Ringer: A Global-Scale Lightweight P2P File Service

Ian Pye
UC Santa Cruz
Santa Cruz, CA
ipye@cs.ucsc.edu

Scott Brandt
UC Santa Cruz
Santa Cruz, CA
sbrandt@cs.ucsc.edu

Carlos Maltzahn
UC Santa Cruz
Santa Cruz, CA
carlosm@cs.ucsc.edu

Whenever there is a need to coordinate on a global scale, solutions which work perfectly on a LAN tend to fall apart. Different ideas are needed. We present Ringer as a solution to the problem of keeping a secure, reliable, simple (and cheap) distributed filesystem working at a global scale along unreliable network connections. Furthermore, Ringer is specifically designed to facilitate distributed document management.

This sounds an awful lot like a problem which the Internet and Google (or your search engine of choice) solves admirably already, at least in the case of HTML documents. Now, consider the case of information stored in general format documents – and especially for documents which exist only internally in a company. The PageRank algorithm works by exploiting the rich interconnectedness of hypertext. However, this highly connected graph structure doesn't exist for most sets of arbitrary documents. Additionally, most information is stored in only a few documents, not 100s! In this context, internet-based search is not successful because the ordering of results returned from a search query is both more important and more difficult to compute.

Suppose we have the following situation: There is a global organization (e.g., the UN) which generates a large amount of information, but this information is not stored in hypertext. Furthermore, this information is rarely duplicated across documents. Our goal is to easily structure these documents in such a way as to make adding AND finding information easy. There is currently no good solution to the problem outlined above in the literature. To solve this problem, Ringer has three features as goals:

1. **Filesystem Semantics**: Concurrency control and close-to-open consistency are essential to keep things manageable in a distributed system with many readers and writers.
2. **Database-Style Indexing**: We need the ability to search on arbitrary file attributes and get good results quickly. Ringer supports different ways of sorting content.
3. **Internet-Style Connectivity**: Files need to be available on demand, but only downloaded (an expensive operation on a WAN) as needed. Most files are accessed rarely.

A naive approach to this problem might be to simply rely on email to spread documents. This has the benefit of being easy, and a process which (most) everyone understands. However, using this approach, one has no idea what documents are out there – you only know what documents you have. As the number of files scales, this approach falls apart. Another idea is for a centralized document repository to be created. However, at a global scale, not all users will have fast access, unless caching is used. Caching makes it difficult to support filesystem semantics when users are actively reading and writing documents. Conversely, documents could be posted online, and then indexed using a PageRank-like algorithm. We previously discussed the shortcomings of this approach.

Instead, we are investigating a third approach: A Hybrid P2P system. The goal of our architecture is to separate the data and metadata. Clients transfer data directly between each other in a traditional P2P manner. Clients find each other by communicating via a pyramid of metadata servers (MDS). This differs from other P2P systems such as Ivy[3]; while there is no single MDS, there is also not a complete lack of centrality. This approach simplifies implementation and search while avoiding the bottlenecks caused by a single central MDS.

As depicted in Figure 1, each client is connected to one MDS. This MDS keeps metadata for all files stored on "its" clients. This MDS also is connected to a parent MDS, which contains pointers to the files indexed in its client MDSs. A parent MDS can also have a parent, allowing for an arbitrary number of MDSs.

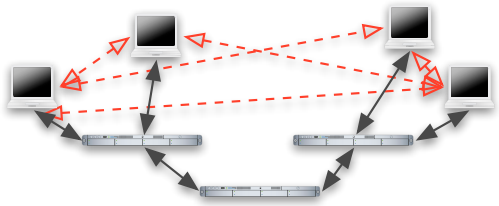

Figure 1: Basic Ringer Architecture

A recent P2P distributed filesystem is Kosha[1]. Kosha is a enhancement to NFS, and is targeted for work in a LAN environment. Conversely, Ringer is optimized for a very wide area network. In this, Ringer is similar to Muthitacharoen's Low Bandwidth FS[2], although LBFS uses a traditional client-server structure.

We hope that Ringer will prove a reliable, efficient and scalable way to facilitate globally distributed collaboration. Ringer's strength lies in its ability to cheaply combine many distinct LANs into a single logical volume, while scaling easily with both users and files. Finally, Ringer was conceived from the ground up to allow files to be effortlessly added – and just as effortlessly found again.

[1] A. Butt, T. Johnson, Y. Zheng, and Y. Hu. Kosha: A Peer-to-Peer Enhancement for the Network File System. *Journal of Grid Computing*, 4(3):323–341, 2006.

[2] A. Muthitacharoen, B. Chen, and D. Mazières. A low-bandwidth network file system. *Proceedings of the eighteenth ACM symposium on Operating systems principles*, pages 174–187, 2001.

[3] A. Muthitacharoen, R. Morris, T. Gil, and B. Chen. Ivy: A Read/Write Peer-to-Peer File System.