Сигнал от измерителей тока и напряжения, подключенных в цепь электродвигателя, поступает на датчики, которые, в свою очередь, передают его на аналого-цифровой преобразователь (АЦП), где происходит его дальнейшее преобразование в цифровой сигнал. Далее на отладочной плате, куда записана специальная программа в микроконтроллер, полученный цифровой сигнал обрабатывается, и полученные данные выводятся на экран, подключенный совместно с отладочной платой.

После чего уже обработанные данные можно передавать на персональный компьютер, где будет показан результат диагностики двигателя и предложены меры по устранению неисправностей.

В процессе исследования современных методов диагностики электродвигателей в данной статье особое внимание было обращено на метод спектрального анализа потребляемого тока. Данный метод делает возможным выявлять неисправности электрической и механической частей машины на ранних стадиях, а также существенно снизить трудоемкость и стоимость проводимой диагностики электрооборудования.

Таким образом, выполнение такой работы позволит провести комплексную диагностику электродвигателей и существенно сократить не только затраты, связанные с внезапными выходами из строя оборудования, но и уменьшить неэффективные затраты электроэнергии на неисправные электродвигатели.

Список литературы

1. Афанасьев Д. О.Обзор методов контроля технического состояния асинхронных двигателей в процессе эксплуатации / Д. О. Афанасьев // Тезисы докладов V Всероссийской конференции "Проблемы разработки месторождений углеводородных и рудных полезных ископаемых", проходящей в рамках Всероссийского молодёжного форума "Нефтегазовое и горное дело", г. Пермь, 14-16 ноября 2012 г. - Пермь: , Изд-во ПНИПУ, , 2012. - С. 130.

2. Передовой спектральный анализ / PeteBechard. PdMA Corporation. Перевод: Нафтулин И.В.

3. Петухов В.С., Соколов В.А. Диагностика состояния электродвигателей. Метод спектрального анализа потребляемого тока // Новости ЭлектроТехники. 2005, № 1(31), С. 50–52.

4. Седунин А. М.Методы вибродиагностики асинхронных электродвигателей / А. М. Седунин, Д. О. Афанасьев, Л. Г. Сидельников // Научные исследования и инновации. - , 2011. - Т. 5, № 2. - С. 191-194.

БЛОЧНАЯ РЕПЛИКАЦИИ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ В КОРПОРАТИВНЫХ СЕТЯХ

Буриченко Михаил Сергеевич
Сибирский Федеральный Университет
г. Красноярск

DOI: 10.31618/nas.2413-5291.2019.1.44.22

BLOCK REPLICATION OF THE CORPORATE NETWORK USERS DATA

Mikhail Sergeevich Burichenko
Siberian State University
Krasnoyarsk

Аннотация

В статье рассматривается проблема потери данных пользователей корпоративных сетей и ее решение, а именно репликация этих данных. Описана существующая технология "Roaming Profile", ее преимущества и недостатки, необходимые возможности альтернативного решения. В качестве модуля репликации данных был выбран проект «Syncthing».

Был разработан программный комплекс для автоматизированного централизованного управления взаимодействием «Syncthing» на рабочих станциях, с учетом многопользовательских операционных систем. Программный комплекс разработан на языке программирования C++ с использованием кросс-платформенной библиотеки QT. Поддерживает операционные системы семейства «Windows NT» начиная с «Windows 7» и GNU/Linux. Он состоит из двух частей – серверной и клиентской.

Abstract

The article considers the problem of the corporate networks users data lost and its solution, specifically the replication of this data. The article describes the existing technology «Roaming Profile», its advantages and disadvantages, the necessary possibilities of alternative solution. The project «Syncthing» was selected as the module of data replication.

The software complex for automated centralized management of «Syncthing» communication on the workstations subject to multi-user operation systems was developed. The software complex is developed in the C++ programming language using the cross-platform library QT. It supports operating systems of «Windows NT» family since «Windows 7» and GNU/Linux. It consists of two parts – the server part and the client part.

Ключевые слова: репликация данных.

Keywords: data replication.

Потеря данных – один из самых серьезных инцидентов, которые могут произойти в ИТ-инфраструктуре. Это данные сотрудников, накопленные годами, данные сервисов и служб, бухгалтерские и кадровые данные – их потеря недопустима.

На современных компьютерах, обладающих достаточными мерами безопасности, относительно редко происходит повреждение или потеря данных. Но иногда это всё же случается.

Чаще всего данные повреждаются во время записи на диск. При сохранении документа могут возникнуть разного рода ошибки. Современное программное обеспечение в большинстве случаев способно обработать такие ошибки и предложить варианты решения проблемы. Но иногда, к сожалению, пользователь не подозревает об этом, пока не попытается через некоторое время получить доступ к данным.

При завершении работы операционной системы, пользователю предлагается должным образом сохранить все файлы, с которыми он работает в данный момент. Когда этого не происходит, например, в случае аварийного отключения питания, у программ нет возможности закрыть все файлы правильно. Это может привести к повреждению данных, которые были открыты в этот момент, включая не только рабочие данные сотрудника, но и самой операционной системы.

Проблемы с жестким диском также могут привести к повреждению данных. Наличие дефектных физических секторов на диске, механические повреждения также являются частой причиной повреждения информации.

И, конечно, вредоносные программы могут испортить данные. Часто факт заражения компьютера обнаруживается, когда большая часть информации повреждена или уничтожена.

Для того чтобы решить проблему, необходимо периодически создавать резервные копии если не всех, то хотя бы критических данных. Желательно на другое устройство, а еще лучше на несколько.

Эта проблема в какой-то мере решается использованием технологии «Roaming Profile» (перемещаемый профиль пользователя) предоставляемой операционными системами «Microsoft Windows», семейства «Windows NT». Данная технология позволяет пользователям получить доступ к своему профилю на компьютерах, подключенных к домену «Active Directory» в пределах корпоративной сети. В профиль пользователя включаются не только документы, но и настройки окружения и установленных программ, а также ассоциации файлов.

Преимущества использования данной технологии в следующем:

- перемещаемый профиль позволяет пользователю автоматически получать доступ к своим данным с любого компьютера с операционной системой семейства «Windows NT», подключенного к корпоративной сети и входящего в домен «Active Directory»;
- использование перемещаемых профилей упрощает резервное копирование данных пользователей, предотвращая потерю данных при выходе из строя носителей информации на рабочих станциях;
- при замене компьютера нет необходимости вручную переносить документы и настройки пользователя на другую рабочую станцию, т.к. при входе в домен они автоматически загружаются с домена.

При всех преимуществах, недостатки данной технологии очень значительные и заключаются в следующем:

- загрузка данных на рабочую станцию происходит только во время входа пользователя в систему, а выгрузка только во время выхода, что делает невозможным полноценно взаимодействовать одновременно с несколькими устройствами;
- пользователь не может взаимодействовать с системой, пока профиль полностью не загрузится с домена (в случае с беспроводным или ограниченным подключением к корпоративной сети это может занимать десятки минут);
- для выгрузки профиля в случае разрыва соединения с доменом (отключение электропитания или активного оборудования), пользователю требуется выполнить повторный вход и выход из системы;
- проблемы с ассоциациями файлов на перенесенном профиле, когда на рабочих станциях установленно разное программное обеспечение закрепленное за одним и тем же расширением файлов («Microsoft Office» на одной и «LibreOffice» на другой). Например вы работали с файлами «*.docx» в «Microsoft Word». При перемещении профиля на другую рабочую станцию операционная система будет искать именно эту программу при попытке открыть файл «*.docx». Пользователю придется вызывать «контекстное меню», вызывать метод «Открыть с помощью...» и сохранять выбранную программу для определенных типов файлов. При работе на предыдущей рабочей станции придется повторять процесс заново, так как изменения перенесутся и на неё.

Из-за существенных недостатков данной технологии был начат поиск альтернативного решения, обладающего следующими возможностями:

- интерактивное взаимодействия с системой, не дожидаясь полной загрузки данных пользователя;
- поддержка загрузки только измененных блоков в данных (актуально для больших файлов);
- репликация данных одновременно между всеми рабочими станциями пользователя в пределах одной сети;
- поддержка нескольких последовательных версий файлов;
- поддержка различных операционных систем (в том числе находящихся в реестре отечественного программного обеспечения);
- передача данных по защищенным каналам связи;
- поддержка многопользовательских систем;

- поддержка централизованного хранения всех данных;
- централизованное управление настройками клиентов (рабочие станции сотрудников).

В качестве модуля репликации данных для программного комплекса был выбран проект «Syncthing» – кросс-платформенное программное обеспечение с открытым исходным кодом, позволяющее реплицировать файлы между несколькими устройствами по технологии «точка-точка».

Для того чтобы «Syncthing» реплицировал данные между двумя или более рабочими станциями необходимо:

- добавить уникальные идентификаторы рабочих станций (который автоматически генерируется при первом запуске «Synchting») в каждую из рабочих станций;
- добавить в каждую рабочую станцию директории, которые будут реплицироваться.

Соответственно, при добавлении новой рабочей станции в неё придется вручную добавлять идентификаторы имеющихся рабочих станций, а в имеющиеся новый идентификатор. Так же «Syncthing» не рассчитан на многопользовательские системы, а значит, не будет реплицировать данные пользователя на определенной рабочей станции, пока это не будет настроено вручную. Такие условия приемлемы для одного пользователя или семьи (домашний ПК, ноутбук, планшет), но не для корпоративной среды.

Основная задача разработанного программного комплекса – автоматизированное централизованное управление взаимодействием «Syncthing» на рабочих станциях, с учетом многопользовательских операционных систем.

Программный комплекс разработан на языке программирования C++ с использованием кросс-платформенной библиотеки QT. Поддерживает операционные системы семейства «Windows NT» начиная с «Windows 7» и GNU/Linux. Он состоит из двух частей – серверной и клиентской.

Серверная часть обеспечивает:

- обмен уникальными идентификаторами. Когда на новой рабочей станции впервые запускается клиентская часть, она передаёт на серверную часть уникальный идентификатор сгенерированный «Syncthing». Серверная часть уведомляет об этом администратора системы и ожидает от него ответа. При одобрении станции её идентификатор записывается в список участников обмена (список идентификаторов). Список рассылается всем участникам обмена по защищенным каналам связи. Таким образом, рабочая станция включается в кластер репликации данных. При удалении идентификатора станции из списка, он заново рассылается клиентам, и дальнейший обмен данными с этим участником прекращается;
- сбор статистики участников обмена. «Syncthing» хранит подробную статистику репликации данных (какие данные, когда и куда они посылаются, возникающие ошибки). Данные посылаются клиентской частью через защищенные каналы связи;
- хранение данных всех пользователей системы. Так как для реплицирования данных требуется минимум два активных участника обмена (две одновременно включенных рабочих станции), то в серверную часть так же включен «Syncthing», который собирает данные со всех участников обмена. Так обеспечивается избыточность – участник обмена, который всегда активен и хранит актуальные данные для репликации;
- передача клиентской части дистрибутива «Syncthing», для обновления до актуальной версии;
- управление настройками клиентской части.

Блок-схема алгоритма работы серверной части представлена на рисунок 1.

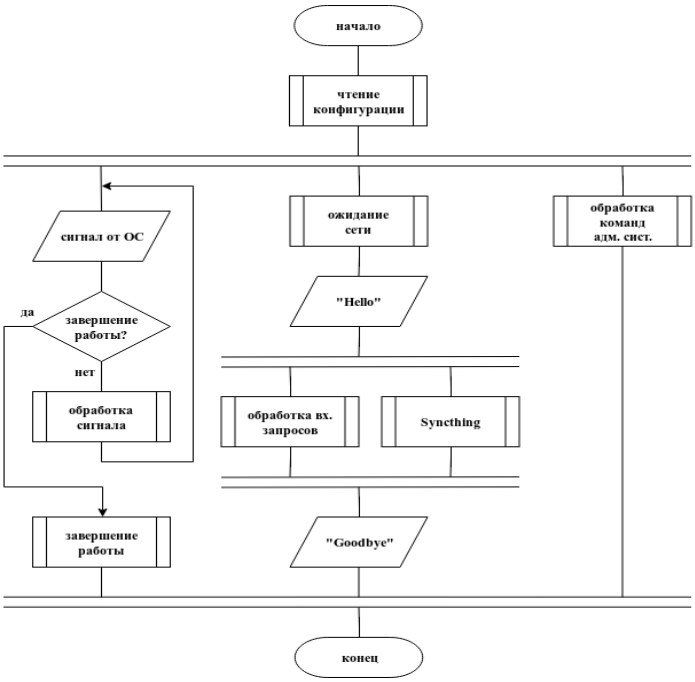

Рисунок 1. Блок-схема алгоритма серверной части

Алгоритм работы серверной части разделяется на несколько параллельных потоков:

- обработка событий (сигналов) от операционной системы. К примеру, обработка сигнала «SIGTERM», отвечающего за завершение процессов в POSIX-совместимых операционных системах;
- обработка входящих запросов от клиентской части.
- обработка команд администратора системы;
- запуск «Syncthing».

Клиентская часть разделена на два модуля – системный и пользовательский. Блок-схема алгоритма работы системного модуля клиентской части представлена на рисунок 2.

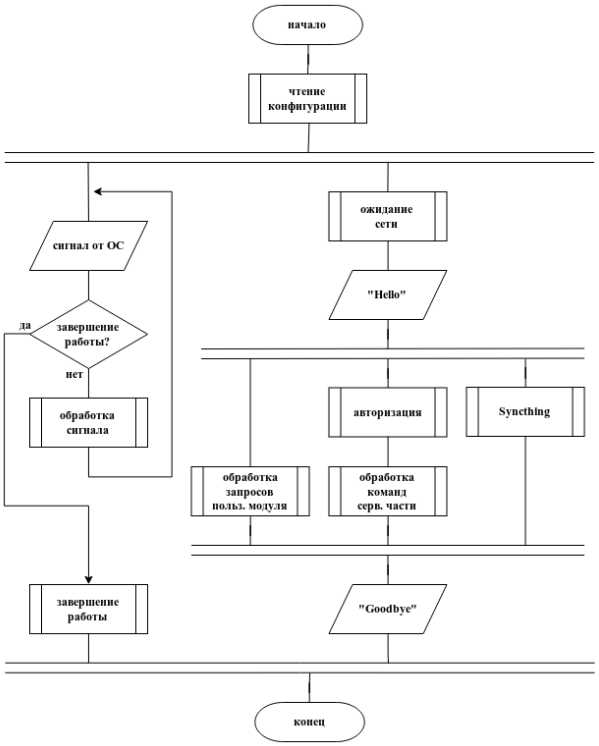

Рисунок 2. Блок-схема алгоритма системного модуля клиентской части

Системный модуль клиентской части работает как служба и обеспечивает:

- получение уникального идентификатора от «Syncthing» и отправка его на серверную часть;
- прием от серверной части списка участников обмена (списка идентификаторов) и передача его в «Syncthing»;
- изучение списка пользователей рабочей станции и добавление директорий с их данными в «Syncthing» для репликации (для операционных систем семейства «Windows NT» – это ветвь реестра «HKEY_LOCAL_MACHINE \ SOFTWARE \ Microsoft \ Windows NT \ CurrentVersion \ ProfileList». Для GNU/Linux требуется пакет «sssd», настроенный для взаимодействия с доменом «Active Directory»);
- прием от серверной части списка поддиректорий (включений, исключений) для репликации. Дополнение списка с учетом текущих пользователей рабочей станции и передача его в «Syncthing»;
- переход в автономный режим при обнаружении VPN подключения или при получении команды с серверной части. Репликация будет проводиться только «по требованию» от пользовательского модуля и только для данных пользователя, который запросил репликацию (необходимо в условиях лимитированного подключения к корпоративной сети);
- обработка запросов от пользовательского модуля клиентской части;
- обновление «Syncthing» до актуальной версии.

Она схожа с блок-схемой серверной части (рисунок 1), за исключением наличия авторизации рабочей станции на серверной части. Причем только для получения команд от серверной части. Для взаимодействия с пользовательской частью и запуска «Syncthing» авторизация не требуется (для возможности репликации данных, когда серверная часть недоступна).

Блок-схема алгоритма работы пользовательского модуля клиентской части представлена на рисунок 3.

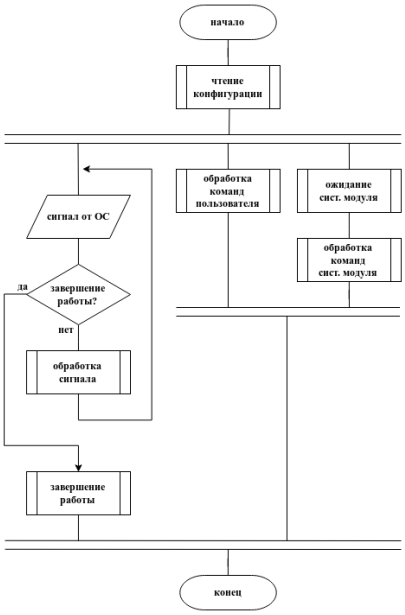

Рисунок 3. Блок-схема алгоритма пользовательского модуля клиентской части

Пользовательский модуль работает как подпрограмма, запущенная от имени активного пользователя рабочей станции и обеспечивает:

- отображение текущего состояния данных (прием, передача, список последних событий) через уведомления;
- добавление поддиректорий пользователя (включение, исключение) для репликации путем подачи запроса системному модулю клиентской части. Список поддиректорий заданный на серверной части имеет приоритет;
- вызов репликации «по требованию» путем подачи запроса системному модулю клиентской части;
- включение «последовательных версий файлов» для критичных, по мнению пользователя, данных. Системному модулю передается запрос на включение данной функции с расположением файлов относительно корневой директории профиля пользователя. Для операционных систем семейства «Windows NT» –это обычно «C:\Users\[username]», где «[username]» – логин пользователя. Также передаётся количество версий, которые требуется хранить (по последней дате изменения).

Когда авторизация клиентской части пройдена успешно, а директории пользователей рабочей станции добавлены – запускается процесс репликации данных. Первым делом «Syncthing» создаёт репозитории для добавленных директорий. Репозиторий представляет из себя таблицу содержащую метаданные (путь, размер, время создания, время изменения, занимаемый размер) для всех найденных файлов в директории и её поддиректориях. К примеру, для двух директорий «C:\Users\Пользователь1» и «C:\Users\Пользователь2» будет создано два репозитория – «Пользователь1» и «Пользователь2».

Каждый файл из репозитория разделяется на блоки определенного размера (от 128КБ до 16МБ, в зависимости от размера файла). Контрольные суммы (хэши) этих блоков добавляются к метаданным. К примеру, если блок равен 1Б, то файл в кодировке «UTF-8» с текстом «АБВ» будет разделен на три блока — «А», «Б» и «В».

Для репликации данных между устройствами «Syncthing» использует специальный протокол обмена блоками, разработанный основателем «Syncthing» Джейкобом Боргом. Взаимодействие по этому протоколу происходит между двумя или более узлами сети, которые образуют кластер. Каждый узел (участник обмена) имеет один или несколько репозиториев файлов образующих локальную модель (состояние данных у конкретного участника обмена). Локальная модель распространяется по защищенным каналам связи между всеми узлами в кластере. Каждый узел объединяет полученные локальные модели (репозитории других участников, содержащие метаданные и контрольные суммы блоков) формируя глобальную модель, включающую в себя последние изменённые версии файлов (по пути и дате изменения). К примеру, есть три участника обмена с репозиторием «Пользователь 1». Их локальные модели (ЛМ1, ЛМ2 и ЛМ3) представлены на рисунок 4.

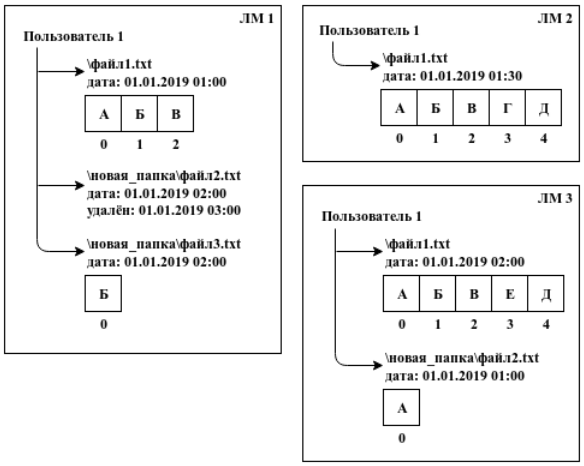

Рисунок 4. Локальные модели трех участников обмена

Из локальных моделей (рисунок 4) видно, что самый «свежий» (по дате изменения) файл «\файл1.txt» находится у третьего участника. Файл «\новая_папка\файл2.txt» есть только у участника один и три, но файл участника один был свежее и был удалён. Файл «\новая_папка\файл3.txt» есть только у участника 1.

С учётом содержания блоков (по контрольной сумме), сформированная участниками обмена глобальная модель будет иметь вид, представленный на рисунок 5.

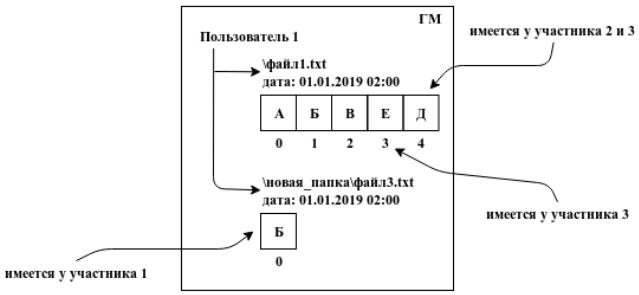

Рисунок 5. Глобальная модель

Каждый узел (участник) будет стремиться синхронизировать локальное хранилище данных с глобальной моделью, запрашивая по защищенным каналам связи у других узлов в кластере отсутствующие или обновившиеся блоки. Причем узлы будут стараться балансировать нагрузку на канал связи своих соседей. К примеру блок №3 для файла «\файл1.txt» есть только у участника №3, а блок №4 есть как у участника №3, так и у участника №2. Поэтому для участника №1 блок №3 файла «\файл1.txt» будет запрошен у участника №3, а блок №4 у участника №2.

Даже если серверная часть программного комплекса выключена или временно недоступна (техническое обслуживание), уже авторизованные клиенты (добавленные в кластер) смогут обмениваться данными между собой в пределах широковещательного домена (одной подсети) так как знают идентификаторы друг-друга и самостоятельно обмениваются локальными моделями формируя глобальную. Это удобно в случае работы нескольких рабочих станций за пределами основной корпоративной сети (к примеру, удаленный офис при обрыве связи). Когда серверная часть (или любой другой участник обмена) снова окажется в одной сети, она отправит сообщение «Hello» в пределах широковещательного домена. Все участники заново обменяются локальными моделями, формируя глобальную. На основе глобальной модели серверная часть запросит все изменения в данных, которые произошли за время отсутствия.

В заключение можно добавить — данный программный комплекс позволит автоматизировать перенос данных между рабочими станциями в корпоративной сети и их резервное копирование, что положительно скажется на рабочем процессе.

Литература:

1. Страуструп, Б. Язык программирования C++. Специальное издание : / Б. Страуструп; пер. с англ. С. Анисимова, М. Кононова; под общ. ред. Ф. Анреева, А. Ушакова – Москва : Бином-Пресс, 2004. – 1104 с.

2. Документация Qt [Электронный курс] – Режим доступа: http://doc.qt.io.

3. Документация Syncthing [Электронный курс] – Режим доступа: https://docs.syncthing.net.

4. ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения.