

## Performance of P2P Systems

1

## Outline

- "Performance" article by T. Hong in P2P book
- SOSP articles on Kazaa measurements and analysis
  - Acknowledgements: use slides from Gummadi et al's SOSP talk

2

## Overview

- Performance
  - Communication costs (number of hops per query, bandwidth consumption)
  - Impact of "free riders"
- Fault Tolerance
  - Impact of node failures
    - Random failures
    - Coordinated/Correlated failures (attack scenario)
- Scalability
  - What happens to performance/fault tolerance as network grows

3

## Small World Model

- "It's a small world"
- Milgram's Experiment
  - In 1967, Milgram mailed 160 letters to a set of randomly chosen people in Omaha, Nebraska
  - Goal: pass the letters to a given person in Boston using only intermediaries known to each other on a first-name basis
  - Result: 42 letters made it through!! Median intermediaries was 5.5
- Do P2P systems like Freenet & Gnutella form a "small world"

4

### P2P Networks and the Small World Model

- P2P Network = Graph with edges corresponding to connections between nodes
- Question 1: Are P2P networks *connected graphs*?
- Question 2: What is the *characteristic pathlength* of the graph?
  - Shortest distance between any two nodes averaged over all pairs

5

### Small World Model cont'd

- Watts-Strogatz "Collective Dynamics of Small World Networks", Nature 1998
  - Explanation for Milgram's Results
- Key Observation: Some individuals are "highly connected" and act as a bridge between clusters of individuals
- Even a small number of bridges can dramatically reduce the path length

6

### Graph Theoretical Background

- Regular graph: ring of  $n$  nodes each of which is connected to its  $k$  nearest neighbors
- Random graph: nodes connected at random (avg  $k$  edges per node)
- Metrics
  - Path length (averaged over all pairs)
  - Clustering coefficient: given  $k$  neighbors of a node, the ratio of the number of edges between the nodes to the maximum number of edges  $k(k-1)/2$

7

### Graph Theoretical Background cont'd

- For a regular graph with  $n \gg k$ , it can be shown that avg path length =  $n/2k$ 
  - If  $n = 4096$ ,  $k = 8$ , avg pathlength = 256
- For a regular graph,  $\lim(\text{cluster coeff})$  as  $n$  goes to infinity is 0.75
- For a random graph,  $\lim(\text{cluster coeff}) = k/n = 0$  as  $n$  goes to infinity
- For a random graph, path length =  $\log n / \log k$ 
  - If  $n = 4096$ ,  $k = 8$ , pathlength = 4, clustering coeff = 0.0002

8

### Watts-Strogatz experiment

- ❑ Starting with a 1000 node random graph,  $k = 10$ , for each edge reconnect it to a random vertex with probability  $p$ 
  - If  $p = 0$ , regular graph
  - If  $p = 1$ , random graph
  - What happens if  $0 < p < 1$ ?
- ❑ As  $p$  increases, clustering remains high but path length drops dramatically
- ❑ ***If high clustering and short pathlength, then graph is a small world graph***

9

### Two implications of Watts-Strogatz experiment

- ❑ Only a small amount of "rewiring" is needed for a regular graph to turn into a small world graph
- ❑ The transition is not noticeable at the local level

10

### Freenet

- ❑ Is Freenet network connected?
  - Yes
    - Each node connects to a connected network
    - Redundant links added while processing queries and inserts
  - But what about node failures?
- ❑ Is Freenet a small world network?

11

### Simulation

- ❑ Configuration
  - 1000 identical nodes
  - Capacity of 50 data items + 200 additional references
  - Each node connects to two nodes numerically before and after it
- ❑ Initial characteristics
  - Path length = 125
  - Clustering coefficient = 0.5

12

### Freenet Simulation cont'd

- Experiment 1
  - At each time step, pick a random node and do a random request/insert with hops-to-live = 20
  - Observation: path length and clustering coefficient evolve into a small world network
- Experiment 2
  - Every 100 time steps, simulate 300 requests from randomly selected nodes (hops to live = 500)
  - Observations
    - Median path length drops from 500 to 6
    - Still some requests can take a long time

13

### Freenet simulation cont'd

- Experiment 3
  - What is the impact of Freenet routing on median path length?
  - If random routing used, median pathlength is around 50
- Experiment 4:
  - Simulating growth
    - Start with 20 nodes, add a new node every 5 time steps until the network has 1000 nodes
    - Connect new node to a random existing node, send announcement with hops to live = 10
    - Insert requests, probes as in earlier experiments
  - Observations: network evolves into a small world network
    - Characteristic pathlength = 2.2, Clustering coefficient = 0.25, median request path length = 5

14

### Freenet simulation: Fault tolerance

- Experiment 1: Remove nodes at random
  - Observation: Median pathlength below 20 when up to 30% of the nodes fail
- Experiment 2: Remove most connected nodes first
  - Observation: Median pathlength > 20 at 18% failure level

15

### Link distribution in Freenet

- Link distribution in Freenet is scale-free
    - $\log p = -k \log L + b$
    - where  $p$  = fraction of nodes and  $L$  = number of links per node
    - $p = A L^{-k}$
- Relationship between  $p$  and  $L$  does not depend on  $N$  (number of nodes in the network)
- Small world networks have been shown to have scale-free link distributions

16

## Other Observations

- Impact of Freeriders
  - Freenet ignores freeriders because if node does not provide files, no nodes will have references to it
  - No impact on path length
  - However, requests will add to the bandwidth load
- Scalability
  - In small world graphs, characteristic path length follows random graph properties, i.e. it is  $\log n / \log k$

17

## Gnutella

- Queries are broadcast, so no small world effect
- But we can examine path length, link distribution, etc as in Freenet simulation
- Gnutella network modeled as a random graph with  $k = 3$
- Similar experiments as Freenet simulation
  - 1000 nodes, 1500 edges ( $k = 3$ ), 2500 data items, 300 queries .....

18

## Simulation Observations

- Query performance
  - Query pathlength = characteristic pathlength
  - BFS leads to optimal paths and better worstcase performance than Freenet
  - Number of nodes contacted per query much larger than Freenet
- Fault tolerance
  - Number of highly connected links not a factor in Gnutella
  - Targeted attack scenario: Gnutella does better
  - Random attack scenario: Freenet does better
- Gnutella vulnerable to free riders because a node cannot distinguish a free rider from other nodes
- Scalability: characteristic pathlength scales logarithmically but bandwidth usage scales linearly

19

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

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

Department of Computer Science and Engineering  
University of Washington  
Seattle, WA

20

## The Internet has changed (again!)

- 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

21

## This talk

- 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

22

## 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

23

## Methodology

- Capture a 6-month long trace of Kazaa traffic at UW
  - trace gathered from May 28<sup>th</sup> – December 17<sup>th</sup>, 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

24

## Trace Characteristics

start date	May 28 <sup>th</sup> , 2002
end date	December 17 <sup>th</sup> , 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

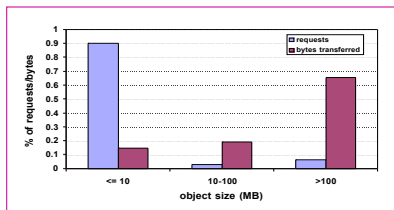
25

## Outline

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

26

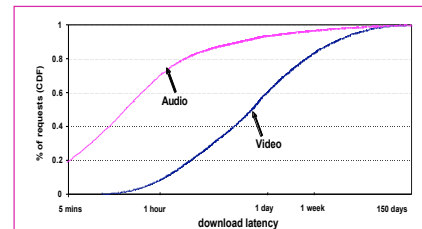
## Kazaa is really 2 workloads



- If you care about:
  - > making users happy: make sure audio arrives quickly
  - > making IT dept. happy: cache or rate limit video

27

## 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**

28

### 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

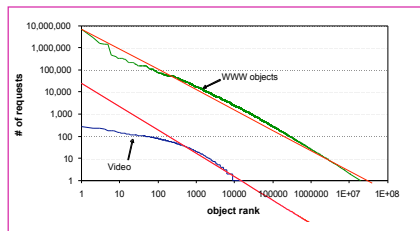
29

### Kazaa popularity has high turnover

- Popularity is short lived
  - 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%]

30

### Kazaa does not obey Zipf's law



- Zipf: popularity( $n^{\text{th}}$  most popular object)  $\sim 1/n^{\alpha}$
- Kazaa: the most popular objects are 100x less popular than Zipf predicts

31

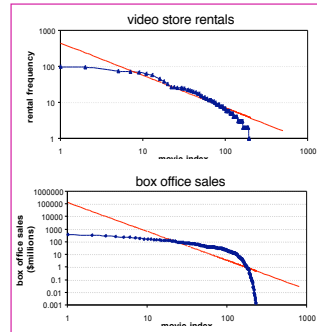
### Factors driving P2P file-sharing workloads

- 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

32

## It's not just Kazaa

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



## Outline

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

34

## 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

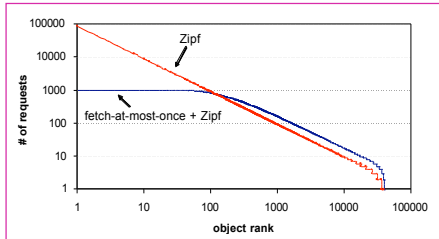
35

## Model parameters

C	# of clients	1,000
O	# of objects	40,000
-R	client req. rate	2 objs/day
-	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
-o	object arrival rate	varies
-c	client arrival rate	varies

36

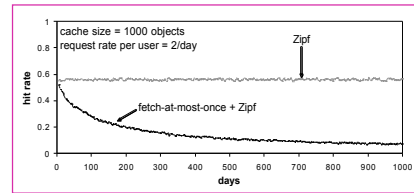
## Fetch-at-most-once flattens Zipf's head



37

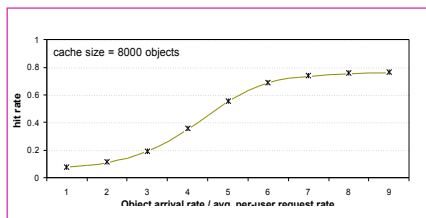
## Caching implications

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



38

## 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

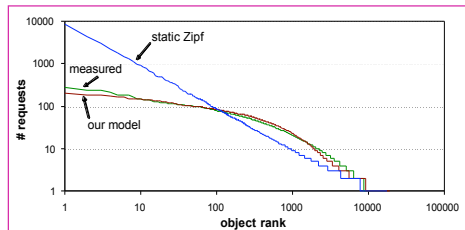
39

## 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

40

## Validating the model



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

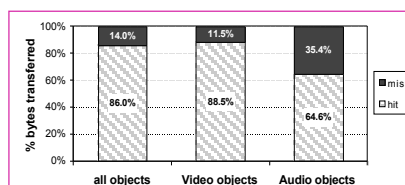
41

## Outline

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

42

## 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

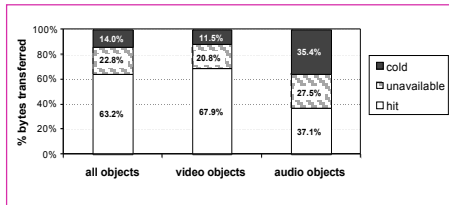
43

## 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

44

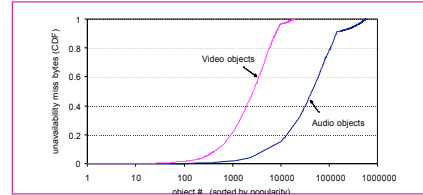
## Locality-aware routing performance



- "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)

45

## 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

46

## 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

47