

# Оптимизация производительности

Для решения проблем производительности, необходимо последовательно на разных уровнях выявить и устранить узкие места.

## Сервер

Каждый запрос пользователя к ПО (сеанс работы с системой) расходует ресурсы сервера. При одновременной работе многих пользователей с ПО свободные ресурсы сервера могут закончиться. Это будет проявляться в медленной работе ПО (равно для операций чтения и записи).

Предварительно, необходимо убедиться, что при нормальной (обычной) нагрузке на сервере остаются свободными 20%-30% оперативной памяти.

Работа ПО связана с интенсивным обращением к дискам (СУБД, файловый кеш). Необходимо убедиться, что дисковая подсистема справляется и не формируются длинные очереди ожидания операций чтения/записи.

Иногда задержки могут быть связаны с проблемами сетевого характера.

## Веб-сервер

Для анализа загрузки веб-сервера Apache используется штатный модуль `mod_status`, информация от которого доступна по пути `http://сервер/server-status`

Информация чувствительная, поэтому по умолчанию доступ к этому модулю ограничен. Чтобы снять ограничения, выполните настройку веб-сервера:

```
docker exec -it alm-app bash
vi /etc/apache2/mods-enabled/status.conf
```

Необходимо заменить **Require local** на **Require all granted**

```
<Location /server-status>
    SetHandler server-status
    Require all granted
</Location>
```

Настройки вступят в силу только после перезапуска контейнера:

```
docker restart alm-app
```

Анализ информации осуществляется следующим образом:

1. Оцениваем количество доступных слотов для подключения (1 на скриншоте ниже). Точки и подчеркивания - это доступные слоты, буквы - занятые слоты. При нормальной нагрузке должны быть доступными 50% слотов для подключений к веб-серверу.
2. Оцениваем время обработки запроса РНР-процессором (2 на скриншоте ниже). Большое время говорит о наличии проблем на уровне приложения или СУБД.
3. Однако, необходимо также проверить запросы к файлам (не приложению), например, загрузку JS или таблиц стилей (CSS) - 3 на скриншоте ниже. Если медленно загружаются эти файлы

(статика), то проблема на уровне веб-сервера. Например, может быть некорректно сконфигурирован `php-fpm`, его таймауты или ограничения по числу подключений, формируют узкое место в приложении.

# Apache Server Status for devprom.local (via 127.0.0.1)

Server Version: Apache/2.4.39 (Win32) OpenSSL/1.1.1g PHP/7.4.33  
Server MPM: WinNT  
Apache Lounge VC15 Server built: May 29 2019 10:38:01

Current Time: Tuesday, 28-Mar-2023 09:45:18 RTZ 2 (çèià)  
Restart Time: Sunday, 26-Mar-2023 16:53:27 RTZ 2 (çèià)  
Parent Server Config. Generation: 1  
Parent Server MPM Generation: 0  
Server uptime: 1 day 16 hours 51 minutes 51 seconds  
Server load: -1.00 -1.00 -1.00  
Total accesses: 13593 - Total Traffic: 111.3 MB - Total Duration: 154069519  
.0924 requests/sec - 793 B/second - 8.4 kB/request - 11334.5 ms/request  
2 requests currently being processed, 62 idle workers

\_\_\_\_\_R\_\_\_\_\_W\_\_\_\_\_

1

Scoreboard Key:

"\_" Waiting for Connection, "s" Starting up, "r" Reading Request,  
"w" Sending Reply, "k" Keepalive (read), "d" DNS Lookup,  
"c" Closing connection, "l" Logging, "g" Gracefully finishing,  
"I" Idle cleanup of worker, "." Open slot with no current process

Srv	PID	Acc	M	SS	Req	Dur	Conn	Child	Slot	Client	Protocol	VHost	
0-0	29604	0/586/586	_	43642	7	13724495	0.0	5.71	5.71	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/2133/2133	_	44344	4370	11926551	0.0	1.93	1.93	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/1716/1716	R	0	2	23142051	0.0	14.58	14.58	127.0.0.1	http/1.1		
0-0	29604	0/1173/1173	_	135679	429	6303262	0.0	16.84	16.84	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/1184/1184	_	10	2	14212842	0.0	14.08	14.08	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/857/857	_	43654	10246	12168413	0.0	7.42	7.42	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/1174/1174	_	43642	2	19243188	0.0	12.47	12.47	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/1306/1306	W	0	0	14471589	0.0	7.89	7.89	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/1794/1794	_	44344	4410	12769249	0.0	10.48	10.48	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/535/535	_	55637	7267	11605598	0.0	4.74	4.74	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/695/695	_	55637	7245	11043342	0.0	10.47	10.47	127.0.0.1	http/1.1	devprom.local:80	GE
0-0	29604	0/440/440	_	135680	3771	3328935	0.0	4.72	4.72	127.0.0.1	http/1.1	devprom.local:80	GE

Srv Child Server number - generation

PID OS process ID

Acc Number of accesses this connection / this child / this slot

M Mode of operation

SS Seconds since beginning of most recent request

Req Milliseconds required to process most recent request

Dur Sum of milliseconds required to process all requests

Conn Kilobytes transferred this connection

При большом количестве одновременно работающих пользователей может не хватать доступных слотов для подключения к веб-серверу. Выражается это в том, что пользователь ожидает ответа от сервера, но до приложения запрос не доходит, а стоит в очереди. Для управления настройкой доступных слотов необходимо конфигурировать модуль `mpm_prefork.conf`

```
docker exec -it alm-app bash
vi /etc/apache2/mods-enabled/mpm_prefork.conf
```

Ниже представлен один из вариантов конфигурации:

```
<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers  5
    MaxSpareServers  15
    ServerLimit      900
    MaxClients       900
    MaxRequestsPerChild 150
</IfModule>
```

Настройки вступят в силу только после перезапуска контейнера:

```
docker restart alm-app
```

При использовании PHP-FPM настройка количества слотов осуществляется в файле `/etc/php-fpm.d/www.conf`

Необходимо установить параметр `max_children`, например, следующим образом:

```
pm.max_children = 900
```

## Приложение

Само приложение вряд ли будет являться узким местом, поскольку не обращается ни к каким внешним ресурсам, кроме СУБД. Однако, возможны неоптимальные запросы, которые требуют оптимизации силами разработчиков.

## СУБД

Для анализа производительности СУБД лучше всего начать с простых проверок. Необходимо подключиться к СУБД и выполнить запрос:

```
show processlist;
```

Необходимо проанализировать перечень текущих запросов под нормальной нагрузкой:

1. В столбце `Time` отображается время выполнения запроса в секундах. Большое значение - это признак потенциальной проблемы. Однако, необходимо учитывать, что иногда запросы по природе своей могут быть тяжелыми (долгими), например, когда запрашивается большой объем данных. Проблема в работе СУБД - это медленно выполняющийся простой запрос.
2. Блокировки иногда являются причиной долгого выполнения запросов. Блокировки могут быть вызваны ошибками в работе СУБД (требуется перезагрузка), либо ошибками в работе приложения (требуется оптимизация приложения).

```
mysql> show processlist;
+-----+-----+-----+-----+-----+-----+-----+
| Id    | User  | Host          | db      | Command | Time | Status |
+-----+-----+-----+-----+-----+-----+-----+
| 89951 | root  | localhost     | NULL    | Query   | 0    | start |
| 89981 | devprom | localhost:49056 | db_     | Sleep   | 1    |      |
| 90002 | devprom | localhost:49134 | db_instance | Query   | 11   | Send |
| 90064 | devprom | localhost:49350 | db_instance | Query   | 0    | Send |
| 90067 | devprom | localhost:49364 | db_     | Sleep   | 0    |      |
| 90068 | devprom | localhost:49368 | db_instance | Query   | 0    | Send |
| 90069 | devprom | localhost:49370 | db_instance | Sleep   | 0    |      |
+-----+-----+-----+-----+-----+-----+-----+

```

Если в режиме реального времени не удается обнаружить аномальную нагрузку, то необходимо включить логирование запросов в конфигурационном файле, например, /etc/mysql/conf.d/z-devprom.conf

```
slow_query_log = 1
slow_query_log_file = /var/log/mysql/mysql-slow.log
long_query_time = 5
```

После перезапуска MySQL в указанном логе будут собираться запросы, которые выполнялись более 5 секунд.

Если обнаруживаются медленные запросы (Query\_time, сек.), при обработке которых затрагивалось небольшое количество строк (Rows\_examined) - это признак наличия узких мест на сервере (отсутствие доступных ресурсов сервера, либо медленная дисковая подсистема). Неоптимальное использование ресурсов сервера также может быть вызвано недостаточной оптимизацией самой СУБД, которая выполняется различными настройками (кеши разных уровней).

```
# Time: 2023-03-20T19:01:21.434997Z
# User@Host: devprom[devprom] @ localhost [127.0.0.1] Id: 7362
# Query_time: 3.074892 Lock_time: 0.394212 Rows_sent: 0 Rows_examined: 10
SET timestamp=1679088078;
DELETE FROM co_AffectedObjects WHERE UNIX_TIMESTAMP(NOW()) - UNIX_TIMESTAMP(RecordModified) > 40;
```