

# Миграция БД Zabbix с Oracle на PostgreSQL

Александр НИКИТИН,  
старший администратор баз данных, ГК ЦФТ

# О ЧЁМ БУДУ ГОВОРИТЬ?

- История Zabbix в ЦФТ
- Анализ путей миграции
- Миграция
- Тестирование
- Настройка
- Репликация
- Партиционирование
- Расширения и дополнительное ПО



ZABBIX

# ZABBIX В ЦФТ: ИСТОРИЧЕСКИЙ ЭКСКУРС

- 2014 год Zabbix (Oracle)
- Рост (на март 2019 г. - 650 тысяч айтемов, ~4000 nvrps, БД 2 Тб)
- Проблемы
- Настройка
- Проблемы
- Октябрь 2018 года – задача по переходу на другую БД



**Problem**  
**Analysis**  
**Solution**

- Ora2pg



<https://habr.com/ru/company/custis/blog/262605/>

- ANTLR  
(ANother Tool  
for Language  
Recognition)



## Ora2pg

```
ora2pg --init_project zab2pg
```

```
ora2pg.conf:
```

- ORACLE\_HOME /u/app/oracle/product/11.2.0
- ORACLE\_DSN dbi:Oracle:host=192.168.77.90;sid=zabdb;port=1521
- ORACLE\_USER zx
- ORACLE\_PWD SuperPassword1
- PG\_DSN dbi:Pg:dbname=zabbix;host=127.0.0.1;port=5432
- PG\_USER postgres
- PG\_PWD SuperPassword2



<http://ora2pg.darold.net/documentation.html>

## Ora2pg

- `time ora2pg -b ./data -c ./config/ora2pg.conf -j10 -J10 -P8 -l ./log.log -t COPY -o data.sql`

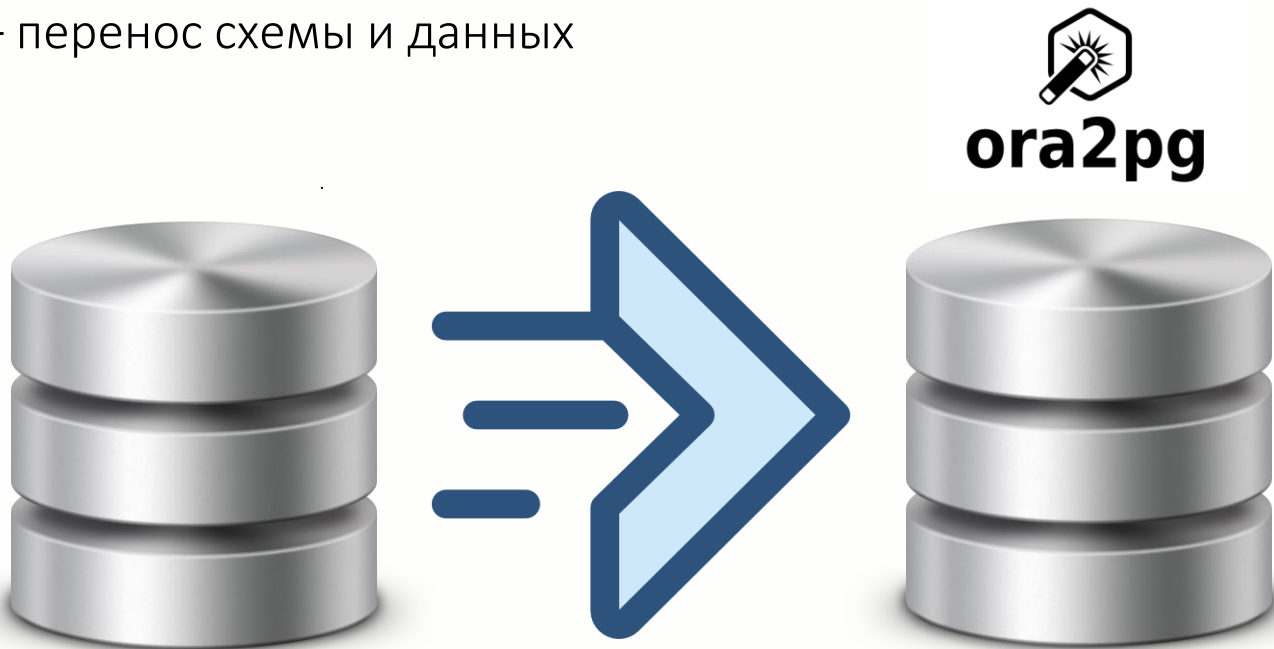
-j | --jobs num : Number of parallel process to send data to PostgreSQL

-J | --copies num : Number of parallel connections to extract data from Oracle

-P | --parallel num: Number of parallel tables to extract at the same time

# МИГРАЦИЯ

Ora2pg – перенос схемы и данных



# МИГРАЦИЯ

Схема БД от самого Zabbix

Ora2pg – перенос только данных

ZABBIX



ora2pg







[ГРАБЛИ] №1

# МИГРАЦИЯ

## 1 Zabbix 3.4 database structure

### 2 Columns by data type

<b>Zabbix</b>	<b>PostgreSQL</b>	<b>Oracle</b>
id	bigint	number(20)
integer	integer	number(10)
double	numeric(16,4)	number(20,4)
bigint	numeric(20)	number(20)
char	char	nvarchar2
time	integer	number(10)
varchar	varchar	nvarchar2
text	text	nvarchar2(2048)
shorttext	text	nvarchar2(2048)

# МИГРАЦИЯ

## 1 Zabbix 3.4 database structure

### 2 Columns by data type

Zabbix	PostgreSQL	Oracle
id	bigint	number(20)
integer	integer	number(10)
double	numeric(16,4)	number(20,4)
bigint	numeric(20)	number(20)
char	char	nvarchar2
time	integer	number(10)
varchar	varchar	nvarchar2
text	text	nvarchar2(2048)
shorttext	text	nvarchar2(2048)

# МИГРАЦИЯ

- Изменяем скрипт создания БД
- Создаём дамп:  

```
pg_dump -f /dbbackup/backup/zabbix.dmp -d zabbix --create
```
- Делим файл на 2 части – создание таблиц и создание индексов



[ГРАБЛИ] №2

# МИГРАЦИЯ

## ORACLE®

```
CREATE TABLE maintenances  
(maintenanceid number(20) NOT NULL,  
name nvarchar2(128) DEFAULT "",  
...  
description nvarchar2(2048) DEFAULT "",  
active_since number(10) DEFAULT '0' NOT NULL,  
active_till number(10) DEFAULT '0' NOT NULL,  
PRIMARY KEY (maintenanceid));
```



## PostgreSQL

```
CREATE TABLE maintenances  
(maintenanceid bigint NOT NULL,  
name varchar(128) DEFAULT "" NOT NULL,  
...  
description text DEFAULT "" NOT NULL,  
active_since integer DEFAULT '0' NOT NULL,  
active_till integer DEFAULT '0' NOT NULL,  
PRIMARY KEY (maintenanceid));
```

# МИГРАЦИЯ

## ORACLE®

```
CREATE TABLE maintenances  
(maintenanceid number(20) NOT NULL,  
name nvarchar2(128) DEFAULT "",  
...  
description nvarchar2(2048) DEFAULT "",  
active_since number(10) DEFAULT '0' NOT NULL,  
active_till number(10) DEFAULT '0' NOT NULL,  
PRIMARY KEY (maintenanceid));
```



## PostgreSQL

```
CREATE TABLE maintenances  
(maintenanceid bigint NOT NULL,  
name varchar(128) DEFAULT "" NOT NULL,  
...  
description text DEFAULT "" NOT NULL,  
active_since integer DEFAULT '0' NOT NULL,  
active_till integer DEFAULT '0' NOT NULL,  
PRIMARY KEY (maintenanceid));
```

# МИГРАЦИЯ

PostgreSQL:

```
alter table maintenances
```

```
    alter column description drop not null;
```



# МИГРАЦИЯ

37 таблиц, 185 полей...

Создаём скрипт `before_insert`

Создаём скрипт `after_insert` – замена null значений +  
возврат ограничений

Не забываем про индексы

# МИГРАЦИЯ

Тип данных Nclob

(~280 млн. строк только в history\_text)– 2 суток при самом благоприятном раскладе: параллельность и быстрый сервер

- 1) ProLiant DL360p Gen8 Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz – 2,5 суток
- 2) SUN FIRE X4170 M3 Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHz – 2 суток

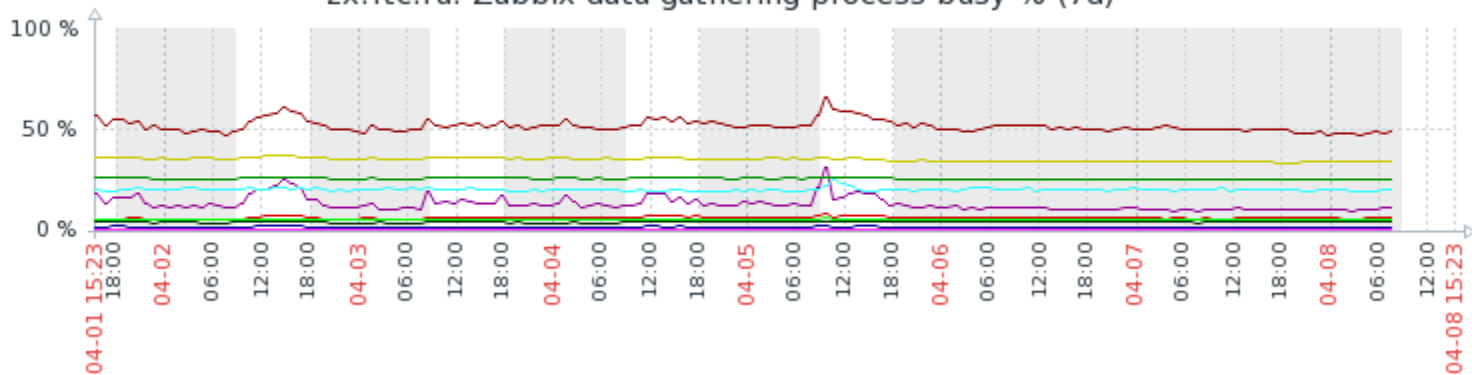
# МИГРАЦИЯ

- Полная миграция 2-5 суток
- Без исторических данных – 30-40 минут на ВСЁ!



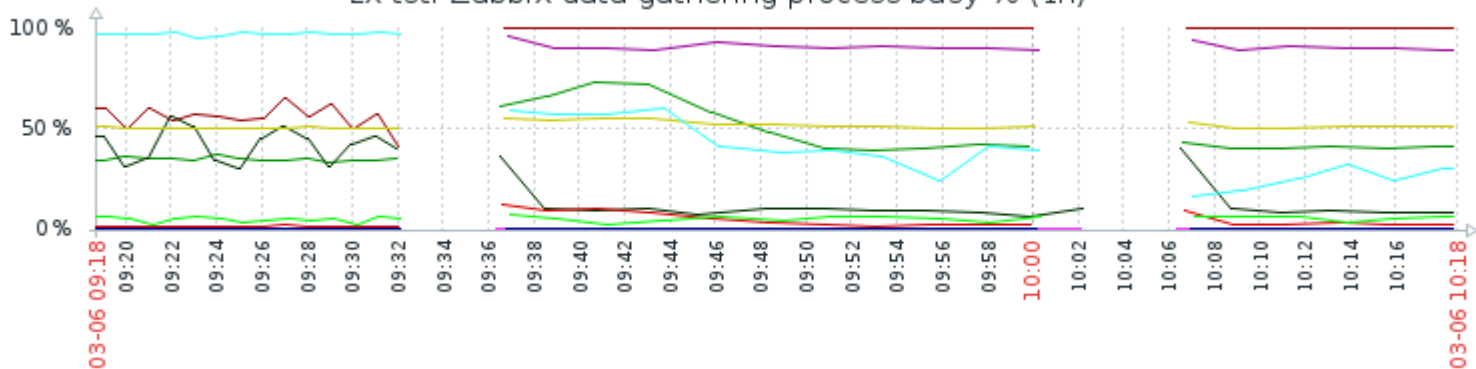
# ТЕСТИРОВАНИЕ

zx.ftc.ru: Zabbix data gathering process busy % (7d)



ORACLE®

zx-tst: Zabbix data gathering process busy % (1h)



PostgreSQL

# ТЮНИНГ БД

- Mamonsu (report, tune)

```
REPORT
Version      : 2.3.4
Platform     : linux2
Python       : 2.6.6 (r266:84292, Aug 18 2016, 08:36:59) [GCC 4.4.7 20120313 (Red Hat 4.4.7-17)]

SYSTEM
Date         : 2019-04-24 08:30:32 UTC (local TZ:+07 +0700)
Host         : zx
Uptime       : 15:30:32 up 30 days, 5:11, 2 users, load average: 13.72, 13.98, 14.92
Boot time    : Mon Mar 25 10:18:33 +07 2019
System       : N/A; N/A; N/A (N/A)
Serial       : N/A
Release      : "Oracle Linux Server release 6.10"
Kernel:
  name: 3.8.13-118.31.1.el6uek.x86_64
  cmdline: ro root=UUID=fd082310-a71d-4a4f-8bcd-d7394c26b126 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=267M@0M
Arch         : CPU = 64-bit, OS = 64-bit
Virt         : N/A

PROCESSORS
Total        : physical = 1, cores = 22, virtual = 22, hyperthreading = False
Speed        : 2494.036 MHz
Model        : Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50GHZ
Flags        : pae (physical address extensions), ht (hyper-threading), lm (64-bit), constant_tsc (Constant Time Stamp Counter), aes (AES-NI)
Cache        : 15360 KB
Bench        : 3.62
```

# ТЮНИНГ БД

<http://pgconfigurator.cybertec.at/>



Select your version of PostgreSQL:

11

GB of RAM in your server:

0 64 2,048

Number of CPUs:

1 8 72

Disk Type:

HDD

Number of disks:

1 4 32

```
# DISCLAIMER - Software and the resulting config files are provided "AS IS" - IN NO EVENT SHALL
# BE THE CREATOR LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
# DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION.

# Connectivity
max_connections = 100
superuser_reserved_connections = 3

# Memory Settings
shared_buffers = '16384 MB'
work_mem = '64 MB'
maintenance_work_mem = '620 MB'
huge_pages = try # NB! requires also activation of huge pages via kernel params, see here for more:
                # https://www.postgresql.org/docs/current/static/kernel-resources.html#LINUX-HUGE-PAGE
effective_cache_size = '45 GB'
effective_io_concurrency = 2 # concurrent IO only really activated if OS supports posix_fadvise function

# Monitoring
shared_preload_libraries = 'pg_stat_statements' # per statement resource usage stats
track_io_timing=on # measure exact block IO times
track_functions=pl # track execution times of pl-language procedures if any

# Replication
wal_level = minimal # consider using at least 'replica'
```



☰ POSTGRESQLCO.NF

ВЕРСИЯ POSTGRESQL:

9.1 9.2 9.3 9.4 9.5 9.6 10 11 12

СТАБИЛЬНЫЙ

ПАРАМЕТРЫ ☰ КАТЕГОРИИ

Поиск параметра

- allow\_system\_table\_mods
- application\_name
- archive\_cleanup\_command **+v12**
- archive\_command
- archive\_mode**
- archive\_timeout
- array\_nulls
- authentication\_timeout
- autovacuum

### archive\_mode

Документация / Write-Ahead Log / Archiving / archive\_mode

---

#### PARAMETER INFO

Тип: `enum`

По умолчанию: `off`

Контекст: `postmaster`

Перезапуск: `true`

Значения: `[always, on, off]`

---

#### DESCRIPTION

Allows archiving of WAL files using `archive_command`

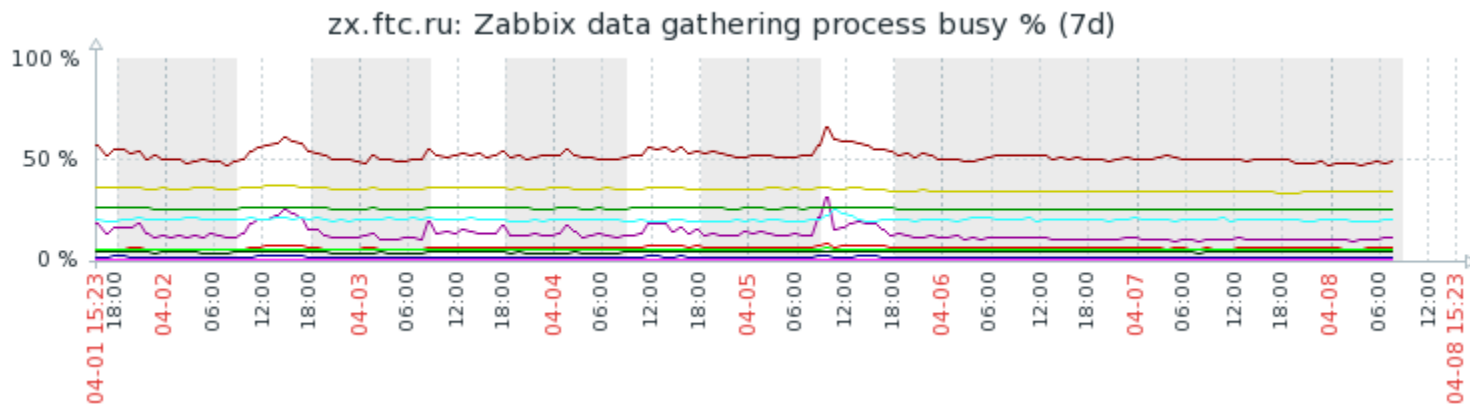
Когда параметр `archive_mode` включён, полные сегменты WAL передаются в хранилище архива командой `archive_command`. Помимо значения `off` (выключающего архивацию) есть ещё два: `on` (вкл.) и `always` (всегда). В обычном состоянии эти два режима не различаются, но в режиме `always` архивация WAL активна и во время восстановления архива, и в резерве. В этом режиме все файлы, восстановленные из архива или полученные при потоковой репликации, будут архивироваться (снова). За подробностями обратитесь к 3.

Параметры `archive_mode` и `archive_command` разделены, чтобы команду архивации (`archive_command`) можно было изменять, не отключая режим архивации. Этот параметр можно задать только при запуске сервера. Режим архивации нельзя включить, когда установлен минимальный уровень WAL (`wal_level`): имеет значение `minimal`.

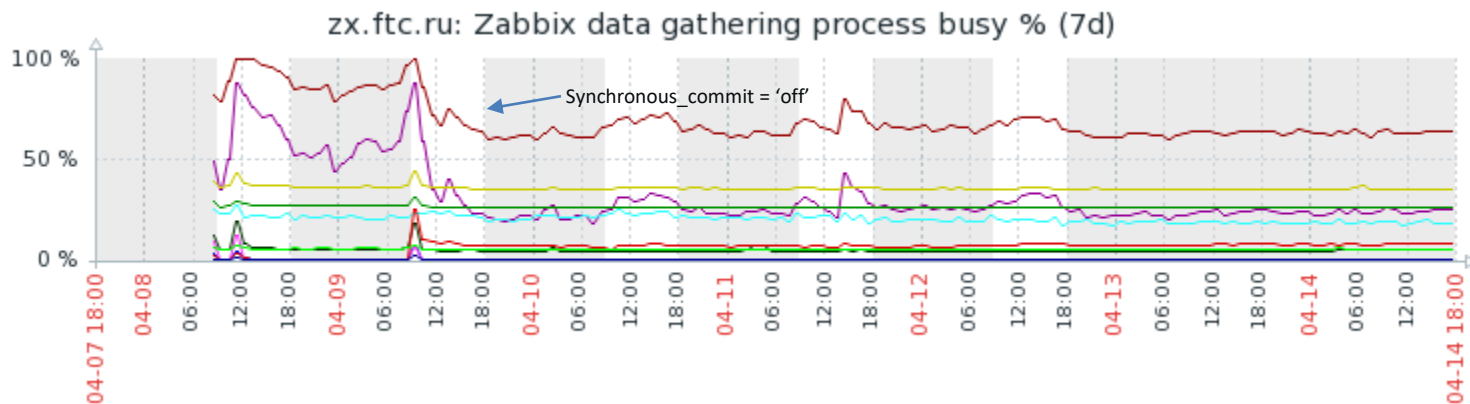
[ОФИЦИАЛЬНАЯ ДОКУМЕНТАЦИЯ \[EN\]](#) перевод [POSTGRESPRO](#)

[ПОИСК В ИСХОДНОМ КОДЕ](#)

# ТЕСТИРОВАНИЕ



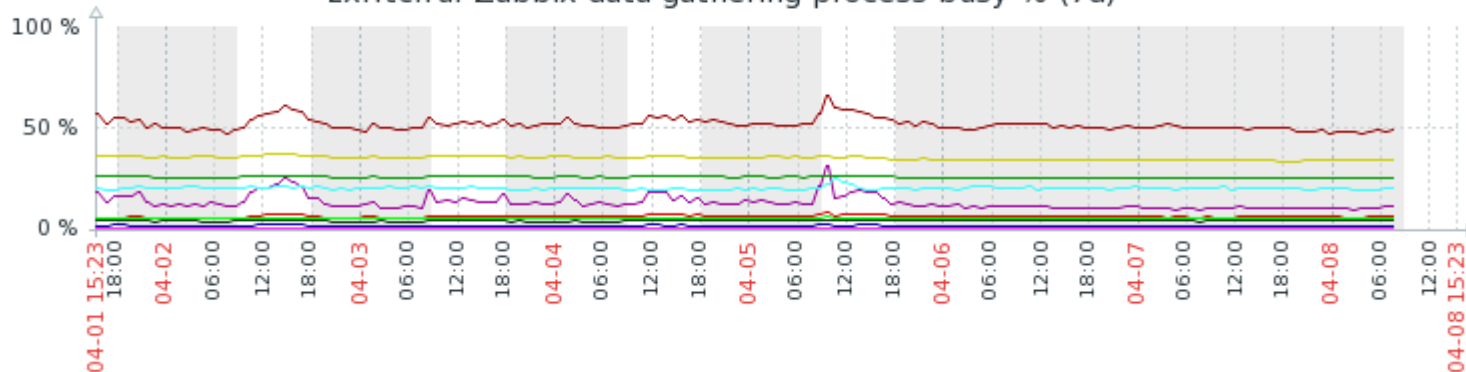
ORACLE®





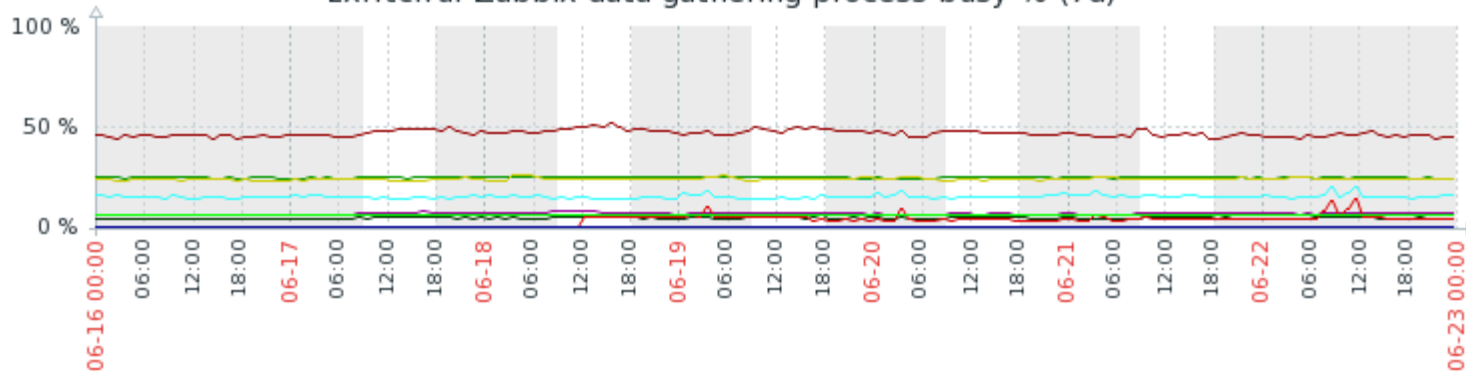
# ТЕСТИРОВАНИЕ

zx.ftc.ru: Zabbix data gathering process busy % (7d)



ORACLE®

zx.ftc.ru: Zabbix data gathering process busy % (7d)



PostgreSQL

# РЕПЛИКАЦИЯ

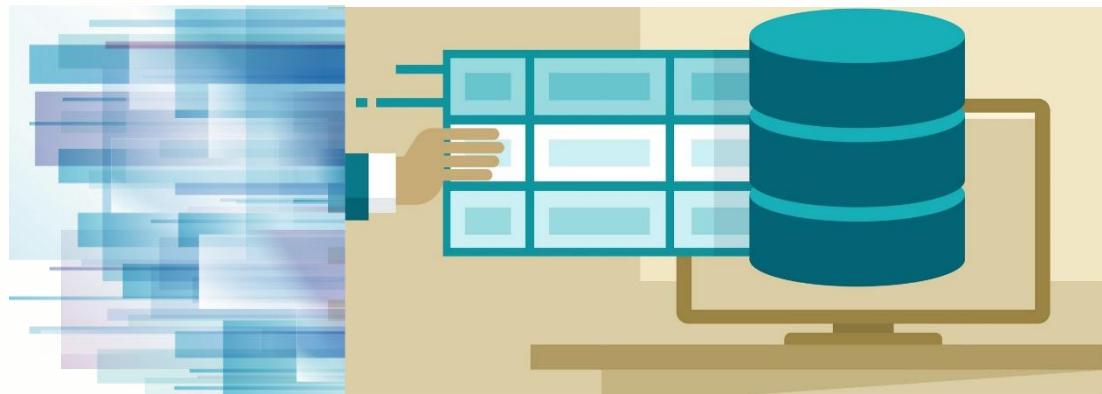
- Поточковая репликация с использованием слота репликации
- Доклад Алексея Лесовского SECON 2017 г.



<https://www.youtube.com/watch?v=-AEG35vWno>

# ПАРТИЦИОНИРОВАНИЕ

- Стандартный подход
- Pg\_pathman
- Pg\_partman



# РАСШИРЕНИЯ

- amcheck
- auth\_delay
- pg\_stat\_statements
- pg\_prewarm
  
- pg\_profile
- pg\_pathman



# РАСШИРЕНИЯ

- amcheck
- auth\_delay
- pg\_stat\_statements
- ~~• pg\_prewarm~~
- pg\_profile
- pg\_pathman



# ДОПОЛНИТЕЛЬНОЕ ПО

- pgcenter
- pgbouncer
- pg\_probackup
- ASH-Viewer
- pgAdmin



# РЕЗЮМЕ

- История Zabbix в ЦФТ
- Анализ путей миграции
- Миграция
- Тестирование
- Настройка
- Репликация
- Партиционирование
- Расширения и дополнительное ПО



**ZABBIX**

# СПАСИБО ЗА ВНИМАНИЕ!

Александр НИКИТИН,  
старший администратор баз данных, ГК ЦФТ  
[al.nikitin@cft.ru](mailto:al.nikitin@cft.ru)

**ЦФТ** ЦЕНТР  
ФИНАНСОВЫХ  
ТЕХНОЛОГИЙ

