



gitって最近どうなってるの？

OSC2005-Fall出張カーネル読書会
2005年9月17日

上川 純一



はじめに

- 「SCMって何？」
 - Source Code Management とか Source Control Management とか言われているバージョン管理ツールとか呼ばれているもの
 - Supply Chain Managementではない
- 「gitって何？」
 - 2005年4月からLinuxの開発に利用しているSCM
 - Linusは最初はtarballで管理
 - その後、BitKeeperを使っていた



なぜgitか

- 既存のツールでカーネルを扱うのが難しい
 - Subversionは中央集権型
 - CVSはLinusは昔から嫌い
 - archは遅い
 - monotoneも遅い
 - カーネルは大きすぎる
- 1ファイル1バージョン
 - 不要なデータをキャッシュしない
- → カーネル開発者・ファイルシステム開発者の視点で設計している



開発状況 2005年9月16日

- タイムライン
 - 4月7日ころ： Linusがgit 0.02をリリース。
チェックアウトとコミットだけができるようなバージョン
 - 4月21日ころ： 初のgitベースのLinuxカーネルリリース
 - cogitoが主として利用される
 - 7月： packed formatが導入され、それに追隨していないcogitoが動かない時期が続く
 - 7月末： メンテナンスがLinusからJunio C Hamanoに移管
 - 9月中旬： git-coreのコマンド名がまたごっそり変更



packedフォーマットの登場

- sha-1 ベースでBLOBをそれぞれ1ファイルにしていた
- ファイルシステムにやさしくない
 - 各ファイルの各バージョンでそれぞれi-nodeを消費する
- ファイル転送にやさしくない
 - httpで転送する際に、細かいファイルをたどるよりも大きなファイルをごっそり取得するほうが効率よい



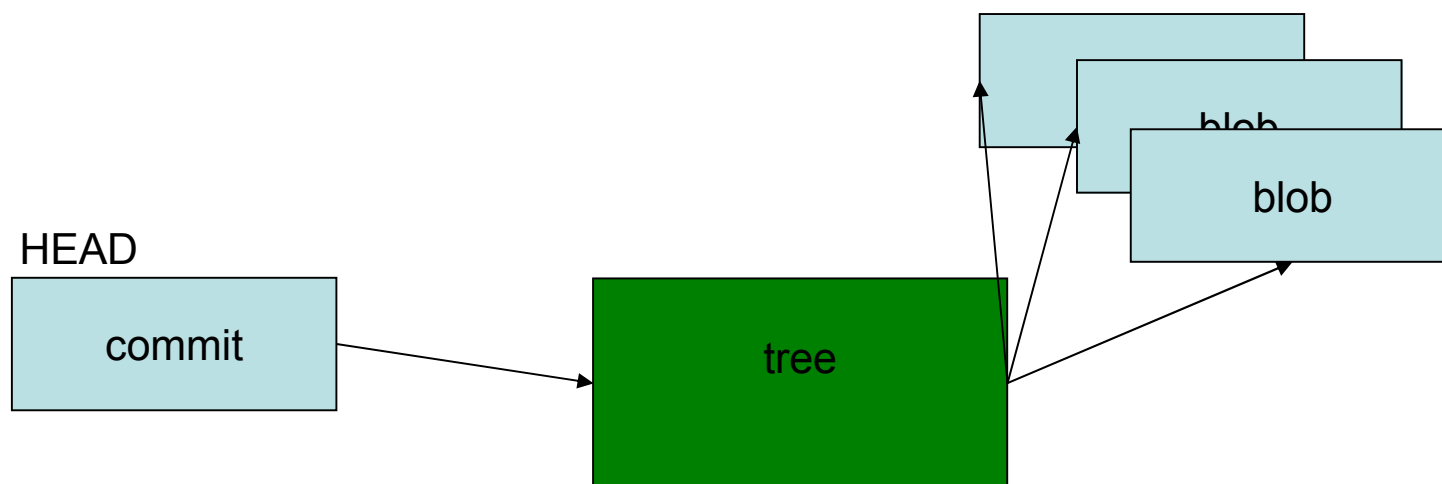
gitはそれだけでも使えるように

- 当初は「cogitoがインタフェースで、gitがバックエンドのデータベース」みたいな雰囲気だった
- 最近は「gitだけで使えるように」という雰囲気になりつつある？
- cogitoなどの問題点：
 - cogitoなどのインタフェースがgitの開発とインタフェースの変更同期できていない
 - Linusは気にせずカーネルのレポジトリを新しいフォーマットに変更してしまったりする
 - 非互換になってしまったから一週間放置されていたら開発が続けられない



gitでのチェックアウトメカニズム

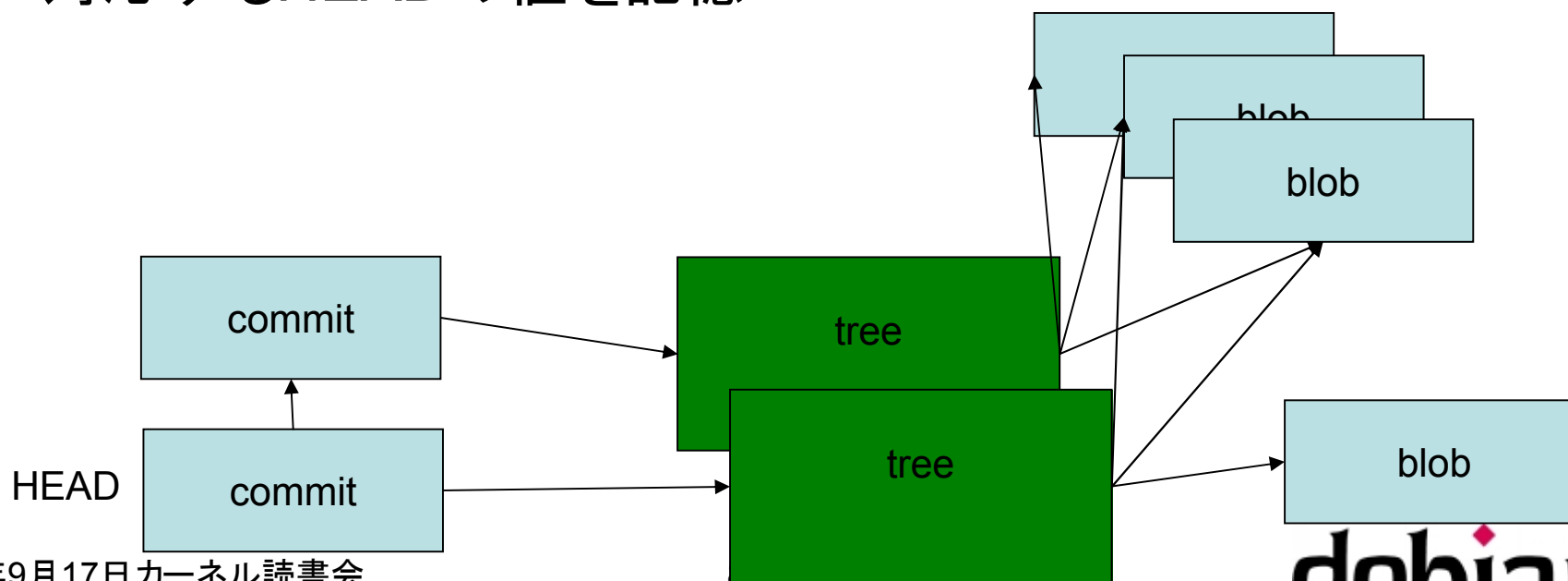
- commit objectの情報からたどり、treeの情報を取得、blobを展開





gitでのコミットメカニズム

- tree objectを新規に作成
 - write-tree
- commit objectを新規に作成
 - commit-tree ツリー名 -p 以前のHEAD名
- 対応するHEADの値を記憶





git : データ構造

- オブジェクト
 - 配置 `:.git/objects/xx/`以下に内容物のSHA-1ハッシュの16進表記に基づいたファイル名で配置される
 - 保存形式 : zlibを利用してDeflateして保管
- blob
 - 管理したいデータ。実際のファイルに相当する。
- tree
 - ある瞬間のディレクトリ構成に対応、その瞬間にそのディレクトリ構成を構成していたblobの一覧情報を持つ
- commit
 - changesetに対応。以前のcommitの情報と、commitした結果できたtreeの情報を持つ
- tag
 - タグの情報



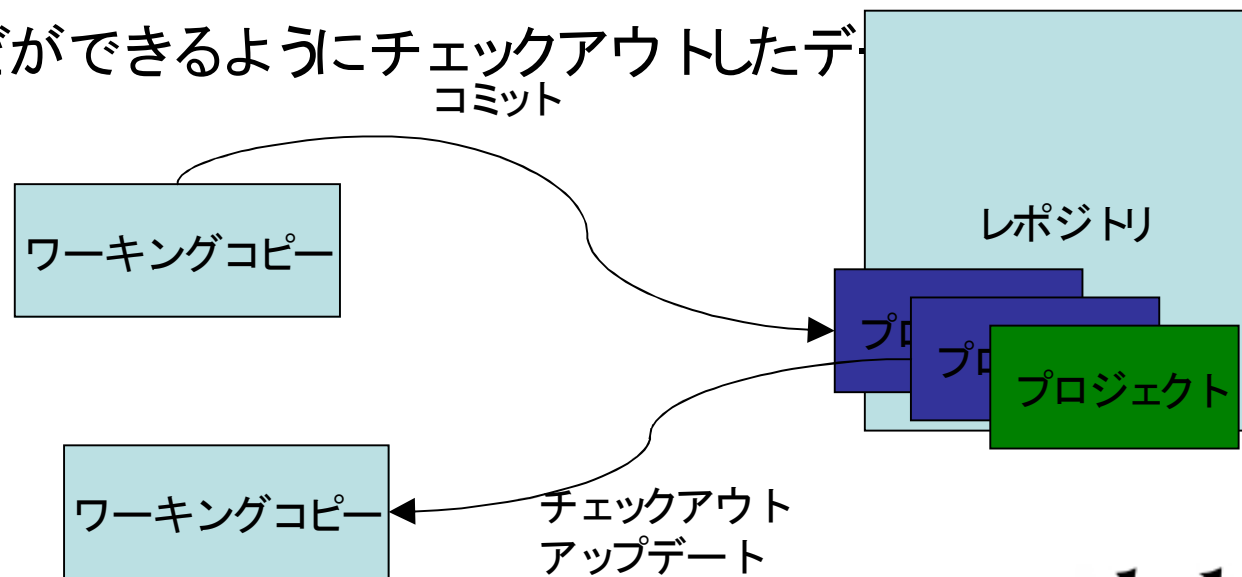
git :ディレクトリ構造

- .git/
 - HEAD 最新版のSHA-1ファイルの記録してあるファイルへのシンボリックリンク
 - index ディレクトリキャッシュインデックス
 - objects/
 - XX/ (16進数)
 - XX (38文字、16進数) 各オブジェクトがそのSHA-1ハッシュの値で保存
 - pack/ パックされたオブジェクトの保持場所
 - info/
 - packs/
 - alternatives/
 - refs/
 - heads/
 - 名前 : ブランチ
 - tags/
 - 名前 : タグ
 - info/ (git-core側の拡張、使われているのか?)
 - refs、grafts、exclude



分散していない バージョン管理ツールでの 一般的な概念

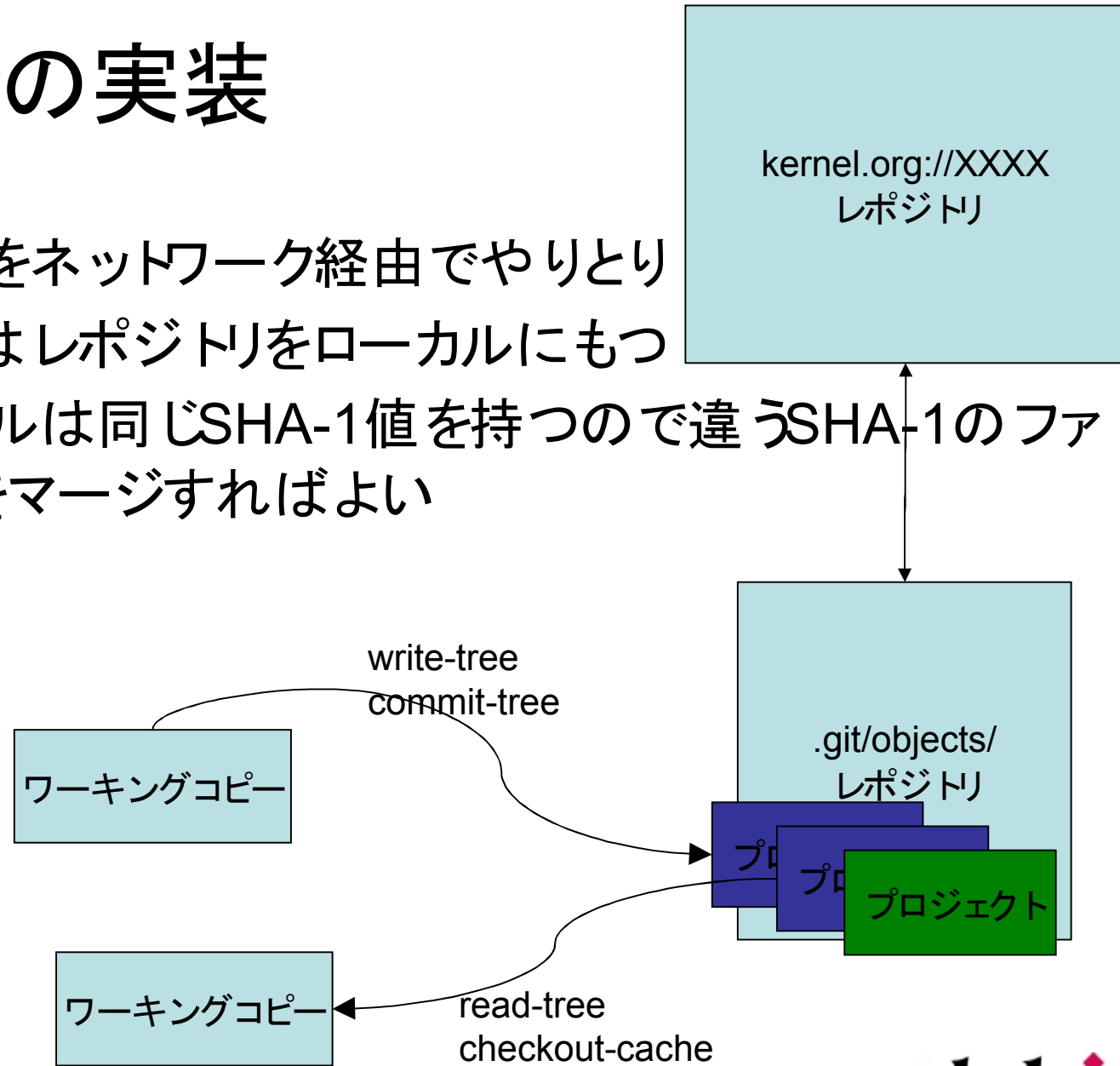
- レポジトリ
 - ファイルの履歴を保存する倉庫
- プロジェクト
 - レポジトリの中にある履歴管理されている単位
- ワーキングコピー (作業用コピー)
 - 編集などができるようにチェックアウトしたデータ
コミット





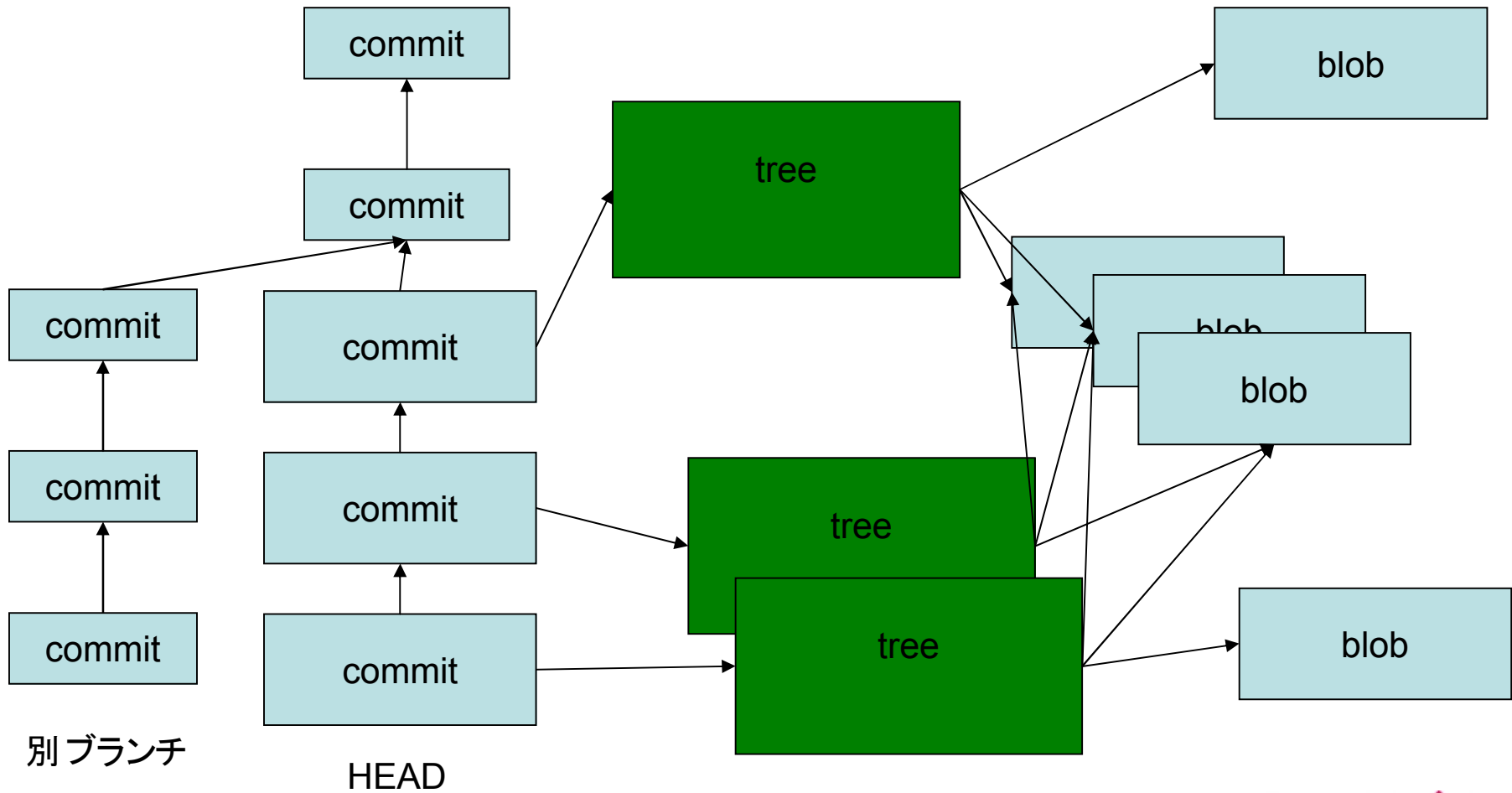
gitでの実装

- レポジトリをネットワーク経由でやりとり
- 各ユーザはレポジトリをローカルにもつ
- 同じファイルは同じSHA-1値を持つので違うSHA-1のファイルだけをマージすればよい





そしてどんどん開発がすすむ





カーネルのgitレポジトリの場所

- 現在
 - rsync://kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git/
 - <http://kernel.org/git> にウェブインタフェース
- git
 - rsync://kernel.org/pub/scm/git/git.git/
- cogito
 - rsync://kernel.org/pub/scm/cogito/cogito.git/



gitを使う

- チェックアウト
 - git-clone レポジトリの場所 ディレクトリ名
 - cd ディレクトリ名 && git-checkout
- アップデート (リモートから最新版を取得してローカルにマージ)
 - git pull アップデート元
- コミット
 - git-update-index
 - git-commit
- 公開
 - git-push レポジトリの場所
- 差分を見る
 - git-diff, git-whatchanged
- ログを見る
 - git-log



cogitoを使う

- チェックアウト
 - cg-clone レポジトリの場所
- アップデート (リモートから最新版を取得してローカルにマージ)
 - cg-update
- コミット
 - cg-status
 - cg-add
 - cg-commit
- 公開
 - cg-push
- 差分を見る
 - cg-diff
- 履歴を見る
 - cg-log



git関連ツール

- git-core: Linusがはじめたツール、現在
- cogito : Petr Baudisの開発のツール
- stgit : stacked git, quilt風を実装しているらしい
- darcs-git : darcsのgitバックエンド、darcsにマージされた
- gitweb : <http://kernel.org/git> を提供
- GUI系 :
 - qgit
 - gitk



gitの

リモートレプリケーションプロトコル

- HTTP
 - git-http-fetch
- SSH
 - git-ssh-pull, git-ssh-push, git-ssh-upload, git-ssh-fetch
- DAEMON
 - git-daemon
 - git-fetch-pack, git-upload-pack
 - IANAに9月15日登録されたい



gitの課題

- 相変わらず仕様が変更
- opensslライセンスとGPLライセンスについての議論
- cogitoがついていけてないことによる混乱



参考文献

- USAGI補完計画 「git 入門」
- git: Linux kernelのためのバージョン管理システム (VCS)
<http://www.netfort.gr.jp/~dancer/column/200504-git.html.ja>
- git: Documentation/ 以下、とソースコード