

基于 LVS 和 Keepalived 的车联网网关集群系统构建研究

刘晨琦

(上海势航网络科技有限公司, 上海 201702)

摘要 在车联网系统中, 为确保通信的实时性并减少网络流量, 车载终端与车联网平台之间建立 TCP 进行通信数据传输。平台网关模块负责建立 TCP 通信连接并进行连接池管理, 对终端上报的协议报文进行接收和解析。然而, 当平台接入终端数超过网关最大负荷时, 网关需要维持大量的连接和连接池管理, 并在内存中进行复杂的数据处理操作, 这种单节点网关架构难以支持数万辆车载终端的数据通信和传输, 无法保证系统的稳定性。因此, 本文提出采用 LVS+Keepalived 负载均衡技术来构建车联网网关集群, 并实现负载均衡, 确保网关在大量终端接入的情况下和网关发生故障情况进行故障转移来保持高可用性。通过网关集群代理解决了多网关集群模式下终端上下线状态维护以及指令下发的相关问题。

关键词 车联网; 网关; 指令; 负载均衡; 集群代理

中图分类号: U12; TP3

文献标志码: A

文章编号: 2097-3365(2024)07-0013-03

我国大力发展信息化产业, 物联网技术得到了大力的发展, 车联网平台技术和产业也获得了成长。在经济发展繁荣的前提下, 物流运输车辆也在不断地增长, 按交通运输部和各省市的安全监管要求及工业和信息化部对于电动车的相关要求, 规定内的车辆必须接入车联网联控系统, 同时企业自身运营监管, 也需要车联网平台, 随着平台车辆的增加, 单点网关再难以支撑数以万计的车载终端的数据上报和指令下发需求, 必须转为集群网关来应对当前的情况。在有大量车载终端与平台通信时, 需要启动多个网关, 每个网关对不同的 IP 或端口进行监听。并且这些网关之间缺乏协同工作能力, 无法实现动态负载均衡或高可用性。当某个网关发生故障时, 连接到该网关的终端不具备自动切换网关的能力, TCP 连接断开终端掉线。此外, 恢复车辆在线状态也会给车联网平台的数据库和缓存数据库造成额外压力, 尤其是在短时间内多次发生网关故障时。随着接入终端数量的增加, 网关需要重新部署节点并接入新终端, 系统无法做到动态部署对网关服务进行扩容, 导致终端管理的复杂度提高。

1 系统设计与实现

基于 LVS+Keepalived 在车联网平台^[1-2]中的应用, 系统包含负载均衡服务、网关服务、网关集群代理服务和车联网服务端四个服务, 数据由车载终端发起, 到负载均衡服务器, 再转发到网关, 进行相应的数据处理。

LVS^[3]负责将多台服务器组成一个虚拟服务器, 在网络 OSI 第四层作为分发之用, 对内存和 CPU 消耗较低, 工作模式分为 NAT 模式、TUN 模式, 以及 DR 模式三种工作模式, 支持 10 种调度算法, 本文将采用 NAT 工作模式^[4], 使用轮询调度和加权轮询调度算法。Keepalived 是一种基于虚拟路由冗余协议 (VRRP) 的软件, 用于实现高可用性和故障转移, 使用 VRRP 协议实现的多台服务器之间的状态同步, 定期对各节点进行健康检查用以确保可用性, 在发生故障时, 虚拟 IP 会分配到灾备节点来提供服务, 确保整个集群高可用性。使用 Redis 作为分布式缓存, Redis 是一个开源的使用 ANSI C 语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value 数据库, 当有多个集群代理时, 某一个集群代理宕机重启时, 不需要再将数据初始化到本服务内存中, 减少重启时间和内存占用量大的问题。使用 Kafka 作为分布式消息集群进行数据通信, 通过 Kafka 提供的消息发布订阅解决方案来解决车载终端数据传输的分布式高吞吐量需求。系统架构如图 1 所示。

1.1 服务器搭建及配置

1. 在 Linux 服务器 RS1 上, IP 地址为 RIP1, 在该服务器上添加一块回环网卡, IP 地址记为 VIP, 对内核参数进行修改以实现 ARP 抑制。启动 RS1 上的网关服务, 监听 TCP 通信端口 Port, 网关编号为 G1。参考配置如下:

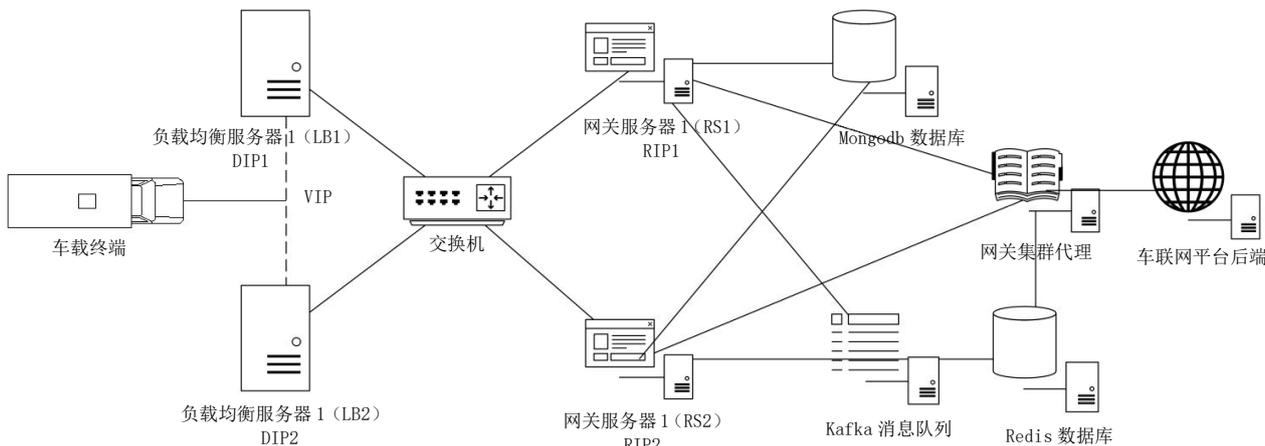


图1 系统架构图

(1) 使用 `ifconfig` 命令创建一个名为 `lo:0` 的虚拟网络接口，并指定虚拟的 IP 地址为 VIP 分配给该接口，子网掩码配置为 `255.255.255.255`，确保只有一个 IP 有效，设置广播地址为 VIP，这样做是为了将 VIP 绑定到本地环回接口。

(2) 使用 `route` 命令将 VIP 添加到本地环回接口的虚拟子接口 `lo:0` 上。

(3) 设置系统参数 `/proc/sys/net/ipv4/conf/lo/arp_ignore` 为 1，表示忽略来自 `lo` 接口的 ARP 请求。

(4) 设置系统参数 `/proc/sys/net/ipv4/conf/lo/arp_announce` 为 2，表示在 ARP 请求中，设置系统在回复时使用绑定到 VIP 的 MAC 地址。

(5) 设置系统参数 `/proc/sys/net/ipv4/conf/all/arp_ignore` 为 1，表示忽略所有接口的 ARP 请求。

(6) 设置系统参数 `/proc/sys/net/ipv4/conf/all/arp_announce` 为 2，表示在 ARP 请求中，使用所有接口的请求 IP 的 MAC 地址。

(7) 使用 `sysctl` 命令重新加载配置，并输出服务启动成功。

2. 在 Linux 服务器 RS2 上添加一块 IP 为 VIP 的回环网卡，在同一物理网络内 IP 地址为 RIP2，对内核参数进行修改以实现 ARP 抑制。启动 RS2 上的网关服务，对 TCP 端口 Port 进行监听，网关编号为 G2。参考配置同上一步骤。

3. 在内核版本为 2.4.23 以上的 Linux 服务器 LB1 上安装 Keepalived，在同一物理网络中其 IP 地址为 DIP1。使用 Keepalived 来对 LVS 进行管理，虚拟的地址为 VIP，在网络接口 `ens192` 上配置了一个虚拟 IP 地址 `172.1.1.82`，子网掩码为 24（即 `255.255.255.0`）。此配置将虚拟 IP 地址绑定到了接口 `ens192` 上，

为了标识该虚拟 IP 地址，使用了 `label` 参数，设置其标签为 `ens192:0`。

4. 在 LB1 的 Keepalived 设置中，选择了 DR 模式（直接路由）作为 `lvs_method` 参数的配置，并且采用了 `wrr` 算法（加权轮循调度）作为 `lvs_sched` 参数的配置。

5. 在 LB1 的 Keepalived 配置中，将 RIP1 网关服务器和 RIP2 网关服务器的 `real_server` 进行配置，并对 `TCP_CHECK` 进行配置来检查健康状态，检查网关监听端口是否正常，`weight` 为权重配置，可以根据服务器性能和可连接数，进行区分配置不同权重。

6. 在同一物理网络的 linux 服务器 LB2 上安装相同的版本的 Keppalived，其 IP 地址为 DIP2。修改 `state` 为 Backup 保存配置文件，标记为灾备服务器，在服务故障时进行切换，其他配置参数与上三个步骤保持相同，具体配置参考 LB1 的配置。

1.2 网关运行方式

1. 在同一物理网络的 linux 服务器上部署网关集群代理，代理主要作用记录各网关的连接信息，并将信息进行缓存，如：终端 ID、网关编号、连接时间等。集群代理和网关通过 Kafka 的发布订阅模式进行数据互通。

2. 车载终端通过 VIP: Port 的地址建立 TCP 连接。

3. LB1 在收到 TCP 的数据后，根据配置的调度算法从 RS1 和 RS2 中分配一台服务器作为目标服务器 RSn，目标网关服务器的编号为 Gn。然后，LB1 将使用 RSn 的 MAC 地址作为 TCP 数据包的目标 MAC 地址，并将修改后的数据包进行分发。

4. 目标网关服务器 RSn 收到数据包后，按照相关的车联网协议进行数据解析，网关将解析后的数据写入数据库中持久化存储，再将终端编号 TID、网关编号

Gn、上线状态、上线时间写入 kafka 消息队列。

5. 集群代理在消费到 kafka 的数据时,将对终端编号 TID、网关编号 Gn、上线状态、上线时间写入 redis 缓存数据库中进行缓存。

6. 当终端下线时, TCP 连接会随之断开, 终端所在的网关 Gn 调用集群代理, 从 Redis 缓存中查询该终端是否在线以及 TCP 连接断开时连接的网关编号 Gm。

7. Gn 和 Gm 编号一致, 则设备为离线状态, 将终端编号 TID、网关编号 Gn、下线状态、下线时间组成的订阅消息发布到 Kafka 消息队列, 集群代理将消费到数据缓存到 Redis 中。Gn 和 Gm 编号不一致, 则认为设备连接其他网关服务中, 不需要写入下线状态。

8. 车联网平台下发指令时, 平台调用网关集群代理 API。将终端编号 TID 和其他指令参数通过 API 接口传输给网关集群代理。

9. 网关集群代理在 Redis 缓存数据库, 查询出该终端所在网关编号 Gn 及其在线状态。

10. 如果终端的上线状态为在线, 根据查询到的网关编号 Gn, 集群代理服务调用 Gn 网关的指令下发 API 接口, 对设备发送指令。

11. 网关 Gn 对下发的指令请求进行处理, 在连接池选取该设备连接, 将指令下发到终端。

12. 如果终端处于下线状态, 网关集群代理对该指令进行缓存, 等网关下次上线时, 集群代理通过终端编号 TID 查询是否有待发指令, 再进行指令下发操作。

2 系统测试与验证

系统环境如表 1 所示。

表 1 系统环境配置

网关	系统	CPU	内存
G1	Centos7	16 核	16G
G2	Centos7	12 核	16G
G3	Centos7	8 核	8G
G4	Centos7	8 核	8G

根据服务器配置, 我们采用了加权轮询调度配置, 配置 lvs_sched 为 wrr 模式, 按照服务器的性能, G1>G2>G3 的权重进行配置, 为了进一步测试调度, G4 配置为 0, 不参与调度, 在内部监控系统中, 可以看到 G1 在线数为 29594, G2 在线数为 20848, G3 在线数为 11425, G4 在线数为 0, 数据并没有分发到 G4, 达到了权重配置后的目标结果, 测试结果达成。

再使用测试轮询调度模式, 配置 lvs_sched 为 rr

模式, 该测试过程, 采用两台同样配置的服务器, 并且接入更多的车辆, 在线数分布较接近, 在线数分别为 65232、65071, 测试结果达成。

在完成网关集群的负载测试后, 进行平台指令下发测试, 检查平台指令下发是否能正常到达设备端, 利用内部售后管理工具, 对设备进行心跳设置, 心跳设置值为 30 s。通过车联网平台, 进行终端参数查询, 在下发查询终端参数指令后, 在消息参数界面, 终端心跳发送间隔查询结果为 30 s, 测试结果达到测试目的。

本方案已经在生产环境中使用, 网关已经实现了 Docker 容器化^[5]部署, 在系统发生故障停止运行时, 可以自动恢复运行, 目前已经部署两套网关集群, 均采用 LVS+Keepalived 的方式进行搭建, 一套网关集群用于 39 万台车的车联网监控平台, 一套网关集群用于 65 万套终端的售后服务平台, 均运行良好, 解决了单个网关集群的负载压力, 同时也避免了在网关代理中开发负载均衡和解决性能问题带来的成本上升问题。

3 结束语

本文的解决方案解决了车联网在大量终端接入时单个网关负载不足的问题和上下行数据交换互通的问题, 在该集群模式下, 网关可以横向扩展来增加整个系统的车辆接入能力, 只需要增加网关服务器, 再进行配置, 可以方便地进行扩容, 同时不再需要从集群代理服务器上解决网关负载均衡, 从而使运维工作简单化并且节省维护成本。从系统复杂性考虑, 还有后续优化的方向, 可以使用 Kubernetes (k8s) 来进行容器的管理, 在车辆上线量增加, 超过目前负荷的情况下, 由 k8s 来进行自动伸缩 / 扩展操作, 进一步提供更稳定的车联网网关集群, 保障系统的高可用性和稳定性。

参考文献:

- [1] 张海, 董云泰, 牛晓, 等. 基于 LVS+rsync+MySQL 的高可用系统构建研究 [J]. 电子质量, 2023(03):16-19.
- [2] 陈建国. 面向车联网的智能网联云控平台研究 [J]. 软件工程, 2024, 27(01):10-13, 18.
- [3] 陈艳, 李文娟. 出租车车联网系统设计与实现 [J]. 信息技术与信息化, 2023(05):64-67.
- [4] 李丽芬, 陈炜, 葛燕平, 等. NAT 模式下 Linux 虚拟机联网及 FTP 服务器的实现 [J]. 电子质量, 2023(07):7-11.
- [5] 陈瑜, 张雨萌. 前后端分离应用容器化集成部署方案的设计与实现 [J]. 软件, 2023, 44(01):77-80.