The Infinit File System

Julien Quintard

UNIVERSITY OF CAMBRIDGE
Computer Laboratory
Systems Research Group

Motivation

Computer networks are rapidly growing in importance as a medium for the storage and exchange of information. Advanced storage systems are required for enabling people to reliably store and backup their files but also share those files with other users. Such systems should provide the following fundamental properties:

- **Storage** - files should be accessible anywhere, anytime until they are deleted by their owners;
- **Sharing** - people should be able to share their files as easily as through peer-to-peer file sharing applications;
- **Security** - users should keep control over who is allowed to access their files;
- **Semantics** - since end-users are familiar with hierarchical organisations, such network storage systems should take the form of file systems.

Unfortunately, neither file systems — being centralised, version-based and/or distributed — such as *Elephant*, *NFS*, *OceanStore etc.* nor company-owned data centre-based storage systems such as *Amazon S3* nor file sharing applications such as *eDonkey*, *Bittorent*, *Freenet etc.* provide the required properties.

Overview

Infinit is a peer-to-peer file system that provides users the ability to store their files in a secure and reliable way.

The peer-to-peer architecture ensures that no administrative entity has control over the whole system while distributing the storage and bandwidth load amongst nodes. The *Infinit* design guarantees users a fine grain control over their files allowing them to share those files with individuals and/or groups of friends *etc.*

Challenges

- **Overlay Network**: routing a message should succeed even in the presence of a large portion of misbehaving nodes;
- **Naming**: users should be able to virtually name and organise the hierarchy in their preferred way;
- **Reliability**: users should be able to access the exact files they stored in the system, from anywhere, anytime;
- **Security**: users should be able to easily specify who is allowed to access their objects, providing an easy way of sharing;
- **Privacy**: nobody should know what objects a user is actually accessing and/or locally storing;
- **Manageability**: users should be able, together, to make global decisions about the file system: files organisation, users restrictions *etc.*

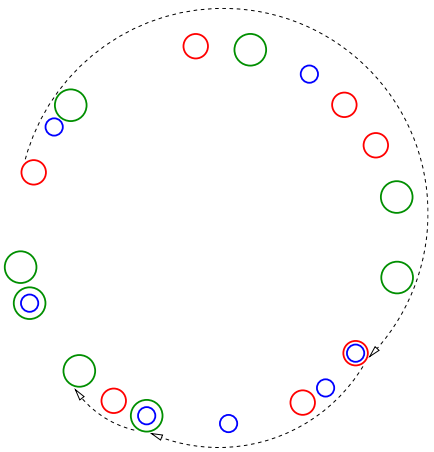
Below are detailed some preliminary ideas regarding the challenges listed above.

Overlay Network

Infinit is likely to be layered on top of a *Distributed Hash Table*, which will itself be layered on top of an *Overlay Network*.

Since structured overlay networks such as *Chord*, *Pastry*, *CAN etc.* suffer from their incapacity to handle byzantine nodes, new approaches need to be explored.

One of those approaches consists in establishing overlay connections between trusted nodes forming multiple, possibly overlapping, small worlds. The routing process would therefore consist in traversing the small worlds until one containing the destination node is reached.



A routing process taking place between three groups or small worlds

Unfortunately, such an approach will suffer from the following issues: how to represent and dynamically adjust trust relationships, how to provide routing guarantees, how to prevent hotspots, where to store blocks *etc.*

Reliability

Reliability will be achieved through widespread use of replication so that the system guarantees, with very high probability, that there is always one living and up-to-date replica controlled by a non-byzantine node.

However, the replication strategy, protocols and algorithms remain to be defined.

Security

Security encapsulates both access control and sharing since an easy access control scheme would provide users the necessary tool for sharing. Access control will be achieved through the use of advanced cryptographic techniques including convergent encryption *etc.*, ensuring protection of file system objects from unauthorised read/write accesses while blocks are replicated and stored on multiple potentially untrusted nodes.

Last but not least, *Infinit* will support both hierarchical groups and roles so that access control can be more easily specified and managed by end-users.

Privacy

Users care about confidentiality and do not want either governments or other users to know what files they are accessing. *Infinit* will provide confidentiality by making replication and access undistinguishable by nodes providing data.

Manageability

Such a large-scale peer-to-peer storage system has to provide support for enabling some sort of management tasks at least for resolving update conflicts and controlling who has the right to write in the root directory.

Since *Infinit* relies on a peer-to-peer architecture *i.e.* without any administrative entity, the management feature must be completely decentralised while preventing single users from taking drastic decisions on their own.

Providing manageability in peer-to-peer networks therefore sounds contradictory if not impossible. *Infinit* will try to provide such a feature through the use of voting schemes.

julien.quintard@cl.cam.ac.uk