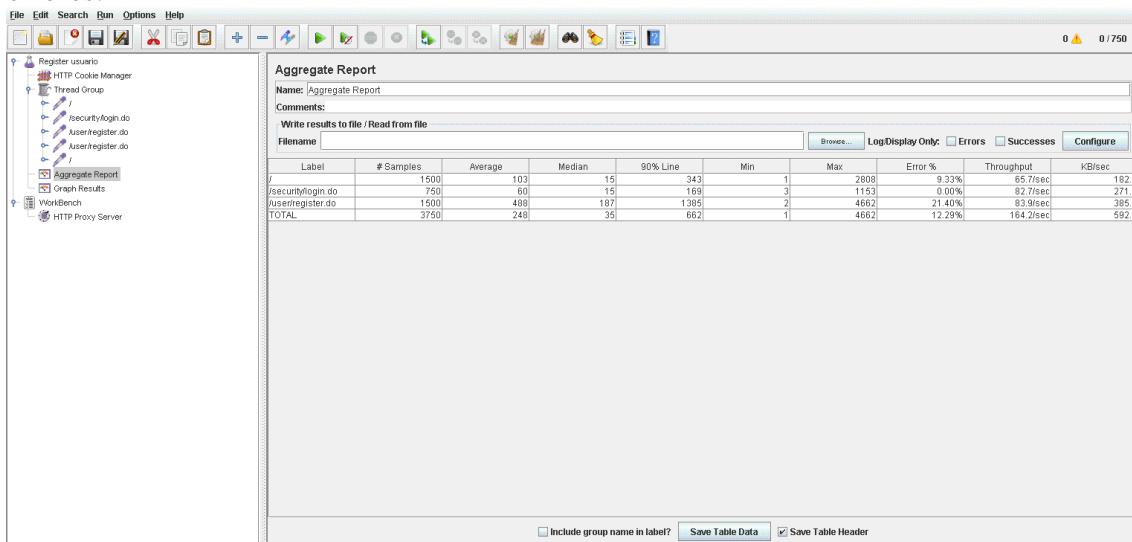
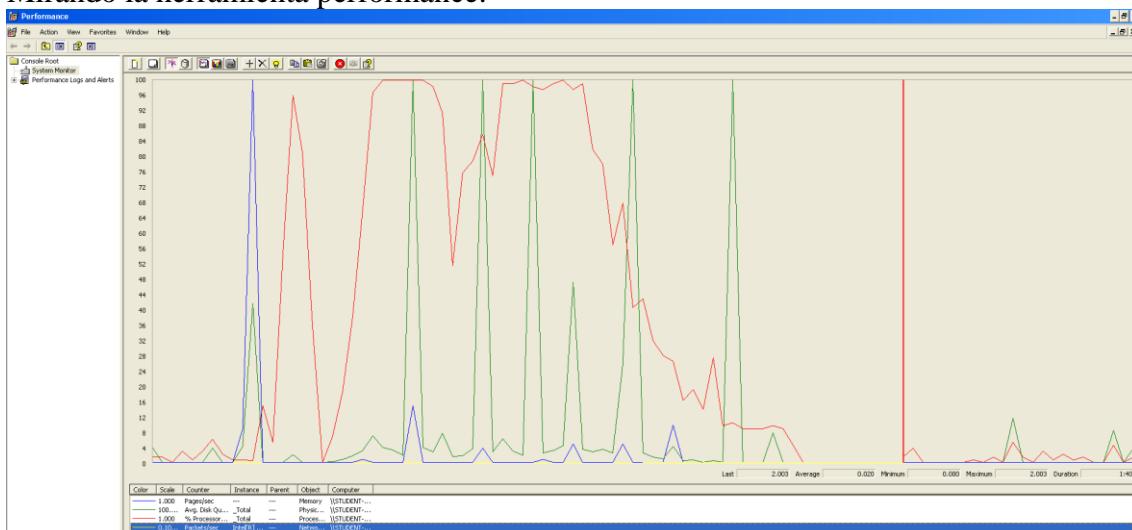


UC1: Registrar Como Usuario (Acme-Newspaper 4.1)

Para este caso de uso hemos comenzado usando 10,50,150,300,500, 750 usuarios respectivamente. Hasta 500 todo ha ido bien, pero con 750 usuario comienzan a haber errores:

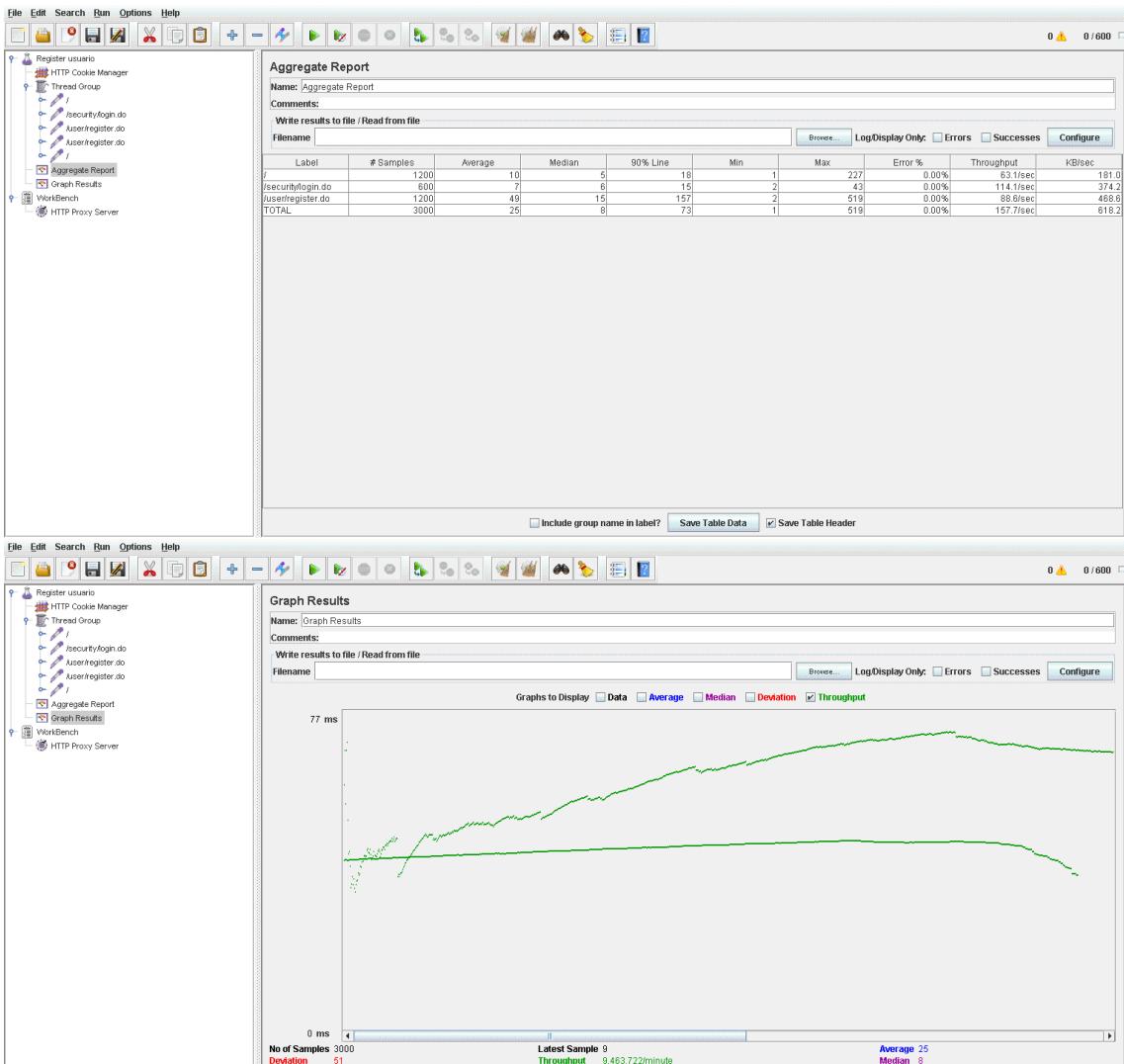


Mirando la herramienta performance:



Vemos que se está formando un cuello de botella y que la CPU, Disco duro y Ram hacen picos.

Bajamos los usuarios para encontrar la capacidad máxima del sistema y probamos con 600:



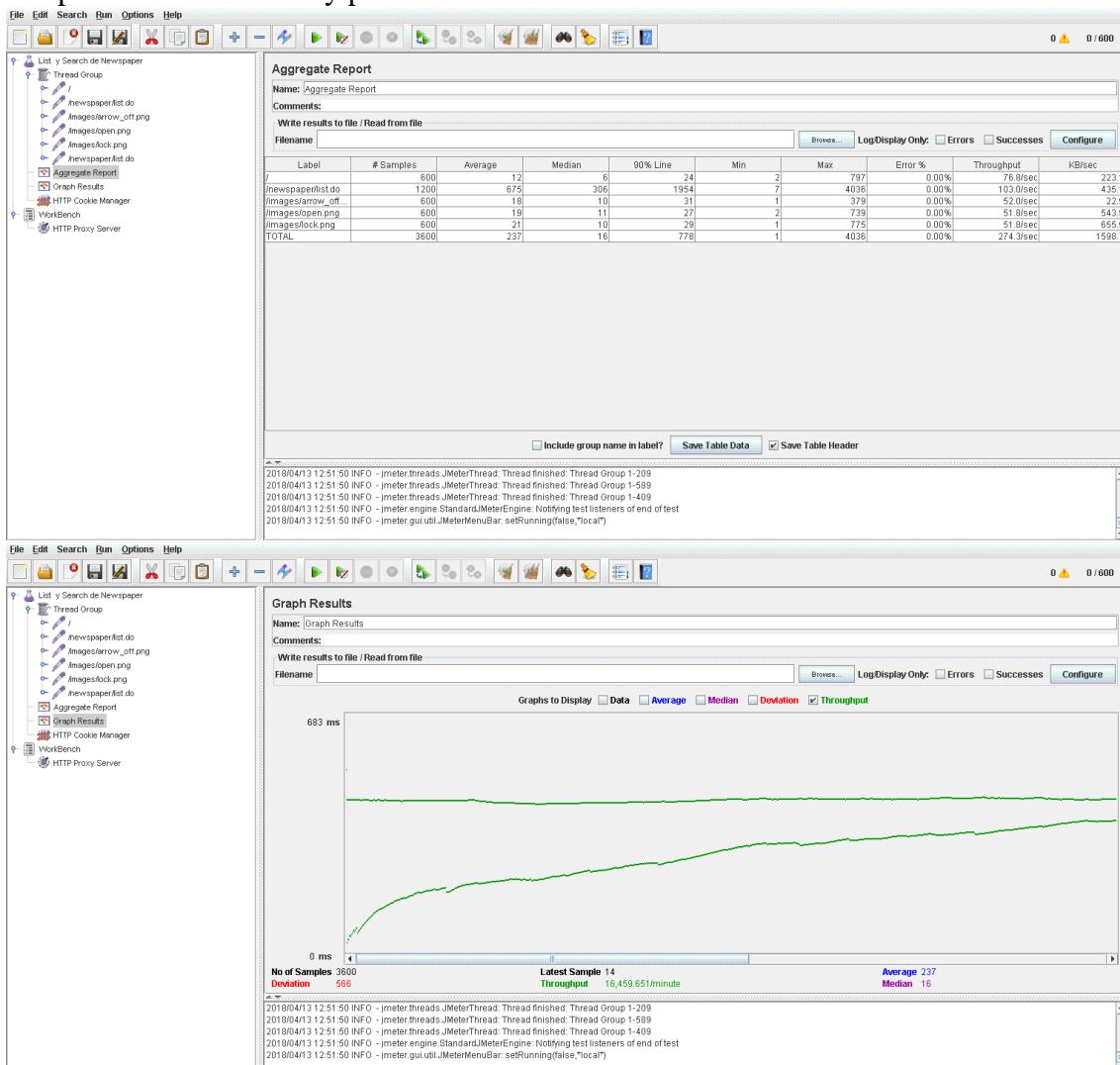
Con 600, 3 segundos de comienzo y Loop = 1 no hay problema alguno. Pero si cambiamos el Loop para que cada usuario ejecute el script más de una vez vuelven a haber errores, por lo que fijamos el número de usuarios máximo a 600.

Usuarios máximos: 600

UC2: Listar y buscar newspapers (Acme-Newspaper 4.2/4.5)

Para este caso de uso hemos ido incrementando el número de usuarios de la misma manera sin tener ningún problema hasta nuevamente llegar a los 600.

En este punto ocurre igual que en los casos anteriores, con 600 usuarios ejecutando el Script una sola vez no hay problemas:



Pero si intentamos que 600 personas ejecuten el script más de una vez (Loop = 3) empiezan a haber errores, lo mismo pasa si intentamos que más personas (750 en este caso) ejecute el Script una sola vez:

File Edit Search Run Options Help

Aggregate Report
Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename [] LogDisplay Only: Errors Successes

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
j	750	123	15	507	2	826	3.73%	65.4/sec	190.3
/newspaperlist.do	1500	536	337	1365	10	3170	0.33%	95.9/sec	404.3
/images/arrow_cff.png	750	135	32	359	1	1478	0.53%	48.3/sec	21.7
/images/open.png	750	125	29	348	2	1143	0.40%	48.2/sec	504.6
/images/lock.png	750	64	28	243	1	1562	0.00%	48.1/sec	609.9
TOTAL	4500	256	58	931	1	3170	0.09%	264.9/sec	1537.4

 Include group name in label? Save Table Header

```
2018/04/13 12:47:16 INFO - jmeter.threads.JMeterThread: Thread finished: Thread Group 1-437
2018/04/13 12:47:16 INFO - jmeter.threads.JMeterThread: Thread finished: Thread Group 1-463
2018/04/13 12:47:16 INFO - jmeter.threads.JMeterThread: Thread finished: Thread Group 1-219
2018/04/13 12:47:16 INFO - jmeter.engine.StandardJMeterEngine: Nothing test listeners of end of test
2018/04/13 12:47:16 INFO - jmeter.gui.util.JMeterMenuBar: setRunning(false,"local")
```

File Edit Search Run Options Help

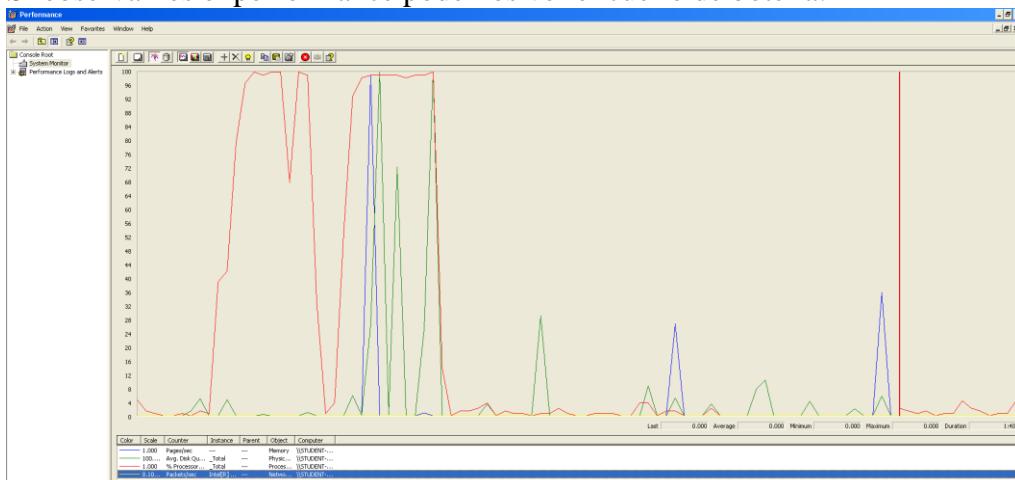
Aggregate Report
Name: Aggregate Report
Comments:
Write results to file / Read from file
Filename [] LogDisplay Only: Errors Successes

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
j	600	100	15	364	2	1001	0.00%	89.3/sec	243.0
/newspaperlist.do	1300	919	650	2169	11	4494	0.05%	101.3/sec	405.1
/images/arrow_cff.png	650	284	25	1110	1	2096	5.85%	51.6/sec	30.2
/images/open.png	650	216	30	449	1	2148	2.00%	51.4/sec	530.6
/images/lock.png	650	122	28	409	2	2085	3.23%	51.1/sec	630.0
TOTAL	3900	438	68	1560	1	4494	5.46%	277.1/sec	1577.2

 Include group name in label? Save Table Header

```
2018/04/13 12:50:34 INFO - jmeter.threads.JMeterThread: Thread finished: Thread Group 1-611
2018/04/13 12:50:34 INFO - jmeter.threads.JMeterThread: Thread finished: Thread Group 1-395
2018/04/13 12:50:34 INFO - jmeter.threads.JMeterThread: Thread finished: Thread Group 1-465
2018/04/13 12:50:34 INFO - jmeter.engine.StandardJMeterEngine: Nothing test listeners of end of test
2018/04/13 12:50:34 INFO - jmeter.gui.util.JMeterMenuBar: setRunning(false,"local")
```

Si observamos el performance podemos ver el cuello de botella:

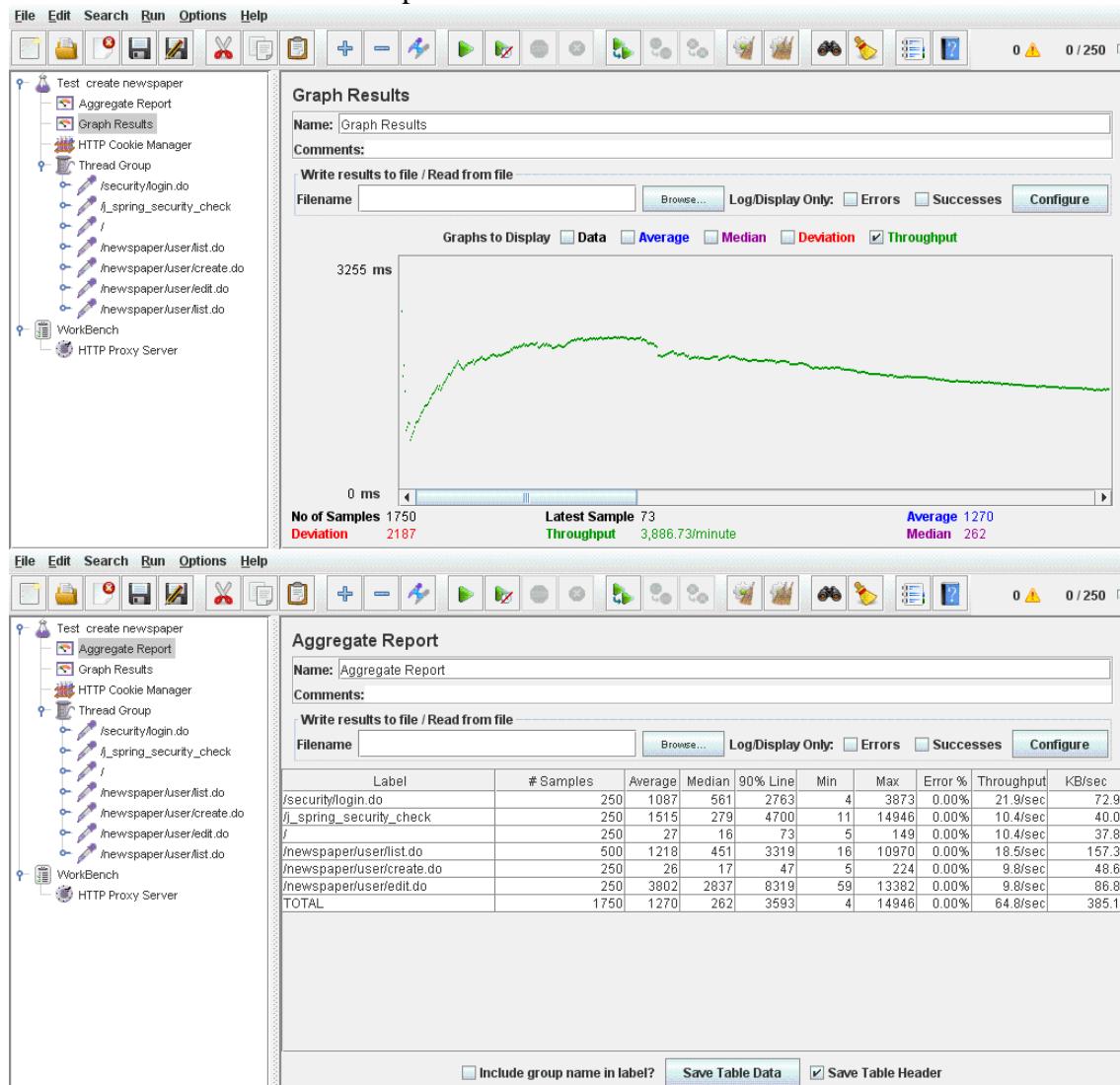


Parece ser causado por la CPU, el disco duro y la memoria RAM.

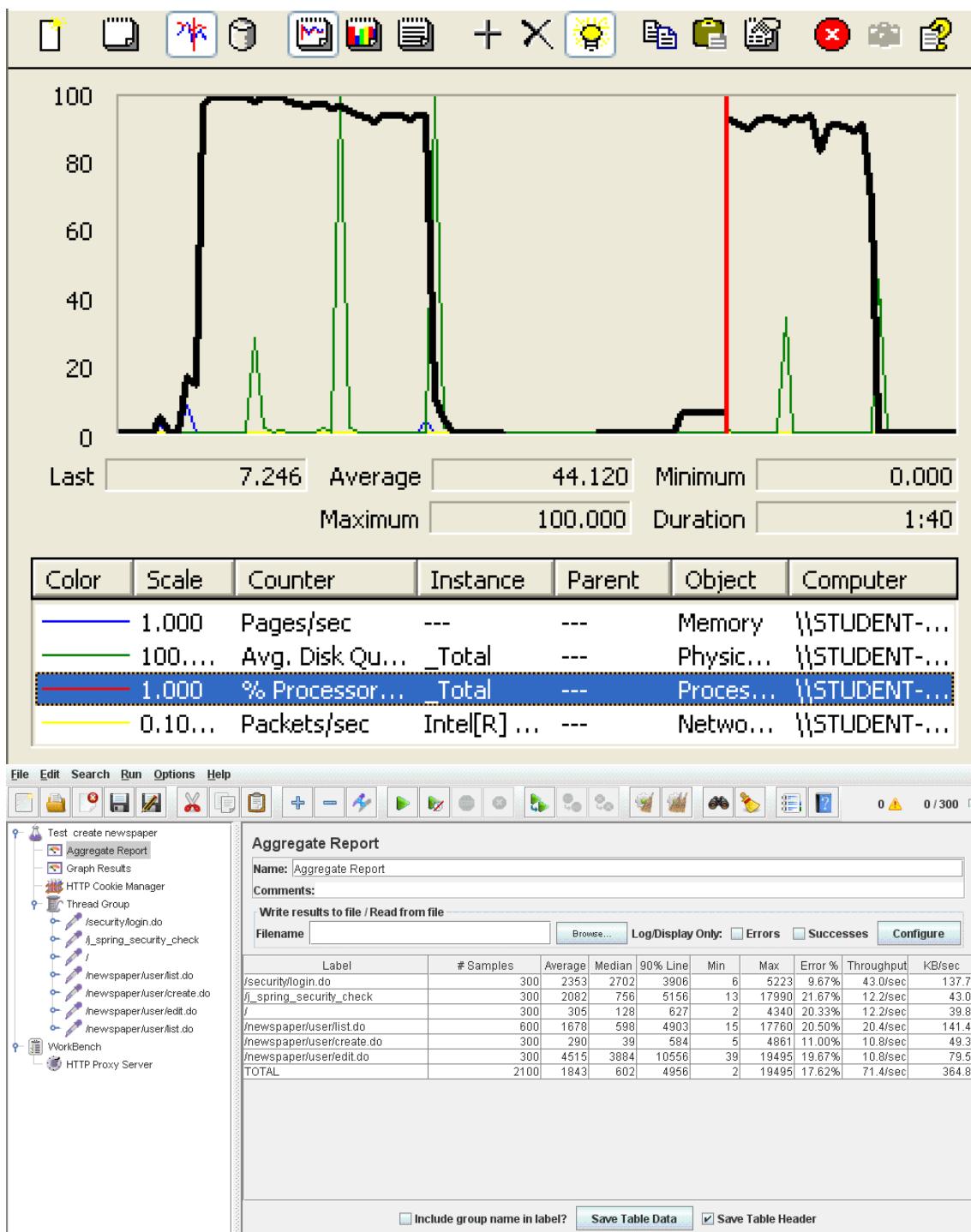
Usuarios máximos: 600

UC3: Create Newspaper (Acme-Newspaper 6.1)

En este caso de uso hemos ido probando hasta alcanzar los 250 usuarios:



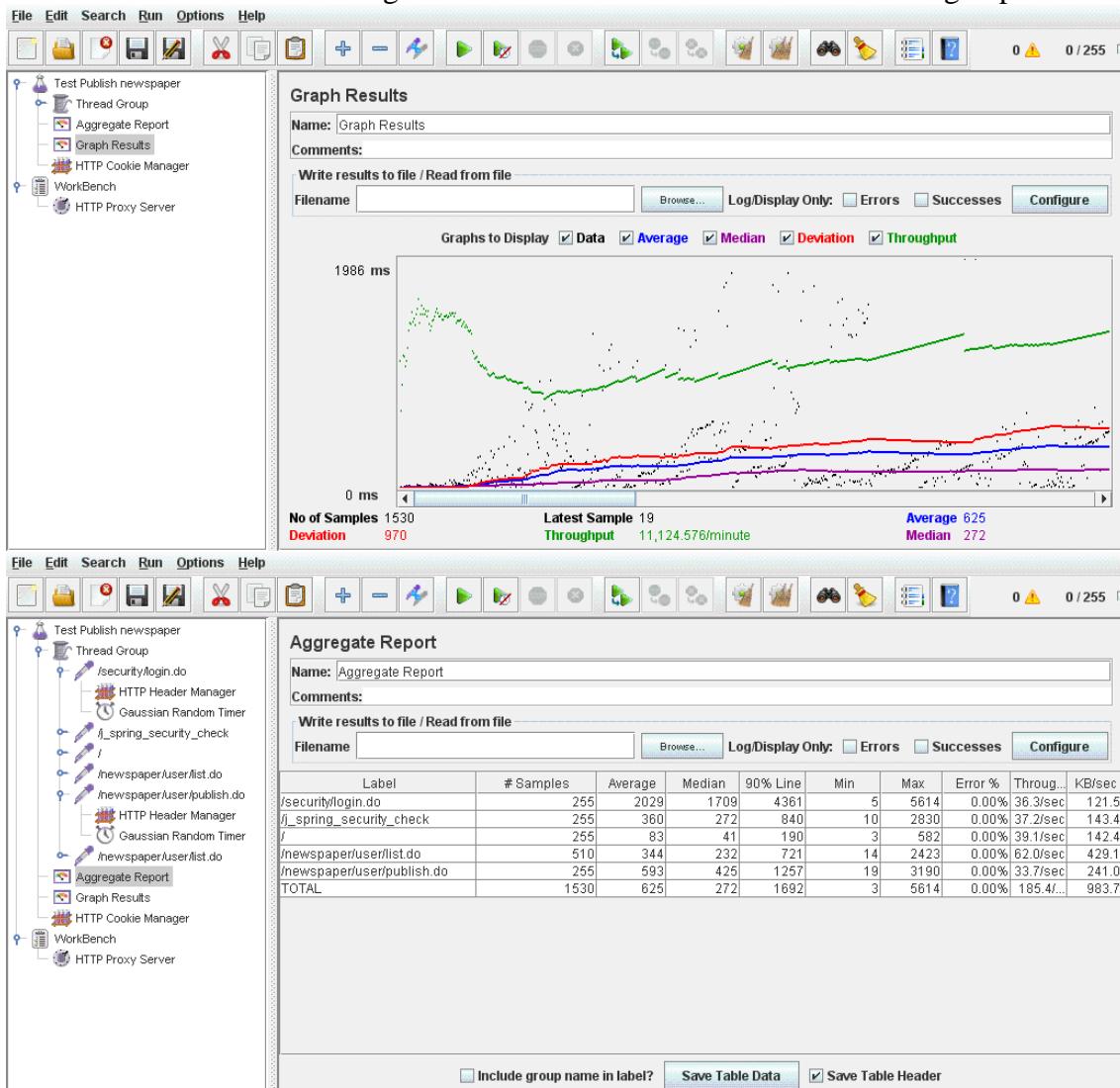
Sin embargo al alcanzar los 300 usuarios comienza a haber cuello de botella causado por el procesador:



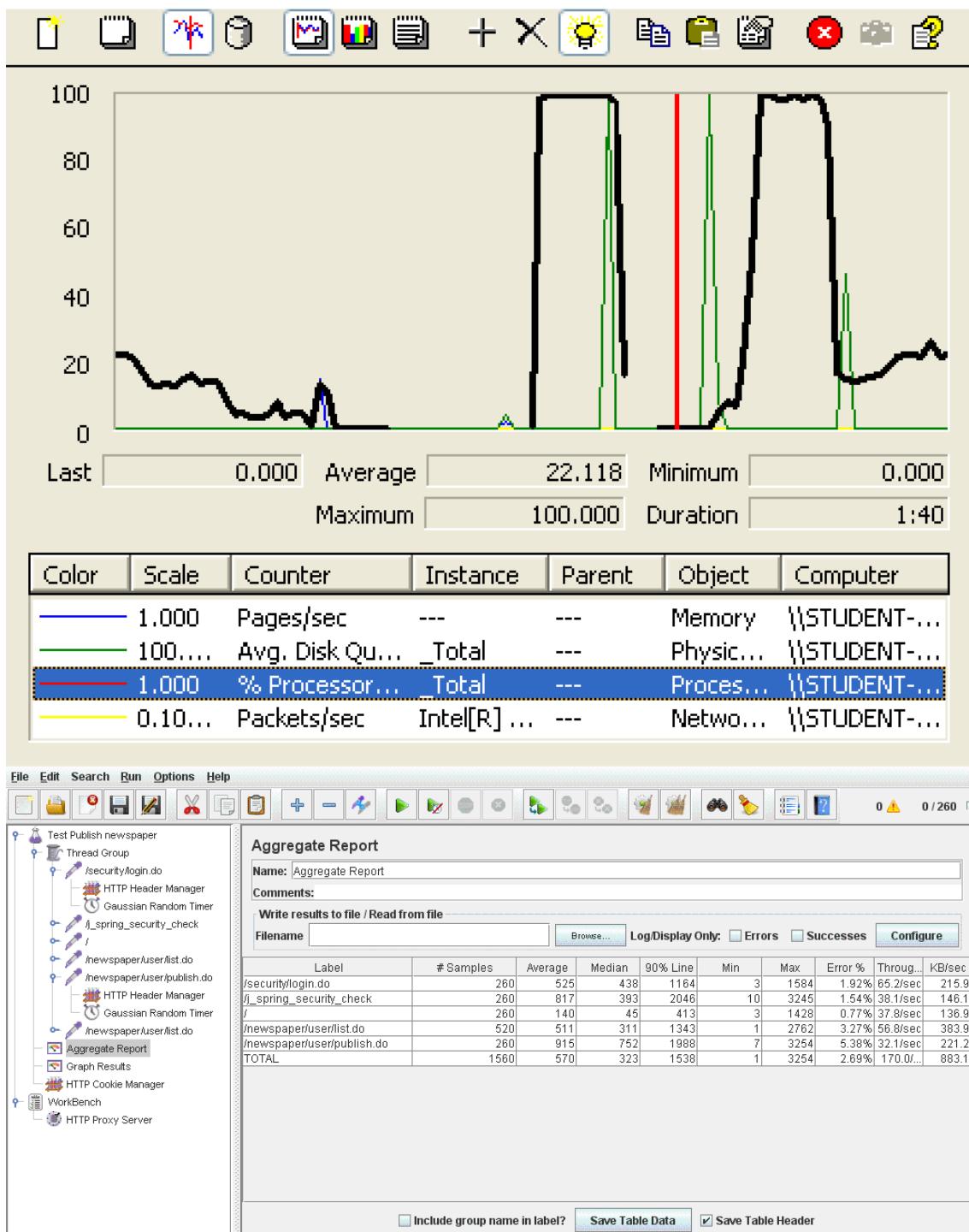
Máx usuarios: 250

UC4: Publicar un newspaper (Acme-Newspaper 6.2)

En este caso de uso hemos llegado hasta los 255 usuarios de límite sin ningún problema:



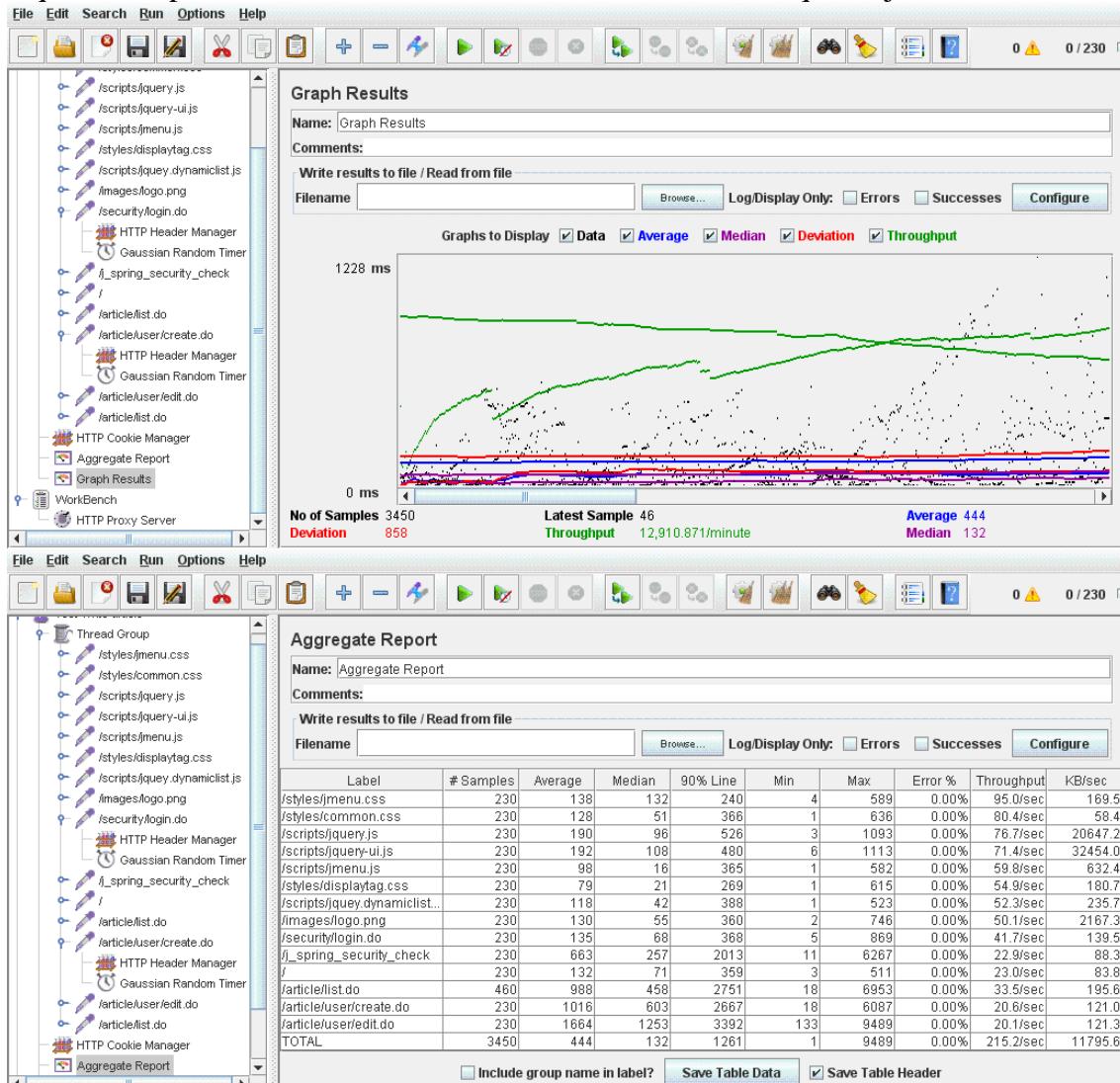
Sin embargo una vez llegados a los 260 comienzan a surgir errores provocados por la CPU y el disco duro:



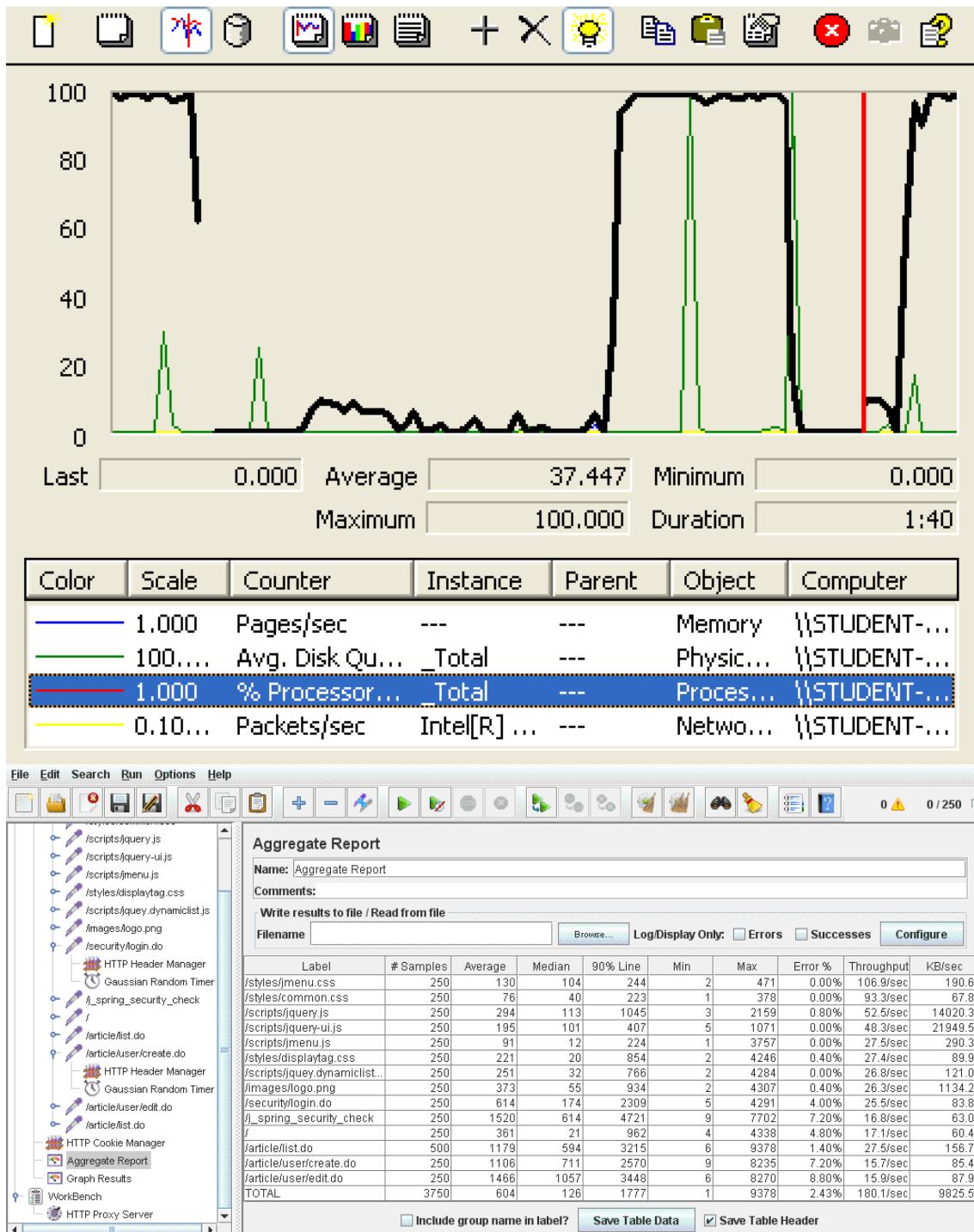
Máximo de usuarios: 255

UC5: Escribir un artículo (Acme-Newspaper 6.3)

Aquí hemos podido alcanzar los 230 usuarios concurrentes sin que surjan errores:



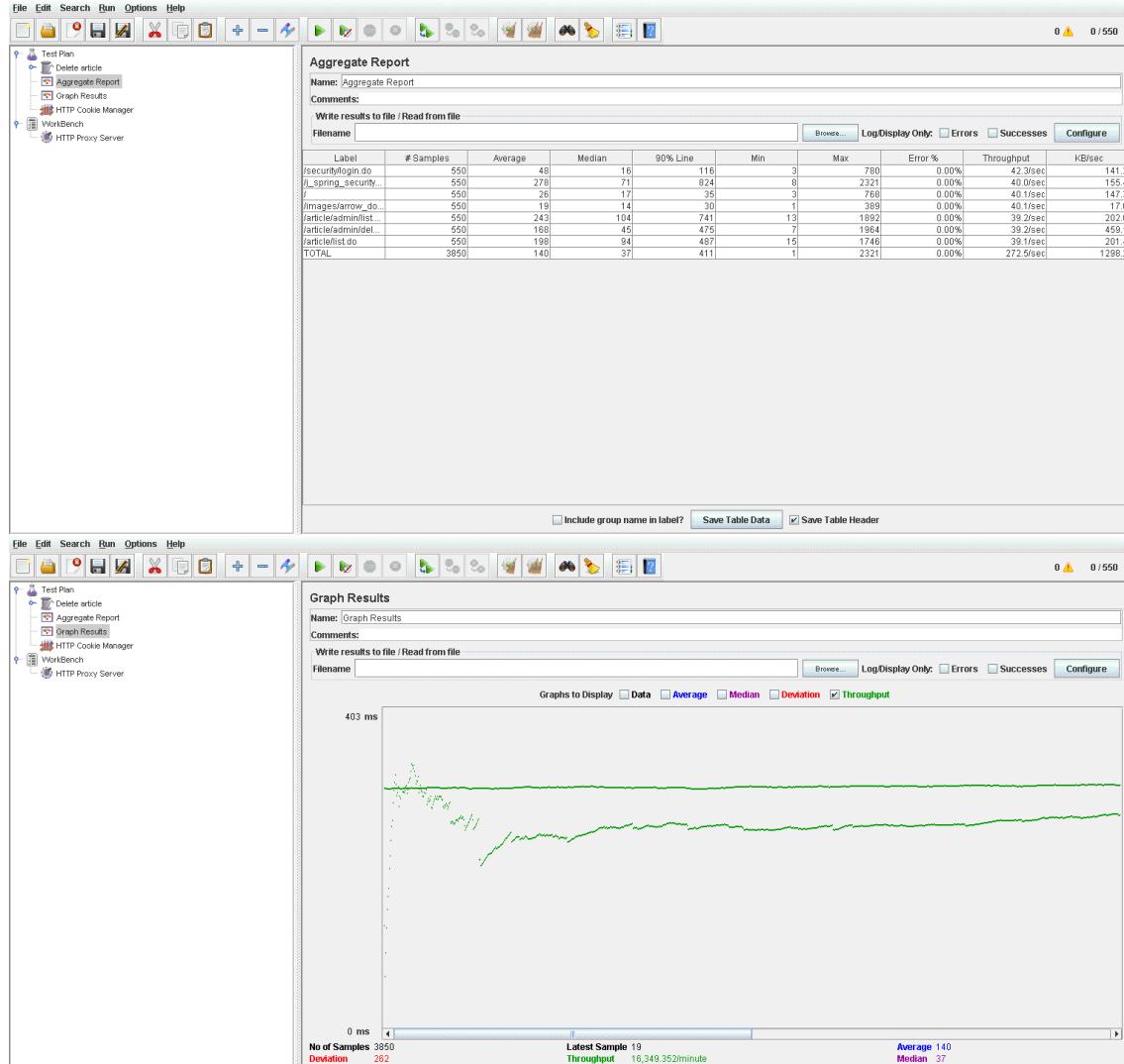
Pero una vez llegado a los 250 usuarios comenzamos a tener un cuello de botella causado por la CPU y el disco duro:



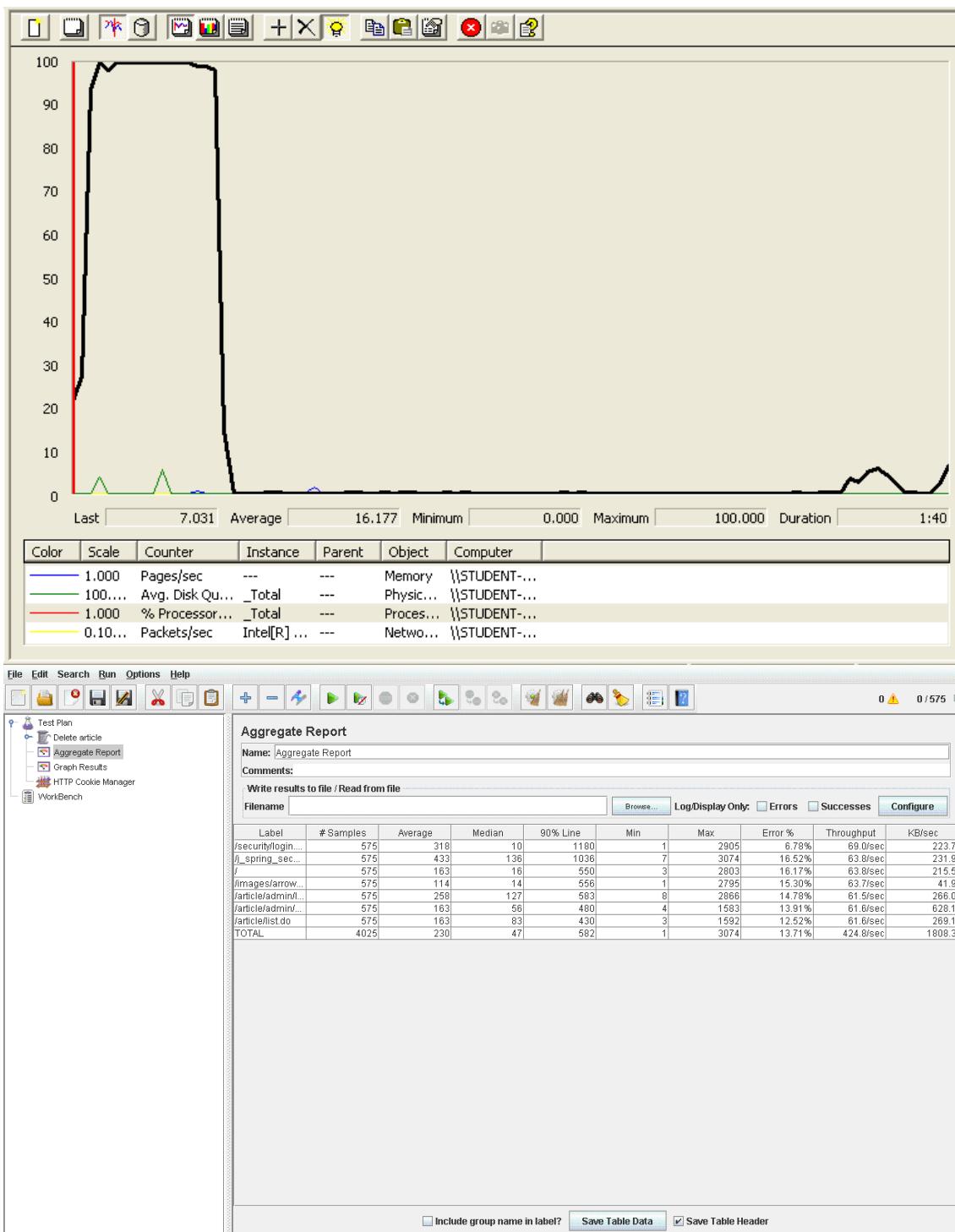
Máximo de usuarios: 230

UC6: Borrar un artículo (Acme-Newspaper 7.1)

En este caso de uso hemos llegado a los 550 usuarios concurrentes sin errores:



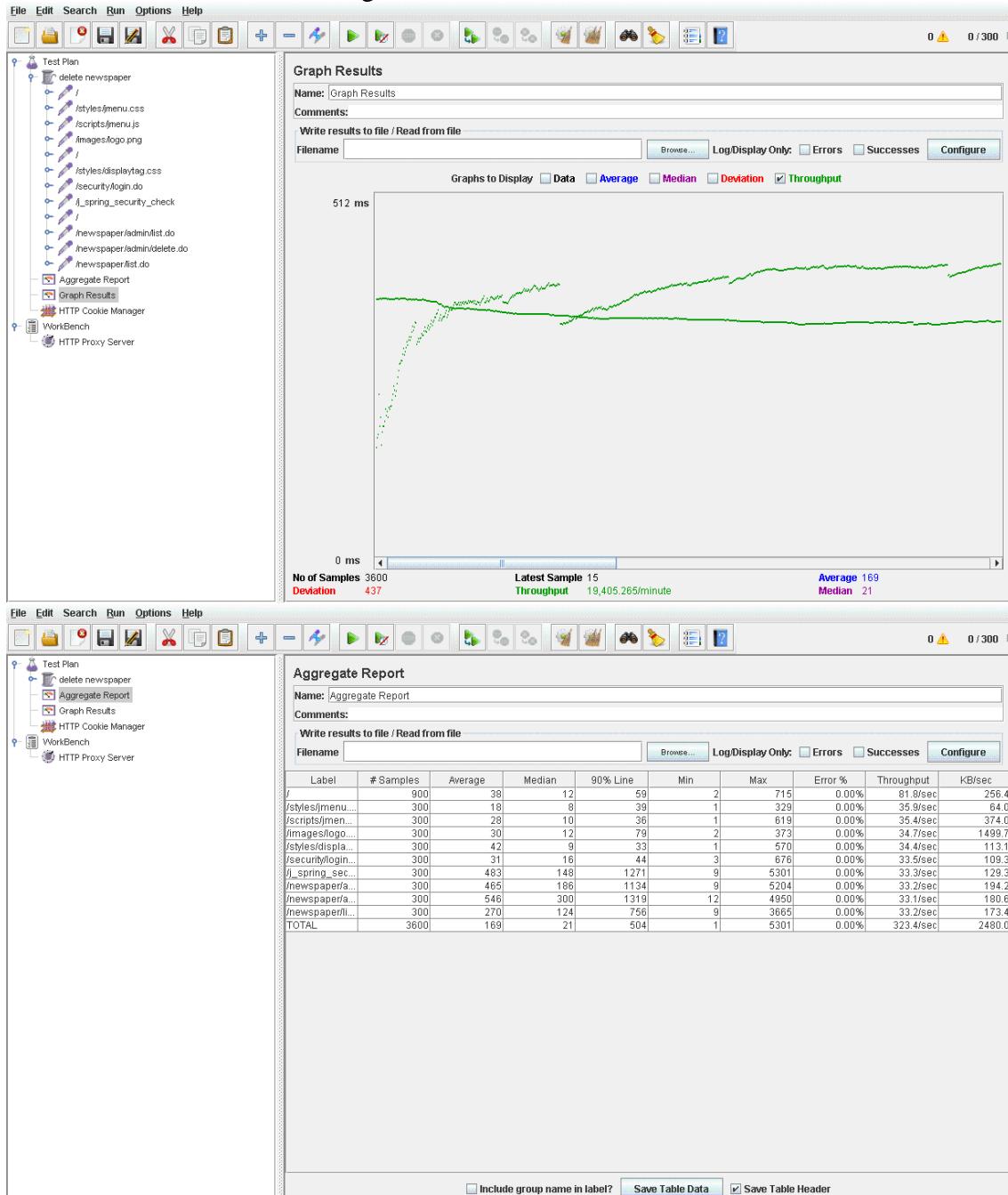
Una vez llegados a los 575 comienza a haber errores por la CPU, que produce un cuello de botella:



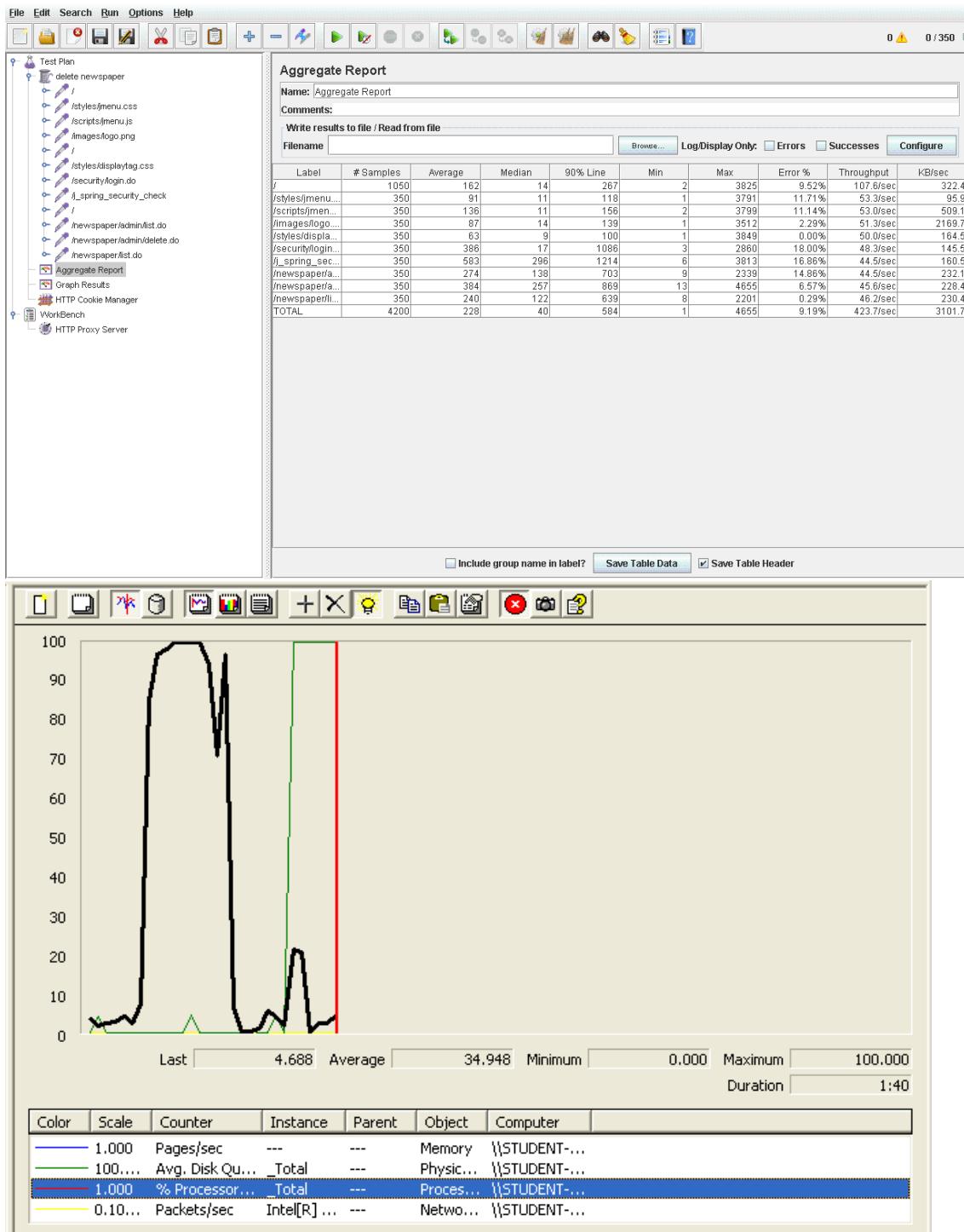
Máx usuarios: 550

UC7: Borrar Newspaper (Acme-Newspaper 7.2)

Con este caso de uso hemos llegado a los 300 usuarios concurrentes sin errores:



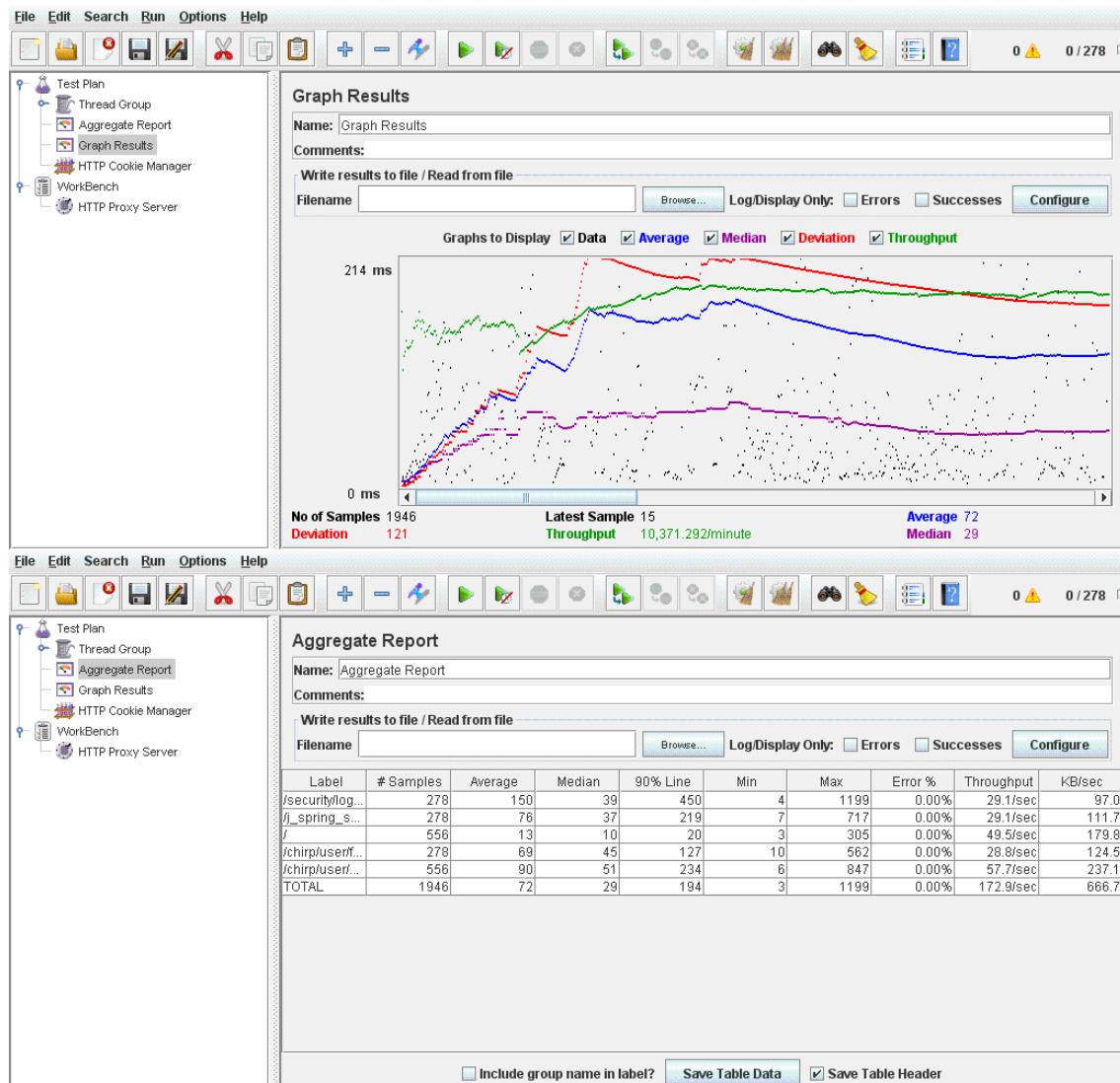
Pero cuando aumentamos hasta 350 usuarios, comienza a producirse un cuello de botella producido por la CPU y el disco duro:



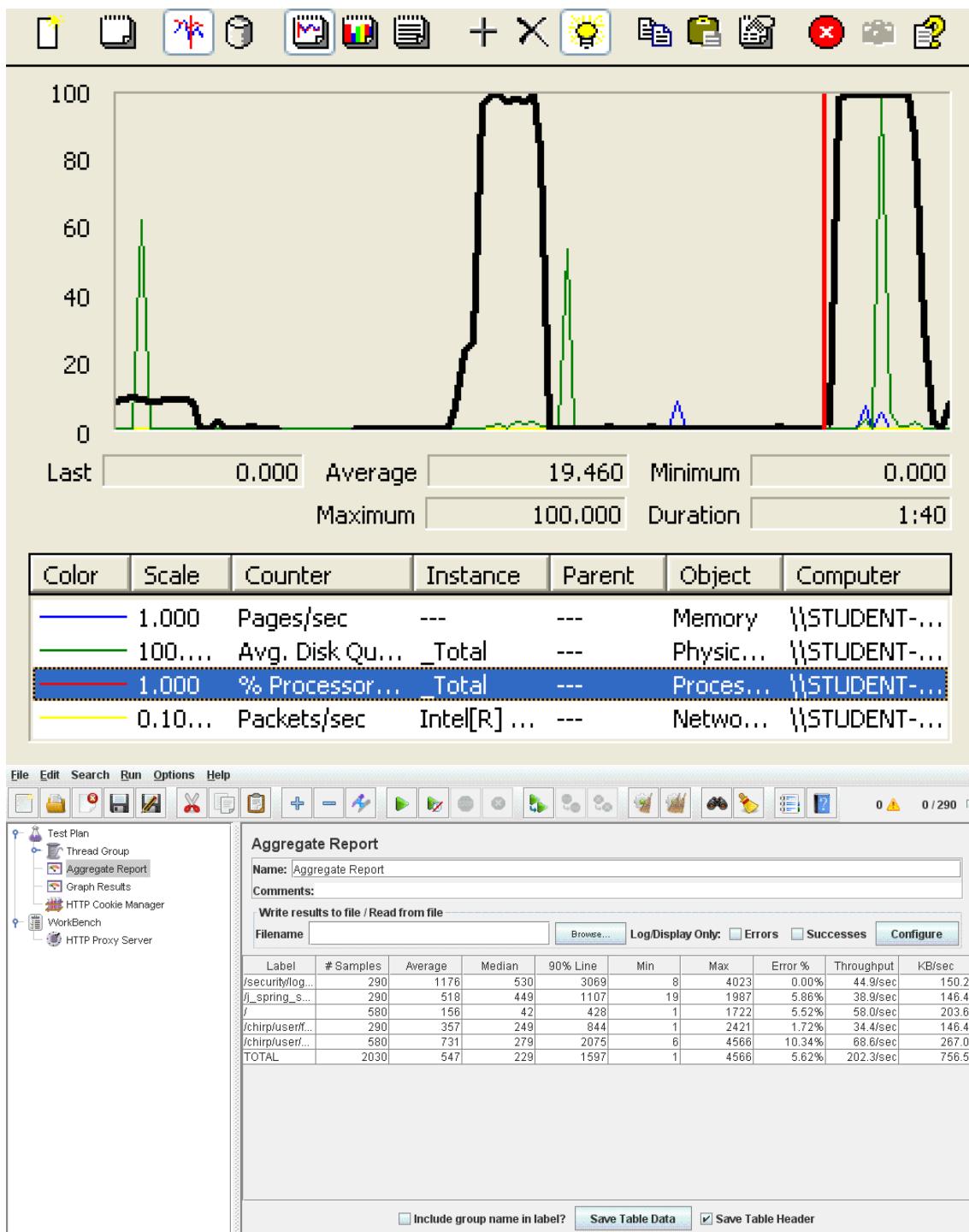
Máximos usuarios: 300

UC8: Listar Chirps y escribir uno (Acme Newspaper 16.1, 16.5)

En este caso de uso hemos llegado a los 278 usuarios concurrentes sin conseguir errores:



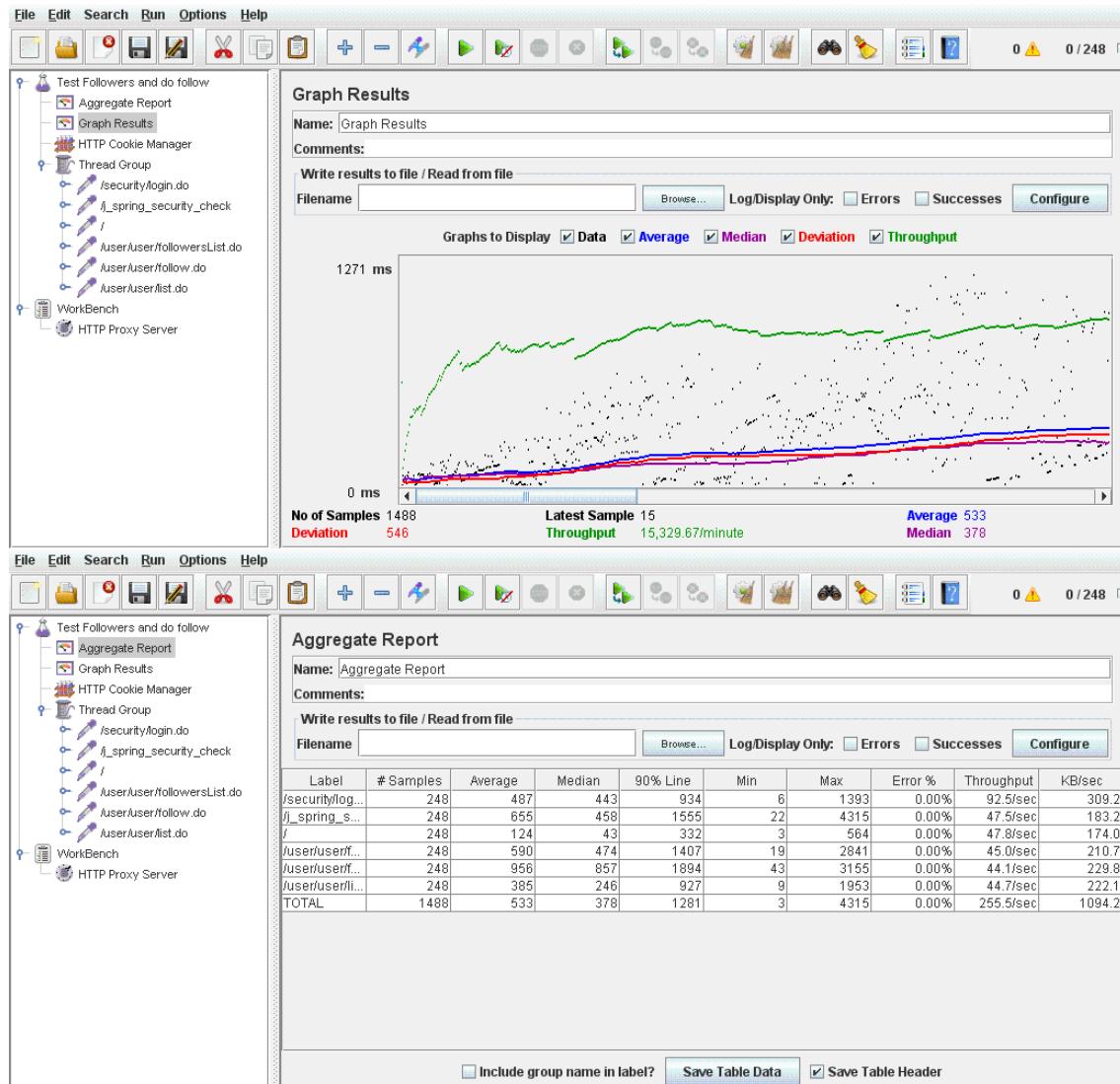
Una vez llegados a los 290 comienzan a haber errores causados por el cuello de botella que produce la CPU y el disco duro:



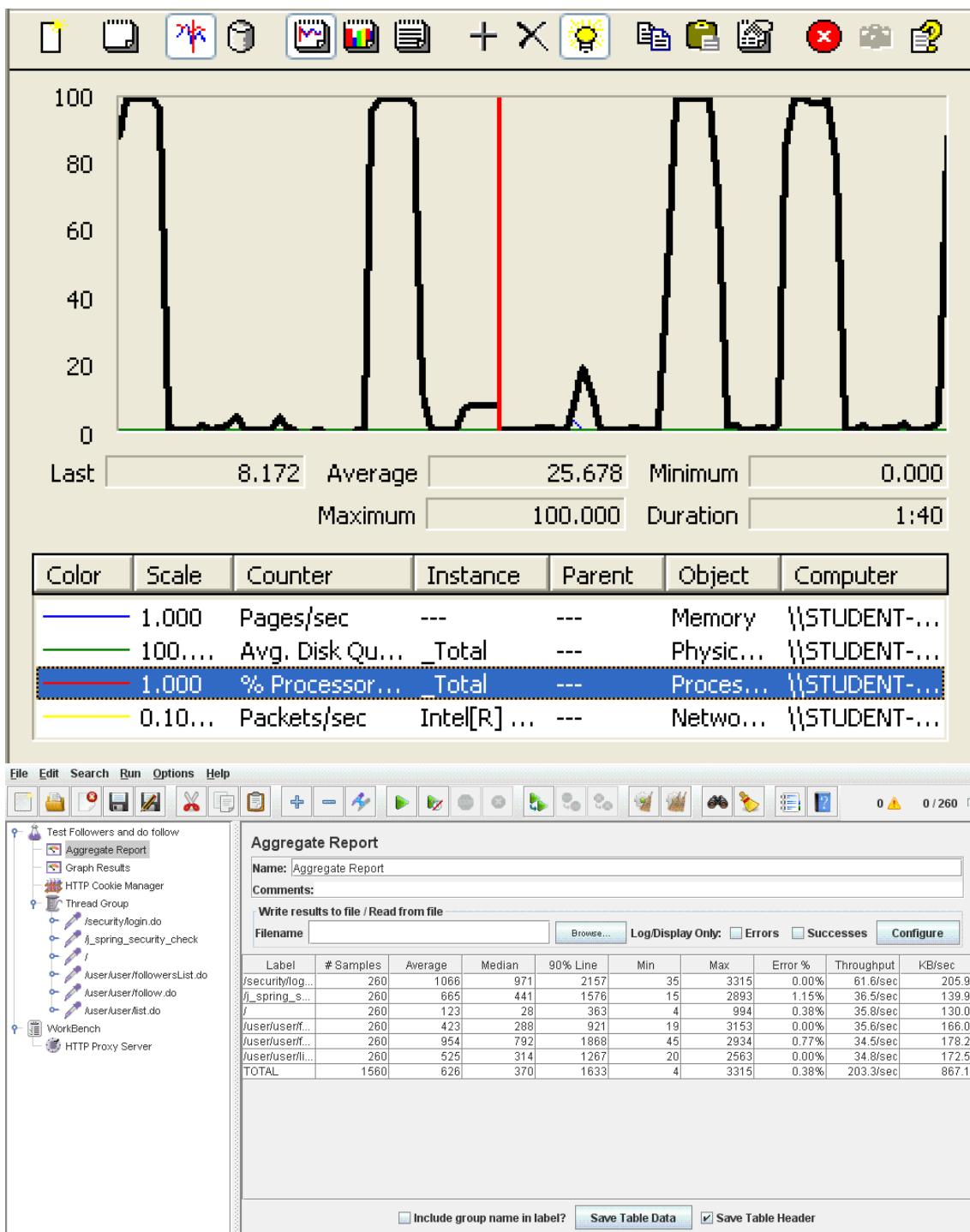
Máximo de usuarios: 278

UC9: Listar Seguidores y hacer follow (Acme-Newspaper 16.2/16.3/16.4)

Para este caso de uso hemos ido probando usuarios hasta alcanzar una cifra límite de 248 usuarios:



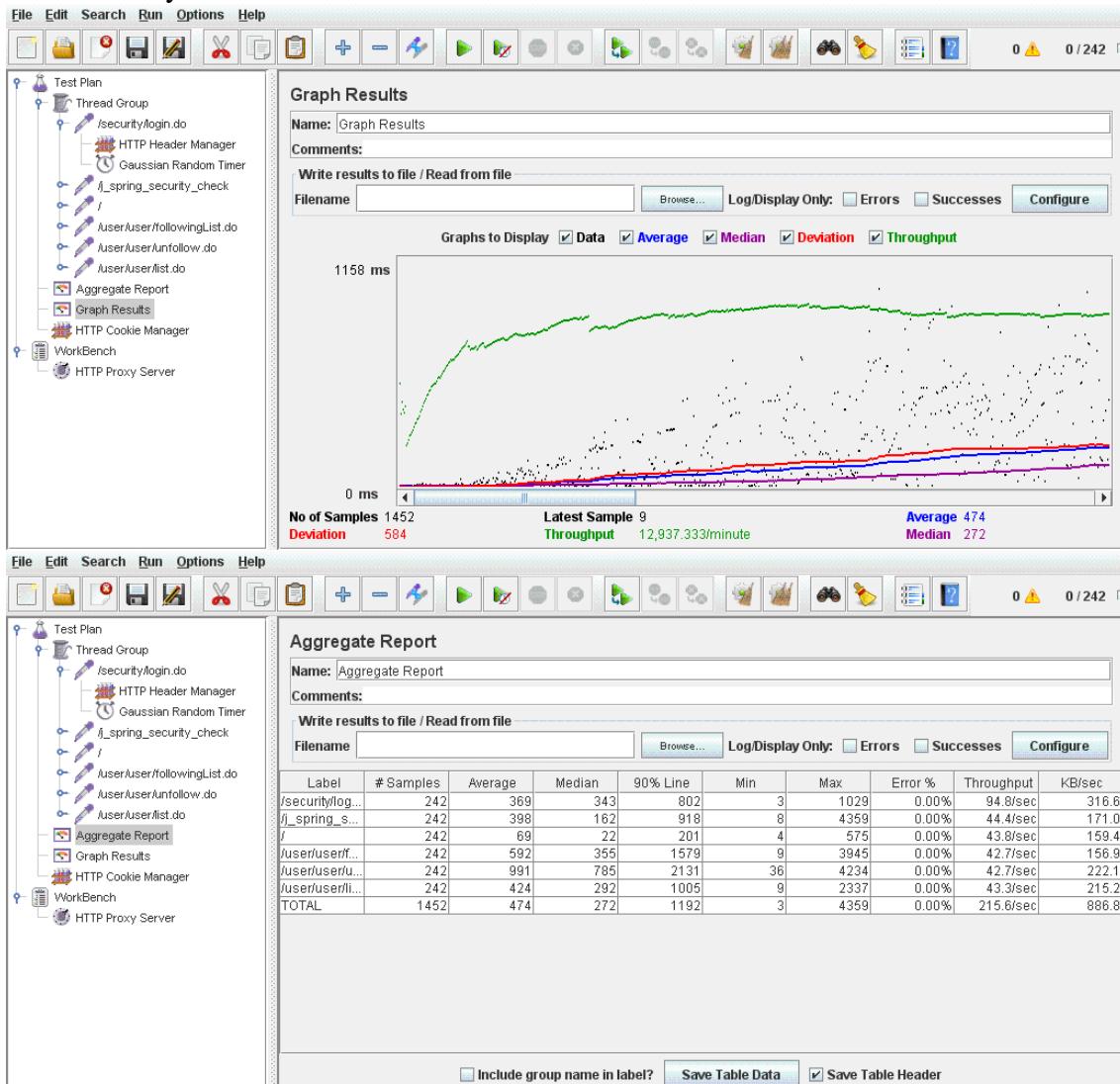
Sin embargo si aumentamos un poco más la cifra, comienzan a haber errores debidos al cuello de botella que produce el procesador:



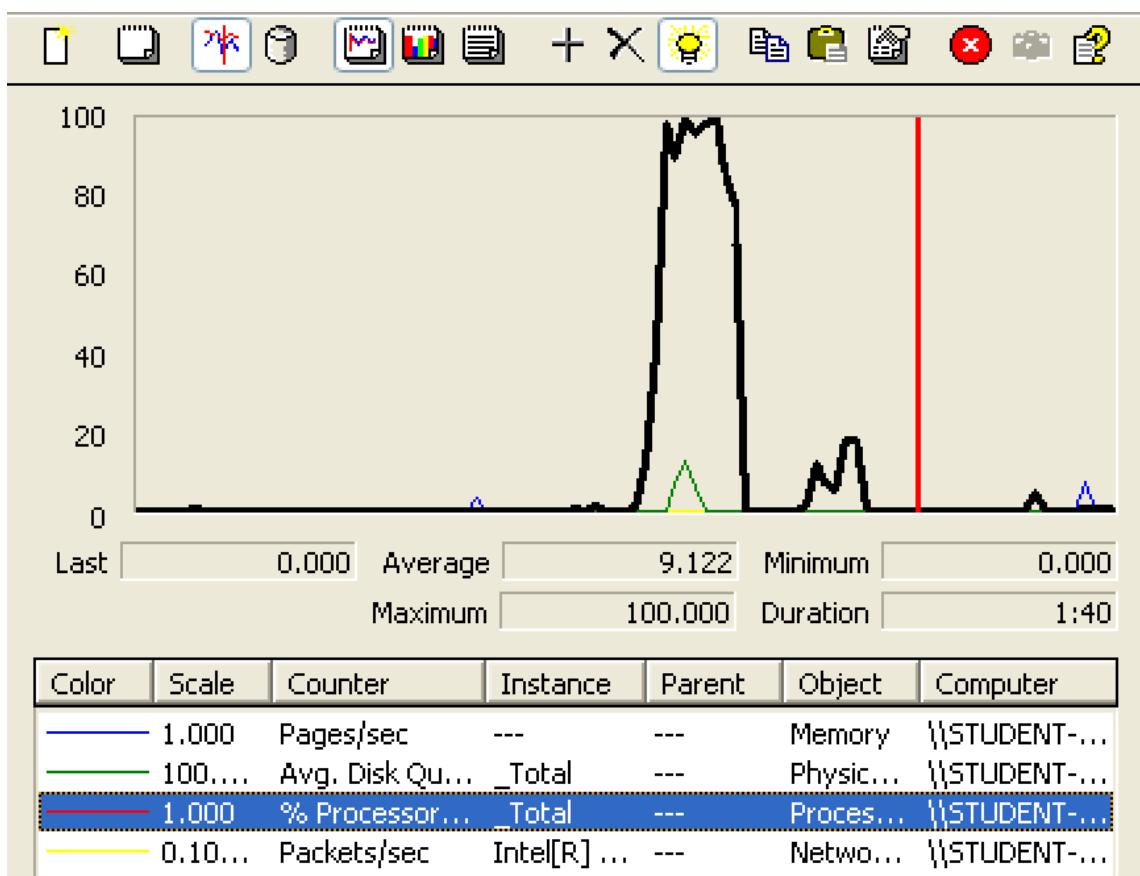
Máximo de usuarios: 248

UC10: Listar seguidores y hacer unfollow (Acme-Newspaper 16.2/16.3/16.4)

Este caso de uso es muy similar al anterior, hemos aproximado los usuarios al caso de uso anterior y hemos obtenido como usuarios límite 242:



Sin embargo al aumentar un poco más los usuarios hasta 247 comienzan a haber errores nuevamente por la CPU:



File Edit Search Run Options Help

Test Plan

- Thread Group
 - /security/login.do
 - HTTP Header Manager
 - Gaussian Random Timer
 - /_spring_security_check
 - /user/user/followingList.do
 - /user/user/unfollow.do
 - /user/UserList.do
- Aggregate Report
- Graph Results
- HTTP Cookie Manager
- WorkBench
- HTTP Proxy Server

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

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

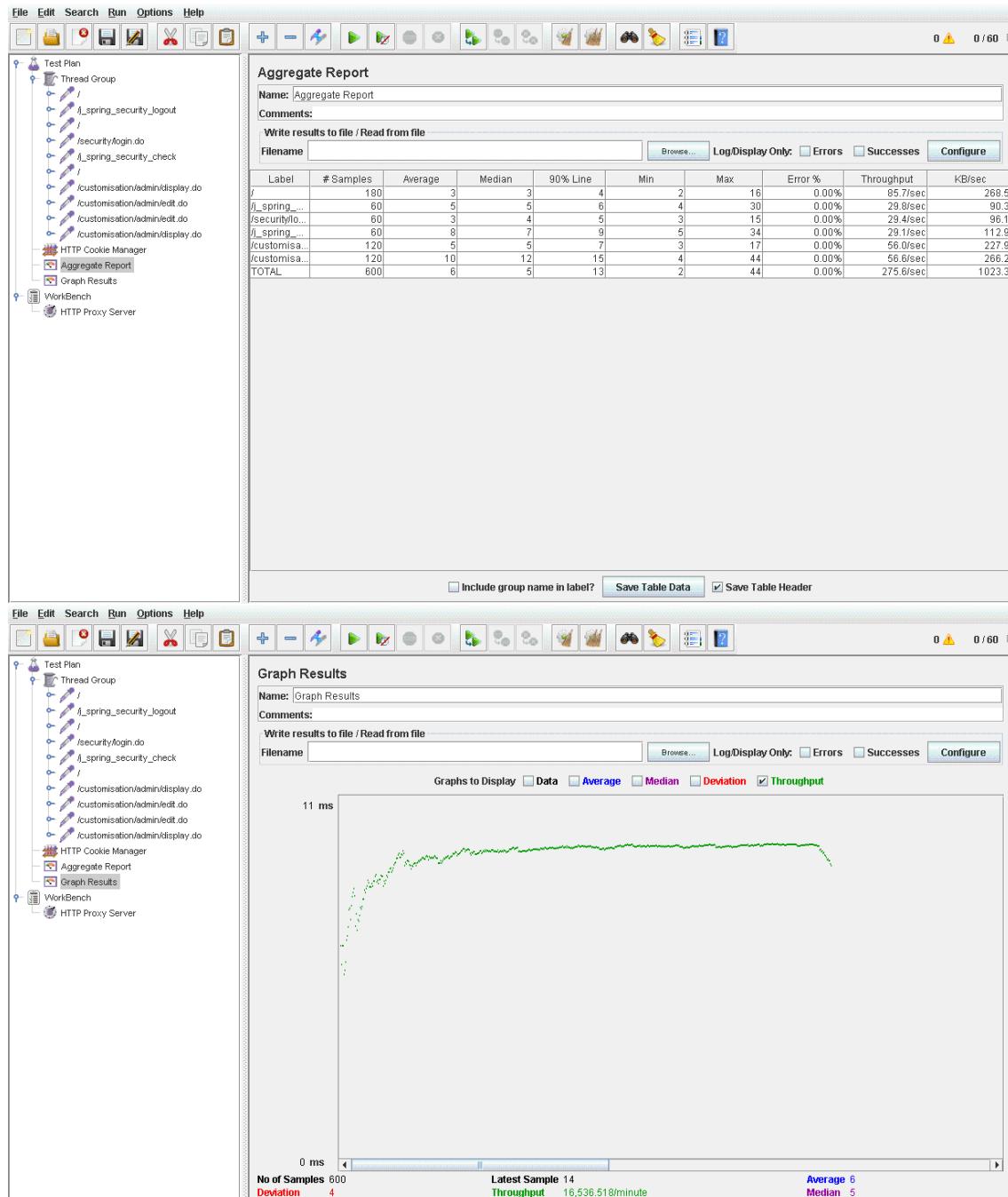
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/log...	247	571	573	1182	4	1570	0.00%	99.4/sec	332.5
/_spring_s...	247	505	387	1221	8	2137	1.21%	56.1/sec	211.4
/	247	98	29	400	3	491	0.00%	54.9/sec	199.8
/user/userf...	247	412	290	910	12	1982	0.00%	50.0/sec	183.2
/user/useru...	247	831	732	1488	89	2999	0.00%	46.9/sec	243.9
/user/userli...	247	416	298	929	15	2941	0.40%	48.5/sec	240.0
TOTAL	1482	472	327	1151	3	2999	0.27%	238.7/sec	980.2

Include group name in label? Save Table Header

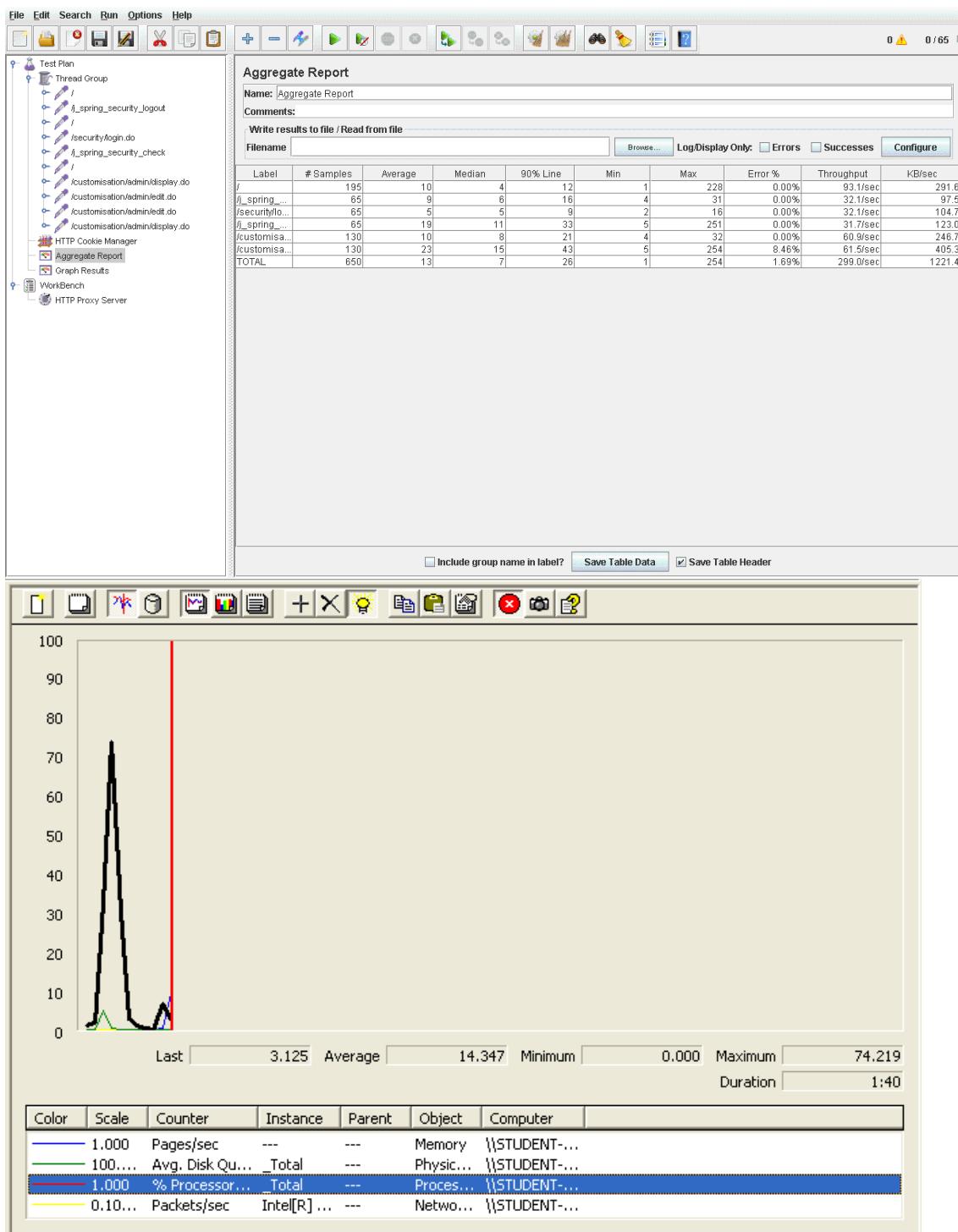
Usuarios máximos: 242

UC11: Ver y editar las Taboo Words (Acme-Newspaper 17.1)

En esta caso de uso hemos alcanzado tan sólo 60 usuarios concurrentes como máximo:



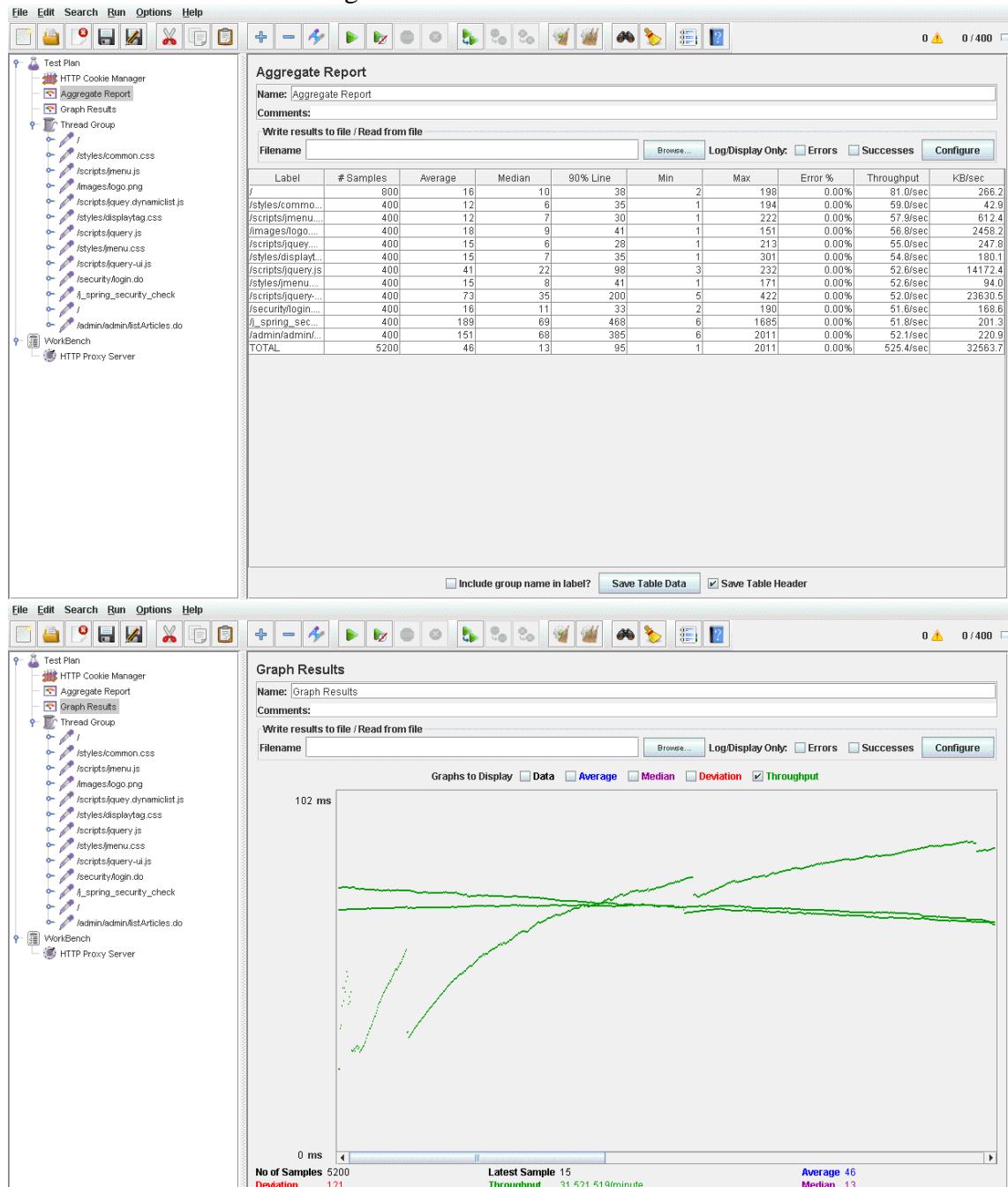
No hemos detectado un cuello de botella en este caso ni hemos conseguido sacar la causa a través del log de Jmeter. Pensamos que la causa puede deberse a como está realizado el Edit TabooWords ya que está hecho con javascript para poder añadir múltiples TabooWords en la misma vista y puede que esto genere algún tipo de problema con Jmeter. Igualmente no debe de ser un problema ya que este caso de uso es exclusivo de administradores y nuestro sistema tan solo tiene un administrador.



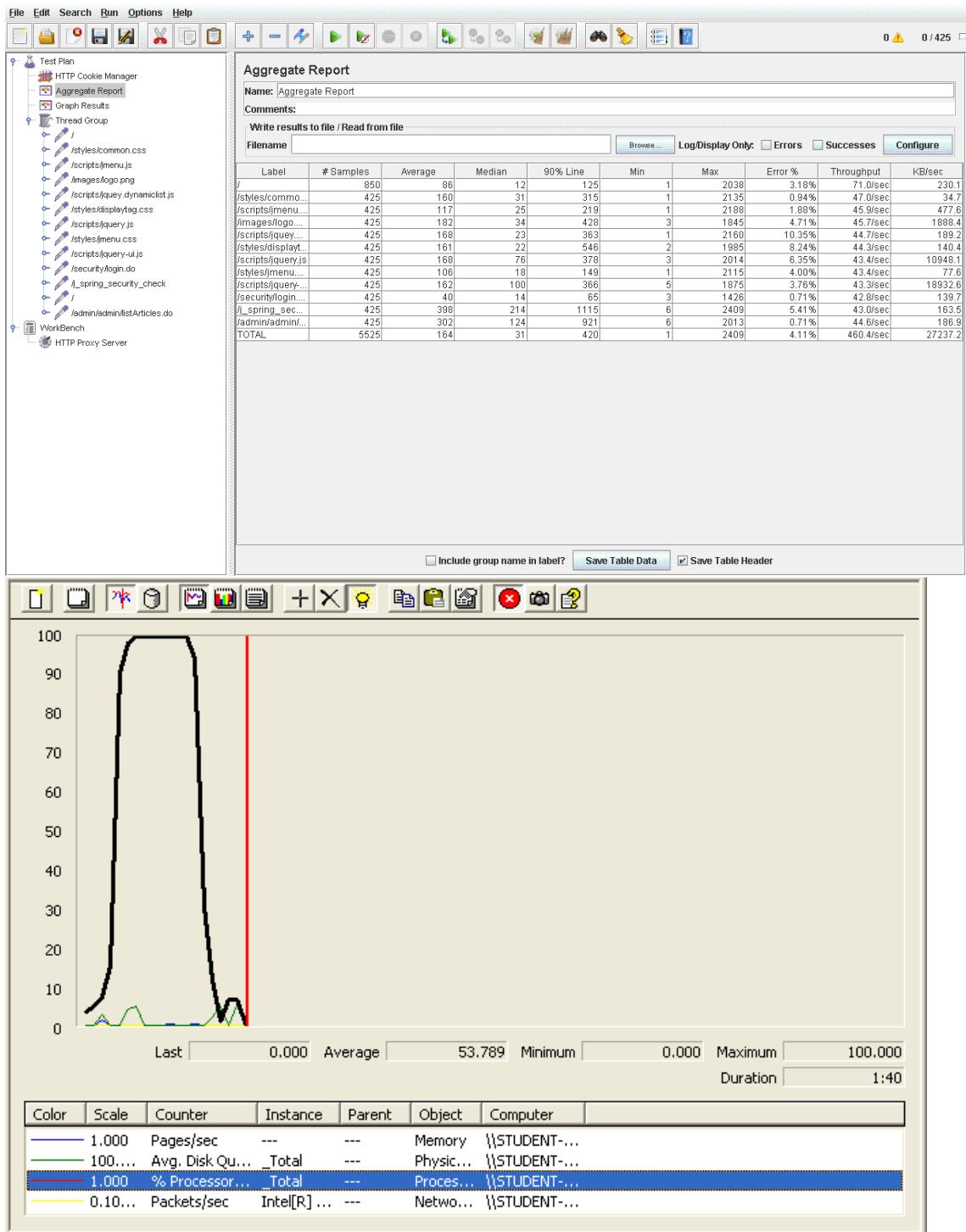
Máximo de usuarios: 60

UC12: Listar artículos Taboo (Acme-Newspaper 17.2)

En este caso de uso hemos llegado a los 400 usuarios concurrentes sin errores:



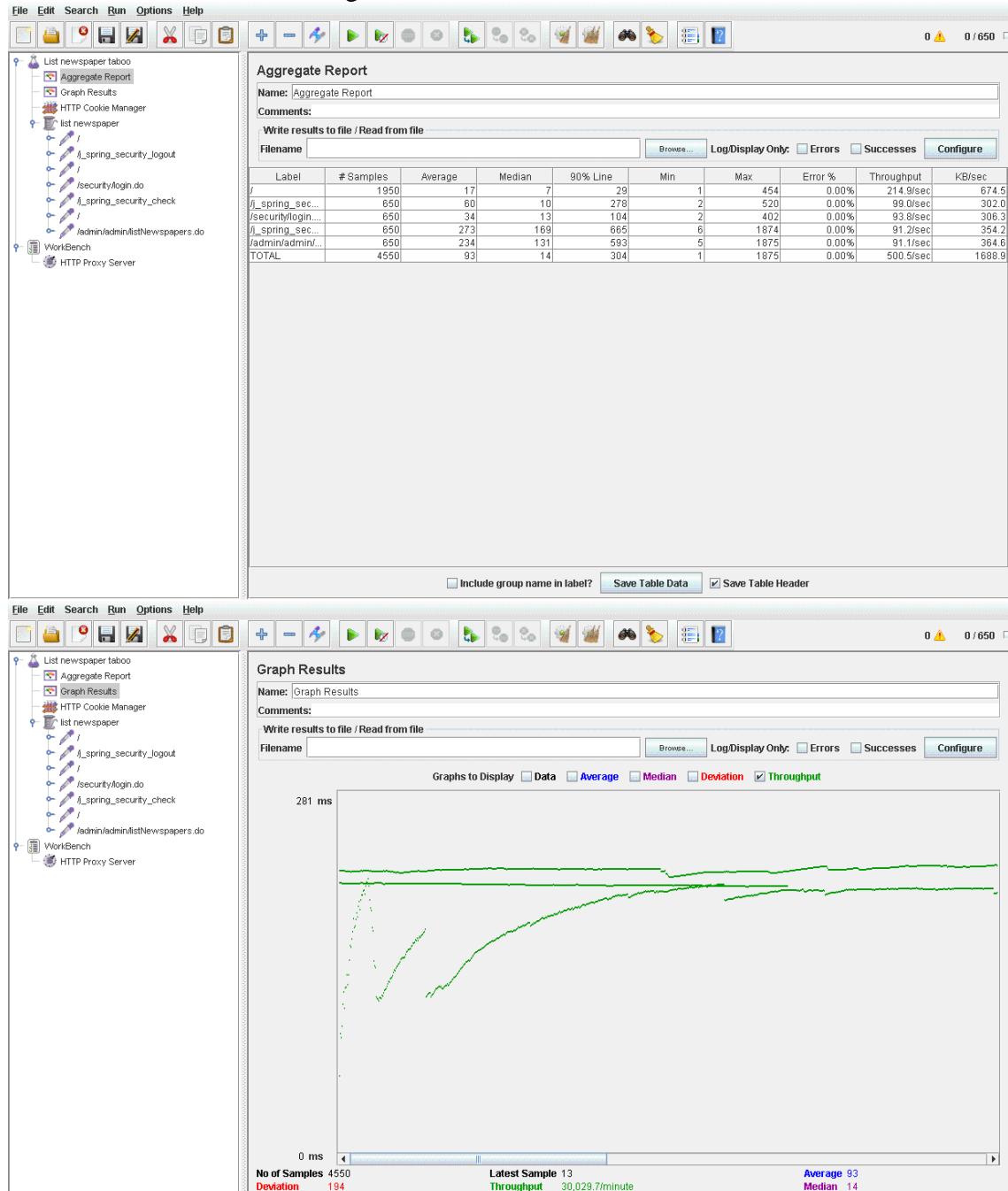
Sin embargo cuando llegamos a los 425 usuarios concurrentes ya comienzan a haber errores causados por la CPU, la cual produce un cuello de botella.



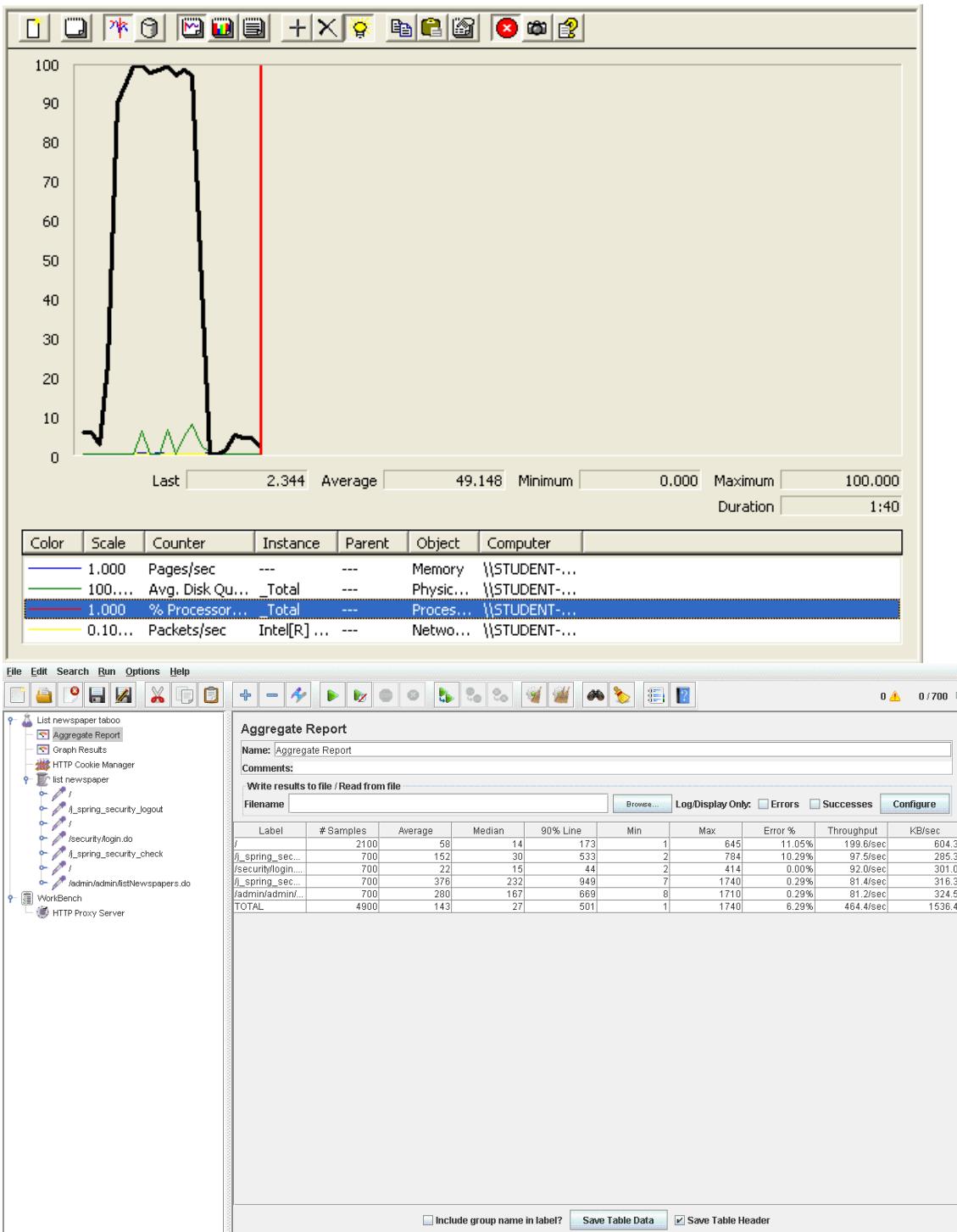
Usuarios máximo: 400

UC13: Listar newspapers Taboo (Acme-Newspaper 17.3)

En este caso de uso hemos llegado hasta los 650 usuarios concurrentes sin errores:



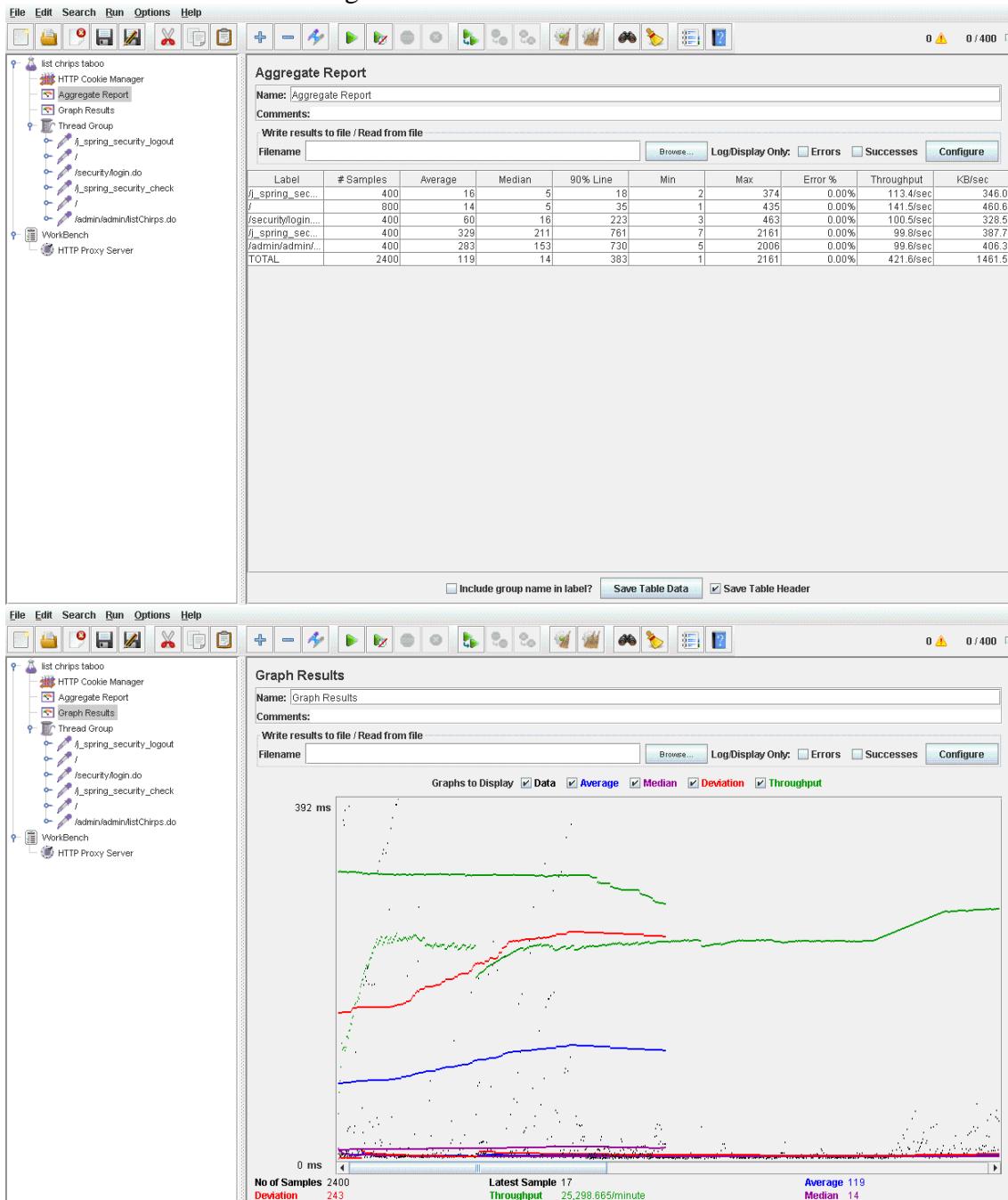
Sin embargo al llegar a 700 usuarios comienza a haber cuello de botella producido por la CPU:



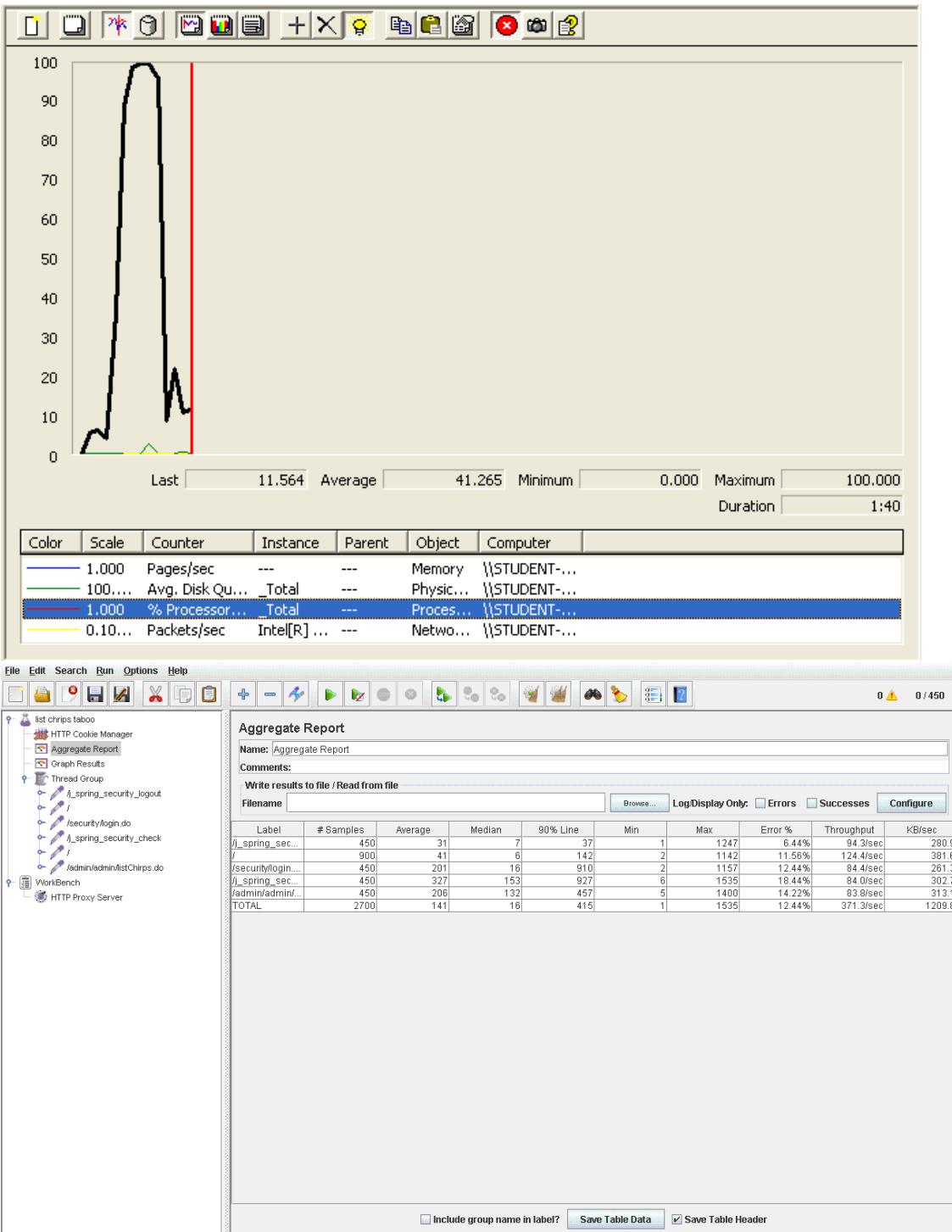
Usuarios máximo: 650

UC14: Listar Chirps Taboo (Acme-Newspaper 17.4)

En este caso de uso hemos llegado hasta los 400 usuarios concurrentes sin errores:



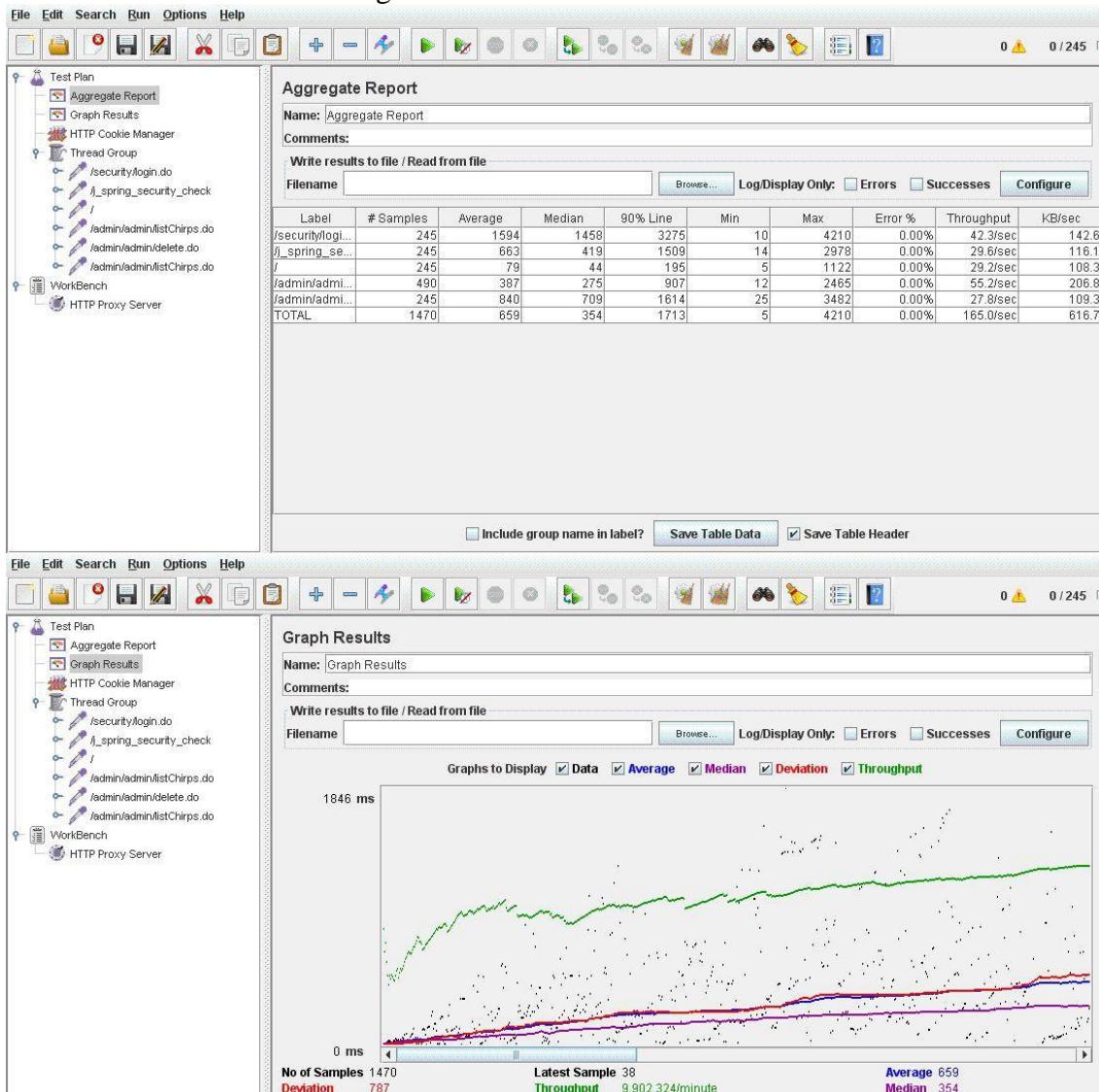
Sin embargo al llegar a 450 usuarios comienza a haber cuello de botella producido por la CPU:



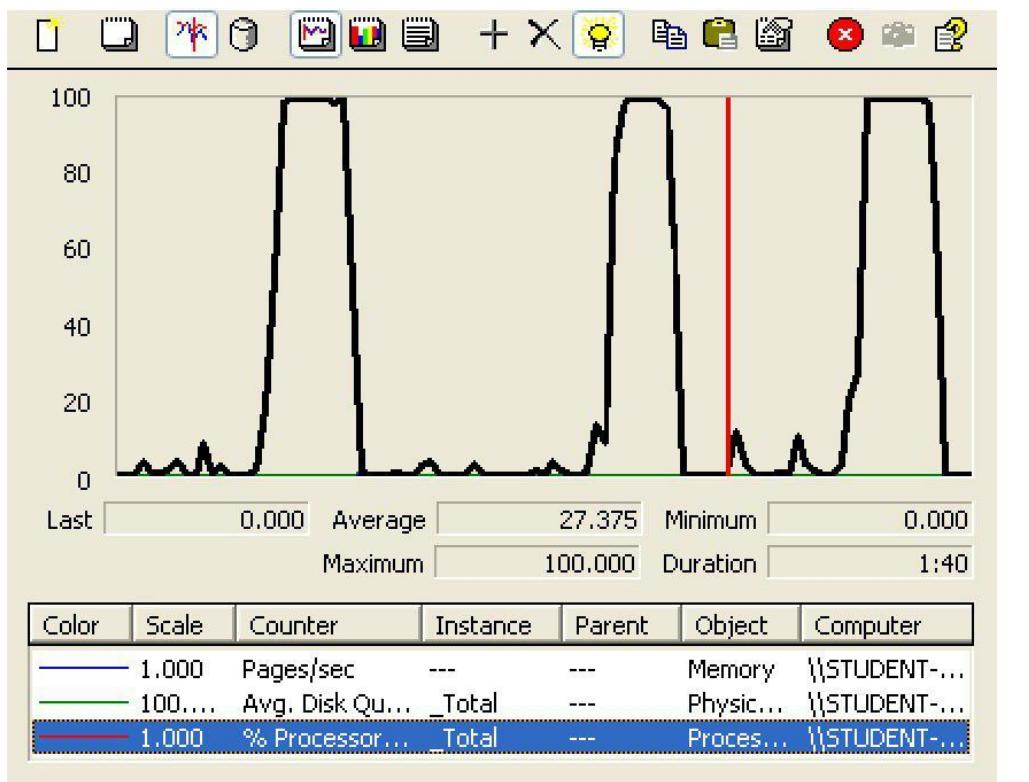
Usuarios máximo: 400

UC15: Delete Chirp (Acme-Newspaper 17.5)

En este caso de uso hemos llegado hasta los 245 usuarios concurrentes sin errores:



Sin embargo al llegar a 255 usuarios comienza a haber cuello de botella producido por la CPU:



The screenshot shows the JMeter Aggregate Report interface. The left sidebar displays the Test Plan structure, including a Thread Group with several requests like '/security/login.do' and '/admin/admin/listChirps.do'. The main area shows an 'Aggregate Report' table with the following data:

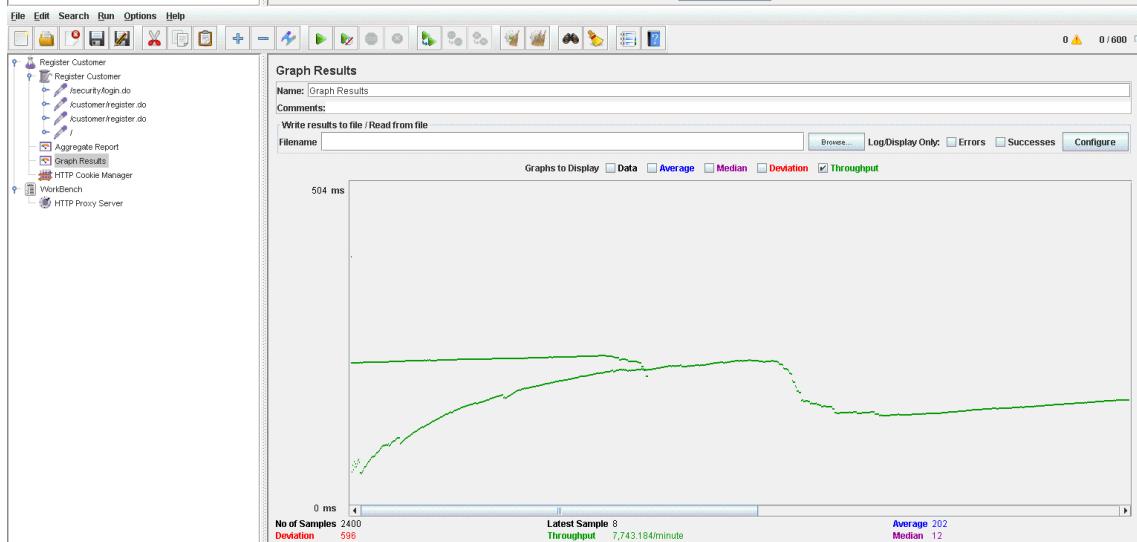
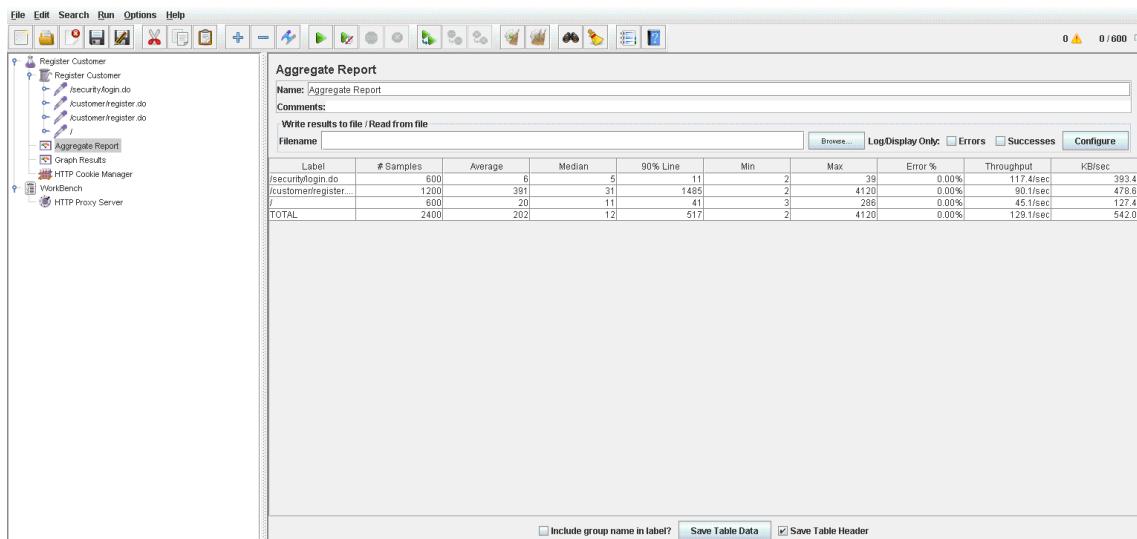
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login...	255	1589	1710	3506	6	4122	0.00%	49.8/sec	167.9
/j_spring_se...	255	801	665	1645	17	4025	0.39%	31.5/sec	123.5
/	255	220	164	525	5	939	0.00%	31.5/sec	116.7
/admin/admin...	510	679	441	1761	10	4211	0.20%	57.5/sec	214.9
/admin/admin...	255	1119	981	2182	23	4998	0.78%	28.9/sec	113.2
TOTAL_	1530	848	518	2316	5	4998	0.26%	171.6/sec	640.7

At the bottom, there are checkboxes for 'Include group name in label?', 'Save Table Data', and 'Save Table Header'.

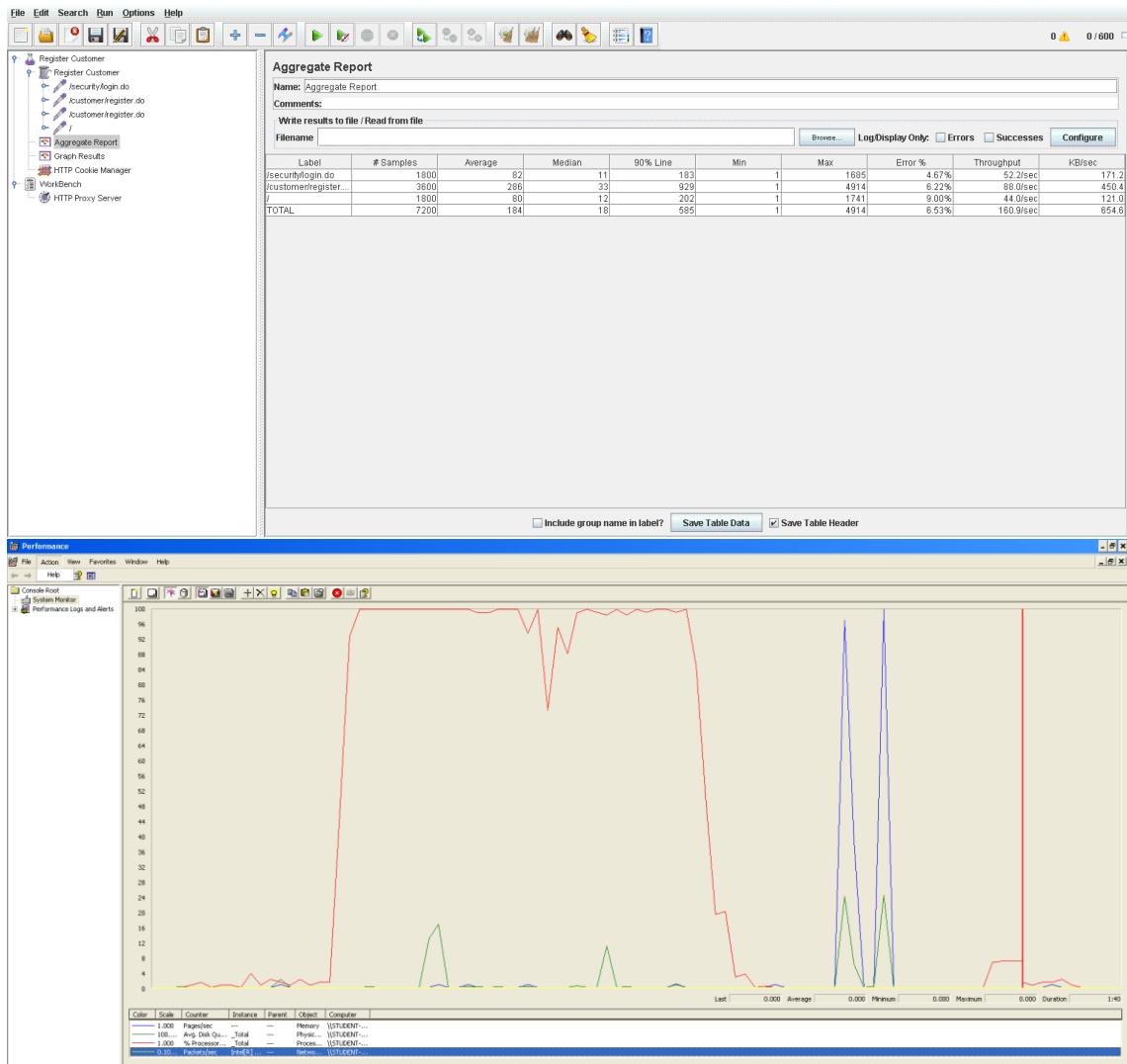
Usuarios máximo: 245

UC16: Registrar como Customer (Acme-Newspaper 21.1)

Este caso de uso es muy similar al registrar como Usuario por lo que nos hemos guiado por los resultados del anterior caso de Uso. Hemos ido probando usuarios hasta llegar a 600:



Con 600 usuarios ejecutando una vez el Script no parece haber problema, pero si cambiamos el Loop a 3 para que cada usuario ejecute el Script 3 veces vuelven a haber errores causados por el cuello de botella:

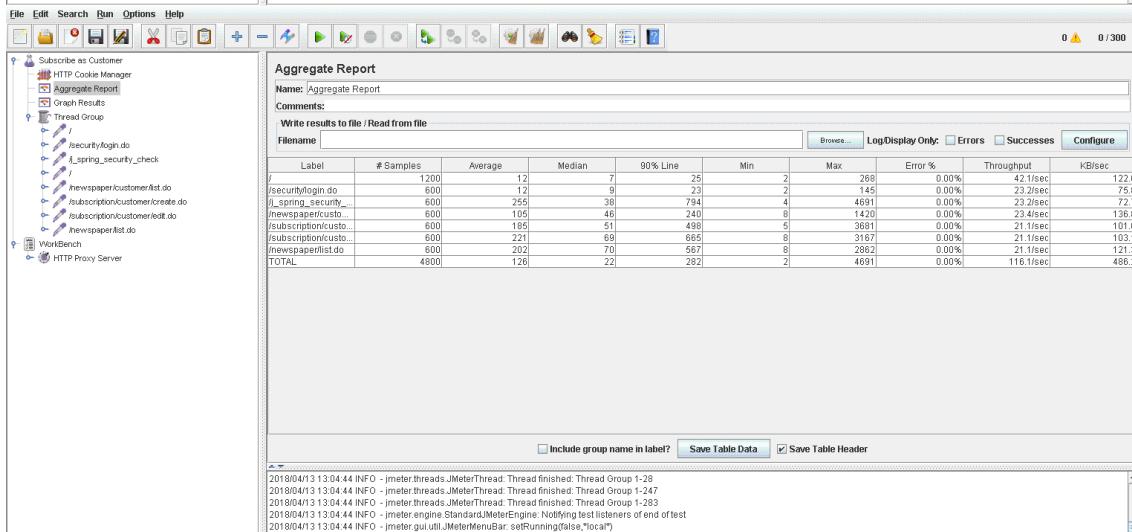
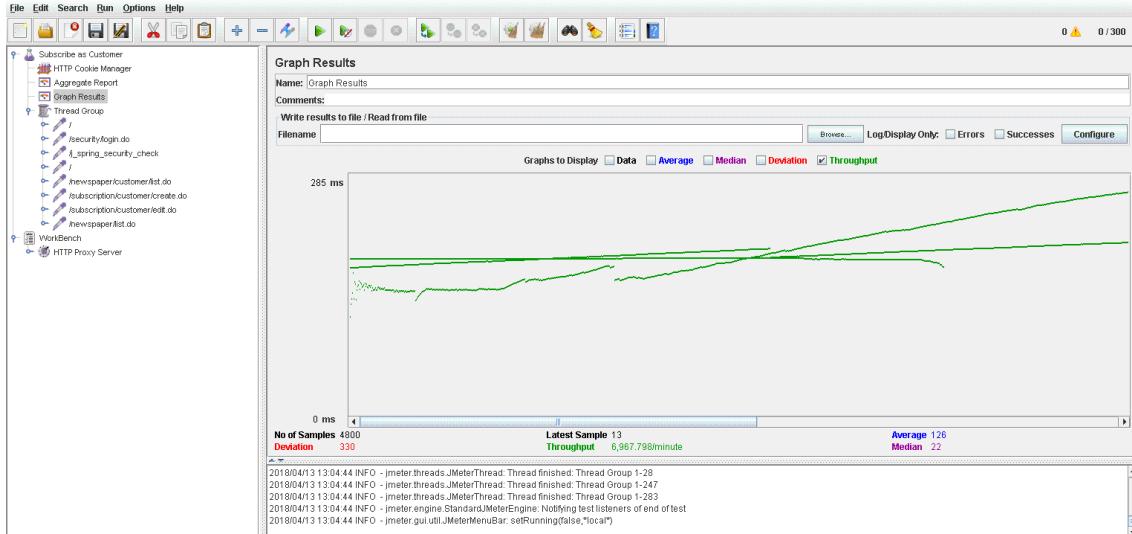


Parece ser que el causante principal es la CPU.

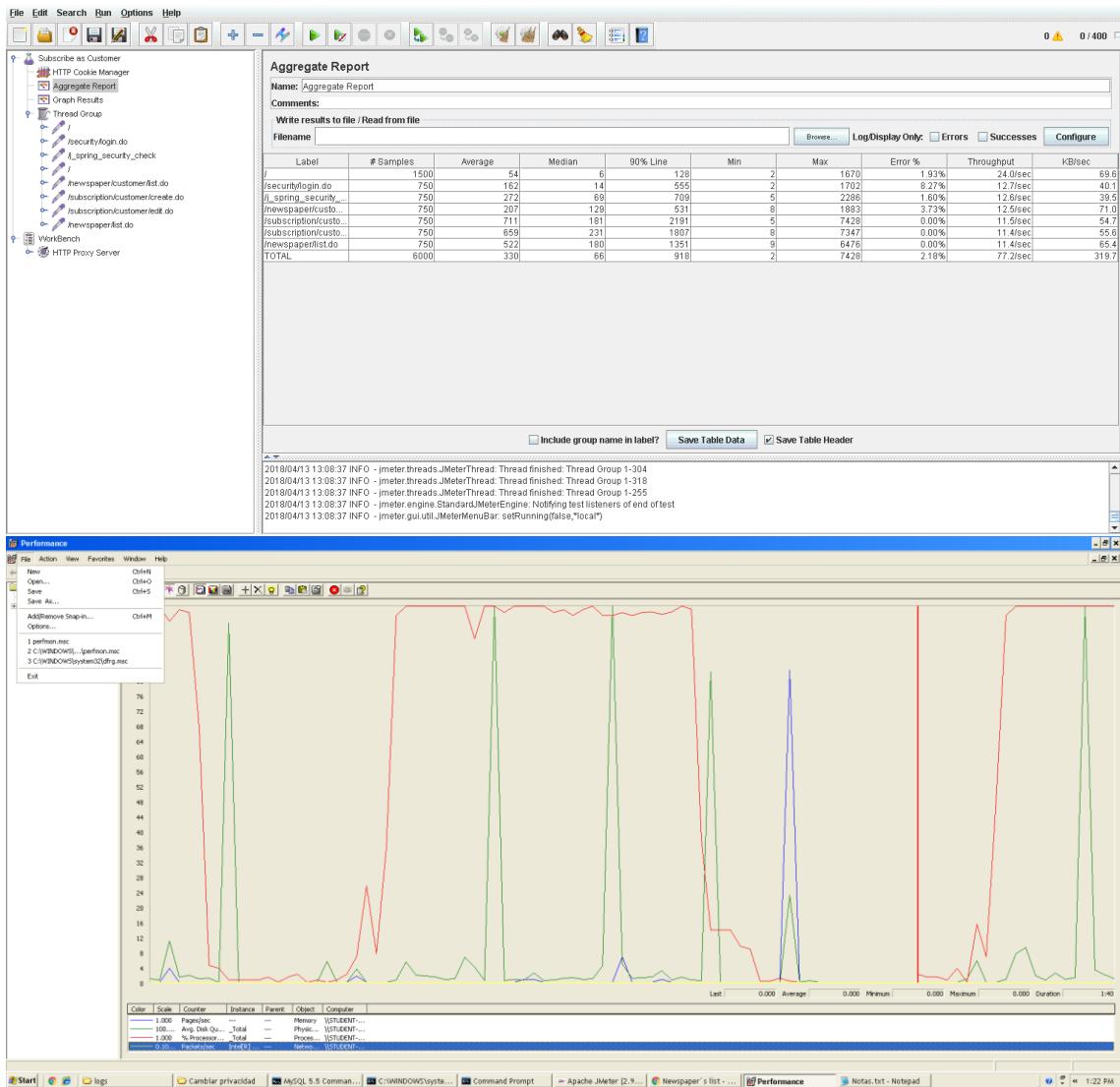
Usuarios máximos: 600

UC17: Suscribirse a un Newspaper como Customer (Acme-Newspaper 22.1)

En este caso de uso comenzamos a observar una bajada notable de rendimiento en el sistema. Cuando empezamos a llegar a los 300 usuarios en el sistema, aún no encontramos ningún problema incluso si cada usuario ejecuta el Script 2 veces (Loop = 2):



Pero si subimos el número de usuarios a 400 ya comienza a haber cuello de botella:

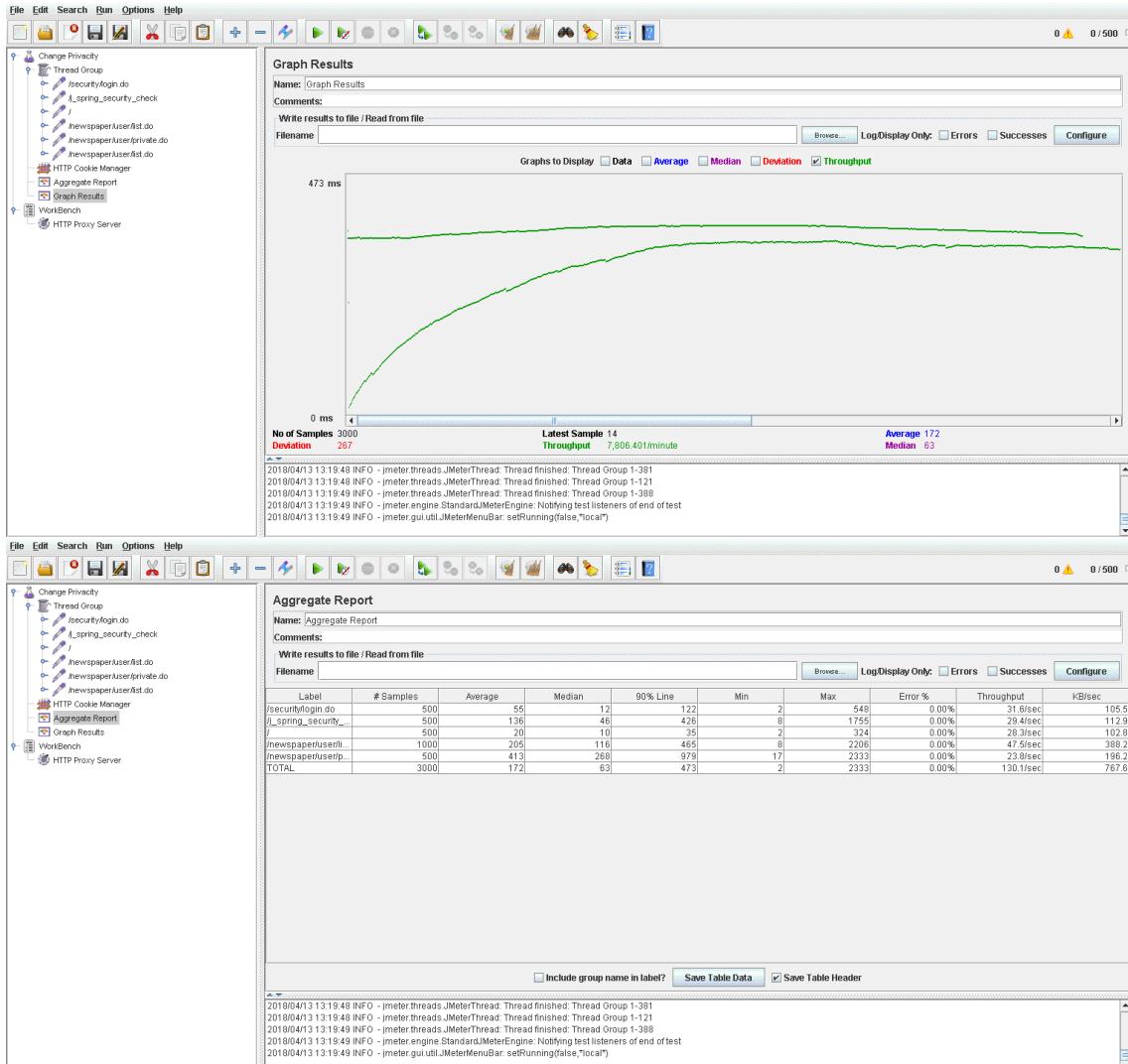


Vemos que efectivamente la CPU está la mayor parte del tiempo usándose un 100% provocando picos del disco duro.

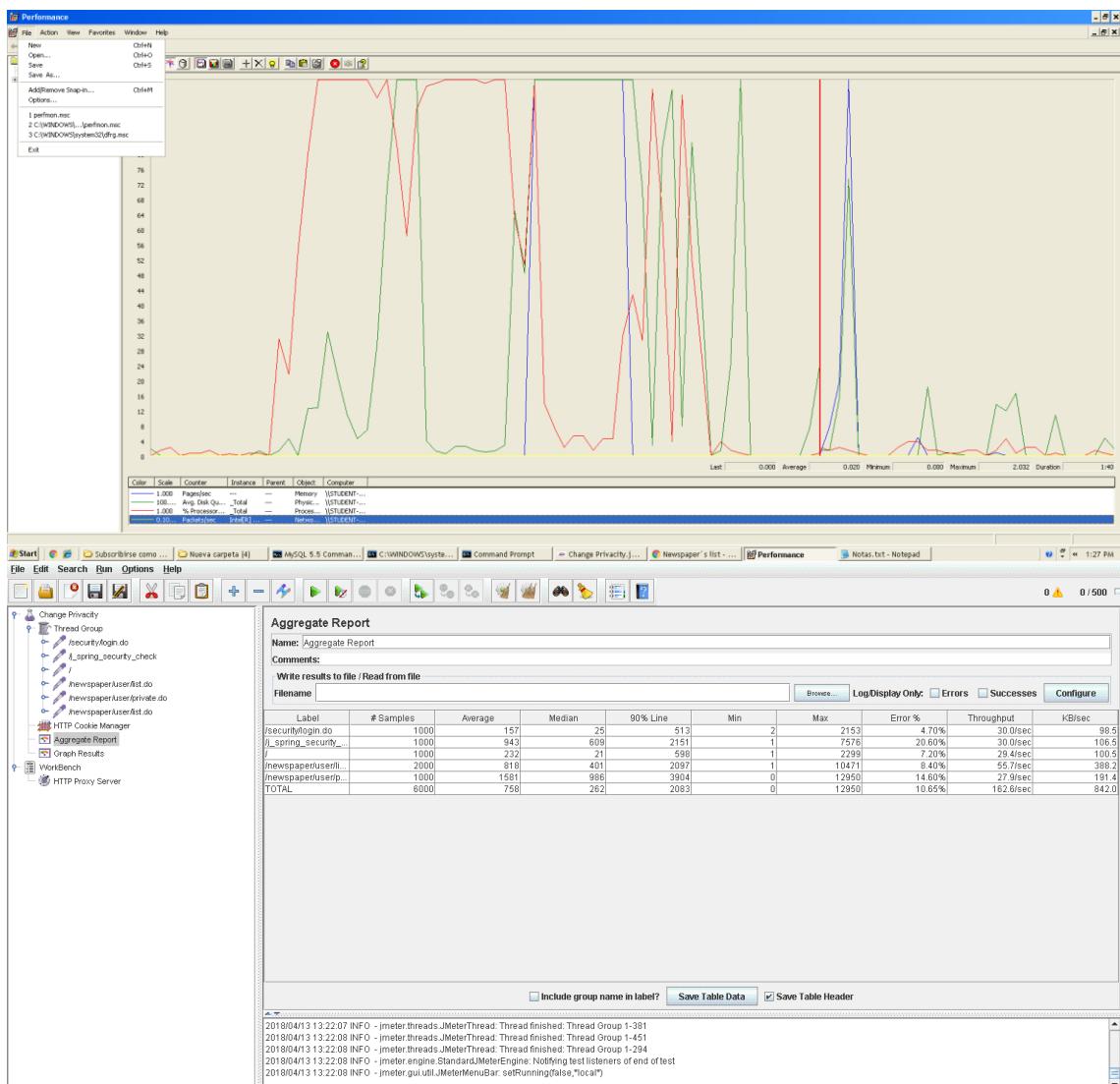
Usuarios máximos: 300

UC18: Cambiar privacidad de newspaper (Acme-Newspaper 23.1)

Comenzamos aumentando el número de usuarios de la misma manera de siempre, hasta que llegamos a 500 usuarios ejecutando una sola vez el script, obteniendo estos resultados:



No parece haber problema en este caso, pero si lo intentamos con 500 usuarios ejecutando 2 veces el script cada uno, comenzamos a observar errores debido a un cuello de botella:

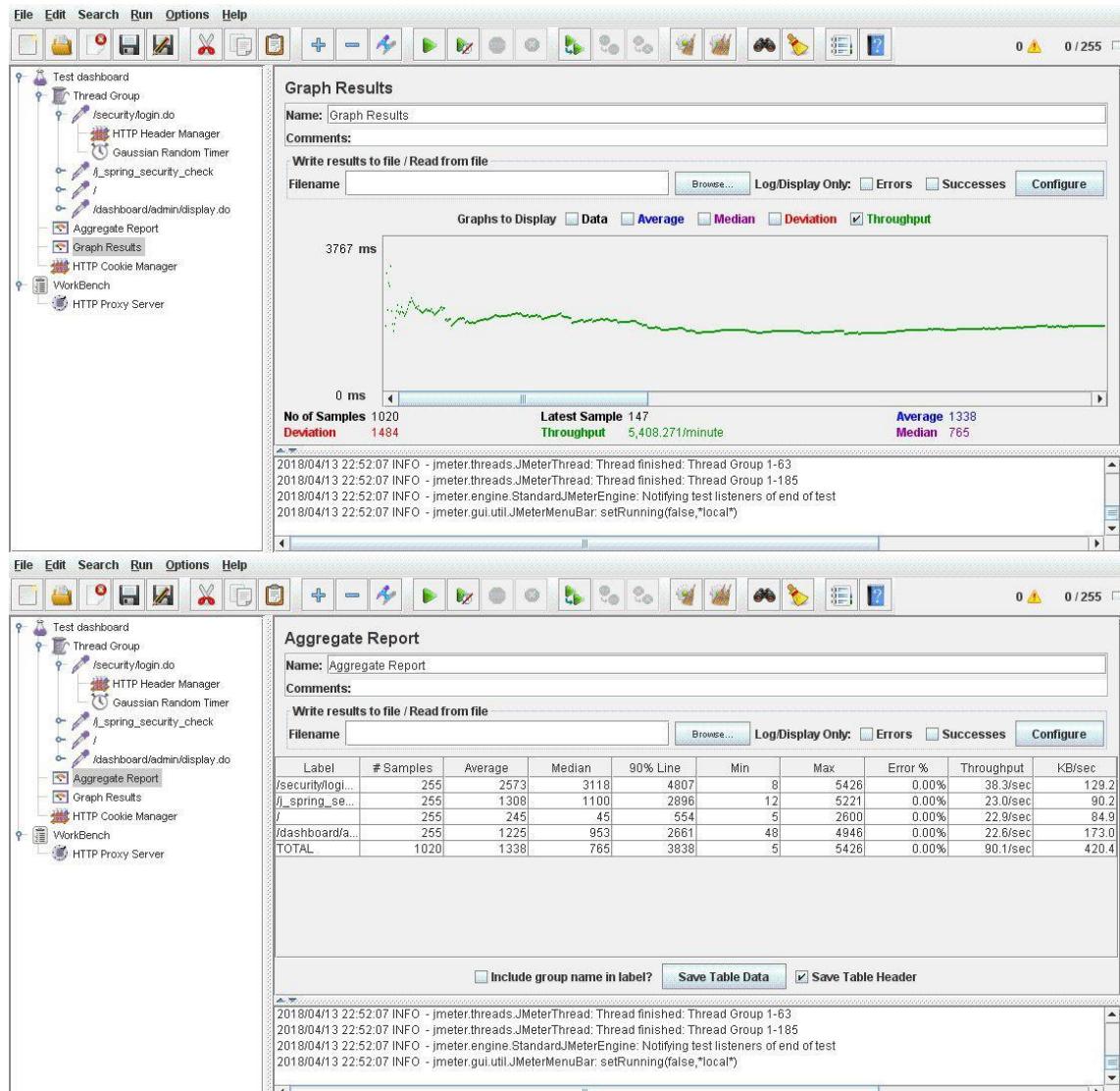


La causa parece ser la CPU y el Disco duro.

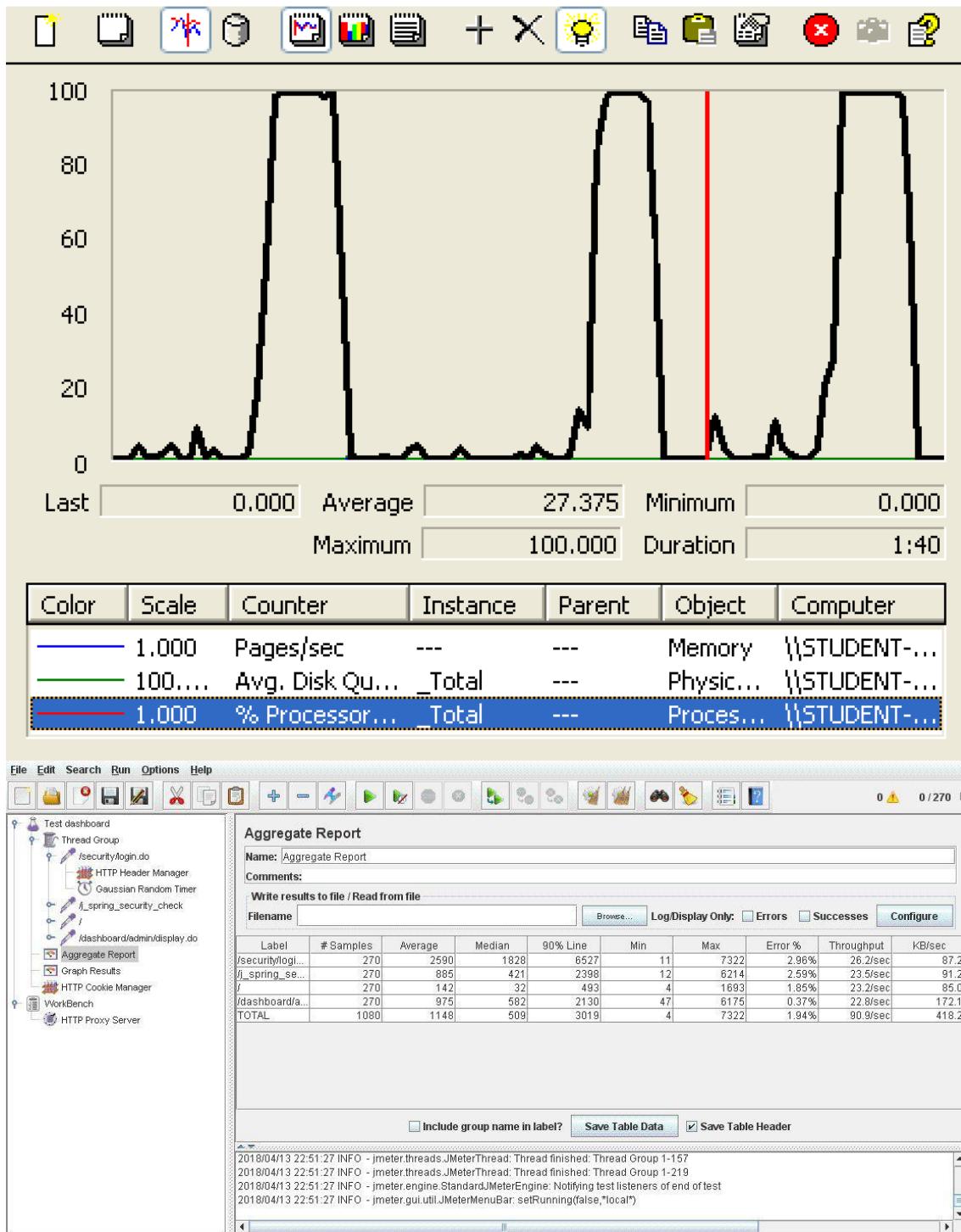
Número máximo de Usuarios: 500

UC19: Dashboard

Para este caso de uso hemos obtenido 255 usuarios concurrentes sin errores:



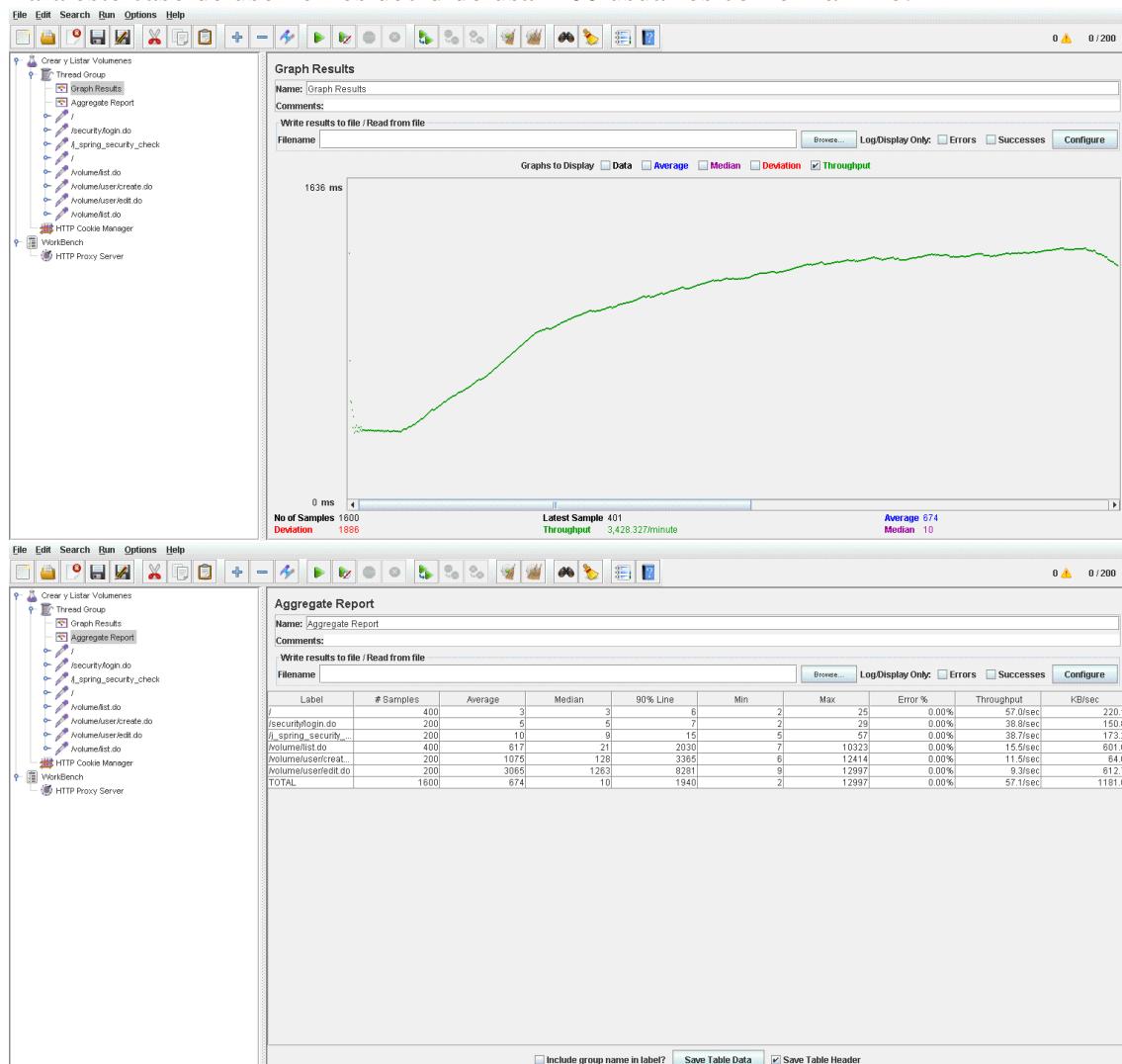
Sin embargo a partir de 255 usuarios comienza a haber cuello de botella producido por la CPU:



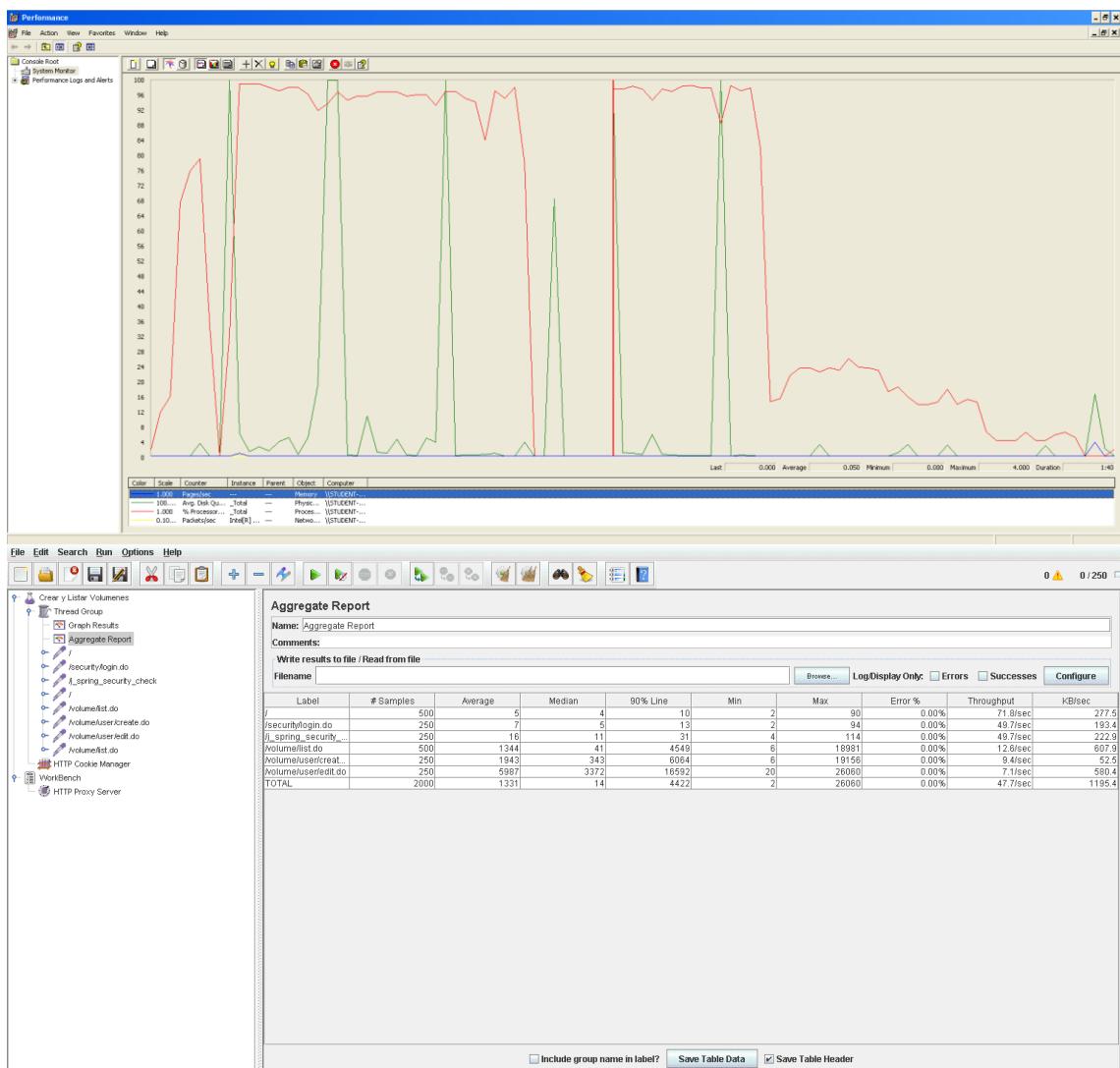
Usuarios máximo: 255

UC20: List and Create Volumes

Para este caso de uso hemos decidido usar 200 usuarios como máximo:



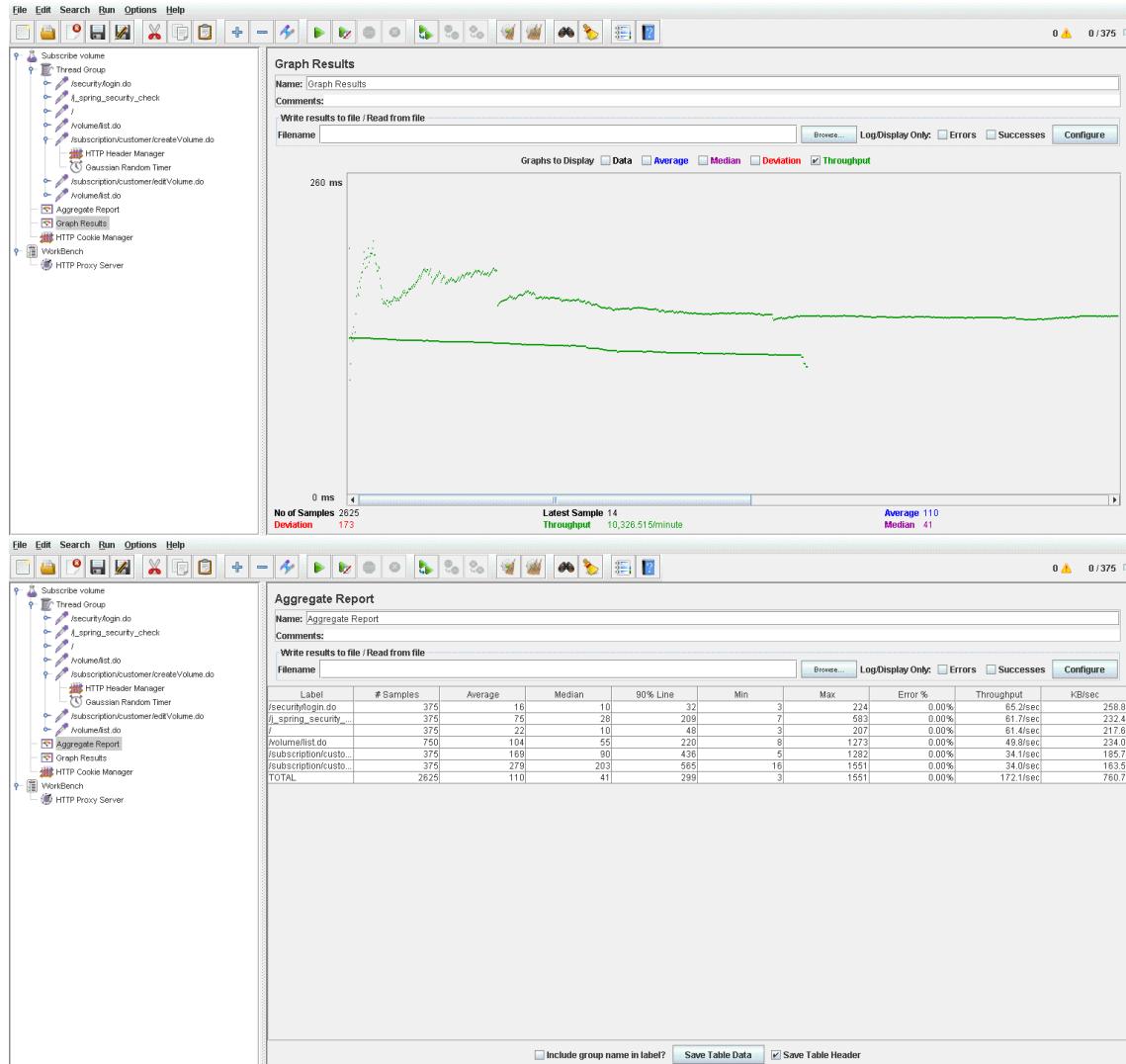
Ya que en este caso de uso no hemos obtenido errores al aumentar un poco más la cantidad de usuarios, pero sí una notable ralentización del sistema debido a un cuello de botella producido por la CPU y el disco duro:



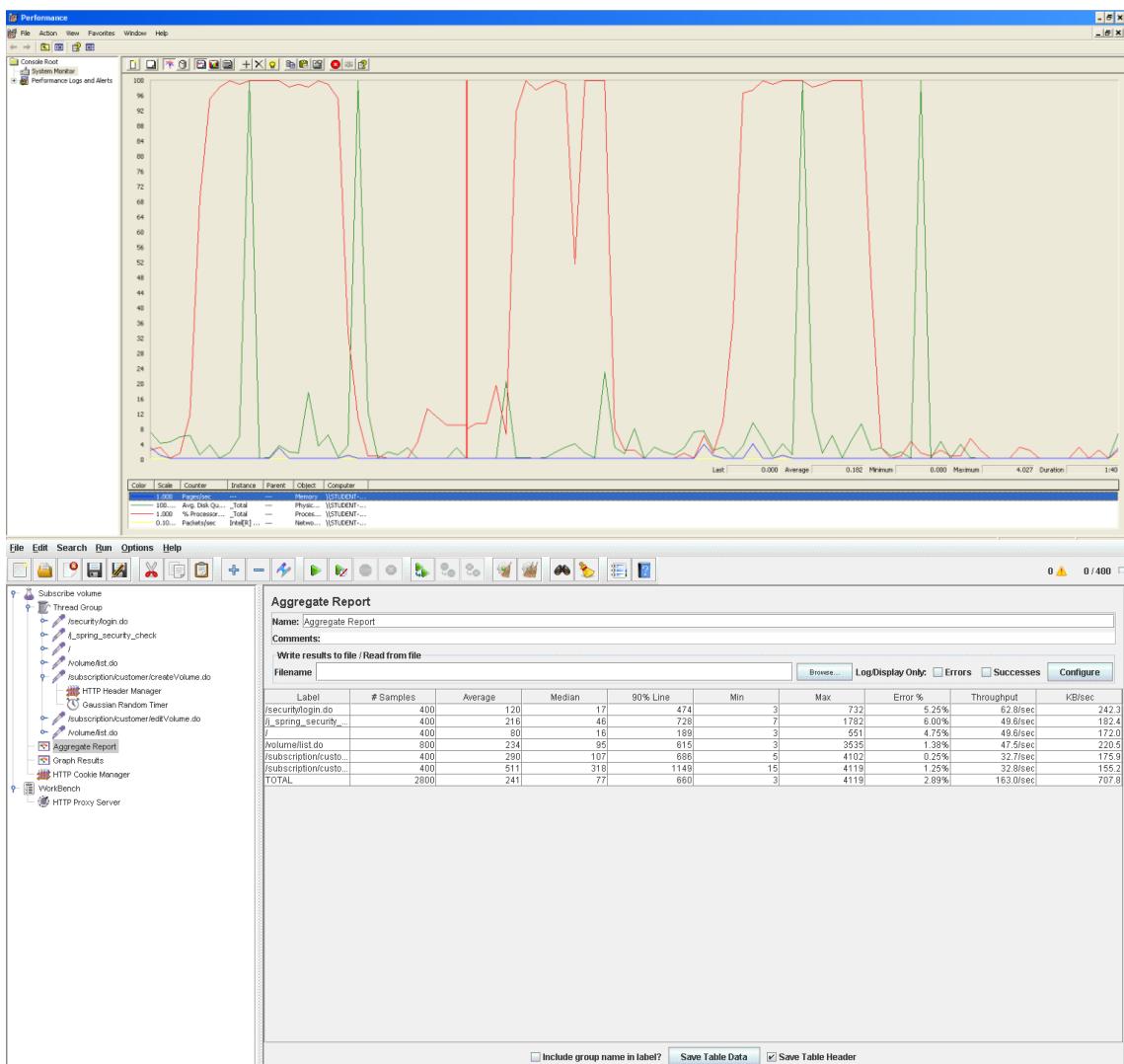
Usuarios máximo: 200

UC21: Subscribe to a Volume

Para este caso de uso hemos obtenido 375 usuarios concurrentes sin errores:



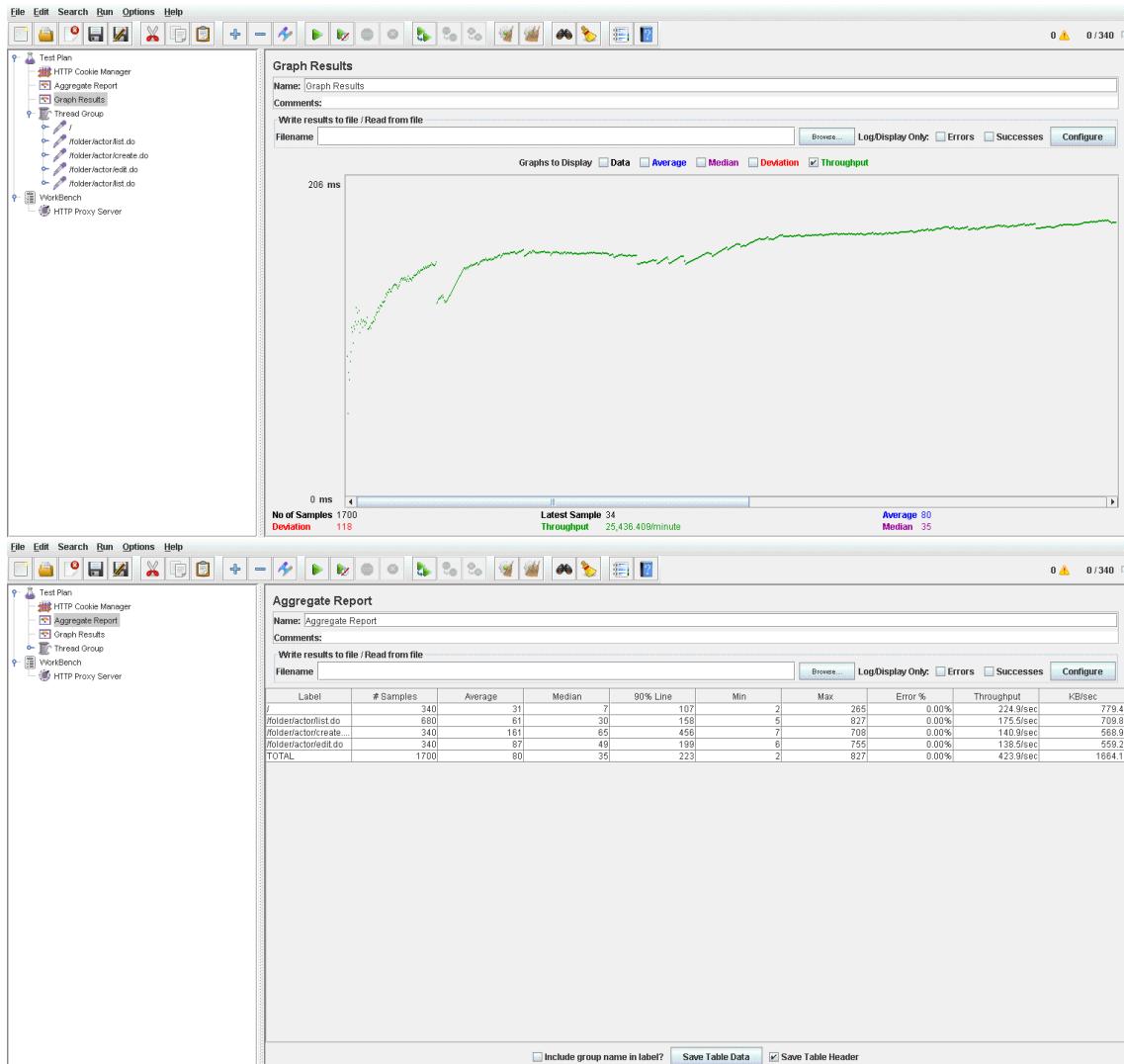
Sin embargo a partir de 400 usuarios comienza a haber cuello de botella producido por la CPU y el disco duro:



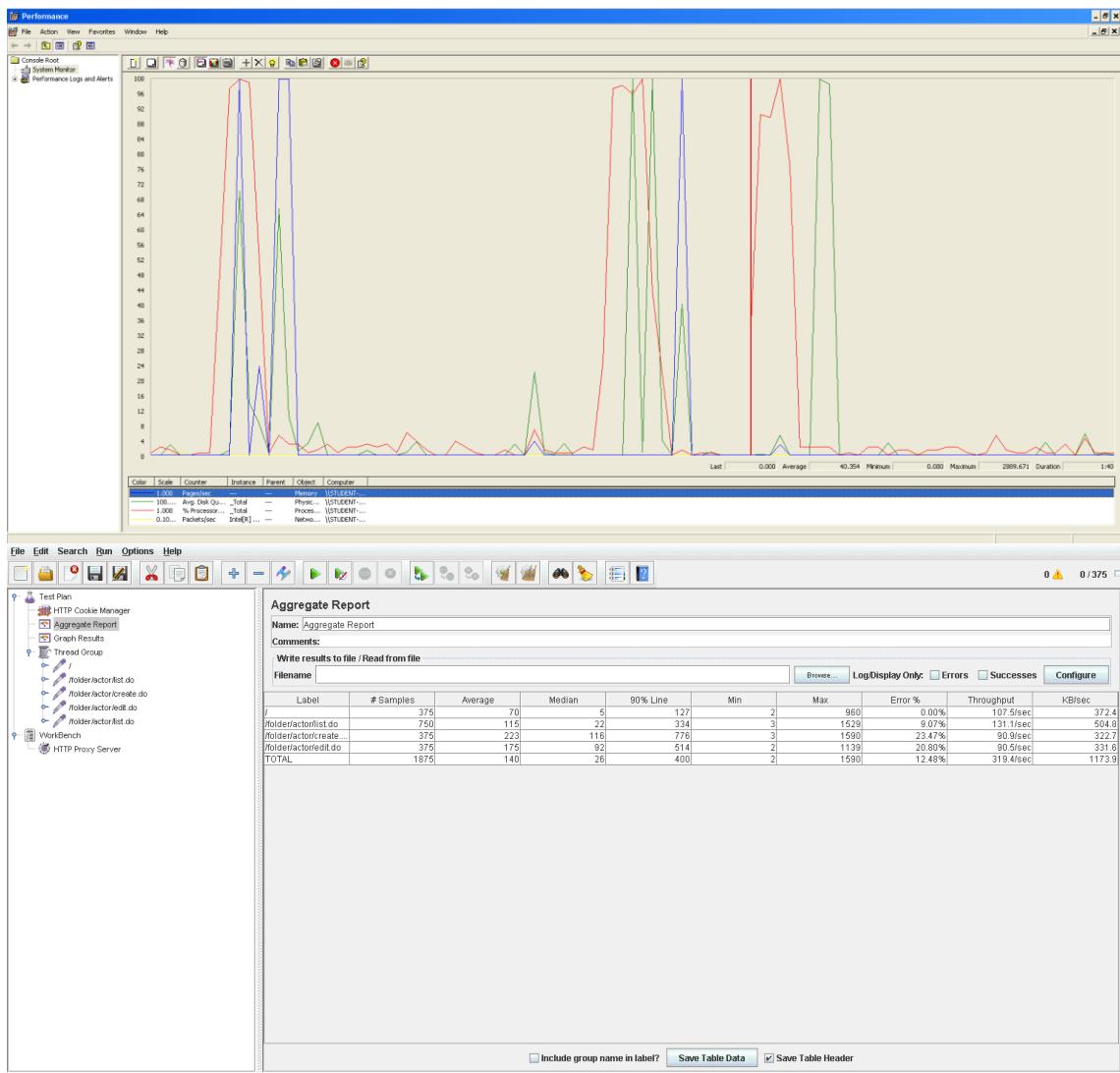
Usuarios máximo: 375

UC22: Crear Folders

Para este caso de uso hemos obtenido 340 usuarios concurrentes sin errores:



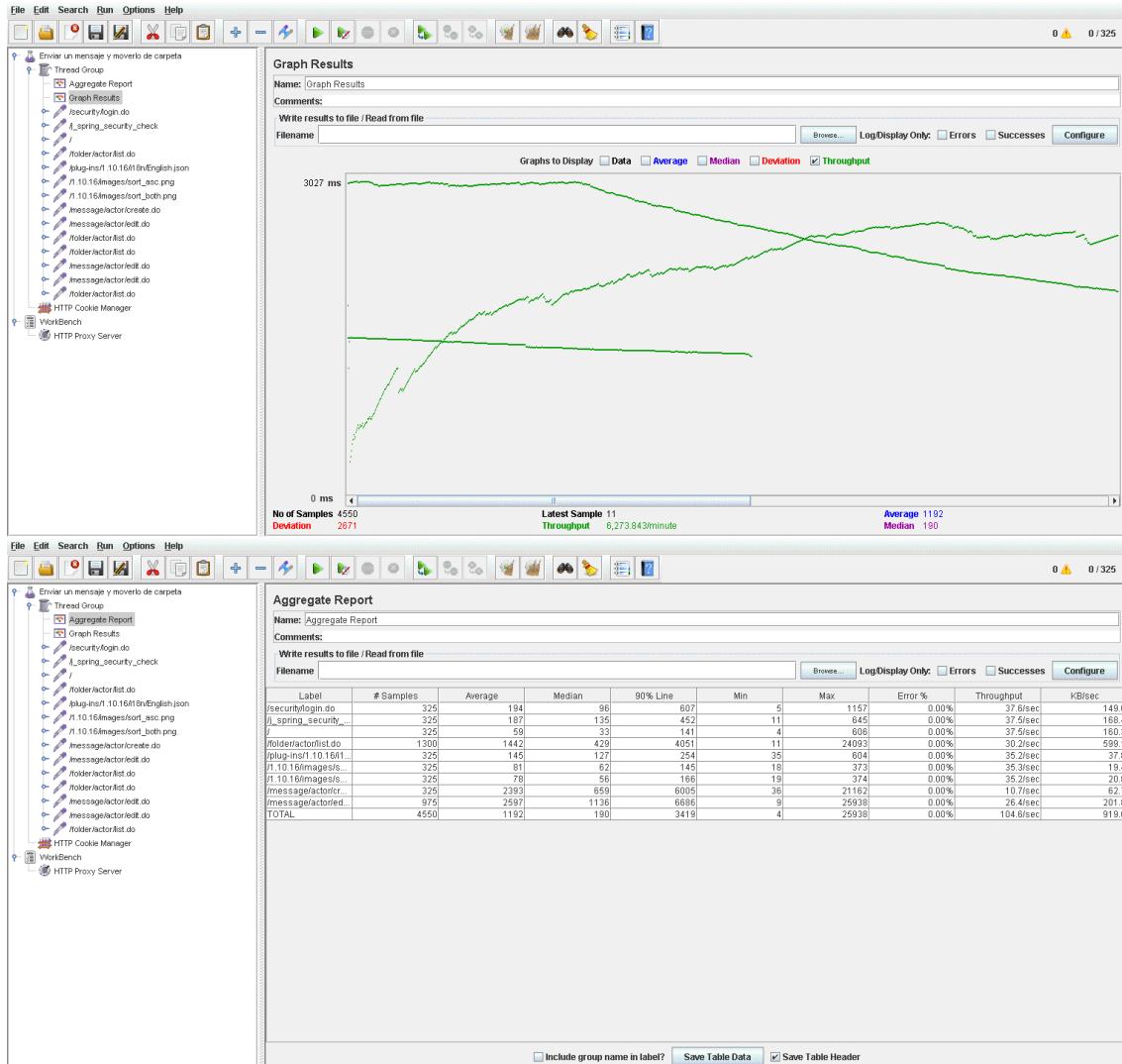
Sin embargo a partir de 375 usuarios comienza a haber cuello de botella producido por la RAM y el Disco duro:



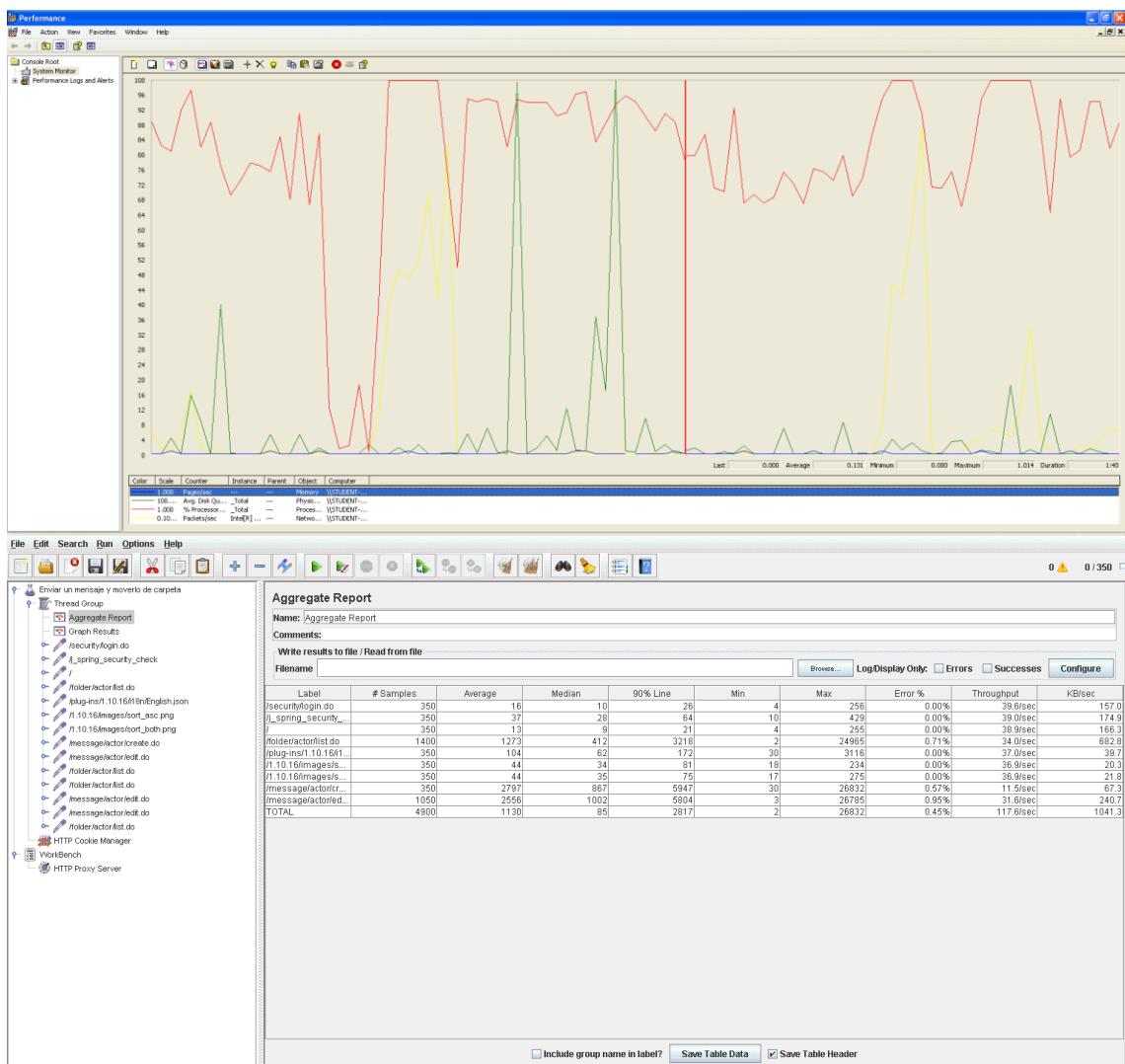
Usuarios máximo: 340

UC23: Crear mensajes y moverlos de carpeta

Para este caso de uso hemos obtenido 325 usuarios concurrentes sin errores:



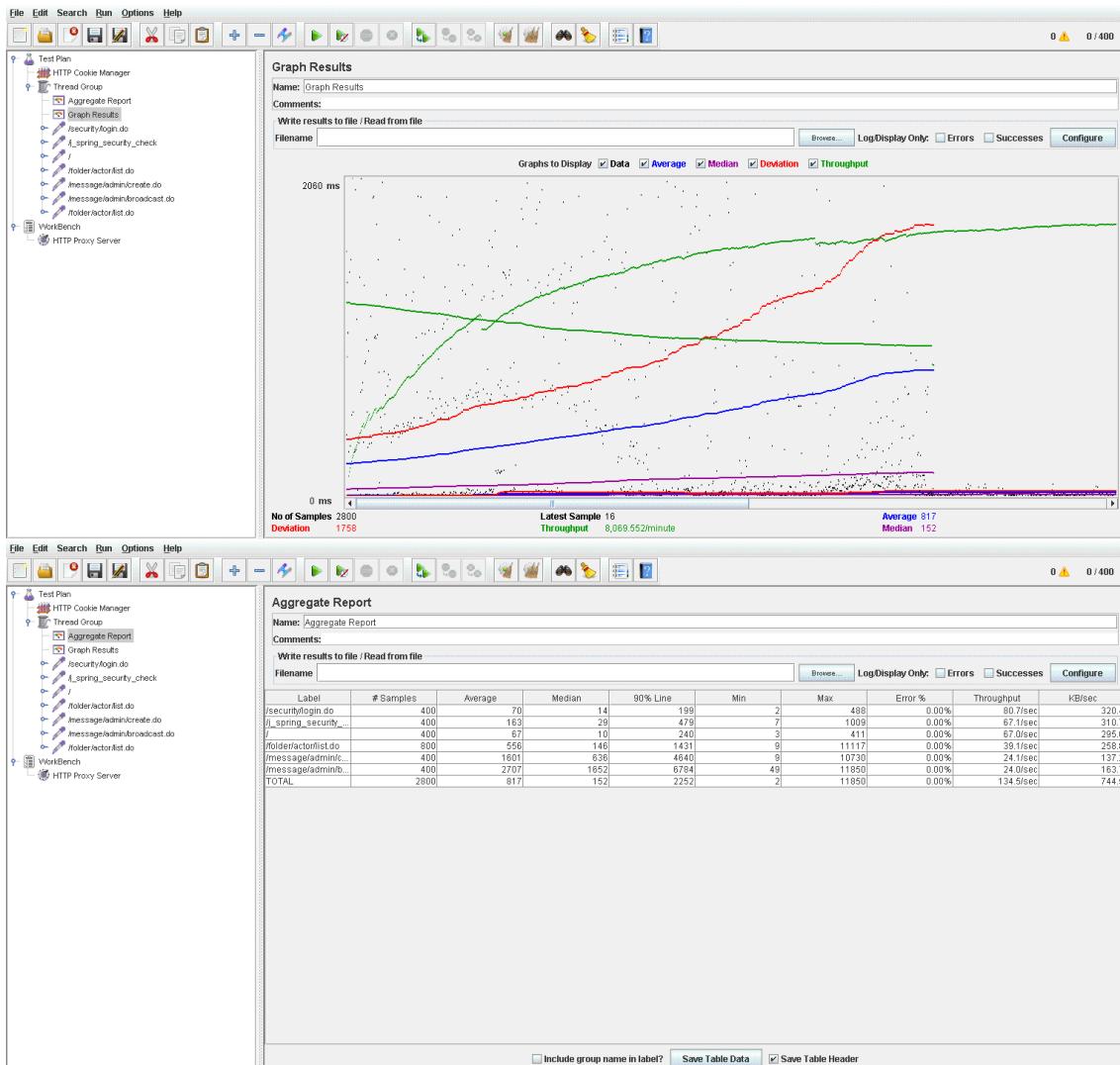
Sin embargo a partir de 350 usuarios comienza a haber cuello de botella producido por la CPU y el disco duro:



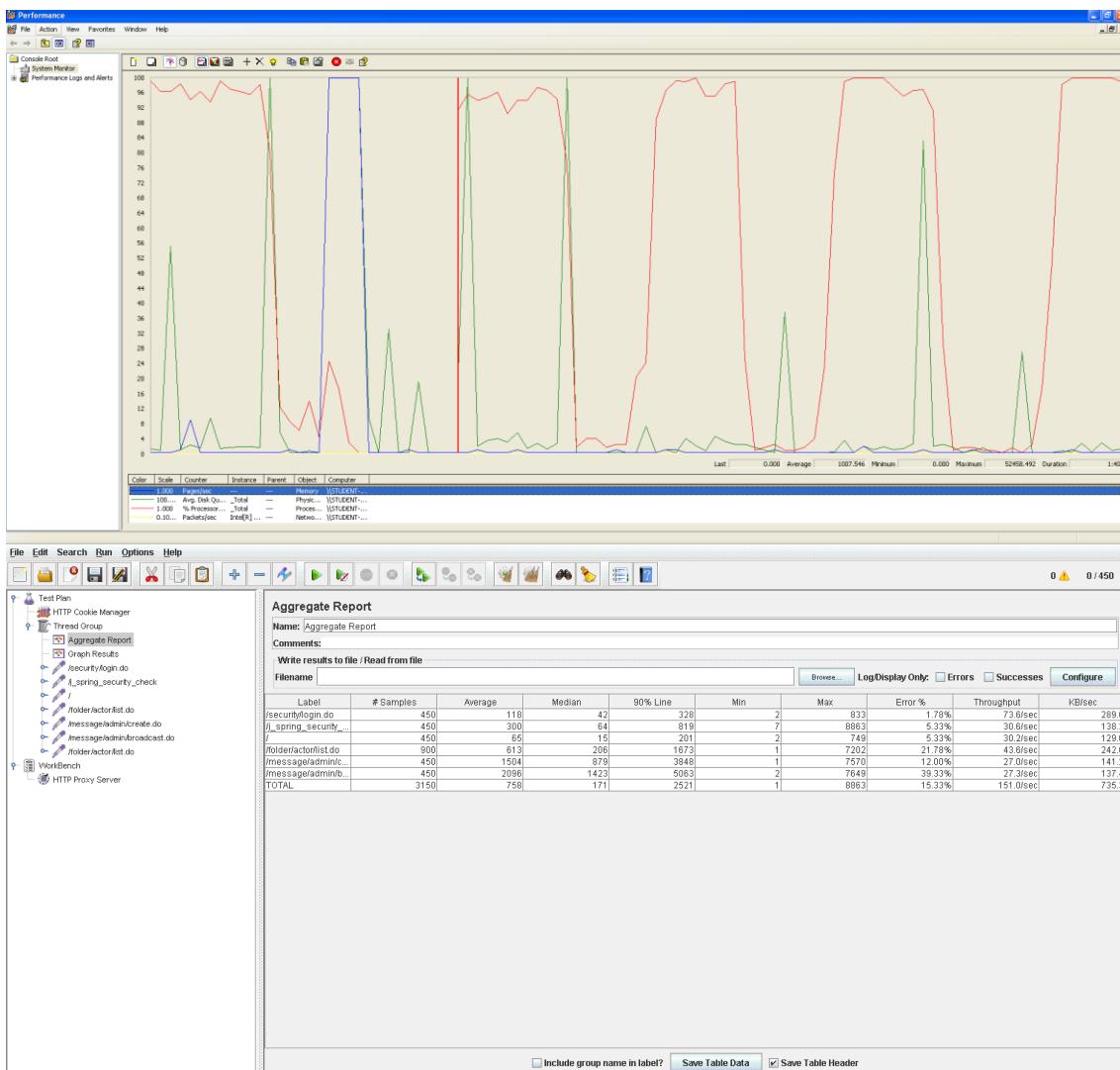
Usuarios máximo: 325

UC24: Enviar un mensaje a todos los usuarios como admin

Para este caso de uso hemos obtenido 400 usuarios concurrentes sin errores:



Sin embargo a partir de 450 usuarios comienza a haber cuello de botella producido por la CPU, el disco duro y la memoria RAM:



Usuarios máximo: 400

UC25: Registrarse en el sistema como agente (Newspaper 2.0 – C)

Para este caso de uso no iniciaremos sesión y haremos el registro de un nuevo agente.

Para este caso de uso se han registrado hasta 180 usuarios concurrentes sin observar ningún problema.

Screenshot of JMeter Aggregate Report results:

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
/	1080	167	36	433	4	2254	0.00%	23.7/sec	83.3
/scripts/jqu...	360	94	34	274	2	1174	0.00%	9.3/sec	41.8
/scripts/jqu...	360	206	122	525	3	2019	0.00%	9.3/sec	2500.6
/styles/dis...	360	97	14	390	2	1226	0.00%	9.3/sec	30.6
/scripts/jm...	360	99	27	276	2	1048	0.00%	9.3/sec	98.5
/styles/co...	360	103	34	297	2	1233	0.00%	9.4/sec	7.9
/styles/ime...	360	104	30	318	2	1131	0.00%	9.4/sec	16.8
/scripts/jqu...	360	217	143	508	5	1642	0.00%	9.4/sec	4274.7
/favicon.ico	360	114	58	287	2	892	0.00%	9.5/sec	307.7
/security/lo...	720	265	167	676	6	2223	0.00%	16.4/sec	63.9
/agent/regi...	720	1779	364	5809	6	18621	0.00%	19.0/sec	111.2
/j_spring...	360	2248	717	7335	12	15965	0.00%	9.8/sec	38.2
TOTAL	5760	492	71	819	2	18621	0.00%	126.5/sec	6412.1

Include group name in label? Save Table Header

Screenshot of JMeter Graph Results visualization:

Graph Results

Name: Graph Results
Comments:

Write results to file / Read from file

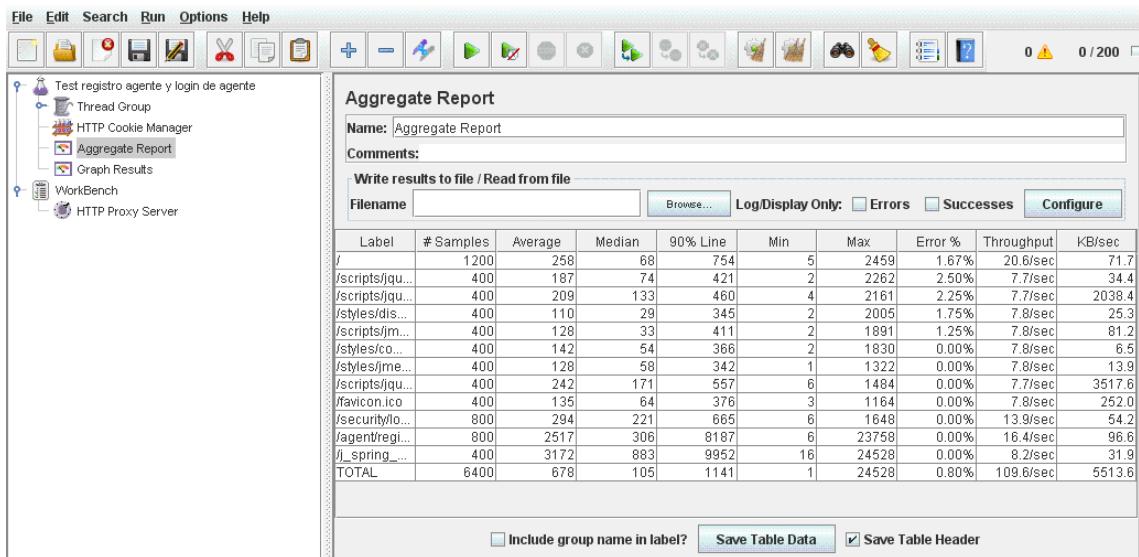
Graphs to Display: Data Average Median Deviation Throughput

878 ms

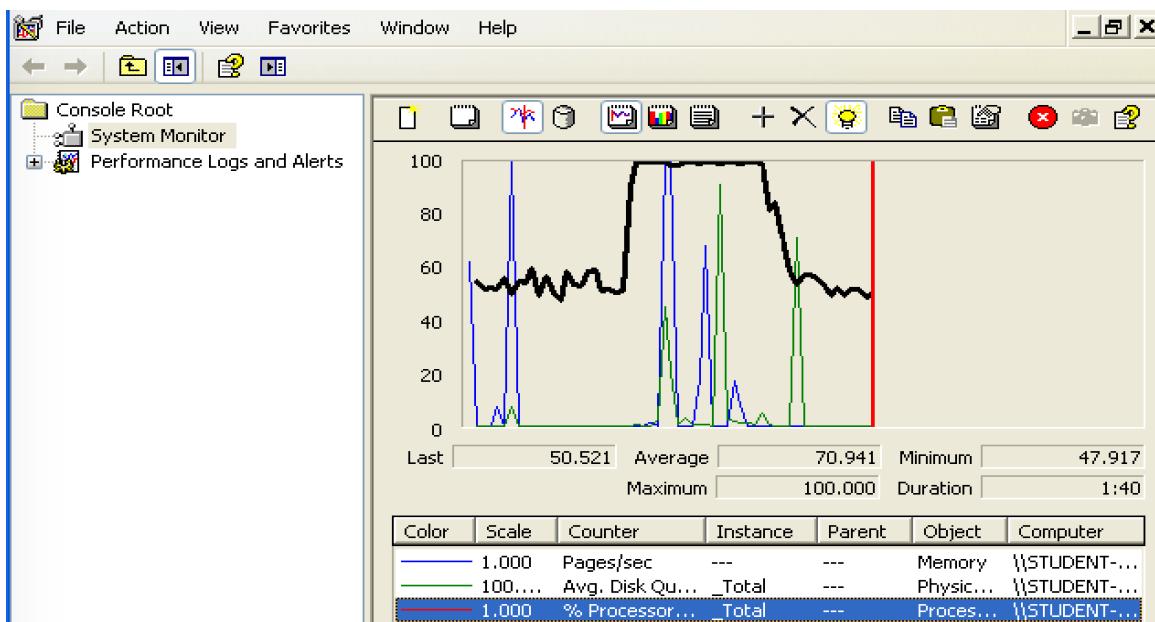
0 ms

No of Samples: 5760
Deviation: 1523
Latest Sample: 13
Throughput: 7,589.599/minute
Average: 492
Median: 71

Sin embargo, a partir de 200 usuarios hemos podido ver que comenzaba a no funcionar correctamente.



El motivo de ello es una carga de trabajo para la CPU demasiado alta.



Usuarios máximos: 180

UC26: Listar los periódicos donde un agente ha colocado anuncios y crear un anuncio (Newspaper 2.0 – C)

Para este caso de uso iniciaremos sesión como agente, iremos a listar los periódicos que contienen anuncios de un agente y crearemos un nuevo anuncio, lo que nos devolverá a la misma vista de periódicos que contienen anuncios de un agente.

El siguiente caso de uso ha soportado un número de usuarios concurrentes sin problemas de 70.

Screenshot of JMeter Aggregate Report results:

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
/	140	46	21	90	5	400	0.0%	72.8/sec	260.3
/security/login.do	70	104	61	269	8	520	0.0%	42.0/sec	163.0
/j_spring_security_check	70	124	80	296	12	463	0.0%	37.8/sec	147.1
/advertisement/agent/listPlacedAds.do	140	168	117	395	25	1113	0.0%	28.5/sec	172.3
/advertisement/agent/create.do	70	258	58	847	9	1996	0.0%	22.2/sec	146.1
HTTP Header Manager									
Gaussian Random Timer									
/advertisement/agent/edit.do	70	662	626	1035	194	1422	0.0%	20.9/sec	143.8
/advertisement/agent/listPlacedAds.do									
TOTAL	560	197	79	549	5	1996	0.0%	111.7/sec	565.0

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

Screenshot of JMeter Graph Results:

Graph Results

Name: Graph Results
Comments:

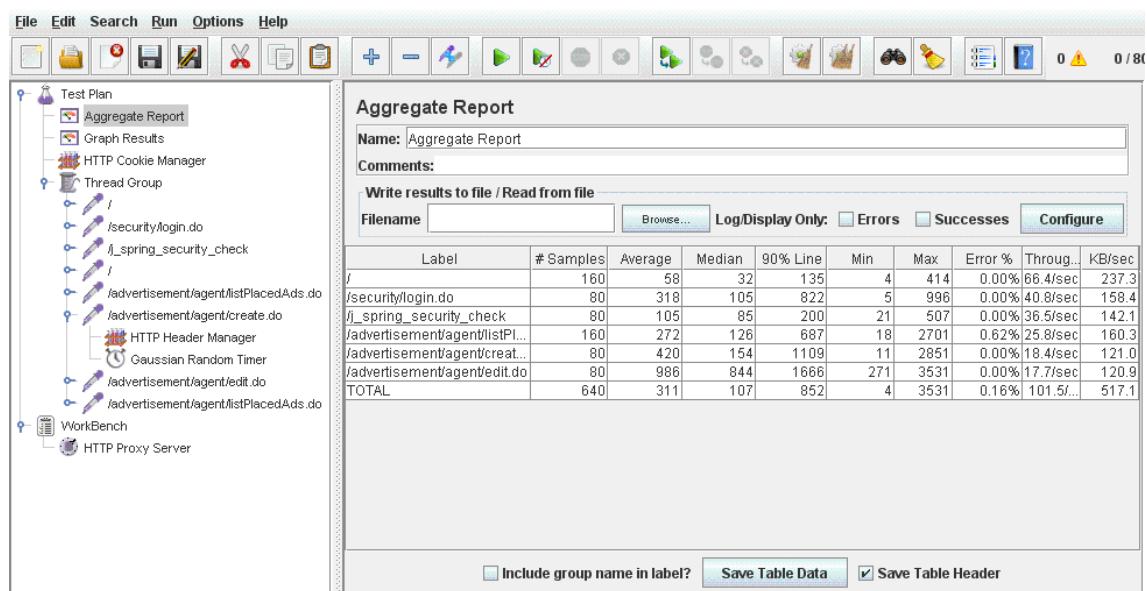
Write results to file / Read from file

Graphs to Display: Data Average Median Deviation Throughput

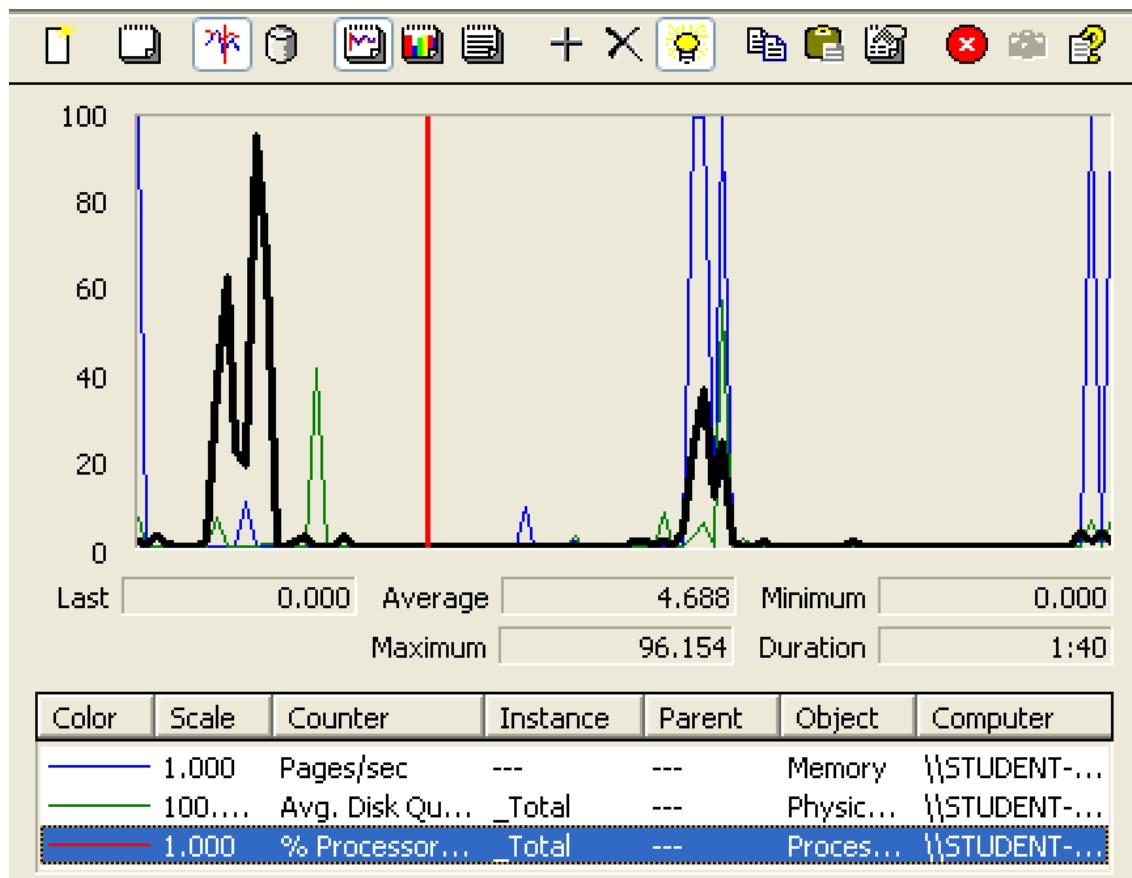
496 ms
0 ms

No of Samples 560 Latest Sample 41 Average 197
Deviation 284 Throughput 6,699.9/min Median 79

Observamos que a partir de los 80 usuarios comienza a dar fallos el listar periódicos, creemos que puede deberse a la tabla que generamos usando un plug-in de JQuery. Mirando en los logs de Tomcat no hemos encontrado nada extraño.



Los componentes de hardware funcionan correctamente y sin sobrecarga, incluso la CPU.



Usuarios máximos: 70

UC27: Listar los periódicos donde un agente no ha colocado anuncios y crear un anuncio (Newspaper 2.0 – C)

Para este caso de uso iniciaremos sesión como agente, iremos a listar los periódicos que no contienen anuncios de un agente y crearemos un nuevo anuncio, lo que nos devolverá a los periódicos en los que has colocado anuncios.

Este caso de uso ha soportado un número de usuarios concurrentes de 48.

Screenshot of JMeter Aggregate Report results:

Aggregate Report

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

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	96	14	9	19	3	224	0.00%	41.5/sec	148.1
/security/login.do	48	29	10	91	7	169	0.00%	21.6/sec	84.0
/_spring_security_check	48	28	18	51	13	148	0.00%	21.2/sec	82.6
/advertisement/agent/listNotPlacedAds.do	48	51	36	95	17	264	0.00%	20.5/sec	124.1
/advertisement/agent/create.do	48	47	36	96	16	188	0.00%	17.5/sec	115.1
HTTP Header Manager									
Gaussian Random Timer									
/advertisement/agent/edit.do	48	601	566	944	210	1236	0.00%	16.6/sec	113.3
/advertisement/agent/listPlacedAds.do	48	72	54	134	20	251	0.00%	21.8/sec	132.1
TOTAL:	384	107	27	355	3	1236	0.00%	88.1/sec	445.3

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

Screenshot of JMeter Graph Results:

Graph Results

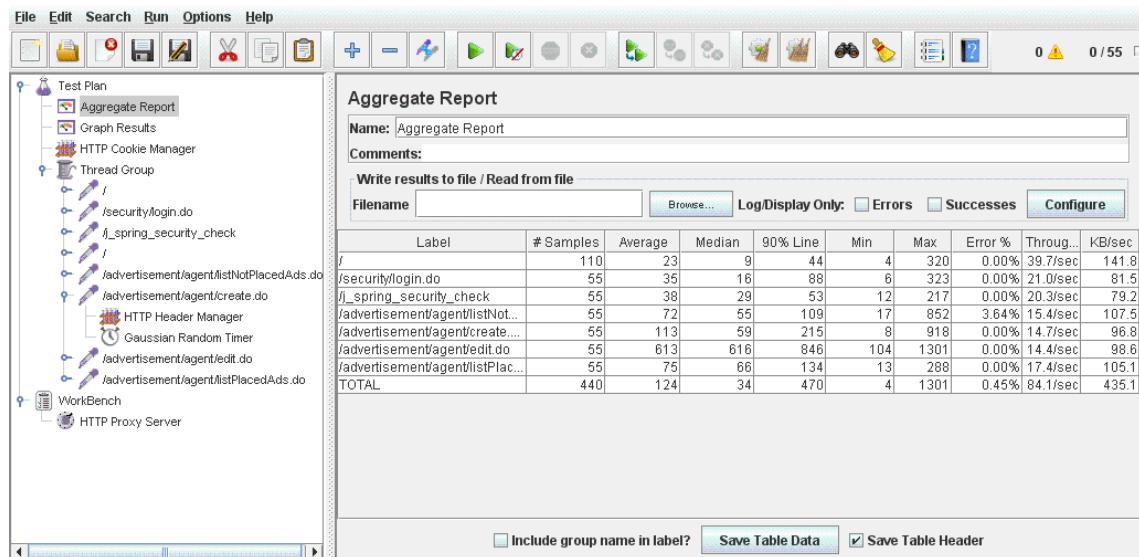
Name: Graph Results
Comments:
Write results to file / Read from file
Filename: Browse... Log/Display Only: Errors Successes Configure

Graphs to Display: Data Average Median Deviation Throughput

No of Samples: 384 Latest Sample: 30 Average: 107
Deviation: 209 Throughput: 5,284.404/minute Median: 27

Creemos que esto se debe al mismo motivo anterior, en el código no observamos ninguna acción extraña, ni durante el uso o testeo de esta página en concreto.

El número máximo de usuarios concurrentes es de 55.



De nuevo no observamos que los componentes de hardware se comporten de manera anómala.

Usuarios máximos: 48

UC28: Listar los anuncios que contienen palabras tabú y eliminar uno que considere inadecuado (Newspaper 2.0 – C)

En este caso de uso iniciaremos sesión como admin, iremos a listar los anuncios que contienen palabras tabú y eliminaremos un anuncio que consideremos inadecuado, lo que nos devolverá a la lista de anuncios que contienen palabras tabú.

El número de usuarios aceptables para esta prueba es de 240.

Screenshot of JMeter Aggregate Report results:

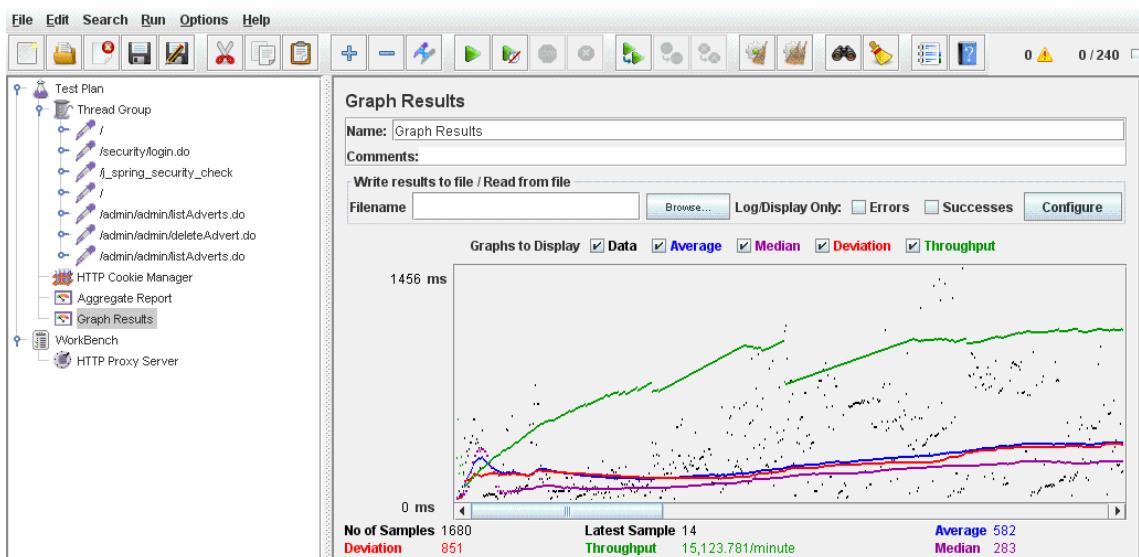
Test Plan Structure:

- Test Plan
- Thread Group
 - /
 - /security/login.do
 - /j_spring_security_check
 - /
 - /admin/admin/listAdverts.do
 - /admin/admin/deleteAdvert.do
 - /admin/admin/listAdverts.do
- HTTP Cookie Manager
- Aggregate Report
- Graph Results
- WorkBench
- HTTP Proxy Server

Aggregate Report Data (Table):

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	480	224	113	612	3	927	0.00%	75.7/sec	297.4
/security/lo...	240	2077	2317	3883	24	4720	0.00%	43.4/sec	168.6
/j_spring_...	240	399	329	772	25	1892	0.00%	37.9/sec	174.7
/admin/ad...	480	303	215	724	7	2148	0.00%	77.9/sec	359.9
/admin/ad...	240	544	462	1071	58	2350	0.00%	39.1/sec	187.7
TOTAL	1680	582	283	1418	3	4720	0.00%	252.1/sec	1094.9

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



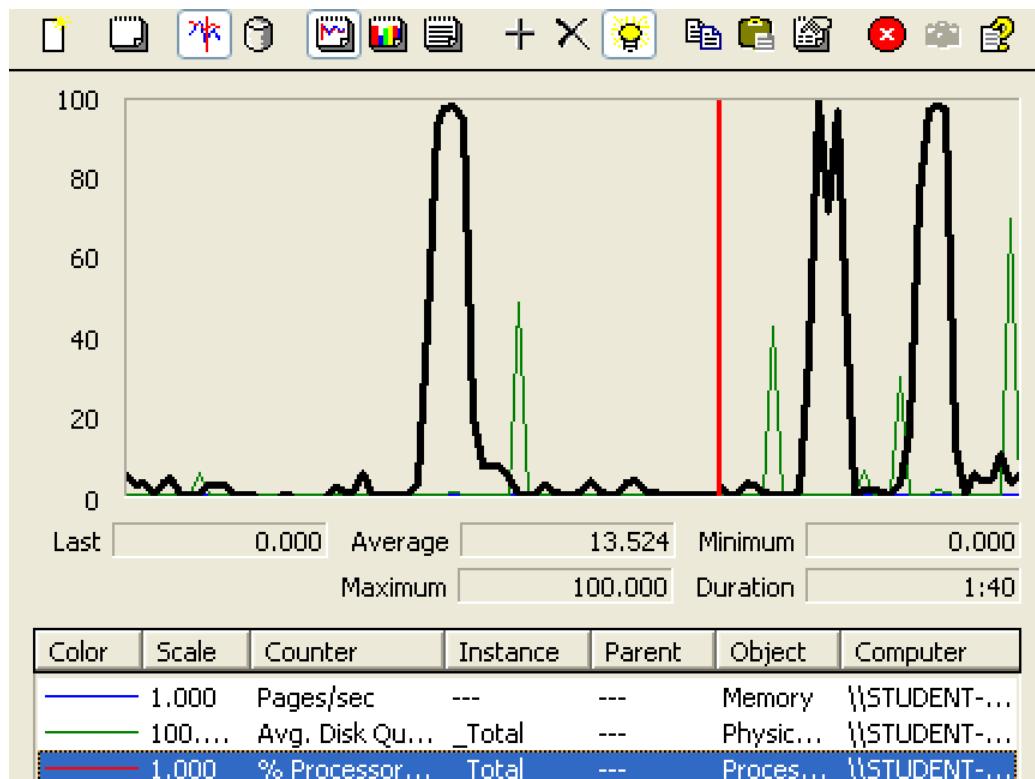
A partir de unos 250 usuarios, consideraremos que el test comienza a tener fallos.

The screenshot shows the JMeter interface with the 'Aggregate Report' component selected in the left sidebar under the 'Test Plan' tree. The main panel displays an 'Aggregate Report' table with the following data:

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	500	173	105	385	5	1044	0.00%	68.0/sec	267.1
/security/lo...	250	1365	1449	2446	6	3173	0.00%	59.0/sec	229.2
/j_spring_secur...	250	623	399	1445	23	3532	1.20%	34.5/sec	158.4
/admin/adminlistAdverts...	500	435	271	990	9	3967	0.20%	68.9/sec	317.7
/admin/admin/deleteAdvert...	250	902	746	1798	34	3662	1.20%	35.0/sec	167.0
TOTAL	1750	587	284	1638	5	3967	0.40%	234.9/sec	1017.4

At the bottom of the report panel, there are buttons for 'Include group name in label?' (unchecked), 'Save Table Data' (checked), and 'Save Table Header' (checked).

El motivo principal al examinar el uso de los componentes creemos que es llegar al máximo de uso de la CPU de nuestra máquina de pruebas. El resto de componentes, no muestran alto trabajo.



Usuarios máximos: 240

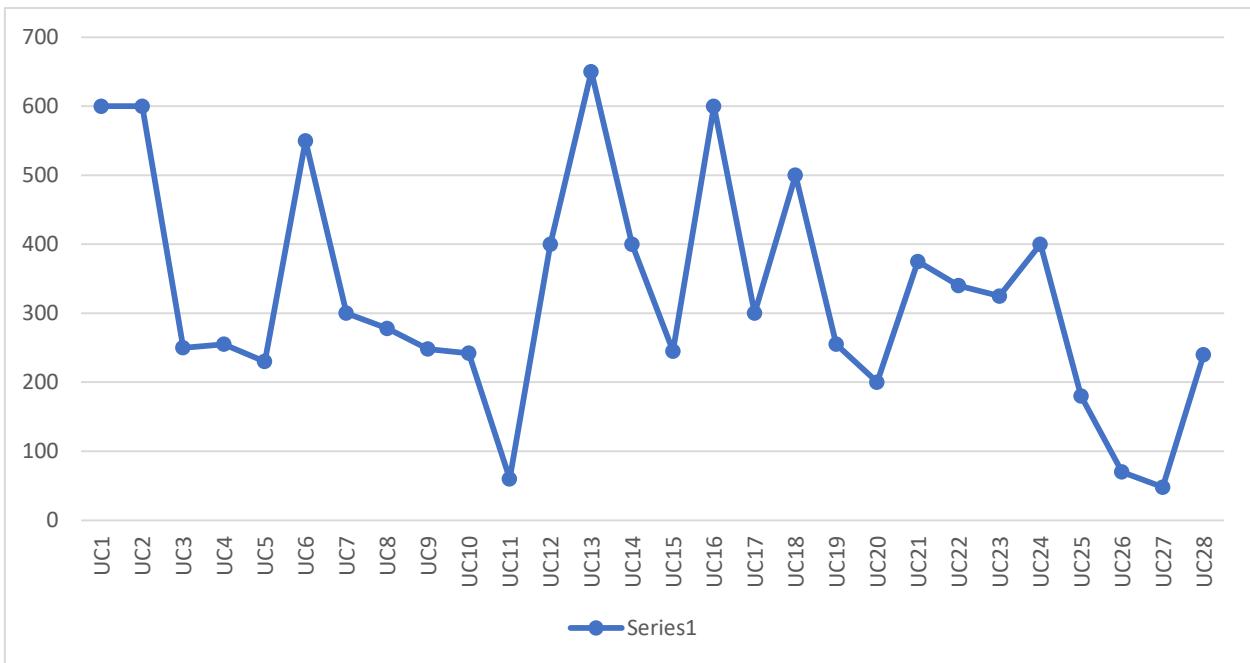
Resumen de los test de rendimiento

Tras realizar todos los test de rendimiento podemos decidir cual sería el número de usuarios máximo que nuestro sistema puede soportar.

Como vemos en la gráfica el caso de uso xx es el que menos usuarios puede soportar, sin embargo, tenemos que tener en cuenta que este caso de uso es exclusivo de admins por lo que nos indica ese dato es que el sistema no podría soportar más de 60 admins. En nuestro sistema tan solo tenemos un admin por lo que podemos obviar ese dato. Tras ese, el caso de uso que menos actores en general soporta es el xx.

Para estas pruebas hemos usado los equipos proporcionados por la asignatura. Están configurados con Windows XP Professional 64 bit, 30gb de disco duro, 2Gb de ram y un procesador de 2 núcleos. Para medir el uso de los recursos usados durante las pruebas hemos usado performance.

Caso de uso	Máx. Usuarios
UC1	600
UC2	600
UC3	250
UC4	255
UC5	230
UC6	550
UC7	300
UC8	278
UC9	248
UC10	242
UC11	60
UC12	400
UC13	650
UC14	400
UC15	245
UC16	600
UC17	300
UC18	500
UC19	255
UC20	200
UC21	375
UC22	340
UC23	325
UC24	400
UC25	180
UC26	70
UC27	48
UC28	240



Los casos de uso más bajos son el UC26 y UC27 los cuales tan solo soportan 70 y 48 agentes concurrentes, este caso de uso es de agente, por lo que podríamos decir que el número máximo de agentes concurrentes en el sistema será de 48.

El segundo más bajo es el UC25 el cual soporta 180 usuarios como máximo, por lo tanto podemos determinar que el **número máximo de usuarios del sistema es de 180**.