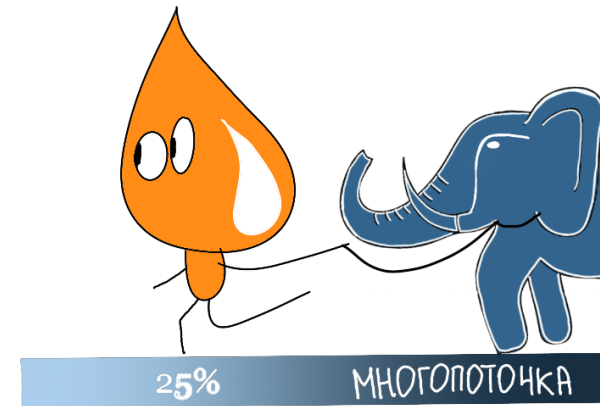
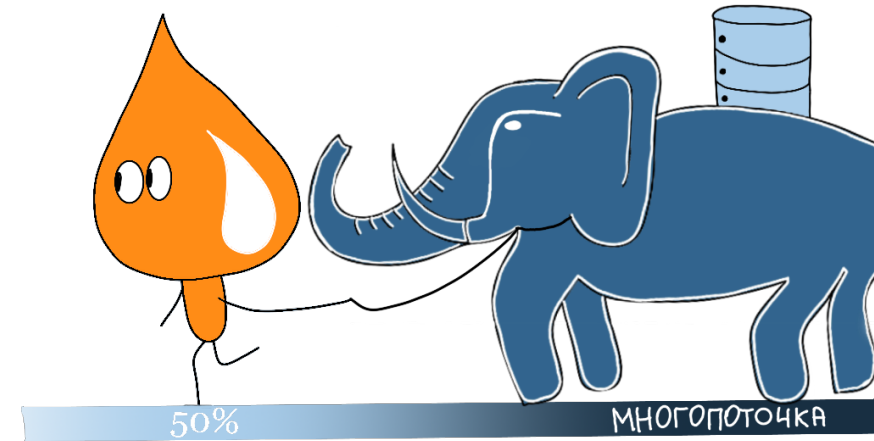
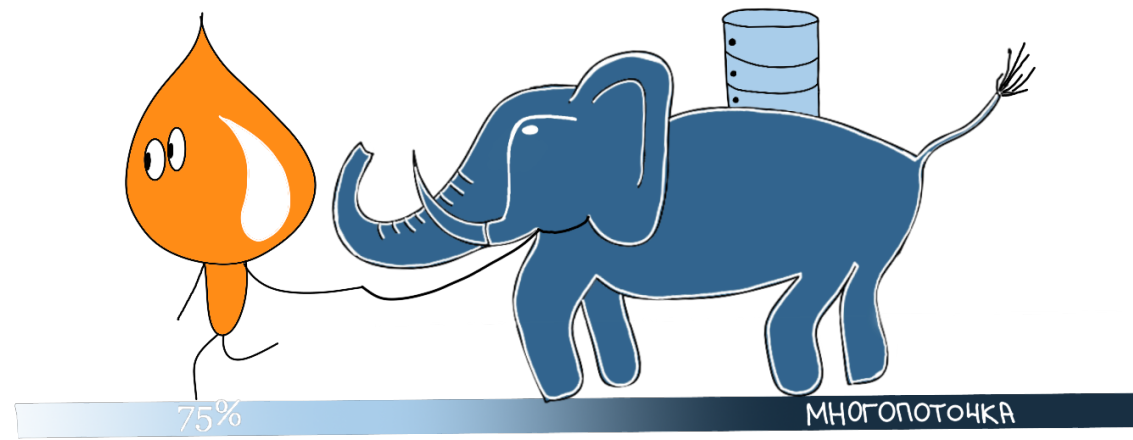


Как деплоить в 5 раз быстрее
или рассказ
о нашей реализации
параллельного выполнения миграций
в Liquibase

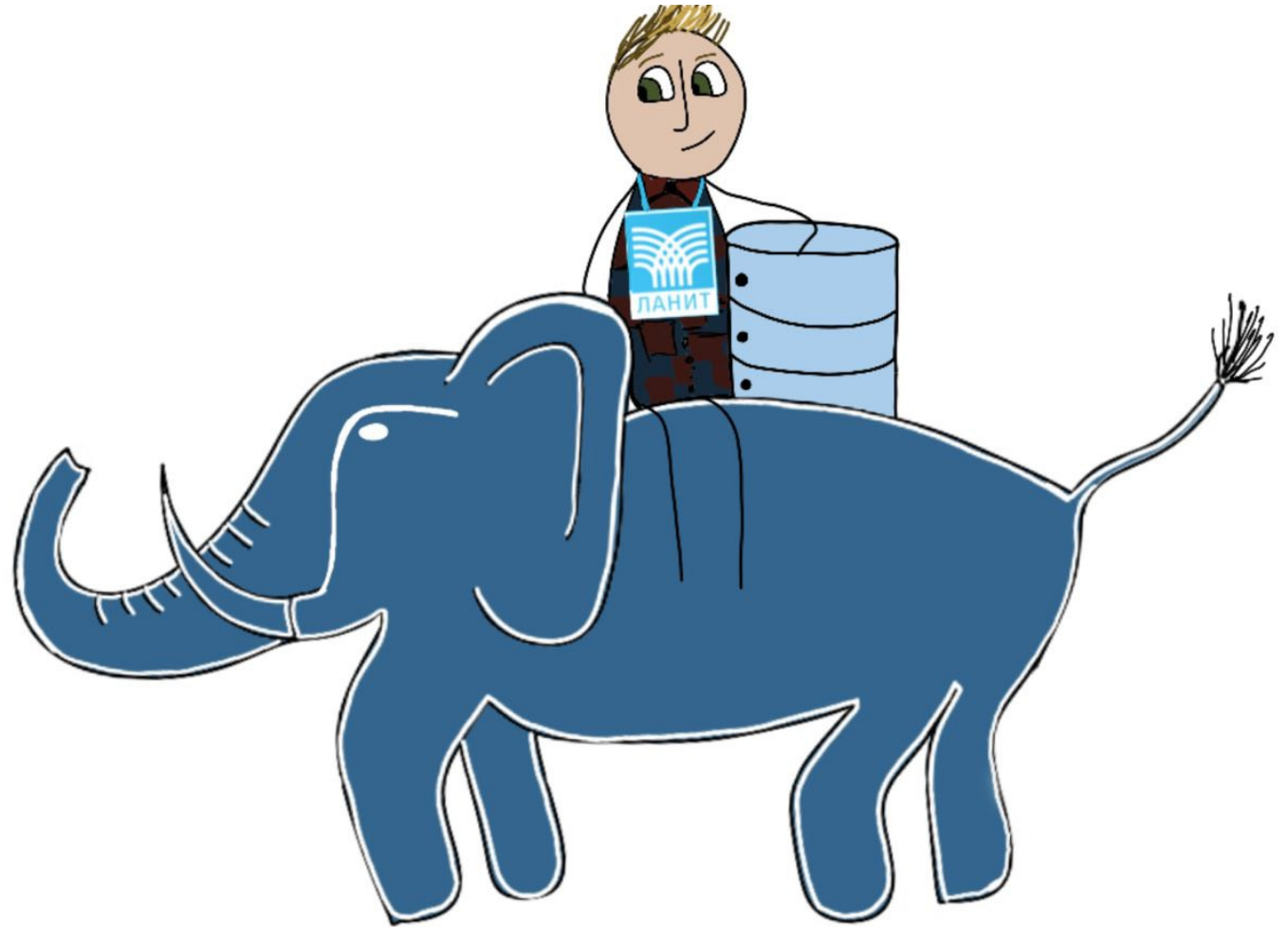


Анатолий Солдатов
PGConf.Russia 2018



Обо мне

- Разработчик БД
- Работаю на проекте ГИС ЖКХ в ЛАНИТ
- Люблю все автоматизировать



О ГИС ЖКХ

- Все регионы
- 20 миллионов домов
- 125 миллионов лицевого счетов

Статистика внедрения ГИС ЖКХ по субъектам Российской Федерации

Данные по информации, размещенной в ГИС ЖКХ, по состоянию на 03.02.2018 23:57 (МСК, UTC + 3)

Вами выбраны:



87 264 организации



1 522 068 многоквартирных домов



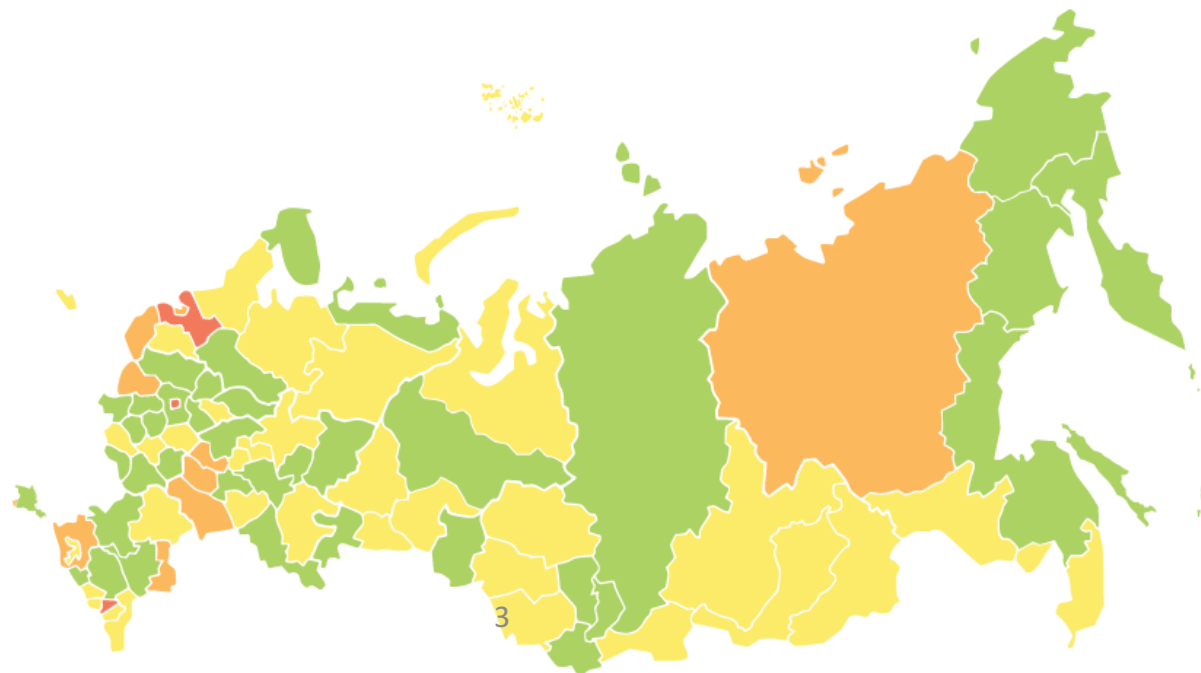
16 720 986 жилых домов

Всего в системе:

87 264 зарегистрированные организации

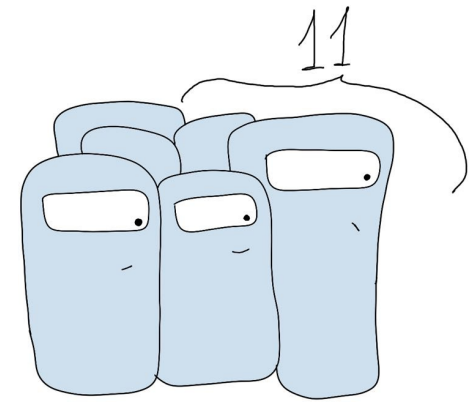
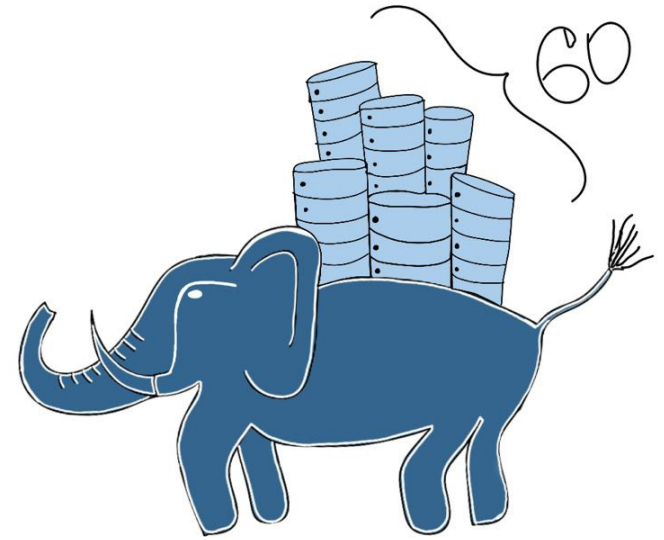
1 522 068 многоквартирных домов

16 720 986 жилых домов



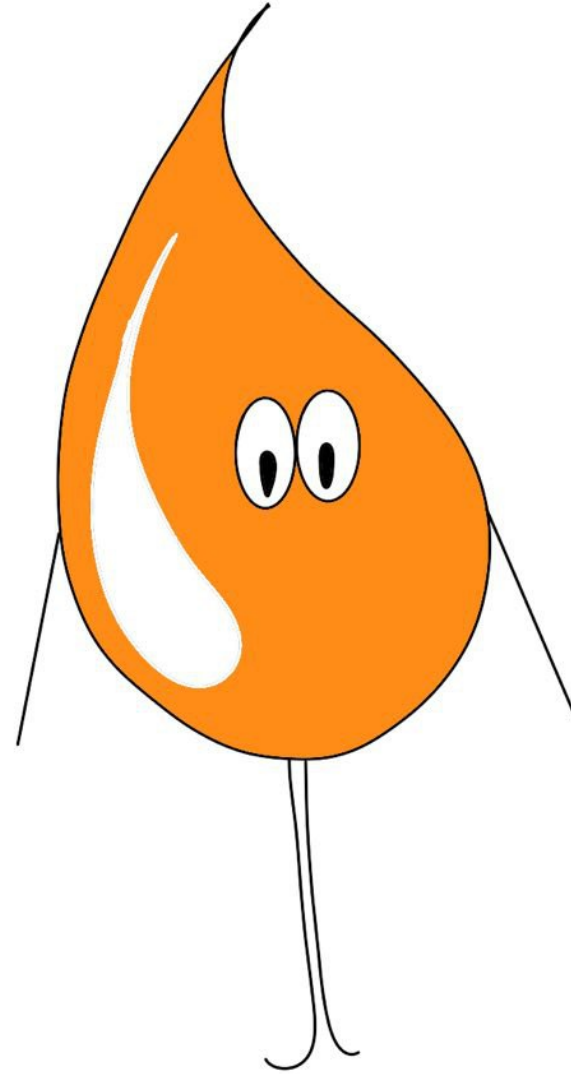
О проекте

- 60 баз данных
- 11 серверов
- 120 млн платежных документов в день
- 30 млн показаний приборов учета в день
- 30 млн фактов оплаты в день



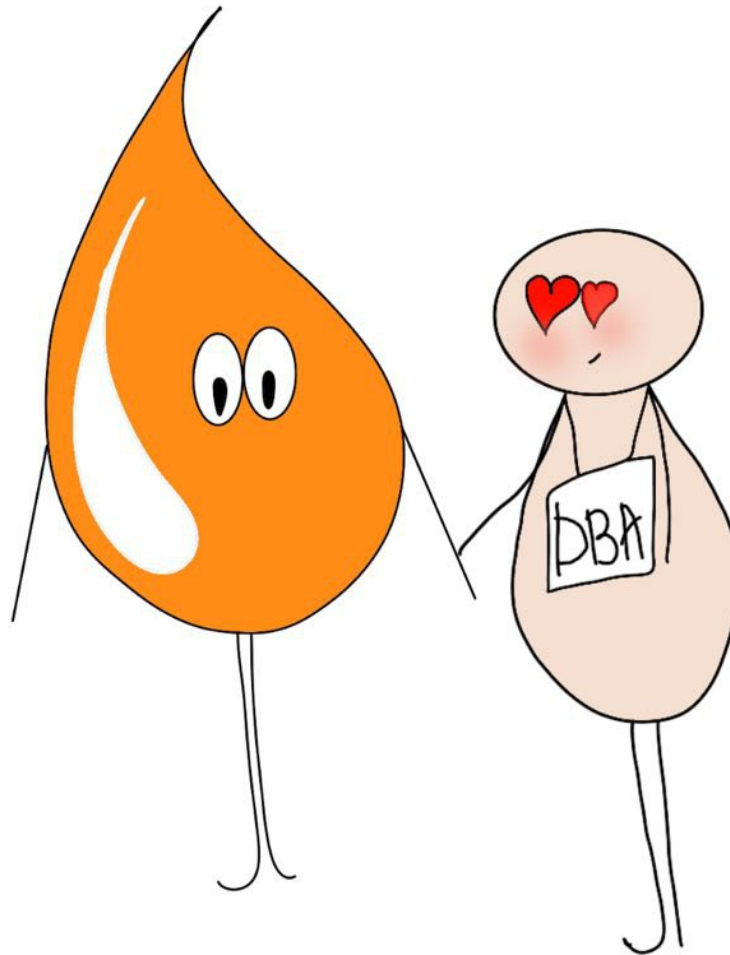
О процессе разработки

- Версионная разработка
- Код java и миграции БД вместе в VCS



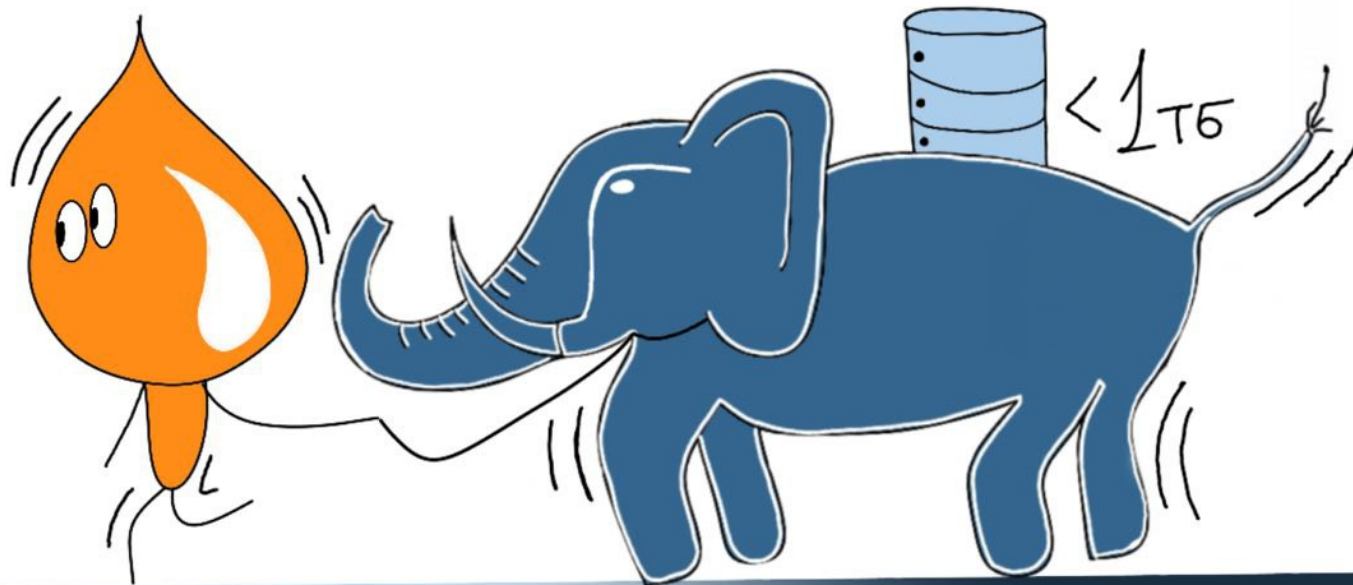
О процессе разработки

- Liquibase помогает DBA



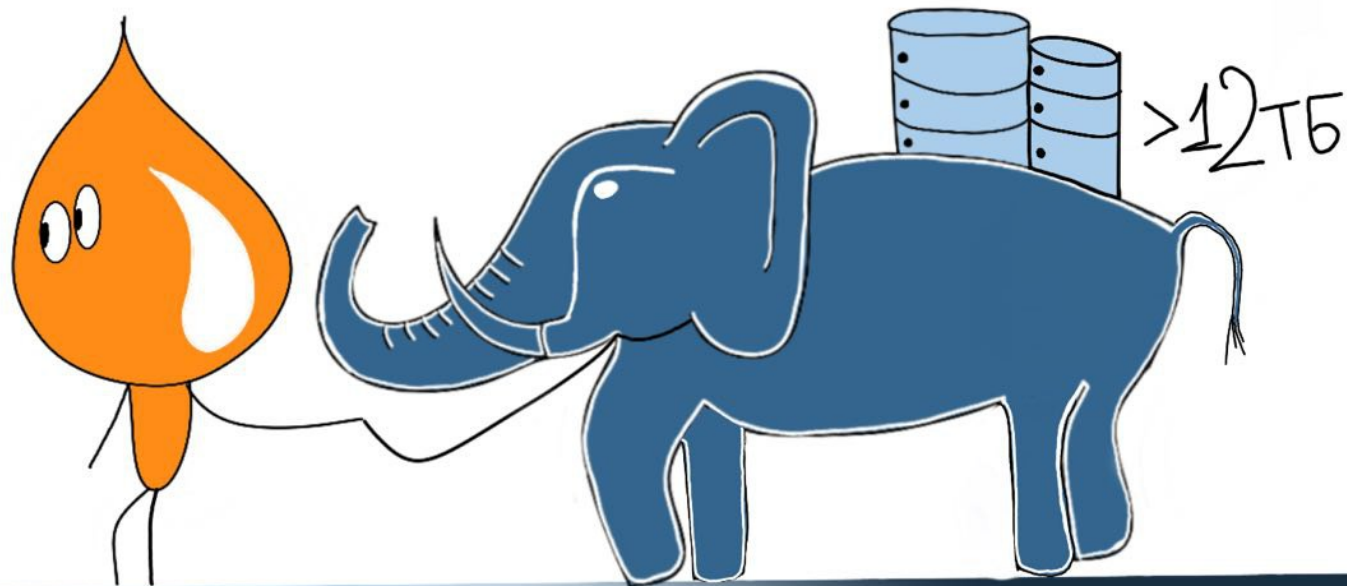
О проблеме

- Сперва был 1 Тб в БД



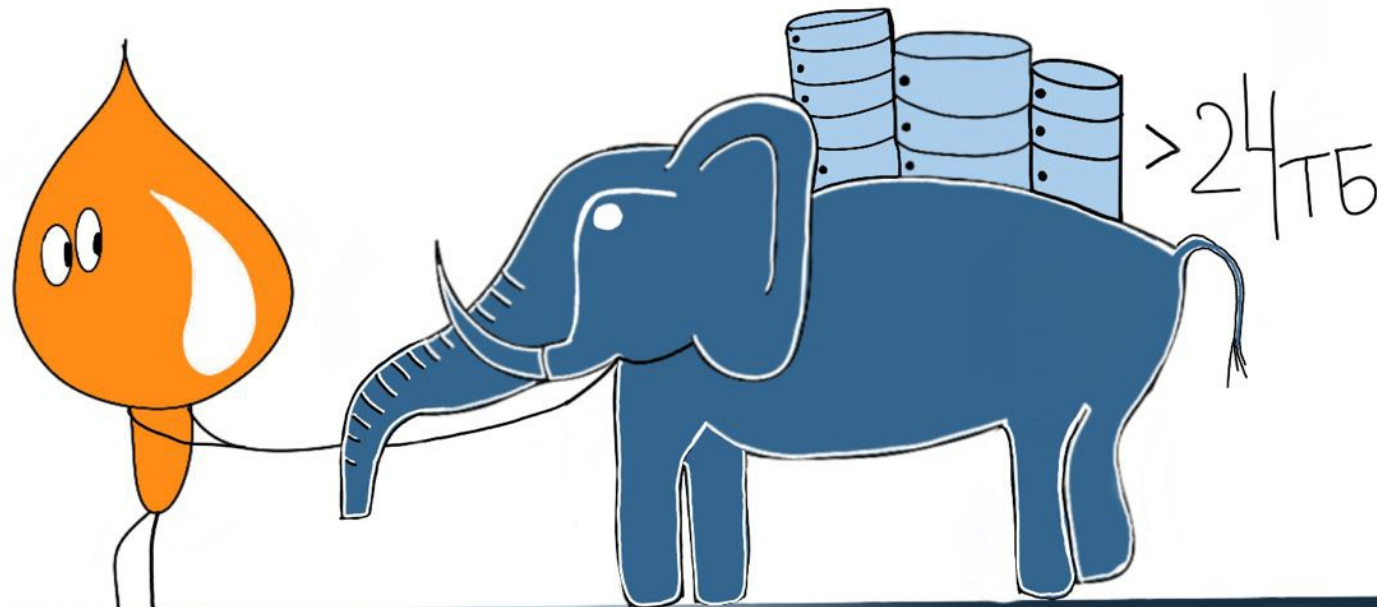
О проблеме

- Затем стало 12 Тб в БД



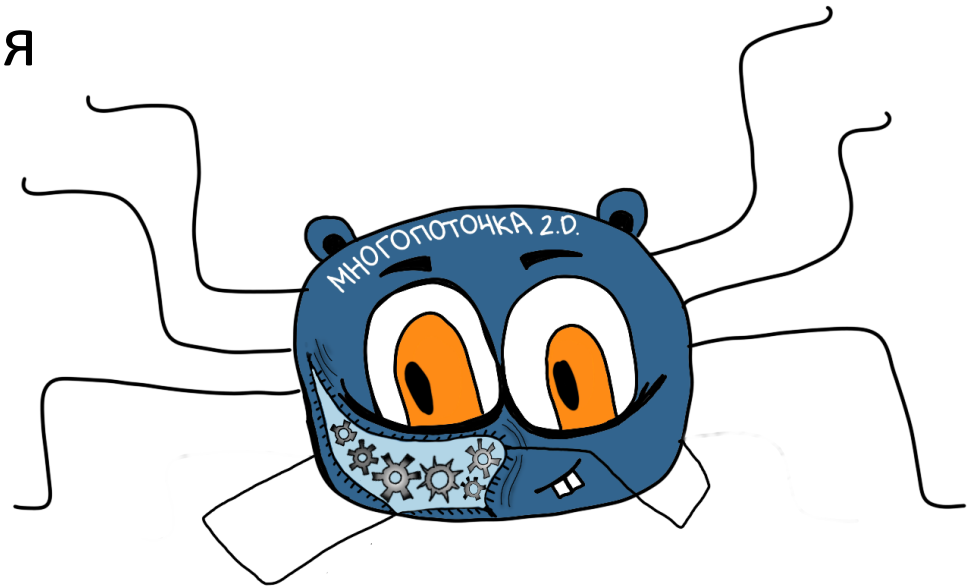
О проблеме

- За последние 3 месяца выросли еще на 12 Тб



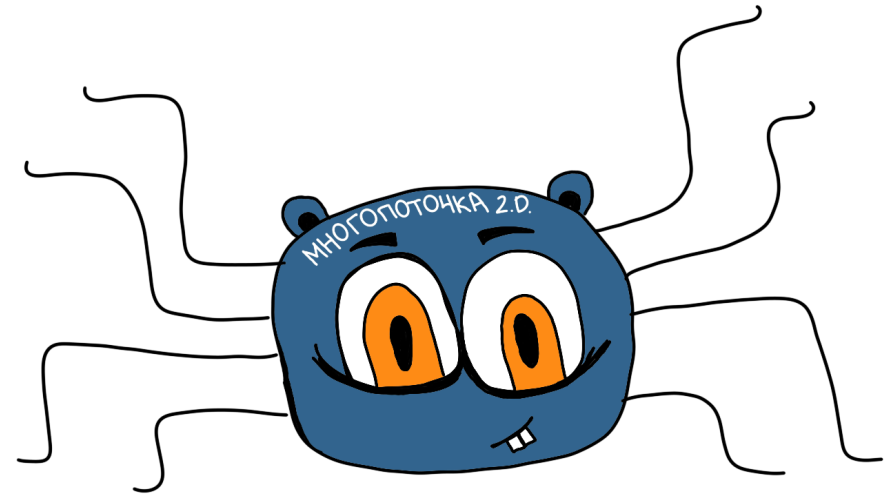
О том, как работает «многопоточка»

- Разделяем большую таблицу на маленькие непересекающиеся диапазоны
- Каждый процесс блокирует свой диапазон



О кейсах применения

- UPDATE таблиц
> 100 миллионов строк
- UPDATE со сложной логикой
- Неблокирующий UPDATE
- Параллельное выполнение
changeSet`ов в Liquibase
- ALTER TABLE ADD COLUMN
+ заполнение данными
- Сложная логика по
обработке данных в отчетах



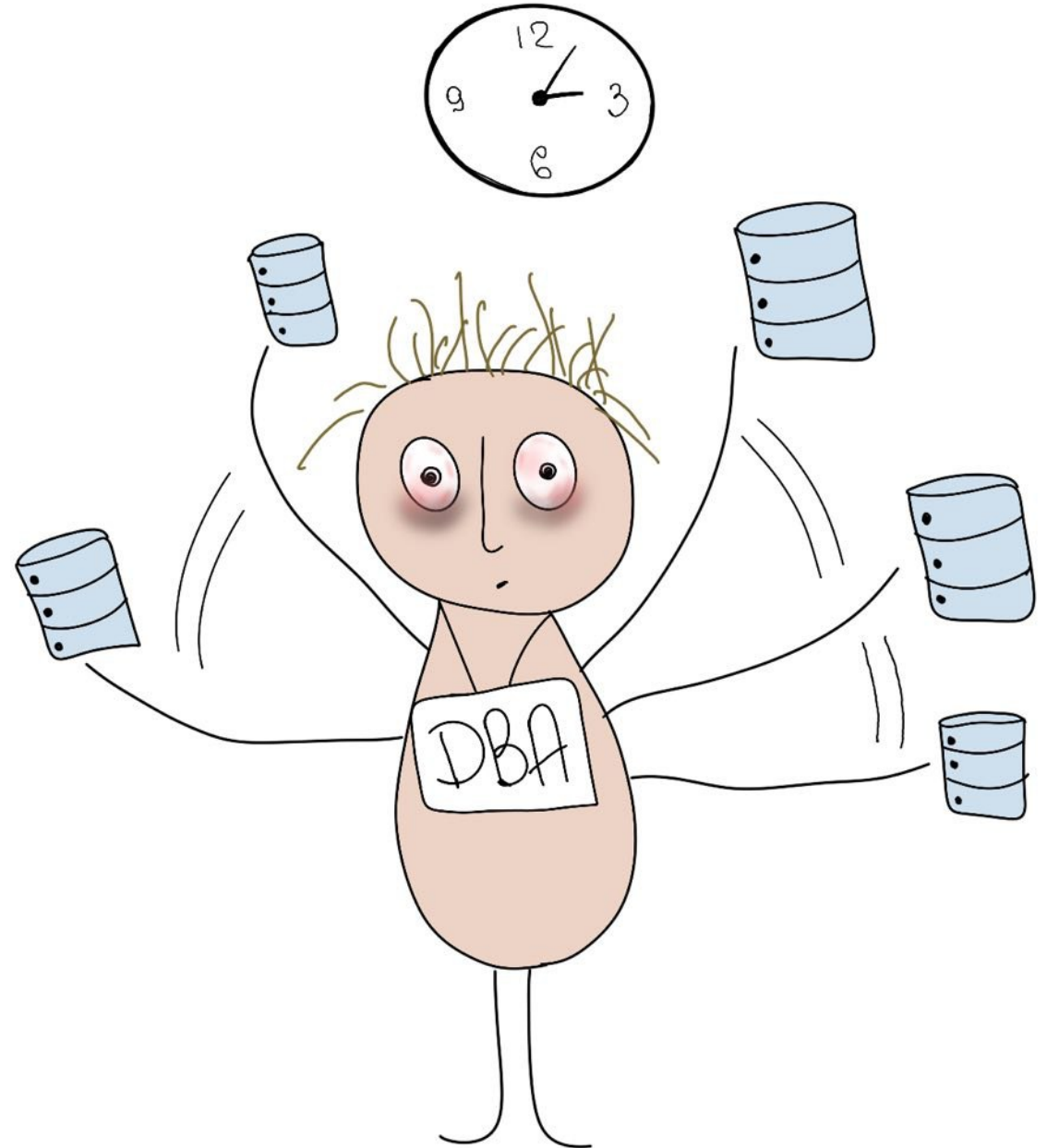
О «многопоточке 1.0»

- Быстро
- Неудобно



Опять о проблемах

- Нет автоматизации
- По ночам хочется спать...



О «МНОГОПОТОЧКЕ 2.0»

- 48 Cores
- 400 GB RAM
- 3 TB SSD
- 32 MB work_mem
- 96 GB shared_buffers



О «многопоточке 2.0»

```
1  SELECT count(1) FROM pgbench_accounts;
2  count
3  -----
4  500000000
5  (1 row)
6
7  SELECT pg_size_pretty(pg_total_relation_size('pgbench_accounts'));
8  pg_size_pretty
9  -----
10  62 Gb
11  (1 row)
12
13  UPDATE pgbench_accounts
14  SET abalance = 42;
15  -- Время выполнения 49 минут
16
17  vacuum full analyze verbose pgbench_accounts;
--
```

О «многопоточке 2.0»

```
1  INSERT INTO public.pg_parallel_tASK (name, threads_count) values ('PGCONF-2018-UPDATE', 25);
2
3  INSERT INTO public.pg_parallel_tASK_statements (tASK_name, sql_statement)
4  SELECT 'PGCONF-2018-UPDATE' tASK_name,
5         FORMAT('UPDATE pgbench_accounts
6                SET abalance = 42
7                WHERE aid >= ''%s'' AND aid <= ''%s'';',
8                MIN(d.aid), MAX(d.aid)) sql_statement
9  FROM (SELECT aid, ntile(25) over (order by aid) part
10        FROM pgbench_accounts) d
11  GROUP BY d.part;
12  -- Время выполнения 10 минут
```

О «многопоточке 2.0»

```
1  INSERT INTO public.pg_parallel_tASK (name, threads_count) values ('PGCONF-2018-UPDATE', 25);
2
3  -- Можно дробить по ctid, но получится неравномерно и нужно чтобы эту таблицу никто не изменял в процессе многопоточки
4  INSERT INTO public.pg_parallel_tASK_statements (tASK_name, sql_statement)
5  SELECT 'PGCONF-2018-UPDATE' tASK_name,
6         FORMAT('UPDATE pgbench_accounts
7                SET abalance = 45
8                WHERE (ctid::text::point)[0]::text > '%s' AND (ctid::text::point)[0]::text <= '%s';',
9                (d.min_ctid), (d.max_ctid)) sql_statement
10 FROM (
11     WITH max_ctid AS (
12         SELECT MAX((ctid::text::point)[0]::int) FROM pgbench_accounts)
13     SELECT generate_series - (SELECT max / 25 FROM max_ctid) AS min_ctid, generate_series AS max_ctid
14     FROM generate_series((SELECT max / 25 FROM max_ctid), (SELECT max FROM max_ctid), (SELECT max / 25 FROM max_ctid)) d;
15 -- Время выполнения 9 мин
```

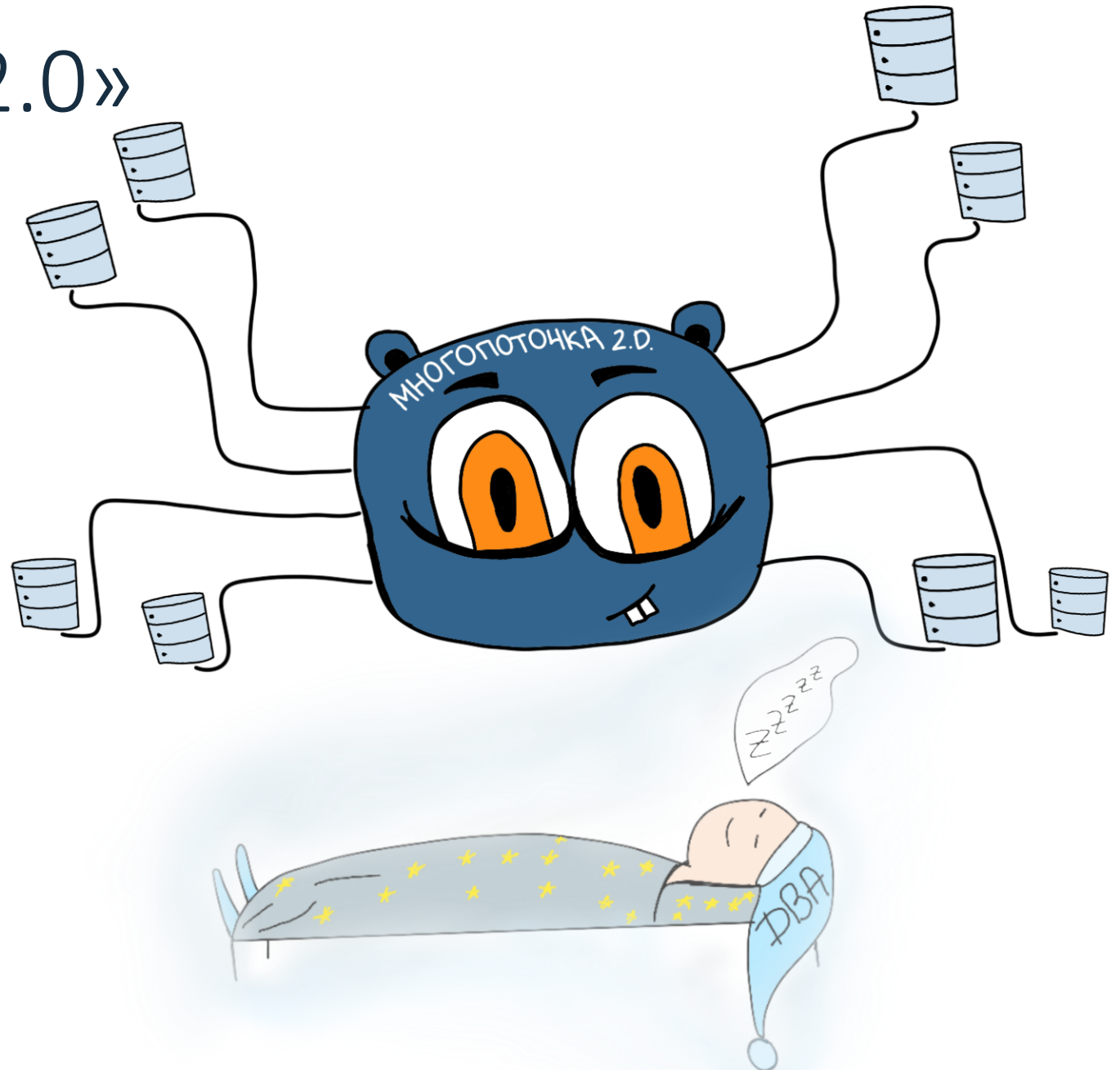
О «многопоточке 2.0»

- Ускорились на самом UPDATE в 7 раз!
- Ускорились в сумме в 3 раза

```
1  ./pgpar-linux-amd64 jdbc:postgresql://localhost:5432 soldatov '' pgconfdb PGCONF-2018-UPDATE
2  -- Время выполнения 7 минут
```

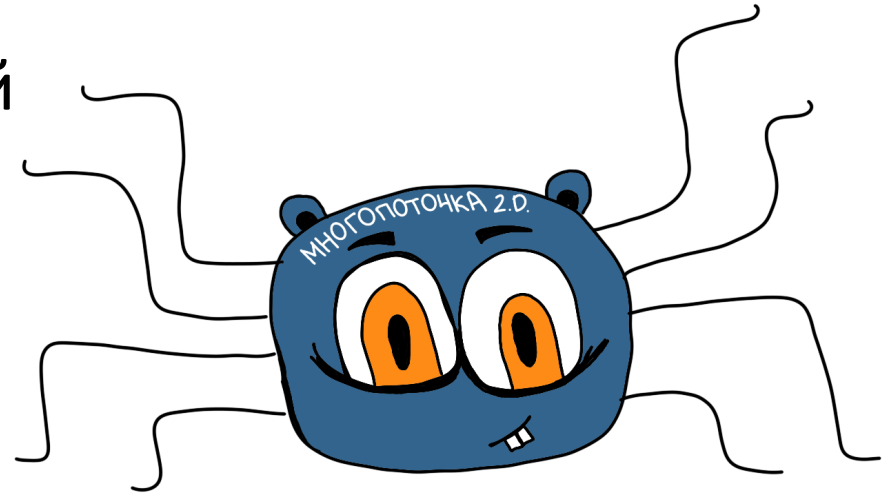
О «МНОГОПОТОЧКЕ 2.0»

- Автоматизация!
- Интеграция с Liquibase
- Еще много всего



О преимуществах

- Миграции быстрее
- Интеграция с Liquibase
- Сохранение примененных миграций
- История миграций для анализа



О преимуществах

- Progress bar!
- Логирование ошибок
- Web-интерфейс



О прогнозе времени

коп-во запущенных процессов

Threads count: 5

[Add worker] [Remove worker] [Show statements] [Refresh]

ДОБАВИТЬ ПРОЦЕСС
УДАЛИТЬ ПРОЦЕСС
ПОКАЗАТЬ текущие запросы
ОБНОВЛЕНИЕ



	Status	Count
New	НЕ НАЧАТЫЕ	9985
In progress	В РАБОТЕ	5
Done	ЗАВЕРШЕННЫЕ	10
Error	ЗАКОНЧИВШИЕСЯ ОШИБКОЙ	0

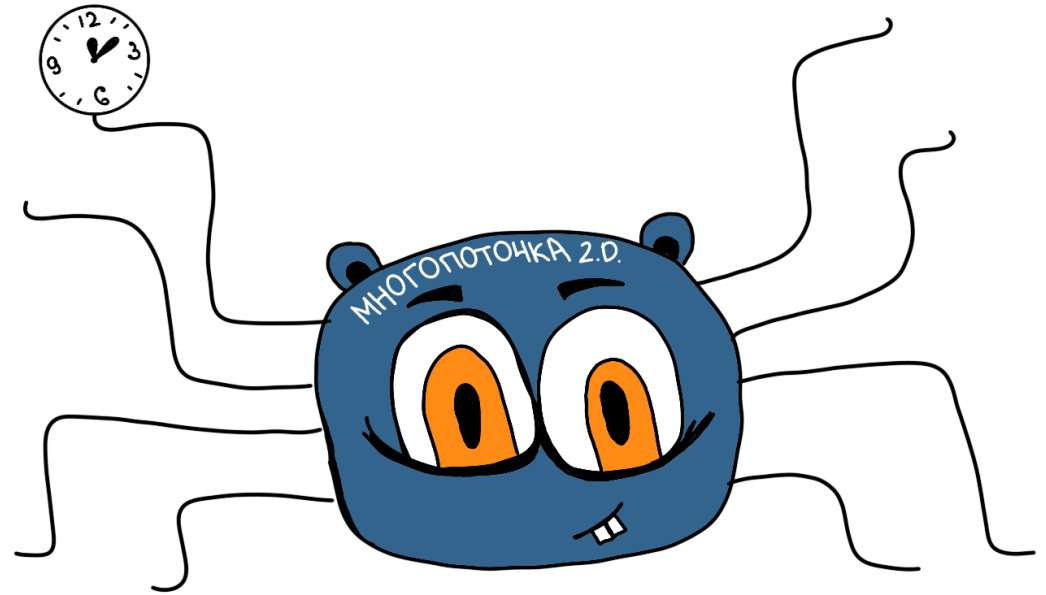
Avg statement time: 12s
Estimated time remaining: 6h56m50s

ОСТАВШЕЕСЯ ВРЕМЯ

СРЕДНЕЕ ВРЕМЯ ВЫПОЛНЕНИЯ одного процесса

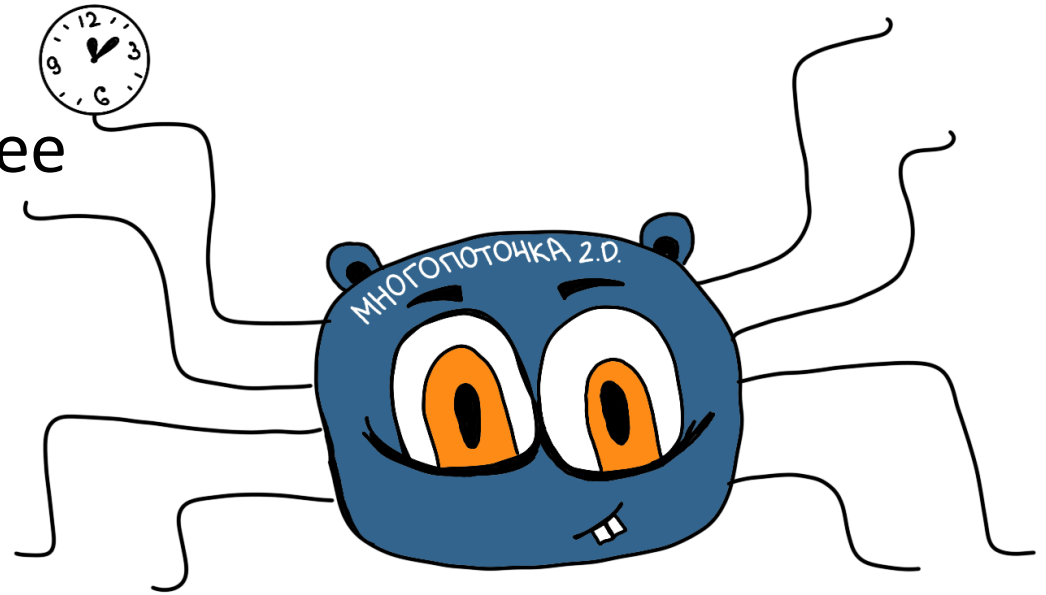
О проблемах прогноза времени

- Нельзя предсказать заранее
- Неточность в начале



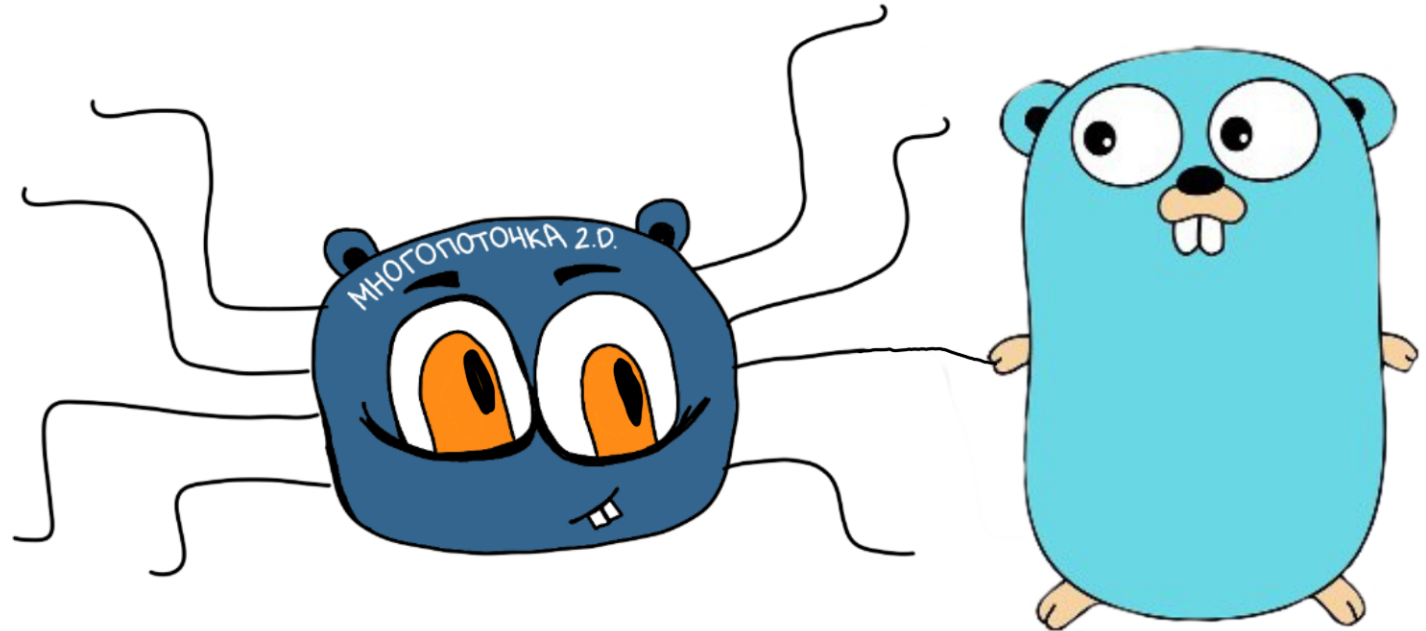
О планируемом прогнозе времени

- Анализ накопленных данных
- Предсказание времени заранее



ОГО

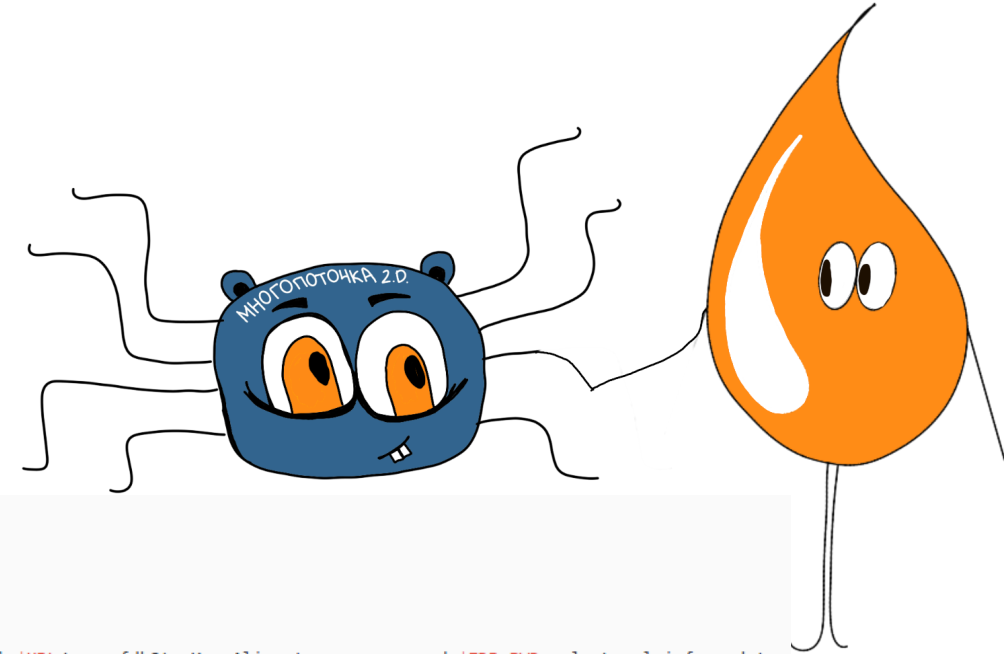
- Быстрее Python
- Приятнее C++
- Потому что модно :)



○ Liquibase

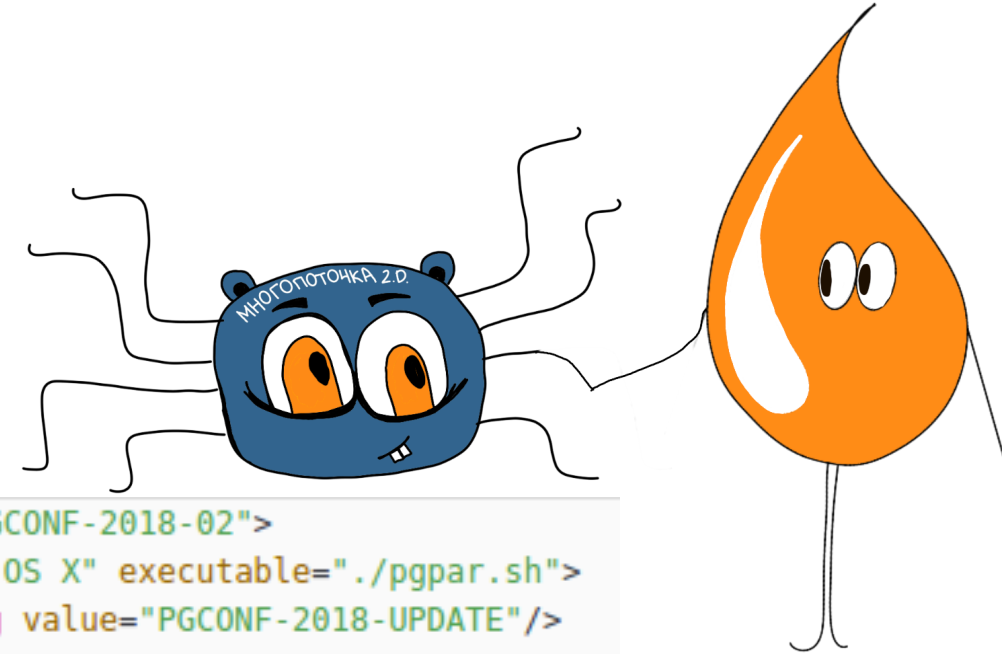
- Получаем логин, пароль и URL подключения из скриптов деплоя

```
1 export EDB_USR
2 export EDB_PWD
3 export URL
4
5 log "Running UPDATE using url: $URL"
6
7 ./liquibase --driver=org.postgresql.Driver --username=$EDB_USR --liquibaseSchemaName=public --changeLogFile=pgconf.xml --url=$URL/pgconfdb?tcpKeepAlive=true --password=$EDB_PWD --logLevel=info update
```



○ Liquibase

- Получаем имя БД и имя задания из changeSet



```
1 <changeSet author="soldatov" id="PGCONF-2018-02">
2   <executeCommand os="Linux, Mac OS X" executable="./pgpar.sh">
3     <arg value="pgconfdb"/><arg value="PGCONF-2018-UPDATE"/>
4   </executeCommand>
5 </changeSet>
```

О первом примере

```
1 <changeSet author="soldatov" id="PGCONF-2018-01">
2   <sql>
3     <![CDATA[
4       INSERT INTO public.pg_parallel_task (name, threads_count) values ('PGCONF-2018-UPDATE', 10);
5     ]]>
```

О первом примере

```
1 <changeSet author="soldatov" id="PGCONF-2018-01">
2   <sql>
3     <![CDATA[
4       INSERT INTO public.pg_parallel_task (name, threads_count) values ('PGCONF-2018-UPDATE', 10);
5
6       INSERT INTO public.pg_parallel_task_statements (task_name, sql_statement)
7       SELECT 'PGCONF-2018-UPDATE' task_name,
8         FORMAT('UPDATE pgbench_accounts
9           SET abalance = 42
10          WHERE filler IS NULL
11            AND aid >= '%s' AND aid <= '%s';',
12             MIN(d.aid), MAX(d.aid)) sql_statement
13       FROM (SELECT aid, ntile(10000) over (order by aid) part
14             FROM pgbench_accounts
15             WHERE filler IS NULL) d
16       GROUP BY d.part;
17     ]>
18   </sql>
19 </changeSet>
```

О первом примере

```
1 <changeSet author="soldatov" id="PGCONF-2018-01">
2   <sql>
3     <![CDATA[
4       INSERT INTO public.pg_parallel_task (name, threads_count) values ('PGCONF-2018-UPDATE', 10);
5
6       INSERT INTO public.pg_parallel_task_statements (task_name, sql_statement)
7       SELECT 'PGCONF-2018-UPDATE' task_name,
8         FORMAT('UPDATE pgbench_accounts
9           SET abalance = 42
10          WHERE filler IS NULL
11            AND aid >= '%s' AND aid <= '%s';',
12             MIN(d.aid), MAX(d.aid)) sql_statement
13       FROM (SELECT aid, ntile(10000) over (order by aid) part
14             FROM pgbench_accounts
15             WHERE filler IS NULL) d
16       GROUP BY d.part;
17     ]]>
18   </sql>
19 </changeSet>
20
21 <changeSet author="soldatov" id="PGCONF-2018-02">
22   <executeCommand os="Linux, Mac OS X" executable="./pgpar.sh">
23     <arg value="pgconfdb"/><arg value="PGCONF-2018-UPDATE"/>
24   </executeCommand>
25 </changeSet>
```

О втором примере

```
1 <changeSet author="soldatov" id="PGCONF-2018-01">
2   <sql>
3     <![CDATA[
4       INSERT INTO pg_parallel_task (name, threads_count)
5       VALUES ('PGCONF-2018-VACUUM', 2);
6
7       INSERT INTO pg_parallel_task_statements (task_name, sql_statement)
8       SELECT 'PGCONF-2018-VACUUM' task_name, 'VACUUM FULL ANALYZE pgbench_accounts;' sql_statement;
9
10      INSERT INTO pg_parallel_task_statements (task_name, sql_statement)
11      SELECT 'PGCONF-2018-VACUUM' task_name, 'VACUUM FULL ANALYZE pgbench_branches;' sql_statement;
12    ]]>
13  </sql>
14 </changeSet>
15
16 <changeSet author="soldatov" id="PGCONF-2018-02">
17   <executeCommand os="Linux, Mac OS X" executable="./pgpar.sh">
18     <arg value="pgconfdb"/><arg value="PGCONF-2018-VACUUM"/>
19   </executeCommand>
20 </changeSet>
```

О третьем примере

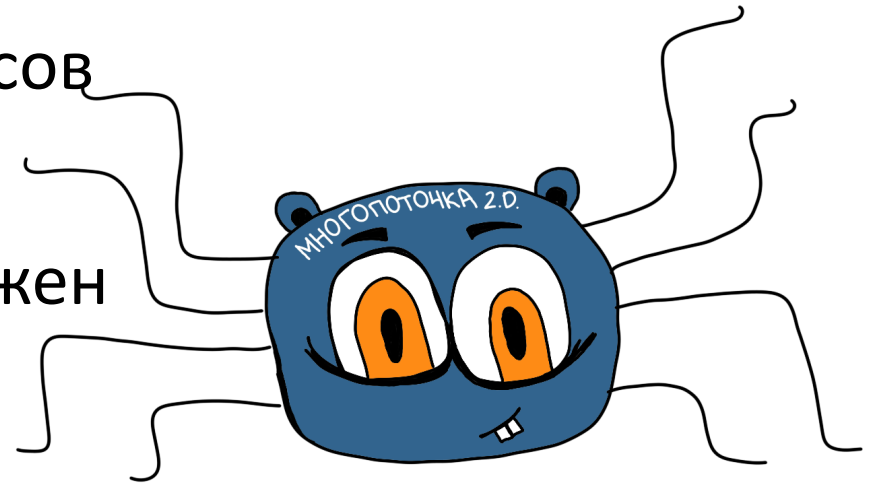
```
1  -- SQL part
2  ALTER TABLE pgbench_accounts ADD COLUMN account_number text;
3
4  INSERT INTO public.pg_parallel_task (name, threads_count) VALUES ('PGCONF-2018-ADD', 10);
5
6  INSERT INTO public.pg_parallel_task_statements (task_name, sql_statement)
7  SELECT 'PGCONF-2018-ADD' task_name,
8         FORMAT('UPDATE pgbench_accounts
9                SET account_number = aid::text || filler
10               WHERE aid >= '%s' AND aid <= '%s';',
11               MIN(d.aid), MAX(d.aid)) sql_statement
12  FROM (SELECT aid,
13         ntile(50000) over (order by device_version_guid) part
14        FROM pgbench_accounts) d
15  GROUP BY d.part;
```

О третьем примере

```
1 # shell part
2 ./pgpar-linux-amd64 jdbc:postgresql://localhost:5432 soldatov secret_password pgconfdb PGCONF-2018-ADD
```

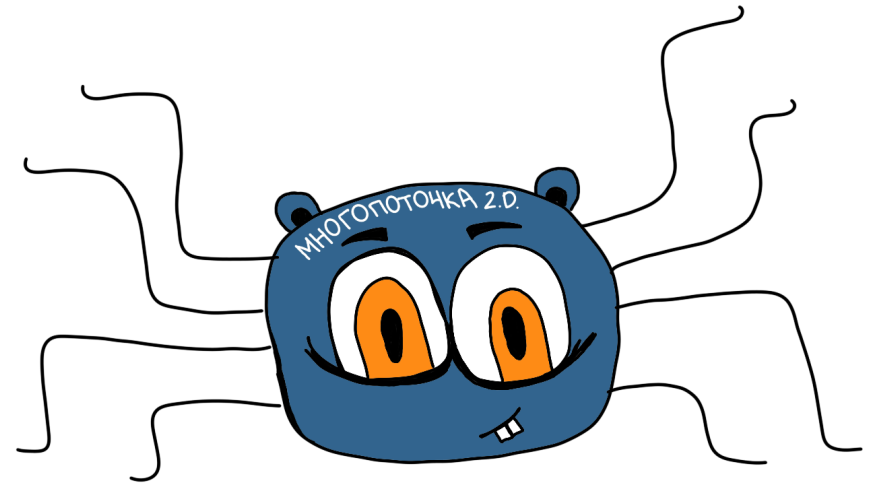
Об итогах

- Просто реализовать
- Много вариантов применения
- Быстрее обычных миграций
- Есть progress bar и web-интерфейс
- Динамическое изменение числа процессов
- Проще проводить онлайн-миграции
- Если ключевым полем выбран не int, нужен предварительный full scan



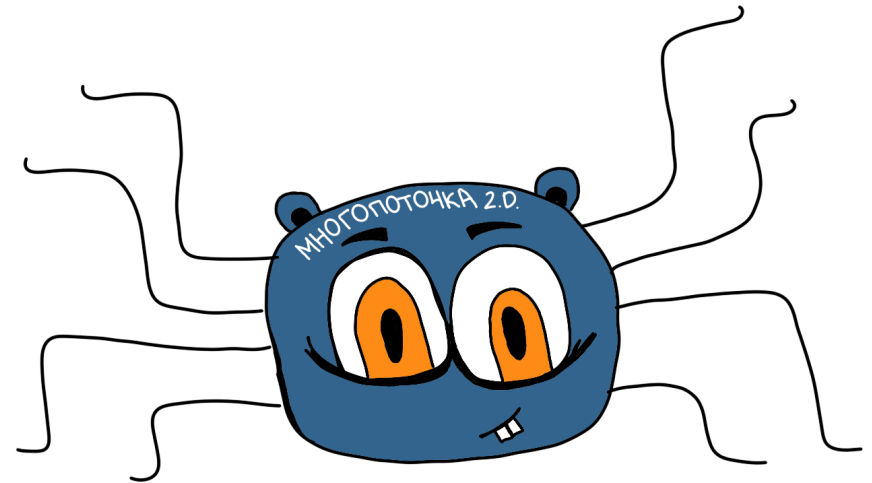
О планах на будущее

- Улучшить предсказание времени
- Сделать вызов из PostgreSQL
- Украсить web-интерфейс 😊



О ссылках

- ГИС ЖКХ на хабре
<https://habrahabr.ru/company/lanit/blog/321476/>
- Liquibase на TechGuruDay
<https://www.youtube.com/watch?v=JNstiXQoppE>
- Приходите к нам работать 😊
<https://job.lanit.ru>



Очень большое спасибо
за внимание!

Email - ASoldatov@lanit.ru

Telegram - [@AnatolySK](https://t.me/AnatolySK)

