

НОВЫЕ ВОЗМОЖНОСТИ **Mamonsu 3.0**

Александра Кузнецова
Postgres Professional, 2021

Mamonsu – активный агент мониторинга для PostgreSQL, исполненный на базе Zabbix.

- Написан на Python3
- Передача данных через *zabbix trapper*
- Можно добавлять свои кастомные плагины
- ~40 системных метрик, ~90 метрик PostgreSQL
- Дополнительные инструменты для настройки СУБД, получения отчетов о состоянии системы и др.

<https://github.com/postgrespro/mamonsu>

 Mamonsu 3.0 несовместима с предыдущими версиями

Что изменилось?

- Новое название шаблона
- Изменились способы вычисления некоторых метрик
- Оптимизированы графики
- Добавлены новые метрики для PostgreSQL 14
- Добавлены скрины
- Усовершенствован механизм bootstrap
- Усовершенствована поддержка Zabbix
- Добавлено тестирование

Название шаблона

- Текущее название не отражало сути шаблона;
- Шаблон было трудно искать после его загрузки;

PostgresPro-OS  Mamonsu PostgreSQL OS

Изменение метрик: PostgreSQL transactions committed

PostgreSQL transactions total  PostgreSQL transactions committed

```
select
sum(xact_commit) from
pg_catalog.pg_stat_database;
```

Изменение метрик: PostgreSQL uptime

Name ▲	Last check	Last value
App-PostgresPro-Linux (1 Item)		
PostgreSQL: service uptime	2021-10-19 15:38:32	06:05:34

seconds



Name ▲	Last check	Last value
Mamonsu PostgreSQL Linux (1 Item)		
PostgreSQL: service uptime	2021-10-19 15:19:02	2021-10-19 09:32:58

unixtime

Изменение метрик: PostgreSQL cache hit ratio

В предыдущем варианте метрика рассчитывалась на основании данных за все время работы СУБД. Теперь расчет выполняется в Zabbix по приращениям *blocks hit* и *blocks read*.

```
select
round(sum(blks_hit)*100/
sum(blks_hit+blks_read), 2)
from
pg_catalog.pg_stat_database;
```



* Name	PostgreSQL: cache hit ratio
Type	Calculated
* Key	pgsql.cache[hit]
* Formula	$\text{last}(\text{//pgsql.blocks[hit]}) * 100 / (\text{last}(\text{//pgsql.blocks[hit]}) + \text{last}(\text{//pgsql.blocks[read]})$
Type of information	Numeric (unsigned)
Units	%
* Update interval	60

Изменение метрик: Archive Command plugin

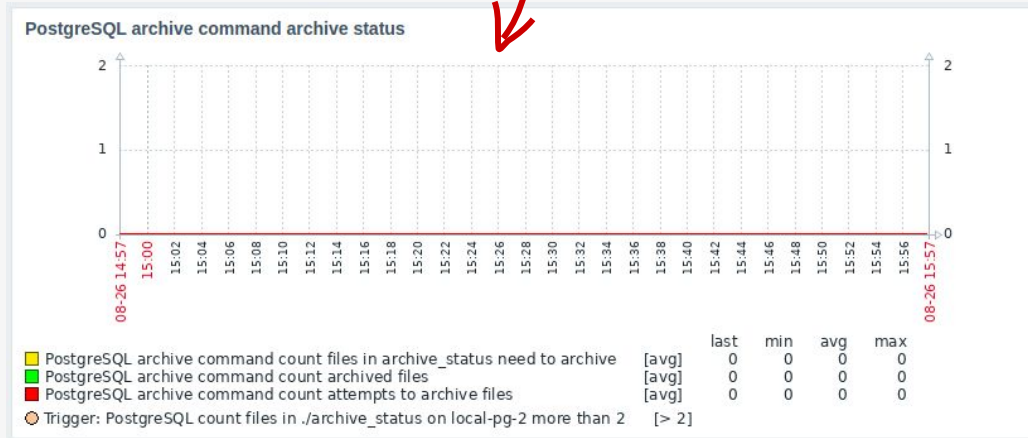
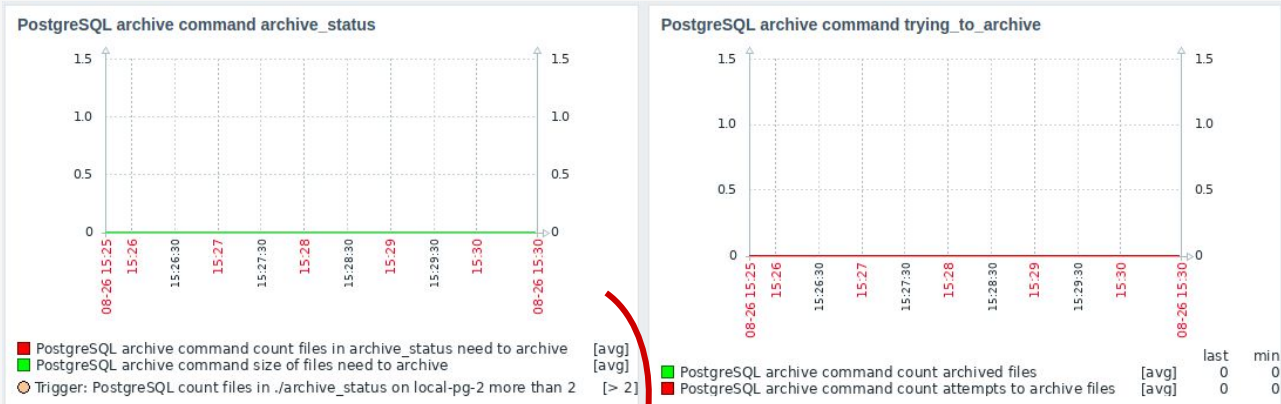
- Больше нет обращения к ФС;
- Считает только для master;

```
wal_segment_size  
+  
pg_stat_archiver.last_archived_wal  
+  
pg_current_wal_lsn()
```

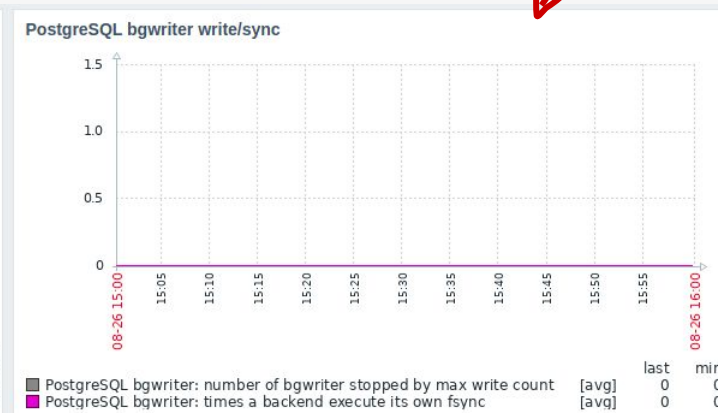
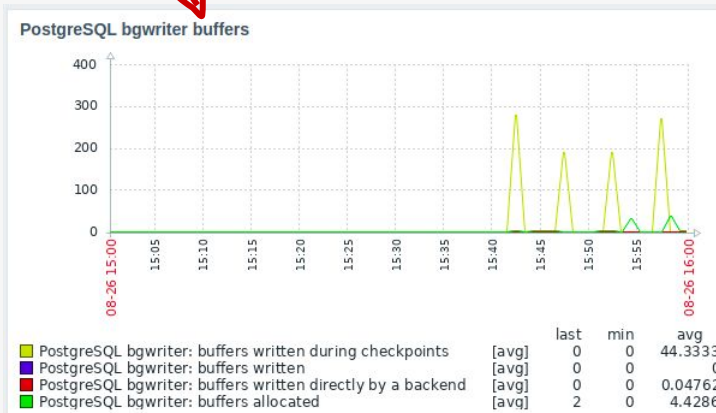
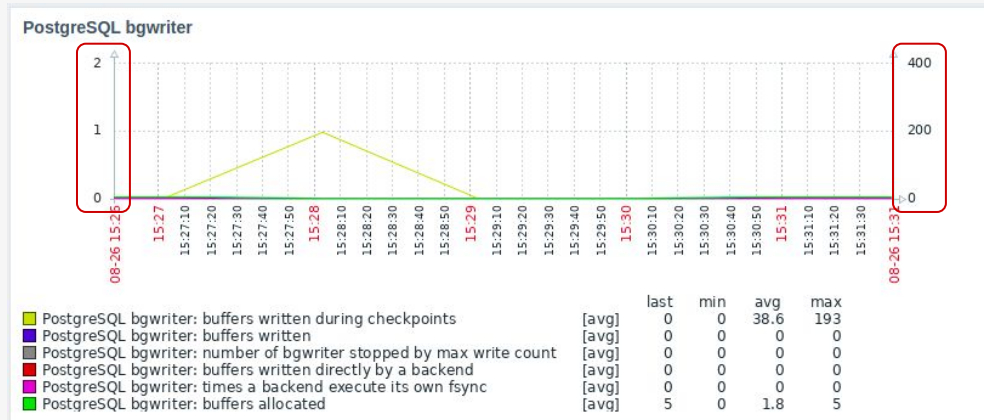

Удаление графиков

- PostgreSQL uptime
- PostgreSQL cfs compression: current ratio
- PostgreSQL cfs compression: compressed files
- PostgreSQL cfs compression: written bytes
- PostgreSQL cfs compression: total ratio
- PostgreSQL oldest transaction running time
- PostgreSQL age of oldest xid
- PostgreSQL number of parallel queries being executed now
- PostgreSQL write-ahead log generation speed
- PostgreSQL replication lag in second
- PostgreSQL count of xlog files
- System load average
- System: count of opened files
- System up_time

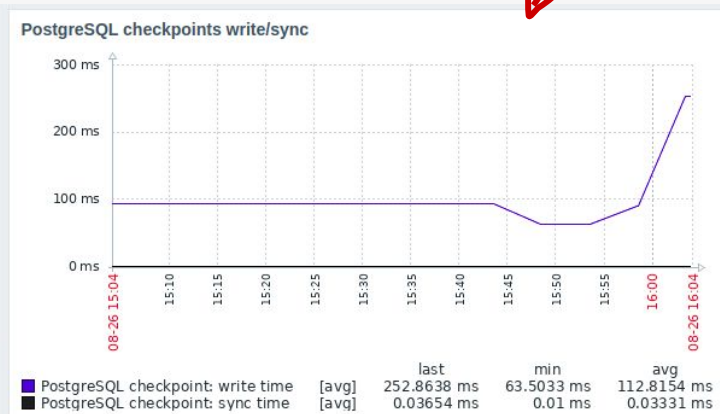
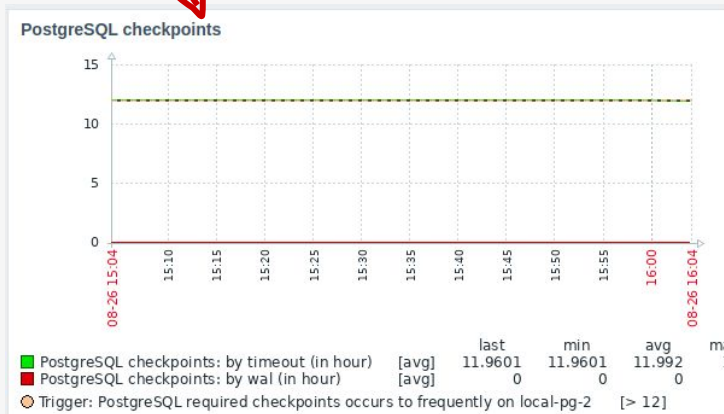
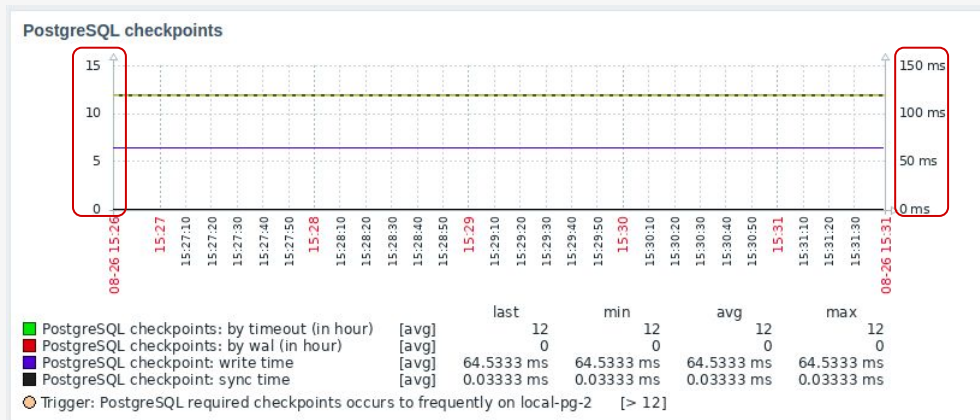
Изменение графиков: PostgreSQL archive command



Изменение графов: PostgreSQL bgwriter



Изменение графов: PostgreSQL checkpoints



Новые метрики: pg_stat_wal

pg_stat_wal
wal_records
wal_fpi
wal_bytes
wal_buffers_full
wal_write
wal_sync
wal_write_time
wal_sync_time
stats_reset

PostgreSQL: WAL records generated

key: `pgsql.wal.records.count[]`

delta: Speed Per Second

Количество сгенерированных за секунду записей WAL

PostgreSQL: WAL full page images generated

key: `pgsql.wal.fpi.count[]`

delta: Speed Per Second

Количество сгенерированных за секунду полных образов страниц WAL (full_page_writes = on)

PostgreSQL: WAL buffers full

key: `pgsql.wal.buffers_full`

delta: Speed Per Second

Количество случаев, когда данные журналов были принудительно записаны на диск из-за переполнения буферов → корректировка wal_buffers

Новые метрики: pg_stat_wal

pg_stat_wal
wal_records
wal_fpi
wal_bytes
wal_buffers_full
wal_write
wal_sync
wal_write_time
wal_sync_time
stats_reset

PostgreSQL: WAL write time (ms)

key: `pgsql.wal.write_time`

delta: Speed Per Second

Время, затраченное на запись WAL на диск

PostgreSQL: WAL sync time (ms)

key: `pgsql.wal.sync_time`

delta: Speed Per Second

Время, затраченное на синхронизацию данных WAL с данными на диске

PostgreSQL: WAL sync duty (%)

key: `pgsql.wal.sync_duty`

delta: Speed Per Second

Процент времени, затраченного на синхронизацию WAL с диском (fsync), за период сбора метрик

Новые метрики: pg_stat_statements_info

pg_stat_statements_info
dealloc
stats_reset

PostgreSQL: statements dealloc

key: `pgsql.stat_info[dealloc]`

delta: Simple Change

Количество случаев, когда из-за превышения `pg_stat_statements.max` была совершена очистка записей

PostgreSQL: statements last reset

key: `pgsql.stat_info[stats_reset]`

delta: As Is

Время последнего сброса статистики `pg_stat_statements`

Изменение структуры плагина

В функциях для формирования шаблона *items()*, *discovery_rules()*, *graphs()*, *triggers()* появился новый аргумент - *dashboard*.

Параметры:

'name' - item key/graph name
'page' - название скрина
'size' - размер виджета
'position' - положение на скрине

[documentation](#)

```
def items(self, template, dashboard=False):
    result = ''
    if self.Type == "mamonu":
        delay = self.plugin_config('interval')
        value_type = Plugin.VALUE_TYPE.numeric_unsigned
    else:
        delay = 5 # TODO check delay
        value_type = Plugin.VALUE_TYPE.numeric_float
    result += template.item({
        'name': 'PostgreSQL: ping',
        'key': self.right_type(self.key_ping),
        'value_type': Plugin.VALUE_TYPE.numeric_float,
        'units': Plugin.UNITS.ms,
        'delay': delay
    })
    if not dashboard:
        return result
    else:
        return [{'dashboard': {'name': self.right_type(self.key_ping),
                                'page':
ZbxTemplate.dashboard_page_instance['name'],
                                'size':
ZbxTemplate.dashboard_widget_size_medium,
                                'position': 1}}]
```

Новое во взаимодействии с Zabbix: поддержка разных версий

Поддерживаемые версии Zabbix: 3.0 - 5.4

Какие были сложности?

```
$ mamonsu zabbix export template template.xml
```

- Ошибки совместимости API различных версий Zabbix с Mamonsu;
- Исправлены косметические ошибки в инструменте *Zabbix CLI*;
- Не было аргументов *'deleteMissing'* для корректной процедуры обновления шаблона;

Новое в инструменте *bootstrap*: автоматическое создание пользователя и схемы

```
$ mamonsu bootstrap -U postgres -d mamonsu_db -M mamonsu_user
```

Пользователь по умолчанию: *mamonsu/mamonsu*, схема *mamonsu*

```
mamonsu_db=# \dn+ mamonsu
                List of schemas
  Name  | Owner  | Access privileges | Description
-----+-----+-----+-----
 mamonsu | mamonsu | mamonsu=UC/mamonsu |

```

```
mamonsu_db=# \df mamonsu.*
                List of functions
 Schema |          Name
-----+-----
 mamonsu | archive_command_files
 mamonsu | archive_stat
 mamonsu | buffer_cache
 mamonsu | count_autovacuum
 mamonsu | count_wal_files
<...>
```

Новое в инструменте *bootstrap*: `-x/--create-extensions`

```
$ mamonsu bootstrap -U postgres -d mamonsu_db
```

```
mamonsu_db=# \dx
List of installed extensions
-[ RECORD 1 ]-----
Name          | plpgsql
Version       | 1.0
Schema        | pg_catalog
Description   | PL/pgSQL procedural language
```

```
mamonsu_db=# \df mamonsu.buffer*
(0 rows)
```

```
$ mamonsu bootstrap -U postgres -d mamonsu_db -x
```

```
List of installed extensions
-[ RECORD 1 ]-----
Name          | pg_buffercache
Version       | 1.3
Schema        | mamonsu
Description   | examine the shared buffer cache
-[ RECORD 2 ]-----
Name          | plpgsql
Version       | 1.0
Schema        | pg_catalog
Description   | PL/pgSQL procedural language
```

```
mamonsu_db=# \df mamonsu.buffer*
List of functions
-[ RECORD 1 ]-----+-----
Schema          | mamonsu
Name            | buffer_cache
Type            | func
```

Частый вызов функций *pg_buffercache* может повлиять на производительность, поэтому необходимо было организовать опциональную установку расширений.

Table of Contents

Mamonsu: concepts

Requirements

Features

Metrics

Zabbix Screens

Tools

Build

Installation

Usage

Screenshots

Configuration

Template creation and upload

Run

Template update

Best practices

Additional chapters

Configuration

1. Optionally, bootstrap Mamonsu

If you omit this step, metrics can only be collected on behalf of a superuser, which is not recommended.

Create a non-privileged database user for Mamonsu. For example:

```
CREATE USER mamonsu_user WITH PASSWORD 'mamonsu_password';
```

Create a database that will be used for connection to PostgreSQL. For example:

```
CREATE DATABASE mamonsu_database OWNER mamonsu_user;
```

Run the following command to bootstrap Mamonsu:

```
mamonsu bootstrap [-M mamonsu_user] [-x | --create-extensions] [connection_options]
```

For details of usage, see "Tools".

As the result of this operation, monitoring functions are created in the `mamonsu_database` in `mamonsu` schema, and the right to execute them is granted to the `mamonsu_user`. Thus, a superuser connection is no longer required. Mamonsu also creates several tables in the specified database. Do not delete these tables as they are required for Mamonsu to work.

2. Configure Mamonsu

Edit the `agent.conf` configuration file.

Configure Zabbix-related settings. The address field must point to the running Zabbix server, while the client field must provide the name of the Zabbix host. You can find the list of hosts available for your account in the Zabbix web interface under Configuration > Hosts.

```
[zabbix]
; enabled by default
enabled = True
client = zabbix_host_name
address = zabbix_server
```

By default, Mamonsu will collect both PostgreSQL and system metrics. If required, you can disable metrics collection of either type by setting the `enabled` parameter to `False` in the `[postgres]` or `[system]` section of the `agent.conf` file, respectively.

Обновление Mamonsu

 Mamonsu 3.0.0 несовместима с предыдущими версиями

Не сохранять старые данные

1. Сгенерировать новый шаблон для Zabbix;
2. Запустить bootstrap ещё раз, если выполнялось для старой версии;
3. Загрузить новый шаблон на сервер Zabbix;
4. Привязать к узлу новый шаблон вместо старого.

Сохранить старые данные

1. Сгенерировать новый шаблон для Zabbix;
2. Запустить bootstrap ещё раз, если выполнялось для старой версии;
3. Загрузить новый шаблон на сервер Zabbix;
4. Переименовать узел, для которого необходимо сохранить данные, и оставьте привязку к старому шаблону;
5. Создать новый узел для той же машины и привязать к нему новый шаблон.

Обновление Mamonsu

Scripts

```
zabbix=# SELECT host, name FROM hosts
zabbix=# WHERE host LIKE '%local-pg%';
-[ RECORD 1 ]----
host | local-pg-2
name | local-pg-2
-[ RECORD 2 ]----
host | local-pg-3
name | local-pg-3

zabbix=# UPDATE hosts
zabbix=# SET host = host || ' OLD-MAMONSU',
zabbix=#     name = name || ' OLD-MAMONSU'
zabbix=# WHERE host LIKE '%local-pg%';
UPDATE 2
zabbix=# SELECT host, name FROM hosts
zabbix=# WHERE host LIKE '%local-pg%';
-[ RECORD 1 ]-----
host | local-pg-2 OLD-MAMONSU
name | local-pg-2 OLD-MAMONSU
-[ RECORD 2 ]-----
host | local-pg-3 OLD-MAMONSU
name | local-pg-3 OLD-MAMONSU
```

```
curl -H "Content-type: application/json-rpc" -X POST
http://zabbix/api_jsonrpc.php -d'
{
  "jsonrpc": "2.0",
  "method": "host.update",
  "params": {
    "hostid": "HOST_ID",
    "host": "Local-pg-3 OLD-MAMONSU",
    "name": "local-pg-3 OLD-MAMONSU"
  },
  "auth": "AUTH_TOKEN",
  "id": 1
}'

$ rename_zabbix_hosts.sh --url=http://localzabbix/
--pattern="local-pg" --suffix="OLD-MAMONSU"
```

Тестирование Mamonsu

Пример сборки

MASTER

- OS: Ubuntu 20.04, CentOS 7, CentOS 8;
- Zabbix: 4.0, 5.0, 5.4;
- PostgreSQL: 9.6, 10, 11, 12, 13, 14;

DEV

- OS: CentOS 7;
- Zabbix: 5.4;
- PostgreSQL: 9.6, 10, 11, 12, 13, 14;

- > Build and install Mamonsu
- > Test MAMONSU BOOTSTRAP
- > Test MAMONSU AGENT
- > Test MAMONSU REPORT
- > Test MAMONSU EXPORT
- > Test MAMONSU ZABBIX CLI
- > Test Mamonsu metrics on master

```
14 -----> mamonsu export config mamonsu.conf --add-plugins
15 Configuration file for mamonsu has been saved as mamonsu.conf
16 mamonsu.conf: ASCII text
17
18 -----> mamonsu export template template.xml --add-plugins
19 Template for mamonsu has been saved as template.xml
20 template.xml: XML 1.0 document, ASCII text, with very long lines
21
22 -----> mamonsu export zabbix-parameters zabbix.conf --add-
    plugins=/etc/mamonsu/plugins --config=/etc/mamonsu/agent.conf --pg-version=13
23 Configuration file for zabbix-agent has been saved as zabbix.conf
24 Bash scripts for native zabbix-agent have been saved to //scripts
25 zabbix.conf: ASCII text, with very long lines
26
```


Спасибо за внимание

Вопросы?