

Механизм защищенной синхронизации данных на базе протокола BitTorrent в распределенной вычислительной сети

BitTorrent Protocol as a Basic Mechanism of Data Protected Synchronization in a Distributed Computer Network

А. С. Макаревич, А. С. Карауш

Муниципальная информационная библиотечная система, Томск, Россия

*Alexander Makarevich and Alexander Karaush
Municipal Information Library System, Tomsk, Russia*

Описан новый алгоритм синхронизации данных на основе использования протоколов пиринговых сетей. Алгоритм предполагает разбиение синхронизируемого файла на блоки, каждый из которых передается по каналам связи самостоятельно. Таким образом, при незначительных изменениях больших файлов баз данных потребуется передать для синхронизации лишь незначительный объем информации. При этом всё взаимодействие между субъектами распределенной сети обеспечивается программным обеспечением, использующим протоколы пиринговых сетей.

Информационные технологии (ИТ) – одно из важнейших направлений развития любой современной организации. Использование вычислительной техники, набора программного обеспечения и каналов связи дает значительный прирост производительности во многих процессах, но по мере развития эти средства вырастают из простого инструмента в целую область деятельности, требующую серьезных вложений и постоянного обновления. Такая ситуация наблюдается часто: технических средств достаточно, ресурсы становятся распределенными, появляется необходимость координировать территориально разделенные вычислительные средства и хранилища данных, передавать информацию по каналам связи, синхронизировать базы данных. В результате этих процессов инфраструктура значительно усложняется. Дополнительной проблемой становится возрастающая нагрузка на вычислительные ресурсы и каналы связи – оба эти ресурса ограничены и могут быть исчерпаны на определенном этапе развития.

Необходимые понятия:

- уникальный ресурс (УР) – набор представленных в виде файлов данных, необходимых для работы каждому субъекту распределенной сети;
- субъекты распределенной сети – источник УР и клиенты;
- источник УР – подключенная к сети передачи данных ЭВМ, на дисковом пространстве которой хранится УР;
- клиент – подключенная к сети передачи данных ЭВМ, получающая УР по каналам связи от источника УР;
- длительность процесса синхронизации – время, необходимое для получения всеми клиентами актуальных копий УР;
- механизм синхронизации – приемы и правила, в соответствии с которыми проходит процесс синхронизации.

На практике часто появляется необходимость в рамках вычислительной сети осуществлять периодическую синхронизацию большого объема данных между несколькими субъектами. Такая сеть обладает следующими характеристиками:

1. имеются территориально распределенные, находящиеся за пределами одной локальной вычислительной сети, хранилища данных;
2. присутствует источник УР;
3. УР обладает значительным объемом – прямая передача всего объема по каналам связи всем клиентам не может быть осуществлена за заданный промежуток времени;
4. наличие каналов связи с малой пропускной способностью, в том числе с коммутируемым доступом, между источником УР и клиентами.

Задачи, которые необходимо выполнить для нормального функционирования рассматриваемой сети:

1. обеспечить всех клиентов актуальными копиями УР с заданной периодичностью обновления;
2. снизить нагрузку на каналы связи и вычислительные мощности источника УР;

3. сократить длительность процесса синхронизации;
4. обеспечить конфиденциальность передаваемой информации.

Для решения вышеперечисленных задач используются следующие технологии:

1. прямая передача УР по каналам связи;
2. клиент-серверная технология доступа к УР.

Первая технология решения неприемлема из-за большой длительности процесса синхронизации, вторая – из-за большой нагрузки на источник УР, которая возникает в связи с необходимостью централизованного управления процессом синхронизации и передачи копий УР каждому клиенту, а также низкой надежности при коммутируемом доступе. Для устранения этих ограничений предлагается учесть следующие особенности работы механизма синхронизации:

1. Сеть децентрализована. Это позволяет снизить нагрузку на вычислительные мощности источника УР и его каналы связи. Данное утверждение используется для построения пиринговых сетей.

2. УР передается не полностью. Передает только часть данных, изменившаяся с момента последней синхронизации. Файлы УР разбиваются на блоки, каждый из которых идентифицируется уникальным признаком, все клиенты получают список уникальных признаков, производят сравнение с уникальными признаками блоков данных, имеющихся у них, и запрашивают у источника УР недостающие либо изменившиеся блоки. Это решение также позволяет снизить нагрузку на вычислительные средства источника УР, так как сравнение признаков производит каждый клиент индивидуально. В результате сокращается объем передаваемых данных, что уменьшает время для синхронизации;

В рамках предложенных особенностей для нового механизма синхронизации необходимо определить список граничных условий применимости:

1. максимальный размер блока данных, в пределах которого возможно выделить единичное изменившееся состояние;
2. отношение максимального размера блока данных к общему размеру УР;
3. отношение размера блока данных к размеру уникального признака, характеризующего его;
4. время синхронизации при минимальном и максимальном объеме изменений за заданный период времени.

Практическую реализацию нового механизма синхронизации можно выполнить на базе peer-to-peer протокола BitTorrent [1]. Преимущества протокола [1] заключаются в его открытости и имеющемся в наличии большом количестве свободно распространяемого программного обеспечения с открытым кодом для удобства создания программных продуктов. При использовании данного протокола последовательность обмена информацией следующая [1]:

1. Источник УР создает файл метаданных, содержащий информацию об УР. В метафайл вносятся хэш-функции блоков, на которые последовательно разбиваются исходные файлы, контрольные суммы всех файлов и служебная информация.

2. Файл метаданных публикуется на сетевом ресурсе. К УР открывается общий доступ из сети при помощи приложения, работающего по протоколу BitTorrent.

3. Клиент получает файл метаданных и начинает загрузку УР при помощи специального приложения.

4. Если какие-то блоки данных УР уже присутствуют у клиента, приложение закачает только недостающие блоки.

5. По мере получения блоков данных УР клиент сам становится источником полученных частей УР: клиентское приложение продолжает загрузку недостающих блоков и открывает доступ другим клиентам по протоколу BitTorrent к уже полученным частям.

Особенностью работы протокола BitTorrent является возможность оперирования данными на уровне файла, т.е. единицей данных, для которой строится блок метаданных, является файл. В результате измененный файл (уникальный ресурс) считается заново созданным и его необходимо передать заново полностью. Устранить описанный недостаток можно, проведя дополнительную подготовку УР перед передачей по протоколу BitTorrent, а также обратные преобразования на стороне клиента.

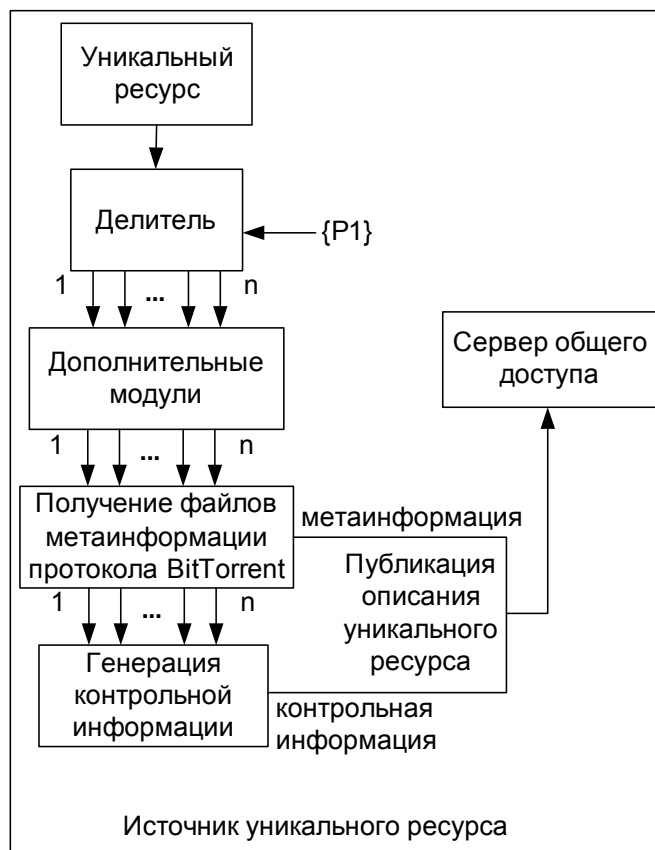


Рис. 1 Подготовка данных УР перед началом работы протокола BitTorrent.

Описание алгоритма подготовки данных:

1. файлы **УР** поступают на «**Делитель**», где происходит их линейное разбиение в соответствии с набором параметров $\{P1\}$. На выходе блока имеем n файлов заданного объема;
2. в блоке «**Дополнительные модули**» реализуются специальные функции, например шифрация данных;
3. «**Получение файлов метаданных**» для каждого из n блоков в соответствии со спецификацией протокола BitTorrent [2];
4. для полученного множества файлов описаний производится «**Генерация контрольной информации**», необходимой для проверки корректности передачи;
5. полученные файлы метаданных и контрольная информация публикуются в сети на сервере общего доступа (которым может являться сам источник УР) для получения их клиентами.

Клиенты перед началом очередной синхронизации данных производят деление, дополнительные преобразования и получение метаданных блоков имеющейся у них версии УР. После опубликования в общем доступе файлов метаданных и контрольной информации источником УР клиенты принимают их, производят проверку корректности полученных данных и начинают прием блоков данных УР по протоколу BitTorrent. В соответствии с этим закладываются изменившиеся блоки, а также те, которых у данного клиента нет. Как только все необходимые данные получены, проверяется корректность передачи, и производятся обратные преобразования блоков с последующим их мультиплексированием в файлы данных УР.

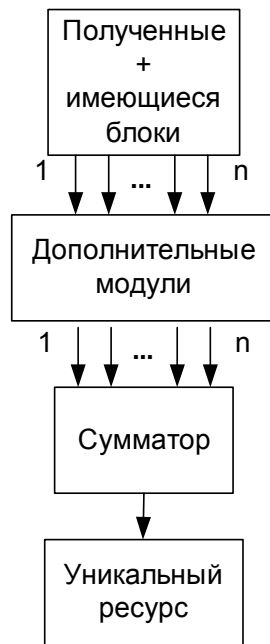


Рис. 2 Преобразование полученных блоков клиентом

Кроме описанного возможен еще «интеллектуальный» режим работы алгоритма синхронизации:

1. источник УР выделяет изменения между актуальным УР и его предыдущей версией и публикует как информацию обо всем УР, так и только о последних изменениях;

2. клиент, имеющий отметку об успешной предыдущей синхронизации, получает только информацию о последних изменениях в УР;

3. клиент, не имеющий отметки о успешной предыдущей синхронизации, оценивает возможность получения упущенных изменений в течение приемлемого времени, исходя из имеющейся ширины канала. В случае если передача невозможна, клиент оповещает об этом администратора.

Таким образом, система на основе предложенного алгоритма синхронизации имеет возможность оценки затрачиваемых ресурсов и способность к самоорганизации.

Для работы описанного механизма синхронизации предполагается, что физическое представление данных УР имеет фиксированную структуру и единичное изменение происходит в некоторых границах без общего сдвига данных. Изменения носят линейный характер, новые данные дописываются в конец файла или заменяют данные внутри файла. Эффективность в сокращении объема передаваемых данных достигается при подобранном оптимальном отношении «размер УР / размер блока / размер уникального признака». Данные отношения должны удовлетворять следующему условию:

$$V_{\text{УР}} \ll V_A \ll V_{\text{УР}},$$

где $V_{\text{УР}}$ – объем УР;

V_A – размер блока при разбиении УР;

$V_{\text{УР}}$ – размер уникального признака, характеризующего блок данных УР.

Преимущества описанного алгоритма синхронизации:

1. разгрузка каналов связи источника УР;
2. снижение объема передаваемой информации между субъектами распределенной сети;
3. независимость от содержания передаваемой информации;
4. гибкость настройки под внутреннюю структуру данных;
5. сохранение конфиденциальности передаваемой информации;
6. надежность, достигаемая за счет специальных проверок корректности передачи данных;
7. возможность расширения механизма за счет дополнительных специальных модулей.

Выводы

Идея разбиения одного «большого файла» на части для синхронизации каждой по отдельности работает лишь в случае незначительных изменений, производимых в «большом файле» за период времени между очередными синхронизациями. Таким образом, требуется передавать лишь те части «большого файла», в которых были изменены данные. При этом если каждую из частей, передаваемую при синхронизации, шифровать, то можно обеспечить дополнительную защиту при передаче данных.

Предложенный алгоритм хорошо подходит для синхронизации файлов баз данных, работающих по принципу «добавления измененных записей в конец файла» и позволяет упростить построение распределенных сетей с синхронизацией данных.

Литература

1. BitTorrent protocol specification // BitTorrent.org. – 2006. – Режим доступа: <http://bittorrent.org/protocol.html>, свободный. – Загл. с экрана.
2. BitTorrent Specification // Creative Commons. – 2006. – Режим доступа: <http://wiki.theory.org/BitTorrent-Specification>, свободный. – Загл. с экрана.