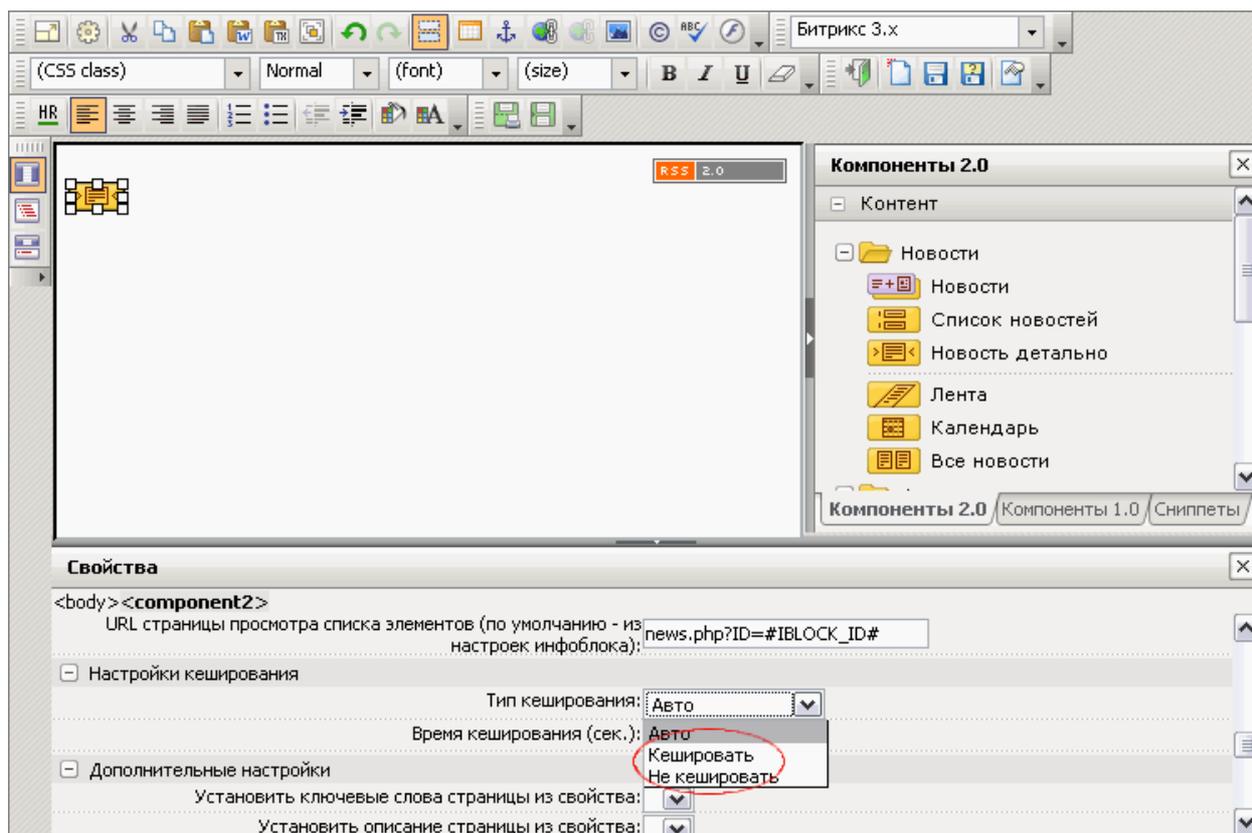


Настройка кеширования

Суть автокеширования заключается в том, что все динамические компоненты, которые используются для создания веб-страниц получили **встроенную поддержку управления кешированием**. Для использования новой технологии нетехническому специалисту достаточно **включить** автокеширование **одной кнопкой** на административной панели. При этом все компоненты, у которых был включен режим автокеширования создадут кеш и полностью перейдут в режим работы **без запросов к базе данных**.



Настройки автокеширования компонента

Евгений Новиков,
руководитель отдела разработок компании Articul Media



«Чем отличается правильно от неправильно? Нет, не только частицей «не». Дело в том, что когда правильно, то наступает гармония и все работает. А вот когда неправильно, наступает что-то другое и, соответственно – не работает ничего.

Большинство проектов, разрабатываемых компанией Articul Media, имеют достаточно сложную архитектуру и высокую посещаемость. Конечно же, под эти задачи мы оптимизируем запросы и настраиваем серверное окружение. Однако, этого недостаточно. На всех своих проектах мы используем кеширование, сокращая запросы к базе на страницах до необходимого минимума. Логика управления кешем, реализованная в «1С-Битрикс: Управление сайтом», удобна в использовании и прозрачна для разработчиков и администраторов сайта. Затраты на реализацию кеширования динамического контента минимальны, а результат, достигаемый использованием кеширования, многократно повышает производительность работы наших сайтов.

И используем кеширование мы не потому, что это модно, а потому что это правильно и гармонично».

Управляемое кеширование

Термин "управляемое кеширование" объясняет суть самого механизма. Кешированные данные обновляются незаметно для пользователя по мере обновления первичных данных, т.е. управляются самими данными.

Управляемое кеширование является одной из наиболее эффективных и незаметных для пользователя технологий масштабирования.

Технология управляемого кеширования используется в основном в ядре продукта, на самом нижнем уровне и позволяет автоматически кешировать наиболее часто используемую информацию из таблиц базы данных. При изменении данных в таблицах **кеш автоматически чистится и перестраивается**, что происходит незаметно для пользователя и для разработчика. Результаты кеширования сохраняются в виде файлов в каталоге управляемого кеша `/bitrix/managed_cache/`.

Управляемое кеширование реализовано для большинства объектов ядра. Результатом использования этой технологии является **существенная минимизация числа запросов к базе данных** на веб-страницах.

Редакции продукта «Старт», «Стандарт» и «Малый бизнес» могут работать, не открывая соединение с базой данных и не выполняя SQL-запросы. Запрос к базе данных произойдет только в случае изменений данных в ней.

Управляемое кеширование также **очень эффективно** для проектов, размещенных **на виртуальном хостинге**, поскольку встроенные механизмы кеширования SQL-запросов базы данных работают обычно крайне неэффективно в силу высокой вариативности данных большого числа пользователей.

Неуправляемое кеширование

Фактически это возможность разработчикам задать правила кеширования ресурсоемких частей страниц. Результаты кеширования сохраняются в виде файлов в каталоге `/bitrix/cache/`. Если время кеширования не истекло, то вместо ресурсоемкого кода будет подключен предварительно созданный файл кеша.

Кеширование называется неуправляемым, поскольку кеш не перестраивается автоматически после модификации исходных данных, а действует указанное время после создания. Часто такая схема является оптимальной.

LRU-кеширование

Last Recently Used кеш - это кеш базы данных в традиционных терминах. Мы реализовали **эмуляцию** данного кеша в продукте. Цель появления такого кеша такова: некоторые данные могут иметь теоретически неограниченно большой размер, поэтому их полное кеширование не представляется возможным. Но в реальной жизни и на реальном сайте эти данные могут быть весьма ограничены и/или какие-то из этих данных будут более востребованы, чем остальные. Причем некоторые данные очень часто запрашиваются на сайте.

Мы посчитали, что имеет смысл кешировать эти данные. Но так, чтобы кеш не оказался многомегабайтной длины (что очевидно не только не ускорит, но и даже затормозит работу). LRU кеш является управляемым и имеет управляемый размер (настраивается для каждого кешируемого объекта).

Если происходит запрос к кешируемому объекту, то объект сначала проверяется в кеше. Если там уже есть необходимые данные, то они передвигаются в начало кеша и возвращаются без запросов и вычислений. Если данных в кеше нет, то они выбираются (и вычисляются) и добавляются в начало кеша. Если при этом кеш перерос ограничения по длине, то последний элемент из него вытесняется.

Таким образом, кеш ограничен по размеру на случай очень большого числа данных в базе. Если

данных окажется реально немного, то они будут полностью кешированы. Если данных много, но одни из них запрашиваются чаще других, то высока вероятность, что часто запрашиваемые данные будут постоянно жить в кеше (так как из кеша при переполнении удаляются те данные, которые дольше всего не запрашивались).

Первоначально этот тип кеширования будет применен к некоторым объектам в каталоге. Например, к скидкам.

Но для данных, которых заведомо мало, лучше использовать управляемый кеш. В частности, в новом модуле валют вообще не будет возникать запросов к базе данных при использовании большинства функций. Управляемый кеш сейчас будет применяться и в инфоблоках для обработки наиболее частных операций.

Пример использования LRU кеша

К классу `CStackCacheManager`. Объект `stackCacheManager` создается автоматически. Можно установить длину кеша методом `SetLength(entity, length)`, где `entity` - некий код сущности (например, `catalog_discount` для скидок), а `length` - длина. Можно проверить существование записи для данной сущности методом `Exist(entity, key)`, где `entity` - некий код сущности, а `key` - код записи. Если запись с кодом `key` у сущности `entity` существует, то ее можно получить методом `Get(entity, key)`. При этом запись в кеше перемещается на первое место. Если записи нет, то ее можно добавить методом `Set(entity, key, value)`. При этом запись ставится в кеше на первое место, а последняя запись убирается, если кеш перерос свою длину. Методом `Clear(entity)` можно очистить все записи данной сущности.

Кеширование графических файлов

Начиная с версии главного модуля 5.1.4 в дополнение к имеющимся механизмам управляемого кеша добавился механизм **кеширования графических файлов из библиотеки изображений** (таблица `b_file`).

Рекомендуется включать этот механизм на **виртуальных хостингах** для **уменьшения числа SQL-запросов** к удаленной или медленной базе данных. На выделенных машинах этот тип кеширования заметного результата может не дать. Рационально использовать такой кеш, если число файлов не очень большое (не больше 10 тысяч) или на странице выводится одновременно много элементов.

Для этого типа информации использована интересная методика кеширования с разделением хранилища на секции. Это позволяет уменьшить число файлов и сделать кеш эффективней. Причем данные кешируются **с опережением**, что даже при включении позволяет уменьшить число запросов. Но отметим, что по умолчанию этот механизм выключен, так как нет пока возможности определять динамически число файлов в `b_files` во время работы, а у проектов с большим числом файлов возможно создание объемного кеша. Но мы рекомендуем попробовать включить кеширование на виртуальном хостинге и индивидуально для своего проекта определить применимость данного инструмента.

Кеширование меню

Применяется режим **кеширования права доступа и части проверок для меню** для неавторизованных пользователей. Учитывая, что большая часть посетителей - это незарегистрированные пользователи, к ним применен специальный алгоритм кеширования. Кеш управляемый и обновляется при редактировании меню или изменении прав доступа к файлам и папкам через административный интерфейс и API. Т.е. обычные пользователи вообще не заметят, что существует какой-то алгоритм кеширования. Если только разработчики правят меню вручную, то им потребуется скинуть кеш меню в интерфейс.

HTML кеширование

Новый тип кеширования, получивший название "HTML кеш", доступен во всех редакциях продукта начиная с версии 6.5.8 главного модуля. Располагается он на вкладке "Настройки"- "Настройки продукта"- "Автокеширование"- "HTML кеш"

Попаданий в кеш	Пропусков мимо кеша	Очистка кеша из-за ограничений дискового пространства	Очистка кеша вызванных модификациями данных
7	10	0	4

Механизм HTML-кеширования лучше всего включить на какой-нибудь редко изменяющийся раздел с регулярным посещением анонимных посетителей, так как при включенном HTML-кешировании происходят следующие процессы:

- ♦ механизмом HTML-кеша обрабатываются только страницы, не указанные в маске исключения и указанные в маске включения;
- ♦ если на такие страницы заходит не авторизованный пользователь, то выполняется проверка существования файла кеша и если таковой найден, то выдается страница из кеша, не задействуя никакие модули продукта; например, не будет работать

Настройки HTML кеширования

За счет применения HTML кеширования обеспечивается серьезный ресурсный рост для редко изменяющихся сайтов или страниц. Например, **тестовый сайт обработал 70 запросов в секунду, а с HTML кеш через NGINX уже 1200-1600 страниц в секунду**. Безусловно, HTML кеширование станет технологией повседневного использования для очень многих небольших проектов. Технология проста в эксплуатации, не требует от пользователя отслеживать изменения, защищает дисковой квотой от накрутки данных и автоматически восстанавливает работоспособность при превышении квоты или изменении данных. Как только пользователь авторизуется в продукте, он уже работает с сайтом без кеширования, как с обычным продуктом. Технология HTML кеширования работает в автоматическом режиме AJAX с компонентами 2.0.

«1С-Битрикс»

«1С-Битрикс» - совместное предприятие, созданное фирмой «1С» и ООО «Битрикс». «1С-Битрикс» занимается продажей, развитием и продвижением продукта «1С-Битрикс: Управление сайтом» на территории стран бывшего СССР, а также построением партнерской сети для его распространения, внедрения и поддержки.

На основе продукта работает более 15000 веб-проектов: сайты государственных и правительственных структур, крупных промышленных предприятий, образовательных учреждений, СМИ, разработчиков программного обеспечения, некоммерческих организаций.

«1С-Битрикс» является технологическим партнером для дилерской сети, включающей более 2500 компаний, и предоставляет возможность дизайн-студиям и независимым разработчикам использовать продукты компании для реализации своих решений.

Тел./факс: +7 (495) 775-26-18
Тел.: +7 (495) 363-37-53

E-mail: sales@1c-bitrix.ru
Сайт: <http://www.1c-bitrix.ru>