网络爬虫, Python和数据分析

王澎
中国科技大学

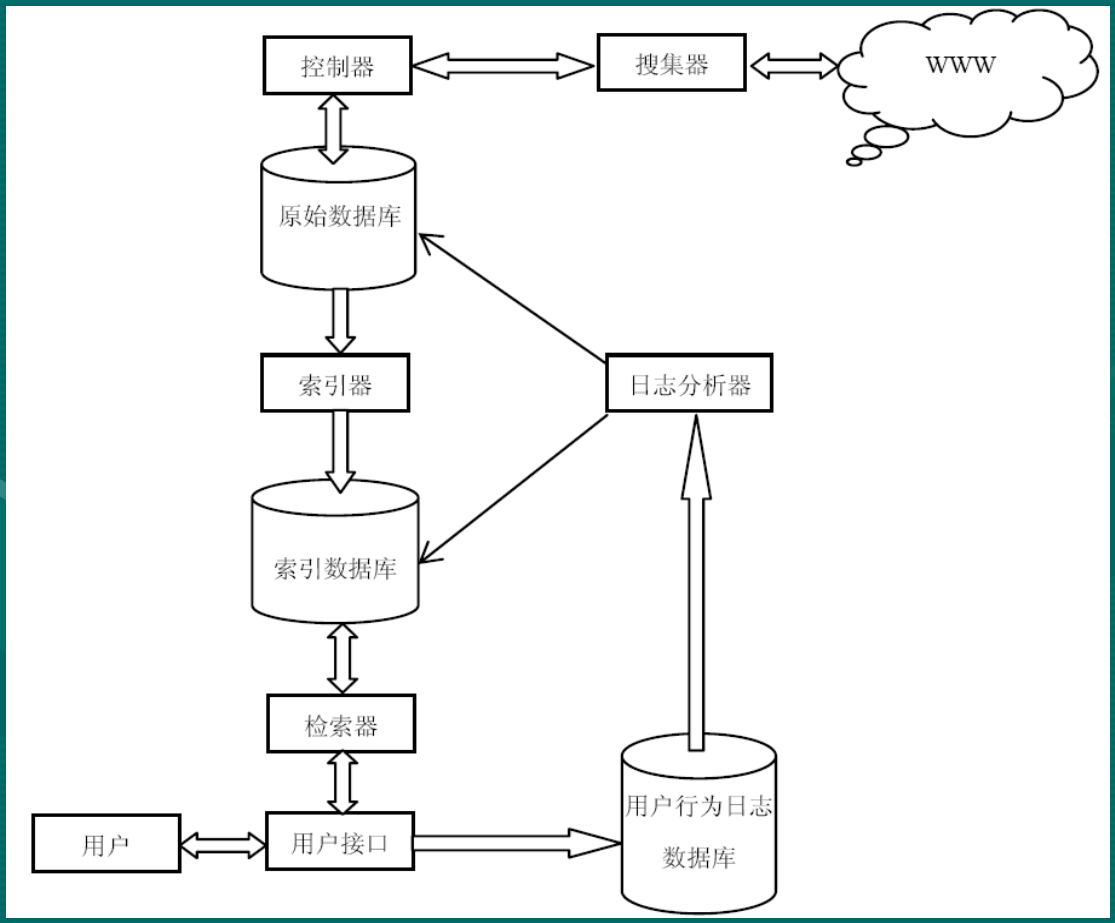
什么是网络爬虫？

- 网络爬虫是一个自动提取网页的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成。传统爬虫从一个或若干初始网页的URL开始，获得初始网页上的URL，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队列,直到满足系统的一定停止条件

爬虫有什么用？

- 做为通用搜索引擎网页收集器。（google,baidu）
- 做垂直搜索引擎.(找工作的搜索引擎: www.deepdo.com,数据来源于：www.51job.com , www.zhaoping.com , www.chinahr.com 等等)
- 科学研究：在线人类行为，在线社群演化，人类动力学研究，计量社会学，复杂网络，数据挖掘，等领域的实证研究都需要大量数据，网络爬虫是收集相关数据的利器。
- 偷窥，hacking，发垃圾邮件……（《google hack》….）

爬虫是搜索引擎的第一步
也是最容易的一步



- 网页搜集
- 建立索引
- 查询排序

用什么语言写爬虫？

- C，C++。高效率，快速，适合通用搜索引擎做全网爬取。缺点，开发慢，写起来又臭又长，例如：天网搜索源代码。
- 脚本语言：Perl, Python, Java, Ruby。简单，易学，良好的文本处理能方便网页内容的细致提取，但效率往往不高，适合对少量网站的聚焦爬取
- C#？（貌似信息管理的人比较喜欢的语言）

我曾经用来写过爬虫的语言

- Perl: 古老的脚本语言，hack 语言，被用来写爬虫有着悠久的历史，因此，书本支持相当丰富：《spidering hacks》，《Perl & LWP》；强大的文本处理能力，数据库支持能力。缺点：有点怪异。
- Python：相对年轻一点的语言。对于爬虫来说各方面能力还行，并且还在完善中，没有Perl那样有专门的爬虫书籍，不过网上能搜到一些文章。

为什么最终选择Python？

- 跨平台，对Linux和windows都有不错的支持。
- 科学计算，数值拟合：Numpy，Scipy
- 可视化：2d：Matplotlib(做图很漂亮), 3d: Mayavi2
- 复杂网络：Networkx
- 统计：与R语言接口：Rpy
- 交互式终端
- 网站的快速开发？

从一个简单的Python爬虫开始

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import re
import urllib2
import MySQLdb
from BeautifulSoup import BeautifulSoup

url1="http://bbs.ustc.edu.cn/cgi/bbstdoc?board=PieBridge&start=3558"#赋一个URL
fp= urllib2.urlopen(url1)       #打开此URL
s=fp.read()                     #把上操作的结果读出来赋给S
soup = BeautifulSoup(s)         #用Beautifulsoup分析S
polist=soup.findAll('span')     #找到所有tag <span>的内容
print polist[0].contents[0]     #打印出第一个tag <span>中间的内容
```

```
wisherg@wisherg-desktop:~/program/program2$ python tryclaw.py
瀚海星云
```

说明：加说明语句时要注意#需要英文编码里的，而不能是中文输入法中的#号，所以添加中文说明时先在英文输入法下打入#号后再切换到中文输入

瀚海星云Pie 版的网页部分代码

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<link rel="stylesheet" type="text/css" href="mycss?css=bbs.css" />
<script type="text/javascript" src="/right.js"></script>
</head>
<body class="postlist_body">
<div class="info">
    <span class="sitename">瀚海星云</span>
    <span>主题阅读讨论区: <strong>PieBridge</strong></span>
    <span>版主: <a href="bbsqry?userid=zdollar">zdollar</a></span>
    <span>文章数[5294]</span>
    <span>在线[68]</span>
    <span>主题[3640]</span>
</div>
```

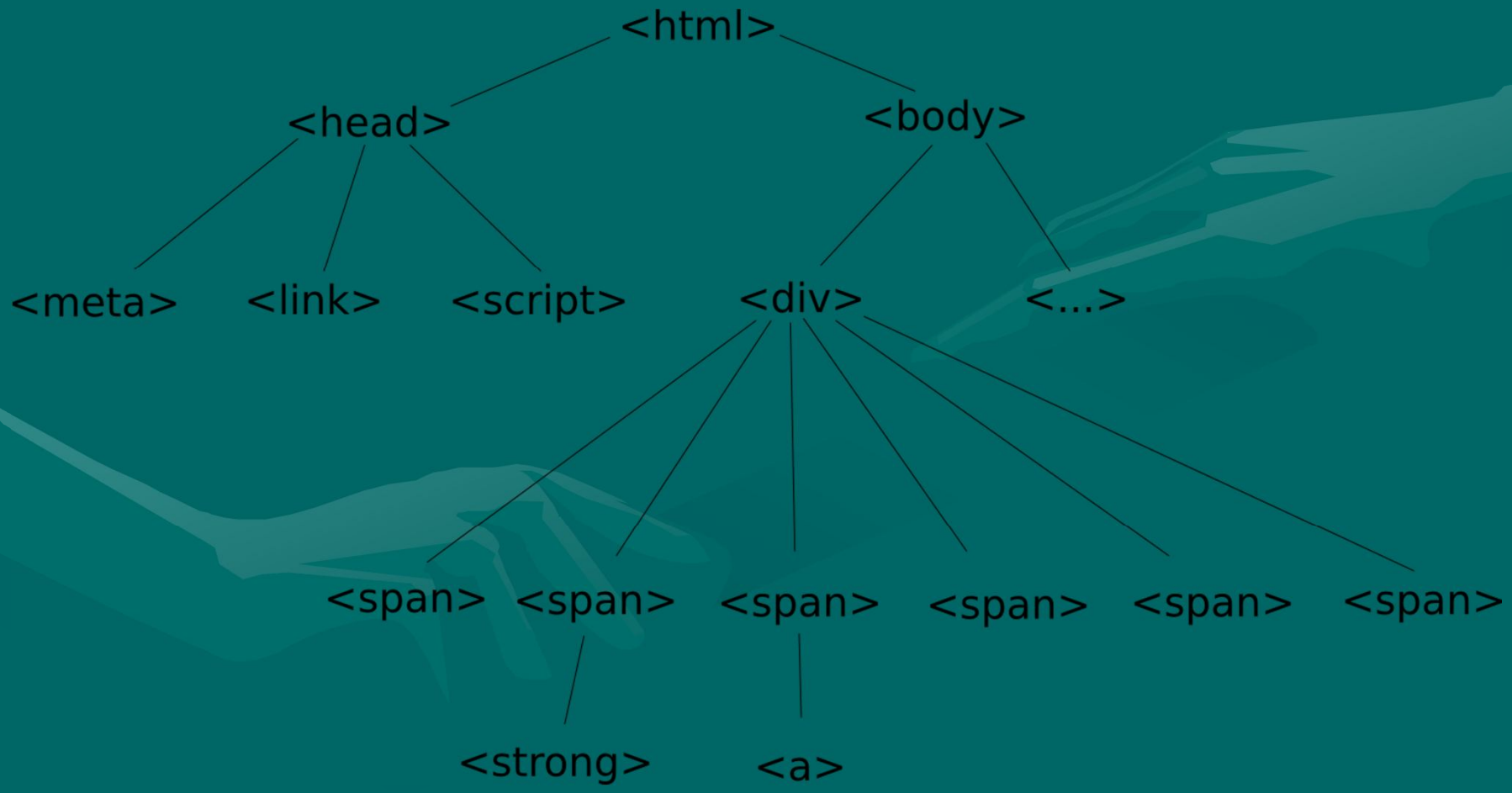
```
</body>
</html>
```

瀚海星云 主题阅读讨论区: **PieBridge** 版主: zdollar 文章数[5354] 在线[71] 主题[3645]

进版画面 | 文摘区 | 精华区 | 一般模式 | 主题模式 | XML

序号	状态	作者	日期	标题
▲置顶	G	licnep	Mon Sep 10	○ 鹊桥飞渡板FAQ
▲置顶	G	leinad	Thu Apr 16	○ 替MM征科大GG（有PP）
▲置顶	G	ayending	Wed Apr 15	○ 替邻居88mm征友（有pp）
▲置顶	G	DVI	Mon Apr 13	○ 代好友征GG（附PP哦）。
▲置顶	G	yxy	Sat Apr 11	○ 征gg(手机号已更正，十分
3626	G	leinad	Thu Apr 16	○ 替MM征科大GG（有PP）
3627		sworddancer	Thu Apr 16	○ 代一MM征BF

Pie版的Html树部分结构



利用BeautifulSoup剖析树

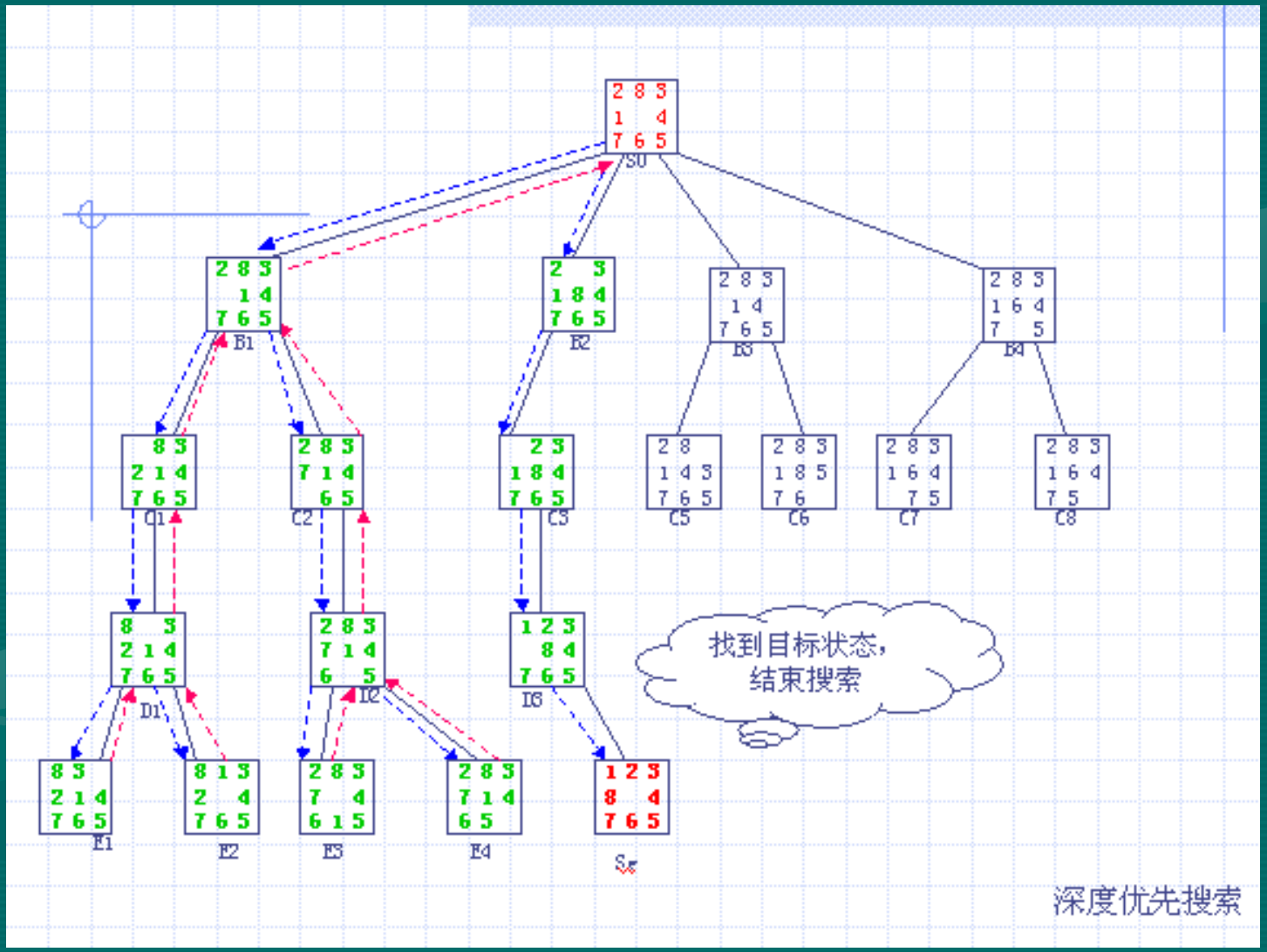
- 剖析树
  - Tags的属性
- Navigating 剖析树
  - parent
  - contents
  - string
  - nextSibling and previousSibling
  - next and previous
  - 遍历Tag
  - 使用标签名作为成员
- Searching 剖析树
  - The basic find method: findAll(name, attrs, recursive, text, limit, **kwargs)
    - 使用CSS类查找
    - 像 findall一样调用tag
  - find(name, attrs, recursive, text, **kwargs)
  - first哪里去了?
- Searching 剖析树内部
  - findNextSiblings(name, attrs, text, limit, **kwargs) and findNextSibling(name, attrs, text, **kwargs)
  - findPreviousSiblings(name, attrs, text, limit, **kwargs) and findPreviousSibling(name, attrs, text, **kwargs)
  - findAllNext(name, attrs, text, limit, **kwargs) and findNext(name, attrs, text, **kwargs)
  - findAllPrevious(name, attrs, text, limit, **kwargs) and findPrevious(name, attrs, text, **kwargs)
- Modifying 剖析树
  - 改变属性值
  - 删除元素
  - 替换元素
  - 添加新元素

FindAll()是最方便最好用的函数

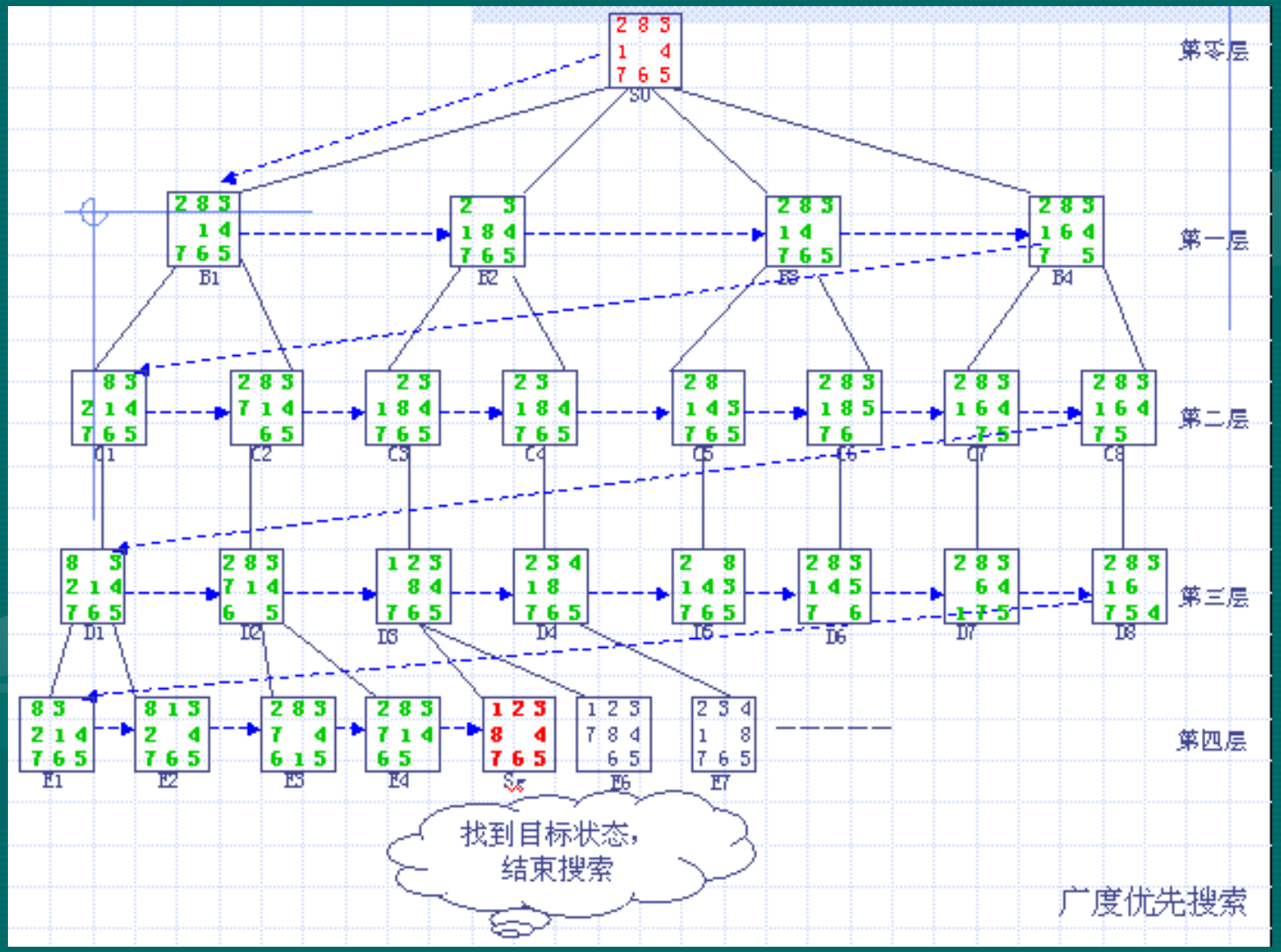
通用搜索策略

- 页面中的link

- 深度优先



- 广度优先



现实中的策略是多种多样的

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import re
import urllib2
import MySQLdb
from BeautifulSoup import BeautifulSoup

aaa=3640
url1="http://bbs.ustc.edu.cn/cgi/bbstdoc?board=PieBridge&start=" #赋一个URL
while aaa>0:
    aaa=aaa-20
    aaa1=str(aaa)
    url11=url1+aaa1
    fp= urllib2.urlopen(url11)      #打开此URL
    s=fp.read()                     #把上操作的结果读出来赋给S
    soup = BeautifulSoup(s)         #用Beautifulsoup分析S
    polist=soup.findAll('span')     #找到所有tag <span>的内容
    print polist[0].contents[0]     #打印出第一个tag <span>中间的内容
```

- 因为瀚海星云link有很简单的规律，每页递减20，所以利用这个规律设置每次赋入的URL，这样爬完了PIE版所有帖子

运行结果

```
wisherg@wisherg-desktop:~/program/program2$ python tryclaw1.py
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
å«ºÐÇÖÆ
瀚海星云
瀚海星云
```

```
瀚海星云
å«ºÐÇÔÆ
瀚海星云
瀚海星云
å«ºÐÇÔÆ
瀚海星云
瀚海星云
å«ºÐÇÔÆ
瀚海星云
瀚海星云
```

- 有乱码！！

爬取中文网页常有的问题：不规格的编码模式

- 解决方法：编码转换

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import re
import urllib2
import MySQLdb
from BeautifulSoup import BeautifulSoup

aaa=3640
url1="http://bbs.ustc.edu.cn/cgi/bbstdoc?board=PieBridge&start="
while aaa>0:
    aaa=aaa-20
    aaa1=str(aaa)
    url11=url1+aaa1
    fp= urllib2.urlopen(url11)
    try:
        s = fp.read ().decode ("gb2312",'ignore') #把 gb2312 修改成网页的编码
        #在这里添加一段代码，修改网页内容s的编码设置
        s = re.sub ("charset=gb2312", "charset=utf-8", s, re.I)
        s = s.encode ('utf-8', 'ignore')
    except:
        s = fp.read ()
    soup = BeautifulSoup(s)
    polist=soup.findAll('span')
    print polist[0].contents[0]
```

最后的结果

```
wisherg@wisherg-desktop:~/program/program2$ python tryclaw1.py
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
瀚海星云
```

- Perfect!

请温柔的对待瀚海星云！！

- 设置延迟时间（对于一个论坛，如果假设一个真实的浏览者每10秒翻开一个新的网页的话，一个不延时的爬虫每秒可以抓10个网页，这样一个爬虫相当于占用了100个人的带宽！）

```
In [ ]: import time

In [ ]: time.sleep(1)
```

- 在午夜爬取可以适当加快速度

道上的规矩：

1.3 平衡礼貌策略

爬虫相比于人，可以有更快的检索速度和更深的层次，所以，他们可能使一个站点瘫痪。不需要说一个单独的爬虫一秒钟要执行多条请求，下载大的文件。一个服务器也会很难响应多线程爬虫的请求。

就像Koster（Koster，1995）所注意的那样，爬虫的使用对很多工作都是很有用的，但是对一般的社区，也需要付出代价。使用爬虫的代价包括：

- 网络资源：在很长一段时间，爬虫使用相当的带宽高度并行地工作。
- 服务器超载：尤其是对给定服务器的访问过高时。
- 质量糟糕的爬虫，可能是服务器或者路由器瘫痪，或者会尝试下载自己无法处理的页面。
- 个人爬虫，如果过多的人使用，可能是网络或者服务器阻塞。

对这些问题的一个部分解决方法是漫游器排除协议（Robots exclusion protocol），也被称为robots.txt协议定书（Koster，1996），这份协议对于管理员指明网络服务器的那一部分不能到达是一个标准。这个标准没有包括重新访问一台服务器的间隔的建议，虽然访问间隔是避免服务器超载的最有效的办法。最近的商业搜索软件，如Ask Jeeves，MSN和Yahoo可以在robots.txt中使用一个额外的 “Crawl-delay”参数来指明请求之间的延迟。

对连接间隔时间的第一个建议由Koster 1993年给出，时间是60秒。按照这个速度，如果一个站点有超过10万的页面，即使我们拥有零延迟和无穷带宽的完美连接，它也会需要两个月的时间来下载整个站点，并且，这个服务器中的资源，只有一小部分可以使用。这似乎是不可以接受的。

Cho（Cho和Garcia-Molina, 2003）使用10秒作为访问的间隔时间，WIRE爬虫(Baeza-Yates and Castillo, 2002)使用15秒作为默认间隔。MercatorWeb(Heydon 和Najork, 1999)爬虫使用了一种自适应的平衡策略：如果从某一服务器下载一个文档需要t秒钟，爬虫就等待10t秒的时间，然后开始下一个页面。Dill等人 (Dill et al., 2002) 使用1秒。

对于那些使用爬虫用于研究目的的，一个更详细的成本-效益分析是必要的，当决定去哪一个站点抓取，使用多快的速度抓取的时候，伦理的因素也需要考虑进来。

访问记录显示已知爬虫的访问间隔从20秒钟到3-4分钟不等。需要注意的是即使很礼貌，采取了所有的安全措施来避免服务器超载，还是会引来一些网络服务器管理员的抱怨的。Brin和Page注意到：运行一个针对超过50万服务器的爬虫，会产生很多的邮件和电话。这是因为有无数的人在上网，而这些人不知道爬虫是什么，因为这是他们第一次见到。（Brin和Page，1998）

用Mysql存储数据

- 先要在自己数据库里建立一个空的表，这里，这里我已经建立了一个名为lilybbs的数据库，表名为hunan_a
- 导入相应的模块

```
import MySQLdb
```

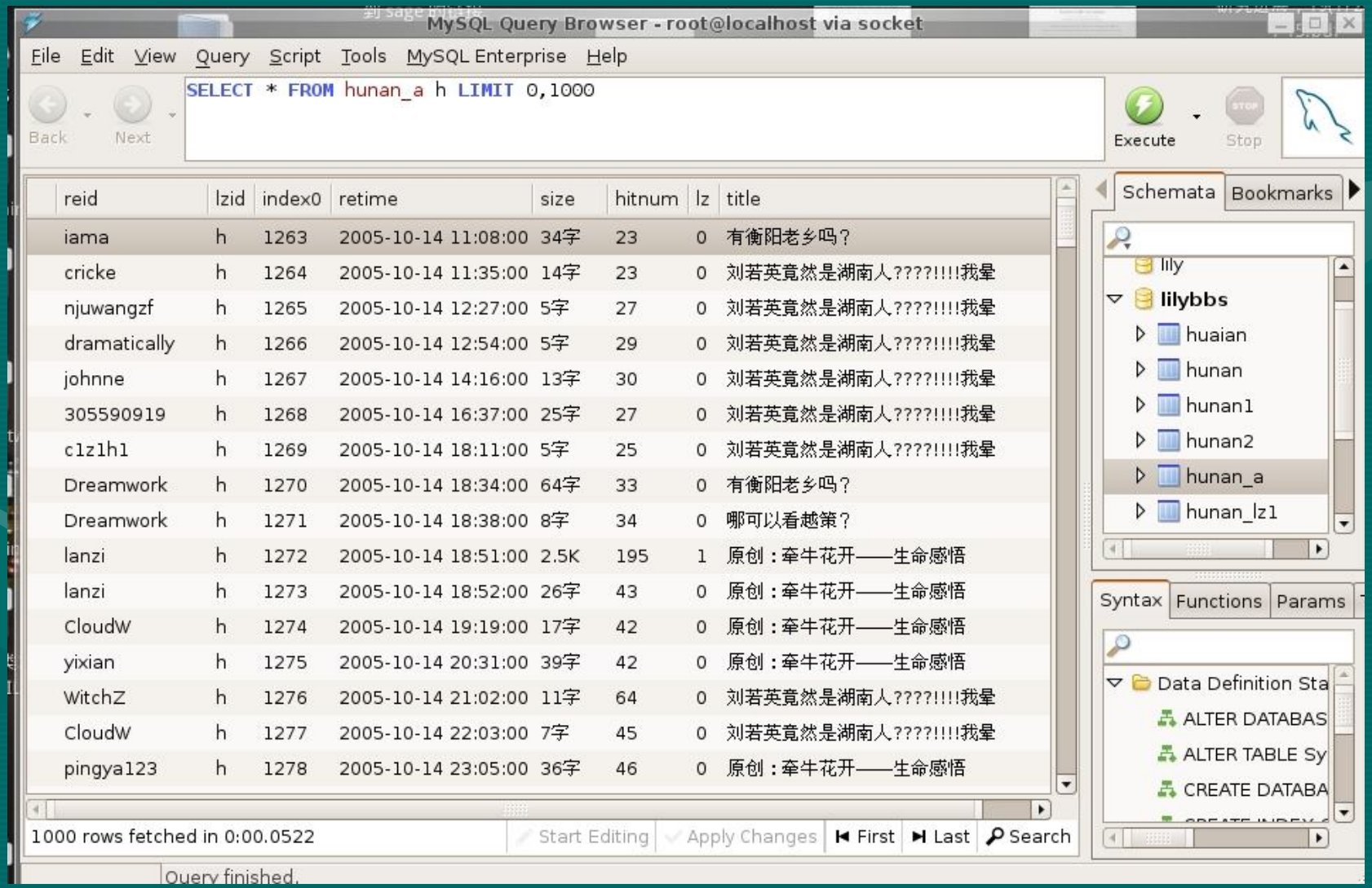
- 与相应的数据库连接

```
db=MySQLdb.connect
(host="localhost",user="root",passwd="0101",db="lilybbs",use_unicode=1,
charset='utf8')
cursor=db.cursor()
```

- 写入

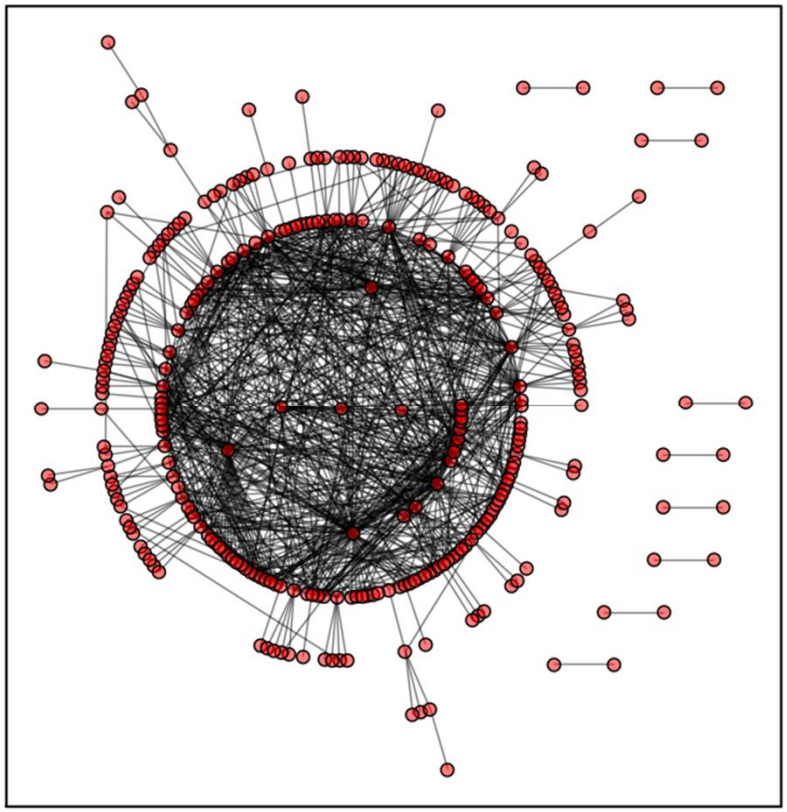
```
    for i in range(20):
      cursor.execute("insert into hunan_a values (%s,%s,%s,%s,%s,%s,%s,%s)",
(id[i], 'h', index[i], time1[i], size[i], hit[i], lz[i], title1[i]))
```

数据库里的结果

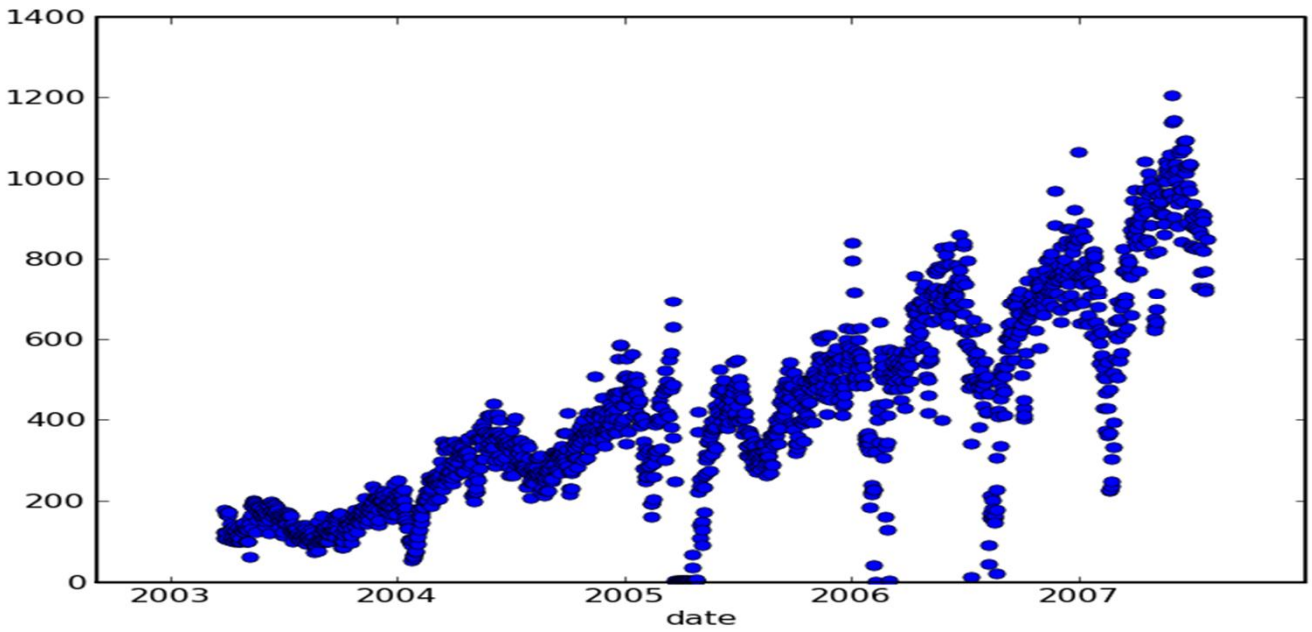

MySQL Query Browser - root@localhost via socket
File Edit View Query Script Tools MySQL Enterprise Help
SELECT * FROM hunan_a h LIMIT 0,1000
Back
Next
Execute
Stop
reid lzid index0 retime size hitnum lz title
iama h 1263 2005-10-14 11:08:00 34字 23 0 有衡阳老乡吗？
cricke h 1264 2005-10-14 11:35:00 14字 23 0 刘若英竟然是湖南人????!!!!我晕
njuwangzf h 1265 2005-10-14 12:27:00 5字 27 0 刘若英竟然是湖南人????!!!!我晕
dramatically h 1266 2005-10-14 12:54:00 5字 29 0 刘若英竟然是湖南人????!!!!我晕
johnne h 1267 2005-10-14 14:16:00 13字 30 0 刘若英竟然是湖南人????!!!!我晕
305590919 h 1268 2005-10-14 16:37:00 25字 27 0 刘若英竟然是湖南人????!!!!我晕
c1z1h1 h 1269 2005-10-14 18:11:00 5字 25 0 刘若英竟然是湖南人????!!!!我晕
Dreamwork h 1270 2005-10-14 18:34:00 64字 33 0 有衡阳老乡吗？
Dreamwork h 1271 2005-10-14 18:38:00 8字 34 0 哪可以看越策？
lanzi h 1272 2005-10-14 18:51:00 2.5K 195 1 原创：牵牛花开——生命感悟
lanzi h 1273 2005-10-14 18:52:00 26字 43 0 原创：牵牛花开——生命感悟
CloudW h 1274 2005-10-14 19:19:00 17字 42 0 原创：牵牛花开——生命感悟
yixian h 1275 2005-10-14 20:31:00 39字 42 0 原创：牵牛花开——生命感悟
WitchZ h 1276 2005-10-14 21:02:00 11字 64 0 刘若英竟然是湖南人????!!!!我晕
CloudW h 1277 2005-10-14 22:03:00 7字 45 0 刘若英竟然是湖南人????!!!!我晕
pingya123 h 1278 2005-10-14 23:05:00 36字 46 0 原创：牵牛花开——生命感悟
1000 rows fetched in 0:00.0522
Start Editing
Apply Changes
First
Last
Search
Query finished.
Schemata
Bookmarks
lily
lilybbs
huaian
hunan
hunan1
hunan2
hunan_a
hunan_lz1
Syntax
Functions
Params
Data Definition Sta
ALTER DATABAS
ALTER TABLE Sy
CREATE DATABA

统计和做图

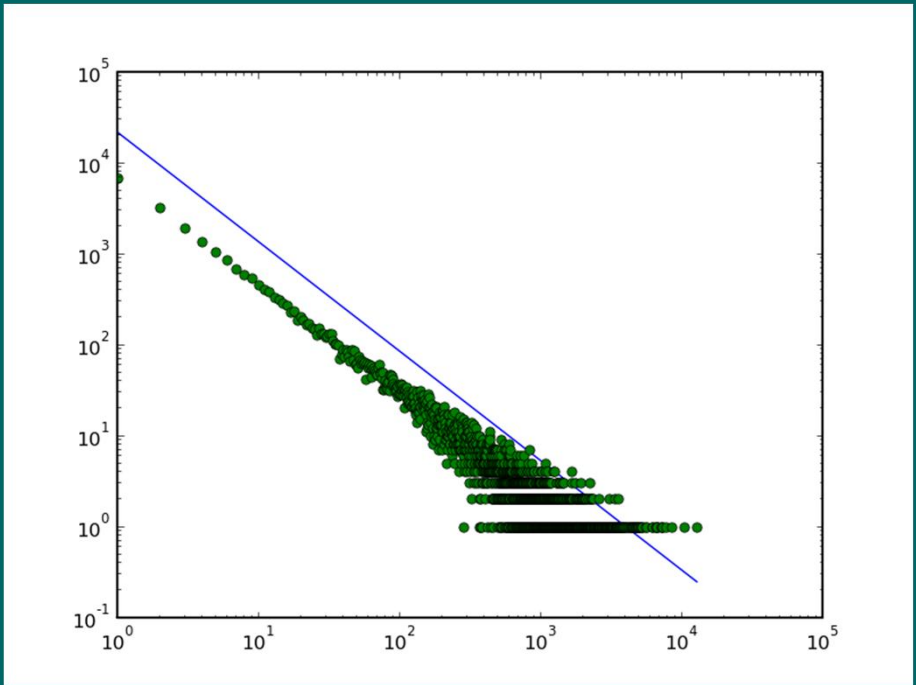
- 这部分主要用于科研方面，利用爬取到的数据做一些简单的统计工作
- 右图是某论坛的回复网络，使用python的networkx包做的。

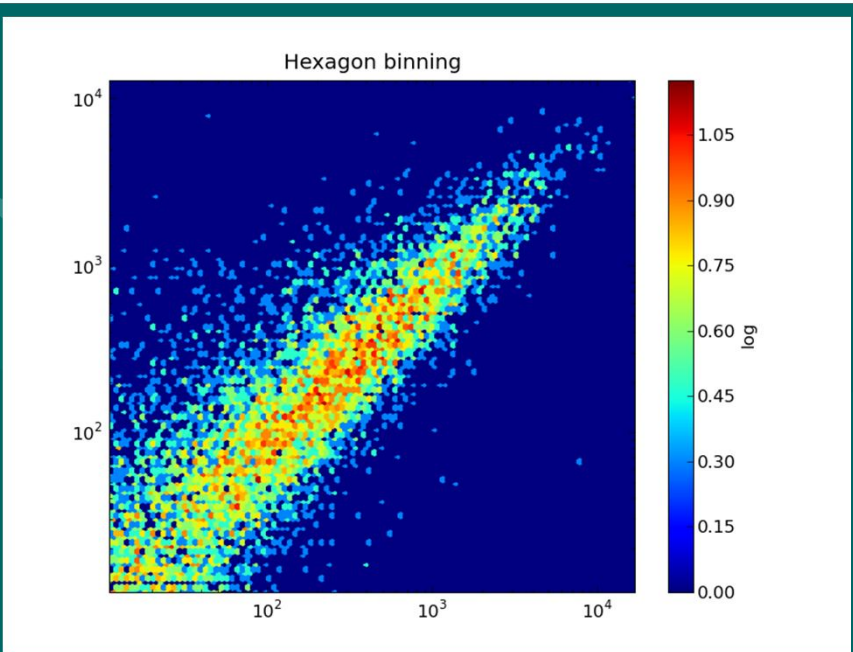


```
#!/usr/bin/env python

import MySQLdb
import pylab

db=MySQLdb.connect(host="localhost",user="root",passwd="0101",db="lily")
cursor=db.cursor()
cursor.execute("select patime, count(*) from paperlist where patime >
'2000-01-01' and patime<'2007-07-25' group by patime")
result=cursor.fetchall()          #把cursor执行得到的值赋给result，是有两个元素的list
dates1=[q1[0] for q1 in result]   #把result的第一个元素赋给dates1
num1=[q1[1] for q1 in result]     #把result的第二个元素赋给num1
pylab.plot_date(dates1,num1,'o')  #以dates1为x，num1为y，做日期图
pylab.show()                      #显示做出来的图
```



- Pylab 是matplotlib作图包的一部分
- 左图是某blog四年间每天发表文章的数量





- 左一是某blog网站每个blog评论数的统计，x是blog评论数量，y是有这样数量的blog的数量。可以看到是标准的“power-law”分布，幂指数为-1.2左右，拟合使用了Scipy包的optimize.leastsq, 函数，具体可见scipy cookbook页面的fitting data一栏

- 左二是blog的comment networks 的入度与出度的散点图，也就是每个点的坐标x，y分别代表某个人获得的评论和发出的评论数。颜色代表这样点的数量。本图使用了matplotlib中的hexbin函数

高级主题（一）：编写更健壮的爬虫

```
#!/usr/bin/env python

import httplib, re, urllib2, socket

Accept = "image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*"
AcceptLanguage = "zh-cn"
UserAgent = "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"



def get (url, data = None, headers = None, referer = None):
   request = urllib2.Request (url)
   if data :
       request.add_data (data)
   if headers:
       for key in headers.keys ():
           request.add_header (key, headers[key])
   if not request.has_header ("User-Agent"):
       request.add_header ("User-Agent", UserAgent)
   if not request.has_header ("Referer") and referer :
       request.add_header ("Referer", referer)
   if not request.has_header ("Accept"):
       request.add_header ("Accept", Accept)
   if not request.has_header ("Accept-Language"):
       request.add_header ("Accept-Language", AcceptLanguage)
   if None and not request.has_header ("Accept-Encoding"):
       request.add_header ("Accept-Encoding", AcceptEncoding)

   method = request.get_method ()
   try:
       result = urllib2.urlopen (request)
       return result
   except urllib2.URLError, error:
       pass
   except socket.error, error:
       pass
   except:
       pass
   return None
```

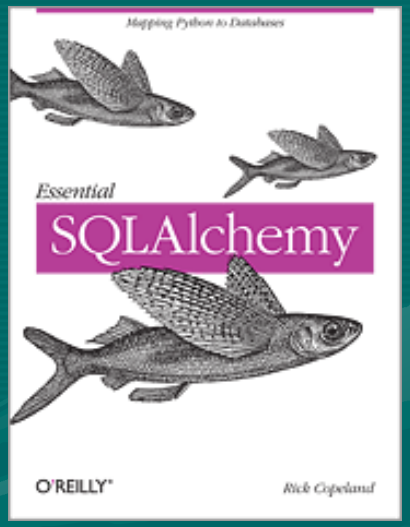
- 伪装成浏览器
- 容错

高级主题（二）：由内嵌脚本产生的动态网页的爬取

- 如何爬取像左图这样的网页呢？
- 它显示的内容并不会呈现在html文件里。

高级主题（三）：SQLAlchemy

Mysql这样关系型数据库的缺点：在表示复杂网络这样一对多，和多对多的关系时，非常冗余；一旦需要做比较复杂的统计，sql语句会变得异常复杂。



- 当你越关注性能，就会发现 SQL 数据库离对象集合越来越远；当你越关注抽象，就会发现对象集合离表和行这些概念越来越远。SQLAlchemy 将致力于尽量包容这两个世界。

- SQLAlchemy 并不把数据库简单地视为数据表的集合；它把它们看作是关系代数引擎。它的关系对象映射能够让类以不同的方式映射到数据库。SQL 工具包也不光能够对数据表进行 select 操作——你还能对连接、子查询和联合进行 select。这样数据库关系和领域对象模型之间的耦合从一开始就得以很好地解开，使得两个领域都得以发挥其各自的极致。

- 我写过的某个繁琐的调用

```
select group_concat(cmm), case when ascii(comperson) > ascii(author) then
concat(comperson,author) else concat(author,comperson) end as aucom1,
count(*) as cmm1 from (select comperson, author, count(*) as cmm from
commentlist where author < 'a' and comperson <> author group by comperson,
author) as aucom group by aucom1 having cmm1 > 1
```

- 号称能更简洁明了的SQLAlchemy会成为mysql的替代品么？

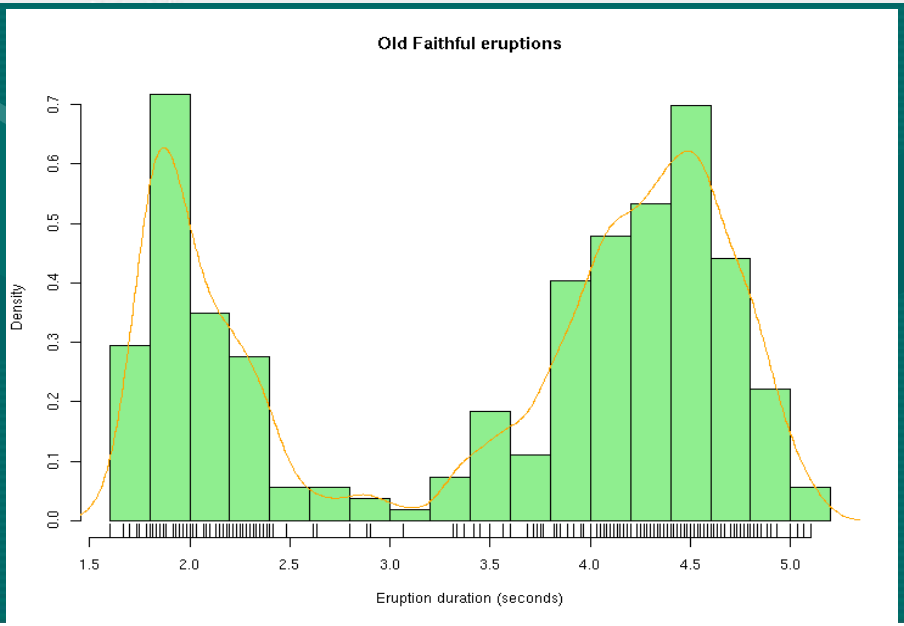
```
>>> query = query.column(addresses).select_from(users.outerjoin(addresses)).apply_labels()
```

Let's bake for .0001 seconds and see what rises:

```
>>> conn.execute(query).fetchall()
SELECT users.id AS users_id, users.name AS users_name, users.fullname AS users_fullname, addresses.id AS addresses_id,
FROM users LEFT OUTER JOIN addresses ON users.id = addresses.user_id
WHERE users.name = ? AND (EXISTS (SELECT addresses.id
FROM addresses
WHERE addresses.user_id = users.id AND addresses.email_address LIKE ?)) ORDER BY users.fullname DESC
['jack', '%@msn.com']
[(1, u'jack', u'Jack Jones', 1, 1, u'jack@yahoo.com'), (1, u'jack', u'Jack Jones', 2, 1, u'jack@msn.com')]
```

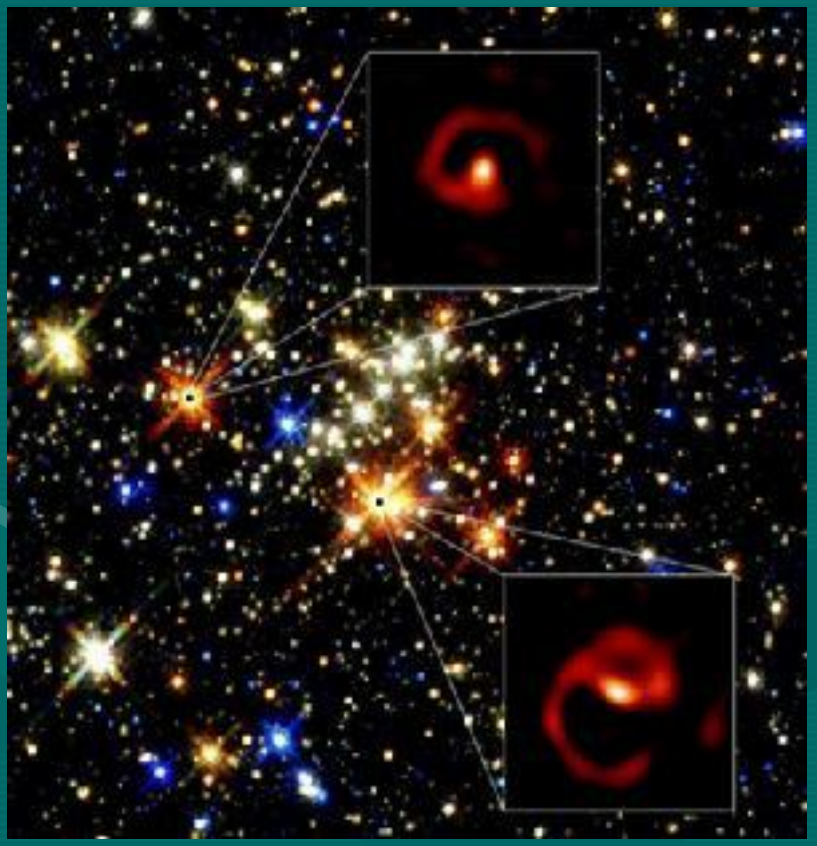
高级主题（四）：统计利器R语言

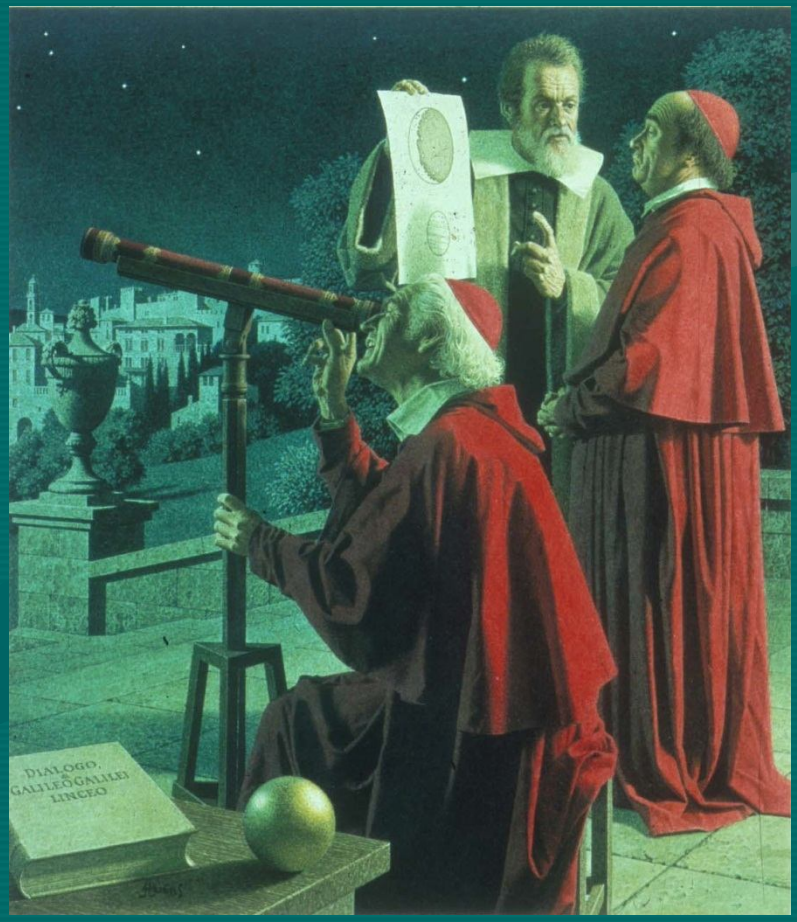
```
from rpy import *

r.png('faithful_histogram.png',width=733,height=550)
r.hist(ed,r.seq(1.6, 5.2, 0.2), prob=1,col="lightgreen",
     main="Old Faithful eruptions",xlab="Eruption duration (seconds)")
r.lines(r.density(ed,bw=0.1),col="orange")
r.rug(ed)
r.dev_off()
```



- 求方差，聚类，判别，拟合，团簇探测，时间序列分析，生存分析，甚至复杂网络，这些R语言里都有很好的函数
- 可以直接使用R语言，也可以利用Rpy在python里面调用R的函数，不过Rpy仍然开发中，还不是很成熟

曾经我们获取数据的手段：
我们用望远镜来洞察宇宙





昂贵的实验
只是为了获取大自然的数据

Internet 带给我们了海量的数据
善用数据，了解我们自己

Google's Hidden Wealth

Type the word "science" into the Google search engine, and a list of one-and-a-half million Web pages appears in a fraction of a second. Behind this service lies an enormous reservoir of data that researchers would like to harness for science of their own, in fields from social psychology to global economics. But although some computer-based companies such as Microsoft have eagerly embraced scientific collaboration, Google so far has not. "Google has a reputation ... for being very negative to letting researchers in," says Richard Swedberg, a sociologist at Cornell University. This could soon change, a Google spokesperson has told *Science*.

Google's data are a potential social science gold mine, "both for observing social interactions in real time and also for measuring their consequences for individual and collective behavior," says Duncan Watts, a sociologist at Columbia University. The key is the electronic "cookie." As you browse the Internet, many Web sites such as Google's record a string of text—the cookie—representing the identity of your computer. And when you use Google, its servers keep track not only of what you

www.sciencemag.org SCIENCE VOL 314 10 NOVEMBER 2006
Published by AAAS

浩瀚的比特海是另一片未知的星空

谢谢大家！