S T U D I E N / D I P L O M A R B E I T

Protokoll-Vereinfachungen und Netzwerk-Locking für `omirr`

In einer vorhergehenden Diplomarbeit wurde `omirr` entwickelt, mit dem sich Verzeichnishierarchien über das Internet spiegeln lassen, und zwar in symmetrischer Weise, d.h. es gibt keine ausgezeichnete Master-Kopie der Daten, und jeder Teilnehmer darf die ständig online gespiegelten Daten gleichberechtigt ändern. Eine frühe Version dieses Systems befindet sich auf `ftp://ftp.isa.de/pub/home/luik/omirr`; die neueste Version ist beim Betreuer einsehbar.

Das Kommunikationsprotokoll zwischen dem Kernel und dem Daemon einerseits und zwischen Daemonen auf verschiedenen Hosts andererseits soll vereinfacht werden. Grundlage hierfür ist die neue, bereits implementierte Kernel-Schnittstelle für `omirr` (siehe `ftp://ftp.informatik.uni-stuttgart.de/pub/Linux/schoebel/` auf Basis der zu diesem Zweck erfundenen `dentries`, bei der die vollständigen absoluten Pfadnamen bereits im Kernel berechnet werden, so daß dieser Aufwand nicht mehr in `omirrd` getrieben werden muß. Durch diese Vereinfachung soll der Code kleiner und schneller werden.

Als neue Fähigkeit soll das transparente verteilte Locking (Systemaufruf `flock()`) zusätzlich implementiert werden. Dazu soll das `omirrd`-Protokoll um entsprechende Lock-Anweisungen erweitert werden.

Zusatzaufgabe bei Diplomarbeit:

Das Locking soll *ausfallsicher* implementiert werden, indem ein neuartiges Votierungs-Verfahren (Mehrheits-Entscheidungs-Verfahren) eingesetzt wird, das vom Betreuer entwickelt wurde. Die Ausfallsicherheit bezieht sich sowohl auf unterbrochene Netz-Verbindungen als auch auf Knotenausfälle.

Weiterhin soll optional *implizites Locking* möglich sein, das bezüglich einzelner `read()` und `write()`-Operationen dieselbe Semantik simuliert, als wäre ein einziges Dateisystem auf einem einzigen Rechner vorhanden. Dazu muß jede einzelne Zugriffsoperation mitprotokolliert werden, so daß inkrementelle Updates stattfinden. Zusätzlich ist die bereits implementierte `flock()`-Funktionalität zur Simulation von Mutex-Bereichen zu nutzen.

Das Ergebnis der Arbeit soll unter GPL veröffentlicht werden.

Interessenten wenden sich an Thomas Schöbel-Theuer, Zi. 1.133