**a)Показатели вывода корзины пользователя**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Время загрузки, мс** | **Количество SQL-запросов** | **Количество дубликатов** |
| **до применения декоратора @cached\_property** | 14,25 | 35 | 32 |
| **@cached\_property** | 12,24 | 21 | 12 |

**siege -f /home/django/geekshop/gb\_url.txt -d0 -r20 –c25**. В корзине 20 товаров.

|  |  |  |
| --- | --- | --- |
| **Параметр** | **До применения @cached\_property** | **После применения @cached\_property** |
| Число переходов | 8000 | 8000 |
| Доступность сервера | 100% | 100% |
| Время теста, с | 107,50 | 109,76 |
| Время отклика, с | 0,66 | 0,68 |
| Запросов в секунду | 74,42 | 72,89 |
| Пропускная способность,  МБ/сек | 1,02 | 0,99 |
| Согласованность | 49,27 | 49,30 |
| Удачных транзакций | 8000 | 8000 |
| Неудачных транзакций | 0 | 0 |
| Самая долгая транзакция, с | 3,32 | 3,53 |

Большой разницы не получилось, даже с декоратором немного стала хуже производительность, однако есть улучшения в части sql-запросов: количество дубликатов снизилось с 32 до 12 и время снизилось с 14,25 до 12,24 мс (14%).

b) **Показатели вывода страницы контактов (где в том числе подгружаются данные по общему количеству и общей стоимости товаров)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Время загрузки, мс** | **Количество SQL-запросов** | **Количество дубликатов** |
| **до применения ‘with’** | 8,52 | 13 | 5 |
| **с ‘with’** | 5,57 | 6 | 2 |

**siege -f /home/django/geekshop/gb\_url.txt -d0 -r20 –c25**.

|  |  |  |
| --- | --- | --- |
| **Параметр** | **До применения ‘with’** | **После применения ‘with’** |
| Число переходов | 7000 | 7000 |
| Доступность сервера | 100% | 100% |
| Время теста, с | 61,18 | 62,83 |
| Время отклика, с | 0,43 | 0,44 |
| Запросов в секунду | 114,42 | 111,41 |
| Пропускная способность,  МБ/сек | 1,79 | 1,75 |
| Согласованность | 48,76 | 48,78 |
| Удачных транзакций | 7000 | 7000 |
| Неудачных транзакций | 0 | 0 |
| Самая долгая транзакция, с | 3,39 | 3,37 |

Как видим, серьезного улучшения проверке siege не получилось, однако некоторые улучшения в части sql-запросов налицо: количество дубликатов снизилось с 5 до 2, время запросов снизилось с 8,52 до 5,57 мс (35%).

c) Кэширование на низком уровне на примере страницы http://80.78.244.33/products/category/0/

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Время загрузки, мс** | **Количество SQL-запросов** | **Количество дубликатов** |
| **Без LOW\_CACHE** | 7,68 | 11 | 2 |
| **C LOW\_CACHE** | 7,26 | 11 | 2 |

**siege -f /home/django/geekshop/gb\_url.txt -d0 –r50 –c5**.

|  |  |  |
| --- | --- | --- |
| **Параметр** | **До применения** LOW\_CACHE | **После применения** LOW\_CACHE |
| Число переходов | 3694 | 3670 |
| Доступность сервера | 100% | 100% |
| Время теста, с | 37,92 | 39,05 |
| Время отклика, с | 0,05 | 0,05 |
| Запросов в секунду | 97,42 | 93,98 |
| Пропускная способность,  МБ/сек | 1,07 | 1,04 |
| Согласованность | 4,95 | 4,97 |
| Удачных транзакций | 3694 | 3670 |
| Неудачных транзакций | 0 | 0 |
| Самая долгая транзакция, с | 0,97 | 0,99 |

Полагаю, что у меня что-то слетело, потому что разницы никакой не увидел. Даже еще хуже стало.

Кэширование на низком уровне на примере страницы http://80.78.244.33/products/product/10/

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Время загрузки, мс** | **Количество SQL-запросов** | **Количество дубликатов** |
| **Без LOW\_CACHE** | 6,75 | 8 | 2 |
| **C LOW\_CACHE** | 6,60 | 7 | 2 |

**siege -f /home/django/geekshop/gb\_url.txt -d0 –r10 –c5**.

|  |  |  |
| --- | --- | --- |
| **Параметр** | **До применения** LOW\_CACHE | **После применения** LOW\_CACHE |
| Число переходов | 400 | 400 |
| Доступность сервера | 100% | 100% |
| Время теста, с | 5,77 | 5,68 |
| Время отклика, с | 0,07 | 0,07 |
| Запросов в секунду | 69,32 | 70,42 |
| Пропускная способность,  МБ/сек | 0,95 | 0,96 |
| Согласованность | 4,88 | 4,75 |
| Удачных транзакций | 400 | 400 |
| Неудачных транзакций | 0 | 0 |
| Самая долгая транзакция, с | 0,48 | 0,44 |

А вот здесь снизилось количество sql-запросов, время запросов снизилось на 2%.

d) Кэширование шаблонов на примере страницы http://80.78.244.33/order/update/11/

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Время загрузки, мс** | **Количество SQL-запросов** | **Количество дубликатов** |
| **Без {% load cache %}** | 9,55 | 14 | 8 |
| **C {% load cache %}** | 6,38 | 6 | - |

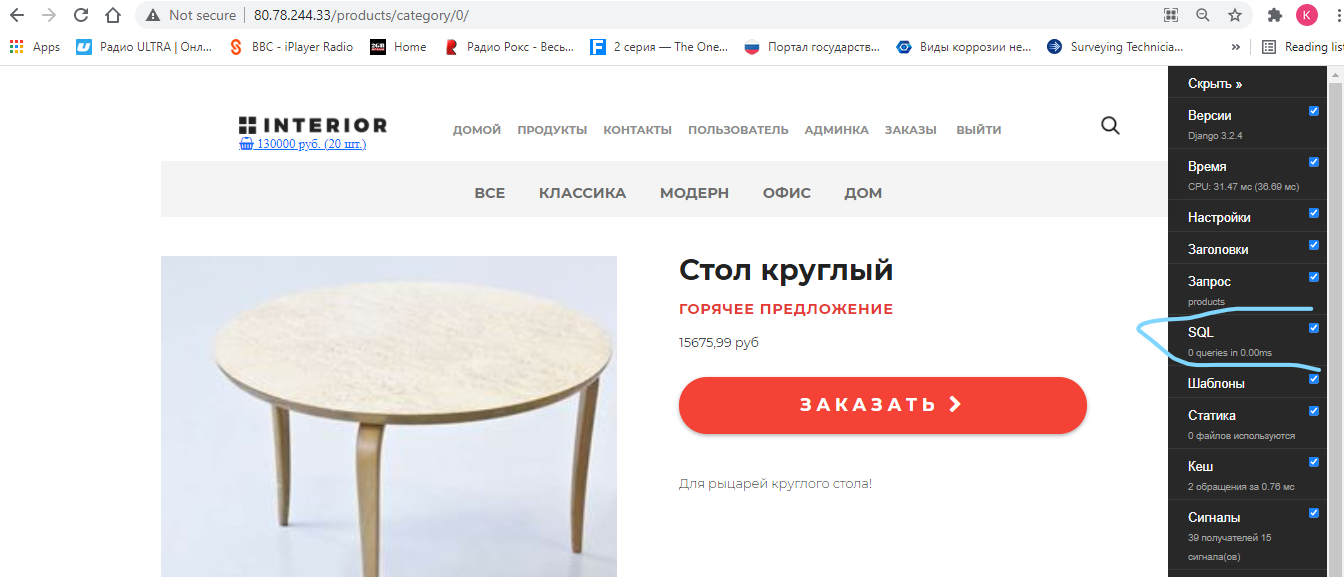
**siege -f /home/django/geekshop/gb\_url.txt -d0 –r10 –c5**.

|  |  |  |
| --- | --- | --- |
| **Параметр** | **До применения {% load cache %}** | **После применения {% load cache %}** |
| Число переходов | 500 | 500 |
| Доступность сервера | 100% | 100% |
| Время теста, с | 23,28 | 19,08 |
| Время отклика, с | 0,22 | 0,18 |
| Запросов в секунду | 21,48 | 26,21 |
| Пропускная способность,  МБ/сек | 0,50 | 0,59 |
| Согласованность | 4,83 | 4,82 |
| Удачных транзакций | 500 | 500 |
| Неудачных транзакций | 0 | 0 |
| Самая долгая транзакция, с | 2,86 | 2,79 |

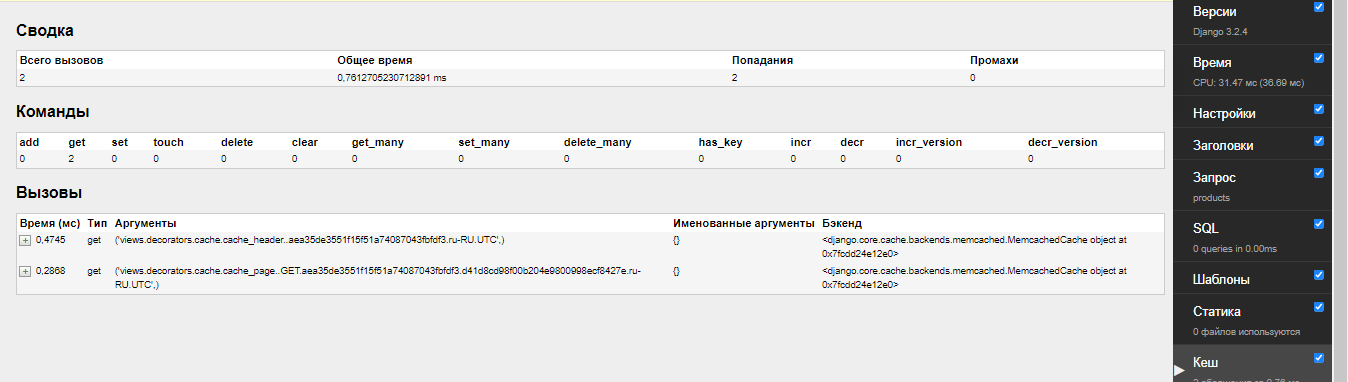
Ощутимое снижение количества sql-запросов с 14 до 6 без дубликатов, время теста снизилось на 18%!!!

e) Кэширование контроллера на примере страницы <http://80.78.244.33/products/category/0/> c помощью @cache\_page(3600)

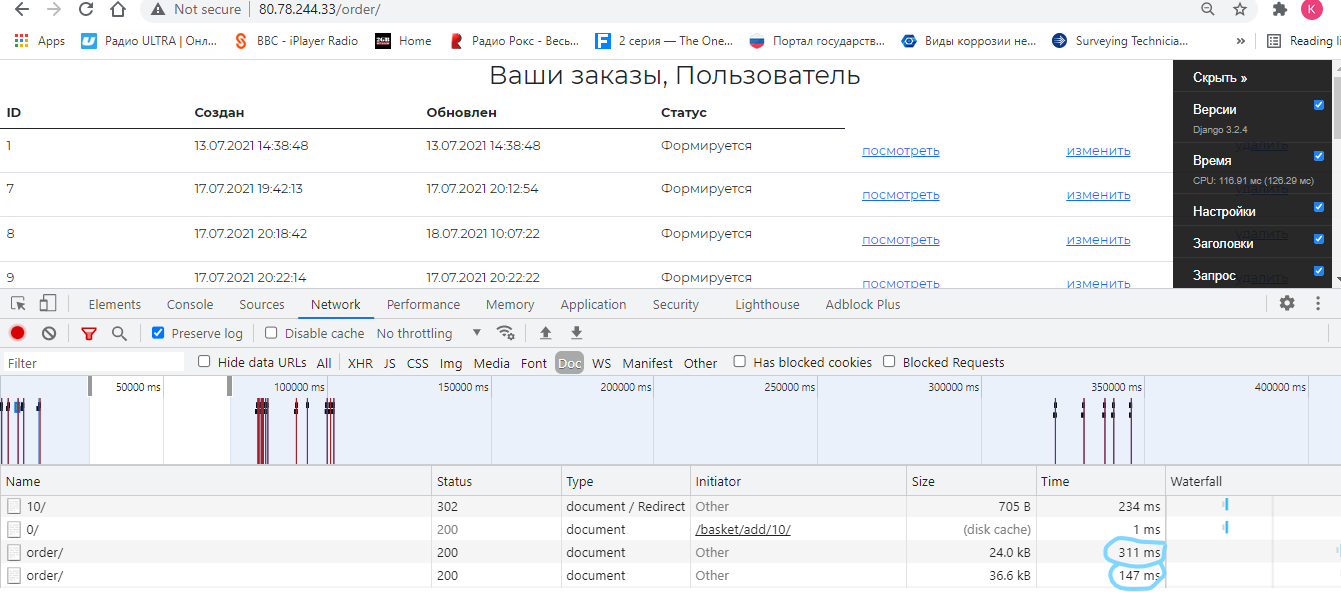
Нет ни одного запроса после реализации кэширования контроллера



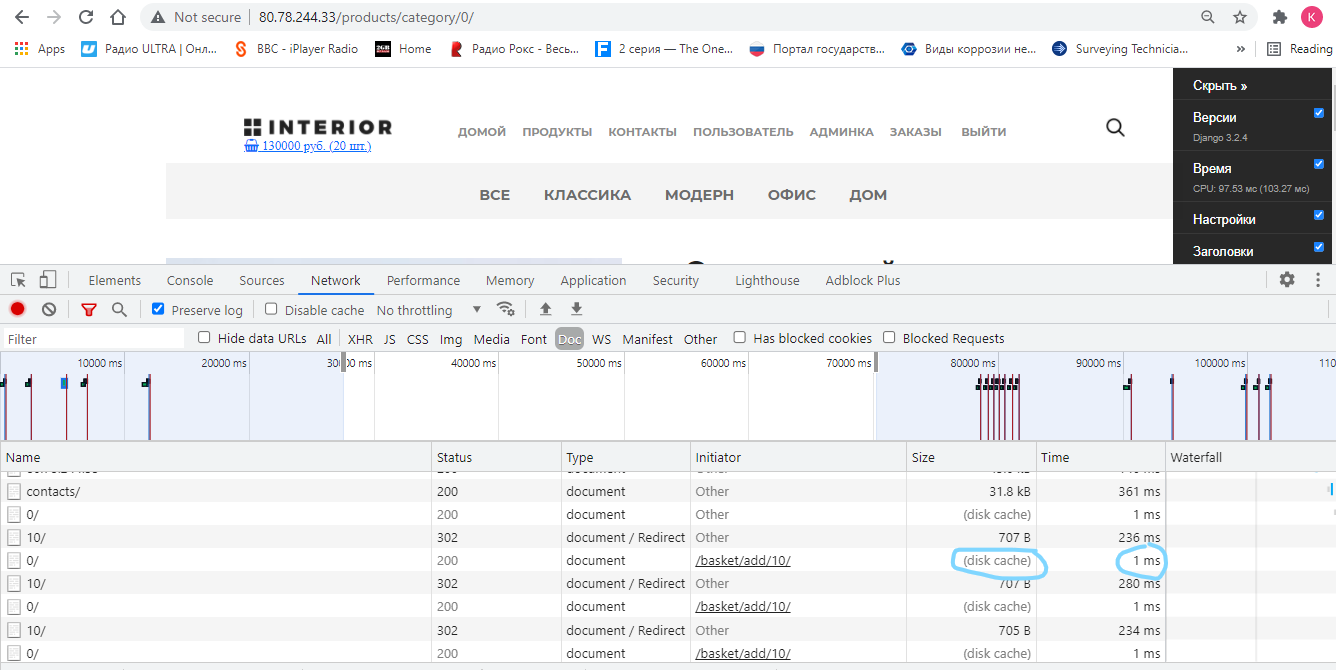
Все данные страницы упали в кэш:



f) Кэширование сайта



Загрузка страницы снизилась с 311 мс до 147 мс.



Страница закэшировалась, но есть проблемы с добавлением товаров в корзину. При нажатии на товар не меняется цена и количество в корзине, потому что страница грузится из кэша.