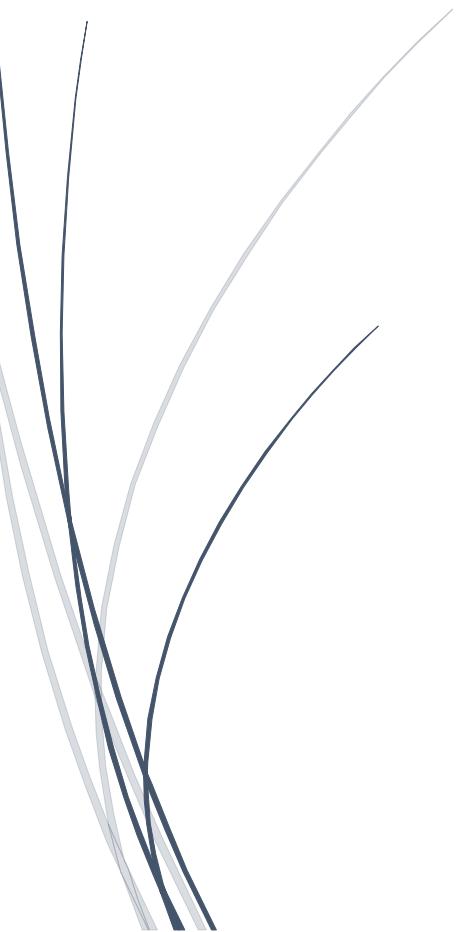


16-5-2019

Acme Rookies

Performance Testing



Maria Jimenez Vega
Álvaro Calle González
Julia García Gallego
Antonio Nolé Anguita
Fernando Manuel Ruiz Pliego

INTRODUCTION

In the following document we will analyze the maximum performance of our system after performing the performance tests using the jmeter tool.

The document consists of all the test cases carried out, for each test case the characteristics of each computer in which the test has been carried out, a series of captures and a conclusion are explained.

It should be noted that although they have been made on different computers all have run the virtual machine "**Pre-Production 1.18.2**".

In the analysis of the tests we have considered that a time superior to 2500 ms in 90% line is not considered acceptable.

PERFORMANCE TESTING

Acme Rookies /Req. 3.2 - An actor who is authenticated as an auditor must be able to create and delete an audit.

- RAM: 8 GB
- CPU: Intel Core i7-8550U 1,8GHz
- Hard disk: 1TB
- Network card: Intel(R) Dual Band Wireless-AC 3165

Test case description: The user login as an auditor, go to available positions list and display a position. Click on AUDIT link, complete the form and save. Finally, when a user is in the audit list click on edit and delete it. The user logout to the system

Maximum workload test case: 140 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties

Number of Threads (users):

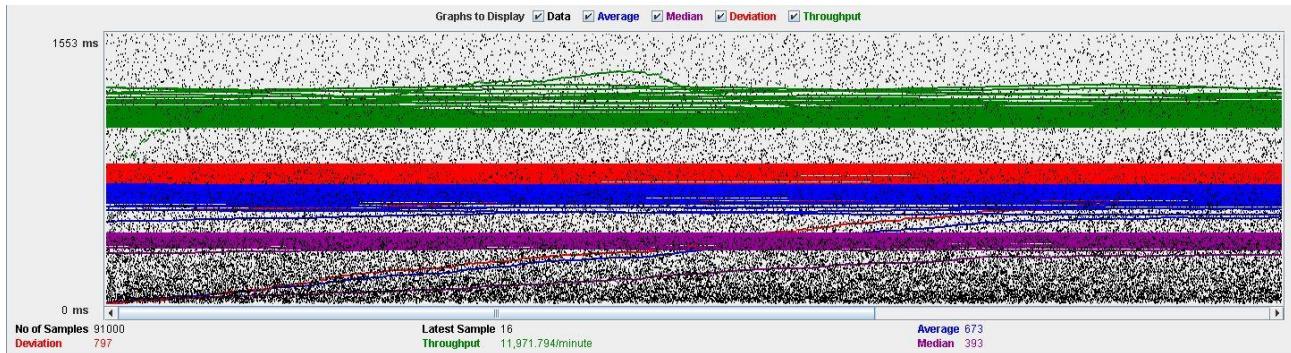
Ramp-Up Period (in seconds):

Loop Count: Forever

Delay Thread creation until needed

With this configuration we make sure that the system doesn't produce any errors and has a good response time.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	7000	541	297	1340	5	6415	0.00%	15.4/sec	63.8
/j_spring_security_check	7000	1113	806	2471	19	8785	0.00%	15.4/sec	63.6
/	14000	549	304	1373	4	10106	0.00%	30.7/sec	119.7
/position/availableList.do	7000	545	300	1327	5	7695	0.00%	15.4/sec	87.8
/position/display.do	7000	583	339	1386	17	7895	0.00%	15.4/sec	80.6
/audit/audit/create.do	7000	1069	775	2365	21	9657	0.00%	15.4/sec	63.1
/audit/audit/edit.do	21000	718	434	1764	5	8997	0.00%	46.1/sec	406.6
/audit/audit/list.do	14000	548	308	1355	6	8720	0.00%	30.7/sec	146.7
/j_spring_security_logout	7000	551	304	1376	12	8237	0.00%	15.4/sec	62.6
TOTAL	91000	673	393	1658	4	10106	0.00%	199.5/sec	1093.8

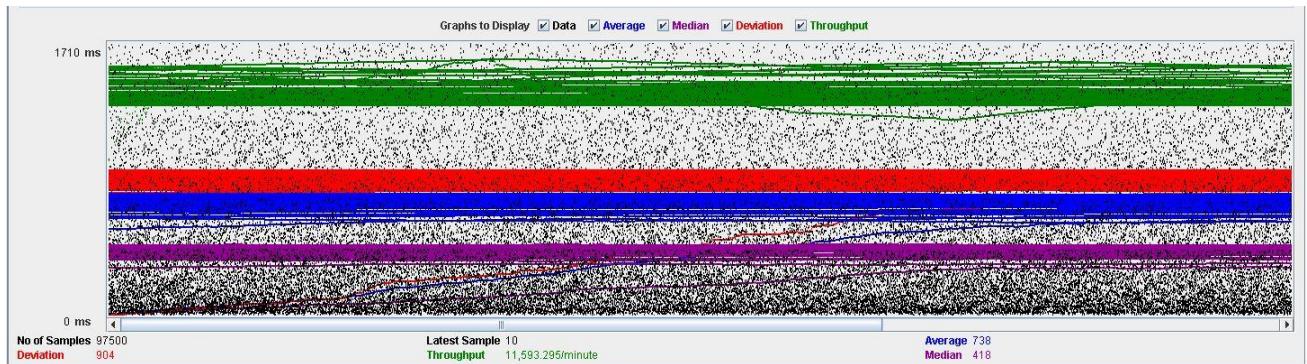


Overload test case: 150 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties	
Number of Threads (users):	150
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	

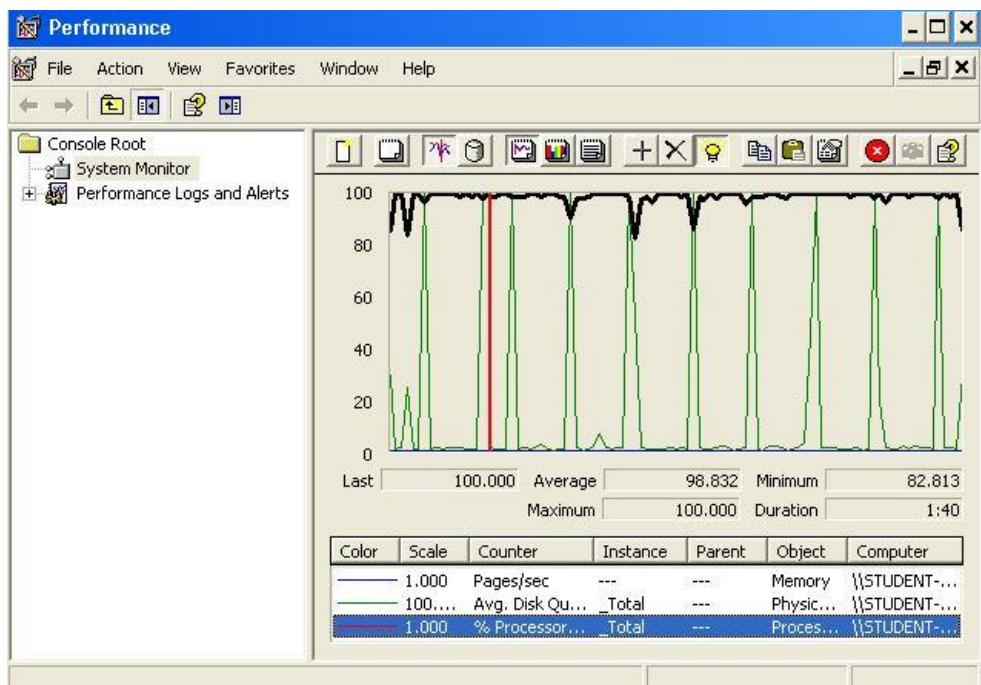
Although there aren't errors, the 90%-line time is not acceptable. The requests with highest time are "j_spring_security_check". However, this method is an internal unit of Spring. That's means that we cannot improve the performance by refactoring the code.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	7600	603	307	1513	3	9060	0.00%	14.9/sec	61.8
j_spring_security_check	7500	1208	870	2663	12	10625	0.00%	14.9/sec	61.6
/	15000	588	308	1500	3	8588	0.00%	29.7/sec	115.9
/position/availableList.do	7500	598	318	1503	7	7328	0.00%	14.9/sec	85.1
/position/display.do	7500	648	367	1568	16	10048	0.00%	14.9/sec	78.0
/audit/auditfor/read.do	7500	1187	845	2673	15	14545	0.00%	14.9/sec	61.1
/audit/auditfor/edit.do	22500	787	465	1956	5	10460	0.00%	44.6/sec	393.7
/audit/auditfor/list.do	15000	597	313	1499	5	9286	0.00%	29.7/sec	142.0
j_spring_security_logout	7500	619	333	1541	11	10112	0.00%	14.9/sec	60.7
TOTAL	97500	738	418	1847	3	14545	0.00%	193.2/sec	1059.2



With the following picture is performance analysis in which we can see that CPU are working more than 90% all the time. So, if we improve the resources of the CPU are possible that the system supports more user concurrently.

In this analysis we can also see that hard disk make some peak. We think that is when the system goes to memory to get the list.



Conclusion: The maximum number of concurrent users supported by the system is 140 because the current CPU is a bottleneck.

Acme Rookies /Req. 3.2 - An actor who is authenticated as an auditor must be able to edit his or her audit.

- RAM: 8 GB
- CPU: Intel Core i7-8550U 1,8GHz
- Hard disk: 1TB
- Network card: Intel(R) Dual Band Wireless-AC 3165

Test case description: The user login as an auditor and go to audit. Click on edit link, complete the form and save. Finally, the user logout to the system.

Maximum workload test case: 150 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties

Number of Threads (users): 150

Ramp-Up Period (in seconds): 1

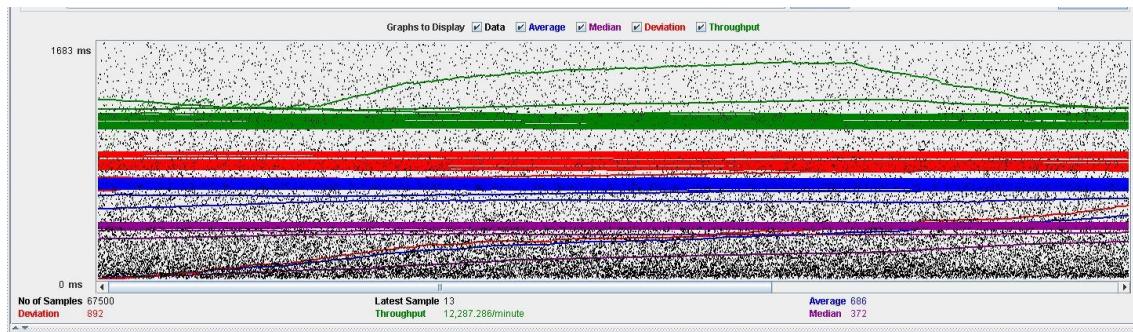
Loop Count: Forever 50

Delay Thread creation until needed

Scheduler

With this configuration we make sure that the system doesn't produce any errors and has a good response time.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	7500	559	290	1376	3	9467	0.00%	22.8/sec	94.6
/spring_security_check	7500	1112	769	2483	11	16785	0.00%	22.8/sec	94.2
/audit/auditonedit.do	15000	560	290	1413	4	10381	0.00%	45.6/sec	177.5
/audit/auditoreedit.do	15000	563	291	1413	8	16202	0.00%	45.6/sec	216.7
/spring_security_logout	7500	544	290	1329	8	12459	0.00%	22.8/sec	92.7
TOTAL	67500	686	372	1711	3	16785	0.00%	204.8/sec	898.2



Overload test case: 160 concurrent users, 50 of loop count and 1 as ramp-up period.

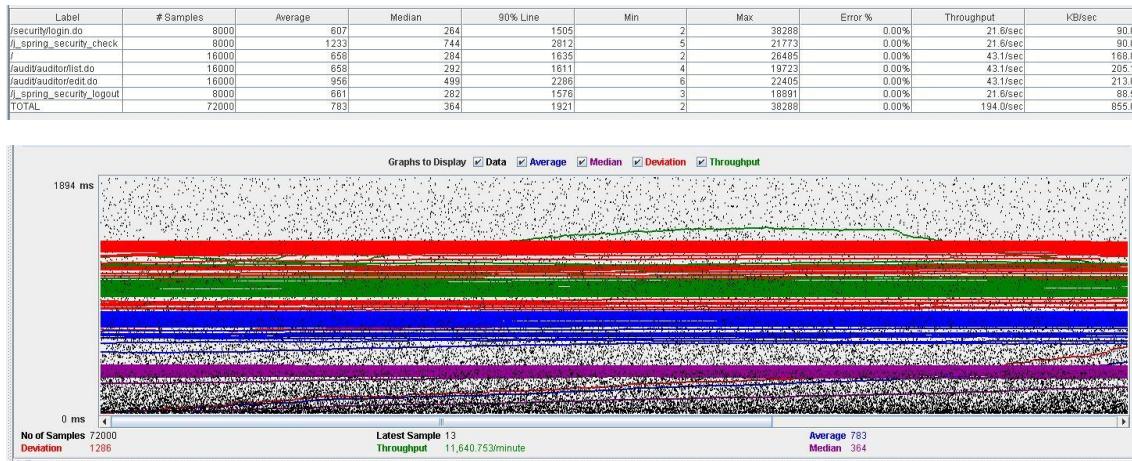
Thread Properties

Number of Threads (users): 160

Ramp-Up Period (in seconds): 1

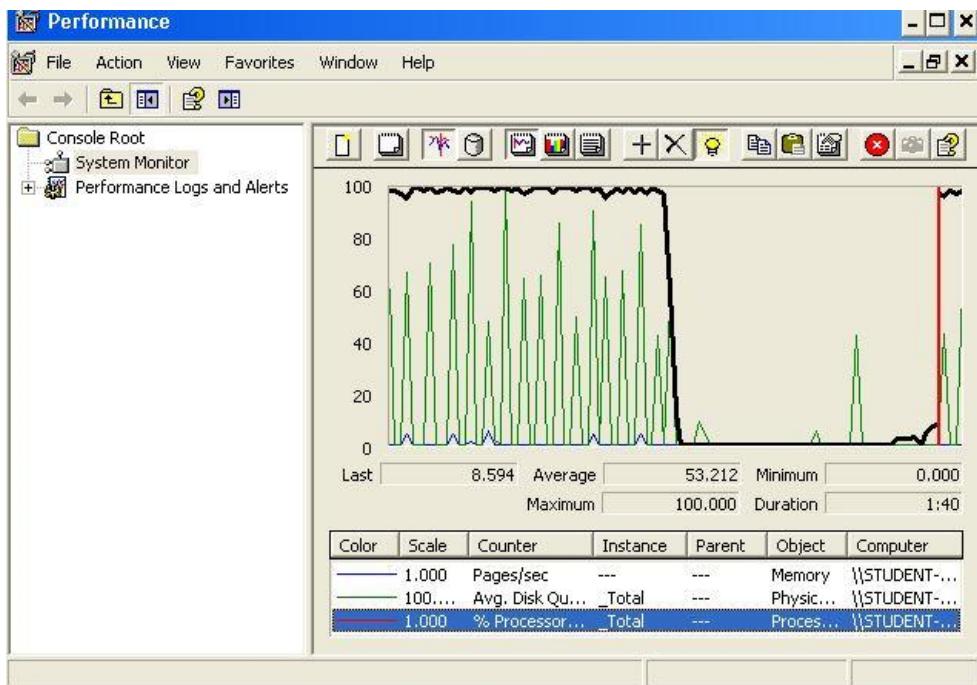
Loop Count: Forever 50

Although there aren't errors, the 90%-line time is not acceptable. The requests with highest time are "j_spring_security_check". However, this method is an internal unit of Spring. That's means that we cannot improve the performance by refactoring the code.



With the following picture is performance analysis in which we can see that CPU are working more than 90% all the time. So, if we improve the resources of the CPU are possible that the system supports more user concurrently.

In this analysis we can also see that hard disk make some peak. We think that is when the system goes to memory to get the list.



Conclusion: The maximum number of concurrent users supported by the system is 150 because the current CPU is a bottleneck.

Acme Rookies /Req. 3.2 - An actor who is authenticated as an auditor must be able to list and display his or her audit.

- RAM: 8 GB
- CPU: Intel Core i7-8550U 1,8GHz
- Hard disk: 1TB
- Network card: Intel(R) Dual Band Wireless-AC 3165

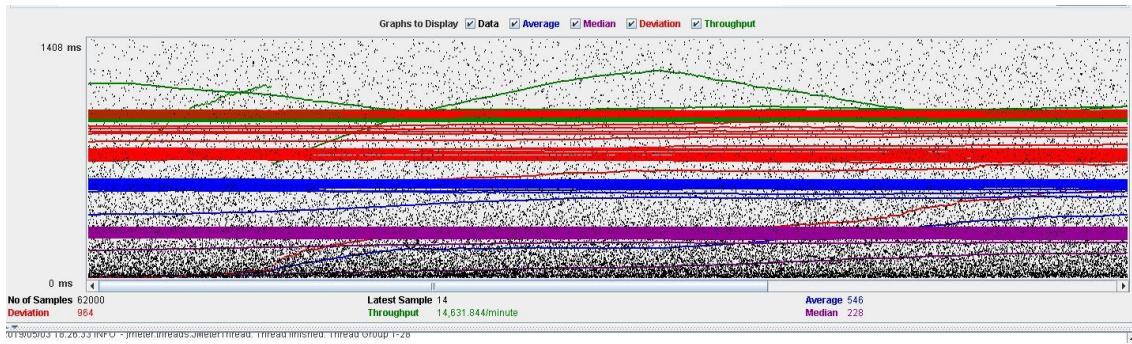
Test case description: The user login as an auditor and go to audit. Click on edit link and display an audit. Finally, the user logout to the system.

Maximum workload test case: 155 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties	
Number of Threads (users):	155
Ramp-Up Period (in seconds):	1
Loop Count:	<input type="checkbox"/> Forever 50

With this configuration we make sure that the system doesn't produce any errors and has a good response time.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	23250	482	189	1181	2	22053	0.00%	91.4/sec	368.2
/security/login.do	7750	482	195	1219	3	20778	0.00%	30.5/sec	124.8
/_spring_security_check	7750	979	591	2344	7	16631	0.00%	30.5/sec	126.7
/audit/auditorlist.do	7750	470	199	1147	5	14431	0.00%	30.5/sec	132.8
/audit/display.do	7750	488	212	1189	6	13590	0.00%	30.5/sec	129.7
/_spring_security_logout	7750	498	199	1222	3	18209	0.00%	30.5/sec	124.6
TOTAL	62000	546	228	1346	2	22053	0.00%	243.9/sec	996.2

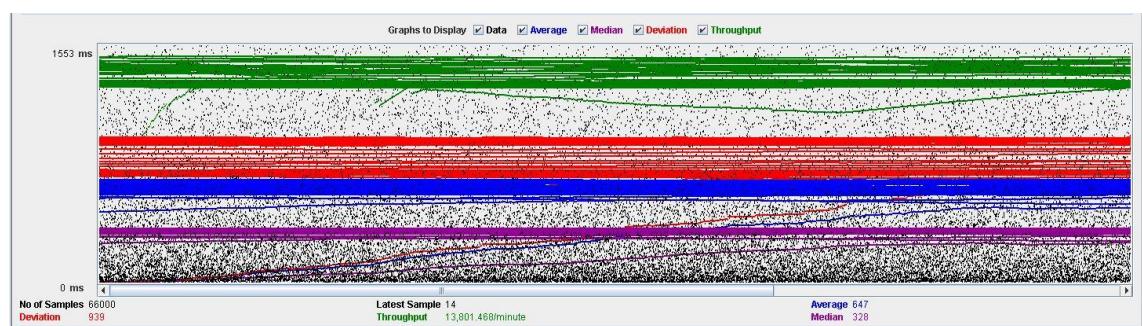


Overload test case: 165 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties	
Number of Threads (users):	165
Ramp-Up Period (in seconds):	1
Loop Count:	<input type="checkbox"/> Forever 50

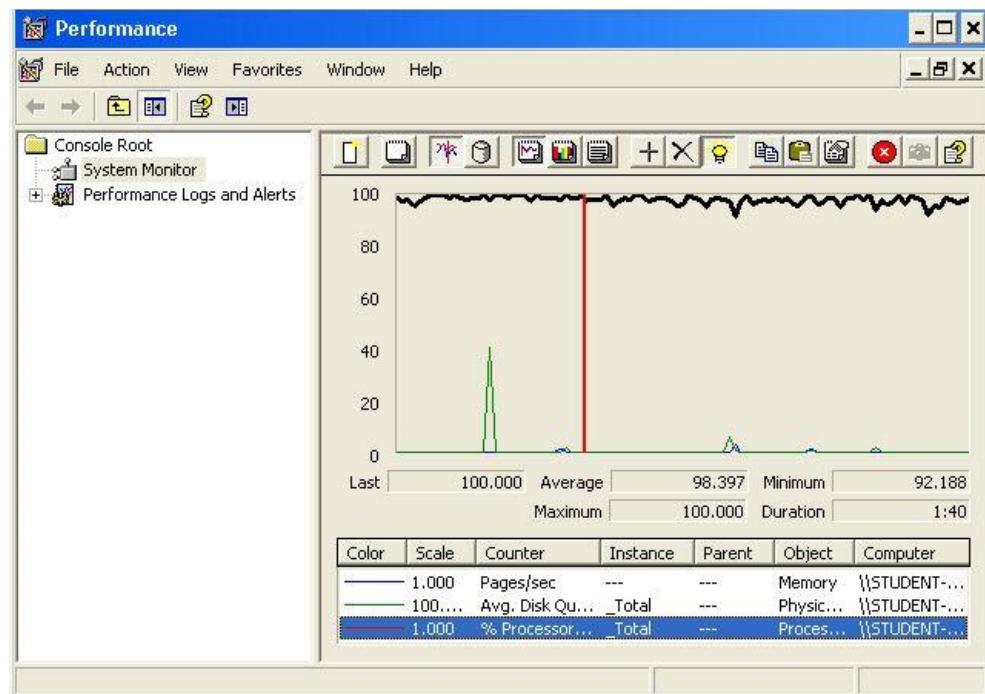
Although there aren't errors, the 90%-line time is not acceptable. The requests with highest time are "j_spring_security_check". However, this method is an internal unit of Spring. That's means that we cannot improve the performance by refactoring the code.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	24750	565	276	1434	2	19295	0.00%	88.3/sec	338.6
j_spring_security_check	8250	598	290	1495	2	23201	0.00%	28.8/sec	117.9
/audit/auditonlist.do	8250	1164	798	2631	7	16378	0.00%	28.8/sec	120.0
/audit/display.do	8250	564	287	1396	6	23876	0.00%	28.8/sec	125.6
j_spring_security_logout	8250	599	305	1476	5	27792	0.00%	28.8/sec	122.6
TOTAL	66000	647	328	1630	2	27792	0.00%	230.0/sec	942.0



With the following picture is performance analysis in which we can see that CPU are working more than 90% all the time. So, if we improve the resources of the CPU are possible that the system supports more user concurrently.

In this analysis we can't see that hard disk make peak like the other similar user case.



Conclusion: The maximum number of concurrent users supported by the system is 155 because the current CPU is a bottleneck.

Acme Rookies /Req. 4.1 - An actor who is authenticated as an administrator must be able to run a procedure to notify the existing users of the rebranding. The system must guarantee that the process is run only once.

We aren't going to do a test performance about this requirement because, as requirement's statement indicates, the process will be only executed once. It doesn't make sense creating a test performance with 1 concurrent user and 1 of loop count as parameter of the "thread group". Any other value in those parameters would cause errors.

Acme Rookies /Req. 4.2 - An actor who is authenticated as an administrator must be able to create user accounts for new auditors.

Technical details of the computer on which the test has been executed:

- Memory RAM: 8,00 GB
- CPU: Intel Core i7-7700HQ
- Hard Disk: 256 GB SSD + 921 GB HDD
- Interface network: Intel(R) Dual Band Wireless-AC 3168

Test case description: First, the user logs in the system as an administrator. Then, he/she goes to Administrator option menu and click on Register Auditor, he/she fills the form and save it. Next, the user logs out the system and return to logs in as the new auditor created.

Maximum workload test case: 350 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties	
Number of Threads (users):	350
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

According to the performance test, this is the maximum workload that can be supported by the system without errors and failures or insufficient performance.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	87500	214	119	522	2	6108	0.00%	109.1/sec	438.2
/j_spring_security_check	35000	209	113	514	2	5370	0.00%	43.9/sec	179.7
/actor/administrator/regi...	35000	425	303	928	5	5606	0.00%	43.9/sec	192.8
/welcome/index.do	35000	249	140	579	4	5672	0.00%	44.2/sec	432.0
/j_spring_security_logout	17500	246	137	594	3	4433	0.00%	22.1/sec	97.3
TOTAL	245000	252	142	813	2	6108	0.00%	305.4/sec	1510.5



Overload test case: 375 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties

Number of Threads (users):	375
Ramp-Up Period (in seconds):	1
Loop Count:	<input type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

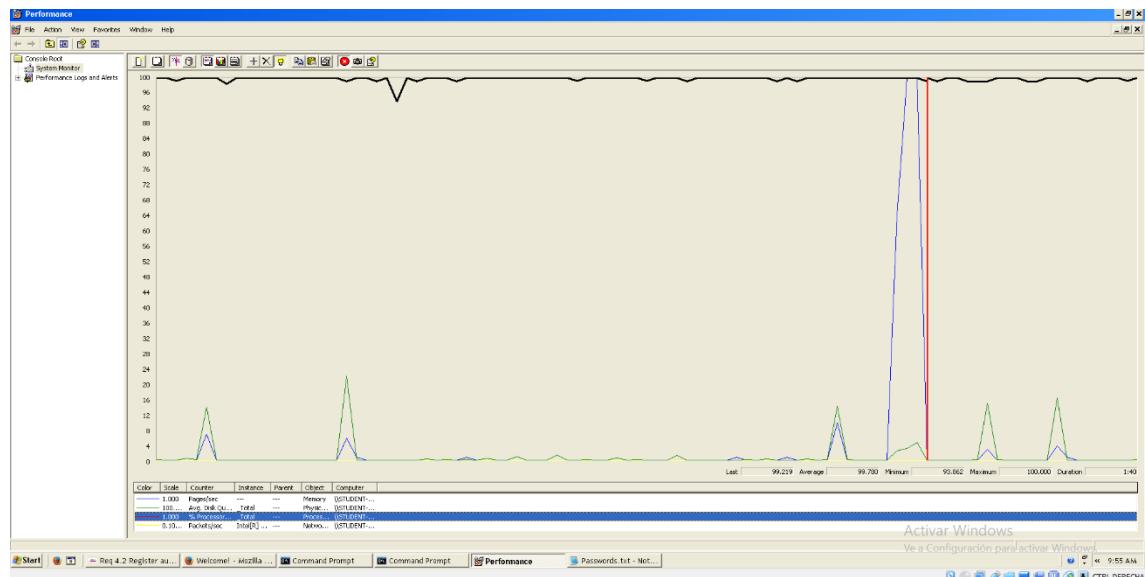
Even though the average time per request is still acceptable, it begins to produce some errors as we can see in the following picture. The errors are always the same: I/O exception (java.net.SocketException) caught when processing request: Connection reset by peer: socket write error.

This exception is not related with the implementation of our application, but with tomcat. The system can't handle this number of concurrent users properly.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	83750	321	156	814	1	8566	0.01%	104.4/sec	419.8
/security/login.do	37500	326	160	830	2	8850	0.00%	42.0/sec	172.0
/j_spring_security_check	37500	653	437	1491	1	8767	0.01%	42.0/sec	184.7
/actor/administrator/regis...	37500	348	177	870	4	9044	0.00%	42.3/sec	413.6
/welcome/index.do	18750	340	167	868	3	7740	0.01%	21.2/sec	83.1
/j_spring_security_logout	37600	341	174	863	2	7688	0.02%	42.1/sec	173.2
TOTAL	262500	377	181	958	1	9844	0.01%	282.4/sec	1446.8



As we can see in the graph below, there is a bottleneck with the CPU. Probably we could improve the maximum workload of the application if we assign more processors to the virtual machine.



Conclusion: The maximum number of concurrent users supported by the system is 350 because the current CPU is a bottleneck.

Acme Rookies /Req. 4.3 - An actor who is authenticated as an administrator must be able to launch a process to compute an audit score for every company. The audit score is computed as the average of the audit scores that the positions offered by a company has got, but normalised to range 0.00 up to +1.00 using a linear homothetic transformation. Note that the audit score of a company that hasn't got any audits is not 0.00, but nill.

Technical details of the computer on which the test has been executed:

- Memory RAM: 16 GB DDR4 Memory.
- CPU: Intel(R) Core (TM) i7-7500U CPU @ 2.70 GHz (4 CPUs) ~ 2.90 GHz.
- Hard Disk: 256 GB SSD + 1000 GB HDD.
- Interface network: Ethernet: Realtek PCIe GBE Family Controller | Wi-Fi: Qualcomm Atheros QCA9377 Wireless Network Adapter.

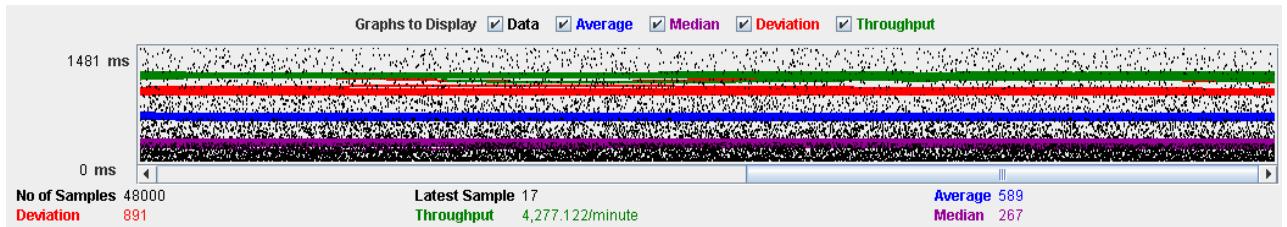
Test case description: first, a user logs in as administrator. Then, the administrator lists actors of the system. Later, the administrator launches the process. Finally, the administrator logs out.

Maximum workload test case: 120 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties	
Number of Threads (users):	120
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

According to the performance test, this is the maximum workload that can be supported by the system without errors and failures or insufficient performance.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
/security/login.do	6000	416	143	1070	7	10494	0.00%	9.0/sec	37.4
/j_spring_security_check	6000	904	483	2219	17	14143	0.00%	9.0/sec	41.2
/	12000	449	175	1150	8	10946	0.00%	17.9/sec	73.5
/actor/administrator/list.do	12000	531	256	1236	37	9909	0.00%	17.9/sec	124.8
/actor/administrator/auditScoreProcess.do	6000	1001	592	2306	31	11455	0.00%	9.0/sec	66.8
/j_spring_security_logout	6000	432	161	1141	13	8917	0.00%	9.0/sec	36.7
TOTAL	48000	589	267	1502	7	14143	0.00%	71.3/sec	377.7



Overload test case: 130 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

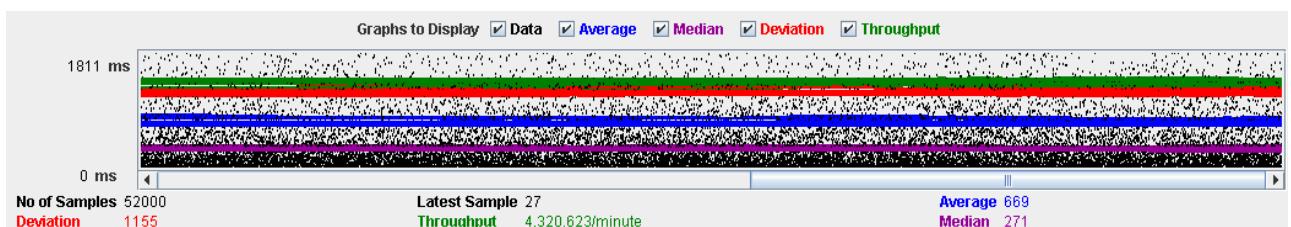
Loop Count: Forever

Delay Thread creation until needed

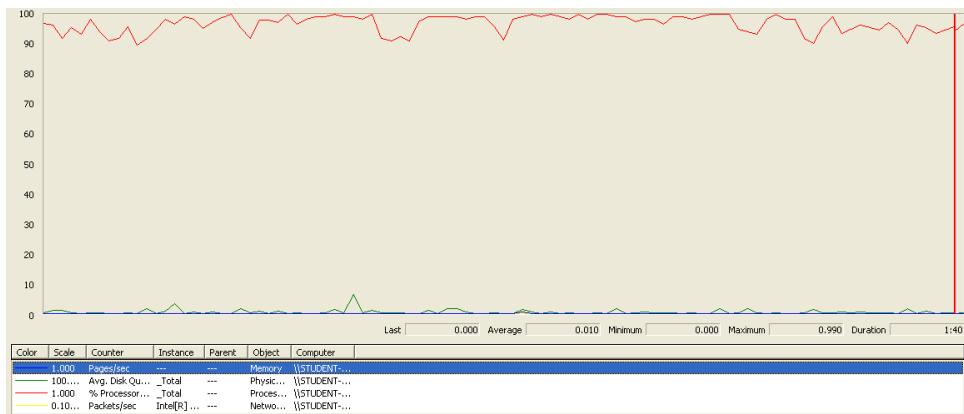
Scheduler

Although there aren't errors, the average time per request is not acceptable. The requests with highest time are "/actor/administrator/auditScoreProcess.do" and "j_spring_security_check". "j_spring_security_check" is an internal unit of Spring. That's means that we cannot improve the performance by refactoring the code. Regarding "/actor/administrator/auditScoreProcess.do" is an implementation of the developer team. It's possible refactoring the code.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
/security/login.do	6500	460	137	1134	7	13907	0.00%	9.1/sec	37.7
/j_spring_security_check	6500	1006	506	2567	17	18251	0.00%	9.1/sec	41.5
/	13000	509	181	1252	9	14526	0.00%	18.0/sec	74.3
/actor/administrator/list.do	13000	597	259	1379	32	15670	0.00%	18.1/sec	126.1
/actor/administrator/auditScoreProcess.do	6500	1181	611	2879	46	17906	0.00%	9.1/sec	67.5
/j_spring_security_logout	6500	497	171	1221	4	14737	0.00%	9.1/sec	37.0
TOTAL	52000	669	271	1674	4	18251	0.00%	72.0/sec	381.5



In the following performance analysis we can appreciate that CPU is running at 90-100% all the time. So, there is a bottleneck with CPU. If we use a more powerful CPU, the maximum workload would be higher.



Conclusion: The maximum number of concurrent users supported by the system is 120 because the current CPU is a bottleneck.

Acme Rookies /Req. 4.4 - An actor who is authenticated as an administrator must be able to display a dashboard

Technical details of the computer on which the test has been executed:

- Memory RAM: 8,00 GB
- CPU: Intel Core i7-7700HQ
- Hard Disk: 256 GB SSD + 921 GB HDD
- Interface network: Intel(R) Dual Band Wireless-AC 3168

Test case description: The user logs in the system as an administrator, then, he/she goes to Administrator option menu and select option 'Display dashboard'. Finally, the user logs out the system.

Maximum workload test case: 320 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties

Number of Threads (users):	320
Ramp-Up Period (in seconds):	1
Loop Count:	<input type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

According to the performance test, this is the maximum workload that can be supported by the system without errors and failures or insufficient performance.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
security/login.do	48000	515	297	1247	1	11374	0.01%	89.7/sec	365.5
_spring_security_check	16000	482	267	1199	1	10535	0.02%	30.3/sec	124.0
dashboard/administrat...	16000	1005	719	2203	3	14491	0.01%	30.3/sec	139.4
_spring_security_logout	16000	607	387	1364	2	15543	0.01%	30.3/sec	380.8
OTAL	112000	595	353	1432	1	20470	0.02%	30.3/sec	124.2



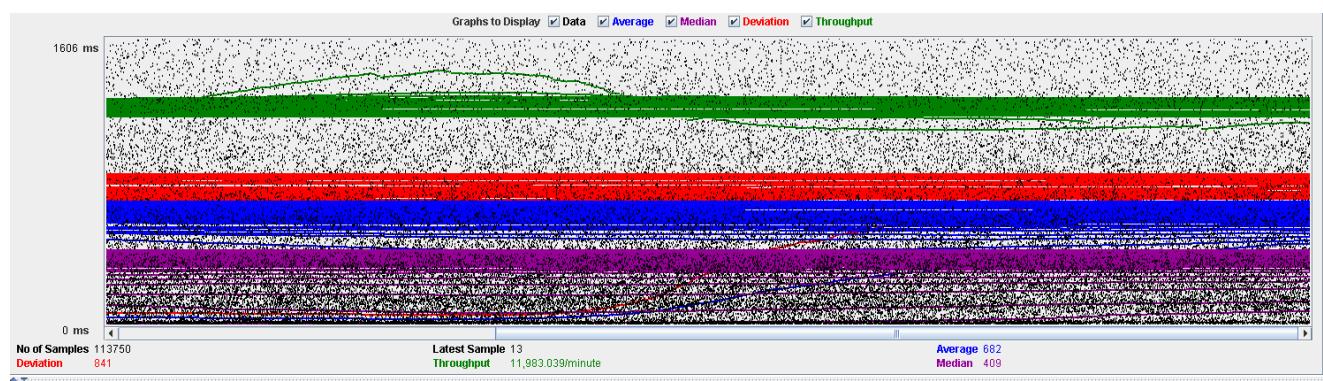
Overload test case: 325 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties

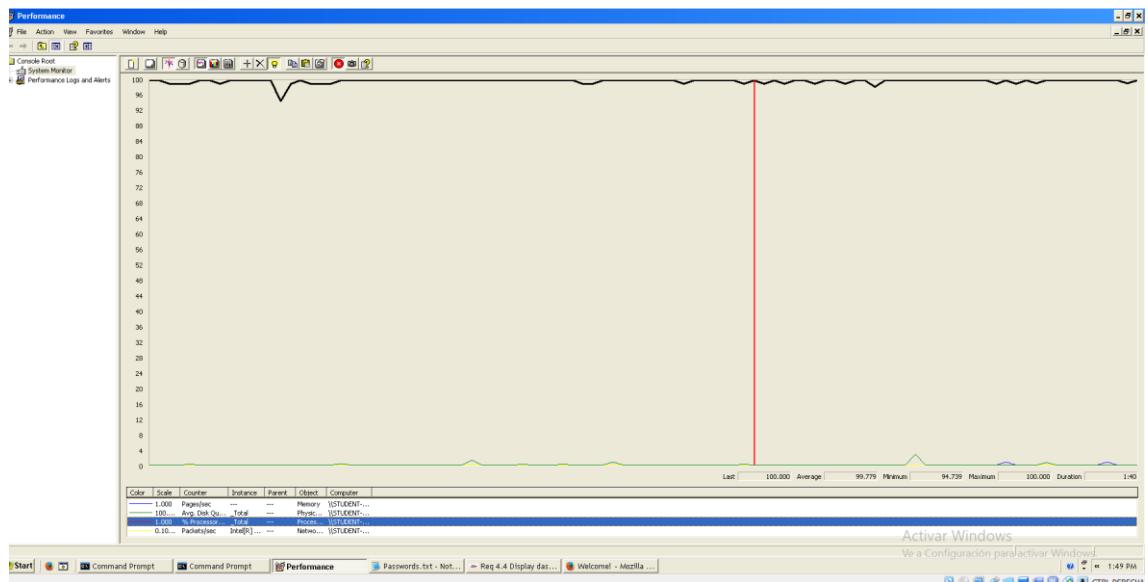
Number of Threads (users):	325
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

Although there aren't errors, the average time per request is not acceptable. The requests with highest time is "j_spring_security_check" this is an internal unit of Spring. That's means that we cannot improve the performance by refactoring the code.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
security/login.do	48750	580	336	1395	2	16859	0.00%	85.0/sec	348.7
spring_security_check	16250	573	330	1388	2	9653	0.00%	28.8/sec	118.2
ashboard/administrato...	16250	1178	858	2538	4	18318	0.00%	28.8/sec	132.8
spring_security_logout	16250	692	441	1544	19	16490	0.00%	28.8/sec	362.9
JTAL	113750	591	354	1412	4	10131	0.00%	28.8/sec	118.3
								199.7/sec	1073.3



In the following performance analysis we can appreciate that CPU is running at 95-100% all the time. So, there is a bottleneck with CPU. If we use a more powerful CPU, the maximum workload would be higher.



Conclusion: The maximum number of concurrent users supported by the system is 320 because the current CPU is a bottleneck.

Acme Rookies /Req. 9.1 - An actor who is not authenticated must be able to browse the list of providers and navigate to their items.

Technical details of the computer on which the test has been executed:

- Memory RAM: 8,00 GB
- CPU: Intel Core i7-7700HQ
- Hard Disk: 256 GB SSD + 921 GB HDD
- Interface network: Intel(R) Dual Band Wireless-AC 3168

Test case description: The user goes to Provider list menu option and then, display one of the providers of the list. In the profile, click on Items to see the items of the provider and display one of them.

Maximum workload test case: 450 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: Forever

Delay Thread creation until needed

Scheduler

With this configuration we make sure that the system doesn't produce any errors and has a good response time.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
providerlist.do	22500	342	168	875	2	10162	0.00%	55.6/sec	219.7
actordisplay.do	22500	357	186	900	6	8307	0.00%	55.6/sec	294.2
itemlist.do	22500	349	175	886	4	7117	0.00%	55.6/sec	275.6
itemdisplay.do	22500	346	168	880	4	8741	0.00%	55.6/sec	250.5
TOTAL	112500	349	174	887	2	10162	0.00%	273.6/sec	1264.9



Overload test case: 475 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: Forever

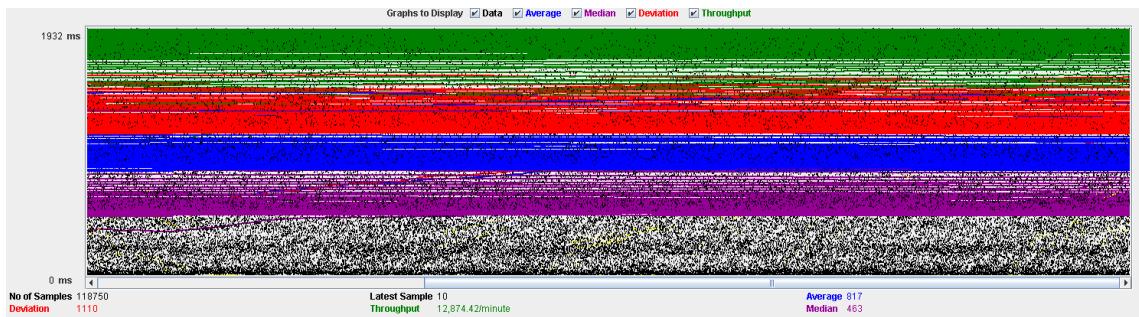
Delay Thread creation until needed

Scheduler

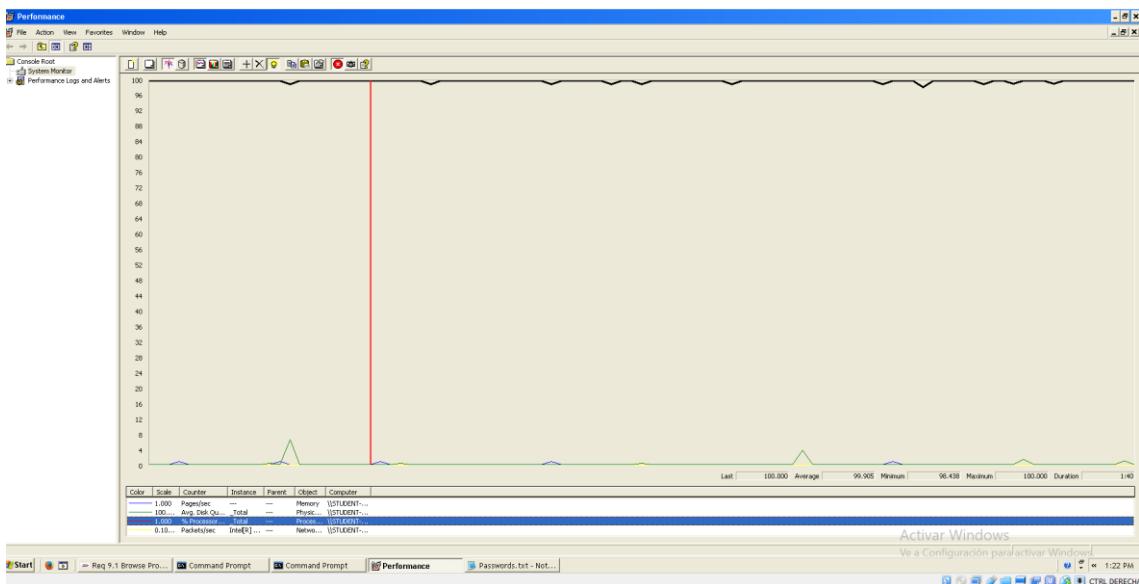
Even though the average time per request is still acceptable, it begins to produce some errors as we can see in the following picture. The errors are always the same: I/O exception (java.net.SocketException) caught when processing request: Connection reset by peer: socket write error.

This exception is not related with the implementation of our application, but with tomcat. The system can't handle this number of concurrent users properly.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	23750	816	439	1974	0	21360	0.31%	43.4/sec	171.4
/providerlist.do	23750	833	493	1912	0	21739	1.65%	43.4/sec	227.3
/actor/display.do	23750	819	472	1911	0	27461	1.49%	43.4/sec	213.3
/itemlist.do	23750	817	464	1925	0	19869	1.45%	43.4/sec	194.3
/item/display.do	23750	800	447	1903	0	16848	1.30%	43.5/sec	189.8
TOTAL	118750	817	463	1925	0	27461	1.24%	214.8/sec	984.6



In the following performance analysis we can appreciate that CPU is running almost 100% all the time. So, there is a bottleneck with CPU. If we use a more powerful CPU, the maximum workload would be higher.



Conclusion: The maximum number of concurrent users supported by the system is 450 because the current CPU is a bottleneck.

Acme Rookies /Req. 9.2 – An actor who is not authenticated must be able to browse the list of items and navigate to their providers.

Technical details of the computer on which the test has been executed:

- Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
- CPU: Intel Core i5-4200U
- Disco duro: 240 GB SSD

Test case description:

- The unauthenticated user clicks on "Items List" from the menu.
- The unauthenticated user clicks on the link of any provider on the list.

Maximum workload test case. 70 concurrent users and 50 of loop count:

Thread Properties

Number of Threads (users):

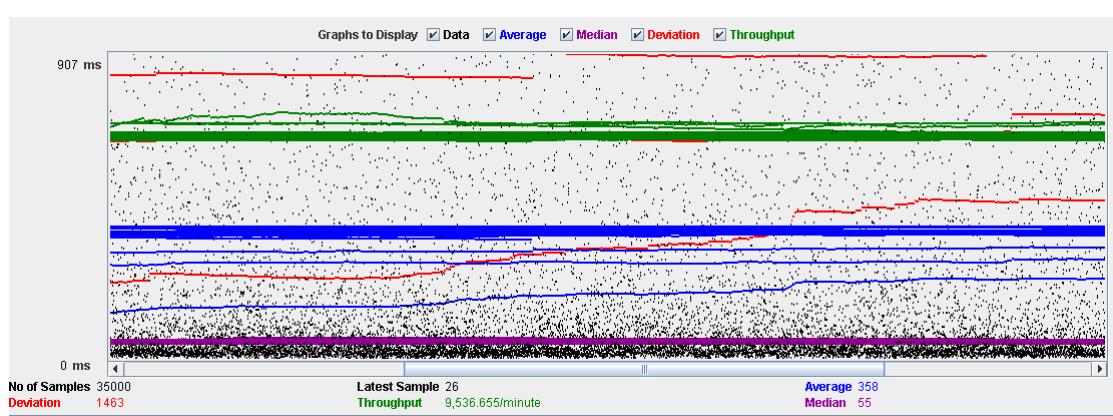
Ramp-Up Period (in seconds):

Loop Count: **Forever**

Delay Thread creation until needed

Scheduler

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3500	1036	409	2155	8	97652	0.00%	15.9/sec	61.5
/styles/jmenu.css	3500	40	26	76	1	1524	0.00%	15.9/sec	29.9
/styles/common.c...	3500	37	25	75	1	966	0.00%	15.9/sec	9.1
/scripts/jmenu.js	3500	38	26	74	2	857	0.00%	15.9/sec	168.1
/scripts/jquery-ui.js	3500	177	112	363	6	2991	0.00%	15.9/sec	7226.4
/scripts/md5-min.js	3500	38	26	73	0	1268	0.00%	15.9/sec	85.8
/scripts/jquery.js	3500	117	73	236	4	2414	0.00%	15.9/sec	4283.1
/styles/displaytag....	3500	36	25	68	1	1485	0.00%	15.9/sec	47.4
/item/allItemsList....	3500	1091	519	2236	21	73270	0.00%	15.9/sec	81.0
/actor/display.do	3500	970	453	2214	12	31534	0.00%	15.9/sec	78.6
TOTAL	35000	358	55	817	0	97652	0.00%	15.9/sec	12063.2



Overload test case: 80 concurrent users and 50 of loop count:

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

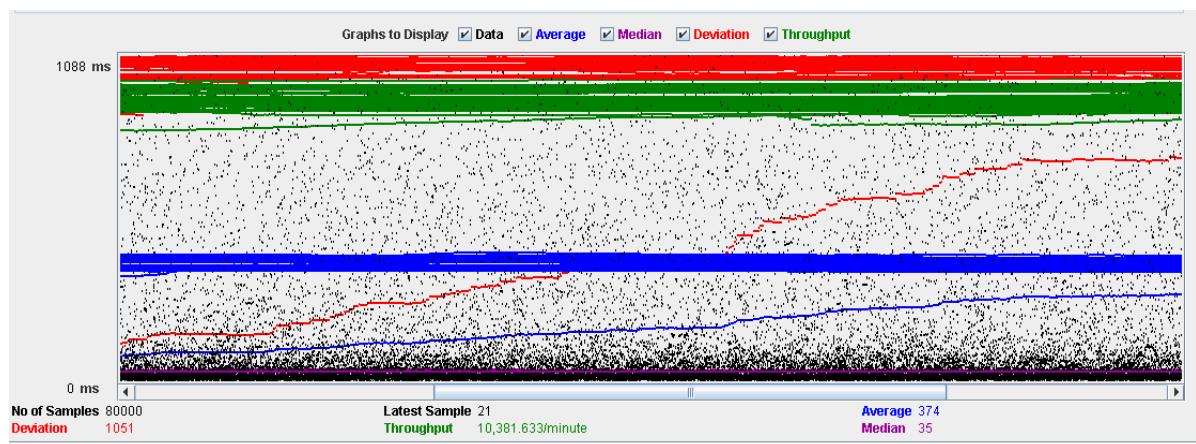
Loop Count: Forever

Delay Thread creation until needed

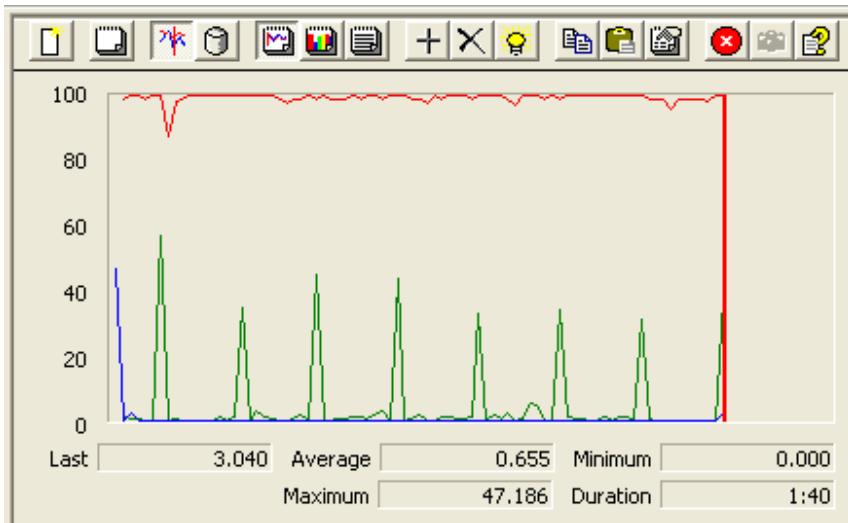
Scheduler

As we can see in the report if we increase the number of threads the 90% line gets a value too high.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4000	1117	581	2781	10	27185	0.00%	18.3/sec	70.9
/styles/jmenu.css	4000	24	19	39	1	325	0.00%	18.3/sec	34.5
/styles/common.c...	4000	22	18	37	1	466	0.00%	18.3/sec	10.5
/scripts/jmenu.js	4000	24	19	39	1	654	0.00%	18.3/sec	193.9
/scripts/jquery-ui.js	4000	83	71	146	7	703	0.00%	18.3/sec	8337.2
/scripts/md5-min.js	4000	23	18	37	1	379	0.00%	18.4/sec	99.0
/scripts/jquery.js	4000	57	48	98	4	877	0.00%	18.4/sec	4942.1
/styles/displaytag....	4000	22	18	36	1	391	0.00%	18.4/sec	54.7
/item/allItemsList....	4000	1173	641	2679	22	20196	0.00%	18.4/sec	93.5
/actor/display.do	4000	1187	625	2687	17	19747	0.00%	18.4/sec	90.8
TOTAL	40000	373	36	1042	1	27185	0.00%	183.3/sec	13912.4



As we can see in the graph below, there is a bottleneck with the CPU. Probably we could improve the maximum workload of the application if we assign more processors to the virtual machine.



Conclusion: The maximum number of concurrent users supported by this test case is 70 and we could improve it by assigning more CPU's resources to our system.

Acme Rookies /Req. 9.3 - An actor who is not authenticated must be able to register to the system as a provider.

Technical details of the computer on which the test has been executed:

- Memory RAM: 8,00 GB
- CPU: Intel Core i7-7700HQ
- Hard Disk: 256 GB SSD + 921 GB HDD
- Interface network: Intel(R) Dual Band Wireless-AC 3168

Test case description: The user goes to Register Actor option menu and then, click on ‘Register provider’. The user fills the form and save it. Then, he/she logs in the system with the new account.

Maximum workload test case: 325 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties

Number of Threads (users):	325
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

With this configuration we make sure that the system doesn't produce any errors and has a good response time.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
actor/registerProvider.do	48750	105	53	257	3	2637	0.00%	109.9/sec	432.8
welcome/index.do	32500	120	65	283	4	2270	0.00%	74.4/sec	711.8
security/login.do	16250	108	54	271	3	2294	0.00%	37.3/sec	145.1
spring_security_check	16250	100	49	251	2	2703	0.00%	37.3/sec	152.8
spring_security_logout	16250	204	134	451	5	3109	0.00%	37.3/sec	156.1
OTAL	146250	115	59	275	3	2612	0.00%	37.3/sec	153.0



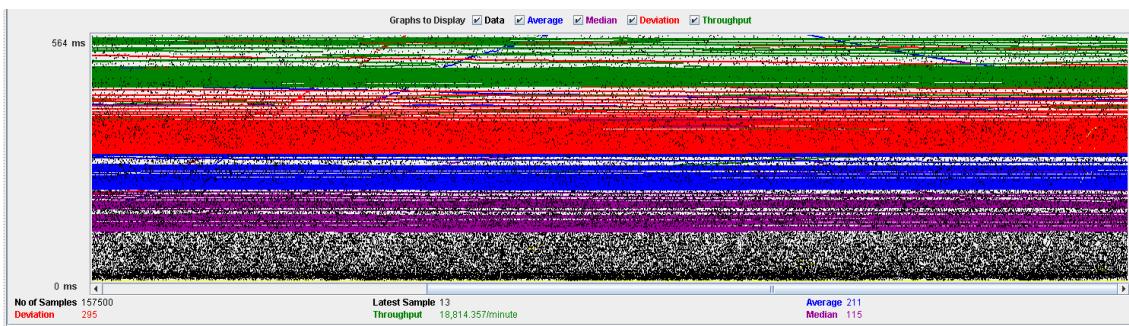
Overload test case: 350 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties	
Number of Threads (users):	350
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

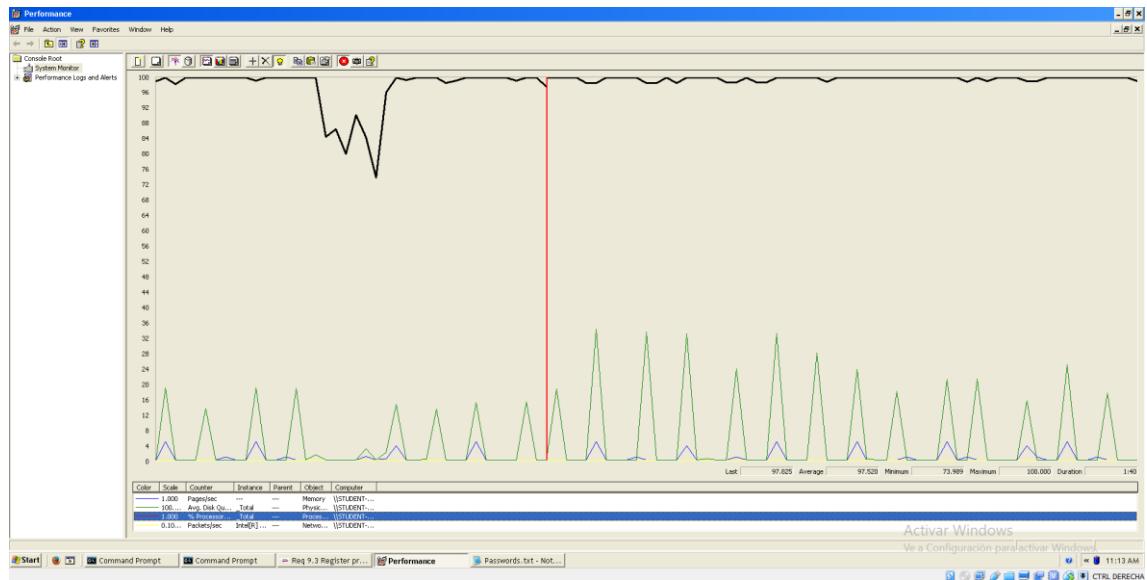
Even though the average time per request is still acceptable, it begins to produce some errors as we can see in the following picture. The errors are always the same: I/O exception (java.net.SocketException) caught when processing request: Connection reset by peer: socket write error.

This exception is not related with the implementation of our application, but with tomcat. The system can't handle this number of concurrent users properly.

Label	#Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
	52500	183	100	444	0	5878	3.25%	104.5/sec	404.4
actor/registerProvider.do	35000	208	114	485	0	4707	3.20%	70.5/sec	657.6
welcome/index.do	17500	199	106	486	0	4334	3.65%	35.4/sec	134.9
security/login.do	17500	175	92	436	0	4594	3.43%	35.4/sec	142.1
!_spring_security_check	17500	371	246	840	0	8193	3.53%	35.4/sec	145.4
!_spring_security_logout	17500	191	107	463	0	5433	3.34%	35.5/sec	142.9
TOTAL	157500	211	115	511	0	8193	3.34%	313.6/sec	1610.3



In the following performance analysis we can appreciate that CPU is running almost 100% long time but there're moments that not, so we could improve our code to try to run CPU 100% all time.



Conclusion: The maximum number of concurrent users supported by the system is 325 because the current CPU is a bottleneck.

Acme Rookies /Req. 10.1 – An actor who is authenticated as a provider must be able to manage his or her catalogue of items, which includes listing, showing, creating, up-dating, and deleting them.

Technical details of the computer on which the test has been executed:

- Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
- CPU: Intel Core i5-4200U
- Disco duro: 240 GB SSD

Test case description:

- The user logs in as a provider.
- Display her/his profile.
- List her/his items.
- Display item.
- The provider closes session.

Maximum workload test case. 120 concurrent users and 50 of loop count:

Thread Properties

Number of Threads (users):

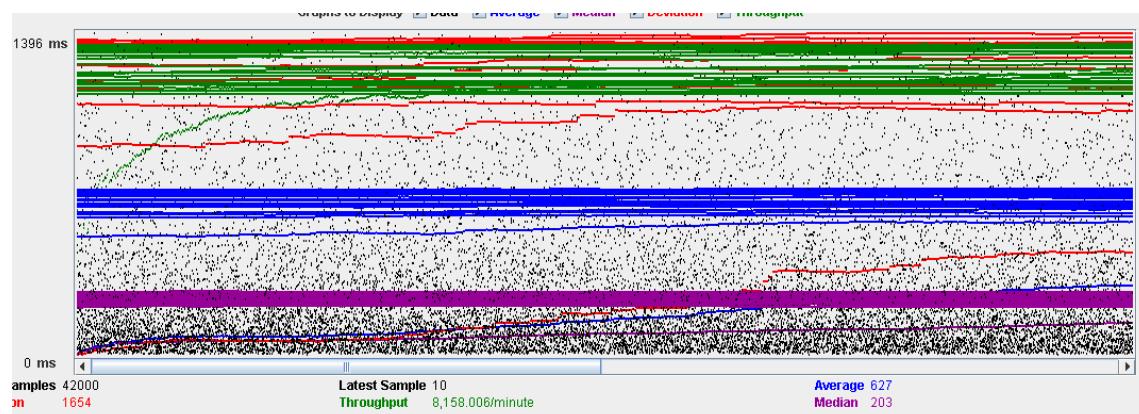
Ramp-Up Period (in seconds):

Loop Count: Forever

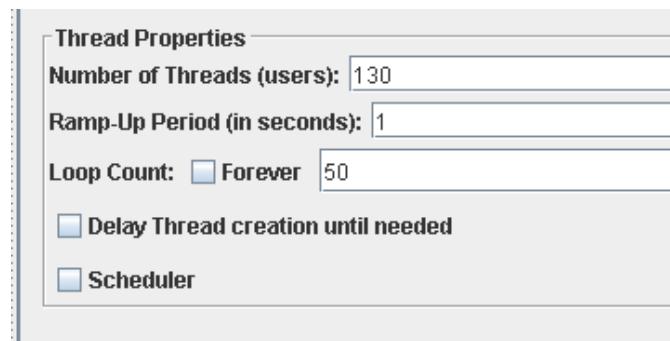
Delay Thread creation until needed

Scheduler

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
security/login.do	6000	523	163	1104	8	28414	0.00%	19.4/sec	79.4
j_spring_security...	6000	1104	448	2274	14	61434	0.00%	19.4/sec	80.7
	12000	540	164	1118	8	60930	0.00%	38.9/sec	151.7
item/allItemsList....	6000	598	210	1160	20	34868	0.00%	19.4/sec	107.0
item/display.do	6000	553	177	1153	10	26543	0.00%	19.4/sec	88.7
j_spring_security...	6000	531	172	1095	11	32209	0.00%	19.4/sec	79.2
TOTAL	42000	627	203	1292	8	61434	0.00%	136.0/sec	584.4



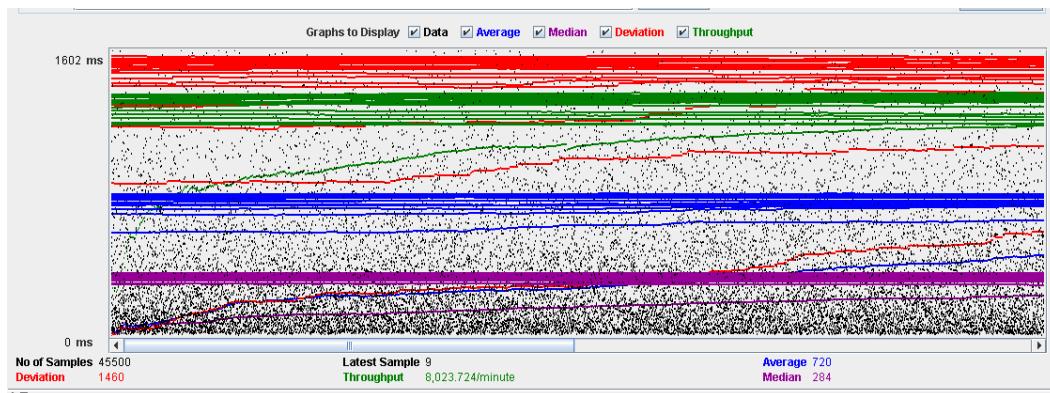
Overload test case: 130 concurrent users and 50 of loop count:



As we can see in the report if we increase the number of threads the 90% line gets a value too high.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	6500	598	230	1310	8	21085	0.00%	19.1/sec	78.1
/j_spring_security...	6500	1234	624	2685	14	25167	0.00%	19.1/sec	79.4
/	13000	624	236	1389	7	24658	0.00%	38.2/sec	149.3
/item/allItemsList...	6500	647	280	1362	20	23383	0.00%	19.1/sec	105.2
/item/display.do	6500	678	237	1423	11	25448	0.00%	19.1/sec	85.3
/j_spring_security...	6500	634	249	1358	10	24391	0.00%	19.1/sec	77.9
TOTAL	45500	720	284	1574	7	25448	0.00%	133.7/sec	574.8

As we can see in the graph below, there is a bottleneck with the CPU. Probably we could improve the maximum workload of the application if we assign more processors to the virtual machine.





Conclusion: The maximum number of concurrent users supported by this test case is 120 and we could improve it by assigning more CPU's resources to our system.

Acme Rookies /Req. 10.1 – An actor who is authenticated as a provider must be able to manage his or her catalogue of items, which includes listing, showing, creating, up-dating, and deleting them.

Technical details of the computer on which the test has been executed:

- Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
- CPU: Intel Core i5-4200U
- Disco duro: 240 GB SSD

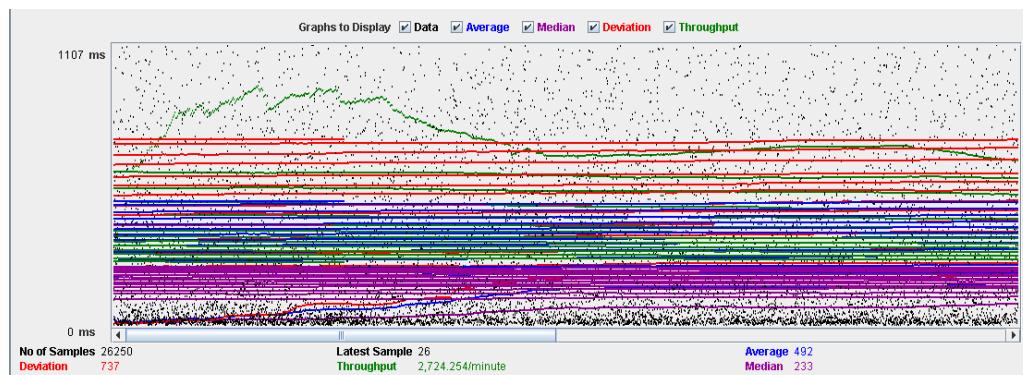
Test case description:

- The user logs in as a provider.
- Display her/his profile.
- List her/his items.
- Create new item.
- Delete item
- The provider closes session.

Maximum workload test case. 25 concurrent users and 50 of loop count:

Thread Properties	
Number of Threads (users):	25
Ramp-Up Period (in seconds):	1
Loop Count:	<input type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	3750	517	291	1192	8	6564	0.00%	6.5/sec	25.2
/scripts/md5-min.js	1250	35	20	77	1	487	0.00%	2.2/sec	11.7
/styles/jmenu.css	1250	33	20	75	1	364	0.00%	2.2/sec	4.1
/scripts/jquery-ui.js	1250	130	88	286	7	1050	0.00%	2.2/sec	985.1
/scripts/jquery.js	1250	88	56	211	5	686	0.00%	2.2/sec	583.9
/scripts/jmenu.js	1250	35	21	78	2	425	0.00%	2.2/sec	22.9
/styles/displaytag....	1250	35	20	78	1	649	0.00%	2.2/sec	6.5
/styles/common.c....	1250	37	20	89	1	505	0.00%	2.2/sec	1.2
/security/login.do	1250	459	244	1107	10	7619	0.00%	2.2/sec	8.9
/j_spring_security...	1250	987	679	2221	33	10182	0.00%	2.2/sec	9.0
/actor/display.do	1250	552	338	1235	18	7340	0.00%	2.2/sec	12.3
/item/list.do	3750	732	526	1510	28	9314	0.00%	6.5/sec	42.0
/item/provider/cre...	1250	486	269	1133	13	11077	0.00%	2.2/sec	11.2
/item/provider/edit...	3750	1071	761	2295	25	10712	0.00%	6.5/sec	32.8
/j_spring_security...	1250	497	308	1096	28	5292	0.00%	2.2/sec	8.8
TOTAL	26250	492	233	1287	1	11077	0.00%	45.4/sec	1761.0



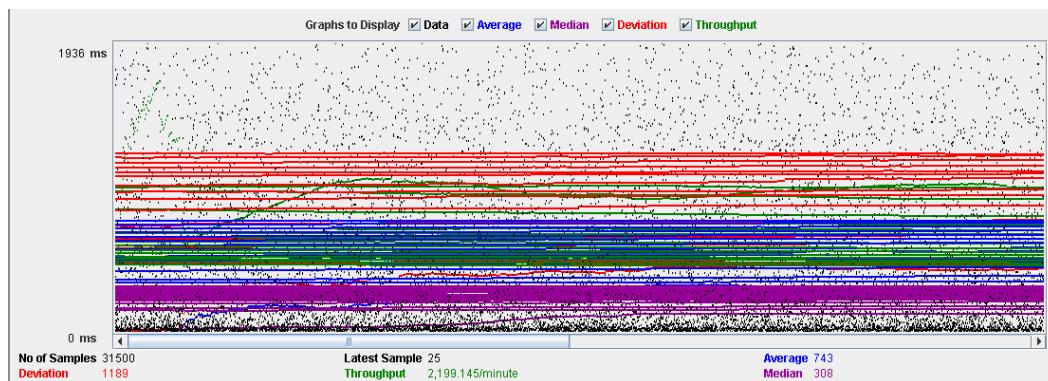
Overload test case: 30 concurrent users and 50 of loop count:

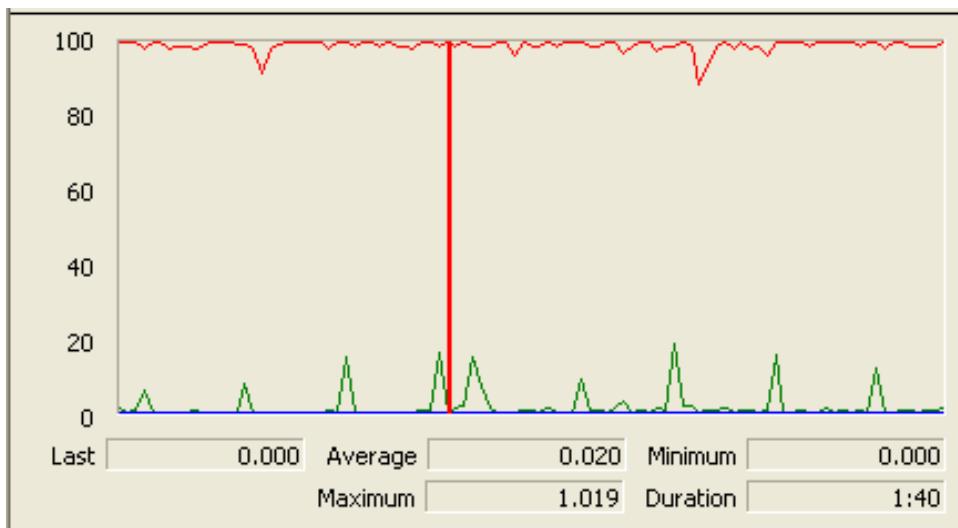
Thread Properties	
Number of Threads (users):	30
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

As we can see in the report if we increase the number of threads the 90% line gets a value too high.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4500	772	402	1942	10	14748	0.00%	5.2/sec	20.4
/scripts/md5-min.js	1500	35	21	79	1	720	0.00%	1.7/sec	9.4
/styles/jmenu.css	1500	35	20	79	1	697	0.00%	1.7/sec	3.3
/scripts/jquery-ui.js	1500	139	95	298	7	1153	0.00%	1.7/sec	794.7
/scripts/jquery.js	1500	95	60	215	6	788	0.00%	1.7/sec	471.0
/scripts/jmenu.js	1500	36	21	79	1	685	0.00%	1.7/sec	18.5
/styles/displaytag....	1500	36	20	91	1	720	0.00%	1.7/sec	5.2
/styles/common.c...	1500	35	21	76	1	897	0.00%	1.7/sec	1.0
/security/login.do	1500	766	354	1928	12	10503	0.00%	1.7/sec	7.1
/ji_spring_security...	1500	1616	1090	3510	35	25468	0.00%	1.7/sec	7.3
/actor/display.do	1500	839	465	1947	18	12010	0.00%	1.7/sec	10.0
/item/list.do	4500	1069	718	2268	43	15513	0.00%	5.2/sec	33.9
/item/provider/cre...	1500	762	396	1749	16	9980	0.00%	1.8/sec	9.0
/item/provider/edit...	4500	1629	1136	3588	25	16462	0.00%	5.3/sec	26.5
/ji_spring_security...	1500	804	425	1795	28	13649	0.00%	1.8/sec	7.2
TOTAL	31500	743	308	1972	1	25468	0.00%	36.7/sec	1421.6

As we can see in the graph below, there is a bottleneck with the CPU. Probably we could improve the maximum workload of the application if we assign more processors to the virtual machine.





Conclusion: The maximum number of concurrent users supported by this test case is 25 and we could improve it by assigning more CPU's resources to our system.

Acme Rookies /Req. 10.1 – An actor who is authenticated as a provider must be able to manage his or her catalogue of items, which includes listing, showing, creating, up- dating, and deleting them.

Technical details of the computer on which the test has been executed:

- Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
- CPU: Intel Core i5-4200U
- Disco duro: 240 GB SSD

Test case description:

- The user logs in as a provider.
- Display her/his profile.
- List her/his items.
- Edit item
- The provider closes session.

Maximum workload test case. 120 concurrent users and 50 of loop count:

Thread Properties

Number of Threads (users):

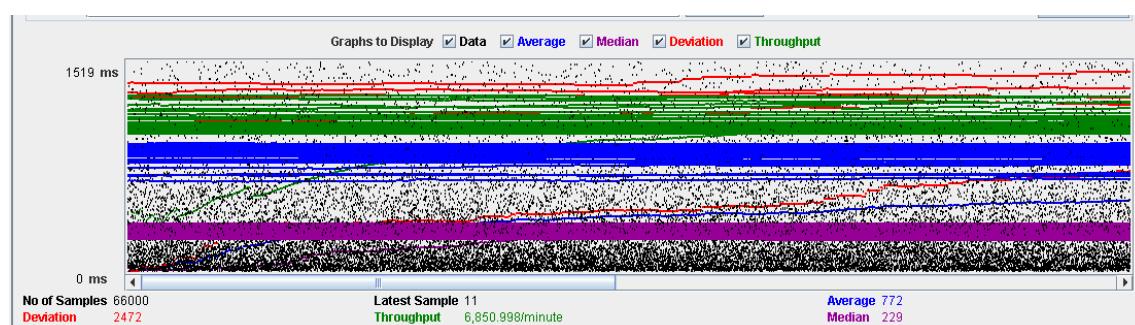
Ramp-Up Period (in seconds):

Loop Count: Forever

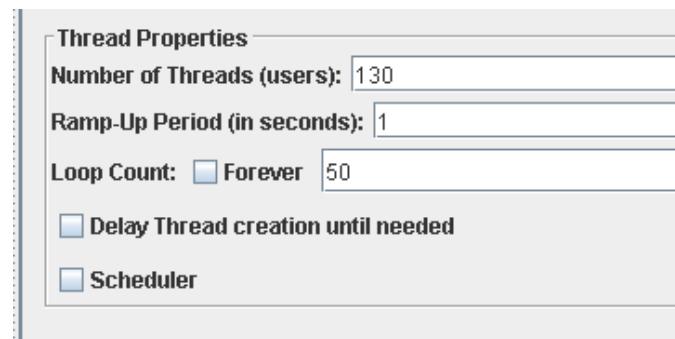
Delay Thread creation until needed

Scheduler

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	18000	644	177	1167	8	72164	0.00%	31.1/sec	121.2
/security/login.do	6000	680	166	1123	7	52314	0.00%	10.4/sec	42.4
/j_spring_security...	6000	1272	478	2410	13	69448	0.00%	10.4/sec	43.1
/actor/display.do	6000	662	206	1220	16	56387	0.00%	10.4/sec	59.3
/item/list.do	12000	651	200	1157	12	45904	0.00%	20.8/sec	101.5
/item/provider/edit...	12000	1004	341	1794	11	70530	0.00%	20.8/sec	111.1
/j_spring_security...	6000	642	175	1123	10	61156	0.00%	10.4/sec	42.5
TOTAL	66000	772	229	1393	7	72164	0.00%	114.2/sec	520.1



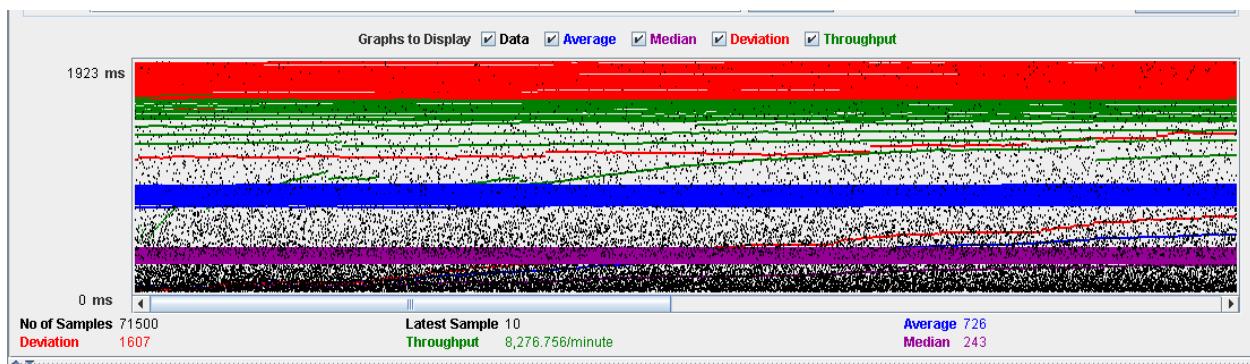
Overload test case: 130 concurrent users and 50 of loop count:

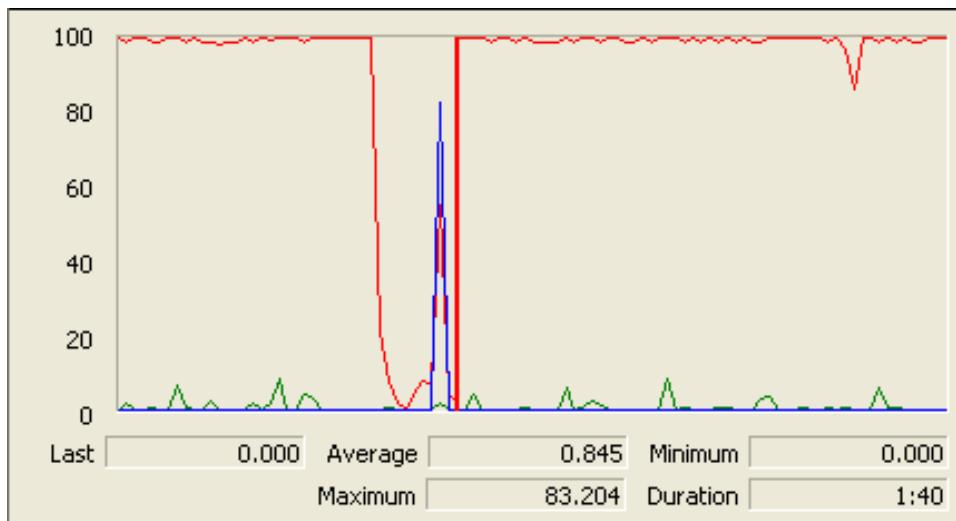


As we can see in the report if we increase the number of threads the 90% line gets a value too high.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	19500	620	191	1410	8	29299	0.00%	37.6/sec	146.4
/security/login.do	6500	598	177	1300	7	29197	0.00%	12.6/sec	51.3
/j_spring_security...	6500	1150	514	2737	14	27377	0.00%	12.6/sec	52.2
/actor/display.do	6500	646	213	1455	17	30832	0.00%	12.6/sec	71.5
/item/list.do	13000	642	209	1434	12	37924	0.00%	25.1/sec	122.4
/item/provider/edit...	13000	912	351	2101	11	30905	0.00%	25.1/sec	134.0
/j_spring_security...	6500	623	192	1406	3	29775	0.00%	12.6/sec	51.2
TOTAL	71500	726	243	1655	3	37924	0.00%	137.9/sec	628.3

As we can see in the graph below, there is a bottleneck with the CPU. Probably we could improve the maximum workload of the application if we assign more processors to the virtual machine.





Conclusion: The maximum number of concurrent users supported by this test case is 120 and we could improve it by assigning more CPU's resources to our system.

Acme Rookies /Req. 13 - An actor who is authenticated as a provider must be able to manage his or her sponsorships, which includes listing, showing, creating, updating, and deleting them.

Technical details of the computer on which the test has been executed:

- Memory RAM: 8,192 (1x) MB, DDR3L RAM (1,600 MHz)
- CPU: Intel Core i7-5500U
- Hard Disk: 1 TB HDD - 5,400 rpm
- Interface network: Gigabit Ethernet LAN - 10BASE-T/100BASE-TX/1000BASE-T

Test case description:

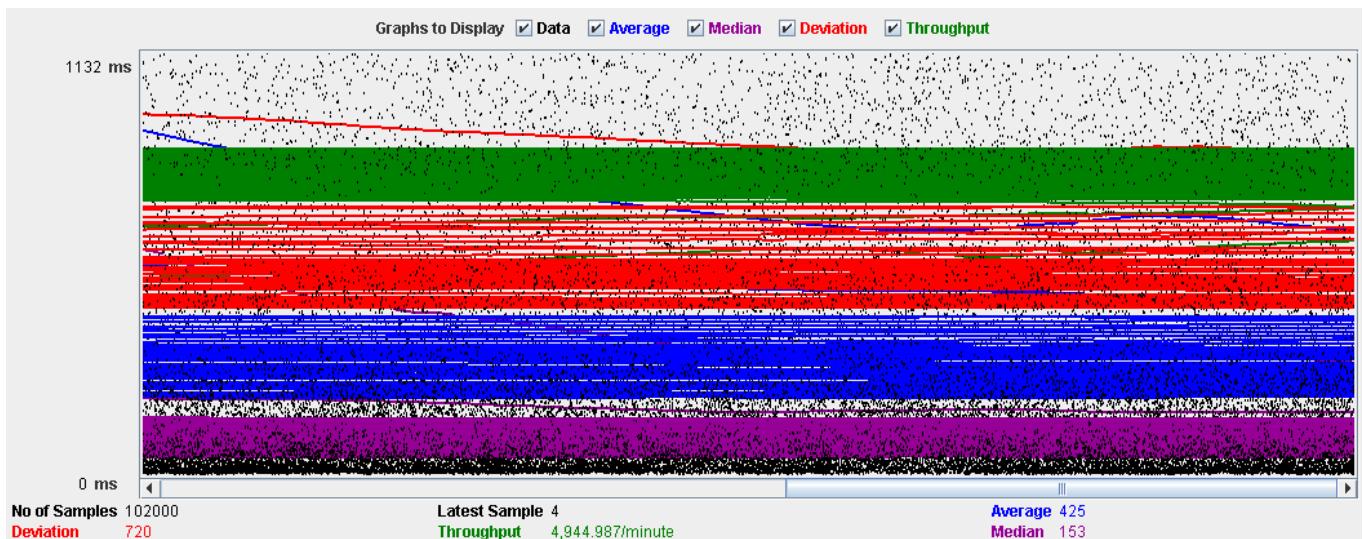
- Log in as provider
- List available positions
- Display one of the positions
- Create a sponsorship in the position displayed
- Edit the sponsorship created
- Delete the sponsorship
- List all sponsorships
- Log out

Maximum workload test case: 120 concurrent users, 50 of loop count and 1 of ramp-up period:

Thread Properties	
Number of Threads (users):	120
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

This is the maximum workload of the test case without any crash or excessive delay. As we can see in the picture below, we don't have any errors and the average time per request is acceptable.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	6000	318	70	958	3	9027	0.00%	4.9/sec	20.4
/j_spring_security_check	6000	511	196	1357	5	9991	0.00%	4.9/sec	20.4
/	12000	240	57	623	4	8331	0.00%	9.7/sec	37.9
/position/availableList.do	6000	339	98	943	8	7676	0.00%	4.9/sec	28.1
/position/display.do	12000	482	244	1183	23	7790	0.00%	9.7/sec	55.0
/sponsorship/provider/create.do	6000	330	92	953	8	10459	0.00%	4.9/sec	30.6
/sponsorship/provider/edit.do	30000	534	200	1506	6	14258	0.00%	24.4/sec	143.3
/sponsorship/provider/display.do	12000	475	198	1214	6	13610	0.00%	9.8/sec	40.4
/sponsorship/provider/list.do	6000	325	139	790	13	10871	0.00%	4.9/sec	33.7
/j_spring_security_logout	6000	333	82	952	7	11180	0.00%	4.9/sec	20.1
TOTAL	102000	425	153	1164	3	14258	0.00%	82.4/sec	425.8



Overload test case: 130 concurrent users, 50 of loop count and 1 of ramp-up period:

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: Forever

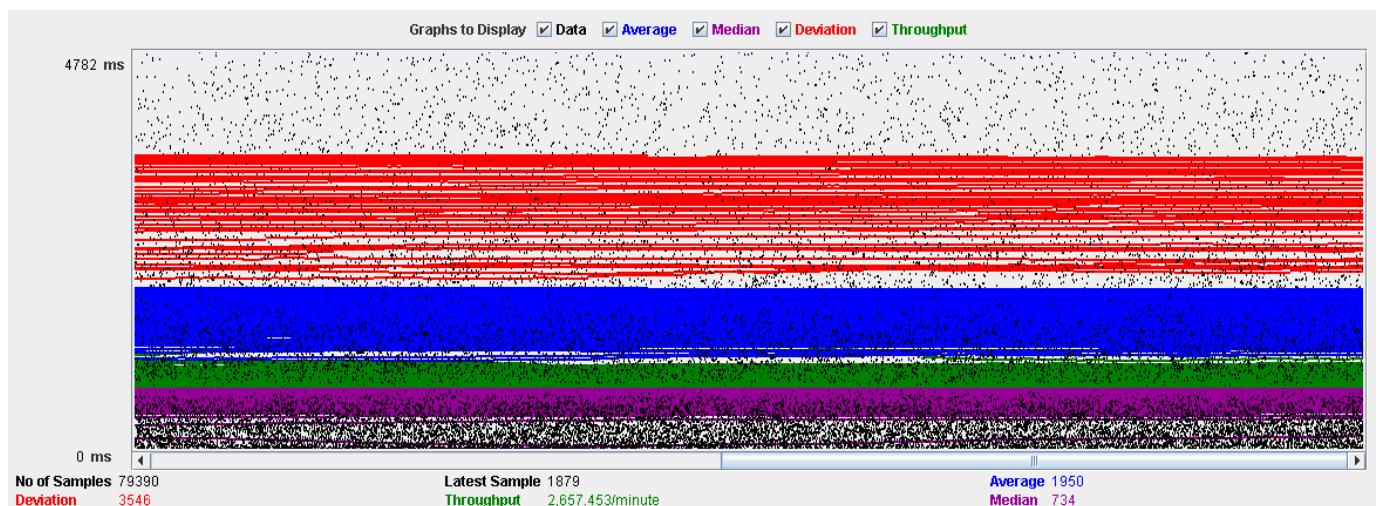
Delay Thread creation until needed

Scheduler

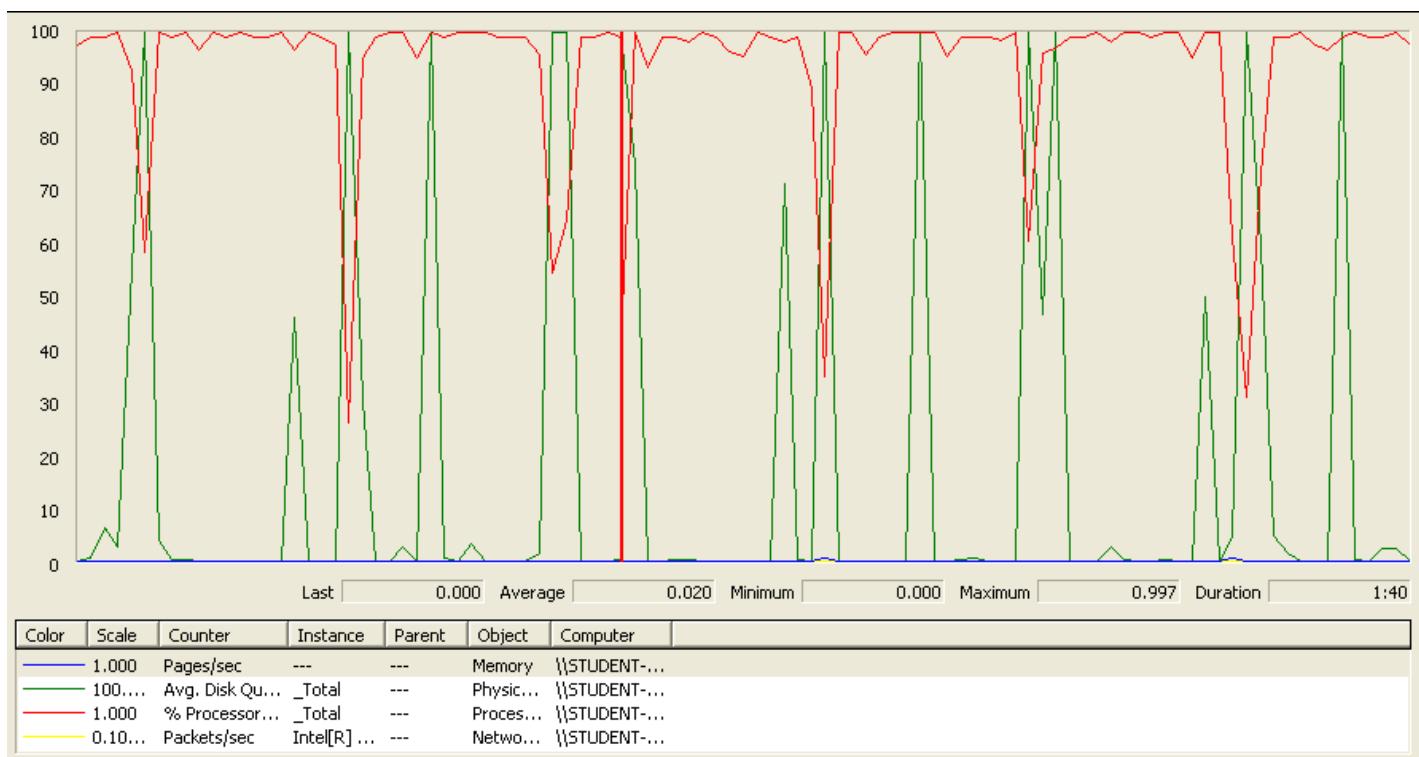
If we analyse the table below, 130 users clearly exceeds the capacity of the system because of two reasons:

- Excessive delay in all requests.
- Even though there are no errors registered, after some time executing the test, Tomcat crash and stop working, blocking MySQL too. It is a very strange behaviour so we cannot dismiss the possibility that it is some kind of bug.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/security/login.do	4732	1392	431	3546	3	38932	0.00%	2.6/sec	11.0
/j_spring_security_...	4725	2523	928	6745	9	58909	0.00%	2.6/sec	11.0
/	9329	1241	344	3239	4	44027	0.00%	5.2/sec	20.3
/position/available...	4712	1328	478	3150	7	33402	0.00%	2.6/sec	15.1
/position/display.do	9324	1808	898	4049	127	38388	0.00%	5.2/sec	29.4
/sponsorship/provi...	4701	1403	454	3197	7	44211	0.00%	2.6/sec	16.4
/sponsorship/provi...	23291	2474	1023	6344	5	74954	0.00%	13.0/sec	76.5
/sponsorship/provi...	9332	2572	948	6981	10	47607	0.00%	5.2/sec	21.6
/sponsorship/provi...	4629	1518	594	3475	99	50013	0.00%	2.6/sec	17.8
/j_spring_security_...	4615	1375	463	3302	7	55137	0.00%	2.6/sec	10.6
TOTAL	79390	1950	734	4979	3	74954	0.00%	44.3/sec	228.8



As we can see in the graph below there is a bottleneck with the CPU and Hard Disk. The CPU is working almost 100% of capacity all the time except some times where the hard disk is collapsed and the CPU don't have any work to do.



Conclusion: The maximum number of concurrent users supported by this test case is 120. We could improve it using a faster hard drive (or with larger queue) and by assigning more CPU's resources to our system. However, we have to pay attention to Tomcat because with 130 users Tomcat crashed. If this crash is caused by some kind of bug, it would be a good idea trying to update Tomcat to a newer version.

Additional Performance Tests

1 – An auditor edits his/her personal data.

Technical details of the computer on which the test has been executed:

- Memory RAM: 8,00 GB
- CPU: Intel Core i7-7700HQ
- Hard Disk: 256 GB SSD + 921 GB HDD
- Interface network: Intel(R) Dual Band Wireless-AC 3168

Test case description: The auditor logs in the system, then, he/she goes to display the profile, select to edit personal information, next the user edits the information and save it. Finally, the user logs out the system.

Maximum workload test case: 350 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: Forever

Delay Thread creation until needed

Scheduler

With this configuration we make sure that the system doesn't produce any errors and has a good response time.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	52500	409	193	1022	3	9319	0.00%	70.9/sec	278.3
/security/login.do	17500	407	189	1011	2	10729	0.00%	23.9/sec	98.0
/_spring_security_check	17500	799	515	1794	7	13385	0.00%	23.9/sec	99.7
/actor/display.do	35000	429	208	1060	5	8910	0.00%	47.6/sec	259.5
/actor/administrator/audit..	35000	665	372	1632	6	12358	0.00%	47.7/sec	309.7
/_spring_security_logout	17500	443	212	1093	4	8982	0.00%	23.9/sec	98.1
TOTAL	175000	507	252	1266	2	13385	0.00%	236.2/sec	1134.5



Overload test case: 375 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

Loop Count: Forever

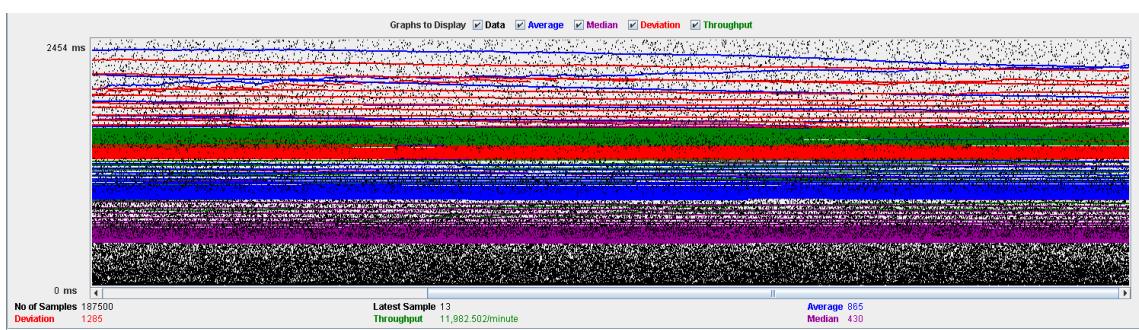
Delay Thread creation until needed

Scheduler

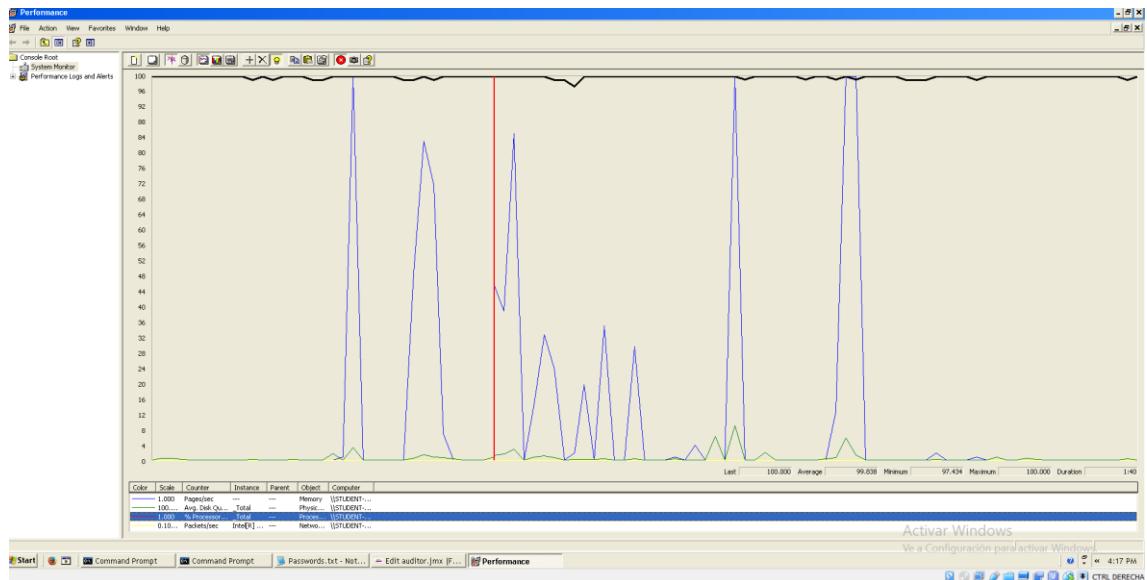
It begins to produce some errors as we can see in the following picture. The errors are always the same: I/O exception (java.net.SocketException) caught when processing request: Connection reset by peer: socket write error.

This exception is not related with the implementation of our application, but with tomcat. The system can't handle this number of concurrent users properly.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	56250	718	334	1777	1	34686	0.13%	59.9/sec	235.3
/security/login.do	18750	715	342	1784	2	18986	0.01%	20.2/sec	82.8
/j_spring_security_check	18750	1440	919	3227	1	20269	0.02%	20.2/sec	84.4
/actor/display.do	37500	734	354	1775	5	33400	0.00%	40.3/sec	219.8
/actor/administrator/audit...	37500	1075	589	2594	3	19013	0.01%	40.4/sec	262.5
/j_spring_security_logout	18750	721	349	1756	2	19882	0.02%	20.3/sec	83.4
TOTAL	187500	865	430	2133	1	34686	0.04%	199.7/sec	959.5



In the following performance analysis we can appreciate that CPU is running almost 100% all time so to try to increase the concurrent users we have to improve the characteristics of the virtual machine.



Conclusion: The maximum number of concurrent users supported by the system is 375 because the current CPU is a bottleneck.

2 – A provider edits his/her personal data.

Technical details of the computer on which the test has been executed:

- Memory RAM: 8,00 GB
- CPU: Intel Core i7-7700HQ
- Hard Disk: 256 GB SSD + 921 GB HDD
- Interface network: Intel(R) Dual Band Wireless-AC 3168

Test case description: The provider logs in the system, then, he/she goes to display the profile, select to edit personal information, next the user edits the information and save it. Finally, the user logs out the system.

Maximum workload test case: 325 concurrent users, 50 of loop count and 1 of ramp-up period.

Thread Properties

Number of Threads (users):

Ramp-Up Period (in seconds):

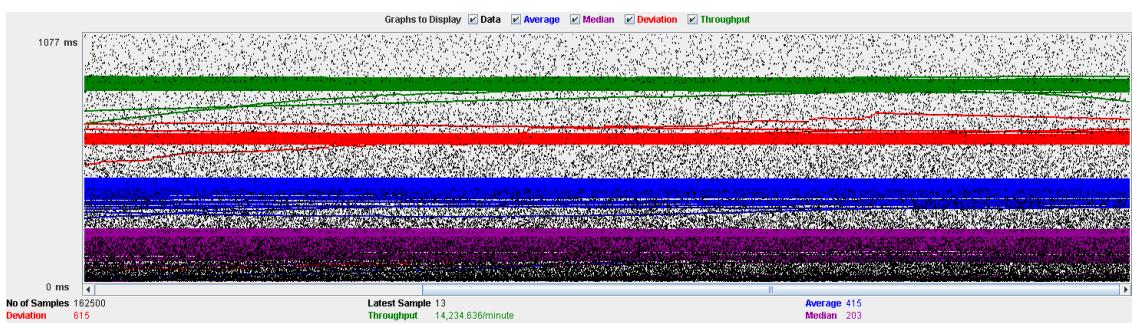
Loop Count: Forever

Delay Thread creation until needed

Scheduler

With this configuration we make sure that the system doesn't produce any errors and has a good response time.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
	49750	332	157	830	2	14897	0.00%	71.2/sec	279.7
security/login.do	16250	330	146	829	3	8822	0.00%	24.1/sec	98.5
_spring_security_check	16250	641	393	1512	4	10603	0.00%	24.0/sec	100.5
vctordisplay.do	32500	357	172	885	5	9500	0.00%	47.8/sec	273.5
vctoradministrator/audit...	32500	558	304	1362	5	11034	0.00%	48.0/sec	323.5
_spring_security_logout	16250	358	172	898	4	12285	0.00%	24.0/sec	98.6
OTAL	162500	415	203	1044	2	14897	0.00%	237.2/sec	1164.6



Overload test case: 350 concurrent users, 50 of loop count and 1 as ramp-up period.

Thread Properties	
Number of Threads (users):	350
Ramp-Up Period (in seconds):	1
Loop Count:	<input checked="" type="checkbox"/> Forever 50
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> Scheduler	

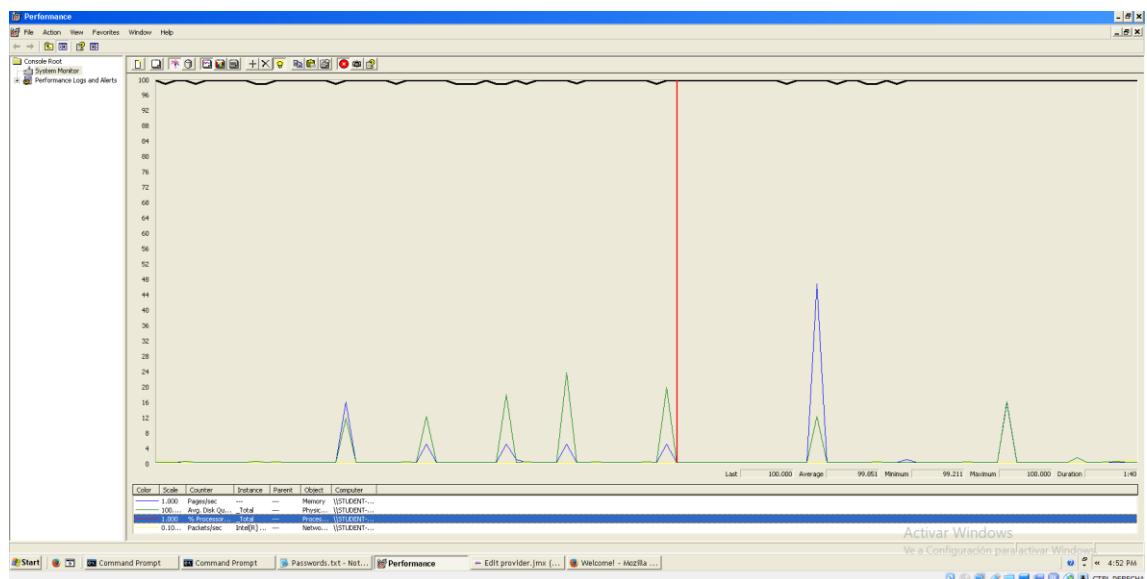
Even though the average time per request is still acceptable, it begins to produce some errors as we can see in the following picture. The errors are always the same: I/O exception (java.net.SocketException) caught when processing request: Connection reset by peer: socket write error.

This exception is not related with the implementation of our application, but with tomcat. The system can't handle this number of concurrent users properly.

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
security/login.do	52500	521	263	1299	1	13349	0.10%	85.2/sec	256.1
spring_security_check	17500	516	263	1284	3	13553	0.02%	22.0/sec	90.1
ctor/display.do	17500	1035	701	2257	1	13583	0.01%	22.0/sec	92.0
ctor/administrator/audit..	35000	542	277	1317	3	17515	0.02%	43.8/sec	250.3
spring_security_logout	17500	535	273	1331	2	10609	0.03%	43.9/sec	296.0
OTAL	175000	634	337	1552	1	17515	0.05%	217.2/sec	1066.7



In the following performance analysis we can appreciate that CPU is running almost 100% all time so to try to increase the concurrent users we have to improve the characteristics of the virtual machine.



Conclusion: The maximum number of concurrent users supported by the system is 325 because the current CPU is a bottleneck.