

**基于Scrapy的模板化爬虫程序管理平台**

**性能测试报告**

v1.1

沈一聪 SY1906510

2020年5月21日

**版本变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 修改人 | 修改章节 | 备注 |
| 1.0 | 2020/5/19 | 沈一聪 | / | 初版 |
| 1.1 | 2020/5/21 | 沈一聪 | 全文 | 添加变更记录；修改表一样式和错误引用；统一Easyspider表述；调整图2缩进；修改表表头3.4字体 |

1. 分布式爬虫的性能优化测试

如表一为Easyspider的性能测试方案，由于时间的限制，目前针对豆瓣电影爬虫进行了相同工作量下任务完成时间的对比测试。

表1 Easyspider爬取性能测试方案

|  |  |  |
| --- | --- | --- |
| 性能场景 | 执行策略 | 结果分析方法 |
| 1. 采用Easyspider完成爬虫任务 2. 采用一般的爬虫脚本完成爬虫任务 | 1. 分别采用两种不同的性能场景下，完成豆瓣电影、豆瓣电影以及豆瓣音乐相关的爬虫任务；任务负载从低到高，覆盖所有的使用场景 2. 分别记录单位时间内，两种不同的性能场景下，持续爬取数据的数据量大小 | 得到不同工作负载特点下，相同工作量的任务消耗时间或是相同时间下完成的工作量，并通过绘制图表的方式进行对比 |

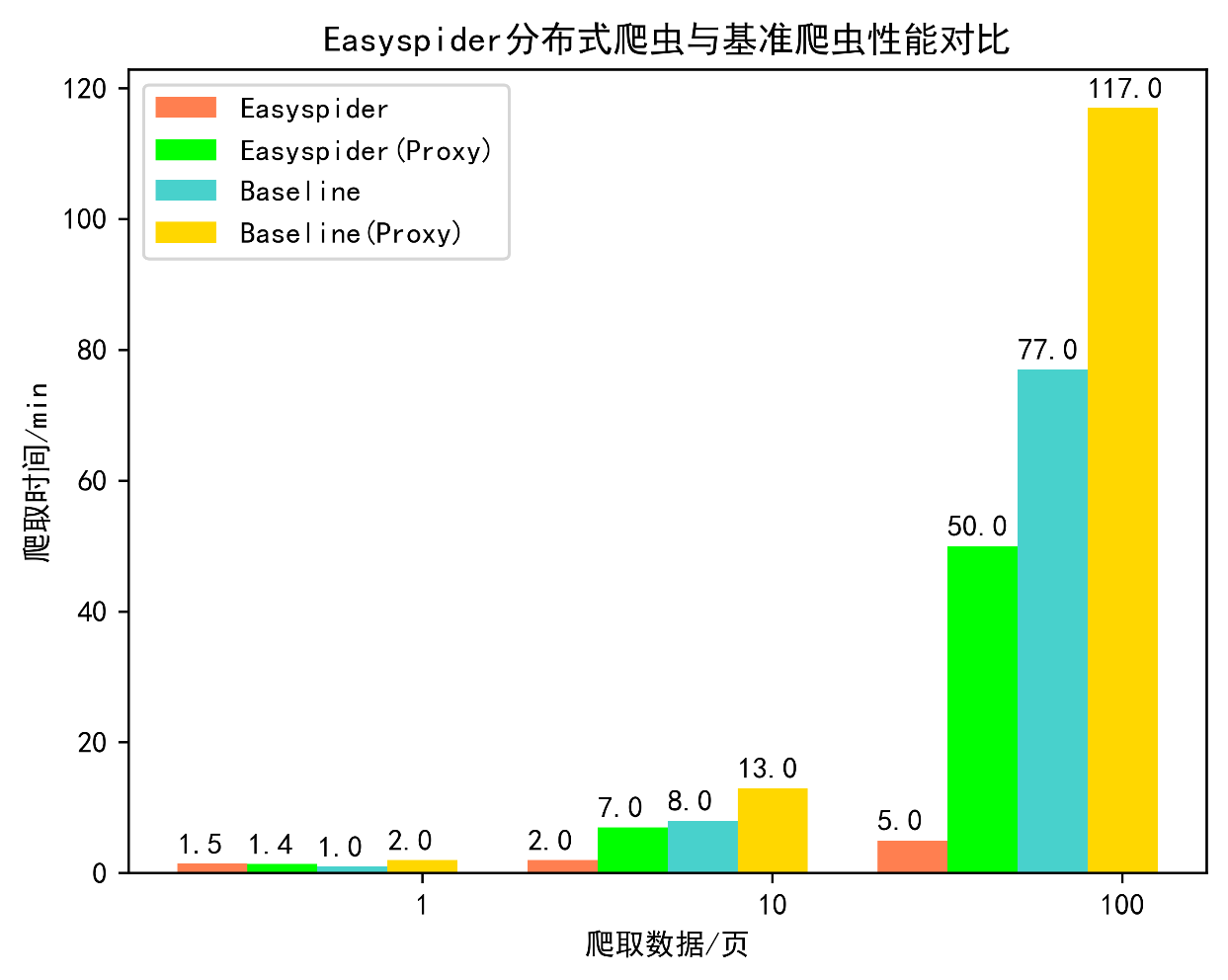


图1 Easyspider性能测试结果

图1为Easyspider的性能测试结果。测试的性能场景包括Easyspider（4服务器节点）、Easyspider+Proxy、Baseline（本地的Scrapy项目）以及Baseline+Proxy。从测试结果可以看出，在工作负载越重的情况下，Easyspider实现的分布式爬虫带来的性能优势就越明显。

1. Easyspider平稳性测试

如表2所示，采用长时间连续爬取来测试Easyspider的稳定性。

表2 Easyspider平稳性测试方案

|  |  |  |
| --- | --- | --- |
| 性能场景 | 执行策略 | 结果分析方法 |
| 采用Easyspider完成爬虫任务 | 采用Easyspider在一段较长时间内持续爬取数据，记录爬取数据量与时间的关系 | 通过分析吞吐量与时间的图线关系，分析系统平稳特性 |

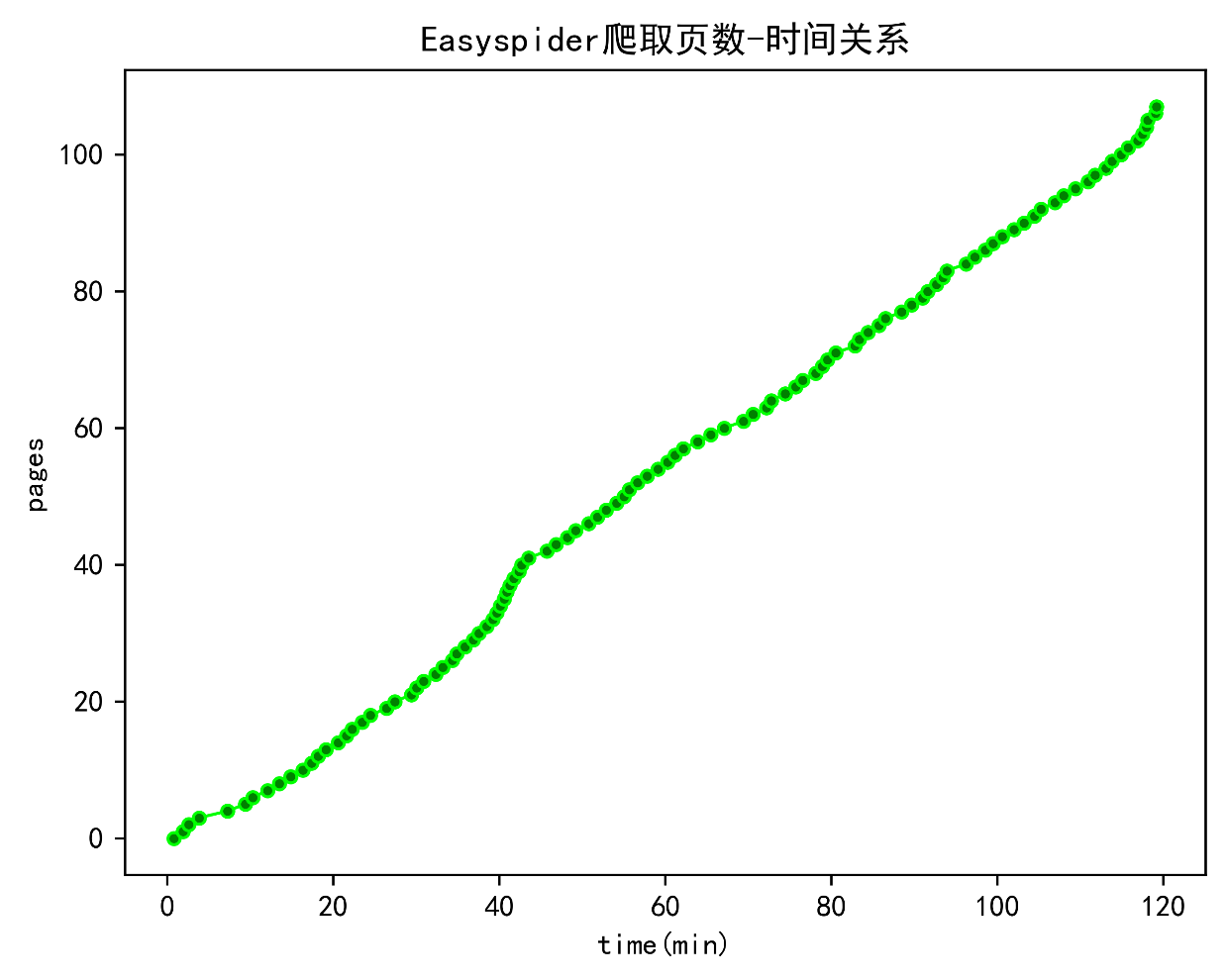


图 2 Easyspider的稳定性测试结果

图2展示了Easyspider在单服务器情况下（便于统计结果），连续爬取120min中数据的的运行情况。从图中可以看出，Easyspider的爬取速度基本稳定不变，爬取页数-时间呈现线性关系。

1. 负载/压力测试

如表3所示，对Easyspider网站进行负载/压力测试。

表3 Easyspider负载/压力测试

|  |  |  |
| --- | --- | --- |
| 性能场景 | 执行策略（并发数、时长） | 备注 |
| 发送请求，等待响应 | 10用户并发，发送请求，等待响应，重复操作10分钟，退出。 | 得到不同并发数下系统的性能指标  对系统的容量做出估计  列出测试的数据指标项有哪些，值在什么区间内 |
| 50用户并发，，发送请求，等待响应，重复操作10分钟，退出。 |
| 100用户并发，发送请求，等待响应，重复操作15分钟 |

表4 Easyspider负载/压力测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| 性能场景 | bytes/connection | fetches/sec | msecs/connect |
| 10用户并发，重复操作10分钟 | 6287 | 20.0333 | 42.8058 mean, 1254.93 max, 31.37 min |
| 50用户并发，重复操作10分钟 | 6273.17 | 20.03 | 46.4516 mean, 3041.28 max, 31.272 min |
| 100用户并发，重复操作15分钟 | 6226.23 | 20.1322 | 56.7554 mean, 3042.01 max, 31.481 min |

测试结果如表4所示，可以看出在长时间的负载以及压力测试下，Easyspider运行稳定，没有出现大规模连接超时的情况。