Устранение проблем

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>
```

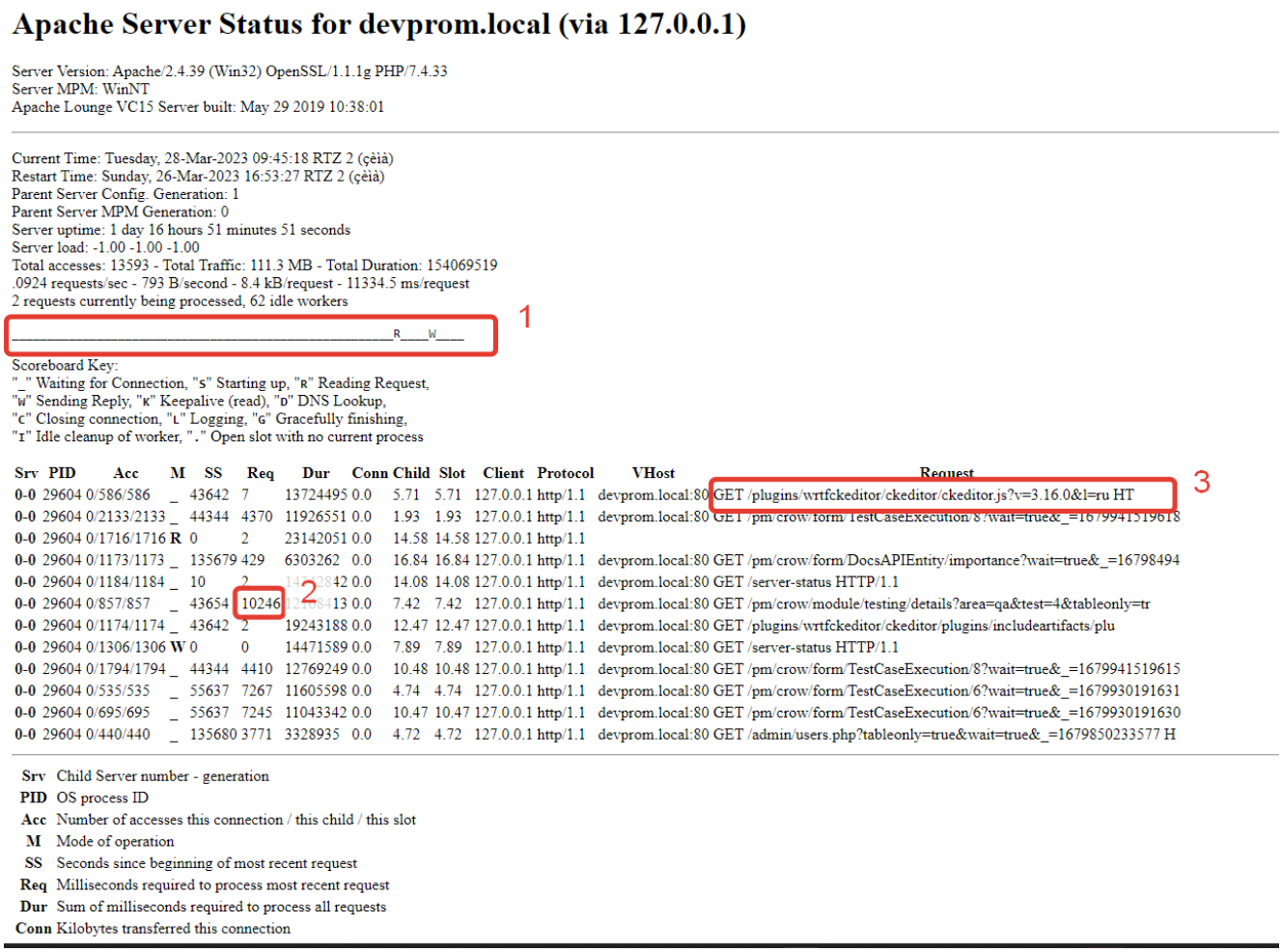
Настройки вступят в силу только после перезапуска контейнера:

```
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 PHP-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 (çèìà)
Restart Time: Sunday, 26-Mar-2023 16:53:27 RTZ 2 (çèìà)
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

1

_____R___W___

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	Request
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	GET /plugins/wrtfckeditor/ckeditor/ckeditor.js?v=3.16.0&l=ru HT
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	GET /pm/crow/form/TestCaseExecution/8?wait=true&_=1679941519618
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	GET /pm/crow/form/DocsAPIEntity/importance?wait=true&_=16798494
0-0	29604	0/1184/1184	_	10	2	14[illegible]28442	0.0	14.08	14.08	127.0.0.1	http/1.1	devprom.local:80	GET /server-status HTTP/1.1
0-0	29604	0/857/857	_	43654	10246	12[illegible]8413	0.0	7.42	7.42	127.0.0.1	http/1.1	devprom.local:80	GET /pm/crow/module/testing/details?area=qa&test=4&tableonly=tr
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	GET /plugins/wrtfckeditor/ckeditor/plugins/includeartifacts/plu
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	GET /server-status HTTP/1.1
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	GET /pm/crow/form/TestCaseExecution/8?wait=true&_=1679941519615
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	GET /pm/crow/form/TestCaseExecution/6?wait=true&_=1679930191631
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	GET /pm/crow/form/TestCaseExecution/6?wait=true&_=1679930191630
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	GET /admin/users.php?tableonly=true&wait=true&_=1679850233577 H

2

3

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.

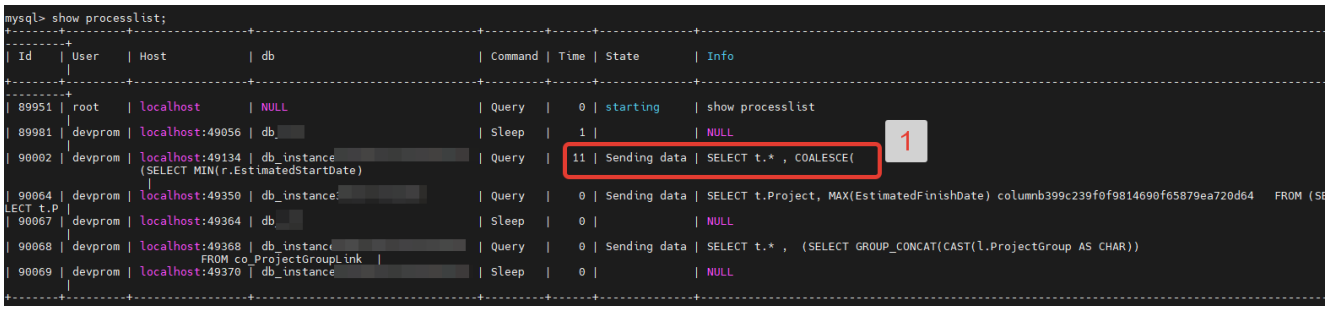
СУБД

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 вызываны ошибками в работе СУБД (требуется перезагрузка), либо ошибками в работе приложения (требуется оптимизация приложения).



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 медленая дисковая подсистема). Неоптимальное использование ресурсов сервера также может быть вызывано недостаточной оптимизацией самой СУБД, которая выполняется различными настройками (кеши разных уровней).

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

В логах приложения (system.log) могут встречаться ошибки вида:

```
ERROR System - Too many connections
...
ERROR System - [2006] MySQL server has gone away
```

Это может говорить о том, что нет свободных подключений в пуле подключений СУБД, либо недостаточно памяти для создания новых подключений.

Если памяти достаточно, то можно увеличить лимит подключений настройкой файла /home/devprom/docker/mysql/devprom.cnf (или непосредственно на хосте с СУБД /etc/mysql/conf.d/z-devprom.conf)

```
max_connections = 400
```

Для выбора остальных настроек СУБД в соответствии с наличием свободной памяти и количества одновременных пользователей, можно использовать следующий запрос:

```
SELECT ( @@key_buffer_size + @@tmp_table_size + @@innodb_buffer_pool_size +
@@innodb_log_buffer_size + @@max_connections * ( @@read_buffer_size +
@@read_rnd_buffer_size + @@sort_buffer_size + @@join_buffer_size +
@@binlog_cache_size + @@thread_stack ) ) / 1073741824 AS MAX_MEMORY_GB;
```

Проблемы с самоподписными сертификатами в Git

После подключения репозитория к Devprom ALM в логах наблюдается такая ошибка:

```
2022-08-25 05:11:49,461 Refresh repository revisions: https://***
2022-08-25 05:11:49,462 Latest persisted revision:
2022-08-25 05:11:49,462 GitConnector: Getting all commits
2022-08-25 05:11:49,462 GIT COMMAND
(/var/www/devprom/git_repo/c8b873acceb52d14c79706e97235ae0b)
'/usr/bin/git' fetch
2022-08-25 05:11:49,501 GitConnector: fatal: unable to access 'https://***': server
certificate verification failed. CAfile: none CRLfile: none
2022-08-25 05:11:49,501 There are 0 more revisions in the repository
```

Для исправления ситуации попробуйте выполнить следующие команды на сервере, где установлен Git-клиент:

```
git config --list
git config --global http.sslverify true
git config --list
git clone https://***
```

Попробуйте явно прописать настройки использования SSL в конфигах репозитория, отредактируйте файл /var/www/devprom/git_repo/c8b873acceb52d14c79706e97235ae0b/config и добавьте в него следующие данные:

```
[http]
        sslCAinfo = /etc/ssl/certs/ca-certificates.crt
        sslVerify = false
```

Не приходит письмо с восстановлением пароля

Если на сервере не настроена почта, либо отправка почтовых уведомлений не работает, а пароль к учетной записи потерян, можно сделать следующее:

- Обратиться к администратору приложения и попросить сбросить пароль на какое-то простое значение, а также установить галочку "Запросить смену пароля при авторизации" в

карточке пользователя.
- Обойти механизм аутентификации.

Обход аутентификации

Если в приложении нет активных пользователей, то механизм аутентификации не используется и можно выполнить любые действия с полномочиями администратора.

Для этого необходимо выполнить действия в базе данных приложения. Подключаемся к БД:

```
mysql --user=devprom --password=devprom_pass --database=devprom
```

В открывшейся консоли выполняем запросы:

```
UPDATE cms_User SET IsAdmin = 'N';
INSERT INTO cms_BlackList (SystemUser) SELECT cms_UserId FROM cms_User;
```

Очищаем кеш приложения (Администрирование - Настройки - Приложение, кнопка "Очистить кеш") и перезапускаем сервис Apache2

Открываем приложение с использованием URL <server-name>/admin и переходим в меню Пользователи - Блокировки. Удаляем все блокировки. Затем переходим к списку пользователей, находим нужную учетную запись, открываем на редактирование и меняем пароль, а также устанавливаем галочку "Является администратором".

Теперь можно аутентифицироваться под своей учетной записью.

Конфигурация с LDAP-аутентификацией

Если вы настроили веб-сервер apache для аутентификации посредством LDAP-каталога, то для обхода аутентификации требуется дополнительно выполнить настройки в конфигурационном файле apache httpd.conf или devprom.conf (выделено жирным шрифтом):

<LocationMatch "/(join-project|plugins|login|recovery|reset|404|50[\d]{1}|cache|scripts|images|img|favicon|app/non-auth|tasks|**admin**)">

После перезапуска apache можно получить доступ к административному разделу без аутентификации.

Затем, когда доступ к учетной записи будет восстановлен, необходимо отменить выполненные изменения.

Невозможно загрузить файл на сервер

Если вам не удается загрузить обновление, фотографию или добавить приложение к заявке, то проверьте, включена ли возможность загрузки файлов на сервер. Убедитесь, что параметры PHP имеют значения, как показано ниже. Сделать это можно открыв страницу приложения по адресу /admin/info (только под администратором приложения).

```
file_uploads = on
upload_max_filesize = 300M
post_max_size = 300M
```

Если значения этих параметров отличаются, то необходимо их установить, отредактировав файл настроек PHP:

```
/etc/php/7.x/apache2/conf.d/devprom.ini
```

Если эти параметры установлены, то проверьте загрузку файла следующим образом. Выполните этот скрипт, предварительно указав пути, соответствующие вашему варианту установки Devprom:

```
cat > /var/www/devprom/htdocs/uploads.php << EOF
<?php
\$target_path = "cache/";
\$target_path = \$target_path . basename( \$_FILES['uploadedfile']['name']);
if(move_uploaded_file(\$_FILES['uploadedfile']['tmp_name'], \$target_path)) {
echo "The file ".  basename( \$_FILES['uploadedfile']['name']). " has been uploaded";
} else{
echo "There was an error uploading the file, please try again!";
}
?>
EOF
chmod 775 /var/www/devprom/htdocs/uploads.php
echo text > /tmp/file
curl -F "uploadedfile=@/tmp/file" http://127.0.0.1/uploads.php
cat /var/www/devprom/htdocs/cache/file
unlink /var/www/devprom/htdocs/uploads.php
```

Если все настроено верно, то в консоли вы увидите два сообщения:

```
The file file has been uploaded
text
```