

DROPCAN

Co to jest:

- zestaw programów klient-serwer służący do tworzenia lokalnych kopii plików z danego folderu
- zachowywanie archiwalnych plików i możliwość ich przywrócenia przez program kliencki
- utrzymanie w katalogach obsługiwanych przez programy klienckie spójnej wersji plików

Problemy:

- jak utrzymać maksymalną wydajność dostępu do zasobów wspólnych
- jak zsynchronizować wszystkie programy klienckie tak, żeby nie "uploadowały" równocześnie zmian
- jak wymusić spójność danych podczas przywracania archiwalnej wersji

Jak to jest zrobione:

1. pamięć współdzielona z informacjami o bieżących wersjach plików
2. kolejka i semaforey POSIX (pewna adaptacja problemu czytelników i pisarzy, patrz ad2.)
3. semaforey dla poszczególnych plików tworzone w runtime (blokujące na czas modyfikacji)
4. folder na zmiany oczekujące, patrz ad3.
5. serwer utrzymuje strukturę synchronizowanego katalogu z wykorzystaniem dowiązań twardych do wybranych wersji pliku
6. klient skanuje swój folder w poszukiwaniu zmian
7. klient skanuje współdzieloną tablicę w poszukiwaniu zmian
8. pliki znaczników czasowych po stronie klienta odpowiadające za możliwość zmian
9. przywracanie wersji powoduje blokadę danego pliku aż do czasu zakończenia procesu przywracania (utrzymujemy spójny stan)

Szczegóły

ad1. przechowujemy aktualną zawartość serwera w postaci listy plików wraz ze znacznikami o aktywności pliku(czy nie został usunięty), jego względnej ścieżce oraz czas modyfikacji

ad2. dostęp do zasobu tablicy jest organizowany na zasadzie serwer=pisarz, klient=czytelnik; taka organizacja wynika z modelu współpracy - serwer zarządza aktualną wersją i jako jedyny modyfikuje tablicę, klient odczytuje jej zawartość w poszukiwaniu zmian do pobrania;

OPTYMALNA WYDAJNOŚĆ

ad3. gdy klient zechce zgłosić zmianę (i jest uprawniony) blokuje modyfikację danego pliku - generowany jest i blokowany semafor dla tego pliku (aby nikt inny nie zmieniał pliku); ów semafor odblokowywany jest przez serwer po pomyślnym przetworzeniu zgłoszenia zmiany; blokujemy jedynie wybrane pliki - OPTYMALNA WYDAJNOŚĆ

ad4. ponieważ tylko jedna wersja pliku może być zgłaszana równocześnie do zmiany

modyfikowane wersje umieszczane są w folderze oczekujących zmian który skanuje serwer; uploadowane są pliki z zachowaniem wszelkich informacji dotyczących czasów modyfikacji itp.

ad5. serwer w ukrytych katalogach przetrzymuje wersje archiwalne tymczasem odwzorowując strukturę katalogu za pomocą dowiązań twardych - zapobiega to mnożeniu plików i gwarantuje dostęp do właściwych wersji przez programy klienckie

ad8. klient w momencie modyfikacji pliku w swoim katalogu tworzy plik posiadający w nazwie znacznik czasowy - dopóki ów znacznik nie będzie zgodny z wersją serwera klient musi wstrzymać się z wysyłaniem modyfikacji; w ten sposób klient nie wyśle nieaktualnej wersji pliku (poprzedniej) ani nie nadpisze zmiany wysyłanej przez inny program kliencki

ad9. na czas wybierania wersji należy "zamrozić" dany plik, aby nikt nie zmodyfikował go mimo naszej wiedzy w czasie między odebraniem listy dostępnych wersji (klient skanuje odpowiednie struktury) a wyznaczeniem wersji aktualnej; wyznaczenie aktualnej to skopiowanie wybranej jako nowa (nowy znacznik czasowy) - w ten sposób mamy pełną historię zmian z wyróżnieniem wszelkich wersji (można dowiedzieć się o wersji, która była przywróceniem archiwalnej)

Ograniczenia ("jak na spowiedzi"):

- nie może wystąpić katalog ".committed" w katalogu synchronizowanym
- nie mogą wystąpić pliki "...[napis]" w katalogu synchronizowanym
- puste katalogi nie są synchronizowane (jako, że nie niosą informacji)
- ścieżki względne nie mogą mieć więcej niż 256 znaków (można zmienić makro)
- po uruchomieniu ponownym pierwszy uruchamiający się klient ustala aktualną wersję - archiwalne pozostają bez zmian ale klient bez swojej wersji wymaże aktualną wersję