## **Which Server To Write**

## 1. 一个layout实例

```
"layoutServers": [
    "localhost:9000",
    "localhost:9001",
    "localhost:9002",
    "localhost:9003",
    "localhost:9004",
    "localhost:9005",
    "localhost:9006"
],
"sequencers": [
    "localhost:9000",
    "localhost:9001"
],
"segments": [
    {
        "replicationMode": "CHAIN_REPLICATION",
        "start": 0,
        "end": 20000,
        "stripes": [
            {
                 "logServers": [
                     "localhost:9002",
                     "localhost:9003"
                ]
            },
                 "logServers": [
                     "localhost:9004"
                ]
            }
        1
    },
        "replicationMode": "CHAIN_REPLICATION",
        "start": 20000,
        "end": -1,
        "stripes": [
            {
                "logServers": [
                     "localhost:9005"
                ]
            },
                 "logServers": [
                     "localhost:9006"
                ]
            }
```

```
}
],
"epoch": 775825
}
```

如上所示,假设系统中一共有7个Server (CorfuServer) ,从9000到9006。

首先,每个Server都要必须被配置为layoutServer,但是可同时选择配置为sequencerServer和/或logUnitServer(或者都不)。

每个Server上都存有一份上述形式的layout。

segments项中,每个{}包围的部分是一个segment,其内容表示该segment的REPLICATION\_MODE、地址区间(end为-1表示正无穷)以及各stripe对应的logUnitServer的地址。同一个stripes中的同一个logServers中的不同地址表示同一日志的多份拷贝。

## 2. 写过程

- client构造一个CorfuRuntime实例,连接以上任意一个Server,获得最新layout。
  - o 连接:以server的"ip:port"为目的地址构造一个channel (默认类型为 NioSocketChannel) ,并通过该channel发送一个 HANDSHAKE\_INITIATE 消息,server回 应一个 HANDSHAKE\_RESPONSE 消息,连接建立,这是getRouter函数的主要作用。
  - o 获得layout: client发送一个 LAYOUT\_REQUEST 消息,server返回一个 LAYOUT\_RESPONSE 消息,内含最新layout。
- client从layout中得知所有sequencerServer,从中选择一个server,连接该server,向其发送TOKEN\_REQ 消息,server回复一个 TOKEN\_RES 消息,内含(读/写)所需的地址。
  - o 当前实现是,对于所有该种请求,总是选择上述sequencers列表中第一个sequencer来请求 token,这可能会成为瓶颈。
- 对于写操作,client会利用得到的地址,查询layout得到该地址所在的segment和stripe,然后连接该stripe中的server,向其发起写请求(WRITE 消息)。server会将其写在其本地磁盘上,当前实现中磁盘上每个日志文件存储1w条记录(StreamLogFiles 类)。写成功后,server会返回一个WRITE\_OK 消息。