StarRocks 存算分离 3.1 性能调优手册

说明

本文针对 StarRocks 存算分离版本如何进行性能调优手册，供实践参考。本手册主要针对 StarRocks 3.1.x 正式版，且其中仅列举了存算分离版本中特有的参数，与存算一体公共的调优参数没有写入本手册，请知晓。

查询

参数名称	参数含义	默认值	修改命令
starlet_fs_stream_buffer_size_bytes	BE节点上控制每次读取后端对象存储（如S3）的 IO 大小，该值越大，访问对象存储的次数也就越少，对应的查询性能越高。该值一般针对冷数据查询效果比较明显。但调大该值也会相应地增大内存消耗。	131072（128KB）	修改be.conf相关配置项，不支持动态修改
lake_metadata_cache_limit	BE 节点上缓存 Tablet Meta 的内存大小，该值设置的越大，能缓存的 Tablet Meta 越多，也能一定程度上提升查询性能。	2147483648（2GB）	修改be.conf相关配置项，不支持动态修改
pipeline_connector_scan_thread_num_per_cpu	BE节点上每个 CPU 的 scan IO 线程数。 IO 线程数 = 该值 * CPU 核数。 如果 IO 时间很长，可以调大该值。	8	修改be.conf相关配置项，不支持动态修改
disable_column_pool	BE节点上禁用 column pool，关闭 column 复用。内存小的机器建议设置为 true。	false	修改be.conf相关配置项，不支持动态修改

导入

参数名称	参数含义	默认值	修改方式
create_tablet_worker_count	BE 节点上创建 Tablet 线程池的工作线程数，如果存在大量创建 Partition / Tablet 的场景，建议调大该值	3	修改be.conf相关配置项并重启 BE，暂不支持动态修改
enable_new_publish_mechanism	FE节点上新的publish version 机制，可以提升 publish version性能，建议设置为 true	false	在所有 FE 节点上执行： admin set frontend config ("enable_new_publish_mechanism" = "true");
flush_thread_num_per_store	BE节点上控制导入时 IO 刷对象存储的线程池大小，该值越大，写入吞吐越高	2	修改 be.conf 相关配置项，不支持动态修改
transaction_publish_version_worker_count	BE节点上执行 publish version 任务的线程池数量上限，该值越大，publish version 任务会越快地执行，相应提升写入吞吐	0（表示根据 CPU Core决定，但不会小于8）	修改 be.conf 相关配置项，不支持动态修改
number_tablet_writer_threads	BE节点上控制导入时写入 Mem Table 的线程池大小，BE 上所有 Tablet 的chunk 写入请求被放入队列并交由该线程池处理，提高该值对导入性能影响巨大。	16	修改 be.conf 相关配置项，不支持动态修改

Compaction

参数名称	参数含义	默认值	修改命令
lake_compaction_max_tasks	FE 上可同时发起的 Compaction 任务数量 默认值为-1，即FE会根据系统中 BE 数量自动计算。如	-1	admin set frontend config ("lake_compaction_max_tasks" = "xxx");

	果为0，则FE不再发起任何Compaction任务		
lake_compaction_score_selector_min_score	最小的Compaction score，如果 Partition 的 Compaction Score 低于该值，则不会对其发起 Compaction 任务	10.0	admin set frontend config ("lake_compaction_score_selector_min_score" = "xxx");
lake_compaction_history_size	控制show proc '/compactions' 显示的结果数量	12	admin set frontend config ("lake_compaction_history_size" = "xxx");
compact_threads	控制 BE 上同时执行 Compaction任务的线程数，也即 BE 上可同时为多少个Tablet进行 Compaction	4	修改be.conf相关配置项，不支持动态修改
compact_thread_pool_queue_size	BE 上控制 Compaction任务队列大小，控制可接收来自FE的最大Compaction 任务数	100	修改be.conf相关配置项，不支持动态修改

GC

参数名称	参数含义	默认值	修改命令
drop_tablet_worker_count	BE 节点上执行 GC 任务的线程池的工作线程数，如果导入比较频繁，建议调大该值以便能更快地清理无用数据。	3	修改be.conf相关配置项，不支持动态修改
lake_autovacuum_grace_period_minutes	FE 节点上控制历史数据文件保留时间（minute），一旦文件被 Compaction 后，且其创建时间距离当下超过该值，就会被清理。如果您的场景中有一些大的 ETL 任务，可能需要释放调大该值，否则可能容易造成查询	5	admin set frontend config ("lake_autovacuum_grace_period_minutes" = "xxx");

	时原始数据文件被清除的风险。		
lake_autovacuum_parallel_partitions	StarRocks 存算分离表中 FE 端控制可同时执行 vacuum 的 Partition 数量，默认值为8	8	admin set frontend config ("lake_autovacuum_parallel_partitions" = "xxx");
lake_autovacuum_partition_naptime_seconds	对于任一 Partition，两次 vacuum 之间的时间间隔（秒），默认为180 秒，如果您想加快数据清理进度，可以适当降低该值	180	admin set frontend config ("lake_autovacuum_partition_naptime_seconds" = "xxx");
lake_autovacuum_stale_partition_threshold	如果 Partition 最后更新时间距离当前时间超过该阈值，那么就不再为该 Partition 执行清理		admin set frontend config ("lake_autovacuum_stale_partition_threshold" = "xxx");

Cache

参数名称	参数含义	默认值	修改命令
starlet_cache_thread_num	BE 节点上开启 File Cache 时控制后台异步拉取缓存数据的线程数，该值越大，拉取的效率越高。但最终的效率也取决于BE节点的网络和磁盘吞吐，而且一味地调大该参数可能会导致其他任务资源争抢。	64	修改be.conf相关配置项，不支持动态修改
starlet_cache_evict_interval	BE 节点开启 File Cache 时检查 Cache Disk 空间是否足够的周期（second）	60	修改be.conf相关配置项，不支持动态修改
starlet_cache_evict_low_water	BE 节点开启 Cache 淘汰的低水位（百分比）。一旦 Disk 空闲空间低于该值，开始进行缓存淘汰	0.1	修改be.conf相关配置项，不支持动态修改
starlet_cache_evict_high_water	BE 节点停止 Cache 淘汰的高水位（百分比）。一旦	0.2	修改be.conf相关配置项，不支持动态修改

	Disk 空闲空间高于该值，停止进行缓存淘汰		
starlet_use_star_cache	BE 节点是否开启 Block Cache 能力，该能力在 3.1 版本引入，作为对当前 File Cache 的能力补充，每次 Cache Miss 时，只会从后端对象存储获取部分相关内容，避免 File Cache 在该情况下拉取整个文件带来的资源浪费和效率较低。如果用户冷查场景较多，可以考虑开启该功能	false	修改be.conf相关配置项，不支持动态修改