

# apt-p2p: A Peer-to-Peer Distribution System for Software Package Releases and Updates

*authors:*

Cameron Dale, Jiangchuan Liu

*presented by:*

Cameron Dale

Simon Fraser University

Burnaby, BC, Canada

[camerond@cs.sfu.ca](mailto:camerond@cs.sfu.ca)

*INFOCOM, April 22<sup>nd</sup>, 2009, Rio de Janeiro, Brazil*

# Overview

- Opportunity that is available
- Difficulties encountered
- Our solution
- The apt-p2p implementation
- Discussion

# Free Software Package Distribution

- Traditionally uses the client-server model
- Users are altruistic, willing to contribute
- No current mechanism for them to do so



# Similarities

- Software packages are all freely available in an archive on one or more servers
- The server(s) is always available
- Cryptographic hashes of the packages' files are available separately from the package
- Altruistic peers exist willing to aid in the distribution of these packages

# Debian Linux Distribution

- Packages are downloaded and installed by the `apt` tool
- Archive is served by a network of 300 mirrors mostly donated to the project
- Archive servers are simple HTTP servers
- Information on packages, including hashes, are contained in index files
- Security is maintained using GPG signing of index files

# Archive Dimensions

- Most of the packages are very small, but some are very large
  - i.e. 80% < 512KB, some > 100 MB
- There are too many packages to distribute them individually
  - i.e. 22,298 packages
- The whole archive is too large to distribute in its entirety
  - i.e. 116 GB

# Package Updates

- The packages are constantly updated
- Developers, working independently around the world, do not synchronize updates
- Even “stable” releases are updated with security fixes or serious bug fixes
- i.e. 1.5% of Debian's package archive is updated every single day

# Limited Interest

- The users interest in a specific package is mostly very limited
- Most packages are optional or extra, and only of interest to a small number of users
- This is exacerbated by there being multiple versions of each package available
- i.e. 80% of Debian's packages are installed by less than 1% of its users



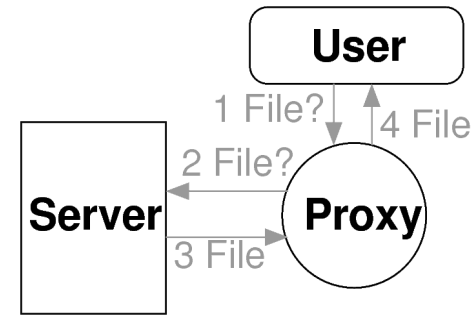
# Why not BitTorrent?

- Suitable for CD images, not entire updated software archives
- Fractures the downloading population
  - How to divide the archive into torrents?
  - Daily updates create new torrents constantly
- Fixed piece sizes will cause unwanted parts of other packages to be downloaded

# Peer-to-Peer Assisted Solution

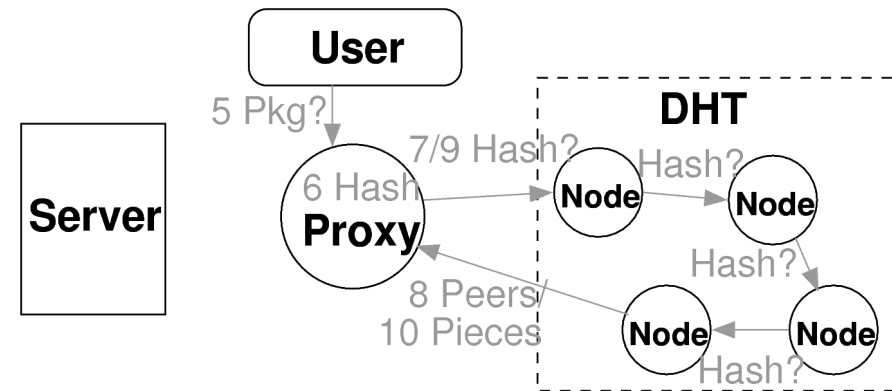
- Transparent proxy between the user and server
- Requests for packages are first searched for in an indexing structure (DHT)
- Uses the existing server as a seed for the packages in the P2P system
  - Packages not found by P2P means can be downloaded from the server
- The user then becomes a source of the package in the P2P system

# Proxy



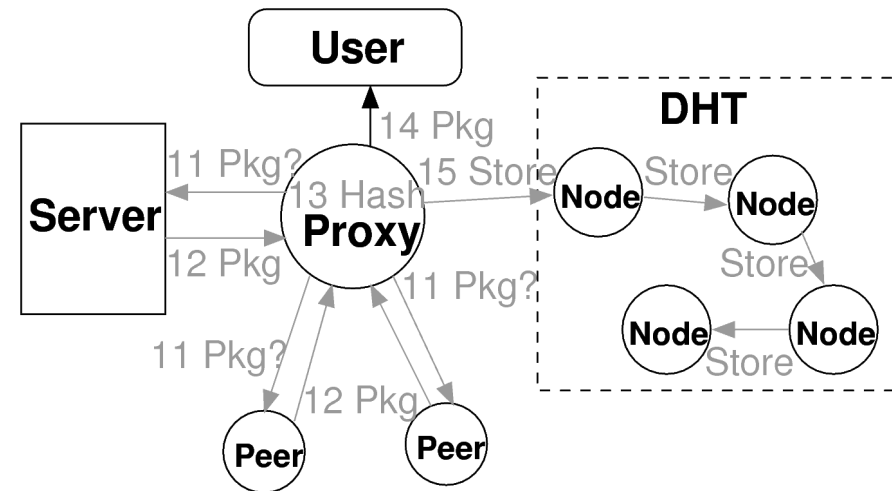
- All requests to the server from the user (1) are proxied by the p2p program (2)
- Files (3) that contain hashes are cached for future lookups
- All files are returned to the user (4)

# DHT Request



- Requests for packages (5) trigger a lookup of the package's hash (6)
- The hash will be looked up in the P2P indexing structure (DHT) (7)
  - Peers that have the package are returned (8)
- Additionally, hashes of pieces (chunks) of the package are looked up in the DHT (9)
  - Hashes of pieces of the package are returned (10)

# Package Download



- The package is downloaded from peers (11/12)
  - If peers can not be found, the package is downloaded from the server (11/12)
- The package is hash checked (13) and returned to the user (14)
- The user's location is stored in the DHT as a new source for the package (15)

# Notes

- Trust of a package is always guaranteed, whether downloaded from peers or a server
- To maximize downloading bandwidth, large packages are split into pieces (with hashes)
- Sparse interest in a constantly updated set of files (hashes) is perfect for a DHT
- Keep it simple: e.g. use the same protocol for peers as the server uses

# Debian Implementation: `apt-p2p`

- Caching HTTP proxy
- Peers are also HTTP servers to other peers
- HTTP/1.1 is used for downloading of pieces and pipelining of multiple requests
- Indexing system is a DHT based on Kademlia
- Hashes of pieces of packages are also stored in the DHT

# Piece Hashes

- 512KB size pieces
- 78% of packages are a single piece
- 2-4 piece hashes are stored with the list of peers in the DHT (14% of packages)
- 5-69 pieces are stored separately in the DHT (7% of packages)
- 70+ pieces are served by HTTP requests to the peer (less than 1% of packages)



# Implementation Availability

- Source code is available under the terms of the GNU General Public License (GPL)
  - Other distributors can modify it to suit their system
- Available in the main archives of Debian and Ubuntu for easy installation by all users
- Featured recently in a Slashdot article
  - <http://slashdot.org/article.pl?sid=09/04/12/0232244>

# Peer Lifetimes

- Statistics gathered by “walking” the DHT
- 186 users in 2 months
- Peers stay online longer than in other similar P2P systems
  - Due to being installed as a system daemon
- 50% of connections last longer than 5 hours
- 80% of peers will remain online another hour
  - Surpasses Kademlia design principle of 50%

# Peer Downloads

- Users are not required to share download information
- 36 unique users in 16 days
- Peers downloaded 15 GB from other peers
- 20% of their downloads are from peers
- Saving the mirror network 1GB per day

# Issues

- 50% of peers are firewalled or unable to be contacted
- Package downloads are delayed by slow DHT lookups
  - Timeouts waiting for firewalled peers are a problem
  - Lots of work has been done, still more needed

# Future Work

- Faster DHT response time
  - Use of STUN to make more peers contactable
  - Performance tuning of DHT parameters
- Other improvements
  - Stale cache removal
  - Package delta storage

# apt-p2p: A Peer-to-Peer Distribution System for Software Package Releases and Updates

*authors:*

Cameron Dale, Jiangchuan Liu

*presented by:*

Cameron Dale

Simon Fraser University

Burnaby, BC, Canada

[camerond@cs.sfu.ca](mailto:camerond@cs.sfu.ca)

*INFOCOM, April 22<sup>nd</sup>, 2009, Rio de Janeiro, Brazil*