**中文摘要:** 随着以互联网为代表的大数据信息时代的到来信息数量急速膨胀，网上信息用海量形容，使得专门负责信息采集的网络爬虫技术面临了一个巨大的挑战。网络爬虫以其灵活的可定制性和信息采集速度和规模，满足了人日益益增长的对信息获取的手段，用以满足对大数据的数据挖掘与数据分析提供数据获取、搜索引擎的需求。

针对上述问题，本文以Python2.7和Scrapy环境为基础，基于Scrapy-Redis分布式框架的网络爬虫，并以“新浪微博”Web为爬取的对象，在学习并分析当前爬虫技术的原理、核心模块以及运行流程的基础上，探索性地实现一个基于Scrap-Redis框架,多线程，高并发，强鲁棒性的网络爬虫，完成数据抓取等目标，然后对爬到的数据进行简单的数据分析。

首先，本文简明给出了爬虫技术的原理和发展现状，介绍爬虫工程中一些关键技术，并着重介绍了在本研究中有深刻影响的Cookie池和user-agent欺骗来突破站点限制、信息过滤、搜索策略。

其次，通过使用基于Python语言开发的Scrapy开源爬虫框架来进行爬虫开发，指出了Redis的内存数据库做去重、任务调度、提高爬取速度、支持“断点继爬”，同时指出了MongoDB为代表的NoSQL数据库在元数据存储中的巨大作用。详细介绍了Scrapy-Redis开发爬虫的流程和实现细节。

再次，讨论了对于爬虫设计领域的关键问题，本文实现的自定义爬虫的解决方法。与对抗微博反爬技术，破解验证码，URL去重防止环路的出现，和多线程并发的问题，则采用并分析Scrapy-Redis自带的解决方案。

最后对爬虫爬取到的数据做简单的数据分析得出一系列小结论。

**关键词: 新浪微博, Scrapy-Redis ,Python, Web, 爬虫, 数据分析**

**Abstract**

With the rapid expansion of the number of incoming information in the era of big data information represented by the Internet, online information is described in terms of mass, making the web crawler technology responsible for information collection face a great challenge. With its flexible customizability and information collection speed and scale, the web crawler meets people's increasingly growing means of information acquisition to meet the needs of data acquisition and search engine for data mining and data analysis of big data.

In view of the above problems, based on the Python2.7 and Scrapy environment, this article is based on Scrapy-Redis distributed framework web crawler, and uses "Sina Weibo" Web as crawling object to learn and analyze the current crawling technology principles. Based on the core modules and the running process, an exploratory implementation of a Scrap-Redis framework, multithreading, high concurrency, and robust web crawler is performed to complete the data capture and other objectives, and then the crawled data is simply data analysis.

    First of all, this paper gives a brief description of the principle and development status of crawler technology, introduces some key technologies in crawler engineering, and focuses on the deep impact of the Cookie pool and user-agent deception in this study to break site restrictions, information filtering, Search strategy.

    Secondly, by using the Scrapy open source crawler framework developed based on the Python language for crawler development, Redis's in-memory database is deduplicated, tasked, crawled, and crawled, and supported by "breakpoints, climbs," and MongoDB is indicated. The NoSQL database has a huge role in metadata storage. Details the process and implementation details of the Scrapy-Redis development crawler.

     Again, the key issues in the field of crawler design are discussed, and the solutions for custom crawlers implemented in this article are discussed. With anti-climb technology against microblogging, crack verification code, URL deduplication to prevent the emergence of loops, and multi-threaded concurrent issues, then use and analyze Scrapy-Redis's own solution.

Finally, a simple data analysis of the data crawled by the crawler yields a series of small conclusions.

**Keywords：Sina Weibo, Scrapy-Redis, Python, Web, Crawler, Data Analysis**