



Matonsu – швейцарский нож

postgrespro.ru

Васильев Д.
Смолкин Г.
Postgres Pro

Обзор средств мониторинга

- ◆ Cloud-Based: OKMeter, NewRelic, Datadog
- ◆ Self-Hosted: Zabbix, Nagios

Zabbix достоинства и недостатки:

- ◆ Распространенность
- ◆ Триггеры
- ◆ Статистика
- ◆ Расширяемость
- ◆ API
- ◆ Документация
- ◆ Комбайн
- ◆ Визуализация

Мониторинг PostgreSQL в Zabbix



- ◆ pg_monz
- ◆ libzbxpgsql
- ◆ lesovsky/zabbix-extensions

libzbxpgsql

```
int PG_QUERIES_LONGEST(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    int ret = SYSINFO_RET_FAIL; // Request result code
    const char *__function_name = "PG_QUERIES_LONGEST"; // Function name for log file

    char query[MAX_QUERY_LEN];
    char clause[MAX_CLAUSE_LEN];
    PGparams params = NULL; // freed later in pg_exec
    int version = 0;

    zabbix_log(LOG_LEVEL_DEBUG, "In %s()", __function_name);

    // build the filter clause
    memset(clause, 0, sizeof(clause));
    if (0 == build_activity_clause(request, result, clause, &params, 1))
        goto out;

    // build the full sql query
    memset(query, 0, MAX_QUERY_LEN);

    if (0 == (version = pg_version(request, result)))
```

psql + template

```
# pgbuffercache - extension which allow examining what's happening in the shared buffer cache in real time. http://www.postgresql.org/doc
UserParameter=pgsql.buffercache.clear[*],psql -qAtX $1 -c "select count(*) from pg_buffercache where not isdirty"
UserParameter=pgsql.buffercache.dirty[*],psql -qAtX $1 -c "select count(*) from pg_buffercache where isdirty"
UserParameter=pgsql.buffercache.used[*],psql -qAtX $1 -c "select count(*) from pg_buffercache where reldatabase is not null"
UserParameter=pgsql.buffercache.total[*],psql -qAtX $1 -c "select count(*) from pg_buffercache"

# General info
UserParameter=pgsql.ping[*],/bin/echo -e "\\t\\timing \\n select 1" | psql -qAtX $1 | tail -n 1 | cut -d' ' -f2
UserParameter=pgsql.uptime[*],psql -qAtX $1 -c "select date_part('epoch', now() - pg_postmaster_start_time())::int"
UserParameter=pgsql.cache.hit[*],psql -qAtX $1 -c "select round(sum(blks_hit)*100/sum(blks_hit+blks_read), 2) from pg_stat_database"

# Connections
UserParameter=pgsql.connections.active[*],psql -qAtX $1 -c "select count(*) from pg_stat_activity where state = 'active'"
UserParameter=pgsql.connections.idle[*],psql -qAtX $1 -c "select count(*) from pg_stat_activity where state = 'idle'"
UserParameter=pgsql.connections.idle_in_transaction[*],psql -qAtX $1 -c "select count(*) from pg_stat_activity where state = 'idle in tra"
UserParameter=pgsql.connections.total[*],psql -qAtX $1 -c "select count(*) from pg_stat_activity"
UserParameter=pgsql.connections.total_pct[*],psql -qAtX $1 -c "select count(*)*100/(select current_setting('max_connections')::int) from "
UserParameter=pgsql.connections.waiting[*],psql -qAtX $1 -c "select count(*) from pg_stat_activity where waiting"
UserParameter=pgsql.connections.prepared[*],psql -qAtX $1 -c "select count(*) from pg_prepared_xacts"
```

- ◆ Плагины на python
- ◆ Сбор метрик + визуализация + реакция на события
- ◆ Полностью свободная BSD-license

Мамонсу agent: план действий

- ◆ Установка агента
- ◆ Настройка агента
- ◆ Экспорт темплейта в Zabbix
- ◆ Создание хоста в Zabbix

- ◆ Через PackageCloud.io (linux)

```
curl -s https://packagecloud.io/install/repositories/postgrespro/mamonsu/script.deb.sh | sudo bash
```

```
curl -s https://packagecloud.io/install/repositories/postgrespro/mamonsu/script.rpm.sh | sudo bash
```

- ◆ Make (linux self-hosted)

```
make deb
```

```
make rpm
```

- ◆ Windows installer:

```
https://postgrespro.ru/products/download
```

- ◆ Другие OS:

```
python setup.py install
```

Настройка: конфигурация в ini-файле

- ◆ Если вам хватает штатного:
`/etc/mamonsu/agent.conf`
- ◆ Если вы пишете новые плагины:
`mamonsu config export agent.conf`

Секция [postgres] – подключение к инстансу



```
[postgres]

# клиент
user = postgres

# база по умолчанию
database = postgres

# пытаемся подключиться через сокет или ip
host = auto

# не забудьте про права на файл!
password = None

# порт для подключения
port = 5432

# для того чтобы легко идентифицировать приложения
application_name = mamonsu
```

Секция [zabbix] – подключение к Zabbix

```
[zabbix]
```

```
# включить отправку в Zabbix (включено по умолчанию)
```

```
enabled = True
```

```
# идентификация на стороне Zabbix-сервер (hostname)
```

```
client = dell.simpson
```

```
# ip, dns zabbix сервера
```

```
address = 127.0.0.1
```

```
# порт zabbix сервера
```

```
port = 10051
```

Секция [plugins] – пользовательские плагины

```
[plugins]  
enabled = True  
directory = /var/log/mamonsu
```

Секция [metric_log]

```
[metric_log]
```

```
enabled = False  
directory = /var/log/mamonsu  
max_size_mb = 1024
```

Секция [agent] – http-апи для общения с запущенным mamonsu agent

```
[agent]
```

```
# включено по умолчанию
```

```
enabled = True
```

```
host = 127.0.0.1
```

```
port = 10052
```

Мамонсу bootstrap: мониторинг PostgreSQL без прав суперпользователя

```
mamonsu bootstrap -d DB -U postgres
```

- ◆ Версия агента `mamonsu` напрямую связана с темплейтом, который описывает как хранить данные, какие графики по нему строить и как реагировать на те или иные изменения
- ◆ Вместе с пакетом распространяет предгенеренный конфиг, но если вы написали дополнительные плагины и указали их в конфиге в секции `plugins` вам необходимо выгрузить новый темплейт:

```
mamonsu export template template.xml -c agent.conf
```

Демонстрация: mamonsu zabbix

```
# переменные окружения, которые можно также передать через атрибуты
export ZABBIX_USER=Admin
export ZABBIX_PASSWORD=zabbix
export ZABBIX_URL='http://zabbix.company.org/zabbix'
```

```
# загрузка темплейта
mamonsu zabbix template export /usr/share/mamonsu/template.xml
# получим и запомним template id
template_id=$(mamonsu zabbix template id 'PostgresPro')
# получим и запомним hostgroup id
hostgroup_id=$(mamonsu zabbix hostgroup id 'Linux servers')
# создание хоста postgres.company.org
mamonsu zabbix host create postgres.company.org $hostgroup_id $template_id postgres.company.org
```

Демонстрация: mamonsu zabbix

`mamonsu zabbix item error postgres.company.org`

Демонстрация: mamonsu report



Джентельменский минимум информации о системе и PostgreSQL

Демонстрация: mamonsu report

- ◆ Платформа железа
- ◆ Ос, ядро и параметры запуска
- ◆ Процессор, микро-бенч
- ◆ Top-cpu, Top-memory процессы
- ◆ Memory
- ◆ sysctl: шедулинг, файловый дескрипторы, память
- ◆ Диски, iostat, lvm и raid, lspci

Демонстрация: mamonsu report

- ◆ Полная версия PostgreSQL
- ◆ Cache hit, tps и rollbacks
- ◆ Текущие соединения
- ◆ Настройки автовакуума и bgwriter
- ◆ Подключенные библиотеки
- ◆ Data safe
- ◆ Статистика по таблицам

Демонстрация: `matonsu tune`



Тюнинг системы и настройка PostgreSQL

Демонстрация: mamonsu tune

Придется поверить на слово, так как тест длинный:

Azure VM: DS3_V2 (4 ядра, 14Gb RAM)

Размер ДБ: 28Gb

Команда: `pgbench -M prepared -j 5 -c 15 -T 7200 -r -P 1`

2 часа vanilla: 408

2 часа с tune: 5990

Mamonsu: metric_log

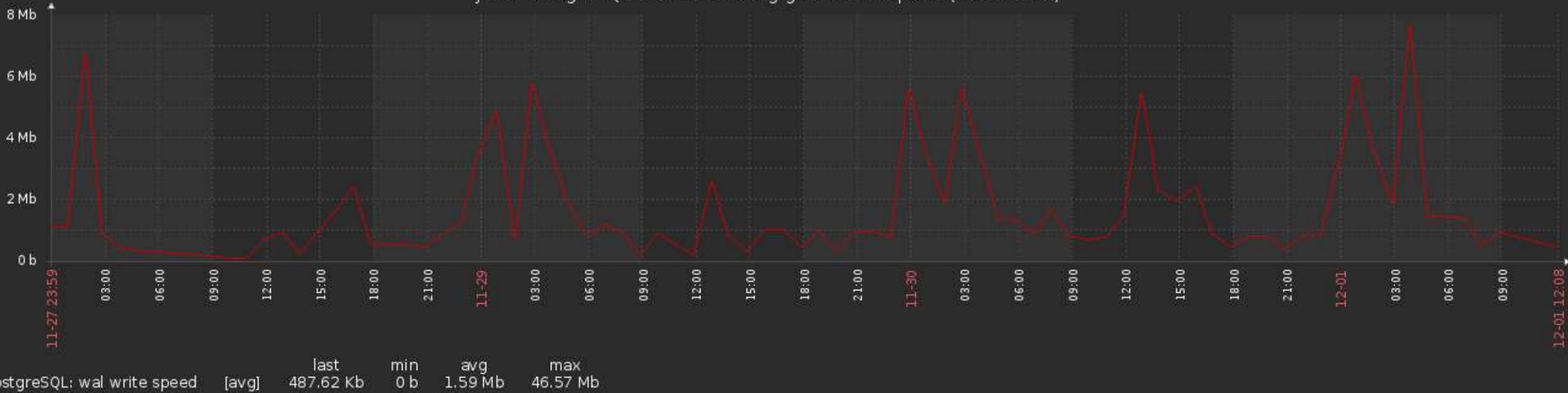
- ◆ Позволяет сохранять статистику без использования сторонних сервисов
- ◆ Строить по ним графики
- ◆ В планах дописать статическую html-страницу

Мамонсу: дополнительные плагины

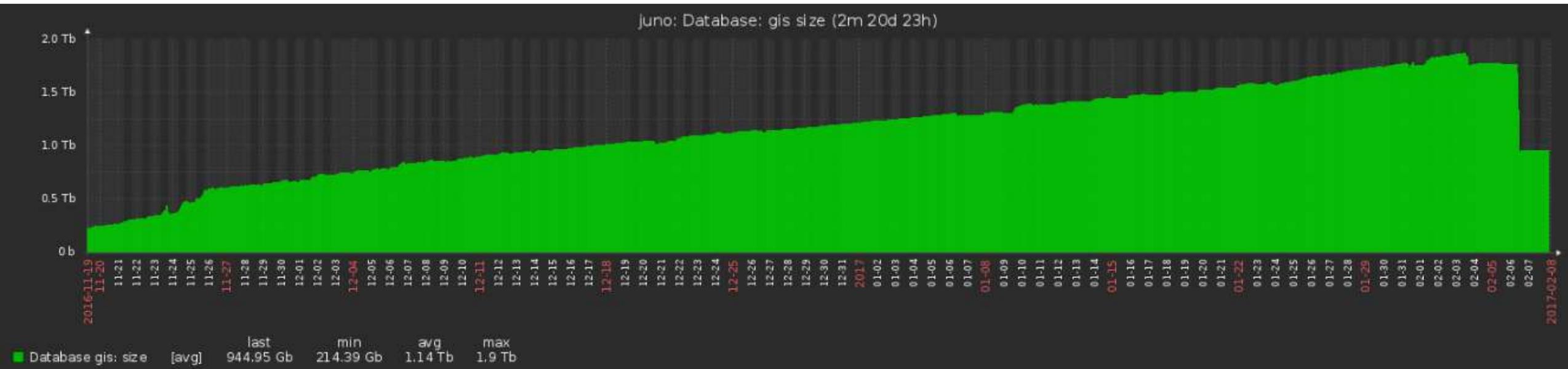
<https://github.com/tarabanton/mamonsu-plugins>

WAL generation speed

juno: PostgreSQL write-ahead log generation speed (3d 12h 9m)



Database size



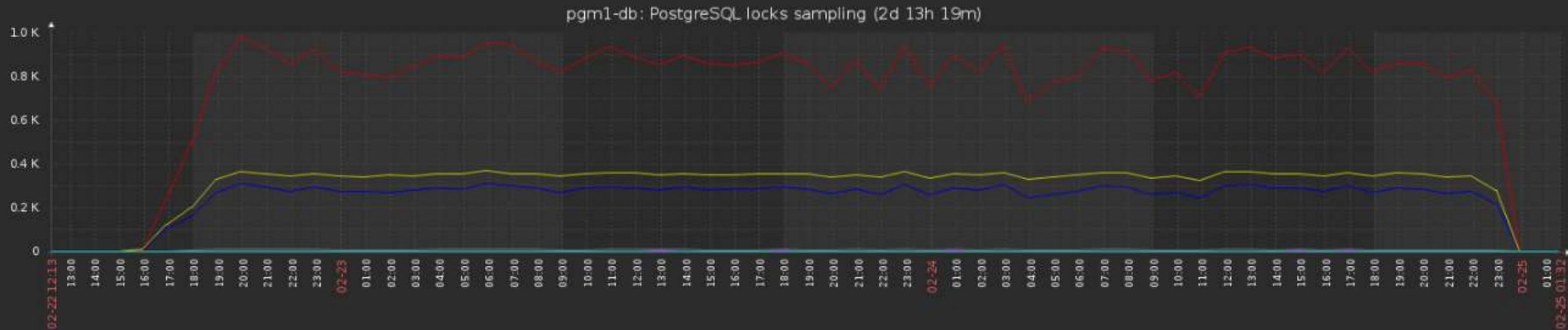
System processes

juno: Processes overview (3d 13h)



		last	min	avg	max
Processes: in state running	[avg]	1.81	1	16.87	66
Processes: in state blocked	[avg]	0.1321	0	1.06	123
Processes: forkrate	[avg]	11.02	2.87	26.2	1.47 K
Trigger: Process fork-rate to frequently on juno			[> 500]		

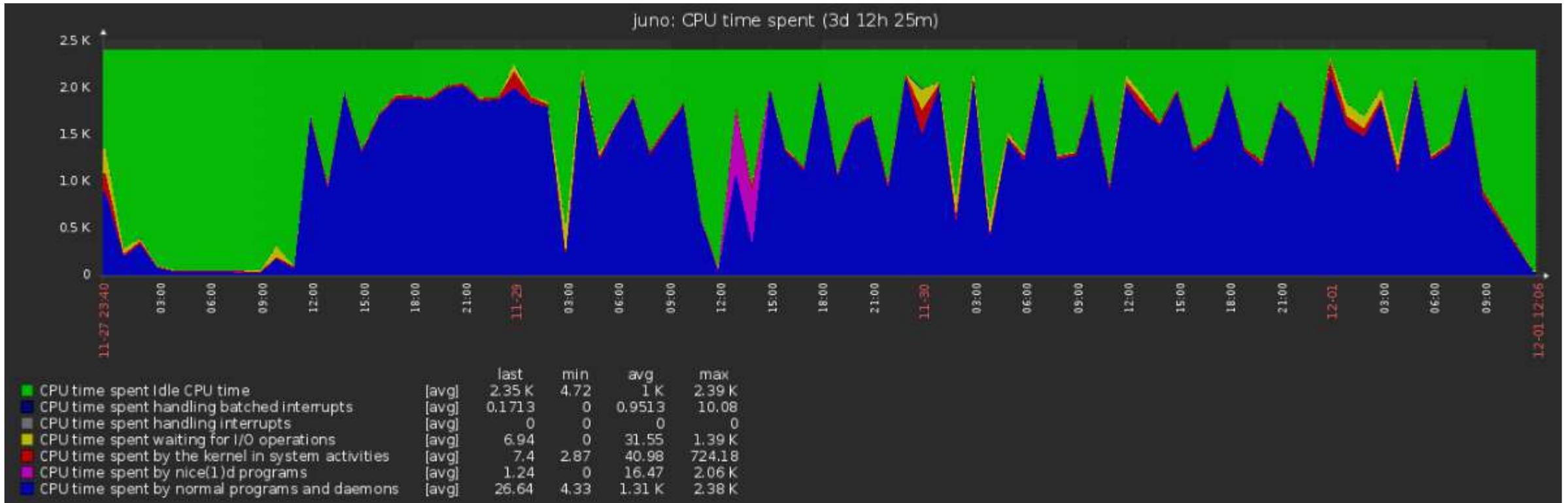
Logical locks



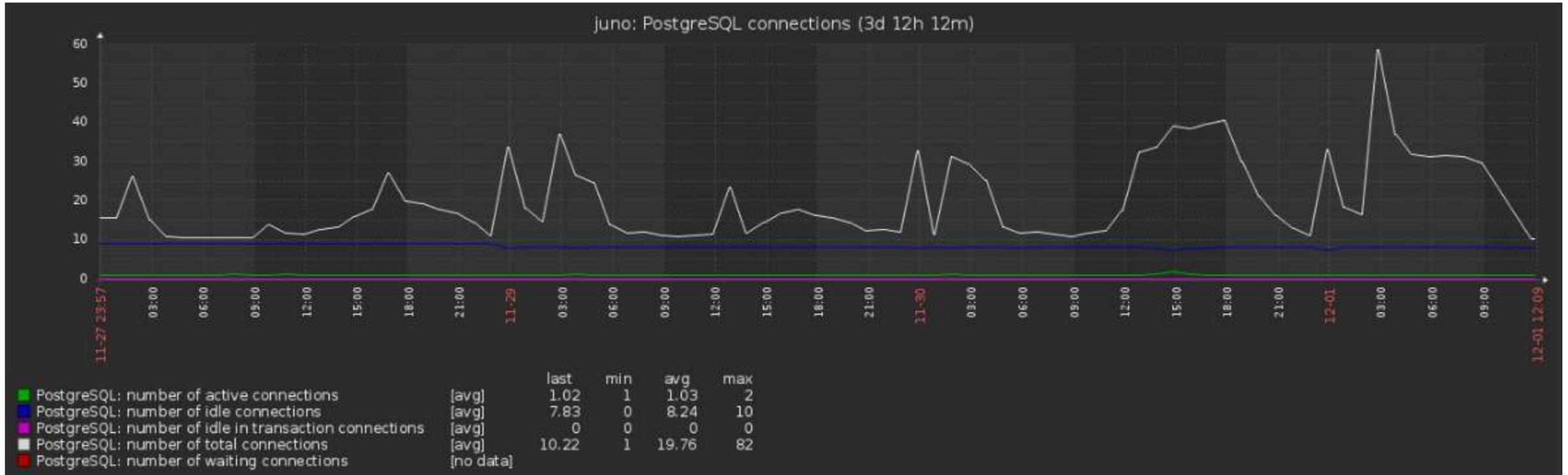
- PostgreSQL locks: Read only queries
- PostgreSQL locks: SELECT FOR SHARE and SELECT FOR UPDATE
- PostgreSQL locks: Write queries
- PostgreSQL locks: VACUUM, ANALYZE, CREATE INDEX CONCURRENTLY
- PostgreSQL locks: CREATE INDEX
- PostgreSQL locks: Locks from application
- PostgreSQL locks: Locks from 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	249.13	403
[avg]	0	0	0	0
[avg]	0	0	755.39	1.39 K
[avg]	0	0	1.64	3
[avg]	0	0	6.43	33
[avg]	0	0	0	0
[avg]	1	1	309.34	527
[avg]	0	0	0	2

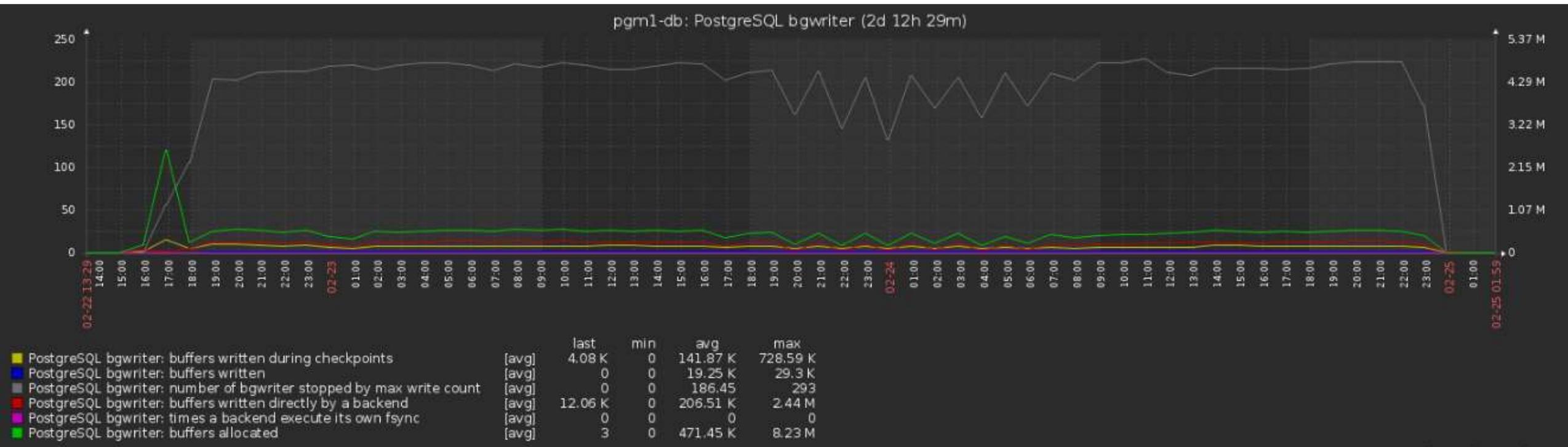
CPU



Connections



BGWriter



Oldest xid



Дашборд для zabbix

<https://github.com/alexanderzobnin/grafana-zabbix>

Postgres Professional

<http://postgrespro.ru/>

+7(495)1500691

info@postgrespro.ru

The background is a collage of hexagonal tiles in various shades of blue, orange, and grey. Some tiles contain abstract patterns like wavy lines, dots, or silhouettes of buildings and trees. The text "postgrespro.ru" is centered in an orange diamond-shaped tile.

postgrespro.ru