Kudzu: A Self Balancing P2P File Transfer System

Sean Barker, Marius Catalin Iordan, Jeannie Albrecht, and Barath Raghavan
Williams College and UC San Diego
{09sb, 09mci, jra1}@williams.edu and barath@ucsd.edu

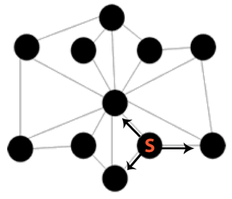
Goal: Provide a scalable file transfer system with autonomous organization

Motivation

- Problem: How do we design a file transfer system that sacrifices neither scalability nor efficiency?
- Existing systems typically employ central control (e.g., BitTorrent), do not scale well (e.g., Gnutella), or possess other undesirable properties (e.g., DHTs)
- We want to provide a decentralized, self-governing system with both short-term and long-term organizational strategies
- Kudzu takes Gnutella's decentralized model and imposes a self-governing structure on top of it

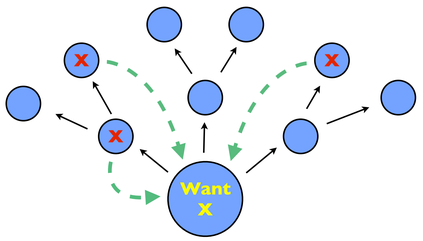
Architecture

- A Kudzu network is comprised of a set of functionally homogeneous peers
- New peers may join by contacting any existing peer
- Peers are connected to a small subset of peers in the network
- No central peer manager - each node independently manages its peer set



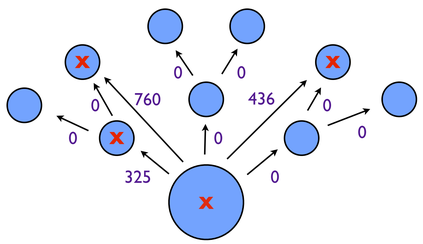
Query Propagation

- Peers flood queries to all known neighbors, which then forward queries along
- Every peer receives every query and connects back to requester if the desired file is found
- No query TTL, but queries already received and forwarded are discarded to ensure that no infinite propagation occurs
- Downloads proceed simultaneously from all answering peers



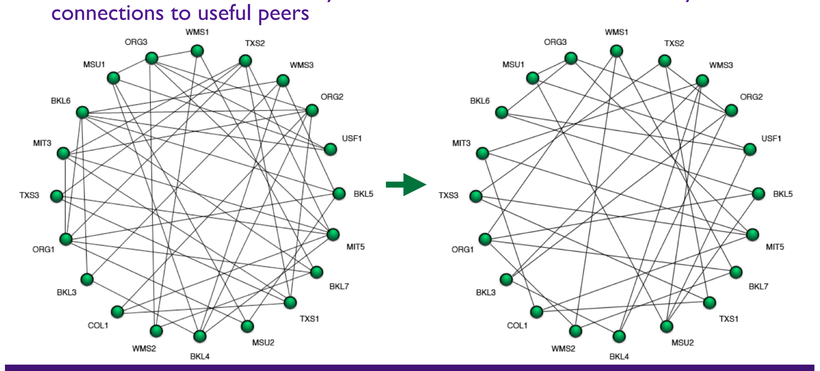
Connection Management

- Minimum number of peer connections per node to maintain a connected network and reasonable query dissemination speed
- Initial connections established randomly from known peers
- New connections established when files are transferred
- Connections ranked in importance by number of bytes transferred



Network Balancing

- Query duplicates from flooding used as measure of connection redundancy
- Large numbers of duplicate queries result in disconnecting neighbors, thus reducing the number of future duplicates
- Low-use connections favored for disconnection, high-use connections protected from disconnection
- Network settles into relatively stable state of reasonable redundancy and connections to useful peers



PlanetLab Preliminary Results

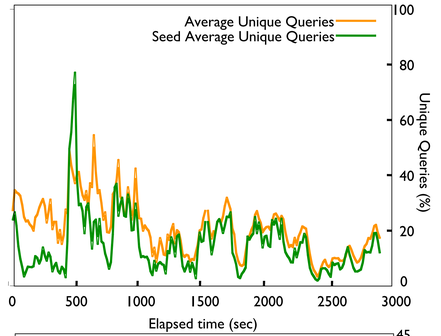
- Experiments run on PlanetLab to test preliminary performance in a large, wide-area network
- Run on ~400 active nodes

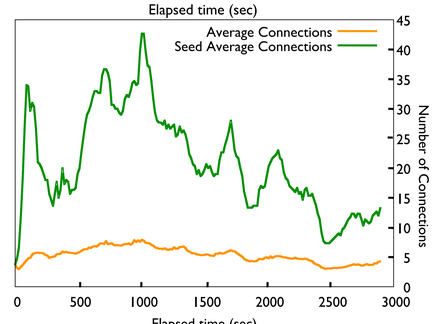
Network Load Tests

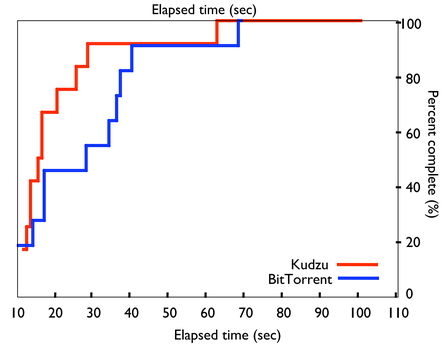
- A few nodes (<1% of total) seeded with a set of files
- All nodes generate large number of queries (both for actual files and nonexistent ones) to stress the network
- Percentage of unique queries among total queries (uniques plus duplicates) recorded
- Network becomes more heavily interconnected during large-scale transfer activity
- Also measured average number of peer connections
- Network gradually offloads bandwidth request from seeders and trends towards fewer connections as transfers complete

Download Speed Tests

- Fixed a set of seeder and leecher nodes scattered around PlanetLab
- Tested leecher download times in a Kudzu and BitTorrent network
- Kudzu compares favorably in raw performance







Machine Learning

Intelligent Peer Selection

- Low-information approaches such as amount of connection use work well in the short term, but can be tuned on a per-node basis via parameters such as enforced number of connections or the connection 'high-use' cutoff
- Longer-term network evolution can be improved via learning algorithms, such as SVM learning to choose neighbors based on file and query similarities as described in [Beverly and Afergan 2007].
- Other sources of peer information may come from users themselves from potentially out-of-band social networks [Pouwelse et al. 2008]

Query Forwarding Behavior

- Possibilities besides complete flooding being investigated include partial flooding or random walks as described in [Chawathe et al. 2003]
- Modified forwarding behavior may be used as a substitute or supplement for adding or removing peer connections
- Minimizing the total number of queries presents the most important scalability issue in a decentralized system like Kudzu

Ongoing Work

File Searching

- Provide intelligent keyword or content-based queries

Incentives and Collaboration

- Incorporate sharing incentives such as those employed by BitTorrent
- Support collaborative downloads involving peers neither requesting nor seeding (may be tied to the incentive model)

PlanetLab Experiments

- Simulate typical-use load or incorporate preexisting data so as to reliably evaluate ML techniques in a 'live' setting
- Complete tests for a variety of network balancing schemes in Kudzu

Summary of Contributions

- Preliminary implementation of Kudzu, a decentralized P2P file transfer system
- Autonomous network organization techniques with no central control
- High performance transfers relative to existing systems
- Ongoing work in ML applications to improve self-adaptive behavior

References

R. Beverly and M. Afergan. "Machine learning for efficient neighbor selection in unstructured P2P networks." SysML 2007.

Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. "Making gnutella-like P2P systems scalable." SIGCOMM 2003.

J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D.H.J. Epema, M. Reinders, M. van Steen, and H.J. Sips. "Tribler: A social based peer-to-peer system." Concurrency and Computation: Practice and Experience 2008.