



内容分发

15 minutes de visite de la Content Delivery

主讲

赵楚洋、张腾、刘建东

组员

薛润桢、王昊、齐森茂、游二炜、左建辉

山东大学 计算机科学与技术
17级菁英班

The development of the content distribution...

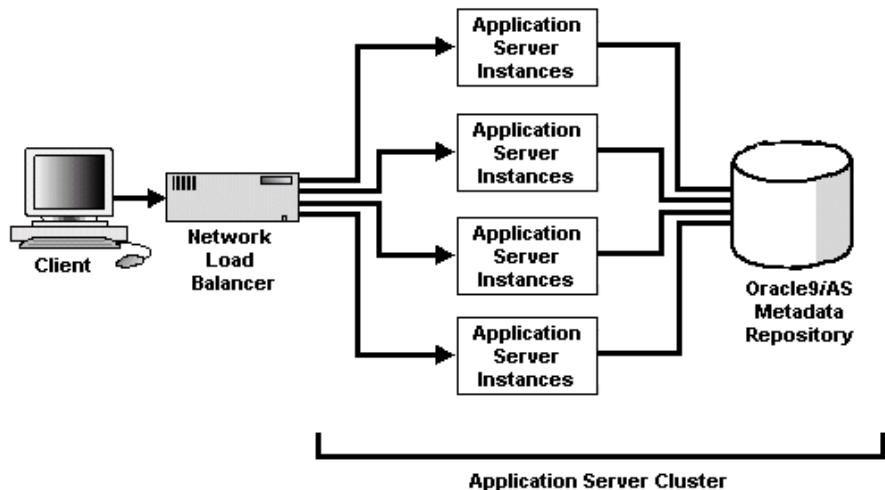


Why are we developing so many technologies?



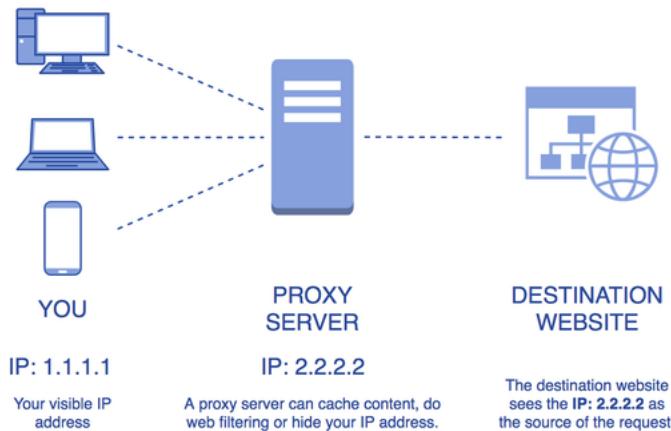
服务器农场

Server Clustering



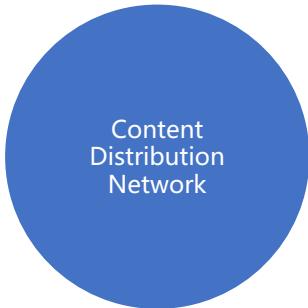
Web代理

Web Proxy



How to handle this problems?

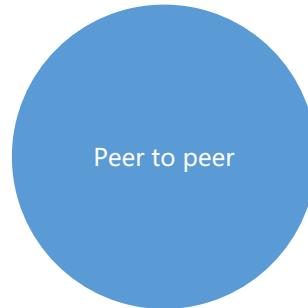
We got two ways to solve the problems.



Content
Distribution
Network

CDN

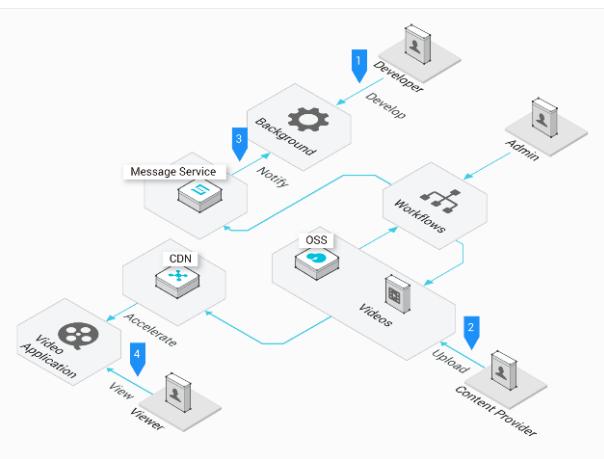
The ISPs set up a set of distributed machines in some location to enhance the preference of the internet.



Peer to peer

P2P

A group of computer pool their resources and provide content to each other.



内容分发网络

Content Delivery Network , CDN

内容分发网络

Content Delivery Network , CDN



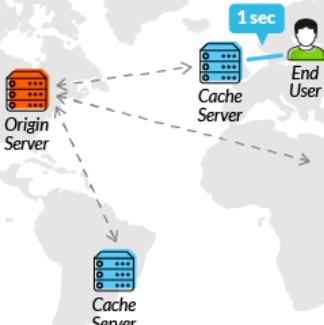
内容分发网络

Content Delivery Network , CDN

简要定义

Definition

一种扭转传统 Web 的缓存方法，根据用户的位置，选择最近的缓存节点服务器获取资源的资源分发形式。



内容分发网络

Content Delivery Network , CDN

CDN的诞生 The Birth of CDN

1. 传统 Web 资源分发方式的性能制约、安全隐患
2. 客户端对 Web 资源获取的效率体验



Physical
Distances



Number of
Intermediate Nodes



Amount of
Traffic



Transmission
Mediums

Every Second Counts

Studies show that a second-long delay causes a 7% drop in conversions, an 11% drop in page views and a 16% drop in customer satisfaction.



内容分发网络

Content Delivery Network , CDN



So What Can a CDN Do for Me?

Modern CDNs can handle numerous IT tasks,
helping you to:

- ✓ Improve page load speed
- ✓ Handle high traffic loads
- ✓ Block spammers, scrapers and other bad bots
- ✓ Localize coverage without the cost
- ✓ Reduce bandwidth consumption
- ✓ Load balance between multiple servers
- ✓ Protect your website from DDoS attacks
- ✓ Secure your application And more



CDN 的优点

The Advantages Of CDN

1. 解决了传统 Web 资源分发方式的性能制约、安全隐患
2. 提升了客户端对 Web 资源获取效率及体验

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title> 学生选课管理信息系统 </title>
8     <link href="./static/css/bootstrap/bootstrap.css" rel="stylesheet">
9     <link href="./static/css/nprogress/nprogress.css" rel="stylesheet">
10 </head>
11 <body>
12     <!-- ... -->
13     <script src="./static/js/jquery/jquery-1.12.4.js" type="text/javascript"></script>
14     <script src="./static/js/bootstrap/bootstrap.js" type="text/javascript"></script>
15     <script src="./static/js/nprogress/nprogress.js" type="text/javascript"></script>
16     <script src="./static/js/scmis/common.js" type="text/javascript"></script>
17     <script src="./static/js/scmis/login.js" type="text/javascript"></script>
18 </body>
19 </html>
```

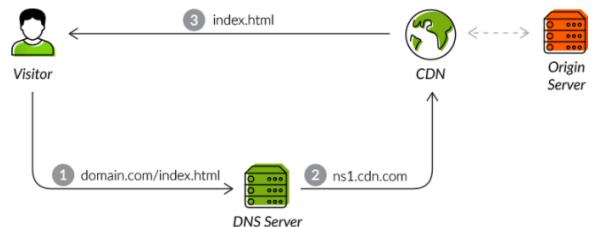
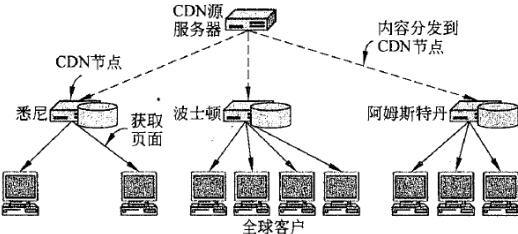
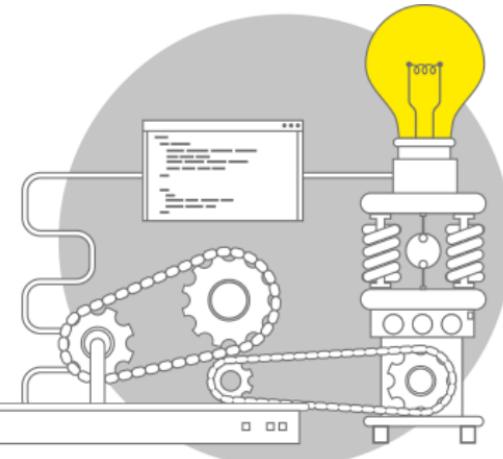
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title> 学生选课管理信息系统 </title>
8     <link href="https://cdn.staticfile.org/twitter-bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
9     <link href="https://cdn.staticfile.org/nprogress/0.2.0/nprogress.min.css" rel="stylesheet">
10 </head>
11 <body>
12     <!-- ... -->
13     <script src="https://cdn.staticfile.org/jquery/1.12.4/jquery.min.js" type="text/javascript"></script>
14     <script src="https://cdn.staticfile.org/twitter-bootstrap/3.3.7/js/bootstrap.min.js" type="text/javascript"></script>
15     <script src="https://cdn.staticfile.org/nprogress/0.2.0/nprogress.min.js" type="text/javascript"></script>
16     <script src="./static/js/scmis/common.js" type="text/javascript"></script>
17     <script src="./static/js/scmis/login.js" type="text/javascript"></script>
18 </body>
19 </html>
```

内容分发网络

Content Delivery Network , CDN

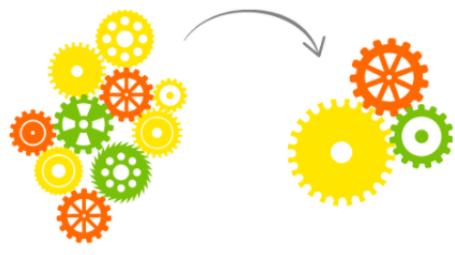
CDN 的技术 The Technology of CDN

1. CDN 分发树，及其简单实现——镜像
2. DNS 重定向 CDN



内容分发网络

Content Delivery Network , CDN

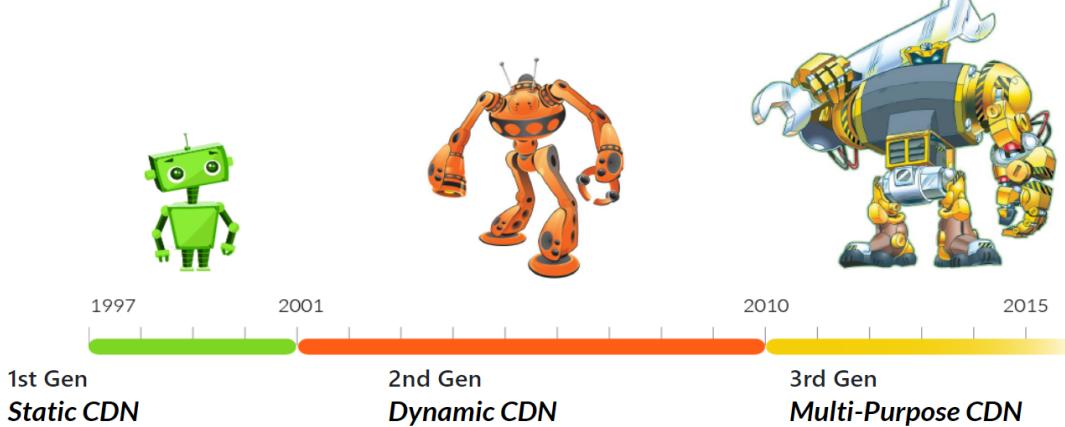


■ JavaScript files ■ CSS files ■ Images

CDN 的发展

The Development of CDN

我们更多关注于用 CDN 分发静态资源，比如图像和视频，但 CDN 发展已能很好地支持动态页面创建、流媒体等更多内容





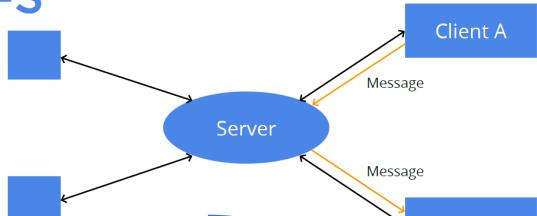
对等网络

Peer-to-Peer , P2P

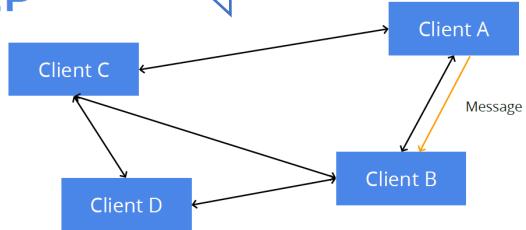
对等网络

Peer-to-Peer, P2P

C-S



P2P



简要定义
Definition

一种颠覆了传统 C-S 模型的网络架构思想，在P2P结构中，不存在中心节点，每个节点既是信息消费者，也是信息提供者。

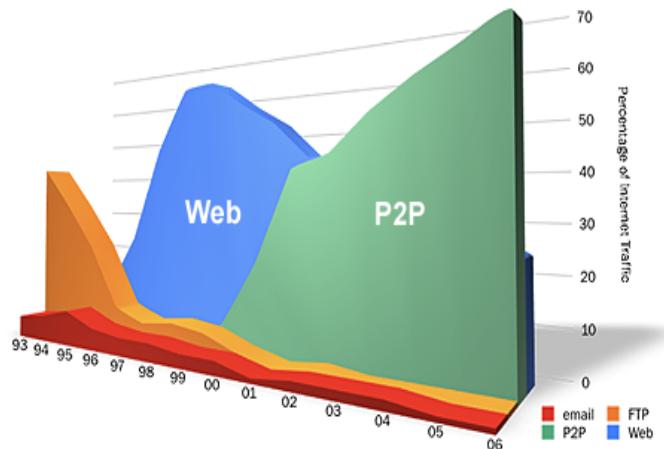
对等网络

Peer-to-Peer, P2P

P2P的优点

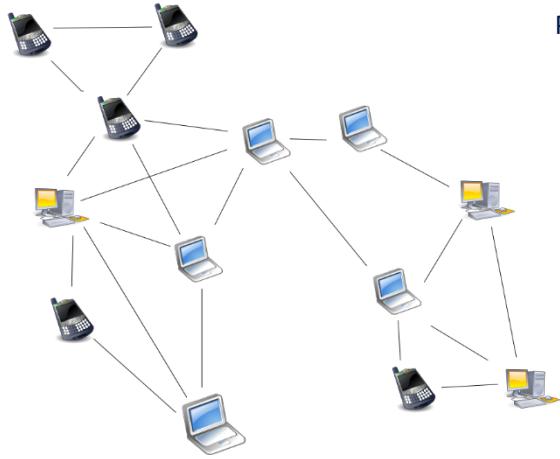
The Advantage of P2P

1. 权力下放 Decentralization
2. 鲁棒性 Robust
3. 可拓展性 Extensibility
4. 隐私保护 Privacy protection



对等网络

Peer-to-Peer, P2P



无结构化网络：无特定组织方式，节点间随机连接

优点：

- 搭建简便
- 灵活，允许局部优化
- 高鲁棒性健壮性，大量用户加入和离开足够稳定

缺点：

- 泛洪导致的资源分发延迟和不稳定

路由与资源探寻

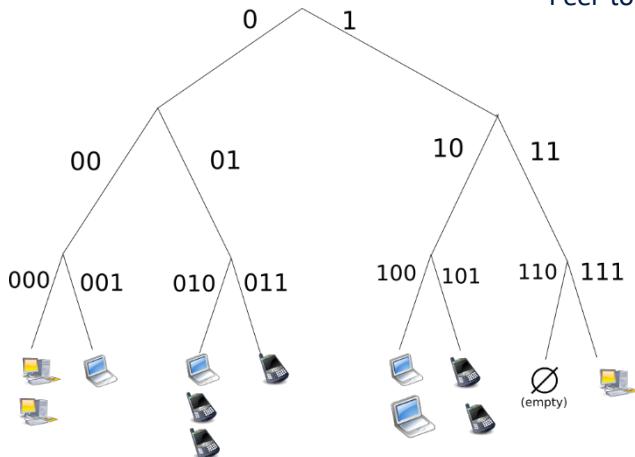
Routing and Resource Discovery

基于对等网络中节点间的连接方式、资源标记和定位方式，将网络分成三种：

1. 无结构化 unstructured
2. 结构化的 structured
3. 两者混合 hybrid between the two

对等网络

Peer-to-Peer, P2P



结构化网络：有特定拓扑关系

- 节点维护邻居的信息 (鲁棒性较无结构化网络低)
- 使用分布式哈希表 (Distributed Hash Table , DHT) 来识别和定位节点以及资源。
- 增强了节点对资源的探寻效率 (尽管资源是稀有的)

路由与资源探寻

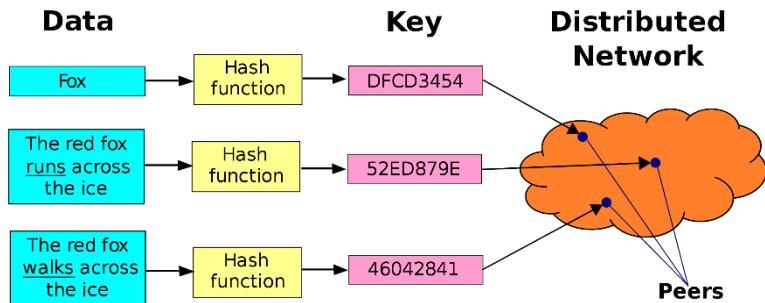
Routing and Resource Discovery

基于对等网络中节点间的连接方式、资源标记和定位方式，将网络分成三种：

1. 无结构化 unstructured
2. 结构化的 structured
3. 两者混合 hybrid between the two

对等网络

Peer-to-Peer, P2P



分布式哈希表：

1. 计算 Data 的 SHA1 散列值 k
2. 将 $\text{put}(k, \text{Data})$ 送给分布式散列表中的任意参与节点。
此消息在延展网络中被转送，直到抵达在“关键值空间分割”中被指定负责存储关键值 (k, Data) 的节点
3. 其他节点只需要将 $\text{get}(k)$ 分布式散列表中的任意参与节点，以此来找与 k 相关的数据。

路由与资源探寻

Routing and Resource Discovery

基于对等网络中节点间的连接方式、资源标记和定位方式，将网络分成三种：

1. 无结构化 unstructured
2. 结构化的 structured
3. 两者混合 hybrid between the two

对等网络

Peer-to-Peer, P2P

无结构化与结构化混合：

- 有一个中心服务器帮助寻找周边节点，与Tracker不同，当节点找到资源后，中心服务器就没什么用了

路由与资源探寻

Routing and Resource Discovery

基于对等网络中节点间的连接方式、资源标记和定位方式，将网络分类成三种：

1. 无结构化 unstructured
2. 结构化的 structured
3. 两者混合 hybrid between the two



BitTorrent™

P2P的技术实现

Applications of P2P

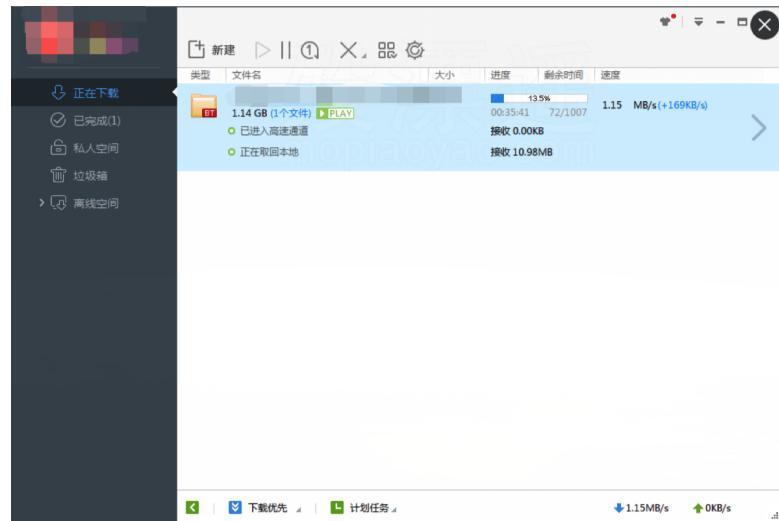
BT 1.0 — BitTorrent

简要定义

Defination

P2P文件共享通信协议，用于在Internet上分发数据和电子文件。

- 2001年由 Brahm Cohen 开发
- 目的：快速方便地共享文件
- 需要解决的问题：
 - (1) 对等用户如何找到其他对等用户下载内容？
 - (2) 如何鼓励对等用户上传内容给他人？



Thunder Download

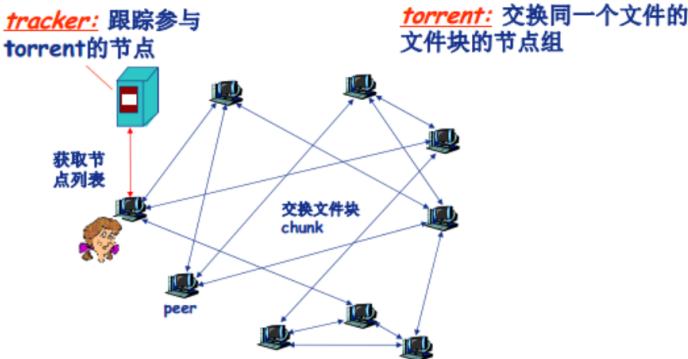
BT 1.0 — BitTorrent

问题1：

对等用户如何找到其他对等用户下载内容？

问题2：

如何鼓励对等用户上传内容给他人？



种子文件：

- 跟踪器 (Tracker) 的信息
- 完整文件的分块信息

Tracker :

· 一个服务器，维护着主动上传和下载该内容的所有其他对等用户列表

完整步骤：

- 获取种子文件
- 访问跟踪器找到对等用户
- 下载文件并将信息存储到跟踪器中

针锋相对策略 (tit-for-tat) :

· 将贡献率高的节点匹配下载速率更快的节点

BitTorrent 组成



- announce
- info
 - files
 - length
 - path
 - length
 - name
 - piece length
 - pieces

```
name: ubuntu-18.04-desktop-amd64.iso
filename: ubuntu-18.04-desktop-amd64.iso.torrent
comment: Ubuntu CD releases.ubuntu.com
date: 26.04.2018 04:58:28 PM (1524776308)
created_by:
files: (1)
  1: ubuntu-18.04-desktop-amd64.iso
size: 2 (1921843200)
announce: http://torrent.ubuntu.com:6969/announce
announce_list:
  - http://torrent.ubuntu.com:6969/announce
  - http://ipv6.torrent.ubuntu.com:6969/announce
info_hash: e4be9e4db876e3e3179778b03e906297be5
```

BT 2.0 — Distributed Hash Table

BitTorrent 必须依赖追踪器的存在才能正常工作，不是一个完全分布式的系统。

当追踪器故障或失效时，整个系统就会被破坏，因此我们进入了BT 2.0时代，分布式哈希表。



Distributed Hash Table

Trackerless Torrent

The King is dead,
long live the King!



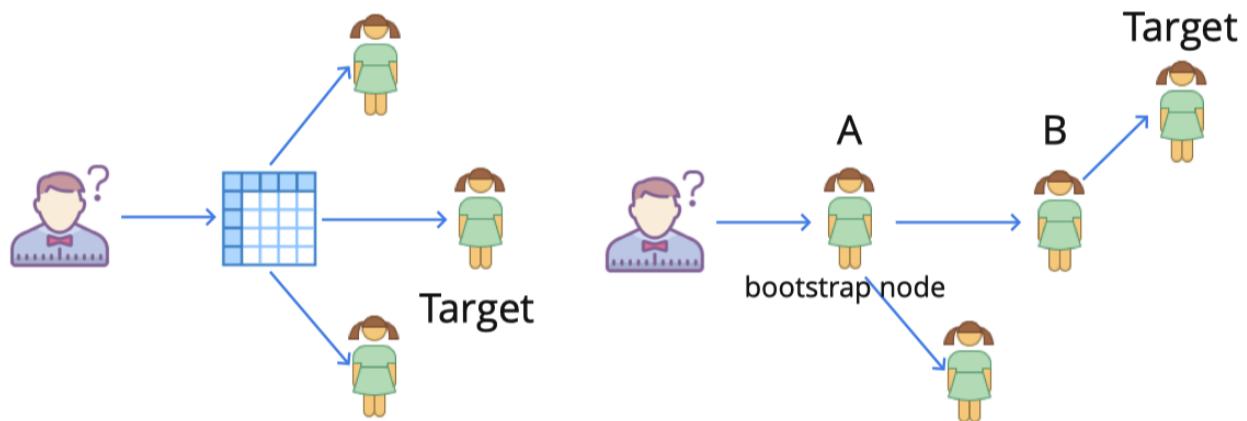
magnet:?xt=urn:sha1:YNCKHTQCWBTRNJIV4WNAE52SJUQCZO5C

BT 2.0 — DHT

· 在不需要服务器的情况下，每个客户端负责一个小范围的路由，并负责存储一小部分数据，从而实现整个DHT网络的寻址和存储。

· DHT网络协议 和 BitTorrent 是完全兼容的。

· 常用的BT软件允许同时连接DHT网络和Tracker，即在完全不连上Tracker服务器的情况下，也可以很好的下载，因为它可以在DHT网络中寻找下载同一文件的其他用户。



find_node: request

No	Time	Source	Destination	Protocol	Length	Info
92	7.183223000	192.168.18.33	212.129.33.50	BT-DHT	146	
96	9.897595000	192.168.18.33	212.129.33.50	BT-DHT	143	
97	10.148189000	212.129.33.50	192.168.18.33	BT-DHT	317	reply=8 nodes

Frame 96: 143 bytes on wire (1144 bits), 143 bytes captured (1144 bits) on interface 0

Ethernet II, Src: Tp-Link_T_f6:73:05 (14:75:90:f6:73:05), Dst: Netgear_3a:80:86 (c4:04:15:3a:80:86)

Internet Protocol Version 4, Src: 192.168.18.33 (192.168.18.33), Dst: 212.129.33.50 (212.129.33.50)

User Datagram Protocol, Src Port: 19614 (19614), Dst Port: 6881 (6881)

BitTorrent DHT Protocol

- Request arguments: Dictionary...
 - Key: a
 - Value: Dictionary...
 - id: 909f9cbdedf4f7e29e820e3fd5e00a2965450b8a
 - Key: id
 - Value: 909f9cbdedf4f7e29e820e3fd5e00a2965450b8a
 - target: 909f9cbdedf4f7e29e820e3fd5e00a2965450b8a
 - Key: target
 - Value: 909f9cbdedf4f7e29e820e3fd5e00a2965450b8a
- Request type: find_node
 - Key: q
 - Value: find_node
- Transaction ID: 059b
 - Key: t
 - Value: 059b
- Version: 4c540010
 - Key: v
 - Value: 4c540010

find_node: response

No	Time	Source	Destination	Protocol	Length	Info
183	17.706938000	194.177.60.9	192.168.18.33	BT-DHT	308	reply=8 nodes
191	17.818123000	192.168.18.33	194.177.60.9	BT-DHT	143	
198	18.155935000	194.177.60.9	192.168.18.33	BT-DHT	308	reply=8 nodes

Frame 198: 308 bytes on wire (2464 bits), 308 bytes captured (2464 bits) on interface 0
Ethernet II, Src: Netgear_3a:80:86 (c4:04:15:3a:80:86), Dst: Tp-LinkT_f6:73:05 (14:75:90:f6:73:05)
Internet Protocol Version 4, Src: 194.177.60.9 (194.177.60.9), Dst: 192.168.18.33 (192.168.18.33)
User Datagram Protocol, Src Port: 64329 (64329), Dst Port: 19614 (19614)
BitTorrent DHT Protocol
Response values: Dictionary...
Key: r
Value: Dictionary...
id: 90d237c2cdfea7766ced7ea1201538dab3e9e686
Key: id
Value: 90d237c2cdfea7766ced7ea1201538dab3e9e686
nodes: 8
Key: nodes
Value: 8 nodes
Node 1 (id: 03ba8a380a41f4c26b7587e7fcefd3f821d26935, IP/Port: 31.134.138.4:50972)
Node 2 (id: 05e3e9298a5df482bbc1c9a9adb97cae404f7a12, IP/Port: 185.34.3.149:17328)
Node 3 (id: 099d9e8813e9ae7a7ad6a193b0bfff6c6a6230b, IP/Port: 185.34.3.201:19202)
Node 4 (id: 1f6dc350c9ed479970b5f7b5df65364581c580c6, IP/Port: 84.215.181.4:45433)
Node 5 (id: 27668269700c379de1b40b806a9abe0da830c97b, IP/Port: 185.34.3.160:21369)
Node 6 (id: 3584219cead1d7ff8bcd8843ffff0df3c5b4eb4f7, IP/Port: 87.231.175.126:32563)
Node 7 (id: 4ae3902d20c3e428726baf1c261d1bb67e432ea6, IP/Port: 185.34.3.189:11339)
Node 8 (id: 58e65cd7e7e8adb8f211823853754467e6a09c2, IP/Port: 2.34.4.116:6881)

当前节点的节点ID
被请求节点的路由表中最接近目标节点的8个节点的ID和联系信息

Transaction ID: 2f6b
Key: t
Value: 2f6b

get_peers: request

No	Time	Source	Destination	Protocol	Length	Info
506	26.397368000	192.168.18.33	178.61.21.187	BT-DHT	146	
522	26.755547000	178.61.21.187	192.168.18.33	BT-DHT	1169	reply=8 nodes reply=100 peers

Frame 506: 146 bytes on wire (1168 bits), 146 bytes captured (1168 bits) on interface 0
Ethernet II, Src: Tp-LinkT_f6:73:05 (14:75:90:f6:73:05), Dst: Netgear_3a:80:86 (c4:04:15:3a:80:86)
Internet Protocol Version 4, Src: 192.168.18.33 (192.168.18.33), Dst: 178.61.21.187 (178.61.21.187)
User Datagram Protocol, Src Port: 19614 (19614), Dst Port: 57060 (57060)

BitTorrent DHT Protocol

Request arguments: Dictionary...

Key: a

Value: Dictionary...

id: 0231607e44aeee9aa750e49e6045094f351869efd

Key: id

Value: 0231607e44aeee9aa750e49e6045094f351869efd

info_hash: 1619ecc9373c3639f4ee3e261638f29b33a6cbd6

Key: info_hash

Value: 1619ecc9373c3639f4ee3e261638f29b33a6cbd6

Request type: get_peers

Key: q

Value: get_peers

Transaction ID: e7b0

Key: t

Value: e7b0

Version: 4c540010

Key: v

Value: 4c540010

请求方的节点ID

Torrent文件的hash值

get_peers: response

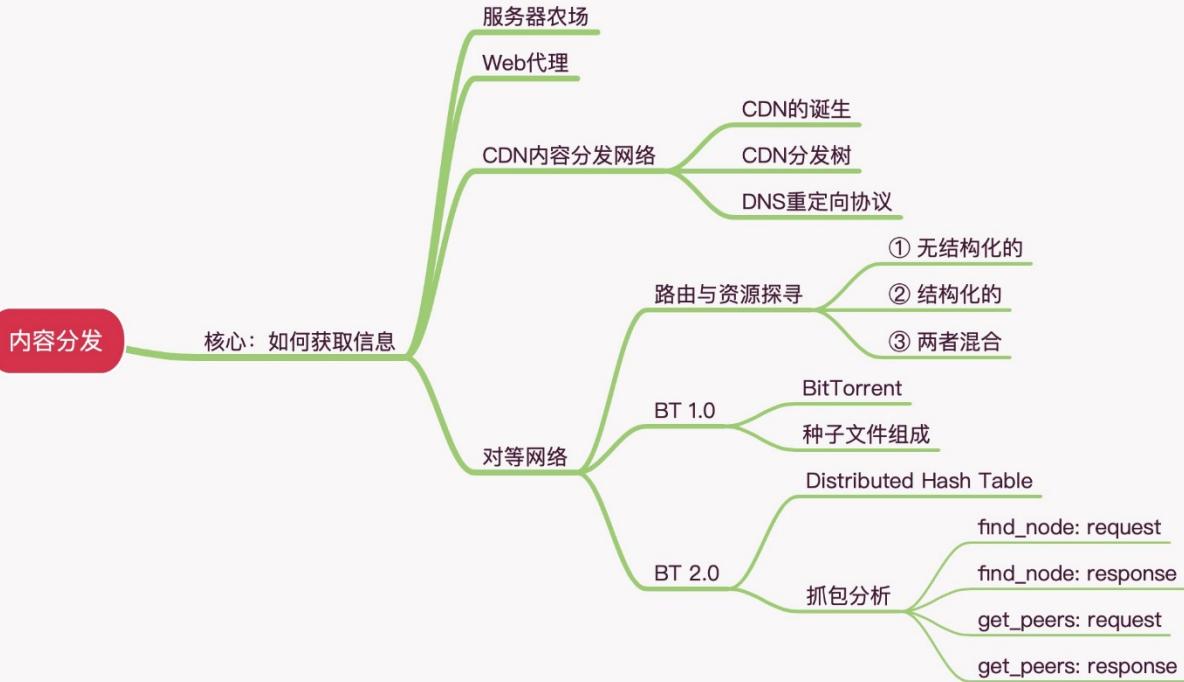
No	Time	Source	Destination	Protocol	Length	Info
506	26.397368000	192.168.18.33	178.61.21.187	BT-DHT	146	
522	26.755547000	178.61.21.187	192.168.18.33	BT-DHT	1169	reply=8 nodes reply=100 peers

当前节点的ID
被请求节点路由表中离info_hash最近的8个节点

```
[{"@version": 1, "id": "75901a324c9e", "ip": "192.168.18.33", "port": 26755547, "info_hash": "1619e979fde7b9c6824fd8a384c63c5aed20b932", "nodes": [{"id": "1619ec780358fbda0938ad0ef26165ba9c81a9f2", "port": 118.173.21.187:6881}, {"id": "1619ec1fd1318984d6ba4b99b901f5ec75610a3a", "port": 218.173.149.187:1463}, {"id": "1619edda45f1d83896c23d6e31912ab34aba5325", "port": 82.132.231.113:57942}, {"id": "1619ed3f0a3e53b3a9b43b61aad2c26bf8e3d81d", "port": 196.210.160.104:10193}, {"id": "1619ee9c9c5ae61a61252490a595a6f90394885d", "port": 82.132.230.244:49604}, {"id": "1619ee92db28da58b9b66445fe25fa151ea6e6bd", "port": 82.132.230.244:50307}, {"id": "1619ee1b618064e707a04f382ad1da3d3a48f46d", "port": 79.21.3.204:17377}, {"id": "1619ef47bc9f67274ca475e2ae04017b6b76f405", "port": 60.242.32.104:51324}], "token": "3c3cc068078ac97d23999c35c18909af2f651fe6"}]
```

总结

Summary



Thanks to Internet and Chen Tsuei