3 Ландо Т. Извлечение из текстов объектов и фактов (Text mining) [Текст] – Яндекс, Отдел лингвистических технологий, 2015. – 49 с.

4 Никоненко, А.А. Обзор баз знаний онтологического типа [Текст] – Киев: Киевский национальный университет имени Т. Шевченко, 2009. – 12 с.

А.А. Шарипов, А.Р. Мавлютов, А.Ф. Атнабаев

АНАЛИЗ И ИЗВЛЕЧЕНИЕ СОДЕРЖИМОГО ИНФОРМАЦИОННЫХ РЕСУРСОВ СЕТИ INTERNET СРЕДСТВАМИ ЯЗЫКА PHP И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

(Уфимский государственный авиационный технический университет, г.Уфа)

К концу 2016 года по всему миру будет передано 1,1 зеттабайта данных. Это настолько много информации, что ни один человек даже за тысячу жизней не смог бы проанализировать. Не сложно заметить, что сайты, принадлежащие либо магазинам, либо разного рода организациям не содержат в себе столько данных, сколько содержат гиганты как Википедия или Ютуб, но даже для их анализа у человека потребуется большое количество времени.

Чаще всего найти и собрать полезную информацию вручную просто нереально, проблема не только в объеме информации, но и в допущении человеком ошибок, и это нормально, ведь человек не машина и ему свойственно совершать ошибки. Поэтому авторами была поставлена цель разработать программу для сбора, анализа и сохранения проанализированной полезной информации, которая бы решала все возможные проблемы. Изучив информацию по данной тематике стало известно, что такого рода программы в англоговорящих странах называют “Parser”. Они могут быстро обрабатывать информацию, но чаще всего платные и стоят не малых денег.

В информационных технологиях программы синтаксического анализа называются “Parser” (далее – парсер), а сам процесс получения информации с помощью программы — парсинг (от англ. Parsing, далее – парсинг), принятое в информатике определение синтаксического анализа, то есть сопоставление лексем с формальной грамматикой. Работу парсера можно сравнить с человеком, который ищет необходимую ему информацию и записывает ее, или сохраняет, чтобы использовать ее в дальнейшем. Алгоритм сопоставления описывается в математической модели на одном из языков программирования. Например, PHP, Perl, Ruby.

Независимо от того на каком формальном языке программирования написан парсер, алгоритм его действия остается одинаковым:

- выход в интернет, получение доступа к коду веб-ресурса и его скачивание;
- чтение, извлечение и обработка данных;
- представление извлеченных данных в удобоваримом виде – файлы .txt, .sql, .xml, .html и других форматах.

Парсер не может воспринимать информацию как человек, и чтобы получить результат он должен сравнить заданный программистом набор букв, слов, выражений и знаков программного синтаксиса с кодом с веб-ресурса. Такой набор называется «регулярное выражение». Чтобы парсер понимал регулярные выражения, он должен быть написан на языке, поддерживающем их в работе со строками. Такая возможность есть в php, Perl[1].

В качестве написание парсера был выбран серверный язык PHP, он имеет свои плюсы и минусы, но входе анализа было принято решение использовать его, так как он обладает набором качеств для удобного парсинга[2]:

- у него есть встроенная библиотека libcurl, с помощью которой скрипт подключается к любым типам серверов, в том числе работающих по протоколам https (зашифрованное соединение), ftp, telnet;
- PHP поддерживает регулярные выражения, с помощью которых парсер обрабатывает данные;
- у него есть библиотека DOM для работы с XML – расширяемым языком разметки текста, на котором обычно представляются результаты работы парсера;
- он отлично ладит с HTML, поскольку создавался для его автоматической генерации.

Процесс анализа среднестатистического источника информации представленного в виде популярного веб-ресурса является достаточно долгим и может достигать в среднем около месяца. Так как необходимо было проанализировать огромное количество страниц, что в свою очередь заняло бы длительное количество времени, была задача сократить это время. Проблема в том, что обычный парсер не использует все ресурсы компьютера и интернета. Когда php скрипт делает запрос к странице сайта, то он достаточно долго ожидает ответ этого сайта и только после ответа переходит к следующему шагу алгоритма. Для решения этой задачи использовался Ajax – это библиотека для javascript предоставляющая возможность асинхронности процессов в веб-приложениях. Скорость парсинга при этом ускорилась в несколько раз. Рекомендуется не открывать потоков больше 100, потому что большая частота запросов может подорвать работу ресурса и будет идентифицирована как DDoS-атака, что в свою очередь является нарушением закона Российской Федерации, при этом это не подрывает ни каким образом работоспособность анализированного веб-ресурса [3].

В ходе проектирования была выявлена проблема, что долго работающий парсер, отработав не малое количество времени мог прерваться из-за разного рода ошибок. И это приводило к тому, что терялась возможность понять, сколько сайтов было обработано. Решением этой проблемы является созданием логов, например, файла txt, или же подключением скриптов к базе данных, и записывание в файл txt последние обработанные веб-страницы.

Изучив различные методы получения нужной части кода, предлагается, что самым удобным будет использование библиотеки phpQuery, так как регулярные выражения не очень удобны для разбора HTML кода. Использование

библиотеки phpQuery, позволяет облегчить работу, так как не приходится писать сложные регулярки для получения блоков сайта, а вместо этого можно обращаться к ним с помощью селекторов CSS.

Так как разрабатываемый инструмент анализа данных предназначен для работы на персональном компьютере, предполагается наличие следующих проблем: отключение интернет соединения, нестабильная работа компьютера, ограничение количества потоков и блокировка IP адреса. Одними из способов решения этих проблем являются:

- проверкой интернет соединения перед каждым циклом;
- выставление более продолжительной задержки;
- уменьшение количества потоков;
- увеличение времени в параметре Apache timeout, он ограничивал время соединения с сервером в 32 секунды, что не позволило бы запускать больше 30 потоков.

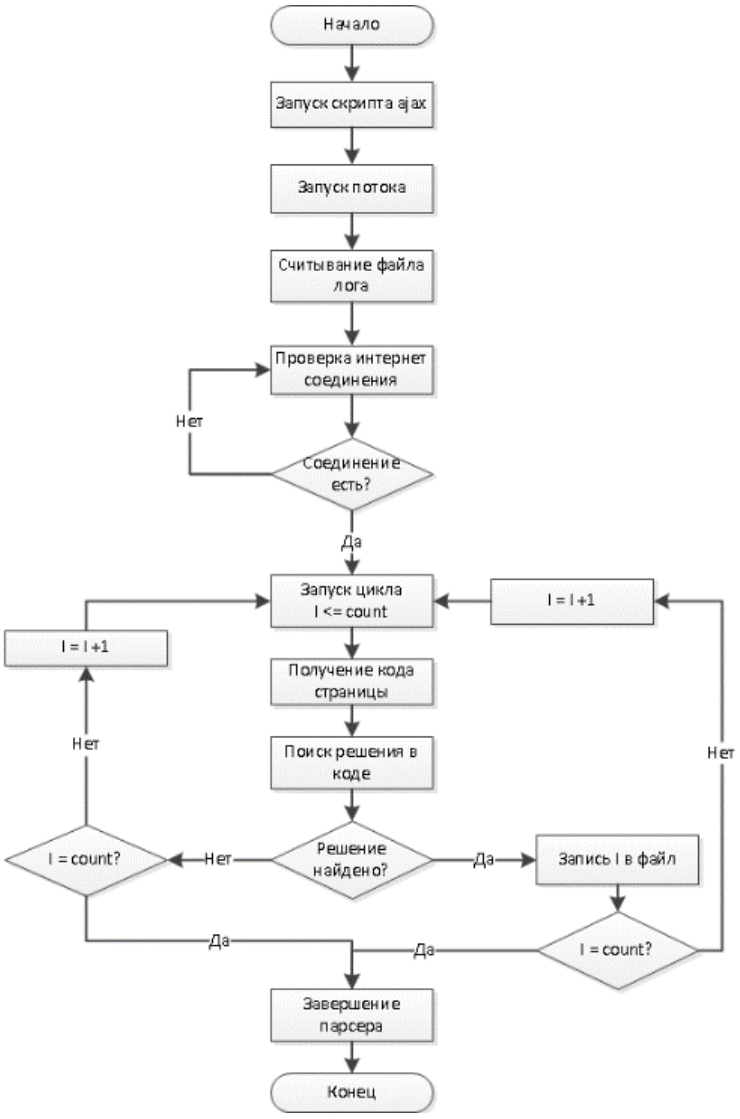

Рис. 1

Авторы попробовали использовать интернет технологии и IT- технологии для поиска полезной информации, при этом, не нарушая работоспособность анализируемого сайта, связанные с бесперебойной работой ПО, оборудования и какого-либо нарушения авторских прав. Для анализа работоспособности созданной программы, были проанализированы данные лежащие в свободном доступе на популярных веб-ресурсах. В результате удалось проанализировать и извлечь полезную информацию, для использования их в будущих работах. Парсер работал в 50 потоков и за 24 часа проверил больше миллиона страниц, из которых полезной информации составило 4%. Парсер работал стабильно, за все время работы сбоев обнаружено не было. Ниже представлена схема работы парсера:

Литература

1. Преимущества PHP: [Электронный ресурс] // URL: http://www.php.su/php/?opport (Дата обращения: 25.02.2018).
2. Использование вредоносных компьютерных программ: [Электронный ресурс] // URL: https://www.zakonrf.info/uk/273 / (Дата обращения: 27.02.2018).
3. phpMyAdmin по-русски / Установка PHP 5.3.10: [Электронный ресурс] // URL: http://php.net/manual/ru/intro-whatcando.php (Дата обращения: 25.02.2018).
4. My PHP / Возможности PHP: [Электронный ресурс] // URL: https://php-myadmin.ru/learning/instrument-php.html (Дата обращения: 25.02.2018).
5. Павлов С.В., Христодуло О.И. Разработка метода объединения данных из различных информационных систем в единую информационную систему Минэкологии РБ // Научный журнал «Вестник УГАТУ». – Уфа, Т.15, № 2(42). 2011г.– С.3-7.

Р.А. Шаталин, П.Е. Овчинников, В.Р. Фидельман

АЛГОРИТМ ОБНАРУЖЕНИЯ НЕХАРАКТЕРНОГО ПОВЕДЕНИЯ НА ОСНОВЕ ГЛАВНЫХ КОМПОНЕНТ И КОМБИНАЦИИ ХАРАКТЕРИСТИК ПЛОТНЫХ ТРАЕКТОРИЙ ДВИЖЕНИЯ

(Нижегородский государственный университет им. Н.И. Лобачевского)

За последние годы множество алгоритмов для обнаружения конкретных нештатных ситуаций было предложено и реализовано на основе строгих правил [1]. Данные алгоритмы позволяют с высокой надежностью обнаруживать заранее известные типы нештатных ситуаций, но их модификация для обнаружения иного типа аномалий нетривиальна. Для устранения этого недостатка было предложено несколько статистических подходов к задаче видеонаблюдения.

Одним из подходов является распознавание событий на основе статистических методов классификации. Успехи в области распознавания событий на