

PostgreSQL 数据库安装教程

(仅供参考)

目 录

1. Linux安装PostgreSQL数据库	3
1.1. 概述.....	3
1.2. 新建postgres用户和目录	3
1.3. 配置postgres用户的环境变量:	3
1.4. yum安装.....	3
1.5. 初始化数据库目录.....	3
1.6. 给postgres赋予权限	4
1.7. 配置监听地址、端口.....	4
1.8. 配置支持远程连接.....	4
1.9. 设置开机启动.....	4
1.10. 启动服务.....	4
1.11. 修改postgres帐号密码	4
2. Windows安装PostgreSQL数据库	5
2.1. 概述.....	5
2.2. 安装.....	5
2.3. 配置监听地址、端口.....	5
2.4. 配置支持远程连接.....	5
3. 注意事项.....	6
4. 附录一: PostgreSQL数据库介绍	6
5. 附录二: PostgreSQL与MySQL比较	6

1. Linux安装PostgreSQL数据库

1.1. 概述

以 CentOS 6.4（或以上）**X64（64 位）** 版本，PostgreSQL8.4 为例子说明。

本文仅供参考，如果安装不成功，或需要更多 PostgreSQL 技术资料，请自行上网搜索；

1.2. 新建postgres用户和目录

```
mkdir /var/lib/pgsql  
groupadd postgres  
useradd -g postgres postgres
```

1.3. 配置postgres用户的环境变量：

```
cat >>/var/lib/pgsql/.bash_profile<<EOF  
PGDATA=/var/lib/pgsql/data  
export PGDATA  
PGCLIENTENCODING=EUC_CN  
export PGCLIENTENCODING  
EOF
```

1.4. yum安装

```
yum -y install postgresql-server postgresql-libs
```

1.5. 初始化数据库目录

```
su postgres  
/usr/bin/initdb -D /var/lib/pgsql/data
```

1.6. 给postgres赋予权限

exit 退回 root 用户，执行下面命令：

```
chown postgres:postgres /var/lib/pgsql/data
```

1.7. 配置监听地址、端口

```
vi /var/lib/pgsql/data/postgresql.conf
```

取消以下两行的注释以及修改以下参数：

```
listen_addresses = '*'
```

```
port = 5432
```

```
max_connections = 2000
```

1.8. 配置支持远程连接

```
vi /var/lib/pgsql/data/pg_hba.conf
```

在文件最后增加以下一行：

```
host all all 0.0.0.0/0 md5
```

1.9. 设置开机启动

```
chkconfig postgresql on
```

1.10. 启动服务

```
service postgresql start
```

1.11. 修改postgres帐号密码

```
su postgres
```

```
psql
```

```
ALTER USER postgres PASSWORD '你的密码';
```

```
\q
```

2. Windows安装PostgreSQL数据库

2.1. 概述

Windows 版本 PostgreSQL 安装最新 9.3 版本，下载地址：

<http://www.postgresql.org/download/windows/>

<http://www.enterprisedb.com/products-services-training/pgdownload#windows>

2.2. 安装

不需要重造相同的轮子，网上已经有很不错的安装教程，这里随机搜索了二个，如果时间久了，链接无效，也可以自己到网上搜索：

[PostgreSQL安装详细步骤（windows）](#)

[Windows系统中安装PostgreSQL数据库图文教程](#)

安装完成后，记得修改配置监听地址、端口，和配置支持远程连接，请看后面内容。

2.3. 配置监听地址、端口

打开 C:\Program Files (x86)\PostgreSQL\9.3\data\postgresql.conf 文件

修改以下参数：

```
listen_addresses = '*'
```

```
port = 5432
```

```
max_connections = 2000
```

2.4. 配置支持远程连接

打开 C:\Program Files (x86)\PostgreSQL\9.3\data\pg_hba.conf 文件

在文件最后增加以下一行：

```
host all all 0.0.0.0/0 md5
```

3. 注意事项

- A、一定要配置远程 IP 访问，否则会导致数据库连接失败；或者尝试使用 localhost；
- B、数据库帐号不要使用 postgres 系统帐号；
- C、如果确实新建数据库帐号、新建数据库及导入数据库数据失败，请使用 PostgreSQL 客户端管理工具 pgAdmin3 来进行操作，包括新建帐号，新建数据库和导入数据；恩布安装程序只是配置保存信息即可；
- D、利用 pgAdmin3 来导入数据，需要先新建帐号如 ebuser，新建数据库如 entboost，选择 UTF-8 字符集，和数据库所有者 ebuser，新建完成后选择恢复数据库；
- E、尽量避免把 PostgreSQL 数据库装到中文目录下，特别是 Windows 环境下的安装，如果发现数据库服务不能正常启动，请卸载后重新安装到默认英文环境目录下；

4. 附录一： PostgreSQL数据库介绍

PostgreSQL 百度百科：<http://baike.baidu.com/view/28196.htm>

5. 附录二： PostgreSQL与MySQL比较

<http://bbs.chinaunix.net/thread-1688208-1-1.html>

http://www.oschina.net/question/23734_62623

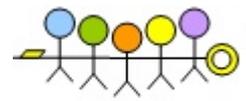
特性	MySQL	PostgreSQL
实例	通过执行 MySQL 命令 (mysqld) 启动实例。一个实例可以管理一个或多个数据库。一台服务器可以运行多个 mysqld 实例。一个实例管理器可以监视 mysqld 的各个实例。	通过执行 Postmaster 进程 (pg_ctl) 启动实例。一个实例可以管理一个或多个数据库，这些数据库组成一个集群。集群是磁盘上的一个区域，这个区域在安装时初始化并由一个目录组成，所有数据都存储在这个目录中。使用 initdb 创建第一个数据库。一台机器上可以启动多个实例。
数据库	数据库是命名的对象集合，是与实例中的其他数据库分离的实体。一个 MySQL 实例中的所有数据库共享同一个系统编目。	数据库是命名的对象集合，每个数据库是与其他数据库分离的实体。每个数据库有自己的系统编目，但是所有数据库共享 pg_databases。
数据	通过 innodb_buffer_pool_size 配置参数设置	Shared_buffers 缓存。在默认情况下分

缓冲区	数据缓冲区。这个参数是内存缓冲区的字节数, InnoDB 使用这个缓冲区来缓存表的数据和索引。在专用的数据库服务器上, 这个参数最高可以设置为机器物理内存量的 80%。	配 64 个缓冲区。默认的块大小是 8K。可以通过设置 postgresql.conf 文件中的 shared_buffers 参数来更新缓冲区缓存。
数据库连接	客户机使用 CONNECT 或 USE 语句连接数据库, 这时要指定数据库名, 还可以指定用户 id 和密码。使用角色管理数据库中的用户和用户组。	客户机使用 connect 语句连接数据库, 这时要指定数据库名, 还可以指定用户 id 和密码。使用角色管理数据库中的用户和用户组。
身份验证	MySQL 在数据库级管理身份验证。基本只支持密码认证。	PostgreSQL 支持丰富的认证方法: 信任认证、口令认证、Kerberos 认证、基于 Ident 的认证、LDAP 认证、PAM 认证
加密	可以在表级指定密码来对数据进行加密。还可以使用 AES_ENCRYPT 和 AES_DECRYPT 函数对列数据进行加密和解密。可以通过 SSL 连接实现网络加密。	可以使用 pgcrypto 库中的函数对列进行加密/解密。可以通过 SSL 连接实现网络加密。
审计	可以对 querylog 执行 grep。	可以在表上使用 PL/pgSQL 触发器来进行审计。
查询解释	使用 EXPLAIN 命令查看查询的解释计划。	使用 EXPLAIN 命令查看查询的解释计划。
备份、恢复和日志	InnoDB 使用写前 (write-ahead) 日志记录。支持在线和离线完全备份以及崩溃和事务恢复。需要第三方软件才能支持热备份。	在数据目录的一个子目录中维护写前日志。支持在线和离线完全备份以及崩溃、时间点和事务恢复。可以支持热备份。
JDBC 驱动程序	可以从 参考资料 下载 JDBC 驱动程序。	可以从 参考资料 下载 JDBC 驱动程序。
表类型	取决于存储引擎。例如, NDB 存储引擎支持分区表, 内存引擎支持内存表。	支持临时表、常规表以及范围和列表类型的分区表。不支持哈希分区表。由于 PostgreSQL 的表分区是通过表继承和规则系统完成了, 所以可以实现更复杂的分区方式。
索引类型	取决于存储引擎。MyISAM: BTREE, InnoDB: BTREE。	支持 B-树、哈希、R-树和 Gist 索引。
约束	支持主键、外键、唯一和非空约束。对检查约束进行解析, 但是不强制实施。	支持主键、外键、唯一、非空和检查约束。
存储过程和用户定义函数	支持 CREATE PROCEDURE 和 CREATE FUNCTION 语句。存储过程可以用 SQL 和 C++ 编写。用户定义函数可以用 SQL、C 和 C++ 编写。	没有单独的存储过程, 都是通过函数实现的。用户定义函数可以用 PL/pgSQL (专用的过程语言)、PL/Tcl、PL/Perl、PL/Python、SQL 和 C 编写。

数		
触 发 器	支持行前触发器、行后触发器和语句触发器，触发器语句用过程语言复合语句编写。	支持行前触发器、行后触发器和语句触发器，触发器过程用 C 编写。
系 统 配 置 文 件	my.conf	Postgresql.conf
数 据 库 配 置	my.conf	Postgresql.conf
客 户 机 连 接 文 件	my.conf	pg_hba.conf
XML 支持	有限的 XML 支持。	有限的 XML 支持。
数 据 访 问 和 管 理 服 务 器	<p>OPTIMIZE TABLE —— 回收未使用的空间并消除数据文件的碎片</p> <p>myisamchk -analyze —— 更新查询优化器所使用的统计数据 (MyISAM 存储引擎)</p> <p>mysql —— 命令行工具</p> <p>MySQL Administrator —— 客户机 GUI 工具</p>	<p>Vacuum —— 回收未使用的空间</p> <p>Analyze —— 更新查询优化器所使用的统计数据</p> <p>psql —— 命令行工具</p> <p>pgAdmin —— 客户机 GUI 工具</p>
并 发 控 制	支持表级和行级锁。InnoDB 存储引擎支持 READ_COMMITTED、READ_UNCOMMITTED、REPEATABLE_READ 和 SERIALIZABLE。使用 SET TRANSACTION ISOLATION LEVEL 语句在事务级设置隔离级别。	支持表级和行级锁。支持的 ANSI 隔离级别是 Read Committed (默认 —— 能看到查询启动时数据库的快照) 和 Serialization (与 Repeatable Read 相似 —— 只能看到在事务启动之前提交的结果)。使用 SET TRANSACTION 语句在事务级设置隔离级别。使用 SET SESSION 在会话级进行设置。

MySQL 相对于 PostgreSQL 的劣势:

MySQL	PostgreSQL
最重要的引擎 InnoDB 很早就由 Oracle 公司控制。目前整个 MySQL 数据库都由 Oracle 控制。	BSD 协议，没有被大公司垄断。
对复杂查询的处理较弱，查询优化器不够成熟	很强大的查询优化器，支持很复杂的查询处理。
只有一种表连接类型: 嵌套循环连接 (nested-loop), 不支持排序-合并连接 (sort-merge)	都支持



join)与散列连接(hash join)。	
性能优化工具与度量信息不足	提供了一些性能视图，可以方便的看到一个表和索引上的 select、delete、update、insert 统计信息，也可以看到 cache 命中率。网上有一个开源的 pgstatspack 工具。
InnoDB 的表和索引都是按相同的方式存储。也就是说表都是索引组织表。这一般要求主键不能太长而且插入时的主键最好是按顺序递增，否则对性能有很大影响。	不存在这个问题。
大部分查询只能使用表上的单一索引；在某些情况下，会存在使用多个索引的查询，但是查询优化器通常会低估其成本，它们常常比表扫描还要慢。	不存在这个问题
表增加列，基本上是重建表和索引，会花很长时间。	表增加列，只是在数据字典中增加表定义，不会重建表
存储过程与触发器的功能有限。可用来编写存储过程、触发器、计划事件以及存储函数的语言功能较弱	除支持 pl/pgsql 写存储过程，还支持 perl、python、Tcl 类型的存储过程：pl/perl, pl/python, pl/tcl。 也支持用 C 语言写存储过程。
不支持 Sequence。	支持
不支持函数索引，只能在创建基于具体列的索引。 不支持物化视图。	支持函数索引，同时还支持部分数据索引，通过规则系统可以实现物化视图的功能。
执行计划并不是全局共享的，仅仅在连接内部是共享的。	执行计划共享
MySQL 支持的 SQL 语法(ANSI SQL 标准)的很小一部分。不支持递归查询、通用表表达式 (Oracle 的 with 语句) 或者窗口函数 (分析函数)。	都支持
不支持用户自定义类型或域 (domain)	支持。
对于时间、日期、间隔等时间类型没有秒以下级别的存储类型	可以精确到秒以下。

身份验证功能是完全内置的, 不支持操作系统认证、PAM 认证, 不支持 LDAP 以及其它类似的外部身份验证功能。	支持 OS 认证、Kerberos 认证、Ident 的认证、LDAP 认证、PAM 认证
不支持 database link。有一种叫做 Federated 的存储引擎可以作为一个中转将查询语句传递到远程服务器的一个表上, 不过, 它功能很粗糙并且漏洞很多	有 dblink, 同时还有一个 dbi-link 的东西, 可以连接到 oracle 和 mysql 上。
Mysql Cluster 可能与你的想象有较大差异。开源的 cluster 软件较少。 复制(Replication)功能是异步的, 并且有很大的局限性。例如, 它是单线程的(single-threaded), 因此一个处理能力更强的 Slave 的恢复速度也很难跟上处理能力相对较慢的 Master。	有丰富的开源 cluster 软件支持。
explain 看执行计划的结果简单。	explain 返回丰富的信息。
类似于 ALTER TABLE 或 CREATE TABLE 一类的操作都是非事务性的。它们会提交未提交的事务, 并且不能回滚也不能做灾难恢复	DDL 也是有事务的。

PostgreSQL 主要优势:

1. PostgreSQL 完全免费, 而且是 BSD 协议, 如果你把 PostgreSQL 改一改, 然后再拿去卖钱, 也没有人管你, 这一点很重要, 这表明了 PostgreSQL 数据库不会被其它公司控制。oracle 数据库不用说了, 是商业数据库, 不开放。而 MySQL 数据库虽然是开源的, 但现在随着 SUN 被 oracle 公司收购, 现在基本上被 oracle 公司控制, 其实在 SUN 被收购之前, MySQL 中最重要的 InnoDB 引擎也是被 oracle 公司控制的, 而在 MySQL 中很多重要的数据都是放在 InnoDB 引擎中的, 反正我们公司都是这样的。所以如果 MySQL 的市场范围与 oracle 数据库的市场范围冲突时, oracle 公司必定会牺牲 MySQL, 这是毫无疑问的。

2. 与 PostgreSQL 配合的开源软件很多, 有很多分布式集群软件, 如 pgpool、pgcluster、slony、plplox 等等, 很容易做读写分离、负载均衡、数据水平拆分等方案, 而这在 MySQL 下则比较困难。

3. PostgreSQL 源代码写的很清晰, 易读性比 MySQL 强太多了, 怀疑 MySQL 的源代码被混淆过。所以很多公司都是基本 PostgreSQL 做二次开发的。

4. PostgreSQL 在很多方面都比 MySQL 强, 如复杂 SQL 的执行、存储过程、触发器、索引。同时 PostgreSQL 是多进程的, 而 MySQL 是线程的, 虽然并发不高时, MySQL 处理速度快, 但当并发高的时候, 对于现在多核的单台机器上, MySQL 的总体处理性能不如 PostgreSQL, 原因是 MySQL 的线程无法充分利用 CPU 的能力。

目前只想到这些, 以后想到再添加, 欢迎大家拍砖。

PostgreSQL 与 oracle 或 InnoDB 的多版本实现的差别

PostgreSQL 与 oracle 或 InnoDB 的多版本实现最大的区别在于最新版本和历史版本是否分离存储，PostgreSQL 不分，而 oracle 和 InnoDB 分，而 innodb 也只是分离了数据,索引本身没有分开。

PostgreSQL 的主要优势在于:

1. PostgreSQL 没有回滚段,而 oracle 与 innodb 有回滚段,oracle 与 InnoDB 都有回滚段。对于 oracle 与 InnoDB 来说,回滚段是非常重要的,回滚段损坏,会导致数据丢失,甚至数据库无法启动的严重问题。另由于 PostgreSQL 没有回滚段,旧数据都是记录在原先的文件中,所以当数据库异常 crash 后,恢复时,不会象 oracle 与 InnoDB 数据库那样进行那么复杂的恢复,因为 oracle 与 InnoDB 恢复时同步需要 redo 和 undo。所以 PostgreSQL 数据库在出现异常 crash 后,数据库起不来的几率要比 oracle 和 mysql 小一些。

2. 由于旧的数据是直接记录在数据文件中,而不是回滚段中,所以不会象 oracle 那样经常报 ora-01555 错误。

3. 回滚可以很快完成,因为回滚并不删除数据,而 oracle 与 InnoDB,回滚时很复杂,在事务回滚时必须清理该事务所进行的修改,插入的记录要删除,更新的记录要更新回来(见 row_undo 函数),同时回滚的过程也会再次产生大量的 redo 日志。

4. WAL 日志要比 oracle 和 InnoDB 简单,对于 oracle 不仅需要记录数据文件的变化,还要记录回滚段的变化。

PostgreSQL 的多版本的主要劣势在于:

- 1、最新版本和历史版本不分离存储,导致清理老旧版本需要作更多的扫描,代价比较大,但一般的数据库都有高峰期,如果我们合理安排 VACUUM,这也不是很大的问题,而且在 PostgreSQL9.0 中 VACUUM 进一步被加强了。

- 2、由于索引中完全没有版本信息,不能实现 Coverage index scan,即查询只扫描索引,直接从索引中返回所需的属性,还需要访问表。而 oracle 与 InnoDB 则可以;

进程模式与线程模式的对比

PostgreSQL 和 oracle 是进程模式,MySQL 是线程模式。

进程模式对多 CPU 利用率比较高。

进程模式共享数据需要用到共享内存,而线程模式数据本身就是在进程空间内都是共享的,不同线程访问只需要控制好线程之间的同步。

线程模式对资源消耗比较少。

所以 MySQL 能支持远比 oracle 多的更多的连接。

对于 PostgreSQL 的来说,如果不使用连接池软件,也存在这个问题,但 PostgreSQL 中有优秀的连接池软件软件,如 pgbouncer 和 pgpool,所以通过连接池也可以支持很多的连接。

堆表与索引组织表的的对比

Oracle 支持堆表,也支持索引组织表

PostgreSQL 只支持堆表,不支持索引组织表

InnoDB 只支持索引组织表

索引组织表的的优势:

表内的数据就是按索引的方式组织,数据是有序的,如果数据都是按主键来访问,那么访问数据比较快。

而堆表，按主键访问数据时，是需要先按主键索引找到数据的物理位置。

索引组织表的劣势：

索引组织表中上再加其它的索引时，其它的索引记录的数据位置不再是物理位置，而是主键值，所以对于索引组织表来说，主键的值不能太大，否则占用的空间比较大。

对于索引组织表来说，如果每次在中间插入数据，可能会导致索引分裂，索引分裂会大大降低插入的性能。所以对于使用 `innodb` 来说，我们一般最好让主键是一个无意义的序列，这样插入每次都发生在最后，以避免这个问题。

由于索引组织表是按一个索引树，一般它访问数据块必须按数据块之间的关系进行访问，而不是按物理块的访问数据的，所以当做全表扫描时要比堆表慢很多，这可能在 `OLTP` 中不明显，但在数据仓库的应用中可能是一个问题。

PostgreSQL 9.0 中的特色功能：

PostgreSQL 中的 Hot Standby 功能

也就是 `standby` 在应用日志同步时，还可以提供只读服务，这对做读写分离很有用。这个功能是 `oracle11g` 才有的功能。

PostgreSQL 异步提交 (Asynchronous Commit) 的功能：

这个功能 `oracle` 中也是到 `oracle11g R2` 才有的功能。因为在很多应用场景中，当宕机时是允许丢失少量数据的，这个功能在这样的场景中就特别合适。在 PostgreSQL 9.0 中把 `synchronous_commit` 设置为 `false` 就打开了这个功能。需要注意的是，虽然设置为了异步提交，当主机宕机时，PostgreSQL 只会丢失少量数据，异步提交并不会导致数据损坏而数据库起不来的情况。MySQL 中没有听说过有这个功能。

PostgreSQL 中索引的特色功能：

PostgreSQL 中可以有部分索引，也就是只能表中的部分数据做索引，`create index` 可以带 `where` 条件。同时 PostgreSQL 中的索引可以反向扫描，所以在 PostgreSQL 中不必建专门的降序索引了。