数据不均衡

前言

- 数据不均衡：顾名思义即我们的数据集样本类别极不均衡

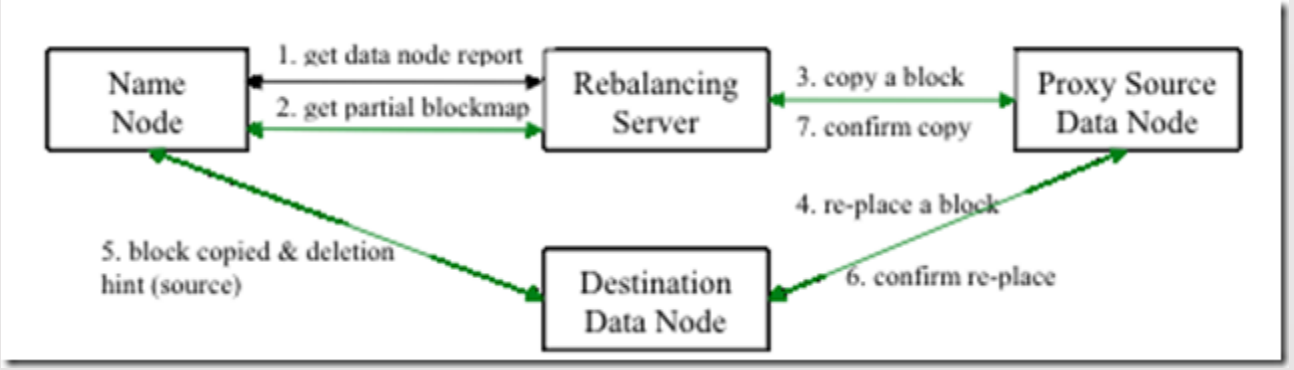
- 相信对于广大的Hadoop集群的使用者和维护者，集群在长时间的使用过程中，肯定或多或少碰到节点间数据不均衡的现象。比如有些节点可能磁盘使用率已经达到90%，而有些节点可能就10%。

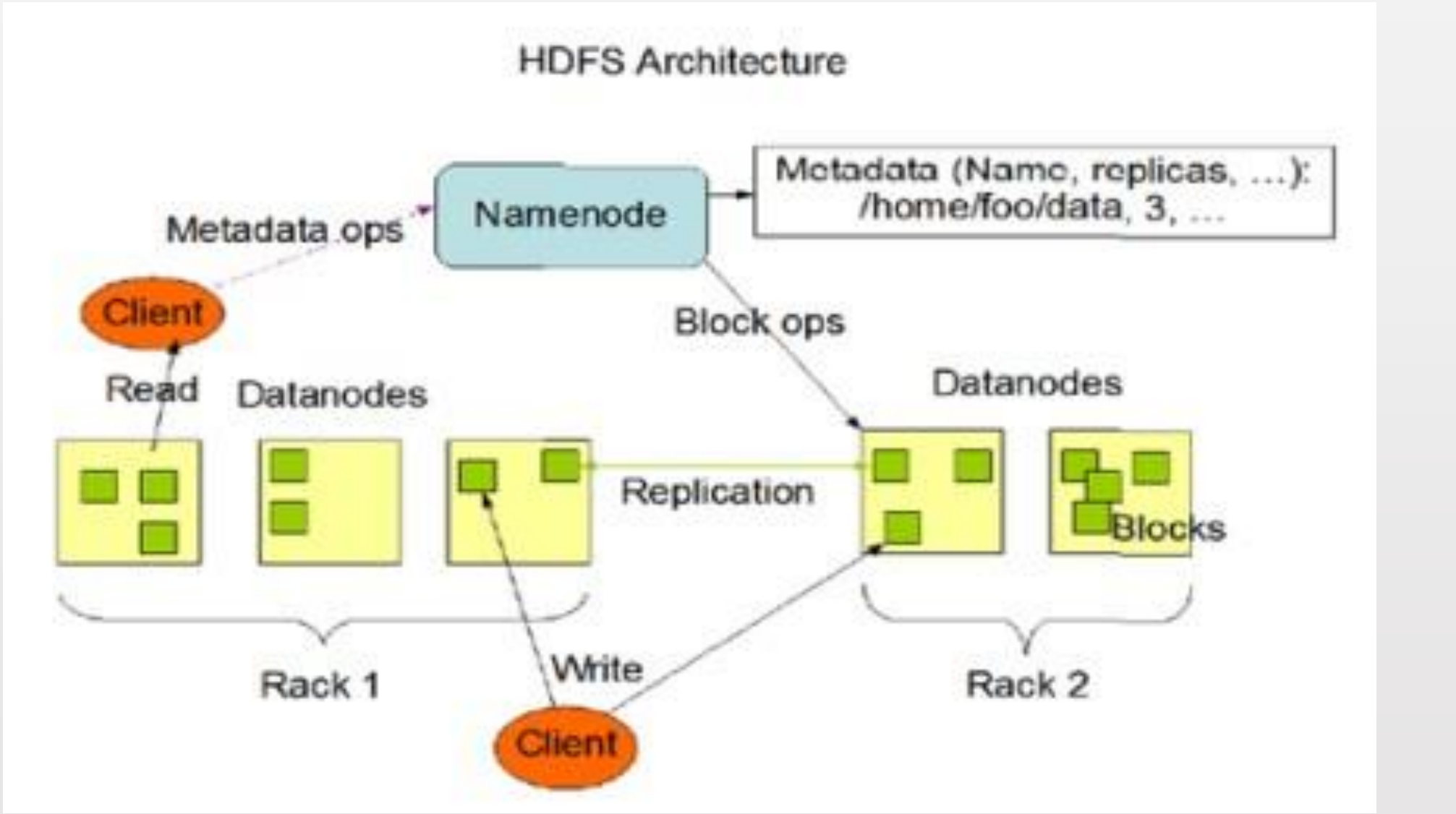
DataNode节点磁盘数据不均衡带来的问题

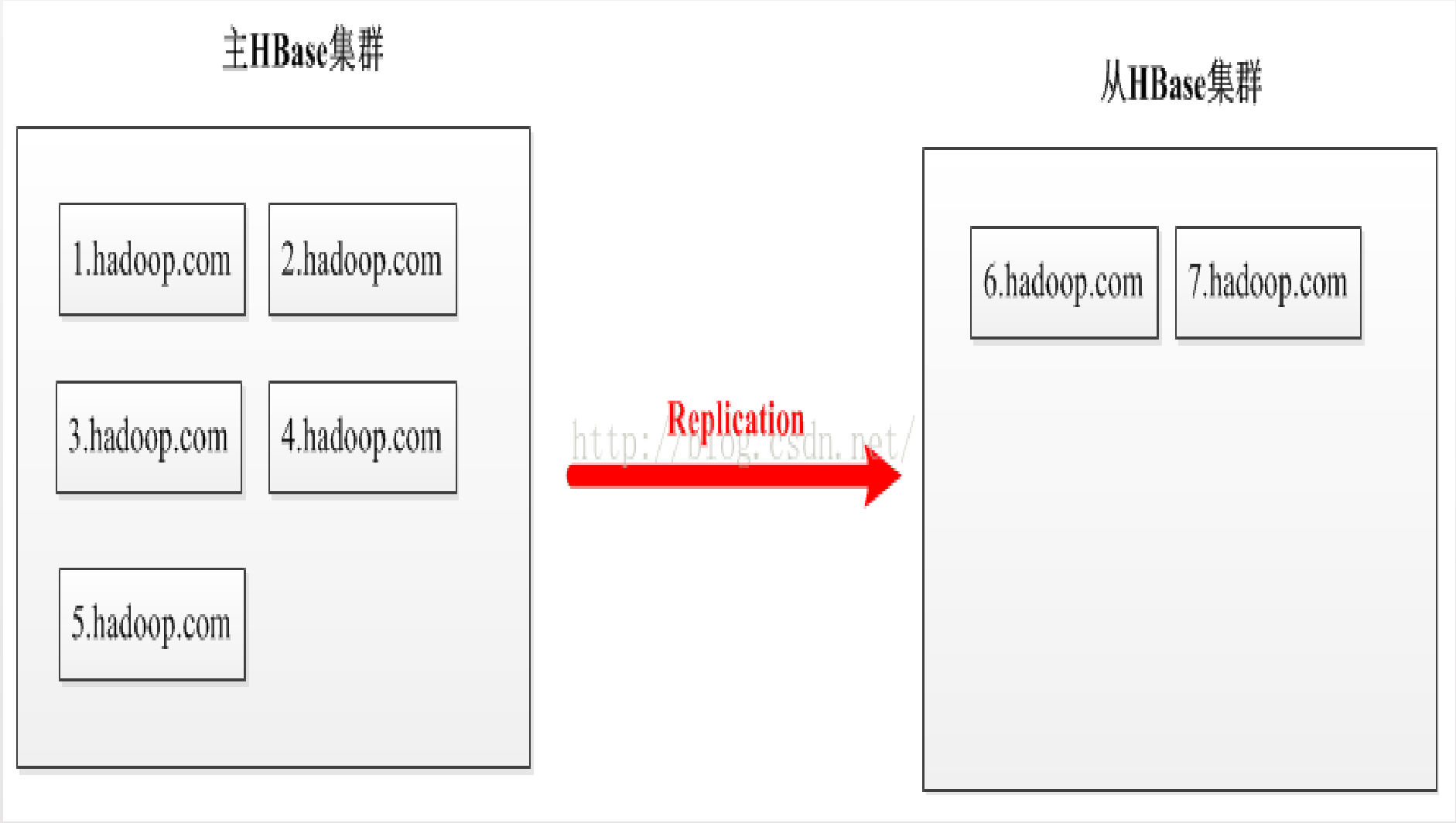
- 磁盘间数据不均衡间接导致磁盘I/O压力不同。我们知道，HDFS上的数据访问频率是很高的，这就涉及到大量的读写磁盘操作，数据多的磁盘自然就会有更高的频率访问，如果一块磁盘的IO操作非常密集的话，势必会对读写性能造成影响
- 高使用率的磁盘导致节点写数据块的时候，可选的存储目录减少。
- HDFS在写block的时候，会挑选剩余空间满足待写的block的大小情况下，才会进行挑选，如果高使用率磁盘目录过多，会导致这样的候选变少。

Balancer

- 1 Rebalance Server从Name Node中获取所有的Data Node情况：每一个Data Node磁盘使用情况。
- 2 Rebalance Server计算哪些机器需要将数据移动，哪些机器可以接受移动的数据。并且从Name Node中获取需要移动的数据分布情况。（按使用占比）
- 3 Rebalance Server计算出来可以将哪一台机器的block移动到另一台机器中去。
- 4,5,6 需要移动block的机器将数据移动的目的机器上去，同时删除自己机器上的block数据。
- 7 Rebalance Server获取到本次数据移动的执行结果，并继续执行这个过程，一直没有数据可以移动或者HDFS集群以及达到了平衡的标准为止




HDFS Architecture
Namenode
Metadata (Name, replicas, ...):
/home/foo/data, 3, ...
Metadata ops
Client
Block ops
Read
Datanodes
Datanodes
Replication
Blocks
Rack 1
Write
Rack 2
Client


主HBase集群
1.hadoop.com
2.hadoop.com
3.hadoop.com
4.hadoop.com
5.hadoop.com
Replication
从HBase集群
6.hadoop.com
7.hadoop.com

Metadata

- 元数据被定义为：描述数据的数据，对数据及信息资源的描述性信息。
- 元数据（Metadata）是描述其它数据的数据（data about other data），或者说是用于提供某种资源的有关信息的结构数据（structured data）。元数据是描述信息资源或数据等对象的数据，其使用目的在于：识别资源；评价资源；追踪资源在使用过程中的变化；实现简单高效地管理大量网络化数据；实现信息资源的有效发现、查找、一体化组织和对使用资源的有效管理。 元数据的基本特点主要有：
- a）元数据一经建立，便可共享。元数据的结构和完整性依赖于信息资源的价值和使用环境；元数据的开发与利用环境往往是一个变化的分布式环境；任何一种格式都不可能完全满足不同团体的不同需要；
- b）元数据首先是一种编码体系。元数据是用来描述数字化信息资源，特别是网络信息资源的编码体系，这导致了元数据和传统数据编码体系的根本区别；元数据的最为重要的特征和功能是为数字化信息资源建立一种机器可理解框架。

balancer解决缺点

- 当然我们说，在使用百分比明细不均衡的情况下，我们可以用HDFS提供的Balancer工具帮我们解决这个问题。但是这不能解决所有的情况，比如说存在异构节点的集群。举一个简单的例子，集群内2个节点：A节点磁盘容量100T,B节点磁盘容量10T，如果按照默认Balancer平衡策略（按照使用百分比的策略），比如说最终会趋向于A节点使用70T，空闲30T，B节点使用7T,空闲3T。这种情况下，我们显然不希望把数据放到B节点上了，因为按照绝对值来讲，A节点剩余的30T空间显然会大很多。

默认数据平衡策略的缺陷

- 在讲述本文主题之前，我们首先得理解现有策略的缺陷和不足，然后我们才能知道怎么去改进。如前言中已经提过，现有默认的Balancer策略更适应于完全同质化的节点结构（比如说相同磁盘空间），这样的话，它能够始终保持这些节点都存有差不多数据量的块数据。

- 但是在磁盘容量差距巨大的情况，比如说集群20个节点，10个节点拥有超大磁盘容量（100T）,而另10个节点则是普通的10T，这个时候我们当然更倾向于将更多的数据往大容量节点机器上放。一种情况，我们限制磁盘的使用率在90%，这是小磁盘容量剩余1T，这能够接受，但是大磁盘容量每台机器，就剩了10T，10个节点就是100T，可是很大的空间浪费啊。如果你想利用掉着10T,那么相对应的小磁盘容量机器会受不了，它的剩余容量绝对值已经很少了，到时机器的读写性能也可想而知。

- 所以针对此，笔者想到了基于剩余空间量的数据均衡策略。此策略的最终目的是使各个节点的剩余空间相等，而不是按照使用占比。

- 基于这类场景，我们可能需要一种基于剩余空间的数据均衡策略，使上面的例子最终平衡的效果变为A节点使用97T,空闲3T，B节点使用7T，空闲3T。
- 链接：https://blog.csdn.net/androidlushangderen/article/details/78308893