

Quorum NWR

基本概念

特点 ○ Quorum NWR算法的最大特点就是能够自定义数据的一致性级别，相较于Raft和Paxos，更加的灵活

N表示某一个数据的副本数，又称之为复制因子。数据副本数不一定等于集群节点数，并且当对数据进行数据分区时，每一个分区的副本数视重要程度而有不同的副本数量



W即Write的缩写，表示数据写入时的一致性级别，通常表示完成W个副本数据的写入，Write操作才算成功



R即Read的缩写，表示数据读取时的一致性级别，通常表示读取R个副本的数据，并返回这R个数据中最新的数据

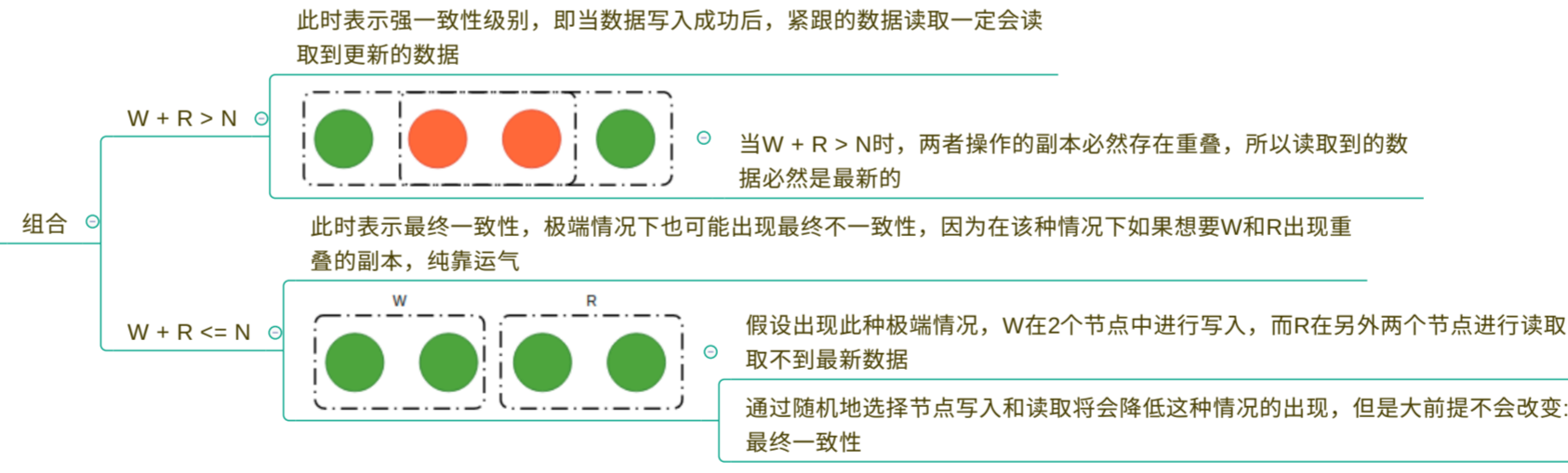


如上图所示，即使在写入时并没有对节点B的DATA-2数据副本进行更新，在Read level为2的情况下，依然能够读取到写入数据的节点。两者数据进行对比，返回最新的数据即可

原理 ○ 正如同该算法的名称(Quorum NWR)一样，对数据进行读取和写入时均访问一定数量的副本数据，即可实现不同的数据一致性级别

一致性级别

因为Quorum NWR算法中的N、W以及R均表示数据副本数，那么不同的组合将会带来不同的一致性级别



实际应用

- InfluxDB ○ 在InfluxDB中，我们可以在数据写入时指定一致性级别
 - any ○ 任何一个节点写入成功后即返回，此时可能仅仅只写入至了节点的缓存汇中
 - one ○ 任何一个节点写入成功后即返回，但是并不包括写入节点缓存中，也就是说，必然存在一个节点成功的对数据进行了持久化
 - quorum ○ 集群中大多数节点写入成功后才会返回
 - all ○ 仅在所有节点均写入成功后才会返回
- MongoDB ○ MongoDB同样的实现了诸如InfluxDB的写一致性级别，只不过一致性级别的名词不同而已

不同场景下的NWR取值

- 当N等于节点数量，即数据的副本数与节点数相同时，此时容灾效果最佳
- 当W = N时，数据读取只需要读取一个节点的数据，那么读性能将达到最佳
- 当R = N时，数据写入只需要写入到一个节点，那么写性能将达到最佳