rproxy: dynamic web caching

Martin Pool
Linuxcare, Inc.

mbp@linuxcare.com.au
http://linuxcare.com.au/rproxy/

Problem Statement

People use web resources repeatedly

Therefore: cache recently-used resources on client or proxy

On each request, check currency: either reload or use same

Increasingly, content is dynamic: all-or-nothing caches are less effective

WIBNI

It would be nice if we could transfer only differences

Must interoperate smoothly with HTTP

Must work on dynamic documents

Must fit into popular HTTP software

rsync

Fast file transfer protocol

Finds identical blocks between two files, therefore the delta

Send per-block checksums

Search for matching blocks

Whatever’s left is the difference

Integration with HTTP

Request/respond protocol

Streaming

Proxies

Every response may be different

Protocol

Client transmits signature of cached resource to server

Server computes & sends differences

Signature sent as new HTTP header

Delta as HTTP Transfer-Encoding

Ignored if not supported

Standalone Proxy

Run on on client, one upstream

Compress across slow links

Already in Debian/Woody

libhsync

Integrate smoothly with many apps

Become the encoding library for rsync 3.0

LGPL license for nonfree apps

Hosting Applications:

Mozilla: threaded

Apache: multi-process-model

Squid: select/poll-based

Therefore: do no IO in library; caller supplies buffer

Process through a state machine

Privacy problems?

Client holds server-supplied data & retransmits

A "stealth cookie"?

No more so than normal Last-Modified

Client-generated signatures are even safer

Tuning

Encode particular content-types

Fuzzy-matching of resources

Cache signatures

Choose block size

~90% saving

Other schemes

Explicit versioning

Client-side variable portions

Bonus slide: rsync 3.0

Fewer hardcoded features

Scriptable (Perl/Python/...)

Scale to larger trees (~100GB, 1M files)

Simpler client-server architecture

SSL?

rdiff tool: rsync-over-email?

Resources

http://linuxcare.com.au/rproxy/