

# How to save 150,000 dollars a year

## InHome: Peer-to-Peer Local Area Caching

Mihir Kedia, Raluca Ada Popa, Irene Zhang

May 9, 2008

# Motivation

- Wide-area bandwidth is becoming increasingly scarce
  - Bandwidth-hungry applications like Youtube are outpacing infrastructure upgrades
- Local-area bandwidth is cheap and often unused
- Much of the data traversing outgoing links is redundant
  - 25-40% of web requests made within a given organization are duplicates of previous requests.

# Objective

**Reduce external bandwidth usage by sharing data among peers inside an organization.**

System requirements:

- Clients should not see a significant increase in latency
- Clients should not store data they are not interested in
- System should be customizable for organization sizes
- System should not require new hardware or maintenance

# System Overview

- InHome is implemented as a peer-to-peer network that operates like a distributed cache
- Clients run a background daemon that automatically syncs metadata from InHome-aware applications
  - Application-specific plugins query the InHome network for data before falling back to the origin server
- Consistent hashing is used for fast object lookup
- Interface:

```
put(name, data)
data = get(name)
```

## Example Usage: Web Caching

For each HTTP request:

- 1 Mozilla plugin queries the InHome client for the URL
- 2 InHome client hashes the URL into a 160-bit object ID
- 3 InHome client searches the InHome peers for the object ID
- 4 If the search succeeds, Mozilla plugin will return cached data after checking the TTL
- 5 If the search fails or times out, the Mozilla plugin tells Mozilla to fetch the page from the origin server in the normal way
- 6 Mozilla plugin registers the new data with the InHome client by inserting the data with the URL

# Search Algorithms

## Basic Consistent Hashing

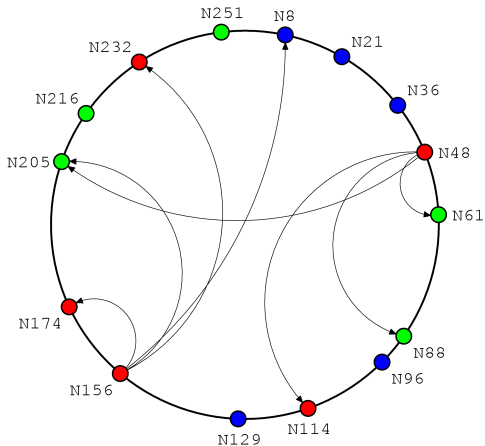
- Full membership
- One-hop lookup
- Metadata maintenance
- No fate sharing

## Data-oriented Chord

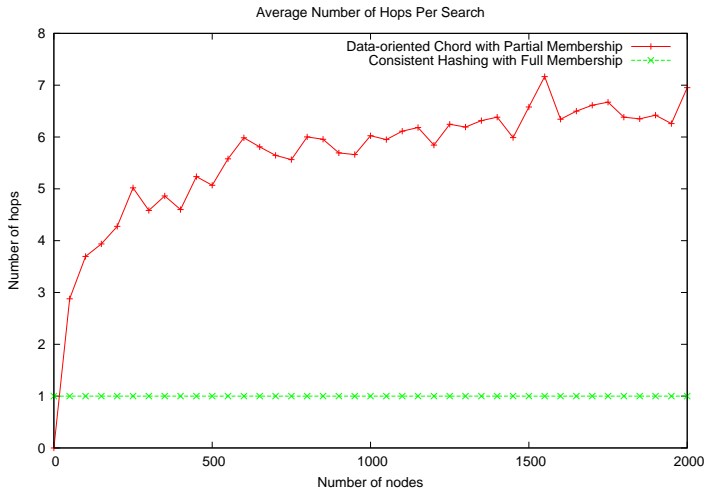
- Partial Membership
- $\log(n)$  hop lookup
- No metadata
- Fate sharing



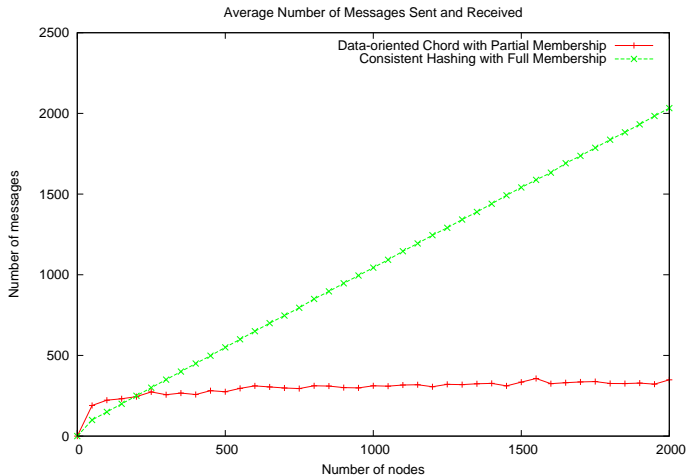
# Data-Oriented Chord



# Performance Comparison



# Bandwidth Comparison



# Bandwidth Savings

## UC Berkeley Traces – 11/96

- Duration: 4 hours
- Hit rate: 24.3%
- Bandwidth Savings: 27.6%

## IRCache Traces – 1/10/07

- Duration: 1 day
- Hit rate: 37.6%
- Bandwidth Savings: 41.5%

## Bandwidth Savings, cont.

### Zipf Distribution

- Hit rate: 43.2%
  - Bandwidth Savings: 45.7%
- 
- For an institution like MIT, a 35% reduction in web traffic could save \$210,000/year.

## Related Work

- First distributed caching solution (most companies use centralized proxy server)
- Has been research into local BitTorrent – selecting local peers first
  - Ono project
  - Stanford analysis

- InHome can save wide-area bandwidth by fetching data from local peers
- InHome does not worsen user experience
- InHome realizes substantial savings in external bandwidth

Questions?