CSC2231: File-Sharing Workloads

http://www.cs.toronto.edu/~stefan/courses/csc2231/05au

Stefan Saroiu

Department of Computer Science

University of Toronto

Administrivia

- **Schedule for the rest of the semester:**
  - Thursday: 1 paper to read only -- Tiger
  - Monday: no lecture!!!
  - Wednesday: project presentations (from 4 to 6:30pm)
  - Thursday: wrap-up lecture/introspection <-- don't miss this
  - Thursday: project write-up due
  - Following 2 weeks after that: food + sleep!
- **Tuesday: December 6th**
  - Student meeting with Stefan Savage

Question

- **One common conclusion after our three-week DHT immersion:**
  - Not clear what applications need P2P/DHTs?
  - Lots of problems: heterogeneity, security, performance
- **Then -- why bother with P2P workloads?**

Motivation for P2P Workloads

Study 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

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)
  - cannot see internal Kazaa traffic
- **No more DHCP/IP aliasing problems!!!**

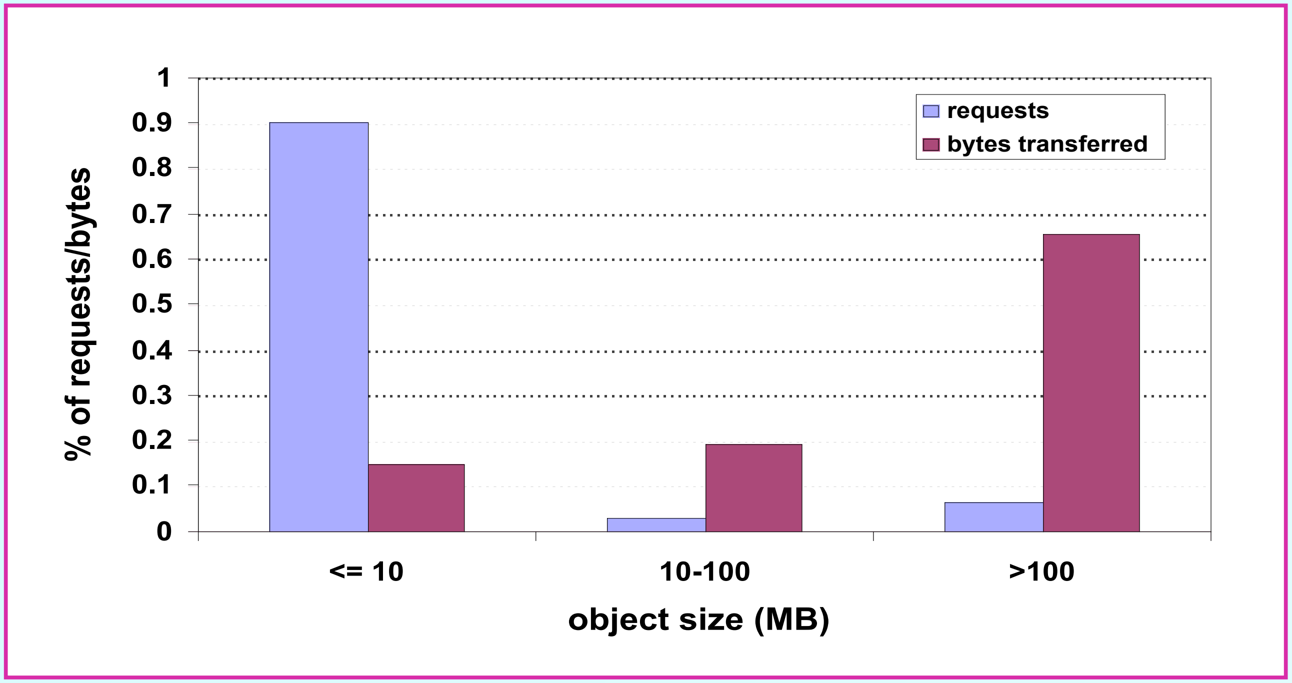
Trace Characteristics

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

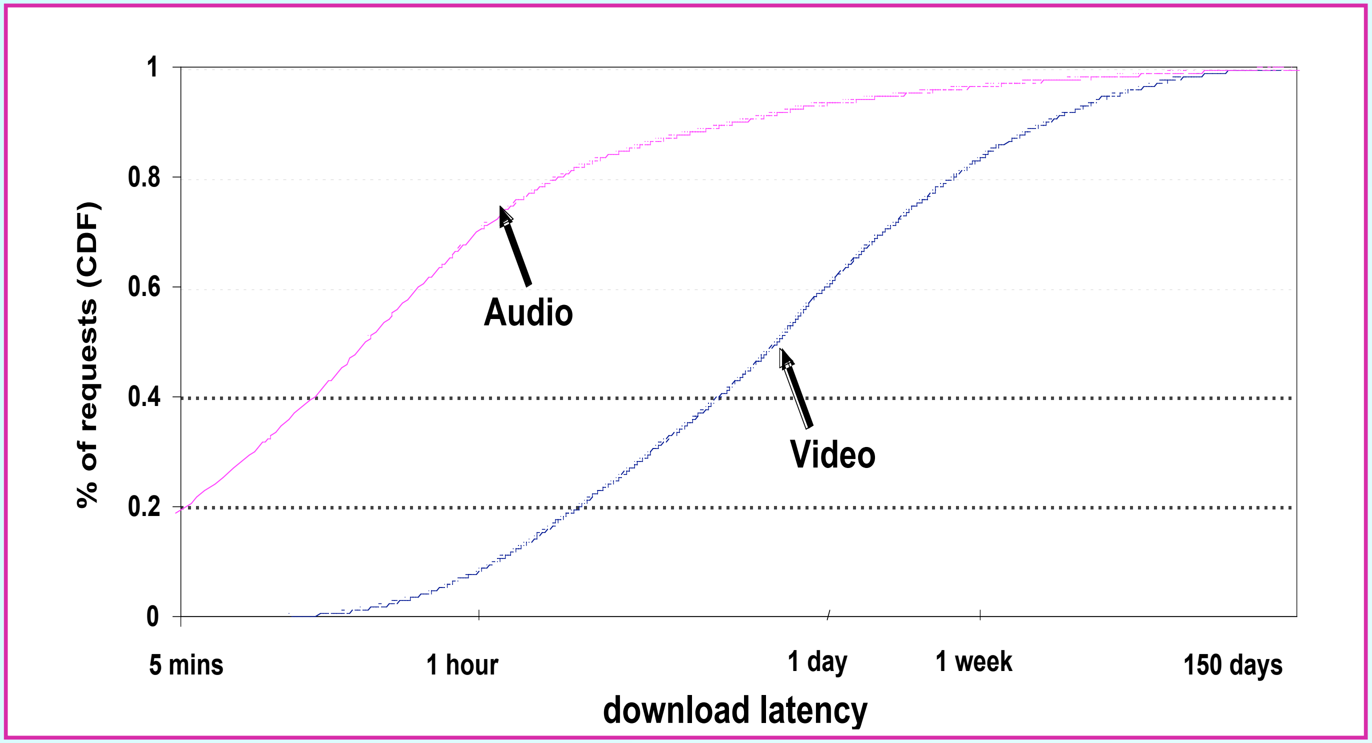
- **Introduction**
- **Some observations about Kazaa**
- **A model for studying multimedia workloads**
- **Locality-aware P2P request distribution**
- **Conclusions**

Kazaa is really 2 workloads



- **If you care about:**
  - making users happy: make sure audio 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 are willing to wait weeks!
  - Kazaa is a batch system, while the Web is interactive

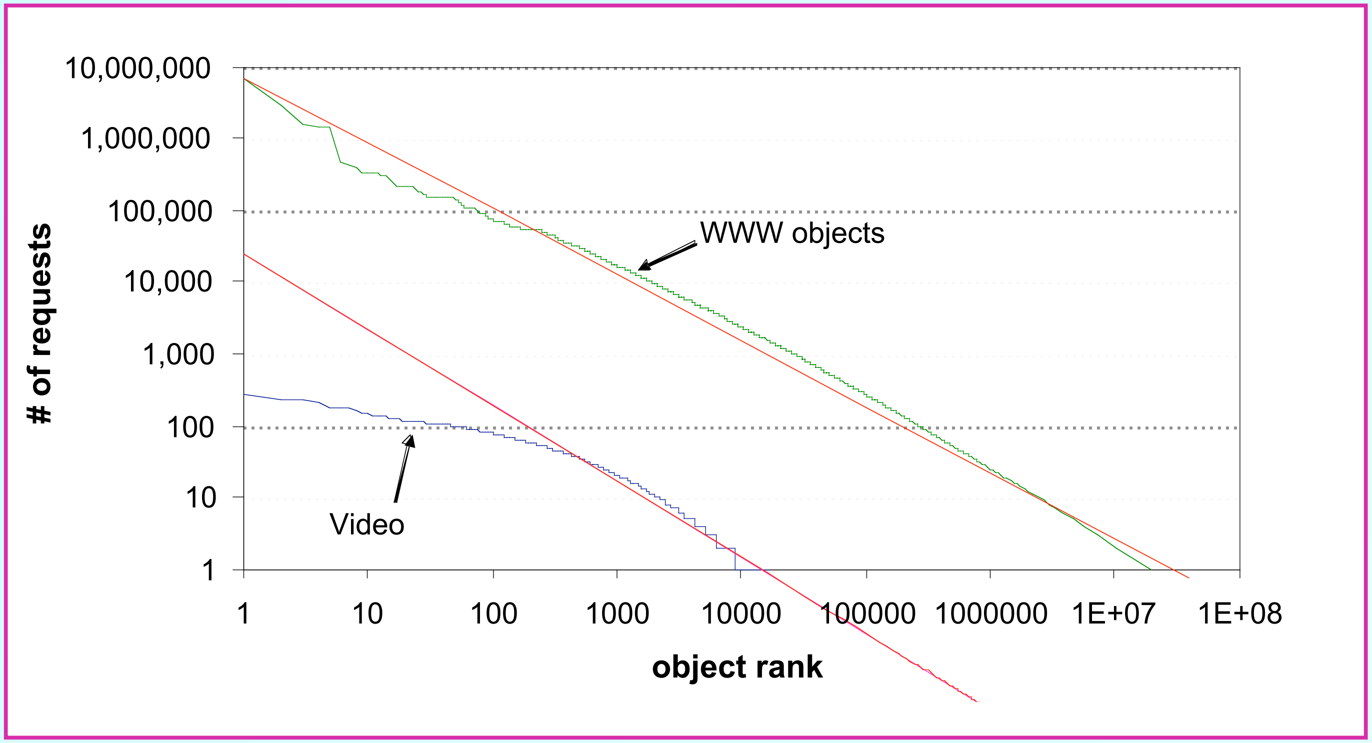
What drives Web workloads?

- **What makes users download a web page?**
  - Why do you go this course's web page over and over?

Kazaa objects are immutable

- **The Web is driven by object change**
  - 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 does not obey Zipf's law



Most popular objects 100x less popular than Zipf predicts

Weight is in the head not in the tail

Factors driving P2P file-sharing workloads

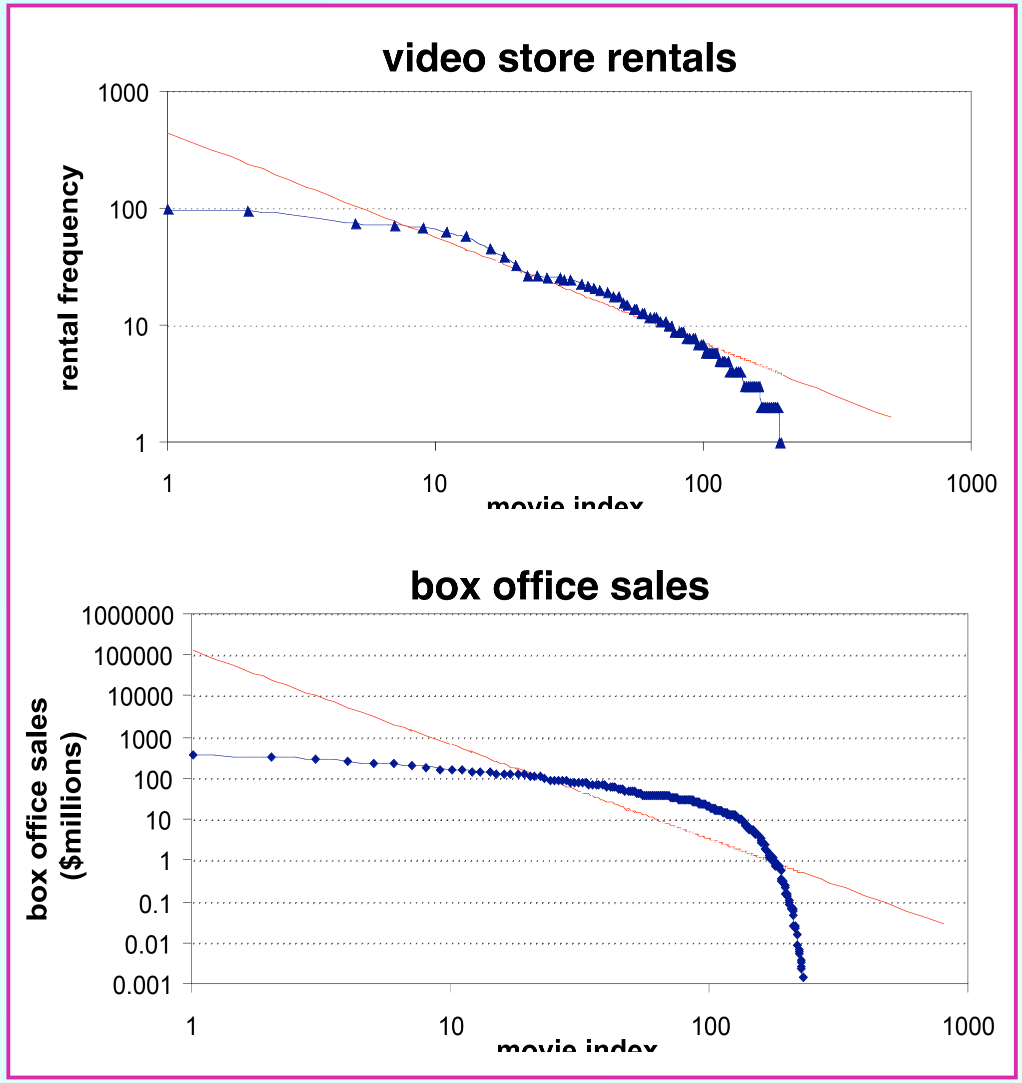
- **Our traces suggest two factors drive P2P workloads:**

  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



Alternative Reasons?

- **Head is flat:**
  - Fetch-at-most-once seems plausible
- **Tail is short:**
  - Not long-enough trace?
  - Broken search in Kazaa: unpopular objects are hard to find
  - The tail is not captured by video store rentals and box office sales
    - Learned that from the first article to review today

Outline

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

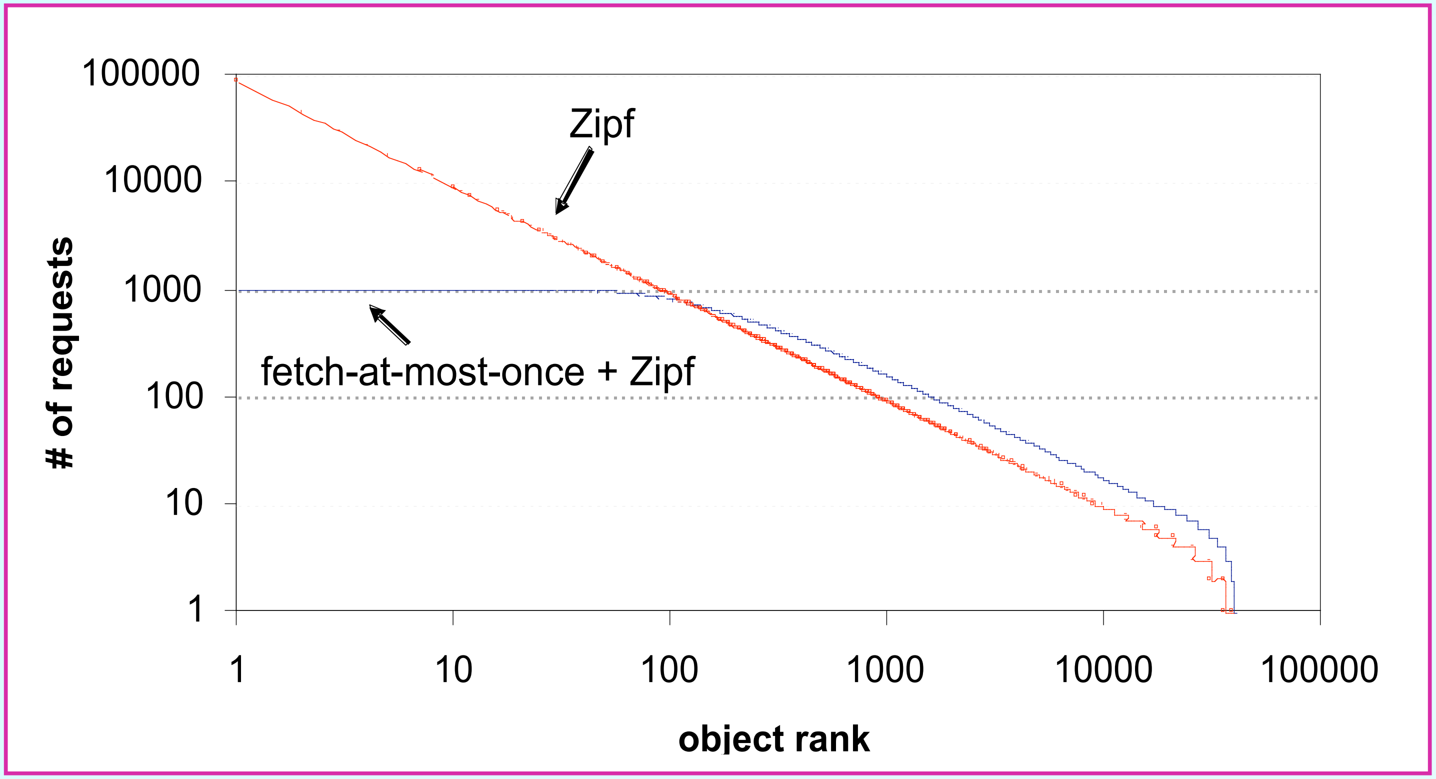
Model basics

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

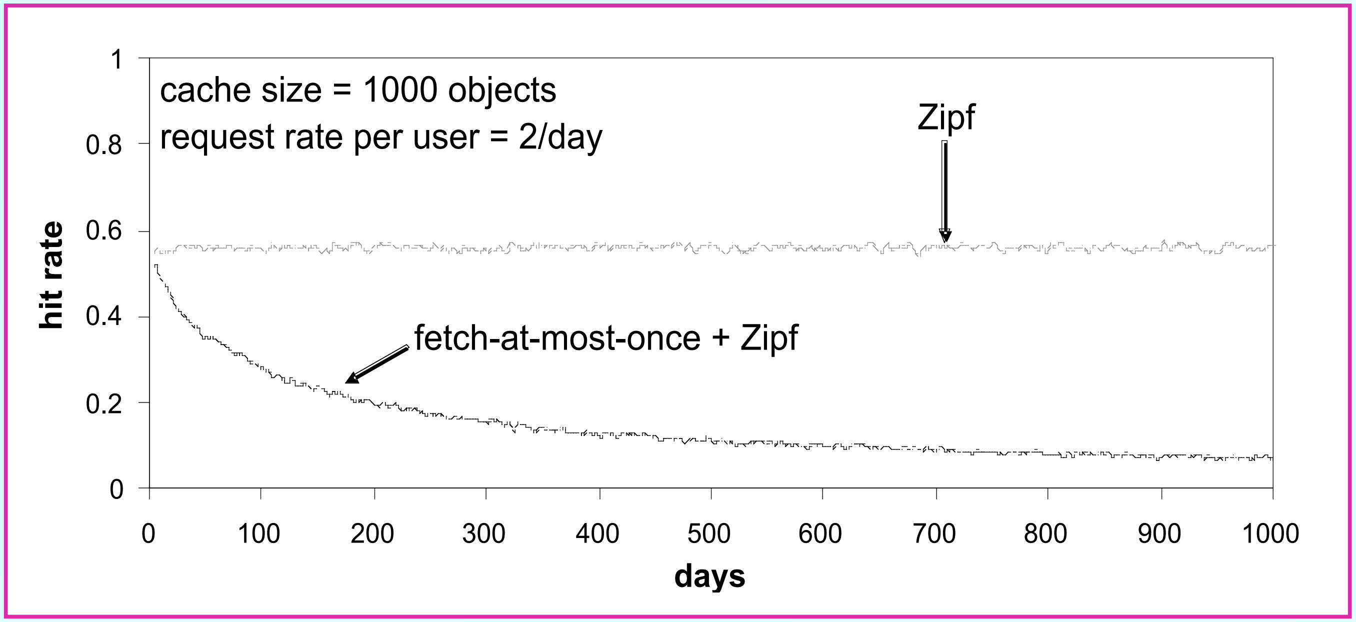
3. **Fold in user, object dynamics**
   - new objects inserted with initial popularity drawn from Zipf
     - new popular objects displace the old popular objects
   - new users begin with a fresh Zipf curve

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

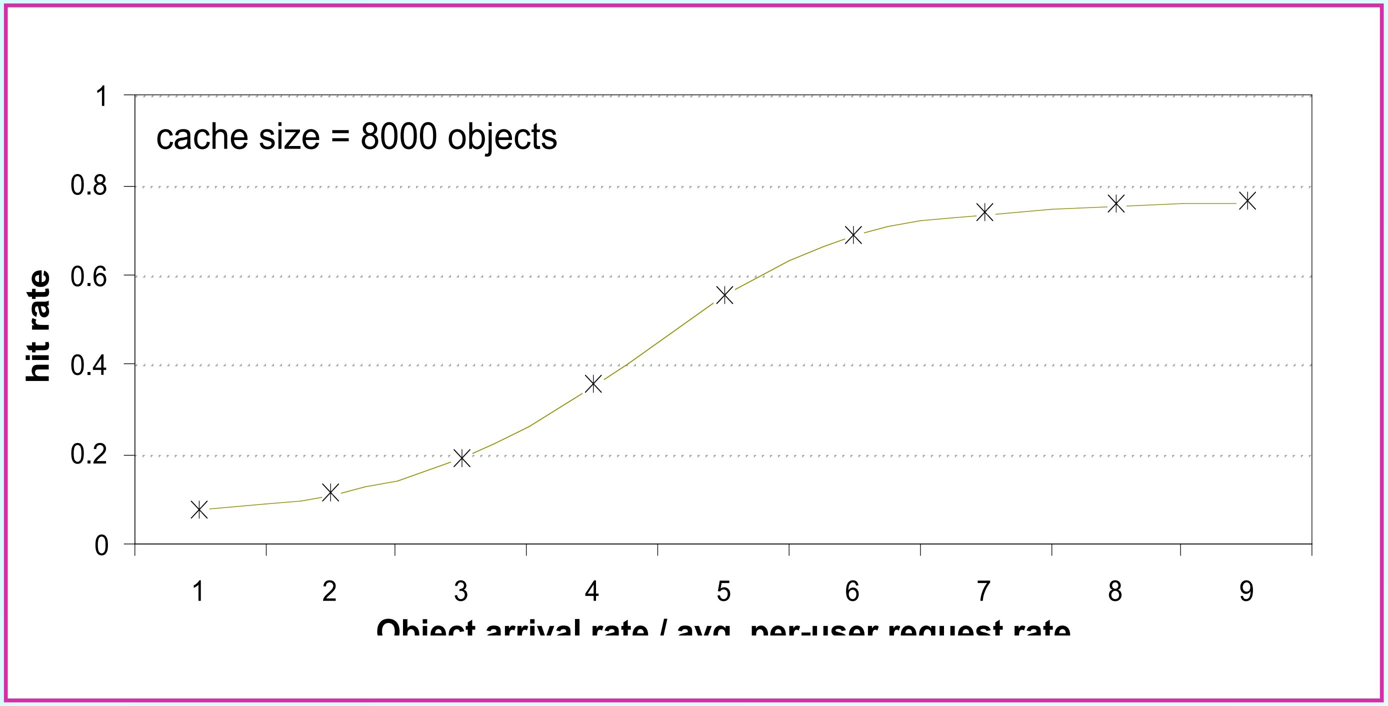


Caching implications

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



New objects help (not hurt)

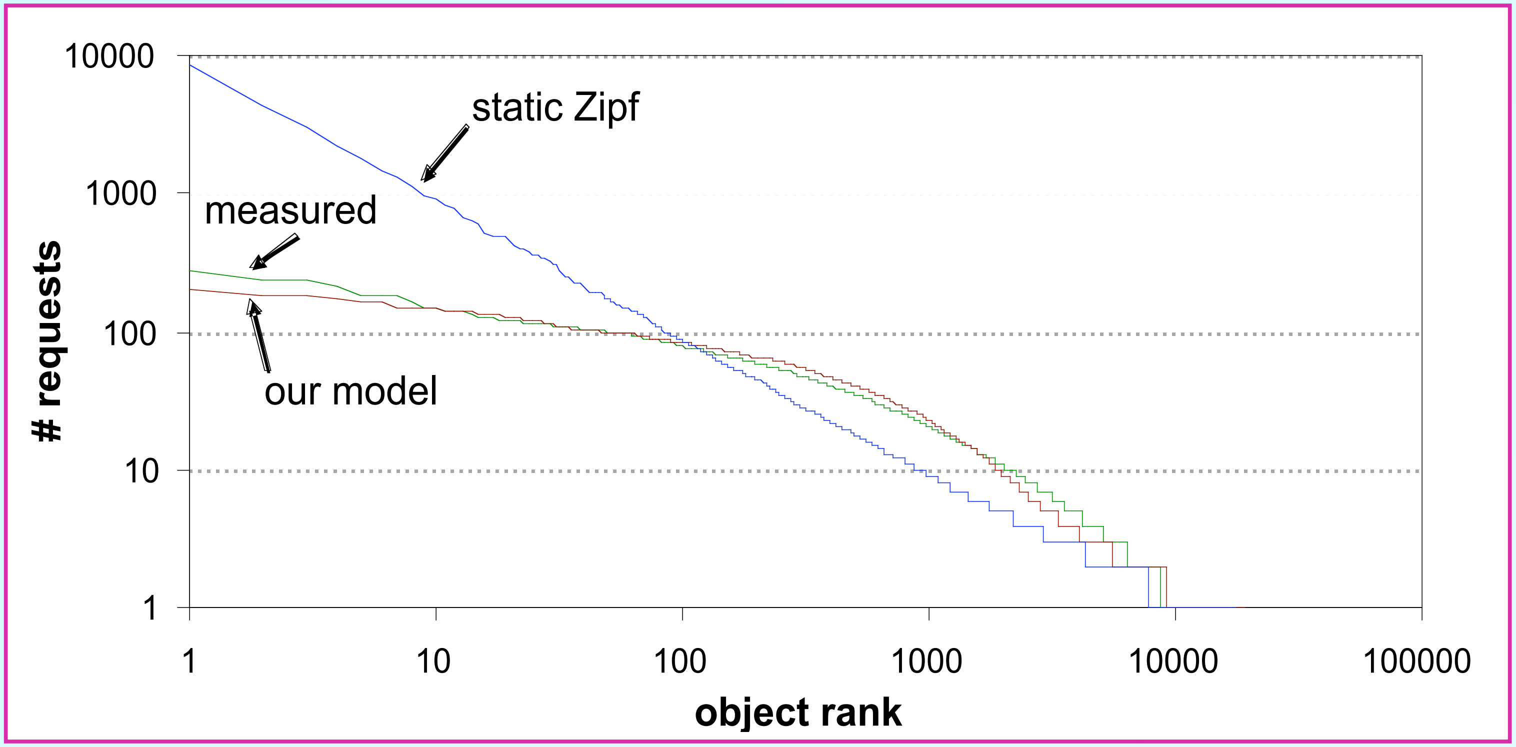


- **New objects do cause cold misses**
  - but they replenish the highly cacheable part of the Zipf curve
- **A slow, constant arrival rate stabilizes performance**
  - rate needed is proportional to avg. per-user request rate

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

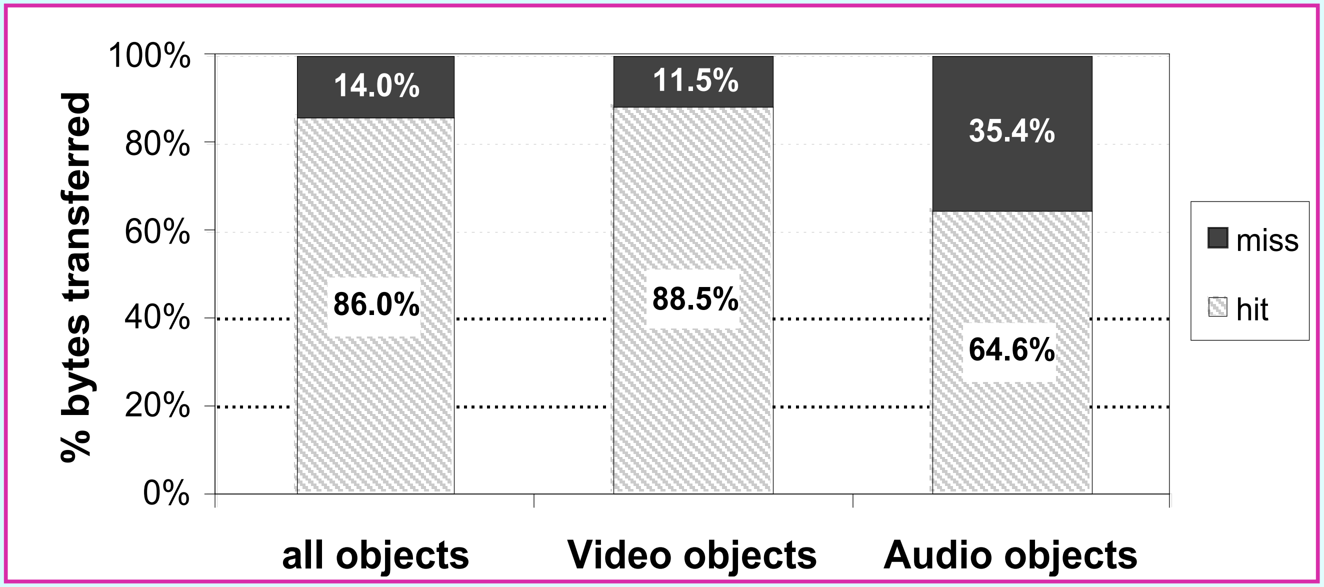


- **We parameterized our model using measured trace values**
  - its output closely matches the trace itself

Outline

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

Kazaa has significant untapped locality

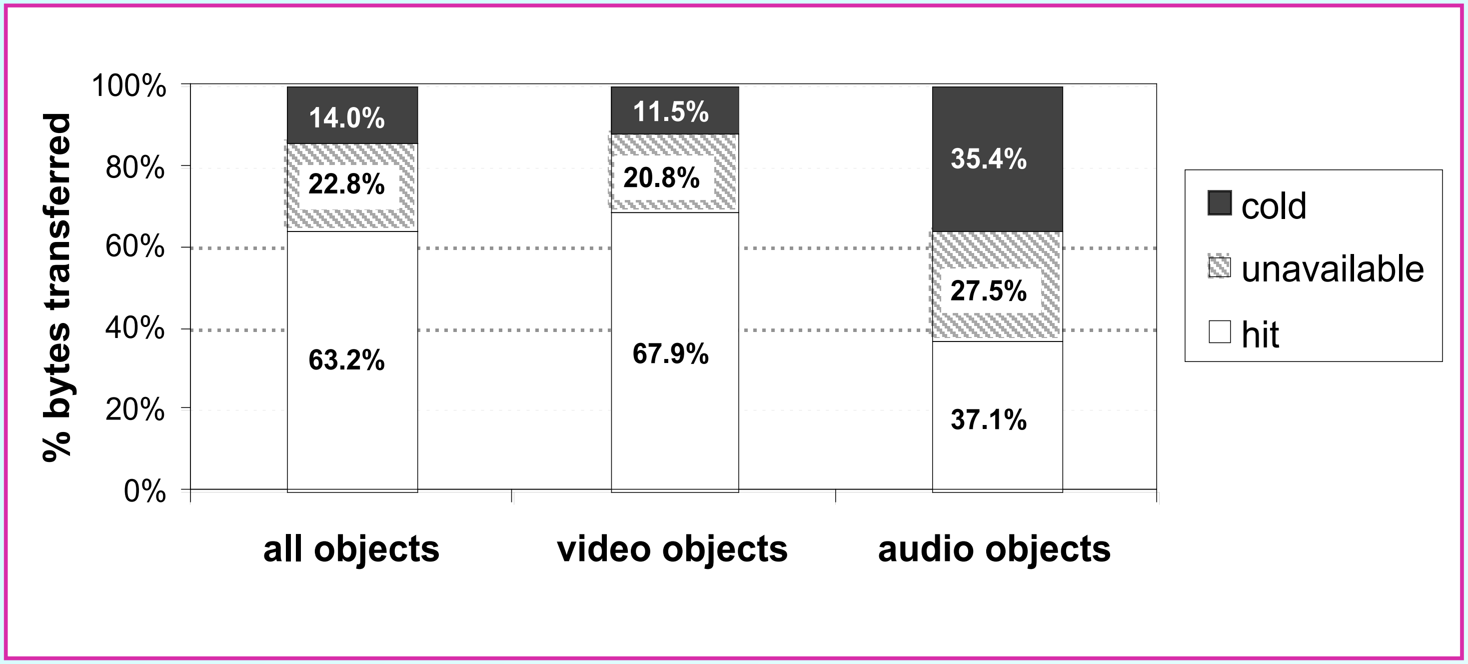


- **We simulated a proxy cache for UW P2P environment**
  - 86% of Kazaa bytes already exist within UW when they are downloaded externally by a UW peer

Locality Aware Request Routing

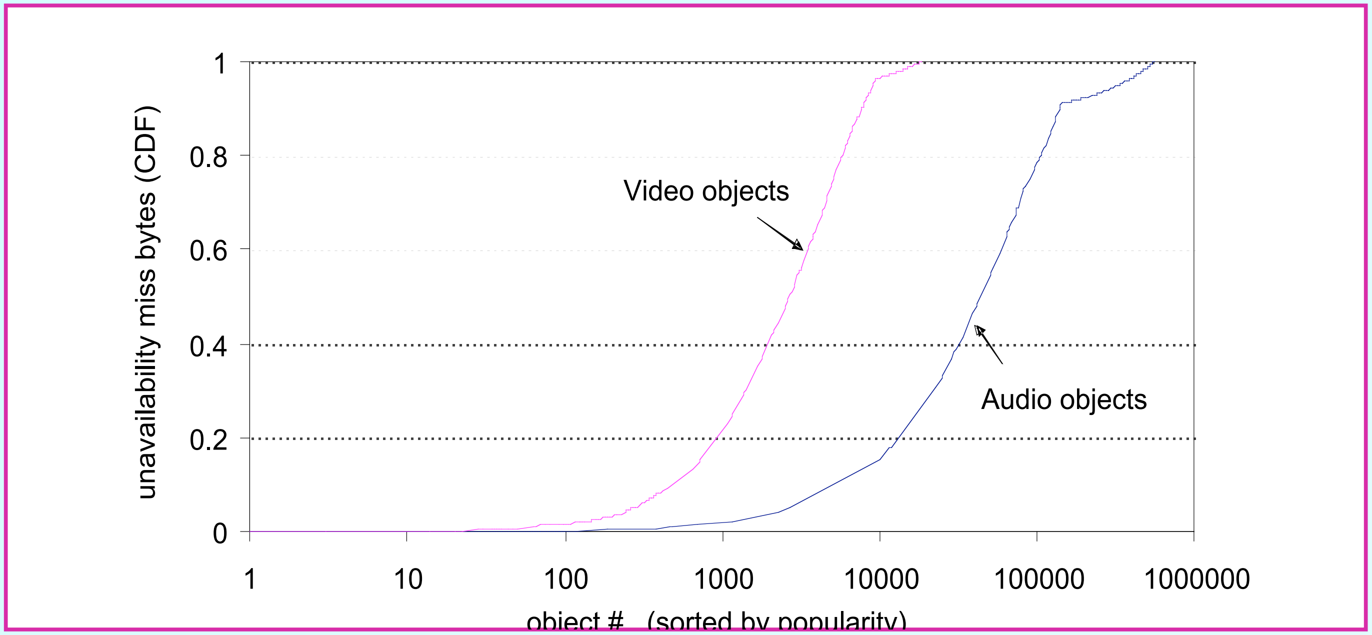
- **Idea: download content from local peers, if available**
  - local peers as a distributed cache instead of a proxy cache
- **Can be implemented in several ways**
  - scheme 1: use a redirector instead of a cache
    - redirector sits at organizational border, indexes content, reflects download requests 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)

How can we eliminate 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 2 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

Discussion

- **If availability is so poor in P2Ps, how could redirection have a 65+% hit rate?**

Discussion

- **If availability is so poor in P2Ps, how could redirection have a 65+% hit rate?**
  - “kernel” of peers with a lot of content and always available
  - It’s the head of the popularity curve that matters, not the tail
- **Implications:**
  - Need to redefine availability to capture these “effects”
  - Need to understand how P2P behaves in the case of the tail