



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA

1c2021

75.61 - TALLER DE PROGRAMACIÓN III

Trabajo práctico n.1: Sitio Institucional
Documento de pruebas de carga

INTEGRANTES:

Rozanec, Matías - mrozanec@fi.uba.ar - #97404

Índice

1. Introducción	2
2. Pruebas	2
2.1. Prueba 1	2
2.2. Prueba 2	3
2.3. Prueba 3	3
2.4. Prueba 4	5
2.5. Prueba 5	6
3. Conclusiones	8

1. Introducción

En el presente documento se propone mostrar los resultados de las pruebas de carga realizados sobre el sistema desarrollado. Durante la ejecución de las pruebas realizadas se ha intentado mantener una línea lógica que haga que las distintas pruebas sean comparables entre sí y que permitan sacar conclusiones del sistema: de nada serviría llevar a cabo un set de pruebas desconexas.

2. Pruebas

Si bien se han ido realizando varias pruebas preliminares para familiarizarse con el entorno y poder tener evaluaciones parciales, en este documento se incluyen las últimas cuatro realizadas solamente, con la intención de dejar la imagen más clara posible sin abundar en casos que de haberlos incluido probablemente opacarían el panorama dificultando así la interpretación de los resultados.

En todos los casos se ha intentado correr las pruebas más largas posibles: la duración fue en todos los casos medida en base a la cantidad de escrituras en Datastore, que tiene un límite diario gratuito de 20k escrituras.

2.1. Prueba 1

En esta primera prueba no se ha limitado la cantidad de instancias, pero se ha decidido acceder solamente a las páginas crudas, es decir, sin contador ni recursos adicionales. Se puede observar que el load de cada página conlleva unos 2.7kb, es decir que son páginas livianas.

La duración de la prueba ha sido de **aprox. 5 minutos**: es una duración relativamente corta si se compara con la expectativa de unos 15 o 20 minutos, pero ha sido suficiente para poder observar la evolución del sistema y las distintas etapas por las que ha pasado. En la Figura 2.2 se puede ver cómo los primeros requests son los que más tardan, normalizándose luego la situación entre los 200 y 300 ms, un tiempo más que aceptable. Los RPS han ido aumentando linealmente hasta llegar a los 150. En la figura 2.3 se puede ver que no ha habido errores. Esto se debe, probablemente al manejo de instancias que ha hecho AppEngine de forma transparente: a las 23:18 se ha llegado a tener 11 instancias corriendo. Esto coincide con el inicio del último minuto de la prueba de carga.



Figura 2.1: Total requests per second de prueba 1.

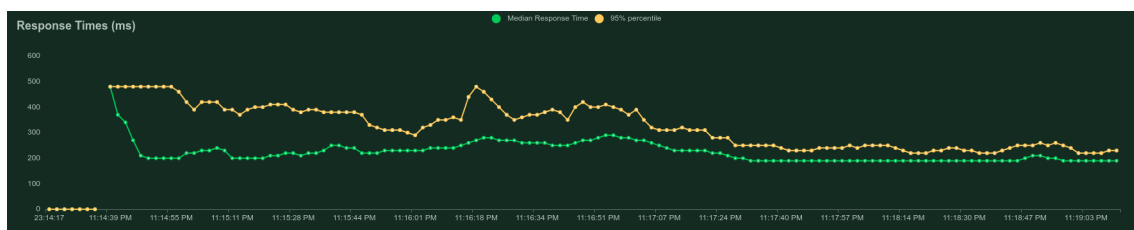


Figura 2.2: Response times de prueba 1.

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	5020	0	200	280	223	178	3675	2716	36.7	0
GET	/about	4951	0	200	280	225	178	4386	2717	32.7	0
GET	/about/legals	4924	0	200	280	226	178	4372	2724	33.7	0
GET	/jobs	4930	0	200	280	221	178	4370	2716	35.9	0
Aggregated		19825	0	200	280	224	178	4386	2718	139	0

Figura 2.3: General stats de prueba 1.

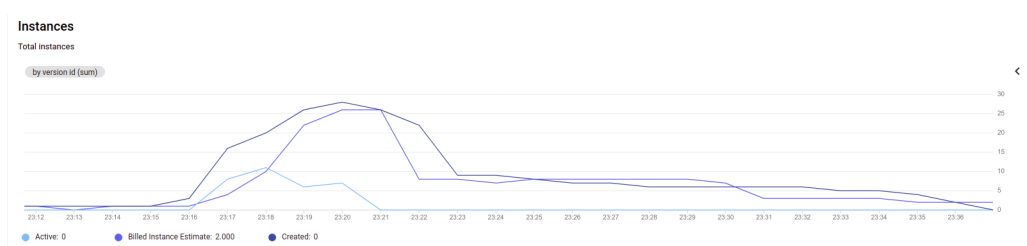


Figura 2.4: Number of instances used de prueba 1.

2.2. Prueba 2

En este caso se han incluido los accesos a recursos compartidos. Se puede ver rápidamente que el pdf de 3.3 mb esta matando el tiempo de rta. y está cargando más de la cuenta al server. Como esto normalmente reside en algo símil AWS s3, es decir, un gestor de almacenamiento dedicado a servir este tipo de recursos, con réplicas y CDNs, no tiene sentido mantener esta carga para el test de carga. Después de todo, lo que nos interesa es que esto responda rápido lo que tiene que salir sí o sí de AppEngine. No es el caso con recursos de gran tamaño: nunca en un escenario real, menos si se quiere que sea escalable, se va a dejar eso hosteado ahí. La duración de la prueba ha sido mínima.

En retrospectiva, hubiese sido interesante prolongar el test para ver su evolución. Lo que hay llevado a su suspensión fue el hecho que en las estadísticas se veía un clarísimo cuello de botella con el recurso, donde la mediana del tiempo de respuesta del pdf superaba por 36 veces el de un html común o archivo estático de menor tamaño (imagen de unos 50kb).

2.3. Prueba 3

En esta prueba se decide acceder a recursos estáticos, pero se omite el pdf de 3.3mb, logrando así un acceso a recursos con un peso total de unos 200kb, distribuidos de la siguiente manera: imagen de Martha Argerich pesa unos 50kb, un favicon pesa unos 150kb.

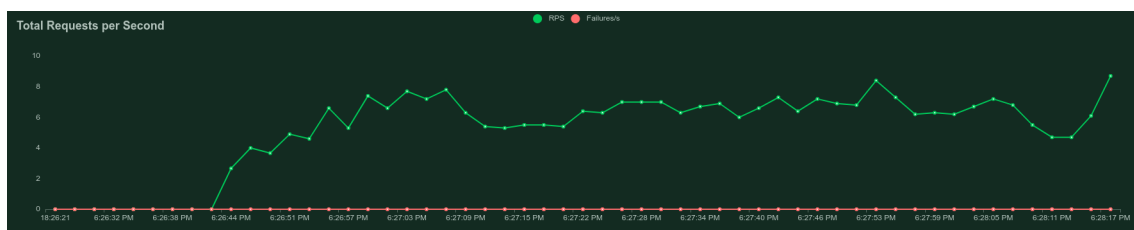


Figura 2.5: Total requests per second de prueba 2.

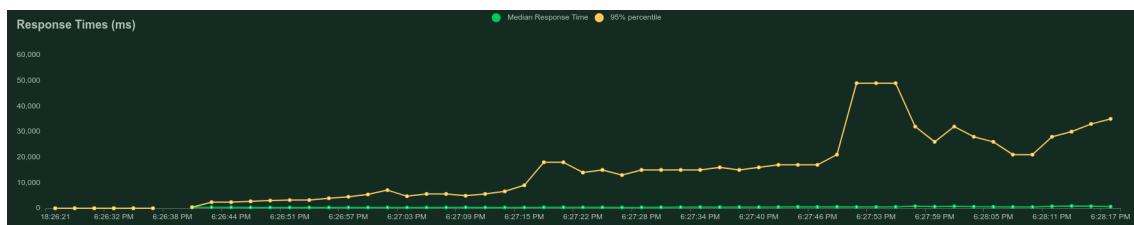


Figura 2.6: Response times de prueba 2.

Type	Name	# Requests	# Fails	Median (ms)	90thile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/hear-tech-wsg-bridge-for-dante.pdf	152	0	11000	30000	14948	1264	79917	3296283	2.6	0
GET	/about/legals	72	0	310	1600	728	187	11844	2724	0.7	0
GET	/	63	0	350	1300	590	188	2936	2716	1.2	0
GET	/about	52	0	400	1300	658	189	3335	2717	0.6	0
GET	/jobs	69	0	300	1100	512	184	4981	2716	0.9	0
GET	/argerich.jpeg	254	0	350	990	610	185	8504	31583	3.3	0
Aggregated		662	0	450	13000	3907	184	79917	770020	9.3	0

Figura 2.7: General stats de prueba 2.

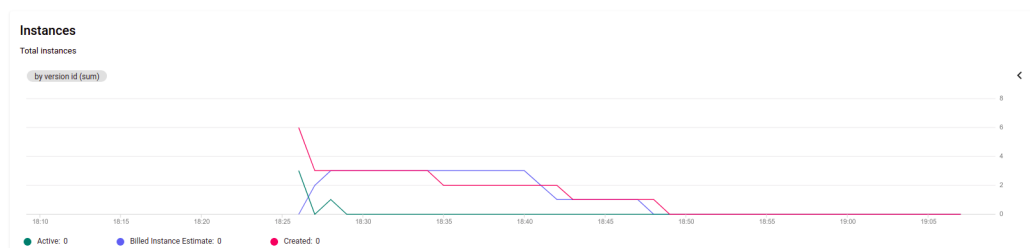


Figura 2.8: Number of instances used de prueba 2.

Si se observa el gráfico de RPS en la figura 2.9, se puede ver que los valores no han sido tan altos: se ha logrado un pico de 100, pero el mismo ha pasado a oscilar entre 60 y 80.

En la figura 2.10 se puede observar que la mediana y el percentil 95 de los tiempos de respuesta van presentando diferencias cada vez mayores. Si bien el p95 tiene un pico de unos 80 segundos y presenta tiempos relativamente altos, la mediana se ha mantenido por debajo de los 1500ms durante toda la prueba de carga, lo que indica que los resultados fueron en general muy buenos.

Si se observan las estadísticas en la Figura 2.11 se puede ver que el p95 del gráfico anterior está ampliamente sesgado por el del favicon que es el recurso más pesado. Del mismo modo, es el promedio y mediana del tiempo de respuesta del resto de los recursos entrega tiempos muy favorables.

Según la figura 2.12, a las 22.05 ha habido 5 instancias activas. Llama la atención que no se hayan levantado más instancias, considerando que la prueba corrió de 22.00 a 22.15 aprox., es decir que el máximo de instancias se dio en el primer tercio de la prueba, y luego incluso descendió, mostrando nuevos picos de 3 instancias.

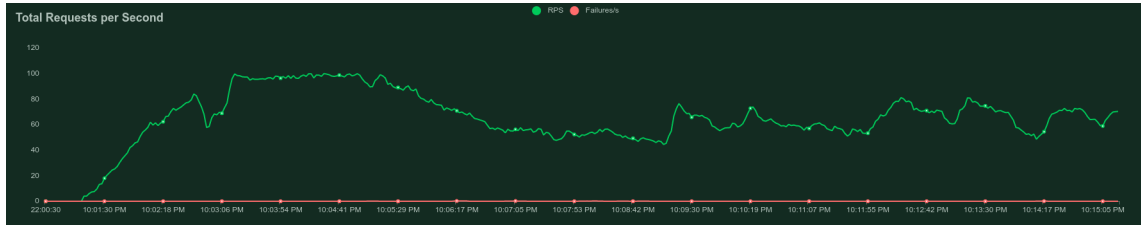


Figura 2.9: Total requests per second de prueba 3.

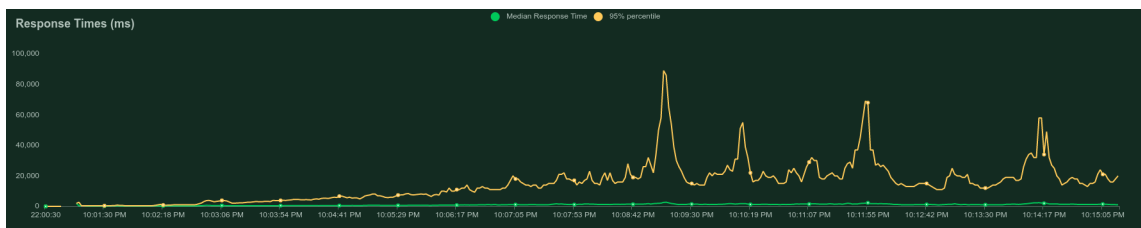


Figura 2.10: Response times de prueba 3.

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	4853	3	250	900	635	182	193540	2714	5.8	0
GET	/about	4820	2	250	890	618	182	115081	2716	5.9	0
GET	/about/legals	4708	2	250	910	608	183	118127	2723	6.4	0
GET	/argenich.jpeg	19118	1	810	3700	2383	168	240310	31581	21.9	0
GET	/favicon.ico	18747	0	3800	15000	8989	324	277582	165989	22.4	0
GET	/jobs	4818	3	260	900	684	183	146783	2714	5.9	0
Aggregated		57064	11	780	7400	3966	168	277582	66026	68.3	0

Figura 2.11: General stats de prueba 3.

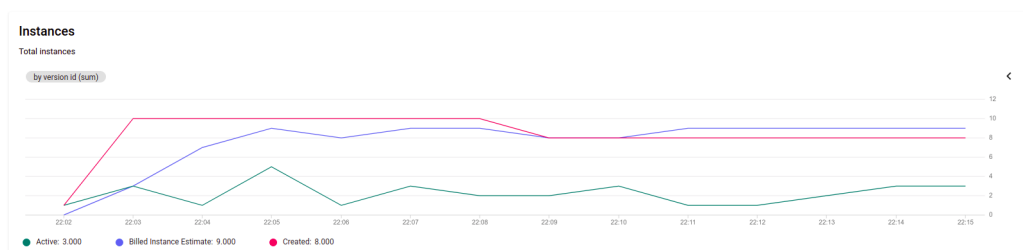


Figura 2.12: Number of instances used de prueba 3.

2.4. Prueba 4

En esta prueba ya se ha incluido el acceso a los contadores, omitida hasta este momento, ya haciendo uso de la cache. Se han mantenido los accesos a los recursos estáticos.

Observando la figura 2.13 se puede ver que la cantidad de requests por segundo se ha mantenido alrededor de 120, con pequeños valles que llegan a los 90 aprox.

En la figura 2.14 se puede ver una marcadísima diferencia entre el percentil 95 y la mediana, cosa que además se comprueba en la figura siguiente: los tiempo de la mediana son aceptables en general, teniendo nuevamente el favicon el valor más grande, de unos 4 segundos.

Nuevamente se observa que los recursos de pequeño tamaño no presentan ningún problema al servicio (Figura 2.15).

En Fig. 2.16 se puede ver que se ha utilizado un máximo de 5 instancias, llegando al pico a las 22.06, mientras que la prueba corrió hasta las 22.12.



Figura 2.13: Total requests per second de prueba 4.

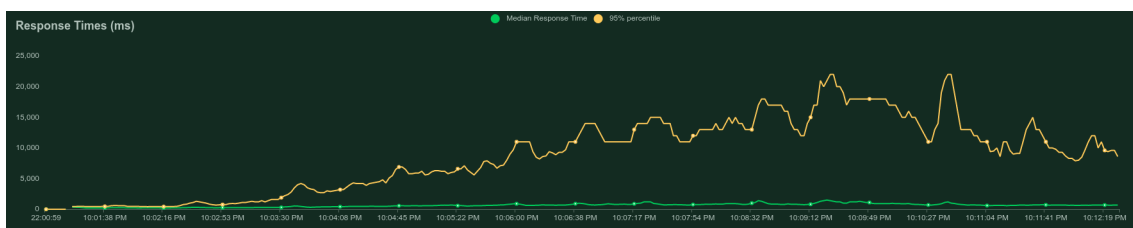


Figura 2.14: Response times de prueba 4.

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	4612	0	300	850	540	183	64633	2716	8.1	0
GET	/about	4790	0	290	780	532	185	65697	2717	9.8	0
GET	/about/legals	4532	0	300	870	551	184	105836	2724	7.9	0
GET	/api/counter/about-counter	4680	0	230	530	340	164	53520	5	8.2	0
GET	/api/counter/about-legals-counter	4421	0	230	520	346	165	34814	5	8.4	0
GET	/api/counter/home-counter	4500	0	230	530	347	164	38663	5	9.1	0
GET	/api/counter/jobs-counter	4453	0	230	530	339	164	63608	5	8	0
GET	/argenich.jpg	18395	0	1000	3200	1712	170	134255	31583	32.7	0
GET	/favicon.ico	18066	0	4200	15000	6669	329	205714	165989	33.8	0
GET	/jobs	4550	0	300	860	515	184	26840	2716	7.7	0
Aggregated		72999	0	540	5900	2302	164	205714	49728	133.7	0

Figura 2.15: General stats de prueba 4.

2.5. Prueba 5

En esta prueba se ha limitado la cantidad de instancias a 1, lo que se puede comprobar en la Figura 2.20. Si bien el servicio nunca se cae, se puede observar que por momentos hay requests que fallan. Los tiempos de respuesta son relativamente elevados también (aunque no exageradamente). Sí se pueden ver valores de RPS

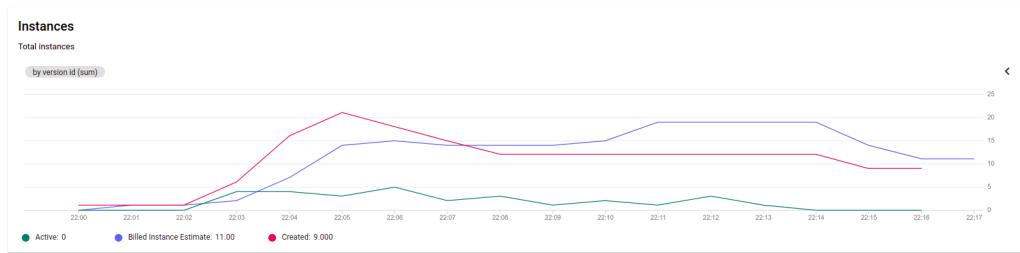


Figura 2.16: Number of instances used de prueba 4.

bastante más bajos: mientras que en los casos anteriores se ha llegado a tener el doble o un poco más, además de contar con valores levemente más estables.

En líneas generales, lo que se puede decir de la limitación a una sola instancia es que hace que la prueba sea bastante menos estable: hay requests que no pueden ser atendidas y se pierden (fig. 2.17), y las requests que sí se logran hacer sin problema presentan tiempos de respuesta elevadísimos en comparación con los casos anteriores. Notar también en la figura 2.18 que casi no hay diferencia entre la mediana y el p95.

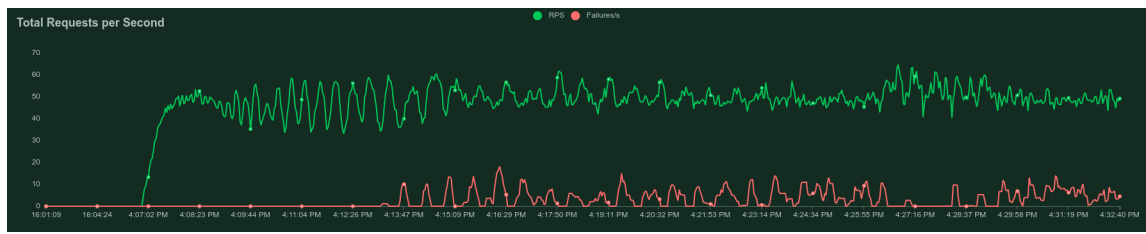


Figura 2.17: Total requests per second de prueba 5.

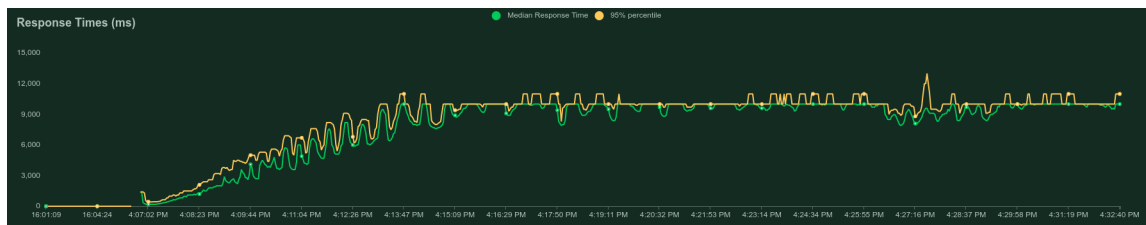


Figura 2.18: Response times de prueba 5.

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	4856	443	9600	10000	8200	181	12397	2498	2.6	0.2
GET	/about	4686	460	9600	10000	8163	180	12110	2482	2.6	0.2
GET	/about/legals	4817	449	9600	10000	8168	180	11281	2500	2.2	0
GET	/api/counter/about-counter	4602	200	9200	10000	7916	164	15972	19	3.3	0
GET	/api/counter/about-legals-counter	4736	215	9300	10000	7970	165	17667	19	3.4	0
GET	/api/counter/home-counter	4785	210	9200	10000	7945	163	11117	19	2.3	0
GET	/api/counter/jobs-counter	4695	211	9200	10000	7956	163	12707	19	3	0.1
GET	/argerich.jpeg	19033	1286	9500	10000	8224	169	16268	29471	12.5	1
GET	/favicon.ico	18941	877	9500	10000	8205	324	21809	158318	15.6	0.6
GET	/jobs	4774	407	9600	10000	8176	180	12129	2512	2.6	0.2
Aggregated		75925	4758	9400	10000	8139	163	21809	47518	50.1	2.3

Figura 2.19: General stats de prueba 5.



Figura 2.20: Number of instances used de prueba 5.

3. Conclusiones

Partiendo de la experiencia de la materia Sistemas Distribuidos I, en el presente TP se ha podido aplicar dicho conocimiento de la mano de herramientas utilizadas en la industria: en cierta forma se ha logrado obtener una visión más real del desarrollo de los sistemas distribuidos. Adicionalmente, realizar pruebas de carga brinda una nueva herramienta: hasta ahora hemos monitoreados los sistemas observando el comportamiento de colas en RabbitMQ; en este trabajo se ha dado el paso a tener que exigir a los sistemas de una forma distinta, más realista quizás, y teniendo acceso a herramientas adicionales.