Laboratorio desempeño y escalabilidad

Grupo buses nacionales

Integrantes

Ferney Esteban Lizarazo López (Scrum Master)
Johan Camilo Mantilla Mantilla
Daniel Felipe Sabogal Fajardo
Andrés Francisco Rodríguez Peraza

Modificaciones que se realizarán a la arquitectura

Dado que se espera que la aplicación presente una gran cantidad de carga, debido a que se espera que funcione a nivel nacional, el sistema debe soportar el ingreso de muchos usuarios en unidades de tiempo cortas, y dar respuesta a todos los usuarios en intervalos de tiempo tolerables.

Para esta aplicación, se espera que en un intervalo de 1 segundo, se presenten como máximo 150 solicitudes de inicio de sesión, las cuales deben ser respondidas en un rango de 2 segundos

Para evaluar el cumplimiento este requerimiento, se harán pruebas usando la herramienta JMeter, en la cual se simulará el inicio de sesión con varias cantidades de usuarios (simulados usando hilos) hasta llegar a los 500 usuarios en un periodo de tiempo de un segundo, usando primero una computadora portátil y otra de sobremesa con mayor capacidad de procesamiento. Para empezar, se hará una implementación de la aplicación en una máquina virtual, usando Kubuntu como sistema operativo. en este se pondrán límites al procesamiento de la máquina, para con esto descubrir en qué punto se puede dejar de incrementar la capacidad de computación si se cumple el requerimiento de desempeño. en caso de no lograr el requerimiento en la primera máquina usada, se usará una máquina con un procesador mucho más rápido, que permita cumplir el requerimiento sin el uso de cluster.

Es de destacar se implementarán máquinas virtuales para la realización de las pruebas, debido a que la herramienta VirtualBox permite una rápida y sencilla configuración de los límites de procesamiento y un mejor control de los valores de hardware que se desean utilizar.

Una vez el requerimiento anteriormente mencionado se encuentre realizado, se hará una ampliación de la cantidad de usuarios que soporta la aplicación. se sabe que cuando la cantidad de usuarios supera cierto límite, el servidor empieza a fallar y no responde a todas las solicitudes, ya que se encuentra sobrecargado. esta pérdida de respuesta es en parte dependiente del hardware en el que se encuentra la aplicación. para que el servidor siga trabajando aun cuando se presentan estos problemas, se hará un cluster para la aplicación, con el cual se reducirá la carga en uno de los servidores haciendo que el otro reciba parte

de las solicitudes. Esto permitirá responder a más de 500 solicitudes de usuarios. Posteriormente se realizarán pruebas con la nueva arquitectura y se compararán con los resultados obtenidos de la primera prueba.

Es oportuno aclarar que la nueva arquitectura se implementará en máquinas virtuales, con el fin de dar continuidad a la los datos obtenidos en la primera implementación, además de facilitar la tarea de interconexión entre las partes de la misma, al no requerir hardware de conexión como cables y switches.

Como se decía anteriormente, después de cumplir el primer requerimiento (500 usuarios en 1 segundo, con su respuesta en menos de 10000 milisegundos), el objetivo es ampliar el número de usuarios hasta notar que hay ciertos usuarios que se quedan sin respuesta del servidor, con lo cual se debe hacer un replanteamiento de la arquitectura repartiendo el trabajo en diversos nodos. Al hacer esto, se podrá observar el cambio en los tiempos de respuesta, suavizando la carga en cada uno de los servidores y logrando responder a todas las solicitudes.

Para realizar esta prueba, se utilizó un computador que cuenta con un procesador AMD A8-4555M APU a 1.60 GHz y que cuenta con 8GB de RAM. Se utilizó una máquina virtual de Kubuntu, inicialmente configurada para tener un límite de ejecución de procesador de 25%, 50%, 75% y 100%. Los resultados obtenidos para la petición inicial de la página y un de inicio de sesión son los siguientes:

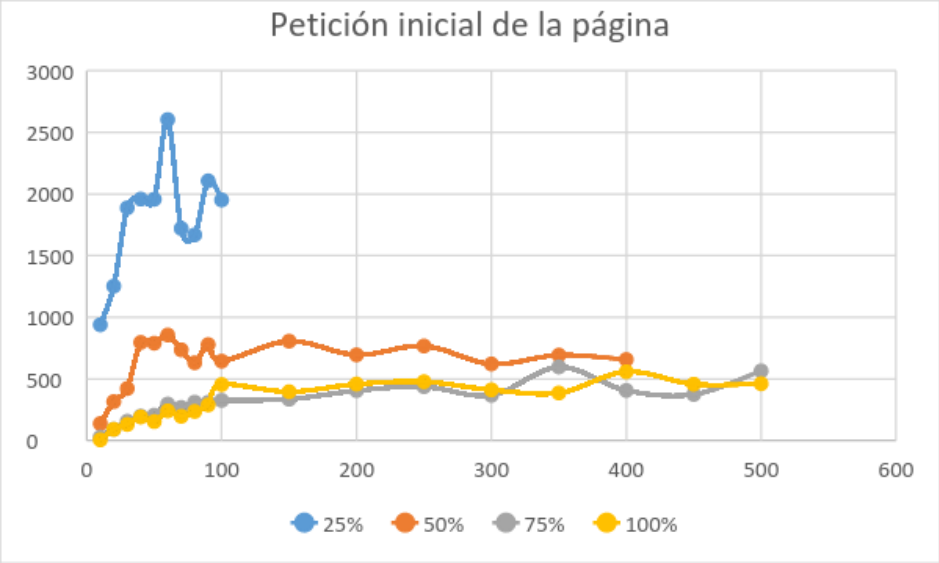


Figura 1

En la Figura 1 se observa el desempeño en los límites anteriormente mencionados para ésta máquina, es evidente que conforme el límite de ejecución asciende, es posible realizar pruebas con más hilos sin que ello afecte significativamente el tiempo promedio de respuesta. Esto se ve ilustrado en que el hilo de color azul (el cual corresponde a los tiempos promedios con un límite de ejecución de un 25%), éste es más corto que los siguientes, debido a que al realizar pruebas con más

de 100 hilos puede aumentar el tiempo de respuesta más de 3000 ms. Es oportuno aclarar que dicha gráfica corresponde al tiempo en el cual el servicio responde a una petición de acceso al sitio inicial de la aplicación.

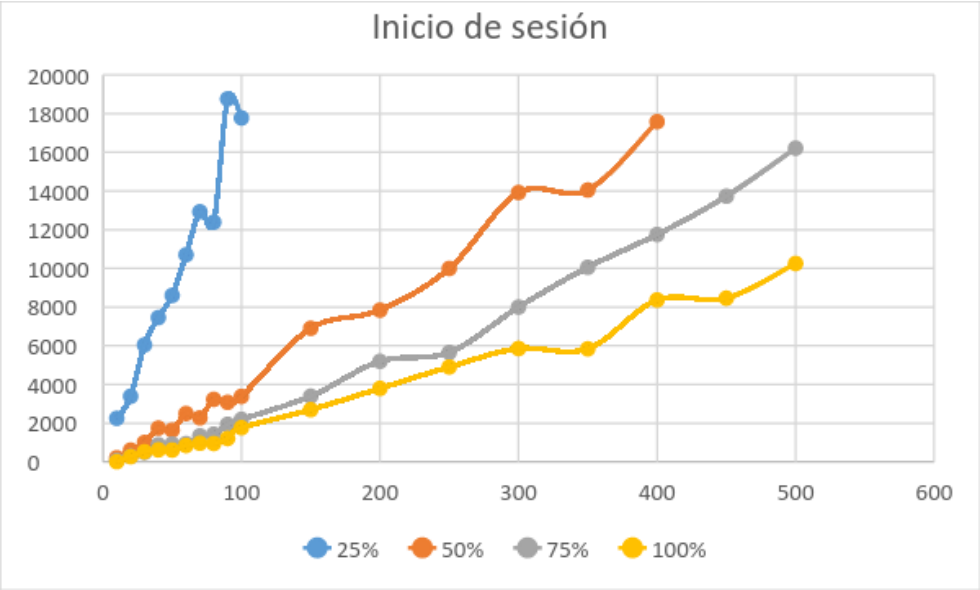


Figura 2

Luego se realizó una prueba de respuesta para el inicio de sesión, cuyos resultados pueden observarse en la Figura 2, debido a que existe un tiempo de consulta, entre la petición de acceso del host y la respuesta del servidor, los tiempos de cada capacidad de procesamiento se incrementan considerablemente. Sin embargo, puede notarse que los cambios más marcados se mostraron en los hilos con mayor capacidad de procesamiento, esto es debido a que el aumento considerable de tiempo se dio en la consulta de la base de datos y no en la capacidad de respuesta del hardware.

Al repetir las pruebas sobre un hardware Intel i5 4670k a 3.9Ghz, se denota una marcada mejora:

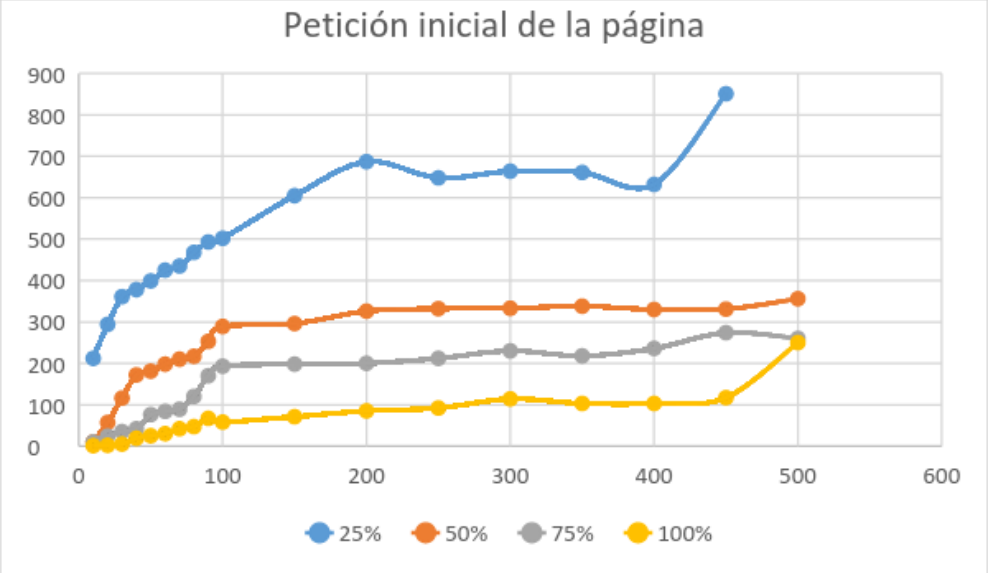


Figura 3

En la figura 3 (análoga a la Figura 1) es evidente que los tiempos de respuesta a las peticiones mejoran considerablemente, reduciéndose éstos hasta 20 veces, nótese que el límite superior de tiempo en la Figura 1 es de 20 000 ms, mientras que el mismo tiempo en la Figura 3 es de 900 ms.

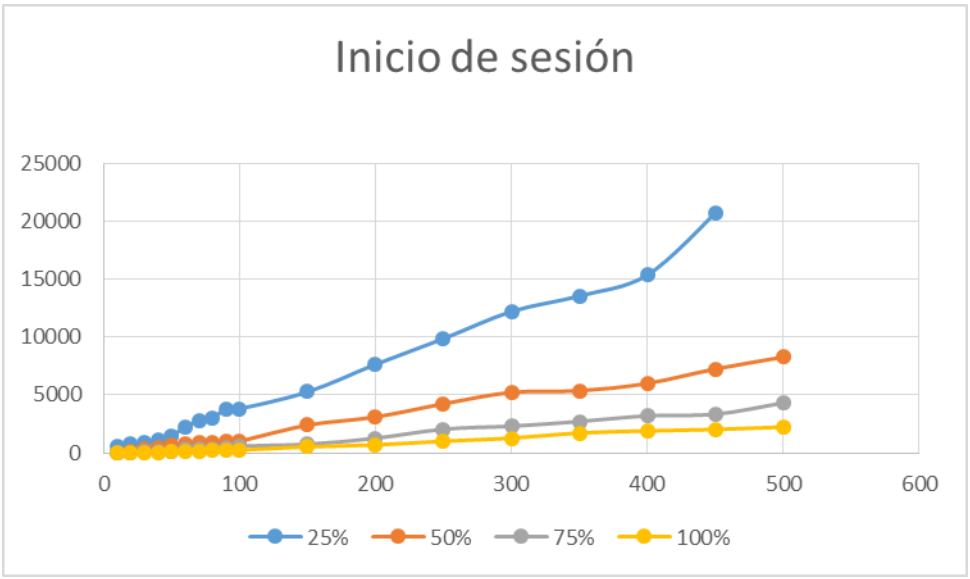


Figura 4

En la Figura 4 puede verse un comportamiento similar al de la Figura 3, sin embargo, dadas las características del hardware, los tiempos son considerablemente más reducidos. Cabe hacer la observación que la diferencia entre éstos y los de la Figura 2 no son tan marcados como los que se observaban entre las Figuras 1 y 3, lo cual se explica por los tiempos de consulta realizados a la base de datos para responder la petición de acceso.

Al superar las 500 peticiones por segundo, se observó que algunas de estas no se realizan, generando un porcentaje de error en peticiones. Este número es independiente al porcentaje de uso del procesador y evidencia la limitación que existe en la arquitectura existente.

En las tablas expuestas a continuación ilustran los resultados superiores a 500 peticiones por segundo, La prueba con el procesamiento al 25% fue obviada debido a que ésta tomaría demasiado tiempo para ser una prueba medible y no aportaría datos nuevos por lo cual se toman los siguientes datos:

Tabla 1

Número de hilos	Error en 50% de capacidad	Error en 75% de capacidad	Error en 100% de capacidad
600	16.6%	14.5%	13.3%
700	33.3%	26.6%	28.21%

Tabla 2

Número de hilos	Error en 50% de capacidad	Error en 75% de capacidad	Error en 100% de capacidad
600	24.9%	20.78%	18.2%
700	31.64%	31.66%	28.21%

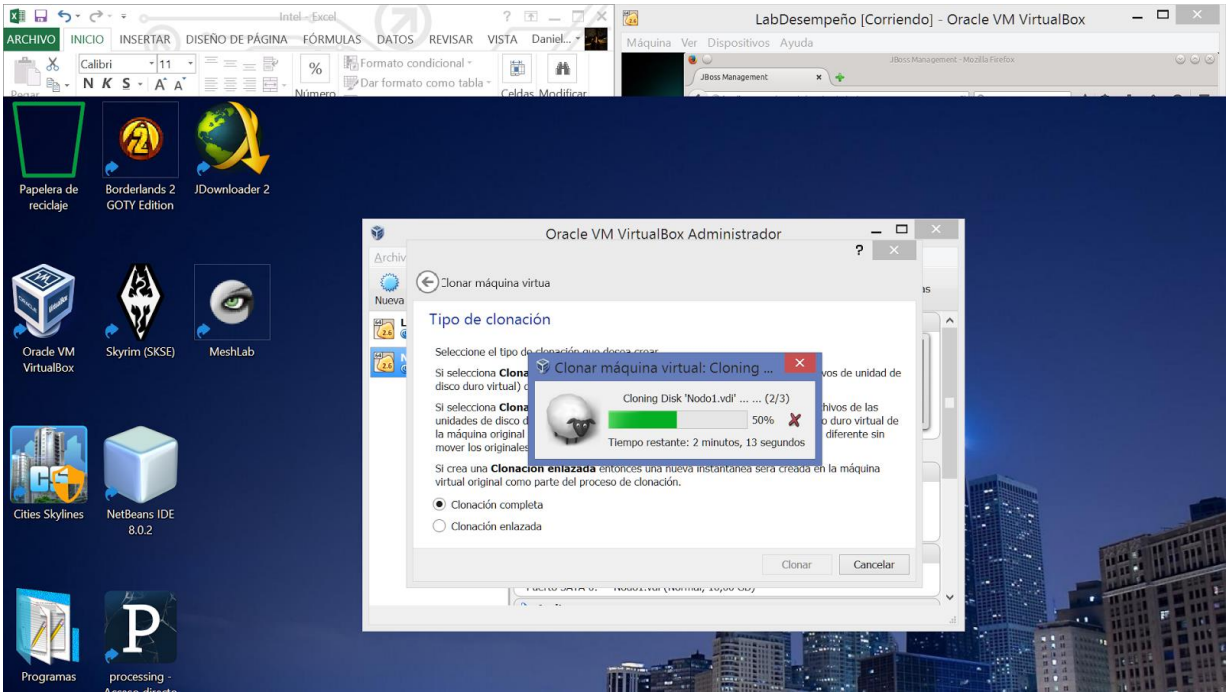
La Tabla 1 en éste caso, corresponde al equipo de sobremesa, de mayor capacidad, mientras que la Tabla 2 corresponde a la computadora portátil. Como se había esperado, se observa mayor cantidad de peticiones no respondidas en la computadora portátil, sin embargo, es evidente que si bien ambos resultados eran de esperarse, no es suficiente con un rendimiento marcadamente superior, para cumplir requerimientos superiores a 500 peticiones por segundo, ya que en ambos casos se producen errores, los cuales en ninguno de los dos casos superan el cuartil.

En lo último, cabe resaltar que si bien hay una marcada diferencia en rendimiento entre ambos equipos, está y la cantidad de peticiones que manejan es diferente, el porcentaje de peticiones no contestadas no sobrepasan el primer cuartil, lo cual se deba, posiblemente, a que ambos implementan la misma clase de arquitectura.

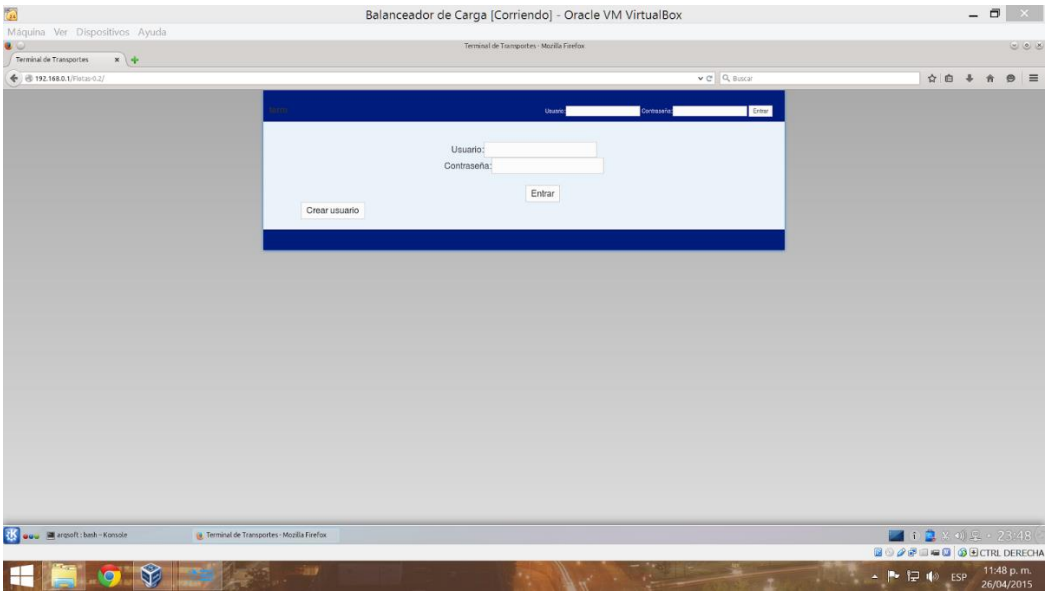
Aplicando escalabilidad horizontal

Terminadas las pruebas de desempeño, se procedió al cambio de la arquitectura, para la cual se implementó un balanceador de carga y dos nodos, con el fin de aumentar la cantidad de usuarios por segundo que puede soportar la plataforma.

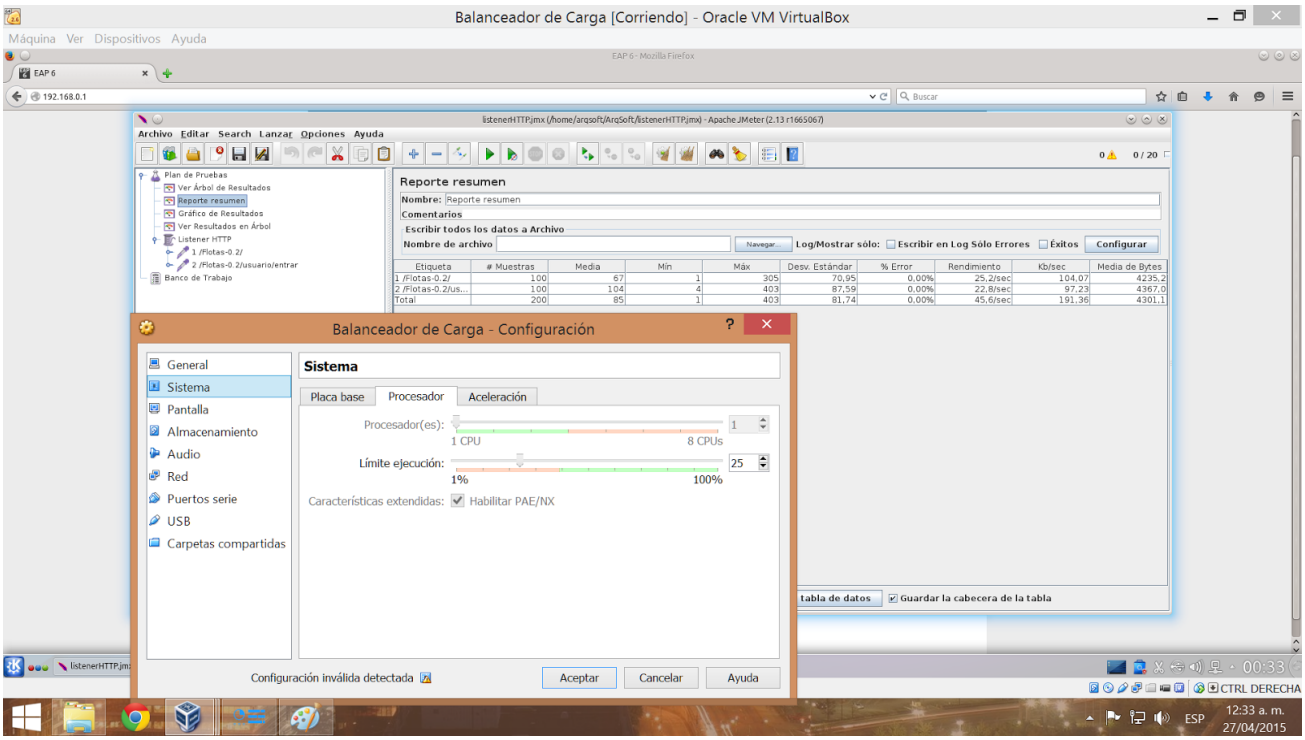
En primer lugar, se procedió a clonar la máquina virtual que se trabajó en las pruebas de desempeño para así tener las máquinas virtuales que harían el papel de nodos.



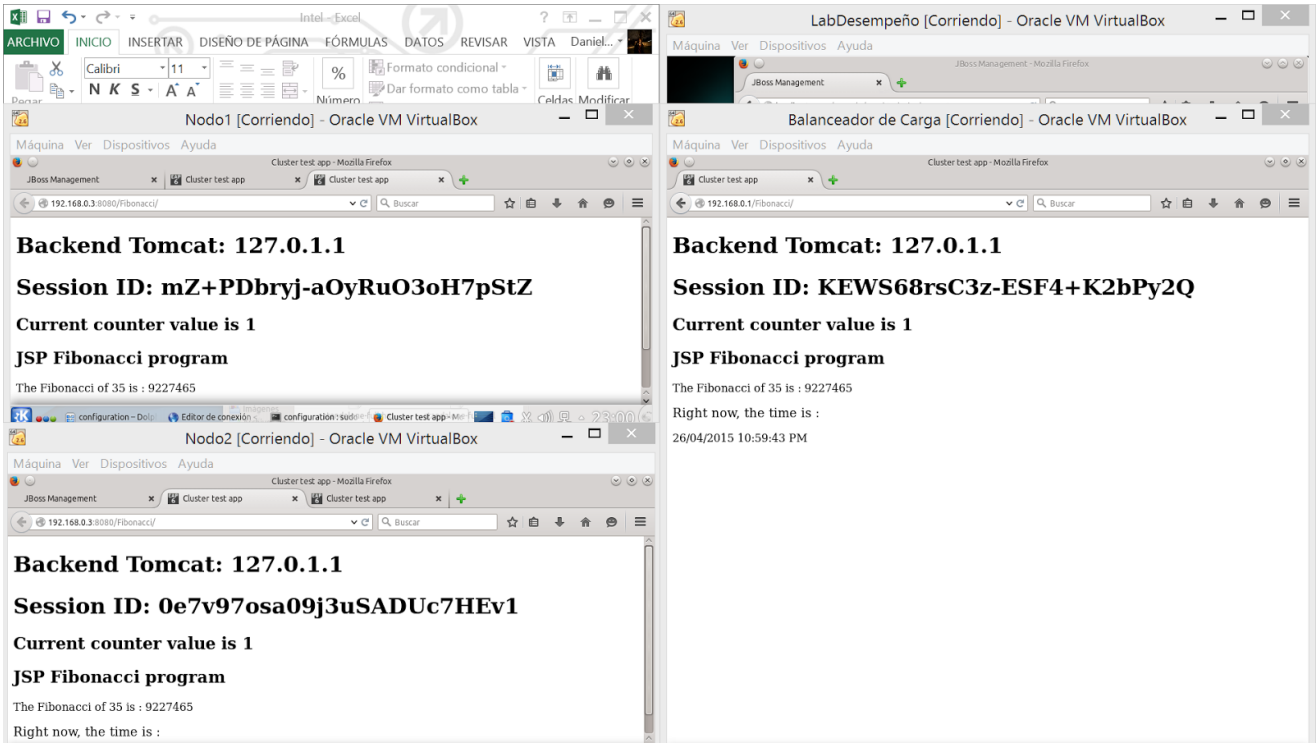
Con estas máquinas (dos de ellas) y el balanceador de carga, se procedió a configurar la interconexión, antes de ello se probó el funcionamiento del balanceador de carga con la aplicación que sería implementada en la plataforma:



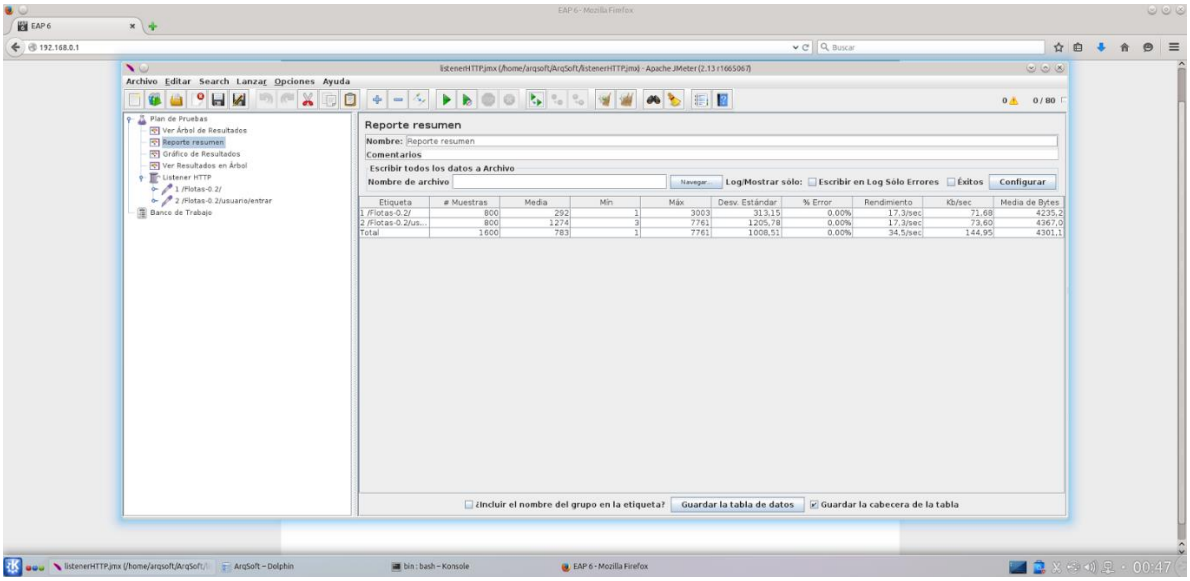
Luego se probó la aplicación con JMeter en la misma máquina virtual:



Después de haber comprobado el correcto funcionamiento del Balanceador de carga, se procedió a conectar éste con dos nodos clonados anteriormente. Una vez que se logró con éxito un ping entre los nodos y el balanceador, se procedió a probar el archivo WAR compartido en clase:



Y finalmente se realizó la prueba de nuestra aplicación, la cual se observa corriendo 80 hilos con una capacidad del 25% de procesamiento en ambos nodos y la misma capacidad en el balanceador de carga:



Tan solo en esta prueba, se observa una mejora en el tiempo de respuestas, ya que el tiempo de respuesta al loguearse era de 12387 ms, mientras que en la prueba que se muestra en la imagen fue de 1234 ms, diez veces más rápido.

Sin más preámbulo, se muestran los datos recabados en la prueba:

	1 nodo		2 nodos		
Hilos	Ingreso	Logueo	Ingreso	Logueo	error
10	212	525	38	74	0%
20	294	754	109	185	0%
30	361	841	148	344	0%
40	378	1163	187	387	0%
50	399	1448	184	425	0%
60	425	2265	213	666	0%
70	435	2748	288	766	0%
80	468	3000	291	1274	0%
90	493	3797	316	1316	0%
100	502	3818	333	1432	0%
150	605	5264	345	1631	0%
200	687	7639	420	2858	1,50%
250	648	9859	362	3081	3,56%
300	664	12206	447	2699	6,47%
350	661	13548	363	2939	11,60%
400	632	15357	370	3348	9,83%
450	851	20698	464	1624	18,53%
500			410	2935	22,28%

Tabla 3

En la Tabla 3 es posible observar el tiempo de respuesta para ambas pruebas (ingreso al sitio e intento de logueo), se observan los resultados tanto para un solo nodo como para ambos, siendo el que implementa los dos nodos el que tiene mejores resultados, y siendo ambos mejores en comparación con los de la prueba de desempeño del 25%.

Los gráficos ayudarán a ilustrar mejor los datos observados en el la Tabla 3, para ello se mostrará el desempeño de cada prueba y luego la comparación entre ambas:

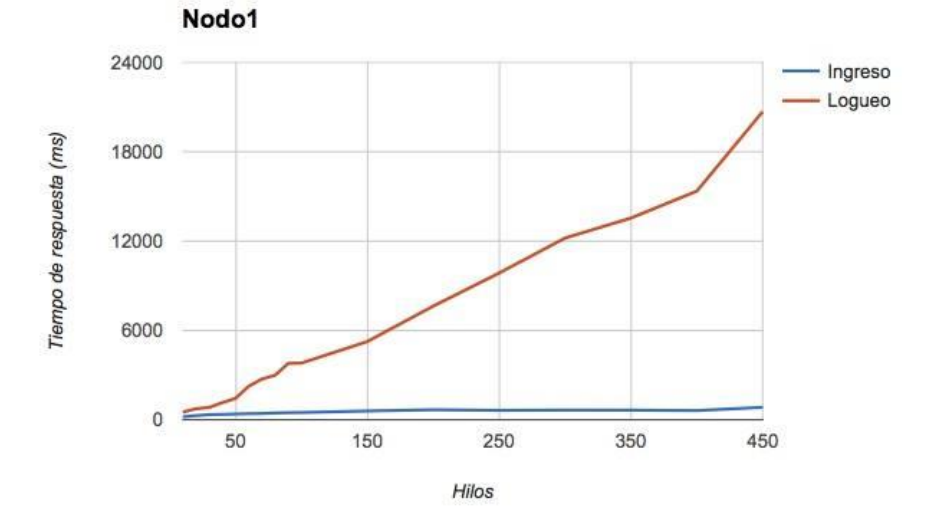


Figura 5

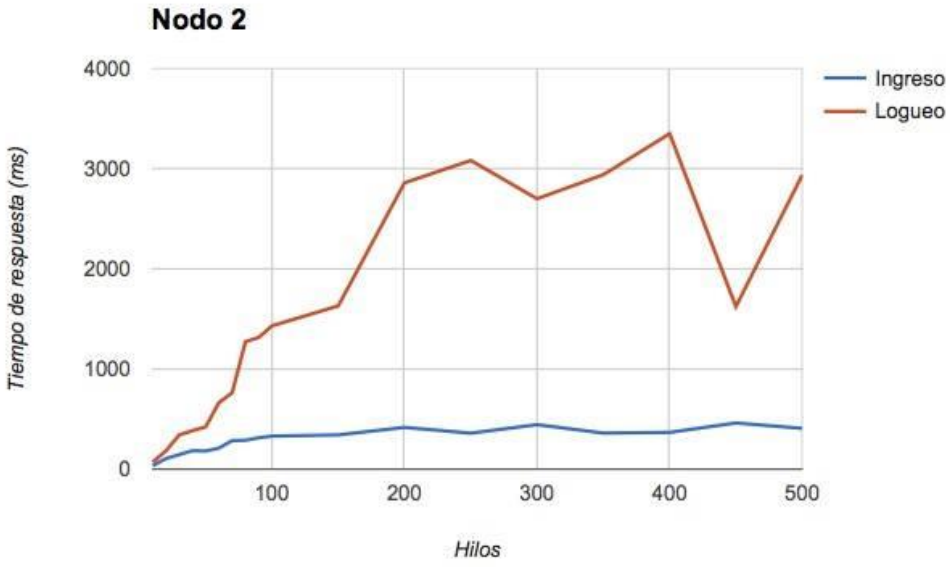


Figura 6

Se observa en la Figura 5 y 6, que, como en la prueba de desempeño, sin embargo es evidente una extraña fluctuación en la prueba de intento de logueo cuando se utilizan ambos nodos, esto será mencionado en mayor detalle más adelante.

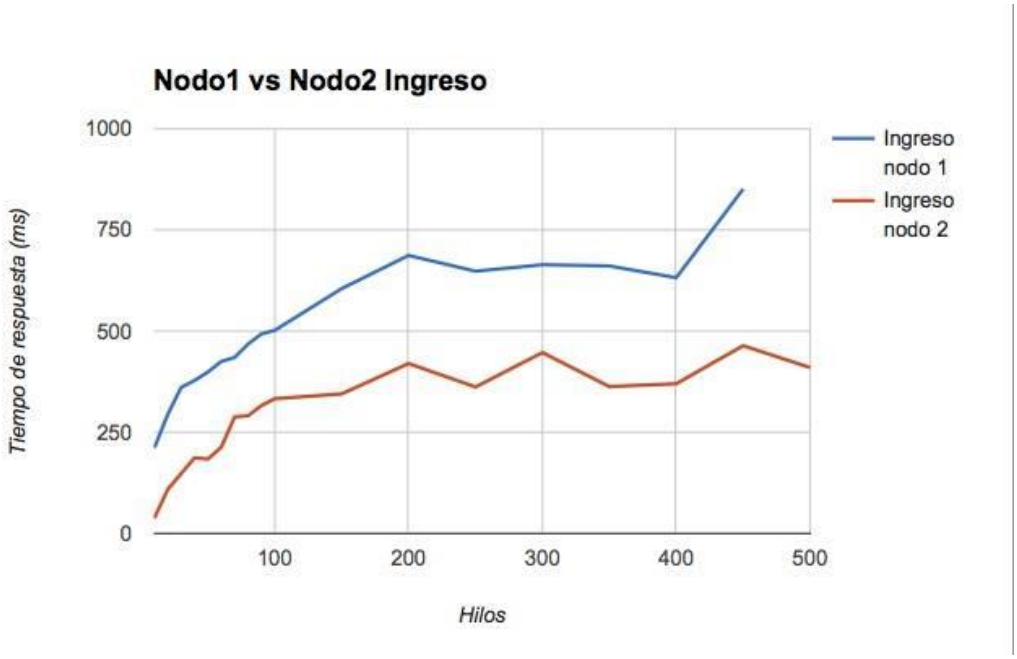


Figura 7

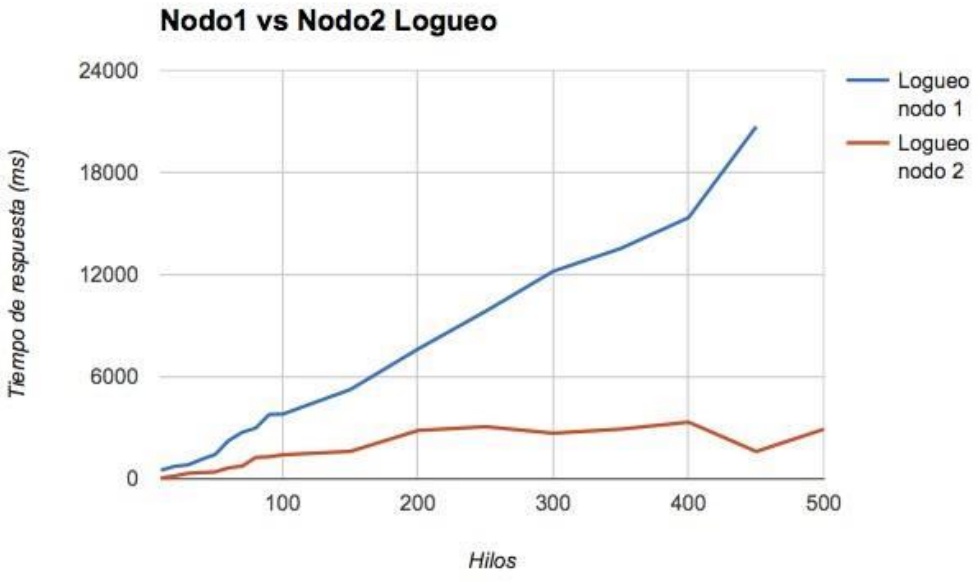


Figura 8

En las Figuras 7 y 8, se pueden observar los desempeños en las pruebas de acceso a la aplicación e intento de logueo respectivamente, se hace más evidente la mejora de desempeño al aumentar los nodos.

Pese a los resultados mostrados anteriormente, se ha observado una anomalía a partir de la Figura 6 (Nodo 2) en la cual hay una abrupta disminución del tiempo de respuesta. Si se observa las Tabla 3, se observa que al implementar 200 hilos en la prueba con 2 nodos, se comienzan a presentar errores que crecen a partir de dicho punto.

Los errores, son peticiones no procesadas, esto indica que a partir de los 200 hilos las respuestas tienden a ser más rápidas y en las gráficas en las que se visualiza la prueba hecha con 2 nodos se desacelera el crecimiento de los tiempos de respuesta e incluso decrecen dichos tiempos en algunos intervalos.