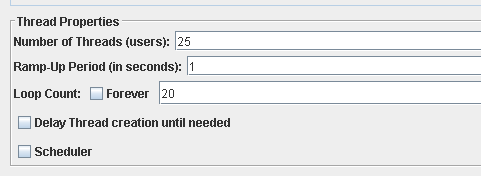
**Req. 10.1** – Manage his or her applications, which includes listing them grouped by status, showing them, creating them, and updating them. When an application is created, the system assigns an arbitrary problem to it (from the set of problems that have been registered for the corresponding position). Updating an application consists in submitting a solution to the corresponding problem (a piece of text with explanations and a link to the code), registering the submission moment, and changing the status to SUBMITTED.

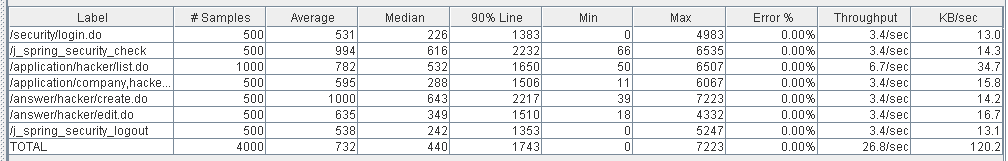
Technical details of the computer on which the test has been executed:

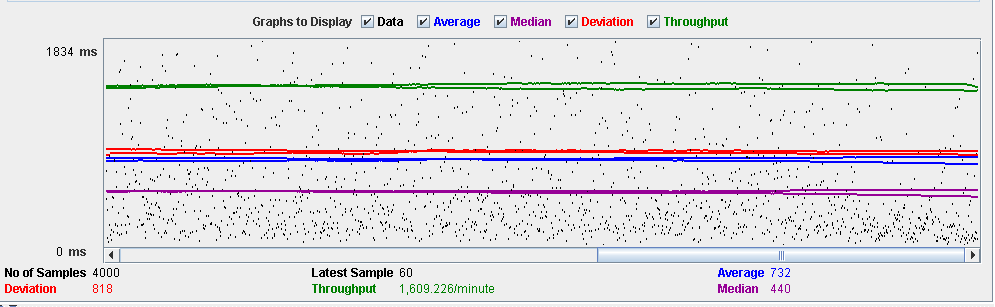
* Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
* CPU: Intel Core i5-4200U
* Disco duro: 240 GB SSD

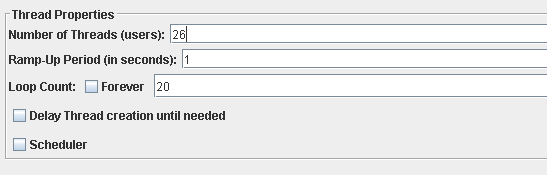
**Test case description**:

* The user logs in as a hacker.
* The hacker accesses the list of own applications.
* The hacker clicks on a pending application.
* The hacker displays the application's display.
* The hacker creates an answer.
* The hacker sends that answer.
* The hacker closes session.

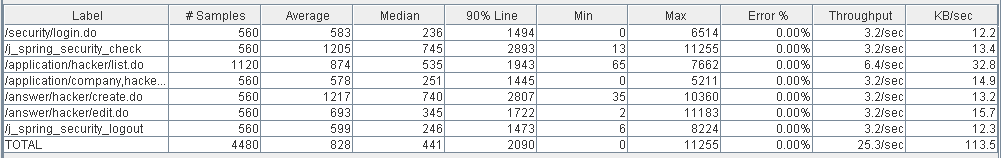
**Maximum workload test case.** 25 concurrent users and 20 of loop count:

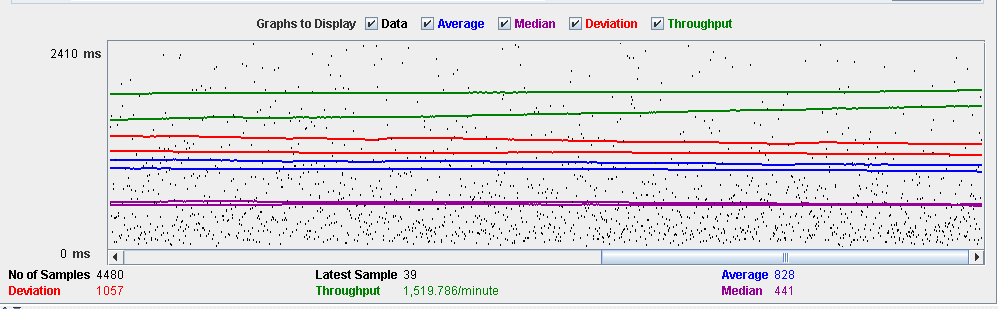




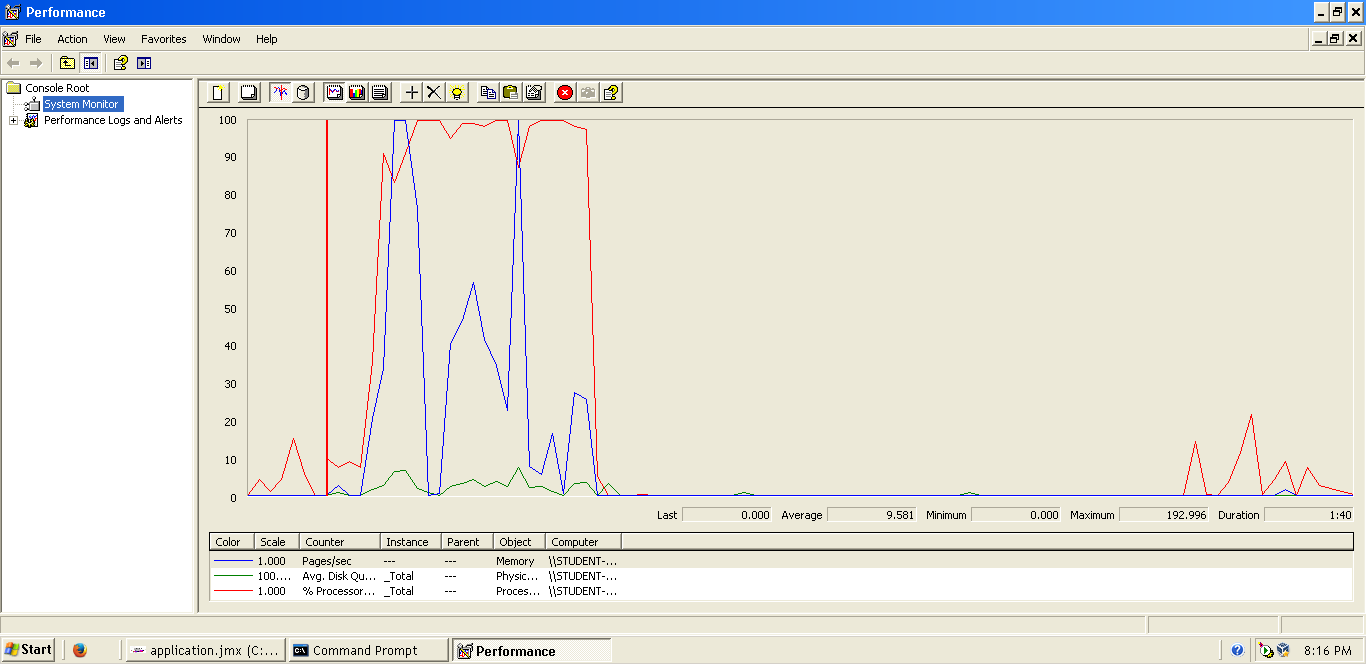
**Overload test case:** 26 concurrent users and 20 of loop count:

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



****

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.

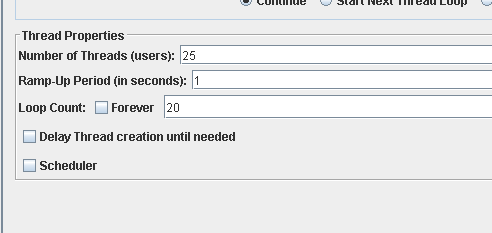
**Req. 10.1** – Manage his or her applications, which includes listing them grouped by status, showing them, creating them, and updating them. When an application is created, the system assigns an arbitrary problem to it (from the set of problems that have been registered for the corresponding position). Updating an application consists in submitting a solution to the corresponding problem (a piece of text with explanations and a link to the code), registering the submission moment, and changing the status to SUBMITTED.

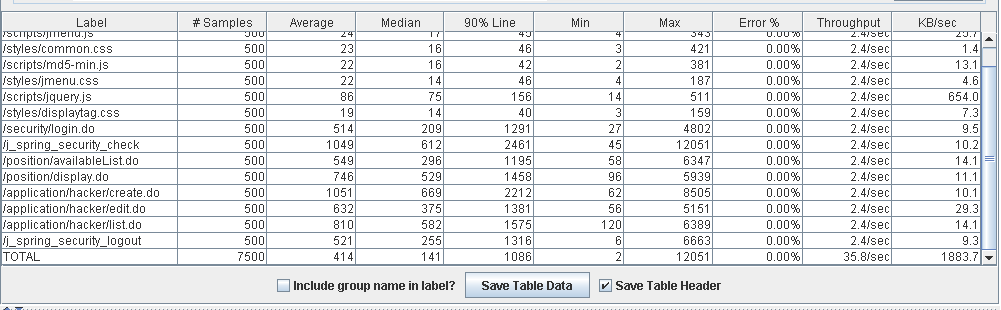
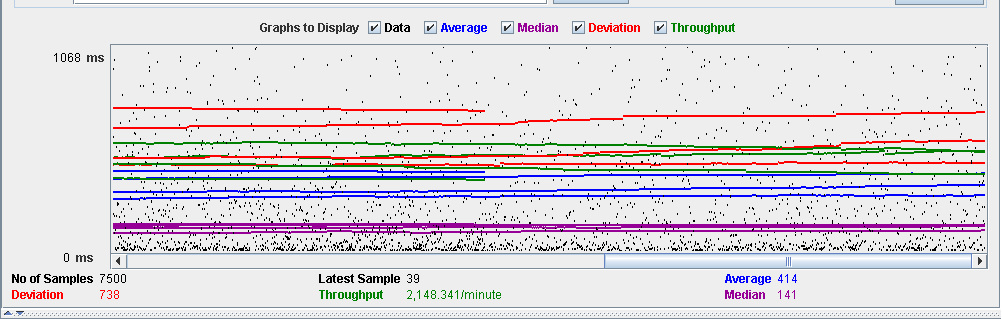
Technical details of the computer on which the test has been executed:

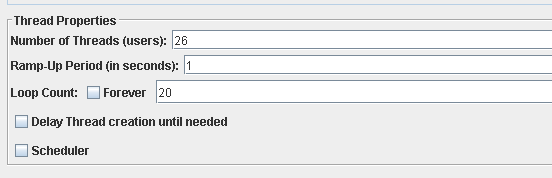
* Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
* CPU: Intel Core i5-4200U
* Disco duro: 240 GB SSD

**Test case description**:

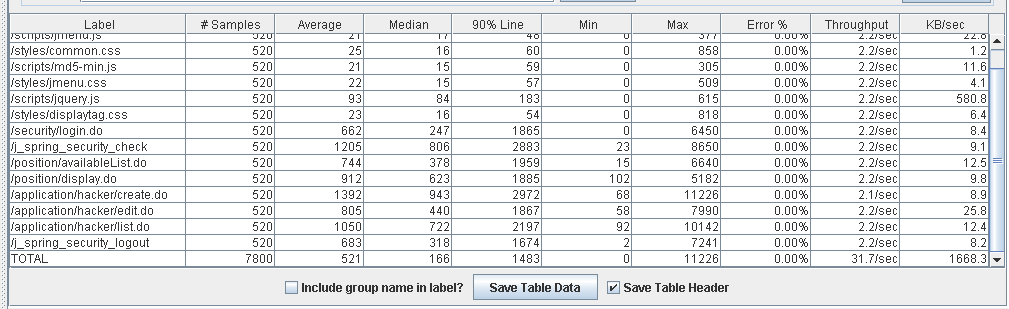
* The user logs in as a hacker.
* The hacker accesses the list of available positions.
* The hacker accesses the display of the chosen position.
* The hacker clicks to request the position.
* The hacker chooses the curriculum and saves.
* The hacker closes session.

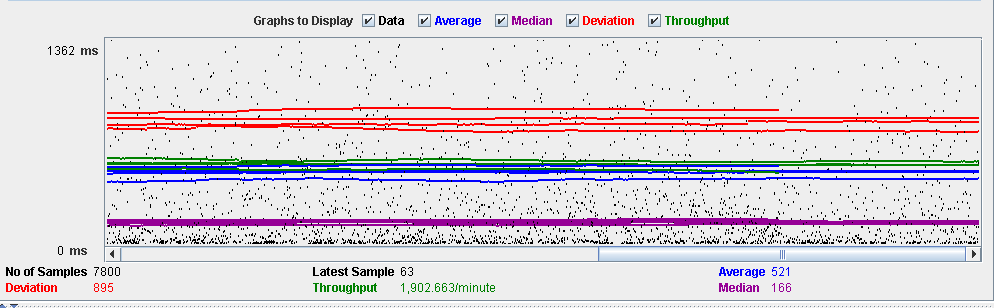
**Maximum workload test case.** 25 concurrent users and 20 of loop count:



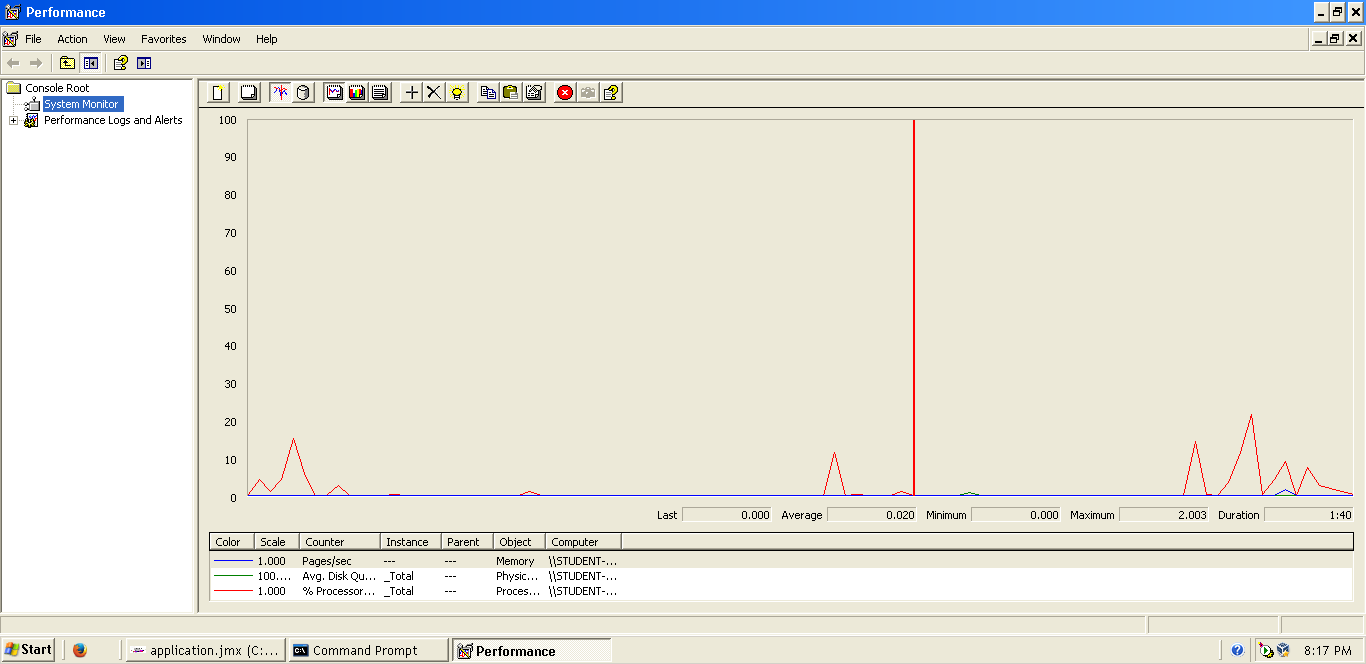
**Overload test case:** 26 concurrent users and 20 of loop count:

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



****

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.

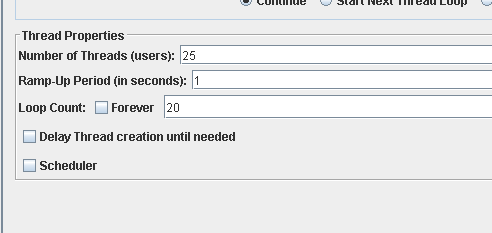
**Req. 9.3** – Manage the applications to their positions, which includes listing them grouped by status, showing them, and updating them. Updating an application amounts to making a decision on it: an application whose status is SUBMITTED may change to status ACCEPTED or REJECTED.

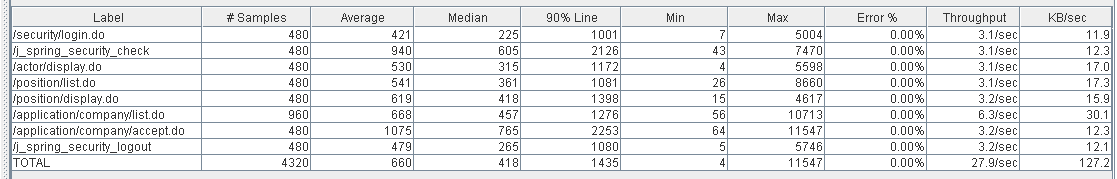
Technical details of the computer on which the test has been executed:

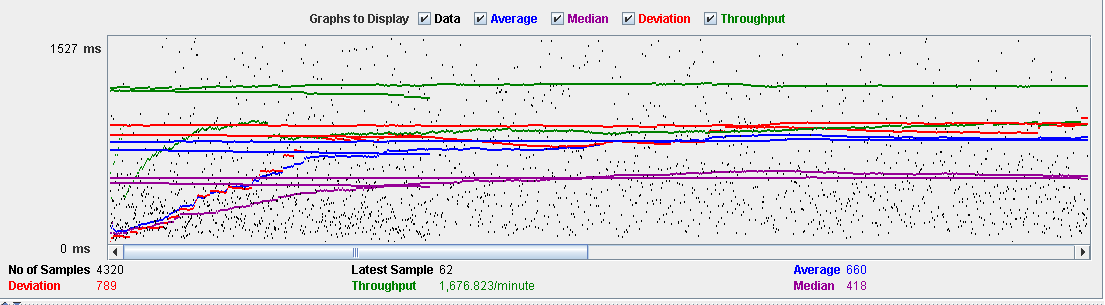
* Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
* CPU: Intel Core i5-4200U
* Disco duro: 240 GB SSD

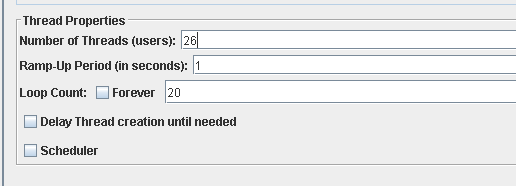
**Test case description**:

* The user logs in as a company.
* Display her/his profile.
* List positions.
* Display position.
* List applications.
* Accepted application.
* The company closes session.

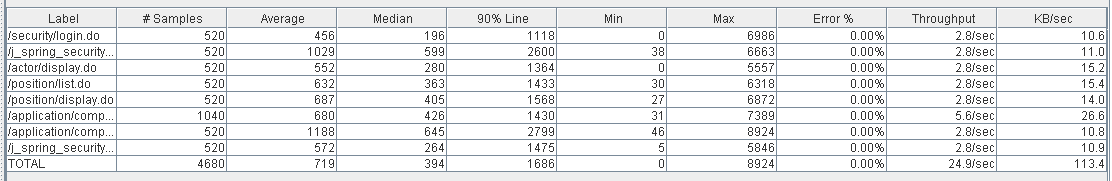
**Maximum workload test case.** 25 concurrent users and 20 of loop count:

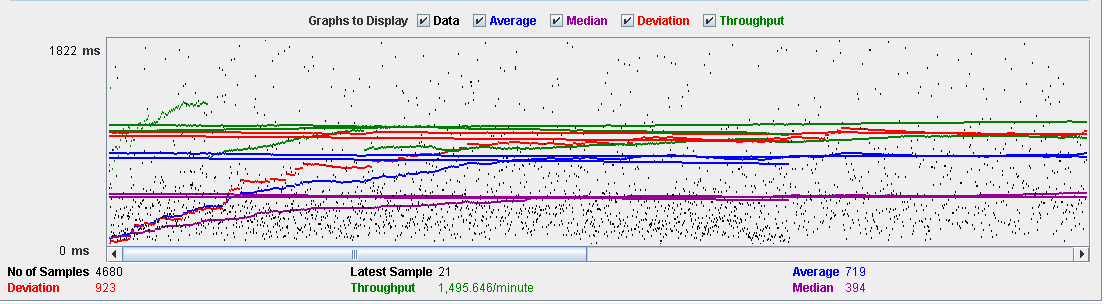




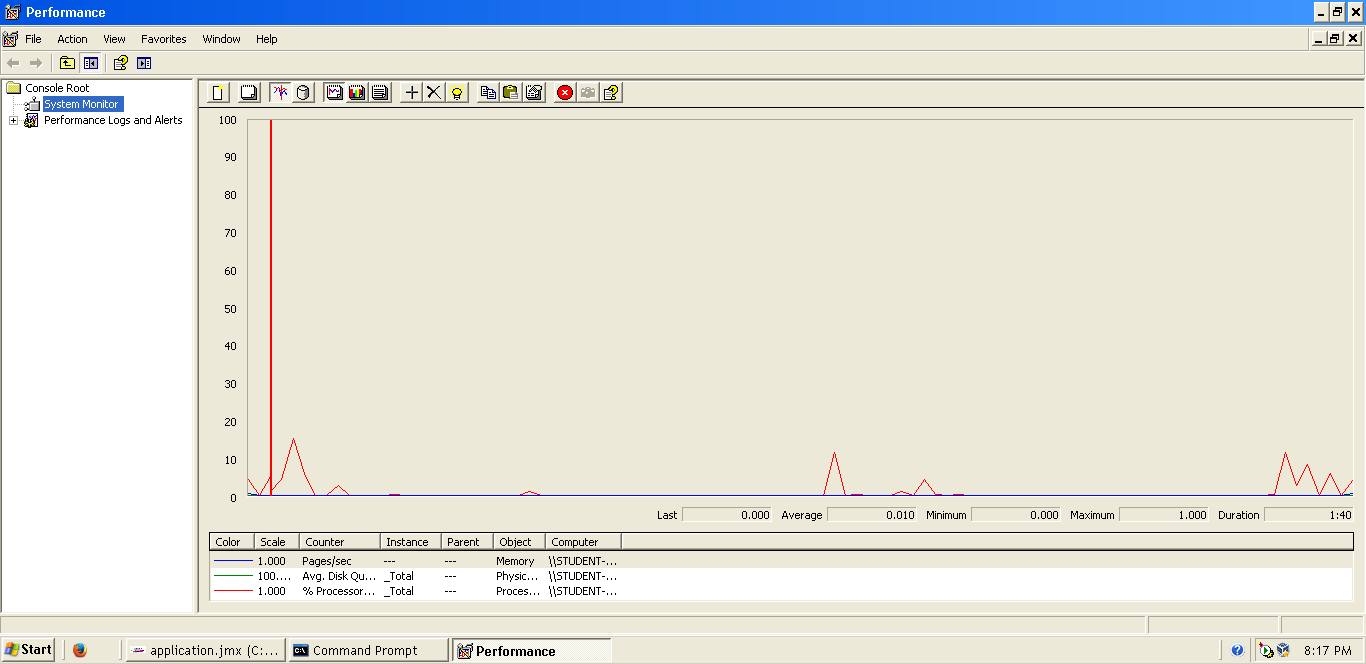
**Overload test case:** 26 concurrent users and 20 of loop count:

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



****

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.

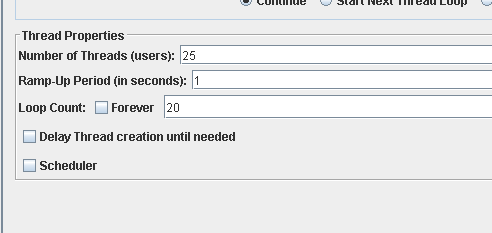
**Req. 9.3** – Manage the applications to their positions, which includes listing them grouped by status, showing them, and updating them. Updating an application amounts to making a decision on it: an application whose status is SUBMITTED may change to status ACCEPTED or REJECTED.

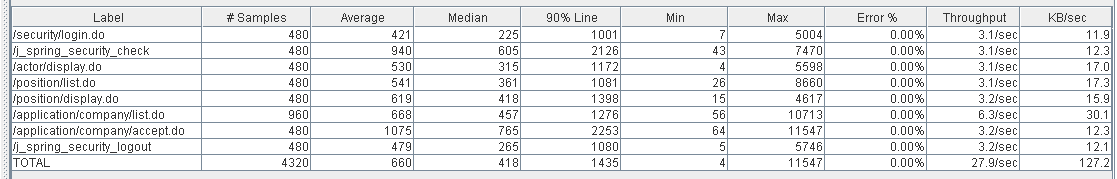
Technical details of the computer on which the test has been executed:

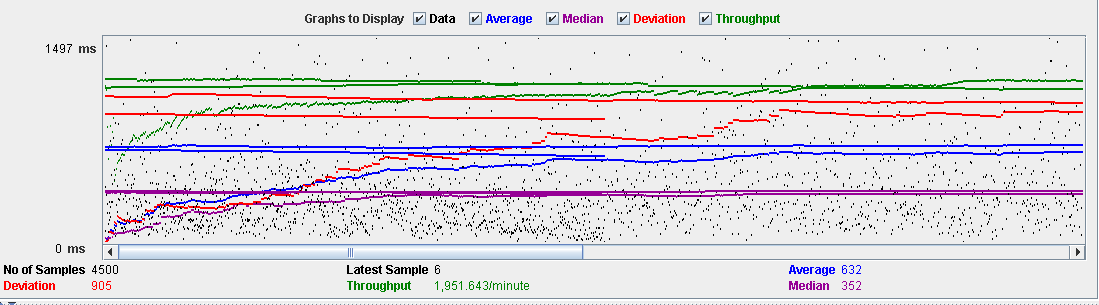
* Ram: 8,0 (1x) GB, DDR3 RAM (1,600 MHz)
* CPU: Intel Core i5-4200U
* Disco duro: 240 GB SSD

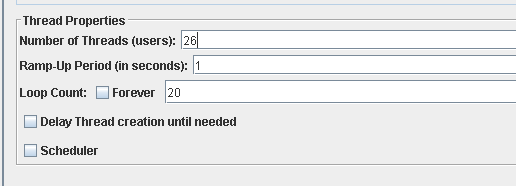
**Test case description**:

* The user logs in as a company.
* Display her/his profile.
* List positions.
* Display position.
* List applications.
* Rejected application.
* The company closes session.

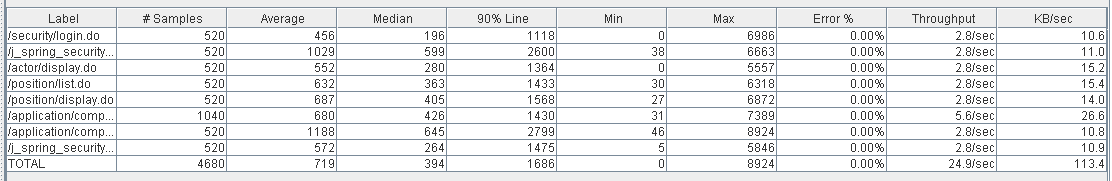
**Maximum workload test case.** 25 concurrent users and 20 of loop count:

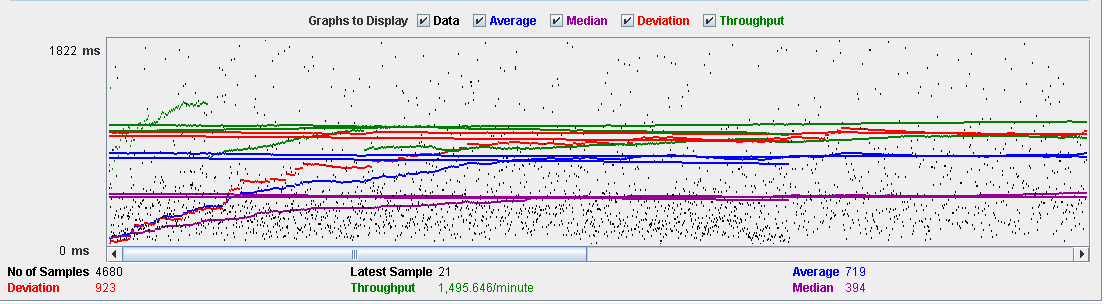




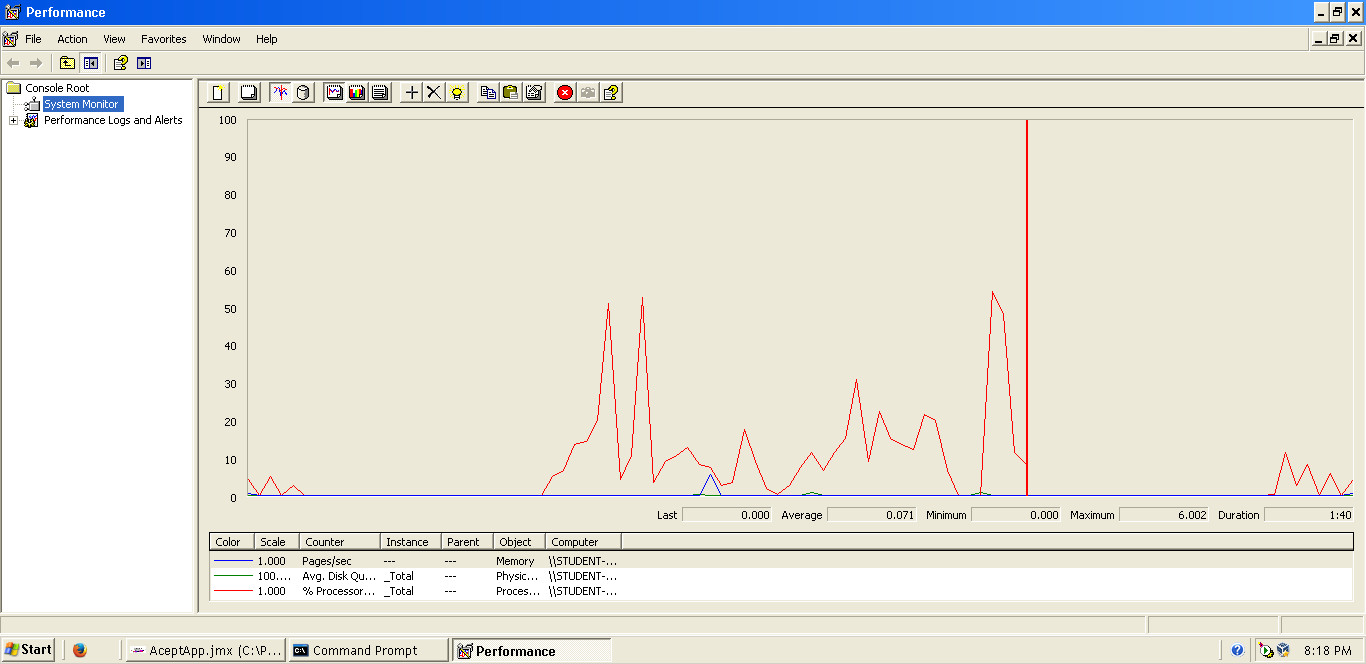
**Overload test case:** 26 concurrent users and 20 of loop count:

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



****

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.