

© 2012 г. **В.Л. ОРЛОВ, канд. техн. наук,**
 Е.А. КУРАКО
 (Институт проблем управления
 им. В.А. Трапезникова РАН, Москва)

ЗАДАЧИ УПРАВЛЕНИЯ ПРОЦЕССОМ ОБНОВЛЕНИЯ МНОГОКОМПОНЕНТНЫХ СИСТЕМ В УДАЛЕННЫХ ПОДРАЗ- ДЕЛЕНИЯХ ОРГАНИЗАЦИИ

В данной работе рассматривается обновление ПО в распределенных информационных системах. Анализируются современные технологии обновления ПО. Предлагается модифицированный метод каскадного обновления.

TASKS OF UPDATING PROCESS CONTROL OF MULTICOMPONENT SYSTEMS IN REMOTE PARTS OF ORGANIZATION / V.L. Orlov, E.A. Kurako (V.A. Trapeznikov Institute of Control Sciences, Profsoyuznaya 65, Moscow 117342, Russia, E-mail: kea@ipu.ru). The present work considers software updating process is distributed informational system. A brief analysis of modern methods of software updating process is performed. The modified method of cascade updating of the software is offered.

1. Введение

На современном этапе развития информационных технологий происходит интеграция разнородных информационных подсистем, которые территориально распределены. Если территориальная распределенность велика, то, как правило, в каждом подразделении имеется свое программное обеспечение (ПО) и своя база данных (БД). В процессе интеграции стандартизируется макет БД, справочники и ПО. Во время эксплуатации информационных подсистем БД заполняются данными, которые в каждом подразделении свои. При этом параллельно происходит изменение внешних условий, например, изменяется законодательство, изменяется название абонентов, преобразуется среда передачи данных или обнаруживаются ошибки в коде программ. Таким образом, возникает задача обновления ПО и данных в распределенной системе организации и ее удаленных подразделениях.

Стандартные механизмы синхронизации и репликации баз данных здесь неприменимы из-за того, что хранимая информация различается в каждом подразделении, но обновлять общую часть, справочники, необходимо. Очевидно, что существуют справочники, которые имеют общую и частную информацию. Например, справочник абонентов может содержать как абонентов общих для всех, так и абонентов, которые интересны только данному подразделению. Ввиду того, что если простым объединением получить общий справочник для всех, то работать с таким справочником будет затруднительно из-за огромного количества излишних данных.

Вторым препятствием на пути стандартных средств обновления является низкая пропускная способность каналов передачи данных, которая часто идет с блокировкой всех протоколов, кроме SMTP/POP3, HTTP и специализированных протоколов.

Дополнительно следует учесть высокую связность ПО и данных. То есть очень часто для работы новой версии программы следует изменить макет БД или изменить несколько записей в справочниках, причем в определенном порядке.

Кроме того, следует учесть, что в удаленных подразделениях не предусмотрена должность системного администратора, а персонал обладает ограниченным набором знаний в данной области.

Из вышеперечисленного можно выделить условия, с учетом которых, требуется решить задачу обновления программного обеспечения:

- 1) Задача должна быть решена для случая распределенной информационной системы, которая может быть построена на открытых сетях;
- 2) Система обновления должна выдерживать высокие нагрузки;
- 3) При доставке обновлений должны использоваться протоколы HTTP/HTTPS или SMTP/POP3(или надстройки над данными протоколами);
- 4) Должна присутствовать возможность обновления данных, как простого обновления таблицы, так и выборочного набора, отфильтрованного по заданным критериям;
- 5) При обновлении должен применяться пакет изменений, который либо применяется целиком, либо может откатиться с любого шага до исходной точки;
- 6) Подсистема обновления должна работать в автоматизированном режиме.

2. Обзор существующих решений

Авторами были рассмотрены различные подсистемы обновления, такие как подсистем обновления Windows, Linux и других отдельных продуктов. Все исследуемые подсистемы обновлений постепенно переводились на автоматизированный режим. Например, в Windows Vista, Windows Server 2008 и Windows 7 больше невозможно использовать обновления через веб-сайт. Вместо этого теперь используются автоматические обновления, которые имеют более широкий функционал[1].

При обновлении ПО Linux используется механизм репозитория[2]. Данный механизм направлен на децентрализацию хранения обновлений и таким образом увеличения скорости передачи обновлений клиенту. Так как клиент может выбрать ближайший к себе репозиторий с хорошей скоростью загрузки. Ввиду огромного количества прикладного ПО для Linux, репозитории создавались как независимые хранилища. То есть набор обновляемого ПО репозитория А может кардинально отличаться от набора обновляемого ПО репозитория В.

Если рассматривать обновления отдельного программного продукта, например, Антивирус Касперского 2012[3], то можно заметить, что алгоритм обновления является автоматическим, с возможностью переключения на ручной режим. При этом существует возможность реализовать процесс ретрансляции антивирусных баз и программных модулей в локальный источник (папку на диске, сетевой ресурс). Данный механизм предназначен для обеспечения полноценного обновления при существующем разделении локальных сетей или для разгрузки канала из внутренней сети предприятия во внешний мир.

3. Метод каскадного обновления

Авторами предлагается метод каскадного обновления ПО[3], который предполагает, что все пользователи разделяются на условные подгруппы, каскады. Условием объединения некоторых пользователей распределенной системы в каскад, может являться территориальная близость или уже существующее разделение пользователей. В каскаде выделяется компьютер, называемый сервером каскада, который скачивает обновления, а все остальные пользователи каскада скачивают обновления уже с данного компьютера. Таким образом, можно выстроить цепочку каскадных групп пользователей (Рис. 1.). Каскад нулевого уровня - это центральный узел обновления ПО. Каскад уровня N+1 - это группа пользователей сервера каскада, который скачивает обновления с сервера каскада N. При этом следует учесть, что при отказе сервера каскада N+1, его пользователи будут обращаться на сервер каскада N, что повышает отказоустойчивость системы обновления.

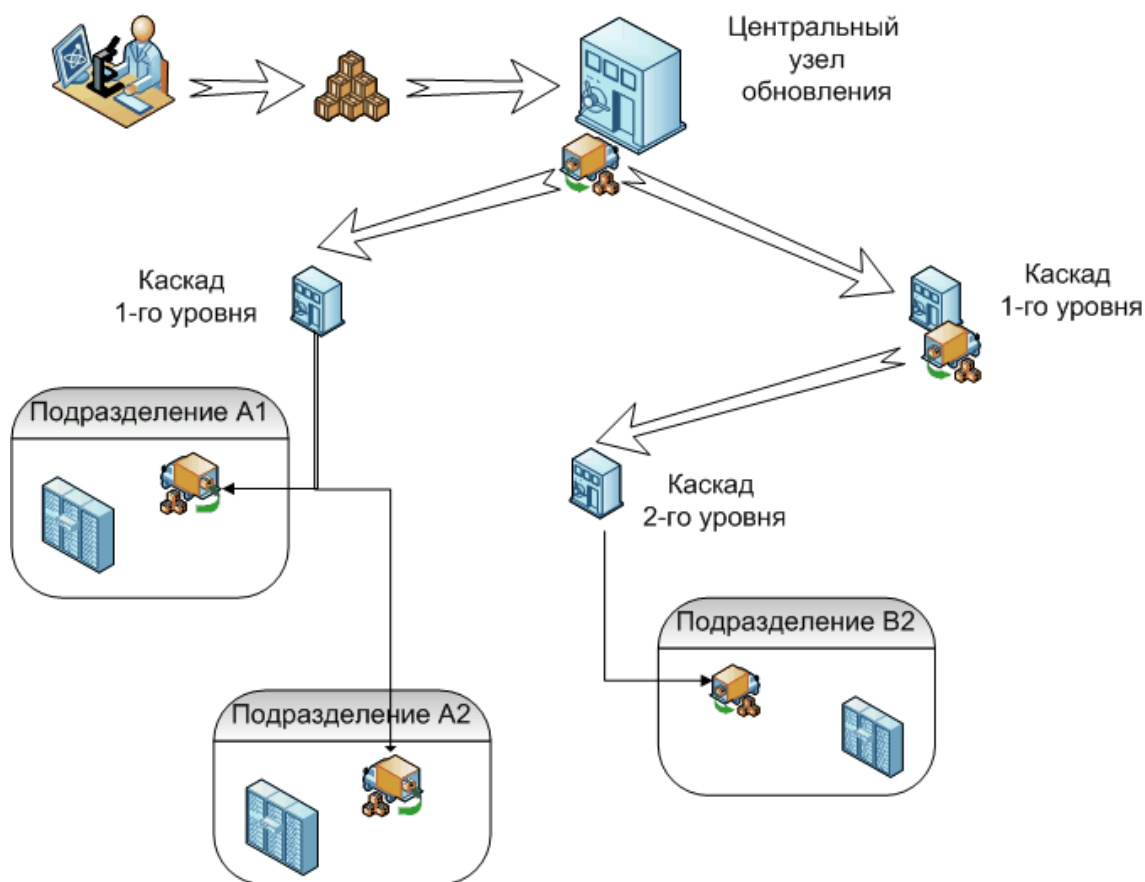


Рисунок 1. Каскадное обновление

Сервер каскада представляет собой web-сервер, который в автоматическом режиме опрашивает каскад предыдущего уровня. Доступ к обновлениям расположенным на данном каскаде происходит по протоколу SOAP.

Данный метод позволяет функционировать подсистеме обновлений в распределенной системе, где пользователь может быть закрыт от сервера приложений несколькими брандмауэрами и прямой доступ к файлам запрещен. Благодаря построенной системе каскадов обновления идут из одного источника, при этом нагрузка на центральный узел обновления ПО снижается, позволяя избежать отказов в обслуживании.

4. Модификация метода каскадного обновления

Чтобы учесть необходимость модификации отдельной части данных и структуры БД, вводится понятие типа обновления. Тип обновления определяет для клиентской стороны алгоритм, позволяющий определенным образом обработать полученное обновление. В результате по типу обновления определяется соответствующий обработчик программ или данных. С помощью выполняемых скриптов можно тонко донастраивать структуры БД или выборочно менять связанные данные.

Чтобы учесть порядок выполнения и единовременность нескольких обновлений, такие обновления объединяются в группу обновлений, называемую пакетом обновления.

Таким образом, разработчик или системный администратор подготавливает пакеты обновлений, выкладывает их на центральном узле (или каскаде 0-го уровня), а дальше обновления расходятся по каскадам. После появления нового обновления на своем каскаде информационная система подразделения, скачивает нужные пакеты, распаковывает и автоматически выполняет полученные обновления.

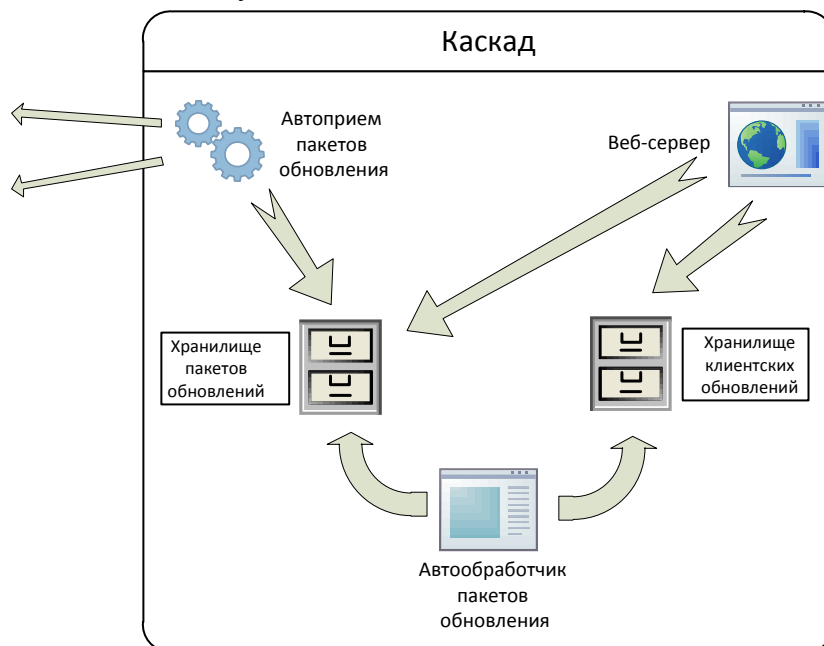


Рисунок 2. Схема сервера каскада с управляемым обновлением

На рис. 2 представлена схема сервера каскада с управляемым обновлением.

Подсистема автоматического приема пакетов обновления обеспечивает доставку пакетов на сервер каскада из различных источников. Как правило, им является веб-сервер предыдущего каскада или в случае организованного физического разрыва между сетями CD-диск или флешка с пакетами обновлений. После проверки пакета обновления на отсутствие ошибок, полученных при передаче, пакет обновления заносится в специальное хранилище.

Подсистема автоматической обработки пакетов обновления при появлении нового пакета в хранилище запускает процесс обработки:

- распаковывается пакет обновления;
- по шагам выполняется каждое обновление, в зависимости от его типа;

- в случае ошибки применения обновления на любом шаге, происходит восстановление исходного состояния системы до начала применения данного пакета обновления.

В случае обновлений для клиентской части программы автообработчик складывает полученные обновления в хранилище клиентских обновлений механизма обновления на основе динамически обновляемых модулей[5].

Подсистема распространения пакетов обновления и обновления клиентской части состоит из веб-сервера, который имеет доступ к соответствующим хранилищам.

5. Выводы

Данный метод позволяет решить задачу обновления ПО в распределенной информационной системе организации с выполнением заданных условий.

К достоинствам полученного решения стоит отнести:

- масштабируемость системы обновлений, например, при добавлении нового подразделения необходимо добавить сервер каскада или подключиться к ближайшему;
- возможность обновления по протоколу широко распространенному HTTP или с помощью переносного носителя информации, при наличии физического разрыва сети;
- снижение нагрузки на центральный узел информационной системы;
- автоматическая точечная синхронизация общей части баз данных каскадов;
- единовременное обновление связанных частей подсистемы;
- повышение управляемостью обновлений и снижение нагрузки на системных администраторов, повышение оперативности и надежности обновления ПО и справочников в подразделениях.

СПИСОК ЛИТЕРАТУРЫ

1. Центр обновления Windows // http://ru.wikipedia.org/wiki/%D6%E5%ED%F2%F0_%EE%E1%ED%EE%E2%EB%E5%ED%E8%FF_Windows.
2. Репозитории, обновление // <http://www.ubuntologia.ru/repositories>.
3. Настройка Обновления // <http://support.kaspersky.ru/kis2012/settings#!kis2012/settings/update>.
4. Курако Е.А., Лебедев В.Н., Орлов В.Л. Метод каскадного обновления программного обеспечения в распределенных крупномасштабных системах - «Управление развитием крупномасштабных систем MLSD'2011», Том II, М. ИПУ РАН 2011 г., С. 260-262.
5. Лебедев В.Н., Курако Е.А., Москальков В.Е., Орлов В.Л. Построение и развитие распределенных систем на основе динамически обновляемых модулей // «Управление развитием крупномасштабных систем MLSD'2008» Том II, М. ИПУ РАН 2008 г., С. 114-115.