Antfarm: Efficient Content Distribution with Managed Swarms

Ryan S. Peterson and Emin Gün Sirer

Department of Computer Science, Cornell University
United Networks, LLC

April 22, 2009
Problem Domain

What is the most efficient way to disseminate a large set of files to a large set of clients?
Client-Server

server

clients
Client-Server

Inefficient
High cost of ownership

clients
Peer-to-Peer

peer

block transfer
Peer-to-Peer

Limited information
No control or performance guarantees
Peer-to-Peer

swarm
Antfarm Goals

- High performance
- Low cost of deployment
- Performance guarantees
  - Administrator control over swarm performance
- Accounting
  - Enables different resource contribution policies
Antfarm Approach

• Key insight: view content distribution as an optimization problem

• Hybrid architecture
  • P2P swarming with a logically centralized coordinator

• Clean slate protocol
Coordinator optimally allocates total seeder bandwidth $B$.
Antfarm

Overview

The System

Evaluation
Antfarm

Overview
The System
Evaluation
Strawman Coordinator

• One could schedule every data transfer in the system
  • All packets for all time
  • Unscalable, impractical!

• Antfarm coordinator makes critical decisions based on observed dynamics
Antfarm Coordinator

• Models swarm dynamics
  • Measures and extracts key parameters
• Formulates optimization problem
  • Calculates optimal bandwidth allocation
• Enacts allocation decisions
  • Maximizes aggregate bandwidth
  • Minimizes average download time
Antfarm Formalization

Maximize system-wide aggregate bandwidth
subject to a bandwidth constraint
Response Curves

![Graph showing response curves with slopes 1 and 0.]

- **Swarm aggregate bandwidth**
- **Seeder bandwidth**

---

$slope = 1$

$slope = 0$
Response Curves

Swarm aggregate bandwidth (KB/s)

Seeder bandwidth (KB/s)

0 25 50 75 100

0 50 100 1500
Swarms exhibit different dynamics based on size, peer resources, network conditions. . .
Swarm Dynamics
Antfarm Optimization

- Swarm aggregate bandwidth
- Seeder bandwidth

Graph shows the relationship between swarm aggregate bandwidth and seeder bandwidth for different scenarios labeled A, B, and C.
Antfarm Optimization

\[ \sigma_A + \sigma_B + \sigma_C = B \]
Performance Control

- Can provide swarm performance guarantees
  - Guarantee minimum level of service
  - Prioritize swarms
Swarm aggregate bandwidth

Seeder bandwidth

Antfarm Allocation

$$\sigma_A' + \sigma_B' + \sigma_C' = B$$
Adapting to Change

- Swarm dynamics change
  - Churn
  - Network conditions
- Antfarm updates response curves
  - Coordinator explores around point of operation
Wire Protocol

- Coordinator mints small, unforgeable tokens
- Peers trade each other tokens for blocks
- Peers return spent tokens to the coordinator as proof of contribution
Antfarm

Overview

The System

Evaluation
Antfarm Performance

![Bar chart showing Antfarm, BitTorrent, and Client-server performance.]

- **Antfarm**:
  - Zipf, 60 KB/s seeder: 1800 KB/s
  - Zipf, 200 KB/s seeder: 3800 KB/s

- **BitTorrent**:
  - Zipf, 60 KB/s seeder: 300 KB/s
  - Zipf, 200 KB/s seeder: 400 KB/s

- **Client-server**:
  - Zipf, 60 KB/s seeder: 0 KB/s
  - Zipf, 200 KB/s seeder: 0 KB/s
Swarm Starvation

BitTorrent starves the singleton swarm

- self-sufficient swarm
- singleton swarm

Avg bandwidth per peer (KB/s)
BitTorrent: Starves New Swarm

Swarms, ordered largest to smallest

- self-sufficient
- new
- singleton

Bandwidth (KB/s)

- total seeder bandwidth
- avg bandwidth per peer
Antfarm: Seeds New Swarm

Swarms, ordered largest to smallest:

- self-sufficient
- new
- singleton

Bandwidth (KB/s)

- total seeder bandwidth
- avg bandwidth per peer
Scalability

- Number of peers: 0, 20K, 40K, 60K, 80K
- Aggregate bandwidth: 1 GB/s, 2 GB/s, 3 GB/s, 4 GB/s, 5 GB/s

Graph showing scalability with different numbers of peers and aggregate bandwidths.
Scalability

Single PC can compute allocations for 10,000 swarms with 1,000,000 peers in 6 seconds.
Antfarm Implications

- No fine-tuning
- Subsumes hacks devised for BitTorrent
  - Share ratio
  - Manual pruning
Related Work

• Content Distribution Networks
  - Akamai, CoBlitz, CoDeeN, ECHOS, Coral, Slurpie, YouTube, Hulu, GridCast, Tribler, Joost, Huang et al. 2008, ...

• P2P Swarming
  - BitTorrent, BitTyrant, PropShare, BitTornado, BASS, Annapureddy et al. 2007, Guo et al. 2005, ...

• Incentives and microcurrencies
  - Dandelion, BAR Gossip, Samsara, Karma, SHARP, PPay, Kash et al. 2007, ...
Conclusions

• Model swarm dynamics and allocate bandwidth optimally

• Novel hybrid architecture

• PlanetLab deployment shows that Antfarm outperforms client-server and P2P
Questions?