# 实验报告

计算技术研究所-201728013229100-徐心仪

## 一．系统设计简述

本系统采用B/S架构，由服务端和数据端组成，其中服务端采用django模板渲染机制处理网页逻辑，数据端使用gserver。

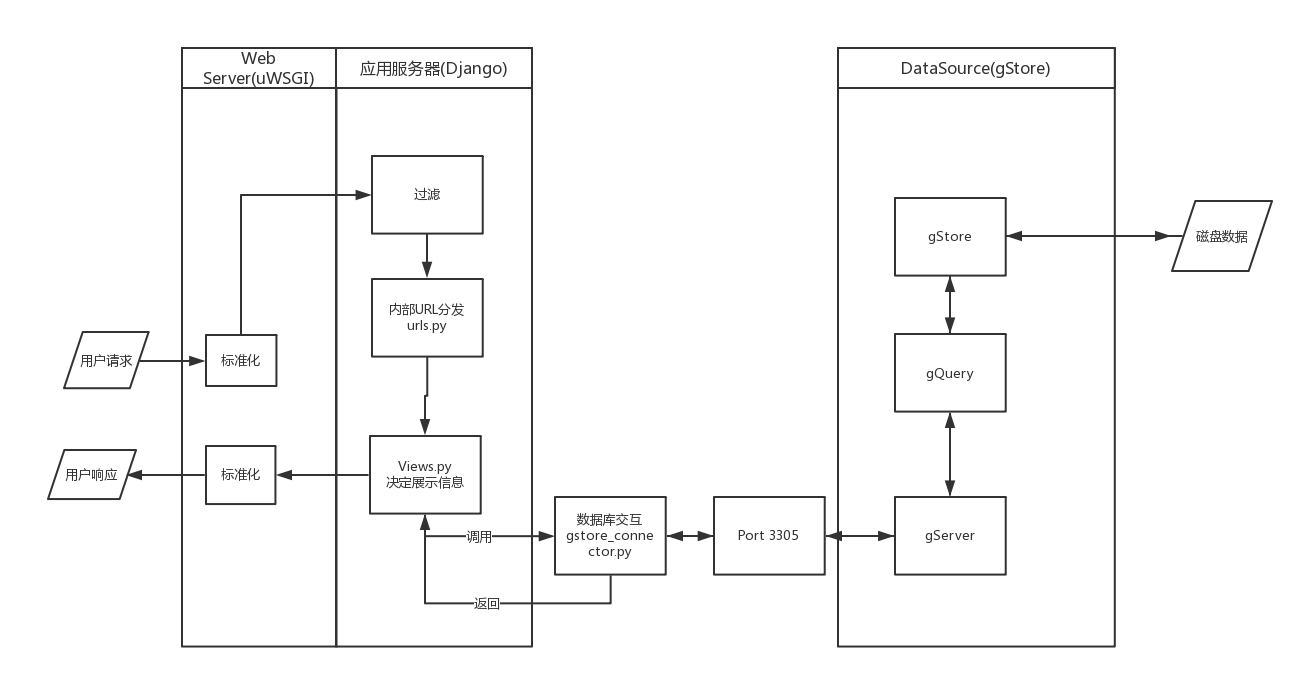
### 运行环境

整体机器: windows10, 8核i7，内存8G

服务端: windows10, python2 + django

数据端: ubuntu16.04 + gstore

### 主要架构及模块



## 二．应用系统界面

### 查找



### 列出

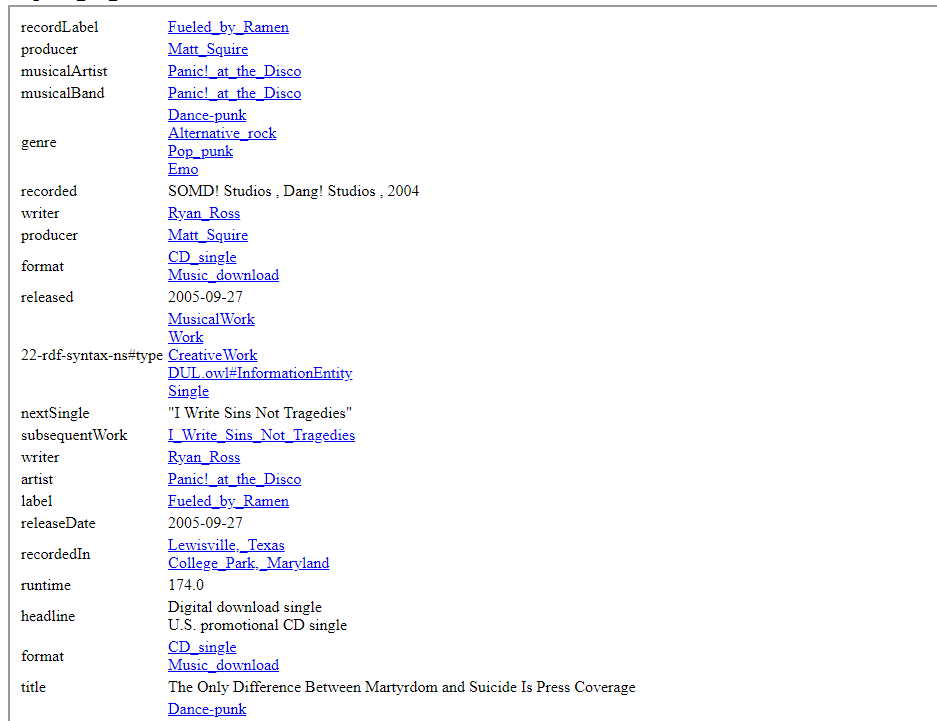
单击条目，可以列出下方详情



### 详情

列出条目涉及的谓语和宾语

单击链接可以转到详情



## 三．SPARQL查询语句举例

### 查询流派

query select distinct ?genre

where {

?x <http://dbpedia.org/ontology/genre> ?genre.

?x <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/MusicalWork>.}

### 查询乐曲

query

select distinct ?uri

where {

?uri <http://dbpedia.org/ontology/genre> <http://dbpedia.org/resource/Dance-punk>.

?uri <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/Album>.

?uri <http://xmlns.com/foaf/0.1/name> "But It's Better If You Do"@en.}

LIMIT 200

### 查询细节

query

select distinct ?pred ?objc

where {

<http://dbpedia.org/resource/The\_Only\_Difference\_Between\_Martyrdom\_and\_Suicide\_Is\_Press\_Coverage> ?pred ?objc. }

## 四．关键系统代码

系统没有什么特别/关键的代码

例如这是用于与gserver通信的代码

@staticmethod

def get\_instance():

if GstoreConnector.\_instance is None:

GstoreConnector.\_instance = GstoreConnector('192.168.5.136', 3305)

GstoreConnector.\_instance.unload()

GstoreConnector.\_instance.load('dbpedia\_1')

return GstoreConnector.\_instance

@staticmethod

def get\_subjects(name, res\_type, genre):

where\_sql = ''

if not name and not res\_type and not genre:

return []

if not res\_type:

return GstoreConnector.get\_subjects(name, 'Album', genre) \

+ GstoreConnector.get\_subjects(name, 'Single', genre)

name = name.replace('"', '\"').replace('\_', ' ')

if genre:

where\_sql += '?uri <http://dbpedia.org/ontology/genre> <{}>.\n'.format(genre)

if res\_type:

where\_sql += '?uri <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://dbpedia.org/ontology/{}>.\n'.format(

res\_type)

if name:

where\_sql += '?uri <http://xmlns.com/foaf/0.1/name> "{}"@en.\n'.format(name)

sparql = '''

select distinct ?uri

where {%s}

LIMIT 200

''' % where\_sql

answer\_obj = GstoreConnector.get\_instance().query(sparql)

assert isinstance(answer\_obj, list)

for ele in answer\_obj:

ele['res\_type'] = res\_type

ele['name'] = ele['uri'][ele['uri'].rindex('/') + 1:].replace('%27', '\'').replace('%22', '\"').replace('%20', ' ')

return answer\_obj

## 五．其他

1. 在json解码中遇到困难，因此修改了gQuery源码：<https://github.com/Caesar11/gStore/pull/33>

2. 由于虚拟机内存不够，因此仅仅使用了700mb的音乐数据子集用于项目展示。