

OFS: Ein allgemeines Offline-Dateisystem auf Basis von FUSE

Tobias Jähnel und Peter Trommler

Fakultät Informatik

Georg-Simon-Ohm-Hochschule

Nürnberg

Übersicht

- Hintergrund
- Caching und Offline Fähigkeit
- Umsetzung mit FUSE
- Konfliktlösung
- Vorführung

Hintergrund

- Laptop mit Linux
- Dateien liegen auf CIFS und Novell server
- Windows und CIFS funktioniert leidlich
- AFS, Coda, Intermezzo, ...
- Vorstudie: Diplomarbeit
- Projekt Ohm File System: Jähnel, Gsellmann, Kolassa
- Open Source Projekt „offlinefs“

Anforderungen

- Unabhängigkeit vom Netzwerk-Dateisystem
- Keine Modifikation an den Servern
- Mindestens so komfortabel wie Windows
 - Weiterarbeiten bei Abbruch der Netzwerkverbindung
 - Weiterarbeiten bei nicht verfügbarem Dateiserver
- Eingriff in den Kernel vermeiden

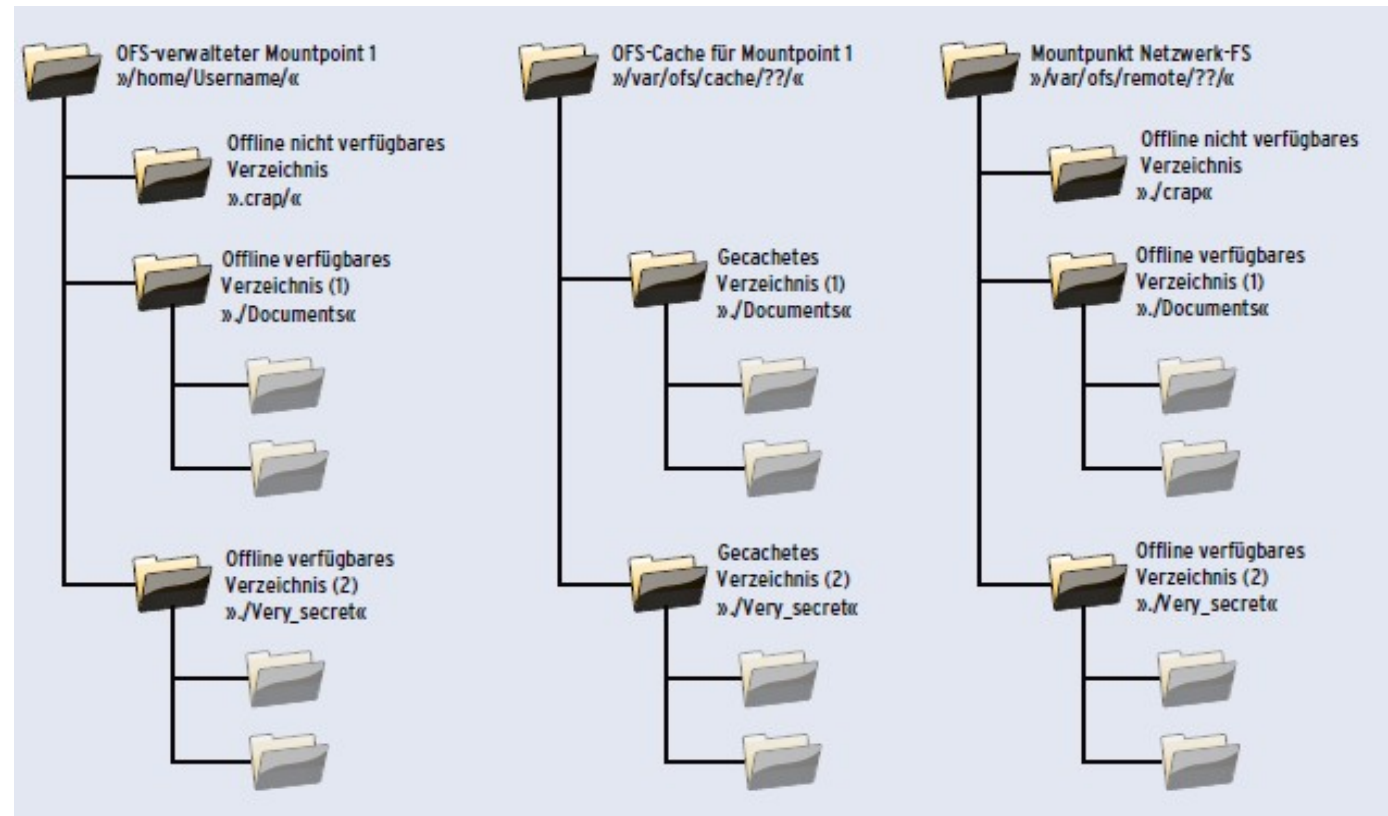
Randbedingungen

- große Festplatten in Laptops verfügbar
- Konflikte selten
 - Homedirectory, nur Eigentümer ändert Dateien, aber Desktoprechner!
 - Lesender Zugriff auf gemeinsame Dateien (Vorlagen, Software, ...)
 - Keine gemeinsame Entwicklung auf einem Verzeichnis! (verwende: cvs, svn, ...)

Offline-Fähigkeit

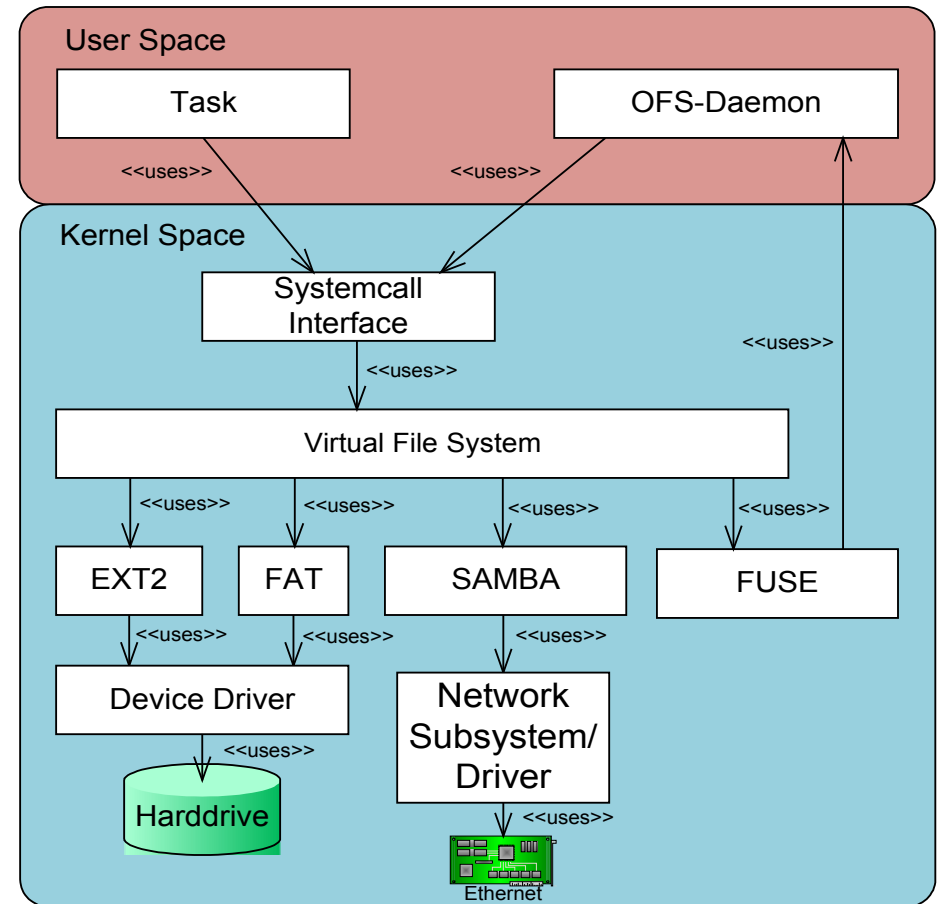
■ drei Verzeichnisse

- Verzeichnis des Benutzers
- Schattenkopie auf dem Client
- Dateisystem des Servers



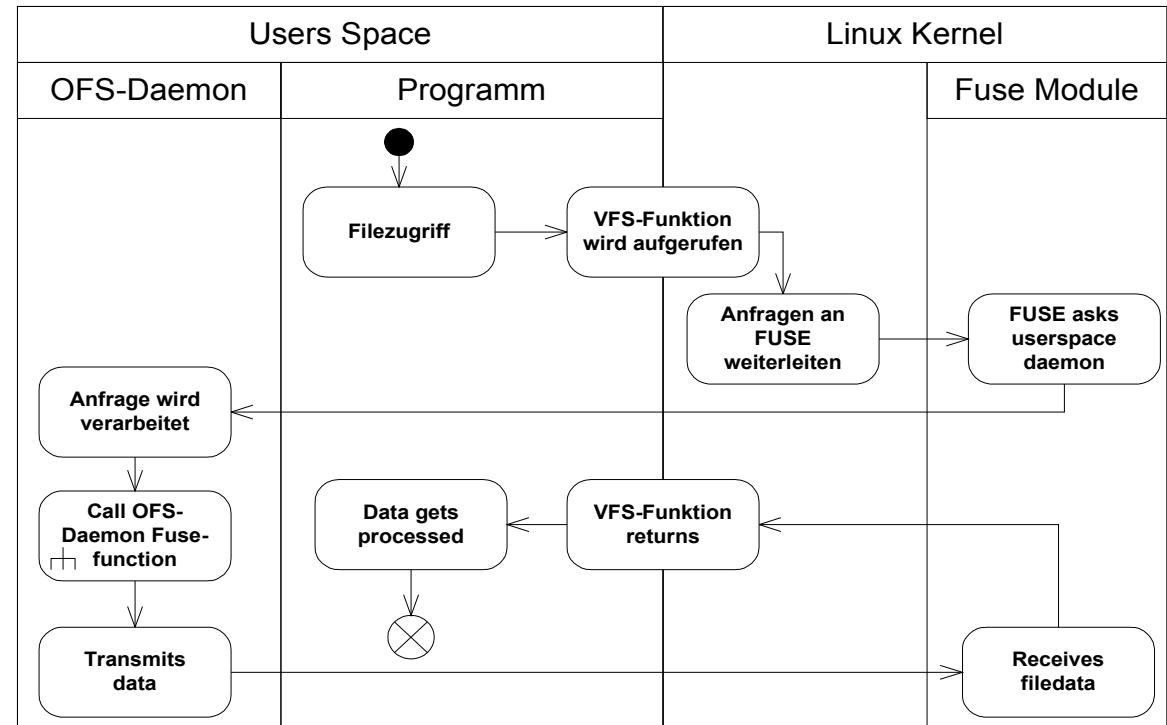
Interaktion FUSE und OFS-Daemon

- Anwendungen (Prozesse) verwenden das *System Call Interface*
- Filesystemzugriff auf OFS Mountpoint
- VFS verwendet Dateisystem Implementationen
- FUSE verwendet OFS-Daemon
- OFS-Daemon verwendet wieder das *System Call Interface*
- Nun Zugriff auf mounted remote share oder/und Schattenkopie



Ablauf eines Dateizugriffs

- Programm greift auf Datei zu
- VFS leitet an FUSE weiter
- FUSE leitet an OFS-Daemon
- OFS Daemon bearbeitet Zugriff
 - mounted remote share/Schattenkopie
 - erneuter VFS Zugriff
- Antwort des OFS-Daemon
 - umgekehrte Reihenfolge
 - an FUSE und
 - über VFS an Programm



Interaktion mit OFS

■ Erweiterte Datei-Attribute

- `ofs.offline`: Datei soll offline verfügbar sein.
- `ofs.available`: Remote Datei ist erreichbar und wird verwendet.

■ Attribute nicht auf dem Server gespeichert

■ Attribute virtuell

- nur Zustandsinformation
- `available` vererbt von Mountpoint (root des mount)
- `offline` als Liste in Konfigurationsdatei

Synchronisation

- Aktualisierung der Schattenkopie vor Trennung
- Im offline Zustand: Änderungen werden mitgeschrieben
- Bei Verbindung zum Server:
 1. Durchführung der Änderungen auf dem Server (Reintegration)
 2. Aktualisierung der Schattenkopie
- Konflikterkennung anhand der Änderungszeit

Konfliktlösung

- Remote Datei und Datei im Cache geändert
- Cache hält Schreib-Log
- Gemeinsamer Vorfahr rekonstruierbar
- Textdateien: diff3
- Andere Handler analog MIME types
 - Drei-Wege diff
 - Zwei-Wege diff

Open Source Projekt

- Sourceforge: <http://offlinefs.sourceforge.net>
- Leitung: Tobias Jähnel
- Neue Projektarbeit an Ohm-Hochschule
- Wir freuen uns über Euere Beiträge!

Demo