CS 425
Distributed Systems
Fall 2011

Slides by Indranil Gupta

Measurement Studies

Acknowledgments: Jay Patel

Motivation

- We design algorithms, implement and deploy them
- But when you factor in the real world, unexpected characteristics may arise
- Important to understand these characteristics to build better distributed systems for the real world
- We'll look at two areas: P2P systems, Clouds

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

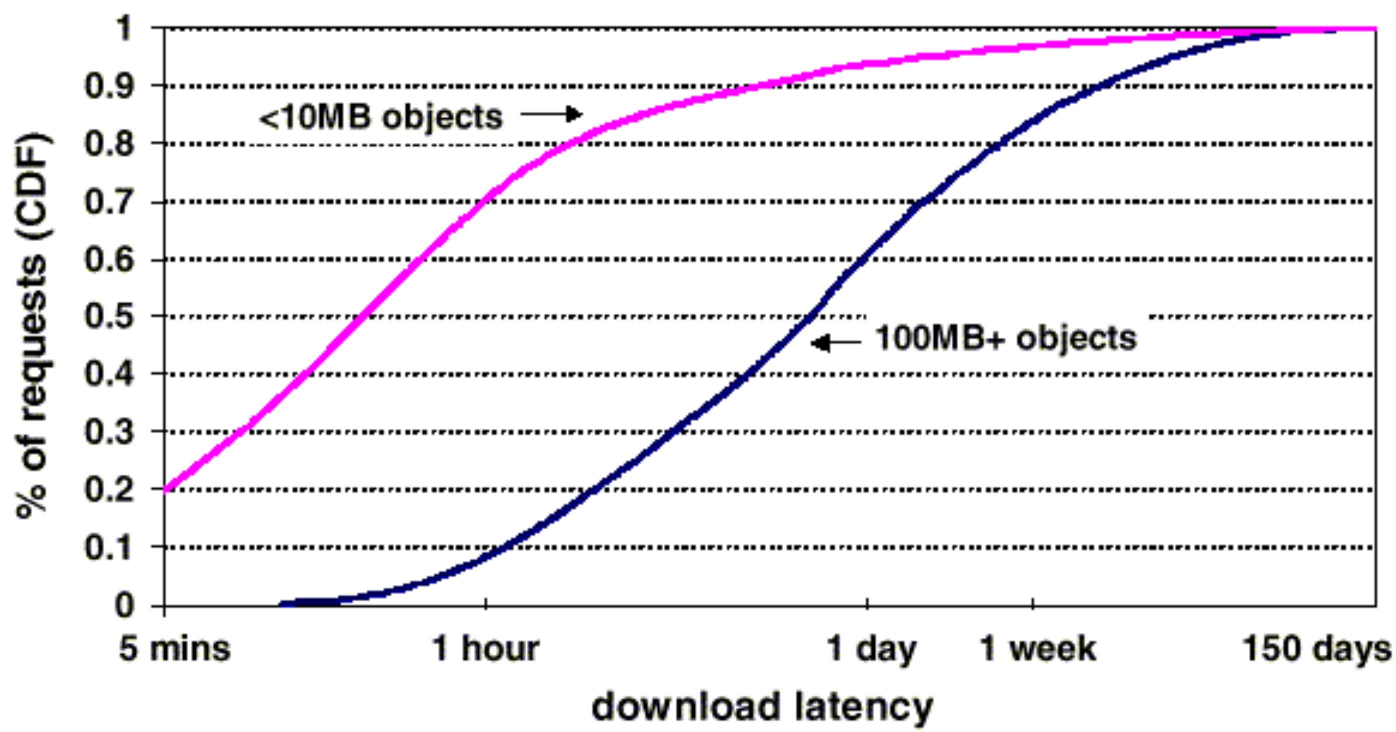
- 2003 paper analyzed 200-day trace of Kazaa traffic
- Considered only traffic going from U. Washington to the outside
- Developed a model of multimedia workloads

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

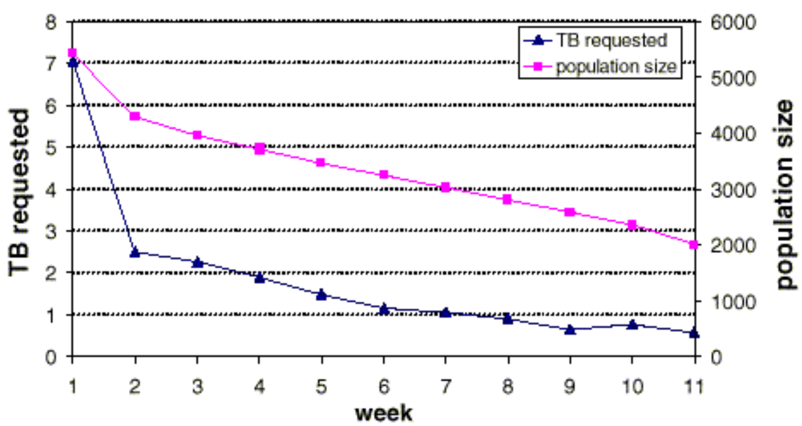
User characteristics (1)

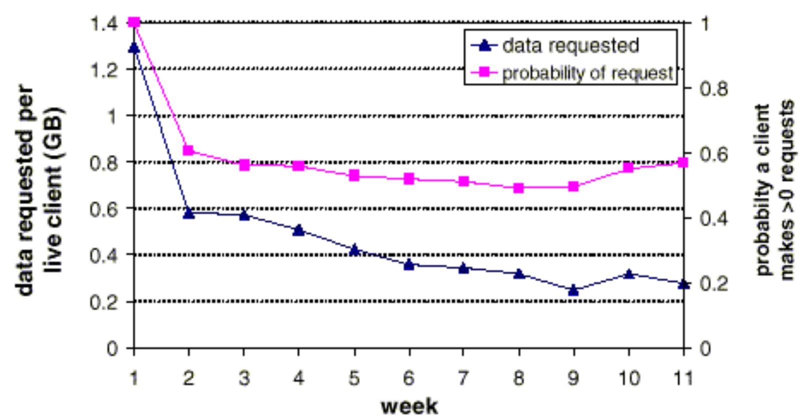
- Users are patient



User characteristics (2)

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



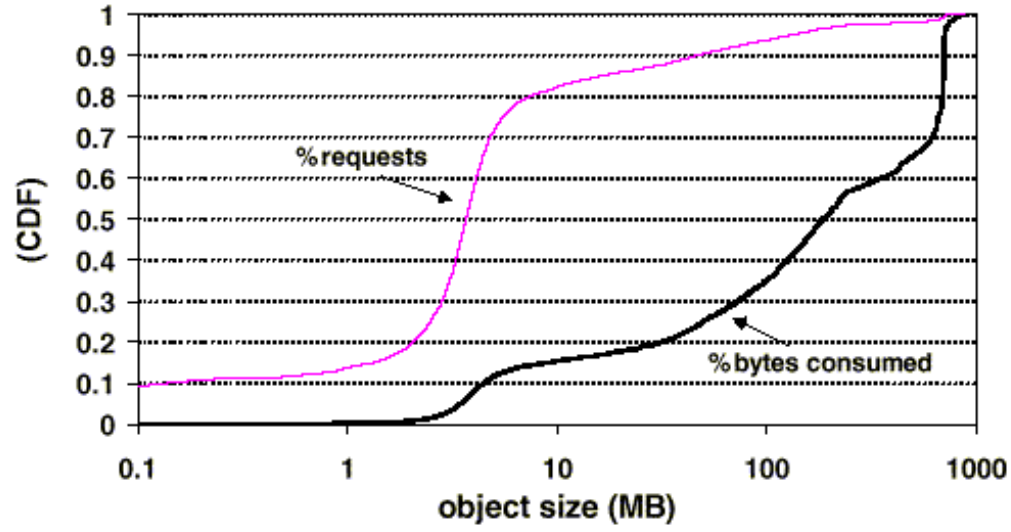


User characteristics (3)

- 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

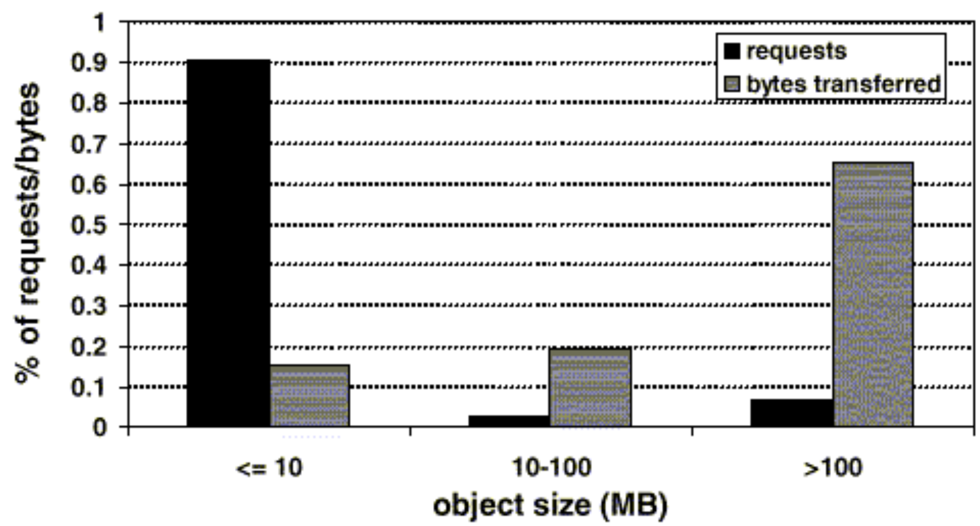
Object characteristics (1)

- Kazaa is not one workload



(a)

•This does not account for connection overhead

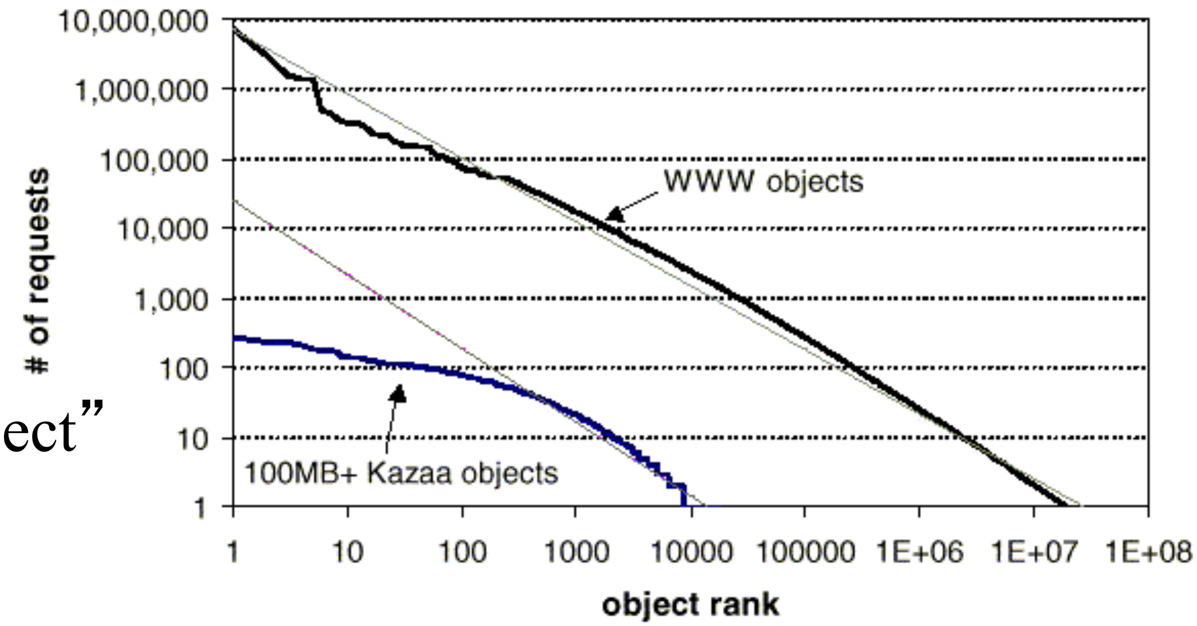


Object characteristics (2)

- 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?

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

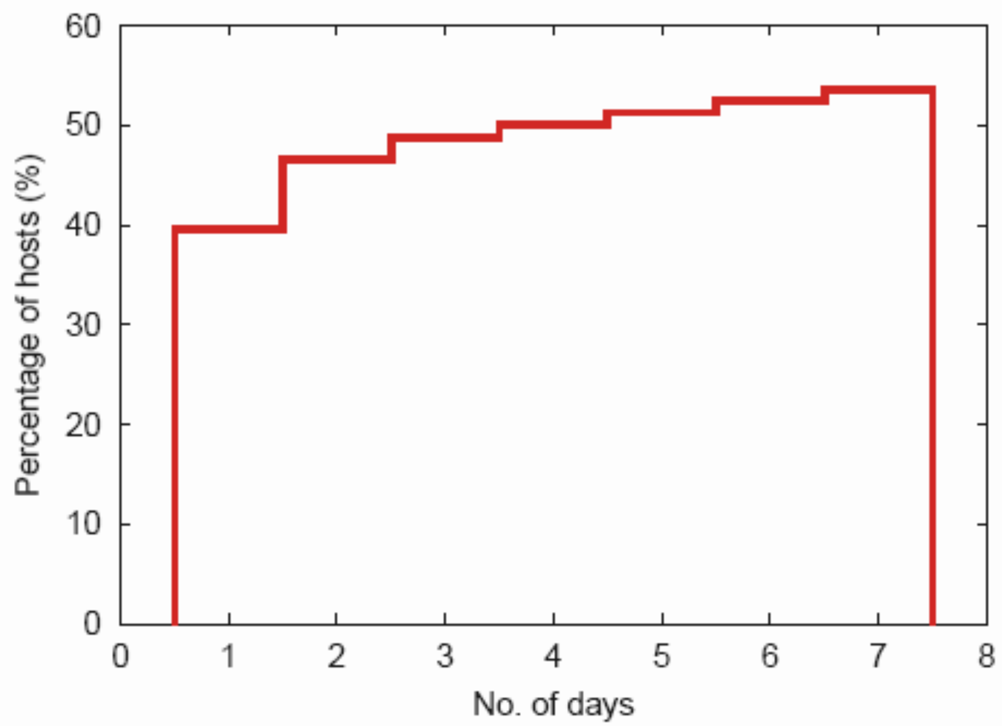


Figure 1: Percentage of hosts that have more than one IP address across different periods of time.

Availability

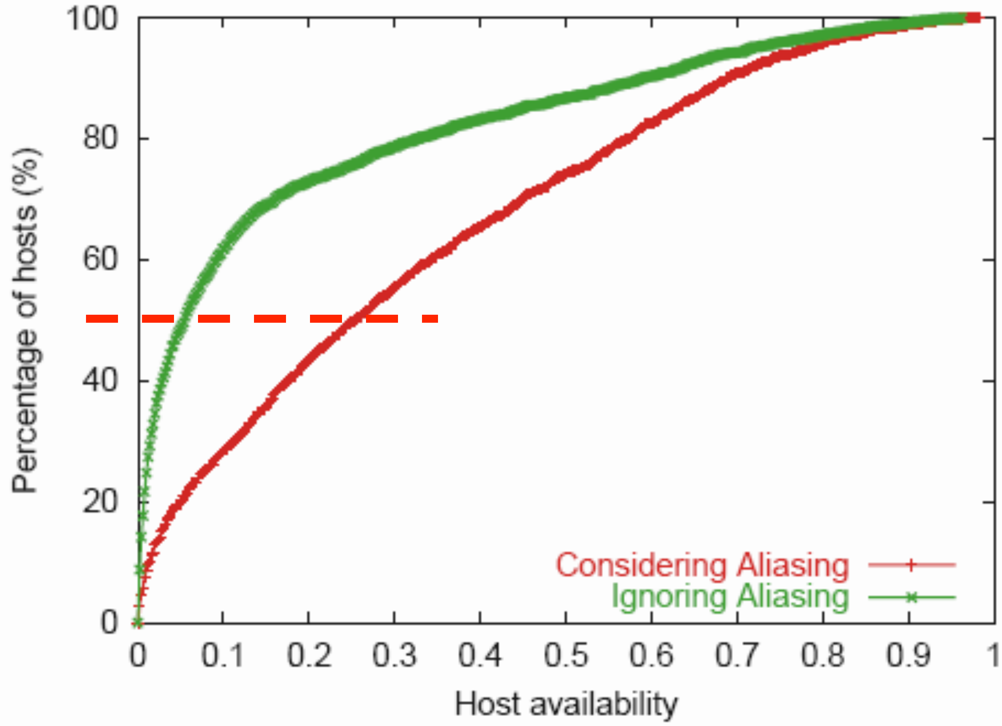
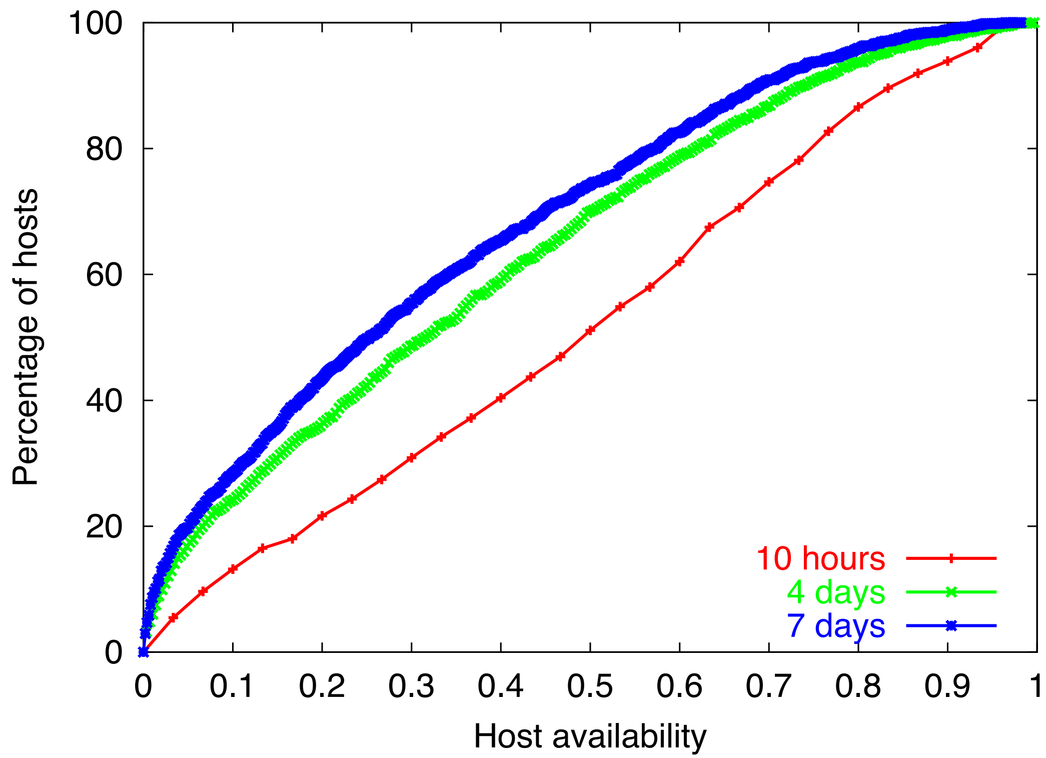
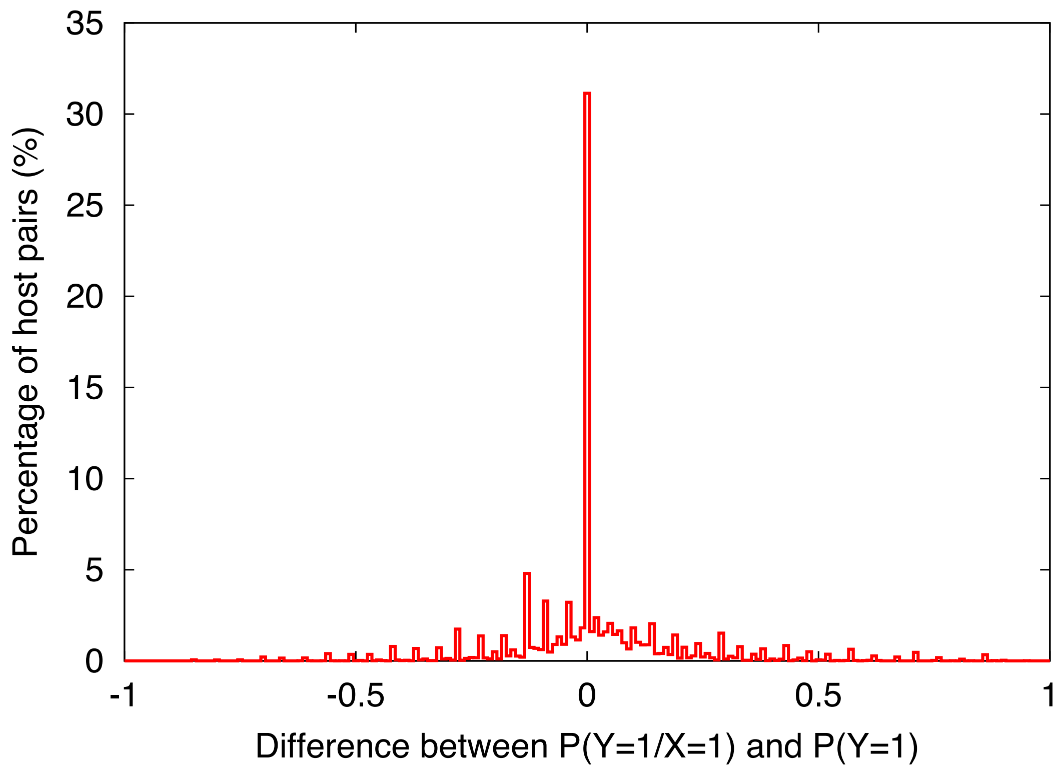


Figure 2: Host availability derived using unique host ID probes vs. IP address probes.

Availability vs. duration



Independence



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.

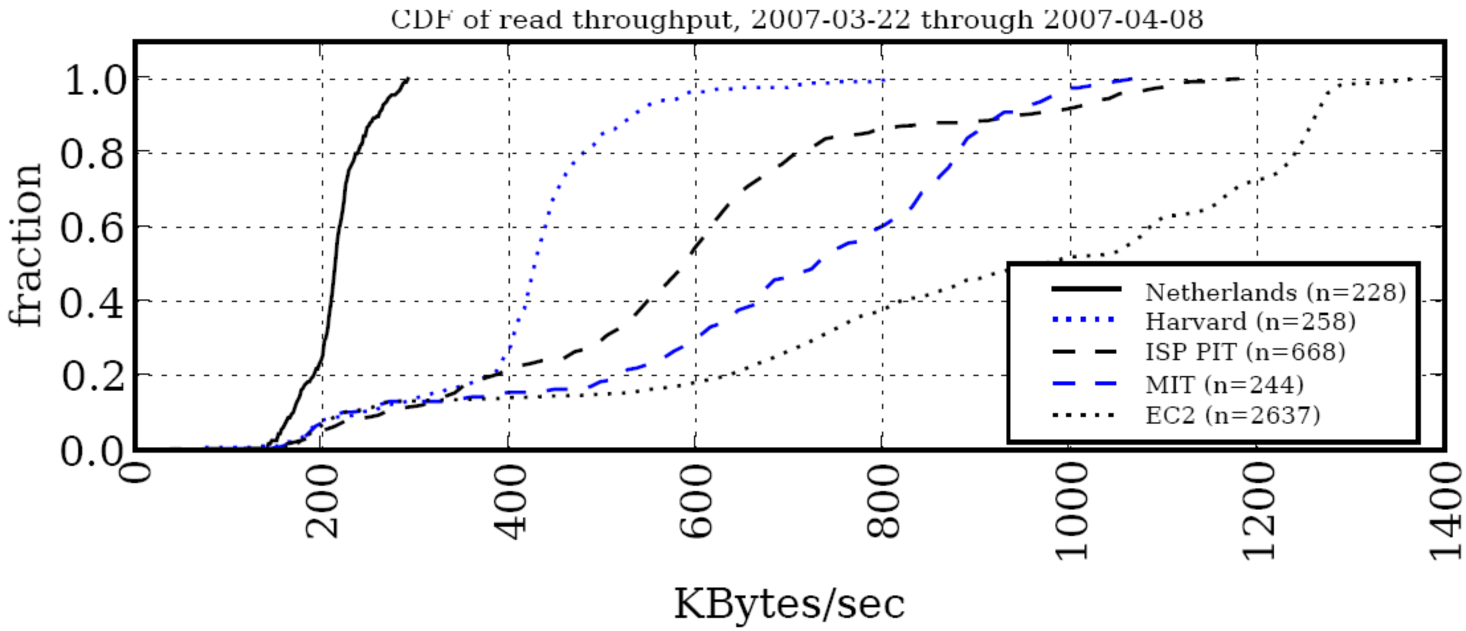


Figure 9: Cumulative Distribution Function (CDF) plots for 1MB GET transactions from four locations on the Internet and from EC2.

Effective Bandwidth varies heavily based on (network) geography!

100 MB Get Ops from EC2 to S3

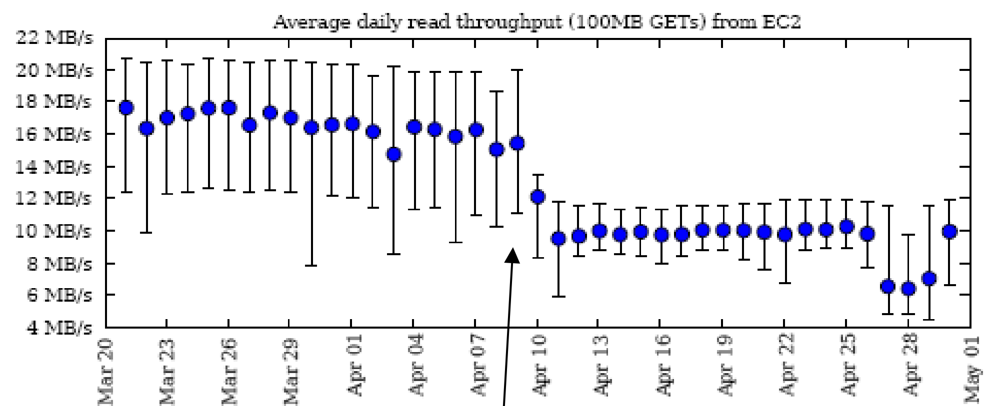


Figure 1: Average daily throughput as measured by 100MB GET operations from EC2. Error bars show the 5th and 95th percentile for each day's throughput measurement.

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

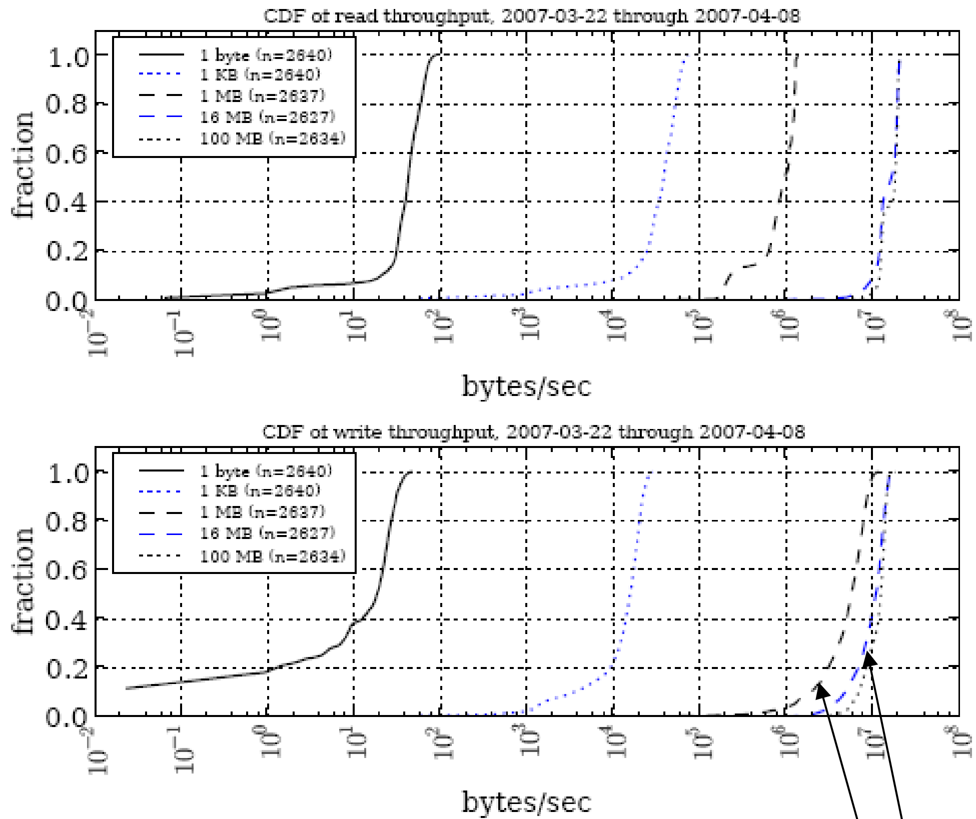


Figure 4: Cumulative Distribution Function (CDF) plots for transactions from EC2 to S3 for transactions of various sizes.

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

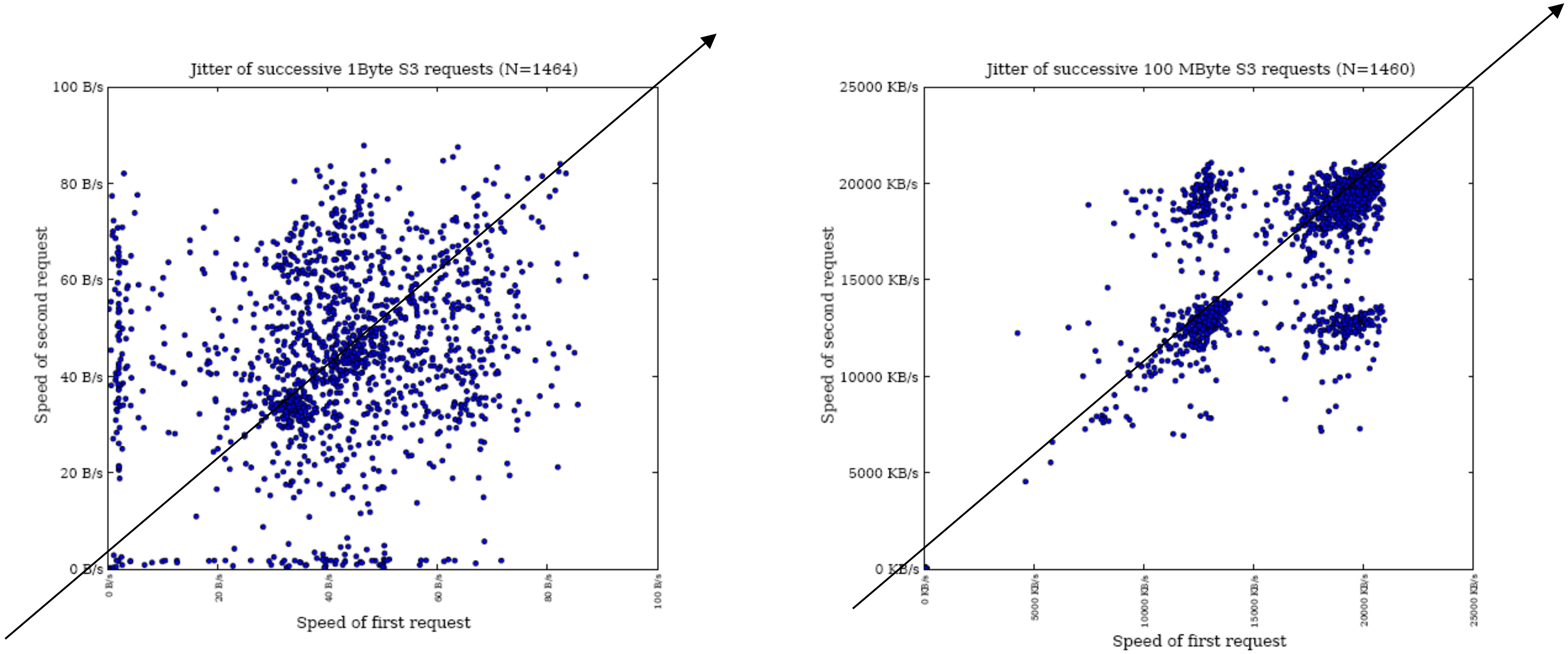


Figure 7: Scatter plots of bandwidth successive S3 GET requests for 1 Byte (left) and 100 Megabyte (right) transactions. The X axis indicates the speed of the first request, while the Y axis indicates the speed of the second.

Concurrency: Consecutive requests receive performance that are highly correlated.

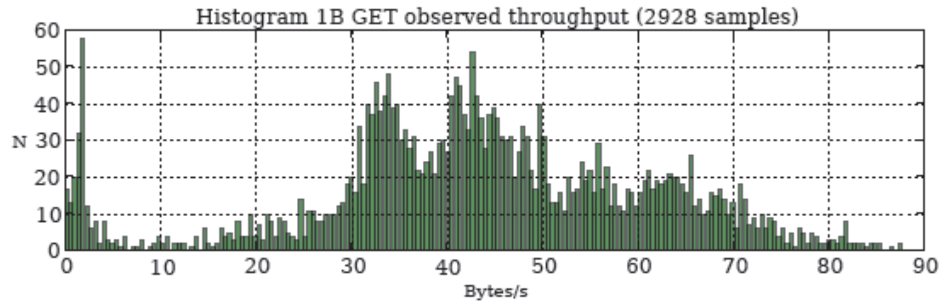


Figure 5: Histogram of 1 byte GET throughput, March 20 through April 7.

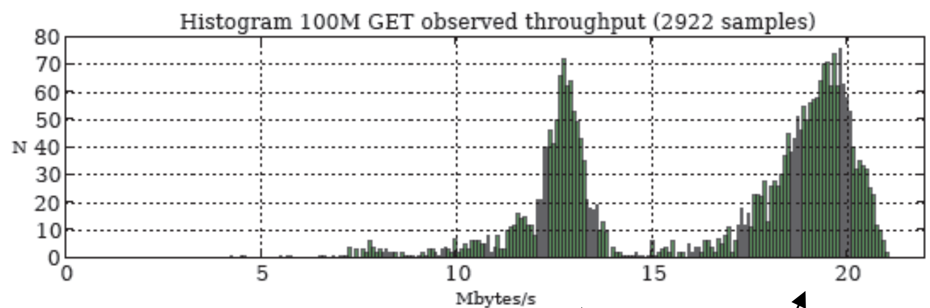


Figure 6: Histogram of 100Mbyte GET throughput, March 20 through April 7.

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

Summary

- We design algorithms, implement and deploy them
- But when you factor in the real world, unexpected characteristics may arise
- Important to understand these characteristics to build better distributed systems for the real world
- Reading for this lecture: see links on course website