IEEE
distributed systems
ONLINE
Expert-authored articles and resources

IEEE DISTRIBUTED SYSTEMS ONLINE 1541-4922 © 2005 Published by the IEEE Computer Society
Vol. 6, No. 7; July 2005

Computational Sociology for Systems "In the Wild": The Case of BitTorrent

David Hales, *University of Bologna*
Simon Patarin, *University of Bologna*

It's generally agreed that future software systems should be distributed, open, self-organizing, scalable, and robust. Fully distributed systems can't rely on centralized control, and open systems can't ensure that malicious or selfish components don't invade them. The requirement for high scalability means that systems should run at least as well, and ideally better, when scaled to millions of units. How do you begin to formulate methods, techniques, and protocols that can deliver on these tough demands?

One approach, often adopted in the multiagent-systems community, is to start from scratch, designing agents and platforms with provable properties using specialized logics and/or sophisticated simulation models. However, this approach is particularly difficult when dealing with open systems containing adaptive agents. This is because the designer can't be sure how other agents will behave in future states of the system.[1] Worse, much of the desirable behavior of the system as a whole, such as high levels of altruism or cooperation, often result from emergent properties that are little understood and not easily reducible to individual behaviors. However, researchers are making progress with this approach.[2]

Systems "in the wild"

Another possible approach is to consider existing deployed systems that actually deliver at least some of these desirable properties. Specifically, peer-to-peer file-sharing systems are massively deployed and appear to support high levels of desirable emergent functionality such as cooperation and altruism. However, these systems generally aren't engineered according to known methodologies, because such methodologies don't exist. Rather, these systems are implemented informally within open-source communities and essentially tested "in the wild." Systems that work will tend to be adopted, resulting in millions of users downloading and running the P2P client software on their machines and connecting over the Internet, which becomes a node in the system.

What's particularly challenging and intriguing about such systems in the wild is that apparently nobody fully understands why they work and if they will continue to work! Although client software and protocols are known, predicting the complex interactions between nodes due to the clients' constantly changing ecology is almost impossible. However, by analyzing these systems, we might be able to identify key mechanisms that can serve to inform the development of engineering methods.

Computational sociology

One way of beginning to understand such systems is to employ methods and models developed in the social sciences, particularly the emerging area of computational sociology. (For examples of research in this area, see the *Journal of Artificial Societies and Social Simulation*, http://jasss.soc.surrey.ac.uk/.) This shouldn't come as a surprise because P2P systems are essentially social systems realized in the computational domain. In the following sections, we consider the specific example of the BitTorrent file-sharing system. We propose a novel hypothesis concerning why it works, informed by recent ideas emerging from social-simulation research.

BitTorrent and the tit-for-tat strategy

BitTorrent (http://bittorrent.com/) is currently king of the popular file-sharing clients. Some reports have claimed that BT accounts for most peer-to-peer traffic on the Internet; others have reported that Hollywood is doomed owing to the ease of downloading movies through BT. Bram Cohen, BT's inventor, is in high demand on the invited-talks circuit. However, leaving the media attention aside for one moment, we ask, why does BT work so well?

BT attempts to build robustness to freeloading (that is, downloading without uploading) by implementing a tit-for-tat-like strategy in its protocol. This strategy is often believed responsible for the high levels of cooperation found among BT users. Robert Axelrod championed the TFT strategy in his now classic, 1980's book *The Evolution of Cooperation*.[3] He held computer tournaments in which different researchers' programs repeatedly played the canonical game of cooperation—the prisoner's dilemma. He found that in a round-robin tournament, TFT performed best on average against the other strategies. TFT is relatively simple. It starts by selecting a cooperative move and then for subsequent moves copies its opponent's last move. This strategy is encapsulated in the BT tagline: "Give and ye shall receive."

In BitTorrent, groups of peers (called *swarms*) with an interest in downloading a specific file coordinate and cooperate to accelerate the process.[4] Essentially, a *tracker* node stores a list of peers in the swarm, thus letting new peers join the swarm. Each peer stores pieces of the file. Cooperating peers download and upload required pieces. If a peer stops uploading, other peers will likely "choke" it; that is, they stop uploading to it. This implements the TFT-like process.

Seeders, peers that store the whole file, are crucial to a swarm's functioning. If a swarm contains no seeders, eventually some pieces of the file might be completely missing from the swarm. Because peers gain nothing themselves by being seeders, the system requires some altruistic behavior from peers. This requirement is reflected by the mantra often repeated on BT Web sites: leave your download running for a little while after you've got the entire file.

We argue that this TFT-like strategy doesn't adequately explain the high levels of cooperation found in BT because

- other, less cooperative strategies can outperform the TFT strategy,
- users can create fake identities by modifying the client, thus circumventing TFT, and
- to operate, BT requires unconditional altruism in any case.

Given that such loopholes exist, why don't freeloaders dominate the system?

Hypothesis: Group selection

We hypothesize that BT might resist freeloaders and support altruism, at least partly, in a way that hasn't been previously fully comprehended. Ironically, this process relies on what is commonly believed to be a weakness of BT—the lack of integrated metadata search. One consequence of this is the BT network's partitioning into numerous isolated swarms—often with several independent swarms for an identical file. Such partitioning is a necessary condition for a kind of novel group-selective process recently identified in similar simulated systems in the context of both computational sociology and simulated P2P file sharing.

Essentially, if users move between swarms (leave one swarm and enter another) on the basis of the quality of the service they receive, swarms containing many freeloaders will tend to "die" as peers leave the swarm for better swarms. Swarms that contain altruists will tend to grow because they support a quality service. Computational-sociology researchers have advanced similar models.[5,6,7]

A further implication of our hypothesis is that, given the choice, users might choose unconditional altruism rather than the more restrictive reciprocal approach.[8] This is because the group-selective process selects for pure altruism—peers acting for the group's benefit at their own individual cost.[9]

Conclusion

One way to test our hypothesis empirically would be to implement and distribute a modified BT client that lets users select pure altruism. This might be the subject of future research.

References

1. Z. Guessoum, "Adaptive Agents and Multiagent Systems,"*IEEE Distributed Systems Online*, vol. 5, no. 7, July 2004, http://csdl2.computer.org/comp/mags/ds/2004/07/o7004.pdf. (PDF)
2. G. Di Marzo Serugendo et al., eds., *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering,* LNAI 2977, Springer-Verlag, 2004.
3. R. Axelrod, *The Evolution of Cooperation,* Basic Books, 1984.
4. B. Cohen, "Incentives Build Robustness in BitTorrent," presented at the 1st Workshop Economics of Peer-2-Peer Systems, 2003; www.sims.berkeley.edu/research/conferences/p2pecon/papers/s4-cohen.pdf. (PDF)
5. D. Hales "Cooperation without Memory or Space: Tags, Groups and the Prisoner's Dilemma,"*Proc. 2nd Int'l Workshop Multi-Agent-Based Simulation,* LNAI 1979, Springer-Verlag, 2000, pp. 157–166; www.davidhales.com.
6. R. Riolo, M.D.Cohen and R. Axelrod, "Evolution of Cooperation without Reciprocity,"*Nature* , vol. 414, no. 6862, 2001, pp. 441–443.
7. D. Hales, "From Selfish Nodes to Cooperative Networks—Emergent Link-Based Incentives in Peer-to-Peer Networks," http://doi.ieeecomputersociety.org/10.1109/PTP.2004.1334942 *Proc. 4th IEEE Int'l Conf. Peer-to-Peer Computing* (P2P 2004), IEEE CS Press, 2004, pp. 151–158.
8. R. Trivers, "The Evolution of Reciprocal Altruism,"*Quarterly Rev. of Biology* , vol. 46, 1971, pp. 35–57.
9. D. Hales and S. Patarin , *How to Cheat BitTorrent and Why Nobody Does,* Univ. of Bologna, Dept. of Computer Science, tech. report UBLCS-2005-12. http://www3.cs.unibo.it/pub/TR/UBLCS/2005/2005-12.pdf (PDF)

David Hales is a postdoctoral researcher in computer science at the University of Bologna. Contact him at dave@davidhales.com.

Simon Patarin is a postdoctoral researcher in computer science at the University of Bologna. Contact him at patarin@cs.unibo.it.

Related Links

Purchase or Digital Library members log in:

Free Riding on Gnutella Revisited: The Bell Tolls?
(http://dsonline.computer.org/portal/pages/dsonline/0506/o6001.html)

DS Online Peer-to-Peer Community (free)
(http://dsonline.computer.org/portal/site/dsonline/index.jsp)

Dependability in Peer-to-Peer Systems
(http://csdl.computer.org/comp/mags/ic/2004/04/w4054abs.htm)

Cite this article: David Hales and Simon Patarin, "Computational Sociology for Systems "In the Wild": The Case of BitTorrent," *IEEE Distributed Systems Online*, vol. 6, no. 7, 2005.