Robert Jones

Prof. Lupo

CPE 458-01

Chord Response

Even as someone who is unfamiliar with peer to peer systems in general, I can appreciate the technical advancement that is the Chord protocol. As one of the first four peer to peer/distributed hashing protocols developed, along with CAN, Tapestry, and Pastry, the fact that it is still in use today is a testament to the ingenuity of the underlying concepts that make up the protocol. I feel that as technology progresses and the hardware available to our computing platforms becomes more and more robust, the major areas of advancement will be in the algorithmic approaches to our use of distributed systems. Before long I can see the major criteria of memory and CPU usage of applications shift towards things like load balancing, decentralization, scalability, and flexibility, as is described in the MIT Laboratory for Computer Science paper titled “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications” section 3. Specifically I found Chord’s implementation of consistent hashing to be particularly interesting. It makes intuitive sense to involve a modulo of the total size of the system in nodes as part of the hashing function. The one part of the paper that I currently do not understand completely would have to be the section on stabilization. I suppose further research would help explain how the nodes could reach a situation where there was a risk of the graph breaking into multiple disjoint cycles.