Secure and Reliable Decentralized Peer-to-peer Web Cache

Presented by
Chi-Hung Chao

Outline

- Introduction
- Decentralized Web Cache
- Performance Evaluation
- Conclusion

Introduction

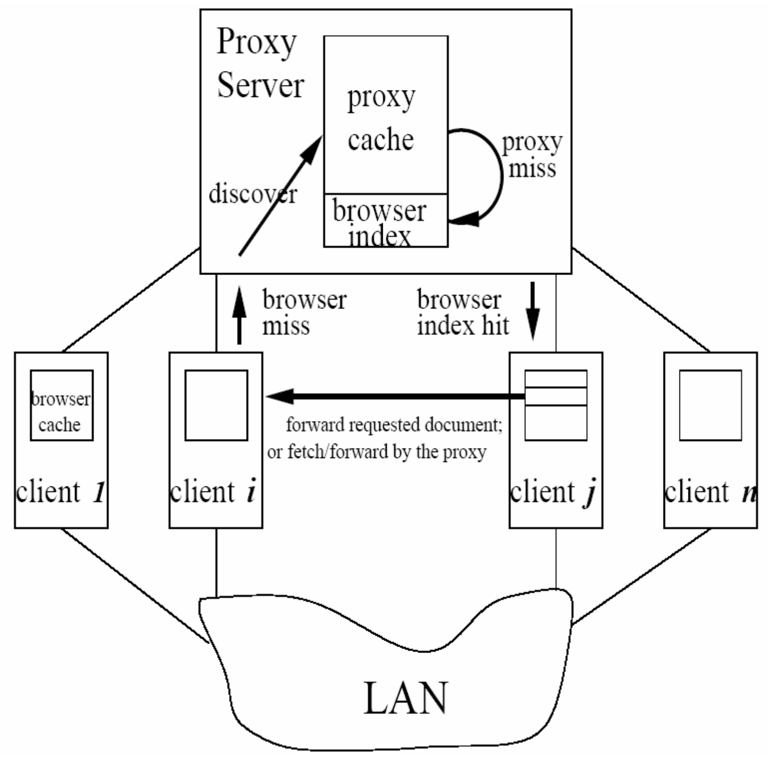
- Web caching is a well developed scheme for improving the performance of web browsing.
- Users' requests can be satisfied by the local cache or the cache of a nearby proxy, instead of a remote web server.
- Performance Studies[3,11,22].
- Various proxy caching techniques and replacement[1,4,12,14,22].
- Cooperative caching architectures and mechanism[5,15,21,23].

Introduction

- One class of the cooperative caching strategies is based on the client-side caching.
  - In a LAN environment, the clients' cache space can be organized and utilized to reduce the total outgoing traffic.
  - A client may get web contents from other clients.
  - Simulation results indicate that sharing cache among clients can significantly decrease the external traffic.
  - However, client-side caching has some fundamental problems:
    - Availabilities of the client platforms are unpredictable.
    - The privacy of users' access activities should be protected.

Introduction

- Centralized peer-to-peer web caching protocol[24].
- Centralized approaches incur a higher management cost, especially in a large domain.

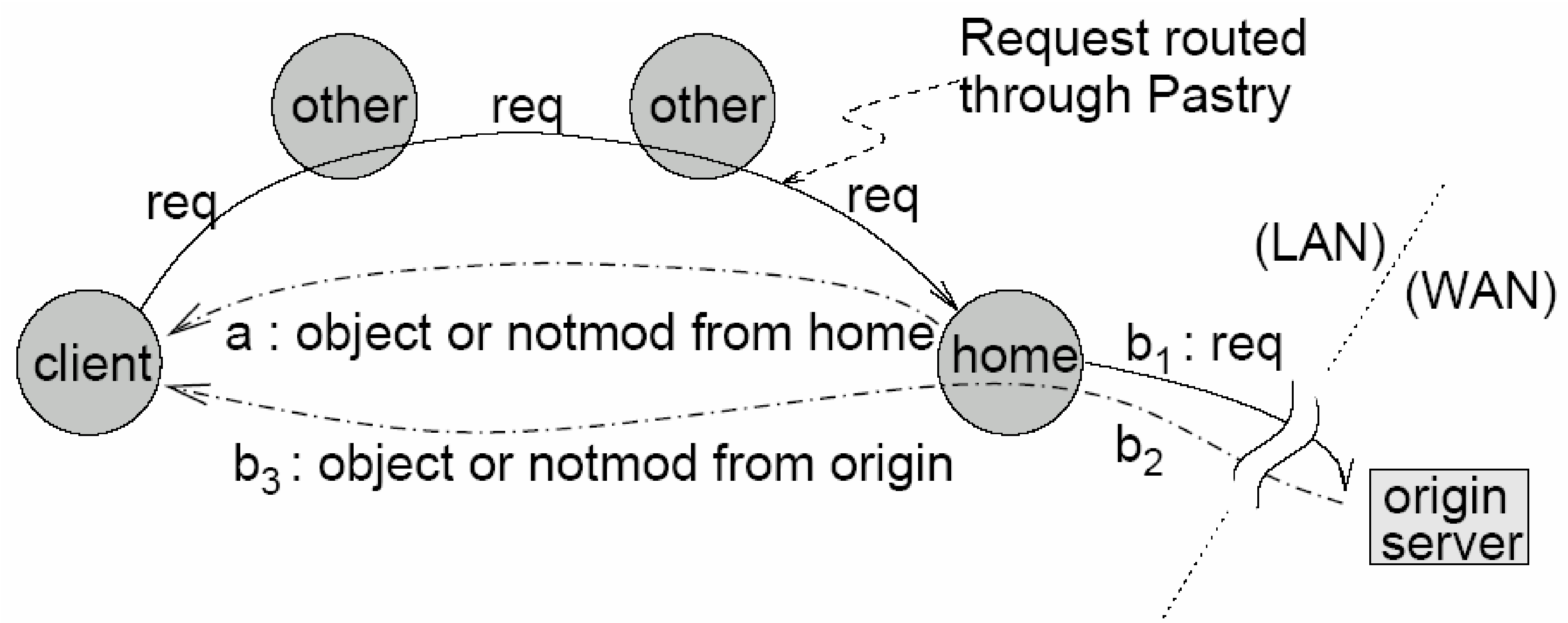


Introduction

- Decentralized peer-to-peer web cache mechanism – Squirrel[9].
- Squirrel considers two document look-up protocols, home store and directory.

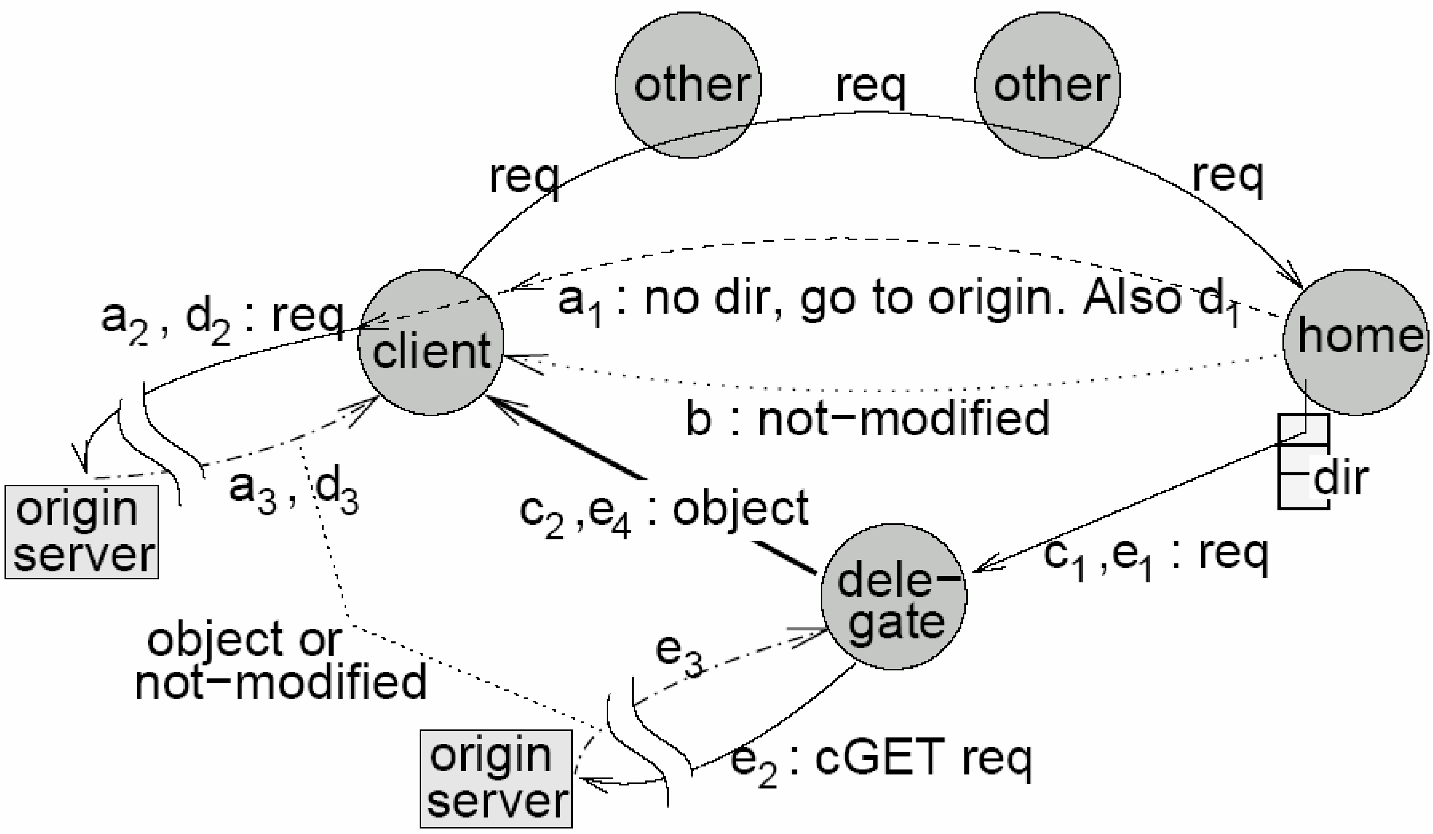
Introduction

- Home store


Request routed through Pastry
other
req
other
req
req
(LAN)
(WAN)
client
a : object or notmod from home
home
b_1 : req
b_2
origin server
b_3 : object or notmod from origin

Introduction

- Directory



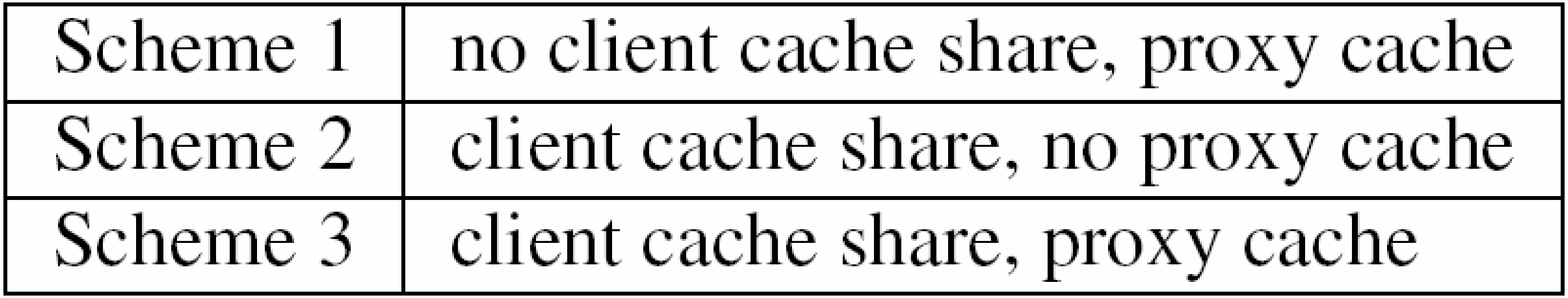
Introduction

- Privacy issues are not considered in either of the look-up protocols.
- Though these schemes yield a lower external bandwidth, they incur a much higher internal traffic.
- The high hit ratio in these schemes are obtained based on some idealistic assumptions that may not be satisfied in a real environment, such as low failure rate and some necessary operations when nodes join or leave the system.

Decentralized Web Cache

- This paper presents a web caching model based on the decentralized peer-to-peer architecture.
- Using a hybrid policy that combines existing home store and directory approaches.
- The combined approach eliminates the problems in the original home store and directory schemes.
- Developing new algorithms and policies to resolve certain problems that arise due to the combination.
- A new cache replacement policy based on the hybrid approach has been developed.
- Encoding the web contents and hiding IP information to protect clients’ privacy.

Decentralized Web Cache



Scheme 1	no client cache share, proxy cache
Scheme 2	client cache share, no proxy cache
Scheme 3	client cache share, proxy cache

- The home store method represents scheme 1.
- The directory approach yields scheme 2.
- The new design in this paper is based on scheme 3.

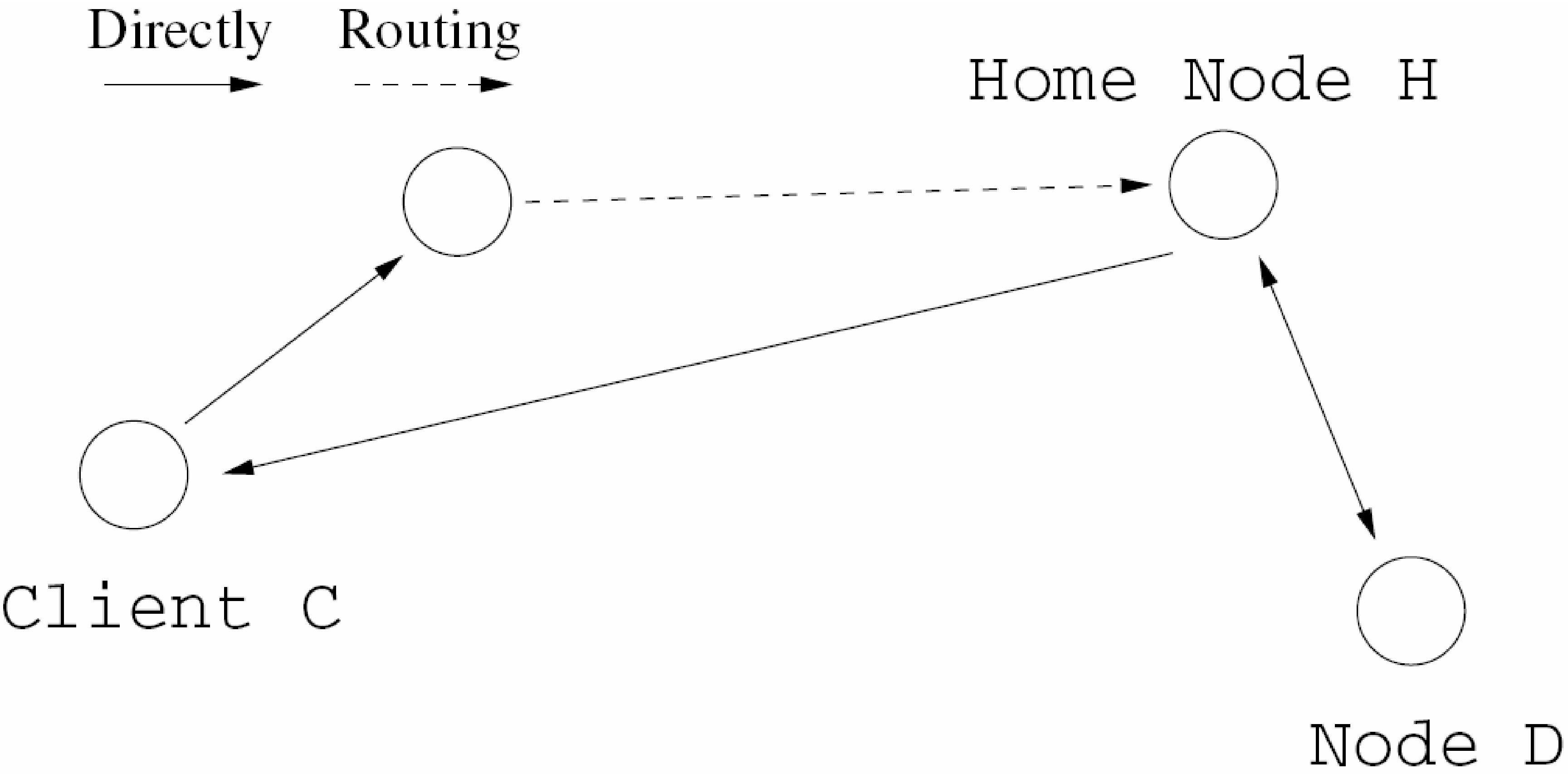
Decentralized Web Cache

- Splitting the client cache space into two parts.
  - Local cache and Home node cache
  - *psize* denote the Home node cache size
  - *lsize* denote the Local cache size
  - Both *psize* and *lsize* are arguments that can be set by users.

Decentralized Web Cache

- Access Request Processing
  - While a client C requests a web object, it first checks its local cache. If the object is not found, a request message will be forward to the home node H.
  - If the home node has cached a copy, it sends the object to C. Otherwise, H randomly pick a delegate D from the directory for the transferring. If the directory is empty, the requesting client will fetch the object from the original server.

Decentralized Web Cache



Decentralized Web Cache

- Cache Replacement
  - Remove pages with expired TTL.
  - When the home node evicts a web object it hosts, it informs all holders of that object so that they can increase the priority of that object in their local cache.
  - On the other hand, when a client evicts a hot page, if it is the last copy or one of the last few copies, it sends the page to the home node.

Decentralized Web Cache

- Node Load Smoothing
  - It’s important to keep the load evenly distributed among them.
  - In the directory approach, the number of served objects per second can be burst.
  - A stricter solution for load balancing is to use a reference counter. Each node keeps an upper bound *rlimit,* which represents the maximum number of nodes it can be referenced by.
    - *When a home node H wants to add a client C to its directory, H first sends a message to C. C checks its current value of the reference counter. If it’s less then rlimit, C sends an ‘accept’ response to H and increases the reference counter. Otherwise, C sends ‘’deny response to H.*

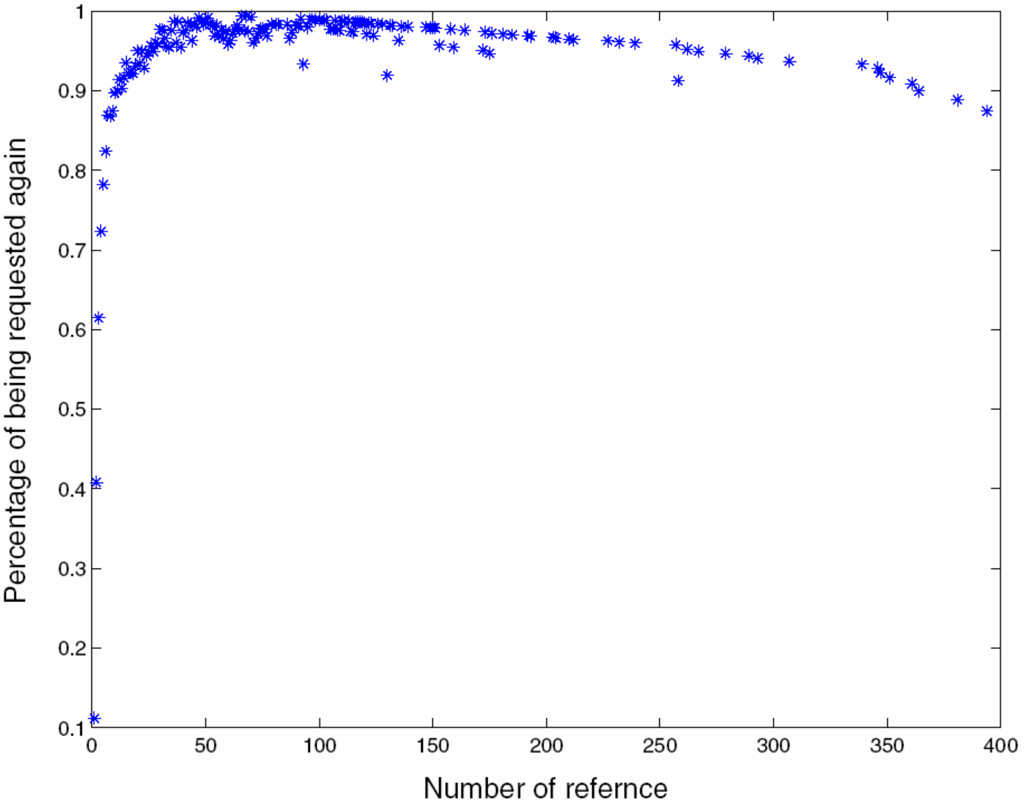
Decentralized Web Cache

- Privacy and Security Issues
  - In web caching systems, a node may not want the others to trace its access history.
  - In the centralized scheme, this could be done by the trusted proxy servers. However, it’s more complex in this self-organizing approach.
  - (IP, Hash(URL) / $(P)_k$)

    K=f(URL)

Performance Evaluation

- Access Frequency

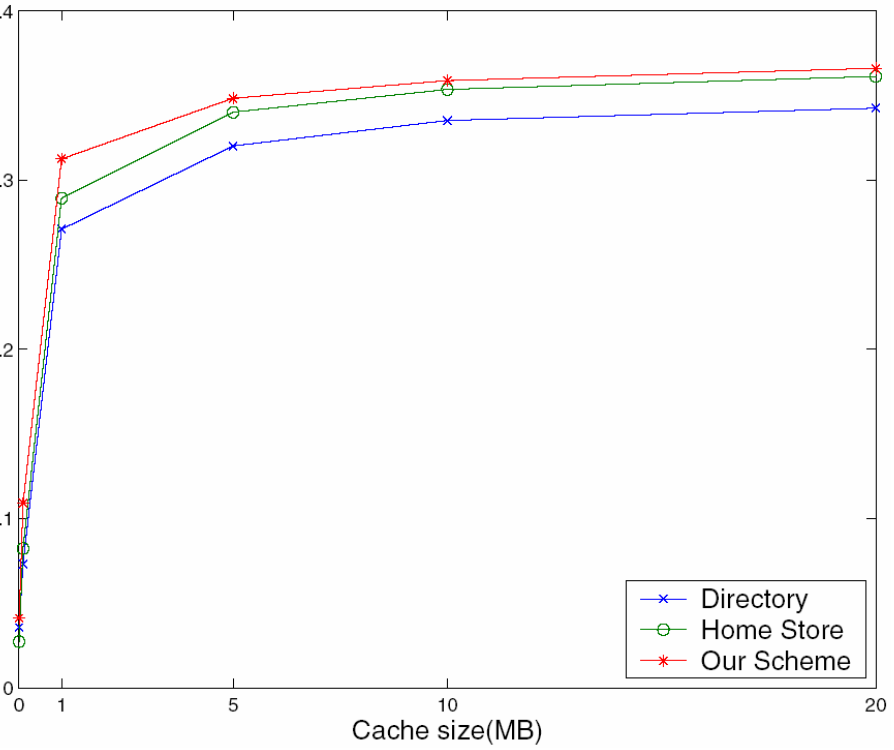


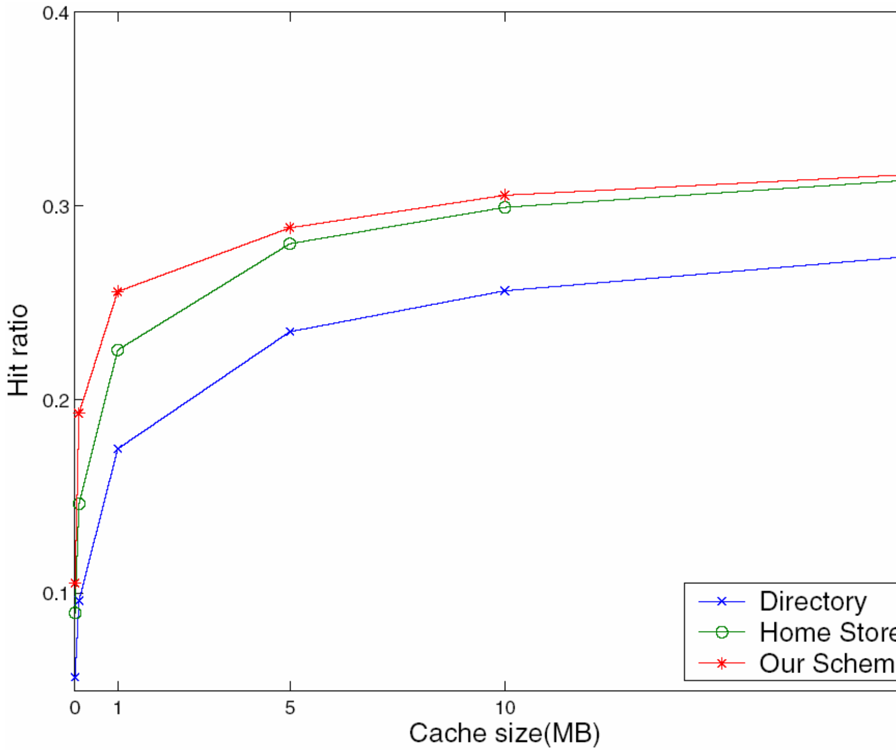
Performance Evaluation

- Performance
  - We assume that each node has the same size of cache space and we assign the size to 10K, 100K, 1M, 5M, 10M and 20M.
  - We set *psize* to 30% of the total cache size and TTL to one hour.

Performance Evaluation

- Performance





Conclusion

- The experimental study shows that it performs better than the existing schemes in term of hit ratio and node load.
- It may be necessary to partition these objects and cache them on multiple client platforms to avoid occupying a large cache space of a single client and facilitate load smoothing.