



# *Distributed Caching:*

*Gaining Speed by Reduplicating Data*




*Christopher R. Hertel*

*Samba Geek*

*Senior Principal Software Engineer, Red Hat*

*Novemberly, 2012*



# Introductions



# Introductions

## Me: Your Friendly Neighborhood CIFS Geek

- ▶ Samba Team member (since 1998-ish)
- ▶ jCIFS Project co-founder
- ▶ CIFS Author (shameless plug)
- ▶ Network Storage Geek
- ▶ Incurable Idealist
- ▶ Etc., etc., ad nauseum



A ruminant mammal (Geekus geekus) with long legs, humped shoulders, and broadly palmated antlers.

# Introductions

---



redhat®

The opinions expressed are my own  
and not necessarily those of my employer, my spouse,  
kids, pets, so-called friends, or "the Voices".



# Introductions

---



Members of the Samba Team gather at the 10<sup>th</sup> annual Samba eXPerience conference in Göttingen, Germany.

# Introductions



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

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

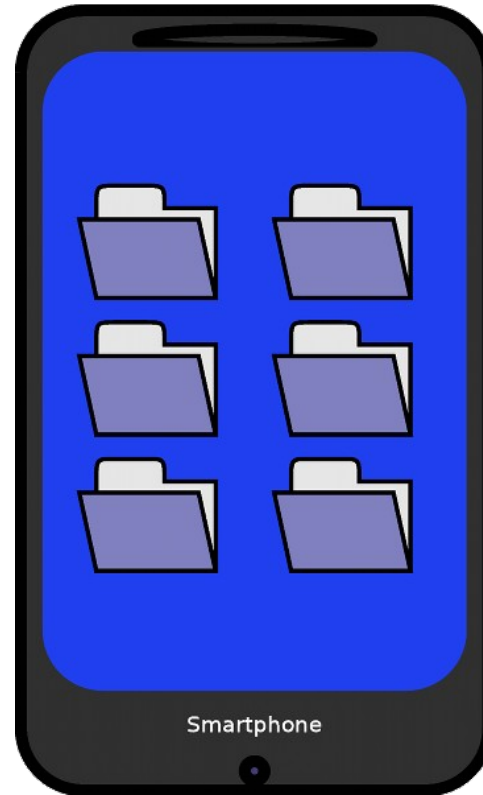


# Getting Sidetracked



# Sidetracks

---





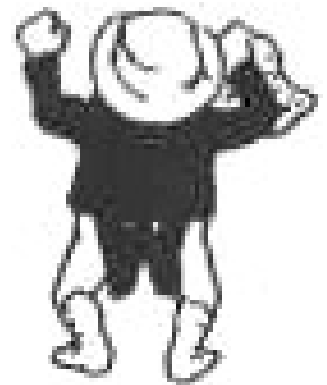
# Sidetracks

---



## Cellphones/Tablets as Cloud Terminals

- ★ The Cloud stores Mass Quantities
- ☯ ...but Bandwidth is low
- ☯ ...but you only need a little at a time



# Sidetracks

---



## Common Data Transfer Methods:

- ★ Local (USB, Bluetooth)
- ★ Web (HTTP/HTTPS)
- ★ Proprietary (Dropbox, etc.)
- ★ Windows (SMB/CIFS via jCIFS)





# Sidetracks

---

## ICIFS

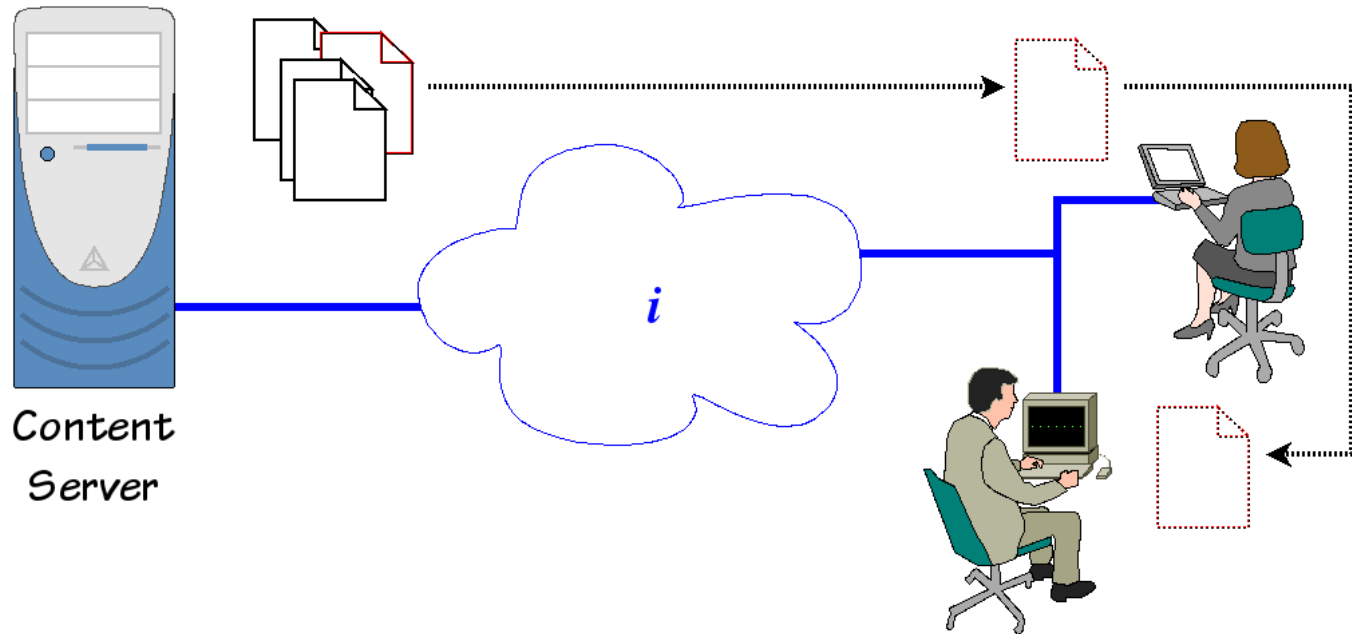
- A little project I started many years ago
- A Java client for SMB/CIFS



# BranchCache Overview



# BranchCache Overview

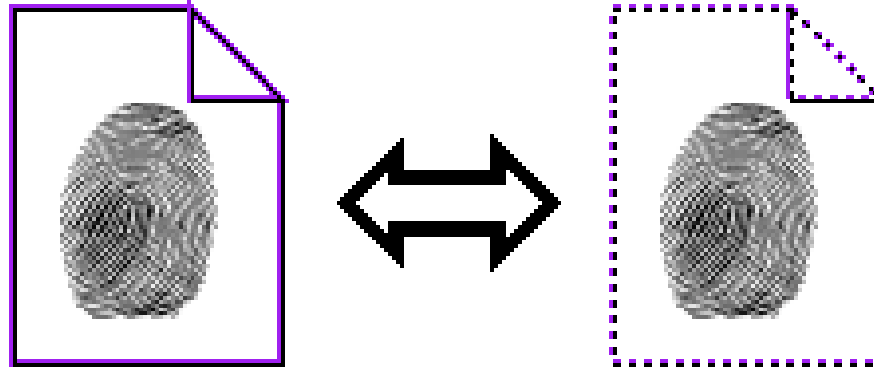


## Accessing content over a WAN link

- 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

---

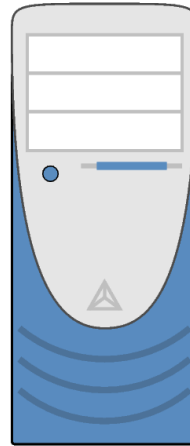


## Clients request “fingerprints”

- 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

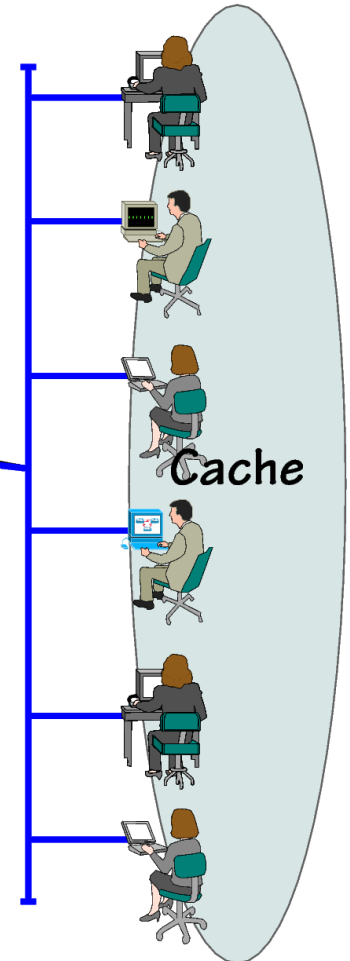
Content Server



WAN Link/VPN



Cache

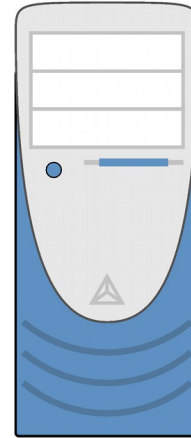


## Distributed Cache Mode:

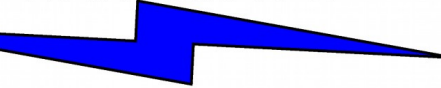
- Each node keeps a cache of content it has downloaded
- Clients broadcast to find content
- The cache is distributed across peers
- Limited to the local LAN

# BranchCache Overview

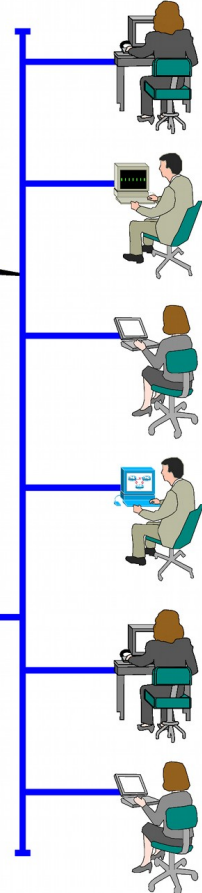
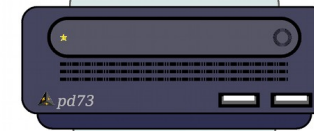
Content  
Server



WAN Link/VPN



Cache



## Hosted Cache Mode:

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

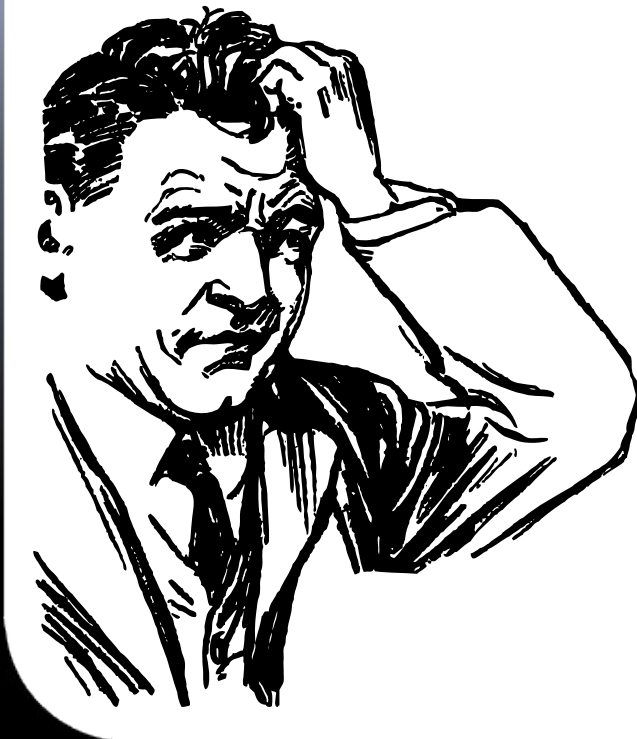




# BranchCache Overview

---

Any questions about  
BranchCache basics?



# Prequel Project





# Prequel Project

---

## The Prequel Project

- An Open Source implementation of the PeerDist protocol
- 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 Cache
- ♥ 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

[MS-PCHC] Peer Content Caching and Retrieval:  
Hosted Cache Protocol Specification



# 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**

## **Cache File Access**

### **API defined**

### **Code should be done soon**

## **Supports only PeerDist v1**

### **Design allows for PeerDist v2**







# PrequelD

## Configuration File



```
socket /var/run/prequeld.sock;
logfile /var/log/prequel.log;

cachedir /var/prequel_cache
{
  hash1 sha256;      # Most common hash type
  hash2 none;
  keyfile /etc/prequel/prequeld.key;
  sourcedir /data/music
  {
    keyfile /etc/prequel/music.key;
    minblocks 4;
    verbosity 0;
    exclude *.tmp, Queen;
  }
}
```

# PrequelID

## Configuration File

- **Scoped configuration**
  - **Global section**
    - **Most settings have reasonable defaults**
  - **cachedir sections**
    - **Identify the target directory**
    - **Contained settings are section local**
    - **Settings apply to sourcedirs that follow**
    - **Order is important!**
  - **sourcedir lines or sections**
    - **Identify directories of source files**
- **Comments start with a '#'**





# PrequelID

---

## Design Challenges



**The hash cache is not directly connected to the source content**

- **Cache files can get out of sync**

- **May need to be re-hashed**

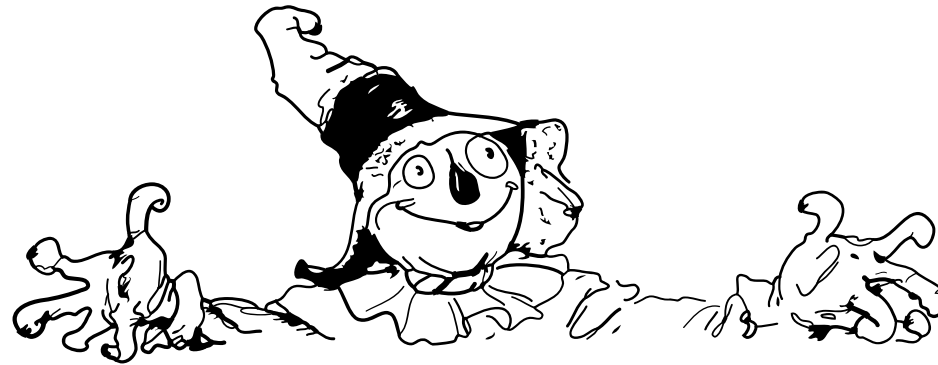


**Hashes should be removed on source file write or delete**

- **Cache files can become “orphans”**



**Would it be better to keep the cache within the file system?**



# PrequelHC: the Prequel Hosted Cache Server



# PrequelHC: the Prequel Hosted Cache Server

( Coming Soon )

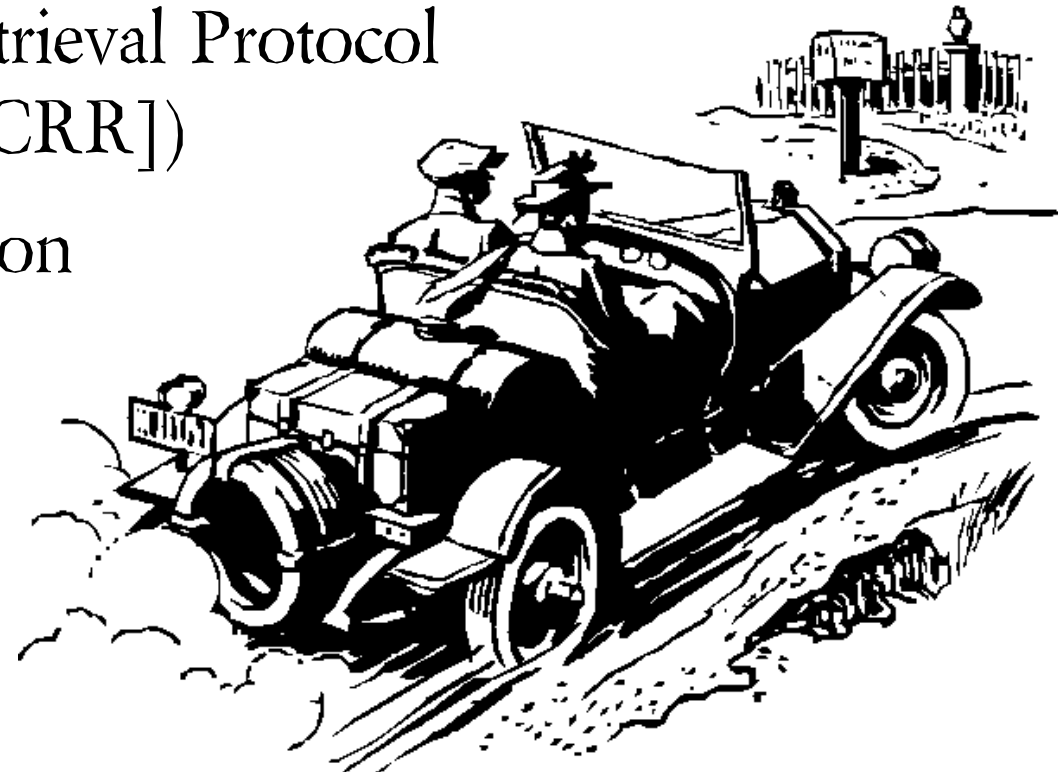




# PrequelHC

---

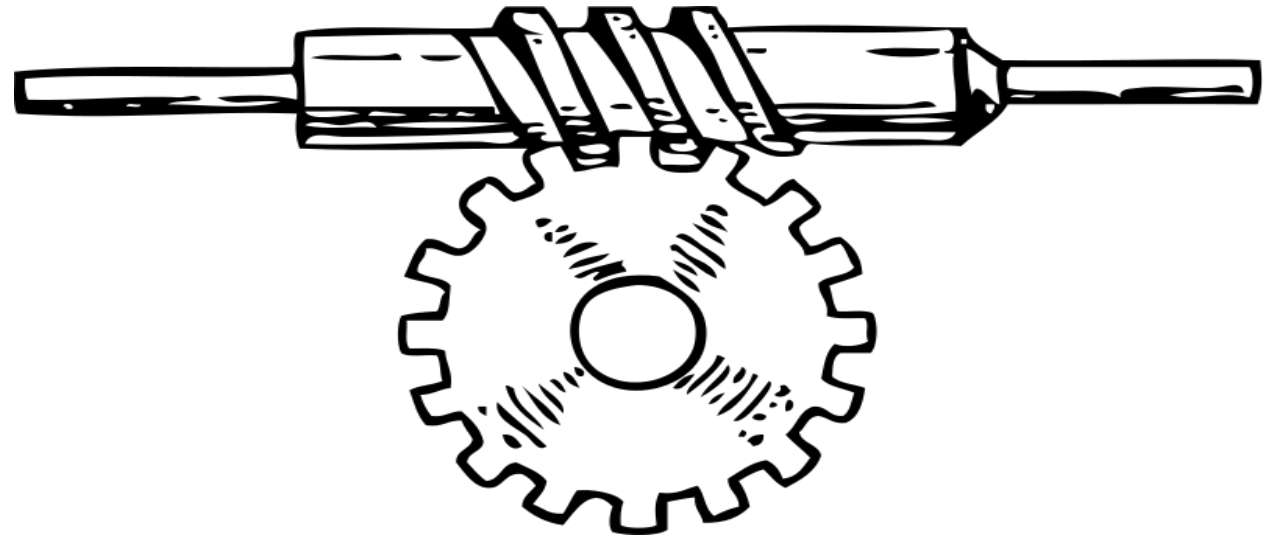
- ✿ Stand-alone HTTP server
- ✿ Implements two sub-protocols:
  - ✿ PeerDist Hosted Cache Protocol ([MS-PCHC])
  - ✿ PeerDist Retrieval Protocol ([MS-PCCR])
- ✿ Written in Python









# PrequelHC

---



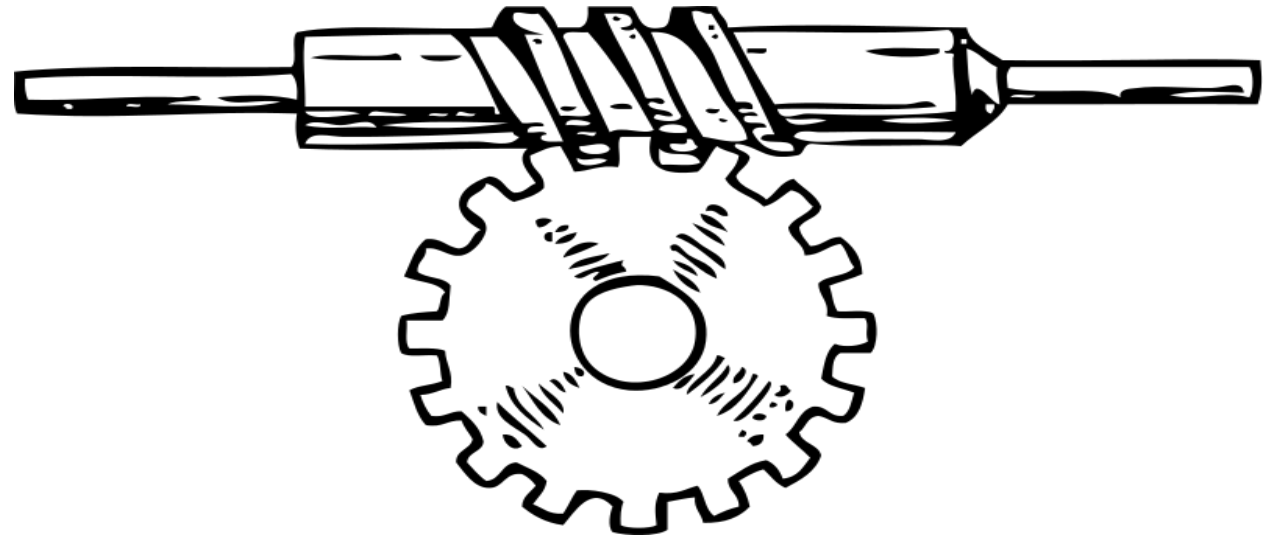
## Hosted Cache Protocol

-  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 over 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

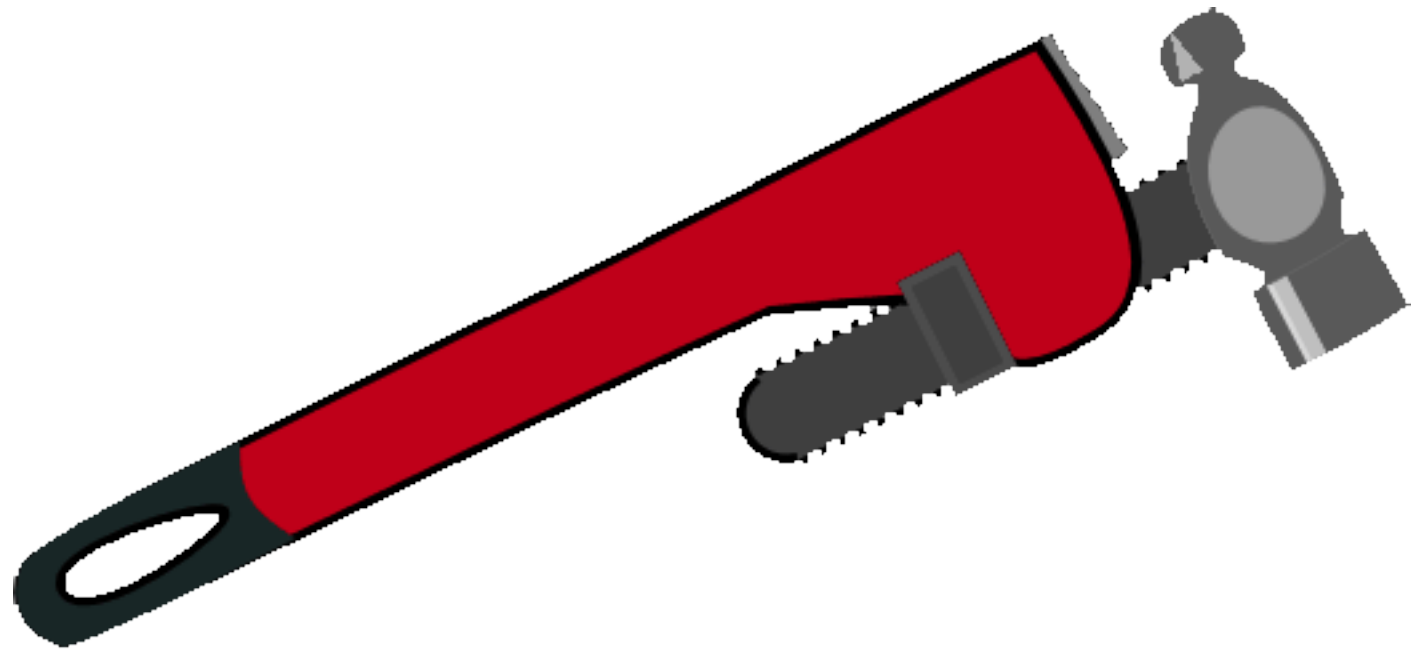
---

- ✈ **Δ user-land client would be fairly easy**
  - **Simple user management**
  - **Applications would need to call it directly**
- ✈ **Δn in-kernel client is more daunting**
  - **Could integrate with the "CIFS" client**
  - **Could mesh with the file system cache**
  - **Δvailable to sync with de-duplication**





# Prequel Tools: Catch as Catch Can





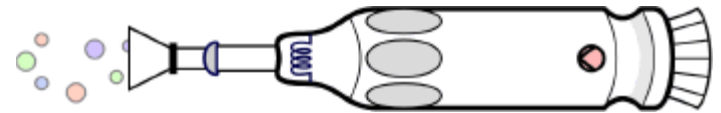
# Prequel Tools

---

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

[MS-PCHC] Peer Content Caching and Retrieval:  
Hosted Cache Protocol Specification



Questions?

**The Tacky  
Tie Show**