

>> Gentoo Distcc Dokumentation

[Bitte Kapitel auswählen]

1. Einleitung

1.1 Was ist distcc?

Distcc ist ein Programm, welches sich um die Verteilung von Kompilierungsprozessen an teilnehmende Rechner kümmert. Es besteht aus einem Serverteil, *distccd* und einem Clientprogramm, *distcc*. Mit etwas Konfigurationsaufwand funktioniert distcc mit [ccache](#) und Portage zusammen.

2. Einrichtung

2.1 Abhängigkeiten

Befehlsauflistung 1: Distcc Abhängigkeiten (Version 2.8 bis 2.10)

```
>=sys-apps/portage-2.0.46-r11
// Distcc-2.11 und höher erfordern >=sys-apps/portage-2.0.49-r6
>=sys-devel/gcc-config-1.3.1
sys-apps/shadow
// (Version 2.8 bis 2.11): Distcc hat folgende weitere Abhängigkeiten, wenn Sie gtk in Ihren USE
>=x11-libs/gtk+-2.2.1
```

Befehlsauflistung 2: Distcc Abhängigkeiten (Version 2.11 und höher)

```
>=sys-apps/portage-2.0.49-r6
>=sys-devel/gcc-config-1.3.1
sys-apps/shadow
// In dieser Revision können Sie zwischen einer Gnome und GTK GUI Überwachung auswählen
// Für GTK:
>=x11-libs/gtk+-2.0.0
>=gnome-base/libglade-2.0.0
x11-libs/pango
// Für Gnome:
>=x11-libs/gtk+-2.0.0
>=gnome-base/libglade-2.0.0
x11-libs/pango
>=gnome-base/libgnomeui-2.0.0.0
>=gnome-base/libgnome-2.0.0
```

2.2 Installing Distcc

Distcc unter Portage einzurichten ist sehr einfach. Setzen Sie einfach Ihre gewünschten USE Flags und führen Sie **emerge distcc** aus. Allerdings gibt es einige Optionen über die Sie bescheid wissen sollten.

Distcc kommt mit einem grafischen Überwachungstool, um einzelne Aufgaben, die an andere PCs verteilt werden, aufzuzeichnen. Als Gnome Benutzer sollten sie 'gnome' in Ihren USE Flags setzen. Verwenden Sie kein Gnome, wollen jedoch die Funktionalität des grafischen Tools nicht missen, dann sollten Sie 'gtk' in den USE Flags gesetzt haben.

2.3 Portage so einrichten, dass es Distcc benutzt

Distcc unter Portage einzurichten ist sehr einfach. Folgen Sie den nachfolgenden Schritten auf jedem Computer, auf dem Sie distcc verwenden möchten:

Befehlsauflistung 3: Einrichten von Distcc und Portage

```
# emerge distcc
# nano -w /etc/make.conf
```

Wichtig

Sollten Sie distcc-2.11.1 oder höher unter \geq portage-2.0.49-r6 verwenden, dann genügt es zur Einrichtung die Zielrechner (siehe nächste Codezeile) einzutragen, die MAKEOPTS in `/etc/make.conf` anzupassen, sowie 'distcc' zu FEATURES in `/etc/make.conf` einzutragen.

Notiz

Tragen Sie in Ihre FEATURES-Flags den Parameter "distcc" ein.

Notiz

unkommentieren Sie: `DISTCC_DIR=DISTCC_TMPDIR` in `/etc/make.conf`.

Notiz

Desweiteren müssen Sie die Zeile `DISTCC_DIR=${PORTAGE_TMPDIR}/portage/.distcc` in der letzten Zeile des Dokumentes eintragen.

Als nächsten Schritt müssen Sie angeben, welche Rechner benutzt werden sollen. Um dies zu erreichen, können Sie `distcc-config` benutzen. Dieses Tool ermöglicht Ihnen, eine Liste von teilnehmenden Rechnern anzugeben. Hier ein Beispiel für eine solche Liste:

Befehlsauflistung 4: Beispielhafte Liste der Rechnerdefinition

```
192.168.0.1          192.168.0.2          192.168.0.3
192.168.0.1/2       192.168.0.2          192.168.0.3/10
192.168.0.1:4000/2  192.168.0.2/1        192.168.0.3:3632/4
@192.168.0.1        @192.168.0.2:/usr/bin/distccd  192.168.0.3
```

Das mag nun alles etwas kompliziert aussehen, aber in den meisten Fällen wird eine Variation aus Zeile 1 oder 2 für Sie funktionieren. Nun eine Erklärung zu den einzelnen Zeilen: Zeile 1 ist eine mit Leerzeichen getrennte Auflistung von Rechnern, da alle mit Standardeinstellungen laufen. Zeile 2 zeigt eine Liste in der die maximale Anzahl der Prozesse, die zu einem Rechner an einem Zeitpunkt gesendet und verarbeitet werden sollen, definiert ist. Die Anzahl der zu einem Zeitpunkt zu sendenden Prozesse wird durch `/N` definiert. Da wohl die meisten Benutzer Zeile 3 und 4 nicht benutzen werden, verweise ich an die [Distcc-Dokumentation \(englisch\)](#).

Folgendes Beispiel zeigt die Einrichtung der Konfiguration, wie sie in Zeile 1 zu sehen ist:

Befehlsauflistung 5: Beispielaufruf, um teilnehmende Rechner bekannt zu geben (Beispiel: Zeile 1)

```
# /usr/bin/distcc-config --set-hosts "192.168.0.1 192.168.0.2 192.168.0.3"
```

Nun müssen Sie nur noch einen Eintrag in der Datei `/etc/make.conf` abändern, um distcc in Portage zu integrieren. Hierzu ändern Sie `MAKEOPTS` so ab, dass Sie `-jN` (wobei N eine Ganzzahl ist) auf einen Wert setzen, der üblicherweise der "Anzahl der Prozessoren + 1" der teilnehmenden Rechner entspricht.

Befehlsauflistung 6: Letzte Schritte der Konfiguration: make.conf

```
# nano -w /etc/make.conf
MAKEOPTS=-jN
```

Wichtig

Vergessen Sie bitte nicht, den Distcc-Dienst durch ein `/etc/init.d/distccd start` zu starten!

2.4 Distcc so einrichten, dass es mit Automake funktioniert

Dies ist in manchen Fällen einfacher als die Einrichtung es Portage-Teiles. Sie müssen zunächst Ihre `PATH` Umgebungsvariablen so anpassen, dass `/usr/lib/distcc/bin` vor dem Pfad von gcc (`/usr/bin`) gestellt wird. Wie auch immer, es gibt eine Vorsichtsmaßnahme zu beachten. Wenn Sie ccache benutzen, so müssen Sie den Distcc-Teil hinter dem ccache-Teil stellen. Dies würde in etwa wie folgt aussehen:

Befehlsauflistung 7: PATH setzen

```
# export PATH="/usr/lib/ccache/bin:/usr/lib/distcc/bin:${PATH}"
```

Dann, wenn Sie normalerweise `make` eingeben würden, geben Sie `make -jN` (wobei N eine Ganzzahl ist) ein. Der Wert für N variiert je nach Anzahl und der Typen der Computer, die Sie für den Kompilierprozess verwenden. Ich benutzte hierfür `-j5`, um mit einem Dual-P3 und einem K6-2/400 alles für meinen Desktoprechner (ein 1200Mhz Athlon Thunderbird) zu kompilieren. Am besten probieren Sie einige Variationen aus, um den für Sie geeignetsten Wert herauszufinden.

3. Cross-Kompilierung

3.1 Ein Bemerkung zur Cross-Kompilierung

Cross-Kompilierung benutzt eine Architektur, um für eine andere Architektur Programme zu bauen. Dies kann so einfach sein, wie einen Athlon (i686) zum Bauen eines Programm für einen K6-2 (i586) zu benutzen. Eine andere Möglichkeit wäre, dass man eine Sparc dazu benutzt, Programme für eine PowerPC zu bauen.

3.2 Ein persönliche Bemerkung zur Cross-Kompilierung

Ich liebe es im Projekt "Gentoo Cross-Compiling" aushelfen zu können, aber ich habe unglücklicherweise keine nicht-x86 Rechner zur Verfügung. Ich kann Dinge produzieren, die **in der Theorie** funktionieren, aber ich muss andere Leute bemühen dass sie die Dinge testen, die ich schreibe. Dies geht auch bis zu einem gewissen Grad, aber es ist schon schwierig. ;-)

Ich hoffe, dass ich in der nahen Zukunft eine Sparc erstehen kann, um ein Gentoo zu installieren und dann etwas damit Zuhause herumspielen kann.

4. Distcc für den Bootstrap-Prozess verwenden

4.1 Schritt 1: Setup auf einem neuen Rechner

Als nächstes müssen Sie auf Ihrem zu installierenden System die Gentoo Linux LiveCD booten und allen Schritten bis zum Bootstrapping-Prozess folgen. Dann beginnen Sie mit einigen Vorbereitungen für die Einrichtung.

Befehlsauflistung 8: Einrichtung vorbereiten

```
// Hier fügen Sie distcc den FEATURES hinzu
# nano -w /etc/make.conf
// Fügen Sie "distcc" zu FEATURES hinzu.
// Passen Sie MAKEOPTS in /etc/make.conf so an, dass Sie -jN (wie oben beschrieben hinzufügen).
```

Danach fügen Sie der `/etc/passwd` den User distcc hinzu:

Befehlsauflistung 9: User distcc in `/etc/passwd` anlegen

```
# echo "distcc:x:240:2:distccd:/dev/null:/bin/false" >>/etc/passwd
// Bitten vergessen Sie nicht die `>>' bei der Eingabe des Kommandos.
```

4.2 Schritt 2: Distcc aufsetzen

Den nächsten Schritt führen Sie in einer Chroot-Umgebung auf dem neuen Rechner aus.

Befehlsauflistung 10: Distcc auf dem neuen Rechner

```
#emerge --nodeps distcc
```

4.3 Schritt 3: Distcc einrichten

Nun richten Sie distcc selbst ein.

Befehlsauflistung 11: Letzte Einstellungen zur distcc Einrichtung

```
# /usr/bin/distcc-config --install
# /usr/bin/distcc-config --set-hosts "localhost host1 host2 host3 ..."
```

Distcc sollte nun zum Bootstrap bereit sein! Fahren Sie mit der offiziellen Installationsanweisung fort und vergessen Sie nicht distcc nach dem *emerge system* erneut zu emergem.

5. Problembehebung

5.1 Mozilla und Xfree

Wenn Sie verschiedene Pakete emergen werden Sie bemerken, dass nicht alle verteilt kompiliert (und vor allem noch nicht einmal parallel gebaut) werden. Der Grund ist, dass die Entwickler der Mozilla und XFree .ebuilds paralleles Kompilieren abgestellt haben, da bekannt ist dass dies Probleme verursachen kann. Es muss also nicht unbedingt ein distcc Problem sein.

Das heisst nicht, dass die Kompilierung mit distcc nicht auch bei dem einen oder anderen Paket fehlschlagen kann.

5.2 Eine Mixtur aus hardened-gcc und nicht-hardened-gcc Rechner kann zu Fehlern führen

Mit einer solch langen Überschrift werden weitere Erklärungen fast überflüssig. Wie auch immer. Wenn Sie planen distcc-Rechner mit Rechnern in einem Netzwerk zu verwenden wobei einige den [PaX/hardened-gcc](#) verwenden und einige nicht, werden Sie zwangsläufig auf Probleme stoßen.

Die Lösung verlangt von Ihnen ein wenig Voraussicht; dazu führen Sie *hardened-gcc -R* auf dem Rechner mit PaX/hardened-gcc aus, oder Sie müssen PaX-Protection in ihren Kernel einbauen und danach erst ein *emerge hardened-gcc* ausführen. Beide Pakete sind eine gute Sache, denn zum größten Teil steht ihnen der Schutz beider zur Verfügung und bietet auch dem Benutzer genügend Transparenz.

5.3 Verschiedene GCC Versionen

Wenn Sie verschiedene GCC Versionen auf Ihren Rechnern installiert haben, können sehr seltsame Probleme auftreten. Die Lösung ist sicherzustellen, dass auf allen Rechnern die gleiche GCC Version vorhanden ist.

6. Distcc Extras

6.1 Distcc Überwachung

Distcc wird mit zwei Überwachungstools ausgeliefert. Das textbasierte wird immer mit gebaut und heisst *distccmon-text*. Wenn Sie dieses zum ersten Mal starten, mag es etwas verwirrend wirken, aber im Prinzip ist es sehr einfach zu benutzen. Wenn Sie das Programm ohne Parameter ausführen, startet es sofort. Wenn Sie aber eine Zahl als Parameter übergeben, wird es sich alle N Sekunden aktualisieren. Hierbei ist N der Wert, den Sie beim Aufruf übergeben haben.

Das andere Überwachungstool wird nur gebaut, wenn Sie *gtk* in den *USE*-Flags stehen haben. Dieses basiert auf GTK+ und läuft unter X und ist wirklich sehr schön.

Eine Bemerkung zu den Programmen: Möchten Sie einen emerge überwachen, so müssen Sie die Überwachungstools wie folgt starten:

Befehlsauflistung 12: Überwachung starten

```
# DISTCC_DIR=/Verzeichnis/zum/distcc-Verzeichnis portage distccmon-text N
// Oder Sie starten das grafische Tool...
# DISTCC_DIR=/Verzeichnis/zum/distcc-Verzeichnis portage distccmon-gnome
```

Wichtig

Benutzen Sie Portage und das oben genannte Beispiel, dann gilt: *DISTCC_DIR=/var/tmp/portage/.distcc*

Notiz

Sollten Sie Portage nicht verwenden, müssen Sie das *DISTCC_DIR* nicht explizit angeben. Es genügt dann einfach *distccmon-text N*.

7. Zukunftspläne für Distcc und Gentoo

7.1 distcc-subnetscan

distcc-subnetscan ist ein Perlprogramm welches sich in Entwicklung befindet und ein Netzwerk nach partizipierenden distcc-Diensten durchsucht. Es wird sogar insofern verbessert werden, dass es testet, ob ein Rechner konform der *CHOST* Einstellung ist, um Cross-Kompilieren einfacher zu gestalten.

Das Perl-Script ist [hier](#) zu finden, bis ein passenderer Platz dafür gefunden ist.

7.2 distcc-config

distcc-config, ist veraltet und wird demnächst neu geschrieben und somit auf den Stand der aktuellen Distcc Version gebracht.

