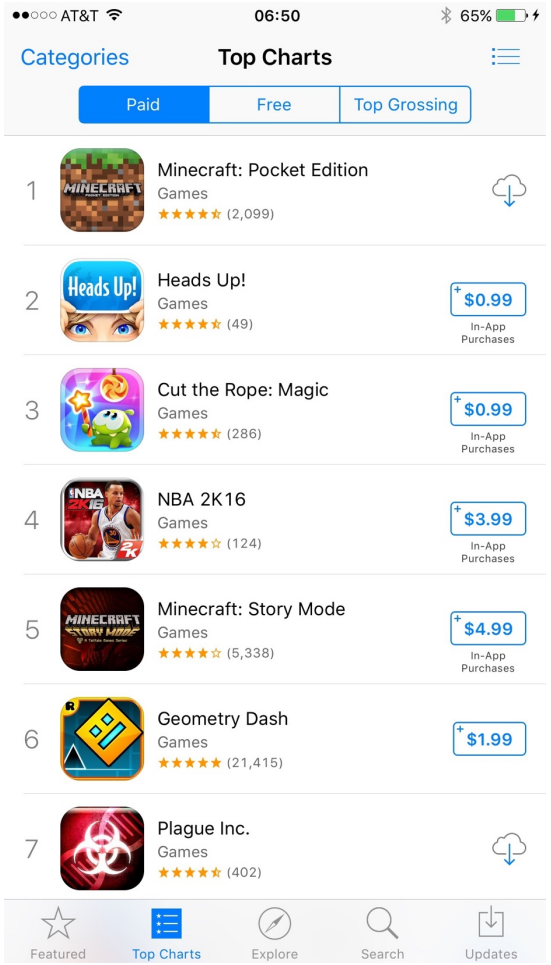
Tiger Micro-Project

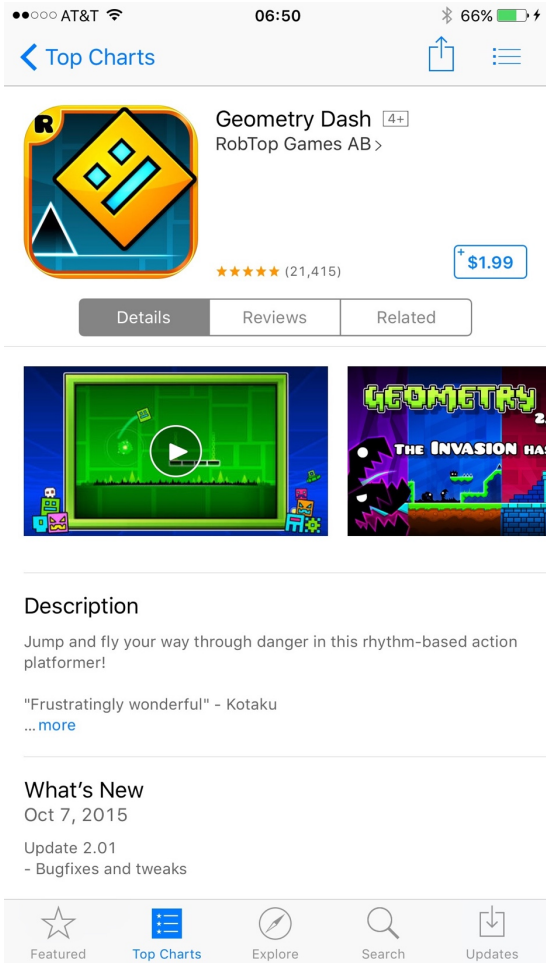
Design & Implementation of APP Store

Qinyuan Feng

Tiger App Store

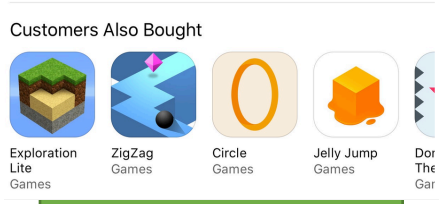


List page



Detail page

Architecture

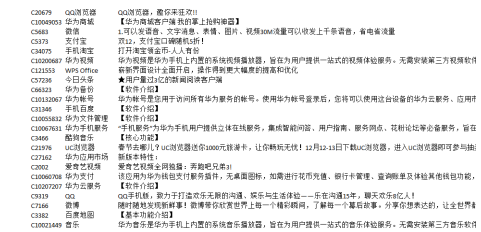


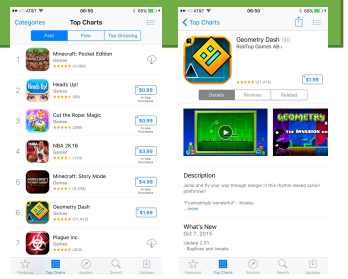
Recommender

Crawler

Database

Front-end





Crawler - goal

- Crawl 1,000,000 App information from Huawei App Store

C20679	QQ浏览器	QQ浏览器，邀你来狂欢!!
C10049053	华为商城	【华为商城客户端 我的掌上抢购神器】
C5683	微信	1.可以发语音、文字消息、表情、图片、视频30M流量可以收发上千条语音，省电省流量
C5373	支付宝	双12，支付宝口碑随机5折！
C34075	手机淘宝	打开淘宝领金币-人人有份
C10200687	华为视频	华为视频是华为手机上内置的系统视频播放器，旨在为用户提供一站式的视频体验服务。

Skill

- Python, Scrapy, MongoDB, Proxy, Scrapyjs

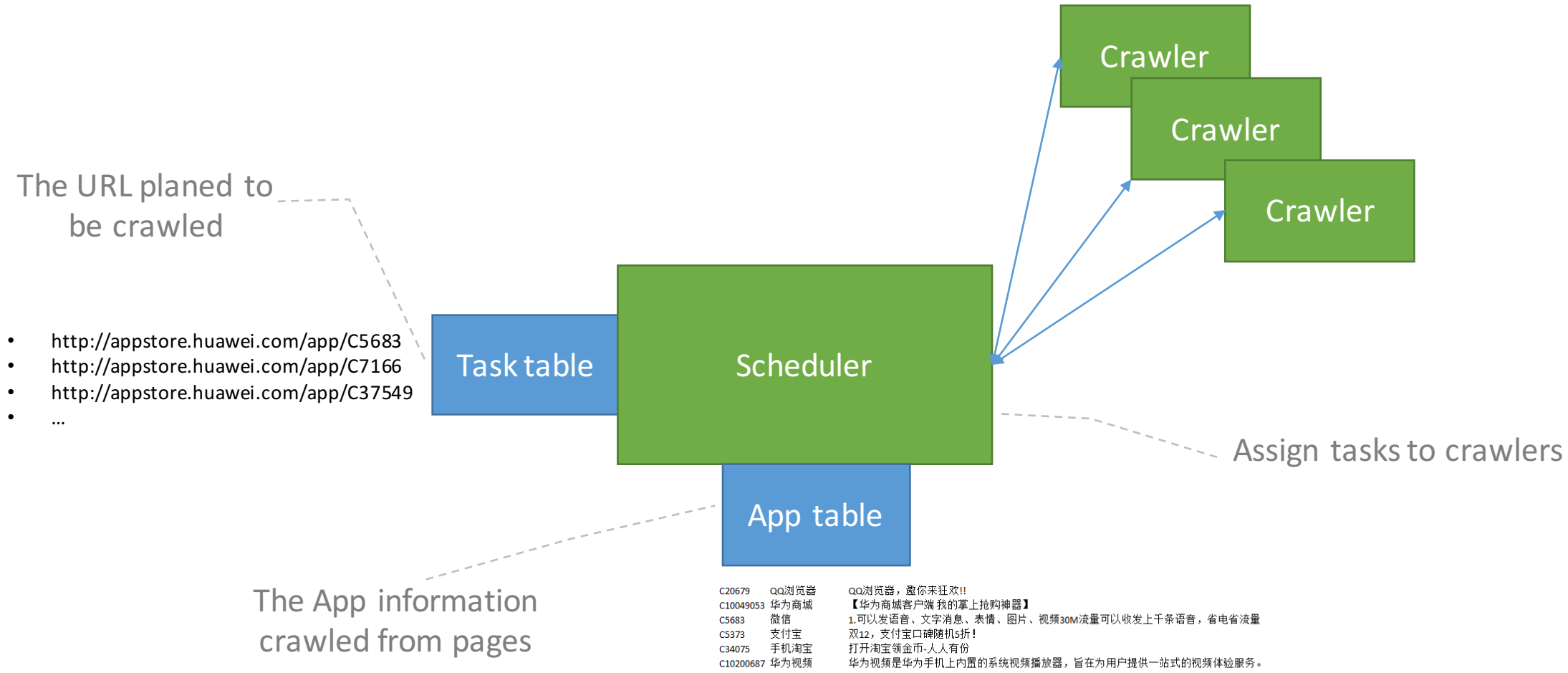
Performance

- 100 pages/second (vs 30K/second)

Terms

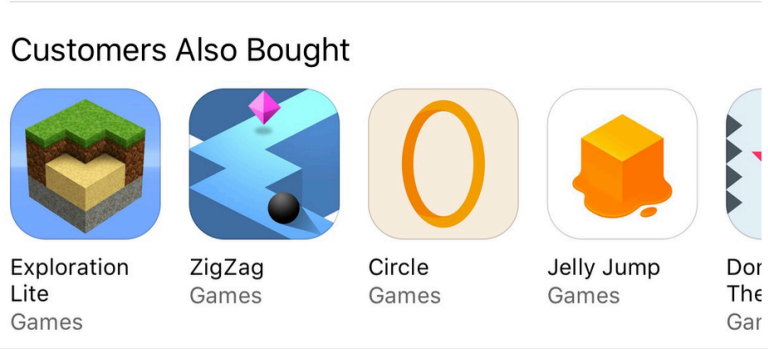
- Python: a programming language focus on readability and less code
- Scrapy: open source framework of crawlers based on Python
- MongoDB: document-oriented database
- Proxy: bypass the constrain of web server by changing your IP address
- Scrapyjs: help to process JavaScript in Scrapy

Crawler - architecture



Recommender - goal

- Recommend ten Apps related for each App



Skill

- Python/Java, collaborative-filtering algorithm, cosine-similarity

Performance

- 1 second/app (vs with 10ms/app)

Recommender - architecture

All data about Apps

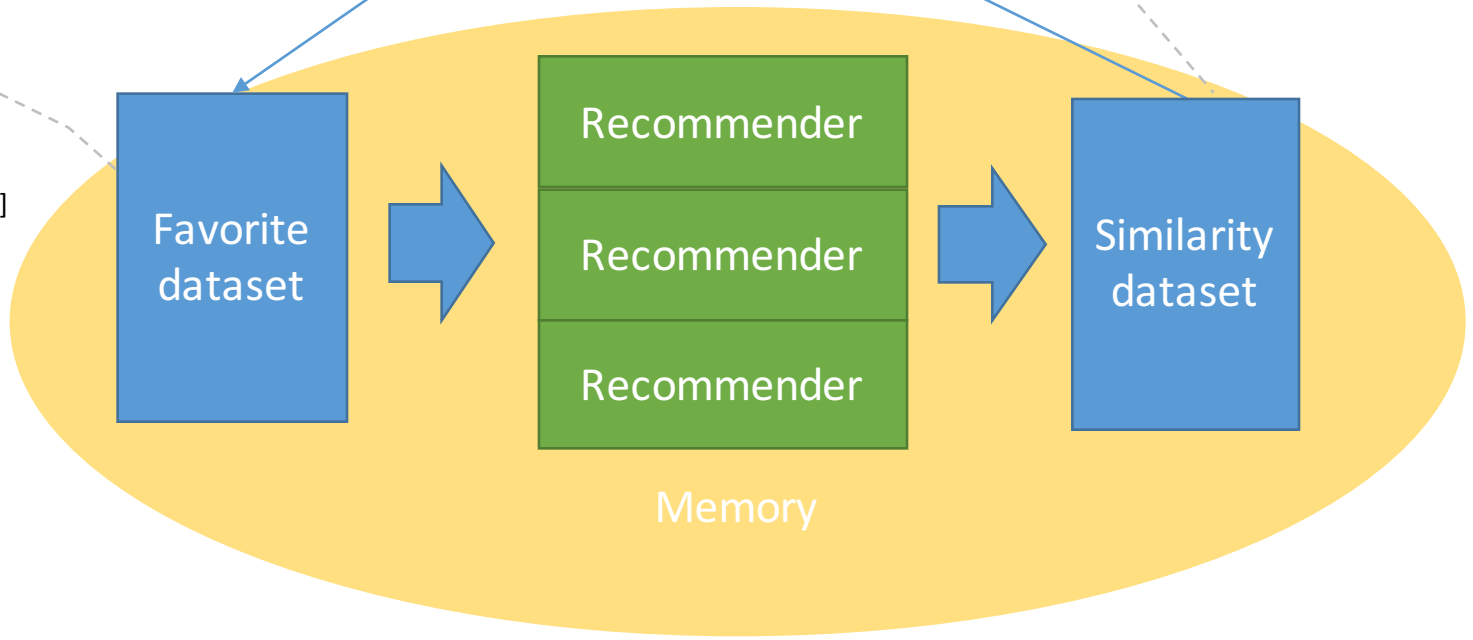
MongoDB

The lists of each App's most similar Apps

- WeChat similar to [WhatsApp, Facebook messenger…]
- Sohu similar to [Yahoo news, Medium …]
- AngryBirds similar to [Candy crush sage, Cut the rope …]
- …

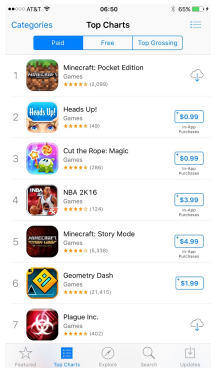
The lists of each user's favorite Apps

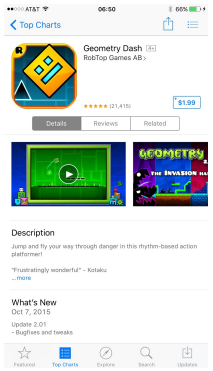
- U1 likes [WeChat, Sohu, Sina, Facebook]
- U2 likes [WhatsApp, Google, Chrome, WeChat]
- U3 likes [Chrome, AngryBirds, WeChat, Slack]
- …



Front-end goal

- Display the list-page and detail-page





Skill

- Java, Spring MVC, Hibernate
- JavaScript, Node.js, Meteor, MongoDB

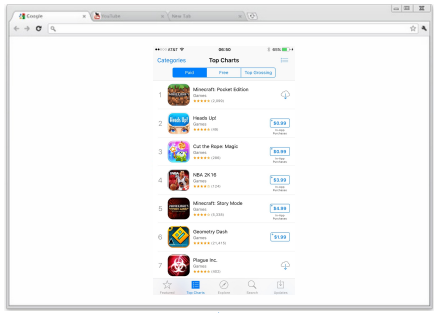
Performance

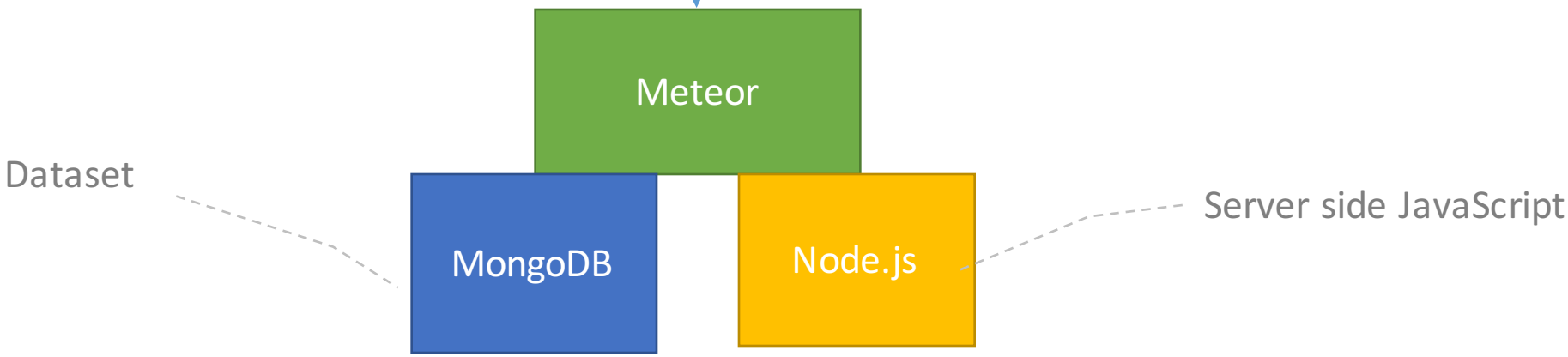
- 1k QPS (vs 10K QPS)

Terms

- JavaScript: the scripting language for Web pages
- Node.js: server side JavaScript
- Meteor: is a complete platform for building web and mobile apps in pure JavaScript

Front-end architecture





Summary

- Tiger micro project: "App Store"
  - Crawler
    - Goal: Crawl 1,000,000 App information from Huawei App Store
    - Skill: Python, Scrapy, MongoDB, Proxy, Scrapyjs
    - Performance: 100 pages/second
  - Recommender
    - Goal: Recommend ten Apps related for each App
    - Skill: Python/Java, collaborative-filtering algorithm, cosine-similarity
    - Performance: 1 second/app
  - Front-end
    - Goal: Display the list-page and detail-page
    - Skill: Java, Spring MVC, Hibernate/ JavaScript, Node.js, Meteor, MongoDB
    - Performance: 1k QPS

Timeline

2015 Dec. 19

- Product design

2016 Jan. 3

- Crawler implementation

2016 Jan. 9

- Recommender implementation

2016 Jan. 16

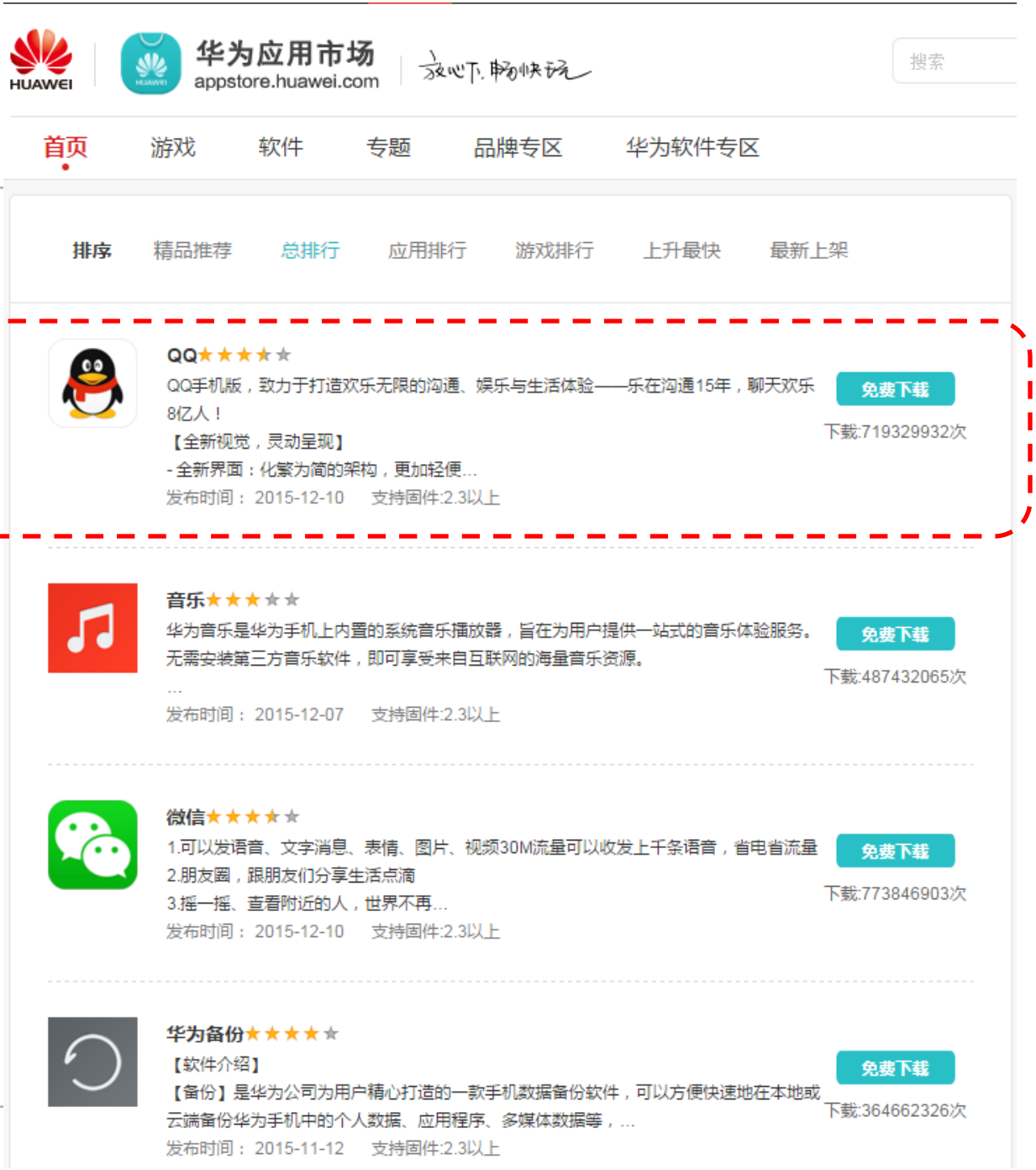
- Front-end implementation

Scrapy – Crawl the Web

Tiger AppStore

http://appstore.huawei.com/more/all

- Name
- Thumbnail
- Description
- Published
- … …

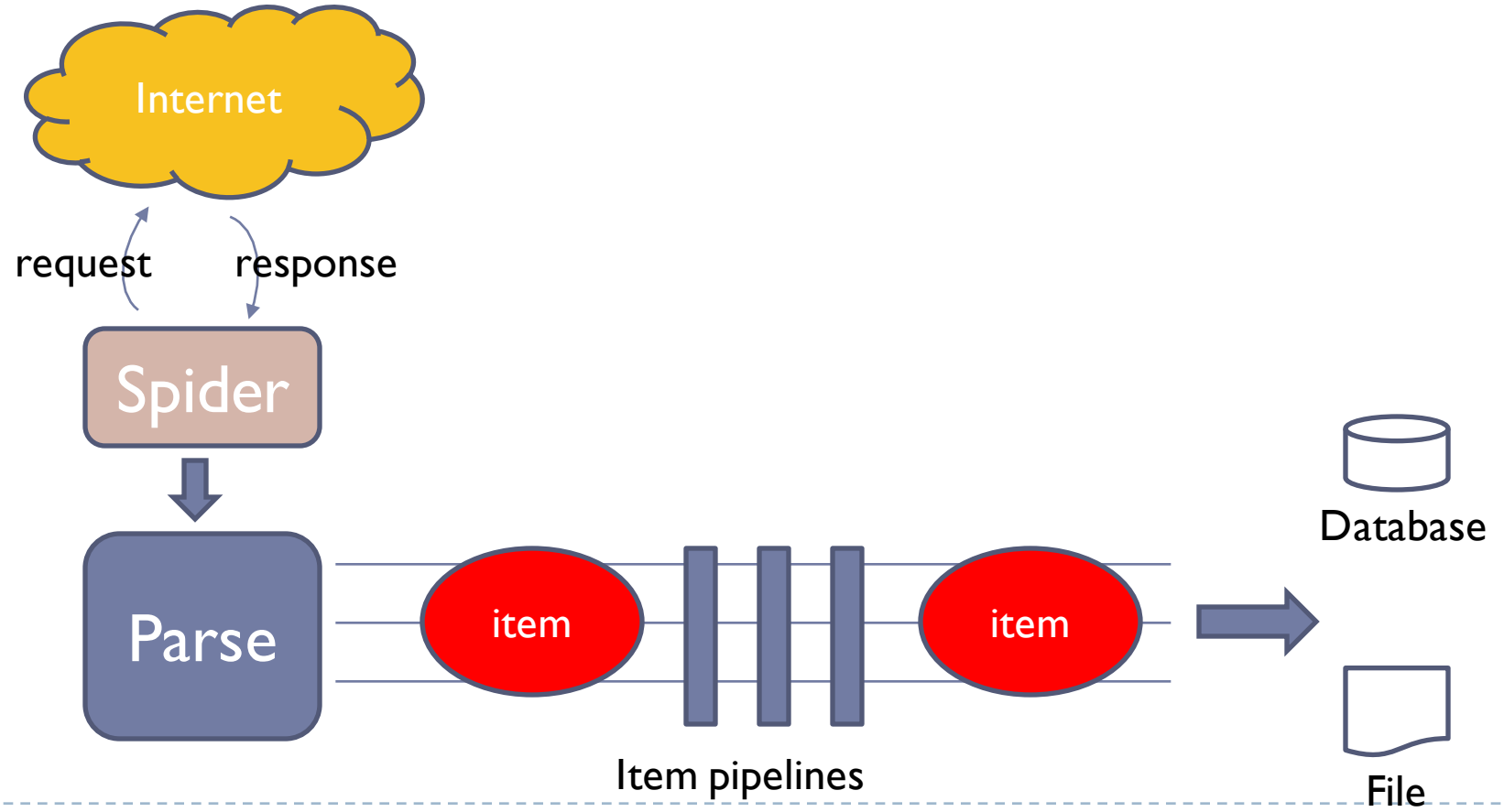


Huawei AppStore Data

C20679	QQ浏览器	QQ浏览器，邀你来狂欢!!
C10049053	华为商城	【华为商城客户端 我的掌上抢购神器】
C5683	微信	1.可以发语音、文字消息、表情、图片、视频30M流量可以收发上千条语音，省电省流量
C5373	支付宝	双12，支付宝口碑随机5折！
C34075	手机淘宝	打开淘宝领金币-人人有份
C10200687	华为视频	华为视频是华为手机上内置的系统视频播放器，旨在为用户提供一站式的视频体验服务。无需安装第三方视频软件
C121553	WPS Office	崭新界面设计全面开启，操作得到更大幅度的提高和优化
C57236	今日头条	★用户量过3亿的新闻阅读客户端
C66323	华为备份	【软件介绍】
C10132067	华为帐号	华为帐号是您用于访问所有华为服务的帐号。使用华为帐号登录后，您将可以使用这台设备的华为云服务、应用市
C31346	手机百度	【软件介绍】
C10055832	华为文件管理	【软件介绍】
C10067631	华为手机服务	“手机服务”为华为手机用户提供立体在线服务，集成智能问答、用户指南、服务网点、花粉论坛等必备服务，旨在
C3466	酷狗音乐	【核心功能】
C21976	UC浏览器	春节去哪儿？UC浏览器送你1000元旅游卡，让你畅玩无忧！12月12-13日下载UC浏览器，进入UC浏览器即可参与抽
C27162	华为应用市场	新版本特性：
C2002	爱奇艺视频	爱奇艺视频全网独播：奔跑吧兄弟3!
C10060708	华为支付	该应用为华为钱包支付服务插件，无桌面图标，如需进行花币充值、银行卡管理、查询账单及体验其他钱包功能，
C10207207	华为云服务	【软件介绍】
C9319	QQ	QQ手机版，致力于打造欢乐无限的沟通、娱乐与生活体验——乐在沟通15年，聊天欢乐8亿人！
C7166	微博	随时随地发现新鲜事！微博带你欣赏世界上每一个精彩瞬间，了解每一个幕后故事。分享你想表达的，让全世界看
C3382	百度地图	【基本功能介绍】
C10021449	音乐	华为音乐是华为手机上内置的系统音乐播放器，旨在为用户提供一站式的音乐体验服务。无需安装第三方音乐软件

Scrapy at a Glance

- an open source framework for crawling web sites
- extracting structured data



Three Mini Projects

- Crawl homepage Huawei Appstore
- Save crawled data in a file
- Follow URLs in homepage to get more data

QQ★★★★★

QQ手机版，致力于打造欢乐无限的沟通、娱乐与生活体验——乐在沟通15年，聊天欢乐8亿人！

【全新视觉，灵动呈现】

- 全新界面：化繁为简的架构，更加轻便...

发布时间： 2015-12-10　　支持固件:2.3以上

免费下载

下载:726307055次

appid

name

```
<div class="game-info  whole">
    <h4 class="title"><a href="http://appstore.huawei.com:80/app/C9319" title="QQ">QQ</a><span class='
    <div class="game-info-dtail part">
        <p class="content">QQ手机版，致力于打造欢乐无限的沟通、娱乐与生活体验—乐在沟通15年，聊天欢乐8亿人！
        <p class="date"><span>发布时间： 2015-12-10</span><span>支持固件:2.3以上</span></p>
    </div>
    <div class="app-btn"><a class="btn-blue down" onclick="zhytools.downloadApp('C9319','QQ' , 'list_2
dl/application/apk/51/51614a0c767e4258b87016cf2ec9ecc2/com.tencent.mobileqq.1512100947.apk?sign=portal

</div>
```

description

Crawl Appstore Homepage Steps

- Prerequisites
  - OS: Linux
  - Python2.7, pip
- Install scrapy: `http://doc.scrapy.org/en/latest/intro/install.html`
  - `pip install scrapy`
- Start a scrapy project
- Add crawling logic in a scrapy spider
- Run your first spider

Crawl Huawei AppstoreHomepage

- Create a scrapy project
  - `scrapy startproject appstore`

- create huawei_spider.py under spiders/

```
.
├── __init__.py
├── __init__.pyc
├── items.py
├── items.pyc
├── pipelines.py
├── settings.py
├── settings.pyc
└── spiders
    ├── huawei_spider.py
    ├── huawei_spider.pyc
    ├── __init__.py
    └── __init__.pyc
```

Define Extracted Data Schema

- Edit appstore/appstore/items.py, add the following:

```
import scrapy


class AppstoreItem(scrapy.Item):
    # define the fields for your item here like:
    title = scrapy.Field()
    url = scrapy.Field()
    appid = scrapy.Field()
    intro = scrapy.Field()
```

```
<div class="game-info  whole">
    <h4 class="title"><a href="http://appstore.huawei.com:80/app/C9319" title="QQ">QQ</a><span class='
    <div class="game-info-dtail part">
        <p class="content">QQ手机版，致力于打造欢乐无限的沟通、娱乐与生活体验—乐在沟通15年，聊天欢乐8亿人！
        <p class="date"><span>发布时间：  2015-12-10</span><span>支持固件:2.3以上</span></p>
    </div>
    <div class="app-btn"><a class="btn-blue down" onclick="zhytools.downloadApp('C9319','QQ' , 'list_2
dl/application/apk/51/51614a0c767e4258b87016cf2ec9ecc2/com.tencent.mobileqq.1512100947.apk?sign=portal

</div>
```

```python
import scrapy
import re
from scrapy.selector import Selector
from appstore.items import AppstoreItem

class HuaweiSpider(scrapy.Spider):
  name = "huawei"
  allowed_domains = ["huawei.com"]

  start_urls = [
    "http://appstore.huawei.com/more/all"
  ]

  def parse(self, response):
    page = Selector(response)

    divs = page.xpath('//div[@class="game-info  whole"]')

    for div in divs:
      item = AppstoreItem()
      item['title'] = div.xpath('.//h4[@class="title"]/a/text()'). \
          extract_first().encode('utf-8')
      item['url'] = div.xpath('.//h4[@class="title"]/a/@href').extract_first()
      appid = re.match(r'http://.*/(.*)', item['url']).group(1)
      item['appid'] = appid
      item['intro'] = div.xpath('.//p[@class="content"]/text()'). \
          extract_first().encode('utf-8')
      yield item
```

Time to Run Your First Spider

```
cd appstore
scrapy crawl huawei
```

In the commandline, you will see something look like:

```
{'appid': u'C31346',
 'intro': '\xe3\x80\x90\xe8\xbd\xaf\xe4\xbb\xb6\xe4\xbb\x8b\xe7\xbb\x8d\xe3\x80\x9
1 ',
 'title': '\xe6\x89\x8b\xe6\x9c\xba\xe7\x99\xbe\xe5\xba\xa6',
 'url': u'http://appstore.huawei.com:80/app/C31346'}
2015-12-23 22:20:28 [scrapy] DEBUG: Scraped from <200 http://appstore.huawei.com/m
ore/all>
```

Three Mini Projects

- Crawl homepage Huawei Appstore
- Save crawled data in a file
- Follow URLs in homepage to get more data

Enable Data Pipeline in Scrapy

- Edit appstore/appstore/settings.py

```
ITEM_PIPELINES = {
   'appstore.pipelines.AppstorePipeline': 300,
}
DOWNLOAD_DELAY=5
```

- Edit appstore/appstore/pipelines.py

```
class AppstorePipeline(object):
  def __init__(self):
    self.file = open('appstore.dat', 'wb')


  def process_item(self, item, spider):
    val = "{0}\t{1}\t{2}\n".format(item['appid'], item['title'], item['intro'])
    self.file.write(val)
    return item
```

C10207207　　华为云服务　　【软件介绍】

Time to Run Your Spider again

```
cd appstore
scrapy crawl huawei
cat appstore.dat
```

Huawei AppStore Data

C20679	QQ浏览器	QQ浏览器，邀你来狂欢!!
C10049053	华为商城	【华为商城客户端 我的掌上抢购神器】
C5683	微信	1.可以发语音、文字消息、表情、图片、视频30M流量可以收发上千条语音，省电省流量
C5373	支付宝	双12，支付宝口碑随机5折！
C34075	手机淘宝	打开淘宝领金币-人人有份
C10200687	华为视频	华为视频是华为手机上内置的系统视频播放器，旨在为用户提供一站式的视频体验服务。无需安装第三方视频软件
C121553	WPS Office	崭新界面设计全面开启，操作得到更大幅度的提高和优化
C57236	今日头条	★用户量过3亿的新闻阅读客户端
C66323	华为备份	【软件介绍】
C10132067	华为帐号	华为帐号是您用于访问所有华为服务的帐号。使用华为帐号登录后，您将可以使用这台设备的华为云服务、应用市
C31346	手机百度	【软件介绍】
C10055832	华为文件管理	【软件介绍】
C10067631	华为手机服务	“手机服务”为华为手机用户提供立体在线服务，集成智能问答、用户指南、服务网点、花粉论坛等必备服务，旨在
C3466	酷狗音乐	【核心功能】
C21976	UC浏览器	春节去哪儿？UC浏览器送你1000元旅游卡，让你畅玩无忧！12月12-13日下载UC浏览器，进入UC浏览器即可参与抽
C27162	华为应用市场	新版本特性：
C2002	爱奇艺视频	爱奇艺视频全网独播：奔跑吧兄弟3!
C10060708	华为支付	该应用为华为钱包支付服务插件，无桌面图标，如需进行花币充值、银行卡管理、查询账单及体验其他钱包功能，
C10207207	华为云服务	【软件介绍】
C9319	QQ	QQ手机版，致力于打造欢乐无限的沟通、娱乐与生活体验——乐在沟通15年，聊天欢乐8亿人！
C7166	微博	随时随地发现新鲜事！微博带你欣赏世界上每一个精彩瞬间，了解每一个幕后故事。分享你想表达的，让全世界看
C3382	百度地图	【基本功能介绍】
C10021449	音乐	华为音乐是华为手机上内置的系统音乐播放器，旨在为用户提供一站式的音乐体验服务。无需安装第三方音乐软件

Three Mini Projects

- Crawl homepage Huawei Appstore
- Save crawled data in a file
- Follow URLs in homepage to get more data

Follow URLs in Homepage

Define Extracted Data Schema

- Add a new field to the schema

```
class AppstoreItem(scrapy.Item):
    # define the fields for your item here like:
    title = scrapy.Field()
    url = scrapy.Field()
    appid = scrapy.Field()
    intro = scrapy.Field()
    recommended = scrapy.Field()
```

Modify huawei_spider.py

```
  def parse(self, response):
    page = Selector(response)

    hrefs = page.xpath('//h4[@class="title"]/a/@href')

    for href in hrefs:
      url = href.extract()
      yield scrapy.Request(url, callback=self.parse_item)


  def parse_item(self, response):
    page = Selector(response)
    item = AppstoreItem()

    item['title'] = page.xpath('//ul[@class="app-info-ul nofloat"]/li/p/span[@cl
ass="title"]/text()').extract_first().encode('utf-8')
    item['url'] = response.url
    item['appid'] = re.match(r'http://.*/(.*)', item['url']).group(1)
    item['intro'] = page.xpath('//meta[@name="description"]/@content').extract_f
irst().encode('utf-8')

    divs = page.xpath('//div[@class="open-info"]')
    recomm = ""
    for div in divs:
      url = div.xpath('./p[@class="name"]/a/@href').extract_first()
      recommended_appid = re.match(r'http://.*/(.*)', url).group(1)
      name = div.xpath('./p[@class="name"]/a/text()').extract_first().encode('ut
f-8')
      recomm += "{0}:{1},".format(recommended_appid, name)
    item['recommended'] = recomm
    yield item
```

Extract urls from homepage

Extract title, url, intro, recommended apps from the app page

Time to Run Your Spider again

```
cd appstore
scrapy crawl huawei
cat appstore.dat
```

Huawei Recommended Apps Data


```
C2002   http://appstore.huawei.com:80/app/C2002 爱奇艺视频   C183901:WiFi信号增强…,C20252:京东,C10047107:滴滴出行,C43719:迅雷,C39289
C34075  http://appstore.huawei.com:80/app/C34075    手机淘宝 C20252:京东,C5373:支付宝,C57804:天猫,C40224:相片组合 Phot…,C10193357:微
C3466   ht                                         466 酷狗音乐 C34075:手机淘宝,C5683:微信,C183901:WiFi信号增强…,C9319:QQ,C10047107:滴滴出行
C5373   ht                                         373 支付宝   C20252:京东,C34075:手机淘宝,C3382:百度地图,C10047107:滴滴出行,C174391:百度云
C27162  ht                                         7162    华为应用市场 C20252:京东,C183901:WiFi信号增强…,C10047107:滴滴出行,C54626:铃声多多
C3382   ht                                         382 百度地图 C20252:京东,C34075:手机淘宝,C183440:签证侠香香,C10179513:单机斗地主,C39289:
C10207207                                          /C10207207 华为云服务   C20252:京东,C183901:WiFi信号增强…,C23707:违章查询,C10067631:华为
C10067631                                          /C10067631 华为手机服务 C9319:QQ,C5683:微信,C183901:WiFi信号增强…,C2217:我查查,C3466:酷
C66323  ht                                         5323    华为备份 C5683:微信,C9319:QQ,C54626:铃声多多,C183901:WiFi信号增强…,C174391:百度云
C5683   ht                                         583 微信    C183901:WiFi信号增强…,C20252:京东,C9319:QQ,C3466:酷狗音乐,C34075:手机淘宝,C
C10021449                                          /C10021449 音乐    C3466:酷狗音乐,C10220136:QQ音乐,C10191116:图片合成器,C5683:微信,C17
C9319   ht                                         319 QQ  C20252:京东,C5683:微信,C34075:手机淘宝,C5373:支付宝,C2217:我查查,C3466:酷狗音乐,
C10132067                                          /C10132067 华为帐号 C10057661:Google Play…,C21976:UC浏览器,C10168550:亲情关怀,C5373:支
C31346  ht                                         1346    手机百度 C174391:百度云,C43397:百度浏览器,C10047107:滴滴出行,C183901:WiFi信号增强
C57236  ht                                         7236    今日头条 C20252:京东,C43442:新浪新闻,C2217:我查查,C19168:凤凰新闻,C183901:WiFi信
C21976  http://appstore.huawei.com:80/app/C21976    UC浏览器 C20252:京东,C183901:WiFi信号增强…,C174391:百度云,C10114178:流量宝,C1005
```

Summary

- Three mini projects
  - Crawl homepage Huawei Appstore
  - Save crawled data in a file
  - Follow URLs in homepage to get more data

Homework

- Get the URL of thumbnail of each app

Tiger AppStore

Scrapy – Crawl the Web

Advanced Techniques

Outlines

- Render Javascript

- Use Proxy

Render Javascript

- splash - a Javascript rendering service
  - Installation: https://splash.readthedocs.org/en/stable/install.html
    - Recommended: Linux + Docker
- scrapyjs
  - Installation: https://github.com/scrapinghub/scrapy-splash
- Start splash service
  - `sudo docker run -p 8050:8050 scrapinghub/splash`

```
Splash now available at 0.0.0.0 at ports 8050
```

Render Javascript Continue

- In settings.py

```
DOWNLOADER_MIDDLEWARES = {
  'scrapyjs.SplashMiddleware': 725,
}

SPLASH_URL = 'http://localhost:8050/'
DUPEFILTER_CLASS = 'scrapyjs.SplashAwareDupeFilter'
HTTPCACHE_STORAGE = 'scrapyjs.SplashAwareFSCacheStorage'
```

Refer DOWNLOADER_MIDDLEWARES_BASE for the right middleware order

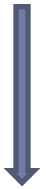
Render Javascript Continue

- In huawei_spider.py

```
def parse(self, response):
  page = Selector(response)

  hrefs = page.xpath('//h4[@class="title"]/a/@href')

  for href in hrefs:
    url = href.extract()
    yield scrapy.Request(url, callback=self.parse_item)
```



```
yield scrapy.Request(url, self.parse, meta={
  'splash': {
    'endpoint': 'render.html',
    'args': {'wait': 0.5}
  }
})
```

Use Proxy

- Create random proxy for user-agent
  - Default user agent: `Scrapy/VERSION (+http://scrapy.org)`
  - Ref: http://stackoverflow.com/questions/23152739/how-to-make-scrapy-show-user-agent-per-download-request-in-log
- Create random proxy for IP address
  - Ref: https://github.com/aivarsk/scrapy-proxies

Use Proxy Continue

- In settings.py

```
DOWNLOADER_MIDDLEWARES = {
  'scrapy.downloadermiddlewares.useragent.UserAgentMiddleware' : None,
  'appstore.random_useragent.RandomUserAgentMiddleware' :400,
  'appstore.random_proxy.RandomProxyMiddleware': 100,
}
```

Set Random User-agent

```
class RandomUserAgentMiddleware(UserAgentMiddleware):
  def __init__(self, settings, user_agent='Scrapy'):
    super(RandomUserAgentMiddleware, self).__init__()
    self.user_agent = user_agent


  def process_request(self, request, spider):
    ua = random.choice(self.user_agent_list)
    # ua = "Scrapy/VERSION (+http://scrapy.org)"
    print "********Current UserAgent:%s************" %ua
    request.headers.setdefault('User-Agent', ua)
    '''
    the default user_agent_list composes chrome,IE,firefox,Mozilla,
    opera,netscape
    for more user agent strings,you can find it in
    http://www.useragentstring.com/pages/useragentstring.php
    '''
  user_agent_list = [
    "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.
0.2228.0 Safari/537.36",
    "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:43.0) Gecko/20100101 Firefox/43.0",
  ]
```

Homework Answer

```
item['thumbnailurl']  = page.xpath('//ul[@class="app-info-u1
nofloat"]/li[@class="img"]/img[@class="app-ico"]/@lazyload').extract_first()
```

Tiger AppStore

Crawl More Apps

路广成

Crawler Workflow

- Traverse like a tree
  - Actually a graph, but we stop at each cycle
  - Removing duplicate request URL is enabled by default
- Start URLs -> root of traversal
- Following link -> start point of branches

Start URL

- Lists on the home page
- Get all apps in each list
  - Follow Qinyi's instruction
    - Splash, crapyjs
    - You will get everything by doing this automatic method
  - Structured URL convention
    - Nothing fancy by easiest method

Lists on the home page

Lists on the Home Page

总排行-安卓软件排行榜|安

appstore.huawei.com/more/all

Study | 九章 | 羽毛球 | 综艺 | 动漫 | TV Series | 跳蚤 | ML | BigData | TOP 50 BEST COVER...

华为官网 华为荣耀 华为商城 **软件应用** 花粉俱乐部 登录 / 注册 选择区域 / 语言

华为应用市场 appstore.huawei.com

搜索 开心消消乐 百度 视频

首页 游戏 软件 专题 品牌专区 华为软件专区

排序 精品推荐 总排行 应用排行 游戏排行 上升最快 最新上架

QQ

QQ手机版，致力于打造欢乐无限的沟通、娱乐与生活体验——乐在沟通15年，聊天欢乐8亿人！

【全新视觉，灵动呈现】

- 全新界面：化繁为简的架构，更加轻便…

发布时间： 2015-12-28 支持固件:2.3以上

免费下载

下载:772804941次

微信

1.可以发语音、文字消息、表情、图片、视频30M流量可以收发上千条语音，省电省流量

2.朋友圈，跟朋友们分享生活点滴

3.摇一摇、查看附近的人，世界不再…

发布时间： 2016-01-04 支持固件:2.3以上

免费下载

下载:799789437次

热门专题推荐 更多>

1 要啥自拍神器

2 妈妈手机必装

3 专治无聊综合征

4 自己做大厨

5 家有萌宠

6 远离"白领病"

7 最爱二次元

8 健人就是较轻

9 12月最受欢迎游戏

Get all apps in each list

Get All Apps in Each List

What is the URL for page 2

总排行-安卓软件排行榜|安

appstore.huawei.com/more/all

First user

Study 九章 跳蚤 ML BigData TOP 50 BEST COVER...

华为官网 华为荣耀 华为商城 软件应用 花粉俱乐部 登录 / 注册 选择区域 / 语言

HUAWEI 华为应用市场 appstore.huawei.com 放心下，畅快玩

搜索 开心消消乐 百度 视频

首页 游戏 软件 专题 品牌专区 华为软件专区

排序 精品推荐 总排行 应用排行 游戏排行 上升最快 最新上架

QQ★★★★★

QQ手机版，致力于打造欢乐无限的沟通、娱乐与生活体验——乐在沟通15年，聊天欢乐8亿人！

【全新视觉，灵动呈现】

- 全新界面：化繁为简的架构，更加轻便…

发布时间： 2015-12-28 支持固件:2.3以上

免费下载

下载:772804941次

微信★★★★★

1.可以发语音、文字消息、表情、图片、视频30M流量可以收发上千条语音，省电省流量

2.朋友圈，跟朋友们分享生活点滴

3.摇一摇、查看附近的人，世界不再…

发布时间： 2016-01-04 支持固件:2.3以上

免费下载

下载:799789437次

热门专题推荐 更多>

1 要啥自拍神器

2 妈妈手机必装

3 专治无聊综合征

4 自己做大厨

5 家有萌宠

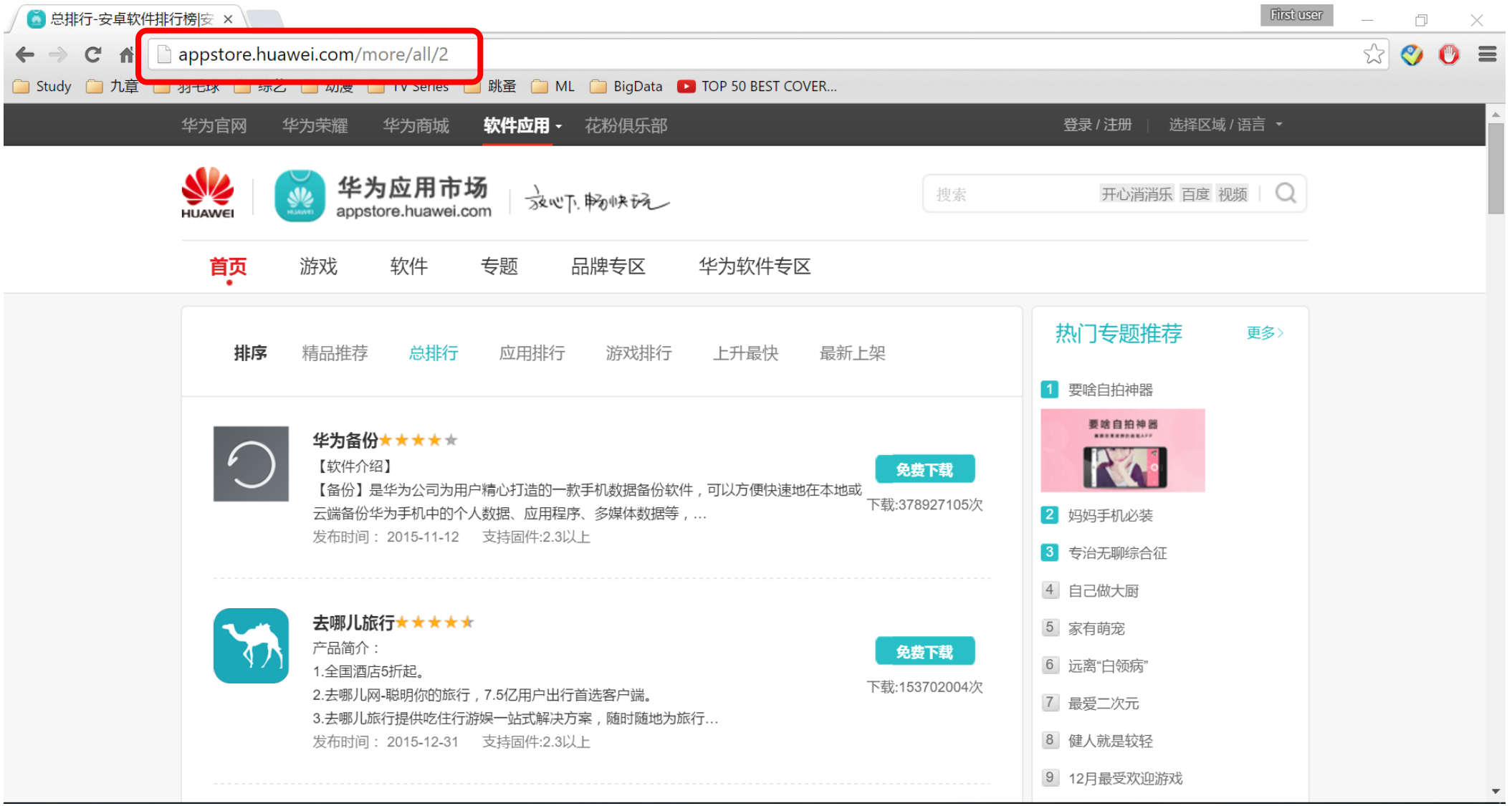
6 远离"白领病"

7 最爱二次元

8 健人就是较轻

9 12月最受欢迎游戏

Get All Apps in Each List



Structured URL

```
start_urls = [
    "http://appstore.huawei.com/more/all/1",
    "http://appstore.huawei.com/more/recommend/1",
    "http://appstore.huawei.com/more/soft/1",
    "http://appstore.huawei.com/more/game/1",
    "http://appstore.huawei.com/more/newPo/1",
    "http://appstore.huawei.com/more/newUp/1",
```

```
# figure out the next page to crawl
def find_next_page(self, url):
    try:
        page_num_str = url.split('/')[-1]
        page_num = int(page_num_str) + 1
        # limit page count for testing
        # if page_num > 1: # crawl only specified number of pages
        #     return "http://google.com"
        url = url[:-len(page_num_str)] + str(page_num)
        return url
    except ValueError:
        print "### page cannot be handled: ",
        print url
        return "http://google.com"
```

Digging

- How to create more list?
  - We have limited list now
  - Means we have limited reachability

The answer is searching

Lists of Search Results

Including Search Lists to Start URL

```
start_urls = [
    "http://appstore.huawei.com/more/all/1",
    "http://appstore.huawei.com/more/recommend/1",
    "http://appstore.huawei.com/more/soft/1",
    "http://appstore.huawei.com/more/game/1",
    "http://appstore.huawei.com/more/newPo/1",
    "http://appstore.huawei.com/more/newUp/1",
    # abnormal searches
    "http://appstore.huawei.com/search/0/1", # search "0"
    "http://appstore.huawei.com/search/1/1", # search "1"
    "http://appstore.huawei.com/search/2/1", # search "2"
    "http://appstore.huawei.com/search/3/1", # search "3"
    "http://appstore.huawei.com/search/4/1", # search "4"
    "http://appstore.huawei.com/search/5/1", # search "5"
    "http://appstore.huawei.com/search/6/1", # search "6"
    "http://appstore.huawei.com/search/7/1", # search "7"
    "http://appstore.huawei.com/search/8/1", # search "8"
    "http://appstore.huawei.com/search/9/1", # search "9"
    "http://appstore.huawei.com/search/%25E5%25BA%2594%25E7%2594%25A8/1", # search "app"
    "http://appstore.huawei.com/search/%25E6%25B8%25B8%25E6%2588%258F/1", # search "game"
    "http://appstore.huawei.com/search/%25E5%2585%25A8/1", # search "all"
    "http://appstore.huawei.com/search/%25E8%25BD%25AF%25E4%25BB%25B6/1" # search "software"
]
```

Digging Deeper

- How to create more search?
  - The more searches the better chance we get more apps
  - What we need here is really diversity
- How to make the search return more?
  - Generalized search keyword
  - Try to understand backend DB query formation
    - General search keyword could give shorter list
    - Search result for “用” and “应用” shown in next page

Searching Examples