Topics

- Popular data
- Creating broadcasting storage system
- Bittorrent protocol
- Creating swarms and destroying swarms
- Tracker-less torrent
- Dealing with uncooperative peers
Popular Data

- Trending images and Videos
  - Active for small period of time (hours)

- Newly released TV shows and movies
  - Active for days or even weeks
Accessing Popular Data
Accessing Popular Data

- The data between downloaders and storage node is redundant.
- The throughput decreases as the number of the downloaders increases.
- The data can be far from the downloader which affects the throughput.
Content Distribution Network
Content Distribution Network

- **Pros**
  - Data is usually closer to the downloaders
  - Server load is balanced

- **Cons**
  - Usually expensive because of the infrastructure
  - There are several persistent copies of the data
Creating broadcast storage system

- Scaling performance as the number of downloaders increase
- Keeping the load on storage system bounded
- Making sure that the storage system delivers unique packets
- Achieving this without spending a fortune
Bittorent protocol

- .torrent file: File and tracker information
- Tracker: Help downloaders find each other
- Swarm: Downloaders coordinate with each other to download file in pieces
Storage System and Bittorrent Protocol
Default Seeder

- This is a gateway that can directly communicate with backend storage system
- The default seeder caches the entire file
  - If the file is popular and there are not enough peers
  - If the file is popular and there are free riders
Storage System and Bittorrent Protocol

Default Seeder

[Diagram showing the connection between a default seeder and various devices (phone, laptop, cloud)]
Storage System and Bittorrent Protocol

Default Seeder
Storage System and Bittorrent Protocol

Default Seeder

[Diagram showing various devices connected through arrows, indicating data flow]
When To Create a Swarm

- Implementing Amazon S3 API
  - GET /quotes/Nelson?torrent
- Selectively return .torrect file with existing interface for your storage system
  - for the objects that are frequently used
  - if storage system is under load do not provide 503
- if bittorent overhead is less than file size
Trackerless implementation

- The tracker is SPoF in most P2P networks
- Using the Mainline DHT
  - Distributed method to find peers
- Instead of using ‘announce’ key use ‘nodes’
- The Default Seeder acts as bootstrapping node
Free Riding Peers

- Conscious Free Riders
  - Inform their partners that they are unwilling or unable to upload data
  - Devices with limited CPU and bandwidth

- Oblivious Free Riders
  - Do not inform their partners that they are unwilling or unable to upload data
  - Malfunctioning and uncooperative clients
Latencies Distribution For File Pieces

- Conscious Free Riders
- Oblivious Free Riders

From “Can Peer-to-Peer Live Streaming Systems Coexist with Free Riders?”
Joaõo F. A. e Oliveira*, Ítalo Cunha* et al
Dealing with Uncooperative Peers

- Chocking/Unchoking is not sufficient
- Update .torrent to exclude uncooperative peers
  - This prevents new peers from talking to uncooperative peers
- Different billing rates
  - Bill normal for direct access to the object
  - Discount for swarm with no free riders
  - Free access for swarm with free riders
Destroying The Swarm

- Default seeder keeps track of active peers
  - If there are no active peers the file can be cleared from cache on Default seeder
  - Nullify the ‘nodes’ in .torrent file so there is no swarm
Conclusion

- A swarm is created as the popularity of the data increases
- A swarm can help scale the throughput of the storage system
- Uncooperative peers are challenging but the impact can be curbed by different billing rates
- The swarm eventually disappears as the data is no longer popular
Questions & Answers

Yogesh Vedpathak
yvedpathak@cleversafe.com