Arla—a really likeable AFS-client

Johan Danielsson
Parallelldatorcentrum, KTH
joda@pdc.kth.se

Assar Westerlund
Swedish Institute of Computer Science
assar@sics.se

Arla—a really likeable AFS-client

free

portable

efficient

supports disconnected operation and encryption of the data stream

What is AFS?

a world-wide distributed file system

also known as Andrew File System

originally developed at CMU

later commercialised by Transarc

currently 150 public cells around the world

AFS (cont.)

files are stored at dedicated file servers

untrusted clients (cache managers) cache files on local disk

clients have to prove themselves to servers

Consistency in AFS

when retrieving a file, the client gets a promise that the server will notify it before changing the file

the notification is called a *callback*

allows the client to read cached files without any network activity

Implementation of Arla

- a kernel module (*xfs*)
- a user-level daemon (*arlad*)

these communicate by sending messages over a character device

Advantages

easier development

better portability

better development tools

possible to use normal libraries

Disadvantages

performance?

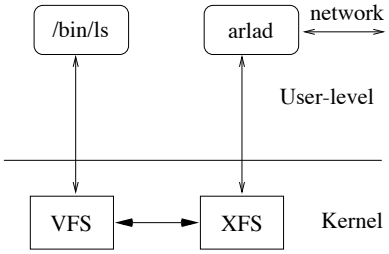
Kernel module (*xfs*)

small, $\sim$5000 lines, 32 KB on i386

implements

- a system call
- a character device
- a virtual file system

Implementation (cont.)



Performance

	UFS cold	Arla cold	Arla warm
(I) mkdir	2	2	3
(II) cp	6	18	11
(III) r.stat	2	2	3
(IV) r.grep	5	5	5
(V) compile	34	36	34

(elapsed time in seconds of Andrew Benchmark on a ThinkPad 560)

Portability

$\sim$10% operating system dependent

lwp requires machine-specific context switch code

daemon runs on any Unix-like system

works on NT with cygwin32

Portability (cont.)

Kernel modules for:

- SunOS
- Solaris
- NetBSD, FreeBSD, OpenBSD
- Linux
- AIX
- HP-UX
- Digital Unix

Security and Encryption

- Kerberos 4-based (rxkad)
- implementation written outside US
- supports encryption of all data

Disconnected operation

- allow file system operations without network connectivity
- read from already cached data
- write operations to a log and replay later

Disconnected operation (cont.)

	fetch on miss	no fetch on miss
consistent	connected	N/A
not consistent	fetch-only	disconnected

Future work

- more performance
- more platforms
- more disconnected operations
- file + database servers

Acknowledgments

It's all Björn Grönvall's fault

Magnus Ahltorp
Robert Burgess
Artur Grabowski
Love Hörnquist-Åstrand

and lots of other people (see `THANKS`)

Availability

`http://www.stacken.kth.se/projekt/arla/`