Quorum NWR 算法

jask

2024-08-11

Quorum NWR 算法

一致性与强一致性的区别

强一致性能保证写操作完成后，任何后续访问都能读到更新后的值；

最终一致性只能保证如果对某个对象没有新的写操作了，最终所有后续访问都能读到相同的最近更新的值。也就是说，写操作完成后，后续访问可能会读到旧数据。

你需要了解 Quorum NWR 中的三个要素，N、W、R。因为它们是 Quorum NWR 的核心内容，我们就是通过组合这三个要素，实现自定义一致性级别的。

Quorum NWR 三要素

1.N 表示副本数，又叫做复制因子（Replication Factor）。也就是说，N 表示集群中同一份数据有多少个副本，就像下图的样子：

在这个三节点的集群中，DATA-1 有 2 个副本，DATA-2 有 3 个副本，DATA-3 有 1 个副本。也就是说，副本数可以不等于节点数，不同的数据可以有不同的副本数。

在实现 Quorum NWR 的时候，你需要实现自定义副本的功能。

2.W，又称写一致性级别（Write Consistency Level），表示成功完成 W 个副本更新，才完成写操作：

DATA-2 的写副本数为 2，也就说，对 DATA-2 执行写操作时，完成了 2 个副本的更新（比如节点 A、C），才完成写操作。

3.R，又称读一致性级别（Read Consistency Level），表示读取一个数据对象时需要读 R 个副本。你可以这么理解，读取指定数据时，要读 R 副本，然后返回 R 个副本中最新的那份数据：

DATA-2 的读副本数为 2。也就是说，客户端读取 DATA-2 的数据时，需要读取 2 个副本中的数据，然后返回最新的那份数据。

无论客户端如何执行读操作，哪怕它访问的是写操作未强制更新副本数据的节点（比如节点 B），但因为 W(2) + R(2) > N(3)，也就是说，访问节点 B，执行读操作时，因为要读 2 份数据副本，所以除了节点 B 上的 DATA-2，还会读取节点 A 或节点 C 上的 DATA-2，就像上图

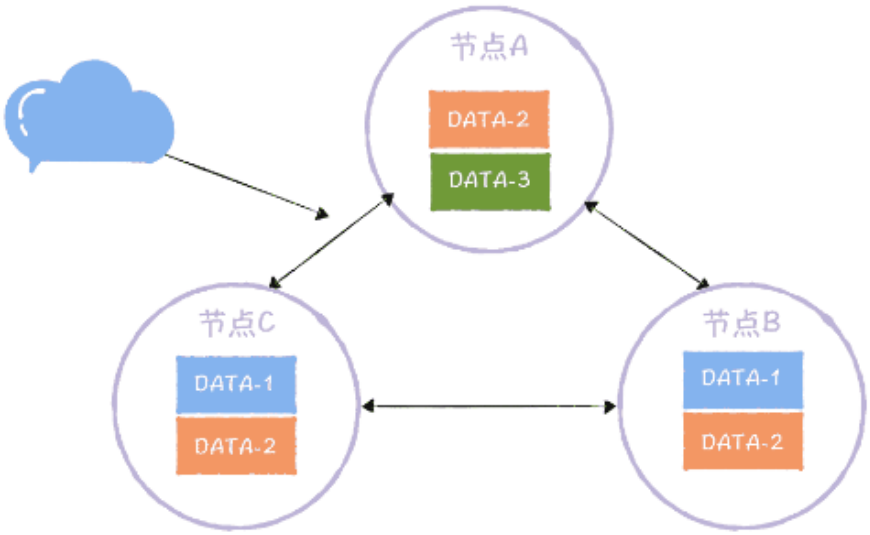

Figure 1: 示意

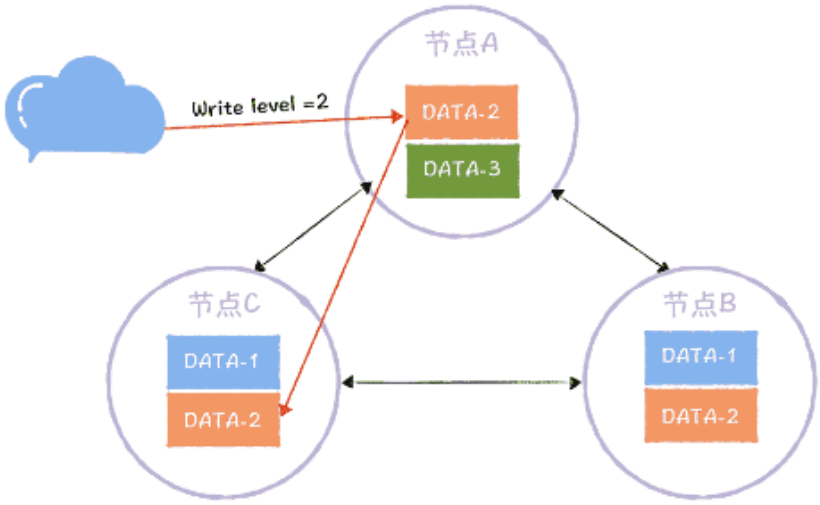

Figure 2: 示意

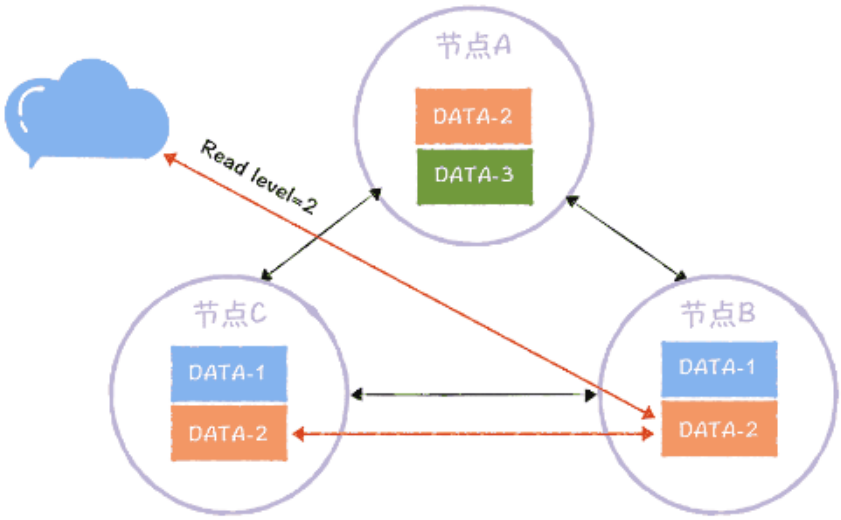

Figure 3: 示意

的样子（比如节点 C 上的 DATA-2），而节点 A 和节点 C 的 DATA-2 数据副本是强制更新成功的。这个时候，返回给客户端肯定是最新的那份数据。

一致性效果

当 W + R > N 的时候，对于客户端来讲，整个系统能保证强一致性，一定能返回更新后的那份数据。

当 W + R < N 的时候，对于客户端来讲，整个系统只能保证最终一致性，可能会返回旧数据。

如何实现 Quorum NWR?

InfluxDB 可以创建保留策略，支持“any、one、quorum、all”4 种写一致性级别，具体的含义是这样的。

any：任何一个节点写入成功后，或者接收节点已将数据写入 Hinted-handoff 缓存（也就是写其他节点失败后，本地节点上缓存写失败数据的队列）后，就会返回成功给客户端。

one：任何一个节点写入成功后，立即返回成功给客户端，不包括成功写入到 Hinted-handoff 缓存。

quorum：当大多数节点写入成功后，就会返回成功给客户端。此选项仅在副本数大于 2 时才有意义，否则等效于 all。

all：仅在所有节点都写入成功后，返回成功。

总结

1. 一般而言，不推荐副本数超过当前的节点数，因为当副本数据超过节点数时，就会出现同一个节点存在多个副本的情况。当这个节点故障时，上面的多个副本就都受到影响了。

2. 当 W + R > N 时，可以实现强一致性。另外，如何设置 N、W、R 值，取决于我们想优化哪方面的性能。比如，N 决定了副本的冗余备份能力；如果设置 W = N，读性能比较好；如果设置 R = N，写性能比较好；如果设置 W = (N + 1) / 2、R = (N + 1) / 2，容错能力比较好，能容忍少数节点（也就是 (N - 1) / 2）的故障。

Quorum NWR 是非常实用的一个算法，能有效弥补 AP 型系统缺乏强一致性的痛点，给业务提供了按需选择一致性级别的灵活度，建议你的开发实现 AP 型系统时，也实现 Quorum NWR。