



ZABBIX 5.0

ОБНОВЛЕНИЕ ZABBIX НА НОВУЮ
ВЕРСИЮ С TIMESCALEDB



Александр Петров-Гаврилов
Инженер технической поддержки

01

ЗАЧЕМ НУЖНО ОБНОВЛЕНИЕ?



КОРОТКО

ШЛЁПАЕТ КРЫШКУ СЕРВЕРА

ZABVIX

**МОЖЕТ ВМЕСТИТЬ СЕБЯ СТОЛЬКО МЕТРИК
И ПРИ ЭТОМ БЫТЬ БЕЗОПАСНЫМ И ЭФФЕКТИВНЫМ**

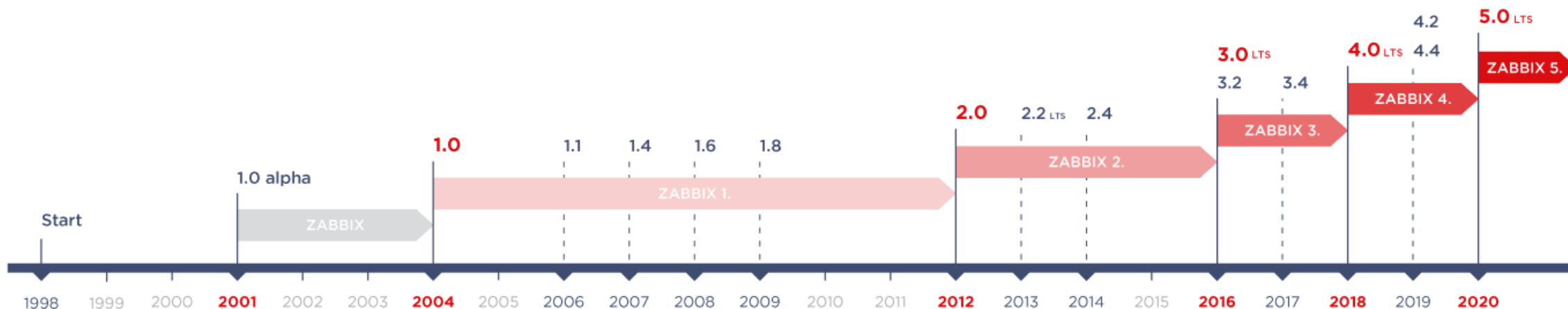
imgflip.com

И ВСЁ ТАКИ ПОЧЕМУ?

- ✓ **Стабильнее**
- ✓ **Производительнее**
- ✓ **Безопаснее**
- ✓ **Новый функционал**



И КОНЕЧНО ЖЕ, **ЖИЗНЕННЫЙ ЦИКЛ**



Zabbix LTS (Долгосрочная поддержка). Выпуски Zabbix LTS поддерживаются для клиентов Zabbix в течении пяти (5) лет

Стандарт: до следующего релиза

КАКУЮ ВЕРСИЮ ВЫБРАТЬ?

5.0 LTS

02

ЧТО ТАКОЕ TIMESCALEDDB И
ПОЧЕМУ?



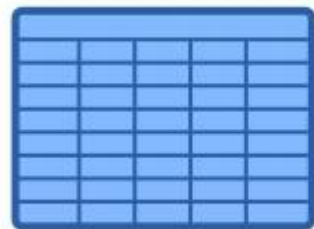
ПОЧЕМУ?



ЧТО ТАКОЕ **TIMESCALEDB**?

TimescaleDB это расширение для PostgreSQL, внедряющее производительность временных рядов и оптимизацию в управление данными к стандартной базе данных PostgreSQL, основанных на таких архитектурных решениях как:

- ✓ **Hypertables(гипертаблицы)**
 - Абстракция одной непрерывной и основная точка доступа к данным через стандартный SQL.
- ✓ **Chunks(фрагменты)**
 - Гипертаблицы автоматически разбиваются на фрагменты; каждый соответствует определенному временному интервалу и области



Hypertable



Chunk

ПОЧЕМУ **TIMESCALEDB**?

- ✓ **Это расширение**
 - Оно не требует дополнительных ресурсов, отдельных виртуальных машин или особых изменений инфраструктуры, Вы всё так же можете использовать PostgreSQL и привычные инструменты для выполнения запросов.
- ✓ **Код остаётся без изменений**
 - Весь относящийся к базе код остаётся нетронут
- ✓ **Производительность**
 - Значительные улучшения в производительности Zabbix history syncer и housekeeper.



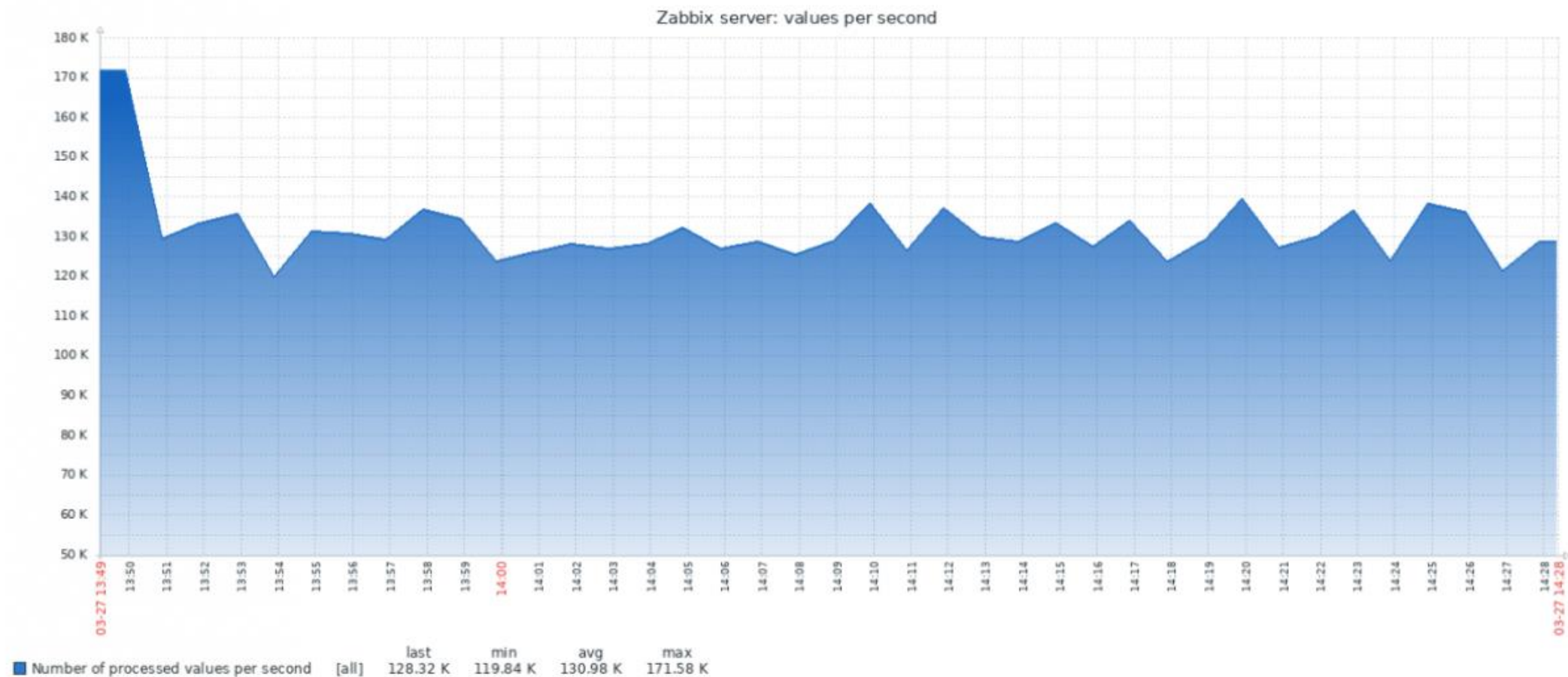
ПОЧЕМУ **TIMESCALEDB** - ПРОИЗВОДИТЕЛЬНОСТЬ

PostgreSQL без расширения TSDB



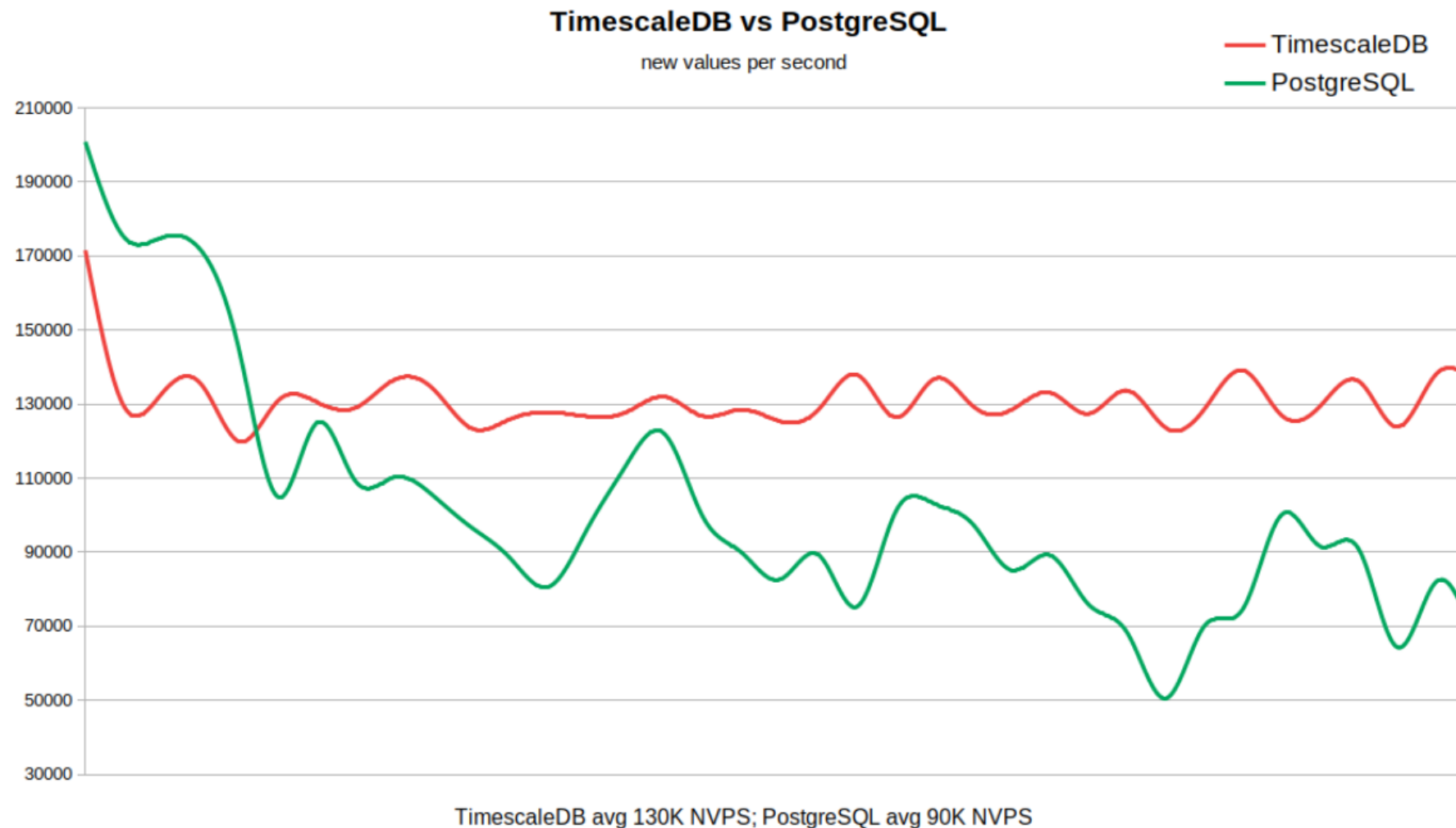
ПОЧЕМУ **TIMESCALEDB** - ПРОИЗВОДИТЕЛЬНОСТЬ

PostgreSQL с расширением TSD



ПОЧЕМУ **TIMESCALEDB** - ПРОИЗВОДИТЕЛЬНОСТЬ

Прямое сравнение



ПОЧЕМУ **TIMESCALEDB** – СЖАТИЕ ДАННЫХ

Возможность сэкономить до 90% используемого места

Нагрузка	До сжатия	После	Экономия
IT metrics (from Telco beta tester)	1396 GB	77.0 GB	94% savings
Industrial IoT monitoring data (from beta tester)	1.445 GB	0.077 GB	95% savings
IT metrics (DevOps dataset from TSBS)	125 GB	5.5 GB	96% savings
IoT monitoring data (IoT dataset from TSBS)	251 GB	23.8 GB	91% savings

КАК РАБОТАЕТ СЖАТИЕ

До сжатия

time	device_id	cpu	disk_io	energy_consumption
12:00:02	1	88.2	20	0.8
12:00:01	2	300.5	30	0.9
12:00:01	1	88.6	25	0.85
12:00:01	2	299.1	40	0.95

После сжатия

time	device_id	cpu	disk_io	energy_consumption
[12:00:02, 12:00:02, 12:00:01, 12:00:1]	[1, 2, 1, 2]	[88.2, 300.5, 88.6, 299.1]	[20, 30, 25, 40]	[0.8, 0.9, 0.85, 0.95]

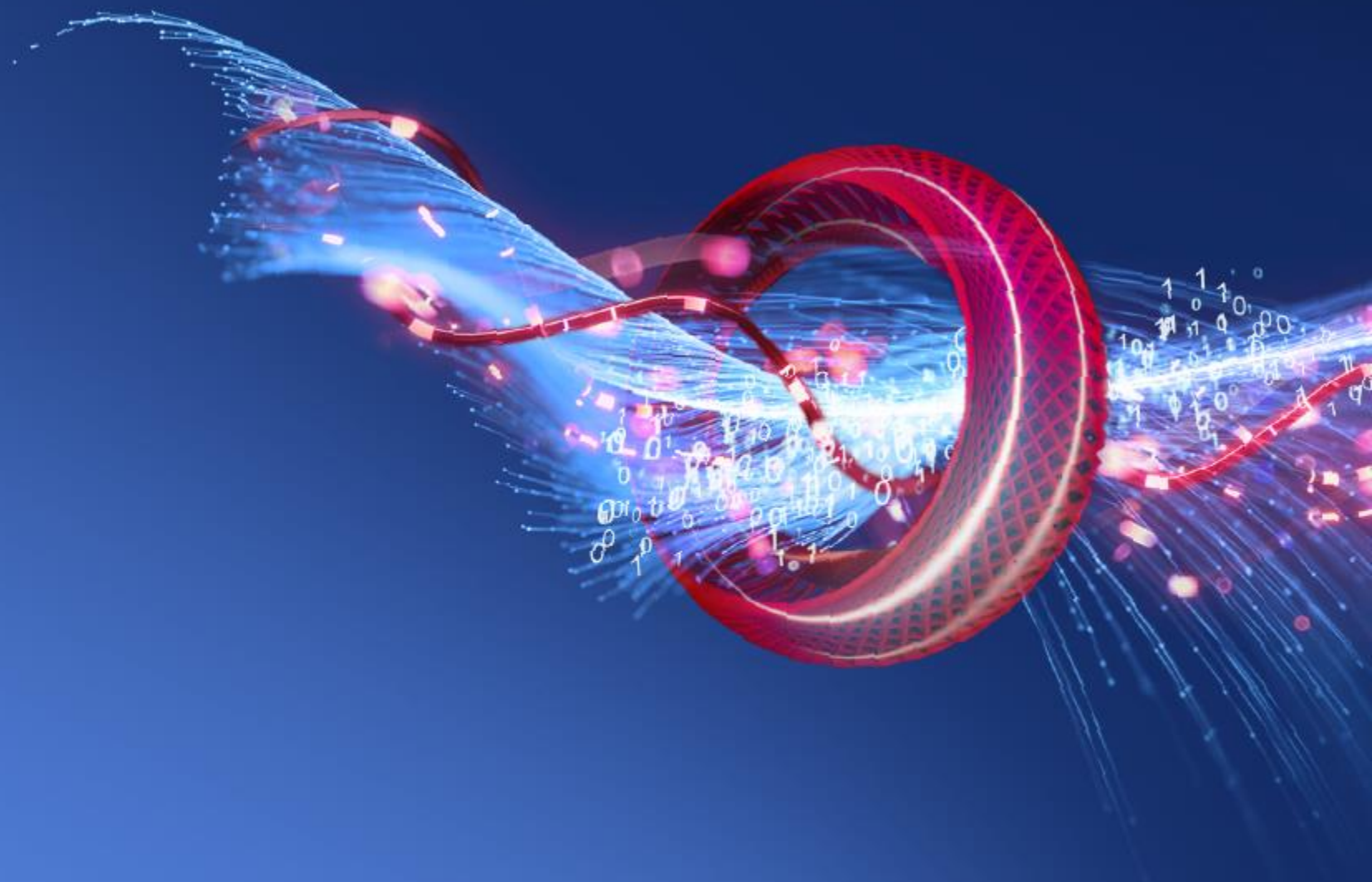
TIMESCALEDB – ВСЕ ТОТ ЖЕ SQL

- ✓ **Поддержка всех стандартных SQL операций и запросов**
- ✓ **Совместимость с экосистемой и инструментарием PostgreSQL**
- ✓ **Прозрачное партиционирование по временным промежуткам/месту для вертикального масштабирования**
- ✓ **Остаётся вполне доступной для новичков**



03

ЧТО ОБНОВЛЯТЬ?



ОБНОВЛЕНИЕ ЧЕГО



(Ну почти)

ЧТО НУЖНО ОБНОВИТЬ

- ✓ **Zabbix сервер**
- ✓ **Базу данных (в зависимости от версии)**
- ✓ **Frontend**
- ✓ **Прокси**
- ✓ **Java Gateway**



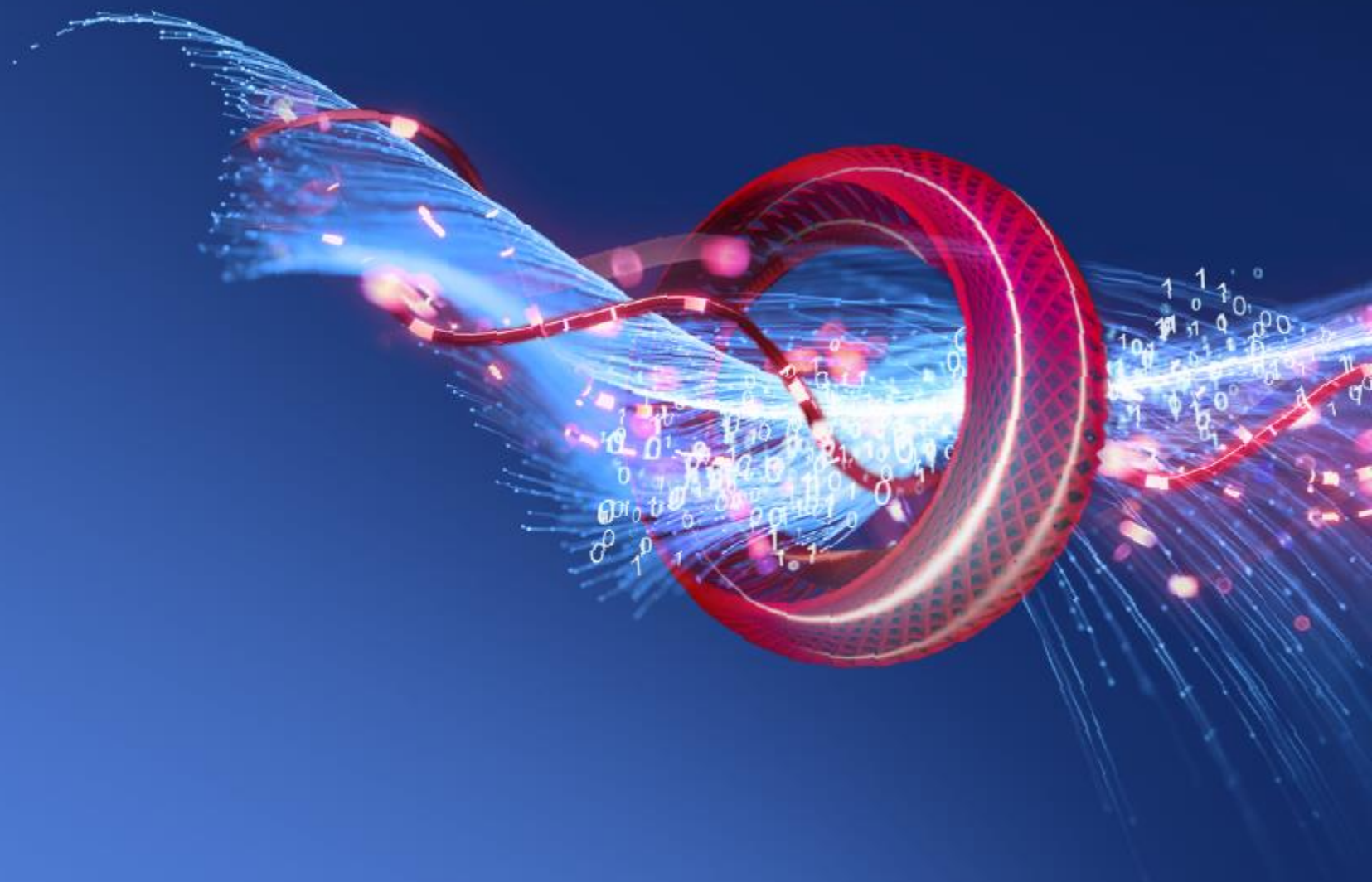
ОБНОВЛЯТЬ НЕ ТРЕБУЕТСЯ

- ✓ **Zabbix агент**
- ✓ **Базу данных (в зависимости от версии)**



04

КАК ОБНОВИТЬ?



КАК ОБНОВИТЬ?



СИСТЕМА

- ✓ **CentOS 7**
- ✓ **PostgreSQL 9.6**
- ✓ **Zabbix server 4.0**
- ✓ **Среднестатистическая VM**

UPGRADING POSTGRESQL

Используете старую версию PostgreSQL, но хотели бы использовать полный функционал TimescaleDB, включая сжатие?

Убедитесь, что обновили БД хотя бы до версии PostgreSQL 10.2 или выше. Как обновить PostgreSQL?

1. Установите репозиторий RPM версии которую будете использовать:

```
yum install https://download.postgresql.org/pub/repos/yum/reposms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

2. Установите новую версию PostgreSQL

```
yum install postgresql12-server
```

3. Остановите используемую на данный момент

```
systemctl stop postgresql-9.6
```

4. Инициализируйте новую

```
sudo su postgres  
cd ~/  
/usr/pgsql-12/bin/initdb -D /var/lib/pgsql/12/data/
```

5. Смигрируйте данные

```
/usr/pgsql-12/bin/pg_upgrade --old-datadir /var/lib/pgsql/9.6/data/  
--new-datadir /var/lib/pgsql/12/data/ --old-bindir /usr/pgsql-9.6/bin/  
--new-bindir /usr/pgsql-12/bin/
```

6. Запустите новую

```
systemctl start postgresql-12
```

СДЕЛАЙТЕ РЕЗЕРВНУЮ КОПИЮ

✓ Ознакомьтесь с заметками по обновлению и важными изменениями

https://www.zabbix.com/documentation/5.0/manual/installation/upgrade/packages/rhel_centos

✓ Сделайте резервную копию базы данных Zabbix

```
pg_dump dbname > dbname.bak
```

✓ Или резервную копию конфигурационных файлов

```
$ cp /etc/zabbix/zabbix_server.conf /<backup directory>/ # Zabbix server config file
$ cp /etc/zabbix/zabbix_agentd.conf /<backup directory>/ # Zabbix agent config file
$ cp /usr/share/zabbix/alertscripts/* /<backup directory>/ # Alert scripts
$ cp /usr/share/zabbix/externalscripts/* /<backup directory>/ # External scripts
$ cp -R /usr/share/zabbix/ /<backup directory>/ # Web frontend PHP files
$ cp /etc/httpd/conf/httpd.conf /<backup directory>/ # Apache config. files
$ cp /etc/httpd/conf.d/zabbix.conf /<backup directory>/ # Zabbix/PHP parameters
$ cp /etc/zabbix/web/zabbix.conf.php /<backup directory>/ # Zabbix frontend parameters.
```

НАЧНИТЕ ОБНОВЛЕНИЕ

1. Остановите Zabbix сервер, чтобы убедиться о том что новые данные не отправляются в базу данных

```
systemctl stop zabbix-server
```

2. Обновите текущую версию репозитория

```
rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/7/x86\_64/zabbix-release-5.0-1.el7.noarch.rpm
```

3. Обновите компоненты Zabbix

```
yum upgrade zabbix-server-pgsql zabbix-agent
```

ОБНОВИТЕ FRONTEND

1. Удалите старый frontend

```
yum remove zabbix-web-*
```

2. Установите репозиторий SCL

```
yum install centos-release-scl
```

3. Отредактируйте файл /etc/yum.repos.d/zabbix.repo

```
[zabbix-frontend]
name=Zabbix Official Repository frontend - $basearch
baseurl=http://repo.zabbix.com/zabbix/5.0/rhel/7/$basearch/frontend
enabled=1
```

4. Установите пакеты нового frontend

```
yum install zabbix-web-pgsql-scl
```

5. Обновите временную зону в файле /etc/opt/rh/rh-php72/php-fpm.d/zabbix.conf.

6. Запустите и добавьте в автозапуск сервис php-fpm.

```
systemctl start rh-php72-php-fpm
```

7. Перезагрузите Apache

```
systemctl restart httpd
```

УСТАНОВИТЕ **TIMESCALEDB**

1. Добавьте репозиторий TimescaleDB:

<https://docs.timescale.com/latest/getting-started/installation/rhel-centos/installation-yum>

```
sudo tee /etc/yum.repos.d/timescale_timescaledb.repo <<EOL
[timescale_timescaledb]
name=timescale_timescaledb
baseurl=https://packagecloud.io/timescale/timescaledb/el/7/\$basearch
repo_gpgcheck=1
gpgcheck=0
enabled=1
gpgkey=https://packagecloud.io/timescale/timescaledb/gpgkey
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
metadata_expire=300
EOL
sudo yum update -y
```

2. Установите пакеты для соответствующей версии P

```
yum install -y timescaledb-postgresql-12
```

3. Настройте Ваш PostgreSQL

```
timescaledb-tune --pg-config=/usr/pgsql-12/bin/pg_config
```


ПОДГОТОВЬТЕ ВАШУ БД ДЛЯ TIMESCALEDB

1. Запустите Zabbix server для обновления схемы БД
2. Остановите Zabbix сервер
3. Включите расширение TimescaleDB:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

4. Запустите скрипт timescaledb.sql находящийся в database/postgresql

```
cd /usr/share/doc/zabbix-server-pgsql-5.0.0/  
cat timescaledb.sql | sudo -u zabbix psql zabbix
```

5. Ждите. Миграция уже существующих исторических данных и трендов может занять долгое время. Zabbix сервер и frontend должны быть остановлены на время миграции.
6. Как только миграция данных закончится – запускайте Zabbix сервер

ПОДГОТОВЬТЕ ВАШУ БД ДЛЯ TIMESCALEDB

Ждите. Миграция уже существующих исторических данных и трендов может занять долгое время. Zabbix сервер и frontend должны быть остановлены на время миграции.

```
[root@localhost zabbix-server-pgsql-5.0.0]# cat timescaledb.sql | sudo -u zabbix psql zabbix
NOTICE: migrating data to chunks
DETAIL: Migration might take a while depending on the amount of data.
      create_hypertable
-----
(1,public,history,t)
(1 row)

NOTICE: migrating data to chunks
DETAIL: Migration might take a while depending on the amount of data.
      create_hypertable
-----
(2,public,history_uint,t)
(1 row)

      create_hypertable
-----
(3,public,history_log,t)
(1 row)

      create_hypertable
-----
(4,public,history_text,t)
(1 row)

NOTICE: migrating data to chunks
DETAIL: Migration might take a while depending on the amount of data.
      create_hypertable
```

ПОДГОТОВЬТЕ ВАШУ БД ДЛЯ TIMESCALEDDB

1. Убедитесь, что вывод выглядел не так:

```
ERROR: column "db_extension" of relation "config" does not exist
LINE 1: UPDATE config SET db_extension='timescaledb',hk_history_glob...
                        ^

ERROR: column "compression_status" of relation "config" does not exist
LINE 1: UPDATE config SET compression_status=1,compress_older='7d';
                        ^
```

ПРОВЕРЬТЕ НАСТРОЙКИ **HOUSEKEEPING**

History

Enable internal housekeeping ☒

Override item history period ☒

* Data storage period

Trends

Enable internal housekeeping ☒

Override item trend period ☒

* Data storage period

History and trends compression

Enable compression ☒

* Compress records older than

[Update](#)

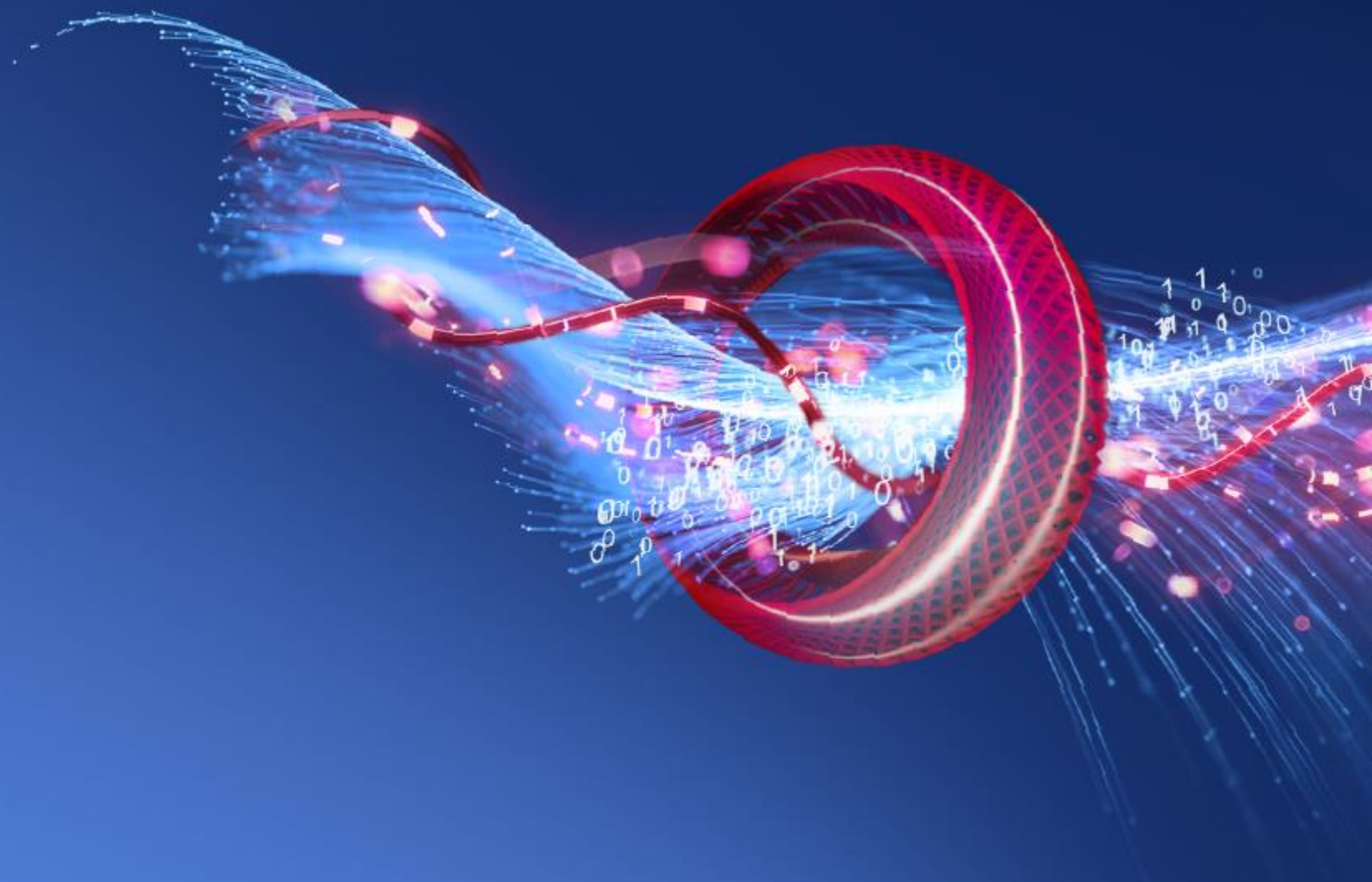
[Reset defaults](#)

ЧТО ДАЛЬШЕ?

Наслаждайтесь
мониторингом!

05

РЕЗУЛЬТАТЫ

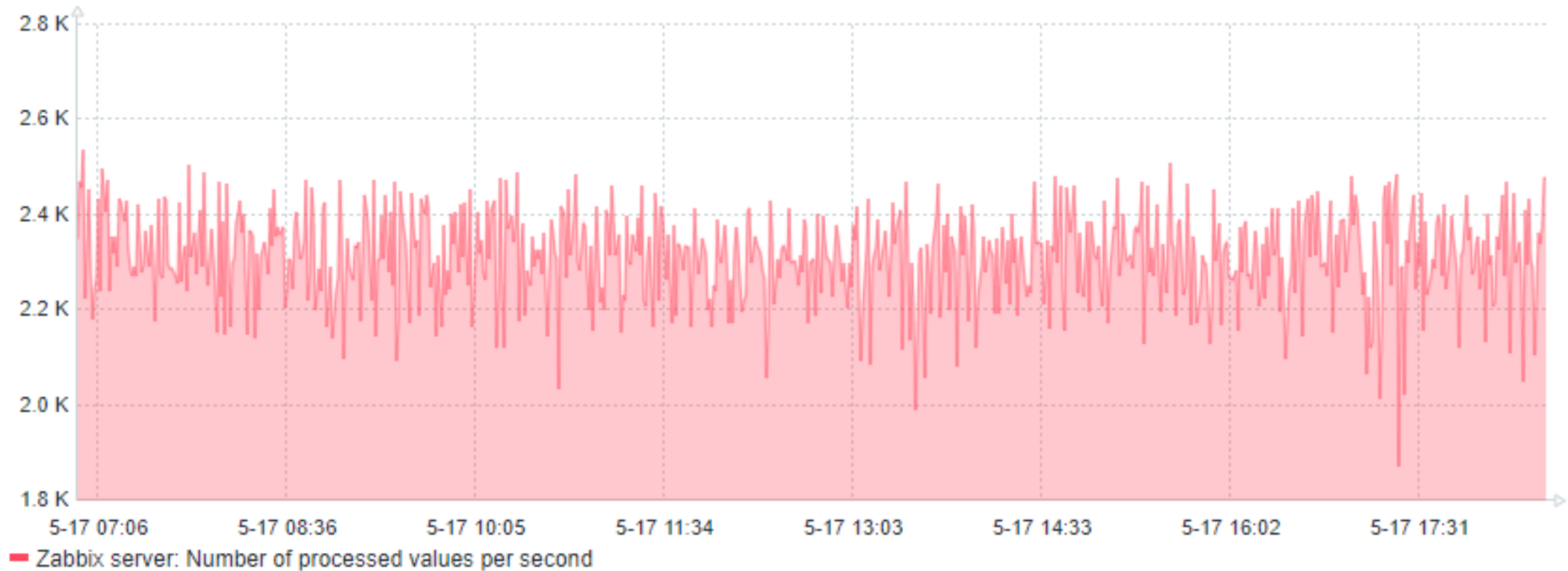


ПОХВАСТАЙТЕСЬ РЕЗУЛЬТАТАМИ



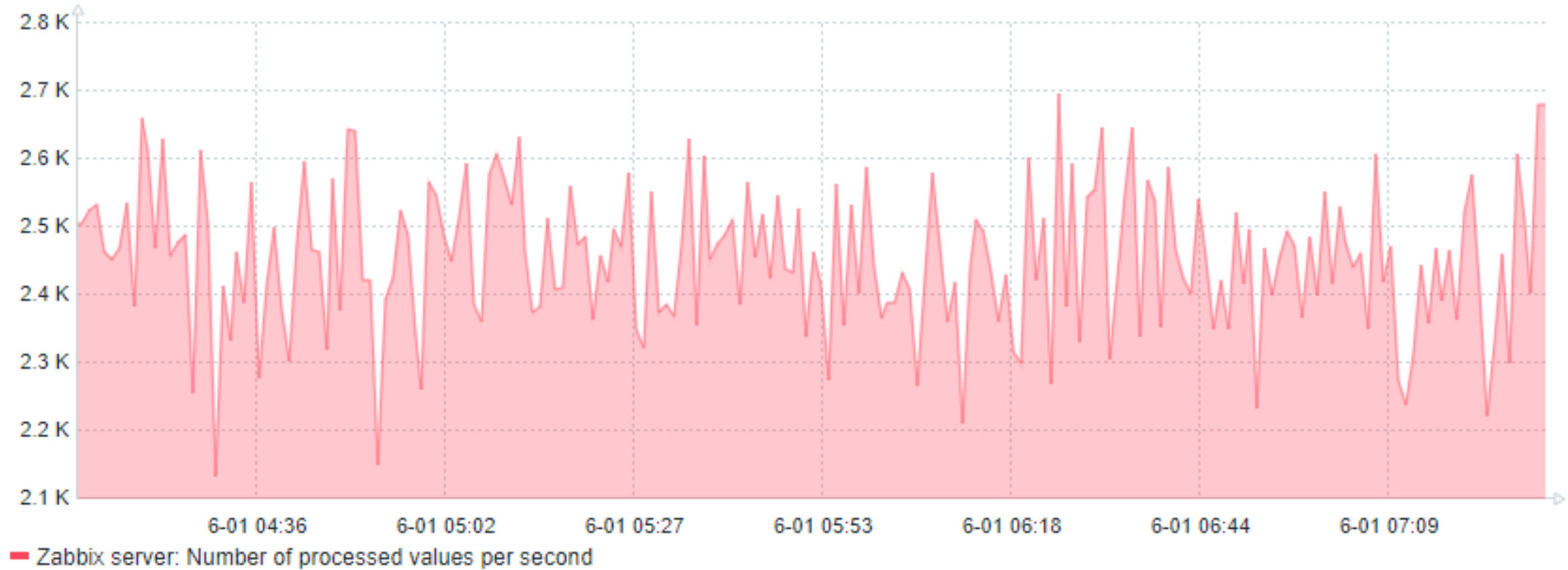
ПРОИЗВОДИТЕЛЬНОСТЬ ДО

Graph



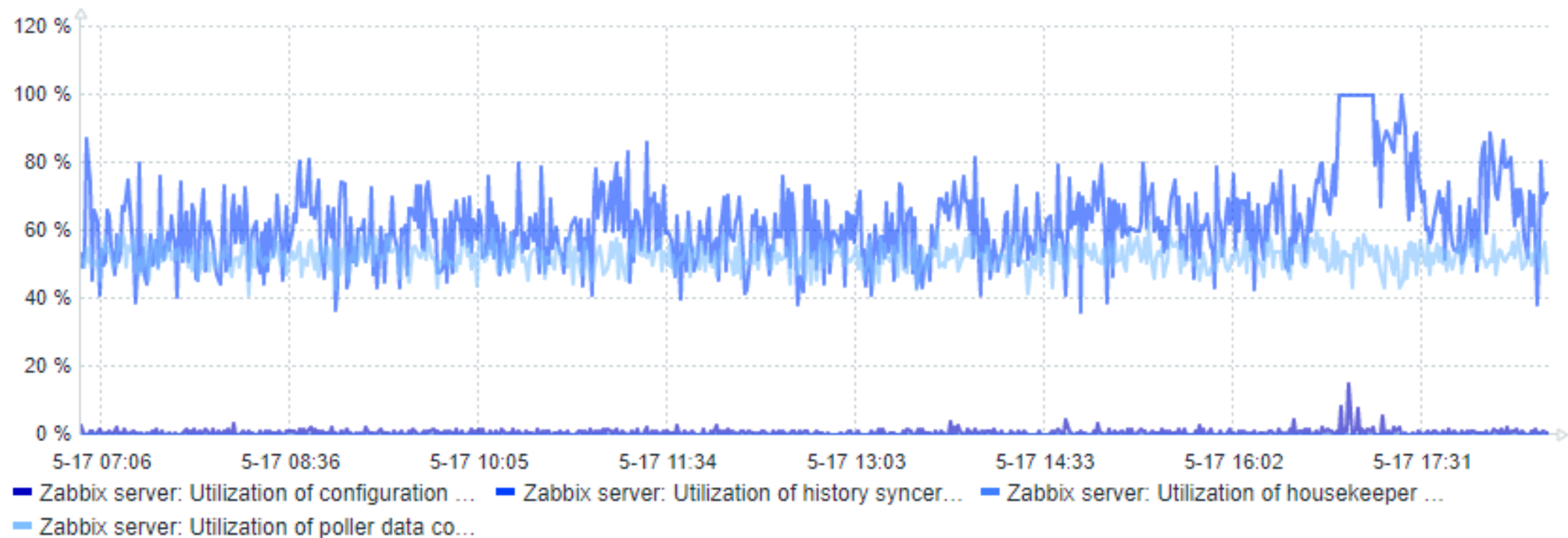
ПРОИЗВОДИТЕЛЬНОСТЬ ПОСЛЕ

Graph



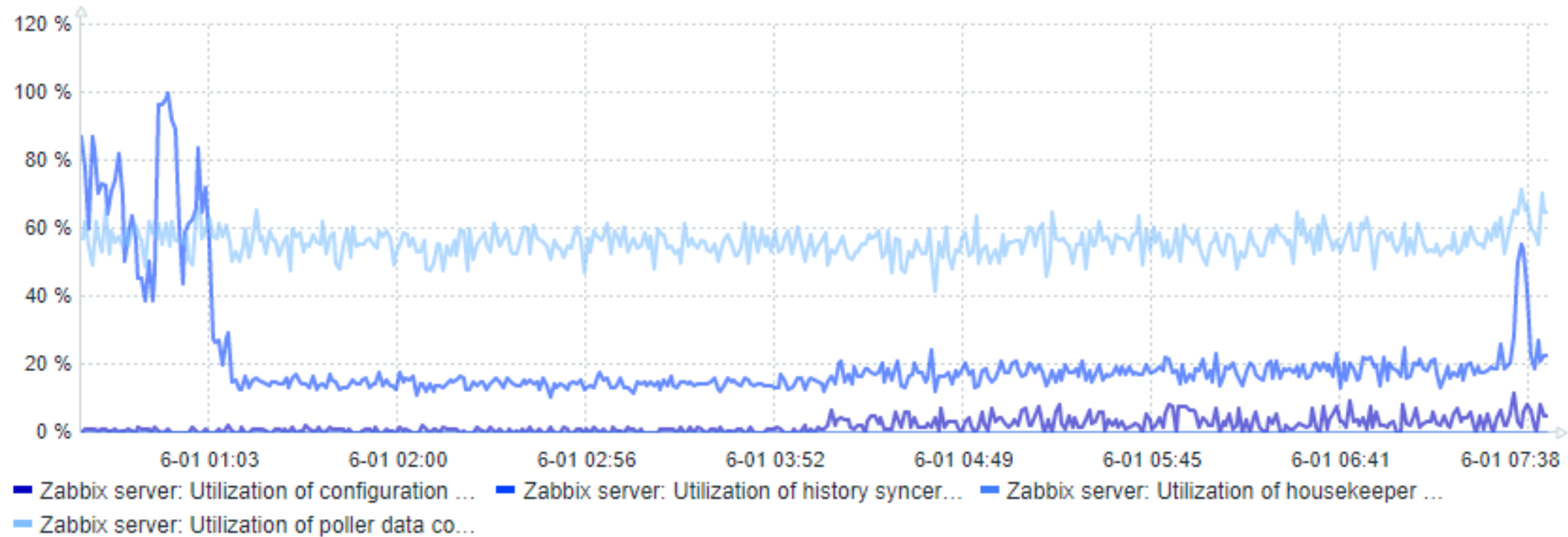
ПРОИЗВОДИТЕЛЬНОСТЬ ДО

Graph



ПРОИЗВОДИТЕЛЬНОСТЬ ПОСЛЕ

Graph



РАЗМЕР БАЗЫ ДАННЫХ ДО

```
zabbix=> SELECT pg_size_pretty( pg_database_size('zabbix') );  
pg_size_pretty  
-----  
30 GB  
(1 row)
```

```
zabbix=> SELECT pg_size_pretty( pg_database_size('zabbix') );  
pg_size_pretty  
-----  
300 GB  
(1 row)
```


РАЗМЕР БАЗЫ ДАННЫХ ПОСЛЕ

```
zabbix=> SELECT pg_size_pretty( pg_database_size('zabbix') );  
pg_size_pretty  
-----  
5818 MB  
(1 row)
```

```
zabbix=> SELECT pg_size_pretty( pg_database_size('zabbix') );  
pg_size_pretty  
-----  
65 GB  
(1 row)
```

ПОСЛЕСЛОВИЕ (НЕМНОГО ОХЛАЖДЕНИЯ)

- ⊙ **Всё еще экспериментально**
- ⊙ **Операции изменения сжатых частей(inserts, deletes, updates) не возможны**
- ⊙ **А что теперь с другими базами данных? (IBM DB2 больше не поддерживается)**
- ⊙ **Ваша средняя система**

Спасибо!

