写入数据时自动注册设计文档（分布式版）

报告人: 康愈圆

写入数据时如果没有事先创建存储或时间序列、创建时间序列时如果没有对应的存储组，系统都应当自动生成对应的存储组或时间序列（这个过程统称为自动注册），并保证数据的正确写入。

目录

1 节点之间的通信

本节内容以 3 节点 2 副本为例，枚举各场景下数据写入触发自动注册的场景，描述节点间的通信过程。各个节点在集群中的角色如图 1 所示。

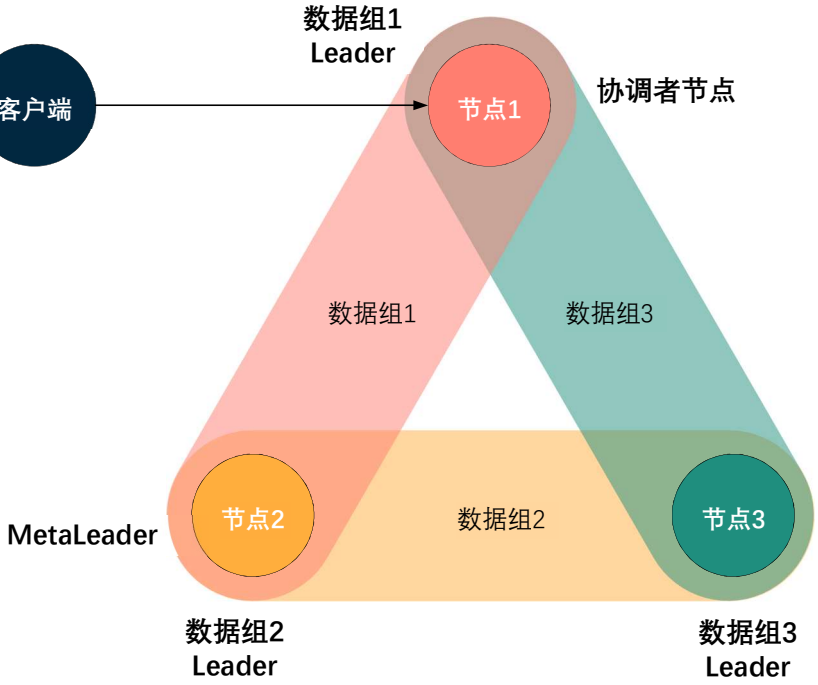

图 **1** 3 节点 2 副本集群中各节点的角色

集群中有 3 个节点，分别为节点 1，节点 2，节点 3。节点 2 为 meta leader。集群中有 3 个数据组，数据组 1 中包含节点 1 和节点 2，其中节点 1 为 data leader；数据组 2 中包含节点 2 和节点 3，其中节点 2 为 data leader；数据组 3 中包含节点 1 和节点 3，其中节点 3 为 data leader。

- 物理计划和 log: leader 节点将物理计划转化为 log，log 中包含物理计划，log 的 index 为 *i*，是一个递增的值。leader 和 follower 节点上还会记录 commit index，记录的是已执行的 log 的 index。log 在节点上是按 index 的大小顺序执行的。

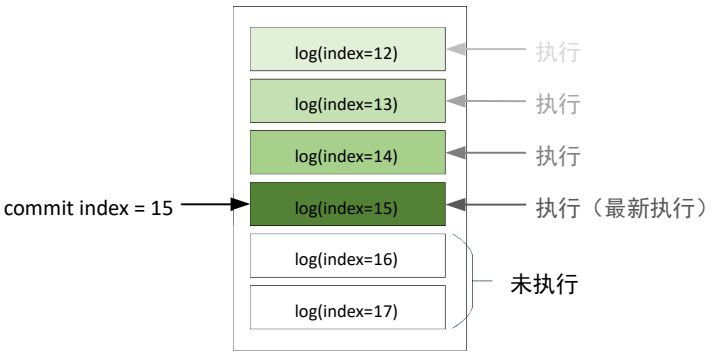

图 2 log index 和 commit index

- 广播和同步物理计划：广播指的是 leader 节点首先向各个 follower 发送含有物理计划的 log，各节点收到 log（未执行）后返回“接收成功”。当 leader 节点收到成功反馈的数量超过$\lceil \frac{N_{\text{follower}}}{2} \rceil$时，执行该 log 中的物理计划，并更新相应的 commit index。节点在同步时，心跳数据里包含 leader 的 commit index。如果 follower 接收到的心跳数据中 leader 的 commit index 比该 follower 本地的 commit index 大，则该 follower 将执行 index 大小在本地 commit index 到 leader commit index 之间的所有 log。本文档涉及的节点间通信主要关注广播过程，不关注同步过程。

1.1 写入数据时不存在存储组

本小节介绍写入 (InsertPlan) 数据时，存储组不存在的情况。

1.1.1 协调者节点为目标数据组的 leader

协调者节点为目标数据组 leader 的情况如图 3 所示。

1. 写入数据：客户端向协调者节点发送写入数据的请求。
2. 创建存储组：协调者节点获取 InsertPlan 中的 device id，在本地查询和 device id 对应的存储组。协调者节点发现不存在满足条件的存储组，向 meta leader 发送创建存储组请求。

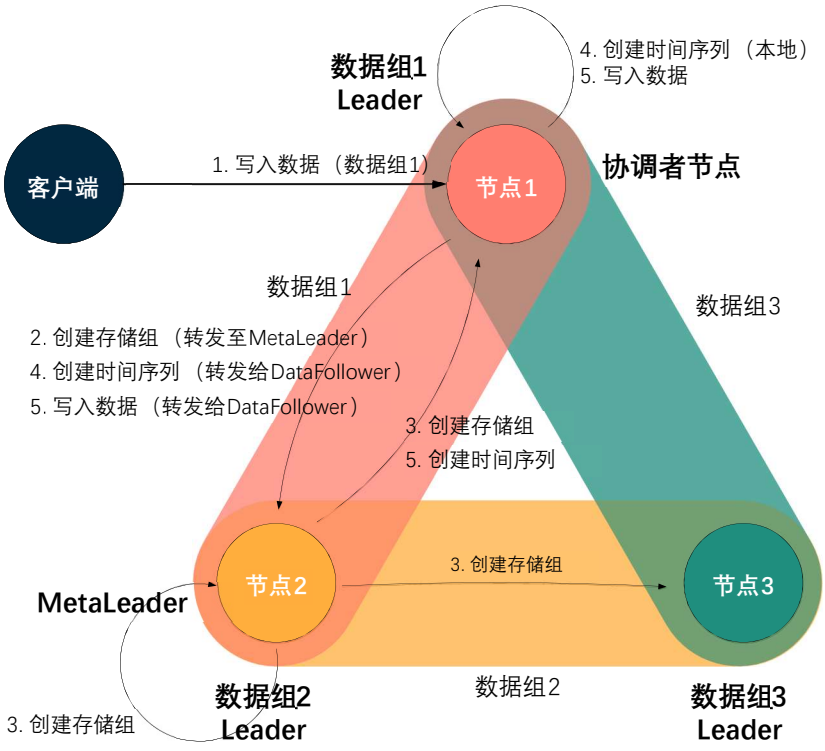

图 3 写入数据时存储组不存在且协调者节点为目标数据组 leader

3. 创建存储组：meta leader 向集群中的所有节点广播创建存储组请求。

4. 创建时间序列：节点 1 通过第 1 步得到的 InsertPlan 自动生成 CreateTimeseriesPlan。此时，对应的存储组已经创建，节点 1 通过哈希得到该存储组对应的数据组是数据组 1，并且节点 1 是该数据组的 data leader，节点 1 在数据组中广播创建时间序列请求。

5. 写入数据：时间序列创建成功后，节点 1 再在数据组 1 中广播写入数据的请求，此时写入数据应当可以正常执行。

1.1.2 协调者节点为目标数据组的 follower

协调者节点为目标数据组 follower 的情况如图 4 所示。

1. 写入数据：客户端向协调者节点发送写入数据的请求。

2. 创建存储组：协调者节点获取 InsertPlan 中的 device id，在本地查询和 device id 对应的存储组。协调者节点发现不存在满足条件的存储组，向 meta leader 发送创建存储组请求。

3. 创建存储组：meta leader 向集群中的所有节点广播创建存储组请求。

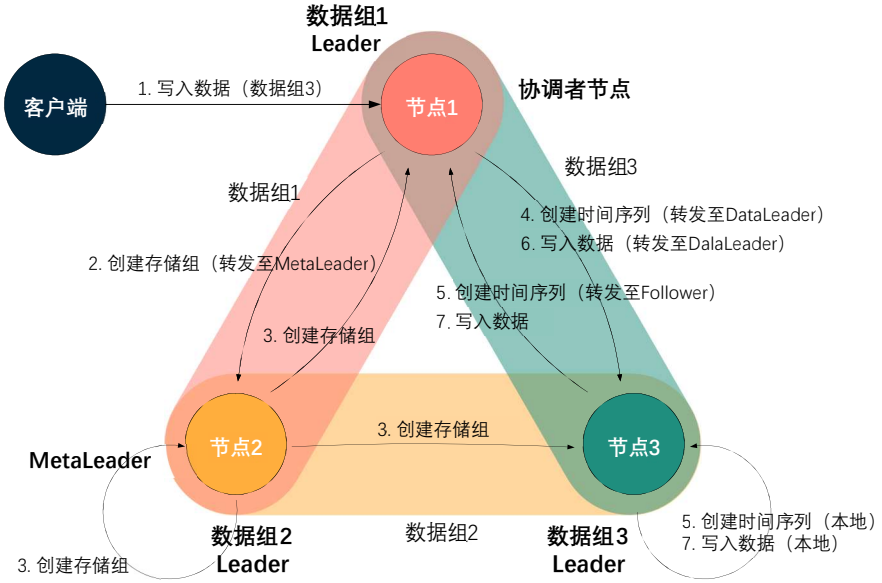

图 **4** 写入数据时存储组不存在且协调者节点为目标数据组 follower

4. 4. 创建时间序列：节点 1 通过第 1 步得到的 InsertPlan 自动生成 CreateTimeseriesPlan。此时，对应的存储组已经创建，节点 1 通过哈希得到该存储组对应的数据组是数据组 3，节点 1 是该数据组的 follower，因此就将该 CreateTimeseriesPlan 转发给节点 3，即该数据组的 data leader。

5. 创建时间序列：节点 3 在数据组 3 中广播创建时间序列请求。

6. 写入数据：节点 1 将从第 1 步得到的 InsertPlan 转发给节点 3（其所在数据组的 data leader）.

7. 写入数据：节点 3 在数据组 3 中广播写入数据的请求。

考虑到另一种策略，从第 4 步开始，节点 1 发送写入请求给节点 3（数据组 3 的 data leader），节点 3 在数据组内广播写入请求，节点 3 物理计划执行失败后，在数据组内广播创建时间序列请求，最后广播写入数据请求。这么做可以降低协调者节点 (节点 1) 的压力，但是存在以下问题：

- 信息丢失：集群系统对时间序列不存在这一感知丢失了。
- 通信次数：假设数据组中的节点个数为 m（副本数），满足 $m \geq 2$，因为副本数为 1 时不存在通信操作。此时，数据组中有 1 个 data leader，有 $(m-1)$ 个 data follower，则会产生 $(3m-2)$ 次通信。如果采用图 4 中的策列，会产生 $2m$ 次通信。
- 通信数据量：假设创建时间序列请求大小为 c_{create}，写入数据请求大小为 c_{insert}。一般地，$c_{\text{create}} \ll c_{\text{insert}}$。那么使用降低协调者节点压力的方法，通信数据量大小约为 $[(2m-1)c_{\text{insert}} + (m-1)c_{\text{create}}]$。使用图 4 中的策列，通信数据量大小约为 $[mc_{\text{insert}} + mc_{\text{create}}]$。

$$\frac{(2m-1)c_{\text{insert}} + (m-1)c_{\text{create}}}{mc_{\text{insert}} + mc_{\text{create}}} = \frac{1}{1+\frac{c_{\text{create}}}{c_{\text{insert}}}} + 1 - \frac{1}{m}$$

$f\left(\frac{c_{\text{create}}}{c_{\text{insert}}}\right) = \frac{1}{1+\frac{c_{\text{create}}}{c_{\text{insert}}}} + 1 - \frac{1}{m}$在其定义域$\frac{c_{\text{create}}}{c_{\text{insert}}} \in (0,1)$上的值域为$(\frac{3}{2} - \frac{1}{m}, 2 - \frac{1}{m})$。

在分布式场景里，当副本数 $m \geq 2$ 时，$f\left(\frac{c_{\text{create}}}{c_{\text{insert}}}\right) > 1$恒成立。

1.1.3 协调者节点不在目标数据组中

协调者节点不在目标数据组中的情况如图 5 所示。

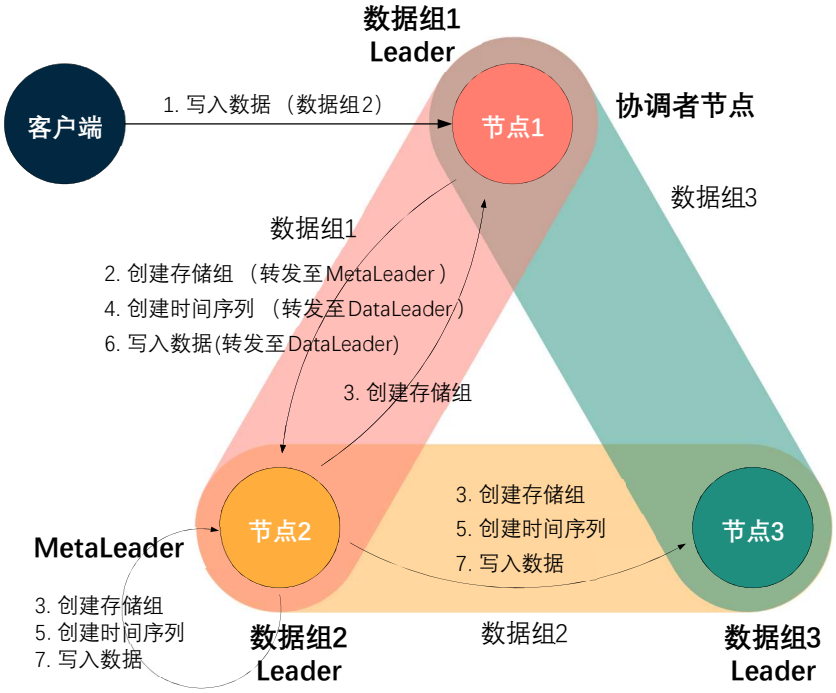

图 **5** 写入数据时存储组不存在且协调者节点不在目标数据组内

该过程与第 1.1.2 小节基本一致。

1.2 写入数据时不存在时间序列

本节内容描述写入数据时，分布式系统中存在对应的存储组但是不存在对应的时间序列的情况。

1.2.1 协调者节点为目标数据组的 leader

写入数据时不存在时间序列且协调者节点为目标数据组 leader 的情况见图 6。

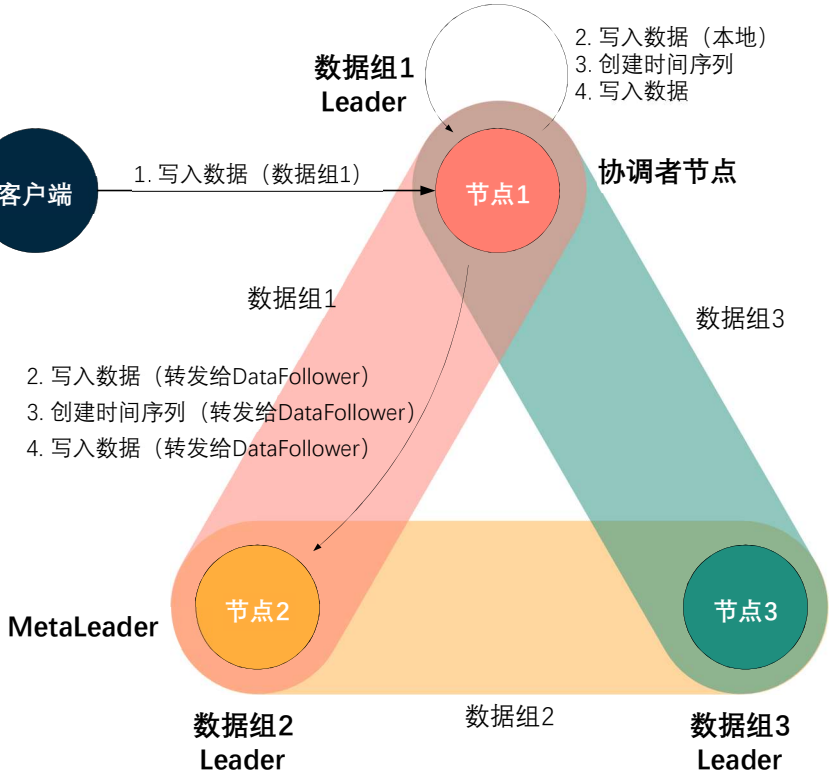

图 **6** 写入数据时时间序列不存在且协调者节点为数据组 leader

1. 写入数据：客户端向协调者节点发送写入数据的请求。
2. 创建存储组：协调者节点获取 InsertPlan 中的 device id，在本地查询和 device id 对应的存储组。协调者节点找到了对应的存储组并且发现对应的数据组是数据组 1，由于节点 1 是数据组 1 的 data leader，它将在数据组中广播写入数据请求。
3. 创建时间序列：节点 1 执行写入数据时，由于没有对应的时间序列，会报错。节点 1 在数据组 1 内广播创建时间序列的请求。
4. 写入数据：节点 1 在数据组 1 内广播写入数据的请求。

以上设计可能有改进空间。按照现在分布式的逻辑，节点 1 在第 1 步之后会将含有写入数据物理计划的 log 发送给数据组 1 中的其他节点。由于 log 是按顺序执行的，这些节点最终总是会执行这一查询计划并报错。

1.2.2 协调者节点为目标数据组的 follower

写入数据时不存在时间序列且协调者节点为目标数据组 follower 的情况见图 7。

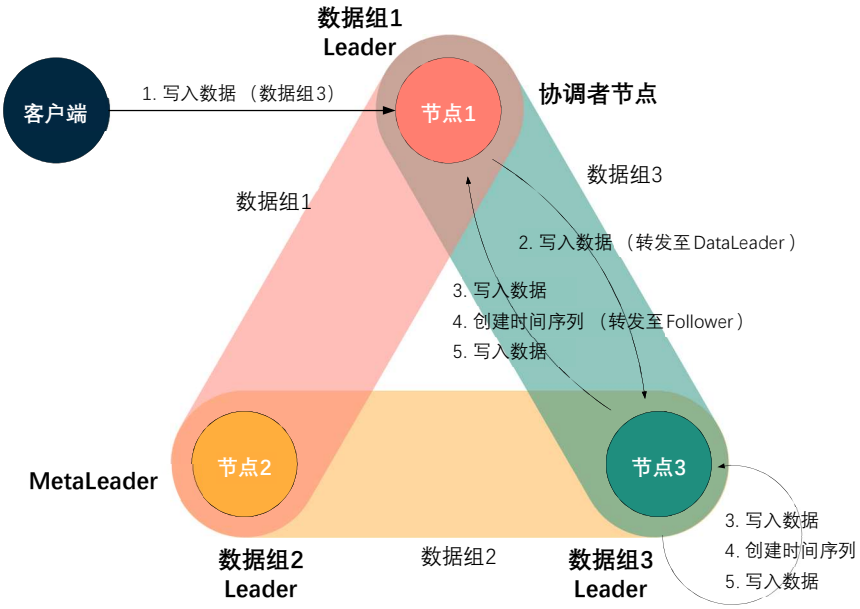

图 7 写入数据时时间序列不存在且协调者节点为数据组 follower

1. 写入数据：客户端向协调者节点发送写入数据的请求。
2. 写入数据：协调者节点获取 InsertPlan 中的 device id，在本地查询和 device id 对应的存储组。协调者节点找到了对应的存储组并且发现对应的数据组是数据组 3，由于节点 1 是数据组 3 的 follower，它将把物理计划转发给节点 3（data leader）。
3. 写入数据：节点 3 在数据组 3 中广播写入数据请求。
4. 创建时间序列：在节点 3 执行写入数据时，由于不存在对应的时间序列，会报错。节点 3 在数据组 3 中广播创建时间序列请求。
5. 写入数据：节点 3 在数据组 3 中广播写入数据请求。

1.2.3 协调者节点不在目标数据组中

写入数据时不存在时间序列且协调者节点不在目标数据组中的情况如图 8 所示。该过程与第 1.2.2 小节基本一致。

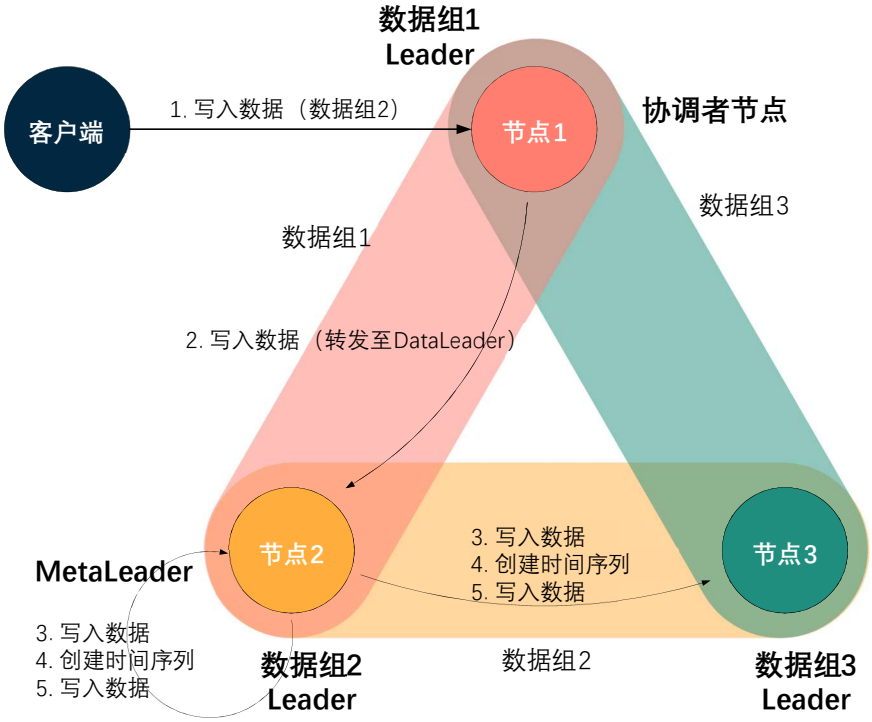

图 **8** 写入数据时时间序列不存在且协调者节点不在目标数据组中

1.3 创建时间序列时不存在存储组

1.3.1 协调者节点是目标数据组的 leader

创建时间序列时不存在存储组，且协调者节点时目标数据组 leader 的情况如图 9 所示。

1. 创建时间序列：客户端向协调者节点发送创建时间序列的请求。
2. 创建存储组：协调者节点获取 CreateTimeseriesPlan 中的 device id，在本地查询和 device id 对应的存储组。协调者节点没有找到对应的存储组，报错。协调者节点发送创建存储组请求至 meta leader。
3. 创建存储组：节点 2（meta leader）在集群中广播创建存储组请求。
4. 创建时间序列：节点 1（数据组 1 的 data leader）在数据组 1 中广播创建时间序列请求。

1.3.2 协调者节点是目标数据组的 follower

创建时间序列时不存在存储组，且协调者节点是目标数据组 follower 的情况如图 10 所示。

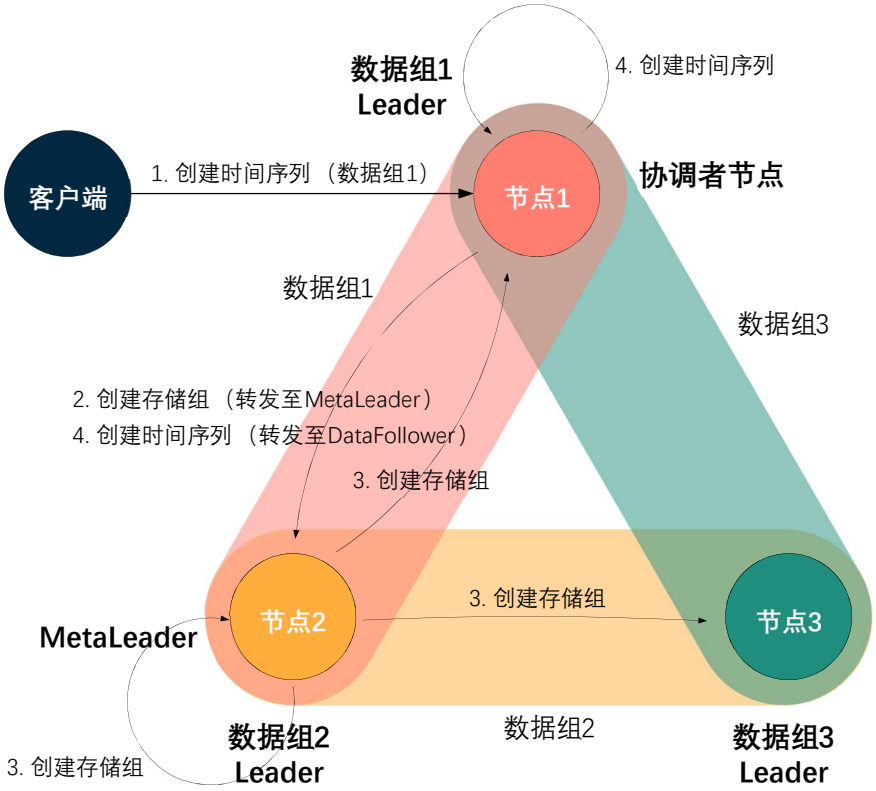

图 **9** 创建时间序列时存储组不存在且协调者节点为数据组 leader

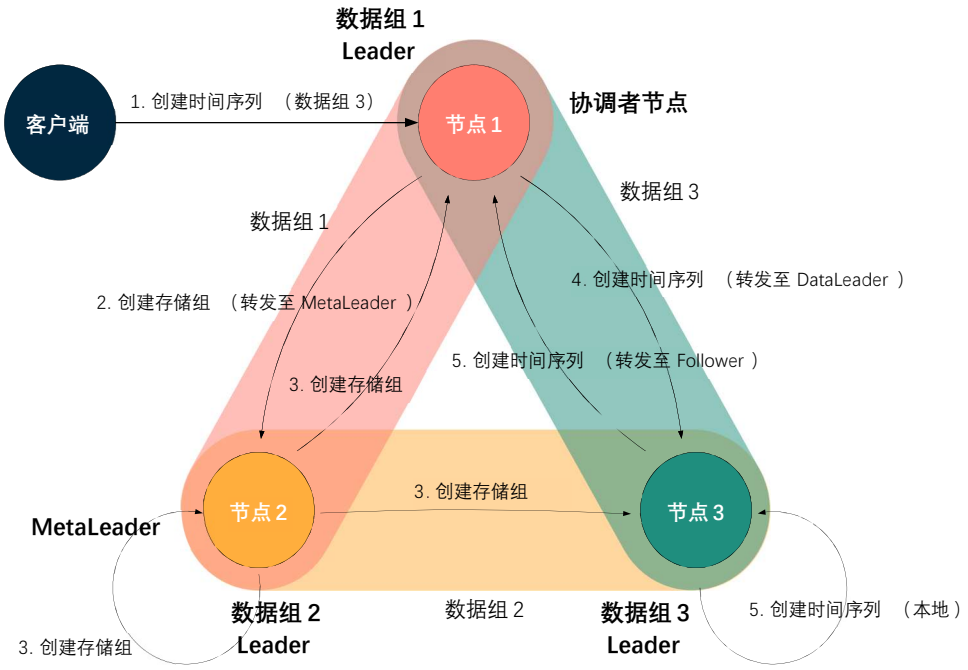

图 **10** 创建时间序列时存储组不存在且协调者节点为数据组 follower

1. 创建时间序列：客户端向协调者节点发送创建时间序列的请求。
2. 创建存储组：协调者节点获取 CreateTimeseriesPlan 中的 device id，在本地查询和 device id 对应的存储组。协调者节点没有找到对应的存储组，报错。协调者节点发送创建存储组请求至 meta leader。
3. 创建存储组：节点 2(meta leader) 在集群中广播创建存储组请求。
4. 创建时间序列：节点 1 转发创建时间序列请求给节点 3（数据组 3 的 data leader）。
5. 创建时间序列：节点 3（数据组 3 的 data leader）在数据组 3 中广播创建时间序列请求。

1.3.3 协调者节点不在目标数据组中

创建时间序列时不存在存储组，且协调者节点不在目标数据组中的情况如图 11 所示。该过程与第 1.3.2 小节基本一致。

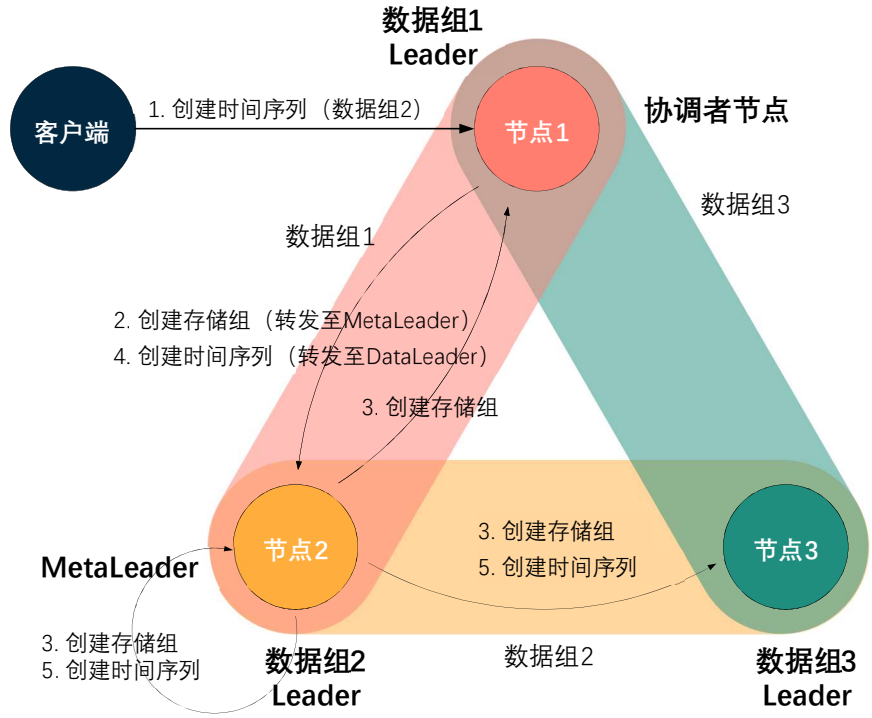

图 **11** 创建时间序列时存储组不存在且协调者节点不在目标数据组中