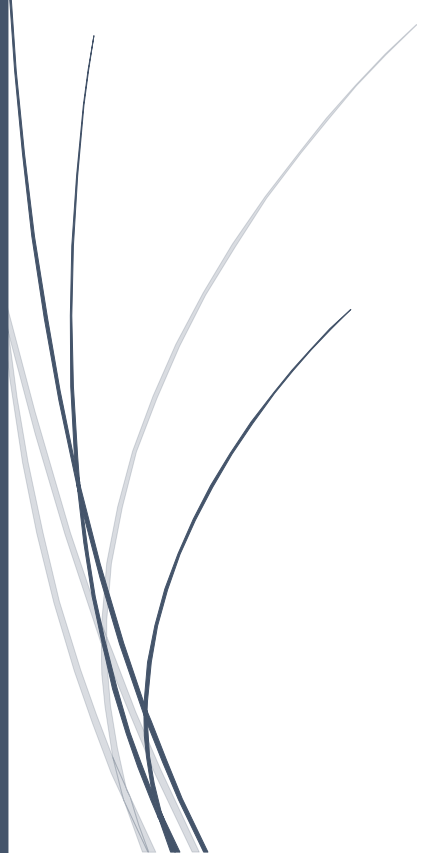




19.12.2021

# Платформа OMMG 3.0

Инструкция по эксплуатации



Георгий Хазан  
ОММДЖИ ТЕХНОЛОДЖИ

## Оглавление

Введение.....	2
Эксплуатация фронтенд-сервера.....	3
Эксплуатация сервера базы данных.....	5
Эксплуатация файлового хранилища.....	7

# Введение.

Платформа ОММГ (далее **Платформа**) включает в себя три вида серверов:

- фронтенд-сервера;
- сервера базы данных;
- файловые хранилища.

В зависимости от выбранных условий и типа развертывания, сервера определённого типа могут размещаться на одних и тех же физических или виртуальных машинах, вплоть до того, что вся Платформа может быть развёрнута на одном и том же сервере. В этом случае рекомендации для различных типов серверов будут применяться к одному и тому же серверу.

Все сервера Платформы работают под управлением ОС Linux, для мониторинга которой стандартом де-факто является программа zabbix. Для неё в дальнейшем и будут приводиться все примеры, при использовании других средств мониторинга ответственность за применение описанных правил мониторинга возлагается на администраторов системы. Готовый шаблон ОММГ для zabbix (далее Шаблон) входит в поставку Платформы и может быть импортирован из файла **`/var/www/docs/ommg3.zabbix.xml`**.

Предполагается, что перед изучением этого документа была проведена соответствующая установка Платформы в соответствии с документом "Платформа ОММГ 3.0. Инструкция по развёртыванию".

## Фронтенд-сервер

При эксплуатации фронтенд-сервера основное внимание уделяется трём важнейшим параметрам:

- нагрузка на сеть;
- нагрузка на память;
- нагрузка на процессор.

Т.к. в процессе работы фронтенд практически не использует жёсткий диск, то никаких особенных требований к дискам для серверов этого типа не применяется.

## Программное обеспечение

ПО, развёрнутое на фронтенде, состоит из четырёх главных частей:

- nginx, являющийся брокером SSL и распределителем нагрузки между внешней сетью и локальными серверами приложений;
- Redis, являющийся высоконагруженным кэшем данных для серверов приложений (работает на порту 6301 и должен быть привязан к localhost для установки на один сервер или к сетевому интерфейсу ЛВС для многомашинных установок);
- основной сервер приложений Платформы (`/var/www/ommg3_nodejs/app.bin`), работающий по протоколу HTTP на локальном порту 60000;
- вспомогательный сервер приложений Платформы (`/var/www/ommg3_nodejs/storage.bin`), работающий на портах 58000-58100, которые должны быть доступны по сетевому интерфейсу ЛВС для многомашинных установок.

## Мониторинг

Мониторинг nginx производится в zabbix посредством стандартного пресета правил, который можно скачать с <https://share.zabbix.com/cat-app/web-servers/nginx-template-for-zabbix-3-4>. Основной упор нужно сделать на следующих моментах:

- наличие и работоспособность всех рабочих процессов nginx (по умолчанию – 4, Шаблон → `OMMG_Nginx_worker_count`);
- суммарная нагрузка на процессор для всех процессов nginx не свыше 330% CPU (Шаблон → `OMMG_Nginx_cpu_usage`);
- суммарное количество HTTPS-запросов не свыше 10000 в секунду (Шаблон → `OMMG_Nginx_requests`).

Мониторинг Redis производится с помощью стандартного пресета правил, который можно скачать с [https://share.zabbix.com/databases/db\\_redis/redis-for-zabbix-3-2](https://share.zabbix.com/databases/db_redis/redis-for-zabbix-3-2). Главным параметром для мониторинга Redis является `memory commit` – не свыше 2GB (Шаблон →

OMMG\_Redis\_memory\_usage), также следует проверять нагрузку на процессор – не выше 80% CPU (Шаблон → OMMG\_Redis\_cpu\_usage).

Мониторинг серверов приложений производится для процессов, порождённых Node.js. Особое внимание обращается на следующие параметры:

- наличие и работоспособность всех рабочих процессов вспомогательного сервера приложений (по умолчанию – 4, Шаблон → OMMG\_Node\_Storagejs\_count);
- наличие и работоспособность всех рабочих процессов основного сервера приложений (по умолчанию – 16, Шаблон → OMMG\_Node\_Workerjs\_count);
- средняя нагрузка на процессор не выше 92% на один рабочий процесс (Шаблон → OMMG\_Node\_Storagejs\_perprocess\_cpu, Шаблон → OMMG\_Node\_Workerjs\_perprocess\_cpu);
- отсутствие в логе сервера приложений (/var/log/php\_log/node.log) фатальных ошибок, приводящих к перезапуску рабочих потоков (Шаблон → OMMG\_Node\_log).

Также можно контролировать суммарную нагрузку на процессор, порождаемую основным (Шаблон → OMMG\_Node\_Workerjs\_total\_cpu) и вспомогательным (Шаблон → OMMG\_Node\_Storagejs\_total\_cpu) серверами приложений.

## Масштабируемость и отказоустойчивость

Все фронтенды Платформы объединяются в один кластер с помощью файлов конфигурации, где указывается полный список IP серверов, несущих фронтенды, а также индекс данного сервера в этом списке (индекс первого элемента равняется нулю):

```
"serverIps": ["xxx.xxx.xxx.1", "xxx.xxx.xxx.2"],  
"serverId": 0,
```

При выходе из строя одного из серверов, входящих в этот список, Платформа прекращает его использование, но при этом теряется возможность использования кэша Redis или вспомогательного сервера приложений storage.bin, находящегося на повреждённом сервере. Чтобы оставшиеся сервера не испытывали перебоев с работой кэша, необходимо перенести IP-адрес повреждённого сервера на один из работающих с помощью аппаратуры Cisco или пакета usarp, входящего в дистрибутив Linux. В этом случае формирование кэша будет продолжаться на уцелевшем сервере, и работоспособность Платформы не нарушится.

## Серверы базы данных

При эксплуатации сервера базы данных основное внимание уделяется трем важнейшим параметрам:

- нагрузка на сеть;
- нагрузка на диск;
- нагрузка на процессор.

## Программное обеспечение

В качестве СУБД в Платформе используется MySQL версии 5.7 или любая совместимая с ней.

По умолчанию Платформа ставится на 19 баз данных:

- база для глобальных настроек (db);
- база статистики (dbstats);
- база стикеров (stickers);
- 16 независимых друг от друга баз данных (шардов), внутри которых хранится вся информация о пользователях Платформы, включая переписку и списки контактов.

## Мониторинг

Мониторинг сервера MySQL производится с помощью стандартного пресета правил, который можно скачать с <https://share.zabbix.com/databases/mysql/template-mysql-800-items>. Особое внимание нужно обратить на следующие показатели:

- суммарное количество запросов не свыше 3000 в секунду на один экземпляр MySQL;
- суммарное количество запросов INSERT/UPDATE/DELETE не свыше 2000 в секунду на один экземпляр MySQL;
- суммарная загрузка процессора не свыше 300% (Шаблон → OMMG\_Mysql\_cpu\_usage);
- свободное место на разделе для баз данных не менее 5% от общего размера раздела, если на этом сервере нет файлового хранилища, и не менее 10% от общего размера, если на этом же сервере есть файловое хранилище.

## Масштабируемость и отказоустойчивость

Платформа спроектирована таким образом, что различные шарды могут обслуживаться разными серверами баз данных, допуская таким образом горизонтальное масштабирование Платформы. Номер шарда для пользователя вычисляется как остаток от деления хэша логина на число серверов БД, таким образом в приведенном примере

при использовании двух серверов БД чётные шарды будут попадать на сервер db2 из файла конфигурации, а нечётные – на сервер db1:

```
'shards': {  
  'hosts': [  
    'mysql://db2.ommg.local:3306/',  
    'mysql://db1.ommg.local:3306/',  
  ]  
}
```

Если нагрузка на сервера базы данных превышает критическую, то на каждом сервере БД можно запустить ещё по одному экземпляру MySQL, что позволяет удвоить нагрузку на базу данных из-за того, что MySQL не очень интенсивно использует процессор. При этом добавочный экземпляр заводится на порт 3307, после чего файл конфигурации примет вид

```
'shards': {  
  'hosts': [  
    'mysql://db2.ommg.local:3307/',  
    'mysql://db1.ommg.local:3307/',  
    'mysql://db2.ommg.local:3306/',  
    'mysql://db1.ommg.local:3306/',  
  ]  
}
```

Штатной рекомендацией по резервированию информации, если должностной инструкцией не предусматривается иное, является репликация шардов крест-накрест (т.е. чётные шарды реплицируются на сервер db1, а нечётные – на сервер db2), как это описано в документе "Платформа OMMG 3.0. Инструкция по репликации данных". При этом достигается главное условие устойчивой работы: наличие только одного места, куда производится запись данных, что позволяет избегать конфликтов при вставке новых записей и изменении существующих.

При выходе из строя одного из серверов БД его ip-адрес может быть передан на другой сервер БД, на который ранее производилась репликация данных. Перемещение ip-адреса осуществляется автоматически с помощью аппаратуры Cisco или с помощью пакета usarp, входящего в дистрибутив Linux. В этом случае реплика базы данных начинает работать как основной сервер БД до тех пор, пока не восстановится работоспособность первоначального сервера, тогда с помощью обратной репликации данных можно восстановить на нем точную копию всех шардов с резервного сервера и впоследствии переключить ip-адрес обратно, чтобы восстановить балансировку нагрузки.

## Файловое хранилище.

Особенностью файлового хранилища является периодическая необходимость конвертации аудио- и видеофайлов в стандартный формат, который поддерживается всеми мобильными платформами, поэтому периодически может возникать пиковая нагрузка на процессор при перекодировании аудио- и видеофайлов.

Основными параметрами, которые отслеживаются на файловом хранилище, являются:

- нагрузка на диски;
- нагрузка на сеть;
- нагрузка на процессор.

На одном сервере с файловым хранилищем может располагаться сервисное ПО, входящее в состав Платформы и написанное на php: консоль администратора, сайт статистики, сайт раздачи обновлений и другого статического контента.

## Программное обеспечение

Файловое хранилище Платформы сделано на базе php-fpm (работает на локальном порту 9001), обрабатывающего HTTPS-запросы к виртуальным URL. Внешним брокером SSL для php является nginx, он же обрабатывает запросы за заливку и скачивание статического контента.

## Мониторинг

Нагрузка на диски является основным видом нагрузки файлового хранилища. Необходимо следить, чтобы оставалось не менее 10% свободного места на разделе, который предназначен для хранения файлов (символическая ссылка или хардлинк на `/var/www/ommg3_files/upload`), особенно если специфика клиента позволяет пользователям хранение файлов большого размера.

Мониторинг nginx производится в zabbix посредством стандартного пресета правил, который можно скачать с <https://share.zabbix.com/cat-app/web-servers/nginx-template-for-zabbix-3-4>. Основной упор нужно сделать на следующих моментах:

- наличие и работоспособность всех рабочих процессов nginx (по умолчанию – 4, Шаблон → OMMG\_Nginx\_worker\_count) ;
- суммарная нагрузка на процессор для всех процессов nginx не выше 330% CPU (Шаблон → OMMG\_Nginx\_cpu\_usage) ;
- суммарное количество HTTPS-запросов не выше 10000 в секунду (Шаблон → OMMG\_nginx\_requests).



Мониторинг php производится в zabbix посредством стандартного пресета правил, который можно скачать с <https://share.zabbix.com/cat-app/web-servers/php-fpm-template-for-zabbix-3-4>. Также необходимо проверить:

- наличие и работоспособность рабочих процессов php (по умолчанию – 2, Шаблон → OMMG\_php\_fpm\_process\_count);
- в сумме процессы php не должны потреблять больше 180% процессора (Шаблон → OMMG\_php\_fpm\_cpu\_usage).

Также можно проверять число одновременных соединений с php (Шаблон → OMMG\_php\_fpm\_connections).