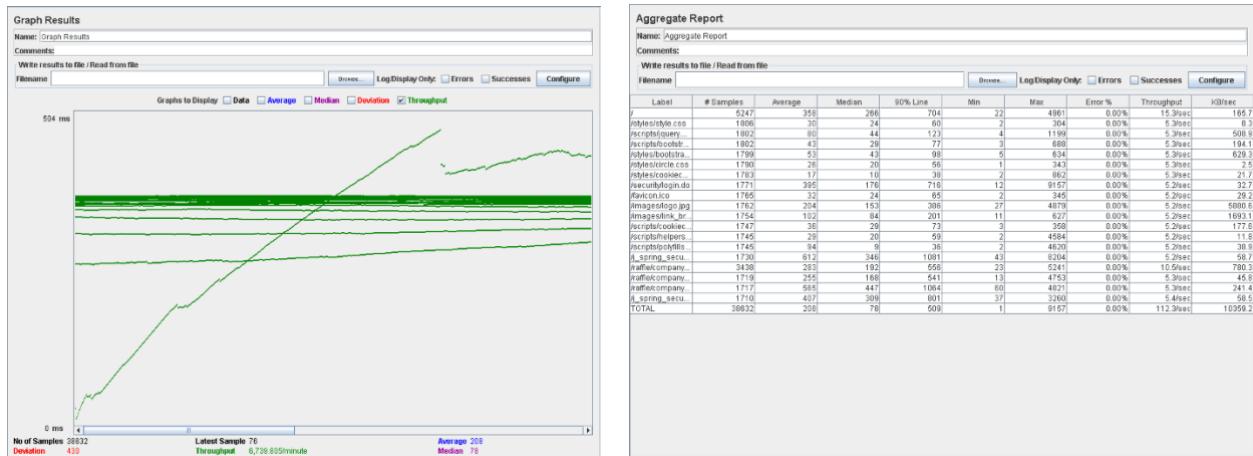
Aggregate Report

Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only Errors Successes Configure

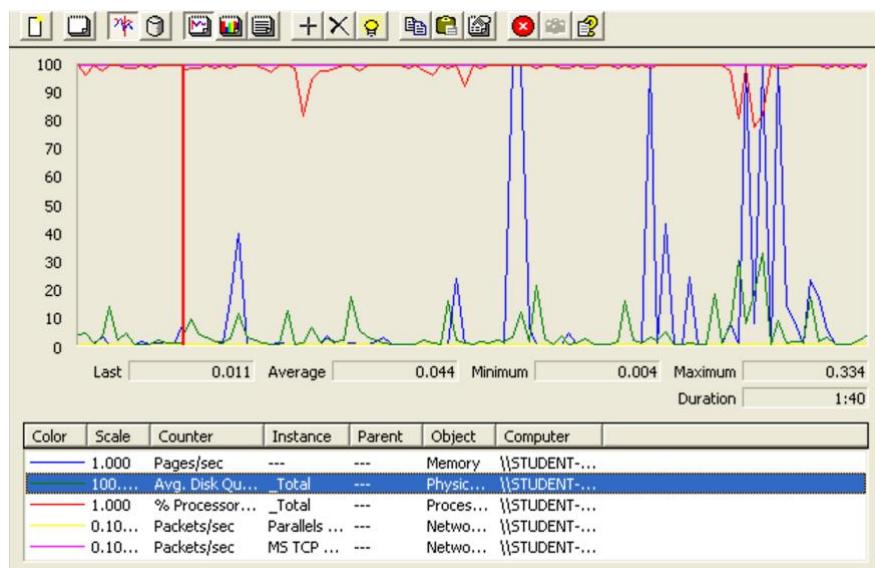
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/favicon.ico	3600	377	300	722	24	3914	0.00%	10.9/sec	116.3
/images/logo.jpg	1800	51	33	100	2	807	0.00%	5.6/sec	31.2
/scripts/helpers...	1800	51	39	103	3	546	0.00%	5.7/sec	194.1
/scripts/polyfills...	1800	40	26	78	2	902	0.00%	5.7/sec	12.9
/scripts/bootstrap...	1800	31	24	60	2	384	0.00%	5.7/sec	42.6
/scripts/jquery....	1800	42	33	82	3	559	0.00%	5.7/sec	206.5
/styles/bootstrap...	1800	64	50	121	6	441	0.00%	5.7/sec	672.1
/scripts/circle.css	1800	53	43	103	5	308	0.00%	5.7/sec	539.7
/styles/cookiec...	1800	22	15	47	2	645	0.00%	5.7/sec	2.7
/styles/style.css	1800	17	13	35	2	588	0.00%	5.7/sec	23.5
/raffle/list.do	1800	6	4	10	2	222	0.00%	5.7/sec	8.9
/raffle/display.do	1800	101	59	201	14	1452	0.00%	5.7/sec	58.8
/images/link.br...	1800	208	107	472	13	3970	0.00%	5.7/sec	40.5
TOTAL	28800	125	44	339	2	3970	0.00%	86.9/sec	9772.2

Caso de uso 010: Crear rifa

Procedemos a realizar el test de rendimiento del caso de uso de crear una rifa. Comenzamos con 100 usuarios concurrentes y los datos que nos muestra el Aggregate Report indican el 90% de los usuarios sufre una espera de más de un segundo al cargar página principal, iniciar sesión entre otras acciones.



Bajamos hasta 80 usuarios y los tiempos que ahora vemos son mucho mejores y entran dentro del rango que consideramos aceptable, por ende, podemos concluir que el sistema para este caso de uso soporta hasta 80 usuarios concurrentes. El throughput es 6485 y como vemos en la gráfica que nos muestra el perfmon durante la ejecución del test de rendimiento, el procesador se encuentra al máximo de su capacidad y el disco duro tiene más uso en este caso de uso porque estamos creando una rifa que tiene mucho que escribir en la base de datos.



Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename

Log/Display Only:

Errors

Successes

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	203	165	390	20	1708	0.00%	14.7/sec	160.1
/styles/style.css	2400	21	15	45	2	585	0.00%	5.0/sec	7.9
/scripts/jquery....	2400	45	34	82	4	676	0.00%	5.1/sec	482.5
/scripts/bootstrap...	2400	29	23	60	3	371	0.00%	5.1/sec	185.6
/styles/bootstrap...	2400	45	36	84	5	1386	0.00%	5.1/sec	606.2
/styles/circle.css	2400	20	14	44	1	277	0.00%	5.1/sec	2.4
/styles/cookiec....	2400	12	7	25	2	552	0.00%	5.1/sec	21.2
/security/login.do	2400	148	106	278	11	1219	0.00%	5.1/sec	32.0
/favicon.ico	2400	25	16	51	2	814	0.00%	5.1/sec	28.7
/images/logo.jpg	2400	152	118	271	27	1926	0.00%	5.1/sec	5777.5
/images/link_br...	2400	88	67	156	9	1357	0.00%	5.1/sec	1671.4
/scripts/cookiec...	2400	30	20	56	3	808	0.00%	5.1/sec	176.3
/scripts/helpers...	2400	19	13	41	2	339	0.00%	5.2/sec	11.7
/scripts/polyfills...	2400	11	7	21	2	479	0.00%	5.2/sec	38.8
/j_spring_secu...	2400	242	193	470	36	1421	0.00%	5.2/sec	58.8
/raffle/company...	4800	138	104	277	18	1787	0.00%	10.2/sec	346.7
/raffle/company...	2400	135	98	271	13	1484	0.00%	5.2/sec	44.7
/raffle/company...	2400	334	284	593	44	2214	0.00%	5.2/sec	132.3
/j_spring_secu...	2400	230	184	430	27	1980	0.00%	5.2/sec	56.3
TOTAL	52800	112	59	285	1	2214	0.00%	108.1/sec	9448.5

Include group name in label? Save Table Header

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

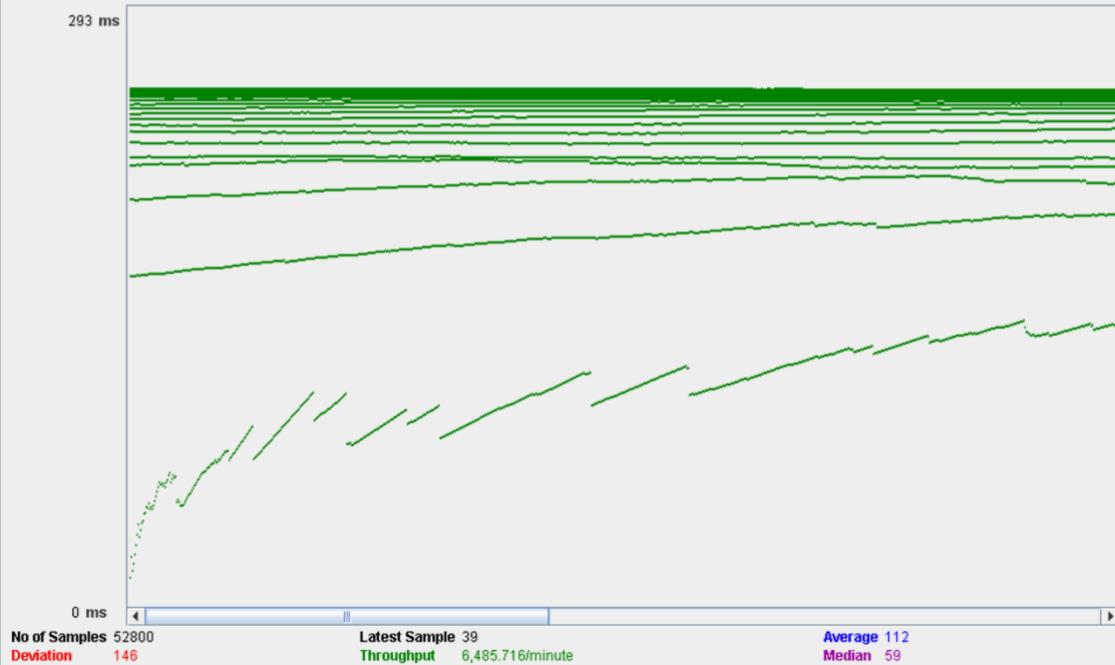
Filename

Log/Display Only:

Errors

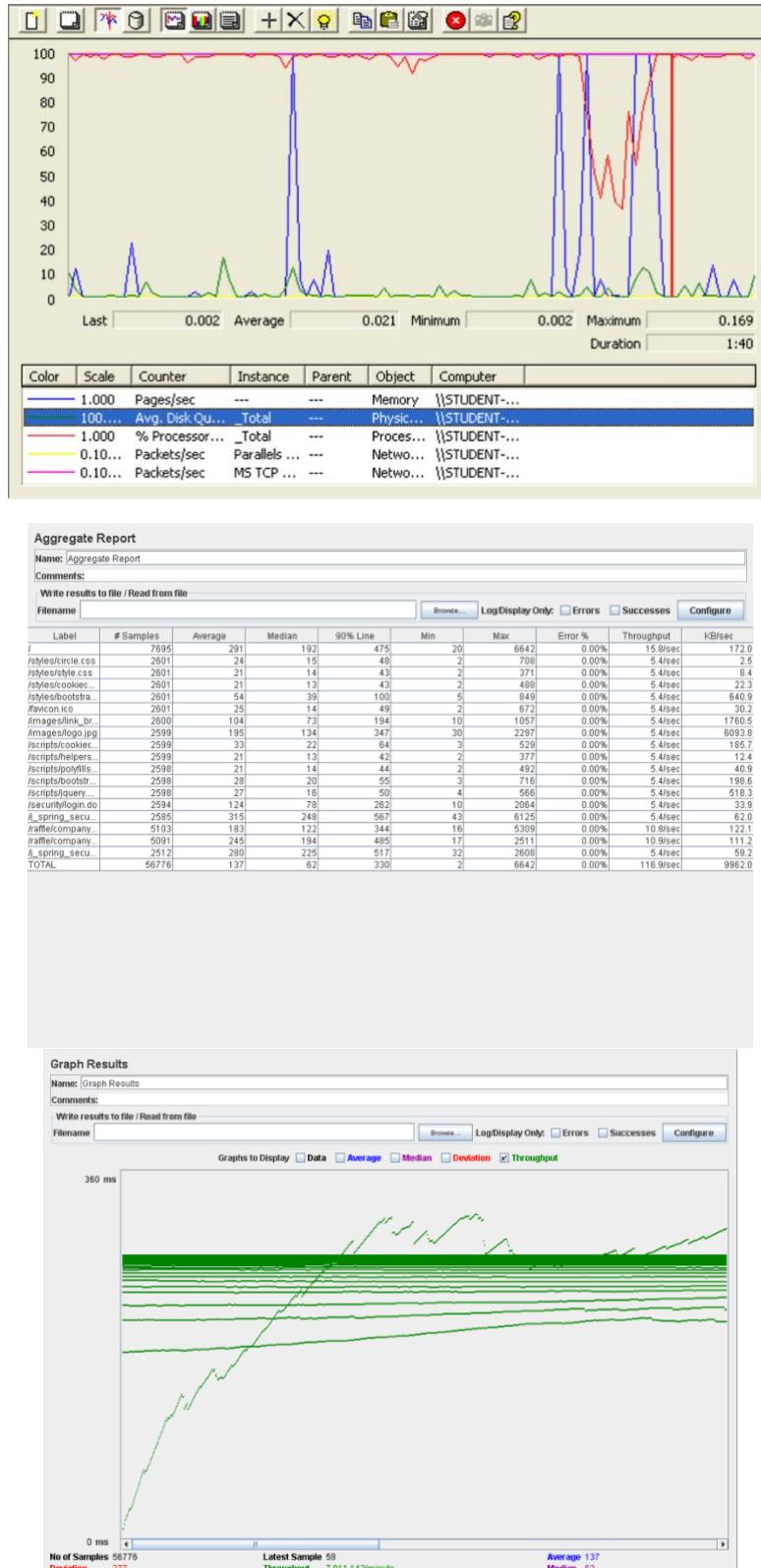
Successes

Graphs to Display Data Average Median Deviation Throughput

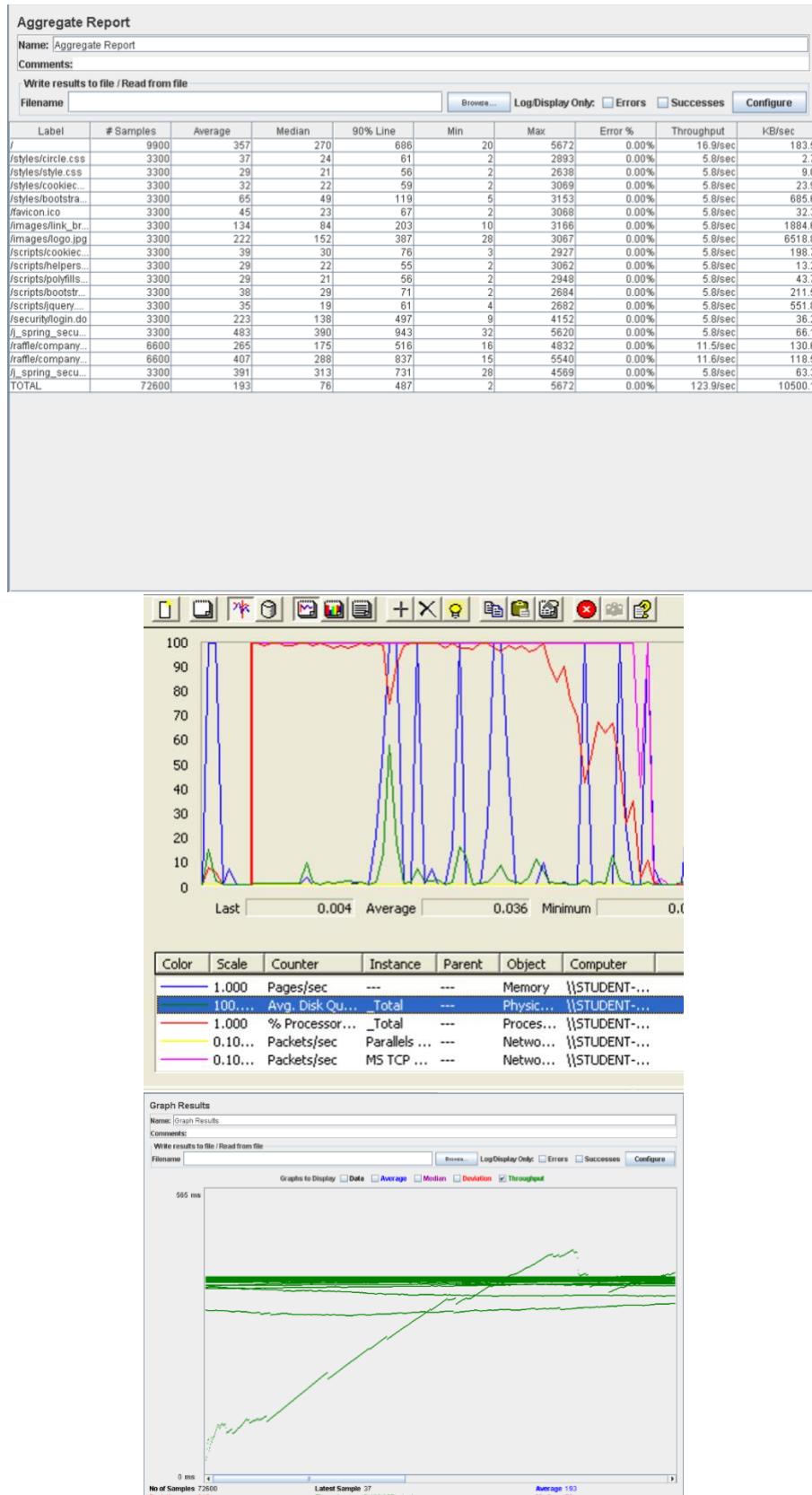


Caso de uso 011: Editar rifa

Procedemos a realizar el test de rendimiento del caso de uso de crear una rifa. Comenzamos con la cifra de 90 usuarios y los resultados que obtenemos son muy buenos pero no nos acercamos al límite que es lo que realmente queremos encontrar con estos test, hasta donde llega nuestro sistema.



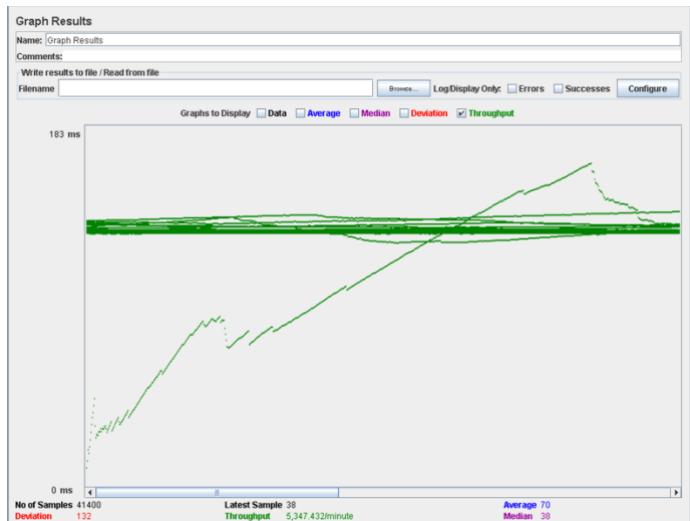
Subimos pues el número de usuarios hasta 110 y ya si vemos que está cerca del límite por lo que podemos concluir que hemos encontrado el límite del sistema en 80 usuarios concurrentes. El throughput es 7435 y el cuello de botella del sistema se encuentra en la tarjeta de red y el procesador.



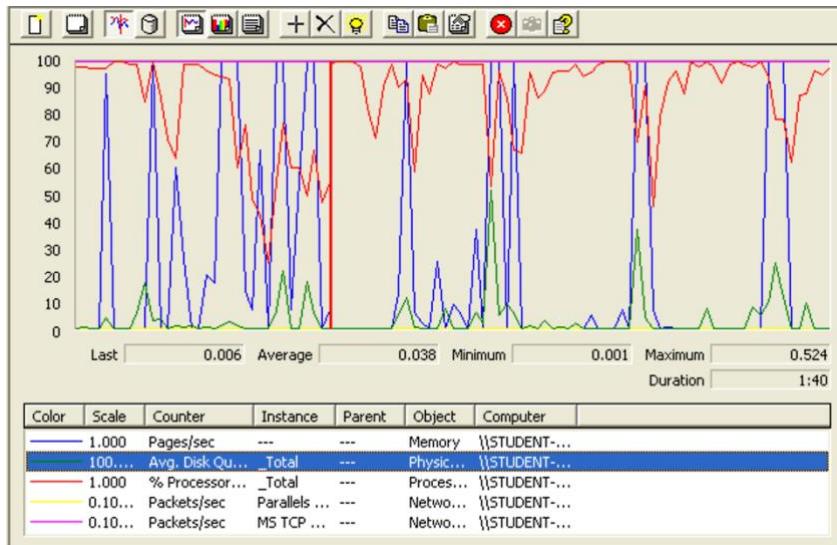
Caso de uso 012: Sortear rifa

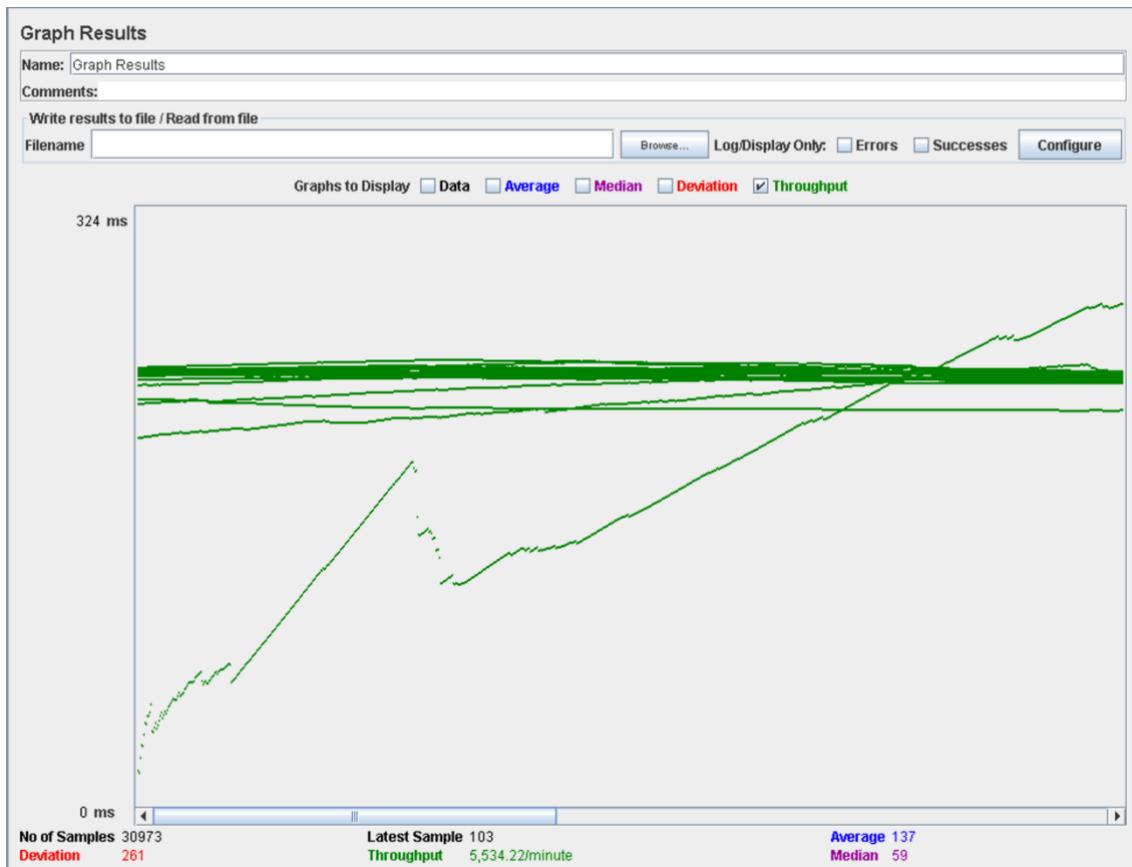
Procedemos a realizar el test de rendimiento del caso de uso de sortear una rifa. Comenzamos por 60 usuarios concurrentes y vemos que los tiempos son bajos por lo que vamos a aumentar el número de usuarios para ver donde está el límite.

Aggregate Report								
Name: Aggregate Report								
Comments:								
Write results to file / Read from file								
Filename: <input type="text"/> Browse... LogDisplay Only: <input type="checkbox"/> Errors: <input type="checkbox"/> Successes: <input type="checkbox"/> Configure								
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput
/Menciones.css	5400	137	85	287	16	4299	0.00%	11.9usec
/Recon.cgi	1800	26	12	45	2	2950	0.00%	4.0usec
/Images.cgi	1800	33	17	45	2	1953	0.00%	5.0usec
/ImagesLogo.jpg	1800	153	92	263	25	2312	0.00%	4.0usec
/ImagesLink.br	1800	76	60	141	9	1454	0.00%	4.0usec
/ScriptCookies	1800	26	14	41	2	1950	0.00%	4.0usec
/ScriptLinks	1800	17	11	36	2	956	0.00%	4.0usec
/ScriptStyle.css	1800	16	11	36	2	251	0.00%	4.0usec
/ScriptButtons	1800	23	14	45	3	1760	0.00%	4.0usec
/ScriptFormularia	1800	24	15	52	5	1771	0.00%	4.0usec
/ScriptLogueo	1800	29	21	55	4	1958	0.00%	4.0usec
/ScriptCookies	1800	16	10	29	2	1929	0.00%	4.0usec
/ScriptStyle.css	1800	9	4	14	2	944	0.00%	4.0usec
/ScriptButtons	1800	48	20	97	8	2400	0.00%	4.0usec
8. spring_secu.	1800	127	92	238	23	1676	0.00%	4.0usec
/afficherDetail	5400	88	54	159	16	3890	0.00%	11.9usec
/afficherdisplay.do	1800	67	47	195	14	1000	0.00%	4.0usec
8. spring_secu.	1800	71	52	143	13	598	0.00%	5.7usec
8. spring_secu.	1800	172	108	356	20	3108	0.00%	4.0usec
TOTAL	41400	70	38	155	2	4299	0.00%	89.1usec
								728.9



Aumentamos el número de usuarios hasta los 80, y vemos como el tiempo está más cercano del límite, por ende, podemos concluir que el sistema para este caso de uso soporta como mucho alrededor de 80 usuarios concurrentes, con un throughput de 5534. El perfmon arroja información sobre el uso del hardware del sistema y como vemos el uso de la tarjeta de red es total.





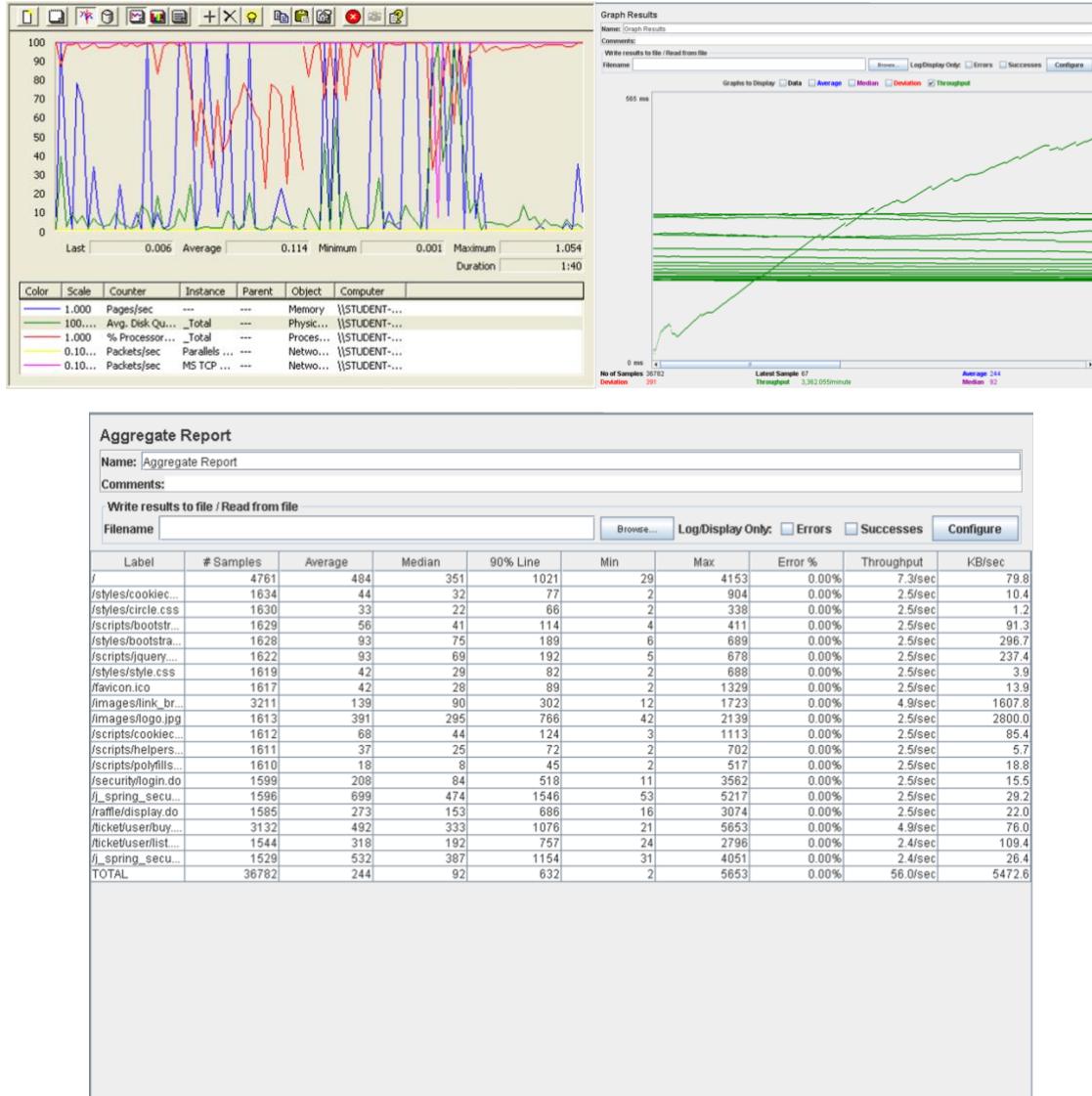
Aggregate Report

Name: Aggregate Report
Comments:
 Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

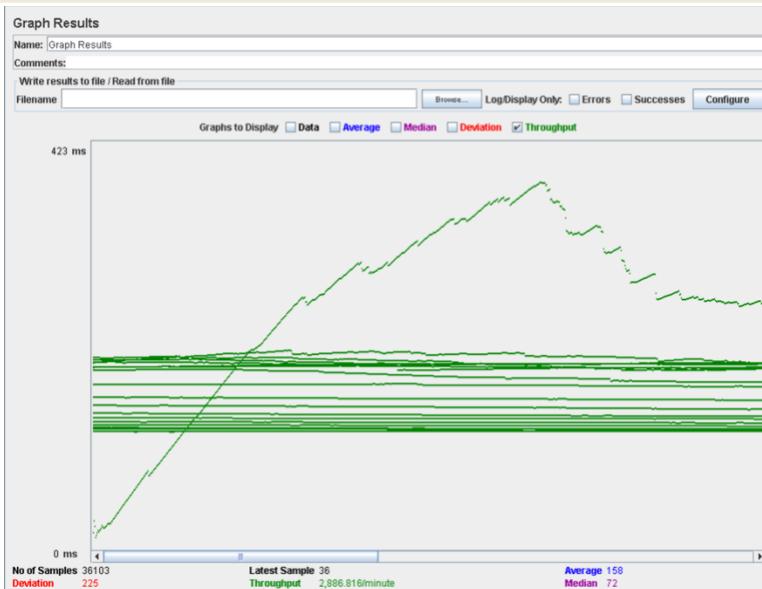
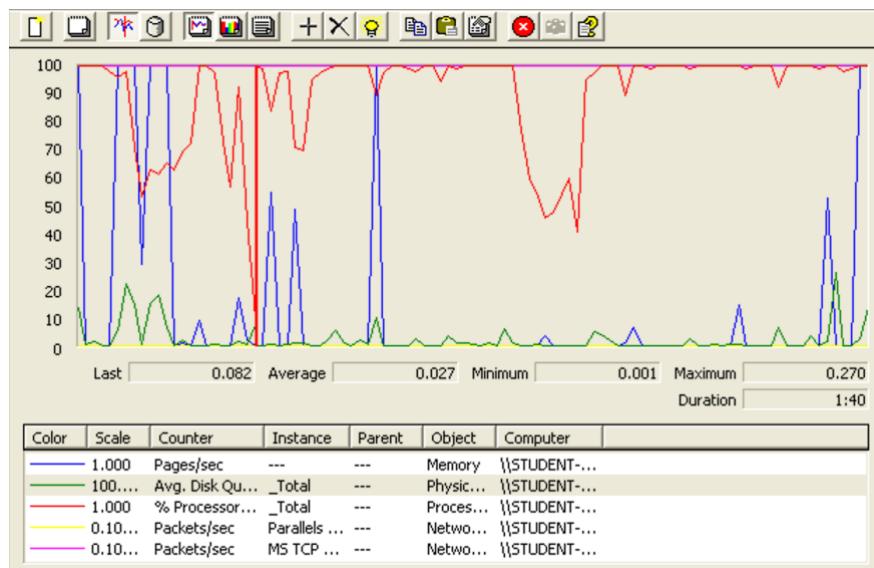
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4064	275	171	617	21	4449	0.00%	11.9/sec	130.4
/styles/circle.css	1392	49	28	87	2	2949	0.00%	4.1/sec	1.9
/favicon.ico	1377	46	26	76	2	3119	0.00%	4.1/sec	22.8
/images/logo.jpg	1370	358	191	751	31	3751	0.00%	4.1/sec	4569.7
/images/link_b...	1367	175	106	341	11	3105	0.00%	4.1/sec	1323.3
/scripts/cookiec...	1365	61	32	91	3	3072	0.00%	4.1/sec	139.9
/scripts/helpers...	1360	45	22	67	2	3097	0.00%	4.1/sec	9.3
/scripts/polyfills...	1359	32	21	59	2	2976	0.00%	4.1/sec	30.7
/scripts/bootstrap...	1357	44	29	94	2	2983	0.00%	4.1/sec	149.9
/styles/bootstrap...	1357	73	45	146	6	3163	0.00%	4.1/sec	485.0
/scripts/query....	1357	57	37	121	4	2977	0.00%	4.1/sec	387.3
/styles/cookiec...	1356	25	15	54	2	888	0.00%	4.1/sec	16.8
/styles/style.css	1353	15	4	17	2	2910	0.00%	4.0/sec	6.3
/security/login.do	1352	85	44	196	9	2413	0.00%	4.1/sec	25.3
/j_spring_secu...	1350	233	160	474	34	2295	0.00%	4.2/sec	48.2
/raffle/moderat...	4027	148	82	307	17	3709	0.00%	12.2/sec	148.2
/raffle/display.d...	1341	138	75	258	14	3653	0.00%	4.1/sec	30.6
/raffle/moderat...	1337	141	90	283	18	3817	0.00%	4.1/sec	59.6
/j_spring_secu...	1321	318	239	653	33	2508	0.00%	4.1/sec	44.6
TOTAL	31162	137	59	326	2	4449	0.00%	91.0/sec	7498.2

Caso de uso 013: Comprar tiques

Procedemos a medir el rendimiento del caso de uso que consiste en comprar un tique utilizando una tarjeta de crédito. Comenzamos con 110 usuarios y vemos como en varias partes se supera el segundo que hemos marcado como límite de lo aceptable.



Bajamos hasta los 70 usuarios y ya vemos unos tiempos mucho mejores dentro del rango que hemos estipulado aceptable, es decir, por debajo de un segundo para el 90% de los usuarios. Por ende, podemos concluir que el sistema para este caso de uso es capaz de soportar hasta 70 usuarios. El throughput es 2886 y el estado del sistema muestra que la tarjeta de red actúa como cuello de botella.



Aggregate Report

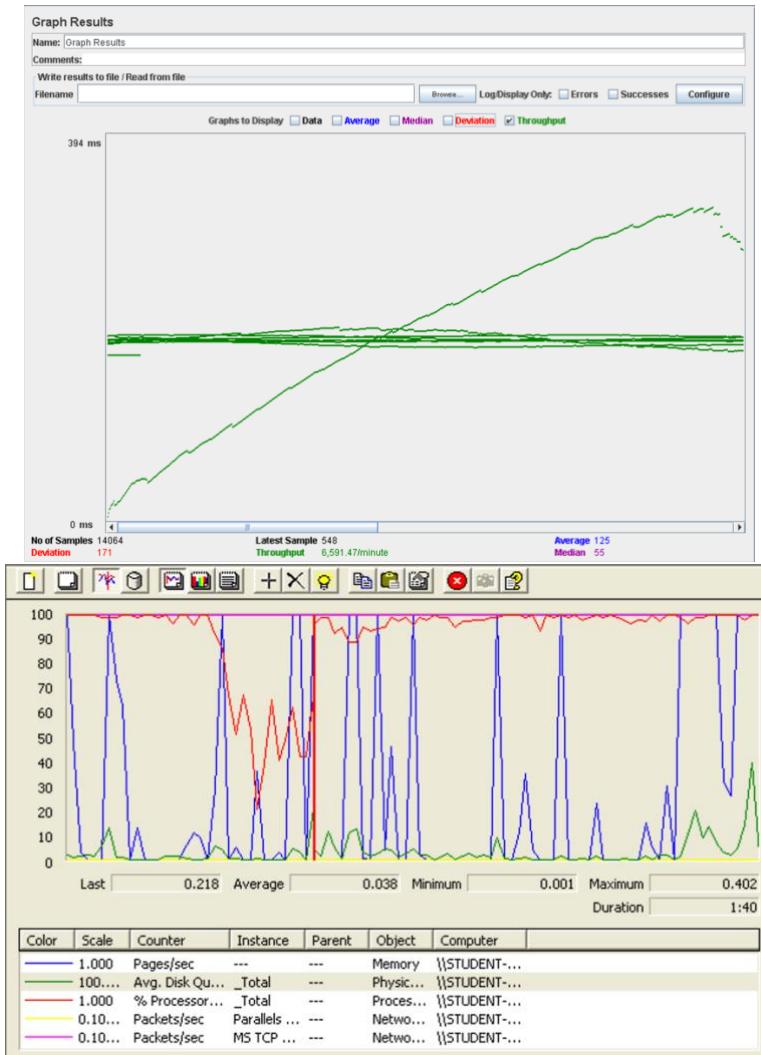
Name: Aggregate Report
Comments:
Write results to file / Read from file

Filename: [] Browse... LogDisplay Only: Errors: Successes: Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4693	289	234	535	28	3482	0.00%	6.3/sec	68.8
/styles/cookiec...	1604	32	26	66	2	449	0.00%	2.1/sec	8.9
/styles/circle.css	1603	30	22	60	1	257	0.00%	2.1/sec	1.0
/scripts/bootstrap...	1598	47	37	96	3	637	0.00%	2.1/sec	78.1
/scripts/bootstrap...	1588	83	72	151	6	457	0.00%	2.1/sec	252.5
/scripts/jquery....	1586	73	61	143	4	745	0.00%	2.1/sec	202.6
/scripts/style.css	1585	34	26	66	2	343	0.00%	2.1/sec	3.3
/favicon.ico	1585	34	24	66	2	792	0.00%	2.1/sec	11.9
/images/film_k_br...	3131	112	75	258	13	885	0.00%	4.2/sec	1366.3
/images/logo.jpg	1578	338	269	632	48	2799	0.00%	2.1/sec	2387.0
/scripts/cookiec...	1576	45	35	90	3	462	0.00%	2.1/sec	72.6
/scripts/helpers...	1565	27	19	60	2	186	0.00%	2.1/sec	4.8
/scripts/polyfills...	1564	12	8	28	2	130	0.00%	2.1/sec	15.8
/security/login.do	1558	95	41	228	11	1779	0.00%	2.1/sec	13.1
/j_spring_secu...	1555	357	272	711	30	2263	0.00%	2.1/sec	24.7
/raffledisplay.do	1549	156	78	379	17	1666	0.00%	2.1/sec	18.8
/ticketuserbuy...	3098	305	223	655	20	3031	0.00%	4.2/sec	114.7
/ticketuserlist...	1547	252	151	586	29	2072	0.00%	2.1/sec	201.5
/j_spring_secu...	1540	332	253	654	35	3718	0.00%	2.1/sec	23.0
TOTAL	36103	158	72	403	1	3718	0.00%	48.1/sec	4834.6

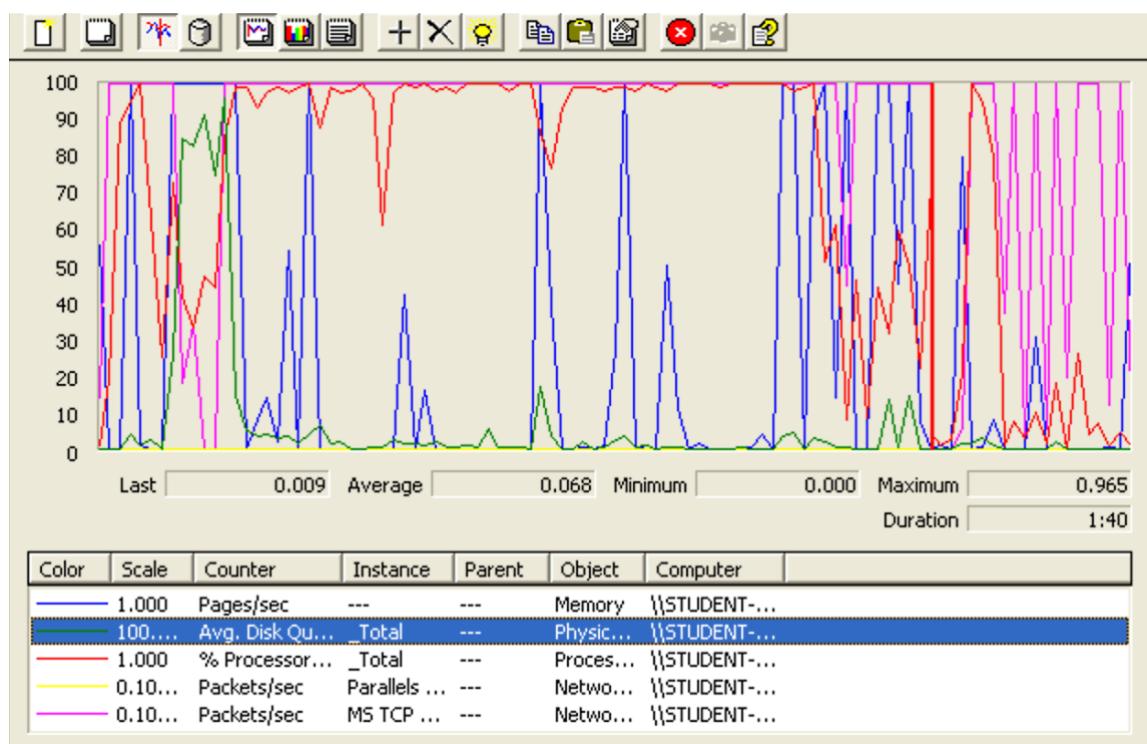
Caso de uso 014: Listar tiques

Procedemos a medir el rendimiento del caso de uso en el que el usuario lista los tiques que ha adquirido. Empezamos el test con 80 usuarios, y vemos resultados muy buenos dentro de los límites impuestos.

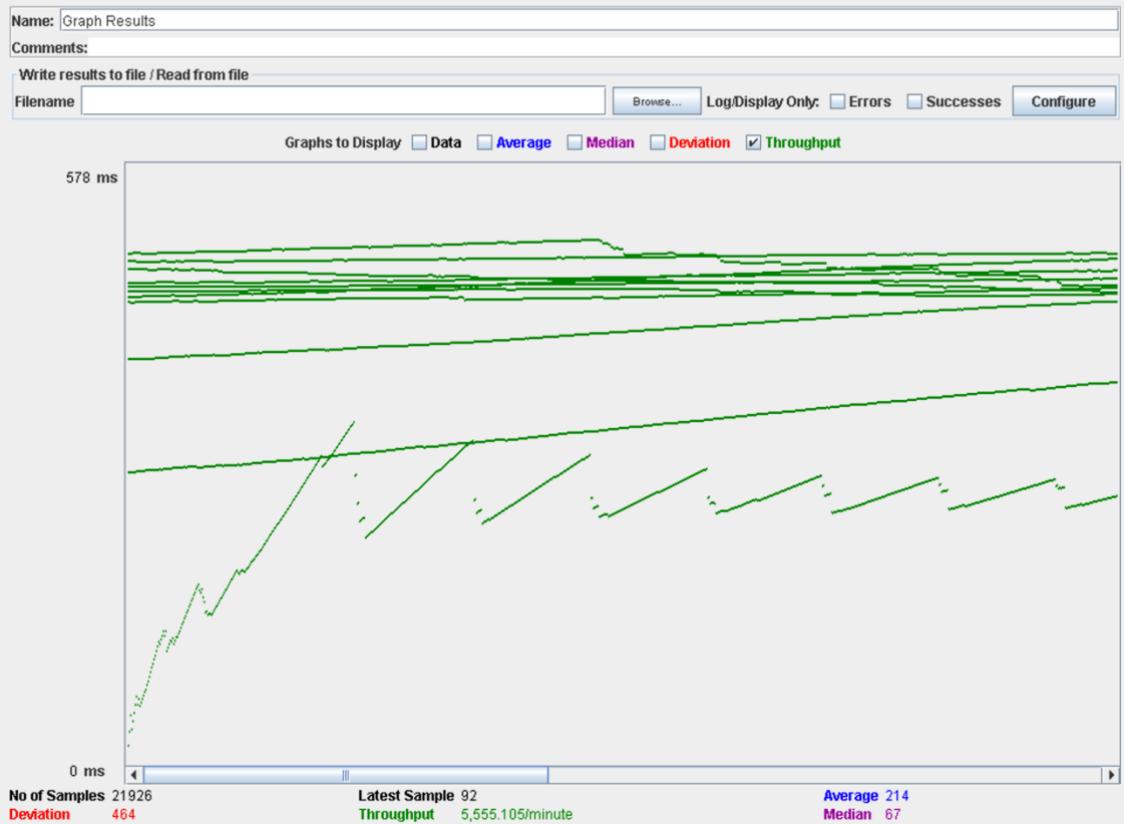


Aggregate Report										
<input type="text" value="Aggregate Report"/> Name: <input type="text" value="Comments:"/> <input type="checkbox"/> Write results to file / Read from file <input type="text" value="Filename"/> <input type="button" value="Browse..."/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	2259	261	193	541	28	1745	0.00%	17.1/sec	187.5	
/scripts/cookiec...	801	76	29	113	3	930	0.00%	6.1/sec	208.2	
/scripts/helpers...	792	30	21	65	2	354	0.00%	6.0/sec	13.8	
/scripts/polyfills...	776	28	20	58	2	280	0.00%	5.9/sec	44.7	
/scripts/bootstrap...	771	37	30	76	3	248	0.00%	5.9/sec	215.1	
/scripts/jquery...	763	53	41	107	4	257	0.00%	5.8/sec	556.0	
/styles/circle.css	758	31	22	62	2	682	0.00%	5.8/sec	2.7	
/styles/style.css	753	29	23	57	2	685	0.00%	5.8/sec	9.0	
/styles/cookiec...	745	32	21	68	2	685	0.00%	5.8/sec	23.8	
/styles/bootstrap...	742	70	50	136	5	922	0.00%	5.8/sec	685.9	
/favicon.ico	742	47	22	76	2	656	0.00%	5.7/sec	31.9	
/images/logo.jpg	739	210	149	378	29	1775	0.00%	5.7/sec	6418.8	
/images/link_br...	736	62	41	134	9	455	0.00%	5.7/sec	1851.3	
/security/login.do	730	109	72	244	10	648	0.00%	5.8/sec	36.1	
/j_spring_secu...	727	278	244	525	44	1192	0.00%	6.2/sec	72.7	
/ticketUserList...	727	208	141	444	22	1305	0.00%	6.0/sec	54.3	
/j_spring_secu...	726	295	226	608	33	1355	0.00%	6.2/sec	67.9	
TOTAL	14287	124	55	335	2	1775	0.00%	105.8/sec	10055.5	

Buscando el límite del sistema, aumentamos un poco más el número de usuarios hasta los 100. Los resultados arrojados en el aggregate report muestran unos tiempos por encima del segundo. Por ende, podemos concluir que nuestro sistema aguanta hasta 80 usuarios concurrentes para este caso de uso. El throughput es de 6591 y la grafica del estado del sistema durante el test nos muestra que de nuevo la tarjeta de red actúa como cuello de botella.



Graph Results



Aggregate Report

Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3404	394	281	831	27	3644	0.00%	14.4/sec	157.8
/scripts/cookiec...	1176	44	34	91	3	558	0.00%	5.1/sec	174.1
/scripts/helpers...	1175	36	23	76	2	1168	0.00%	5.1/sec	11.6
/scripts/polyfills...	1174	36	25	77	2	1025	0.00%	5.1/sec	38.7
/scripts/bootstrap...	1172	51	35	104	4	733	0.00%	5.2/sec	188.6
/scripts/jquery...	1171	64	47	123	4	1221	0.00%	5.2/sec	496.4
/styles/circle.css	1170	35	25	70	2	454	0.00%	5.2/sec	2.4
/styles/style.css	1167	40	27	77	2	708	0.00%	5.3/sec	8.2
/styles/cookiec...	1166	43	29	86	2	856	0.00%	5.3/sec	21.9
/styles/bootstrap...	1164	86	53	200	4	895	0.00%	5.3/sec	634.1
/favicon.ico	1163	48	33	101	2	851	0.00%	5.3/sec	29.9
/images/logo.jpg	1160	384	174	1030	28	3159	0.00%	5.3/sec	6007.5
/images/link_b...	1158	86	44	215	9	1002	0.00%	5.3/sec	1739.9
/security/login.d...	1154	501	150	718	11	6579	0.00%	5.4/sec	33.5
/j_spring_secu...	1145	632	425	1073	47	6956	0.00%	5.4/sec	63.2
/ticketUserList...	1112	373	232	826	23	3999	0.00%	5.4/sec	49.0
/j_spring_secu...	1095	478	362	929	42	3529	0.00%	5.4/sec	58.7
TOTAL	21926	214	67	515	2	6956	0.00%	92.6/sec	8966.4

Caso de uso 015: Listar y desplegar tarjetas de crédito

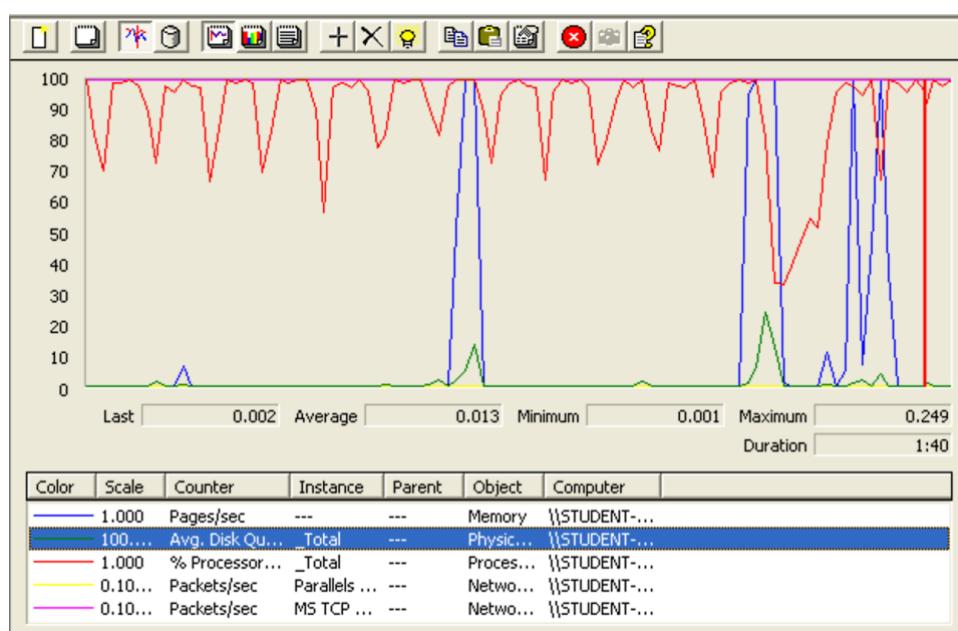
Procedemos a realizar el test de rendimiento del caso de uso de listar y desplegar tarjetas de crédito. Comenzamos con 90 usuarios, y vemos como los valores superan levemente el límite del segundo, por ende, vamos a bajar el número de usuarios.

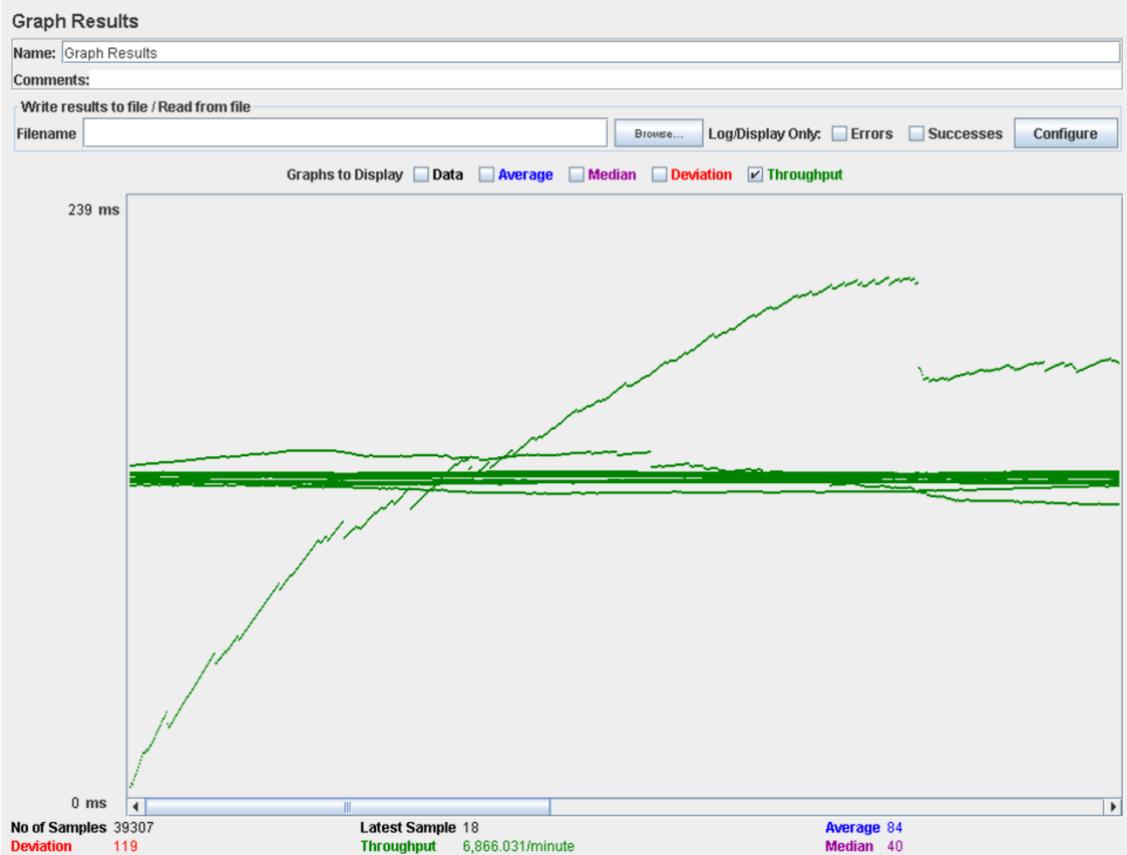


Aggregate Report										
Name: Aggregate Report Comments: <input type="checkbox"/> Write results to file / Read from file Filename <input type="text"/> Browse... Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="checkbox"/> Configure										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	3769	251	171	492	20	6948	0.00%	12.0/sec	131.7	
/scripts/cookiec...	1314	37	23	68	2	5386	0.00%	4.3/sec	146.2	
/scripts/helpers...	1314	37	16	51	2	5599	0.00%	4.3/sec	9.7	
/scripts/polyfills...	1314	40	15	51	2	5370	0.00%	4.3/sec	32.0	
/scripts/jquery...	1314	44	28	74	4	1535	0.00%	4.3/sec	405.1	
/scripts/bootstrap...	1314	33	21	58	3	1064	0.00%	4.3/sec	155.3	
/styles/circle.css	1314	24	14	43	2	5677	0.00%	4.3/sec	2.0	
/styles/bootstrap...	1314	43	33	84	5	1012	0.00%	4.3/sec	504.8	
/styles/cookiec...	1314	18	11	41	2	226	0.00%	4.3/sec	17.6	
/styles/style.css	1314	20	7	32	2	4263	0.00%	4.3/sec	6.6	
/security/login.do	1309	241	109	567	10	6303	0.00%	4.2/sec	26.3	
/favicon.ico	1305	64	18	82	2	5583	0.00%	4.2/sec	23.5	
/images/logo.jpg	1263	393	156	840	29	6223	0.00%	4.1/sec	4592.9	
/images/link_br...	1241	156	35	156	8	4906	0.00%	4.0/sec	1313.1	
/j_spring_secu...	1240	421	277	936	30	7014	0.00%	4.0/sec	47.2	
/creditcard/user...	1230	193	68	335	14	7072	0.00%	4.1/sec	32.3	
/creditcard/user...	1228	245	86	391	13	7155	0.00%	4.1/sec	30.4	
/j_spring_secu...	1225	310	205	561	21	7131	0.00%	4.1/sec	44.5	
TOTAL	25636	151	45	343	2	7155	0.00%	81.6/sec	7409.0	

Include group name in label? Save Table Data Save Table Header

Con 75 usuarios concurrentes, vemos unos tiempos aceptables para el 90% de los usuarios de nuestro sistema. Por ende, podemos concluir que, para este caso de uso, el sistema es capaz de soportar 75 usuarios concurrentes. El throughput es 6866 y vemos en la gráfica que arroja el perfmon que el procesador está mayormente en uso durante el transcurso del test.





Aggregate Report

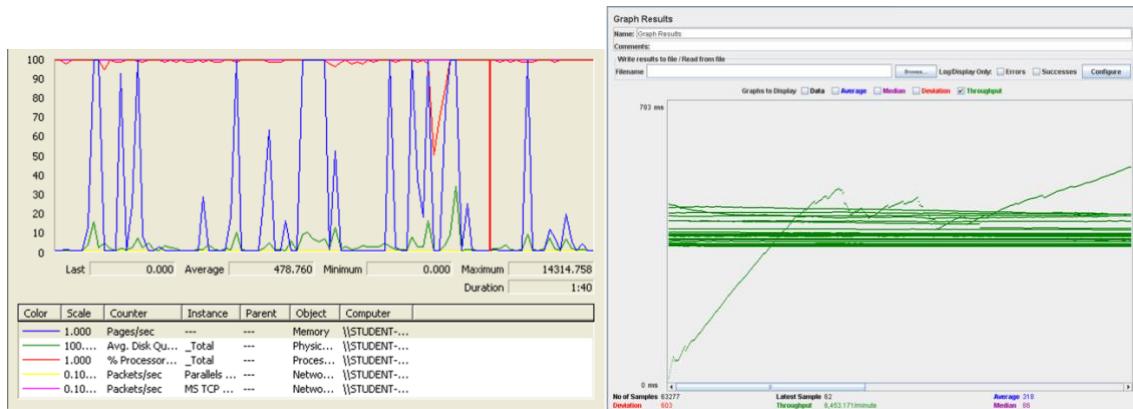
Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Throughput

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5653	155	117	307	24	1765	0.00%	17.1/sec	187.8
/scripts/cookiec...	1984	32	26	62	3	255	0.00%	5.9/sec	202.4
/scripts/helpers...	1984	24	18	50	2	282	0.00%	5.9/sec	13.5
/scripts/polyfills...	1984	22	16	43	2	229	0.00%	5.9/sec	44.4
/scripts/jquery....	1984	33	28	64	5	238	0.00%	5.9/sec	581.5
/scripts/bootstrap...	1984	26	20	51	3	259	0.00%	5.9/sec	215.3
/styles/circle.css	1984	18	14	38	2	130	0.00%	5.9/sec	2.8
/styles/bootstrap...	1984	30	24	58	5	287	0.00%	5.9/sec	699.1
/styles/cookiec...	1984	16	12	33	2	175	0.00%	5.9/sec	24.3
/styles/style.css	1984	14	6	34	2	211	0.00%	5.8/sec	9.1
/security/login.do	1983	103	69	219	12	1082	0.00%	5.9/sec	36.7
/favicon.ico	1961	32	18	76	2	474	0.00%	5.8/sec	32.5
/images/logo.jpg	1953	176	115	371	31	1638	0.00%	5.9/sec	6678.8
/images/link_br...	1953	56	34	124	9	690	0.00%	5.9/sec	1929.1
/j_spring_secu...	1953	254	208	473	48	1550	0.00%	5.9/sec	69.2
/creditcard/user...	1953	86	52	191	14	848	0.00%	5.9/sec	46.8
/creditcard/user...	1923	110	67	249	12	1269	0.00%	5.8/sec	43.5
/j_spring_secu...	1919	205	163	402	34	1169	0.00%	5.8/sec	63.4
TOTAL	39307	84	40	222	2	1765	0.00%	114.4/sec	10470.7

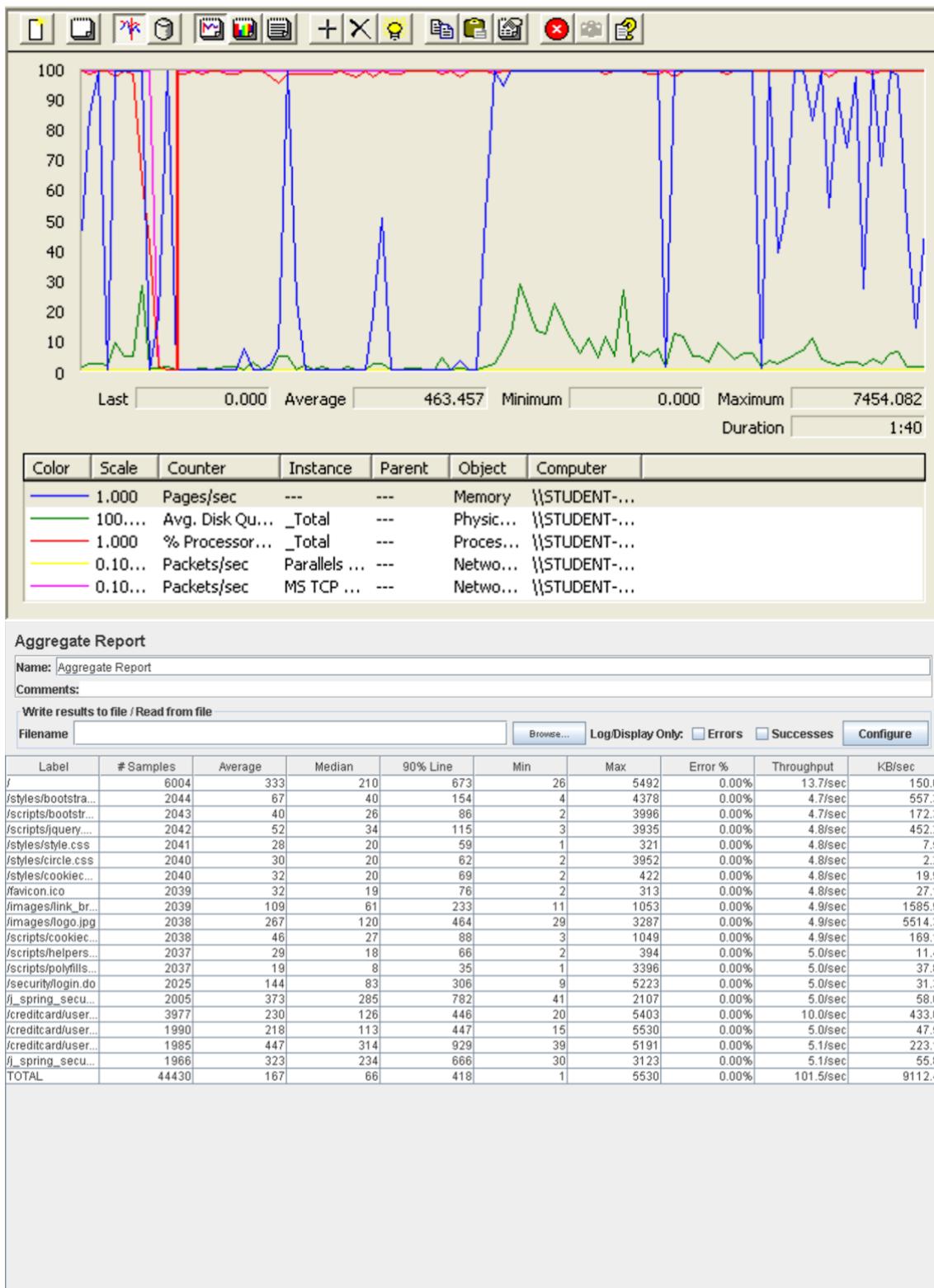
Caso de uso 016: Crear tarjeta de crédito

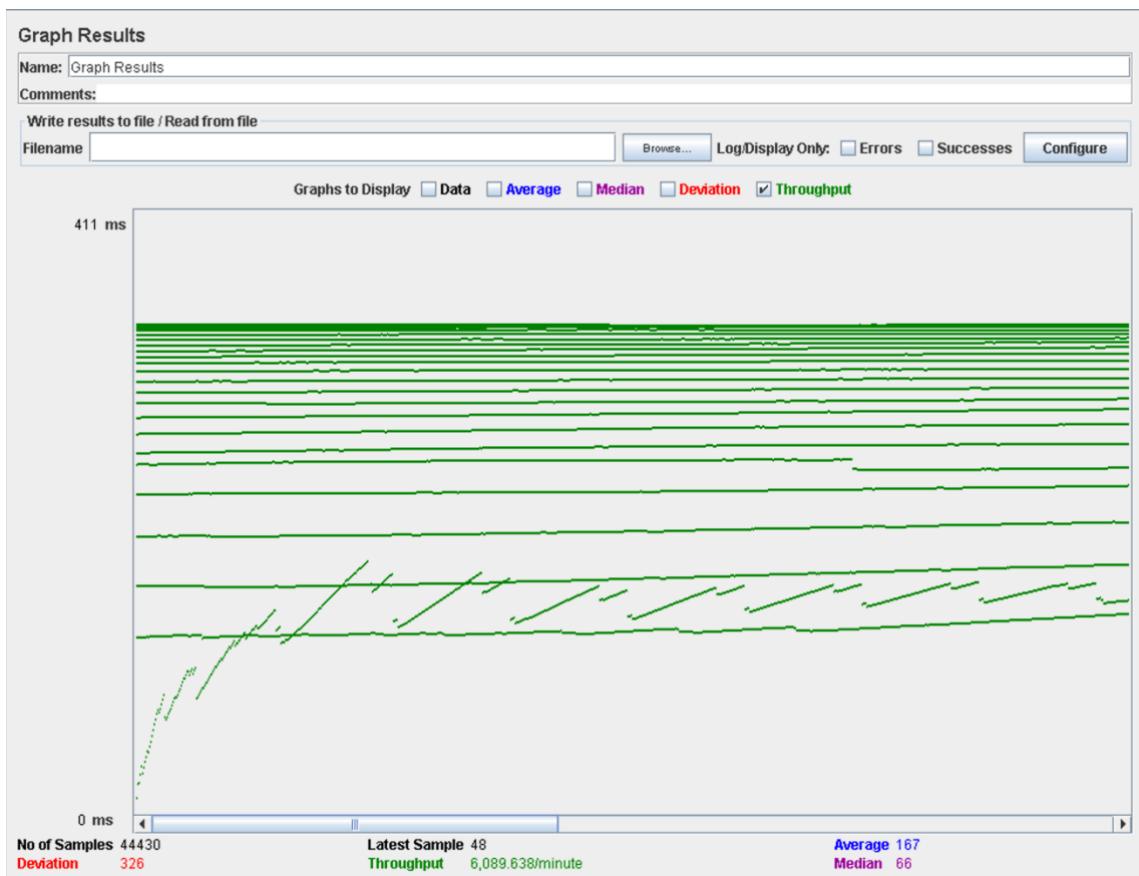
Procedemos a medir el rendimiento en el caso de uso que consiste en crear una tarjeta de crédito. Comenzamos con 115 usuarios, pero vemos como el aggregate report arroja tiempos para el 90% de los usuarios que superan el segundo que hemos marcado como límite para ser considerado aceptable.

Aggregate Report										
Name:		Aggregate Report								
Comments:										
Write results to file / Read from file		<input type="checkbox"/> Browse... <input type="checkbox"/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="checkbox"/> Configure								
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	63277	599	394	1234	23	9052	0.00%	14.8/sec	160.3	
/stylesheets/style.css	2914	62	35	118	4	2389	0.00%	4.0/sec	590.8	
/scripts/bootstrap...	2912	46	27	89	3	2235	0.00%	5.0/sec	181.6	
/scripts/jquery...	2907	61	34	111	4	2236	0.00%	5.0/sec	472.0	
/styles/shield.css	2905	37	21	70	2	2225	0.00%	5.0/sec	7.0	
/styles/circle.css	2902	41	22	74	1	2399	0.00%	5.0/sec	2.3	
/scripts/cookiec...	2898	40	22	76	2	2400	0.00%	5.0/sec	20.5	
/favicon.ico	2889	39	23	76	2	2121	0.00%	4.9/sec	27.6	
/images/link_b...	2887	93	59	190	9	2438	0.00%	4.9/sec	1609.6	
/images/logo.jpg	2885	179	114	356	24	3531	0.00%	4.9/sec	5574.9	
/scripts/cookiec...	2885	47	28	88	2	2156	0.00%	4.9/sec	169.5	
/scripts/helpers...	2884	38	22	73	2	2226	0.00%	4.9/sec	11.3	
/scripts/polyfills...	2884	30	12	51	2	2399	0.00%	4.9/sec	37.2	
/security/login.d...	2879	408	224	682	9	6681	0.00%	4.9/sec	30.9	
/j_spring_secu...	2876	936	673	1856	28	11670	0.00%	5.0/sec	58.1	
/creditarduser...	5685	535	320	1138	24	7979	0.00%	9.8/sec	1415.3	
/creditarduser...	2846	488	282	1065	13	10197	0.00%	4.9/sec	46.4	
/creditarduser...	2834	1024	748	1990	42	12951	0.00%	4.9/sec	712.7	
/j_spring_secu...	2815	609	406	1187	32	14437	0.00%	4.9/sec	53.7	
TOTAL	63277	318	86	880	1	14437	0.00%	107.6/sec	11076.6	



Bajamos hasta los 100 usuarios y ahora sí se cumple la condición que antes comentamos. Por ende, ya podemos concluir que el sistema para este caso de uso soporta hasta 100 usuarios y tiene un throughput de 6089. El perfmon arroja como el procesador y la tarjeta de red está al 100% de su capacidad limitando así nuestra capacidad de soportar más usuarios.

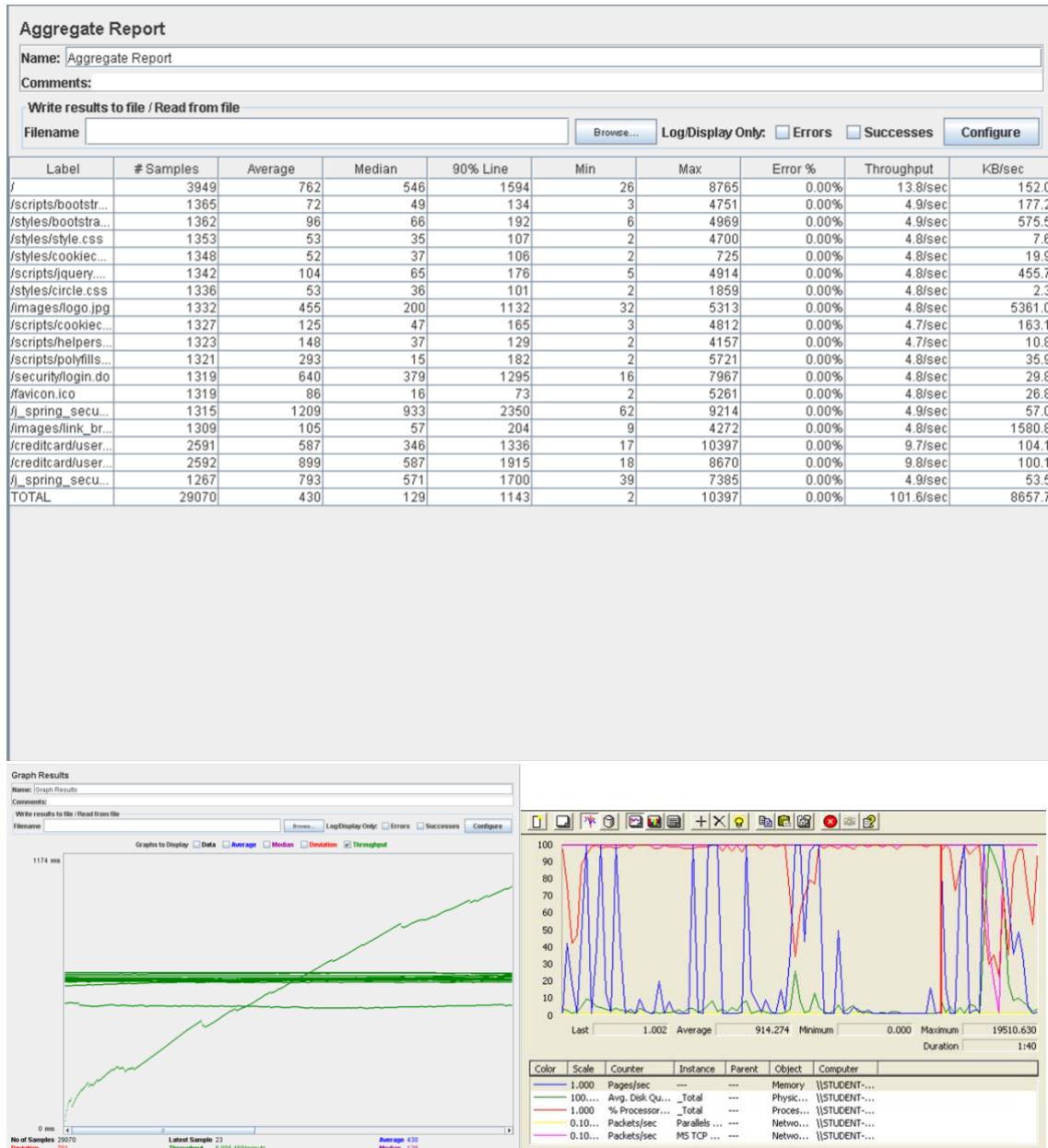




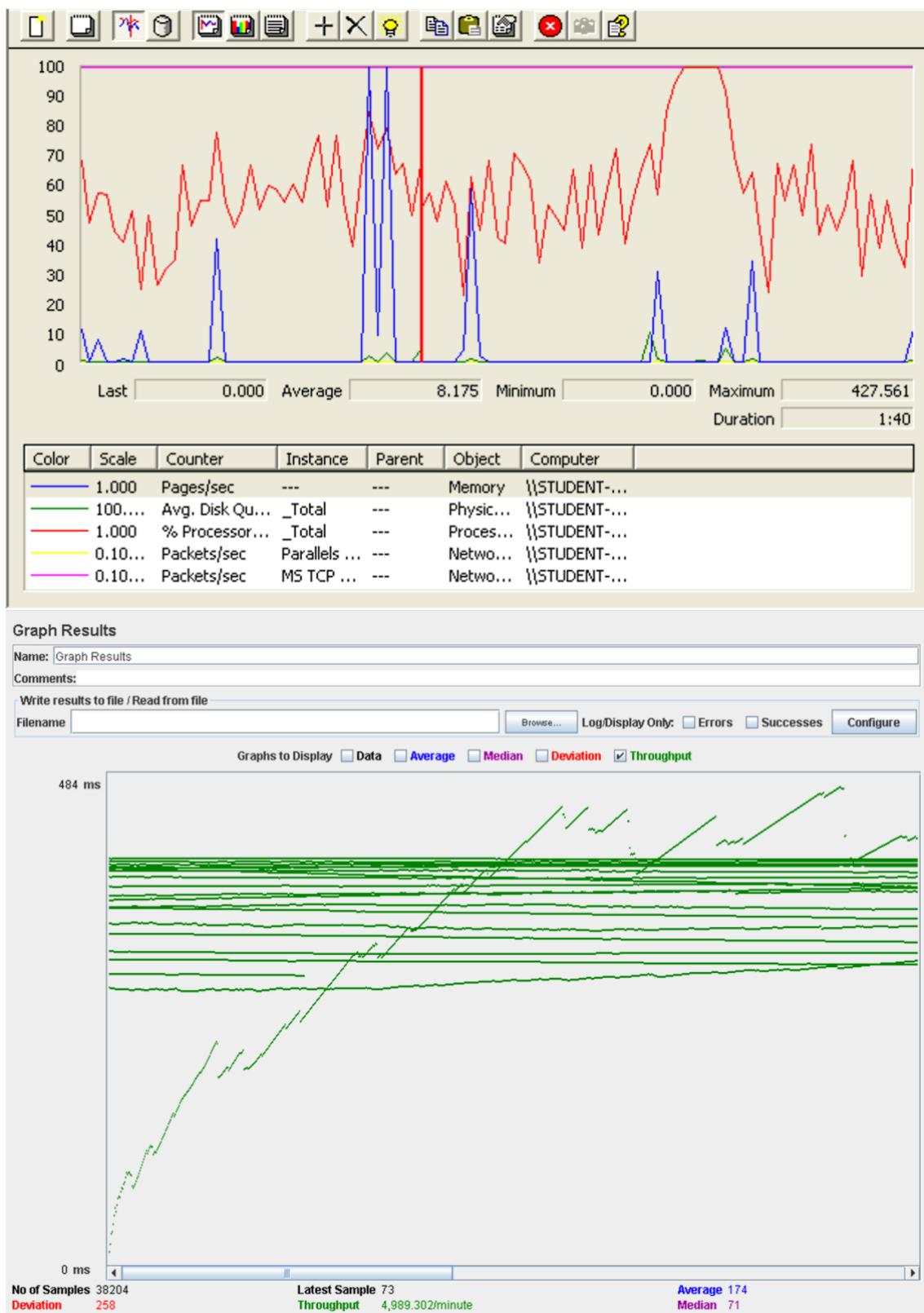
Caso de uso 017: Editar tarjeta de crédito

Procedemos a medir el rendimiento en el caso de uso de editar una tarjeta de crédito.

Comenzamos con 130 usuarios, pero nos damos cuenta de que los tiempos que se reflejan son demasiado altos.



Bajamos el número de usuarios hasta los 110 y ahora sí vemos unos tiempos dentro de los límites aceptables, es por ello que concluimos que el sistema para este caso aguanta hasta los 110 usuarios concurrentes. El throughput es 4989 y el perfmon arroja que es la tarjeta de red quien actua de cuello de botella y no el procesador.



Aggregate Report

Name: Aggregate Report

Comments:

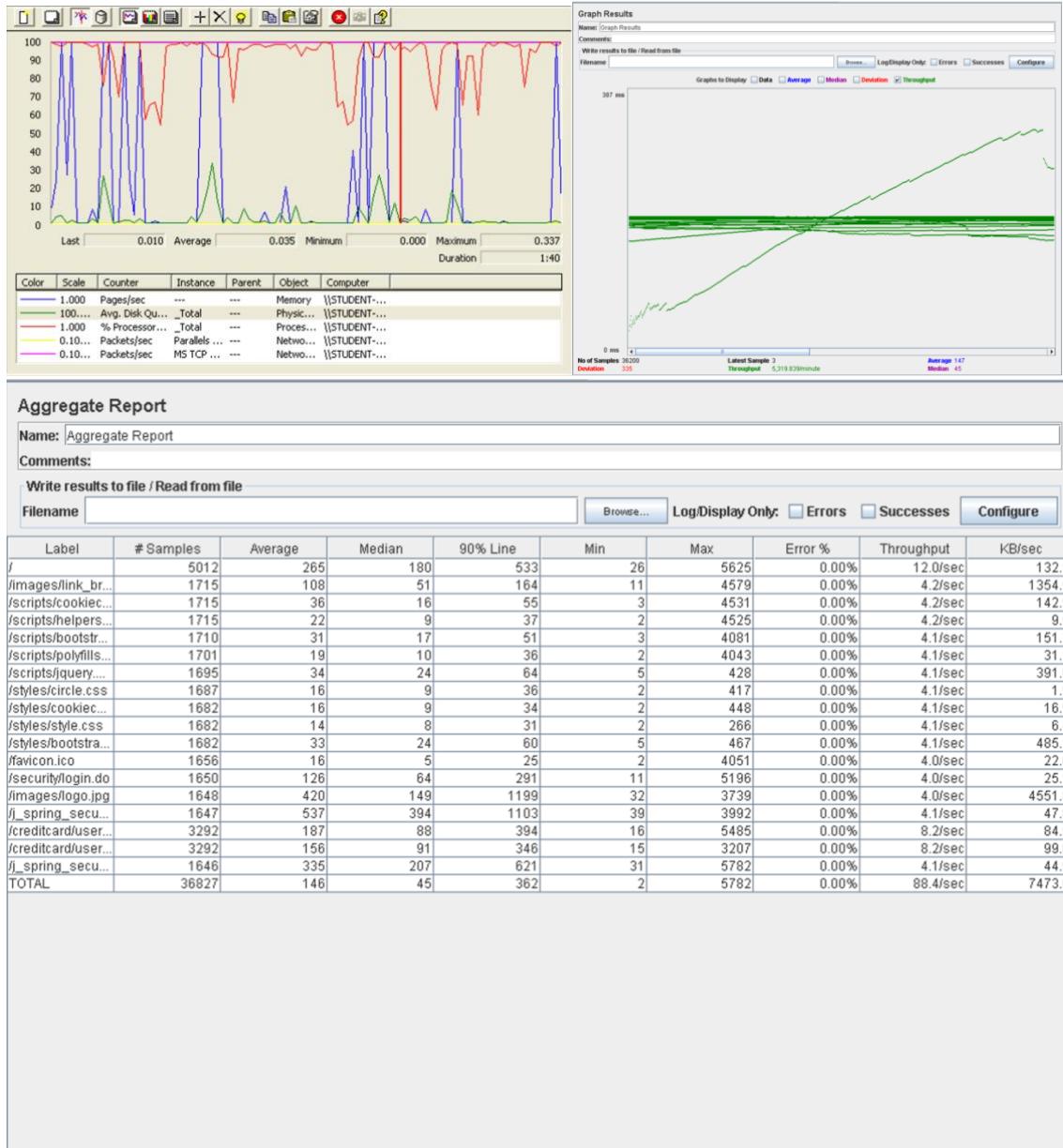
Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes

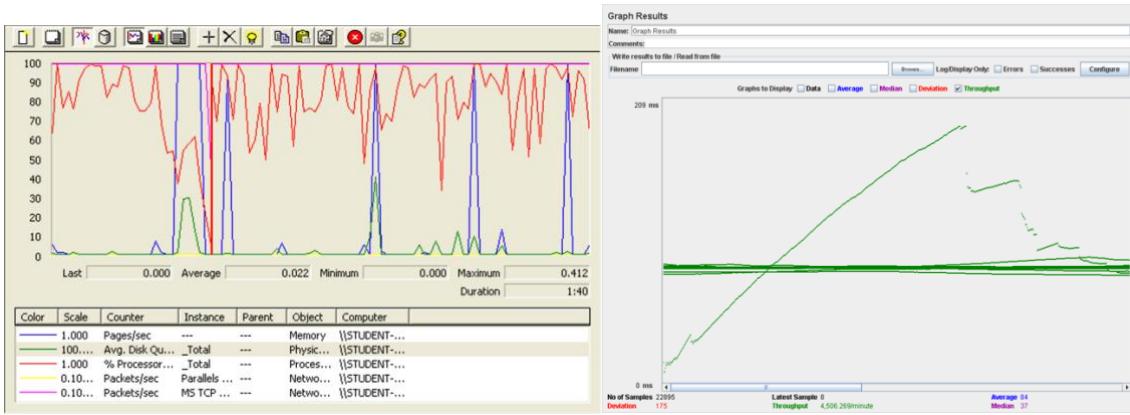
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5201	361	283	685	27	3449	0.00%	11.4/sec	124.7
/scripts/bootstrap...	1785	42	32	79	3	605	0.00%	3.9/sec	142.7
/styles/bootstrap...	1779	73	49	120	5	1253	0.00%	3.9/sec	462.8
/styles/style.css	1765	29	19	56	1	543	0.00%	3.9/sec	6.1
/styles/cookiec...	1754	30	18	64	2	1331	0.00%	3.9/sec	15.9
/scripts/jquery...	1749	60	43	114	4	1255	0.00%	3.8/sec	366.5
/styles/circle.css	1746	30	19	61	2	544	0.00%	3.8/sec	1.8
/images/logo.jpg	1744	210	150	388	30	1575	0.00%	3.8/sec	4332.8
/scripts/cookiec...	1740	41	28	76	3	694	0.00%	3.9/sec	132.4
/scripts/helpers...	1740	26	18	49	2	640	0.00%	3.8/sec	8.8
/scripts/polyfills...	1738	12	7	24	2	735	0.00%	3.8/sec	28.0
/security/login.do	1735	199	132	443	13	1925	0.00%	3.9/sec	24.1
/favicon.ico	1735	12	6	27	2	236	0.00%	3.9/sec	21.5
/j_spring_secu...	1733	446	354	873	49	2558	0.00%	3.9/sec	45.6
/images/link_br...	1732	61	37	112	9	1097	0.00%	3.9/sec	1271.2
/creditcard/user...	3420	218	144	474	17	3255	0.00%	7.8/sec	83.3
/creditcard/user...	3425	348	254	726	22	4729	0.00%	7.8/sec	78.7
/j_spring_secu...	1683	364	286	665	35	3497	0.00%	3.9/sec	42.3
TOTAL	38204	174	71	464	1	4729	0.00%	83.2/sec	7073.0

Caso de uso 17: Eliminar tarjeta de crédito

Procedemos a medir el rendimiento del sistema a la hora de borrar una tarjeta de crédito. Comenzamos con 75 usuarios realizando el caso de uso y vemos que los tiempos que arroja el 90% Line superan el segundo por lo que debemos bajar el número de usuarios.



Bajamos hasta los 50 usuarios y ya vemos unos tiempos mucho mejores. Por ende, podemos concluir que el sistema, para este caso de uso, soporta 50 usuarios concurrentes. El throughput será de 4506 y vemos que el procesador es el principal cuello de botella.



Aggregate Report

Name: Aggregate Report

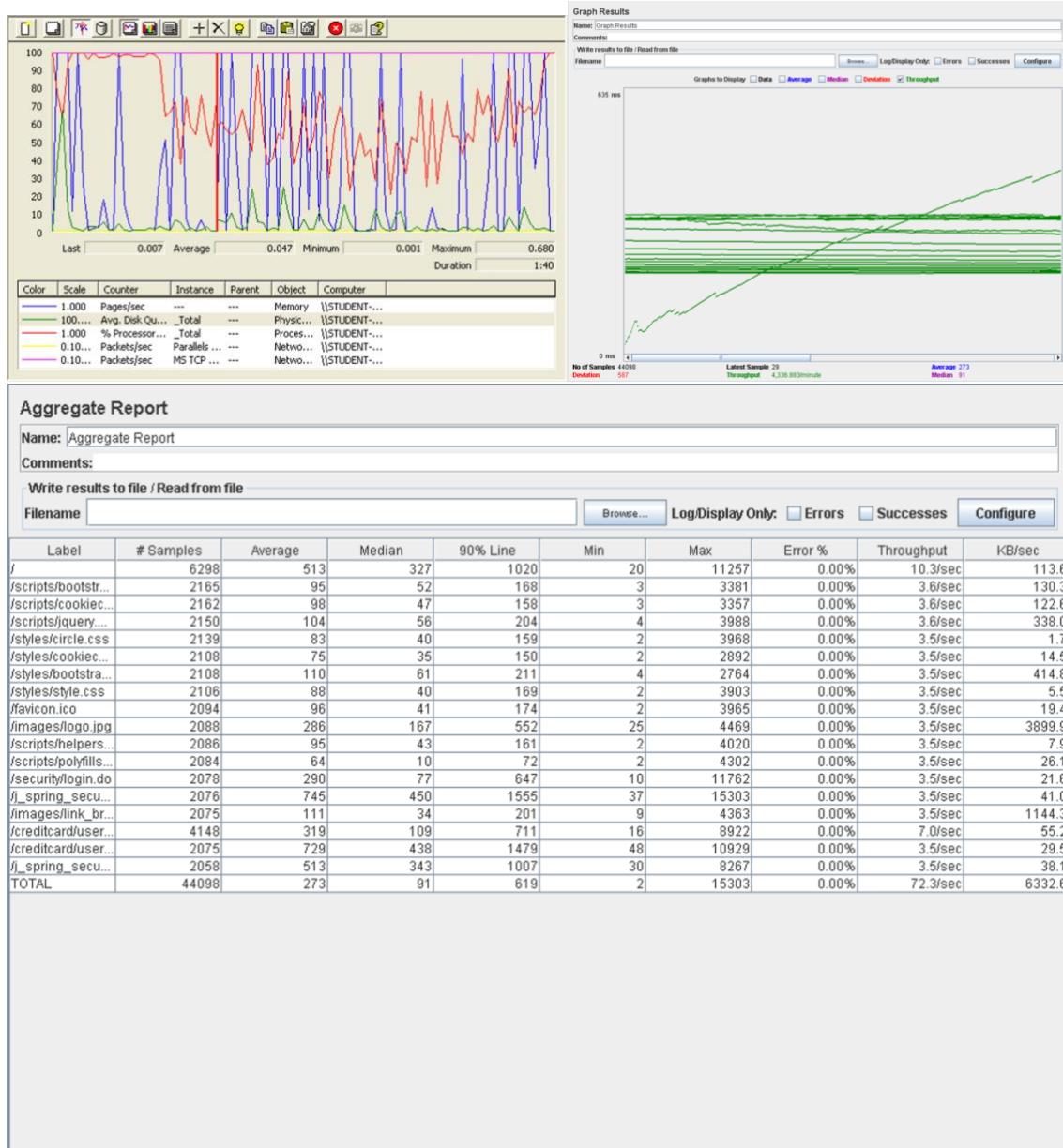
Comments:

Write results to file / Read from file

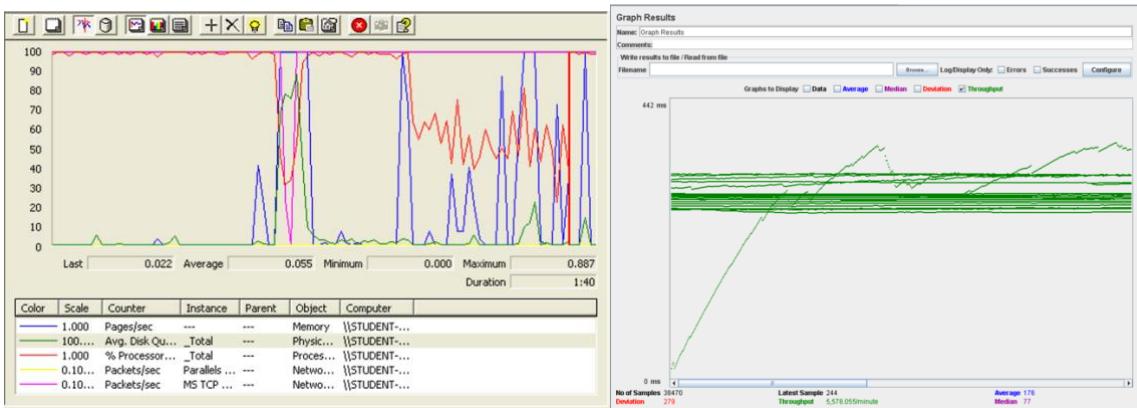
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3163	151	102	284	24	2939	0.00%	10.1/sec	111.0
/images/link_br...	1086	74	43	119	14	811	0.00%	3.5/sec	1149.2
/scripts/cookiec...	1086	25	16	52	3	268	0.00%	3.5/sec	121.1
/scripts/helpers...	1086	19	12	38	2	394	0.00%	3.5/sec	8.1
/scripts/bootstrap...	1086	22	16	42	3	809	0.00%	3.5/sec	128.7
/scripts/polyfills...	1086	18	12	37	2	419	0.00%	3.5/sec	26.6
/scripts/jquery....	1086	26	20	48	5	273	0.00%	3.5/sec	335.6
/styles/circle.css	1086	15	11	30	1	691	0.00%	3.5/sec	1.7
/styles/cookiec...	1086	17	11	32	2	649	0.00%	3.5/sec	14.6
/styles/style.css	1086	15	11	28	1	640	0.00%	3.5/sec	5.5
/styles/bootstrap...	1086	26	21	44	6	315	0.00%	3.5/sec	417.7
/favicon.ico	1086	10	5	19	2	521	0.00%	3.5/sec	19.7
/security/login.do	1086	64	37	120	10	2132	0.00%	3.5/sec	21.9
/images/logo.jpg	1045	248	99	563	35	3444	0.00%	3.4/sec	3834.9
/j_spring_secu...	1045	323	214	628	35	3413	0.00%	3.4/sec	39.9
/creditcard/user...	2073	93	55	193	16	2764	0.00%	6.8/sec	70.8
/creditcard/user...	2072	77	52	158	15	1351	0.00%	7.1/sec	85.1
/j_spring_secu...	1036	164	117	314	27	1817	0.00%	3.5/sec	38.3
TOTAL	23466	83	37	190	1	3444	0.00%	75.0/sec	6315.6

Caso de uso 018: Usar tarjeta de crédito como primaria

Procedemos a medir el rendimiento que ofrece el sistema para el caso de uso en el que marcamos una tarjeta de crédito por defecto para que así sea utilizada cuando vaya a comprar algo. Comenzamos con 140 usuarios, pero los tiempos se salen con creces del segundo que hemos marcado como límite.



Bajamos hasta los 85 usuarios donde ya vemos valores del 90% line, que es el tiempo que experimentan el 90% de los usuarios, muy buenos. Por ende, concluimos que el sistema aguanta 85 usuarios para este caso de uso. El throughput es 5578 y como vemos en la gráfica del perfmon, el procesador y la tarjeta de red se mantiene casi al 100% durante la mayor parte del test.



Aggregate Report

Name: Aggregate Report

Comments:

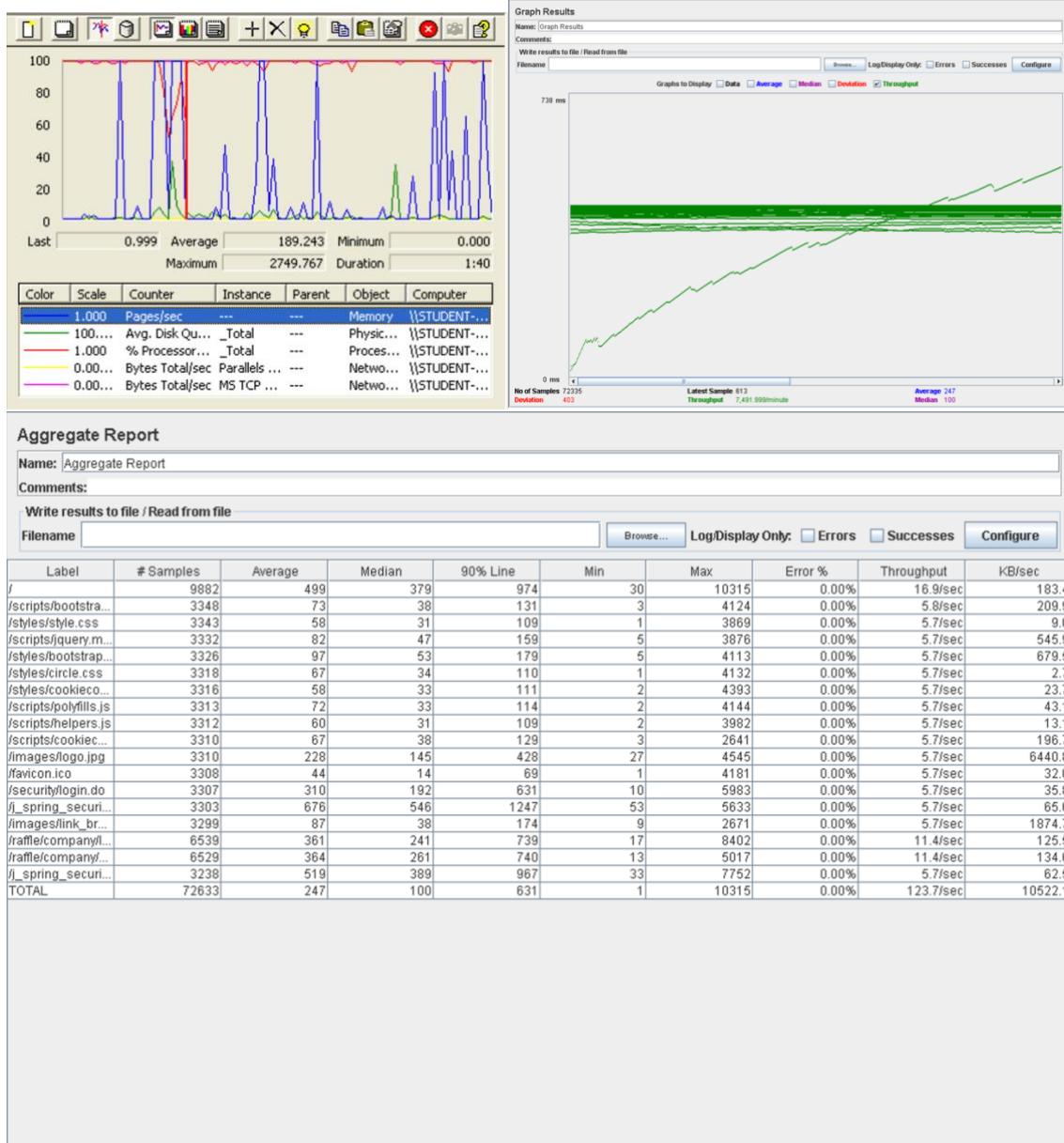
Write results to file / Read from file

Filename: LogDisplay Only Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5415	339	288	632	25	5107	0.00%	13.3/sec	145.6
/scripts/bootstrap...	1836	53	39	98	3	1951	0.00%	4.5/sec	164.9
/scripts/cookiec...	1835	52	32	91	4	3835	0.00%	4.5/sec	155.0
/scripts/jquery....	1834	69	48	115	4	3842	0.00%	4.5/sec	429.6
/styles/circle.css	1832	45	27	79	2	3814	0.00%	4.5/sec	2.1
/styles/cookiec...	1829	46	25	81	2	3886	0.00%	4.5/sec	18.6
/styles/bootstrap...	1827	78	56	145	6	3841	0.00%	4.5/sec	533.6
/styles/style.css	1821	42	26	80	2	3759	0.00%	4.5/sec	7.0
/favicon.ico	1821	40	27	85	2	579	0.00%	4.5/sec	25.1
/images/logo.jpg	1821	245	190	449	38	2710	0.00%	4.5/sec	5054.2
/scripts/helpers...	1817	39	27	85	2	695	0.00%	4.5/sec	10.2
/scripts/polyfills...	1814	23	12	46	2	2416	0.00%	4.5/sec	33.9
/security/login.do	1813	174	99	368	10	4032	0.00%	4.6/sec	28.6
/j_spring_secu...	1813	466	375	890	43	4539	0.00%	4.6/sec	53.5
/images/link_br...	1813	105	56	170	9	3784	0.00%	4.6/sec	1484.4
/creditcard/user...	3581	214	141	435	17	4482	0.00%	9.0/sec	71.1
/creditcard/user...	1801	427	361	783	45	2155	0.00%	4.5/sec	38.1
/j_spring_secu...	1770	353	288	677	33	4373	0.00%	4.5/sec	48.6
TOTAL	38093	175	76	459	2	5107	0.00%	93.3/sec	8227.3

Caso de uso 019: Eliminar rifa

Procedemos a realizar el test de rendimiento del caso de uso de crear una rifa. Comenzamos con 125 usuarios y vemos como se supera suavemente el límite del segundo, por lo que habrá que bajar la cantidad de usuarios.



Bajamos hasta los 100 usuarios y ahora sí los tiempos para el 90% de los usuarios es menor al segundo. Es por ello por lo que podemos concluir que el sistema para este caso de uso soporta un máximo de 100 usuarios. El throughput es 3386 y el cuello de botella se encuentra en la tarjeta de red.



Usuario crear evaluación.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a las compañías, luego al display de una de ellas y a crear evaluación.

Comenzamos probando con 60 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

Aggregate Report										
Name: CreateEvaluationUserTestAggregate		Comments:								
Write results to file / Read from file		Filename		Browse			Log Display Only			Configure
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	5400	351	263	682	20	7256	0.00%	10.4/sec	114.2	
/styles/cookieconsent.m	1800	13	9	23	2	418	0.00%	3.0/sec	14.7	
/scripts/bootstrap.min.js	1800	19	15	35	3	637	0.00%	3.0/sec	129.7	
/styles/style.css	1800	13	9	22	2	528	0.00%	3.0/sec	5.6	
/scripts/bootstrap.min.css	1800	37	20	72	5	417	0.00%	3.0/sec	421.8	
/scripts/circle.css	1800	13	9	23	2	531	0.00%	3.0/sec	1.7	
/scripts/fontawesome-all.js	1800	30	24	57	5	425	0.00%	3.0/sec	389.4	
/scripts/cookieconsent.js	1800	17	13	32	3	429	0.00%	3.0/sec	12.0	
/scripts/helpers.js	1800	12	9	23	2	512	0.00%	3.0/sec	8.1	
/scripts/poofilla.js	1800	12	9	22	2	391	0.00%	3.0/sec	26.8	
/images/logo.jpg	1800	183	169	314	30	893	0.00%	3.0/sec	4007.3	
/images/link_broken.png	1800	83	68	156	11	561	0.00%	3.0/sec	1159.6	
/security/login.do	1800	201	132	410	10	3242	0.00%	3.0/sec	22.2	
/_spring_security_check	1800	484	352	1011	7	6336	0.00%	3.0/sec	41.7	
/actor/companylist.do	1800	221	154	436	19	3243	0.00%	3.0/sec	35.5	
/actor/companyprofile.do	1800	199	135	406	15	5363	0.00%	3.0/sec	28.4	
/evaluationonuseredit.do	1800	277	189	436	18	4022	0.00%	3.0/sec	29.9	
/evaluationonuseredit.do	1800	568	435	1076	36	6066	0.00%	3.0/sec	171.7	
/evaluationonuseredit.do	1800	332	251	630	43	4415	0.00%	3.0/sec	1103.2	
/images/5stars.png	1800	14	11	29	2	266	0.00%	3.0/sec	38.7	
/images/yes.png	1800	13	10	28	2	98	0.00%	3.0/sec	5.2	
/_spring_security_logout	1800	396	296	762	24	3920	0.00%	3.0/sec	38.9	
TOTAL	43200	175	56	470	2	7256	0.00%	83.3/sec	8271.9	

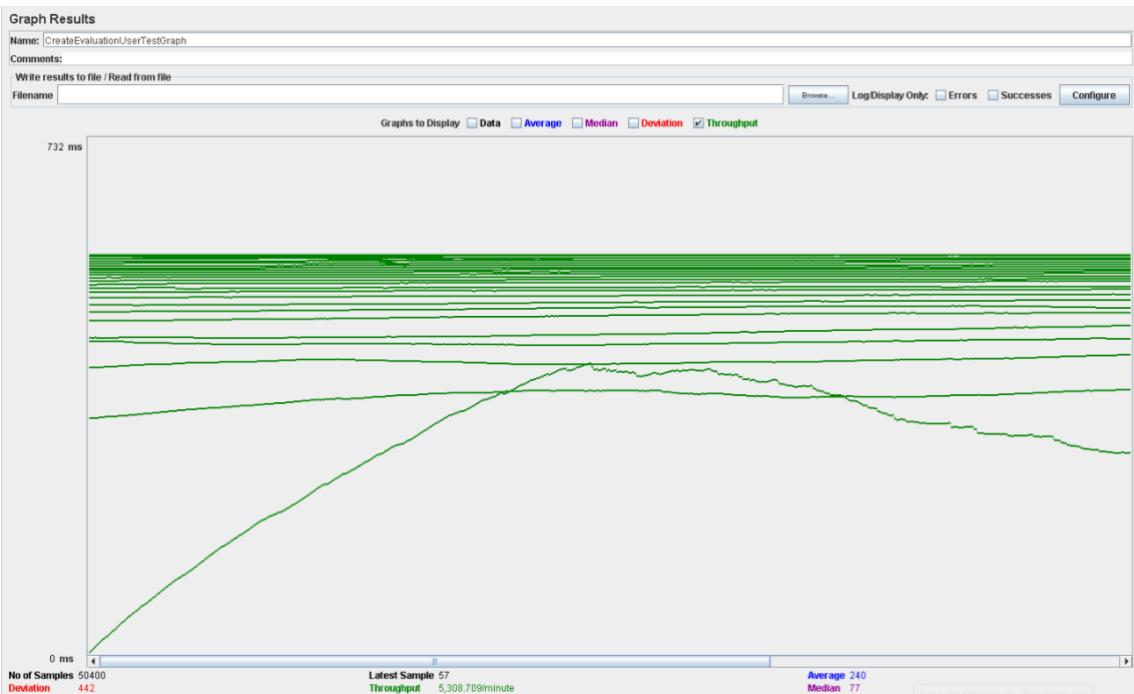


En esta tabla, el rendimiento es de aproximadamente 5.000 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 70 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: CreateEvaluationUserTestAggregate										
Comments:										
Write results to file / Read from file										
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	Configure
/	6300	524	394	905	17	7624	0.00%	11.1/sec	121.4	
/scripts/cookieconsent.m...	2100	19	13	33	2	494	0.00%	3.6/sec	15.8	
/scripts/bootstrap.min.js	2100	27	19	51	3	370	0.00%	3.6/sec	139.3	
/shescircle.css	2100	17	12	33	1	117	0.00%	3.6/sec	6.0	
/shescircle.map.min.css	2100	51	39	101	4	2194	0.00%	3.6/sec	453.7	
/scripts/jquery.min.js	2100	19	12	33	2	1959	0.00%	3.6/sec	1.8	
/scripts/cookieconsent.js	2100	47	32	85	5	2318	0.00%	3.6/sec	364.5	
/scripts/polyfills.js	2100	25	17	47	2	1962	0.00%	3.6/sec	131.5	
/scripts/helpers.js	2100	19	12	32	2	2082	0.00%	3.6/sec	8.7	
/images/logo.jpg	2100	22	12	32	2	2585	0.00%	3.6/sec	28.9	
/images/link_broken.png	2100	278	215	493	29	3292	0.00%	3.6/sec	422.4	
/images/link_broken.png	2100	111	88	225	10	685	0.00%	3.6/sec	126.3	
/security/login	247	186	487	10	4056	0.00%	3.6/sec			
/scripts/company/checkX	2100	647	457	1707	34	446	0.00%	3.6/sec	45.0	
/actor/company/list.do	2100	320	274	691	16	12297	0.00%	3.6/sec	30.4	
/actor/company/profile.do	2100	279	226	636	16	10754	0.00%	3.6/sec	30.8	
/evaluation/user/create.do	2100	408	338	726	16	13084	0.00%	3.6/sec	32.3	
/evaluation/user/edit.do	2100	716	671	1239	47	7921	0.00%	3.6/sec	671.0	
/evaluation/user/list	2100	427	320	766	36	14401	0.00%	3.6/sec	1029.1	
/images/5stars.png	2100	18	14	35	2	230	0.00%	3.6/sec	42.1	
/images/yes.png	2100	16	12	33	2	239	0.00%	3.6/sec	5.6	
/j_spring_security_logout	2100	484	390	920	21	10183	0.00%	3.6/sec	42.3	
TOTAL	50400	240	77	634	1	14401	0.00%	88.6/sec	8457.5	

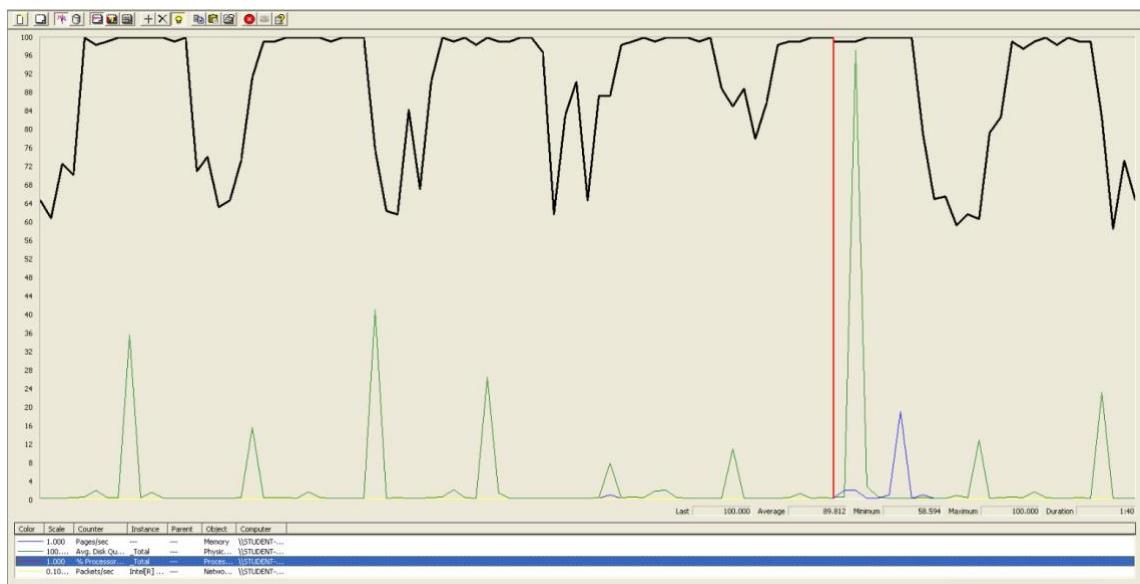
Como podemos observar fijándonos en 90% Line, la dirección /evaluation/user/edit.do genera un tiempo de respuesta (1239 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.300 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



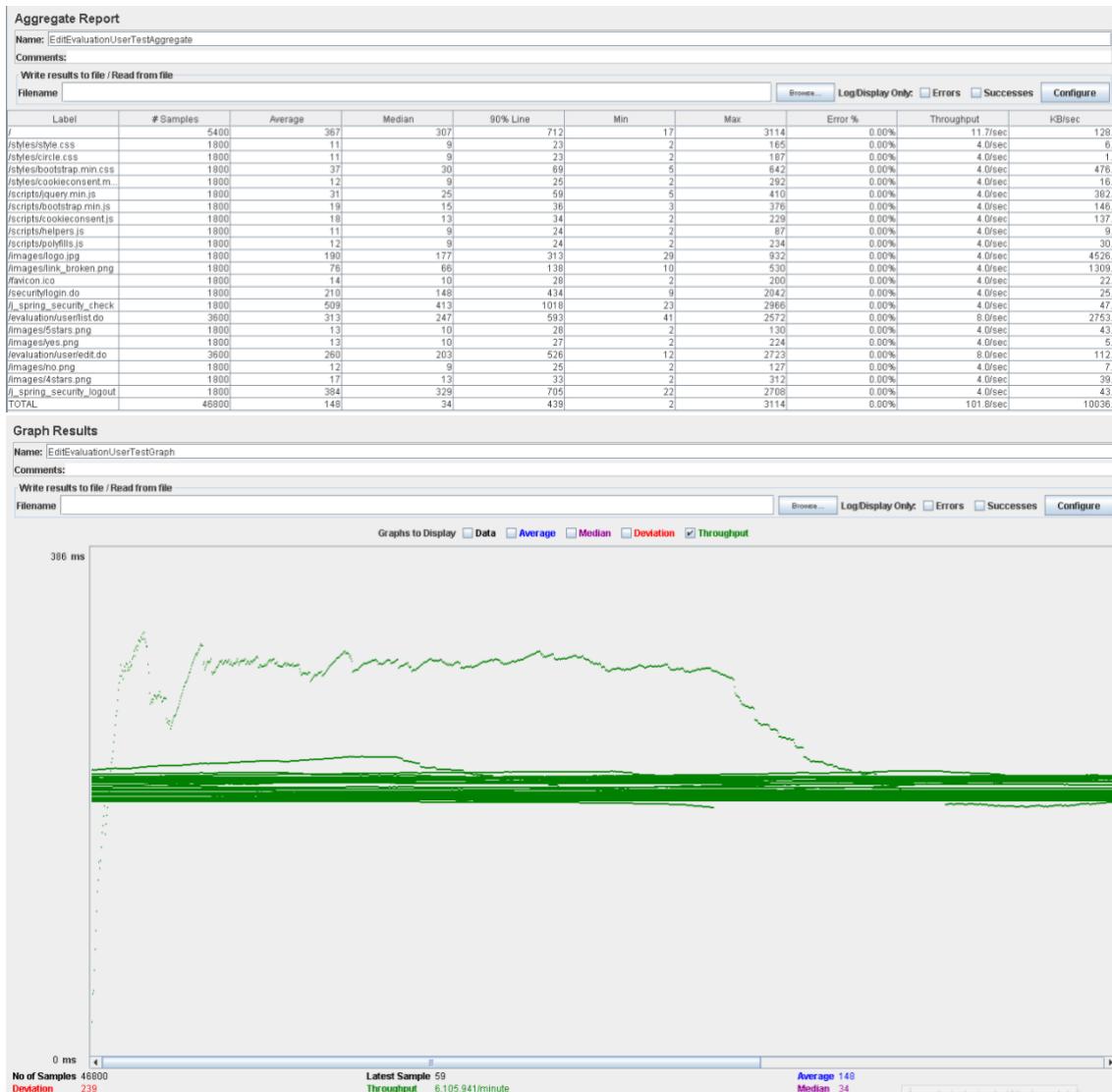
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 60 y 70.

Usuario editar evaluación.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a nuestras evaluaciones y pulsamos en editar alguna de ellas.

Comenzamos probando con 60 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:



En esta tabla, el rendimiento es de aproximadamente 6.100 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 70 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
<input type="button" value="Browse"/> <input type="button" value="Log"/> <input type="button" value="Display Only"/> <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	6300	527	413	1019	19	5874	0.00%	12.2/sec	133.4	
/styles/style.css	2100	12	10	22	2	269	0.00%	4.1/sec	6.5	
/styles/circle.css	2100	12	10	21	2	364	0.00%	4.1/sec	2.0	
/scripts/bootstrap.min.css	2100	41	34	74	6	889	0.00%	4.1/sec	492.2	
/scripts/bootstrap.min.js	2100	12	10	23	2	702	0.00%	4.1/sec	1.7	
/scripts/jquery.min.js	2100	24	20	61	4	441	0.00%	4.1/sec	395.0	
/scripts/bootstrap.min.js	2100	21	16	37	3	723	0.00%	4.2/sec	151.4	
/scripts/cookieconsent.js	2100	18	15	33	3	224	0.00%	4.2/sec	142.5	
/scripts/helpers.js	2100	13	10	23	2	131	0.00%	4.2/sec	9.5	
/scripts/polyfill.js	2100	13	10	23	2	427	0.00%	4.2/sec	31.3	
/images/logo.jpg	2100	206	194	329	30	742	0.00%	4.2/sec	4860.7	
/images/link_broken.png	2100	84	74	152	10	778	0.00%	4.2/sec	1354.1	
/favicon.ico	2100	14	11	29	2	225	0.00%	4.2/sec	23.2	
/securuth/login.co	2100	354	219	819	9	4332	0.00%	4.2/sec	25.9	
/_spring_security_check	2100	807	671	1662	32	537	0.00%	4.2/sec	48.7	
/securuth/logoutedit.do	4200	456	340	952	44	5310	0.00%	0.8/sec	285.0	
/images/5stars.png	2100	15	12	28	2	135	0.00%	4.2/sec	45.2	
/images/4stars.png	2100	14	11	29	2	191	0.00%	4.2/sec	6.0	
/evaluationonuseredit.do	4200	395	278	809	14	4795	0.00%	8.3/sec	116.7	
/images/no.png	2100	12	10	24	2	118	0.00%	4.2/sec	7.2	
/images/4stars.png	2100	17	14	35	2	433	0.00%	4.2/sec	41.1	
/j_spring_security_logout	2100	561	426	1092	25	6475	0.00%	4.2/sec	45.2	
TOTAL	54600	214	39	585	2	6475	0.00%	105.4/sec	10391.6	

Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check genera un tiempo de respuesta (1662 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 6.300 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 60 y 70.

Usuario borrar evaluación.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a nuestras evaluaciones y pulsamos en editar alguna, finalmente en borrar.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

Aggregate Report										
Name: DeleteEvaluationUserTestAggregate										
Comments:										
Write results to file / Read from file										Configure
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	4500	393	325	745	16	4721	0.00%	10.3sec	113.2	
/styles/bootstrap.min.css	1500	39	32	75	5	301	0.00%	3.5sec	419.5	
/styles/cookieconsent.m	1500	12	10	24	2	128	0.00%	3.5sec	14.6	
/styles/circle.css	1500	13	10	25	2	239	0.00%	3.5sec	1.7	
/styles/style.css	1500	13	10	26	1	417	0.00%	3.5sec	5.5	
/scripts/bootstrap.min.js	1500	21	17	41	3	162	0.00%	3.5sec	129.7	
/scripts/helpers.js	1500	13	10	26	1	125	0.00%	3.5sec	8.0	
/scripts/jquery.min.js	1500	34	27	63	5	273	0.00%	3.5sec	335.7	
/scripts/jquery.js	1500	18	15	34	3	177	0.00%	3.5sec	121.1	
/scripts/bootstrap.js	1500	13	10	27	2	199	0.00%	3.5sec	26.0	
/images/film_broken.png	1500	67	72	167	11	599	0.00%	3.5sec	114.4	
/images/logo.jpg	1500	221	197	378	31	1168	0.00%	3.5sec	3975.5	
/securelogin.do	1500	234	166	447	9	3592	0.00%	3.5sec	22.0	
/spring_security_check	1500	537	444	1006	30	4489	0.00%	3.5sec	41.4	
/evaluation/user/list.do	3000	356	294	658	45	2547	0.00%	7.0sec	2405.0	
/images/4stars.png	1500	16	12	32	2	445	0.00%	3.5sec	34.9	
/images/no.png	1500	13	10	26	2	156	0.00%	3.5sec	6.1	
/images/5stars.png	1500	16	12	31	2	149	0.00%	3.5sec	38.3	
/images/yes.png	1500	13	9	28	2	120	0.00%	3.5sec	51	
/evaluation/user/edit.do	3000	298	237	568	15	4270	0.00%	7.0sec	846.0	
/spring_security_logout	1500	380	320	740	20	4343	0.00%	3.5sec	39.4	
TOTAL	37500	168	43	480	1	4721	0.00%	86.0sec	8787.8	



En esta tabla, el rendimiento es de aproximadamente 5.100 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 60 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report											
<input type="button" value="Browse..."/> <input type="checkbox"/> LogDisplay Only <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="checkbox"/> Configure											
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec		
/	5400	661	551	1272	23	9782	0.00%	10.4/sec	113.8		
/styles/bootstrap.min.css	1800	42	33	80	7	414	0.00%	3.5/sec	419.2		
/styles/cookieconsent.m...	1800	16	13	33	2	276	0.00%	3.5/sec	14.6		
/styles/circle.css	1800	16	13	33	2	399	0.00%	3.5/sec	1.7		
/scripts/bootstrap.css	1800	24	19	48	3	284	0.00%	3.5/sec	5.5		
/scripts/bootstrap.min.js	1800	17	13	34	2	161	0.00%	3.5/sec	199.9		
/scripts/helpers.js	1800	38	30	74	6	207	0.00%	3.5/sec	81		
/scripts/jquery.min.js	1800	23	18	45	3	819	0.00%	3.5/sec	121.3		
/scripts/cookieconsent.js	1800	16	13	32	2	148	0.00%	3.5/sec	26.6		
/images/link_broken.png	90	74	168	13	407	0.00%	3.5/sec	1151.6			
/images/logo.jpg	1800	236	220	397	38	819	0.00%	3.5/sec	3978.1		
/securelogin.do	1800	498	332	1076	10	6956	0.00%	3.5/sec	22.0		
/j_spring_security_check	1800	1126	966	2061	38	6178	0.00%	3.5/sec	41.4		
/evaluation/list.do	3000	592	471	1165	57	4824	0.00%	7.0/sec	2413.2		
/images/start.png	1800	19	15	38	2	243	0.00%	3.5/sec	38.0		
/images/logo.png	1800	16	13	33	2	107	0.00%	3.5/sec	6.1		
/images/fstar.png	1800	19	16	40	2	220	0.00%	3.5/sec	38.4		
/images/love.png	1800	18	14	37	2	32	0.00%	3.5/sec	5.1		
/evaluation/user/edit.do	3600	526	404	1096	16	5438	0.00%	7.1/sec	94.7		
/j_spring_security_logout	1800	688	547	1370	23	5427	0.00%	3.5/sec	38.5		
TOTAL	45000	286	52	826	2	9782	0.00%	88.4/sec	8940.7		

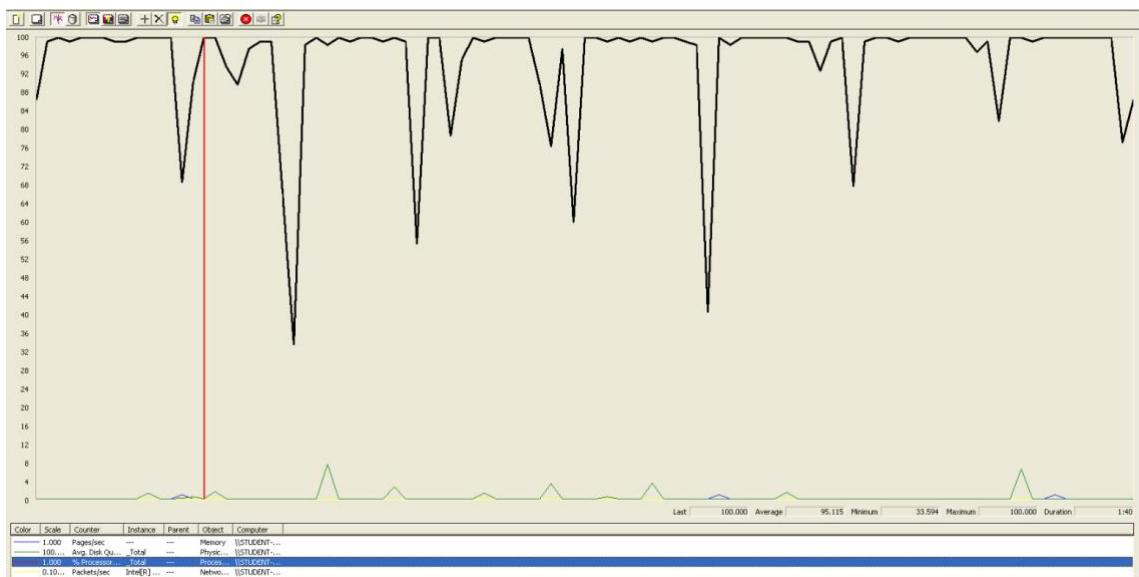
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check y /evaluation/user/list.do generan un tiempo de respuesta (2061 y 1165 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.200 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



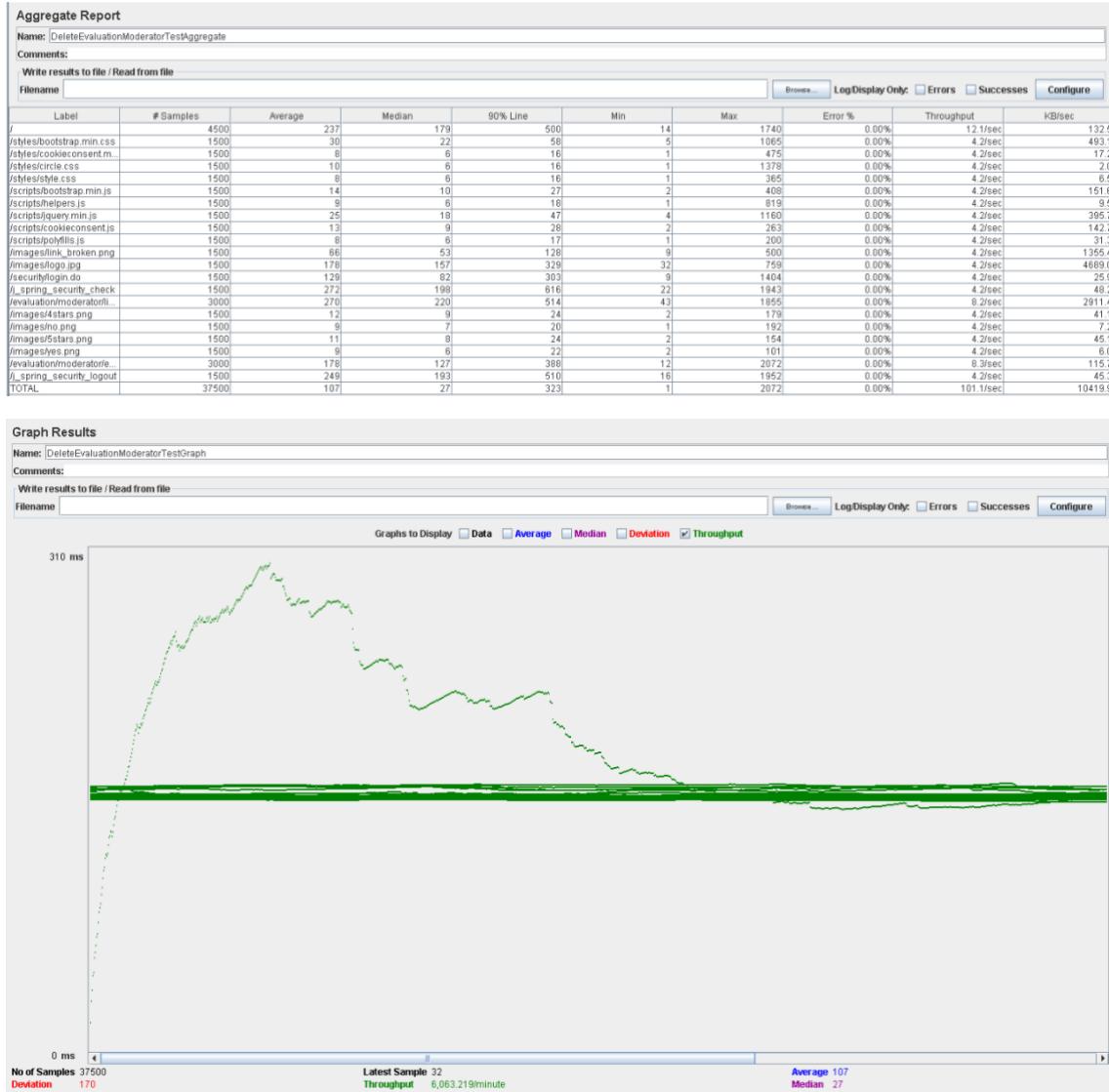
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 60.

Moderador borrar evaluación.

En este caso de uso, accedemos a la página, nos logueamos como moderador, accedemos a las evaluaciones y pulsamos en editar alguna de ellas, finalmente pulsamos en borrar.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:



En esta tabla, el rendimiento es de aproximadamente 6.000 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 60 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: DeleteEvaluationModeratorTestAggregate										
Comments:										
Write results to file / Read from file										
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	Configure
/	5400	534	431	1020	18	4993	0.00%	11.1/sec	121.3	
/styles/bootstrap.min.css	1800	43	36	78	5	391	0.00%	3.8/sec	448.4	
/styles/cookieconsent.m...	1800	13	10	23	2	201	0.00%	3.8/sec	15.6	
/styles/circle.css	1800	13	10	23	2	683	0.00%	3.8/sec	1.8	
/styles/style.css	1800	13	10	24	2	321	0.00%	3.8/sec	5.9	
/scripts/bootstrap.min.js	1800	22	19	41	3	358	0.00%	3.8/sec	11.0	
/scripts/helpers.js	1800	13	10	26	2	426	0.00%	3.8/sec	0.6	
/scripts/cookieconsent.js	1800	38	30	71	6	414	0.00%	3.8/sec	359.0	
/scripts/cookieconsent.js	1800	19	15	37	3	261	0.00%	3.8/sec	129.8	
/scripts/polyfill.js	1800	13	10	24	2	167	0.00%	3.8/sec	28.5	
/images/links_broken.png	1800	100	85	189	13	663	0.00%	3.8/sec	1232.4	
/images/logo.jpg	1800	231	211	381	35	1023	0.00%	3.8/sec	4262.6	
/securelogin.do	1800	386	270	796	11	3996	0.00%	3.8/sec	23.6	
/j_spring_security_check	1800	896	730	1760	27	5059	0.00%	3.8/sec	43.9	
/evaluation/moderator...	3600	548	436	1011	48	4409	0.00%	7.5/sec	2067.9	
/evaluation/moderator...	1800	17	14	32	2	77	0.00%	3.8/sec	1.5	
/images/logo.png	1800	13	11	25	2	116	0.00%	3.8/sec	0.6	
/images/start.png	1800	16	13	30	2	157	0.00%	3.8/sec	41.1	
/images/error.png	1800	14	11	30	2	132	0.00%	3.8/sec	5.5	
/evaluation/moderator...	3600	430	318	887	14	8101	0.00%	7.5/sec	105.4	
/j_spring_security_logout	1800	597	460	1183	20	4948	0.00%	3.8/sec	41.2	
TOTAL	45000	241	49	658	2	6101	0.00%	92.5/sec	9540.6	

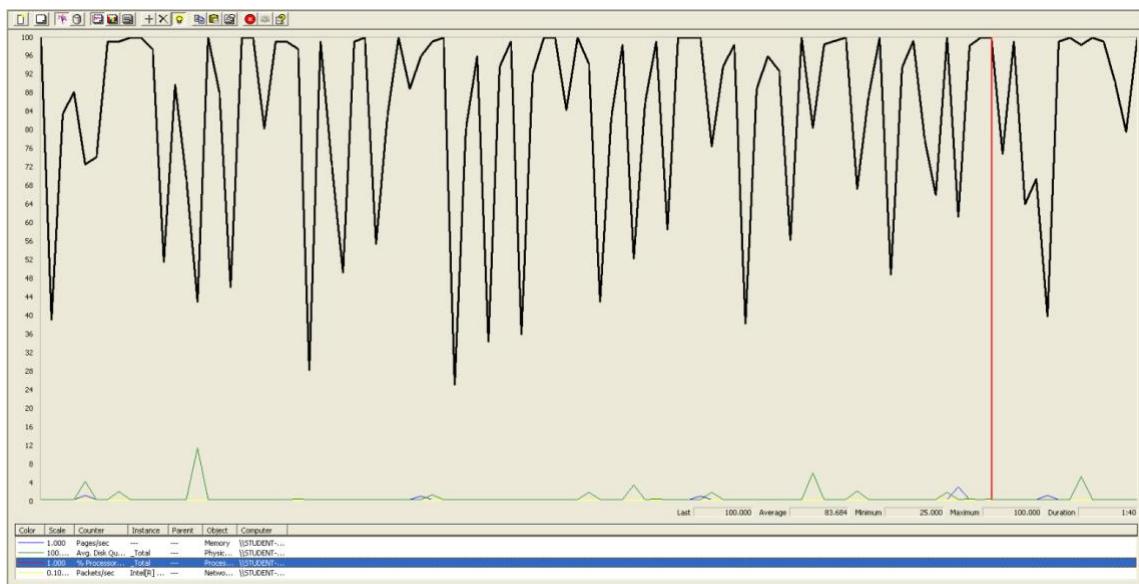
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check y /evaluation/moderator/list.do generan un tiempo de respuesta (1760 y 1011 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.500 ejecuciones por minuto (desciende levemente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



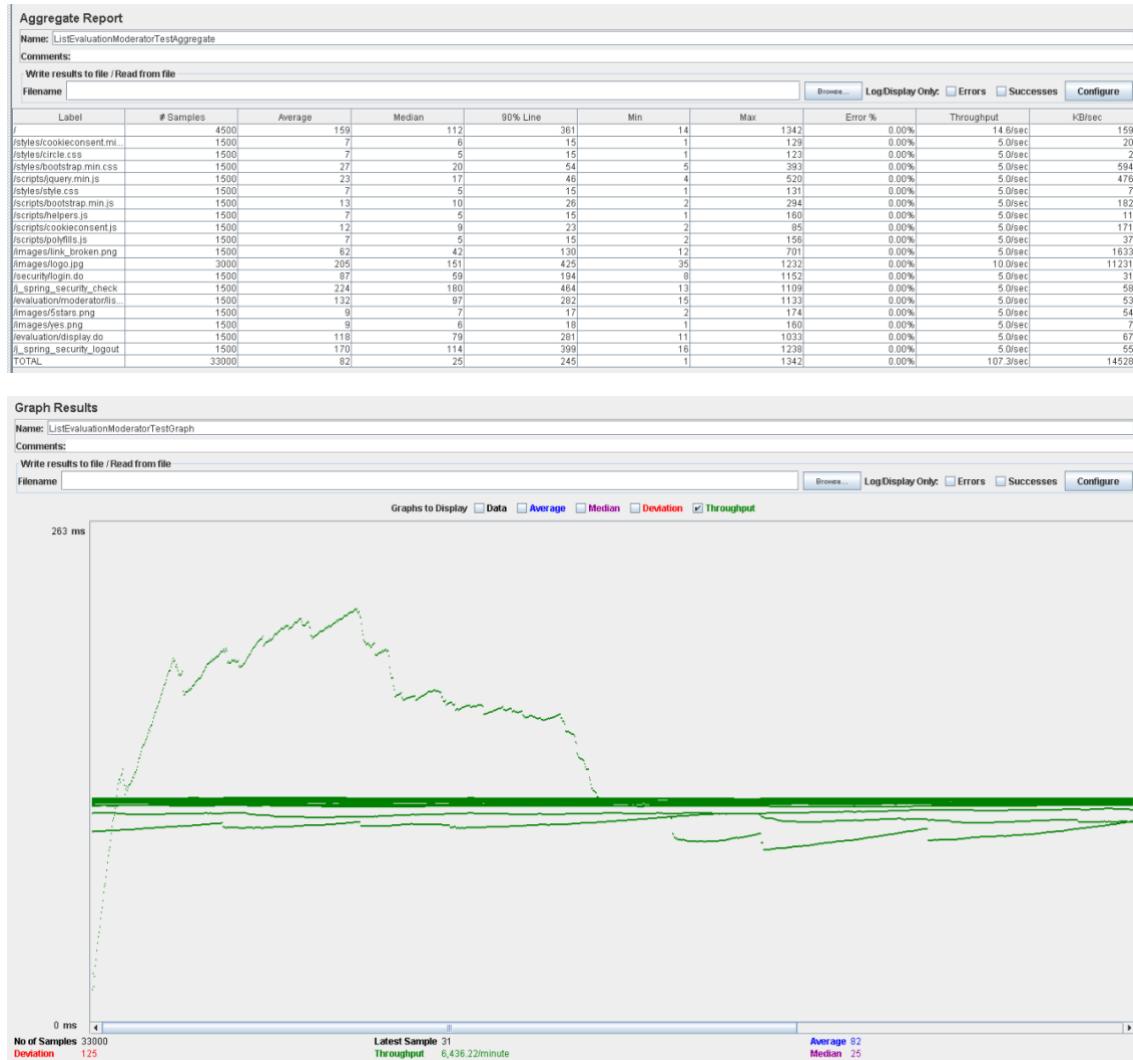
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 60.

Moderador listar las evaluaciones y desplegar evaluación.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a las evaluaciones y pulsamos en desplegar alguna de ellas.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

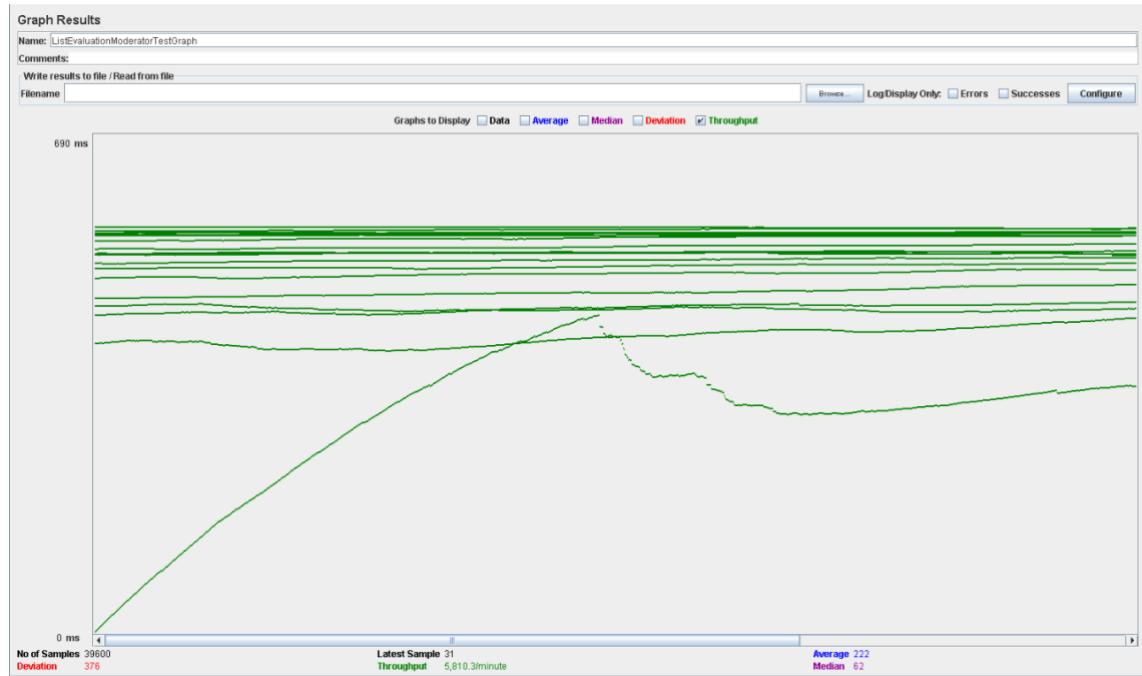


En esta tabla, el rendimiento es de aproximadamente 6.400 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 60 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report											
Name: ListEvaluationModeratorTestAggregate											
Comments:											
Write results to file / Read from file									Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes	<input checked="" type="checkbox"/> Configure
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec		
/	5400	507	390	912	14	5749	0.00%	13.2/sec	144.3		
/shiroCookieConsent.mi...	1800	24	14	35	2	880	0.00%	4.6/sec	18.8		
/shiroCookieConsent.cs...	1800	27	13	34	2	148	0.00%	4.6/sec	2.2		
/scripts/jquery.min.css	1800	63	42	123	6	1473	0.00%	4.6/sec	540.9		
/scripts/jquery.min.js	1800	56	39	102	6	1172	0.00%	4.6/sec	434.1		
/scripts/bootstrap.min.css	1800	19	14	33	2	532	0.00%	4.6/sec	7.1		
/scripts/bootstrap.min.js	1800	29	21	56	3	594	0.00%	4.6/sec	166.5		
/scripts/helpers.js	1800	19	13	35	1	519	0.00%	4.6/sec	10.4		
/scripts/cookieconsent.js	1800	30	20	51	3	685	0.00%	4.6/sec	156.7		
/scripts/fontawesome.js	1800	22	14	36	2	609	0.00%	4.6/sec	34.4		
/images/logo.png	1800	109	105	200	13	2001	0.00%	4.6/sec	1493.7		
/images/logo.jpg	3600	376	304	743	36	3087	0.00%	9.1/sec	10760.0		
/securelogin.do	1800	283	218	589	9	2991	0.00%	4.6/sec	28.5		
/j_spring_security_check	1800	689	590	1317	22	4354	0.00%	4.6/sec	53.1		
/evaluationModeratorList...	1800	388	328	743	18	3054	0.00%	4.6/sec	49.9		
/images/5stars.png	1800	18	15	35	2	142	0.00%	4.6/sec	50.2		
/images/yes.png	1800	19	14	36	2	538	0.00%	4.6/sec	6.7		
/evaluationdisplay.do	1800	377	271	680	11	4994	0.00%	4.6/sec	61.9		
/j_spring_security_logout	1800	448	403	850	18	2850	0.00%	4.6/sec	50.4		
TOTAL	39600	222	621	1	1	5749	0.00%	98.9/sec	13115.4		

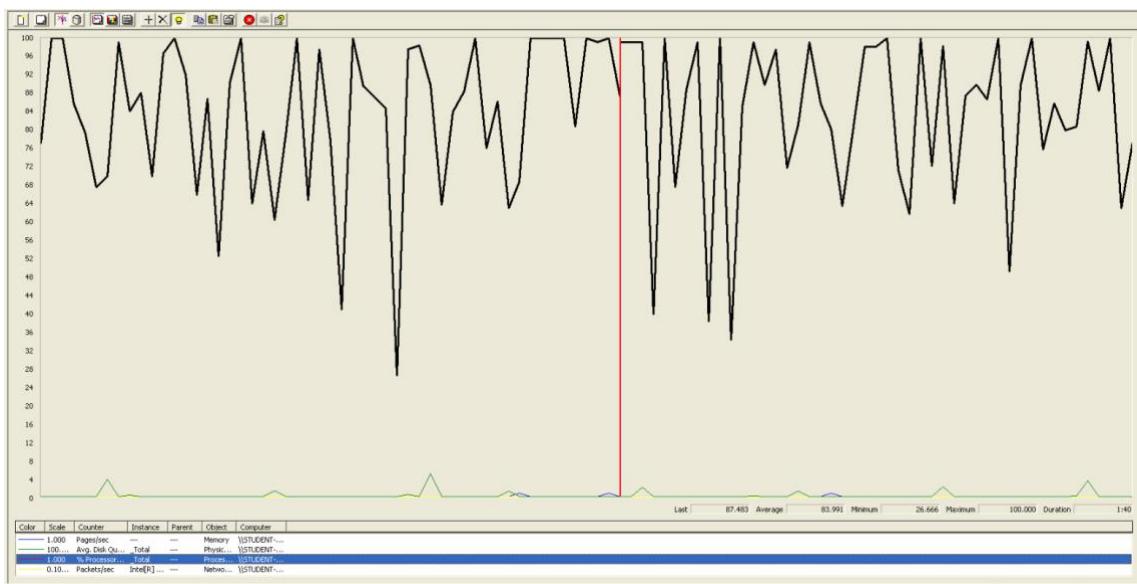
Como podemos observar fijandonos en 90% Line, la dirección /j_spring_security_check genera un tiempo de respuesta (1317 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.800 ejecuciones por minuto (desciende levemente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



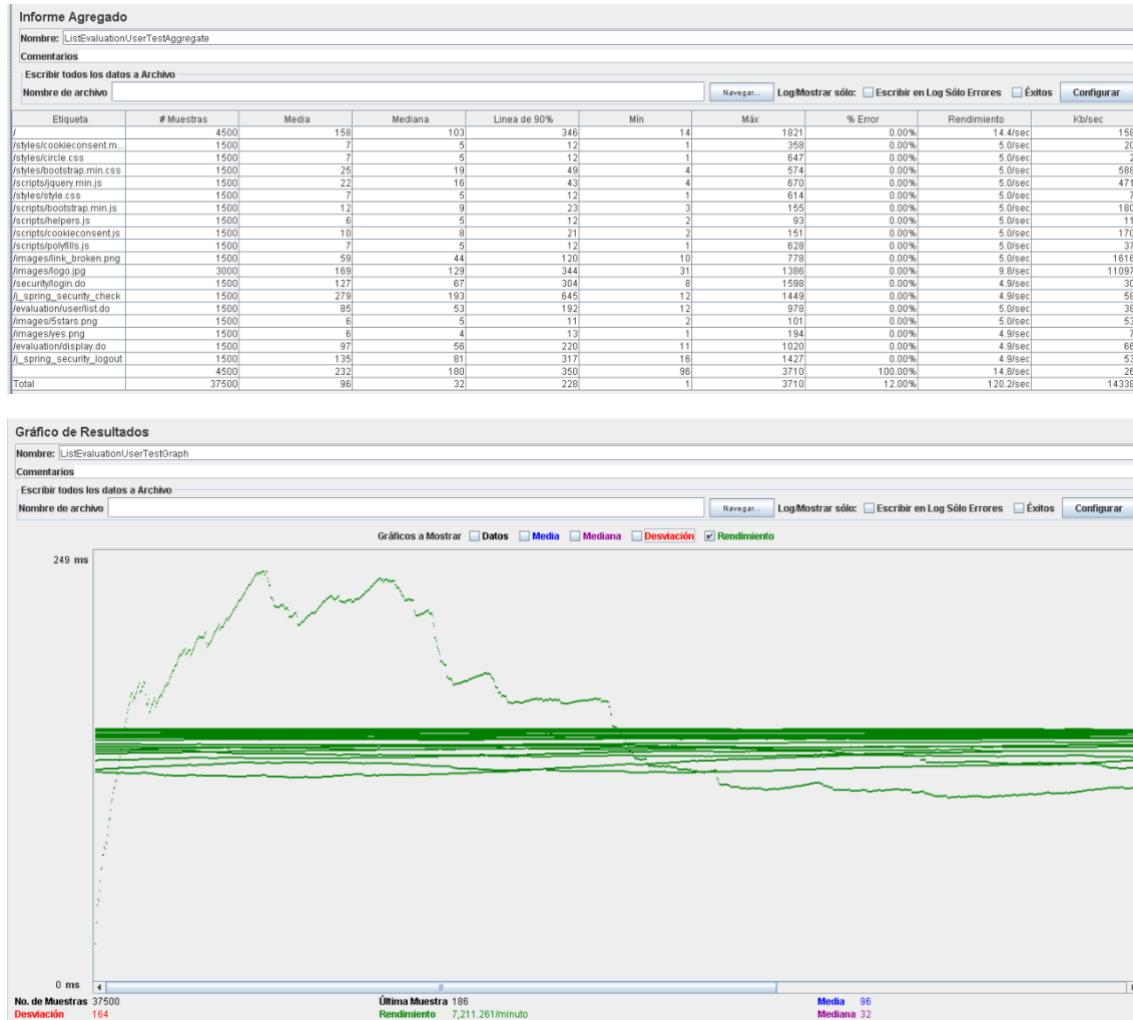
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 60.

Usuario listar mis evaluaciones y desplegar evaluación.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a nuestras evaluaciones y pulsamos en desplegar alguna de ellas.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:



En esta tabla, el rendimiento es de aproximadamente 7.200 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 60 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: ListEvaluationUserTestAggregate										
Comments:										
Write results to file / Read from file										
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	5400	576	437	1124	18	5518	0.00%	13.1/sec	143.9	
/styles/cookieconsent.m...	1800	12	10	22	2	275	0.00%	4.5/sec	18.4	
/styles/circle.css	1800	11	9	21	2	89	0.00%	4.5/sec	2.1	
/scripts/bootstrap.min.css	1800	45	35	82	5	1760	0.00%	4.5/sec	528.2	
/scripts/bootstrap.js	1800	37	30	67	5	100	0.00%	4.5/sec	423.9	
/scripts/bootstrap.css	1800	12	10	21	2	625	0.00%	4.5/sec	7.0	
/scripts/bootstrap.min.js	1800	20	17	37	3	139	0.00%	4.5/sec	162.5	
/scripts/helpers.js	1800	11	10	21	2	222	0.00%	4.5/sec	10.2	
/scripts/cookieconsent.js	1800	18	15	32	3	127	0.00%	4.5/sec	152.9	
/scripts/polyfills.js	1800	12	10	21	2	180	0.00%	4.5/sec	33.5	
/imageoflink_broken.png	1800	95	78	184	11	677	0.00%	4.5/sec	1452.7	
/images/logo.jpg	3600	273	225	495	35	1879	0.00%	8.9/sec	10914.6	
/securelogin.do	1800	425	269	1017	12	4317	0.00%	4.5/sec	27.8	
/j_spring_security_check	1800	1007	611	1918	31	511	0.00%	4.5/sec	62.3	
/securelogout.do	1800	433	300	950	12	3121	0.00%	4.5/sec	34.9	
/images/favstar.png	1800	14	12	26	2	329	0.00%	4.5/sec	6.7	
/images/sves.png	1800	14	11	27	2	230	0.00%	4.5/sec	6.5	
/evaluationdisplay.do	1800	484	330	996	12	4663	0.00%	4.5/sec	60.6	
/j_spring_security_logout	1800	557	424	1095	21	4199	0.00%	4.5/sec	49.8	
TOTAL	39600	249	54	686	2	5711	0.00%	96.2/sec	13915.4	

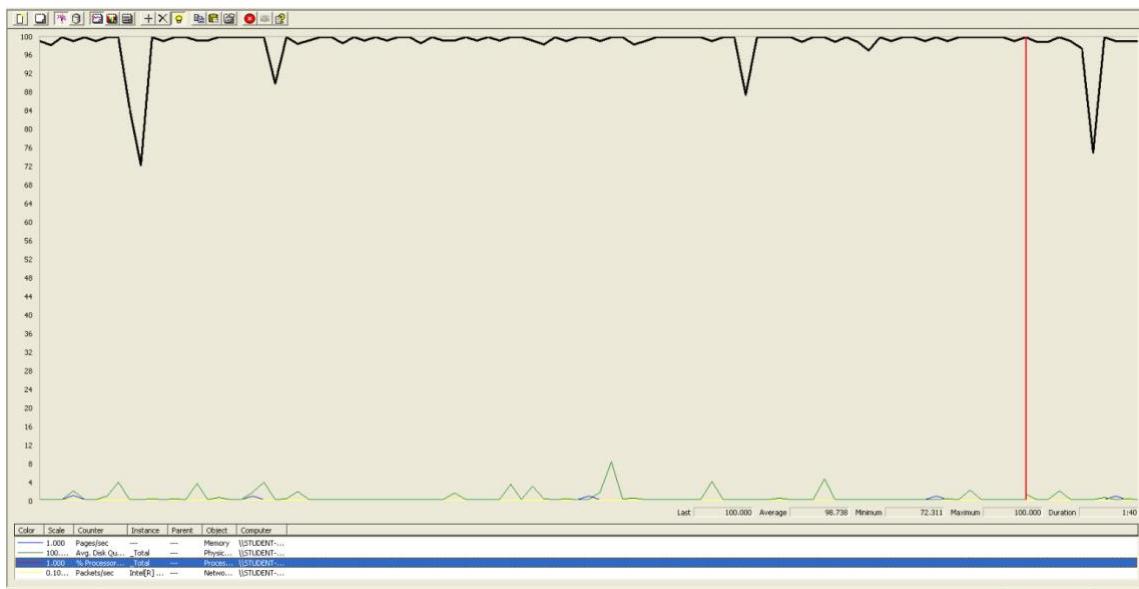
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check genera un tiempo de respuesta (1918 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.800 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



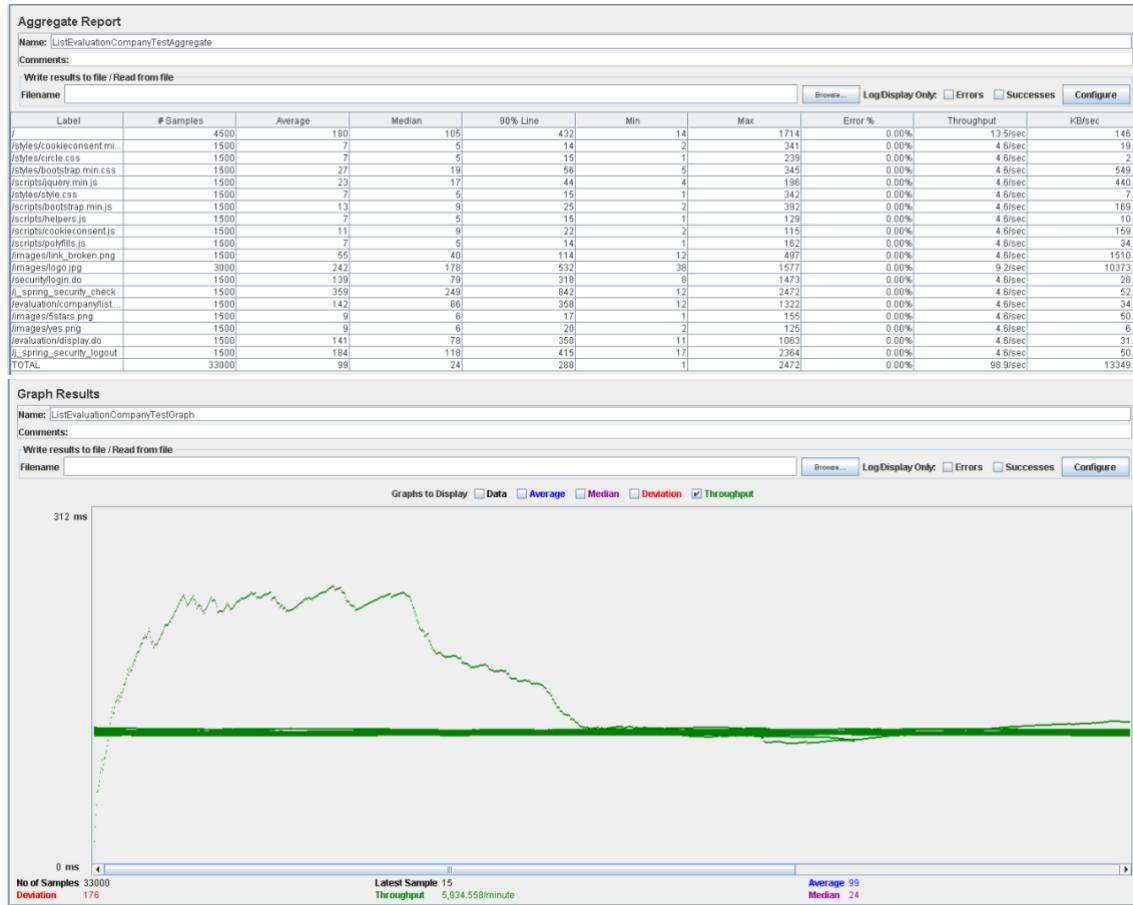
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 60.

Compañía listar mis evaluaciones y desplegar evaluación.

En este caso de uso, accedemos a la página, nos logueamos como compañía, accedemos a las evaluaciones recibidas y pulsamos en desplegar alguna de ellas.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

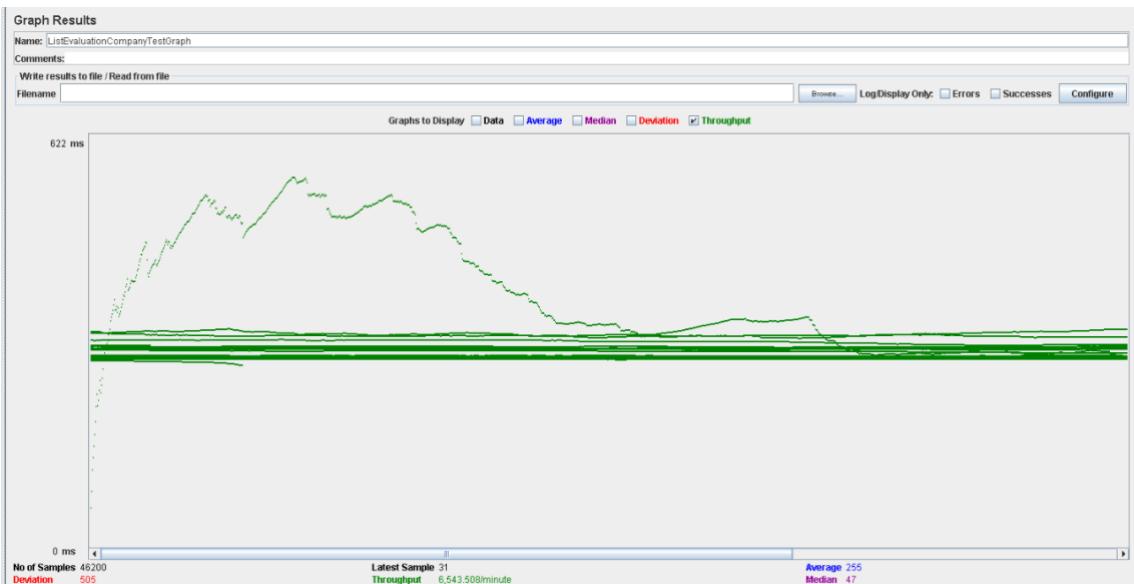


En esta tabla, el rendimiento es de aproximadamente 6.000 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 70 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: ListEvaluationCompanyTestAggregate										
Comments:										
Write results to file / Read from file										Configure
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	6300	575	389	1245	14	8934	0.00%	14.9/sec	161.5	
/styles/cookieconsent.mi...	2100	11	9	17	1	1142	0.00%	5.1/sec	20.9	
/	2100	111	93	171	2	1143	0.00%	5.1/sec	21.4	
/scripts/bootstrap.min.css	2100	37	31	69	6	748	0.00%	5.1/sec	600.0	
/scripts/jquery.min.js	2100	31	25	54	5	492	0.00%	5.1/sec	481.4	
/style/style.css	2100	10	8	19	1	296	0.00%	5.1/sec	7.9	
/scripts/bootstrap.min.js	2100	18	15	32	3	1179	0.00%	5.1/sec	184.5	
/scripts/helpers.js	2100	10	8	17	2	371	0.00%	5.1/sec	11.5	
/scripts/cookieconsent.js	2100	16	14	27	3	483	0.00%	5.1/sec	173.6	
/scripts/polifills.js	2100	10	9	17	2	320	0.00%	5.1/sec	38.1	
/images/line_broken.png	2100	83	69	156	13	816	0.00%	5.1/sec	1548.7	
/	4200	255	200	379	46	2107	0.00%	10.4/sec	1135.5	
/securelogin.do	2100	457	254	1098	8	8425	0.00%	5.1/sec	31.5	
/j_spring_security_check	2100	1145	862	2379	21	9130	0.00%	5.1/sec	57.7	
/evaluation/companylist...	2100	472	313	1052	12	5683	0.00%	5.1/sec	37.3	
/images/5stars.png	2100	12	11	21	2	348	0.00%	5.1/sec	55.1	
/images/yes.png	2100	11	9	20	1	251	0.00%	5.1/sec	7.3	
/evaluation/display/do	2100	508	315	1154	12	6762	0.00%	5.1/sec	34.3	
/j_spring_security_logout	2100	569	395	1261	12	6521	0.00%	5.1/sec	55.3	
TOTAL	46200	255	47	666	1	9130	0.00%	10.9/sec	14219.5	

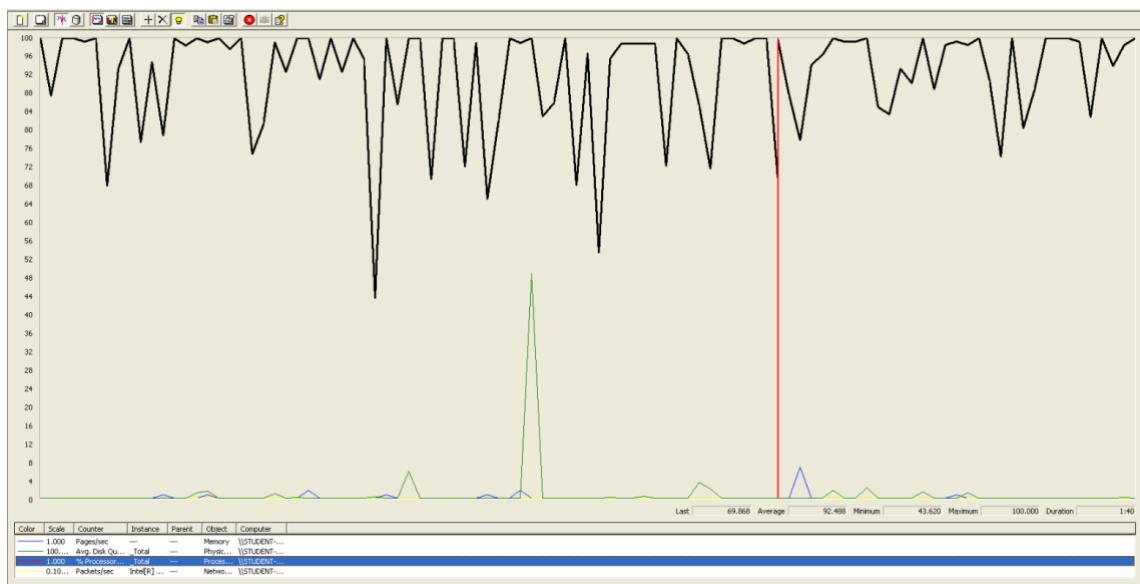
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check genera un tiempo de respuesta (2379 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 6.500 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



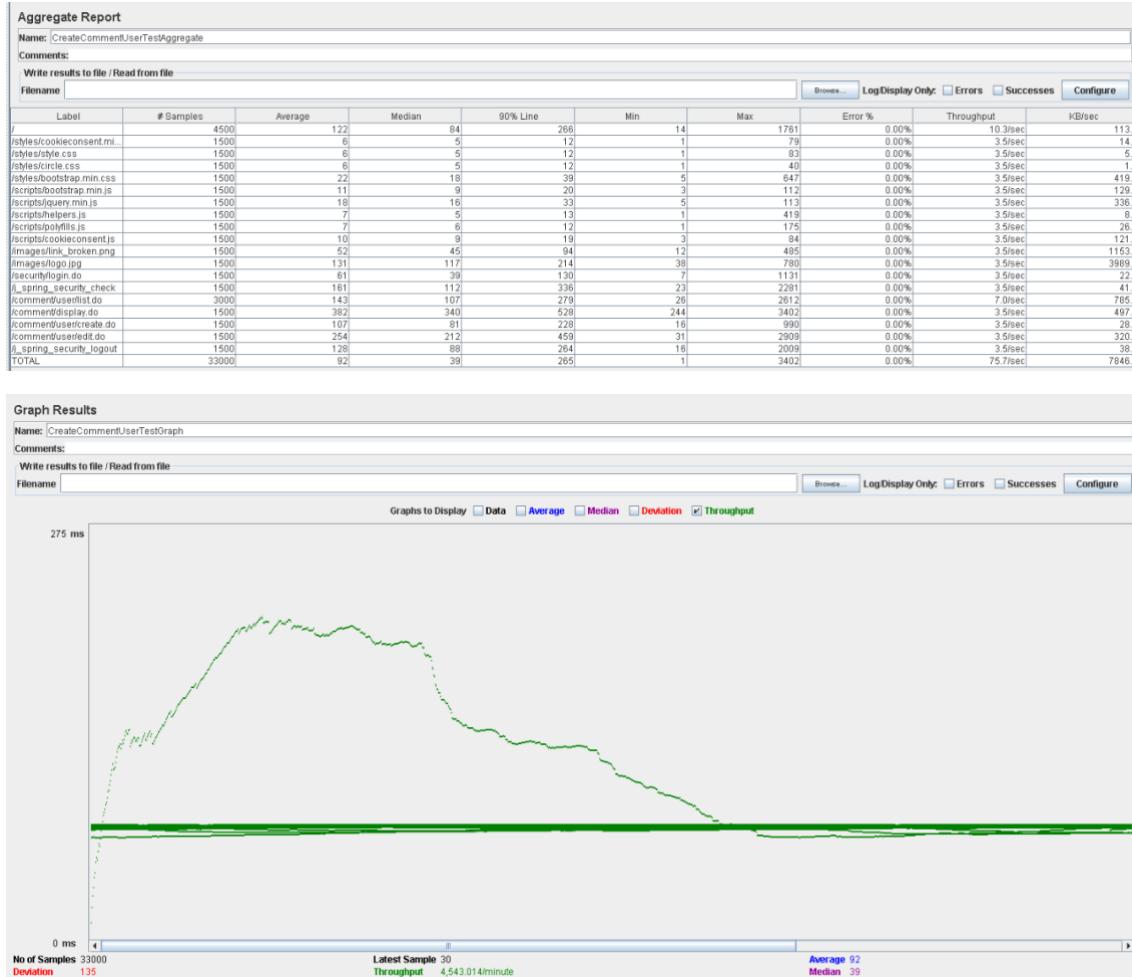
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 70.

Usuario crear comentario.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a mis comentarios y pulsamos en desplegar alguno de ellos, finalmente pulsamos en comentar.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

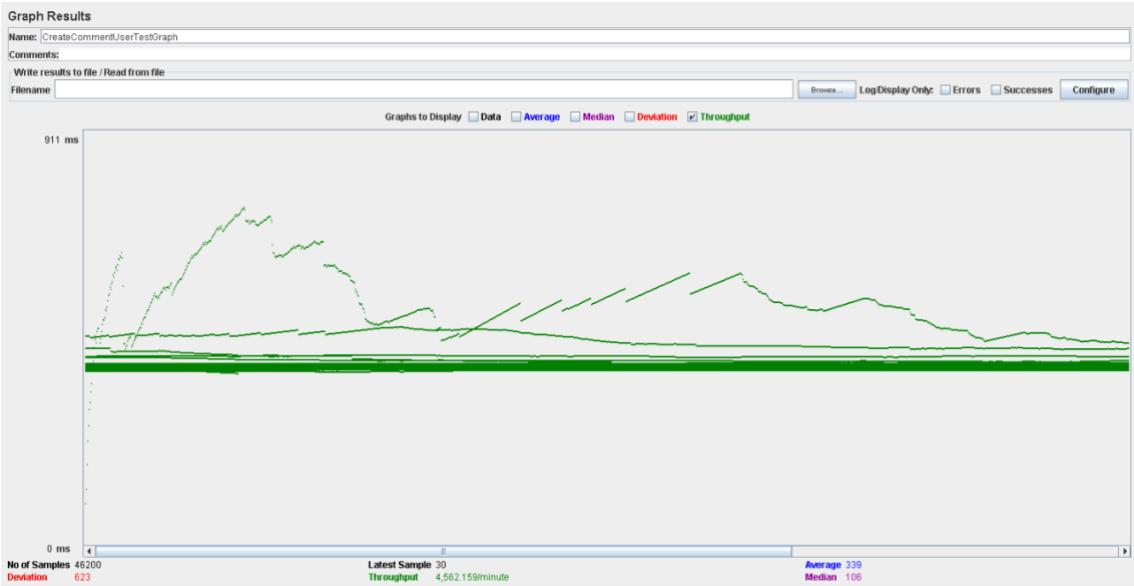


En esta tabla, el rendimiento es de aproximadamente 4.500 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 70 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: CreateCommentUserTestAggregate										
Comments:										
Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	6300	525	349	1110	15	6784	0.00%	10.4/sec	113.8	
/styles/cookieconsent.mi...	2100	12	10	22	2	453	0.00%	3.5/sec	14.5	
/views/_comment/_comment...	2100	12	9	23	2	453	0.00%	3.5/sec	14.5	
/views/circle.css	2100	12	10	23	1	185	0.00%	3.5/sec	17	
/views/bootstrap.min.css	2100	33	27	59	7	534	0.00%	3.5/sec	417.1	
/scripts/bootstrap.min.js	2100	19	15	34	4	567	0.00%	3.5/sec	128.2	
/scripts/helpers.js	2100	13	10	24	2	569	0.00%	3.5/sec	8.0	
/scripts/pohfills.js	2100	12	10	23	2	180	0.00%	3.5/sec	26.5	
/scripts/cookieconsent.js	2100	18	15	33	3	663	0.00%	3.5/sec	120.7	
/scripts/broken.png	2100	72	60	177	16	844	0.00%	3.5/sec	11.1	
/images/logo.png	2100	399	169	342	49	921	0.00%	3.5/sec	3969.6	
/securitylogin.do	2100	440	217	1036	9	5840	0.00%	3.5/sec	22.0	
/j_spring_security_check	2100	1000	668	2167	4	8037	0.00%	3.5/sec	41.3	
/comment/userlist.do	4200	553	388	1132	22	9491	0.00%	7.0/sec	216.5	
/comment/display.do	2100	872	659	1453	235	22233	0.00%	3.5/sec	131.9	
/comment/usercreate.do	2100	512	344	1133	18	5823	0.00%	3.5/sec	28.6	
/comment/useredit.do	2100	967	691	1967	51	9042	0.00%	3.5/sec	80.6	
/j_spring_security_logout	2100	536	343	1125	18	7069	0.00%	3.5/sec	38.5	
TOTAL	46200	339	106	970	1	22233	0.00%	78.0/sec	6726.2	

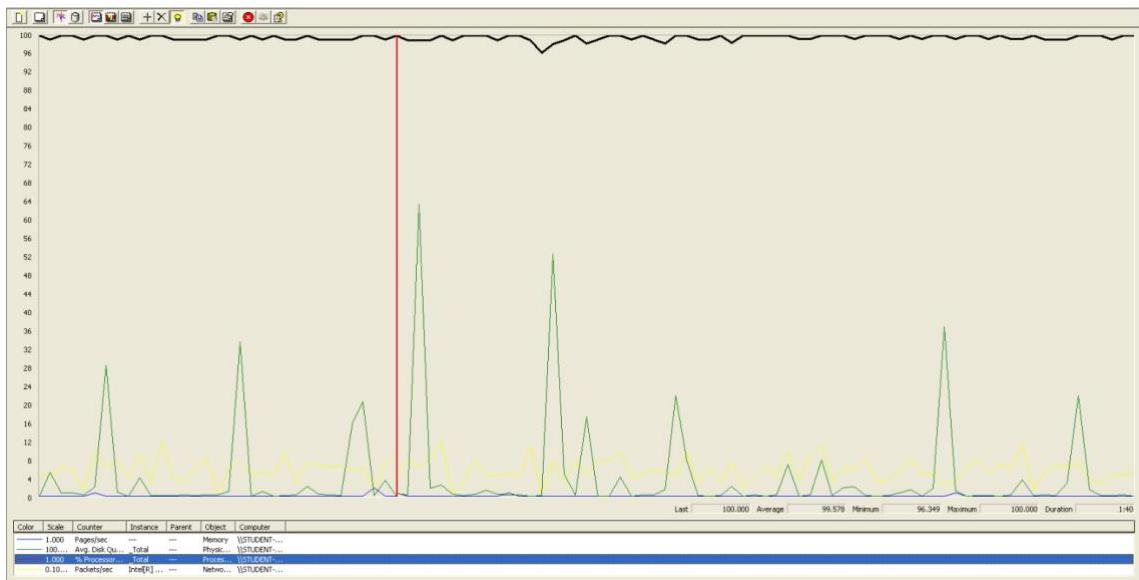
Como podemos observar fijándonos en 90% Line, las direcciones con mayores tiempos de respuesta son /j_spring_security_check, /comment/user/display.do y /comment/user/edit.do, que generan un tiempo de respuesta de 2167, 1453 y 1967 ms respectivamente.



En esta tabla, el rendimiento es de aproximadamente 4.500 ejecuciones por minuto (aumenta muy levemente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



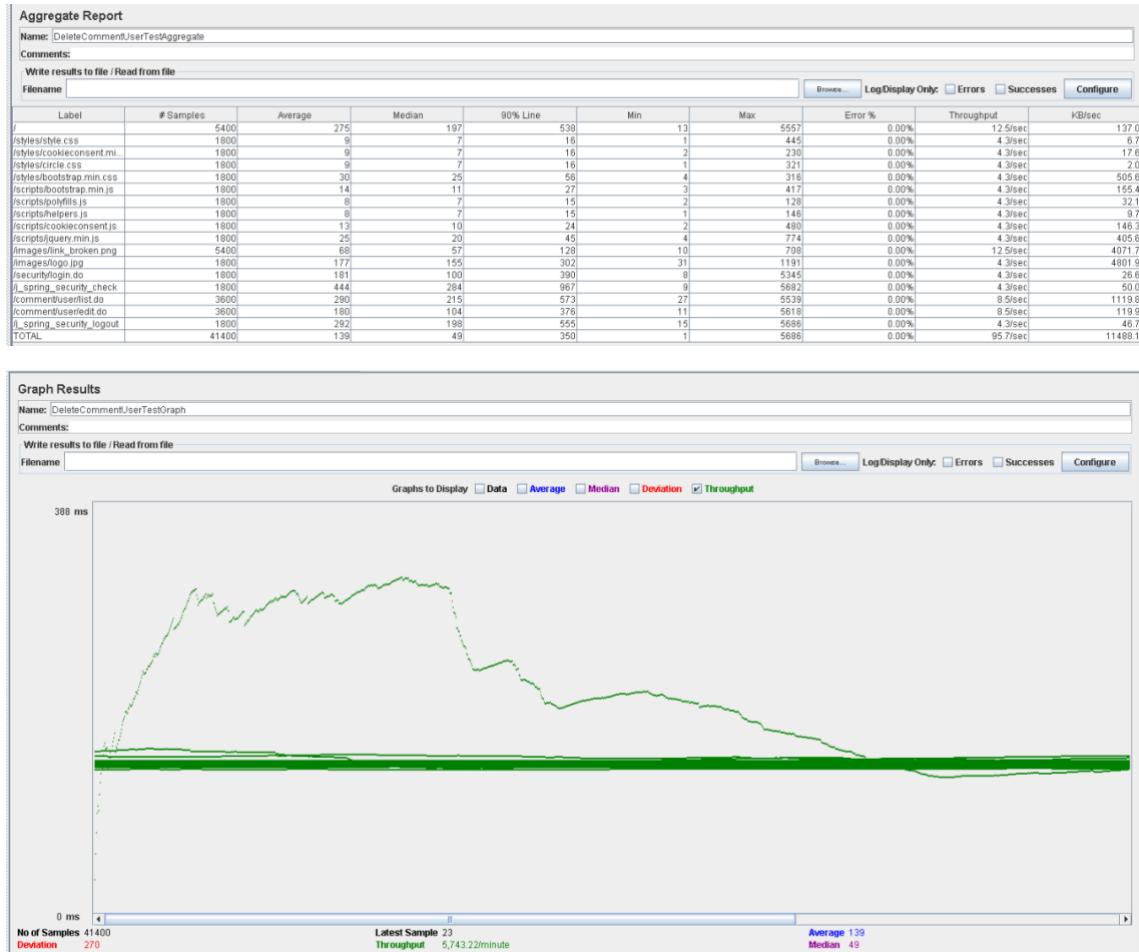
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 70.

Usuario borrar comentarios.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a mis comentarios y pulsamos en editar alguno de ellos.

Comenzamos probando con 60 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

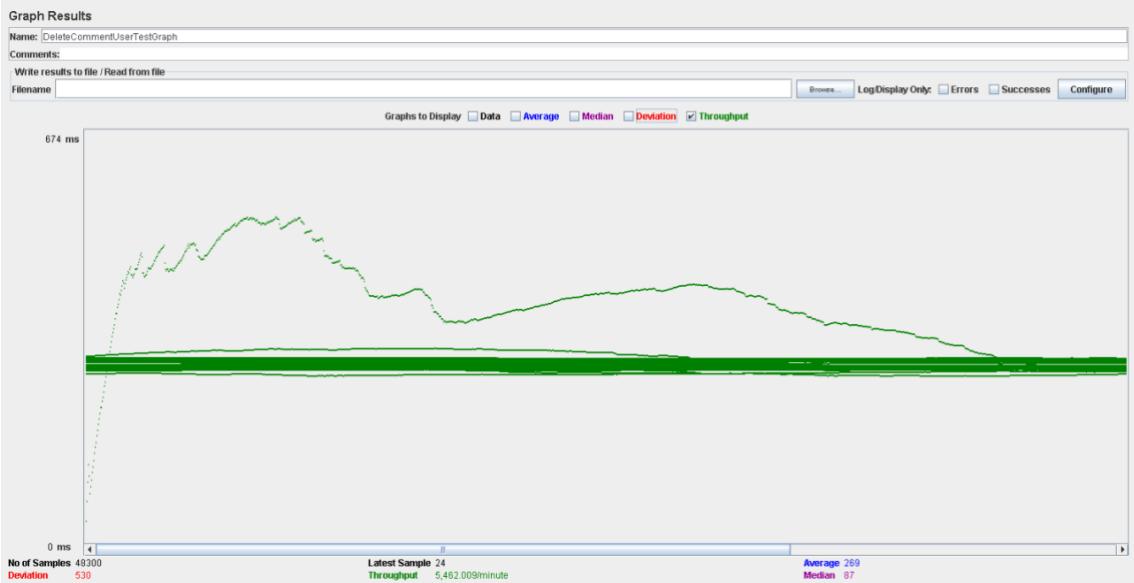


En esta tabla, el rendimiento es de aproximadamente 5.700 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 70 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										Browses	Log	Display Only	Errors	Successes	Configure
Comments:															
Write results to file / Read from file															
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec						
/styles/style.css	6300	522	349	1032	14	9700	0.00%	11.9/sec	130.3						
/styles/style.css	2100	12	9	21	2	1102	0.00%	4.0/sec	0.3						
/styles/cookieconsent.min.js	2100	11	9	20	2	120	0.00%	4.0/sec	16.7						
/styles/style.css	2100	11	9	21	2	276	0.00%	4.0/sec	1.9						
/scripts/bootstrap.min.css	2100	36	33	67	7	568	0.00%	4.0/sec	400.0						
/scripts/bootstrap.min.js	2100	18	16	22	3	123	0.00%	4.0/sec	147.6						
/scripts/bootstrap.js	2100	11	9	20	2	247	0.00%	4.0/sec	30.5						
/scripts/helpers.js	2100	11	9	19	2	250	0.00%	4.0/sec	9.2						
/scripts/cookieconsent.js	2100	16	14	30	3	206	0.00%	4.0/sec	138.9						
/scripts/bootstrap.min.js	2100	32	26	56	5	872	0.00%	4.0/sec	385.2						
/images/logo_broken.png	6300	67	76	152	9	797	0.00%	11.9/sec	393.7						
/images/logo.png	2100	213	197	343	23	878	0.00%	4.0/sec	4562.3						
/scripts/login.do	2100	409	235	901	9	7820	0.00%	4.0/sec	25.2						
/_spring_security_check	2100	1047	696	2218	26	15196	0.00%	4.1/sec	47.6						
/commentUserList.do	4200	573	413	1087	29	8229	0.00%	8.1/sec	1064.4						
/commentUserEdit.do	4200	416	262	876	11	9165	0.00%	8.1/sec	113.9						
/_spring_security_logout	2100	556	360	1144	17	8098	0.00%	4.1/sec	44.3						
TOTAL	48306	266	87	638	2	15196	0.00%	91.0/sec	10925.6						

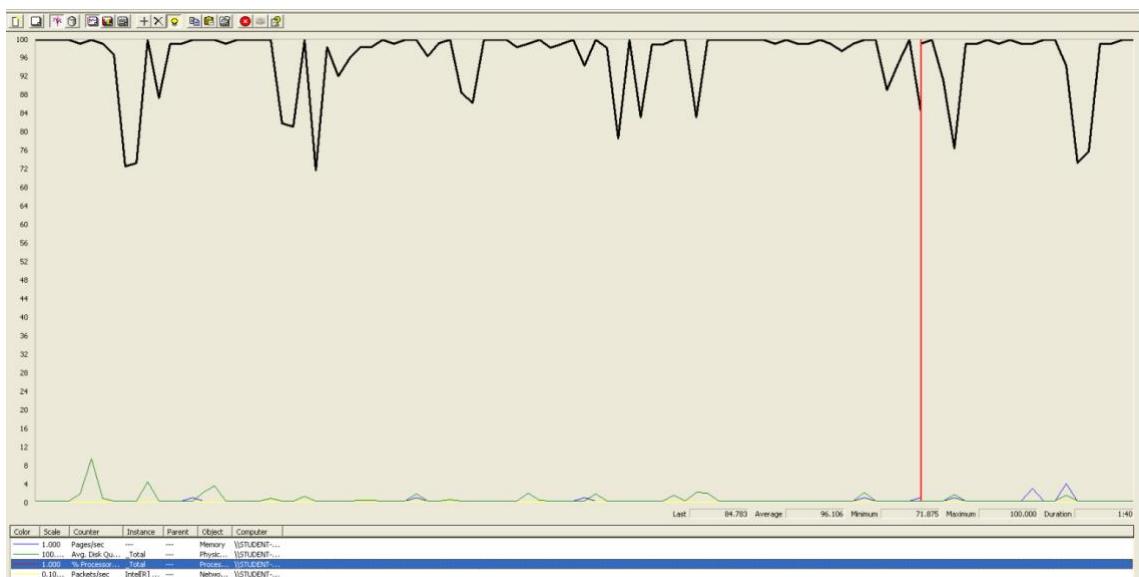
Como podemos observar fijándonos en 90% Line, las direcciones /j_spring_security_check, /comment/user/list.do y /comment/user/edit.do generan un tiempo de respuesta (2218, 1087 y 1144 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.400 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



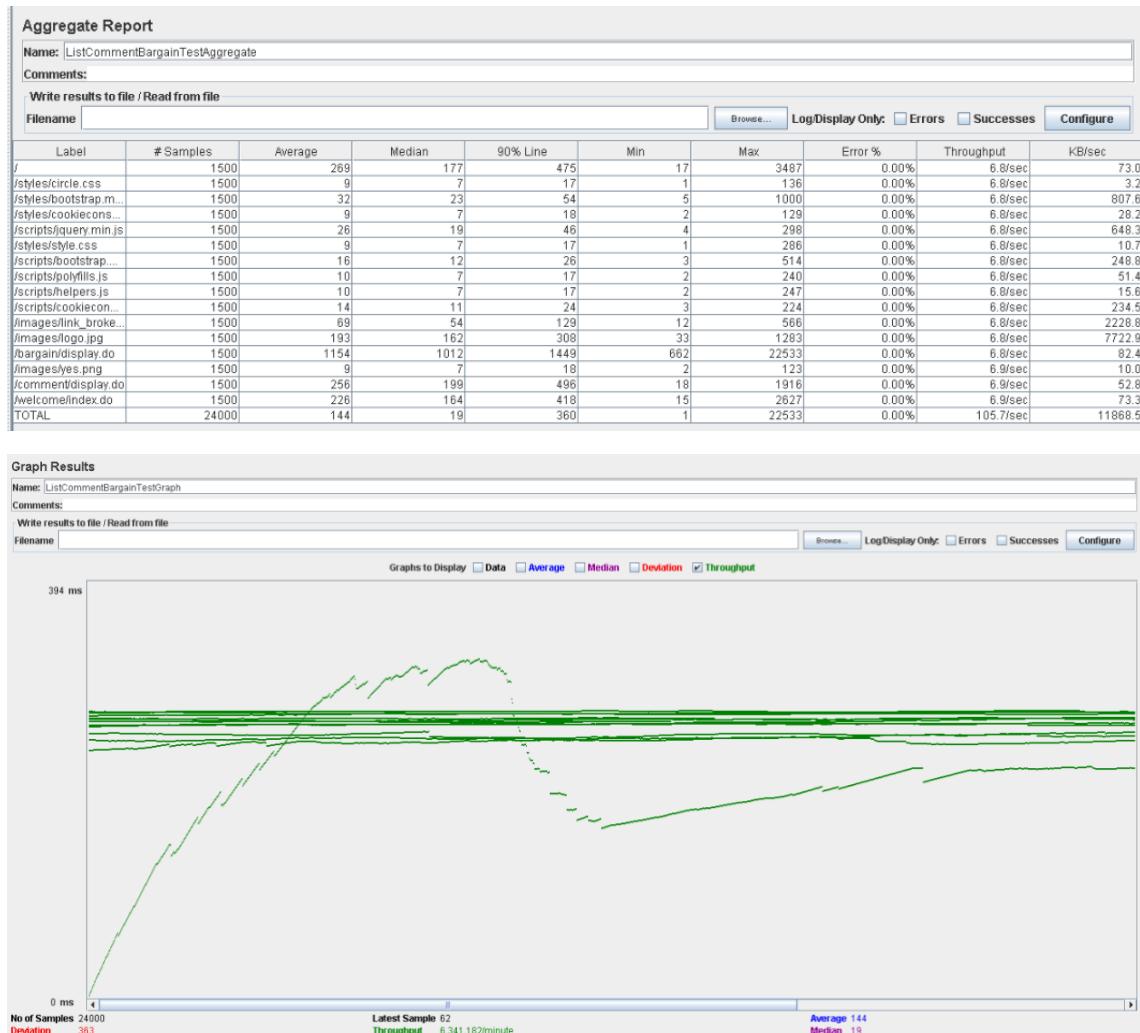
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 60 y 70.

Listar los comentarios de un chollo y desplegar comentario.

En este caso de uso, accedemos a la página, accedemos al display de un chollo y accedemos a sus comentarios.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:



En esta tabla, el rendimiento es de aproximadamente 6.300 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 60 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: ListCommentBargainTestAggregate										
Comments:										
Write results to file / Read from file									Browse...	Log Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input checked="" type="checkbox"/> Configure
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	1900	1072	9	384	3057	16	13886	5.0/sec	62.6	
/bargain/style.css	1900	9	6	19	1	252	0.00%	5.0/sec	3.8	
/bargain/bootstrap.min.css	1900	29	21	69	6	295	0.00%	5.0/sec	689.3	
/bargain/cookieconsent.min.js	1900	10	7	20	2	158	0.00%	5.0/sec	24.0	
/scripts/jquery.min.js	1900	25	18	49	4	368	0.00%	5.0/sec	553.1	
/styles/style.css	1900	10	6	19	1	215	0.00%	5.0/sec	9.1	
/scripts/bootstrap.min.js	1900	16	11	30	3	285	0.00%	5.0/sec	212.0	
/scripts/polyfills.js	1900	10	7	19	2	258	0.00%	5.0/sec	43.8	
/scripts/help.js	1900	10	7	20	1	208	0.00%	5.0/sec	13.3	
/comment/commentsent.js	1900	14	10	30	2	243	0.00%	5.0/sec	16.6	
/image/film_broken.png	1900	62	47	121	16	474	0.00%	5.0/sec	1894.5	
/images/icon.jpg	1900	183	149	346	29	993	0.00%	5.0/sec	6569.7	
/bargain/display.do	1900	2679	1923	4812	695	31113	0.00%	5.0/sec	70.2	
/images/yes.png	1900	11	7	24	1	212	0.00%	5.0/sec	8.5	
/comment/display.do	1900	1061	420	2963	14	14237	0.00%	5.0/sec	44.0	
/welcome/index.do	1900	1094	391	3007	13	14229	0.00%	5.0/sec	62.2	
TOTAL	28900	393	19	1346	1	31113	0.00%	91.9/sec	10316.5	

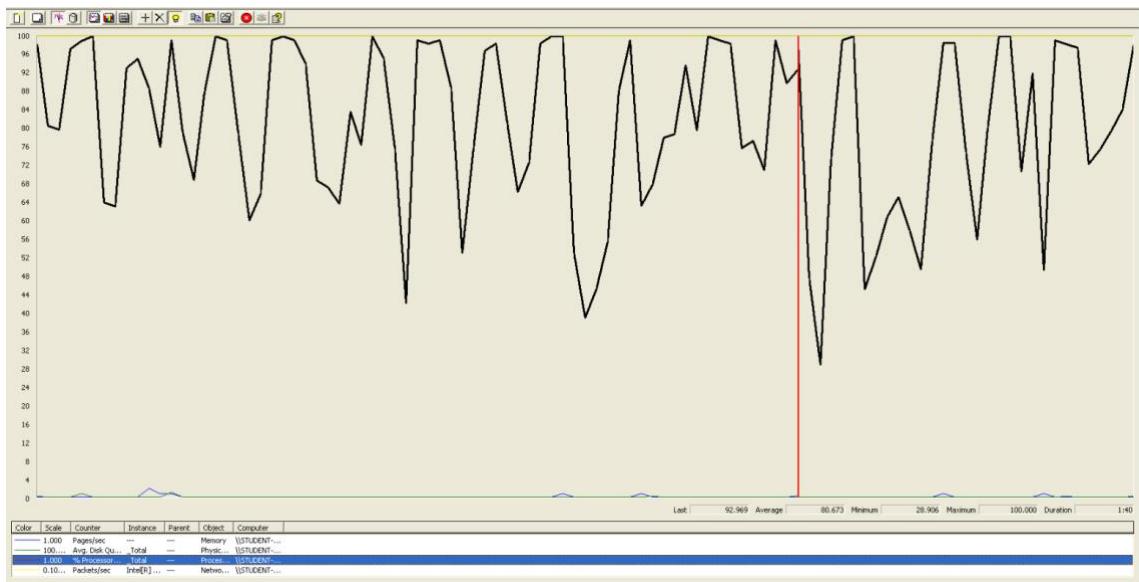
Como podemos observar fijandonos en 90% Line, las direcciones /bargain/display.do, /comment/display.do y /welcome/index.do generan un tiempo de respuesta (4812, 2963 y 3007 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.500 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



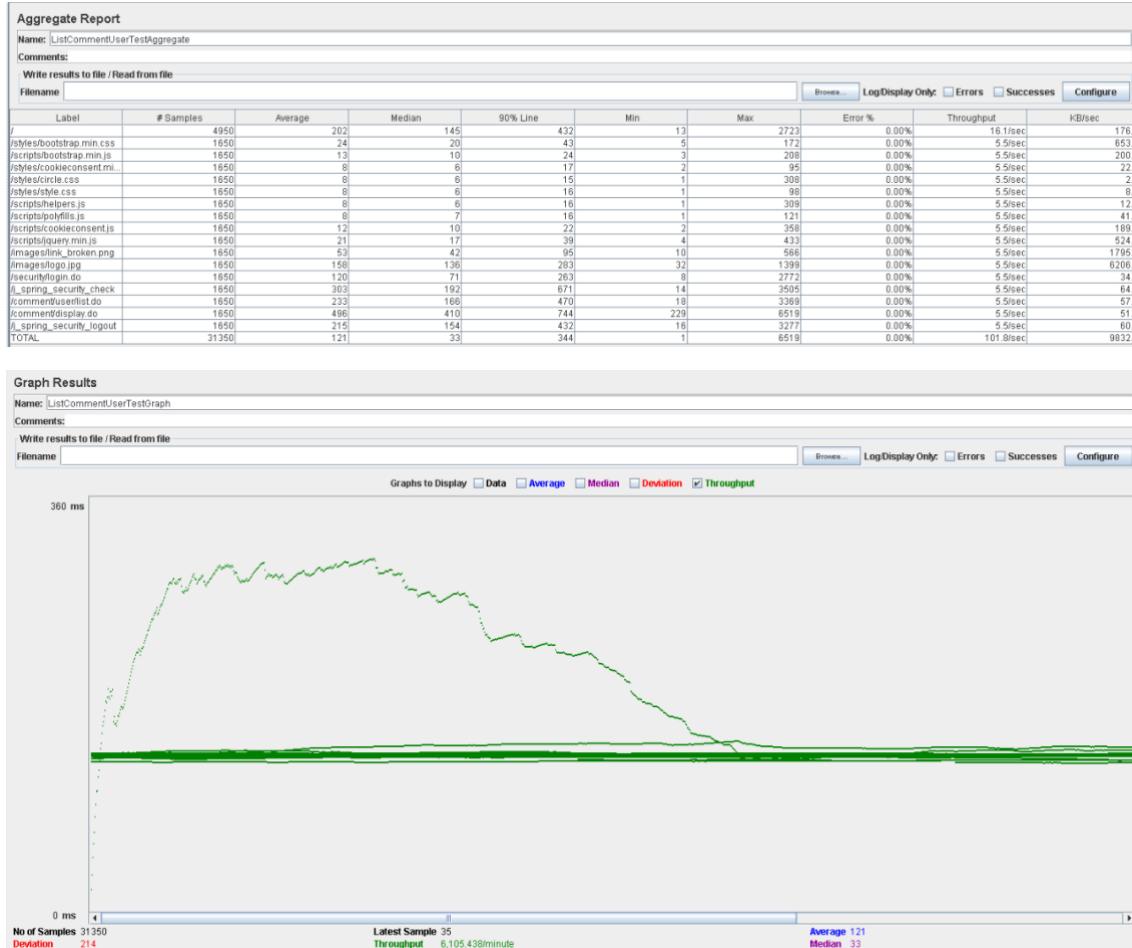
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 60.

Usuario listar mis comentarios y desplegar comentario.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a mis comentarios y pulsamos en desplegar alguno de ellos.

Comenzamos probando con 55 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

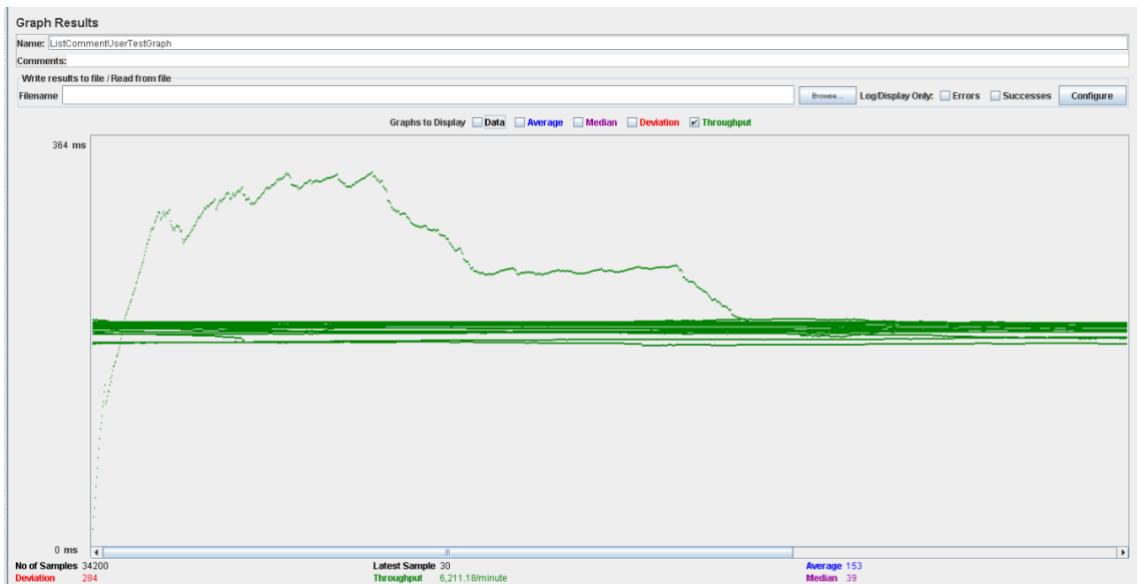


En esta tabla, el rendimiento es de aproximadamente 6.100 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 60 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report											
Name: ListCommentUserTestAggregate											
Comments:											
Write results to file / Read from file											
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec		
/	5400	270	189	620	14	5318	0.00%	16.3/sec	179.4		
/styles/bootstrap.min.css	1800	26	21	47	5	325	0.00%	5.6/sec	663.8		
/scripts/bootstrap.min.js	1800	14	11	29	2	340	0.00%	5.6/sec	20.1		
/scripts/cookieconsent.min.js	1800	10	8	18	2	525	0.00%	5.6/sec	23.1		
/styles/circle.css	1800	9	7	17	1	314	0.00%	5.6/sec	2.7		
/styles/style.css	1800	9	7	18	2	186	0.00%	5.6/sec	8.7		
/scripts/helpers.js	1800	9	7	18	1	121	0.00%	5.6/sec	12.8		
/scripts/polyfill.js	1800	10	8	19	2	121	0.00%	5.6/sec	42.1		
/scripts/cookieconsent.js	1800	13	11	25	2	117	0.00%	5.6/sec	192.1		
/scripts/cookieconsent.min.js	1800	24	19	43	4	211	0.00%	5.6/sec	27		
/images/broken.png	1800	27	46	105	11	297	0.00%	5.6/sec	1824.6		
/images/logo.png	1800	169	145	387	23	931	0.00%	5.6/sec	6305.6		
/securitylogin.do	1800	106	107	388	8	3903	0.00%	5.6/sec	34.9		
/j_spring_security_check	1800	446	291	1001	23	4722	0.00%	5.6/sec	65.6		
/commentUserList.do	1800	289	220	561	18	3226	0.00%	5.6/sec	58.5		
/commentIdisplay.do	1800	544	451	834	229	5517	0.00%	5.6/sec	52.5		
/j_spring_security_logout	1800	287	208	561	19	5749	0.00%	5.6/sec	60.9		
TOTAL	34200	153	39	412	1	5749	0.00%	103.5/sec	10002.7		

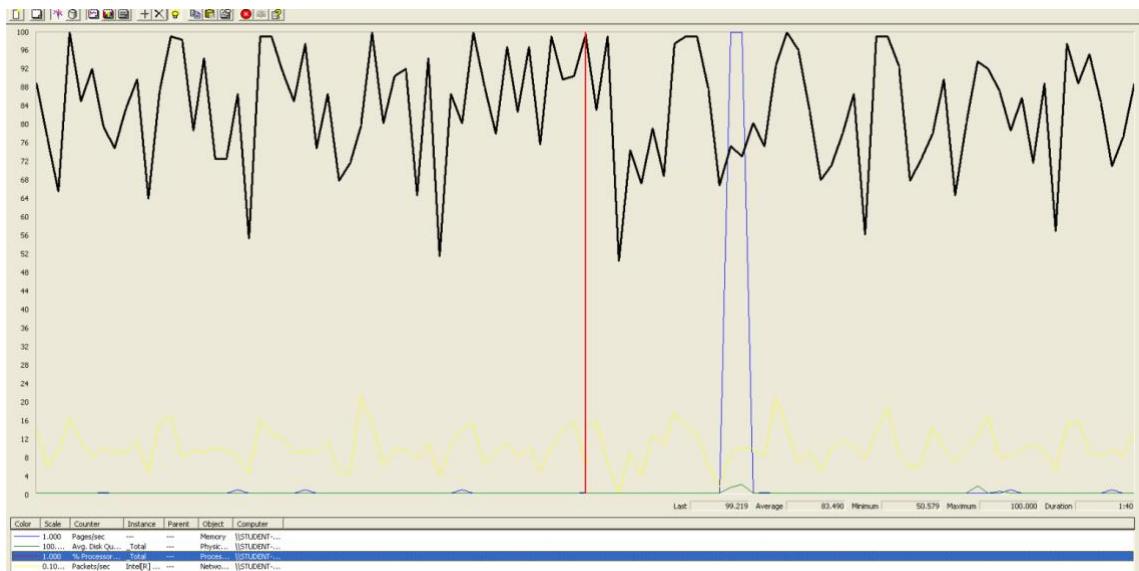
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check genera un tiempo de respuesta (1001 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 6.200 ejecuciones por minuto (aumenta al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



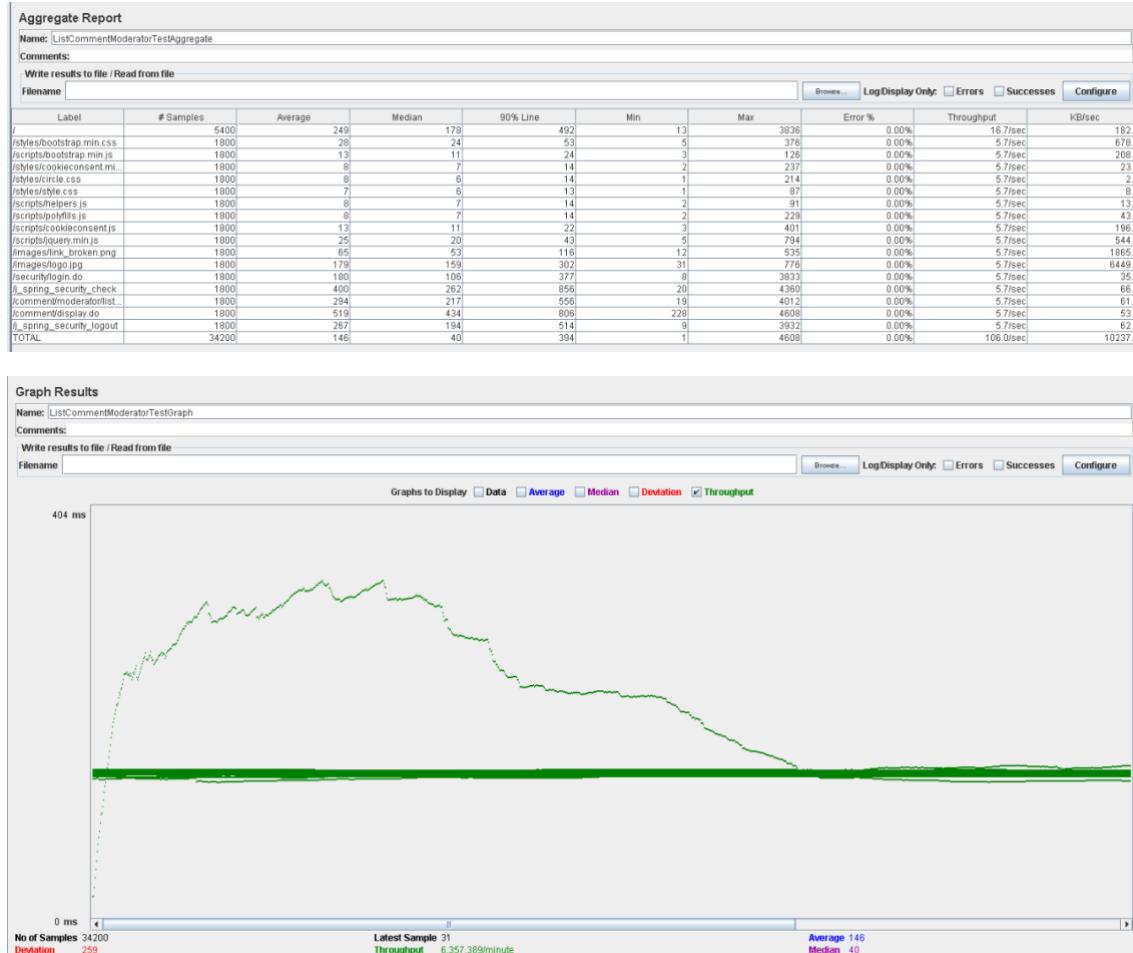
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 55 y 60.

Moderador listar los comentarios y desplegar comentario.

En este caso de uso, accedemos a la página, nos logueamos como moderador, accedemos a los comentarios y pulsamos en desplegar alguno de ellos

Comenzamos probando con 60 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:



En esta tabla, el rendimiento es de aproximadamente 6.300 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 70 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report												
Name: ListCommentModeratorTestAggregate												
Comments:												
Write results to file / Read from file												
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec			
/	6300	295	208	568	13	4688	0.00%	18.4/sec	201.4			
/styles/bootstrap.min.css	2100	29	24	50	4	910	0.00%	6.3/sec	747.0			
/scripts/bootstrap.min.js	2100	14	11	23	2	652	0.00%	6.3/sec	229.7			
/scripts/cookieconsent.min.js	2100	7	6	19	2	324	0.00%	6.3/sec	26.0			
/stylestrict.css	2100	7	6	12	2	239	0.00%	6.3/sec	3.0			
/stylestrictle.css	2100	7	6	13	1	414	0.00%	6.3/sec	9.0			
/scripts/helpers.js	2100	8	6	14	1	283	0.00%	6.3/sec	14.4			
/scripts/polyfills.js	2100	8	6	13	1	345	0.00%	6.3/sec	47.4			
/scripts/cookieconsent.js	2100	12	10	21	2	313	0.00%	6.3/sec	216.1			
/scripts/query.min.js	2100	24	19	43	4	875	0.00%	6.3/sec	599.3			
/images/link_broken.png	2100	66	56	119	10	450	0.00%	6.3/sec	2052.0			
/commentdisplay.do	2100	179	154	289	33	207	0.00%	6.3/sec	705.0			
/securelogin.do	2100	223	131	404	71	3464	0.00%	6.3/sec	39.3			
/j_spring_security_check	2100	513	347	1134	17	6250	0.00%	6.3/sec	73.0			
/comment/moderatorlist	2100	356	255	688	17	4543	0.00%	6.3/sec	67.4			
/commentdisplay.do	2100	620	507	1024	226	5014	0.00%	6.3/sec	58.3			
/j_spring_security_logout	2100	321	231	626	15	4711	0.00%	6.3/sec	68.6			
TOTAL	39900	173	43	464	1	6250	0.00%	116.7/sec	11274.3			

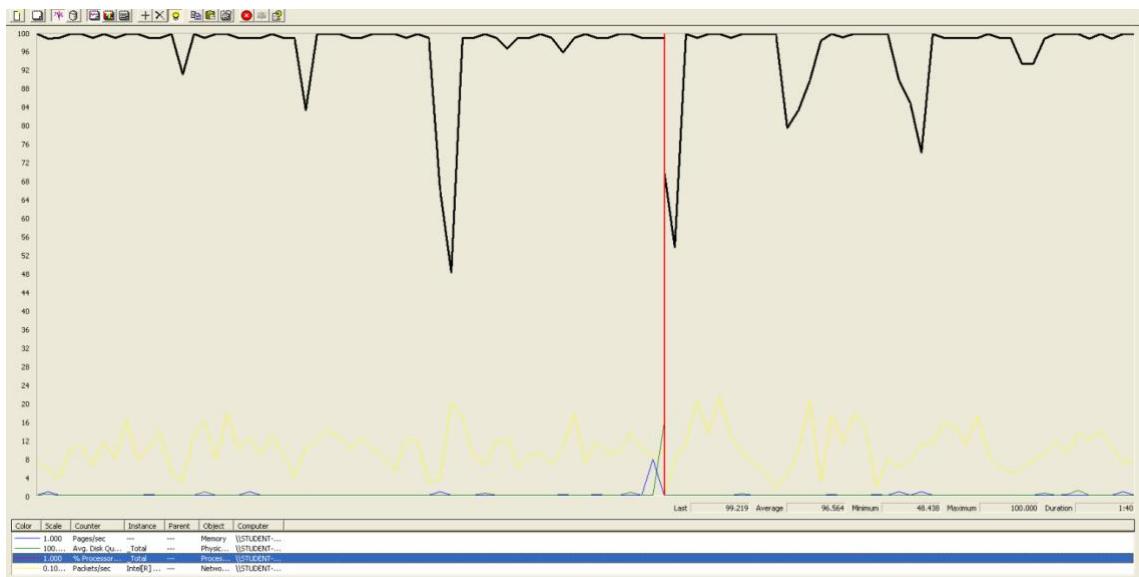
Como podemos observar fijándonos en 90% Line, las direcciones /j_spring_security_check, /comment/display.do generan un tiempo de respuesta (1134 y 1024 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 7.000 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



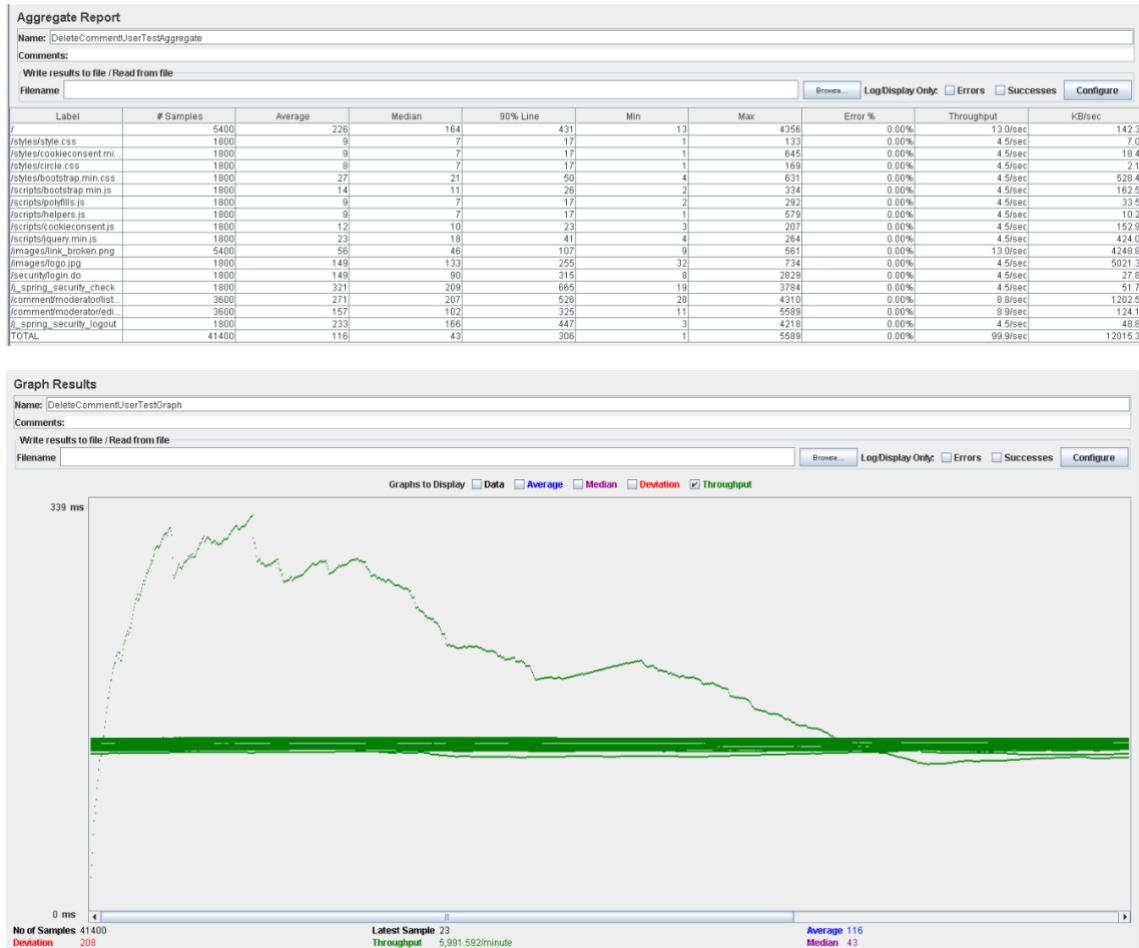
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 60 y 70.

Moderador borrar comentarios.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a mis comentarios y pulsamos en editar alguno de ellos, finalmente pulsamos en borrar.

Comenzamos probando con 60 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:



En esta tabla, el rendimiento es de aproximadamente 6.000 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 70 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name:	DeleteCommentUserTestAggregate									
Comments:										
Write results to file / Read from file										
Filename	Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/		6300	407	289	754	14	6770	0.00%	12.9/sec	141.1
/styles/style.css		2100	11	8	20	2	298	0.00%	4.4/sec	6.9
/scripts/comment.js		2100	111	8	201	2	700	0.00%	4.4/sec	18.2
/scripts/bootstrap.css		2100	11	8	21	2	221	0.00%	4.4/sec	2.1
/scripts/bootstrap.min.css		2100	35	29	64	5	328	0.00%	4.4/sec	522.4
/scripts/bootstrap.min.js		2100	18	14	32	3	463	0.00%	4.4/sec	160.6
/scripts/helpers.js		2100	11	9	20	2	513	0.00%	4.4/sec	33.2
/scripts/cookieconsent.js		2100	10	8	19	1	392	0.00%	4.4/sec	10.0
/scripts/jquery.min.js		2100	15	12	27	3	458	0.00%	4.4/sec	151.2
/scripts/jquery.cookie.png		6300	83	70	149	10	1100	0.00%	12.9/sec	419.1
/images/logo_broke.png		2100	29	24	53	4	393	0.00%	4.4/sec	4211.2
/securitylogin.do		2100	314	193	561	30	1501	0.00%	4.4/sec	409.4
/j_spring_security_check		2100	316	180	692	8	5748	0.00%	4.4/sec	27.5
/comment/moderatelist...		4200	716	477	1565	23	8023	0.00%	4.4/sec	51.2
/comment/moderateedi...		4200	468	362	885	30	6372	0.00%	8.7/sec	1187.6
/comment/moderateedi...		4200	327	214	643	11	7594	0.00%	8.8/sec	122.5
/j_spring_security_logout		2100	430	294	920	17	7568	0.00%	4.4/sec	48.1
TOTAL		48300	212	75	508	1	8023	0.00%	99.0/sec	11912.1

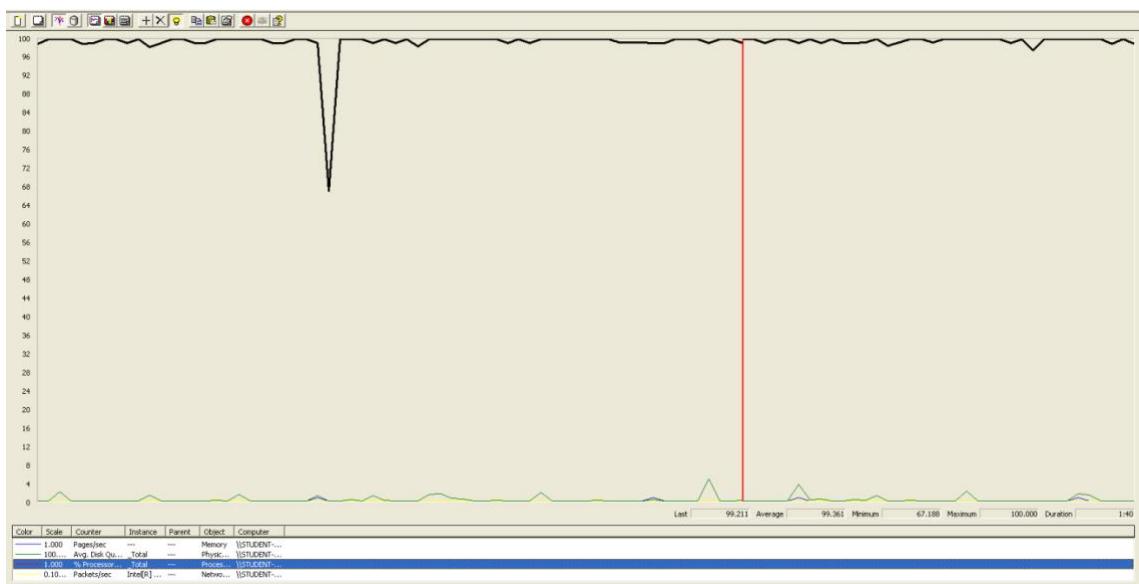
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check, genera un tiempo de respuesta (1565 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.900 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



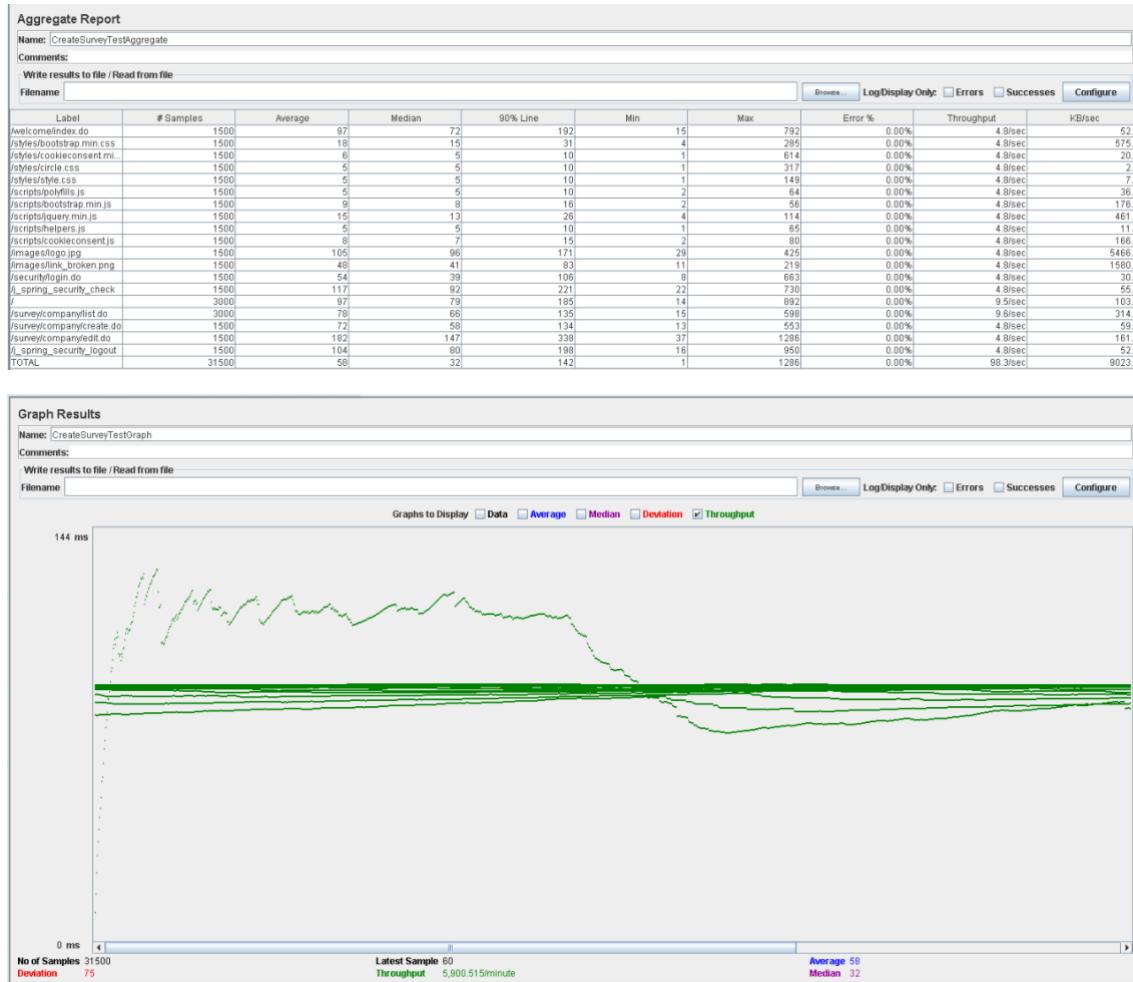
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 60 y 70.

Compañía/patrocinador/moderador crear encuesta.

En este caso de uso, accedemos a la página, nos logueamos como compañía, accedemos a las encuestas y pulsamos en crear encuesta.

Comenzamos probando con 50 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

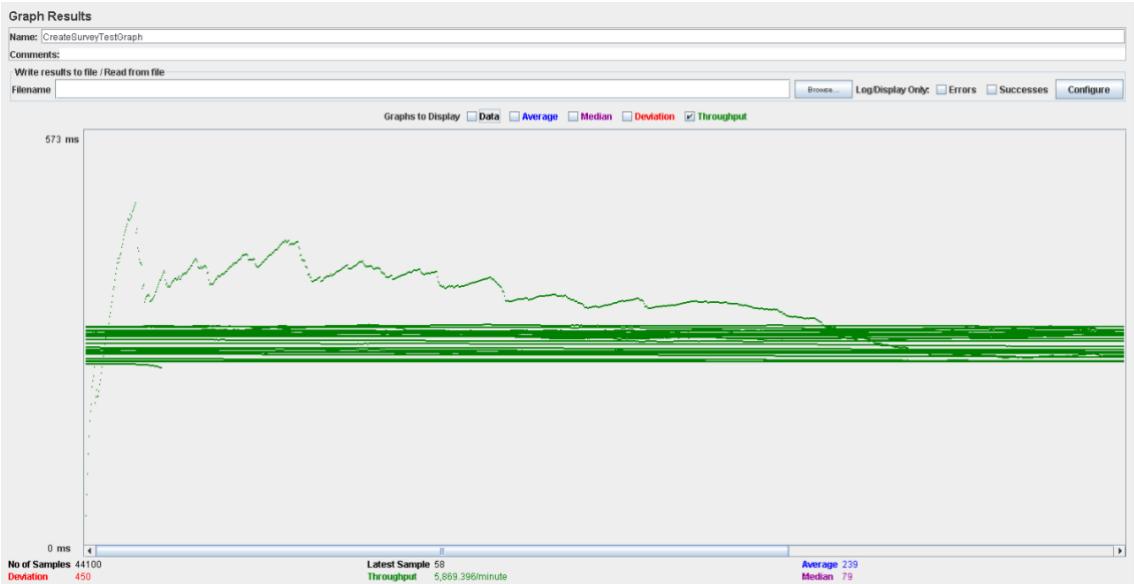


En esta tabla, el rendimiento es de aproximadamente 5.900 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 50 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name:	CreateSurveyTestAggregate									
Comments:										
Write results to file / Read from file										
Filename	Browse...	Log	Display Only	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input checked="" type="checkbox"/> Configure				
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/welcome/index.do	2100	395	263	768	15	4325	0.00%	4.8/sec	51.4	
/survey/bootstrap.min.css	2100	33	28	59	5	271	0.00%	4.8/sec	565.0	
/scripts/cookieconsent.js	2100	10	8	17	2	330	0.00%	4.8/sec	19.7	
/scripts/bootstrap.css	2100	9	8	17	2	300	0.00%	4.8/sec	2.3	
/scripts/bootstrap.min.js	2100	11	9	18	2	253	0.00%	4.8/sec	7.4	
/scripts/bootstrap.min.js	2100	10	8	17	2	411	0.00%	4.8/sec	35.9	
/scripts/query.min.js	2100	16	14	28	2	316	0.00%	4.8/sec	173.7	
/scripts/helpers.js	2100	26	23	49	5	291	0.00%	4.8/sec	453.3	
/scripts/helpers.js	2100	9	8	17	2	394	0.00%	4.8/sec	10.9	
/scripts/cookieconsent.js	2100	14	12	24	3	801	0.00%	4.8/sec	163.5	
/image/logo.png	2100	176	159	291	31	250	0.00%	4.8/sec	5369.1	
/image/logo-transparent.png	2100	171	143	243	11	250	0.00%	4.8/sec	159.4	
/securelogin.do	2100	302	178	603	8	5998	0.00%	4.8/sec	39.7	
/j_spring_security_check	2100	722	469	1530	21	9757	0.00%	4.8/sec	54.4	
/j_spring_security_check	4200	404	292	814	16	5356	0.00%	9.4/sec	102.7	
/survey/companylist.do	2100	366	244	814	14	6332	0.00%	4.8/sec	405.4	
/survey/company/create.do	2100	364	229	768	15	8759	0.00%	4.8/sec	58.4	
/survey/company/edit.do	2100	895	614	1622	36	8322	0.00%	4.8/sec	206.5	
/j_spring_security_logout	2100	419	296	777	12	6281	0.00%	4.8/sec	52.0	
TOTAL	44100	239	79	599	2	9757	0.00%	97.8/sec	9113.9	

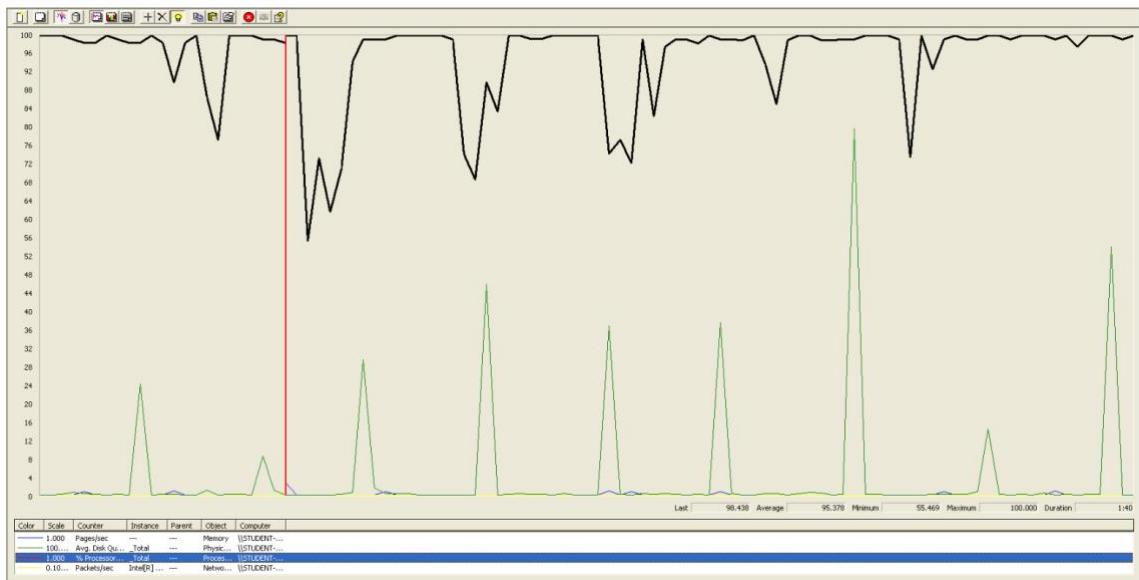
Como podemos observar fijándonos en 90% Line, las direcciones `/j_spring_security_check` y `/survey/company/edit.do` generan un tiempo de respuesta (1530 y 1622 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 5.800 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos `performance.exe` y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



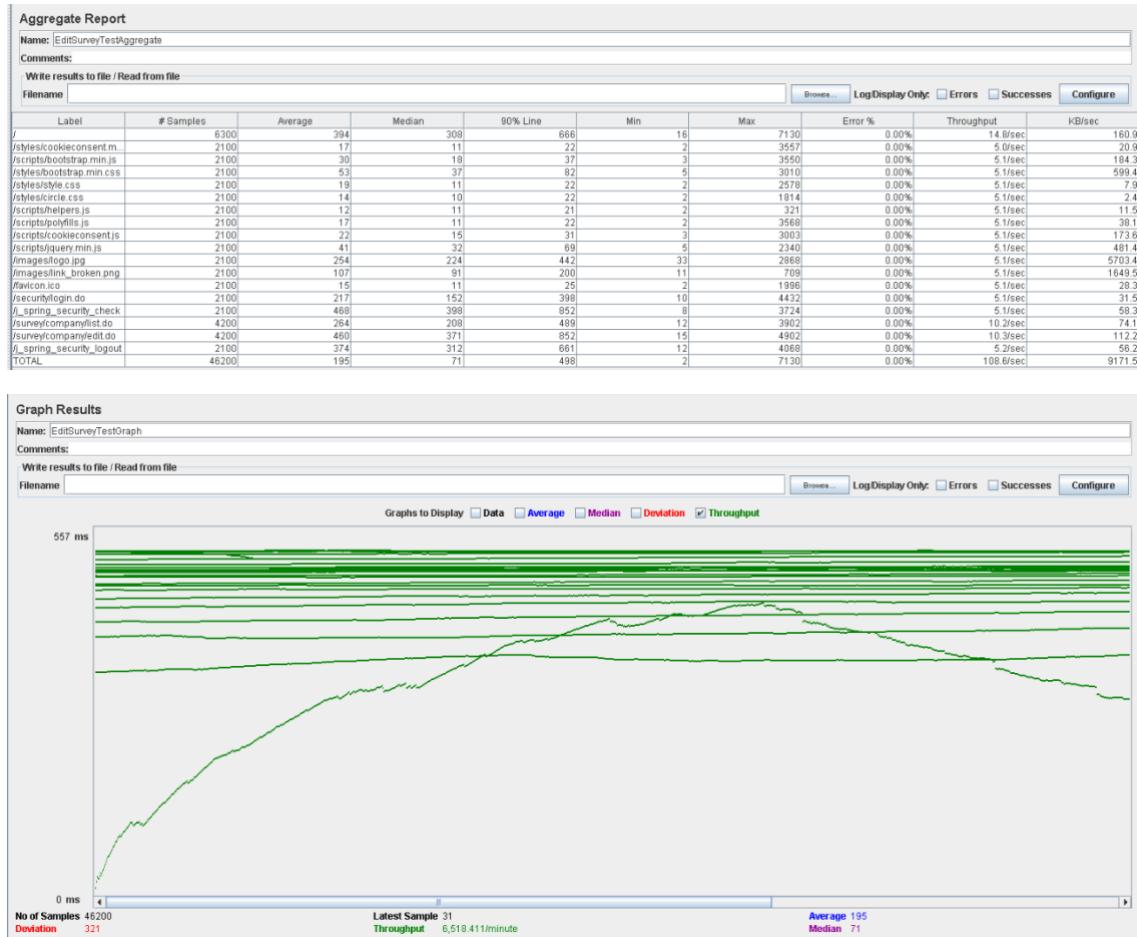
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 50 y 70.

Compañía/patrocinador/moderador editar encuesta.

En este caso de uso, accedemos a la página, nos logueamos como compañía, accedemos a las encuestas y pulsamos en editar alguna de ellas.

Comenzamos probando con 70 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

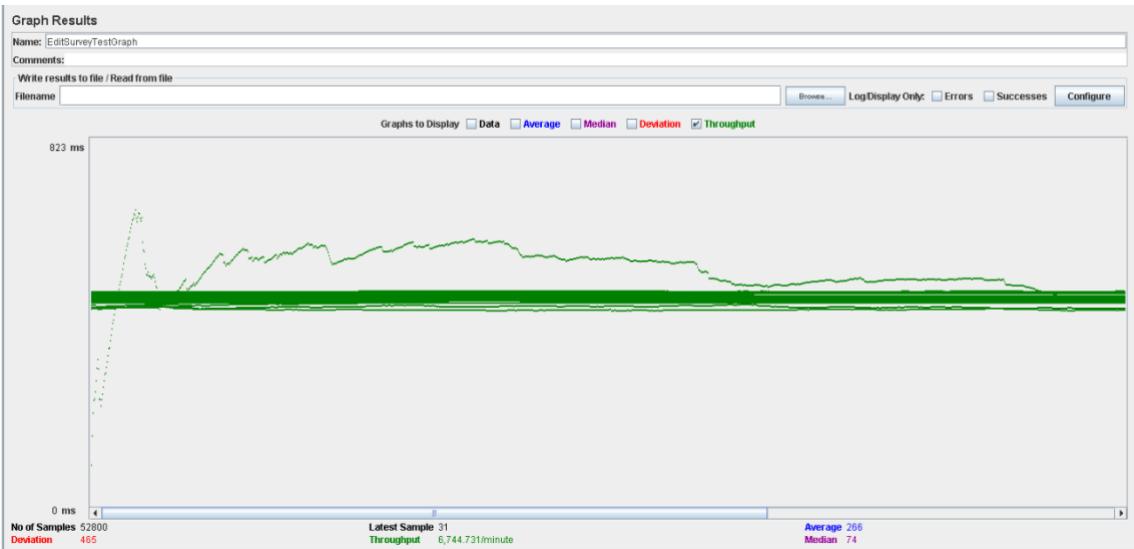


En esta tabla, el rendimiento es de aproximadamente 6.500 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 80 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
<input type="button" value="Browse..."/> <input type="checkbox"/> LogDisplay Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="checkbox"/> Configure										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/styles/bootstrap.css	7200	500	385	987	13	13902	0.00%	15.3/sec	166.5	
/styles/bootstrapconsent.m	2400	13	11	24	2	665	0.00%	5.2/sec	21.6	
/scripts/bootstrap.min.js	2400	22	18	41	3	385	0.00%	5.2/sec	190.2	
/scripts/bootstrap.min.css	2400	45	34	87	5	654	0.00%	5.2/sec	618.5	
/scripts/style.css	2400	13	11	24	2	764	0.00%	5.2/sec	8.1	
/scripts/circle.css	2400	13	10	23	1	894	0.00%	5.2/sec	2.4	
/scripts/bootstrap.js	2400	13	11	24	2	854	0.00%	5.2/sec	1.0	
/scripts/profiles.js	2400	13	11	24	2	176	0.00%	5.2/sec	39.2	
/scripts/cookieconsent.js	2400	20	16	36	3	626	0.00%	5.2/sec	179.0	
/scripts/jquery.min.js	2400	36	28	71	3	418	0.00%	5.2/sec	496.2	
/images/logo.jpg	2400	245	189	483	29	1233	0.00%	5.2/sec	5877.9	
/images/film_broken.png	2400	94	68	193	10	1273	0.00%	5.2/sec	1699.7	
/favicon.ico	2400	14	11	26	2	556	0.00%	5.2/sec	29.2	
/security/login.do	2400	382	240	937	8	8673	0.00%	5.2/sec	32.5	
/j_spring_security_check	2400	795	614	1490	22	10948	0.00%	5.2/sec	59.5	
/survey/companylist.do	4800	424	299	872	11	14600	0.00%	10.4/sec	75.3	
/survey/companyedit.do	4800	646	504	1257	15	9975	0.00%	10.4/sec	113.8	
/j_spring_security_logout	2400	494	391	905	17	10985	0.00%	5.2/sec	57.0	
TOTAL	52800	266	74	717	1	14900	0.00%	11.2/sec	9490.0	

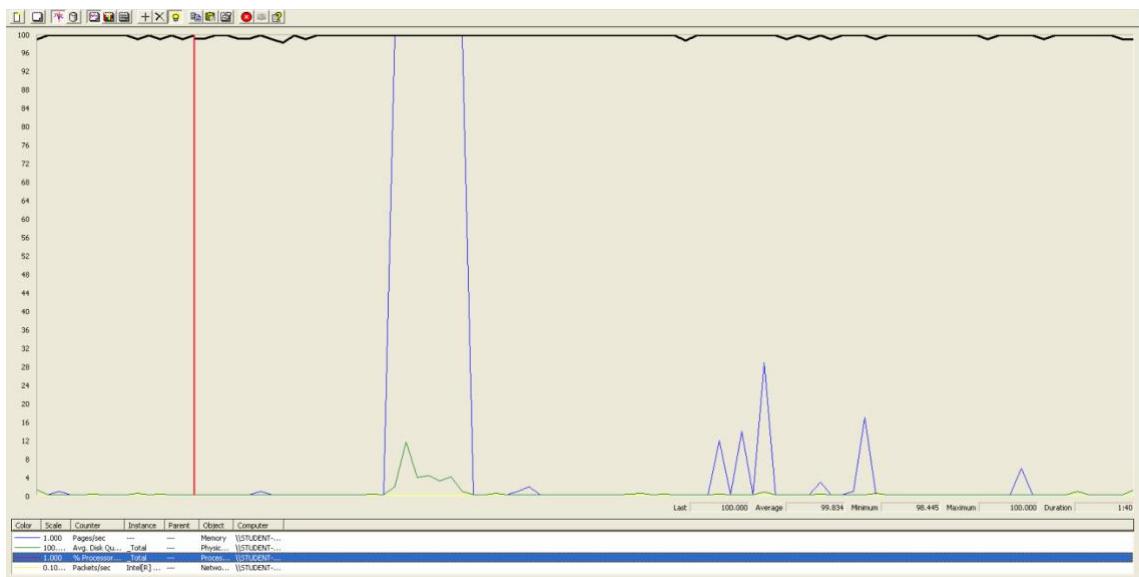
Como podemos observar fijándonos en 90% Line, las direcciones /j_spring_security_check y /survey/company/edit.do generan un tiempo de respuesta (1490 y 1257 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 6.700 ejecuciones por minuto (aumenta al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 70 y 80.

Compañía/patrocinador/moderador borrar encuesta.

En este caso de uso, accedemos a la página, nos logueamos como compañía, accedemos a las encuestas y pulsamos en editar alguna de ellas, luego a borrar.

Comenzamos probando con 70 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

Aggregate Report										
Name: DeleteSurveyTestAggregate										
Comments:										
Write results to file / Read from file										
Filename:										
Browse										
Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes Configure										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	iOPS/sec	KB/sec
!survey!bootstrap_min.css	6300	274	148	475	16	4390	0.00%	17.4sec	189	6.0/sec
!survey!bootstrap.css	2100	26	21	50	5	242	0.00%	6.0/sec	709	6.0/sec
!styles!cookieconsent.mli	2100	8	6	15	1	598	0.00%	6.0/sec	9	6.0/sec
!styles!circle.css	2100	7	6	14	1	127	0.00%	6.0/sec	24	6.0/sec
!scripts!bootstrap_min.js	2100	13	10	24	2	429	0.00%	6.0/sec	2	6.0/sec
!scripts!bootstrap.js	2100	22	17	43	4	425	0.00%	6.0/sec	569	6.0/sec
!scripts!headers.js	2100	8	6	14	1	524	0.00%	6.0/sec	13	6.0/sec
!scripts!jonyMills.js	2100	8	6	15	1	305	0.00%	6.0/sec	45	6.0/sec
!scripts!fogic_cookie.js	2100	12	10	23	3	264	0.00%	6.0/sec	205	6.0/sec
!image!fogo.jpg	2100	146	127	253	36	703	0.00%	6.0/sec	6739	6.0/sec
!image!fogic_broken.png	2100	69	56	135	11	489	0.00%	6.0/sec	1947	6.0/sec
!feedback!do	2100	153	81	332	9	5607	0.00%	6.0/sec	37	6.0/sec
!spring_security_check	2100	325	200	772	17	3742	0.00%	6.0/sec	68	6.0/sec
?survey!companylist.do	4200	187	106	401	13	4157	0.00%	11.9sec	85	6.0/sec
?survey!companyedit.do	4200	177	101	367	11	4173	0.00%	11.9sec	169	6.0/sec
?_spring_security_logout	2100	230	150	479	15	3810	0.00%	6.0/sec	65	6.0/sec
TOTAL	44100	116	40	305	1	4390	0.00%	121.9sec	10777.	6.0/sec



En esta tabla, el rendimiento es de aproximadamente 7.300 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 80 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report											
Name: DeleteSurveyTestAggregate											
Comments:											
Write results to file / Read from file											
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec		
/	7200	329	265	699	18	4216	0.00%	17.7/sec	191.7		
/styles/bootstrap.min.css	2400	32	26	55	5	1006	0.00%	6.0/sec	715.6		
/styles/style.css	2400	10	7	16	1	640	0.00%	6.0/sec	9.4		
/styles/cookieconsent.mi...	2400	9	7	15	1	606	0.00%	6.0/sec	24.9		
/styles/circle.css	2400	9	7	15	1	702	0.00%	6.0/sec	2.9		
/scripts/jquery.min.js	2400	14	13	25	3	411	0.00%	6.0/sec	20.0		
/scripts/helpers.js	2400	25	21	49	4	456	0.00%	6.0/sec	574.1		
/scripts/pollfill.js	2400	9	7	15	1	497	0.00%	6.0/sec	13.8		
/scripts/cookieconsent.js	2400	9	8	16	2	240	0.00%	6.0/sec	45.4		
/images/logo.jpg	2400	14	12	23	3	603	0.00%	6.0/sec	207.1		
/images/link_broken.png	2400	79	67	144	13	1042	0.00%	6.0/sec	1966.4		
/authn/login.do	2400	263	163	602	10	3632	0.00%	6.0/sec	37.6		
/j_spring_security_check	2400	501	437	1164	27	697	0.00%	6.0/sec	69.6		
/survey/companylist.do	4800	284	204	696	13	4010	0.00%	12.0/sec	86.9		
/survey/companyedit.do	4800	280	196	669	13	3764	0.00%	12.1/sec	172.0		
/j_spring_security_logout	2400	335	257	638	22	3295	0.00%	6.0/sec	65.9		
TOTAL	50400	176	76	435	1	6912	0.00%	123.6/sec	10934.1		

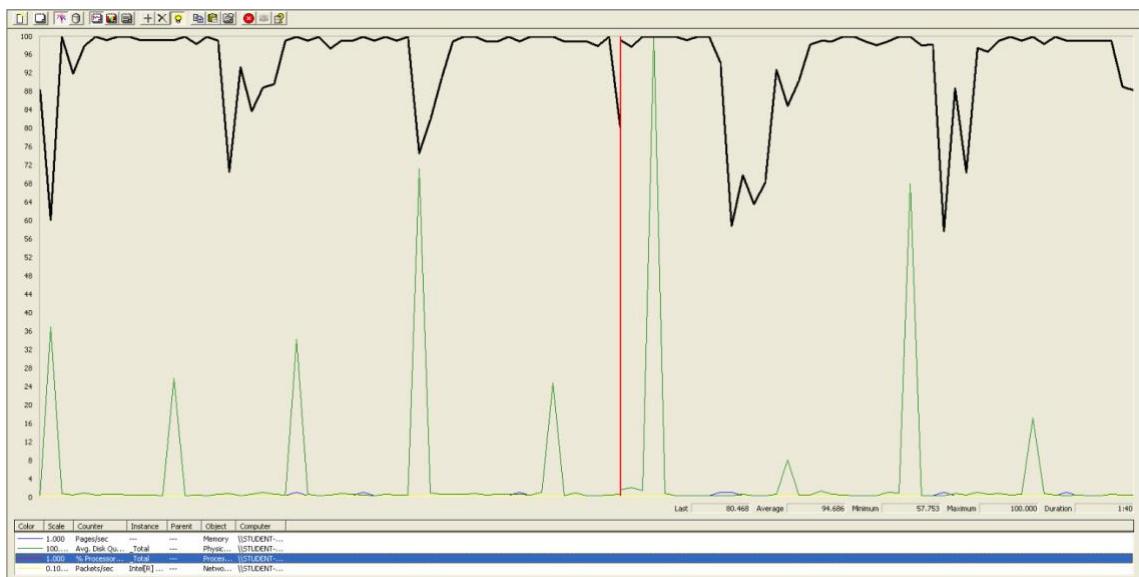
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check, genera un tiempo de respuesta (1164 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 7.400 ejecuciones por minuto (aumenta al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



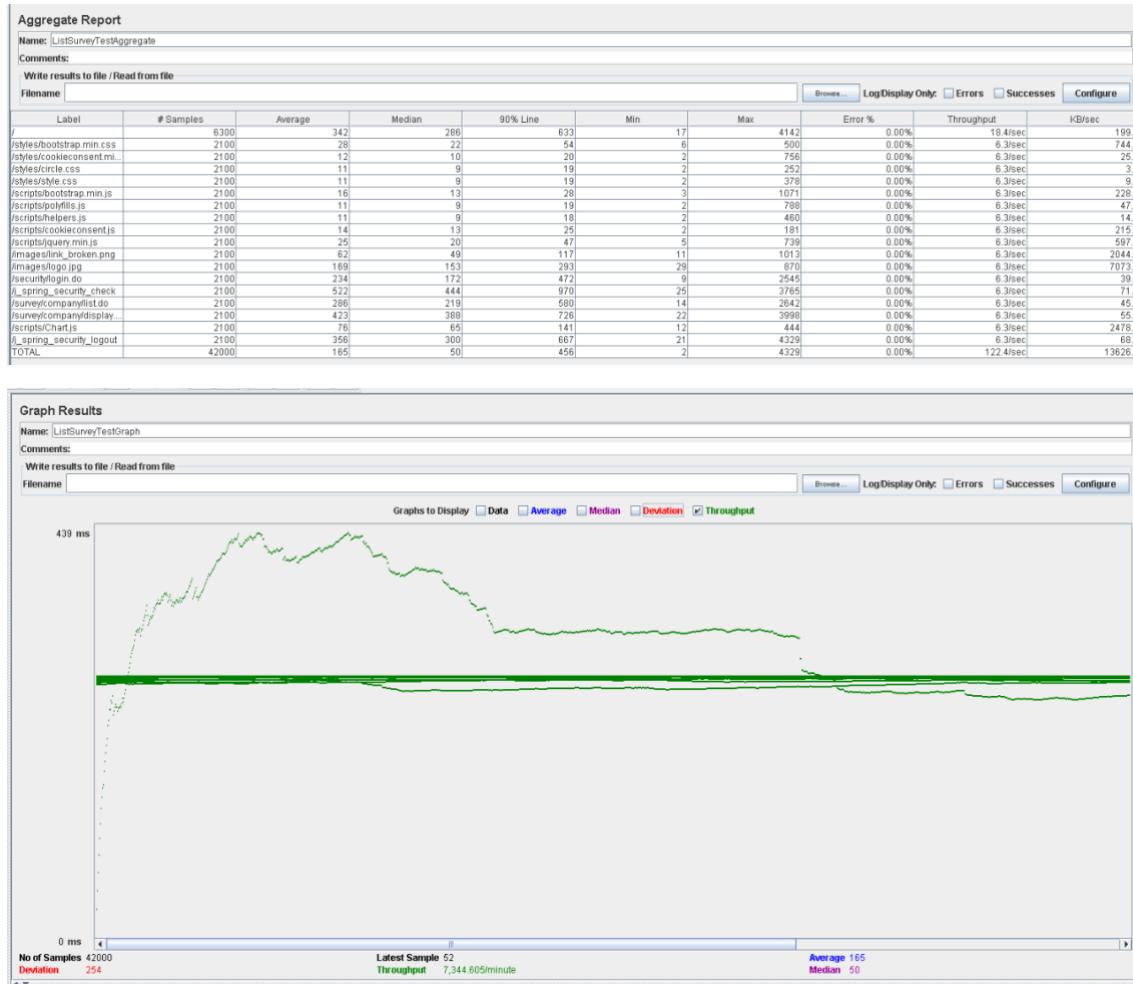
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 70 y 80.

Compañía/patrocinador/moderador listar mis encuestas y desplegar informe

En este caso de uso, accedemos a la página, nos logueamos como compañía, accedemos a las encuestas y pulsamos en desplegar alguna de ellas.

Comenzamos probando con 70 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:



En esta tabla, el rendimiento es de aproximadamente 7.300 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 80 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report															
Name: ListSurveyTestAggregate															
Comments:															
Write results to file / Read from file											Browse... Log Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input checked="" type="checkbox"/> Configure				
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput		KB/sec					
/	7200	289	221	534	17	4224	0.00%	21.6/sec		234.9					
/styles/bootstrap.min.css	2400	26	21	46	5	388	0.00%	7.4/sec		881.9					
/styles/cookieconsent.mi...	2400	8	6	15	1	425	0.00%	7.4/sec		30.7					
/styles/circle.css	2400	8	6	14	1	363	0.00%	7.4/sec		3.6					
/scripts/trap_min.js	2400	9	6	14	1	329	0.00%	7.4/sec		1.1					
/scripts/polyfill.js	2400	12	10	23	3	238	0.00%	7.4/sec		271.2					
/scripts/helpers.js	2400	8	7	15	2	256	0.00%	7.4/sec		56.0					
/scripts/cookieconsent.js	2400	11	10	20	3	128	0.00%	7.4/sec		255.2					
/scripts/jquery.min.js	2400	21	18	39	5	263	0.00%	7.4/sec		707.6					
/imageLink_broken.png	2400	55	46	101	9	443	0.00%	7.4/sec		2423.3					
/images/logo.jpg	2400	134	243	301	981	0.00%	7.4/sec		8349.6						
/scripts/submit.js	2400	219	135	405	3	406	0.00%	7.4/sec		4.2					
/j_spring_security_check	2400	503	360	1049	26	6077	0.00%	7.4/sec		84.5					
/survey/companyList.do	2400	265	179	510	15	5047	0.00%	7.4/sec		53.6					
/survey/companyDisplay	2400	360	299	629	24	2240	0.00%	7.4/sec		65.0					
/scripts/Charts.js	2400	73	63	128	12	437	0.00%	7.4/sec		2920.9					
/j_spring_security_logout	2400	295	220	550	10	3364	0.00%	7.4/sec		80.7					
TOTAL	48000	145	48	367	1	5077	0.00%	144.1/sec		16045.3					

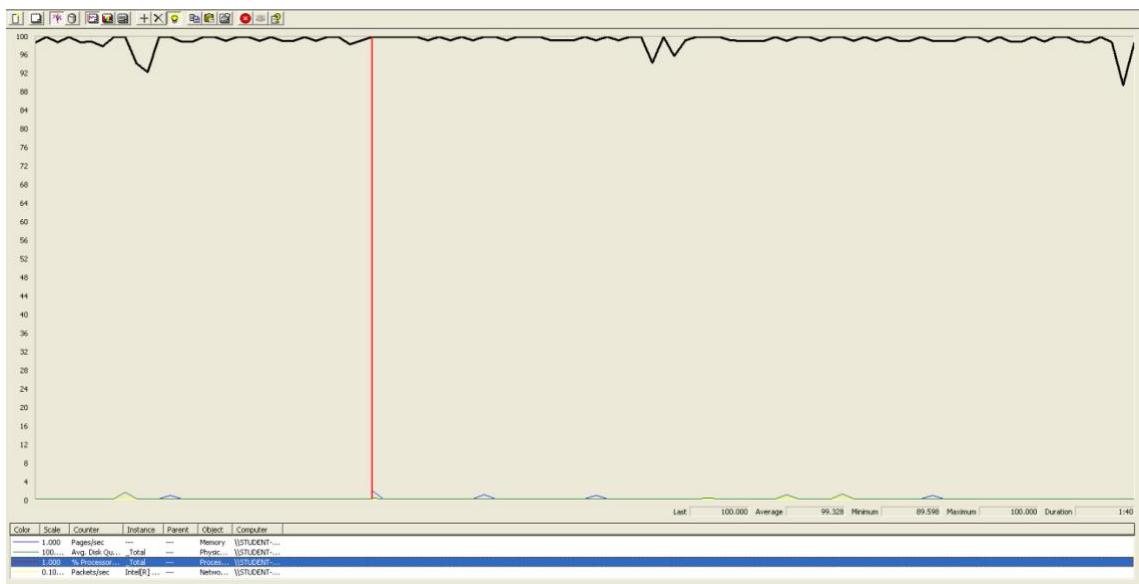
Como podemos observar fijandonos en 90% Line, la dirección /j_spring_security_check, genera un tiempo de respuesta (1049 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 8.600 ejecuciones por minuto (aumenta al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



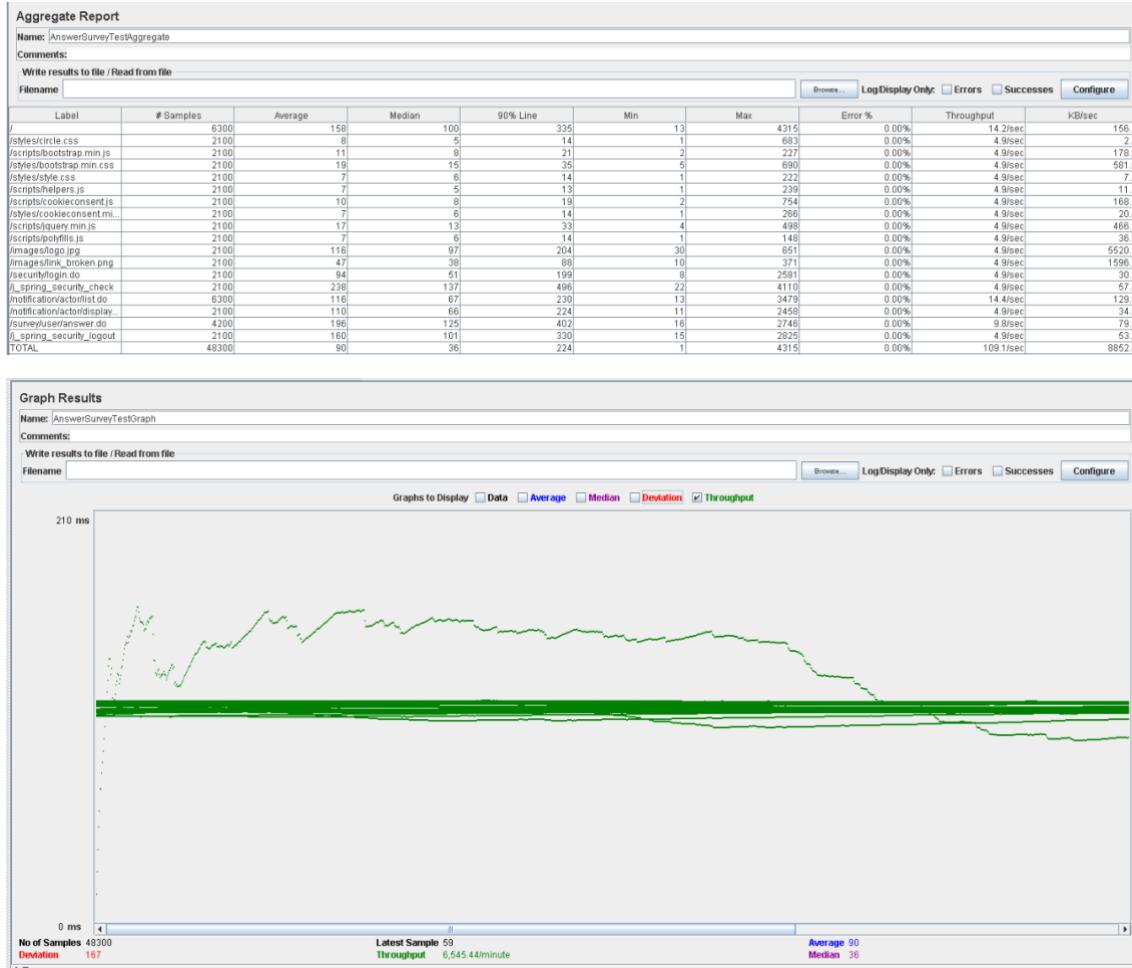
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 70 y 80.

Usuario/patrocinador responder encuesta.

En este caso de uso, accedemos a la página, nos logueamos como usuario, accedemos a las notificaciones recibidas y pulsamos en la de encuesta para acceder al formulario.

Comenzamos probando con 70 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

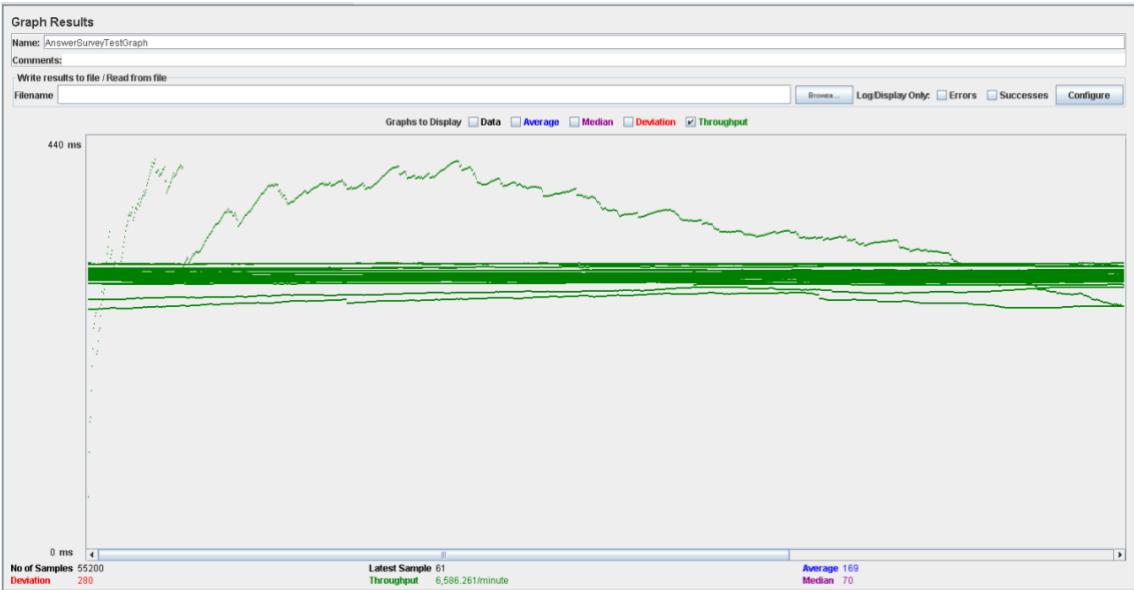


En esta tabla, el rendimiento es de aproximadamente 6.500 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 80 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: AnswerSurveyTestAggregate										
Comments:										
Write results to file / Read from file										
Filename:	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	7200	294	221	669	16	5091	0.00%	14.3/sec	157.1	
/styles/circle.css	2400	9	7	16	1	178	0.00%	4.9/sec	2.3	
/scripts/cap.min.js	2400	14	12	25	3	119	0.00%	4.9/sec	178.9	
/scripts/bootstrap.min.css	2400	26	21	47	5	390	0.00%	4.9/sec	50.0	
/scripts/table.css	2400	9	7	16	1	346	0.00%	4.9/sec	7.7	
/scripts/shablon.js	2400	9	7	16	1	302	0.00%	4.9/sec	11.2	
/scripts/cookieconsent.js	2400	13	11	23	2	320	0.00%	4.9/sec	168.3	
/styles/cookieconsent.mi...	2400	9	7	16	1	462	0.00%	4.9/sec	20.3	
/scripts/jquery.min.js	2400	22	18	41	4	792	0.00%	4.9/sec	466.8	
/scripts/polffills.js	2400	9	7	17	2	523	0.00%	4.9/sec	36.9	
/images/logo.jpg	2400	152	140	250	29	764	0.00%	4.9/sec	6529.9	
/scripts/login.png	2400	57	56	123	10	279	0.00%	4.9/sec	1550.4	
/secunlogin.do	2400	212	129	449	8	3155	0.00%	4.9/sec	30.6	
/j_spring_security_check	2400	498	357	1080	26	3936	0.00%	4.9/sec	57.6	
/notificationactionlist.do	7200	233	160	458	13	6363	0.00%	14.5/sec	130.6	
/notificationactiondisplay...	2400	229	154	479	15	3180	0.00%	4.9/sec	34.2	
/surveyuseranswer.do	4800	368	268	728	16	5451	0.00%	9.8/sec	79.0	
/j_spring_security_logout	2400	292	219	643	22	5003	0.00%	4.9/sec	53.5	
TOTAL	55200	169	70	420	1	6363	0.00%	109.8/sec	8905.8	

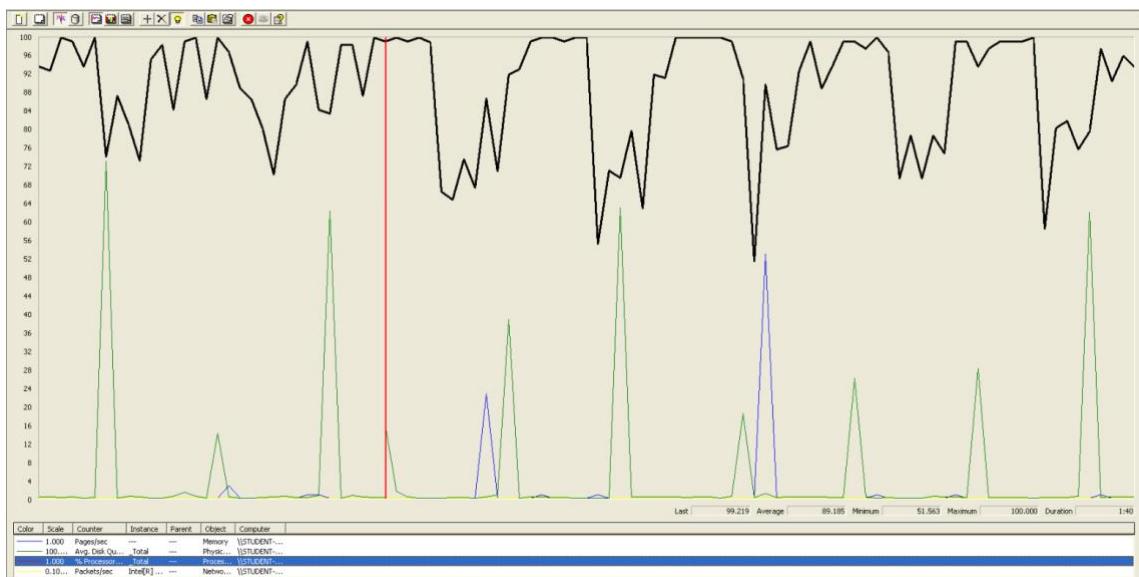
Como podemos observar fijándonos en 90% Line, la dirección /j_spring_security_check, genera un tiempo de respuesta (1080 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 6.600 ejecuciones por minuto (aumenta al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



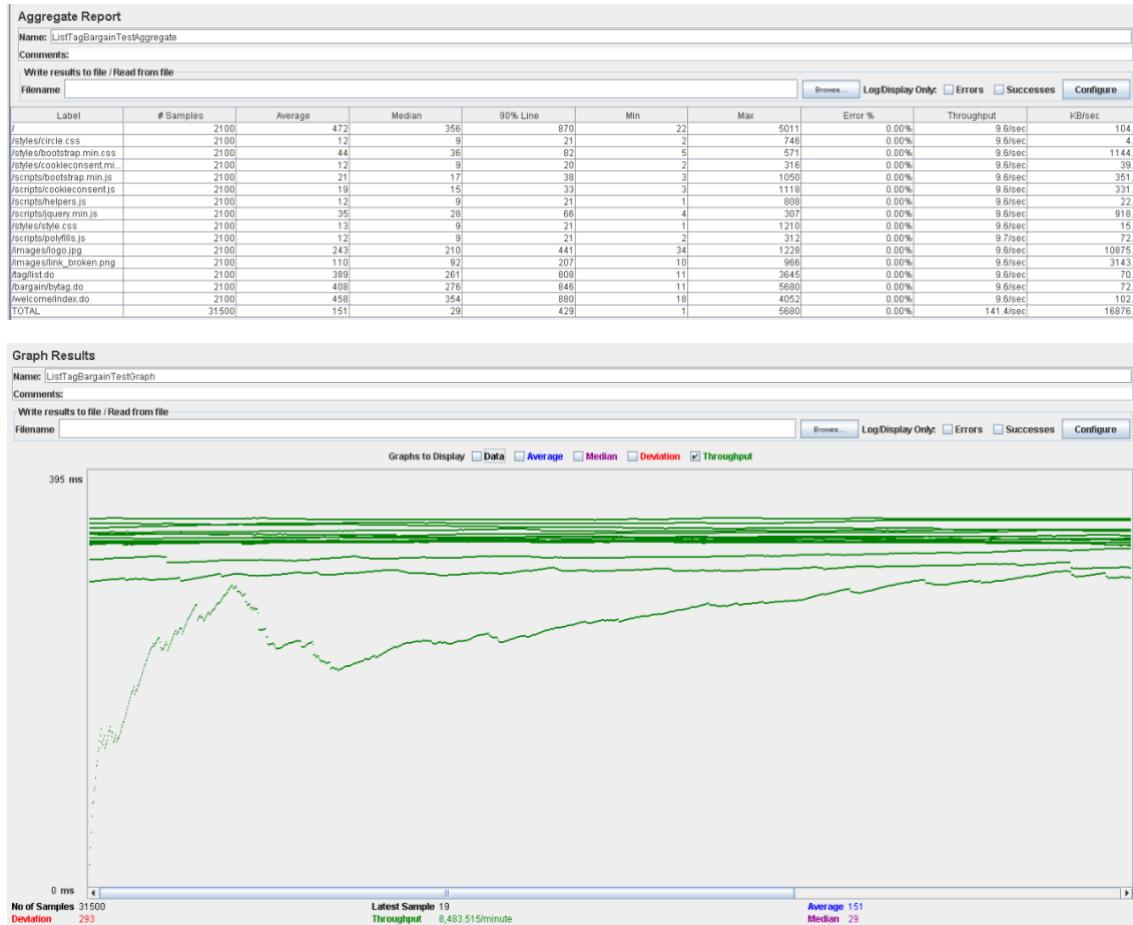
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 70 y 80.

Listar los chollos de una etiqueta.

En este caso de uso, accedemos a la página, accedemos al listado de etiquetas y de ahí a los chollos de alguna de ellas.

Comenzamos probando con 70 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos de espera demasiado altos:

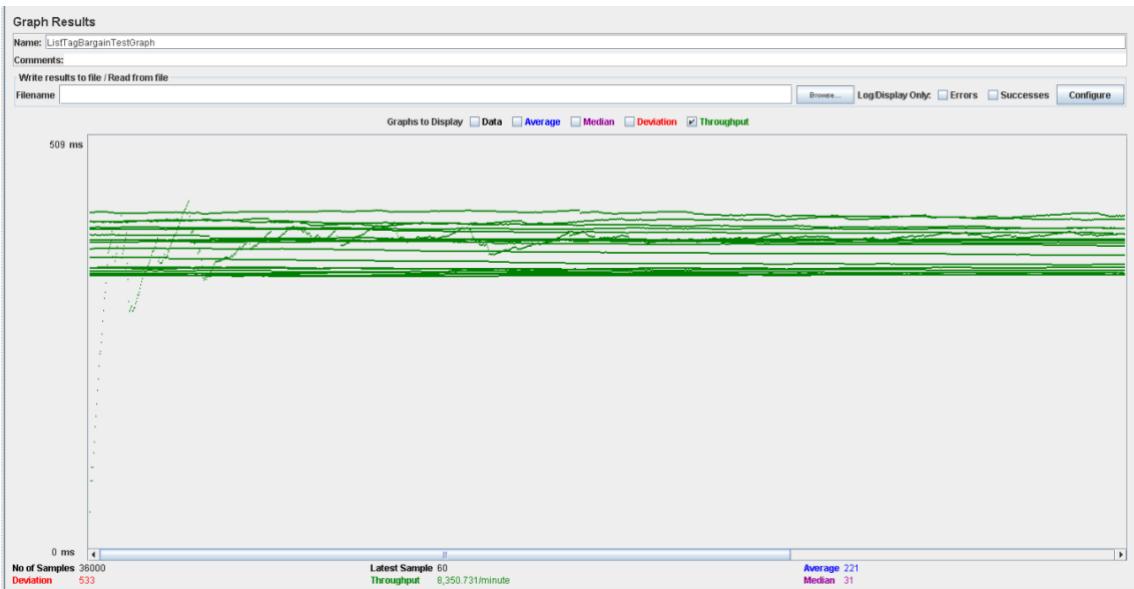


En esta tabla, el rendimiento es de aproximadamente 8.500 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

La configuración en esta ocasión es de 80 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Aggregate Report										
Name: ListTagBargainTestAggregate Comments: Write results to file / Read from file Filename:										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/styles/circle.css	2400	719	470	1538	15	8664	0.00%	9.4/sec	101.9	
/scripts/bootstrap.min.js	2400	13	10	23	1	298	0.00%	9.4/sec	4.5	
/scripts/bootstrap.min.js.map	2400	47	36	87	5	1077	0.00%	9.4/sec	110.9	
/scripts/cookieconsent.min.js	2400	14	10	22	2	487	0.00%	9.4/sec	30.1	
/scripts/bootstrapstrap.min.js	2400	22	17	39	3	621	0.00%	9.4/sec	344.3	
/scripts/helpers.js	2400	20	16	37	5	560	0.00%	9.4/sec	324.3	
/scripts/jquery.min.js	2400	13	10	22	2	398	0.00%	9.4/sec	21.6	
/styles/style.css	2400	13	10	23	2	603	0.00%	9.4/sec	14.8	
/scripts/polyfills.js	2400	13	10	24	2	558	0.00%	9.4/sec	71.1	
/image/logo.jpg	2400	255	219	469	29	1374	0.00%	9.4/sec	10048.5	
/image/logo_broken.png	2400	119	96	235	1	244	0.00%	9.4/sec	30.7	
/taglist.do	2400	627	366	1561	12	9054	0.00%	9.4/sec	69.0	
/bargain/bytag.do	2400	641	369	1433	14	9082	0.00%	9.5/sec	71.1	
/welcome/index.do	2400	735	459	1654	14	10184	0.00%	9.5/sec	100.4	
TOTAL	36000	221	31	565	1	10184	0.00%	139.2/sec	16812.6	

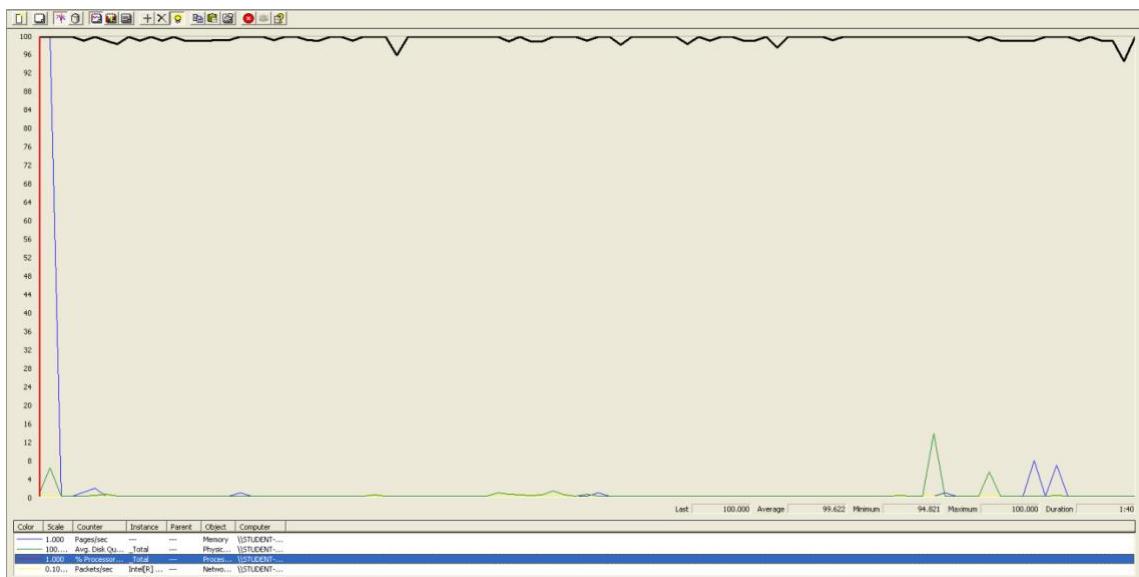
Como podemos observar fijándonos en 90% Line, las direcciones /tag/list.do, /bargain/bytag.do y /welcome/index.do generan un tiempo de respuesta (1561, 1433 y 1654 ms respectivamente) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 8.300 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



Performance con Processor Time.

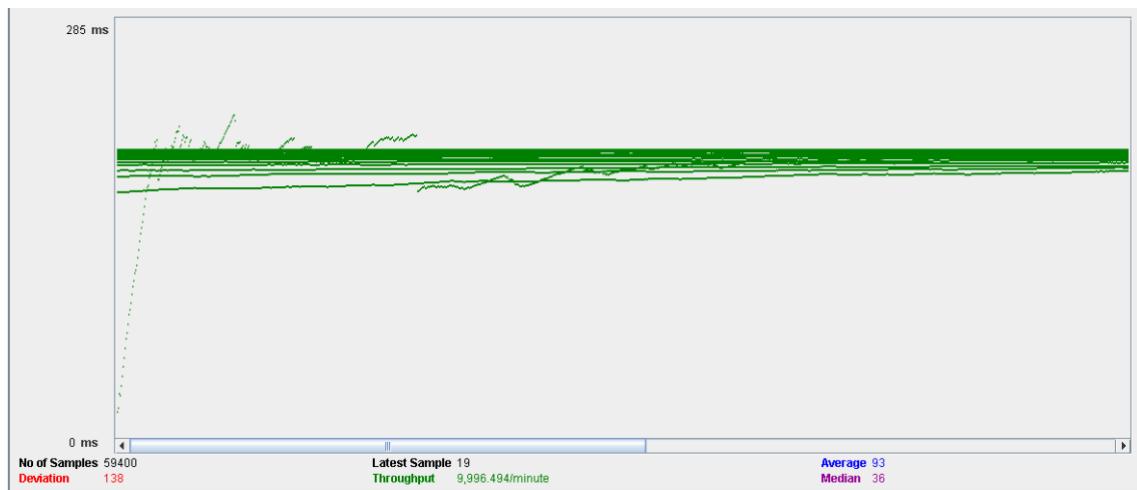
Por lo que, en este caso de uso, los usuarios permitidos serán entre 70 y 80.

Caso de uso 060: Editar plan de pago

En este caso de uso, accedemos a la página, nos autenticamos como administrador, accedemos a la vista de los planes y editamos uno.

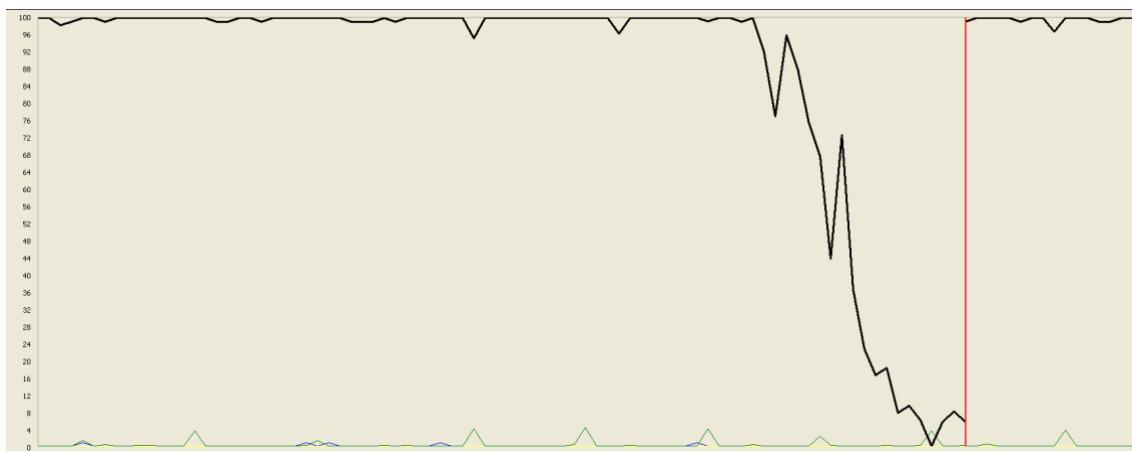
Comenzamos probando con 90 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que lleguen a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	172	144	314	12	2402	0.00%	22.7/sec	245.9
/scripts/jquery.min.js	2700	18	15	32	4	317	0.00%	7.8/sec	739.3
/styles/cookiecons...	2700	7	6	13	1	198	0.00%	7.8/sec	32.1
/styles/style.css	2700	7	6	13	1	538	0.00%	7.8/sec	12.1
/styles/circle.css	2700	7	6	12	1	197	0.00%	7.8/sec	3.7
/scripts/bootstrap.m...	2700	22	18	40	4	306	0.00%	7.8/sec	921.6
/scripts/bootstrap....	2700	11	9	20	3	276	0.00%	7.8/sec	283.4
/scripts/polyfills.js	2700	8	6	13	1	302	0.00%	7.8/sec	58.5
/scripts/helpers.js	2700	7	6	12	1	278	0.00%	7.8/sec	17.7
/scripts/cookiecon...	2700	11	9	20	2	298	0.00%	7.8/sec	266.6
/images/logo.jpg	2700	131	126	222	26	457	0.00%	7.8/sec	8757.3
/favicon.ico	2700	8	7	15	1	128	0.00%	7.8/sec	43.5
/security/login.do	2700	119	86	232	7	2304	0.00%	7.8/sec	48.5
/j_spring_security...	2700	257	210	486	19	2220	0.00%	7.8/sec	89.1
/images/link_broke...	2700	49	41	91	9	268	0.00%	7.8/sec	2537.1
/plan/display.do	5400	136	104	256	10	3317	0.00%	15.4/sec	137.1
/plan/administrator...	5400	208	161	416	11	3174	0.00%	15.6/sec	134.4
/j_spring_security...	2700	178	152	323	15	2758	0.00%	7.8/sec	85.0
TOTAL	59400	93	36	245	1	3317	0.00%	166.6/sec	14054.5



En esta tabla, el rendimiento es de 9996 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

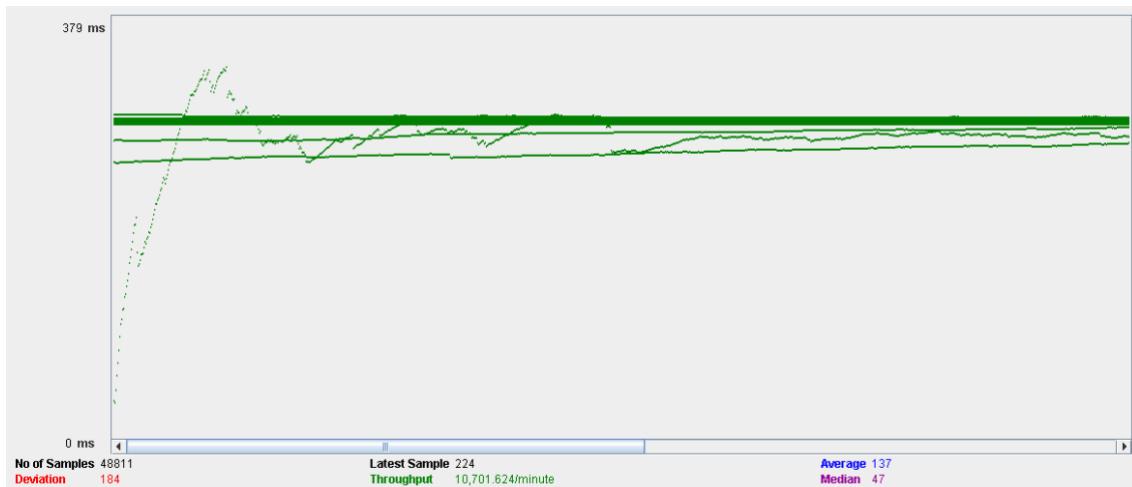
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso del procesador es bastante elevado aún dando unos tiempos adecuados.



La configuración en esta ocasión es de 100 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose errores, y unos tiempos que se encuentran en el límite establecido es cual ronda los 1.2 segundos, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7390	260	225	508	12	1960	0.04%	24.1/sec	261.3
/scripts/jquery.min.js	2506	28	22	53	3	527	0.00%	8.2/sec	779.9
/styles/cookiecons...	2506	11	9	21	1	216	0.00%	8.2/sec	33.9
/styles/style.css	2506	11	9	20	1	535	0.00%	8.2/sec	12.8
/styles/circle.css	2506	11	9	21	1	577	0.00%	8.2/sec	3.9
/styles/bootstrap.m...	2506	32	26	80	4	380	0.00%	8.2/sec	972.0
/scripts/bootstrap...	2506	17	14	31	2	459	0.00%	8.2/sec	298.8
/scripts/polyfills.js	2506	12	9	21	1	835	0.00%	8.2/sec	61.7
/scripts/helpers.js	2506	11	9	21	1	631	0.00%	8.2/sec	18.7
/scripts/cookiescon...	2506	15	13	28	2	239	0.00%	8.2/sec	281.2
/images/logo.jpg	2506	189	166	364	25	826	0.00%	8.2/sec	9234.8
/favicon.ico	2506	12	10	24	1	168	0.00%	8.2/sec	46.0
/security/logind...	2488	170	137	350	7	1489	0.00%	8.2/sec	51.2
/j_security_check...	2473	372	339	697	10	1704	0.00%	8.2/sec	93.6
/images/link_broke...	2471	72	60	145	8	1241	0.00%	8.2/sec	2674.8
/plan/display.do	4894	204	170	397	9	1923	0.04%	16.3/sec	144.6
/plan/administrator...	4867	302	249	616	9	1689	0.00%	16.4/sec	140.9
/j_spring_security...	2412	268	234	507	7	1568	0.09%	8.2/sec	88.9
TOTAL	54551	136	47	391	1	1960	0.01%	178.0/sec	15120.9

Como podemos observar fijándonos en 90% Line, el grupo formado por la autenticación inicial y la vuelta a la pantalla de inicio genera un tiempo de respuesta de alrededor de 1200 ms que podría aceptarse, pero en este caso hay errores

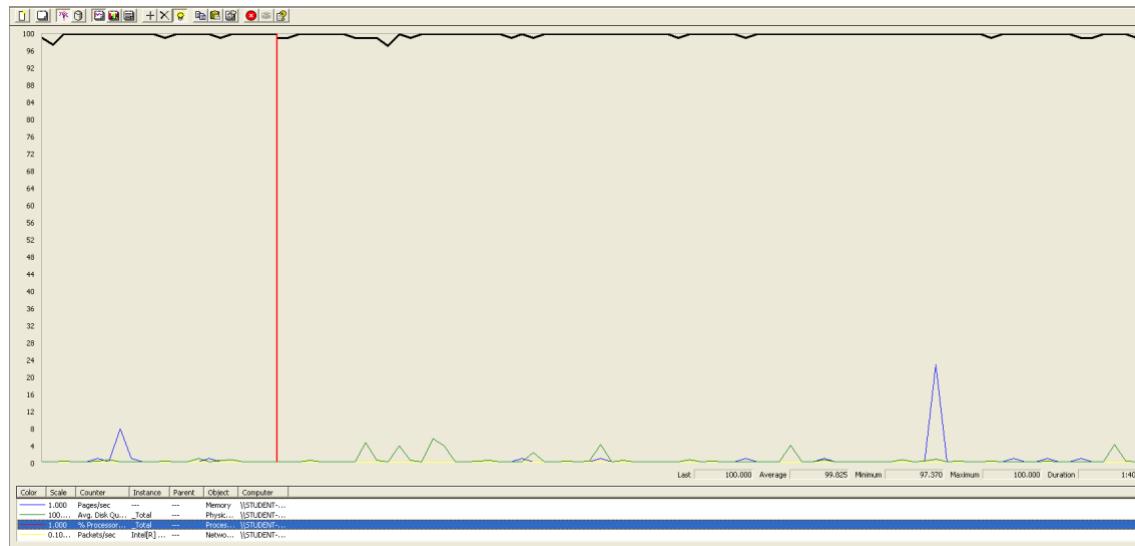


En esta tabla, el rendimiento es de aproximadamente 10701 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los

hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



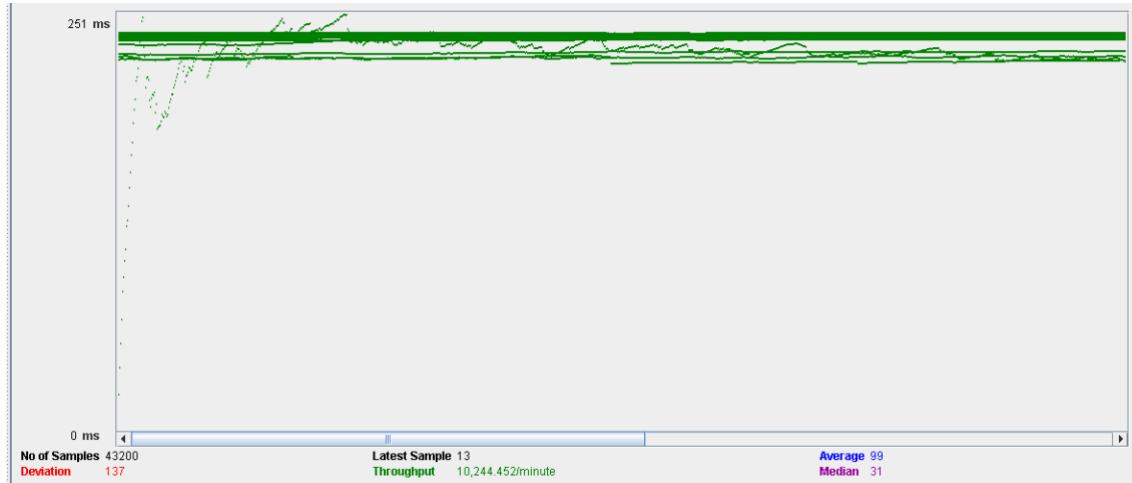
Por lo que, en este caso de uso, los usuarios permitidos serán entre 90 y 100.

Caso de uso 061: Desplegar planes de pago

En este caso de uso, accedemos a la página, nos autenticamos como administrador, y accedemos a la vista de los planes.

Comenzamos probando con 80 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que lleguen a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	213	190	368	13	1985	0.00%	28.5/sec	308.0
/styles/bootstrap.m...	2400	24	21	41	5	266	0.00%	9.7/sec	1154.9
/scripts/jquery.min.js	2400	19	17	33	5	417	0.00%	9.7/sec	926.5
/scripts/cookiecons...	2400	8	6	13	1	224	0.00%	9.7/sec	40.2
/styles/style.css	2400	8	6	12	1	582	0.00%	9.7/sec	15.2
/styles/circle.css	2400	7	6	13	1	180	0.00%	9.7/sec	4.6
/scripts/bootstrap...	2400	12	10	21	2	173	0.00%	9.7/sec	355.1
/scripts/polyfills.j...	2400	8	6	13	2	509	0.00%	9.7/sec	73.3
/scripts/helpers.js	2400	8	6	12	1	281	0.00%	9.7/sec	22.2
/scripts/cookiecon...	2400	11	9	19	2	385	0.00%	9.7/sec	334.2
/images/logo.jpg	2400	147	143	248	28	1229	0.00%	9.7/sec	10973.7
/images/link_broke...	2400	56	50	100	9	378	0.00%	9.7/sec	3172.0
/security/login.do	2400	149	116	287	7	1994	0.00%	9.7/sec	60.7
/j_spring_security_...	2400	312	276	547	18	1572	0.00%	9.7/sec	111.4
/plan/display.do	2400	165	137	301	9	1730	0.00%	9.7/sec	86.5
/j_spring_security_...	2400	210	188	358	16	1884	0.00%	9.7/sec	106.2
TOTAL	43200	99	31	272	1	1994	0.00%	170.7/sec	17303.2



En esta tabla, el rendimiento es de 10244 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

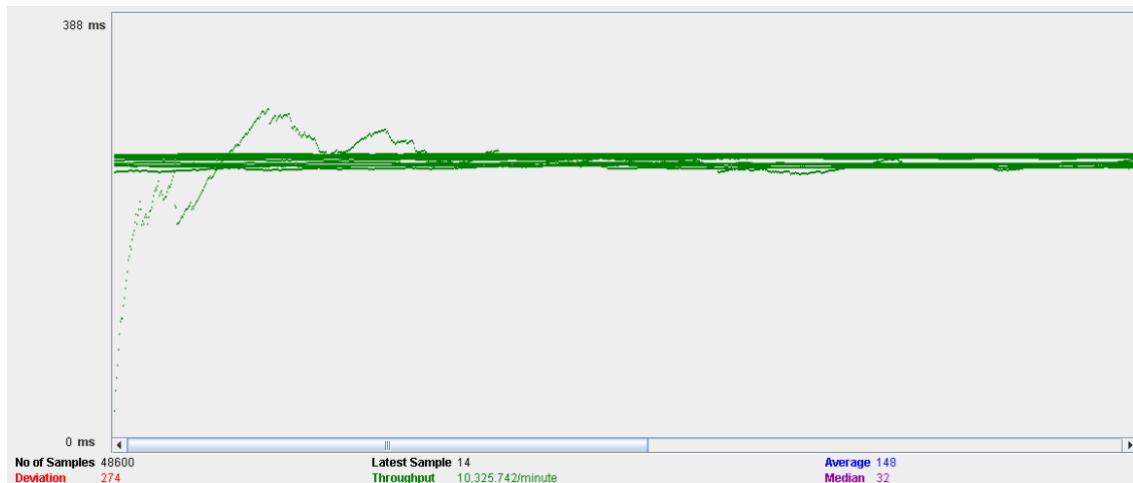
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso del procesador es bastante elevado aun dando unos tiempos adecuados.



La configuración en esta ocasión es de 90 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos que son mucho mayores que un segundo, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	310	226	597	13	4005	0.00%	28.7/sec	310.5
/styles/bootstrap.m...	2700	24	21	41	4	726	0.00%	9.8/sec	1160.2
/scripts/jquery.min.js	2700	20	17	34	4	365	0.00%	9.8/sec	930.8
/styles/cookiecons...	2700	8	6	13	1	569	0.00%	9.8/sec	40.4
/styles/style.css	2700	8	6	13	1	110	0.00%	9.8/sec	15.3
/styles/circle.css	2700	8	6	13	1	446	0.00%	9.8/sec	4.7
/scripts/bootstrap...	2700	13	10	22	2	860	0.00%	9.8/sec	356.8
/scripts/polyfills.js	2700	8	6	13	1	750	0.00%	9.8/sec	73.6
/scripts/helpers.js	2700	8	6	13	1	604	0.00%	9.8/sec	22.3
/scripts/cookiecon...	2700	12	10	20	2	208	0.00%	9.8/sec	335.7
/images/logo.jpg	2700	150	144	252	26	953	0.00%	9.8/sec	11027.4
/images/link_broke...	2700	56	51	102	8	295	0.00%	9.8/sec	3189.8
/security/login.do	2700	252	157	572	8	2896	0.00%	9.8/sec	61.0
/j_spring_security...	2700	587	436	1216	18	4576	0.00%	9.8/sec	112.0
/plan/display.do	2700	280	182	612	11	4278	0.00%	9.8/sec	87.0
/j_spring_security...	2700	303	231	565	16	4524	0.00%	9.8/sec	106.9
TOTAL	48600	148	32	382	1	4576	0.00%	172.1/sec	17440.5

Como podemos observar fijándonos en 90% Line, el grupo formado por la autenticación inicial y la vuelta a la pantalla de inicio genera un tiempo de respuesta de alrededor de 1800 ms que es inaceptable



En esta tabla, el rendimiento es de aproximadamente 10325 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos perform.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



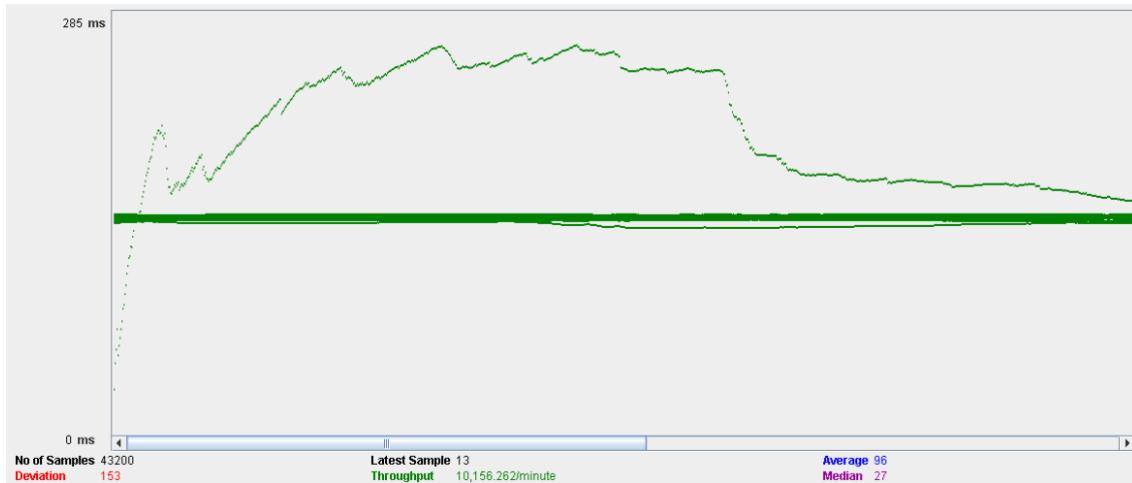
Por lo que, en este caso de uso, los usuarios permitidos serán entre 80 y 90.

Caso de uso 062: Ver tu suscripción a un plan de pago

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la vista de mi suscripción.

Comenzamos probando con 80 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	212	156	462	11	1907	0.00%	28.2/sec	308.1
/styles/style.css	2400	9	7	20	1	154	0.00%	9.6/sec	15.1
/styles/bootstrap.m...	2400	23	16	48	4	593	0.00%	9.6/sec	1144.5
/styles/circle.css	2400	9	7	19	1	216	0.00%	9.6/sec	4.6
/scripts/jquery.min.js	2400	20	14	43	4	615	0.00%	9.6/sec	918.2
/styles/cookiecons...	2400	10	7	20	1	167	0.00%	9.6/sec	39.9
/scripts/bootstrap....	2400	14	10	30	2	170	0.00%	9.6/sec	351.9
/scripts/polyfills.j...	2400	9	7	19	1	96	0.00%	9.6/sec	72.6
/scripts/helpers.js	2400	9	7	19	1	182	0.00%	9.6/sec	22.0
/scripts/cookiescon...	2400	13	9	26	2	218	0.00%	9.6/sec	331.1
/images/logo.jpg	2400	136	92	267	24	1034	0.00%	9.6/sec	10876.5
/springSecurity...	2400	123	85	267	6	1402	0.00%	9.6/sec	60.2
/images/link_broke...	2400	48	30	106	8	726	0.00%	9.7/sec	3140.4
/subscription/user/...	2400	171	122	372	10	1655	0.00%	9.7/sec	71.2
/j_spring_security...	2400	209	155	447	12	2053	0.00%	9.7/sec	105.2
TOTAL	43200	96	27	268	1	2973	0.00%	189.3/sec	17145.5



En esta tabla, el rendimiento es de 10156 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

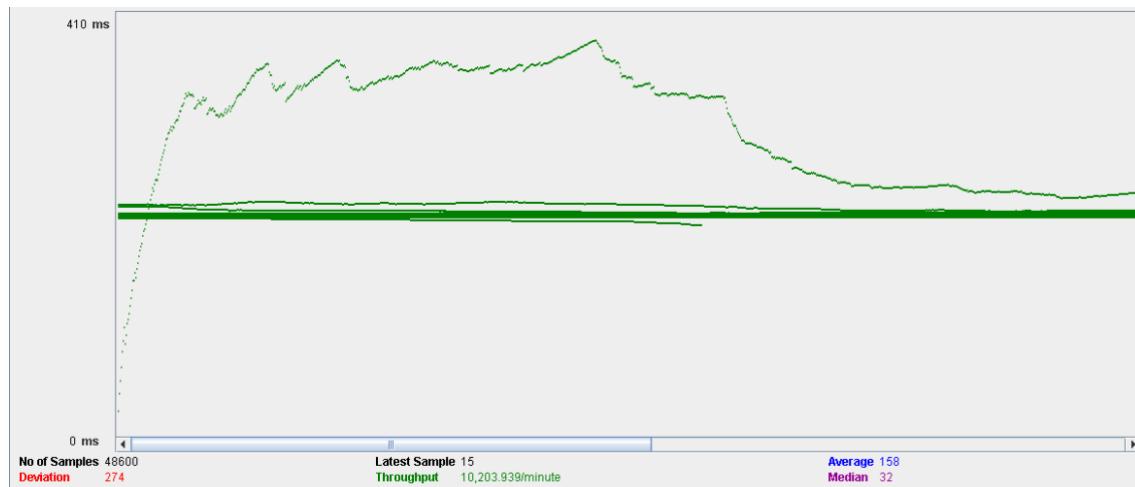
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso del procesador es bastante elevado aun dando unos tiempos adecuados.



La configuración en esta ocasión es de 90 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos que son mucho mayores que un segundo, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	343	269	686	10	3312	0.00%	28.3/sec	309.5
/styles/style.css	2700	10	8	19	1	768	0.00%	9.6/sec	15.1
/styles/bootstrap.m...	2700	25	19	50	4	185	0.00%	9.6/sec	1143.5
/styles/circle.cs...	2700	9	7	18	1	174	0.00%	9.6/sec	4.6
/scripts/jquery.min.js	2700	21	16	42	3	221	0.00%	9.6/sec	917.4
/scripts/cookiecons...	2700	9	7	18	1	303	0.00%	9.6/sec	39.9
/scripts/bootstrap...	2700	14	11	26	2	188	0.00%	9.6/sec	351.6
/scripts/polyfills.js	2700	9	7	18	1	98	0.00%	9.6/sec	72.6
/scripts/helpers.js	2700	9	7	19	1	161	0.00%	9.6/sec	22.0
/scripts/cookiecon...	2700	12	10	24	2	207	0.00%	9.6/sec	330.9
/images/logo.jpg	2700	141	131	261	24	575	0.00%	9.6/sec	10863.7
/security/login.do	2700	258	163	592	6	3148	0.00%	9.6/sec	60.1
/j_spring_security...	2700	597	496	1145	18	4305	0.00%	9.6/sec	113.1
/images/link_broke...	2700	55	47	107	8	435	0.00%	9.6/sec	3146.1
/subscription/user...	2700	307	230	636	10	2893	0.00%	9.7/sec	71.3
/j_spring_security...	2700	338	262	684	14	3259	0.00%	9.7/sec	105.4
TOTAL	48600	158	32	453	1	4305	0.00%	170.1/sec	17226.0

Como podemos observar fijándonos en 90% Line, el grupo formado por la autenticación inicial y la vuelta a la pantalla de inicio genera un tiempo de respuesta de alrededor de 1800 ms que es inaceptable



En esta tabla, el rendimiento es de aproximadamente 10203 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos perform.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



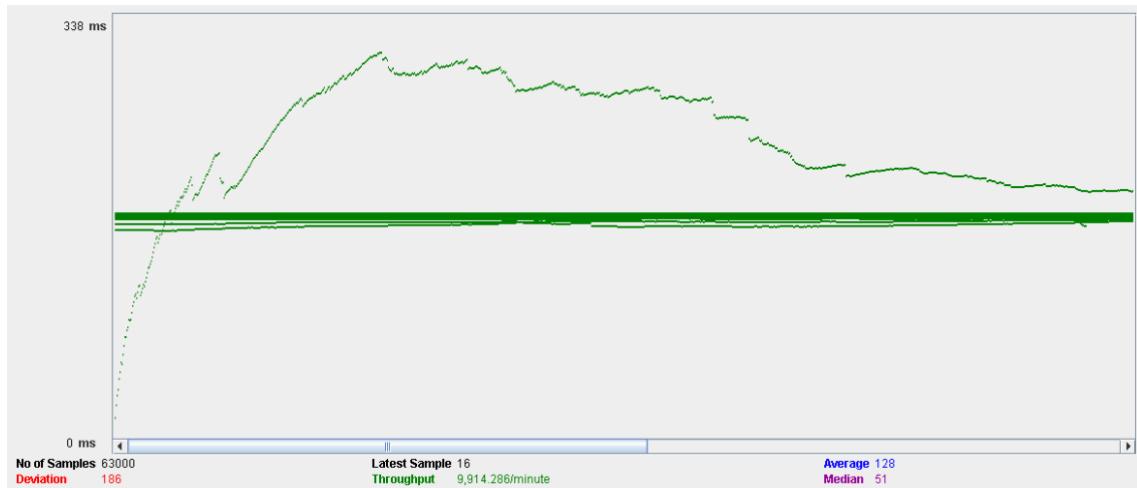
Por lo que, en este caso de uso, los usuarios permitidos serán entre 80 y 90.

Caso de uso 063: Editar suscripción

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la vista de mi suscripción y la editamos.

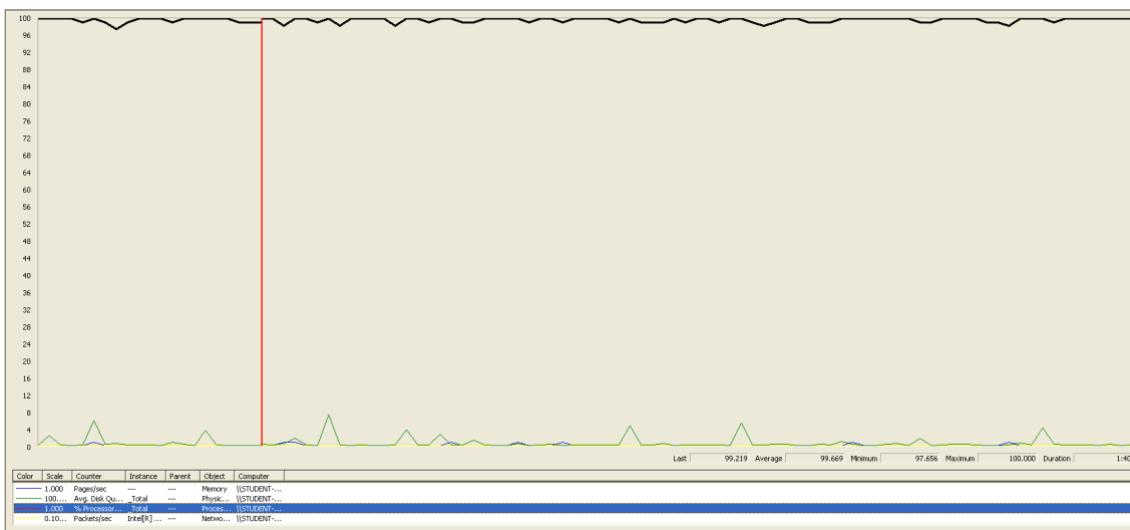
Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9000	227	184	427	11	2567	0.00%	23.6/sec	257.8
/styles/bootstrap.m...	3000	21	17	37	4	551	0.00%	8.1/sec	957.4
/styles/cookiecons...	3000	8	6	12	1	785	0.00%	8.1/sec	33.4
/styles/style.css	3000	7	6	12	1	404	0.00%	8.1/sec	12.6
/scripts/jquery.min.js	3000	19	15	32	3	684	0.00%	8.1/sec	768.1
/styles/circle.css	3000	7	6	13	1	266	0.00%	8.1/sec	3.9
/scripts/bootstrap....	3000	11	9	19	2	223	0.00%	8.1/sec	294.4
/scripts/polyfills.js	3000	7	6	13	1	211	0.00%	8.1/sec	60.8
/scripts/cookiecon...	3000	11	9	19	2	191	0.00%	8.1/sec	277.0
/scripts/helpers.js	3000	7	6	13	1	590	0.00%	8.1/sec	18.4
/images/link_broke...	3000	45	39	84	8	392	0.00%	8.1/sec	2631.5
/images/logo.jpg	3000	125	118	219	24	760	0.00%	8.1/sec	9102.2
/security/login.do	3000	163	114	329	7	2203	0.00%	8.1/sec	60.3
/j_spring_security...	3000	372	312	704	20	2396	0.00%	8.1/sec	94.8
/subscription/user/...	6000	195	149	386	10	2331	0.00%	16.0/sec	118.2
/subscription/user/...	6000	299	239	585	12	2907	0.00%	16.1/sec	131.3
/j_spring_security...	3000	213	174	400	13	2232	0.00%	8.1/sec	87.9
TOTAL	63000	128	51	336	1	2907	0.00%	165.2/sec	14532.1



En esta tabla, el rendimiento es de 9914 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

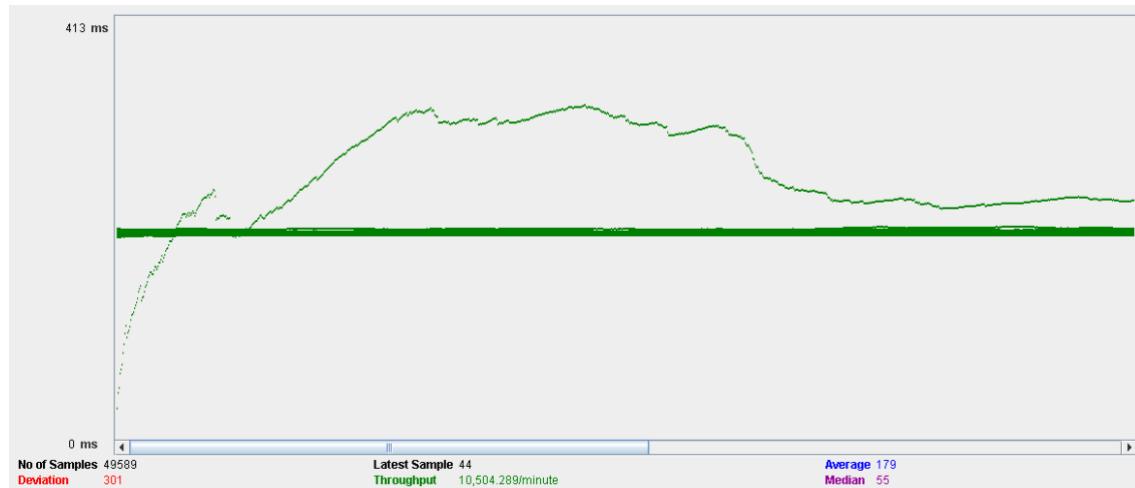
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso del procesador es bastante elevado aun dando unos tiempos adecuados, aunque no está continuamente al 100% de uso sino que tiene pequeñas caídas.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos que son mucho mayores que un segundo, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7029	319	221	670	15	4015	0.03%	24.8/sec	271.0
/styles/bootstrap.m...	2390	20	16	37	4	361	0.00%	8.5/sec	1003.4
/styles/cookiecons...	2390	7	6	12	1	366	0.00%	8.5/sec	35.0
/styles/style.css	2390	8	6	14	1	296	0.00%	8.5/sec	13.2
/scripts/jquery.min.js	2390	18	14	34	3	461	0.00%	8.5/sec	804.9
/styles/circle.css	2390	7	6	13	0	81	0.00%	8.5/sec	4.0
/scripts/bootstrap...	2390	11	9	21	2	180	0.00%	8.5/sec	308.5
/scripts/polylines.js	2390	7	6	14	1	98	0.00%	8.5/sec	63.7
/scripts/cookiecon...	2390	11	9	20	2	168	0.00%	8.5/sec	280.3
/scripts/helpers.js	2390	7	6	14	1	195	0.00%	8.5/sec	19.3
/images/link_broke...	2390	42	33	84	7	529	0.00%	8.5/sec	2756.3
/images/logo.jpg	2389	124	101	233	25	732	0.04%	8.5/sec	9566.6
/security/login.do	2367	250	144	595	7	2349	0.00%	8.4/sec	52.7
/j_spring_security...	2355	592	448	1216	19	6091	0.00%	8.4/sec	98.7
/subscription/user...	4638	283	187	626	12	3514	0.11%	16.8/sec	123.6
/j_spring_security...	4627	430	305	920	13	3386	0.09%	16.8/sec	136.7
/j_spring_security...	2284	308	206	637	19	3299	0.04%	8.4/sec	91.3
TOTAL	49589	179	55	492	0	6091	0.03%	175.1/sec	15555.8

Como podemos observar fijándonos en 90% Line, el grupo formado por la autenticación inicial y la vuelta a la pantalla de inicio genera un tiempo de respuesta de alrededor de 1900 ms que es inaceptable



En esta tabla, el rendimiento es de aproximadamente 10504 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos perform.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



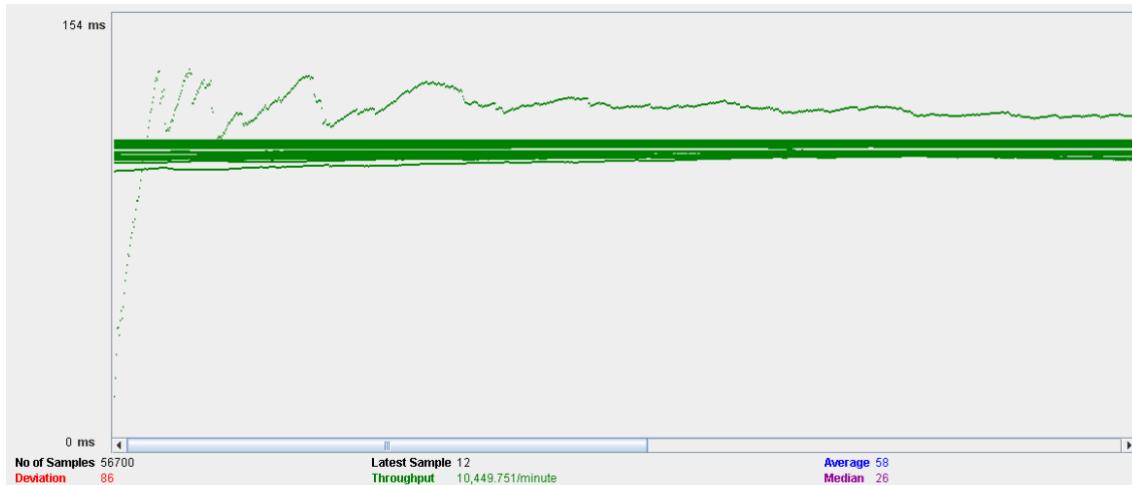
Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 064: Borrar suscripción

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la vista de mi suscripción y la editamos y borramos.

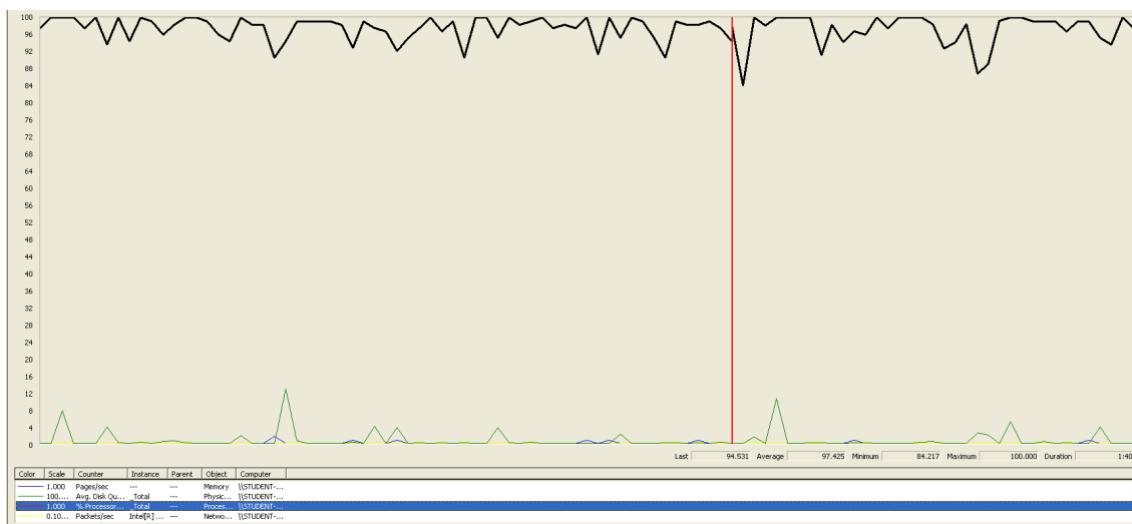
Comenzamos probando con 90 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	105	77	219	11	1785	0.00%	24.9/sec	271.7
/styles/style.css	2700	7	5	13	1	92	0.00%	8.5/sec	13.3
/styles/bootstrap.m...	2700	18	15	35	4	567	0.00%	8.5/sec	1012.8
/styles/cookiecons...	2700	7	5	13	1	308	0.00%	8.5/sec	35.3
/styles/circle.css	2700	7	5	12	1	1001	0.00%	8.5/sec	4.1
/scripts/jquery.min.js	2700	16	13	29	3	267	0.00%	8.5/sec	812.6
/scripts/bootstrap...	2700	10	8	19	2	95	0.00%	8.5/sec	311.5
/scripts/polyfills.j...	2700	7	5	13	1	108	0.00%	8.5/sec	64.3
/scripts/cookiecon...	2700	10	8	47	2	1291	0.00%	8.5/sec	293.1
/scripts/helpers.js	2700	7	5	13	1	554	0.00%	8.5/sec	19.5
/images/logo.jpg	2700	113	97	203	23	667	0.00%	8.5/sec	9626.3
/images/link_broke...	2700	43	35	82	8	434	0.00%	8.5/sec	2783.7
/security/login.do	2700	60	38	128	6	1154	0.00%	8.5/sec	53.3
/spring_security...	2700	146	105	302	12	1738	0.00%	8.5/sec	100.3
/subscription/user...	5400	79	53	166	8	1472	0.00%	16.9/sec	115.9
/subscription/user...	5400	91	62	190	9	2402	0.00%	17.1/sec	197.7
/spring_security...	2700	105	76	215	14	1043	0.00%	8.5/sec	93.1
TOTAL	56700	58	26	150	1	2402	0.00%	174.2/sec	15365.3



En esta tabla, el rendimiento es de 10449 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

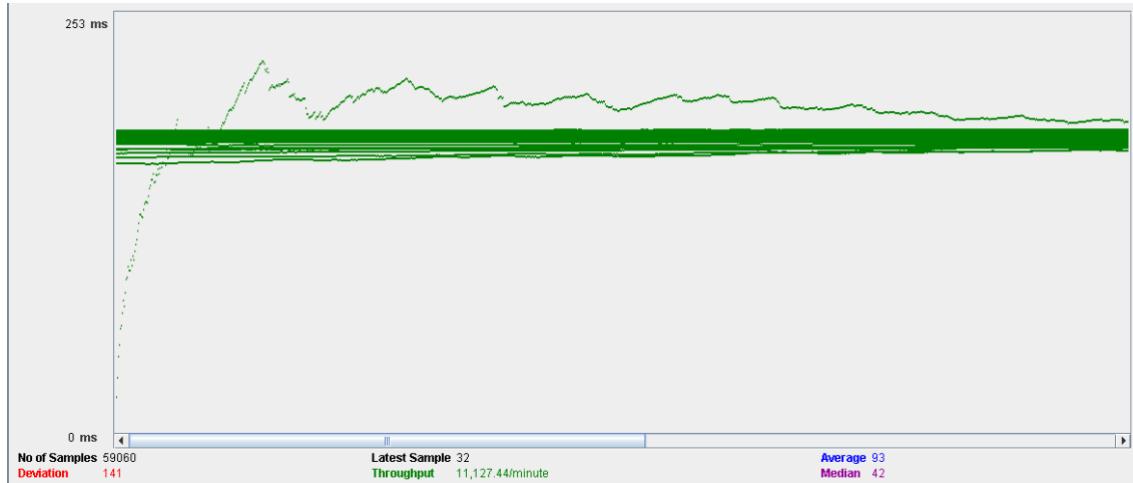
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso del procesador es de aproximadamente el 90% de su uso



La configuración en esta ocasión es de 100 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose errores, y unos tiempos adecuados, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8389	179	147	320	12	2613	0.01%	26.3/sec	287.7
/styles/style.css	2840	7	5	12	1	194	0.00%	8.9/sec	14.0
/styles/bootstrap.m...	2840	19	16	34	4	392	0.00%	8.9/sec	1059.6
/styles/cookiecons...	2840	7	5	11	1	287	0.00%	8.9/sec	36.9
/styles/circle.css	2840	7	5	11	1	899	0.00%	8.9/sec	4.3
/scripts/jquery.min.js	2839	17	14	29	3	211	0.00%	8.9/sec	850.0
/scripts/bootstrap....	2838	10	9	18	2	418	0.00%	8.9/sec	325.6
/scripts/polyfills.js	2838	7	6	12	1	746	0.00%	8.9/sec	67.2
/scripts/cookiecon...	2838	10	8	17	2	157	0.00%	8.9/sec	306.4
/scripts/helpers.js	2838	7	5	12	1	674	0.00%	8.9/sec	20.4
/images/logo.jpg	2838	119	114	201	24	524	0.00%	8.9/sec	10064.8
/images/link_broke...	2838	44	40	81	8	237	0.00%	9.0/sec	2923.0
/security/login.do	2819	114	83	227	6	1976	0.00%	8.9/sec	55.8
/j_spring_security...	2802	291	228	556	20	2713	0.00%	8.9/sec	104.6
/subscription/userf...	5549	147	113	264	9	3106	0.02%	17.8/sec	121.8
/subscription/userf...	5529	156	123	283	11	2152	0.02%	17.8/sec	206.4
/j_spring_security...	2745	178	150	309	5	1813	0.00%	8.9/sec	97.0
TOTAL	59060	93	42	233	1	3106	0.01%	185.5/sec	16494.6

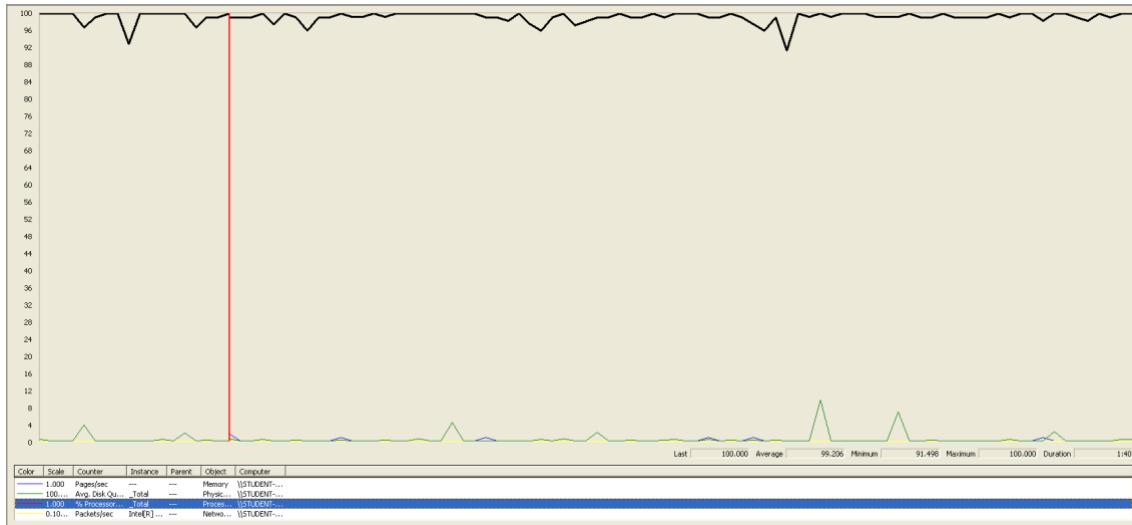
Como podemos observar fijándonos en 90% Line los tiempos son aceptables ninguno conjunto llegando al segundo de duración, sin embargo se han producido errores en la pantalla de inicio y al borrar la suscripción



En esta tabla, el rendimiento es de aproximadamente 11127 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en muchas ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



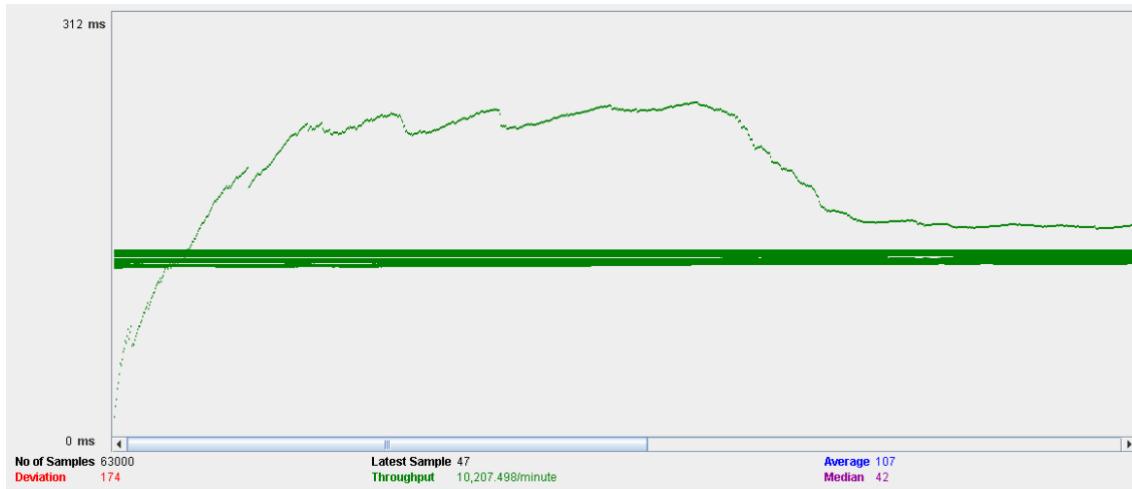
Por lo que, en este caso de uso, los usuarios permitidos serán entre 90 y 100.

Caso de uso 065: Suscribirse a un plan de pago

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la vista de los planes y nos suscribimos a uno.

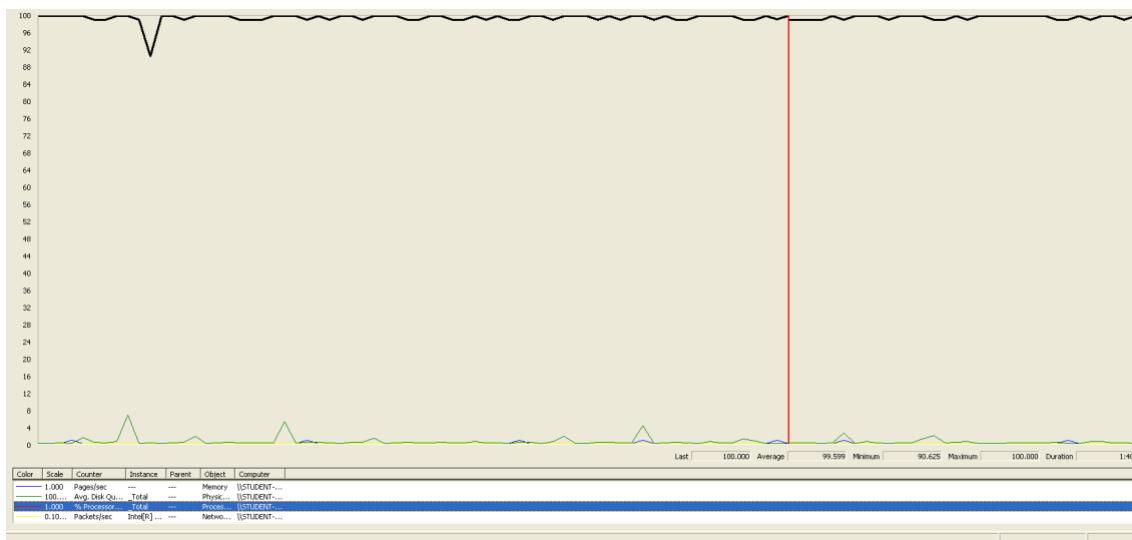
Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8999	191	142	377	10	5467	0.00%	24.5/sec	267.4
/styles/bootstrap.m...	3000	23	17	47	4	434	0.00%	8.3/sec	987.0
/styles/style.css	3000	9	6	18	1	258	0.00%	8.3/sec	13.0
/styles/circle.css	3000	9	6	18	1	369	0.00%	8.3/sec	4.0
/scripts/jquery.min.js	3000	21	14	41	3	496	0.00%	8.3/sec	791.8
/styles/cookiecons...	3000	9	6	18	1	197	0.00%	8.3/sec	34.4
/scripts/bootstrap....	3000	14	10	27	2	234	0.00%	8.3/sec	303.5
/scripts/polyfills.js	3000	9	6	19	1	200	0.00%	8.3/sec	62.6
/scripts/helpers.js	3000	9	6	19	1	197	0.00%	8.3/sec	19.0
/scripts/cookiecon...	3000	13	9	25	2	500	0.00%	8.3/sec	285.7
/images/logo.jpg	3000	141	115	276	24	865	0.00%	8.3/sec	9385.6
/security/login.do	3000	141	90	286	7	4306	0.00%	8.3/sec	51.9
/j_spring_security_...	3000	323	241	650	12	3306	0.00%	8.3/sec	97.7
/images/link_broke...	3000	52	36	113	9	544	0.00%	8.3/sec	2715.3
/plan/display.do	3000	162	123	328	10	2637	0.00%	8.3/sec	71.4
/subscription/user...	3000	161	114	319	10	2782	0.00%	8.3/sec	121.6
/subscription/user...	3000	216	165	438	16	4294	0.00%	8.3/sec	70.3
/subscription/user...	2999	165	121	322	11	4217	0.00%	8.4/sec	61.9
/j_spring_security_...	2999	194	148	365	13	2560	0.00%	8.4/sec	91.4
TOTAL	62997	107	42	274	1	5467	0.00%	170.9/sec	15096.6



En esta tabla, el rendimiento es de 10207 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

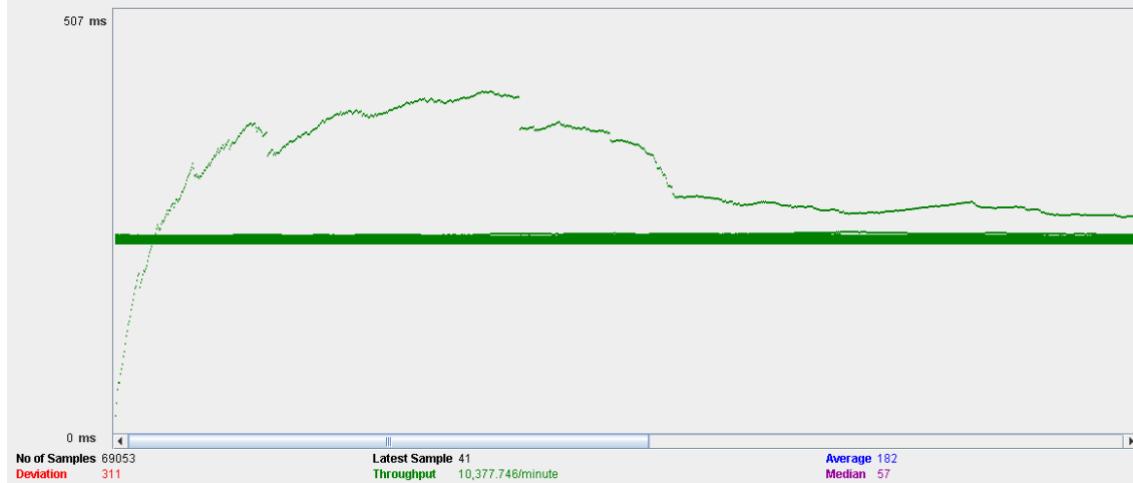
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso del procesador es de aproximadamente el 100% de su uso casi todo el rato aún dando unos tiempos razonables.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9850	330	228	714	11	3937	0.00%	24.7/sec	269.6
/styles/bootstrap.m...	3296	22	17	42	4	512	0.00%	8.3/sec	980.8
/styles/style.css	3296	8	6	15	1	497	0.00%	8.3/sec	12.9
/styles/circle.css	3296	7	6	13	1	189	0.00%	8.3/sec	3.9
/scripts/jquery.min.js	3296	18	15	34	4	376	0.00%	8.3/sec	786.9
/scripts/cookiecons...	3296	8	6	15	1	461	0.00%	8.3/sec	34.2
/scripts/bootstrap....	3296	12	9	21	2	345	0.00%	8.3/sec	301.6
/scripts/polyfills.j...	3296	8	6	14	1	359	0.00%	8.3/sec	62.3
/scripts/helpers.js	3296	8	6	14	1	442	0.00%	8.3/sec	18.9
/scripts/cookiecon...	3296	11	9	20	2	385	0.00%	8.3/sec	283.8
/images/logo.jpg	3296	131	115	234	24	1291	0.00%	8.3/sec	9347.1
/security/login.do	3292	275	156	666	7	4732	0.00%	8.3/sec	51.8
/j_spring_security...	3288	649	496	1333	20	4744	0.00%	8.3/sec	97.7
/images/link_broke...	3288	48	40	92	8	415	0.00%	8.3/sec	2717.9
/plan/display.do	3283	321	210	745	12	3382	0.00%	8.4/sec	71.6
/subscription/userf...	3280	303	195	672	12	3794	0.00%	8.4/sec	122.4
/subscription/userf...	3273	360	272	721	16	2931	0.00%	8.4/sec	70.7
/subscription/userf...	3273	320	199	735	11	4284	0.00%	8.4/sec	62.0
/j_spring_security...	3266	331	225	698	9	3911	0.00%	8.4/sec	91.4
TOTAL	69053	182	57	491	1	4744	0.00%	173.0/sec	15302.6

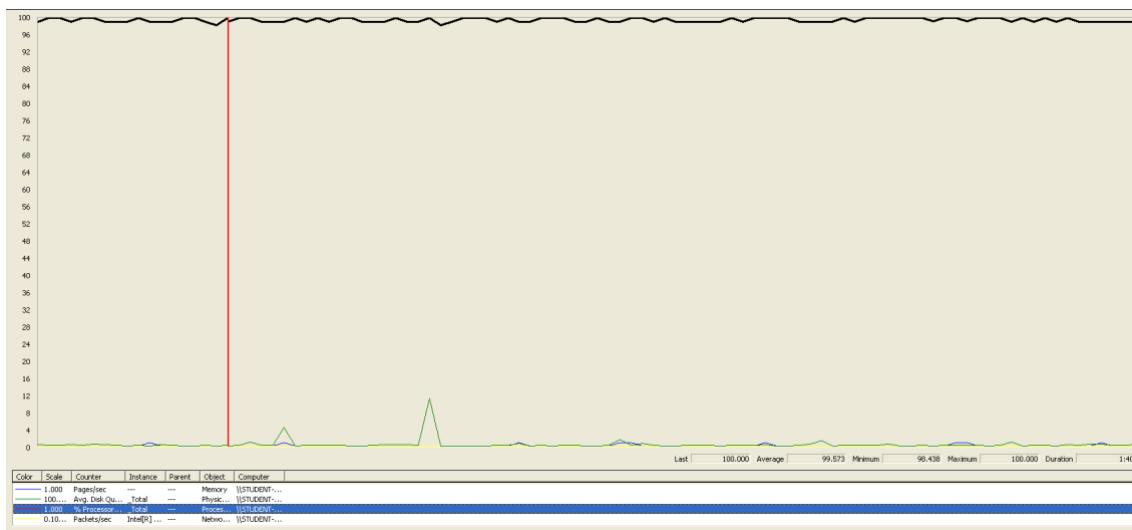
Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que da lugar a 2000 ms que supera con creces los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 10377 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos perform.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



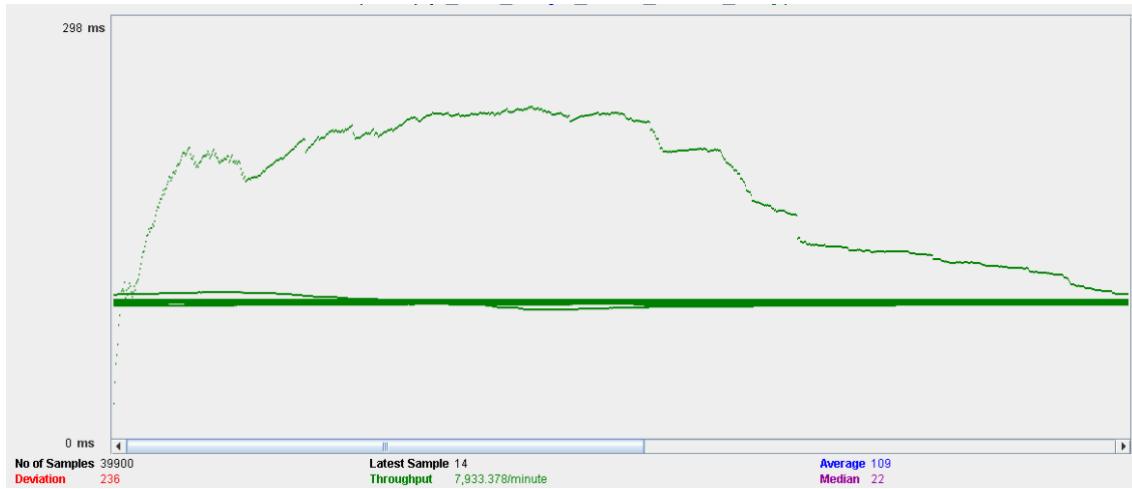
Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 066: Listar los niveles y desplegar

En este caso de uso, accedemos a la página, nos autenticamos como administrador, y accedemos a la lista de los niveles y desplegamos uno

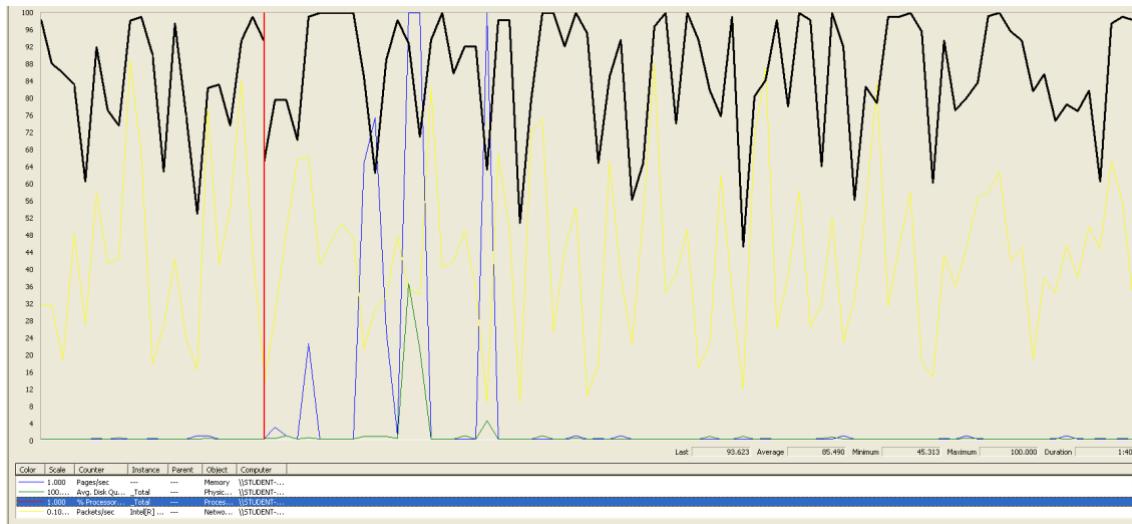
Comenzamos probando con 70 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	6300	139	81	317	11	2216	0.00%	20.9/sec	226.0
/styles/style.css	2100	7	5	12	1	291	0.00%	7.2/sec	11.2
/styles/cookiecons...	2100	7	5	12	1	167	0.00%	7.2/sec	29.6
/scripts/jquery.min.js	2100	16	12	30	4	511	0.00%	7.2/sec	682.5
/styles/circle.css	2100	6	4	12	1	166	0.00%	7.2/sec	3.4
/scripts/bootstrap....	2100	10	8	19	2	129	0.00%	7.2/sec	261.6
/styles/bootstrap.m...	2100	18	14	33	4	195	0.00%	7.2/sec	850.7
/scripts/polyfills.js	2100	7	5	13	1	526	0.00%	7.2/sec	54.0
/scripts/helpers.js	2100	6	5	12	1	135	0.00%	7.2/sec	16.4
/scripts/cookiecon...	2100	9	7	16	2	216	0.00%	7.2/sec	246.1
/images/logo.jpg	2100	117	91	225	24	641	0.00%	7.2/sec	8091.3
/security/login.do	2100	85	38	189	6	1633	0.00%	7.2/sec	44.7
/j_spring_security....	2100	175	68	436	17	1866	0.00%	7.2/sec	82.0
/images/link_broke...	2100	35	26	69	8	253	0.00%	7.2/sec	2339.0
/levelList.do	2100	100	44	224	10	2024	0.00%	7.2/sec	63.3
/levelIdDisplay.do	2100	865	752	1199	581	4221	0.00%	7.1/sec	50.2
/j_spring_security....	2100	186	107	422	14	3716	0.00%	7.2/sec	78.1
TOTAL	39900	109	22	281	1	4221	0.00%	132.2/sec	12743.0



En esta tabla, el rendimiento es de 7993 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

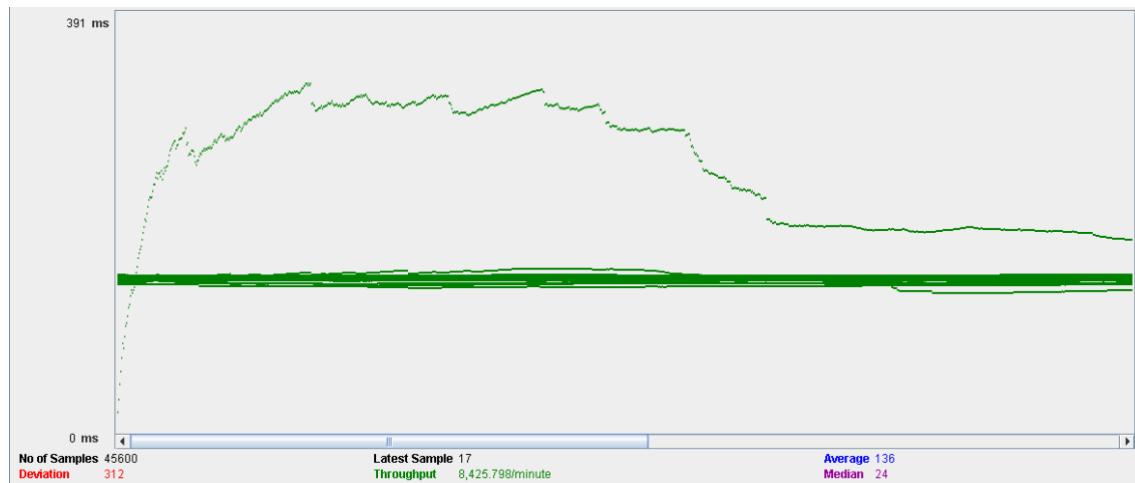
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso del procesador oscila entre el 50% y el 100% y la tarjeta de red entre el 20% y el 80%.



La configuración en esta ocasión es de 80 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	185	106	395	11	3340	0.00%	22.2/sec	240.0
/styles/style.css	2400	7	5	13	1	186	0.00%	7.7/sec	12.0
/styles/cookiecons...	2400	7	5	13	1	378	0.00%	7.7/sec	31.8
/scripts/jquery.min.js	2400	16	13	31	3	392	0.00%	7.7/sec	731.0
/styles/circle.css	2400	6	5	12	1	169	0.00%	7.7/sec	3.7
/scripts/bootstrap...	2400	11	8	20	2	531	0.00%	7.7/sec	280.2
/scripts/bootstrap.m...	2400	19	15	37	4	233	0.00%	7.7/sec	911.3
/scripts/polyfills.js	2400	7	5	13	1	297	0.00%	7.7/sec	57.8
/scripts/helpers.js	2400	7	5	13	1	474	0.00%	7.7/sec	17.5
/scripts/cookiecon...	2400	10	7	18	2	349	0.00%	7.7/sec	263.7
/images/logo.jpg	2400	117	90	224	25	539	0.00%	7.7/sec	8656.4
/security/login.do	2400	143	80	329	6	4647	0.00%	7.7/sec	47.9
/j_spring_security....	2400	257	151	618	6	2887	0.00%	7.7/sec	87.9
/images/link_broke...	2400	38	28	76	8	255	0.00%	7.7/sec	2506.0
/levelList.do	2400	150	64	342	10	3122	0.00%	7.7/sec	67.9
/level/display.do	2400	995	766	1616	581	6569	0.00%	7.6/sec	53.2
/j_spring_security....	2400	237	127	540	3	3479	0.00%	7.6/sec	82.9
TOTAL	45600	136	24	375	1	6569	0.00%	140.4/sec	13533.9

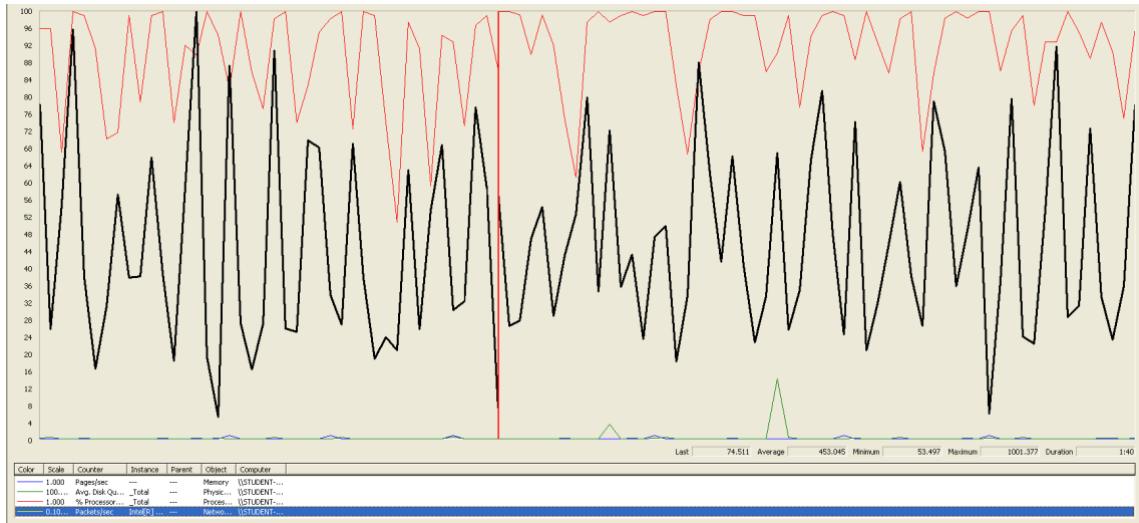
Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al hacer el display del nivel que se tarda 1616 ms que supera con creces los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 8425 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, al 100% es algunas ocasiones con la CPU y en otras por la tarjeta de red, así que el límite de tiempo viene debido por ambos elementos.



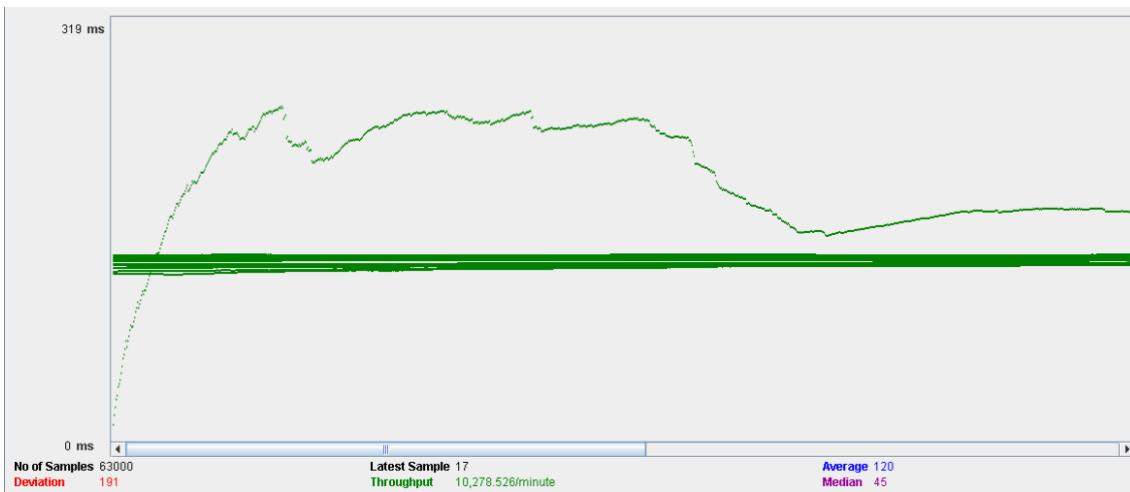
Por lo que, en este caso de uso, los usuarios permitidos serán entre 70 y 80.

Caso de uso 067: Editar un nivel

En este caso de uso, accedemos a la página, nos autenticamos como administrador, y accedemos a la lista de los niveles y editamos uno

Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9000	210	161	390	12	3813	0.00%	24.5/sec	264.9
/styles/style.css	3000	8	6	14	1	1210	0.00%	8.4/sec	13.1
/styles/cookiecons...	3000	8	6	14	1	818	0.00%	8.4/sec	34.6
/scripts/jquery.min.js	3000	19	15	34	3	1926	0.00%	8.4/sec	797.3
/styles/circle.css	3000	7	6	14	1	102	0.00%	8.4/sec	4.0
/scripts/bootstrap....	3000	12	10	21	2	1215	0.00%	8.4/sec	305.6
/styles/bootstrap.m...	3000	22	19	39	4	779	0.00%	8.4/sec	994.0
/scripts/polyfills.js	3000	8	6	13	1	1185	0.00%	8.4/sec	63.1
/scripts/cookiecon...	3000	12	9	20	2	578	0.00%	8.4/sec	287.6
/scripts/helpers.js	3000	7	6	13	1	278	0.00%	8.4/sec	19.1
/images/logo.jpg	3000	140	128	240	24	1554	0.00%	8.4/sec	9446.4
/security/login.do	3000	150	96	294	6	2764	0.00%	8.4/sec	52.2
/_spring_security....	3000	340	252	706	17	3164	0.00%	8.4/sec	95.9
/images/link_broke...	3000	48	41	90	8	502	0.00%	8.4/sec	2735.1
/level/list.do	6000	162	119	315	10	3008	0.00%	16.6/sec	147.0
/level/administrator...	6000	279	225	536	11	2743	0.00%	16.8/sec	160.7
/_spring_security....	3000	215	166	391	13	2600	0.00%	8.4/sec	91.4
TOTAL	63000	120	45	305	1	3813	0.00%	171.3/sec	15108.6



En esta tabla, el rendimiento es de 10278 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

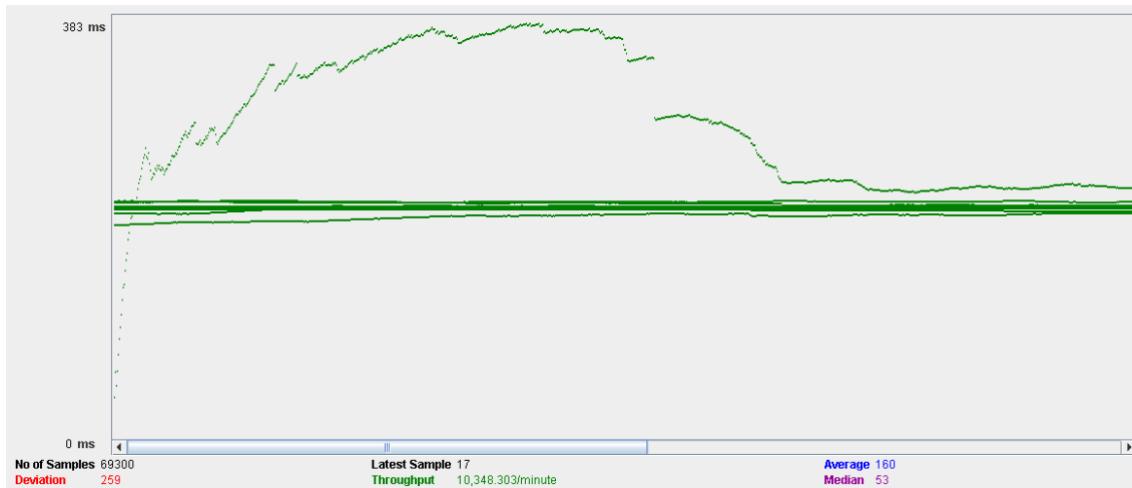
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es casi del 100% en todo momento a pesar de salir unos tiempos admisibles.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9900	267	191	554	13	3231	0.00%	24.6/sec	266.7
/styles/style.css	3300	8	6	15	1	300	0.00%	8.4/sec	13.1
/styles/cookiecons...	3300	8	6	15	1	272	0.00%	8.4/sec	34.8
/scripts/jquery.min.js	3300	19	15	36	3	438	0.00%	8.4/sec	800.4
/styles/circle.css	3300	9	6	16	1	344	0.00%	8.4/sec	4.0
/scripts/bootstrap....	3300	12	10	22	2	200	0.00%	8.4/sec	306.8
/scripts/bootstrap.m...	3300	22	17	42	4	407	0.00%	8.4/sec	997.7
/scripts/polyfills.j...	3300	8	6	15	1	128	0.00%	8.4/sec	63.3
/scripts/cookiecon...	3300	11	9	21	2	252	0.00%	8.4/sec	288.6
/scripts/helpers.js	3300	8	6	15	1	441	0.00%	8.4/sec	19.2
/images/logo.jpg	3300	134	117	248	24	699	0.00%	8.4/sec	9481.5
/security/login.do	3300	219	132	502	7	3843	0.00%	8.4/sec	52.5
/j_spring_security_...	3300	491	379	984	19	3329	0.00%	8.4/sec	96.3
/images/link_broke...	3300	50	42	89	8	493	0.00%	8.4/sec	2743.3
/level/list.do	6600	243	162	535	12	3388	0.00%	16.7/sec	147.6
/level/administrator...	6600	395	300	822	13	3132	0.00%	16.9/sec	161.5
/j_spring_security_...	3300	284	204	575	16	4704	0.00%	8.4/sec	91.9
TOTAL	69300	160	53	431	1	4704	0.00%	172.5/sec	15211.1

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1500 ms que supera los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 10348 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



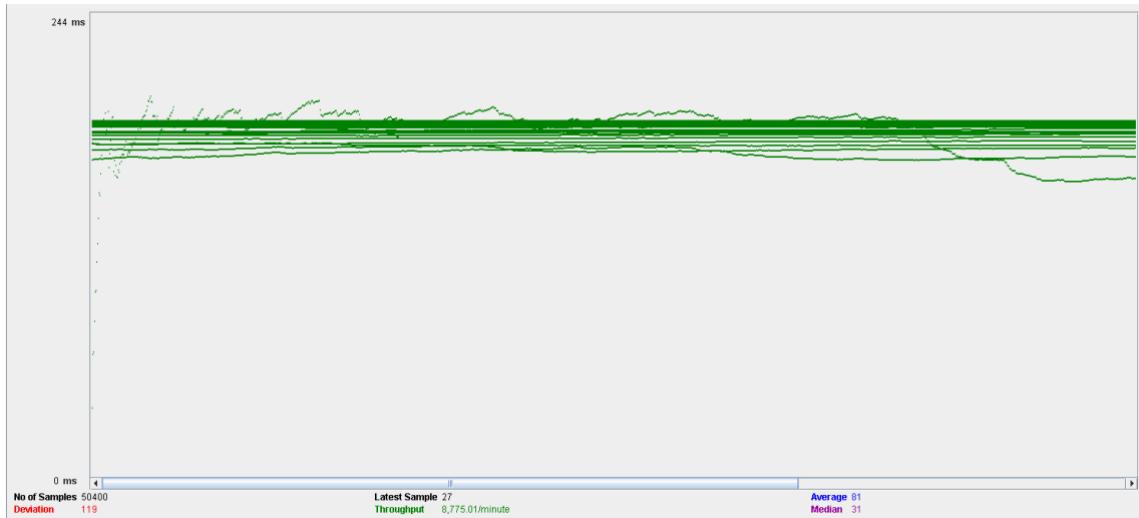
Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 068: Crear un nivel

En este caso de uso, accedemos a la página, nos autenticamos como administrador, y accedemos a la lista de los niveles y creamos uno

Comenzamos probando con 80 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	148	103	334	13	1305	0.00%	20.9/sec	226.1
/styles/style.css	2400	8	6	16	1	366	0.00%	7.1/sec	11.2
/styles/bootstrap.min.css	2400	22	15	43	4	342	0.00%	7.1/sec	847.3
/styles/cookieconsent.mi...	2400	8	6	16	1	106	0.00%	7.1/sec	29.5
/styles/circle.css	2400	8	6	16	1	268	0.00%	7.1/sec	3.4
/scripts/bootstrap.js	2400	19	13	35	1	399	0.00%	7.1/sec	67.6
/scripts/bootstrap.min.js	2400	12	9	23	2	389	0.00%	7.1/sec	260.6
/scripts/helper.js	2400	8	6	16	0	412	0.00%	7.1/sec	53.0
/scripts/helpers.js	2400	8	6	16	1	406	0.00%	7.1/sec	16.3
/scripts/cookieconsent.js	2400	12	9	23	2	326	0.00%	7.1/sec	245.2
/images/logo.jpg	2400	126	99	246	27	1085	0.00%	7.1/sec	8053.6
/images/link_broken.png	2400	49	35	102	9	383	0.00%	7.1/sec	2329.1
/security/login.do	2400	90	61	200	7	1396	0.00%	7.1/sec	44.6
/_spring_security_check	2400	204	149	439	10	2976	0.00%	7.1/sec	10
/level/administrator/cre...	2400	113	79	251	11	1166	0.00%	14.2/sec	139.7
/level/administrator/edit.d...	2400	148	104	320	12	2524	0.00%	7.2/sec	69.7
/_spring_security_logout	2400	152	108	351	12	1255	0.00%	7.2/sec	70.6
TOTAL	50400	81	31	229	0	2576	0.00%	146.3/sec	12905.6



En esta tabla, el rendimiento es de 8775 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

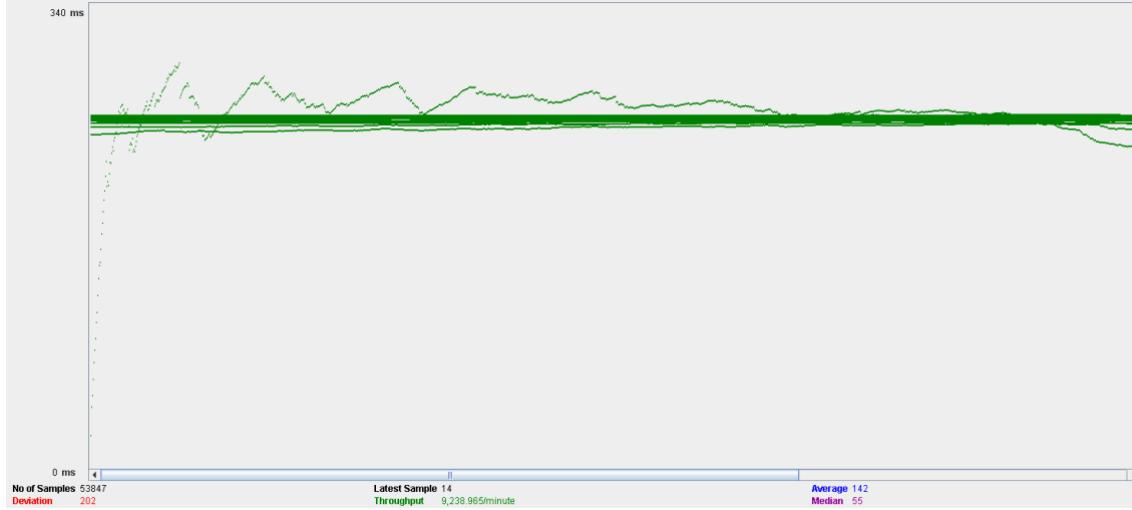
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU oscila entre el 80% y el 100% llegando a este último en pocas ocasiones.



La configuración en esta ocasión es de 90 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7648	271	221	625	13	2721	0.04%	21.9/sec	236.7
/static/style.css	2567	10	8	19	1	262	0.00%	7.4/sec	11.6
/static/bootstrap.min.css	2567	27	22	51	5	411	0.00%	7.4/sec	87.2
/static/cookieconsent.mi...	2566	10	8	20	1	253	0.00%	7.4/sec	30.6
/static/circle.css	2566	10	8	19	1	366	0.00%	7.4/sec	3.5
/scripts/jquery.min.js	2566	23	19	43	4	396	0.00%	7.4/sec	705.1
/scripts/bootstrap.min.js	2566	15	12	28	2	302	0.00%	7.4/sec	270.2
/scripts/poofills.js	2566	10	8	19	1	145	0.00%	7.4/sec	55.8
/scripts/helpers.js	2566	10	8	20	1	466	0.00%	7.4/sec	16.9
/scripts/cookieconsent.js	2566	15	12	28	3	488	0.00%	7.4/sec	254.3
/images/logo.jpg	2566	157	146	281	29	727	0.00%	7.4/sec	8349.4
/images/link_broken.png	2566	61	53	116	9	252	0.00%	7.4/sec	2422.8
/security/login.do	2572	191	140	392	7	2314	0.00%	7.4/sec	46.3
/spring_security_check	2568	412	339	600	20	3087	0.08%	7.4/sec	84.7
/levellist.do	5062	222	172	447	12	2278	0.00%	14.8/sec	134.7
/level/administrator/creat...	2536	257	209	492	11	2836	0.00%	7.4/sec	72.2
/level/administrator/reedit.do	2521	261	212	496	14	2310	0.04%	7.4/sec	72.8
/spring_security_logout	2502	271	225	496	4	1946	0.00%	7.4/sec	80.3
TOTAL	53847	142	55	382	1	3087	0.01%	154.0/sec	13690.7

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1300 ms que supera los 1000 ms que se consideran aceptables, además como se ve se están produciendo errores.



En esta tabla, el rendimiento es de aproximadamente 9238 ejecuciones por minuto (aumenta ligeramente al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



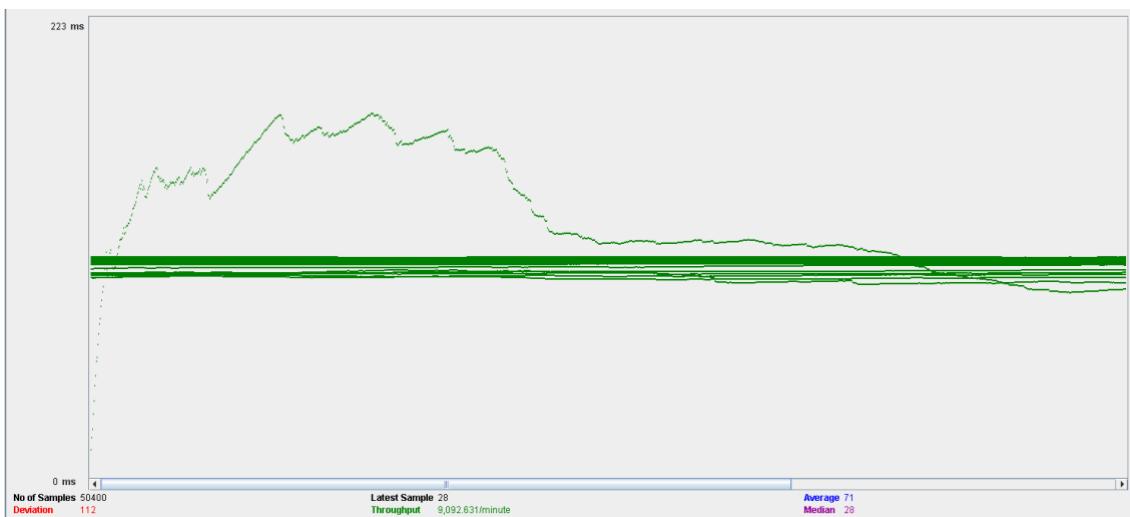
Por lo que, en este caso de uso, los usuarios permitidos serán entre 80 y 90.

Caso de uso 069: Borrar un nivel

En este caso de uso, accedemos a la página, nos autenticamos como administrador, y accedemos a la lista de los niveles y editamos y borramos uno

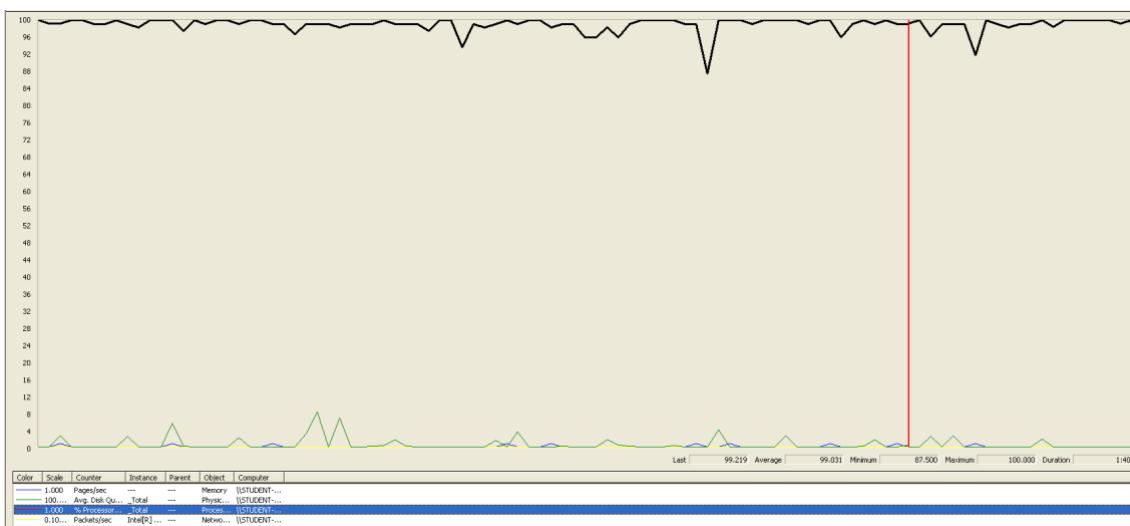
Comenzamos probando con 80 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	140	84	306	12	3232	0.00%	21.8/sec	234.3
/stylesheets/bootstrap.min.css	2400	21	15	42	3	915	0.00%	7.4/sec	881.3
/stylesheets.css	2400	8	6	16	1	162	0.00%	7.4/sec	11.6
/stylesheets/cookieconsent.mi...	2400	7	6	14	1	238	0.00%	7.4/sec	30.7
/scripts/jquery.min.js	2400	17	13	34	3	357	0.00%	7.4/sec	707.0
/stylesheets/circle.css	2400	7	5	14	1	325	0.00%	7.4/sec	3.5
/scripts/bootstrap.min.js	2400	11	9	22	2	303	0.00%	7.4/sec	271.0
/scripts/bootstrap-fills.js	2400	8	6	15	1	299	0.00%	7.4/sec	9.0
/scripts/fontawesome.js	2400	8	6	16	1	306	0.00%	7.4/sec	16.8
/scripts/cookieconsent.js	2400	11	9	21	2	267	0.00%	7.4/sec	256.0
/images/logo.png	2400	122	100	228	26	778	0.00%	7.4/sec	8373.9
/security/login.do	2400	85	64	183	8	1728	0.00%	7.4/sec	46.3
/_spring_security_check	2400	183	132	380	6	1871	0.00%	7.4/sec	85.0
/images/link_broken.png	2400	44	33	90	8	526	0.00%	7.4/sec	2422.9
/level/list.do	4800	102	72	217	11	1754	0.00%	14.7/sec	130.0
/level/administrator/edit.do	4800	100	69	220	10	2008	0.00%	14.9/sec	203.7
/_spring_security_logout	2400	145	100	307	11	2088	0.00%	7.4/sec	81.0
TOTAL	50400	71	28	195	1	3232	0.00%	151.5/sec	13424.9



En esta tabla, el rendimiento es de 9092 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU oscila entre el 95% y el 100% llegando a este último en pocas ocasiones.



La configuración en esta ocasión es de 90 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	258	212	497	12	2520	0.00%	21.5/sec	232.9
/styles/bootstrap.min.css	2700	28	22	65	4	212	0.00%	7.4/sec	873.0
/styles/style.css	2700	11	8	21	1	419	0.00%	7.4/sec	11.6
/styles/cookieconsent.mi	2700	11	8	20	1	766	0.00%	7.4/sec	30.4
/scripts/query.min.js	2700	24	19	45	3	746	0.00%	7.4/sec	700.4
/styles/circle.css	2700	10	8	21	1	283	0.00%	7.4/sec	3.5
/scripts/bootstrap.min.js	2700	15	12	30	2	399	0.00%	7.4/sec	26.6
/scripts/helpers.js	2700	10	8	21	1	325	0.00%	7.4/sec	55.4
/scripts/helpers.js	2700	10	8	20	1	193	0.00%	7.4/sec	18.8
/scripts/cookieconsent.js	2700	15	12	28	2	209	0.00%	7.4/sec	252.6
/images/logo.jpg	2700	166	149	297	27	1180	0.00%	7.4/sec	8297.7
/security/login.do	2700	179	131	362	7	2161	0.00%	7.4/sec	45.9
/j_spring_security_check	2700	375	319	717	19	2487	0.00%	7.4/sec	84.3
/images/link_broken.png	2700	61	50	121	9	217	0.00%	7.4/sec	2402.9
/availableList.do	5400	206	159	416	12	2800	0.00%	14.0/sec	129.0
/newAdminAdministratorredit.do	5400	117	105	381	10	2915	0.00%	14.7/sec	20.9
/j_spring_security_logout	2700	266	218	511	9	2859	0.00%	7.4/sec	80.2
TOTAL	56700	131	51	356	1	2915	0.00%	150.6/sec	13344.7

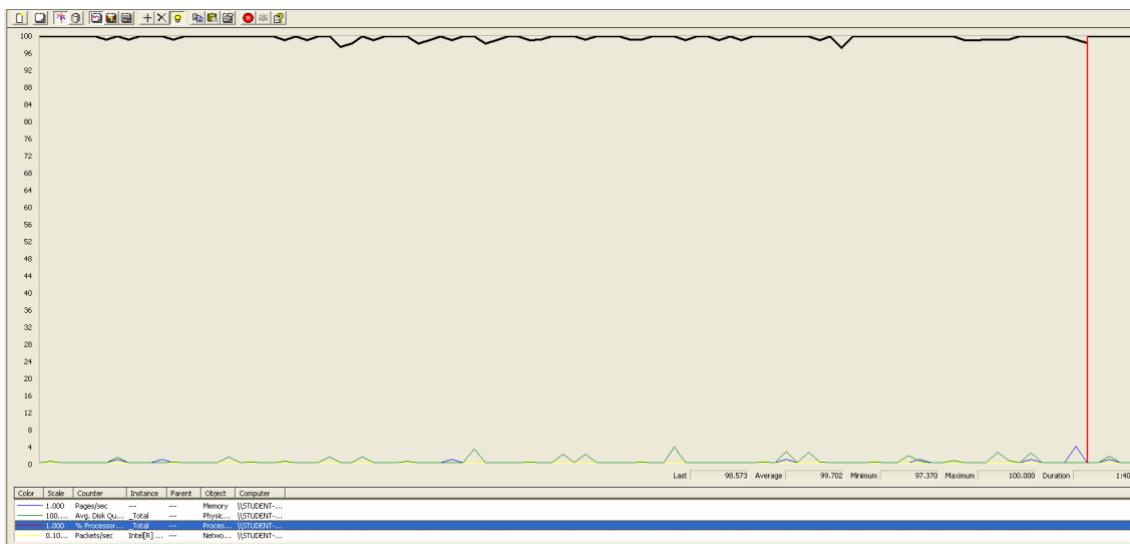
Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1230 ms que supera por demasiado los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 9038 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



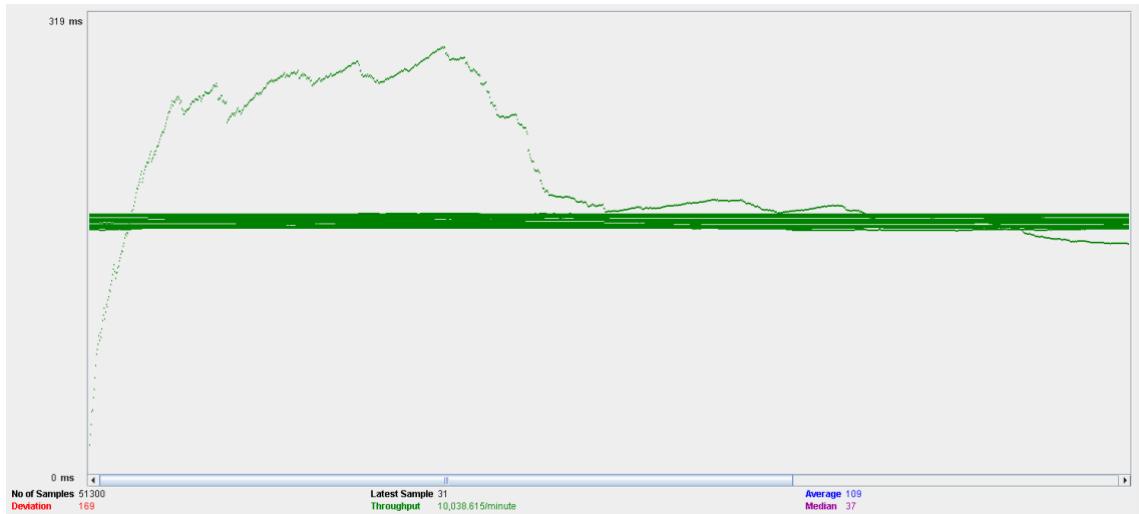
Por lo que, en este caso de uso, los usuarios permitidos serán entre 80 y 90.

Caso de uso 070: Listar tus participaciones y desplegar conjunta

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de mis participaciones y desplegamos la conjunta de una de ellas.

Comenzamos probando con 90 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	217	184	375	10	4169	0.00%	26.4/sec	288.5
/js/participationconsent.html	2700	8	6	13	1	225	0.00%	0.0/sec	37.4
/js/participationmin.js	2700	19	16	34	4	240	0.00%	0.0/sec	861.3
/style/circle.css	2700	7	6	12	1	197	0.00%	0.0/sec	1.0
/style/style.css	2700	7	6	12	2	103	0.00%	0.0/sec	14.1
/style/bootstrap.min.css	2700	22	19	41	4	415	0.00%	9.0/sec	1073.6
/scripts/bootstrap.min.js	2700	12	10	20	2	196	0.00%	9.0/sec	330.1
/scripts/polyfills.js	2700	7	6	13	2	288	0.00%	9.0/sec	68.1
/scripts/helpers.js	2700	7	6	12	1	233	0.00%	9.0/sec	20.6
/js/participationconsent.js	2700	11	9	19	2	204	0.00%	9.0/sec	30.6
/images/logo.png	2700	142	139	244	24	410	0.00%	0.0/sec	10189.6
/secure/login.do	2700	139	103	262	7	4141	0.00%	9.1/sec	56.5
/_spring_security_check	2700	356	295	826	20	3822	0.00%	9.1/sec	109.2
/images/link_broken.png	2700	50	46	94	8	269	0.00%	9.1/sec	2952.3
/participationUserList.do	2700	201	168	345	15	2216	0.00%	9.0/sec	80.6
/groupOnDisplay.do	2700	214	179	359	13	3975	0.00%	9.0/sec	80.6
/_spring_security_logout	2700	213	180	371	5	2259	0.00%	9.0/sec	98.6
TOTAL	51300	109	97	283	1	4169	0.00%	167.3/sec	16146.7



En esta tabla, el rendimiento es de 10038 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

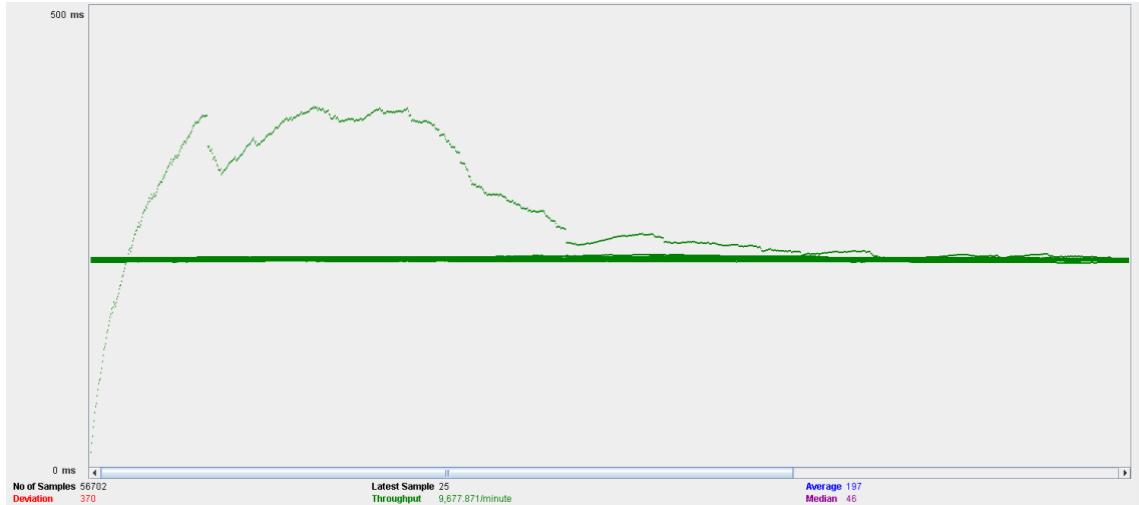
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% a pesar de tener unos tiempos admisibles.



La configuración en esta ocasión es de 100 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8926	387	261	821	13	5381	0.00%	25.4/sec	277.7
/statics/cookieconsent.min.js	2993	9	7	17	2	342	0.00%	8.6/sec	36.3
/scripts/querymin.js	2993	21	17	39	4	506	0.00%	8.5/sec	812.2
/statics/circle.css	2993	8	6	15	1	160	0.00%	8.5/sec	4.1
/statics/mie.css	2993	8	7	15	1	341	0.00%	8.5/sec	13.3
/statics/bootstrap.min.css	2993	25	20	48	4	841	0.00%	8.5/sec	1012.5
/scripts/bootstrap.min.js	2993	15	11	25	2	1010	0.00%	8.5/sec	311.3
/scripts/profiles.js	2993	8	7	15	2	144	0.00%	8.5/sec	64.3
/scripts/login.js	2993	9	6	16	1	230	0.00%	8.5/sec	19.5
/scripts/cookieconsent.js	2993	12	10	21	3	296	0.00%	8.5/sec	290.0
/images/login.jpg	2993	155	144	276	28	819	0.00%	8.6/sec	9649.0
/security/login.do	2992	323	178	778	8	7859	0.00%	8.6/sec	53.6
/_spring_security_check	2993	788	584	1646	23	5717	0.00%	8.6/sec	100.7
/images/link_broken.png	2993	57	50	107	9	304	0.03%	8.6/sec	2807.1
/participationUserList.do	2977	371	246	799	16	4472	0.00%	8.6/sec	76.8
/groupOnDisplay.do	2981	395	267	855	15	4670	0.00%	8.6/sec	76.8
/j_spring_security_logout	2950	391	257	826	16	5179	0.00%	8.6/sec	93.9
TOTAL	56702	197	146	516	1	7956	0.00%	181.3/sec	1597.1

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 2450 ms que supera por mucho los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 9667 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



Por lo que, en este caso de uso, los usuarios permitidos serán entre 90 y 100.

Caso de uso 071: Editar una participación

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de mis participaciones y editamos una.

Comenzamos probando con 90 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	kB/sec
/	8100	195	180	386	11	3456	0.00%	22.8/sec	245.8
/styles/circle.css	2700	8	7	16	1	120	0.00%	7.7/sec	3.7
/styles/cookieconsent.mi...	2700	8	7	15	1	178	0.00%	7.7/sec	31.9
/scripts/jquery.min.js	2700	19	14	36	4	342	0.00%	7.7/sec	733.9
/styles/style.css	2700	8	7	16	1	219	0.00%	7.7/sec	12.0
/styles/bootstrap.min.css	2700	21	16	42	4	205	0.00%	7.7/sec	914.9
/scripts/bootstrap.min.js	2700	12	10	23	2	123	0.00%	7.7/sec	281.3
/scripts/helpers.js	2700	9	7	16	2	336	0.00%	7.7/sec	58.1
/scripts/helpers.js	2700	9	7	17	1	356	0.00%	7.7/sec	16
/scripts/cookieconsent.js	2700	12	9	22	2	404	0.00%	7.7/sec	284.7
/images/logo.png	2700	122	110	223	24	558	0.00%	7.7/sec	8694.7
/security/login.do	2700	119	98	242	7	1718	0.00%	7.7/sec	48.1
/_spring_security_check	2700	303	249	575	18	3099	0.00%	7.7/sec	90.5
/images/link_broken.png	2700	45	35	91	9	350	0.00%	7.7/sec	2516.3
/participationUserList.do	5400	170	139	333	12	2163	0.00%	15.3/sec	136.0
/participationUserEdit.do	5400	227	172	471	8	2666	0.00%	15.4/sec	130.0
/_spring_security_logout	2700	193	158	367	13	2717	0.00%	7.7/sec	84.0
TOTAL	56700	108	40	297	1	3456	0.00%	157.5/sec	13883.0



En esta tabla, el rendimiento es de 9452 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% a pesar de tener unos tiempos admisibles.



La configuración en esta ocasión es de 100 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8000	311	239	629	11	2911	0.00%	22.2/sec	242.1
/static/circle.css	3000	9	7	17	1	119	0.00%	7.6/sec	3.6
/static/cookieconsent.js	3000	9	7	17	1	249	0.00%	7.6/sec	31.3
/scripts/jquery.min.js	3000	20	15	38	3	311	0.00%	7.6/sec	720.2
/static/style.css	3000	9	7	17	1	196	0.00%	7.6/sec	11.0
/static/bootstrap.min.css	3000	23	18	45	4	332	0.00%	7.6/sec	897.7
/scripts/bootstrap.min.js	3000	13	10	25	2	253	0.00%	7.6/sec	276.0
/scripts/helpers.js	3000	9	7	17	1	272	0.00%	7.6/sec	31.0
/scripts/helpers.js	3000	9	7	17	1	274	0.00%	7.6/sec	17.3
/scripts/cookieconsent.js	3000	12	10	23	2	236	0.00%	7.6/sec	259.7
/images/logo.jpg	3000	141	123	256	23	899	0.00%	7.6/sec	8531.4
/security/login.do	3000	229	148	504	7	2675	0.00%	7.6/sec	47.2
/_spring_security_check	3000	531	439	1034	20	3888	0.00%	7.6/sec	88.8
/images/link_broken.png	3000	49	40	98	8	287	0.00%	7.6/sec	2471.0
/participationUserList.do	6000	287	220	676	12	4633	0.00%	15.0/sec	133.8
/participationUserEdit.do	6000	390	295	819	10	4789	0.00%	15.0/sec	127.8
/_spring_security_logout	3000	294	227	590	12	3343	0.00%	7.6/sec	82.6
TOTAL	63000	173	54	478	1	4788	0.00%	155.2/sec	13078.9

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1660 ms que supera por mucho los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 9313 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



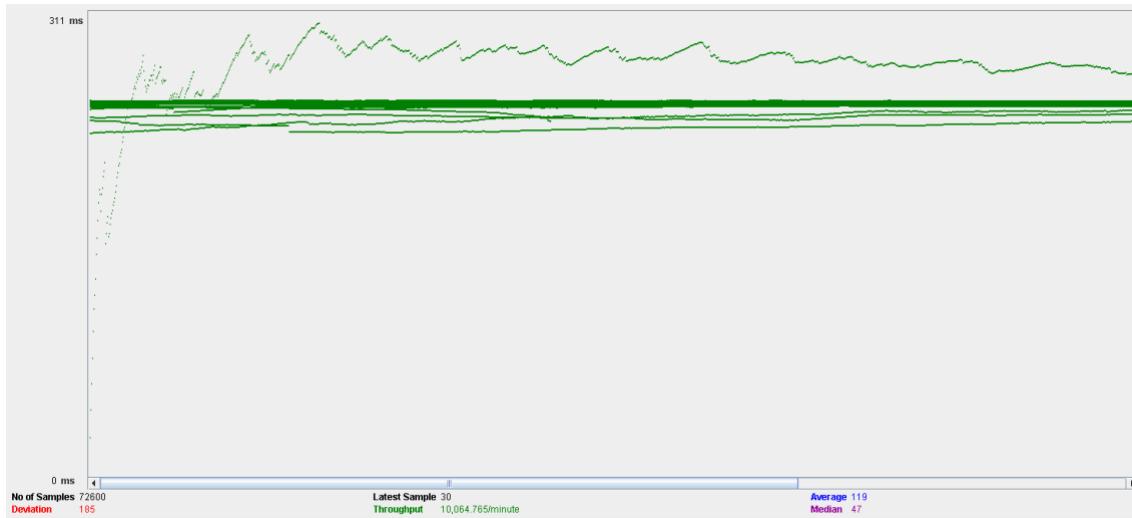
Por lo que, en este caso de uso, los usuarios permitidos serán entre 90 y 100.

Caso de uso 072: Crear una participación

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de conjuntas, vamos al display de una donde no tengamos una participación creada y participamos en ella.

Comenzamos probando con 110 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/styles/bootstrap.min.css	9899	202	148	421	12	3268	0.00%	23.1/sec	251.8
/styles/bootstrap.min.css	3300	28	20	55	5	1128	0.00%	7.8/sec	929.9
/styles/cookieconsent.mi...	3300	12	8	24	1	238	0.00%	7.8/sec	32.4
/styles/cookieconsent.mi...	3300	11	7	24	0	269	0.00%	7.8/sec	12.3
/styles/circle.css	3300	12	8	24	1	381	0.00%	7.8/sec	3.7
/styles/circle.css	3300	25	18	49	4	310	0.00%	7.8/sec	74.0
/scripts/bootstrap.min.js	3300	16	11	32	2	216	0.00%	7.8/sec	205.9
/scripts/bootstrap.min.js	3300	11	8	23	1	221	0.00%	7.8/sec	59.0
/scripts/cookieconsent.js	3300	16	11	32	2	340	0.00%	7.8/sec	269.0
/scripts/helpers.js	3300	11	8	23	1	189	0.00%	7.8/sec	17.9
/images/logo.jpg	3300	158	138	297	24	785	0.00%	7.8/sec	8836.6
/images/link_broken.png	3300	62	49	125	8	695	0.00%	7.8/sec	2555.1
/security/login.do	3300	147	92	328	6	2602	0.00%	7.8/sec	49.8
/security/empty_check	3300	358	267	745	9	3016	0.00%	7.8/sec	11.0
/group/list.do	3300	192	136	419	13	2460	0.00%	7.8/sec	86.4
/group/display.do	3300	192	137	421	12	2658	0.00%	7.8/sec	69.2
/participationUser/create	3300	176	119	380	11	2096	0.00%	7.8/sec	114.3
/participationUser/edit	3300	185	138	387	14	3705	0.00%	7.8/sec	58.7
/participationUser/list	3299	193	134	413	12	3288	0.00%	7.9/sec	63.0
/spring_security_logout	3299	210	157	431	14	2256	0.00%	7.9/sec	86.0
TOTAL	72597	119	47	304	0	38181	0.00%	168.3/sec	14275.3



En esta tabla, el rendimiento es de 10064 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

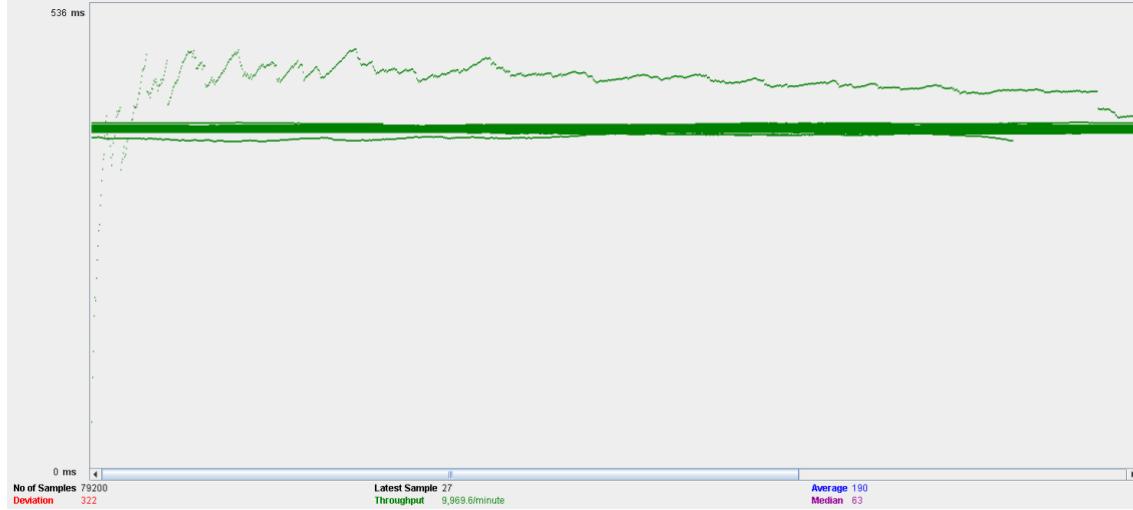
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% a pesar de tener unos tiempos admisibles.



La configuración en esta ocasión es de 120 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	10800	330	208	761	11	4648	0.00%	22.7/sec	247.4
/styles/bootstrap.min.css	3600	32	24	62	4	659	0.00%	7.7/sec	917.8
/styles/cookieconsent.mi	3600	13	9	27	1	449	0.00%	7.7/sec	32.0
/styles/style.css	3600	13	8	26	1	1003	0.00%	7.7/sec	12.1
/styles/circle.css	3600	12	8	26	1	244	0.00%	7.7/sec	3.7
/scripts/jquery.min.js	3600	27	21	55	4	361	0.00%	7.7/sec	736.3
/scripts/bootstrap.min.js	3600	18	13	37	2	375	0.00%	7.7/sec	20.2
/scripts/cookieconsent.js	3600	13	9	29	1	467	0.00%	7.7/sec	69.3
/scripts/helpers.js	3600	17	12	35	2	335	0.00%	7.7/sec	265.6
/images/logo.jpg	3600	13	9	26	1	290	0.00%	7.7/sec	17.7
/images/link_broken.png	3600	171	152	325	23	1135	0.00%	7.7/sec	8720.8
/security/login.do	3600	69	56	136	8	700	0.00%	7.7/sec	2521.6
/security/logout.do	3600	273	145	680	5	5330	0.00%	7.7/sec	48.3
/j_spring_security_check	3600	655	489	1429	20	4307	0.00%	7.7/sec	90.8
/groupedit.do	3600	315	195	733	16	5232	0.00%	7.7/sec	62.7
/groupondisplay.do	3600	316	195	732	13	4016	0.00%	7.7/sec	67.4
/participationUserCreate	3600	269	170	679	11	3455	0.00%	7.7/sec	113.0
/participationUserEdit.do	3600	315	198	715	12	2996	0.00%	7.7/sec	58.0
/participationUserList.do	3600	306	194	711	11	3525	0.00%	7.7/sec	61.8
/j_spring_security_logout	3600	321	208	734	15	3777	0.00%	7.7/sec	84.3
TOTAL	79200	190	63	518	1	5330	0.00%	166.2/sec	14091.2

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 2200 ms que supera por mucho los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 9969 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abriremos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



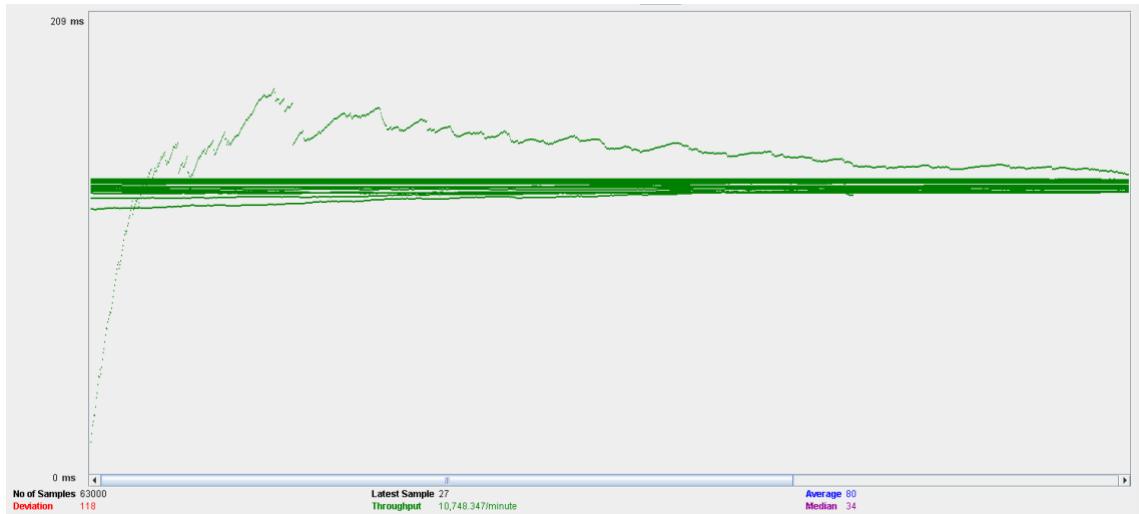
Por lo que, en este caso de uso, los usuarios permitidos serán entre 110 y 120.

Caso de uso 073: Borrar una participación

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de mis participaciones, editamos una y la borramos.

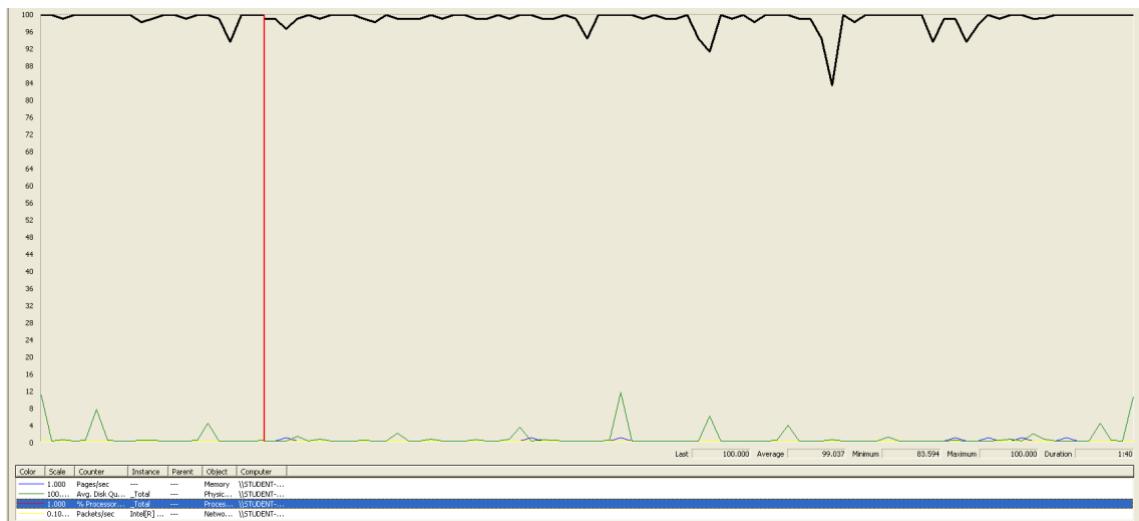
Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9000	139	102	287	11	2078	0.00%	25.0/sec	279.4
/styles/cookieconsent.min.js	9000	12	8	24	1	31	0.00%	8.0/sec	30.4
/styles/bootstrap.min.css	9000	28	20	57	5	479	0.00%	8.0/sec	1043.3
/scripts/jquery.min.js	9000	23	17	47	4	345	0.00%	8.0/sec	837.1
/styles/style.css	3000	10	7	21	1	265	0.00%	8.0/sec	13.7
/styles/circle.css	3000	11	7	22	1	432	0.00%	8.0/sec	4.2
/scripts/bootstrap.min.js	3000	15	11	32	2	219	0.00%	8.0/sec	320.9
/scripts/polyfills.js	3000	11	7	23	1	278	0.00%	8.0/sec	66.2
/scripts/usernameclient.js	3000	15	10	30	2	670	0.00%	8.0/sec	30.0
/scripts/helpers.js	3000	11	7	27	1	507	0.00%	8.0/sec	20.1
/images/logo.png	3000	152	131	294	24	1026	0.00%	0.0/sec	9819.1
/images/link_broken.png	3000	49	46	117	8	517	0.00%	8.0/sec	2866.7
/secure/login.do	3000	91	56	202	7	1656	0.00%	8.0/sec	54.8
/_spring_security_check	3000	216	157	454	19	1816	0.00%	8.0/sec	103.1
/participation/user/list.do	6000	124	91	260	13	1824	0.00%	17.4/sec	148.8
/participation/user/edit.do	6000	105	72	222	9	1660	0.00%	17.5/sec	195.0
/_spring_security_logout	3000	142	108	283	14	1948	0.00%	8.0/sec	95.6
TOTAL	63000	80	34	209	1	2078	0.00%	179.1/sec	15825.8



En esta tabla, el rendimiento es de 10748 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

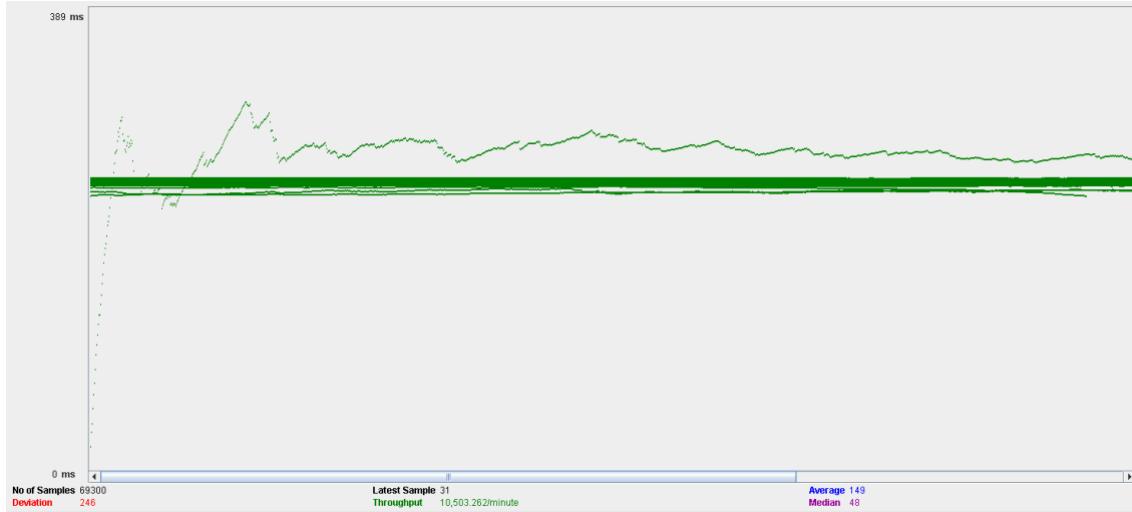
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU está entre el 80% y el 100%.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8600	277	191	633	11	3283	0.00%	25.0/sec	273.1
/styles/cookieconsent.min.css	3200	44	9	31	1	209	0.00%	8.6/sec	26.3
/styles/bootstrap.min.css	3200	31	20	70	4	295	0.00%	8.5/sec	1012.6
/scripts/query.min.js	3200	27	18	59	3	381	0.00%	8.5/sec	812.3
/styles/style.css	3200	12	8	28	1	255	0.00%	8.5/sec	13.3
/styles/circle.css	3200	12	8	27	1	208	0.00%	8.5/sec	4.1
/scripts/bootstrap.min.js	3200	11	41	41	2	261	0.00%	8.5/sec	311.4
/scripts/polls.js	3200	13	8	50	1	266	0.00%	8.5/sec	64.3
/scripts/cookieconsent.js	3200	17	12	39	2	234	0.00%	8.5/sec	293.0
/scripts/heplers.js	3200	12	6	28	1	211	0.00%	8.5/sec	19.5
/images/logo.png	3200	167	113	363	23	999	0.00%	8.5/sec	9621.9
/images/link_broken.png	3200	68	45	150	8	428	0.00%	8.5/sec	2782.3
/security/login.do	3200	204	128	463	6	2706	0.00%	8.5/sec	53.3
/_spring_security_check	3200	473	357	1025	20	4281	0.00%	8.5/sec	100.2
/participationUserList.do	6600	249	171	665	12	3953	0.00%	16.9/sec	145.0
/participationUserEdit.do	6600	224	146	512	9	3145	0.00%	17.1/sec	189.9
/_spring_security_logout	3200	266	197	656	9	5610	0.00%	8.5/sec	93.1
TOTAL	69300	149	48	411	1	5610	0.00%	175.1/sec	15404.9

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1650 ms que supera por mucho los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 10503 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



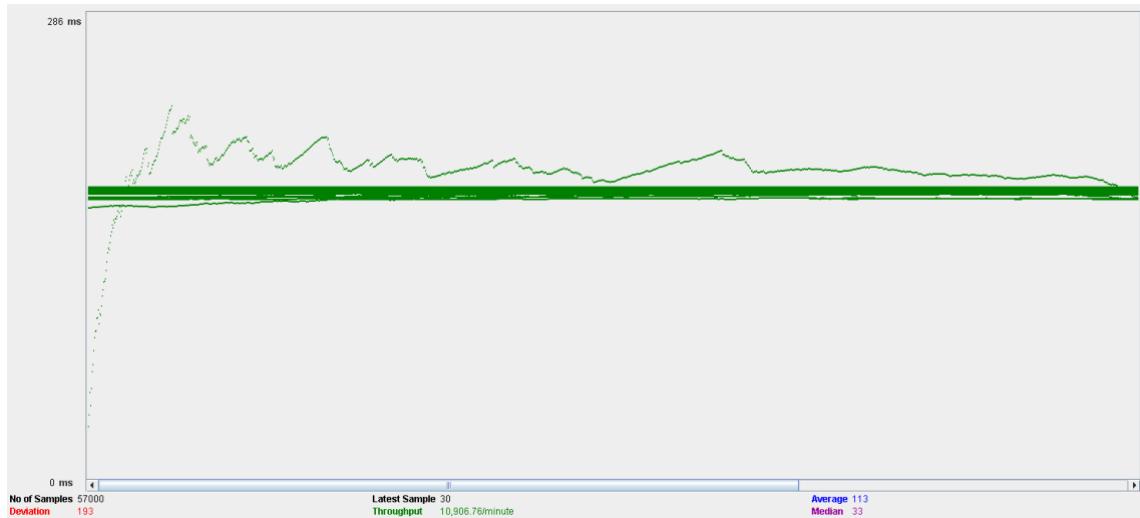
Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 074-075: Listar conjuntas y desplegarlas

En este caso de uso, accedemos a la página, nos autenticamos como administrador, y accedemos a la lista de conjuntas que puede ver el actor autenticado y desplegamos una.

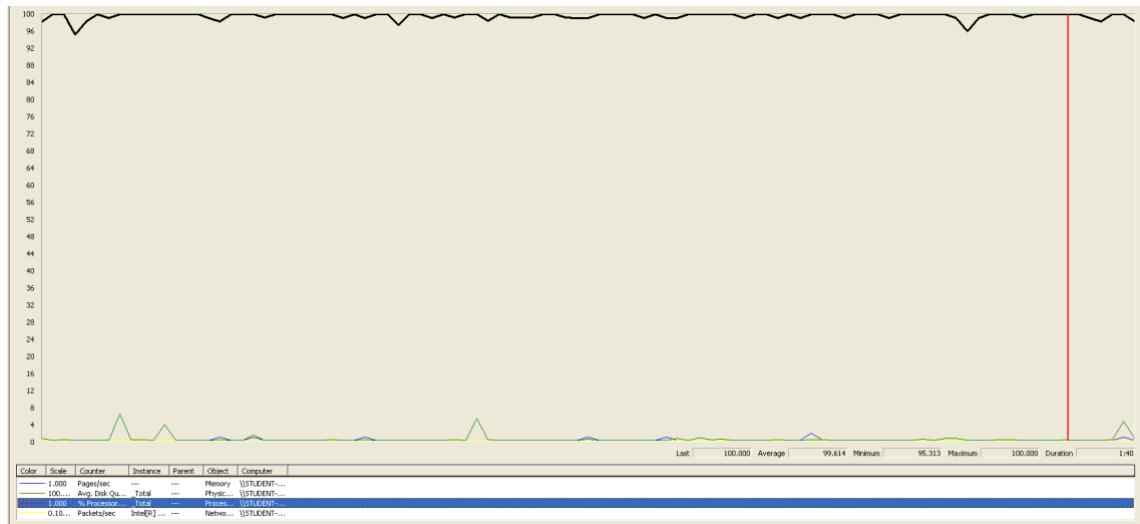
Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9000	226	170	449	10	2870	0.00%	28.7/sec	310.6
/styles/bootstrap.min.css	3000	25	19	49	4	305	0.00%	9.8/sec	1163.0
/scripts/jquery.min.js	3000	21	16	41	4	410	0.00%	9.8/sec	933.7
/styles/cookieconsent.mi...	3000	9	7	18	2	289	0.00%	9.8/sec	40.6
/styles/style.css	3000	9	6	18	1	914	0.00%	9.8/sec	15.3
/styles/circle.css	3000	9	7	18	1	243	0.00%	9.8/sec	4.7
/scripts/fontawesome.min.js	3000	14	10	28	2	337	0.00%	9.8/sec	367.9
/scripts/fontawesome-all.js	3000	9	7	19	1	284	0.00%	9.8/sec	373.0
/scripts/fontawesome.js	3000	9	7	18	1	213	0.00%	9.8/sec	22.4
/scripts/cookieconsent.js	3000	13	10	24	2	195	0.00%	9.8/sec	336.9
/images/logo.jpg	3000	136	114	260	24	1242	0.00%	9.8/sec	11064.6
/image/link_broken.png	3000	53	42	107	8	326	0.00%	9.8/sec	3200.7
/security/login	3000	165	108	341	6	2580	0.00%	9.8/sec	61.2
/j_spring_security_check	370	285	738	17	4353	0.00%	9.8/sec		112.4
/group/add.do	3000	191	142	377	13	293	0.00%	9.8/sec	114
/group/display.do	3000	192	139	367	9	2814	0.00%	9.8/sec	83.0
/j_spring_security_logout	3000	235	175	463	14	2512	0.00%	9.8/sec	107.1
TOTAL	57000	113	33	305	1	4353	0.00%	181.8/sec	17547.8



En esta tabla, el rendimiento es de 10906 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente descendiendo hasta estabilizarse.

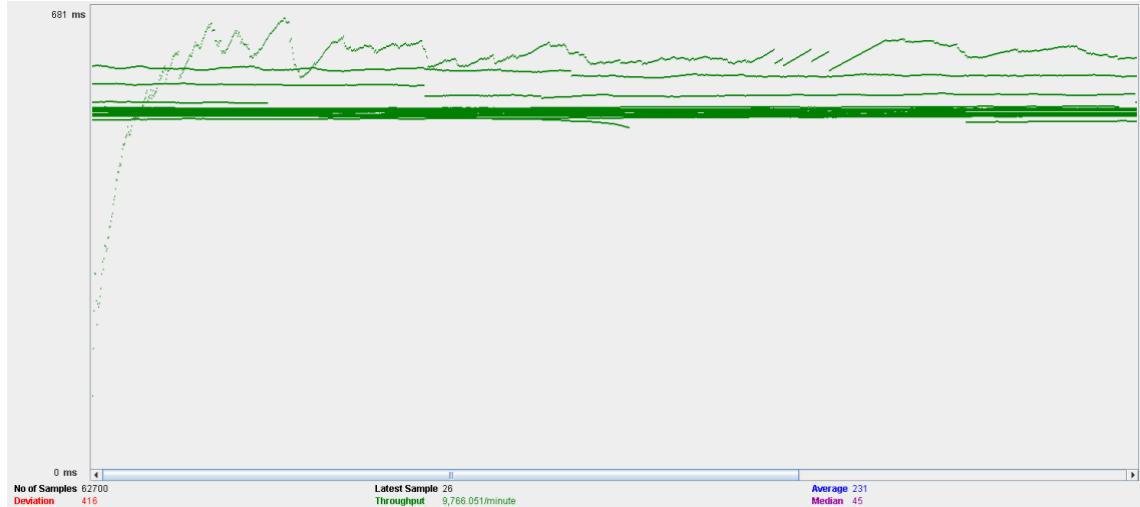
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% aun siendo admisibles los tiempos.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9900	468	327	1044	11	5105	0.00%	25.7/sec	279.2
/styles/bootstrap.min.css	3300	22	26	63	4	440	0.00%	8.7/sec	1037.7
/scripts/jquery.min.js	3300	26	20	63	3	382	0.00%	8.7/sec	832.5
/scripts/cookieconsent.mi...	3300	12	9	23	1	670	0.00%	8.7/sec	36.2
/styles/style.css	3300	11	9	23	1	224	0.00%	8.7/sec	13.7
/styles/circle.css	3300	12	9	23	1	263	0.00%	8.7/sec	4.2
/scripts/bootstrap.min.js	3300	17	13	34	2	609	0.00%	8.7/sec	30.1
/scripts/fontawesome.js	3300	12	9	23	1	312	0.00%	8.7/sec	65.9
/scripts/fontawesome.s...	3300	11	9	22	1	375	0.00%	8.7/sec	20.0
/scripts/cookieconsent.js	3300	16	13	30	2	264	0.00%	8.7/sec	300.3
/images/logo.jpg	3300	172	149	327	24	866	0.00%	8.7/sec	9862.5
/images/link_broken.png	3300	71	57	140	8	656	0.00%	8.7/sec	2851.7
/security/login.do	3300	390	228	954	6	4942	0.00%	0.8/sec	54.6
/j_spring_security_check	3300	866	707	1855	17	4886	0.00%	0.8/sec	100.2
/grouponlist.do	3300	435	285	1021	12	4879	0.00%	0.8/sec	91.2
/groupondisplay.do	3300	412	254	986	8	4362	0.00%	0.8/sec	73.0
/j_spring_security_logout	3300	478	334	1053	3	4904	0.00%	0.8/sec	95.4
TOTAL	62700	231	45	685	1	5105	0.00%	162.8/sec	16712.5

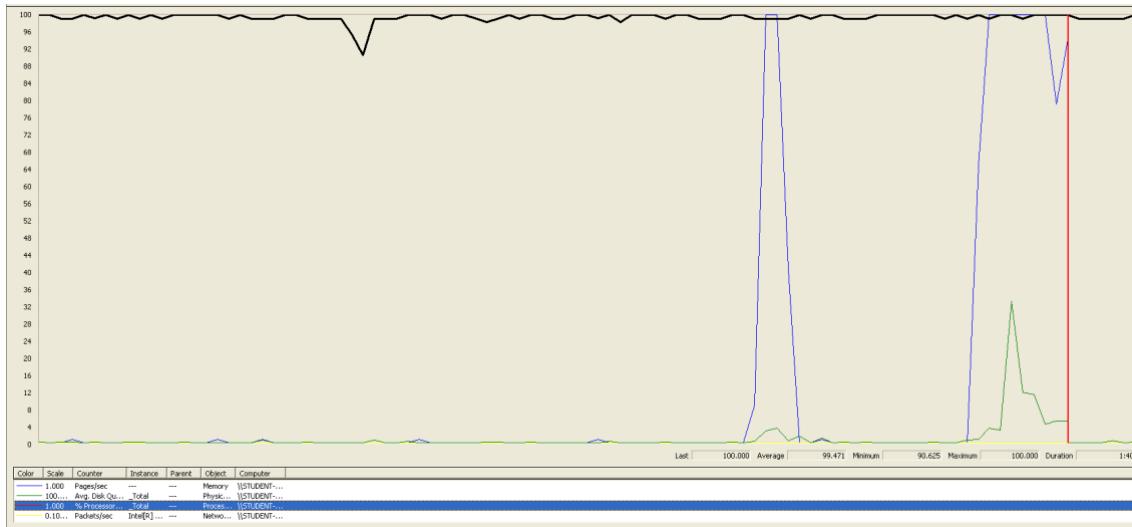
Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 2900 ms que supera por mucho los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 9766 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente y en algunas ocasiones la memoria, por lo que el número de usuarios concurrentes vienen limitados por la CPU y la memoria.



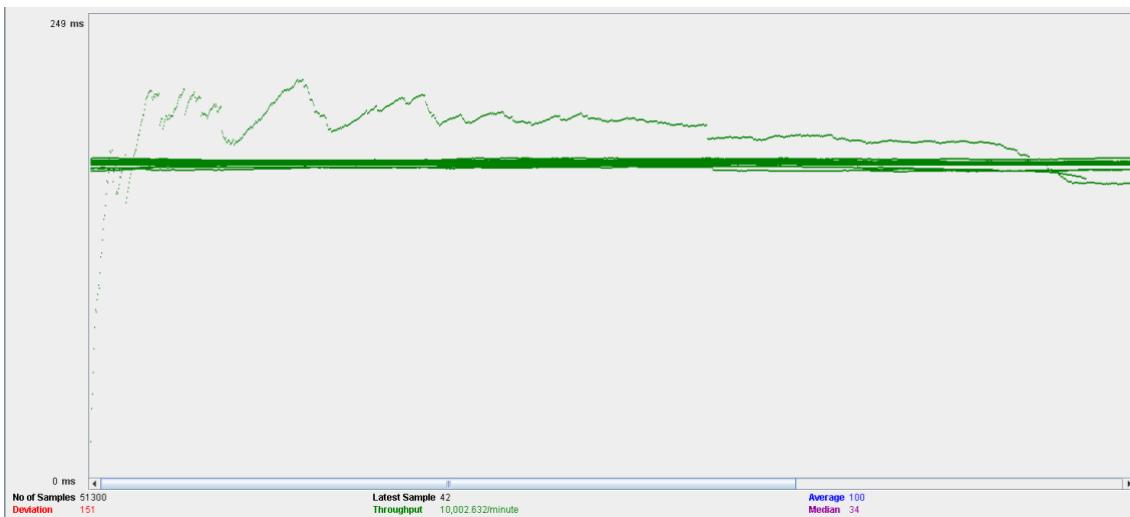
Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 076: Listar conjuntas de un usuario y desplegarlas

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de mis conjuntas y desplegamos una.

Comenzamos probando con 90 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	192	143	427	11	1990	0.00%	26.3/sec	287.4
/abiertobdstrap.min.css	2700	31	23	67	4	375	0.00%	0/sec	100.5
/abiertocookieconsent.mi...	2700	19	9	26	1	393	0.00%	0/sec	37.2
/styles/style.css	2700	11	9	26	1	337	0.00%	0/sec	14.0
/scripts/jquery.min.js	2700	27	19	66	3	370	0.00%	9/sec	865.6
/scripts/bootstrap.min.js	2700	17	12	34	2	239	0.00%	9/sec	328.0
/scripts/circle.css	2700	12	8	26	1	191	0.00%	9/sec	4.3
/scripts/polyfills.js	2700	12	8	25	1	251	0.00%	9/sec	67.7
/scripts/helpers.js	2700	12	8	26	1	322	0.00%	9/sec	20.5
/scripts/assessment.js	2700	16	12	35	2	246	0.00%	9/sec	30.6
/images/logo.png	2700	169	139	339	23	686	0.00%	9/sec	10135.2
/images/link_broken.png	2700	70	51	149	0	483	0.00%	9/sec	2931.2
/securelogin.do	2700	130	86	299	7	1597	0.00%	9/sec	56.1
/j_spring_security_check	2700	292	223	640	19	1950	0.00%	9/sec	105.6
/groupon/userlist.do	2700	168	121	373	12	1881	0.00%	9/sec	105.6
/groupon/display.do	2700	142	105	313	10	1258	0.00%	9/sec	80.1
/j_spring_security_logout	2700	194	149	416	14	1611	0.00%	9/sec	98.2
TOTAL	51300	100	34	280	1	1990	0.00%	166.7/sec	16113.7



En esta tabla, el rendimiento es de 10002 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

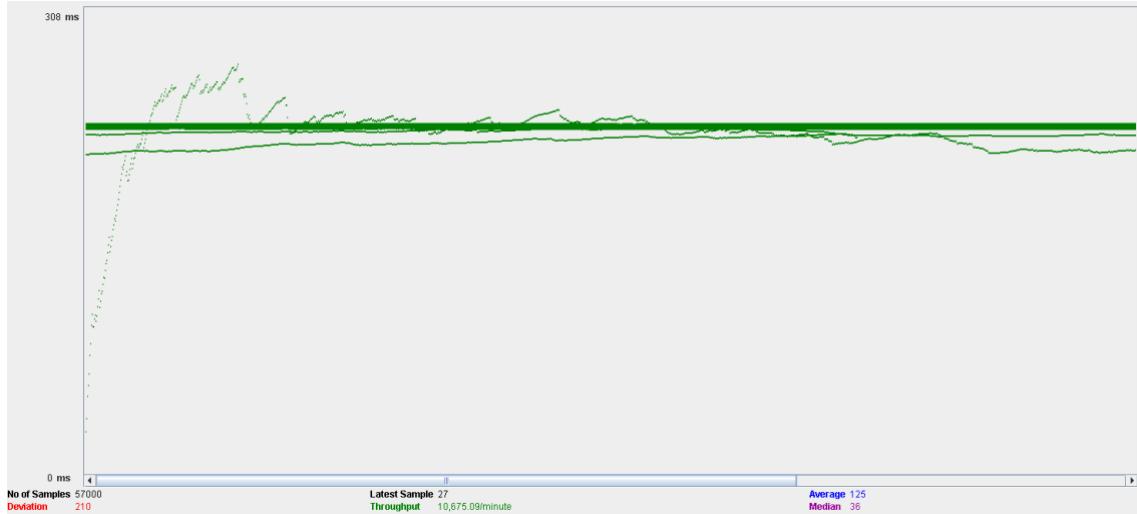
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% aun siendo admisibles los tiempos.



La configuración en esta ocasión es de 100 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8000	253	193	508	12	2772	0.00%	28.1/sec	306.8
/static/bootstrap.min.css	3000	26	21	47	4	360	0.00%	9.6/sec	1140.6
/static/cookieconsent.js	3000	9	7	17	1	155	0.00%	9.6/sec	39.7
/static/style.css	3000	9	7	17	1	264	0.00%	9.6/sec	15.0
/scripts/query.min.js	3000	21	18	39	3	273	0.00%	9.6/sec	915.1
/scripts/bootstrap.min.js	3000	13	11	25	2	306	0.00%	9.6/sec	350.8
/scripts/circle.css	3000	9	7	17	1	201	0.00%	9.6/sec	4.6
/scripts/circle.js	3000	9	7	17	1	189	0.00%	9.6/sec	72.4
/scripts/helpers.js	3000	9	7	16	1	21	0.00%	9.6/sec	21.0
/scripts/cookieconsent.js	3000	13	10	23	2	184	0.00%	9.6/sec	330.2
/images/logo.jpg	3000	147	140	262	23	541	0.00%	9.6/sec	10944.3
/imagesthink_broken.png	3000	57	52	107	8	324	0.00%	9.6/sec	3135.3
/security/login.do	3000	180	120	392	7	2684	0.00%	9.6/sec	60.0
/j_spring_security_check	3000	440	328	901	20	3516	0.00%	9.6/sec	112.8
/groupOnUserList.do	3000	220	156	446	12	2789	0.00%	9.6/sec	112.7
/groupOnDisplay.do	3000	214	143	430	9	2676	0.00%	9.6/sec	85.6
/j_spring_security_logout	3000	241	187	462	15	2217	0.00%	9.6/sec	104.8
TOTAL	57000	125	36	332	1	3916	0.00%	177.9/sec	17197.0

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1400 ms que supera por bastante los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 10675 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



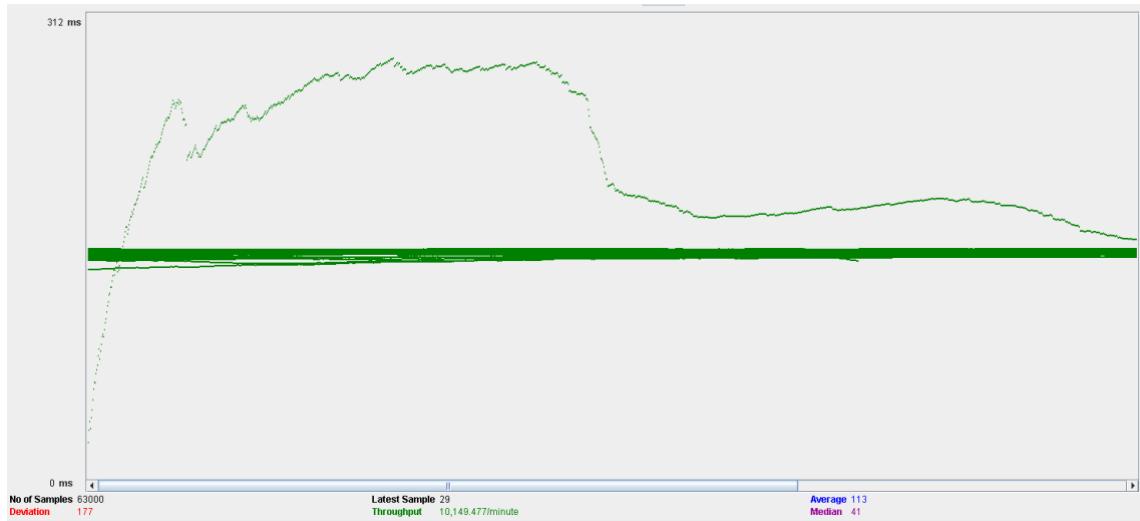
Por lo que, en este caso de uso, los usuarios permitidos serán entre 90 y 100.

Caso de uso 077: Editar una conjunta

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de mis conjuntas y editamos una.

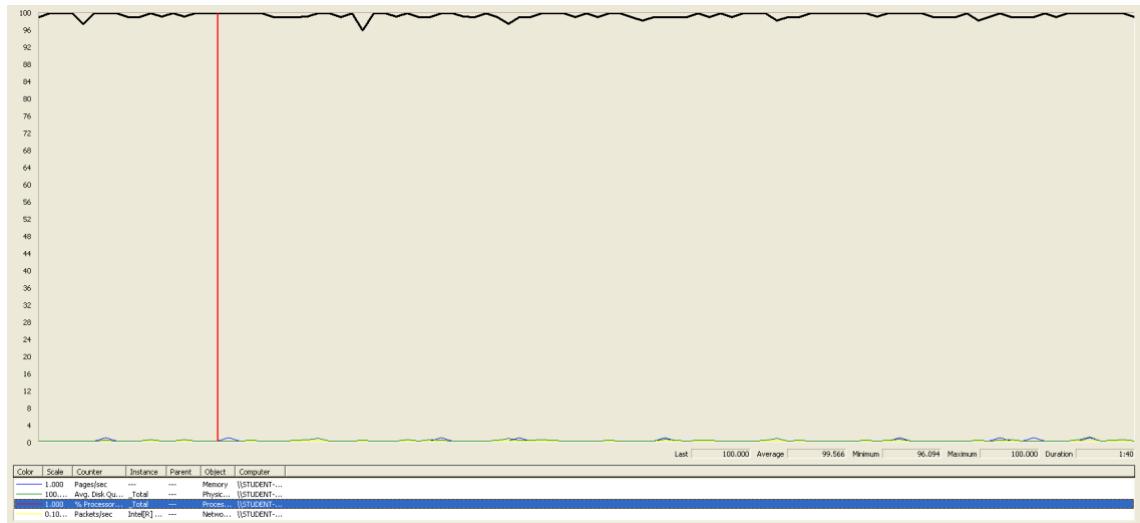
Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8000	199	144	400	11	3167	0.00%	24.2/sec	263.9
/scripts/green/min.js	30000	21	16	43	4	374	0.00%	8.3/sec	787.8
/abienlistable.css	30000	10	7	19	1	201	0.00%	8.3/sec	12.9
/scripts/cookieconsent.mi...	30000	9	7	19	1	443	0.00%	8.3/sec	4.2
/scripts/bootstrap.min.js	30000	14	11	28	2	408	0.00%	8.3/sec	302.0
/scripts/circle.css	30000	9	7	18	1	148	0.00%	8.3/sec	4.0
/scripts/bootstrap.min.css	30000	24	18	47	3	388	0.00%	8.3/sec	982.3
/scripts/polyfills.js	30000	10	7	19	1	256	0.00%	8.3/sec	62.3
/scripts/helpers.js	30000	10	7	20	1	191	0.00%	8.3/sec	18.9
/scripts/cookieconsent.js	30000	13	10	25	2	175	0.00%	8.3/sec	284.2
/scripts/login.js	30000	142	117	269	24	525	0.00%	8.3/sec	913.6
/securelogin.do	30000	125	90	287	7	1086	0.00%	8.3/sec	51.6
/j_spring_security_check	30000	311	241	628	20	4268	0.00%	8.3/sec	97.1
/images/link_broken.png	30000	51	39	104	8	475	0.00%	8.3/sec	2099.7
/groupOnUserList.do	6000	164	119	340	12	2926	0.00%	16.4/sec	192.4
/groupOnUserEdit.do	6000	242	182	503	11	2693	0.00%	16.5/sec	182.8
/j_spring_security_logout	30000	195	148	392	14	2258	0.00%	8.3/sec	90.2
TOTAL	63000	113	41	301	1	4268	0.00%	169.2/sec	14993.8



En esta tabla, el rendimiento es de 10002 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

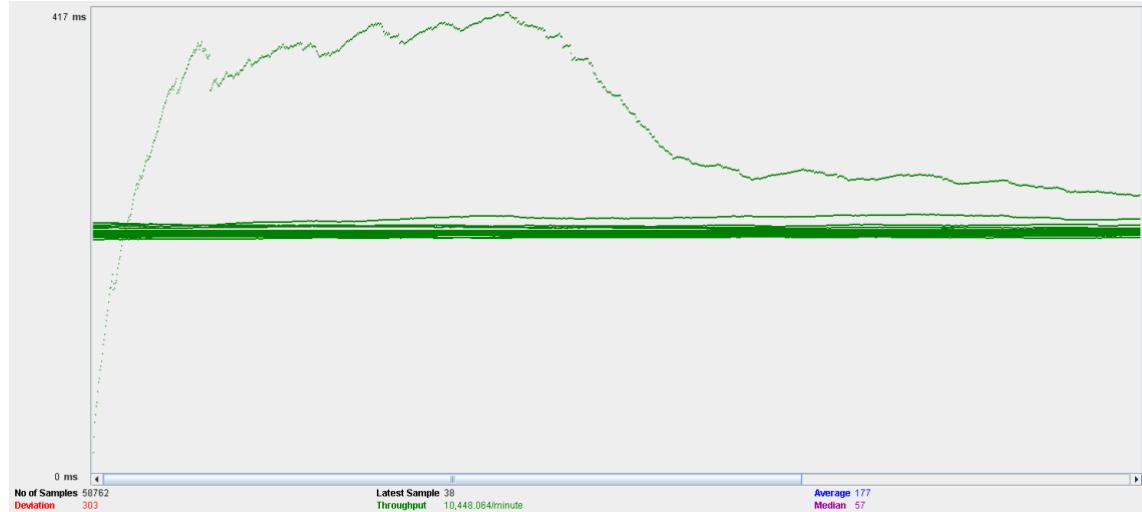
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% aun siendo admisibles los tiempos.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8346	298	206	623	13	4978	0.04%	24.7/sec	270.1
/scripts/jquery.min.js	2831	24	17	47	4	2724	0.00%	8.4/sec	800.1
/styles/style.css	2831	10	7	21	1	797	0.00%	8.4/sec	13.1
/styles/cookieconsent.min.js	2830	10	7	20	1	165	0.00%	8.4/sec	34.7
/scripts/bootstrap.min.js	2829	16	11	32	2	298	0.00%	8.4/sec	306.4
/styles/circle.css	2829	11	7	22	1	332	0.00%	8.4/sec	4.0
/scripts/blurtrap.min.css	2829	19	51	333	4	333	0.00%	8.4/sec	90.7
/scripts/helpers.js	2829	10	7	22	1	296	0.00%	8.4/sec	63.8
/scripts/cookieconsent.js	2829	10	7	21	1	339	0.00%	8.4/sec	19.2
/images/logo.jpg	2829	14	10	28	2	181	0.00%	8.4/sec	289.4
/security/login.do	2810	253	145	575	7	5060	0.04%	8.4/sec	52.4
/_spring_security_check	2792	544	403	1119	22	4958	0.04%	8.4/sec	99.3
/images/link_broken.png	2788	62	46	120	8	1877	0.00%	8.4/sec	2735.2
/grouponuseinfist.do	6371	282	164	604	14	5202	0.00%	19.7/sec	195.3
/grouponuseredit.do	6494	410	290	876	13	5179	0.00%	16.7/sec	184.1
/_spring_security_logout	2726	312	214	644	15	4131	0.00%	8.3/sec	90.7
TOTAL	58762	177	57	478	1	5202	0.02%	174.1/sec	16545.6

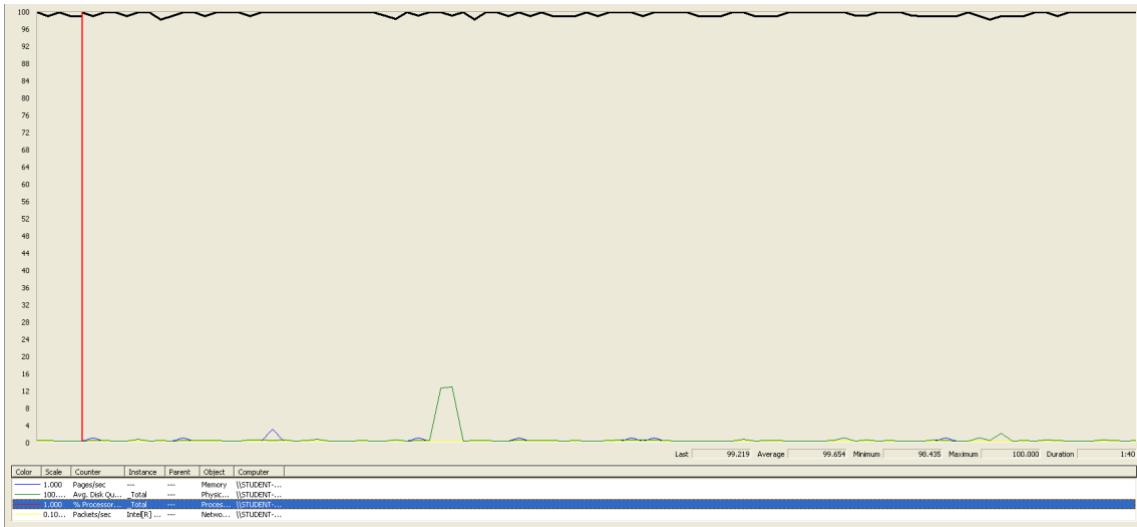
Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1740 ms que supera por bastante los 1000 ms que se consideran aceptables, además hay errores en la ejecución.



En esta tabla, el rendimiento es de aproximadamente 10448 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente por lo que el número de usuarios concurrentes vienen limitados por la CPU.



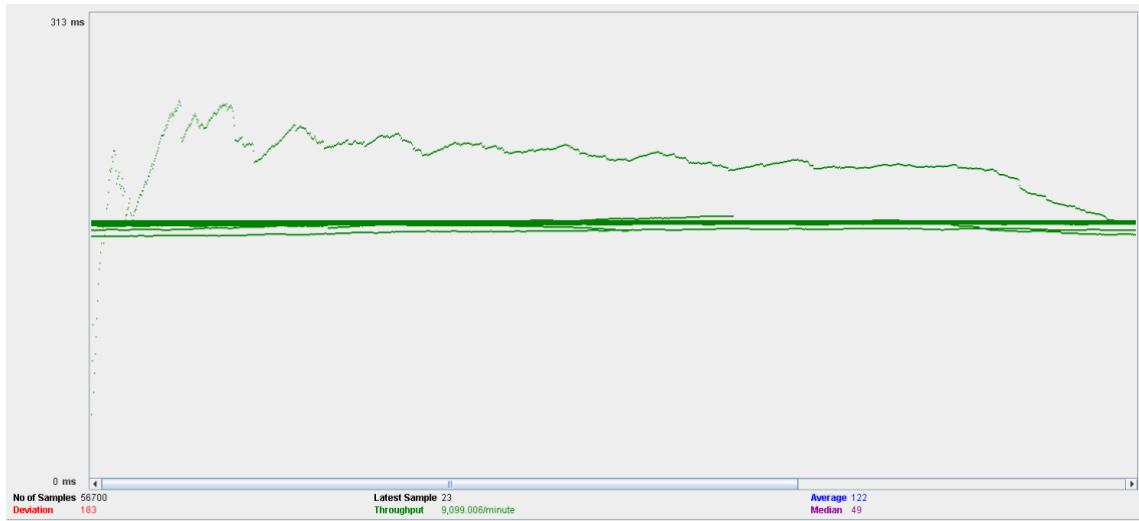
Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 078: Crear una conjunta

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de mis conjuntas y creamos una.

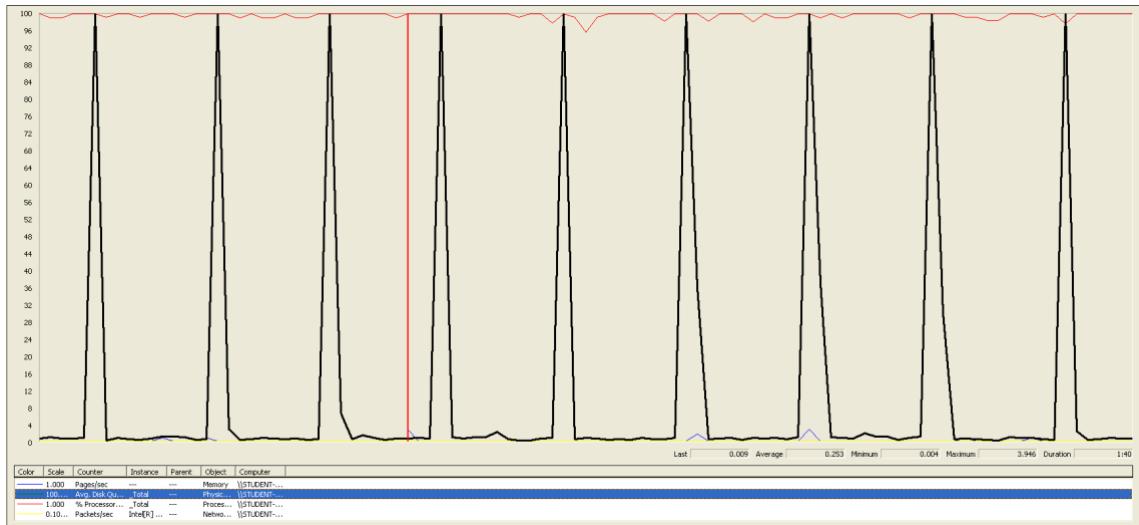
Comenzamos probando con 90 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	81000	215	170	414	12	3236	0.00%	21.7/sec	236.6
/styles/style.css	27000	9	7	20	1	250	0.00%	7.4/sec	11.6
/styles/cookieconsent.mi...	27000	10	7	20	1	510	0.00%	7.4/sec	30.7
/scripts/query.min.js	27000	22	16	43	3	1505	0.00%	7.4/sec	705.6
/styles/circle.css	27000	9	7	19	1	283	0.00%	7.4/sec	3.5
/styles/bootstrap.min.css	27000	25	19	50	4	284	0.00%	7.4/sec	879.6
/scripts/bootstrap.min.js	27000	14	10	28	2	218	0.00%	7.4/sec	270.5
/scripts/fontawesome.min.js	27000	10	7	19	1	152	0.00%	7.4/sec	10.6
/scripts/helpers.js	27000	10	7	20	1	200	0.00%	7.4/sec	10.9
/scripts/cookieconsent.js	27000	14	10	27	3	398	0.00%	7.4/sec	254.5
/images/lopi.png	27000	146	130	275	25	711	0.00%	7.4/sec	8359.4
/images/link_broken.png	27000	56	46	110	8	448	0.00%	7.4/sec	2418.0
/security/login.do	27000	141	101	282	7	2117	0.00%	7.4/sec	46.2
/_spring_security_check	27000	345	280	668	13	2335	0.00%	7.4/sec	87.0
/grouponuserlist.do	54000	200	157	382	17	2524	0.00%	14.7/sec	167.7
/grouponusercreate.do	27000	165	121	324	9	2307	0.00%	7.4/sec	69.0
/grouponuseredit.do	27000	329	289	626	19	3786	0.00%	7.4/sec	632.8
/_spring_security_logout	27000	214	173	404	14	3783	0.00%	7.4/sec	80.8
TOTAL	56700	122	49	318	1	3783	0.00%	151.7/sec	15426.4



En esta tabla, el rendimiento es de 9099 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

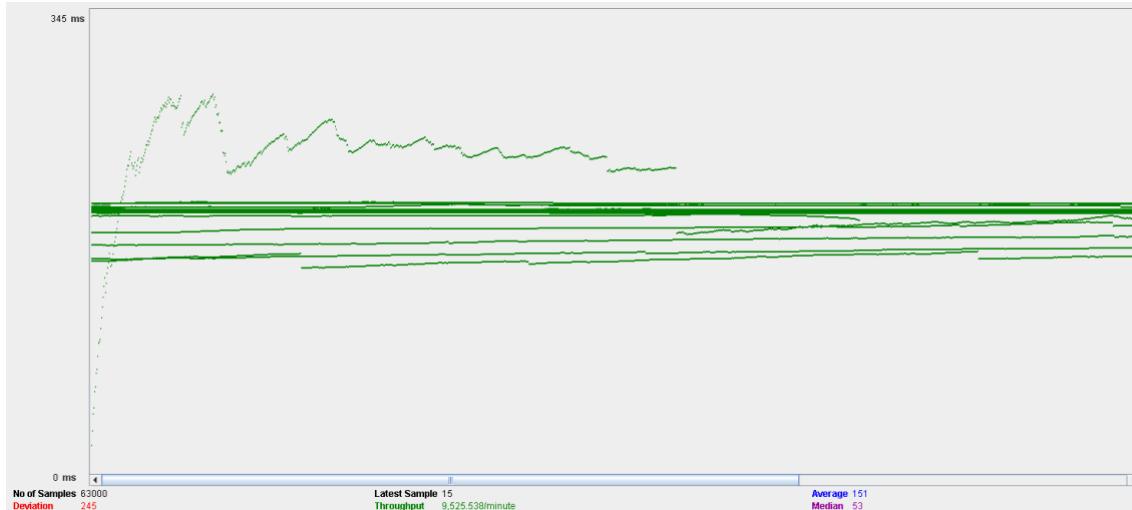
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% y el disco también sube hasta el 100% periódicamente.



La configuración en esta ocasión es de 100 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/styles/style.css	9000	263	194	538	12	3940	0.00%	22.7/sec	247.6
/styles/cookieconsent.min.js	3000	11	7	22	1	236	0.00%	7.7/sec	12.1
/scripts/jquery.min.js	3000	11	7	22	1	542	0.00%	7.7/sec	32.0
/styles/circle.css	3000	23	17	47	4	440	0.00%	7.7/sec	737.4
/scripts/main.css	3000	11	7	22	1	314	0.00%	7.7/sec	3.7
/scripts/bootstrap.min.css	3000	27	20	55	4	445	0.00%	7.7/sec	919.3
/scripts/bootstrap.min.js	3000	15	11	30	2	291	0.00%	7.7/sec	29.7
/scripts/helpers.js	3000	11	7	22	1	250	0.00%	7.7/sec	59.3
/scripts/helpers.js	3000	11	7	22	1	491	0.00%	7.7/sec	17.7
/scripts/cookieconsent.js	3000	14	10	27	2	314	0.00%	7.7/sec	266.0
/images/logo.jpg	3000	150	128	284	25	775	0.00%	7.7/sec	8736.8
/images/link_broken.png	3000	59	45	119	8	343	0.00%	7.7/sec	2526.3
/security/login.do	3000	199	130	424	7	2758	0.00%	7.7/sec	48.3
/spring_security_check	3000	470	364	900	71	410	0.00%	7.7/sec	90.0
/groupouserlist.do	8000	241	177	800	15	4845	0.00%	15.6/sec	169.0
/groupouserlist.do	3000	225	151	400	9	2882	0.00%	7.8/sec	73.4
/groupouseredit.do	3000	407	311	799	16	4955	0.00%	7.8/sec	283.0
/spring_security_logout	3000	263	200	520	16	4472	0.00%	7.8/sec	85.1
TOTAL	63000	151	53	400	1	4955	0.00%	158.8/sec	14775.9

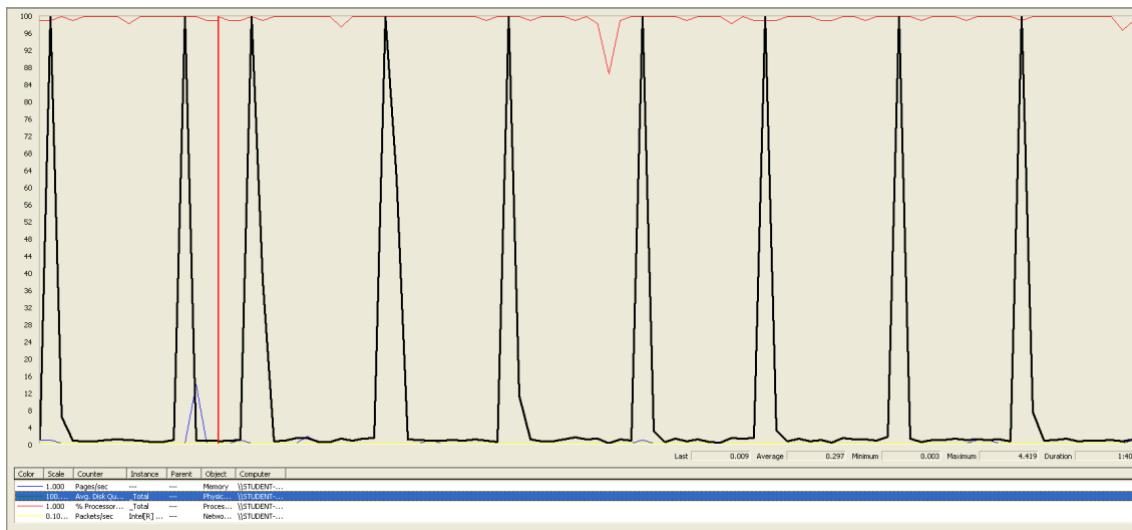
Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1480 ms que supera por bastante los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 9525 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente y el disco también llega al 100% periódicamente probablemente debido a las lecturas-escritura que está haciendo en el disco. Por lo que el número de usuarios concurrentes vienen limitados por la CPU y el disco.



Por lo que, en este caso de uso, los usuarios permitidos serán entre 90 y 100.

Caso de uso 079: Borrar una conjunta como usuario

En este caso de uso, accedemos a la página, nos autenticamos como usuario, y accedemos a la lista de mis conjuntas y editamos y borramos una.

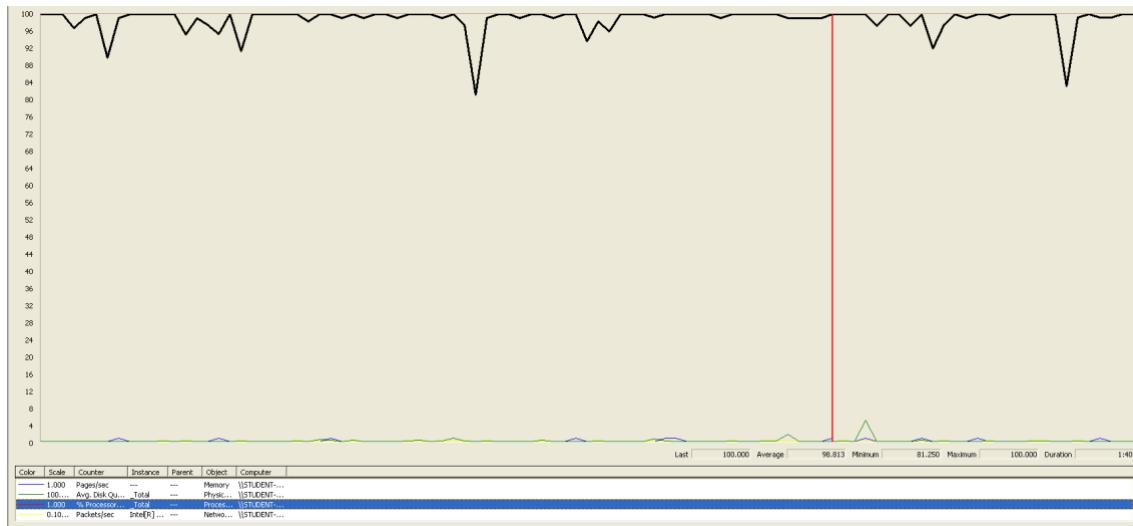
Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8000	147	108	304	12	2214	0.00%	25.4/sec	277.0
/styles/circle.css	3000	12	9	26	1	537	0.00%	8.7/sec	4.2
/scripts/jquery.min.js	3000	26	17	64	3	300	0.00%	8.7/sec	826.0
/styles/style.css	3000	11	7	24	1	420	0.00%	8.7/sec	13.6
/styles/cookieconsent.mi...	3000	11	7	24	1	196	0.00%	8.7/sec	35.9
/styles/bootstrap.min.css	3000	28	20	59	4	457	0.00%	8.7/sec	1030.9
/scripts/bootstrap.min.js	3000	17	11	36	2	319	0.00%	8.7/sec	317.0
/scripts/helpers.js	3000	11	7	25	1	478	0.00%	8.7/sec	65.4
/scripts/cookiesent.js	3000	16	11	32	2	325	0.00%	8.7/sec	29.0
/scripts/helpers.js	3000	12	7	25	1	207	0.00%	8.7/sec	19.0
/images/link_broken.png	3000	59	43	125	8	468	0.00%	8.7/sec	2833.2
/images/logo.jpg	3000	163	128	328	25	1110	0.00%	8.7/sec	9794.3
/security/login.do	3000	98	61	211	6	1993	0.00%	8.7/sec	54.2
/j_spring_security_check	3000	222	159	466	20	3069	0.00%	8.7/sec	102.0
/groupon/user/list.do	6000	122	86	263	12	3061	0.00%	17.2/sec	195.8
/groupon/user/edit.do	6000	119	79	253	9	4183	0.00%	17.4/sec	211.5
/j_spring_security_logout	3000	153	108	302	8	2580	0.00%	8.7/sec	94.0
TOTAL	63000	84	55	221	1	4183	0.00%	177.6/sec	15753.0



En esta tabla, el rendimiento es de 10654 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU oscila entre el 80% y el 100%, llegando a este último en pocas ocasiones.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9900	236	169	503	12	3024	0.00%	25.4/sec	277.6
/statics/circle.css	3300	12	8	26	1	857	0.00%	8.7/sec	4.1
/statics/query.min.js	3300	24	17	48	4	1256	0.00%	8.7/sec	825.0
/statics/style.css	3300	11	7	23	1	660	0.00%	8.7/sec	13.6
/statics/cookieconsent.min.js	3300	11	7	23	1	553	0.00%	8.7/sec	35.9
/statics/bootstrap.min.css	3300	26	19	56	3	368	0.00%	8.7/sec	1029.4
/statics/fontawesome.min.js	3300	15	11	31	2	291	0.00%	8.7/sec	36.5
/statics/cookies.js	3300	11	7	22	1	430	0.00%	8.7/sec	65.3
/statics/cookieconsent.js	3300	15	10	30	2	288	0.00%	8.7/sec	237.8
/statics/helpers.js	3300	10	7	22	1	440	0.00%	8.7/sec	19.8
/images/link_broken.png	3300	54	39	111	7	678	0.00%	8.7/sec	2829.1
/images/logo.png	3300	154	120	301	24	866	0.00%	8.7/sec	9783.7
/security/login.do	3300	184	106	423	7	3251	0.00%	8.7/sec	54.1
/_spring_security_check	3300	432	318	937	1	3058	0.00%	8.7/sec	10.9
/grouporuserlist.do	6600	215	141	466	12	294	0.00%	17.4/sec	195.0
/grouporuserredit.do	6600	169	133	440	10	2341	0.00%	17.4/sec	211.2
/_spring_security_logout	3300	237	170	493	14	2428	0.00%	8.7/sec	94.6
TOTAL	69300	130	44	347	1	3251	0.00%	178.0/sec	15787.4

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1400 ms que supera por bastante los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 10677 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente y por lo que el número de usuarios concurrentes vienen limitados por la CPU.



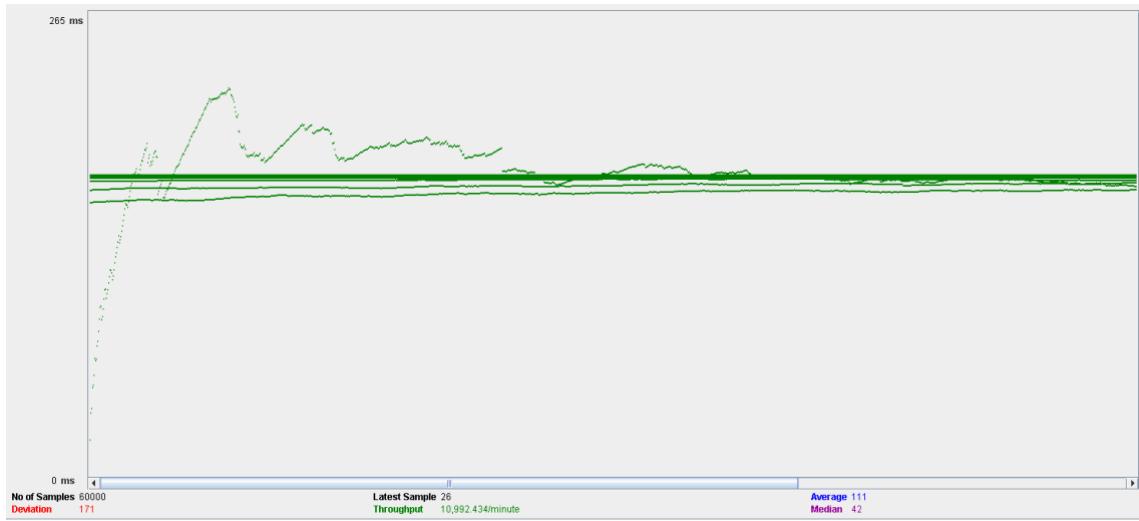
Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 080: Borrar una conjunta como moderador

En este caso de uso, accedemos a la página, nos autenticamos como moderador, y accedemos a la lista de conjuntas para poder borrar una y la borramos.

Comenzamos probando con 100 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8000	199	149	415	11	2368	0.00%	27.5/sec	206.8
/styles/bootstrap.min.css	3000	31	21	63	4	665	0.00%	9.4/sec	1115.8
/styles/style.css	3000	13	8	28	1	293	0.00%	9.4/sec	14.7
/scripts/jquery.min.js	3000	26	17	56	4	223	0.00%	9.4/sec	895.2
/styles/cookieconsent.mi...	3000	13	8	28	1	765	0.00%	9.4/sec	38.9
/styles/circle.css	3000	12	8	27	1	194	0.00%	9.4/sec	4.5
/scripts/bootstrap.min.js	3000	18	12	39	2	308	0.00%	9.4/sec	343.2
/scripts/scripts.js	3000	13	8	26	1	279	0.00%	9.4/sec	70.8
/scripts/counteragent.js	3000	18	11	38	2	322	0.00%	9.4/sec	33.0
/scripts/heinejs.js	3000	13	8	26	1	359	0.00%	9.4/sec	21.5
/images/logo.png	3000	187	127	346	23	1380	0.00%	9.4/sec	10609.2
/images/link_broken.png	3000	66	48	139	8	423	0.00%	9.4/sec	3067.4
/security/login.do	3000	163	102	381	6	2293	0.00%	9.4/sec	58.7
/_spring_security_check	3000	334	258	724	17	2151	0.00%	9.4/sec	109.2
/groupon/moderatorlist.do	3000	187	134	409	12	3073	0.00%	9.4/sec	104.3
/groupon/moderatorDelete...	3000	175	122	380	9	2092	0.00%	9.4/sec	126.0
/groupon/list.do	3000	180	127	388	11	2046	0.00%	9.4/sec	99.1
/_spring_security_logout	3000	204	157	427	15	2084	0.00%	9.4/sec	102.6
TOTAL	60000	111	42	287	1	3073	0.00%	183.2/sec	1652.4



En esta tabla, el rendimiento es de 10992 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

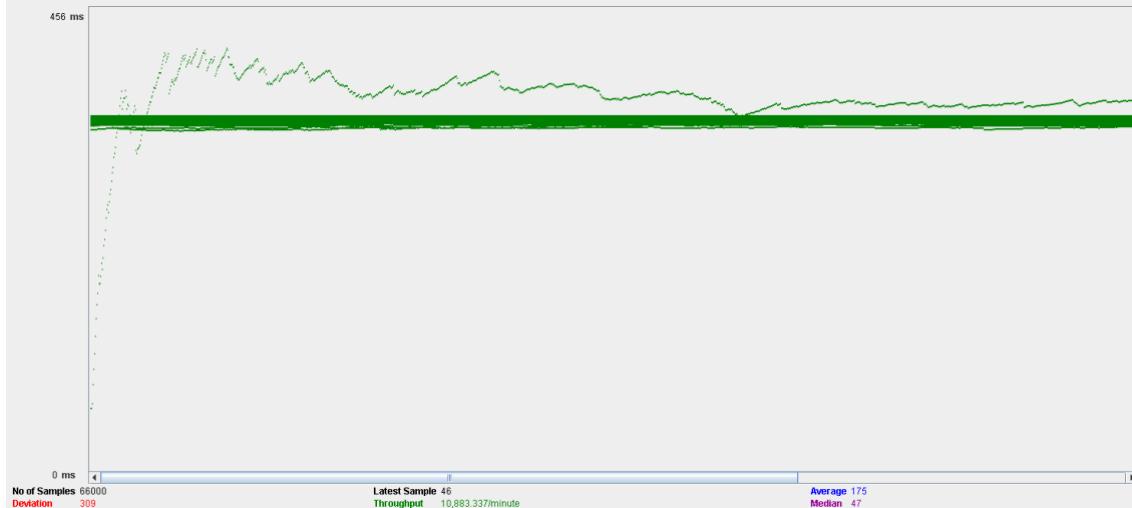
Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU es cercano al 100% aun dando tiempos admisibles.



La configuración en esta ocasión es de 110 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9900	327	214	732	12	4893	0.00%	27.2/sec	205.9
/static/bootstrap.min.css	3300	29	21	67	4	1319	0.00%	9.3/sec	1098.8
/static/style.css	3300	12	8	26	1	908	0.00%	9.3/sec	14.6
/scripts/jquery.min.js	3300	24	18	48	3	593	0.00%	9.3/sec	801.7
/scripts/cookieconsent.min.js	3300	12	8	26	1	273	0.00%	9.3/sec	38.3
/static/circle.css	3300	11	8	24	1	1138	0.00%	9.3/sec	4.4
/scripts/sticky-map.min.js	3300	17	12	34	2	571	0.00%	9.3/sec	337.9
/scripts/cookies.js	3300	11	8	23	1	236	0.00%	9.3/sec	60.7
/scripts/cookiesconsent.js	3300	15	11	31	2	313	0.00%	9.3/sec	318.0
/scripts/helpers.js	3300	12	8	25	1	1149	0.00%	9.3/sec	21.1
/images/logo.jpg	3300	154	127	305	24	667	0.00%	9.3/sec	10444.0
/images/link_broken.png	3300	63	51	130	8	453	0.00%	9.3/sec	3020.1
/security/login.do	3300	276	157	672	6	4334	0.00%	9.3/sec	57.8
/j_spring_security_check	3300	631	480	1329	18	4149	0.00%	9.3/sec	107.4
/groupAdministratorList.do	3300	330	298	764	12	2789	0.00%	9.3/sec	100.0
/groupAdminModeratorDelete..	3300	200	165	674	9	2779	0.00%	9.3/sec	124.4
/groupAdminList.do	3300	366	185	712	13	3045	0.00%	9.3/sec	97.0
/j_spring_security_logout	3300	336	219	744	14	3579	0.00%	9.3/sec	101.2
TOTAL	66000	175	47	493	1	4893	0.00%	181.4/sec	16784.1

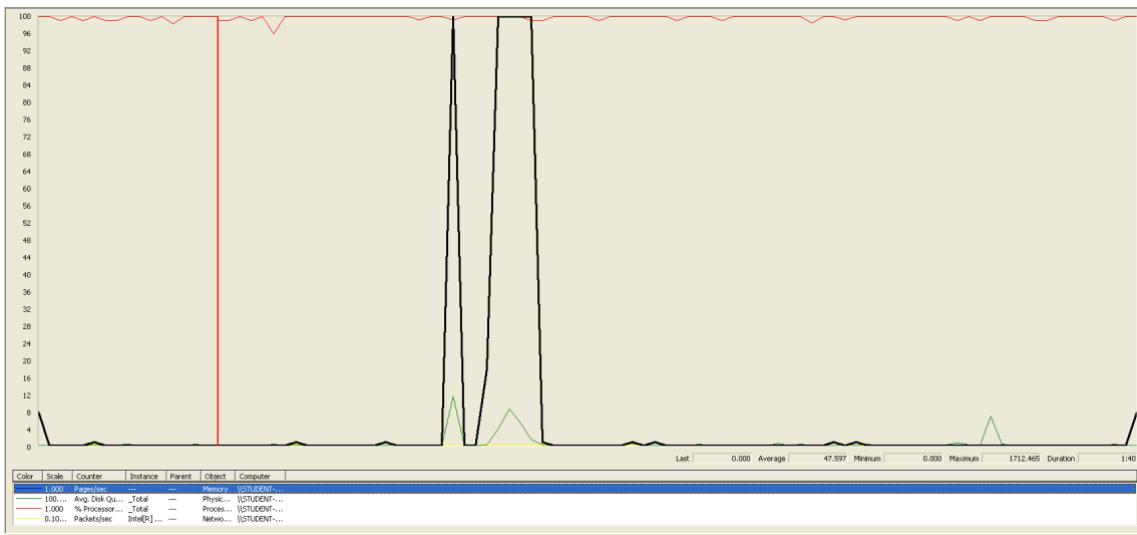
Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 2050 ms que supera por bastante los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 10833 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% casi constantemente, y la memoria llegando al 100% de forma puntual, por lo que el número de usuarios concurrentes vienen limitados por la CPU y en parte por la memoria.



Por lo que, en este caso de uso, los usuarios permitidos serán entre 100 y 110.

Caso de uso 081: Desplegar el tablón de mandos

En este caso de uso, accedemos a la página, nos autenticamos como administrador, y accedemos al dashboard y vamos entrando y saliendo de las distintas listas que este tiene

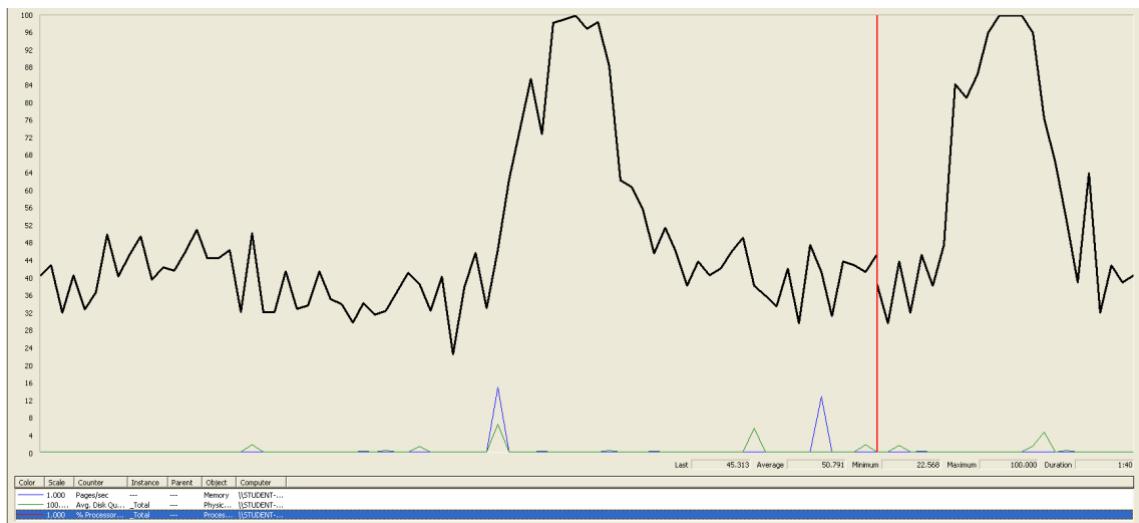
Comenzamos probando con 80 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni tiempos que superen por mucho a un segundo de duración, por lo que esta configuración de usuarios concurrentes es aceptable:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/styles/style.css	4000	56	32	106	12	1561	0.00%	4.2/sec	45.1
/styles/bootstrap.min.css	1360	9	5	20	1	114	0.00%	1.4/sec	2.2
/styles/bootstrap.min.js	1360	18	11	35	3	285	0.00%	1.4/sec	135.3
/styles/cookieconsent.min.js	1360	8	5	18	1	167	0.00%	1.4/sec	5.9
/styles/bootstrap.min.css	1360	19	12	37	4	146	0.00%	1.4/sec	168.7
/styles/circle.css	1360	9	5	20	1	181	0.00%	1.4/sec	7
/scripts/bootstrap.min.js	1360	13	8	27	2	407	0.00%	1.4/sec	51.9
/scripts/polyfills.js	1360	9	5	20	1	155	0.00%	1.4/sec	10.7
/scripts/helpers.js	1360	9	5	22	1	212	0.00%	1.4/sec	3.2
/scripts/cookieconsent.js	1360	12	7	24	2	382	0.00%	1.4/sec	48.8
/images/logo.jpg	1360	117	72	241	26	1344	0.00%	1.4/sec	1600.8
/security/login	1360	29	18	58	6	429	0.00%	1.4/sec	8.8
/_spring_security_check	1360	65	41	120	5	825	0.00%	1.4/sec	16.2
/category/administrator	1360	30	21	55	3	457	0.00%	1.4/sec	40.6
/dashboard/administrat...	2355.2	31	25	44	12	1402	0.00%	24.3/sec	316.3
/bargain/administrator	1360	23	19	36	9	399	0.00%	1.4/sec	10.2
/bargain/administrator...	1360	22	19	32	10	282	0.00%	1.4/sec	11.7
/bargain/administrator...	1360	20	19	30	10	134	0.00%	1.4/sec	12.7
/survey/administrator/	1359	19	18	29	10	157	0.00%	1.4/sec	9.9
/company/administrator/	1340	19	18	29	11	148	0.00%	1.4/sec	11.1
/company/administrator/	1324	20	18	30	9	106	0.00%	1.4/sec	11.0
/company/administrator/	1310	20	19	31	9	86	0.00%	1.4/sec	10.9
/user/administrator/	1292	24	22	37	12	80	0.00%	1.4/sec	7.1
/user/administrator/	1283	22	20	34	10	65	0.00%	1.4/sec	11.3
/category/administrator	1280	16	17	26	9	72	0.00%	1.4/sec	10.0
/company/administrator/	1280	21	19	31	10	114	0.00%	1.4/sec	13.8
/user/administrator/	1280	18	17	26	9	95	0.00%	1.4/sec	10.5
/groupon/administrator/	1280	21	19	32	11	99	0.00%	1.4/sec	13.8
/user/administrator/	1280	19	17	29	9	102	0.00%	1.4/sec	10.5
/user/administrator/	1280	20	17	32	9	188	0.00%	1.4/sec	10.5
/user/administrator/	1280	29	21	47	9	379	0.00%	1.4/sec	11.1
/user/administrator/purc...	1280	47	26	85	10	1290	0.00%	1.4/sec	11.2
/_spring_security_logout	1280	69	39	135	12	1828	0.00%	1.4/sec	15.4
TOTAL	68740	29	21	47	1	1928	0.00%	70.6/sec	3034.0



En esta tabla, el rendimiento es de 4236 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché, finalmente desciende hasta estabilizarse.

Ahora veremos a continuación una captura sacada de perfom.exe en el cual se ve que el uso de la CPU oscila entre el 40% y el 100% llegando a este último en muy pocas ocasiones.



La configuración en esta ocasión es de 90 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, y unos tiempos inadmisibles, los que podemos ver en el Aggregate Report y en el Graph Results de a continuación:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1800	184	60	497	12	3996	0.00%	4.8/sec	51.7
/styles/style.css	630	24	11	47	2	716	0.00%	1.7/sec	2.6
/scripts/jquery.min.js	630	40	22	92	4	836	0.00%	1.7/sec	160.7
/styles/cookieconsent.m.	630	21	10	45	1	495	0.00%	1.7/sec	7.0
/styles/bootstrap.min.css	630	48	23	109	4	991	0.00%	1.7/sec	200.4
/styles/circle.css	630	21	11	45	1	567	0.00%	1.7/sec	.8
/scripts/bootstrap.min.js	630	31	17	72	3	766	0.00%	1.7/sec	61.6
/scripts/polifills.js	630	19	12	47	2	229	0.00%	1.7/sec	12.7
/scripts/helpers.js	630	21	10	47	1	703	0.00%	1.7/sec	7.9
/scripts/cookieconsent.js	630	28	13	66	2	751	0.00%	1.7/sec	59.0
/images/logo.jpg	630	22	125	600	27	1157	0.00%	1.7/sec	1903.9
/seunthlogin.do	630	137	25	337	7	3790	0.00%	1.7/sec	10.6
/_spring_security_check	630	371	84	1007	15	5566	0.00%	1.7/sec	19.4
/images/link_broken.png	630	53	26	146	9	539	0.00%	1.7/sec	554.9
/dashboard/administrat.	10271	49	27	68	14	1764	0.00%	26.9/sec	350.5
/bargain/administratorl.	630	77	23	147	11	1300	0.00%	1.7/sec	12.4
/bargain/administratort.	630	30	21	47	11	341	0.00%	1.7/sec	14.2
/bargain/administratori.	630	26	21	40	11	198	0.00%	1.7/sec	15.5
/survey/administratorm.	628	25	20	37	11	324	0.00%	1.7/sec	12.1
/category/administrat.	606	22	19	34	11	190	0.00%	1.7/sec	13.3
/company/administratof.	598	23	19	33	12	97	0.00%	1.6/sec	13.0
/company/administratof.	585	21	19	32	11	99	0.00%	1.6/sec	12.7
/user/administratophP.	555	27	24	41	15	173	0.00%	1.6/sec	14.2
/user/administratorm.	542	25	20	39	12	380	0.00%	1.6/sec	13.1
/category/administrat.	540	21	18	31	9	207	0.00%	1.6/sec	13.9
/company/administratof.	540	23	20	35	11	227	0.00%	1.6/sec	17.8
/user/administratorm.	540	22	18	34	9	202	0.00%	1.6/sec	13.5
/groupadmin/administrat.	540	23	20	36	11	128	0.00%	1.6/sec	17.8
/user/administratorm.	540	21	18	34	11	121	0.00%	1.6/sec	13.5
/user/administratophP.	540	25	20	40	9	200	0.00%	1.6/sec	13.5
/user/administratophP.	540	29	24	71	10	362	0.00%	1.6/sec	14.3
/user/administratophP.	540	101	39	249	13	1234	0.00%	1.6/sec	14.3
/_spring_security_logout	540	143	71	354	17	1467	0.00%	1.6/sec	19.6
TOTAL	30493	62	25	100	1	5566	0.00%	77.4/sec	3438.7

Como podemos observar fijándonos en 90% Line el tiempo más alto es el que se produce al autenticarnos y volver a la pantalla de inicio que son aproximadamente 1500 ms que supera por bastante los 1000 ms que se consideran aceptables.



En esta tabla, el rendimiento es de aproximadamente 5153 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando, y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos perform.exe y vemos que la CPU tienen porcentajes muy altos, el porcentaje oscila entre el 40 y el 100%. Por lo tanto, el número de usuarios concurrentes en este caso de uso viene limitado por la CPU.



Por lo que, en este caso de uso, los usuarios permitidos serán entre 80 y 90.

Añadir categoría a chollo

En este caso de uso, una compañía puede añadir categorías a su chollo. Para ello, sigue la siguiente secuencia de pasos:

Mis chollos > (Escoger uno) > Añadir categoría

El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec	
/	4500	98	58	210	15	2237	0.00%	10.5/sec	114.3	
/styles/style.css	1500	10	6	21	1	313	0.00%	3.6/sec	5.6	
/styles/cookieconsent.min.css	1500	10	6	21	1	244	0.00%	3.6/sec	14.9	
/scripts/jquery.min.js	1500	22	15	45	4	597	0.00%	3.6/sec	343.5	
/styles/bootstrap.min.css	1500	26	18	52	5	708	0.00%	3.6/sec	428.2	
/styles/circle.css	1500	9	6	20	1	77	0.00%	3.6/sec	1.7	
/scripts/bootstrap.min.js	1500	14	10	28	3	389	0.00%	3.6/sec	131.7	
/scripts/polyfills.js	1500	10	6	21	1	320	0.00%	3.6/sec	27.2	
/scripts/helpers.js	1500	10	6	20	1	169	0.00%	3.6/sec	8.2	
/scripts/cookieconsent.js	1500	15	10	30	3	807	0.00%	3.6/sec	123.9	
/images/logo.jpg	1500	153	114	307	34	1089	0.00%	3.6/sec	4070.1	
/images/link_broken.png	4500	53	38	110	10	430	0.00%	10.5/sec	3430.0	
/favicon.ico	1500	11	7	25	1	112	0.00%	3.6/sec	20.2	
/security/login.do	1500	48	26	106	8	542	0.00%	3.6/sec	22.5	
/j_spring_security_check	1500	105	66	220	16	874	0.00%	3.6/sec	41.3	
/bargain/company/list.do	1500	63	41	130	14	1658	0.00%	3.6/sec	36.2	
/bargain/display.do	3000	824	787	1035	587	2230	0.00%	7.1/sec	85.0	
/images/yes.png	1500	10	6	23	1	114	0.00%	3.6/sec	5.2	
/category/company/addCatego...	1500	79	45	159	14	1687	0.00%	3.6/sec	32.9	
/bargain/company/addCategor...	1500	73	42	157	13	1664	0.00%	3.6/sec	47.8	
/j_spring_security_logout	1500	120	72	251	20	1638	0.00%	3.6/sec	39.4	
TOTAL	39000	111	30	268	1	2237	0.00%	91.1/sec	8863.5	

Como podemos observar, a la hora de desplegar el chollo, tarda aproximadamente 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 5463 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

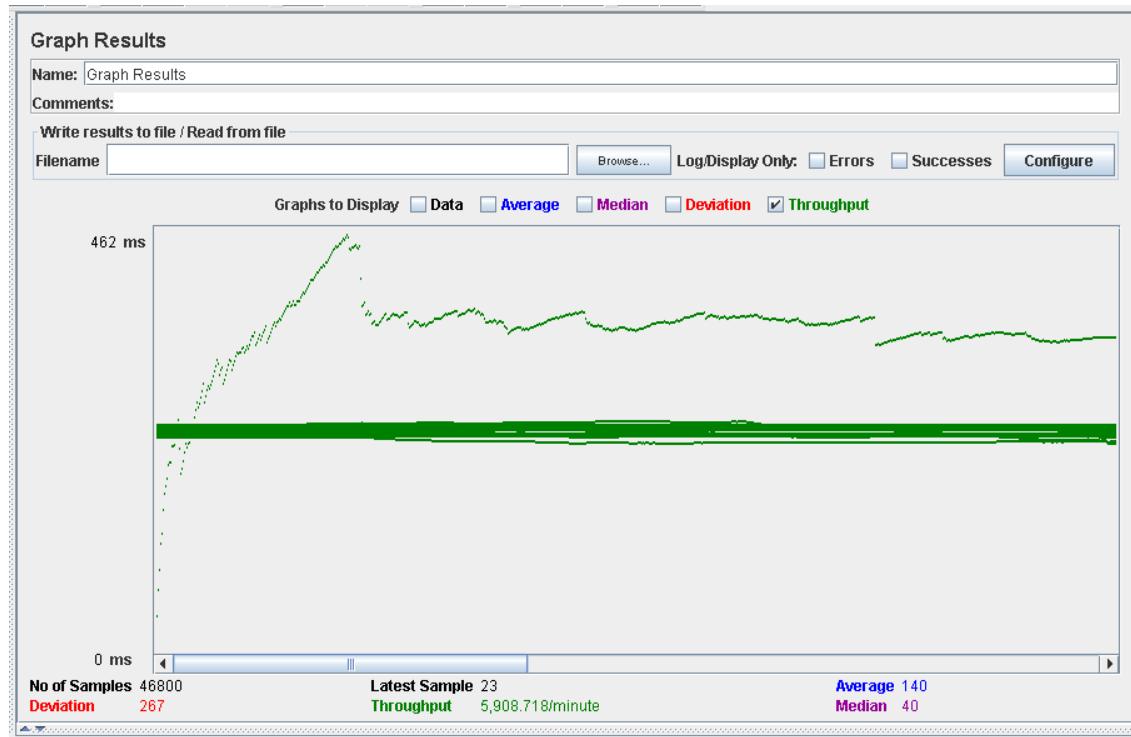
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
/	5400	155	81	341	13	4869	0.00%	11.4/sec	123.6
/styles/style.css	1800	15	10	33	1	782	0.00%	3.9/sec	6.1
/styles/cookieconsent.min.css	1800	15	10	34	2	339	0.00%	3.9/sec	16.1
/scripts/jquery.min.js	1800	31	22	61	5	560	0.00%	3.9/sec	370.5
/styles/bootstrap.min.css	1800	37	25	78	5	632	0.00%	3.9/sec	461.8
/styles/circle.css	1800	14	9	33	1	511	0.00%	3.9/sec	1.9
/scripts/bootstrap.min.js	1800	22	14	47	2	558	0.00%	3.9/sec	142.0
/scripts/polyfills.js	1800	15	9	33	1	281	0.00%	3.9/sec	29.3
/scripts/helpers.js	1800	15	10	34	1	706	0.00%	3.9/sec	8.9
/scripts/cookieconsent.js	1800	19	13	41	3	220	0.00%	3.9/sec	133.7
/images/logo.jpg	1800	185	128	382	35	1018	0.00%	3.9/sec	4391.0
/images/link_broken.png	5400	65	45	136	11	795	0.00%	11.4/sec	3709.1
/favicon.ico	1800	15	9	31	1	420	0.00%	3.9/sec	21.8
/security/login.do	1800	84	41	184	8	2717	0.00%	3.9/sec	24.3
/j_spring_security_check	1800	172	100	358	6	5088	0.00%	3.9/sec	44.4
/bargain/company/list.do	1800	97	55	204	15	2904	0.00%	3.9/sec	38.9
/bargain/display.do	3600	905	829	1200	574	3780	0.00%	7.7/sec	91.6
/images/yes.png	1800	13	8	30	1	475	0.00%	3.9/sec	5.6
/category/company/addCatego...	1800	123	67	265	14	2923	0.00%	3.9/sec	35.4
/bargain/company/addCategor...	1800	120	63	253	13	2815	0.00%	3.9/sec	51.4
/j_spring_security_logout	1800	177	106	382	10	3021	0.00%	3.9/sec	42.5
TOTAL	46800	140	40	421	1	5088	0.00%	98.5/sec	9585.2

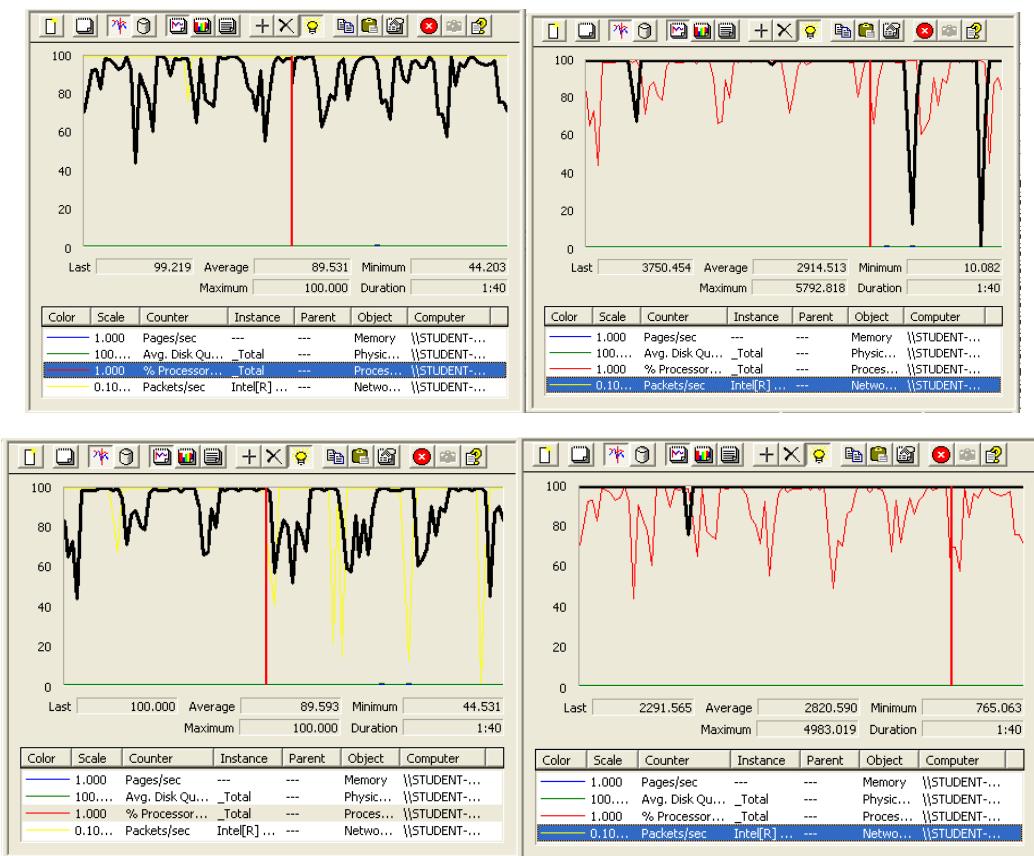
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de desplegar el chollo, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 5908.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y de la tarjeta de red. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



Editar y desplegar configuración

En este caso de uso, un administrador puede cambiar y ver la configuración del sistema. Para ello, sigue la siguiente secuencia de pasos:

Configuración

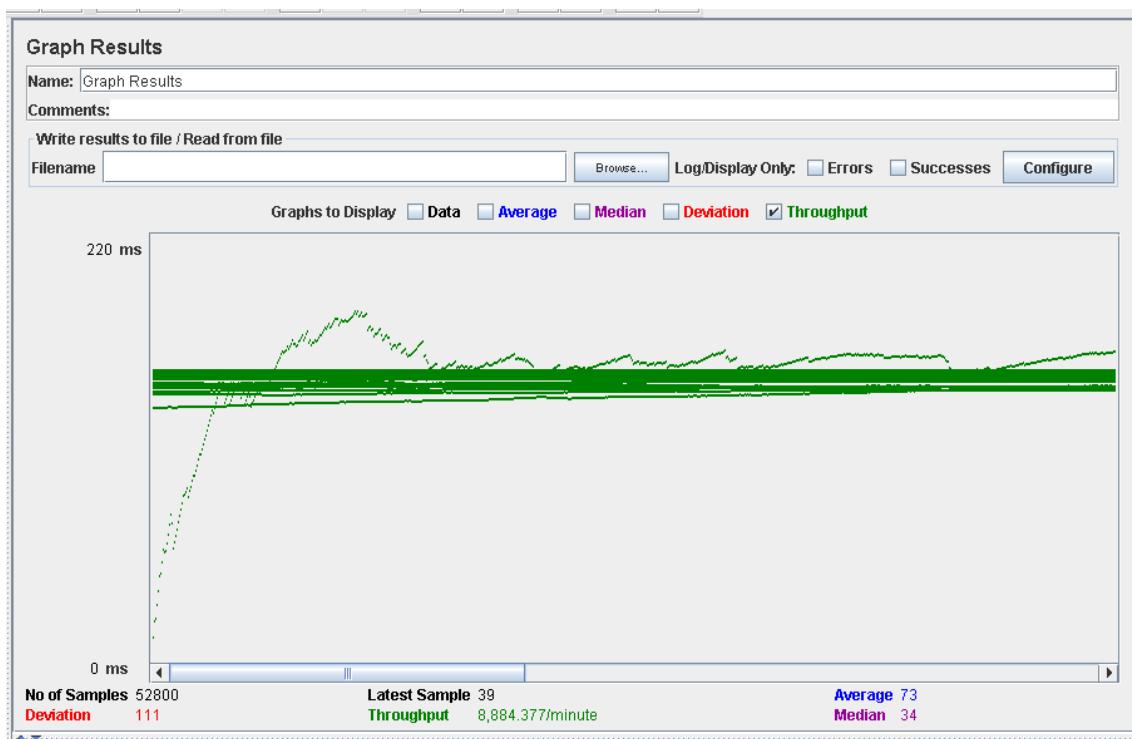
El límite de este caso de uso se sitúa en 80 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Filename <input type="text"/>					Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	122	73	269	1	2763	0.00%	20.2/sec	219.7
/styles/style.css	2400	12	9	28	0	931	0.00%	6.9/sec	10.8
/scripts/bootstrap...	2400	19	14	39	0	1125	0.00%	6.9/sec	252.4
/styles/circle.c...	2400	12	9	28	0	336	0.00%	6.9/sec	3.3
/styles/bootstrap...	2400	31	24	65	0	404	0.00%	6.9/sec	821.2
/scripts/jquery...	2400	27	21	58	0	412	0.00%	6.9/sec	658.9
/styles/cookie...	2400	13	9	29	0	400	0.00%	6.9/sec	28.6
/scripts/polyfill...	2400	13	9	29	0	454	0.00%	6.9/sec	52.1
/scripts/help...	2400	12	9	28	0	127	0.00%	6.9/sec	15.8
/images/logo....	2400	176	124	358	22	1168	0.00%	6.9/sec	7809.7
/scripts/cookie...	2400	17	13	36	0	917	0.00%	6.9/sec	237.8
/favicon.ico	2400	14	10	31	0	147	0.00%	6.9/sec	38.7
/security/login...	2400	70	40	154	0	1488	0.00%	6.9/sec	43.2
/j_spring_sec...	2400	156	101	327	8	1202	0.00%	6.9/sec	79.4
/images/link....	2400	60	43	129	0	511	0.00%	6.9/sec	2261.7
/configuration/...	4800	93	56	205	0	1026	0.00%	13.7/sec	103.5
/configuration/...	4800	141	88	322	0	3028	0.00%	13.9/sec	116.6
/j_spring_sec...	2400	131	82	271	3	1891	0.00%	6.9/sec	75.6
TOTAL	52800	73	34	188	0	3028	0.00%	148.1/sec	12471.3

Como podemos observar, a la hora de solicitar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8884 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 90 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

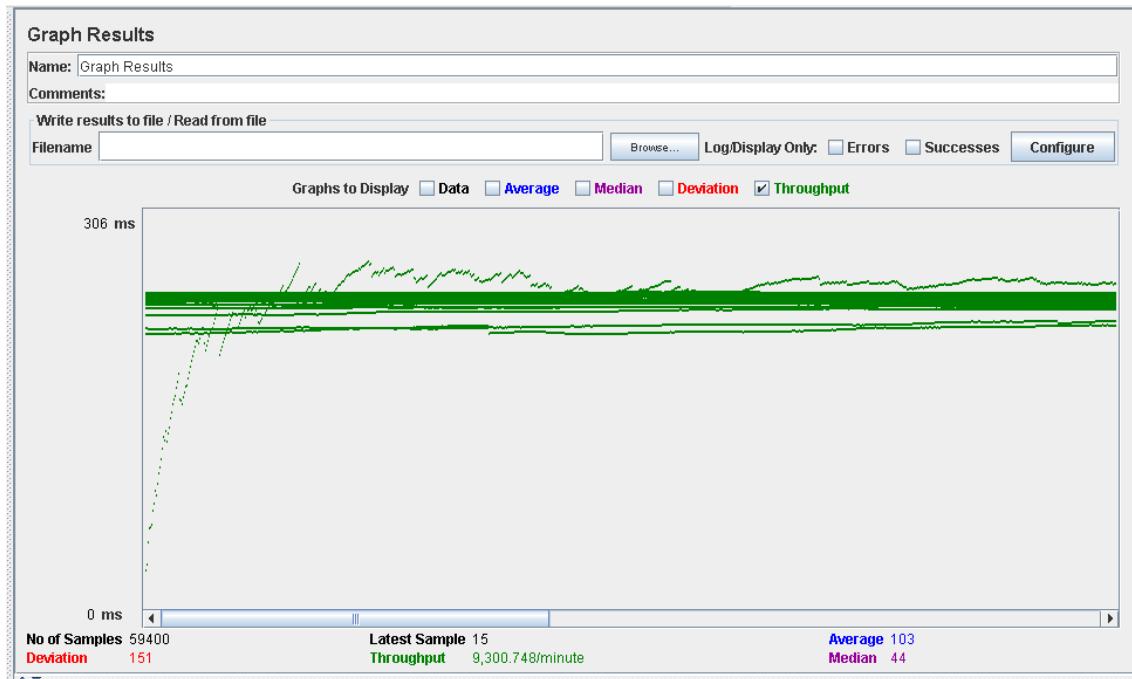
Write results to file / Read from file
Filename Browse... Log/Display Only Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	172	128	366	0	2305	0.00%	21.1/sec	230.0
/styles/style.css	2700	17	12	36	0	487	0.00%	7.2/sec	11.3
/scripts/bootstrap.min.js	2700	24	18	50	0	381	0.00%	7.2/sec	263.6
/styles/circle.css	2700	16	12	34	0	303	0.00%	7.2/sec	3.5
/styles/bootstrap.min.css	2700	42	32	86	0	491	0.00%	7.2/sec	857.5
/scripts/jquery.min.js	2700	36	27	74	0	397	0.00%	7.2/sec	688.0
/styles/cookieconsent.min.css	2700	16	12	35	0	172	0.00%	7.2/sec	29.9
/scripts/polyfills.js	2700	16	12	36	0	273	0.00%	7.2/sec	54.4
/scripts/helpers.js	2700	16	11	35	0	314	0.00%	7.2/sec	16.5
/images/logo.jpg	2700	228	186	449	20	1147	0.00%	7.2/sec	8152.2
/scripts/cookieconsent.js	2700	22	16	45	0	390	0.00%	7.2/sec	248.3
/favicon.ico	2700	17	13	37	0	188	0.00%	7.2/sec	40.4
/security/login.do	2700	106	70	241	0	1157	0.00%	7.2/sec	45.2
/j_spring_security_check	2700	245	185	517	10	3046	0.00%	7.2/sec	82.9
/images/link_broken.png	2700	81	63	163	0	665	0.00%	7.2/sec	2361.9
/configuration/administrator/display.do	5400	137	97	302	0	3012	0.00%	14.4/sec	108.3
/configuration/administrator/edit.do	5400	212	146	465	0	4662	0.00%	14.5/sec	121.8
/j_spring_security_logout	2700	178	130	371	2	3225	0.00%	7.2/sec	78.9
TOTAL	59400	103	44	276	0	4662	0.00%	155.0/sec	130555.8

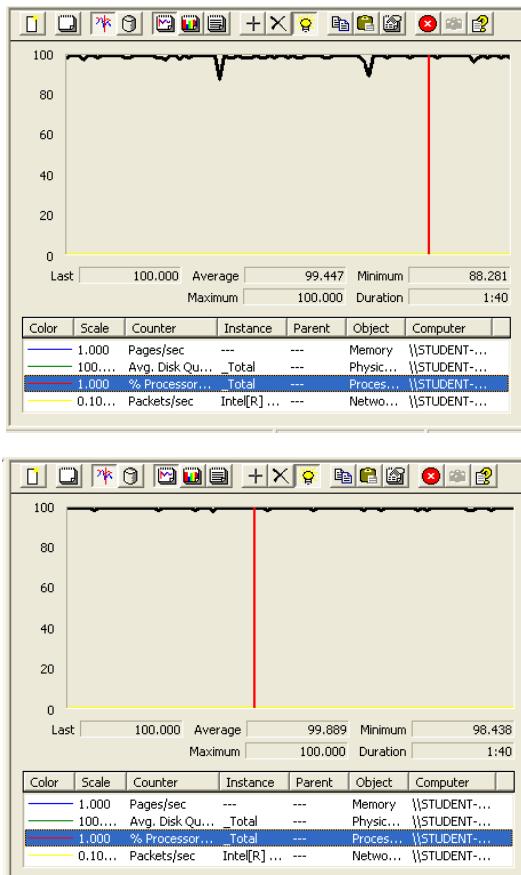
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 9300.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador.
Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Crear categoría

En este caso de uso, un moderador puede crear una categoría. Para ello, sigue la siguiente secuencia de pasos:

Categorías > Nueva categoría

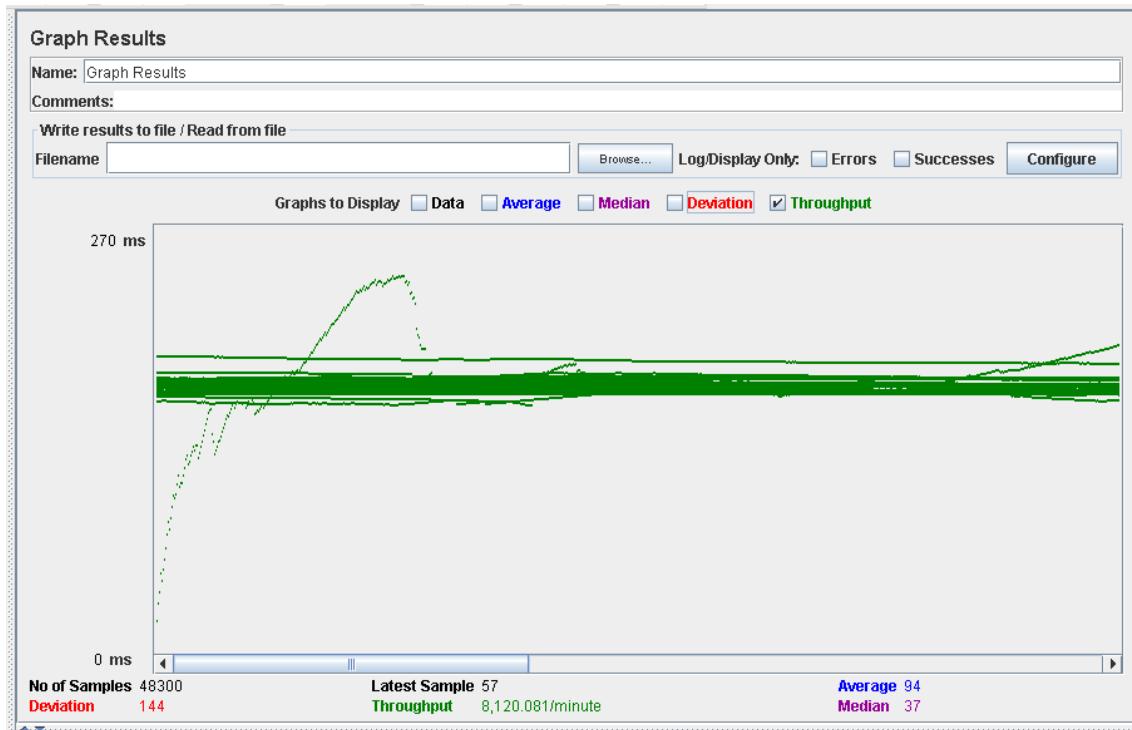
El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	6300	146	90	327	1	2238	0.00%	17.7/sec	193.2	
/styles/cookieconsent.min....	2100	13	10	28	0	892	0.00%	6.1/sec	25.1	
/styles/style.css	2100	13	9	28	0	427	0.00%	6.1/sec	9.5	
/scripts/jquery.min.js	2100	28	22	57	0	273	0.00%	6.1/sec	577.0	
/styles/bootstrap.min.css	2100	32	25	66	0	481	0.00%	6.1/sec	719.3	
/scripts/bootstrap.min.js	2100	19	14	39	0	283	0.00%	6.1/sec	221.2	
/styles/circle.css	2100	12	9	27	0	206	0.00%	6.1/sec	2.9	
/scripts/polyfills.js	2100	12	10	27	0	219	0.00%	6.1/sec	45.7	
/scripts/helpers.js	2100	12	9	26	0	112	0.00%	6.1/sec	13.8	
/scripts/cookieconsent.js	2100	18	13	37	0	1001	0.00%	6.1/sec	208.2	
/images/logo.jpg	2100	199	152	384	24	1811	0.00%	6.1/sec	6840.0	
/images/link_broken.png	4200	68	51	138	0	544	0.00%	12.0/sec	3903.5	
/favicon.ico	2100	15	11	31	0	325	0.00%	6.1/sec	34.0	
/security/login.do	2100	82	51	190	0	1322	0.00%	6.1/sec	37.9	
/j_spring_security_check	2100	184	124	394	10	2331	0.00%	6.1/sec	70.5	
/category/display.do	4200	298	375	574	0	1455	0.00%	12.0/sec	111.2	
/category/moderator/creat...	2100	93	59	202	0	2725	0.00%	6.1/sec	47.1	
/category/moderator/edit.do	2100	113	72	258	0	945	0.00%	6.1/sec	47.5	
/j_spring_security_logout	2100	150	99	339	2	1440	0.00%	6.1/sec	66.2	
TOTAL	48300	94	37	271	0	2725	0.00%	135.3/sec	12836.3	

Como podemos observar, a la hora cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8120 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file

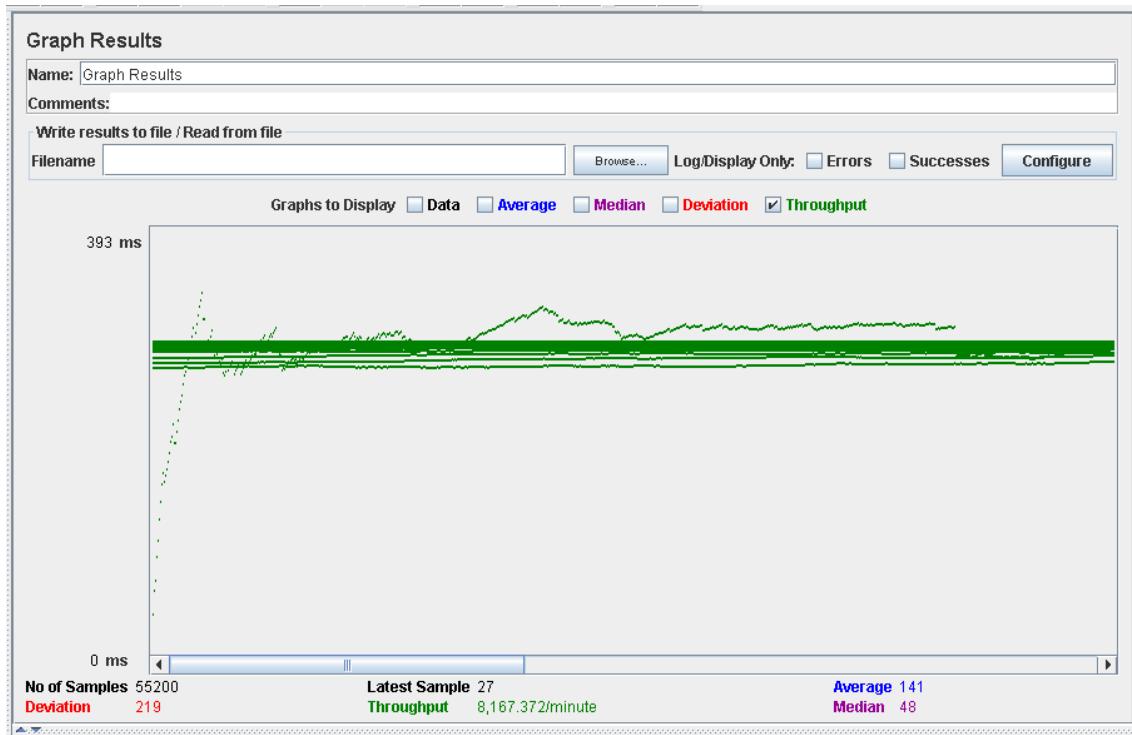
Filename Browse... Log/Display Only: Errors Successes Throughput

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	242	158	550	1	4112	0.00%	17.8/sec	194.3
/styles/cookieconsent.min....	2400	17	12	38	0	329	0.00%	6.1/sec	25.1
/styles/style.css	2400	17	12	37	0	350	0.00%	6.1/sec	9.5
/scripts/jquery.min.js	2400	38	27	78	0	1108	0.00%	6.1/sec	576.9
/scripts/bootstrap.min.css	2400	41	30	82	0	441	0.00%	6.1/sec	719.2
/scripts/bootstrap.min.js	2400	24	18	49	0	377	0.00%	6.1/sec	221.1
/styles/circle.css	2400	16	11	35	0	345	0.00%	6.1/sec	2.9
/scripts/polyfills.js	2400	17	11	36	0	352	0.00%	6.1/sec	45.6
/scripts/helpers.js	2400	16	11	33	0	373	0.00%	6.1/sec	13.8
/scripts/cookieconsent.js	2400	23	17	47	0	1250	0.00%	6.1/sec	208.1
/images/logo.jpg	2400	244	184	494	23	1638	0.00%	6.1/sec	6834.4
/images/link_broken.png	4800	86	62	183	0	1658	0.00%	12.0/sec	3909.4
/favicon.ico	2400	18	13	39	0	239	0.00%	6.1/sec	33.9
/security/login.do	2400	156	95	353	0	3103	0.00%	6.1/sec	37.8
/j_spring_security_check	2400	338	236	758	8	3623	0.00%	6.1/sec	70.5
/category/display.do	4800	380	401	741	0	2848	0.00%	12.1/sec	111.7
/category/moderator/creat...	2400	168	104	377	0	3118	0.00%	6.1/sec	47.2
/category/moderator/edit.do	2400	195	126	458	0	3098	0.00%	6.1/sec	47.6
/j_spring_security_logout	2400	249	168	560	3	2708	0.00%	6.1/sec	66.4
TOTAL	55200	141	48	417	0	4112	0.00%	136.1/sec	12911.0

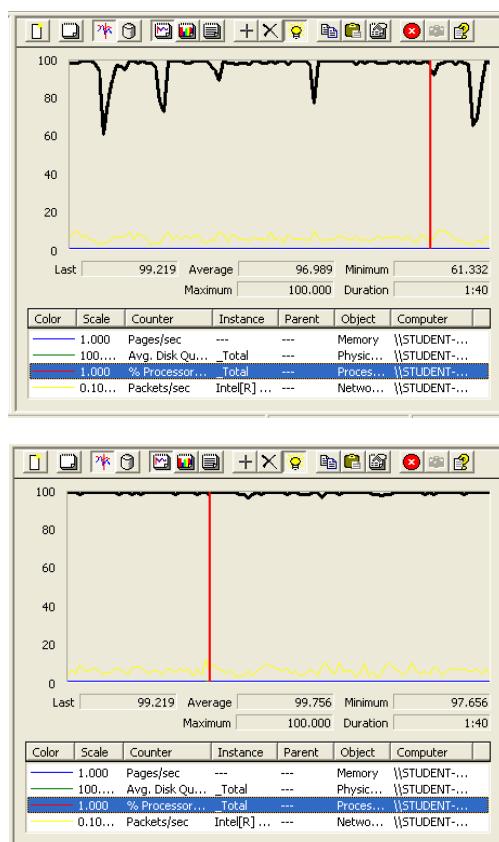
Include group name in label? Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 8167.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Crear patrocinio

En este caso de uso, un patrocinador puede crear un patrocinio para un chollo. Para ello, sigue la siguiente secuencia de pasos:

Nuevo patrocinio > (Escoger chollo)

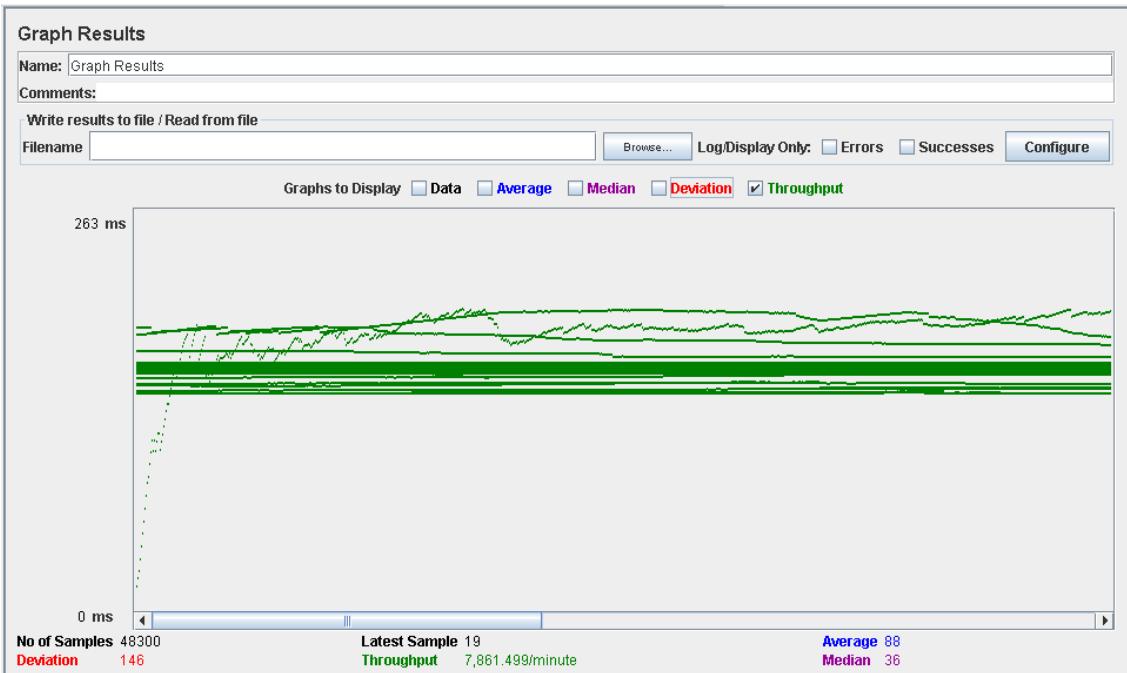
El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Filename <input type="text"/>					Browse...	Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="checkbox"/> Configure				
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	6300	173	108	410	0	2720	0.00%	17.1/sec	185.3	
/styles/style.css	2100	11	9	27	0	186	0.00%	5.9/sec	9.2	
/styles/cookieconsent.min.css	2100	12	8	27	0	266	0.00%	5.9/sec	24.4	
/styles/bootstrap.min.css	2100	29	22	59	0	363	0.00%	5.9/sec	700.9	
/styles/circle.css	2100	12	9	28	0	157	0.00%	5.9/sec	2.8	
/scripts/bootstrap.min.js	2100	16	12	34	0	211	0.00%	5.9/sec	215.5	
/scripts/jquery.min.js	2100	25	18	51	0	517	0.00%	5.9/sec	562.4	
/scripts/polyfills.js	2100	12	9	27	0	169	0.00%	5.9/sec	44.5	
/scripts/helpers.js	2100	11	9	26	0	261	0.00%	5.9/sec	13.5	
/scripts/cookieconsent.js	2100	16	11	33	0	380	0.00%	5.9/sec	202.9	
/images/logo.jpg	2100	162	123	321	17	885	0.00%	5.9/sec	6662.5	
/favicon.ico	2100	13	10	29	0	247	0.00%	5.9/sec	32.9	
/security/login.do	2100	102	57	243	0	1621	0.00%	5.9/sec	36.6	
/j_spring_security_check	2100	227	146	526	1	2592	0.00%	5.9/sec	66.5	
/images/link_broken.png	4200	54	41	114	0	479	0.00%	11.5/sec	3759.1	
/bargain/sponsor/list.do	2100	121	77	279	0	1903	0.00%	5.8/sec	48.3	
/sponsorship/sponsor/create.do	2100	135	85	317	0	2148	0.00%	5.8/sec	44.0	
/sponsorship/sponsor/edit.do	2100	157	101	358	0	2052	0.00%	5.8/sec	44.5	
/sponsorship/sponsor/list.do	2100	154	96	339	0	1970	0.00%	5.8/sec	58.0	
/j_spring_security_logout	2100	188	112	441	0	3728	0.00%	5.8/sec	63.4	
TOTAL	48300	88	36	239	0	3728	0.00%	131.0/sec	12420.4	

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 7851 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

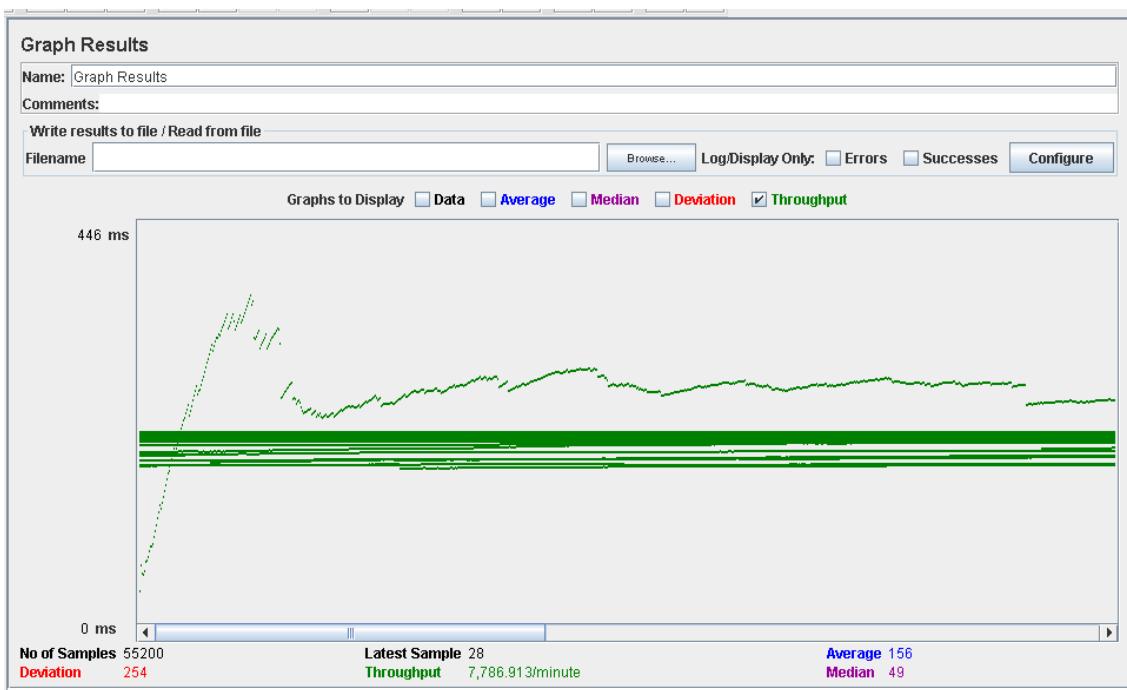
Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	309	200	736	0	3558	0.00%	16.9/sec	183.6
/styles/style.css	2400	15	11	34	0	165	0.00%	5.8/sec	9.0
/styles/cookieconsent.min.css	2400	15	11	34	0	546	0.00%	5.8/sec	23.9
/styles/bootstrap.min.css	2400	38	26	80	0	614	0.00%	5.8/sec	685.6
/styles/circle.css	2400	15	11	32	0	612	0.00%	5.8/sec	2.8
/scripts/bootstrap.min.js	2400	22	15	47	0	604	0.00%	5.8/sec	210.8
/scripts/jquery.min.js	2400	33	22	68	0	600	0.00%	5.8/sec	550.0
/scripts/polyfills.js	2400	16	11	34	0	401	0.00%	5.8/sec	43.5
/scripts/helpers.js	2400	15	11	33	0	547	0.00%	5.8/sec	13.2
/scripts/cookieconsent.js	2400	20	14	43	0	425	0.00%	5.8/sec	198.4
/images/logo.jpg	2400	206	147	423	21	1333	0.00%	5.8/sec	6515.6
/favicon.ico	2400	18	12	38	0	215	0.00%	5.8/sec	32.3
/security/login.do	2400	209	132	478	0	2159	0.00%	5.8/sec	36.0
/j_spring_security_check	2400	452	313	997	1	4974	0.00%	5.8/sec	65.6
/images/link_broken.png	4800	75	49	161	0	967	0.00%	11.4/sec	3730.0
/bargain/sponsor/list.do	2400	264	166	622	0	3141	0.00%	5.8/sec	47.8
/sponsorship/sponsor/create.do	2400	269	176	614	0	2028	0.00%	5.8/sec	43.7
/sponsorship/sponsor/edit.do	2400	289	188	674	0	4222	0.00%	5.8/sec	44.2
/sponsorship/sponsor/list.do	2400	291	197	668	0	4023	0.00%	5.8/sec	57.7
/j_spring_security_logout	2400	318	221	708	0	4052	0.00%	5.8/sec	63.1
TOTAL	55200	156	49	450	0	4974	0.00%	129.8/sec	12302.6

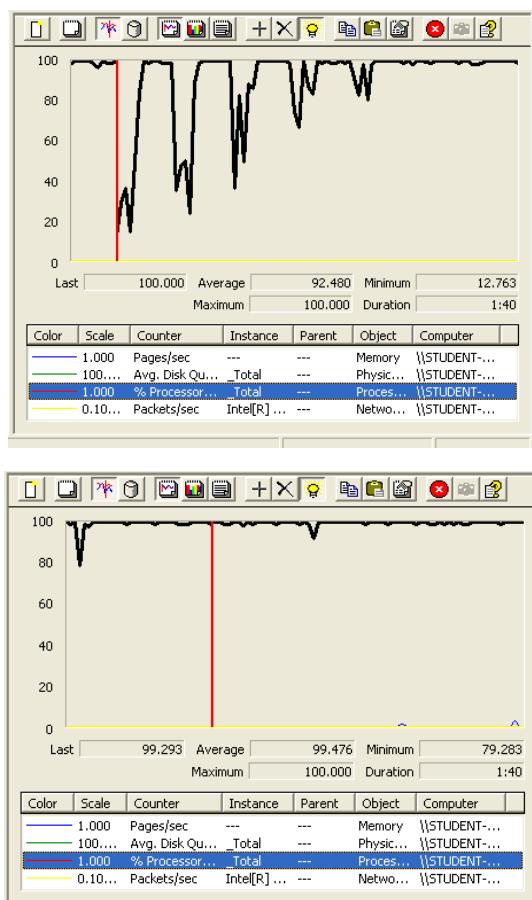
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de guardar el patrocinio, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 7785.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador.
Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Crear chollo

En este caso de uso, una compañía puede crear un chollo. Para ello, sigue la siguiente secuencia de pasos:

Mis chollos > Crear

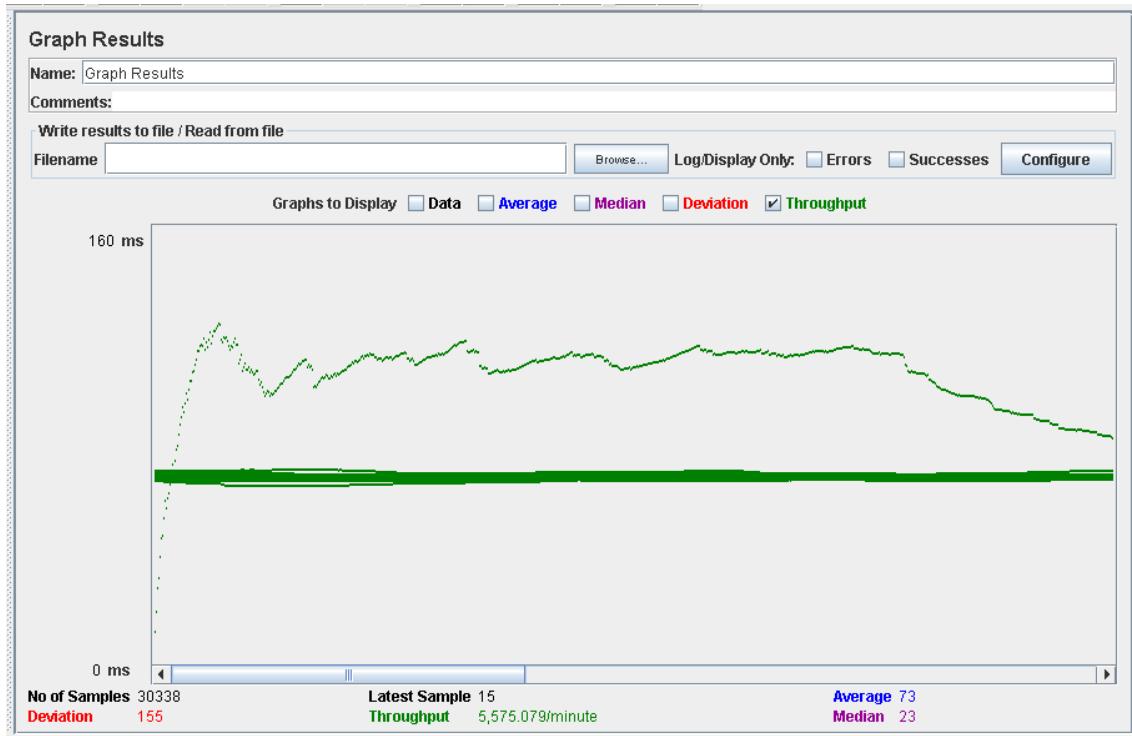
El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Filename <input type="text"/>				<input type="button" value="Browse..."/>		<input type="checkbox"/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes		<input type="button" value="Configure"/>		
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	3937	109	64	264	14	1039	0.00%	12.1/sec	131.1	
/styles/style.css	1333	8	5	16	1	140	0.00%	4.1/sec	6.4	
/styles/cookieconsent.min.c...	1332	8	5	17	1	276	0.00%	4.1/sec	16.9	
/styles/circle.css	1332	9	5	17	1	697	0.00%	4.1/sec	2.0	
/scripts/bootstrap.min.js	1332	12	8	23	2	757	0.00%	4.1/sec	149.3	
/scripts/jquery.min.js	1332	17	12	34	4	244	0.00%	4.1/sec	389.5	
/styles/bootstrap.min.css	1332	20	14	41	5	160	0.00%	4.1/sec	485.4	
/scripts/polyfills.js	1332	8	5	16	1	89	0.00%	4.1/sec	30.8	
/scripts/helpers.js	1332	7	5	16	1	92	0.00%	4.1/sec	9.3	
/scripts/cookieconsent.js	1332	11	8	22	2	118	0.00%	4.1/sec	140.4	
/images/logo.jpg	1332	111	84	200	29	823	0.00%	4.1/sec	4611.4	
/favicon.ico	1332	7	4	13	1	516	0.00%	4.1/sec	22.9	
/security/login.do	1324	59	28	149	8	796	0.00%	4.1/sec	25.5	
/j_spring_security_check	1321	125	74	286	22	977	0.00%	4.1/sec	46.7	
/images/link_broken.png	1320	31	23	55	10	691	0.00%	4.1/sec	1338.9	
/bargain/company/list.do	2801	85	45	197	14	1253	0.08%	8.1/sec	168.1	
/category/company/createBa...	1306	56	30	127	11	1149	0.00%	4.1/sec	36.2	
/bargain/company/create.do	1300	72	38	179	12	714	0.00%	4.1/sec	41.8	
/bargain/company/edit.do	1293	506	358	1127	26	2579	0.15%	4.1/sec	55.2	
/j_spring_security_logout	1283	122	77	285	17	1132	0.00%	4.1/sec	44.4	
TOTAL	30338	73	23	181	1	2579	0.01%	92.9/sec	7706.7	

Como podemos observar, a la hora de guardar el chollo, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 5575 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file
Filename: Browse... Log/Display Only: Errors Successes Configure

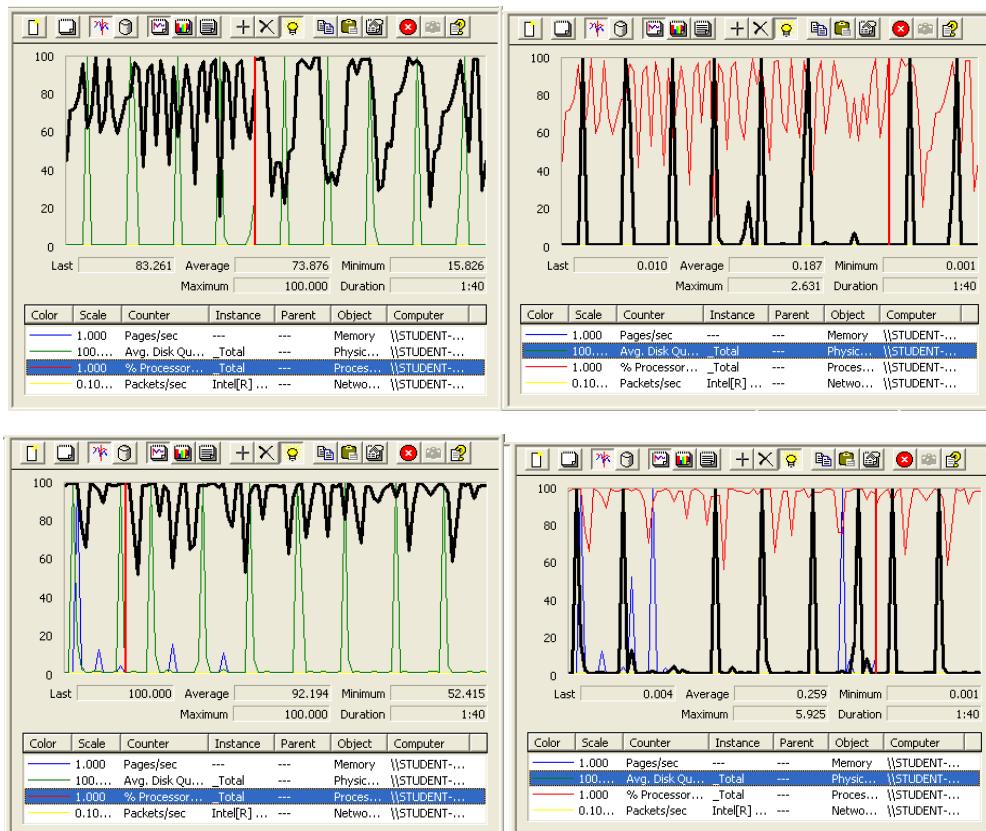
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5400	179	110	429	14	1808	0.00%	12.9/sec	140.6
/styles/style.css	1800	9	7	20	1	158	0.00%	4.4/sec	6.9
/styles/cookieconsent.min.c...	1800	9	7	19	1	87	0.00%	4.4/sec	18.3
/styles/circle.css	1800	9	6	20	1	127	0.00%	4.4/sec	2.1
/scripts/bootstrap.min.js	1800	13	9	27	3	292	0.00%	4.4/sec	161.8
/scripts/jquery.min.js	1800	20	15	38	4	228	0.00%	4.4/sec	422.3
/styles/bootstrap.min.css	1800	23	17	46	5	310	0.00%	4.4/sec	526.4
/scripts/polyfills.js	1800	10	7	20	2	220	0.00%	4.4/sec	33.4
/scripts/helpers.js	1800	9	6	19	1	283	0.00%	4.4/sec	10.1
/scripts/cookieconsent.js	1800	13	10	27	2	148	0.00%	4.4/sec	152.3
/images/logo.jpg	1800	132	100	251	28	775	0.00%	4.4/sec	5002.0
/favicon.ico	1800	8	5	17	1	219	0.00%	4.4/sec	24.8
/security/login.do	1800	102	50	243	7	1420	0.00%	4.4/sec	27.7
/j_spring_security_check	1800	225	125	562	16	2024	0.00%	4.4/sec	50.6
/images/link_broken.png	1800	39	27	76	10	465	0.00%	4.4/sec	1448.2
/bargain/companylist.do	3600	131	76	318	14	1951	0.00%	8.8/sec	186.4
/category/company/createBa...	1800	90	43	232	12	1254	0.00%	4.4/sec	39.2
/bargain/company/create.do	1800	133	76	325	12	1323	0.00%	4.4/sec	45.3
/bargain/company/edit.do	1800	747	596	1615	28	3844	0.00%	4.4/sec	57.4
/j_spring_security_logout	1800	186	127	430	16	1437	0.00%	4.4/sec	48.4
TOTAL	41400	112	28	302	1	3844	0.00%	99.2/sec	8177.4

En este caso vemos como a la hora de guardar el chollo y desplegarlo, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 5954.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y del disco duro. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



Borrar chollo

En este caso de uso, un usuario puede borrar un chollo. Para ello, sigue la siguiente secuencia de pasos:

Mis chollos > (Escoger uno)

El límite de este caso de uso se sitúa en 80 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Filename <input type="text"/>				<input type="button" value="Browse..."/>	<input type="checkbox"/> Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	202	121	472	13	2528	0.00%	18.8/sec	204.5
/scripts/jquery...	2400	44	28	95	4	831	0.00%	6.4/sec	610.5
/styles/bootstrap...	2400	46	29	100	5	528	0.00%	6.4/sec	781.1
/styles/style.css	2400	21	13	47	1	313	0.00%	6.4/sec	10.0
/styles/circle.c...	2400	22	14	49	2	495	0.00%	6.4/sec	3.1
/styles/cookie...	2400	21	13	48	1	365	0.00%	6.4/sec	26.5
/scripts/bootstrap...	2400	29	18	65	3	714	0.00%	6.4/sec	234.0
/scripts/polyfill...	2400	21	13	48	2	350	0.00%	6.4/sec	48.3
/scripts/help...	2400	22	13	49	1	465	0.00%	6.4/sec	14.6
/scripts/cookie...	2400	27	17	57	3	453	0.00%	6.4/sec	220.3
/images/logo....	2400	273	179	565	34	2656	0.00%	6.4/sec	7235.3
/favicon.ico	2400	24	16	53	2	457	0.00%	6.4/sec	35.9
/security/login...	2400	132	74	325	8	1791	0.00%	6.4/sec	40.1
/j_spring_sec...	2400	280	182	666	18	3082	0.00%	6.4/sec	73.5
/images/link....	2400	98	59	210	10	1121	0.00%	6.4/sec	2097.0
/bargain/com...	4800	189	102	400	13	1765	0.00%	12.8/sec	127.7
/bargain/com...	4800	164	96	395	11	2167	0.00%	12.9/sec	177.3
/j_spring_sec...	2400	228	146	517	18	1919	0.00%	6.4/sec	70.4
TOTAL	52800	116	47	311	1	3082	0.00%	137.7/sec	11693.5

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8260 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 90 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	364	278	765	14	3292	0.00%	19.0/sec	206.5
/scripts/jquery...	2700	32	22	64	4	509	0.00%	6.5/sec	614.5
/styles/bootstrap...	2700	35	24	75	4	402	0.00%	6.5/sec	766.0
/styles/style.css	2700	15	10	27	1	1871	0.00%	6.5/sec	10.1
/styles/circle.c...	2700	15	10	30	1	255	0.00%	6.5/sec	3.1
/styles/cookie...	2700	14	11	29	1	318	0.00%	6.5/sec	26.7
/scripts/bootstrap...	2700	21	15	42	2	386	0.00%	6.5/sec	235.6
/scripts/polyfill...	2700	14	11	30	1	186	0.00%	6.5/sec	48.6
/scripts/help...	2700	14	10	28	1	333	0.00%	6.5/sec	14.7
/scripts/cookie...	2700	20	14	40	2	550	0.00%	6.5/sec	221.7
/images/logo....	2700	206	142	432	34	1731	0.00%	6.5/sec	7279.7
/favicon.ico	2700	16	12	32	2	293	0.00%	6.5/sec	36.1
/security/login...	2700	182	597	7	3458	0.00%	6.5/sec	40.3	
/j_spring_sec...	2700	565	474	1148	22	3413	0.00%	6.5/sec	73.8
/images/link...	2700	77	52	162	10	867	0.00%	6.5/sec	2106.4
/bargain/com...	5400	310	236	651	14	3146	0.00%	12.8/sec	128.4
/bargain/com...	5400	303	223	658	11	3524	0.00%	12.9/sec	178.0
/j_spring_sec...	2700	393	305	802	16	3221	0.00%	6.5/sec	70.7
TOTAL	59400	183	52	529	1	3524	0.00%	139.0/sec	11808.8

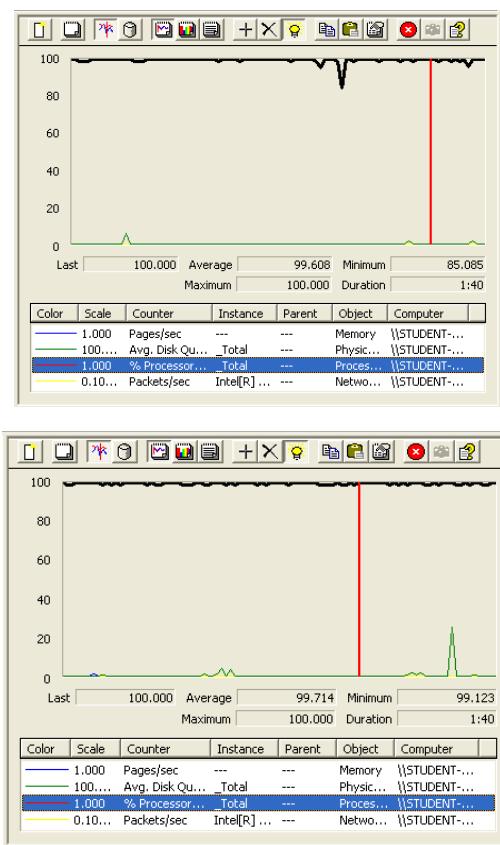
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de cargar la página principal o iniciar sesión, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 8341.



Nuestro sistema se comporta así en este caso debido a las limitaciones del. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Borrar chollo (moderador)

En este caso de uso, un usuario puede borrar un chollo. Para ello, sigue la siguiente secuencia de pasos:

Borrar Chollos

El límite de este caso de uso se sitúa en 80 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
<input type="text" value="Aggregate Report"/> Name: Aggregate Report <input type="text"/> Comments: <input type="checkbox"/> Write results to file / Read from file <input type="text" value="Filename"/> <input type="button" value="Browse..."/> <input type="checkbox"/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	149	92	330	13	2176	0.00%	22.2/sec	243.0
/styles/style.css	2400	16	11	34	1	239	0.00%	7.6/sec	11.9
/styles/circle.c...	2400	15	10	33	1	359	0.00%	7.6/sec	3.6
/styles/bootstrap...	2400	35	27	70	5	406	0.00%	7.6/sec	902.6
/styles/cookie...	2400	14	10	29	1	214	0.00%	7.6/sec	31.5
/scripts/bootstrap...	2400	19	15	39	2	223	0.00%	7.6/sec	277.6
/scripts/jquery...	2400	29	21	59	4	405	0.00%	7.6/sec	724.4
/scripts/polyfill...	2400	13	10	28	2	193	0.00%	7.6/sec	57.3
/scripts/cookie...	2400	18	13	36	2	267	0.00%	7.6/sec	261.3
/scripts/help...	2400	12	9	26	1	147	0.00%	7.6/sec	17.4
/images/logo....	2400	274	199	595	29	2756	0.00%	7.6/sec	8571.8
/security/login...	2400	150	81	378	8	2144	0.00%	7.6/sec	47.4
/favicon.ico	2400	15	9	32	1	363	0.00%	7.6/sec	42.6
/j_spring_sec...	2400	184	113	416	21	2148	0.00%	7.6/sec	88.4
/images/link...	2400	93	61	208	10	1126	0.00%	7.6/sec	2482.5
/bargain/mod...	4800	133	74	317	13	1316	0.00%	15.2/sec	152.2
/bargain/mod...	2400	96	54	223	11	1212	0.00%	7.6/sec	102.1
/j_spring_sec...	2400	194	132	437	15	1560	0.00%	7.6/sec	83.1
TOTAL	50400	90	35	242	1	2756	0.00%	155.5/sec	13731.8

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 9327 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 90 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Throughput

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	225	142	532	14	2493	0.00%	22.4/sec	244.8
/styles/style.css	2700	19	12	42	1	453	0.00%	7.6/sec	11.9
/styles/circle.css	2700	19	11	40	0	465	0.00%	7.6/sec	3.6
/styles/bootstrap.min.css	2700	41	28	85	5	529	0.00%	7.6/sec	906.6
/styles/cookieconsent.min.css	2700	17	11	37	1	391	0.00%	7.6/sec	31.6
/scripts/bootstrap.min.js	2700	24	16	50	3	691	0.00%	7.6/sec	278.8
/scripts/jquery.min.js	2700	36	24	74	4	493	0.00%	7.6/sec	727.5
/scripts/polyfills.js	2700	17	11	39	2	886	0.00%	7.6/sec	57.8
/scripts/cookieconsent.js	2700	22	15	46	2	217	0.00%	7.6/sec	262.4
/scripts/helpers.js	2700	16	10	35	1	432	0.00%	7.6/sec	17.4
/images/logo.jpg	2700	274	188	592	29	1847	0.00%	7.6/sec	8615.0
/security/login.do	2700	195	114	499	8	1837	0.00%	7.6/sec	47.7
/favicon.ico	2700	20	12	43	2	359	0.00%	7.6/sec	42.8
/j_spring_security_check	2700	320	213	750	11	3011	0.00%	7.7/sec	88.8
/images/link_broken.png	2700	98	63	211	10	1116	0.00%	7.6/sec	2493.7
/bargain/moderator/list.do	5400	205	118	509	13	2116	0.00%	15.2/sec	152.9
/bargain/moderator/delete.do	2700	170	97	408	11	2194	0.00%	7.6/sec	102.5
/j_spring_security_logout	2700	266	180	619	18	1972	0.00%	7.6/sec	83.4
TOTAL	56700	126	42	355	0	3011	0.00%	156.6/sec	13834.8

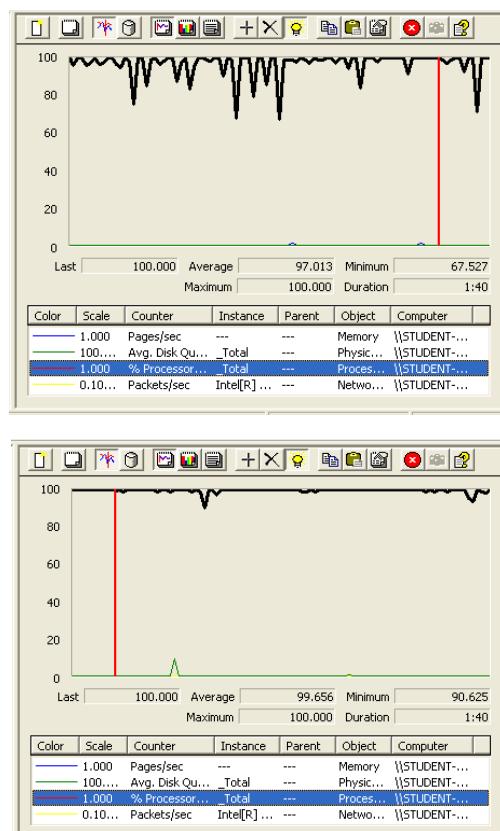
Include group name in label? Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 9397.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Borrar categoría

En este caso de uso, un moderador puede borrar una categoría. Para ello, sigue la siguiente secuencia de pasos:

Categorías > (Escoger una)

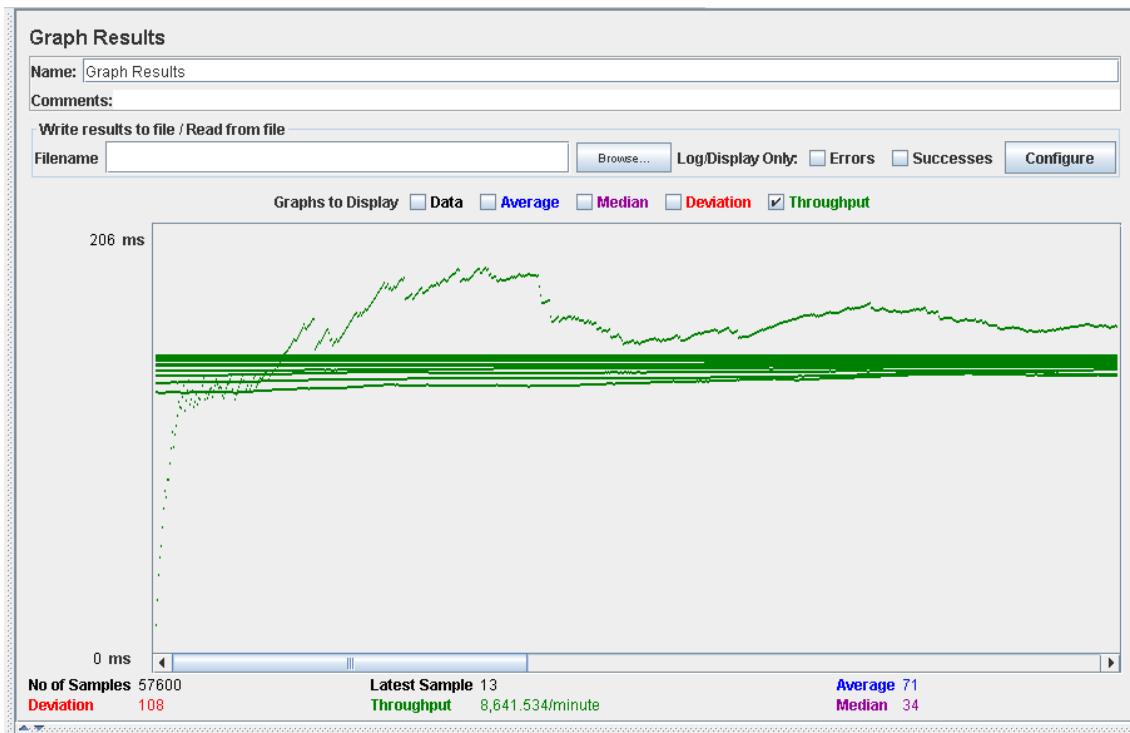
El límite de este caso de uso se sitúa en 80 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	7200	135	83	300	1	1655	0.00%	18.0/sec	197.0	
/styles/cookieconsent.min.css	2400	11	8	26	0	218	0.00%	6.2/sec	25.5	
/styles/style.css	2400	12	8	27	0	136	0.00%	6.2/sec	9.6	
/styles/bootstrap.min.css	2400	28	20	61	0	541	0.00%	6.2/sec	731.9	
/scripts/jquery.min.js	2400	25	18	51	0	689	0.00%	6.2/sec	587.2	
/scripts/circle.css	2400	11	8	25	0	481	0.00%	6.2/sec	2.9	
/scripts/bootstrap.min.js	2400	17	13	37	0	161	0.00%	6.2/sec	225.0	
/scripts/polyfills.js	2400	12	9	26	0	1102	0.00%	6.2/sec	46.4	
/scripts/helpers.js	2400	12	8	27	0	476	0.00%	6.2/sec	14.1	
/scripts/cookieconsent.js	2400	16	12	34	0	481	0.00%	6.2/sec	211.8	
/images/logo.jpg	2400	166	113	341	21	1228	0.00%	6.2/sec	6958.9	
/images/link_broken.png	4800	56	39	116	0	556	0.00%	12.0/sec	3918.0	
/favicon.ico	2400	13	9	29	0	392	0.00%	6.2/sec	34.5	
/security/login.do	2400	75	46	177	0	963	0.00%	6.2/sec	38.5	
/j_spring_security_check	2400	178	119	396	7	1474	0.00%	6.2/sec	71.7	
/category/display.do	7200	92	58	205	0	2450	0.00%	18.3/sec	198.6	
/category/moderator/edit.do	4800	96	61	218	0	1372	0.00%	12.4/sec	132.7	
/j_spring_security_logout	2400	143	92	305	3	1490	0.00%	6.2/sec	67.5	
TOTAL	57600	71	34	178	0	2450	0.00%	144.0/sec	13210.6	

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8641 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 90 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file

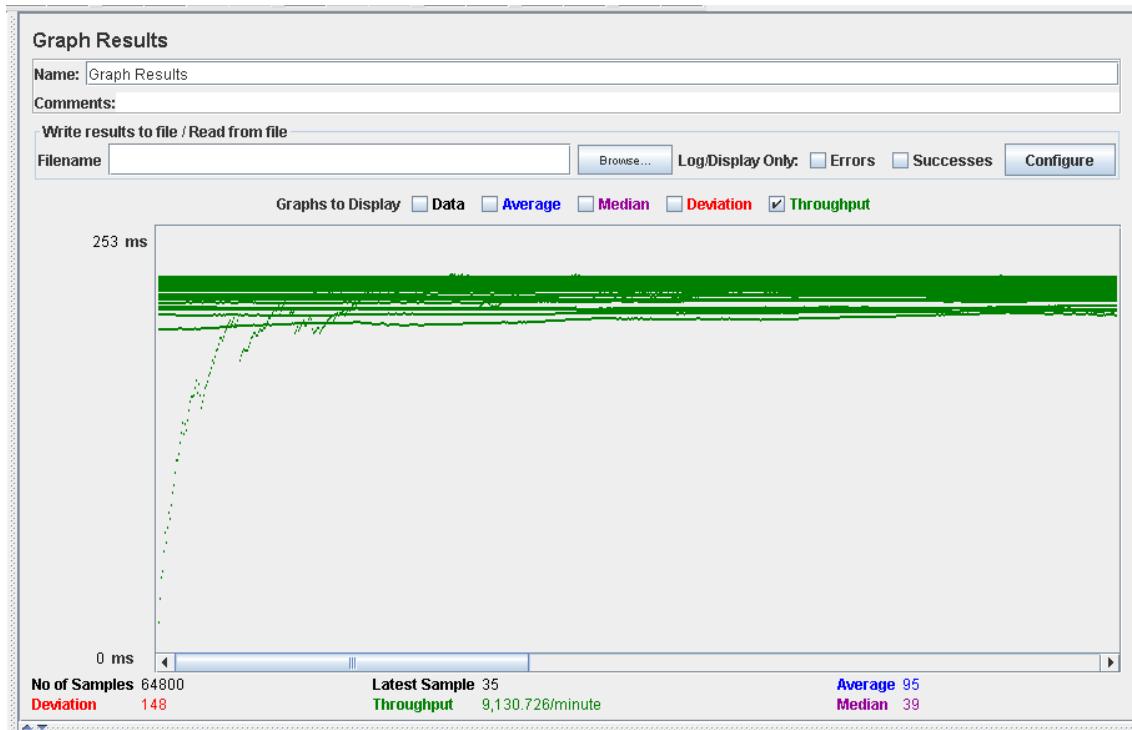
Filename Browse... Log/Display Only: Errors Successes Throughput KB/sec Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	182	114	430	0	2114	0.00%	19.0/sec	208.1
/styles/cookieconsent.min.css	2700	14	11	33	0	335	0.00%	6.5/sec	26.9
/styles/style.css	2700	15	11	32	0	860	0.00%	6.5/sec	10.2
/styles/bootstrap.min.css	2700	35	25	75	0	459	0.00%	6.5/sec	772.3
/scripts/jquery.min.js	2700	31	22	65	0	589	0.00%	6.5/sec	619.6
/styles/circle.css	2700	14	10	30	0	449	0.00%	6.5/sec	3.1
/scripts/bootstrap.min.js	2700	20	15	44	0	311	0.00%	6.5/sec	237.5
/scripts/polyfills.js	2700	14	10	30	0	127	0.00%	6.5/sec	49.0
/scripts/helpers.js	2700	14	10	29	0	213	0.00%	6.5/sec	14.9
/scripts/cookieconsent.js	2700	20	14	40	0	944	0.00%	6.5/sec	223.5
/images/logo.jpg	2700	203	141	423	23	1344	0.00%	6.5/sec	7341.9
/images/link_broken.png	5400	71	49	148	0	923	0.00%	12.7/sec	4140.2
/favicon.ico	2700	15	11	34	0	219	0.00%	6.5/sec	36.4
/security/login.do	2700	113	65	272	0	1901	0.00%	6.5/sec	40.6
/j_spring_security_check	2700	237	159	541	9	1992	0.00%	6.5/sec	75.7
/category/display.do	8100	129	81	293	0	2883	0.00%	19.3/sec	209.5
/category/moderator/edit.do	5400	131	83	304	0	2855	0.00%	13.0/sec	138.9
/j_spring_security_logout	2700	189	122	433	2	2182	0.00%	6.5/sec	71.2
TOTAL	64800	95	39	245	0	2883	0.00%	152.2/sec	13958.5

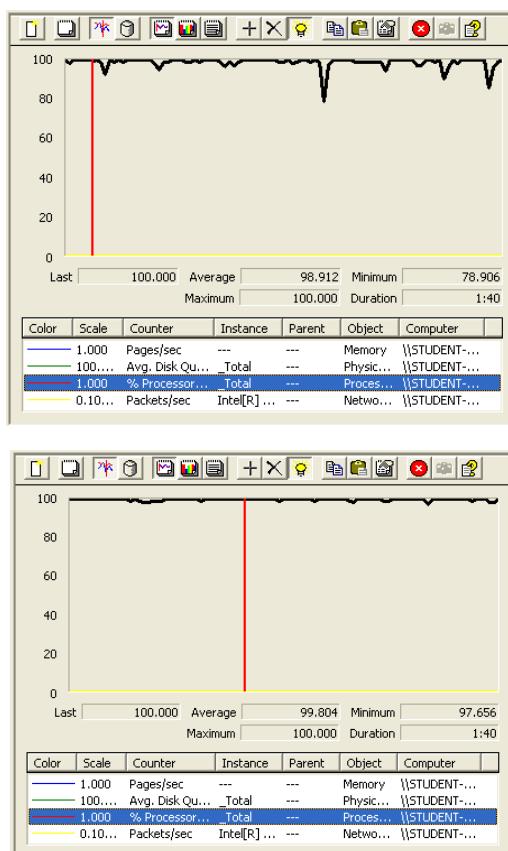
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 9130.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Borrar notificación

En este caso de uso, un usuario puede borrar una notificación. Para ello, sigue la siguiente secuencia de pasos:

Mis notificaciones > (Escoger una)

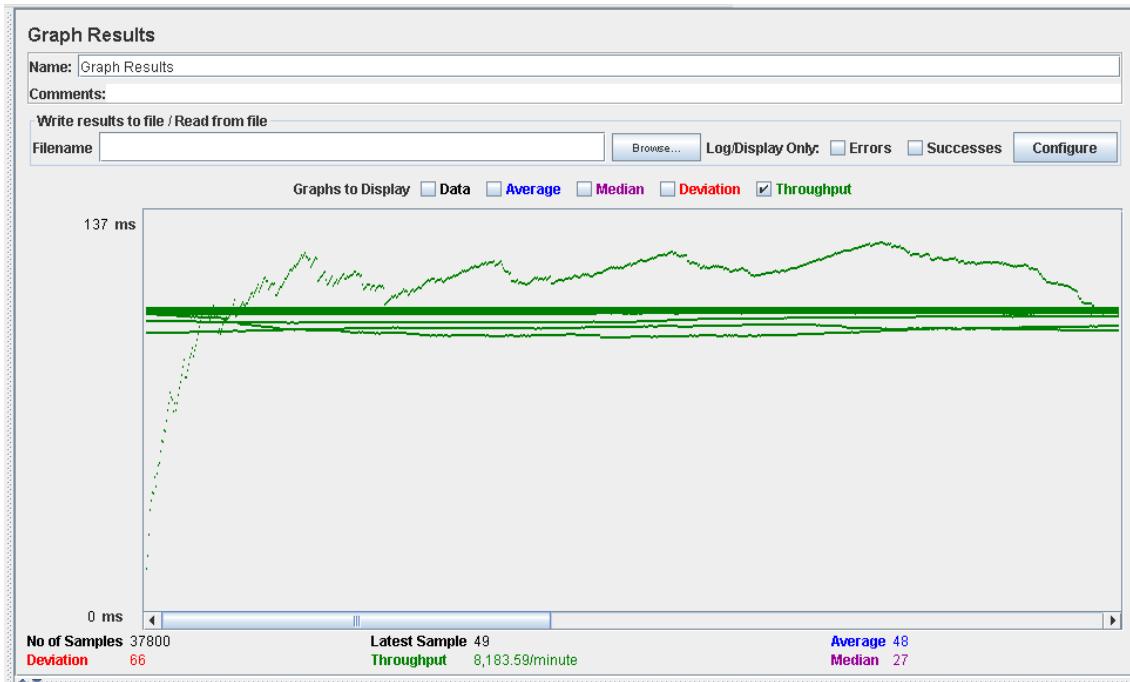
El límite de este caso de uso se sitúa en 60 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	5400	80	55	169	1	958	0.00%	19.5/sec	213.9	
/styles/circle.css	1800	9	7	22	0	194	0.00%	6.7/sec	3.2	
/styles/bootstrap.min.css	1800	26	20	53	0	405	0.00%	6.7/sec	792.5	
/styles/cookieconsent.min.c...	1800	10	7	24	0	186	0.00%	6.7/sec	27.6	
/scripts/jquery.min.js	1800	22	18	45	0	284	0.00%	6.7/sec	635.9	
/styles/style.css	1800	9	7	22	0	248	0.00%	6.7/sec	10.4	
/scripts/bootstrap.min.js	1800	14	11	31	0	195	0.00%	6.7/sec	243.8	
/scripts/polyfills.js	1800	9	7	23	0	118	0.00%	6.7/sec	50.3	
/scripts/helpers.js	1800	10	7	24	0	114	0.00%	6.7/sec	15.3	
/scripts/cookieconsent.js	1800	14	11	30	0	152	0.00%	6.7/sec	229.5	
/images/logo.jpg	1800	152	116	296	20	1341	0.00%	6.7/sec	7536.4	
/favicon.ico	1800	10	8	24	0	127	0.00%	6.7/sec	37.4	
/security/login.do	1800	39	29	85	0	301	0.00%	6.7/sec	41.7	
/j_spring_security_check	1800	107	77	217	12	1240	0.00%	6.7/sec	78.5	
/images/link_broken.png	1800	49	39	101	0	348	0.00%	6.7/sec	2186.4	
/notification/actor/list.do	3600	66	47	135	0	781	0.00%	13.3/sec	120.5	
/notification/actor/delete.do	1800	60	42	124	0	965	0.00%	6.7/sec	98.4	
/j_spring_security_logout	1800	89	62	182	4	1084	0.00%	6.7/sec	73.1	
TOTAL	37800	48	27	119	0	1341	0.00%	136.4/sec	12043.9	

Como podemos observar, a la hora de cargar la página principal, tarda más que en las demás operaciones.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8183 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 70 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

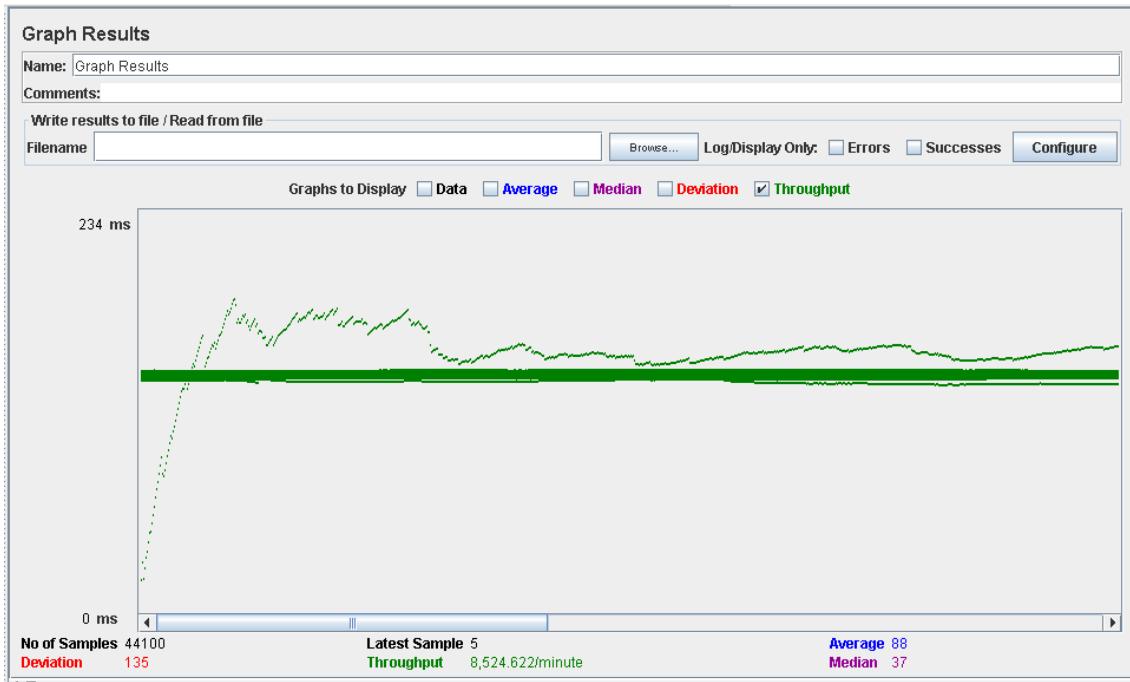
Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	6300	164	103	369	1	1502	0.00%	20.3/sec	222.8
/styles/circle.css	2100	15	11	33	0	495	0.00%	6.9/sec	3.3
/styles/bootstrap.min.css	2100	38	27	83	0	641	0.00%	6.9/sec	823.2
/styles/cookieconsent.min.c...	2100	15	11	34	0	443	0.00%	6.9/sec	28.7
/scripts/jquery.min.js	2100	33	23	71	0	364	0.00%	6.9/sec	660.5
/styles/style.css	2100	14	10	32	0	228	0.00%	6.9/sec	10.8
/scripts/bootstrap.min.js	2100	21	15	47	0	179	0.00%	6.9/sec	253.2
/scripts/polyfills.js	2100	15	11	33	0	120	0.00%	6.9/sec	52.3
/scripts/helpers.js	2100	16	11	35	0	415	0.00%	6.9/sec	15.8
/scripts/cookieconsent.js	2100	21	15	47	0	205	0.00%	6.9/sec	238.3
/images/logo.jpg	2100	222	163	459	18	1388	0.00%	6.9/sec	7828.8
/favicon.ico	2100	18	12	37	0	269	0.00%	6.9/sec	38.8
/security/login.do	2100	93	58	217	0	1113	0.00%	6.9/sec	43.3
/j_spring_security_check	2100	226	152	500	7	1903	0.00%	6.9/sec	81.6
/images/link_broken.png	2100	78	54	185	0	727	0.00%	6.9/sec	2265.7
/notification/actor/list.do	4200	126	84	284	0	1171	0.00%	13.8/sec	125.1
/notification/actor/delete.do	2100	116	74	265	0	1153	0.00%	6.9/sec	102.0
/j_spring_security_logout	2100	175	113	400	4	1593	0.00%	7.0/sec	75.9
TOTAL	44100	88	37	237	0	1903	0.00%	142.1/sec	12545.8

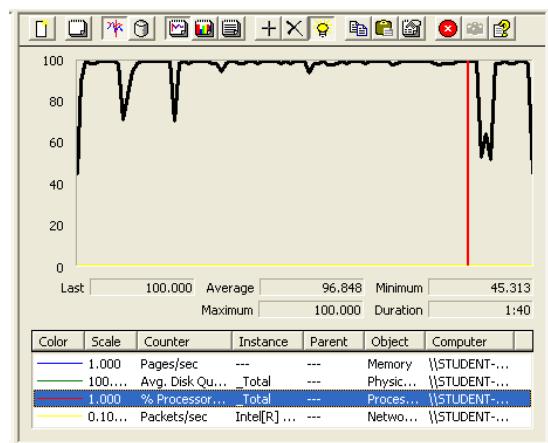
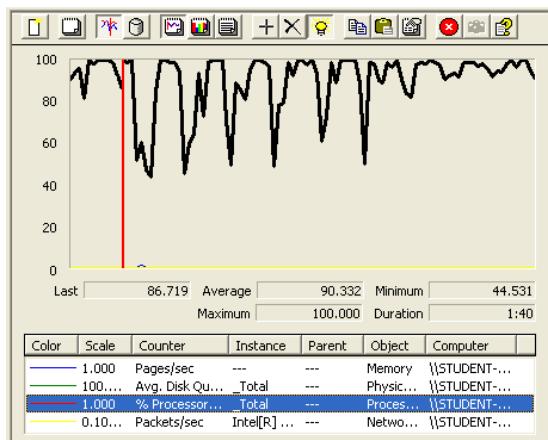
Include group name in label? Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 8524.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador.
Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Borrar patrocinio

En este caso de uso, un patrocinador puede borrar sus patrocinios. Para ello, sigue la siguiente secuencia de pasos:

Mis patrocinios > (Escoger uno)

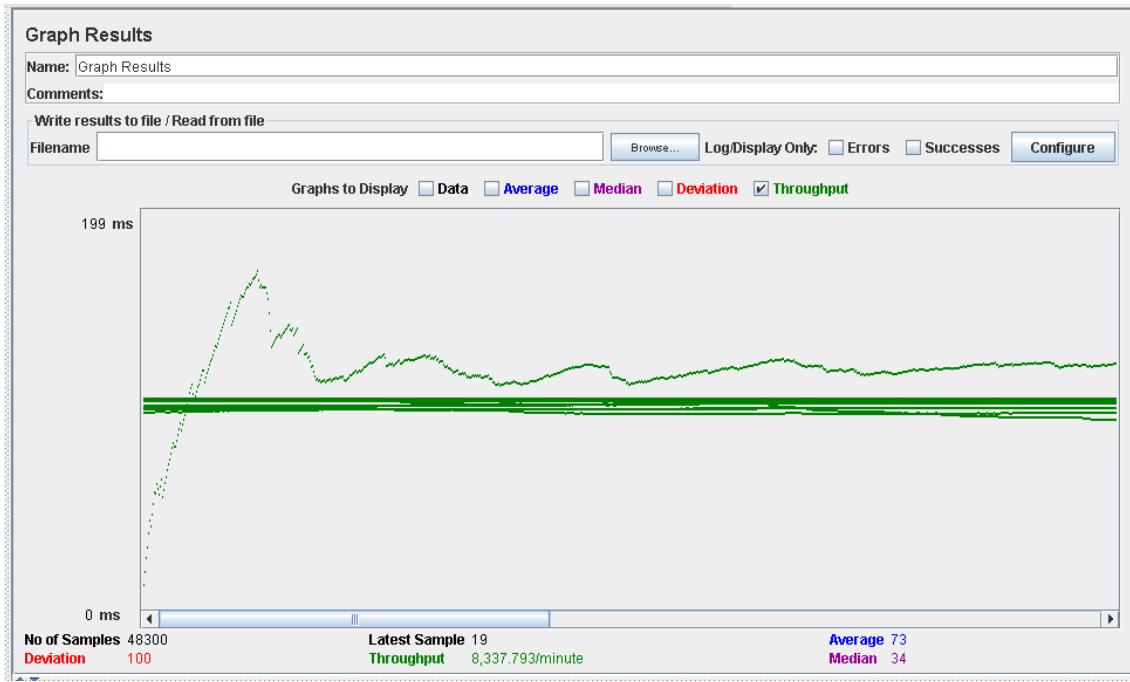
El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report																	
Name: Aggregate Report																	
Comments:																	
Write results to file / Read from file																	
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec								
/	6300	145	111	306	0	2154	0.00%	18.1/sec	196.5								
/styles/style.css	2100	9	7	23	0	294	0.00%	6.2/sec	9.7								
/styles/cookieconsent.min.css	2100	10	7	24	0	747	0.00%	6.2/sec	25.7								
/scripts/jquery.min.js	2100	22	18	45	0	289	0.00%	6.2/sec	591.4								
/styles/bootstrap.min.css	2100	26	21	52	0	223	0.00%	6.2/sec	737.2								
/scripts/bootstrap.min.js	2100	14	11	30	0	624	0.00%	6.2/sec	226.7								
/styles/circle.css	2100	9	7	21	0	254	0.00%	6.2/sec	3.0								
/scripts/polyfills.js	2100	9	7	22	0	190	0.00%	6.2/sec	46.8								
/scripts/helpers.js	2100	9	7	23	0	219	0.00%	6.2/sec	14.2								
/scripts/cookieconsent.js	2100	13	11	29	0	690	0.00%	6.2/sec	213.4								
/images/logo.jpg	2100	156	131	281	16	1107	0.00%	6.2/sec	7008.3								
/favicon.ico	2100	11	8	25	0	878	0.00%	6.2/sec	34.8								
/security/login.do	2100	77	56	167	0	1033	0.00%	6.2/sec	38.8								
/j_spring_security_check	2100	179	143	356	3	1262	0.00%	6.2/sec	70.5								
/images/link_broken.png	4200	54	42	106	0	982	0.00%	12.3/sec	3998.5								
/sponsorship/sponsor/list.do	4200	120	93	250	0	1143	0.00%	12.3/sec	122.8								
/sponsorship/sponsor/edit.do	4200	103	76	214	0	1222	0.00%	12.4/sec	136.2								
/j_spring_security_logout	2100	148	111	311	0	1191	0.00%	6.2/sec	67.8								
TOTAL	48300	73	34	199	0	2154	0.00%	139.0/sec	13223.7								

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8337 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

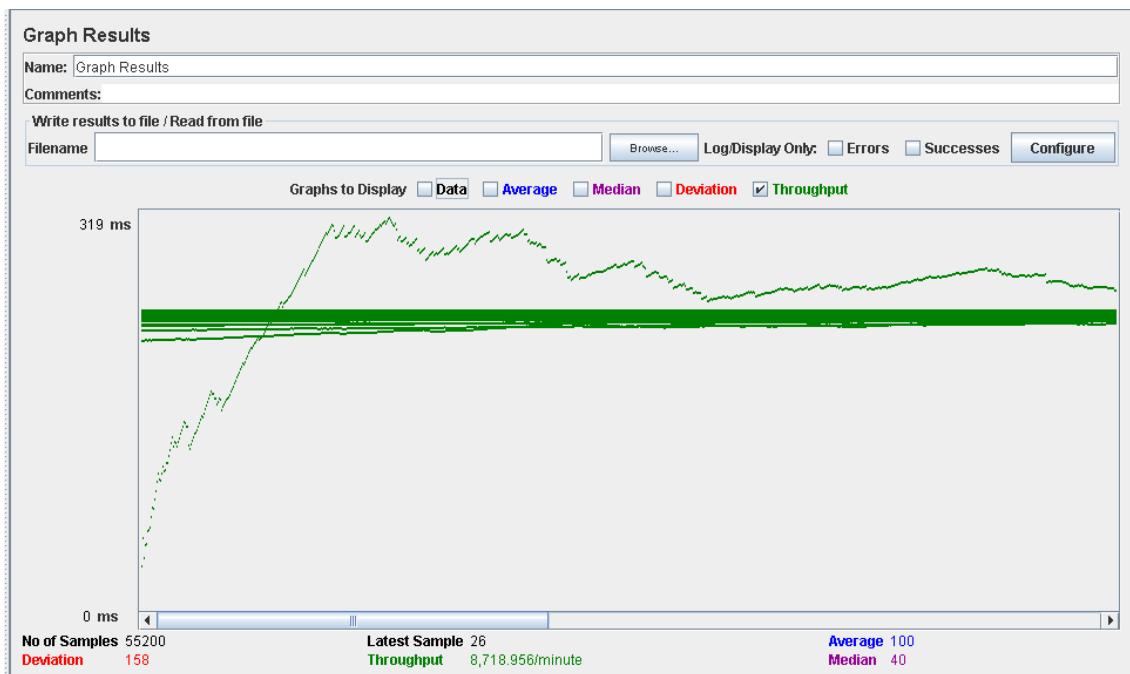
Write results to file / Read from file
Filename: Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	188	124	431	0	1782	0.00%	19.0/sec	205.5
/styles/style.css	2400	15	11	33	0	315	0.00%	6.5/sec	10.1
/styles/cookieconsent.min.css	2400	15	11	32	0	178	0.00%	6.5/sec	26.8
/scripts/jquery.min.js	2400	33	27	67	0	348	0.00%	6.5/sec	617.2
/styles/bootstrap.min.css	2400	39	30	80	0	563	0.00%	6.5/sec	769.4
/scripts/bootstrapstrap.min.js	2400	22	16	46	0	353	0.00%	6.5/sec	236.7
/styles/circle.css	2400	15	11	33	0	381	0.00%	6.5/sec	3.1
/scripts/polyfills.js	2400	15	11	34	0	177	0.00%	6.5/sec	48.9
/scripts/helpers.js	2400	15	11	34	0	205	0.00%	6.5/sec	14.8
/scripts/cookieconsent.js	2400	20	15	41	0	168	0.00%	6.5/sec	222.9
/images/logo.jpg	2400	222	170	446	16	1310	0.00%	6.5/sec	7323.2
/favicon.ico	2400	15	11	34	0	168	0.00%	6.5/sec	36.3
/security/login.do	2400	113	65	255	0	1939	0.00%	6.5/sec	40.5
/j_spring_security_check	2400	254	166	582	0	2736	0.00%	6.5/sec	73.7
/images/link_broken.png	4800	72	52	152	0	713	0.00%	12.0/sec	4184.7
/sponsorship/sponsor/list.do	4800	156	100	351	0	1913	0.00%	12.0/sec	128.4
/sponsorship/sponsor/edit.do	4800	138	84	322	0	3534	0.00%	13.0/sec	142.3
/j_spring_security_logout	2400	200	132	433	0	2267	0.00%	6.5/sec	70.8
TOTAL	55200	100	40	266	0	3534	0.00%	145.3/sec	13828.2

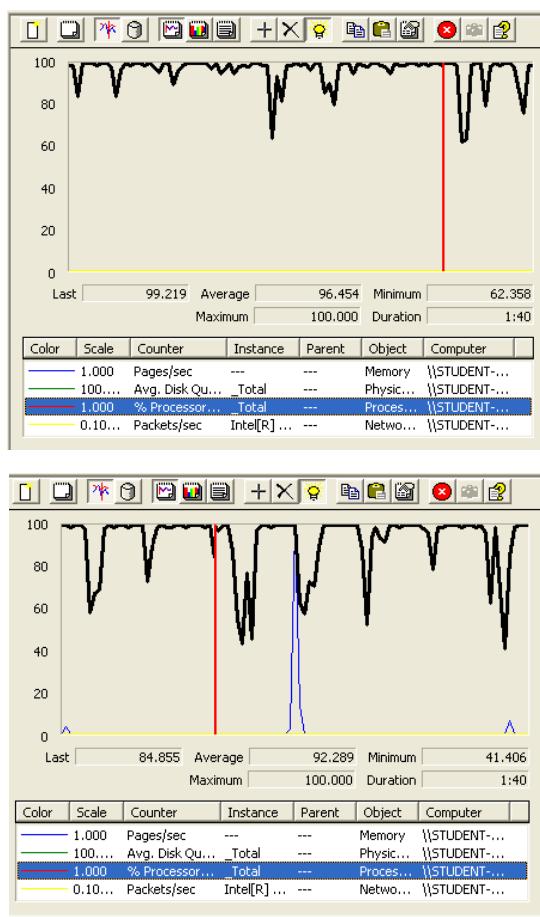
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 8718.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y de la tarjeta de red. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Desplegar sobre nosotros

En este caso de uso, cualquier usuario puede ver información sobre nosotros. Para ello, sigue la siguiente secuencia de pasos:

Sobre nosotros

El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										Browse...
Filename										Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="checkbox"/> Configure
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/welcome/index...	3000	159	137	302	0	1035	0.00%	23.7/sec	253.5	
/styles/bootstrap...	1500	26	21	50	0	576	0.00%	12.2/sec	1441.9	
/scripts/jquery....	1500	22	17	42	0	168	0.00%	12.2/sec	1156.7	
/styles/cookiec...	1500	9	7	22	0	305	0.00%	12.2/sec	50.3	
/styles/style.css	1500	9	7	21	0	278	0.00%	12.2/sec	19.0	
/styles/circle.css	1500	9	7	21	0	277	0.00%	12.2/sec	5.8	
/scripts/polyfill...	1500	9	7	20	0	300	0.00%	12.2/sec	91.5	
/scripts/bootstrap...	1500	14	11	29	0	207	0.00%	12.2/sec	443.7	
/scripts/helper....	1500	8	7	21	0	104	0.00%	12.2/sec	27.7	
/scripts/cookie...	1500	13	11	28	0	171	0.00%	12.2/sec	417.6	
/images/link_b...	1500	56	45	111	0	481	0.00%	12.2/sec	3965.9	
/images/logo.jpg	1500	156	135	294	16	598	0.00%	12.2/sec	13712.0	
/favicon.ico	1500	11	8	25	0	148	0.00%	12.2/sec	68.0	
/aboutUs/displ...	1500	88	74	174	0	478	0.00%	12.2/sec	82.1	
TOTAL	22500	50	16	154	0	1035	0.00%	177.6/sec	21172.4	

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 10657 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

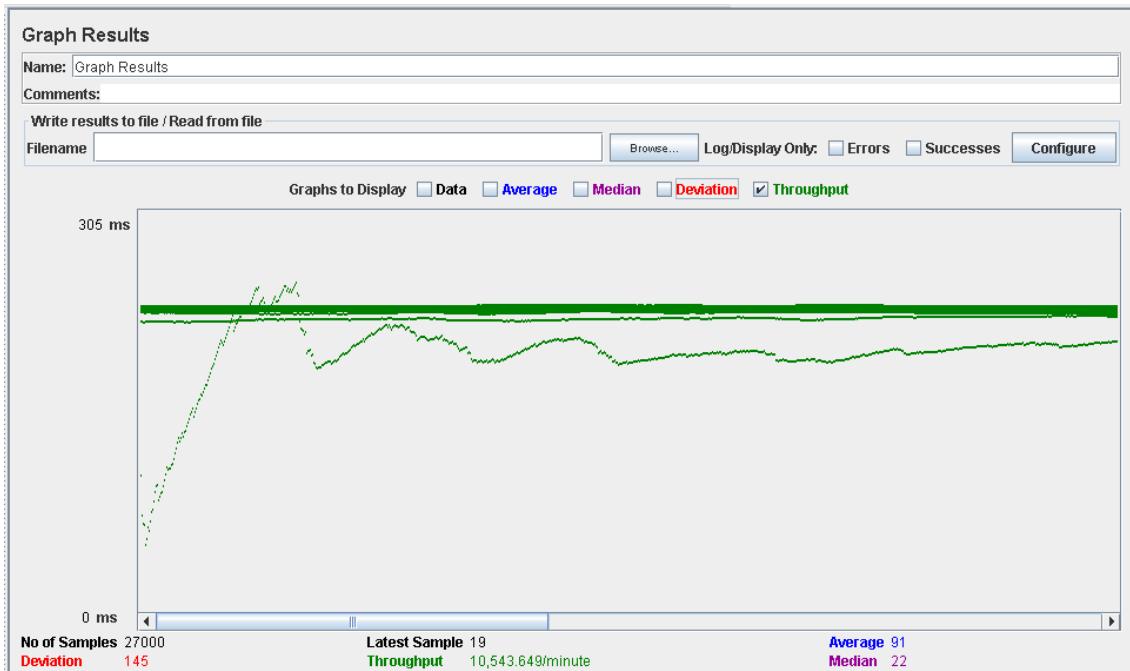
Aggregate Report

Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

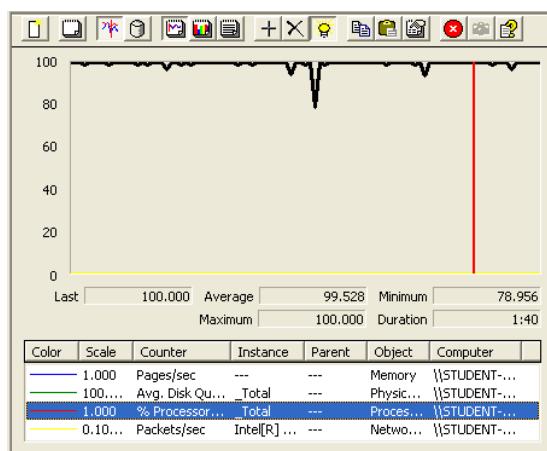
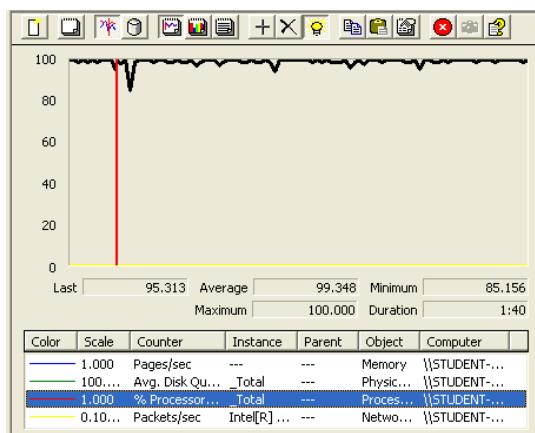
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/welcome/index...	3600	318	301	551	0	2003	0.00%	23.4/sec	250.8
/styles/bootstrap...	1800	38	35	71	0	213	0.00%	12.0/sec	1418.8
/scripts/jquery...	1800	33	29	80	0	526	0.00%	12.0/sec	1138.5
/styles/cookiec...	1800	12	10	22	0	652	0.00%	12.0/sec	49.4
/styles/style.css	1800	11	10	22	0	110	0.00%	12.0/sec	18.7
/styles/circle.css	1800	13	10	25	0	410	0.00%	12.0/sec	5.7
/scripts/polyfill...	1800	12	10	23	0	93	0.00%	12.0/sec	90.1
/scripts/bootstrap...	1800	20	17	36	0	216	0.00%	12.0/sec	436.5
/scripts/helper...	1800	12	10	22	0	413	0.00%	12.0/sec	27.3
/scripts/cookie...	1800	18	16	33	0	160	0.00%	12.0/sec	411.0
/images/link_b...	1800	92	85	184	0	756	0.00%	12.0/sec	3905.0
/images/logo.jpg	1800	256	248	452	28	1334	0.00%	12.0/sec	13506.3
/favicon.ico	1800	14	12	28	0	285	0.00%	12.0/sec	67.1
/aboutUs/displ...	1800	195	169	360	0	1052	0.00%	12.0/sec	80.9
TOTAL	27000	91	22	305	0	2003	0.00%	175.7/sec	20946.3

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 10543.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador.
Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Desplegar términos y condiciones

En este caso de uso, cualquier un usuario puede ver los términos y condiciones del sistema.
Para ello, sigue la siguiente secuencia de pasos:

Términos y condiciones

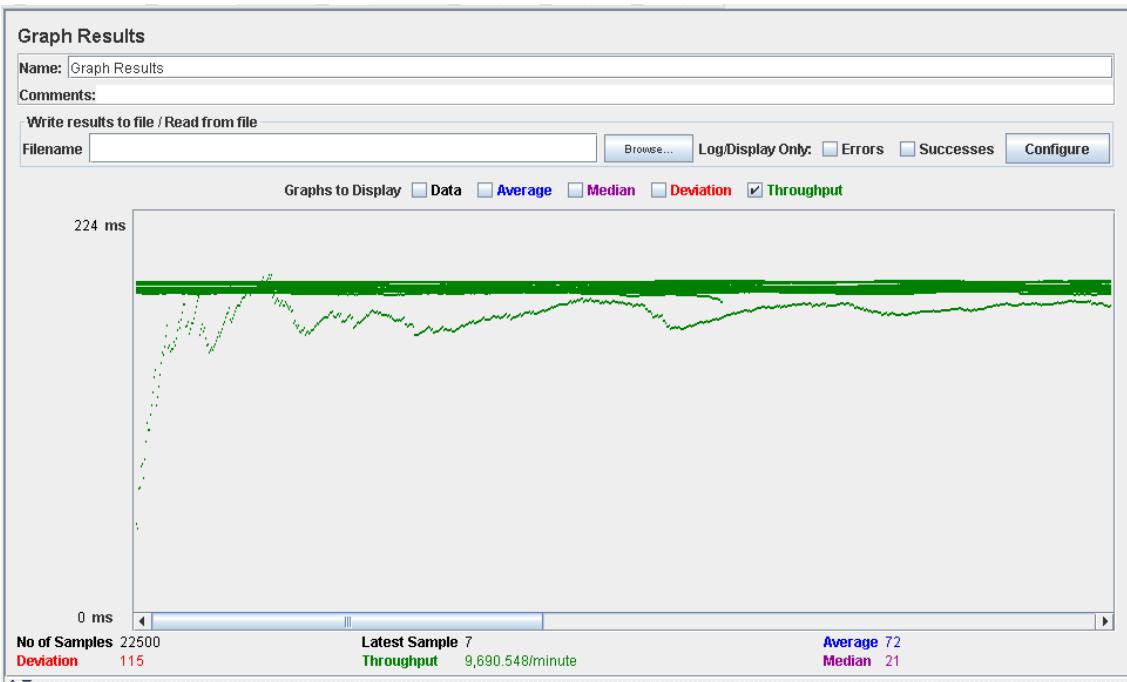
El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	1500	239	207	461	0	934	0.00%	11.1/sec	119.2	
/styles/circle.css	1500	12	9	25	0	214	0.00%	11.1/sec	5.3	
/styles/cookieconsent.min.css	1500	12	9	25	0	202	0.00%	11.1/sec	45.7	
/scripts/jquery.min.js	1500	29	24	55	0	266	0.00%	11.1/sec	1052.3	
/styles/style.css	1500	11	9	25	0	108	0.00%	11.1/sec	17.3	
/styles/bootstrap.min.css	1500	35	29	65	0	625	0.00%	11.1/sec	1311.8	
/scripts/bootstrap.min.js	1500	18	15	35	0	194	0.00%	11.1/sec	403.4	
/scripts/polyfills.js	1500	12	9	24	0	520	0.00%	11.1/sec	83.3	
/scripts/cookieconsent.js	1500	16	14	33	0	210	0.00%	11.1/sec	379.6	
/scripts/helpers.js	1500	11	9	24	0	162	0.00%	11.1/sec	25.3	
/images/link_broken.png	1500	77	67	144	0	380	0.00%	11.1/sec	3607.2	
/images/logo.jpg	1500	210	195	390	14	787	0.00%	11.1/sec	12477.9	
/favicon.ico	1500	14	10	30	0	186	0.00%	11.1/sec	61.9	
/termCondition/display.do	1500	154	131	299	0	1131	0.00%	11.0/sec	81.0	
/welcome/index.do	1500	237	207	449	0	1022	0.00%	11.0/sec	117.3	
TOTAL	22500	72	21	230	0	1131	0.00%	161.5/sec	19257.8	

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 9690 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Throughput

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1800	339	297	669	0	2105	0.00%	11.7/sec	126.7
/styles/circle.css	1800	16	11	34	0	500	0.00%	11.7/sec	5.6
/styles/cookieconsent.min.css	1800	15	11	32	0	256	0.00%	11.7/sec	48.6
/scripts/jquery.min.js	1800	38	29	76	0	956	0.00%	11.7/sec	1118.2
/styles/style.css	1800	15	11	31	0	495	0.00%	11.7/sec	18.4
/styles/bootstrap.min.css	1800	43	34	84	0	462	0.00%	11.7/sec	1394.0
/scripts/bootstrap.min.js	1800	23	18	48	0	324	0.00%	11.7/sec	428.6
/scripts/polyfills.js	1800	15	12	30	0	304	0.00%	11.7/sec	88.5
/scripts/cookieconsent.js	1800	22	16	45	0	166	0.00%	11.8/sec	403.4
/scripts/helpers.js	1800	15	11	31	0	373	0.00%	11.8/sec	26.8
/images/link_broken.png	1800	93	76	188	0	552	0.00%	11.8/sec	3832.7
/images/logo.jpg	1800	248	218	473	18	1291	0.00%	11.8/sec	13254.7
/favicon.ico	1800	17	13	38	0	232	0.00%	11.8/sec	65.8
/termCondition/display.do	1800	228	192	462	0	1499	0.00%	11.8/sec	86.4
/welcome/index.do	1800	336	297	662	0	1538	0.00%	11.8/sec	125.3
TOTAL	27000	97	25	320	0	2105	0.00%	172.7/sec	20587.4

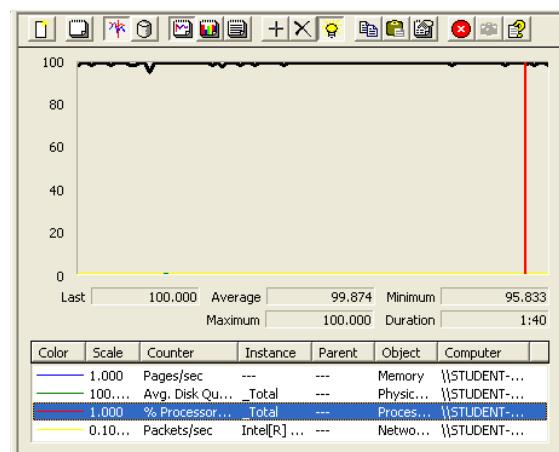
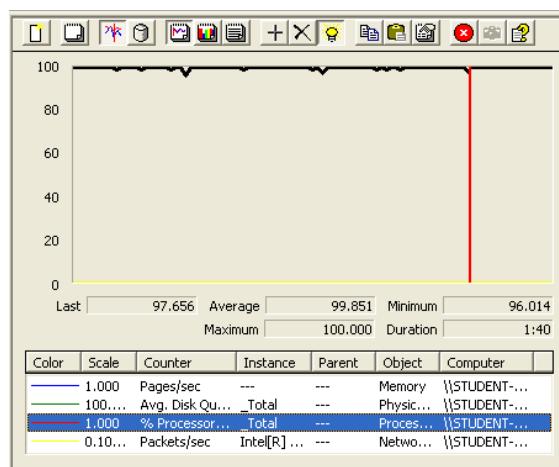
Include group name in label? Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 10359.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador.
Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Editar chollo

En este caso de uso, una compañía puede editar sus chollos. Para ello, sigue la siguiente secuencia de pasos:

Mis chollos > (Escoger uno)

El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name:	Aggregate Report								
Comments:									
Write results to file / Read from file									
Filename	<input type="text"/>			<input type="button" value="Browse..."/>	<input type="checkbox"/> Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	6300	135	83	314	13	1345	0.00%	17.8/sec	193.4
/styles/cookieconsent.min.css	2100	12	8	25	1	188	0.00%	6.1/sec	25.2
/styles/bootstrap.min.css	2100	28	19	58	5	785	0.00%	6.1/sec	721.9
/scripts/bootstrap.min.js	2100	17	12	35	3	465	0.00%	6.1/sec	222.0
/scripts/jquery.min.js	2100	24	17	48	5	273	0.00%	6.1/sec	579.2
/styles/style.css	2100	12	8	26	1	229	0.00%	6.1/sec	9.5
/styles/circle.css	2100	11	8	24	1	149	0.00%	6.1/sec	2.9
/scripts/polyfills.js	2100	12	8	26	1	160	0.00%	6.1/sec	45.8
/scripts/helpers.js	2100	11	8	24	2	117	0.00%	6.1/sec	13.9
/scripts/cookieconsent.js	2100	15	11	31	2	260	0.00%	6.1/sec	208.9
/images/link_broken.png	2100	56	40	118	9	443	0.00%	6.1/sec	1984.5
/images/logo.jpg	2100	159	113	332	30	1036	0.00%	6.1/sec	6863.5
/favicon.ico	2100	13	9	27	1	131	0.00%	6.1/sec	34.1
/security/login.do	2100	77	45	181	8	1872	0.00%	6.1/sec	38.0
/j_spring_security_check	2100	176	117	397	10	1227	0.00%	6.1/sec	69.5
/bargain/companylist.do	4200	107	67	249	13	1202	0.00%	12.1/sec	123.6
/bargain/companyedit.do	4200	179	120	406	14	1480	0.00%	12.2/sec	127.3
/j_spring_security_logout	2100	144	90	337	17	1310	0.00%	6.1/sec	66.5
TOTAL	46200	79	33	215	1	1872	0.00%	130.4/sec	11041.8

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 7826 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

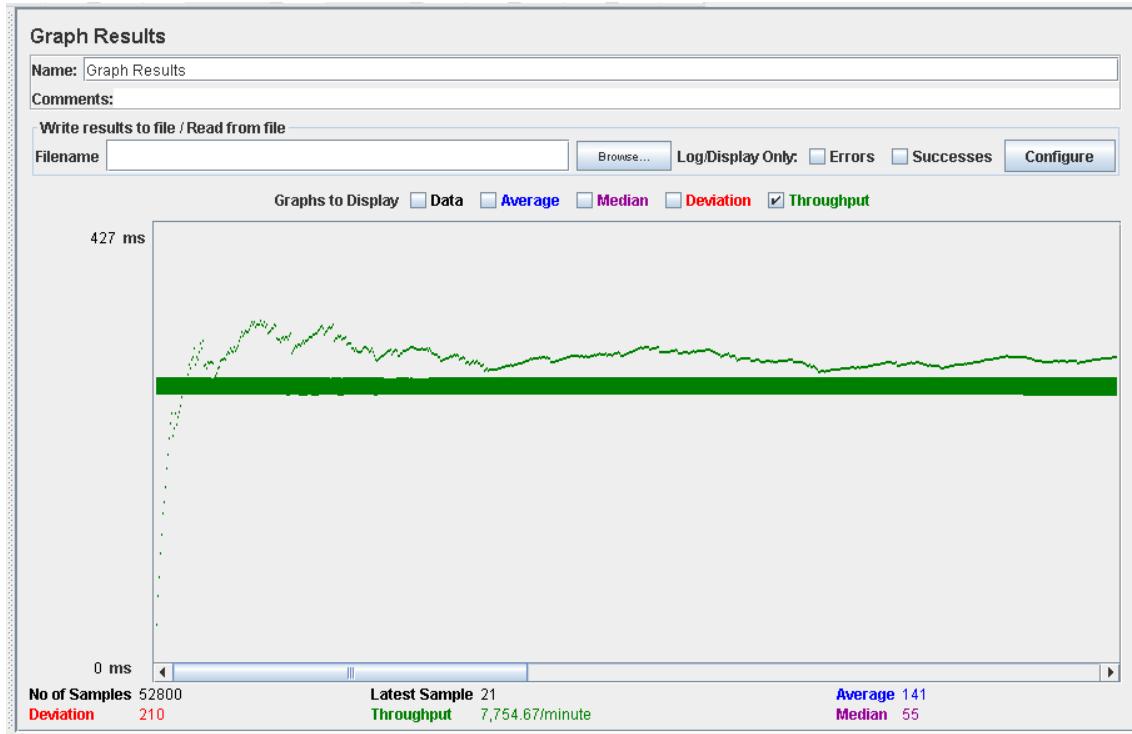
Write results to file / Read from file
Filename: Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	233	144	548	14	2111	0.00%	17.6/sec	191.7
/styles/cookieconsent.min.css	2400	22	14	50	1	448	0.00%	6.0/sec	24.9
/styles/bootstrap.min.css	2400	48	32	103	5	628	0.00%	6.0/sec	713.5
/scripts/bootstrap.min.js	2400	29	19	62	3	557	0.00%	6.0/sec	219.4
/scripts/jquery.min.js	2400	43	30	88	5	987	0.00%	6.0/sec	572.4
/styles/style.css	2400	21	14	47	1	567	0.00%	6.0/sec	9.4
/styles/circle.css	2400	21	14	48	2	423	0.00%	6.0/sec	2.9
/scripts/polyfills.js	2400	22	15	51	2	394	0.00%	6.0/sec	45.3
/scripts/helpers.js	2400	21	14	47	1	516	0.00%	6.0/sec	13.7
/scripts/cookieconsent.js	2400	28	19	61	2	370	0.00%	6.0/sec	206.5
/images/link_broken.png	2400	99	64	222	11	817	0.00%	6.0/sec	1960.8
/images/logo.jpg	2400	269	184	577	37	1791	0.00%	6.0/sec	6780.9
/favicon.ico	2400	23	16	51	2	434	0.00%	6.0/sec	33.6
/security/login.do	2400	162	101	378	8	2353	0.00%	6.0/sec	37.5
/j_spring_security_check	2400	337	238	727	21	2952	0.00%	6.0/sec	68.7
/bargain/companylist.do	4800	198	127	453	13	2186	0.00%	11.9/sec	122.0
/bargain/company/edit.do	4800	301	204	680	14	3252	0.00%	12.0/sec	125.5
/j_spring_security_logout	2400	252	184	571	15	2394	0.00%	6.0/sec	65.7
TOTAL	52800	141	55	399	1	3252	0.00%	129.2/sec	10940.5

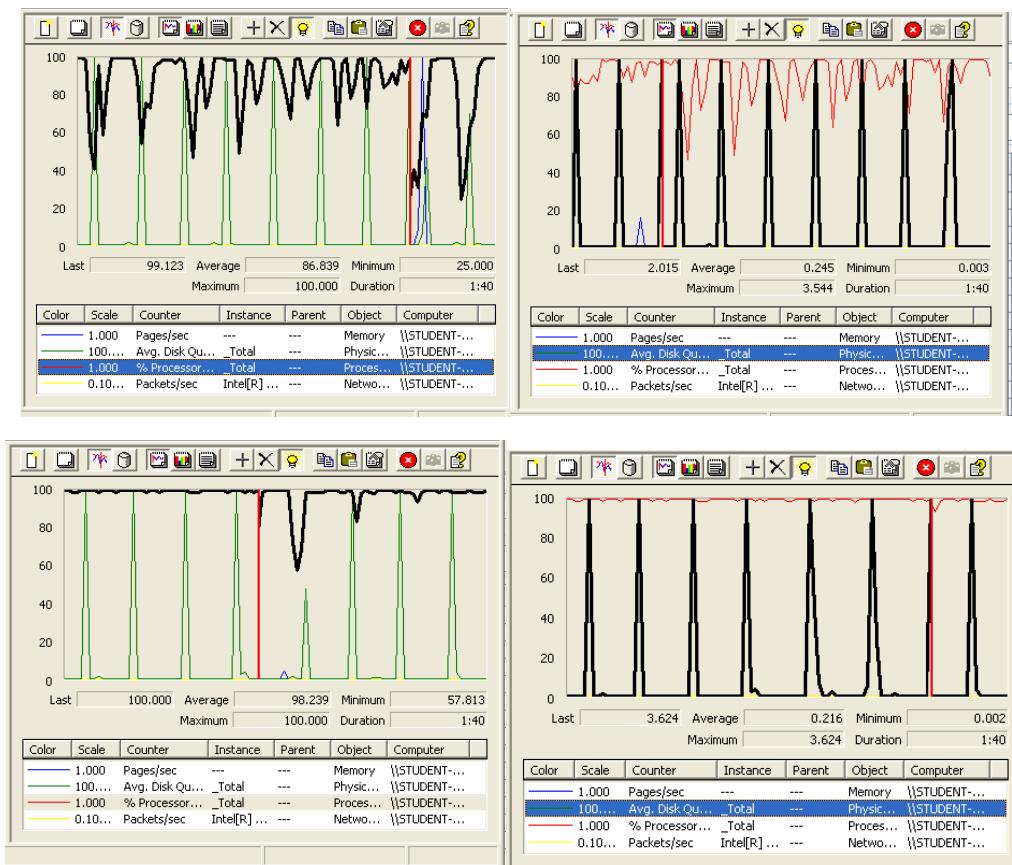
Include group name in label? Save Table Header

En este caso vemos como a la hora de iniciar sesión, el sistema tarda en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 7754.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y el disco duro. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



Editar categoría

En este caso de uso, un moderador puede editar categorías. Para ello, sigue la siguiente secuencia de pasos:

Categorías > (Escoger una)

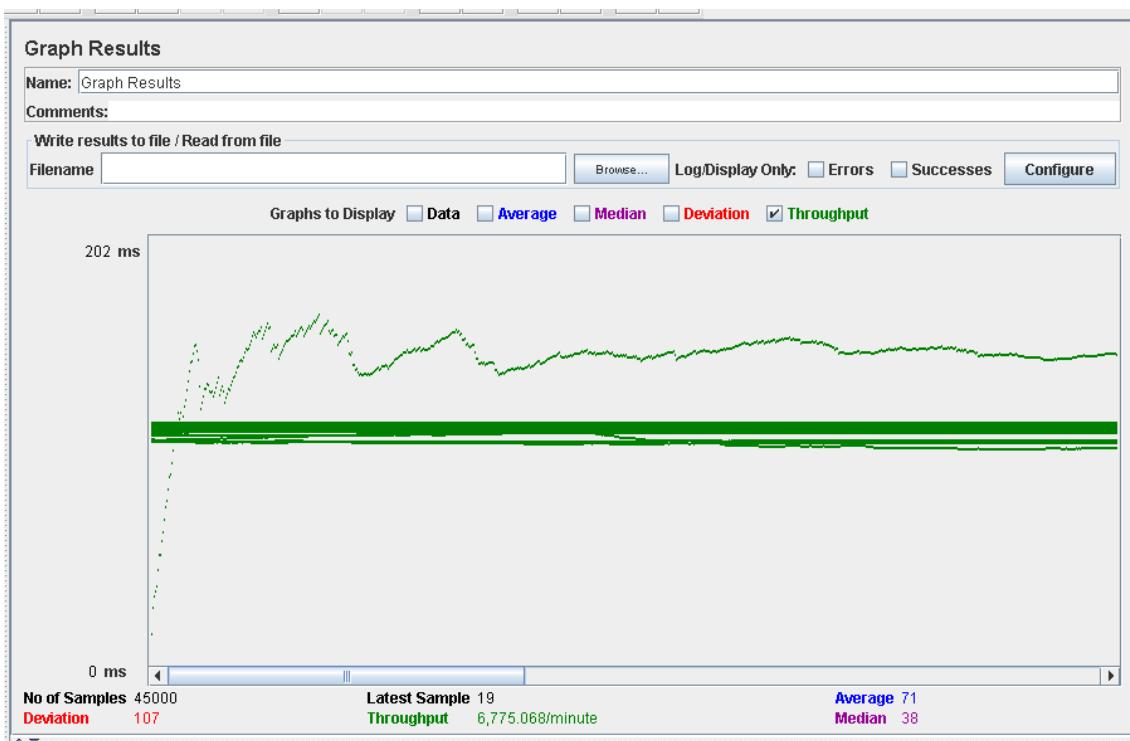
El límite de este caso de uso se sitúa en 60 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	5400	110	61	248	0	1597	0.00%	13.6/sec	148.3	
/styles/style.css	1800	11	7	27	0	170	0.00%	4.7/sec	7.3	
/styles/bootstrap.min.css	1800	27	20	54	0	452	0.00%	4.7/sec	553.3	
/styles/circle.css	1800	10	7	24	0	174	0.00%	4.7/sec	2.2	
/styles/cookieconsent.min.css	1800	11	7	25	0	701	0.00%	4.7/sec	19.3	
/scripts/jquery.min.js	1800	24	17	53	0	533	0.00%	4.7/sec	444.0	
/scripts/polyfills.js	1800	10	7	26	0	146	0.00%	4.7/sec	35.1	
/scripts/bootstrap.min.js	1800	15	11	35	0	276	0.00%	4.7/sec	170.2	
/scripts/helpers.js	1800	10	7	26	0	162	0.00%	4.7/sec	10.6	
/scripts/cookieconsent.js	1800	14	10	33	0	161	0.00%	4.7/sec	160.1	
/images/logo.jpg	1800	159	112	317	23	1524	0.00%	4.7/sec	5259.4	
/images/link_broken.png	3600	55	39	114	0	428	0.00%	9.0/sec	2948.0	
/favicon.ico	1800	12	8	28	0	110	0.00%	4.7/sec	26.1	
/security/login.do	1800	62	37	139	0	907	0.00%	4.7/sec	29.1	
/j_spring_security_check	1800	138	83	309	8	1387	0.00%	4.7/sec	54.2	
/category/display.do	5400	109	80	220	0	1567	0.00%	13.8/sec	123.3	
/wp-content/uploads/Default...	1800	79	74	109	35	219	0.00%	4.7/sec	501.3	
/category/moderator/edit.do	3600	152	105	339	0	1775	0.00%	9.3/sec	77.2	
/j_spring_security_logout	1800	127	75	299	4	1219	0.00%	4.7/sec	50.8	
TOTAL	45000	71	38	170	0	1775	0.00%	112.9/sec	10381.5	

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 6775 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 70 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	6300	210	131	505	0	2453	0.00%	14.1/sec	154.0
/styles/style.css	2100	17	11	35	0	1103	0.00%	4.8/sec	7.5
/styles/bootstrap.min.css	2100	40	28	82	0	966	0.00%	4.8/sec	570.6
/styles/circle.css	2100	16	12	35	0	697	0.00%	4.8/sec	2.3
/styles/cookieconsent.min.css	2100	16	11	34	0	356	0.00%	4.8/sec	19.9
/scripts/jquery.min.js	2100	35	25	71	0	795	0.00%	4.8/sec	457.9
/scripts/polyfills.js	2100	16	12	36	0	441	0.00%	4.8/sec	36.2
/scripts/bootstrap.min.js	2100	22	17	46	0	257	0.00%	4.8/sec	175.5
/scripts/helpers.js	2100	16	11	33	0	776	0.00%	4.8/sec	11.0
/scripts/cookieconsent.js	2100	21	18	44	0	629	0.00%	4.8/sec	165.1
/images/logo.jpg	2100	230	168	484	21	1188	0.00%	4.8/sec	5423.9
/images/link_broken.png	4200	77	56	158	0	858	0.00%	9.4/sec	3061.2
/favicon.ico	2100	16	12	36	0	183	0.00%	4.8/sec	26.9
/security/login.do	2100	121	67	285	0	1861	0.00%	4.8/sec	30.0
/j_spring_security_check	2100	278	183	683	7	1914	0.00%	4.8/sec	55.9
/category/display.do	6300	188	127	411	0	2671	0.00%	14.3/sec	127.9
/wp-content/uploads/Default...	2100	100	88	156	35	394	0.00%	4.8/sec	518.3
/category/moderator/edit.do	4200	280	180	673	0	2909	0.00%	9.6/sec	80.0
/j_spring_security_logout	2100	231	145	522	5	2627	0.00%	4.8/sec	52.6
TOTAL	52500	123	56	323	0	2909	0.00%	117.2/sec	10779.5

Include group name in label? Save Table Header

En este caso vemos como a la hora de guardar el chollo y desplegarlo, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 7034.



Nuestro sistema se comporta así en este caso debido a las limitaciones del. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Editar patrocinios

En este caso de uso, un patrocinador puede editar patrocinios. Para ello, sigue la siguiente secuencia de pasos:

Mis patrocinios > (Escoger uno)

El límite de este caso de uso se sitúa en 80 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report																		
Name:	Aggregate Report																	
Comments:																		
Write results to file / Read from file																		
Filename	<input type="text"/>			<input type="button" value="Browse..."/>	<input type="checkbox"/> Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec									
/	7200	209	153	450	0	2844	0.00%	19.0/sec	206.3									
/styles/style.css	2400	11	9	24	0	215	0.00%	6.5/sec	10.2									
/styles/circle.css	2400	11	8	24	0	224	0.00%	6.5/sec	3.1									
/scripts/jquery.min.js	2400	25	18	50	0	232	0.00%	6.5/sec	619.2									
/styles/bootstrap.min.css	2400	28	20	57	0	495	0.00%	6.5/sec	771.9									
/styles/cookieconsent.min.css	2400	11	9	24	0	115	0.00%	6.5/sec	26.9									
/scripts/polyfills.js	2400	11	9	25	0	122	0.00%	6.5/sec	49.0									
/scripts/bootstrap.min.js	2400	16	13	34	0	213	0.00%	6.5/sec	237.4									
/scripts/helpers.js	2400	11	9	23	0	162	0.00%	6.5/sec	14.8									
/scripts/cookieconsent.js	2400	16	12	32	0	238	0.00%	6.5/sec	223.4									
/images/logo.jpg	2400	166	125	328	28	1372	0.00%	6.5/sec	7336.0									
/favicon.ico	2400	12	10	27	0	193	0.00%	6.5/sec	36.4									
/security/login.do	2400	120	85	256	0	1467	0.00%	6.5/sec	40.6									
/j_spring_security_check	2400	279	206	585	3	2640	0.00%	6.5/sec	73.8									
/images/link_broken.png	2400	59	44	120	0	370	0.00%	6.5/sec	2124.3									
/sponsorships/sponsor/list.do	4800	176	127	372	0	2308	0.00%	12.5/sec	128.9									
/sponsorships/sponsor/edit.do	4800	251	178	562	0	4340	0.00%	13.0/sec	118.7									
/j_spring_security_logout	2400	214	155	458	0	1915	0.00%	6.5/sec	71.1									
TOTAL	52800	112	41	316	0	4340	0.00%	139.5/sec	11787.8									

Como podemos observar, a la hora de iniciar sesión y cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8370 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 90 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

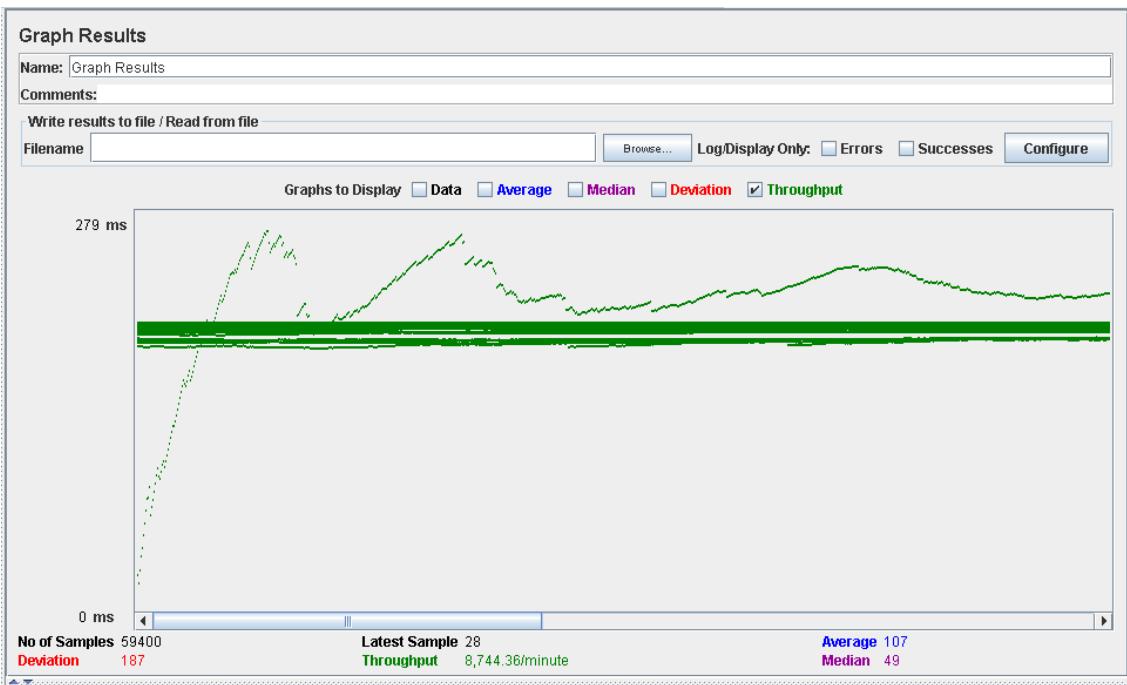
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	172	104	367	0	5285	0.00%	19.9/sec	215.5
/styles/style.css	2700	22	14	49	0	568	0.00%	6.8/sec	10.6
/styles/circle.css	2700	21	14	48	0	503	0.00%	6.8/sec	3.2
/scripts/jquery.min.js	2700	41	29	83	0	5104	0.00%	6.8/sec	645.7
/styles/bootstrap.min.css	2700	47	32	94	0	5091	0.00%	6.8/sec	805.1
/styles/cookieconsent.min.css	2700	20	14	47	0	501	0.00%	6.8/sec	28.1
/scripts/polyfills.js	2700	20	14	47	0	388	0.00%	6.8/sec	51.1
/scripts/bootstrap.min.js	2700	30	20	58	0	5218	0.00%	6.8/sec	247.6
/scripts/helpers.js	2700	22	14	46	0	5217	0.00%	6.8/sec	15.5
/scripts/cookieconsent.js	2700	26	19	56	0	347	0.00%	6.8/sec	233.0
/images/logo.jpg	2700	233	168	475	26	1653	0.00%	6.8/sec	7653.6
/favicon.ico	2700	23	14	50	0	624	0.00%	6.8/sec	38.0
/security/login.do	2700	111	62	257	0	5282	0.00%	6.8/sec	42.4
/j_spring_security_check	2700	231	151	498	4	2581	0.00%	6.8/sec	77.1
/images/link_broken.png	2700	85	60	182	0	769	0.00%	6.8/sec	2216.1
/sponsorship/sponsor/list.do	5400	148	87	324	0	5220	0.00%	13.5/sec	134.5
/sponsorship/sponsor/edit.do	5400	213	131	463	0	5370	0.00%	13.6/sec	123.8
/j_spring_security_logout	2700	184	122	383	1	4028	0.00%	6.8/sec	74.1
TOTAL	59400	107	49	276	0	5370	0.00%	145.7/sec	12314.8

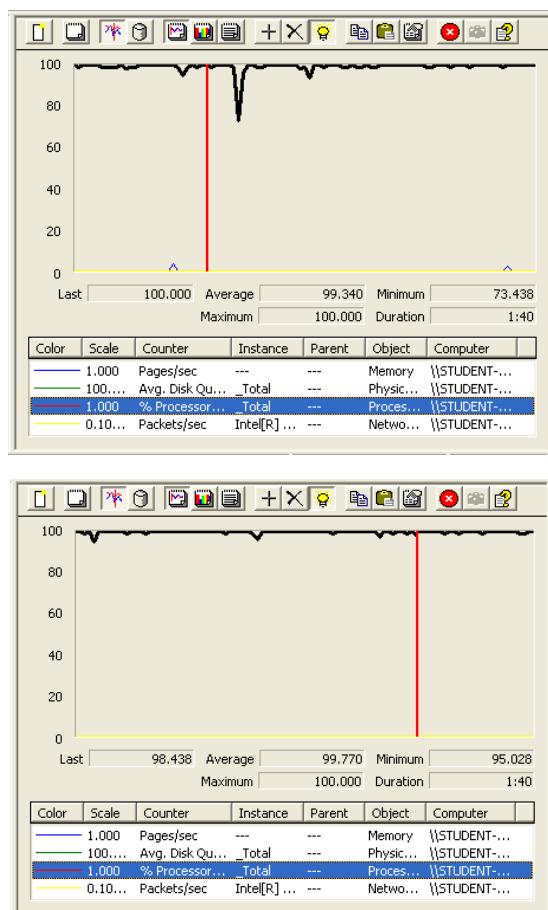
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 8744.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador.
Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Editar términos y condiciones

En este caso de uso, un administrador puede editar los términos y condiciones. Para ello, sigue la siguiente secuencia de pasos:

Términos y condiciones > (Editar en el idioma deseado)

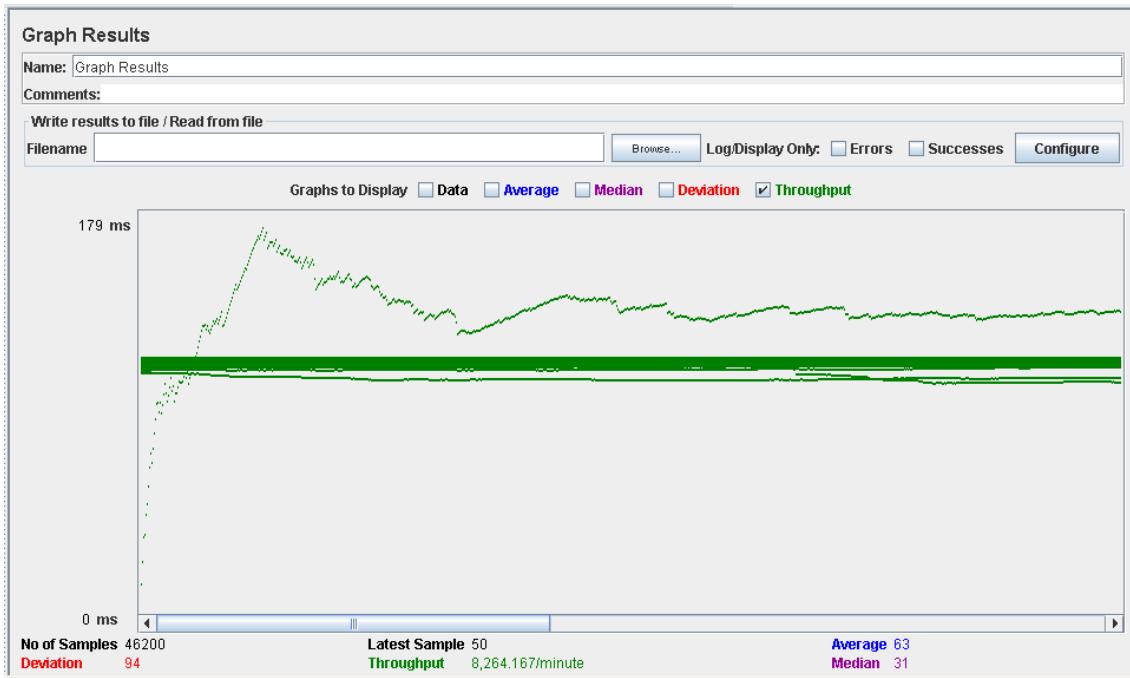
El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Name: Aggregate Report		Comments:								
Write results to file / Read from file		Filename <input type="text"/>		Browse...		Log/Display Only:		<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec	
/welcome/index.do	2100	110	72	249	0	1705	0.00%	6.4/sec	69.4	
/styles/style.css	2100	11	7	27	0	202	0.00%	6.4/sec	10.1	
/styles/cookieconsent.min.css	2100	11	7	27	0	190	0.00%	6.4/sec	26.6	
/scripts/jquery.min.js	2100	24	18	51	0	266	0.00%	6.4/sec	612.8	
/styles/circle.css	2100	10	7	27	0	223	0.00%	6.4/sec	3.1	
/styles/bootstrap.min.css	2100	27	20	59	0	244	0.00%	6.4/sec	763.9	
/scripts/bootstrap.min.js	2100	16	12	35	0	269	0.00%	6.4/sec	234.9	
/scripts/polyfills.js	2100	10	7	27	0	394	0.00%	6.4/sec	48.5	
/scripts/helpers.js	2100	10	7	26	0	174	0.00%	6.4/sec	14.7	
/scripts/cookieconsent.js	2100	14	11	33	0	622	0.00%	6.4/sec	221.1	
/images/logo.jpg	2100	163	121	315	14	1312	0.00%	6.4/sec	7261.4	
/favicon.ico	2100	12	8	29	0	569	0.00%	6.4/sec	36.0	
/security/login.do	2100	60	39	136	0	708	0.00%	6.4/sec	40.2	
/j_spring_security_check	2100	132	90	279	1	1119	0.00%	6.4/sec	73.8	
/	4200	102	65	225	0	1091	0.00%	12.8/sec	138.2	
/images/link_broken.png	2100	56	41	112	0	503	0.00%	6.4/sec	2102.1	
/termCondition/administrator/dis...	4200	84	53	184	0	1096	0.00%	12.8/sec	129.6	
/termCondition/administrator/edit....	4200	119	74	265	0	1589	0.00%	12.8/sec	126.6	
/j_spring_security_logout	2100	111	74	239	0	790	0.00%	6.4/sec	70.3	
TOTAL	46200	63	31	162	0	1705	0.00%	137.7/sec	11650.9	

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8264 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

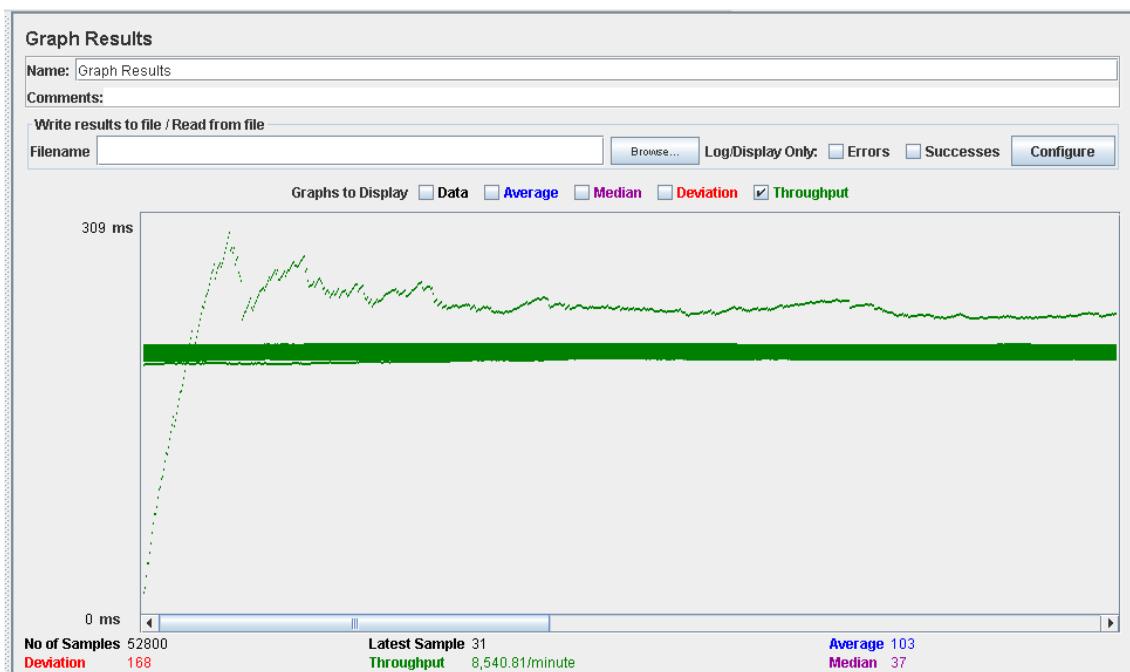
Name: Aggregate Report
Comments:

Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

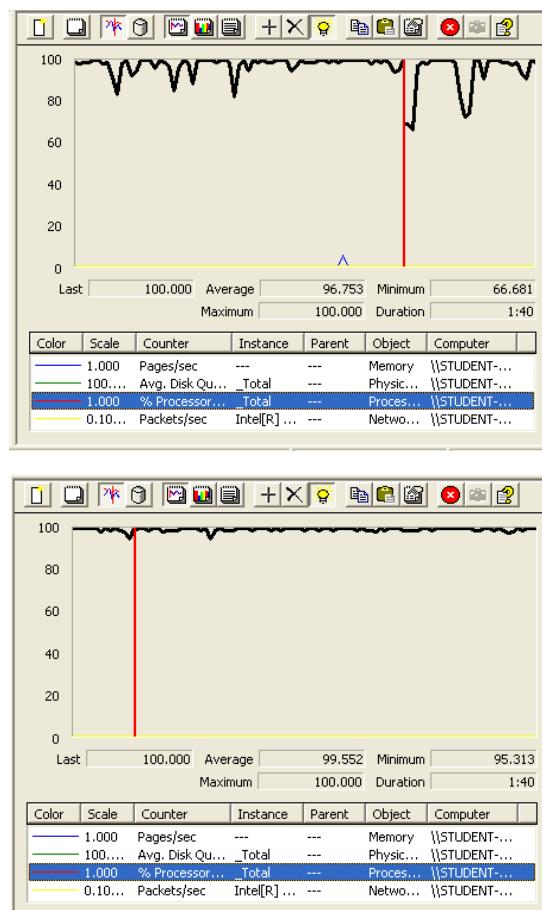
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
/welcome/index.do	2400	204	142	453	0	2528	0.00%	6.6/sec	71.5
/styles/style.css	2400	10	8	23	0	339	0.00%	6.6/sec	10.4
/styles/cookieconsent.min.css	2400	10	8	23	0	220	0.00%	6.6/sec	27.4
/scripts/jquery.min.js	2400	23	18	47	0	535	0.00%	6.6/sec	630.9
/styles/circle.css	2400	10	8	23	0	224	0.00%	6.6/sec	3.2
/styles/bootstrap.min.css	2400	26	20	53	0	238	0.00%	6.6/sec	786.4
/scripts/bootstrap.min.js	2400	15	11	32	0	134	0.00%	6.6/sec	241.8
/scripts/polyfills.js	2400	10	8	23	0	136	0.00%	6.6/sec	49.9
/scripts/helpers.js	2400	10	8	23	0	238	0.00%	6.6/sec	15.1
/scripts/cookieconsent.js	2400	14	11	30	0	200	0.00%	6.6/sec	227.6
/images/logo.jpg	2400	150	117	292	15	872	0.00%	6.6/sec	7474.8
/favicon.ico	2400	12	9	26	0	267	0.00%	6.6/sec	37.0
/security/login.do	2400	120	74	259	0	3325	0.00%	6.6/sec	41.4
/j_spring_security_check	2400	275	187	592	4	3657	0.00%	6.6/sec	76.0
/	4800	185	129	410	0	2876	0.00%	13.1/sec	142.8
/images/link_broken.png	2400	53	38	108	0	461	0.00%	6.6/sec	2167.1
/termCondition/administrator/dis...	4800	156	106	343	0	2400	0.00%	13.2/sec	133.8
/termCondition/administrator/edit....	4800	221	149	489	0	3459	0.00%	13.3/sec	130.5
/j_spring_security_logout	2400	200	139	425	0	3549	0.00%	6.7/sec	72.5
TOTAL	52800	103	37	290	0	3657	0.00%	142.3/sec	12041.0

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 8540.



Nuestro sistema se comporta así en este caso debido a las limitaciones del. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Listar los chollos de una categoría

En este caso de uso, cualquier usuario puede listar los chollos de una categoría. Para ello, sigue la siguiente secuencia de pasos:

Categorías > (Escoger una) > Chollos

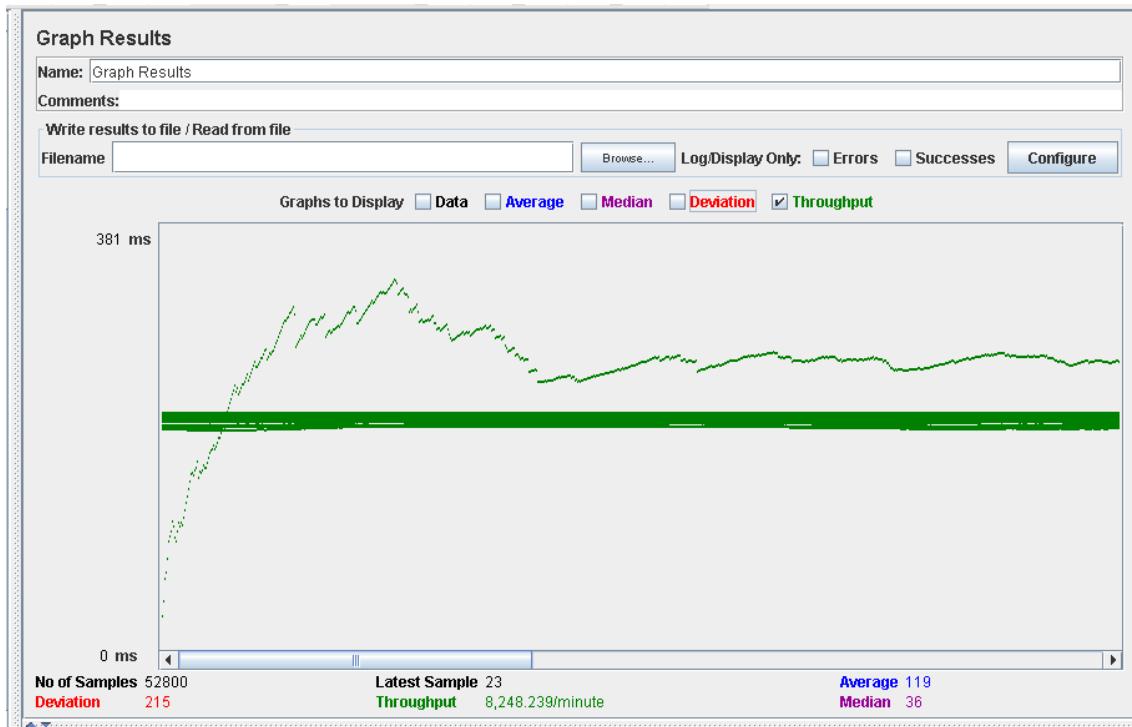
El límite de este caso de uso se sitúa en 80 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	219	135	485	0	3358	0.00%	18.7/sec	204.9
/styles/circle.c...	2400	11	9	25	0	324	0.00%	6.4/sec	3.1
/scripts/bootstrap...	2400	17	14	35	0	517	0.00%	6.4/sec	234.3
/styles/style.css	2400	11	9	23	0	489	0.00%	6.4/sec	10.0
/styles/cookie...	2400	12	9	26	0	177	0.00%	6.4/sec	26.6
/scripts/jquery...	2400	26	20	53	0	229	0.00%	6.4/sec	611.4
/styles/bootstrap...	2400	29	24	60	0	248	0.00%	6.4/sec	762.2
/scripts/polyfill...	2400	11	9	25	0	110	0.00%	6.4/sec	48.4
/scripts/help...	2400	12	9	25	0	375	0.00%	6.4/sec	14.7
/scripts/cookie...	2400	16	13	32	0	300	0.00%	6.4/sec	220.6
/images/logo....	2400	171	134	333	25	705	0.00%	6.4/sec	7245.9
/favicon.ico	2400	13	9	28	0	256	0.00%	6.4/sec	35.9
/security/login...	2400	135	77	309	0	2720	0.00%	6.4/sec	40.1
/j_spring_sec...	2400	299	190	667	9	3573	0.00%	6.4/sec	74.6
/images/link...	4800	58	43	121	0	522	0.00%	12.7/sec	4138.5
/category/disp...	4800	331	365	628	0	4933	0.00%	12.8/sec	124.4
/bargain/bycat...	2400	184	107	394	1	4789	0.00%	6.4/sec	60.8
/j_spring_sec...	2400	236	153	510	3	4876	0.00%	6.4/sec	70.1
TOTAL	52800	119	36	364	0	4933	0.00%	137.5/sec	13598.6

Como podemos observar, a la hora de mostrar los chollos de una categoría, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8248 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 90 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes Throughput

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	338	215	740	0	5783	0.00%	18.5/sec	202.6
/styles/circle.c...	2700	15	10	32	0	366	0.00%	6.3/sec	3.0
/scripts/bootstrap...	2700	23	16	45	0	1159	0.00%	6.3/sec	230.8
/styles/style.css	2700	15	10	32	0	423	0.00%	6.3/sec	9.9
/styles/cookie...	2700	16	11	33	0	1110	0.00%	6.3/sec	26.2
/scripts/jquery...	2700	34	25	72	0	429	0.00%	6.3/sec	602.4
/styles/bootstrap...	2700	38	28	76	0	1055	0.00%	6.3/sec	750.9
/scripts/polyfill...	2700	16	10	32	0	1324	0.00%	6.3/sec	47.7
/scripts/helperr...	2700	16	11	33	0	471	0.00%	6.3/sec	14.4
/scripts/cookies...	2700	21	15	42	0	372	0.00%	6.3/sec	217.3
/images/logo....	2700	209	163	409	25	1893	0.00%	6.3/sec	7136.8
/favicon.ico	2700	16	11	34	0	244	0.00%	6.3/sec	35.4
/security/login...	2700	258	131	636	0	4308	0.00%	6.3/sec	39.5
/j_spring_sec...	2700	550	366	1271	7	5067	0.00%	6.3/sec	73.5
/images/link....	5400	76	56	150	0	862	0.00%	12.5/sec	4078.9
/category/displ...	5400	450	407	881	0	5288	0.00%	12.6/sec	122.4
/bargain/bycat...	2700	291	175	685	0	3771	0.00%	6.3/sec	59.9
/j_spring_sec...	2700	341	224	753	2	3760	0.00%	6.3/sec	69.0
TOTAL	59400	178	49	505	0	5783	0.00%	136.0/sec	13450.6

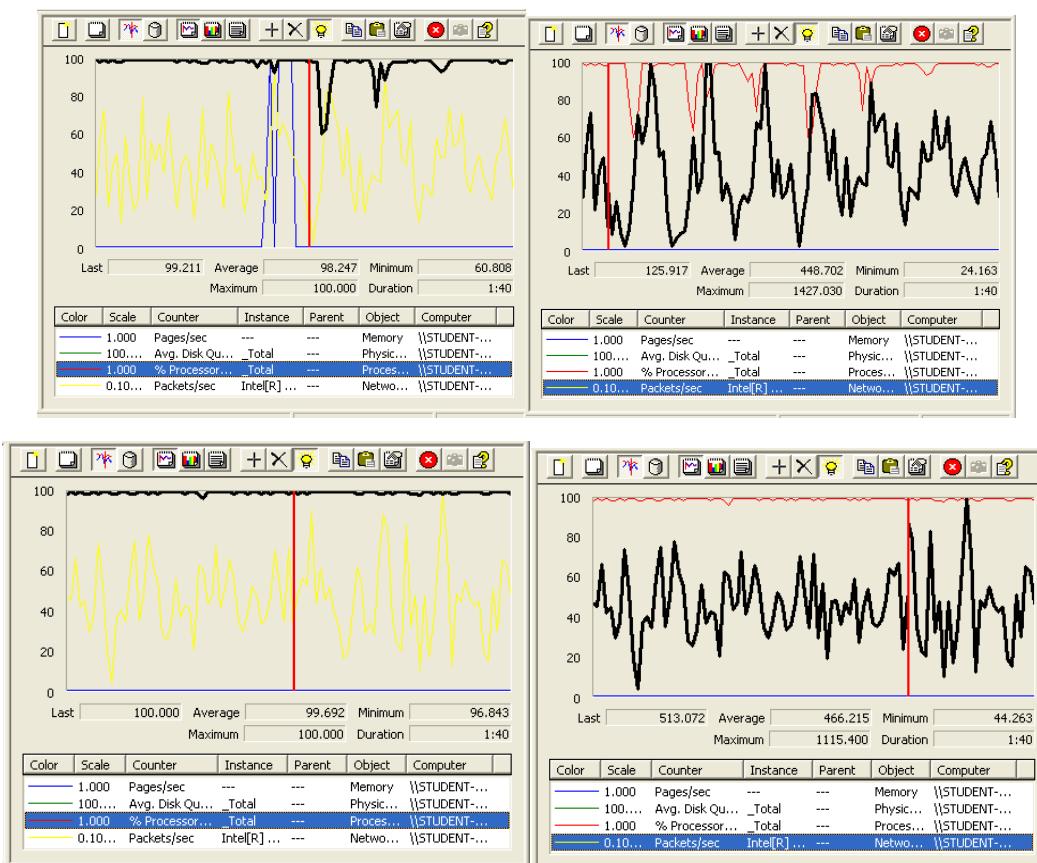
Include group name in label? Save Table Header

En este caso vemos como a la hora de listar los chollos de una categoría, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 8158.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y de la tarjeta de red. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



Listar y desplegar chollos

En este caso de uso, cualquier usuario puede listar y desplegar los chollos. Para ello, sigue la siguiente secuencia de pasos:

Chollos

El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1500	122	76	262	14	1413	0.00%	7.7/sec	83.4
/styles/circle.c...	1500	10	6	24	1	220	0.00%	7.7/sec	3.7
/styles/bootstrap...	1500	23	16	44	6	439	0.00%	7.7/sec	916.5
/scripts/jquery...	1500	21	15	42	5	221	0.00%	7.7/sec	735.3
/styles/style.css	1500	11	6	23	1	184	0.00%	7.7/sec	12.1
/styles/cookie...	1500	11	6	23	1	225	0.00%	7.7/sec	31.9
/scripts/bootstrap...	1500	14	9	28	2	236	0.00%	7.7/sec	281.9
/scripts/polyfill...	1500	11	7	23	1	152	0.00%	7.7/sec	58.2
/scripts/cookie...	1500	14	9	28	2	265	0.00%	7.7/sec	265.3
/scripts/help...	1500	10	6	23	1	125	0.00%	7.7/sec	17.6
/images/logo....	1500	121	97	204	38	751	0.00%	7.7/sec	8714.7
/images/link....	1500	45	35	81	11	401	0.00%	7.7/sec	2520.7
/favicon.ico	1500	11	7	25	2	191	0.00%	7.7/sec	43.3
/bargain/list.do	1500	75	48	155	11	2381	0.00%	7.7/sec	67.4
/bargain/displ...	1500	867	804	1068	614	2644	0.00%	7.7/sec	81.8
/images/yes.p...	1500	12	7	28	2	142	0.00%	7.8/sec	11.3
/welcome/ind...	1500	120	74	258	15	1384	0.00%	7.8/sec	82.6
TOTAL	25500	88	17	162	1	2644	0.00%	128.1/sec	13580.8

Como podemos observar, a la hora de desplegar el chollo, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 7687 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file

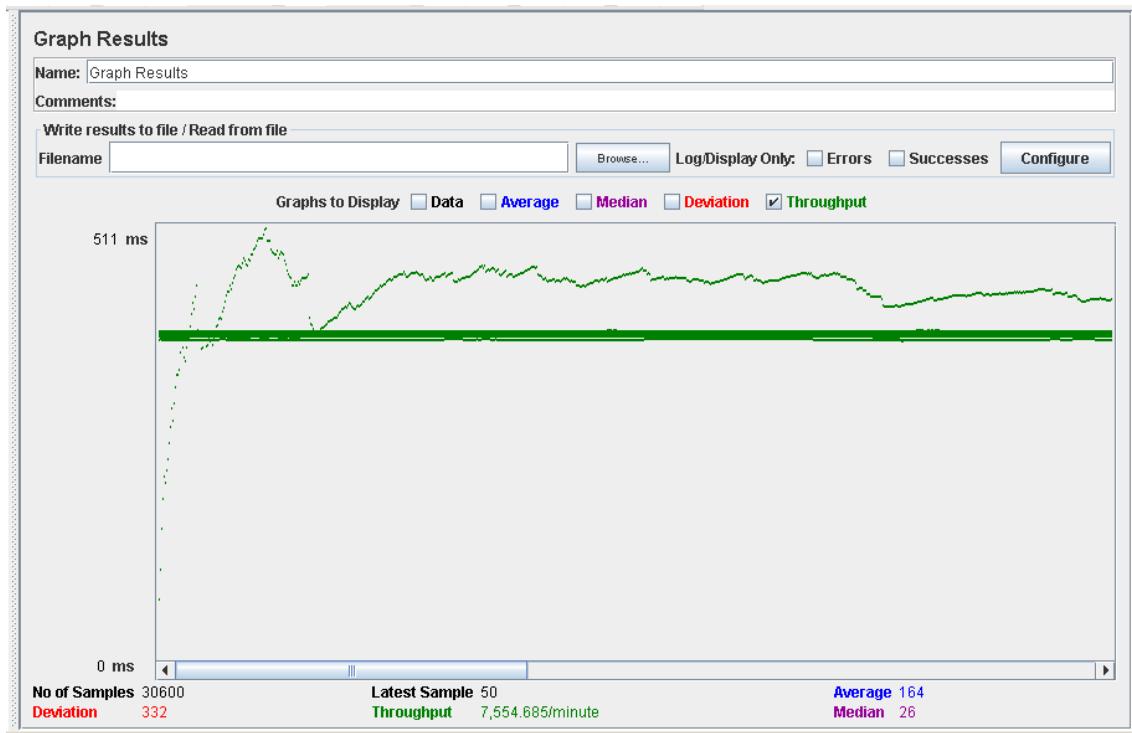
Filename Browse... Log/Display Only Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1800	367	284	736	18	2417	0.00%	7.6/sec	81.8
/styles/circle.c...	1800	15	11	29	1	238	0.00%	7.6/sec	3.6
/styles/bootstrap...	1800	39	29	74	6	1180	0.00%	7.6/sec	898.8
/scripts/jquery...	1800	34	26	64	5	582	0.00%	7.6/sec	721.1
/styles/style.css	1800	14	11	31	1	217	0.00%	7.6/sec	11.8
/styles/cookie...	1800	15	11	29	1	242	0.00%	7.6/sec	31.3
/scripts/bootstrap...	1800	21	16	42	3	178	0.00%	7.6/sec	276.4
/scripts/polyfill...	1800	15	11	30	2	247	0.00%	7.6/sec	57.0
/scripts/cookie...	1800	20	16	39	3	242	0.00%	7.6/sec	260.1
/scripts/helperr...	1800	15	11	31	1	221	0.00%	7.6/sec	17.3
/images/logo....	1800	227	206	421	38	898	0.00%	7.6/sec	8543.5
/images/link....	1800	79	63	152	11	785	0.00%	7.6/sec	2471.1
/favicon.ico	1800	17	12	36	2	1155	0.00%	7.6/sec	42.4
/bargain/list.do	1800	300	223	634	12	2831	0.00%	7.6/sec	66.0
/bargain/displ...	1800	1204	1123	1650	622	4140	0.00%	7.6/sec	80.1
/images/yes.p...	1800	17	12	35	2	216	0.00%	7.6/sec	11.0
/welcome/ind...	1800	389	302	776	17	3463	0.00%	7.6/sec	80.7
TOTAL	30600	164	26	516	1	4140	0.00%	125.9/sec	13346.5

Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de desplegar el chollo, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 7554.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Listar y desplegar mis chollos

En este caso de uso, una compañía puede listar y desplegar sus chollos. Para ello, sigue la siguiente secuencia de pasos:

Mis chollos

El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/welcome/index...	1500	108	67	243	15	921	0.00%	5.1/sec	54.6
/styles/style.css	1500	9	8	20	1	283	0.00%	5.1/sec	7.9
/styles/cookie...	1500	9	6	20	1	139	0.00%	5.1/sec	20.9
/styles/circle.c...	1500	9	6	18	1	142	0.00%	5.1/sec	2.4
/scripts/jquery...	1500	22	16	44	4	242	0.00%	5.1/sec	481.4
/scripts/bootstrap...	1500	25	19	48	5	384	0.00%	5.1/sec	600.1
/scripts/bootstrap...	1500	14	10	27	3	498	0.00%	5.1/sec	184.5
/scripts/polyfill...	1500	9	7	20	1	172	0.00%	5.1/sec	38.1
/scripts/helperr...	1500	10	6	19	1	416	0.00%	5.1/sec	11.5
/scripts/cookies...	1500	14	10	27	2	243	0.00%	5.1/sec	173.7
/images/logo....	1500	151	120	276	37	990	0.00%	5.1/sec	5704.3
/images/link....	3000	56	41	110	10	777	0.00%	9.8/sec	3209.1
/favicon.ico	1500	10	6	22	2	360	0.00%	5.1/sec	28.3
/security/login....	1500	51	29	113	8	630	0.00%	5.1/sec	31.6
/j_spring_sec...	1500	115	75	245	23	950	0.00%	5.1/sec	57.9
/	3000	102	63	228	14	1498	0.00%	10.0/sec	109.1
/bargain/com...	1500	77	49	162	14	615	0.00%	5.1/sec	50.7
/bargain/displ...	1500	848	801	1068	614	1818	0.00%	5.1/sec	59.6
/images/yes.p...	1500	9	6	19	1	123	0.00%	5.1/sec	7.3
/j_spring_sec...	1500	115	73	252	5	1237	0.00%	5.1/sec	55.2
TOTAL	33000	87	25	188	1	1818	0.00%	108.1/sec	10664.5

Como podemos observar, a la hora de desplegar el chollo, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 6486 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file
Filename Browse... Log/Display Only Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/welcome/index.do	1800	173	101	404	15	2734	0.00%	5.3/sec	56.9
/styles/style.css	1800	15	10	34	1	271	0.00%	5.3/sec	8.2
/styles/cookieconsent.min.css	1800	15	10	34	1	233	0.00%	5.3/sec	21.8
/styles/circle.css	1800	15	9	32	1	1078	0.00%	5.3/sec	2.5
/scripts/jquery.min.js	1800	33	23	72	4	370	0.00%	5.3/sec	501.3
/styles/bootstrap.min.css	1800	36	25	75	4	410	0.00%	5.3/sec	624.9
/scripts/bootstrap.min.js	1800	21	14	44	3	294	0.00%	5.3/sec	192.2
/scripts/polyfills.js	1800	16	11	32	2	959	0.00%	5.3/sec	39.7
/scripts/helpers.js	1800	16	10	33	2	177	0.00%	5.3/sec	12.0
/scripts/cookieconsent.js	1800	21	14	45	2	217	0.00%	5.3/sec	180.8
/images/logo.jpg	1800	209	151	415	35	1638	0.00%	5.3/sec	5939.0
/images/link_broken.png	3600	75	53	156	10	959	0.00%	10.3/sec	3352.4
/favicon.ico	1800	17	11	36	1	751	0.00%	5.3/sec	29.5
/security/login.do	1800	157	54	238	8	5789	0.00%	5.3/sec	32.8
/j_spring_security_check	1800	217	133	481	22	6925	0.00%	5.3/sec	60.2
/	3600	189	104	416	17	6341	0.00%	10.4/sec	113.7
/bargain/companylist.do	1800	158	82	317	15	6141	0.00%	5.3/sec	52.7
/bargain/display.do	1800	1026	891	1357	614	8165	0.00%	5.3/sec	62.0
/images/yes.png	1800	16	10	34	1	308	0.00%	5.3/sec	7.6
/j_spring_security_logout	1800	196	128	430	5	1756	0.00%	5.3/sec	57.5
TOTAL	39600	131	36	316	1	8165	0.00%	113.0/sec	11144.0

Include group name in label? Save Table Header

En este caso vemos como a la hora de desplegar el chollo, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 6777.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Listar y desplegar mis patrocinios

En este caso de uso, un patrocinador puede listar y desplegar sus patrocinios. Para ello, sigue la siguiente secuencia de pasos:

Mis patrocinios > (Escoger uno)

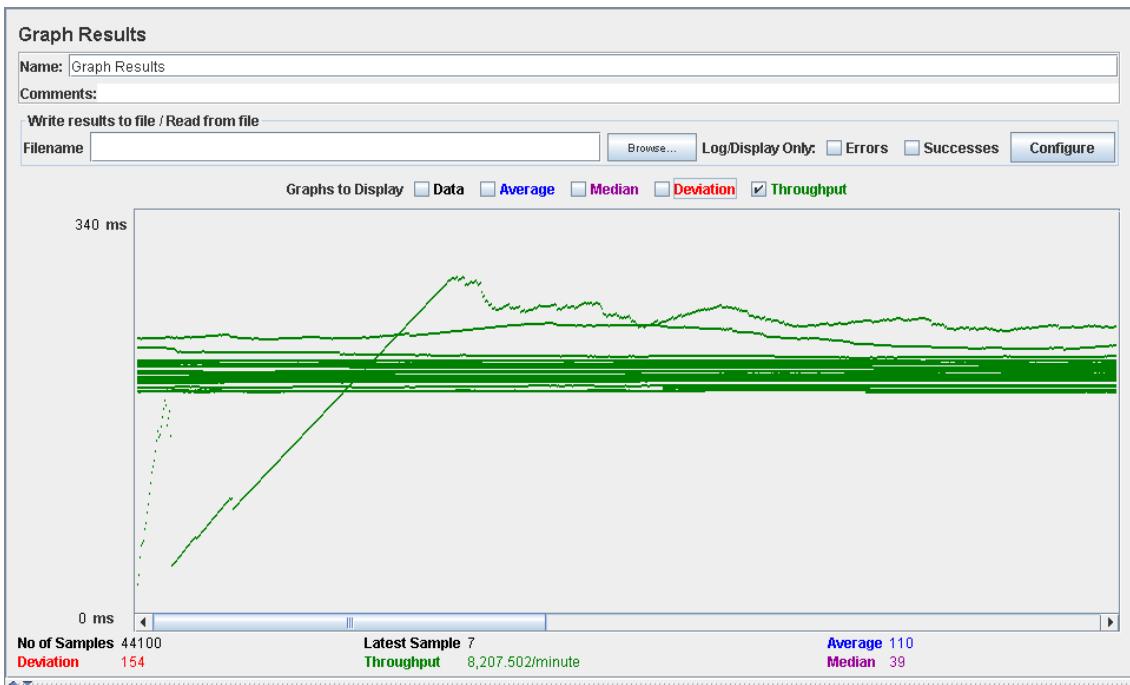
El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report																		
Name:	Aggregate Report																	
Comments:																		
Write results to file / Read from file																		
Filename				Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="checkbox"/> Configure										
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec									
/welcome/index.do	2100	220	187	420	0	1441	0.00%	6.6/sec	71.3									
/scripts/jquery.min.js	2100	26	21	49	0	270	0.00%	6.6/sec	629.4									
/styles/style.css	2100	10	8	22	0	240	0.00%	6.6/sec	10.3									
/styles/bootstrap.min.css	2100	31	25	60	0	984	0.00%	6.6/sec	784.4									
/styles/circle.css	2100	10	8	24	0	162	0.00%	6.6/sec	3.2									
/styles/cookieconsent.min.css	2100	10	8	22	0	497	0.00%	6.6/sec	27.3									
/scripts/bootstrap.min.js	2100	16	13	32	0	206	0.00%	6.6/sec	241.3									
/scripts/polyfills.js	2100	10	9	23	0	117	0.00%	6.6/sec	49.8									
/scripts/helpers.js	2100	10	8	22	0	402	0.00%	6.6/sec	15.1									
/scripts/cookieconsent.js	2100	15	13	30	0	333	0.00%	6.6/sec	226.9									
/images/logo.jpg	2100	189	168	339	17	1259	0.00%	6.6/sec	7453.8									
/images/link_broken.png	4200	64	53	126	0	474	0.00%	13.1/sec	4271.6									
/favicon.ico	2100	11	9	25	0	172	0.00%	6.6/sec	37.1									
/security/login.do	2100	132	100	273	0	1297	0.00%	6.7/sec	41.5									
/j_spring_security_check	2100	290	235	562	4	2271	0.00%	6.7/sec	75.5									
/	4200	220	184	434	0	1489	0.00%	13.2/sec	143.4									
/sponsorship/sponsor/list.do	2100	194	163	368	0	1505	0.00%	6.6/sec	66.2									
/sponsorship/sponsor/display.do	2100	341	301	534	112	1825	0.00%	6.7/sec	47.7									
/j_spring_security_logout	2100	234	195	440	0	1857	0.00%	6.7/sec	72.9									
TOTAL	44100	110	39	318	0	2271	0.00%	136.8/sec	14095.8									

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8207 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

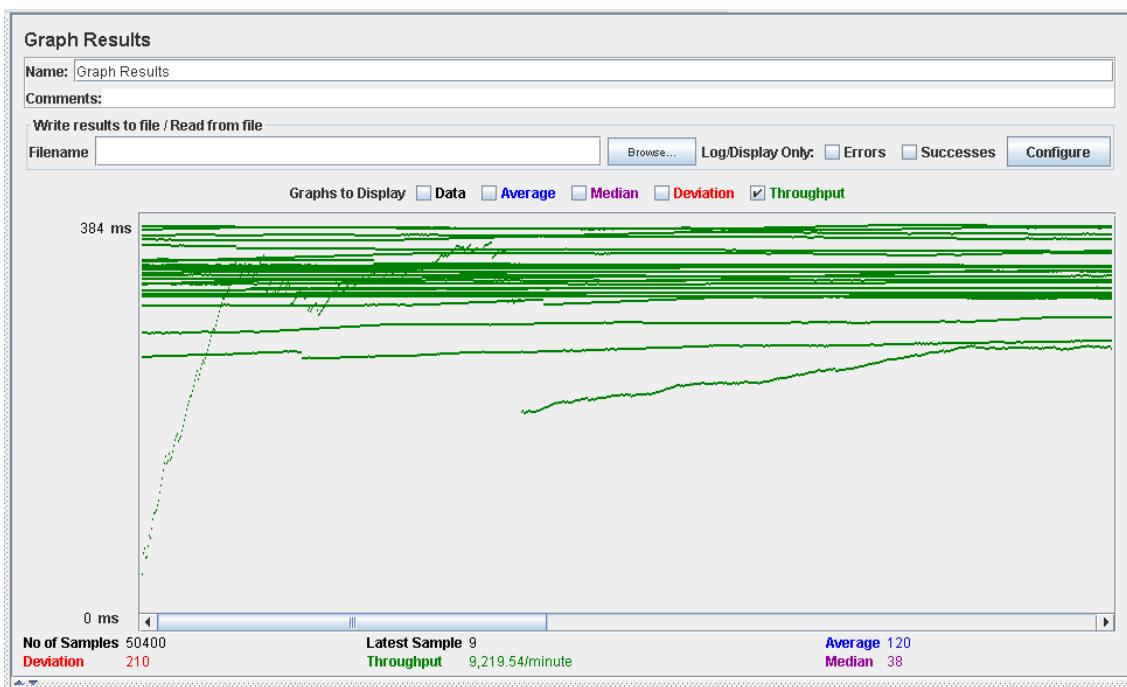
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Throughput KB/sec Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/welcome/index.do	2400	251	160	541	1	2843	0.00%	7.6/sec	81.6
/scripts/jquery.min.js	2400	27	20	55	0	441	0.00%	7.6/sec	720.1
/styles/style.css	2400	12	8	25	0	970	0.00%	7.6/sec	11.8
/styles/bootstrap.min.css	2400	29	21	61	0	577	0.00%	7.6/sec	898.0
/styles/circle.css	2400	12	8	26	0	311	0.00%	7.6/sec	3.6
/styles/cookieconsent.min.css	2400	11	8	25	0	149	0.00%	7.6/sec	31.3
/scripts/bootstrap.min.js	2400	17	12	35	0	709	0.00%	7.6/sec	276.3
/scripts/polyfills.js	2400	12	8	25	0	410	0.00%	7.6/sec	57.0
/scripts/helpers.js	2400	12	8	26	0	348	0.00%	7.6/sec	17.3
/scripts/cookieconsent.js	2400	17	12	34	0	360	0.00%	7.6/sec	260.0
/images/logo.jpg	2400	172	127	335	13	1039	0.00%	7.6/sec	8538.3
/images/link_broken.png	4800	61	44	126	0	456	0.00%	14.9/sec	4849.4
/favicon.ico	2400	12	9	27	0	402	0.00%	7.6/sec	42.4
/security/login.do	2400	164	85	379	0	2565	0.00%	7.6/sec	47.3
/j_spring_security_check	2400	346	214	791	4	3046	0.00%	7.6/sec	88.0
/	4800	237	158	513	0	3984	0.00%	14.9/sec	161.9
/sponsorship/sponsor/list.do	2400	213	135	472	0	2935	0.00%	7.6/sec	75.9
/sponsorship/sponsor/display.do	2400	369	287	642	110	2873	0.00%	7.6/sec	54.2
/j_spring_security_logout	2400	253	173	525	0	3917	0.00%	7.6/sec	82.6
TOTAL	50400	120	38	333	0	3984	0.00%	153.7/sec	15834.0

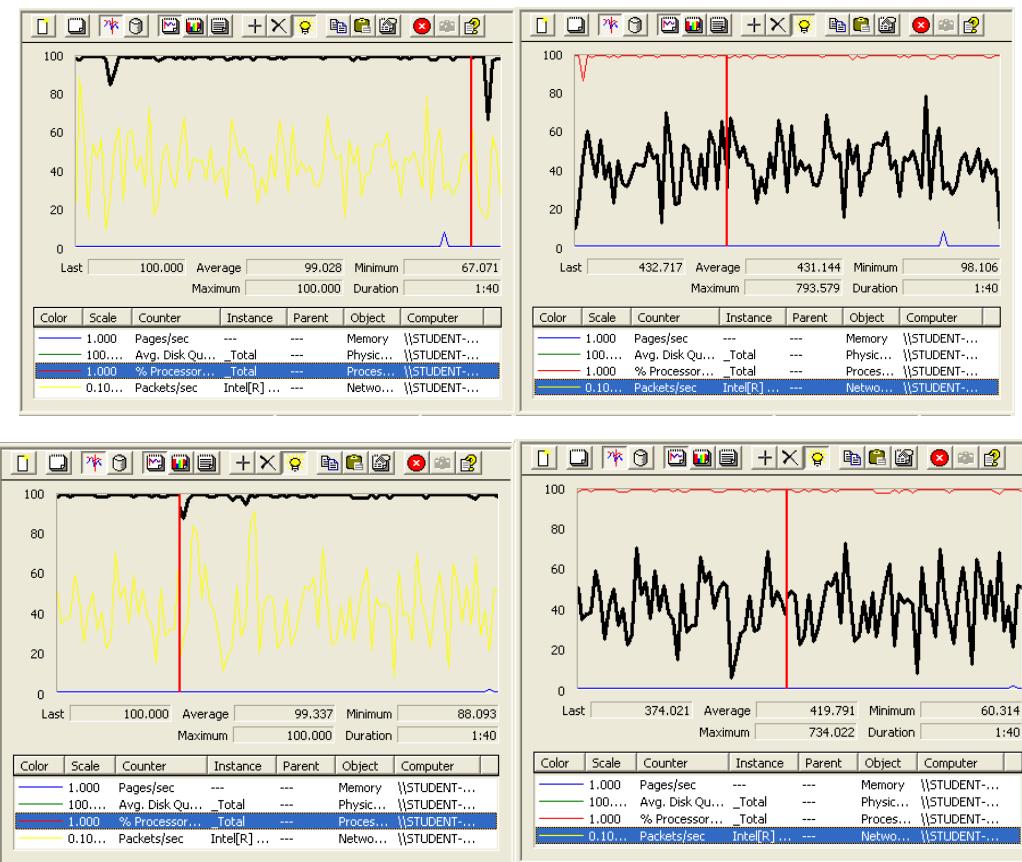
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 9219.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y de la tarjeta de red. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



Listar y desplegar mis notificaciones

En este caso de uso, un usuario puede listar y desplegar sus notificaciones. Para ello, sigue la siguiente secuencia de pasos:

Mis notificaciones > (Escoger una)

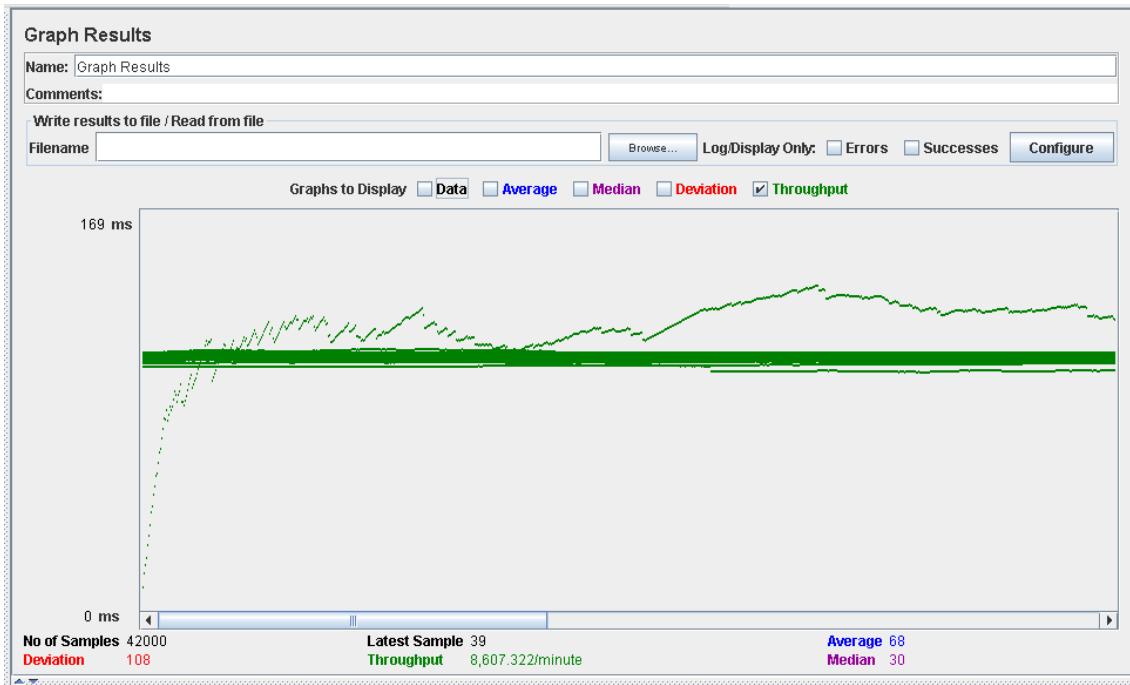
El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Filename	<input type="button" value="Browse..."/>	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>				
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	6300	135	83	297	0	1396	0.00%	21.5/sec	236.2
/styles/cookiec...	2100	12	8	28	0	158	0.00%	7.4/sec	30.5
/styles/style.css	2100	12	8	27	0	168	0.00%	7.4/sec	11.5
/styles/bootstrap...	2100	27	20	56	0	375	0.00%	7.4/sec	875.7
/styles/circle.css	2100	11	8	27	0	136	0.00%	7.4/sec	3.5
/scripts/jquery....	2100	25	17	52	0	523	0.00%	7.4/sec	702.5
/scripts/bootstrap...	2100	17	12	38	0	227	0.00%	7.4/sec	269.3
/scripts/polyfill...	2100	12	8	27	0	196	0.00%	7.4/sec	55.6
/scripts/helper....	2100	12	8	26	0	243	0.00%	7.4/sec	16.8
/scripts/cookie...	2100	16	11	34	0	182	0.00%	7.4/sec	253.4
/images/logo.jpg	2100	153	107	301	25	959	0.00%	7.4/sec	8323.9
/favicon.ico	2100	14	9	30	0	308	0.00%	7.4/sec	41.3
/security/login.do	2100	73	45	162	0	902	0.00%	7.4/sec	46.0
/j_spring_secu...	2100	185	130	393	0	1649	0.00%	7.4/sec	86.7
/images/link_b...	2100	57	41	113	0	450	0.00%	7.4/sec	2410.6
/notification/act...	2100	101	62	225	0	962	0.00%	7.4/sec	66.8
/notification/act...	2100	94	61	201	0	885	0.00%	7.4/sec	49.9
/j_spring_secu...	2100	136	88	293	5	1229	0.00%	7.4/sec	80.5
TOTAL	42000	68	30	176	0	1649	0.00%	143.5/sec	13179.4

Como podemos observar, a la hora de cargar la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 8607 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

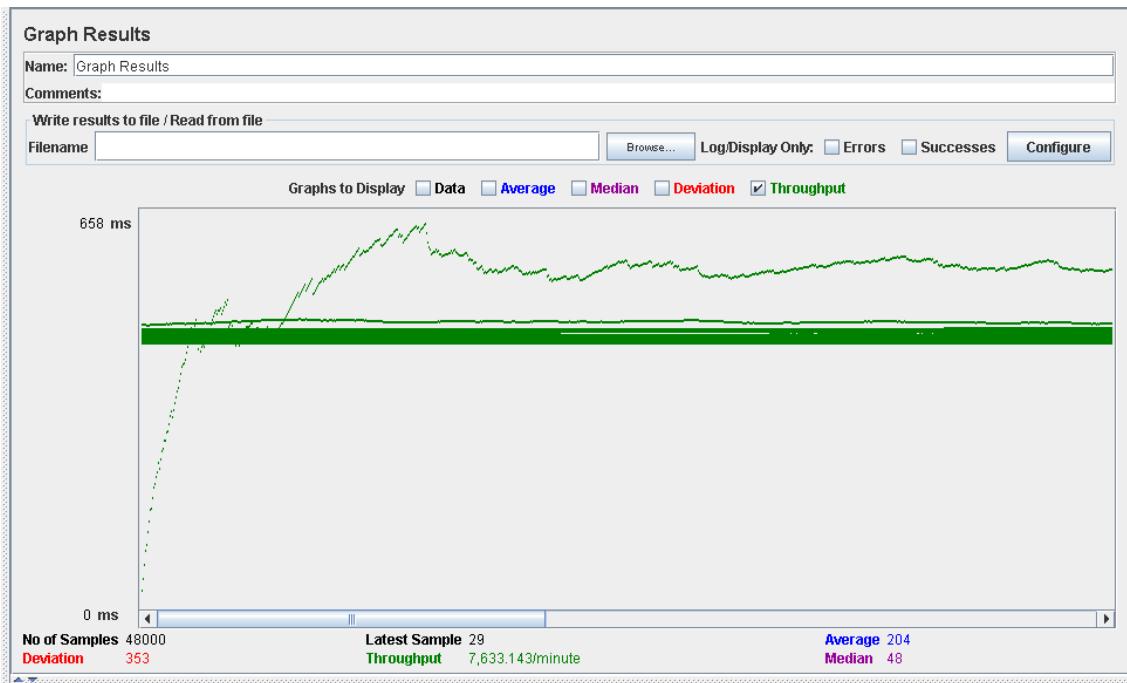
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	430	314	907	0	4902	0.00%	19.1/sec	209.4
/styles/cookiec...	2400	16	12	35	0	355	0.00%	6.5/sec	26.9
/styles/style.css	2400	16	12	35	0	322	0.00%	6.5/sec	10.2
/styles/bootstrap...	2400	43	28	96	0	617	0.00%	6.5/sec	771.6
/styles/circle.css	2400	16	12	35	0	220	0.00%	6.5/sec	3.1
/scripts/jquery....	2400	38	26	83	0	383	0.00%	6.5/sec	619.1
/scripts/bootstrap...	2400	25	17	52	0	309	0.00%	6.5/sec	237.3
/scripts/polyfill...	2400	15	12	34	0	179	0.00%	6.5/sec	49.0
/scripts/helper...	2400	15	11	33	0	234	0.00%	6.5/sec	14.8
/scripts/cookie...	2400	22	16	48	0	384	0.00%	6.5/sec	223.3
/images/logo.jpg	257	167	575	21	2114		0.00%	6.5/sec	7333.7
/favicon.ico	2400	17	13	37	0	161	0.00%	6.5/sec	38.4
/security/login.do	2400	309	213	650	0	4304	0.00%	6.5/sec	40.6
/j_spring_secur...	713	587	1419	6	5441		0.00%	6.5/sec	76.3
/images/link_b...	2400	100	64	224	0	734	0.00%	6.5/sec	2123.9
/notification/act...	2400	366	261	753	0	5569	0.00%	6.5/sec	58.9
/notification/act...	370	263	745	0	5388		0.00%	6.5/sec	44.0
/j_spring_secur...	2400	441	327	905	0	4095	0.00%	6.5/sec	71.0
TOTAL	48000	204	48	608	0	5569	0.00%	127.2/sec	11687.7

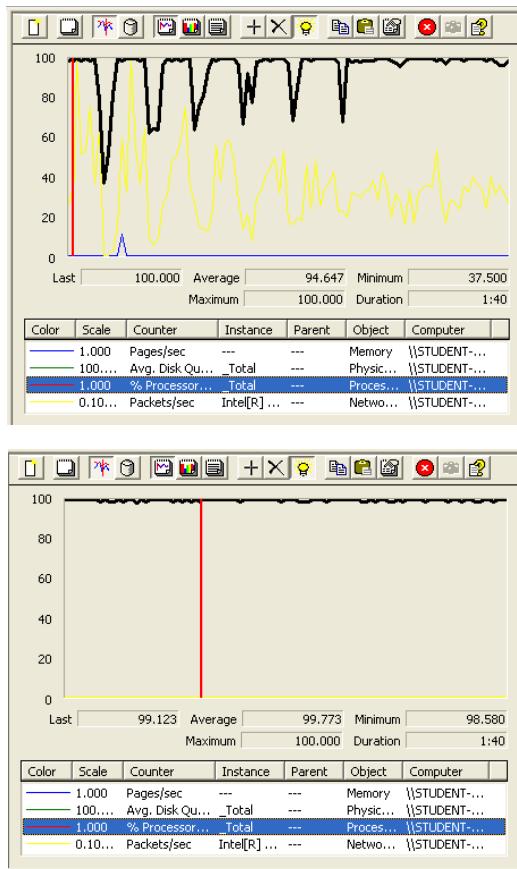
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de iniciar sesión y cargar la página principal, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 7633.



Nuestro sistema se comporta así en este caso debido a las limitaciones del. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Listar patrocinios de un chollo

En este caso de uso, cualquier usuario puede listar patrocinios de un chollo. Para ello, sigue la siguiente secuencia de pasos:

Chollos > (Escoger uno) > Patrocinios

El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Filename	Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	Configure				
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1500	125	89	257	0	1113	0.00%	6.1/sec	65.6
/styles/cookieconsent.min.c...	1500	7	6	24	0	199	0.00%	6.1/sec	25.2
/styles/style.css	1500	7	6	24	0	71	0.00%	6.1/sec	9.5
/styles/circle.css	1500	6	5	23	0	190	0.00%	6.1/sec	2.9
/styles/bootstrap_min.css	1500	23	18	46	0	574	0.00%	6.1/sec	722.1
/scripts/jquery.min.js	1500	19	15	39	0	759	0.00%	6.1/sec	579.2
/scripts/bootstrap_min.js	1500	12	10	29	0	118	0.00%	6.1/sec	222.0
/scripts/polyfills.js	1500	7	6	24	0	461	0.00%	6.1/sec	45.8
/scripts/helpers.js	1500	7	6	23	0	116	0.00%	6.1/sec	13.9
/scripts/cookieconsent.js	1500	12	9	29	0	130	0.00%	6.1/sec	209.0
/images/line_broken.png	1500	47	39	91	0	328	0.00%	6.1/sec	1984.7
/images/logo.jpg	1500	136	114	240	18	550	0.00%	6.1/sec	6863.9
/favicon.ico	1500	9	7	25	0	182	0.00%	6.1/sec	34.0
/bargain/list.do	1500	86	64	181	0	497	0.00%	6.1/sec	52.9
/bargain/display.do	1500	724	686	948	471	1386	0.00%	6.1/sec	71.4
/images/yes.png	1500	8	6	25	0	87	0.00%	6.1/sec	8.8
/sponsorship/list.do	1500	437	412	594	228	1000	0.00%	6.1/sec	50.4
/welcome/index.do	1500	127	96	277	0	932	0.00%	6.1/sec	64.7
TOTAL	27000	100	20	372	0	1386	0.00%	106.4/sec	10708.7

Como podemos observar, a la hora de desplegar el chollo, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 6384 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file
Filename: Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1800	293	215	627	4	2000	0.00%	6.2/sec	66.4
/styles/cookieconsent.min.c...	1800	15	10	31	0	386	0.00%	6.2/sec	25.5
/styles/style.css	1800	14	10	29	0	234	0.00%	6.2/sec	9.6
/styles/circle.css	1800	14	9	29	0	681	0.00%	6.2/sec	2.9
/styles/bootstrap.min.css	1800	37	28	76	0	368	0.00%	6.2/sec	730.4
/scripts/jquery.min.js	1800	31	23	62	0	383	0.00%	6.2/sec	586.0
/scripts/bootstrap.min.js	1800	20	15	44	0	219	0.00%	6.2/sec	224.6
/scripts/polyfills.js	1800	15	10	33	0	377	0.00%	6.2/sec	46.4
/scripts/helpers.js	1800	13	10	30	0	281	0.00%	6.2/sec	14.0
/scripts/cookieconsent.js	1800	19	14	40	0	187	0.00%	6.2/sec	211.4
/images/link_broken.png	1800	74	55	147	0	671	0.00%	6.2/sec	2008.6
/images/logo.jpg	1800	205	167	387	26	1760	0.00%	6.2/sec	6947.3
/favicon.ico	1800	16	11	35	0	378	0.00%	6.2/sec	34.5
/bargain/list.do	1800	223	162	480	0	2353	0.00%	6.2/sec	53.7
/bargain/display.do	1800	966	892	1395	465	3803	0.00%	6.2/sec	72.6
/images/yes.png	1800	15	11	33	0	356	0.00%	6.2/sec	9.0
/sponsorship/list.do	1800	638	567	977	244	2856	0.00%	6.2/sec	51.2
/welcome/index.do	1800	297	219	628	0	3182	0.00%	6.2/sec	65.8
TOTAL	32400	161	28	571	0	3803	0.00%	108.1/sec	10874.6

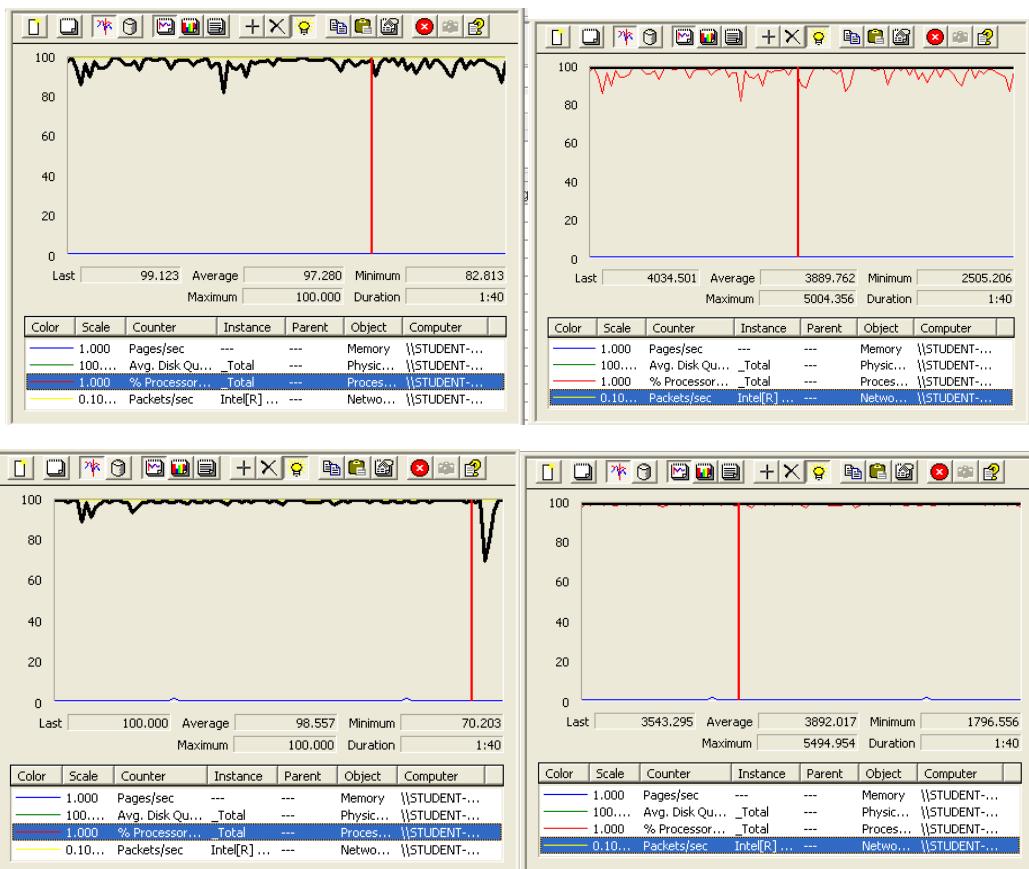
Include group name in label? Save Table Header

En este caso vemos como a la hora de desplegar un chollo, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 6483.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y de la tarjeta de red. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



Reorganizar categorías

En este caso de uso, un moderador puede reorganizar categorías. Para ello, sigue la siguiente secuencia de pasos:

Categorías > (Escoger una) > Mover

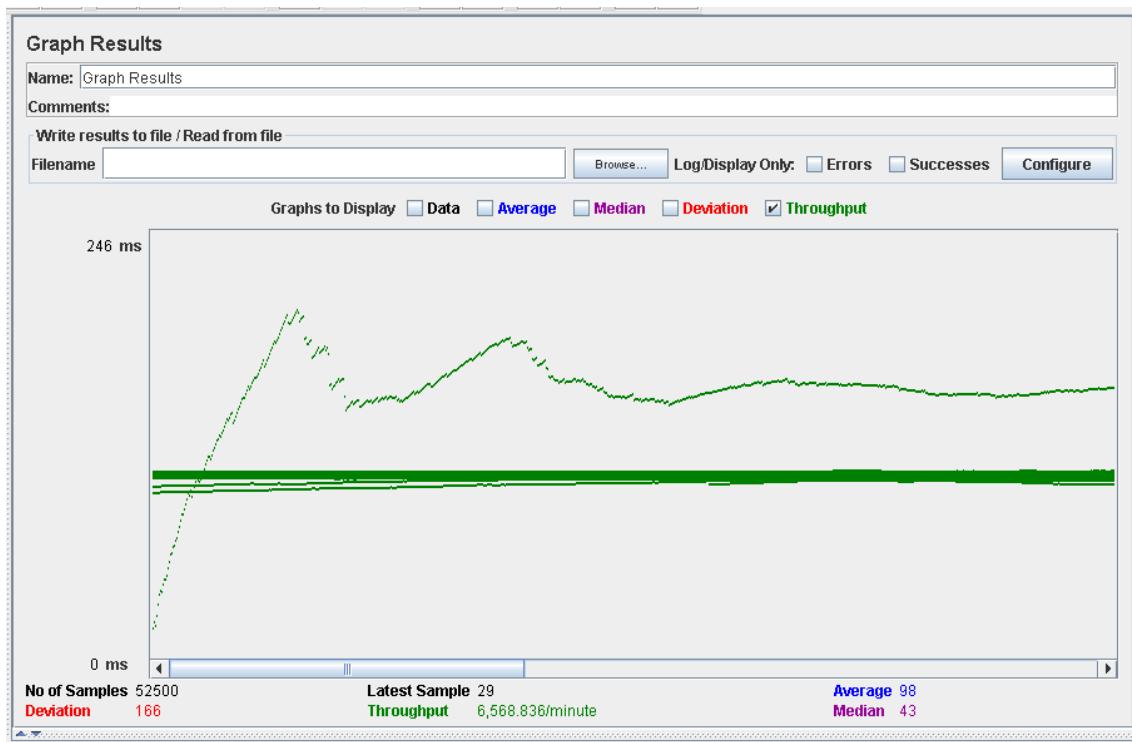
El límite de este caso de uso se sitúa en 70 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Filename <input type="text"/>				Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>	
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	6300	132	76	298	1	1798	0.00%	13.1/sec	143.7
/styles/style.css	2100	12	8	30	0	173	0.00%	4.5/sec	7.0
/styles/bootstrap.min.css	2100	31	23	64	0	414	0.00%	4.5/sec	534.5
/scripts/jquery.min.js	2100	27	20	57	0	332	0.00%	4.5/sec	428.8
/styles/circle.css	2100	12	8	28	0	413	0.00%	4.5/sec	2.2
/scripts/bootstrap.min.js	2100	18	14	40	0	233	0.00%	4.5/sec	164.4
/styles/cookieconsent.min.css	2100	13	9	29	0	395	0.00%	4.5/sec	18.6
/scripts/polyfills.js	2100	12	9	29	0	405	0.00%	4.5/sec	33.9
/scripts/helpers.js	2100	13	9	28	0	293	0.00%	4.5/sec	10.3
/scripts/cookieconsent.js	2100	17	12	36	0	425	0.00%	4.5/sec	154.7
/images/logo.jpg	2100	176	127	363	33	1104	0.00%	4.5/sec	5081.7
/images/link_broken.png	4200	66	47	142	0	448	0.00%	8.8/sec	2860.2
/favicon.ico	2100	14	10	33	0	241	0.00%	4.5/sec	25.2
/security/login.do	2100	72	40	170	0	823	0.00%	4.5/sec	28.1
/j_spring_security_check	2100	159	103	341	7	1477	0.00%	4.5/sec	52.4
/category/display.do	10500	170	134	320	0	3411	0.00%	22.3/sec	208.0
/category/moderator/reorganising...	2100	349	247	624	79	4017	0.00%	4.5/sec	40.4
/j_spring_security_logout	2100	151	88	338	4	2264	0.00%	4.5/sec	49.2
TOTAL	52500	98	43	245	0	4017	0.00%	109.5...	9645.5

Como podemos observar, a la hora de guardar la categoría y volver a la página principal, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 6568 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 80 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

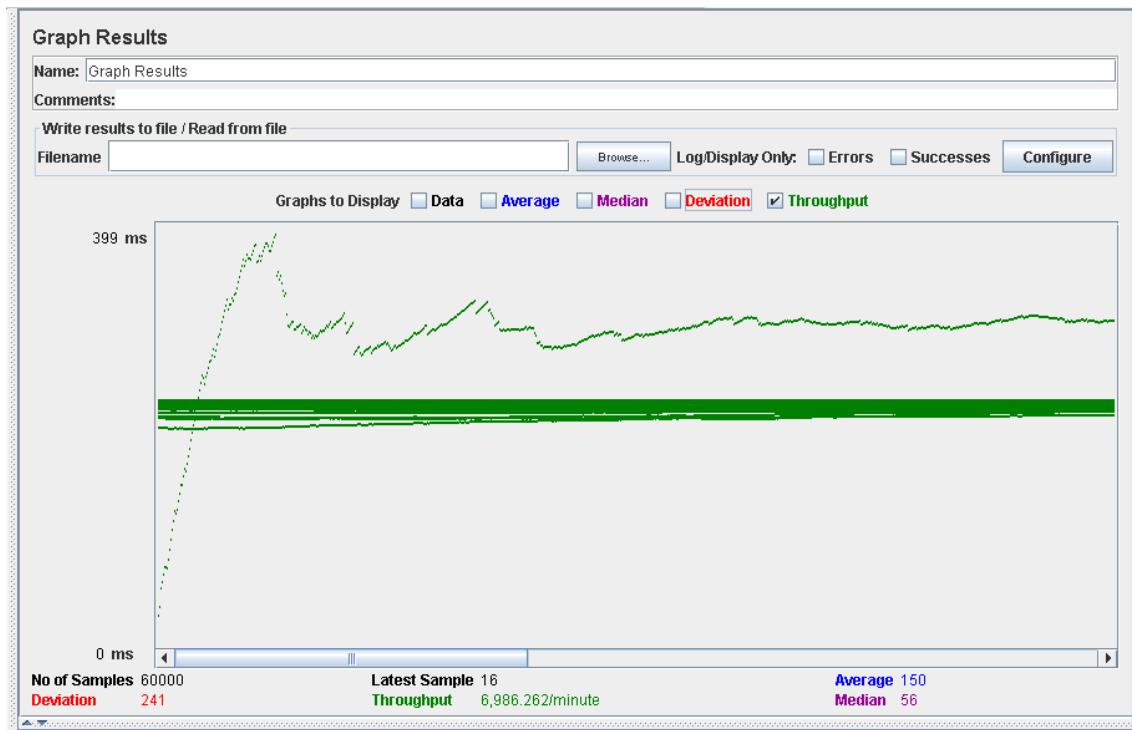
Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	238	158	538	2	3047	0.00%	14.0/sec	152.9
/styles/style.css	2400	13	9	29	0	1004	0.00%	4.8/sec	7.5
/styles/bootstrap.min.css	2400	32	23	62	0	699	0.00%	4.8/sec	567.3
/scripts/jquery.min.js	2400	26	20	52	0	519	0.00%	4.8/sec	455.1
/scripts/circle.css	2400	12	9	26	0	171	0.00%	4.8/sec	2.3
/scripts/bootstrap.min.js	2400	18	13	36	0	462	0.00%	4.8/sec	174.4
/styles/cookieconsent.min.css	2400	12	9	26	0	300	0.00%	4.8/sec	19.8
/scripts/polyfills.js	2400	12	9	27	0	180	0.00%	4.8/sec	36.0
/scripts/helpers.js	2400	12	9	27	0	133	0.00%	4.8/sec	10.9
/scripts/cookieconsent.js	2400	17	13	34	0	200	0.00%	4.8/sec	164.2
/images/logo.jpg	2400	189	120	336	32	1068	0.00%	4.8/sec	5392.1
/images/link_broken.png	4800	61	45	126	0	427	0.00%	9.3/sec	3040.6
/favicon.ico	2400	14	10	31	0	227	0.00%	4.8/sec	26.7
/security/login.do	2400	150	84	340	0	2361	0.00%	4.8/sec	29.8
/j_spring_security_check	2400	326	214	730	6	3360	0.00%	4.8/sec	55.6
/category/display.do	12000	260	196	521	0	3905	0.00%	23.7/sec	218.8
/category/moderator/reorganising...	2400	546	410	1035	84	5178	0.00%	4.8/sec	42.9
/j_spring_security_logout	2400	249	184	558	3	2211	0.00%	4.8/sec	52.3
TOTAL	60000	150	56	398	0	5178	0.00%	116.4/sec	10258.5

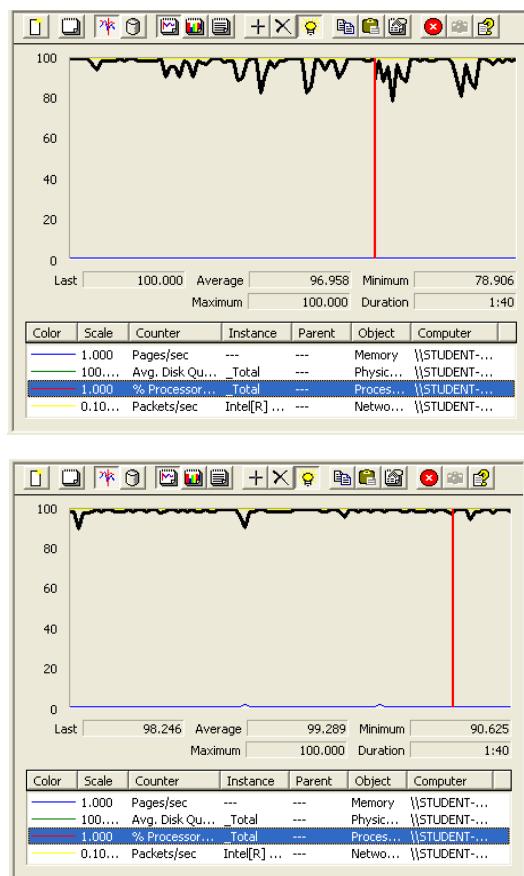
Include group name in label? Save Table Data Save Table Header

En este caso vemos como a la hora de reorganizar la categoría, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 6986.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento del componente en las dos situaciones mencionadas anteriormente, en el mismo orden.



Borrar categoría del chollo

En este caso de uso, una compañía puede borrar categorías del chollo. Para ello, sigue la siguiente secuencia de pasos:

Mis chollos > (Escoger uno) > Eliminar categoría

El límite de este caso de uso se sitúa en 50 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
<input type="text" value="Aggregate Report"/> Name: Aggregate Report <input type="text" value=""/> Comments: <input type="checkbox"/> Write results to file / Read from file <input type="text" value=""/> Filename <input type="button" value="Browse..."/> Log/Display Only: <input type="checkbox"/> Errors <input type="checkbox"/> Successes <input type="button" value="Configure"/>									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/styles/cookieconsent.min.css	4500	72	44	149	15	2373	0.00%	10.9/sec	118.6
/styles/style.css	1500	8	5	18	1	130	0.00%	3.7/sec	15.4
/styles/circle.css	1500	8	5	17	1	131	0.00%	3.7/sec	5.8
/scripts/bootstrap.min.js	1500	12	8	25	2	179	0.00%	3.7/sec	136.4
/scripts/bootstrap_min.css	1500	22	15	45	5	157	0.00%	3.7/sec	443.6
/scripts/jquery.min.js	1500	19	14	38	4	172	0.00%	3.7/sec	355.9
/scripts/polyfills.js	1500	9	5	18	1	492	0.00%	3.7/sec	28.2
/scripts/helpers.js	1500	8	5	18	1	132	0.00%	3.7/sec	8.5
/scripts/cookieconsent.js	1500	12	8	24	2	530	0.00%	3.7/sec	128.4
/images/logo.jpg	1500	133	97	254	36	2203	0.00%	3.7/sec	4216.7
/images/link_broken.png	1500	44	30	88	11	335	0.00%	3.7/sec	1220.4
/security/login.do	1500	36	21	71	8	1962	0.00%	3.7/sec	23.4
/j_spring_security_check	1500	72	52	136	5	786	0.00%	3.7/sec	42.8
/bargain/company/list.do	1500	53	33	95	14	1992	0.00%	3.7/sec	37.5
/bargain/display.do	3000	837	740	1041	594	5315	0.00%	7.4/sec	87.9
/images/yes.png	1500	6	4	14	1	84	0.00%	3.8/sec	5.4
/category/company/removeC...	1500	55	30	92	12	1912	0.00%	3.7/sec	27.8
/bargain/company/removeCa...	1500	55	35	101	14	1269	0.00%	3.7/sec	49.6
/j_spring_security_logout	1500	83	52	158	18	2715	0.00%	3.8/sec	41.0
TOTAL	34500	110	26	261	1	5315	0.00%	83.5/sec	6796.2

Como podemos observar, a la hora de desplegar el chollo, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

Aunque no hay errores, la aplicación comienza a ir más lenta de lo normal.

En la siguiente imagen (abajo), vemos un gráfico que representa el throughput, es decir, la media del número de veces que se ha ejecutado nuestro script en un minuto. En este caso es de 5011 veces por minuto.



Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 60 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report

Name: Aggregate Report
Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	1102	126	65	250	18	2202	0.00%	12.2/sec	132.3
/styles/cookieconsent.min.css	395	18	11	40	2	161	0.00%	4.4/sec	18.0
/styles/style.css	395	19	11	45	1	190	0.00%	4.4/sec	6.8
/styles/circle.css	394	17	9	40	2	161	0.00%	4.4/sec	2.1
/scripts/bootstrap.min.js	394	24	14	56	3	169	0.00%	4.4/sec	159.8
/styles/bootstrap.min.css	394	40	26	89	6	382	0.00%	4.4/sec	519.6
/scripts/jquery.min.js	394	36	23	85	6	215	0.00%	4.4/sec	416.7
/scripts/polyfills.js	394	22	11	53	2	293	0.00%	4.4/sec	33.0
/scripts/helpers.js	393	16	9	39	2	150	0.00%	4.4/sec	10.0
/scripts/cookieconsent.js	392	22	13	50	3	199	0.00%	4.4/sec	149.7
/images/logo.jpg	392	186	127	395	32	889	0.26%	4.4/sec	4892.4
/images/link_broken.png	391	56	36	121	12	427	0.00%	4.4/sec	1433.0
/security/login.do	379	45	22	108	7	879	0.00%	4.4/sec	27.2
/j_spring_security_check	372	99	54	225	22	1683	0.00%	4.3/sec	49.2
/bargain/companylist.do	368	71	34	128	13	1362	0.00%	4.4/sec	43.7
/bargain/display.do	716	839	771	1083	318	3102	1.26%	8.4/sec	99.2
/images/yes.png	363	9	4	21	2	139	0.00%	4.4/sec	6.4
/category/company/removeC...	358	57	38	116	12	481	0.28%	4.4/sec	32.9
/bargain/company/removeCa...	352	92	48	190	14	1183	0.00%	4.4/sec	58.0
/j_spring_security_logout	335	163	103	361	24	1623	0.00%	4.3/sec	47.0
TOTAL	8673	128	33	397	1	3102	0.13%	94.8/sec	7966.2

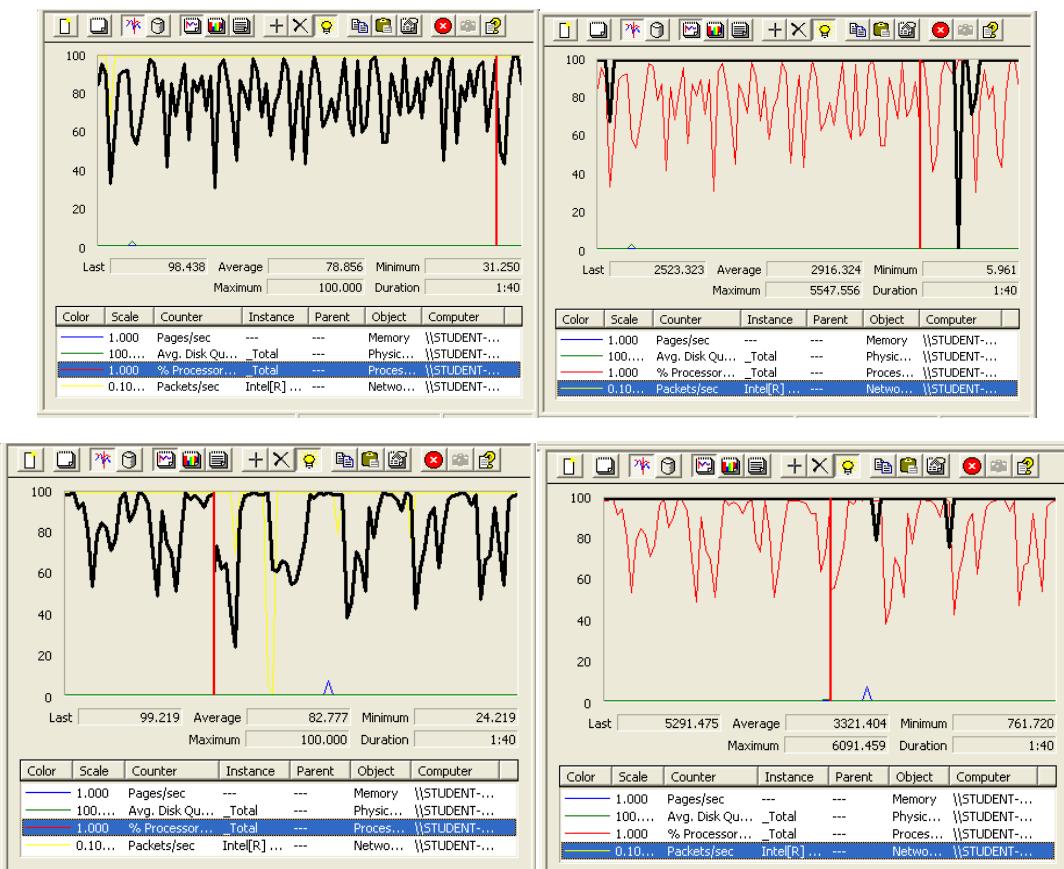
Include group name in label? Save Table Header

En este caso vemos como a la hora de desplegar el chollo, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 5686.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y de la tarjeta de red. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



Conclusión

El mínimo número de usuarios de forma concurrente que pueden utilizar el sistema realizando acciones de forma simultánea es de 45. Este número simboliza el menor número que ha aparecido al realizar los test de rendimiento. De esta forma podemos asegurar que el sistema ha tenido un rendimiento notable en todos los casos cuando el número de usuarios concurrentes no ha superado el valor 45.

Por otro lado, como hemos podido observar, en la mayoría de las ocasiones el procesador es nuestro cuello de botella. Además, en los distintos tests hemos observado que la imagen que utilizamos de logo causa que la página se ralentice debido a su gran tamaño. Sacamos como conclusión que las imágenes del sistema es importante que estén comprimidas pues pueden ser la causa de la mala experiencia para el usuario. Por último, hemos observado que durante los tests de rendimiento de casos de uso como crear o editar chollo tiene un rendimiento algo menor debido a que son actualizadas distintas entidades como son chollos, etiquetas y categoría y el disco duro empieza a actuar como cuello de botella. Lo mismo ocurre con la tarjeta de red, puesto que cuando hay que cargar muchas imágenes se realiza una petición para ver si están rotas y el rendimiento disminuye.