最近巷で流行っているらしいgitって何者？」
「よく知らない、誰それ？」

第50回カーネル読書会
2005年4月20日

上川　純一

debian

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
- 「linuxって何？カーネルって何？」
  - …そんな人は今日は来ない予定です
  - SCMの観点から見ると
    - ファイル数：24612くらい (2.6.12rc2)
    - ファイルサイズ： 200MBくらい
    - 過去3年間のパッチの数が数万規模（2.6.12rc2=28238 BK2CVSより）
    - 1000くらいの独立して開発しているブランチがありときどき相互にマージしあっているらしい

debian

LinuxカーネルのSCMの変遷

- 初期
  - Linusはメールでパッチを管理
  - 各開発者はメールでLinusにパッチを投げ、Linusはtar.gzで最新版を公開した
- 最近まで
  - BitKeeperを利用
  - 各自BitKeeperの独立したブランチをもつ(bknetがホスティング)
  - LinusがBk pullで変更を取り込む
  - 変化→最新版が常に公開
- BitKeeperを使わなくなって
  - gitを活用？

debian

gitとは

- Linusが開発しているらしいSCMの基本的なツール
  - 「plumbingだけをする」
  - 一般的なSCMとしての機能は上にgit-paskyなどのツールで実装
- ファイルのリビジョン単位でデータは保持する
- TREE・COMMITという概念があり、ファイルツリー、一回の変更単位で把握している
- 全ての瞬間のSNAPSHOTをファイルシステム上に持っている
  - 参考： PLAN9、PDUMPFS

debian

開発状況　2005年4月20日

- タイムライン
  - 4月5日ころ:　BitKeeperがフリー版のメンテを停止すると発表、Linusが新しいSCMツールを求む、みたいなメールを出す
  - 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13日ころ：LKMLからGITメーリングリストが独立
  - 4月17日ころ：初のマージ処理が成功、LinusのツリーをRMKのARMのツリーとマージ
  - 4月20日ころ：そろそろメインラインにしましょうか、とLinusがメール
- gitメーリングリストには一日100-200通くらいのメール
- 今日は、マージのコードをgitで実装して、それをgit-paskyが対応しているところ

debian

git (Linus git版)で チェックアウトしてみる

- export SHA1_FILE_DIRECTORY=$(pwd)/.git/objects
- cat-file commit $(cat .git/HEAD)
  - tree c66274f2229d1999a1e29cb197bd7b4a66c9dc2e
  - parent 4dbcd02f642c07813421f6368d17122925b4b6fd
  - 後略）
- read-tree c66274f2229d1999a1e29cb197bd7b4a66c9dc2e
- checkout-cache -a

debian

git(Linus版)で コミットしてみる

- write-tree
  - c66274f2229d1999a1e29cb197bd7b4a66c9dc2e
- commit-tree c66274f2229d1999a1e29cb197bd7b4a66c9dc2e -p `cat .git/HEAD `
  - ChangeLogを入力
  - 4c7d11d6e5a4dbeb215660b01430805306bca30a
- echo 4c7d11d6e5a4dbeb215660b01430805306bca30a > .git/HEAD

debian

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
- GITは必要ないファイルをstat()しなくてすむ設計
  - ファイル名前はSHA-1で決定
  - ディレクトリの構成の中で変更したファイルが何かを調べるのに.git/以下のファイルはindexファイルのみ
- 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

debian

バックエンドの仕様からみた比較

CVS	ARCH	Subversion	Monotone	git
RCS形式 1ファイルの変更履歴→1ファイル	ツリーのtar.gz パッチのtar.gz	fsfs/bdb	SQLite	1ファイル1バージョン→1ファイル
最新版があり、それ以外はパッチを適用する	初期版とキャッシュ版がありそれ以外はパッチを順に適用			各バージョンは圧縮してそのまま1ファイル
集中レポジトリモデル	分散レポジトリモデル	集中レポジトリモデル	分散レポジトリモデル	分散レポジトリモデル

debian

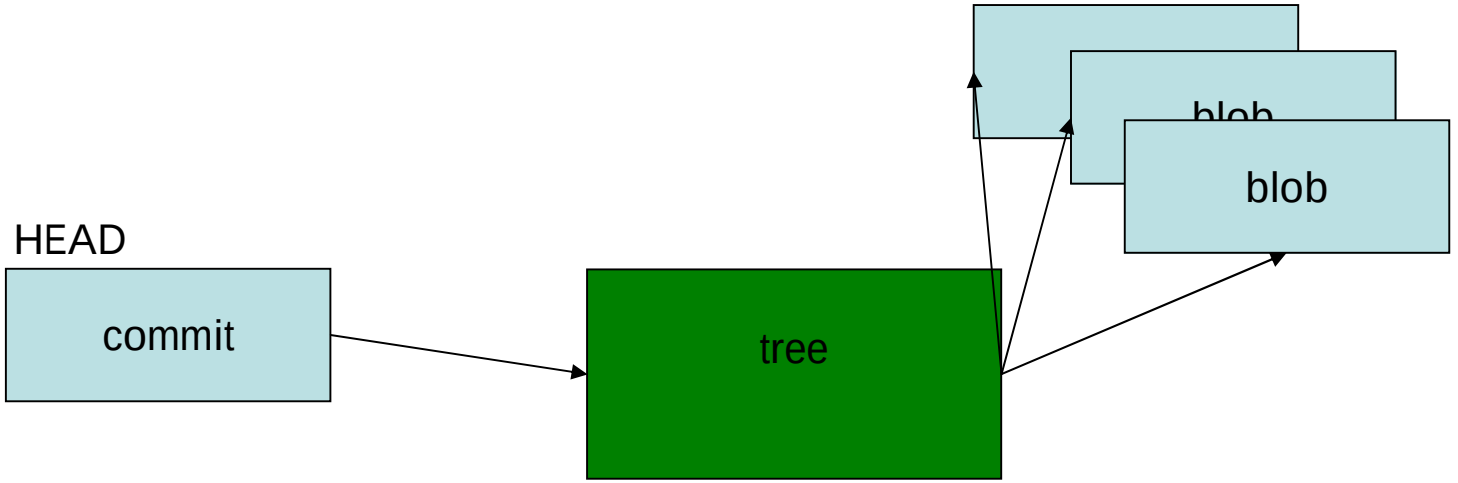
gitの基本構成

- 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

debian

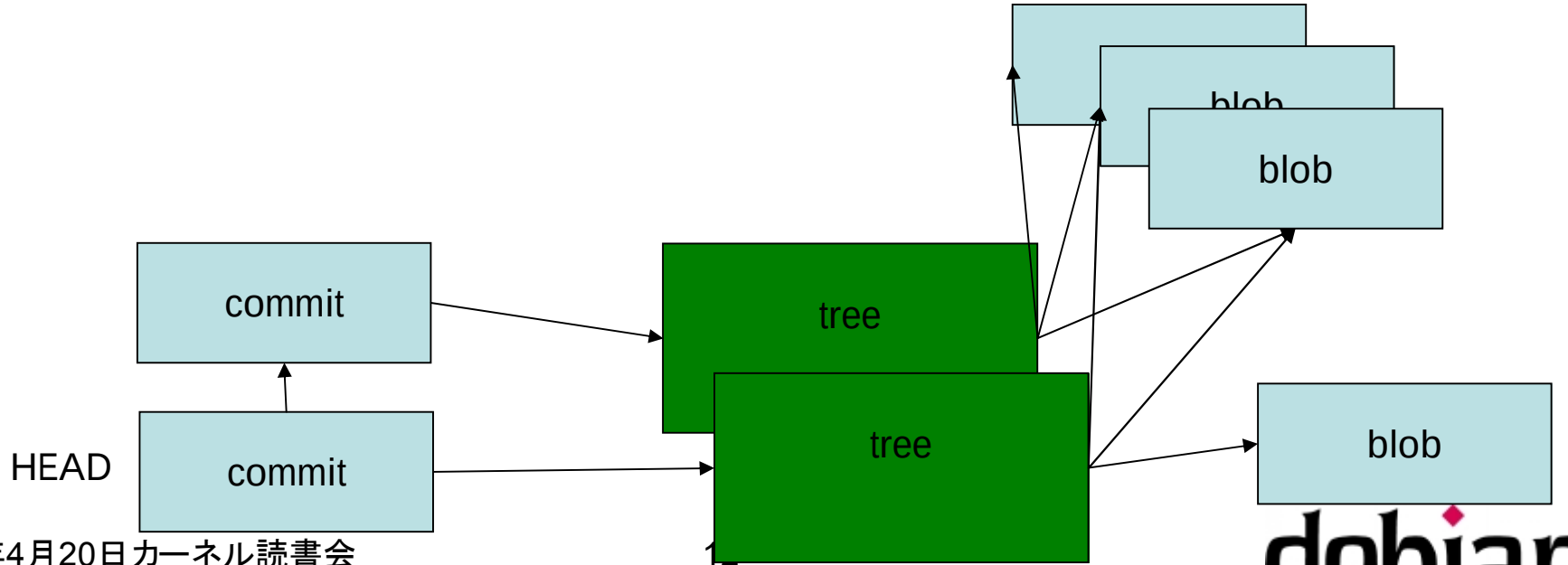
gitでのチェックアウトメカニズム

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



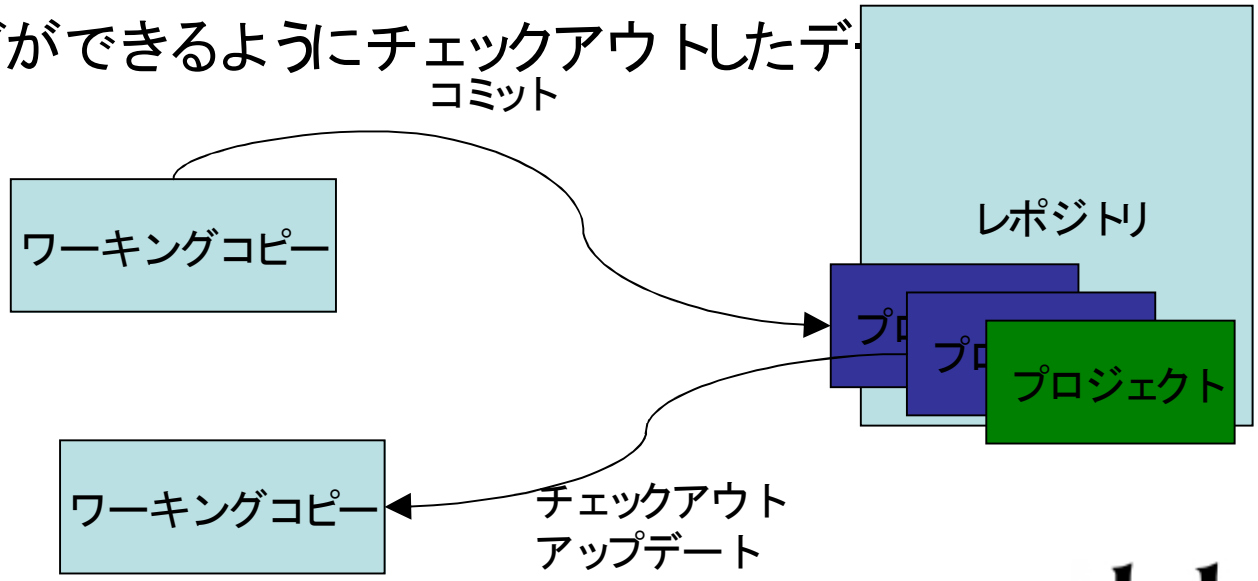
gitでのコミットメカニズム

- 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



バージョン管理ツールでの おそらく一般的な概念

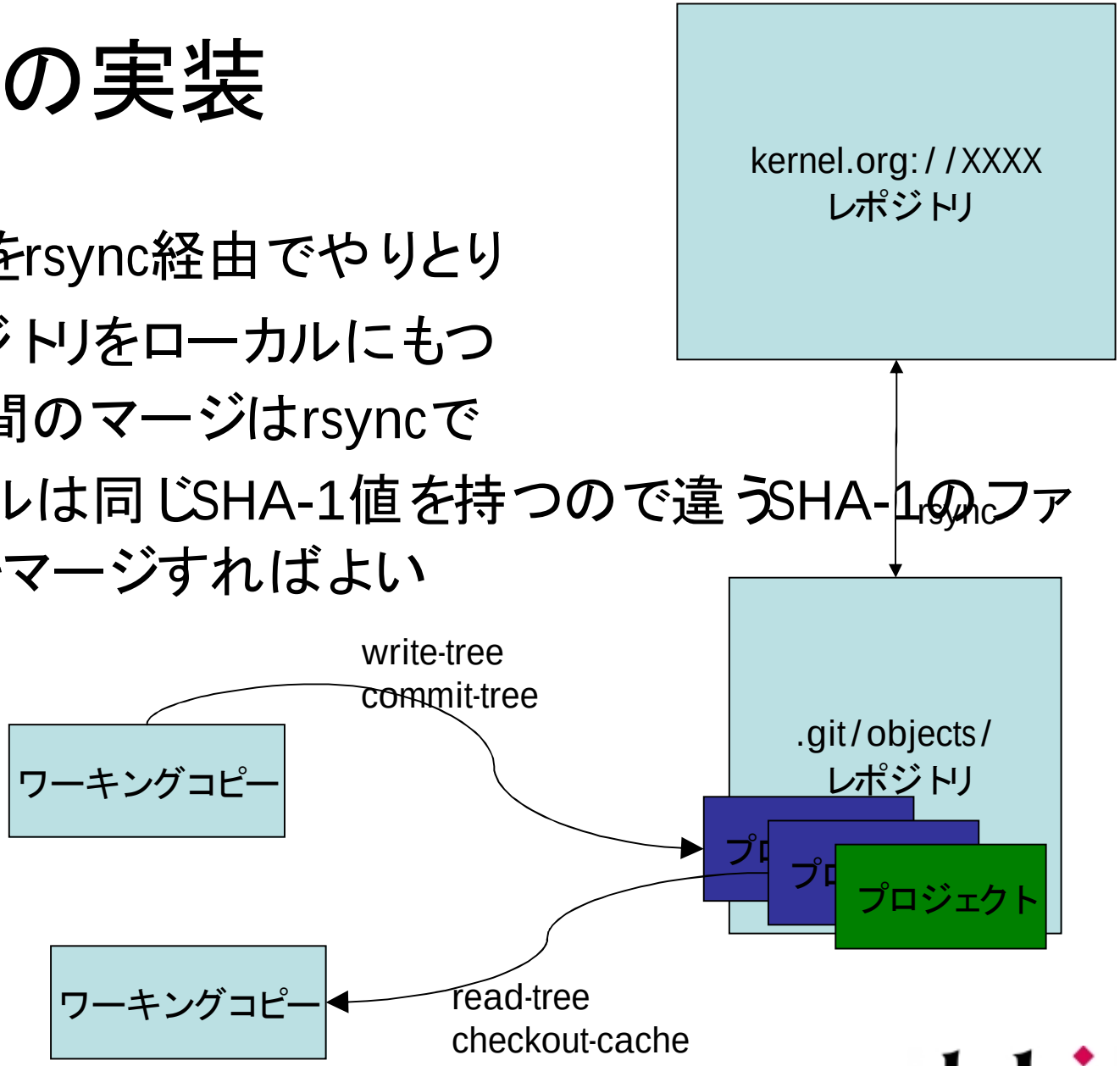
- レポジトリ
  - ファイルの履歴を保存する倉庫
- プロジェクト
  - レポジトリの中にある履歴管理されている単位
- ワーキングコピー（作業用コピー）
  - 編集などができるようにチェックアウトしたデ



debian

gitでの実装

- レポジトリをrsync経由でやりとり
- 全員レポジトリをローカルにもつ
- レポジトリ間のマージはrsyncで
- 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



debian

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
  - index ディレクトリキャッシュインデックス (read-treeで作成)
  - objects/
    - XX/
      - XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX (38文字)
  - heads/ (pasky拡張)
    - 各ブランチのヘッドのSHA-1の記録してあるファイル
  - tags/ (pasky拡張)
    - 各タグのSHA-1の記録してあるファイル

debian

利用できるツール(1/2)

- git
  - Linusが開発している
  - 人が使うのには不親切なインタフェース
  - Cで実装
  - 古いバージョン
    - http://www.kernel.org/pub/linux/kernel/people/torvalds/git-0.04.tar.bz2
  - 最新版
    - rsync://kernel.org/pub/linux/kernel/people/torvalds/git.git/
      - gitレポジトリになっているので、gitがないとチェックアウト出来ない…
- git-tools
  - Linusはメールでもらったパッチをgitに取り込むのに利用しているらしい
    - rsync://www.kernel.org/pub/linux/kernel/people/torvalds/git-tools.git/

debian

利用できるツール(2/2)

- git-pasky （cogito？）
  - Petr Baudisが開発しているツール、みんなこれを使っているようだ
  - BitKeeperとCVSを模倣したインタフェースになっている
  - シェルで実装
  - ダウンロード
    - http://pasky.or.cz/~pasky/dev/git/git-pasky-base.tar.bz2
  - 展開したあとに git pullで最新版になる
- witなど
  - ウェブフロントエンド

debian

カーネルのgitレポジトリの場所

- 現在
  - rsync://kernel.org/pub/linux/kernel/people/torvalds/linux-2.6.git
- 巨大なディレクトリ構造をミラーするのは負荷がかかるため、
  移動する予定？（4/20時点ではまだファイルがない）
  - rsync://kernel.org/pub/scm/linux/kernel/git

debian

git (4月19日のHEAD) のコマンド (1/2)

- cat-file blob/tree/commit SHA-1 内容を確認
- check-files ファイルの内容を確認
- checkout-cache キャッシュをチェックアウト
- commit-tree ツリーをコミット(ログをstdinから入力)
- diff-tree ツリー間の差分をとる
- fsck-cache キャッシュの状況を確認
- git-export gitの情報をexport, pasky必要
- init-db .gitディレクトリの作成
- ls-tree SHA-1 ツリーの一覧

debian

git (4月19日のHEAD) のコマンド (2/2)

- merge-base SHA-1 SHA-1 二つのコミットをマージするためのベースを探す
- merge-cache プログラム ファイル名 プログラムを利用してマージ?
- read-tree キャッシュに読み込む
- rev-tree commitの履歴を表示
- show-diff 現在のワーキングコピーとの差分を表示
- show-files キャッシュの管理下にあるファイルの一覧，無視されているファイルの一覧などを出力
- unpack-file SHA-1 blobをテンポラリに展開するだけの使えないコード
- update-cache --refresh コミットしていない情報を確認してくれるのと同期してくれる
- write-tree 現在のワーキングツリーからtreeオブジェクトを作成する

debian

git-pasky 0.5 (4月19日) の gitコマンドのオプション(1/2)

- add FILE... レポジトリに追加
- addremote RNAME RSYNC_URL
- apply stdinから入力したパッチを適用
- cancel
- ci, commit [FILE]... コミット $tdinからログを入力）
- diff [-p] [-r FROM_ID[:TO_ID]] [FILE]...
- export DESTDIR [TREE_ID]
- fork BNAME BRANCH_DIR [COMMIT_ID]
- help
- init RSYNC_URL

debian

git-pasky 0.5 (4月19日) の gitコマンドのオプション(2/2)

- log
- ls [TREE_ID]　　ツリーの中身をみる
- lsobj [OBJTYPE]　オブジェクトの中身をみる
- lsremote
- merge [-b BASE_ID] FROM_ID
- pull [RNAME] レポジトリの同期
- rm FILE...
- seek [COMMIT_ID]
- status
- tag TNAME [COMMIT_ID]
- track [RNAME]
- version

debian