Proxy Cluster

Hironori Ito
Brookhaven National Labortory

Reasons for Proxy

- Storage are behind firewall.
  - Data traffic must go through the proxy servers.
- You want to control the access to the storage.
  - Proxy servers are used to control and/or limit the number of client requests to avoid instabilities in production storage

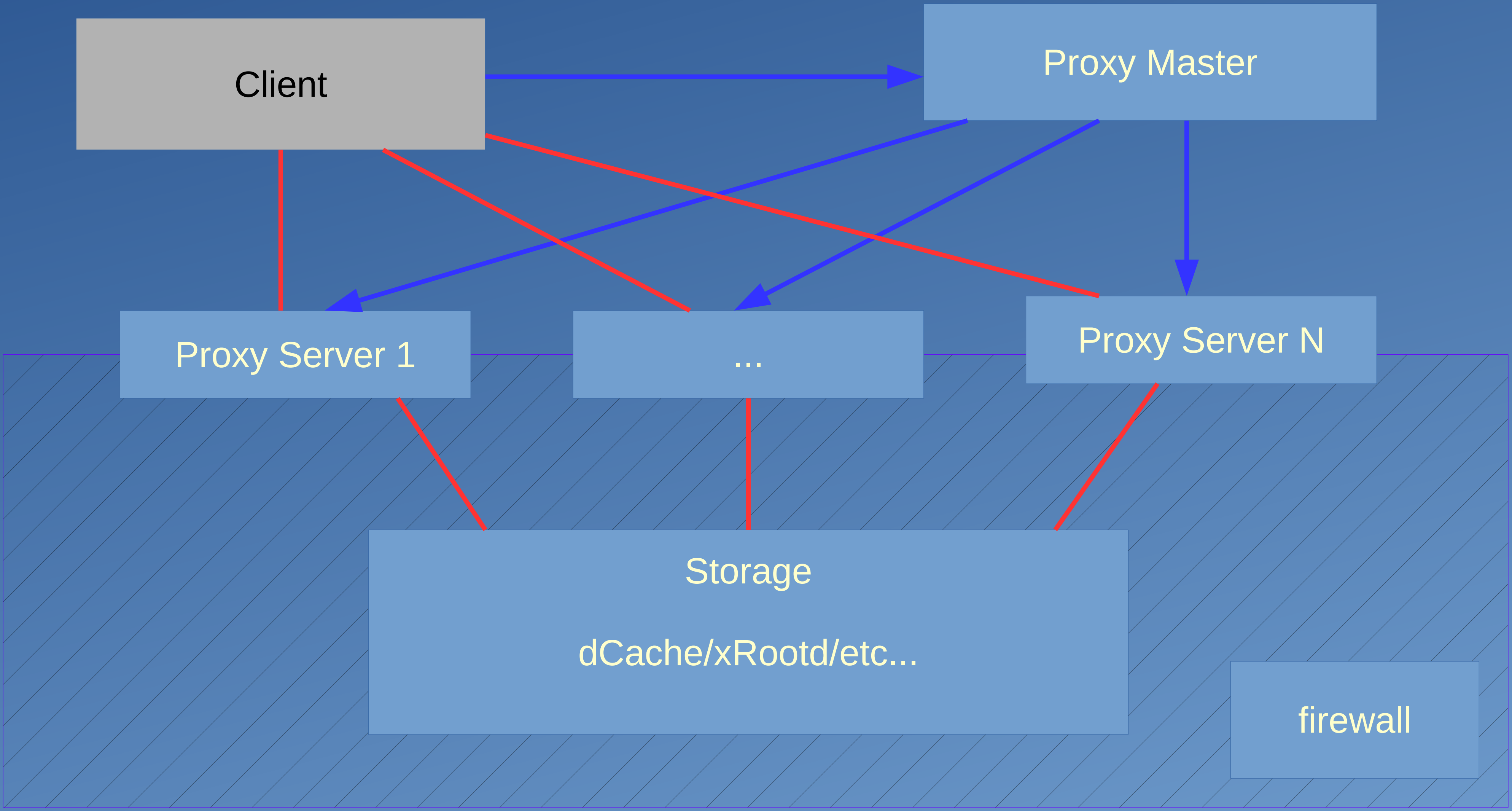
Behind firewall

- No choice but to poke a hole in the wall somewhere.
- Proxy Master + Proxy Servers
  - Both Master and Server must be outside the firewall.
    - Needs x509 (or other auth)
- Proxy Master
  - xRootd: xRootd local redirector
    - Data will not go through the service. Only control/meta-data are going through the service.
  - Https:
    - Reverse proxy is not great.
    - F5 smart switch or DNS alias/round robin
- Proxy Server:
  - Data will go through the servers.
  - Must have access to storage behind the firewall.
  - xRootd: xRootd data servers
  - Https: https server

Controlling access

- Don't underestimate the power of single, uneducated user
  - A single user has managed to shut down the source storage site
  - A single user has managed to shut down his/her own local/destination storage
- All traffic goes through proxy server(s).
- Number of proxy servers can be set according to the need.
  - eg.
    - To limit the remote access to 1Gb, use one 1Gb nic server for proxy server
    - To limit the number of concurrent access, set the limit in the proxy server.

xRootd Proxy



Client

Proxy Master

Proxy Server 1

...

Proxy Server N

Storage

dCache/xRootd/etc...

firewall

Basic ATLAS FAX Proxy Configuration with dCache

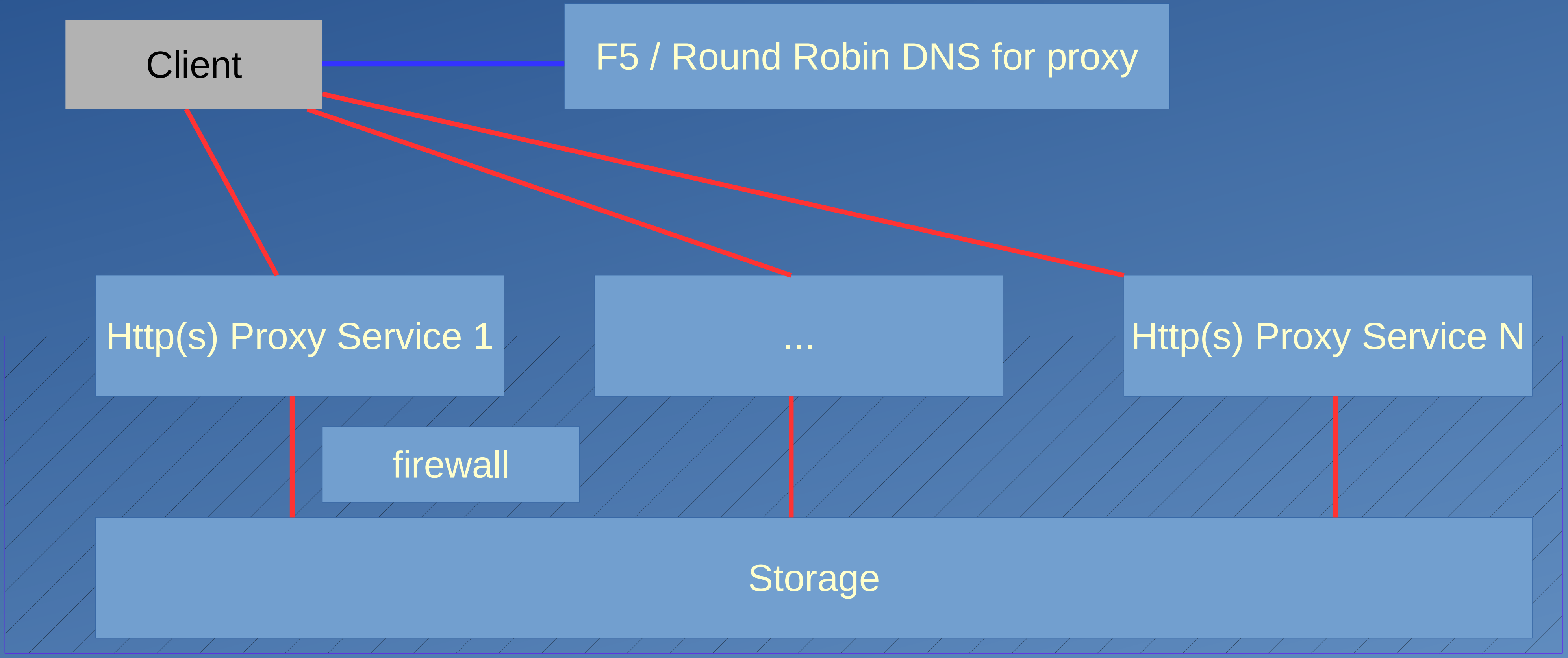
```
all.manager proxy PROXYMASTER_HOST:1213
all.manager meta REGIONAL_GLOBAL_REDIRECTOR_HOST:1095
all.export /atlas r/o

if PROXYMASTER_HOST
   all.role proxy manager
   cms.dfs lookup central
   cms.delay servers 0
else
   all.role proxy server
   xrootd.redirect REGIONAL_GLOBAL_REDIRECTOR_HOST:1094 ? /atlas
fi

pss.origin NATIVE_DCACHE_XROOTD_HOST:1096
pss.namelib /usr/lib64/XrdOucName2NameLFC.so
sec.protparm gsi -vomsfun:/usr/lib64/libXrdSecgsiVOMS.so -vomsfunparms:certfmt=raw|vos=atlas|grps=/atlas
xrootd.seclib /usr/lib64/libXrdSec.so
sec.protocol /usr/lib64 gsi -ca:1 -crl:3 -gmapopt:10
acc.authdb /etc/xrootd/auth_file
acc.authrefresh 60
ofs.authorize

all.sitename MY_SITE_NAME
```

Cluster Http(s)



Difference between Https and xRootd with x509 certificate
Https: Authentification happens at the first host (proxy master)
Use of X509v3 extensions - X509v3 Subject Alternative Name:

Https Cluster by Metalink

- Metalink
  - XML formatted file that describes one or more files for download
  - Features:
    - Checksum
    - Size
    - Multiple sources
    - Preference of sources
- Various clients: Firefox plugin (downthemall), aria2c, etc...
- Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<metalink xmlns="http://www.metalinker.org/" version="3.0">
  <files>
    <file name="MYFILE">
      <resources>
        <url type="https" location="us" preference="1">https://host1:PORT/PATH/MYFILE</url>
        <url type="https" location="us" preference="1">https://host2:PORT/PATH2/MYFILE</url>
      </resources>
    </file>
  </files>
</metalink>
```

Good / bad of cluster

- Cluster:
  - Good
    - Increase reliabilities
    - Control the level of access to the storage
  - BAD if DNS Round Robin (https)
    - If one fail, client request will always fail one out of N times.
- Metalink
  - Good
    - Various clients on the market
    - No need to create cluster master
    - Some clients are very smart.
      - Use multiple sources
        - Use M number of sources out of given N sources.
      - Can cope with dead sources
        - Use alternate sources in the given list.
  - Bad
    - Metalink must be created.