УДК 004.65:519.688

ПРОГРАММЫ-КРАУЛЕРЫ ДЛЯ СБОРА ДАННЫХ О ПРЕДСТАВИТЕЛЬСКИХ САЙТАХ ЗАДАННОЙ ПРЕДМЕТНОЙ ОБЛАСТИ – АНАЛИТИЧЕСКИЙ ОБЗОР

[1]Печников А.А., [2]Сотенко Е.М.

[1]ФГБУН «Институт прикладных математических исследований Карельского научного центра Российской академии наук», Петрозаводск, e-mail: pechnikov@krc.karelia.ru;
[2]ФГБОУ ВО «Санкт-Петербургский государственный университет», Санкт-Петербург, e-mail: katherinmail@gmail.com

Анализ данных, представленных в Вебе, – распространенная на сегодняшний день исследовательская задача. Для её решения необходимы программы-инструменты, позволяющие собирать данные из Веба. Для обозначения таких программ часто используется термин «краулер». Сегодня существует широкий спектр известных краулеров, и в ряде случаев это позволяет не писать с нуля новые, а использовать существующие программы. Если это краулеры с открытым кодом, то их можно дорабатывать под цели конкретных задач, формирующиеся и видоизменяющиеся в процессе исследования. В данной работе в качестве альтернативы рассмотрены несколько краулеров, из которых требуется сделать выбор наиболее подходящего для решения задачи сбора данных с некоторого заданного ограниченного множества представительских сайтов (таких, например, как сайты гостиниц). Приводятся основные требования, предъявляемые к краулерам, и их классификация по основным типам. Дается аналитический обзор наиболее популярных краулеров, для которых в качестве одного из главных критериев отбора является наличие открытого исходного кода. В результате проведенного анализа проведен отбор трех наиболее перспективных краулеров.

Ключевые слова: Интернет, Веб, сбор данных о Вебе, краулер, открытое программное обеспечение

CRAWLERS FOR COLLECTIG THE DATA FROM OFFICIAL SITES OF GIVEN SUBJECT AREA – ANALITICAL REVIEW

[1]Pechnikov A.A., [2]Sotenko E.M.

[1]Institute of Applied Mathematical Research of the Karelian Research Centre of the Russian Academy of Sciences, Petrozavodsk, e-mail: pechnikov@krc.karelia.ru;
[2]St. Petersburg State University, Saint-Petersburg, e-mail: katherinmail@gmail.com

The analysis of data, presented in Web, widely spread nowadays research task. In order to solve it usually used programs, gathering data from the Web, which are called «crawlers». There are a lot of already built crawlers, that allows not to write your own, but use one of existing instead. Some crawlers are open source that allows to modify them in case they don't provide required functionality. This paper provides the overview of several crawlers in order to solve the task of gathering the data from set of official sites (for example, official sites of hotels). Also were provided main requirements, which crawlers must meet, crawler's classification and analysis of their functionality. Is the crawler open source – was one of the main criteria of crawler's selection. The result of performed analysis was the selection of three most perspective crawlers.

Keywords: Internet, Web, web data collection, crawler, open source

Для получения больших объемов информации, необходимых для апробации и изучения моделей в рамках вебометрического анализа, можно применить несколько известных подходов, таких как использование возможностей расширенного поиска поисковых систем, использование открытых баз данных, созданных другими исследователями, покупка вебометрических данных у компаний, специализирующихся на таких видах деятельности, разработка собственных инструментов.

Наконец, можно использовать специализированные программы, предназначенные для сбора данных о Вебе, которые могут быть свободными или коммерческими.

Основные преимущества и недостатки каждого из подходов были проанализированы еще 15 лет назад в работе [5], и их рассмотрение выходит за рамки данной статьи. Мы исходим из того, что открытые специализированные программы для сбора данных о Вебе сегодня позволяют не писать с нуля новые, а использовать существующие [19] и дорабатывать их под потребности конкретных задач.

Термин «краулер» (англ. crawler) в общем случае обозначает программу, реализующую процесс перемещения по страницам и/или документам Веба с целью сбора определенной информации, статистики или сохранения ресурсов сайта. Общие принципы работы краулеров изложены в [17]. В самом общем виде работу краулера по сканированию сайта можно описать следующим образом: сканирование сайта начи-

нается с начальной страницы и затем программа использует ссылки, размещенные на ней, для перехода на другие страницы. Каждая страница сайта анализируется на наличие требуемой информации, которая копируется в соответствующее хранилище в случае обнаружения. Процесс повторяется до тех пор, пока будет проанализировано требуемое число страниц или достигнута некая цель.

В данной статье дается аналитический обзор наиболее популярных краулеров, для которых одним из главных критериев включения в данный обзор является наличие открытого исходного кода и нацеленность на представительские сайты организаций.

Класс содержательных задач и связанные с ним ограничения

В вебометрических исследованиях довольно часто рассматриваются задачи исследования связей между представительскими сайтами организаций, занимающихся одинаковыми видами деятельности и расположенных в одном географическом регионе. Понятие региона может рассматриваться достаточно широко – от города до континента.

Одним из примеров такой задачи может служить исследование сайтов гостиниц, расположенных в крупном городе. Данные, собранные с таких сайтов, должны содержать информацию о том, каким образом отели связаны между собой и сторонними сервисами (в частности, посредством гиперссылок), какой набор услуг они предоставляют помимо проживания, какими средствами они продвигают себя в Вебе.

Для того чтобы обеспечить сбор указанной информации, необходимо, чтобы искомый краулер поддерживал работу со следующими данными:

1) HTML, JavaScript, так как большинство сайтов разработано с использованием этих технологий;

2) Plain text, PDF и другие форматы представления текстовых данных;

3) URLs, с возможностью построения на их основе графа веб-ресурсов.

Краулер должен быть производительным, поскольку согласно первичной оценке объема данных, которые предстоит собрать и обработать, нужно извлечь данные более с чем 3000 представительских сайтов, при условии того, что каждый из них в среднем ссылается примерно на 11000 страниц, то есть обработать около 3300000 страниц. При средней скорости обработки 2 страницы/сек, которую обеспечивают краулеры с низкой производительностью [20], потребуется около 20 суток, чтобы собрать указанные данные. Такое время одной итерации неприемлемо, имея в виду, что предполагается наличие многих итераций для исследования динамики изменения связей.

В качестве отдельного ограничения отметим стоимость искомого инструмента. Для научных исследовательских целей правильнее рассматривать только инструменты с открытым исходным кодом, т.к. они распространяются бесплатно и их исходный код доступен для анализа и редактирования, что позволяет настраивать инструменты под меняющиеся требования, зачастую возникающие в ходе научного процесса.

Типы краулеров

Рассмотрим основные типы краулеров, которые принято выделять в литературе [6, 7], для того чтобы категоризовать искомые инструменты и понимать накладываемые на них ограничения.

Сфокусированный краулер (Focused Web Crawler) – это краулер, задача которого заключается в том, чтобы загрузить связанные с друг другом страницы по конкретной теме. Такой вид краулеров также называют тематическим (Topic Crawler).

Принцип его работы основывается на том, что каждый раз переходя к новому документу, данный краулер проверяет, насколько он релевантен обозначенной теме, переход осуществляется только на документы, соответствующие теме. Его достоинства состоят в том, что он экономичен и не требует значительных вычислительных ресурсов.

Инкрементный краулер (Incremental Crawler) – традиционный краулер, который периодически обновляет собранные в своем хранилище документы. Помимо замены старых версий документов на новые, он может обновлять ранг документов, сортируя их на менее важные и более важные.

Распределенный краулер (Distributed Crawler) – это тип краулера, базирующегося на распределенных вычислениях. Как правило, он реализован на нескольких вычислительных узлах, один из которых назначается главным, а другие являются дочерними узлами.

Этот тип краулеров использует алгоритмы типа Page Rank для улучшения релевантности поиска. Достоинство этого вида краулеров в их надежности и отказоустойчивости.

Параллельный краулер (Parallel Crawler) – такой краулер состоит из нескольких краулер-процессов, каждый из которых работает над своим выбранным множеством данных.

Кроссплатформенный краулер (Cross-platform Crawler) – такой краулер должен одинаково устанавливаться и настраиваться на машинах с различными операционными системами.

Требования к краулерам

Приведем основной стандартный набор требований к краулерам в соответствии с [19, 26]:

1. **Надежность.** Веб содержит ресурсы, которые могут вводить краулер в бесконечный цикл или отправлять на недоступные сервисы, ожидать выполнения которых он не должен.

2. **Вежливость.** Веб-ресурсы имеют явные и неявные политики, регулирующие частоту, с которой краулер может посетить их (как правило, они описаны в файле robots.txt).

3. **Распределенность.** Краулер должен иметь возможность запускаться на выполнение в распределенном режиме на нескольких машинах.

4. **Масштабируемость.** Краулер должен поддерживать возможность увеличения производительности за счет добавления дополнительных вычислительных узлов.

5. **Производительность и эффективность.** Краулер должен обеспечивать эффективное использование системных ресурсов.

6. **Качество.** Краулер должен уметь отделять спам-страницы от полезных, реализовывать процедуры нормализации URL, предотвращая дублирование обработки.

7. **Актуальность.** Краулер должен поддерживать обновление собранных данных.

8. **Расширяемость.** Краулер должен позволять добавлять новую функциональность для анализа новых форматов данных, протоколов и т.д.

Для нашего класса содержательных задач необходим инструмент, который должен:

1) быть кроссплатформенным, чтобы его можно было одинаково конфигурировать на вычислительных узлах с разными операционными системами;

2) обеспечивать производительность обработки порядка 100 страниц/сек, чтобы время сбора описанного выше объема данных составляло часы, а не дни;

3) интегрироваться с базой данных для хранения информации и полнотекстовым индексом, позволяющим быстро извлекать собранные данные;

4) реализовывать стратегию сбора данных в ширину и вертикального поиска, так как в нашей задаче необходима информация о предметной области, а не узкое множество фактов.

Обзор основных первоисточников

Есть несколько источников информации, дающих представление относительно тех или иных программных продуктов с открытым исходным кодом, но далеко не все из них можно воспринимать как достоверные.

Например, из перечня на блог-ресурсе «Top 50 open source web crawlers» [27] можно почерпнуть названия веб-краулеров с открытым исходным кодом, но при этом не разъясняется, на основании чего было произведено ранжирование краулеров. Более того, здесь содержатся ссылки на устаревшие и не развивающиеся инструменты.

Другой список предоставляет Best Open Source: 36 best open source web crawler projects [6]. Рейтинг основан на общественном рейтинге проектов с открытым исходным кодом. Список не является полным, например в него не попал известный и широко используемый краулер Scrapy. Обоснование ранжирования краулеров можно увидеть в академических обзорах, таких как [11, 23].

Первоисточники позволили определить базовый перечень из восьми краулеров, которые будут рассмотрены далее.

Краулеры с открытым исходным кодом

Далее приводится краткое описание наиболее популярных краулеров с открытым исходным кодом, взятых из приведенных выше первоисточников.

1. **Nutch** [3]. Типизация: инкрементный, параллельный, распределенный, кроссплатформенный. Инструмент поиска в Вебе и краулер, написанный на java, поддерживающий граф связей узлов, различные фильтры и нормализацию URL [22].

Интегрирован со свободной библиотекой полнотекстового поиска Apache Lucene [2], набором утилит, библиотек и фреймворков для разработки и выполнения распределённых программ Apache Hadoop [4] и позволяет использовать хранилища данных, такие как СУБД Cassandra [1].

Является масштабируемым (до 100 узлов в кластере), легко настраивается и расширяется, является «вежливым». Заявленная производительность 1000 страниц/сек. Более детальное описание можно найти в [29].

2. **Scrapy** [25]. Типизация: сфокусированный, параллельный, кроссплатформенный. Расширяемый и гибкий краулер, написанный на Python, который легко устанавливается, поддерживает выгрузку данных в форматах JSON, XML, CSV.

Подходит для сфокусированного сбора данных в Вебе. «Вежливый», устойчивый, расширяемый краулер. Считается, что данный краулер менее производительный, нежели Apache Nutch [23].

3. **Open Search Server** [18]. Типизация: инкрементный, параллельный, кроссплатформенный. Краулер и поисковый движок, ядро которого написано на java, являющийся «вежливым» и поддерживающий современные подходы к полнотекстовому поиску.

Обеспечивает автоматическую классификацию текстов и подключение базы синонимов, поддерживает 18 языков. Использует такие технологии, как Apache Lucene, фреймворк для разработки веб-приложений ZK [17] и другие. Это надежный и производительный инструмент [9, 10].

4. **Norconex HTTP Collector** [16]. Типизация: инкрементный, параллельный, кроссплатформенный. Позволяет выполнять задачи крулинга и сохранять собранные данные в любое настроенное хранилище, в том числе и в поисковый индекс, написан на java, является «вежливым».

Производительный, поддерживает все востребованные функции: фильтрацию, нормализацию, а также распознавание языков, гибкость извлечения данных и т.д. Активно развивается и поддерживается коммерческим проектом [14].

5. **Bixo** [7]. Типизация: инкрементный, параллельный, распределенный, кроссплатформенный. Написан на java. Работает на основе Cascading (платформа для создания приложений на Hadoop) [8].

Позволяет переносить данные в предметно-ориентированную информационную базу данных Data Warehouse [9].

Расширяемый, настраиваемый, устойчивый и «вежливый». Масштабируем до 1000 вычислительных узлов и подходит для анализа данных больших объемов [23].

Разработчики этого инструмента сделали его после разработки одного из проектов для вертикального поиска, основанного на Nutch [11]. Данный инструмент рекомендуют использовать для таких задач, как нахождение и анализ комментариев по поводу конкретного продукта, отслеживание популярности объекта в социальной сети, анализ данных о стоимости продукта и т.д. [7].

6. **Crawler4j** [12]. Типизация: параллельный, кроссплатформенный. Написан на java с простым API (Application Programming Interface – интерфейс прикладного программирования). С его помощью можно легко организовать многопоточный краулинг.

Данный инструмент можно легко встроить в проект, но при этом не поддерживается индексирование. Является «вежливым», но обнаружено, что может порождать излишнюю нагрузку на исследуемый хост [20, 23].

7. **Arachnode.net** [13]. Типизация: инкрементный, параллельный, кроссплатформенный. Написан на C# с использованием платформы .NET и SQL Server 2008.

Является «вежливым» и поддерживает загрузку, индексацию и сохранение веб-контента, включая адреса e-mail, файлы и веб-страницы.

Сведения о производительности разноречивы: в [21] говорится о 3145 страницах/сек, но в свободной версии данный краулер имеет гораздо более низкую производительность 1,29 страниц/сек [20].

8. **GNU Wget** [30]. Типизация: сфокусированный. Продолжающий развиваться инструмент, написанный на C на Linux-платформе, который позволяет получать файлы из наиболее широко использующихся интернет-протоколов (HTTP, HTTPS, FTP). Данный краулер подходит в основном для выгрузки конкретных данных или сайта, но не масштабного краулинга сегментов Веба.

Выбор краулера

Кратко напомним дополнительные требования из пункта 1:

1) кроссплатформенность,

2) производительность обработки порядка 100 страниц/сек,

3) интеграция с базой данных и полнотекстовым индексом,

4) стратегия сбора данных «вначале вширь, потом в глубину»,

5) актуальность.

Таким образом, следуя этим требованиям:

- ввиду требования (1) GNU Wget не подходит, так как может использоваться только на платформе Linux;
- ввиду требования (2) Arachnode.net не подходит, так как в свободной версии данный краулер имеет производительность 1,29 страниц/сек;
- ввиду требования (3) Crawler4j и Norconex HTTP Collector не подходят, так как не имеют интегрированного хранилища и/или поискового индекса;
- ввиду требования (4) Scrapy не подходит, так как является сфокусированным, то есть не реализующим стратегию сбора данных «вначале вширь потом в глубину»;
- требование (5) выполняется всеми рассмотренными краулерами с открытым исходным кодом.

Из оставшихся трех программных продуктов Nutch, Open Search Server и Bixo по типизации все они почти одинаковы (у Open Search Server отсутствует распределенность, что, по-видимому, и влияет на производительность). По заявленной производительности наиболее предпочтительным представляется выбор Apache Nutch.

Заключение

Рассмотрены основные типы краулеров и дополнительные требования, сформулированные как результат предварительного изучения предметной области, представляющей собой некоторое заданное ограниченное, но достаточно большое множество представительских веб-сайтов.

В результате систематизации и анализа соответствия основным и дополнительным требованиям проведен отбор трех наиболее перспективных краулеров.

Список литературы

1. Apache Cassandra. http://cassandra.apache.org (дата обращения: 06.02.2017).

2. Apache Lucene. http://lucene.apache.org (дата обращения: 06.02.2017).

3. Apache Nutch™. http://nutch.apache.org (дата обращения: 01.03.2017).

4. Apache Hadoop. https://wiki.apache.org/hadoop (дата обращения: 06.02.2017).

5. Bar-Ilan J. Data collection methods on the Web for infometric purposes: A review and analysis // Scientometrics. January 2001. – Vol. 50(1). – P. 7–32.

6. Best Open Source: 36 best open source web crawler projects. http://www.findbestopensource.com/tagged/webcrawler?start=0 (дата обращения: 11.01.2017).

7. Bixo – Web Mining Toolkit. https://openbixo.files.wordpress.com/2010/01/bixo-web-mining-talk-at-hug.pdf (дата обращения: 02.02.2017).

8. Cascading | Application Platform for Enterprise Big Data. http://www.cascading.org (дата обращения: 06.02.2017).

9. Data Warehousing Overview. http://dwreview.com/DW_Overview.html (дата обращения: 06.02.2017).

10. Examples using Common Crawl Data. http://commoncrawl.org/the-data/examples (дата обращения: 06.02.2017).

11. Granados N.F., Kauffman R.J., King B. The Emerging role of vertical search engines in travel distribution: a newly-vulnerable electronic markets perspective // Proceedings of the 41st Hawaii International Conference on System Sciences. – 2008. – P. 389.

12. GitHub – yasserg/crawler4j: Open Source Web Crawler for Java. https://github.com/yasserg/crawler4j (дата обращения: 01.03.2017).

13. Home – arachnode.net. http://arachnode.net (дата обращения: 01.03.2017).

14. Khurana D., Kumar S. Web Crawler: A Review // International Journal of Computer Science & Management Studies. – 2012. – № 12–1. – P. 401–405.

15. Leading Enterprise Java Web Framework | ZK. https://www.zkoss.org (дата обращения: 06.02.2017).

16. Norconex HTTP Collector. http://www.norconex.com/product/collector-http (дата обращения: 01.03.2017).

17. Olston C., Najork M. Web Crawling // Foundations and Trends in Information Retrieval. – 2010. – Vol. 4, № 3. – P. 175–246.

18. OpenSearchServer | Open Source Search Engine and Search API. http://www.open-search-server.com (дата обращения: 01.03.2017).

19. Papavassiliou V., Prokopidis V., Thurmair G. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web // Proceedings of the 6th Workshop on Building and Using Comparable Corpora., Sofia, August 8, 2013. – P. 43–51.

20. Performance Benchmark. https://arachnode.net/blogs/arachnode_net/archive/2008/12/31/performance-benchmark-etc.aspx (дата обращения: 4.02.2017).

21. Performance Tuning. https://arachnode.net/blogs/arachnode_net/archive/2015/04/12/performance-tuning.aspx (дата обращения: 04.02.2017).

22. RFC 3986 – Uniform Resource Identifier (URI). 2005. https://tools.ietf.org/html/rfc3986 (дата обращения: 04.02.2017).

23. Ricardo A., Serrão C. Comparison of existing open-source tools for Web crawling and indexing of free Music // Journal of telecommunications. – 2013. – № 18–1. https://ru.scribd.com/doc/123153248/Comparison-of-existing-open-source-tools-for-Web-crawling-and-indexing-of-free-Music.

24. Scrapinghub Platform. http://scrapinghub.com/platform (дата обращения: 12.01.2017).

25. Scrapy A Fast and Powerful Scraping and Web Crawling Framework. http://scrapy.org (дата обращения: 01.03.2017).

26. Sharma G., Sharma S., Singla H. Evolution of web crawler its challenges // International Journal of Computer Technology and Applications. – 2016. – № 9(11). – P. 5357–5368.

27. Top 50 open source web crawlers for data mining. http://bigdata-madesimple.com/top-50-open-source-web-crawlers-for-data-mining (дата обращения: 11.01.2017).

28. Udapure T.V., Kale R.D., Dharmik R.C. Study of Web Crawler and its Different Types // Journal of Computer Engineering. 2014. № 16-1. http://iosrjournals.org/iosr-jce/papers/Vol16-issue1/Version-6/A016160105.pdf.

29. Web Crawling with Apache Nutch. http://events.linuxfoundation.org/sites/events/files/slides/aceu2014-snagel-web-crawling-nutch.pdf (дата обращения: 05.02.2017).

30. Wget – GNU Project – Free Software Foundation. http://www.gnu.org/software/wget (дата обращения: 01.03.2017).