Measurement, Modeling and Analysis of a Peer-to-Peer File-Sharing Workload

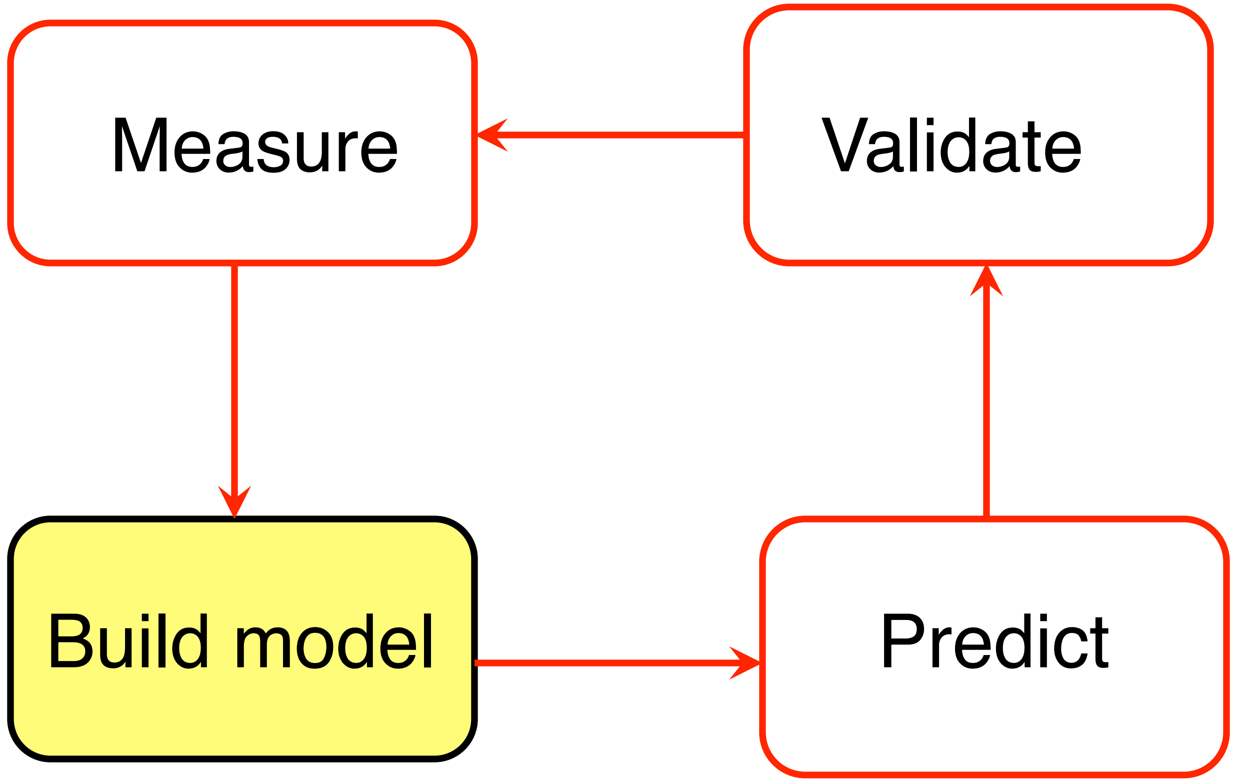
Krishna Gummadi, Richard Dunn, Stefan Saroiu
Steve Gribble, Hank Levy, John Zahorjan

Several slides were taken from the original presentation by Gummadi

The Internet has changed

- Explosive growth of P2P file-sharing systems
  - now the dominant source of Internet traffic
  - its workload consists of large multimedia (audio, video) files
- P2P file-sharing is very different than the Web
  - in terms of both workload and infrastructure
  - we understand the dynamics of the Web, but the dynamics of P2P are largely unknown

Why measure?



The current paper

Studies the KazaA peer-to-peer file-sharing system, to understand two separate phenomena

- Multimedia workloads
  - ***what*** files are being exchanged
  - goal: to identify the forces driving the workload and understand the potential impacts of future changes in them
- P2P delivery infrastructure
  - ***how*** the files are being exchanged
  - goal: to understand the behavior of Kazaa peers, and derive implications for P2P as a delivery infrastructure

KazaA: Quick Overview

- Peers are individually owned computers
  - most connected by modems or broadband
  - no centralized components
- Two-level structure: some peers are "super-nodes"
  - super-nodes index content from peers underneath
  - files transferred in segments from multiple peers simultaneously
- The protocol is proprietary

Methodology

- Capture a 6-month long trace of Kazaa traffic at UW
  - trace gathered from May 28th – December 17th, 2002
    - passively observe all objects flowing into UW campus
    - classify based on port numbers and HTTP headers
    - anonymize sensitive data before writing to disk
- Limitations:
  - only studied one population (UW)
  - could see data transfers, but not encrypted control traffic
  - cannot see internal Kazaa traffic

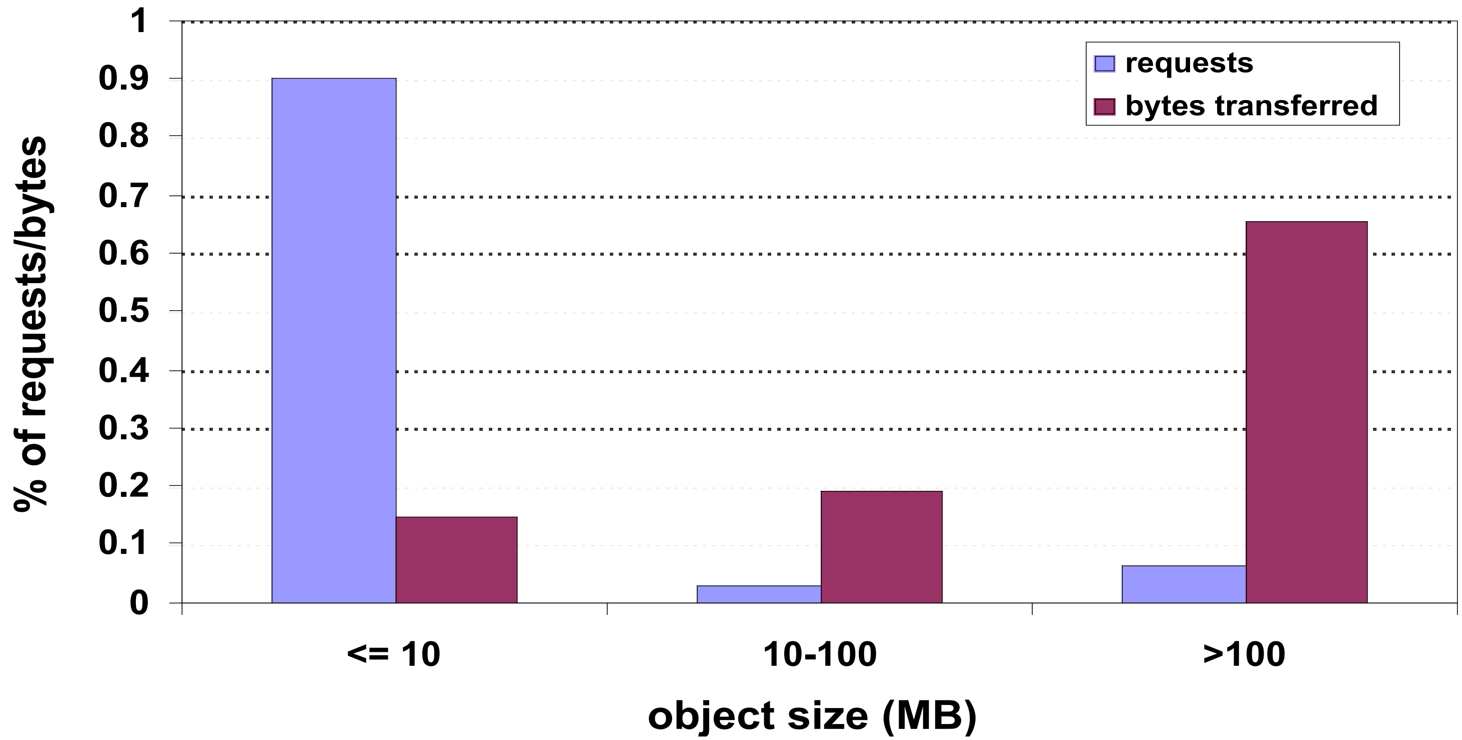
Trace Characteristics

start date	May 28th, 2002
end date	December 17th, 2002
trace length	203 days, 5 hours, 6 minutes
# of requests	1,640,912
# of transactions	98,997,622
# of unsuccessful transactions	65,505,165 (66.2%)
# of clients	24,578
# of unique objects	633,106 (totaling 8.85TB)
bytes transferred	22.72TB
content demanded	43.87TB

Outline

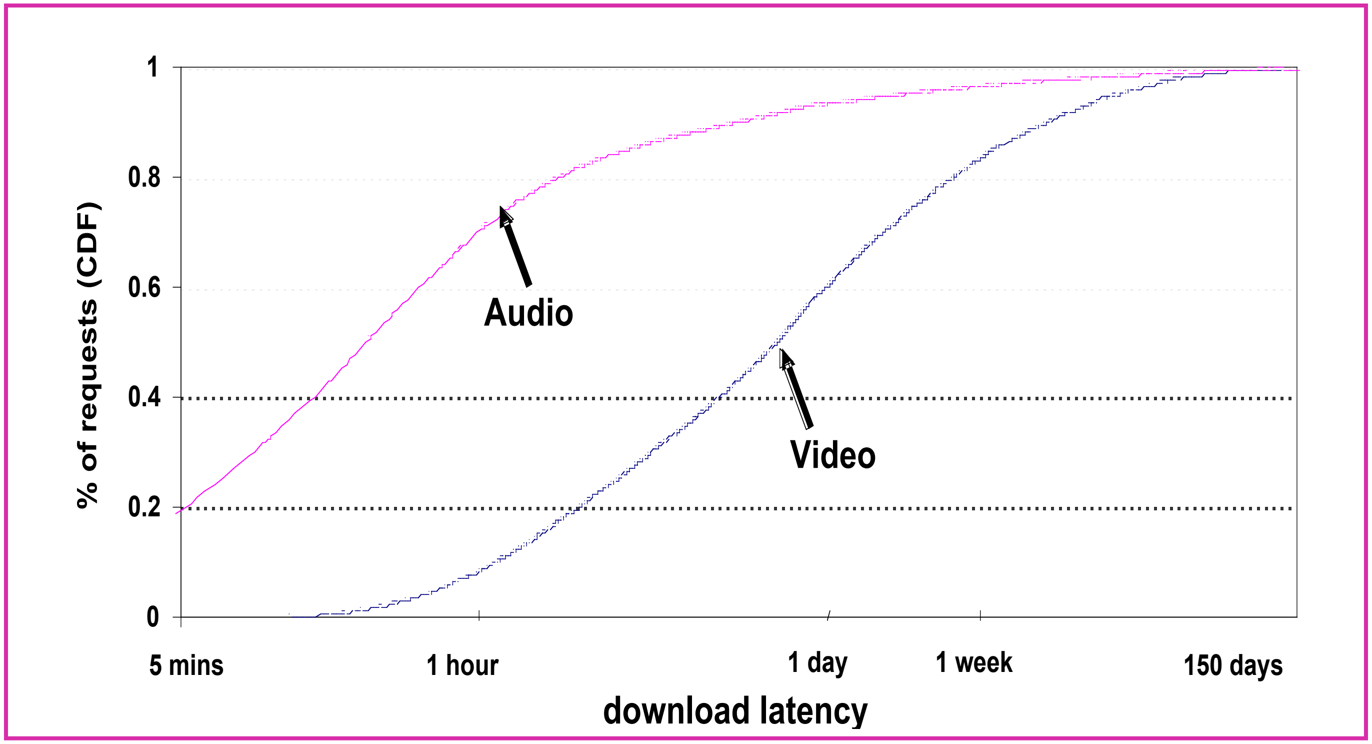
- Introduction
- Some observations about Kazaa
- A model for studying multimedia workloads
- Locality-aware P2P request distribution
- Conclusions

Kazaa is really 2 workloads



- making users happy: make sure audio/video arrives quickly
- making IT dept. happy: cache or rate limit video

Kazaa users are very patient



- audio file takes 1 hr to fetch over broadband, video takes 1 day
  - but in either case, Kazaa users were willing to wait for weeks!

Kazaa objects are immutable

- **The Web is driven by object change**
  (many visit cnn.com every hour. Why?)
  - users revisit popular sites, as their content changes
  - rate of change limits Web cache effectiveness [Wolman 99]
- **In contrast, Kazaa objects never change**
  - as a result, users rarely re-download the same object
    - 94% of the time, a user fetches an object at-most-once
    - 99% of the time, a user fetches an object at-most-twice
  - implications:
    - # requests to popular objects bounded by user population size

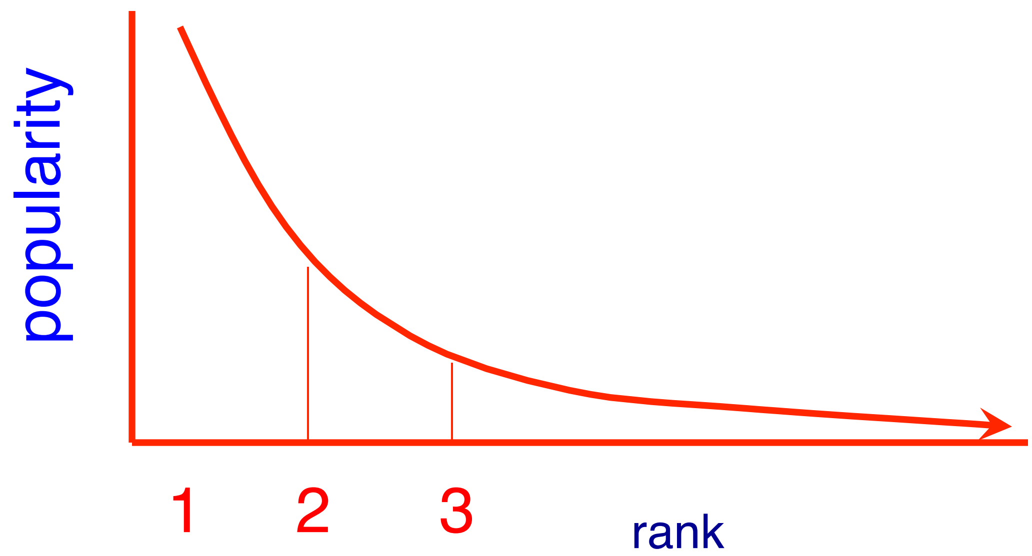
Kazaa popularity has high turnover

- **Popularity is short lived: rankings constantly change**
  - only 5% of the top-100 audio objects stayed in the top-100 over our entire trace [video: 44%]

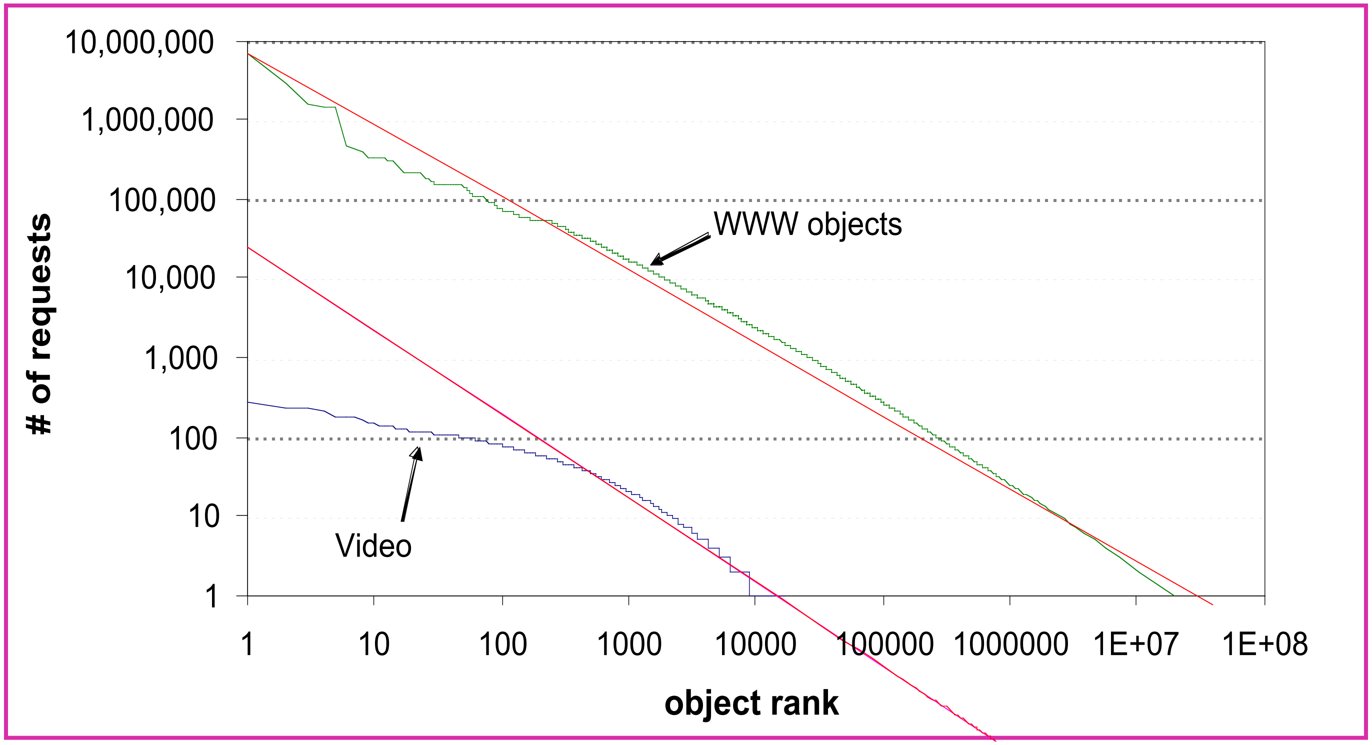
- **Newly popular objects tend to be recently born**
  - of audio objects that “broke into” the top-100, 79% were born a month before becoming popular [video: 84%]

Zipf distribution

Zipf's Law states that the popularity of an object of rank k is $1/k^r$ of the popularity of the top-ranked object ($1 < r < 2$).



Kazaa does not obey Zipf's law



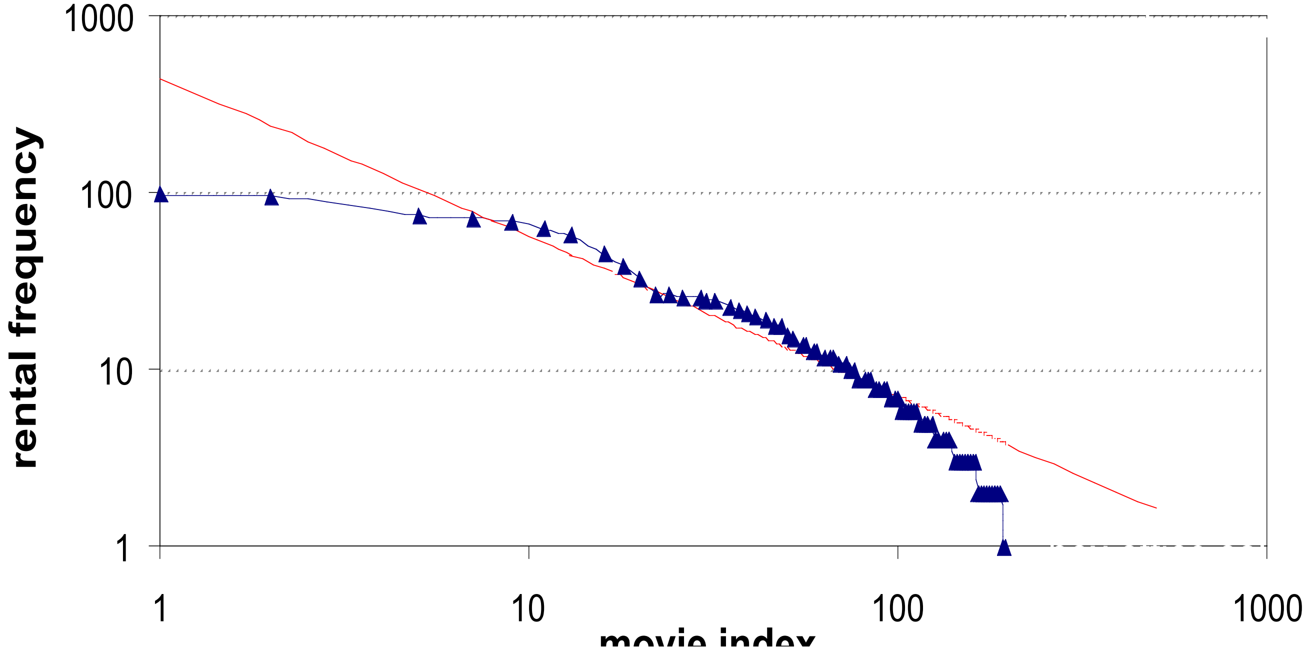
- Kazaa: the most popular objects are 100x less popular than Zipf predicts

Factors driving P2P file-sharing workloads

- The traces suggest two factors drive P2P workloads:
  1. Fetch-at-most-once behavior
     - resulting in a “flattened head” in popularity curve
  2. The “dynamics” of objects and users over time
     - new objects are born, old objects lose popularity, and new users join the system

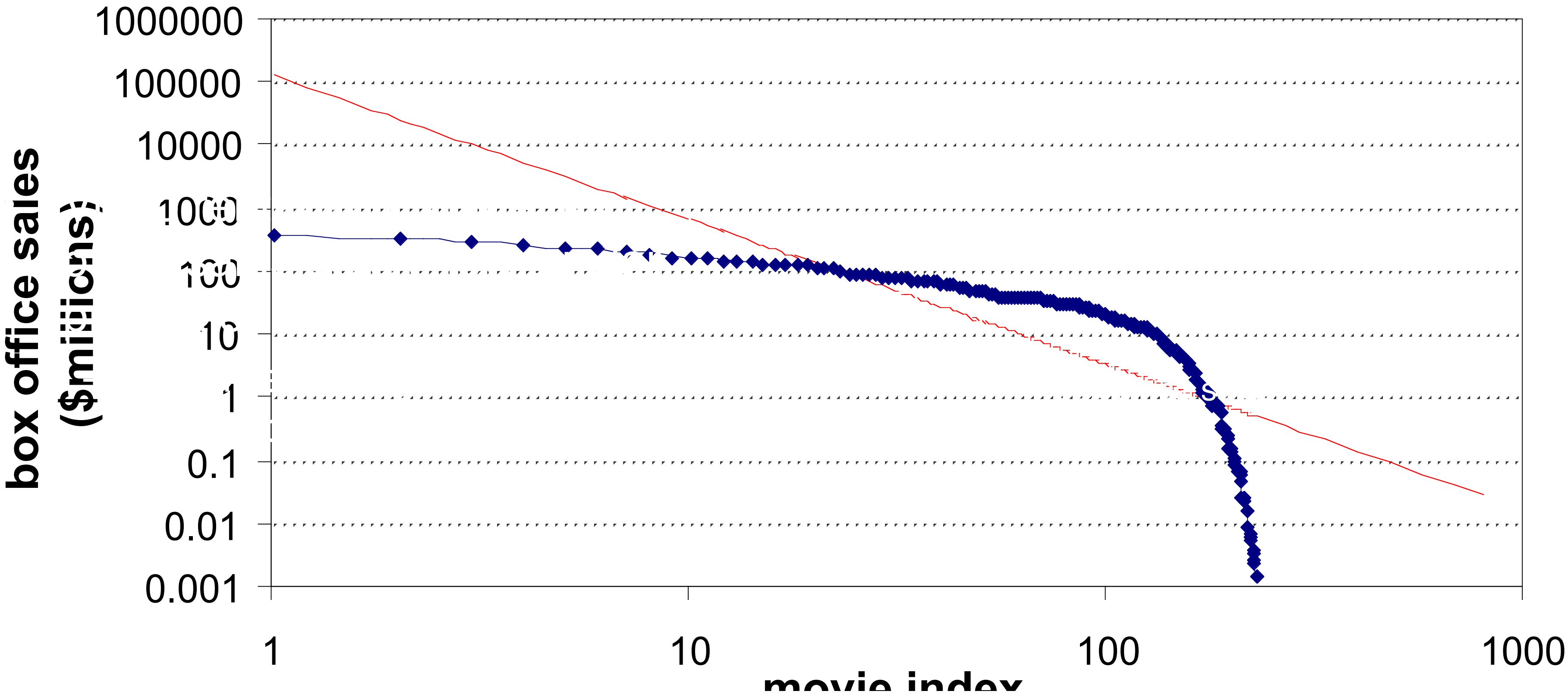
- Let’s build a model to gain insight into these factors

It’s not just Kazaa



- Video rental and movie box office sales data show similar properties
  - multimedia in general seems to be non-Zipf

It's not just Kazaa



Outline

- Introduction
- Some observations about Kazaa
- A model for studying multimedia workloads
- Locality-aware P2P request distribution
- Conclusions

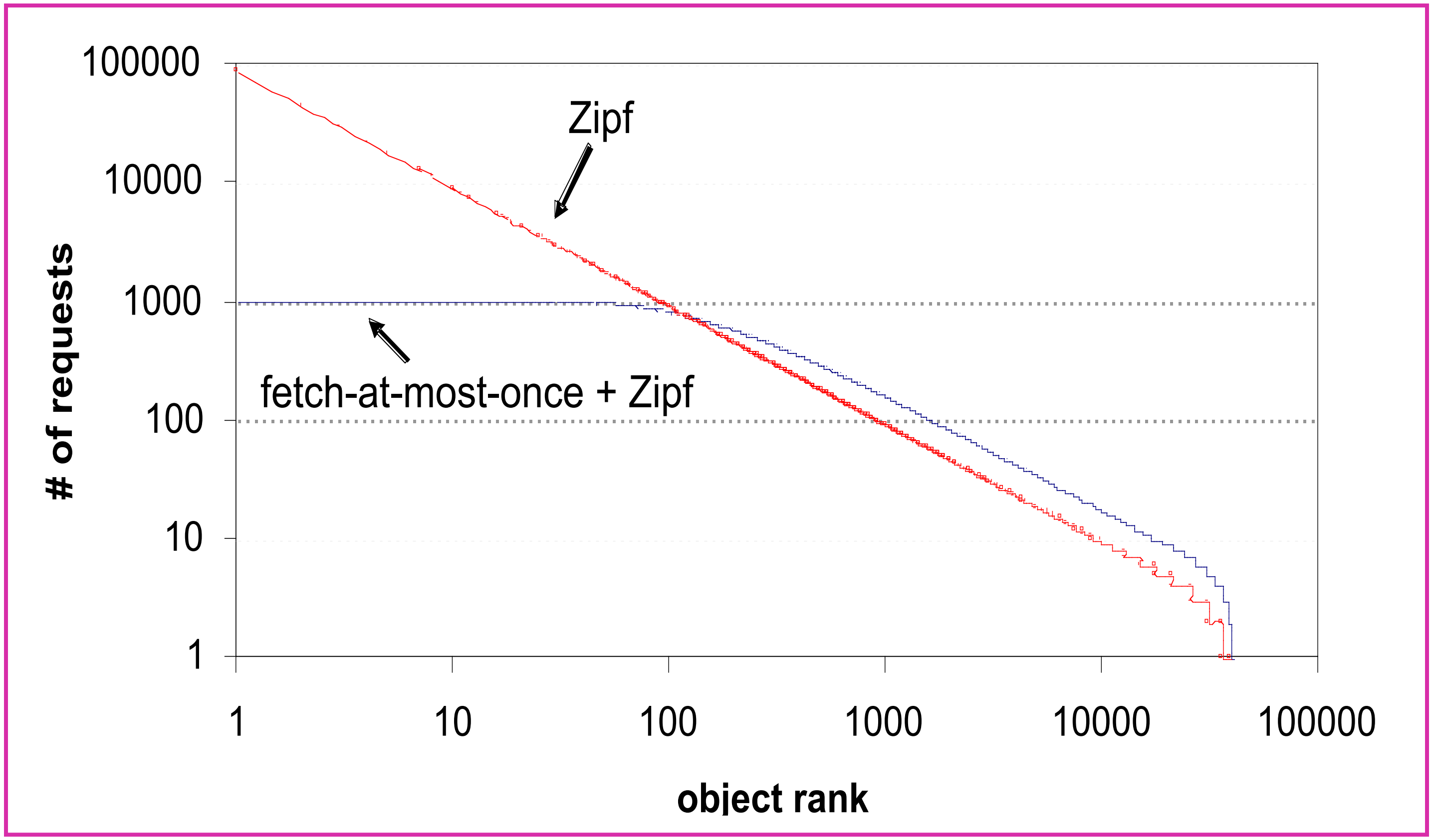
Model basics

1. Objects are chosen from an underlying Zipf curve
2. But we enforce "fetch-at-most-once" behavior
   - when a user picks an object, it is removed from her distribution
3. Fold in "user, object dynamics"
   - new objects inserted with initial popularity drawn from Zipf
     - new popular objects displace the old popular objects
   - new users begin with a fresh Zipf curve

Model parameters

C	# of clients	1,000
O	# of objects	40,000
λ_R	client req. rate	2 objs/day
r	Zipf param driving obj. popularity	1.0
P(x)	prob. client req. object of pop rank x	Zipf (1.0) + fetch-at-most-once
A(x)	prob. of new object inserted at pop rank x	Zipf (1.0)
M	cache size (frac. of obj)	varies
λ_O	object arrival rate	varies
λ_c	client arrival rate	varies

Fetch-at-most-once flattens Zipf's head

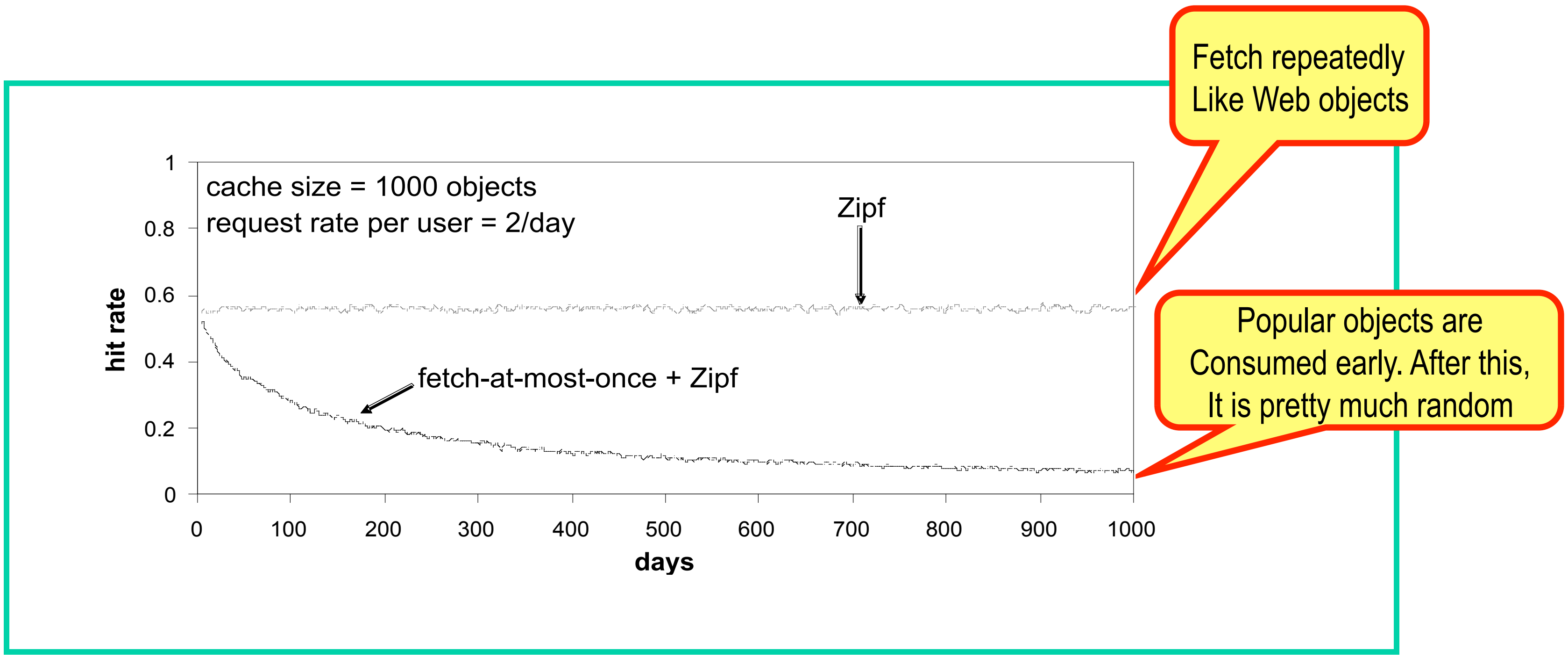


File sharing effectiveness

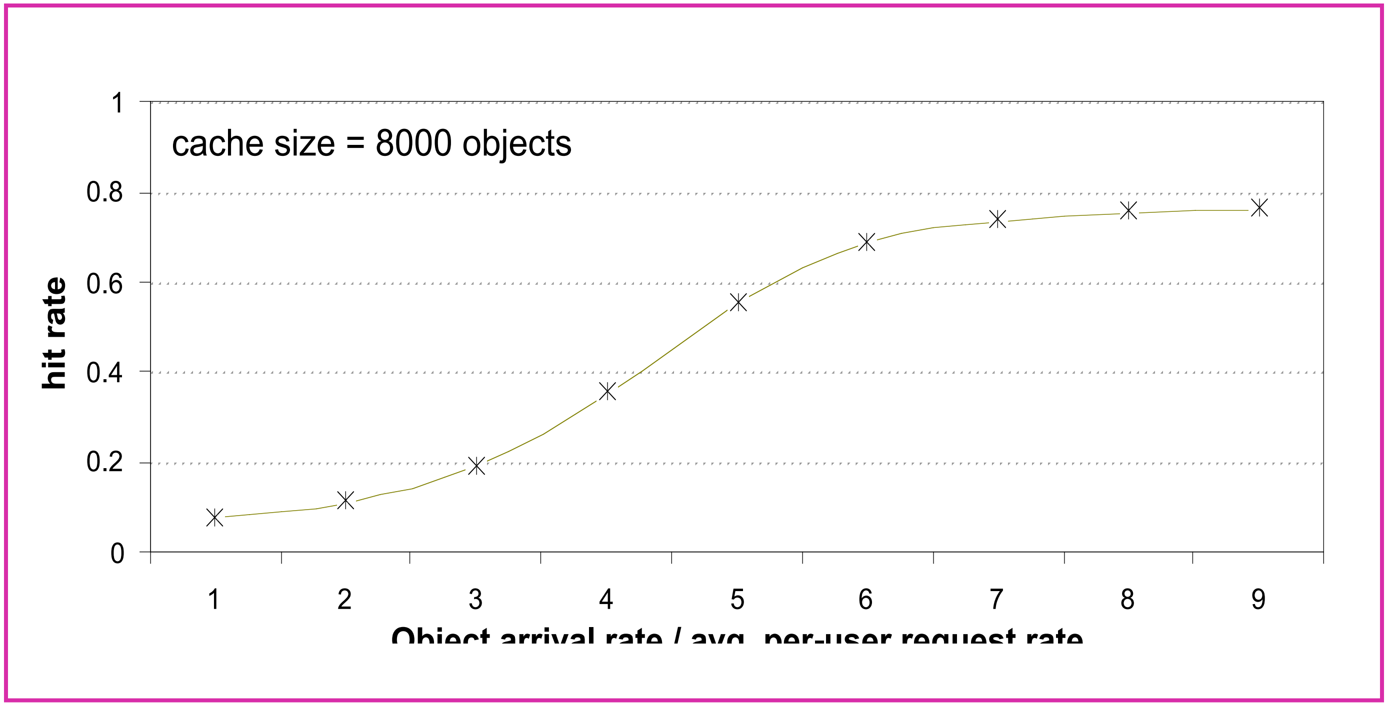
An organization is experiencing too much demand for external bandwidth for P2P applications. How will the demand change if a proxy cache is used? Let us examine the hit ratio of the proxy cache.

Caching implications

- In the absence of new objects and users
  - fetch-many: cache hit rate is stable
  - fetch-at-most-once: hit rate degrades over time



New objects help (not hurt)

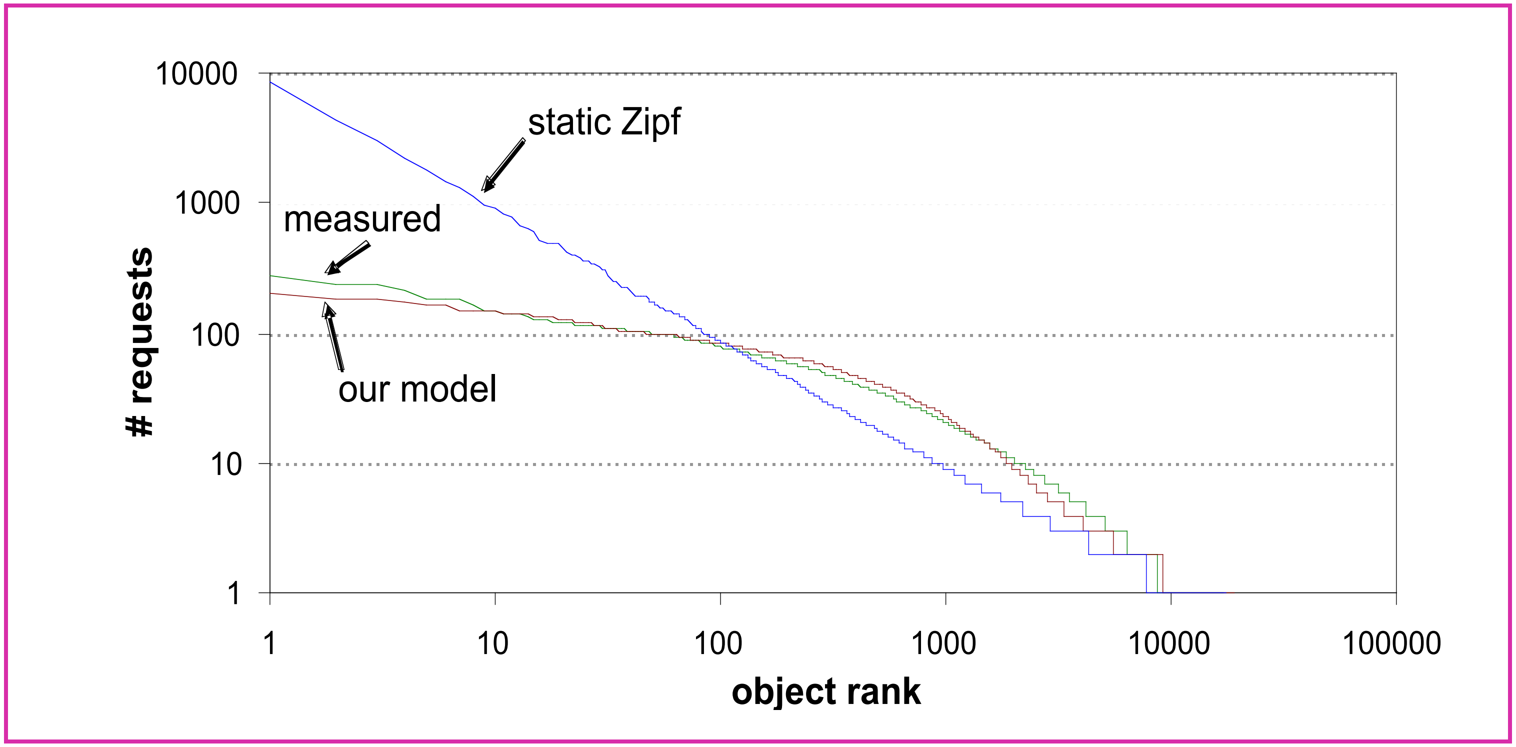


- New objects do cause cold misses
  - but they replenish the supply of popular objects that are the source of file sharing hits
- A slow, constant arrival rate stabilizes performance
  - rate needed is proportional to avg. per-user request rate

New users cannot help

- They have potential…
  - new users have a “fresh” Zipf curve to draw from
  - therefore will have a high initial hit rate
- But the new users grow old too
  - ultimately, they increase the size of the “elderly” population
  - to offset, must add users at exponentially increasing rate
    - not sustainable in the long run

Validating the model

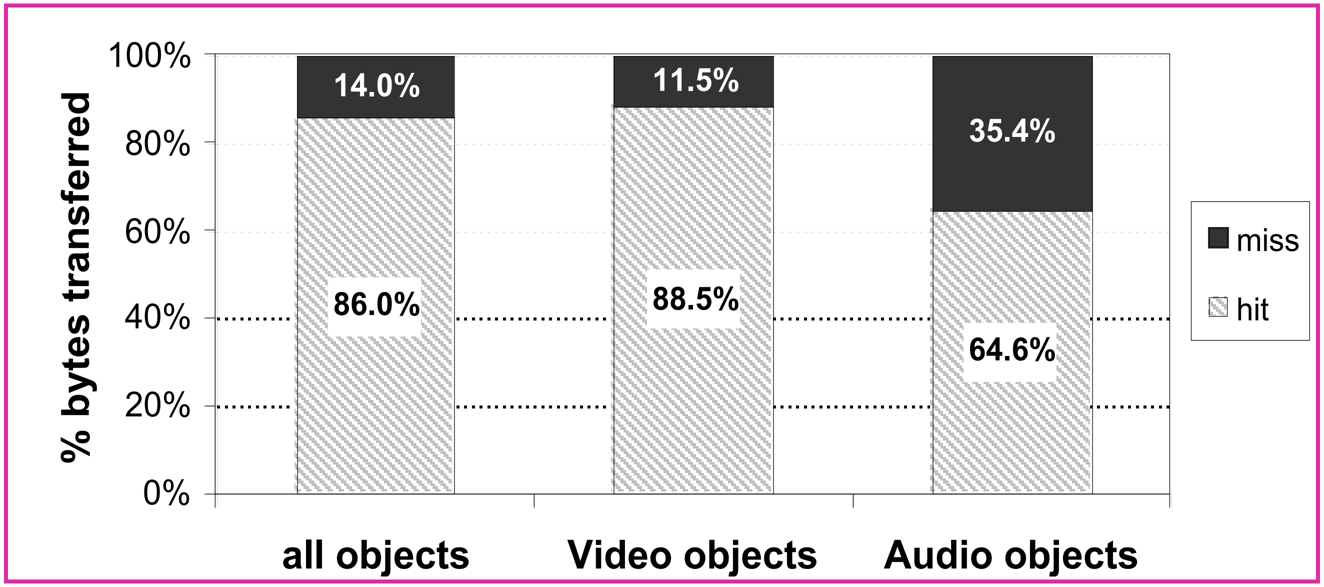


- We parameterized our model using measured trace values
  - its output closely matches the trace itself

Outline

- Introduction
- Some observations about Kazaa
- A model for studying multimedia workloads
- Locality-aware P2P request distribution
- Conclusions

Kazaa has significant untapped locality

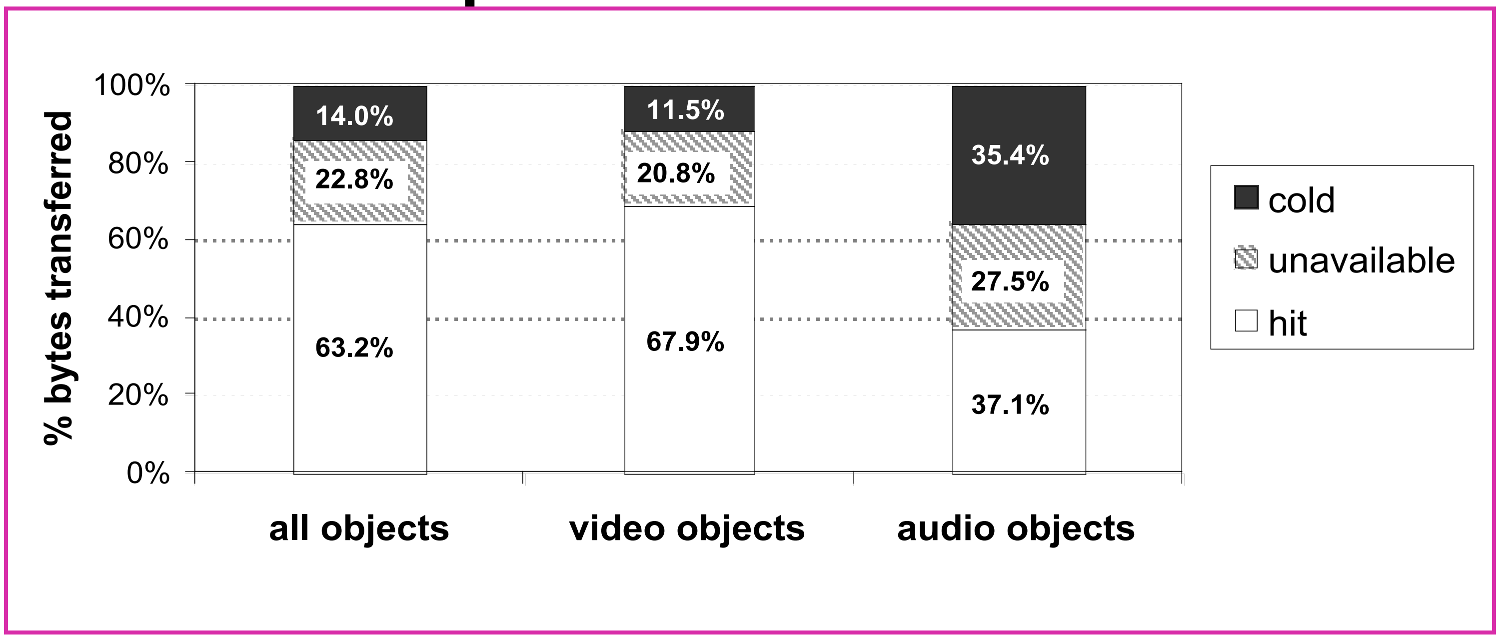


- We simulated a proxy cache for UW P2P environment
  - 86% of Kazaa bytes already exist within UW when they are downloaded externally by a UW peer

Locality Aware Request Routing

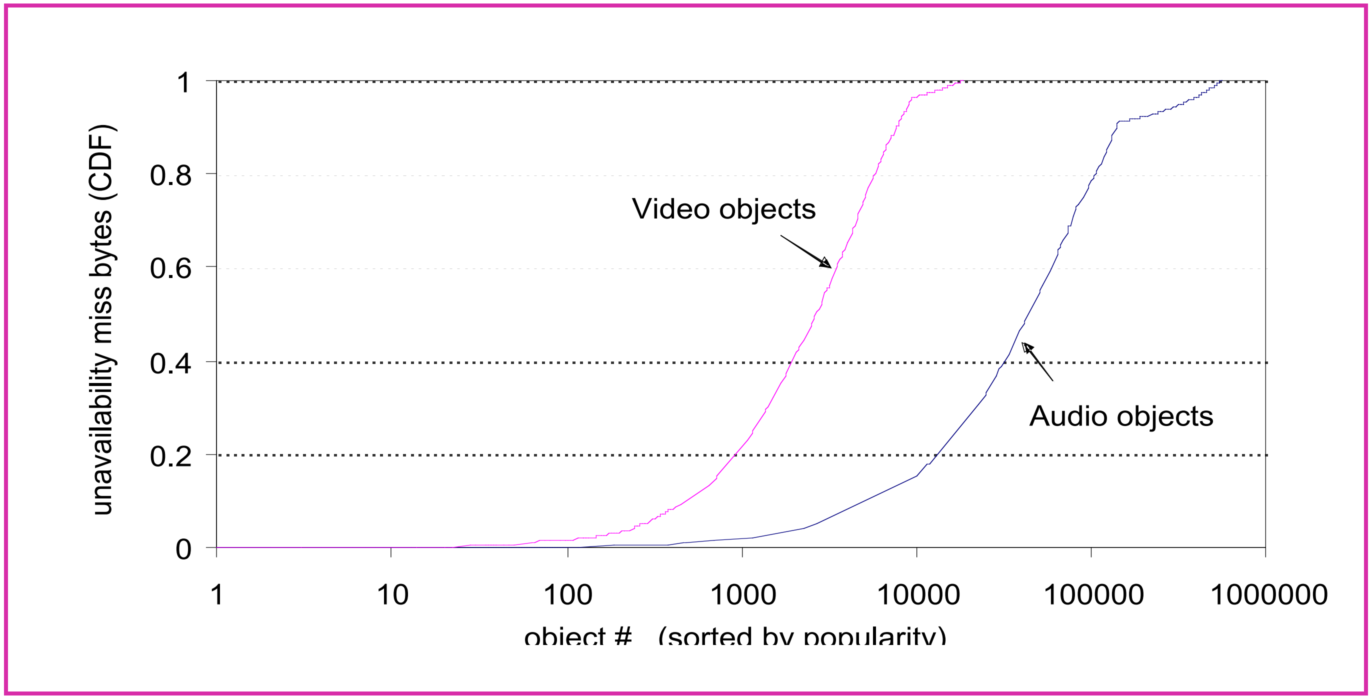
- Idea: download content from local peers, if available
  - local peers as a distributed cache instead of a proxy cache
- Can be implemented in several ways
  - scheme 1: use a redirector instead of a cache
    - redirector sits at organizational border, indexes content, reflects download requests to peers that can serve them
  - scheme 2: decentralized request distribution
    - use location information in P2P protocols (e.g., a DHT)
- We simulated locality-awareness using our trace data
  - note that both schemes are identical w.r.t the simulation

Locality-aware routing performance



- "P2P-ness" introduces a new kind of miss: "unavailable" miss
  - even with pessimistic peer availability, locality-awareness saves significant bandwidth
  - goal of P2P system: minimize the new miss types
    - achieve upper bound imposed by workload (cold misses only)

Eliminating unavailable misses



- Popularity drives a kind of "natural replication"
  - descriptive, but also predictive
    - popular objects take care of themselves, unpopular can't help
    - focus on "middle" popularity objects when designing systems

Conclusions

- P2P file-sharing driven by different forces than the Web
- Multimedia workloads:
  - driven by two factors: fetch-at-most-once, object/user dynamics
  - constructed a model that explains non-zipf behavior and validated it
- P2P infrastructure:
  - current file-sharing architectures miss opportunity
  - locality-aware architectures can save significant bandwidth
  - a challenge for P2P: eliminating unavailable misses