

повреждений в оборудовании. Значение, большее нуля, позволяет говорить о том, что присутствуют некие отклонения в вибрации, следовательно, необходимо более пристальное внимание к вибрирующему компоненту.

Список литературы:

1. de Lorenzo F. Kurtosis: A Statistical Approach to Identify Defect in Rolling Bearings / F. de Lorenzo, M. Calabro // Proceedings of 2nd International Conference on Marine Research and Transportation, Naples, – 2007. – P. 17 - 24.
2. Sawalhi N. The application of spectral kurtosis to bearing diagnostics / N. Sawalhi, R. Randall // School of Mechanical and Manufacturing Engineering, The University of New South Wales, Gold Coast, Australia, – 2004. – P. 393 - 397.
3. Vibration monitoring of rolling element bearings [Электронный ресурс]: Technical Note / 01db metradiv technologies. – Электронные данные. – Режим доступа: bearing_vibration_monitoring.pdf.

АЛГОРИТМ ПРЕДОСТАВЛЕНИЯ АВТОНОМНОГО ДОСТУПА К НОВОСТНЫМ РЕСУРСАМ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Зезюлина Ю.М.

Серебряная Л.В. – к.т.н., доцент

С развитием информационных технологий, стремительным темпом роста охвата пространства беспроводным интернетом и мобильными технологиями резко увеличилось количество интернет-ориентированных потребителей новостной информации. Однако при отсутствии Интернета пользователь теряет возможность доступа к интернетисточникам. Решением данной проблемы будет система, которая может работать в автономном режиме. В данной работе описан алгоритм реализации системы, осуществляющей доступ к новостным ресурсам в офлайн-режиме.

До недавнего времени не было ни одного практического способа реализовать подобную систему с автономным доступом к новостным ресурсам, однако с развитием HTML5 технологии появилась возможность кешировать данные, что позволило работать веб-приложению в режиме офлайн так же, как и в онлайн.

HTML5 предоставляет несколько видов автономных технологий хранения данных:

1. Веб-хранилище, в котором реализовано хранение объектов в виде ключ-значение, а именно использование объектов *localStorage* и *sessionstorage*.

2. Веб-база данных SQL поддерживает все возможности и функции структурированной реляционной базы данных SQL, но в настоящий момент эта технология уже устарела из-за низкой производительности.

3. Индексированная база данных – технология, включающая преимущества веб-хранилища и веб-базы данных SQL. Как и в веб-хранилище в ней напрямую сопоставляются ключи и значения, однако в то же время имеется поддержка индексов, как в реляционных базах данных. Это позволяет осуществлять быстрый поиск объектов, соответствующих конкретному полю: перебор всех объектов в хранилище происходит автоматически.

Использование данных технологий в разработке системы позволит:

1. Организовать работу приложения в автономном режиме с последующей обратной синхронизацией данных после повторного подключения к сети.

2. Повысить эффективность работы с приложением: при возврате на сайт пользователю не нужно ждать повторной загрузки большого набора данных – их можно показать сразу.

3. Благодаря отсутствию серверной инфраструктуры упрощается программирование.

Перечисленные технологии, позволяющие хранить данные на клиенте, появились недавно, поэтому они еще не успели найти широкое применение. Кроме того, владельцам новостных ресурсов необходимы дополнительные затраты времени и средств для внедрения этих технологий. Поэтому большинство владельцев новостных ресурсов не используют возможности автономных технологий. Решением названной проблемы может быть использование системой агрегаторов новостных лент. Рассмотрим работу системы подробнее.

Изначально система периодически опрашивает добавленные в список источников новостные серверы, загружая их RSS-ленты и проверяя наличие свежих, еще не обработанных статей. Если такие статьи появились с момента прошлого обхода сканера, происходит загрузка статьи с сайта издания в базу сервера. Параллельно этому идет загрузка непрочитанных статей в клиентскую базу данных пользователя, а также выгрузка из данной базы устаревших статей. Таким образом, при обращении пользователя на веб-сервер, будут представлены результаты из локальной памяти, что значительно увеличит скорость загрузки страниц. При выборе автономной технологии следует учитывать, что полноценные сохранения могут быть очень массивными, так как сохраняется большое количество новостей, что может быть более тысячи килобайт. WebStorage имеет ограничение около пяти мегабайт и варьируется в зависимости от браузера.

Возможно, этого будет недостаточно для сохранения новых статей. Поэтому было решено использовать новую систему IndexedDB для хранения данных. Эта база поддерживает большие объемы информации, обычно около 50 мегабайт.

Таким образом, был разработан алгоритм реализации системы, осуществляющей автономный доступ к новостным ресурсам, который основан на использовании HTML5 технологии хранения данных на клиенте IndexedDB, а также RSS-лент целевых новостных ресурсов.

Список использованных источников:

1. pixelcom [Электронный ресурс]. – 2014. – Режим доступа <http://www.pixelcom.crimea.ua/html5-appcache-indexeddb.html> Дата доступа: 02.01.14
2. MSDN [Электронный ресурс]. – 2014. – Режим доступа [http://msdn.microsoft.com/ru/library/jj154905\(v=vs.85\).aspx](http://msdn.microsoft.com/ru/library/jj154905(v=vs.85).aspx) Дата доступа: 02.01.14

ПРОГРАММНОЕ СРЕДСТВО ГАРАНТИРОВАННОЙ ДОСТАВКИ СООБЩЕНИЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Белов А.А.

Фадеева Е. Е. – ассистент

Одним из ключевых особенностей крупных коммерческих продуктов, является наличие модуля интеграции с другими системами. Таким образом, разработанная система может обмениваться сообщениями с другими системами посредством каналов передачи данных. Причём различные системы могут иметь несовместимые протоколы передачи данных, а также быть недоступными в какой-либо момент времени. Решением данных проблем возлагается на модуль гарантированной доставки сообщений.

Применение подобного модуля упрощает решение задачи интеграции конечных систем. Модуль поддерживает большинство протоколов взаимодействия и прозрачно для конечных систем выполняет преобразование конвертов и форматов сообщений, обеспечивая совместимость а также единую точку интеграции. До введения подобного модуля, конечные системы были связаны по принципу «каждый с каждым», что значительно усложняло архитектуру системы, усложняло задачу интеграции (различные системы могут использовать множество протоколов) – рис.1. После введения системы взаимодействуют с ним по известным им протоколам, возлагают на него ответственность за передачу сообщений и обработку ошибочных ситуаций – рис. 2.

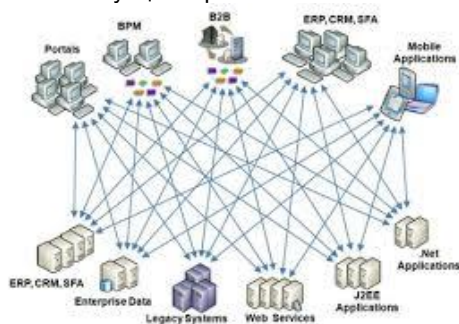


Рис. 1 – Связь «каждый с каждым» при интеграции систем

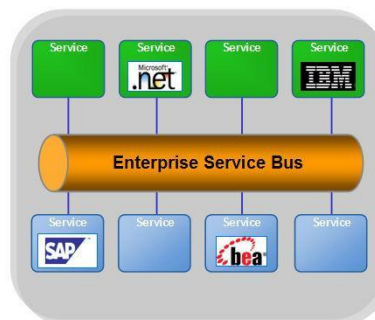


Рис. 2 – Топология после добавления модуля интеграции

Ключевой особенностью разрабатываемого модуля является гарантированная доставка сообщений. Это значит, что конечная система 1 (КС1) отправляет сообщение, с указанием тела сообщения и получателя, получает подтверждение о том, что сообщение принято в обработку и далее за доставку конечной системе 2 (КС2) отвечает модуль интеграции. Сам модуль с интервалом времени пытается доставить сообщение КС2, и если количество попыток (оно настраивается) превысило предельно допустимое, сообщение попадает в специальную очередь, и в работу включается администратор, который может просмотреть эту очередь или повторно запустить механизм отправки.

На протяжении отправки сообщения из КС1 в КС2 сообщению присваиваются определённые статусы – ключевыми из которых являются: «Получено из КС1», «Принято в обработку», «Отправлено в КС2». Таким образом, возможно отследить статус сообщения во времени и быстрее обнаружить место потенциальной ошибки в процессе обработки. Для администратора доступна консоль, с помощью которой он может отследить каждое сообщение по его идентификатору.

Выше описан механизм асинхронной доставки сообщений. В системе также предусмотрена возможность синхронной отправки сообщений. В этом случае КС1 отправляет сообщение, затем