Presence-based Availability and P2P Systems

Richard Dunn, John Zahorjan, Steven Gribble, Henry Levy

Department of Computer Science and Engineering
University of Washington
August 31, 2005
What this talk is about

- We introduce a new metric for computing object availability in P2P systems
- We show why this metric is better than existing method
- We show why it matters
Outline

• Motivation and Problem
• Metrics
• Evaluation
• Conclusions
Availability 101

- **Object availability** in P2P systems
  - How often are object requests successful?
  - It depends on:
    1. The *number* of peers holding object
    2. The *availability* of those peers

- Want to map these two factors to object availability
Why do we care about availability?

• We want to compare the reliability and availability of different P2P architectures
  – Use in *offline* manner to guide algorithm choice

• We want to design highly available P2P systems
  – Use *online* as part of algorithm.
  – Need to know object availability so we can adjust other factors on-the-fly
    • e.g., creating new replicas
The old world: distributed services

- An object is replicated on a set of N servers
  - Servers are centrally managed, failures are rare
  - Operational assumptions:
    i. Servers fail independently
    ii. All servers have similar uptimes

- Object availability is simple to compute
  - is the probability that at least one server is up
  \[= \quad 1 - (1 - \text{mean uptime of servers})^N\]
The new world: P2P systems

- An object is held by a set of $N$ peers
  - No central management, peers are often disconnected
  - Replication may not be managed
  - Operational realities:
    i. Peers’ downtimes are not independent -- peers prefer being online at certain times of day
    ii. Peers have very different uptimes

- Object availability is harder to compute
  - Mean uptime is not useful given these realities
Example

• System 1
  – 24 peers
  – Each peer up a different hour of the day

• System 2
  – 24 peers
  – All peers up the same hour of the day

• Both systems have the same mean uptime
  – but they provide very different object availability!
• Using mean uptime underestimates object availability
Outline

• Motivation and Problem
• Metrics
  – Mean uptime
  – Presence-based
• Evaluation
• Conclusions
Current metric

• Suppose object on \( n \) peers, need \( k \) for service (usually 1)

• Probability of one peer up:
  – Calculate uptime of each peer
  – Take mean and use as prob. for each peer (mean uptime)

• Object availability = probability that \( k \) of \( n \) peers up
New metric: Presence-based availability

• Changes to probability based on distribution

• Weighted uptime (presence)
  – Uptime of peer (or set of peers) weighted by how many other peers are up
  – Accounts for more requests during those times

• Probability different for each set
  – Object can be on any set of $n$ peers
  – Weight each set by probability it is the set with the object
Outline

- Motivation and Problem
- Presence-based availability
- Evaluation
- Conclusions
Comparison with measured availability

• Trace of availability/requests from existing P2P system (Kazaa) drives simulation
• For each request:
  – Calculate number of copies in existence
  – Mark if successful
• Find average success rate given $n$ copies
• Compare to predicted values (mean uptime vs. presence-based)
Prediction vs. measured availability
Prediction vs. measured availability

- Presence-based much closer to measured
Using availability prediction

- Can use more copies to increase availability
- How many copies to achieve availability $x$?
- Inverse of prediction function
- In practice:
  - Continuously monitor number of copies
  - Replicate as necessary to maintain minimum
- Overhead is cost of replication
Comparison with existing system

• Total Recall
  – Implements above strategy using mean uptime
• Overestimates number of copies
  – Unnecessarily high availability
  – Result: higher overhead
• Our prediction closer to the truth
Achieved availability

- Our prediction closer to target availability
• Our prediction introduces lower overhead
Outline

• Motivation and Problem
• Presence-based availability
• Evaluation
• Conclusions
Conclusions

• Object availability traditionally predicted using mean uptime
• This fails for common peer workload
• Solution: presence-based availability
• Results:
  – More closely predicts measured availability
  – Used as a parameter, achieves more correct results with lower overhead
Presence-based prediction

- **Presence** is the weighted uptime of a set
- Weights proportional to uptime of set

\[ A^{<k,n>} \equiv \frac{\sum_{S} \text{uptime}_S \cdot \text{presence}_S}{\sum_{S} \text{uptime}_S} \]

- Accounts for time-of-day and differing uptimes
Background

• Consider a P2P storage system
  – Peers with differing availabilities (uptime)
  – Requests to objects stored on these peers

• Performance based on request success

• Success based on object availability
  – Probability over time that a peer with the object is available
  – Depends on peer availability and number of copies
The Problem

• Know peer availability
• Want to *predict* object availability
  – Function of peer uptimes and number of copies
  – Result is estimated object availability
• Uses:
  – Comparing two peer workloads
  – Predicting performance of system for tuning
Formal statement

- Consider a single object
- N peers, of which \( n \) have copy of object
- Uptimes of peers over interval \( T \)
- Need \( k \) copies (typically 1) for success
- Given uptimes, what is the probability we can get object over all sets of \( n \) peers?
  - Object availability = \( F(k,n,\text{uptimes}) \)
The old world: distributed services

- An object is replicated on a set of servers
- Servers are centrally managed
- Failures are rare
- Each server has an independent uptime
- A request is successful if one server with the object is up
- Availability is simple to compute
  - Use mean uptime averaged over servers
The new world: P2P systems

• An object is held by a set of peers
• No central management
• Peers may be unreliable, disconnection is common
• Replication is unplanned and random
• Uptimes not independent (e.g., may be time dependent)
• Availability is difficult to compute
  – Mean uptime has no meaning over (very) heterogeneous peers
Traditional measure

• Calculate uptime % ($U$) for each peer
• Take mean uptime as proxy for peer availability
• Object availability is probability that $k$ of $n$ peers are up, all with probability $U$

\[ U_{<k,n>} = \sum_{j=k}^{n} \binom{n}{j} U^j (1 - U)^{n-j} \]
Problems with mean uptime

• Fails to account for two properties of P2P peers
• Peers prefer particular time of day
  – More peers up at particular time $\rightarrow$ higher object availability
• Peers have unequal uptimes
  – Object on high-uptime peers $\rightarrow$ higher availability (and vice versa)
• In general, underestimates availability
Presence-based availability

- Presence-based availability of set $S$:
  - Consider times when $S$ is available and not ($k$ peers up or not)
  - Weight each by number of other peers up at the time
  - Fraction is presence of set
- Maximum when $S$ is up at all times others are up