

New mod_proxy Features and Uses for httpd 2.0

Graham Leggett

minfrin@apache.org

Chuck Murcko

chuck@apache.org

ApacheCon 2002 – TU09

November 19, 2002

New mod_proxy Features and Uses

- What does it do?
- A short history of mod_proxy
- Applying mod_proxy
- mod_proxy v2.0 architecture
- Future enhancements

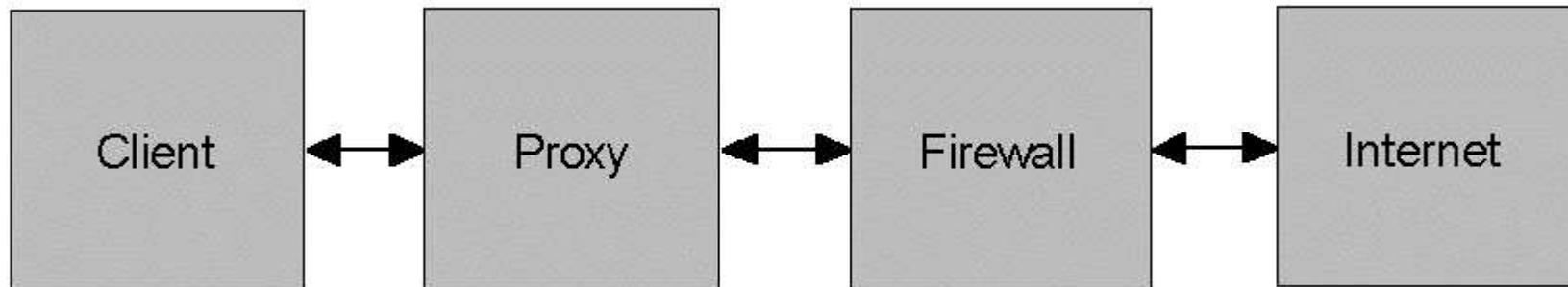
Applying mod_proxy

- Security
- URL Management
- SSL Acceleration
- Centralised Logging
- Centralised Authentication/Authorisation
- Centralised Caching/Compression

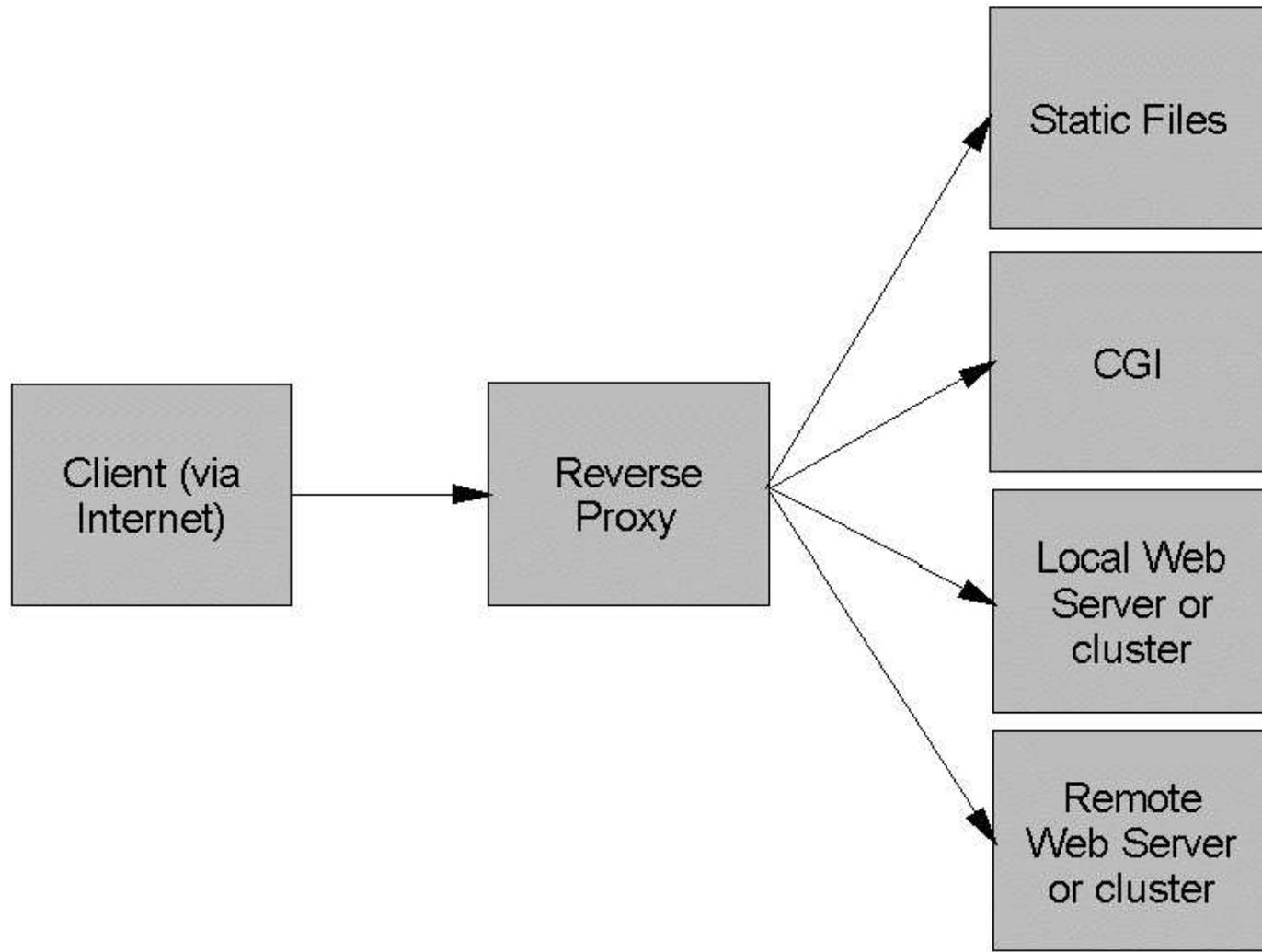
Forward Proxy example



Forward Proxy example



Reverse Proxy example



mod_proxy v2.0 Architecture

- Changes from v1.3
- How an HTTP request is handled
- How an FTP request is handled
- How a CONNECT request is handled

mod_proxy v2.0 architecture

Front end filter stack

network filters

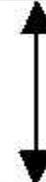


other filters
(SSL, etc.)

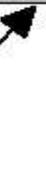


Back end filter stack

network filters



other filters
(SSL, etc.)



proxy handler



mod_proxy v2.0 Architecture

Changes from v1.3

- Cache code moved to mod_cache
- HTTP 1.1 support
- Separate protocol support modules
- Uses httpd 2.0 filter mechanism

mod_proxy v2.0 Architecture

How an HTTP request is handled

- Connections
- Keepalives
- Send Request
- Read Response
- Closing connections
- HTTP Pipelining
- Filter Stacks

mod_proxy v2.0 Architecture

How an FTP request is handled

- Connections
- Transfers
- FTP directory listings
- Reusing FTP connections

mod_proxy v2.0 Architecture

How a CONNECT request is handled

- In short, very simply
- CONNECT method is run as an HTTP tunnel

Future enhancements

- Better handling of back end connections
- Dynamically responding to server load
- Ability to specify multiple servers
- Session Persistence
- Pipelining
- Supporting more protocols
- Cookie rewriting
- Configuring the Back end Filter Stack

Most recent version is available at
<http://www.topsail.org/goodies/>

Thanks!