# 插件的安装

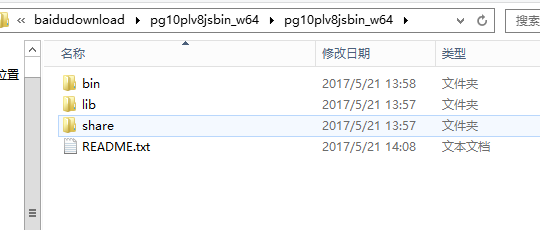
所有的插件都是通过源代码编译的

安装方法

在window上需要自己用msvc去编译，在linux用 gcc

## 1 在window上用别人编译好的binary 版

下载binary版 ，对照postgresql 目录，复制相应文件到对应文件中去



以plv8为例，对照 PG\_HOME 。

## 2 linux 上 pg\_xs  make

下载源代码 用 c/c++编译器编译

USE\_PGXS=1 make

USE\_PGXS=1 make install

## 3  pgxn 工具

pgxn是一款类似 pip一样的包管理工具，原理就是下载源代码，然后编译；

确定是 PGXN 的reposity不是那么全；所以综合来说，还是 手动make好

# 有哪些不能错过的插件

## 自带插件

查看官方文档，里面的包



## language包

plpython2u

plpython3u

plv8

pl/tcl pl/r

特别的 plpython

自带的情况下，postgresql09.6与 python3.3 对应 ，postgresql10 与python3.4对应

postgresql11与python3.6对应

window上如何 安装 plpython3u

centos7 上如何安装plpython3u

centos7上需要对python3重新编译，增加动态链接支持

192.168.3.171  postgres  since2015

## fdw

外部表，可以把数据库、文件、接口等数据源的数据，当作自己的表

### odbc\_fdw配置 mysql

实验平台window+postgresql10

1 在postgresql 上安装 odbc\_fdw

2 在 window上安装 myql odbc driver  ,配置数据源

3 postgresql 中创建外部表

/\* 1 创建server

drop server if exists mysql\_192\_168\_3\_172;

create server mysql\_192\_168\_3\_172 foreign data wrapper odbc\_fdw

options(dsn'mysql1',encoding'gbk');

/

/\*2 创建 用户映射

create user mapping for postgres server  mysql\_192\_168\_3\_172

options(odbc\_UID'root',odbc\_PWD'since2015');

\*/

定义要导入的表

import foreign schema mysql

from  server mysql

into fdw

options(

odbc\_DATABASE 'mysql',

table 'mysql',

sql\_query 'select \* from mysql.user'

);

odbc\_fdw配置 sqlserver

### file\_fdw

create server file\_server foreign data wrapper ;

drop foreign table if exists file;

create foreign table file (

jgmc text,

xzqh text,

lyd text

)

server file\_server options(format'csv',filename'C:\Users\Administrator\Desktop\pdoc\example.csv',header'true');;

tds\_fdw mssql 专属fdw

mongo\_fdw

hive\_fdw

redis\_fdw

hdfs\_fdw

hive\_fdw

fdw 可以用来做ETL ，可以用来做分析平台的读写，可以用来做中间件（负载均衡、sql分发）

ETL 想到技术体系问题，ETL 可以用kettle  用python  用web api 来做，也可以用postgresql 做

启示：数据分析的世界太浩瀚，我们不求面面俱到、所有技术都掌握，但求起码有一种我们很熟悉的一套体系；这个体系让我们知道数据分析产业的总体结构，知道

这个产业需要哪些功能；这个体系是我们最稳的一条路，是我们第一时间想到的，也是我们最后的一条路。有时任务紧急，要求的是立马实现什么功能，我们在没有时间和精力

去验证更好的方案时，能直接用上这个体系；

这样一个体系要满足什么条件，第一就是要全，再者精；mysql体系做不到，它不全，postgresql体系可以，python体系可以，java体系可以，hadoop体系可以；

它也足够精，它的效率是工业级别的；

先纵向，后横向；先全后精；

## madlib

apache顶级开源  机器学习模块

postgreql  greenplum  hawq

安装

预装plpythonu(python2)

yum install   madlibxxx.rpm

su postgres

/usr/local/madlib/bin/madpack -s madlib –p postgres install

使用 例子  线性回归

DROP TABLE IF EXISTS regr\_example;

CREATE TABLE regr\_example (

id int,

y int,

x1 int,

x2 int

);

INSERT INTO regr\_example VALUES

(1, 5, 2, 3),

(2, 10, 7, 2),

(3, 6, 4, 1),

(4, 8, 3, 4);

x1,x2   ---> y

训练

SELECT madlib.linregr\_train (

'regr\_example', -- source table

'regr\_example\_model', -- output model table

'y', -- dependent variable

'ARRAY[1, x1, x2]' -- independent variables

);

The intercept is computed by setting one of the independent variables to a constant 1, as shown in the preceding example.

array[1,x1,x2] 中的1  指的是 因变量维度为1

预测

SELECT regr\_example.\*,

madlib.linregr\_predict ( ARRAY[1, x1, x2], m.coef ) as predict,

y - madlib.linregr\_predict ( ARRAY[1, x1, x2], m.coef ) as residual

FROM regr\_example, regr\_example\_model m;

类比 python中的 sklearn  fit    predict

## pg\_jieba

分词模块

## http

爬虫模块

gist

rum

## 自定义插件

三个条件

写一个c语言库，丢到pg/lib里面去

写一个 .sql 文件，提供函数接口

写一个.control，定义版本和依赖关系

what i count create ,i cant not understand   Feiman