

Тестируем кэширование при помощи *memcached*

Результат работы *siege* по следующим url:

<http://151.248.113.248/products/>

<http://151.248.113.248/products/category/0>

<http://151.248.113.248/products/category/1>

<http://151.248.113.248/products/category/2>

<http://151.248.113.248/products/category/3>

<http://151.248.113.248/products/category/4>

<http://151.248.113.248/products/product/9>

без кэширования:

```
(venv) kris@151-248-113-248:~/Django-2021-1
$ siege -f products_url.txt -d0 -r20 -c255
** SIEGE 4.0.4
** Preparing 255 concurrent users for battle.
The server is now under siege ...
Transactions:          30439 hits
Availability:          100.00 %
Elapsed time:          105.37 secs
Data transferred:      237.53 MB
Response time:          0.86 secs
Transaction rate:       288.88 trans/sec
Throughput:             2.25 MB/sec
Concurrency:           247.41
Successful transactions: 27379
Failed transactions:     0
Longest transaction:    8.67
Shortest transaction:    0.00
```

Добавляем функции низкоуровневого кэширования, активируем кэширование. Получаем результат:

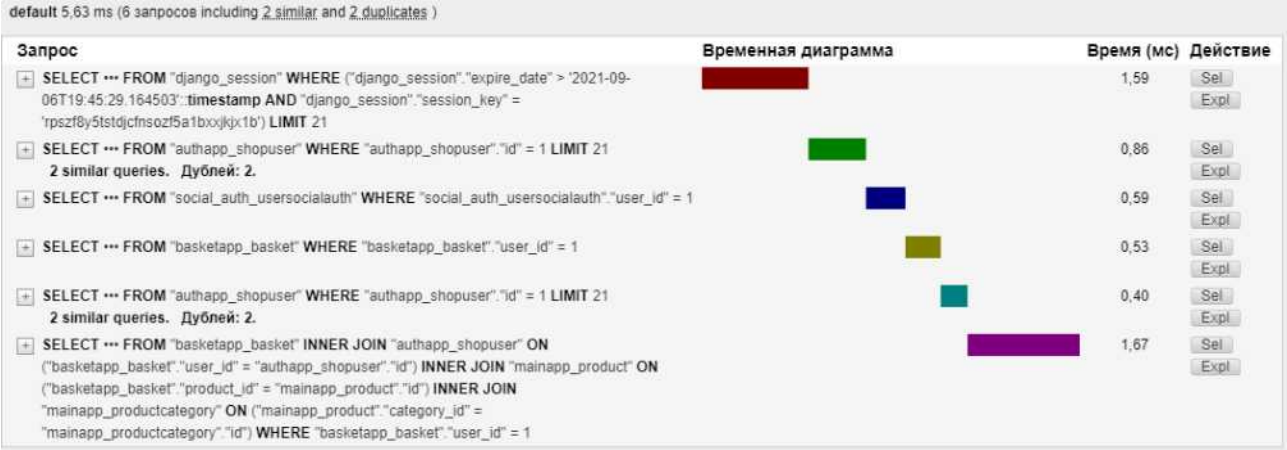
```
(venv) kris@151-248-113-248:~/Django-2021-1
$ siege -f products_url.txt -d0 -r20 -c255
** SIEGE 4.0.4
** Preparing 255 concurrent users for battle.
The server is now under siege ...
Transactions:          30505 hits
Availability:          100.00 %
Elapsed time:          91.37 secs
Data transferred:      238.37 MB
Response time:          0.75 secs
Transaction rate:       333.86 trans/sec
Throughput:             2.61 MB/sec
Concurrency:           249.01
Successful transactions: 27445
Failed transactions:     0
Longest transaction:    5.85
Shortest transaction:    0.00
```

Время выполнения теста уменьшилось примерно на 13.29%, что, конечно, немного. С другой на больших объёмах данные результаты могут быть более впечатляющими.

Кэширование корзины (@cached\_property)  
без кэширования:

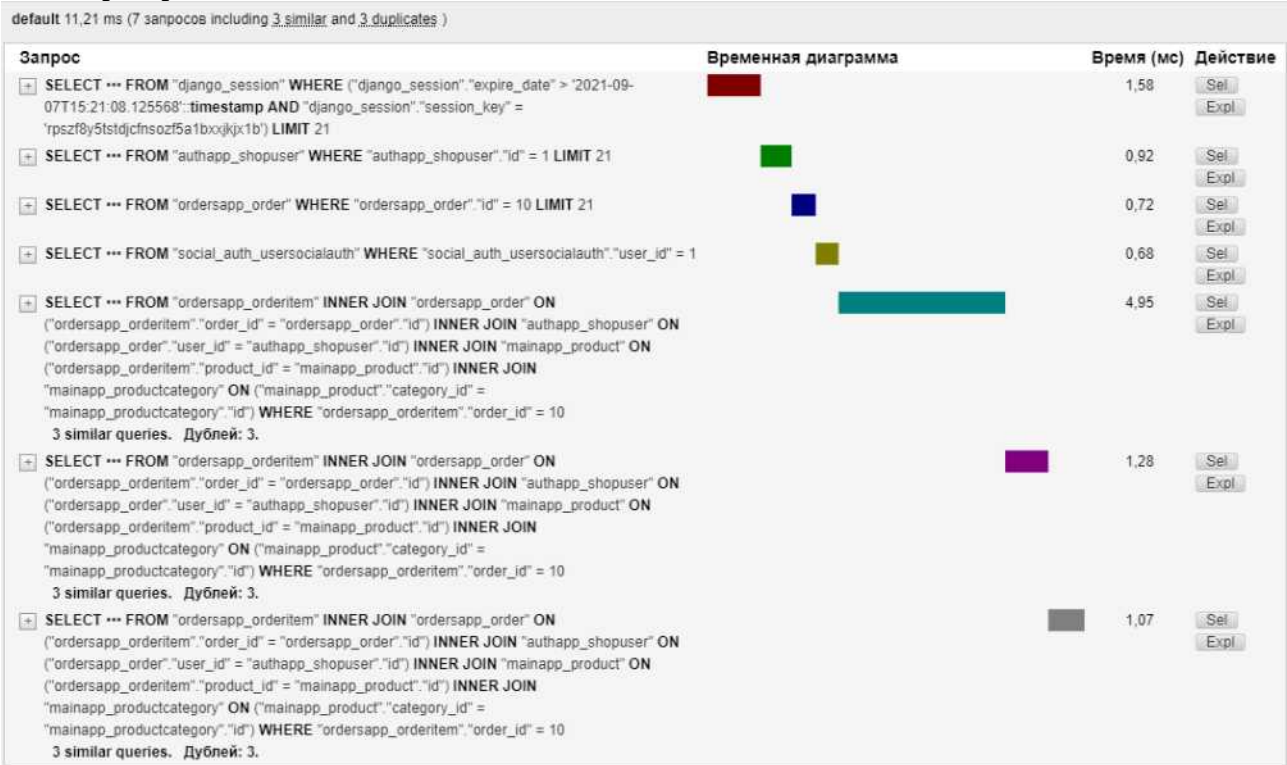


С кэшированием:

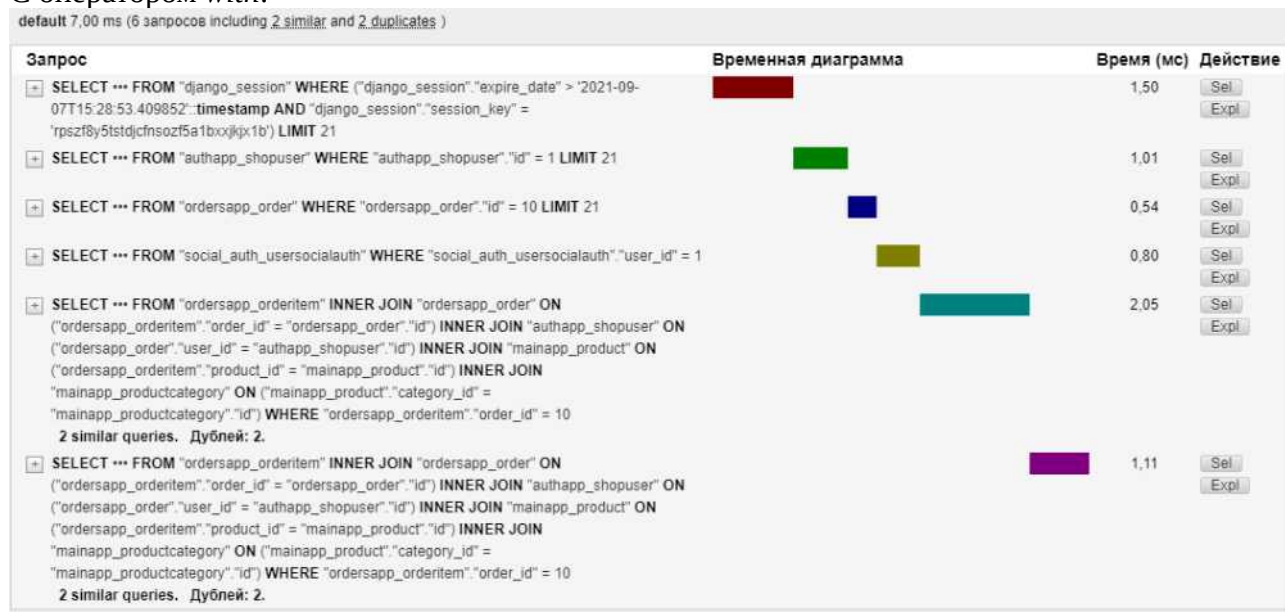


Хотя количество дублирующих запросов уменьшилось, время исполнения увеличилось.

Оптимизация заказа (оператор шаблона with).  
Без оператора with:



С оператором *with*:



Более 37% выигрыша во времени!

Кэширование шаблонов. Предварительно устанавливаем settings.CACHE\_LOW = False.

Результат работы siege по следующим url:

<http://151.248.113.248/order/update/1>

<http://151.248.113.248/order/update/2>

<http://151.248.113.248/order/update/3>

<http://151.248.113.248/order/update/4>

<http://151.248.113.248/order/update/5>

<http://151.248.113.248/order/update/6>

<http://151.248.113.248/order/update/7>

<http://151.248.113.248/order/update/8>

<http://151.248.113.248/order/update/9>

<http://151.248.113.248/order/update/10>

без кэширования:

```
kris@151-248-113-248:~/Django-2021-1
$ siege -i -f order_update_urls.txt -d0 -r20 -c255
** SIEGE 4.0.4
** Preparing 255 concurrent users for battle.
The server is now under siege ...
Transactions:          66300 hits
Availability:          100.00 %
Elapsed time:          341.32 secs
Data transferred:      834.32 MB
Response time:          1.28 secs
Transaction rate:       194.25 trans/sec
Throughput:             2.44 MB/sec
Concurrency:           248.52
Successful transactions: 61200
Failed transactions:      0
Longest transaction:     14.22
Shortest transaction:     0.00
```



С кэшированием:

```
kris@151-248-113-248:~/Django-2021-1
$ siege -i -f order_update_urls.txt -d0 -r20 -c255
** SIEGE 4.0.4
** Preparing 255 concurrent users for battle.
The server is now under siege...
Transactions:          66300 hits
Availability:          100.00 %
Elapsed time:          289.33 secs
Data transferred:      833.96 MB
Response time:          1.09 secs
Transaction rate:       229.15 trans/sec
Throughput:             2.88 MB/sec
Concurrency:           248.69
Successful transactions: 61200
Failed transactions:    0
Longest transaction:    13.34
Shortest transaction:   0.00
```

Получили выигрыш по времени около 15%. Информация от DjDT на примере одной страницы (<http://151.248.113.248/order/update/3/>):

Обращения к кешу от 1 бэкенда

#### Сводка

Всего вызовов	Общее время	Попадания	Пропуски
12	3,1862258911132812 ms	12	0

#### Команды

add	get	set	touch	delete	clear	get_many	set_many	delete_many	has_key	incr	decr	incr_version	decr_version
0	12	0	0	0	0	0	0	0	0	0	0	0	0

#### Вызовы

Время (мс)	Тип	Аргументы	Именованные аргументы	Бэкенд
1,3113	get	("template.cache.orderitemform_product.314adc507266cc256fcc9d72d670729d",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,2742	get	("template.cache.orderitemform_product.247730f9d0d2eaad265a470e32aa0cdf",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,2148	get	("template.cache.orderitemform_product.6846ba559f7f55c20a67d890d03d20e0",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1798	get	("template.cache.orderitemform_product.ac435aa8d7932bc7c26dbe0277119a3",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1676	get	("template.cache.orderitemform_product.cdee9bf40a070d58d14dfa3bb61e0032",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1616	get	("template.cache.orderitemform_product.6846ba559f7f55c20a67d890d03d20e0",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1526	get	("template.cache.orderitemform_product.2fd7e333d36411b88fb3becbaf8188",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1626	get	("template.cache.orderitemform_product.3afcd9feb6ecfdd0ba628696e3cc163",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1643	get	("template.cache.orderitemform_product.6846ba559f7f55c20a67d890d03d20e0",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1326	get	("template.cache.orderitemform_product.6846ba559f7f55c20a67d890d03d20e0",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1330	get	("template.cache.orderitemform_product.9a1c7ee2c7ce38d4bbbf29ab9f2ac1e",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>
0,1318	get	("template.cache.orderitemform_product.6846ba559f7f55c20a67d890d03d20e0",)		<django.core.cache.backends.memcached.MemcachedCa object at 0x7f007a25ae50>

Закешируем страницу products целиком, используя cache\_page. До кеширования:

```
kris@151-248-113-248:~/Django-2021-1
$ siege http://151.248.113.248/products/ -d0 -r10 -c255
** SIEGE 4.0.4
** Preparing 255 concurrent users for battle.
The server is now under siege...
Transactions:      30000 hits
Availability:      100.00 %
Elapsed time:      59.56 secs
Data transferred:  309.44 MB
Response time:     0.47 secs
Transaction rate:  503.69 trans/sec
Throughput:        5.20 MB/sec
Concurrency:       237.52
Successful transactions: 30000
Failed transactions: 0
Longest transaction: 6.15
Shortest transaction: 0.00
```

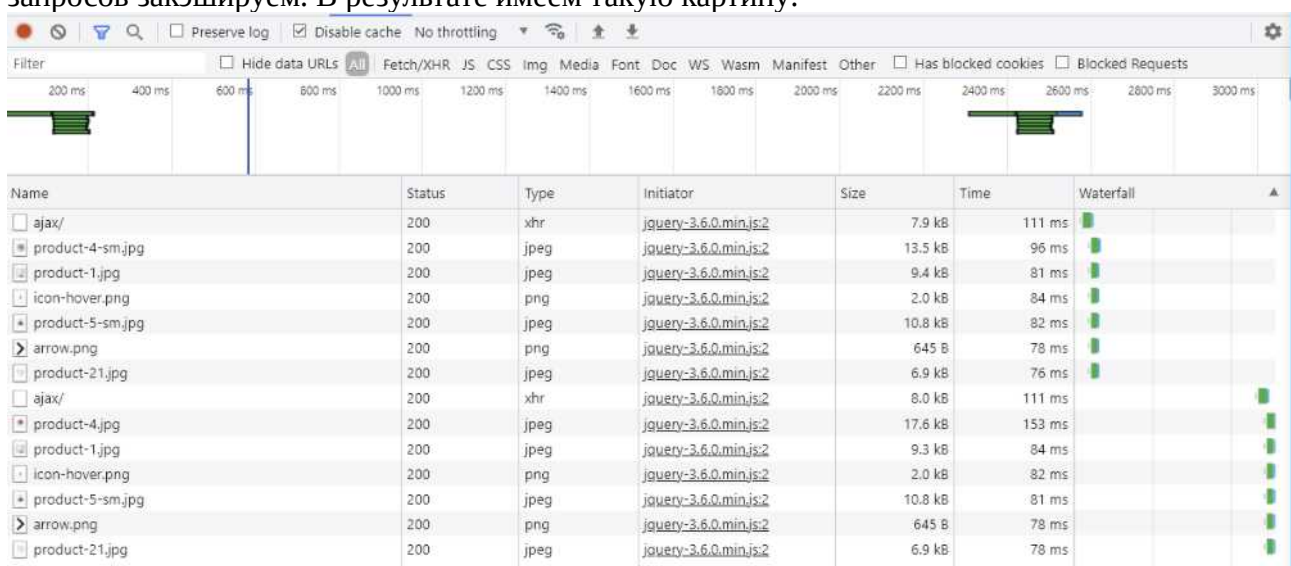
После кеширования:

```
kris@151-248-113-248:~/Django-2021-1
$ siege http://151.248.113.248/products/ -d0 -r10 -c255
** SIEGE 4.0.4
** Preparing 255 concurrent users for battle.

The server is now under siege...
Transactions:      29945 hits
Availability:      100.00 %
Elapsed time:      60.78 secs
Data transferred:  308.69 MB
Response time:     0.49 secs
Transaction rate:  492.68 trans/sec
Throughput:        5.08 MB/sec
Concurrency:       239.64
Successful transactions: 29945
Failed transactions: 0
Longest transaction: 7.03
Shortest transaction: 0.00
```

Выигрыша однако, нет ☹️

Теперь рассмотрим использования AJAX для оптимизации запросов. Добавим к странице products код, который будет обновлять страницу через jQuery запросы (меню категорий, горячее предложение, товары категории, пейджинатор, похожие продукты) а обработчика запросов закешируем. В результате имеем такую картину:



При выборе элемента меню, на запрос jQuery, передаётся совсем немного данных (которые при этом закешированы сервером) плюс догружается только незначительная доля медиа-контента.