Co-operative Caching

What does it mean?

- From last week’s discussion:
  - Caches appropriately placed are effective at reducing network B.W, access latency, server load
  - Typical architecture: On a miss, content is fetched from a central server
- Today’s discussion:
  - Alternate architectures: On a miss, other caches are queried for the content before going to the central server
Outline

- Co-operative caching for Web
  - had a complete life-cycle; a researcher’s dream
  - proposed, developed and killed

- Co-operative caching for P2P

Motivation for the proposal

- Back in ‘97: Web traffic was growing at an exponential rate
  - doubling every 100 days 😄
- The most likely bottleneck was thought to be either the server or the network or the cache itself
- caches behind a common bottle-neck can co-operate with one another
  - Saves B.W, latency, improves load balancing on caches
Various Architectural Developments

- Caches are organized in a hierarchy
  - single hierarchy with more powerful caches at the top
  - multiple hierarchies (one per content) with fault tolerance schemes using consistent hashing
- Caches are organized in a peer-to-peer fashion
  - ICP: broadcast the query to all caches
  - Summary Cache: bloom filters to guess cache contents
- Lots of effort in developing better protocols that scale; little effort in understanding their payoff

The Death:
Are the potential benefits real?

- Relook at 4 presumed benefits
- Savings in network B.W
  - sharing between populations with similar interests
  - increase in effective client population
- Resources can be shared like in a cluster
  - Network resources; avoid overloaded cache, simultaneous downloads
  - Memory and CPU; dividing working set of documents between caches
Savings in network B.W

- Limitation:
  - B.W savings come from commonality in misses
  - Larger individual cache hit-rate => smaller number of misses => lesser savings with co-operation

- Questions:
  - Do clients grouped by their organizational network boundaries access similar or different documents?
  - Within a client group, how does hit-rate vary with population?

Is there a correlation between a client’s organization and its access pattern?

- Some correlation but not much
How does hit-rate vary with client population?

- Last week’s paper:
  - Zipf law => hit-rate a log(client population)
- Today we study it in real-world; specifically
  - co-operation between 160 organizational caches within UW
  - co-operation between caches for UW and MS

Benefits of co-operation for 15 largest organizations in UW

Figure 5. Breakdown of local and global proxy hit rates for the 15 largest UW organizations.
Benefits of co-operation between UW and MS caches

![Bar chart showing hit rate comparison between Cooperative and Local caching strategies for UW Ideal, UW Cacheable, MS Ideal, and MS Cacheable cases. The Cooperative caching strategies consistently outperform the Local ones.]

**Figure 8.** Hit rate benefit of cooperative caching between UW and Microsoft proxies.

Variation of hit-rate with Client population

![Line graph showing cacheable hit rate as a function of client population (log scale) for different load phases: Slow (14 days, 186 days), Mid Slow (1 day, 186 days), Mid Fast (1 hour, 65 days), Fast (6 mins, 85 days).]

**Figure 9.** Cacheable request hit rate as a function of client population (log scale).
Summary of Potential B.W. Savings

- Benefits of co-operative caching decrease with the size of client population
- For caches serving a few hundred or thousand clients, benefits are reasonable
- For caches serving tens of thousands of clients, it does not add much

Potential Benefits of Co-operative Caching

- Savings in network B.W
  - sharing between populations with similar interests
  - increase in effective client population
- Resources can be shared like in a cluster
  - Network resources; avoid overloaded cache, simultaneous downloads
  - Memory and CPU; dividing working set of documents between caches
Sharing cache resources

- Network B.W
  - simultaneous downloads for 10KB web pages??
  - most often the bottleneck is the client’s first hop link not the cache’s B.W
- Memory
  - 1 GB can hold the most popular 100,000 documents
- Processor
  - cannot be the bottleneck for a cache serving web pages

Summary: Co-operative Caching for Web

- Most of the supposed benefits do not materialize
- It’s life-cycle is a researcher’s dream come true
  - papers published proposing the idea
  - papers published developing the idea
  - papers published killing the idea
- Goal: Relive the dream for P2P
Proposal for Co-operative Caching of P2P Workloads

- P2P workloads differ sufficiently from the Web to warrant research
- What does co-operative caching in P2P mean?
  - in p2p world, peer = client = server = cache
  - caching refers to exploit locality in requests
  - peers prefer to fetch content from a nearby peer over a random peer
- Why should co-operative caching work for P2P workloads and not for Web?

Potential Benefits of Co-operative Caching

- Savings in network B.W
  - sharing between populations with similar interests
  - increase in effective client population
- Resources can be shared like in a cluster
  - Network resources; avoid overloaded cache, simultaneous downloads
  - Memory and CPU; dividing working set of documents between caches
Savings in network B.W

- Savings in network B.W
  - client’s interests unlikely to vary with organization they belong to
  - object popularity: resembles sampling from a zipf distribution without replacement => limitations from population size
  - up to 80% cache hit-rates for client populations as small as a few hundred hosts
  - the hit-rate is achieved over a period of few months

Sharing resources

- Network B.W
  - simultaneous downloads can help greatly to reduce load on server and improve latency of download
  - bottleneck is often the peer’s upload B.W
- Memory
  - 1 GB can hold the most popular 1 document
- Disk space
  - Many tens to a few hundred terabytes of storage required