CS 525
Advanced Distributed Systems
Spring 2010

Indranil Gupta (Indy)

Measurement Studies

April 1, 2010

Acknowledgments: Long Vu, Jay Patel

How do you find characteristics of these Systems in Real-life Settings?

- Write a crawler to crawl a real working system
- Collect *traces* from the crawler
- Tabulate the results

- Papers contain plenty of information on how data was collected, the caveats, ifs and buts of the interpretation, etc.
  - These are important, but we will ignore them for this lecture and concentrate on the raw data and conclusions

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

Gummadi et al

Department of Computer Science

University of Washington

What They Did

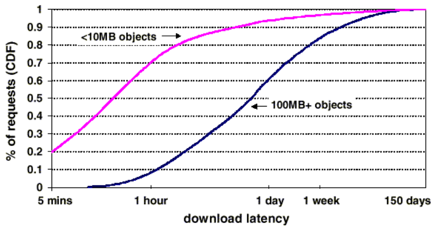
- 2003 paper analyzed 200-day trace of Kazaa traffic
- Considered only traffic going from U. Washington to the outside
- Developed a model of multimedia workloads

Results Summary

1. Users are patient
2. Users slow down as they age
3. Kazaa is not one workload
4. Kazaa clients fetch objects at-most-once
5. Popularity of objects is often short-lived
6. Kazaa is not Zipf

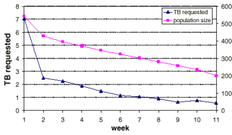
User characteristics (1)

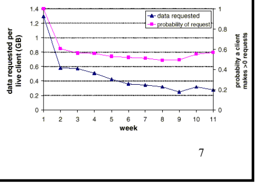
- Users are patient



User characteristics (2)

- Users slow down as they age
  - – clients “die”
  - – older clients ask for less each time they use system



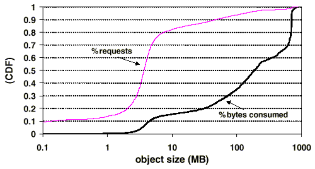


User characteristics (3)

- Client activity
  - – Tracing used could only detect users when their clients transfer data
  - – Thus, they only report statistics on client activity, which is a *lower bound* on availability
  - – Avg session lengths are typically small (median: 2.4 mins)
    - Many transactions fail
    - Periods of inactivity may occur during a request if client cannot find an available peer with the object

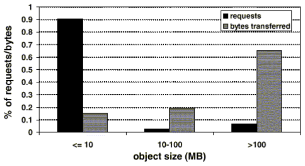
Object characteristics (1)

- Kazaa is not one workload



(a)

•This does not account for connection overhead

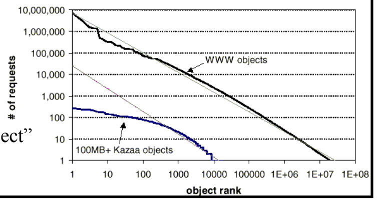


Object characteristics (2)

- Kazaa object dynamics
  - – Kazaa clients fetch objects **at most once**
  - – Popularity of objects is often short-lived
  - – Most popular objects tend to be recently-born objects
  - – Most requests are for old objects (> 1 month)
    - 72% old – 28% new for large objects
    - 52% old – 48% new for small objects

Object characteristics (3)

- Kazaa is not Zipf
- Zipf’s law: popularity of *i*th-most popular object is proportional to $i^{-\alpha}$, (α: Zipf coefficient)
- Web access patterns are Zipf
- Authors conclude that Kazaa is not Zipf because of the at-most-once fetch characteristics



Caveat: what is an “object” in Kazaa?

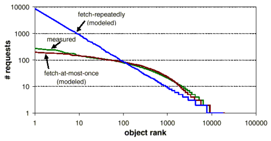
Model of P2P file-sharing workloads

[?] Why a model?

- On average, a client requests 2 objects/day
- P(x): probability that a user requests an object of popularity rank x → Zipf(1)
  - – Adjusted so that objects are requested at most once
- A(x): probability that a newly arrived object is inserted at popularity rank x → Zipf(1)
- All objects are assumed to have same size
- Use caching to observe performance changes (effectiveness → hit rate)

Model – Simulation results

- File-sharing effectiveness diminishes with client age
  - System evolves towards one with no locality and objects chosen at random from large space
- New object arrivals improve performance
  - Arrivals replenish supply of popular objects
- New clients cannot stabilize performance
  - Cannot compensate for increasing number of old clients
  - Overall bandwidth increases in proportion to population size
- By tweaking the arrival rate of of new objects, were able to match trace results (with 5475 new arrivals per year)



Some Questions for You

- "Unique object" : When do we say two objects A and B are "different"?
  - When they have different file names
    - fogonthetyne.mp3 and fogtyne.mp3
  - When they have exactly same content
    - 2 mp3 copies of same song, one at 64 kbps and the other at 128 kbps
  - When A (and not B) is returned by a keyword search, and vice versa
  - …?
- Based on this, does "caching" have a limit? Should caching look *into* file content? Is there a limit to such intelligent caching then?
- Should there be separate overlays for small objects and large objects? For new objects and old objects?
- Or should there be separate caching strategies?
- Most requests for old objects, while most popular objects are new ones – is there a contradiction?

Understanding Availability

R. Bhagwan, S. Savage, G. Voelker
University of California, San Diego

What They Did

- Measurement study of peer-to-peer (P2P) file sharing application
  - Overnet (January 2003)
  - Based on Kademlia, a DHT based on xor routing metric
    - Each node uses a random self-generated ID
    - The ID remains constant (unlike IP address)
    - Used to collect availability traces
  - Closed-source
- Analyze collected data to analyze availability
- Availability = % of time a node is online (node=user, or machine)

What They Did

- Crawler:
  - Takes a snapshot of all the active hosts by repeatedly requesting 50 randomly generated IDs.
  - The requests lead to discovery of some hosts (through routing requests), which are sent the same 50 IDs, and the process is repeated.
  - Run once every 4 hours to minimize impact
- Prober:
  - Probe the list of available IDs to check for availability
    - By sending a request to ID *I*; request succeeds only if *I* replies
    - Does not use TCP, avoids problems with NAT and DHCP
  - Used on only randomly selected 2400 hosts from the initial list
  - Run every 20 minutes
- All Crawler and Prober trace data from this study is available for your project (ask Indy if you want access)

Scale of Data

- Ran for 15 days from January 14 to January 28 (with problems on January 21) 2003
- Each pass of crawler yielded 40,000 hosts.
- In a single day (6 crawls) yielded between 70,000 and 90,000 unique hosts.
- 1468 of the 2400 randomly selected hosts probes responded at least once

Results Summary

1. Overall availability is low
2. Diurnal patterns existing in availability
3. Availabilities are uncorrelated across nodes
4. High Churn exists

Multiple IP Hosts

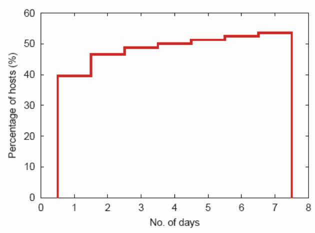


Figure 1: Percentage of hosts that have more than one IP address across different periods of time.

Availability

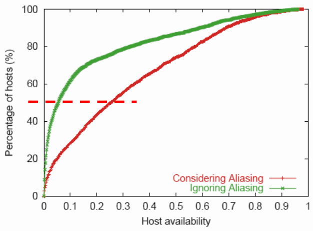
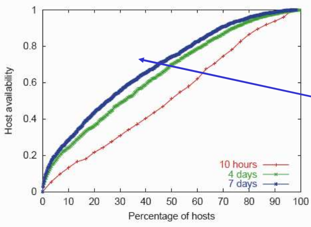


Figure 2: Host availability derived using unique host ID probes vs. IP address probes.

Host Availability

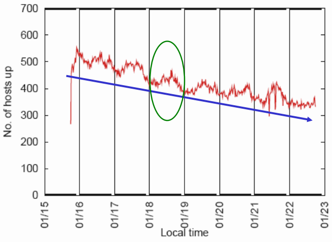


As time interval increased, av. decreases

Figure 3: The dynamic nature of the availability distribution. It varies with the time period over which availability is calculated.

Diurnal Patterns

•Normalized to "local time" at peer, *not* EST



•N changes by only 100/day

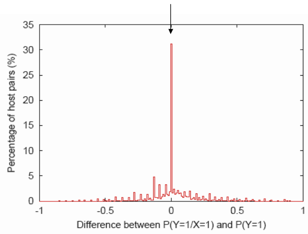
•6.4 joins/host/day

•32 hosts/day lost

Figure 4: Diurnal patterns in number of available hosts.

Are Node Failures Interdependent?

30% with 0 difference, 80% within +-0.2



Should be same if X and Y independent

Figure 5: Probability density function of the difference between P(Y=1/X=1) and P(Y=1).

Arrival and Departure

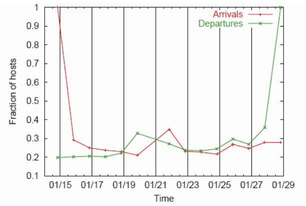


Figure 6: New host arrivals and existing host departures in Overnet as a fraction of all hosts in the system (approximately 85,000 during this period). The high values at the beginning and end of the period are artifacts of starting and ending the trace.

- •20% of nodes each day are new
- •Number of nodes stays about 85,000

Conclusions and *Discussion*

- Each host uses an average 4 different IP addresses within just 15 days
  - *Keeping track of assumptions is important for trace collection studies*
- Strong diurnal patterns
  - *Design p2p systems that are adaptive to time-of-day?*
- Value of N stable in spite of churn
  - *Can leverage this in building estimation protocols, etc., for p2p systems.*

Measurement and Modeling of a Large-scale Overlay for Multimedia Streaming

Long Vu, Indranil Gupta, Jin Liang, Klara Nahrstedt
UIUC

This was a CS525 Project (Spring 2006).
Published in QShine 2007 conference, and ACM TOMCCAP.

Motivation

- IPTV applications have flourished (SopCast, PPLive, PeerCast, CoolStreaming, TVUPlayer, etc.)
- IPTV growth: (*MRG Inc. April 2007)*
  - Subscriptions: 14.3 million in 2007, 63.6 million in 2011.
  - Revenue: $3.6 billion in 2007, $20.3 billion in 2011
- Largest IPTV in the world today are P2P streaming systems
- A few years ago, this system was PPLive: 500K users at peak, multiple channels and per-channel overlay, nodes may be recruited as relays for other channels. (Data from 2006)
- **Do peer to peer IPTV systems have the same overlay characteristics as peer to peer file-sharing systems?**

Summary of Results

P2P Streaming overlays are different from File-sharing P2P overlays in a few ways:

1. Users are **im**patient: Session times are small, and exponentially distributed (think of TV channel flipping!)
2. Smaller overlays are random (and not power-law or clustered)
3. Availability is highly correlated across nodes within same channel
4. Channel population varies by 9x over a day.

Results

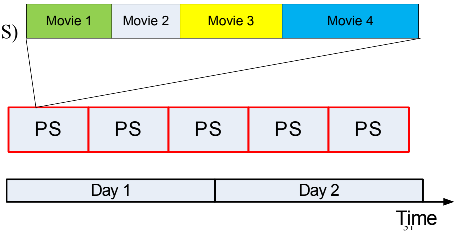
	PPLive	**P2P File Sharing**
Channel Size	Varied over time and channel content	Stable
Node Degree	Scale-free	Scale-free
Overlay Randomness	- Small overlay, more random - Large overlay, more clustered	Small-world
Node Availabiltiy	- Nodes in one snapshot are correlated - Random nodes are independent	Independent
Node Session Length	- Short (Impatient) - Session lengths are Geometric series	Long (Patient)

PPLive Channels

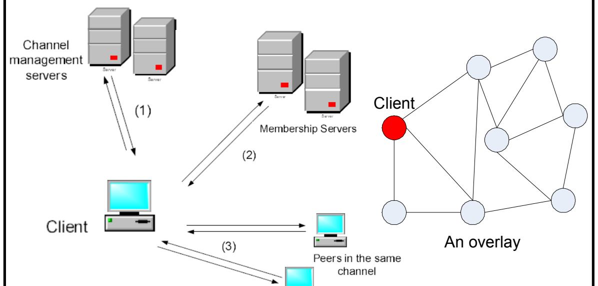
Catalog Name	Number of channels
TV	52
Information	29
Sports	1
PhonenixTV	5
Movies	79
Teleplay	66
Entertainment	68
Cartoon	30
Game	28
Others	52
Summary	410

A Program Segment (PS)

An episode channel



PPLive Membership Protocol

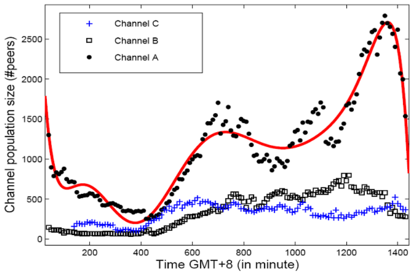


Challenges

PPLive is a closed source system:

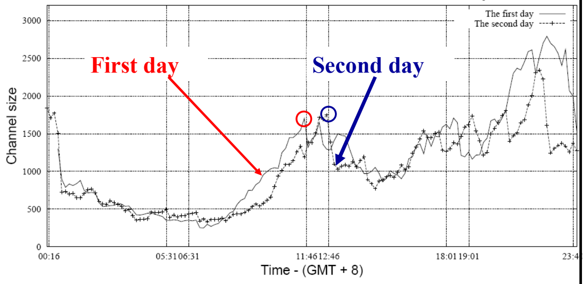
Makes measurement challenging – have to select metrics carefully!

Channel Size Varies over a day



- Use 10 PlanetLab geographically distributed nodes to crawl peers
- Popular channel varies 9x, less popular channel varies 2x

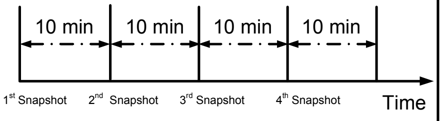
Channel Size Varies over Days



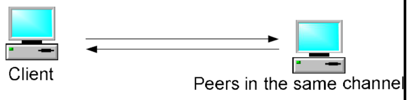
The same channel, same program: Peaks drift

Operations

Snapshot collects peers in one channel



PartnerDiscovery collects partners of responsive peers

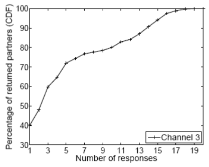


Studied channels

Name	Channel Population in 24 hours	Type	Program Segment
A	35K-45K	Movie	6h15m
B	8K-12K	Cartoon	4d4h
C	10K-15K	Cartoon	1d2h16m

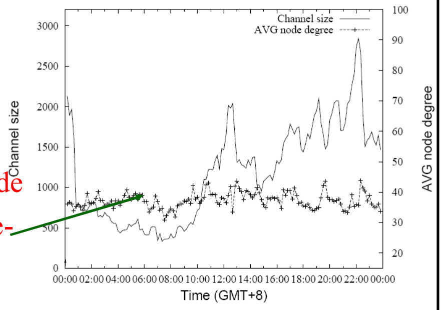
K-degree

- Problem: When PPLive node is queried for membership list, it sends back a fixed size list.
  - Subsequent queries return slightly different lists
- One option: figure out why
  - Lists changing?
  - Answers random?
  - …
- Our option: define
  - K-degree = Union of answers received when K consecutive membership queries are sent to the PPLive node
- K=5-10 gives half of entries as K=20



(b) k response degree

Node Degree is Independent of Channel Size



Average node degree scale-free

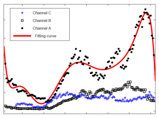
Similar to P2P file sharing [*Ripeanu 02*]

Overlay Randomness

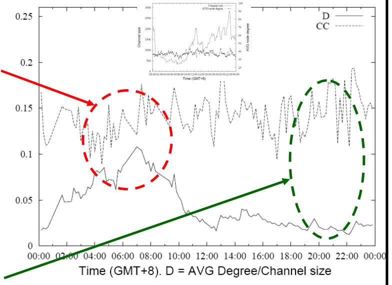
- Clustering Coefficient (CC) [Watts 98]
  - for a random node *x* with two neighbors *y* and *z*, the CC is the probability that either *y* is a neighbor of *z* or vice versa
- Probability that two random nodes are neighbors (D)
  - Average degree of node / channel size
- Graph is more clustered if CC is far from D [*well-known results theory of networks and graphs*]

Smaller Overlay, More Random

- Small overlay, more random

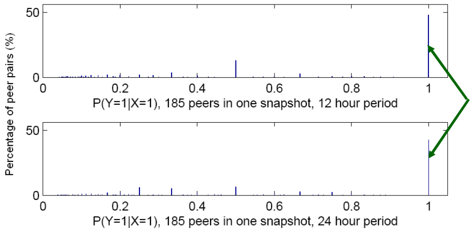


- Large overlay, more clustered



P2P file sharing overlays are clustered. [*Ripeanu 02, Saroiu 03*]

Nodes in one Snapshot Have Correlated Availability

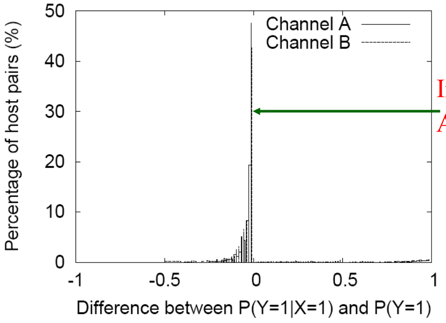


Correlated Availability

Nodes appearing together is likely appear together again

In P2P file sharing, nodes are uncorrelated [Bhagwan 03]

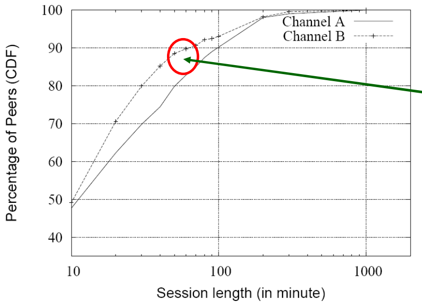
Random Node Pairs (across snapshots) Have Independent Availabilities



Independent Availabilities

Similar to P2P file sharing [Bhagwan 03]

PPLive Peers are Impatient



90% sessions are less than 70 minutes

In P2P file sharing, peers are patient [*Saroiu 03*]

Feasible Directions/Discussion

- Nodes are homogeneous due to their memoryless session lengths. Does a protocol that treats all nodes equally is simple and work more effectively?
- As PPLive overlay characteristics depend on application behavior, a deeper study of user behavior may give better design principle
- Designing "generic" P2P substrates for a wide variety of applications is challenging
- Node availability correlations can be used to create sub-overlays of correlated nodes or to route media streams?
- Simulation of multimedia streaming needs to take this bimodal availability into account?
- Geometrically distributed session lengths can be used to better simulate node arrival/departure behavior

An Evaluation of Amazon's Grid Computing Services: EC2, S3, and SQS

Simson L. Garfinkel

SEAS, Harvard University

What they Did

- Did bandwidth measurements
  - From various sites to S3 (Simple Storage Service)
  - Between S3, EC2 (Elastic Compute Cloud) and SQS (Simple Queuing Service)

Results Summary

1. Effective Bandwidth varies heavily based on geography!
2. Throughput is relatively stable, except when internal network was reconfigured.
3. Read and Write throughputs: larger is better
   – Decreases overhead
4. Consecutive requests receive performance that are highly correlated.
5. QoS received by requests fall into multiple "classes"

Host	Location	N	Read Avg	Read top 1%	Read Stdev	Write Avg	Write top 1%	Write Stdev
Netherlands	Netherlands	1,572	212	294	34	382	493	142
Harvard	Cambridge, MA	914	412	796	121	620	844	95
ISP PIT	Pittsburgh, PA	852	530	1,005	183	1,546	2,048	404
MIT	Cambridge, MA	864	651	1,033	231	2,200	2,741	464
EC2	Amazon	5,483	799	1,314	320	5,279	10,229	2,209

Units are in bytes per second

Table 2: Measurements of S3 read and write performance in KBytes/sec from different locations on the Internet, between 2007-03-29 and 2007-05-03.

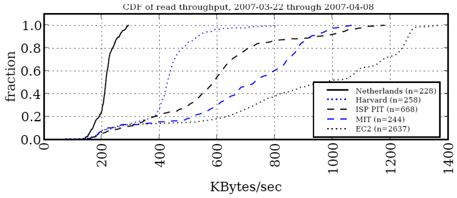


Figure 9: Cumulative Distribution Function (CDF) plots for 1MB GET transactions from four locations on the Internet and from EC2.

Effective Bandwidth varies heavily based on geography!

100 MB Get Ops from EC2 to S3

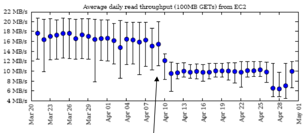


Figure 1: Average daily throughput as measured by 100MB GET operations from EC2. Error bars show the 5th and 95th percentile for each day's throughput measurement.

Throughout is relatively stable, except when internal network was reconfigured.

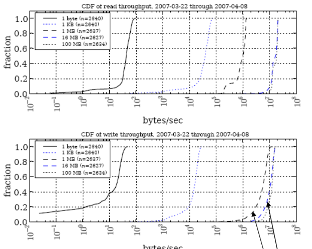


Figure 4: Cumulative Distribution Function (CDF) plots for transactions from EC2 to S3 for transactions of various sizes.

Read and Write throughputs: larger is better
(but beyond some block size, it makes little difference).

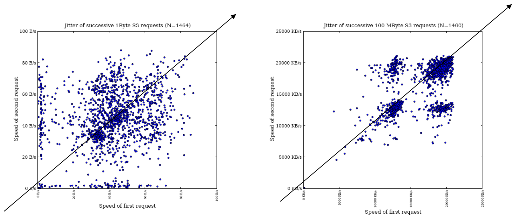


Figure 7: Scatter plots of bandwidth successive S3 GET requests for 1 Byte (left) and 100 Megabyte (right) transactions. The X axis indicates the speed of the first request, while the Y axis indicates the speed of the second.

Concurrency: Consecutive requests receive performance that are highly correlated.

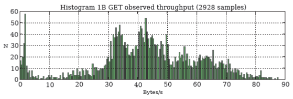


Figure 5: Histogram of 1 byte GET throughput, March 20 through April 7.

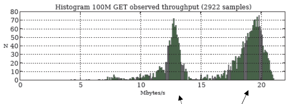


Figure 6: Histogram of 100Mbyte GET throughput, March 20 through April 7.

QoS received by requests fall into multiple "classes"
- 100 MB xfers fall into 2 classes.

Feasible Directions

1. Effective Bandwidth varies heavily based on geography!
   - *Wide-area network transfer algorithms!*
2. Throughput is relatively stable, except when internal network was reconfigured.
   - *Guess the structure of an internal datacenter (like AWS)? Datacenter tomography*
3. Read and Write throughputs: larger is better
   - *Make these better?*
4. Consecutive requests receive performance that are highly correlated.
   - *Really concurrent? Improve?*
5. QoS received by requests fall into multiple "classes"
   - *Make QoS explicitly visible? Adapt SLAs?*

Backup slides

Recommendations for P2P IPTV designers

- Node availability correlations can be used to create sub-overlays of correlated nodes or to route media streams
- Simulation of multimedia streaming needs to take this bimodal availability into account
- Geometrically distributed session lengths can be used to simulate node arrival/departure behavior
- Nodes are homogeneous due to their memoryless session lengths. A protocol treats all nodes equally is simple and works effectively
- As PPLive overlay characteristics depend on application behavior, a deeper study of user behavior may give better design principle
- Designing "generic" P2P substrates for a wide variety of applications is challenging