

FreeBSD Ports und Packages

CCC Stuttgart

Kurt Jaeger, pi@nepustil.net

<http://nepustil.net/>

Stuttgart, 10. September 2009

Übersicht

- ▶ Allgemein
- ▶ FreeBSD
- ▶ Ports
- ▶ Packages
- ▶ Zusammenfassung

Kurzvorstellung

- ▶ Geschäftsführer eines regionalen ISPs
- ▶ FreeBSD und Linux Anwender seit ca. 1992
- ▶ Kundenserver mit FreeBSD und Linux und Windows und Solaris und ...
- ▶ /sw

Allgemein

- ▶ Was ist ein Betriebssystem ?
 - ▶ Kernel (Systemprogramm mit Sonderrechten)
 - ▶ Userspace (Systemprogramme ohne Sonderrechte)
 - ▶ Anwendungen
- ▶ Was sind Anwendungen ?
 - ▶ OpenOffice
 - ▶ mplayer
 - ▶ ...
- ▶ Was ist eine Distribution ?
- ▶ Packagemanagement

Welche Packagemanagement Systeme gibt es ?

- ▶ debian pkg [debian]
- ▶ Redhat Package Manager, RPM
- ▶ pkgsrc (netbsd u.a.) [pkgsrc]
- ▶ solaris ? pkgadd oder IPS
- ▶ AIX installp
- ▶ MacOS X Fink
- ▶ <http://www.openpkg.org/>
- ▶ <http://www.macports.org/>
- ▶ Microsoft Installer, .msi
- ▶ ...
- ▶ FreeBSD Ports und Packages

Warum Packagemanagement ?

- ▶ Bei jeder Änderung das ganze Betriebssystem überschreiben ?
- ▶ Und dann auch noch die Anwendungen ?
- ▶ Braucht viel Bandbreite
- ▶ Wahnsinnig viel Plattenplatz
- ▶ Überschreiben von lokalen Änderungen ?
- ▶ dauert sehr lang
- ▶ braucht einen Reboot
- ▶ Optimierung!
- ▶ Viele Änderungen lassen sich eingrenzen

Warum überhaupt Änderungen ?

- ▶ Fehler
- ▶ Funktion
- ▶ Sicherheit
- ▶ Wartung
- ▶ Hardwareänderungen
- ▶ Performance
- ▶ Haftung

Was sind die Unterschiede ?

- ▶ Freie Software vrs. binary-only: Beides
- ▶ build from source vrs. binaries: From Source
- ▶ Multiplattform oder nicht: Multiplattform
- ▶ Multi-OS: Teilweise
- ▶ Stabilität oder Aktualität: Aktualität

FreeBSD

- ▶ Betriebssystem: Kernel **und** Drumherum!
- ▶ Anwendungen: Ports
- ▶ Fertig compiliert: Packages
- ▶ Build-from-Source
- ▶ Release-Management

Release-Management

- ▶ CURRENT
- ▶ STABLE – ABI Stabilität
- ▶ RELEASE
- ▶ Patches

Siehe auch [release]

Vorgehen bei einem Release

- ▶ Feature Freeze
- ▶ Ports Freeze
- ▶ BETA, RC (release candidate), RELEASE, Patches
- ▶ Wie lange wird welche Version gepflegt ? [support]
- ▶ MFC: Merge from CURRENT

Ports ?

Verzeichnis: /usr/ports/

- ▶ Makefile und Mk/*
- ▶ INDEX*
- ▶ UPDATING
- ▶ UIDs und GIDs
- ▶ 63 Verzeichnisse, z.B. security/, databases/, usw
- ▶ ca. 20600 Ports
- ▶ <http://www.freshports.org/>
- ▶ /usr/ports/distfiles/
- ▶ Ständige Updates

Wie baut man selbst einen Port ?

Beispiel:

```
cd /usr/ports/www/p5-HTML-Perlinfo  
make install
```

- ▶ Das holt ggf. alle Abhängigkeiten und baut sie!
- ▶ Leider fragen manche Abhängigkeiten nach!
- ▶ Sinnvolle Defaults

make ?

- ▶ Programm, um in definierten Schritten Source Code zu übersetzen in Programme
- ▶ Variante: `bsd-make`
<http://www.freebsd.org/doc/en/books/pmake/>
- ▶ Nicht dasselbe wie `gnu-make`!
Unterschiede siehe [`make1`, `make2`]
- ▶ Viele Macros für Ports `/usr/ports/Mk`

Was passiert bei einem make install ?

- ▶ fetch
- ▶ extract
- ▶ patch
- ▶ configure
- ▶ build
- ▶ install

Weitere Targets

- ▶ deinstall
- ▶ reinstall
- ▶ package
- ▶ config
- ▶ all == check-sanity fetch checksum extract patch
configure build

man 7 ports

make config ?

- ▶ Varianten des Pakets, z.B. mit SSL oder Perl oder LDAP
- ▶ Auswahl abgelegt in `/var/db/ports/port/options`
 - WITH_DEBUG=true
 - WITH_IPV6=true
 - WITHOUT_POLICY=true
- ▶ Problem: initiale interaktive Abfrage

Zahlen und Fakten

- ▶ fbsd6/i386, fbsd7/amd64, fbsd8/i386
- ▶ ca. 1450 Ports
- ▶ ca. 4-5 GB Sources
- ▶ ca. 2.5 GB Packages als Ergebnis, bis zu 200MB (netbeans)
- ▶ Auf einer 2GHz P4 Maschine:
2-3 Wochen für initialen Lauf
Hauptverzögerung: "make config" Abfragen!
und unresolved dependencies
- ▶ OpenOffice: ca. 10 GB für den Build!
- ▶ JDKs: 2-5 GB
- ▶ Browser usw: meist 200-500 MB
- ▶ Peak-Memory ca. 1 GB (!)
- ▶ Upgrades nach initialem Lauf: Meist ein paar Stunden,
ca. 10-20 Upgrades

Upgrades

- ▶ Ports verwenden CVS
- ▶ CVS-Server: einige Mirrors
- ▶ Vorgehen:
 `cd /usr/ports`
 `csup -L 2 /root/ports-supfile`
 `make fetchindex`
 `portupgrade -arR -m BATCH=yes`
- ▶ in ca. 1/4 der Fälle: `pkgdb -F`
- ▶ wenn alles gut ging: packages bauen
- ▶ einmal im Monat `/usr/ports/UPDATING` anschauen
- ▶ Ziel: alles automatisch im Cron

Was steht in /root/ports-supfile ?

```
*default host=cvsup.se.FreeBSD.org  
*default base=/usr  
*default prefix=/usr  
*default release=cvstag=. date=2009.09.06.19.45.57  
*default delete use-rel-suffix  
*default compress
```

Beispiel pkgdb -F

Packages ?

- ▶ tar Files
- ▶ Ein wenig Meta-Information, alles ASCII!
- ▶ `/var/db/pkg/pkgdb.db`
- ▶ `/var/db/pkg/port-version/`
- ▶ `/var/db/pkg/port-version/+CONTENTS`
Eine Liste aller Files und Details mit Checksummen
- ▶ `pkg_create -j -b`
- ▶ `pkg_add`
- ▶ `pkg_delete`
- ▶ `pkg_info -W /usr/local/bin/wasndas`
- ▶ alle Pfade auch per environment konfigurierbar

Unterschiedliche Versionen derselben Software ?

- ▶ gnupg1 vrs. gnupg2: Parallel verfügbar als gpg und gpg2
- ▶ mysql 5.0.x vrs. 5.1.x: Konflikt
- ▶ openoffice2 vrs. 3: Parallel
- ▶ jdk: jdk16, openjdk6, diablo-jdk16:
Parallel via JAVA_HOME
- ▶ Je nach Einzelfall

Umgang mit Sicherheitslücken ?

- ▶ Vulnerabilities and Exposures Markup Language
- ▶ <http://www.vuxml.org/>
- ▶ security/vuxml
- ▶ make -DDISABLE_VULNERABILITIES

Eigene Ports

[porters]

- ▶ Makefile, pkg-descr
- ▶ make makesum
- ▶ portlint -A .
- ▶ Als SHAR verpacken
- ▶ mittels send-pr versenden

Beispiel: www/p5-HTML-Perlinfo

Warum build from source ?

- ▶ Abhängigkeitsgraphen, keine Listen
- ▶ Wann bricht es ?
- ▶ Verstehen der Änderungen
 - ▶ Frequenz
 - ▶ Umfang
 - ▶ Abhängigkeiten
 - ▶ Vollständigkeit
- ▶ Erfahrung sammeln in der Softwareentwicklung
- ▶ Welche Apps auf Arch xyz ?
- ▶ Welche Apps auf FreeBSD Version abc ?
- ▶ Wie häufig ist der gesamte Baum konsistent ?

Verwendung ?

- ▶ Build from Source ist aufwendig
- ▶ Also nicht auf jedem Rechner alles bauen
- ▶ Referenz-Rechner
- ▶ FreeBSD:
 - ▶ ports-mgmt/tinderbox
 - ▶ Ports Cluster [fehler]
- ▶ Konsistente Builds
- ▶ Diese als Packages einpacken
- ▶ Tausch der aktualisierten Packages auf Zielhosts (cmppp)

Hinzufügen und löschen von Ports ?

- ▶ Schnell mal testen ?
make all-depends-list
make package-depends-list
- ▶ siehe `/usr/ports/Mk/bsd.port.mk` für mehr Details
- ▶ Einfacher Fall: Paket hängt von nix ab
- ▶ Komplizierter Fall: Paket hängt von anderen Dingen ab
- ▶ Zurückschneiden des Paket-Baums ?
- ▶ `/usr/ports/distfiles/` aufräumen ?

Dependency Hell

[dependency]

- ▶ Architektur (i386, amd64, ...)
- ▶ Build und Runtime
- ▶ Libraries, Programmbibliotheken
- ▶ shared libraries
- ▶ Welche Compiler ?
- ▶ Aufrufoptionen ?
- ▶ Konfigurations-Optionen ?
- ▶ Welche Tools ?
- ▶ Threads ?
- ▶ Wie oft dasselbe in verschiedenen Versionen ?

Kombinatorische Explosion

shared libraries ?

- ▶ Zur Laufzeit hinzugelinkte Programmteile
- ▶ Endung: .so
- ▶ z.B. mplayer: 74 Libraries!
- ▶ Aber mit Versionsnummern! libtcl84.a
libtcl84.so
libtcl84.so.1
- ▶ Überhaupt: Versionsnummern!
- ▶ Hauptspeicher, Plattenplatz
- ▶ lookup Prozesse
- ▶ /etc/libmap.conf
- ▶ Versionierung von libraries
- ▶ Was ist eine API, was ist eine ABI ?
- ▶ Mix von static and dynamic linking ?

[shared]

Wann bricht eine Paketverwaltung ?

- ▶ Zwei Anwendungen
 - ▶ installieren Files
 - ▶ mit demselben Pfad
 - ▶ relevante Unterschiede
- ▶ Quell-Files sind nicht mehr zugänglich
- ▶ Komplexität, z.B. `databases/mysql-connector-java` Note: this port now installs the pre-compiled driver without the option of compiling from source, because compilation requires a mix of Java-5 and Java-6 that is not supported by `bsd.java.mk`
- ▶ Generell: Java: Timezones und Sommerzeit
- ▶ CPAN und PEAR und ...
- ▶ Local Changes

Kurz gesagt: Oft!

Zusammenfassung

- ▶ Ports und Packages
- ▶ Build-from-Source
- ▶ Aktualität
- ▶ Funktioniert gut seit ca. 10 Jahren
- ▶ CPU/Platten/IO-Aufwendig
- ▶ Verstehen der Dependency Hell
- ▶ Softwareentwicklung und Grundlagen

Links 1

[gcompare](http://en.wikipedia.org/wiki/Package_manager) http://en.wikipedia.org/wiki/Package_manager
<http://distrowatch.com/dwres.php?resource=package-management>

[debian](http://www.debian.org/doc/manuals/apt-howto/index.en.html) <http://www.debian.org/doc/manuals/apt-howto/index.en.html>

[pkgsrc](http://www.netbsd.org/docs/software/packages.html) <http://www.netbsd.org/docs/software/packages.html>

[openpkg](http://www.openpkg.org/documentation/thesis/) <http://www.openpkg.org/documentation/thesis/>

[release](http://www.atarinja.org/index.py/entries/freebsd/what-freebsd-stable-branch-means.1024px) <http://www.atarinja.org/index.py/entries/freebsd/what-freebsd-stable-branch-means.1024px>

[support](http://people.freebsd.org/~linimon/schedule/schedule.html)

<http://people.freebsd.org/~linimon/schedule/schedule.html>

[make1](http://www.wgdd.de/?p=28) <http://www.wgdd.de/?p=28>

[make2](http://lists.freebsd.org/pipermail/freebsd-questions/2007-April/147533.html) <http://lists.freebsd.org/pipermail/freebsd-questions/2007-April/147533.html>

Links 2

[porters](http://www.freebsd.org/doc/en/books/porters-handbook/) <http://www.freebsd.org/doc/en/books/porters-handbook/>

[pendency](http://en.wikipedia.org/wiki/Dependency_hell) http://en.wikipedia.org/wiki/Dependency_hell

[fehler](http://pointyhat.freebsd.org/errorlogs/#errorlist) <http://pointyhat.freebsd.org/errorlogs/#errorlist>

[shared](http://www.princeton.edu/~jdonald/research/shared_libraries/cs518_report.pdf) http://www.princeton.edu/~jdonald/research/shared_libraries/cs518_report.pdf