# 副本恢复之消息路由表算法

**背景：**

在设计brfs时，为了能提高分布式的最大优势，故设计各个server之间是对等的关系。也就是每个server都是独立的，并且能够共同存储和读取数据。

因此提出了serverId的概念，即服务标识。serverId可以让我们知道那些数据存在那些server中，而返回给客户的fid中就具有serverId。客户端可以通过简单的解析fid，就能够快速的定位到相应的server中读取文件。

Brfs是支持副本数的，若副本数大于1，那么同样的副本，会存放在不同的server中的。

如果某个server挂掉，那么之前位于这个server中的数据，就会发生副本迁移。迁移之后，还得保证迁移的数据能够正常的读取。此时我们使用一种消息路由表算法来进行数据迁移和数据读取。

**原理：**

在讲路由表算法之前，我们可以先听一个生活中经常遇到的事。

假如某公司目前有A,B,C,D四个员工，这四个员工分别持有各种不同的消息。某一天，A员工要离职，那么公司会让A员工把自己所知的消息告诉其他人。因为A持有的消息可能比较多，所以只告诉其中的一个人，那么他的工作压力会变大。因此，公司会通过消息的特征，将A知道的消息尽可能的均匀的告诉其他的三个人。公司只需要记录接手A消息的人，即[A->B,C,D]。此时，若来了两个新员工E,F。过了不久，C也离职了。那么C同样会把消息尽可能的均匀的告诉其他人，公司同样需要记录接手C消息的人。即，[C->B,D,E,F]。

若某一天，客户来找A询问一个消息，但是A已经离职了，那么公司会通过[A->B,C,D]以及消息特征来寻找接手的人。若该消息命中B，因B还在，所以B可以为客户服务。若命中为C，因C也已离职，则继续通过[C->B,D,E,F]以及消息特征来寻找接手的人。若此时命中的是E，因E存在，则由E来为客户进行服务。

在上面的故事中，我们知道接手记录是非常重要的。因为每一次交接，都会记录一次交接记录，方便根据不同的消息特征，来分配不同的人来接手。后面在找由那一个人接手的具体任务时，也有据可循。

上文说的接手记录或交接记录就是我们关系的路由表。

假设有有序s1-s6个节点，有副本xxx\_s1\_s2。当s1挂掉，则由s2来协助副本恢复。迁移记录1[s1->s2,s3,s4,s5,s6]，由于s2已经保存副本，故排除s2，有[s1->s3,s4,s5,s6]，假设迁移到s3上，则s3有副本xxx\_s1\_s2。我们可以通过规则[s1->s2,s3,s4,s5,s6]将查询s1上xxx\_s1\_s2的副本，都可以导航到该s3节点上。假设又新启动了s7节点，此时s2节点挂掉。则有迁移记录2[s2->s3,s4,s5,s6,s7]。根据迁移记录1，我们可以知道s3上已经有了副本，则副本xxx\_s1\_s2只能迁移到[s4,s5,s6,s7]，假设迁移到s4上。则s3,s4上有副本xxx\_s1\_s2。查询s1上的该副本会根据迁移记录1导航到s3上，查询s2上的该副本会根据迁移记录2导航到s4上。此时，若s3挂掉，有迁移记录3[s3->s4,s5,s6,s7]根据迁移记录1，可知s3上有需要恢复的xxx\_s1\_s2副本，根据迁移记录2，可知由s4来恢复。则xxx\_s1\_s2可恢复在[s5,s6,s7]上，此时选择s5来恢复。则s4和s5上有副本xxx\_s1\_s2。查找s1上的该副本，可通过迁移记录1和迁移记录3来查找，查找s2上的该副本，可通过迁移记录2来查找。以此类推。

优点：整体来看，每次在发生故障迁移的时候就会记录一条迁移规则，也就是说发生1000次迁移，只会产生一千条迁移记录。最差需要计算计算1000次（该条数据每次迁移的机器都挂掉，发生概率非常低），最好的性能为计算1次，时间越靠前的查找次数越低。对于新入的正常数据，查询都是一次性命中。只记录迁移记录，节省了大量的空间来记录每份文件的迁移关系。而对于新入的数据，都是一次性命中。

**测试：**

副本数为2，单线程查询10000个文件，主从随机：

|  |  |  |
| --- | --- | --- |
| 平衡次数 | 总共用时（ms） | 平均每条查询用时（ms） |
| 0 | 74 | 0.0074 |
| 100 | 381 | 0.0381 |
| 200 | 546 | 0.0546 |
| 300 | 837 | 0.0837 |
| 400 | 1194 | 0.1194 |
| 500 | 1638 | 0.1638 |

从上面的数据可以看出，当无故障迁移的时候，查询是非常快的，查询10000条记录只需要74毫秒！平均每条记录只需要0.0074毫秒！当故障迁移发生500次的时候，即这10000条数据迁移了500次，即总共产生了500条迁移记录。此时查询10000条数据也仅仅需要1.638秒！平均每条记录查询只需要0.1638毫秒！我们可以想象一个集群能故障迁移500次也是使用了很久很久了。而数据随着时间的变长，数据也会逐渐变为冷数据，不是被删除掉就是很少访问的数据！因此以目前的查询效率来看，对brfs来说，也是十分适合的！