

调优 | Apache Hudi应用调优指南

Original hudi ApacheHudi 6/6

来自专辑

Hudi实战系列

通过Spark作业将数据写入Hudi时，Spark应用的调优技巧也适用于此。如果要提高性能或可靠性，请牢记以下几点。

输入并行性：Hudi对输入进行分区默认并发度为1500，以确保每个Spark分区都在2GB的限制内（在Spark2.4.0版本之后去除了该限制），如果有更大的输入，则相应地进行调整。我们建议设置shuffle的并发度，配置项为 `hoodie.[insert|upsert|bulkinsert].shuffle.parallelism`，以使其至少达到`inputdatasize/500MB`。

Off-heap（堆外）内存：Hudi写入parquet文件，需要使用一定的堆外内存，如果遇到此类故障，请考虑设置类似 `spark.yarn.executor.memoryOverhead` 或 `spark.yarn.driver.memoryOverhead` 的值。

Spark 内存：通常Hudi需要能够将单个文件读入内存以执行合并或压缩操作，因此执行程序的内存应足以容纳此文件。另外，Hudi会缓存输入数据以便能够智能地放置数据，因此预留一些 `spark.memory.storageFraction` 通常有助于提高性能。

调整文件大小：设置 `limitFileSize` 以平衡接收/写入延迟与文件数量，并平衡与文件数据相关的元数据开销。

时间序列/日志数据：对于单条记录较大的数据库/ nosql变更日志，可调整默认配置。另一类非常流行的数据是时间序列/事件/日志数据，它往往更加庞大，每个分区的记录更多。在这种情况下，请考虑通过 `.bloomFilterFPP()` / `bloomFilterNumEntries()` 来调整Bloom过滤器的精度，以加速目标索引查找时间，另外可考虑一个以事件时间为前缀的键，这将使用范围修剪并显着加快索引查找的速度。

GC调优：请确保遵循Spark调优指南中的垃圾收集调优技巧，以避免OutOfMemory错误。[必须]使用G1 / CMS收集器，其中添加到`spark.executor.extraJavaOptions`的示例如下：

```

1.-XX:New-Size=1g-XX:SurvivorRatio=2-XX:+UseCompressedOops-XX:+UseConcMarkSweepGC-XX:+UseParallelNewGC-XX:CMSInitiatingOccupancyFraction=70-XX:+PrintGC

```

OutOfMemory错误：如果出现OOM错误，则可尝试通过如下配置处理：spark.memory.fraction = 0.2，spark.memory.storageFraction = 0.2 允许其溢出而不是OOM（速度变慢与间歇性崩溃相比）。

以下是完整的生产配置

```

spark.driver.extraClassPath /etc/hive/conf
spark.driver.extraJavaOptions -XX:+PrintTenuringDistribution -XX:+PrintGCDetails -
XX:+PrintGCDateStamps -XX:+PrintGCApplicationStoppedTime -XX:+PrintGCApplicationConcurrentTime -XX:+PrintGCTimeStamps -XX:+HeapDumpOnOutOfMemoryError -
XX:HeapDumpPath=/tmp/hoodie-heapdump.hprof
spark.driver.maxResultSize 2g
spark.driver.memory 4g
spark.executor.cores 1
spark.executor.extraJavaOptions -XX:+PrintFlagsFinal -XX:+PrintReferenceGC -
verbose:gc -XX:+PrintGCDetails -XX:+PrintGCTimeStamps -XX:+PrintAdaptiveSizePolicy -XX:+UnlockDiagnosticVMOptions -XX:+HeapDumpOnOutOfMemoryError -XX:Heap-
DumpPath=/tmp/hoodie-heapdump.hprof
spark.executor.id driver
spark.executor.instances 300
spark.executor.memory 6g
spark.rdd.compress true

spark.kryoserializer.buffer.max 512m
spark.serializer org.apache.spark.serializer.KryoSerializer
spark.shuffle.service.enabled true
spark.sql.hive.convertMetastoreParquet false

```

spark.submit.deployMode cluster

spark.task.cpus 1

spark.task.maxFailures 4

spark.yarn.driver.memoryOverhead 1024

spark.yarn.executor.memoryOverhead 3072

spark.yarn.max.executor.failures 100

猜你喜欢

官宣! ASF官方正式宣布Apache Hudi成为顶级项目

刚刚晋升为 Apache 顶级项目的 Hudi 如何在数据湖上玩转增量处理

实战! 配置DataDog监控Apache Hudi应用指标

填坑! 线上Presto查询Hudi表异常排查

Apache Hudi典型应用场景知多少?





ApacheHudi 技术交流 2 群

扫一扫二维码，加入该群。



Read more