360

www.360.cn

HBase在搜索网页库上的应用

赵健博
QIHU 360 系统部
zhaojianbo@360.cn

内容梗概

- Why HBase
- 集群规模与版本
- 问题与改进
- Future Work
- 运维与监控

内容梗概

- Why Hbase
- 集群规模与版本
- 问题与改进
- Future Work
- 运维与监控

- 数据规模巨大
  - 记录数：千亿级别
  - 数据量：PB级别

- 网页多个版本支持
- 高可扩展、高可靠
- M/R支持

- 更新需求
  - 海量数据导入（TB级别）
  - 灵活地增加与修改属性

- 扫描，查询需求：
  - 按列读取
  - 按站点扫描
  - 批量读取
  - 时间范围查询

- Why Hbase
- 集群规模与版本
- 问题与改进
- Future Work
- 运维与监控

- 机器规模
  - HBase：300节点
    - region个数 > 10万
- 平台版本：
  - HBase版本：facebook 0.89-fb
  - HDFS版本： facebook hadoop-20

- Why Hbase
- 集群规模与版本
- 问题与改进
- Future Work
- 运维与监控

- 数据导入方面（4）
- Compaction方面（3）
- Split方面（3）
- 异常恢复（1）

- 问题一：
  - 调用Put接口写入数据，写入性能不高效。
- 原因：
  - 写路径上commitlog的写与sync过程持锁进行IO操作，阻塞并发写线程，不够高效
  - Compaction与写commitlog占用额外的磁盘与网络资源
- 改进：
  - 采用bulkImport方式导入数据，效率极大提升

- 问题二：
  - bulkImport的数据准备阶段对输入文件格式的处理不够通用
- 原因：
  - bulkImport的数据准备阶段程序对输入文件格式所有限制，不能够满足我们需求
- 改进：
  - 提供了通用的数据格式解析框架，适配各种输入格式

- 问题三：
  - bulkImport的数据准备阶段，当region数很大（>10万）时，数据准备阶段时间较长
- 原因：
  - 大量reduce从map中shuffle很少数据，或者甚至没有数据，导致整体shuffle过程低效。
- 改进：
  - 修改了partition与reduce的逻辑，使得一个reduce可以生成多个region的数据
- 效果比较：
  - 10TB数据，shuffle时间消耗：5小时 => 1小时

- 问题四：
  - bulkImport的数据导入阶段较慢
- 原因：
  - bulkImport的数据导入阶段，是单进程串行进行
- 改进：
  - MR版本的数据导入程序，并发了数据导入过程
- 效果比较：
  - 60万文件规模：2~3小时 => 30分钟

- 问题五：
  - bulkImport后，compaction操作会产生大量IO
- 原因：
  - compaction的文件选择算法对bulkImport后的文件支持不好，可能会选择到大文件，从而产生大量IO
- 改进：
  - 手动触发compaction新接口，可选择文件大小范围，时间范围，以及文件个数
  - 提供自动minor compaction的开关，可将其关闭

- 问题六：
  - Compaction的并发调整需要重启regionserver，代价较高
- 原因：
  - Regionserver启动时读取配置，后续不可更改
- 改进：
  - 将compaction并发参数的设置功能通过http服务的方式提供。

- 问题七：
  - 目前compaction的并发可以控制，但是单个compaction线程的执行速度却没法控制
- 原因：
  - 代码尚未实现单并发限速功能
- 改进：
  - 在compaction路径上增加限速功能，提供参数调整接口，可通过http方式动态更改

- 问题八：
  - 多CF(ColumnFamily)的表，可能出现带有引用文件的region也能够被分裂的情况，从而导致该region不能被正常打开
- 原因：
  - Region Split时，仅仅对第一个CF进行了检测
- 改进：
  - Region Split时，检测region中任何一个CF中存有引用文件，则禁止分裂该region

- 问题九：
  - 多CF的Region Split后，两个daughter的数据不均匀
- 原因：
  - Region Split时只根据第一个CF的分裂点进行分裂
- 改进：
  - Region Split时，选择数据量最大的那个CF的分裂点进行分裂

- 问题十：
  - 随着region个数增加，触发region split，时间变长
- 原因：
  - 触发region split接口是通过扫描一次meta来判断split的目标是region还是table。
- 改进：
  - 提供splitRegion接口（跳过扫描meta表）
- 效果比较：
  - 6~7s一个 => 6ms一个

- 问题十一：
  - Meta表到一定规模后（Region > 10万），RS异常宕掉后，Master触发的恢复时间较长
- 原因：
  - Scan meta表，查找异常region的过程效率低下，消耗了大量时间
- 改进：
  - 使用caching模式，扫描meta表
- 效果比较：
  - 20~25分钟 => 2~3分钟

内容梗概

- Why Hbase
- 集群规模
- 问题与改进
- Future Work
- 运维与监控

Future work

- 优化减少Hbase集群的启停时间
- 进一步优化减少RS异常退出后的恢复时间

- Why Hbase
- 集群规模
- 问题与改进
- Future Work
- 运维与监控

- Compaction & Split每天手动触发
- Compaction日常统计
- Region信息统计报表
- Region健康状态监控
- Meta表健康状态监控
- 应用级别监控

Thanks !