6A-04

NetBSD ユーザランド細粒度化の実装と運用 *

深町 賢一 †　　榎本 優樹 ‡
千歳科学技術大学

1 概要

歴史的経緯により，BSD Unix ベースシステムは一つのシステムとして維持管理・ビルドされ，荒い粒度で配布されてきたが，現代では，特にセキュリティアップデートのために，きめ細かな更新機能が提供されていることがのぞましい．我々は NetBSD ベースシステムを細粒度化し細かなアップデートが可能なシステムを開発・運用している．本システム (図 1) は細粒度パッケージを生成するシステム basepkg [1, 2, 3] と配布サービス [4] および概念実証用 (PoC) クライアント (nbpkg.sh) から構成される．本システムはコミュニティベース開発であり，そのリソースは潤沢ではないが，比較的低スペックのマシンでも既存のビルドシステムと連携し，かつバイナリ中の ident(RCS のタグ) 情報をもとに更新部品のパッケージだけを生成・配布するしくみを実装することで NetBSD 8.0 stable ブランチ (以下 stable) の 60 種類超のアーキテクチャの最新パッケージを，daily で提供する運用が可能である．

2 basepkg の概要

basepkg は 1200 行程度の Bourne shell スクリプトである．システムの持続性・互換性のために POSIX 準拠 [5] をこころがけ，コーディングスタイルは ShellCheck(http://www.shellcheck.net/) によりチェックをおこない，可能なかぎり素直なコードを目指している．

basepkg は，メタデータとパッケージ生成部から構成される．ベースシステムを 1000 個あまりのパッケージに分類するメタデータは，NetBSD ソースコード内の src/distrib/sets/lists/ を元に，basepkg 側でバグ修正やフールプルーフ対策をほどこした独自拡張版である．パッケージ生成は，メタデータに基づいてベースシステムを 1000 個あまりのパッケージの集合に再編成する処理である．パッケージの作成には，サードパーティのパッケージを管理するシステム pkgsrc(http://pkgsrc.org) を利用する．basepkg を pkgsrc フレームワーク上に構築することで，高機能なパッケージ管理のしくみをベースシステム管理にも適用できるようになった．これにより，後述するクライアントで “更新時にデーモンの再起動を確実に実行する機能” や “失敗時のロールバック機能” などが実装可能となっている．

3 パッケージ配布システム

basepkg はベースシステムの分割をするだけのソフトウエアであり，OS アップデートのしくみ全体の構築には，別途，配布システムおよびクライアントの作りこみ，そして，それらのサービス運用が必要である．

パッケージ配布システム (図 1) は，(1) ダウンロード部 (2) バイナリの分析部 (3)basepkg によるパッケージ生成部から構成される．また，我々は生成されたパッケージを配布するサーバも運用している．

パッケージの対象数は，ひとつのブランチあたり，60 を超える (stable で 62)．現在，実行しているマシンは，さくらインターネットの VPS(v3) (3 コア，メモリ 2GB，HDD 200GB) で，あまり能力が高いとはいえないが，以下の実装方法を選択したことにより，低コストで，一つのブランチあたりシングルスレッドでも，ほぼ一日での処理が可能となっている．また，ブランチとアーキテクチャごとに独立した処理なので，並行処理をすれば処理速度は向上する．現在の処理速度でも，ほぼ毎日ビルドされている NetBSD 8.0 の開発版と安定版のパッケージを daily で提供可能と見積もられる．

バイナリは nycdn.netbsd.org からダウンロードする．ニューヨークにあるシステムが，最新のソースコードをビルドし，NetBSD daily と呼ばれるバイナリスナップショットを提供している．これが CDN(Fastly) を経由

* Implementation and Operation of NetBSD modular userland
† Ken'ichi Fukamachi, Chitose Institute of Science and Technology
‡ Yuuki Enomoto, Free Software Developer

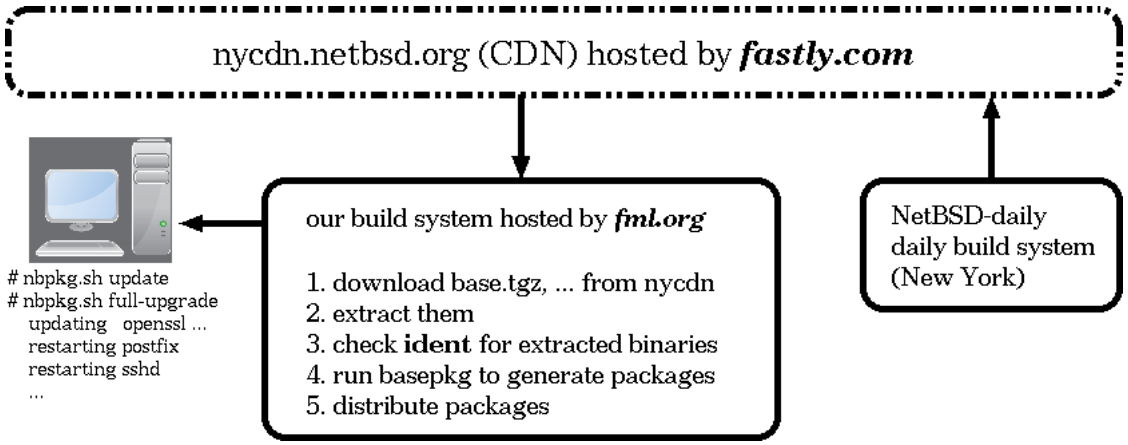

図 1　全体構成図

して配布されているためダウンロードは高速である．生成するパッケージの管理上は，ソースコードからビルドするのが望ましいが，膨大なマシンパワーを必要とするため，ダウンロード方式を採用している．

ダウンロードしたバイナリは ident 情報を元に分析し，更新分を含むパッケージのみを生成する．これは，BSD のソースコードが統一的に ident で管理されているために可能となっている．また，必要なパッケージのみを生成するため，処理速度も向上する．

4　PoC クライアント

NetBSD で Linux ディストリビューションにおける “apt update; apt upgrade” のような自動アップデートを可能とする PoC クライアント (nbpkg.sh) も製作・配布している．車輪の再発明を避け，nbpkg.sh は pkgin(pkgsrc/pkgtools/pkgin) という apt 的な機能を提供するユーティリティの wrapper として実装されている．

我々の配布サーバが提供するパッケージはベースシステム全体ではなく更新分のみであるため，nbpkg.sh の推奨動作 (“nbpkg.sh update; nbpkg full-upgrade”) では，配布パッケージをすべてインストールし，システムを最新版にする．別途，ユーザが一つ一つ確認しながらインストールする操作も，またメジャーリリースへ戻すことも可能だ．これらは，パッケージ管理フレームワークで，いつ何を入れ替えたかの履歴が明確になったことによる利点である．

5　課題と議論

本配布システムはフールプルーフをめざし (1) 推奨動作では危険な/etc の更新を行わなず (2)nbpkg.sh クライアントでの alias 機能を提供している．(1) は自動更新用パッケージ一覧を修正することで対応している．(2) はデフォルトのパッケージ命名規則が分かりにくいため，“nbpkg.sh upgrade openssl” のようなユーザにとって馴染みのある名称での操作を可能にするための機能だ．

現状，迅速な細粒度アップデート運用が実現できており十分実用的と考えるが，(1)basepkg の NetBSD 9.0 へむけた開発ブランチ (current) へのマージ (2) 粒度 (3) パッケージのバージョン名などの課題がある．(2) は当面クライアント側の alias で解決する．(3) は `etc-passwd-20181031` のような日付つきパッケージ名になる問題である．たとえば Debian では `etc-passwd-1.0.2~deb9u1` といったパッケージ名になるが，NetBSD の場合，daily build の生成が日単位のため，`etc-passwd-20181031` のようなパッケージ名にしかできない．しかしながら，一般ユーザはシステムを最新の状態に保ちたいのであって，バージョン名をひとつひとつ確認したいわけではないだろうから現方式で十分妥当と考える．

参考文献

[1] Yuuki Enomoto. `basepkg`. `https://github.com/user340/basepkg`. (accessed 2018-12-31).
[2] 榎本優樹, 深町賢一. NetBSD ベースシステムパッケージ化技法の実装報告. 情報処理学会研究報告, Vol. 2018-OS-142, No. 6, pp. 1–8, 2018.
[3] Yuuki Enomoto and Ken'ichi Fukamachi. Design, Implementation and Operation of NetBSD Base System Packaging. *AsiaBSDCon2018 Proceedings*, pp. 21–31, 2018.
[4] Ken'ichi Fukamachi. NetBSD modular userland. `https://github.com/fmlorg/netbsd-modular-userland`. (accessed 2018-12-31).
[5] 松浦智之, 大野浩之, 當仲寛哲. ソフトウェアの高い互換性と長い持続性を目指す posix 中心主義プログラミング. デジタルプラクティス, Vol. 8, No. 4, pp. 352–360, oct 2017.