INTERNET ТА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.7

Пугач Я.Ю.
Криворожский национальный университет

РАЗРАБОТКА ТЕХНОЛОГИИ ДЛЯ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ КЛИЕНТОВ В СЕТЯХ USENET

В статье рассматривается проблема получения целостных данных в сетях Usenet. Приведены недостатки традиционного подхода, заключающегося в прямом использовании резервных аккаунтов. В качестве альтернативы предлагается решение, основанное на применении промежуточного сервера-диспетчера, контролирующего взаимодействие клиентов с резервными серверами.

В статті розглядається проблема отримання цілісних даних в мережах Usenet. Наведено перелік недоліків традиційного підходу, що полягає у прямому використанні резервних акаунтів. У якості альтернативи запропоновано технологію, в основі якої знаходиться проміжний сервер-диспетчер, що контролює взаємодію клієнтів з резервними серверами.

The paper describes the problem of integral data retrieval in Usenet. The downsides are given for the traditional approach of using supplementary accounts directly. Alternatively it proposes the solution based on the use of the dispatching server that manages the network interaction between clients and supplementary servers.

Ключевые слова: информационное обеспечение, Usenet, сервер-диспетчер.

Введение

Usenet - компьютерная сеть, используемая для общения и публикации файлов. Usenet состоит из новостных групп, в которые пользователи могут посылать сообщения. Сообщения хранятся на серверах, которые обмениваются ими друг с другом. Usenet в настоящее время основан на протоколе NNTP, который является протоколом прикладного уровня модели OSI.

В настоящее время практически весь Usenet-трафик передаётся по Интернету, а формат сообщений и способ их передачи очень похож на электронную почту. Однако если электронная почта используется для общения «один на один», то Usenet действует по принципу «один для всех». Сообщения, которые пользователь публикует в Usenet, организуются в тематические категории, называемые новостными группами или конференциями, которые организуются в иерархию, подобную структуре доменных имён. Например, группы sci.math и sci.physics находятся внутри иерархии sci. С помощью приложений для работы с Usenet можно подписаться на любые доступные конференции.

При отправке сообщения, оно доступно только на его сервере. Но каждый news-сервер обменивается сообщениями с несколькими соседними, и таким образом сообщение распространяется на каждом news-сервере Интернета. Таким образом, отправка сообщения инициируется отправителем, а не получателем. [1]

Объёмы публикуемых сообщений в Usenet составляют терабайты информации, поэтому хранение и распределение этой информации предполагает большие расходы. Именно поэтому доступ платный, а значит трафик здесь только коммерческий. [2]

В последнее время возрастает популярность Usenet как среды для обмена файлами, особенно в Западной Европе, что в первую очередь связано с юридическим давлением на пиринговые сети (торренты).

Постановка задачи

Данные в сетях Usenet хранятся на серверах в сегментированном виде. Серверы частично или полностью дублируют хранящуюся у них информацию. После отправки сообщения на сервер оно постепенно распространяется на все news-сервера. Таким образом, данные на серверах синхронизируются, однако каждый сервер имеет ограничения на объем информации доступной для загрузки с других сер-

веров, кроме того, ограничен и срок хранения данных. Этот срок различен на разных серверах и фактически представляет собой лишь усредненный показатель времени жизни сегмента. Вследствие этой особенности клиент, обращаясь к конкретному серверу, может не получить нужных ему данных. В то же время эти данные могут быть получены с другого сервера.

Наиболее простым решением проблемы отсутствующих сегментов является прямое обращение к резервным серверам, на которых провайдер имеет определенное количество оплаченных аккаунтов. Однако это порождает три другие проблемы:

1. Усложняется управление процессом скачивания (клиентская программа должна сама заботиться о выборе нужного сервера и распознавать их ответы, которые могут в определенной мере отличаться; провайдер же будет испытывать сложности с распределением нагрузки на сервера).

2. Если провайдер решит купить дополнительные аккаунты или, напротив, срок действия некоторых из них закончится, необходимо извещать об этом клиентов и вносить изменения на клиентской стороне. При условии использования клиентского программного обеспечения, произведенного провайдером или его партнерами, этот процесс может быть автоматизирован, однако многие пользователи по тем или иным причинам предпочитают стороннее ПО, не имеющим интеграции с веб-узлами поставщика услуг.

3. Возникает риск кражи данных для аутентификации на резервных серверах, так как логин и пароль отправляются с клиентской машины.

В данной работе рассматривается другой подход – использование специального сервера-диспетчера, который с точки зрения клиента выглядит как обычный NNTP-сервер, но не хранит никаких данных, а лишь пересылает запросы резервным серверам, о существовании которых клиенты могут и не знать.

Изложение материала и результаты

Для получения данных клиент обращается к news-серверу, на котором у него есть аккаунт. Записанный в терминах протокола NNTP диалог сервера и клиента, пытающегося скачать сегмент файла, может выглядеть следующим образом:

[C] AUTHINFO USER DutchMeat
[S] 381 Password Required
[C] AUTHINFO PASS OpenSesame
[S] 281 Authentication Accepted
[C] BODY <i.am.not.there@fake.com>
[S] 430 No Such Article Found

В данном случае, запрошенный сегмент найти не удалось. Как правило, это связано с тем, что сегмент старый, и на его место были записаны новые данные. Другая распространенная причина – законодательная: сегмент был удален из-за нарушения авторских прав. Так, если по каким-либо причинам основной сервер не возвращает данных, клиент может запросить их с другого, резервного сервера, однако каждый резервный сервер также требует наличия аккаунта, предоставляющего доступ к данным. Количество одновременных подключений для каждого аккаунта ограничено, поэтому возможна ситуация, что к резервному серверу в данный момент нельзя подключиться под некоторым логином, не говоря уже о возможных технических проблемах на сервере или на сетевом канале. Возможно и то, что на резервном сервере, с которого клиент пытается скачать сегмент, эти данные также не могут быть найдены. Решение этих проблем тривиально – необходимо попытаться использовать другой резервный аккаунт. Следующий диалог клиента и сервера демонстрирует оба этих случая, а также ситуацию успешного скачивания сегмента:

[C] AUTHINFO USER SupplementaryLogin1
[S] 381 Enter passphrase
[C] AUTHINFO PASS SupplementaryPassword1
[S] 502 Connection Limit Reached
…
[C] AUTHINFO USER SupplementaryLogin2
[S] 381 Password required
[C] AUTHINFO PASS SupplemetaryPassword2
[S] 281 Authentication accepted
[C] BODY <missing.segment@weed.nl>
[S] 430 No such article
…
[C] AUTHINFO USER SupplementaryLoginK
[S] 381 Magic phrase, please
[C] AUTHINFO PASS SupplementaryPasswordK
[S] 281 Welcome to the place of the last hope
[C] BODY <missing.segment@weed.nl>
[S] 222 0 <missing.segment@weed.nl>
[S] You’ve got what you came for.

[S] Now go away.
[S] .

Ситуация, когда данные не могут быть получены с основного сервера, возникает довольно часто, однако не является обычной – необходимость в дополнительных аккаунтах возникает лишь время от времени. Исходя из этого, можно заключить, что для обеспечения N клиентов достаточно K резервных аккаунтов, существенно меньшее N. Резервные аккаунты находясь в общем пользовании, могут быть задействованы различными пользователями по мере необходимости.

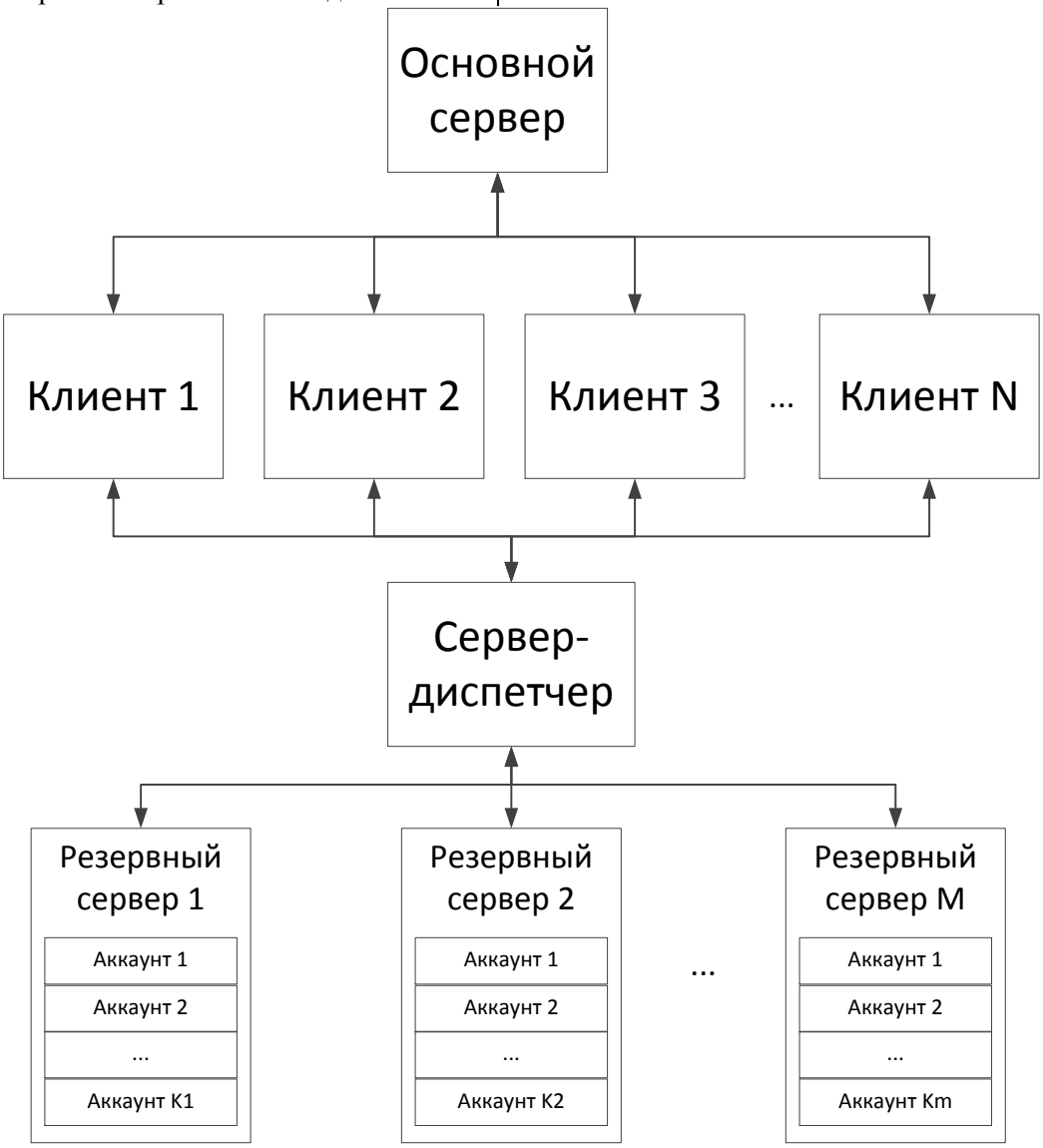

Рис. 1. Технология информационного обеспечения клиентов в сетях Usenet

На рис. 1 показана схема сетевого взаимодействия клиентов и основного сервера, а также резервных серверов через сервер-диспетчер, который устанавливает подключение к резервным серверам данных в случае успешной аутентификации клиента. Клиенты, количество которых может быть произвольным и ограничивается только техническими возможностями сервера, отправляют на него запросы для получения данных. Если данные могут быть получены с основного сервера, они возвращаются клиенту, а диспетчер не используется. В противном случае клиент подключается к серверу-диспетчеру. Как и обычный NNTP-сервер, диспетчер для большинства операций требует аутентификации. Аутентификация на сервере-диспетчере является первичной, для ее осуществления используются тот же логин и пароль, что и для доступа к основному серверу. Эта процедура необходима, чтобы ограничить доступ незарегистрированных пользователей к резервным серверам.

Данные для первичной аутентификации пользователей, которым разрешен доступ к

резервным серверам, хранятся в таблице allowed_users базы данных SQLite.

Табл. 1
Структура таблицы пользователей

Поле	Тип	Ограничение
id	INTEGER	PRIMARY KEY
login	TEXT	NULL
password	TEXT	NULL

В таблице пользователей (табл. 1) хранятся md5-хеши логинов и паролей, что в случае взлома базы очень усложнит задачу определения реальных данных для аутентификации. Первичная аутентификация сводится к поиску записи таблицы, значения полей login и password которой совпадают с хешами переданного логина и пароля:

```
SELECT * FROM allowed_users WHERE login = :login AND password = :password LIMIT 0, 1
```

Если такая запись найдена, первичная аутентификация считается успешно выполненной.

После прохождения первичной аутентификации клиентом при необходимости (например, запрос на скачивание сегмента файла) диспетчер извлекает из локальной базы данных сведения для вторичной аутентификации на одном из резервных серверов данных.

Данные для вторичной аутентификации хранятся в таблице supplementary_accounts.

Табл. 2
Структура таблицы резервных аккаунтов

Поле	Тип	Ограничение
id	INTEGER	PRIMARY KEY
server_address	TEXT	NULL
ssl_server_address	TEXT	NULL
port	NUMERIC	NULL
ssl_port	NUMERIC	NULL
login	TEXT	NULL
password	TEXT	NULL

Поскольку сервер-диспетчер может обслуживать как обычные TCP-подключения, так и безопасные подключения по протоколу SSL, для того чтобы установить соединение с резервным сервером нужны соответствующие адреса сервера и номер порта для NNTPS-сервера, который использует шифрование SSL, и NNTP-сервера, не использующего шифрование. Таким образом, таблица supplementary_accounts (табл. 2) хранит адреса серверов (server_address, ssl_server_address), номера портов (port, ssl_port), а также логин и пароль к соответствующему серверу. В некоторых случаях адреса NNTPS- и NNTP-сервера могут совпадать или же сервер может не предоставлять один из видов подключения, тогда поле остается незаполненным (хранит NULL).

В терминах SQL извлечение данных для вторичной аутентификации на сервере, использующем SSL, можно записать следующим образом:

```
SELECT ssl_server_address, ssl_port, login, password FROM supplementary_accounts WHERE ssl_server_address IS NOT NULL AND ssl_port IS NOT NULL
```

Выборка данных для вторичной аутентификации на сервере, который не использует шифрование SSL, осуществляется аналогично с тем лишь различием, что мы выбираем поля

server_address и port вместо ssl_server_address и ssl_port:

SELECT server_address, port, login, password FROM supplementary_accounts WHERE server_address IS NOT NULL AND port IS NOT NULL

Вторичная аутентификация по причине ошибки или истечения таймера сессии может производиться несколько раз без ведома клиента.

Принципиально важным в работе сервера-диспетчера является выполнение команд по очереди, т.е. новая поступившая команда не должна выполняться до того как будет получен ответ от предыдущей команды. В противном случае возможно смешивание данных или неверный порядок ответов, когда результат выполнения команды, поступившей позже, предшествует результату выполнения команды, поступившей раньше.

Ответы, полученные от серверов данных, анализируются диспетчером и при необходимости подвергаются изменению, затем они возвращаются клиентскому приложению. Если выполнение команды на одном из серверов завершается неудачей, диспетчер, проанализировав ответ, попытается установить новое подключение, используя для этого данные другого аккаунта. Этот процесс повторяется до тех пор, пока команда не завершится удачно, либо не будут использованы все аккаунты. Возможны ситуации, когда диспетчер не будет пытаться использовать другой аккаунт, например, если соединение с сервером прервалось, но часть данных запрошенного сегмента уже получена. В этом случае диспетчер не может продолжать обработку запросов данного подключения, поскольку ответ на предыдущую команду еще не был полностью отправлен клиенту, и он не будет отправлен, так как разорвалось соединение с сервером-источником, поэтому клиентское приложение должно отключиться и при необходимости запросить данные заново.

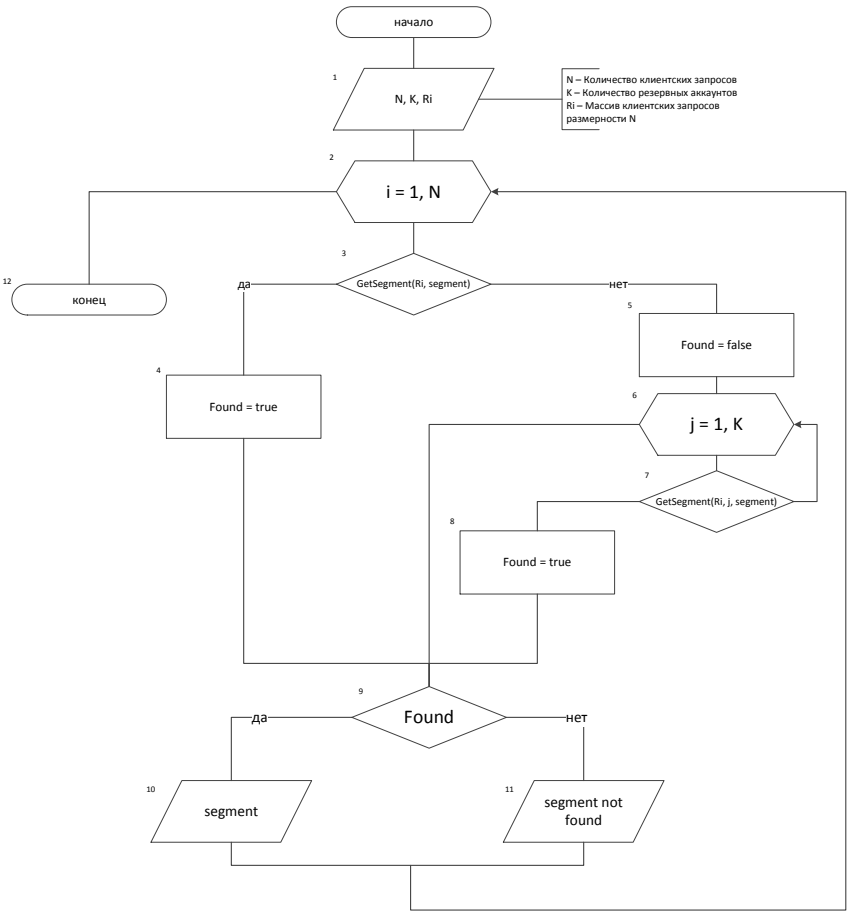

Рис. 2. Технология информационного обеспечения клиентов в сетях Usenet

На рис. 2 представлен алгоритм, описывающий процедуру обмена данными, которая обеспечивает лучшую целостность файлов по сравнению с использованием лишь основного сервера. Исходными данными является количество клиентских запросов N, массив запросов Ri, а также количество резервных аккаунтов K. В блоке 2 циклически обрабатываются

все клиентские запросы на скачивание сегментов. В блоке 3 осуществляется попытка получения сегмента с основного сервера. В случае успеха, сегмент данных возвращается клиенту (блок 10). Если сегмент не найден, выполняется попытка получить его, используя один из резервных аккаунтов (блок 6). Если ни один из аккаунтов не позволил получить сегмент, клиенту возвращается ответ об отсутствие сегмента (блок 11). После обработки всех клиентских запросов алгоритм завершается в блоке 12.

Выводы

В работе предложена технология, позволяющая повысить целостность данных, получаемых из сетей Usenet. Использование сервера-диспетчера предоставляет возможность достаточно гибко контролировать процесс скачивания с резервных серверов, нужным образом организуя поиск сегментов и распределяя нагрузку. Это избавляет клиентские приложения от необходимости реализации собственной процедуры поиска, более того, добавление и изменение резервных аккаунтов происходит быстрее и проще. Также это существенно повышает безопасность данных для аутентификации, так как они отправляются с сервера-диспетчера, клиентский доступ к которому ограничен протоколом NNTP.

Список литературы

1. Usenet. [электронный ресурс] // - Режим доступа: http://ru.wikipedia.org/wiki/Usenet.

2. Usenet and Spotnet. [электронный ресурс] // - Режим доступа: http://www.rxpblog.com/usenet-and-spotnet/.

Сведения об авторе:

Пугач Ярослав Юрьевич, аспирант кафедры моделирования и программного обеспечения, научный руководитель – д.т.н. профессор Азарян А.А., Криворожский национальный университет. Научные интересы – мультимедиа технологии, компьютерные сети, организация данных, объектно-ориентированное программирование, теория графов.

e-mail: pugach.yaroslav2@gmail.com.