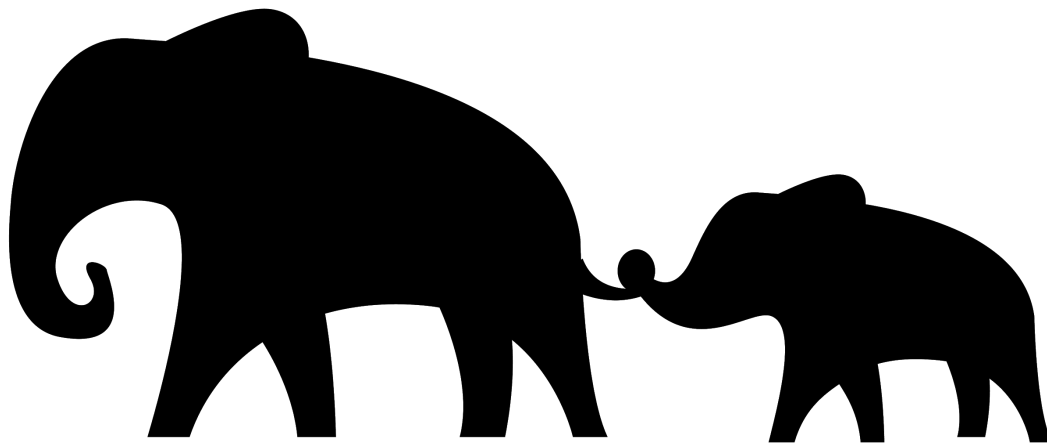




А.И. Попов
Д.В. Вилкова

Matonsu — швейцарский нож для мониторинга PostgreSQL

- Обзор Mamonsu
- Новые возможности Mamonsu



- **NewRelic** и **DataDog** – SaaS, open source client, можно дописывать свои собственные плагины, имеет встроенный плагин для PG.
- **OKMeter** – SaaS, НЕ open source client, удобные графики и представления.
- **Zabbix** – популярность (дефакто решение для большинства российских компаний), умеет всё необходимое, open source. Есть zabbix-agent, который получает данные через вызовы “psql -c “

Что такое Mamonsu

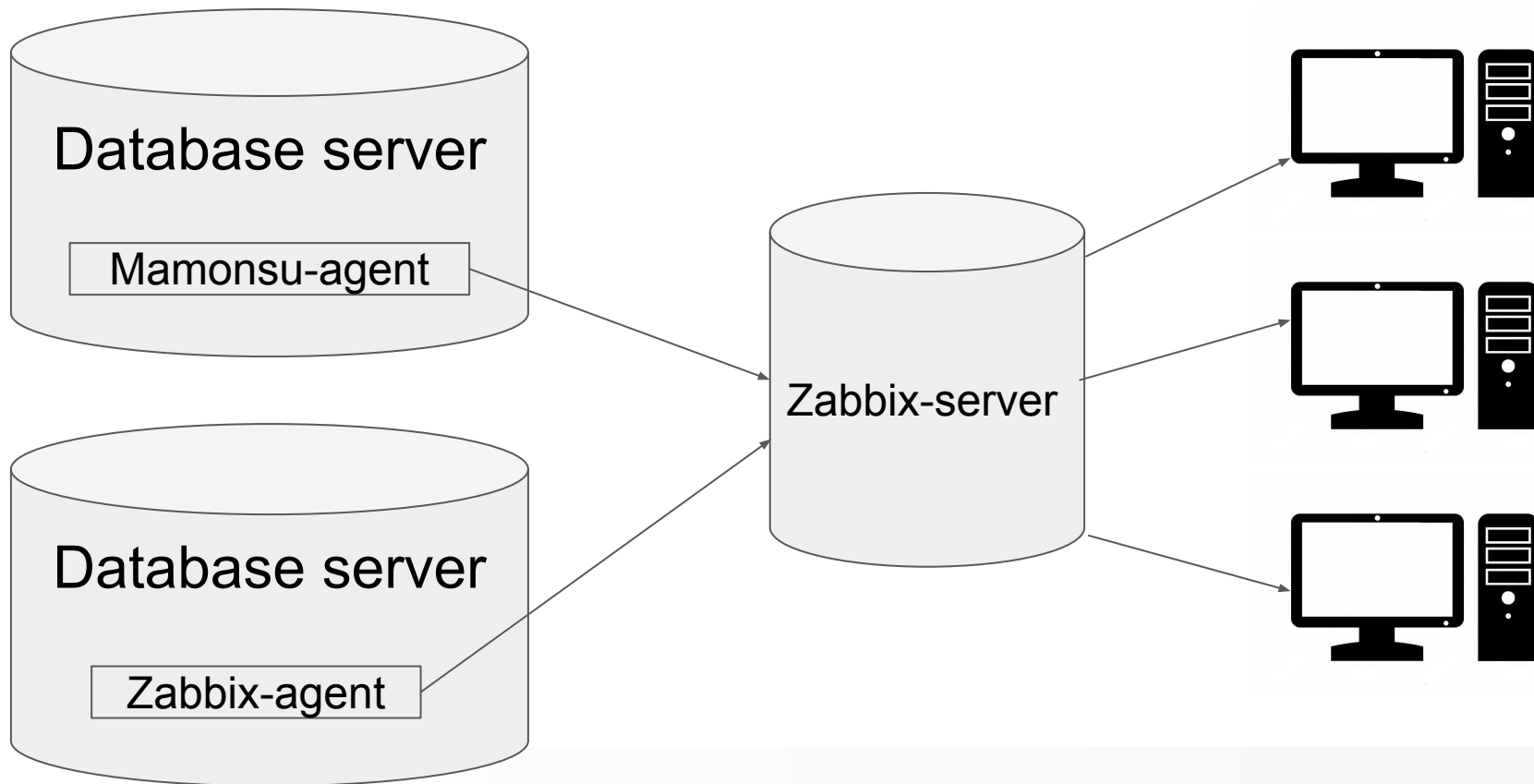
- Это активный агент (zabbix trapper) мониторинга для PostgreSQL и операционной системы
- Использует возможности Zabbix
- Обладает дополнительными инструментами
 - mamonsu tune
 - mamonsu report
- Написан на Python
- Выгружает шаблоны и конфигурационные файлы для zabbix-agent

- Mamonsu - поддерживаемый продукт
- Эффективная работа с PostgreSQL
- Расширяемость:
 - можно самому писать плагины
 - возможность настроить параметры сбора метрик под себя
- Широкий охват метрик для мониторинга, специфичных для PostgreSQL
- Поддержка метрик Postgres Pro Enterprise
- Возможность работы без интернета
- Работа “из коробки”
- Быстрая и простая настройка
- Кроссплатформенный
- BSD 3-clause licence

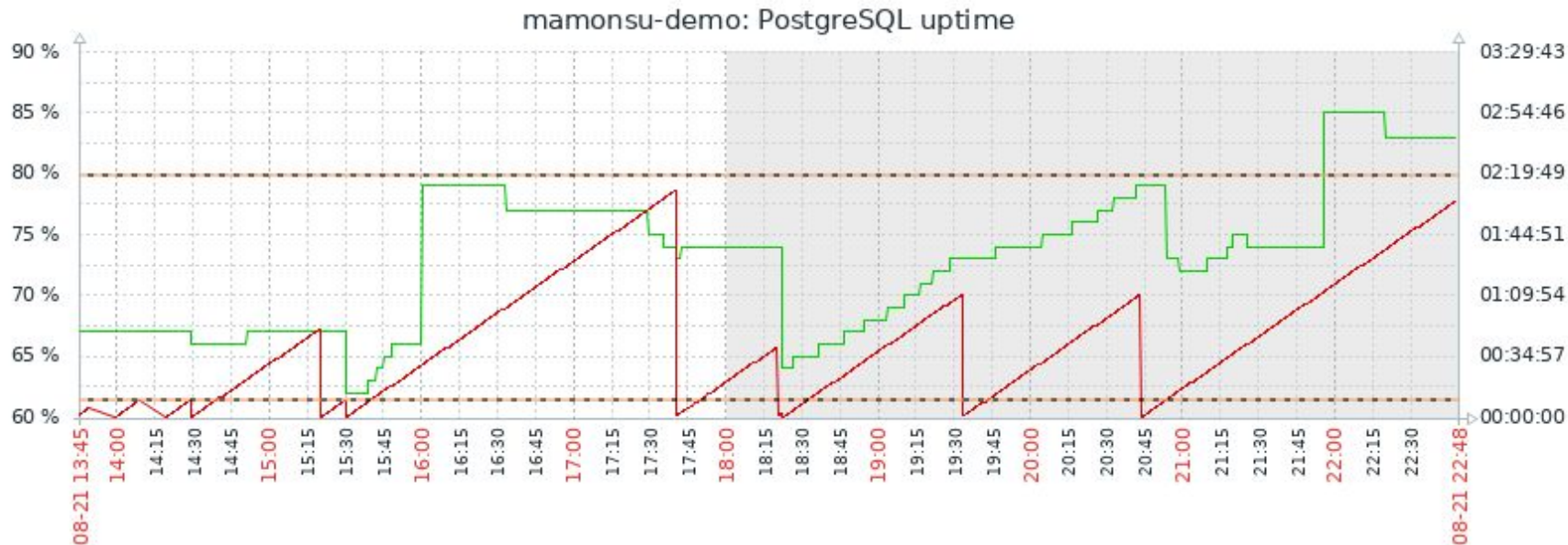
Причины выбора zabbix

Zabbix — система мониторинга и отслеживания статусов разнообразных сервисов

- Широко распространена в России и мире
- Триггеры
- Визуализация
- Статистика
- Расширяемость
- API
- Документация



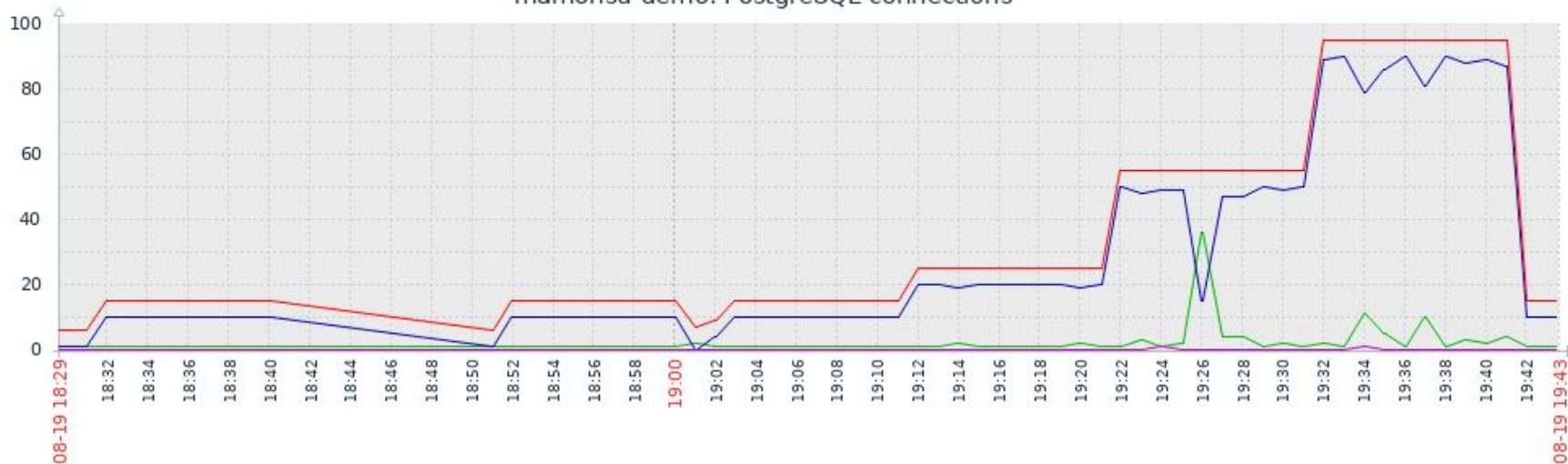
- Доступность СУБД
- Количество соединений
- Размеры баз данных
- Чекпоинты
- Скорость чтения / записи
- Блокировки
- Автовакуум
- Скорость генерации WAL



	last	min	avg	max
PostgreSQL: cache hit ratio [avg]	83 %	62 %	73.28 %	85 %
PostgreSQL: service uptime [avg]	02:03:33	00:00:00	00:44:45	02:09:30
Trigger: PostgreSQL cache hit ratio too low on mamonsu-demo (74 %)				[< 80]
Trigger: PostgreSQL service was restarted on mamonsu-demo (uptime=15:01:43)				[< 600]

Пример графика: количество соединений

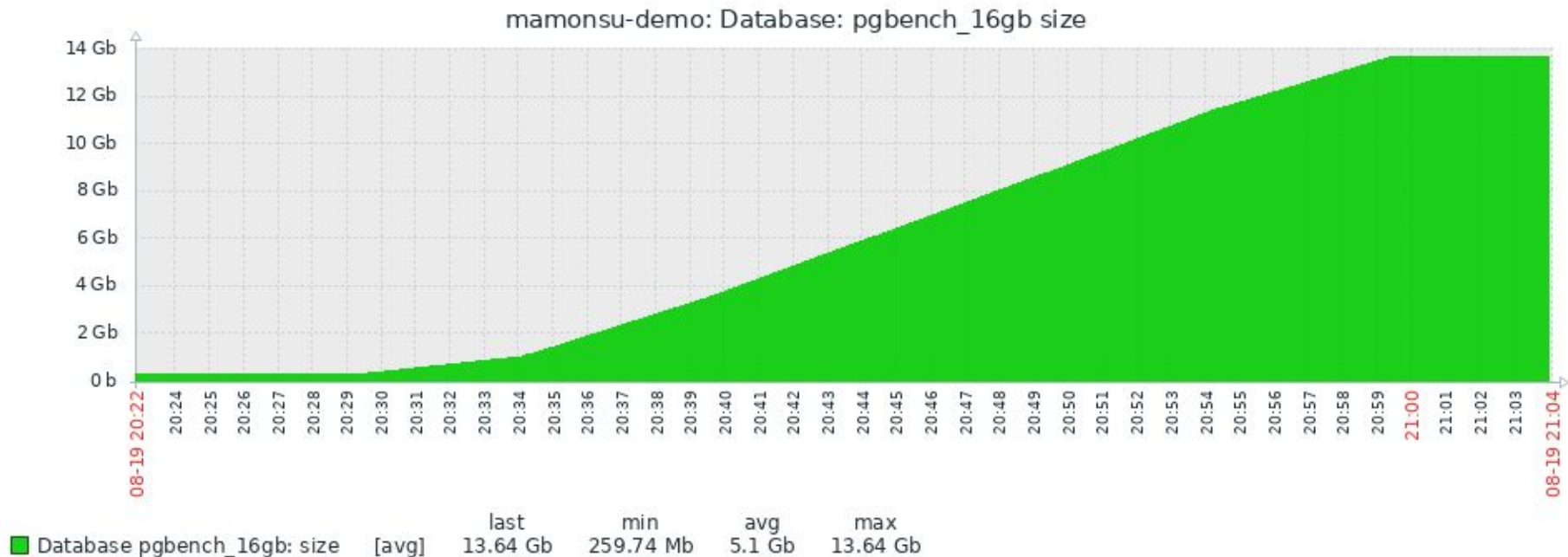
mamonsu-demo: PostgreSQL connections



- PostgreSQL: number of active connections
- PostgreSQL: number of idle connections
- PostgreSQL: number of idle in transaction connections
- PostgreSQL: number of total connections
- PostgreSQL: number of waiting connections

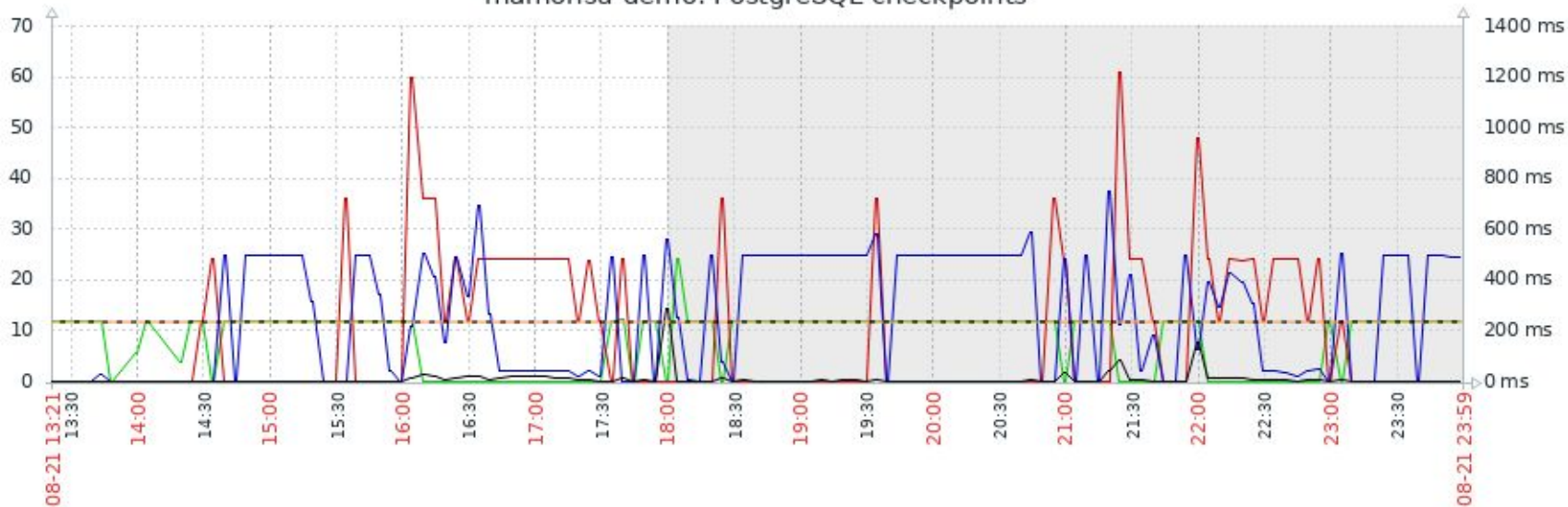
	last	min	avg	max
[avg]	1	1	2.24	36
[avg]	10	0	28.84	90
[avg]	0	0	0.0317	1
[avg]	15	6	35.13	95
[no data]				

Пример графика: размер базы данных



Пример графика: чекпоинты

mamonsu-demo: PostgreSQL checkpoints

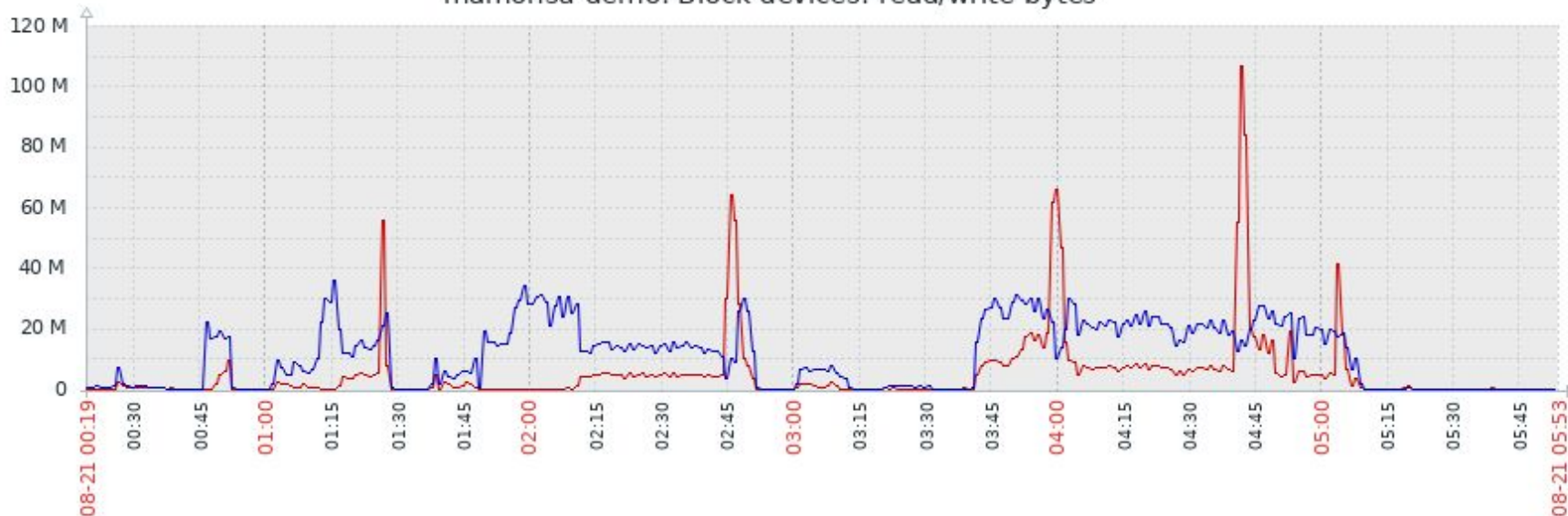


	last	min	avg	max
PostgreSQL checkpoints: by timeout (in hour)	[avg] 11.96	0	8.11	24
PostgreSQL checkpoints: by wal (in hour)	[avg] 0	0	8.93	61.02
PostgreSQL checkpoint: write time	[avg] 491.04 ms	0 ms	271.54 ms	751.9 ms
PostgreSQL checkpoint: sync time	[avg] 0.95 ms	0 ms	9.26 ms	286.89 ms

Trigger: PostgreSQL required checkpoints occurs to frequently on mamonsu-demo (> 12)

Пример графика: скорость чтения / записи

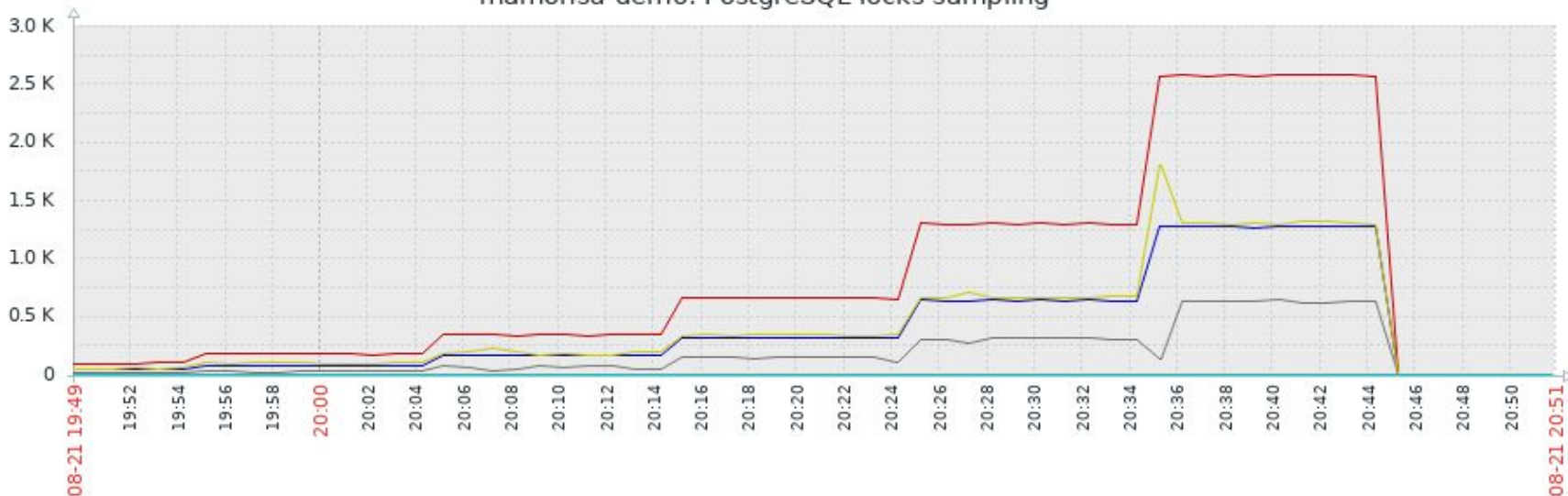
mamonsu-demo: Block devices: read/write bytes



	[avg]	last	min	avg	max
Block devices: read byte/s		3.62 K	68.27	5.08 M	106.79 M
Block devices: write byte/s		47.24 K	20.48 K	10.86 M	35.75 M

Пример графика: блокировки

mamonsu-demo: PostgreSQL locks sampling



- PostgreSQL locks: Read only queries
- PostgreSQL locks: SELECT FOR SHARE and SELECT FOR UPDATE
- PostgreSQL locks: Write queries
- PostgreSQL locks: VACUUM, ANALYZE, CREATE INDEX CONCURRENTLY
- PostgreSQL locks: CREATE INDEX
- PostgreSQL locks: Locks from application
- PostgreSQL locks: Locks from application or some operation on system catalogs
- PostgreSQL locks: ALTER TABLE, DROP TABLE, TRUNCATE, REINDEX, CLUSTER, VACUUM FULL, LOCK TABLE

	last	min	avg	max
[avg]	1	1	402.19	1.28 K
[avg]	0	0	0	0
[avg]	0	0	821.89	2.58 K
[avg]	0	0	0	0
[avg]	0	0	180.47	639
[avg]	0	0	0	0
[avg]	1	1	431.13	1.8 K
[avg]	0	0	0	0

Пример графика: автовакуум

mamonsu-demo: Database bloating overview: pgbench_1gb

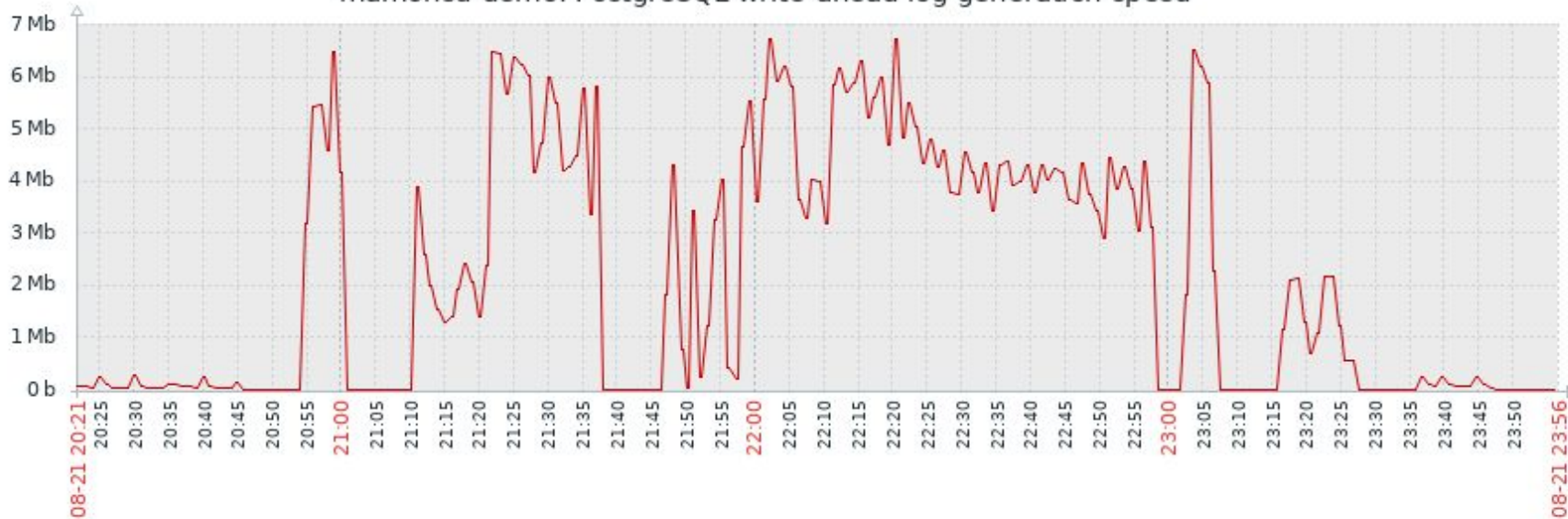


■ Count of bloating tables in database: pgbench_1gb
■ PostgreSQL: count of autovacuum workers

	last	min	avg	max
[avg]	2	0	1.51	3
[avg]	0	0	0.0333	1

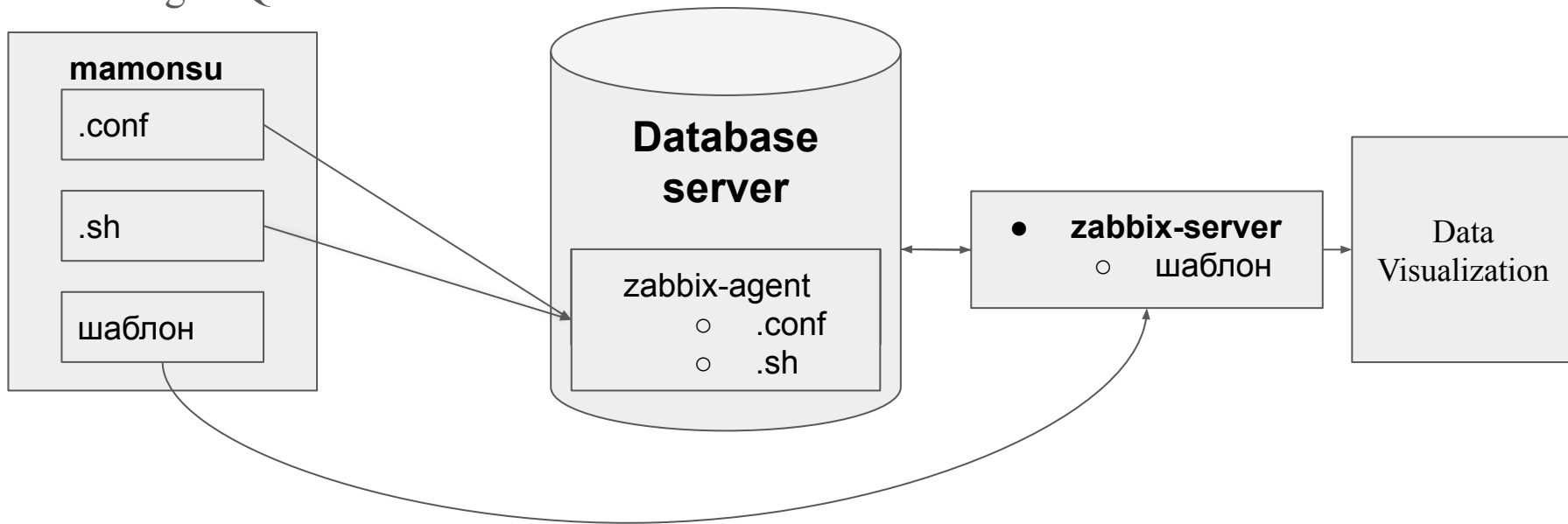
Пример графика: скорость генерации WAL

mamonsu-demo: PostgreSQL write-ahead log generation speed



■ PostgreSQL: wal write speed [avg] last 4.27 b min 4.2 b avg 2.2 Mb max 6.73 Mb

- выгрузка из Mamonsu шаблона для zabbix-agent
- выгрузка из Mamonsu конфигурационного файла для zabbix-agent для всех версий PostgreSQL



- Невозможность внедрения mamonsu некоторым заказчикам из-за инфраструктурных ограничений либо ограничений безопасности
- Использование существующей системы мониторинга заказчика при проведении аудита
- Необходимость ставить новое ПО (Mamonsu + python)

- Плюсы
 - готовое решение для Zabbix
- Минусы
 - zabbix-agent создает новое подключение к СУБД для каждого запроса (каждой метрики)
 - из коробки, нет готовых решений для мониторинга PostgreSQL (<https://www.zabbix.com/integrations/postgresql> есть решения сторонних разработчиков)
 - недостаточное количество метрик для мониторинга PostgreSQL
 - для zabbix-agent затруднительно собирать сложные запросы.

- Поддержка PostgreSQL 12
- Новые метрики и плагины
- Плагин для pgbouncer
- Получение топ запросов по нагрузке на систему и по времени выполнения
- Mamonsu в docker контейнере

- Установка агента MamonSU.
- Настройка агента MamonSU.
- Экспорт шаблона в Zabbix server.
- Добавление хоста в Zabbix server.
- Настройка bootstrap.
- Запуск.

Из исходников:

```
$ git clone ... && cd mamonsu && python setup.py build && python  
setup.py install
```

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

Для подключения и работы с СУБД редактируем конфигурационный файл:

`/etc/mamonsu/agent.conf`

```
[postgres]
```

```
# клиент
```

```
user = postgres
```

```
# пытаемся подключиться через сокет или ip
```

```
host = auto
```

```
# порт для подключения
```

```
port = 5432
```



```
[zabbix]
```

```
# идентификация на стороне Zabbix сервера (hostname)
```

```
client = testpg
```

```
# ip, dns zabbix сервера
```

```
address = 127.0.0.1
```

```
# порт zabbix сервера (по умолчанию 10051)
```

```
port = 10051
```

Пример для плагина **connections**:

```
[connections]
```

```
enabled = True
```

```
percent_connections_tr = 90
```

```
interval = 60
```

- Через веб-интерфейс Zabbix server.
- Используя Mamonsu:

```
$ mamonsu export template /usr/share/mamonsu/example.xml
```

```
$ mamonsu zabbix template export /usr/share/mamonsu/template.xml
```

```
--url=http://192.168.27.51/zabbix/ --user=Admin --password=zabbix
```

```
<name>PostgreSQL: number of total connections</name>
<type>2</type>
<snmp_community/>
<multiplier>0</multiplier>
<inventory_link>0</inventory_link>
<key>pgsql.connections[total]</key>
<snmp_oid/>
```

- Через веб-интерфейс Zabbix server.
- Используя Mamonsu:
 - `mamonsu zabbix host create <client name> <hostgroup id> <template id> <ip>`
`--url=http://192.168.27.51/zabbix/ --user=Admin --password=zabbix:`

```
$ export ZABBIX_USER=Admin
$ export ZABBIX_PASSWD=zabbix
$ export ZABBIX_URL=http://192.168.122.55/zabbix
$ mamonsu zabbix template id 'PostgresPro-Linux2'
10263
$ mamonsu zabbix hostgroup create mamonsu-demo
16
$ mamonsu zabbix host create mamonsu-demo 16 10263 192.168.27.133
{u'hostids': [u'10343']}
```

Опция **bootstrap** позволяет проводить мониторинг от non-superuser. Для применения **bootstrap** необходимо выполнить следующие действия:

- Отредактировать настройки аутентификации в **pg_hba.conf**.
- Выполнить следующие команды:

```
$ createuser mamonsu
```

```
$ mamonsu bootstrap -M mamonsu
```

```
Bootstrap successfully completed
```

```
$ service mamonsu start
```

Лог Mamonsu для проверки корректности работы:

```
[INFO] 2019-08-20 11:00:17,305 - root - Start mamonsu
[INFO] 2019-08-20 11:00:17,306 - HEALTH - started ...
[DEBUG] 2019-08-20 11:00:17,306 - PGSQL-(host= db=postgres user=postgres port=5433) - Run: "select buffe
ffers_alloc from pg_catalog.pg_stat_bgwriter;"
[INFO] 2019-08-20 11:00:17,306 - BGWRITER - started ...
[DEBUG] 2019-08-20 11:00:17,306 - PGSQL-(host= db=postgres user=postgres port=5433) - reconnecting
[DEBUG] 2019-08-20 11:00:17,307 - PGSQL-(host= db=postgres user=postgres port=5433) - connecting
[INFO] 2019-08-20 11:00:17,307 - CONNECTIONS - started ...
[INFO] 2019-08-20 11:00:17,307 - DATABASES - started ...
[INFO] 2019-08-20 11:00:17,307 - PGHEALTH - started ...
[INFO] 2019-08-20 11:00:17,307 - INSTANCE - started ...
[INFO] 2019-08-20 11:00:17,307 - XLOG - started ...
[INFO] 2019-08-20 11:00:17,308 - PGSTATSTATEMENT - started ...
[INFO] 2019-08-20 11:00:17,308 - PGBUFFERCACHE - started ...
[INFO] 2019-08-20 11:00:17,308 - PGWAITSMAMPLING - started ...
[INFO] 2019-08-20 11:00:17,308 - CHECKPOINT - started ...
[INFO] 2019-08-20 11:00:17,308 - OLDEST - started ...
[INFO] 2019-08-20 11:00:17,309 - PGLOCKS - started ...
[INFO] 2019-08-20 11:00:17,309 - CFS - started ...
[INFO] 2019-08-20 11:00:17,309 - ARCHIVECOMMAND - started ...
```

- Описать метрики.
- Добавить параметры для построения графиков.
- Добавить условия для триггеров.
- Добавить секцию нового плагина в конфигурационный файл Mamonsu.
- Указать директорию с пользовательскими плагинами в конфигурационном файле Mamonsu.
- Экспортировать новый шаблон.
- Загрузить новый шаблон на Zabbix server.

Пример добавления нового плагина

- Добавим новый плагин, в котором будем получать количества новых пользователей каждые пять минут.
- Укажем запрос, ключи, график и триггер для новой метрики.
- Добавим имя нового плагина в конфигурационном файле.


```
$ vim /etc/mamonsu/plugins/NewUsers.py
```

```
class NewUsers(Plugin):  
    def run(cls, zbx):  
        query = "SELECT count(*) FROM USERS_TABLE"  
        result = Pooler.query(query)  
        zbx.send('pgsql.count_users[]',int(result[0][0]),  
self.DELTA_CHANGE)
```

Создание элемента шаблона для нового плагина

```
$ vim /etc/mamonsu/plugins/NewUsers.py
```

```
def items(self, template):  
    item = template.item({  
        'key': self.right_type('pgsql.count_users{0}'),  
        'name': 'PostgreSQL count new users',  
        'value_type': self.VALUE_TYPE.numeric_unsigned  
    })  
    return item
```

```
$ vim /etc/mamonsu/plugins/NewUsers.py
```

```
def graphs(self, template):  
    graph = template.graph({  
        'name': 'PostgreSQL count new users',  
        'items': [  
            {'color': 'CC0000',  
             'key': self.right_type('pgsql.count_users{0}')}  
        ]  
    })  
    return graph
```

```
$ vim /etc/mamonsu/plugins/NewUsers.py
```

```
DEFAULT_CONFIG = {  
    'max_visitors': str(1000),  
}  
  
def triggers(self, template):  
    trigger = template.trigger({  
        'name': 'PostgreSQL count of new users is too high '  
            'on {HOSTNAME} (value={ITEM.LASTVALUE})',  
        'expression': '#{TEMPLATE:' +  
self.right_type('pgsql.count_users') + '.last()}' +  
        self.plugin_config('max_visitors')  
    })  
    return trigger
```

- Указать директорию для дополнительных плагинов:

[plugins]

enabled = True

directory = /etc/mamonsu/plugins

- Добавить секцию для нового плагина:

[newplugin]

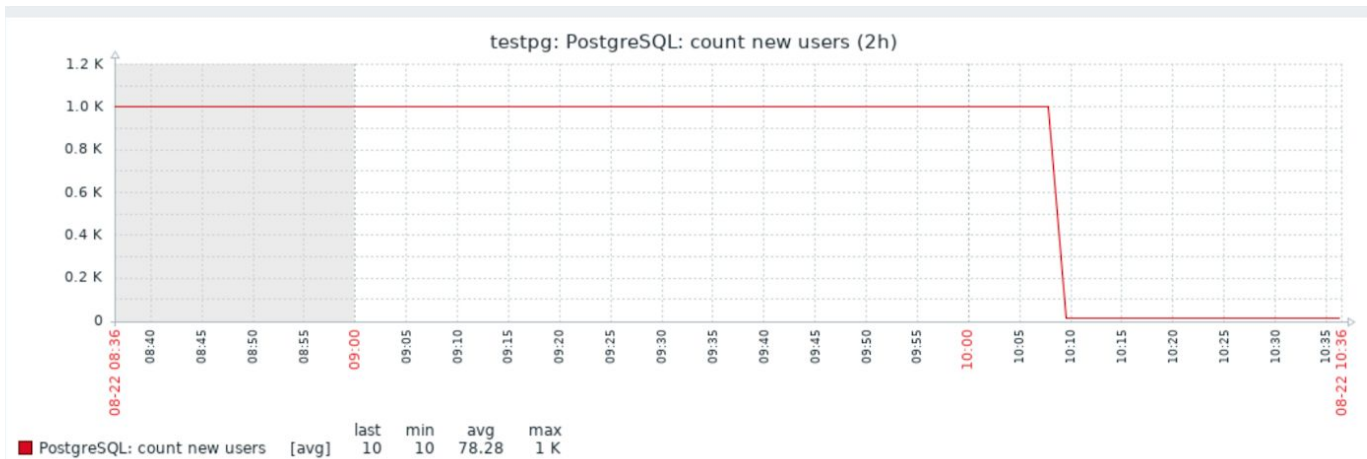
max_visitors = 1000

interval = 300

- `mamonsu export template_2.0.xml -a /etc/mamonsu/plugins`
- Добавить новый шаблон на Zabbix server.

```
$ service mamonsu start
```

График для нового плагина



10:06:59

10:09:50

RESOLVED

testpg

PostgreSQL too many new users on testpg (count of new users is more than 100)

2m 51s

Внедрение метрик Mamonsu в zabbix-agent

- Установить zabbix-agent.
- Получить шаблон и конфигурационный файл для работы zabbix-agent.

С установкой Mamonsu:

- `$ mamonsu export zabbix-template /etc/mamonsu/example.xml --plugin-type=all --pg-version=10`
- `$ mamonsu export zabbix-parameters /etc/zabbix/zabbix_agentd.d/userparameter_pgsql.conf --plugin-type=all --pg-version=10`
- загрузить шаблон на Zabbix server

Без установки Mamonsu:

- скачать нужную версию шаблона и конфигурационного файла для zabbix-agent <https://postgrespro.ru/products/extensions/mamonsu>,
- добавить конфигурационный файл в `/etc/zabbix/zabbix_agentd.d/`,
- загрузить шаблон на Zabbix server через веб интерфейс.

- **{\$PG_CONNINFO}**
-qAt -p 5433 -U postgres
- **{\$PG_PATH}**
/opt/pgpro/std-10/bin/psql

Template macros	Inherited and template macros
Macro	Value
<input type="text" value="{\$PG_CONNINFO}"/>	<input type="text" value="-qAt -p 5433 -U postgres"/> Remove
<input type="text" value="{\$PG_PATH}"/>	<input type="text" value="/opt/pgpro/std-10/bin/psql"/> Remove
Add	
<input type="button" value="Update"/>	<input type="button" value="Clone"/> <input type="button" value="Full clone"/> <input type="button" value="Delete"/> <input type="button" value="Delete and clear"/> <input type="button" value="Cancel"/>

Пример из конфигурационного файла для zabbix-agent

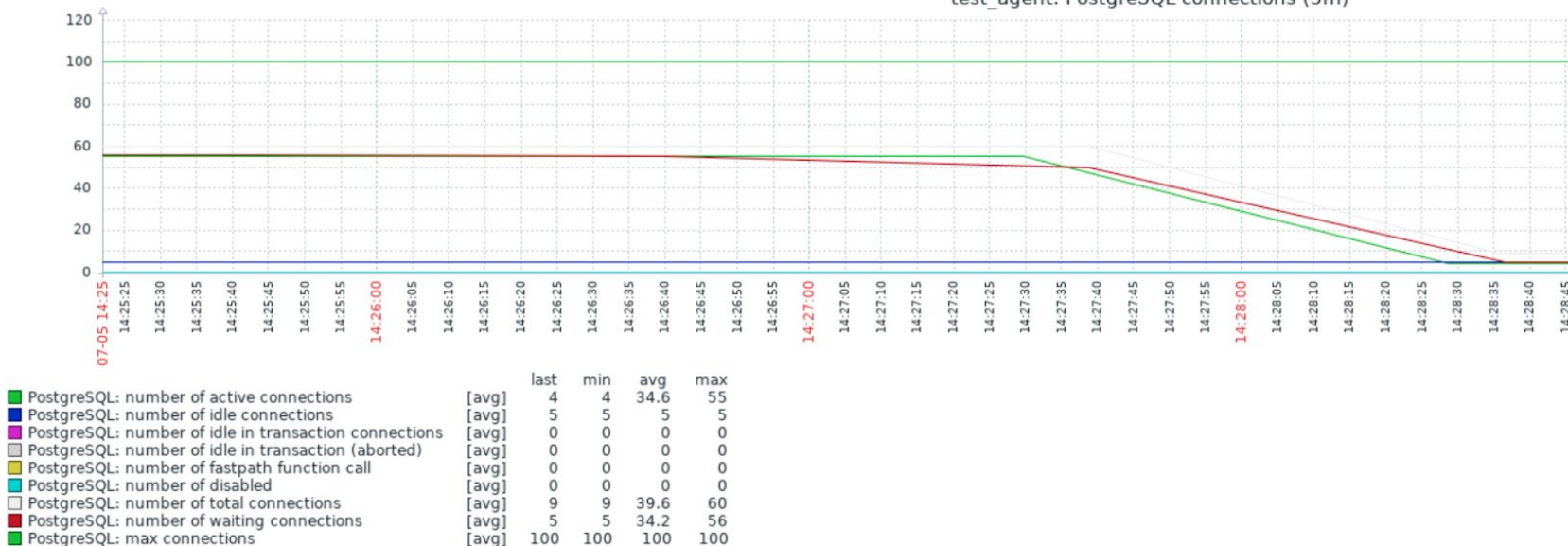
- `UserParameter=pgsql.uptime[*],$2 $1 -c "select date_part('epoch', now() - pg_postmaster_start_time());"`
- `UserParameter=pgsql.connections.active[*],$2 $1 -c "select count(*) from pg_catalog.pg_stat_activity where state = 'active';"`

Пример графика для количества соединений

Zoom: **5m** 15m 30m 1h 2h 3h 6h 12h 1d 3d 7d 14d All

◀
 « 7d 1d 12h 1h 5m | 5m 1h 12h 1d 7d »

test_agent: PostgreSQL connections (5m)



Оптимизация конфигурации системы с учётом собранной информации об используемом оборудовании и параметрах PostgreSQL.

- вывод рекомендаций по оптимизации на экран, или их мгновенное применение,
- возможность задавать различные опции при оптимизации.

```
$ mamonsu tune [--dry-run] [--disable-sudo] [--log-level  
{INFO|DEBUG|WARN}] [--dont-tune-pgbadger]  
[--dont-reload-postgresql]
```

Вывод подробной информации об используемом оборудовании и параметрах работы PostgreSQL.

- сохранение отчета со всеми параметрами в файле,
- информация об используемой памяти, настройках автовакуума, топ таблиц и БД по занимаемой ими памяти,
- регулирование через опции при запуске для каких параметров выводить отчет.

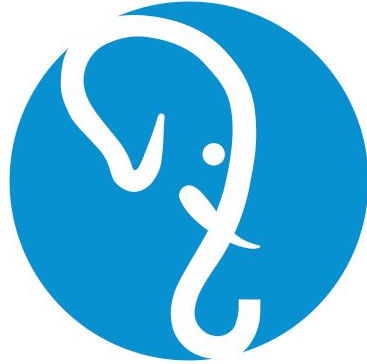
```
$ mamonsu report --run-system=Boolean]
[--run-postgres=Boolean] [--print-report=Boolean]
[--report-path report_file] [--disable-sudo]
[connection_options]
```

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

Документация: <https://postgrespro.ru/docs/postgrespro/11/mamonsu>

Шаблоны и файлы настроек для zabbix-agent:
<https://postgrespro.ru/products/extensions/mamonsu>

Демо-стенд Mamonsu: <https://mamonsu-demo.postgrespro.ru/>



Postgres Professional

www.postgrespro.ru

+7 (495) 150-06-91

info@postgrespro.ru