DISTRIBUTED MONOTONICITY

Kevin Cheek
Alex Maskovyak
Joseph Pecoraro

AGENDA

- Overview
- Describe Problem
- Approach
- Deliverable
- Paper Citations

OVERVIEW

- We plan on building a distributed backup system.
- The system will offer a RAID-like data redundancy on a distributed system of computers.
- We will make use of an Open Source Peer-to-Peer System Framework to keep track of file divisions, reliable redundancy across the “cloud,” and efficient routing, and network maintenance chores.

PROBLEM

- Partition files and replicate them across a semi-reliable cluster of distributed computers with typical bandwidth limitations.
  - We are researching RAID to find an efficient solution to the backup problem. We will have to weigh in the bandwidth limitations in our system.
  - Nodes in the distributed system can leave at any time. The data that they held will have to be replicated on more nodes in order to ensure the best reliability.

TERMINOLOGY

- Cloud: All the active nodes in the distributed system.
- Node: An participating client in the *cloud*. Since a node is a client, the node may be active or inactive.
- Originating Client: Client submitting a file for backup.
- Division: One of the pieces of a partitioned file.
- Division Group: Group of nodes holding a content equivalent division.
- Pastry: A substrate for peer-to-peer applications.

APPROACH

- File partitioning should happen on the originating client. Hash values of each division are calculated (like a checksum) for data integrity.
- Communication with the *cloud* will result in a list of nodes to upload the divisions.
- Minimize the amount of network traffic the originating client needs to perform. Try to hide all the expensive bandwidth operations between nodes in the cloud.

APPROACH CONT.

- We envision multiple *Master Nodes* that know the details concerning which nodes store which divisions of the original file.
- Maintenance of redundancy of file divisions across the *cloud* by way of heartbeat (keep-alive) communication between *Division Group Nodes*. Likewise there is communication between Masters.

FREE PASTRY

- Inserting and Accessing key value pairs for any Object.
- Availability and Persistence guaranteed.
- Handles Routing efficiently.
- No support for handling malicious nodes.

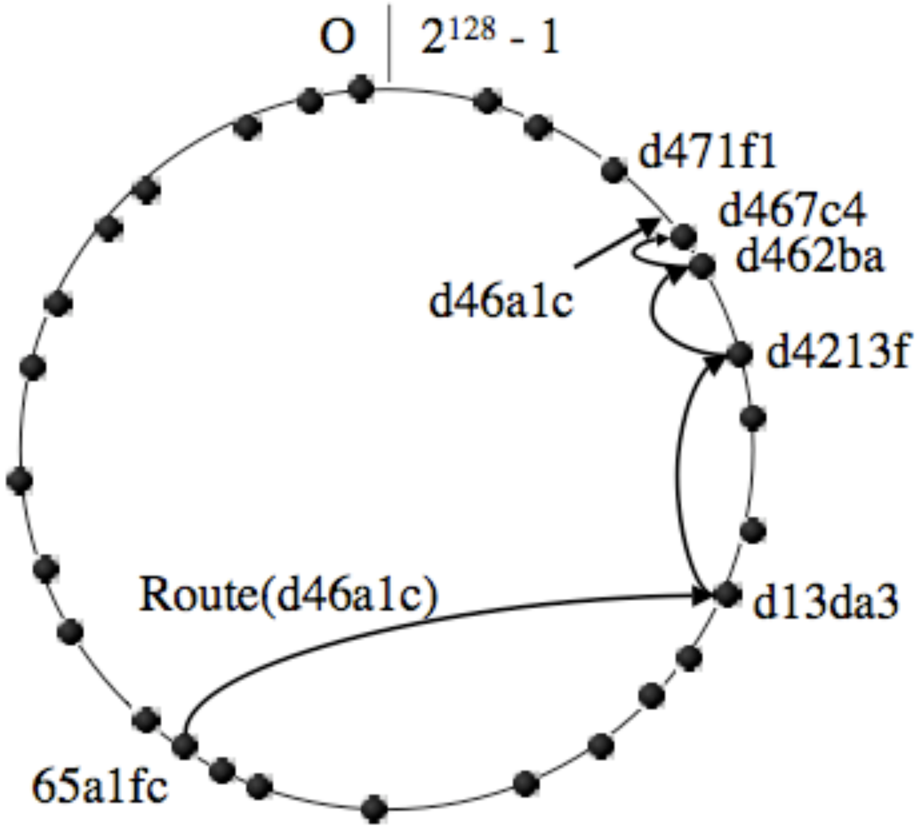

Figure 2: Routing a message from node $65a1fc$ with key $d46a1c$. The dots depict live nodes in Pastry's circular namespace.

SOFTWARE DELIVERABLE

- A client application that participates in the distributed backup system. The application will:
  - Allow for submitting backup requests and retrieving previous stored backups.
  - Participate in the *Cloud* by storing divisions of other member's backups and other chores.
- An integrity protocol and trust model that ensures good behavior within the *Cloud*.

FUTURE WORK

- Encrypted communication and storage for obvious reasons.
- Fully featured invitation and authentication protocol to ensure trustworthy nodes in the *Cloud*.
- Fully implement and test Erasure codes in Pastry.

PAPER #1 - PASTRY

- A. Rowstron and P. Druschel, "*Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems*". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.
- http://research.microsoft.com/en-us/um/people/antr/PAST/pastry.pdf
- Understand how to leverage a Distributed Hash Table in a large Peer-to-Peer application.

PAPER #2 - ERASURE

- Goodson, G.R.; Wylie, J.J.; Ganger, G.R.; Reiter, M.K., "*Efficient Byzantine-tolerant erasure-coded storage,*" Dependable Systems and Networks, 2004 International Conference on , vol., no., pp. 135-144, 28 June-1 July 2004
- http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1311884&isnumber=29105
- Decentralized consistency protocol for survivable storage.

PAPER #3 - PAST

- P. Druschel and A. Rowstron, "*PAST: A large-scale, persistent peer-to-peer storage utility*", HotOS VIII, Schoss Elmau, Germany, May 2001.
- http://research.microsoft.com/en-us/um/people/antr/PAST/hotos.pdf
- Generic file replication and caching in Pastry.

EXTRA PAPER - TRUST

- Aberer, K. and Despotovic, Z. 2001. *Managing trust in a peer-2-peer information system*. In Proceedings of the Tenth international Conference on information and Knowledge Management (Atlanta, Georgia, USA, October 05 - 10, 2001). H. Paques, L. Liu, and D. Grossman, Eds. CIKM '01. ACM, New York, NY.
- http://lsirpeople.epfl.ch/despotovic/CIKM2001-trust.pdf
- Trust Building within a Peer-to-Peer System.

LINKS

- Team Website: http://www.cs.rit.edu/~jjp1820/distributed/
- FreePastry: http://www.freepastry.org/