

Latencia Promedio vs. Iteración - Lectura/Escritura (25/75)

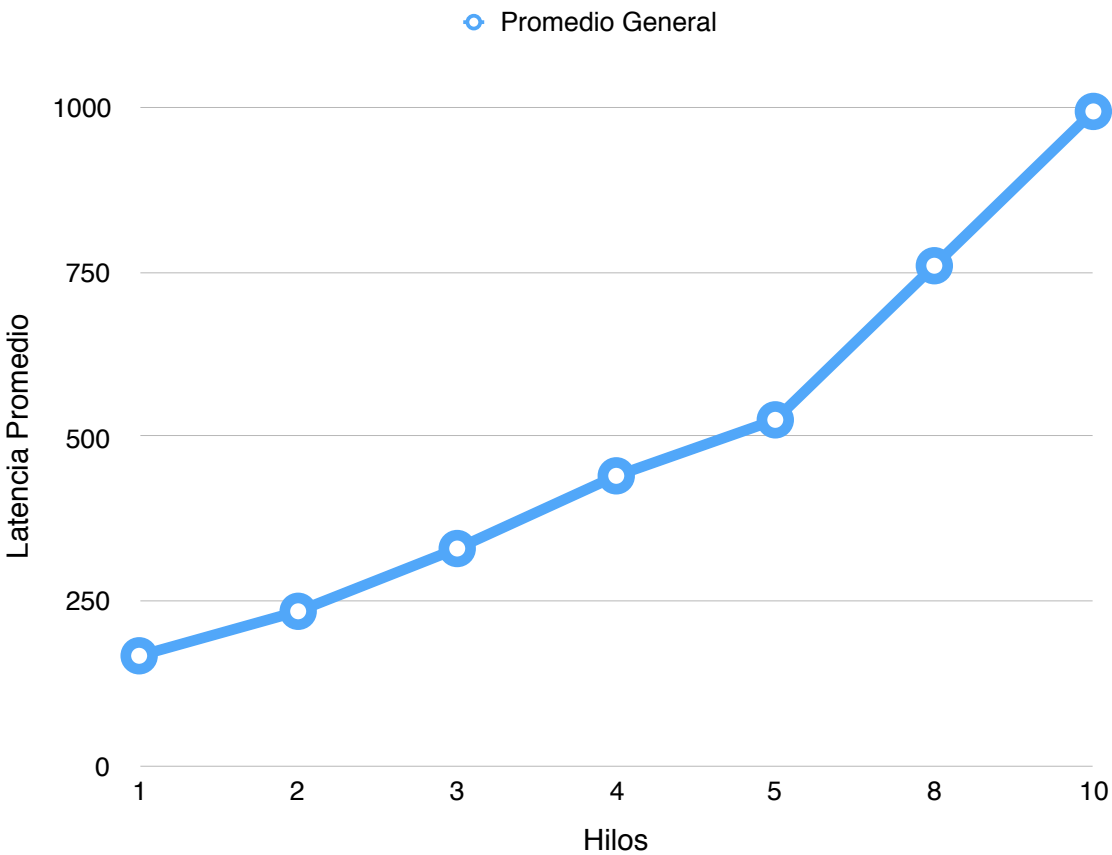
Lectura 25%	1	2	3	4	5	8	10
1	114.74081469648563	168.88021044111696	235.1426905132193	306.1524193548387	347.04784688995215	509.350179068842	684.7210300429184
2	111.94056748466258	164.47498014297062	222.13830211239537	288.3389898989899	357.2155804087929	422.1932644469958	706.9468599033817
3	112.13498622589532	166.69754329440192	215.8454404945904	331.72459560348403	394.53788509575355	514.883445278546	739.1808723489396
4	117.78434442270058	166.36150793650793	243.7664912990692	297.73005160778087	365.3008356545961	522.9584627329192	625.8667976424362
5	110.81067393844955	147.99209173586397	223.6345929100117	289.9911368015414	335.92127573677834	502.9562706270627	607.7833663366337
	114.740814696486	168.880210441117	235.142690513219	306.152419354839	347.047846889952	509.350179068842	684.721030042918
	111.940567484663	164.474980142971	222.138302112395	288.33898989899	357.215580408793	422.193264446996	706.946859903382
	112.134986225895	166.697543294402	215.84544049459	331.724595603484	394.537885095754	514.883445278546	739.18087234894
	117.784344422701	166.361507936508	243.766491299069	297.730051607781	365.300835654596	522.958462732919	625.866797642436
	110.81067393845	147.992091735864	223.634592910012	289.991136801541	335.921275736778	502.956270627063	607.783366336634
Promedio	113.482277353639	162.881266710172	228.105503465857	302.787438653327	360.004684757175	494.468324430873	672.899785254862

Table 1

Escritura 75%	1	2	3	4	5	8	10
1	217.79042155816435	308.7594634081551	418.8539310716209	565.924335106383	722.7613454351308	1078.2835581674904	1379.0324055398683
2	215.45955086580088	303.7139802191927	456.7818715792284	572.5043189368771	719.4464695558256	905.8418843914985	1369.9405268759979
3	213.20686419091032	292.43235333244644	405.0064759848894	573.1017261826328	691.0935772571729	1078.7462846431918	1311.4776696440474
4	230.20792478173271	329.401871657754	444.6274405631558	588.022590562759	667.9572592493656	1063.0391971982758	1283.7598926894702
5	217.85712363783128	290.11711952884485	427.7407507063097	585.8848075624578	649.5788913997076	996.969640971489	1233.831170568562
	217.790421558164	308.759463408155	418.853931071621	565.924335106383	722.761345435131	1078.28355816749	1379.03240553987
	215.459550865801	303.713980219193	456.781871579228	572.504318936877	719.446469555826	905.841884391499	1369.940526876
	213.20686419091	292.432353332446	405.006475984889	573.101726182633	691.093577257173	1078.74628464319	1311.47766964405
	230.207924781733	329.401871657754	444.627440563156	588.022590562759	667.957259249366	1063.03919719828	1283.75989268947
	217.857123637831	290.117119528845	427.74075070631	585.884807562458	649.578891399708	996.969640971489	1233.83117056856
Promedio	218.904377006888	304.884957629279	430.602093981041	577.087555670222	690.167508579441	1024.57611307439	1315.60833306359

Table 2

Tabla promedio	1	2	3	4	5	8	10
Promedio General	166.193327180263	233.883112169725	329.353798723449	439.937497161774	525.086096668308	759.522218752631	994.254059159226



Para este experimento su carga de trabajo tuvo las siguientes características:

- Carga previa de 10.000 registros
- Numero de operaciones 10.000
- 25% de operaciones eran lecturas
- 75% de operaciones eran de inserción
- El tamaño del registro es de 100 bytes

El experimento consiste en variar la cantidad de hilos (1,2,3,4,5,8 y 10) con respecto a la carga de trabajo, para observar como se ve afectada la latencia de la base Redis.

El experimento se ejecuto 5 veces para obtener un valor promedio general de Latencia. Antes de interpretar los resultados debemos intervenir con la información de Redis acerca de hilos. Redis usualmente suele usar un diseño de un hilo principal, esto significa que un solo proceso atiende todas las peticiones de un cliente usando la técnica multiplexacion, es decir Redis puede atender una petición en cada momento dado, por lo tanto todas las peticiones son atendidas secuencialmente.

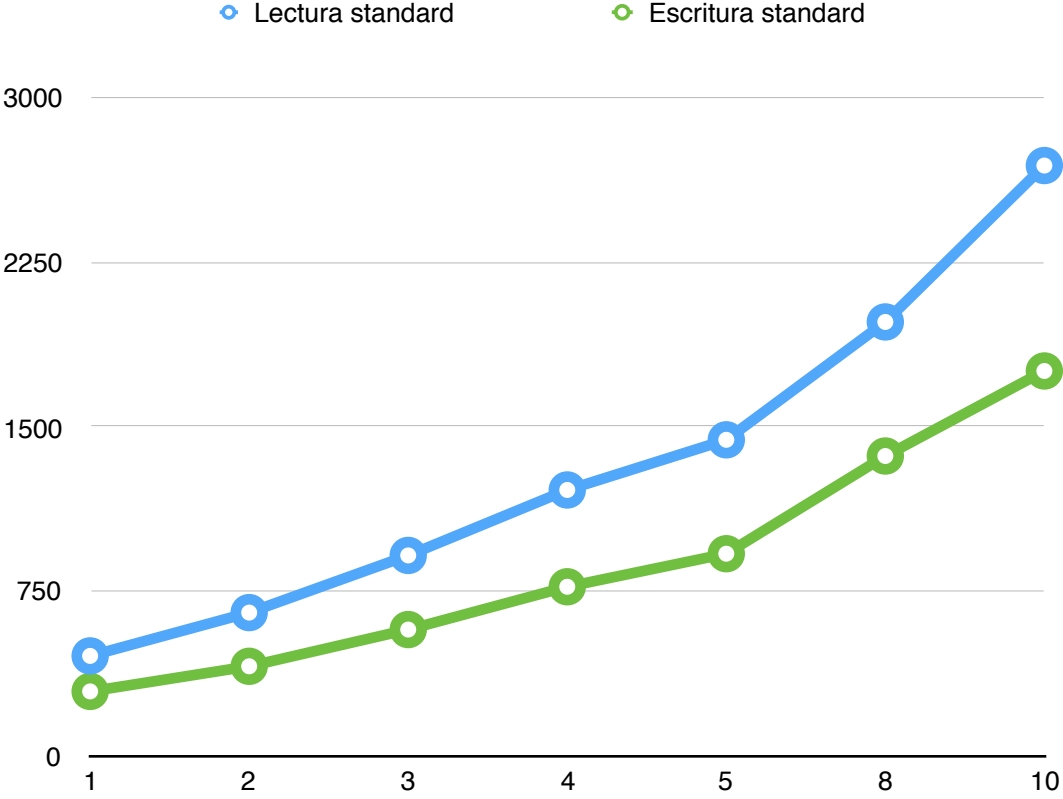
Entonces los resultados muestran que, mientras la cantidad de hilos aumenta su latencia también aumenta, podemos argumentar los siguientes puntos

1. Redis sin importar la operación que se encuentre ejecutando mientras aumente la cantidad de hilos, aumentara su latencia.

Table 3

	1	2	3	4	5	8	10
Lectura 25%	113.482277353639	162.881266710172	228.105503465857	302.787438653327	360.004684757175	494.468324430873	672.899785254862
Escritura 75%	218.904377006888	304.884957629279	430.602093981041	577.087555670222	690.167508579441	1024.57611307439	1315.60833306359

	1	2	3	4	5	8	10
Lectura standard	453.929109414555	651.525066840689	912.422013863429	1211.14975461331	1440.0187390287	1977.87329772349	2691.59914101945
Escritura standard	291.872502675851	406.513276839038	574.136125308054	769.450074226963	920.223344772587	1366.10148409919	1754.14444408479



Si estandarizamos los valores promedio de escritura y lectura y los ponemos una en contra al otro podremos darnos cuenta que el las operaciones de inserción nos dan una latencia muy menor a la que nos generan las operaciones de lectura. Esto se da por el propio hecho de que Redis es una base No SQL, las operaciones de inserción son mas rápidas que las de consulta de por si.

Si observamos sus gráficas independientes, podremos observar que sucede el mismo efecto, claro esta que la latencia de lectura es menor a la de escritura pero eso también es por el hecho de que la cantidad de operaciones de lectura corresponde al 25% de la cantidad de operaciones total que se dieron en este experimento.

