

文章编号: 1005—0523(2009)01—0082—05

BitTorrent模型原理分析

王 珏, 周 莉

(华东交通大学 信息工程学院, 江西 南昌 330013)

摘要: BitTorrent是一个技术框架, 在 P2P的基本架构下定义了一个高效的文件分发模型, 该模型不仅规定了文件分发中的不同实体, 而且对实体之间的信息交互方式, 所遵循的规范进行了严格定义, 最为关键的是该协议规范中提出了分断共享的概念. 在文件传输关键环节中提出的算法, 使得在不稳定网络中的大文件分发更为高效.

关 键 词: BT下载; 元信息文件; 追踪服务器; 种子; 结点

中图分类号: TP393.09

文献标识码: A

BitTorrent是一个基于 P2P(Peer to Peer)的文件分发协议, 其用途是多用户的文件上传和下载. 所以, 基于 BitTorrent协议的下载方式被称为 BitTorrent下载, 简称“BT下载”. 传统的 HTTP、FTP下载方式需要在网络中存在一个文件服务器, 所有的文件下载客户端需要连接到该服务器进行下载, 增加了文件服务器及其所载网络的负荷, 容易造成网络使用效率的降低甚至是网络的瘫痪^[1].

P2P下载的特点是, 下载客户端不连接到唯一的文件服务器, 而是相互连接, 每个客户端从其它客户端下载文件片断的同时, 也为其它客户端上传其已经拥有的该文件的片断. 但是, BT下载与传统的 P2P文件分发模型不同的是, 在 BT下载模型中设立了一个下载协调者, 负责协调各个结点之间的信息, 解决了传统 P2P下载模型中下载客户端之间传递协调信息造成的网络拥堵, 极大地提高了 P2P分发模型的下载效率^[2].

1 BitTorrent模型概述

BitTorrent下载中的所有实体以及实体之间为了完成文件下载功能而进行的交互中所遵循的技术规范统称为 BitTorrent模型^[3].

BitTorrent模型需要以下实体^[3]:

(1) 一个普通的 Web服务器; (2) 一个元信息 (meta info)文件 (扩展名为 .torrent); (3) 一个追踪 (Tracker)服务器; (4) 一个文件提供者, 被称为“种子”(seed); (5) 运行 BitTorrent客户端程序的下载终端, 被称为结点 (peers).

要完成一个完整的 BitTorrent模型的部署, 其过程从功能上可以划分为 3个步骤:

(1) 追踪服务器的部署; (2) 文件信息的发布; (3) 结点之间的交互式下载^[4].

1.1 追踪服务器的部署^[5]

BitTorrent下载过程中需要一个追踪服务器. 追踪服务器负责帮助结点获取其它结点的信息, 并协调不同结点之间的信息. 追踪服务器和结点之间使用 HTTP协议进行交互, 结点向追踪服务器注册下载的文件、

收稿日期: 2008—11—28

基金项目: 江西省教育厅科技资助项目 (GJJ08255)

作者简介: 王 珏 (1978—), 男, 江西南昌人, 讲师, 硕士, 研究方向为计算机网络与信息安全.

IP地址、端口以及相关信息,追踪服务器告诉结点其它结点的注册信息,结点利用这些信息相互之间建立连接。

BitTorrent下载模型部署的第一步就是建立并运行一个追踪服务器。追踪服务器必须有一个 URI `http://域名(IP地址):端口号/announce`

1.2 文件信息的发布^[5]

原始结点将文件的基本信息以及追踪服务器的 URI 创建一个元信息文件,并将该文件上传到 Web 服务器,并提供一个链接。但是,原始结点必须运行 BitTorrent 客户端程序,以便其它结点能够与其建立连接。原始结点和其他结点的角色是相同的,所不同的是只上传数据,并不下载,所以被称为“种子”。

1.3 结点之间的交互式下载^[6]

结点从普通 Web 服务器获得元信息文件,通过该文件获得追踪服务器的 URI 根据该 URI 向追踪服务器注册,并从追踪服务器得到其他结点(包括“种子”结点)的注册信息。结点根据获得的信息与其他结点之间建立连接,开始下载和上传数据。

BitTorrent 下载方式中,将下载文件划分为若干个片断(pieces),结点下载文件时,选择自己需要的某些片断进行下载,同时将自己拥有的片断提供给其他感兴趣的结点。当某个结点完成了所有文件片断的下载后,停止下载,但是上传仍然在继续。由于拥有一份完整的文件,所以这个结点也成为“种子”,一直到退出 BitTorrent 下载。

BitTorrent 下载模型中,完整的文件可以由“种子”提供的,也可以是多个结点所拥有的片断的并集。

2 元信息文件结构

元信息文件是一个包含下列信息的文件。

2.1 info

info 是描述下载文件的结构,有两种形式:一种是针对无目录结构的单个文件的情况(单文件模式);另一种是针对可能包含子目录树的多 BT 文件的情况(多文件模式)^[2]。

单文件模式下,info 结构 C 语言定义如下

```
struct single_file_info {
    int length; //下载文件的长度
    char md5sum[32];
    //文件的 32 个字符的 MD5 验证码
    char* name; //文件名
    int piece_length; //每个文件片断的长度
    char* pieces;
    //SHA1 散列值串,每个散列值对应一个片断
}
```

多文件模式下,info 结构 C 语言定义如下

```
struct multi_files_info {
    struct single_file_info* files;
    //指向文件列表的指针,列表中的每个元素为上述的单文件 info 结构
    char* name; //包含所有文件的根目录的名称
    int piece_length; //每个文件片断的长度
    char* pieces; //片断的 SHA1 散列值组成的字符串
}
```

2.2 announce

announce 的值是一个字符串,指明追踪服务器的 URI

除了 info 和 announce 两个字段,元信息文件中还包含一些关于元信息文件本身信息的字段:

3 追踪服务器通信协议

追踪服务器接受结点的 HTTP GET请求, 请求包括来自结点的定期“注册”, 这些“注册”消息使得追踪服务器拥有关于下载结点的所有统计信息. 追踪服务器的响应的信息, 包括一个结点列表, 这个列表中的信息使得结点能够参与下载.

3.1 结点的 HTTP GET请求

在结点发给追踪服务器的 GET请求中的参数如下^[1]

- ✓ info-hash: 元信息文件中 info关键字 值的 20个字符的 SHA1散列值.
- ✓ peer-id: 标识结点唯一的 20个字节的字符串.
- ✓ port: 结点监听的端口号.
- ✓ uploaded: 上传的字节数.
- ✓ downloaded: 下载的字节数.
- ✓ left: 还需要下载的字节数.
- ✓ event: 值必须是 “started, completed, stoppe”其中之一, 或者为空, 等同于未使用.
- ✓ ip: 指明客户端机器的真实 IP地址.
- ✓ numwant: 可选参数, 指明结点希望从追踪服务器得到的其他结点的个数. 缺省值通常是 50.

3.2 追踪服务器给结点的响应消息

追踪服务器响应消息结构的 C语言定义如下

```
union answer_msg {
char* failure_reason;
/请求失败原因的错误消息, 若该关键字有效, 则其它关键字就不需要
struct success { /请求成功时返回的消息结构
int interval /结点发送请求间隔秒数
int complete /“种子”结点的个数
int incomplete /非“种子”结点的个数
char* * peers /描述结点信息的字符串数组
int port /结点使用的端口号
}
}
```

4 结点线路协议

结点线路协议 (PeerWire Protocol)是基于 TCP协议的应用层协议, 是结点之间进行数据交换所使用的协议.

4.1 连接状态信息

每个结点必须知道和其他结点之间每个连接的状态信息, 包括下列两种状态:

choked: 远程结点是否“堵塞 (choked)”与该结点的连接. 一个结点堵塞了该结点, 该结点的任何请求都不会得到应答, 直到被“疏通 (unchoked)”.

interested: 远程结点“关注 (interested)”该结点. 当连接被疏通时, 远程结点即将开始请求获得文件. “阻塞”和“关注”都是双向的, 不仅仅是某个结点针对其他结点的, 其他结点也针对该结点.

4.2 握手 (Handshake)消息

“握手”必须是结点发送的第 1个消息, 消息的 C语言格式定义如下

```
struct Handshake {
int pstrlen; /字符串 pstr的长度
char* pstr; /协议标识
char reserved[8]; /保留
```

```
char info_hash[20]; //结点的 SHA1 散列值
char peer_id[20]; //结点标识
}
```

4.3 普通消息格式

协议中其他消息都采取下列格式

```
struct Message {
    int len; //长度前缀
    int id; //消息 ID
    char* payload; //数据有效负载
}
```

结点线路协议中常见的消息如下:

✓ keep-alive 消息: 格式 <len=0000>

功能: 当结点间没有数据交互时, 发送此消息保持结点间的连接.

✓ choke 消息: 格式 <len=0001><id=0>

功能: 发送该消息“阻塞”某个结点.

✓ unchoke 消息: 格式 <len=0001><id=1>

功能: 发送该消息“疏通”某个结点.

✓ interested 消息: 格式 <len=0001><id=2>

功能: 发送该消息“关注”某个结点.

✓ not interested 消息: 格式 <len=0001><id=3>

功能: 发送该消息, 表示不“关注”某个结点.

✓ have 消息: 格式 <len=0005><id=4><片断信息>

功能: 表示结点拥有的片断的信息.

✓ bitfield 消息: 格式 <len=0001+有效负载长度><id=5><信息>

功能: 在“握手”消息之后, 其他消息之前发送. 表示已成功下载的块信息.

✓ request 消息: 格式 <len=0013><id=6><片断信息>

功能: 请求获得片断, 有效负载表示片断信息.

✓ piece 消息: 格式 <len=0009+有效负载长度><id=7><片断信息和数据>

功能: 传送数据;

✓ cancel 消息: 格式 <len=0013><id=8><片断信息>

功能: 取消片断请求, 有效负载和 request 消息相同.

5 算法

5.1 片断选择策略^[7]

选择一个好的顺序来下载片断, 对提高性能非常重要. 一个差的片断选择算法可能导致所有的片断都处于下载中, 或者是另一种情况, 没有任何片断被上传给其它结点.

片断选择的第一个策略是: 一旦请求了某个片断的一部分, 那么该片断剩下的部分下一次请求时被优先选择. 这样, 可以尽快获得一个完整的片断.

对一个结点来说, 在选择下一个下载的片断时, 通常选择的是其他的结点所拥有的最少的那个片断, 也就是所谓的“最少优先”策略. 这种技术, 确保了每个结点都拥有其它结点最希望得到的那些片断. 这也确保了将那些分布较广的片断放在最后下载. 从而减少了这种可能性: 某个结点拥有的文件片断不被其它结点所“关注”.

“最少优先”策略的一个例外是在下载刚开始的时候, 第一个下载结点从“种子”那里选择第一个片断是随机的, 直到第一个片断下载完成, 才切换到“最少优先”策略.

5.2 最后阶段策略

当下载基本完成时, 最后少数几个片断的传输会变得十分缓慢. 为了提高速度, 客户端向所有结点发送请求, 请求获得所有尚未下载的部分. 这样很容易造成网络拥塞^[7], 为了避免这种情况, 当一个片断接收完

成时, 结点会向该片断的所有提供结点发送一个 cancel 消息.

5.3 阻塞算法和尝试性疏通

当某个结点被其它结点“阻塞”时, 尝试改变“阻塞”状态或连接尚未连接的其它结点的策略实现成为阻塞算法. 一个好的阻塞算法必须符合如下几个标准:

(1) 必须限制同时上传数, 保证 TCP 良好的性能; (2) 必须避免在阻塞和疏通之间“颤动”一样的频繁转换; (3) 必须和被允许下载的其它结点进行交互; (4) 应该偶尔尝试连接从未连接过的结点, 找出比现有连接更好的连接, 这叫做尝试性疏通.

6 结语

B i f T o r r e n t 模型以无结构 P2P 模型为基础, 结合相关传统网络技术 (例如 HTTP 协议), 为文件高效安全地分发提供一个稳定的应用平台.

B i f T o r r e n t 模型广泛应用于文件分发和文件下载. 在国内所有的 P2P 应用体系中, 基于 B i f T o r r e n t 模型的应用占据了大半的用户群. 2006 年底发布的中国互联网统计报告显示: 在中国 1.11 亿网民中有 27.8% 的人使用过 BT 软件^[8]. 以 B i f T o r r e n t 模型为基础的应用软件有 B i f T o r r e n t, B i C a m e t, B i S p i r i t 等.

同时, 由于 B i f T o r r e n t 模型提供文件高速分发, 也逐渐开始应用于网络流媒体系统, 例如视频点播、媒体广播等. 以其为基础的流媒体软件有 E Z T V、T V a n t s 等.

B i f T o r r e n t 针对传统 P2P 模型中结点和资源缺乏统一管理和结点间交互的效率问题进行了改进, 引入追踪服务器, 对结点和资源信息统一进行描述和发布, 并引入传统计算机网络中 W e b 服务器作为信息的发布平台, 使 P2P 网络与传统网络间有机结合. 从而能够将应用于传统计算机网络的管理策略、安全技术等引入 P2P 网络, 极大地促进了 P2P 网络在管理和安全上的发展.

参考文献:

- [1] 汪 燕, 柳 斌. B i f T o r r e n t 协议分析及控制策略 [J]. 实验技术与管理, 2006, 23(1): 54—56.
- [2] 陈宝钢, 张 凌, 等. P2P 文件共享系统行为特性分析和研究 [J]. 计算机应用, 2006, 26 (12): 2 809—2 812.
- [3] 孙建华, 王战国, 等. 基于 S—RTT 策略的 B i f T o r r e n t 文件共享模型 [J]. 计算机应用, 2008, 28 (2): 367—370.
- [4] Elliott Mitchell. B i f T o r r e n t protocol specification v1.0 [EB/OL]. <http://wiki.theory.org/BifTorrentSpecification>, 2007—01—12.
- [5] Bram Cohen. Incentives build robustness in B i f T o r r e n t [EB/OL]. <http://www.bittorrent.com/bittorrentecon.pdf>, 2008—04—17.
- [6] 王续宗, 廖建新, 等. 一种基于 MS 的 B i f T o r r e n t 文件分布系统架构 [J]. 计算机工程, 2008, 34 (1): 273—275.
- [7] 王利娥, 黄 昆, 等. B i f T o r r e n t 系统中一种近似全局最少优先的文件块选择策略 [J]. 计算机应用研究, 2008, 25 (5): 1 419—1 421.
- [8] 陈贵海, 李振华. 对等网络: 结构、应用与设计 [M]. 北京: 清华大学出版社, 2007. 33—43.

An Analysis on Model Principle of B i f T o r r e n t

WANG Jue ZHOU Li

(School of Information Engineering East China Jiaotong University Nanchang 330013, China)

Abstract B i f T o r r e n t is a technology frame class which defines an effective file distribution model. The model presents not only the different units in file distributing but strictly defines the communication method among the units and the specification about it. The most important thing is the concept of piece sharing in the specification. The algorithm in file transmission makes the big file distribution in unstable web more effective.

Key words B i f T o r r e n t; downloading; meta-info; file; tracker; seed; peer

(责任编辑: 刘棉玲)