 Ingeniería Informática del Software

**Diseño y Pruebas**

**Informe jMeter – D11: Item 5.**

Barrientos Mohedano, Rubén

Egea Guerrero, Simón

García da Silva, Felipe Javier

Lorenz Rosado, Nicolás

Índice

[Introducción 2](#_Toc482116908)

[Análisis máximo rendimiento 3](#_Toc482116909)

[Register 3](#_Toc482116910)

[Edit personal data 4](#_Toc482116911)

[Browse chorbies 5](#_Toc482116912)

[Browse chorbies who like them 6](#_Toc482116913)

[Like and cancel like 7](#_Toc482116914)

[Edit searchTemplate and browse results 9](#_Toc482116915)

[Write chirp 10](#_Toc482116916)

[Forward chirp 11](#_Toc482116917)

[Reply chirp 12](#_Toc482116918)

[Delete chirp 13](#_Toc482116919)

[Ban and unban chorbi 14](#_Toc482116920)

[Edit cache time 15](#_Toc482116921)

[Change banners 16](#_Toc482116922)

[Dashboards 17](#_Toc482116923)

[Browse available events 18](#_Toc482116924)

[Browse all events 19](#_Toc482116925)

[Manage events 20](#_Toc482116926)

[Register to an event 21](#_Toc482116927)

[Un-register to an event 22](#_Toc482116928)

[Browse registered events 23](#_Toc482116929)

[Change the fee 24](#_Toc482116930)

[Update chorbies monthly fees 25](#_Toc482116931)

[Broadcast 26](#_Toc482116932)

[Browse chorbies who like me 27](#_Toc482116933)

[Conclusiones 29](#_Toc482116934)

# Introducción

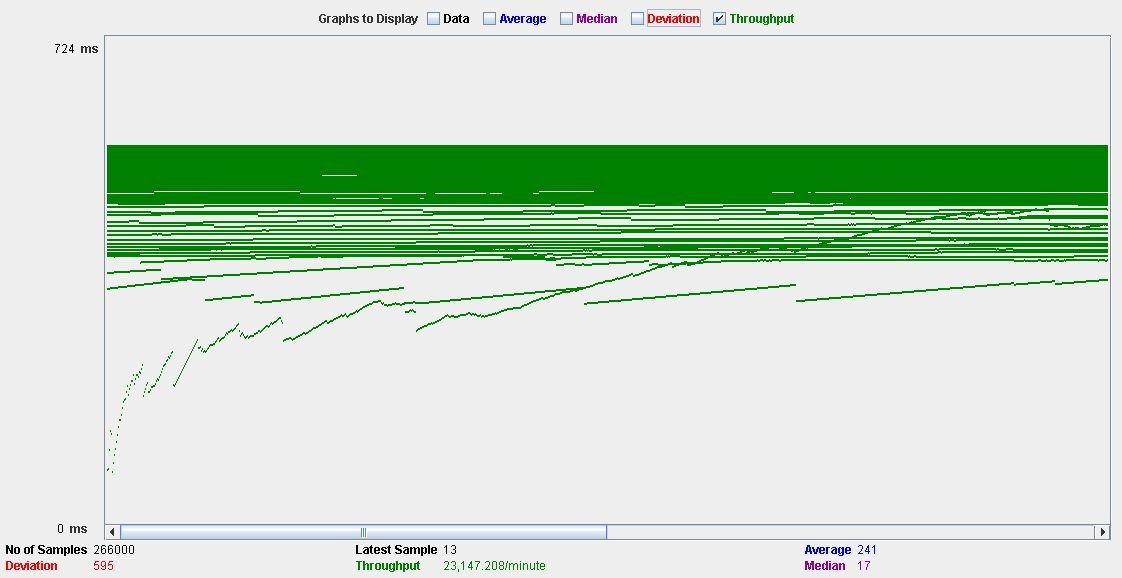
En las próximas páginas se mostrará un análisis de las pruebas realizadas con la herramienta jMeter sobre nuestro proyecto.

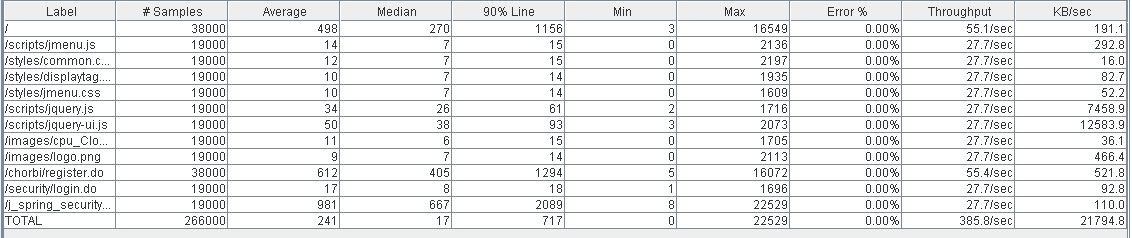
Se destacará cual es el máximo rendimiento del sistema y se mostrarán capturas que lo confirmen.

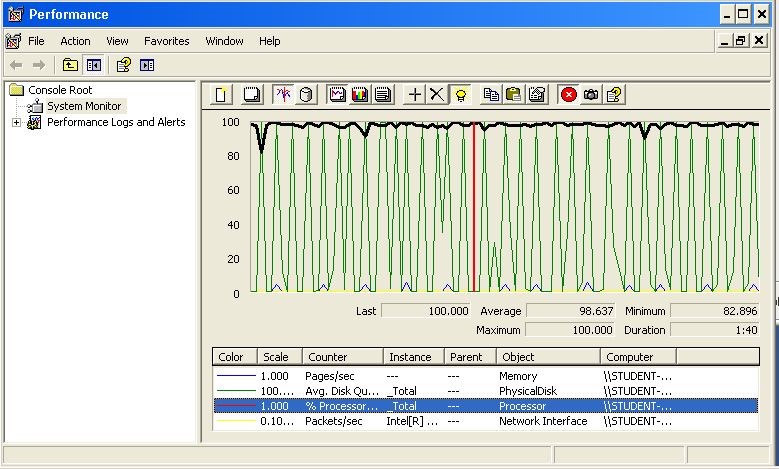
# Análisis máximo rendimiento

Mediante el uso de JMeter hemos obtenido un script para cada caso de uso. Para cada uno de ellos se han realizado pruebas primero comenzando con 10,20,50,100,150 y 200 usuarios. Al usar una carga de 200 usuarios ya comenzaba a dar fallos en la mayoría de casos de uso, pero al reducirlo a 190 aun funcionaban todos de forma correcta. A continuación, vamos a ver las capturas de cada caso de uso con una carga máxima de 190 usuarios.

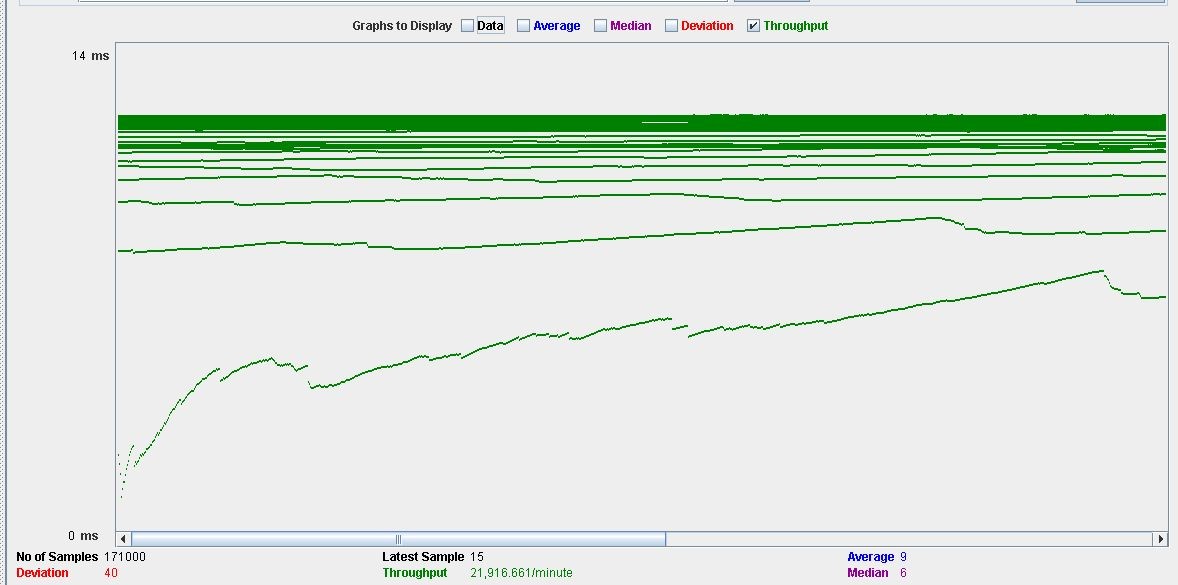
# Register

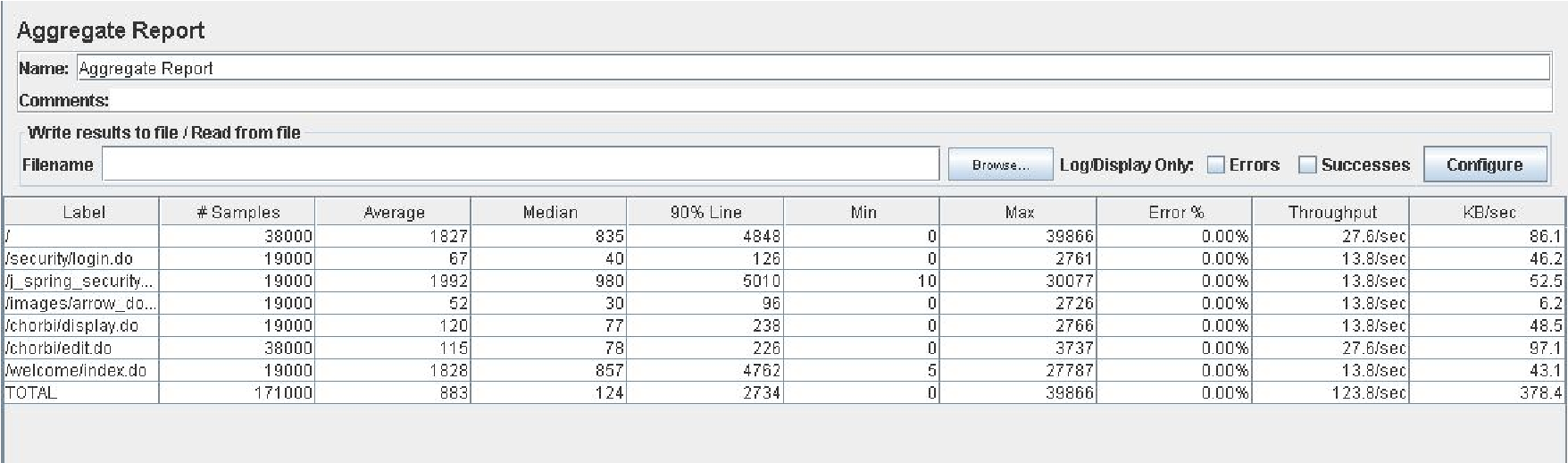






# Edit personal data

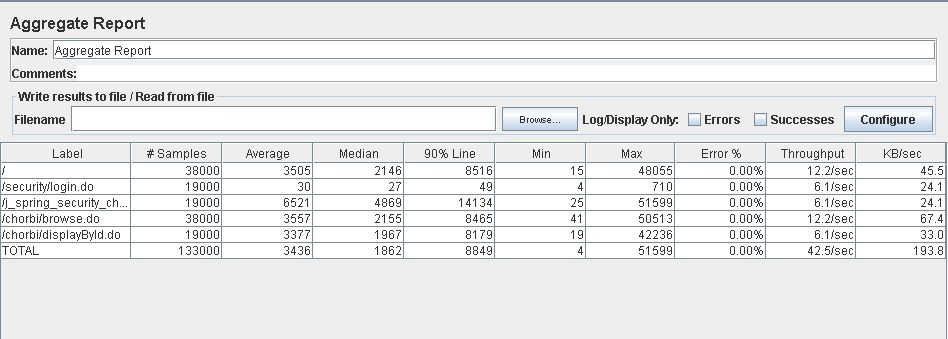


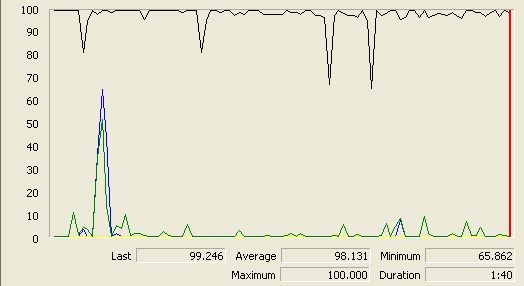




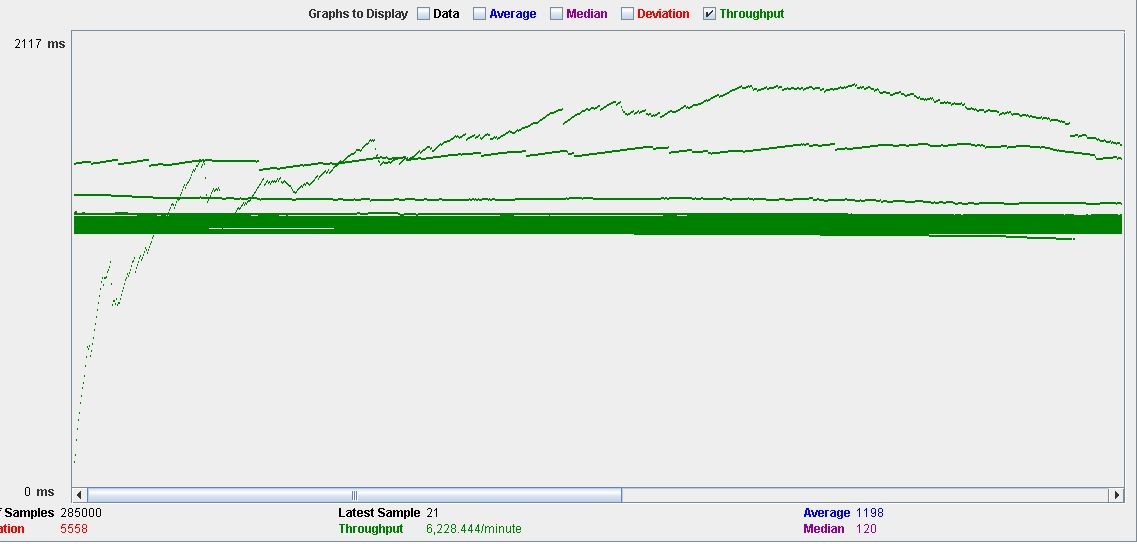
# Browse chorbies

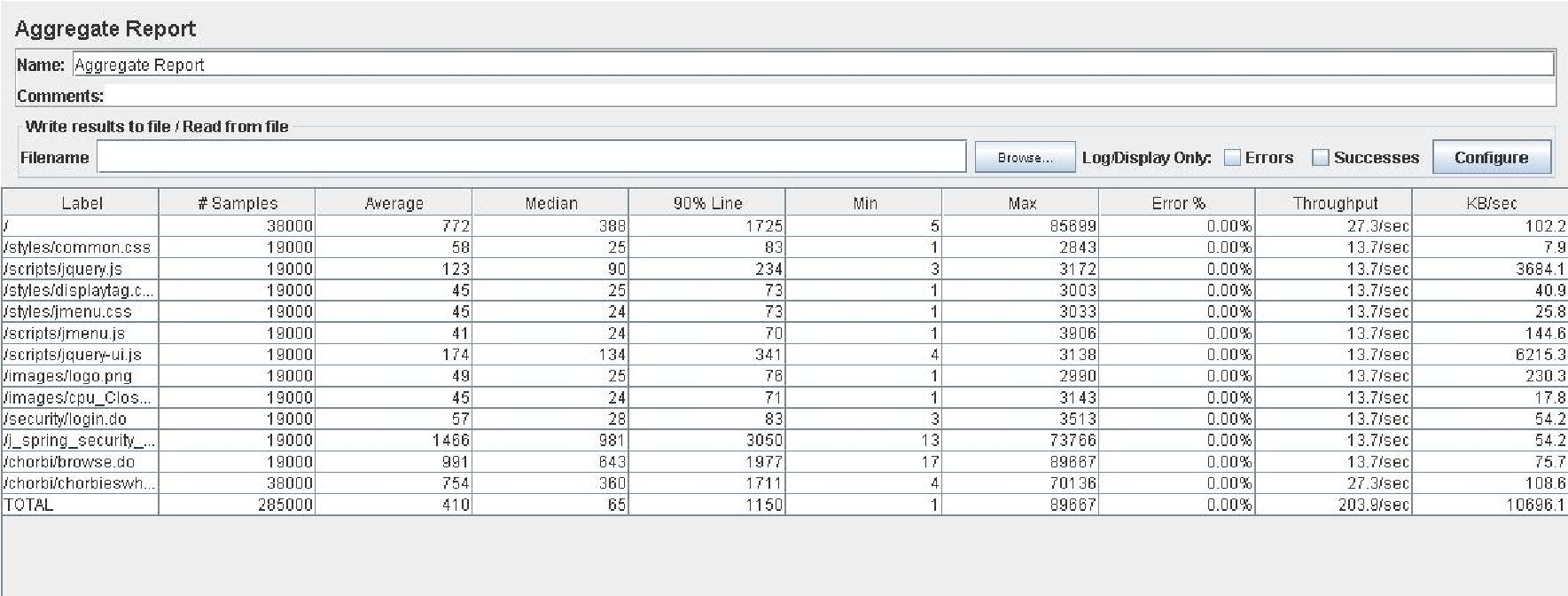


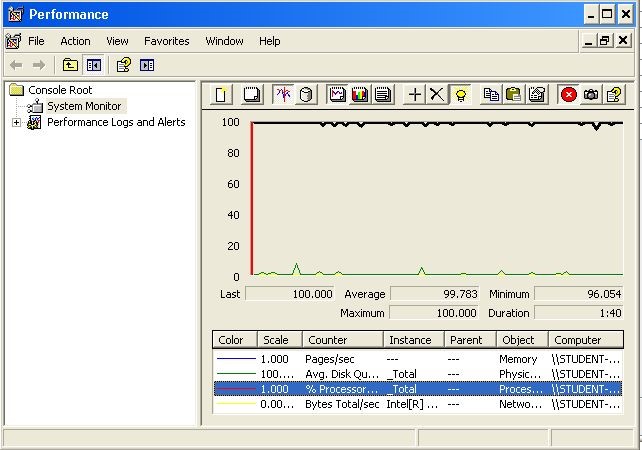




# Browse chorbies who like them

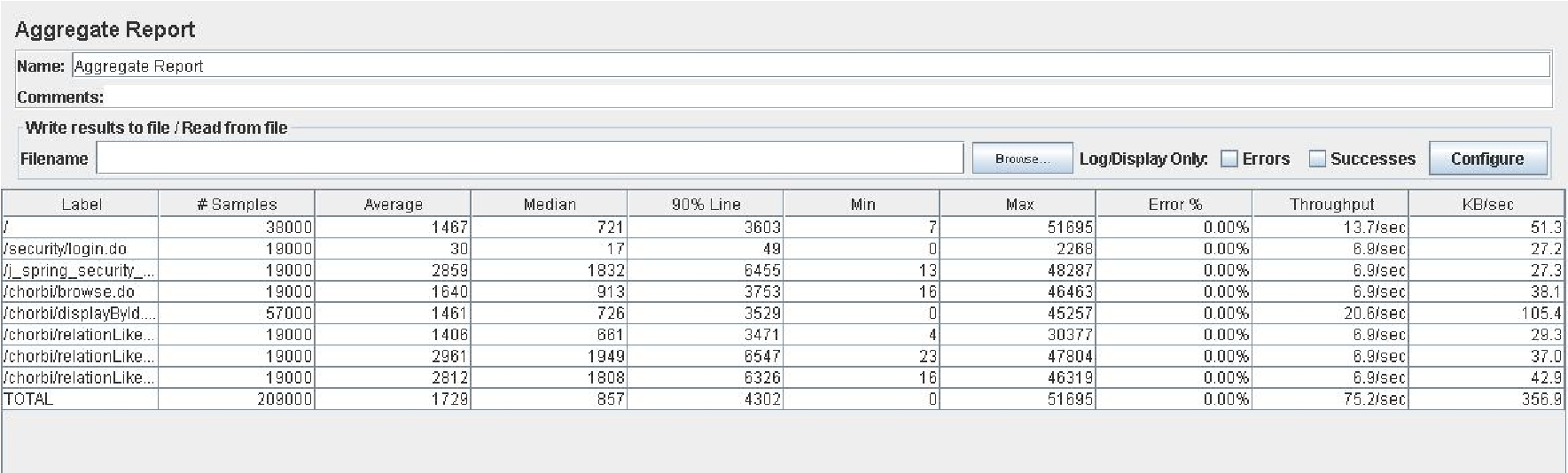


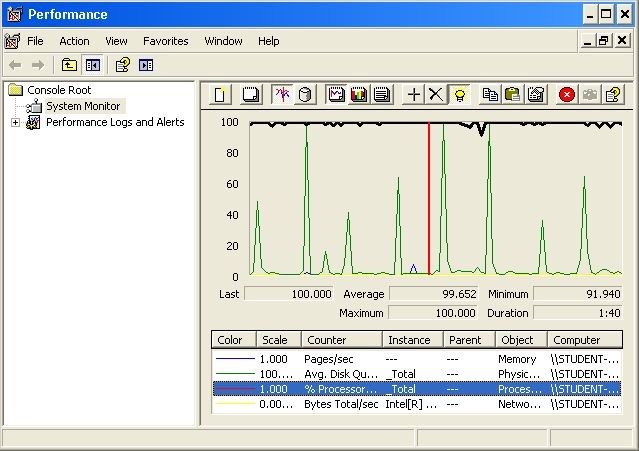




# Like and cancel like

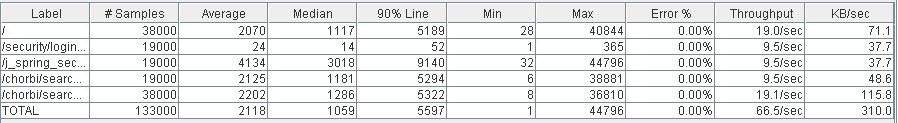


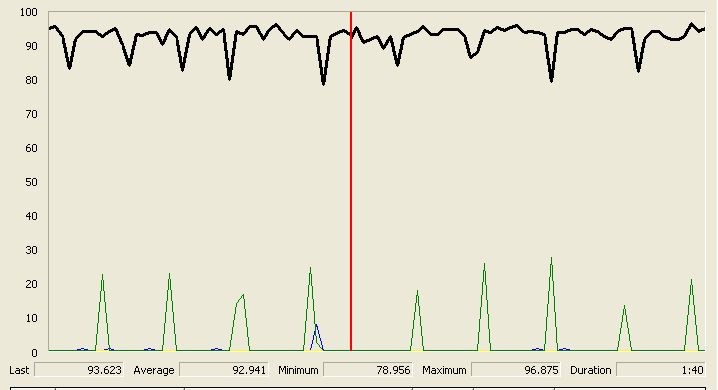




# Edit searchTemplate and browse results



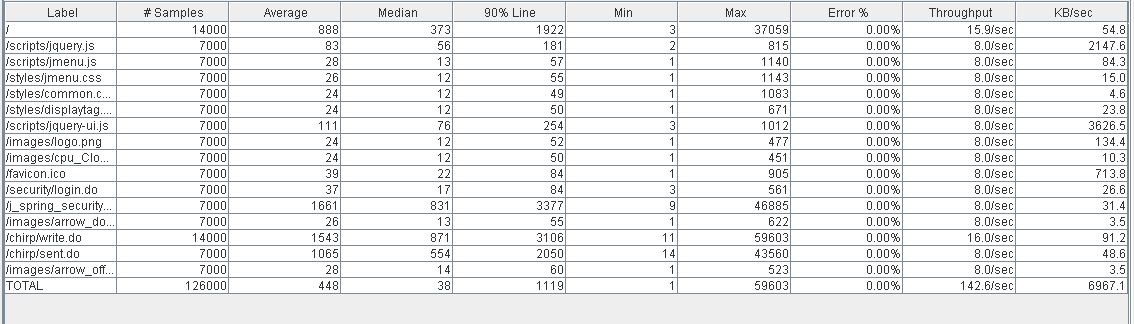


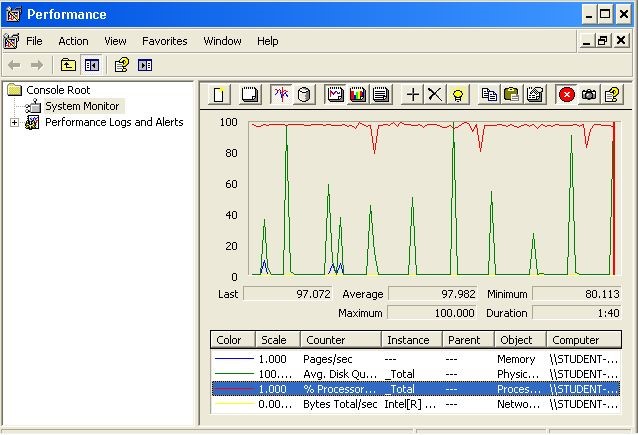


# Write chirp

Para realizar el test de write chirp hemos tenido que realizarlo con 100 usuarios y 70 loops, con 190 funcionaba correctamente pero al no borrar los mensajes que se iban creando los tiempos aumentaban mucho al tener que cargar la lista de mensajes enviados y terminaba quedándose la máquina colgada.



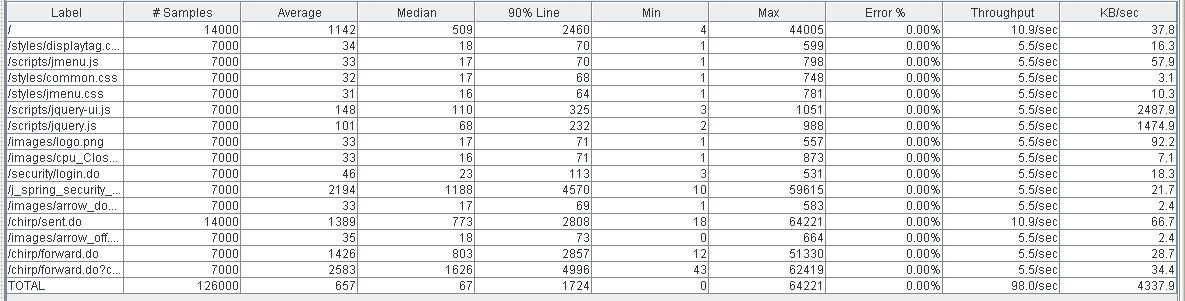


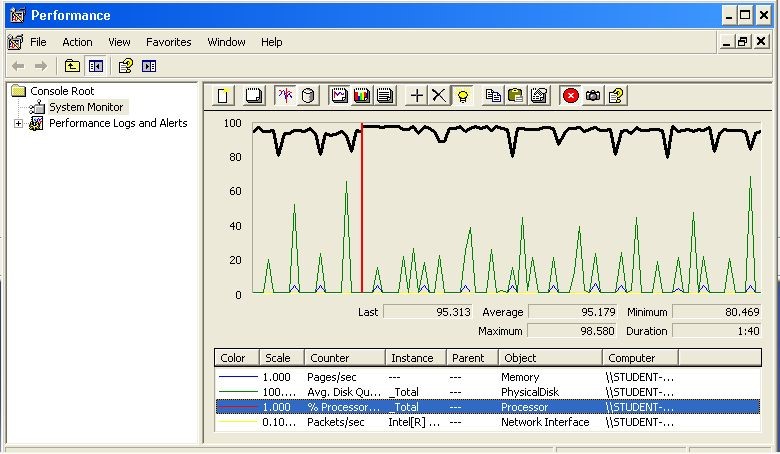


# Forward chirp

Para realizar el test de forward chirp hemos tenido que realizarlo con 100 usuarios y 70 loops, con 190 funcionaba correctamente pero al no borrar los mensajes que se iban creando los tiempos aumentaban mucho y terminaba quedándose la máquina colgada.



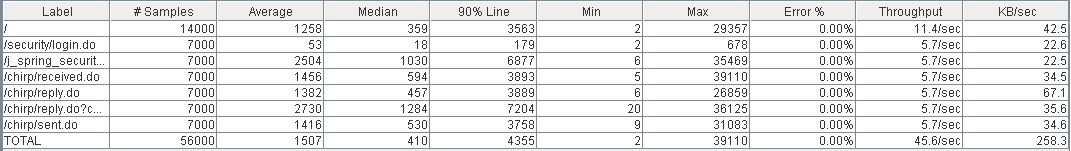


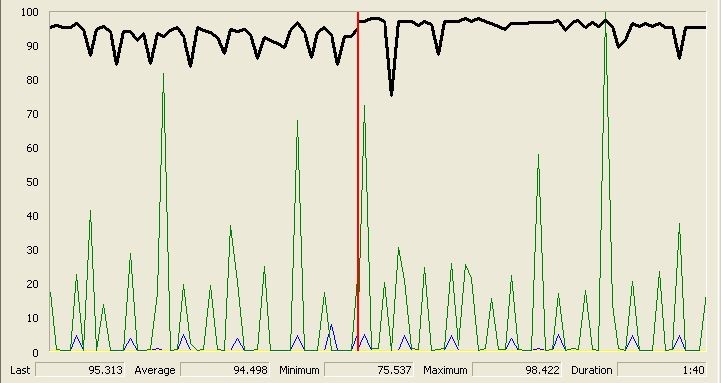


# Reply chirp

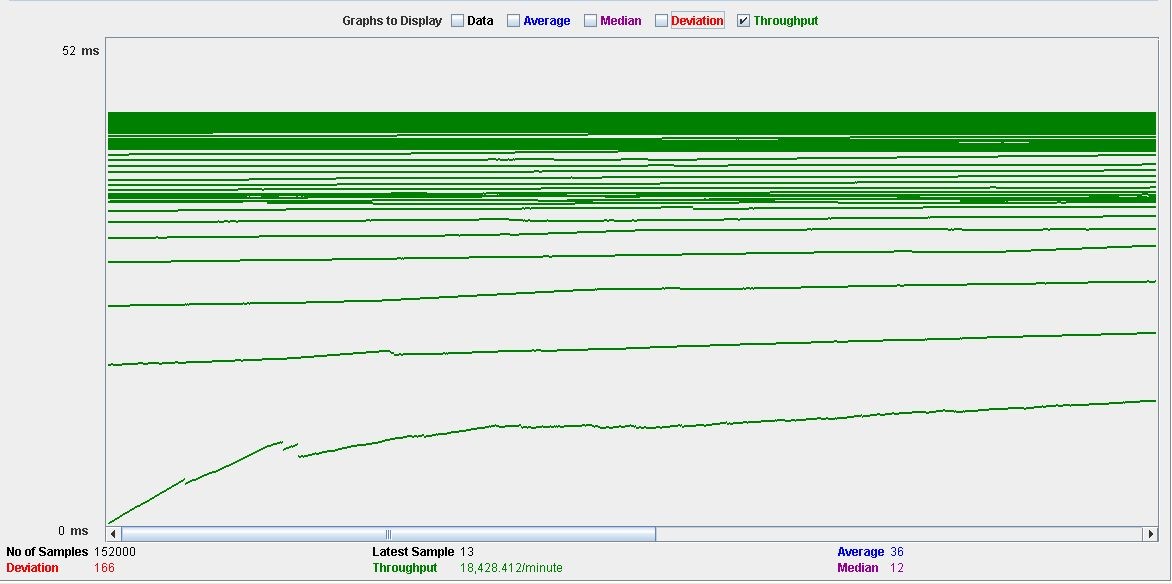
Para realizar el test de reply chirp hemos tenido que realizarlo con 100 usuarios y 70 loops, con 190 funcionaba correctamente pero al no borrar los mensajes que se iban creando los tiempos aumentaban mucho y terminaba quedándose la máquina colgada.

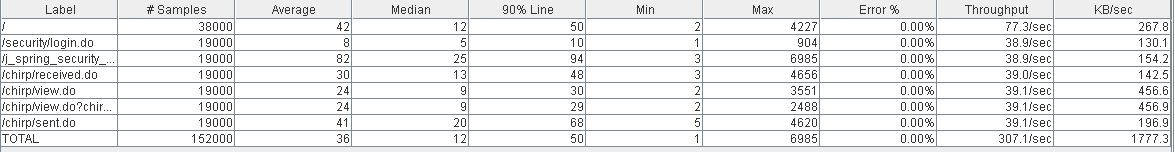


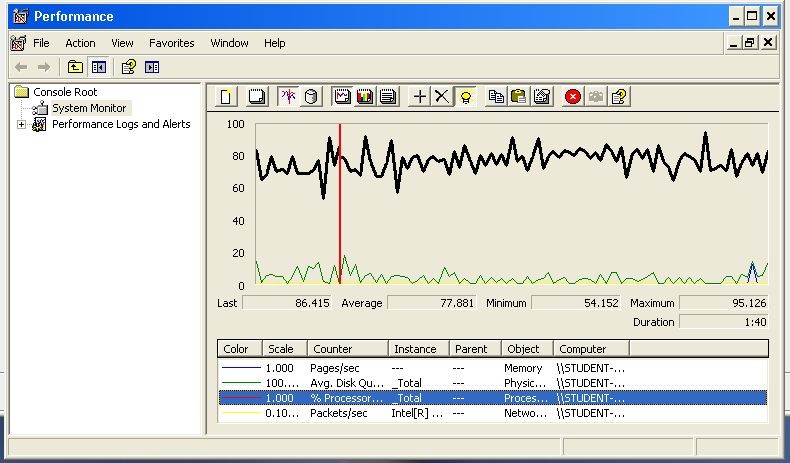




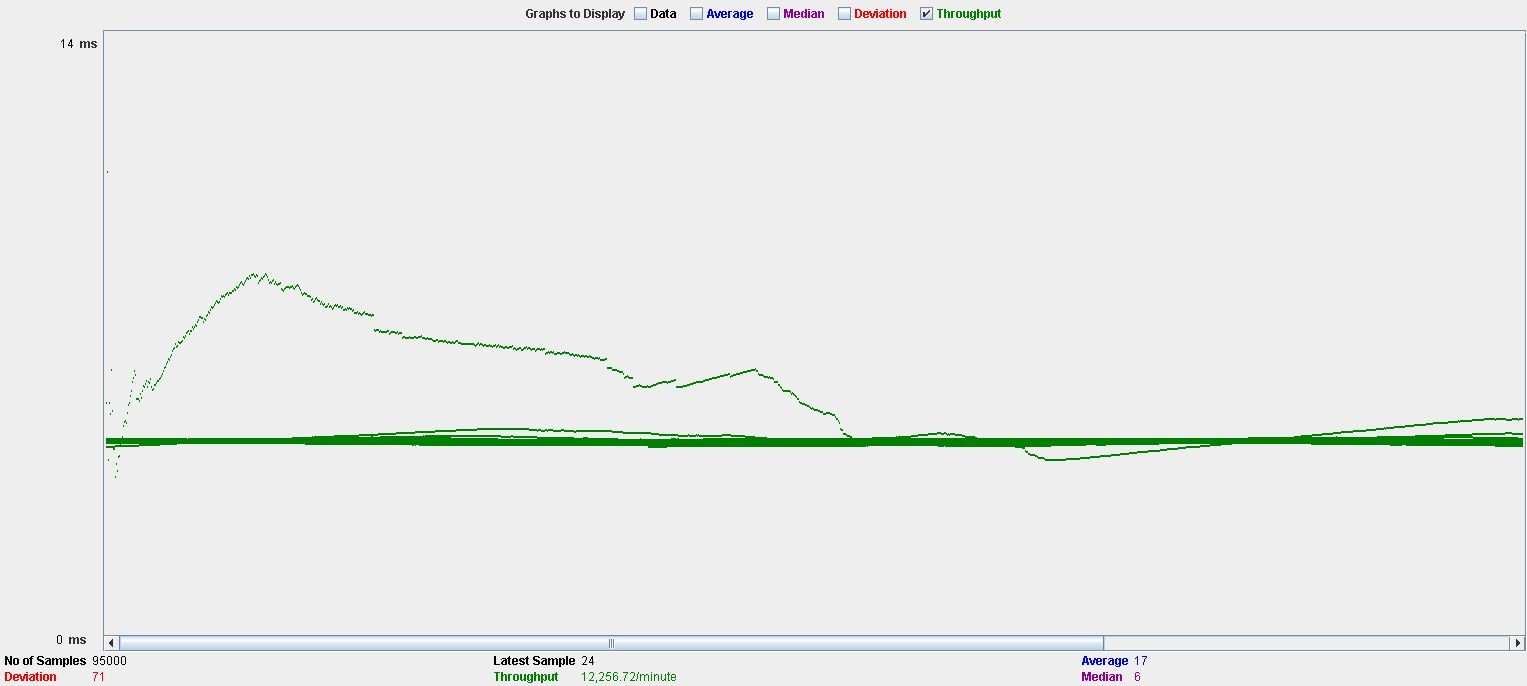
# Delete chirp

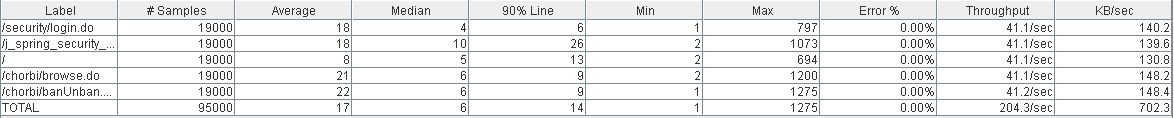


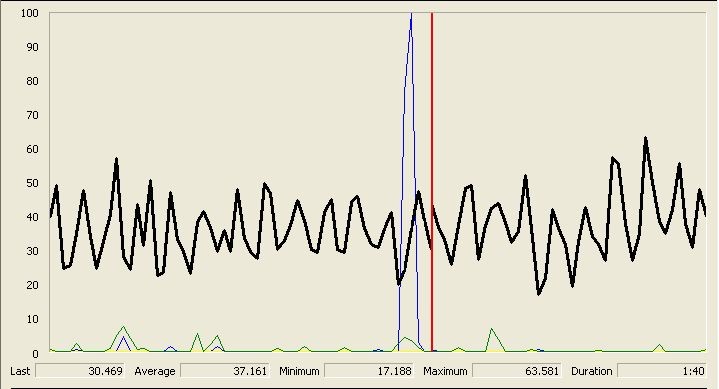




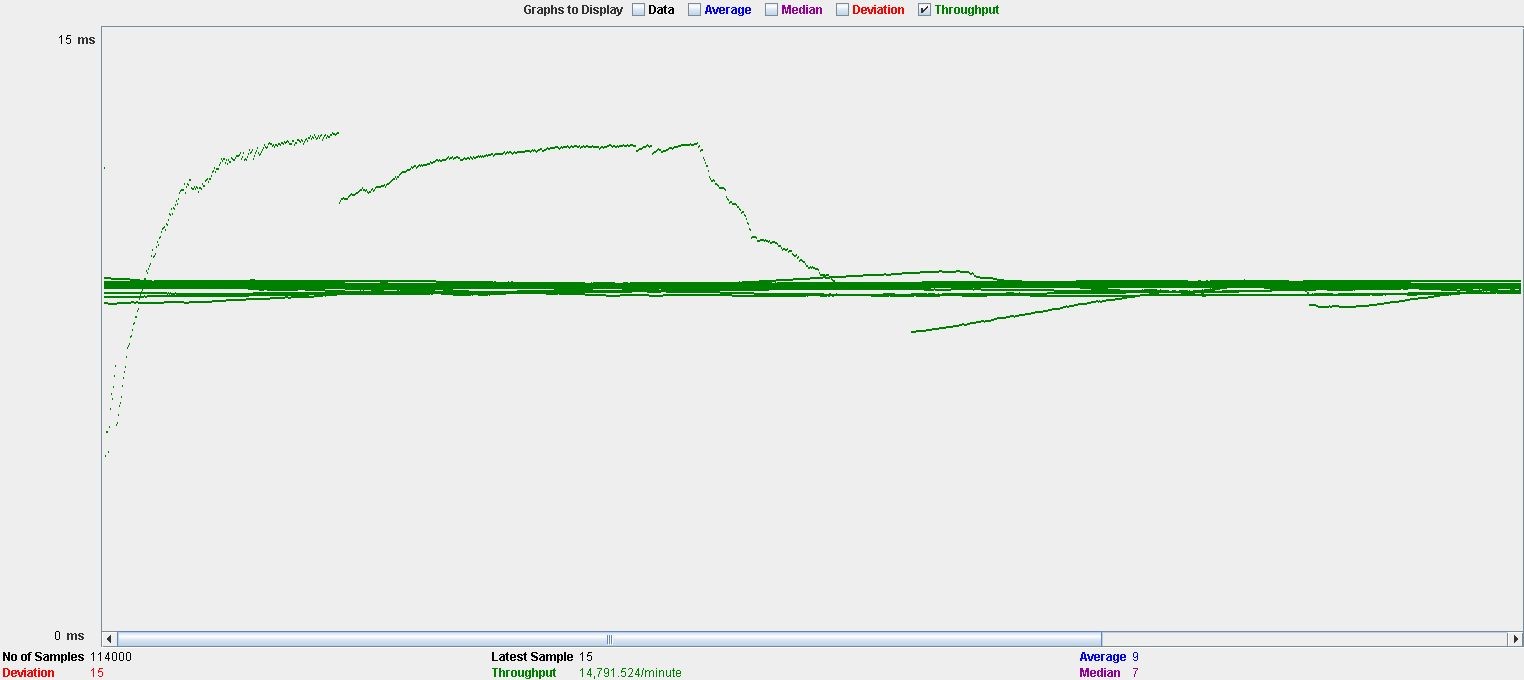
# Ban and unban chorbi

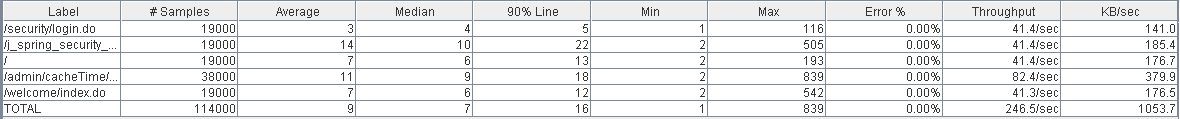


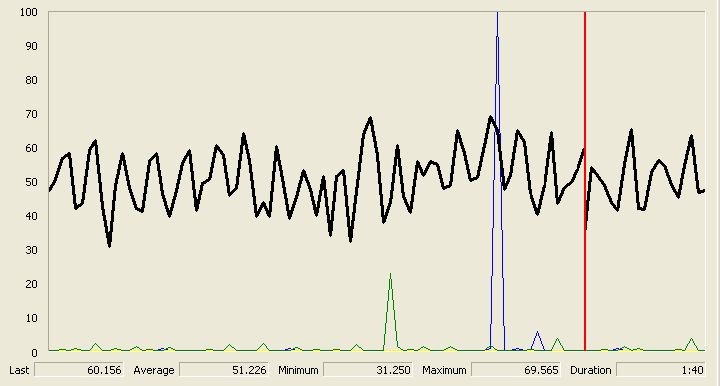




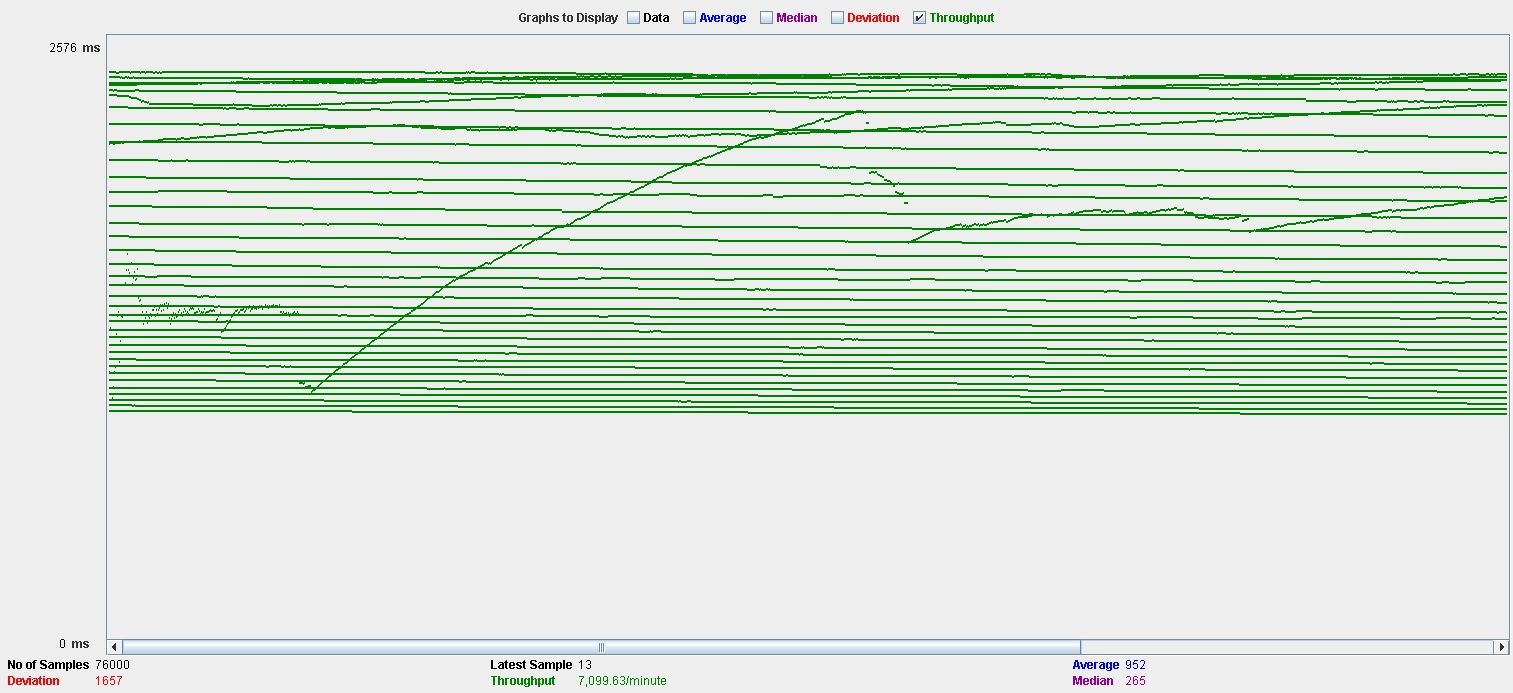
# Edit cache time

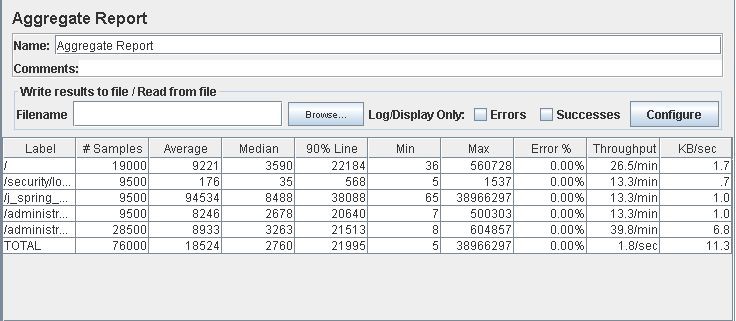






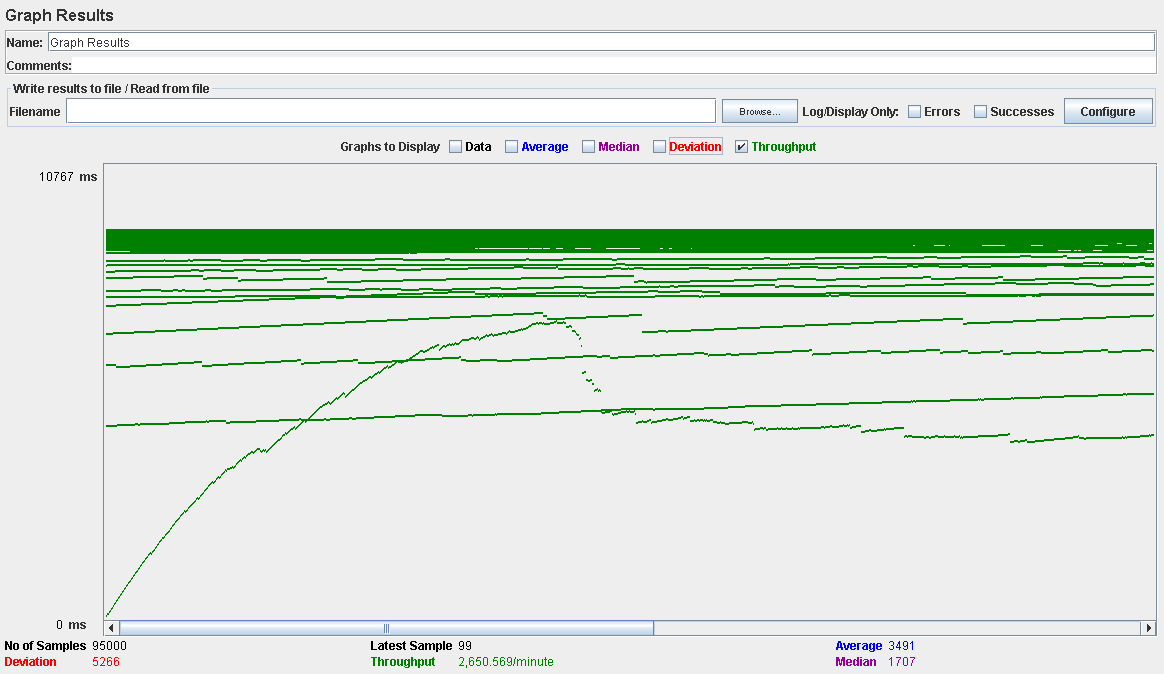
# Change banners

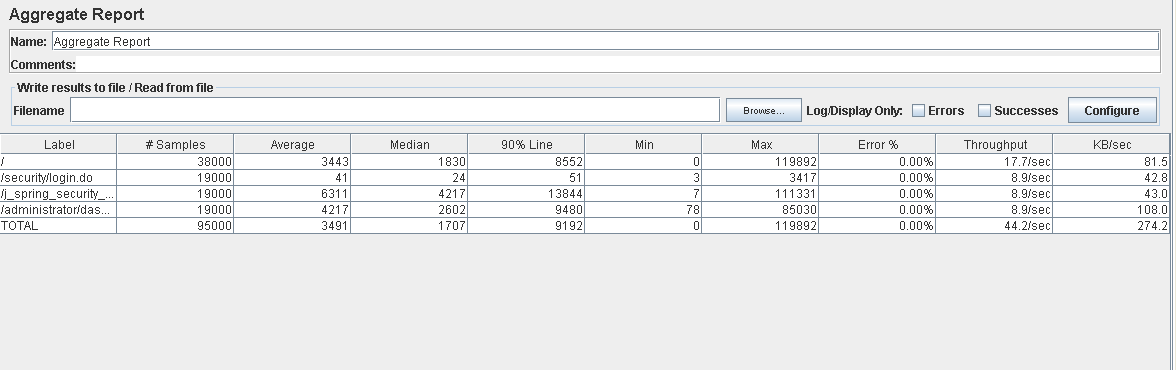


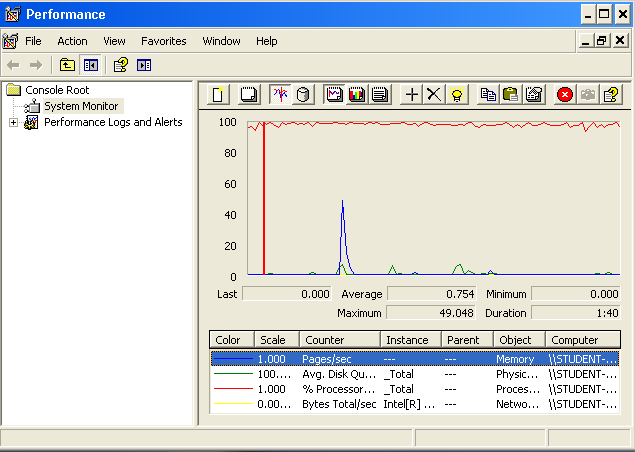




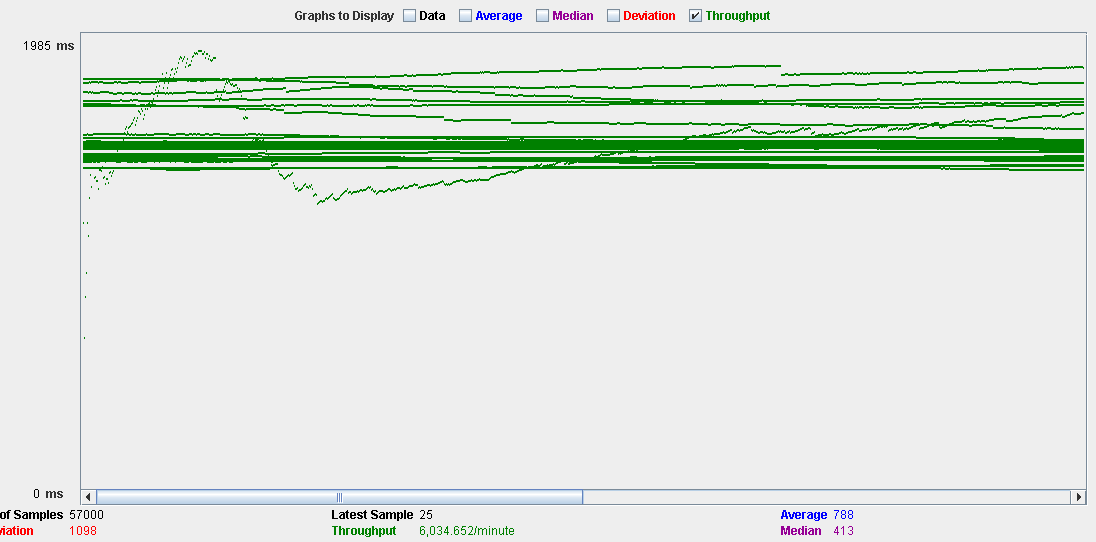
# Dashboards

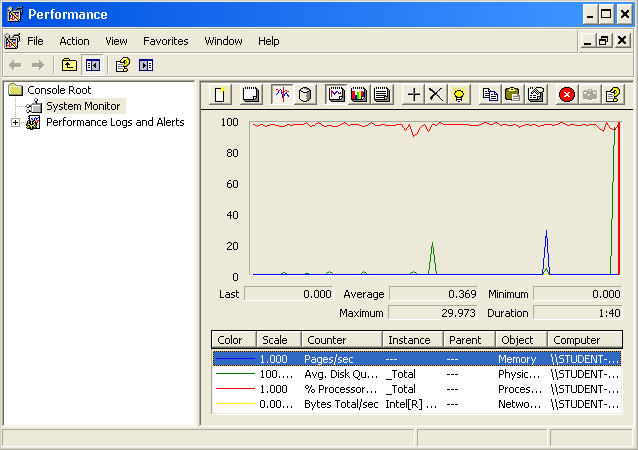


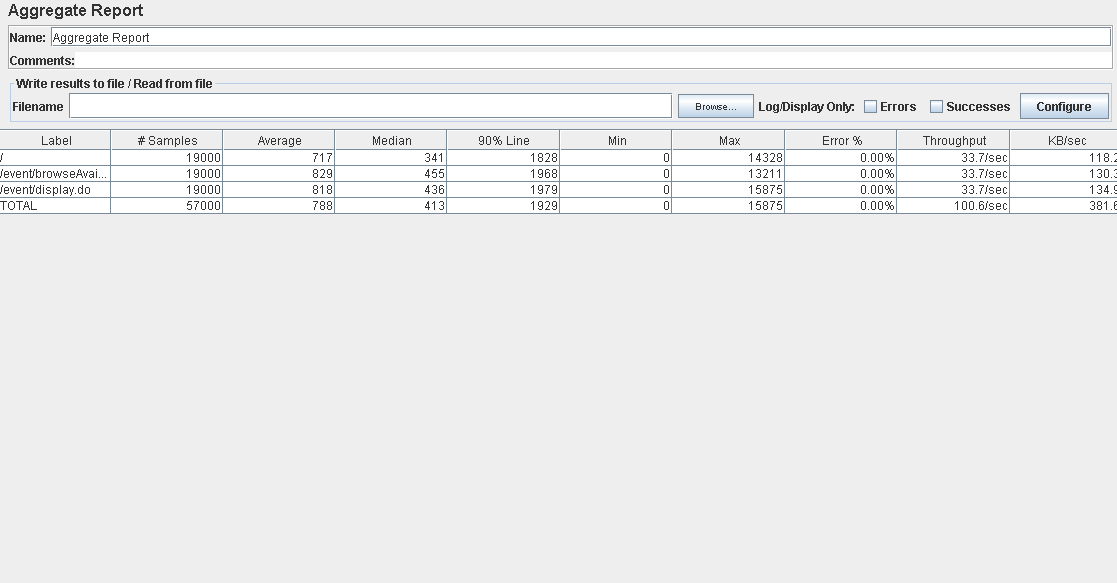




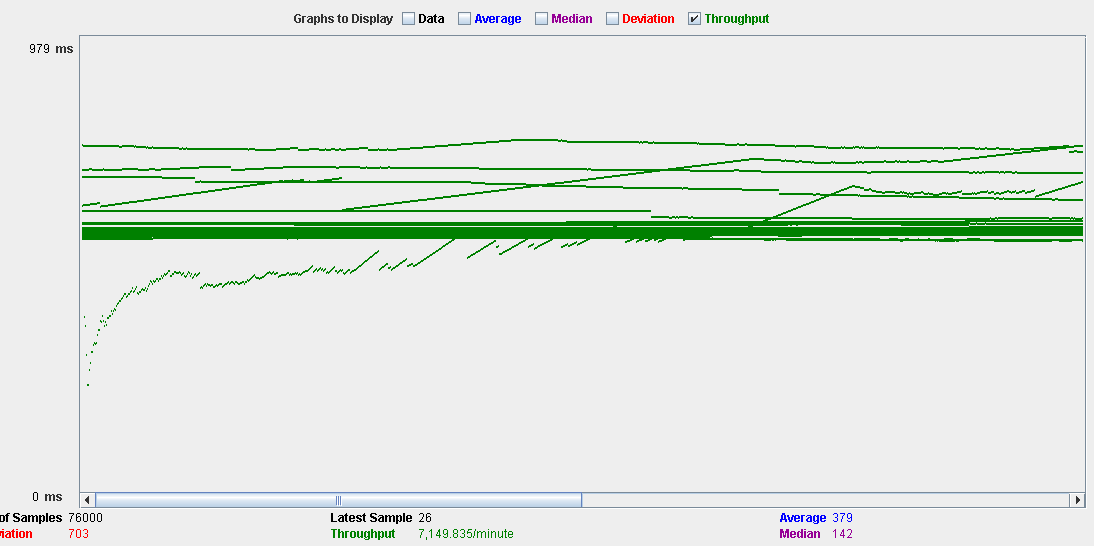
# Browse available events

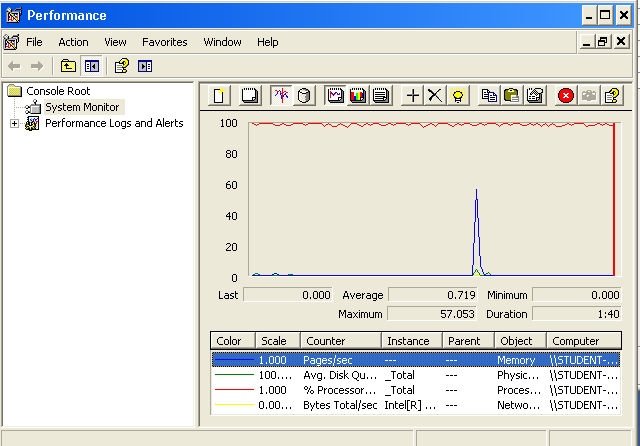


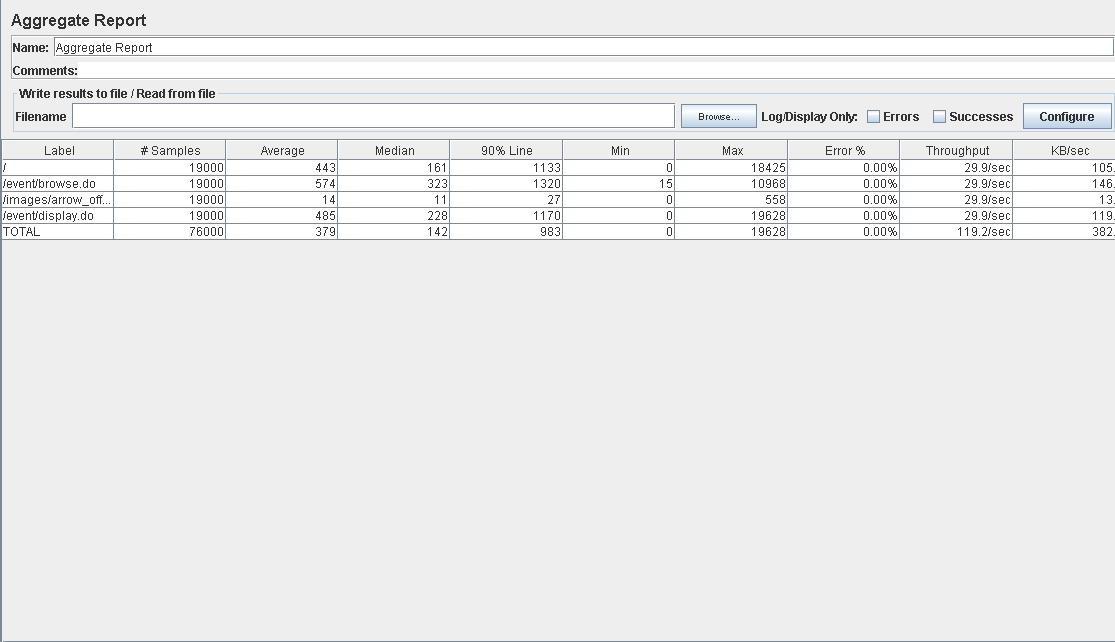




# Browse all events

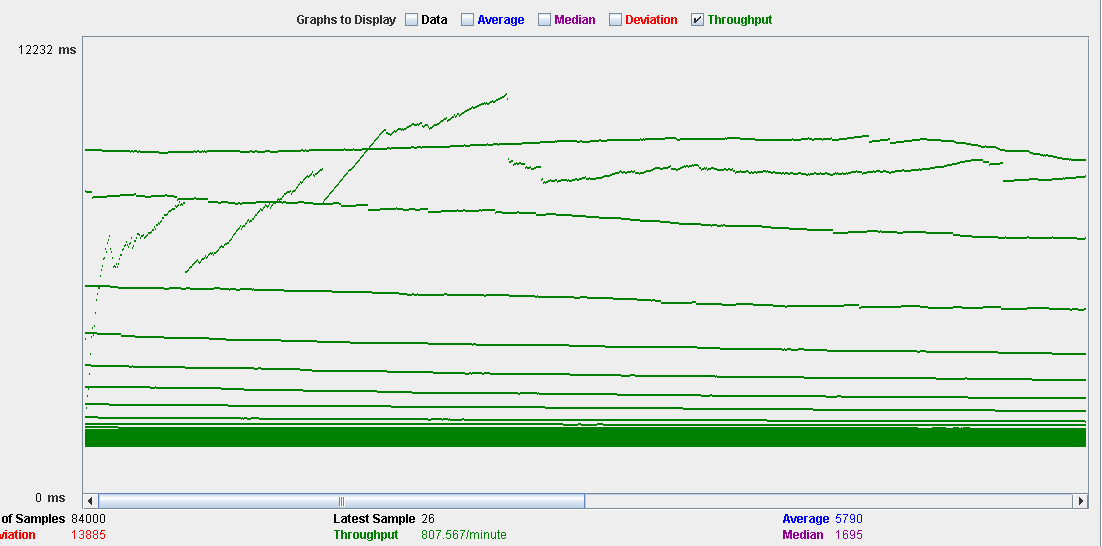




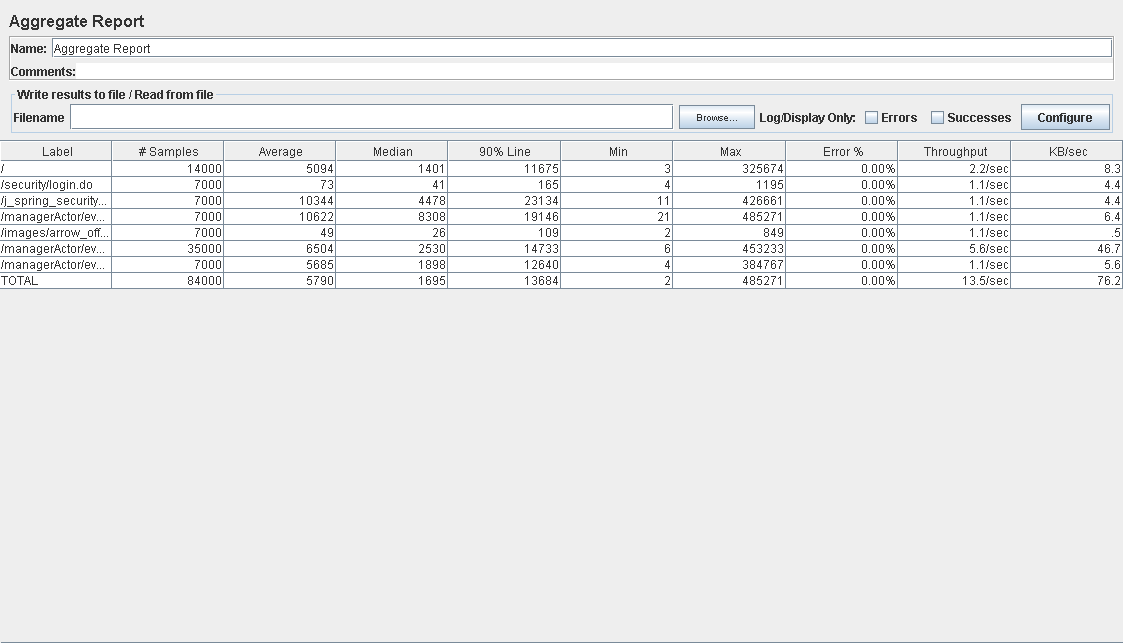


# Manage events

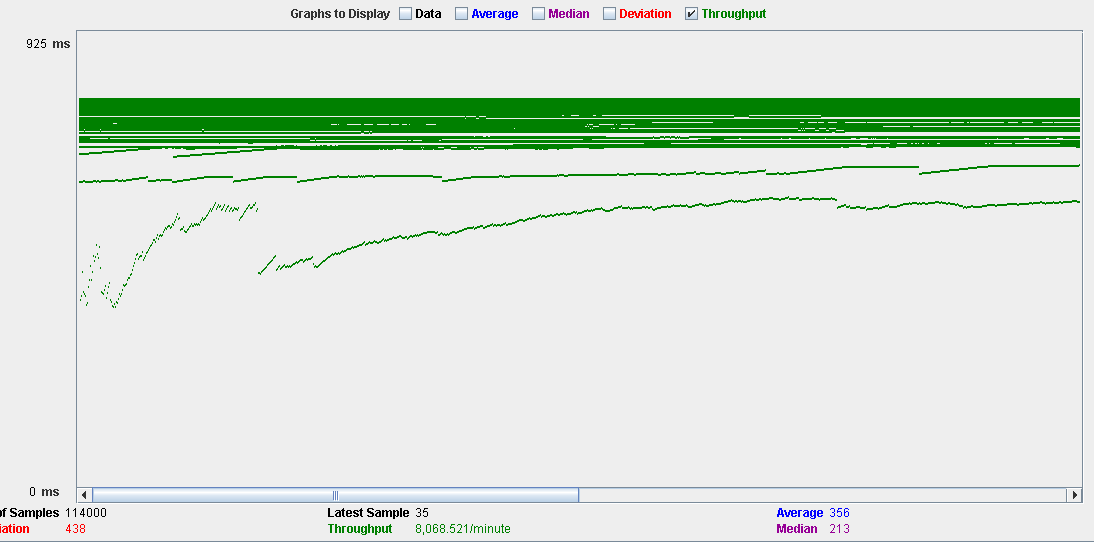
Para la creación de eventos hemos tenido que reducir el número de usuarios simultáneos a 100 ya que llegaba un momento que al tener un manager más de 10000 eventos creados se volvía el sistema lento. Podemos ver en el grafo como aumentan los tiempos a medida que hay más eventos creados.

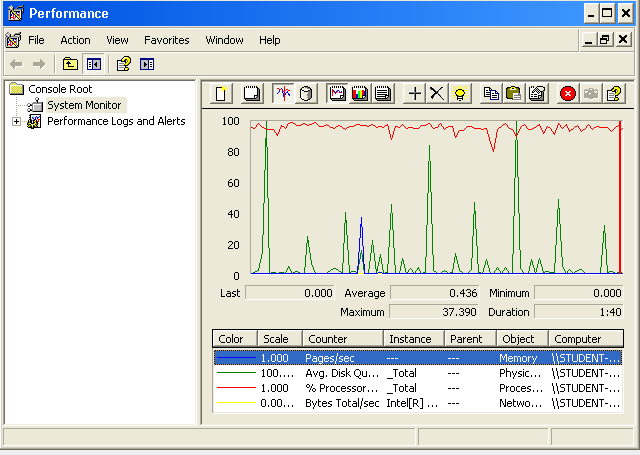


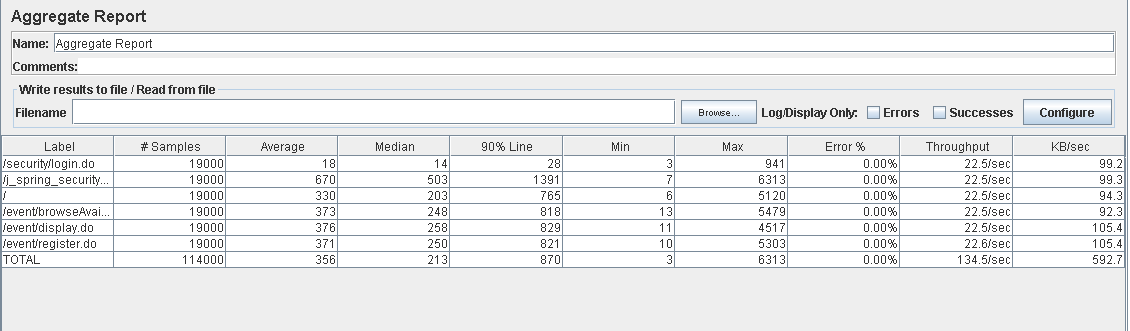




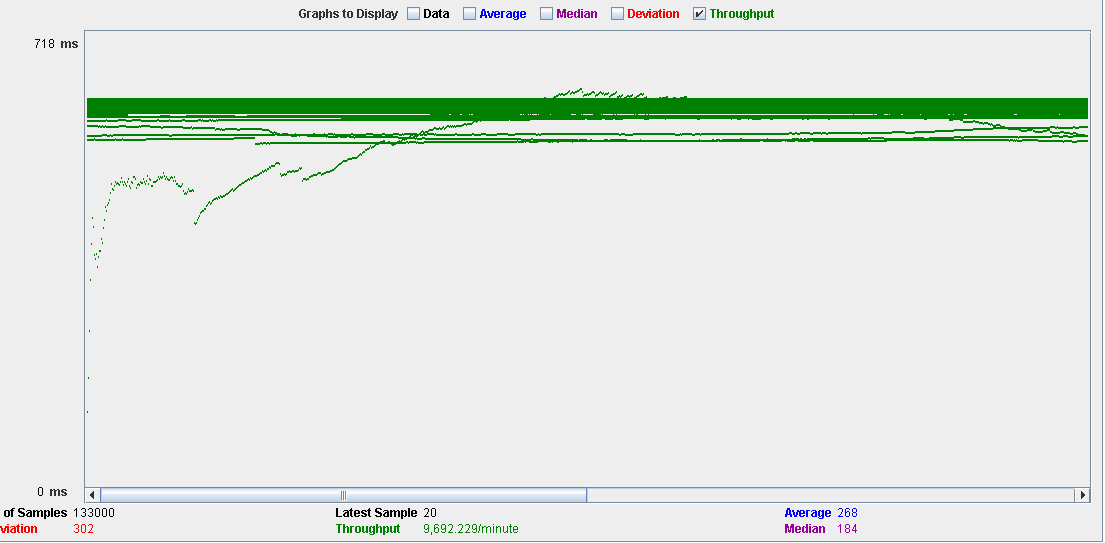
# Register to an event

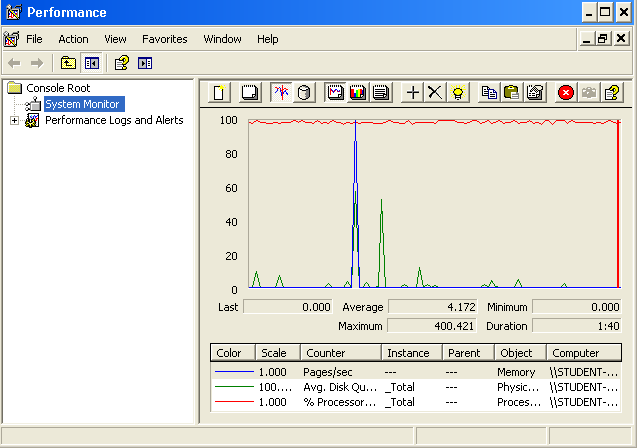


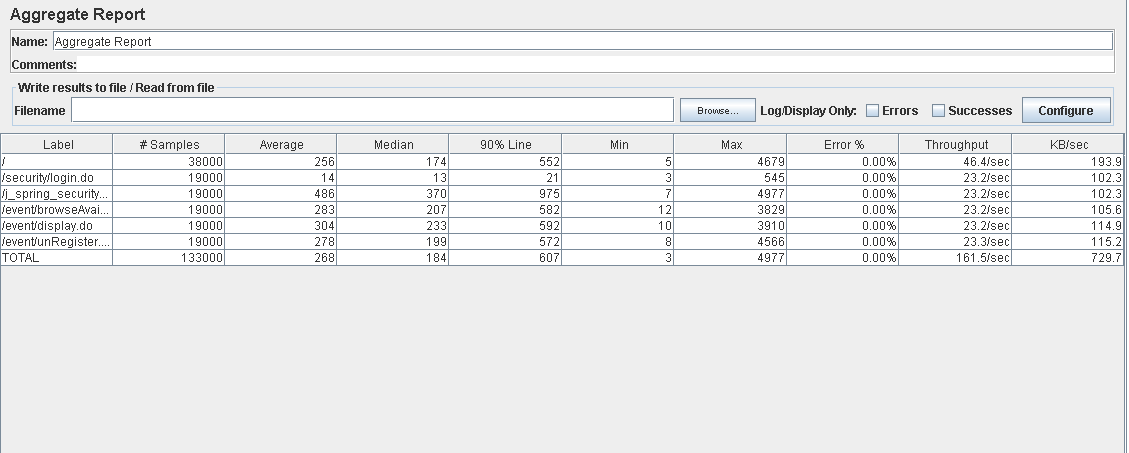




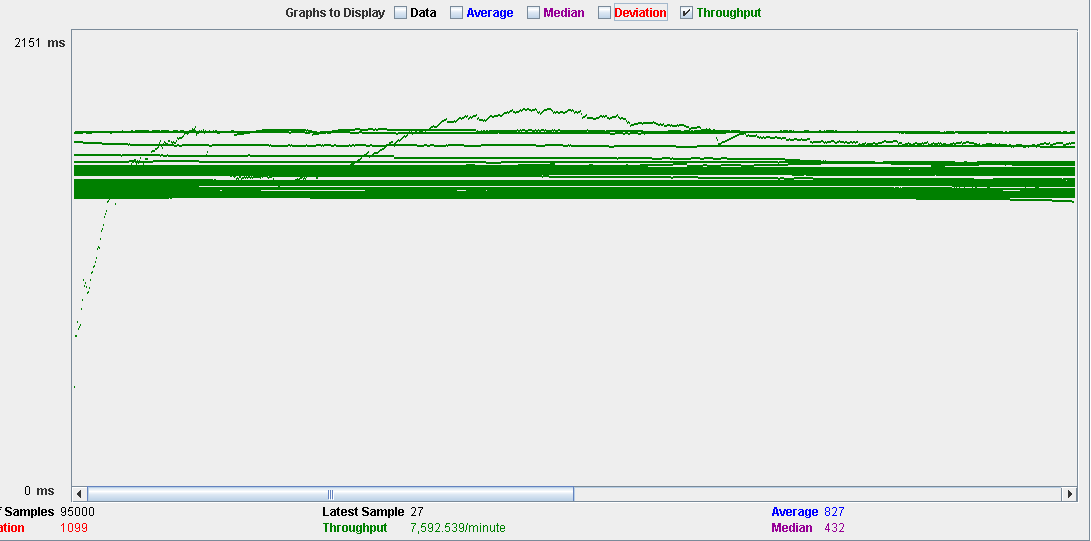
# Un-register to an event

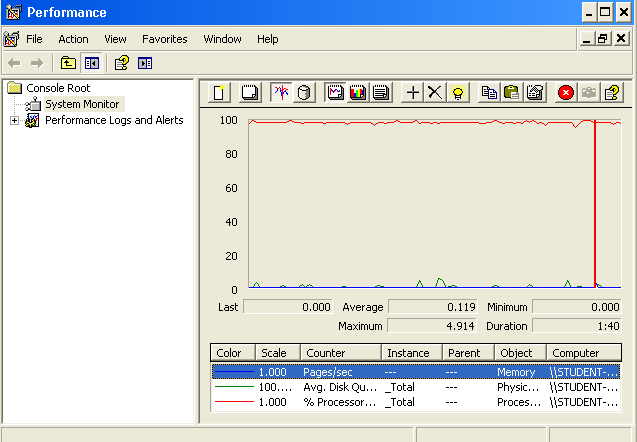


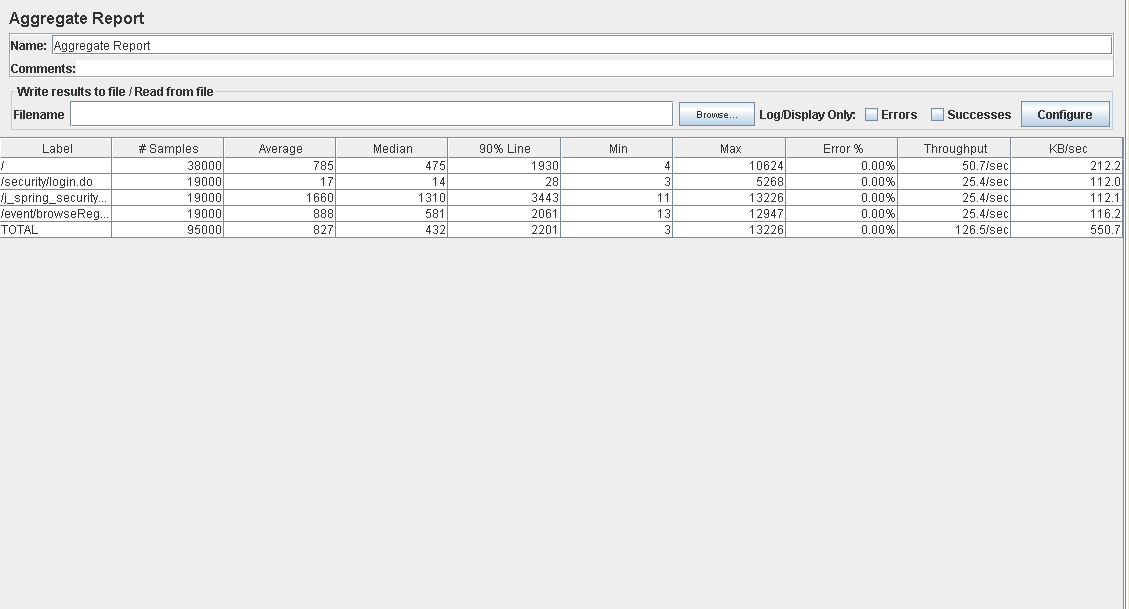




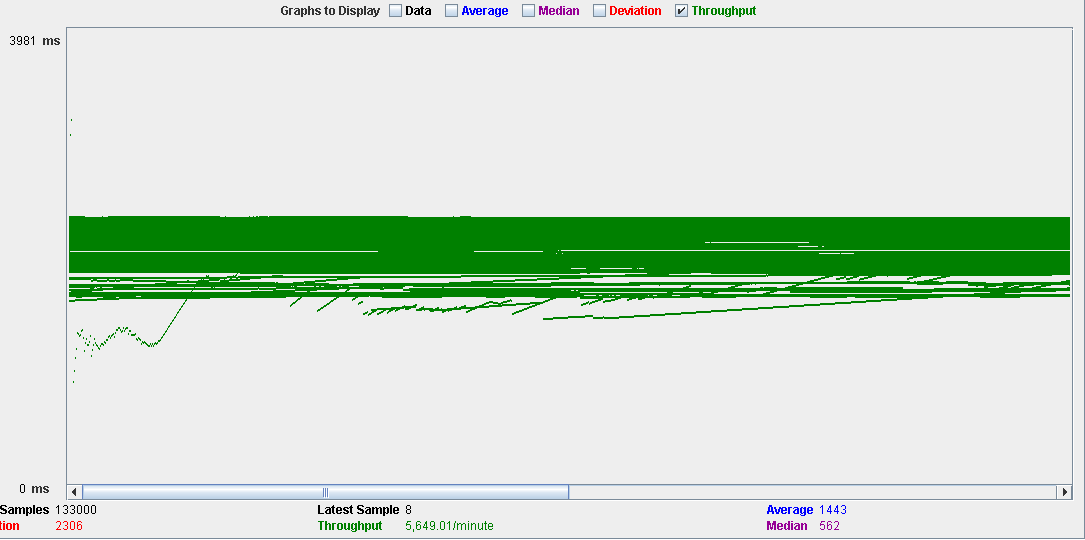
# Browse registered events

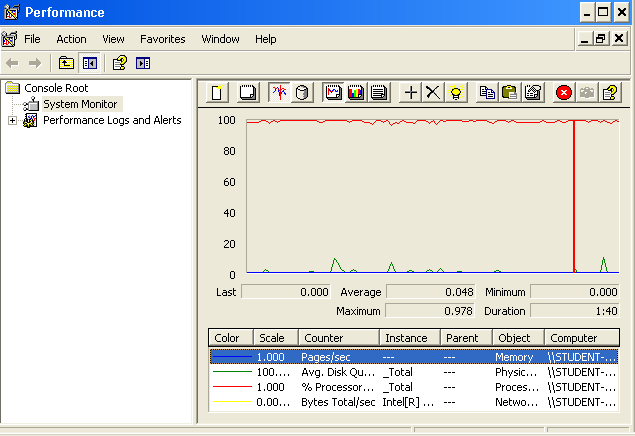


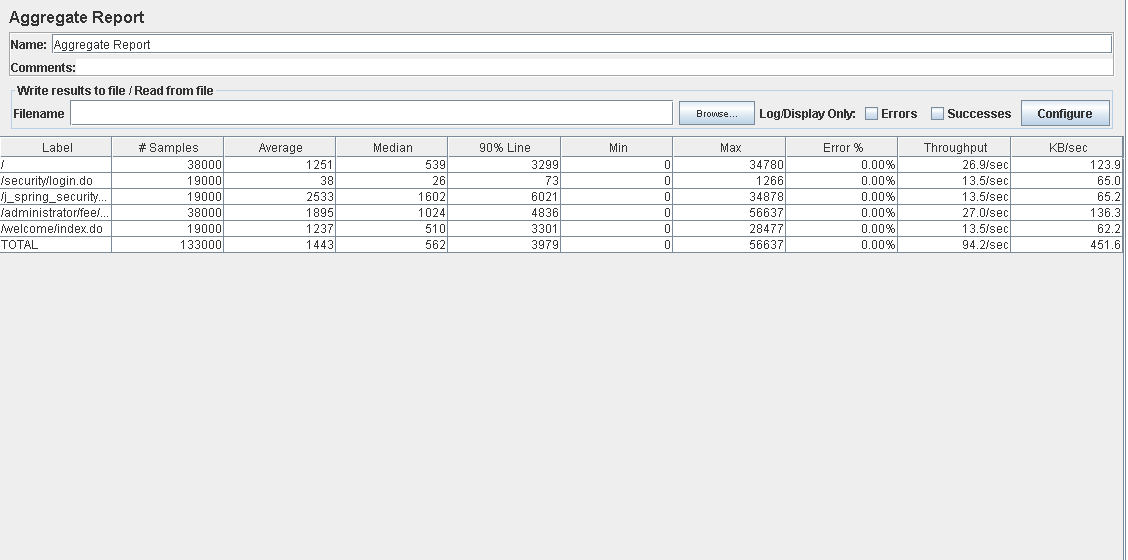




# Change the fee

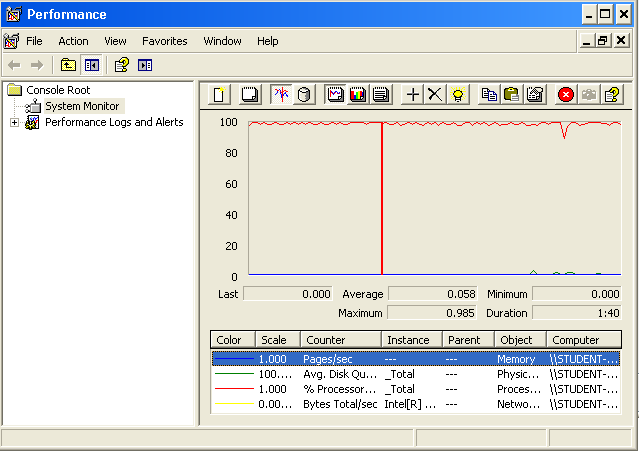


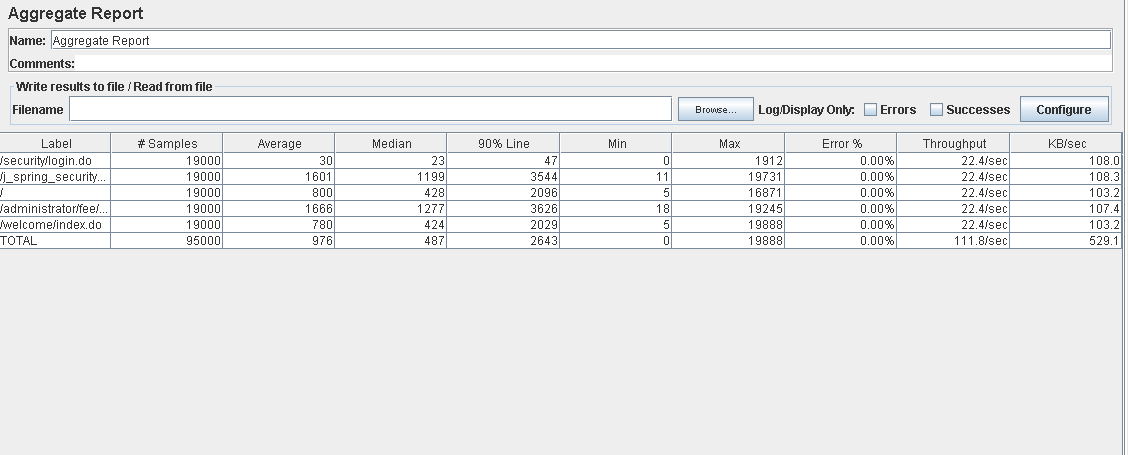




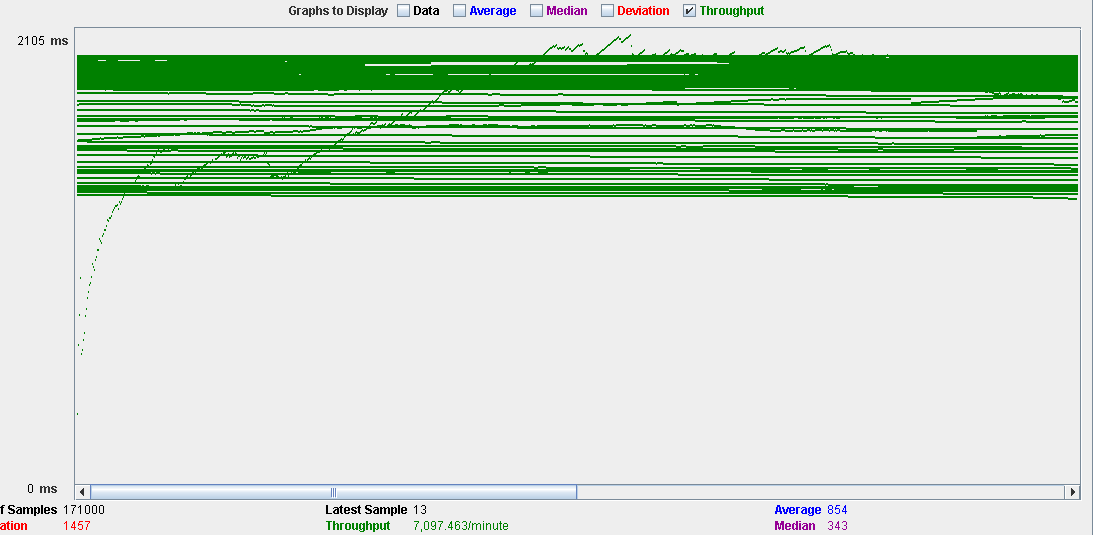
# Update chorbies monthly fees

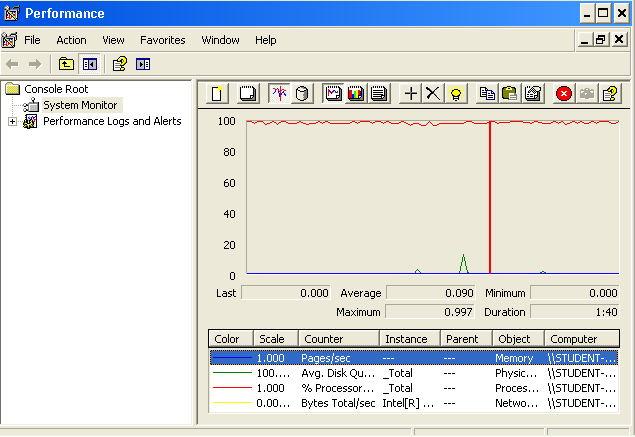


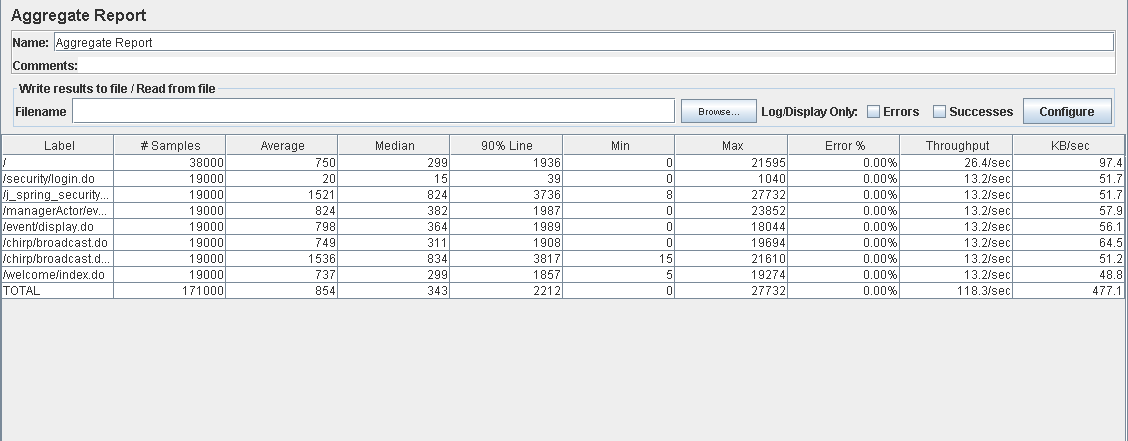




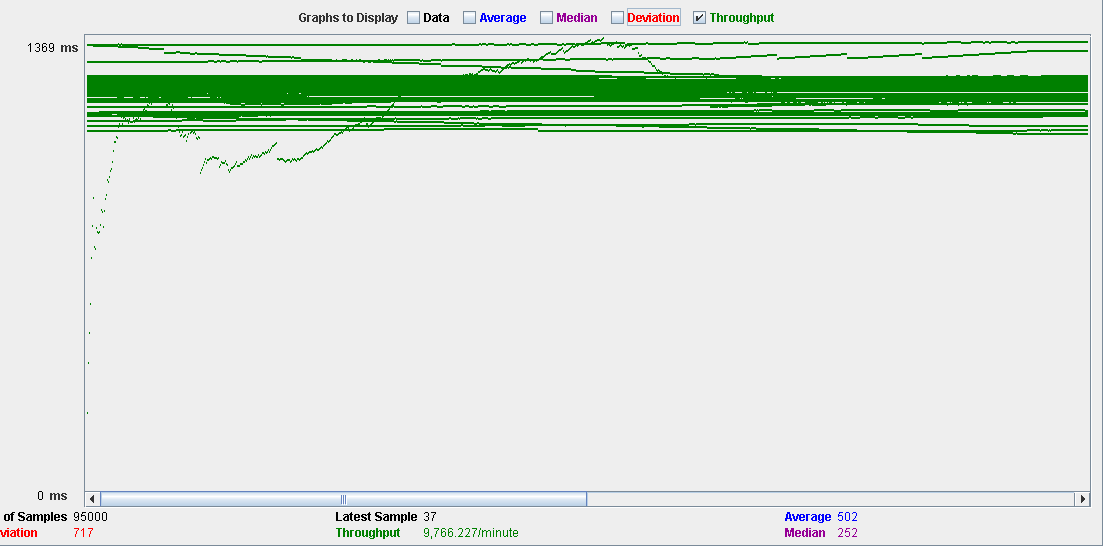
# Broadcast

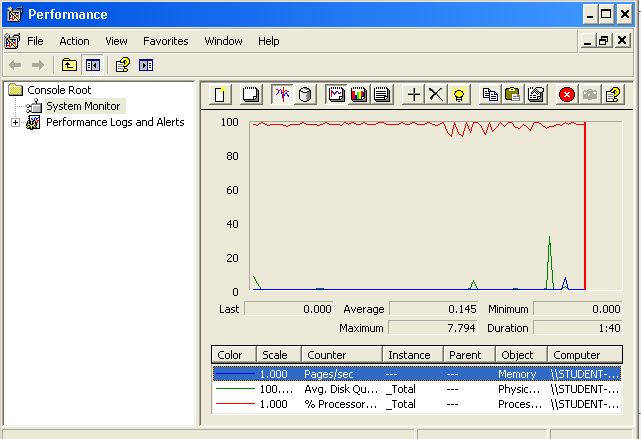


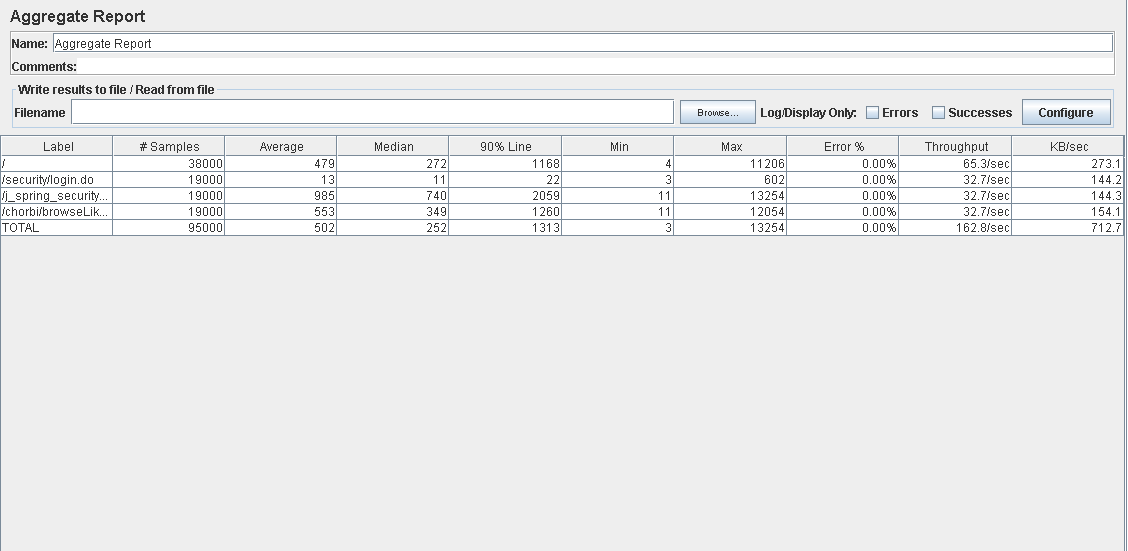




# Browse chorbies who like me







# Conclusiones

Como podemos observar en todos los test al aumentar el número de usuarios más tarda en realizarse cada acción. Además, podemos ver que el procesador es nuestro cuello de botella, si lo reemplazamos por un procesador más potente podríamos aumentar la cantidad de usuarios simultáneos.