**分布式爬虫系统总结**



1. **关键技术**

1.1反爬虫机制

1. 伪造请求头
2. 设置代理

1.2通讯机制

1. 浏览器和后台用websocket通讯，后台使用tornado接收数据
2. 服务器之间用socket通讯（含阻塞型和非阻塞型），使用了socketserver，可以每秒钟处理上千次请求。

1.3爬虫模块

1. 用phantomjs模拟浏览器采集动态网页的内容。
2. 用requests采集静态网页内容
3. 用面向对象方法封装一系列操作

1.4 UI

1. Echarts动态实时刷新图表
2. 基于websocket接收数据
3. 使用js脚本动态刷新数据

1.5 分布式技术

1. redis分布式数据库
2. 分布式去重
3. 监控爬虫子节点的内存占用和cpu使用率
4. 主--从任务分配
5. **创新点**

2.1.自己研发的后台展示系统。

2.2.自己研发爬虫模块。

2.3.自制爬虫框架。

2.4.自制通讯模块。

1. **难点以及解决方法**

3.1.前端数据动态实时刷新图表。

解决方案：

3.2.分布式数据库的汇总。

解决方案：写shell脚本或者windos批处理脚本实现远程的访问控制，将分节点的数据统一发送到主机服务器。

3.3.免费代理数量有限。

解决方案：通过shadowsocks远程连接国外的vps，通过Tor洋葱网络获取更多可用的免费代理资源。

3.4.多进程端口冲突。

解决方案：使用docker容器进行端口映射，避免冲突。但是开发平台使用windows不支持docker，切换平台。

3.5.phantomjs没有提供代理切换功能。

解决方案：通过执行js代码实现切换，但是无法切换UA。

3.6.中文编码在传输中会被转码。

解决方案：在反序列化的过程中使用decode/encode方法进行编码解码的转换。

3.7.在前端和后端交互过程中对json进行反序列化的过程中容易出现任意代码执行漏洞。

解决方案：不使用eval函数而使用json.parse()函数。