

barnfs will be awesome

Timothy G. Abbott*[†] Eric C. Price[‡] Gregory N. Price[§] ...

Abstract

This paper presents barnfs.

1 Introduction

Use naming like AFS, with DNSSec to solve SFS's problem.

2 Modern design

2.1 Use FUSE

portability, stability, efficiency. Not about reinventing the wheel.

2.2 Use SCTP

It's 2008, people shouldn't have to write their own hackish thing over UDP.

2.3 Use Haskell

It's 2008, people shouldn't write their filesystems in C.

Security. Checkability. Functionality. Look at <http://www.seas.upenn.edu/~lipeng/homepage/unify.html>

2.4 Use MySQL Clustering for Replication?

We should investigate this more.

3 Tree Locks and Prefetching

Argue tree locks good general primitive.

Argue soft prefetching into cache good for many cases important for running home directories in file system (not just `ls -l`, but also `find`, `tar`, `untar`, `make`, `svn`, ...).

Discuss trial implementation of this feature in YFS. Maybe a graph?

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA, tabbott,price@mit.edu

[†]Partially supported by an NSF Graduate Research Fellowship and an MIT-Akamai Presidential Fellowship.

[‡]MIT ecprice@mit.edu

[§]Partially supported by an NSF Graduate Research Fellowship

4 Security Model

4.1 PKI

Discuss Kerberos and its limitations; victory of Web over AFS. Plan for cross-realm to Just Work with no action by administrators.

4.2 Authorization

Support delegation of credentials.

5 Replication at all layers

6 Related Work

NFS

AFS

SFS

Ceph

more?

7 Conclusion

barnfs will rock. It is not implemented yet.

8 Acknowledgements

SIPB? 6.824?

References.