Оптимизация сайта на Joomla 1.5+ - кэширование

Когда **нагрузка проекта** начинает переваливать за десятки тысяч просмотров, начинает дико расти объем потребляемой памяти и процессорное время. А если сайт на хостинге с ограничением скажем в 128Мб оперативки. При таком раскладе сайт может отвалиться уже при меньшем числе просмотров в сутки. Конечно это не дело – крутиться на таком железе проекту с такой посещаемостью. Но… «Если не видно разницы, зачем платить больше!» Так вот и надо сделать так, чтоб разницы было не видно.

Обычно **Joomla** выбирают для не очень **нагруженных систем** из-за ее простоты и популярности. Сразу или постепенно сайт начинает дорабатываться: пишутся свои компоненты, естественно, разрабатывается дизайн, собственная часть бд и т. д. Короче говоря, расширяется функционал. База данных усложняется, таблицы растут (порой до нескольких миллионов записей), запросы становятся тяжелее и не всегда оптимальны.

Конечно первое, что надо делать, это **оптимизировать все тяжелые sql-запросы** и ин дексы таблиц
, но это отдельная тема.

Зачастую, собственные расширения, например компоненты, пишутся не дальновидно и **в озможность кэширования**
в них не закладывается. Многие полагают, что
включив кэширование в глобальных настройках сайта
, они обеспечили
кэширование
всего и вся. Я тоже так думал, начав работать с Joomla.

Лирика

Для реализации **механизма кэширования**, разработчики Joomla используют бесплатную библиотеку Cache_Lite. В качестве менеджера кэша выступает класс JCache_Lite_Function. Данный класс содержит метод call, позволяющий вызвать

произвольную функцию по имени и передать ей на вход указанные параметры. При вызове этого метода, в качестве первого параметра указывается имя функции, а далее — произвольное количество переменных, являющихся параметрами этой функции.

При вызове метода call, происходит поиск **файла кэша** для этой функции и заданного набора параметров. Если такой файл найден, и время жизни этого **закэшированного объекта** не истекло, то вместо реального вызова функции, происходит возврат данных из файла. Если файл не найден, то это означает, что данная функция с такими параметрами еще не вызывалась. В этом случае происходит вызов функции, с передачей ей на вход указанных параметров, а результат работы функции помещает в кэш.

Пороемся в коде

Наверняка в Вашем компоненте существует функция или метод класса, который обращается к базе данных и вытаскивает массив или объект страшных размеров для дальнейшей обработки. Мало того, что сам sql-запрос порой занимает порядка секунды (ужас), так еще и сам результат может содержать десятки и мотни тысяч элементов. А если одновременно эту функцию или метод класса запустят сотни, да что там десятки пользователей. Сколько памяти придется выделить интерпретатору php, mysql-у и т. д. Короче, в лучшем случае, это жуткие тормоза, в худшем, недоступность сайта или блокировка его хостером со всеми вытекающими.

При всем при этом, результат выполнения функции или метода класса будет каждый раз одинаков для одинаковых параметров, передаваемых этому методу или функции. Чтобы избежать такого дикого и бесполезного расхода ресурсов сервера, мы будем кэшировать результат, возвращаемый методом или функцией.

Для начала, следует получить ссылку на экземпляр класса, который и будет заниматься **кэшированием**
.

```
$cache =& JFactory::getCache();
```

Объект класса уже «в курсе» глобальных настроек и того, включено ли кэширование на сайте или нет, а также **длительности хранения кэша**. Зачастую для главной страницы лучше поставить иную длительность хранения кэша, нежели для остальных страниц. Если вы хотите, чтобы ваш компонент использовал кэширование, даже если оно отключено в глобальной конфигурации, или, например,
изменить длительность хранения кэша
, Вы можете настроить его сами:

```
$cache->setCaching( 1 ); //включаем кэширование независимо от настроек
$cache->setCaching( 0 ); //выключаем кэширование независимо от настроек
$cache->setLifeTime(300); //устанавливаем время жизни кэша 5 минут
$cache->setLifeTime(1440); //устанавливаем время жизни кэша сутки
```

Подробнее о методах класса тут:
http://api.joomla.org/Joomla-Framework/Cache/JCache.html

Далее вызовем наш метод через объект кэширования:

```
$rows = $cache->call( array( 'TestClass', 'testMethod' ) );
```

Если экземпляр класса создан, можно передать не имя класса, а ссылку на экземпляр.

```
$ob = new TestClass(); $rows = $cache->call( array( $ob, 'testMethod' ) );
```

Если мы хотим вызвать **кэширование конкретной функции**, то не нужно использовать массив:

```
$rows  = $cache->call( 'testFunction' );
```

Если наш метод или функция принимает параметры, передаем из следом:

```
$rows = $cache->call( array( 'TestClass', 'testMethod' ), $par1, $par2, $par3 );
$rows = $cache->call( array( $ob, 'testMethod' ) , $par1, $par2, $par3 );
$rows  = $cache->call( 'testFunction', $par1, $par2, $par3  );
```

Во время тестирования кода, можно очистить кэш:

```
$cache->clean();
```

Теперь при вызове функции или метода единожды, его результат сохранится в файл и позже, в течение всего **срока жизни кэша**, будет загружаться оттуда, не насилуя при этом сервер. По истечении срока жизни файла, функция вновь вызовется единожды и перезапишет **устаревший файл кэша**.

О времени жизни кэша

Преимущества длительного кэширования

По умолчанию, **время жизни кэша** равно 900 секунд или 15 минут. В этом случае, даже при отсутствии изменений на сайте, каждые 15 минут кэш будет перезаписываться вновь. **Увеличение времени жизни кэша** позволяет дольше пользоваться закэшированными данными, снижая общее количество запросов к базе данных.

Оптимальным временем жизни кэша является интервал обновления сайта или конкретной страницы (вот нам настройка времени ручками и пригодилась). Если вы добавляете материал раз в сутки, то время жизни кэша можно ставить равное 86400 секундам (24 часа). Для главной страницы сайта, где часто выводятся новые сообщения с форума или заголовки свежих новостей, время жизни может быть равно 600, 300, а то и 60 секундам, для крупных новостных порталов, где важна каждая минута.

Недостатки длительного кэширования

Если время жизни кэша достаточно велико, то различные модули, например выводящие списки новых материалов при включенном кэшировании могут немного «врать», поскольку **кэш модулей** сбрасывается только при изменении параметров публикации модуля или по истечению времени жизни кэша. Поэтому для сайтов использующих модули, выводящих часто обновляемую информацию (о чем говорилось выше), время жизни кэша лучше устанавливать примерно 80-90% от среднестатистического времени добавления новостей или же вообще **отключать кэширование** в этих модулях, при жесткой оптимизации кода и запросов.

Выводы

- Кэширование помогает снизить **нагрузку на СУБД** и благоприятно сказывается на **скорости загрузки страниц**.
- При **выборе времени жизни кэша** основным фактором является периодичность обновления материалов на сайте.
- **Включение механизма кэширования** в глобальных настройках сайта приведет к кэшированию материалов сайта, а для кэширования модулей сайта необходимо зайти в редактирование параметров этих модулей и выставить параметру

Enable cache

значение «Да».