Squirrel case-study

• Decentralized peer-to-peer web cache
  – Based on the Pastry peer-to-peer middleware system

• Traditional centralized web cache
**Squirrel case-study**

- **Decentralized caching of web pages**
  - use the resources of peers (web browsers/clients) to avoid expense (purchase / operation / maintenance) of centralized server
  - increase availability
  - distribute cache requests throughout the network

- **Cost of decentralized caching**
  - some additional latency (?)
  - additional load on client (network, bandwidth, …)

- **Questions**
  - how do we find a peer with the page we’re looking for?
  - where do we store cached web pages?
Squirrel case-study

• Each peer runs a local Squirrel proxy
  – No need any more for local web browser cache
  – The local cache previously provided by the browser is now provided by Squirrel and the contents of this cache are the peer’s contribution to the wider decentralized Squirrel cache
  – The aggregation of web pages in local caches forms the total, virtual Squirrel cache

• On page request
  – If not cacheable, contact origin server
  – Otherwise, if stored locally and fresh, serve page
  – Otherwise, locate a fresh, cached copy on another peer

• How do we locate another cached copy?
  – Compute GUID for page (GUID of URL)
  – Send (using Pastry) to the (numerically) closest node
  – This (numerically) closest node is the home node for the page
Squirrel case-study

- There are two alternative implementations of Squirrel and they diverge at this point in the implementation

- **Home-store** implementation
  - caches the page at the home node

- **Directory** implementation
  - caches the page at the clients that request the page
  - home node maintains a directory containing the K nodes that recently accessed each page that is mapped to the node
  - clients requests are forwarded by the home node to another (randomly chosen) node from the directory entry for the requested page
  - the home node is responsible for determining the “cachability” and freshness of pages
Squirrel case-study

Request routed through Pastry

a: object or notmod from home
b₃: object or notmod from origin

b₁: req

LAN

WAN

origin server
Squirrel case-study

a₁: no dir, go to origin. Also d₁
b: not-modified
c₁, e₁: req
e₂: cGET req
c₂, e₄: object
dir

a₂, d₂: req
a₃, d₃

client

other

home

delegate

origin server

object or not-modified

origin server
Squirrel case-study

• Five scenarios for the directory implementation
  1. Object not in local cache or home node directory – fetch from origin
  2. Object expired (not fresh) in local cache but validated by home node
  3. Object not present or present but not fresh in local cache and there is a fresh cached copy elsewhere, to which the request is forwarded
     - Object not present or present but not fresh in local cache and the directory entry tells us the cached copies are stale
  4. If original request was a conditional GET, original client is told to do a conditional GET from the origin server and update the home node with the response. Depending on the response, the home node will either revalidate its directory entries or set the client as the only delegate for the page
  5. If original request was just GET, then a randomly chosen delegate sends a conditional GET to the origin server and then forwards the page to the requesting client. In the mean time, the other delegates are disabled. Depending on the result, the chosen delegate will tell the home node to either (i) revalidate and re-enable the other delegates or (ii) set the client and delegate as the only delegates
**Squirrel case-study**

- **Squirrel performance**
  - External bandwidth
  - (hit ratio is directly related to external bandwidth)
Squirrel case-study

- Squirrel performance
  - Latency (measured in network hops)

- Mean: 4.11 hops and 4.56 hops for home-store and directory respectively
Squirrel case-study

- Squirrel performance
  - Peer load (maximum number of objects served by any node per second)
  
  - Average maximum: 1.5 objects/s and 6.6 objects/s for home-store and directory respectively
Squirrel case-study

• Squirrel performance
  – Fault tolerance
  – Average and maximum loss of harvest on single node failure?

• Squirrel benefits
  – Many parallel disk accesses with Squirrel
  – More in-memory disk cache hits with Squirrel