

集群内高效可靠的数据文件分发方案的设计

杨永凯^{1,2}, 彭明田^{1,2}, 王炜东¹

(1. 中国民航信息网络股份有限公司, 北京 101318;

2. 民航旅客服务智能化应用技术重点实验室, 北京 101318)

摘要:为了实现局域网集群内数据文件的高效和可靠的分发,设计了数据流和控制流相结合的数据文件分发方案。在数据流方面,以 ZeroMQ 为基础设计了 UDP 协议下的多播数据文件传输方案。在控制流方面,基于 TCP 协议设计了多播数据文件传输控制方案。传输控制方案包括监控模块、检测模块、调速模块和计速模块,监控模块和检测模块实现了对数据文件传输开关的控制,调速模块和计速模块实现了对数据文件传输速率的控制,两方面结合实现了对多播数据文件传输的有效调度,达到了数据文件分发过程中的高效和可靠的双重目的。工程实践表明,在保证一定的多播传输速率的基础上,可以提升数据文件分发的一次多播成功率,减少多播传输的轮次,能够实现集群内高效可靠的数据文件分发。

关键词:集群;文件分发;ZeroMQ;多播;传输控制

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2019)11-0163-05

doi:10.3969/j.issn.1673-629X.2019.11.033

Design of Efficient and Reliable Data File Distribution Solution in Cluster

YANG Yong-kai^{1,2}, PENG Ming-tian^{1,2}, WANG Wei-dong¹

(1. TravelSky Technology Limited, Beijing 101318, China;

2. Key Laboratory of Intelligent Passenger Service of Civil Aviation, Beijing 101318, China)

Abstract: In order to achieve efficient and reliable data file distribution in cluster, a data file distribution solution combining data flow and control flow is designed. In the data flow, based on ZeroMQ, the multicast data file transmission scheme based on UDP is designed, and in the control flow, a multicast data file transmission control scheme based on TCP is designed. The transmission control scheme includes monitor module, check module, shift module and meter module. The monitor module and check module realize the control of the data file transmission switch, the shift module and the meter module realize the control of the data file transmission rate, and the combination of the two aspects realizes the effective scheduling of the multicast data file transmission, achieving the dual purpose of high efficiency and reliability in the data file distribution. Engineering practice shows that, on the basis of guaranteeing a certain multicast transmission rate, the success rate of one multicast for data file distribution can be improved, the round of multicast transmission can be reduced, and the efficient and reliable data file distribution within the cluster can be realized.

Key words: cluster; file distribution; ZeroMQ; multicast; transmission control

0 引言

在信息化工程领域,经常需要面对局域网集群内一对多模式下的数据文件分发问题,即从一台生产者主机向集群内若干台订阅者主机进行数据文件的分发放传输。以民航信息化行业为例,航班的舱位状态报文(SSIM报)是一种高频的数据文件,全球所有航班的任何一次舱位变化(销售、取消、控制等)均会触发SSIM报文;而全球的民航运价数据在经过预处理后

会形成大体量的数据文件,部分文件压缩后仍达到几十G左右,这种数据文件在以约每小时一次的频率更新变化。在民航工程应用中,需要将这两类数据文件从生产者主机高效准确地分发到集群中数十甚至上百台订阅者主机上,进而在订阅者主机(即计算节点)上实现高性能的可用机票搜索计算服务。在局域网集群内的数据分发方案设计方面,高效和可靠往往意味着系统优化的两个不同方向:FTP、HTTP等基于TCP协

收稿日期:2019-01-08

修回日期:2019-05-09

网络出版时间:2019-06-27

基金项目:国家核高基课题(2014ZX010450101);国家发改委2014年云计算工程项目(发改办高技[2014]1799号)

作者简介:杨永凯(1977-),男,硕士,研究方向为民航信息化技术。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20190627.1054.020.html>

议实现的通信协议,可以实现准确可靠的数据传输,但是 TCP 协议只能支持一对一(端到端)的传输模式,在生产/订阅这种一对多模式下,意味着必须重复多次传输,占用大量的网络带宽资源,无法实现高效的目的;而 UDP 协议可以实现一对多的数据传输,能够大幅节省网络带宽资源,但是由于 UDP 协议本身是轻量级的无连接协议,无法保障传输的准确可靠性。在工程实践中,基于开源架构的消息中间件 ZeroMQ 和自行设计的控制逻辑构建了一套以 UDP 传输为主,辅以 TCP 协议进行控制的集群内高效可靠的多播数据文件分发方案,成功解决了高频报文或者大体量数据文件的一对多数据分发问题^[1-5]。

1 ZeroMQ 简介

ZeroMQ 不是单独的服务或者程序,仅仅是一套组件,其封装了网络通信、消息队列、线程调度等功能,向上层提供简洁的 API,应用程序通过加载库文件,调用 API 函数来实现高性能网络通信。由于 ZeroMQ 是用 C/C++ 开发地,并且其协议格式定义得很简单,所以它的性能远远高于其他消息队列中间件。

ZeroMQ 提供了 4 种基础消息通讯模式,分别是一对一结对模式 (Exclusive - Pair)、请求应答模式 (Request - Reply)、发布订阅模式 (Publish - Subscribe) 和推拉模式 (Push - Pull)。这 4 种模式总结出了通用的网络通信模型,使用者可以根据具体应用场景,使用其中的任何一种或多种模式,构建自己的通讯架构解决方案。文中方案所涉及的主要是发布订阅模式。

ZeroMQ 的发布订阅模式封装了 UDP 通信协议的实现细节,简化了系统设计和编程实现的开发工作,基于 ZeroMQ 组件可以简单快捷地实现基于 UDP 协

议的高效的多播数据分发。在 ZeroMQ 的发布订阅模式下,发布端负责单向分发数据,且不关心是否把全部数据发送给订阅端。如果发布端开始发布数据时,订阅端尚未连接上来,则这些数据会被直接丢弃。订阅端只负责接收数据,而不能反馈,且在订阅端处理速度慢于发布端处理速度的情况下,会在订阅端堆积数据。其结构如图 1 所示^[6-7]。

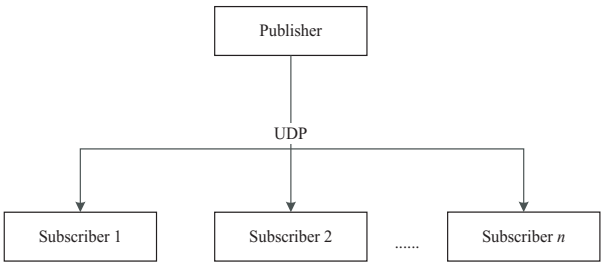


图 1 发布订阅模式

在工程实践中,除了需要考虑高效外,还要兼顾可靠性。为了保障数据文件分发传输的可靠性,必须在 UDP 多播的基础上增加控制流,以实现多播分发的有效调度,从而避免丢包、乱序、不一致、数据堆积等方面的问题。控制流的实现必须通过可靠的 TCP 协议来自行设计实现。

2 系统设计

2.1 总体结构

系统在设计实现上采取了数据流和控制流相对分离的设计机制,系统整体结构如图 2 所示。数据流基于 ZeroMQ 包装的 UDP 协议实现,图中用实线箭头表示;控制流通过自主设计的基于 TCP 协议的控制机制实现,图中用虚线箭头表示。数据流和控制流结合达到数据文件分发过程中高效和可靠的目的。

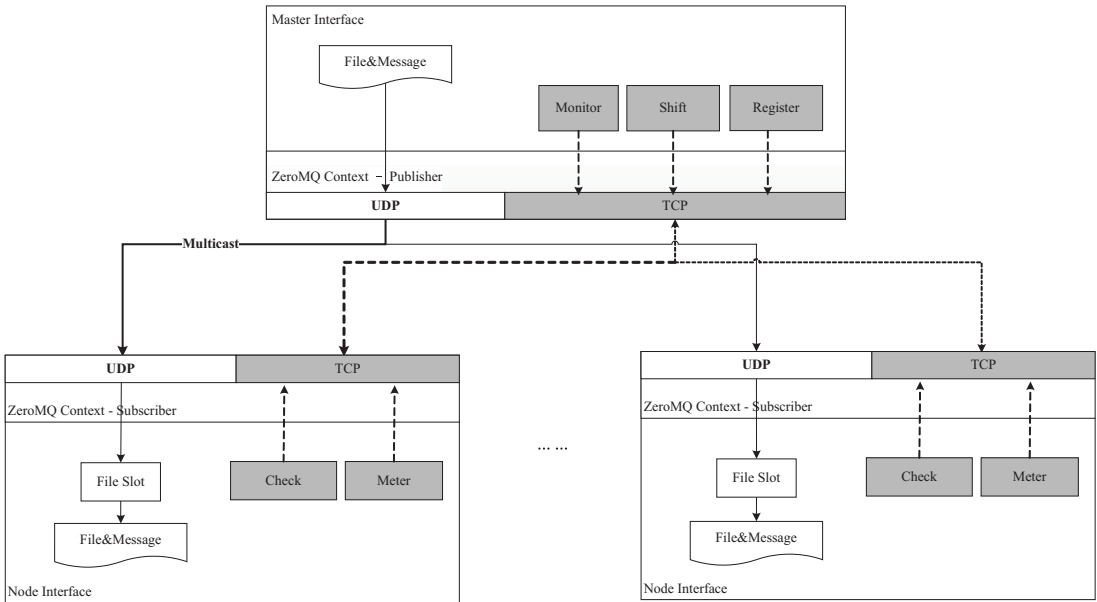


图 2 系统整体结构

将集群中负责数据文件分发的主机定义为 Master 主机,Master 主机实现发布者 Publisher 的功能,将集群中接收数据文件的主机定义为 Node 主机,Node 主机实现订阅者 Subscriber 的功能,Master 和 Node 之间通过 ZeroMQ 的发布订阅模式实现多播数据的传输,从而构成数据流的设计。

控制流的设计主要由 Master 主机上的监控服务 (Monitor) 模块、调速服务 (Shift) 模块以及 Node 主机上的检测服务 (Check) 模块、计速服务 (Meter) 模块组成。Master 主机上的 Monitor 服务与各 Node 主机上的 Check 服务配对实现针对数据流的传输开关控制,包括传输端口的控制 (打开及关闭) 以及数据流的启停等。Master 主机上的 Shift 模块与各 Node 主机上的 Meter 模块配对实现针对数据流的传输速率控制,兼顾网络环境和 Node 节点的处理效率,保障 Master 主机以合适的速率在集群内分发数据文件。

另外在 Master 主机设计了 Register 服务模块,当有新的 Node 主机连接入集群时,其本地的 Check 服务首先与 Master 主机的 Register 服务建立 TCP 连接,将新加入的 Node 主机的信息注册到 Master 主机,从而达到新加入的 Node 主机实时参与数据流多播的目的。

为了保障数据文件的完整,在 Node 主机设计了 File Slot 模块,用来开辟缓存存储大体量数据文件和高频报文文件的多播数据包,待传输完成后,将相应数据文件存入 Node 主机的指定位置。

高效可靠的多播数据传输方案的设计核心是控制流和数据流的交互模式,重点体现在控制流对数据流的传输开关控制和传输速率控制两个方面。传输开关控制和传输速率控制的目的是在可接受的传输速率下,尽可能降低同一份文件的多播分发轮次,最理想的情况是 Master 主机执行一次多播传输即可完成所有 Node 主机的数据文件接收,从而实现传输效率的最大化^[8-12]。

2.2 数据流的传输开关控制

传输开关控制的目的是确定一个文件是否成功完成了分发传输,即所有 Node 主机均接收到了该文件,并启动下一个新文件的分发传输。具体流程如下:当 Master 主机需要分发数据文件时,会通过 Monitor 服务告知各个 Node 主机节点的 Check 服务,随后 Check 服务会检查本地是否已经有 Master 主机所要分发的数据文件,如果没有则启用数据流的 Subscriber 服务,Subscriber 服务打开相应的 UDP 端口,准备接收数据;如果有则不启用 Subscriber 服务,不打开相应的 UDP 端口。Node 主机的 Check 服务会将最后的检查结果通过 TCP 连接告知 Master 主机,Master 主机会检测所

有 Node 主机的返回结果,判断集群中是否有节点需要接收当前的数据文件,如果有则启用本地的 Publisher 服务,Publisher 服务会打开相应的 UDP 端口通过多播分发数据文件;如果没有则不启用 Publisher 服务,不进行当前数据文件的分发,也标志着 Master 主机已经完成了当前数据文件的分发。综上所述,Monitor 服务的操作流程如图 3 所示,Check 服务的操作流程如图 4 所示^[10-11]。

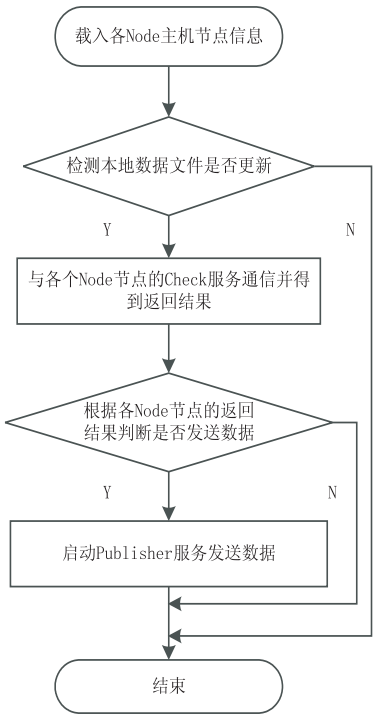


图 3 Monitor 服务流程

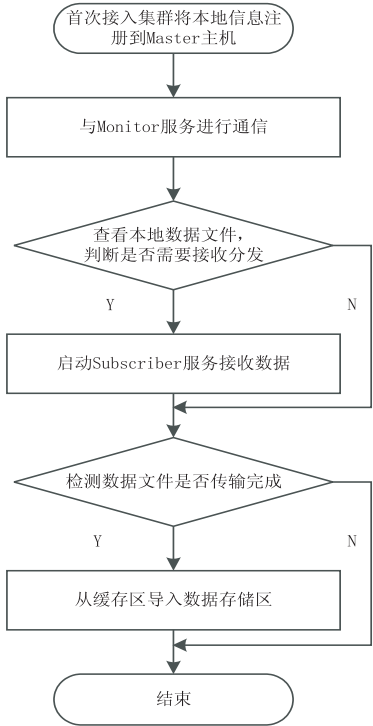


图 4 Check 服务流程

2.3 数据流的传输速率控制

传输速率控制的目的是保障 Master 主机的数据传输速率能够和 Node 主机的数据处理速率适配,从而提升多播传输的一次成功率。根据集群网络的带宽情况和 Master 与 Node 主机的硬件情况,确定多播传输速率的上限和下限数值,在数据流传输过程中,传输速率始终保持在上限和下限区间内,这个速率区间可以定义为集群的高效多播区间。超过上限可能会频繁出现 Node 接收失败的问题,导致同一个数据文件的多轮次分发,低于下限可能导致多播传输速率过低,文件分发时间过长。

具体流程如下: Master 主机的 Shift 服务读取 Config 配置信息,包括多播速率的上下限、初始速率等,之后将相关信息发送给 Publisher 服务和 Node 主机上的 Meter 服务;根据数据流的传输开关控制机制,

在本轮次多播分发启动后,Publisher 服务会按照初始速率通过多播分发数据,Node 主机的 Subscriber 服务负责接收多播数据;如果在接收数据的过程中,Subscriber 服务出现数据积压的情况会反馈给 Meter 服务,或者当监控到网络繁忙的情况后,Meter 服务会主动和 Master 主机上的 Shift 服务通信,Shift 服务会调整 Publisher 的多播速率以适应所有 Node 的处理速率;如果 Shift 服务将速率调整到速率下限后仍旧有 Node 主机反馈存在数据积压的情况,则通知相应 Node 主机的 Meter 服务停止接收本轮次多播分发,并提升整个集群多播传输速率;被停止的 Node 主机可以等待当前文件的下一轮次多播来完成文件接收,等待的过程同时也是该 Node 主机恢复数据处理的过程。传输速率控制的设计如图 5 所示^[11-12]。

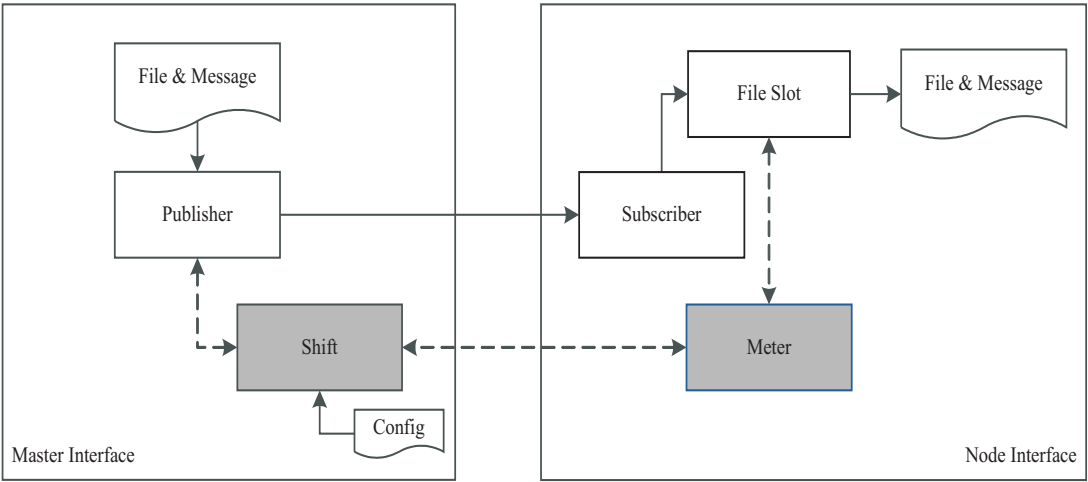


图 5 传输速率控制的设计

3 系统实现和工程验证

基于上述数据流和控制流相结合的设计方案,在工程上进行了编码实现和实践验证,具体的开发环境和部署环境如表 1 所示。

表 1 工程实践的应用环境

参数	取值
编程语言	C++
编译环境	Linux Redhat v6.5
主机类型	x86 服务器
CPU 核数	16core
内存	512 G
硬盘	2.9T SSD
MTU 配置	9 000
集群网络	万兆局域网
Master 数量	2 台
Node 数量	100 台
多播速率上限	100 MiB/s
多播速率下限	30 MiB/s

目前该套系统已经部署在生产环境,负责集群内大体量文件的分发和高频报文的分发任务。为了验证该方案针对大体量文件分发的有效性,任意截取了连续 15 天的文件分发数据作为 15 份样本进行方案验证,每 1 天的数据组成 1 份样本。在每天内的每个整点启动 1 个轮次多播分发,根据实际业务的更新情况,每轮次分发的数据文件总数在 10 至 15 个左右,最大的文件 40 G 左右,最小的 10 M 左右,每轮次需分发的文件总容量大约为 90 G 左右。

定义 n 次分发成功率参数分别为 R_1, R_2, \dots, R_n :

$$R_n = \sum_{i=1}^n (i \text{ 次分发成功的文件数}) / \text{文件总数}$$

如果 R_1 越接近于 1,且 R_n 越早收敛到 1,则表明该方案的控制流有效性越高。对上述工程实践中样本的汇总如表 2 所示: R_1 的平均值为 0.983 3,绝大多数样本 R_2 即收敛为 1,只有 1 个样本在 R_3 收敛为 1,另外每组样本的多播平均传输速率接近设定的上限 100 Mib/s。从实际工程角度看,该方案的控制流有效性

较高,可以满足实际项目的需求^[13-15]。

表 2 工程实践的验证结果

样本 序号	文件 总数	1 次分发 成功的 文件数	2 次分发 成功的 文件数	2 次以上 分发成功 的文件数	R ₁	平均传 输速率/ (MiB/s)
1	292	286	6	0	0.979 5	89
2	310	306	4	0	0.987 1	96
3	317	312	5	0	0.984 2	92
4	300	296	4	0	0.986 7	96
5	282	276	6	0	0.978 7	95
6	284	277	6	1(3 次)	0.975 4	96
7	304	300	4	0	0.986 8	85
8	310	304	6	0	0.980 6	96
9	293	288	5	0	0.982 9	97
10	324	320	4	0	0.987 7	92
11	324	318	6	0	0.981 5	96
12	330	326	4	0	0.987 9	92
13	289	282	7	0	0.975 8	89
14	287	284	3	0	0.989 5	82
15	297	292	5	0	0.983 2	91
汇总	4 543	4 476	75	1	0.983 3	---

该方案同时还承担高频报文的分发任务,报文的大小基本在 1 至 5K,非常适合于多播模式的传输,基于控制流的辅助,在生产集群中高频报文的分发成功率 R₁基本约等于 1,不再赘述。

4 结束语

综上所述,该方案通过基于 UDP 多播的数据流和基于 TCP 协议的控制流相结合的方式,有效解决了集群内数据文件分发过程中的高效和可靠兼得的问题,即适用于集群内大体量的数据文件分发,又适用于集群内高频报文文件的分发,在工程实践中具有很重要的典型意义。

参考文献:

[1] STEVENS W R. TCP/IP 详解(卷一)[M]. 北京:机械工业出版社,2000.

[2] COULOURIS G,DOLLIMORE J,KINDBERG T. Distributed system:concept and design[M]. 5th ed. London:Person Education Asia Ltd,2012.

[3] 刘盛志. 基于 UDP 的高效大文件分发应用协议设计与实现[D]. 长春:吉林大学,2011.

[4] 高 磊. 分布式文件传输系统的关键技术研究[D]. 哈尔滨:哈尔滨工程大学,2013.

[5] 张昆朋,吕延庆,谢华成. 基于多播的以太网文件传送协议的设计与实现[J]. 制造业自动化,2012,34(20):36-38.

[6] 李冬琦,高 键. 基于 ZeroMQ 的 SOA 分布式通信系统设计[J]. 计算机与数字工程,2018,46(6):1257-1262.

[7] 蒲凤平,陈建政. 基于 ZeroMQ 分布式系统[J]. 电子测试,2012(7):24-29.

[8] 董淑松,康慕宁. 基于可靠组播文件传输协议的设计与分析[J]. 科学技术与工程,2010,10(9):2195-2198.

[9] 陶振江,邢 卫,鲁东明. 一个面向 CATV 网络的可靠多播文件传输系统[J]. 计算机工程,2006,32(5):116-118.

[10] 薛鹏飞,胡荣贵,胡劲松. 基于 ZeroMQ 的分布式系统通信方法[J]. 计算机应用,2015,35(S2):34-37.

[11] HINTJENS P. ZeroMQ:messaging for many applications[M]. [s.l.]:O'Reilly Media,2013.

[12] 张 杰,王丽娜,赵 媛,等. 基于 ZeroMQ 消息通讯的多源空中目标跟踪处理平台设计[J]. 计算机测量与控制,2018,26(9):219-222.

[13] STEVENS W R,RAGO S A. Advanced programming in the UNIX(R) environment[M]. [s.l.]:Addison-Wesley Professional,2005.

[14] GRIGORIK I. High performance browse networking[M]. [s.l.]:O'Reilly Media,2013.

[15] 李永胜,黄兰红,刘红军. 基于 UDP 协议的多文件传输[J]. 广西民族大学学报:自然科学版,2007,13(2):68-71.