

KYLIN-5317 优化获取 Working Dir 容量的计算方式

Dev Design

问题描述

Kylin 产生大量对 HDFS Namenode 的请求，导致 Namenode 服务性能低下。从而影响整个HDFS集群的性能和稳定性。

背景

用户大数据平台出具报告发现 Kylin 启动后，整体大数据集群 HDFS RPC 相应时间延长，因此判断是 Kylin 影响了整个大数据集群的稳定运行，经过排查，确认是由于 Kylin 侧的行为导致，主要涉及功能“存储配额”、“prometheus监控”、“influxDB监控”等。

添加如下配置后平均 RPC 队列响应时间基本稳定：

```
1 kylin.metrics.prometheus-enabled=false
2 kylin.job.check-quota-storage-enabled=false
3 kylin.metrics.polling-interval-secs=86400
4 kylin.source.record-source-usage-enabled=false
5 kylin.source.hive.database-access-filter-enabled=false
```

实现方案

- 新增统一获取 Working Dir 容量计算方式的功能开关，默认开启。
- 新增定时线程池（额外的命名以区分线程日志），对外暴露可配置的轮询时间参数，默认 5min/次。
- 在内存中维护已有目录的 ContentSummary 信息，超过一定时间或者内存定额大小则失效
- 所有通过定时线程获取 Working Dir 容量的地方都接入此逻辑处理，仅临时获取一次的地方不做处理，如构建任务涉及到的目录等。
- 仅有一个 leader 节点进行维护操作，其他节点读取即可。

Comment Reply

关于在内存中维护的信息 考虑存储在哪，元数据或者文件系统上，不要所有的 Kylin 节点都去计算 WorkingDir 的大小

- 可以肯定的是，每个节点的 Kylin 内存中都会维护一份数据
 - 仅在内存中维护
 - 使用时不需要进行判断（即不用从其他存储介质进行文件读取访问等操作），相对较快
 - 每个节点都需要计算一次重复的 WorkingDir 大小，Kylin 服务重启后数据失效
 - 在内存和文件系统中维护

- 使用时需要进行判断，当内存中未找到数据时，需要对额外的文件进行读取，可能存在多次打开/关闭文件读取的操作，另外还需要考虑文件的清理逻辑（按照时间或者大小），不会伴随 Kylin 服务重启数据失效，而是随着文件清理数据失效
- 本地文件系统
 - 内存只维护一部分数据，定时或者定量覆写在本地文件，因为是同节点，操作很快
 - 由于是本地文件系统，所以各个 Kylin 节点还需要额外的数据同步机制（比如 RPC 机制）以避免重复目录多次发起请求
- 分布式文件系统 HDFS
 - 内存只维护一部分数据，定时或者定量覆写在分布式文件系统上即可，但会对分布式文件系统有额外的负载，需要评估验证读写的压力和直接通过 Hadoop 的 API 获取 `ContentSummary` 的压力是否值得
- 在内存和元数据库中维护
 - 使用时需要进行判断，当内存中未找到数据时，需要到元数据库进行查询，对元数据库有额外的连接查询操作
 - 内存只维护一部分数据，定时或者定量覆写到元数据库中，每个 Kylin 节点可直接通过连接元数据库进行读取，不需要额外的数据同步机制
 - 还需要考虑数据在元数据库中的大小，定时或者定量进行数据的清理操作
 - 不会伴随 Kylin 服务重启数据失效，而是随着元数据库的清理数据失效

考虑到放在 HDFS 上的一次读写负载压力不是很大，从实现的角度来说选择存在 HDFS 上更容易。

说明

- 对外没有接口变更，原先交互行为不变。
- Working Dir 容量的计算方式在时效性上有影响，这取决于对外提供的可配置时间参数，时间越短准确性越高，但 NameNode 的压力也越大；时间越长准确性越低，NameNode 压力就越小。
- 重点关注 `org.apache.hadoop.fs.FileSystem#getContentSummary` 方法
- 确认逻辑影响：
 - a. Kylin 存储配额信息的 `TotalStorageCollector` 会调用 `HadoopUtil.getContentSummary` 以获取存储大小，有参数配置，默认为 30s/次
 - b. `Register project metrics` 每 1min/次 会按照 project 去计算一次 Working Dir 大小
 - c. `org.apache.kylin.rest.config.initialize.MetricsRegistry#refreshTotalStorageSize` 方法会每 10min/次 也刷新统一一次的数据