

PostgreSQL für FreeBSD

MIKE FELDMEIER, SYSTEMADMINISTRATOR

In der letzten Ausgabe der freeX konnten Sie nachlesen, wie Sie mit PostgreSQL effiziente Queries durchführen. Dabei wurde vorausgesetzt, daß Sie das Datenbanksystem erfolgreich installiert haben. In diesem Beitrag erfahren Sie, wie das DBMS erfolgreich unter FreeBSD eingerichtet wird.

Das PostgreSQL-Paket wird bei dynamischem HTML im Web genauso genutzt wie etwa auch bei Datenbanken auf LAN-Servern. Es ist ein Multiuser-Datenbanksystem für diverse Plattformen. Die wichtigsten Features sind objektrelationale Datenbank-Management-Orientierung (ORDBMS), ODBC, JDBC sowie verschiedene API-Schnittstellen für viele Programmiersprachen. Die ORDBMS-Orientierung erlaubt PostgreSQL ein relationales und objektorientiertes Datenbank-Handling. Auf die auf dem Server installierte Datenbank-Engine greifen die Clients zu, die entweder auf dem lokalen Host oder auf entfernten Maschinen liegen können. Nachdem das PostgreSQL-Paket nicht nur für Linux, sondern auch für FreeBSD verfügbar ist, stellt sich natürlich die Frage, wie dieses Datenbanksystem auf einem FreeBSD-Rechner eingerichtet werden kann. Es gehorcht in der aktuellen Version dem ANSI-SQL-92-Standard weitestgehend und macht dabei weniger Abstriche an den Standard als etwa das Konkurrenzprodukt MySQL.

Zuerst die Installation

Die höhere Performance, die MySQL ins Felde führt, ist in der Regel kein Argument gegen PostgreSQL, da das Nadelöhr normalerweise in den Übertragungsraten im Web und nicht beim Server zu suchen ist. Das

MySQL-Paket ist im Gegensatz zu PostgreSQL auch nicht transaktionsorientiert. Hinzu kommt, daß PostgreSQL auch für kommerzielle Anwendungen frei ist, während MySQL für die kommerzielle Anwendung einer Lizenzgebühr unterliegt. Aus diesem Grunde soll hier nun das PostgreSQL-Paket in der Version 7.0.2 für FreeBSD 4.2-RELEASE besprochen werden.

Das PostgreSQL-Paket gehört zu den FreeBSD-Ports. Daher wird zunächst das Paket von der aktuellen freeX-CD auf die Platte gelinkt. Die Befehle

```
#mkdir /usr/ports/distfiles/postgresql
#mount /cdrom
#ln -s /cdrom/Programme/postgresql/* \
    distfiles/postgresql
```

erledigen das. Sie sollten bis zum abgeschlossenen Make-Vorgang die CD-ROM aber nicht unmounten. Im PostgreSQL-Paket ist eine Beispieldatenbank »template1« enthalten. Ihre Installation wird automatisch mit übernommen. Zur Installation sind nun folgende Eingaben zu machen:

```
#cd /usr/ports/databases/postgresql7
#make
#make install
```

Wer den weiter unten erwähnten X11-Client »pgaccess« einsetzen will, muß dem ersten *make*-Lauf den Pa-

rameter »-DWITH_TCL« mitgeben. Nach der Installation ist noch keine Datenbank initialisiert und aktiviert. Daher ist nun die Initialisierung einer eigenen Datenbank als Test der vorangegangenen Installation sinnvoll. Bei der Installation wird automatisch ein User *pgsql* angelegt. Sein Heimatverzeichnis ist »/usr/local/pgsql«.

Konfiguration der PostgreSQL-Datenbank

Die Konfiguration von PostgreSQL umfaßt Startskript, User-Anpassung und die userspezifische Anpassung der Datei »profile«, die beim Start der userspezifischen Login-Shell nach der Datei »/etc/profile« gelesen wird. Die Zugriffsrechte des Users *pgsql* werden bei der Installation bereits angepaßt. Die Befehle

```
#mkdir /usr/local/pgsql
#chown -R pgsql.pgsq1 /usr/local/pgsql
```

wurden bei der Installation bereits durchlaufen. Der User *pgsql* muß aber noch an die Gegebenheiten von FreeBSD angepaßt werden:

```
#chpass pgsql
```

Hier ist die Login-Shell auf »/usr/local/bin/bash« zu setzen. Einige Leser haben diesen Defaultwert zur Anlage eines neuen Users bereits gesetzt. Die Kontrolle aber schadet



nicht. Der »postmaster«, eine PostgreSQL-Komponente, muß vor der Initialisierung einer Datenbank als Prozeß laufen. Mit der Eingabe von

```
#su - postgres
# /usr/local/postgresql/bin/postmaster -D \
  /usr/local/postgresql/data
#exit
```

wird das erreicht. Wie man hier sehen kann, wird ein Sicherheitsrisiko umgangen, da die Datenbank nie als User *root* laufen wird.

Nun wäre PostgreSQL für die Initialisierung einer eigenen Datenbank korrekt aufgesetzt. Der Aufruf funktioniert aber noch nicht, da einige weitere Einstellungen getätigt werden müssen.

Das PostgreSQL-Paket führt einen eigenen User für Zugriffe auf die Datenbanken. Dieser als »Datenbankadministrator« bezeichnete User heißt *pgsql*. Ihm ist alles in Verbindung mit PostgreSQL erlaubt. Das betrifft bei der normalen Arbeit vor allem das Kopieren, Löschen und Anlegen von Datenbanken und Datenbank-Usern. Der User *pgsql* wird durch *passwd postgres* mit einem eigenen Paßwort versehen. Wenn man beim Arbeiten mit PostgreSQL unter einem anderen Usernamen eingeloggt ist, genügt die Angabe von *su - postgres*, um mit dem Datenbanksystem zu arbeiten. Alle User außer *root* werden dann nach dem Paßwort von *pgsql* gefragt. Hierbei wird eine Shell für den Pseudo-User *pgsql* geöffnet. Sie läßt sich mit *exit* später wieder verlassen, etwa wenn man die Arbeit unter dem User *pgsql* beendet hat, jedoch das System nicht herunterfahren möchte.

Das Standard-Datenbankverzeichnis wurde bei der Installation bereits mit

```
#mkdir /usr/local/postgresql
#mkdir /usr/local/postgresql/data
#mkdir /usr/local/postgresql/data/base
```

angelegt. Das Dateisystem ab »*usr/local/postgresql*« kann so leicht rekursiv umkopiert werden und erfaßt dann bis auf das Startskript das komplet-

te PostgreSQL-System inklusive der Programmbibliotheken, Datenbanken und Logdateien.

Im Verzeichnis »*usr/local/postgresql*« wurde bei der Installation die Datei »*.profile*« angelegt. Sie kann und sollte um weitere Einträge erweitert werden:

```
#!/usr/local/postgresql/.profile
#Nach PostgreSQL-Installation vorhanden
PATH=$PATH:/usr/local/postgresql/bin
export PATH
LD_LIBRARY_PATH=/usr/local/postgresql/lib
export LD_LIBRARY_PATH
PGLIB=/usr/local/postgresql/lib
export PGLIB
PGDATA=/usr/local/postgresql/data
export PGDATA
#DISPLAY=:0
DISPLAY=localhost:0.0
export DISPLAY
#
```

Falls diese »*.profile*« Datei fehlt, sollte sie mit *cp /root/.profile /usr/local/postgresql* vom User *root* übernommen und dann um die obigen Zeilen erweitert werden. Einige Einträge können auch ausgedünnt werden, wenn sie für den User *pgsql* nicht benötigt werden. Auch für den Superuser *root* sollten diese Änderungen in der eigenen Datei */root/.profile* vorgenommen werden. Wenn ein anderer User als *root* oder »*pgsql*« die Datenbank benutzen will, muß auch seine »*.profile*« entsprechend angepaßt werden. Zusätzlich muß dem User noch das Paßwort von *pgsql* mitgeteilt werden, da er beim Zugriff auf den Postmaster danach gefragt wird. Im Falle eines Users *mfc00* wäre folgendes einzugeben:

```
#cp /usr/local/postgresql/.profile \
  /home/mfc00
```

Dabei ist aber darauf zu achten, daß keine vorhandene Datei »*.profile*« überschrieben wird. Falls der User *mfc00* bereits über eine solche Datei verfügt, sind hingegen diese Angaben relevant:

```
#echo PATH=$PATH:/usr/local/postgresql/bin \
```

```
>>/home/mfc00/.profile
#echo export PATH \
  >>/home/mfc00/.profile
#echo LD_LIBRARY_PATH=\
  /usr/local/postgresql/lib >>/home/mfc00
#echo export LD_LIBRARY_PATH \
  >>/home/mfc00/.profile
#echo PGLIB=/usr/local/postgresql/lib \
  >>/home/mfc00/.profile
#echo export PGLIB \
  >>/home/mfc00/.profile
#echo PGDATA=/usr/local/postgresql/data \
  >>/home/mfc00/.profile
#echo export PGDATA \
  >>/home/mfc00/.profile
#echo unset DISPLAY \
  >>/home/mfc00/.profile
#echo DISPLAY=localhost:0.0 >>\
  /home/mfc00/.profile
#echo export DISPLAY >>\
  /home/mfc00/.profile
```

Bei der Installation des PostgreSQL-Pakets wurde leider kein für FreeBSD passendes Startskript in das System kopiert, da es offenkundig bei den Ports noch nicht berücksichtigt wurde. Daher ist es nun am Standardstandort für Zusatz-Startskripte einzugeben:

```
#cd /usr/local/etc/rc.d
#touch postgres.sh
#joe postgres.sh
```

Dann wird Listing 1 mit dem Editor »*joe*« eingegeben, es befindet sich aber auch auf der freeX-CD und kann aus »*freeX/postgresql*« übernommen werden.

Das Ziel »*bsdstart*« in Listing 1 ist für den manuellen Aufruf des Postmasters bestimmt, wenn etwa kein Standardstandort für Startskripte erwünscht ist und der Postmaster nur für Testzwecke gestartet werden soll. Nach Eingabe des Startskripts muß nun seine Ausführung auf passende Weise zugelassen werden:

```
#chmod 755 /usr/local/etc/rc.d/postgresql.sh
#chown postgres.sh root:bin
```

Wer auch die Manpages des PostgreSQL-Pakets lesen will, fügt der Datei »*etc/manpath.config*« fol-

gende Zeile hinzu:

```
# /etc/manpath.config
# Auszug
OPTIONAL_MANPATH /usr/local/pgsql/man
```

Die anderen installierten Manpages bleiben dann ebenfalls lesbar.

Test der Datenbank-Engine

Der Test der Datenbank umfaßt zunächst das Ausloggen mit *logout* sowie ein erneutes Einloggen mit der Angabe von User *root* und Paßwort. Nun wird das Startskript »*pgsql.sh*« aufgerufen. Der Postmaster-Prozeß

läßt sich mit

```
#killall postmaster
```

abschießen. Eleganter und sicherer ist die Eingabe von */usr/local/etc/rc.d/pgsql.sh stop*. Ob der Postmaster überhaupt läuft, wird mit

```
#ps auxw | grep -v grep | \
grep postmaster
```

erspäht. Besser ist es hier natürlich, wenn beim Hochfahren der FreeBSD-Maschine der *postmaster* automatisch gestartet und beim Herunterfahren des Systems automatisch gestoppt wird. Das saubere

Beenden des Postmasters ist außerdem zwingend notwendig, da ansonsten eventuell irreparable Schäden an den Datenbanken entstehen können. In der Datei »*/etc/rc.conf*« ist in der letzten Zeile angegeben, wo nach weiteren Start-Stop-Skripten gesucht wird:

```
#/etc/rc.conf
# Auszug letzte Zeile
local_startup=\
"/usr/local/etc/rc.d/usr/X11R6/etc/rc.d"
```

Es versteht sich, daß die Datenbank nur benutzt werden kann, wenn vorher der Postmaster-Prozeß gestartet wurde.

Die Initialisierung des Datenbanksystems wird mit

```
#su - postgres
#initdb
```

vorgenommen. Dabei wird unterstellt, daß der Postmaster per Startskript »*pgsql.sh*« automatisch oder manuell gestartet wurde. Nach einer kurzen Wartezeit ist das Datenbanksystem nun einsatzbereit. Wer will, kann sich unter »*www.postgresql.org*« registrieren lassen. Die Registrierung ist kostenlos und auch der kommerzielle Einsatz des PostgreSQL-Pakets ist frei, so daß normalerweise einer Registrierung nichts im Wege stehen sollte.

Administration der Engine

Nach der Initialisierung kommt nun die Administration zum Tragen. Zuerst wird ein Beispieluser angelegt. Dies wird mit *createuser myuser* erreicht. Dazu muß man als User *pgsql* eingeloggt sein. Der angelegte User läßt sich mit *destroyuser myuser* bei Bedarf auch wieder entsorgen. Nach den Usern ist dann die Datenbank an der Reihe. Mit *createdb mydata* läßt sich eine Datenbank anlegen. Sie hat noch keine Struktur wie Feldnamen oder Triggered Procs; sie ist noch leer. Mit *destroydb mydata* läßt sich diese Datenbank jeder Zeit wieder entfernen. Mit *exit* verläßt man die

```
#!/bin/sh
# For postmaster startup options,
# edit $PGDATA/postmaster.opts.default
# Preinstalled options are -i -o "-F"

case $1 in
bsdstart)
[ -x /usr/local/pgsql/bin/postmaster ] &&
{
su -l postgres -c ' exec /usr/local/pgsql/bin/postmaster
-D/usr/local/pgsql/data
-S -o -F >/usr/local/pgsql/pgerr.log' &
echo -n ' postgres'
}
;;
start)
[ -x /usr/local/pgsql/bin/pg_ctl ] &&
{
su -l postgres -c \
'exec /usr/local/pgsql/bin/pg_ctl -w start > '\
'/usr/local/pgsql/pgerr.log 2>&1'
echo -n ' postgres'
}
;;
stop)
[ -x /usr/local/pgsql/bin/pg_ctl ] &&
{
su -l postgres -c 'exec /usr/local/pgsql/bin/pg_ctl \
-w -m fast stop'
}
;;
status)
[ -x /usr/local/pgsql/bin/pg_ctl ] &&
{
su -l postgres -c 'exec /usr/local/pgsql/bin/pg_ctl status'
}
;;
*)
echo "usage: `basename $0` {start|stop|status}" >&2
exit 64
;;
esac
```

Listing 1: Startskript für PostgreSQL unter FreeBSD

Shell des Users *pgsql* und landet in der vorigen Shell.

Jetzt geht es darum, einzelne Datensätze in die Datenbank einzutragen. Dies wird mit dem konsolenbasierten PostgreSQL-Client »psql« vorgenommen. Er wurde automatisch mitinstalliert, da er zum Lieferumfang des Pakets gehört. In diesem Falle wird

```
#su - pgsq1
#psql mydata
```

einggegeben, um mit der angelegten Datenbank zu arbeiten. Wer selbst SQL-Skripte in einem »myskript.sql« anlegen will, anstelle die SQL-Befehlszeilen interaktiv von »psql« abarbeiten zu lassen, läßt diese dann mit *psql -f myskript.sql* abarbeiten.

Skripten

Diese Skripte können alle Aspekte wie Datenbankstruktur, Feldnamen, Datensatz-Zufügungen, Datensatz-Löschungen oder Datenbankabfragen abdecken. Nach Skript-Ende wird in diesem Beispiel der psql-Client mitbeendet. Wie bereits erwähnt, läßt sich die »pgsql«-Shell mit *exit* beenden. Wer lieber unter X11 arbeitet, verwendet den Client »pgaccess«. Er wird in einem Terminalfenster mit *pgaccess &* angeworfen. Wenn »pgaccess« noch nicht installiert wurde, muß dies mit

```
#cd /usr/ports/databases/pgaccess
#make USE_TCL=yes
```

| Pfad/Datei | Funktion |
|--------------------------------------|------------------------------------|
| /usr/local/etc/rc.d/pgsql.sh | Das Startskript aus Listing 1. |
| <i>Verzeichnisse:</i> | |
| /usr/local/pgsql/data | Der PostgreSQL-Standardpfad. |
| /usr/local/pgsql/data/base | Das PostgreSQL-DB-SYS-Verzeichnis. |
| /usr/local/pgsql/data/base/template1 | Die Beispieldatenbank. |
| <i>Log-Dateien:</i> | |
| /usr/local/pgsql/data/pg-xlog | Die Logdatei von PostgreSQL. |
| /usr/local/pgsql/pgerr.log | Die Fehler-Logdatei. |
| /root/server.log | Die Logdatei PostgreSQL/root. |
| <i>PID:</i> | |
| /var/run/pgsql.pid | Prozeß-ID-Datei. |
| <i>Andere:</i> | |
| /etc/passwd | Einträge zum User <i>pgsql</i> . |
| /usr/local/share/doc/pgsql | Die Dokumentation zu PostgreSQL. |
| /usr/local/pgsql/bin/pgsql | SQL-Client. |
| /usr/local/pgsql/bin/postmaster | SQL-Backend. |

Tabelle 1: Die wichtigsten Pfade und Dateien des PostgreSQL-Pakets

```
#make all
#make install
```

nachgeholt werden. Das Paket erfordert die Anwesenheit der Dateien »tcl-8.3.1.tar.gz« sowie »tk-8.3.1plus.tar.gz« im Verzeichnis »/usr/ports/distfiles«. Wer im Web eingewählt ist, kann vor dem make-Lauf das angeführte Linken der angegebenen Dateien von CD-ROM in das Verzeichnis »/usr/ports/distfiles« unterlassen. Weitere Informationen zu »pgaccess« sind unter »www.flex.ro/pgaccess« erhältlich. Auf »www.de.freebsd.org« liegen

weitere Clients wie etwa »GtkSQL« für das verbreitete PostgreSQL-Datenbanksystem zum Download bereit. Es ließe sich noch viel über das »PostgreSQL«-Paket schreiben. Dazu gehören Transaction-Tracking und die Erörterung der ANSI-SQL92-Implementation genauso wie die Kommunikation mit der Datenbank-Engine via »socks« oder die Einbindung von Perl-Modulen. Dieser Beitrag sollte dazu dienen, die Grundvoraussetzungen für die Installation und Initialisierung des PostgreSQL-Pakets auf einem x86-FreeBSD-System zu beleuchten. ◆

