Hypercubes: A Superior Topology for Real-Time Genealogical Collaboration Networks

Ronald Crowther and Scott N. Woodfield
Department of Computer Science,
Brigham Young University
Provo, UT 84602 USA

Genealogical research can be significantly enhanced through real-time collaboration. Such collaboration reduces redundant research and provides timely information that can better guide one to discover new knowledge. In real-time collaboration, information recorded by one participant is nearly instantaneously and automatically broadcast to all other people interested in that information. Upon reception the information is semi-automatically incorporated into the recipients database. Real-time collaboration requires a communication infrastructure that is fast, reliable and scalable. Currently, client/server topologies are used for most real-time collaboration frameworks but client/server technologies are expensive, they do not scale well, are relatively unreliable, and are not that fast when dealing with large numbers of users.

Peer-to-peer systems appear to be faster, more reliable, scalable and cheaper. They are cheaper because there is no central server to purchase and maintain. Peers need only pay for local hardware and connections to the internet which they would also pay for in a client/server- based system. However, there are different types of peer-to-peer topologies. To determine which is best we compare the broadcast speed, reliability, and scalability of several different peer-to-peer topologies with the client/server topology. The different peer-to-peer topologies include: random trees, random graphs, gnutella-like topologies, and hypercube networks. Random trees are formed by randomly adding new nodes to one of the existing nodes in the network. Random graphs are created by connecting a new node to two or more of the existing nodes. For gnutella-like topologies we analyzed a real gnutella network and created a graph building algorithm that would build random gnutella-like networks. Hypercube topologies are structured networks in which each node is connected to exactly log_2 other nodes in the network.

To measure speed we determined the time it would take to broadcast a message to all nodes in a network. We measured time by the longest chain of point-to-point communication events required to broadcast a message. For the client/server and hypercube topologies we determined the amount of time analytically. For the other networks we simulated thousands of broadcasts on hundreds of randomly created networks. From the simulations we derived average broadcast times.

Reliability was measured by the expected number of communication failures a user would experience in a year. For this metric we tried to approximate the average amount of traffic expected in a genealogical network. We assumed that most participants in a network would most likely just receive information and would not contribute new information for dissemination. We approximated this by assuming that there was a 1 in 500 chance that any person would broadcast information on any given day. We also assumed that client/server systems were 99.9% reliable (the claimed reliability of many providers) and that machines in a peer-to-peer network were 99% reliable. Based on these assumptions, we calculated the number of broadcasts a year that a user would not receive (or would be delayed) because of node failures in the network. As with the speed studies, we calculated the results for the hypercube and client/server topologies analytically. For the other topologies we derived the results from simulation studies. While other assumptions may change the magnitude of the reliability results, they should not change the relative ranking of the different topologies.

Scalability was determined by measuring how speed and reliability changed for a given network as that network grew in size. To do this, we measured speed and reliability metrics for networks of 2, 4, 8, 16, 32, 64, 128, 256, 512, and 1024 nodes. Thus, the sizes grew exponentially. Since the broadcast speed and reliability of client/server topologies grows linearly with respect to the number of nodes in a network, a peer-to-peer topology would be considered more scalable if it grew sub-linearly (e.g. logarithmically) or if the slope of the increase was smaller than that of the client-server system.

Our results show that the hypercube was the fastest topology for all ten sizes of networks. Random graphs seemed to be the next fastest in most cases, followed by random trees. Gnutella-like networks were faster than random trees and random graphs for small networks but were much slower for larger networks. Our results also showed that for speed, hypercubed topologies scaled best. They scaled logarithmically. It also appears that random graphs and random trees grew logarithmically but were inferior to hypercubes. Random trees had slightly higher values than random graphs and random graphs had values that were about three times higher than hypercube networks. The gnutella network seemed to scale at best linearly with network size. It could even be exponential.

The reliability results were a little more mixed. In general, the reliability for random graphs and hypercubes were excellent with random graphs being a little better than hypercubes in small networks and hypercubes being better than random graphs for large networks. The reliability of client/server systems was also excellent but just a little worse than random graphs and hypercubes. Gnutella networks were good when networks were small but became relatively worse as networks became bigger. The random tree was reliable for small networks but became less reliable for larger systems. As for scalability, hypercubes were the best. They actually became more and more reliable as networks became larger. Random graphs were also very good. Their reliability metric remained essentially unchanged for networks of from 2 to 1024 nodes. It does appear that their reliability may start to diminish for larger networks. The reliability of both the random tree and the gnutella network declined linearly with the size of the network. The random tree was worse than the gnutella network.

From these results we conclude that the hypercube network is most suitable for real-time genealogical collaboration networks. It is the fastest, one of the most reliable, and the most scalable topology. Random graphs may also be suitable. We suggest that client/server, random tree, and gnutella topologies not be considered.