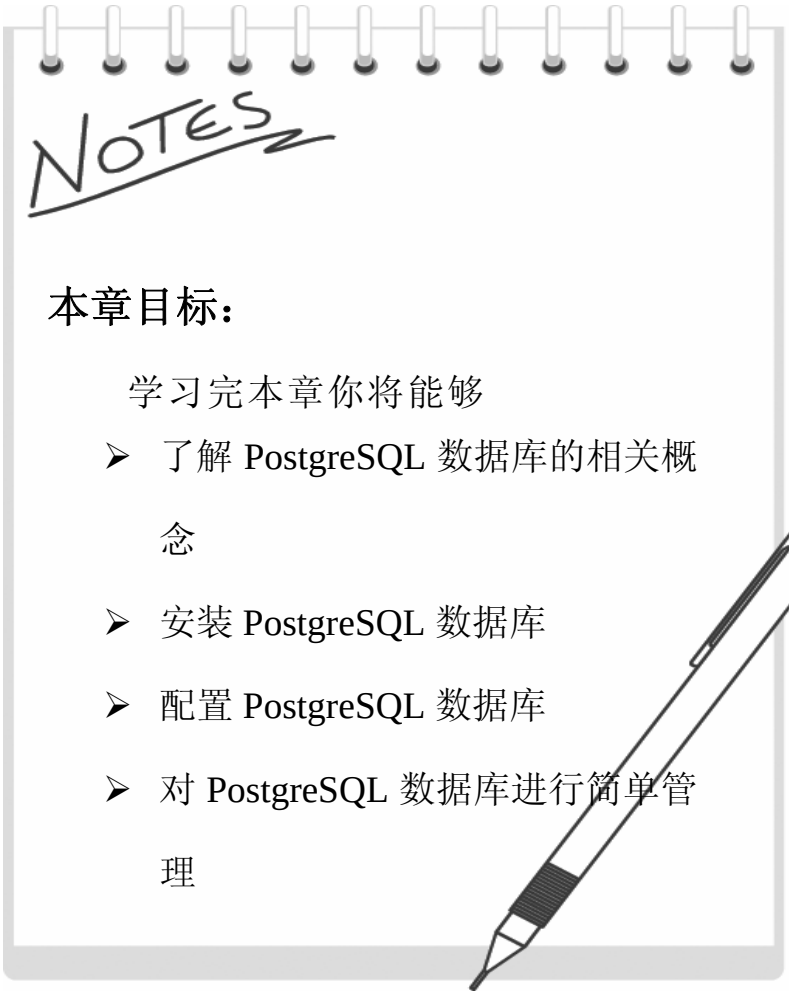
第八章　PostgreSQL 数据库的安装与配置

Linux 操作系统下的 PostgreSQL 数据库属于关系型数据库范畴，它支持 SQL92/SQL3 的事务完整性与可扩展性，它提供了一种更强大、更完善的开发接口的功能。并且它是公开源代码的。在 Linux 系统下，是一个广泛运用的数据库库之一。本章将介绍 PostgreSQL 的概念及其应用。



1. PostgreSQL 简介

PostgreSQL 最早是由 UCBerkley 大学计算机系开发的，当初 Michael Stonebraker 教授领导的 Postgres 项目是由高级防务研究项目局（Defense Advanced Research Projects Agency）（DARPA）、陆军研究处（ARO）、国家科学基金会（National Science Foundation）（NSF）和 ESL 公司赞助进行的。它先进的“对象－关系”概念现在已经在一些商业数据库里得到应用，PostgreSQL 支持 SQL92/SQL3 的事务完整性和可扩展性，是一个源于 Berkley 代码并公开源代码的数据库。

Postgres 通过一种让用户可以很容易扩展系统的方法整合了下面的四种基本概念，使其能提供很多的附加功能：

类/表（classes）
继承（inheritance）
类型（types）
函数（functions）
其他特性还灵活地提供了附加的功能：
约束（constraints）
触发器（triggers）
规则（rules）
事务完整（transactionintegrity）

这些特性将 Postgres 置于对象－关系（object-relational）型数据库的范畴。尽管 Postgres 有一些面向对象的特性，但它仍然属于关系型数据库的范畴。事实上，一些商用数据库已经集成了一些 Postgres 所领先的特性。

2. 安装

安装 PostgresSQL 可以使用二进制代码包安装，也可以从源代码安装。而使用源代码安装更具通用性。因此，我们的课程中很多软件的安装都会以源代码的安装来进行讲解。

2.1 安装前的准备

一般说来，一个现代的与 Unix 兼容的平台应该就能运行 PostgreSQL。在发布的 doc 子目录里面有许多平台相关的 FAQ 文档，如果你碰到问题可以参考它们。

从源代码安装 PostgreSQL 需要下面几样东西：

（1）GNU make 工具 这个制作过程不能使用其他的 make 程序。 GNU make 常被安装为 gmake 的名字。本文档将一直使用这个名字称呼她。 （在一些系统上 GNU make 是名字叫 make 的缺省工具。）要想测试 GNU make，敲入：

gmake --version

我们建议使用版本 3.76.1 或者更新的版本。

（2）ISO/ANSI C 编译器。推荐使用最近版本的 GCC，不过，PostgreSQL 可以利用许多不同厂商的不同编译器进行编译。例如 intel 的 C/C++编译器。

（3）缺省时将自动使用 GNU Readline， （这样你可以方便地编辑和检索命令历史。） 如果你不想用它，那么你必需给 configure 声明 --without-readline 选项。（在 NetBSD 上，libedit 库是 readline 兼容的， 如果没有发现 libreadline，则使用这个库。） 如果你使用的是一个基于包的 Linux 发布，那么要注意你需要 readline 和 readline-devel 两个包，特别是如果这两个包在你的版本里是分开的时候。

（4）要制作服务器端编程语言 PL/Perl，你需要一个完整的 Perl 安装， 包括 libperl 库和头文件。 因为 PL/Perl 是一个共享库， libperl 库在大多数平台上也必须是一个共享库。

（5）要制作 Python 接口模块或者 PL/Python 服务器端编程语言， 你需要一个 Python 的安装，包括头文件和 distutils 模块。对于 Python 1.6 或者更高版本，缺省时就带有 distutils 模块； 更早版本的 Python 需要自己安装它。

（6）如果你想制作 PL/Tcl 语言， 你当然需要安装 Tcl 了。

（7）要打开本地语言支持 （NLS），也就是说，用英语之外的语言显示程序的信息，你需要一个 Gettext API 的实现。有些操作系统内置了这些（比如 Linux，NetBSD ， Solaris ）， 对 于 其 它 系 统 ， 你 可 以 从 这 里 ： http://developer.postgresql.org/~petere/bsd-gettext/ 下载一个额外的包。如果在使用 GNU C 库里面的 gettext 实现， 那么你就额外需要 GNU Gettext 包，因为我们需要里面的几个工具程序。对于任何其它的实现，你应该不需要它。

（8）Kerberos，OpenSSL，和/或 PAM， 如果你想支持使用这些服务的认证或者加密，那你需要这些包。

（9）请检查一下，看看你是否有足够的磁盘空间。 你将大概需要近 65MB 用于存放安装过程中的源码树和大约 15 MB 用于安装目录。 一个空数据库大概需要 25 MB。 然后在使用过程中大概需要在一个平面文本文件里存放同等数据量数据五倍的空间存储数据。 如果你要运行回归测试，还临时需要额外的 90MB。请用 df 命令检查剩余磁盘空间。

2.2 安装

1.下载软件：

创建存放软件的目录： mkdir /backup/software (可以存放在任何自己想存放的目录)

到 ftp://ftp.postgresql.org/pub/source/ 下载最新稳定版的源代码，放到/backup/software。 例如:

ftp://ftp.postgresql.org/pub/source/v8.0.3/postgresql-8.0.3.tar.bz2

2.解压软件： tar -jxvf postgresql-8.0.3.tar.bz2

3.进入源代码目录： cd ostgresql-8.0.3

4.配置编译选项：

./configure --prefix=/mnt/software/pgsql

选项说明：

--prefix=/mnt/software/pgsql：将所有文件安装到/mnt/software/pgsql 目录下如果你需要其它选项，请使用 ./configure --help 查看或参看文档 http://www.postgresql.org/docs/8.0/interactive/install-procedure.html

5.编译: gmake

6.安装: gmake install

7.创建 PostgreSQL 的运行用户

adduser postgres

8.创建数据库目录：

```
mkdir -p /mnt/database/pgsql_data
```

9.将数据库目录的拥有者改为 PostgreSQL 的运行用户

```
chown postgres /mnt/database/pgsql_data
```

10.切换到 PostgreSQL 的运行用户身份：

```
su - postgres
```

11.初始化数据库：

```
/mnt/software/pgsql/bin/initdb -D /mnt/database/pgsql_data
```

其中“-D”选项用于指定数据库德存储位置。

12.设置共享库（Shared Libraries）

```
/sbin/ldconfig /mnt/software/pgsql/lib
或者:
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/mnt/software/pgsql/lib
```

13.设置环境变量：

如果要使用本地的客户端工具，并且不想每次都输入完整路径，就需要设置 PATH 变量。

```
export PATH=$PATH:/mnt/software/pgsql/bin
```

14.启动数据库服务：

```
/mnt/software/pgsql/bin/postmaster -D /mnt/database/pgsql_data > logfile 2>&1 &
或者：
/mnt/software/pgsql/bin/pg_ctl -D /mnt/database/pgsql_data -l logfile start
```

其中“-D”选项用于指定数据库德存储位置。

如果要通过远程网络访问数据库，加上 -i 选项，例如：

```
/mnt/software/pgsql/bin/pg_ctl -i -D /mnt/database/pgsql_data -l logfile start
```

15.测试：

创建一个数据库 test

```
/mnt/software/pgsql/bin/createdb test
```

使用数据库 test:

```
/mnt/software/pgsql/bin/psql test
```

16.启动脚本(使系统启动时自动启动 PostgreSQL 数据库服务)：

在 Linux 系统下：

```
cp /backup/software/postgresql-8.0.3/contrib/start-scripts/linux /etc/init.d/postgresql
```

编辑文件件/etc/init.d/postgresql，修改 prefix 和 PGDATA 参数，修改如下：

```
prefix=/mnt/software/pgsql
PGDATA="/mnt/database/pgsql_data"
```

如果要通过远程网络访问数据库,同时修改 start、stop、reload 函数，修改如下:

```
start)
$ECHO_N "Starting PostgreSQL: "$ECHO_C
su - $PGUSER -c "$DAEMON -i -D '$PGDATA' &" >>$PGLOG 2>&1
echo "ok"
;;
stop)
echo -n "Stopping PostgreSQL: "
su - $PGUSER -c "$PGCTL stop -D '$PGDATA' -s -m fast"
echo "ok"
;;
restart)
echo -n "Restarting PostgreSQL: "
su - $PGUSER -c "$PGCTL stop -D '$PGDATA' -s -m fast -w"
su - $PGUSER -c "$DAEMON -i -D '$PGDATA' &" >>$PGLOG 2>&1
echo "ok"
;;
```

修改权限：

```
chmod a+x /etc/init.d/postgresql
chkconfig --add postgresql
```

3. PostgreSQL 的使用与管理

PostgreSQL 是一个书库管理系统，由于篇幅有限，在此仅介绍基本的使用方法。如果要深入了解 PostgreSQL 数据库的管理与使用，请参考科卡在先

（http://www.koorka.com）的另一门课程《PostgreSQL 数据库管理》以及 PostgreSQL 的相关文档。

3.1 登录到数据库

要管理 PostgreSQL 数据库服务器，首先需要使用 psql 工具连接(登陆)到服务器，基本用法如下：

用户 postgres 登陆到服务器 192.168.10.18 的 maildb 数据库上：

```
psql -h 192.168.10.18 -d maildb -U postgres
```

3.2 用户管理

3.2.1 创建和删除用户：

(1)创建用户：

CREATE USER bearzhang;

(2)删除用户

DROP USER bearzhang;

(3)查看系统中的已有用户：

select * from pg_user;

3.2.2 用户属性管理

(1)口令管理：

设置明文口令：ALTER USER bearzhang WITH PASSWORD '123456';

设置加密口令：ALTER USER bearzhang WITH ENCRYPTED PASSWORD '123456';

(2)用户口令的有效日期:

ALTER USER bearzhang VALID UNTIL 'Jan 31 2030';

使用户的口令永远有效:

ALTER USER bearzhang VALID UNTIL 'infinity';

(3)使用户成为超级用户(superuser):

ALTER USER bearzhang CREATEUSER;

使用户成为普通用户：

ALTER USER bearzhang NOCREATEUSER;

(4)使用户拥有创建数据库的权限：

ALTER USER bearzhang CREATEDB;

superuser 拥有对数据库的完全控制权限，因此，superuser 用户不需要赋予 CREATEDB 权限就可以创建数据库。

取消用户创建数据库的权限：

ALTER USER bearzhang NOCREATEDB;

(5)重命名用户：

ALTER USER bearzhang RENAME TO newuser;

3.3 用户组管理

用户组，即所谓的角色，它是一个劝降的集合。在新版本的 PostgreSQL 数据库中使用 CREATE ROLE 来创建角色。

(1)创建用户组 mail_group:

CREATE GROUP mail_group;

(2)将用户 bearzhang 和 john 加入到用户组 mail_group:

ALTER GROUP mail_group ADD USER bearzhang,john;

(3)删除用户组 mail_group:

DROP GROUP mail_group;

(4)查看当前用户组的信息:

SELECT * FROM pg_group;

3.4 用户权限管理

(1) 用户权限的类型:

数据库中的用户可以拥有如下的权限:

SELECT, INSERT, UPDATE, DELETE, RULE, REFERENCES, TRIGGER, CREATE, TEMPORARY, EXECUTE, 和 USAGE.

(2) 使用户 bearzhang 对当前数据库内的 mailbox 表拥有 SELECT，UPDATE，DELETE 权限

GRANT SELECT,UPDATE,DELETE ON mailbox TO bearzhang;

(3)使 mail_group 用户组的用户对当前数据库内的 mailbox 表拥有 SELECT 权限:

GRANT SELECT ON mailbox TO GROUP mail_group;

(4)使用户 bearzhang 对当前数据库内的 mailbox 表拥有所有权限:

GRANT ALL ON mailbox TO bearzhang;

(5)删除除用户 bearzhang 对当前数据库内的 mailbox 表拥有的所有权限:

REVOKE ALL ON mailbox FROM bearzhang;

删除用户组 mail_group 对当前数据库内的 mailbox 表的 SELECT 权限:

REVOKE ALL ON mailbox FROM bearzhang;

3.5 数据库管理

在此主要介绍数据库的添加删除和用户管理以及创建表格

3.5.1 系统数据库

当启动数据库时，数据库如何知道系统中有哪些用户？有哪些数据库？这些信息都存储在系统数据库中。

template0 (该数据库为模版数据库,不会被系统或用户改动，创建数据库时可以使用该数据库为模版创建一个“干净”的数据库。)

template1 (该数据库为用户模版数据库，同时也是系统数据库。缺省情况下，用户新创建的数据库由该模版生成，当用户对该模版修改后，新创建的数据库也会随模版改变。)

查看当前数据库的信息：

SELECT * FROM pg_database;

3.5.2 创建数据库

(1)使用缺省参数创建数据库 maildb:

CREATE DATABASE maildb;

(2)创建数据库，并使数据库的拥有者为用户 bearzhang:

CREATE DATABASE maildb OWNER bearzhang;

(3)创建数据库 maildb，并使数据库的拥有者为 bearzhang，同时将数据库的数据放到 mailspace 表空间上：

CREATE DATABASE maildb OWNER bearzhang TABLESPACE mailspace;

3.5.3 创建表空间

(1)创建表空间 mailspace,其物理存储空间为/mnt/sda1/postgresql/data ：

a. 在 shell 命令提示符下使用 root 用户创建目录/mnt/sda1/postgresql/data:

mkdir -p /mnt/sda1/postgresql/data

chown postgres /mnt/sda1/postgresql/data

b. 在 psql 工具中使用 SQL 语句创建表空间：

CREATE TABLESPACE mailspace LOCATION '/mnt/sda1/postgresql/data';

(2) 在创建表格时使用表空间：

在 test 数据库中创建表格 person，使其数据存储在表空间 mailspace 上（即/mnt/sda1/postgresql/data 目录中）

CREATE TABLE person(

id int,

lastname varchar(30),

address varchar(300)

) TABLESPACE mailspace;

(3)查看当前表空间的信息：

SELECT * FROM pg_tablespace

3.6 数据表管理

表格是数据库中同类数据的集合。例如在一个电子商务数据库中，我们将产品信息放在一个表中，将客户信息放在一个表中，将订单信息放在一个表中……

(1)认识数据类型

数据库中的数据类型主要用来限制用户输入的数据的值和类别，例如，在下面的员工信息表中格：

姓名	职位	工资
张磊	技术支持	4000
李亮	咨询师	五千
……	……	……

如果我们要统计员工的平均工资，由于有的是数字，有的是字符，因此无法进行统计，为了方便统计员工工资，必须规定工资这一列使用同一种数据格式。这就是数据类型的主要目的和作用，当然它还有其他的用途 ，请在使用过程中慢慢体会。

PostgreSQL 常用数值型数据类型：

| 数据类型 | 占用的存储空间 | 取值范围

| smallint | 2 bytes | -32768 to +32767

| integer | 4 bytes | -2147483648 to +2147483647

| bigint | 8 bytes | -9223372036854775808 to 9223372036854775807

| real | 4 bytes | 6 decimal digits precision

| double precision| 8 bytes | 15 decimal digits precision

| serial | 4 bytes | 1 to 2147483647

| bigserial | 8 bytes | 1 to 9223372036854775807

常用字符型数据类型:

变长字符串：varchar(n)

定长字符串：character(n), char(n)

文本型：text

常用日期时间型：

日期时间型: timestamp

日期型：date

时间型：time

分别执行以下三条 SQL 语句，可以看到三种日期时间的输出：

select current_timestamp(2);

select current_date;

select current_time;

布尔型:

boolean

(值： TRUE ，可以使用 't' 'true' 'y' 'yes' '1' 来表示；

FALSE，可以使用 'f' 'false' 'n' 'no' '0' 来表示)

(2)创建表格

创建表格实际上就是定义我们日常生活中的一张表的表头，例如我们要用一张表格来存储员工的信息，表的格式如下：

员工编号	姓名	联系电话	家庭住址	工资
1	张磊	12345678	北京	4000
2	李量	87654321	上海	5000

现在我们用 emp_id 来代表员工编号，username 代表员工姓名，phone 代表员工联系电话，address 代表家庭住址，salary 代表员工工资，则在数据库内应该定义的表格如下：

CREATE TABLE employees (

emp_id serial,

username varchar(20),

phone varchar(20),

address varchar(300),

salary real

)

(3)修改表:

将 products 表中的 product_no 重命名为 product_number:

ALTER TABLE products RENAME COLUMN product_no TO product_number;

将 products 表的名称重命名为 items:

ALTER TABLE products RENAME TO items;

在 products 表中增加一列 description，其数据类型是 text:

ALTER TABLE products ADD COLUMN description text;

在 products 表中增加一列 description，其数据类型是 text,并且该列的值不能为空字符串:

ALTER TABLE products ADD COLUMN description text CHECK (description <> '');

在 products 表中删除 description 列：

ALTER TABLE products DROP COLUMN description;

将 products 表中的 price 列的数据类型修改为数据长度（精度）为 10 位，小数点后取 2 位的数值型:

ALTER TABLE products ALTER COLUMN price TYPE numeric(10,2);

将 products 表的 price 的缺省值设置为 7.77:

ALTER TABLE products ALTER COLUMN price SET DEFAULT 7.77;

创建约束:

ALTER TABLE products ADD CHECK (name <> '');

ALTER TABLE products ADD CONSTRAINT some_name UNIQUE (product_no);

ALTER TABLE products ADD FOREIGN KEY (product_group_id) REFERENCES product_groups;

删除约束:

ALTER TABLE products DROP CONSTRAINT some_name;

更多修改表格的方法，请参阅：数据定义 和 ALTER TABLE 的使用

3.7 客户端认证配置

缺省安装完 PostgreSQL 数据库服务器后，它只接受本机的连接，为了使网络用户也能使用该数据库，我们必须配置它的用户认证(客户端认证)配置文件 pg_hba.conf,该文件在数据库的初始化目录下，即$PGDATA/pg_hba.conf。

该文件中每行为一条配置记录（项）,每条配置中共有 5 个域(列)，列之间用一个制表符（或空格）分隔，例如：

host all all 127.0.0.1/32 trust

依次代表：类型(TYPE)，数据库(DATABASE)，用户(USER)，子网(CIDR-ADDRESS)，认证方式(METHOD)。

下面分别列出出每个域可使用的值（或方式）:

TYPE: local,host,hostssl,hostnossl

DATABASE: all,databaselist,@filename,sameuser

USER: all,userlist,+groupname,@filename

CIDR-ADDRESS: 192.168.0.0/24

METHOD: trust,reject,md5,crypt,password,krb4,krb5,ident,pam

Example pg_hba.conf entries:

TYPE DATABASE USER CIDR-ADDRESS METHOD

local all all trust

host all all 127.0.0.1/32 trust

host all all 192.168.54.1/32 reject

host all all 192.168.0.0/16 password

host db1,db2,@dblistfile @userlist,+mygroup 192.168.0.0/16 password

(@filename is in $PGDATA/filename)

例如，要使 IP 为 192.168.100.8 的主机可以使用 mail_user 用户访问 maildb 数据库，在访问时需要提供用户名和密码，那么需要在$PGDATA/pg_hba.conf 文件中添加下面的行：

host maildb mail_user 192.168.100.8/32 password