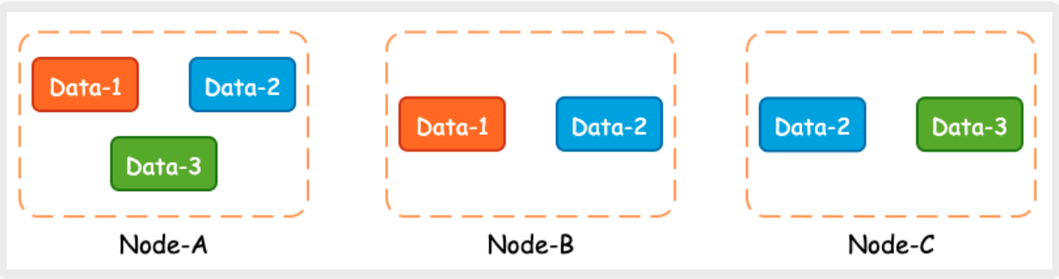
Quorum NWR

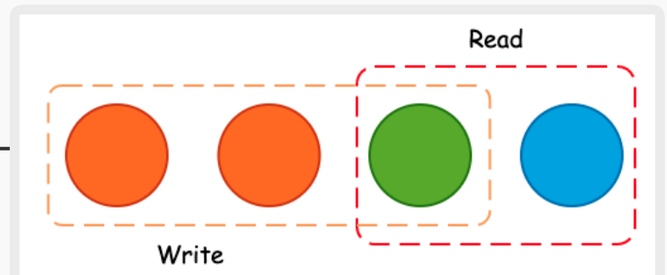
基本概念

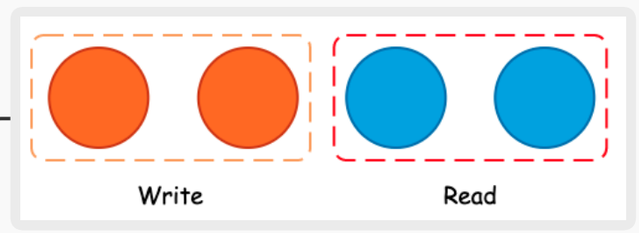
- 特点
  - Quorum NWR 算法的最大特点就是能够自定义数据的一致性级别，相较于 Raft、Paxos 或者是 Gossip，更加的灵活
- 组成
  - ① N
    - N 表示某一个数据的副本数，或者是复制因子。数据副本数不一定等于集群节点数。当对数据进行数据分区时，每一个分区的副本数视重要程度有不同的副本数量，比如 Elasticsearch 对于每一个 Index 都可以有不同的副本数量
    - 

    - 集群中共有 3 个节点，其中 Data-1 和 Data-3 的副本数量为2，Data-2 的副本数量为 3，也就是 Data-2 的重要程度更高
  - ② W
    - 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
    - 假设对 Data-2 的数据写入级别为 2，则只要有 2 个副本成功写入，就算写操作成功。副本位于不同的节点上，所以也可以认为是写入多少个节点才算写入成功
  - ③ R
    - 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，并不是返回 R 个数据
    - 假设对 Data-2 的数据读取级别为 3，那么需要读取 3 个副本的数据，并返回最新的数据
- 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（节点），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

- W + R > N
  - 

  - 当 W + R > N 时，两者操作的副本必然存在重叠，所以读取到的数据必然是最新的。此时表示强一致性级别，即当数据写入成功后，紧跟的数据读取一定会读取到更新的数据
- W + R <= N
  - 

  - 此时表示最终一致性，极端情况下也可能出现最终不一致性，因为在该种情况下 W 和 R 出现重叠的副本是基于概率的

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
  - 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称不同而已
- 不同场景下的 NWR 取值
  - **① 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**
  - **② 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**
  - **③ 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**
- Quorum NWR 并没有像 Paxos、Raft 那样非常复杂的算法流程，而是利用了一个非常简单的数学集合概念。并且，Quorum 不管是在 Paxos 还是 Raft 中都起到了决定性的作用，因为它们有着 "多数节点" 这一至关重要的写入条件
- **Quorum NWR 拥有很强的灵活性，即使是在 Gossip 实现的最终一致性模型中，我们依然可以对其添加读、写时需要操作的副本数，来实现某些数据的强一致性**