

Git - Rozproszony System Kontroli Wersji

Bartosz Borkowski

Systemy Rozproszone

29 października 2009

Agenda

- Historia
- Specyfikacja
- Implementacja
- Błędy i niedociągnięcia
- Podsumowanie

Słownik

Git

- push
- pull
- clone

SVN

- commit
- update
- checkout

Wspólne pojęcia

- branch
- merge

Historia

- 1972 - SCCS
- 1980s - RCS
- 1990 - CVS
- 2000 - SVN i BitKeeper
- 2005 - Git i Mercurial

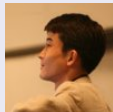
Twórcy

Linus Torvalds



Pomysłodawca

Junio Hamano



Obecny zarządca

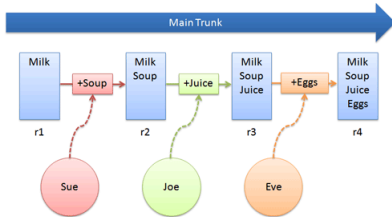
Uzasadnienie biznesowe

- Zmiana polityki BitMover
- Nikt nie chciał powracać do tarball'i i patch'ów
- Brak dostępnych rozwiązań:
 - rozproszonych
 - niezawodnych
 - wydajnych

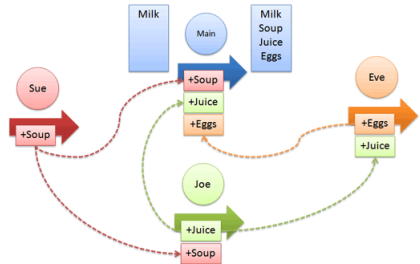
Inspiracja

- Tarball'e i patch'e
- BitKeeper jako przykład pozytywny
- CVS jako przykład negatywny
 - "WWCVSND" - What Would CVS Not Do?

Centralized VCS



Distributed VCS



<http://betterexplained.com/articles/intro-to-distributed-version-control-illustrated>

Branching

- Szybki i łatwy w stworzeniu
- Tani w utrzymaniu
- Lokalny

Prawa do commit'owania

Problem

Ludzie są:

- nierzetelni
- roztrzepani
- złośliwi
- niekompetentni

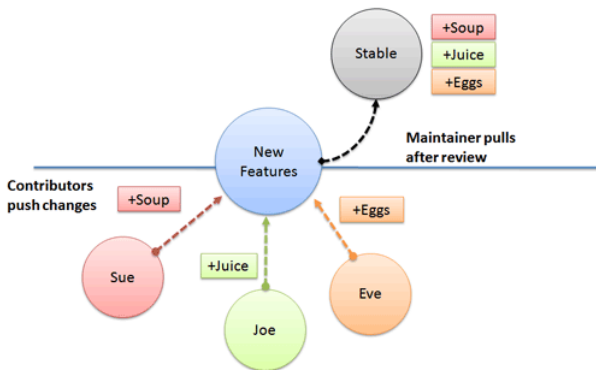
Jak ograniczyć dostęp takich ludzi do repozytorium?

Rozwiązanie

- Wprowadzić mnóstwo ograniczeń i złej atmosfery do projektu.
- Przejść na model rozproszony.

Sieć zaufania

Distributed Push/Pull Model



<http://betterexplained.com/articles/intro-to-distributed-version-control-illustrated>

Wpływ na zarządzanie

- Uproszczenie przestrzeni nazw
- Synchronizacja pracy
- Ułatwienie pracy wewnątrz zespołów
- Delegowanie pracy

Wydajność

SVN

- Tworzenie nowych gałęzi w czasie $\mathcal{O}(1)$
- Nowa gałąź początkowo nie zmienia rozmiaru repozytorium
- Utrzymanie gałęzi w zamortyzowanym czasie stałym

Git

- Utworzenie gałęzi to stworzenie 41 bajtowego pliku
- Utrzymanie gałęzi podobne do rozwiązań z SVN

Scalanie w SVN

Rozwiązanie

- Stwórz diff z dwóch gałęzi
- Zaaplikuj zmiany do jednej z nich
- Wprowadź zmienioną gałąź do repozytorium jako jeden commit

Problemy

Użytkownik musi

- planować z tygodniowym wyprzedzeniem
- zarezerwować na operację jeden dzień
- pożegnać się z historią zmian

Scalanie w Git

Rozwiązanie

- Znajdź wspólny węzeł w historii
- Scal od tego węzła
- Wprowadź do repozytorium nowy węzeł z dwoma rodzicami
- Stwórz plik diff stat

Wydajność

Różnica jakościowa

- Podczas prac nad jądrem Linuksa osiągnięto średnio 4.5 scalenia na dzień
- Jedno scalenie trwało ok. 3 sekund

Zmiana mentalności

Robiąc gałęzie i scalenia częściej, działając na mniejszych fragmentach kodu, unikamy bardzo poważnych problemów.

Śledzenie zawartości

Kolekcja

Git śledzi kolekcję plików, a nie pojedyncze pliki

Konsekwencje

Duże repozytoria będą działać wolno

Rozwiązanie

Repozytorium zawierające wskaźniki do repozytoriów.

Wydajność

Porównanie wielkości repozytorium dla projektu KDE

- CVS: ~4GB
- SVN: ~12GB
- Git: ~1.5GB

SHA1

- Jest funkcją skrótu
- Opracowany w 1995 przez NSA i NIST
- Zastąpił SHA0
- Zastąpiony przez rodzinę SHA2
- Był częścią DSA

SHA1 a Git

- Git liczy sumy kontrolne dla danych
- Git sprawdza sumy kontrolne dla danych
- SHA1 nie jest mechanizmem bezpieczeństwa, lecz weryfikacji

Standardowe rozwiązania

Metody wersjonowania

- Tekst
- Splot
- Delt

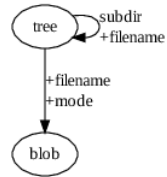
Delt

- Tylko najnowsza wersja przechowywana jest w całości
- Pozostałe reprezentowane są przez delty względem swoich następców
- Ulepszenia: kombinacje delt i skip-delt

Rozwiązania Git'a

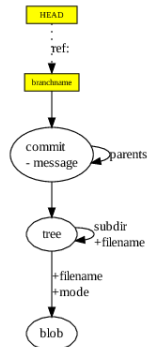
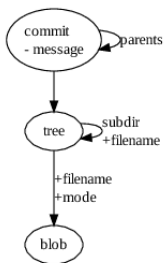
- Deltę liczone są między wybranymi obiektami
- Obiekt wybiera się heurystycznie
- Otrzymujemy grafy acykliczne delt zamiast łańcucha
- Wagami w grafie są rozmiary delt

Blob i Tree



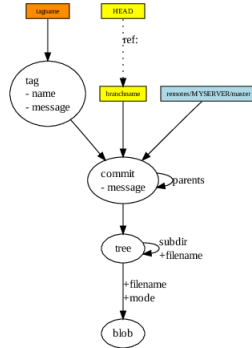
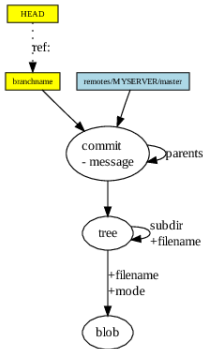
<http://eagain.net/articles/git-for-computer-scientists>

Commit i Ref



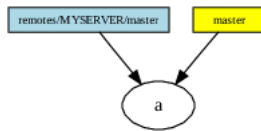
<http://eagain.net/articles/git-for-computer-scientists>

RemoteRef i Tag



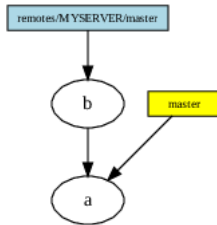
<http://eagain.net/articles/git-for-computer-scientists>

Szybkie scalanie



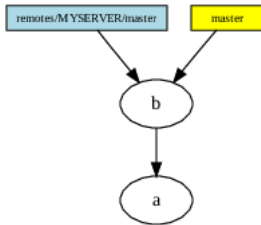
<http://eagain.net/articles/git-for-computer-scientists>

Szybkie scalanie



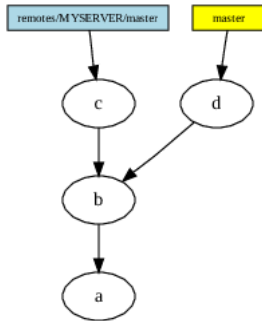
<http://eagain.net/articles/git-for-computer-scientists>

Szybkie scalanie



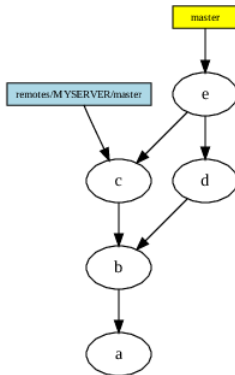
<http://eagain.net/articles/git-for-computer-scientists>

Scalanie



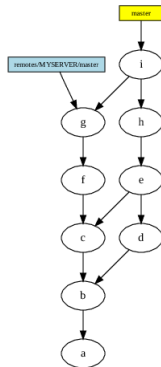
<http://eagain.net/articles/git-for-computer-scientists>

Scalanie



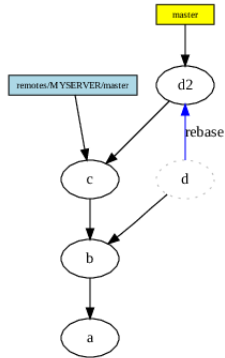
<http://eagain.net/articles/git-for-computer-scientists>

Scalanie



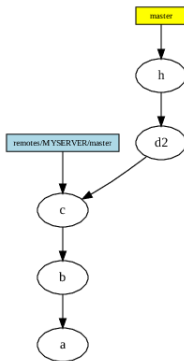
<http://eagain.net/articles/git-for-computer-scientists>

Przebazowanie



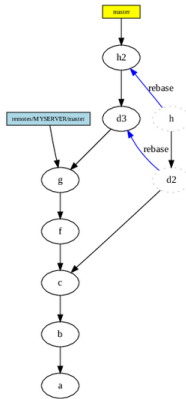
<http://eagain.net/articles/git-for-computer-scientists>

Przebazowanie



<http://eagain.net/articles/git-for-computer-scientists>

Przebazowanie



<http://eagain.net/articles/git-for-computer-scientists>

Ograniczenia

- Stopień trudności
- Nie ma "najnowszej" wersji
- Kłopoty w środowisku Windows
- SHA1 przestaje być bezpieczny

Podsumowanie

- Rozproszony jest lepszy!
- Git jest bardzo dobrym narzędziem
- Jest jednak TYLKO narzędziem

Dziękuję

Dziękuję za uwagę.

Bibliografia

- Linus Torvalds on Git:
<http://www.youtube.com/watch?v=4XpnKHJAok8>
- Git for Computer Scientists:
<http://eagain.net/articles/git-for-computer-scientists>
- Intro to Distributed Version Control (Illustrated):
<http://betterexplained.com/articles/intro-to-distributed-version-control-illustrated>
- DVCSAnalysis:
<http://code.google.com/p/support/wiki/DVCSAnalysis>

Bibliografia

- File versioning:
<http://web.mit.edu/ghudson/thoughts/file-versioning>
- SVNBook:
<http://svnbook.red-bean.com/en/1.1/index.html>
- Opis algorytmu SHA1:
http://e-handel.mm.com.pl/crypto/opis_algorytmu_sha.htm
- Cryptoanalysis of SHA1:
http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html