云数据仓库 for Apache Doris

运维指南

产品文档

腾讯云

文档目录

运维指南
创建与销毁

最近更新时间：2022-06-07 11:37:20

新建集群

在 云数据仓库 for Apache Doris 介绍页，单击**立即购买**。或登录 云数据仓库 for Apache Doris 控制台，单击**新建集群**。进入购买页，根据页面提示，进行配置并购买。
如下图所示，购买页包括购买须知、基础配置、集群配置等，填写和选择后最下面可以看到计费情况，确认配置费用后，最后点击立即购买就可以开始新建集群了。

购买须知

使用说明　不清楚如何选型和购买？可以参考 计费概述 或 联系我们 进行咨询

基础配置

计费模式　包年包月　按量计费

地域

华东地区	华北地区	东南亚地区	西南地域	华南地区	美国西部地区
上海	北京	新加坡	重庆	广州	硅谷

处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择；建议选择最靠近您客户的地域，可降低访问时延。

可用区　广州四区　广州六区　广州七区

网络　vpc-dwbfd4df | fengluan-test　subnet-rae818xg | fengluan-subnet1

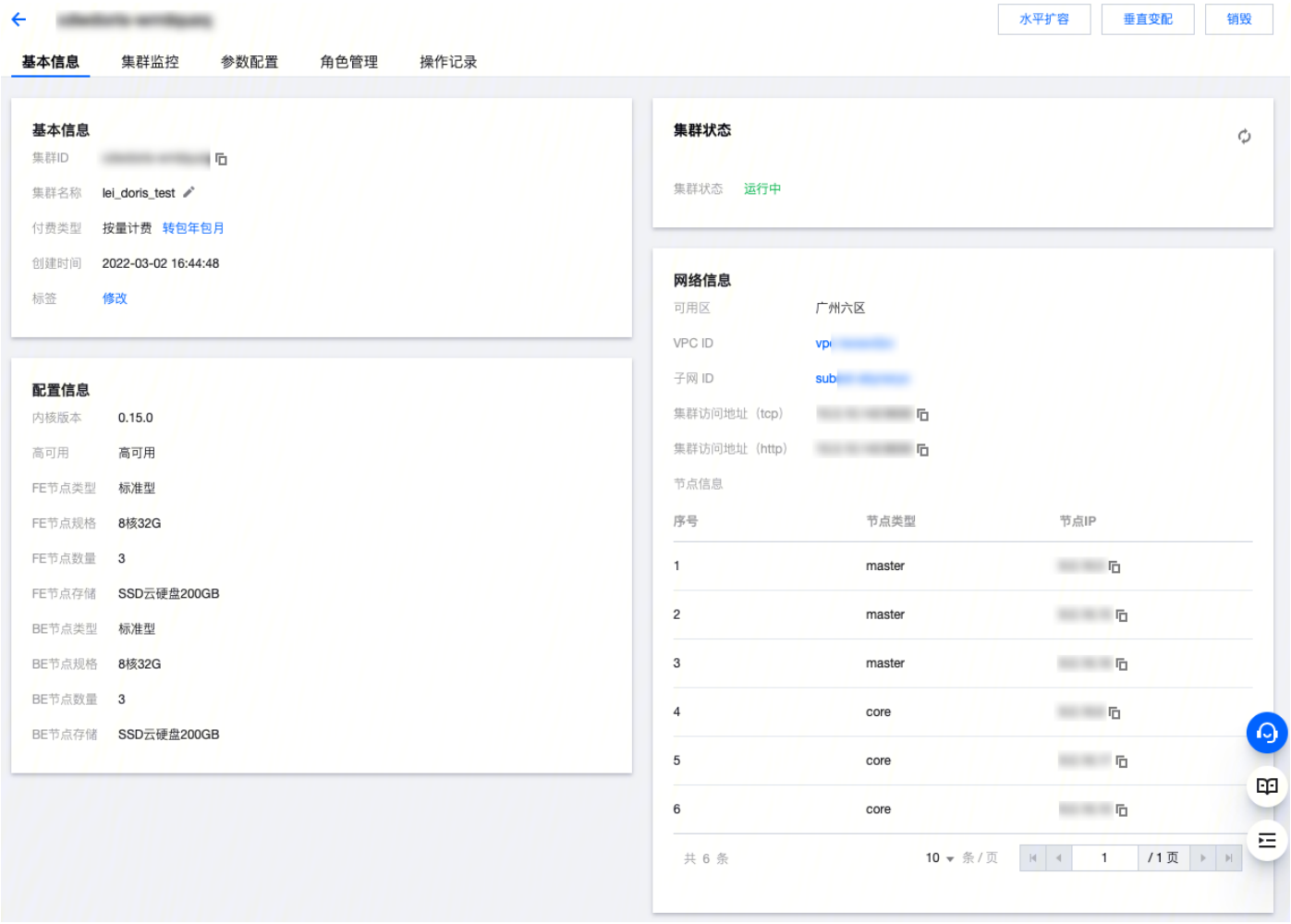
共 253 个子网IP，剩 202 个可用。
如现有的网络不合适，您可以去控制台新建私有网络或新建子网。

查看集群基本信息

集群创建后，您即可进入 云数据仓库 for Apache Doris 控制台，在集群列表，单击**集群 ID/名称**，可以进入到集群详情页。

- 在详情页，可以查看集群基本信息、集群监控、可进行集群参数配置、也可以查看集群操作记录。

- 通过单击集群名称后面的编辑按钮，可对集群名字进行修改。



监控、参数配置、角色管理和操作记录

在集群详情页，选择**集群监控**页签，可以查看集群的各种监控指标。
通过参数配置，可以配置FE、BE和Broker的各种参数。
通过橘色管理可以进行进程的启停管控。
操作记录中记录了集群操作情况。

集群销毁

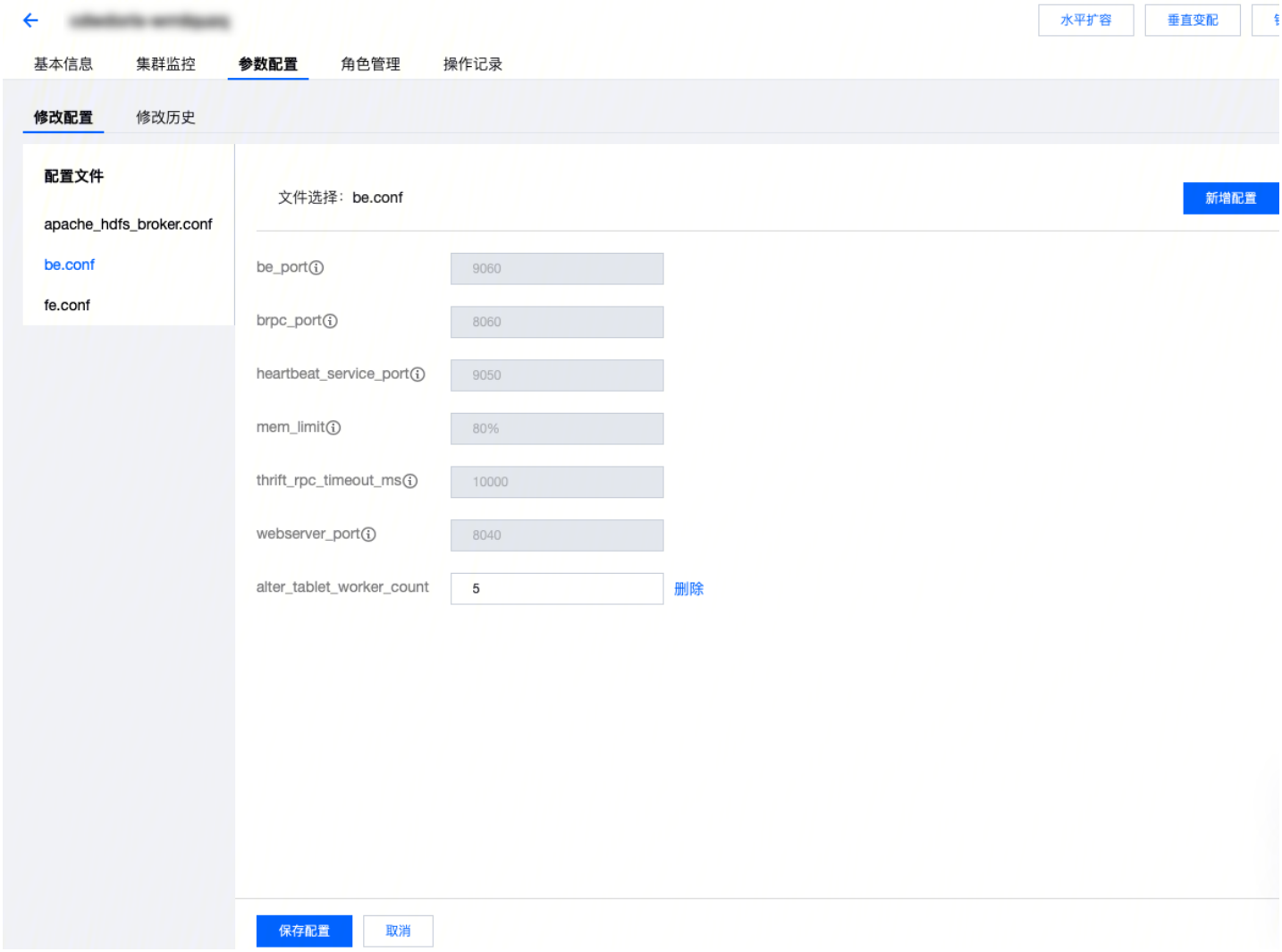
选择集群列表中**操作 > 销毁**，或者集群信息页面右上角的销毁按钮，根据提示，完成集群销毁。

- 包年包月的集群：集群销毁后，会以“已隔离”的状态保留7天，期间集群不可用，在此期间可进行续费，续费后集群可正常使用。

- 按量计费的集群：集群销毁后，则会在24小时内，释放资源，清除数据。

参数配置

在集群详情页面，选择**参数配置**页签，进入到参数配置页面。修改参数后，单击【 应用到集群 】，修改后的参数即可生效。



查看操作记录

在集群详情页面，选择**操作记录**页签，可以查看一段时间内所有操作。

水平扩容与垂直变配

最近更新时间：2022-06-07 11:37:26

功能介绍

当集群当前的规模及配置不满足使用需求时，Doris 集群管理提供了水平扩容、垂直变配功能，帮助您完成集群横向集群节点扩展，纵向节点规格等调整。

> 说明：
>
> - 操作集群需处于稳定的运行中状态。
> - 操作主账号无欠费、待支付订单情况。

水平扩容

> 说明：
>
> - 水平扩容过程中，系统读写仍可进行，但是可能出现一些抖动，执行操作大约需要5 – 15分钟，请选择在非业务高峰期进行。
> - 在数据存储量及查询量均相对增长时，优先选择水平扩容。

1. 登录 云数据仓库 for Apache Doris 控制台，在需要操作的集群中选择**操作 > 水平扩容**。

2. 在集群水平变配页，选择将节点个数增加至需要量。

3. 单击**确定**，确认页完成订单支付，集群状态“变更中”将持续5 – 15分钟。
4. 扩容完成后，进入集群详情页，可以查看到新扩容的节点。

垂直变配

Doris 集群的垂直变配包括：计算节点规格升、降配，存储规格扩展；高可用集群的 FE 节点规格升、降配，存储规格扩展。

> **说明：**
>
> - 垂直变配系统不可读、不可写。
> - 您可按需升配节点规格和存储规格，变配操作结果对集群节点均生效。
> - 支持节点规格升降配，存储规格升配。
> - 变更费用说明：包年包月计费形式变更费用指单月差价费用，按量计费形式变更费用指变更后小时单价。

1. 登录 云数据仓库 for Apache Doris 控制台，在需要操作的集群中选择**操作 > 垂直变配**。

2. 在集群水平变配页，选择节点变配的计算规格，按需升配存储规格。

> **注意：**
>
> 支持节点的计算规格和存储规格支持同时或单项升配，当操作计算规格降配时不可操作存储规格。

3. 单击**确定**，确认页完成订单支付，集群状态“变更中”将持续5～15分钟。

Doris 错误代码表

最近更新时间：2022-03-14 15:08:14

错误码	错误信息
1005	创建表格失败，在返回错误信息中给出具体原因
1007	数据库已经存在，不能创建同名的数据库
1008	数据库不存在，无法删除
1044	数据库对用户未授权，不能访问
1045	用户名及密码不匹配，不能访问系统
1046	没有指定要查询的目标数据库
1047	用户输入了无效的操作指令
1049	用户指定了无效的数据库
1050	数据表已经存在
1051	无效的数据表
1052	指定的列名有歧义，不能唯一确定对应列
1053	为 Semi-Join/Anti-Join 查询指定了非法的数据列
1054	指定的列在表中不存在
1058	查询语句中选择的列数目与查询结果的列数目不一致
1060	列名重复
1064	没有存活的 Backend 节点
1066	查询语句中出现了重复的表别名
1094	线程 ID 无效
1095	非线程的拥有者不能终止线程的运行
1096	查询语句没有指定要查询或操作的数据表
1102	数据库名不正确
1104	数据表名不正确

错误码	错误信息
1105	其它错误
1110	子查询中指定了重复的列
1111	在 Where 从句中非法使用聚合函数
1113	新建表的列集合不能为空
1115	使用了不支持的字符集
1130	客户端使用了未被授权的 IP 地址来访问系统
1132	无权限修改用户密码
1141	撤销用户权限时指定了用户不具备的权限
1142	用户执行了未被授权的操作
1166	列名不正确
1193	使用了无效的系统变量名
1203	用户使用的活跃连接数超过了限制
1211	不允许创建新用户
1227	拒绝访问，用户执行了无权限的操作
1228	会话变量不能通过 SET GLOBAL 指令来修改
1229	全局变量应通过 SET GLOBAL 指令来修改
1230	相关的系统变量没有缺省值
1231	给某系统变量设置了无效值
1232	给某系统变量设置了错误数据类型的值
1248	没有给内联视图设置别名
1251	客户端不支持服务器请求的身份验证协议；请升级 MySQL 客户端
1286	配置的存储引擎不正确
1298	配置的时区不正确
1347	对象与期望的类型不匹配

错误码	错误信息
1353	SELECT 和视图的字段列表具有不同的列数
1364	字段不允许 NULL 值，但是没有设置缺省值
1372	密码长度不够
1396	用户执行的操作运行失败
1471	指定表不允许插入数据
1507	删除不存在的分区，且没有指定如果存在才删除的条件
1508	无法删除所有分区，请改用 DROP TABLE
1517	出现了重复的分区名字
1567	分区的名字不正确
1621	指定的系统变量是只读的
1735	表中不存在指定的分区名
1748	不能将数据插入具有空分区的表中。使用“ SHOW PARTITIONS FROM tbl”来查看此表的当前分区
1749	表分区不存在
5000	指定的表不是 OLAP 表
5001	指定的 PROC 路径无效
5002	必须在列置换中明确指定列名
5003	Key 列应排在 Value 列之前
5004	表至少应包含1个 Key 列
5005	集群 ID 无效
5006	无效的查询规划
5007	冲突的查询规划
5008	数据插入提示：仅适用于有分区的数据表
5009	PARTITION 子句对于 INSERT 到未分区表中无效
5010	列数不等于 SELECT 语句的选择列表数

错误码	错误信息
5011	无法解析表引用
5012	指定的值不是一个有效数字
5013	不支持的时间单位
5014	表状态不正常
5015	分区状态不正常
5016	分区上存在数据导入任务
5017	指定列不是 Key 列
5018	值的格式无效
5019	数据副本与版本不匹配
5021	BE 节点已离线
5022	非分区表中的分区数不是1
5023	alter 语句中无任何操作
5024	任务执行超时
5025	数据插入操作失败
5026	通过 SELECT 语句创建表时使用了不支持的数据类型
5027	没有设置指定的参数
5028	没有找到指定的集群
5030	某用户没有访问集群的权限
5031	没有指定参数或参数无效
5032	没有指定集群实例数目
5034	集群名已经存在
5035	集群已经存在
5036	集群中 BE 节点不足
5037	删除集群之前，必须删除集群中的所有数据库

错误码	错误信息
5037	集群中不存在这个 ID 的 BE 节点
5038	没有指定集群名字
5040	未知的集群
5041	没有集群名字
5042	没有权限
5043	实例数目应大于0
5046	源集群不存在
5047	目标集群不存在
5048	源数据库不存在
5049	目标数据库不存在
5050	没有选择集群，请输入集群
5051	应先将源数据库连接到目标数据库
5052	集群内部错误：BE 节点错误信息
5053	没有从源数据库到目标数据库的迁移任务
5054	指定数据库已经连接到目标数据库，或正在迁移数据
5055	数据连接或者数据迁移不能在同一集群内执行
5056	不能删除数据库：它被关联至其它数据库或正在迁移数据
5056	不能重命名数据库：它被关联至其它数据库或正在迁移数据
5056	集群中 BE 节点不足
5056	集群内已存在指定数目的 BE 节点
5059	集群中存在处于下线状态的 BE 节点
5062	不正确的群集名称（名称'default_cluster'是保留名称）
5063	类型名不正确
5064	通用错误提示

错误码	错误信息
5063	Colocate 功能已被管理员禁用
5063	Colocate 数据表不存在
5063	Colocate 表必须是 OLAP 表
5063	Colocate 表应该具有同样的副本数目
5063	Colocate 表应该具有同样的分桶数目
5063	Colocate 表的分区列数目必须一致
5063	Colocate 表的分区列的数据类型必须一致
5064	指定表不是 Colocate 表
5065	指定的操作是无效的
5065	指定的时间单位是非法的，正确的单位包括：HOUR DAY WEEK / MONTH
5066	动态分区起始值应该小于0
5066	动态分区起始值不是有效的数字
5066	动态分区结束值应该大于0
5066	动态分区结束值不是有效的数字
5066	动态分区结束值为空
5067	动态分区分桶数应该大于0
5067	动态分区分桶值不是有效的数字
5066	动态分区分桶值为空
5068	是否允许动态分区的值不是有效的布尔值：true 或者 false
5069	指定的动态分区名前缀是非法的
5070	指定的操作被禁止了
5071	动态分区副本数应该大于0
5072	动态分区副本值不是有效的数字
5073	原始创建表 stmt 为空

错误码	错误信息
5074	创建历史动态分区参数：create_history_partition 无效，期望的是：true 或者 false
5076	指定的保留历史分区时间段为空
5077	指定的保留历史分区时间段无效
5078	指定的保留历史分区时间段必须是成对的时间
5079	指定的保留历史分区时间段对应位置的第一个时间比第二个时间大（起始时间大于结束时间）

BE 端 OLAP 函数的返回值说明

最近更新时间：2022-03-14 15:08:30

BE 端 OLAP 函数的返回值说明

返回值名称	返回值	返回值说明
OLAP_SUCCESS	0	成功
OLAP_ERR_OTHER_ERROR	-1	其他错误
OLAP_REQUEST_FAILED	-2	请求失败

系统错误代码

例如文件系统内存和其他系统调用失败。

返回值名称	返回值	返回值说明
OLAP_ERR_OS_ERROR	-100	操作系统错误
OLAP_ERR_DIR_NOT_EXIST	-101	目录不存在错误
OLAP_ERR_FILE_NOT_EXIST	-102	文件不存在错误
OLAP_ERR_CREATE_FILE_ERROR	-103	创建文件错误
OLAP_ERR_MALLOC_ERROR	-104	内存分配错误
OLAP_ERR_STL_ERROR	-105	标准模板库错误
OLAP_ERR_IO_ERROR	-106	IO 错误
OLAP_ERR_MUTEX_ERROR	-107	互斥锁错误
OLAP_ERR_PTHREAD_ERROR	-108	POSIX thread 错误
OLAP_ERR_NETWORK_ERROR	-109	网络异常错误
OLAP_ERR_UB_FUNC_ERROR	-110	
OLAP_ERR_COMPRESS_ERROR	-111	数据压缩错误
OLAP_ERR_DECOMPRESS_ERROR	-112	数据解压缩错误
OLAP_ERR_UNKNOWN_COMPRESSION_TYPE	-113	未知的数据压缩类型

返回值名称	返回值	返回值说明
OLAP_ERR_MMAP_ERROR	-114	内存映射文件错误
OLAP_ERR_RWLOCK_ERROR	-115	读写锁错误
OLAP_ERR_READ_UNENOUGH	-116	读取内存不够异常
OLAP_ERR_CANNOT_CREATE_DIR	-117	不能创建目录异常
OLAP_ERR_UB_NETWORK_ERROR	-118	网络异常
OLAP_ERR_FILE_FORMAT_ERROR	-119	文件格式异常
OLAP_ERR_EVAL_CONJUNCTS_ERROR	-120	
OLAP_ERR_COPY_FILE_ERROR	-121	拷贝文件错误
OLAP_ERR_FILE_ALREADY_EXIST	-122	文件已经存在错误

通用错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_NOT_INITED	-200	不能初始化异常
OLAP_ERR_FUNC_NOT_IMPLEMENTED	-201	函数不能执行异常
OLAP_ERR_CALL_SEQUENCE_ERROR	-202	调用 SEQUENCE 异常
OLAP_ERR_INPUT_PARAMETER_ERROR	-203	输入参数错误
OLAP_ERR_BUFFER_OVERFLOW	-204	内存缓冲区溢出错误
OLAP_ERR_CONFIG_ERROR	-205	配置错误
OLAP_ERR_INIT_FAILED	-206	初始化失败
OLAP_ERR_INVALID_SCHEMA	-207	无效的 Schema
OLAP_ERR_CHECKSUM_ERROR	-208	检验值错误
OLAP_ERR_SIGNATURE_ERROR	-209	签名错误
OLAP_ERR_CATCH_EXCEPTION	-210	捕捉到异常
OLAP_ERR_PARSE_PROTOBUF_ERROR	-211	解析 Protobuf 出错
OLAP_ERR_SERIALIZE_PROTOBUF_ERROR	-212	Protobuf 序列化错误

返回值名称	返回值	返回值说明
OLAP_ERR_WRITE_PROTOBUF_ERROR	-213	Protobuf 写错误
OLAP_ERR_VERSION_NOT_EXIST	-214	tablet 版本不存在错误
OLAP_ERR_TABLE_NOT_FOUND	-215	未找到 tablet 错误
OLAP_ERR_TRY_LOCK_FAILED	-216	尝试锁失败
OLAP_ERR_OUT_OF_BOUND	-218	内存越界
OLAP_ERR_UNDERFLOW	-219	underflow 错误
OLAP_ERR_FILE_DATA_ERROR	-220	文件数据错误
OLAP_ERR_TEST_FILE_ERROR	-221	测试文件错误
OLAP_ERR_INVALID_ROOT_PATH	-222	无效的根目录
OLAP_ERR_NO_AVAILABLE_ROOT_PATH	-223	没有有效的根目录
OLAP_ERR_CHECK_LINES_ERROR	-224	检查行数错误
OLAP_ERR_INVALID_CLUSTER_INFO	-225	无效的 Cluster 信息
OLAP_ERR_TRANSACTION_NOT_EXIST	-226	事务不存在
OLAP_ERR_DISK_FAILURE	-227	磁盘错误
OLAP_ERR_TRANSACTION_ALREADY_COMMITTED	-228	交易已提交
OLAP_ERR_TRANSACTION_ALREADY_VISIBLE	-229	事务可见
OLAP_ERR_VERSION_ALREADY_MERGED	-230	版本已合并
OLAP_ERR_LZO_DISABLED	-231	LZO 已禁用
OLAP_ERR_DISK_REACH_CAPACITY_LIMIT	-232	磁盘到达容量限制
OLAP_ERR_TOO_MANY_TRANSACTIONS	-233	太多事务积压未完成
OLAP_ERR_INVALID_SNAPSHOT_VERSION	-234	无效的快照版本
OLAP_ERR_TOO_MANY_VERSION	-235	tablet的数据版本超过了最大限制（默认500）
OLAP_ERR_NOT_INITIALIZED	-236	不能初始化
OLAP_ERR_ALREADY_CANCELLED	-237	已经被取消

命令执行异常代码

返回值名称	返回值	返回值说明
OLAP_ERR_CE_CMD_PARAMS_ERROR	-300	命令参数错误
OLAP_ERR_CE_BUFFER_TOO_SMALL	-301	缓冲区太多小文件
OLAP_ERR_CE_CMD_NOT_VALID	-302	无效的命令
OLAP_ERR_CE_LOAD_TABLE_ERROR	-303	加载数据表错误
OLAP_ERR_CE_NOT_FINISHED	-304	命令没有执行成功
OLAP_ERR_CE_TABLET_ID_EXIST	-305	tablet Id 不存在错误
OLAP_ERR_CE_TRY_CE_LOCK_ERROR	-306	尝试获取执行命令锁错误

Tablet 错误异常代码

返回值名称	返回值	返回值说明
OLAP_ERR_TABLE_VERSION_DUPLICATE_ERROR	-400	tablet 副本版本错误
OLAP_ERR_TABLE_VERSION_INDEX_MISMATCH_ERROR	-401	teblet 版本索引不匹配异常
OLAP_ERR_TABLE_INDEX_VALIDATE_ERROR	-402	这里不检查 tablet 的初始版本，因为如果在一个 tablet 进行 schema-change 时重新启动 BE，我们可能会遇到空 tablet 异常
OLAP_ERR_TABLE_INDEX_FIND_ERROR	-403	无法获得第一个 Block 块位置 或者找到最后一行 Block 块失败会引发此异常
OLAP_ERR_TABLE_CREATE_FROM_HEADER_ERROR	-404	无法加载 Tablet 的时候会触发此异常
OLAP_ERR_TABLE_CREATE_META_ERROR	-405	无法创建 Tablet（更改schema），Base tablet 不存在，会触发此异常

返回值名称	返回值	返回值说明
OLAP_ERR_TABLE_ALREADY_DELETED_ERROR	-406	tablet 已经被删除

存储引擎错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_ENGINE_INSERT_EXISTS_TABLE	-500	添加相同的 tablet 两次，添加 tablet 到相同数据目录两次，新 tablet 为空，旧 tablet 存在。会触发此异常
OLAP_ERR_ENGINE_DROP_NOEXISTS_TABLE	-501	删除不存在的表
OLAP_ERR_ENGINE_LOAD_INDEX_TABLE_ERROR	-502	加载 tablet_meta 失败，cumulative rowset无效的 segment group meta，会引发此异常
OLAP_ERR_TABLE_INSERT_DUPLICATION_ERROR	-503	表插入重复
OLAP_ERR_DELETE_VERSION_ERROR	-504	删除版本错误
OLAP_ERR_GC_SCAN_PATH_ERROR	-505	GC 扫描路径错误
OLAP_ERR_ENGINE_INSERT_OLD_TABLET	-506	当 BE 正在重新启动并且较旧的tablet已添加到垃圾收集队列但尚未删除时,在这种情况下,由于 data_dirs 是并行加载的，稍后加载的tablet可能比以前加载的tablet旧，这不应被确认为失败,所以此时返回改代码

Fetch Handler 错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_FETCH_OTHER_ERROR	-600	FetchHandler 其他错误
OLAP_ERR_FETCH_TABLE_NOT_EXIST	-601	FetchHandler 表不存在
OLAP_ERR_FETCH_VERSION_ERROR	-602	FetchHandler 版本错误
OLAP_ERR_FETCH_SCHEMA_ERROR	-603	FetchHandler Schema 错误

返回值名称	返回值	返回值说明
OLAP_ERR_FETCH_COMPRESSION_ERROR	-604	FetchHandler 压缩错误
OLAP_ERR_FETCH_CONTEXT_NOT_EXIST	-605	FetchHandler 上下文不存在
OLAP_ERR_FETCH_GET_READER_PARAMS_ERR	-607	FetchHandler GET 读参数错误
OLAP_ERR_FETCH_SAVE_SESSION_ERR	-608	FetchHandler 保存会话错误
OLAP_ERR_FETCH_MEMORY_EXCEEDED	-609	FetchHandler 内存超出异常

读异常错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_READER_IS_UNINITIALIZED	-700	读不能初始化
OLAP_ERR_READER_GET_ITERATOR_ERROR	-701	获取读迭代器错误
OLAP_ERR_CAPTURE_ROWSET_READER_ERROR	-702	当前Rowset读错误
OLAP_ERR_READER_READING_ERROR	-703	初始化列数据失败，cumulative rowset 的列数据无效，会返回该异常代码
OLAP_ERR_READER_INITIALIZE_ERROR	-704	读初始化失败

BaseCompaction 异常代码信息

返回值名称	返回值	返回值说明
OLAP_ERR_BE_VERSION_NOT_MATCH	-800	BE Compaction 版本不匹配错误
OLAP_ERR_BE_REPLACE_VERSIONS_ERROR	-801	BE Compaction 替换版本错误
OLAP_ERR_BE_MERGE_ERROR	-802	BE Compaction 合并错误
OLAP_ERR_BE_COMPUTE_VERSION_HASH_ERROR	-803	BE Compaction 计算版本哈希错误
OLAP_ERR_CAPTURE_ROWSET_ERROR	-804	找不到 Rowset 对应的版本

返回值名称	返回值	返回值说明
OLAP_ERR_BE_SAVE_HEADER_ERROR	-805	BE Compaction 保存 Header 错误
OLAP_ERR_BE_INIT_OLAP_DATA	-806	BE Compaction 初始化 OLAP 数据错误
OLAP_ERR_BE_TRY_OBTAIN_VERSION_LOCKS	-807	BE Compaction 尝试获得版本锁错误
OLAP_ERR_BE_NO_SUITABLE_VERSION	-808	BE Compaction 没有合适的版本
OLAP_ERR_BE_TRY_BE_LOCK_ERROR	-809	其他 base compaction 正在运行，尝试获取锁失败
OLAP_ERR_BE_INVALID_NEED_MERGED_VERSIONS	-810	无效的 Merge 版本
OLAP_ERR_BE_ERROR_DELETE_ACTION	-811	BE 执行删除操作错误
OLAP_ERR_BE_SEGMENTS_OVERLAPPING	-812	cumulative point 有重叠的 Rowset 异常
OLAP_ERR_BE_CLONE_OCCURRED	-813	将压缩任务提交到线程池后可能会发生克隆任务，并且选择用于压缩的行集可能会发生变化。 在这种情况下，不应执行当前的压缩任务。 返回该代码

PUSH 异常代码

返回值名称	返回值	返回值说明
OLAP_ERR_PUSH_INIT_ERROR	-900	无法初始化读取器，无法创建表描述符，无法初始化内存跟踪器，不支持的文件格式类型，无法打开扫描仪，无法获取元组描述符，为元组分配内存失败，都会返回该代码
OLAP_ERR_PUSH_DELTA_FILE_EOF	-901	
OLAP_ERR_PUSH_VERSION_INCORRECT	-902	PUSH 版本不正确
OLAP_ERR_PUSH_SCHEMA_MISMATCH	-903	PUSH Schema 不匹配

返回值名称	返回值	返回值说明
OLAP_ERR_PUSH_CHECKSUM_ERROR	-904	PUSH 校验值错误
OLAP_ERR_PUSH_ACQUIRE_DATASOURCE_ERROR	-905	PUSH 获取数据源错误
OLAP_ERR_PUSH_CREAT_CUMULATIVE_ERROR	-906	PUSH 创建 CUMULATIVE 错误代码
OLAP_ERR_PUSH_BUILD_DELTA_ERROR	-907	推送的增量文件有错误的校验码
OLAP_ERR_PUSH_VERSION_ALREADY_EXIST	-908	PUSH 的版本已经存在
OLAP_ERR_PUSH_TABLE_NOT_EXIST	-909	PUSH 的表不存在
OLAP_ERR_PUSH_INPUT_DATA_ERROR	-910	PUSH 的数据无效，可能是长度，数据类型等问题
OLAP_ERR_PUSH_TRANSACTION_ALREADY_EXIST	-911	将事务提交给引擎时，发现Rowset存在，但Rowset ID 不一样
OLAP_ERR_PUSH_BATCH_PROCESS_REMOVED	-912	删除了推送批处理过程
OLAP_ERR_PUSH_COMMIT_ROWSET	-913	PUSH Commit Rowset
OLAP_ERR_PUSH_ROWSET_NOT_FOUND	-914	PUSH Rowset 没有发现

SegmentGroup 异常代码

返回值名称	返回值	返回值说明
OLAP_ERR_INDEX_LOAD_ERROR	-1000	加载索引错误
OLAP_ERR_INDEX_EOF	-1001	
OLAP_ERR_INDEX_CHECKSUM_ERROR	-1002	校验码验证错误，加载索引对应的 Segment 错误。
OLAP_ERR_INDEX_DELTA_PRUNING	-1003	索引增量修剪

OLAPData 异常代码信息

返回值名称	返回值	返回值说明
OLAP_ERR_DATA_ROW_BLOCK_ERROR	-1100	数据行 Block 块错误

返回值名称	返回值	返回值说明
OLAP_ERR_DATA_FILE_TYPE_ERROR	-1101	数据文件类型错误
OLAP_ERR_DATA_EOF	-1102	

OLAP 数据写错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_WRITER_INDEX_WRITE_ERROR	-1200	索引写错误
OLAP_ERR_WRITER_DATA_WRITE_ERROR	-1201	数据写错误
OLAP_ERR_WRITER_ROW_BLOCK_ERROR	-1202	Row Block 块写错误
OLAP_ERR_WRITER_SEGMENT_NOT_FINALIZED	-1203	在添加新 Segment 之前，上一 Segment 未完成

RowBlock 错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_ROWBLOCK_DECOMPRESS_ERROR	-1300	Rowblock 解压缩错误
OLAP_ERR_ROWBLOCK_FIND_ROW_EXCEPTION	-1301	获取 Block Entry 失败
OLAP_ERR_ROWBLOCK_READ_INFO_ERROR	-1302	读取 Rowblock 信息错误

Tablet 元数据错误

返回值名称	返回值	返回值说明
OLAP_ERR_HEADER_ADD_VERSION	-1400	tablet 元数据增加版本
OLAP_ERR_HEADER_DELETE_VERSION	-1401	tablet 元数据删除版本
OLAP_ERR_HEADER_ADD_PENDING_DELTA	-1402	tablet 元数据添加待处理增量
OLAP_ERR_HEADER_ADD_INCREMENTAL_VERSION	-1403	tablet 元数据添加自增版本
OLAP_ERR_HEADER_INVALID_FLAG	-1404	tablet 元数据无效的标记
OLAP_ERR_HEADER_PUT	-1405	tablet 元数据 PUT 操作

返回值名称	返回值	返回值说明
OLAP_ERR_HEADER_DELETE	-1406	tablet 元数据 DELETE 操作
OLAP_ERR_HEADER_GET	-1407	tablet 元数据 GET 操作
OLAP_ERR_HEADER_LOAD_INVALID_KEY	-1408	tablet 元数据加载无效 Key
OLAP_ERR_HEADER_FLAG_PUT	-1409	
OLAP_ERR_HEADER_LOAD_JSON_HEADER	-1410	tablet 元数据加载 JSON Header
OLAP_ERR_HEADER_INIT_FAILED	-1411	tablet 元数据 Header 初始化失败
OLAP_ERR_HEADER_PB_PARSE_FAILED	-1412	tablet 元数据 Protobuf 解析失败
OLAP_ERR_HEADER_HAS_PENDING_DATA	-1413	tablet 元数据有待处理的数据

TabletSchema 异常代码信息

返回值名称	返回值	返回值说明
OLAP_ERR_SCHEMA_SCHEMA_INVALID	-1500	Tablet Schema 无效
OLAP_ERR_SCHEMA_SCHEMA_FIELD_INVALID	-1501	Tablet Schema 字段无效
SchemaHandler异常代码信息		
OLAP_ERR_ALTER_MULTI_TABLE_ERR	-1600	ALTER 多表错误
OLAP_ERR_ALTER_DELTA_DOES_NOT_EXISTS	-1601	获取所有数据源失败，Tablet 无版本
OLAP_ERR_ALTER_STATUS_ERR	-1602	检查行号失败，内部排序失败，行块排序失败，这些都会返回该代码
OLAP_ERR_PREVIOUS_SCHEMA_CHANGE_NOT_FINISHED	-1603	先前的 Schema 更改未完成

返回值名称	返回值	返回值说明
OLAP_ERR_SCHEMA_CHANGE_INFO_INVALID	-1604	Schema 变更信息无效
OLAP_ERR_QUERY_SPLIT_KEY_ERR	-1605	查询 Split key 错误
OLAP_ERR_DATA_QUALITY_ERR	-1606	模式更改/物化视图期间数据质量问题导致的错误

Column File 错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_COLUMN_DATA_LOAD_BLOCK	-1700	加载列数据块错误
OLAP_ERR_COLUMN_DATA_RECORD_INDEX	-1701	加载数据记录索引错误
OLAP_ERR_COLUMN_DATA_MAKE_FILE_HEADER	-1702	
OLAP_ERR_COLUMN_DATA_READ_VAR_INT	-1703	无法从 Stream 中读取列数据
OLAP_ERR_COLUMN_DATA_PATCH_LIST_NUM	-1704	
OLAP_ERR_COLUMN_STREAM_EOF	-1705	如果数据流结束，返回该代码
OLAP_ERR_COLUMN_READ_STREAM	-1706	块大小大于缓冲区大小，压缩剩余大小小于Stream头大小，读取流失败 这些情况下会抛出该异常
OLAP_ERR_COLUMN_STREAM_NOT_EXIST	-1707	Stream 为空，不存在，未找到数据流 等情况下返回该异常代码
OLAP_ERR_COLUMN_VALUE_NULL	-1708	列值为空异常
OLAP_ERR_COLUMN_SEEK_ERROR	-1709	如果通过 schema 变更添加列，由于schema变更可能导致列索引存在，返回这个异常代码

DeleteHandler 错误代码

返回值名称	返回值	返回值说明

返回值名称	返回值	返回值说明
OLAP_ERR_DELETE_INVALID_CONDITION	-1900	删除条件无效
OLAP_ERR_DELETE_UPDATE_HEADER_FAILED	-1901	删除更新 Header 错误
OLAP_ERR_DELETE_SAVE_HEADER_FAILED	-1902	删除保存 Header 错误
OLAP_ERR_DELETE_INVALID_PARAMETERS	-1903	删除参数无效
OLAP_ERR_DELETE_INVALID_VERSION	-1904	删除版本无效

Cumulative Handler 错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_CUMULATIVE_NO_SUITABLE_VERSIONS	-2000	Cumulative 没有合适的版本
OLAP_ERR_CUMULATIVE_REPEAT_INIT	-2001	Cumulative Repeat 初始化错误
OLAP_ERR_CUMULATIVE_INVALID_PARAMETERS	-2002	Cumulative 参数无效
OLAP_ERR_CUMULATIVE_FAILED_ACQUIRE_DATA_SOURCE	-2003	Cumulative 获取数据源失败
OLAP_ERR_CUMULATIVE_INVALID_NEED_MERGED_VERSIONS	-2004	Cumulative 无有效需要合并版本
OLAP_ERR_CUMULATIVE_ERROR_DELETE_ACTION	-2005	Cumulative 删除操作错误
OLAP_ERR_CUMULATIVE_MISS_VERSION	-2006	rowsets 缺少版本

返回值名称	返回值	返回值说明
OLAP_ERR_CUMULATIVE_CLONE_OCCURRED	-2007	将压缩任务提交到线程池后可能会发生克隆任务，并且选择用于压缩的行集可能会发生变化。在这种情况下，不应执行当前的压缩任务。否则会触发改异常

OLAPMeta 异常代码

返回值名称	返回值	返回值说明
OLAP_ERR_META_INVALID_ARGUMENT	-3000	元数据参数无效
OLAP_ERR_META_OPEN_DB	-3001	打开 DB 元数据错误
OLAP_ERR_META_KEY_NOT_FOUND	-3002	元数据 key 没发现
OLAP_ERR_META_GET	-3003	GET 元数据错误
OLAP_ERR_META_PUT	-3004	PUT 元数据错误
OLAP_ERR_META_ITERATOR	-3005	元数据迭代器错误
OLAP_ERR_META_DELETE	-3006	删除元数据错误
OLAP_ERR_META_ALREADY_EXIST	-3007	元数据已经存在错误

Rowset 错误代码

返回值名称	返回值	返回值说明
OLAP_ERR_ROWSET_WRITER_INIT	-3100	Rowset 写初始化错误
OLAP_ERR_ROWSET_SAVE_FAILED	-3101	Rowset 保存失败
OLAP_ERR_ROWSET_GENERATE_ID_FAILED	-3102	Rowset 生成 ID 失败
OLAP_ERR_ROWSET_DELETE_FILE_FAILED	-3103	Rowset 删除文件失败

返回值名称	返回值	返回值说明
OLAP_ERR_ROWSET_BUILDER_INIT	-3104	Rowset 初始化构建失败
OLAP_ERR_ROWSET_TYPE_NOT_FOUND	-3105	Rowset 类型没有发现
OLAP_ERR_ROWSET_ALREADY_EXIST	-3106	Rowset 已经存在
OLAP_ERR_ROWSET_CREATE_READER	-3107	Rowset 创建读对象失败
OLAP_ERR_ROWSET_INVALID	-3108	Rowset 无效
OLAP_ERR_ROWSET_LOAD_FAILED	-3109	Rowset 加载失败
OLAP_ERR_ROWSET_READER_INIT	-3110	Rowset 读对象初始化失败
OLAP_ERR_ROWSET_READ_FAILED	-3111	Rowset 读失败
OLAP_ERR_ROWSET_INVALID_STATE_TRANSITION	-3112	Rowset 无效的事务状态

磁盘空间管理

最近更新时间：2022-03-14 16:38:36

Doris 的数据磁盘空间如果不加以控制，会因磁盘写满而导致进程挂掉。因此我们监测磁盘的使用率和剩余空间，通过设置不同的警戒水位，来控制 Doris 系统中的各项操作，尽量避免发生磁盘被写满的情况。本文档主要介绍和磁盘存储空间有关的系统参数和处理策略。

名词解释

- FE：Frontend，Doris 的前端节点。负责元数据管理和请求接入。
- BE：Backend，Doris 的后端节点。负责查询执行和数据存储。
- Data Dir：数据目录，在 BE 配置文件 be.conf 的 storage_root_path 中指定的各个数据目录。通常一个数据目录对应一个磁盘、因此下文中 磁盘 也指代一个数据目录。

基本原理

BE 定期（每隔一分钟）会向 FE 汇报一次磁盘使用情况。FE 记录这些统计值，并根据这些统计值，限制不同的操作请求。
在 FE 中分别设置了 高水位（High Watermark）和 危险水位（Flood Stage）两级阈值。危险水位高于高水位。当磁盘使用率高于高水位时，Doris 会限制某些操作的执行（如副本均衡等）。而如果高于危险水位，则会禁止某些操作的执行（如导入）。
同时，在 BE 上也设置了 危险水位（Flood Stage）。考虑到 FE 并不能完全及时的检测到 BE 上的磁盘使用情况，以及无法控制某些 BE 自身运行的操作（如 Compaction）。因此 BE 上的危险水位用于 BE 主动拒绝和停止某些操作，达到自我保护的目的。

FE 参数

高水位：

```
storage_high_watermark_usage_percent 默认 85 (85%)。
storage_min_left_capacity_bytes 默认 2GB。
```

当磁盘空间使用率大于 storage_high_watermark_usage_percent，**或者磁盘空间剩余大小小于** storage_min_left_capacity_bytes **时，该磁盘不会再被作为以下操作的目的路径：**

- Tablet 均衡操作（Balance）
- Colocation 表数据分片的重分布（Relocation）
- Decommission

危险水位：

```
storage_flood_stage_usage_percent 默认 95 (95%)。
storage_flood_stage_left_capacity_bytes 默认 1GB。
```

当磁盘空间使用率大于storage_flood_stage_usage_percent，**并且磁盘空间剩余大小小于** storage_flood_stage_left_capacity_bytes **时，该磁盘不会再被作为以下操作的目的路径，并禁止某些操作：**

- Tablet 均衡操作（Balance）
- Colocation 表数据分片的重分布（Relocation）
- 副本补齐
- 恢复操作（Restore）
- 数据导入（Load/Insert）

BE 参数

危险水位：

```
capacity_used_percent_flood_stage 默认 95 (95%)。
capacity_min_left_bytes_flood_stage 默认 1GB。
```

当磁盘空间使用率大于 storage_flood_stage_usage_percent，**并且磁盘空间剩余大小小于** storage_flood_stage_left_capacity_bytes **时，该磁盘上的以下操作会被禁止：**

- Base/Cumulative Compaction。
- 数据写入。包括各种导入操作。
- Clone Task。通常发生于副本修复或均衡时。
- Push Task。发生在 Hadoop 导入的 Loading 阶段，下载文件。
- Alter Task。Schema Change 或 Rollup 任务。
- Download Task。恢复操作的 Downloading 阶段。

磁盘空间释放

当磁盘空间高于高水位甚至危险水位后，很多操作都会被禁止。此时可以尝试通过以下方式减少磁盘使用率，恢复系统。

- 删除表或分区
  通过删除表或分区的方式，能够快速降低磁盘空间使用率，恢复集群。

> ⚠ 注意：
> 只有 DROP 操作可以达到快速降低磁盘空间使用率的目的， DELETE 操作不可以。

```
DROP TABLE tbl;
ALTER TABLE tbl DROP PARTITION p1;
```

- 扩容 BE
  扩容后，数据分片会自动均衡到磁盘使用率较低的 BE 节点上。扩容操作会根据数据量及节点数量不同，在数小时或数天后使集群到达均衡状态。

- 修改表或分区的副本
  可以将表或分区的副本数降低。例如默认3副本可以降低为2副本。该方法虽然降低了数据的可靠性，但是能够快速的降低磁盘使用率，使集群恢复正常。该方法通常用于紧急恢复系统。请在恢复后，通过扩容或删除数据等方式，降低磁盘使用率后，将副本数恢复为 3。
  修改副本操作为瞬间生效，后台会自动异步的删除多余的副本。

```
ALTER TABLE tbl MODIFY PARTITION p1 SET("replication_num" = "2");
```

- 删除多余文件
  当 BE 进程已经因为磁盘写满而挂掉并无法启动时（此现象可能因 FE 或 BE 检测不及时而发生）。需要通过删除数据目录下的一些临时文件，保证 BE 进程能够启动。以下目录中的文件可以直接删除：
  - log/：日志目录下的日志文件。
  - snapshot/: 快照目录下的快照文件。
  - trash/：回收站中的文件。
    这种操作会对从 BE 回收站中恢复数据产生影响。

    如果BE还能够启动，则可以使用 ADMIN CLEAN TRASH ON(BackendHost:BackendHeartBeatPort); 来主动清理临时文件，会清理 **所有** trash 文件和过期 snapshot 文件，这将影响从回收站恢复数据的操作 。
    如果不手动执行 ADMIN CLEAN TRASH ，系统仍将会在几分钟至几十分钟内自动执行清理，这里分为两种情况：
  - 如果磁盘占用未达到 危险水位(Flood Stage) 的90%，则会清理过期 trash 文件和过期 snapshot 文件，此时会保留一些近期文件而不影响恢复数据。
  - 如果磁盘占用已达到 危险水位(Flood Stage) 的90%，则会清理 **所有** trash 文件和过期 snapshot 文件，**此时会影响从回收站恢复数据的操作** 。

自动执行的时间间隔可以通过配置项中的 max_garbage_sweep_interval 和 max_garbage_sweep_interval 更改。

出现由于缺少 trash 文件而导致恢复失败的情况时，可能返回如下结果：

```
{"status": "Fail","msg": "can find tablet path in trash"}
```

- 删除数据文件

  当以上操作都无法释放空间时，需要通过删除数据文件来释放空间。数据文件在指定数据目录的 data/目录下。删除数据分片（Tablet）必须先确保该 Tablet 至少有一个副本是正常的，否则删除唯一副本会导致数据丢失。假设我们要删除 id 为12345的 Tablet：

  - 找到 Tablet 对应的目录，通常位于 data/shard_id/tablet_id/下。如：

    ```
    data/0/12345/。
    ```

  - 记录 tablet id 和 schema hash。其中 schema hash 为上一步目录的下一级目录名。如下为 352781111：

    ```
    data/0/12345/352781111
    ```

- 删除数据目录

  ```
  rm -rf data/0/12345/
  ```

- 删除 Tablet 元数据（具体参考 Tablet 元数据管理工具）

  ```
  ./lib/meta_tool --operation=delete_header --root_path=/path/to/root_path --tablet_id=12345 --schema_hash= 352781111
  ```

元数据运维

最近更新时间：2022-03-14 15:09:36

本文档主要介绍在实际生产环境中，如何对 Doris 的元数据进行管理。包括 FE 节点建议的部署方式、一些常用的操作方法以及常见错误的解决方法。

> ⚠ 注意：
>
> 当前元数据的设计是无法向后兼容的。即如果新版本有新增的元数据结构变动（可以查看 FE 代码中的 FeMetaVersion.java 文件中是否有新增的 VERSION），那么在升级到新版本后，通常是无法再回滚到旧版本的。

元数据目录结构

我们假设在 fe.conf 中指定的 meta_dir 的路径为 /path/to/palo-meta 。那么一个正常运行中的 Doris 集群，元数据的目录结构应该如下：

```
/path/to/palo-meta/
|-- bdb/
| |-- 00000000.jdb
| |-- je.config.csv
| |-- je.info.0
| |-- je.info.0.lck
| |-- je.lck
| `-- je.stat.csv
`-- image/
|-- ROLE
|-- VERSION
`-- image.xxxx
```

1. bdb 目录。
   - 我们将 bdbje 作为一个分布式的 kv 系统，存放元数据的 journal。这个 bdb 目录相当于 bdbje 的 “数据目录”。
   - 其中 .jdb 后缀的是 bdbje 的数据文件。这些数据文件会随着元数据 journal 的不断增多而越来越多。当 Doris 定期做完 image 后，旧的日志就会被删除。所以正常情况下，这些数据文件的总大小从几 MB 到几 GB 不等（取决于使用 Doris 的方式，如导入频率等）。当数据文件的总大小大于 10GB，则可能需要怀疑是否是因为 image 没有成功，或者分发 image 失败导致的历史 journal 一直无法删除。

- je.info.0 是 bdbje 的运行日志。这个日志中的时间是 UTC+0 时区的。我们可能在后面的某个版本中修复这个问题。通过这个日志，也可以查看一些 bdbje 的运行情况。

2. image 目录。

- image 目录用于存放 Doris 定期生成的元数据镜像文件。通常情况下，您会看到有一个 image.xxxxx 的镜像文件。其中 xxxxx 是一个数字。这个数字表示该镜像包含 xxxxx 号之前的所有元数据 journal。而这个文件的生成时间（通过 ls -al 查看即可）通常就是镜像的生成时间。
- 您也可能会看到一个 image.ckpt 文件。这是一个正在生成的元数据镜像。通过 du -sh 命令应该可以看到这个文件大小在不断变大，说明镜像内容正在写入这个文件。当镜像写完后，会自动重名为一个新的 image.xxxxx 并替换旧的 image 文件。
- 只有角色为 Master 的 FE 才会主动定期生成 image 文件。每次生成完后，都会推送给其他非 Master 角色的 FE。当确认其他所有 FE 都收到这个 image 后，Master FE 会删除 bdbje 中旧的元数据 journal。所以，如果 image 生成失败，或者 image 推送给其他 FE 失败时，都会导致 bdbje 中的数据不断累积。
- ROLE 文件记录了 FE 的类型（FOLLOWER 或 OBSERVER），是一个文本文件。
- VERSION 文件记录了这个 Doris 集群的 cluster id，以及用于各个节点之间访问认证的 token，也是一个文本文件。
- ROLE 文件和 VERSION 文件只可能同时存在，或同时不存在（如第一次启动时）。

基本操作

启动单节点 FE

单节点 FE 是最基本的一种部署方式。一个完整的 Doris 集群，至少需要一个 FE 节点。当只有一个 FE 节点时，这个节点的类型为 Follower，角色为 Master。

1. 第一次启动

   i. 假设在 fe.conf 中指定的 meta_dir 的路径为 /path/to/palo-meta 。

   ii. 确保 /path/to/palo-meta 已存在，权限正确，且目录为空。

   iii. 直接通过 sh bin/start_fe.sh 即可启动。

   iv. 启动后，您应该可以在 fe.log 中看到如下日志：

```
* Palo FE starting...
* image does not exist: /path/to/palo-meta/image/image.0
* transfer from INIT to UNKNOWN
* transfer from UNKNOWN to MASTER
* the very first time to open bdb, dbname is 1
* start fencing, epoch number is 1
* finish replay in xxx msec
```

```
* QE service start
* thrift server started
```

以上日志不一定严格按照这个顺序，但基本类似。

v. 单节点 FE 的第一次启动通常不会遇到问题。如果您没有看到以上日志，一般来说是没有仔细按照文档步骤操作，请仔细阅读相关 wiki。

2. 重启

i. 直接使用 sh bin/start_fe.sh 可以重新启动已经停止的 FE 节点。

ii. 重启后，您应该可以在 fe.log 中看到如下日志：

```
* Palo FE starting...
* finished to get cluster id: xxxx, role: FOLLOWER and node name: xxxx
```

iii. 如果重启前还没有 image 产生，则会看到：

```
* image does not exist: /path/to/palo-meta/image/image.0
```

iv. 如果重启前有 image 产生，则会看到：

```
* start load image from /path/to/palo-meta/image/image.xxx. is ckpt: false
* finished load image in xxx ms

* transfer from INIT to UNKNOWN
* replayed journal id is xxxx, replay to journal id is yyyy
* transfer from UNKNOWN to MASTER
* finish replay in xxx msec
* master finish replay journal, can write now.
* begin to generate new image: image.xxxx
* start save image to /path/to/palo-meta/image/image.ckpt. is ckpt: true
* finished save image /path/to/palo-meta/image/image.ckpt in xxx ms. checksum is xxxx
* push image.xxx to other nodes. totally xx nodes, push successed xx nodes
* QE service start
* thrift server started
```

以上日志不一定严格按照这个顺序，但基本类似。

3. 常见问题

对于单节点 FE 的部署，启停通常不会遇到什么问题。如果有问题，请先参照相关 wiki，仔细核对您的操作步骤。

添加 FE 注意事项和常见问题

1. 注意事项

- 在添加新的 FE 之前，一定先确保当前的 Master FE 运行正常（连接是否正常，JVM 是否正常，image 生成是否正常，bdbje 数据目录是否过大等等）
- 第一次启动新的 FE，一定确保添加了 -helper 参数指向 Master FE。再次启动时可不用添加 -helper。（如果指定了 -helper，FE 会直接询问 helper 节点自己的角色，如果没有指定，FE会尝试从 palo-meta/image/ 目录下的 ROLE 和 VERSION 文件中获取信息）。
- 第一次启动新的 FE，一定确保这个 FE 的 meta_dir 已经创建、权限正确且为空。
  - 启动新的 FE，和执行 ALTER SYSTEM ADD FOLLOWER/OBSERVER 语句在元数据添加 FE，这两个操作的顺序没有先后要求。如果先启动了新的 FE，而没有执行语句，则新的 FE 日志中会一直滚动 current node is not added to the group. please add it first. 字样。当执行语句后，则会进入正常流程。
- 请确保前一个 FE 添加成功后，再添加下一个 FE。
- 建议直接连接到 MASTER FE 执行 ALTER SYSTEM ADD FOLLOWER/OBSERVER 语句。

2. 常见问题

i. this node is DETACHED

当第一次启动一个待添加的 FE 时，如果 Master FE 上的 palo-meta/bdb 中的数据很大，则可能在待添加的 FE 日志中看到 this node is DETACHED. 字样。这时，bdbje 正在复制数据，您可以看到待添加的 FE 的 bdb/ 目录正在变大。这个过程通常会在数分钟不等（取决于 bdbje 中的数据量）。之后，fe.log 中可能会有一些 bdbje 相关的错误堆栈信息。如果最终日志中显示 QE service start 和 thrift server started，则通常表示启动成功。可以通过 mysql-client 连接这个 FE 尝试操作。如果没有出现这些字样，则可能是 bdbje 复制日志超时等问题。这时，直接再次重启这个 FE，通常即可解决问题。

ii. 各种原因导致添加失败

- 如果添加的是 OBSERVER，因为 OBSERVER 类型的 FE 不参与元数据的多数写，理论上可以随意启停。因此，对于添加 OBSERVER 失败的情况。可以直接停止 OBSERVER FE 的进程，清空 OBSERVER 的元数据目录后，重新进行一遍添加流程。
- 如果添加的是 FOLLOWER，因为 FOLLOWER 是参与元数据多数写的。所以有可能FOLLOWER 已经加入 bdbje 选举组内。如果这时只有两个 FOLLOWER 节点（包括 MASTER），那么停掉一个 FE，可能导致另一个 FE 也因无法进行多数写而退出。此时，我们应该先通过 ALTER SYSTEM DROP FOLLOWER 命令，从元数据中删除新添加的 FOLLOWER 节点，然后再停止 FOLLOWER 进程，清空元数据，重新进行一遍添加流程。

删除 FE

通过 ALTER SYSTEM DROP FOLLOWER/OBSERVER 命令即可删除对应类型的 FE。以下有几点注意事项：

- 对于 OBSERVER 类型的 FE，直接 DROP 即可，无风险。
- 对于 FOLLOWER 类型的 FE。首先，应保证在有奇数个 FOLLOWER 的情况下（3个或以上），开始删除操作。
  - 如果删除非 MASTER 角色的 FE，建议连接到 MASTER FE，执行 DROP 命令，再停止进程即可。
  - 如果要删除 MASTER FE，先确认有奇数个 FOLLOWER FE 并且运行正常。然后先停止 MASTER FE 的进程。这时会有某一个 FE 被选举为 MASTER。在确认剩下的 FE 运行正常后，连接到新的 MASTER FE，执行 DROP 命令删除之前老的 MASTER FE 即可。

高级操作

故障恢复

FE 有可能因为某些原因出现无法启动 bdbje、FE 之间无法同步等问题。现象包括无法进行元数据写操作、没有 MASTER 等等。这时，我们需要手动操作来恢复 FE。手动恢复 FE 的大致原理，是先通过当前 meta_dir 中的元数据，启动一个新的 MASTER，然后再逐台添加其他 FE。请严格按照如下步骤操作：

1. 首先，停止所有 FE 进程，同时停止一切业务访问。保证在元数据恢复期间，不会因为外部访问导致其他不可预期的问题。

2. 确认哪个 FE 节点的元数据是最新：
   i. 首先，**务必先备份所有 FE 的 meta_dir 目录。**
   ii. 通常情况下，Master FE 的元数据是最新的。可以查看 meta_dir/image 目录下，image.xxxx 文件的后缀，数字越大，则表示元数据越新。
   iii. 通常，通过比较所有 FOLLOWER FE 的 image 文件，找出最新的元数据即可。
   iv. 之后，我们要使用这个拥有最新元数据的 FE 节点，进行恢复。
   v. 如果使用 OBSERVER 节点的元数据进行恢复会比较麻烦，建议尽量选择 FOLLOWER 节点。

3. 以下操作都在由第2步中选择出来的 FE 节点上进行。
   i. 如果该节点是一个 OBSERVER，先将 meta_dir/image/ROLE 文件中的 role=OBSERVER 改为 role=FOLLOWER。（从 OBSERVER 节点恢复会比较麻烦，先按这里的步骤操作，后面会有单独说明）
   ii. 在 fe.conf 中添加配置：metadata_failure_recovery=true。
   iii. 执行 sh bin/start_fe.sh 启动这个 FE。
   iv. 如果正常，这个 FE 会以 MASTER 的角色启动，类似于前面 启动单节点 FE 一节中的描述。在 fe.log 应该会看到 transfer from XXXX to MASTER 等字样。
   v. 启动完成后，先连接到这个 FE，执行一些查询导入，检查是否能够正常访问。如果不正常，有可能是操作有误，建议仔细阅读以上步骤，用之前备份的元数据再试一次。如果还是不行，问题可能就比较严重了。
   vi. 如果成功，通过 show frontends; 命令，应该可以看到之前所添加的所有 FE，并且当前 FE 是 master。
   vii. 将 fe.conf 中的 metadata_failure_recovery=true 配置项删除，或者设置为 false，然后重启这个 FE（**重要**）。

如果您是从一个 OBSERVER 节点的元数据进行恢复的，那么完成如上步骤后，通过 show frontends; 语句您会发现，当前这个 FE 的角色为 OBSERVER，但是 IsMaster 显示为 true 。这是因为，这里看到的“OBSERVER” 是记录在 Doris 的元数据中的，而是否是 master，是记录在 bdbje 的元数据中的。因为我们是从一个 OBSERVER 节点恢复的，所以这里出现了不一致。请按如下步骤修复这个问题（这个问题我们会在之后的某个版本修复）：

- 先把除了这个 “OBSERVER” 以外的所有 FE 节点 DROP 掉。
- 通过 ADD FOLLOWER 命令，添加一个新的 FOLLOWER FE，假设在 hostA 上。
- 在 hostA 上启动一个全新的 FE，通过 -helper 的方式加入集群。
- 启动成功后，通过 show frontends; 语句，您应该能看到两个 FE，一个是之前的 OBSERVER，一个是新添加的 FOLLOWER，并且 OBSERVER 是 master。
- 确认这个新的 FOLLOWER 是可以正常工作之后，用这个新的 FOLLOWER 的元数据，重新执行一遍故障恢复操作。
- 以上这些步骤的目的，其实就是人为的制造出一个 FOLLOWER 节点的元数据，然后用这个元数据，重新开始故障恢复。这样就避免了从 OBSERVER 恢复元数据所遇到的不一致的问题。

metadata_failure_recovery=true 的含义是，清空 "bdbje" 的元数据。这样 bdbje 就不会再联系之前的其他 FE 了，而作为一个独立的 FE 启动。这个参数只有在恢复启动时才需要设置为 true。恢复完成后，一定要设置为 false，否则一旦重启，bdbje 的元数据又会被清空，导致其他 FE 无法正常工作。

4. 第3步执行成功后，我们再通过 ALTER SYSTEM DROP FOLLOWER/OBSERVER 命令，将之前的其他的 FE 从元数据删除后，按加入新 FE 的方式，重新把这些 FE 添加一遍。
5. 如果以上操作正常，则恢复完毕。

FE 类型变更

如果您需要将当前已有的 FOLLOWER/OBSERVER 类型的 FE，变更为 OBSERVER/FOLLOWER 类型，请先按照前面所述的方式删除 FE，再添加对应类型的 FE 即可

FE 迁移

如果您需要将一个 FE 从当前节点迁移到另一个节点，分以下几种情况。

1. 非 MASTER 节点的 FOLLOWER，或者 OBSERVER 迁移。
   直接添加新的 FOLLOWER/OBSERVER 成功后，删除旧的 FOLLOWER/OBSERVER 即可。

2. 单节点 MASTER 迁移。
   当只有一个 FE 时，参考 故障恢复 一节。将 FE 的 palo-meta 目录拷贝到新节点上，按照 故障恢复 一节中，步骤3的方式启动新的 MASTER

3. 一组 FOLLOWER 从一组节点迁移到另一组新的节点。
   在新的节点上部署 FE，通过添加 FOLLOWER 的方式先加入新节点。再逐台 DROP 掉旧节点即可。在逐台 DROP 的过程中，MASTER 会自动选择在新的 FOLLOWER 节点上。

更换 FE 端口

FE 目前有以下几个端口：

- **edit_log_port：bdbje 的通信端口。**
  如果需要更换这个端口，则需要参照 故障恢复 一节中的操作，进行恢复。因为该端口已经被持久化到 bdbje 自己的元数据中（同时也记录在 Doris 自己的元数据中），需要通过设置 metadata_failure_recovery=true 来清空 bdbje 的元数据。

- **http_port：http 端口，也用于推送 image。**
  所有 FE 的 http_port 必须保持一致。所以如果要修改这个端口，则所有 FE 都需要修改并重启。修改这个端口，在多 FOLLOWER 部署的情况下会比较复杂（涉及到鸡生蛋蛋生鸡的问题...），所以不建议有这种操作。如果必须，直接按照 故障恢复 一节中的操作吧。

- **rpc_port：FE 的 thrift server port。**
  修改配置后，直接重启 FE 即可。Master FE 会通过心跳将新的端口告知 BE。只有 Master FE 的这个端口会被使用。但仍然建议所有 FE 的端口保持一致。

- **query_port：Mysql 连接端口。**
  修改配置后，直接重启 FE 即可。这个只影响到 mysql 的连接目标。

从 FE 内存中恢复元数据

在某些极端情况下，磁盘上 image 文件可能会损坏，但是内存中的元数据是完好的，此时我们可以先从内存中 dump 出元数据，再替换掉磁盘上的 image 文件，来恢复元数据，整个**不停查询服务**的操作步骤如下：

1. 集群停止所有 Load，Create，Alter 操作。
2. 执行以下命令，从 Master FE 内存中 dump 出元数据：（下面称为 image_mem）。

```
curl -u $root_user:$password http://$master_hostname:8030/dump
```

3. 用 image_mem 文件替换掉 OBSERVER FE 节点上meta_dir/image目录下的 image 文件，重启 OBSERVER FE 节点，验证 image_mem 文件的完整性和正确性（可以在 FE Web 页面查看 DB 和 Table 的元数据是否正常，查看 fe.log 是否有异常，是否在正常 replayed journal）。
4. 依次用 image_mem 文件替换掉 FOLLOWER FE 节点上meta_dir/image目录下的 image 文件，重启 FOLLOWER FE 节点，确认元数据和查询服务都正常。
5. 用 image_mem 文件替换掉 Master FE 节点上meta_dir/image目录下的 image 文件，重启 Master FE 节点，确认 FE Master 切换正常， Master FE 节点可以通过 checkpoint 正常生成新的 image 文件。
6. 集群恢复所有 Load，Create，Alter 操作。

> ⚠ 注意：

- 如果 Image 文件很大，整个操作过程耗时可能会很长，所以在此期间，要确保 Master FE 不会通过 checkpoint 生成新的 image 文件。
- 当观察到 Master FE 节点上 meta_dir/image 目录下的 image.ckpt 文件快和 image.xxx 文件一样大时，可以直接删除掉 image.ckpt 文件。

查看 BDBJE 中的数据

FE 的元数据日志以 Key-Value 的方式存储在 BDBJE 中。某些异常情况下，可能因为元数据错误而无法启动 FE。在这种情况下，Doris 提供一种方式可以帮助用户查询 BDBJE 中存储的数据，以方便进行问题排查。

- 首先需在 fe.conf 中增加配置：enable_bdbje_debug_mode=true，之后通过 sh start_fe.sh --daemon 启动 FE。
- 此时，FE 将进入 debug 模式，仅会启动 http server 和 MySQL server，并打开 BDBJE 实例，但不会进行任何元数据的加载及后续其他启动流程。
- 这时，我们可以通过访问 FE 的 web 页面，或通过 MySQL 客户端连接到 Doris 后，通过 show proc /bdbje; 来查看 BDBJE 中存储的数据。

```
mysql> show proc "/bdbje";
+----------+---------------+---------+
| DbNames | JournalNumber | Comment |
+----------+---------------+---------+
| 110589 | 4273 | |
| epochDB | 4 | |
| metricDB | 430694 | |
+----------+---------------+---------+
```

第一级目录会展示 BDBJE 中所有的 database 名称，以及每个 database 中的 entry 数量。

```
mysql> show proc "/bdbje/110589";
+-----------+
| JournalId |
+-----------+
| 1 |
| 2 |

...
| 114858 |
| 114859 |
| 114860 |
```

```
| 114861 |
+----------+
4273 rows in set (0.06 sec)
```

进入第二级，则会罗列指定 database 下的所有 entry 的 key。

```
mysql> show proc "/bdbje/110589/114861";
+----------+-------------+-----------------------------------------+
| JournalId | OpType | Data |
+----------+-------------+-----------------------------------------+
| 114861 | OP_HEARTBEAT | org.apache.doris.persist.HbPackage@6583d5fb |
+----------+-------------+-----------------------------------------+
1 row in set (0.05 sec)
```

第三级则可以展示指定 key 的 value 信息。

最佳实践

如果您并不十分了解 FE 元数据的运行逻辑，或者没有足够 FE 元数据的运维经验，我们强烈建议在实际使用中，只部署一个 FOLLOWER 类型的 FE 作为 MASTER，其余 FE 都是 OBSERVER，这样可以减少很多复杂的运维问题！ 不用过于担心 MASTER 单点故障导致无法进行元数据写操作。首先，如果您配置合理，FE 作为 java 进程很难挂掉。其次，如果 MASTER 磁盘损坏（概率非常低），我们也可以用 OBSERVER 上的元数据，通过**故障恢复**的方式手动恢复。

FE 进程的 JVM 一定要保证足够的内存。我们**强烈建议** FE 的 JVM 内存至少在 10GB 以上，推荐 32GB 至 64GB。并且部署监控来监控 JVM 的内存使用情况。因为如果FE出现OOM，可能导致元数据写入失败，造成一些**无法恢复**的故障！

FE 所在节点要有足够的磁盘空间，以防止元数据过大导致磁盘空间不足。同时 FE 日志也会占用十几G 的磁盘空间。

其他常见问题

1. fe.log 中一直滚动 meta out of date. current time: xxx, synchronized time: xxx, has log: xxx, fe type: xxx 。

   这个通常是因为 FE 无法选举出 Master。例如配置了 3 个 FOLLOWER，但是只启动了一个 FOLLOWER，则这个 FOLLOWER 会出现这个问题。通常，只要把剩余的 FOLLOWER 启动起来就可以了。如果启动起来后，仍然没有解决问题，那么可能需要按照**故障恢复**一节中的方式，手动进行恢复。

2. Clock delta: xxxx ms. between Feeder: xxxx and this Replica exceeds max permissible delta: xxxx ms. 。
   bdbje 要求各个节点之间的时钟误差不能超过一定阈值。如果超过，节点会异常退出。我们默认设置的阈值为 5000 ms，由 FE 的参数 max_bdbje_clock_delta_ms 控制，可以酌情修改。但我们建议使用 ntp 等时钟同步方式保证 Doris 集群各主机的时钟同步。

3. image/ 目录下的镜像文件很久没有更新。
   Master FE 会默认每 50000 条元数据 journal，生成一个镜像文件。在一个频繁使用的集群中，通常每隔半天到几天的时间，就会生成一个新的 image 文件。如果您发现 image 文件已经很久没有更新了（例如超过一个星期），则可以顺序的按照如下方法，查看具体原因：
   i. 在 Master FE 的 fe.log 中搜索 memory is not enough to do checkpoint. Committed memroy xxxx Bytes, used memory xxxx Bytes. 字样。如果找到，则说明当前 FE 的 JVM 内存不足以用于生成镜像（通常我们需要预留一半的 FE 内存用于 image 的生成）。那么需要增加 JVM 的内存并重启 FE 后，再观察。每次 Master FE 重启后，都会直接生成一个新的 image。也可用这种重启方式，主动地生成新的 image。注意，如果是多 FOLLOWER 部署，那么当您重启当前 Master FE 后，另一个 FOLLOWER FE 会变成 MASTER，则后续的 image 生成会由新的 Master 负责。因此，您可能需要修改所有 FOLLOWER FE 的 JVM 内存配置。
   ii. 在 Master FE 的 fe.log 中搜索 begin to generate new image: image.xxxx。如果找到，则说明开始生成 image 了。检查这个线程的后续日志，如果出现 checkpoint finished save image.xxxx，则说明 image 写入成功。如果出现 Exception when generate new image file，则生成失败，需要查看具体的错误信息。

4. bdb/ 目录的大小非常大，达到几个G或更多。
   如果在排除无法生成新的 image 的错误后，bdb 目录在一段时间内依然很大。则可能是因为 Master FE 推送 image 不成功。可以在 Master FE 的 fe.log 中搜索 push image.xxxx to other nodes. totally xx nodes, push successed yy nodes 。如果 yy 比 xx 小，则说明有的 FE 没有被推送成功。可以在 fe.log 中查看到具体的错误 Exception when pushing image file. url = xxx 。
   同时，您也可以在 FE 的配置文件中添加配置： edit_log_roll_num=xxxx 。该参数设定了每多少条元数据 journal，做一次 image。默认是 50000。可以适当改小这个数字，使得 image 更加频繁，从而加速删除旧的 journal。

5. FOLLOWER FE 接连挂掉。
   因为 Doris 的元数据采用多数写策略，即一条元数据 journal 必须至少写入多数个 FOLLOWER FE 后（例如 3 个 FOLLOWER，必须写成功 2 个），才算成功。而如果写入失败，FE 进程会主动退出。那么假设有 A、B、C 三个 FOLLOWER，C 先挂掉，然后 B 再挂掉，那么 A 也会跟着挂掉。所以如 最佳实践 一节中所述，如果您没有丰富的元数据运维经验，不建议部署多 FOLLOWER。

6. fe.log 中出现 get exception when try to close previously opened bdb database. ignore it 。
   如果后面有 ignore it 字样，通常无需处理。如果您有兴趣，可以在 BDBEnvironment.java 搜索这个错误，查看相关注释说明。

7. 从 show frontends; 看，某个 FE 的 Join 列为 true，但是实际该 FE 不正常。
   通过 show frontends; 查看到的 Join 信息。该列如果为 true，仅表示这个 FE 曾经加入过 集群。并不能表示当前仍然正常的存在于集群中。如果为 false，则表示这个 FE 从未加入过 集群。

8. 关于 FE 的配置 master_sync_policy，replica_sync_policy 和 txn_rollback_limit。
   master_sync_policy 用于指定当 Leader FE 写元数据日志时，是否调用 fsync()，replica_sync_policy 用于指定当 FE HA 部署时，其他 Follower FE 在同步元数据时，是否调用 fsync()。在早期的 Doris 版本中，这两个参数默认是 WRITE_NO_SYNC，即都不调用 fsync()。在最新版本的 Doris 中，默认已修改为 SYNC，即都调用 fsync()。调用 fsync() 会显著降低元数据写盘的效率。在某些环境下，IOPS 可能降至几百，延迟增加到 2-3ms（但对于 Doris 元数据操作依然够用）。因此我们建议以下配置：
   i. 对于单 Follower FE 部署，master_sync_policy 设置为 SYNC，防止 FE 系统宕机导致元数据丢失。
   ii. 对于多 Follower FE 部署，可以将 master_sync_policy 和 replica_sync_policy 设为 WRITE_NO_SYNC，因为我们认为多个系统同时宕机的概率非常低。
   如果在单 Follower FE 部署中，master_sync_policy 设置为 WRITE_NO_SYNC，则可能出现 FE 系统宕机导致元数据丢失。这时如果有其他 Observer FE 尝试重启时，可能会报错：

   ```
   Node xxx must rollback xx total commits(numPassedDurableCommits of which were durable) to the earliest point indicated by transaction xxxx in order to rejoin the replication group, but the transaction rollback limit of xxx prohibits this.
   ```

   意思是有部分已经持久化的事务需要回滚，但条数超过上限。这里我们的默认上限是 100，可以通过设置 txn_rollback_limit 改变。该操作仅用于尝试正常启动 FE，但已丢失的元数据无法恢复。

参数配置
FE 配置参数

最近更新时间：2022-03-14 16:40:23

本文档主要介绍 FE 的相关配置项。FE 的配置文件 fe.conf 通常存放在 FE 部署路径的 conf/ 目录下。而在 0.14 版本中会引入另一个配置文件 fe_custom.conf 。该配置文件用于记录用户在运行是动态配置并持久化的配置项。

FE 进程启动后，会先读取 fe.conf 中的配置项，之后再读取 fe_custom.conf 中的配置项。

fe_custom.conf 中的配置项会覆盖 fe.conf 中相同的配置项。

fe_custom.conf 文件的位置可以在 fe.conf 通过 custom_config_dir 配置项配置。

查看配置项

FE 的配置项有两种方式进行查看：

1. FE 前端页面查看。

   在浏览器中打开 FE 前端页面 http://fe_host:fe_http_port/variable 。在 Configure Info 中可以看到当前生效的 FE 配置项。

2. 通过命令查看。

   FE 启动后，可以在 MySQL 客户端中，通过以下命令查看 FE 的配置项：

   ```
   ADMIN SHOW FRONTEND CONFIG;
   ```

   结果中各列含义如下：

   - Key：配置项名称。
   - Value：当前配置项的值。
   - Type：配置项值类型，如果整型、字符串。
   - IsMutable：是否可以动态配置。如果为 true，表示该配置项可以在运行时进行动态配置。如果false，则表示该配置项只能在 fe.conf 中配置并且重启 FE 后生效。
   - MasterOnly：是否为 Master FE 节点独有的配置项。如果为 true，则表示该配置项仅在 Master FE 节点有意义，对其他类型的 FE 节点无意义。如果为 false，则表示该配置项在所有 FE 节点中均有意义。
   - Comment：配置项的描述。

设置配置项

FE 的配置项有两种方式进行配置：

1. 静态配置。

   在 conf/fe.conf 文件中添加和设置配置项。fe.conf 中的配置项会在 FE 进程启动时被读取。没有在 fe.conf 中的配置项将使用默认值。

2. 通过 MySQL 协议动态配置。
   FE 启动后，可以通过以下命令动态设置配置项。该命令需要管理员权限。
   ADMIN SET FRONTEND CONFIG ("fe_config_name" = "fe_config_value");
   不是所有配置项都支持动态配置。可以通过 ADMIN SHOW FRONTEND CONFIG; 命令结果中的 IsMutable 列查看是否支持动态配置。 如果是修改 MasterOnly 的配置项，则该命令会直接转发给 Master FE 并且仅修改 Master FE 中对应的配置项。
   通过该方式修改的配置项将在 FE 进程重启后失效。
   更多该命令的帮助，可以通过 HELP ADMIN SET CONFIG; 命令查看。

3. 通过 HTTP 协议动态配置。
   该方式也可以持久化修改后的配置项。配置项将持久化在 fe_custom.conf 文件中，在 FE 重启后仍会生效。

应用举例

1. 修改 async_pending_load_task_pool_size 。
   通过 ADMIN SHOW FRONTEND CONFIG; 可以查看到该配置项不能动态配置（ IsMutable 为 false ）。则需要在 fe.conf 中添加： async_pending_load_task_pool_size=20 ，之后重启 FE 进程以生效该配置。

2. 修改 dynamic_partition_enable 。
   通过 ADMIN SHOW FRONTEND CONFIG; 可以查看到该配置项可以动态配置（ IsMutable 为 true ）。并且是 Master FE 独有配置。则首先我们可以连接到任意 FE，执行如下命令修改配置：

   ```
   ADMIN SET FRONTEND CONFIG ("dynamic_partition_enable" = "true");`
   ```

   之后可以通过如下命令查看修改后的值：

   ```
   set forward_to_master=true;
   ADMIN SHOW FRONTEND CONFIG;
   ```

   通过以上方式修改后，如果 Master FE 重启或进行了 Master 切换，则配置将失效。可以通过在 fe.conf 中直接添加配置项，并重启 FE 后，永久生效该配置项。

3. 修改 max_distribution_pruner_recursion_depth 。
   通过 ADMIN SHOW FRONTEND CONFIG; 可以查看到该配置项可以动态配置（ IsMutable 为 true ）。并且不是 Master FE 独有配置。
   同样，我们可以通过动态修改配置的命令修改该配置。因为该配置不是 Master FE 独有配置，所以需要单独连接到不同的 FE，进行动态修改配置的操作，这样才能保证所有 FE 都使用了修改后的配置值。

配置项列表

max_dynamic_partition_num

- 默认值：500。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 用于限制创建动态分区表时可以创建的最大分区数，避免一次创建过多分区。 数量由动态分区参数中的“开始”和“结束”决定。

grpc_max_message_size_bytes

- 默认值：1G。
- 用于设置 GRPC 客户端通道的初始流窗口大小，也用于设置最大消息大小。当结果集较大时，可能需要增大该值。

enable_outfile_to_local

- 默认值：false。
- 是否允许outfile函数将结果导出到本地磁盘。

enable_access_file_without_broker

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
  此配置用于在通过代理访问 bos 或其他云存储时尝试跳过代理。

enable_bdbje_debug_mode

- 默认值：false。
- 如果设置为 true，FE 将在 BDBJE 调试模式下启动，在 Web 页面 System->bdbje 可以查看相关信息，否则不可以查看。

enable_fe_heartbeat_by_thrift

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 这个配置是用来解决 FE 心跳响应读超时的问题，当 config 设置为 true 时，master 会通过 thrift 协议而不是 http 协议来获取 fe heartbeat response。 为了保持与旧版本的兼容性，默认为 false，在所有 fe 都升级之前不能将配置改为 true。

enable_alpha_rowset

- 默认值：false。

- 是否支持创建 alpha rowset。默认为 false，只应在紧急情况下使用，此配置应在未来的某个版本中删除。

enable_http_server_v2

- 默认值：从官方 0.14.0 release 版之后默认是 true，之前默认 false。
- HTTP Server V2 由 SpringBoot 实现。它采用前后端分离的架构。只有启用 httpv2 才能用户使用新的前端 UI 界面。

jetty_server_acceptors

- 默认值：2。

jetty_server_selectors

- 默认值：4。

jetty_server_workers

- 默认值：0。

以上3个参数，Jetty 的线程架构模型非常简单，分为 acceptors，selectors 和 workers 三个线程池。acceptors 负责接受新连接，然后交给 selectors 处理 HTTP消息协议的解包，最后由 workers 处理请求。前两个线程池采用非阻塞模型，一个线程可以处理很多 socket 的读写，所以线程池数量较小。

大多数项目，acceptors 线程只需要1-2个，selectors 线程配置2～4个足矣。workers 是阻塞性的业务逻辑，往往有较多的数据库操作，需要的线程数量较多，具体数量随应用程序的 QPS 和 IO 事件占比而定。QPS 越高，需要的线程数量越多，IO 占比越高，等待的线程数越多，需要的总线程数也越多。

workers 线程池默认不做设置，根据自己需要进行设置。

jetty_server_max_http_post_size

- 默认值：100 * 1024 * 1024（100MB）。

这个是put或post方法上传文件的最大字节数，默认值：100MB。

default_max_filter_ratio

- 默认值：0。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。

可过滤数据的最大百分比（由于数据不规则等原因）默认值为0。表示严格模式，只要数据有一条被过滤掉整个导入失败。

default_db_data_quota_bytes

- 默认值：1PB。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 用于设置默认数据库数据配额大小，设置单个数据库的配额大小可以使用：

```
设置数据库数据量配额，单位为B/K/KB/M/MB/G/GB/T/TB/P/PB
ALTER DATABASE db_name SET DATA QUOTA quota;
查看配置
show data （其他用法：HELP SHOW DATA）
```

enable_batch_delete_by_default

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 创建唯一表时是否添加删除标志列，具体原理参照官方文档：**操作手册 > 数据导入 > 批量删除**。

recover_with_empty_tablet

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 在某些情况下，某些 tablet 可能会损坏或丢失所有副本。 此时数据已经丢失，损坏的 tablet 会导致整个查询失败，无法查询剩余的健康 tablet。 在这种情况下，您可以将此配置设置为 true。 系统会将损坏的药片替换为空药片，以确保查询可以执行。（但此时数据已经丢失，所以查询结果可能不准确）。

max_allowed_in_element_num_of_delete

- 默认值：1024。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 用于限制 delete 语句中 Predicate 的元素个数。

cache_result_max_row_count

- 默认值：3000。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：false。

设置可以缓存的最大行数，详细的原理可以参考官方文档：**操作手册 > 分区缓存**。

cache_last_version_interval_second

- 默认值：900。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：false。
- 缓存结果时上一版本的最小间隔，该参数区分离线更新和实时更新。

cache_enable_partition_mode

- 默认值：true。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：false。
- 如果设置为 true，fe 将从 be cache 中获取数据，该选项适用于部分分区的实时更新。

cache_enable_sql_mode

- 默认值：true。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：false。
- 如果设置为 true，fe 会启用 sql 结果缓存该选项适用于离线数据更新场景。

	case1	case2	case3	case4
enable_sql_cache	false	true	true	false
enable_partition_cache	false	false	true	true

min_clone_task_timeout_sec 和 max_clone_task_timeout_sec

- 默认值：最小3分钟，最大两小时。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- min_clone_task_timeout_sec 和 max_clone_task_timeout_sec 用于限制克隆任务的最小和最大超时间。一般情况下，克隆任务的超时时间是通过数据量和最小传输速度（5MB/s）来估计的。但在特殊情况下，您可能需要手动设置这两个配置，以确保克隆任务不会因超时而失败。

agent_task_resend_wait_time_ms

- 默认值：5000。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 当代理任务的创建时间被设置的时候，此配置将决定是否重新发送代理任务， 当且仅当当前时间减去创建时间大于 agent_task_task_resend_wait_time_ms 时，ReportHandler 可以重新发送代理任务。

- 该配置目前主要用来解决 PUBLISH_VERSION 代理任务的重复发送问题, 目前该配置的默认值是5000，是个实验值，由于把代理任务提交到代理任务队列和提交到be存在一定的时间延迟，所以调大该配置的值可以有效解决代理任务的重复发送问题，但同时会导致提交失败或者执行失败的代理任务再次被执行的时间延长。

enable_odbc_table。

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 是否启用 ODBC 表，默认不启用，在使用的时候需要手动配置启用，该参数可以通过：ADMIN SET FRONTEND CONFIG(“key”=“value”)方式进行设置。

enable_spark_load

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 是否临时启用 spark load，默认不启用。

disable_storage_medium_check

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果 disable_storage_medium_check 为true， ReportHandler 将不会检查 tablet 的存储介质， 并使得存储冷却功能失效，默认值为false。当您不关心 tablet 的存储介质是什么时，可以将值设置为true 。

drop_backend_after_decommission

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 该配置用于控制系统在成功下线（Decommission）BE 后，是否 Drop 该 BE。如果为 true，则在 BE 成功下线后，会删除掉该BE节点。如果为 false，则在 BE 成功下线后，该 BE 会一直处于 DECOMMISSION 状态，但不会被删除。
- 该配置在某些场景下可以发挥作用。假设一个 Doris 集群的初始状态为每个 BE 节点有一块磁盘。运行一段时间后，系统进行了纵向扩容，即每个 BE 节点新增2块磁盘。因为 Doris 当前还不支持 BE 内部各磁盘间的数据均衡，所以会导致初始磁盘的数据量可能一直远高于新增磁盘的数据量。此时我们可以通过以下操作进行人工的磁盘间均衡：
  i. 将该配置项置为 false。
  ii. 对某一个 BE 节点，执行 decommission 操作，该操作会将该 BE 上的数据全部迁移到其他节点中。

iii. decommission 操作完成后，该 BE 不会被删除。此时，取消掉该 BE 的 decommission 状态。则数据会开始从其他 BE 节点均衡回这个节点。此时，数据将会均匀的分布到该 BE 的所有磁盘上。

iv. 对所有 BE 节点依次执行 2，3 两个步骤，最终达到所有节点磁盘均衡的目的。

period_of_auto_resume_min

- 默认值：5（s）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 自动恢复 Routine load 的周期。

max_tolerable_backend_down_num

- 默认值：0。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 只要有一个BE宕机，Routine Load 就无法自动恢复。

enable_materialized_view

- 默认值：true。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 该配置用于开启和关闭创建物化视图功能。如果设置为 true，则创建物化视图功能开启。用户可以通过 CREATE MATERIALIZED VIEW 命令创建物化视图。如果设置为 false，则无法创建物化视图。
- 如果在创建物化视图的时候报错 The materialized view is coming soon 或 The materialized view is disabled 则说明改配置被设置为了 false，创建物化视图功能关闭了。可以通过修改配置为 true 来启动创建物化视图功能。
- 该变量为动态配置，用户可以在 FE 进程启动后，通过命令修改配置。也可以通过修改 FE 的配置文件，重启 FE 来生效。

check_java_version

- 默认值：false。
- 如果设置为 true，Doris 将检查已编译和运行的 Java 版本是否兼容。

max_running_rollup_job_num_per_table

- 默认值：1。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 控制 Rollup 作业并发限制。

dynamic_partition_enable

- 默认值：true。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 是否启用动态分区，默认启用。

dynamic_partition_check_interval_seconds

- 默认值：600秒，10分钟。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 检查动态分区的频率。

disable_cluster_feature

- 默认值：true。
- 是否可以动态配置：true。
- 多集群功能将在 0.12 版本中弃用 ，将此配置设置为 true 将禁用与集群功能相关的所有操作，包括：
  i. 创建/删除集群。
  ii. 添加、释放BE/将BE添加到集群/停用集群balance。
  iii. 更改集群的后端数量。
  iv. 链接/迁移数据库。

force_do_metadata_checkpoint

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果设置为 true，则无论 jvm 内存使用百分比如何，检查点线程都会创建检查点。

metadata_checkpoint_memory_threshold

- 默认值：60（60%）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果 jvm 内存使用百分比（堆或旧内存池）超过此阈值，则检查点线程将无法工作以避免 OOM。

max_distribution_pruner_recursion_depth

- 默认值：100。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：false。

- 这将限制哈希分布修剪器的最大递归深度。 例如：其中 a in（5 个元素）和 b in（4 个元素）和 c in（3 个元素）和 d in（2 个元素）。 a/b/c/d 是分布式列，所以递归深度为 5 * 4 * 3 * 2 = 120，大于 100， 因此该分发修剪器将不起作用，只会返回所有 buckets。 增加深度可以支持更多元素的分布修剪，但可能会消耗更多的 CPU。
- 通过 ADMIN SHOW FRONTEND CONFIG; 可以查看到该配置项可以动态配置（IsMutable 为 true）。并且不是 Master FE 独有配置。
- 同样，我们可以通过动态修改配置的命令修改该配置。因为该配置不是 Master FE 独有配置，所以需要单独连接到不同的 FE，进行动态修改配置的操作，这样才能保证所有 FE 都使用了修改后的配置值。

using_old_load_usage_pattern

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果设置为 true，处理错误的 insert stmt 仍将返回一个标签给用户。 用户可以使用此标签来检查加载作业的状态。 默认值为 false，表示插入操作遇到错误，不带加载标签，直接抛出异常给用户客户端。

small_file_dir

- 默认值：DORIS_HOME_DIR + “/small_files”。
- 保存小文件的目录。

max_small_file_size_bytes

- 默认值：1M。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- SmallFileMgr 中单个文件存储的最大大小 。

max_small_file_number

- 默认值：100。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- SmallFileMgr 中存储的最大文件数。

max_routine_load_task_num_per_be

- 默认值：5。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。

- 每个 BE 的最大并发例 Routine Load 任务数。这是为了限制发送到 BE 的 Routine Load 任务的数量，并且它也应该小于 BE config routine_load_thread_pool_size（默认 10），这是 BE 上的 Routine Load 任务线程池大小。

max_routine_load_task_concurrent_num

- 默认值：5。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 单个 Routine Load 作业的最大并发任务数。

max_routine_load_job_num

- 默认值：100。
- 最大 Routine Load 作业数，包括 NEED_SCH。EDULED, RUNNING, PAUSE。

max_backup_restore_job_num_per_db

- 默认值：10。
- 此配置用于控制每个 DB 能够记录的 backup/restore 任务的数量。

max_running_txn_num_per_db

- 默认值：100。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 这个配置主要是用来控制同一个 db 的并发导入个数的。
- 当集群中有过多的导入任务正在运行时，新提交的导入任务可能会报错：

```
current running txns on db xxx is xx, larger than limit xx
```

该遇到该错误时，说明当前集群内正在运行的导入任务超过了该配置值。此时建议在业务侧进行等待并重试导入任务。一般来说不推荐增大这个配置值。过高的并发数可能导致系统负载过大。

enable_metric_calculator

- 默认值：true。
- 如果设置为 true，指标收集器将作为守护程序计时器运行，以固定间隔收集指标。

report_queue_size

- 默认值： 100。
- 是否可以动态配置：true。

- 是否为 Master FE 节点独有的配置项：true。
- 这个阈值是为了避免在 FE 中堆积过多的报告任务，可能会导致 OOM 异常等问题。并且每个 BE 每 1 分钟会报告一次 tablet 信息，因此无限制接收报告是不可接受的。以后我们会优化 tablet 报告的处理速度。

> 说明：
> 不建议修改这个值。

partition_rebalance_max_moves_num_per_selection

- 默认值：10。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 仅在使用 PartitionRebalancer 时有效。

partition_rebalance_move_expire_after_access

- 默认值：600 (s)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 仅在使用 PartitionRebalancer 时有效。如果更改，缓存的移动将被清除。

tablet_rebalancer_type

- 默认值：BeLoad。
- 是否为 Master FE 节点独有的配置项：true。
- rebalancer 类型（忽略大小写）：BeLoad、Partition。如果类型解析失败，默认使用 BeLoad。

max_balancing_tablets

- 默认值：100。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果 TabletScheduler 中的 balance tablet 数量超过 max_balancing_tablets，则不再进行 balance 检查。

max_scheduling_tablets

- 默认值：2000。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果 TabletScheduler 中调度的 tablet 数量超过 max_scheduling_tablets， 则跳过检查。

disable_balance

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果设置为 true，TabletScheduler 将不会做 balance。

balance_load_score_threshold

- 默认值：0.1 (10%)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 集群 balance 百分比的阈值，如果一个BE的负载分数比平均分数低10%，这个后端将被标记为低负载，如果负载分数比平均分数高10%，将被标记为高负载。

schedule_slot_num_per_path

- 默认值：2。
- tablet 调度程序中每个路径的默认 slot 数量。

tablet_repair_delay_factor_second

- 默认值：60（s）。
- 是否可以动态配置：true
- 是否为 Master FE 节点独有的配置项：true。
- 决定修复 tablet 前的延迟时间因素：
  i. 如果优先级为 VERY_HIGH，请立即修复。
  ii. HIGH，延迟 tablet_repair_delay_factor_second * 1；。
  iii. 正常：延迟 tablet_repair_delay_factor_second * 2。
  iv. 低：延迟 tablet_repair_delay_factor_second * 3。

es_state_sync_interval_second

- 默认值：10。
- fe 会在每隔 es_state_sync_interval_secs 调用 es api 获取 es 索引分片信息。

disable_hadoop_load

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 默认不禁用，将来不推荐使用 hadoop 集群 load。 设置为 true 以禁用这种 load 方式。

db_used_data_quota_update_interval_secs

- 默认值：300 (s)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 一个主守护线程将每 db_used_data_quota_update_interval_secs 更新数据库 txn 管理器的数据库使用数据配额 。
- 为了更好的数据导入性能，在数据导入之前的数据库已使用的数据量是否超出配额的检查中，我们并不实时计算数据库已经使用的数据量，而是获取后台线程周期性更新的值。
- 该配置用于设置更新数据库使用的数据量的值的时间间隔。

disable_load_job

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 不禁用，如果这设置为 true。
  - 调用开始 txn api 时，所有挂起的加载作业都将失败。
  - 调用 commit txn api 时，所有准备加载作业都将失败 。
  - 所有提交的加载作业将等待发布。

catalog_try_lock_timeout_ms

- 默认值：5000（ms）。
- 是否可以动态配置：true。
- 元数据锁的 tryLock 超时配置。 通常它不需要改变，除非您需要测试一些东西。

max_query_retry_time

- 默认值：2。
- 是否可以动态配置：true。
- 查询重试次数。 如果我们遇到 RPC 异常并且没有将结果发送给用户，则可能会重试查询。 您可以减少此数字以避免雪崩灾难。

remote_fragment_exec_timeout_ms

- 默认值：5000（ms）。
- 是否可以动态配置：true。
- 异步执行远程 fragment 的超时时间。 在正常情况下，异步远程 fragment 将在短时间内执行。 如果系统处于高负载状态，请尝试将此超时设置更长的时间。

enable_local_replica_selection

- 默认值：false。
- 是否可以动态配置：true。
- 如果设置为 true，Planner 将尝试在与此前端相同的主机上选择 tablet 的副本。
  在以下情况下，这可能会减少网络传输：
  i. N 个主机，部署了 N 个 BE 和 N 个 FE。
  ii. 数据有N个副本。
  iii. 高并发查询均匀发送到所有 Frontends
  在这种情况下，所有 Frontends 只能使用本地副本进行查询。

max_unfinished_load_job

默认值：1000。
是否可以动态配置：true。
是否为 Master FE 节点独有的配置项：true。
最大加载任务数，包括 PENDING、ETL、LOADING、QUORUM_FINISHED。 如果超过此数量，则不允许提交加载作业。

max_bytes_per_broker_scanner

默认值：3 * 1024 * 1024 * 1024L（3G）。
是否可以动态配置：true。
是否为 Master FE 节点独有的配置项：true。
broker scanner 程序可以在一个 broker 加载作业中处理的最大字节数。 通常，每个 BE 都有一个 broker scanner 程序。

enable_auth_check

- 默认值：true。
- 如果设置为 false，则身份验证检查将被禁用，以防新权限系统出现问题。

tablet_stat_update_interval_second

- 默认值：300，（5分钟）。
- tablet 状态更新间隔- 所有 FE 将在每个时间间隔从所有 BE 获取 tablet 统计信息。

storage_flood_stage_usage_percent

- 默认值：95（95%）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。

storage_flood_stage_left_capacity_bytes

- 默认值：

- storage_flood_stage_usage_percent : 95 (95%)。
- storage_flood_stage_left_capacity_bytes : 1 * 1024 * 1024 * 1024 (1GB)。

- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果磁盘容量达到 storage_flood_stage_usage_percent 和 storage_flood_stage_left_capacity_bytes 以下操作将被拒绝：
  i. load 作业。
  ii. restore 工作。

storage_high_watermark_usage_percent

默认值：85 (85%)。
是否可以动态配置：true。
是否为 Master FE 节点独有的配置项：true。

storage_min_left_capacity_bytes

默认值： 2 * 1024 * 1024 * 1024 (2GB)。
是否可以动态配置：true。
是否为 Master FE 节点独有的配置项：true。

- storage_high_watermark_usage_percent 限制BE端存储路径的最大容量使用百分比。
  storage_min_left_capacity_bytes限制BE端存储路径的最小剩余容量。 如果达到这两个限制，则不能选择此存储路径作为 tablet 存储目的地。 但是对于 tablet 恢复，我们可能会超过这些限制以尽可能保持数据完整性。

backup_job_default_timeout_ms

- 默认值：86400 * 1000 (1天)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 备份作业的默认超时时间。

with_k8s_certs

- 默认值：false。
- 如果在本地使用 k8s 部署管理器，请将其设置为 true 并准备证书文件。

dpp_hadoop_client_path

- 默认值：/lib/hadoop-client/hadoop/bin/hadoop。

dpp_bytes_per_reduce

- 默认值：100 * 1024 * 1024L; // 100M。

dpp_default_cluster

- 默认值：palo-dpp。

dpp_default_config_str

- 默认值：

```
{
"hadoop_configs : '"
"mapred.job.priority=NORMAL;"
"mapred.job.map.capacity=50;"
"mapred.job.reduce.capacity=50;"
"mapred.hce.replace.streaming=false;"
"abaci.long.stored.job=true;"
"dce.shuffle.enable=false;"
"dfs.client.authserver.force_stop=true;"
"dfs.client.auth.method=0"
"'}
```

dpp_config_str

- 默认值：

```
{palo-dpp : {"
+ "hadoop_palo_path : '/dir',"
+ "hadoop_configs : '"
+ "fs.default.name=hdfs://host:port;"
+ "mapred.job.tracker=host:port;"
+ "hadoop.job.ugi=user,password"
+ "'}"
+ "}
```

enable_deploy_manager

- 默认值：disable。
- 如果使用第三方部署管理器部署 Doris，则设置为 true。
  - 有效的选项是：
  - disable：没有部署管理器。
  - k8s：Kubernetes。
  - ambari：Ambari。

- local：本地文件（用于测试或 Boxer2 BCC 版本）。

enable_token_check

- 默认值：true。
- 为了向前兼容，稍后将被删除。 下载image文件时检查令牌。

expr_depth_limit

- 默认值：3000。
- 是否可以动态配置：true。
- 限制 expr 树的深度。 超过此限制可能会导致在持有 db read lock 时分析时间过长。

expr_children_limit

- 默认值：10000。
- 是否可以动态配置：true。
- 限制 expr 树的 expr 子节点的数量。 超过此限制可能会导致在持有数据库读锁时分析时间过长。

proxy_auth_magic_prefix

- 默认值：x@8。

proxy_auth_enable

- 默认值：false。

meta_publish_timeout_ms

- 默认值：1000ms。
- 默认元数据发布超时时间。

disable_colocate_balance

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 此配置可以设置为 true 以禁用自动 colocate 表的重新定位和平衡。 如果 disable_colocate_balance'设置为 true，则 ColocateTableBalancer 将不会重新定位和平衡并置表。

> ⚠ 注意：
>
> - 一般情况下，根本不需要关闭平衡。
> - 因为一旦关闭平衡，不稳定的 colocate 表可能无法恢复
> - 最终查询时无法使用 colocate 计划。

query_colocate_join_memory_limit_penalty_factor

- 默认值：1
- 是否可以动态配置：true
- colocote join PlanFragment instance 的 memory_limit = exec_mem_limit / min (query_colocate_join_memory_limit_penalty_factor, instance_num)

max_connection_scheduler_threads_num

- 默认值：4096。
- 查询请求调度器中的最大线程数。
- 前的策略是，有请求过来，就为其单独申请一个线程进行服务。

qe_max_connection

- 默认值：1024。
- 每个 FE 的最大连接数。

check_consistency_default_timeout_second

- 默认值：600（10分钟）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 单个一致性检查任务的默认超时。 设置足够长以适合您的 tablet 大小。

consistency_check_start_time

- 默认值：23。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 一致性检查开始时间。
- 一致性检查器将从 consistency_check_start_time 运行到 consistency_check_end_time。 默认为 23:00 至 04:00。

consistency_check_end_time

- 默认值：04。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 一致性检查结束时间。
- 一致性检查器将从 consistency_check_start_time 运行到 consistency_check_end_time。 默认为 23:00 至 04:00。

export_tablet_num_per_task

- 默认值：5。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 每个导出查询计划的 tablet 数量。

export_task_default_timeout_second

- 默认值：2 * 3600（2小时）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 导出作业的默认超时时间。

export_running_job_num_limit

- 默认值：5。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 运行导出作业的并发限制，默认值为 5，0 表示无限制。

export_checker_interval_second

- 默认值：5。
- 导出检查器的运行间隔。

default_load_parallelism

- 默认值：1。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 单个节点broker load导入的默认并发度。
  如果用户在提交broker load任务时，在properties中自行指定了并发度，则采用用户自定义的并发度。
  此参数将与max_broker_concurrency、min_bytes_per_broker_scanner等多个配置共同决定导入任务的并发度。

max_broker_concurrency

- 默认值：10。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- broker scanner 的最大并发数。

min_bytes_per_broker_scanner

- 默认值：67108864L (64M)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 单个 broker scanner 将读取的最小字节数。

catalog_trash_expire_second

- 默认值：86400L (1天)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 删除数据库（表/分区）后，您可以使用 RECOVER stmt 恢复它。 这指定了最大数据保留时间。 一段时间后，数据将被永久删除。

storage_cooldown_second

- 默认值：30 * 24 * 3600L （30天）。
- 创建表（或分区）时，可以指定其存储介质（HDD 或 SSD）。 如果设置为 SSD，这将指定tablet在 SSD 上停留的默认时间。 之后，tablet将自动移动到 HDD。 您可以在 CREATE TABLE stmt 中设置存储冷却时间。

default_storage_medium

- 默认值：HDD。
- 创建表（或分区）时，可以指定其存储介质（HDD 或 SSD）。 如果未设置，则指定创建时的默认介质。

max_backend_down_time_second

- 默认值：3600（1小时）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 如果 BE 关闭了 max_backend_down_time_second，将触发 BACKEND_DOWN 事件。

alter_table_timeout_second

- 默认值：86400（1天）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- ALTER TABLE 请求的最大超时时间。 设置足够长以适合您的表格数据大小 。

capacity_used_percent_high_water

- 默认值：0.75（75%）。
- 是否可以动态配置：true。

- 是否为 Master FE 节点独有的配置项：true。
- 磁盘容量的高水位使用百分比。 这用于计算后端的负载分数。

clone_distribution_balance_threshold

- 默认值：0.2 。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- BE副本数的平衡阈值。

clone_capacity_balance_threshold

- 默认值：0.2 。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- BE 中数据大小的平衡阈值。
- 平衡算法为：
  - a. 计算整个集群的平均使用容量（AUC）（总数据大小/BE数）。
  - b. 高水位为(AUC * (1 + clone_capacity_balance_threshold))。
  - c. 低水位为(AUC * (1 − clone_capacity_balance_threshold))。
- 克隆检查器将尝试将副本从高水位 BE 移动到低水位 BE。

replica_delay_recovery_second

- 默认值：0。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 副本之间的最小延迟秒数失败，并且尝试使用克隆来恢复它。

clone_high_priority_delay_second

- 默认值：0。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 高优先级克隆作业的延迟触发时间。

clone_normal_priority_delay_second

- 默认值：300（5分钟）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 正常优先级克隆作业的延迟触发时间。

clone_low_priority_delay_second

- 默认值：600（10分钟）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 低优先级克隆作业的延迟触发时间。 克隆作业包含需要克隆（恢复或迁移）的tablet。 如果优先级为 LOW，则会延迟 clone_low_priority_delay_second，在作业创建之后然后被执行。 这是为了避免仅因为主机短时间停机而同时运行大量克隆作业。
- 注意这个配置（还有 clone_normal_priority_delay_second ）如果它小于 clone_checker_interval_second 将不起作用。

clone_max_job_num

- 默认值：100。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 低优先级克隆作业的并发数。 高优先级克隆作业的并发性目前是无限的。

clone_job_timeout_second

- 默认值：7200 (2小时)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 单个克隆作业的默认超时。 设置足够长以适合您的副本大小。 副本数据越大，完成克隆所需的时间就越多。

clone_checker_interval_second

- 默认值：300（5分钟）。
- 克隆检查器的运行间隔。

tablet_delete_timeout_second

- 默认值：2。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 与 tablet_create_timeout_second 含义相同，但在删除 tablet 时使用。

async_loading_load_task_pool_size

- 默认值：10。
- 是否可以动态配置：false。
- 是否为 Master FE 节点独有的配置项：true。
- loading_load任务执行程序池大小。 该池大小限制了正在运行的最大 loading_load任务数。

- 当前，它仅限制 broker load的 loading_load任务的数量。

async_pending_load_task_pool_size

- 默认值：10。
- 是否可以动态配置：false。
- 是否为 Master FE 节点独有的配置项：true。
- pending_load任务执行程序池大小。 该池大小限制了正在运行的最大 pending_load任务数。
- 当前，它仅限制 broker load和 spark load的 pending_load任务的数量。
- 它应该小于 max_running_txn_num_per_db的值。

async_load_task_pool_size

- 默认值：10。
- 是否可以动态配置：false。
- 是否为 Master FE 节点独有的配置项：true。
- 此配置只是为了兼容旧版本，此配置已被 async_loading_load_task_pool_size取代，以后会被移除。

disable_show_stream_load

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 是否禁用显示 stream load 并清除内存中的 stream load 记录。

max_stream_load_record_size

- 默认值：5000。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 可以存储在内存中的最近 stream load 记录的默认最大数量。

fetch_stream_load_record_interval_second

- 默认值：120
- 是否可以动态配置：true
- 是否为 Master FE 节点独有的配置项：true
- 获取 stream load 记录间隔

desired_max_waiting_jobs

- 默认值：100
- 是否可以动态配置：true

- 是否为 Master FE 节点独有的配置项：true
- routine load V2 版本加载的默认等待作业数 ，这是一个理想的数字。 在某些情况下，例如切换 master，当前数量可能超过desired_max_waiting_jobs

yarn_config_dir

默认值：PaloFe.DORIS_HOME_DIR + "/lib/yarn-config"。
默认的 yarn 配置文件目录每次运行 yarn 命令之前，我们需要检查一下这个路径下是否存在 config 文件，如果不存在，则创建它们。

yarn_client_path

- 默认值：PaloFe.DORIS_HOME_DIR + "/lib/yarn-client/hadoop/bin/yarn"。
- 默认 Yarn 客户端路径。

spark_launcher_log_dir

- 默认值： sys_log_dir + "/spark_launcher_log"。
- 指定的 spark 启动器日志目录。

spark_resource_path

- 默认值：空。
- 默认值的 Spark 依赖路径。

spark_home_default_dir

- 默认值：PaloFe.DORIS_HOME_DIR + "/lib/spark2x"。
- 默认的 Spark home 路径。

spark_load_default_timeout_second

- 默认值：86400 (1天)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 默认 Spark 加载超时时间。

spark_dpp_version

- 默认值：1.0.0。
- Spark 默认版本号。

hadoop_load_default_timeout_second

- 默认值：86400 * 3 (3天)。

- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- Hadoop 加载超时时间。

min_load_timeout_second

- 默认值：1（1秒）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- mini load 超时时间，适用于所有类型的加载。

max_stream_load_timeout_second

- 默认值：259200（3天）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- stream load 和 mini load 最大超时时间。

max_load_timeout_second

- 默认值：259200（3天）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- load 最大超时时间，适用于除 stream load 之外的所有类型的加载。

stream_load_default_timeout_second

- 默认值：600（s）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 默认 stream load 和 mini load 超时时间。

insert_load_default_timeout_second

- 默认值：3600（1小时）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 默认 insert load 超时时间。

mini_load_default_timeout_second

- 默认值：3600（1小时）。
- 是否可以动态配置：true。

- 是否为 Master FE 节点独有的配置项：true。
- 默认非 stream load 类型的 mini load 的超时时间。

broker_load_default_timeout_second

- 默认值：14400（4小时）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- Broker load 的默认超时时间。

load_running_job_num_limit

- 默认值：0。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- Load 任务数量限制，默认0，无限制。

load_input_size_limit_gb

- 默认值：0。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- Load 作业输入的数据大小，默认是0，无限制。

delete_thread_num

- 默认值：10。
- 删除作业的并发线程数。

load_etl_thread_num_normal_priority

- 默认值：10。
- NORMAL 优先级 etl 加载作业的并发数。

load_etl_thread_num_high_priority

- 默认值：3
- 高优先级 etl 加载作业的并发数。

load_pending_thread_num_normal_priority

- 默认值：10。
- NORMAL 优先级挂起加载作业的并发数。

load_pending_thread_num_high_priority

- 默认值：3。
- 高优先级挂起加载作业的并发数。 加载作业优先级定义为 HIGH 或 NORMAL。 所有小批量加载作业都是 HIGH 优先级，其他类型的加载作业是 NORMAL 优先级。 设置优先级是为了避免慢加载作业长时间占用线程。 这只是内部优化的调度策略。 目前，您无法手动指定作业优先级。

load_checker_interval_second

- 默认值：5（s）。
- 负载调度器运行间隔。 加载作业将其状态从 PENDING 转移到 LOADING 到 FINISHED。 加载调度程序将加载作业从 PENDING 转移到 LOADING 而 txn 回调会将加载作业从 LOADING 转移到 FINISHED。 因此，当并发未达到上限时，加载作业最多需要一个时间间隔才能完成。

max_layout_length_per_row

- 默认值：100000。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 一行的最大内存布局长度。 默认为 100 KB。
  在 BE 中，RowBlock 的最大大小为 100MB（在 be.conf 中配置为 max_unpacked_row_block_size）。
- 每个 RowBlock 包含 1024 行。 因此，一行的最大大小约为 100 KB。
- 例如：schema：k1(int), v1(decimal), v2(varchar(2000))
  那么一行的内存布局长度为：8(int) + 40(decimal) + 2000(varchar) = 2048 (Bytes)。
  查看所有类型的内存布局长度，在 mysql-client 中运行 help create table。
  如果要增加此数字以支持一行中的更多列，则还需要增加 be.conf 中的 max_unpacked_row_block_size，但性能影响未知。

load_straggler_wait_second

- 默认值：300。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 负载中落后节点的最大等待秒数。
  例如：有 3 个副本 A, B, C load 已经在 t1 时仲裁完成 (A,B) 并且 C 没有完成，如果 (current_time-t1)> 300s，那么 doris会将 C 视为故障节点，将调用事务管理器提交事务并告诉事务管理器 C 失败。
- 也用于等待发布任务时。

> ⚠ 注意：
> 这个参数是所有作业的默认值，DBA 可以为单独的作业指定它。

thrift_server_type

- 该配置表示FE的Thrift服务使用的服务模型, 类型为string, 大小写不敏感。
- 若该参数为 SIMPLE, 则使用 TSimpleServer 模型, 该模型一般不适用于生产环境，仅限于测试使用。
- 若该参数为 THREADED, 则使用 TThreadedSelectorServer 模型，该模型为非阻塞式I/O模型，即主从 Reactor 模型，该模型能及时响应大量的并发连接请求，在多数场景下有较好的表现。
- 若该参数为 THREAD_POOL, 则使用 TThreadPoolServer 模型，该模型为阻塞式I/O模型，使用线程池处理用户连接，并发连接数受限于线程池的数量，如果能提前预估并发请求的数量，并且能容忍足够多的线程资源开销，该模型会有较好的性能表现，默认使用该服务模型。

thrift_server_max_worker_threads

- 默认值：4096。
- Thrift Server最大工作线程数。

publish_version_interval_ms

- 默认值：10（ms）。
- 两个发布版本操作之间的最小间隔。

publish_version_timeout_second

- 默认值：30（s）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 一个事务的所有发布版本任务完成的最大等待时间。

max_create_table_timeout_second

- 默认值：60（s）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 为了在创建表（索引）不等待太久，设置一个最大超时时间。

tablet_create_timeout_second

- 默认值：1（s）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 创建单个副本的最长等待时间。
- 例如：如果您为每个表创建一个包含 m 个 tablet 和 n 个副本的表，创建表请求将在超时前最多运行 (m * n * tablet_create_timeout_second)。

max_mysql_service_task_threads_num

- 默认值：4096。
- mysql 中处理任务的最大线程数。

cluster_id

- 默认值：-1。
- 如果节点（FE 或 BE）具有相同的集群 id，则将认为它们属于同一个Doris 集群。 Cluster id 通常是主 FE 首次启动时生成的随机整数。 您也可以指定一个。

auth_token

- 默认值：空
- 用于内部身份验证的集群令牌。

cluster_name

- 默认值： Apache doris。
  集群名称将显示为网页标题。

mysql_service_io_threads_num

- 默认值：4。
- mysql 中处理 io 事件的线程数。

mysql_service_nio_enabled

- 默认值：true。
- mysql 服务 nio 选项是否启用，默认启用。

query_port

- 默认值：9030。
- Doris FE 通过 mysql 协议查询连接端口。

rewrite_count_distinct_to_bitmap_hll

- 默认值：true。
- 该变量为 session variable，session 级别生效。
- 类型：boolean。
- 描述：**仅对于 AGG 模型的表来说**，当变量为 true 时，用户查询时包含 count(distinct c1) 这类聚合函数时，如果 c1 列本身类型为 bitmap，则 count distnct 会改写为 bitmap_union_count(c1)。 当 c1 列本身类型为 hll，则 count distinct 会改写为 hll_union_agg(c1) 如果变量为 false，则不发生任何改写。

rpc_port

- 默认值：9020。
- FE Thrift Server 的端口。

thrift_backlog_num

- 默认值：1024。
- thrift 服务器的 backlog_num 当您扩大这个 backlog_num 时，您应该确保它的值大于 linux /proc/sys/net/core/somaxconn 配置。

thrift_client_timeout_ms

- 默认值：0。
- thrift 服务器的连接超时和套接字超时配置 thrift_client_timeout_ms 的默认值设置为零以防止读取超时。

mysql_nio_backlog_num

- 默认值：1024。
- mysql nio server 的 backlog_num 当您放大这个 backlog_num 时，您应该同时放大 linux /proc/sys/net/core/somaxconn文件中的值。

http_backlog_num

- 默认值：1024。
- netty http server 的 backlog_num 当您放大这个 backlog_num 时，您应该同时放大 linux /proc/sys/net/core/somaxconn文件中的值。

http_max_line_length

- 默认值：4096。
- HTTP 服务允许接收请求的 URL 的最大长度，单位为比特。

http_max_header_size

- 默认值：8192。
- HTTP 服务允许接收请求的 Header 的最大长度，单位为比特。

http_max_chunk_size

- 默认值：8192。
- http 上下文 chunk 块的最大尺寸。

http_port

- 默认值：8030。
- FE http 端口，当前所有 FE http 端口都必须相同。

max_bdbje_clock_delta_ms

- 默认值：5000（5秒）。
- 设置非主 FE 到主 FE 主机之间的最大可接受时钟偏差。每当非主 FE 通过 BDBJE 建立到主 FE 的连接时，都会检查该值。如果时钟偏差大于此值，则放弃连接。

ignore_meta_check

- 默认值：false。
- 是否可以动态配置：true。
- 如果为 true，非主 FE 将忽略主 FE 与其自身之间的元数据延迟间隙，即使元数据延迟间隙超过 meta_delay_toleration_second。非主 FE 仍将提供读取服务。当您出于某种原因尝试停止 Master FE 较长时间，但仍希望非 Master FE 可以提供读取服务时，这会很有帮助。

metadata_failure_recovery

- 默认值：false。
- 如果为 true，FE 将重置 bdbje 复制组（即删除所有可选节点信息）并应该作为 Master 启动。如果所有可选节点都无法启动，我们可以将元数据复制到另一个节点并将此配置设置为 true 以尝试重新启动 FE。

priority_networks

- 默认值：空。
- 为那些有很多 ip 的服务器声明一个选择策略。请注意，最多应该有一个 ip 与此列表匹配。这是一个以分号分隔格式的列表，用 CIDR 表示法，例如 10.10.10.0/24 ， 如果没有匹配这条规则的ip，会随机选择一个。

txn_rollback_limit

- 默认值：100。
- 尝试重新加入组时 bdbje 可以回滚的最大 txn 数。

max_agent_task_threads_num

- 默认值：4096
- 是否为 Master FE 节点独有的配置项：true。
- 代理任务线程池中处理代理任务的最大线程数。

heartbeat_mgr_blocking_queue_size

- 默认值：1024
- 是否为 Master FE 节点独有的配置项：true。

- 在 heartbeat_mgr 中存储心跳任务的阻塞队列大小。

heartbeat_mgr_threads_num

- 默认值：8。
- 是否为 Master FE 节点独有的配置项：true。
- heartbeat _mgr 中处理心跳事件的线程数。

bdbje_replica_ack_timeout_second

- 默认值：10 。
- 元数据会同步写入到多个 Follower FE，这个参数用于控制 Master FE 等待 Follower FE 发送 ack 的超时时间。当写入的数据较大时，可能 ack 时间较长，如果超时，会导致写元数据失败，FE 进程退出。此时可以适当调大这个参数。

bdbje_lock_timeout_second

- 默认值：1。
- bdbje 操作的 lock timeout 如果 FE WARN 日志中有很多 LockTimeoutException，可以尝试增加这个值。

bdbje_heartbeat_timeout_second

- 默认值：30。
- master 和 follower 之间 bdbje 的心跳超时。 默认为 30 秒，与 bdbje 中的默认值相同。 如果网络遇到暂时性问题，一些意外的长 Java GC 使您烦恼，您可以尝试增加此值以减少错误超时的机会。

replica_ack_policy

- 默认值：SIMPLE_MAJORITY。
- 选项：ALL, NONE, SIMPLE_MAJORITY。
- bdbje 的副本 ack 策略。更多信息，请参见 官方资料。

replica_sync_policy

- 默认值：SYNC。
- 选项：SYNC, NO_SYNC, WRITE_NO_SYNC。
- bdbje 的 Follower FE 同步策略。

master_sync_policy

- 默认值：SYNC。
- 选项：SYNC, NO_SYNC, WRITE_NO_SYNC。

- Master FE 的 bdbje 同步策略。 如果您只部署一个 Follower FE，请将其设置为“SYNC”。 如果您部署了超过 3 个 Follower FE，您可以将这个和下面的 replica_sync_policy 设置为 WRITE_NO_SYNC。 更多信息，请参见 官网资料。

meta_delay_toleration_second

- 默认值：300（5分钟）。
- 如果元数据延迟间隔超过 meta_delay_toleration_second，非主 FE 将停止提供服务。

edit_log_roll_num

- 默认值：50000。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- Master FE will save image every edit_log_roll_num meta journals.。

edit_log_port

- 默认值：9010。
- bdbje端口。

edit_log_type

- 默认值：BDB。
- 编辑日志类型。
  - BDB：将日志写入 bdbje。
  - LOCAL：已弃用。

tmp_dir

- 默认值：PaloFe.DORIS_HOME_DIR + "/temp_dir"。
- temp dir 用于保存某些过程的中间结果，例如备份和恢复过程。 这些过程完成后，将清除此目录中的文件。

meta_dir

- 默认值：PaloFe.DORIS_HOME_DIR + "/doris-meta"
- Doris 元数据将保存在这里。 强烈建议将此目录的存储为：
  i. 高写入性能（SSD）。
  ii. 安全（RAID）。

custom_config_dir

- 默认值：PaloFe.DORIS_HOME_DIR + "/conf"。
- 自定义配置文件目录。

- 配置 fe_custom.conf 文件的位置。默认为 conf/ 目录下。
- 在某些部署环境下，conf/ 目录可能因为系统的版本升级被覆盖掉。这会导致用户在运行是持久化修改的配置项也被覆盖。这时，我们可以将 fe_custom.conf 存储在另一个指定的目录中，以防止配置文件被覆盖。

log_roll_size_mb

- 默认值：1024（1G）。
- 一个系统日志和审计日志的最大大小。

sys_log_dir

- 默认值：PaloFe.DORIS_HOME_DIR + "/log"。
- sys_log_dir: 这指定了 FE 日志目录。 FE 将产生 2 个日志文件：
  i. fe.log：FE进程的所有日志。
  ii. fe.warn.log FE 进程的所有警告和错误日志。

sys_log_level

- 默认值：INFO。
- 日志级别，可选项：INFO, WARNING, ERROR, FATAL。

sys_log_roll_num

- 默认值：10。
- 要保存在 sys_log_roll_interval 内的最大 FE 日志文件。 默认为 10，表示一天最多有 10 个日志文件。

sys_log_verbose_modules

- 默认值：{}。
- sys_log_verbose_modules：详细模块。 VERBOSE 级别由 log4j DEBUG 级别实现。
- 例如：sys_log_verbose_modules = org.apache.doris.catalog，这只会打印包 org.apache.doris.catalog 及其所有子包中文件的调试日志。

sys_log_roll_interval

- 默认值：DAY。
- sys_log_roll_interval:
  - DAY: log 前缀是 yyyyMMdd。
  - HOUR: log 前缀是 yyyyMMddHH。

sys_log_delete_age

- 默认值：7d。
- sys_log_delete_age：默认为 7 天，如果日志的最后修改时间为 7 天前，则将其删除。

- 支持格式：7d：7天，10小时：10 小时，60m：60分钟，120s：120秒。

audit_log_dir

- 默认值：PaloFe.DORIS_HOME_DIR + "/log"。
- 审计日志目录：
  - 这指定了 FE 审计日志目录。
  - 审计日志 fe.audit.log 包含所有请求以及相关信息，如 user, host, cost, status 等。

audit_log_roll_num

- 默认值：90。
- 保留在 audit_log_roll_interval 内的最大 FE 审计日志文件。

audit_log_modules

- 默认值：{"slow_query", "query", "load", "stream_load"}。
- 慢查询包含所有开销超过 *qe_slow_log_ms* 的查询。

qe_slow_log_ms

- 默认值：5000（5秒）。
- 如果查询的响应时间超过此阈值，则会在审计日志中记录为 slow_query。

audit_log_roll_interval

- 默认值：DAY。
- DAY: log 前缀是：yyyyMMdd。
- HOUR: log 前缀是：yyyyMMddHH。

audit_log_delete_age

- 默认值：30d。
- 默认为 30 天，如果日志的最后修改时间为 30 天前，则将其删除。
- 支持格式：7d：7天，10小时：10 小时，60m：60分钟，120s：120秒。

plugin_dir

- 默认值：DORIS_HOME + "/plugins。
- 插件安装目录。

plugin_enable

- 默认值：true。
- 是否可以动态配置：true。

- 是否为 Master FE 节点独有的配置项：true。
- 插件是否启用，默认启用。

label_keep_max_second

- 默认值：3 * 24 * 3600 (3天)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- label_keep_max_second后将删除已完成或取消的加载作业的标签。
  i. 去除的标签可以重复使用。
  ii. 设置较短的时间会降低 FE 内存使用量（因为所有加载作业的信息在被删除之前都保存在内存中）。
- 在高并发写的情况下，如果出现大量作业积压，出现 call frontend service failed的情况，查看日志如果是元数据写占用锁的时间太长，可以将这个值调成12小时，或者更小6小时。

streaming_label_keep_max_second

- 默认值：43200（12小时）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 对于一些高频负载工作，例如：INSERT、STREAMING LOAD、ROUTINE_LOAD_TASK 。如果过期，则删除已完成的作业或任务。

history_job_keep_max_second

- 默认值：7 * 24 * 3600（7天）。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：true。
- 某些作业的最大保留时间。像 schema 更改和 Rollup 作业。

label_clean_interval_second

- 默认值：4 * 3600（4小时）。
- load 标签清理器将每隔 label_clean_interval_second 运行一次以清理过时的作业。

delete_info_keep_max_second

- 默认值：3 * 24 * 3600 (3天)。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：false。
- 删除元数据中创建时间大于delete_info_keep_max_second的delete信息。
- 设置较短的时间将减少 FE 内存使用量和镜像文件大小。（因为所有的deleteInfo在被删除之前都存储在内存和镜像文件中）。

transaction_clean_interval_second

- 默认值：30。
- 如果事务 visible 或者 aborted 状态，事务将在 transaction_clean_interval_second 秒后被清除 ，我们应该让这个间隔尽可能短，每个清洁周期都尽快。

default_max_query_instances

- 默认值：-1。
- 用户属性max_query_instances小于等于0时，使用该配置，用来限制单个用户同一时刻可使用的查询instance个数。该参数小于等于0表示无限制。

use_compact_thrift_rpc

- 默认值：true。
- 是否使用压缩格式发送查询计划结构体。开启后，可以降低约50%的查询计划结构体大小，从而避免一些 "send fragment timeout" 错误。但是在某些高并发小查询场景下，可能会降低约10%的并发度。

disable_tablet_scheduler

- 默认值：false。
- 是否可以动态配置：true。
- 是否为 Master FE 节点独有的配置项：false。
- 如果设置为true，将关闭副本修复和均衡逻辑。

BE 配置参数

最近更新时间：2022-04-13 15:29:23

BE 的配置文件 be.conf 通常存放在 BE 部署路径的 conf/ 目录下。 而在 0.14 版本中会引入另一个配置文件 be_custom.conf 。该配置文件用于记录用户在运行时动态配置并持久化的配置项。本文档主要介绍 BE 的相关配置项。

BE 进程启动后，会先读取 be.conf 中的配置项，之后再读取 be_custom.conf 中的配置项。be_custom.conf 中的配置项会覆盖 be.conf 中相同的配置项。

查看配置项

用户可以通过访问 BE 的 Web 页面查看当前配置项： http://be_host:be_webserver_port/varz

设置配置项

BE 的配置项有两种方式进行配置：

1. 静态配置

   在 conf/be.conf 文件中添加和设置配置项。 be.conf 中的配置项会在 BE 进行启动时被读取。没有在 be.conf 中的配置项将使用默认值。

2. 动态配置

   BE 启动后，可以通过以下命令动态设置配置项。

   ```
   curl -X POST http://{be_ip}:{be_http_port}/api/update_config?{key}={value}'
   ```

   在 0.13 版本及之前，通过该方式修改的配置项将在 BE 进程重启后失效。在 0.14 及之后版本中，可以通过以下命令持久化修改后的配置。修改后的配置项存储在 be_custom.conf 文件中。

   ```
   curl -X POST http://{be_ip}:{be_http_port}/api/update_config?{key}={value}&persist=true'
   ```

应用举例

1. 静态方式修改 max_compaction_concurrency 。

   通过在 be.conf 文件中添加： max_compaction_concurrency=5 ，之后重启 BE 进程以生效该配置。

2. 动态方式修改 streaming_load_max_mb

   BE 启动后，通过下面命令动态设置配置项 streaming_load_max_mb：

curl -X POST http://{be_ip}:{be_http_port}/api/update_config?streaming_load_max_mb=1024

返回值如下，则说明设置成功。

```
{
"status": "OK",
"msg": ""
}
```

BE 重启后该配置将失效。如果想持久化修改结果，使用如下命令：

```
curl -X POST http://{be_ip}:{be_http_port}/api/update_config?streaming_load_max_mb=1024
\&persist=true
```

配置项列表

alter_tablet_worker_count

- 默认值：3。
- 进行schema change的线程数。

base_compaction_check_interval_seconds

- 默认值：60（s）。
- BaseCompaction线程轮询的间隔。

base_compaction_interval_seconds_since_last_operation

- 默认值：86400。
- BaseCompaction触发条件之一：上一次BaseCompaction距今的间隔。

base_compaction_num_cumulative_deltas

- 默认值：5。
- BaseCompaction触发条件之一：Cumulative文件数目要达到的限制，达到这个限制之后会触发BaseCompaction。

base_compaction_num_threads_per_disk

- 默认值：1。
- 每个磁盘执行BaseCompaction任务的线程数目。

base_compaction_write_mbytes_per_sec

- 默认值：5（MB）。
- BaseCompaction任务每秒写磁盘最大速度。

base_cumulative_delta_ratio

- 默认值：0.3（30%）。
- BaseCompaction触发条件之一：Cumulative文件大小达到Base文件的比例。

base_compaction_trace_threshold

- 类型：int32。
- 描述：打印base compaction的trace信息的阈值，单位秒。
- 默认值：10。

base compaction是一个耗时较长的后台操作，为了跟踪其运行信息，可以调整这个阈值参数来控制trace日志的打印。打印信息如下：

```
W0610 11:26:33.804431 56452 storage_engine.cpp:552] Trace:
0610 11:23:03.727535 (+ 0us) storage_engine.cpp:554] start to perform base compaction
0610 11:23:03.728961 (+ 1426us) storage_engine.cpp:560] found best tablet 546859
0610 11:23:03.728963 (+ 2us) base_compaction.cpp:40] got base compaction lock
0610 11:23:03.729029 (+ 66us) base_compaction.cpp:44] rowsets picked
0610 11:24:51.784439 (+108055410us) compaction.cpp:46] got concurrency lock and start to d
o compaction
0610 11:24:51.784818 (+ 379us) compaction.cpp:74] prepare finished
0610 11:26:33.359265 (+101574447us) compaction.cpp:87] merge rowsets finished
0610 11:26:33.484481 (+125216us) compaction.cpp:102] output rowset built
0610 11:26:33.484482 (+ 1us) compaction.cpp:106] check correctness finished
0610 11:26:33.513197 (+ 28715us) compaction.cpp:110] modify rowsets finished
0610 11:26:33.513300 (+ 103us) base_compaction.cpp:49] compaction finished
0610 11:26:33.513441 (+ 141us) base_compaction.cpp:56] unused rowsets have been moved to
GC queue
Metrics: {"filtered_rows":0,"input_row_num":3346807,"input_rowsets_count":42,"input_rowsets_d
ata_size":1256413170,"input_segments_num":44,"merge_rowsets_latency_us":101574444,"merg
ed_rows":0,"output_row_num":3346807,"output_rowset_data_size":1228439659,"output_segmen
ts_num":6}
```

be_port

- 类型：int32。

- 描述：BE 上 thrift server 的端口号，用于接收来自 FE 的请求。
- 默认值：9060。

be_service_threads

- 类型：int32。
- 描述：BE 上 thrift server service的执行线程数，代表可以用于执行FE请求的线程数。
- 默认值：64。

brpc_max_body_size

这个配置主要用来修改 brpc 的参数 max_body_size 。
有时查询失败，在 BE 日志中会出现 body_size is too large 的错误信息。这可能发生在 SQL 模式为 multi distinct + 无 group by + 超过1T 数据量的情况下。这个错误表示 brpc 的包大小超过了配置值。此时可以通过调大该配置避免这个错误。

brpc_socket_max_unwritten_bytes

这个配置主要用来修改 brpc 的参数 socket_max_unwritten_bytes 。
有时查询失败，BE 日志中会出现 The server is overcrowded 的错误信息，表示连接上有过多的未发送数据。当查询需要发送较大的 bitmap 字段时，可能会遇到该问题，此时可能通过调大该配置避免该错误。

brpc_num_threads

该配置主要用来修改 brpc 中 bthreads 的数量. 该配置的默认值被设置为-1, 这意味着 bthreads 的数量将被设置为机器的 CPU 核数。
用户可以将该配置的值调大来获取更好的QPS性能。更多的信息可以参考 示例。

brpc_port

- 类型：int32。
- 描述：BE 上的 brpc 的端口，用于 BE 之间通讯。
- 默认值：8060。

buffer_pool_clean_pages_limit

- 默认值：20G。
- 清理可能被缓冲池保存的 Page。

buffer_pool_limit

- 类型：string。
- 描述：buffer pool 之中最大的可分配内存。
- 默认值：80G。

BE 缓存池最大的内存可用量，buffer pool 是 BE 新的内存管理结构，通过 buffer page 来进行内存管理，并能够实现数据的落盘。并发的所有查询的内存申请都会通过 buffer pool 来申请。当前 buffer pool 仅作用在 AggregationNode 与 ExchangeNode。

check_auto_compaction_interval_seconds

- 类型：int32。
- 描述：当自动执行compaction的功能关闭时，检查自动compaction开关是否被开启的时间间隔。
- 默认值：5。

check_consistency_worker_count

- 默认值：1。
- 计算tablet的校验和(checksum)的工作线程数。

chunk_reserved_bytes_limit

- 默认值：2147483648。
- Chunk Allocator的reserved bytes限制，默认为2GB，增加这个变量可以提高性能，但是会获得更多其他模块无法使用的空闲内存。

clear_transaction_task_worker_count

- 默认值：1。
- 用于清理事务的线程数。

clone_worker_count

- 默认值：3。
- 用于执行克隆任务的线程数。

cluster_id

- 类型：int32。
- 描述：配置 BE 的所属于的集群 id。
- 默认值：-1。

该值通常由 FE 通过心跳向 BE 下发，不需要额外进行配置。当确认某 BE 属于某一个确定的 Drois 集群时，可以进行配置，同时需要修改数据目录下的 cluster_id 文件，使二者相同。

column_dictionary_key_ratio_threshold

- 默认值：0。
- 字符串类型的取值比例，小于这个比例采用字典压缩算法。

column_dictionary_key_size_threshold

- 默认值：0。
- 字典压缩列大小，小于这个值采用字典压缩算法。

compaction_tablet_compaction_score_factor

- 类型：int32。
- 描述：选择tablet进行compaction时，计算 tablet score 的公式中 compaction score的权重。
- 默认值：1。

compaction_tablet_scan_frequency_factor

- 类型：int32。
- 描述：选择tablet进行compaction时，计算 tablet score 的公式中 tablet scan frequency 的权重。
- 默认值：0。

选择一个 tablet 执行 compaction 任务时，可以将 tablet 的 scan 频率作为一个选择依据，对当前最近一段时间频繁 scan 的 tablet 优先执行 compaction。
tablet score 可以通过以下公式计算：
tablet_score = compaction_tablet_scan_frequency_factor * tablet_scan_frequency + compaction_tablet_compaction_score_factor * compaction_score

compaction_task_num_per_disk

- 类型：int32。
- 描述：每个磁盘可以并发执行的 compaction 任务数量。
- 默认值：2。

compress_rowbatches

- 类型：bool。
- 描述：序列化RowBatch时是否使用Snappy压缩算法进行数据压缩。
- 默认值：true。

create_tablet_worker_count

- 默认值：3。
- BE创建tablet的工作线程数。

cumulative_compaction_rounds_for_each_base_compaction_round

- 类型：int32。

- 描述：Compaction任务的生产者每次连续生产多少轮cumulative compaction任务后生产一轮base compaction。
- 默认值：9。

disable_auto_compaction

- 类型：bool。
- 描述：关闭自动执行compaction任务。
- 默认值：false。

一般需要为关闭状态，当调试或测试环境中想要手动操作compaction任务时，可以对该配置进行开启。

cumulative_compaction_budgeted_bytes

- 默认值：104857600。
- BaseCompaction触发条件之一：Singleton文件大小限制，100MB。

cumulative_compaction_check_interval_seconds

- 默认值：10（s）。
- CumulativeCompaction线程轮询的间隔。

cumulative_compaction_num_threads_per_disk

- 默认值：1。
- 每个磁盘执行CumulativeCompaction线程数。

cumulative_compaction_skip_window_seconds

- 默认值：30（s）。
- CumulativeCompaction会跳过最近发布的增量，以防止压缩可能被查询的版本（以防查询计划阶段花费一些时间）。改参数是设置跳过的窗口时间大小。

cumulative_compaction_trace_threshold

- 类型：int32。
- 描述：打印cumulative compaction的trace信息的阈值，单位秒。
- 默认值：2。

与 base_compaction_trace_threshold 类似。

disable_compaction_trace_log

- 类型: bool。
- 描述: 关闭compaction的trace日志。

- 默认值: true。

如果设置为true，cumulative_compaction_trace_threshold 和 base_compaction_trace_threshold 将不起作用。并且trace日志将关闭。

cumulative_compaction_policy

- 类型：string。
- 描述：配置 cumulative compaction 阶段的合并策略，目前实现了两种合并策略，num_based 和 size_based。
- 默认值：size_based。

详细说明，ordinary 是最初版本的 cumulative compaction 合并策略，做一次 cumulative compaction 之后直接 base compaction 流程。size_based，通用策略是 ordinary 策略的优化版本，仅当 rowset 的磁盘体积在相同数量级时才进行版本合并。合并之后满足条件的 rowset 进行晋升到 base compaction 阶段。能够做到在大量小批量导入的情况下：降低 base compact 的写入放大率，并在读取放大率和空间放大率之间进行权衡，同时减少了文件版本的数据。

cumulative_size_based_promotion_size_mbytes

- 类型：int64。
- 描述：在 size_based 策略下，cumulative compaction 的输出 rowset 总磁盘大小超过了此配置大小，该 rowset 将用于 base compaction。单位是m字节。
- 默认值：1024。

一般情况下，配置在2G 以内，为了防止 cumulative compaction 时间过长，导致版本积压。

cumulative_size_based_promotion_ratio

- 类型：double。
- 描述：在 size_based 策略下，cumulative compaction 的输出 rowset 总磁盘大小超过 base 版本 rowset 的配置比例时，该 rowset 将用于 base compaction。
- 默认值：0.05。

一般情况下，建议配置不要高于0.1，低于0.02。

cumulative_size_based_promotion_min_size_mbytes

- 类型：int64。
- 描述：在size_based策略下，cumulative compaction的输出rowset总磁盘大小低于此配置大小，该rowset将不进行base compaction，仍然处于cumulative compaction流程中。单位是m字节。
- 默认值：64。

一般情况下，配置在512m 以内，配置过大会导致 base 版本早期的大小过小，一直不进行 base compaction。

cumulative_size_based_compaction_lower_size_mbytes

- 类型：int64。
- 描述：在 size_based 策略下，cumulative compaction 进行合并时，选出的要进行合并的 rowset 的总磁盘大小大于此配置时，才按级别策略划分合并。小于这个配置时，直接执行合并。单位是m字节。
- 默认值：64。

一般情况下，配置在128m 以内，配置过大会导致 cumulative compaction 写放大较多。

custom_config_dir

配置 be_custom.conf 文件的位置。默认为 conf/ 目录下。
在某些部署环境下， conf/ 目录可能因为系统的版本升级被覆盖掉。这会导致用户在运行是持久化修改的配置项也被覆盖。这时，我们可以将 be_custom.conf 存储在另一个指定的目录中，以防止配置文件被覆盖。

default_num_rows_per_column_file_block

- 类型：int32。
- 描述：配置单个 RowBlock 之中包含多少行的数据。
- 默认值：1024。

default_rowset_type

- 类型：string。
- 描述：标识 BE 默认选择的存储格式，可配置的参数为："ALPHA", "BETA"。主要起以下两个作用：
  i. 当建表的 storage_format 设置为 Default 时，通过该配置来选取 BE 的存储格式。
  ii. 进行 Compaction 时选择 BE 的存储格式。
- 默认值：BETA。

delete_worker_count

- 默认值：3。
- 执行数据删除任务的线程数。

disable_mem_pools

- 默认值：false。
- 是否禁用内存缓存池，默认不禁用。

disable_storage_page_cache

- 类型：bool。
- 描述：是否进行使用 page cache 进行 index 的缓存，该配置仅在 BETA 存储格式时生效。

- 默认值：false。

disk_stat_monitor_interval

- 默认值：5（s）。
- 磁盘状态检查时间间隔。

doris_cgroups

- 默认值：空。
- 分配给 doris 的 cgroups。

doris_max_pushdown_conjuncts_return_rate

- 类型：int32。
- 描述：BE 在进行 HashJoin 时，会采取动态分区裁剪的方式将 join 条件下推到 OlapScanner 上。当 OlapScanner 扫描的数据大于32768行时，BE 会进行过滤条件检查，如果该过滤条件的过滤率低于该配置，则 Doris 会停止使用动态分区裁剪的条件进行数据过滤。
- 默认值：90。

doris_max_scan_key_num

- 类型：int。
- 描述：用于限制一个查询请求中，scan node 节点能拆分的最大 scan key 的个数。当一个带有条件的查询请求到达 scan node 节点时，scan node 会尝试将查询条件中 key 列相关的条件拆分成多个 scan key range。之后这些 scan key range 会被分配给多个 scanner 线程进行数据扫描。较大的数值通常意味着可以使用更多的 scanner 线程来提升扫描操作的并行度。但在高并发场景下，过多的线程可能会带来更大的调度开销和系统负载，反而会降低查询响应速度。一个经验数值为 50。该配置可以单独进行会话级别的配置，具体可参阅 变量 中 max_scan_key_num 的说明。
- 默认值：1024。

当在高并发场景下发下并发度无法提升时，可以尝试降低该数值并观察影响。

doris_scan_range_row_count

- 类型：int32。
- 描述：BE 在进行数据扫描时，会将同一个扫描范围拆分为多个 ScanRange。该参数代表了每个ScanRange 代表扫描数据范围。通过该参数可以限制单个 OlapScanner 占用 io 线程的时间。
- 默认值：524288。

doris_scanner_queue_size

- 类型：int32。

- 描述：TransferThread 与 OlapScanner 之间 RowBatch 的缓存队列的长度。Doris 进行数据扫描时是异步进行的，OlapScanner 扫描上来的 Rowbatch 会放入缓存队列之中，等待上层 TransferThread 取走。
- 默认值：1024。

doris_scanner_row_num

- 默认值：16384。
- 每个扫描线程单次执行最多返回的数据行数。

doris_scanner_thread_pool_queue_size

- 类型：int32。
- 描述：Scanner 线程池的队列长度。在 Doris 的扫描任务之中，每一个 Scanner 会作为一个线程 task 提交到线程池之中等待被调度，而提交的任务数目超过线程池队列的长度之后，后续提交的任务将阻塞直到队列之中有新的空缺。
- 默认值：102400。

doris_scanner_thread_pool_thread_num

- 类型：int32。
- 描述：Scanner 线程池线程数目。在 Doris 的扫描任务之中，每一个 Scanner 会作为一个线程 task 提交到线程池之中等待被调度，该参数决定了 Scanner 线程池的大小。
- 默认值：48。

download_low_speed_limit_kbps

- 默认值：50 (KB/s)。
- 下载最低限速。

download_low_speed_time

- 默认值：300 (s)。
- 下载时间限制，默认300秒。

download_worker_count

- 默认值：1。
- 下载线程数，默认1个。

drop_tablet_worker_count

- 默认值：3。
- 删除 tablet 的线程数。

enable_metric_calculator

默认值：true。

如果设置为 true，metric calculator 将运行，收集 BE 相关指标信息，如果设置成 false 将不运行。

enable_partitioned_aggregation

- 类型：bool。
- 描述：BE 节点是否通过 PartitionAggregateNode 来实现聚合操作，如果 false 的话将会执行 AggregateNode 完成聚合。非特殊需求场景不建议设置为 false。
- 默认值：true。

enable_prefetch

- 类型：bool。
- 描述：当使用PartitionedHashTable进行聚合和join计算时，是否进行HashBuket的预取，推荐设置为true。
- 默认值：true。

enable_quadratic_probing

- 类型：bool。
- 描述：当使用 PartitionedHashTable 时发生 Hash 冲突时，是否采用平方探测法来解决 Hash 冲突。该值为 false 的话，则选用线性探测发来解决 Hash 冲突。关于平方探测法可参考：quadratic_probing。
- 默认值：true。

enable_system_metrics

- 默认值：true。
- 用户控制打开和关闭系统指标。

enable_token_check

- 默认值：true。
- 用于向前兼容，稍后将被删除。

es_http_timeout_ms

- 默认值：5000 (ms)。
- 通过 http 连接 ES 的超时时间，默认是5秒。

es_scroll_keepalive

- 默认值：5m。
- es scroll Keeplive保持时间，默认5分钟。

etl_thread_pool_queue_size

- 默认值：256。
- ETL线程池的大小。

etl_thread_pool_size

exchg_node_buffer_size_bytes

- 类型：int32。
- 描述：ExchangeNode 节点 Buffer 队列的大小，单位为 byte。来自 Sender 端发送的数据量大于 ExchangeNode 的 Buffer 大小之后，后续发送的数据将阻塞直到 Buffer 腾出可写入的空间。
- 默认值：10485760。

file_descriptor_cache_capacity

- 默认值：32768。
- 文件句柄缓存的容量，默认缓存32768个文件句柄。

cache_clean_interval

- 默认值：1800 (s)。
- 文件句柄缓存清理的间隔，用于清理长期不用的文件句柄。同时也是 Segment Cache 的清理间隔时间。

flush_thread_num_per_store

- 默认值：2。
- 每个store用于刷新内存表的线程数。

force_recovery

fragment_pool_queue_size

- 默认值：2048。
- 单节点上能够处理的查询请求上限。

fragment_pool_thread_num_min

- 默认值：64。

fragment_pool_thread_num_max

- 默认值：256。
- 查询线程数，默认最小启动64个线程，后续查询请求动态创建线程，最大创建256个线程。

heartbeat_service_port

- 类型：int32。
- 描述：BE 上心跳服务端口（thrift），用于接收来自 FE 的心跳。
- 默认值：9050。

heartbeat_service_thread_count

- 类型：int32。
- 描述：执行 BE 上心跳服务的线程数，默认为1，不建议修改。
- 默认值：1。

ignore_broken_disk

当 BE 启动时，会检查 <dx-inline-code-holder></dx-inline-code-holder> 配置下的所有路径。

- storage_root_path
  如果路径不存在或路径下无法进行读写文件(坏盘)，将忽略此路径，如果有其他可用路径则不中断启动。
- ignore_broken_disk=true
  如果路径不存在或路径下无法进行读写文件(坏盘)，将中断启动失败退出。
- 默认为 false。

ignore_load_tablet_failure

- 类型：bool。
- 描述：用来决定在有tablet 加载失败的情况下是否忽略错误，继续启动be。
- 默认值：false。

BE 启动时，会对每个数据目录单独启动一个线程进行 tablet header 元信息的加载。默认配置下，如果某个数据目录有 tablet 加载失败，则启动进程会终止。同时会在 ignore_broken_disk=false 日志中看到如下错误信息：

```
load tablets from header failed, failed tablets size: xxx, path=xxx
```

表示该数据目录共有多少 tablet 加载失败。同时，日志中也会有加载失败的 tablet 的具体信息。此时需要人工介入来对错误原因进行排查。排查后，通常有两种方式进行恢复：

1. tablet 信息不可修复，在确保其他副本正常的情况下，可以通过 be.INFO 工具将错误的tablet删除。
2. 将 meta_tool 设置为 true，则 BE 会忽略这些错误的 tablet，正常启动。

ignore_rowset_stale_unconsistent_delete

- 类型：bool。
- 描述：用来决定当删除过期的合并过的rowset后无法构成一致的版本路径时，是否仍要删除。

- 默认值：false。

合并的过期 rowset 版本路径会在半个小时后进行删除。在异常下，删除这些版本会出现构造不出查询一致路径的问题，当配置为 false 时，程序检查比较严格，程序会直接报错退出。
当配置为 true 时，程序会正常运行，忽略这个错误。一般情况下，忽略这个错误不会对查询造成影响，仅会在 fe 下发了合并过的版本时出现-230错误。

inc_rowset_expired_sec

- 默认值：1800（s）。
- 导入激活的数据，存储引擎保留的时间，用于增量克隆。

index_stream_cache_capacity

- 默认值：10737418240。
- BloomFilter/Min/Max 等统计信息缓存的容量。

load_data_reserve_hours

- 默认值：4（小时）。
- 用于 mini load。mini load 数据文件将在此时间后被删除。

load_error_log_reserve_hours

- 默认值：48（小时）。
- load 错误日志将在此时间后删除。

load_process_max_memory_limit_bytes

- 默认值：107374182400。
- 单节点上所有的导入线程占据的内存上限，默认值：100G。

将这些默认值设置得很大，因为我们不想在用户升级 Doris 时影响负载性能。 如有必要，用户应正确设置这些配置。

load_process_max_memory_limit_percent

- 默认值：80。
- 单节点上所有的导入线程占据的内存上限比例，默认80%。

将这些默认值设置得很大，因为我们不想在用户升级 Doris 时影响负载性能。 如有必要，用户应正确设置这些配置。

log_buffer_level

- 默认值：空。
- 日志刷盘的策略，默认保持在内存中。

madvise_huge_pages

- 默认值：false。
- 是否使用linux内存大页，默认不启用。

make_snapshot_worker_count

- 默认值：5。
- 制作快照的线程数。

max_client_cache_size_per_host

- 默认值：10。

每个主机的最大客户端缓存数，BE 中有多种客户端缓存，但目前我们使用相同的缓存大小配置。 如有必要，使用不同的配置来设置不同的客户端缓存。

max_compaction_threads

- 类型：int32。
- 描述：Compaction 线程池中线程数量的最大值。
- 默认值：10。

max_consumer_num_per_group

- 默认值：3。
- 一个数据消费者组中的最大消费者数量，用于 routine load。

min_cumulative_compaction_num_singleton_deltas

- 默认值：5。
- cumulative compaction 策略：最小增量文件的数量。

max_cumulative_compaction_num_singleton_deltas

- 默认值：1000。
- cumulative compaction 策略：最大增量文件的数量。

max_download_speed_kbps

- 默认值：50000（kb/s）。
- 最大下载速度限制。

max_free_io_buffers

- 默认值：128。

- 对于每个 io 缓冲区大小，IoMgr 将保留的最大缓冲区数从1024B 到8MB 的缓冲区，最多约为2GB 的缓冲区。

max_garbage_sweep_interval

- 默认值：3600。
- 磁盘进行垃圾清理的最大间隔，默认一个小时。

max_memory_sink_batch_count

- 默认值：20。
- 最大外部扫描缓存批次计数，表示缓存 max_memory_cache_batch_count * batch_size row，默认为20，batch_size 的默认值为1024，表示将缓存20 * 1024行。

max_percentage_of_error_disk

- 类型：int32。
- 描述：存储引擎允许存在损坏硬盘的百分比，损坏硬盘超过改比例后，BE 将会自动退出。
- 默认值：0。

max_pushdown_conditions_per_column

- 类型：int。
- 描述：用于限制一个查询请求中，针对单个列，能够下推到存储引擎的最大条件数量。在查询计划执行的过程中，一些列上的过滤条件可以下推到存储引擎，这样可以利用存储引擎中的索引信息进行数据过滤，减少查询需要扫描的数据量。例如等值条件、IN 谓词中的条件等。这个参数在绝大多数情况下仅影响包含 IN 谓词的查询。如 ignore_load_tablet_failure。较大的数值意味值 IN 谓词中更多的条件可以推送给存储引擎，但过多的条件可能会导致随机读的增加，某些情况下可能会降低查询效率。该配置可以单独进行会话级别的配置，具体可参阅 变量 中 WHERE colA IN (1,2,3,4,...) 的说明。
- 默认值：1024
- 示例
  表结构为 max_pushdown_conditions_per_column，查询请求为 id INT, col2 INT, col3 varchar(32), ...，如果 IN 谓词中的条件数量超过了该配置，则可以尝试增加该配置值，观察查询响应是否有所改善。

max_runnings_transactions_per_txn_map

- 默认值：100。
- txn 管理器中每个 txn_partition_map 的最大 txns 数，这是一种自我保护，以避免在管理器中保存过多的 txns。

max_send_batch_parallelism_per_job

- 类型：int。

- 描述：OlapTableSink 发送批处理数据的最大并行度，用户为 ... WHERE id IN (v1, v2, v3, ...) 设置的值不允许超过 send_batch_parallelism ，如果超过， max_send_batch_parallelism_per_job 将被设置为 send_batch_parallelism 的值。
- 默认值：1。

max_tablet_num_per_shard

- 默认：1024。
- 每个 shard 的 tablet 数目，用于划分 tablet，防止单个目录下 tablet 子目录过多。

max_tablet_version_num

- 类型：int。
- 描述：限制单个 tablet 最大 version 的数量。用于防止导入过于频繁，或 compaction 不及时导致的大量 version 堆积问题。当超过限制后，导入任务将被拒绝。
- 默认值：500。

mem_limit

- 类型：string。
- 描述：限制 BE 进程使用服务器最大内存百分比。用于防止 BE 内存挤占太多的机器内存，该参数必须大于0，当百分大于100%之后，该值会默认为100%。
- 默认值：80%。

memory_limitation_per_thread_for_schema_change

- 默认值：2（GB）。
- 单个 schema change 任务允许占用的最大内存。

memory_maintenance_sleep_time_s

- 默认值：10 。
- 内存维护迭代之间的休眠时间（以秒为单位）。

memory_max_alignment

- 默认值：16。
- 最大校对内存。

read_size

- 默认值：8388608。
- 读取大小是发送到 OS 的读取大小。 在延迟和整个过程之间进行权衡，试图让磁盘保持忙碌但不引入寻道。 对于 8 MB 读取，随机 io 和顺序 io 的性能相似。

min_buffer_size

默认值：1024。

最小读取缓冲区大小（以字节为单位）。

min_compaction_failure_interval_sec

- 类型：int32。
- 描述：在 cumulative compaction 过程中，当选中的 tablet 没能成功的进行版本合并，则会等待一段时间后才会再次有可能被选中。等待的这段时间就是这个配置的值。
- 默认值：600。
- 单位：秒。

min_compaction_threads

- 类型：int32。
- 描述：Compaction 线程池中线程数量的最小值。
- 默认值：10。

min_file_descriptor_number

- 默认值：60000。
- BE 进程的文件句柄 limit 要求的下限。

min_garbage_sweep_interval

- 默认值：180。
- 磁盘进行垃圾清理的最小间隔，时间秒。

mmap_buffers

- 默认值：false。
- 是否使用 mmap 分配内存，默认不使用。

num_cores

- 类型：int32。
- 描述：BE 可以使用 CPU 的核数。当该值为0时，BE 将从/proc/cpuinfo 之中获取本机的 CPU 核数。
- 默认值：0。

num_disks

- 默认值：0。
- 控制机器上的磁盘数量。 如果为0，则来自系统设置。

num_threads_per_core

- 默认值：3。
- 控制每个内核运行工作的线程数。 通常选择2倍或3倍的内核数量。 这使核心保持忙碌而不会导致过度抖动。

num_threads_per_disk

- 默认值：0。
- 每个磁盘的最大线程数也是每个磁盘的最大队列深度。

number_tablet_writer_threads

- 默认值：16。
- tablet 写线程数。

path_gc_check

- 默认值：true。
- 是否启用回收扫描数据线程检查，默认启用。

path_gc_check_interval_second

- 默认值：86400。
- 回收扫描数据线程检查时间间隔，单位秒。

path_gc_check_step

- 默认值：1000。

path_gc_check_step_interval_ms

- 默认值：10 (ms)。

path_scan_interval_second

- 默认值：86400。

pending_data_expire_time_sec

- 默认值：1800。
- 存储引擎保留的未生效数据的最大时长，默认单位：秒。

periodic_counter_update_period_ms

- 默认值：500。
- 更新速率计数器和采样计数器的周期，默认单位：毫秒。

plugin_path

- 默认值：${DORIS_HOME}/plugin。
- 插件路径。

port

- 类型：int32。
- 描述：BE 单测时使用的端口，在实际环境之中无意义，可忽略。
- 默认值：20001。

pprof_profile_dir

- 默认值：${DORIS_HOME}/log。
- pprof profile 保存目录。

priority_networks

- 默认值：空。
- 为那些有很多 IP 的服务器声明一个选择策略。 请注意，最多应该有一个 IP 与此列表匹配。 这是一个以分号分隔格式的列表，用 CIDR 表示法，例如 10.10.10.0/24 ， 如果没有匹配这条规则的 IP，会随机选择一个。

priority_queue_remaining_tasks_increased_frequency

- 默认值：512。
- the increased frequency of priority for remaining tasks in BlockingPriorityQueue。

publish_version_worker_count

- 默认值：8。
- 生效版本的线程数。

pull_load_task_dir

- 默认值：${DORIS_HOME}/var/pull_load。
- 拉取 laod 任务的目录。

push_worker_count_high_priority

- 默认值：3。
- 导入线程数，用于处理 HIGH 优先级任务。

push_worker_count_normal_priority

- 默认值：3。

- 导入线程数，用于处理 NORMAL 优先级任务。

push_write_mbytes_per_sec

- 类型：int32。
- 描述：导入数据速度控制，默认最快每秒10MB。适用于所有的导入方式。
- 单位：MB。
- 默认值：10。

query_scratch_dirs

- 类型：string。
- 描述：BE 进行数据落盘时选取的目录来存放临时数据，与存储路径配置类似，多目录之间用;分隔。
- 默认值：${DORIS_HOME}。

release_snapshot_worker_count

- 默认值：5。
- 释放快照的线程数。

report_disk_state_interval_seconds

- 默认值：60。
- 代理向 FE 报告磁盘状态的间隔时间（秒）。

report_tablet_interval_seconds

- 默认值：60。
- 代理向 FE 报告 olap 表的间隔时间（秒）。

report_task_interval_seconds

- 默认值：10。
- 代理向 FE 报告任务签名的间隔时间（秒）。

result_buffer_cancelled_interval_time

- 默认值：300。
- 结果缓冲区取消时间（单位：秒）。

routine_load_thread_pool_size

- 默认值：10。
- routine load 任务的线程池大小。这应该大于 FE 配置 'max_concurrent_task_num_per_be'（默认 5）。

row_nums_check

- 默认值：true。
- 检查 BE/CE 和 schema 更改的行号。true 是打开的，false 是关闭的。

row_step_for_compaction_merge_log

- 类型：int64。
- 描述：Compaction 执行过程中，每次合并 row_step_for_compaction_merge_log 行数据会打印一条 LOG。如果该参数被设置为0，表示 merge 过程中不需要打印 LOG。
- 默认值： 0。
- 可动态修改：是。

scan_context_gc_interval_min

- 默认值：5。
- 此配置用于上下文 gc 线程调度周期 ， 注意：单位为分钟，默认为 5 分钟。

send_batch_thread_pool_thread_num

- 类型：int32。
- 描述：SendBatch 线程池线程数目。在 NodeChannel 的发送数据任务之中，每一个 NodeChannel 的 SendBatch 操作会作为一个线程 task 提交到线程池之中等待被调度，该参数决定了 SendBatch 线程池的大小。
- 默认值：256。

send_batch_thread_pool_queue_size

- 类型：int32。
- 描述：SendBatch线程池的队列长度。在NodeChannel的发送数据任务之中，每一个NodeChannel的SendBatch操作会作为一个线程task提交到线程池之中等待被调度，而提交的任务数目超过线程池队列的长度之后，后续提交的任务将阻塞直到队列之中有新的空缺。
- 默认值：102400。

serialize_batch

- 默认值：false。
- BE 之间 rpc 通信是否序列化 RowBatch，用于查询层之间的数据传输。

sleep_one_second

- 类型：int32。
- 描述：全局变量，用于 BE 线程休眠1秒，不应该被修改。
- 默认值：1。

small_file_dir

- 默认值：${DORIS_HOME}/lib/small_file/。
- 用于保存 SmallFileMgr 下载的文件的目录。

snapshot_expire_time_sec

- 默认值：172800。
- 快照文件清理的间隔，默认值：48小时。

status_report_interval

- 默认值：5。
- 配置文件报告之间的间隔；单位：秒。

storage_flood_stage_left_capacity_bytes

- 默认值：1073741824。
- 数据目录应该剩下的最小存储空间，默认1G。

storage_flood_stage_usage_percent

- 默认值：95（95%）。
- storage_flood_stage_usage_percent 和 storage_flood_stage_left_capacity_bytes 两个配置限制了数据目录的磁盘容量的最大使用。 如果这两个阈值都达到，则无法将更多数据写入该数据目录。 数据目录的最大已用容量百分比。

storage_medium_migrate_count

- 默认值：1。
- 要克隆的线程数。

storage_page_cache_limit

- 默认值：20%。
- 缓存存储页大小。

index_page_cache_percentage

- 类型：int32。
- 描述：索引页缓存占总页面缓存的百分比，取值为[0, 100]。
- 默认值：10。

storage_root_path

- 类型：string。
  描述：BE 数据存储的目录,多目录之间用英文状态的分号“;”分隔。可以通过路径区别存储目录的介质，HDD 或 SSD。可以添加容量限制在每个路径的末尾，通过英文状态逗号“,”隔开。

- 示例1：
  如果是 SSD 磁盘要在目录后面加上.SSD，HDD 磁盘在目录后面加 .HDD。
  storage_root_path=/home/disk1/doris.HDD,50;/home/disk2/doris.SSD,10;/home/disk2/doris
  - /home/disk1/doris.HDD, 50，表示存储限制为50GB, HDD。
  - /home/disk2/doris.SSD 10， 存储限制为10GB，SSD。
  - /home/disk2/doris，存储限制为磁盘最大容量，默认为 HDD。

- 示例2：
  storage_root_path=/home/disk1/doris,medium:hdd,capacity:50;/home/disk2/doris,medium:ssd,capacity:50
  - /home/disk1/doris,medium:hdd,capacity:10，表示存储限制为10GB, HHD。
  - /home/disk2/doris,medium:ssd,capacity:50，表示存储限制为50GB, SSD。

> ⚠ 注意：
> 不论 HHD 磁盘目录还是 SSD 磁盘目录，文件夹目录名称都无需添加后缀，storage_root_path 参数里指定 medium 即可。

- 默认值：${DORIS_HOME}。

storage_strict_check_incompatible_old_format

- 类型：bool。
- 描述：用来检查不兼容的旧版本格式时是否使用严格的验证方式。
- 默认值： true。
- 可动态修改：否。

配置用来检查不兼容的旧版本格式时是否使用严格的验证方式，当含有旧版本的 hdr 格式时，使用严谨的方式时，程序会打出 fatal log 并且退出运行；否则，程序仅打印 warn log.。

streaming_load_max_mb

- 类型：int64。
- 描述：用于限制数据格式为 csv 的一次 Stream load 导入中，允许的最大数据量。单位 MB。
- 默认值： 10240。
- 可动态修改：是。

Stream Load 一般适用于导入几个 GB 以内的数据，不适合导入过大的数据。

streaming_load_json_max_mb

- 类型：int64。
- 描述：用于限制数据格式为 json 的一次 Stream load 导入中，允许的最大数据量。单位 MB。
- 默认值： 100。
- 可动态修改：是。

一些数据格式，如 JSON，无法进行拆分处理，必须读取全部数据到内存后才能开始解析，因此，这个值用于限制此类格式数据单次导入最大数据量。

streaming_load_rpc_max_alive_time_sec

- 默认值：1200。
- TabletsChannel 的存活时间。如果此时通道没有收到任何数据， 通道将被删除。

sync_tablet_meta

- 默认值：false。
- 存储引擎是否开 sync 保留到磁盘上。

sys_log_dir

- 类型：string。
- 描述：BE 日志数据的存储目录。
- 默认值：${DORIS_HOME}/log。

sys_log_level

- 默认值：INFO。
- 日志级别，INFO < WARNING < ERROR < FATAL。

sys_log_roll_mode

- 默认值：SIZE-MB-1024。
- 日志拆分的大小，每1G 拆分一个日志文件。

sys_log_roll_num

- 默认值：10。
- 日志文件保留的数目。

sys_log_verbose_level

- 默认值：10。
- 日志显示的级别，用于控制代码中 VLOG 开头的日志输出。

sys_log_verbose_modules

- 默认值：空。
- 日志打印的模块，写 olap 就只打印 olap 模块下的日志。

tablet_map_shard_size

- 默认值：1。
- tablet_map_lock 分片大小，值为 2^n, n=0,1,2,3,4 ，这是为了更好地管理 tablet。

tablet_meta_checkpoint_min_interval_secs

- 默认值：600（秒）。
- TabletMeta Checkpoint 线程轮询的时间间隔。

tablet_meta_checkpoint_min_new_rowsets_num

- 默认值：10。
- TabletMeta Checkpoint 的最小 Rowset 数目。

tablet_scan_frequency_time_node_interval_second

- 类型：int64。
- 描述：用来表示记录 metric 'query_scan_count' 的时间间隔。为了计算当前一段时间的 tablet 的 scan 频率，需要每隔一段时间记录一次 metric 'query_scan_count'。
- 默认值：300。

tablet_stat_cache_update_interval_second

- 默认值：300。
- tablet状态缓存的更新间隔，单位：秒。

tablet_rowset_stale_sweep_time_sec

- 类型：int64。
- 描述：用来表示清理合并版本的过期时间，当前时间 now() 减去一个合并的版本路径中 rowset 最近创建创建时间大于 tablet_rowset_stale_sweep_time_sec 时，对当前路径进行清理，删除这些合并过的 rowset, 单位为s。
- 默认值：1800。

当写入过于频繁，磁盘空间不足时，可以配置较少这个时间。不过这个时间过短小于5分钟时，可能会引发 fe 查询不到已经合并过的版本，引发查询-230错误。

tablet_writer_open_rpc_timeout_sec

- 默认值：60。
- 在远程 BE 中打开 tablet writer 的 rpc 超时。 操作时间短，可设置短超时时间。

tablet_writer_ignore_eovercrowded

- 类型：bool。
- 描述：写入时可忽略 brpc 的'[E1011]The server is overcrowded'错误。
- 默认值：false。

当遇到'[E1011]The server is overcrowded'的错误时，可以调整配置。项 `max_send_batch_parallelism_per_job` ，但这个配置项不能动态调整。所以可通过设置此项为 `brpc_socket_max_unwritten_bytes` 来临时避免写失败。注意，此配置项只影响写流程，其他的 rpc 请求依旧会检查是否 overcrowded。

tc_free_memory_rate

- 默认值：20 (%)。
- 可用内存，取值范围：[0-100]。

tc_max_total_thread_cache_bytes

- 类型：int64。
- 描述：用来限制 tcmalloc 中总的线程缓存大小。这个限制不是硬限，因此实际线程缓存使用可能超过这个限制。具体可参阅 TCMALLOC_MAX_TOTAL_THREAD_CACHE_BYTES。
- 默认值： 1073741824。

如果发现系统在高压力场景下，通过 BE 线程堆栈发现大量线程处于 tcmalloc 的锁竞争阶段，如大量的 `true` 相关堆栈，则可以尝试增大该参数来提升系统性能。具体可参考：这里。

tc_use_memory_min

- 默认值：10737418240
- TCmalloc 的最小内存，当使用的内存小于这个时，不返回给操作系统。

thrift_client_retry_interval_ms

- 类型：int64。
- 描述：用来为 be 的 thrift 客户端设置重试间隔, 避免 fe 的 thrift server 发生雪崩问题，单位为ms。
- 默认值：1000。

thrift_connect_timeout_seconds

- 默认值：3。
- 默认 thrift 客户端连接超时时间（单位：秒）。

thrift_rpc_timeout_ms

- 默认值：5000。
- thrift 默认超时时间，默认：5秒。

thrift_server_type_of_fe

- 该配置表示 FE 的 Thrift 服务使用的服务模型, 类型为 string, 大小写不敏感,该参数需要和 fe 的 thrift_server_type 参数的设置保持一致。目前该参数的取值有两个,SpinLock和THREADED。
- 若该参数为THREAD_POOL, 该模型为非阻塞式 I/O 模型。
- 若该参数为THREADED, 该模型为阻塞式 I/O 模型。

total_permits_for_compaction_score

- 类型：int64。
- 描述：被所有的 compaction 任务所能持有的 "permits" 上限，用来限制 compaction 占用的内存。
- 默认值：10000。
- 可动态修改：是。

trash_file_expire_time_sec

- 默认值：259200。
- 回收站清理的间隔，72个小时，当磁盘空间不足时，trash 下的文件保存期可不遵守这个参数。

txn_commit_rpc_timeout_ms

- 默认值：10000。
- txn 提交 rpc 超时，默认10秒。

txn_map_shard_size

- 默认值：128。
- txn_map_lock 分片大小，取值为2^n，n=0,1,2,3,4。这是一项增强功能，可提高管理 txn 的性能。

txn_shard_size

- 默认值：1024。
- txn_lock 分片大小，取值为2^n，n=0,1,2,3,4， 这是一项增强功能，可提高提交和发布 txn 的性能。

unused_rowset_monitor_interval

- 默认值：30。
- 清理过期 Rowset 的时间间隔，单位：秒。

upload_worker_count

- 默认值：1。
- 上传文件最大线程数。

use_mmap_allocate_chunk

- 默认值：false。
- 是否使用 mmap 分配块。 如果启用此功能，最好增加 vm.max_map_count 的值，其默认值为 65530。您可以通过“sysctl -w vm.max_map_count=262144”或“echo 262144 > /proc/sys/vm/”以 root 身份进行操作 max_map_count" ，当这个设置为 true 时，您必须将 chunk_reserved_bytes_limit 设置为一个相对较大的数字，否则性能非常非常糟糕。

user_function_dir

- 默认值：${DORIS_HOME}/lib/udf。
- udf 函数目录。

webserver_num_workers

- 默认值：48。
- webserver 默认工作线程数。

webserver_port

- 类型：int32。
- 描述：BE 上的 http server 的服务端口。
- 默认值：8040。

write_buffer_size

- 默认值：104857600。
- 刷写前缓冲区的大小。

zone_map_row_num_threshold

- 类型： int32。
- 描述: 如果一个 page 中的行数小于这个值就不会创建 zonemap，用来减少数据膨胀。
- 默认值: 20。

aws_log_level

- 类型: int32。
- 描述: AWS SDK 的日志级别。

```
Off = 0,
Fatal = 1,
Error = 2,
Warn = 3,
Info = 4,
Debug = 5,
Trace = 6
```

- 默认值: 3。

mem_tracker_level

- 类型: int16。
- 描述: MemTracker 在 Web 页面上展示的级别，等于或低于这个级别的 MemTracker 会在 Web 页面上展示。

```
RELEASE = 0
DEBUG = 1
```

- 默认值: 0

max_segment_num_per_rowset

- 类型: int32
- 描述: 用于限制导入时，新产生的 rowset 中的 segment 数量。如果超过阈值，导入会失败并报错 -238。过多的 segment 会导致 compaction 占用大量内存引发 OOM 错误。
- 默认值: 200。

remote_storage_read_buffer_mb

- 类型: int32。
- 描述: 读取 hdfs 或者对象存储上的文件时，使用的缓存大小。
- 默认值: 16MB。

增大这个值，可以减少远端数据读取的调用次数，但会增加内存开销。

external_table_connect_timeout_sec

- 类型: int32。

- 描述: 和外部表建立连接的超时时间。
- 默认值: 5秒。

segment_cache_capacity

- 类型: int32。
- 描述: Segment Cache 缓存的 Segment 最大数量。
- 默认值: 1000000。

默认值目前只是一个经验值，可能需要根据实际场景修改。增大该值可以缓存更多的 segment 从而避免一些 IO。减少该值则会降低内存使用。

用户配置项

最近更新时间：2022-03-14 15:10:48

本文档主要介绍了 User 级别的相关配置项。User 级别的配置生效范围为单个用户。每个用户都可以设置自己的 User property。相互不影响。

查看配置项

FE 启动后，在 MySQL 客户端，通过下面命令查看 User 的配置项：
SHOW PROPERTY [FOR user] [LIKE key pattern]；具体语法可通过命令： help show property; 查询。

设置配置项

FE 启动后，在MySQL 客户端，通过下面命令修改 User 的配置项：
SET PROPERTY [FOR 'user'] 'key' = 'value' [, 'key' = 'value']；具体语法可通过命令： help set property; 查询。

User 级别的配置项只会对指定用户生效，并不会影响其他用户的配置。

应用举例

1. 修改用户 Billie 的 max_user_connections。
   通过 SHOW PROPERTY FOR 'Billie' LIKE '%max_user_connections%'; 查看 Billie 用户当前的最大链接数为 100。
   通过 SET PROPERTY FOR 'Billie' 'max_user_connections' = '200'; 修改 Billie 用户的当前最大连接数到 200。

配置项列表

max_user_connections

用户最大的连接数，默认值为100。一般情况不需要更改该参数，除非查询的并发数超过了默认值。

max_query_instances

用户同一时间点可使用的 instance 个数, 默认是-1，小于等于0将会使用配置 default_max_query_instances.

resource

quota

default_load_cluster

load_cluster

监控配置
FE 监控项

最近更新时间：2022-04-13 17:05:55

文档主要介绍 FE 的相关监控项。

查看监控项

FE 的监控项可以通过以下方式访问： http://fe_host:fe_http_port/metrics ；默认显示为 Prometheus 格式。

通过以下接口可以获取 Json 格式的监控项： http://fe_host:fe_http_port/metrics?type=json 。

监控项列表

doris_fe_snmp{name="tcp_in_errs"}

该监控项为 /proc/net/snmp 中的 Tcp: InErrs 字段值。表示当前接收到的错误的 TCP 包的数量。
结合采样周期可以计算发生率。通常用于排查网络问题。

doris_fe_snmp{name="tcp_retrans_segs"}

该监控项为 /proc/net/snmp 中的 Tcp: RetransSegs 字段值。表示当前重传的 TCP 包的数量。
结合采样周期可以计算发生率。通常用于排查网络问题。

doris_fe_snmp{name="tcp_in_segs"}

该监控项为 /proc/net/snmp 中的 Tcp: InSegs 字段值。表示当前接收到的所有 TCP 包的数量。
通过 (NEW_tcp_in_errs - OLD_tcp_in_errs) / (NEW_tcp_in_segs - OLD_tcp_in_segs) 可以计算接收到的 TCP 错误包率。通常用于排查网络问题。

doris_fe_snmp{name="tcp_out_segs"}

该监控项为 /proc/net/snmp 中的 Tcp: OutSegs 字段值。表示当前发送的所有带 RST 标记的 TCP 包的数量。
通过 (NEW_tcp_tcp_retrans_segs - OLD_tcp_retrans_segs) / (NEW_tcp_out_segs - OLD_tcp_out_segs) 可以计算 TCP 重传率。通常用于排查网络问题。

doris_fe_meminfo{name="memory_total"}

该监控项为 /proc/meminfo 中的 MemTotal 字段值。表示所有可用的内存大小，总的物理内存减去预留空间和内核大小。通常用于排查内存问题。

doris_fe_meminfo{name="memory_free"}

该监控项为 /proc/meminfo 中的 MemFree 字段值。表示系统尚未使用的内存。通常用于排查内存问题。

doris_fe_meminfo{name="memory_available"}

该监控项为 /proc/meminfo 中的 MemAvailable 字段值。真正的系统可用内存，系统中有些内存虽然已被使用但是可以回收的，所以这部分可回收的内存加上MemFree才是系统可用的内存。通常用于排查内存问题。

doris_fe_meminfo{name="buffers"}

该监控项为 /proc/meminfo 中的 Buffers 字段值。表示用来给块设备做缓存的内存(文件系统的metadata、pages)。通常用于排查内存问题。

doris_fe_meminfo{name="cached"}

该监控项为 /proc/meminfo 中的 Cached 字段值。表示分配给文件缓冲区的内存。通常用于排查内存问题。

jvm_thread{type="count"}

该监控项表示 FE 节点当前 JVM 总的线程数量，包含 daemon 线程和非 daemon 线程。通常用于排查 FE 节点的 JVM 线程运行问题。

jvm_thread{type="peak_count"}

该监控项表示 FE 节点从 JVM 启动以来的最大峰值线程数量。
通常用于排查 FE 节点的 JVM 线程运行问题。

jvm_thread{type="new_count"}

该监控项表示 FE 节点 JVM 中处于 NEW 状态的线程数量。
通常用于排查 FE 节点的 JVM 线程运行问题。

jvm_thread{type="runnable_count"}

该监控项表示 FE 节点 JVM 中处于 RUNNABLE 状态的线程数量。
通常用于排查 FE 节点的 JVM 线程运行问题。

jvm_thread{type="blocked_count"}

该监控项表示 FE 节点 JVM 中处于 BLOCKED 状态的线程数量。
通常用于排查 FE 节点的 JVM 线程运行问题。

jvm_thread{type="waiting_count"}

该监控项表示 FE 节点 JVM 中处于 WAITING 状态的线程数量。
通常用于排查 FE 节点的 JVM 线程运行问题。

jvm_thread{type="timed_waiting_count"}

该监控项表示 FE 节点 JVM 中处于 TIMED_WAITING 状态的线程数量。
通常用于排查 FE 节点的 JVM 线程运行问题。

jvm_thread{type="terminated_count"}

该监控项表示 FE 节点 JVM 中处于 TERMINATED 状态的线程数量。

通常用于排查 FE 节点的 JVM 线程运行问题。

BE 监控项

最近更新时间：2022-03-14 15:11:23

本文主要介绍 BE 的相关监控项。

查看监控项

BE 的监控项可以通过以下方式访问： http://be_host:be_webserver_port/metrics ；默认显示为 Prometheus 格式。
通过以下接口可以获取 Json 格式的监控项： http://be_host:be_webserver_port/metrics?type=json 。

监控项列表

doris_be_snmp{name="tcp_in_errs"}

该监控项为 /proc/net/snmp 中的 Tcp: InErrs 字段值。表示当前接收到的错误的 TCP 包的数量。
结合采样周期可以计算发生率。通常用于排查网络问题。

doris_be_snmp{name="tcp_retrans_segs"}

该监控项为 /proc/net/snmp 中的 Tcp: RetransSegs 字段值。表示当前重传的 TCP 包的数量。
结合采样周期可以计算发生率。通常用于排查网络问题。

doris_be_snmp{name="tcp_in_segs"}

该监控项为 /proc/net/snmp 中的 Tcp: InSegs 字段值。表示当前接收到的所有 TCP 包的数量。
通过（NEW_tcp_in_errs - OLD_tcp_in_errs) / (NEW_tcp_in_segs - OLD_tcp_in_segs）可以计算接收到的 TCP 错误包率。通常用于排查网络问题。

doris_be_snmp{name="tcp_out_segs"}

该监控项为 /proc/net/snmp 中的 Tcp: OutSegs 字段值。表示当前发送的所有带 RST 标记的 TCP 包的数量。
通过（NEW_tcp_tcp_retrans_segs - OLD_tcp_retrans_segs) / (NEW_tcp_out_segs - OLD_tcp_out_segs）可以计算 TCP 重传率。通常用于排查网络问题。

doris_be_compaction_mem_current_consumption

该监控项为 Compaction 使用的 MemPool 总和（所有 Compaction 线程）。通过该值，可以迅速判断 Compaction 是否占用过多内存，引起高内存占用，甚至 OOM 等问题。
通常用于排查内存使用问题。

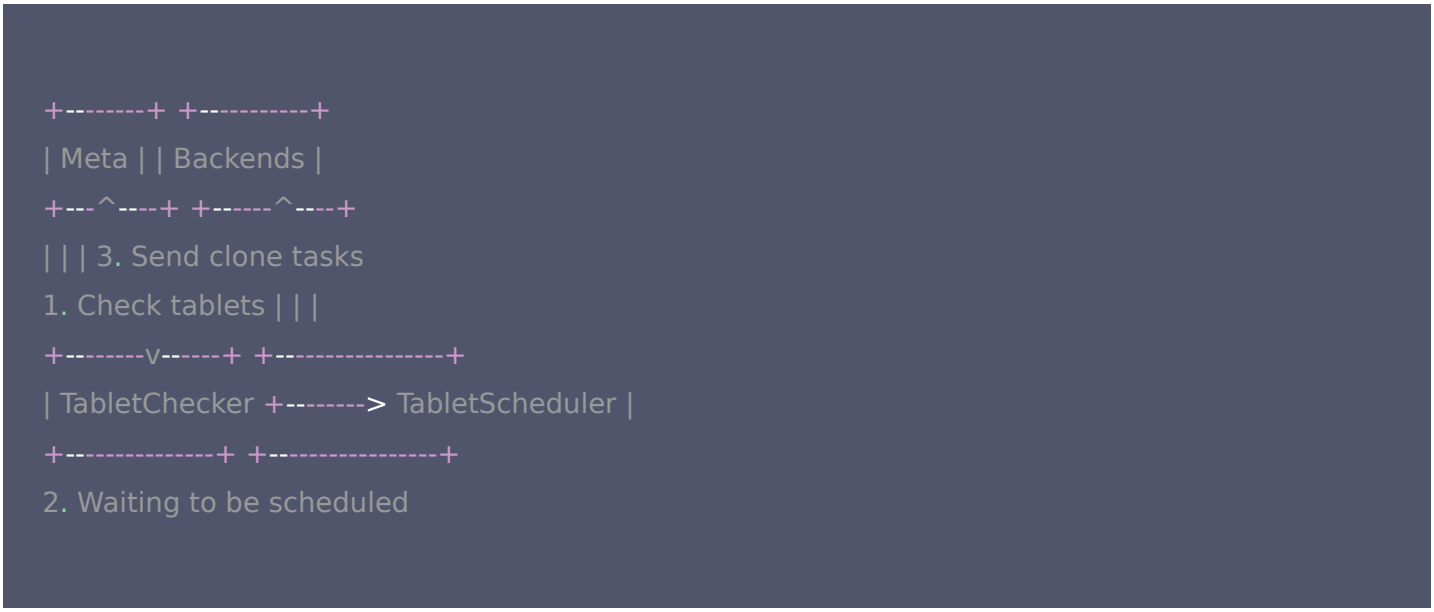
数据副本管理

最近更新时间：2022-06-30 14:23:26

从 0.9.0 版本开始，Doris 引入了优化后的副本管理策略，同时支持了更为丰富的副本状态查看工具。本文档主要介绍 Doris 数据副本均衡、修复方面的调度策略，以及副本管理的运维方法。帮助用户更方便的掌握和管理集群中的副本状态。

名词解释

- Tablet：Doris 表的逻辑分片，一个表有多个分片。
- Replica：分片的副本，默认一个分片有3个副本。
- Healthy Replica：健康副本，副本所在 Backend 存活，且副本的版本完整。
- TabletChecker（TC）：是一个常驻的后台线程，用于定期扫描所有的 Tablet，检查这些 Tablet 的状态，并根据检查结果，决定是否将 tablet 发送给 TabletScheduler。
- TabletScheduler（TS）：是一个常驻的后台线程，用于处理由 TabletChecker 发来的需要修复的 Tablet。同时也会进行集群副本均衡的工作。
- TabletSchedCtx（TSC）：是一个 tablet 的封装。当 TC 选择一个 tablet 后，会将其封装为一个 TSC，发送给 TS。
- Storage Medium：存储介质。Doris 支持对分区粒度指定不同的存储介质，包括 SSD 和 HDD。副本调度策略也是针对不同的存储介质分别调度的。



> 说明：
> 上图是一个简化的工作流程。

副本状态

一个 Tablet 的多个副本，可能因为某些情况导致状态不一致。Doris 会尝试自动修复这些状态不一致的副本，让集群尽快从错误状态中恢复。

一个 Replica 的健康状态有以下几种：

1. BAD
   即副本损坏。包括但不限于磁盘故障、BUG等引起的副本不可恢复的损毁状态。

2. VERSION_MISSING
   版本缺失。Doris 中每一批次导入都对应一个数据版本。而一个副本的数据由多个连续的版本组成。而由于导入错误、延迟等原因，可能导致某些副本的数据版本不完整。

3. HEALTHY
   健康副本。即数据正常的副本，并且副本所在的 BE 节点状态正常（心跳正常且不处于下线过程中）

一个 Tablet 的健康状态由其所有副本的状态决定，有以下几种：

1. REPLICA_MISSING
   副本缺失。即存活副本数小于期望副本数。

2. VERSION_INCOMPLETE
   存活副本数大于等于期望副本数，但其中健康副本数小于期望副本数。

3. REPLICA_RELOCATING
   拥有等于 replication num 的版本完整的存活副本数，但是部分副本所在的 BE 节点处于 unavailable 状态（例如 decommission 中）。

4. REPLICA_MISSING_IN_CLUSTER
   当使用多 cluster 方式时，健康副本数大于等于期望副本数，但在对应 cluster 内的副本数小于期望副本数。

5. REDUNDANT
   副本冗余。健康副本都在对应 cluster 内，但数量大于期望副本数。或者有多余的 unavailable 副本。

6. FORCE_REDUNDANT
   这是一个特殊状态。只会出现在当期望副本数大于等于可用节点数时，并且 Tablet 处于副本缺失状态时出现。这种情况下，需要先删除一个副本，以保证有可用节点用于创建新副本。

7. COLOCATE_MISMATCH
   针对 Colocation 属性的表的分片状态。表示分片副本与 Colocation Group 的指定的分布不一致。

8. COLOCATE_REDUNDANT
   针对 Colocation 属性的表的分片状态。表示 Colocation 表的分片副本冗余。

9. HEALTHY
   健康分片，即条件[1-8]都不满足。

副本修复

TabletChecker 作为常驻的后台进程，会定期检查所有分片的状态。对于非健康状态的分片，将会交给 TabletScheduler 进行调度和修复。修复的实际操作，都由 BE 上的 clone 任务完成。FE 只负责生成这些 clone 任务。

> 说明：
>
> - 副本修复的主要思想是先通过创建或补齐使得分片的副本数达到期望值，然后再删除多余的副本。
> - 一个 clone 任务就是完成从一个指定远端 BE 拷贝指定数据到指定目的端 BE 的过程。

针对不同的状态，我们采用不同的修复方式：

1. REPLICA_MISSING/REPLICA_RELOCATING
   选择一个低负载的，可用的 BE 节点作为目的端。选择一个健康副本作为源端。clone 任务会从源端拷贝一个完整的副本到目的端。对于副本补齐，我们会直接选择一个可用的 BE 节点，而不考虑存储介质。

2. VERSION_INCOMPLETE
   选择一个相对完整的副本作为目的端。选择一个健康副本作为源端。clone 任务会从源端尝试拷贝缺失的版本到目的端的副本。

3. REPLICA_MISSING_IN_CLUSTER
   这种状态处理方式和 REPLICA_MISSING 相同。

4. REDUNDANT
   通常经过副本修复后，分片会有冗余的副本。我们选择一个冗余副本将其删除。冗余副本的选择遵从以下优先级：
   - 副本所在 BE 已经下线。
   - 副本已损坏。
   - 副本所在 BE 失联或在下线中。
   - 副本处于 CLONE 状态（该状态是 clone 任务执行过程中的一个中间状态）。
   - 副本有版本缺失。
   - 副本所在 cluster 不正确。
   - 副本所在 BE 节点负载高。

5. FORCE_REDUNDANT
   不同于 REDUNDANT，因为此时虽然 Tablet 有副本缺失，但是因为已经没有额外的可用节点用于创建新的副本了。所以此时必须先删除一个副本，以腾出一个可用节点用于创建新的副本。删除副本的顺序同 REDUNDANT。

6. COLOCATE_MISMATCH
   从 Colocation Group 中指定的副本分布 BE 节点中选择一个作为目的节点进行副本补齐。

7. COLOCATE_REDUNDANT
   删除一个非 Colocation Group 中指定的副本分布 BE 节点上的副本。

Doris 在选择副本节点时，不会将同一个 Tablet 的副本部署在同一个 host 的不同 BE 上。保证了即使同一个 host 上的所有 BE 都挂掉，也不会造成全部副本丢失。

调度优先级

TabletScheduler 里等待被调度的分片会根据状态不同，赋予不同的优先级。优先级高的分片将会被优先调度。目前有以下几种优先级。

1. VERY_HIGH
   - REDUNDANT。对于有副本冗余的分片，我们优先处理。虽然逻辑上来讲，副本冗余的紧急程度最低，但是因为这种情况处理起来最快且可以快速释放资源（例如磁盘空间等），所以我们优先处理。
   - FORCE_REDUNDANT。同上。

2. HIGH
   - REPLICA_MISSING 且多数副本缺失（例如3副本丢失了2个）。
   - VERSION_INCOMPLETE 且多数副本的版本缺失。
   - COLOCATE_MISMATCH 我们希望 Colocation 表相关的分片能够尽快修复完成。
   - COLOCATE_REDUNDANT。

3. NORMAL
   - REPLICA_MISSING 但多数存活（例如3副本丢失了1个）。
   - VERSION_INCOMPLETE 但多数副本的版本完整。
   - REPLICA_RELOCATING 且多数副本需要 relocate（例如3副本有2个）。

4. LOW
   - REPLICA_MISSING_IN_CLUSTER。
   - REPLICA_RELOCATING 但多数副本 stable。

手动优先级

系统会自动判断调度优先级。但是有些时候，用户希望某些表或分区的分片能够更快的被修复。因此我们提供一个命令，用户可以指定某个表或分区的分片被优先修复： ADMIN REPAIR TABLE tbl [PARTITION (p1, p2, ...)]; 。
这个命令告诉 TC，在扫描 Tablet 时，对需要优先修复的表或分区中的有问题的 Tablet，给予 VERY_HIGH 的优先级。

> ⚠ 注意：

> 这个命令只是一个 hint，并不能保证一定能修复成功，并且优先级也会随 TS 的调度而发生变化。并且当 Master FE 切换或重启后，这些信息都会丢失。

可以通过以下命令取消优先级： ADMIN CANCEL REPAIR TABLE tbl [PARTITION (p1, p2, ...)]; 。

优先级调度

优先级保证了损坏严重的分片能够优先被修复，提高系统可用性。但是如果高优先级的修复任务一直失败，则会导致低优先级的任务一直得不到调度。因此，我们会根据任务的运行状态，动态的调整任务的优先级，保证所有任务都有机会被调度到。

- 连续5次调度失败（如无法获取资源，无法找到合适的源端或目的端等），则优先级会被下调。
- 持续 30 分钟未被调度，则上调优先级。
- 同一 tablet 任务的优先级至少间隔 5 分钟才会被调整一次。

同时为了保证初始优先级的权重，我们规定，初始优先级为 VERY_HIGH 的，最低被下调到 NORMAL。而初始优先级为 LOW 的，最多被上调为 HIGH。这里的优先级调整，也会调整用户手动设置的优先级。

副本均衡

Doris 会自动进行集群内的副本均衡。目前支持两种均衡策略，负载/分区。负载均衡适合需要兼顾节点磁盘使用率和节点副本数量的场景；而分区均衡会使每个分区的副本都均匀分布在各个节点，避免热点，适合对分区读写要求比较高的场景。但是，分区均衡不考虑磁盘使用率，使用分区均衡时需要注意磁盘的使用情况。策略只能在 FE 启动前配置，不支持运行时切换。

负载均衡

负载均衡的主要思想是，对某些分片，先在低负载的节点上创建一个副本，然后再删除这些分片在高负载节点上的副本。同时，因为不同存储介质的存在，在同一个集群内的不同 BE 节点上，可能存在一种或两种存储介质。我们要求存储介质为 A 的分片在均衡后，尽量依然存储在存储介质 A 中。所以我们根据存储介质，对集群的 BE 节点进行划分。然后针对不同的存储介质的 BE 节点集合，进行负载均衡调度。
同样，副本均衡会保证不会将同一个 Tablet 的副本部署在同一个 host 的 BE 上。

BE 节点负载

我们用 ClusterLoadStatistics（CLS）表示一个 cluster 中各个 Backend 的负载均衡情况。
TabletScheduler 根据这个统计值，来触发集群均衡。我们当前通过 **磁盘使用率** 和 **副本数量** 两个指标，为每个 BE计算一个 loadScore，作为 BE 的负载分数。分数越高，表示该 BE 的负载越重。

磁盘使用率和副本数量各有一个权重系数，分别为 **capacityCoefficient** 和 **replicaNumCoefficient**，其 **和衡为 1**。其中 capacityCoefficient 会根据实际磁盘使用率动态调整。当一个 BE 的总体磁盘使用率在 50% 以下，则 capacityCoefficient 值为 0.5，如果磁盘使用率在 75%（可通过 FE 配置项 capacity_used_percent_high_water 配置）以上，则值为 1。如果使用率介于 50% ~ 75% 之间，则该权重系数

平滑增加，公式为：

capacityCoefficient= 2 * 磁盘使用率 - 0.5

该权重系数保证当磁盘使用率过高时，该 Backend 的负载分数会更高，以保证尽快降低这个 BE 的负载。

TabletScheduler 会每隔 20s 更新一次 CLS。

分区均衡

分区均衡的主要思想是，将每个分区的在各个 Backend 上的 replica 数量差（即 partition skew），减少到最小。因此只考虑副本个数，不考虑磁盘使用率。

为了尽量少的迁移次数，分区均衡使用二维贪心的策略，优先均衡partition skew最大的分区，均衡分区时会尽量选择，可以使整个 cluster 的在各个 Backend 上的 replica 数量差（即 cluster skew/total skew）减少的方向。

skew 统计

skew 统计信息由 ClusterBalanceInfo 表示，其中， partitionInfoBySkew 以 partition skew 为key排序，便于找到max partition skew； beByTotalReplicaCount 则是以 Backend 上的所有 replica 个数为key排序。

ClusterBalanceInfo 同样保持在CLS中, 同样 20s 更新一次。

max partition skew 的分区可能有多个，采用随机的方式选择一个分区计算。

均衡策略

TabletScheduler 在每轮调度时，都会通过 LoadBalancer 来选择一定数目的健康分片作为 balance 的候选分片。在下一次调度时，会尝试根据这些候选分片，进行均衡调度。

资源控制

无论是副本修复还是均衡，都是通过副本在各个 BE 之间拷贝完成的。如果同一台 BE 同一时间执行过多的任务，则会带来不小的 IO 压力。因此，Doris 在调度时控制了每个节点上能够执行的任务数目。最小的资源控制单位是磁盘（即在 be.conf 中指定的一个数据路径）。我们默认为每块磁盘配置两个 slot 用于副本修复。一个 clone 任务会占用源端和目的端各一个 slot。如果 slot 数目为零，则不会再对这块磁盘分配任务。该 slot 个数可以通过 FE 的 schedule_slot_num_per_path 参数配置。

另外，我们默认为每块磁盘提供 2 个单独的 slot 用于均衡任务。目的是防止高负载的节点因为 slot 被修复任务占用，而无法通过均衡释放空间。

副本状态查看

副本状态查看主要是查看副本的状态，以及副本修复和均衡任务的运行状态。这些状态大部分都**仅存在于** Master FE 节点中。因此，以下命令需直连到 Master FE 执行。

副本状态

1. 全局状态检查。

通过 SHOW PROC '/statistic'; 命令可以查看整个集群的副本状态。

```
+----------+--------------------------+----------+-------------+----------+-----------+------------+----------------
---+----------------------+
| DbId | DbName | TableNum | PartitionNum | IndexNum | TabletNum | ReplicaNum | UnhealthyT
abletNum | InconsistentTabletNum |
+----------+--------------------------+----------+-------------+----------+-----------+------------+----------------
---+----------------------+
| 35153636 | default_cluster:DF_Newrisk | 3 | 3 | 3 | 96 | 288 | 0 | 0 |
| 48297972 | default_cluster:PaperData | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5909381 | default_cluster:UM_TEST | 7 | 7 | 10 | 320 | 960 | 1 | 0 |
| Total | 240 | 10 | 10 | 13 | 416 | 1248 | 1 | 0 |
+----------+--------------------------+----------+-------------+----------+-----------+------------+----------------
---+----------------------+
```

其中 UnhealthyTabletNum 列显示了对应的 Database 中，有多少 Tablet 处于非健康状态。
InconsistentTabletNum 列显示了对应的 Database 中，有多少 Tablet 处于副本不一致的状态。最后一行 Total 行对整个集群进行了统计。正常情况下 UnhealthyTabletNum 和 InconsistentTabletNum 应为0。如果不为零，可以进一步查看具体有哪些 Tablet。如上图中，UM_TEST 数据库有 1 个 Tablet 状态不健康，则可以使用以下命令查看具体是哪一个 Tablet。
SHOW PROC '/statistic/5909381';，其中 5909381 为对应的 DbId。

```
+-----------------+-------------------+
| UnhealthyTablets | InconsistentTablets |
+-----------------+-------------------+
| [40467980] | [] |
+-----------------+-------------------+
```

上图会显示具体的不健康的 Tablet ID（40467980）。后面我们会介绍如何查看一个具体的 Tablet 的各个副本的状态。

2. 表（分区）级别状态检查。

用户可以通过以下命令查看指定表或分区的副本状态，并可以通过 WHERE 语句对状态进行过滤。如查看表 tbl1 中，分区 p1和 p2上状态为 NORMAL 的副本：

```
ADMIN SHOW REPLICA STATUS FROM tbl1 PARTITION (p1, p2) WHERE STATUS = "NORMAL";
```

```
+----------+----------+----------+---------+-----------------+------------------+----------------+-----------+-
----------+-------+--------+--------+
```

```
| TabletId | ReplicaId | BackendId | Version | LastFailedVersion | LastSuccessVersion | Committe
dVersion | SchemaHash | VersionNum | IsBad | State | Status |
+----------+-----------+-----------+---------+-------------------+--------------------+----------------+-----------+-
-----------+-------+--------+--------+
| 29502429 | 29502432 | 10006 | 2 | -1 | 2 | 1 | -1 | 2 | false | NORMAL | OK |
| 29502429 | 36885996 | 10002 | 2 | -1 | -1 | 1 | -1 | 2 | false | NORMAL | OK |
| 29502429 | 48100551 | 10007 | 2 | -1 | -1 | 1 | -1 | 2 | false | NORMAL | OK |
| 29502433 | 29502434 | 10001 | 2 | -1 | 2 | 1 | -1 | 2 | false | NORMAL | OK |
| 29502433 | 44900737 | 10004 | 2 | -1 | -1 | 1 | -1 | 2 | false | NORMAL | OK |
| 29502433 | 48369135 | 10006 | 2 | -1 | -1 | 1 | -1 | 2 | false | NORMAL | OK |
+----------+-----------+-----------+---------+-------------------+--------------------+----------------+-----------+-
-----------+-------+--------+--------+
```

这里会展示所有副本的状态。其中 IsBad **列为** true **则表示副本已经损坏。而** Status **列则会显示另外的其他状态。具体的状态说明，可以通过** HELP ADMIN SHOW REPLICA STATUS; **查看帮助。**

ADMIN SHOW REPLICA STATUS **命令主要用于查看副本的健康状态。用户还可以通过以下命令查看指定表中副本的一些额外信息：**SHOW TABLET FROM tbl1;**。**

```
+----------+-----------+-----------+-----------+---------+------------+-----------------+----------------------+---
--------------+--------------------+--------------+----------+----------+--------+-----------------------+----------
---+--------------------+-------------+--------------------+--------------------+--------------------+
| TabletId | ReplicaId | BackendId | SchemaHash | Version | VersionHash | LstSuccessVersion | L
stSuccessVersionHash | LstFailedVersion | LstFailedVersionHash | LstFailedTime | DataSize | Ro
wCount | State | LstConsistencyCheckTime | CheckVersion | CheckVersionHash | VersionCount |
PathHash | MetaUrl | CompactionStatus |
+----------+-----------+-----------+-----------+---------+------------+-----------------+----------------------+---
--------------+--------------------+--------------+----------+----------+--------+-----------------------+----------
---+--------------------+-------------+--------------------+--------------------+--------------------+
| 29502429 | 29502432 | 10006 | 1421156361 | 2 | 0 | 2 | 0 | -1 | 0 | N/A | 784 | 0 | NORMAL |
N/A | -1 | -1 | 2 | -5822326203532286804 | url | url |
| 29502429 | 36885996 | 10002 | 1421156361 | 2 | 0 | -1 | 0 | -1 | 0 | N/A | 784 | 0 | NORMAL |
N/A | -1 | -1 | 2 | -1441285706148429853 | url | url |
| 29502429 | 48100551 | 10007 | 1421156361 | 2 | 0 | -1 | 0 | -1 | 0 | N/A | 784 | 0 | NORMAL |
N/A | -1 | -1 | 2 | -4784691547051455525 | url | url |
+----------+-----------+-----------+-----------+---------+------------+-----------------+----------------------+---
--------------+--------------------+--------------+----------+----------+--------+-----------------------+----------
---+--------------------+-------------+--------------------+--------------------+--------------------+
```

上图展示了包括副本大小、行数、版本数量、所在数据路径等一些额外的信息。

> **说明：**
> 这里显示的 State 列的内容不代表副本的健康状态，而是副本处于某种任务下的状态，例如 CLONE、SCHEMA_CHANGE、ROLLUP 等。

此外，用户也可以通过以下命令，查看指定表或分区的副本分布情况，来检查副本分布是否均匀。

```
ADMIN SHOW REPLICA DISTRIBUTION FROM tbl1;
```

```
+-----------+------------+-------+---------+
| BackendId | ReplicaNum | Graph | Percent |
+-----------+------------+-------+---------+
| 10000 | 7 | | 7.29 % |
| 10001 | 9 | | 9.38 % |
| 10002 | 7 | | 7.29 % |
| 10003 | 7 | | 7.29 % |
| 10004 | 9 | | 9.38 % |
| 10005 | 11 | > | 11.46 % |
| 10006 | 18 | > | 18.75 % |
| 10007 | 15 | > | 15.62 % |
| 10008 | 13 | > | 13.54 % |
+-----------+------------+-------+---------+
```

这里分别展示了表 tbl1的副本在各个 BE 节点上的个数、百分比，以及一个简单的图形化显示。

4. Tablet 级别状态检查。

   当我们要定位到某个具体的 Tablet 时，可以使用如下命令来查看一个具体的 Tablet 的状态。如查看 ID 为 29502553 的 tablet： SHOW TABLET 29502553;。

```
+----------------------+-----------+-------------+-----------+---------+---------+------------+---------+-------
-+-----------------------------------------------------------------+
| DbName | TableName | PartitionName | IndexName | DbId | TableId | PartitionId | IndexId | IsS
ync | DetailCmd |
+----------------------+-----------+-------------+-----------+---------+---------+------------+---------+-------
-+-----------------------------------------------------------------+
| default_cluster:test | test | test | test | 29502391 | 29502428 | 29502427 | 29502428 | true |
SHOW PROC '/dbs/29502391/29502428/partitions/29502427/29502428/29502553'; |
```

```
+----------------------+----------+-------------+----------+----------+---------+------------+---------+-------
-+----------------------------------------------------------------------+
```

上图显示了这个 tablet 所对应的数据库、表、分区、上卷表等信息。用户可以复制 DetailCmd 命令中的命令继续执行：SHOW PROC '/dbs/29502391/29502428/partitions/29502427/29502428/29502553';

```
+-----------+-----------+---------+------------+-----------------+---------------------+-----------------+---------
------------+--------------+-----------+----------+----------+--------+-------+-------------+---------------------+
----------+-----------------+
| ReplicaId | BackendId | Version | VersionHash | LstSuccessVersion | LstSuccessVersionHash | L
stFailedVersion | LstFailedVersionHash | LstFailedTime | SchemaHash | DataSize | RowCount | St
ate | IsBad | VersionCount | PathHash | MetaUrl | CompactionStatus |
+-----------+-----------+---------+------------+-----------------+---------------------+-----------------+---------
------------+--------------+-----------+----------+----------+--------+-------+-------------+---------------------+
----------+-----------------+
| 43734060 | 10004 | 2 | 0 | -1 | 0 | -1 | 0 | N/A | -1 | 784 | 0 | NORMAL | false | 2 | -8566523878
520798656 | url | url |
| 29502555 | 10002 | 2 | 0 | 2 | 0 | -1 | 0 | N/A | -1 | 784 | 0 | NORMAL | false | 2 | 18858261964
44191611 | url | url |
| 39279319 | 10007 | 2 | 0 | -1 | 0 | -1 | 0 | N/A | -1 | 784 | 0 | NORMAL | false | 2 | 16565086312
94397870 | url | url |
+-----------+-----------+---------+------------+-----------------+---------------------+-----------------+---------
------------+--------------+-----------+----------+----------+--------+-------+-------------+---------------------+
----------+-----------------+
```

上图显示了对应 Tablet 的所有副本情况。这里显示的内容和 SHOW TABLET FROM tbl1; 的内容相同。但这里可以清楚的知道，一个具体的 Tablet 的所有副本的状态。

副本调度任务

1. 查看等待被调度的任务。

   SHOW PROC '/cluster_balance/pending_tablets';

```
+----------+--------+----------------+---------+----------+----------+-------+---------+--------+----------+--------
-+--------------------+--------------------+--------------------+----------+------+------------+--------------+-----
---------------+-----------+--------------------+--------+--------------------+-----------------------------+
| TabletId | Type | Status | State | OrigPrio | DynmPrio | SrcBe | SrcPath | DestBe | DestPath | Ti
meout | Create | LstSched | LstVisit | Finished | Rate | FailedSched | FailedRunning | LstAdjPrio |
```

```
VisibleVer | VisibleVerHash | CmtVer | CmtVerHash | ErrMsg |
+---------+--------+---------------+---------+---------+---------+------+--------+-------+---------+-------
-+-------------------+------------------+------------------+---------+-----+-----------+-------------+-----
--------------+-----------+------------------+-------+------------------+---------------------------+
| 4203036 | REPAIR | REPLICA_MISSING | PENDING | HIGH | LOW | -1 | -1 | -1 | -1 | 0 | 2019-02-2
1 15:00:20 | 2019-02-24 11:18:41 | 2019-02-24 11:18:41 | N/A | N/A | 2 | 0 | 2019-02-21 15:00:
43 | 1 | 0 | 2 | 0 | unable to find source replica |
+---------+--------+---------------+---------+---------+---------+------+--------+-------+---------+-------
-+-------------------+------------------+------------------+---------+-----+-----------+-------------+-----
--------------+-----------+------------------+-------+------------------+---------------------------+
```

各列的具体含义如下：

- TabletId：等待调度的 Tablet 的 ID。一个调度任务只针对一个 Tablet。
- Type：任务类型，可以是 REPAIR（修复）或 BALANCE（均衡）。
- Status：该 Tablet 当前的状态，如 REPLICA_MISSING（副本缺失）。
- State：该调度任务的状态，可能为 PENDING/RUNNING/FINISHED/CANCELLED/TIMEOUT/UNEXPECTED。
- OrigPrio：初始的优先级。
- DynmPrio：当前动态调整后的优先级。
- SrcBe：源端 BE 节点的 ID。
- SrcPath：源端 BE 节点的路径的 hash 值。
- DestBe：目的端 BE 节点的 ID。
- DestPath：目的端 BE 节点的路径的 hash 值。
- Timeout：当任务被调度成功后，这里会显示任务的超时时间，单位：秒。
- Create：任务被创建的时间。
- LstSched：上一次任务被调度的时间。
- LstVisit：上一次任务被访问的时间。这里“被访问”指包括被调度，任务执行汇报等和这个任务相关的被处理的时间点。
- Finished：任务结束时间。
- Rate：clone 任务的数据拷贝速率。
- FailedSched：任务调度失败的次数。
- FailedRunning：任务执行失败的次数。
- LstAdjPrio：上一次优先级调整的时间。
- CmtVer/CmtVerHash/VisibleVer/VisibleVerHash：用于执行 clone 任务的 version 信息。
- ErrMsg：任务被调度和运行过程中，出现的错误信息。

2. 查看正在运行的任务。

```
SHOW PROC '/cluster_balance/running_tablets';
```

其结果中各列的含义和 pending_tablets 相同。

3. 查看已结束任务。

SHOW PROC '/cluster_balance/history_tablets';

我们默认只保留最近 1000 个完成的任务。其结果中各列的含义和 pending_tablets 相同。如果 State 列为 FINISHED，则说明任务正常完成。如果为其他，则可以根据 ErrMsg 列的错误信息查看具体原因。

集群负载及调度资源查看

1. 集群负载。

通过以下命令可以查看集群当前的负载情况：

SHOW PROC '/cluster_balance/cluster_load_stat';

首先看到的是对不同存储介质的划分：

```
+---------------+
| StorageMedium |
+---------------+
| HDD |
| SSD |
+---------------+
```

单击某一种存储介质，可以看到包含该存储介质的 BE 节点的均衡状态：

SHOW PROC '/cluster_balance/cluster_load_stat/HDD';

```
+----------+-----------------+-----------+--------------+---------------+------------+-----------+----------+-------
----+-------------------+-------+
| BeId | Cluster | Available | UsedCapacity | Capacity | UsedPercent | ReplicaNum | CapCoeff | R
eplCoeff | Score | Class |
+----------+-----------------+-----------+--------------+---------------+------------+-----------+----------+-------
----+-------------------+-------+
| 10003 | default_cluster | true | 3477875259079 | 19377459077121 | 17.948 | 493477 | 0.5 |
0.5 | 0.9284678149967587 | MID |
| 10002 | default_cluster | true | 3607326225443 | 19377459077121 | 18.616 | 496928 | 0.5 |
0.5 | 0.948660871419998 | MID |
| 10005 | default_cluster | true | 3523518578241 | 19377459077121 | 18.184 | 545331 | 0.5 |
0.5 | 0.9843539990641831 | MID |
| 10001 | default_cluster | true | 3535547090016 | 19377459077121 | 18.246 | 558067 | 0.5 |
0.5 | 0.9981869446537612 | MID |
| 10006 | default_cluster | true | 3636050364835 | 19377459077121 | 18.764 | 547543 | 0.5 |
```

```
0.5 | 1.0011489897614072 | MID |
| 10004 | default_cluster | true | 3506558163744 | 15501967261697 | 22.620 | 468957 | 0.5 |
0.5 | 1.0228319835582569 | MID |
| 10007 | default_cluster | true | 4036460478905 | 19377459077121 | 20.831 | 551645 | 0.5 |
0.5 | 1.057279369420761 | MID |
| 10000 | default_cluster | true | 4369719923760 | 19377459077121 | 22.551 | 547175 | 0.5 |
0.5 | 1.0964036415787461 | MID |
+----------+-----------------+-----------+--------------+---------------+------------+-----------+----------+-------
----+-------------------+-------+
```

其中一些列的含义如下：

- Available：为 true 表示 BE 心跳正常，且没有处于下线中。
- UsedCapacity：字节，BE 上已使用的磁盘空间大小。
- Capacity：字节，BE 上总的磁盘空间大小。
- UsedPercent：百分比，BE 上的磁盘空间使用率。
- ReplicaNum：BE 上副本数量。
- CapCoeff/ReplCoeff：磁盘空间和副本数的权重系数。
- Score：负载分数。分数越高，负载越重。
- Class：根据负载情况分类，LOW/MID/HIGH。均衡调度会将高负载节点上的副本迁往低负载节点。

用户可以进一步查看某个 BE 上各个路径的使用率，例如 ID 为 10001 这个 BE：

SHOW PROC '/cluster_balance/cluster_load_stat/HDD/10001'; 。

```
+-----------------+-----------------+--------------+--------------+---------+--------+----------------------+
| RootPath | DataUsedCapacity | AvailCapacity | TotalCapacity | UsedPct | State | PathHash |
+-----------------+-----------------+--------------+--------------+---------+--------+----------------------+
| /home/disk4/palo | 498.757 GB | 3.033 TB | 3.525 TB | 13.94 % | ONLINE | 4883406271918338267 |
| /home/disk3/palo | 704.200 GB | 2.832 TB | 3.525 TB | 19.65 % | ONLINE | -5467083960906519443 |
| /home/disk1/palo | 512.833 GB | 3.007 TB | 3.525 TB | 14.69 % | ONLINE | -7733211489989964053 |
| /home/disk2/palo | 881.955 GB | 2.656 TB | 3.525 TB | 24.65 % | ONLINE | 4870995507205544622 |
| /home/disk5/palo | 694.992 GB | 2.842 TB | 3.525 TB | 19.36 % | ONLINE | 1916696897889786739 |
+-----------------+-----------------+--------------+--------------+---------+--------+----------------------+
```

这里显示了指定 BE 上，各个数据路径的磁盘使用率情况。

2. 调度资源

用户可以通过以下命令，查看当前各个节点的 slot 使用情况：

SHOW PROC '/cluster_balance/working_slots'; 。

```
+---------+------------------+-----------+-----------+------------+--------------------+
| BeId | PathHash | AvailSlots | TotalSlots | BalanceSlot | AvgRate |
+---------+------------------+-----------+-----------+------------+--------------------+
| 10000 | 8110346074333016794 | 2 | 2 | 2 | 2.459007474009069E7 |
| 10000 | -5617618290584731137 | 2 | 2 | 2 | 2.4730105014001578E7 |
| 10001 | 4883406271918338267 | 2 | 2 | 2 | 1.6711402709780257E7 |
| 10001 | -5467083960906519443 | 2 | 2 | 2 | 2.7540126380326536E7 |
| 10002 | 9137404661108133814 | 2 | 2 | 2 | 2.417217089806745E7 |
| 10002 | 1885826196444191611 | 2 | 2 | 2 | 1.6327378456676323E7 |
+---------+------------------+-----------+-----------+------------+--------------------+
```

这里以数据路径为粒度，展示了当前 slot 的使用情况。其中 AvgRate 为历史统计的该路径上 clone 任务的拷贝速率，单位是字节/秒。

3. 优先修复查看

以下命令，可以查看通过 ADMIN REPAIR TABLE 命令设置的优先修复的表或分区。

SHOW PROC '/cluster_balance/priority_repair'； 。

其中 RemainingTimeMs 表示，这些优先修复的内容，将在这个时间后，被自动移出优先修复队列。以防止优先修复一直失败导致资源被占用。

调度器统计状态查看

我们收集了 TabletChecker 和 TabletScheduler 在运行过程中的一些统计信息，可以通过以下命令查看：

SHOW PROC '/cluster_balance/sched_stat'； 。

```
+-------------------------------------------------+------------+
| Item | Value |
+-------------------------------------------------+------------+
| num of tablet check round | 12041 |
| cost of tablet check(ms) | 7162342 |
| num of tablet checked in tablet checker | 18793506362 |
| num of unhealthy tablet checked in tablet checker | 7043900 |
| num of tablet being added to tablet scheduler | 1153 |
| num of tablet schedule round | 49538 |
```

```
| cost of tablet schedule(ms) | 49822 |
| num of tablet being scheduled | 4356200 |
| num of tablet being scheduled succeeded | 320 |
| num of tablet being scheduled failed | 4355594 |
| num of tablet being scheduled discard | 286 |
| num of tablet priority upgraded | 0 |
| num of tablet priority downgraded | 1096 |
| num of clone task | 230 |
| num of clone task succeeded | 228 |
| num of clone task failed | 2 |
| num of clone task timeout | 2 |
| num of replica missing error | 4354857 |
| num of replica version missing error | 967 |
| num of replica relocating | 0 |
| num of replica redundant error | 90 |
| num of replica missing in cluster error | 0 |
| num of balance scheduled | 0 |
+------------------------------------------------+------------+
```

各行含义如下：

- num of tablet check round：Tablet Checker 检查次数。
- cost of tablet check(ms)：Tablet Checker 检查总耗时。
- num of tablet checked in tablet checker：Tablet Checker 检查过的 tablet 数量。
- num of unhealthy tablet checked in tablet checker：Tablet Checker 检查过的不健康的 tablet 数量。
- num of tablet being added to tablet scheduler：被提交到 Tablet Scheduler 中的 tablet 数量。
- num of tablet schedule round：Tablet Scheduler 运行次数。
- cost of tablet schedule(ms)：Tablet Scheduler 运行总耗时。
- num of tablet being scheduled：被调度的 Tablet 总数量。
- num of tablet being scheduled succeeded：被成功调度的 Tablet 总数量。
- num of tablet being scheduled failed：调度失败的 Tablet 总数量。
- num of tablet being scheduled discard：调度失败且被抛弃的 Tablet 总数量。
- num of tablet priority upgraded：优先级上调次数。
- num of tablet priority downgraded：优先级下调次数。
- num of clone task：生成的 clone 任务数量。
- num of clone task succeeded：clone 任务成功的数量。
- num of clone task failed：clone 任务失败的数量。
- num of clone task timeout：clone 任务超时的数量。

- num of replica missing error：检查的状态为副本缺失的 tablet 的数量。
- num of replica version missing error：检查的状态为版本缺失的 tablet 的数量（该统计值包括了 num of replica relocating 和 num of replica missing in cluster error）。
- num of replica relocating：检查的状态为 replica relocating 的 tablet 的数量。
- num of replica redundant error：检查的状态为副本冗余的 tablet 的数量。
- num of replica missing in cluster error：检查的状态为不在对应 cluster 的 tablet 的数量。
- num of balance scheduled：均衡调度的次数。

> 说明：
> 以上状态都只是历史累加值。我们也在 FE 的日志中，定期打印了这些统计信息，其中括号内的数值表示自上次统计信息打印依赖，各个统计值的变化数量。

相关配置说明

可调整参数

以下可调整参数均为 fe.conf 中可配置参数。

- use_new_tablet_scheduler
  - 说明：是否启用新的副本调度方式。新的副本调度方式即本文档介绍的副本调度方式。
  - 默认值：true。
  - 重要性：高。
- tablet_repair_delay_factor_second
  - 说明：对于不同的调度优先级，我们会延迟不同的时间后开始修复。以防止因为例行重启、升级等过程中，产生大量不必要的副本修复任务。此参数为一个基准系数。对于 HIGH 优先级，延迟为 基准系数 * 1；对于 NORMAL 优先级，延迟为 基准系数 * 2；对于 LOW 优先级，延迟为 基准系数 * 3。即优先级越低，延迟等待时间越长。如果用户想尽快修复副本，可以适当降低该参数。
  - 默认值：60秒。
  - 重要性：高。
- schedule_slot_num_per_path
  - 说明：默认分配给每块磁盘用于副本修复的 slot 数目。该数目表示一块磁盘能同时运行的副本修复任务数。如果想以更快的速度修复副本，可以适当调高这个参数。单数值越高，可能对 IO 影响越大。
  - 默认值：2。
  - 重要性：高。
- balance_load_score_threshold

- 说明：集群均衡的阈值。默认为 0.1，即 10%。当一个 BE 节点的 load score，不高于或不低于平均 load score 的 10% 时，我们认为这个节点是均衡的。如果想让集群负载更加平均，可以适当调低这个参数。
- 默认值：0.1。
- 重要性：中。

- storage_high_watermark_usage_percent 和 storage_min_left_capacity_bytes
  - 说明：这两个参数，分别表示一个磁盘的最大空间使用率上限，以及最小的空间剩余下限。当一块磁盘的空间使用率大于上限，或者剩余空间小于下限时，该磁盘将不再作为均衡调度的目的地址。
  - 默认值：0.85 和 1048576000（1GB）。
  - 重要性：中。

- disable_balance
  - 说明：控制是否关闭均衡功能。当副本处于均衡过程中时，有些功能，如 ALTER TABLE 等将会被禁止。而均衡可能持续很长时间。因此，如果用户希望尽快进行被禁止的操作。可以将该参数设为 true，以关闭均衡调度。
  - 默认值：false。
  - 重要性：中。

不可调整参数

以下参数暂不支持修改，仅作说明。

- TabletChecker 调度间隔
  TabletChecker 每20秒进行一次检查调度。

- TabletScheduler 调度间隔
  TabletScheduler 每5秒进行一次调度。

- TabletScheduler 每批次调度个数
  TabletScheduler 每次调度最多 50 个 tablet。

- TabletScheduler 最大等待调度和运行中任务数
  最大等待调度任务数和运行中任务数为 2000。当超过 2000 后，TabletChecker 将不再产生新的调度任务给 TabletScheduler。

- TabletScheduler 最大均衡任务数
  最大均衡任务数为 500。当超过 500 后，将不再产生新的均衡任务。

- 每块磁盘用于均衡任务的 slot 数目
  每块磁盘用于均衡任务的 slot 数目为2。这个 slot 独立于用于副本修复的 slot。

- 集群均衡情况更新间隔
  TabletScheduler 每隔 20 秒会重新计算一次集群的 load score。

- Clone 任务的最小和最大超时时间
  一个 clone 任务超时时间范围是 3min ~ 2hour。具体超时时间通过 tablet 的大小计算。计算公式为 (tablet size) / (5MB/s)。当一个 clone 任务运行失败 3 次后，该任务将终止。

- 动态优先级调整策略
  优先级最小调整间隔为 5min。当一个 tablet 调度失败5次后，会调低优先级。当一个 tablet 30min 未被调度时，会调高优先级。

相关问题

- 在某些情况下，默认的副本修复和均衡策略可能会导致网络被打满（多发生在千兆网卡，且每台 BE 的磁盘数量较多的情况下）。此时需要调整一些参数来减少同时进行的均衡和修复任务数。
- 目前针对 Colocate Table 的副本的均衡策略无法保证同一个 Tablet 的副本不会分布在同一个 host 的 BE 上。但 Colocate Table 的副本的修复策略会检测到这种分布错误并校正。但可能会出现，校正后，均衡策略再次认为副本不均衡而重新均衡。从而导致在两种状态间不停交替，无法使 Colocate Group 达成稳定。针对这种情况，我们建议在使用 Colocate 属性时，尽量保证集群是同构的，以减小副本分布在同一个 host 上的概率。