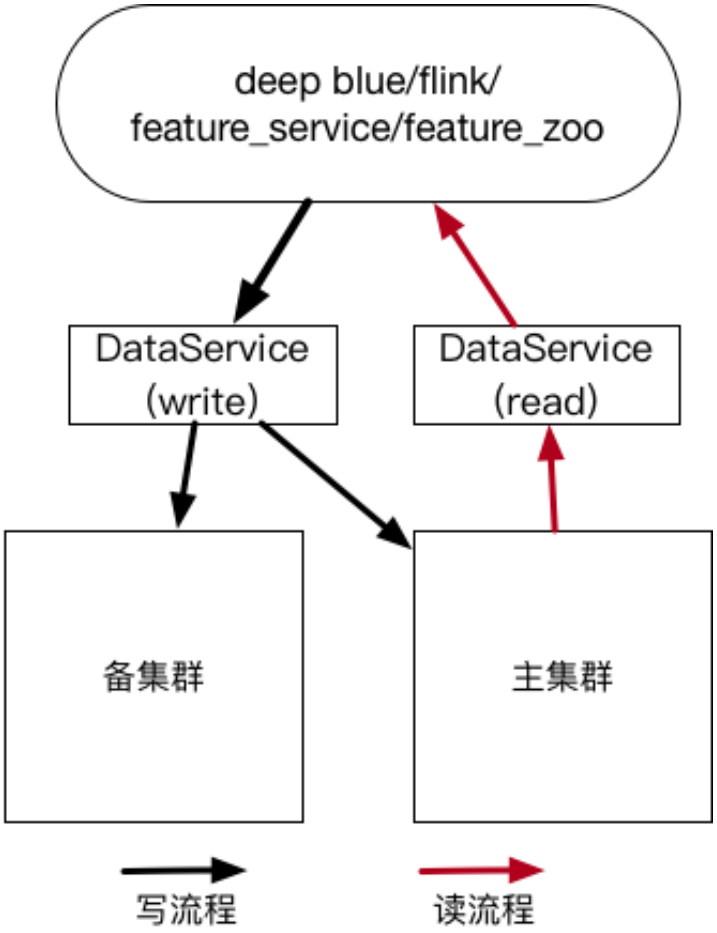
HBase备份方案

玄石 2017-04-10

方案一

方案一如下图所示



方案简介

- 所有访问Hbase的系统，包括flink系统入库，都经过DataService访问，包括读和写；
- DataService读写分离；
    - DataService的写，涉及Hbase主备两个集群，同步写，意即：当主备集群都写入成功，才算写入成功，并返回ok，一旦有一个写入失败，即为写入失败，要么重试，要么报错；
    - DataService的读，涉及Hbase一个集群即可，要么主集群，要么备集群；
- Hbase集群切换通过DataService实现：DataService更换配置后，一台一台启动，实现主备集群无缝切换；

特点

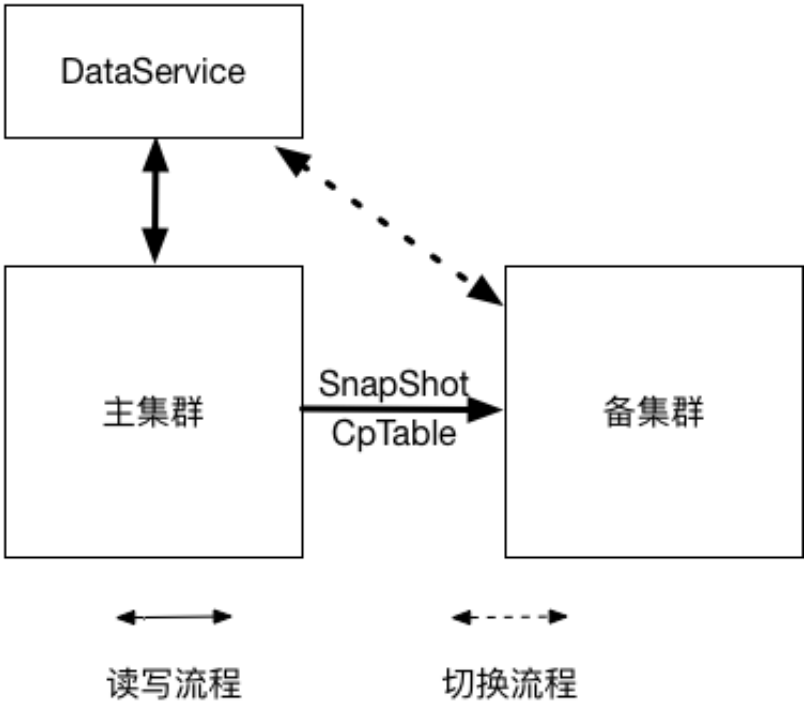
优势

1. 所有访问Hbase的系统，收拢到DataService，在DataService实现权限控制和访问；
2. 通过DataService同步写实现主备集群的数据一致性；
3. DataService做到无状态(数据读写无状态)；
4. 单台物理机上部署多套DataService，通过Docker实现多套部署；

问题

- 会增加读写overhead，因此，对DataService的性能要求较高，需要把DataService的overhead优化到50ms以内；
- 数据一致性通过DataService的同步写实现，可能会阻塞DataService(一个集群出现问题，就写不成功，需要手动干预，从DataService中删除出问题的集群)；
- dataservice实现和部署都要复杂；

方案二



方案简介

- 访问Hbase的系统，通过DataService实现，DataService只读写主集群，主备集群之间的同步通过Hbase本身的replication机制实现；
- 当主集群出现问题，从集群通过DataService实现切换；

特点

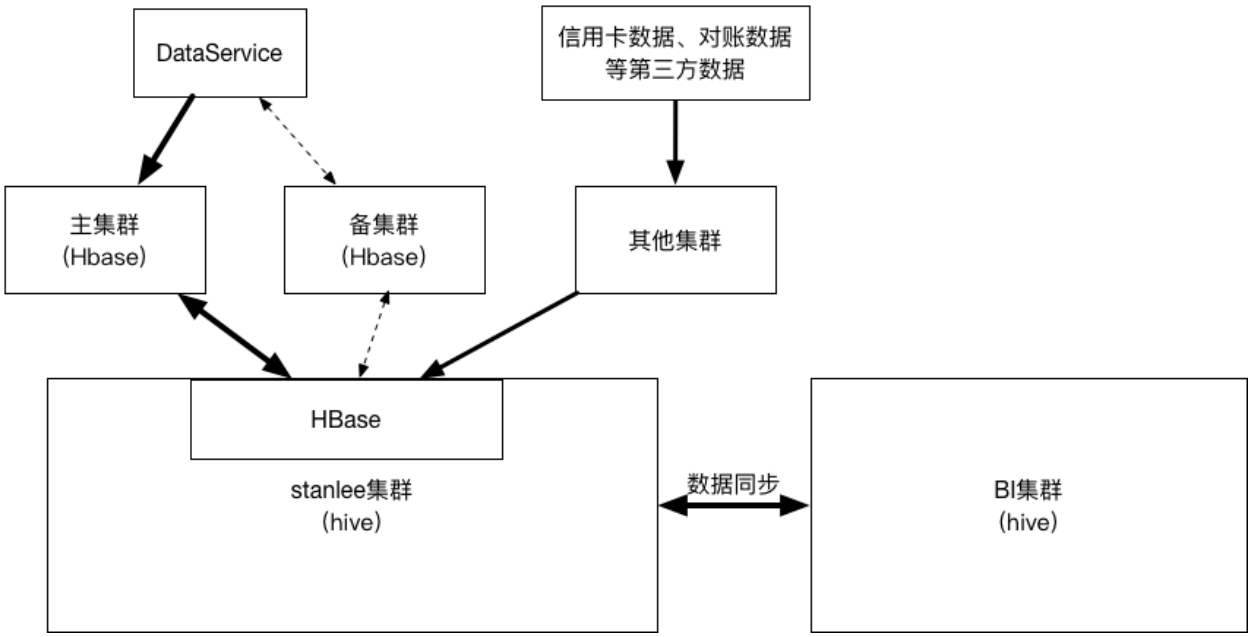
优势

- 应用层只看到一个Hbase，不会增加写两个集群的overhead；
- 应用层部署和实现跟现有系统保持一致；
- 使用Hbase底层的技术实现数据replication，并达到最终一致；

问题

- 主库数据出现问题，备库也会有问题，在方案一中也存在此问题；
- 切换到备库后，主库需要从备库同步数据，需要切换主备角色（主库变备库，备库变主库)；

方案三



方案简介

方案3其实不是独立的方案，写在这里是因为后续Hbase，hive以及BI的hive集群后续会实现数据互通和备份。
因为目前有这样的需求：

- 算法特征数据开发在BI集群，需要推送到线上的Hbase集群；
- 在线Hbase集群不重要的业务入库和查询从主Hbase集群剥离出来，但是最终还是需要在hive集群进行分析；

上图中，在存储中心(Hive)部署Hbase，好处如下：

- 将主Hbase集群的数据，备份Hbase集群的数据，其他Hbase集群(入库信用卡数据等)的数据，都通过Hbase底层的备份机制收集到存储中心(Hive集群)；
- BI或者数仓团队的特征需要推送到线上(备份集群)，先在存储中心的Hbase生成相应的Hbase表，开启同步功能或者手动拷贝hfile到在线(备份)集群，bulkload到线上(备份)集群的Hbase，实现上线；
- hbase2hive工具可以下线了，通过hive建立存储中心的Hbase桥接表，实现查询Hbase数据，不影响在线集群；

方案总结

- DataService承接所有Hbase的读写访问，需要对DataService进行优化；
- Hbase的replication机制一定要用起来，实现数据互备；