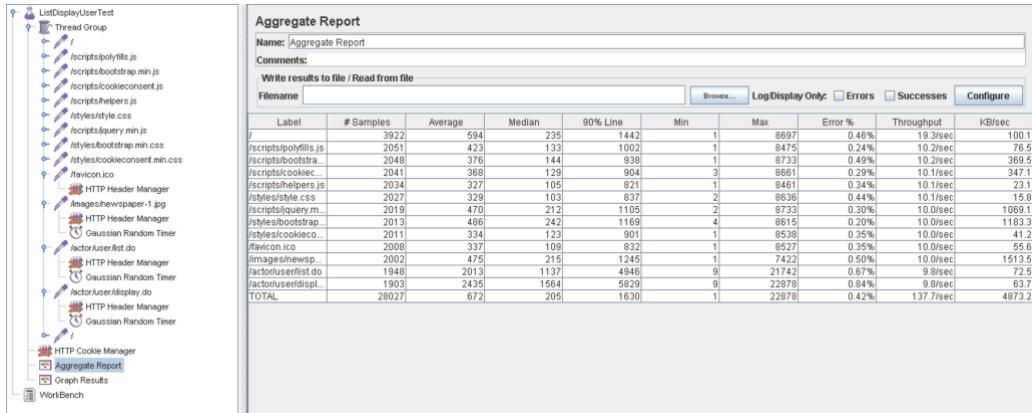


## D11- Test de rendimiento

*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.*

## Listar y desplegar usuarios

En este test vamos a comprobar el rendimiento del sistema a la hora de enfrentarse al caso de uso en el que se listan los usuarios y se despliega uno de ellos. Se ha comenzado el primer test con 200 usuarios, y la ejecución se ha parado a la mitad porque empezaba a haber sendos fallos en distintas peticiones:



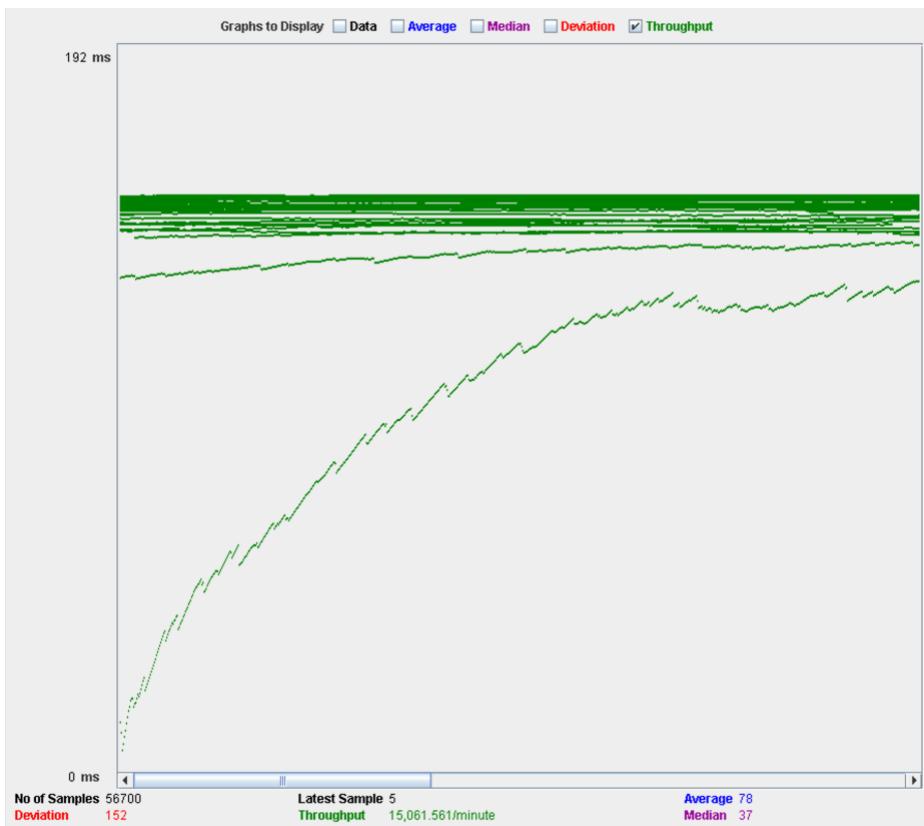
Bajamos a los 150 pero continúa con fallos, aunque muchos más pequeños, además los tiempos empiezan a mostrar valores más aceptables, por lo que estamos cerca de un valor de usuarios más adecuados.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8060	82	46	164	4	5163	0.00%	29.1/sec	151.3
/scripts/polyfill...	4100	62	36	123	2	1842	0.00%	14.9/sec	111.9
/scripts/bootstrap...	4097	81	49	165	3	1828	0.00%	14.9/sec	542.7
/scripts/cookiecon...	4090	78	47	158	3	2018	0.00%	14.8/sec	509.6
/scripts/helpers.j...	4085	74	40	155	1	2052	0.00%	14.8/sec	33.9
/styles/style.css	4083	74	39	158	2	1943	0.00%	14.8/sec	23.3
/scripts/jquery....	4080	123	83	243	5	1868	0.00%	14.9/sec	1593.8
/styles/bootstrap...	4079	123	84	252	5	1937	0.00%	14.9/sec	1763.4
/scripts/cookieco...	4075	78	43	173	2	1925	0.00%	14.8/sec	61.4
/favicon.ico	4074	70	39	142	2	1927	0.00%	14.9/sec	83.1
/images/newspaper...	4072	116	70	244	5	1947	0.00%	14.9/sec	2269.3
/actor/user/list.d...	4048	527	312	1307	12	5922	0.00%	14.9/sec	110.9
/actor/user/displ...	4021	679	431	1828	8	5995	0.10%	14.9/sec	97.7
TOTAL	56964	160	57	355	1	5995	0.01%	205.6/sec	7277.2

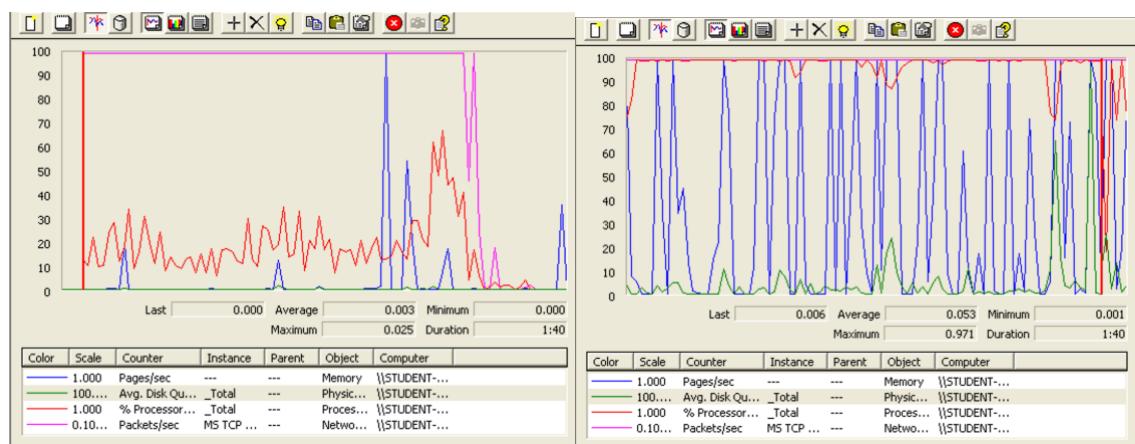
El siguiente valor que probamos es 135 usuarios concurrentes, y efectivamente aquí nos da unos valores mucho más aceptables: no encontramos errores y los tiempos en el 90% de las ocasiones no llega a superar el segundo para realizar una acción. Por ende, podemos concluir que para este caso de uso nuestro sistema no aguanta más de 135 usuarios concurrentes.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	53	32	89	3	5227	0.00%	35.9/sec	186.2
/scripts/polyfill...	4050	40	27	75	2	1414	0.00%	18.5/sec	139.1
/scripts/bootstrap...	4050	52	35	94	3	1466	0.00%	18.5/sec	673.4
/scripts/cookiecon...	4050	49	33	88	2	1606	0.00%	18.5/sec	633.9
/scripts/helpers.j...	4050	42	28	78	2	1633	0.00%	18.5/sec	42.2
/styles/style.css	4050	41	28	78	2	1446	0.00%	18.5/sec	28.9
/scripts/jquery....	4050	68	47	132	4	1649	0.00%	18.5/sec	1982.0
/styles/bootstrap...	4050	70	48	135	5	1488	0.00%	18.5/sec	2193.4
/styles/cookieco...	4050	40	28	76	2	1618	0.00%	18.5/sec	76.5
/favicon.ico	4050	40	27	74	2	1503	0.00%	18.5/sec	103.4
/images/newspaper...	4050	81	46	152	5	1709	0.00%	18.5/sec	2811.1
/actor/user/list.d...	4050	207	139	410	10	2679	0.00%	18.5/sec	137.2
/actor/user/displ...	4050	253	175	500	11	3486	0.00%	18.5/sec	121.2
TOTAL	56700	78	37	165	2	5227	0.00%	251.0/sec	8867.2

El Throughput en la última prueba ha arrojado un valor de 15018.

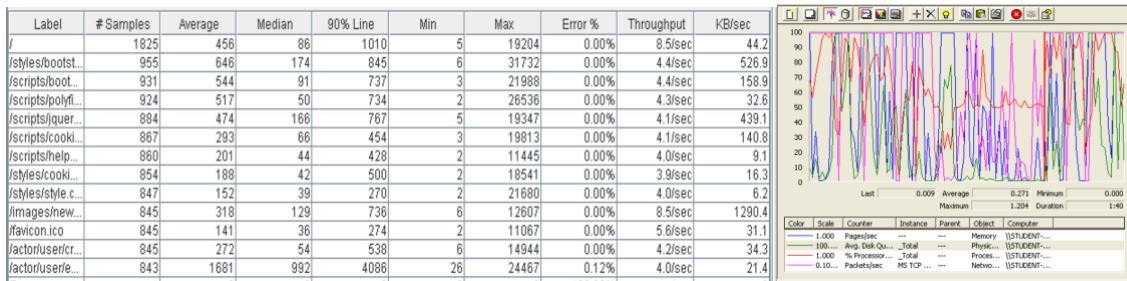


En la imagen de la izquierda podemos ver la gráfica en el caso de 135 usuarios y a la derecha en el caso de 150. Se puede apreciar como el procesador es el mayor limitador de nuestro sistema en este caso de uso, aunque junto con la interfaz de red que está todo el tiempo en sus límites.



## Registrarse como usuario

Para este test hemos utilizado lo aprendido en el A+ del D10 en el que añadíamos más realidad al sistema utilizando un CSV con datos. En este caso nuestro CSV contiene 20000 usuarios que vamos a crear en los distintos tests, comenzando con 150 usuarios concurrentes. Podemos apreciar como el sistema ha llegado prácticamente a su límite pues empezamos a ver errores y la gráfica del rendimiento del equipo muestra como todos los campos, tanto disco duro, como procesador y tarjeta gráfica tienen picos que llegan hasta el 100% de su uso.

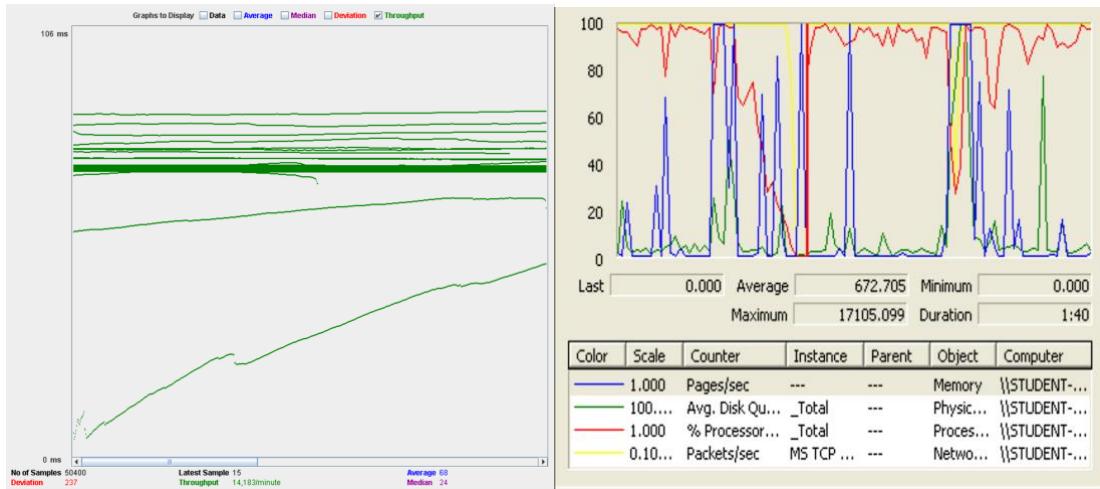


Por ello bajamos hasta los 125 usuarios, pero, como muestra la siguiente imagen, los resultados siguen mostrando tiempos superiores a un segundo, por lo que, para ser exactos, vamos a reducir el número de usuarios un poco, 120 usuarios.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7500	55	28	71	3	2756	0.00%	31.6/sec	164.2
/styles/bootstrap...	3750	68	48	129	5	2736	0.00%	16.3/sec	1932.7
/scripts/bootstrap...	3750	42	31	74	3	863	0.00%	16.3/sec	594.5
/scripts/polyfill...	3750	36	25	81	2	2710	0.00%	16.3/sec	123.0
/scripts/jquery...	3750	63	46	118	4	2778	0.00%	16.3/sec	1750.6
/scripts/cookie...	3750	42	29	77	2	2742	0.00%	16.5/sec	564.9
/scripts/helper...	3750	33	25	60	2	1896	0.00%	16.6/sec	38.0
/styles/cookie...	3750	35	24	80	2	2734	0.00%	16.9/sec	69.8
/styles/style.css	3750	40	25	63	2	2734	0.00%	17.1/sec	26.8
/images/news...	3750	78	52	152	6	2791	0.00%	17.3/sec	2640.6
/favicon.ico	3750	36	16	51	2	2706	0.00%	17.4/sec	97.1
/actor/user/cre...	3750	66	25	89	5	2729	0.00%	17.3/sec	143.2
/actor/user/edi...	3750	499	334	1008	14	5560	0.00%	17.4/sec	91.9
TOTAL	52500	82	31	143	2	5550	0.00%	221.4/sec	7812.7

Ya con un valor de 120 usuarios concurrentes, los valores son más aceptables como podemos apreciar puesto que las acciones no suman más un segundo ni apreciamos errores y por ende, podemos decir que el límite de usuarios que aguanta nuestro sistema es de 120.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7200	40	23	59	3	4651	0.00%	33.8/sec	175.3
/styles/bootstrap...	3600	62	37	122	4	3809	0.00%	17.5/sec	2071.9
/scripts/bootstrap...	3600	43	24	71	2	3599	0.00%	17.5/sec	637.0
/scripts/polyfill...	3600	34	19	53	1	3643	0.00%	17.5/sec	131.5
/scripts/jquery...	3600	56	34	103	4	3601	0.00%	17.5/sec	1876.3
/scripts/cookie...	3600	37	23	66	2	3639	0.00%	17.5/sec	601.1
/scripts/helper...	3600	29	18	52	2	3610	0.00%	17.5/sec	40.0
/styles/cookie...	3600	27	18	50	2	3520	0.00%	17.6/sec	72.7
/styles/style.css	3600	31	19	50	2	3854	0.00%	17.6/sec	27.5
/images/news...	3600	57	38	113	5	3594	0.00%	17.6/sec	2678.7
/favicon.ico	3600	32	11	39	2	3496	0.00%	17.6/sec	98.3
/actor/user/cre...	3600	55	18	51	4	3960	0.00%	17.6/sec	145.1
/actor/user/edi...	3600	412	189	956	13	5057	0.00%	17.6/sec	92.6
TOTAL	50400	68	24	103	1	5057	0.00%	236.4/sec	8341.7

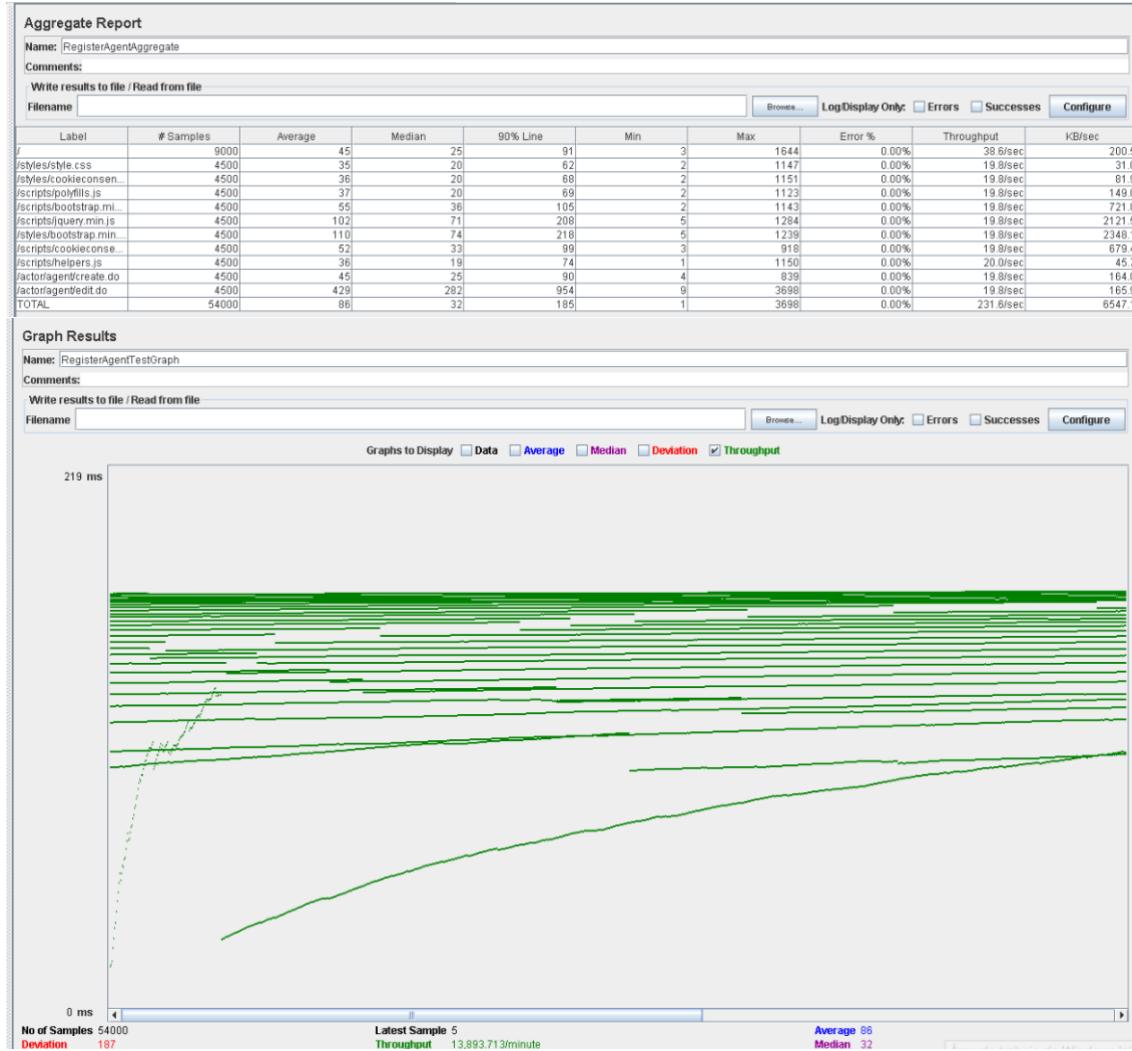


El valor el Throughput es de 14183 y en el gráfico del rendimiento del sistema podemos ver como el procesador se mantiene por mayormente alrededor del 100% y la tarjeta gráfica está al 100% de su capacidad todo el tiempo.

## Registrarse como agente.

En este caso de uso, accedemos a la página, pulsamos en el menú “Create agent”, rellenamos el formulario y guardamos.

Comenzamos probando con 130 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores:



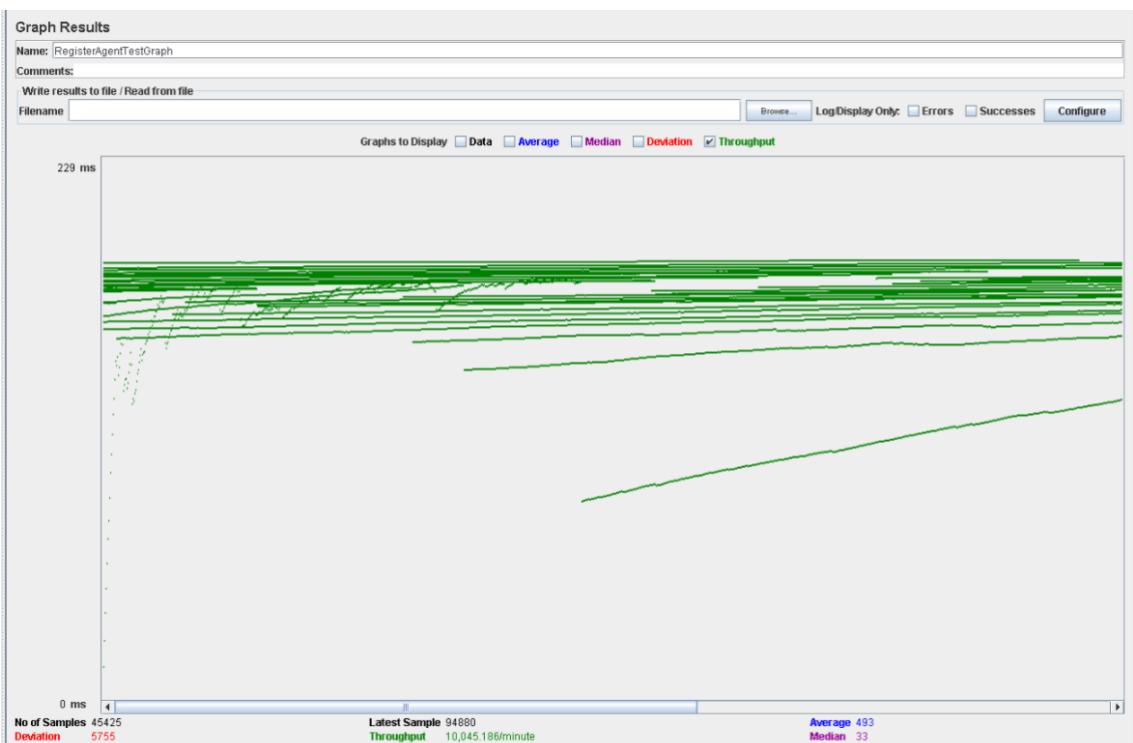
En esta tabla, el rendimiento es de aproximadamente 13.900 ejecuciones por minuto.

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.

La configuración en esta ocasión es de 170 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:

Aggregate Report										
Name: RegisterAgentAggregate										
Comments:										
Write results to file / Read from file										
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	7432	50	26	101	3	981	0.00%	42.3/sec	220.0	
/styles/style.css	3801	35	21	72	2	899	0.00%	21.6/sec	33.9	
/styles/cookieconsen...	3801	36	20	76	2	644	0.00%	21.6/sec	89.6	
/scripts/bootstrap.mi...	3801	37	21	77	2	594	0.00%	21.6/sec	163.0	
/scripts/jquery.min.js	3801	53	35	105	3	834	0.00%	21.6/sec	789.4	
/styles/bootstrap.min...	3801	89	65	180	6	1088	0.00%	21.6/sec	2319.0	
/scripts/cookieconsen...	3801	95	68	196	6	874	0.00%	21.5/sec	2553.8	
/scripts/heplers.js	3801	57	31	131	3	1017	0.00%	21.5/sec	739.1	
/actoragent/create.do	3801	465	28	140	4	91754	0.45%	14.3/sec	118.1	
/actoragent/edit.do	3784	4907	844	3064	16	97749	4.04%	14.2/sec	116.0	
TOTAL	45425	493	33	242	2	97749	0.37%	167.4/sec	4746.0	

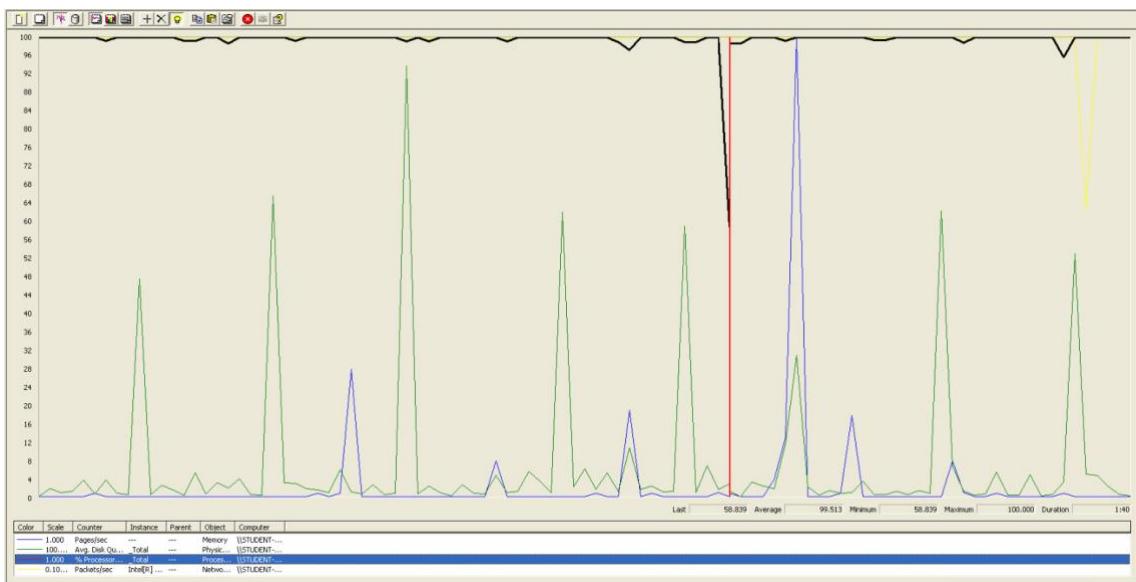
Como podemos observar fijándonos en 90% Line, la dirección /actor/agent/edit.do genera un tiempo de respuesta (3064 ms) mayor a lo establecido (1000 ms).



En esta tabla, el rendimiento es de aproximadamente 10.000 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.

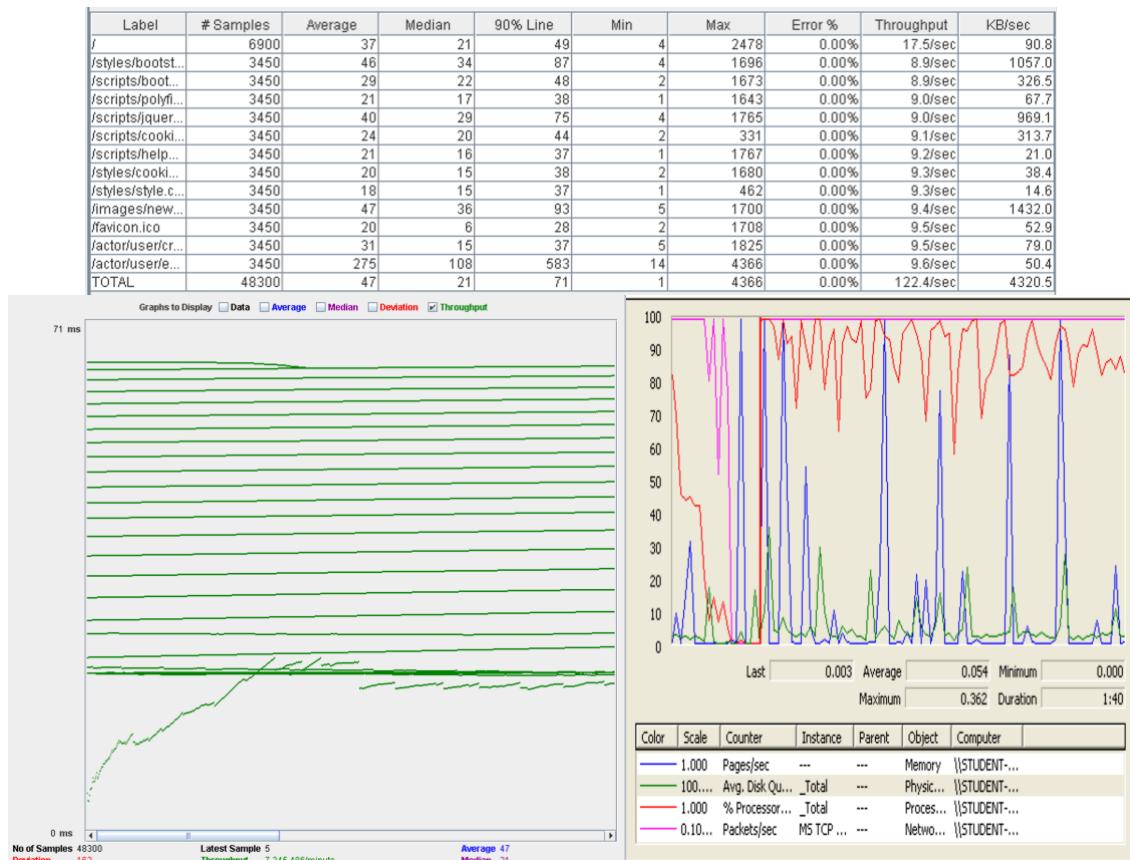
Por lo que, en este caso de uso, los usuarios permitidos serán entre 130 y 140.

## Registrarse como cliente

Este caso de uso es igual que el 02, pero en vez de estar crear usuarios vamos a crear clientes. Al igual que en el caso 02 utilizamos CSV. Comenzamos con un valor comedido de 130 usuarios concurrentes. El resultado (una vez estabilizado el throughput) de este test arroja que, aunque los tiempos están bien, dentro de los rangos aceptables, ocurren errores menores. Nos extrañó que ocurrieran errores con un valor de usuarios tan bajo, pero tras revisar el log de Tomcat no apreciamos fallos.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5794	56	25	73	4	2618	0.00%	37.3/sec	193.8
/styles/bootst...	2928	71	47	141	5	1815	0.00%	19.0/sec	2259.5
/scripts/boot...	2904	48	28	92	3	958	0.00%	19.2/sec	700.0
/scripts/polyfi...	2904	30	22	59	2	948	0.00%	19.0/sec	142.8
/scripts/quer...	2901	62	44	118	5	1815	0.00%	19.1/sec	2042.5
/scripts/cooki...	2899	38	26	77	3	933	0.00%	19.1/sec	654.2
/scripts/help...	2897	32	22	55	2	991	0.00%	19.0/sec	43.3
/styles/cooki...	2894	31	21	60	2	1772	0.00%	19.0/sec	78.4
/styles/style.c...	2893	29	20	54	1	1003	0.00%	19.0/sec	29.7
/images/new...	2892	70	49	136	6	1112	0.00%	19.6/sec	2988.0
/favicon.ico	2892	26	12	42	2	883	0.00%	19.3/sec	108.1
/actor/user/cr...	2891	31	18	52	5	1535	0.00%	19.5/sec	161.8
/actor/user/e...	2890	294	129	776	8	3721	0.10%	19.4/sec	163.6
TOTAL	40579	62	27	111	1	3721	0.01%	261.1/sec	9291.1

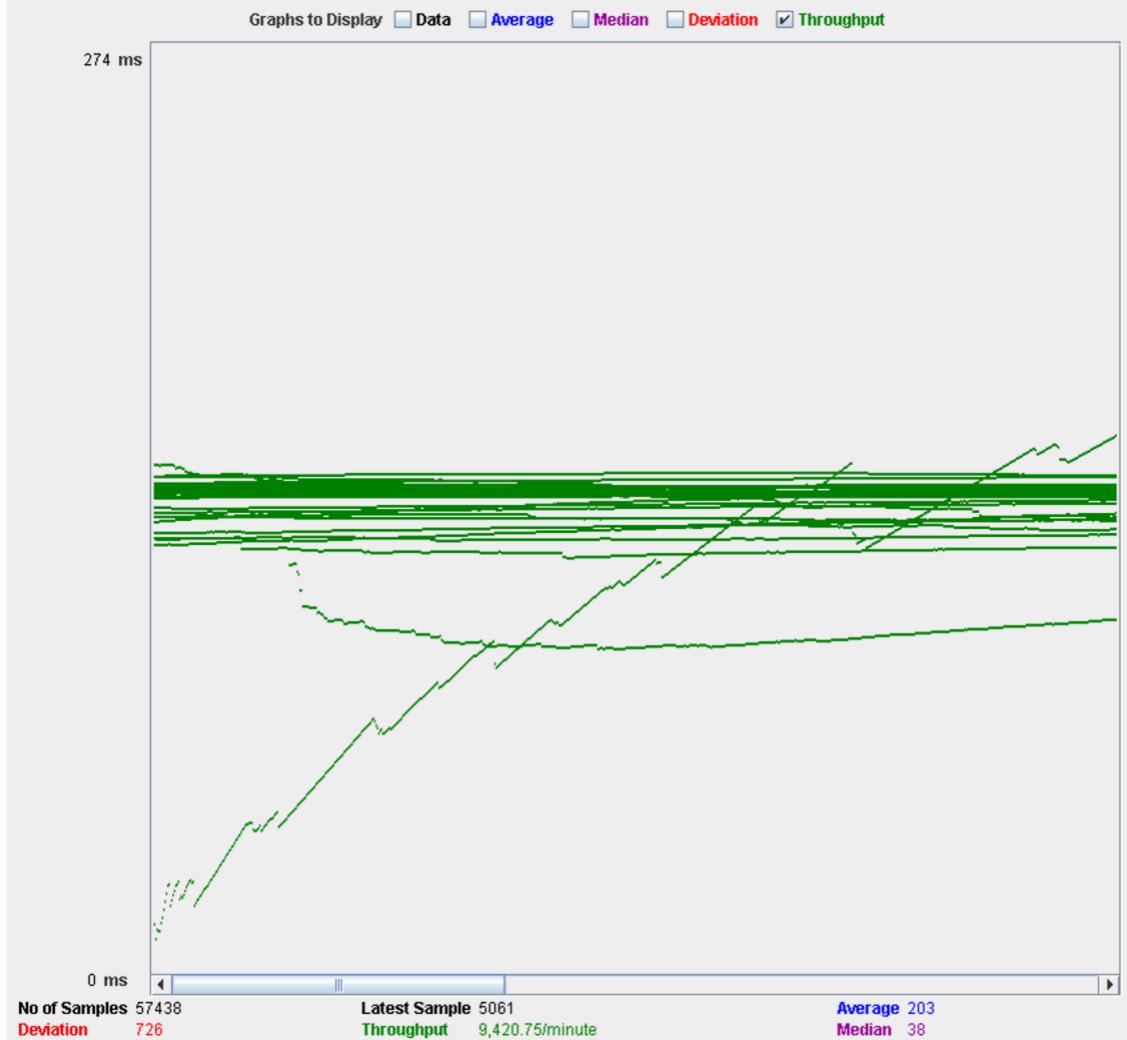
Hemos bajado hasta los 115 usuarios y ya nos da unos valores mucho mejores, sin errores y con unos tiempos perfectos. El Throughput es de 7345 aunque los valores no se han estabilizado como debería. Por último, en cuanto al rendimiento del sistema, podemos ver como la tarjeta de red está al 100% de su capacidad al igual que el procesador que alcanza altas cotas de ocupación como ya ha ocurrido en otros tests.



## Editar usuario

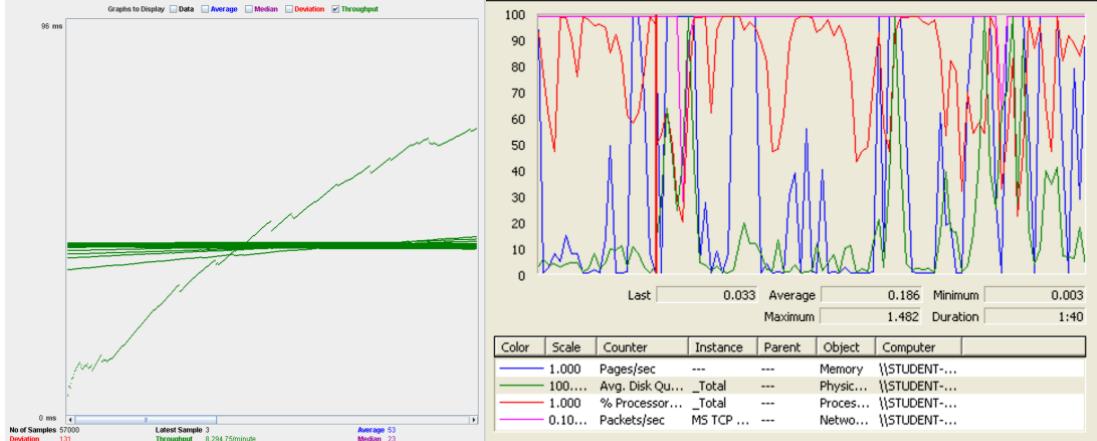
Aprovechando la base de datos con los varios miles de usuarios que creamos en el caso de uso 02, hemos realizado este caso de uso que consiste en editar los usuarios. Primero hemos probado con un valor de 150 usuarios concurrentes pero los resultados nos dan que los tiempos no son nada buenos, provocado en gran medida por la gran cantidad de datos que existe en la base de datos.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	11802	84	30	171	3	3578	0.00%	33.0/sec	181.0
/styles/cooki...	3059	59	29	154	2	1460	0.00%	8.6/sec	35.6
/styles/bootst...	3059	105	63	229	6	1893	0.00%	8.6/sec	1021.9
/styles/style.c...	3059	53	29	112	2	3454	0.00%	8.6/sec	13.5
/scripts/boot...	3059	78	39	180	3	3440	0.00%	8.6/sec	314.7
/scripts/help...	3059	62	28	147	2	3587	0.00%	8.6/sec	19.7
/scripts/jquer...	3059	106	58	217	5	3488	0.00%	8.6/sec	927.1
/scripts/polyfi...	3059	54	27	102	2	1882	0.00%	8.7/sec	65.2
/scripts/cooki...	3059	68	35	109	2	3531	0.00%	8.7/sec	297.1
/images/new...	3058	85	57	187	5	2390	0.00%	8.7/sec	1322.6
/favicon.ico	3053	30	7	47	1	3110	0.00%	8.7/sec	48.7
/security/logi...	3035	52	11	83	4	3864	0.00%	8.7/sec	48.1
/j_spring_se...	2977	568	38	1639	9	7627	0.00%	8.5/sec	51.3
/actor/profile....	2912	1607	547	4888	50	9189	0.00%	8.4/sec	68.7
/actor/user/e...	2910	462	95	972	7	7531	0.00%	8.5/sec	69.4
/j_spring_se...	2909	131	73	291	6	2318	0.00%	8.6/sec	46.6
TOTAL	57128	199	38	278	1	9189	0.00%	159.7/sec	4473.9



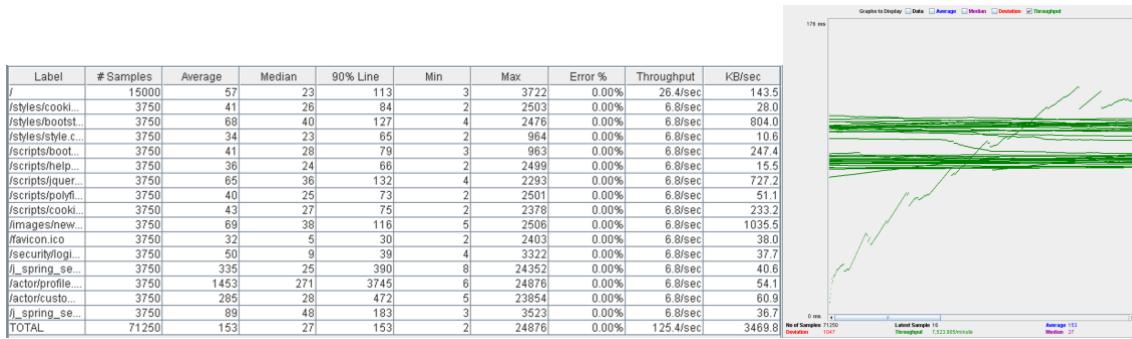
Probamos a disminuir el número de usuarios hasta los 130 lo cual nos dará unos tiempos mucho más favorables y dentro de los rangos de los aceptable para que el usuario no se sienta obligado a abandonar la web. Por otro lado, podemos ver que el throughput que arroja jMeter es de 8294. Por último, el rendimiento del sistema, como podemos apreciar en la imagen que del perfmon, nos muestra que esta tarea ha tenido trabajando al 100% al disco duro, tarjeta de red y procesador, lo cual es lógico teniendo en cuenta que es un caso de uso que contempla tanto la escritura como la lectura de grandes cantidades de datos diferentes.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	12000	31	19	61	3	1208	0.00%	29.1/sec	159.7
/styles/cookie...	3000	24	19	43	2	1069	0.00%	7.5/sec	31.2
/styles/bootstrap...	3000	47	40	81	4	1100	0.00%	7.5/sec	894.9
/styles/style.css	3000	25	20	41	2	1068	0.00%	7.5/sec	11.8
/scripts/bootstrap...	3000	30	25	54	2	1008	0.00%	7.5/sec	275.3
/scripts/help...	3000	21	19	39	2	287	0.00%	7.5/sec	17.2
/scripts/jquery...	3000	43	37	74	4	456	0.00%	7.5/sec	809.4
/scripts/polyfill...	3000	23	20	44	2	1062	0.00%	7.5/sec	56.8
/scripts/cookies...	3000	32	23	52	3	436	0.00%	7.6/sec	259.2
/images/news...	3000	47	39	82	6	660	0.00%	7.6/sec	1150.7
/favicon.ico	3000	7	4	15	2	524	0.00%	7.6/sec	42.2
/security/login...	3000	9	8	13	4	154	0.00%	7.6/sec	41.9
/j_spring_se...	3000	37	18	67	6	791	0.00%	7.6/sec	45.3
/actor/profile....	3000	336	209	729	48	2517	0.00%	7.5/sec	61.3
/actor/user/e...	3000	153	49	404	9	2021	0.00%	7.5/sec	44.8
/j_spring_se...	3000	48	36	81	5	1017	0.00%	7.5/sec	40.6
TOTAL	57000	53	23	90	2	2517	0.00%	138.2/sec	3805.9



## Editar cliente

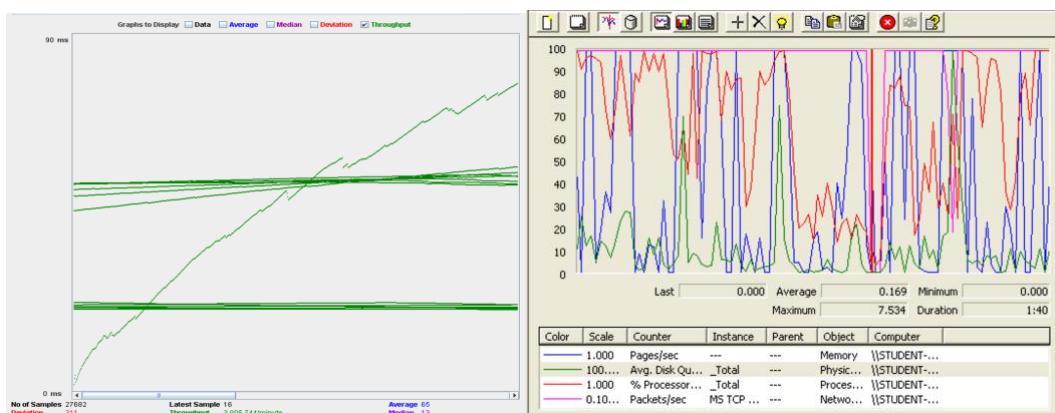
Al igual que en el anterior caso de uso, hemos aprovechado la cantidad de clientes que existen en la base de datos para realizar este caso de uso, por lo que usamos CSV. Empezamos pues el test con un valor de 125 usuarios concurrentes, lo cual da un tiempo demasiado alto en el 90% de los casos, todo ello seguramente provocado porque en la base de datos existen muchos clientes y tiene que buscar uno entre todos ellos.



Hemos bajado la cifra de usuarios concurrente hasta los 100 usuarios y los resultados han estado en el límite del rango de tiempo aceptable, además no hemos tenido ningún error. Es por ello que podemos concluir que nuestro sistema aguanta 100 usuarios concurrente editando sus perfiles, cada usuario con uno distinto gracias a utilizar el CSV.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5825	65	10	61	3	9186	0.00%	10.5/sec	57.5
/styles/cooki...	1508	43	10	38	2	2812	0.00%	2.8/sec	11.6
/styles/bootst...	1477	72	20	77	5	7799	0.00%	2.7/sec	325.5
/styles/style.c...	1477	23	9	30	1	1573	0.00%	2.7/sec	4.3
/scripts/boot...	1469	50	13	51	2	8799	0.00%	2.7/sec	99.6
/scripts/help...	1469	43	9	31	1	6862	0.00%	2.7/sec	6.2
/scripts/jquer...	1456	52	17	61	4	3466	0.00%	2.7/sec	290.3
/scripts/polym...	1456	30	9	31	1	2989	0.00%	2.7/sec	20.4
/scripts/cooki...	1456	30	12	39	2	2481	0.00%	2.7/sec	93.0
/images/innew...	1445	36	20	60	5	1225	0.00%	2.7/sec	409.6
/favicon.ico	1443	7	4	8	2	1035	0.00%	2.7/sec	15.0
/security/logi...	1442	22	7	17	4	6221	0.00%	2.7/sec	14.9
/j_spring_se...	1442	42	17	57	9	2980	0.00%	2.7/sec	16.0
/actor/profile...	1440	331	105	851	45	3258	0.00%	2.7/sec	21.8
/actor/custo...	1438	85	14	178	5	4080	0.00%	2.7/sec	21.8
/j_spring_se...	1438	113	18	86	4	9438	0.00%	2.7/sec	14.4
TOTAL	27682	65	13	82	1	9438	0.00%	50.1/sec	1385.5

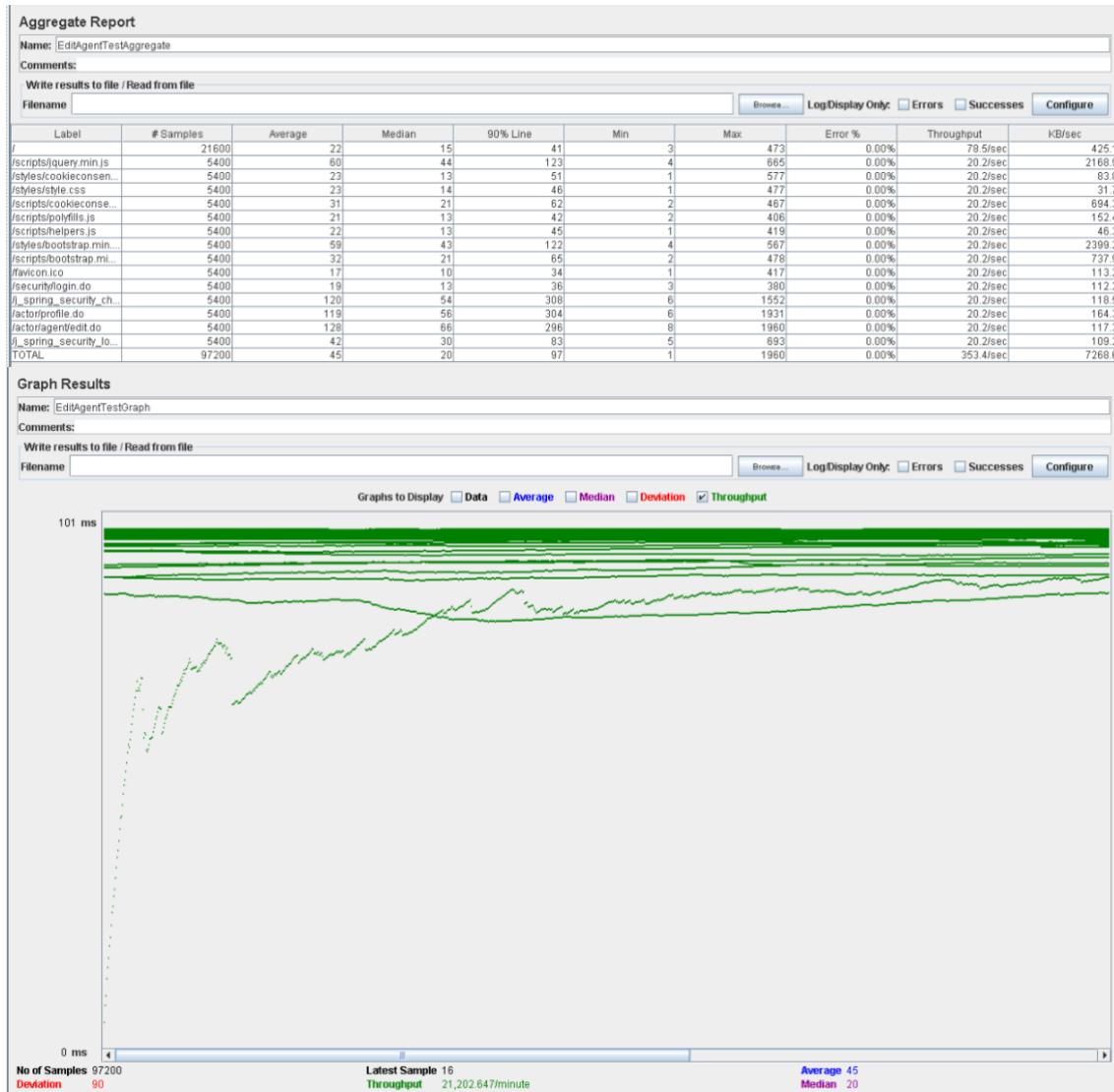
El Throughput se estabilizó en las líneas que vemos debajo de la imagen, con un valor de 3005. Por último, la gráfica de rendimiento nos muestra como el procesador es en buena parte el cuello de botella de la web.



## Editar agente.

En este caso de uso, accedemos a la página, nos autenticamos como agent, accedemos a nuestro perfil, editamos la información que deseemos y nos desautenticamos.

Probamos una configuración 180 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores ni esperas demasiado altas:



En esta tabla, el rendimiento es de aproximadamente 21.200 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.

La configuración en esta ocasión es de 200 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:

Aggregate Report										
<input type="button" value="Browse..."/> <input type="checkbox"/> LogDisplay 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	
/	24000	39	20	86	3	1117	0.00%	0.0/sec	407.4	
/scripts/query_min.js	6000	84	59	173	5	1085	0.00%	20.0/sec	225.6	
/styles/cookieconsen...	6000	40	19	96	1	961	0.00%	20.0/sec	86.0	
/styles/style.css	6000	39	19	96	2	1291	0.00%	20.0/sec	32.6	
/scripts/cookieconse...	6000	48	28	99	3	1006	0.00%	20.0/sec	713.0	
/scripts/bootstrap.js	6000	36	18	84	2	957	0.00%	20.0/sec	156.5	
/scripts/sheepers.js	6000	38	18	91	2	940	0.00%	20.0/sec	47.5	
/styles/bootstrap.min...	6000	82	59	170	6	1074	0.00%	20.0/sec	2484.4	
/scripts/bootstrap.min...	6000	52	29	117	2	1019	0.00%	20.0/sec	767.9	
favicon.ico	6000	35	16	76	1	982	0.00%	20.0/sec	116.3	
/security/login.do	6000	35	19	75	3	909	0.00%	20.0/sec	115.3	
/j_spring_security_ch...	6000	186	99	456	6	2188	0.02%	20.0/sec	121.7	
/actor/profile/do	6000	160	88	368	6	2270	0.00%	20.0/sec	168.7	
/actoragent/edit.do	6000	199	116	457	8	2253	0.02%	20.7/sec	120.4	
/j_spring_security_lo...	6000	71	44	150	5	1198	0.00%	20.7/sec	112.2	
TOTAL	108000	70	29	163	1	2270	0.00%	363.6/sec	7478.5	

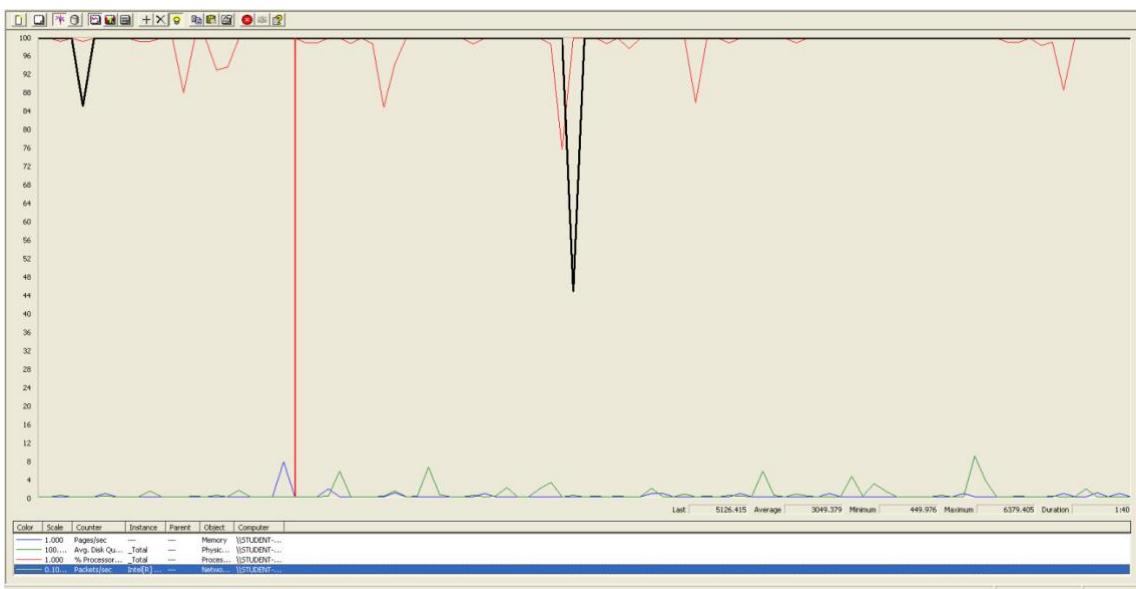
Como podemos observar fijándonos en 90% Line, las direcciones `j_spring_security_check`, `/actor/profile.do` y `/actor/agent/edit.do` generan un tiempo de respuesta mayor a lo normal en este tipo de caso de uso, 456, 368 y 457 ms respectivamente.



En esta tabla, el rendimiento es de aproximadamente 21.800 ejecuciones por minuto (aumenta levemente al aumentar también en poca cantidad los 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.

Por lo que podemos afirmar que la aplicación soporta entre 180 y 200 usuarios en este caso de uso. Ahora estudiaremos a que se deben los tiempos y errores:

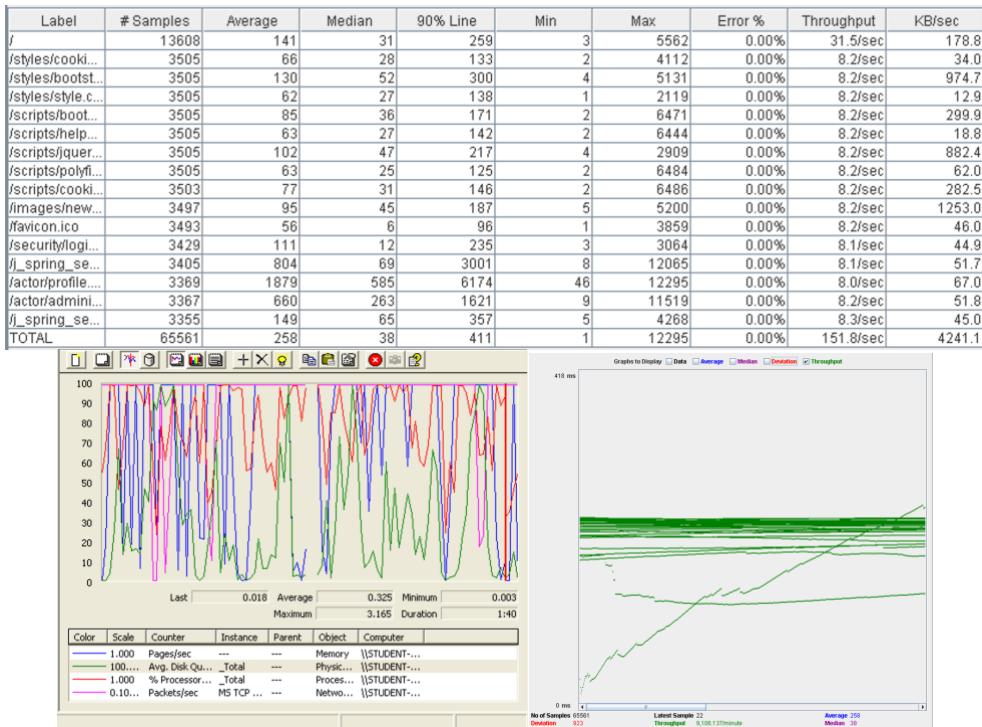
Abrimos `performance.exe` y vemos la tarjeta de red tiene porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de la tarjeta de red.



Performance con Packets/sec.

## Editar administrador

Este caso de uso consiste en editar el administrador. En este test no se ha utilizado un fichero CSV porque en nuestro sistema solo existe un administrador. En el primer test hemos utilizado 150 usuarios concurrentes y aunque no da errores (para ello nos tendríamos que ir alrededor de los 190-200 usuarios) el tiempo que sufren el 90% de los usuarios para realizar las acciones superan enormemente el segundo, por ello se hace necesario bajar los usuarios para intentar obtener los usuarios.



Al bajarlo a 115 usuarios concurrentes, vemos que, aunque los tiempos han mejorado notablemente, sigue estando el tiempo que el 90% de usuarios sufre por un tiempo por encima de un segundo.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13800	38	20	81	3	2615	0.00%	23.7/sec	134.7
/styles/cookie...	3450	42	25	63	2	1880	0.00%	6.3/sec	26.0
/styles/bootstrap...	3450	70	46	127	6	1537	0.00%	6.3/sec	745.1
/styles/style.c...	3450	35	24	63	2	1370	0.00%	6.3/sec	9.8
/scripts/bootstrap...	3450	41	30	79	3	1623	0.00%	6.2/sec	227.9
/scripts/help...	3450	30	23	60	2	413	0.00%	6.2/sec	14.2
/scripts/jquery...	3450	56	42	112	5	632	0.00%	6.2/sec	667.2
/scripts/polyfill...	3450	29	23	56	2	1253	0.00%	6.2/sec	46.8
/scripts/cookie...	3450	35	27	67	3	1373	0.00%	6.2/sec	212.8
/images/new...	3450	61	43	113	6	1433	0.00%	6.2/sec	942.6
/favicon.ico	3450	22	5	20	2	1354	0.00%	6.2/sec	34.5
/security/logi...	3450	14	8	19	3	671	0.00%	6.2/sec	34.2
/j_spring_se...	3450	103	18	126	6	10938	0.00%	6.2/sec	39.3
/actor/profile...	3450	733	214	1368	51	10777	0.00%	6.1/sec	51.1
/actor/admini...	3450	208	54	539	11	9824	0.00%	6.2/sec	39.3
/j_spring_se...	3450	58	43	110	6	607	0.00%	6.2/sec	33.4
TOTAL	65550	89	27	119	2	10938	0.00%	112.7/sec	3113.3

Estos tiempos tan altos son debido a la gran cantidad de datos que existen en la base de datos. Por ello, vamos a hacer la prueba de como rendiría el sistema con 115 usuarios y la base de datos con los datos que vienen en el populate que hemos creado.

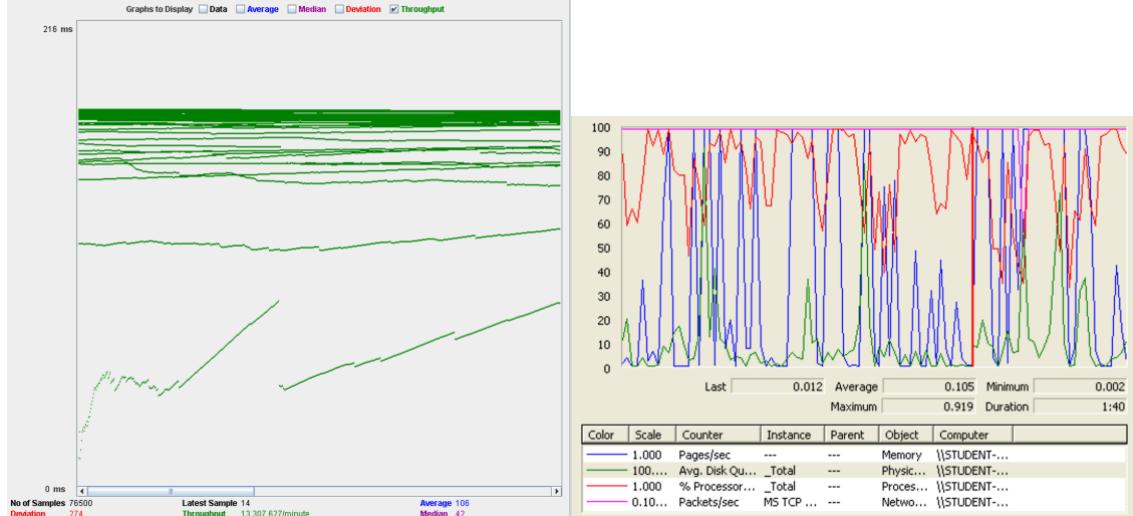
Como se puede apreciar el cambio de rendimiento es bestial. Esta diferencia hace notoria la influencia que una base de datos de gran tamaño tiene sobre el rendimiento de la web y lo conveniente que es optimizarla y contar con un buen sistema de bases de datos.

## Listar chirps

Ahora procedemos a realizar el test de rendimiento del caso de uso de listar chirps autenticado como un usuario. Para empezar, vamos a realizar el test con 150 usuarios que nos da un valor perfecto, sin necesidad de hacer más pruebas porque no supera el segundo ninguno de los tiempos del 90% de los usuarios por lo que podemos concluir que nuestro sistema para este test soporta 150 usuarios concurrentes.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	76	29	125	3	3418	0.00%	39.1/sec	211.0
/favicon.ico	4500	61	39	132	2	3018	0.00%	13.4/sec	75.2
/images/new...	4500	104	71	220	5	3551	0.00%	13.5/sec	2060.8
/scripts/polyfl...	4500	73	40	134	2	3041	0.00%	13.6/sec	102.2
/scripts/help...	4500	82	40	172	2	3146	0.00%	13.6/sec	31.0
/scripts/cookie...	4500	79	46	141	3	3014	0.00%	13.6/sec	465.6
/scripts/boot...	4500	90	48	161	3	3509	0.00%	13.6/sec	495.4
/styles/bootstrap...	4500	132	62	215	5	3042	0.00%	13.6/sec	1612.3
/styles/style.c...	4500	84	33	129	2	3508	0.00%	13.6/sec	21.3
/styles/cookie...	4500	78	26	120	2	3006	0.00%	13.6/sec	56.3
/scripts/jquery...	4500	76	16	119	4	2926	0.00%	13.6/sec	1458.9
/security/login...	4500	107	15	144	3	3501	0.00%	13.6/sec	75.4
/j_spring_se...	4500	127	74	301	9	3714	0.00%	13.6/sec	81.7
/chirp/user/li...	4500	368	177	957	22	3592	0.00%	13.6/sec	102.1
/j_spring_se...	4500	121	73	225	6	3233	0.00%	13.6/sec	73.4
TOTAL	76500	106	42	197	2	3714	0.00%	221.8/sec	6664.4

En cuanto al throughput, el valor de este es 13307 y la grafica sobre el rendimiento de nuestro sistema, muestra que éste durante el caso de uso actual tuvo al procesador a máximo rendimiento y muchas veces el disco duro también empezaba a dar muestra de bajones.



## Crear chirp

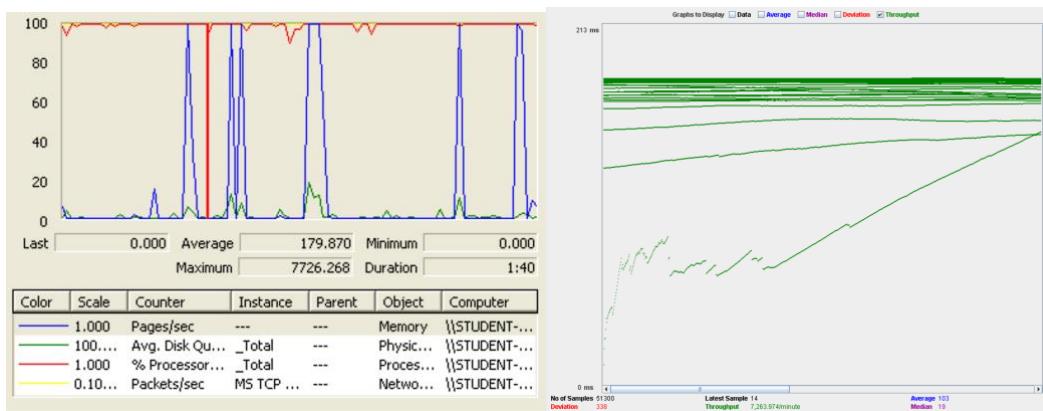
En este caso de uso iniciaremos sesión como un usuario y crearemos un chirp. Aprovechando la base de datos generada con el test de rendimiento antes en el creábamos varios miles de usuarios, vamos a comprobar como el sistema se comporta a la hora de crear un chirp cuando varios miles de usuarios diferentes tratan de realizar dicha acción y la base de datos tiene muchas entradas.

Empezamos con el valor que hemos escogido otras veces para el comienzo, 150 usuarios. El resultado del test con dicho valor arroja unos tiempos para el 90% de los usuarios demasiado alto, llegando hasta los 9 segundos. Para verificar estos números, durante la ejecución del test, he procedido a entrar en la web e intentar crear un chirp y pude comprobar como efectivamente la lentitud reinaba en la web.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5729	106	52	196	4	7012	0.00%	15.5/sec	83.7
/scripts/helpers...	1971	82	49	171	2	2080	0.00%	5.4/sec	12.4
/scripts/cookiec...	1971	107	58	212	3	7032	0.00%	5.4/sec	185.9
/scripts/bootstrap...	1971	104	63	206	4	7044	0.00%	5.4/sec	198.3
/scripts/polyfills...	1971	84	49	167	3	2378	0.00%	5.5/sec	41.2
/scripts/jquery....	1970	156	99	318	7	7292	0.00%	5.5/sec	590.1
/styles/style.css	1970	86	50	181	2	6942	0.00%	5.5/sec	8.7
/styles/bootstrap...	1968	162	110	345	5	1441	0.00%	5.6/sec	659.4
/styles/cookiec...	1968	87	50	182	2	2431	0.00%	5.6/sec	23.2
/favicon.ico	1967	79	49	157	2	7020	0.00%	5.6/sec	31.5
/images/newspl...	1967	176	110	400	7	2365	0.00%	5.7/sec	862.0
/security/login.do	1960	139	51	215	5	2554	0.00%	5.7/sec	31.6
/j_spring_secu...	1948	2289	1370	5609	17	28524	0.00%	5.8/sec	33.8
/chirp/user/crea...	3831	3989	2579	8994	10	34923	0.00%	11.1/sec	73.9
/chirp/user/list.d...	1853	2898	1941	6892	46	33205	0.00%	5.5/sec	35.3
/j_spring_secu...	1822	165	103	304	8	7136	0.00%	5.5/sec	29.7
TOTAL	36837	774	85	2189	2	34923	0.00%	99.2/sec	2763.3

Hemos bajado la cifra de 150 a 90 usuarios concurrentes, un cambio drástico que tampoco acaba de subsanar los tiempos, aunque evidentemente existe una gran mejoría. El valor throughput en este test es de 7263 y en la gráfica donde vemos el rendimiento del sistema durante el test podemos ver como el procesador está continuamente al 100% de su capacidad.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8100	21	14	34	3	1307	0.00%	19.1/sec	103.0
/scripts/helpers...	2700	14	10	28	1	1107	0.00%	6.5/sec	14.9
/scripts/cookiec...	2700	19	14	40	2	405	0.00%	6.5/sec	224.4
/scripts/bootstrap...	2700	19	16	42	2	298	0.00%	6.6/sec	239.6
/scripts/polyfills...	2700	15	11	30	2	262	0.00%	6.6/sec	49.7
/scripts/jquery....	2700	34	23	65	4	440	0.00%	6.6/sec	709.5
/styles/style.css	2700	14	10	31	1	245	0.00%	6.6/sec	10.3
/styles/bootstrap...	2700	34	24	68	4	399	0.00%	6.6/sec	785.9
/styles/cookiec...	2700	13	10	28	2	326	0.00%	6.6/sec	27.4
/favicon.ico	2700	13	10	28	2	191	0.00%	6.6/sec	37.0
/images/newspl...	2700	32	23	64	5	284	0.00%	6.6/sec	1008.8
/security/login.do	2700	18	13	33	4	246	0.00%	6.6/sec	36.7
/j_spring_secu...	2700	116	66	221	8	4283	0.00%	6.6/sec	39.8
/chirp/user/crea...	5400	459	183	1145	7	9265	0.00%	13.2/sec	88.3
/chirp/user/list.d...	2700	607	453	1263	38	8490	0.00%	6.6/sec	43.4
/j_spring_secu...	2700	28	23	52	4	705	0.00%	6.6/sec	35.7
TOTAL	51300	103	19	209	1	9265	0.00%	121.1/sec	3333.5



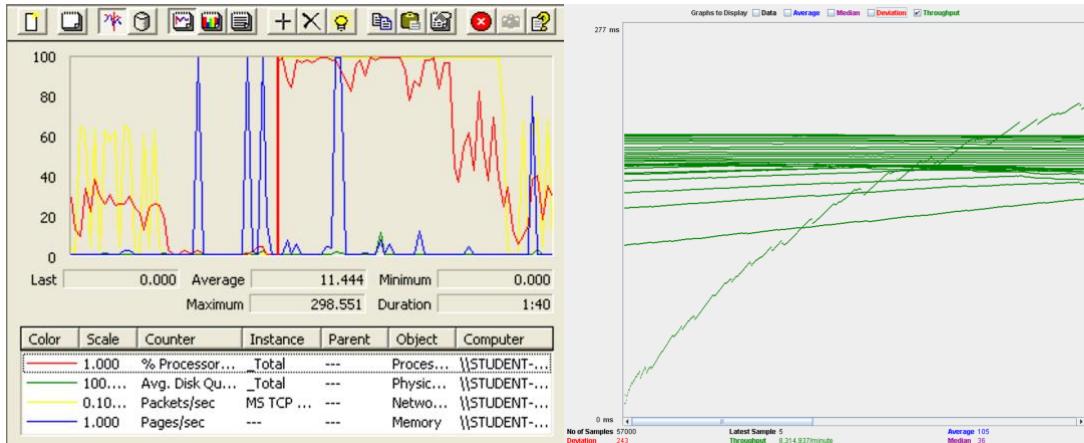
Podemos concluir entonces que el sistema cuando tenga una base de datos de un tamaño considerable va a mermar notablemente su rendimiento haciendo que muchos usuarios puedan abandonar la web. Alrededor de 90 usuarios serían los que el sistema aguantaría en estas condiciones.

## Eliminar Chirp

Vamos a realizar el test de rendimiento del caso de uso en el que iniciamos sesión como usuario y elimina un chirp. He comenzado el test con un número de 100 usuarios concurrentes y los resultados son bastante buenos, los tiempos entran dentro del rango de lo aceptable, todo ello lo podemos apreciar en la siguiente tabla:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9000	52	24	102	4	4046	0.00%	21.9/sec	120.6
/scripts/polyfills.js	3000	57	29	105	2	1460	0.00%	7.5/sec	56.5
/scripts/cookieconsent.js	3000	53	35	104	2	1033	0.00%	7.5/sec	257.7
/scripts/jquery.min.js	3000	83	57	173	4	1030	0.00%	7.5/sec	805.6
/scripts/bootstrap.min.js	3000	50	35	99	2	1327	0.00%	7.5/sec	274.3
/scripts/helpers.js	3000	36	24	74	2	646	0.00%	7.5/sec	17.2
/styles/cookieconsent.min.c...	3000	38	23	75	1	781	0.00%	7.5/sec	31.1
/styles/bootstrap.min.css	3000	75	50	158	4	1219	0.00%	7.5/sec	893.4
/styles/style.css	3000	31	20	65	2	713	0.00%	7.5/sec	11.8
/favicon.ico	3000	30	18	60	2	572	0.00%	7.5/sec	42.2
/images/newspaper-1.jpg	3000	52	26	114	6	1402	0.00%	7.5/sec	1149.1
/security/login.do	3000	39	20	69	5	1007	0.00%	7.5/sec	41.8
/j_spring_security_check	3000	149	80	352	9	3231	0.00%	7.5/sec	48.2
/chirp/administrator/list.do	6000	430	247	1013	24	4924	0.00%	15.0/sec	3412.4
/chirp/administrator/delete.do	3000	202	61	513	7	4013	0.00%	7.6/sec	107.4
/j_spring_security_logout	3000	92	56	185	6	2221	0.00%	7.6/sec	41.1
TOTAL	57000	105	36	234	1	4924	0.00%	138.6/sec	7102.0

El valor de throughput es de 8314 y el rendimiento del sistema que vemos en la gráfica es que la tarjeta de red está mayormente ocupada junto con el procesador, en contraposición del disco duro que apenas es utilizado.



## Listar seguidores/seguidos

Durante este y otros tests, se utilizó una base de datos en la que tenía varios miles de usuario. Es interesante conocer el rendimiento del sistema cuando la base de datos va estando llena, por ello durante este y varios tests se han realizado las pruebas de rendimiento con la base de datos llena.

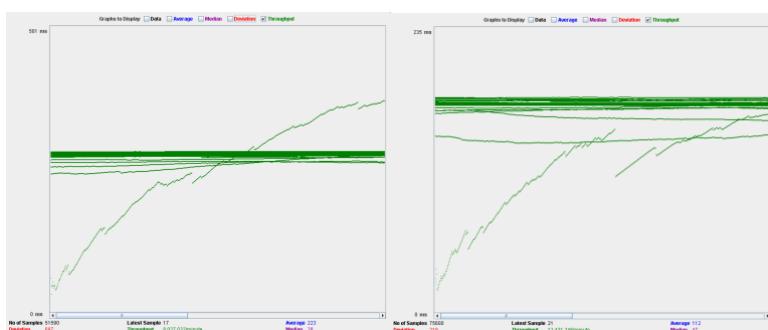
Hemos empezado este test utilizando 100 usuarios, y los resultados eran bastante malos como podemos apreciar en el Aggregate report.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8475	35	22	57	4	1738	0.00%	22.5/sec	121.5
/scripts/polyfill...	2872	25	16	46	2	1657	0.00%	7.7/sec	57.6
/scripts/cookie...	2871	32	21	56	3	1536	0.00%	7.7/sec	262.7
/scripts/bootstr...	2867	34	23	62	3	936	0.00%	7.7/sec	279.4
/styles/style.css	2866	29	16	48	1	1654	0.00%	7.7/sec	12.0
/scripts/jquery....	2866	51	37	98	4	833	0.00%	7.7/sec	821.1
/styles/cookiec...	2865	26	17	49	2	1534	0.00%	7.7/sec	31.7
/styles/bootstr...	2864	51	37	102	4	833	0.00%	7.7/sec	909.5
/scripts/helper...	2863	23	16	46	2	1654	0.00%	7.7/sec	17.5
/images/news...	2860	57	41	114	6	1668	0.00%	7.7/sec	1168.7
/favicon.ico	2848	19	12	39	2	952	0.00%	7.7/sec	42.9
/security/login....	2844	28	20	52	4	650	0.00%	7.7/sec	42.7
/j_spring_secu...	2835	697	268	1757	14	20310	0.00%	7.7/sec	46.1
/chirp/user/list...	2803	1683	1312	3342	65	12701	0.00%	7.7/sec	59.4
/actor/user/foll...	2775	1124	725	2485	47	15691	0.00%	7.6/sec	52.3
/j_spring_secu...	2775	51	37	85	6	1545	0.00%	7.6/sec	41.1
TOTAL	51149	221	28	531	1	20310	0.00%	136.0/sec	3937.2

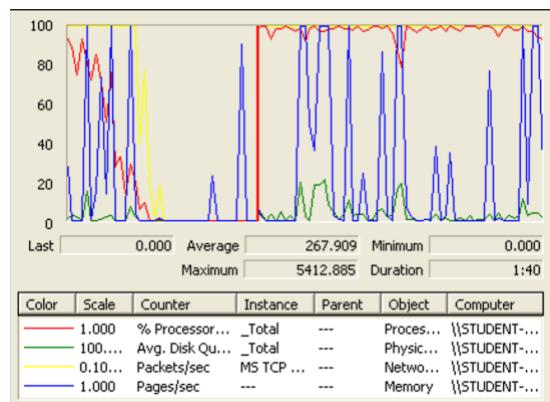
Se probó con 80 pero los resultados no mejoraban por lo que se ha decidido borrar la base de datos y comprobar como sería el rendimiento con los datos que contiene nuestro PopulateDatabase.xml.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	12600	51	32	83	3	3000	0.00%	34.5/sec	186.1
/scripts/polyfill...	4200	48	32	91	2	1559	0.00%	11.9/sec	89.3
/scripts/cookie...	4200	57	40	106	2	1748	0.00%	11.9/sec	407.2
/scripts/bootstr...	4200	60	42	112	4	1659	0.00%	11.9/sec	432.8
/styles/style.css	4200	46	32	87	2	1561	0.00%	11.9/sec	18.6
/scripts/jquery....	4200	90	69	170	5	1053	0.00%	11.9/sec	1273.5
/styles/cookiec...	4200	46	32	85	2	1299	0.00%	11.9/sec	49.2
/styles/bootstr...	4200	98	74	182	5	1799	0.00%	11.9/sec	1409.5
/scripts/helper...	4200	48	32	82	2	2105	0.00%	11.9/sec	27.2
/images/news...	4200	111	83	206	7	2176	0.00%	11.9/sec	1809.5
/favicon.ico	4200	44	23	71	2	1988	0.00%	11.9/sec	66.5
/security/login....	4200	50	30	90	5	2064	0.00%	11.9/sec	65.9
/j_spring_secu...	4200	282	173	637	11	4124	0.00%	11.9/sec	71.4
/chirp/user/list...	4200	321	878	13	4649	0.00%	11.8/sec	89.1	
/actor/user/foll...	4200	373	239	797	11	5074	0.00%	11.9/sec	81.4
/j_spring_secu...	4200	86	65	157	3	1994	0.00%	11.9/sec	64.0
TOTAL	75600	112	47	254	2	5074	0.00%	207.2/sec	5958.6

Los resultados han mejorado notablemente y con 140 usuarios se podría ofrecer una buena calidad, aunque llegando al nivel de exigencia que deseamos en nuestra aplicación. En cuanto al valor del throughput 12431 en contraposición con el de 100 usuarios (con la BD llena) que era 8027.



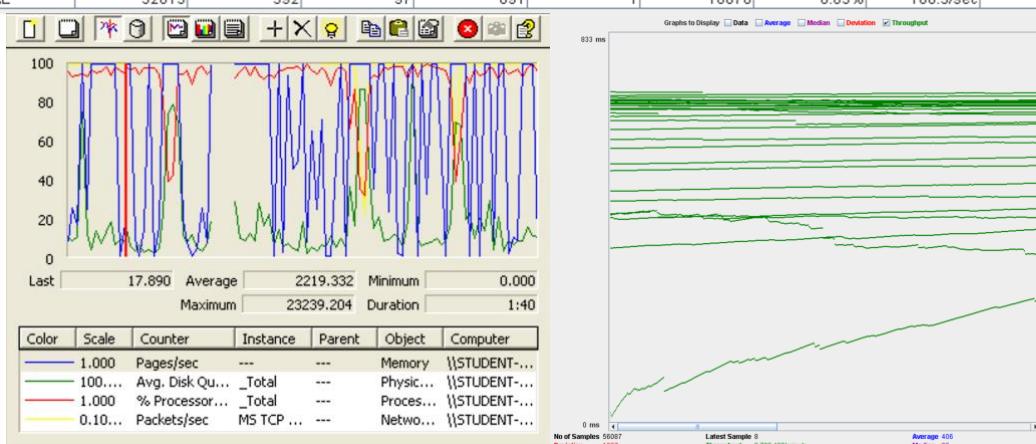
En cuanto al rendimiento del sistema, podemos observar como el procesador es de nuevo un cuello de botella importante cuyo uso se mantiene alrededor del 100% durante toda la ejecución del test.



## Seguir o dejar de seguir a alguien

Para este test se ha utilizado un CSV con los usuarios que están registrados en el sistema. Se ha comenzado el test marcando el número de usuarios en 200 para ver la respuesta del sistema para un número tan alto. El resultado ha sido que, a parte de superar con creces el tiempo que un usuario aguantaría con la página cargando, también han empezado a mostrar errores.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7717	285	66	491	4	8413	0.00%	24.7/sec	127.6
/favicon.ico	2668	262	63	557	2	5864	0.00%	8.7/sec	48.5
/images/news...	2665	261	110	535	7	5248	0.00%	8.7/sec	1325.6
/scripts/helper...	2660	143	59	329	3	2497	0.04%	8.7/sec	19.9
/scripts/polyfill...	2659	153	69	339	2	3019	0.08%	8.7/sec	65.7
/scripts/cookie...	2658	177	76	389	3	5078	0.00%	8.8/sec	301.7
/styles/bootstrap...	2656	198	108	449	5	5421	0.00%	8.8/sec	1044.4
/styles/style.css	2655	147	63	349	2	2842	0.00%	8.8/sec	13.9
/scripts/bootstrap...	2655	153	77	353	4	2924	0.00%	8.8/sec	322.8
/styles/cookie...	2652	139	54	305	2	2986	0.00%	8.8/sec	36.7
/scripts/jquery...	2632	195	60	480	4	4956	0.04%	8.9/sec	949.3
/security/login...	2622	310	48	649	2	6000	0.04%	8.9/sec	49.4
/j_spring_sec...	2601	1114	408	3170	1	18670	0.04%	8.8/sec	54.3
/actor/user/list...	2548	1083	404	3017	11	18468	0.00%	8.8/sec	65.9
/actor/user/dis...	4993	1029	433	2617	11	17252	0.00%	17.5/sec	117.1
/actor/user/foll...	2503	411	147	969	7	5413	0.28%	8.9/sec	51.2
/j_spring_sec...	2471	288	172	529	8	5549	0.04%	8.9/sec	48.1
TOTAL	52015	392	97	891	1	18670	0.03%	166.3/sec	4471.4



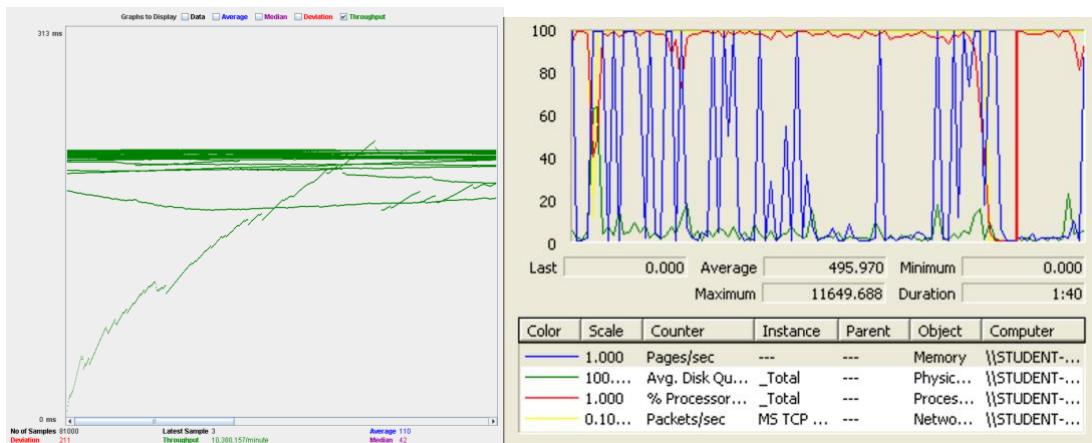
Es por ello por lo que se ha bajado a la cifra de 160, y los resultados seguían sin ser del todo buenos, pero por supuesto existe una mejora sustancial con los 200 usuarios. Por lo que se deja entrever que la cantidad de usuarios que el sistema aguantaría dando unos tiempos aceptables debe estar más o menos en 130-160.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8152	61	38	115	4	4159	0.00%	26.9/sec	145.3
/favicon.ico	2786	61	38	119	2	4253	0.00%	9.3/sec	51.9
/images/news...	2784	139	88	272	7	4350	0.00%	9.3/sec	1412.7
/scripts/helper...	2783	60	36	113	2	4211	0.00%	9.3/sec	21.2
/scripts/polyfill...	2777	65	38	127	2	1471	0.00%	9.3/sec	69.7
/scripts/cookie...	2772	72	46	140	3	1218	0.00%	9.2/sec	317.1
/styles/bootstrap...	2767	123	79	254	5	4307	0.00%	9.2/sec	1094.8
/styles/style.css	2766	60	37	119	2	4269	0.00%	9.2/sec	14.5
/scripts/bootstrap...	2763	83	52	158	4	4257	0.00%	9.2/sec	337.4
/styles/cookie...	2763	57	36	110	2	1369	0.00%	9.2/sec	38.3
/scripts/jquery...	2763	93	57	213	4	1067	0.00%	9.2/sec	989.9
/security/login...	2747	63	34	127	5	4105	0.00%	9.2/sec	51.2
/j_spring_sec...	2742	458	280	1077	15	5006	0.00%	9.2/sec	55.6
/actor/user/list...	2725	498	289	1182	11	5431	0.00%	9.3/sec	75.2
/actor/user/dis...	5366	645	446	1409	13	5976	0.00%	18.3/sec	137.5
/actor/user/foll...	2681	803	538	1815	12	5722	0.00%	9.2/sec	114.7
/j_spring_sec...	2632	117	77	241	8	4193	0.00%	9.2/sec	49.7
TOTAL	54769	209	64	534	2	5976	0.00%	180.8/sec	4914.9

Se ha realizado un último test con 135 usuarios, que nos ha dado unos tiempos muy aceptables y que podemos concluir que la cifra de 135 usuarios sería el máximo que aguantaría el servidor para este caso de uso.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	12150	49	27	74	3	2787	0.00%	26.0/sec	139.9
/favicon.ico	4050	42	28	75	2	1985	0.00%	8.9/sec	49.9
/images/news...	4050	97	65	195	6	1989	0.00%	8.9/sec	1359.1
/scripts/helper...	4050	48	28	81	2	1988	0.00%	8.9/sec	20.4
/scripts/polyfill...	4050	41	27	77	2	1914	0.00%	8.9/sec	67.3
/scripts/cookie...	4050	48	32	84	3	2452	0.00%	8.9/sec	306.6
/styles/bootstrap...	4050	81	55	155	5	2620	0.00%	8.9/sec	1060.0
/styles/style.css	4050	41	26	73	2	1999	0.00%	8.9/sec	14.0
/scripts/bootstrap...	4050	53	34	95	3	1899	0.00%	8.9/sec	325.9
/styles/jquery...	4050	39	25	67	2	2334	0.00%	8.9/sec	37.0
/scripts/jquery...	4050	74	38	139	4	1995	0.00%	8.9/sec	958.6
/security/login...	4050	44	23	67	5	1980	0.00%	8.9/sec	49.6
/j_spring_sec...	4050	194	117	424	9	3082	0.00%	8.9/sec	53.7
/actor/user/list...	4050	202	120	447	11	3039	0.00%	8.9/sec	72.5
/actor/user/dis...	8100	301	212	649	12	2969	0.00%	17.8/sec	133.2
/actor/user/foll...	4050	366	241	826	8	3159	0.00%	8.9/sec	106.0
/j_spring_sec...	4050	69	50	129	7	2112	0.00%	8.9/sec	48.3
TOTAL	81000	110	42	264	2	3159	0.00%	173.0/sec	4655.8

Por otro lado, el throughput arroja un valor de 10380 y en el rendimiento del equipo vemos como el procesador sigue estando a máximos de su capacidad junto con la tarjeta de red.



## Listar y desplegar carpetas

Se ha comenzado el primer test poniendo 150 usuarios ya que este test no tiene una enorme exigencia que tienen otros.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	76	36	142	3	4508	0.00%	23.7/sec	127.9
/styles/style.css	4500	64	36	133	2	2171	0.00%	8.1/sec	12.6
/scripts/jquery...	4500	110	65	245	4	2408	0.00%	8.1/sec	866.3
/styles/bootstrap...	4500	113	65	250	5	2660	0.00%	8.1/sec	964.7
/styles/cookie...	4500	64	37	131	2	2783	0.00%	8.2/sec	33.9
/scripts/polyfill...	4500	60	36	122	2	2152	0.00%	8.2/sec	62.1
/scripts/bootstrap...	4500	74	45	144	2	2883	0.00%	8.3/sec	302.6
/scripts/helper...	4500	63	36	127	2	2887	0.00%	8.3/sec	19.0
/scripts/cookie...	4500	71	43	149	3	2396	0.00%	8.4/sec	287.2
/images/news...	4500	117	70	255	5	2603	0.00%	8.4/sec	1279.8
/favicon.ico	4500	111	19	122	2	4425	0.00%	8.4/sec	47.2
/security/login...	4500	120	31	152	5	4205	0.00%	8.5/sec	47.0
/j_spring_sec...	4500	252	132	550	9	4040	0.00%	8.5/sec	51.2
/folder/actor/rli...	4500	225	95	504	9	3729	0.00%	8.6/sec	64.8
/folder/actor/dl...	4500	278	124	642	8	4157	0.00%	8.6/sec	51.0
/j_spring_sec...	4500	117	81	240	6	2018	0.00%	8.6/sec	46.6
TOTAL	81000	115	49	234	2	4508	0.00%	142.4/sec	4087.0

Como resultado vemos que los números que se reflejan en los tiempos son muy buenos, dentro de los límites aceptables y un indicador de que el sistema aún puede aguantar un poco más por lo que vamos a hacer el test con 160 usuarios para llegar al límite de su capacidad.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	14400	63	34	120	3	4556	0.00%	39.3/sec	211.7
/styles/style.css	4800	61	34	128	2	1569	0.00%	13.5/sec	21.1
/scripts/jquery...	4800	110	72	236	4	2008	0.00%	13.5/sec	1444.8
/styles/bootstrap...	4800	110	71	231	5	2013	0.00%	13.5/sec	1599.2
/styles/cookie...	4800	65	35	136	2	1812	0.00%	13.5/sec	55.8
/scripts/polyfill...	4800	69	35	138	2	2007	0.00%	13.5/sec	101.6
/scripts/bootstrap...	4800	76	43	153	2	1681	0.00%	13.5/sec	492.9
/scripts/helper...	4800	61	34	125	2	1999	0.00%	13.5/sec	30.9
/scripts/cookie...	4800	72	43	144	3	1999	0.00%	13.5/sec	464.4
/images/news...	4800	116	78	246	5	2025	0.00%	13.5/sec	2060.9
/favicon.ico	4800	55	24	110	2	1984	0.00%	13.5/sec	75.8
/security/login...	4800	60	31	121	4	1992	0.00%	13.5/sec	75.2
/j_spring_sec...	4800	196	114	454	9	3236	0.00%	13.6/sec	81.5
/folder/actor/rli...	4800	163	79	341	9	3361	0.00%	13.5/sec	102.3
/folder/actor/dl...	4800	228	117	518	8	3545	0.00%	13.5/sec	80.4
/j_spring_sec...	4800	117	76	254	2	3006	0.00%	13.6/sec	73.2
TOTAL	86400	97	47	213	2	4556	0.00%	235.6/sec	6765.5

Vemos que el resultado se asemeja en parte al de 150, por lo que probaremos una vez más a subir el número de usuario otro diez hasta los 170 concurrentes.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15300	125	59	277	4	3635	0.00%	36.9/sec	199.1
/styles/style.css	5100	122	59	285	2	3581	0.00%	12.6/sec	19.8
/scripts/jquery...	5100	181	100	421	5	2927	0.00%	12.6/sec	1352.6
/styles/bootstrap...	5100	194	106	454	5	3719	0.00%	12.6/sec	1497.2
/styles/cookie...	5100	129	63	315	2	3604	0.00%	12.6/sec	52.3
/scripts/polyfill...	5100	127	62	318	2	3581	0.00%	12.6/sec	95.1
/scripts/bootstrap...	5100	128	69	304	3	2710	0.00%	12.6/sec	461.3
/scripts/helper...	5100	116	58	293	2	2490	0.00%	12.6/sec	28.9
/scripts/cookie...	5100	135	73	330	2	3580	0.00%	12.6/sec	434.3
/images/news...	5100	206	115	492	6	3602	0.00%	12.7/sec	1927.5
/favicon.ico	5100	141	49	329	2	3598	0.00%	12.7/sec	70.9
/security/login...	5100	151	60	351	4	3346	0.00%	12.7/sec	70.3
/j_spring_sec...	5100	550	330	1252	9	7053	0.00%	12.7/sec	76.2
/folder/actor/rli...	5100	449	230	1123	9	8780	0.00%	12.7/sec	95.6
/folder/actor/dl...	5100	535	286	1329	8	6897	0.00%	12.7/sec	75.2
/j_spring_sec...	5100	273	170	591	6	4541	0.00%	12.7/sec	68.4
TOTAL	91800	212	84	508	2	8780	0.00%	221.6/sec	6361.8

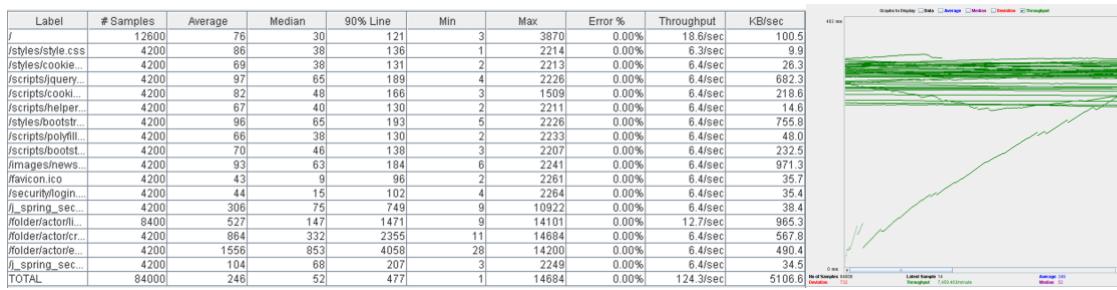
Como vemos, el resultado de este test ya se va fuera de los límites de lo aceptable según nuestros criterios, por ende, podemos concluir que el número de usuarios que aguanta el sistema para este caso de uso estará alrededor de los 160 usuarios.



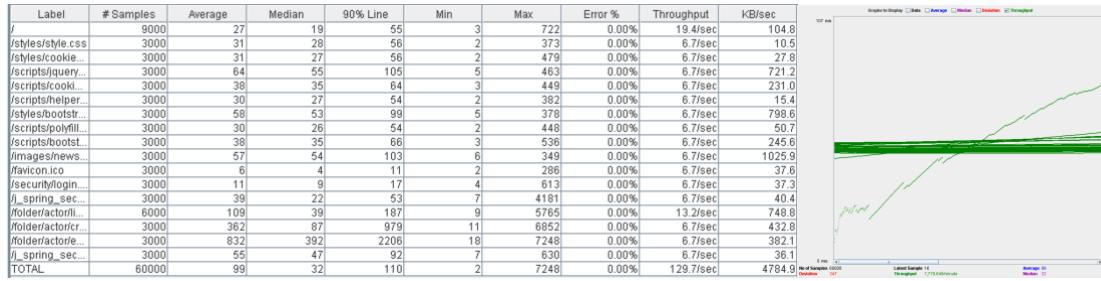
## Crear una carpeta

En este caso de uso vamos a utilizar dos CSV, uno en el que están los usuarios registrados en el sistema (son seis usuarios) y otro con los nombres de varios miles de carpetas para que cada uno cree una diferente y así el test de crear carpetas sea lo más realista posible.

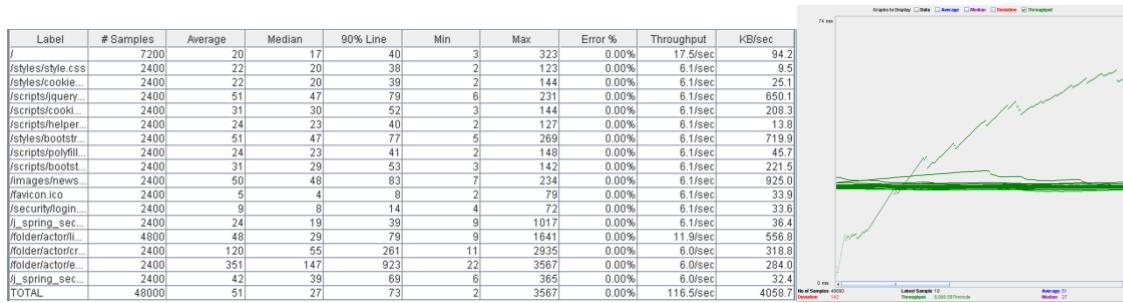
Se ha realizado el primer test con 140 usuarios concurrentes, lo cual da un resultado con unos tiempos muy altos fuera del rango de tiempo que consideramos aceptable.



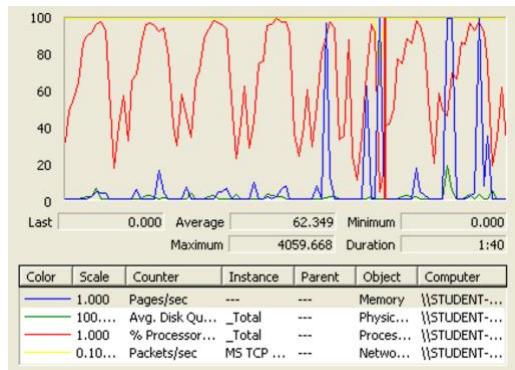
Bajamos hasta los 100 usuarios y aún así los tiempos siguen siendo mayores al segundo que hemos puesto como límite por lo que nuestro sistema no respondería como debería.



Probamos una vez más, esta vez con un cambio más pequeño, yendo hasta los 80 usuarios donde ya vemos como los tiempos entran rozando el rango que consideramos aceptable. Por ello, podemos decir que el sistema para este caso de uso podrá tener 80 usuarios de forma concurrente. En cuanto al throughput el valor de este será de 6990.



Por último, el rendimiento del sistema para este último test muestra como el procesador va dando altibajos, pero mayormente está en uso y como se puede ver la tarjeta de red está al 100% de su capacidad.

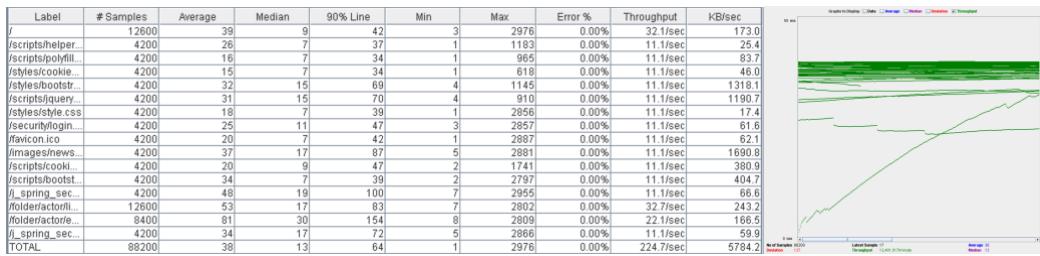


## Editar/borrar una carpeta

En este caso de uso vamos a hacer por un lado editar una carpeta y por otro borrarla. Primeramente, vamos a comenzar a editar una carpeta con 150 usuarios concurrentes lo cual, aunque da tiempos bastante buenos, la aplicación da fallos por lo que el sistema no lo aguanta.



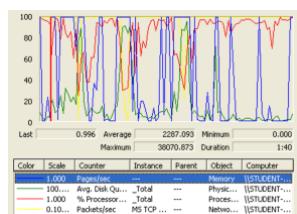
Probamos ahora con 140 usuarios, dando unos tiempos mejores y sin fallos. Por ende, podemos concluir que el sistema aguanta 140 usuarios a la vez y que da un throughput de 13481.



Pasamos ahora a borrar una carpeta, empezando el test con 180 usuarios concurrente. El resultado vemos que es bastante bueno pues no da ningún fallo y los tiempos para el 90% de los usuarios es muy pequeño, por lo que podemos concluir de que para el caso de uso de borrar una carpeta el sistema aguanta 180 usuarios y da un throughput de 14107.



En cuanto al rendimiento del sistema para borrar una carpeta, vemos como el disco duro se mantiene bastante ocupado junto con el procesador que también da un uso alto.

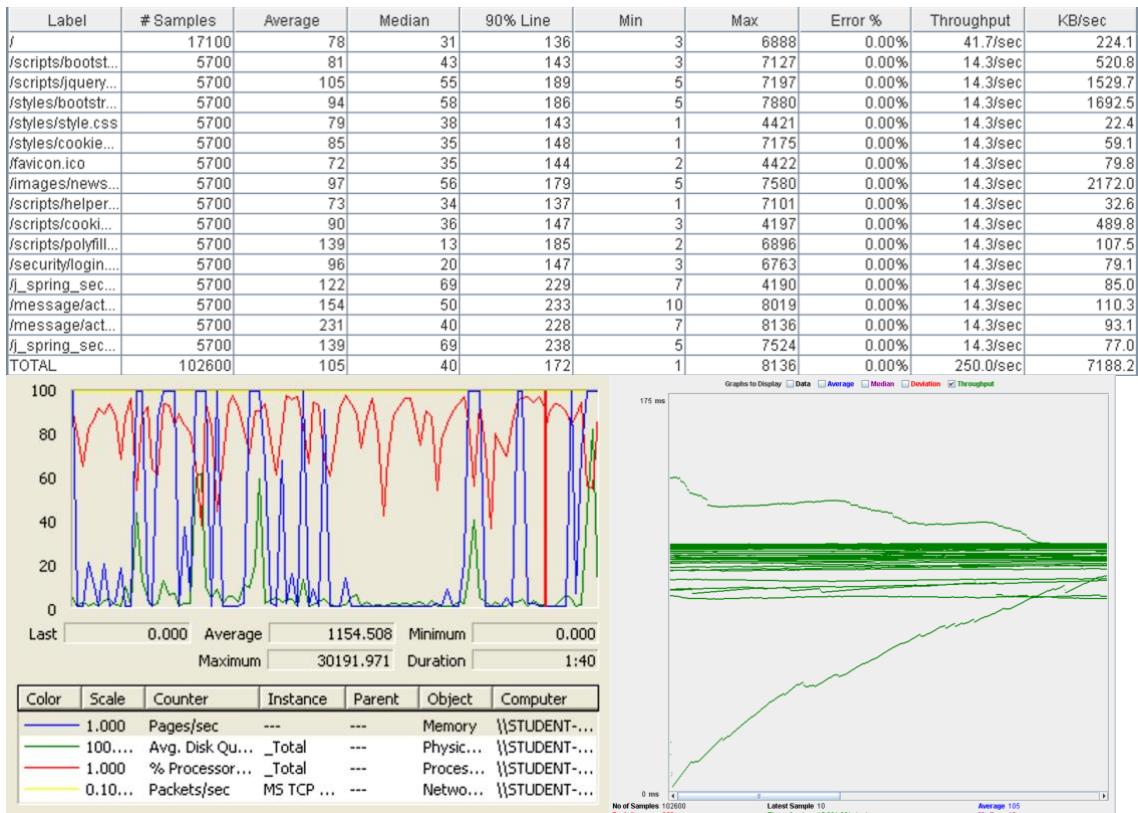


## Listar y desplegar mensaje

Este caso de uso consiste en iniciar sesión como usuario, listar los mensajes y desplegar uno de ellos. Para comenzar, el primer test comienza con 250 usuarios concurrentes y vemos que, aunque los tiempos son aceptables, comienza a ver errores. Vemos en la gráfica del rendimiento del sistema que cuando hay más páginas suba en gran medida el uso del disco duro y baje el uso del procesador, en otros momentos el uso del procesador se pone cercano al 100% y baja el uso del disco duro.

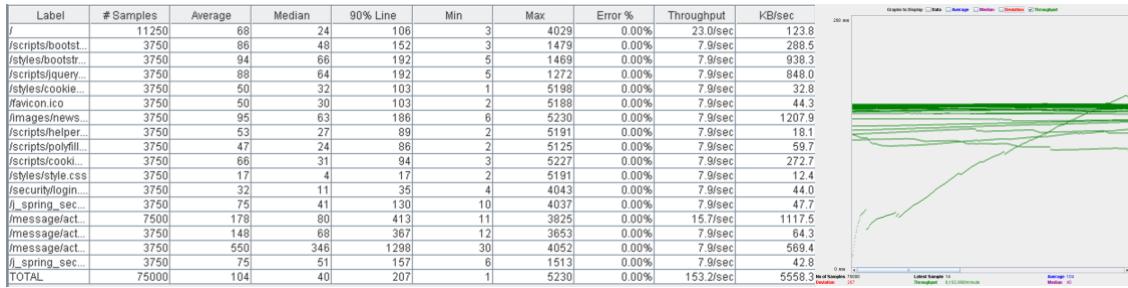


Bajamos hasta los 180 usuarios y vemos que esta vez los tiempos son menores y no se aprecian errores, por ende, podemos concluir que son 180 los usuarios que el sistema puede aguantar para este caso de uso. Por último, el throughput para este test será de 15001 y la grafica de rendimiento donde vemos que el procesador está mayormente en uso y cuando se dan muchas páginas por segundo sube el uso del disco duro.

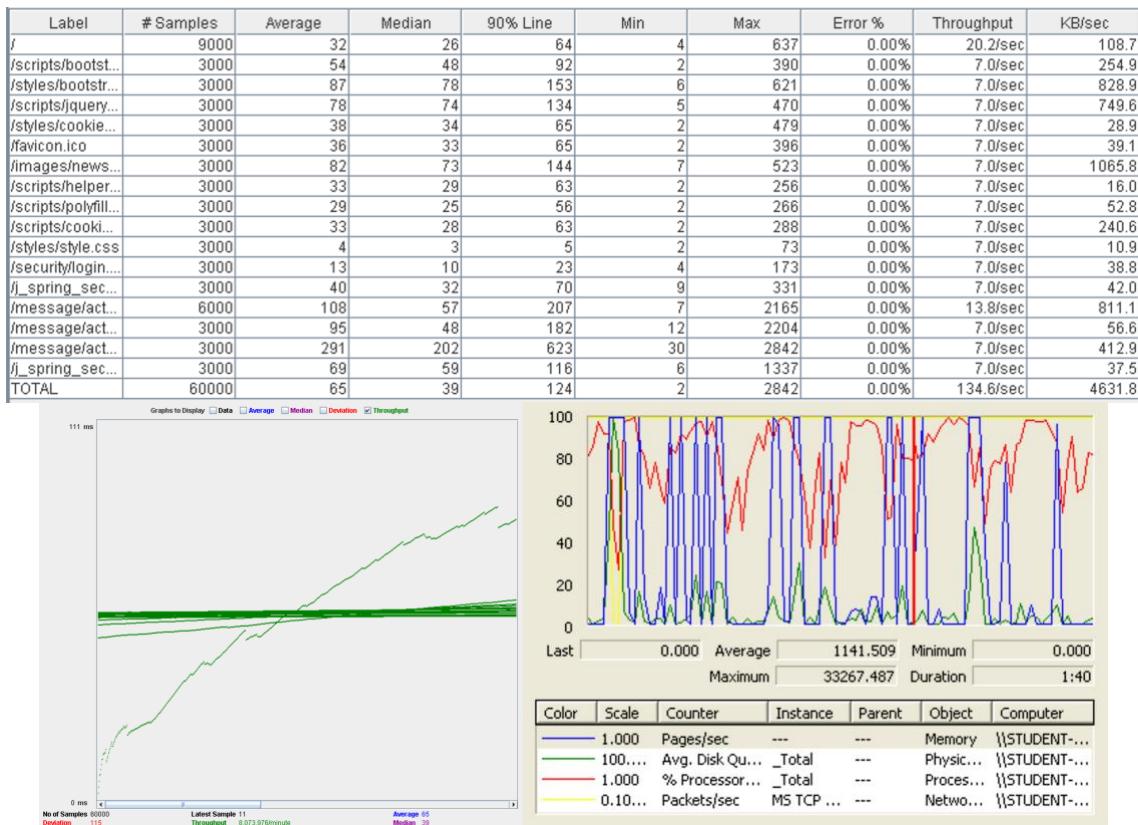


## Enviar/borrar mensaje

Para este caso de uso vamos a hacer por un lado enviar un mensaje y por otro lado borrarlo. Empezamos por el primero realizando un test con 125 usuarios, lo cual da unos tiempos que salen del rango de lo aceptable por lo que vamos a bajar un poco la cantidad de usuarios.

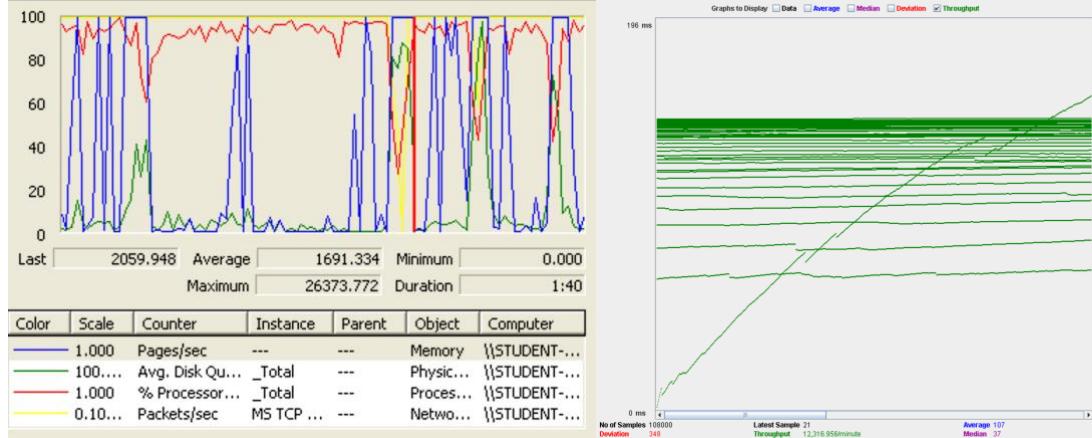


Hemos bajado hasta los 100 usuarios y esta vez si nos da unos tiempos sumamente aceptables. Por otro lado, el throughput es de 8073 y la gráfica del rendimiento del sistema muestra como el procesador tiene bastante uso y la tarjeta de red está a su máximo uso.



Por último, vamos a realizar el caso en el que borramos un mensaje. Empezamos con 180 usuarios concurrentes y el resultado que nos da es bastante bueno ya que los tiempos para el 90% de los usuarios son muy pequeños y no existe ningún tipo de error. Es por ello por lo que podemos concluir que el sistema para el caso de uso en el que borramos un mensaje va a poder aguantar 180 usuarios. El valor del throughput es de 12316. Por último, en cuanto a la gráfica del rendimiento del sistema podemos ver como el procesador está mayormente al 100% de su capacidad.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	16200	95	30	112	3	7340	0.00%	30.8/sec	165.5
/scripts/polyfill...	5400	64	27	98	2	5389	0.00%	10.5/sec	79.4
/scripts/cookie...	5400	70	33	111	3	5565	0.00%	10.5/sec	361.6
/scripts/bootstrap...	5400	71	35	113	4	6416	0.00%	10.5/sec	384.3
/scripts/helper...	5400	54	27	100	2	5209	0.00%	10.5/sec	24.1
/styles/cookie...	5400	57	27	96	2	5568	0.00%	10.5/sec	43.6
/styles/style.css	5400	64	28	100	2	5415	0.00%	10.5/sec	16.5
/scripts/jquery...	5400	94	46	160	5	5438	0.00%	10.5/sec	1128.8
/styles/bootstrap...	5400	98	50	166	5	5530	0.00%	10.5/sec	1249.9
/favicon.ico	5400	65	31	109	2	5013	0.00%	10.5/sec	59.0
/images/news...	5400	170	31	208	5	5191	0.00%	10.5/sec	1604.8
/security/login...	5400	89	21	142	4	4270	0.00%	10.6/sec	58.9
/j_spring_sec...	5400	163	82	314	6	3501	0.00%	10.6/sec	63.5
/message/act...	10800	180	56	371	9	5250	0.00%	21.3/sec	150.4
/message/act...	10800	157	47	344	7	5129	0.00%	21.5/sec	280.0
/j_spring_sec...	5400	121	62	204	7	4635	0.00%	10.8/sec	58.4
TOTAL	108000	107	37	177	2	7340	0.00%	205.3/sec	5578.2



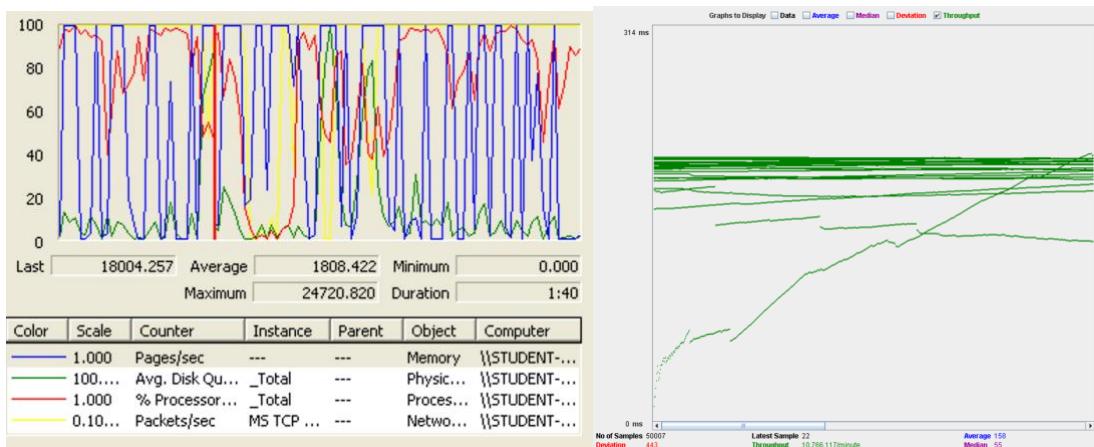
## Mover mensaje de carpeta

Realizamos el test de rendimiento del caso de uso de mover un mensaje de carpeta empezando por 190 usuarios concurrentes. El resultado, mostrado en la siguiente imagen, aunque muestra unos tiempos más o menos correctos, da errores, por lo que tenemos que bajar la cantidad de usuarios.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5863	152	44	261	3	4538	0.02%	18.2/sec	97.6
/favicon.ico	2039	296	60	320	2	4470	0.29%	6.4/sec	35.6
/images/news...	1979	222	100	383	5	4424	0.45%	6.2/sec	943.3
/scripts/helper...	1973	153	55	306	1	4534	0.25%	6.3/sec	14.4
/scripts/cookie...	1966	191	72	309	3	4475	0.10%	6.3/sec	216.7
/scripts/bootstrap...	1955	165	67	314	3	4378	0.05%	6.5/sec	238.4
/scripts/polyfill...	1955	146	49	287	1	3978	0.10%	6.6/sec	49.6
/styles/cookie...	1955	124	53	267	2	3969	0.15%	6.6/sec	27.5
/styles/bootstrap...	1955	158	76	302	1	4401	0.15%	6.6/sec	784.5
/scripts/jquery...	1953	149	78	290	1	4498	0.15%	6.7/sec	719.1
/styles/style.css	1950	101	9	167	2	3583	0.00%	6.8/sec	10.7
/security/login...	1944	109	17	182	4	2514	0.00%	6.9/sec	38.4
/j_spring_sec...	1943	121	63	286	10	4564	0.00%	7.0/sec	41.6
/message/act...	3865	303	77	593	9	5673	0.00%	14.3/sec	101.3
/message/act...	3859	435	236	1024	14	5287	0.00%	14.5/sec	96.6
/j_spring_sec...	1913	284	142	392	7	4600	0.00%	7.5/sec	40.7
TOTAL	39067	207	65	387	1	5673	0.09%	121.0/sec	3216.4

Bajamos pues a los 170 usuarios y ya vemos buenos tiempos para el 90% de usuarios y ningún error, por ende, dado esto podemos concluir que el sistema para el caso de uso de mover una carpeta aguanta 170 usuarios concurrentes y da un throughput de 10766. La gráfica del rendimiento de sistema muestra que existe un uso bastante grande del procesador y un poco menos del disco duro.

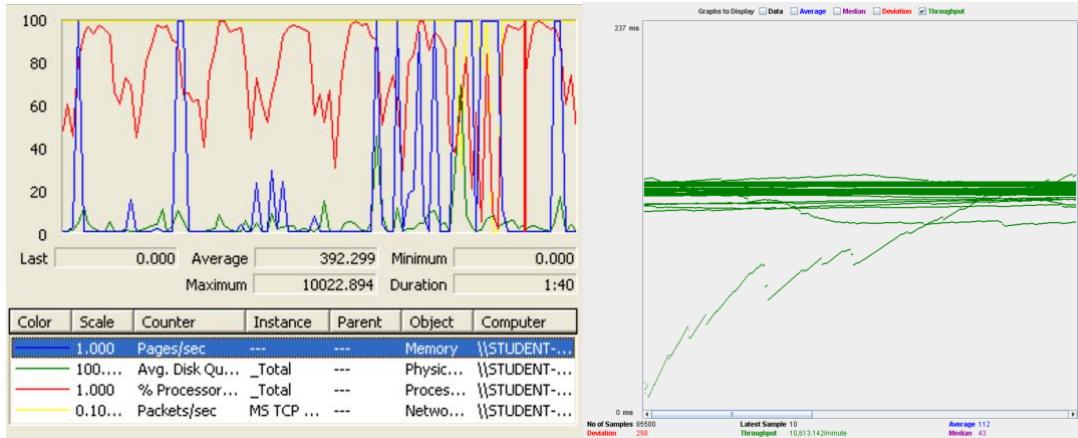
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7454	109	36	176	3	6500	0.00%	26.9/sec	144.4
/favicon.ico	2577	73	41	180	2	1770	0.00%	9.4/sec	52.6
/images/news...	2559	109	71	251	6	1763	0.00%	9.3/sec	1421.6
/scripts/helper...	2532	83	47	199	2	3264	0.00%	9.3/sec	21.2
/scripts/cookie...	2530	105	53	199	3	6470	0.00%	9.4/sec	321.3
/scripts/bootstrap...	2530	138	53	228	3	6498	0.00%	9.3/sec	340.9
/scripts/polyfill...	2530	193	43	236	2	6506	0.00%	9.3/sec	70.3
/styles/cookie...	2530	91	45	172	2	6495	0.00%	9.3/sec	38.6
/styles/bootstrap...	2530	107	68	237	5	6013	0.00%	9.4/sec	1117.9
/scripts/jquery...	2530	105	65	245	4	2991	0.00%	9.4/sec	1009.5
/styles/style.css	2516	153	9	163	2	6165	0.00%	9.4/sec	14.7
/security/login...	2497	365	21	542	4	6500	0.00%	9.3/sec	51.9
/j_spring_sec...	2470	243	70	290	10	7011	0.00%	9.3/sec	55.8
/message/act...	4900	201	70	476	10	6757	0.00%	18.8/sec	145.7
/message/act...	4914	286	133	751	14	6373	0.00%	18.9/sec	126.0
/j_spring_sec...	2408	117	79	243	6	1494	0.00%	9.5/sec	51.1
TOTAL	50007	158	55	282	2	7011	0.00%	179.4/sec	4831.7



## Difundir mensaje

En este caso de uso iniciamos sesión como un administrador y difundimos un mensaje entre todos los usuarios. Empezamos el test con 150 usuarios, pero el tiempo que sale es mayor del que tomamos como un valor aceptable, por ende vamos a bajar un poco el número de usuarios.

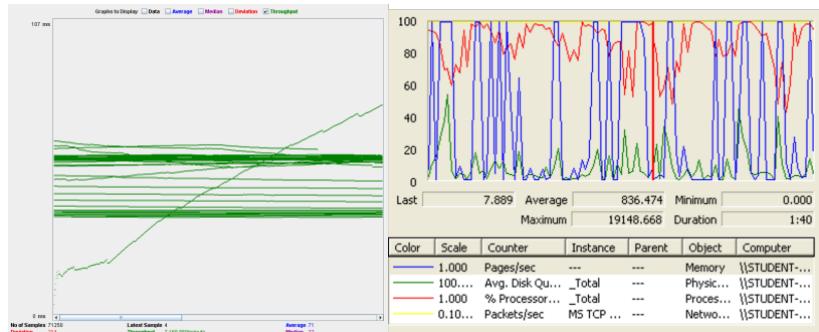
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	18000	71	26	116	3	3416	0.00%	37.2/sec	211.4
/styles/style.css	4500	59	39	133	2	1006	0.00%	9.6/sec	15.0
/favicon.ico	4500	60	40	125	2	3735	0.00%	9.6/sec	53.5
/images/news...	4500	107	83	216	5	2768	0.00%	9.6/sec	1457.5
/scripts/polyfill...	4500	60	41	127	2	2902	0.00%	9.6/sec	72.2
/scripts/cockl...	4500	71	49	156	3	2711	0.00%	9.6/sec	329.3
/scripts/bootst...	4500	74	52	158	3	2711	0.00%	9.6/sec	350.1
/scripts/helper...	4500	100	73	203	5	3192	0.00%	9.6/sec	1138.5
/scripts/helper...	4500	57	37	123	2	2709	0.00%	9.6/sec	22.0
/styles/cookie...	4500	52	33	110	1	2752	0.00%	9.6/sec	39.8
/scripts/jquery...	4500	100	21	105	4	3811	0.00%	9.6/sec	1030.5
/security/login...	4500	105	16	83	4	3766	0.00%	9.6/sec	53.3
/j_spring_sec...	4500	165	68	256	10	3760	0.00%	9.6/sec	61.4
/message/ad...	4500	159	45	455	7	4867	0.00%	9.6/sec	74.1
/message/ad...	4500	585	359	1315	26	6734	0.00%	9.6/sec	64.0
/j_spring_sec...	4500	97	68	193	6	3248	0.00%	9.6/sec	51.8
TOTAL	85500	112	43	207	1	6734	0.00%	176.9/sec	4883.9



Hemos bajado hasta los 125 usuarios y como vemos da unos valores perfectos pues los tiempos para el 90% de los usuarios encajan en el rango que consideramos aceptable que es que de menos de 1000ms, por ende podemos concluir que 125 usuarios concurrente será la cantidad que el servidor aguante para este caso de uso.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15000	49	21	72	3	3509	0.00%	25.1/sec	142.6
/styles/style.css	3750	42	32	79	2	924	0.00%	6.5/sec	10.1
/favicon.ico	3750	42	33	82	2	350	0.00%	6.4/sec	36.1
/images/news...	3750	69	58	130	6	757	0.00%	6.4/sec	982.3
/scripts/polyfill...	3750	42	35	80	1	1061	0.00%	6.4/sec	48.6
/scripts/cockl...	3750	45	38	86	3	311	0.00%	6.4/sec	221.3
/scripts/bootst...	3750	46	39	89	2	318	0.00%	6.4/sec	235.1
/scripts/helper...	3750	60	48	118	5	606	0.00%	6.4/sec	764.7
/scripts/helper...	3750	38	29	79	2	440	0.00%	6.4/sec	14.7
/styles/cookie...	3750	33	25	67	2	3248	0.00%	6.4/sec	26.7
/scripts/jquery...	3750	22	11	36	4	2309	0.00%	6.4/sec	690.7
/security/login...	3750	81	11	32	4	3531	0.00%	6.4/sec	35.7
/j_spring_sec...	3750	64	37	115	9	4393	0.00%	6.4/sec	41.1
/message/ad...	3750	82	23	161	6	4425	0.00%	6.4/sec	49.5
/message/ad...	3750	395	239	788	28	5035	0.00%	6.4/sec	43.1
/j_spring_sec...	3750	83	57	127	7	3406	0.00%	6.4/sec	34.6
TOTAL	71250	71	32	113	1	5035	0.00%	119.3/sec	3294.8

Por último, el throughput da un valor 7159 y como podemos apreciar en la gráfica de rendimiento del sistema, el procesador tiene altas cotas de uso al igual que la tarjeta de red.



## Crear Follow-up

En este caso de uso, un usuario puede crear un follow-up. Para ello, sigue la siguiente secuencia de pasos:

Todos los periódicos > (Escoger uno) > (Escoger artículo) > Follow-up

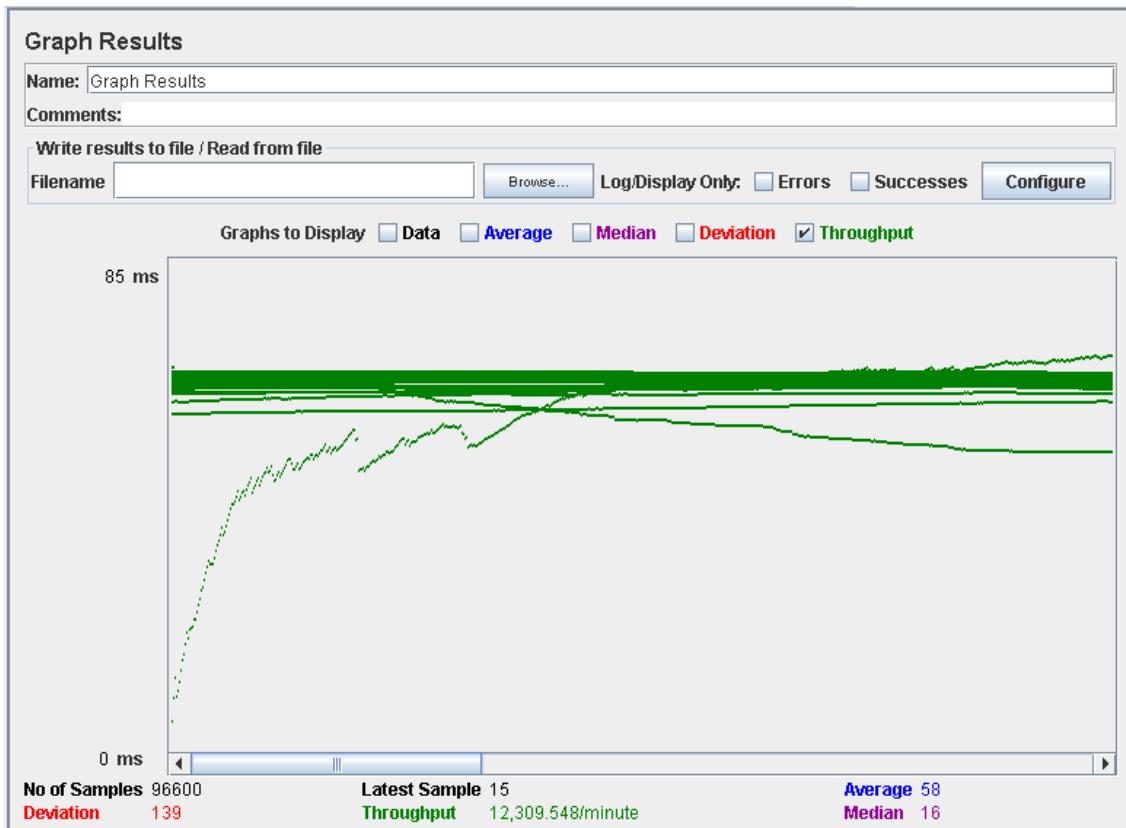
El límite de este caso de uso se sitúa en 140 usuarios concurrentes ejecutando 30 veces la misma acción.

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 checked="" type="checkbox"/> Configure					
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	12600	20	13	41	2	727	0.00%	25.4/sec	137.0	
/styles/style...	4200	18	11	41	1	434	0.00%	8.7/sec	13.7	
/scripts/boo...	4200	27	17	62	2	455	0.00%	8.7/sec	318.3	
/scripts/poly...	4200	18	12	39	1	936	0.00%	8.7/sec	65.7	
/scripts/iqu...	4200	48	28	108	4	2143	0.00%	8.7/sec	935.3	
/scripts/coo...	4200	27	16	56	2	966	0.00%	8.7/sec	299.5	
/scripts/hel...	4200	18	11	40	1	907	0.00%	8.7/sec	19.9	
/styles/boot...	4200	47	29	109	5	671	0.00%	8.7/sec	1035.2	
/styles/cook...	4200	17	11	37	1	484	0.00%	8.7/sec	36.1	
/images/he...	4200	56	33	131	6	884	0.00%	8.7/sec	1329.1	
/favicon.ico	4200	15	8	33	1	402	0.00%	8.7/sec	48.8	
/security/log...	4200	20	13	41	3	459	0.00%	8.7/sec	48.4	
/j_spring_s...	4200	117	50	238	7	3284	0.00%	8.7/sec	52.4	
/newspaper...	4200	111	54	243	11	3124	0.00%	8.7/sec	77.3	
/newspaper...	4200	87	41	211	6	2723	0.00%	8.7/sec	71.1	
/wp-content...	4200	622	604	659	522	1832	0.00%	8.7/sec	991.9	
/article/disp...	4200	118	51	265	9	3134	0.00%	8.7/sec	66.0	
/followUp/u...	4200	137	55	308	7	3500	0.00%	8.7/sec	66.1	
/followUp/u...	4200	315	150	727	14	6007	0.00%	8.7/sec	60.6	
/followUp/dl...	4200	106	44	207	6	3372	0.00%	8.7/sec	58.9	
/j_spring_s...	4200	37	26	76	5	885	0.00%	8.7/sec	47.1	
TOTAL	96600	88	24	209	1	6007	0.00%	194.8/sec	5614.1	

Como podemos observar, a la hora de guardar el follow-up y desplegarlo, 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 12309 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 150 usuarios ejecutando nuevamente 30 acciones cada uno.

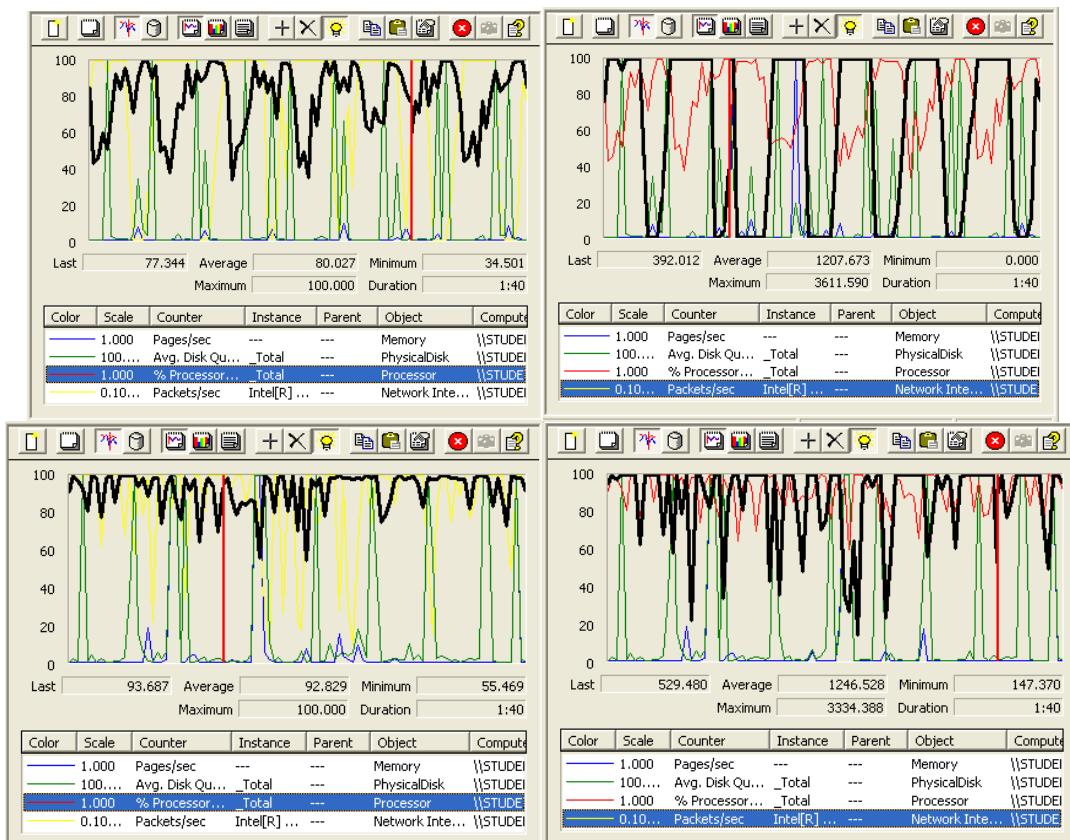
Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Filename				Browse...	Log/Display Only:		<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	Configure	
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	12709	37	15	55	3	4122	0.00%	26.1/sec	140.8	
/styles/style...	4297	35	13	52	1	3232	0.00%	8.9/sec	13.9	
/scripts/boo...	4297	41	19	74	2	2813	0.00%	8.9/sec	323.0	
/scripts/poly...	4297	33	13	51	1	3635	0.00%	8.9/sec	66.7	
/scripts/jqu...	4297	60	30	119	4	2668	0.00%	8.9/sec	949.7	
/scripts/coo...	4297	39	18	71	2	2446	0.00%	8.9/sec	304.1	
/scripts/hel...	4297	28	13	53	1	2004	0.00%	8.9/sec	20.3	
/styles/boot...	4297	61	30	121	4	2586	0.00%	8.9/sec	1052.4	
/styles/cook...	4297	29	12	51	1	1870	0.00%	8.9/sec	36.7	
/images/ne...	4297	67	33	133	6	2629	0.00%	8.9/sec	1351.5	
/favicon.ico	4297	27	10	43	1	2302	0.00%	8.9/sec	49.8	
/security/log...	4282	37	15	59	3	1945	0.00%	8.9/sec	49.3	
/j_spring_s...	4265	205	63	348	7	7998	0.00%	8.9/sec	53.3	
/newspaper...	4258	190	80	387	12	5586	0.00%	8.9/sec	79.1	
/newspaper...	4241	152	66	325	7	4685	0.00%	8.9/sec	72.8	
/wp-content...	4240	682	810	782	66	1969	0.19%	9.0/sec	1017.9	
/article/disp...	4214	177	73	357	9	9972	0.00%	9.0/sec	67.8	
/followUp/u...	4204	167	69	327	8	7257	0.00%	9.0/sec	68.3	
/followUp/u...	4184	387	182	838	19	13113	0.00%	9.0/sec	62.6	
/followUp/di...	4170	141	48	242	6	8351	0.00%	9.0/sec	60.9	
/j_spring_s...	4147	48	30	95	5	1411	0.00%	9.0/sec	48.6	
TOTAL	97884	117	28	289	1	13113	0.01%	200.8/sec	5817.6	

En este caso vemos como a la hora de guardar el follow-up 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 12049.



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 Follow-ups

En este caso de uso, un usuario puede listar sus follow-ups. Para ello, sigue la siguiente secuencia de pasos después de autenticarse:

Usuario > Mis continuaciones a artículo.

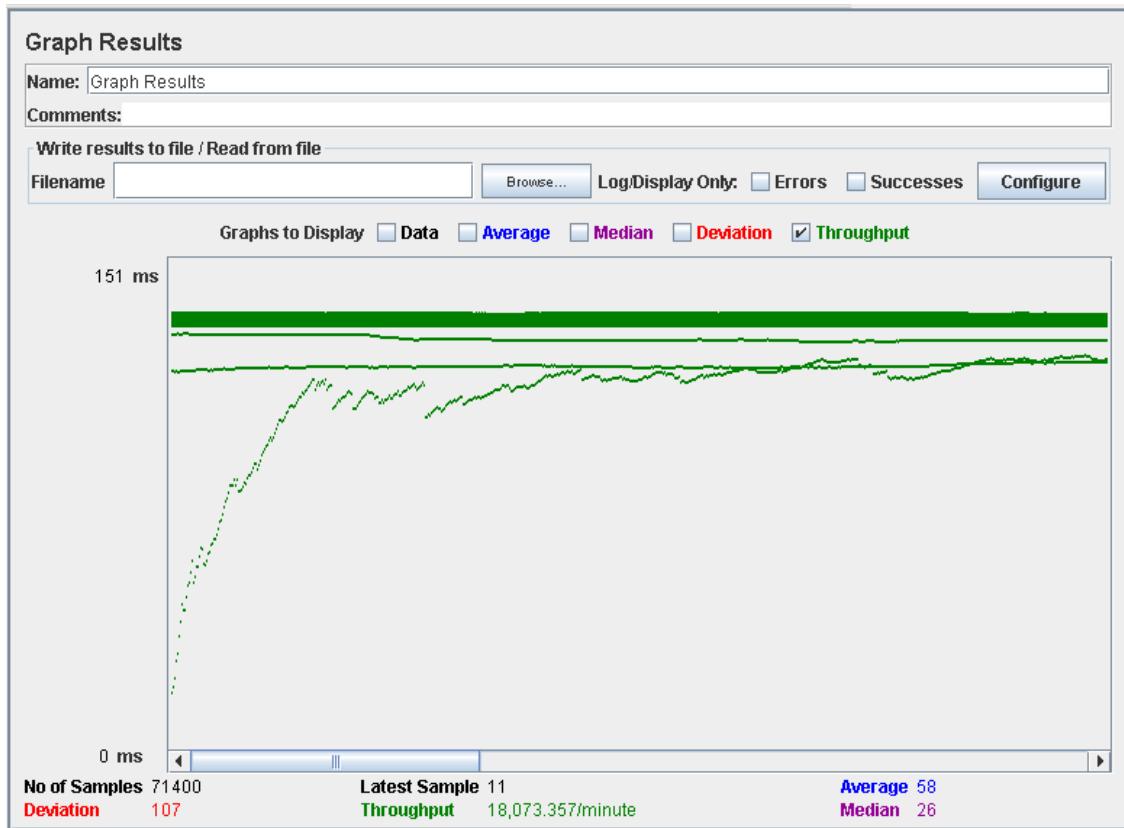
El límite de este caso de uso se sitúa en 140 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Write results to file / Read from file									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	12600	29	19	59	3	812	0.00%	53.2/sec	286.5
/styles/cook...	4200	26	16	54	1	583	0.00%	18.2/sec	75.2
/styles/boot...	4200	71	40	162	5	948	0.00%	18.2/sec	2157.2
/styles/style...	4200	28	17	55	1	698	0.00%	18.2/sec	28.5
/scripts/jqu...	4200	70	40	165	5	978	0.00%	18.2/sec	1949.3
/scripts/hel...	4200	27	16	55	1	910	0.00%	18.2/sec	41.6
/scripts/poly...	4200	28	18	54	1	939	0.00%	18.2/sec	137.0
/scripts/cod...	4200	38	23	83	2	800	0.00%	18.2/sec	624.3
/scripts/boo...	4200	41	23	92	3	822	0.00%	18.2/sec	663.5
/images/ne...	4200	85	46	202	5	1246	0.00%	18.2/sec	2770.6
/favicon.ico	4200	27	16	56	1	829	0.00%	18.2/sec	101.8
/security/log...	4200	31	20	62	4	952	0.00%	18.2/sec	100.9
/j_spring_s...	4200	165	103	358	7	2422	0.00%	18.2/sec	109.3
/followUp/u...	4200	210	144	470	8	2146	0.00%	18.2/sec	127.0
/j_spring_s...	4200	57	37	117	2	882	0.00%	18.2/sec	98.2
TOTAL	71400	58	26	134	1	2422	0.00%	301.2/sec	9041.2

Como podemos observar, a la hora de solicitar la página principal, tarda 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 18073 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 150 usuarios ejecutando nuevamente 30 acciones cada uno.

**Aggregate Report**

Name: Aggregate Report  
Comments:

Write results to file / Read from file

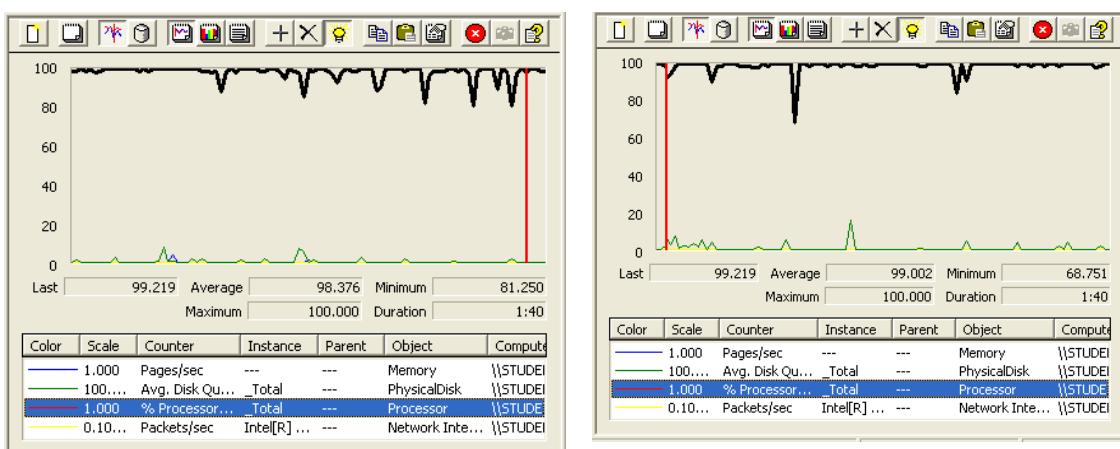
Filename  Browse... Log/Display Only:  Errors  Successes  Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	42	27	83	2	1209	0.00%	52.7/sec	284.0
/styles/cook...	4500	37	24	73	1	690	0.00%	18.0/sec	74.7
/styles/boot...	4500	106	66	251	6	1494	0.00%	18.0/sec	2140.1
/styles/style...	4500	40	25	82	1	877	0.00%	18.0/sec	28.3
/scripts/jqu...	4500	98	62	226	5	1130	0.00%	18.0/sec	1934.4
/scripts/hel...	4500	38	24	78	1	877	0.00%	18.0/sec	41.3
/scripts/poly...	4500	40	24	83	2	833	0.00%	18.0/sec	136.0
/scripts/coo...	4500	52	34	112	3	917	0.00%	18.0/sec	619.6
/scripts/boo...	4500	61	39	132	3	952	0.00%	18.0/sec	658.6
/images/ne...	4500	122	77	293	6	1121	0.00%	18.0/sec	2749.7
/favicon.ico	4500	42	25	88	1	820	0.00%	18.0/sec	100.8
/security/log...	4500	46	29	90	3	913	0.00%	18.0/sec	99.9
/j_spring_s...	4500	242	174	520	7	2898	0.00%	18.0/sec	108.3
/followUp/u...	4500	298	240	603	8	2438	0.00%	18.0/sec	125.7
/j_spring_s...	4500	82	55	174	3	989	0.00%	18.0/sec	97.3
TOTAL	76500	84	38	213	1	2898	0.00%	298.5/sec	8960.0

En este caso vemos como a la hora de solicitar 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 entrar en la aplicación. Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 17910.



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



## Listar Follow-ups de un artículo

En este caso de uso, un usuario puede listar los follow-ups de un artículo en un periódico. Para ello, sigue la siguiente secuencia de pasos:

Todos los periódicos > (Escoger uno) > (Escoger artículo) > Lista de seguidores

El límite de este caso de uso se sitúa en 160 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
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
/	4800	30	18	60	3	511	0.00%	16.2/sec	85.6
/styles/bootstrap...	4800	57	37	122	5	907	0.00%	16.2/sec	1923.1
/scripts/jquery....	4800	58	36	129	5	768	0.00%	16.2/sec	1737.7
/styles/cookie...	4800	28	15	81	1	556	0.00%	16.2/sec	87.1
/styles/style.css	4800	27	14	55	1	595	0.00%	16.2/sec	25.4
/scripts/cookie...	4800	33	20	64	2	720	0.00%	16.2/sec	556.6
/scripts/bootstrap...	4800	33	21	68	3	885	0.00%	16.2/sec	591.6
/scripts/polyfill...	4800	25	14	50	2	546	0.00%	16.2/sec	122.2
/scripts/helper...	4800	24	13	49	1	487	0.00%	16.2/sec	37.1
/images/news...	4800	52	32	120	7	597	0.00%	16.2/sec	2472.0
/newspaper/lin...	4800	116	51	264	7	2737	0.00%	16.2/sec	106.8
/newspaper/di...	4800	121	58	272	7	2962	0.00%	16.2/sec	121.4
/wp-content/u...	4800	665	639	730	562	2271	0.00%	16.1/sec	1837.1
/article/display...	4800	123	58	288	8	2655	0.00%	16.2/sec	110.0
/followUp/list.do	4800	170	88	378	8	3471	0.00%	16.2/sec	99.0
/welcome/ind...	4800	29	16	56	2	773	0.00%	16.2/sec	82.8
TOTAL	76800	100	28	261	1	3471	0.00%	251.5/sec	9677.9

Como podemos observar, a la hora de mostrar el periódico, necesita una imagen lo que implica que tarda cerca de 1 segundo, haciendo que nuestra web comience 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 1588 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 170 usuarios ejecutando nuevamente 30 acciones cada uno.

**Aggregate Report**

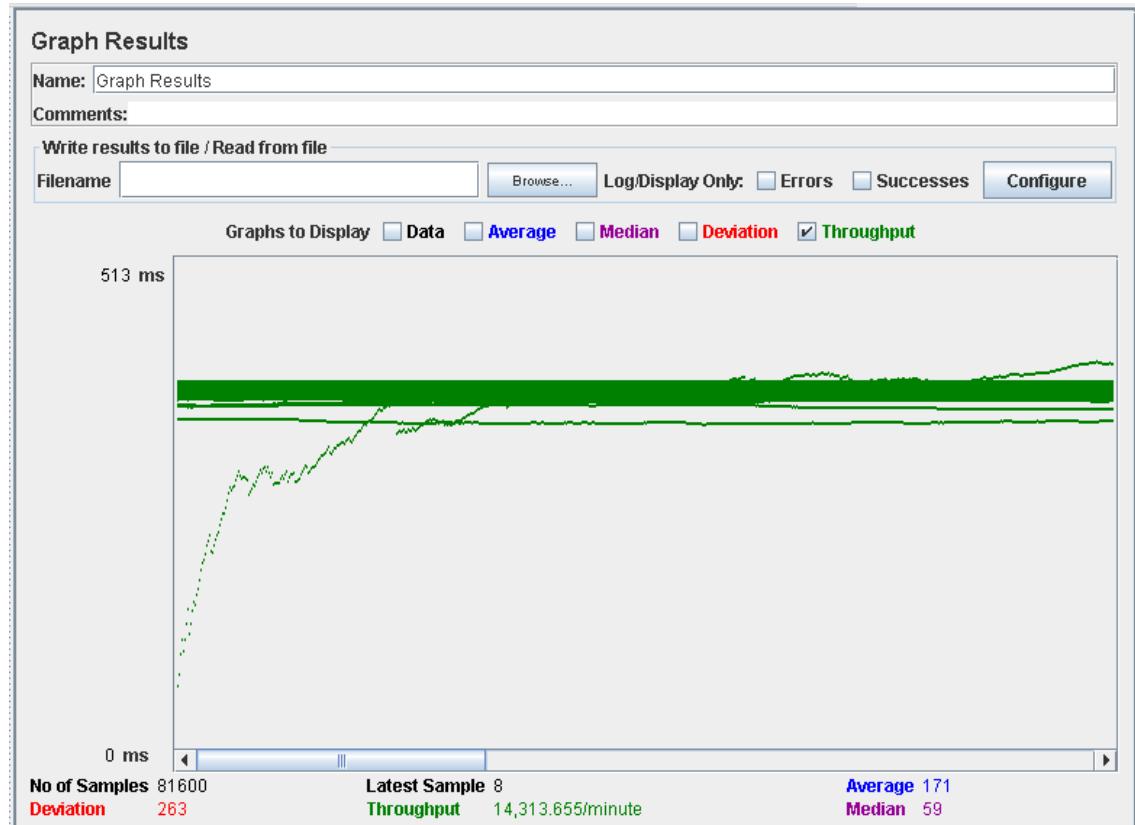
Name: Aggregate Report  
Comments:

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
/	5100	51	32	91	3	923	0.00%	15.3/sec	80.7
/styles/bootstrap...	5100	97	66	204	5	1275	0.00%	15.3/sec	1813.9
/scripts/jquery....	5100	96	64	203	5	1137	0.00%	15.3/sec	1639.0
/styles/cookie...	5100	52	29	107	1	1108	0.00%	15.3/sec	63.3
/styles/style.css	5100	49	28	96	1	1316	0.00%	15.3/sec	24.0
/scripts/cookie...	5100	59	37	116	2	1383	0.00%	15.3/sec	524.9
/scripts/bootstrap...	5100	63	41	129	2	1379	0.00%	15.3/sec	557.9
/scripts/polyfill...	5100	46	27	88	2	1054	0.00%	15.3/sec	115.2
/scripts/helper...	5100	47	28	94	1	1172	0.00%	15.3/sec	35.0
/images/news...	5100	109	71	246	6	1045	0.00%	15.3/sec	2327.9
/newspaper/lin...	5100	302	186	712	7	3455	0.00%	15.3/sec	100.6
/newspaper/dl...	5100	311	213	712	7	3278	0.00%	15.3/sec	114.3
/wp-content/u...	5100	697	651	811	557	1661	0.00%	15.3/sec	1737.3
/article/display...	5100	334	239	757	8	3393	0.00%	15.3/sec	104.1
/followUp/list.do	5100	382	255	809	8	3682	0.00%	15.3/sec	93.7
/welcome/ind...	5100	55	30	110	3	1283	0.00%	15.3/sec	78.4
TOTAL	81600	171	59	588	1	3682	0.00%	238.6/sec	9181.0

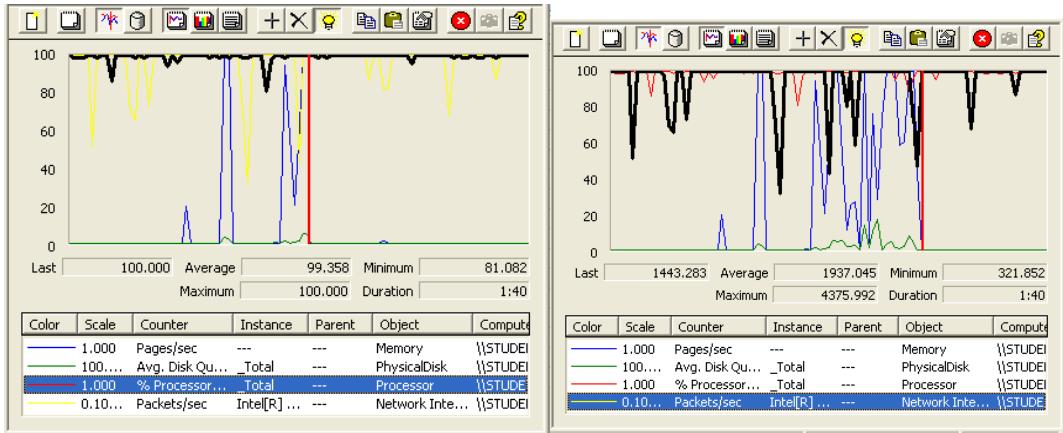
En este caso vemos como a la hora de mostrar el periódico, el sistema tarda de media en el 90% de las ocasiones más de un segundo, haciendo que los usuarios del sistema tengan que esperar más de lo habitual para que se responda a sus peticiones.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 14313.



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.





## Borrar Follow-up

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

Usuario > Mis continuaciones de artículo

El límite de este caso de uso se sitúa en 160 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
/	14400	22	15	42	3	815	0.00%	49.8/sec	268.7
/styles/style.css	4800	21	14	42	1	505	0.00%	17.1/sec	26.8
/styles/cookieconse...	4800	22	14	43	1	505	0.00%	17.1/sec	70.6
/styles/bootstrap.mi...	4800	53	39	107	5	646	0.00%	17.1/sec	2024.2
/scripts/jquery.min.js	4800	52	37	107	4	696	0.00%	17.1/sec	1829.3
/scripts/polyfills.js	4800	23	14	47	2	643	0.00%	17.1/sec	128.6
/scripts/cookiecons...	4800	29	21	59	2	592	0.00%	17.1/sec	585.8
/scripts/bootstrap.m...	4800	30	21	59	2	567	0.00%	17.1/sec	622.9
/scripts/helpers.js	4800	21	14	41	1	572	0.00%	17.1/sec	39.0
/images/newspaper...	4800	51	36	109	6	575	0.00%	17.1/sec	2597.6
/security/login.do	4800	27	17	49	3	647	0.00%	17.1/sec	94.6
/j_spring_security_c...	4800	191	94	479	6	2658	0.00%	17.1/sec	102.6
/followUp/user/list.do	9600	210	116	480	8	3394	0.00%	33.8/sec	222.3
/followUp/user/delet...	4800	137	62	339	5	3032	0.00%	17.0/sec	216.6
/j_spring_security_l...	4800	41	32	81	4	501	0.00%	17.0/sec	92.0
TOTAL	86400	66	25	139	1	3394	0.00%	298.9/sec	8695.1

Como podemos observar, a la hora de borrar el follow-up y redirigir al listado, se 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 17935 veces por minuto.

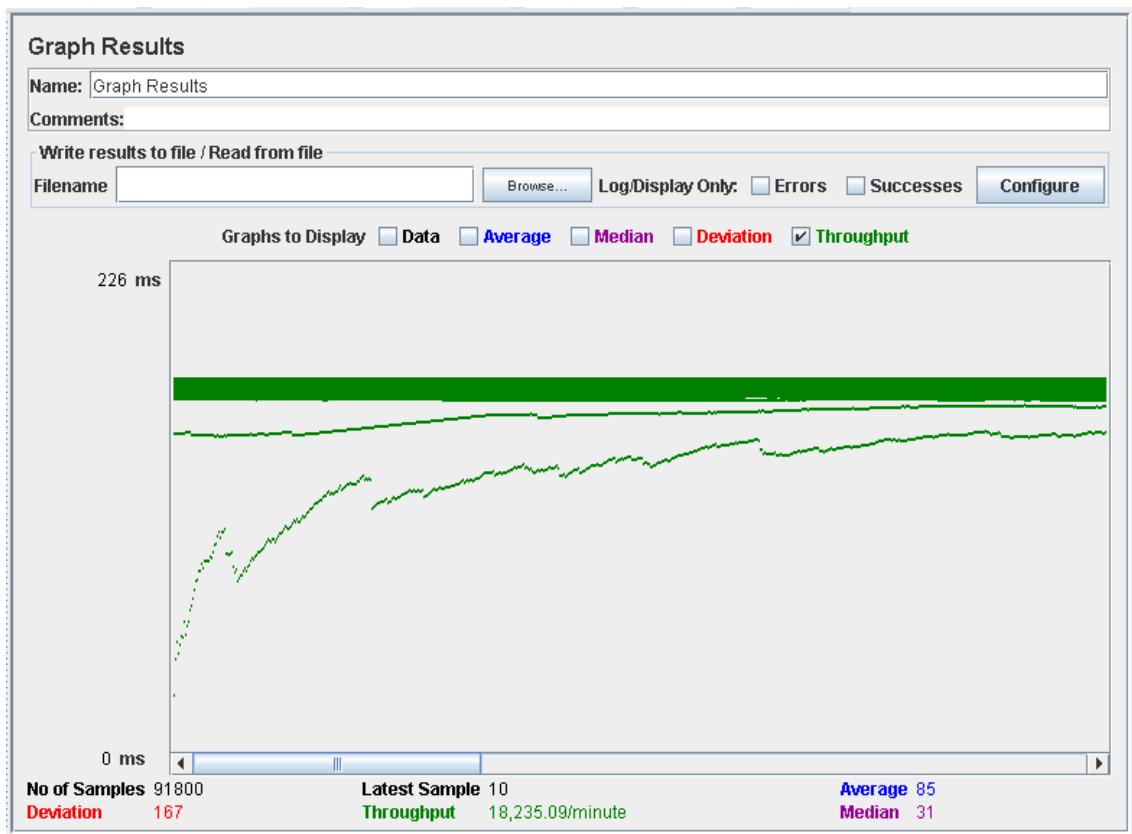


Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.  
En este caso se ha utilizado 170 usuarios ejecutando nuevamente 30 acciones cada uno.

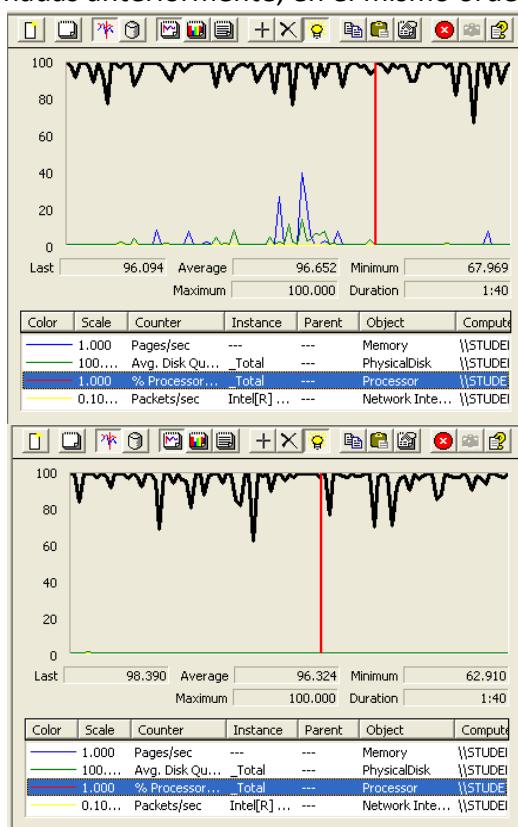
Aggregate Report										
Name: Aggregate Report										
Comments:										
Write results to file / Read from file										
Filename	Browse...			Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="checkbox"/> Configure			
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	15300	30	19	58	3	560	0.00%	50.7/sec	273.2	
/styles/style.css	5100	29	17	57	1	518	0.00%	17.4/sec	27.3	
/styles/cookieconse...	5100	31	18	60	1	520	0.00%	17.4/sec	72.0	
/styles/bootstrap.mi...	5100	64	45	131	6	574	0.00%	17.4/sec	2062.8	
/scripts/jquery.min.js	5100	64	43	139	4	840	0.00%	17.4/sec	1863.8	
/scripts/polyfills.js	5100	34	19	72	1	747	0.00%	17.4/sec	131.0	
/scripts/cookiecons...	5100	41	25	85	2	624	0.00%	17.4/sec	596.9	
/scripts/bootstrap.m...	5100	40	25	82	2	646	0.00%	17.4/sec	634.3	
/scripts/helpers.js	5100	29	17	59	1	525	0.00%	17.4/sec	39.8	
/images/newspaper...	5100	67	45	147	6	749	0.00%	17.4/sec	2648.4	
/security/login.do	5100	37	21	74	3	679	0.00%	17.3/sec	96.1	
/j_spring_security_c...	5100	242	135	602	7	3208	0.00%	17.3/sec	104.2	
/followUp/user/list.do	10200	250	153	598	8	2889	0.00%	34.4/sec	226.1	
/followUp/user/delet...	5100	193	93	512	6	3709	0.00%	17.3/sec	220.7	
/j_spring_security_l...	5100	62	39	133	5	806	0.00%	17.3/sec	93.4	
TOTAL	91800	85	31	201	1	3709	0.00%	303.9/sec	8840.6	

En este caso vemos como a la hora de borrar el follow-up y redirigir al listado, 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 18235.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos una gráfica 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 un usuario no autenticado despliega los términos y condiciones

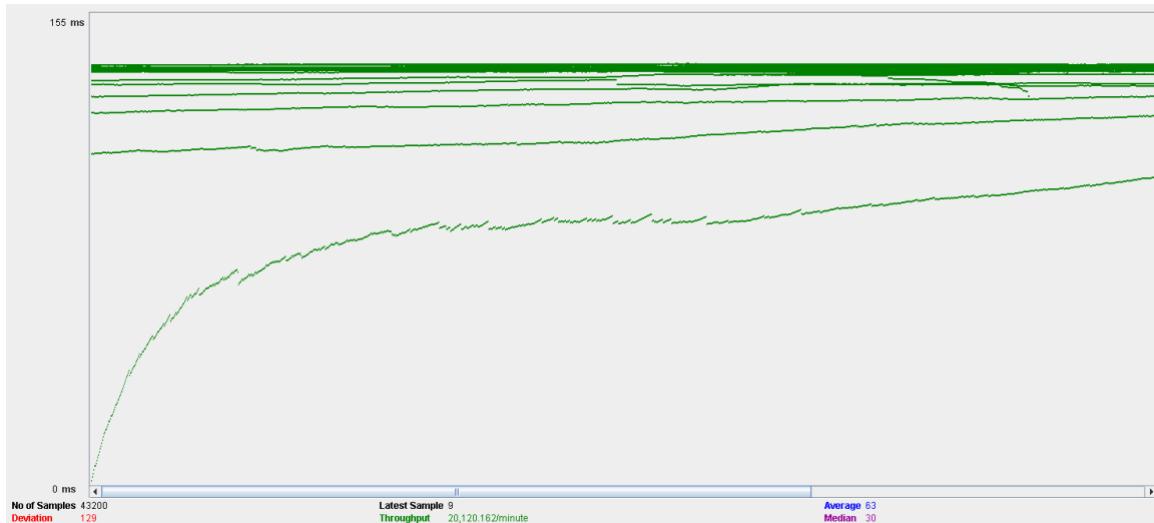
Terms and Condition

El límite de este caso de uso se sitúa en 120 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3600	39	25	65	0	1784	0.00%	28.7/sec	146.6
/styles/style.css	3600	32	22	55	0	1612	0.00%	28.6/sec	44.8
/styles/cookieconsent.mi...	3600	33	22	58	0	1301	0.00%	28.7/sec	118.7
/styles/bootstrap.min.css	3600	87	61	179	0	1320	0.00%	28.6/sec	396.1
/scripts/jquery.min.js	3600	79	63	164	0	2070	0.00%	28.7/sec	307.5
/scripts/polyfills.js	3600	31	20	53	0	657	0.00%	28.6/sec	215.6
/scripts/bootstrap.min.js	3600	43	28	94	0	759	0.00%	28.7/sec	161.0
/scripts/helpers.js	3600	32	18	49	0	1591	0.00%	28.6/sec	65.4
/scripts/cookiesent.js	3600	41	26	76	0	999	0.00%	28.7/sec	89.3
/images/newspaper-1.jpg	3600	96	64	221	0	1620	0.00%	28.7/sec	437.1
/termCondition/display.d...	3600	207	119	491	0	3205	0.00%	28.7/sec	191.0
/welcome/index.do	3600	35	24	58	0	1024	0.00%	28.7/sec	146.6
TOTAL	43200	63	30	171	0	3205	0.00%	335.3/sec	13452.9

Como podemos observar a la hora de desplegar los términos y condiciones con 120 usuarios da unos valores de tiempo asequibles.

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 20120 veces por minuto.



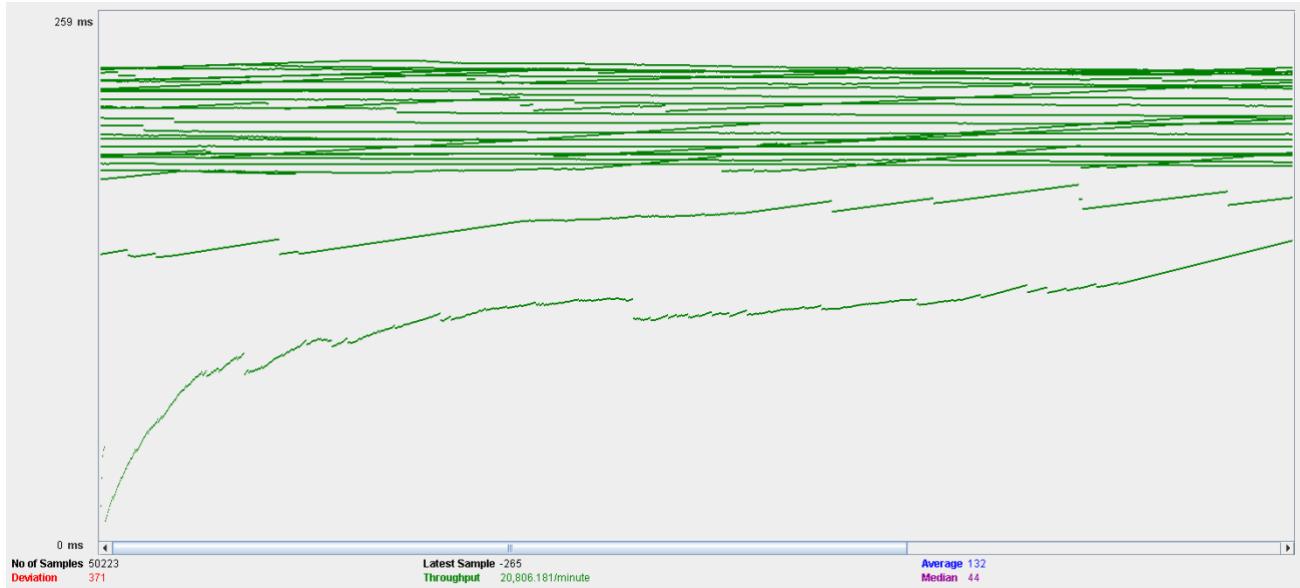
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 130 usuarios ejecutando nuevamente 30 acciones cada uno.

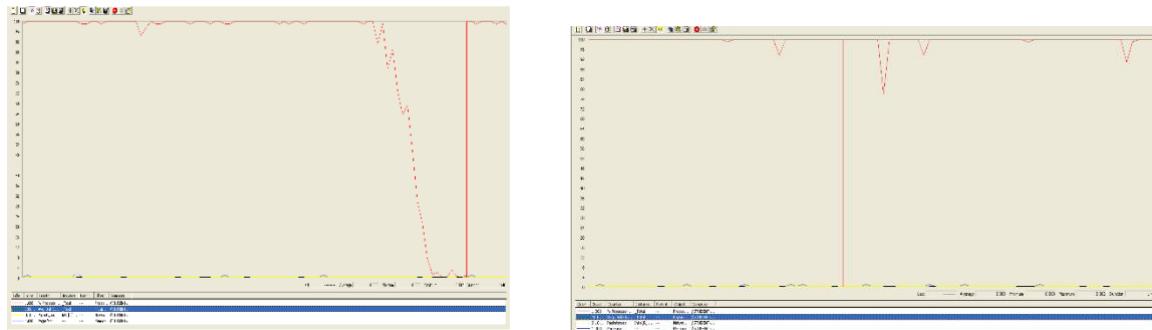
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4189	70	35	105	0	6300	0.00%	28.9/sec	148.0
/styles/style.css	4189	47	30	79	0	1979	0.00%	28.9/sec	45.3
/styles/cookieconsent.mi...	4188	49	31	81	0	1708	0.00%	29.0/sec	119.9
/styles/bootstrap.min.css	4188	122	91	239	0	2422	0.00%	29.0/sec	3438.6
/scripts/jquery.min.js	4188	120	84	244	0	1958	0.00%	29.0/sec	3113.3
/scripts/polyfills.js	4188	55	31	101	0	1542	0.00%	29.0/sec	218.3
/scripts/bootstrap.min.js	4188	65	42	128	0	1731	0.00%	29.0/sec	1059.6
/scripts/helpers.js	4188	48	26	86	0	1483	0.00%	29.0/sec	66.2
/scripts/cookiesent.js	4188	67	36	116	0	1526	0.00%	29.0/sec	994.9
/image/newspaper-1.jpg	4188	141	102	306	0	2889	0.00%	29.3/sec	4457.0
/termCondition/display.d...	4189	749	399	1818	0	7885	0.00%	29.5/sec	198.4
/welcome/index.do	4181	67	37	125	0	1897	0.00%	29.4/sec	150.1
TOTAL	50223	132	44	295	0	7885	0.00%	346.8/sec	13920.3

En este caso vemos al acceder al display de los términos y condiciones, el sistema tarda de media en el 90% de las ocasiones más de 1.8 segundos, 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 20806.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas



anteriormente, en el mismo orden.

## Desplegar About us

En este caso de uso un usuario no autenticado despliega el sobre nosotros de la empresa

### About us

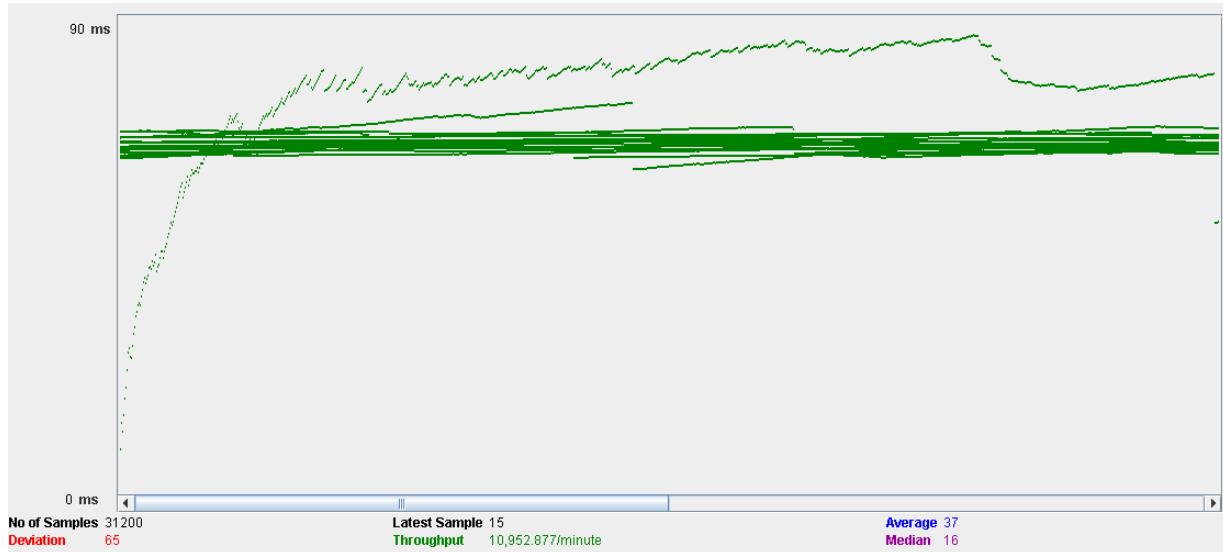
El límite de este caso de uso se sitúa en 80 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	2400	24	17	49	4	498	0.00%	14.3/sec	73.2
/styles/style.css	2400	16	10	37	1	352	0.00%	14.3/sec	22.4
/styles/cookiecon...	2400	16	10	37	1	352	0.00%	14.3/sec	59.3
/scripts/jquery.min...	2400	34	22	73	5	550	0.00%	14.3/sec	1536.8
/scripts/bootstrap....	2400	34	22	71	5	670	0.00%	14.3/sec	1700.8
/scripts/cookiecon...	2400	18	13	40	2	280	0.00%	14.3/sec	492.2
/scripts/polyfills.js	2400	13	9	28	1	250	0.00%	14.3/sec	108.0
/scripts/helpers.js	2400	12	8	26	1	185	0.00%	14.3/sec	32.7
/scripts/bootstrap....	2400	16	11	32	2	260	0.00%	14.3/sec	523.2
/images/newspap...	2400	43	27	92	6	631	0.00%	14.3/sec	2184.8
/aboutUs/display....	2400	18	13	36	3	339	0.00%	14.3/sec	86.7
/wp-content/themes...	2400	214	199	232	181	1983	0.00%	14.3/sec	528.2
/welcome/index.do	2400	23	15	48	4	550	0.00%	14.4/sec	73.3
TOTAL	31200	37	16	93	1	1983	0.00%	182.5/sec	7268.9

Como podemos observar a la hora de desplegar los términos y condiciones con 80 usuarios da unos valores de tiempo asequibles.

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 10952 veces por minuto.

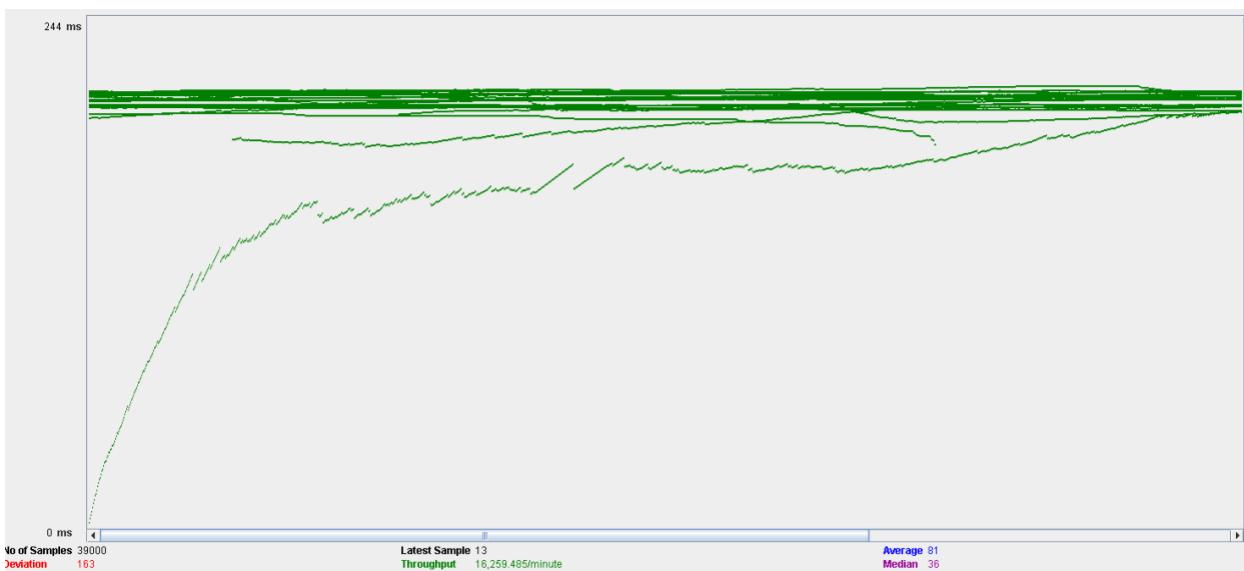


Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

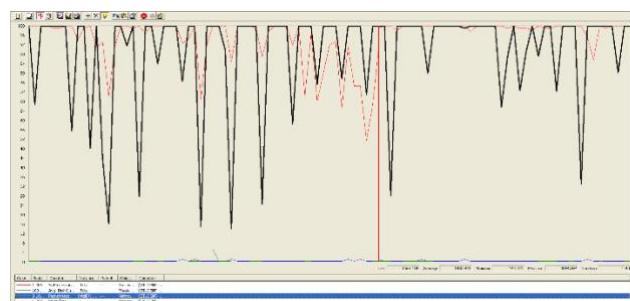
En este caso se ha utilizado 100 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3000	50	31	162	0	1526	0.00%	21.4/sec	109.2
/styles/style.css	3000	50	27	146	0	1749	0.00%	21.4/sec	33.4
/styles/cookieconsent.mi...	3000	45	26	125	0	1463	0.00%	21.4/sec	88.4
/scripts/jquery.min.js	3000	121	68	311	0	1647	0.00%	21.4/sec	2291.5
/styles/bootstrap.min.css	3000	112	68	303	0	1594	0.00%	21.4/sec	2535.9
/scripts/cookieconsent.js	3000	51	33	137	0	1491	0.00%	21.4/sec	734.2
/scripts/polyfills.js	3000	40	20	111	0	1496	0.00%	21.4/sec	161.1
/scripts/helpers.js	3000	33	18	90	0	2058	0.00%	21.4/sec	48.8
/scripts/bootstrap.min.js	3000	55	29	185	0	1605	0.00%	21.4/sec	780.5
/images/newspaper-1.jpg	3000	144	77	373	0	1702	0.00%	21.4/sec	3255.5
/aboutUs/display.do.js...	3000	47	25	142	0	1152	0.00%	21.4/sec	129.4
/wp-content/themes/slimm...	3000	257	212	474	0	4612	0.00%	21.3/sec	786.0
/welcome/index.do	3000	55	30	191	0	1524	0.00%	21.4/sec	109.1
TOTAL	39000	81	36	288	0	4612	0.00%	271.0/sec	10790.6

En este caso vemos que al cargar los datos de la página a iniciarla, el sistema tarda de media en el 90% de las ocasiones más de 2 segundos, 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 16259.



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



## Editar términos y condiciones

En este caso de uso un usuario autenticado como administrador despliega edita los términos y condiciones

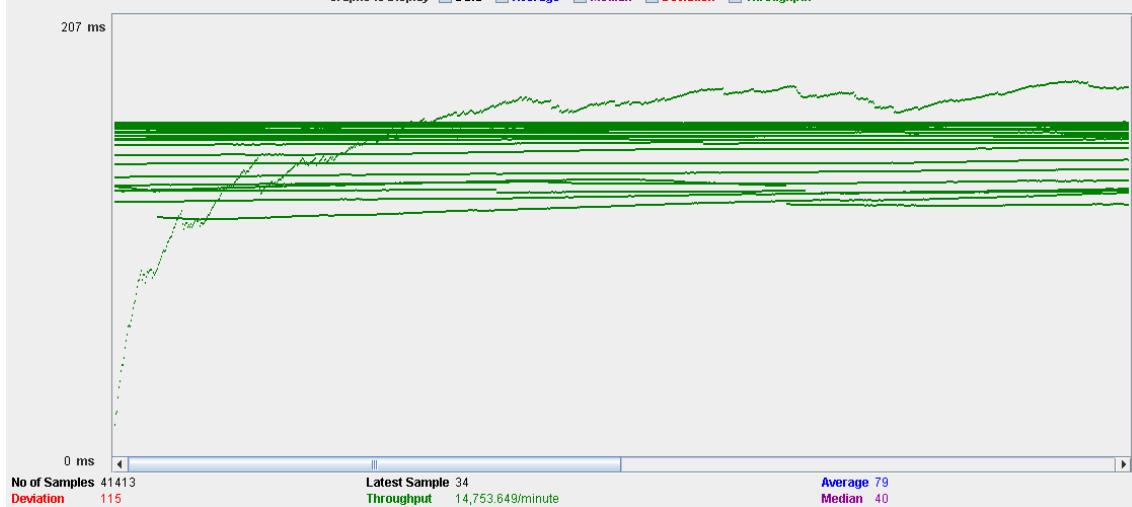
Menú Principal > Entrar [admin] > Términos y condiciones > Editar español

El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7235	35	25	64	4	817	0.00%	38.5/sec	210.2
/scripts/jquery.min...	2482	73	54	150	5	844	0.00%	13.2/sec	1415.5
/styles/style.css	2480	35	22	69	2	583	0.00%	13.2/sec	20.7
/styles/cookiecon...	2477	35	23	63	2	640	0.00%	13.2/sec	54.6
/styles/bootstrap....	2476	75	56	156	6	763	0.00%	13.2/sec	1563.9
/scripts/polyfills.js	2476	33	22	60	2	723	0.00%	13.2/sec	99.3
/scripts/bootstrap....	2474	46	32	94	3	611	0.00%	13.2/sec	480.4
/scripts/helpers.js	2473	31	21	59	1	529	0.00%	13.2/sec	30.1
/scripts/cookiecon...	2473	43	29	85	3	594	0.00%	13.2/sec	452.2
/images/newsrap...	2455	82	56	180	7	1158	0.00%	13.2/sec	2007.5
/security/login.do	2443	36	25	65	4	573	0.00%	13.3/sec	73.5
/j_spring_security...	2443	129	102	233	8	1910	0.00%	13.2/sec	84.6
/termCondition/ad...	4731	160	130	308	8	2067	0.00%	26.2/sec	259.6
/termCondition/ad...	4710	207	146	442	6	2273	0.00%	26.3/sec	249.6
/j_spring_security...	2342	73	52	142	6	903	0.00%	13.3/sec	71.8
TOTAL	46170	80	40	194	1	2273	0.00%	245.5/sec	7020.3

Como podemos observar a la hora editar los términos y condiciones con 150 usuarios concurrentes los valores temporales son inferiores al medio segundo, por lo que es bastante eficiente

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 14753 veces por minuto.

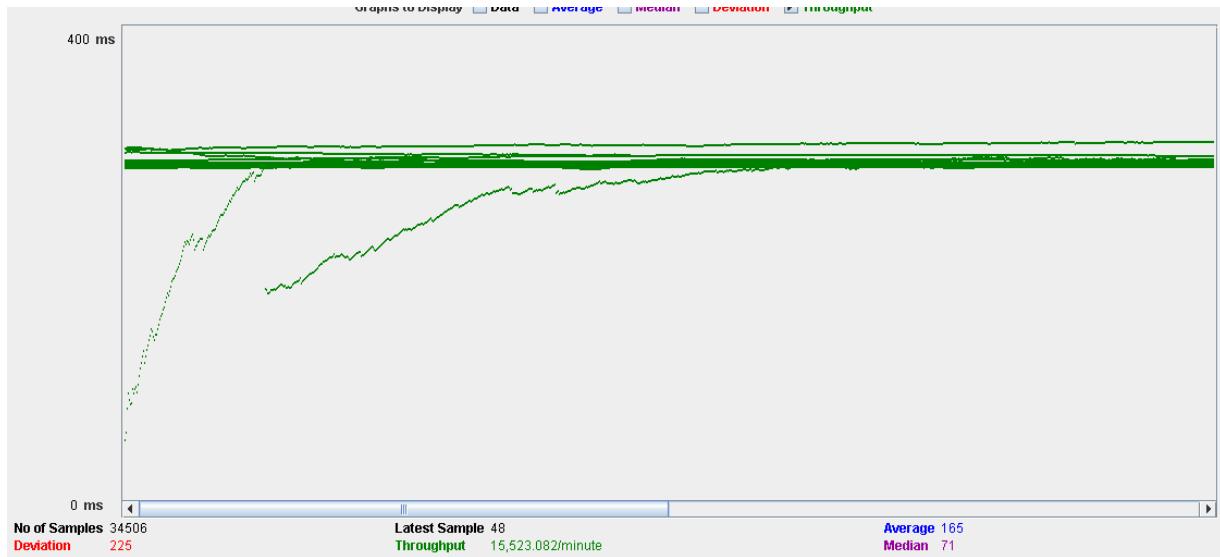


Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

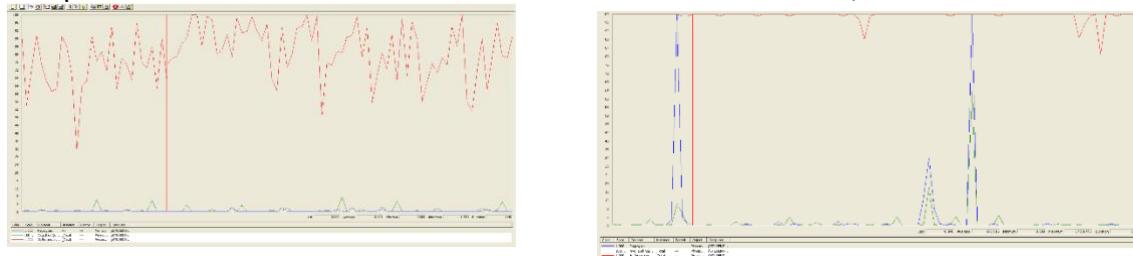
En este caso se ha utilizado 160 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8215	89	38	251	3	1571	0.00%	39.4/sec	215.4
/scripts/jquery.min...	2821	139	83	338	5	1694	0.00%	13.5/sec	1452.1
/styles/style.css	2820	86	34	263	1	1538	0.00%	13.5/sec	21.2
/styles/cookiecon...	2820	86	34	258	2	1101	0.00%	13.5/sec	66.1
/styles/bootstrap....	2819	137	83	339	6	2320	0.00%	13.5/sec	1605.9
/scripts/polyfills.js	2819	89	33	273	2	973	0.00%	13.5/sec	102.0
/scripts/bootstrap....	2818	112	54	307	4	1470	0.00%	13.5/sec	493.7
/scripts/helpers.js	2818	83	34	237	2	1023	0.00%	13.5/sec	30.9
/scripts/cookiecon...	2817	95	47	249	3	1325	0.00%	13.5/sec	464.4
/images/newsrap...	2817	163	95	399	6	2697	0.00%	13.6/sec	2074.7
/security/login.do	2788	97	36	279	4	1630	0.00%	13.6/sec	75.4
/j_spring_security...	2756	363	251	777	12	3124	0.00%	13.4/sec	86.0
/termCondition/ad...	5402	305	224	649	8	2883	0.09%	26.6/sec	263.9
/termCondition/ad...	5367	431	304	939	8	3798	0.00%	26.7/sec	252.8
/j_spring_security...	2641	225	128	549	7	2405	0.00%	13.3/sec	71.6
TOTAL	52538	178	74	482	1	3798	0.01%	261.9/sec	7222.1

En este caso vemos que al cargar los datos de la página a iniciarla, el sistema tarda de media en el 90% de las ocasiones casi 3 segundos, 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 16259.



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



## Actualizar palabras tabúes y comprobar su efecto con los anuncios

En este caso de uso, un administrador puede actualizar las palabras tabúes y comprobar su efecto. Para ello, sigue la siguiente secuencia de pasos:

Administrador > Configuración

El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

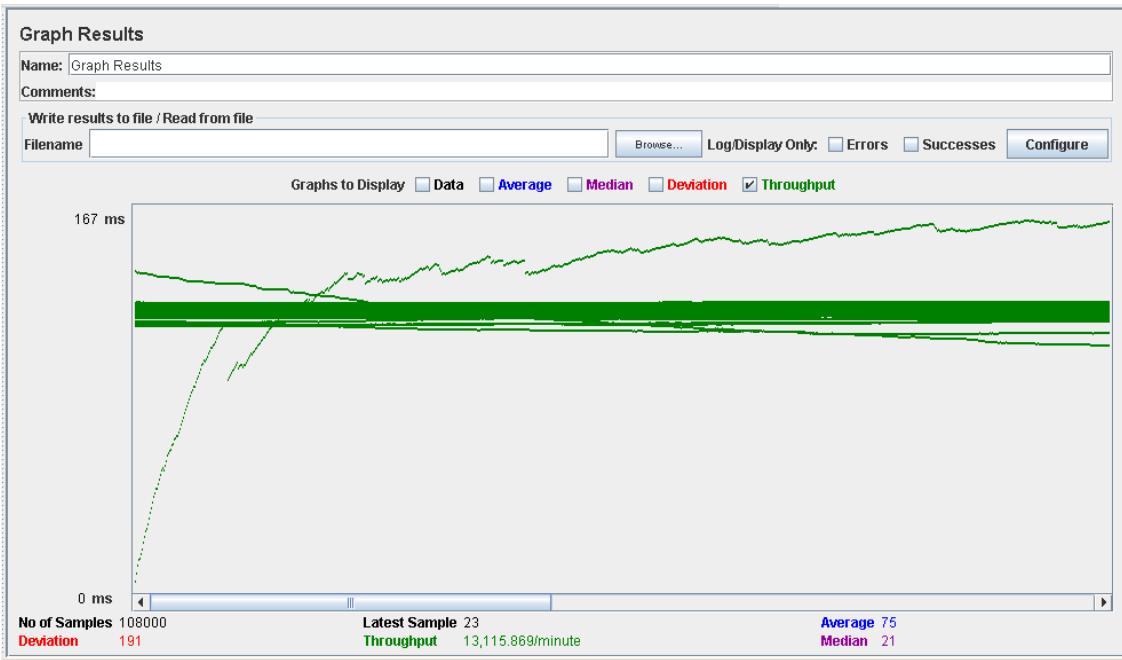
Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Filename <input type="text"/>					Browse...	Log/Display Only:	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	18000	20	12	41	2	743	0.00%	36.4/sec	206.9
/styles/cookieconsent.min.css	4500	18	11	39	1	496	0.00%	9.4/sec	38.8
/styles/style.css	4500	17	10	38	1	541	0.00%	9.4/sec	14.7
/styles/bootstrap.min.css	4500	49	27	106	5	692	0.00%	9.4/sec	1113.5
/scripts/jquery.min.js	4500	47	26	99	5	1051	0.00%	9.4/sec	1006.2
/scripts/helpers.js	4500	17	10	38	1	760	0.00%	9.4/sec	21.5
/scripts/cookieconsent.js	4500	26	15	56	2	715	0.00%	9.4/sec	322.2
/scripts/polyfills.js	4500	18	10	38	1	687	0.00%	9.4/sec	70.7
/scripts/bootstrap.min.js	4500	28	15	61	2	719	0.00%	9.4/sec	342.4
/images/newspaper-1.jpg	4500	57	31	120	6	863	0.00%	9.4/sec	1429.8
/favicon.ico	4500	18	9	40	1	643	0.00%	9.4/sec	52.5
/security/login.do	4500	20	12	41	3	531	0.00%	9.4/sec	52.1
/j_spring_security_check	4500	121	48	286	6	3448	0.00%	9.4/sec	60.0
/advertisement/administrator/list.do	9000	143	52	349	7	4045	0.00%	18.5/sec	154.0
/configuration/administrator/display.do	9000	123	41	300	5	4523	0.00%	18.6/sec	119.4
/configuration/administrator/edit.do	9000	175	73	438	5	5793	0.00%	18.8/sec	127.5
/configuration/administrator/searchTa...	4500	361	203	866	25	5155	0.00%	9.4/sec	59.9
/j_spring_security_logout	4500	40	24	80	2	690	0.00%	9.4/sec	50.6
TOTAL	108000	75	21	166	1	5793	0.00%	218.6/s...	5097.9

Como podemos observar, a la hora de buscar las palabras tabúes y volver a la pantalla de inicio, 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 13115 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 155 usuarios ejecutando nuevamente 30 acciones cada uno.

**Aggregate Report**

Name: Aggregate Report  
Comments:

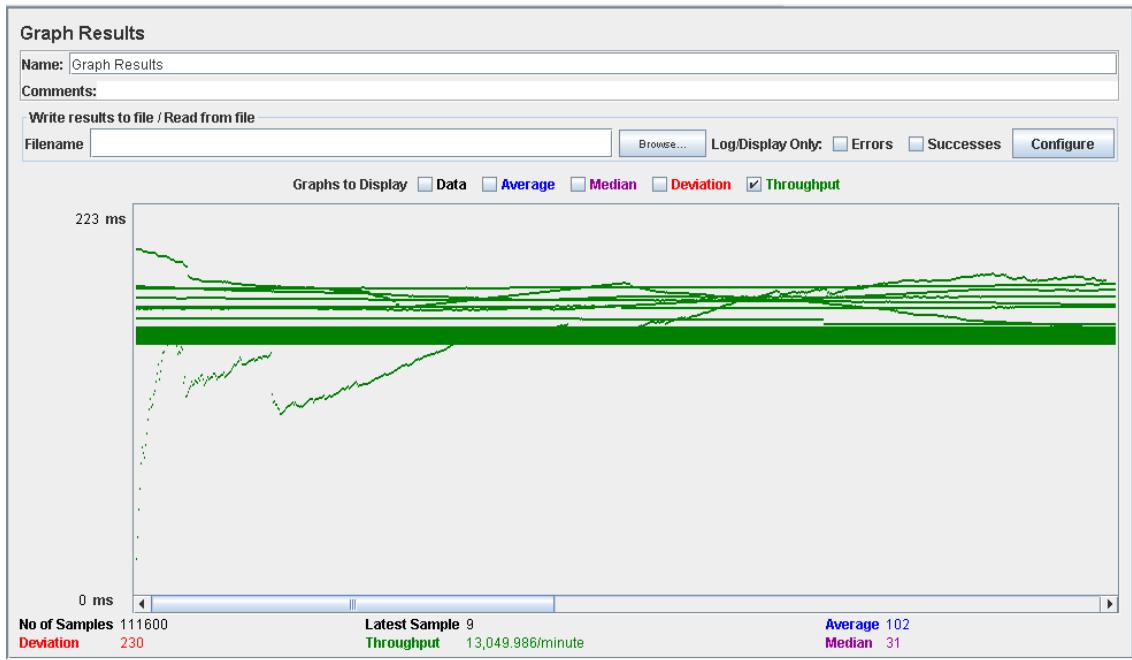
Write results to file / Read from file  
Filename:  Browse... Log/Display Only:  Errors  Successes  Throughput Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	18600	29	18	58	2	1181	0.00%	36.2/sec	205.9
/styles/cookieconsent.min.css	4650	26	16	55	2	650	0.00%	9.3/sec	38.5
/styles/style.css	4650	26	16	53	1	931	0.00%	9.3/sec	14.6
/styles/bootstrap.min.css	4650	68	37	156	5	1591	0.00%	9.3/sec	1104.4
/scripts/jquery.min.js	4650	63	36	146	5	1073	0.00%	9.3/sec	997.9
/scripts/helpers.js	4650	26	17	56	1	522	0.00%	9.3/sec	21.3
/scripts/cookieconsent.js	4650	37	22	83	2	935	0.00%	9.3/sec	319.5
/scripts/polyfills.js	4650	27	17	53	2	961	0.00%	9.3/sec	70.1
/scripts/bootstrap.min.js	4650	39	23	87	2	938	0.00%	9.3/sec	339.6
/images/newspaper-1.jpg	4650	80	42	185	6	1464	0.00%	9.3/sec	1417.9
/favicon.ico	4650	26	15	56	2	1203	0.00%	9.3/sec	52.1
/security/login.do	4650	27	17	56	4	827	0.00%	9.3/sec	61.7
/j_spring_security_check	4650	177	87	409	7	4208	0.00%	9.3/sec	59.5
/advertisement/administrator/list.do	9300	168	83	405	8	3918	0.00%	18.3/sec	152.9
/configuration/administrator/display.do	9300	153	69	378	4	3933	0.00%	18.5/sec	118.4
/configuration/administrator/edit.do	9300	233	120	559	5	5064	0.00%	18.6/sec	126.3
/configuration/administrator/searchTa...	4650	547	356	1217	22	5625	0.00%	9.3/sec	59.3
/j_spring_security_logout	4650	59	37	121	5	1823	0.00%	9.3/sec	50.1
TOTAL	111600	102	31	245	1	5625	0.00%	217.5/s...	5072.6

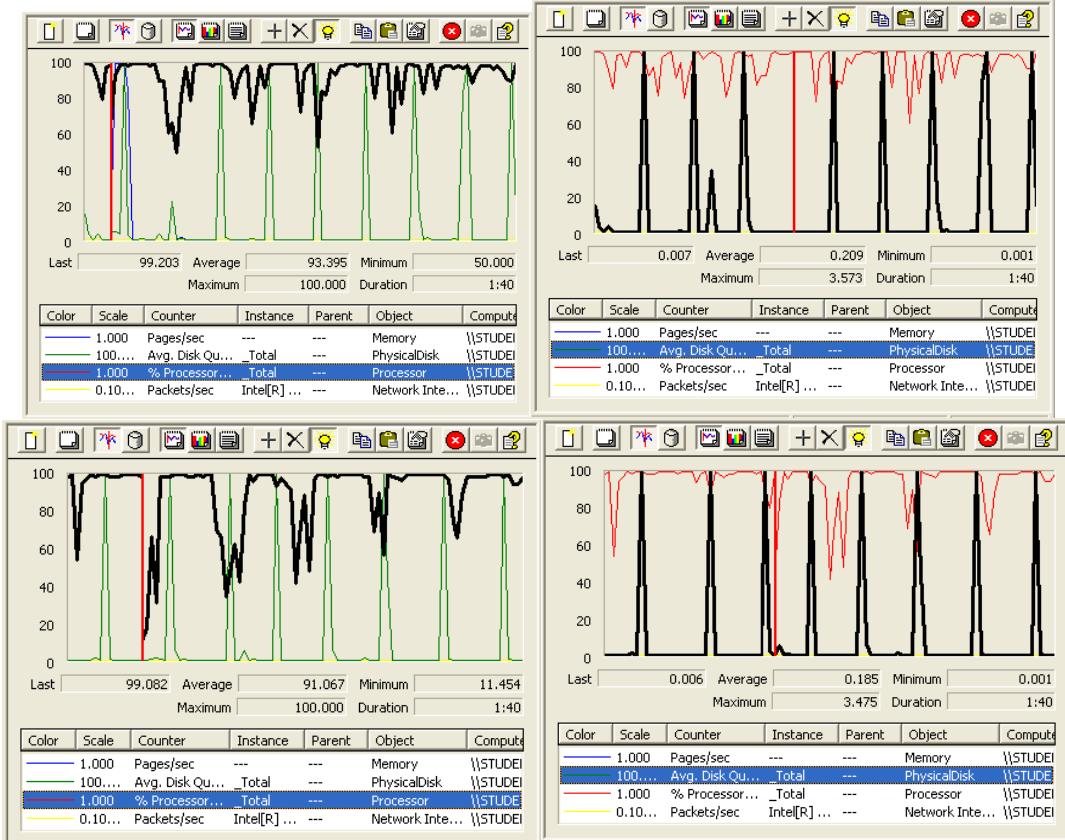
Include group name in label?  Save Table Data  Save Table Header

En este caso vemos como a la hora de buscar las palabras tabúes y volver a 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 13049.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y del disco. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden. Cabe destacar, el fallo del disco, pues se tienen que recorrer varias tablas ya que el administrador a solicitado calcular de nuevo las entidades que poseen tabú.



## Borrar anuncios por administrador

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

Administrador > Lista de anuncios

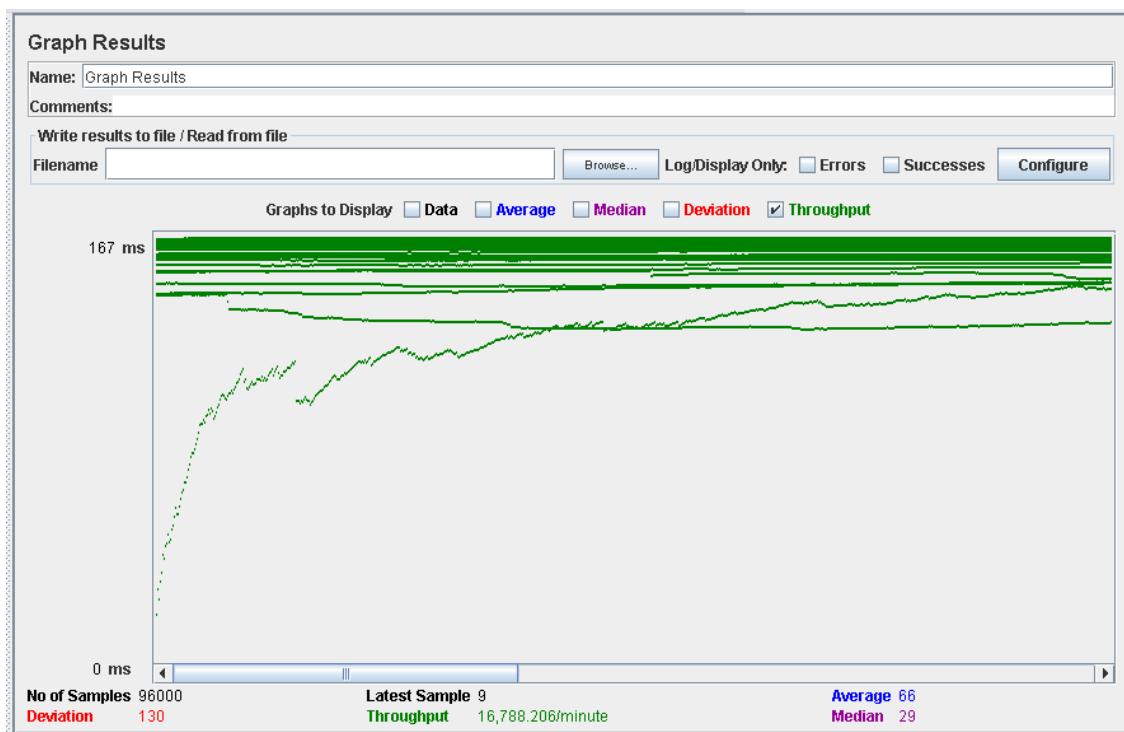
El límite de este caso de uso se sitúa en 160 usuarios concurrentes ejecutando 30 veces la misma acción.

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	
/	14400	27	19	52	2	1108	0.00%	42.0/sec	231.6	
/scripts/jquery.min.js	4800	56	39	118	5	650	0.00%	14.3/sec	1537.8	
/styles/bootstrap.min.css	4800	59	41	125	5	1213	0.00%	14.3/sec	1701.9	
/styles/style.css	4800	26	17	56	1	530	0.00%	14.3/sec	22.5	
/styles/cookieconsent.min.css	4800	25	17	49	1	601	0.00%	14.3/sec	59.4	
/scripts/helpers.js	4800	23	16	47	1	1218	0.00%	14.3/sec	32.8	
/scripts/bootstrap.min.js	4800	35	24	75	2	619	0.00%	14.3/sec	523.5	
/scripts/polyfills.js	4800	24	17	48	1	1122	0.00%	14.3/sec	108.1	
/scripts/cookieconsent.js	4800	33	23	68	2	472	0.00%	14.3/sec	492.6	
/images/newspaper-1.jpg	4800	65	45	144	5	1304	0.00%	14.3/sec	2185.8	
/favicon.ico	4800	23	15	43	2	1174	0.00%	14.4/sec	80.4	
/security/login.do	4800	29	20	54	4	1158	0.00%	14.4/sec	79.7	
/j_spring_security_check	4800	183	110	405	7	2378	0.00%	14.4/sec	91.8	
/advertisement/administrator/list.do	14400	161	92	350	6	3331	0.00%	42.8/sec	338.1	
/advertisement/administrator/delete...	4800	128	72	292	5	2255	0.00%	14.4/sec	189.3	
/j_spring_security_logout	4800	51	38	98	5	597	0.00%	14.4/sec	77.8	
TOTAL	96000	66	29	145	1	3331	0.00%	279.8/sec	7567.6	

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 16788 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 170 usuarios ejecutando nuevamente 30 acciones cada uno.

**Aggregate Report**

Name: Aggregate Report  
Comments:

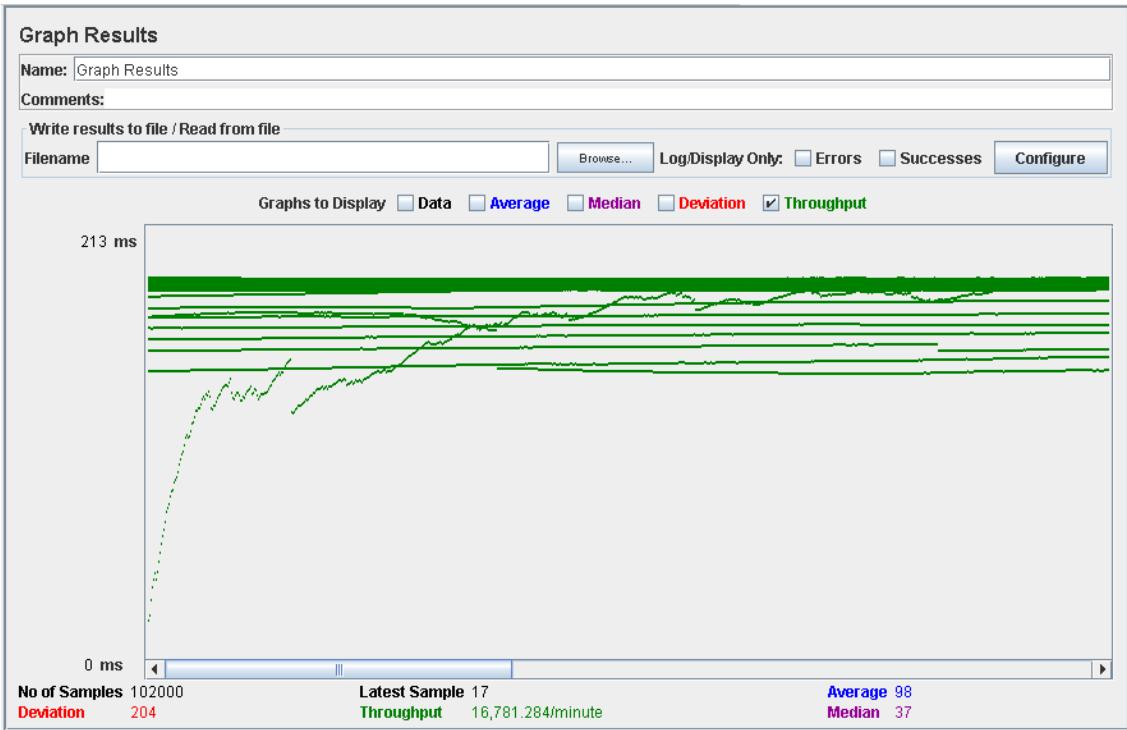
Write results to file / Read from file

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

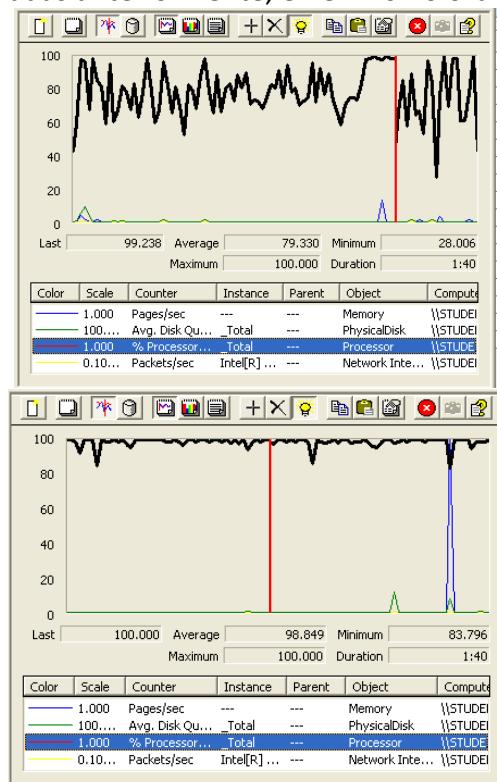
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15300	34	23	64	3	711	0.00%	42.0/sec	231.5
/scripts/jquery.min.js	5100	70	50	143	5	820	0.00%	14.4/sec	1541.0
/styles/bootstrap.min.css	5100	74	52	155	5	814	0.00%	14.4/sec	1705.6
/styles/style.css	5100	37	23	73	1	758	0.00%	14.4/sec	22.8
/styles/cookieconsent.min.css	5100	34	22	66	1	697	0.00%	14.4/sec	59.5
/scripts/helpers.js	5100	33	21	64	1	697	0.00%	14.4/sec	32.9
/scripts/bootstrap.min.js	5100	46	31	91	3	743	0.00%	14.4/sec	524.5
/scripts/polyfills.js	5100	33	21	65	2	698	0.00%	14.4/sec	108.3
/scripts/cookieconsent.js	5100	42	29	86	3	853	0.00%	14.4/sec	493.6
/images/newspaper-1.jpg	5100	81	57	173	6	832	0.00%	14.4/sec	2190.0
/favicon.ico	5100	32	20	63	2	682	0.00%	14.4/sec	80.5
/security/login.do	5100	38	24	73	3	801	0.00%	14.4/sec	79.7
/j_spring_security_check	5100	284	158	665	7	4199	0.00%	14.4/sec	91.8
/advertisement/administrator/list.do	15300	250	130	623	7	4325	0.00%	42.7/sec	338.0
/advertisement/administrator/delete...	5100	229	103	568	5	4801	0.00%	14.4/sec	188.8
/j_spring_security_logout	5100	72	50	138	3	1161	0.00%	14.4/sec	77.9
TOTAL	102000	98	37	212	1	4801	0.00%	279.7/s...	7564.8

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

En este caso vemos como a la hora de borrar y volver al listado, 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 16781.



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



## Listar anuncios y desplegarlos

En este caso de uso, un agente puede listar sus anuncios y desplegarlos. Para ello, sigue la siguiente secuencia de pasos:

Agente > Mis anuncios

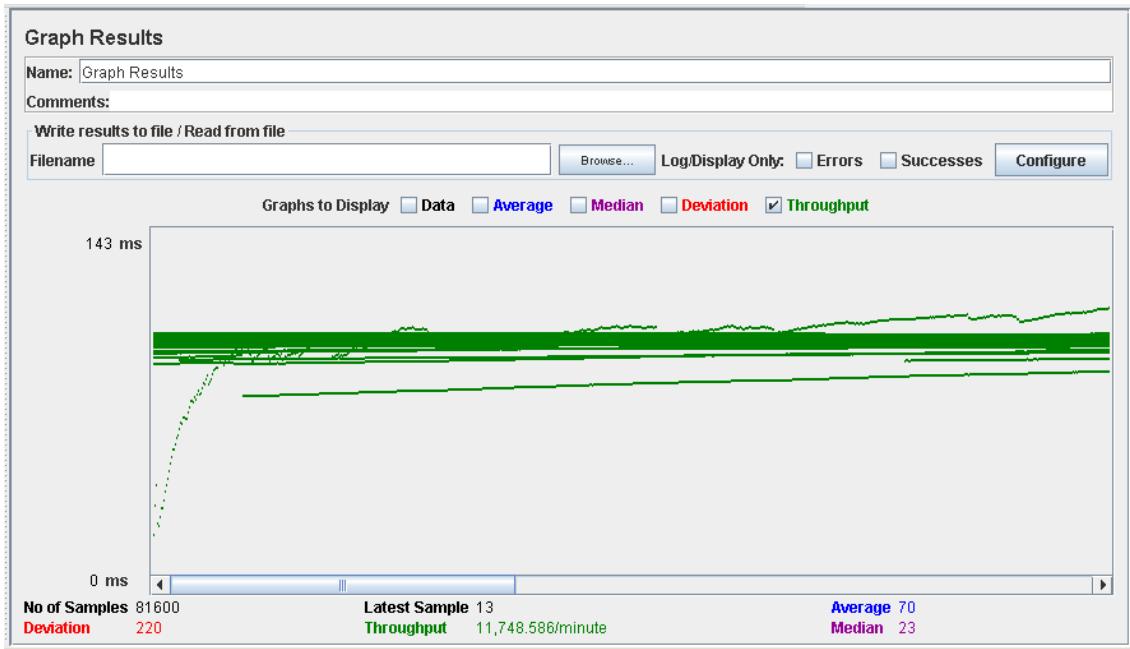
El límite de este caso de uso se sitúa en 160 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report									
Name: Aggregate Report									
Comments:									
Write results to file / Read from file									
Filename <input type="text"/>				<input type="button" value="Browse..."/>		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
/	14400	25	16	45	2	1005	0.00%	34.6/sec	184.5
/styles/style.c...	4800	21	13	42	1	1125	0.00%	11.9/sec	18.6
/styles/cookie...	4800	21	13	41	1	1104	0.00%	11.9/sec	49.2
/styles/bootst...	4800	55	35	122	4	1218	0.00%	11.9/sec	1410.3
/scripts/help...	4800	23	13	46	1	1206	0.00%	11.9/sec	27.2
/scripts/cookie...	4800	31	20	61	2	1185	0.00%	11.9/sec	408.2
/scripts/jquer...	4800	51	33	111	4	964	0.00%	11.9/sec	1274.6
/scripts/polyfil...	4800	22	13	44	1	1198	0.00%	11.9/sec	89.6
/images/new...	4800	59	38	129	6	680	0.00%	11.9/sec	1811.1
/scripts/bootstrap...	4800	32	20	67	3	541	0.00%	11.8/sec	431.0
/security/login...	4800	28	17	50	3	1058	0.00%	11.8/sec	65.5
/j_spring_sec...	4800	333	103	719	7	5155	0.00%	11.8/sec	69.1
/advertiseme...	4800	212	101	553	8	3192	0.00%	11.8/sec	97.0
/advertiseme...	4800	176	67	477	5	2965	0.00%	11.8/sec	75.2
/j_spring_sec...	4800	53	33	96	4	2610	0.00%	11.8/sec	63.8
TOTAL	81600	70	23	128	1	5155	0.00%	195.8/sec	5897.1

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 11748 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 170 usuarios ejecutando nuevamente 30 acciones cada uno.

**Aggregate Report**

Name: Aggregate Report  
Comments:

Write results to file / Read from file

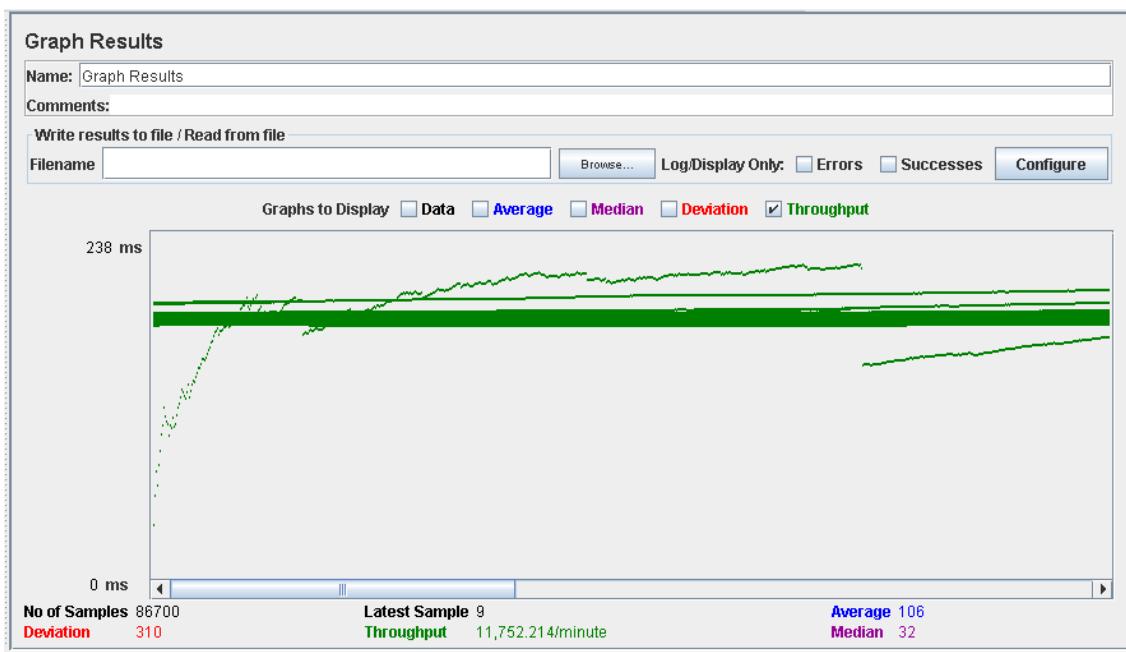
Filename   Log/Display Only:  Errors  Successes

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15300	34	20	60	3	2618	0.00%	34.6/sec	184.6
/styles/style.c...	5100	31	18	58	1	2031	0.00%	11.8/sec	18.5
/styles/cookie...	5100	31	19	57	1	1959	0.00%	11.8/sec	48.9
/styles/bootstrap...	5100	69	47	150	5	1388	0.00%	11.8/sec	1400.0
/scripts/helperr...	5100	32	19	61	1	1334	0.00%	11.8/sec	27.0
/scripts/cookie...	5100	42	27	83	2	806	0.00%	11.8/sec	405.1
/scripts/jquery...	5100	67	45	138	5	1186	0.00%	11.8/sec	1265.1
/scripts/polyfill...	5100	35	19	66	1	2175	0.00%	11.8/sec	88.9
/images/new...	5100	77	52	161	5	1144	0.00%	11.8/sec	1797.7
/scripts/bootstrap...	5100	46	27	93	3	1559	0.00%	11.8/sec	430.4
/security/login...	5100	39	24	76	4	980	0.00%	11.8/sec	65.6
/j_spring_sec...	5100	510	189	1210	8	7247	0.00%	11.8/sec	69.2
/advertiseme...	5100	347	175	878	10	4922	0.00%	11.8/sec	96.6
/advertiseme...	5100	289	119	786	5	4789	0.00%	11.8/sec	75.0
/j_spring_sec...	5100	81	46	139	5	3045	0.00%	11.8/sec	63.7
TOTAL	86700	106	32	203	1	7247	0.00%	195.9/sec	5899.1

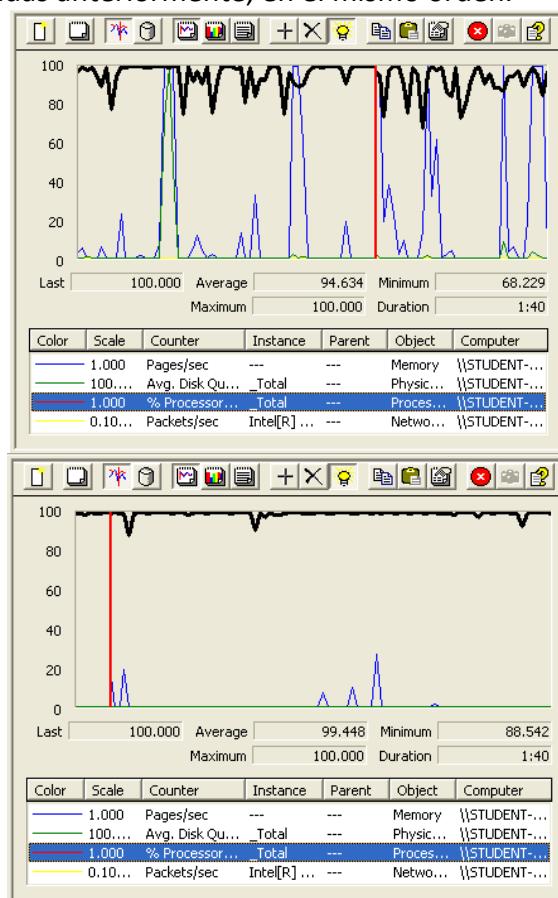
Include group name in label?   Save Table Header

En este caso vemos como a la hora de autenticarse, 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 poder autenticarse.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 11752.



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



## Crear un anuncio

En este caso de uso, un agente puede crear un anuncio. Para ello, sigue la siguiente secuencia de pasos:

Agente > Mis anuncios

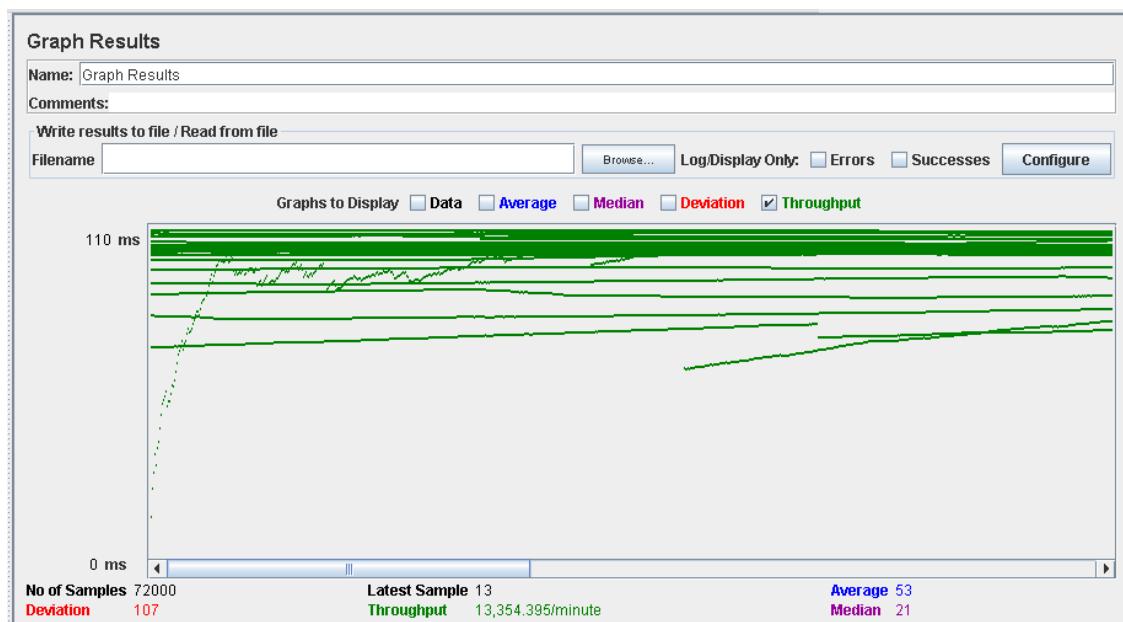
El límite de este caso de uso se sitúa en 120 usuarios concurrentes ejecutando 30 veces la misma acción.

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	
/	10800	18	13	36	2	586	0.00%	33.4/sec	178.2	
/styles/cookieconsent.min.css	3600	16	12	32	1	499	0.00%	11.5/sec	47.5	
/styles/style.css	3600	15	11	33	1	579	0.00%	11.5/sec	18.0	
/styles/bootstrap.min.css	3600	46	28	110	5	1438	0.00%	11.5/sec	1363.9	
/scripts/jquery.min.js	3600	42	26	96	4	758	0.00%	11.5/sec	1232.5	
/scripts/helpers.js	3600	16	12	33	1	460	0.00%	11.5/sec	26.2	
/scripts/cookieconsent.js	3600	22	16	47	2	542	0.00%	11.5/sec	394.8	
/scripts/bootstrap.min.js	3600	23	16	48	2	634	0.00%	11.5/sec	419.4	
/scripts/polyfills.js	3600	15	11	31	1	608	0.00%	11.5/sec	86.6	
/images/newspaper-1.jpg	3600	50	30	123	6	949	0.00%	11.5/sec	1751.6	
/favicon.ico	3600	13	9	29	1	558	0.00%	11.5/sec	64.2	
/security/login.do	3600	18	14	36	3	522	0.00%	11.5/sec	63.9	
/j_spring_security_check	3600	93	60	204	7	1355	0.00%	11.5/sec	67.4	
/advertisementagent/list.do	7200	115	73	253	8	1673	0.00%	22.8/sec	1584.1	
/advertisement/agent/create.do	3600	99	60	220	6	1540	0.00%	11.6/sec	100.8	
/advertisementagent/edit.do	3600	264	175	606	11	3256	0.00%	11.6/sec	807.7	
/j_spring_security_logout	3600	34	27	67	5	901	0.00%	11.5/sec	62.2	
TOTAL	72000	53	21	125	1	3256	0.00%	222.6/sec	8019.7	

Como podemos observar, a la hora de guardar el anuncio y regresar al listado, se 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 13354 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.

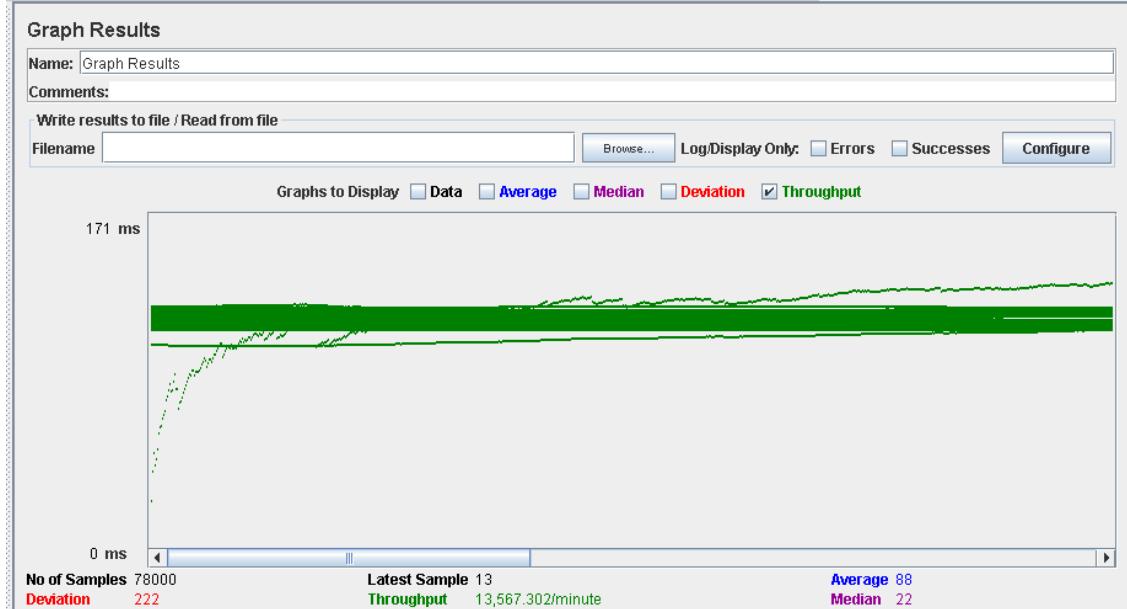
En este caso se ha utilizado 130 usuarios ejecutando nuevamente 30 acciones cada uno.

Aggregate Report										
<input type="text" value="Aggregate Report"/> Name: Aggregate Report <input type="text" value="Comments:"/> Comments: <input type="checkbox"/> Write results to file / Read from file <input type="text" value="Filename"/> 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	
/	11696	19	14	40	3	459	0.00%	34.2/sec	182.7	
/styles/cookieconsent.min.css	3900	17	12	36	1	532	0.00%	11.6/sec	48.0	
/styles/style.css	3900	16	11	35	1	540	0.00%	11.6/sec	18.2	
/styles/bootstrap.min.css	3900	39	24	90	4	779	0.00%	11.6/sec	1377.4	
/scripts/jquery.min.js	3900	36	23	82	4	760	0.00%	11.6/sec	1244.6	
/scripts/helpers.js	3900	16	11	36	1	179	0.00%	11.6/sec	26.5	
/scripts/cookieconsent.js	3900	22	15	47	2	806	0.00%	11.6/sec	398.5	
/scripts/bootstrap.min.js	3900	23	15	52	2	886	0.00%	11.6/sec	423.5	
/scripts/polyfills.js	3900	16	11	35	1	706	0.00%	11.6/sec	87.4	
/images/newspaper-1.jpg	3900	43	26	99	6	952	0.00%	11.6/sec	1768.5	
/favicon.ico	3900	17	11	38	2	497	0.00%	11.6/sec	64.9	
/security/login.do	3900	21	15	44	3	672	0.00%	11.6/sec	64.5	
/j_spring_security_check	3900	216	108	543	7	2760	0.00%	11.6/sec	68.1	
/advertisement/agent/list.do	7796	239	122	597	8	3285	0.00%	23.2/sec	1732.8	
/advertisement/agent/create.do	3896	210	107	506	6	2978	0.00%	11.7/sec	102.2	
/advertisement/agent/edit.do	3896	487	334	1156	20	3752	0.00%	11.7/sec	878.3	
/j_spring_security_logout	3896	40	30	76	3	918	0.00%	11.7/sec	63.2	
TOTAL	77980	88	22	198	1	3752	0.00%	228.2/sec	8397.3	

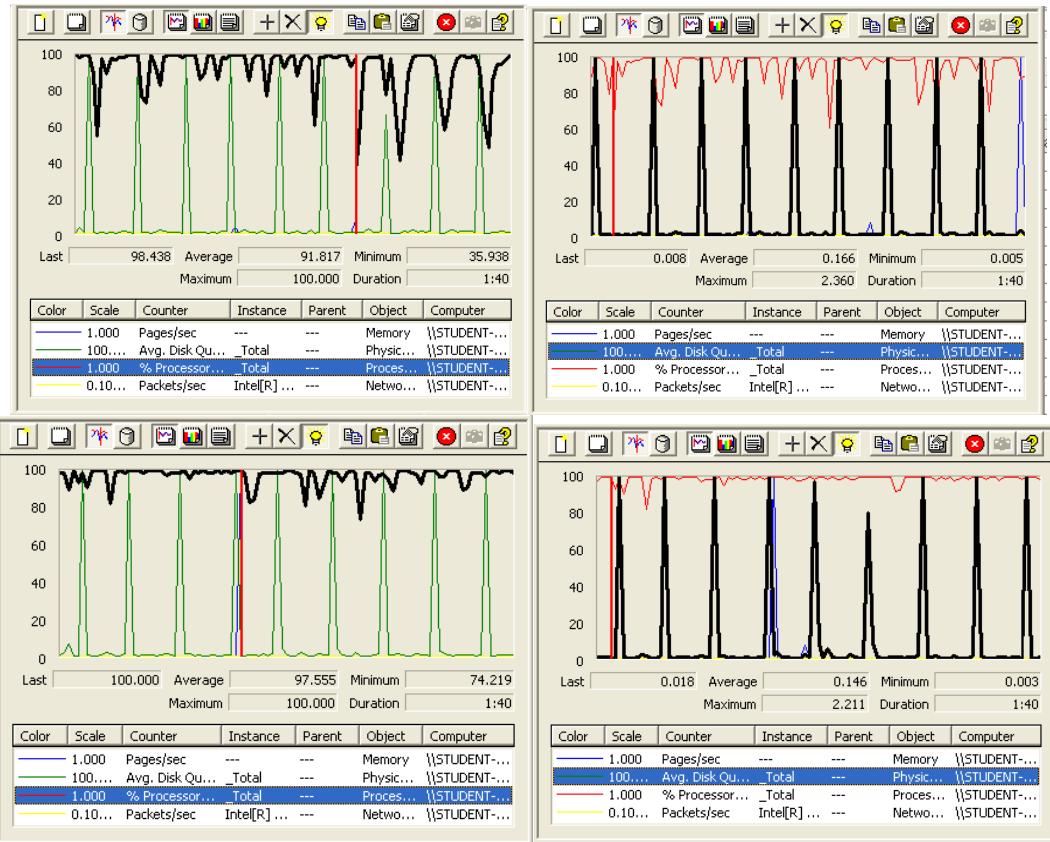
Include group name in label?   Save Table Header

En este caso vemos como a la hora de guardar el anuncio, 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 13567.



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



## Editar anuncio

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

Agente > Mis anuncios

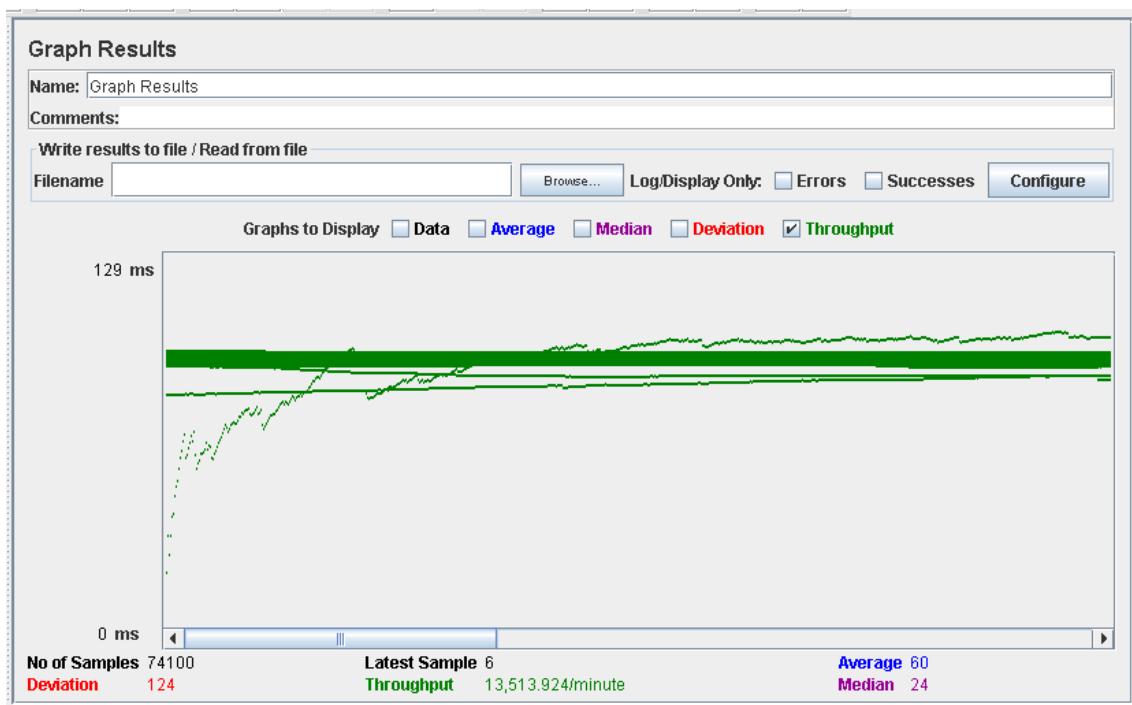
El límite de este caso de uso se sitúa en 130 usuarios concurrentes ejecutando 30 veces la misma acción.

Aggregate Report										
Write results to file / Read from file										
Filename			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	
/	11700	22	15	43	3	1086	0.00%	35.6/sec	189.8	
/styles/style.css	3900	21	14	43	1	705	0.00%	12.2/sec	19.1	
/styles/bootstrap.min.css	3900	60	33	137	4	1217	0.00%	12.2/sec	1445.4	
/scripts/jquery.min.js	3900	57	32	128	4	1161	0.00%	12.2/sec	1306.1	
/styles/cookieconsent.min.css	3900	22	14	45	1	904	0.00%	12.2/sec	50.4	
/scripts/polyfills.js	3900	22	14	45	1	829	0.00%	12.2/sec	91.8	
/scripts/bootstrap.min.js	3900	33	19	65	2	1119	0.00%	12.2/sec	444.4	
/scripts/helpers.js	3900	20	14	41	1	877	0.00%	12.2/sec	27.8	
/scripts/cookieconsent.js	3900	28	18	58	2	886	0.00%	12.2/sec	418.2	
/images/newspaper-1.jpg	3900	60	29	142	6	1454	0.00%	12.2/sec	1855.3	
/security/login.do	3900	24	14	43	3	1892	0.00%	12.2/sec	67.6	
/j_spring_security_check	3900	108	59	234	7	2589	0.00%	12.2/sec	71.3	
/advertisementagent/list.do	7800	116	64	252	8	2575	0.00%	24.2/sec	198.4	
/advertisementagent/edit.do	7800	168	78	440	6	3000	0.00%	24.3/sec	214.2	
/j_spring_security_logout	3900	47	32	90	5	1598	0.00%	12.2/sec	65.7	
TOTAL	74100	60	24	132	1	3000	0.00%	225.2/sec	6299.2	

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 13513 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 140 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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
/	12600	22	16	40	2	1027	0.00%	36.7/sec	195.9
/styles/style.css	4200	18	14	36	1	467	0.00%	12.6/sec	19.7
/styles/bootstrap.min.css	4200	51	35	108	5	823	0.00%	12.6/sec	1494.3
/scripts/jquery.min.js	4200	46	32	94	5	607	0.00%	12.6/sec	1350.2
/styles/cookieconsent.min.css	4200	19	14	37	1	534	0.00%	12.6/sec	52.1
/scripts/polyfills.js	4200	18	14	35	1	605	0.00%	12.6/sec	94.9
/scripts/bootstrap.min.js	4200	27	19	54	3	457	0.00%	12.6/sec	459.5
/scripts/helpers.js	4200	18	14	35	1	641	0.00%	12.6/sec	28.8
/scripts/cookieconsent.js	4200	25	19	49	2	526	0.00%	12.6/sec	432.4
/images/newspaper-1.jpg	4200	54	36	118	5	645	0.00%	12.6/sec	1917.8
/security/login.do	4200	23	18	42	3	447	0.00%	12.6/sec	69.8
/j_spring_security_check	4200	246	120	666	7	2940	0.00%	12.6/sec	73.7
/advertisement/agent/list.do	8400	228	110	615	8	2948	0.00%	25.0/sec	205.3
/advertisement/agent/edit.do	8400	300	137	815	6	3493	0.00%	25.2/sec	222.2
/j_spring_security_logout	4200	44	35	79	2	1652	0.00%	12.6/sec	68.1
TOTAL	79800	90	26	196	1	3493	0.00%	232.4/sec	6500.8

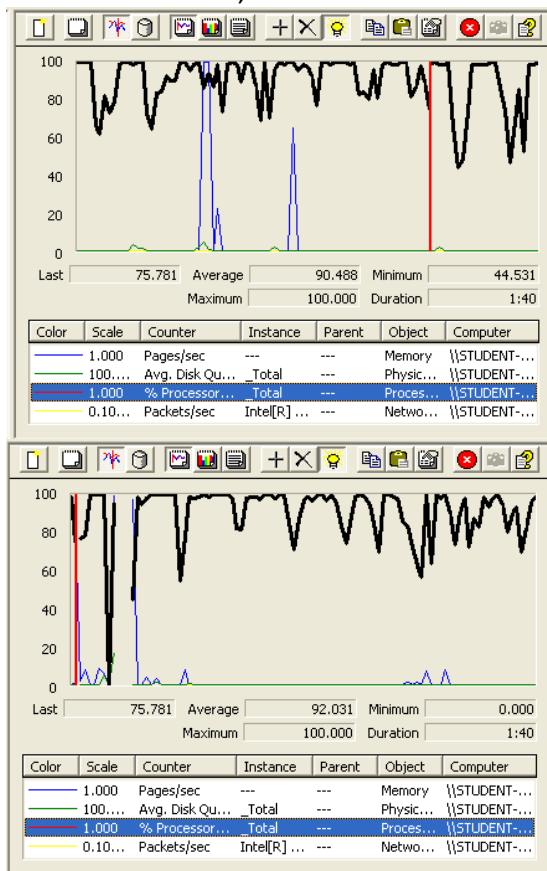
Include group name in label?   Save Table Header

En este caso vemos como a la hora de guardar el anuncio y regresar a su listado, 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 13945.



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



## Borrar anuncio

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

## Agente > Mis anuncios

El límite de este caso de uso es más difícil de observar, situándose en 150 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:  Errors  Successes  Configure

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	18	13	34	2	1298	0.00%	41.8/sec	223.2
/styles/bootstrap.min.css	4500	42	31	84	5	829	0.00%	14.3/sec	1700.5
/scripts/cookieconsent.js	4500	22	16	42	2	575	0.00%	14.3/sec	492.0
/scripts/jquery.min.js	4500	39	29	78	5	869	0.00%	14.3/sec	1536.6
/styles/style.css	4500	16	11	31	1	645	0.00%	14.3/sec	22.5
/styles/cookieconsent.min.css	4500	16	11	32	1	488	0.00%	14.3/sec	59.3
/scripts/bootstrap.min.js	4500	22	17	43	2	462	0.00%	14.3/sec	522.9
/scripts/helpers.js	4500	16	11	30	1	501	0.00%	14.3/sec	32.8
/images/newspaper-1.jpg	4500	48	35	94	6	734	0.00%	14.3/sec	2183.3
/scripts/polyfills.js	4500	16	11	31	1	555	0.00%	14.3/sec	108.0
/favicon.ico	4500	15	11	30	2	337	0.00%	14.3/sec	80.2
/security/login.do	4500	19	14	34	3	428	0.00%	14.3/sec	79.5
/j_spring_security_check	4500	142	65	313	5	3477	0.00%	14.3/sec	83.9
/advertisement/agent/list.do	9000	134	57	298	8	5300	0.00%	28.4/sec	228.6
/advertisement/agent/edit.do	9000	119	45	272	5	4130	0.00%	28.7/sec	317.8
/j_spring_security_logout	4500	34	26	65	3	580	0.00%	14.3/sec	77.5
TOTAL	90000	50	20	95	1	5300	0.00%	278.7/s...	7542.7

Include group name in label?   Save Table Header

Como podemos observar, a la hora de borrar y regresar al listado, tarda menos de 1 segundo, lo que indica que nuestra web está respondiendo bien frente al número de usuarios activos.

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 16723 veces por minuto.



Por otro lado, vamos a presentar el caso negativo que nos permite establecer el límite anterior. Para así ver cómo se comportaría nuestro sistema.

En este caso se ha utilizado 155 usuarios ejecutando nuevamente 30 acciones cada uno.

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			Configure		
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec	
/	13950	29	20	53	2	2169	0.00%	40.0/sec	213.3	
/styles/bootstrap.min.css	4650	70	43	150	5	1506	0.00%	13.7/sec	1629.0	
/scripts/cookieconsent.js	4650	37	25	72	2	1550	0.00%	13.7/sec	471.4	
/scripts/jquery.min.js	4650	66	43	143	5	1218	0.00%	13.7/sec	1472.2	
/styles/style.css	4650	26	18	50	1	733	0.00%	13.7/sec	21.5	
/scripts/cookieconsent.min.css	4650	26	18	49	1	846	0.00%	13.7/sec	56.9	
/scripts/bootstrap.min.js	4650	37	25	72	3	1443	0.00%	13.7/sec	501.0	
/scripts/helpers.js	4650	25	17	46	1	844	0.00%	13.7/sec	31.4	
/images/newspaper-1.jpg	4650	78	45	174	7	1691	0.00%	13.7/sec	2092.0	
/scripts/polyfills.js	4650	25	17	50	1	603	0.00%	13.7/sec	103.5	
/favicon.ico	4650	26	18	49	1	795	0.00%	13.7/sec	76.8	
/security/login.do	4650	30	21	54	3	643	0.00%	13.7/sec	76.1	
/j_spring_security_check	4650	252	132	643	6	3179	0.00%	13.7/sec	80.4	
/advertisement/agent/list.do	9300	247	128	628	8	4700	0.00%	27.1/sec	218.0	
/advertisement/agent/edit.do	9300	220	100	581	6	5353	0.00%	27.4/sec	303.8	
/j_spring_security_logout	4650	59	42	110	5	1066	0.00%	13.7/sec	73.9	
TOTAL	93000	89	31	192	1	5353	0.00%	266.3/s...	7208.0	

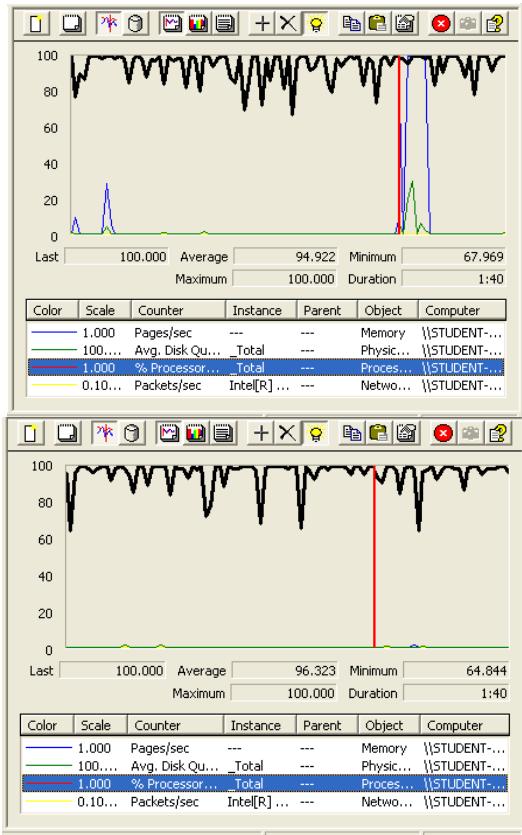
Include group name in label?   Save Table Header

En este caso vemos como a la hora de eliminar el anuncio y volver al listado, 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 15980.



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



## Enlazar anuncio con periódico

En este caso de uso, un agente puede enlazar un anuncio suyo con un periódico. Para ello, sigue la siguiente secuencia de pasos:

Perfil > Todos los periódicos

El límite de este caso de uso se sitúa en 150 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 checked="" type="checkbox"/> Configure					
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec	
/	13500	27	21	48	3	1048	0.00%	40.3/sec	215.1	
/scripts/jquery.min.js	4500	64	40	143	5	913	0.00%	13.8/sec	1478.3	
/styles/style.css	4500	25	19	46	1	856	0.00%	13.8/sec	21.6	
/styles/cookieconsent.min.css	4500	25	19	46	1	1168	0.00%	13.8/sec	57.1	
/styles/bootstrap.min.css	4500	68	42	150	5	1168	0.00%	13.8/sec	1636.0	
/scripts/helpers.js	4500	25	19	47	1	549	0.00%	13.8/sec	31.5	
/scripts/polyfills.js	4500	25	19	46	1	1614	0.00%	13.8/sec	103.9	
/scripts/cookieconsent.js	4500	34	25	64	3	1841	0.00%	13.8/sec	473.4	
/scripts/bootstrap.min.js	4500	36	26	72	2	775	0.00%	13.8/sec	503.1	
/images/newspaper-1.jpg	4500	75	45	171	6	1305	0.00%	13.8/sec	2100.5	
/favicon.ico	4500	25	19	46	2	705	0.00%	13.8/sec	77.2	
/security/login.do	4500	28	22	51	3	584	0.00%	13.8/sec	76.4	
/j_spring_security_check	4500	154	106	324	7	2034	0.00%	13.8/sec	80.7	
/newspaper/list.do	4500	192	141	377	11	1888	0.00%	13.8/sec	132.1	
/advertisement/agent/listLink.do	9000	152	103	306	8	2601	0.00%	27.5/sec	208.2	
/advertisement/agent/link.do	4500	140	95	286	7	2660	0.00%	13.8/sec	188.8	
/j_spring_security_logout	4500	54	44	97	5	1127	0.00%	13.8/sec	74.5	
TOTAL	90000	68	32	160	1	2680	0.00%	268.5/sec	7269.2	

Include group name in label?   Save Table Header

Como podemos observar, a la hora 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 16111 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 160 usuarios ejecutando nuevamente 30 acciones cada uno.

**Aggregate Report**

Name: Aggregate Report  
Comments:

Write results to file / Read from file

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
/	2377	44	29	81	3	904	0.00%	38.6/sec	206.7
/scripts/jquery.min.js	863	73	42	164	6	940	0.12%	14.1/sec	1506.5
/styles/style.css	861	40	22	72	1	786	0.00%	14.0/sec	22.0
/styles/cookieconsent.min.css	861	39	22	76	1	553	0.00%	14.0/sec	58.1
/styles/bootstrap.min.css	861	74	41	171	6	933	0.00%	14.0/sec	1665.1
/scripts/helpers.js	861	41	23	78	1	814	0.00%	14.0/sec	32.1
/scripts/polyfills.js	861	38	22	72	2	801	0.00%	14.0/sec	105.7
/scripts/cookieconsent.js	860	44	26	89	2	808	0.00%	14.0/sec	481.1
/scripts/bootstrap.min.js	860	49	27	108	3	884	0.00%	14.0/sec	511.4
/images/newspaper-1.jpg	860	85	46	183	7	1681	0.00%	14.0/sec	2135.0
/favicon.ico	859	37	18	71	2	678	0.00%	14.3/sec	80.2
/security/login.do	840	45	22	86	3	795	0.00%	14.4/sec	79.8
/j_spring_security_check	810	235	138	602	7	1987	0.00%	14.1/sec	82.4
/newspaper/list.do	790	286	168	726	11	2125	0.00%	14.2/sec	136.4
/advertisement/agent/listLink.do	1489	254	155	627	8	2353	0.00%	27.6/sec	209.1
/advertisement/agent/link.do	728	246	140	674	7	1950	0.14%	13.5/sec	185.1
/j_spring_security_logout	705	105	63	250	5	818	0.14%	13.5/sec	72.8
TOTAL	16346	100	35	262	1	2353	0.02%	265.5/sec	7461.2

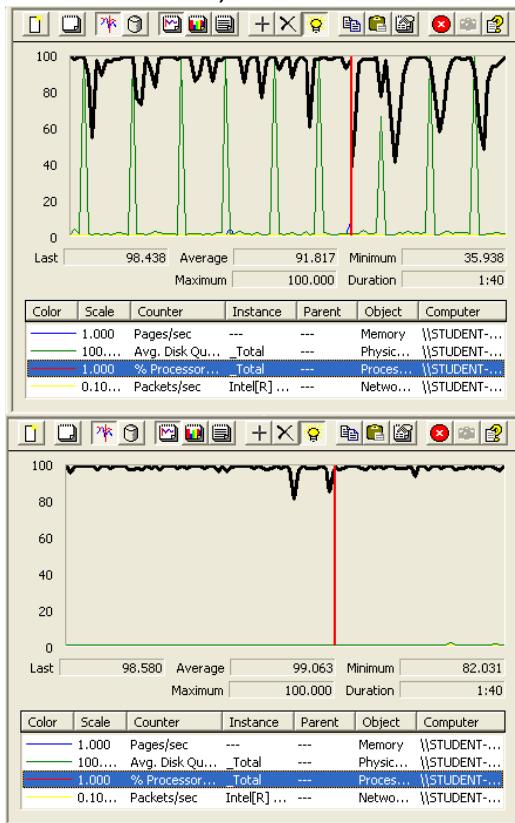
Include group name in label?  Save Table Data  Save Table Header

En este caso vemos como a la hora guardar el enlace y volver al listado, 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 15928.



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



## Eliminar enlace entre anuncio y periódico

En este caso de uso, un agente puede eliminar un enlace de un anuncio suyo con un periódico. Para ello, sigue la siguiente secuencia de pasos:

Perfil > Todos los periódicos

El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

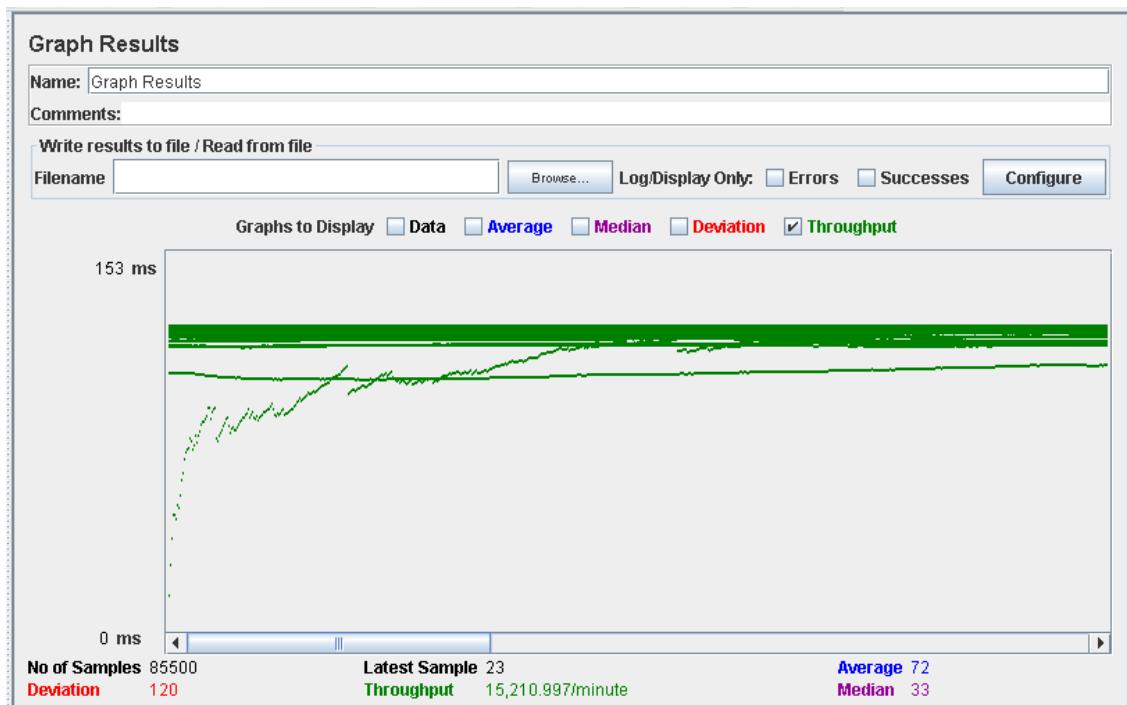
Aggregate Report									
Name: Aggregate Report Comments: Write results to file / Read from file									
Filename				Browse...	Log/Display Only:		<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>
/	# Samples	Average	Median	90% Line	Min	Max	Error %	Throu...	KB/sec
/styles/style.css	13500	31	20	61	3	1384	0.00%	40.0...	213.7
/scripts/jquery.min.js	4500	29	19	58	1	695	0.00%	13.7...	21.5
/styles/bootstrap.min.css	4500	83	46	179	5	1976	0.00%	13.7...	1471.4
/styles/cookieconsent.min.css	4500	88	48	187	6	3368	0.00%	13.7...	1628.4
/scripts/polyfills.js	4500	31	19	63	1	1162	0.00%	13.7...	56.8
/scripts/bootstrap.min.js	4500	48	28	97	3	1374	0.00%	13.7...	500.7
/scripts/helpers.js	4500	29	18	59	2	963	0.00%	13.7...	103.4
/scripts/cookieconsent.js	4500	29	19	57	1	6172	0.00%	13.7...	31.4
/images/newspaper-1.jpg	4500	42	25	82	2	1607	0.00%	13.7...	471.2
/security/login.do	4500	99	50	225	6	3152	0.00%	13.7...	2090.5
/j_spring_security_check	4500	32	21	60	4	872	0.00%	13.7...	76.1
/newspaper/list.do	4500	145	97	321	7	2025	0.00%	13.7...	80.3
/advertisement/agent/listUnlink.do	4500	179	127	375	11	2120	0.00%	13.7...	131.3
/advertisement/agent/unlink.do	9000	126	85	267	6	3090	0.00%	27.3...	155.7
/j_spring_security_logout	4500	125	85	256	7	1854	0.00%	13.7...	187.6
TOTAL	85500	72	33	169	1	6172	0.00%	253.5...	7099.2

Include group name in label?   Save Table Header

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 15210 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 155 usuarios ejecutando nuevamente 30 acciones cada uno.

**Aggregate Report**

Name: Aggregate Report  
Comments:

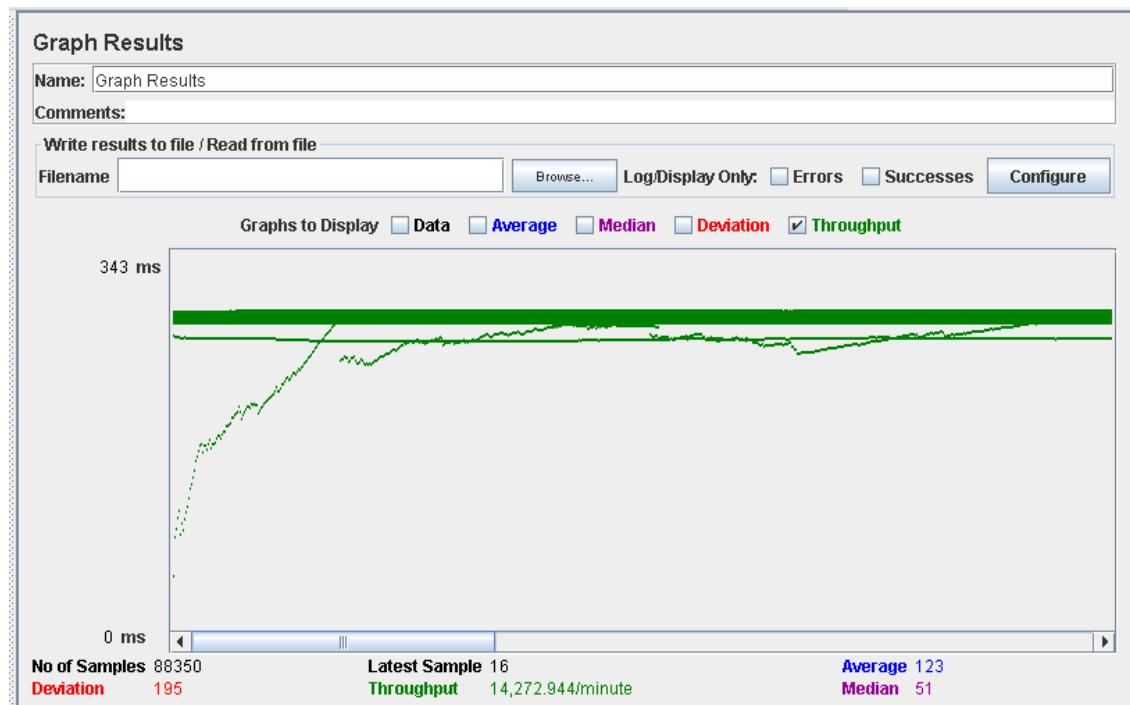
Write results to file / Read from file

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13950	52	31	108	3	2248	0.00%	37.6/...	200.6
/styles/style.css	4650	48	28	104	1	1259	0.00%	12.8/...	20.1
/scripts/jquery.min.js	4650	124	62	314	4	3734	0.00%	12.8/...	1374.2
/styles/bootstrap.min.css	4650	131	65	338	5	2128	0.00%	12.8/...	1520.9
/styles/cookieconsent.min.css	4650	54	30	113	2	1615	0.00%	12.8/...	53.1
/scripts/bootstrap.min.js	4650	72	40	168	2	1657	0.00%	12.8/...	467.7
/scripts/polyfills.js	4650	49	28	102	1	1287	0.00%	12.8/...	96.6
/scripts/helpers.js	4650	48	28	103	1	1001	0.00%	12.8/...	29.3
/scripts/cookieconsent.js	4650	67	37	151	2	1298	0.00%	12.8/...	440.2
/images/newspaper-1.jpg	4650	153	68	417	6	1975	0.00%	12.8/...	1951.9
/security/login.do	4650	52	29	111	3	1374	0.00%	12.8/...	71.1
/j_spring_security_check	4650	267	182	614	6	3503	0.00%	12.8/...	75.1
/newspaper/list.do	4650	292	198	679	11	3384	0.00%	12.8/...	122.9
/advertisement/agent/listUnlink.do	9300	236	155	553	6	4669	0.00%	25.6/...	145.7
/advertisement/agent/unlink.do	4650	235	153	554	6	2919	0.00%	12.8/...	175.7
/j_spring_security_logout	4650	109	66	237	5	1833	0.00%	12.8/...	69.3
TOTAL	88350	123	51	327	1	4669	0.00%	237.9/...	6661.6

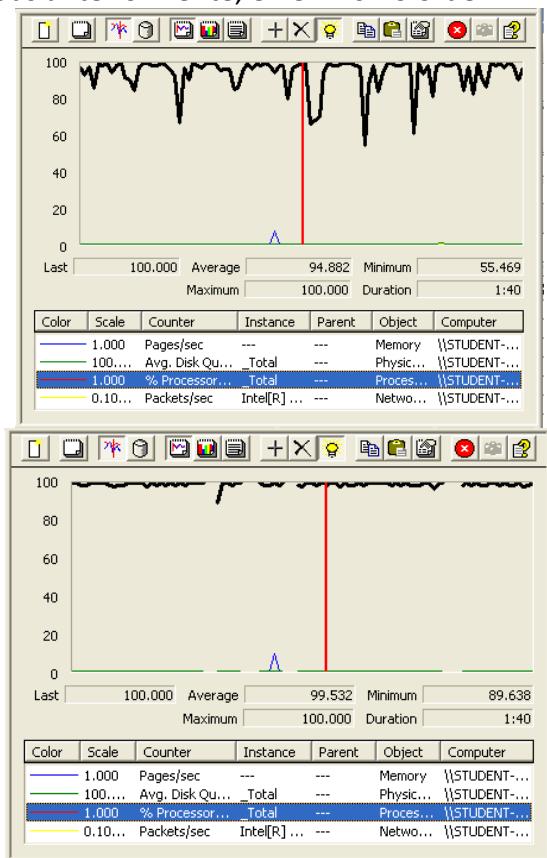
Include group name in label?   Save Table Header

En este caso vemos como a la hora de eliminar el enlace y volver al listado, 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 14272.



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



## Listar los periódicos y mostrar sus artículos (sin autenticarse)

En este caso de uso un usuario no autenticado lista los periódicos y al acceder a su display se muestran sus artículos.

Menú Principal > Todos los periódicos publicados > Desplegar un periódico

El límite de este caso de uso se sitúa en 170 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5100	24	15	46	3	881	0.00%	29.0/sec	148.4
/styles/cookiecons...	5100	20	12	39	1	629	0.00%	29.0/sec	120.1
/styles/style.css	5100	20	12	40	1	630	0.00%	29.0/sec	45.5
/scripts/jquery.min.js	5100	47	35	90	4	895	0.00%	29.0/sec	3110.7
/scripts/bootstrap.m...	5100	50	38	97	4	872	0.00%	29.0/sec	3442.6
/scripts/helpers.js	5100	23	13	50	1	786	0.00%	29.0/sec	66.4
/scripts/cookiecon...	5100	28	18	52	2	739	0.00%	29.0/sec	996.3
/scripts/bootstrap....	5100	28	19	54	2	649	0.00%	29.0/sec	1058.8
/scripts/polyfills.js	5100	20	12	40	1	867	0.00%	29.0/sec	218.7
/images/newspap...	5100	55	41	111	5	1849	0.00%	29.0/sec	4419.0
/newspaper/list.do	5100	231	103	594	6	3941	0.00%	29.0/sec	190.8
/newspaper/displa...	5100	257	119	837	6	4335	0.00%	29.0/sec	214.2
/welcome/index.do	5100	23	14	44	2	587	0.00%	29.0/sec	148.1
TOTAL	66300	64	22	119	1	4335	0.00%	367.0/sec	13800.0

Como podemos observar a la hora de listar los periódicos o desplegar este el número de medio de tiempo va aumentando acercándose a un segundo que más de esto es inadmisible, pero con esta cantidad de usuario aún está dentro de los límites.

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 22021 veces por minuto.



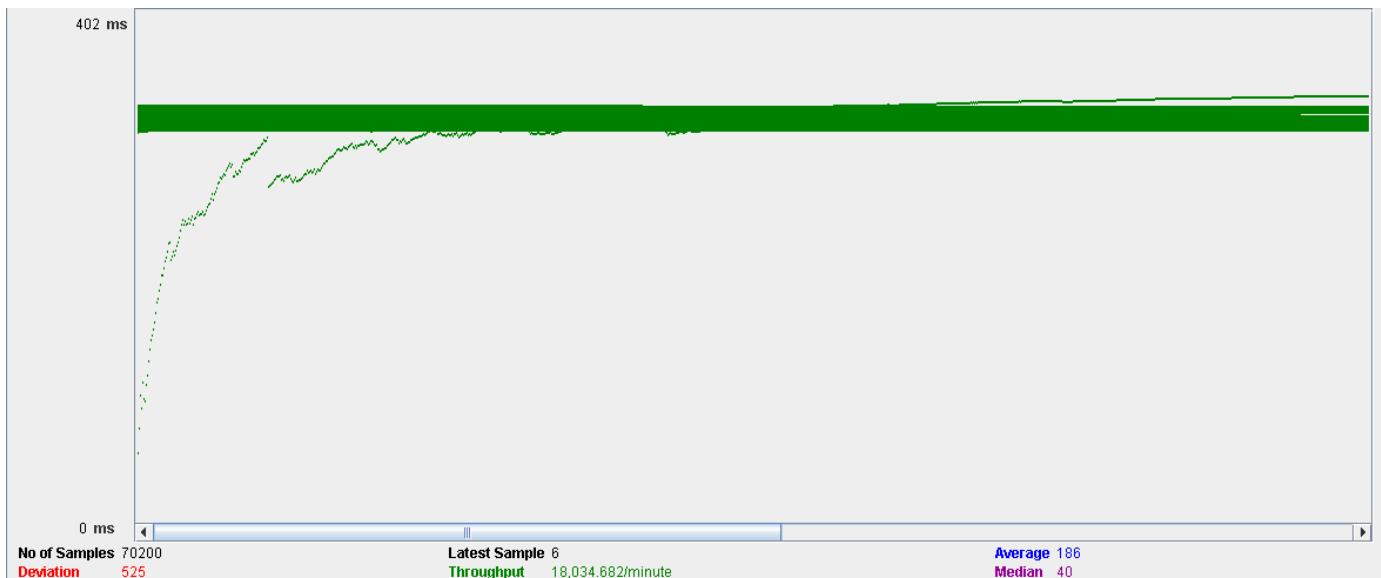
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 180 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5400	59	28	123	2	1414	0.00%	23.6/sec	120.8
/styles/cookiecons...	5400	52	23	108	1	1359	0.00%	23.6/sec	97.8
/styles/style.css	5400	53	23	107	1	1497	0.00%	23.6/sec	37.0
/scripts/jquery.min.js	5400	97	55	215	4	1508	0.00%	23.6/sec	2531.0
/scripts/bootstrap.m...	5400	103	55	238	3	1423	0.00%	23.6/sec	2801.3
/scripts/helpers.js	5400	57	23	120	1	1500	0.00%	23.6/sec	54.1
/scripts/cookiecon...	5400	63	31	131	2	1501	0.00%	23.6/sec	810.7
/scripts/bootstrap....	5400	62	31	131	2	1352	0.00%	23.6/sec	861.6
/scripts/polyfills.js	5400	52	21	104	1	1398	0.00%	23.6/sec	178.0
/images/newspap...	5400	122	69	279	6	1673	0.00%	23.6/sec	3593.9
/newspaper/list.do	5400	800	359	2081	7	16476	0.00%	23.6/sec	155.3
/newspaper/display...	5400	825	395	2177	6	15375	0.00%	23.6/sec	174.3
/welcome/index.do	5400	74	27	175	2	1496	0.00%	23.6/sec	120.5
TOTAL	70200	186	40	399	1	16476	0.00%	300.6/sec	11301.8

En este caso vemos al acceder al listado o al display, el sistema tarda de media en el 90% de las ocasiones más de dos segundos, 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 18034.



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



## Buscar un periódico publicado mediante una keyword (sin autenticarse)

En este caso de uso un usuario no autenticado lista los periódicos y usamos el buscador para buscar todos los publicados que contengan una palabra.

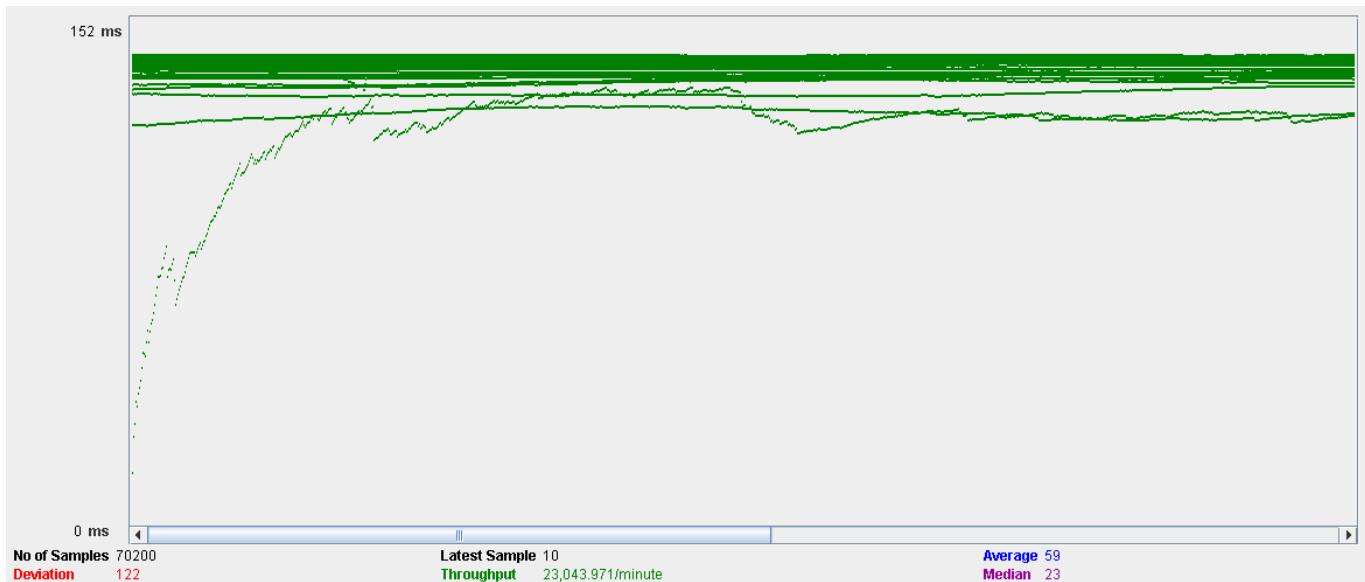
Menú Principal > Todos los periódicos > Rellenar campo del buscador y dar a buscar

El límite de este caso de uso se sitúa en 180 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5400	31	16	61	3	1153	0.00%	30.3/sec	155.2
/styles/cookiecons...	5400	23	13	46	1	942	0.00%	30.3/sec	125.6
/scripts/jquery.min.js	5400	50	37	94	4	1115	0.00%	30.3/sec	3250.9
/styles/style.css	5400	25	13	50	1	783	0.00%	30.3/sec	47.6
/scripts/cookiecon...	5400	32	19	62	2	1063	0.00%	30.3/sec	1041.0
/scripts/polyfills.js	5400	22	13	46	2	797	0.00%	30.3/sec	228.5
/styles/bootstrap.m...	5400	52	37	101	4	1139	0.00%	30.3/sec	3597.8
/scripts/helpers.js	5400	23	12	47	1	699	0.00%	30.3/sec	69.4
/scripts/bootstrap...	5400	30	19	59	3	851	0.00%	30.3/sec	1106.4
/images/newspap...	5400	60	45	119	4	686	0.00%	30.3/sec	4617.5
/newspaperlist.do	5400	197	113	493	7	2001	0.00%	30.3/sec	199.6
/newspaperlistSe...	5400	198	103	515	5	2191	0.00%	30.3/sec	188.4
/welcome/index.do	5400	29	15	58	2	998	0.00%	30.3/sec	154.8
TOTAL	70200	59	23	130	1	2191	0.00%	384.1/sec	14406.1

Como podemos observar a la hora de listar los periódicos o desplegar este el número de medio de el tiempo es de aproximadamente medio segundos lo cual son valores bastante buenos ya que está lejos de superar un segundo.

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 23043 veces por minuto.



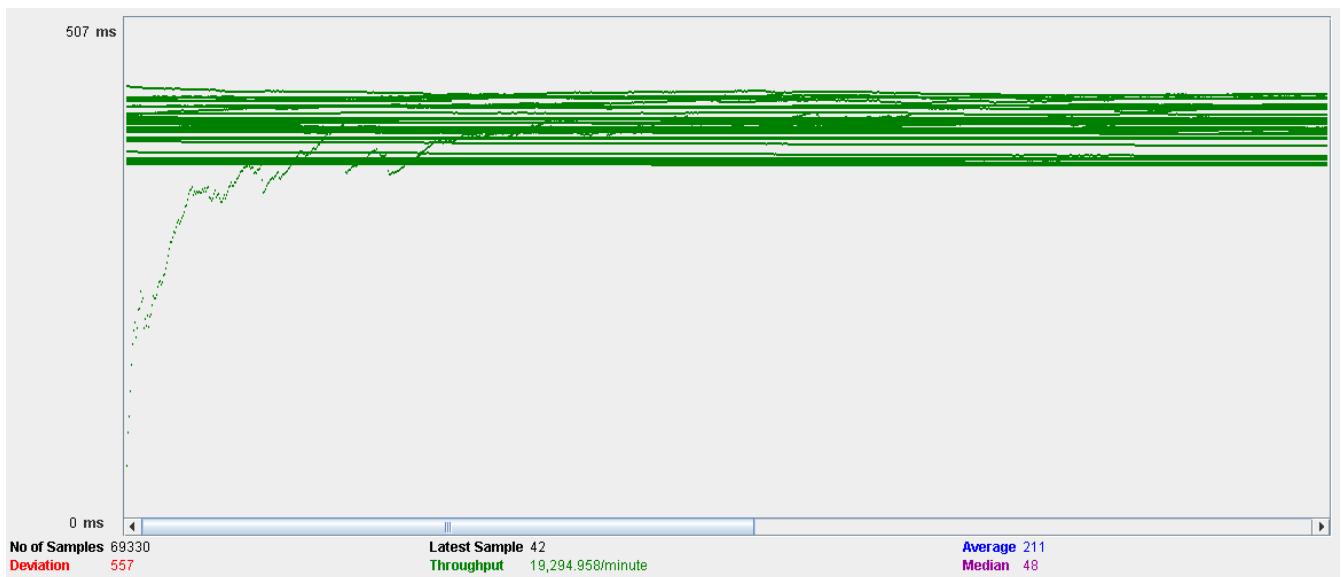
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 190 usuarios ejecutando nuevamente 30 acciones cada uno.

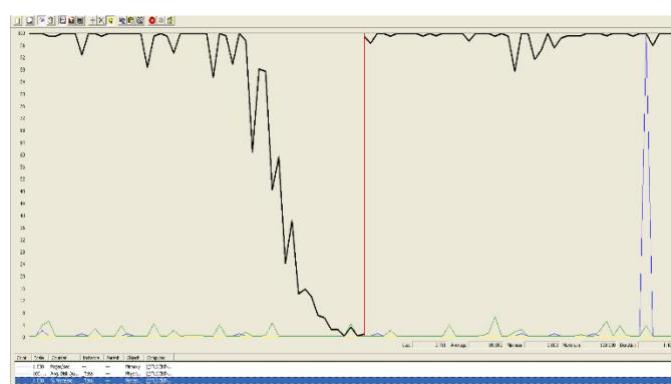
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	5363	93	32	229	3	1905	0.00%	24.9/sec	127.4
/styles/cookiecons...	5362	69	26	151	1	1988	0.00%	24.9/sec	103.1
/scripts/jquery.min.js	5362	121	66	249	5	1911	0.00%	24.9/sec	2688.6
/styles/style.css	5362	83	26	197	1	2515	0.00%	24.9/sec	39.1
/scripts/cookiecon...	5361	92	38	212	3	1931	0.00%	24.9/sec	854.4
/scripts/polyfills.js	5360	75	27	166	1	2019	0.00%	24.9/sec	187.5
/styles/bootstrap.m...	5360	127	67	286	6	2081	0.00%	24.9/sec	2952.0
/scripts/helpers.js	5360	86	25	211	1	2525	0.00%	24.9/sec	57.0
/scripts/bootstrap....	5359	98	38	228	3	1928	0.02%	24.9/sec	907.8
/images/newspap...	5357	162	84	384	6	2443	0.07%	25.0/sec	3809.6
/newspaper/list.do	5295	786	362	1966	9	14218	0.21%	24.9/sec	163.7
/newspaper/listSe...	5240	846	419	2002	7	14378	0.17%	24.8/sec	154.2
/welcome/index.do	5189	129	31	370	3	1890	0.00%	24.6/sec	125.7
TOTAL	69330	211	48	512	1	14378	0.04%	321.6/sec	12109.6

En este caso vemos al acceder a los listados, el sistema tarda de media en el 90% de las ocasiones más de dos segundos, 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 19294.



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



Listar los periódicos y mostrar sus artículos (autenticado)

En este caso de uso un usuario autenticado lista los periódicos y accede al display de uno para mostrar sus artículos.

Menú Principal > Entrar > Perfil > Todos los periódicos > Desplegar un periódico

El límite de este caso de uso se sitúa en 180 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	16200	31	20	55	2	809	0.00%	53.0/sec	283.0
/styles/bootstrap.m...	5400	66	46	140	4	854	0.00%	18.1/sec	2148.9
/scripts/jquery.min.js	5400	64	42	138	4	870	0.00%	18.1/sec	1941.7
/styles/style.css	5400	32	18	66	1	688	0.00%	18.1/sec	28.5
/scripts/helpers.js	5400	30	18	62	1	578	0.00%	18.1/sec	41.5
/styles/cookiecons...	5400	29	18	57	1	809	0.00%	18.1/sec	75.0
/scripts/bootstrap...	5400	38	25	76	2	715	0.00%	18.1/sec	660.9
/scripts/cookiecon...	5400	35	23	72	2	614	0.00%	18.1/sec	621.8
/scripts/polyfills.js	5400	29	17	57	1	610	0.00%	18.1/sec	136.5
/images/newspap...	5400	80	53	174	5	1036	0.00%	18.1/sec	2758.9
/security/login.do	5400	37	21	66	3	1343	0.00%	18.1/sec	100.6
/j_spring_security....	5400	381	212	944	7	5820	0.00%	18.1/sec	109.0
/newspaper/list.do	5400	368	211	874	9	7552	0.00%	18.1/sec	160.6
/newspaper/displa...	5400	369	206	909	6	7314	0.00%	18.1/sec	147.7
/j_spring_security....	5400	74	45	151	4	1384	0.00%	18.1/sec	97.8
TOTAL	91800	101	31	226	1	7552	0.00%	300.3/sec	9091.0

Como podemos observar, listar los periódicos y desplegar uno, tarda cerca de 1 segundo, lo que indica que nuestra web comienza a verse afectada por la cantidad de usuarios concurrentes.

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 18019 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 190 usuarios ejecutando nuevamente 30 acciones cada uno.

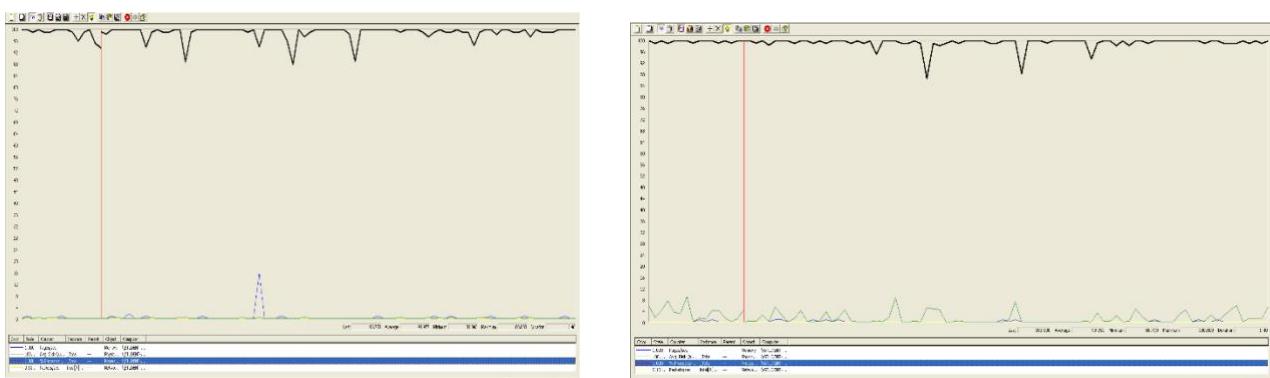
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	17100	44	23	86	2	1061	0.00%	53.2/sec	284.2
/styles/bootstrap.m...	5700	82	52	178	3	906	0.00%	18.2/sec	2156.6
/scripts/jquery.min.js	5700	80	49	175	4	1052	0.00%	18.2/sec	1948.7
/styles/style.css	5700	48	21	110	1	1070	0.00%	18.2/sec	28.6
/scripts/helpers.js	5700	43	21	93	1	875	0.00%	18.2/sec	41.6
/scripts/cookiecons...	5700	41	21	86	1	1050	0.00%	18.2/sec	75.3
/scripts/bootstrap...	5700	52	30	108	3	1044	0.00%	18.2/sec	663.3
/scripts/cookiecon...	5700	51	29	107	3	718	0.00%	18.2/sec	624.1
/scripts/polyfills.js	5700	43	21	90	1	980	0.00%	18.2/sec	137.0
/images/newspaper...	5700	98	63	221	6	1034	0.00%	18.2/sec	2771.5
/security/login.do	5700	49	24	112	3	1312	0.00%	18.2/sec	101.0
/j_spring_security...	5700	470	281	1102	6	5390	0.00%	18.2/sec	109.4
/newspaper/list.do	5700	453	263	1072	10	5282	0.00%	18.2/sec	161.1
/newspaper/display...	5700	461	252	1113	6	5685	0.00%	18.2/sec	148.2
/j_spring_security...	5700	100	54	254	3	1177	0.00%	18.2/sec	98.2
TOTAL	96900	129	37	310	1	5685	0.00%	301.6/sec	9130.3

En este caso vemos al acceder a los listados, 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 18096.



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



## Buscar un periódico publicado mediante una keyword (autenticado)

En este caso de uso un usuario autenticado lista los periódicos busca los publicados mediante una palabra clave.

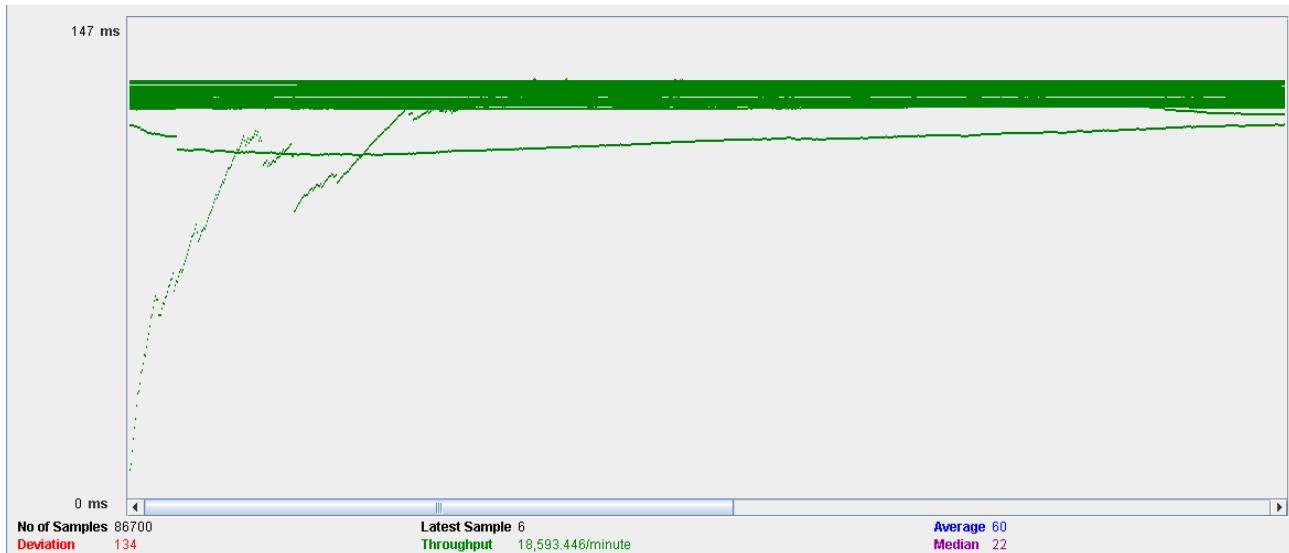
Menú Principal > Entrar > Perfil > Todos los periódicos > Rellenar campo del buscador y dar a buscar

El límite de este caso de uso se sitúa en 170 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15300	23	14	46	2	945	0.00%	54.7/sec	292.0
/scripts/jquery.min.js	5100	50	34	99	4	870	0.00%	18.9/sec	2024.4
/styles/cookiecons...	5100	24	13	49	1	693	0.00%	18.9/sec	78.2
/styles/bootstrap.m...	5100	53	35	108	4	909	0.00%	18.9/sec	2240.3
/styles/style.css	5100	24	12	47	1	741	0.00%	18.9/sec	29.6
/scripts/polyfills.js	5100	23	13	45	1	850	0.00%	18.9/sec	142.3
/scripts/bootstrap...	5100	30	19	59	2	752	0.00%	18.9/sec	689.0
/scripts/helpers.js	5100	21	12	42	1	752	0.00%	18.9/sec	43.2
/scripts/cookiecon...	5100	28	17	54	2	543	0.00%	18.9/sec	648.3
/images/newspap...	5100	54	35	114	4	742	0.00%	18.8/sec	2867.3
/security/login.do	5100	26	15	48	3	807	0.00%	18.8/sec	104.4
/j_spring_security...	5100	192	93	504	7	2596	0.00%	18.8/sec	113.2
/newspaper/list.do	5100	200	116	491	9	2812	0.00%	18.7/sec	166.0
/newspaper/listSe...	5100	178	80	465	7	2833	0.00%	18.7/sec	129.9
/j_spring_security...	5100	50	31	102	4	1173	0.00%	18.8/sec	101.3
TOTAL	86700	60	22	131	1	2833	0.00%	309.9/sec	9358.1

Como podemos observar, listar los periódicos y buscar, tarda menos de medio segundo lo que indica que con esta cantidad de usuarios el sistema funciona de una forma admisible.

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 18593 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 180 usuarios ejecutando nuevamente 30 acciones cada uno.

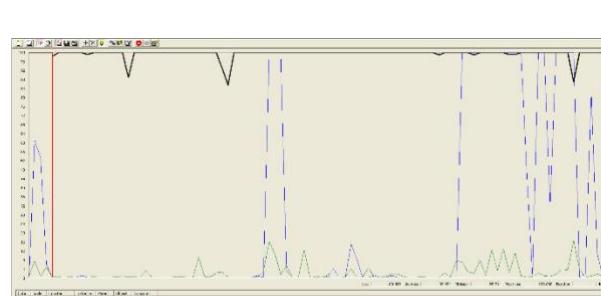
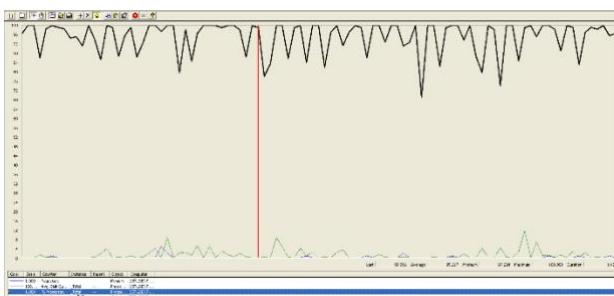
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	16200	112	33	276	3	2779	0.00%	38.9/sec	207.7
/scripts/jquery.min.js	5400	160	85	352	6	2719	0.00%	13.2/sec	1417.6
/styles/cookiecons...	5400	118	30	305	2	2345	0.00%	13.2/sec	54.8
/styles/bootstrap.m...	5400	180	86	423	7	2770	0.00%	13.2/sec	1568.8
/styles/style.css	5400	112	28	299	1	2349	0.00%	13.2/sec	20.8
/scripts/polyfills.js	5400	109	30	282	2	2554	0.00%	13.2/sec	99.6
/scripts/bootstrap....	5400	108	44	262	3	1790	0.00%	13.2/sec	482.4
/scripts/helpers.js	5400	95	28	241	2	2370	0.00%	13.2/sec	30.3
/scripts/cookiecon...	5400	121	41	305	3	2512	0.00%	13.2/sec	454.0
/images/newspap...	5400	197	102	455	7	3223	0.00%	13.2/sec	2013.9
/security/login.do	5400	133	34	378	5	2556	0.00%	13.2/sec	73.4
/j_spring_security....	5400	1039	612	2511	8	11081	0.00%	13.2/sec	79.6
/newspaper/list.do	5400	890	458	2175	15	11543	0.00%	13.2/sec	117.1
/newspaper/listSe...	5400	914	478	2300	8	10872	0.00%	13.2/sec	91.6
/j_spring_security....	5400	278	91	829	5	3014	0.00%	13.2/sec	71.4
TOTAL	91800	282	61	788	1	11543	0.00%	220.4/sec	6657.0

En este caso vemos al acceder a los listados, el sistema tarda de media en el 90% de las ocasiones más de dos 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 13226.



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



## Crear un periódico

En este caso de uso un usuario autenticado como user crea un periódico

Menú Principal > Entrar > Usuario > Mis periódicos > Crear periódico

El límite de este caso de uso se sitúa en 130 usuarios concurrentes ejecutando 30

veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	11700	13	9	24	2	453	0.00%	37.6/sec	200.7
/scripts/jquery.min.js	3900	27	20	57	4	355	0.00%	12.9/sec	1385.7
/styles/cookieconsent.mi...	3900	10	8	21	1	374	0.00%	12.9/sec	53.5
/styles/style.css	3900	10	8	21	1	369	0.00%	12.9/sec	20.2
/scripts/helpers.js	3900	10	8	21	1	277	0.00%	12.9/sec	29.5
/styles/bootstrap.min.css	3900	28	20	58	4	358	0.00%	12.9/sec	1533.4
/scripts/cookieconsent.js	3900	15	11	30	2	386	0.00%	12.9/sec	443.8
/scripts/polyfills.js	3900	11	8	22	1	363	0.00%	12.9/sec	97.4
/scripts/bootstrap.min.js	3900	15	12	31	2	366	0.00%	12.9/sec	471.7
/images/newspaper-1.jpg	3900	32	22	66	6	429	0.00%	12.9/sec	1968.7
/security/login.do	3900	14	10	25	3	489	0.00%	12.9/sec	71.7
jj_spring_security_check	3900	72	38	157	5	1293	0.00%	12.9/sec	77.6
/newspaper/userlist.do	7800	110	66	235	8	2360	0.00%	25.6/sec	1894.4
/newspaper/usercreate....	3900	83	43	178	5	2612	0.00%	12.9/sec	115.4
/newspaper/useredit.do	3900	228	148	495	18	2948	0.00%	12.9/sec	960.2
jj_spring_security_logout	3900	23	19	43	4	360	0.00%	12.9/sec	89.7
TOTAL	74100	44	16	98	1	2948	0.00%	238.2/sec	9136.9

Como podemos observar, listar los periódicos y crear uno, tarda menos de medio segundo lo que indica que con esta cantidad de usuarios el sistema funciona de una forma admisible.

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 14291 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 140 usuarios ejecutando nuevamente 30 acciones cada uno.

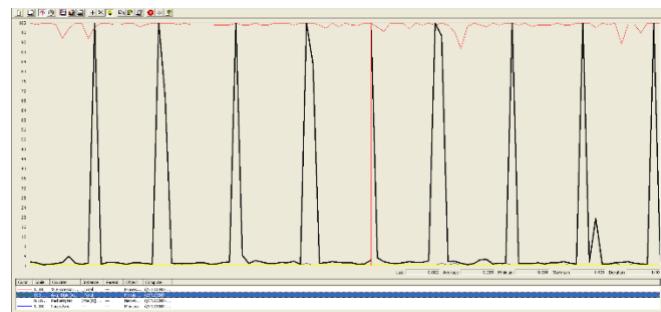
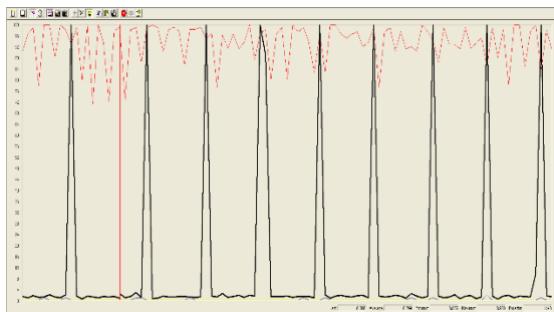
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	8511	19	15	35	2	305	0.01%	37.2/sec	198.8
/scripts/jquery.min.js	2900	38	29	82	4	324	0.03%	12.7/sec	1360.7
/styles/cookieconsent.mi...	2899	15	12	30	2	483	0.00%	12.7/sec	52.8
/styles/style.css	2899	15	12	31	1	296	0.00%	12.7/sec	20.0
/scripts/helpers.js	2898	15	12	30	1	324	0.00%	12.7/sec	29.1
/scripts/bootstrap.min.css	2898	40	29	87	4	442	0.00%	12.7/sec	1511.3
/scripts/cookieconsent.js	2898	21	16	42	2	321	0.00%	12.7/sec	437.3
/scripts/polyfills.js	2898	15	12	31	2	336	0.00%	12.7/sec	96.0
/scripts/bootstrap.min.js	2898	23	17	46	3	365	0.00%	12.7/sec	464.8
/images/newspaper-1.jpg	2897	46	33	102	5	377	0.00%	12.8/sec	1943.7
/security/login.do	2874	19	16	35	3	333	0.03%	12.7/sec	70.7
/j_spring_security_check	2847	275	114	681	6	4782	0.00%	12.6/sec	75.9
/newspaper/user/list.do	5618	320	190	747	7	4374	0.07%	25.2/sec	1407.8
/newspaper/user/create....	2814	256	127	650	6	3708	0.04%	12.7/sec	113.5
/newspaper/user/edit.do	2790	629	438	1425	18	5537	0.00%	12.7/sec	709.0
/j_spring_security_logout	2765	39	30	63	5	1872	0.07%	12.6/sec	68.0
Error	1	0	0	0	0	0	100.00%	∞/sec	0
TOTAL	54305	111	24	267	0	5537	0.02%	.1/hour	.0

En este caso que cuando creas un periódico y vuelves al listado de tus periódicos, el sistema tarda de media en el 90% de las ocasiones más de dos 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 14006.



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



## Publicar un periódico

En este caso de uso un usuario autenticado como user publica uno de sus periódicos  
Menú Principal > Entrar [user1] > Usuario > Mis periódicos > Editar fecha

El límite de este caso de uso se sitúa en 130 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	11700	18	13	33	3	1271	0.00%	35.3/sec	188.1
/styles/style.css	3900	14	11	28	1	442	0.00%	12.1/sec	18.9
/scripts/jquery.min.js	3900	41	29	84	5	616	0.00%	12.1/sec	1297.6
/styles/bootstrap.min.css	3900	41	29	81	5	1337	0.00%	12.1/sec	1436.0
/styles/cookieconsent.mi...	3900	14	10	28	1	178	0.00%	12.1/sec	50.1
/scripts/polyfills.js	3900	14	11	28	2	364	0.00%	12.1/sec	91.2
/scripts/helpers.js	3900	13	10	27	2	379	0.00%	12.1/sec	27.6
/scripts/bootstrap.min.js	3900	23	16	45	3	595	0.00%	12.1/sec	441.5
/scripts/cookieconsent.js	3900	21	15	39	2	1283	0.00%	12.1/sec	415.4
/images/newspaper-1.jpg	3900	51	35	103	6	764	0.00%	12.1/sec	1843.7
/security/login.do	3900	19	14	35	4	591	0.00%	12.1/sec	67.1
jl_spring_security_check	3900	183	69	424	8	5443	0.00%	12.1/sec	72.7
/newspaperUserList.do	7800	215	100	479	9	4541	0.00%	24.0/sec	180.5
/newspaperUserEditDat...	3900	154	51	346	6	5205	0.00%	12.1/sec	166.9
/newspaperUserEdit.do	3900	190	71	457	8	5150	0.00%	12.1/sec	92.7
jl_spring_security_logout	3900	33	25	61	5	864	0.00%	12.1/sec	65.3
TOTAL	74100	68	21	125	1	5443	0.00%	223.3/sec	6276.8

Como podemos observar, al editar una fecha y volver al listado, tarda menos de un segundo pero está cerca de uno, por lo que está claro que 130 usuarios es el límite a partir del cual la aplicación comenzará a tardar demasiado

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 13399 veces por minuto.



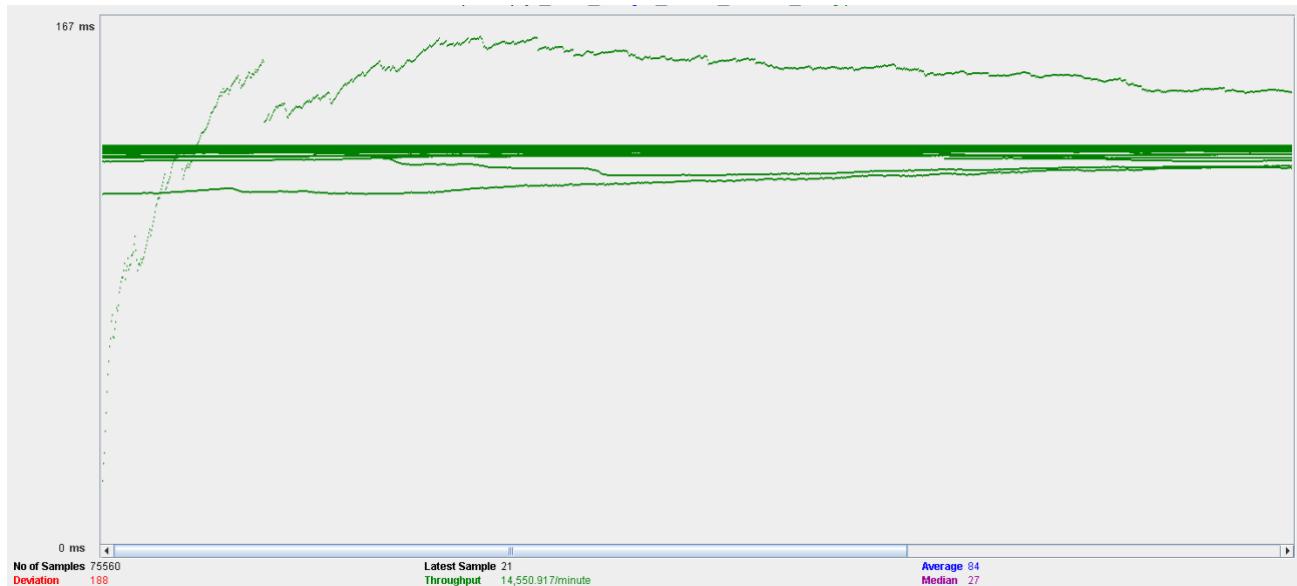
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 140 usuarios ejecutando nuevamente 30 acciones cada uno.

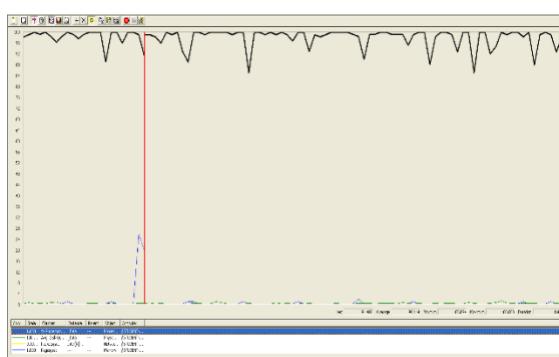
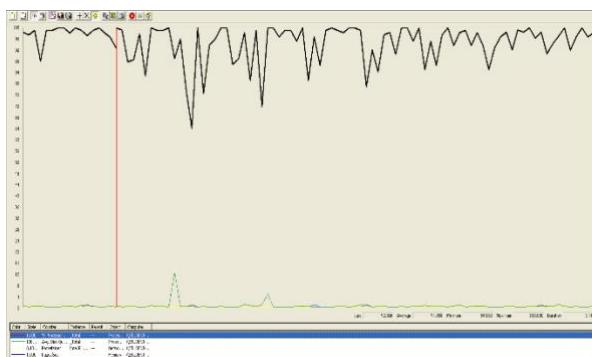
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/styles/style.css	11868	22	17	39	3	781	0.00%	38.1/sec	203.3
/scripts/jquery.min.js	4020	18	14	35	1	762	0.00%	12.9/sec	20.2
/styles/bootstrap.min.css	4020	48	33	103	6	587	0.00%	12.9/sec	1384.1
/styles/cookieconsent.mi...	4019	18	14	35	6	958	0.00%	12.9/sec	1531.8
/scripts/polyfills.js	4019	18	14	34	2	359	0.00%	12.9/sec	53.4
/scripts/helpers.js	4019	17	13	32	2	769	0.00%	12.9/sec	97.2
/scripts/bootstrap.min.js	4018	27	20	54	3	528	0.00%	12.9/sec	470.8
/scripts/cookieconsent.js	4017	25	19	49	3	468	0.00%	12.9/sec	442.9
/images/newspaper-1.jpg	4016	57	38	125	8	550	0.00%	12.9/sec	1972.5
/security/login.do	3991	23	18	41	3	472	0.00%	12.9/sec	71.7
/_spring_security_check	3965	227	123	570	8	3493	0.00%	12.9/sec	77.3
/newspaper/userlist.do	7855	268	168	621	10	4017	0.00%	25.7/sec	193.6
/newspaper/usereditDo...	3923	211	90	587	6	3048	0.00%	12.9/sec	178.1
/newspaper/useredit.do	3907	233	125	600	8	3315	0.00%	12.9/sec	98.6
/_spring_security_logout	3883	41	34	76	3	422	0.00%	12.8/sec	69.3
TOTAL	75560	84	27	194	1	4017	0.00%	242.5/sec	6869.2

En este caso cuando editas una fecha y vuelves al listado, el sistema tarda de media en el 90% de las ocasiones más de 1,3 segundos, 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 14550.



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



## Borrar un periódico

En este caso de uso un usuario autenticado como admin borra un periódico.

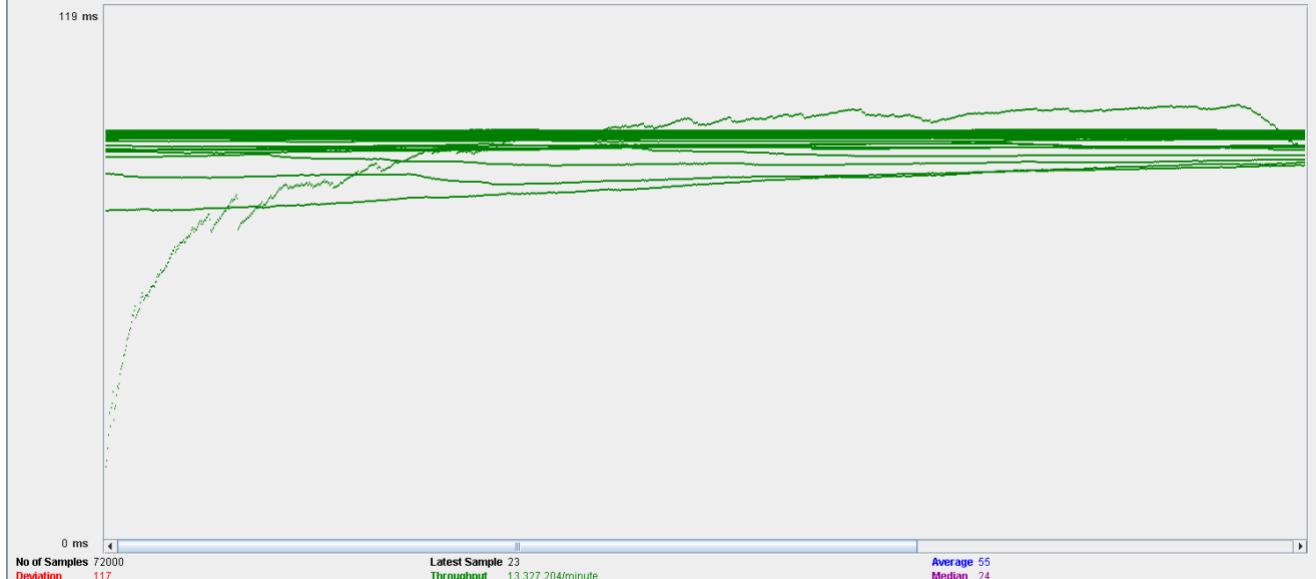
Menú Principal > Entrar [admin] > Perfil > Todos los periódicos > Desplegar > Borrar

El límite de este caso de uso se sitúa en 120 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	10800	20	15	37	3	512	0.00%	33.3/sec	181.9
/scripts/jquery.min.js	3600	43	29	89	5	444	0.00%	11.4/sec	1226.4
/styles/cookieconsent.mi...	3600	15	12	30	2	312	0.00%	11.4/sec	47.3
/styles/bootstrap.min.css	3600	42	29	90	6	736	0.00%	11.4/sec	1357.3
/styles/style.css	3600	15	11	29	2	518	0.00%	11.4/sec	17.9
/scripts/bootstrap.min.js	3600	23	16	48	3	582	0.00%	11.4/sec	417.4
/scripts/polyfills.js	3600	15	12	29	2	780	0.00%	11.4/sec	86.2
/scripts/helpers.js	3600	14	11	29	1	377	0.00%	11.4/sec	26.1
/scripts/cookieconsent.js	3600	23	16	45	3	728	0.00%	11.4/sec	392.8
/images/newspaper-1.jpg	3600	55	35	122	7	1143	0.00%	11.4/sec	1742.9
/security/login.do	3600	20	15	38	3	476	0.00%	11.4/sec	63.4
/j_spring_security_check	3600	120	73	254	6	1975	0.00%	11.4/sec	73.0
/newspaper/list.do	7200	165	104	336	10	2734	0.00%	22.6/sec	198.2
/newspaper/display.do	3600	109	54	242	6	3613	0.00%	11.4/sec	162.3
/Media@20131203@portad...	3600	51	45	71	26	295	0.00%	11.4/sec	753.1
/newspaper/administrat...	3600	129	63	270	7	2888	0.00%	11.4/sec	150.1
/j_spring_security_logout	3600	37	30	66	6	687	0.00%	11.4/sec	61.7
TOTAL	72000	55	24	119	1	3613	0.00%	222.1/sec	6763.0

Como podemos observar, al borrar un periódico y volver al listado de todos los periódicos, se tarda medio segundo que es un tiempo muy admisible por lo que con esta cantidad de usuarios el caso de uso funciona de forma efectiva

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 13327 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 130 usuarios ejecutando nuevamente 30 acciones cada uno.

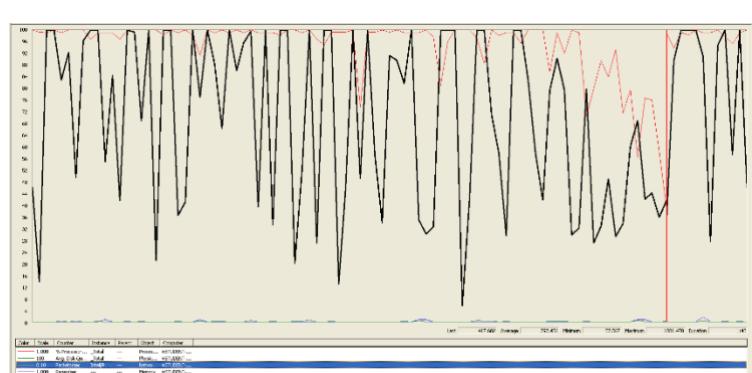
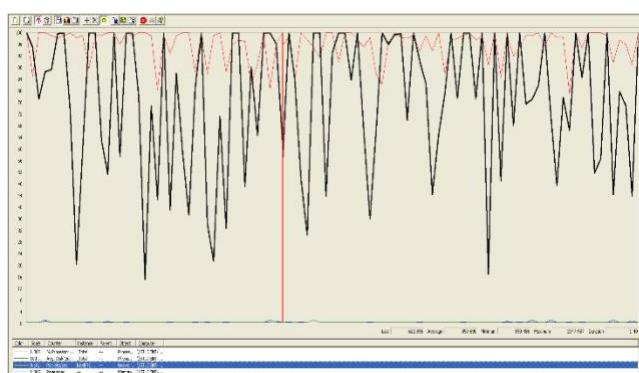
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	11663	24	18	41	3	1337	0.00%	29.3/sec	160.0
/scripts/jquery.min.js	3889	51	34	107	5	1319	0.00%	9.8/sec	1047.3
/styles/cookieconsent.mi...	3889	19	14	36	1	928	0.00%	9.8/sec	40.4
/styles/bootstrap.min.css	3889	49	34	103	4	783	0.00%	9.8/sec	1159.1
/styles/style.css	3889	19	13	34	1	1065	0.00%	9.8/sec	15.3
/scripts/bootstrap.min.js	3889	27	19	55	3	672	0.00%	9.8/sec	356.4
/scripts/polyfills.js	3889	17	13	34	2	630	0.00%	9.8/sec	73.6
/scripts/helpers.js	3889	17	13	34	1	934	0.00%	9.8/sec	22.3
/scripts/cookieconsent.js	3889	26	18	51	2	906	0.00%	9.8/sec	335.3
/images/newspaper-1.jpg	3889	62	41	134	7	1126	0.00%	9.8/sec	1498.0
/security/login.jsp	3889	24	19	44	3	793	0.00%	9.8/sec	54.2
/_spring_security_check	3889	194	107	430	8	4300	0.00%	9.8/sec	62.4
/newspaperlist.do	7774	242	136	563	10	4878	0.00%	19.8/sec	173.6
/newspaperdisplay.do	3888	155	70	359	6	3269	0.00%	10.0/sec	142.0
/Media/2013/203/portad...	3886	1368	51	337	27	81940	0.00%	10.0/sec	658.1
/newspaperadministrat...	3886	226	99	600	5	3863	0.00%	10.0/sec	131.2
/_spring_security_logout	3886	46	35	78	5	1127	0.00%	10.0/sec	53.9
TOTAL	77762	143	29	172	1	81940	0.00%	19.3/sec	5903.5

En este caso cuando borras un periódico y vuelves al listado de periódicos, el sistema tarda de media en el 90% de las ocasiones más de 1,1 segundos, 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 11473.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y fallos en 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.



## Desplegar un periódico (sin loguearse)

En este caso de uso un usuario no autenticado despliega un periódico.

Menú Principal > Todos los periódicos publicados > Desplegar

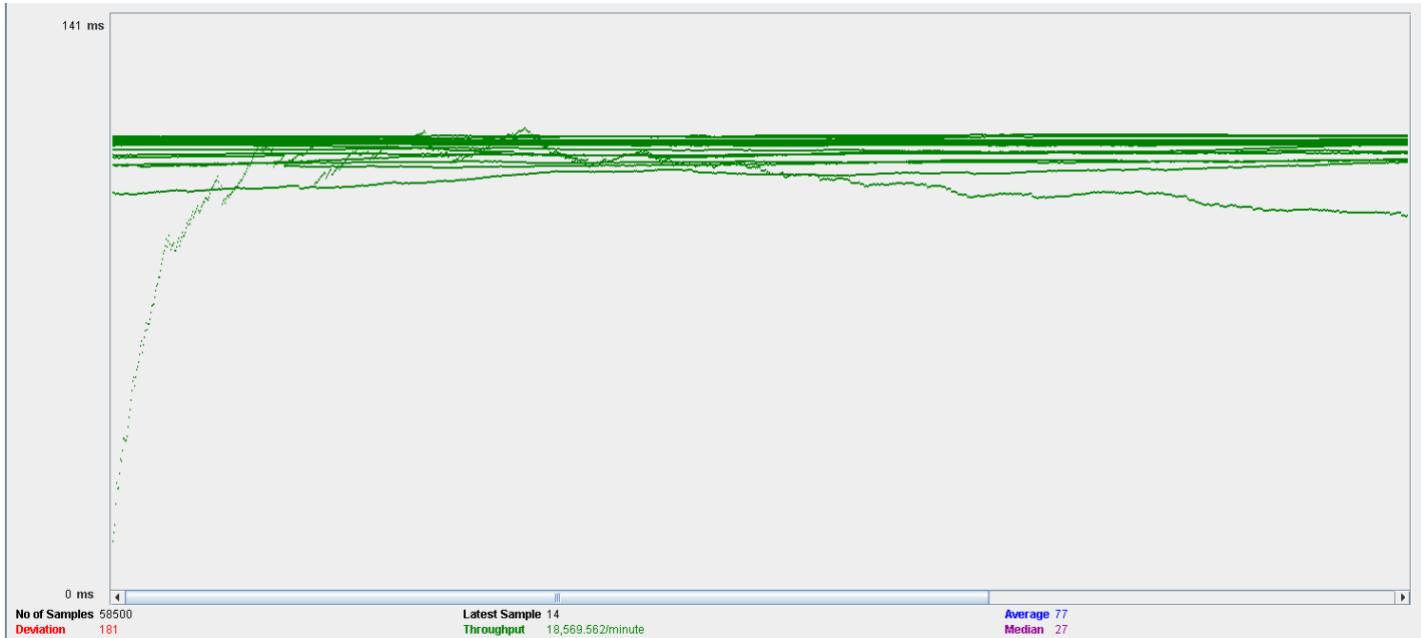
El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4500	30	18	55	3	981	0.00%	24.5/sec	125.4
/styles/style.css	4500	22	14	42	1	849	0.00%	24.5/sec	38.4
/scripts/jquery.min.js	4500	56	45	100	5	941	0.00%	24.5/sec	2627.9
/scripts/bootstrap.min.css	4500	58	45	103	5	2019	0.00%	24.5/sec	2908.3
/styles/cookieconsent.mi...	4500	24	14	48	1	610	0.00%	24.5/sec	101.5
/scripts/helpers.js	4500	24	14	47	1	972	0.00%	24.5/sec	56.0
/scripts/cookieconsent.js	4500	28	21	53	3	571	0.00%	24.5/sec	841.6
/scripts/bootstrap.min.js	4500	31	22	55	3	1298	0.00%	24.5/sec	894.4
/scripts/polyfills.js	4500	22	14	44	2	584	0.00%	24.5/sec	184.7
/images/newspaper-1.jpg	4500	68	54	127	6	713	0.00%	24.5/sec	3730.8
/newspaperlist.do	4500	286	147	746	10	2628	0.00%	24.5/sec	161.0
/newspaper/display.do	4500	325	176	854	10	2825	0.00%	24.4/sec	182.3
/welcome/index.do	4500	27	18	48	3	962	0.00%	24.4/sec	124.7
TOTAL	58500	77	27	150	1	2825	0.00%	309.5/sec	11638.2

Como podemos observar, al acceder al listado o al display de un periódico los tiempos son menores a un segundo, aunque ya que son próximos a uno quiere decir que 150 es cercano al límite de usuarios para que el caso de uso funcione correctamente.

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 13327 veces por minuto.



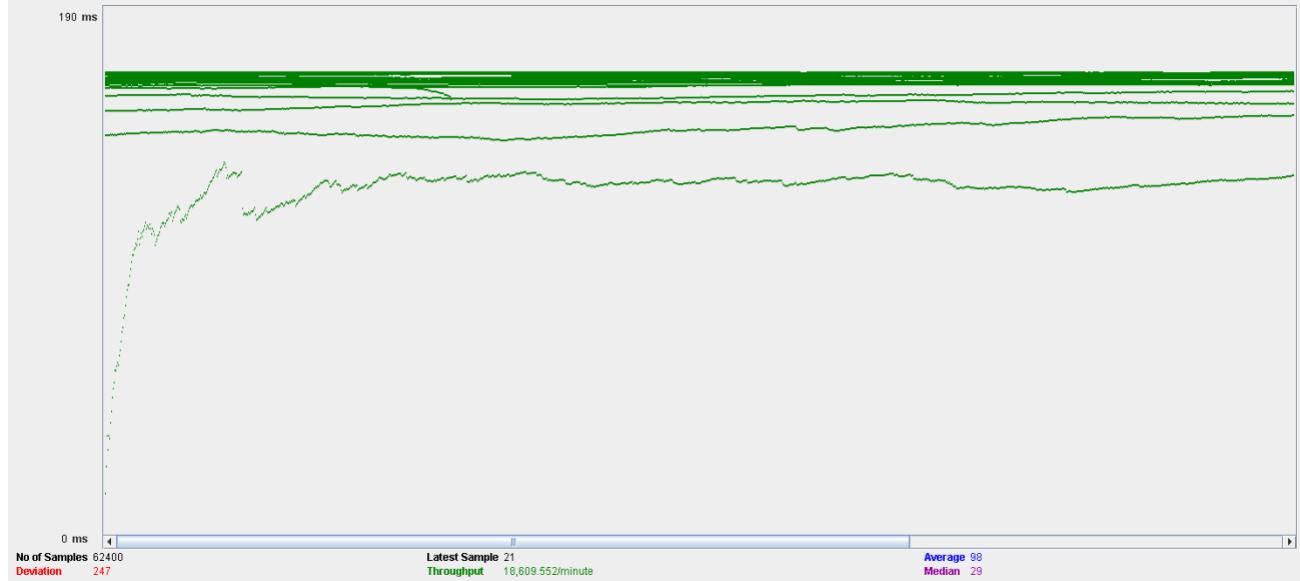
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 160 usuarios ejecutando nuevamente 30 acciones cada uno.

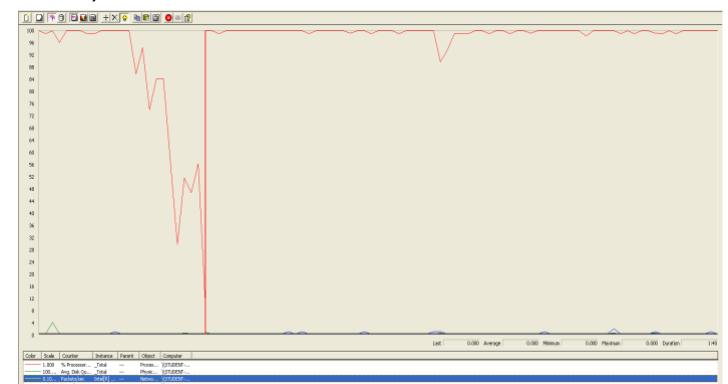
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4800	33	19	59	4	1332	0.00%	24.4/sec	125.0
/styles/style.css	4800	25	16	47	1	930	0.00%	24.4/sec	38.3
/scripts/jquery.min.js	4800	62	48	112	5	1298	0.00%	24.4/sec	2620.6
/scripts/bootstrap.min.css	4800	64	49	117	6	1916	0.00%	24.4/sec	2900.2
/styles/cookieconsent.mi...	4800	27	16	53	2	1048	0.00%	24.4/sec	101.2
/scripts/helpers.js	4800	25	16	49	2	817	0.00%	24.4/sec	55.9
/scripts/cookieconsent.js	4800	33	23	60	3	831	0.00%	24.4/sec	839.2
/scripts/bootstrap.min.js	4800	35	24	64	3	657	0.00%	24.4/sec	891.9
/scripts/polyfills.js	4800	25	16	49	2	631	0.00%	24.4/sec	184.2
/images/newspaper-1.jpg	4800	79	61	146	7	1123	0.00%	24.4/sec	3722.8
/newspaperlist.do	4800	389	202	979	7	4145	0.00%	24.4/sec	160.8
/newspaper/display.do	4800	447	239	1127	8	5510	0.00%	24.4/sec	182.7
/welcome/index.do	4800	32	20	56	3	679	0.00%	24.4/sec	124.9
TOTAL	62400	98	29	196	1	5510	0.00%	310.2/sec	11663.5

En este caso cuando accedes al display de un periódico, el sistema tarda de media en el 90% de las ocasiones más de 1,1 segundos, 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 18609.



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



## Desplegar un newspaper (logueado)

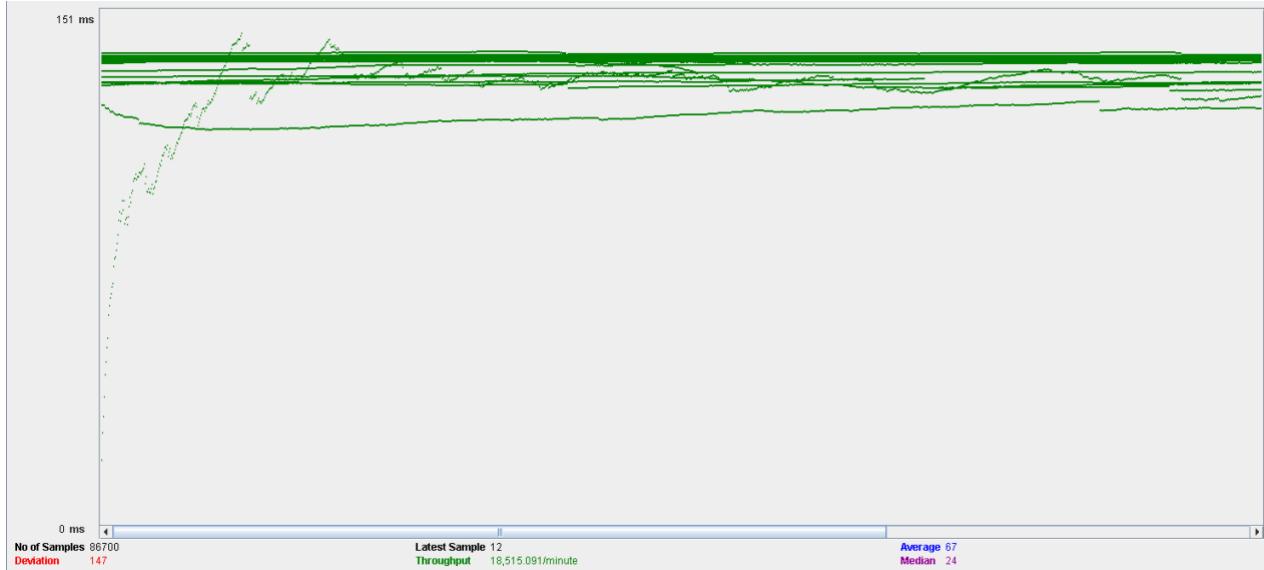
En este caso de uso un usuario autenticado despliega un periódico.  
Menú Principal > Entrar > Perfil > Todos los periódicos > Desplegar

El límite de este caso de uso se sitúa en 170 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15300	26	16	49	2	675	0.00%	54.5/sec	290.7
/scripts/jquery.min.js	5100	52	32	120	4	528	0.00%	18.7/sec	2004.3
/styles/bootstrap.min.css	5100	56	34	127	4	641	0.00%	18.7/sec	2218.1
/styles/cookieconsent.mi...	5100	29	15	57	1	732	0.00%	18.7/sec	77.4
/styles/style.css	5100	26	15	51	1	599	0.00%	18.7/sec	29.3
/scripts/bootstrap.min.js	5100	32	20	69	2	581	0.00%	18.7/sec	682.1
/scripts/cookieconsent.js	5100	32	19	66	2	619	0.00%	18.7/sec	641.8
/scripts/polyfills.js	5100	26	14	51	1	630	0.00%	18.7/sec	140.9
/scripts/helpers.js	5100	24	14	48	1	549	0.00%	18.7/sec	42.8
/images/newspaper-1.jpg	5100	56	32	134	5	803	0.00%	18.7/sec	2846.1
/security/login.do	5100	29	16	53	3	761	0.00%	18.7/sec	103.6
/j_spring_security_check	5100	221	114	559	4	3301	0.00%	18.7/sec	112.3
/newspaper/list.do	5100	205	118	489	8	3330	0.00%	18.7/sec	162.5
/newspaper/display.do	5100	221	123	534	6	3176	0.00%	18.7/sec	152.2
/j_spring_security_logout	5100	57	36	116	2	845	0.00%	18.7/sec	100.8
TOTAL	86700	67	24	154	1	3330	0.00%	308.6/sec	9338.0

Como podemos observar, al acceder al listado o al display de un periódico los tiempos son menores a un segundo, por lo que con esta cantidad de usuarios simultáneos el caso de uso es admisible.

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 13327 veces por minuto.



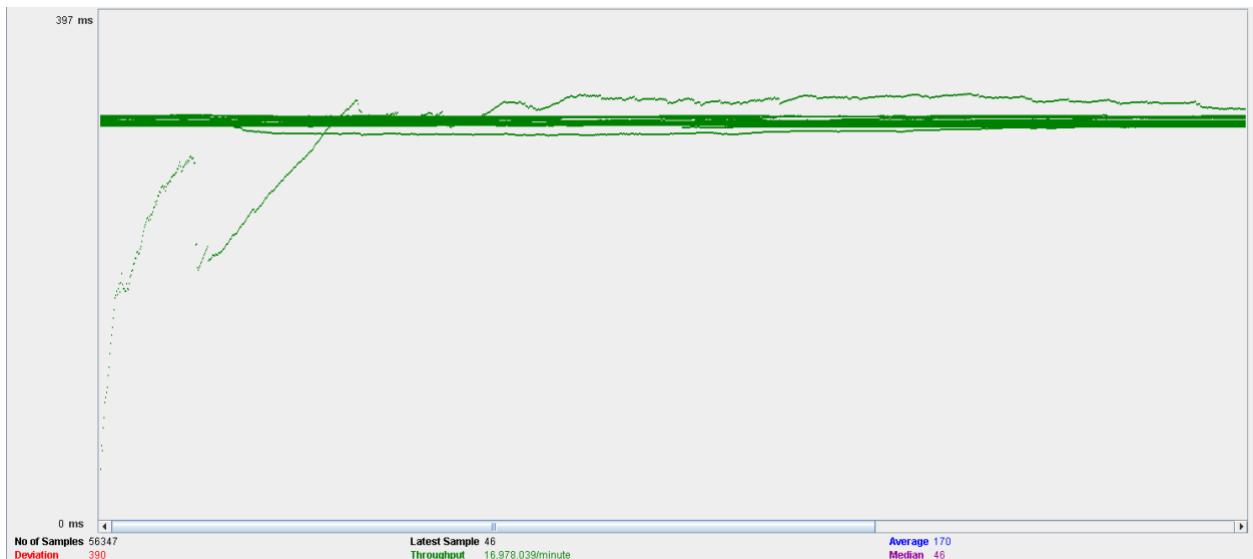
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 180 usuarios ejecutando nuevamente 30 acciones cada uno.

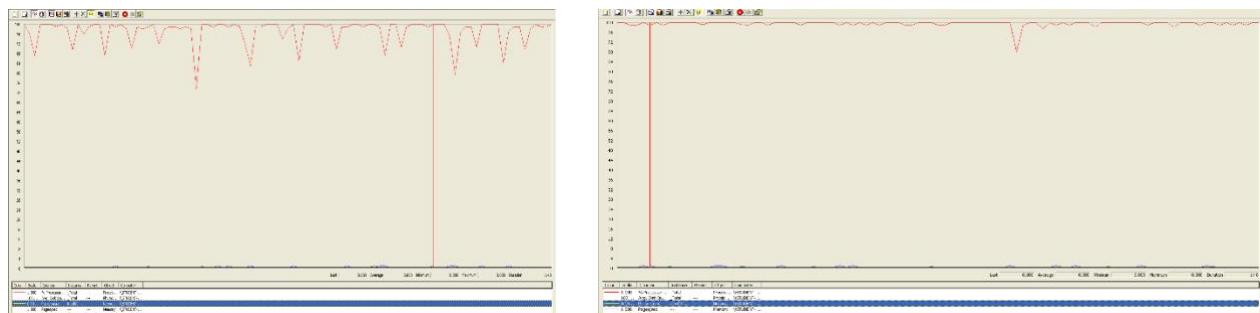
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9830	54	32	109	2	2392	0.00%	49.4/sec	263.8
/scripts/jquery.min.js	3362	92	56	208	4	1423	0.00%	16.9/sec	1813.2
/styles/bootstrap.min.css	3361	98	55	219	4	1965	0.00%	16.9/sec	2006.0
/styles/cookieconsent.mi...	3361	53	28	113	2	1592	0.00%	16.9/sec	70.1
/styles/style.css	3360	56	28	111	1	1576	0.00%	16.9/sec	26.6
/scripts/bootstrap.min.js	3360	62	35	138	3	2442	0.00%	16.9/sec	616.8
/scripts/cookieconsent.js	3360	60	34	125	3	1595	0.00%	16.9/sec	580.3
/scripts/polyfills.js	3360	48	25	101	1	1190	0.00%	16.9/sec	127.4
/scripts/helpers.js	3360	49	25	98	1	1609	0.00%	16.9/sec	38.7
/images/newspaper-1.jpg	3360	110	64	251	4	1339	0.00%	17.0/sec	2591.7
/security/login.do	3329	54	29	111	3	1105	0.00%	17.0/sec	94.3
/j_spring_security_check	3287	661	386	1561	9	6595	0.00%	16.8/sec	101.1
/newspaper/list.do	3259	638	365	1557	7	7140	0.03%	16.9/sec	147.0
/newspaper/display.do	3215	649	374	1595	8	7891	0.00%	16.8/sec	137.0
/j_spring_security_logout	3183	129	72	303	5	1523	0.03%	16.7/sec	90.0
TOTAL	56347	170	46	402	1	7891	0.00%	283.0/sec	8657.5

En este caso cuando accedes al display de un periódico y al listarlos, el sistema tarda de media en el 90% de las ocasiones más de 1,5 segundos, 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 16978.



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



## Listar los periódicos con palabras tabú

En este caso de uso un usuario logueado como admin lista los periódicos con palabras tabú.

Menú Principal > Entrar[admin] > Administrador > Periódicos Taboos

El límite de este caso de uso se sitúa en 180 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	16200	35	18	67	2	1070	0.00%	68.6/sec	374.9
/scripts/jquery.min.js	5400	65	43	131	5	1211	0.00%	23.5/sec	2515.9
/styles/cookieconsent.mi...	5400	35	17	78	1	1044	0.00%	23.5/sec	97.2
/styles/style.css	5400	36	17	82	1	1039	0.00%	23.5/sec	36.8
/styles/bootstrap.min.css	5400	64	44	128	5	1001	0.00%	23.5/sec	2784.5
/scripts/helpers.js	5400	35	17	75	1	972	0.00%	23.5/sec	53.7
/scripts/cookieconsent.js	5400	43	24	91	3	940	0.00%	23.5/sec	805.8
/scripts/bootstrap.min.js	5400	41	24	86	2	1175	0.00%	23.5/sec	856.4
/scripts/polyfills.js	5400	38	17	78	1	977	0.00%	23.5/sec	176.9
/images/newspaper-1.jpg	5400	86	53	187	6	1351	0.00%	23.5/sec	3573.6
/security/login.do	5400	39	19	83	3	968	0.00%	23.5/sec	130.4
/j_spring_security_check	5400	252	133	591	6	5606	0.00%	23.5/sec	150.2
/newspaper/administrat...	5400	210	111	495	6	3818	0.00%	23.5/sec	172.5
/j_spring_security_logout	5400	84	41	201	3	1298	0.00%	23.5/sec	126.8
TOTAL	86400	71	28	158	1	5606	0.00%	365.9/sec	11565.0

Como podemos observar, al acceder al listado el tiempo es de menos de medio segundos por lo que los resultados son muy positivos para este caso de uso y esta cantidad de usuarios simultáneos.

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 21955 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.

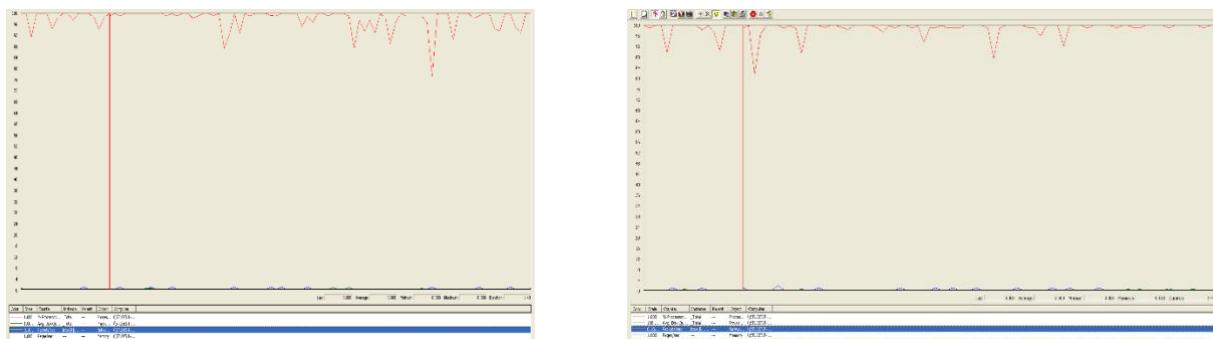
En este caso se ha utilizado 190 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	17009	42	20	95	3	916	0.00%	72.4/sec	395.7
/scripts/jquery.min.js	5696	75	50	157	5	991	0.00%	24.4/sec	2614.7
/styles/cookieconsent.mi...	5696	43	19	104	1	1171	0.00%	24.4/sec	101.0
/styles/style.css	5696	46	19	113	1	862	0.00%	24.4/sec	38.3
/styles/bootstrap.min.css	5696	75	50	159	4	937	0.00%	24.4/sec	2893.3
/scripts/helpers.js	5696	45	18	112	1	856	0.00%	24.4/sec	55.8
/scripts/cookieconsent.js	5696	52	26	125	2	808	0.00%	24.4/sec	837.3
/scripts/bootstrap.min.js	5696	50	27	114	3	906	0.00%	24.4/sec	889.8
/scripts/polyfills.js	5696	42	18	104	2	759	0.00%	24.4/sec	183.8
/Images/newspaper-1.jpg	5696	95	61	209	5	944	0.02%	24.4/sec	3716.7
/security/login.do	5691	47	19	106	3	937	0.00%	24.5/sec	136.2
/j_spring_security_check	5679	229	144	547	7	2393	0.00%	24.5/sec	157.0
/newspaper/administrat...	5668	193	120	465	7	2238	0.00%	24.7/sec	181.7
/j_spring_security_logout	5635	103	48	267	4	1113	0.02%	24.6/sec	133.0
TOTAL	90946	76	31	194	1	2393	0.00%	387.0/sec	12252.4

En este caso cuando al cargar los estilos de la página inicialmente, el sistema tarda de media en el 90% de las ocasiones más de 1,3 segundos, 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 23221.



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



## Poner a público o privado un periódico

En este caso de uso un usuario logueado como user lista sus periódicos y cambia uno de privado y público y viceversa.

Menú Principal > Entrar > Usuario > Mis periódicos > Poner público/ Poner privado

El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	12	9	24	2	512	0.00%	43.1/sec	230.3
/scripts/jquery.min.js	4500	31	21	66	3	735	0.00%	14.8/sec	1587.2
/scripts/cookieconsent.js	4500	17	12	32	2	371	0.00%	14.8/sec	508.3
/styles/cookieconsent.mi...	4500	12	8	23	1	433	0.00%	14.8/sec	61.3
/scripts/polyfills.js	4500	11	8	22	1	332	0.00%	14.8/sec	111.5
/styles/bootstrap.min.css	4500	32	22	65	3	470	0.00%	14.8/sec	1756.6
/scripts/bootstrap.min.js	4500	17	12	35	2	727	0.00%	14.8/sec	540.2
/styles/style.css	4500	10	7	22	1	351	0.00%	14.8/sec	23.2
/scripts/helpers.js	4500	10	7	22	1	392	0.00%	14.8/sec	33.8
/images/newspaper-1.jpg	4500	35	23	74	5	1042	0.00%	14.8/sec	2253.5
/security/login.do	4500	13	9	24	3	371	0.00%	14.8/sec	82.1
/j_spring_security_check	4500	57	34	110	5	1136	0.00%	14.8/sec	88.9
/newspaper/user/list.do	13500	79	48	162	7	2127	0.00%	44.0/sec	331.8
/newspaper/user/putPriv...	4500	74	39	159	5	2262	0.00%	14.8/sec	173.2
/newspaper/user/putPub...	4500	77	38	169	4	1896	0.00%	14.8/sec	173.3
/j_spring_security_logout	4500	24	17	48	4	359	0.00%	14.8/sec	80.0
TOTAL	90000	35	16	77	1	2262	0.00%	287.6/sec	7816.1

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, en torno a las dos décimas de segundo por lo que con este número de usuarios el caso de uso es bastante eficiente.

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 17256 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 160 usuarios ejecutando nuevamente 30 acciones cada uno.

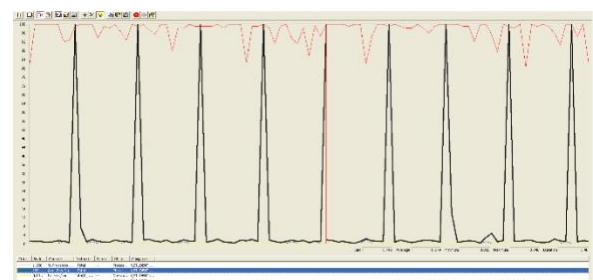
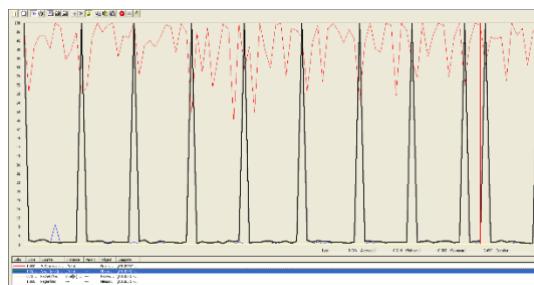
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	14400	18	13	35	2	461	0.00%	41.6/sec	222.1
/scripts/jquery.min.js	4800	38	28	77	4	474	0.00%	14.2/sec	1527.7
/scripts/cookieconsent.js	4800	22	16	43	2	408	0.00%	14.2/sec	489.2
/styles/cookieconsent.mi...	4800	17	12	33	1	1050	0.00%	14.2/sec	59.0
/scripts/polyfills.js	4800	16	12	33	1	396	0.00%	14.2/sec	107.4
/styles/bootstrap.min.css	4800	39	28	80	4	526	0.00%	14.2/sec	1690.7
/scripts/bootstrap.min.js	4800	23	17	47	2	974	0.00%	14.2/sec	519.9
/styles/style.css	4800	16	11	32	1	415	0.00%	14.2/sec	22.4
/scripts/helpers.js	4800	15	11	30	1	363	0.00%	14.2/sec	32.6
/images/newspaper-1.jpg	4800	42	28	86	5	658	0.00%	14.2/sec	2169.7
/security/login.do	4800	18	13	35	3	347	0.00%	14.2/sec	79.0
/j_spring_security_check	4800	204	85	526	5	3144	0.00%	14.2/sec	85.6
/newspaper/userlist.do	14400	244	119	638	7	4473	0.00%	42.3/sec	319.0
/newspaper/user/pulPriv...	4800	246	83	719	4	3873	0.00%	14.2/sec	174.0
/newspaper/user/pulPub...	4800	251	105	675	5	3370	0.00%	14.2/sec	174.0
/j_spring_security_logout	4800	34	26	66	4	468	0.00%	14.2/sec	76.9
TOTAL	96000	88	22	196	1	4473	0.00%	277.3/sec	7550.1

En este caso cuando se pone a público y se vuelve a la página o se pone en privado y se vuelve a la página, el sistema tarda de media en el 90% de las ocasiones más de 1,3 segundos, 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 16635.



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



## Listar periódicos con los anuncios de un agente

En este caso de uso un usuario logueado como agente lista los periódicos donde tiene un anuncio.

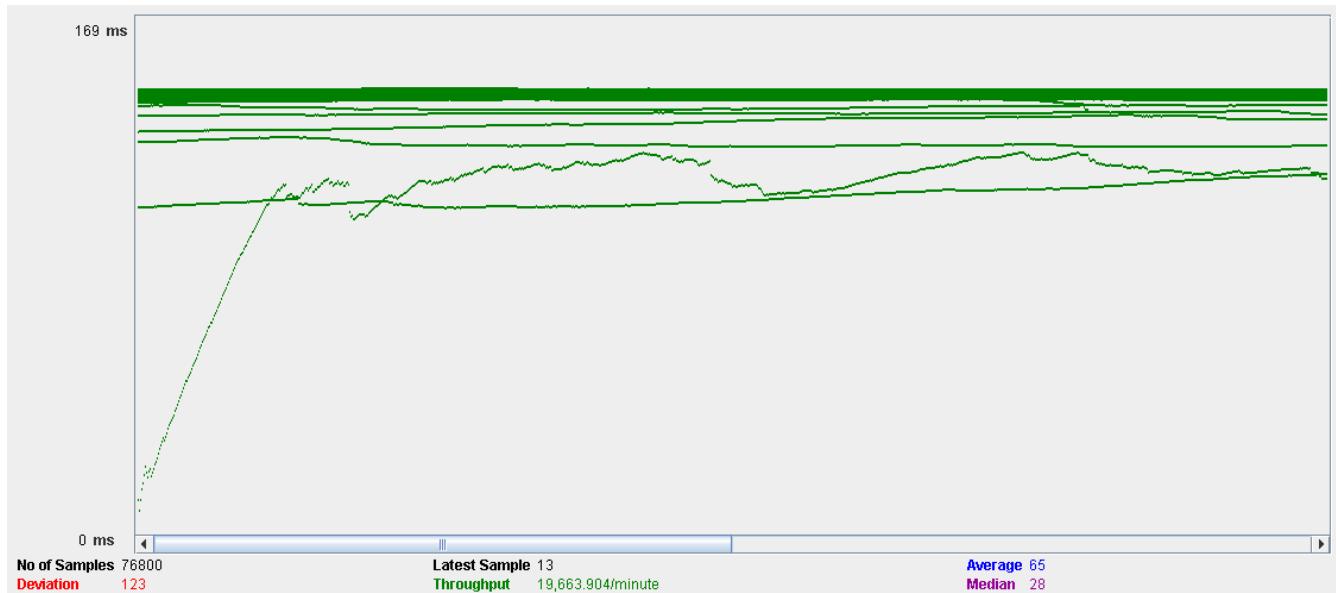
Menú Principal > Entrar > Agente > Periódicos con anuncios

El límite de este caso de uso se sitúa en 170 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	14400	36	19	64	2	3291	0.00%	61.4/sec	324.8
/scripts/bootstrap....	4800	49	29	102	2	781	0.00%	21.1/sec	768.2
/styles/bootstrap....	4800	76	49	164	4	994	0.00%	21.1/sec	2498.2
/scripts/jquery.min....	4800	77	49	165	4	977	0.00%	21.1/sec	2258.8
/styles/cookiecon....	4800	42	20	96	1	850	0.00%	21.1/sec	87.2
/styles/style.css	4800	38	19	85	1	829	0.00%	21.1/sec	33.0
/scripts/helpers.js	4800	32	17	69	1	676	0.00%	21.1/sec	48.2
/scripts/polyfills.js	4800	31	18	61	1	629	0.00%	21.1/sec	158.8
/scripts/cookiecon....	4800	42	25	84	2	1179	0.00%	21.1/sec	723.5
/images/newspaper....	4800	76	45	166	5	1361	0.00%	21.0/sec	3206.0
/security/login.do	4800	46	20	82	3	1954	0.00%	21.0/sec	116.8
/j_spring_security....	4800	166	84	414	7	2234	0.00%	21.0/sec	123.2
/newspaper/agent....	4800	180	96	432	9	2127	0.00%	21.0/sec	143.2
/j_spring_security....	4800	75	42	169	4	1015	0.00%	21.0/sec	113.5
TOTAL	76800	65	28	150	1	3291	0.00%	327.7/sec	10324.6

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, en torno a medio segundo por lo que con este número de usuarios el caso de uso es bastante eficiente.

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 19663 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 180 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	16200	91	28	228	3	3012	0.00%	51.4/sec	272.0
/scripts/bootstrap.min.js	5400	107	39	258	3	2903	0.00%	17.5/sec	639.9
/styles/bootstrap.min.css	5400	143	71	334	6	3243	0.00%	17.5/sec	2080.7
/scripts/jquery.min.js	5400	144	69	342	6	2889	0.00%	17.5/sec	1880.1
/styles/cookieconsent.mi...	5400	110	28	317	2	2781	0.00%	17.5/sec	72.6
/styles/style.css	5400	101	26	267	1	2904	0.00%	17.5/sec	27.5
/scripts/helpers.js	5400	90	25	238	1	2047	0.00%	17.5/sec	40.1
/scripts/polyfills.js	5400	87	26	223	2	2032	0.00%	17.5/sec	132.1
/scripts/cookieconsent.js	5400	101	35	257	3	1993	0.00%	17.5/sec	602.1
/images/newspaper-1.jpg	5400	165	80	412	7	2960	0.00%	17.5/sec	2671.4
/security/login.do	5400	127	29	371	4	2911	0.00%	17.5/sec	97.1
/j_spring_security_check	5400	636	367	1514	9	12031	0.00%	17.5/sec	102.5
/newspaper/agentListWith...	5400	576	289	1392	10	12651	0.00%	17.5/sec	119.3
/j_spring_security_logout	5400	265	89	780	3	3700	0.00%	17.5/sec	94.6
TOTAL	86400	183	46	495	1	12651	0.00%	274.4/sec	8644.6

En este caso cuando se accede al listado de los periódicos y cuando te logueas el sistema tarda de media en el 90% de las ocasiones más de 1,4 segundos, 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 16463.



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



Listar periódicos en los que no tiene anuncios el agente

En este caso de uso un usuario logueado como agente lista los periódicos donde no tiene un anuncio.

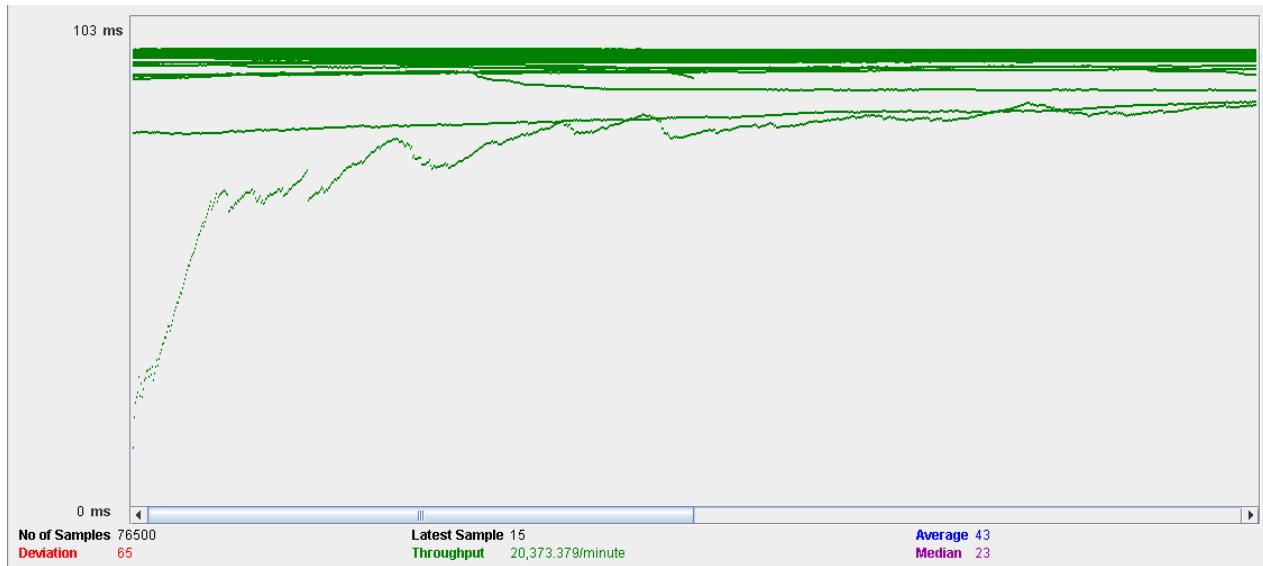
Menú Principal > Entrar > Agente > Periódicos sin anuncios

El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	26	17	51	3	612	0.00%	59.9/sec	316.7
/scripts/jquery.min...	4500	61	42	139	4	697	0.00%	20.6/sec	2204.9
/styles/bootstrap....	4500	64	43	143	4	529	0.00%	20.6/sec	2440.2
/styles/style.css	4500	29	16	61	1	496	0.00%	20.6/sec	32.2
/styles/cookiecon...	4500	27	16	57	1	423	0.00%	20.6/sec	85.1
/scripts/bootstrap....	4500	38	24	79	2	656	0.00%	20.6/sec	750.5
/scripts/cookiecon...	4500	35	22	72	2	454	0.00%	20.6/sec	706.2
/scripts/helpers.js	4500	26	15	53	1	427	0.00%	20.6/sec	47.0
/scripts/polyfills.js	4500	25	15	51	1	446	0.00%	20.6/sec	155.0
/images/newspap...	4500	68	45	152	5	621	0.00%	20.6/sec	3134.6
/favicon.ico	4500	21	11	43	1	596	0.00%	20.6/sec	115.1
/security/login.do	4500	25	16	50	3	622	0.00%	20.6/sec	114.1
/j_spring_security...	4500	79	47	169	7	1440	0.00%	20.6/sec	120.5
/newspaper/agent...	4500	101	63	209	11	1515	0.00%	20.5/sec	151.7
/j_spring_security...	4500	56	37	114	4	1403	0.00%	20.5/sec	110.9
TOTAL	76500	43	23	101	1	1515	0.00%	339.6/sec	10190.8

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, en torno a dos décimas de segundo por lo que con este número de usuarios el caso de uso es bastante eficiente.

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 20733 veces por minuto.



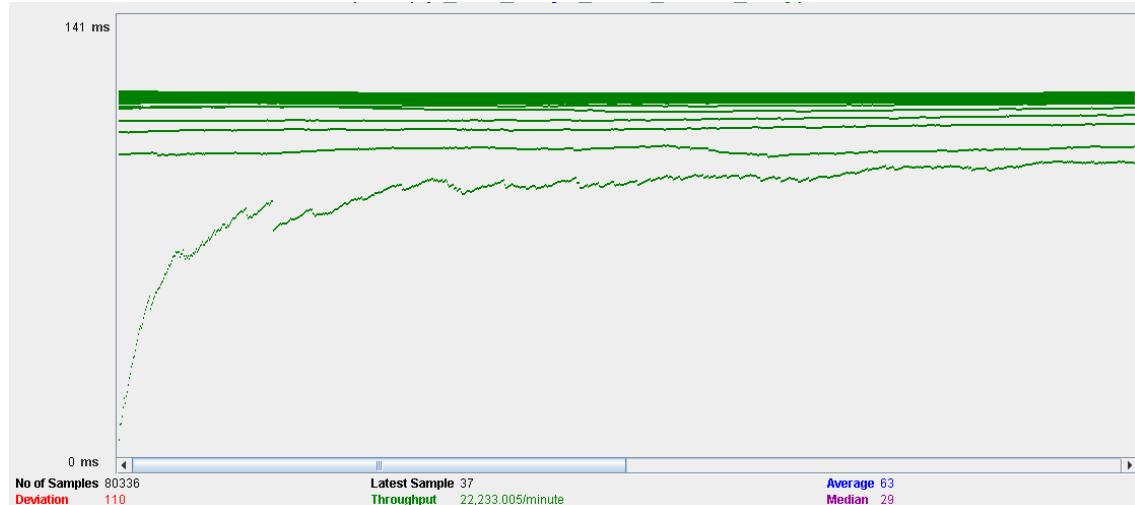
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 160 usuarios ejecutando nuevamente 30 acciones cada uno.

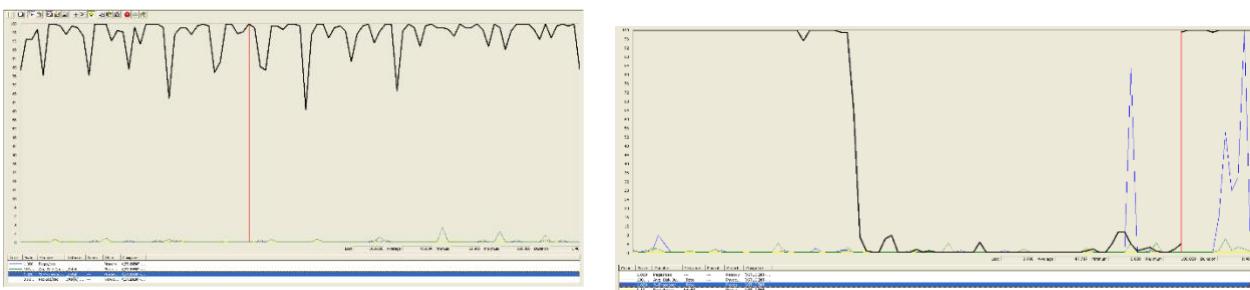
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	9395	67	32	148	2	1149	0.00%	56.0/sec	296.2
/scripts/jquery.min...	3217	128	80	295	6	1509	0.00%	19.2/sec	2057.6
/styles/bootstrap....	3217	126	78	298	6	2091	0.00%	19.2/sec	2276.9
/styles/style.css	3217	81	31	205	2	1406	0.00%	19.2/sec	30.1
/styles/cookiecon...	3216	64	30	158	1	1052	0.03%	19.2/sec	79.4
/scripts/bootstrap....	3215	79	44	190	3	1975	0.03%	19.2/sec	699.6
/scripts/cookiecon...	3213	74	40	167	2	1047	0.03%	19.2/sec	657.9
/scripts/helpers.js	3211	68	30	165	1	1498	0.03%	19.2/sec	43.8
/scripts/polyfills.js	3210	68	30	169	2	1108	0.00%	19.1/sec	144.3
/images/newsnap...	3210	133	88	307	6	1287	0.03%	19.1/sec	2916.3
/favicon.ico	3209	66	27	153	2	1295	0.00%	19.3/sec	108.1
/security/login.do	3167	83	33	204	4	1137	0.00%	19.2/sec	106.7
/j_spring_security...	3134	405	197	989	5	6323	0.03%	19.1/sec	111.6
/newspaper/agent...	3088	448	236	1025	12	7047	0.10%	19.1/sec	141.0
/j_spring_security...	3047	175	87	457	5	1777	0.00%	18.9/sec	101.9
Error:	1	0	0	0	0	0	100.00%	∞/sec	0
TOTAL	53967	128	48	308	0	7047	0.02%	.1/hour	.0

En este caso cuando se accede al listado de los periódicos el sistema tarda de media en el 90% de las ocasiones más de 1 segundos, además que se producen errores debidos a la excesiva cantidad de usuarios que el caso de uso es incapaz de soportar en nuestro sistema.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 22233.



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



## Listar los volúmenes y mostrar sus periódicos (sin autenticar)

En este caso de uso un usuario no autenticado lista todos los volúmenes y muestra uno con sus periódicos

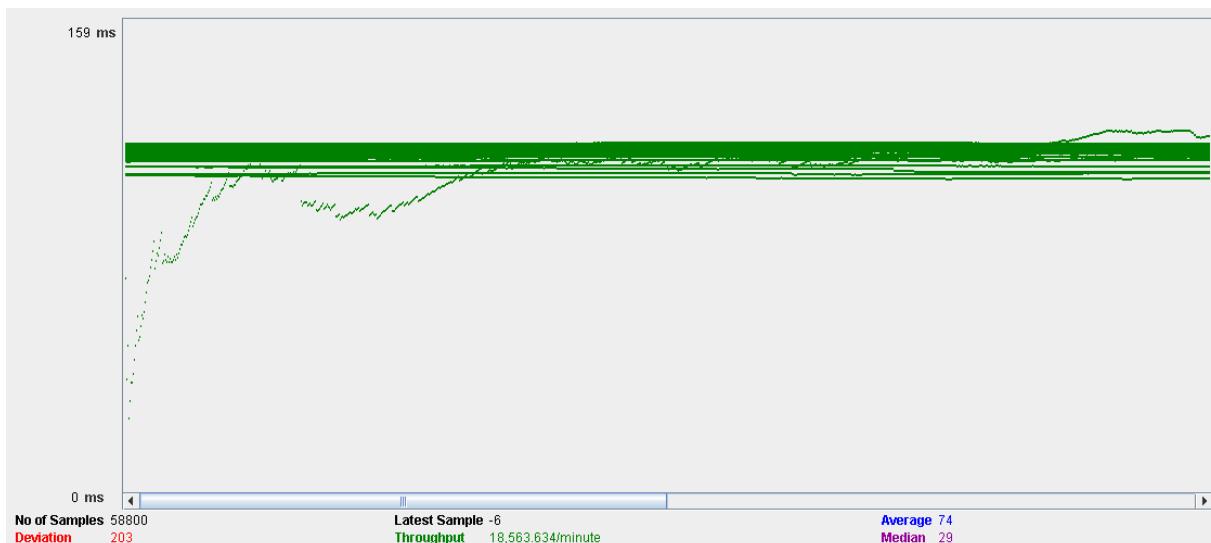
Menú Principal > Todos los volúmenes > Display

El límite de este caso de uso se sitúa en 140 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4200	30	21	56	0	875	0.00%	22.7/sec	116.1
/scripts/jquery.min...	4200	66	51	131	0	829	0.00%	22.7/sec	2433.2
/styles/style.css	4200	25	17	50	0	770	0.00%	22.7/sec	35.6
/scripts/bootstrap....	4200	69	53	136	0	957	0.00%	22.7/sec	2692.2
/scripts/cookiecon...	4200	34	25	65	0	761	0.00%	22.7/sec	779.0
/scripts/bootstrap...	4200	38	28	73	0	1290	0.00%	22.7/sec	827.9
/styles/cookiecon...	4200	25	17	50	0	471	0.00%	22.7/sec	93.9
/scripts/polyfills.js	4200	24	17	49	0	889	0.00%	22.7/sec	170.9
/scripts/helpers.js	4200	22	16	45	0	693	0.00%	22.7/sec	51.8
/images/newspaper...	4200	77	60	156	0	772	0.00%	22.7/sec	3456.7
/favicon.ico	4200	22	17	45	0	455	0.00%	22.7/sec	126.9
/volume/list.do	4200	283	139	623	0	6910	0.00%	22.7/sec	143.5
/volume/display.do	4200	291	146	670	0	8921	0.00%	22.6/sec	146.8
/welcome/index.do	4200	27	20	51	0	1065	0.00%	22.7/sec	115.8
TOTAL	58800	74	29	145	0	8921	0.00%	309.4/sec	10899.2

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo de seis décimas de segundo cuando haces el display y el listado de los volúmenes por lo que con este número de usuarios el caso de uso es bastante eficiente.

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 18563 veces por minuto.



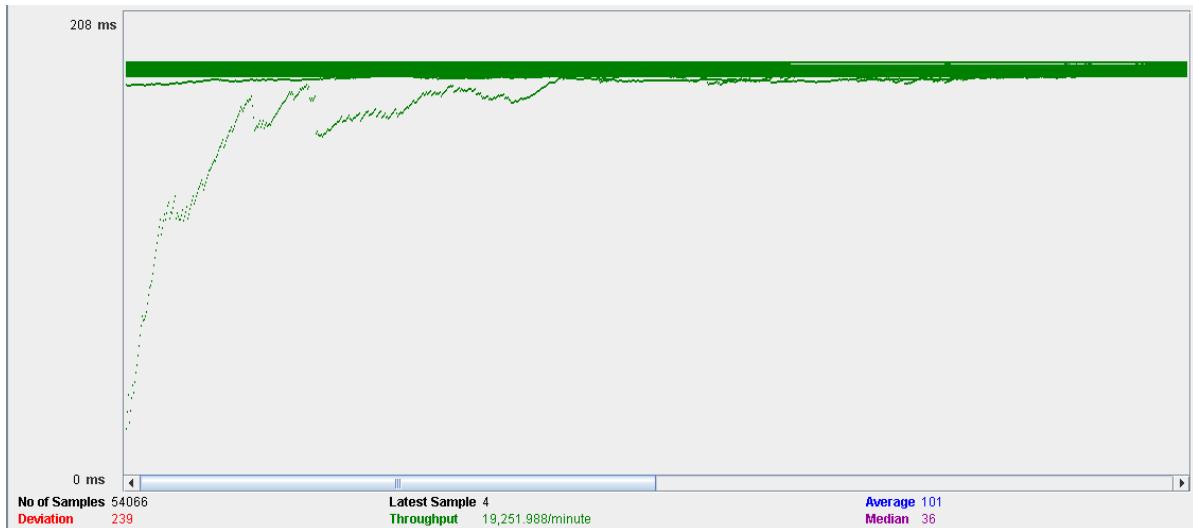
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 150 usuarios ejecutando nuevamente 30 acciones cada uno.

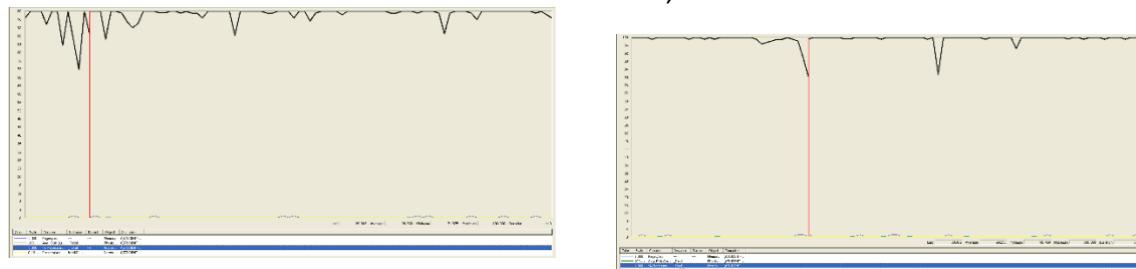
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4295	43	27	81	0	1033	0.00%	22.8/sec	116.5
/scripts/jquery.min...	4295	83	64	164	0	1081	0.00%	22.8/sec	2441.4
/styles/style.css	4293	37	22	77	0	902	0.00%	22.8/sec	35.7
/styles/bootstrap....	4293	84	65	165	0	1386	0.00%	22.8/sec	2704.7
/scripts/cookiecon...	4293	50	33	104	0	878	0.00%	22.8/sec	782.6
/scripts/bootstrap....	4292	52	37	106	0	561	0.00%	22.8/sec	831.5
/styles/cookiecon...	4292	34	22	73	0	859	0.00%	22.8/sec	94.3
/scripts/polyfills.js	4292	34	22	70	0	705	0.00%	22.8/sec	171.7
/scripts/helpers.js	4291	33	22	68	0	563	0.00%	22.8/sec	52.1
/images/newspap...	4290	93	72	187	0	738	0.00%	22.8/sec	3469.1
/favicon.ico	4290	29	19	60	0	833	0.00%	22.9/sec	128.1
/volume/list.do	4206	378	190	1018	0	4257	0.00%	22.6/sec	143.2
/volume/display.do	4166	385	210	977	0	4788	0.00%	22.6/sec	146.5
/welcome/index.do	4150	43	26	87	0	674	0.00%	22.6/sec	115.4
TOTAL	59738	98	37	196	0	4788	0.00%	316.0/sec	11186.2

En este caso cuando se accede al listado de los volúmenes el sistema tarda de media en el 90% de las ocasiones más de 1 segundos, por lo que con esta cantidad de usuarios el sistema no es del todo eficiente.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 19251.



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



## Listar los volúmenes y mostrar sus periódicos (autenticado)

En este caso de uso un usuario autenticado lista todos los volúmenes y muestra uno con sus periódicos

Menú Principal > Entrar > Todos los volúmenes > Display

El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	30	22	57	0	538	0.00%	47.5/sec	253.8
/styles/cookiecon...	4500	30	19	62	0	465	0.00%	16.3/sec	67.5
/styles/style.css	4500	28	19	58	0	726	0.00%	16.3/sec	25.6
/styles/bootstrap....	4500	71	51	152	0	556	0.00%	16.3/sec	1935.0
/scripts/jquery.min...	4500	69	48	149	0	770	0.00%	16.3/sec	1748.8
/scripts/polyfills.js	4500	31	20	63	0	475	0.00%	16.3/sec	122.9
/scripts/helpers.js	4500	28	19	59	0	549	0.00%	16.3/sec	37.3
/scripts/cookiecon...	4500	37	27	78	0	554	0.00%	16.3/sec	560.0
/scripts/bootstrap....	4500	40	28	85	0	510	0.00%	16.3/sec	595.1
/images/newspap...	4500	77	53	168	0	672	0.00%	16.3/sec	2484.0
/security/login.do	4500	29	21	56	0	517	0.00%	16.3/sec	90.5
/j_spring_security...	4500	149	93	307	0	3126	0.00%	16.3/sec	98.0
/volume/list.do	4500	155	101	323	0	3329	0.00%	16.3/sec	114.3
/volume/display.do	4500	133	83	279	0	3348	0.00%	16.3/sec	122.2
/j_spring_security...	4500	63	44	124	0	820	0.00%	16.3/sec	88.0
TOTAL	76500	61	31	139	0	3348	0.00%	269.4/sec	8115.2

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo de tres décimas de segundo cuando haces el display y el listado de los volúmenes por lo que con este número de usuarios el caso de uso es bastante eficiente.

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 16165 veces por minuto.



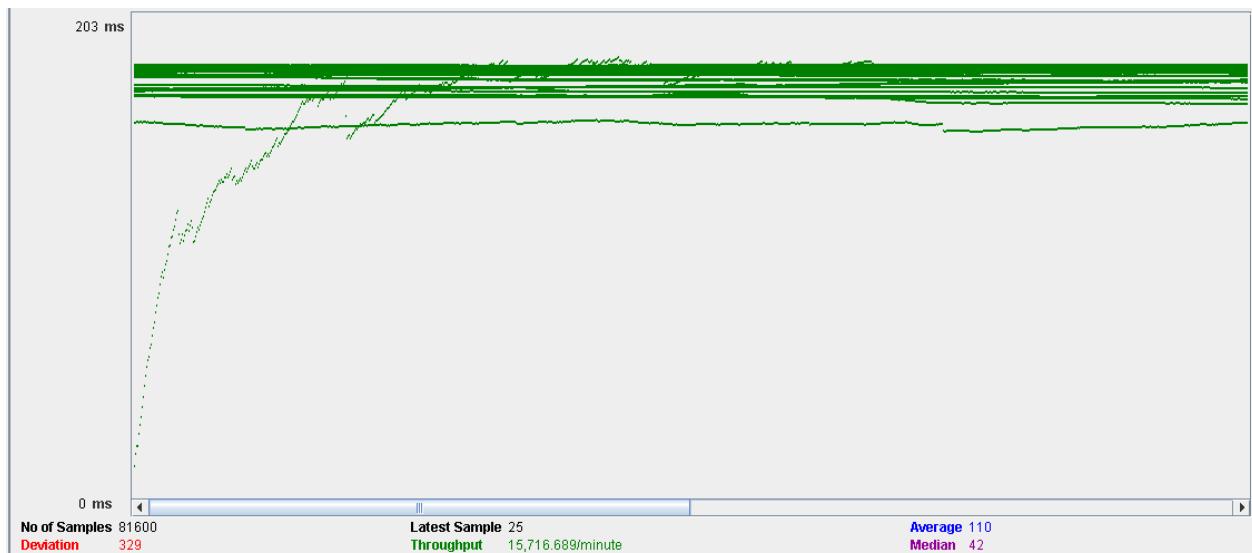
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 160 usuarios ejecutando nuevamente 30 acciones cada uno.

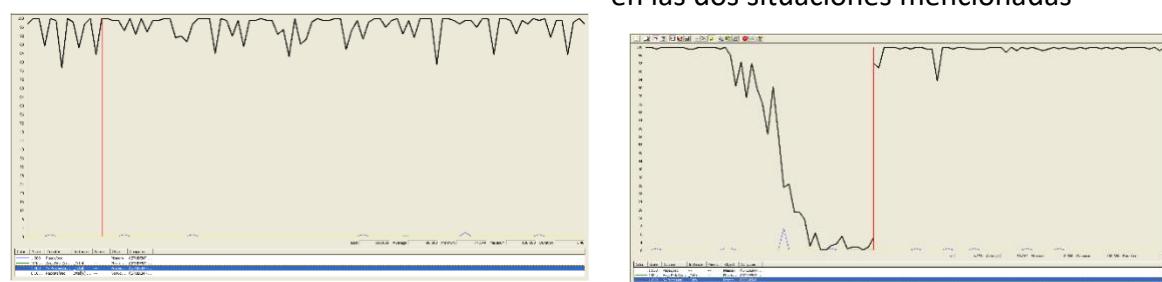
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	14400	43	27	84	0	2187	0.00%	46.2/sec	246.8
/styles/cookiecon...	4800	41	25	84	0	1792	0.00%	15.9/sec	65.6
/styles/style.css	4800	39	25	76	0	898	0.00%	15.9/sec	24.9
/styles/bootstrap....	4800	96	67	194	0	2198	0.00%	15.9/sec	1881.5
/scripts/jquery.min...	4800	87	61	179	0	1687	0.00%	15.9/sec	1700.1
/scripts/polyfills.js	4800	45	26	98	0	1718	0.00%	15.9/sec	119.5
/scripts/helpers.js	4800	42	26	84	0	2763	0.00%	15.9/sec	36.3
/scripts/cookiecon...	4800	53	36	106	0	1116	0.00%	15.9/sec	544.4
/scripts/bootstrap....	4800	57	39	118	0	869	0.00%	15.9/sec	578.5
/images/newspap...	4800	107	75	223	0	1358	0.00%	15.8/sec	2412.1
/security/login.d...	4800	47	27	97	0	744	0.00%	15.8/sec	87.9
/_spring_security...	4800	363	146	845	0	9919	0.00%	15.8/sec	95.2
/volume/list.do	4800	339	144	716	0	10831	0.00%	15.8/sec	110.7
/volume/display.do	4800	336	125	769	0	11880	0.00%	15.8/sec	118.4
/_spring_security...	4800	96	59	203	0	2166	0.00%	15.8/sec	85.3
TOTAL	81600	110	42	206	0	11880	0.00%	261.9/sec	7890.1

En este caso cuando se carga los estilos cuando entras en la página el sistema tarda de media en el 90% de las ocasiones más de 1 segundos, por lo que con esta cantidad de usuarios el sistema no es del todo eficiente.

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 15716.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador.  
Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes  
en las dos situaciones mencionadas



anteriormente, en el mismo orden.

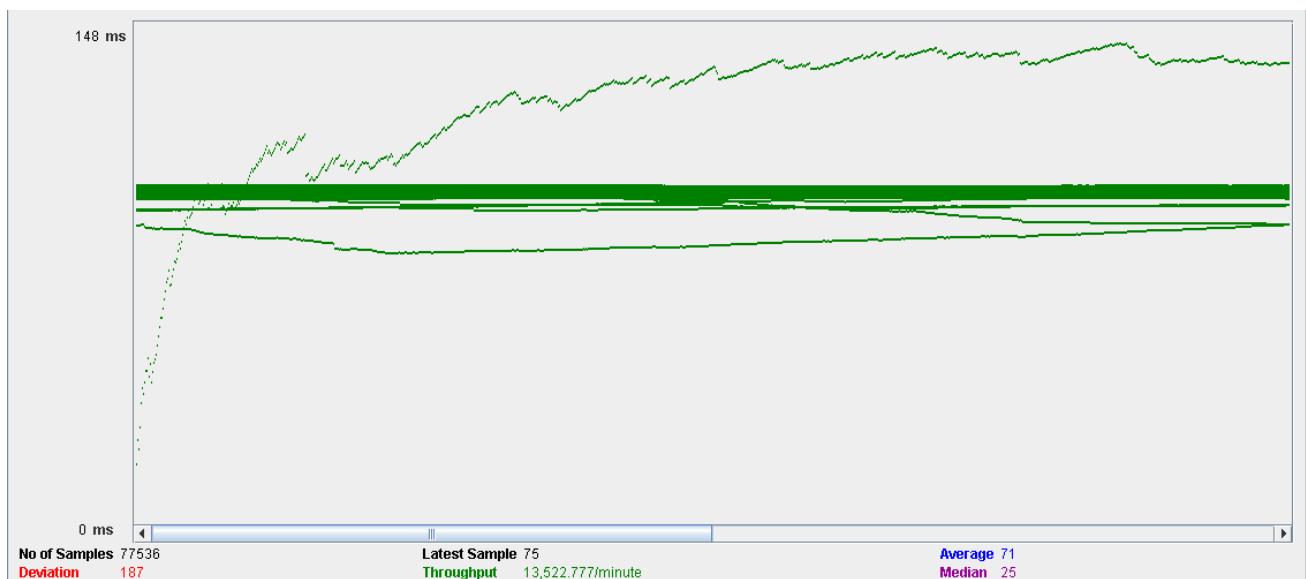
Suscribirse a un Volumen

En este caso de uso un usuario autenticado como customer lista todos los volúmenes, entra en uno al que no esté suscrito y se suscribe  
Menú Principal > Entrar > Perfil > Todos los volúmenes > Display > Suscribirse  
El límite de este caso de uso se sitúa en 140 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	12391	17	13	35	0	850	0.00%	33.8/sec	179.7
/scripts/helpers.js	4181	14	11	32	0	165	0.00%	11.4/sec	26.1
/scripts/cookiecon...	4181	22	17	45	0	334	0.00%	11.4/sec	391.8
/scripts/jquery.min....	4181	41	32	84	0	661	0.00%	11.4/sec	1223.7
/styles/style.css	4181	14	11	31	0	371	0.00%	11.4/sec	17.9
/styles/cookiecon...	4181	14	11	31	0	454	0.00%	11.4/sec	47.2
/scripts/polyfills.js	4181	14	11	31	0	567	0.00%	11.4/sec	86.0
/scripts/bootstrap....	4181	23	18	48	0	429	0.00%	11.4/sec	416.5
/styles/bootstrap....	4181	42	32	86	0	835	0.00%	11.4/sec	1354.3
/images/newspap...	4181	51	37	103	0	873	0.00%	11.4/sec	1742.9
/security/login.do	4165	19	15	36	0	594	0.00%	11.4/sec	63.5
/j_spring_security...	4153	135	59	292	0	3447	0.00%	11.4/sec	68.1
/volume/list.do	4139	142	60	290	0	4600	0.00%	11.5/sec	79.8
/volume/display.do	4118	184	90	357	0	5096	0.00%	11.5/sec	81.5
/subscriptionVolu...	4092	160	65	338	0	3318	0.00%	11.4/sec	157.9
/subscriptionVolu...	4074	232	137	473	0	3569	0.05%	11.4/sec	92.9
/subscriptionVolu...	4072	192	94	413	0	3261	0.00%	11.4/sec	76.6
/j_spring_security...	4057	34	26	62	0	813	0.00%	11.4/sec	61.7
TOTAL	82890	69	25	139	0	5096	0.00%	225.9/sec	6138.8

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo ocho décimas de segundo cuando te suscribes al volumen y vuelves después al display del volumen, por lo que con esta cantidad de usuarios el rendimiento es eficiente..

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 16165 veces por minuto.

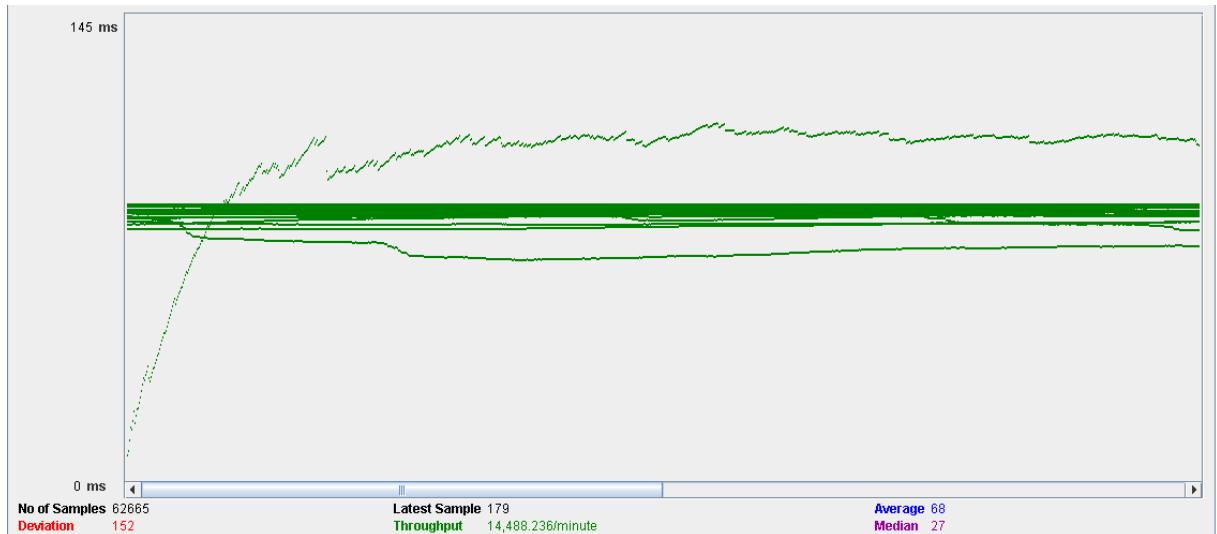


Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.  
En este caso se ha utilizado 150 usuarios ejecutando nuevamente 30 acciones cada uno.

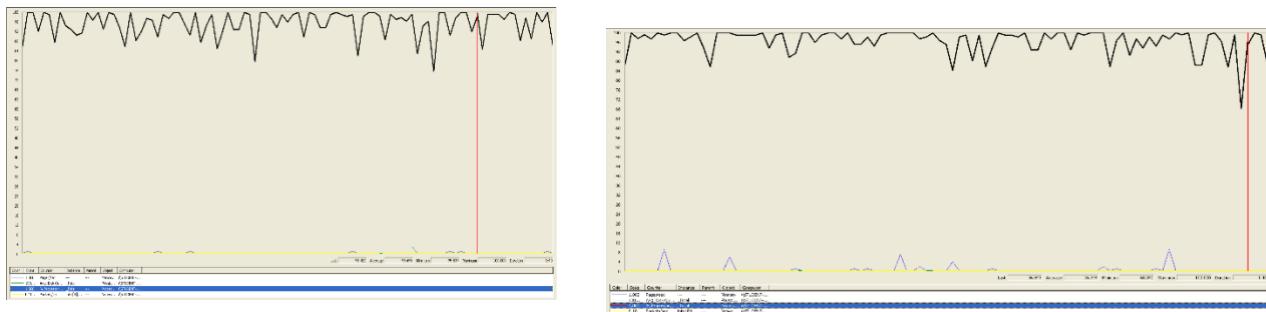
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	10432	20	15	38	0	621	0.00%	35.5/sec	189.0
/scripts/helpers.js	3540	17	12	35	0	589	0.00%	12.1/sec	27.6
/scripts/cookiecon...	3540	25	20	49	0	605	0.00%	12.1/sec	414.7
/scripts/jquery min...	3540	45	35	90	0	655	0.00%	12.1/sec	1295.1
/styles/style.css	3540	17	13	35	0	313	0.00%	12.1/sec	18.9
/styles/cookiecon...	3540	17	13	35	0	192	0.00%	12.1/sec	50.0
/scripts/polyfills.js	3540	16	13	34	0	221	0.00%	12.1/sec	91.0
/scripts/bootstrap....	3540	26	21	53	0	455	0.00%	12.1/sec	440.8
/styles/bootstrap....	3539	46	37	92	0	636	0.00%	12.1/sec	1432.9
/images/newspap...	3539	53	40	105	0	668	0.03%	12.1/sec	1845.8
/security/login.do	3515	22	16	39	0	615	0.00%	12.1/sec	67.1
/j_spring_security...	3502	140	73	299	0	2708	0.03%	12.1/sec	72.0
/volume/list.do	3482	142	76	290	0	3101	0.00%	12.1/sec	84.3
/volume/display.do	3458	176	111	341	0	2602	0.00%	12.1/sec	86.1
/subscriptionVolu...	3438	155	80	322	0	2867	0.00%	12.1/sec	166.9
/subscriptionVolu...	3418	235	169	455	0	3761	0.00%	12.0/sec	98.0
/subscriptionVolu...	3417	194	118	384	0	2972	0.00%	12.1/sec	80.9
/j_spring_security...	3391	37	30	70	0	540	0.00%	12.0/sec	65.0
TOTAL	69911	71	28	160	0	3761	0.00%	237.9/sec	6484.3

En este caso cuando se puede observar que se producen errores en la acción de loguearse, probablemente debido a la cantidad de usuarios concurrentes que no es capaz de soportar el sistema

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 14448.



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



Crear un Volumen

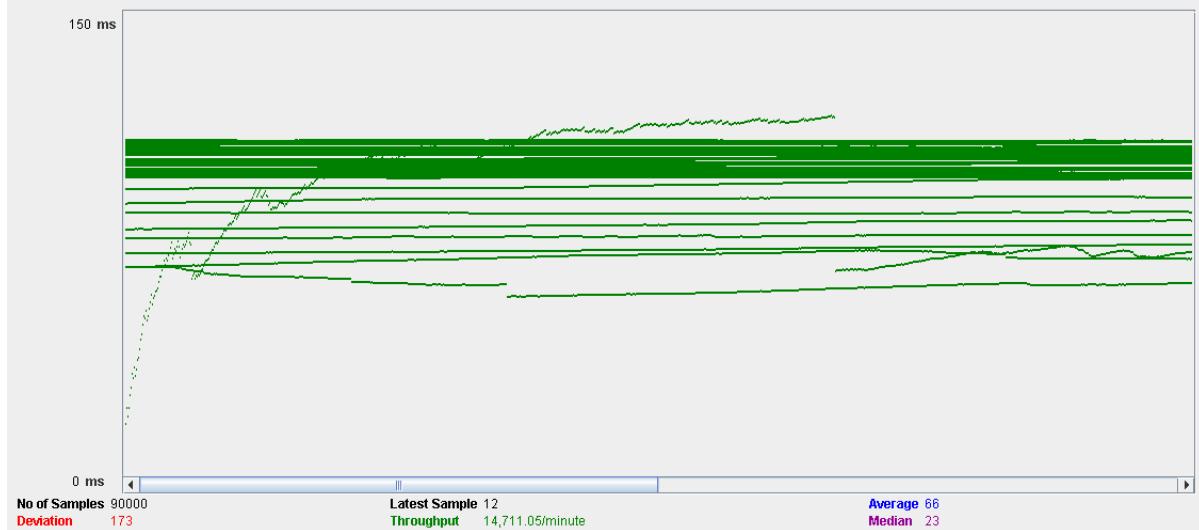
En este caso de uso un usuario autenticado como user accedemos a mis volúmenes y creamos uno

Menú Principal > Entrar > Usuario > Mis volúmenes > Crear > Elegir y crear volumen

El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	15	12	30	0	928	0.00%	36.8/sec	196.3
/styles/bootstrap....	4500	38	30	77	0	999	0.00%	12.7/sec	1512.5
/styles/style.css	4500	13	10	28	0	297	0.00%	12.7/sec	20.0
/styles/cookiecon...	4500	13	10	28	0	404	0.00%	12.7/sec	52.8
/scripts/polyfills.js	4500	13	10	29	0	313	0.00%	12.7/sec	96.0
/scripts/cookiecon...	4500	19	15	39	0	917	0.00%	12.7/sec	437.7
/scripts/bootstrap....	4500	21	17	43	0	1008	0.00%	12.7/sec	465.1
/scripts/jquery.min...	4500	35	28	73	0	490	0.00%	12.7/sec	1366.8
/scripts/helpers.js	4500	13	10	28	0	848	0.00%	12.7/sec	29.2
/images/newsrap...	4500	42	34	85	0	1013	0.00%	12.7/sec	1934.2
/security/login.do	4500	16	13	31	0	388	0.00%	12.8/sec	70.8
/j_spring_security...	4500	123	59	255	0	3120	0.00%	12.8/sec	76.8
/volume/user/list.do	9000	142	74	283	0	6757	0.00%	25.1/sec	2025.3
/newspaper/userfl...	4500	118	59	244	0	4170	0.00%	12.8/sec	97.6
/volume/user/crea...	4500	153	86	314	0	4962	0.00%	12.8/sec	90.0
/volume/user/edit...	4500	343	216	702	5	4786	0.00%	12.8/sec	1038.2
/j_spring_security...	4500	29	23	51	0	1876	0.00%	12.8/sec	69.0
TOTAL	90000	66	23	137	0	6757	0.00%	245.2/sec	9254.2

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo un segundo cuando creas el volumen y vuelves al listado de tus volúmenes, por lo que con esta cantidad es aceptable, pero se va acercando al límite. 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 14711 veces por minuto.

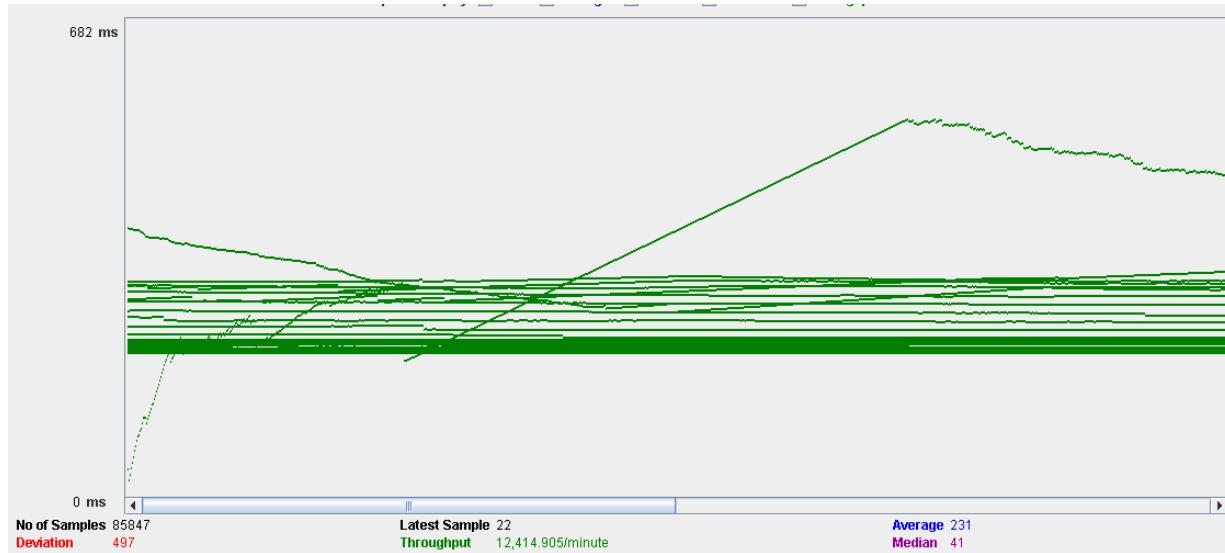


Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema. En este caso se ha utilizado 160 usuarios ejecutando nuevamente 30 acciones cada uno.

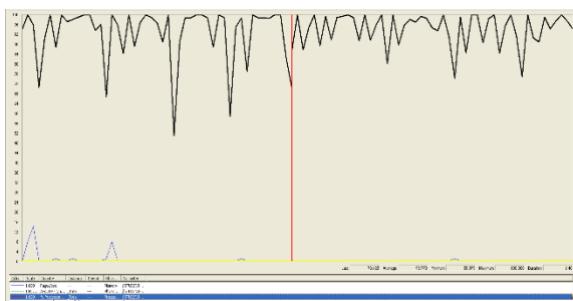
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	12810	27	21	56	0	693	0.02%	30.9/sec	164.9
/styles/bootstrap....	4339	58	46	122	0	885	0.00%	10.5/sec	1242.6
/styles/style.css	4339	24	18	52	0	524	0.02%	10.5/sec	16.4
/styles/cookiecon...	4338	25	18	55	0	569	0.00%	10.5/sec	43.4
/scripts/polyfills.js	4338	24	18	51	0	642	0.00%	10.5/sec	78.9
/scripts/cookiecon...	4338	33	25	69	0	393	0.00%	10.5/sec	359.4
/scripts/bootstrap....	4338	35	26	74	0	404	0.00%	10.5/sec	382.0
/scripts/jquery.min...	4338	58	44	117	0	644	0.00%	10.5/sec	1122.3
/scripts/helpers.js	4338	24	17	53	0	651	0.02%	10.5/sec	24.0
/images/newspap...	4337	72	56	148	0	562	0.00%	10.5/sec	1595.7
/security/login.do	4311	29	23	58	0	389	0.02%	10.4/sec	58.0
/j_spring_security...	4293	564	298	1498	0	5899	0.02%	10.4/sec	62.7
/volume/user/list.do	8487	595	389	1360	3	6475	0.04%	20.7/sec	4625.7
/newspaper/user/l...	4262	544	331	1301	0	5233	0.09%	10.4/sec	79.6
/volume/user/crea...	4243	577	379	1304	0	6789	0.05%	10.4/sec	73.3
/volume/user/edit...	4217	1270	1021	2517	15	8948	0.12%	10.3/sec	2316.2
/j_spring_security...	4181	56	45	102	0	1742	0.00%	10.3/sec	55.5
TOTAL	85847	231	41	720	0	8948	0.02%	206.9/sec	12204.6

En este caso se pueden observar que se producen errores y el tiempo medio para las acciones es bastante alto, superior a dos segundos solo la acción de editar, por lo que con esta cantidad de usuarios el caso de uso es inadmisible

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 12414.



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



## Editar un Volumen

En este caso de uso un usuario autenticado como user accedemos a mis volúmenes y editamos uno.

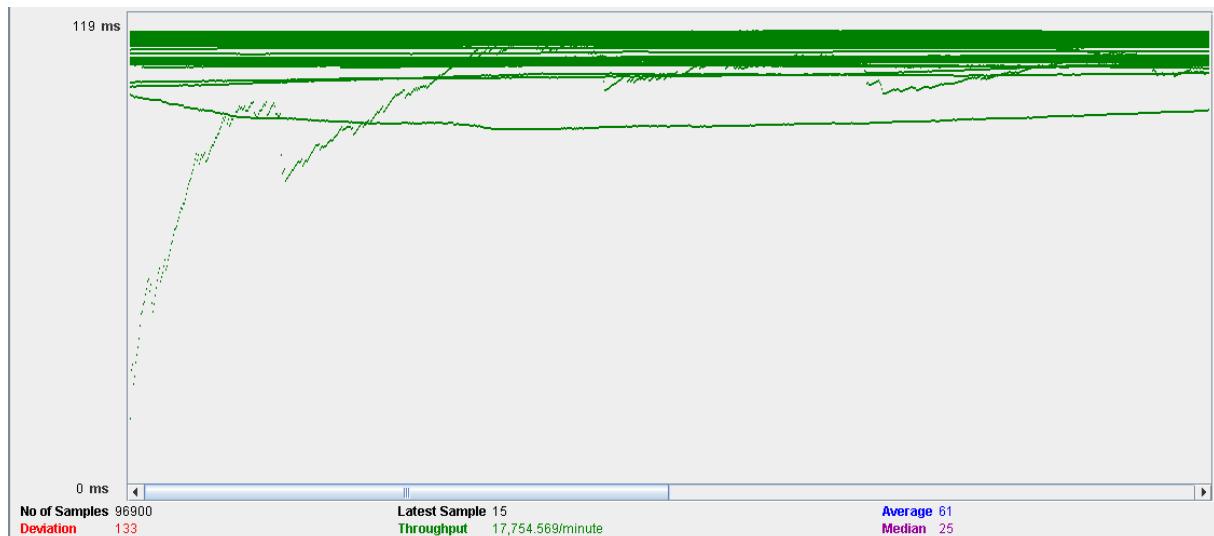
Menú Principal > Entrar > Usuario > Mis volúmenes > Editar

El límite de este caso de uso se sitúa en 170 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15300	25	15	42	0	2509	0.00%	46.7/sec	249.5
/styles/style.css	5100	21	14	41	0	816	0.00%	16.0/sec	25.2
/scripts/jquery.min...	5100	49	34	101	0	1362	0.00%	16.0/sec	1718.4
/scripts/polyfills.js	5100	24	14	42	0	1494	0.00%	16.0/sec	120.8
/styles/bootstrap....	5100	51	35	107	0	1289	0.00%	16.0/sec	1901.8
/styles/cookiecon...	5100	22	14	43	0	1168	0.00%	16.0/sec	66.4
/scripts/cookiecon...	5100	28	20	57	0	1059	0.00%	16.0/sec	550.3
/scripts/helpers.js	5100	20	14	41	0	892	0.00%	16.0/sec	36.7
/scripts/bootstrap....	5100	29	21	60	0	772	0.00%	16.0/sec	584.8
/images/newsrap...	5100	49	33	108	0	1088	0.00%	16.0/sec	2442.2
/security/login.do	5100	22	16	43	0	591	0.00%	16.0/sec	89.0
/j_spring_security...	5100	133	76	303	0	2659	0.00%	16.0/sec	96.5
/volume/user/list.do	10200	126	73	272	0	2269	0.00%	31.8/sec	210.7
/volume/user/edit...	10200	176	86	434	0	2903	0.00%	32.1/sec	222.0
/j_spring_security...	5100	42	31	84	0	615	0.00%	16.0/sec	86.7
TOTAL	96900	61	25	131	0	2903	0.00%	295.9/sec	8171.3

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo siete décimas de segundo cuando editas el volumen y vuelves al listado de tus volúmenes, por lo que con esta cantidad es aceptable, pero se va acercando al límite.

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 17754 veces por minuto.

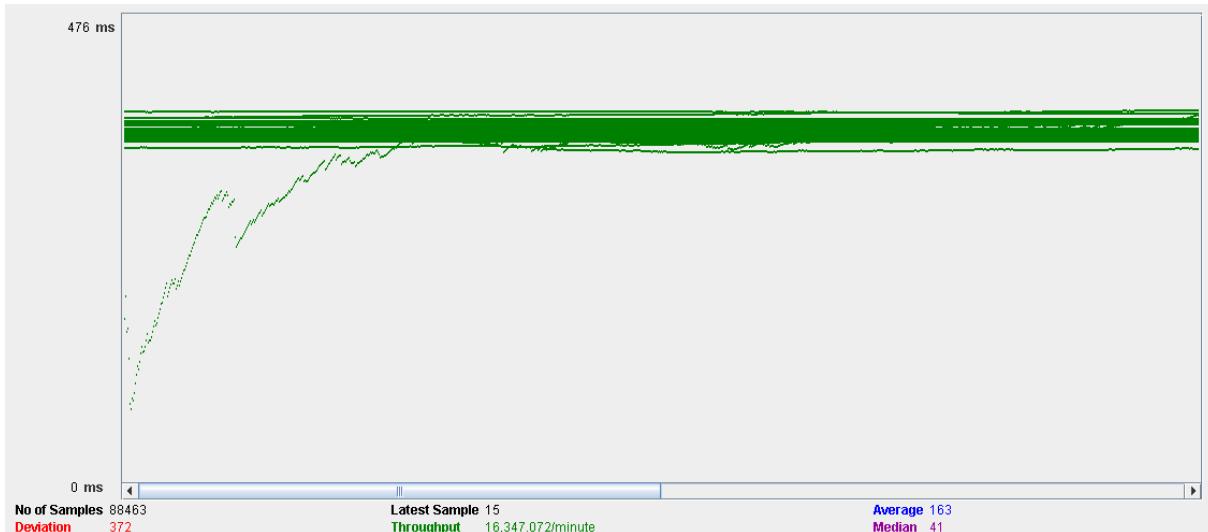


Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

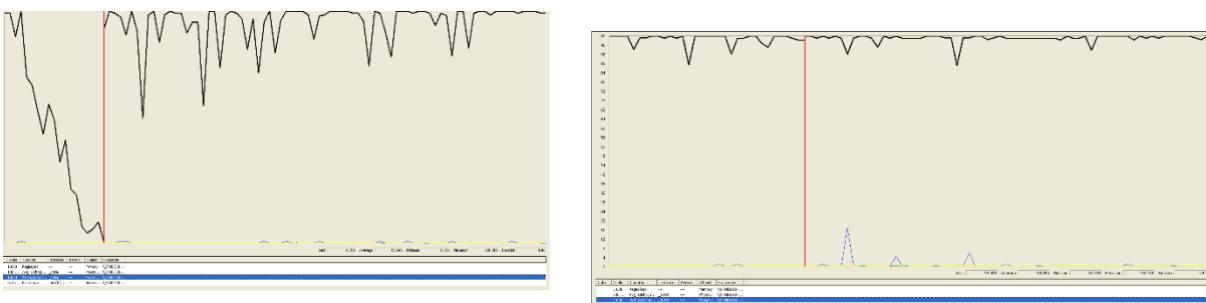
En este caso se ha utilizado 180 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13884	36	22	74	0	1914	0.00%	42.8/sec	228.4
/styles/style.css	4712	32	21	70	0	1970	0.00%	14.5/sec	22.8
/scripts/jquery.min....	4712	70	47	151	0	866	0.00%	14.5/sec	1558.0
/scripts/polyfills.js	4712	35	21	77	0	1535	0.00%	14.5/sec	109.5
/styles/bootstrap....	4710	73	51	153	0	2252	0.00%	14.5/sec	1723.6
/styles/cookiecon...	4709	36	20	81	0	1577	0.00%	14.5/sec	60.2
/scripts/cookiecon...	4707	45	29	97	0	1504	0.00%	14.5/sec	498.7
/scripts/helpers.js	4707	33	20	71	0	1570	0.00%	14.5/sec	33.3
/scripts/bootstrap....	4707	44	30	95	0	1564	0.00%	14.5/sec	530.1
/images/news pap...	4707	83	58	175	0	2112	0.00%	14.5/sec	2221.0
/security/login.do	4671	40	24	84	0	1809	0.00%	14.5/sec	80.7
/j_spring_security...	4641	426	217	1077	0	4607	0.00%	14.5/sec	87.0
/volume/user/list.do	9183	418	194	1081	0	4710	0.01%	28.9/sec	191.1
/volume/user/edit....	9168	590	322	1538	0	5222	0.02%	28.9/sec	200.2
/j_spring_security...	4533	78	49	174	0	1241	0.02%	14.4/sec	77.8
TOTAL	88463	163	41	398	0	5222	0.00%	272.5/sec	7590.9

En este caso se pueden observar que se producen errores y el tiempo medio para las acciones es bastante alto, superior a dos segundos y medio la acción de editar y de volver al listado, por lo que con esta cantidad de usuarios el caso de uso es inadmisible. Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 16347.



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



## Editar una suscripción a volumen

En este caso de uso un usuario autenticado como customer accedemos a mis suscripciones a volumen y editamos uno.

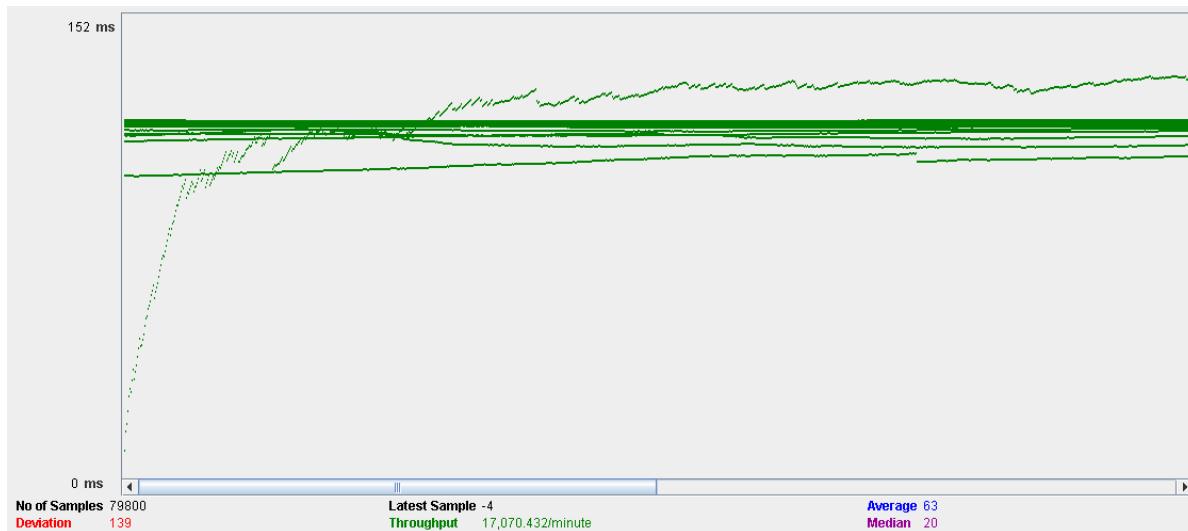
Menú Principal > Entrar > Cliente > Mis suscripciones a volúmenes > Editar

El límite de este caso de uso se sitúa en 140 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	12600	14	12	30	0	483	0.00%	44.9/sec	239.0
/scripts/jquery.min...	4200	35	26	78	0	731	0.00%	15.4/sec	1653.3
/styles/style.css	4200	14	10	30	0	507	0.00%	15.4/sec	24.2
/styles/bootstrap....	4200	37	28	81	0	653	0.00%	15.4/sec	1829.9
/styles/cookiecon...	4200	13	10	29	0	303	0.00%	15.4/sec	63.8
/scripts/helpers.js	4200	14	11	31	0	425	0.00%	15.4/sec	35.3
/scripts/bootstrap....	4200	22	17	49	0	539	0.00%	15.4/sec	562.8
/scripts/cookiecon...	4200	20	16	43	0	493	0.00%	15.4/sec	529.6
/scripts/polyfills.js	4200	13	11	29	0	489	0.00%	15.4/sec	116.2
/images/newspaper...	4200	35	25	76	0	616	0.00%	15.4/sec	2350.3
/security/login.do	4200	16	13	33	0	536	0.00%	15.4/sec	85.5
/j_spring_security...	4200	149	75	390	0	1834	0.00%	15.4/sec	91.9
/subscriptionVolu...	8400	181	116	415	0	2346	0.00%	30.6/sec	233.9
/subscriptionVolu...	8400	195	102	496	0	2528	0.00%	30.6/sec	248.1
/j_spring_security...	4200	28	23	55	0	465	0.00%	15.4/sec	83.1
TOTAL	79800	63	20	157	0	2528	0.00%	284.5/sec	7918.6

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo 0,9 segundos cuando editas la suscripción y vuelves al listado de tus suscripciones, por lo que con esta cantidad es aceptable, pero se va acercando al límite.

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 17070 veces por minuto.

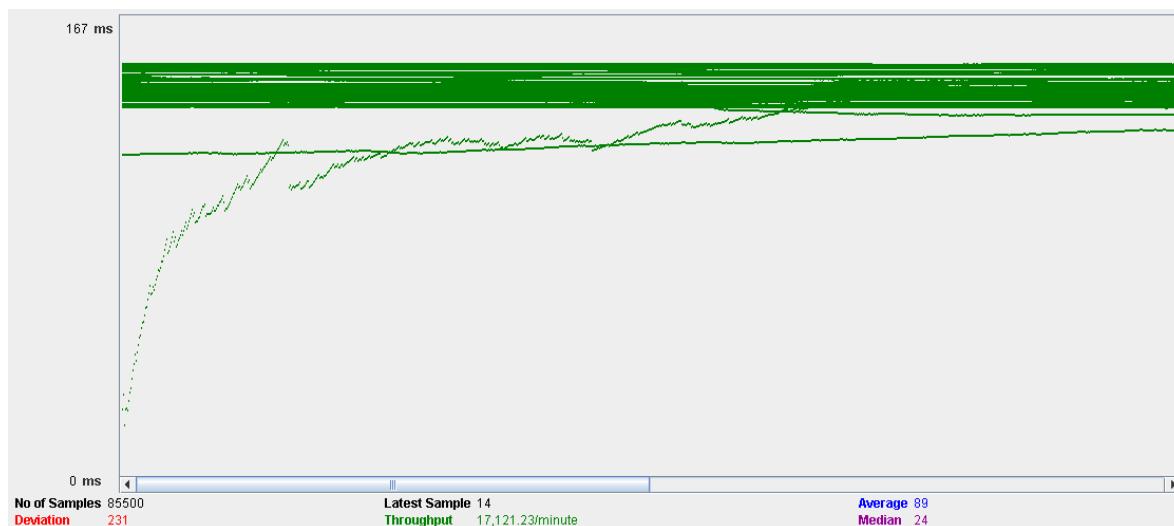


Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

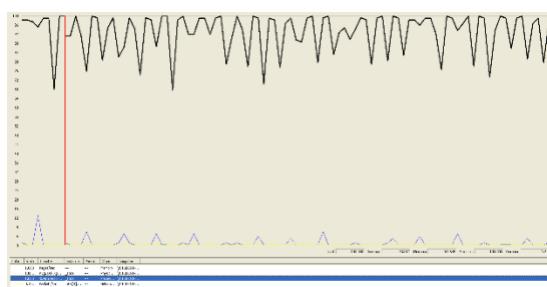
En este caso se ha utilizado 150 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	19	14	39	0	530	0.00%	45.1/sec	239.8
/scripts/jquery.min...	4500	42	30	89	0	632	0.00%	15.4/sec	1654.9
/styles/style.css	4500	18	12	38	0	466	0.00%	15.4/sec	24.2
/styles/bootstrap....	4500	45	31	98	0	633	0.00%	15.4/sec	1831.8
/styles/cookiecon...	4500	18	13	36	0	514	0.00%	15.4/sec	63.9
/scripts/helpers.js	4500	18	13	39	0	537	0.00%	15.4/sec	35.3
/scripts/bootstrap....	4500	25	19	53	0	598	0.00%	15.4/sec	563.4
/scripts/cookiecon...	4500	23	17	49	0	566	0.00%	15.4/sec	530.2
/scripts/polyfills.js	4500	18	13	38	0	410	0.00%	15.4/sec	116.4
/images/newspap...	4500	44	32	91	0	521	0.00%	15.4/sec	2350.8
/security/login.do	4500	19	14	37	0	401	0.00%	15.4/sec	85.6
/j_spring_security...	4500	213	70	615	0	2933	0.00%	15.4/sec	92.0
/subscriptionVolu...	9000	263	118	709	0	4273	0.00%	30.7/sec	234.4
/subscriptionVolu...	9000	293	124	808	0	4731	0.00%	30.8/sec	248.3
/j_spring_security...	4500	37	28	74	0	480	0.00%	15.4/sec	83.3
TOTAL	85500	89	24	182	0	4731	0.00%	285.4/sec	7942.7

En este caso se pueden observar que el tiempo medio para la acción de editar es bastante alto ya que editar y volver al listado supone un tiempo medio de 1.5 segundos, por lo que con esta cantidad de usuarios el caso de uso es inadmisible. Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 17121.



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



## Borrar una suscripción a volumen

En este caso de uso un usuario autenticado como customer accedemos a mis suscripciones a volumen y borramos uno.

Menú Principal > Entrar > Cliente > Mis suscripciones a volúmenes > Editar > Borrar

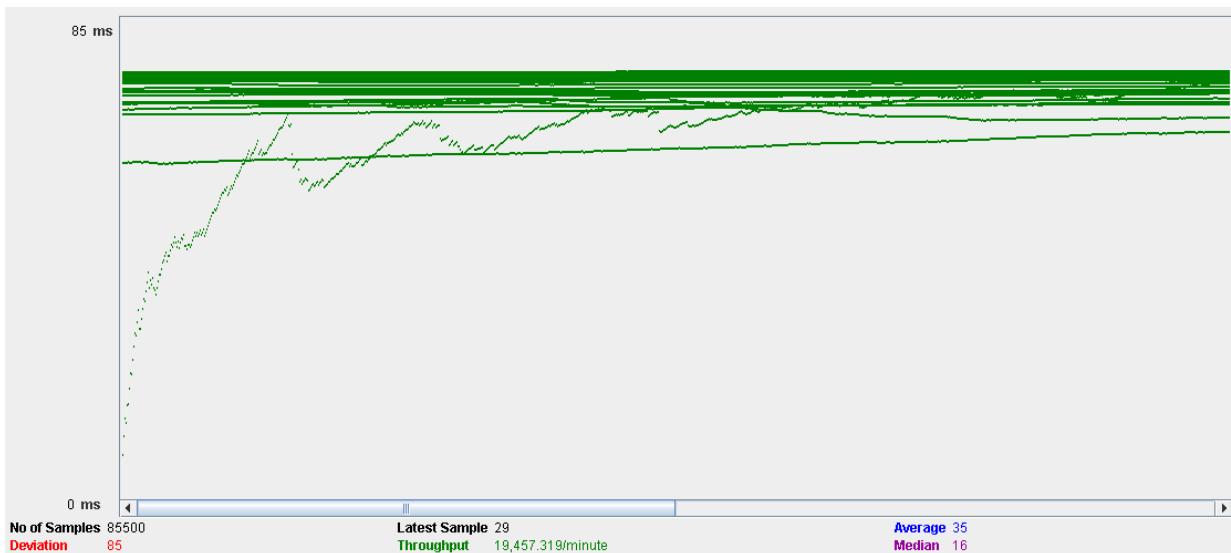
El límite de este caso de uso se sitúa en 150 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	13500	13	9	26	0	824	0.00%	51.2/sec	272.5
/scripts/jquery.min....	4500	32	24	71	0	805	0.00%	17.6/sec	1888.9
/styles/bootstrap....	4500	33	26	70	0	709	0.00%	17.6/sec	2090.6
/styles/cookiecon....	4500	12	9	27	0	349	0.00%	17.6/sec	73.0
/scripts/helpers.js	4500	11	9	26	0	315	0.00%	17.6/sec	40.3
/scripts/bootstrap....	4500	18	15	37	0	472	0.00%	17.6/sec	643.0
/styles/style.css	4500	11	8	26	0	298	0.00%	17.6/sec	27.6
/images/newspaper....	4500	38	28	82	0	484	0.00%	17.6/sec	2684.9
/scripts/cookiecon....	4500	16	13	35	0	439	0.00%	17.6/sec	605.1
/scripts/polyfills.js	4500	8	6	22	0	306	0.00%	17.6/sec	132.5
/security/login.do	4500	13	10	27	0	273	0.00%	17.6/sec	97.6
/j_spring_security....	4500	82	35	189	0	1831	0.00%	17.6/sec	104.8
/subscriptionVolu...	9000	106	49	246	0	2798	0.00%	34.9/sec	249.9
/subscriptionVolu...	9000	63	25	140	0	2004	0.00%	35.1/sec	386.7
/j_spring_security....	4500	24	21	46	0	306	0.00%	17.6/sec	94.8
TOTAL	85500	35	16	73	0	2788	0.00%	324.3/sec	9110.4

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo 0,4 segundos cuando borras la suscripción y vuelves al listado de tus suscripciones, por lo que con esta cantidad es completamente admisible

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 19457 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.

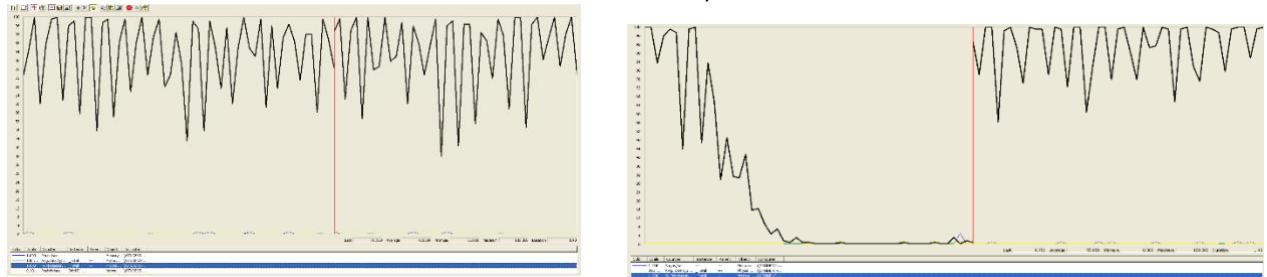
En este caso se ha utilizado 160 usuarios ejecutando nuevamente 30 acciones cada uno.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	14400	19	13	39	0	686	0.00%	47.8/sec	254.2
/scripts/jquery.min...	4800	47	35	97	0	884	0.00%	16.3/sec	1752.4
/styles/bootstrap....	4800	52	38	108	0	622	0.00%	16.3/sec	1939.4
/styles/cookiecon...	4800	22	13	49	0	2752	0.00%	16.3/sec	67.7
/scripts/helpers.js	4800	20	13	45	0	377	0.00%	16.3/sec	37.4
/scripts/bootstrap....	4800	29	21	61	0	354	0.00%	16.3/sec	596.5
/styles/style.css	4800	19	13	43	0	284	0.00%	16.3/sec	25.7
/images/newsnap...	4800	56	42	119	0	503	0.00%	16.3/sec	2490.6
/scripts/cookiecon...	4800	27	20	55	0	724	0.00%	16.3/sec	561.3
/scripts/polyfills.js	4800	14	9	29	0	617	0.00%	16.3/sec	123.2
/security/login.do	4800	25	16	50	0	619	0.00%	16.4/sec	90.8
/j_spring_security...	4800	243	115	686	0	2475	0.00%	16.4/sec	97.5
/subscriptionVolu...	9600	265	145	682	0	3195	0.00%	32.5/sec	232.8
/subscriptionVolu...	9600	169	65	475	0	2852	0.00%	32.7/sec	359.9
/j_spring_security...	4800	38	27	77	0	661	0.00%	16.3/sec	88.3
TOTAL	91200	80	24	181	0	3195	0.00%	302.6/sec	8500.2

En este caso se pueden observar que el tiempo medio para la acción de borrar supera lo admisible ya que borrar y volver al listado supone un tiempo medio de 1.2 segundos, por lo que con esta cantidad de usuarios el caso de uso es inadmisible. Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 18153.



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



## Añadir periódicos a un volumen

En este caso de uso un usuario autenticado como user accede a uno de sus volúmenes y le añade un periódico publicado

Menú Principal > Entrar > Usuario > Mis volúmenes > Display > Añadir un periódico >

Añadir al volumen

El límite de este caso de uso se sitúa en 160 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	14400	10	7	23	0	886	0.00%	42.0/sec	224.1
/scripts/jquery.min....	4800	24	18	50	0	1307	0.00%	14.4/sec	1543.6
/styles/cookiecon....	4800	8	6	22	0	232	0.00%	14.4/sec	59.6
/styles/bootstrap....	4800	24	19	53	0	482	0.00%	14.4/sec	1708.3
/styles/style.css	4800	8	6	21	0	448	0.00%	14.4/sec	22.6
/scripts/polyfills.js	4800	8	6	22	0	413	0.00%	14.4/sec	108.5
/scripts/helpers.js	4800	8	6	22	0	363	0.00%	14.4/sec	32.9
/scripts/bootstrap....	4800	13	10	30	0	755	0.00%	14.4/sec	525.3
/scripts/cookiecon....	4800	11	9	27	0	196	0.00%	14.4/sec	494.3
/images/newspap....	4800	29	22	60	0	486	0.00%	14.4/sec	2193.4
/security/login.do	4800	11	8	24	0	531	0.00%	14.4/sec	80.1
/j_spring_security....	4800	39	25	79	0	1483	0.00%	14.4/sec	86.8
/volume/userlist.do	4800	28	19	57	0	653	0.00%	14.4/sec	95.7
/volume/display.do	9600	36	21	66	0	2126	0.00%	28.7/sec	230.6
/newspaper/user/...	4800	44	26	85	0	2785	0.00%	14.5/sec	98.0
/volume/user/add...	4800	38	23	76	0	1968	0.00%	14.5/sec	199.6
/j_spring_security....	4800	20	16	41	0	511	0.00%	14.5/sec	78.2
TOTAL	96000	21	13	43	0	2785	0.00%	279.9/sec	7568.4

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo 0,27 segundos cuando cargas los datos de estilo cuando inicias la aplicación, por lo que con esta cantidad es completamente admisible

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 16793 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 170 usuarios ejecutando nuevamente 30 acciones cada

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	15300	120	39	215	3	3950	0.00%	28.4/sec	151.7
/scripts/jquery.min.js	5100	169	77	376	6	2931	0.00%	9.7/sec	1035.7
/styles/cookiecons...	5100	117	34	216	2	3362	0.00%	9.7/sec	40.0
/styles/bootstrap.m...	5100	195	79	451	6	4100	0.00%	9.7/sec	1146.1
/styles/style.css	5100	109	33	196	2	4639	0.00%	9.7/sec	15.1
/scripts/polyfills.js	5100	112	34	195	2	3889	0.00%	9.7/sec	72.8
/scripts/helpers.js	5100	97	32	173	2	3343	0.00%	9.7/sec	22.1
/scripts/bootstrap....	5100	120	46	237	3	3862	0.00%	9.7/sec	352.4
/scripts/cookiecon...	5100	118	44	213	3	5205	0.00%	9.7/sec	331.6
/images/newspap...	5100	218	90	520	8	3719	0.00%	9.7/sec	1471.4
/security/login.do	5100	133	39	249	4	3858	0.00%	9.7/sec	53.6
/j_spring_security_...	5100	751	414	1958	10	7514	0.00%	9.7/sec	58.1
/volume/user/list.do	5100	492	244	1290	9	5955	0.00%	9.6/sec	63.9
/volume/display.do	10200	537	237	1447	7	6055	0.00%	19.2/sec	154.2
/newspaper/service...	5100	577	293	1509	12	5132	0.00%	9.6/sec	65.3
/volume/user/addN...	5100	548	259	1475	8	5022	0.00%	9.6/sec	133.0
/j_spring_security_...	5100	318	97	975	6	4674	0.00%	9.6/sec	52.1
TOTAL	102000	275	73	831	2	7514	0.00%	189.5/sec	5124.1

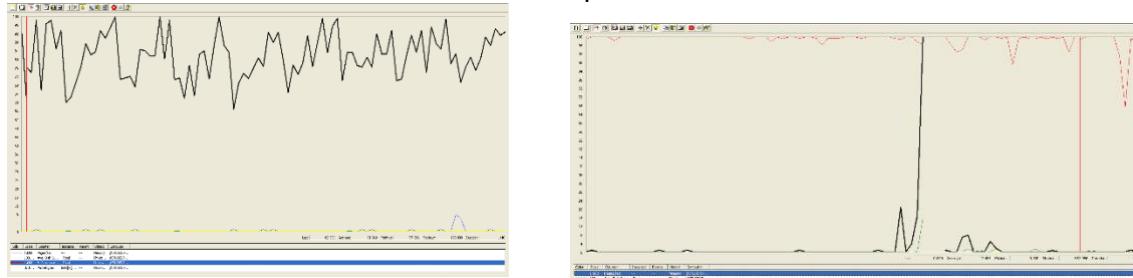
uno.

En este caso se pueden observar que el tiempo medio para la acción de añadir un periódico supera lo admisible ya que añadir y volver al listado supone un tiempo medio de 2.7 segundos, por lo que con esta cantidad de usuarios el caso de uso es inadmisible

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 11369.



Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador y la memoria. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones



mentionadas anteriormente, en el mismo orden.

## Borrar periódicos de un volumen

En este caso de uso un usuario autenticado como user accede a uno de sus volúmenes y le eliminamos uno de sus periódicos

Menú Principal > Entrar > Usuario > Mis volúmenes > Display > Borrar un periódico >

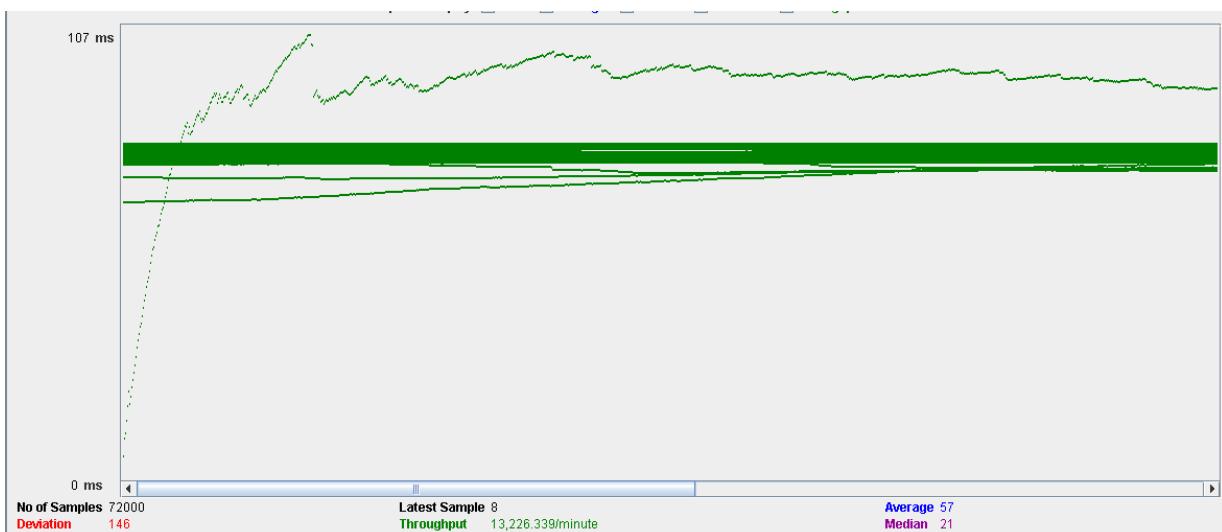
Borrar del volumen

El límite de este caso de uso se sitúa en 120 usuarios concurrentes ejecutando 30 veces la misma acción.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	10800	18	14	34	3	1166	0.00%	33.1/sec	176.4
/styles/style.css	3600	15	11	31	1	584	0.00%	11.3/sec	17.7
/scripts/jquery.min.js	3600	40	26	86	5	1227	0.00%	11.3/sec	1216.1
/styles/bootstrap.m...	3600	40	26	87	6	1032	0.00%	11.3/sec	1345.9
/scripts/cookiecons...	3600	15	11	29	1	843	0.00%	11.3/sec	46.9
/scripts/polyfills.js	3600	14	11	28	1	305	0.00%	11.3/sec	85.4
/scripts/bootstrap....	3600	22	15	46	2	1107	0.00%	11.3/sec	413.9
/scripts/helpers.js	3600	14	10	28	1	858	0.00%	11.3/sec	25.9
/scripts/cookiecon...	3600	20	15	41	3	1194	0.00%	11.3/sec	389.4
/images/newspaper...	3600	47	29	99	7	989	0.00%	11.3/sec	1727.4
/security/login.do	3600	21	15	35	4	1087	0.00%	11.3/sec	62.9
/j_spring_security_...	3600	142	64	324	9	2725	0.00%	11.3/sec	68.1
/volume/user/list.do	3600	140	58	308	8	3646	0.00%	11.3/sec	75.1
/volume/display.do	7200	136	57	290	8	2579	0.00%	22.6/sec	168.0
/newspaper/user/d...	3600	124	52	281	7	2589	0.00%	11.3/sec	144.0
/volume/user/delete...	3600	124	57	256	8	2644	0.00%	11.3/sec	156.4
/j_spring_security_...	3600	34	27	65	5	779	0.00%	11.3/sec	61.1
TOTAL	72000	57	21	110	1	3646	0.00%	220.4/sec	6012.3

Como podemos observar, todas las acciones suponen un tiempo medio muy bajo, como máximo 0,55 segundos cuando borras un periódico y vuelves al display del volumen, por lo que con esta cantidad es completamente admisible

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 13226 veces por minuto.



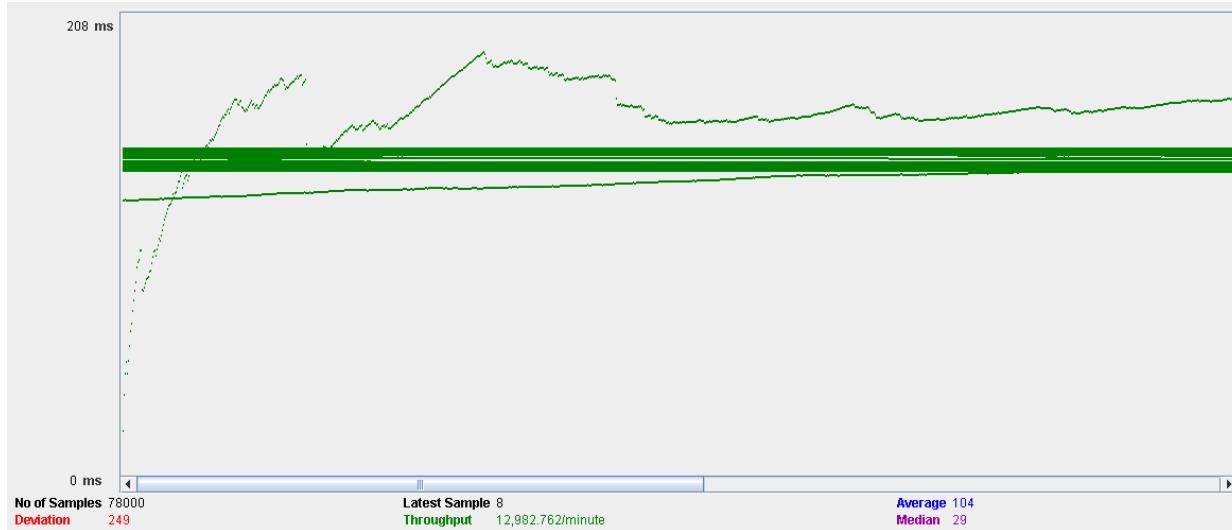
Por otro lado, vamos a presentar un caso negativo que se encuentra cercano al límite para ver la diferencia y como se comportaría nuestro sistema.

En este caso se ha utilizado 130 usuarios ejecutando nuevamente 30 acciones cada uno.

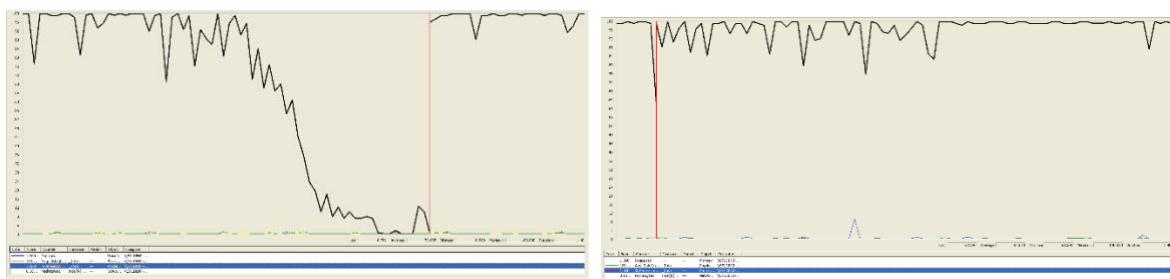
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	11700	26	17	46	3	3778	0.00%	32.5/sec	173.2
/styles/style.css	3900	21	15	41	1	505	0.00%	11.1/sec	17.4
/scripts/jquery.min.js	3900	53	34	117	5	1239	0.00%	11.1/sec	1190.9
/styles/bootstrap.m...	3900	52	34	116	5	756	0.00%	11.1/sec	1317.9
/styles/cookiecon...	3900	19	13	38	1	1094	0.00%	11.1/sec	46.0
/scripts/polyfills.js	3900	18	14	37	2	712	0.00%	11.1/sec	83.7
/scripts/bootstrap...	3900	28	19	61	3	744	0.00%	11.1/sec	405.2
/scripts/helpers.js	3900	18	13	38	1	420	0.00%	11.1/sec	25.4
/scripts/cookiecon...	3900	26	18	55	3	887	0.00%	11.1/sec	381.3
/images/newspaper...	3900	62	38	143	6	921	0.00%	11.1/sec	1691.4
/security/login.do	3900	26	19	49	4	510	0.00%	11.1/sec	61.6
/j_spring_security...	3900	289	144	742	8	3581	0.00%	11.1/sec	66.8
/volume/userlist.do	3900	273	131	720	9	3687	0.00%	11.1/sec	73.6
/volume/display/do	7800	272	116	726	8	3615	0.00%	22.1/sec	164.6
/newspaper/userid...	3900	269	116	717	7	3322	0.00%	11.1/sec	141.0
/volume/user/delete...	3900	249	122	664	8	3620	0.00%	11.1/sec	153.1
/j_spring_security...	3900	48	38	92	6	1213	0.00%	11.1/sec	59.9
TOTAL	78000	104	29	225	1	3778	0.00%	216.4/sec	5902.0

En este caso se pueden observar que el tiempo medio para la acción de borrar un periódico supera lo admisible ya que borrar y volver al listado supone un tiempo medio de 1.3 segundos, por lo que con esta cantidad de usuarios el caso de uso es inadmisible

Nuevamente, abajo nos encontramos una gráfica que mide el throughput, en este caso de 12982.



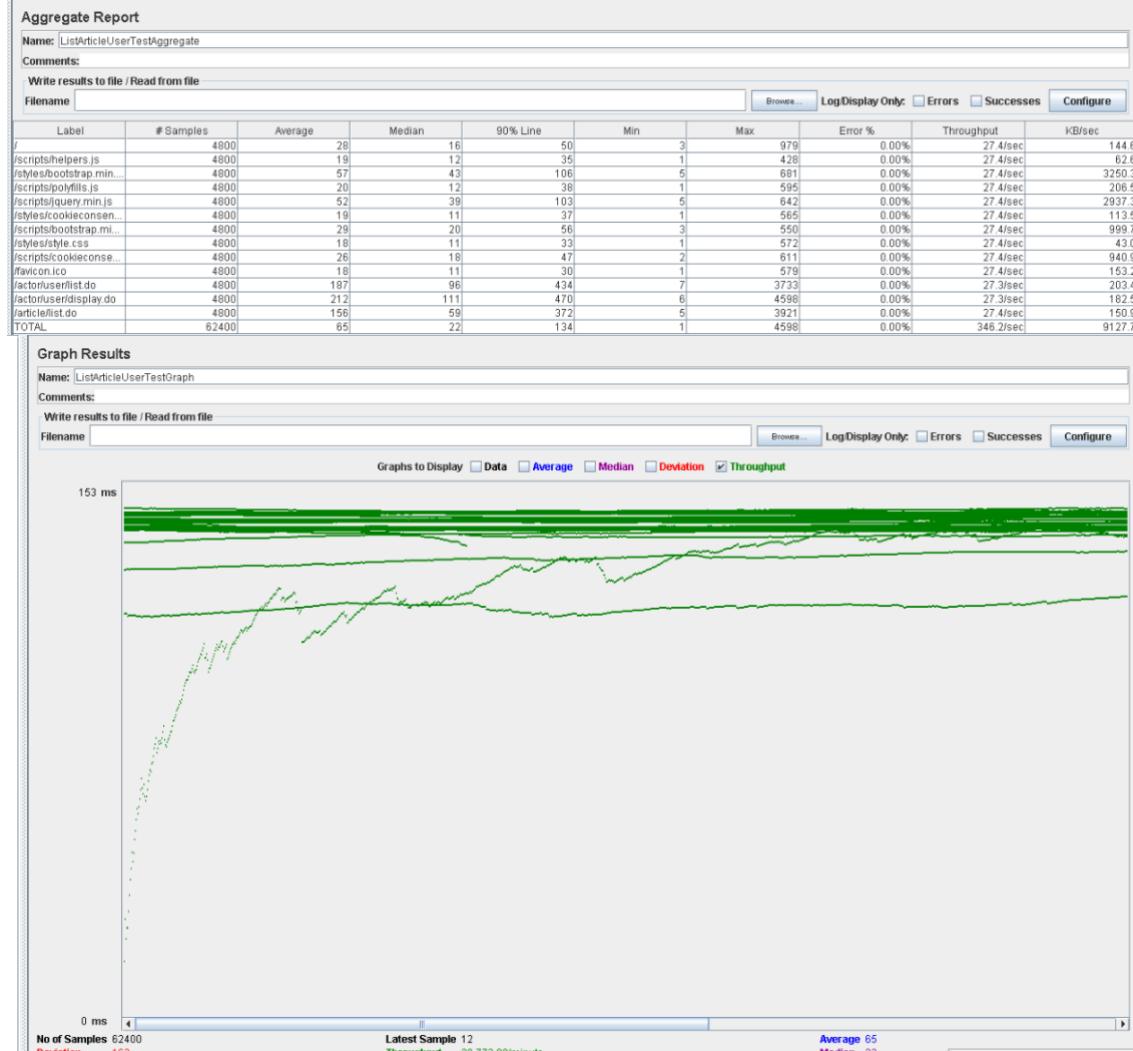
Nuestro sistema se comporta así en este caso debido a las limitaciones del procesador. Adjuntamos dos gráficas en las que se ve el comportamiento de ambos componentes en las dos situaciones mencionadas anteriormente, en el mismo orden.



## Listar artículos de un usuario.

En este caso de uso, accedemos a la página, nos logueamos como un usuario, accedemos a la lista de artículos de un usuario y nos deslogueamos.

La configuración en esta ocasión es de 160 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, aunque las esperas son algo considerables, lo que podemos ver en el Aggregate Report y en el Graph Results:

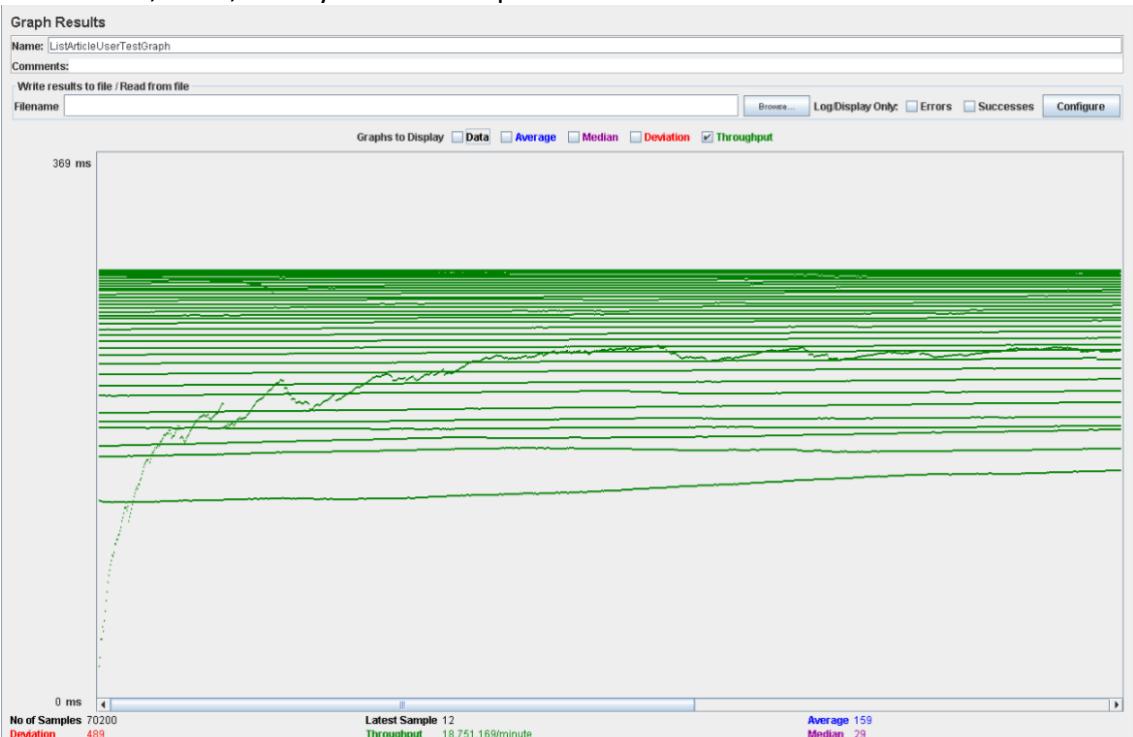


En esta tabla, el rendimiento es de aproximadamente 22.700 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora probamos una configuración de 180 usuarios, 1 ramp-up y 30 acciones cada uno. Las gráficas correspondientes a este test son:

Aggregate Report										
Name: ListArticleUserTestAggregate										
Comments:										
Write results to file / Read from file										
Filename	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/scripts/helpers.js	5400	62	21	100	1	1982	0.02%	24.6/sec	129.9	
/styles/bootstrap.min....	5400	34	16	57	2	1355	0.02%	24.6/sec	56.3	
/scripts/jquery.js	5400	75	53	140	1	2060	0.02%	24.6/sec	2917.2	
/scripts/jquery.min.js	5400	36	15	61	2	1224	0.02%	24.6/sec	185.3	
/scripts/jquery.min.js	5400	74	50	130	2	2059	0.02%	24.6/sec	2636.1	
/styles/bootstrap.mi...	5400	30	15	49	1	1479	0.00%	24.6/sec	101.9	
/scripts/bootstrap.mi...	5400	41	26	70	2	1144	0.00%	24.6/sec	897.3	
/styles/style.css	5400	29	15	45	1	1126	0.00%	24.6/sec	38.7	
/scripts/cookieconsen...	5400	95	22	59	3	1004	0.00%	24.6/sec	844.4	
/favicon.ico	5400	33	16	52	2	1542	0.00%	24.6/sec	137.7	
/action/user/list.do	5400	602	254	1350	1	11583	0.02%	24.6/sec	182.8	
/action/user/display.d...	5400	567	260	1312	1	11859	0.02%	25.2/sec	168.3	
/article/list.do	5400	449	179	1145	1	14821	0.02%	25.4/sec	140.0	
TOTAL	70200	159	29	355	1	14821	0.01%	312.5/sec	8238.9	

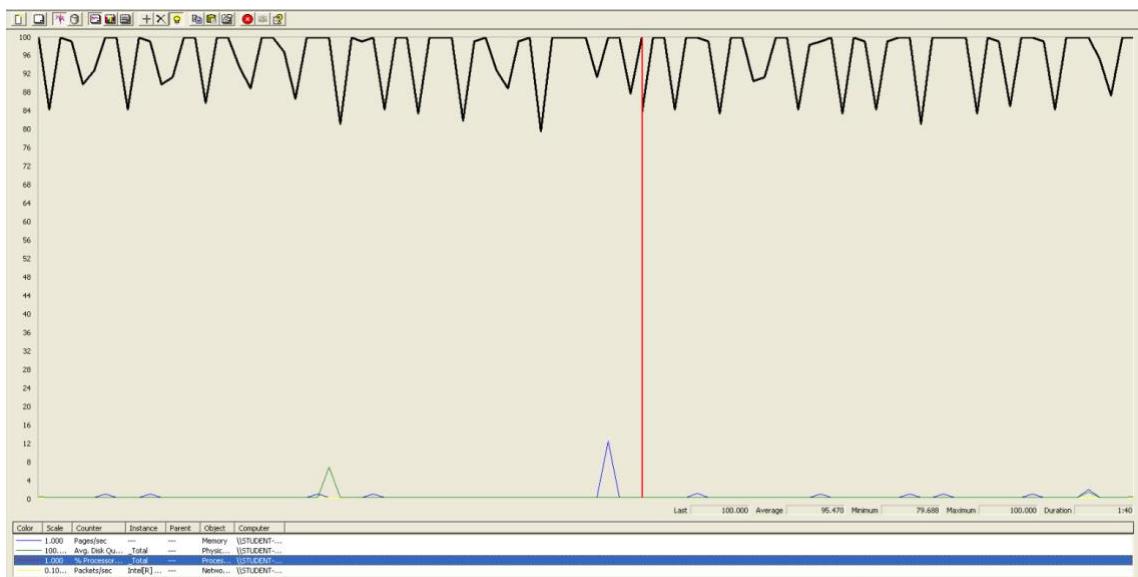
Como podemos observar fijándonos en 90% Line, las direcciones actor/user/list.do, /actor/user/display.do y /article/list.do generan un tiempo de respuesta mayor a lo establecido, 1350, 1312 y 1145 ms respectivamente.



Ya se producen muy leves errores y los tiempos de espera. En esta tabla, el rendimiento es de aproximadamente 18.700 ejecuciones por minuto (desciende al aumentar los 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.

Por lo que podemos afirmar que la aplicación soporta entre 180 y 200 usuarios en este caso de uso. Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos el procesador tiene porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de la CPU.

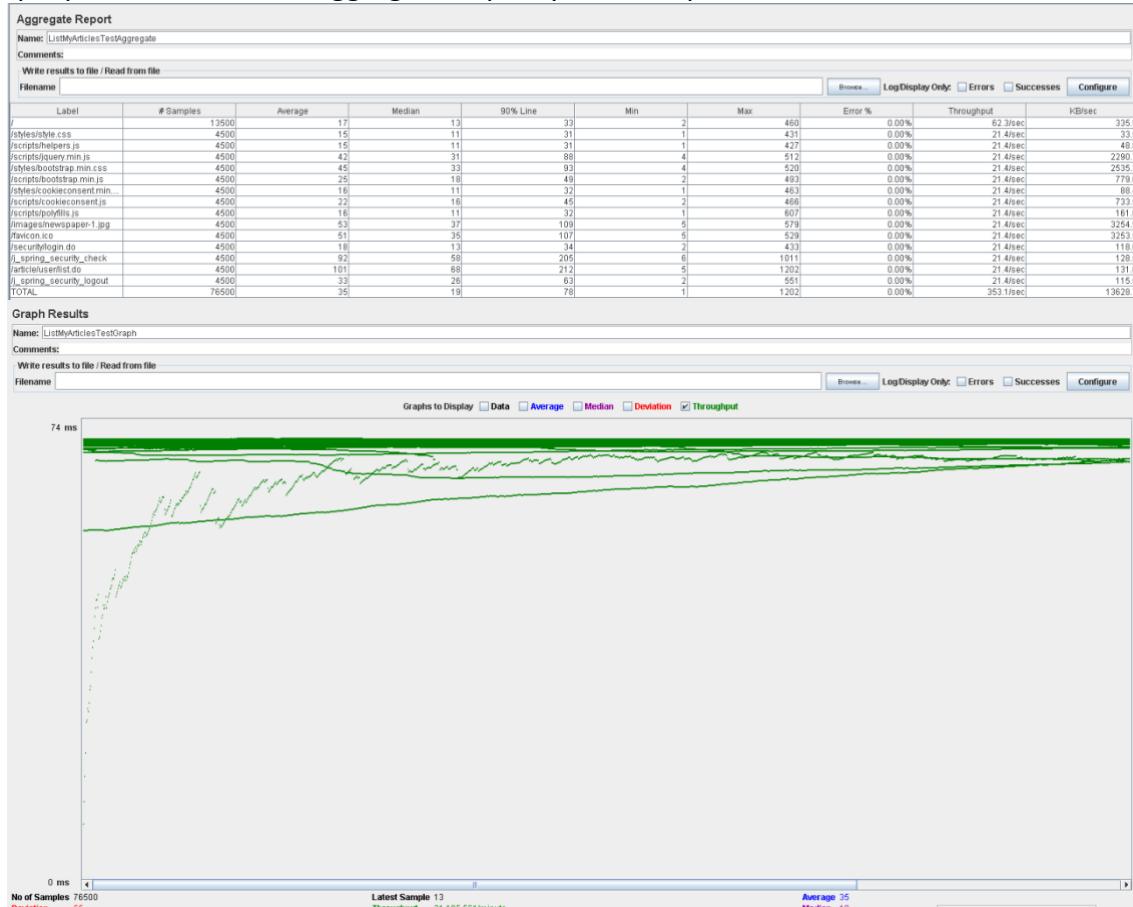


Performance con Processor Time.

## Listar mis artículos.

En este caso de uso, accedemos a la página, nos autentificamos como usuario, pulsamos Mis artículos y nos desautentificamos.

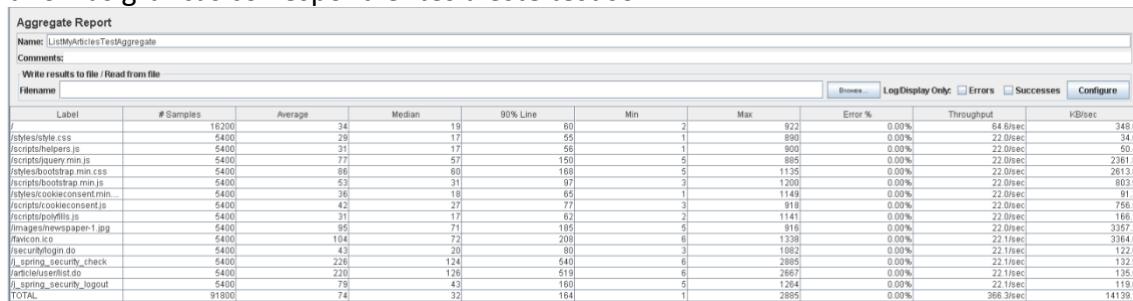
La configuración en esta ocasión es de 150 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose muy leves errores, pero unos tiempos de espera considerables, los que podemos ver en el Aggregate Report y en el Graph Results:



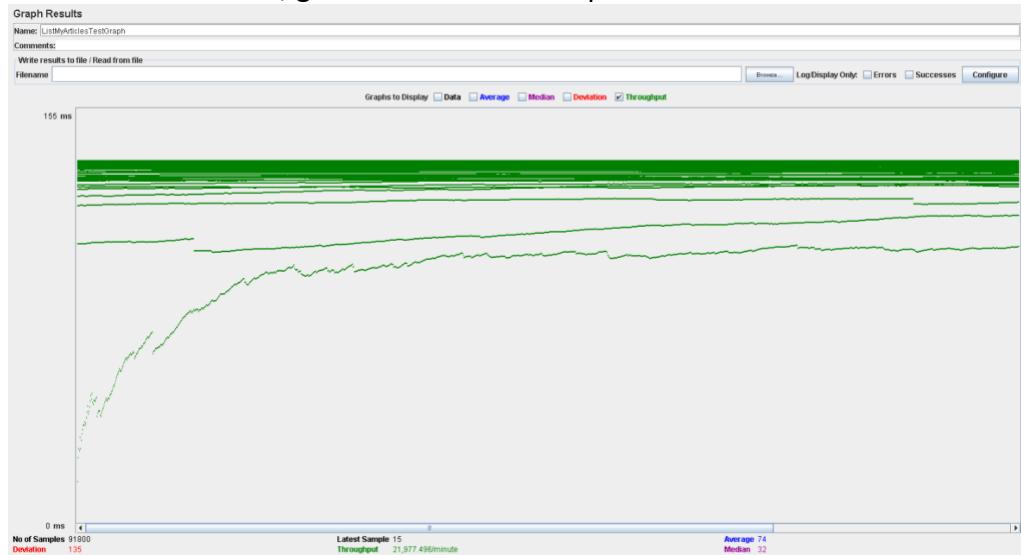
En esta tabla, el rendimiento es de aproximadamente 21.200 ejecuciones por minuto.

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.

Ahora probamos una configuración de 180 usuarios, 1 ramp-up y 30 acciones cada uno. Las gráficas correspondientes a este test son:



Como podemos observar fijándonos en 90% Line, las direcciones /j\_spring\_security\_check y /article/user/list.do generan un tiempo de respuesta (540 y 519 ms) mayor a lo esperado en un caso de uso de listado, además estos tiempos sumados a otros del tests, generan demasiada espera.



En esta tabla, el rendimiento es de aproximadamente 22.000 ejecuciones por minuto (aumenta brevemente al aumentar los 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.

Al aumentar el tiempo de espera, este caso de uso soporta entre 150 y 180 usuarios.

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.

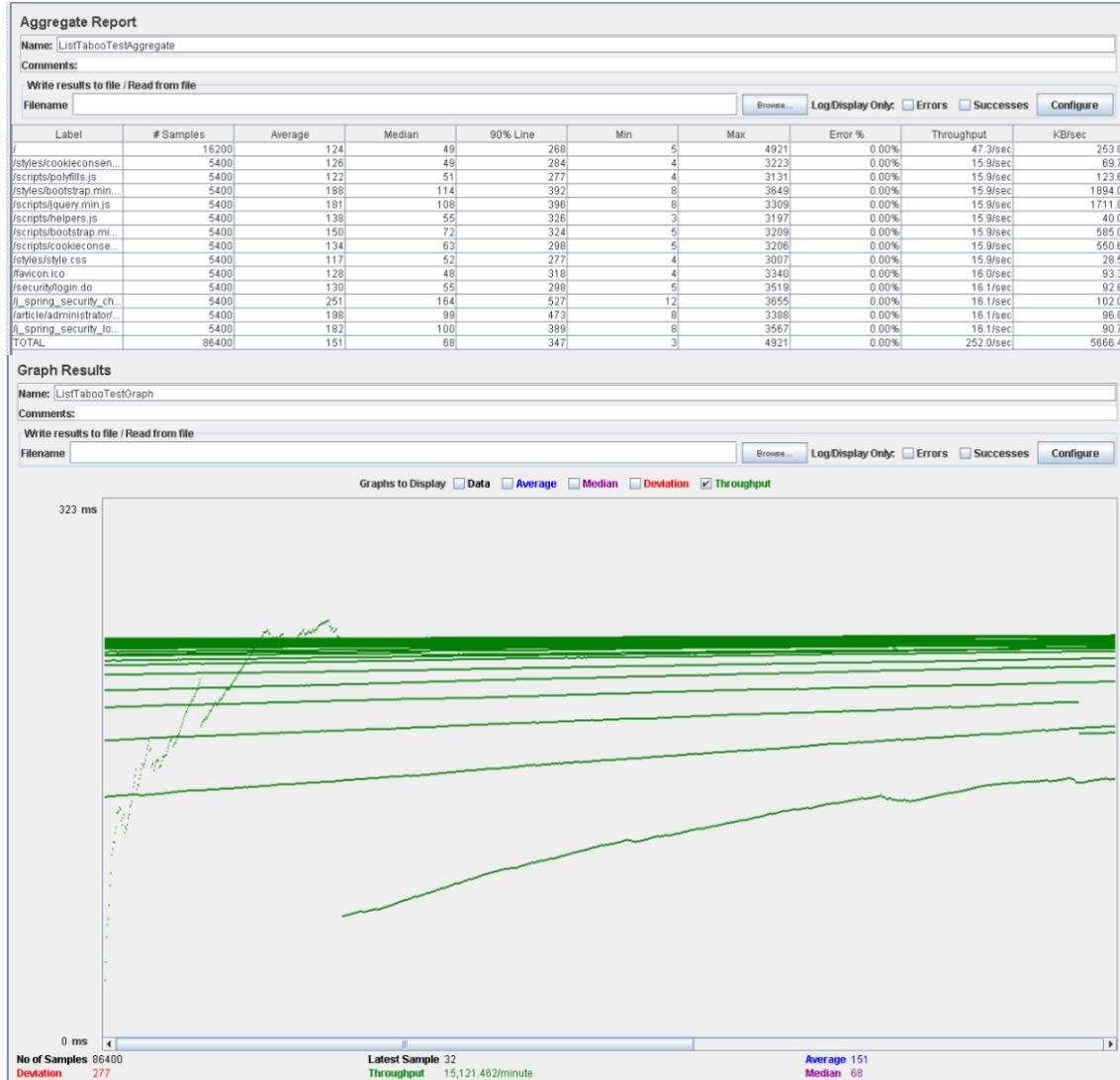


Performance con Processor Time.

## Listar artículos tabús.

En este caso de uso, accedemos a la página, nos logueamos como un administrador, accedemos a la lista de artículos tabús y nos deslogueamos.

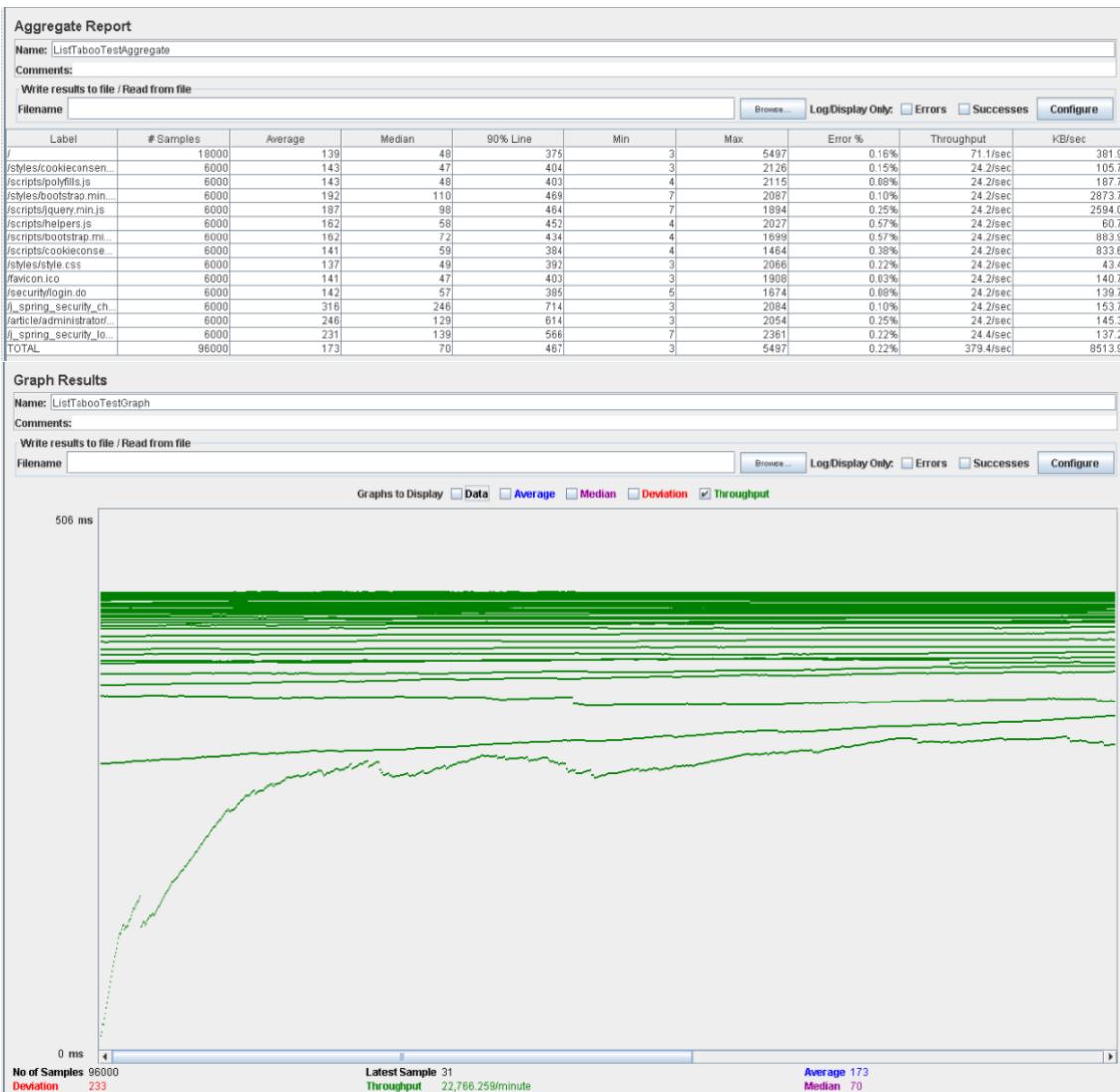
La configuración en esta ocasión es de 180 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, que podemos ver en el Aggregate Report y en el Graph Results:



En esta tabla, el rendimiento es de aproximadamente 15.100 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.

En un caso real, no sería normal encontrar un número tan alto de administradores en la aplicación. No obstante, hemos considerado suficiente ver un valor alto para así tener una estimación bastante buena tanto de los errores como de los tiempos de ejecución.

Ahora probamos una configuración de 200 usuarios, 1 ramp-up y 30 acciones cada uno, obteniéndose ahora errores, por lo que entre 180 y 200 son los usuarios permitidos, aunque en un caso de uso real no existirían tantos administradores al mismo tiempo. Las gráficas correspondientes a este test son:



En esta tabla, el rendimiento es de aproximadamente 22.700 ejecuciones por minuto (aumentando a pesar de los errores). 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.

En este caso, fijándonos en 90% Line, las direcciones /j\_spring\_security\_check y /article/administrator/listTaboo.do generan un tiempo de respuesta mayor al resto, aunque en todas las direcciones se produce unos tiempos de respuesta demasiado altos.

Ahora estudiaremos a que se deben los tiempos y errores:

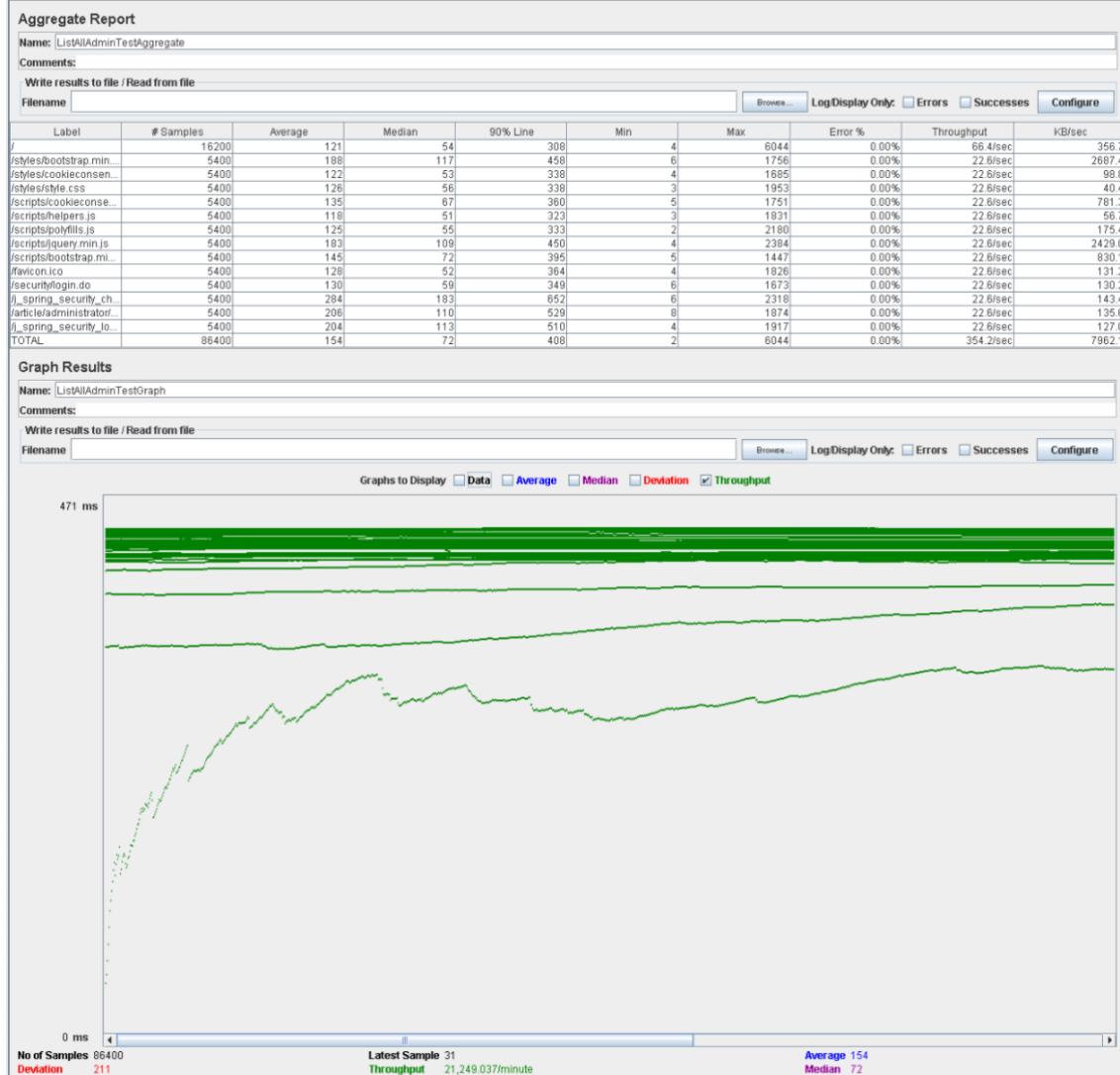
Abrimos performance.exe y vemos la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



Performance con Processor Time.

## Listar todos los artículos administrador.

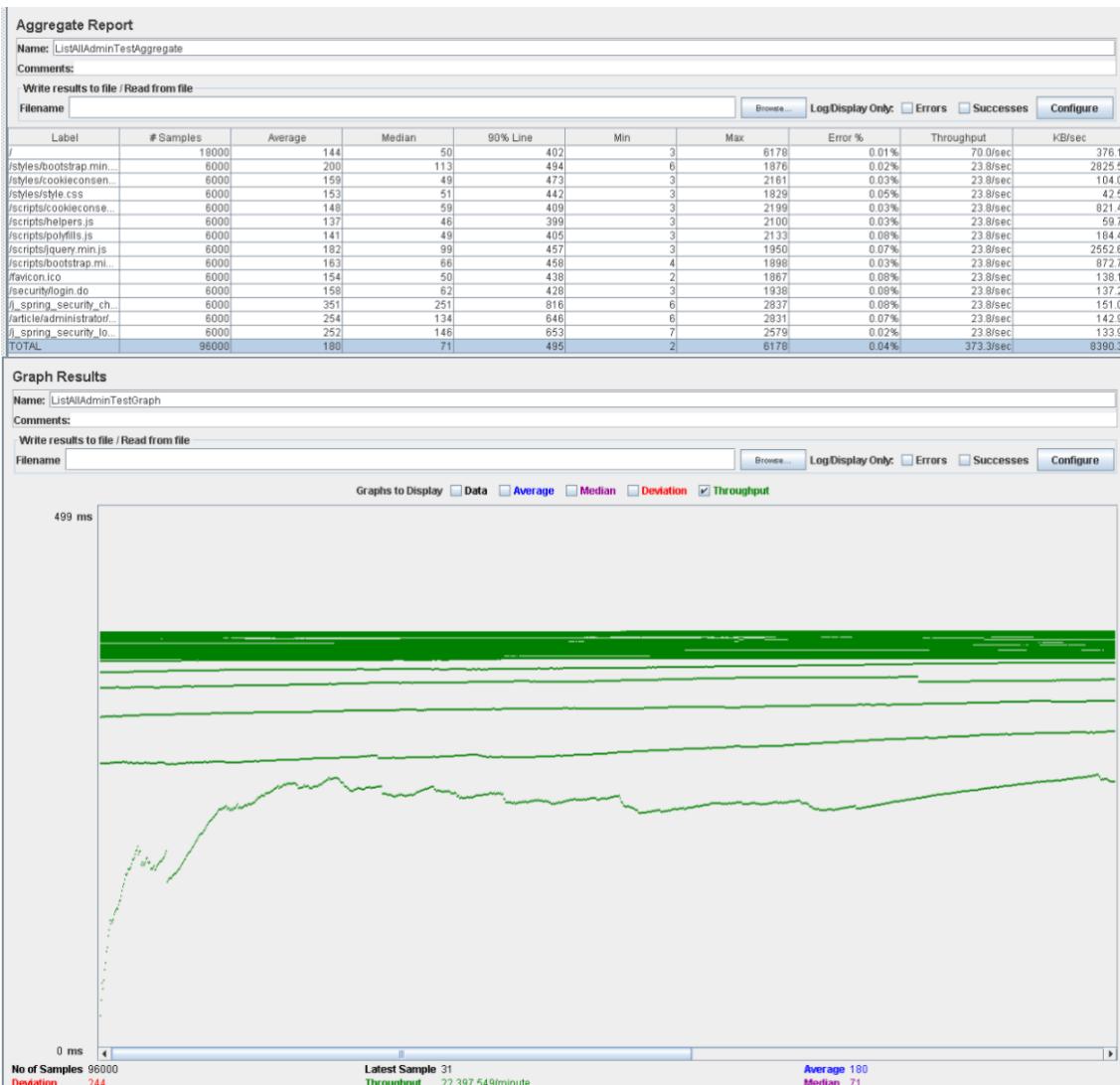
En este caso de uso, accedemos a la página, nos logueamos como un administrador, accedemos a la lista de artículos completa de un administrador y nos deslogueamos. La configuración en esta ocasión es de 180 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero esperas demasiado altas, que podemos ver en el Aggregate Report y en el Graph Results:



En esta tabla, el rendimiento es de aproximadamente 21.200 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.

En un caso real, no sería normal encontrar un número tan alto de administradores en la aplicación. No obstante, hemos considerado suficiente ver un valor alto para así tener una estimación bastante buena tanto de los errores como de los tiempos de ejecución.

Ahora probamos una configuración de 200 usuarios, 1 ramp-up y 30 acciones cada uno, obteniéndose ahora errores, por lo que entre 180 y 200 son los usuarios permitidos, aunque en un caso de uso real no existirían tantos administradores al mismo tiempo. Las gráficas correspondientes a este test son:

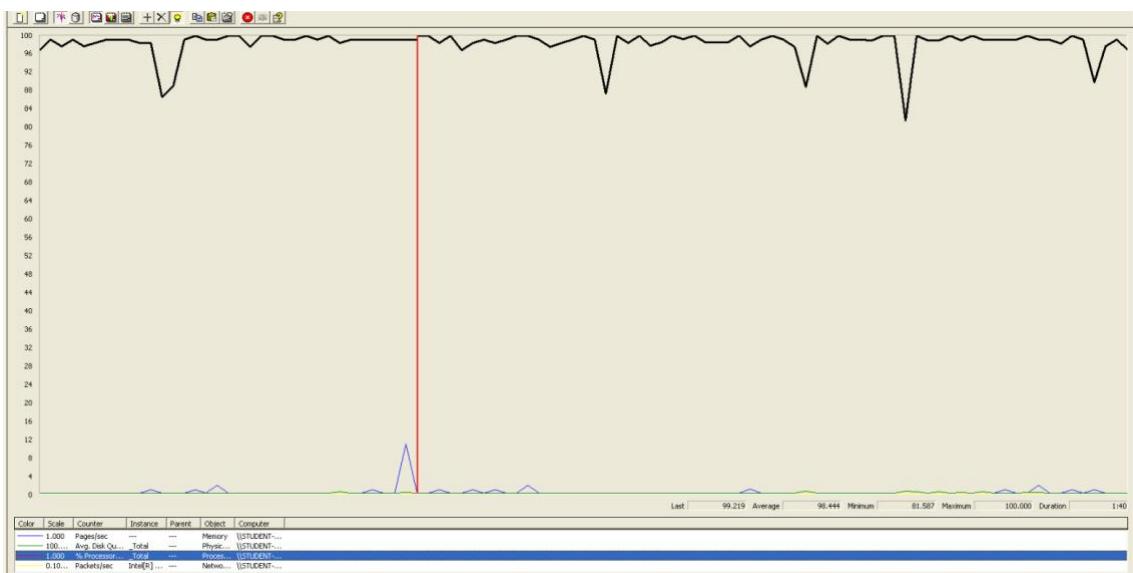


En esta tabla, el rendimiento es de aproximadamente 22.400 ejecuciones por minuto (aumentando brevemente al aumentar los usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

En este caso, fijándonos en 90% Line, las direcciones /j\_spring\_security\_check, /article/administrator/list.do y j\_spring\_security\_logout generan un tiempo de respuesta mayor al resto, aunque en todas las direcciones se produce unos tiempos de respuesta demasiado altos.

Por lo que podemos afirmar que la aplicación soporta entre 180 y 200 usuarios en este caso de uso. Ahora estudiaremos a qué se deben los tiempos y errores:

Abrimos performance.exe y vemos el procesador tiene porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de la CPU.

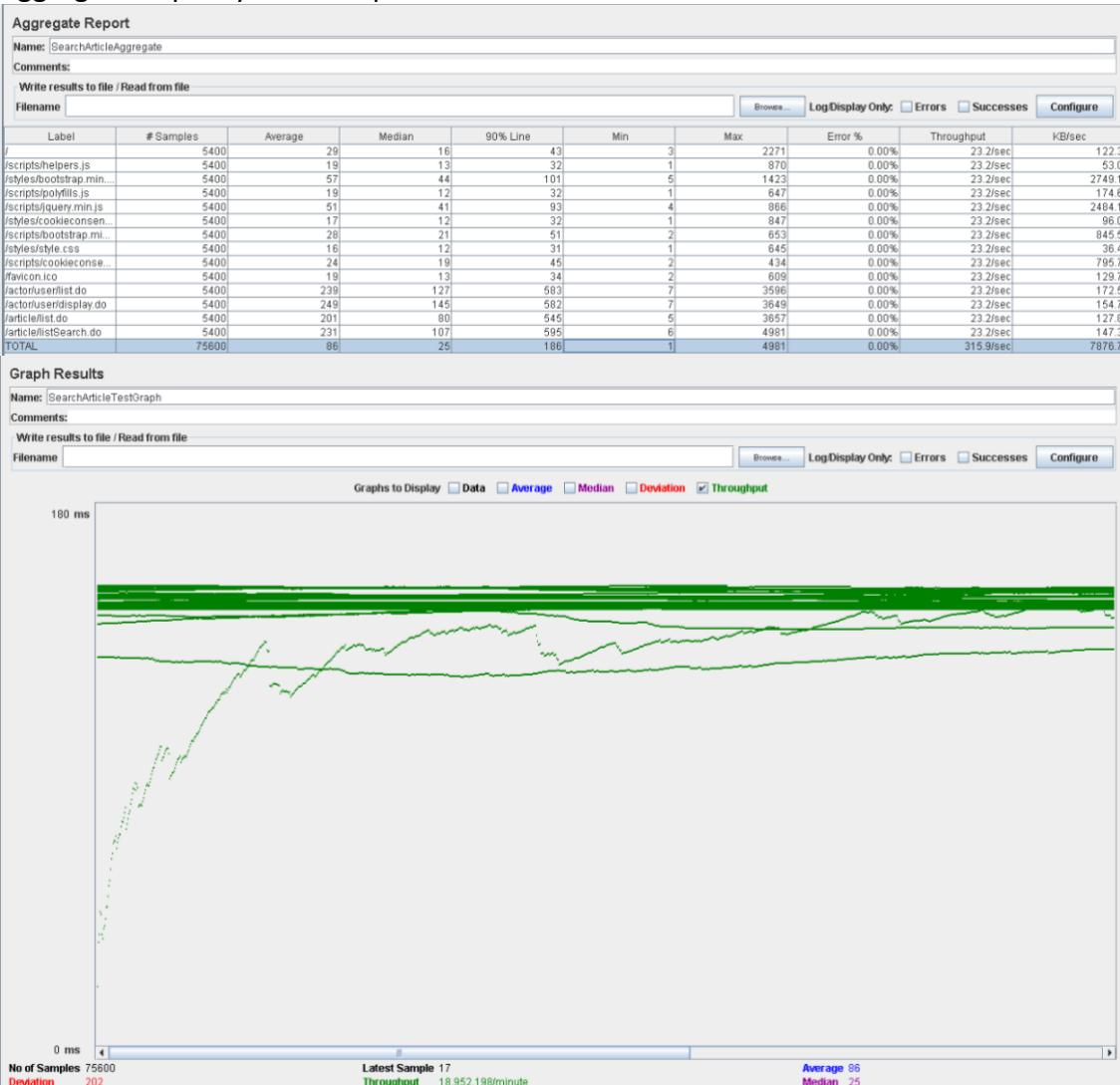


Performance con Processor Time.

## Buscar artículos.

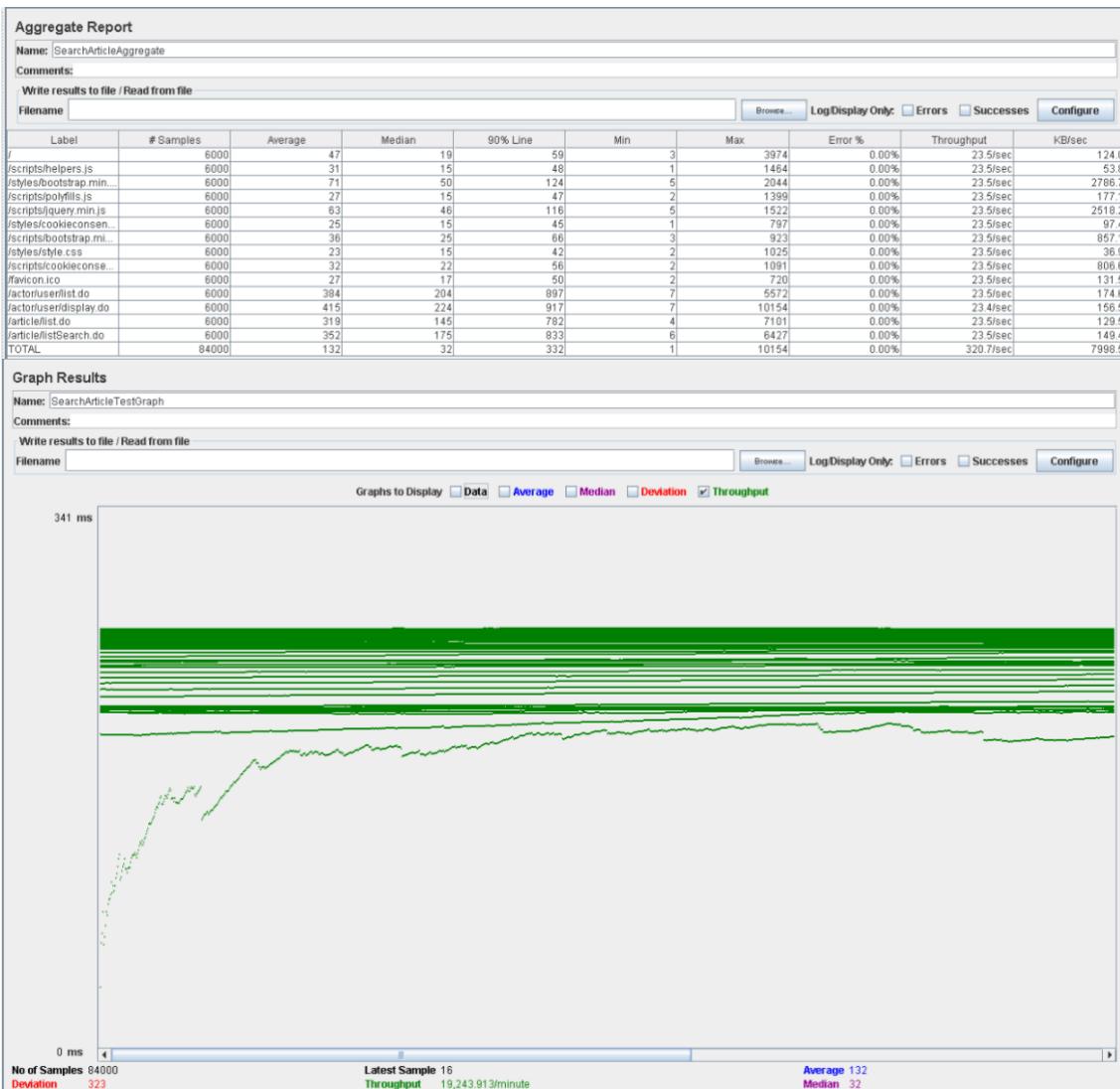
En este caso de uso, accedemos a la aplicación, accedemos a la lista de usuarios, luego al display de uno de ellos, accedemos a Leer más artículos que nos llevará a la vista del listado de artículos de ese usuario, buscamos por un término clave y pasamos a la vista con los resultados. Este caso de uso se repite con todas las vistas de listado de artículos.

La configuración en esta ocasión es de 180 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores ni esperas demasiado altas, que podemos ver en el Aggregate Report y en el Graph Results:



En esta tabla, el rendimiento es de aproximadamente 19.900 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.

Ahora probamos una configuración de 200 usuarios, 1 ramp-up y 30 acciones cada uno, obteniéndose ahora esperas demasiado altas, por lo que entre 180 y 200 son los usuarios permitidos. Las gráficas correspondientes a este test son:



En esta tabla, el rendimiento es de aproximadamente 19.200 ejecuciones por minuto (descendiendo brevemente al aumentar los 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.

En este caso, fijándonos en 90% Line, las direcciones /actor/user/list.do, /actor/user/display.do, /article/list.do y /article/listSearch.do generan unos tiempos de respuesta demasiados altos (897, 917, 782 y 833 ms respectivamente).

Ahora estudiaremos a qué se deben los errores:

Abrimos performance.exe y vemos que la tarjeta de red tiene porcentajes muy altos, llegando al 100% en ocasiones, por lo que el problema se localizará en ella.

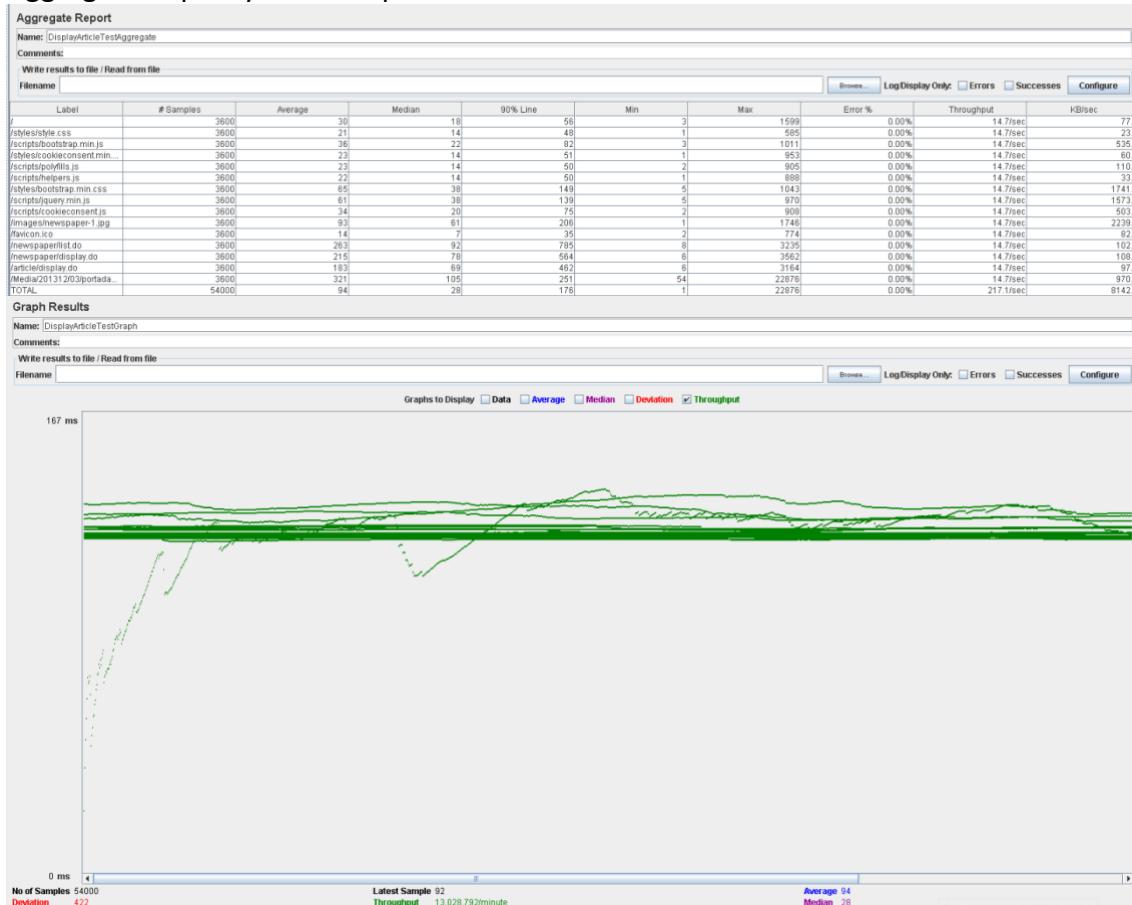


Performance con Packets/sec.

## Desplegar artículo.

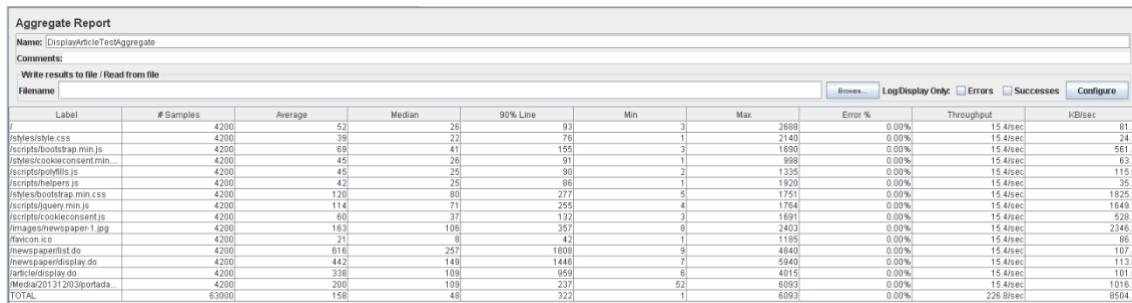
En este caso de uso, accedemos a la aplicación, accedemos a la lista de periódicos, luego al display de uno de ellos y accedemos al display de un artículo.

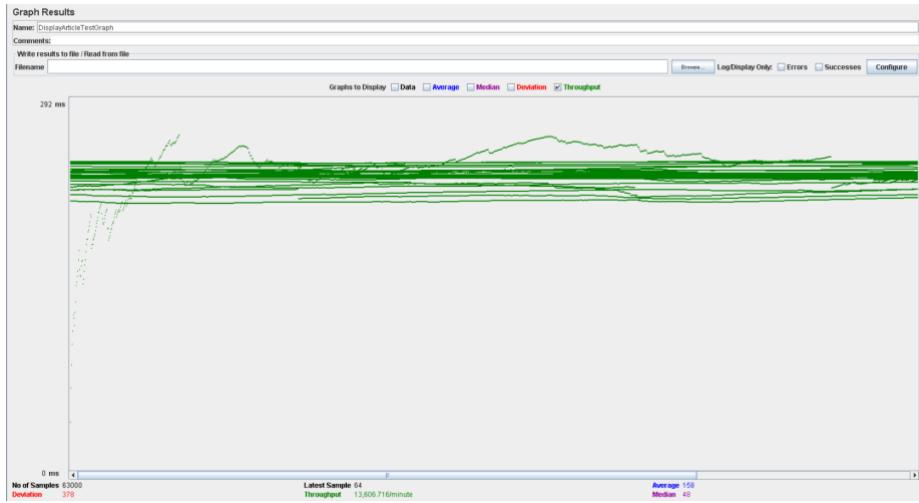
La configuración en esta ocasión es de 120 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores ni esperas demasiado altas, que podemos ver en el Aggregate Report y en el Graph Results:



En esta tabla, el rendimiento es de aproximadamente 13.000 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.

Ahora probamos una configuración de 140 usuarios, 1 ramp-up y 30 acciones cada uno, obteniéndose ahora esperas demasiado altas además de leves errores, por lo que entre 160 y 180 son los usuarios permitidos. Las gráficas correspondientes a este test son:





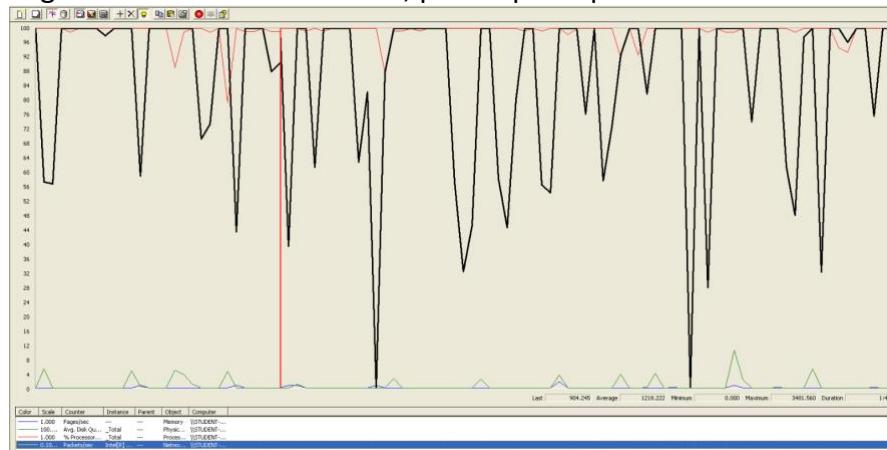
En esta tabla, el rendimiento es de aproximadamente 13.600 ejecuciones por minuto (aumenta brevemente al aumentar los usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

En este caso, fijándonos en 90% Line, las direcciones /newspaper/list.do, /newspaper/display.do y /article/display.do generan unos tiempos de respuesta demasiados altos: 1808, 1446 y 959 ms respectivamente.

Además, la dirección /Media/201312/03/portadas-periodicos--644x362.jpg es una imagen que laстра el rendimiento del sistema porque no está en nuestro servidor.

Ahora estudiaremos a que se deben los errores:

Abrimos performance.exe y vemos que la tarjeta de red tiene porcentajes muy altos, llegando al 100% en ocasiones, por lo que el problema se localizará en ella.

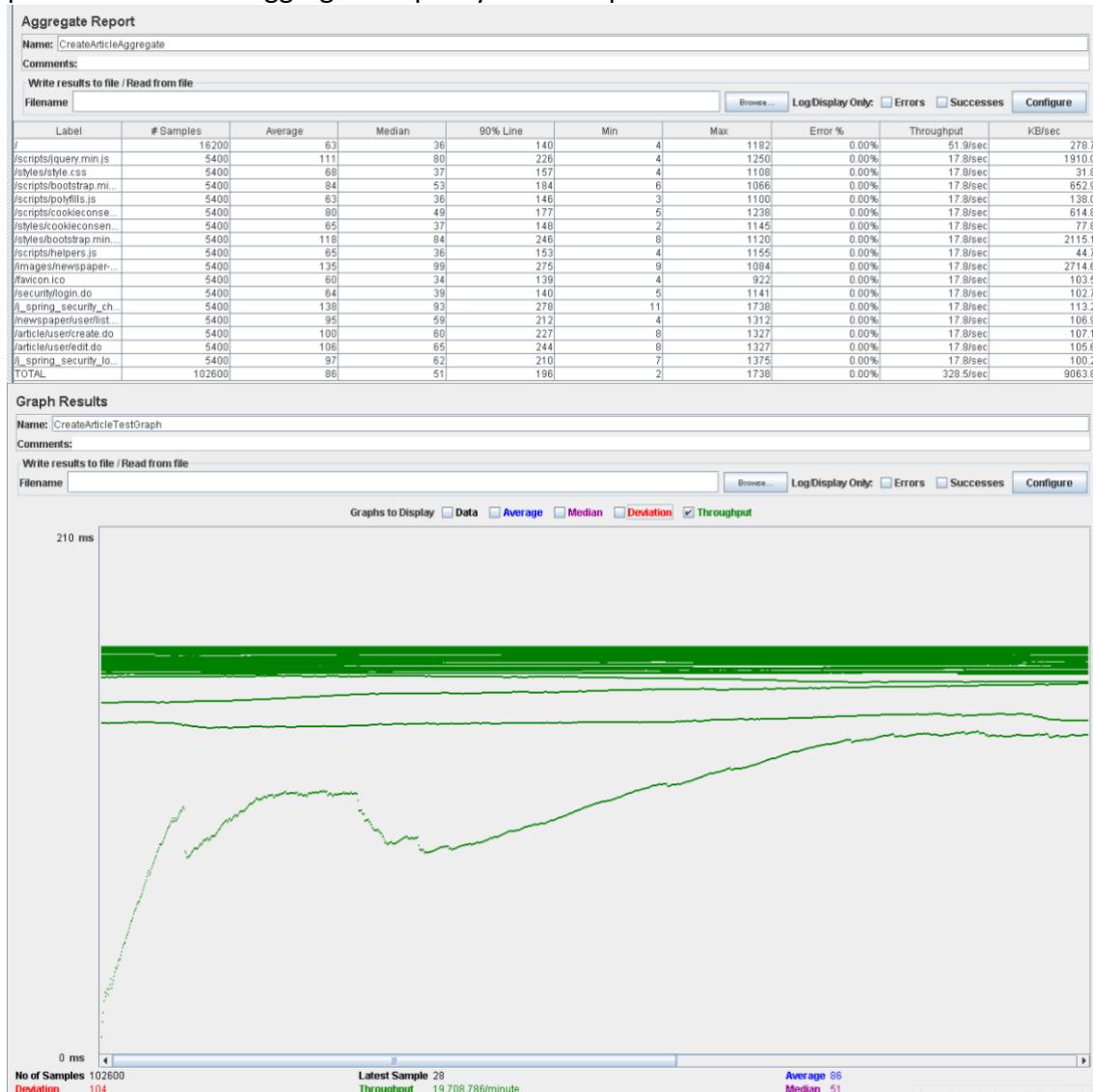


Performance con Packets/sec.

## Crear artículo.

En este caso de uso, accedemos a la página, nos logueamos como un cliente, accedemos a la lista de periódicos, seleccionamos crear artículo, rellenamos el formulario, guardamos y nos deslogueamos.

La configuración en esta ocasión es de 180 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores ni tiempos de espera excesivamente altos, que podemos ver en el Aggregate Report y en el Graph Results.



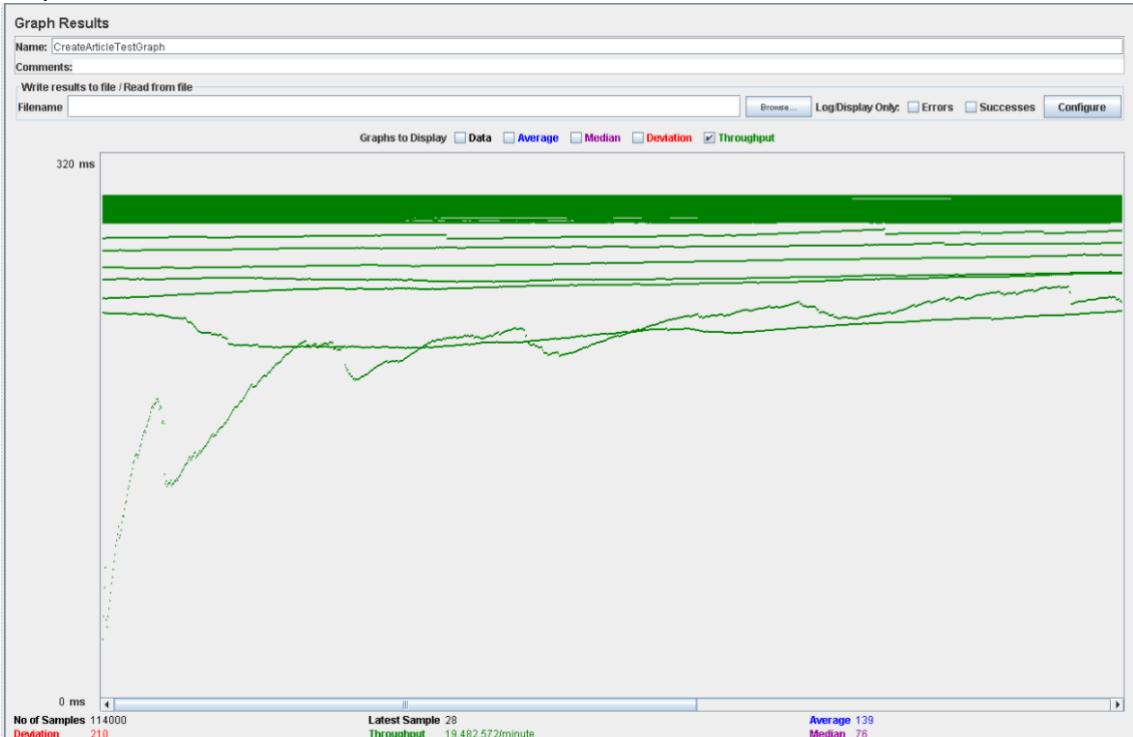
En esta tabla, el rendimiento es de aproximadamente 19.800 ejecuciones por minuto.

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, la configuración es de 200 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose esperas excesivas en algunos casos.

Aggregate Report									
Name: CreateArticleAggregate									
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
/	18000	114	51	241	5	8183	0.00%	51.3/sec	275.5
scripts/jquery.min.js	6000	161	105	346	5	3161	0.00%	17.5/sec	1879.5
styles/style.css	6000	103	50	246	3	3132	0.00%	17.5/sec	31.3
scripts/bootstrap.mi...	6000	123	73	280	5	3179	0.00%	17.5/sec	642.3
scripts/polyfills.js	6000	95	49	230	4	1933	0.00%	17.5/sec	135.8
scripts/cookieconsen...	6000	118	66	268	5	3145	0.00%	17.5/sec	604.8
styles/bootstrap.conse...	6000	98	51	235	4	3108	0.00%	17.5/sec	76.6
scripts/bootstrap.min...	6000	157	110	341	6	3226	0.00%	17.5/sec	2080.1
scripts/helpers.js	6000	102	51	245	3	1739	0.00%	17.5/sec	44.0
images/newspaper-...	6000	175	123	371	4	3274	0.00%	17.5/sec	2669.6
favicon.ico	6000	105	50	257	3	2990	0.00%	17.5/sec	101.8
/security/login.do	6000	112	60	263	4	3001	0.00%	17.5/sec	101.0
j_spring_security_ch...	6000	226	160	475	6	3661	0.00%	17.5/sec	111.3
newspaper/user/list...	6000	180	106	409	6	2955	0.00%	17.5/sec	105.1
/article/user/create.d...	6000	188	107	427	5	3138	0.00%	17.5/sec	105.2
/article/user/edit.d...	6000	187	106	429	8	3243	0.00%	17.5/sec	103.8
j_spring_security_lo...	6000	183	104	351	3	2831	0.00%	17.5/sec	98.5
TOTAL	114000	139	76	319	3	8183	0.00%	324.7/sec	9860.3

Como observamos fijándonos en 90% Line, las direcciones j\_spring\_security\_check, /newspaper/user/list.do, /article/user/create.do y /article/user/edit.do, generan un tiempo de respuesta mayor al resto, 475 ms, 409 ms, 427 ms y 429 ms respectivamente.



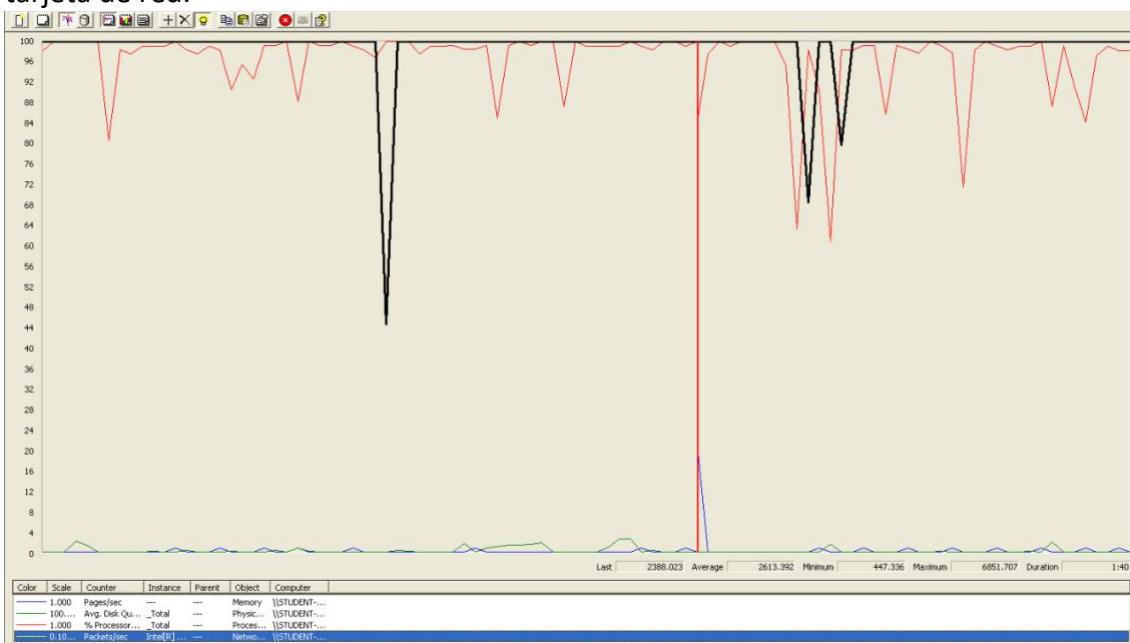
En esta tabla, el rendimiento es de aproximadamente 19.400 ejecuciones por minuto (desciende al aumentar el número de usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Como con 200 usuarios las esperas son excesivas, este caso de uso soporta entre 180 y 200 usuarios.

Ahora estudiaremos a que se deben los tiempos y errores:

Abrimos performance.exe y vemos la tarjeta de red tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que podemos estar ante un problema de la

tarjeta de red.



Performance con Packets/sec.

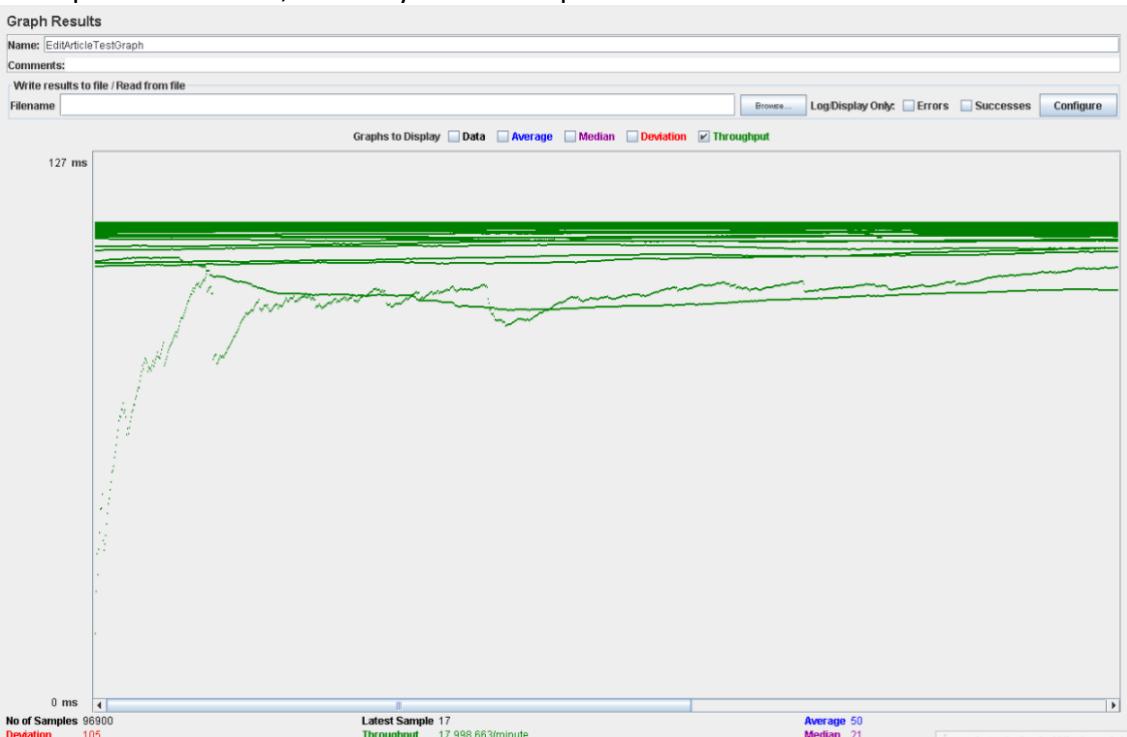
## Editar un artículo.

En este caso de uso, accedemos a la página, nos logueamos como un usuario, accedemos a nuestros artículos, accedemos a la vista para editar, editamos algún parámetro, guardamos y nos deslogueamos.

Hemos establecido el número de hilos máximo acorde con las prestaciones de nuestros ordenadores a 170 usuarios, 1 ramp-up y 30 loop count, para este caso de uso. Con esto, obtenemos las gráficas Aggregate Report y Graph Results:

Aggregate Report											
Name	editArticleAggregate	Comments:									
Write results to file / Read from file											
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec		
/scripts/bootstrap.mi...	15300	19	12	20	2	5067	0.00%	47.4/sec	265.4		
/scripts/bootstrap.js	5100	23	19	42	2	526	0.00%	16.3/sec	593.0		
/scripts/bootstrap.min...	5100	15	10	26	2	500	0.00%	16.3/sec	122.5		
/scripts/cookieconsen...	5100	14	10	26	1	415	0.00%	16.3/sec	67.3		
/scripts/cookieconsen...	5100	13	10	25	1	350	0.00%	16.3/sec	25.5		
/scripts/cookieconsen...	5100	42	33	70	5	608	0.00%	16.3/sec	1742.5		
/scripts/cookieconsen...	5100	43	34	81	5	634	0.00%	16.3/sec	1928.7		
/scripts/cookieconsen...	5100	20	15	36	2	459	0.00%	16.3/sec	558.1		
/scripts/helpers.js	5100	14	10	26	1	645	0.00%	16.3/sec	37.2		
/scripts/helpers.js	5100	64	42	104	6	696	0.00%	16.2/sec	2476.4		
/scripts/login.do	5100	15	12	27	3	578	0.00%	16.3/sec	80.2		
/_spring_security_ch...	5100	119	63	273	7	1409	0.00%	16.3/sec	97.8		
/article/user/list.do	10200	135	82	292	8	1939	0.00%	32.1/sec	221.5		
/article/user/edit.do	10200	112	56	262	6	1985	0.00%	32.4/sec	322.9		
/_spring_security_lo...	5100	28	24	51	5	261	0.00%	16.2/sec	87.8		
TOTAL	96900	50	21	106	1	5067	0.00%	300.0/sec	8400.9		

Como podemos observar fijándonos en 90% Line, las direcciones /article/list.do y /article/user/edit.do generan un tiempo de respuesta mayor al resto pero sin sobrepasar los límites, 292 ms y 262 ms respectivamente.

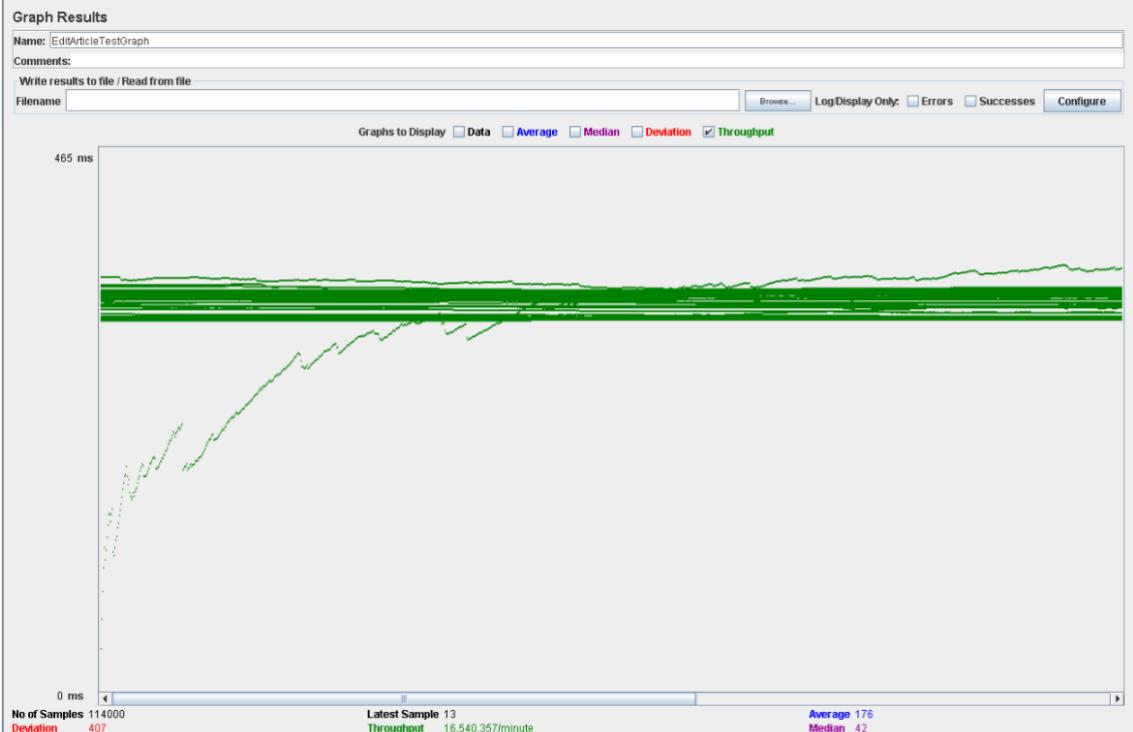


En esta tabla, el rendimiento es de aproximadamente 18.000 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Ahora, la configuración es de 200 usuarios, 1 ramp-up y 30 acciones cada uno, obteniéndose esperas demasiado altas.

Aggregate Report											
Write results to file / Read from file							Browse...	Log/Display Only	Errors	Successes	Configure
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec		
/	18000	38	21	71	3	1205	0.00%	43.5/sec	234.8		
/scripts/bootstrap.mi...	6000	48	31	94	3	870	0.00%	14.8/sec	541.4		
/scripts/polyfills.js	6000	37	19	72	2	855	0.00%	14.8/sec	111.8		
/styles/cookieconsen...	6000	34	19	72	1	1159	0.00%	14.8/sec	61.5		
/styles/style.css	6000	34	18	71	1	1081	0.00%	14.8/sec	23.3		
/scripts/jquery.min.js	6000	77	56	148	5	1227	0.00%	14.8/sec	1590.7		
/styles/bootstrap.min...	6000	82	59	160	6	1211	0.00%	14.8/sec	1760.5		
/scripts/cookieconsen...	6000	49	26	101	3	1179	0.00%	14.8/sec	509.5		
/scripts/helpers.js	6000	37	19	75	2	808	0.00%	14.8/sec	34.0		
/images/newspaper-...	6000	102	74	200	7	1236	0.00%	14.8/sec	2260.8		
/security/login.do	6000	41	21	81	3	1008	0.00%	14.8/sec	82.4		
/j_spring_security_ch...	6000	509	241	1288	7	8667	0.00%	14.8/sec	89.3		
/article/user/list.do	12000	547	281	1348	9	8748	0.00%	29.4/sec	202.9		
/article/user/edit.do	12000	497	225	1278	6	8414	0.00%	29.6/sec	304.2		
/j_spring_security_lo...	6000	85	46	188	6	1139	0.00%	14.8/sec	80.3		
TOTAL	114000	176	42	455	1	8748	0.00%	275.7/sec	7721.2		

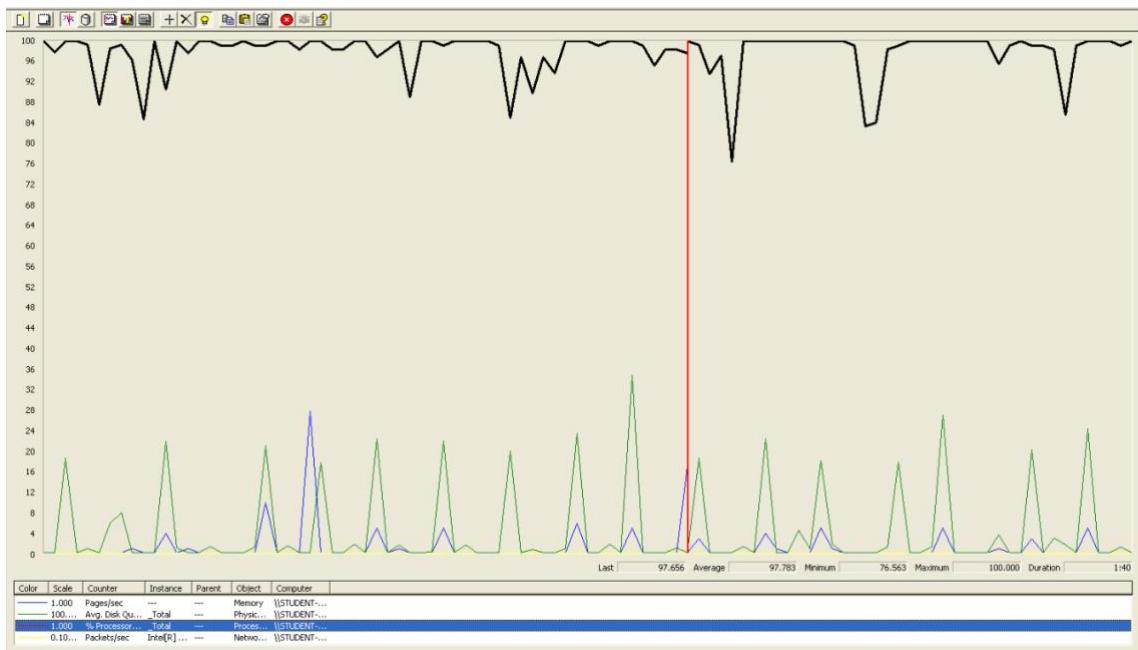
Como observamos fijándonos en 90% Line, las direcciones `j_spring_security_check`, `/article/user/list.do` y `/article/user/edit.do`, generan un tiempo de respuesta mayor al resto, 1288 ms, 1348 ms y 1278 ms respectivamente.



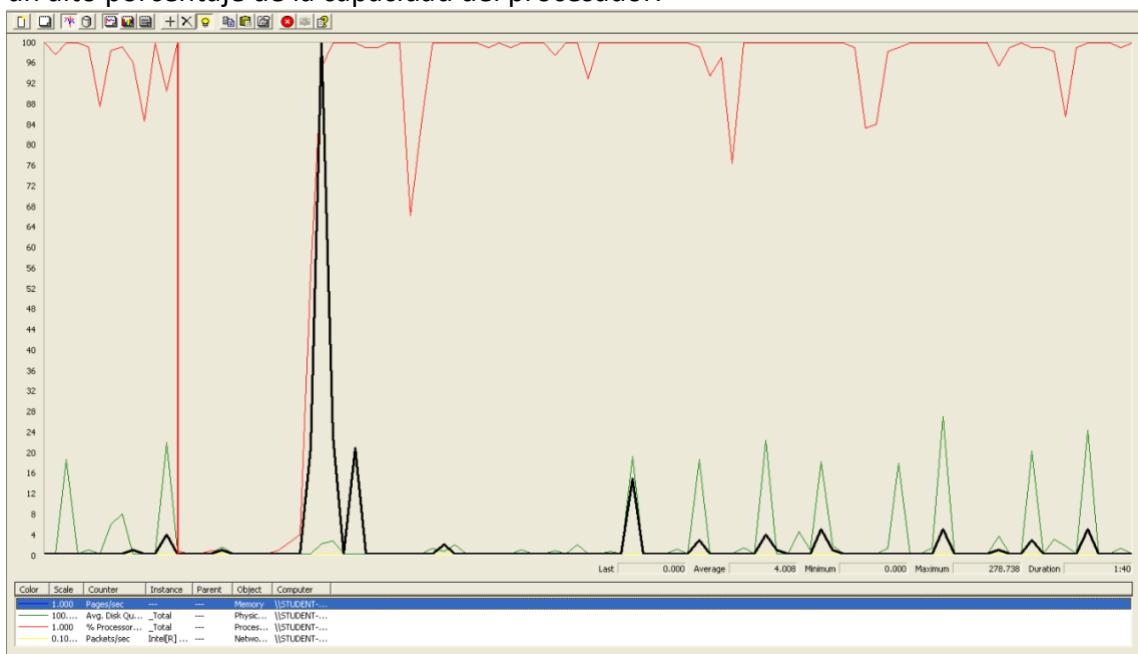
En esta tabla, el rendimiento es de aproximadamente 16.500 ejecuciones por minuto (desciende al aumentar los usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Como con 200 usuarios las esperas son excesivas, este caso de uso soporta entre 170 y 200 usuarios.

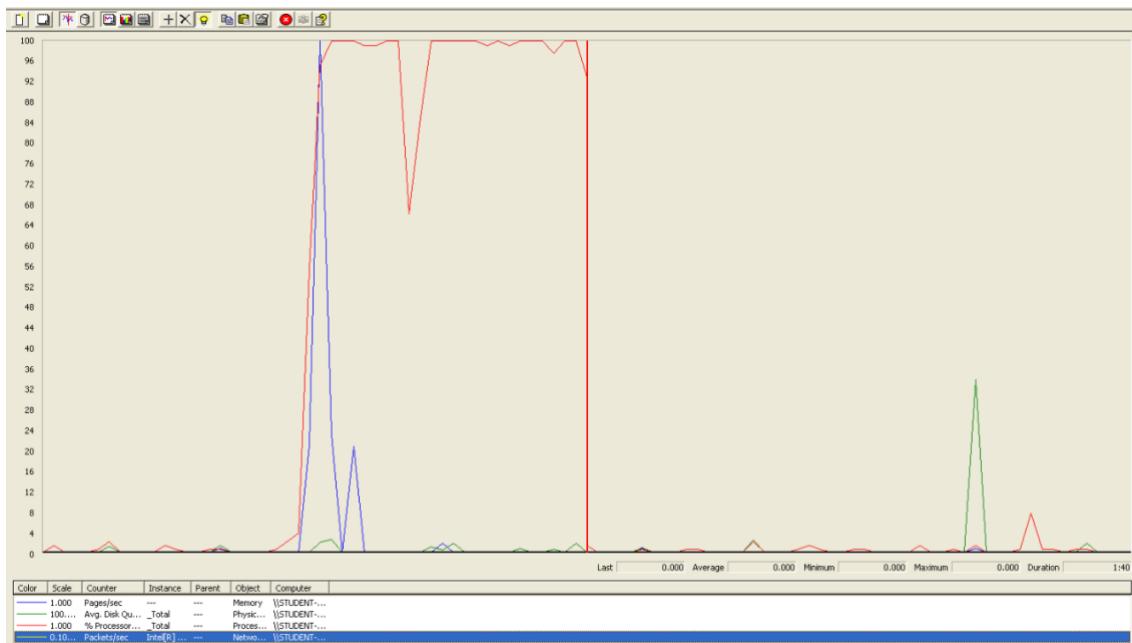
Para encontrar origen a estos resultados, revisamos performance.exe, obteniendo lo siguiente:



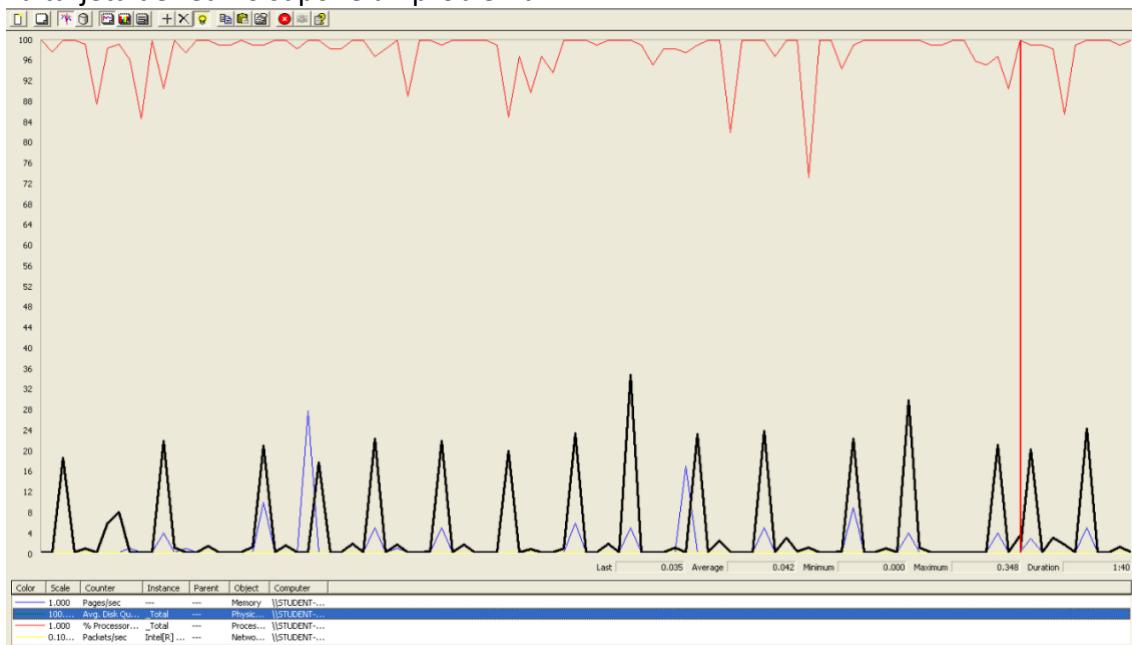
Observamos que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador.



No hay fallos de memoria ya que no destaca altos porcentajes en este caso.



La tarjeta de red no supone un problema.

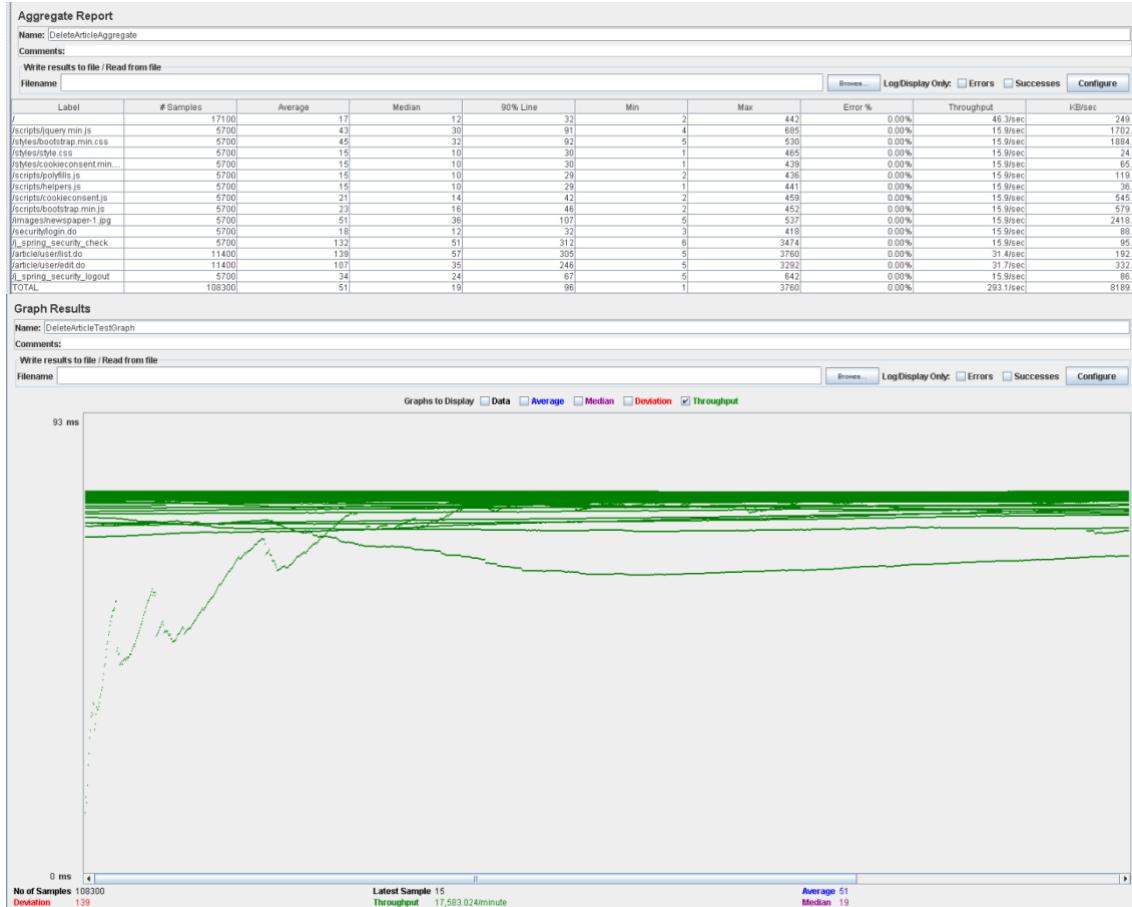


El disco no es el problema, puesto que los valores rondan el 15-20% de media.

## Borrar un artículo.

En este caso de uso, accedemos a la página, nos logueamos como un usuario, accedemos a nuestros artículos, accedemos a la vista para editar, borramos el artículo y nos deslogueamos.

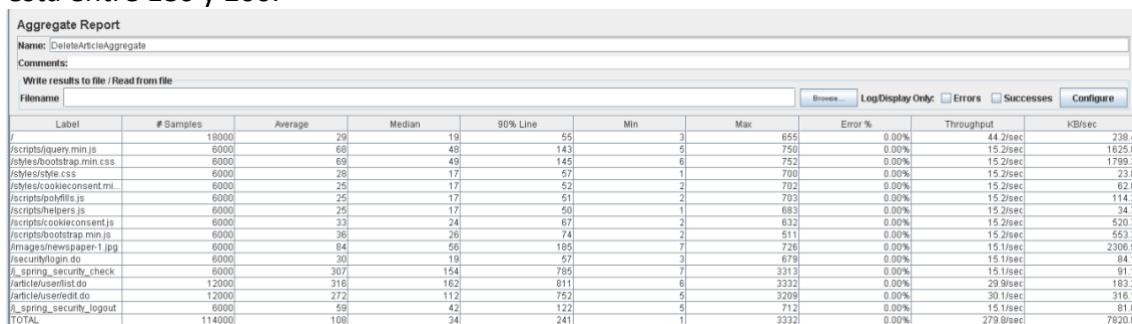
La configuración en esta ocasión es de 190 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores ni tiempos de respuestas muy altos. Obtenemos las gráficas:



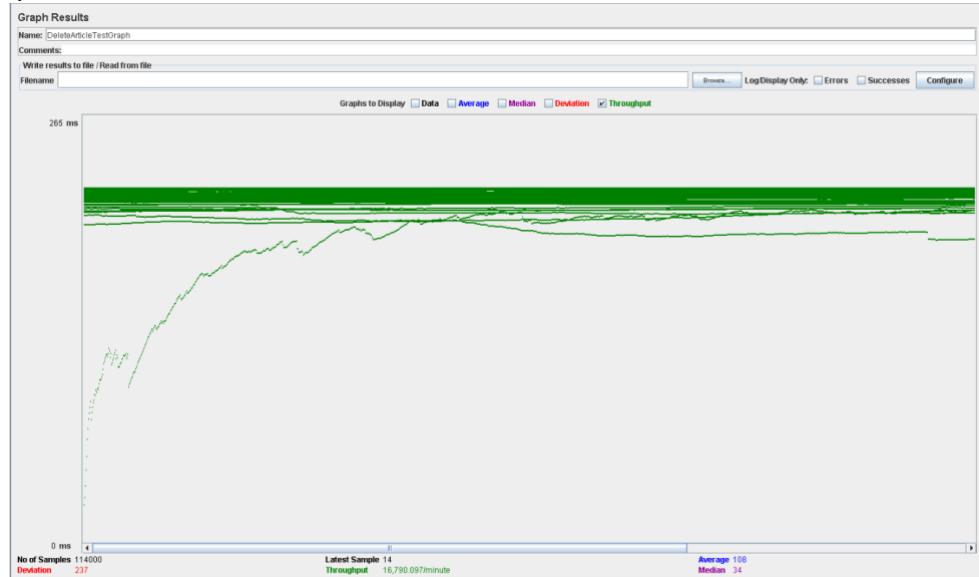
En esta tabla, el rendimiento es de aproximadamente 17.600 ejecuciones por minuto.

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, la configuración es de 200 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose pequeños porcentajes de errores. El número de usuarios permitidos está entre 180 y 200.



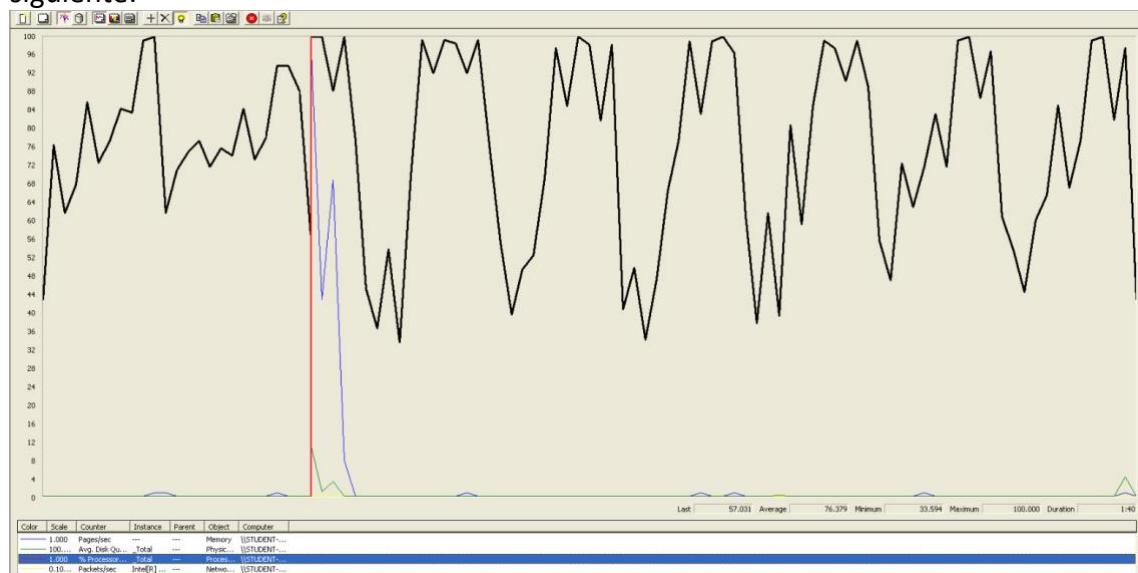
Como observamos fijándonos en 90% Line, las direcciones /article/user/list.do y /article/user/edit.do generan un tiempo de respuesta mayor al resto, 2170 ms y 2067 ms respectivamente.



En esta tabla, el rendimiento es de aproximadamente 16.800 ejecuciones por minuto (desciende en gran medida al aumentar los 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.

Como con 200 usuarios las esperas son excesivas, este caso de uso soporta entre 190 y 200 usuarios.

Para encontrar origen a estos resultados, revisamos performance.exe, obteniendo lo siguiente:

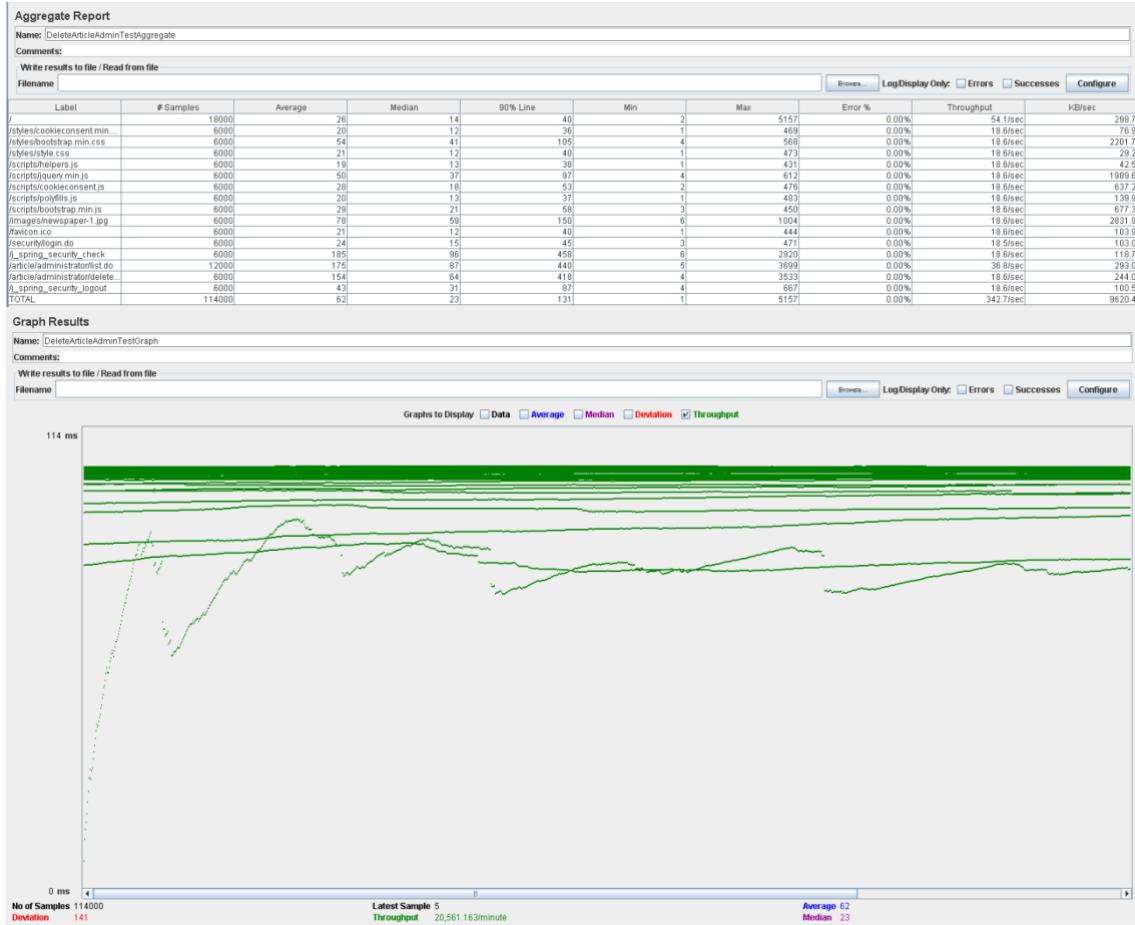


Observamos que podemos estar ante un problema de CPU, ya que se está utilizando un alto porcentaje de la capacidad del procesador la mayoría del tiempo.

## Borrar artículos administrador.

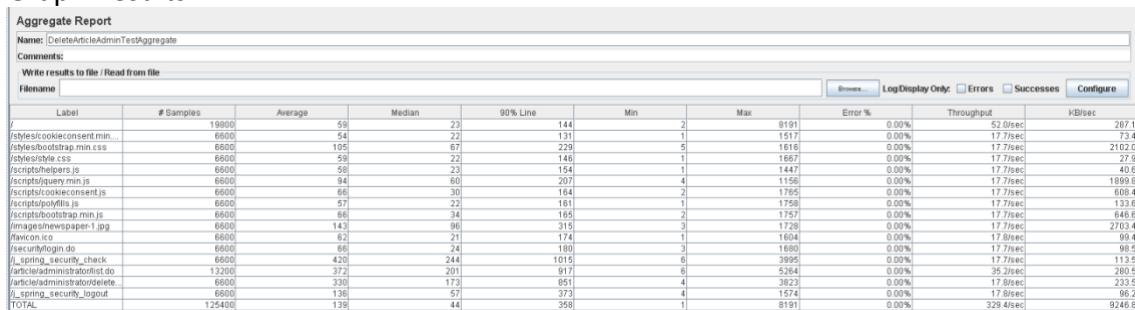
En este caso de uso, accedemos a la página, nos logueamos como un administrador, accedemos a la lista de artículos, borramos el artículo que deseemos y nos deslogueamos.

La configuración en esta ocasión es de 200 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores ni tiempos de respuestas demasiado altos. Obtenemos las gráficas:

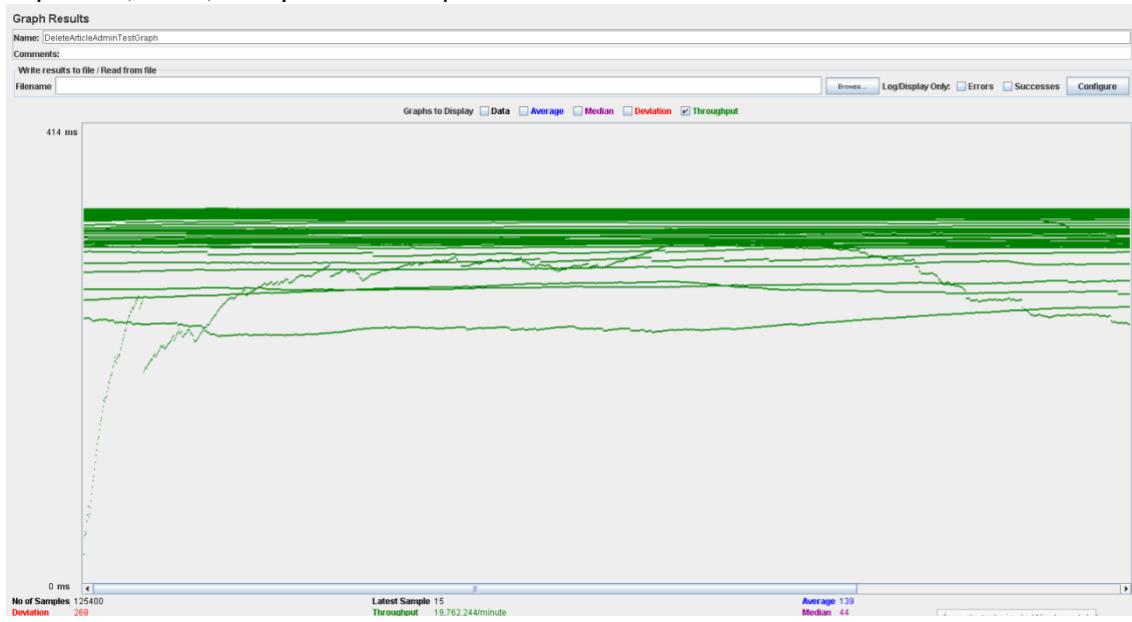


En esta tabla, el rendimiento es de aproximadamente 20.500 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

La configuración en esta ocasión es de 220 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose leves errores, que podemos ver en el Aggregate Report y en el Graph Results:



Como podemos observar fijándonos en 90% Line, las direcciones /j\_spring\_security\_check, /article/administrator/list.do y /article/administrator/deleteList.do generan un tiempo de respuesta mayor a lo esperado, 1015, 917 y 851 ms respectivamente.



En esta tabla, el rendimiento es de aproximadamente 19.700 ejecuciones por minuto (desciende al aumentar los 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 los errores:

Abrimos performance.exe y vemos que la CPU tienen porcentajes muy altos, llegando al 100% en la mayoría de las ocasiones, por lo que el problema se localizará en la CPU.



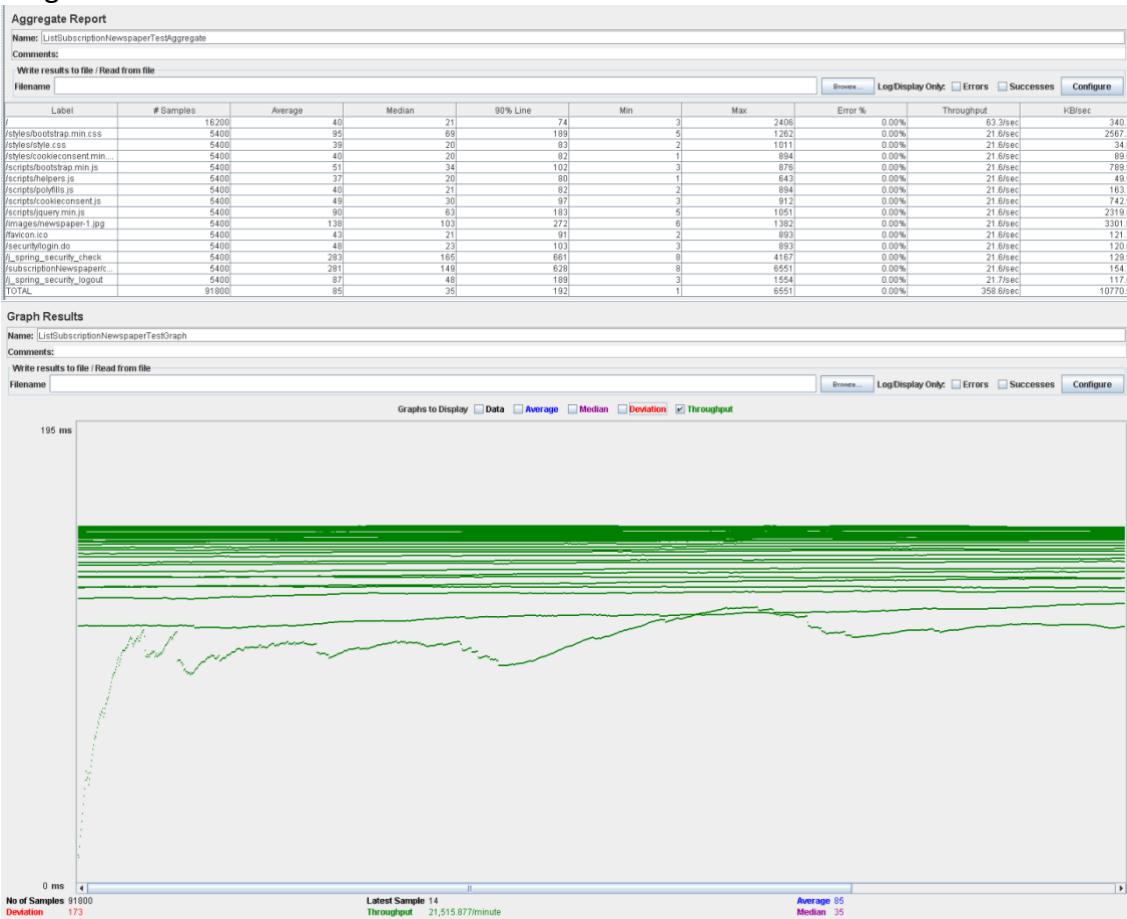
Performance con Processor Time.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 200 y 220.

## Listar suscripciones a periódicos de un cliente.

En este caso de uso, accedemos a la página, nos logueamos como un administrador, accedemos a la lista de artículos, borramos el artículo que deseemos y nos deslogueamos.

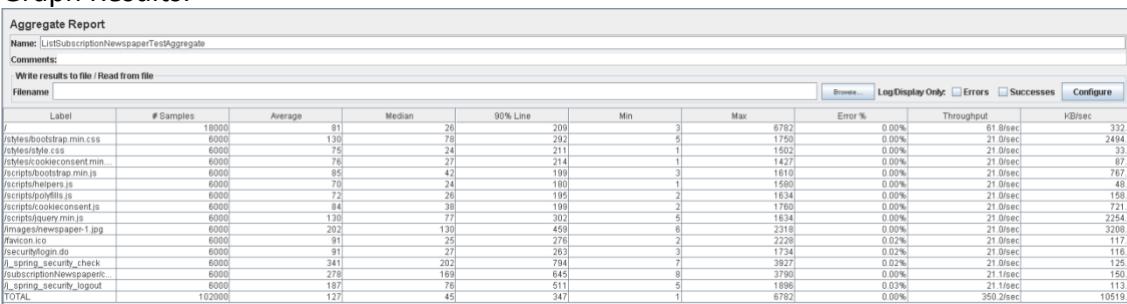
La configuración en esta ocasión es de 180 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores ni tiempos de respuestas demasiado altos. Obtenemos las gráficas:



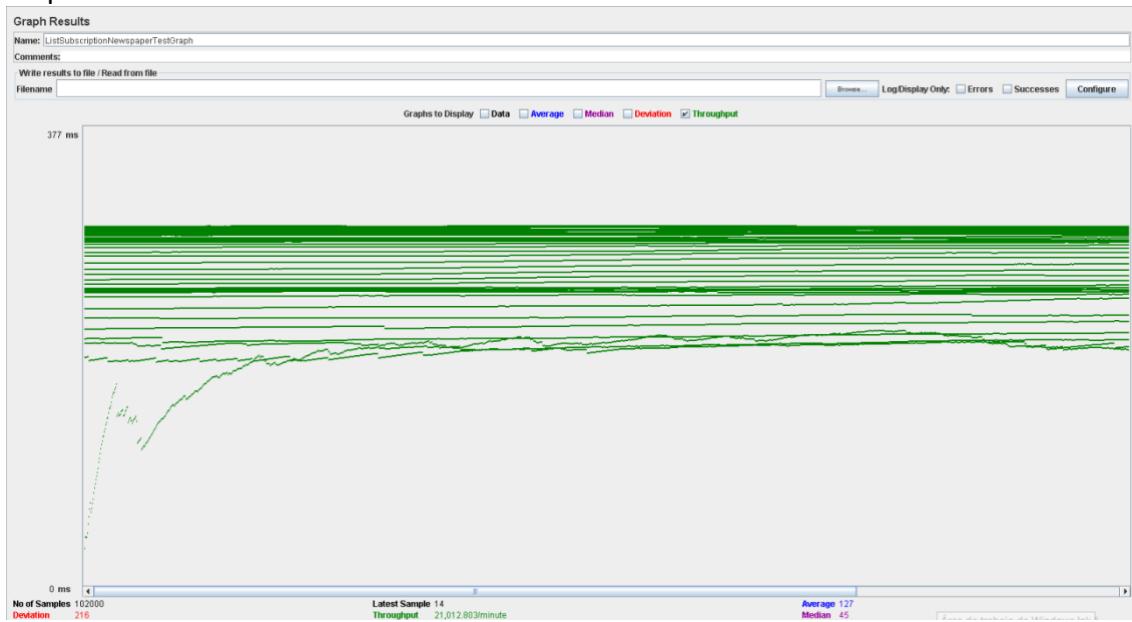
En esta tabla, el rendimiento es de aproximadamente 21.500 ejecuciones por minuto.

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.

La configuración en esta ocasión es de 200 usuarios, 1 ramp-up y 30 acciones cada uno, produciéndose leves errores, que podemos ver en el Aggregate Report y en el Graph Results:



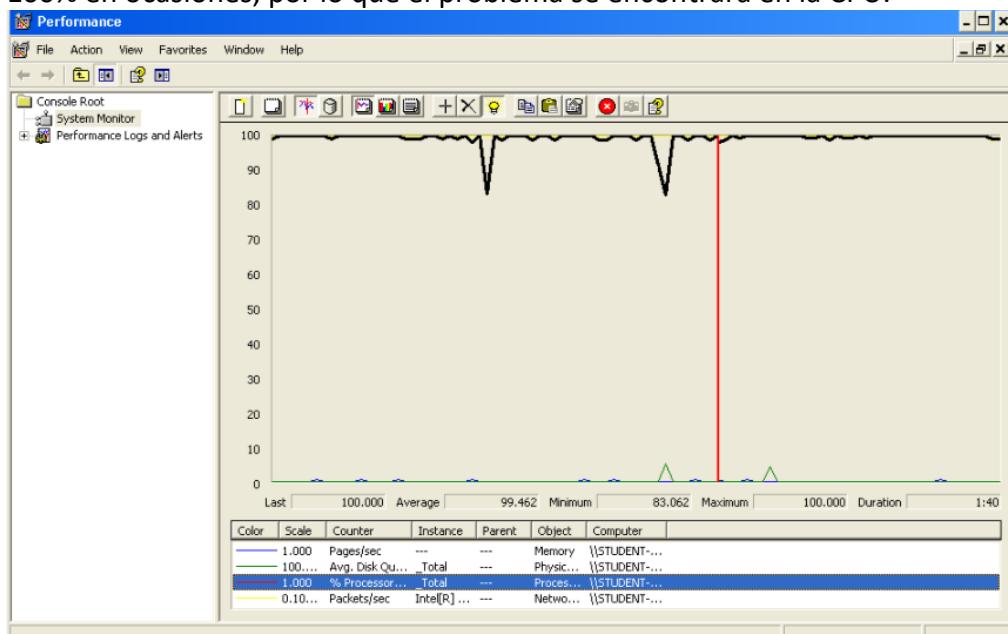
Como podemos observar fijándonos en 90% Line, las direcciones /j\_spring\_security\_check y /subscriptionNewspaper/customer/list.do generan un tiempo de respuesta mayor a lo esperado para un caso de uso de listar, 794 y 645 ms respectivamente.



En esta tabla, el rendimiento es de aproximadamente 21.000 ejecuciones por minuto (disminuye al aumentar los 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 los errores:

Abrimos performance.exe y vemos que la CPU tiene porcentajes muy altos, llegando al 100% en ocasiones, por lo que el problema se encontrará en la CPU.



Performance con Processor Time.

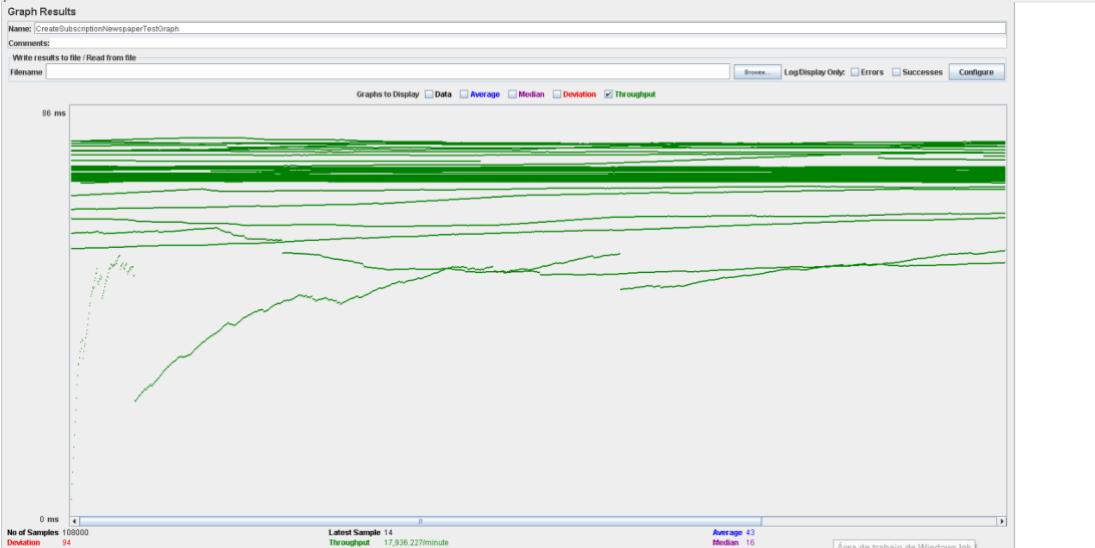
Por lo que, en este caso de uso, los usuarios permitidos serán entre 180 y 200.

## Suscribirse a un periódico.

En este caso de uso, accedemos a la página, nos logueamos como cliente, accedemos al listado de newspapers para suscribirnos, a continuación, al formulario para suscribirse y finalmente accedemos nos deslogueamos.

Probamos una configuración 180 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores y las esperas no son excesivamente altas:

Aggregate Report										
Name:	CreateSubscriptionNewspaperTestAggregate									
Comments:										
Filename:	Write results to file / Read from file									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	16200	14	10	24	2	973	0.00%	44.8/sec	241.1	
/styles/style.css	5400	12	8	22	1	820	0.00%	15.4/sec	24.2	
/styles/cookieconsent.min.css	5400	11	9	22	1	619	0.00%	15.4/sec	63.0	
/bootstrapstrap.min.css	5400	20	27	70	4	789	0.00%	15.4/sec	139.0	
/scripts/jquery.min.js	5400	34	25	68	4	696	0.00%	15.4/sec	165.2	
/scripts/helpers.js	5400	11	8	23	1	313	0.00%	15.4/sec	35.3	
/scripts/cookieconsent.js	5400	16	12	32	2	404	0.00%	15.4/sec	52.9	
/scripts/bootstrapstrap.min.js	5400	18	13	34	2	319	0.00%	15.4/sec	66.4	
/scripts/fontawesome.min.js	5400	11	8	21	1	444	0.00%	15.4/sec	116.2	
/images/newspaper-1.jpg	5400	53	40	101	4	881	0.00%	15.4/sec	2351.7	
/favicon.ico	5400	11	8	20	1	342	0.00%	15.4/sec	89.2	
/scripts/security_check	5400	65	43	231	6	2118	0.00%	15.4/sec	89.5	
/newspaperCustomerInfoForSubscribe.do	5400	98	49	228	6	1832	0.00%	15.4/sec	89.2	
/subscriptionNewspaperCustomerCreate.do	5400	136	81	295	7	2494	0.00%	15.4/sec	124.9	
/subscriptionNewspaperCustomerEdit.do	5400	146	92	318	8	2306	0.00%	15.4/sec	125.9	
/subscriptionNewspaperCustomerList.do	5400	98	82	228	6	2697	0.00%	15.4/sec	118.1	
/_spring_security_logout	5400	25	20	46	4	404	0.00%	15.4/sec	83.5	
TOTAL	108000	43	16	99	1	2897	0.00%	288.9/sec	7988.8	



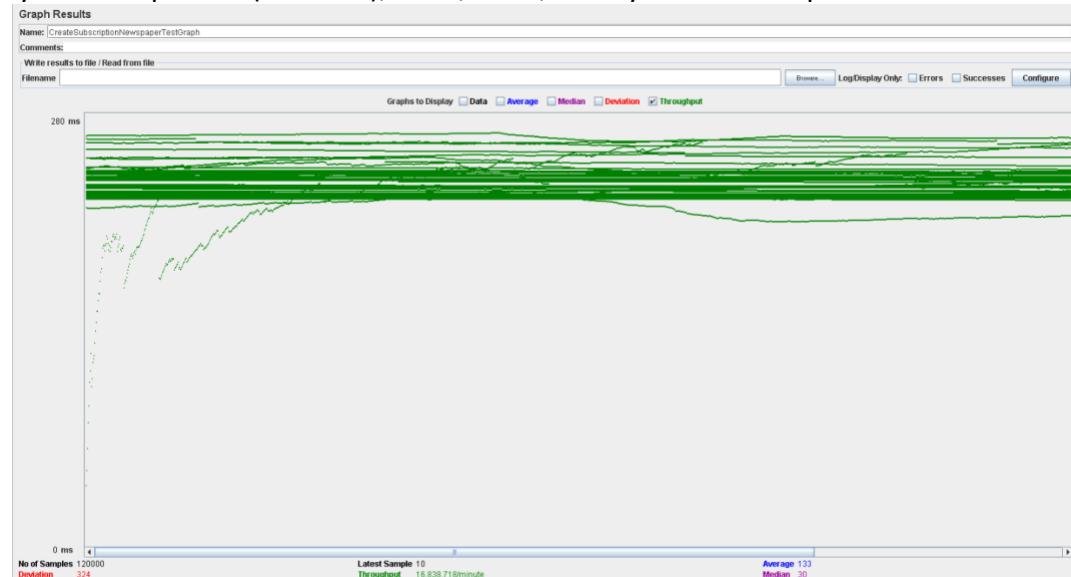
En esta tabla, el rendimiento es de aproximadamente 17.900 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Volvemos a probar con una configuración 200 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero esperas demasiado altas, que podemos ver en el Aggregate Report y en el Graph Results:

Aggregate Report										
Name:	CreateSubscriptionNewspaperTestAggregate									
Comments:										
Filename:	Write results to file / Read from file									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec	
/	18000	24	17	42	2	892	0.00%	12.7/sec	226.4	
/styles/style.css	6000	21	14	39	1	769	0.00%	14.4/sec	22.7	
/styles/cookieconsent.min.css	6000	21	15	38	1	786	0.00%	14.4/sec	59.7	
/bootstrapstrap.min.css	6000	59	47	111	5	833	0.00%	14.4/sec	1710.3	
/scripts/jquery.min.js	6000	56	42	107	5	876	0.00%	14.4/sec	1545.4	
/scripts/helpers.js	6000	23	15	44	1	600	0.00%	14.4/sec	22.0	
/scripts/cookieconsent.js	6000	30	22	55	3	525	0.00%	14.4/sec	646.0	
/scripts/bootstrapstrap.min.js	6000	33	23	61	3	597	0.00%	14.4/sec	526.0	
/scripts/polyfills.js	6000	22	15	41	2	736	0.00%	14.4/sec	108.7	
/images/newspaper-1.jpg	6000	84	66	154	8	885	0.00%	14.4/sec	2199.4	
/favicon.ico	6000	24	16	44	2	395	0.00%	14.4/sec	60.7	
/scripts/login.do	6000	26	18	49	3	507	0.00%	14.4/sec	60.0	
/_spring_security_check	6000	391	172	991	7	9075	0.00%	14.4/sec	86.1	
/newspaperCustomerInfoForSubscribe.do	6000	398	196	1047	7	5847	0.00%	14.4/sec	83.5	
/subscriptionNewspaperCustomerCreate.do	6000	404	286	1129	9	7547	0.00%	14.4/sec	116.0	
/subscriptionNewspaperCustomerEdit.do	6000	403	289	1146	11	7634	0.00%	14.4/sec	117.4	
/subscriptionNewspaperCustomerList.do	6000	395	191	1007	8	8810	0.00%	14.4/sec	110.1	
/_spring_security_logout	6000	50	37	96	5	738	0.00%	14.4/sec	77.9	
TOTAL	120000	133	30	318	1	9075	0.00%	280.6/sec	7481.7	

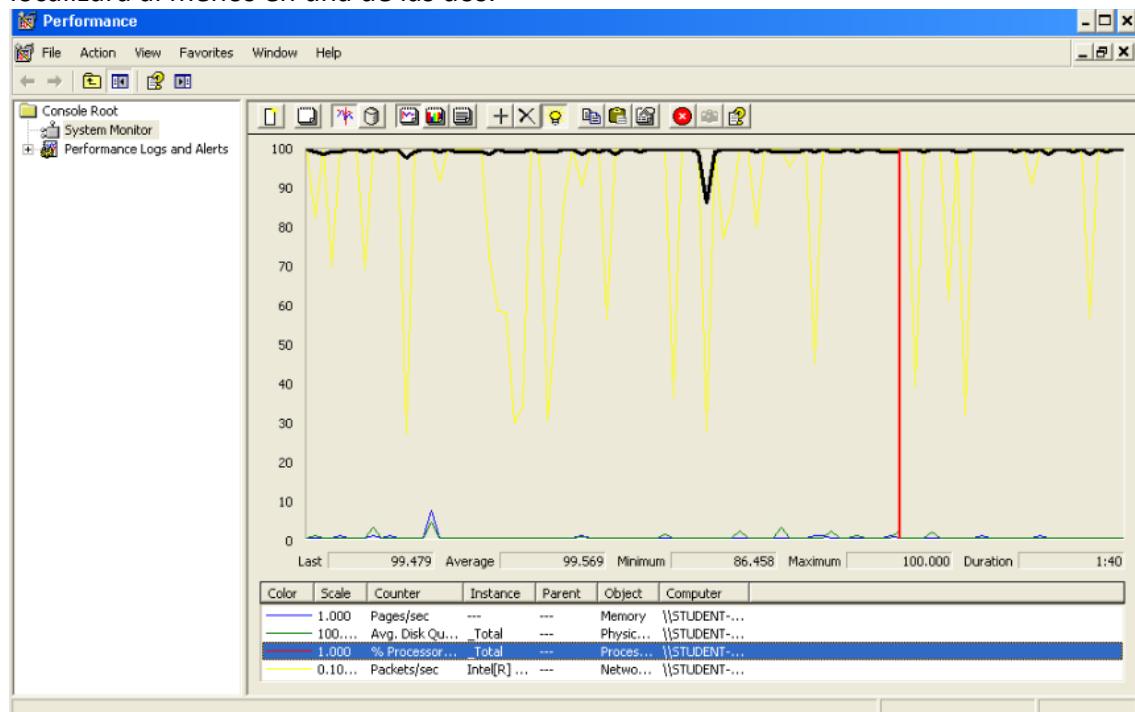
Como podemos observar fijándonos en 90% Line, las direcciones /newspaper/customer/listForSubscribe.do,

/subscriptionNewspaper/customer/create.do,  
 /subscriptionNewspaper/customer/edit.do y  
 /subscriptionNewspaper/customer/list.do generan un tiempo de respuesta mucho mayor a lo esperado (1000 ms), 1047, 1129, 1146 y 1007 ms respectivamente.

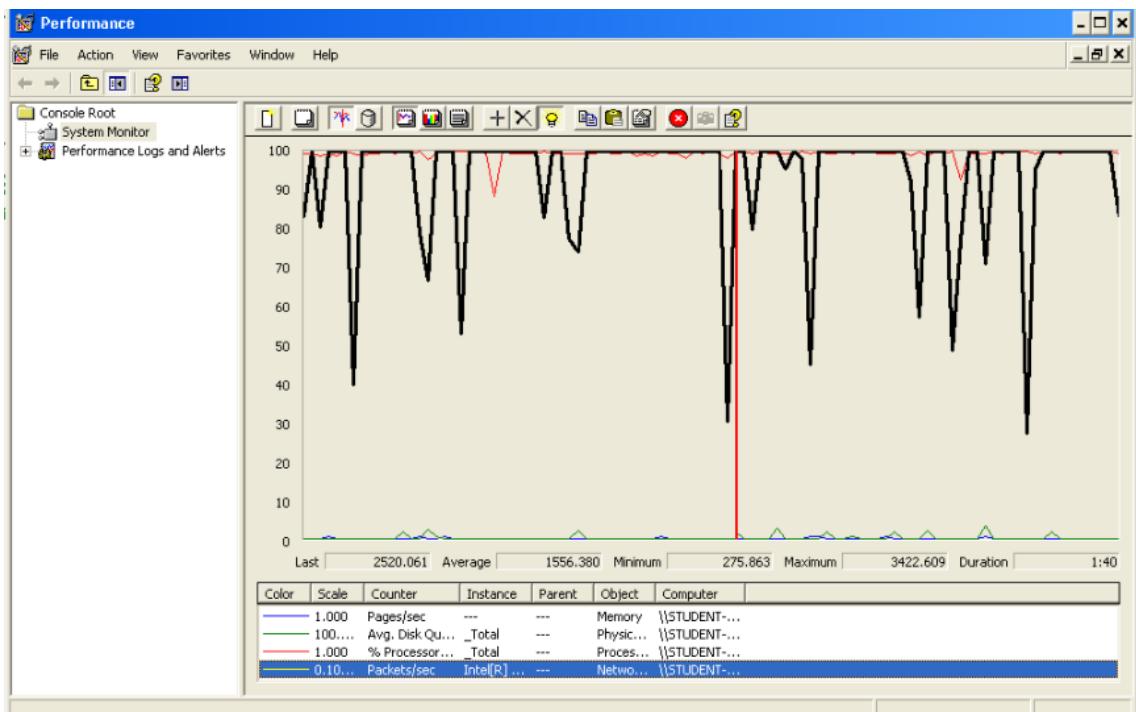


En esta tabla, el rendimiento es de aproximadamente 16.800 ejecuciones por minuto (desciende al aumentar los usuarios). El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché. Ahora estudiaremos a qué se deben los tiempos de espera:

Abrimos performance.exe y vemos que tanto la tarjeta de red como la CPU tienen porcentajes muy altos, llegando al 100% en ocasiones, por lo que el problema se localizará al menos en una de las dos.



Performance con Processor Time. Casi en todo momento la CPU se encuentra al 100%.

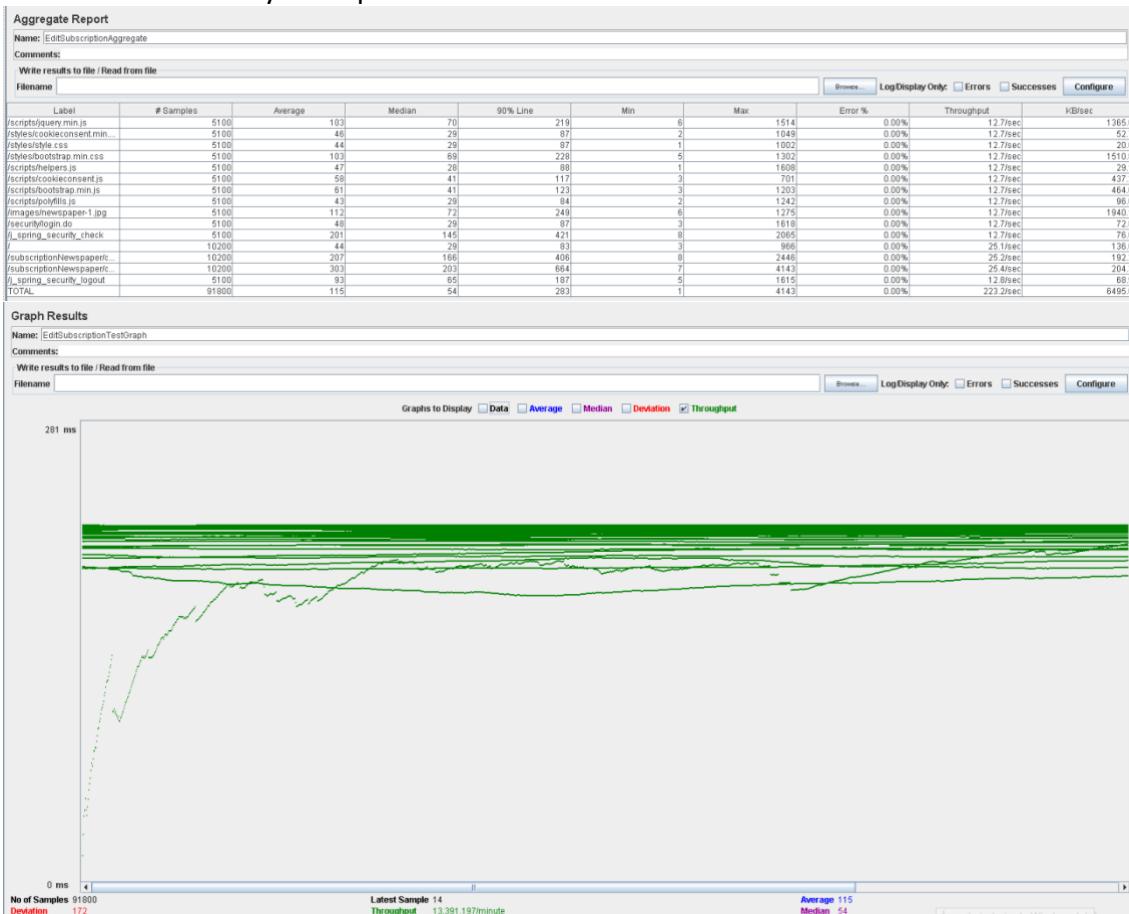


Performance con Packets/sec. Con alto porcentaje de la tarjeta de red.  
Por lo que, en este caso de uso, los usuarios permitidos serán entre 180 y 200.

## Editar suscripción.

En este caso de uso, accedemos a la página, nos logueamos como un cliente, accedemos a nuestras suscripciones, accedemos a la vista para editar, editamos algún parámetro, guardamos y nos deslogueamos.

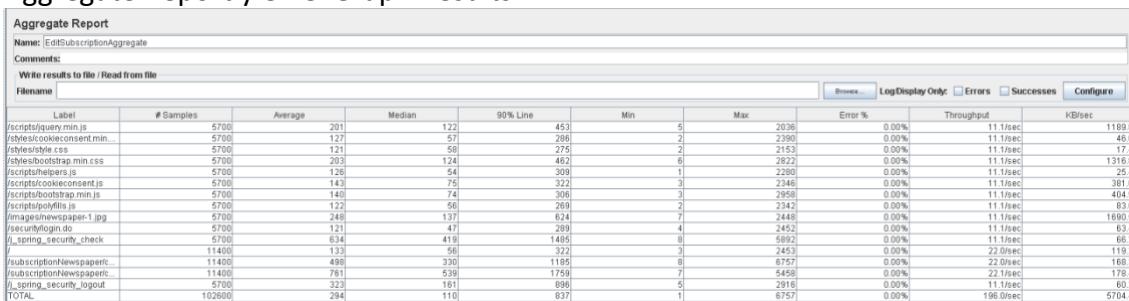
Probamos una configuración 170 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores y las esperas no son excesivamente altas:



En esta tabla, el rendimiento es de aproximadamente 13.400 ejecuciones por minuto.

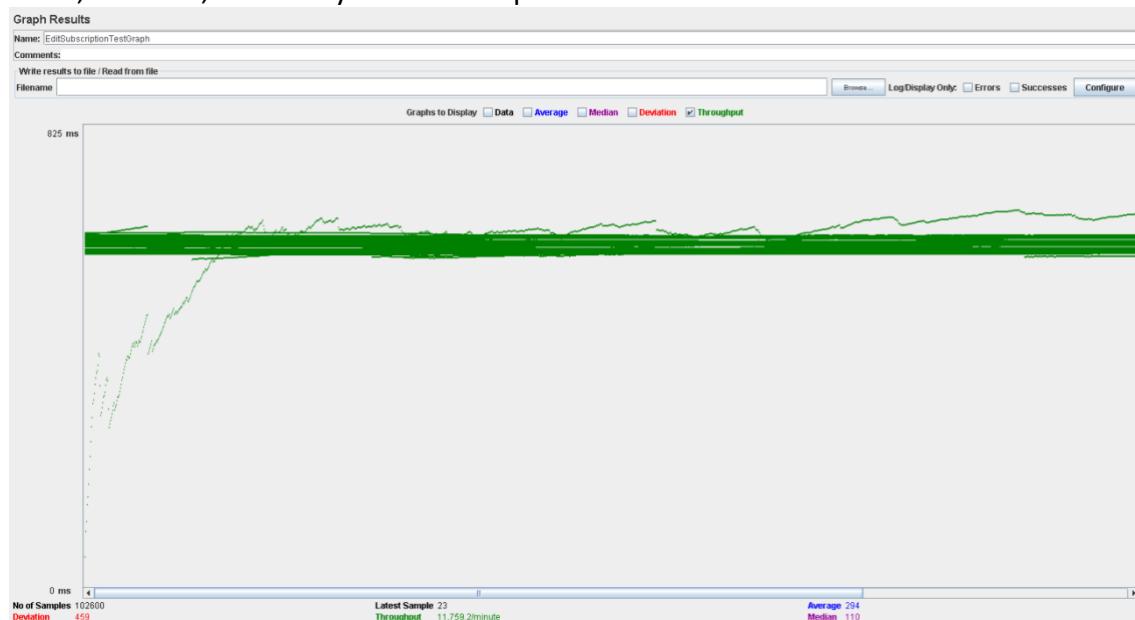
El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Volvemos a probar con una configuración 190 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero esperas demasiado altas, que podemos ver en el Aggregate Report y en el Graph Results:



Como podemos observar fijándonos en 90% Line, las direcciones /j\_spring\_security\_check, /subscriptionNewspaper/customer/list.do y

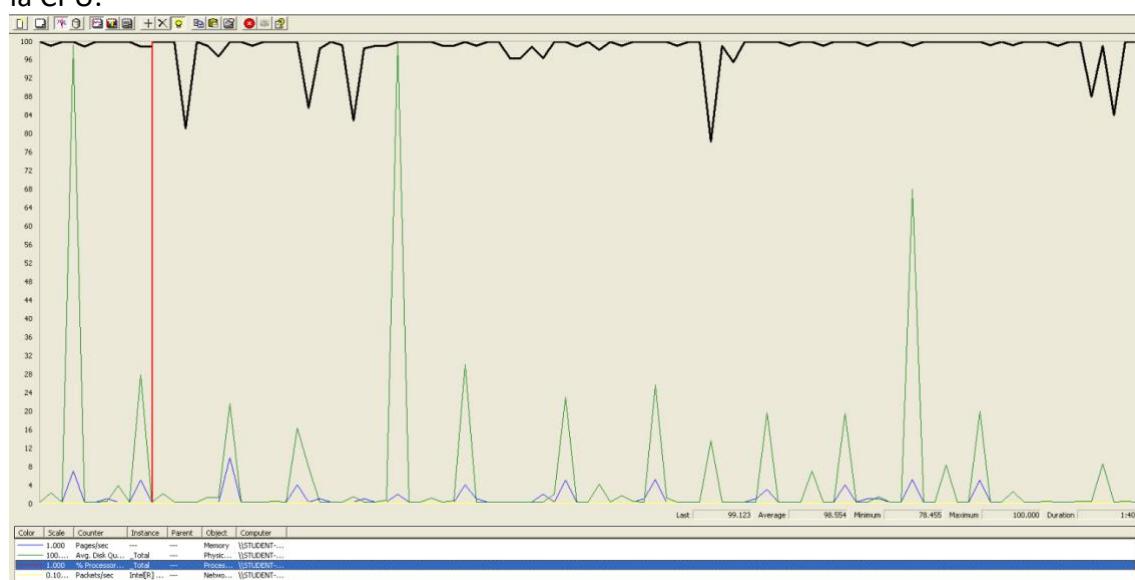
/subscriptionNewspaper/customer/edit.do generan un tiempo de respuesta mayor al resto, 1485 ms, 1185 ms y 1759 ms respectivamente.



En esta tabla, el rendimiento es de aproximadamente 11.700 ejecuciones por minuto (desciende al aumentar los 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 de espera:

Abrimos performance.exe y vemos que el procesador tiene porcentajes muy altos, llegando al 100% casi siempre, por lo que el problema se localizará probablemente en la CPU.

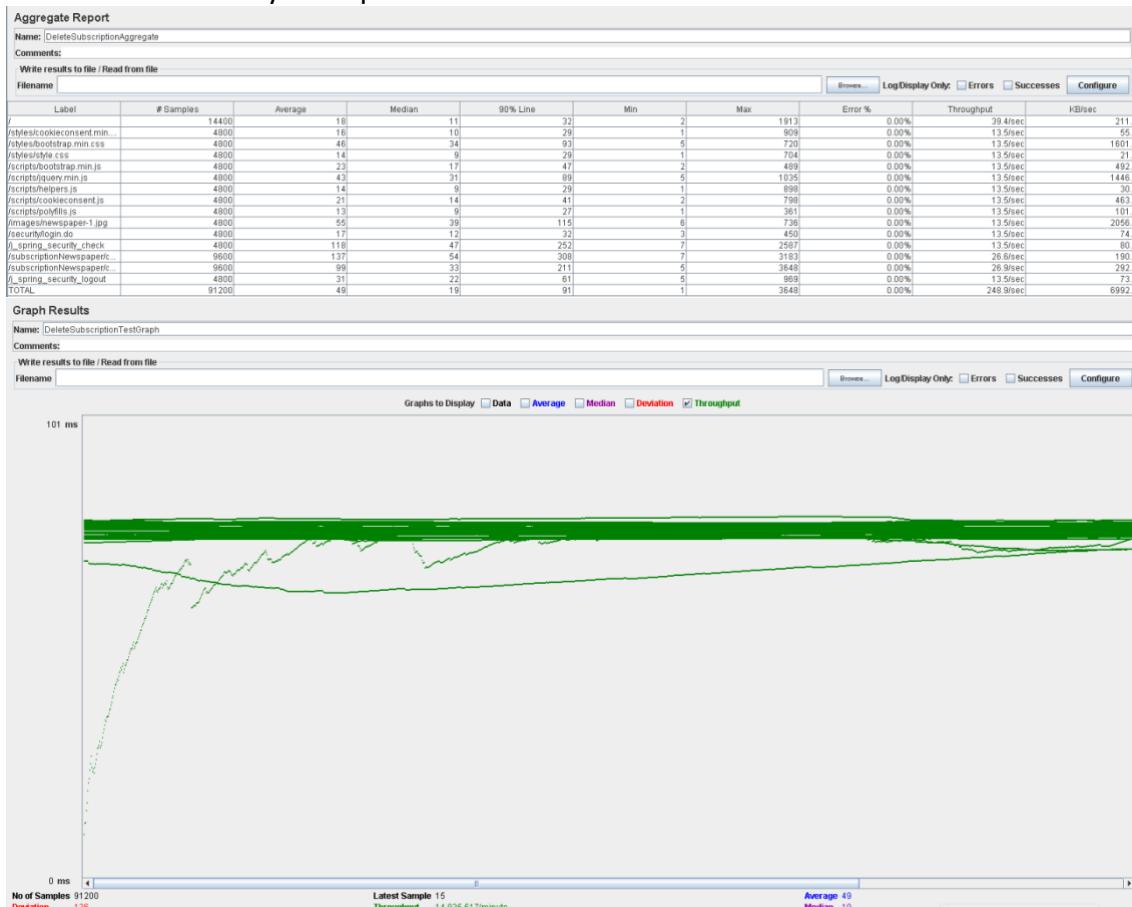


Performance con Processor Time. Casi en todo momento la CPU se encuentra al 100%. Por lo que, en este caso de uso, los usuarios permitidos serán entre 170 y 190.

## Borrar una suscripción.

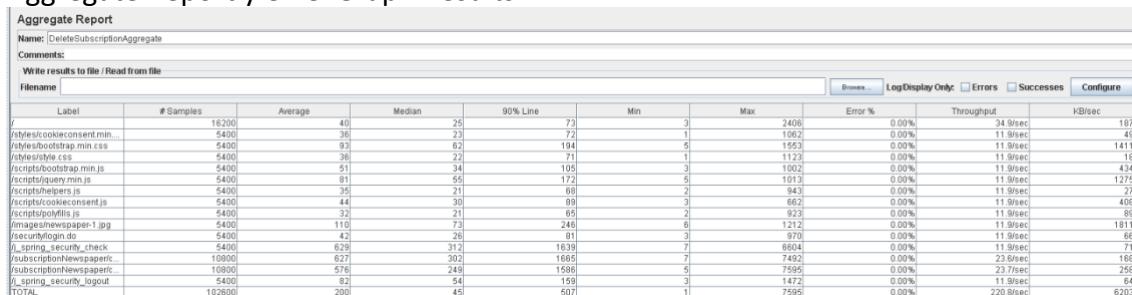
En este caso de uso, accedemos a la página, nos logueamos como un usuario, accedemos a nuestros artículos, accedemos a la vista para editar, borramos el artículo y nos deslogueamos.

Probamos una configuración 160 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores y las esperas no son excesivamente altas:



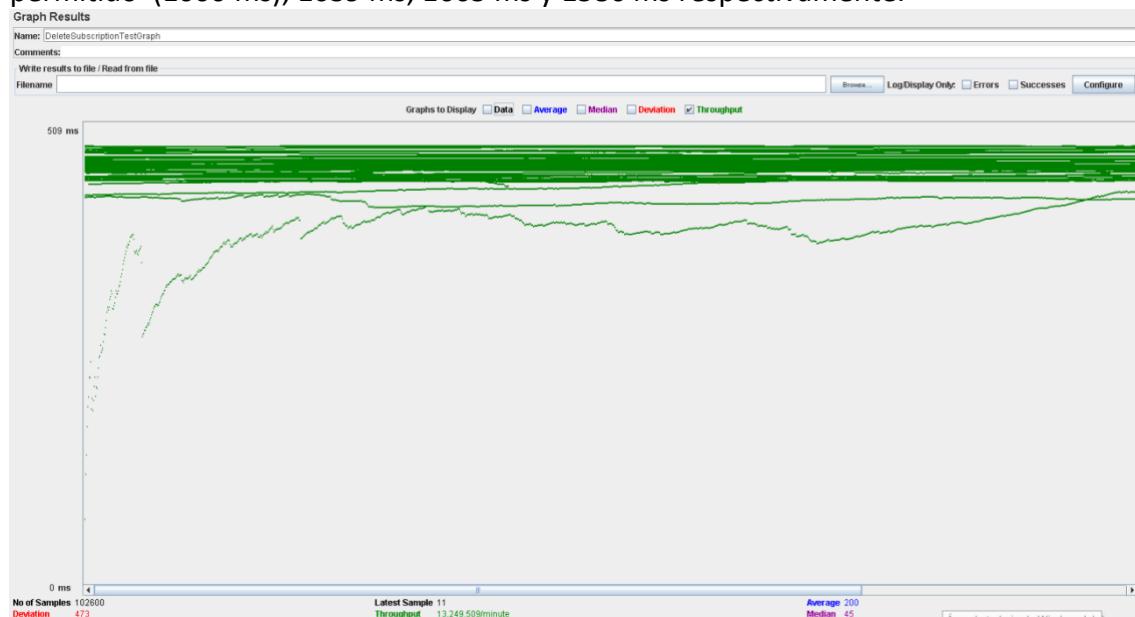
En esta tabla, el rendimiento es de aproximadamente 14.900 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

Volvemos a probar con una configuración 180 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero esperas demasiado altas, que podemos ver en el Aggregate Report y en el Graph Results:



Como podemos observar fijándonos en 90% Line, las direcciones /j\_spring\_security\_check, /subscriptionNewspaper/customer/list.do y

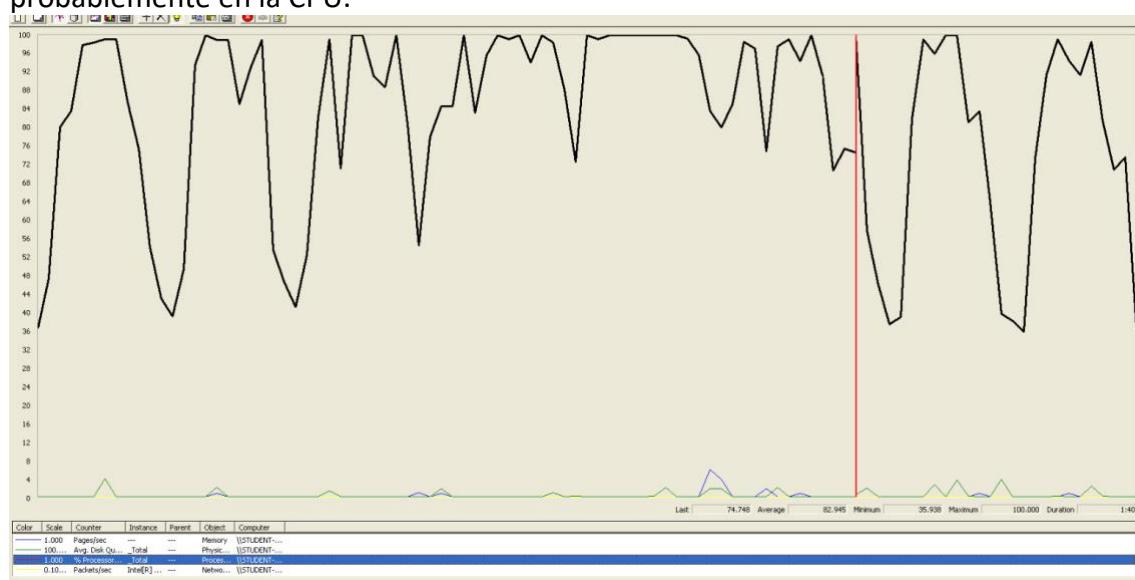
/subscriptionNewspaper/customer/edit.do generan un tiempo de respuesta mayor lo permitido (1000 ms), 1639 ms, 1665 ms y 1586 ms respectivamente.



En esta tabla, el rendimiento es de aproximadamente 13.200 ejecuciones por minuto (desciende al aumentar los 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 de espera:

Abrimos performance.exe y vemos que el procesador tiene porcentajes muy altos, llegando al 100% en varias ocasiones, por lo que el problema se localizará probablemente en la CPU.

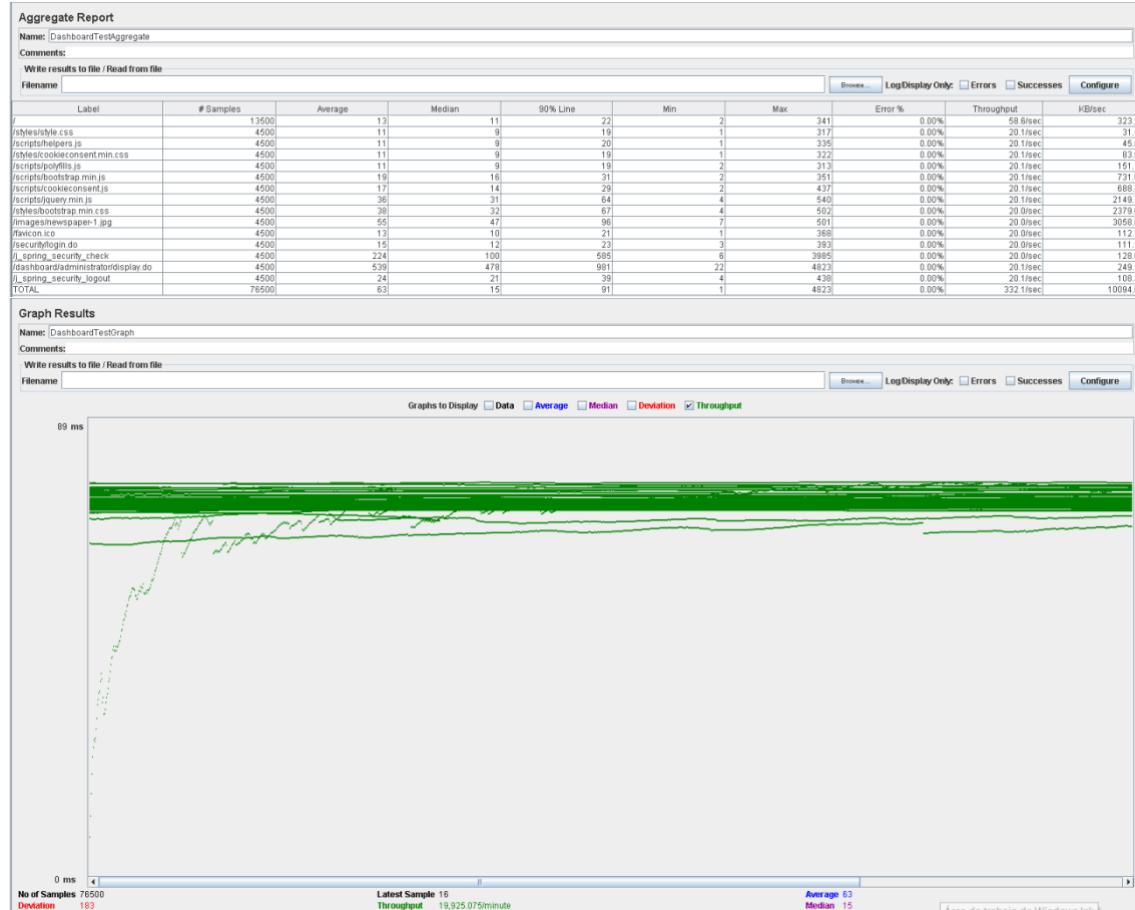


Performance con Processor Time. Casi en todo momento la CPU se encuentra al 100%. Por lo que, en este caso de uso, los usuarios permitidos serán entre 170 y 190.

## Tablón de mandos del administrador.

Nos autenticamos como administrador, vemos el dashboard, navegamos por él, y nos des autenticamos.

Probamos una configuración 150 usuarios, 1 ramp-up y 30 acciones cada uno, no obteniendo errores y aunque las esperas no son excesivamente altas, aunque están en el límite de lo permitido:



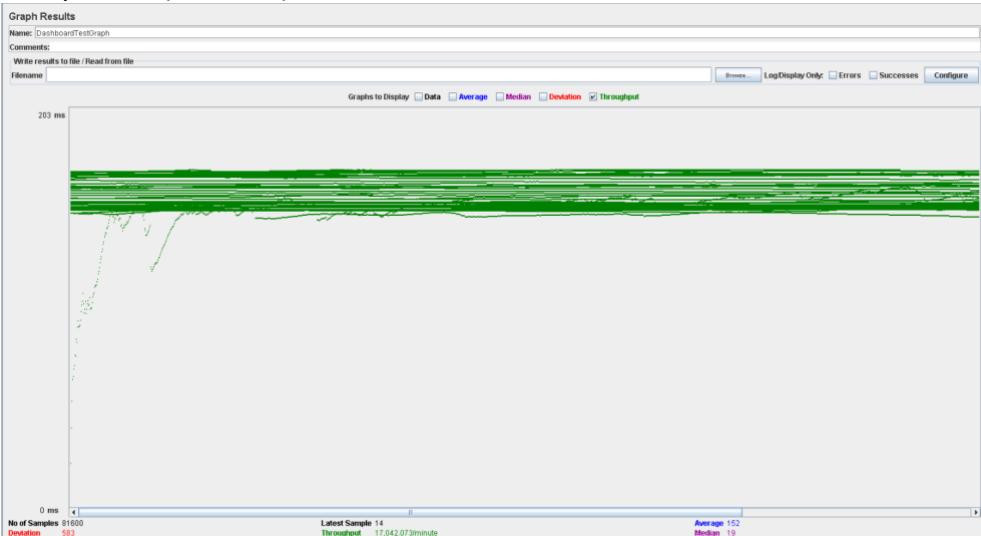
En esta tabla, el rendimiento es de aproximadamente 19.900 ejecuciones por minuto. El rendimiento inicial es pequeño porque los hilos están comenzando y aumenta a medida que aumenta la carga de trabajo y el sistema inicia el almacenamiento en caché.

En un caso real, no sería normal encontrar un número tan alto de administradores en la aplicación. No obstante, hemos considerado suficiente ver un valor alto para así tener una estimación bastante buena tanto de los errores como de los tiempos de ejecución.

Volvemos a probar con una configuración 160 usuarios, 1 ramp-up y 30 acciones cada uno, no produciéndose errores, pero con esperas ya demasiado altas, que podemos ver en el Aggregate Report y en el Graph Results:

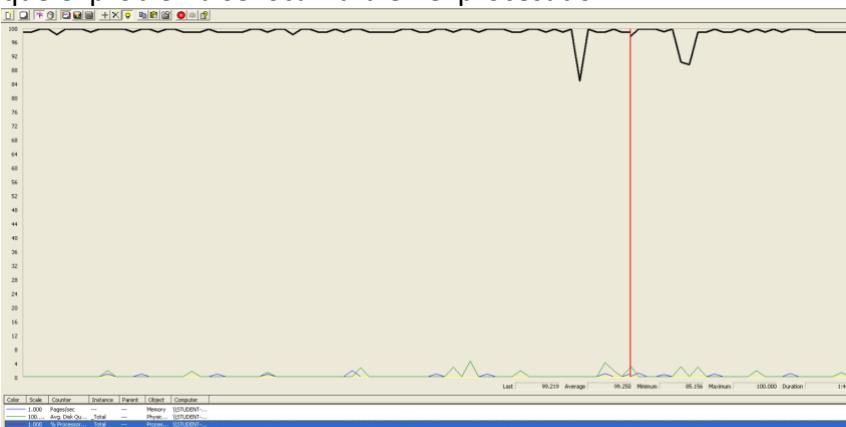
Aggregate Report									<input type="button" value="Browse"/>	<input type="checkbox"/> Log	<input type="checkbox"/> Display Only	<input type="checkbox"/> Errors	<input type="checkbox"/> Successes	<input type="button" value="Configure"/>
Name: DashboardTestAggregate														
Comments:									<input type="checkbox"/> Write results to file / Read from file					
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec					
/	14400	21	14	31	3	3695	0.00%	50.1/sec	276.6					
/styles/style.css	4800	16	12	27	1	1091	0.00%	17.1/sec	26.6					
/scripts/helpers.js	4800	16	11	27	1	1018	0.00%	17.1/sec	39.1					
/scripts/bootstrap.min.css	4800	16	12	27	1	1073	0.00%	17.1/sec	70.3					
/scripts/tether.min.js	4800	16	12	27	1	1089	0.00%	17.1/sec	128.6					
/scripts/bootstrap.min.js	4800	25	20	45	2	1424	0.00%	17.1/sec	623.6					
/scripts/jquery.min.js	4800	48	37	85	4	1002	0.00%	17.1/sec	1833.0					
/scripts/cookieconsent.js	4800	49	39	89	5	1014	0.00%	17.1/sec	2016.5					
/images/newspaper-1.jpg	4800	72	67	127	2	1876	0.00%	17.1/sec	2600.3					
/favicon.ico	4800	21	13	33	2	819	0.00%	17.1/sec	95.7					
/usersiflyt16	4800	23	15	35	3	1075	0.00%	17.1/sec	94.1					
/spring-security-check	4800	893	315	2462	7	2839	0.00%	17.1/sec	109.4					
/dashboard/administrator/display.do	4800	1280	808	2996	19	15521	0.00%	17.1/sec	212.7					
/_spring_logout	4800	37	27	59	4	1155	0.00%	17.1/sec	92.3					
TOTAL	81600	152	19	191	1	15521	0.00%	284.0/sec	8634.2					

Como podemos observar fijándonos en 90% Line, la dirección /dashboard/administrator/display.do genera un tiempo de respuesta mucho mayor a lo esperado (1000 ms), un total de 2866 ms.



En esta tabla, el rendimiento es de aproximadamente 17.000 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.

Por lo que, en este caso de uso, los usuarios permitidos serán entre 150 y 160. Ahora estudiaremos a que se deben los tiempos de espera. Abrimos performance.exe y vemos que la CPU tiene porcentajes muy altos, llegando al 100% en ocasiones, por lo que el problema se localizará en el procesador.



Performance con Processor Time. Casi en todo momento la CPU se encuentra al 100%.

## Conclusión

El mínimo número de usuarios de forma concurrente que pueden utilizar el sistema realizando acciones de forma simultánea es de 80. 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 80.

Por otro lado, como hemos podido observar, en la mayoría de las ocasiones el procesador es nuestro cuello de botella.