Re-ordering the cache manager server lists

Google Summer of Code 2009

Jacob Thebault-Spieker

Problem space

- The server lists are currently ordered based on network architecture assumptions that are no longer useful
- Rankings are based on whether the server is:
  - the local machine
  - on the same subnet as the client
  - on the same class network as the client
  - has no similarity in IP address to the client
- These metric used to suggest something about proximity between clients and servers, but this changed in 1993
- Networks no longer follow these rules, making these metrics obsolete

Goals

- To reimplement the server ordering algorithm based on current network conditions between client and server (best connection gets a better rank than less good connection).
  - Project has three steps:
    - define which network statistics are easily accessible via:
      - rx peer stats
      - network condition inferred from different RPC calls
    - implement the re-ordering algorithm within the unix cache manager
    - implement the re-ordering algorithm within the windows cache manager

Current status

- Working on implementing a basic initial sort based on:
  - the time it takes for a connection to a server to timeout
  - the ratio between packets sent and packets that needed to be resent
  - the round trip time (smoothed to account for short term network hiccups)
- Next steps include:
  - re-checking the network stats at some time interval
  - using data derived from application layer to guess at throughput between client and server

Any questions?

E-mail: theba004@morris.umn.edu
irc/jabber/aim: summatusmentis