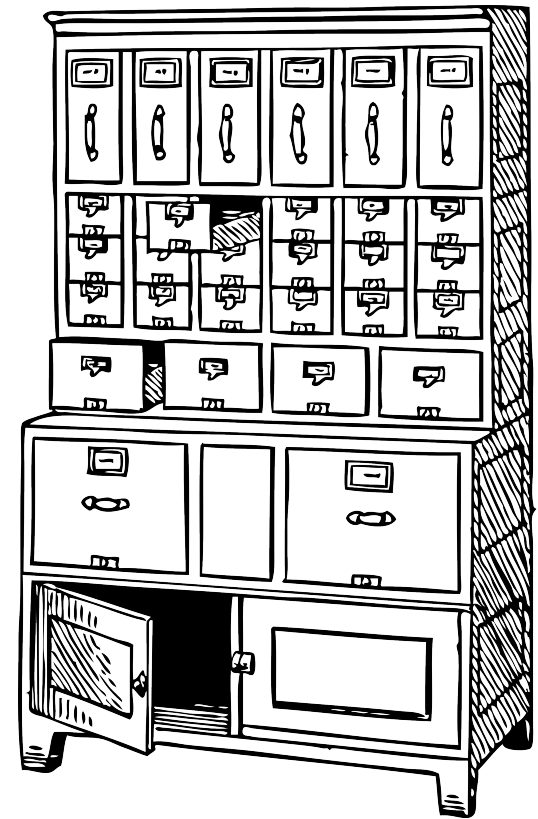


Prequel



WAN Acceleration and Distributed Caching

José A. Rivera
Christopher R. Hertel



Introductio



Introductions

José Angel Rivera
Roa Pérez Amezaga



Swimming in the deep
end of the SMB cesspool
since 2008.

Christopher R. Hertel



Your friendly
neighborhood SMB
geek.

Introductions









redhat®

The opinions expressed are our own
and not necessarily those of our employer,
our respective spouses, pets, or “the Voices”.

Introductions



Where are we going?
*...and what am I doing in this
handbasket?*

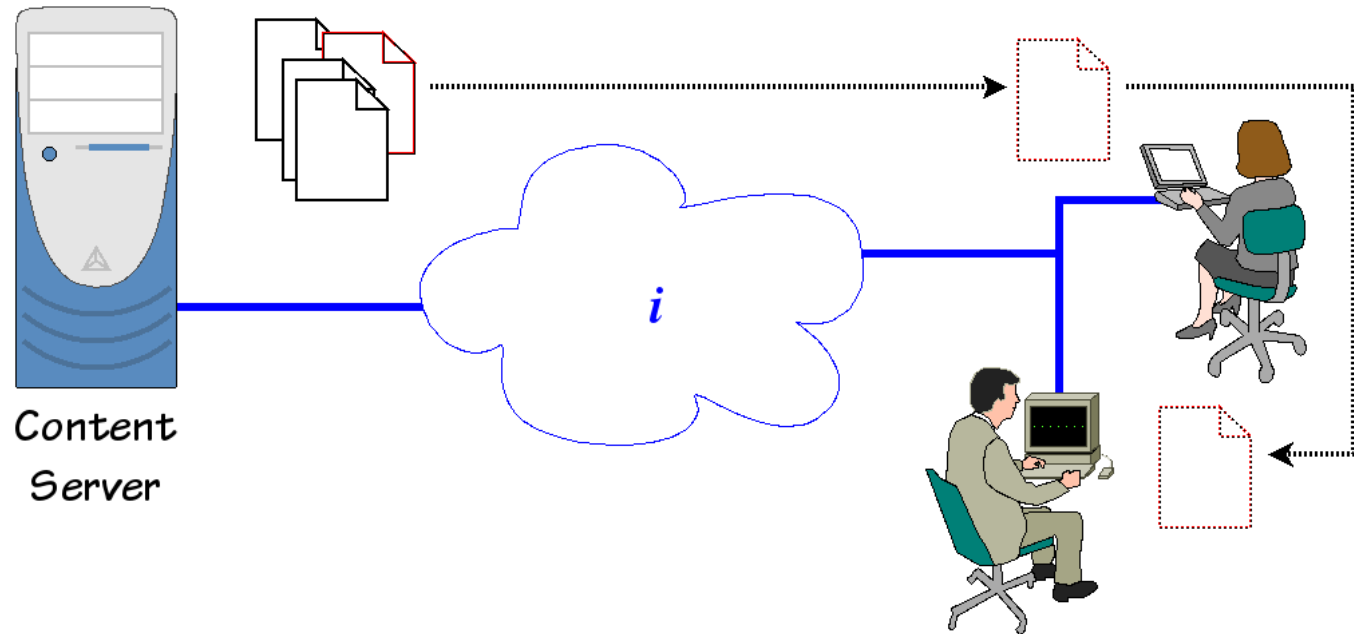
-  BranchCache™ Overview
-  The Prequel Project
 -  PrequelD
 -  PrequelHC
 -  Client Plans
 -  Tools



BranchCache The overview



BranchCache Overview

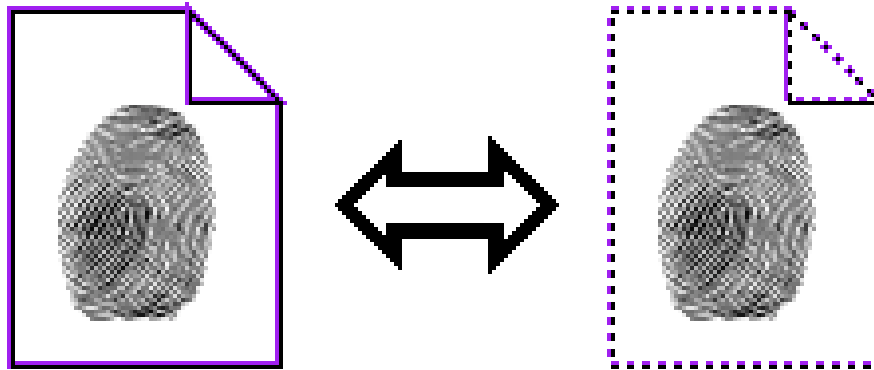


Accessing content over a WAN link

- Minimize content copies over the WAN
 - Cache the copy on the local network
- Ensure that the cached copy is still valid
 - Retrieve fingerprints from the server



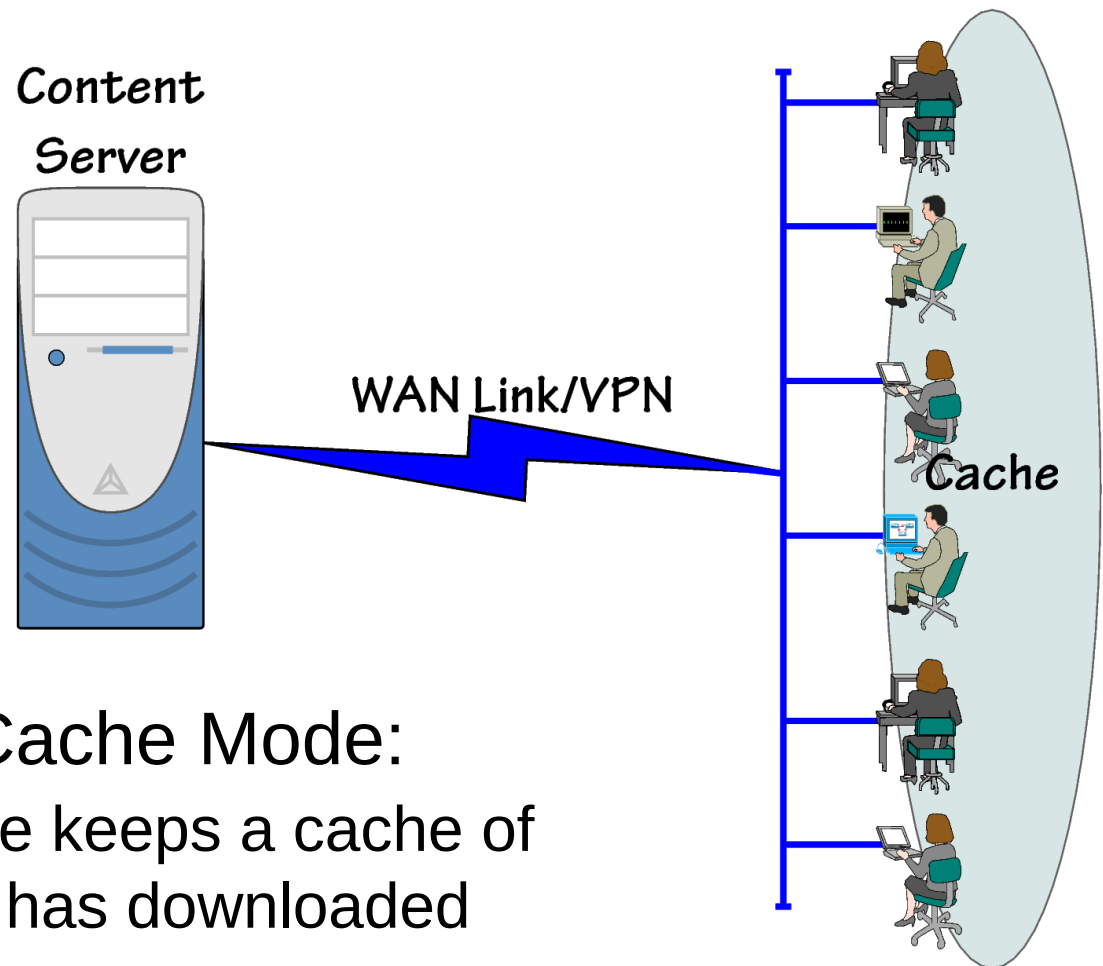
BranchCache Overview



Clients request “fingerprints”

- Each fingerprint maps to a chunk of content
- Fingerprints are used to find cached content
- If content is not found in the local cache, it is retrieved over the WAN
- Cache keeps fingerprint-to-content mapping

BranchCache Overview

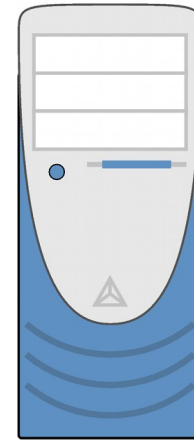


Distributed Cache Mode:

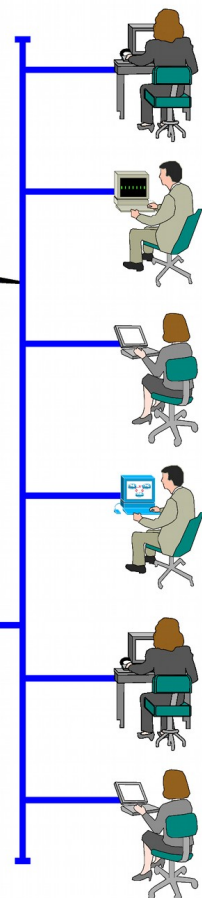
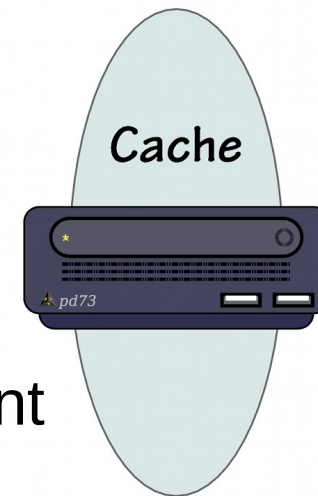
- Each node keeps a cache of content it has downloaded
- Clients broadcast to find content
- The cache is distributed across peers
- Limited to the local LAN

BranchCache Overview

Content
Server



WAN Link/VPN



Hosted Cache Mode:

- Clients tell cache node that they have cache-able content
- The cache node retrieves cache-able content from the client node
- Clients always query the cache node for content
- Not LAN-locked




BranchCache Overview

Any questions about
BranchCache basics?



Prequel Project





Prequel Project

The Prequel Project

- An Open Source PeerDist implementation
- PeerDist is the protocol suite underlying BranchCache





Prequel Project

Prequel Project Goals

- ♥ PrequelD: Server-side hash generation
Interface with:
 - Samba
 - HTTP server (e.g. Apache)
- ♥ PrequelHC: Hosted Caching
- ♥ Prequel Client for Linux
- ♥ Prequel Tools





Prequel Project

Websites:

 <http://fedorahosted.org/prequel/>

Source code repository

 <http://ubiqx.org/proj/Prequel/>
Project home page

Microsoft Docs:

[MS-CCROD] Content Caching and Retrieval Protocols Overview

[MS-PCCRC] Peer Content Caching and Retrieval: Content Identification



Prequeld: the Prequel Server Daemon



PrequelD






- PrequelD is a Userland Dæmon
 - Make “nice”
 - Background hash generation
 - Hashes stored in cache files
- Cache files are “shared read”
- Speak to Dæmon over a socket
 - Threaded communication



PrequelD

Currently “works”

-  Needs signal handling
 - SIGHUP: Reload Config
 - SIGTERM: Clean shutdown
-  Should traverse directories in the background (feature)
-  Should do stale cache cleanup




Server

-  Multi-threaded daemon
-  Builds hash files from content



Client

-  Maps content files to hashes
-  Reads “Content Information”





PrequelD

PrequelD Server Status

- ✖ The code is currently in pieces
 - Client-server communication has been gutted and redesigned
- ✖ Core functions, however, have been working for months
 - Generation of PeerDist v1 Content Information works well



PrequelD

ToDo List

- 🦖 Finish implementation of C client
- 🦖 Write a simple Python client
 - Integrate Prequel into Apache
- 🦖 Complete missing features
 - Exclusion patterns
 - Cache directory cleanup

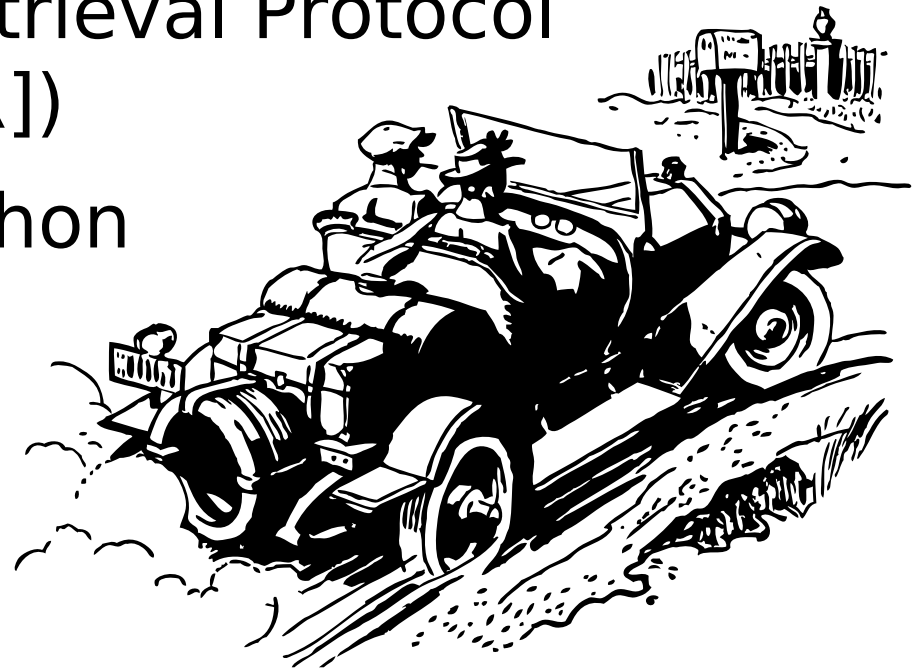


PrequelHC: the Prequel Hosted Cache Server



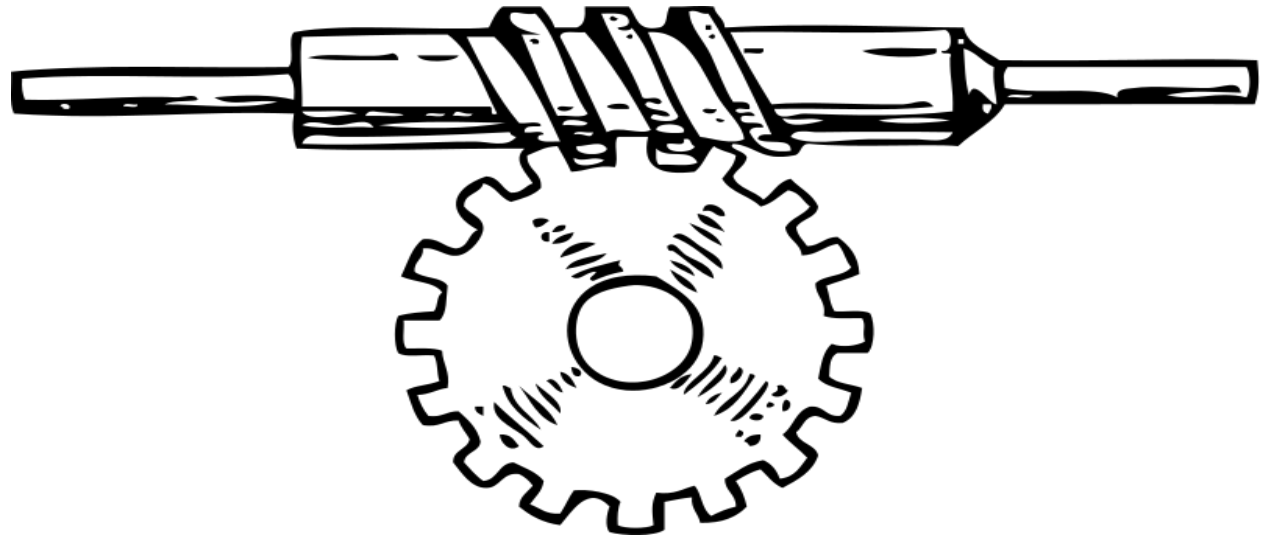
PrequelHC

- ✈ Stand-alone HTTP(S) server
- ✈ Implements two sub-protocols:
 - ✈ PeerDist Hosted Cache Protocol ([MS-PCHC])
 - ✈ PeerDist Retrieval Protocol ([MS-PCCR])
- ✈ Written in Python









PrequelHC

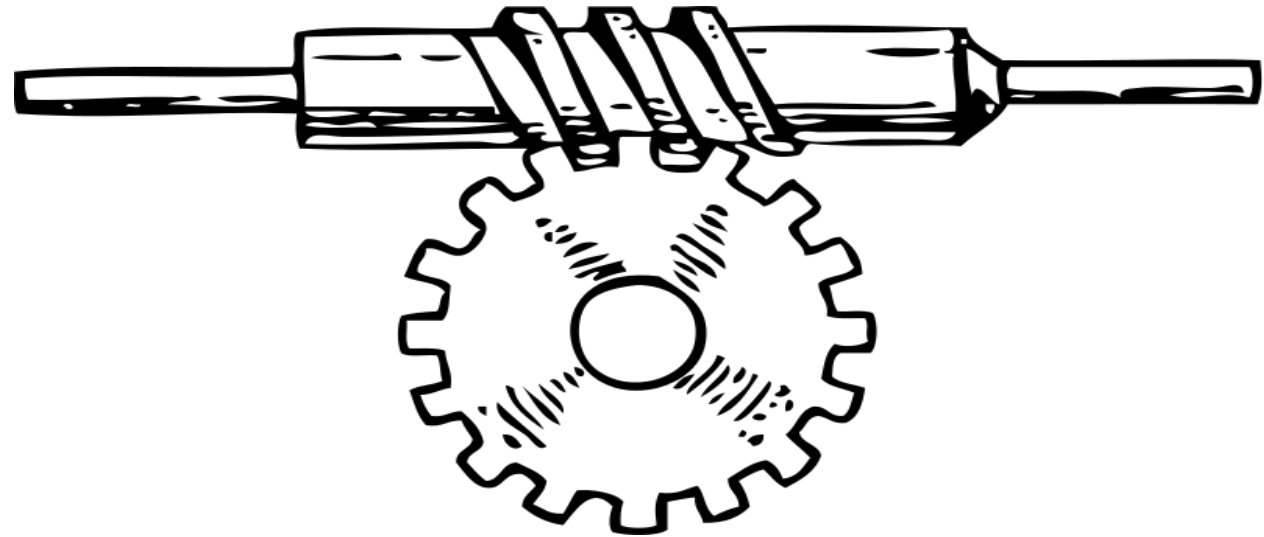


Hosted Cache Protocol




-  Used by clients to offer content to the hosted cache server
-  Used by servers to fetch content information from clients
-  PeerDistv1 requires HTTPS
-  PeerDistv2 requires HTTP



PrequelHC



Retrieval Protocol

-  Used by Hosted Cache server to fetch offered content from clients
-  Transmitted over HTTP
-  Data blocks are encrypted on the wire



PrequelHC

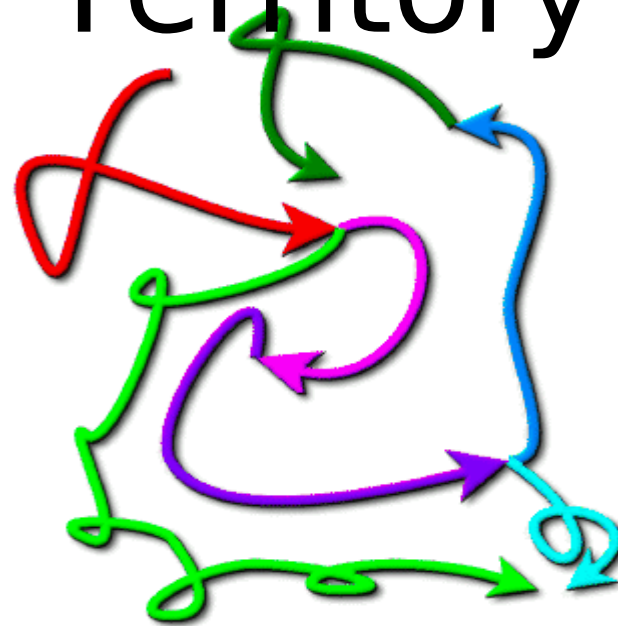
The Future

- 🚢 C libraries for sub-protocols
- 🚢 Apache module? CGI script?
- 🚢 Maintain stand-alone server?





Prequel Client: the Uncharted Territory





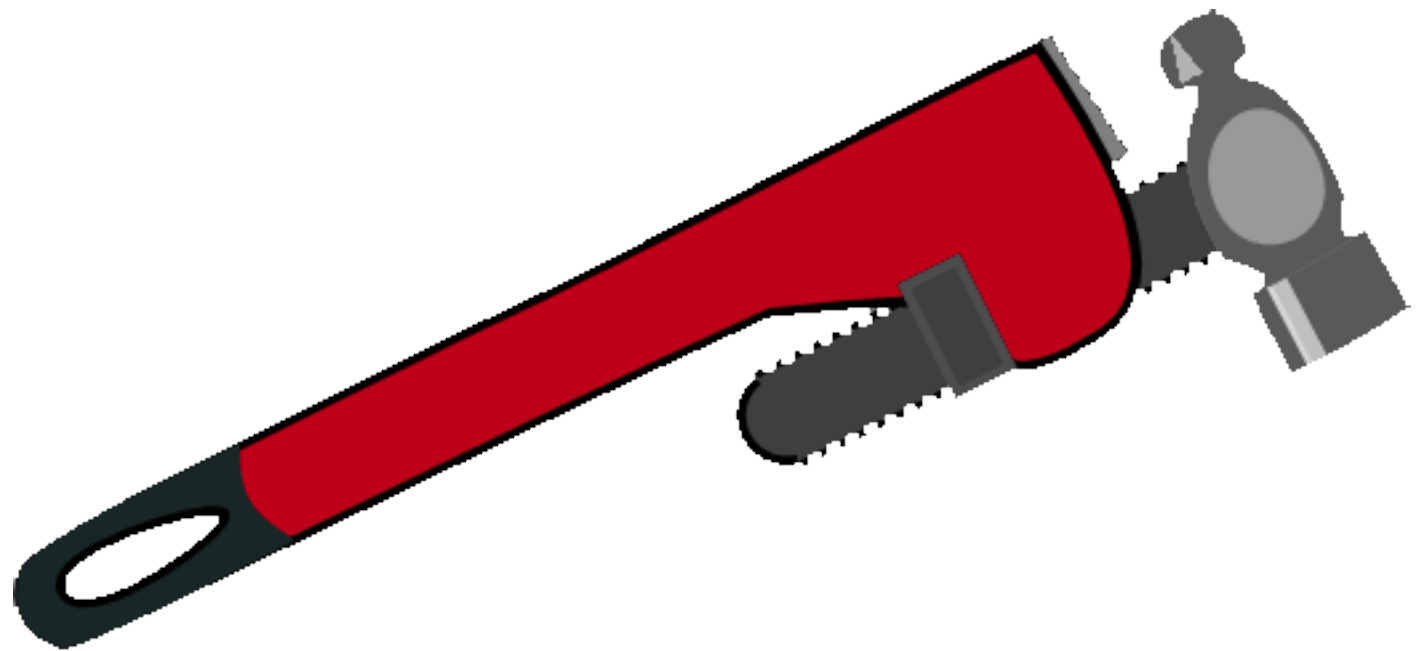
Prequel Client

- ✈ A user-land client would be fairly easy
 - Simple user management
 - Applications would need to call it directly
- ✈ An in-kernel client is more daunting
 - Could integrate with the “CIFS” client





Prequel Tools: Catch as Catch Can






Prequel Tools

Tools we've slapped together as we build and test our implementation.

 **PdDump**

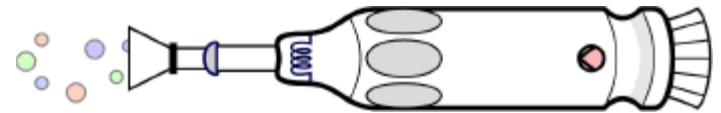
PeerDist v1 Content Information Dump

 **pq_size_calc**

Calculate the Content Information size from the original file size

 **oSSL_key_dx**

Decrypt a BranchCache key extracted from Windows





Prequel Tools

Tools we've slapped together as we build and test our implementation.

 STiB

Retrieve Content Information over HTTP
(also implements BITS protocol)

 pq_cgi

CGI program generates Content Information
on the fly





The End



Prequel Project

Websites:

 <http://fedorahosted.org/prequel/>

Source code repository

 <http://ubiqx.org/proj/Prequel/>
Project home page

Microsoft Docs:

[MS-CCROD] Content Caching and Retrieval Protocols Overview

[MS-PCCRC] Peer Content Caching and Retrieval: Content Identification