

Right Start for Snowflake Optimization

David Teplow, Managing Director of Snowflake
Services

Right Triangle Consulting

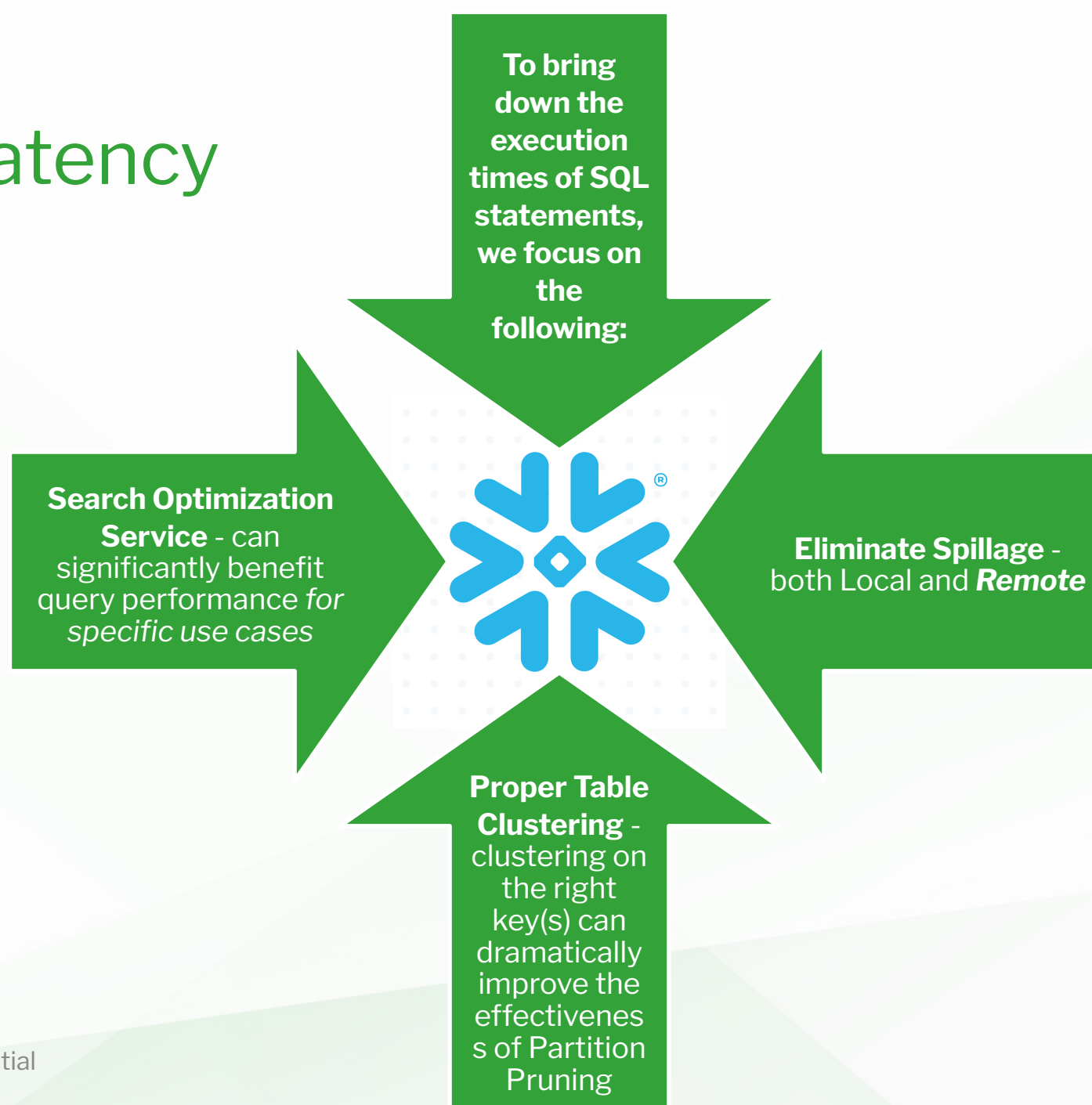


Our Right Start for Snowflake Optimization has 2 goals:

- **Reduce Latency** - bring down the execution times of SQL statements
- **Reduce Costs** - bring down the costs for computing and perhaps also for storage in Snowflake



Reduce Latency

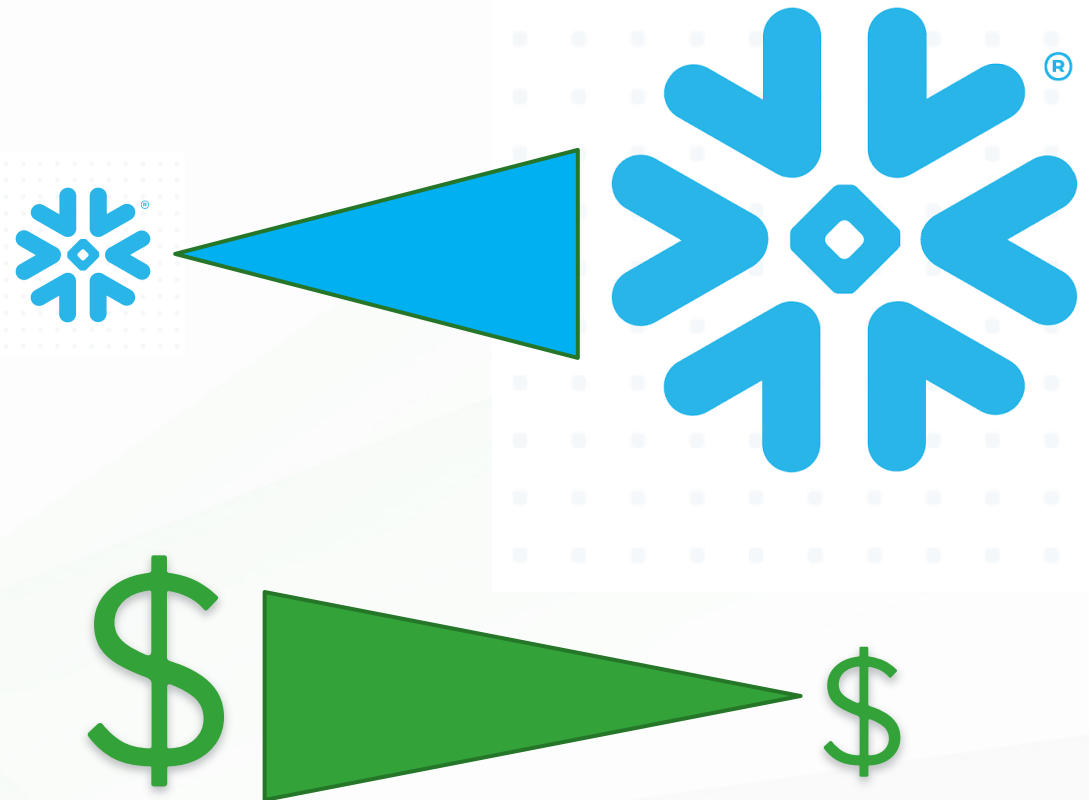


Reduce Costs



To bring down the costs for computing and perhaps also for storage in Snowflake, we focus on the following:

- Rightsize Virtual Warehouses - it's entirely possible in Snowflake to consume credits faster (i.e. use a larger Warehouse) and reduce overall cost
- Streamline the Pool of Virtual Warehouses - to maximize Warehouse (and Data Cache) reuse for common datasets and to minimize Warehouse “thrashing” (unnecessary suspending/resuming)



“To get better performance with Snowflake, you’ve got to be willing to spend less money.”

Deliverables



- Prioritized list of recommendations for improving performance and/or reducing costs
- Detailed documentation on the issues we find with step-by-step instructions on how to address them
- Implementing fixes can be done by you or by us in a separate engagement or in some combination of the two

Sample Deliverable - page 1



Executive Summary

Starting with the end goal of providing business value and the bottom-line we've made some modifications in our test environment and have reduced the Looker Dashboard response times while on average saving Snowflake dollar spend per query, plus Clustering and Search Optimization maintenance costs. The first two queries below are the longest running Looker queries from August and the remaining three are other representative August Looker queries. The variance in these metrics are based upon the queries and to what extent they can take advantage of Clustering and Search Optimization. Supporting documentation in the quest to find the sweet-spot between performance and cost are provided in the Technical Details section below.

"Speed costs money, how fast do you want to go?"

The rows in **red** are the original queries that ran on a Medium Warehouse, while the rows in **green** are the new times and spend that ran on a 3X-Large Warehouse.

	A	E	J	R	U	W	X
1	Total Execution & Statistics			Total	Percentage	\$\$\$	\$\$\$
2	Start Time	Time	Query ID	\$	of Time Saved	\$\$\$ Savings	Percentage Savings
3							
4	8/16/2022, 12:34:08 PM	1:14:57	01a65516-0504-f08e-0000-0b39c743ea16	\$11.99			
14	9/3/2022, 4:51:36 PM	0:01:12	01a6bb57-0505-498f-0000-0b39ca506fbe	\$3.07	98%	\$8.92	74%
18							
19	8/10/2022, 2:48:15 PM	0:47:08	01a63238-0504-dfd6-0000-0b39c64c265a	\$7.54			
24	9/6/2022, 8:24:31 AM	0:01:54	01a6ca3c-0505-5a4d-0000-0b39caaf2732	\$4.86	96%	\$2.68	36%
28							
29	8/11/2022, 2:24:54 PM	0:29:43	01a637c0-0504-ec1b-0000-0b39c6706a3a	\$4.75			
33	9/6/2022, 1:50:29 PM	0:01:55	01a6cb82-0505-5981-0000-0b39cab5db36	\$4.91	94%	-\$0.15	-3%
35							
36	8/24/2022, 9:54:45 AM	0:24:51	01a68176-0505-1a79-0000-0b39c8975932	\$3.98			
37	9/6/2022, 3:47:06 PM	0:01:48	01a6cbf7-0505-5abc-0000-0b39cab81e36	\$4.61	93%	-\$0.63	-16%
38							
39	8/23/2022, 6:37:12 AM	0:25:23	01a67b11-0505-199e-0000-0b39c85f4c82	\$4.06			
40	9/6/2022, 3:56:35 PM	0:00:20	01a6cc00-0505-5ab3-0000-0b39cab8d08a	\$0.85	99%	\$3.21	79%
41							

Zooming in...



	A	C	D	F	G	I	N	V	Y	AA	AB
1		TABLE 1		TABLE 2		Execution & Statistics		Total	Time	\$\$\$	\$\$\$
2	Start Time	SIZE	ROWS	SIZE	ROWS	Time	Query ID	\$	Savings	Savings	Savings
4	8/16/2022, 12:34:08 PM	9.4 TB	35.4 B	1.5 TB	1.8 B	1:14:57	01a65516-05	\$11.99			
14	9/3/2022, 4:51:36 PM	7.2 TB	35.4 B	655 GB	1.8 B	0:01:12	01a6bb57-05	\$3.07	98%	\$8.92	74%
18											
19	8/10/2022, 2:48:15 PM	9.4 TB	35.4 B	1.5 TB	1.8 B	0:47:08	01a63238-05	\$7.54			
24	9/6/2022, 8:24:31 AM	7.2 TB	35.4 B	655 GB	1.8 B	0:01:54	01a6ca3c-05	\$4.86	96%	\$2.68	36%
28											
29	8/11/2022, 2:24:54 PM	9.4 TB	35.4 B	1.5 TB	1.8 B	0:29:43	01a637c0-05	\$4.75			
33	9/6/2022, 1:50:29 PM	7.2 TB	35.4 B	655 GB	1.8 B	0:01:55	01a6cb82-05	\$4.91	94%	-\$0.15	-3%
35											
36	8/24/2022, 9:54:45 AM	9.4 TB	35.4 B	1.5 TB	1.8 B	0:24:51	01a68176-05	\$3.98			
37	9/6/2022, 3:47:06 PM	7.2 TB	35.4 B	655 GB	1.8 B	0:01:48	01a6cbf7-05	\$4.61	93%	-\$0.63	-16%
38											
39	8/23/2022, 6:37:12 AM	9.4 TB	35.4 B	1.5 TB	1.8 B	0:25:23	01a67b11-05	\$4.06			
40	9/6/2022, 3:56:35 PM	7.2 TB	35.4 B	655 GB	1.8 B	0:00:20	01a6cc00-05	\$0.85	99%	\$3.21	79%
41											

Sample Deliverable - page 2



Clustering tables incurs compute costs, speeds up some queries, and makes the tables smaller due to columnar database compression. Smaller tables scan faster.

Table	Currently	New Clustering
YT_VIDEO	1.5 TB	655 GB
YT_VIDEO_STATS_BY_MONTH	9.4 TB	7.2 TB

In our first meeting it was mentioned that some tables are created to speed up queries. Can any of the tuning work eliminate the need for these tables?

Aligning warehouse configuration with Snowflake Best Practices has the potential to reduce spend, reduce response times, and simplify administration. Snowflake recommends grouping similar queries per warehouse instead of per team, also grouping by data queried. Doing this reduces redundant similarly configured warehouses from running simultaneously while at the same time taking advantage of warehouse cache when possible. 32 warehouses are probably not needed. Having fewer, clustered warehouse enables them to handle the workload elastically, to suspend and resume less often, saving spend while waiting for an idle warehouse to suspend, and by not losing the warehouse cache when suspending, queries can run faster if they can serve up some of the data from cache instead of going back to remote disk I/O everytime.

Aligning Access Control with Snowflake Best Practices is pretty much required to properly secure a Snowflake Account. Using ACCOUNTADMIN on ONLY account level activity, SECURITYADMIN for users and roles, and SYSADMIN for just about everything else is an example. Using Access Roles and Functional Roles greatly simplifies Role Based Access Control (RBAC) and the administrative burden of attempting to grant appropriate access to individual objects. This is HIGHLY RECOMMENDED.

Misc recommendations

SNOWFLAKE.ACCOUNT_USAGE.QUERY_HISTORY table data retention is 12 months. We can configure a permanent QUERY_HISTORY table that does not truncate at 12 months.

Give us 2 weeks to show you how your  snowflake® environment can be **faster and cheaper**



Your answers to the following questions will get us started:

- 1) Describe 2 or 3 business processes that are most problematic due to the length of time it takes to process or return data:
- 2) How much are you spending on average per month on Snowflake; and how does that compare to what's budgeted?

Contact Us Today →