

Sistema de Seguimiento Cardíaco

Juan Sebastián Barragán Jerónimo, Sebastián Sánchez Galiano, Santiago Rojas
Herrera

Experimento 2 – Entrega 1

{s.rojas19, js.barragan796, s.sanchez59}@uniandes.edu.co

Fecha de presentación: 25 de marzo de 2017

Bogotá, Colombia

Tabla de contenidos

Pre-experimentación	1
Problemática	1
Objetivo del experimento	1
Descripción del experimento	1
Artefactos a construir	1
Recursos de la experimentación	1
Resultados esperados	2
Duración y etapas	2
Experimentación	2
20 iteraciones de (#threads/20) peticiones en 1 segundo	2
1 iteración de n peticiones en 1 segundo	4
Post-experimentación	6
Duración real	6
Artefactos contruidos	6
Análisis	6
Conclusiones	6

1. Pre-experimentación

1.1. Problemática

Sobre el sistema desarrollado para el manejo de servicios para pacientes del Hospital Cardiológico de Santa fé, se debe implementar un sistema balanceador de carga, que permita la replicación de nodos que despliegan la aplicación, y que distribuya las solicitudes entre estos.

1.2. Objetivo del experimento

Con la implementación del balanceador de carga y el uso de réplicas de computación, se espera mejorar la disponibilidad de los servicios, y observar las implicaciones que esta decisión pueden llevar al desempeño de la aplicación

1.3. Descripción del experimento

En el documento se expondrán pruebas de carga para el envío de mediciones desde un dispositivo externo hacia el servidor de la aplicación. Para esto, se hará uso de la herramienta *JMeter*, que permite simular el envío de múltiples peticiones (en este caso http) hacia la aplicación.

Estas pruebas se harán para la aplicación desplegada a un nodo (conexión directa), y hacia el balanceador de carga, con fin de comparar ambos resultados.

1.4. Artefactos a construir

Con el uso de *nginx* se desplegará un balanceador de carga, configurandolo de forma que pueda distribuir las peticiones entre los distintos nodos desplegados, y que pueda usar un nodo de *backup* en caso de fallo

1.5. Recursos de la experimentación

- Balanceador de carga y nodo de backup
 - Intel Xeon E5-2630 2.6GHz (4 núcleos)
 - 12GB Ram
- Nodos 1 y 2
 - Intel Xeon E5-2630 2.6GHz (4 núcleos)
 - 12GB Ram

- Nodo 3
 - Intel Xeon E5-2680 2.6GHz (2 núcleos)
 - 4GB Ram

1.6. Resultados esperados

Se espera que el desempeño de la aplicación sea comprometido por el incremento de overhead y por los resultados dados en el laboratorio 6, que concluyeron un decremento del desempeño en el sistema, sin importar la técnica de manejo de peticiones usada (*round-robin o least-connection*)

1.7. Duración y etapas

Actividad	Tiempo de desarrollo
Implementación de sistema balanceador de carga	~5 horas
Pruebas de carga	~1.5 horas
Documentación pruebas	~1 horas

2. Experimentación

- 20 iteraciones de (#threads/20) peticiones en 1 segundo

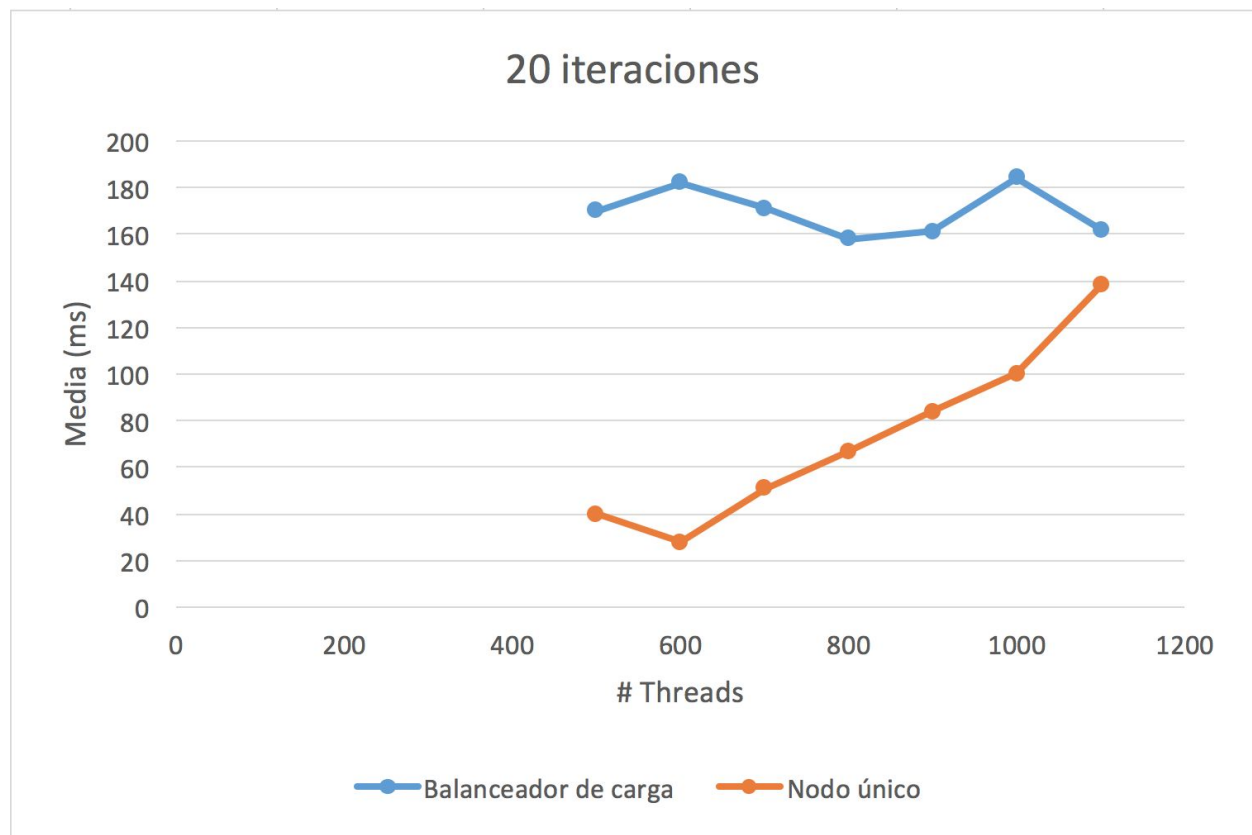
Nodo único

# thread	Media (ms)	Error (%)	Desv. Estandar
1100	138	0	61.01
1000	100	0	63.00
900	84	0	61.62
800	67	0	34.52
700	51	0	27.22
600	28	0	11.06
500	40	0	17.49

Balanceador de carga

# thread	Media (ms)	Error (%)	Desv. Estandar
1100	162	0.00	18.3
1000	184	0.00	16.6
900	161	0.00	14.9
800	158	0.00	13.3
700	171	0.00	11.6
600	182	0.00	10.0
500	170	0.00	8.3

Comparación resultados



- 1 iteración de n peticiones en 1 segundo

Nodo único

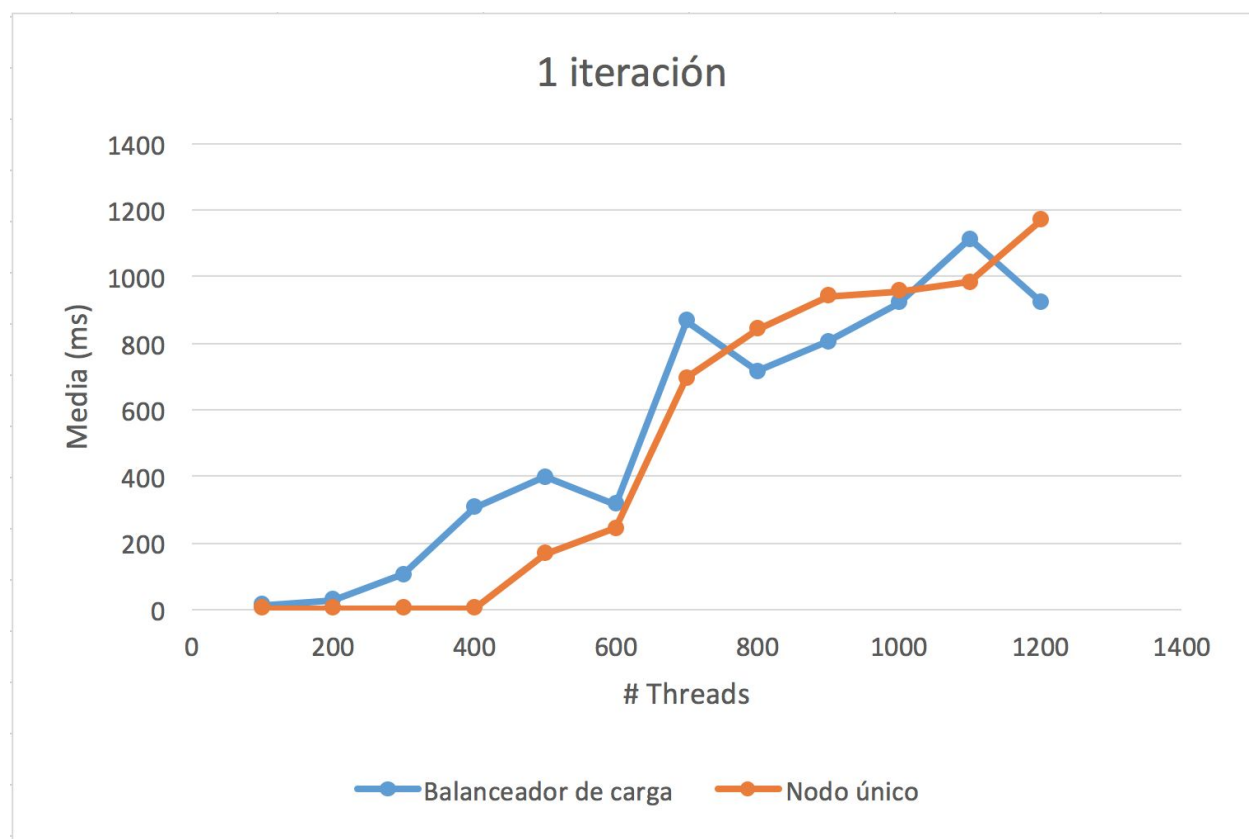
# thread	Media (ms)	Error (%)	Throughput (op/s)
1200	1169	0	305.5
1100	984	0	289.3
1000	955	0	286.6
900	942	0	284.3
800	841	0	290.1
700	696	0	298.0
600	243	0	308.6
500	166	0	318.1
400	2	0	266.3
300	3	0	294.1
200	2	0	173.9
100	3	0	87.8

Balanceador de carga

# thread	Media (ms)	Error (%)	Throughput (op/s)
1200	923	31.00	331.8
1100	1112	41.18	18.3
1000	921	18.70	301.1
900	805	4.22	366.4
800	715	0.00	306.4

700	867	1.57	244.1
600	317	0.00	370.6
500	398	0.00	124.9
400	305	0.00	241.8
300	104	0.00	270.0
200	29	0.00	179.2
100	12	0.00	97.9

Comparación resultados



3. Post-experimentación

3.1. Duración real

Actividad	Tiempo de desarrollo	Δ
Implementación de sistema balanceador de carga	~4.5 horas	~0.5 horas
Pruebas de carga	~1.5 horas	~0 horas
Documentación pruebas	~1 horas	~0 horas

3.2. Artefactos construidos

Todos los artefactos propuestos fueron construidos.

3.3. Análisis

Los resultados de la experimentación, evidencian un decremento considerable del desempeño de la aplicación; En el caso del envío de peticiones en 20 iteraciones en 1 segundo, se puede percibir que la latencia del balanceador de carga siempre es mayor a la de la aplicación desplegada en un único servidor, aunque ambas alternativas se comportan adecuadamente bajo las restricciones de negocio; Por otro lado, cuando se envían todas las peticiones en una iteración en 1 segundo, se nota que la implementación con el balanceador de carga tiene errores desde las 700 peticiones, incluso si mantiene una latencia similar a la implementación de un solo nodo (que responde satisfactoriamente hasta 1200 peticiones).

3.4. Conclusiones

En conclusión, la implementación de un balanceador de carga junto al uso de técnicas de replicación para mejorar la disponibilidad de la aplicación tienen un impacto negativo en el desempeño de la aplicación.

Puede que el uso de otra implementación para el balanceador de carga, o el mejoramiento de la configuración en la usada (*nginx*) disminuya el impacto en desempeño.