МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

СОГЛАСОВАНО
Генеральный директор
ЗАО «АйТи»

Бакиев О.Р.
"__" _____ 2011 г.

УТВЕРЖДАЮ
Ректор НИУ ИТМО

Васильев В.Н.
"__" _____ 2011 г.

МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE

ПРОГРАММНЫЙ КОМПОНЕНТ СБОРА ДАННЫХ В СОЦИАЛЬНЫХ СЕТЯХ В ИНТЕРНЕТ CLAVIRE/HADRAWLER

ОПИСАНИЕ ПРОГРАММЫ

ЛИСТ УТВЕРЖДЕНИЯ

RU.СНАБ.80066-06 13 39-ЛУ

Представители
Организации-разработчика

Руководитель разработки,
профессор НИУ ИТМО

_____ Бухановский А.В.
"__" _____ 2011 г.

Ответственный исполнитель,
с.н.с. НИУ ИТМО

_____ Луценко А.Е.
"__" _____ 2011 г.

Нормоконтролер
ведущий инженер НИУ ИТМО

_____ Позднякова Л.Г.
"__" _____ 2011 г.

2011

Инв.№ подл.	Подп. и дата	Взам.инв.№	Инв.№ дубл.	Подп. и дата

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

УТВЕРЖДЕН

RU.СНАБ.80066-06 13 39-ЛУ

МНОГОПРОФИЛЬНАЯ ИНСТРУМЕНТАЛЬНО-ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА СОЗДАНИЯ И УПРАВЛЕНИЯ РАСПРЕДЕЛЕННОЙ СРЕДОЙ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ CLAVIRE

ПРОГРАММНЫЙ КОМПОНЕНТ СБОРА ДАННЫХ В СОЦИАЛЬНЫХ СЕТЯХ В ИНТЕРНЕТ CLAVIRE/HADRAWLER

ОПИСАНИЕ ПРОГРАММЫ

RU.СНАБ.80066-06 13 39

ЛИСТОВ 15

Подп. и дата	
Инв.№ дубл.	
Взам.инв.№	
Подп. и дата	
Инв.№ подл.	

2011

АННОТАЦИЯ

Документ содержит описание программного компонента CLAVIRE/Hadrawler RU.СНАБ.80066-06 01 39, реализующего сбор индивидуальных данных пользователей социальной сети в Интернете и данных о структуре такой сети, а также обработку собранных данных. Программный компонент реализует различные стратегии обхода сети, основанные как на топологии, так и на индивидуальных свойствах узлов сети. Программный компонент CLAVIRE/Hadrawler разработан в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ

Программный компонент CLAVIRE/Hadrawler RU.СНАБ.80066-06 01 39 предназначен для сбора (краулинга) индивидуальных данных пользователей социальной сети в Интернете и данных о структуре такой сети, а также обработку собранных данных. Компонент реализует различные стратегии обхода сети, основанные как на топологии, так и на индивидуальных свойствах ее узлов. Программный компонент разработан в ходе выполнения проекта «Создание распределенной вычислительной среды на базе облачной архитектуры для построения и эксплуатации высокопроизводительных композитных приложений» (Договор № 21057 от 15 июля 2010 г., шифр 2010-218-01-209) в рамках реализации постановления Правительства РФ № 218 «О мерах государственной поддержки развития кооперации российских высших учебных заведений и организаций, реализующих комплексные проекты по созданию высокотехнологичного производства».

Для работы компонента требуется следующее системное и программное обеспечение: ОС семейства Linux, платформы Java (версии 1.6 и выше), библиотеки с открытым исходных кодом Apache Hadoop (версии 0.20.2 и выше). Hadrawler может функционировать как отдельное приложение, так и как сервис, встраиваемый в распределенную среду облачных вычислений под управлением многопрофильной инструментально-технологической платформы (МИТП) CLAVIRE RU.СНАБ.80066-06.

Программный компонент реализован в виде набора заданий для библиотеки Apache Hadoop, осуществляющих главные функции компонента. Для реализации использовался язык программирования Java.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Программный компонент Hadrawler реализует следующие функции:

1. Сбор данных о структуре сети – граф дружбы пользователей сети.
2. Сбор контекстов пользователей сети. Под контекстом понимаются индивидуальные данные пользователей. В случае сети Livejournal – это интересы пользователей и их текстовые записи.
3. Интеллектуальный обход сети, при котором обходятся неявные сообщества пользователей, заинтересованных в одной теме и «дружащих» друг с другом.

На данный момент в программном компоненте Hadrawler реализован сбор данных только из социальной сети Livejournal, но он может быть адаптирован и для сбора данных из других социальных сетей.

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1. Принципы функционирования

В основе функционирования Hadrawler лежат два модуля: модуль, реализующий сбор информации из сети Интернет, и модуль, реализующий выбор и ранжирование узлов сети, информацию о которых требуется собрать.

Перед компонентом, осуществляющим краулинг, ставится задача обхода многих социальных сетей больших размеров, что непосредственно влияет на выбор его архитектуры. Для реализации требуемых задач требуется надежная и легко масштабируемая база данных, обеспечивающая хранение информации, и масштабируемая система, позволяющая свободно увеличивать число машин, с которых осуществляется обращение к социальным сетям для получения информации. Также следует отметить необходимость минимизации перемещения данных между внутренними узлами системы. Поэтому для основы реализации этого модуля Hadrawler была выбрана библиотека с открытым исходным кодом Apache Hadoop, которая позволила построить поверх распределенной файловой системы Apache HDFS надежное и масштабируемое хранилище данных, предоставив при этом удобный и надежный фреймворк для реализации алгоритмов анализа больших корпусов текстов.

В задачи Hadrawler, реализованного на основе библиотеки Hadoop, входит запуск, мониторинг и контроль распределенных операций. При таком подходе каждый из модулей краулера получает на вход однотипные данные, которые ему необходимо обработать, обрабатывает их и записывает результат в распределенную файловую систему. Результаты работы одного модуля, являются входными данными для работы другого модуля, поэтому следующий модуль начинает работу только после завершения работы текущего модуля. Таким образом, можно говорить, что модули выполняются последовательно друг за другом. Отметим, что, в силу распараллеливания обработки данных реализованную посредством библиотеки Apache Hadoop, каждый модуль будет обрабатывать часть данных параллельно на всех машинах кластера.

В основе принципа распределенной работы Hadrawler лежит разбиение данных по машинам кластера. Все данные в файловой системе Apache HDFS хранятся блоками на разных машинах кластера. Библиотеку Hadoop можно настроить таким образом, чтобы все файлы разбивались на число блоков, равное числу машин в кластере, и чтобы на каждой машине хранился один блок данных. Тогда при обработке этих данных каждый блок

будет обрабатываться одной машиной и тем самым будет реализован принцип распределенной обработки данных.

Для реализации *тематической политики обхода* краулера используется подход, основанный на анализе контекста пользователя социальной сети: его интересов и его документов, в предположении, что люди, интересующиеся схожими темами, образуют неявные сообщества в социальной сети, и между ними существуют связи. Таким образом, обход сети выглядит следующим образом:

1) анализ интересов и/или документов пользователя и проверка их соответствия заданным темам. Интересы и документы пользователя образуют контекст;
2) если контекст пользователя отнесен к определенной теме, то «друзья» добавляются в очередь узлов для обхода;
3) если контекст не отнесен к определенной теме, то продолжить работу.

Для содержательного анализа текстов и интересов пользователя используются методы, основанные на применении списка ключевых слов, которые достаточно полно описывают некоторую тему, например, тему наркотизации населения. Для этого эксперт предметной области составляет список ключевых слов – наиболее известные и однозначные термины, которые четко определяют тематику текста, а также различные многозначные жаргонизмы, что вносит определенную погрешность. Отметим, что каждое ключевое слово семантически отнесено к некоторой более конкретной теме. Так, для темы наркотиков, экспертом были выделены следующие ключевые слова и словосочетания: шприц, иглы, первитин, эфедрон, героин, марихуана, кокаин, опий-сырец, таблетки, инъекции, сленг приготовления, общие слова.

Учитывая морфемное многообразие, эксперт предметной области указывает только паттерн ключевых слов. Выделены следующие паттерны слов.

1) Точная (конкретная) форма слова – например, паттерн «доза».
2) Префикс слова – фиксируется только начало слова, суффиксы и окончания могут быть любыми. Например, паттерн: «наркот*», под который попадают «наркотики», «наркота».
3) Суффикс слова – фиксируется окончание слова, приставки могут быть любыми. Например, паттерн «*курить», под который попадают: «покурить», «вкурить».
4) Задана середина слова – и приставки, и суффиксы могут варьировать. Например, паттерн «*колбасит*».

5) Стемминг слова – посредством различных эвристик, специфичных для каждого языка, от слова отбрасываются окончания и суффиксы, и получается словоформа, близкая по своей структуре к объединению приставки и корня слова. Данные эвристики не всегда имеют высокую точность, но обладают быстродействием, что в рассматриваемом случае является критичным параметром. Существуют различные наборы эвристик для каждого языка, в программном компоненте Hadrawler было реализовано семейство эвристик, называемых «стеммер Портера».

В результате в документе проверяется наличие ключевых слов и фраз – неупорядоченных наборов ключевых слов, которые свидетельствуют о тематической принадлежности текста. При наличии *хотя бы одного ключевого слова или фразы* считается, что документ посвящен заданной теме, что инициирует обход «друзей» текущего пользователя. Данный подход модернизируется за счет введения весов: для окончательного определения принадлежности документа определенной тематике высчитывается его вес, равный сумме весов всех входящих в него ключевых слов и ключевых фраз. Затем полученный вес сравнивается с пороговой величиной: полагается, что все документы с весом больше этого порога имеют заданную тематику. Пользователь с б*о*льшим (относительно порога) весом документа добавляется в очередь для обхода.

3.2. Программная архитектура

Архитектура программного компонента представлена на рис. 3.1. Каждый модуль представляет собой отдельное задание в терминах библиотеки Hadoop, которое считывает некоторую информацию из распределенной базы данных, обрабатывает ее и записывает результат обратно, после чего управление передается следующему модулю. Результаты работы одного модуля, являются входными данными для работы другого модуля, поэтому следующий модуль начинает работу только после завершения работы текущего модуля. Таким образом, можно говорить, что модули выполняются последовательно друг за другом, образуя при этом итерационный процесс. При этом на каждой итерации краулер обрабатывает фиксированное число пользователей.

На рис. 3.1 приведена схема работы краулера на одной итерации (эллипсы – процессы, проводимые над данными, а прямоугольники – данные). Процессы выполняются последовательно друг за другом, и их взаимное расположение по оси абсцисс указывает на порядок выполнения. На первом этапе работы краулера происходит сбор информации об узлах из социальной сети, затем анализируются контексты

пользователей, информацию о которых удалось собрать на этой итерации. На этом этапе и происходит тематическая классификация текстов пользователей согласно принципам, описанным в предыдущей главе. На третьем этапе создается новая очередь для обхода краулером, т. е. создается список не посещенных узлов, ранжированных исходя из значения приоритета их посещения.

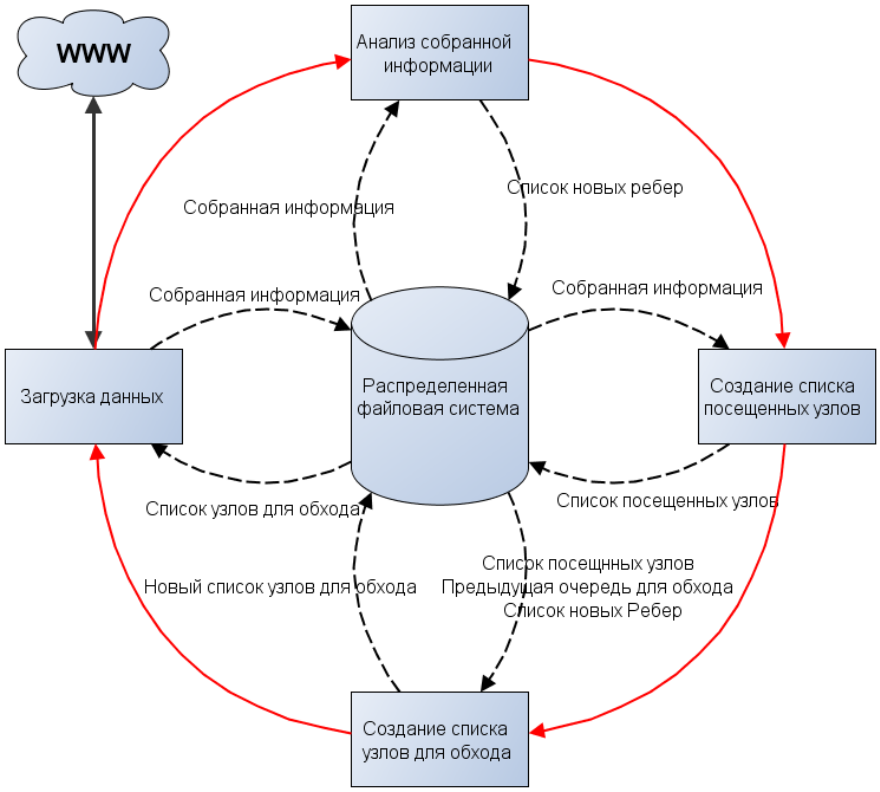

Рисунок 3.1 – Архитектура программного компонента, построенного на основе библиотеки ApacheHadoop

Для этого анализируются все найденные на итерации ребра и исходящие из отобранных тематическим классификатором вершины, определяется, что они ведут в еще не посещенные узлы, для чего в память загружается список посещенных узлов. Полученные вершины представляют собой первую часть новой очереди для обхода. Затем из списка узлов для обхода, построенного на предыдущей итерации, отфильтровываются узлы, которые еще не были посещены. Если обнаружилось, что какой-либо узел не посещен вследствие ошибки, счетчик числа попыток посещения увеличивается. Затем оба списка узлов объединяются, при этом происходит обновление атрибутов узла (числа попыток, затраченных на его посещение и приоритет узла), происходит удаление из очереди узлов, для которых счетчик числа попыток превысил заданное значение. Далее

полученный список узлов ранжируется согласно значению приоритета и становится списком узлов для обхода на новой итерации.

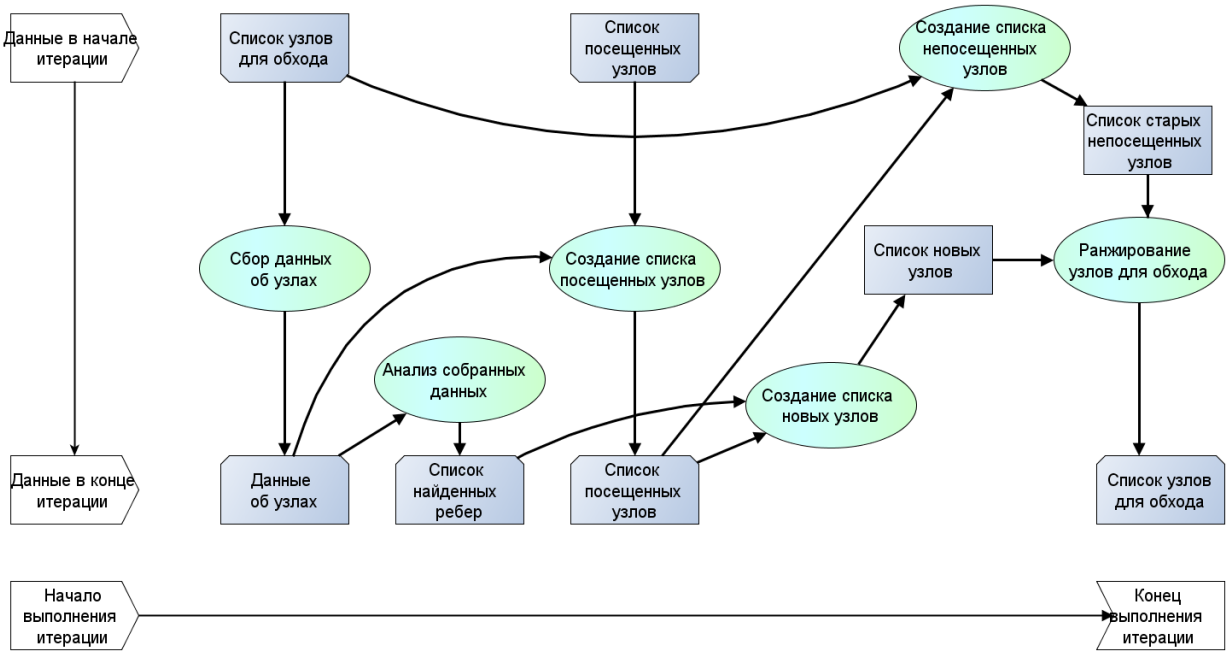

Рисунок 3.2 – Схема работы программного компонента на одной итерации

Приоритет узла считается равным числу найденных ссылок, указывающих на него. Чем больше ссылок, тем выше приоритет и тем вероятнее, что узел содержит информацию по заданной тематике. Число ссылок в узле легко поддержать в актуальном состоянии от итерации к итерации, оно равно сумме ссылок, найденных на предыдущих итерациях, и числу ребер, найденных на текущей итерации, ведущих в этот узел.

3.3. Основные классы

Ниже приводятся сокращенные описания структуры и методов основных классов программного компонента.

3.3.1. Пакет ru.ifmo.hadrawler

Данный пакет содержит высокоуровневые классы для работы с Hadrawler.

Основные классы

- Класс ru.ifmo.hadrawler.Crawler – выполняет итерации.

- Класс ru.ifmo.hadrawler.CrawlDB – абстракция распределенной базы данных краулера.
- Класс ru.ifmo.hadrawler.Segment – абстракция одного сегмента базы данных, хранящего данные, полученные на одной итерации работы краулера.

3.3.2. Пакет ru.ifmo.hadrawler.crawl

Данный пакет содержит классы, являющиеся абстракцией данных, которые используются в программном компоненте.

- Класс ru.ifmo.hadrawler.crawl.NodeInfo – абстракция узла социальной сети.
- Класс ru.ifmo.hadrawler.crawl.FetchInfo – описывает состояние посещения узла.
- Класс ru.ifmo.hadrawler.crawl.FetchData – содержит часть контекста узла сети .

3.3.3. Пакет ru.ifmo.hadrawler.fetcher

Функционально этот модуль соответствует первому этапу работы краулера – сбору информации из сети.

Основные классы

- Класс ru.ifmo.hadrawler.fetcher.LJFetcher – реализует задание для Hadoop, которое обходит узлы, поданные ему на вход.
- Класс ru.ifmo.hadrawler.visitors.LJNodeVisitor – непосредственно скачивает данные контекста пользователя.
- Класс ru.ifmo.hadrawler.fetcher. LJPartsFetcher – скачивает элементы контекста пользователя.

3.3.4. Пакет ru.ifmo.hadrawler.links и ru.ifmo.hadrawler.filters

Функционально этот модуль соответствует второму этапу работы Hadrawler – классификации контекстов пользователей и созданию списка ребер.

Основные классы

- Класс ru.ifmo.hadrawler.links.LinksExtractor – выполняет задание для Hadoop, которое классифицирует узлы и создает список новых ребер.

- Класс ru.ifmo.hadrawler.links.VisitedNodesFilterReducer – классифицирует узлы на посещенные, для которых был получен полный контекст, и на не посещенные.
- Класс ru.ifmo.hadrawler.filters.PhraseMatcher – реализует поиск заданных выражений в тексте.
- Класс ru.ifmo.hadrawler.filters.AhoKeywordsMatcher – реализует поиск слов в тексте.

3.3.5. Пакет ru.ifmo.hadrawler.frontier

Функционально этот модуль соответствует третьему этапу работы Hadrawler – созданию списка посещенных узлов и созданию новой очереди для обхода.

- Класс ru.ifmo.hadrawler.frontier.Frontier – создание списка посещенных узлов и создание новой очереди для обхода.
- Класс ru.ifmo.hadrawler.frontier.VisitInfoCreator – выполняет задание для Hadoop, которое создает список посещенных узлов.
- Класс ru.ifmo.hadrawler.frontier.QueueCreator – выполняет задание для Hadoop, которое создает новую очередь для обхода.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

Для работы программного компонента необходим кластер машин с установленной библиотекой Apache Hadoop. К машинам этого кластера предъявляются следующие требования:

- архитектура процессора – x86, x86_64, IA64;
- объем оперативной памяти – 1 ГБ;
- объем свободного пространства на жестком диске – 100 ГБ;
- тактовая частота процессора – 1 ГГц;
- операционная система – Linux;
- доступ к сети Интернет.

Программный компонент требует для своей работы наличия следующего системного ПО: ОС семейства Linux, Java (версии не ниже 1.6), библиотеки Apache Hadoop (с версией не ниже 0.20.2).

5. ВЫЗОВ И ЗАГРУЗКА

Программный компонент реализован в виде приложения, которое выполняется на кластере машин под управлением библиотеки Apache Hadoop. Для запуска приложения реализован ряд скриптов на языке Bash, осуществляющих необходимую для работы сервиса подготовку параметров среды окружения (формирование переменной CLASSPATH, загрузка в распределенную файловую систему HDFS необходимых библиотек) и запуск приложения средствами Hadoop. При запуске приложения часть кода, осуществляющего процесс управления итерациями краулера, выполняется на машине, осуществившей запуск, а часть кода, реализующего непосредственно процесс краулинга, выполняется на всех машинах кластера, для чего используются средства библиотеки Hadoop.

6. ВХОДНЫЕ ДАННЫЕ

Входными данными для программного компонента являются: словарь терминов для построения классификатора текстов, набор начальных узлов, с которых начинается обход социальной сети, число итераций, которые необходимо совершить краулером.

Словарь терминов для построения классификатора задается в виде xml-файла специального формата, отдельные элементы которого приведены в листинге 6.1.

Листинг 6.1. Пример (отрывок) словаря терминов для пакета Hadrawler

```
<?xml version="1.0" encoding="UTF-8"?>
<words>
    <!-- Weights map -->
    <weight id="w_A" value="10"/>
    <weight id="w_B" value="6"/>
    <weight id="w_C" value="3"/>
    <weight id="w_C2" value="6"/>
    <weight id="w_B1" value="10"/>
    <weight id="w_B2" value="5"/>
    <weight id="w_A2" value="13"/>
<!-- Keywords map for group A -->
    <word_a name="абстяг*" weight="w_A" deff="общие_слова">
        <word_b name="ломк*" weight="w_A+w_B+w_B2" deff="ломка"/>
        <word_b name="отлом*" weight="w_A+w_B+w_B2" deff="ломка"/>
        <word_b name="бахнут*" weight="w_A+w_B+w_B2" deff="инъекция">
            <word_c name="вливат*" weight="w_A+w_B+w_B2+w_C2"
deff="инъекция"/>
            <word_c name="лекарств*" weight="w_A+w_B+w_B2+w_C2"
deff="наркотик"/>
        </word_b>
    </word_a>
<!-- Alias for group A -->
```

```
    <ampl_a word_1="анаш*" word_2=" башатумн*" weight="w_A+w_A2"/>
    <ampl_a word_1="анаш*" word_2=" бошк*" weight="w_A+w_A2"/>
<!-- Keywords map for group B -->
    <word_b name="агрегат*" weight="w_B" deff="шприц"/>
    <word_b name="астрал*" weight="w_B" deff="общие_слова"/>
    <word_b name="барыга*" weight="w_B" deff="общие_слова"/>
    <word_b name="бахнут*" weight="w_B" deff="инъекции"/>
    <!-- Alias for group B -->
    <ampl_b word_1="гарик*" word_2="[дурь]" weight="w_B+w_B1"/>
    <ampl_b word_1="гарик*" word_2="[дым]" weight="w_B+w_B1"/>
    <ampl_b word_1="гарик*" word_2="жарех*" weight="w_B+w_B1"/>
    <!-- Keywords map for group C -->
    <word_c name="боинг*" weight="w_C" deff="шприц"/>
    <word_c name="емкост*" weight="w_C" deff="шприц"/>
    <word_c name="каранд*" weight="w_C" deff="шприц"/>
    <word_c name="[конь]" weight="w_C" deff="шприц"/>
    <word_c name="[машина]" weight="w_C" deff="шприц"/>
</words>
```

Файл, содержащий список узлов, с которых краулер начинает обход сети, имеет вид текстового файла, на каждой строке которого записано имя пользователя. Пример приведен в листинге 6.2.

Листинг 6.2. Пример (отрывок) файла со списком начальных узлов для пакета Hadrawler

```
Likesky
umka379
barhano
```

7. ВЫХОДНЫЕ ДАННЫЕ

В качестве выходных данных программный компонент предоставляет всю собранную им информацию о посещенных пользователях, которая хранится в распределенной файловой системе. Помимо этого он создает на локальном диске файл, содержащий структуру обойденной сети. Формат этого файла следующий:

- Узлы идентифицируются числами.
- Если между узлами есть ребро, то в выходной файл записывается информация вида: `номер_первой вершины--номер_второй_вершины`.

Чтобы этот формат файла был корректно разобран прикладными сервисами МИТП, в начале файла добавляется строка «`cluster_1`», а в конце файла добавляется строка «`cluster_cross`»

Пример:

```
cluster_0
```

1--2

2--3

cluster_cross.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД	База данных
МИТП	Многофункциональная инструментально-технологическая платформа
ОС	Операционная система
ПК	Программный компонент
ПО	Программное обеспечение

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Лист регистрации изменений									
Изм.	**Номера листов (страниц)**				**Всего листов (страниц) в докум.**	**№ документа**	**Входящий № сопроводительного докум. и дата**	**Подп.**	**Дата**
	изме-ненных	**заме-ненных**	**но-вых**	**аннули-рован-ных**					