

О. А. Ковалюк, к. т. н., доц.

АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ КЭШИРОВАНИЯ ДАННЫХ В ВЫСОКОНАГРУЖЕННЫХ СИСТЕМАХ

В статье исследовано программное обеспечение кэширования данных для оптимизации работы высоконагруженных веб-систем и обоснован выбор средства кэширования в зависимости от объема данных.

Ключевые слова: кэширование, высоконагруженные системы, веб-сайты, оптимизация.

Введение

Сегодня значительное количество компьютерных систем создают в виде веб-приложений, когда клиентом выступает веб-браузер, взаимодействующий с веб-сервером. Такую архитектуру реализуют многие компьютерные системы управления, системы управления бизнес-процессами, веб-сайты. Главным преимуществом данного подхода является использование «тонкого клиента» (браузера), что устраняет необходимость инсталляции дополнительного программного обеспечения у пользователя. Одной из главных задач, возникающих при работе веб-систем, является обеспечение их быстродействия. На сегодняшний день наиболее распространенный способ решения этой задачи заключается в кэшировании данных – размещении данных в промежуточном буфере с целью более быстрого доступа к ним.

Существуют различные подходы кэширования данных и способы их реализации [1].

По механике работы выделяют такие виды кэширования [2]:

- lazy cache (ленивый кэш) – сохраняет данные и отдает их, пока кэш не устареет;
- synchronized cache (синхронизированный кэш) – клиент вместе с данными получает метку и при следующем обращении запрашивает, изменились ли данные, чтоб не получать их повторно. Такой подход реализует http-протокол;
- write-through cache (кэш сквозной записи) – любое изменение данных происходит одновременно как в источнике, так и в кэше.

По типу данных можно выделить кэширование:

- данных, которые отображаются пользователю (front-end);
- данных, которые используют для формирования html-разметки (back-end).

По месту расположения кэша можно выделить такие виды кэширования (рис. 1):

- кэширование на стороне клиента (локальный кэш браузера);
- локальный кэш веб-сервера (используют для кэширования статического контента);
- кэширование сервером приложений.

Несмотря на широкое использование кэширования, общих правил выбора системы кэширования не существует [3, 4] Выбор средств кэширования как правило осуществляется разработчиком на основании собственного опыта и предпочтений, поэтому актуальной является задача исследования эффективности средств кэширования для различных объемов данных.

Цель статьи – получить зависимости времени записи/считывания от количества данных в кэше и определить оптимальные средства кэширования на основании экспериментальных исследований.

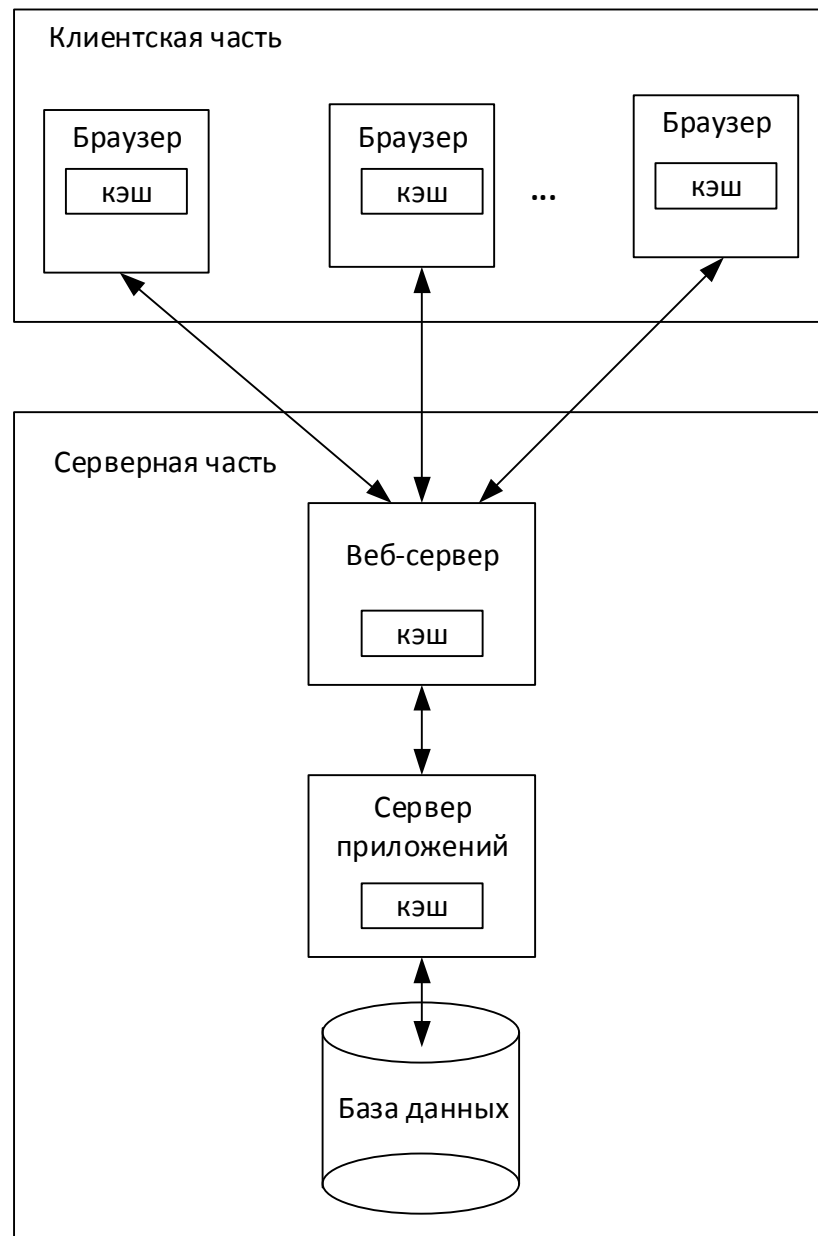


Рис. 1. Возможное размещение средств кэширования

Методика проведения экспериментальных исследований

В исследовании проведён анализ кэширования результатов выполнения SQL-запросов на примере интернет-сайта www.zabudovnyk.com.ua — высоконагруженного портала недвижимости. Кэширование результатов выполнения SQL-запросов обусловлено тем, что обращение к базе данных является «узким местом» многих веб-систем. Портал создано с использованием библиотеки Zend Framework 1, поэтому в качестве средств кэширования будем тестировать программное обеспечение, поддержка которого реализована в Zend Framework:

- APC;
- Memcached;
- Файловый кэш.

Методика проведения эксперимента состоит из таких шагов:

1. Получение экспериментальных данных о скорости *записи* данных в кэш для разных объемов данных. На этом шаге данные выбираются из базы данных, формируют ключ

для идентификации данных в кэше, данные помещаются в кэш. Под временем размещения данных в кэше рассматривают время выполнения функции, которая сохраняет данные в кэш.

2. Получение экспериментальных данных о скорости считывания данных из кэша для данных, помещенных в кэш на шаге 1.
3. Визуализация данных.

Условия проведения эксперимента:

- исследования проводят на тестовом сервере (Linux Fedora, SSD-диски);
- время удаления данных из кэша не исследуют;
- считается, что данные в кэше существуют (попадание в кэш).
- Для получения более точных результатов используют несколько итераций моделирования и определяют усредненные результаты.

С целью гибкой настройки системы и облегчения исследований проведена автоматизация выбора средств кэширования. Тип средства кэширования указывается в конфигурационном файле системы и переключение между средствами кэширования не требует внесения изменений в программный код.

Результаты экспериментальных исследований

Результаты записи данных в кэш приведены в табл. 1 и на рис. 2.

Таблица 1

Время записи данных в кэш, с.

Количество данных/ Кэш	500	5000	50000	500000
APC	0,046	0,46	4,5	45
File	1,8853	70,0000	-	-
Memcached	0,053	0,53	5,3	53

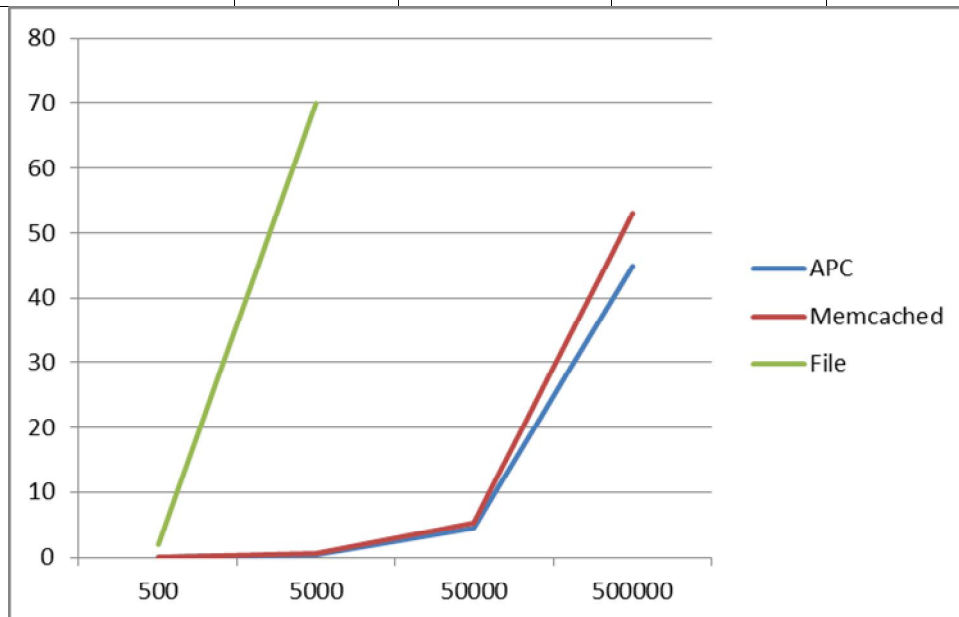


Рис. 2. Результаты записи в кэш

Результаты считывания данных из кэша приведены в табл. 2 и на рис. 3.

Таблица 2

Результаты считывания данных из кэша, с

Количество данных/ Кэш	500	5000	50000	500000
APC	0,0108	0,1089	0,6809	13,3850
File	0,0932	1,0519	-	-
Memcache	0,0178	0,1760	1,7935	18,0930

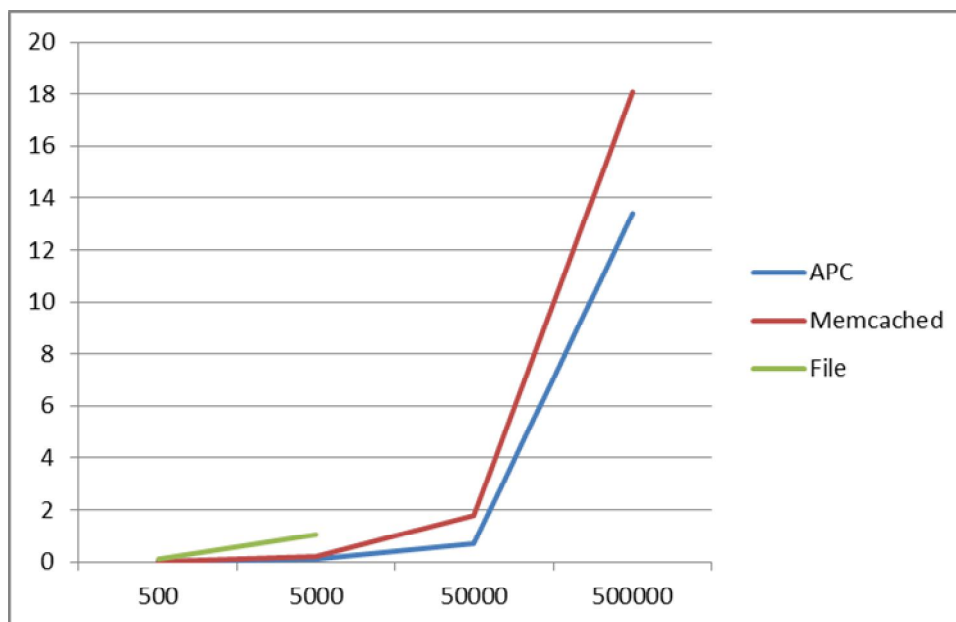


Рис. 3. Результаты считывания данных из кэша

Выводы

Таким образом, проведено исследование средств кэширования, на основе которого можно сделать следующие выводы:

- Средства кэширования данных в оперативной памяти (APC, Memcached) значительно быстрее, чем файловый кэш.
- Для большого количества данных использование файлового кэша нецелесообразно, поскольку каждое значение сохраняется в отдельный файл. По этой причине тестирование файлового кэша для количества итераций более 5000 было прервано через продолжительное время выполнения.
- Время записи превышает время считывания. Для APC и Memcached в несколько раз. Для файлового кэша в несколько десятков раз.
- Для APC и Memcached время записи практически не зависит от количества данных, которые уже находятся в кэше.
- Для APC и Memcached время считывания незначительно увеличивается при увеличении количества данных в кэше.

Таким образом, методику выбора средств кэширования можно сформулировать в виде таких шагов:

1. В качестве основного кэша выбрать программное средство, которое использует для хранения данных оперативную память. Преимущество отдают APC.
2. Если ресурсы оперативной памяти ограничены, можно дополнительно использовать

файловый кэш для кэширования до 5000 сущностей.

В случае использования APC целесообразным будет использование его встроенного веб-интерфейса для мониторинга основных показателей использования кэша (рис. 4).



Рис. 4. Веб-интерфейс APC

СПИСОК ЛИТЕРАТУРЫ

1. Ботыгин И. А. Исследование методов увеличения производительности web-приложений // И. А. Ботыгин, К. А. Каликин / Известия Томского политехнического университета. – 2008. – Т. 312, № 5. – С. 109 – 114.
2. Стас Выщепан Стратегия кэширования в приложении [Электронный ресурс] // Режим доступа : <http://habrahabr.ru/post/168725>.
3. Теория кэша (часть вторая, практическая, дополненная) [Электронный ресурс] // Режим доступа : <http://habrahabr.ru/post/38911/>.
4. Michal Špaček Caching Strategies [Электронный ресурс] // Режим доступа : <http://www.slideshare.net/spaze/caching-strategies>.

Ковалюк Олег Александрович – к. т. н., доцент кафедры компьютерных систем управления. Винницкий национальный технический университет.