

## Introducción

Los test de rendimiento evalúan el rendimiento de nuestra aplicación con el fin de poder transmitirle al cliente cual es la carga máxima que soporta su sistema. Puesto que no se puede probar directamente en el sistema, el lugar donde se debería probar es en la máquina de Pre-Producción ya que es el lugar más parecido al entorno real. Debido a que la máquina de Pre-Producción es una máquina virtual ejecutada desde distintos ordenadores, el rendimiento varía entre ellas y por esto hemos decidido recopilar los dos test que, tras ejecutarlo varias veces en distintas máquinas, obtienen un peor rendimiento (y destacable) entre los otros Test.

A continuación se van a ir detallando los citados test, ya que suponen el cuello de botella de la aplicación y se van a ir proponiendo opciones para mejorar el rendimiento de las acciones realizadas por los test consiguiendo así mejorar el rendimiento de la aplicación en general.

El primer test será denominado de aquí en adelante Test\_A y hace referencia al requisito funcional 6 - D el cual implica gestionar las carreras del sistema. El segundo test (de aquí en adelante Test\_B) hace referencia al requisito funcional 6 - C el cual implica gestionar las ligas.

## Detalles del entorno

Antes de continuar debemos detallar en que máquina se han lanzado dichos tests y que configuraciones tenía tanto la máquina virtual como los propios test:

### Máquina física

Portátil HP-Pavilion con i7, 8GB de RAM y disco HDD

### Máquina virtual

2 Núcleos al 100% con las características PAE/NX habilitadas y con 1536 MB de RAM

### Detalles JMeter

A ambos test se le han introducido un retraso de campana gaussiana de 1500 ms en cada acción que sería realizada por un usuario real (hacer click o rellenar un formulario) con el fin de que sea lo más real posible.

Respecto al número de usuarios, el “ramp-up period” y el número de iteraciones por cada usuario los datos en cada test han sido los siguientes:

- Test\_A: 40 usuarios, en 1 segundo (ramp-up period) y cada usuario se ejecuta 40 veces.
- Test\_B: 50 usuarios, en 1 segundo (ramp-up period) y cada usuario se ejecuta 50 veces.

# Desarrollo de los test

Ambas pruebas tienen como cuello de botella el mismo error, realizar muchas peticiones a la base de datos de escritura/lectura del tipo save() o findAll() para comprobar los datos introducidos. Como podemos observar en ambas gráficas, esto es apreciable en las columna “90% Line” en las líneas listar y crear en el Test\_A y en listar, crear y editar en el Test\_B.

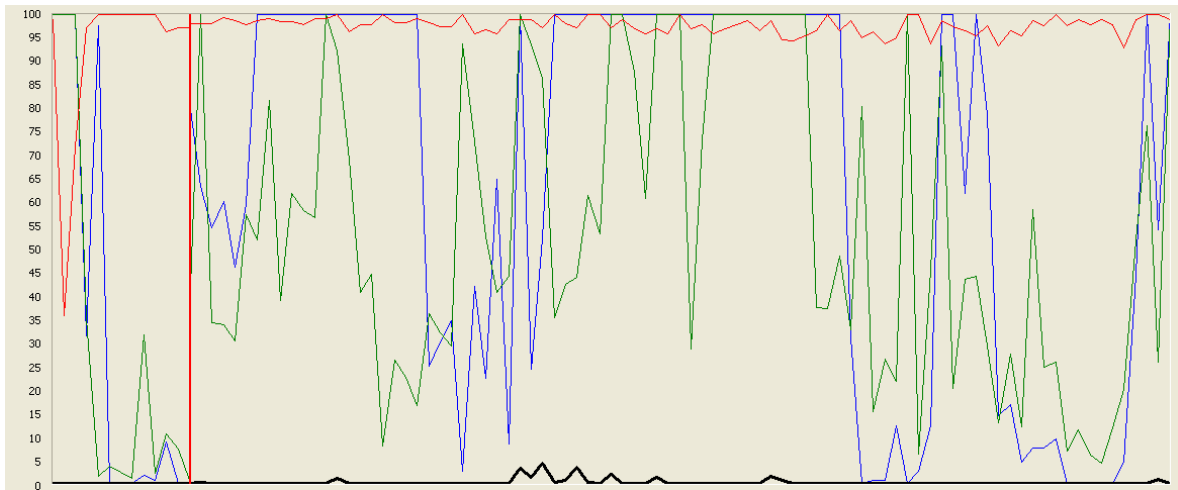
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	4800	143	50	326	3	4671	0.00%	2.6/sec	11.6
/styles/jmenu.css	1600	105	31	281	1	5210	0.00%	52.2/min	1.6
/scripts/jmenu.js	1600	91	20	260	1	1834	0.00%	52.2/min	9.2
/scripts/jquery.js	1600	309	92	892	3	3588	0.00%	52.2/min	234.5
/scripts/jquery-u...	1600	365	52	1127	4	4558	0.00%	52.2/min	395.7
/styles/common...	1600	81	9	245	1	3326	0.00%	52.3/min	.5
/images/logo.png	1600	116	12	366	2	4720	0.00%	52.3/min	57.1
/styles/displayta...	1600	74	7	236	1	2611	0.00%	52.3/min	2.6
/favicon.ico	1600	69	8	218	1	1743	0.00%	52.3/min	1.2
/security/login.do	1600	120	57	302	3	3319	0.00%	52.3/min	3.8
/j_spring_secur...	1600	970	443	2471	9	19457	0.00%	52.3/min	4.3
/race/administr...	9600	1667	1016	4167	24	17491	0.00%	5.2/sec	70.5
/race/administr...	1600	1242	601	3334	11	11843	0.00%	52.3/min	6.3
/race/administr...	8000	2107	846	6022	5	22417	0.00%	4.3/sec	51.6
TOTAL	40000	980	178	3103	1	22417	0.00%	21.5/sec	840.0

Tabla peticiones web Test\_A

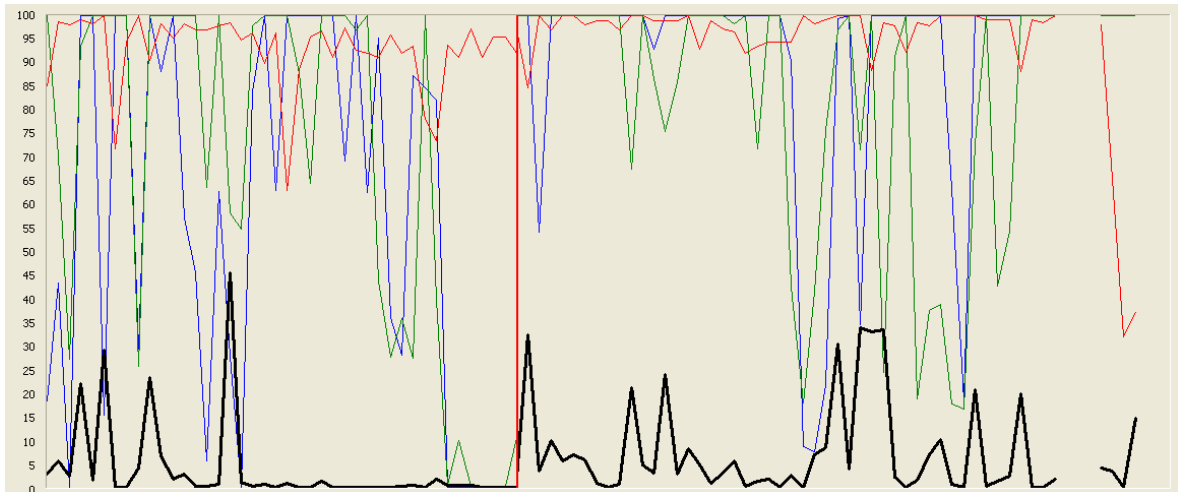
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	7500	238	23	389	2	28665	0.00%	2.7/sec	12.1
/scripts/jquery.js	2500	215	57	650	3	4266	0.00%	54.1/min	242.9
/scripts/jquery-u...	2500	295	66	884	4	11880	0.00%	54.1/min	409.7
/scripts/jmenu.js	2500	72	8	193	1	26337	0.00%	54.1/min	9.5
/images/logo.png	2500	91	10	232	1	29058	0.00%	54.1/min	59.2
/styles/common...	2500	66	7	197	1	6458	0.00%	54.1/min	.5
/styles/displayta...	2500	59	7	186	1	5116	0.00%	54.1/min	2.7
/styles/jmenu.css	2500	69	6	186	1	26337	0.00%	54.1/min	1.7
/favicon.ico	2500	61	6	180	1	12491	0.00%	54.1/min	1.2
/security/login.do	2500	166	27	424	3	13601	0.00%	54.1/min	3.9
/j_spring_secur...	2500	1494	494	3758	7	48747	0.00%	54.1/min	4.4
/league/admini...	15000	2682	1451	6435	13	57529	0.00%	5.4/sec	65.4
/league/admini...	2500	1625	579	4236	7	35151	0.00%	54.1/min	5.8
/league/admini...	12500	2225	701	6027	4	52914	0.00%	4.5/sec	46.1
TOTAL	62500	1286	153	3787	1	57529	0.00%	22.3/sec	858.0

Tabla peticiones web Test\_B

Esto se debe (como podemos observar en las siguientes gráficas) al estado del procesador, la memoria RAM y el acceso al disco del sistema los cuales se encuentran continuamente al límite de sus posibilidades.



*Ilustración carga sistema durante ejecución de Test\_A*



*Ilustración carga sistema durante ejecución de Test\_A*

Nota: Los colores representan la memoria ram (azul), el disco duro (verde), el procesador (rojo) y la tarjeta de red (negro).

Para mejorar estos resultados, la opción por parte del cliente es comprar procesadores con más potencia, discos duros cuyo acceso de lectura escritura sea más rápido y memorias RAM más rápidas y con más capacidad (lo que podría evitar tener que mejorar el disco duro temporalmente) y por parte de los desarrolladores de la aplicación (nosotros) usar estas llamadas “lentas” el menor número de veces posible lo que conllevaría a programar de manera más eficiente en general.

Con todo esto podemos concluir que el límite de nuestro sistema serían unos 40 usuarios por segundo.