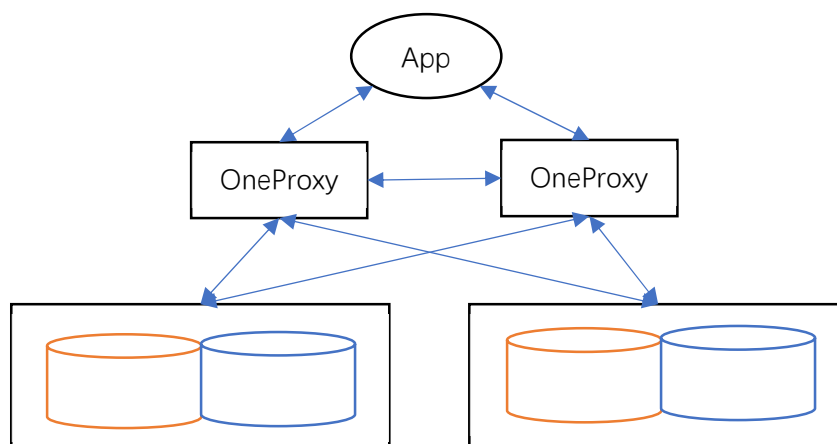


OneProxy 是一个高效稳定的 MySQL 协议层代理软件，可以透明地支持 MySQL 架构的横向扩展。它具备以下令人心动的功能特点：

- 超 10 个用户已经稳定使用长达 5 年之久，商业软件品质保障。
- 低时延，每个查询增加不到 100us 的时延，远超其他同类 Proxy 软件。
- 稳定性，软件采用 C&C++ 语言编写，无内存 GC 问题，多个用户场景验证超过一年不用重起。
- 高性能，代理需要解释 SQL 语句，是 CPU 消耗型的操作，每个核每秒能处理超 25000 的查询转发，良好的单机扩展能力，也可构建 Proxy 集群做横向扩展。
- 后端连接池功能，可以有效控制到 MySQL 节点上的总连接数。
- 支持 MySQL Group Replication 和 Percona Xtradb Cluster 集群架构，自动快速识别后端节点切换识别主节点。
- 支持读写分离等多种流量转发策略，无缝实现读扩展。
- 支持分库分表，并有跨节点的结果集聚合操作支持能力，对应用的透明度较高。
 - 支持 Int、Big Int、Char、Date、Timestamp 类型的单列分区。
 - 支持 Range、Hash、List 分区方式，支持二级子分区。
 - 支持分区信息冗余到不同的字段，以更好地进行分区过滤。
 - 支持跨节点的结果集合并及排序（order by）。
 - 支持跨节点的结果集汇总操作（count/sum/max/min）。
 - 支持跨节点的结果集的分组汇总（group by）。
 - 支持跨节点的结果集分页操作（limit/offset）。
 - 支持基于分片的并行查询操作（parallel query）。
- 支持结果集缓存，透明提升查询操作性能。
- 支持前后端密码分离，确保数据库密码不外泄。
- 支持 IP 白名单，支持表级安全设置，支持 SQL 防火墙，阻止 SQL 注入式攻击。
- 内置 HA 机制，无须安装配置第三方软件，实现代理节点的快速故障转移。
- 具备丰富的 SQL 性能统计信息，内置 Http 服务，可以轻松查看性能数据。
- 内置序号生成器，可以高效生成单向增长的序列值。
- 支持 MySQL 8 协议，支持最新版本的 MySQL JDBC 驱动程序。

今天大量的企业已在使用 MySQL，也熟悉了分库分表操作，一个好的数据访问层（DAL）可以起到事半功倍的作用，OneProxy 就是一个经得起考验的优秀数据访问层软件。

要理解 OneProxy 的关键配置项，就需要了解它的基本布署架构，如下图所示：



由于要支持分库分表，所以需要支持挂载多个 MySQL 主备或 MGR 集群，每个集群需要有唯一的名字，可以称之为组名（Group Name）。每个集群可以是一主多从多个结点，也可以是 MGR 或 PXC 集群。怎么样来添加所有的节点？只要提供三个关键的信息：

- MySQL 节点 IP 地址
- MySQL 节点端口号
- 节点所在集群的名字（Group Name）

配置节点可需要到以下两个选项：

- proxy-master-addresses.[1-256] = 节点 IP:端口@集群的名字
- proxy-slave-addresses.[1-256] = 节点 IP:端口@集群的名字

需要注意的是，这里并不需要提供 MySQL 数据库的名字。MySQL 数据库名字在登录信息中提供，需要包含以下 4 个信息：

- 集群的名字（Group Name）
- 登录用户名
- 登录密码（用自带 mysqlpwd 加密，只支持 mysql_native_password 认证）
- 登录数据库名字

配置登录信息需要用到以下两个选项：

- proxy-user-list.[1-256] = 用户名/加密后口令@数据库名字
- proxy-user-group.[1-256] = 集群的名字:用户名/加密后口令@数据库名字

前者指定了应用连接 OneProxy 的登录信息，后者指定了某个集群的登录信息。如果对同一个用户名，在两个地方都指定了信息，则起到前后端密码分离的作用。

许多 HA 软件(包括 MGR)都会通过设置“read_only”变量来标识节点是读写节点(Primary)还是只读节点(Slave)，可以在 OneProxy 中设置以下变量值为 1 来自动识别读写状态，以设置准确的节点类型：

- proxy-auto-readonly = {0|1}

OneProxy 会以 50ms 的频率去检查后端节点状态，基本上是应用无感的秒切。只要 HA 或 MGR 集群能保证同一时间点只有一个节点会关闭“read_only”变量，就不会存在双写情况。这样可以省去通知前端应用写库切换的工作，在大规模应用布署下，这个通知机制还是比较复杂的，使用 OneProxy 则简洁多了。

只要再提供其他几个关键的选项，就可以启动 OneProxy 来进行测试了。如下所示：

- mysql-version = 虚拟版本号（应当是所有节点中最小的 MySQL 版本号）
- event-threads = 允许的 CPU 核数（一般不超过 16，平均每核 2.5 万 QPS 转发）
- proxy-address = 监听地址（默认是“:3307”，等同于 MySQL 的 Listener 地址）
- proxy-group-policy = 集群的名字:流量策略（master-only 或 read-balance）

接下来就可以来准备一个完整的配置文件（conf/proxy.conf）了，如下所示：

```
[oneproxy]
#proxy-address           = :3307
mysql-version            = 8.0.27
event-threads            = 2

proxy-auto-readonly      = 1
proxy-group-policy       = default:read_balance
proxy-master-addresses.1 = 127.0.0.1:3306@default
proxy-user-list.1        = test/password@test
```

OneProxy 会自动启用一个守护进程，以保证单实例的高可用。接下来就可以开始压测了，如查你用 sysbench 工具，则需要加上“--db-ps-mode=disable”选项（因为在分库分表的情况下，PrepareStatement 的管理过于复杂）。

Good Luck!