

高可用SDN控制器主从分布式系统设计

高荣亮, 郑 乐

(中国电子科技集团公司 第三十四研究所, 广西 桂林 541004)

摘要:软件定义网络(SDN)控制器系统是整个网络的大脑,为了降低控制器单点故障造成的影响,设计了一种简单可靠且可用性更强的主从分布式系统和基于优先级及非抢占的主从选举算法,选举过程不再受限于健康节点数量;简化了Raft算法日志复制系统,设计了一套简单高效的主从增量/全量数据复制流程;结合数据持久化存储及恢复功能,实现系统重启后数据不丢失。试验结果表明:该系统实现了目前主流的主从分布式系统的主要功能,性能更优,能适应分布式SDN控制器系统各种复杂应用场景。

关键词:软件定义网络控制器;分布式系统;可用性;主从;Raft 算法

中图分类号:TN929.11 文献标志码:A 文章编号:1002-5561(2021)07-0059-04

DOI:10.13921/j.cnki.issn1002-5561.2021.07.014

开放科学(资源服务)标识码(OSID):



Design of master-slave distributed system for high availability SDN controller

GAO Rongliang, ZHENG Le

(The 34th Research Institute of CETC, Guilin Guangxi 541004, China)

Abstract: Software defined network (SDN) controller system is the brain of the whole network. In order to reduce the influence of single point fault of controller, this paper designs a simple, reliable and more available master-slave distributed system and master-slave election algorithm based on priority and non preemption. The election process is no longer limited by the number of healthy nodes. The log replication system of Raft algorithm is simplified, and a simple and efficient master-slave incremental/full data replication process is designed. Combined with the data persistence storage and recovery function, the data can not be lost after the system restart. The experiment results show that the system has realized the main functions of the main and subordinate distributed system, and the performance is better, and it can adapt to various complex application scenarios of the distributed SDN controller system.

Key words: software defined network controller; distributed system; availability; master-slave; Raft algorithm

0 引言

软件定义网络(SDN)通常采用控制器集中管理整个网络,管理内容包括节点资源、链路资源、带宽预留和路由规划等。为了降低控制器单点故障造成的影响,需要采用分布式多点部署控制器系统,通过具有选举和复制功能的分布式协议实现多点之间数据共享和同步。Paxos 算法^[1]和 Raft 算法^[2]是分布式系统中

收稿日期:2021-03-15。

作者简介:高荣亮(1987—),男,汉族,安徽马鞍山人,硕士,工程师,现就职于中国电子科技集团公司第三十四研究所,从事数通产品以太网协议软件及应用软件开发的开发工作,主要研究方向为IP路由交换、光网络、软件定义网络和分布式数据库等技术。



使用率非常高的同一类共识算法。以 Raft 算法为例,一次数据复制操作需要多数节点提交成功后才算复制成功。因此,算法提交和复制效率低^[3-4],当网络出现 1/2 数量节点故障或者网络分裂成多个节点数量均小于或等于 1/2 的分区后,系统暂时无法选举和复制,直到故障节点恢复或者网络合并后系统才能恢复正常;另外, Raft 算法的选举结果具有一定的随机性,不适用于选举结果可控的场景。针对上述问题,本文设计一种高可用的 SDN 控制器主从分布式系统和主从选举算法。

1 SDN 控制器主从分布式系统设计

SDN 控制器主从系统架构如图 1 所示,该系统的设计主要包括主从选举、主从复制、数据持久化和分

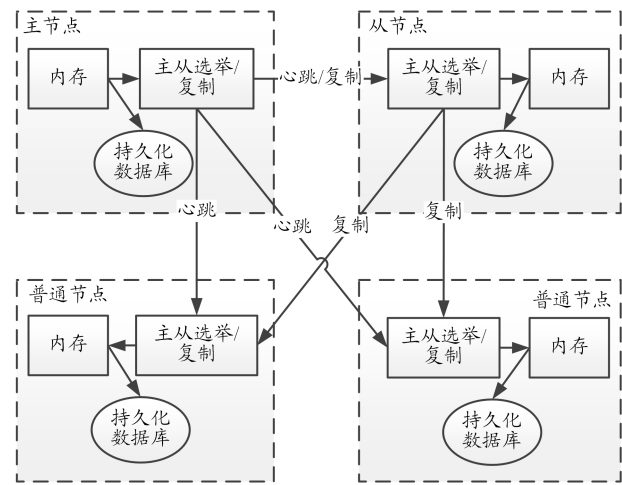


图 1 SDN 控制器主从分布式系统架构

表 1 SDN 控制器主从分布式系统和 Raft 算法分布式系统区别

功能	系统相似点	系统不同点
选举	只有一个主节点或领导者	从分布式系统多一个从节点,主从选举不受节点数量限制,而 Raft 算法选举则受节点数量限制
复制	主节点复制给其它节点	主从复制的主节点只向从节点复制,从节点再复制给其它节点,而 Raft 算法复制领导者复制给所有跟随者;主从复制直接复制数据,Raft 算法复制通过复制日志间接复制数据

区容错性处理等 4 个方面。主从节点选举完成之后,主节点将内存中待同步的数据通过网络复制给从节点,从节点再向普通节点复制;分区合并后,新选举出的主节点完成分区容错处理。

本文设计的 SDN 控制器主从分布式系统和 Raft 算法分布式系统的区别如表 1 所示。

1.1 主从选举

SDN 控制器主从分布式系统由若干个控制器节点组成,每个节点拥有一种角色属性,角色的划分如下:

- ①主节点。负责接收读写请求、指派从节点和向从节点复制数据。
 - ②从节点。负责接收读写请求,当主节点失效后,接替主节点,向普通节点复制数据。
 - ③普通节点。负责接收读写请求,当主从节点失效后,发起选举。
- SDN 控制器采用优先级和非抢占方式选举确定节点角色,其中主节点和从节点分别只能有一个,普通节点数量不受限制。给每个节点预设不同的优先级,所有

节点同时启动后,优先级最高的节点成为主节点,优先级次高的节点成为从节点,其它节点成为普通节点。主节点失效后,从节点直接升级为新的主节点,与 Raft 算法重新选举和可能选举多次相比,主从选举效率更高。

为避免不必要的选举和切换对稳定后的系统造成影响,完成选举之后,更高优先级的节点新加入系统时,不允许抢占主节点和从节点,即非抢占选举。非抢占选举需要每个主节点记录自己的当选时间,当同时出现 2 个主节点时,通过当选时间确定新的主节点。选举过程描述如下:

- ①每个节点启动后,根据配置静态发现系统内所有节点及优先级,初始化为普通节点,启动选举超时定时器。
- ②节点的选举超时定时器到期后,发起选举,向所有静态发现的节点发起投票,推荐本节点为主节点。
- ③收到投票的节点根据静态发现的节点优先级及当选时间确定是否赞成发起投票的节点为主节点。当选时间长的节点获胜;当选时间相同,优先级高的节点获胜;都未当选主节点,优先级高的节点或者已经投票给更高优先级的节点获胜。
- ④发起投票的节点在预选举定时器时间内未发现更高优先级的节点,认为赢得本次选举,正式成为主节点,关闭选举超时定时器,启动心跳超时定时器。
- ⑤节点输掉选举后,停止投票,跟随新当选的主节点,收到主节点的心跳报文后,将选举超时定时器重置。
- ⑥从节点选举超时定时器到期前,若未收到主节点的一个心跳报文,则直接当选新的主节点。
- ⑦普通节点在选举超时定时器到期前,若未收到主节点的任何一帧心跳报文,则触发重新选举。

本文以 4 个节点为例说明选举过程,如图 2 所示。第一步:假设节点 1~节点 4 的优先级由高到低,节点 1 选举超时定时器最先过期,发起投票。第二步:节点

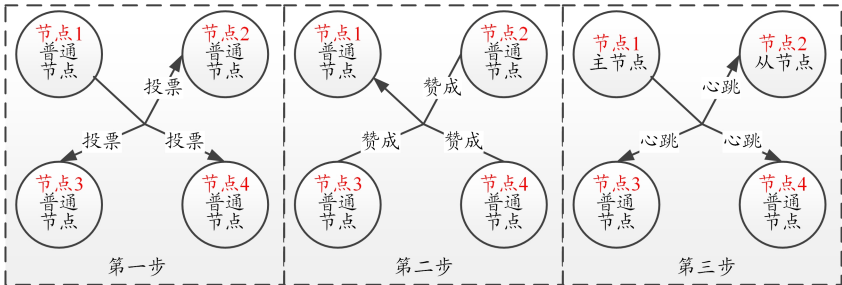


图 2 主从选举典型过程

2~节点 4 分别收到投票,发现节点 1 的优先级比自己高,都赞成节点 1 为主节点。第三步:节点 1 收到投票应答后,发现暂时没有节点拒绝自己当选,进入预当选阶段,等待投票反馈,如果未被任何节点拒绝,则正式成为主节点,并指定从节点。

SDN 控制器系统选举过程使用 3 种定时器,如表 2 所示。

表 2 选举过程使用的 3 种定时器及作用

定时器名称	配置节点	作用
选举超时定时器	从节点、普通节点	触发控制器节点选举。当节点在设定的选举超时时间内未收到主节点的任何一帧心跳报文时,判定主节点已经失效
预当选定时器	普通节点	判定普通节点是否赢得选举。在设定的预当选时间内,发起选举投票的节点如果未发现任何节点投拒绝票,那么发起选举投票的节点赢得本次选举
心跳超时定时器	主节点	主节点和其它节点保活。心跳超时定时器到期后,主节点向所有的其它节点发送心跳报文,其它节点收到心跳报文后应答主节点

1.2 主从复制

Raft 共识算法将日志的新旧程度作为选举领导者的依据之一,均采用复制日志的方式复制数据^[9],并非直接复制数据。日志复制增加了系统处理开销,同时会产生大量的日志,需要日志压缩和快照,实现非常复杂。主从复制基于优先级选举时不考虑日志新旧,不需要日志复制系统,直接复制数据,开销小、效率高和实现简单。当节点数量较多时,采取主节点只向从节点复制数据,从节点向所有普通节点复制数据,减轻了主节点的压力。无从节点或者普通节点时,不复制数据。

主从复制过程如图 2 所示,具体过程如下:主从复制采用增量和全量结合的方式,第一次采用全量复

制,之后采用增量复制。每一次复制操作收到确认后,开始下一次复制,未被确认的复制操作设置超时重传。

完成主从选举之后,从节点向主节点主动发起全量复制数据请求,收到主节点应答后,等待主节点开始全量复制数据。如果主节点未应答,从节点重新发起请求。主节点向从节点全量复制完成后,从节点向普通节点开始全量复制流程。全量复制完成后,发送结束标志。全量复制期间停止增量复制,全量复制完成之后,已发生数据变化的节点通过增量复制完成同步。增量复制同样采用由主节点复制给从节点,再由从节点复制给所有普通节点的复制方式。

1.3 主从分区容错性分析

基于优先级的主从选举提高了系统的可用性,无论如何分区,单个分区内只要有健康的节点,该分区就可以选举出唯一的主节点。但是,相对于 Raft 算法等共识系统,基于优先级的主从分布系统存在“脑裂”问题,即同一时刻出现 2 个独立运行的主节点。“脑裂”的发生和消除可能会导致系统运行异常。

消除“脑裂”问题的影响不能完全依赖主从复制,需要结合应用系统数据的涵义才能消除冲突。一般来说,SDN 控制器管理各种网络资源,主要包括以下几类:

①节点资源,全网所有设备及各个设备转发表项资源;

②链路资源,全网所有链路资源;

③带宽资源,全网所有带宽资源;

④路由计算,全网所有路由规划;

⑤虚拟资源,控制器独立分配使用的非实体资源。

上述网络资源由控制器集中统一分配使用,同时控制器系统连通性又依赖网络的连通性。网络及控制器系统分区后,总的网络资源被划分成若干部分,每个部分可能有控制器管理,也可能处于无管理状态。无论网络如何分区,前 3 种资源的总数固定不变,每个分区的资源数量也是一定的,且无交集。保证这 3 种资源的标识全网唯一,则不会出现冲突。第 4 种资源不固定,实现全网标识唯一,也不会出现冲突。第 5 种资源,是控制器独占的,出现标识冲突的概率很大,控制器应具备重新规划虚拟资源以解决冲突的能力。

2 2 种分布式系统时间测试

本文基于分组传送网(PTN)对 SDN 控制器主从分布式系统进行测试,部署 6 个控制器节点,如图 3 所示。每个节点均使用一台虚拟机运行控制器软件,虚拟机软硬件配置如表 3 所示。

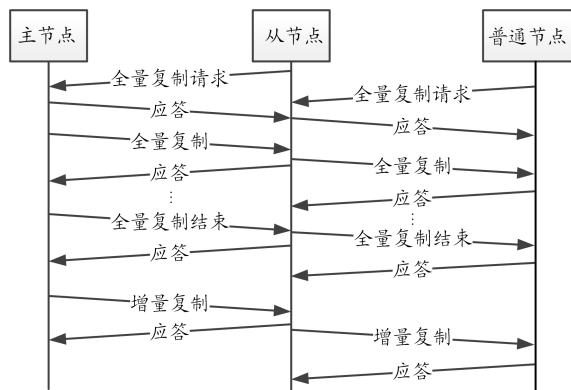


图 2 主从复制过程

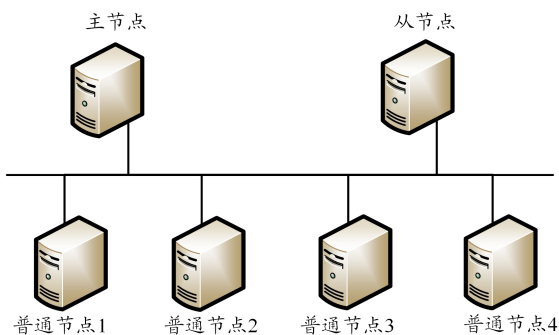


图3 节点分布情况

表3 软硬件配置

Linux 系统	CPU	内存	硬盘
Ubuntu 16.04 LTS	Intel I7-7700 3.60 GHz	1G	20G

控制器软件(包括主从模块)采用 go 语言 1.15.2 版本,节点之间传输协议采用 http1.1 版本,消息格式采用 protobuf 3.0 版本。为验证本文设计的优先级主从分布式系统复制性能,将与同时部署 6 节点 ETCD^[6]组成的 Raft 算法分布式系统做对比测试,ETCD 的版本号为 3.4.7。

2.1 选举时间对比

本文设计的优先级主从分布式系统初次选举时间固定为 10 s,即预当选定时器到期后成为主控制器,比 Raft 算法分布式系统选举时间慢。预当选定时器设置的意义在于尽量保证启动时间相差小于 10 s 的情况下高优先级的节点优先当选主节点。如果节点发起投票后收到了全部节点的赞成票,可以立即当选,仅需要 15 ms,与 Raft 算法分布系统选举时间接近,都是毫秒级的选举时间。

当主节点失效后,优先级主从分布式系统的从节点直接当选,选举时间为 0 s。而 Raft 算法在此种情况下需要重新选举。另外,如果 Raft 算法分布系统分布系统选举失败,例如出现“平票”的情况,此时需要重新选举,直到选举成功。因此,主节点失效的场景下,优先级主从选举比 Raft 选举时间至少快 15 ms,即一次 Raft 选举过程的耗时。

2.2 复制时间对比

ETCD 是基于 key 和 value 的轻量级分布式存储系统,不太适合单个 value 过大的写入。因此,测试时采用了长度为 208 字节的数据进行测试。在 6 节点间分别复制 100 条、1000 条数据的相同环境下,ETCD

Raft 算法领导者节点向其余 5 个跟随者节点复制数据时间分别为 400 ms 和 6000 ms 左右;优先级主从分布式系统主节点向从节点复制数据时间分别为 40 ms 和 370 ms 左右,从节点向其余 4 个普通节点同时复制相同数据的时间总共分别为 45 ms 和 400 ms 左右。由于主节点复制到从节点和从节点复制到普通节点是并发的,减去从节点响应主节点的时间,优先级主从分布式系统所有节点复制 100 条和 1000 条数据的时间分别为 70 ms 和 600 ms 左右。

2 种分布式系统复制方式的时间相差较大的原因是 Raft 算法分布式系统写入 ETCD 时采用了基本的单个 key 请求方式的接口,分别复制 100 条、1000 条数据时总共向 ETCD 分别提交了 100 次和 1000 次写操作;而主从选举算法则采用了批量复制方式,总共向优先级主从分布式系统分别提交了 2 次和 24 次写操作。ETCD 的复制时间大部分消耗在了多次网络传输过程中。

3 结束语

针对基于 Raft 共识算法的分布式系统在 SDN 控制器中应用存在的缺陷,本文设计了一种主从分布式系统,利用主、从和普通 3 种节点减少了控制器故障切换时间,减轻了主节点的压力,提高了控制器节点故障容忍度,复制效率接近主流的分布式系统 ETCD。实验结果表明:与传统的分布式系统相比,本文设计的主从分布式系统可用性和扩展性更强。

参考文献:

- [1] LAMPORT L. Paxos made simple[J]. ACM SIGACT News, 2001, 32(4): 18-25.
- [2] ONGARO D, OUSTERHOUT J K. In search of an understandable consensus algorithm[EB/OL]. [2021-03-15]. https://www.researchgate.net/publication/312391505_In_search_of_an_understandable_consensus_algorithm.
- [3] GILBERTS, LYNCH N. Perspectives on the CAP theorem[J]. Computer, 2012, 45(2): 30-36.
- [4] SAKIC E, SARDIS F, GUCK J W, et al. Towards adaptive state consistency in distributed SDN control plane[EB/OL]. [2021-03-15]. <https://ieeexplore.ieee.org/document/7997164>.
- [5] 王嘉豪,蔡鹏,钱卫宁,等. 集群数据库系统的日志复制和故障恢复[J]. 软件学报, 2017, 28(3): 476-489.
- [6] ETCD[EB/OL]. [2021-03-15]. <https://etcd.io>.