

文章编号: 2095-2163(2020)01-0285-06

中图分类号: TP393

文献标志码: A

面向大规模网络的实时高可靠消息分发技术

刘飞扬, 叶麟, 余翔湛, 赵俊达

(哈尔滨工业大学 计算机科学与技术学院, 哈尔滨 150001)

摘要: 随着互联网规模的飞速发展,网络传输的数据量也在不断增加,日益增长的网络数据对网络带宽和服务器性能提出越来越高的要求,所以解决网络数据分发的实时性与可靠性是一个需要解决的问题。本文提出了一种基于AMQP技术与P2P技术,更好地解决网络数据在分发过程中无法兼顾实时性与可靠性的问题,提升数据分发的速率的架构。文章中对现有的AMQP与P2P等相关技术进行分析,通过采用将AMQP技术与P2P技术相融合的方法,将不同规模的文件分配给不同的传输方式分发,提升分发的速率,实现分发的实时性与高可靠性。

关键词: AMQP; P2P; 网络分发

A real-time and high reliable message distribution technology for large scale networks

LIU Feiyang, YE Lin, YU Xiangzhan, ZHAO Junda

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

[Abstract] With the development of the Internet, the amount of data transmitted by the network is increasing. The increasing number of network data demands higher and higher network bandwidth and server performance. So it is a problem to solve the real-time capability and reliability of the network data distribution. This paper proposes an architecture based on AMQP technology and P2P technology to better solve the problem that network data cannot take into account both real-time capability and reliability in the distribution process, and improve the speed of data distribution. This paper analyzes the existing technologies such as AMQP and P2P. By adopting a method that integrates AMQP technology and P2P technology, the paper uses different transmission methods to distribute files of different sizes. Through these methods, the rate of distribution is increased, and real-time capability and high reliability of distribution are achieved.

[Key words] AMQP; P2P; content distribution

0 引言

如今,高速发展的互联网已经成为生活中不可或缺的一部分,如何实现大规模、实时和高可靠的数据传输一直是互联网研究的热门方向。在互联网出现伊始,由于数据量都比较小,大规模的数据分发不会产生太大的时间与资源消耗。但是随着流媒体的不断发展和网络用户的不断增加,蓝光视频、软件更新和在线视频流等的分发越来越需要更多的网络资源,基于内容的服务和应用呈现指数增长。据估计,视频流量将占有所有消费者互联网流量的82%(2021年)^[1]。而网络内容分发需要提供较低的延迟、快速的传输速度和服务的可靠性。学术界提出了内容分发网络^[2](Content Delivery Network, CDN)和点对点通信^[3](Peer-to-Peer, P2P)的相关概念。

内容分发网络是一个通过策略部署在不同网络

与区域的内容分发整体系统,可通过增加缓存服务器来提高服务质量。缓存服务器通常位于网络的边缘,离用户仅有几跳路由的距离。即内容分发网络架设有源服务器与区域缓存服务器,在用户请求资源时,将请求重定向至距离用户最近的区域缓存服务器,使用户可以就近取得所需的内容,提高用户访问网站的响应速度。同时,区域缓存服务器是内容提供商(Internet Content Provider, ICP)源服务器的一个缓存镜像。这样,内容持有者通过内容分发网络服务提供商传播内容,向用户提供优质的服务。

点对点通信网络是一个节点分布在互联网各个角落的内容共享网络,并通过成员之间的内容共享来提供服务。在点对点通信网络的架构中,每一个节点同时扮演服务器和客户端两种角色,因此就既是内容请求者、又是内容提供者,还能其它节点通过

基金项目: 国家自然科学基金(61771166);国家重点研发计划(2016QY05X1000)。

作者简介: 刘飞扬(1994-),男,硕士研究生,主要研究方向:信息安全、P2P网络、网络安全;叶麟(1982-),男,博士,讲师,主要研究方向:网络安全、P2P网络、网络测量和云计算等;余翔湛(1973-),男,博士,教授,博士生导师,主要研究方向:信息内容安全、网络安全、物联网安全等;赵俊达(1994-),男,硕士研究生,主要研究方向:P2P网络、匿名网络。

收稿日期: 2018-06-05

相互合作来交换内容。点对点通信网络中的节点都是自愿加入的,可以随时加入和离开网络。这使得越来越多的人自愿地加入点对点通信网络。点对点通信网络中的节点越多,整个网络的分发性能就越强。点对点通信网络具有较高的可扩展性、容错性、鲁棒性,适合内容共享网络的部署。

综上所述,在内容分发网络的架构之下,结合点对点传输网络的共享传输,是本文实现的一个方向。在架构上,可以将多个独立的 CDN 使用 P2P 技术互相连接起来^[4],也可以将中心化的 P2P 网络与传统 CDN 技术融合而成的 HCDN 模型,并将边缘服务器作为中心化 P2P 中的索引服务器^[5],在此基础上,文献[6]沿用了模型,进行了一些仿真测试,但都没有可行的实现。本文基于内容分发网络的网络拓扑结构,将点对点传输网络融入其中,使用高级消息队列 AMQP 进行控制信息的传输,连接整个网络,并负责部分小规模数据的分发,使用 P2P 网络进行大规模数据的分发,实现数据分发系统。

1 相关工作

1.1 AMQP 技术

高级消息队列协议(Advanced Message Queuing Protocol, AMQP),是一个架设在计算机网络应用层之上,提供统一消息服务的高级消息传输队列协议^[7]。该技术需要架设中转服务器,系统运行时,发送端将消息投递至中转服务器,由于发送端投递关键字的不同,中转服务器根据这些不同的关键字将消息分发至不同的消息接收端,实现发送端与接收端的异步互联。同时,中转服务器端负责消息的维护和整体系统的负载均衡。该技术通过不同的编程语言编写,可以在不同的编程语言中运行。RabbitMQ 是基于 AMQP 标准实现的一个典型的开源消息中间件。

通俗来讲,AMQP 是架设在计算机网络应用层之上,自恰的一套传输协议。而其典型设计当属 RabbitMQ 服务器,是由 Erlang 语言编写的,用于集群消息分发的开源程序。RabbitMQ 拥有几种不同的消息分发传输模式,分别是消息直传模式、广播/订阅模式和接收关键字模式。其中,直传模式生产者将生产的消息推送至消息队列,所有关注该队列的消费者都可以接收该消息,接收后该消息消失。广播/订阅模式中,生产者在产生消息之后,将该消息投递至所有订阅该消息的接收队列之中,供消费者接收。在接收关键字模式中,生产者只将生产的消息投递至监听关键字符合的队列之中。

1.2 P2P 网络

P2P(peer-to-peer)网络,又称对等式网络,是一种没有中央服务器,依靠众多的用户节点组成的一个庞大的互连网络。该类网络的设计作用在于,依托众多的用户节点,互联互通传输资源,以减少局部网络流量过大、负载过高为宗旨,同时还可降低单点故障带来的问题。在 P2P 网络中,每个节点既是用户节点,又是服务节点,符合网络共享的思维。具体来说,P2P 网络分为结构化的 P2P 网络、非结构化的 P2P 网络和松散结构的 P2P 网络。相应地,结构化的 P2P 网络在节点之间互相连接,构成有一定规则的拓扑结构以供资源搜索时使用,典型代表有 Chord、Kademlia 等。非结构化的 P2P 网络节点之间相互连接,但没有形成有结构的网络拓扑,节点间通行依靠广播形式来完成,典型代表有 Gnutella。松散结构的 P2P 网络介于有结构的 P2P 网络与无结构的 P2P 网络之间,通常会有中央服务器存在,典型代表有 Freenet。

综上探讨后可知,结构化的 P2P 网络效率最高,且以 Kademlia 网络结构为代表。实现这种网络结构的主要程序有 BitTorrent 和 eMule。BitTorrent 是一款由 python 语言编写的开源程序,BitTorrent 协议是架构在 TCP/IP 层上的一个传输通信协议,是一个应用层协议。在 BitTorrent 协议中,可以通过 2 种方式请求资源。一种是通过结构化的 DHT(Distributed Hash Table)网络,另外一种是通过种子文件来请求资源。本文中用到的是改进后的 DHT 网络来分发资源。

2 框架设计及实验分析

2.1 系统架构

从架构层面来说,系统由 2 个模块组成,分别是:控制模块与传输模块。对于前者,在控制层面来讲,控制信息是由 AMQP 来传输发送的,主要由 RabbitMQ 来实现。而传输模块分为 2 部分,也就是:AMQP 传输部分和 P2P 传输部分。

从网络拓扑结构来说,系统可简单地看作一个三层网络拓扑结构,如图 1 所示。在网络结构中,多媒体数据是多媒体资源的生产者,负责生产多媒体资源,通常是各类多媒体平台。网络拓扑结构的第一层是中央服务器,监控着整个系统的资源调取分发,存储着所有的多媒体资源。网络拓扑的第二层是中转服务器,存储有资源副本,架设在各个不同的地区,掌控着向本地区用户的资源分发。网络拓扑的第三层是终端用户,是多媒体资源的请求者和使

用者。

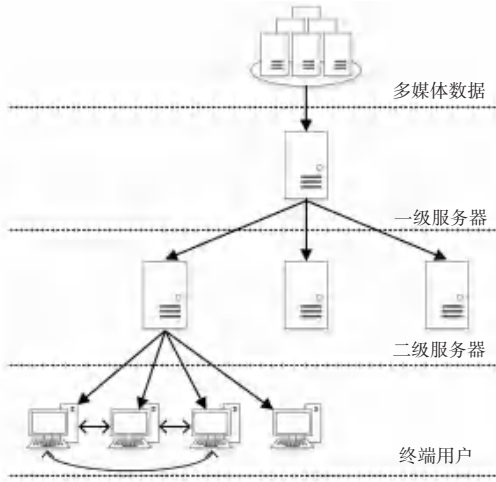


图 1 消息分发系统的网络架构

Fig. 1 Network architecture of message distribution system

2.2 系统任务调度

为了使整个系统获得较好的分发速度,需要使每个服务器都能发挥优质效能。本系统使用高级消息队列 RabbitMQ 进行各个服务器与客户端之间的调度命令传输。在系统运行过程中,维护一个控制信息消息通道,主服务器与地区服务器之间通过此通道传输控制命令。

当系统运行初期,服务器通过控制信息队列向客户端广播一条控制信息,确定当前在线的客户端及其状态。当服务器需要向客户端分发消息时,服务器端通过队列传输接下来所要发送的文件的信息,各个客户端接收到服务器端发送的消息信息后,返回一个接收反馈信息,然后根据所传输的消息的信息,建立对应的消息传输队列。服务器端在消息发送前,会发送一个确认消息,确认当前消息队列的建立情况,满足需求后向该通道广播对应的消息。当所需要发送的消息过大时,系统会使用 P2P 方式进行多媒体文件的分发,会先发送一个种子的长度信息给各个需要接收该消息的客户端,此后客户端反馈接收情况。接下来,服务器端向客户端传输即将需要分发的消息文件的种子文件,客户端收到完整的种子文件后,给客户端发送一个反馈,基于此客户端将开启本地 P2P 下载进程,通过该种子文件中的信息下载该种子文件。其消息队列任务调度如图 2 所示。

研究可知,系统在运行时维护一个当前在线节点列表,当有分发文件到达时,系统读取新来的文件,如果该文件之前已获发送,则不作处理;否则,根据文件的大小,使用对应分发操作对文件进行分发。

关于该算法的主要逻辑可表述如下。

While (True) :

```

BroadcastToNode()
FlushNodeStatus()
nodeList = LoadOnlineNode()
mediaFiles = LoadMultiMediaData()
for newFile in mediaFiles:
if HasBeenDistribute( newFile ) == True:
    continue
if GetFileSize( newFile ) < Threshold:
    for node in nodeList:
        DistributeByAMQP( node , newFile)
else :
    for node in nodeList:
        DistributeByP2P( node , newFile)
UpdateFileStatus( newFile)
sleep( section)

```

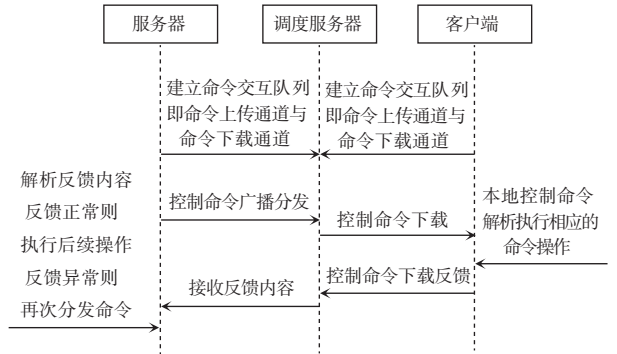


图 2 消息请求队列任务调度流程

Fig. 2 Task scheduling process of message request queue

2.3 AMQP 传输模块

在内容分发网络中,对于小文件,使用 P2P 网络进行分发往往会加大网络中的资源消耗,且分发效率提升不明显。因此,本系统拟采用 RabbitMQ 队列对小文件进行分发。

在文件分发过程中,首先服务器读取当前多媒体文件需要下发的节点,将需要下发文件的文件信息通过控制传输通道传输给当前在线的需要接收的节点。当客户端接收到传送而来的文件基本信息后,将自己的节点信息和接收到的文件配置信息验证反馈给服务器,同时建立相应的文件接收队列。而后服务器将通过控制通道,收集已经建立了通道的用户节点的信息,向建立完成接收通道的节点分发文件内容。在客户端,异步接收该文件,并在接收结束后返回一个正确接收反馈。研究设计流程如图 3 所示。

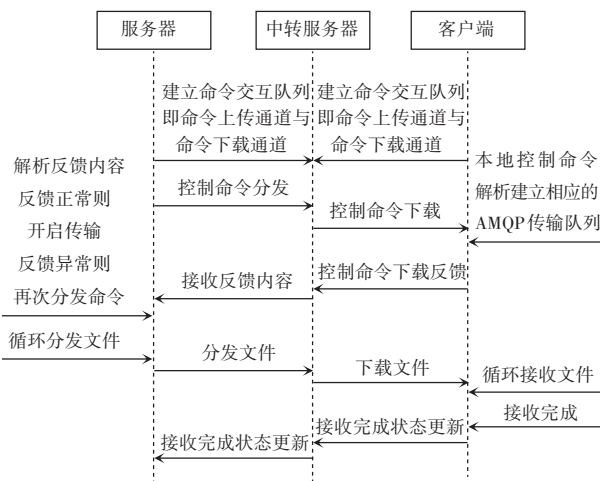


图3 队列文件分发流程

Fig. 3 The process of queue file distribution

在使用 RabbitMQ 对小文件进行分发时,可以选择 2 种分发方式,分别是直接分发模式和订阅/广播分发模式。如果只需要向部分节点传输文件,优先使用直接分发模式,逐一向这些节点分发文件,这样可以减少广播控制信息带来的流量,但是在投递分发的过程中会增加一定的传输工作量。如果需要向全部节点分发文件,可优先使用订阅/广播模式,这种情况下,广播控制信息对整个网络的影响被缩小,同时订阅/广播模式使用较少的文件副本,减少了文件磁盘操作的工作量。

2.4 P2P 传输模块

在内容分发网络中,对于较大的文件,如果利用一对多的传输方式,使得单一节点的网络带宽成为整个系统的性能瓶颈,且一对多的传输方式,就会出现发送节点的网络拥塞,导致分发效率降低。而 P2P 传输网络则完美解决了这一研究课题。结构化的 P2P 网络 BitTorrent,使用的是多对多的传输方式,每个节点在请求未接收到的文件分片的同时,也在分发已经接收到的文件分片^[8-9]。这样虽然某个单一节点的接收速率有走低迹象,但却降低了整个系统的接收时间。因此,使用 P2P 网络进行分发可以加快分发效率,并明显减少全局分发时间。因此,本系统将采用修改后的 BitTorrent 网络对大文件进行分发。在文件分发过程中,采用如下逻辑:

Step 1 中央服务器读取目录,扫描需要分发的文件,并生成文件的种子文件。

Step 2 中央服务器将文件的种子文件信息下发至需要接收该文件的节点。中央服务器接收到接收节点接收通道建立完成的反馈后,使用该通道传输种子文件。

Step 3 中央服务器将文件信息通过通道下发至二级服务器。待二级服务器建立接收通道后,中央服务器向二级服务器下发文件。

Step 4 中央服务器接收到二级服务器接收完成的反馈,向所有终端用户分发接收文件的文件信息,节点在接收到文件信息后,开启 BitTorrent 下载,解析文件的种子信息,向二级服务器请求下载。

Step 5 终端节点在文件接收完成后,向服务器端发送接收完成反馈。

Step 6 如果大部分节点接收完成配置,则返回 Step1; 否则等待终端节点接收。

在 BitTorrent 网络中,能够做到以较高的传输速度共享文件,究其原因即在于其结构化的网络拓扑和文件分片共享机制。当请求产生时,客户端通过迭代的方式向整个 DHT (Distribute Hash Table) 网络查询该内容文件的所有者,并向部分所有者请求该文件。其中,不同于原来的 BitTorrent 网络,查询到的资源拥有者是随机获取的,不同的请求者查询到的资源拥有者也是不同的。在文件的传输过程中,BitTorrent 网络的传输策略是优先向不同的请求者传输不同的文件分片,即优先分发稀有分片。这样,可以更早地开启请求者之间的互传,使整个系统的内容文件整体分发效率尽快达到最优。在研究过程中的消息请求分片传输则如图 4 所示。

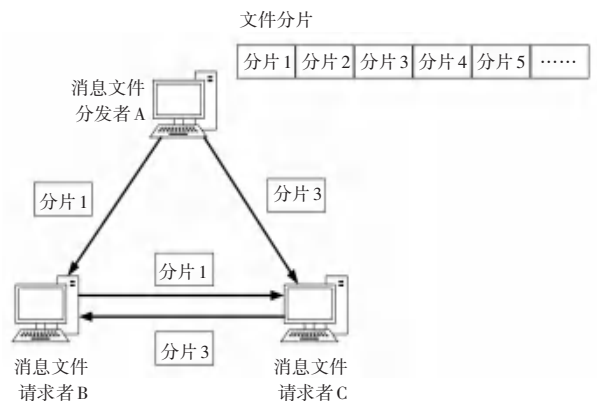


图4 消息请求分片传输

Fig. 4 Piecewise transmission of message requests

2.5 实验分析

实验主要选择了平均下载速度作为测试结果的主要衡量标准,这一参数能够体现系统整体的分发性能,受到波动的影响较小。测试数据的平均速率为文件大小与文件从分发开始到所有节点均接收到的时间的比值,而非单一节点接收到文件。所有实验都在数台服务器之间设计实现,实验过程中使用的平台配置信息见表 1。

表 1 实验平台配置信息

Tab. 1 Experimental platform configuration information

配置名称	配置规格
CPU	Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60HZ
内存	32 G
网卡	Intel Corporation I350 Gigabit Network
操作系统	Ubuntu 16.04.4 LTS
内核	Linux version 4.4.0-62-generic

文中,针对使用 RabbitMQ 或使用 BitTorrent 网络在不同节点数量下的分发传输速率,可得到的研究运行后的实验结果如图 5 所示。由图 5 可以看出,在分发较小规模的文件(100 MB)时,当节点数目较小时,RabbitMQ 传输占有绝对的优势,而当请求节点数目增加时,RabbitMQ 传输也仍是优于 BitTorrent 传输的,只是优势差距在逐渐缩小。在节点数量不断增加的情况下,由于机器性能与网络的原因,传输速率有所下降。由此说明 RabbitMQ 适合一定规模下的小文件传输,具有较高的传输效率。

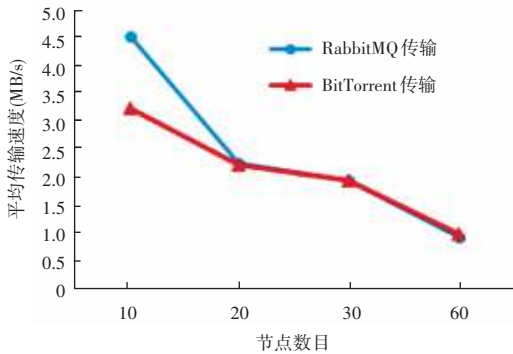


图 5 不同数量节点下的 100 MB 资源分发速率图

Fig. 5 100 MB resource distribution rate diagram under different number of nodes

在此基础上,研究进一步得到了不同数量节点下的 1 000 MB 资源分发速率如图 6 所示。由图 6 可以看出,当传输文件的大小由 100 MB 提升到 1000 MB 时,在节点数目为 10 时,由于系统资源利用率并未达到最大,RabbitMQ 传输速率依然高于 BitTorrent 传输;但当节点数目不断增加时,系统资源负载均衡的情况下,BitTorrent 网络的分片互传机制已逐渐显现出优势,BitTorrent 网络的传输已高于 RabbitMQ 网络。由此说明 BitTorrent 更适合大文件的传输,在节点众多的情况下传输效率颇高。

最后,针对 RabbitMQ 传输、BitTorrent 传输与本系统传输不同大小文件的速率则如图 7 所示。测试数据的计算的时间为从文件分发开始到结束。由图 7 可以看出,整个文件分发的过程中,由于本系统对

RabbitMQ 传输和 BitTorrent 传输进行了部分简化与修改,并在此基础上增加了一些传输控制,故而对全规模段文件均可获得较好的传输速率。在传输小文件时,初始的控制信息传输在整体传输过程中占了较高的比例,所以在文件只有 1MB 大小时,传输速率偏低。在传输大文件时,控制消息所占的比例降低,传输速度波动趋于平稳。本次研发设计充分发挥了节点性能,一定程度上降低了服务器的负载,提升了整个传输过程的效率。

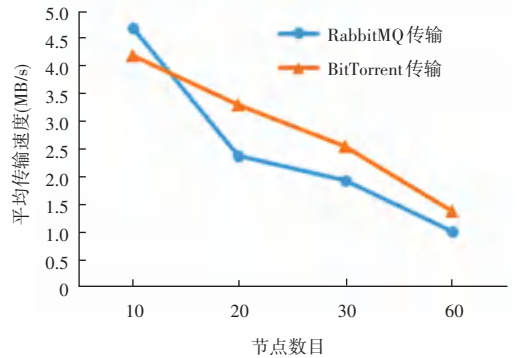


图 6 不同数量节点下的 1 000 MB 资源分发速率图

Fig. 6 1 000 MB resource distribution rate graph under different number of nodes

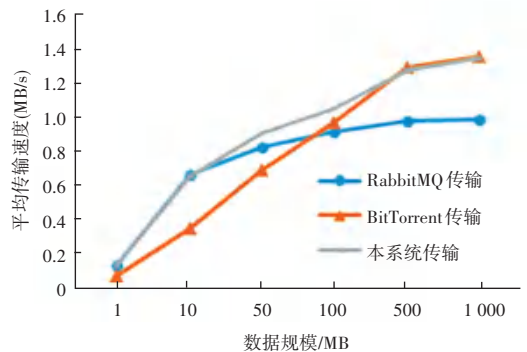


图 7 单机模拟 60 节点分发不同规模资源速率图

Fig. 7 The speed of a single machine simulating 60 nodes to distribute different scale resources

3 结束语

本文从计算机网络入手,围绕着大规模网络下的可靠的数据请求分发问题,首先针对当前流行的 AMQP 和 P2P 网络,阐述了这 2 种网络的特性,并展开了优劣势分析;其次,根据这些网络的特点,提出了将这两者进行结合的研究设想,进而探讨给出了对应的系统设计与架构;最后,基于设计的系统框架,实现了结合 RabbitMQ 与 BitTorrent 的消息分发系统,而且通过实验分析了新系统的性能,并与原有的 2 种单独 RabbitMQ 网络或者 BitTorrent 网络进行了分发对比,实验结果显示了新架构的优越性。

(下转第 293 页)