

В очередной раз пришлось повозиться с настройкой Bitrixenv и сайта на нем. В какой-то момент bitrix сайт стал сыпать 500-е ошибки на некоторые операции. По логам было видно, что не хватает памяти для работы некоторых скриптов, хотя раньше хватало. Пришлось заняться расследованием и оптимизацией потребления памяти bitrix сайтом.

Если у вас есть желание научиться строить и поддерживать высокодоступные и надежные системы, рекомендую познакомиться с **онлайн-курсом «DevOps практики и инструменты»** в OTUS. Курс не для новичков, для поступления нужно пройти .

Содержание:

- 1 Цели статьи
- 2 Введение
- 3 Изменение стандартных настроек BitrixVM
- 4 Оптимизация настроек Mysql
- 5 Оптимизация настроек apache в bitrixenv
- 6 Оптимизация php под bitrix
- 7 Настройка nginx для сайта bitrix
- 8 Заключение

Цели статьи

1. Разобраться с потреблением памяти на сервере с bitrixenv - выяснить, кто больше всех потребляет памяти и приводит к нестабильной работе сервера.
2. Разобраться, где хранятся настройки различных приложений в bitrixenv.
3. Выбрать оптимальные параметры для apache, mysql, php, nginx для равномерного распределения памяти.

Введение

Вопрос с потреблением памяти mysql при работе в bitrixenv я уже разбирал отдельно некоторое время назад - где хранятся настройки mysql. Рекомендую с ней ознакомиться, так как там информация напрямую относящаяся к текущей теме оптимизации использования памяти сайта на bitrix при работе в bitrixenv.

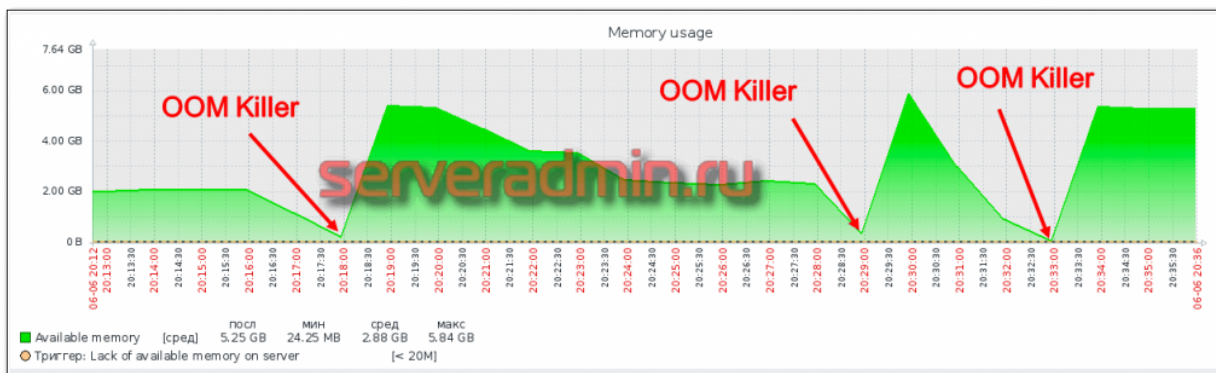
Разработчики bitrixenv упростили работу системных администраторов по настройке сервера, внедрив службу **bvat**, которая автоматически при запуске сервера подбирает оптимальные параметры следующих служб:

- mysql
- apache
- nginx
- php

Настройки будут зависеть от количества доступной оперативной памяти. В целом, это неплохой шаг, который упрощает начальную настройку сервера. Чаще всего конфигурация служб получается адекватной и подходящей для типовых сайтов.

В моем случае стандартные настройки перестали подходить. На сервере время от времени появлялась нехватка оперативной памяти. Приходил **OOM Killer** (OOM — Out of memory) и грохал mysql сервер, так как он потреблял больше всего оперативной памяти. Какое-то время все работало нормально, потом повторилось то же самое.

Мое внимание привлекли события из мониторинга Zabbix, такие как **Lack of available memory on server**. Посмотрел график и все сразу стало ясно, еще до подключения к серверу.



Зашел на сервер, посмотрел системный лог. Увидел там вот это:

```
kernel: Out of memory: Kill process 7382 (mysqld) score 431 or sacrifice child
kernel: Killed process 7382 (mysqld) total-vm:3967860kB, anon-rss:1942144kB, file-rss:0kB, shmem-rss:0kB
systemd: mysql.service: main process exited, code=killed, status=9/KILL
systemd: Unit mysql.service entered failed state.
systemd: mysql.service failed.
systemd: mysql.service holdoff time over, scheduling restart.
systemd: Stopped MySQL Server.
systemd: Starting MySQL Server...
```

```
systemd: Started MySQL Server.
```

Первое, что я сделал - **увеличил swap** раздел до объема всей оперативной памяти. До этого он был размером в 1G. Это сразу помогло и предотвратило регулярный приход OOM Killer. А я стал спокойно разбираться, что делать дальше.

План дальнейшей настройки сервера для стабильной работы сайта на bitrix следующий:

1. Определяем основных потребителей оперативной памяти.
2. Распределяем всю свободную память между ними.
3. Убеждаемся, что под нагрузкой все работает корректно, всем хватает памяти, OOM Killer не приходит.

Изменение стандартных настроек BitrixVM

Как я уже говорил, служба bvat автоматически регулирует некоторые настройки стандартных служб bitrixenv. Чтобы применять наши настройки, нужно их указывать в отдельных конфигурационных файлах.

- MySQL - /etc/mysql/conf.d/z_bx_custom.cnf
- Apache - /etc/httpd/bx/custom/z_bx_custom.conf
- nginx - /etc/nginx/bx/conf/z_bx_custom.conf
- PHP - /etc/php.d/z_bx_custom.ini

А вот общий список всех основных конфигурационных файлов bitrixenv:

- /etc/php.d/bitrixenv.ini - основные настройки php
- /etc/httpd/bx/conf/prefork.conf - параметры модуля Apache - MPM prefork;
- /etc/php.d/z_bx_custom.ini - пользовательские настройки PHP;
- /etc/httpd/bx/custom/z_bx_custom.conf - пользовательские настройки Apache;
- /etc/mysql/conf.d/z_bx_custom.cnf - пользовательские настройки MySQL;
- /etc/nginx/bx/conf/z_bx_custom.conf - пользовательские настройки nginx;
- /etc/nginx/bx/conf/push-im_settings.conf - настройки nginx-push-stream-module.

Оптимизация настроек Mysql

На подопытном сервере имеется 12 Гб оперативной памяти. Я решил половину этой памяти отдать под mysql. Приступим к тюнингу конфигурации mysql. В общем случае достаточно будет одного параметра, который в основном отвечает за потребление памяти:

```
innodb_buffer_pool_size = 4G
```

В моем случае этого было недостаточно. Я решил более внимательно подойти к настройке mysql. Нашел неплохой инструмент - MySQLTuner, который анализируя работу mysql, выдает некоторые рекомендации по настройке. Сам я не разбираюсь в тонкой настройке mysql, поэтому решил довериться утилите. Судя по отзывам, она неплоха и доверять ей можно, если сам не разбираешься в теме. Забегая вперед скажу, что с помощью этого тюнера я настроил mysql на стабильную работу с фиксированным потреблением памяти. Проблем с этим сервером с тех пор не возникало.

Итак, копируем себе на сервер сам скрипт:

```
# wget http://mysqltuner.pl/ -O mysqltuner.pl
```

Запускаем его:

```
# perl mysqltuner.pl
```

Для того, чтобы рекомендации получились более эффективные, служба mysql должна поработать у вас несколько дней. Если накануне перезапускали ее, а я это делал, то рекомендую через несколько дней зайти и еще раз прогнать тесты. Будут новые советы по конфигу.

Для оптимизации потребления памяти, достаточно будет прогнать скрипт в любое время. Я вам рекомендую внимательно изучить его возможности. Подробно на них я сейчас не буду останавливаться, а рассмотрю только то, что касается памяти. Помимо прочего, вы увидите следующую информацию.

```
----- Performance Metrics -----  
[--] Up for: 7d 10h 41m 43s (147M q [228.866 qps], 3M conn, TX: 386G, RX: 43G)  
[--] Reads / Writes: 88% / 12%  
[--] Binary logging is disabled  
[--] Physical Memory      : 11.6G  
[--] Max MySQL memory    : 6.8G  
[--] Other process memory: 0B  
[--] Total buffers: 4.3G global + 36.8M per thread (70 max threads)  
[--] P_S Max memory usage: 72B  
[--] Galera GCache Max memory usage: 0B  
[OK] Maximum reached memory usage: 6.4G (55.59% of installed RAM)  
[OK] Maximum possible memory usage: 6.8G (58.69% of installed RAM)  
[OK] Overall possible memory usage with other process is compatible with memory available
```

У меня уже все оптимизировано под потребление не более примерно 6 Гб памяти. Расскажу, какие параметры за это отвечают. Как уже сказал ранее, это параметр **innodb_buffer_pool_size**. В общем случае для mysql сервера рекомендуют указывать этот параметр равный 80% доступной памяти сервера. Но это в том случае, если у вас кроме mysql на этом сервере ничего не крутится. А у нас там полно других служб, поэтому нам такой совет не подходит.

Дальше нам нужно выяснить, сколько памяти занимает thread (процесс, который порождает соединение) и в соответствии с этим выставить предел числа подключений. Размер thread равен сумме следующих параметров - `read_buffer_size` + `sort_buffer_size` + `join_buffer_size`.

Параметр `read_buffer_size` установлен по-умолчанию в 128 КБ. Я его не стал трогать. Остальные два я изначально выставил по рекомендациям `mysq tuner`, а значение `max_connections`, которое отвечает за максимальное количество подключений, выставил такое, чтобы сумма трех буферов, помноженная на количество подключений не превышала 2 Гб памяти. Сервер немного поработал в таком режиме и выяснилось, что выставленных подключений не хватает. Тогда я снизил `join_buffer_size` до 18 Мб, а количество подключений увеличил. В итоге остановился на таких настройках.

```
innodb_buffer_pool_size = 4G  
sort_buffer_size = 18M  
join_buffer_size = 18M
```

```
max_connections = 70
```

С такими настройками максимальное потребление памяти службой mysql не будет превышать 6.8 Гб, о чем подсказывает вывод mysqltuner. Конкретно моему сайту 70 подключений к mysql достаточно. До этого поставил 50, были сообщения о нехватке подключений. На своем сервере выбирайте параметры сами, у меня не копируйте.

```
[OK] Maximum possible memory usage: 6.8G (58.69% of installed RAM)
```

На практике так и получилось. Через несколько дней я зашел и прогнал еще раз проверку, которая показала, что реально использование памяти не вышло за эти пределы. Плюс, подредактировал некоторые параметры.

Советы по изменению параметров даются в заключительной секции mysqltuner - **Variables to adjust**. Не буду приводить свои рекомендации, так как они будут актуальны только для конкретного сервера. Советую посмотреть все рекомендации, почитать описание параметров и попробовать применить их у себя. Слепо не надо менять то, что там советуют.

Приведу список основных параметров mysql, которые влияют на производительность и на которые надо в первую очередь обращать внимание:

- max_connections
- log_bin
- table_open_cache_size
- table_definitions_cache_size
- open_files_limit
- innodb_buffer_pool_size
- innodb_log_file_size
- innodb_flush_log_at_trx_commin
- innodb_flush_method=O_DIRECT

Список взял отсюда. Очень полезная статья, рекомендую.

Оптимизация настроек apache в bitrixenv

Дальше переходим ко второму основному потребителю оперативной памяти на сервере с сайтом на bitrix - apache. Ему, как и для mysql, служба bvat автоматически выставляет некоторые настройки. Она хранятся в файле `/etc/httpd/bx/conf/prefork.conf`. Нас будут интересовать настройки, касающиеся количества запущенных процессов.

Чтобы узнать, количество запущенных процессов httpd, обслуживающих работу bitrix сайта, введите в консоли сервера команду:

```
# ps ax | grep httpd | wc -l
```

Вы получите число, на 2 больше, чем указано в приведенном конфиге, в параметрах модуля **mpm_prefork**. В моем случае bvat выставлял максимально возможное количество процессов httpd равное 60, но для меня это было слишком много, сервер не тянул такое количество процессов. Я его уменьшил до 30.

Как вы понимаете, в зависимости от bitrix сайта, один процесс httpd будет использовать разное количество памяти, поэтому автоматически невозможно выставить этот параметр корректно для всех сайтов. В данном случае, дефолтный параметр мне не подошел, поэтому я создал свой файл настроек httpd - `/etc/httpd/bx/custom/z_bx_custom.conf`.

```
#
# Define custom settings
#
serveradmin.ru
<IfModule mpm_prefork_module>
  StartServers      30
  MinSpareServers  30
  MaxSpareServers  30
  MaxRequestWorkers 30
  MaxRequestsPerChild 3000
</IfModule>
```


Привожу скриншотом, потому что движок сайта проглатывает все строки в угловых скобках. Не получается выложить полностью конфиг в текстовом виде. В общем случае, вам надо посмотреть, сколько у вас занимает памяти один процесс httpd и рассчитать максимальное количество процессов, которое потянет ваш сервер.

Посмотреть, сколько памяти занимает один процесс httpd можно в htop или с помощью команды:

```
# ps -o vsz, rss, cmd --pid $(pgrep httpd)
```

```
[root@serveradmin custom]# ps -o vsz,rss,cmd --pid $(pgrep httpd)
VSZ  RSS  CMD
1981704 536776 /usr/sbin/httpd -DFOREGROUND
2004780 100564 /usr/sbin/httpd -DFOREGROUND
2009544 105644 /usr/sbin/httpd -DFOREGROUND
2012604 109064 /usr/sbin/httpd -DFOREGROUND
2023984 112324 /usr/sbin/httpd -DFOREGROUND
2005476 101236 /usr/sbin/httpd -DFOREGROUND
2010648 109312 /usr/sbin/httpd -DFOREGROUND
2010860 105928 /usr/sbin/httpd -DFOREGROUND
2009396 102776 /usr/sbin/httpd -DFOREGROUND
2012352 106700 /usr/sbin/httpd -DFOREGROUND
2009456 103312 /usr/sbin/httpd -DFOREGROUND
2010256 107320 /usr/sbin/httpd -DFOREGROUND
2009448 104144 /usr/sbin/httpd -DFOREGROUND
2009404 99044 /usr/sbin/httpd -DFOREGROUND
2011936 118280 /usr/sbin/httpd -DFOREGROUND
2012468 113860 /usr/sbin/httpd -DFOREGROUND
2010764 96964 /usr/sbin/httpd -DFOREGROUND
2006360 91304 /usr/sbin/httpd -DFOREGROUND
2009580 103092 /usr/sbin/httpd -DFOREGROUND
2012280 106236 /usr/sbin/httpd -DFOREGROUND
2009272 105688 /usr/sbin/httpd -DFOREGROUND
2012312 99524 /usr/sbin/httpd -DFOREGROUND
1998876 90884 /usr/sbin/httpd -DFOREGROUND
2010968 105248 /usr/sbin/httpd -DFOREGROUND
2009472 105052 /usr/sbin/httpd -DFOREGROUND
2010968 105232 /usr/sbin/httpd -DFOREGROUND
2012660 104248 /usr/sbin/httpd -DFOREGROUND
2010892 102348 /usr/sbin/httpd -DFOREGROUND
1999900 99476 /usr/sbin/httpd -DFOREGROUND
2009336 95000 /usr/sbin/httpd -DFOREGROUND
2010184 107360 /usr/sbin/httpd -DFOREGROUND
```

Будет один основной процесс, который занимает больше всего памяти и дальше его форки, которые потребляют примерно одинаково. На них и ориентируйтесь. У меня основной процесс потребляет 500 Мб и 30 форков по 100 Мб. В сумме получается 3.5 Гб.

Итого в пике у меня 6.5 Гб использует mysql и 3.5 Гб использует httpd, итого 10 Гб из доступных 12-ти. На практике, свободной памяти обычно больше, чем 2 Гб, так как mysql чаще всего потребляет ниже максимального предела.

Оптимизация php под bitrix

Из настроек php я бы обратил внимание на следующие параметры:

- **memory_limit** - максимальное количество памяти на выполнение php скрипта;
- **sendmail_path** - управляет параметрами отправки сообщений, хотя к теме текущей статьи и не имеет отношение;
- **post_max_size** - максимальный размер данных для всего POST запроса;
- **upload_max_filesize** - максимальный размер файла для загрузки через POST запрос;
- **max_execution_time** - максимальное время в секундах, в течение которого скрипт должен полностью загрузиться.

Так или иначе, эти параметры, кроме sendmail, влияют на производительность сервера и потребление памяти. Не ставьте эти значения слишком большими без особой надобности. Я бы для начала выставил в 256 Мб и увеличивал по мере необходимости. Да, 256 Мб это и так очень много, но сайт на bitrix требует высоких значений этих параметров для корректной работы. 256 мб это общая рекомендация для дефолтных значений.

Свои параметры php вы можете размещать в отдельном конфиге, который не будет перетираться bitrixenv - `/etc/php.d/z_bx_custom.ini`. После изменения настроек надо перезапустить apache для применения.

Настройка nginx для сайта bitrix

В самом nginx в bitrixenv настраивать для производительности особо нечего. Он работает в качестве проху сервера для apache. С помощью проху_pass он перенаправляет все динамические запросы, а сам отдает только статику. В таком режиме работы он потребляет минимум ресурсов и оптимизировать в нем нечего. Если вам все же интересно разобраться в настройках nginx, то читайте мою отдельную подробную статью.

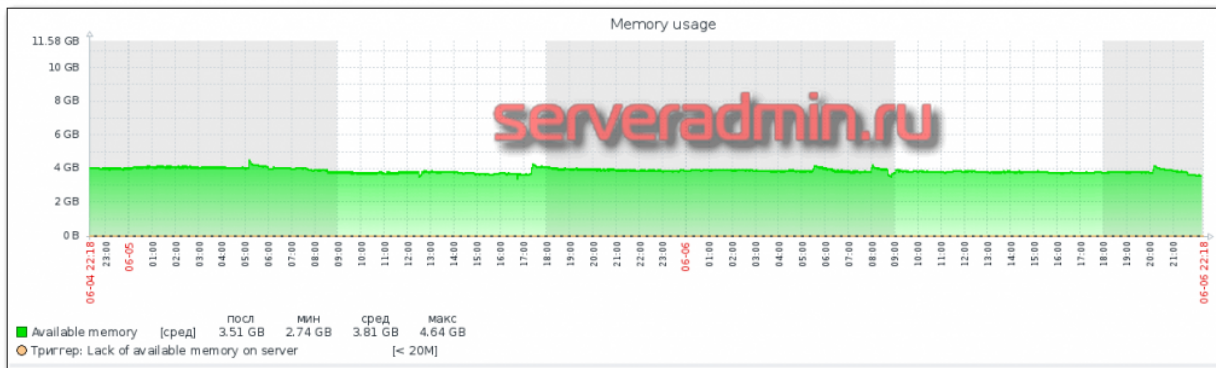
Отдельной настройки требует только модуль **Push and Pull**, если он у вас используется. Его конфигурация располагается в файле `/etc/nginx/bx/conf/push-im_settings.conf`. В контексте данной статьи нас интересует только параметр **push_stream_shared_memory_size**, который отвечает за использование оперативной памяти.

В принципе, дефолтного значения 256 Мб обычно хватает, хотя по сути это небольшие цифры. Но имейте ввиду, что если свободной памяти совсем нет, то можно подрезать этот параметр.

Заключение

Не понравилась статья и хочешь научить меня администрировать? Пожалуйста, я люблю учиться. Комментарии в твоём распоряжении. Расскажи, как сделать правильно!

После оптимизации всех указанных выше параметров в bitrixenv, потребление памяти сервером стабилизировалось. Bitrix сайт стал работать ровно с предсказуемой производительностью без неожиданных тормозов и падений.



На этом у меня все по теме оптимизации настроек сервера под bitrix. Система интересная и многогранная. Всегда любопытно заглянуть под капот bitrixenv. Как по мне, сделано неплохо, хотя и доставляет хлопот при разборе каких-то инцидентов.

В целом считаю, что в общем случае, все сделано удобно и функционально для быстрого запуска bitrix сайта. Справится даже неподготовленный человек, а конкретно какой-нибудь программист. Он бы запарился настраивать эту связку самостоятельно, а тут все из коробки работает. Но вот если возникают проблемы, то разобраться бывает не всегда просто.

Следующим этапом жду появление docker сборок с bitrixenv внутри. Либо один общий образ, либо набор через docker-compose. Это было бы логичное продолжение развития в свете популярности контейнеров и микросервисов.

Онлайн курс по Linux

Если у вас есть желание освоить операционную систему Linux, не имея подходящего опыта, рекомендую познакомиться с **онлайн-курсом Administrator Linux. Basic** в OTUS. Курс для новичков, адаптирован для тех, кто только начинает изучение Linux. Обучение длится 4 месяца. Что даст вам этот курс:

- Вы получите навыки администрирования Linux (структура Linux, основные команды, работа с файлами и ПО).
- Вы рассмотрите следующий стек технологий: Zabbix, Prometheus, TCP/IP, nginx, Apache, MySQL, Bash, Docker, Git, nosql, grafana, ELK.
- Умение настраивать веб-сервера, базы данных (mysql и nosql) и работа с сетью.
- Мониторинг и логирование на базе Zabbix, Prometheus, Grafana и ELK.
- Научитесь командной работе с помощью Git и Docker.

Смотрите подробнее программу по .

Помогла статья? Подписывайся на telegram канал автора

Анонсы всех статей, плюс много другой полезной и интересной информации, которая не попадает на сайт.