

Управление сценариями миграции большого объёма данных из Oracle в PostgreSQL

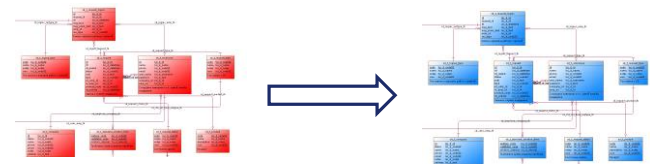
Александр Любушкин
Андрей Чибук

Миграция структур данных

Миграция программ на языках СУБД

Миграция данных

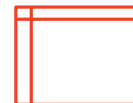
Миграция клиентских программ и интерфейсов



PL/SQL



PL/pgSQL



Наши решения

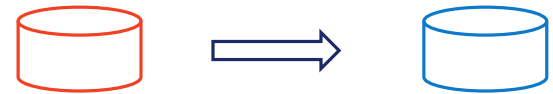


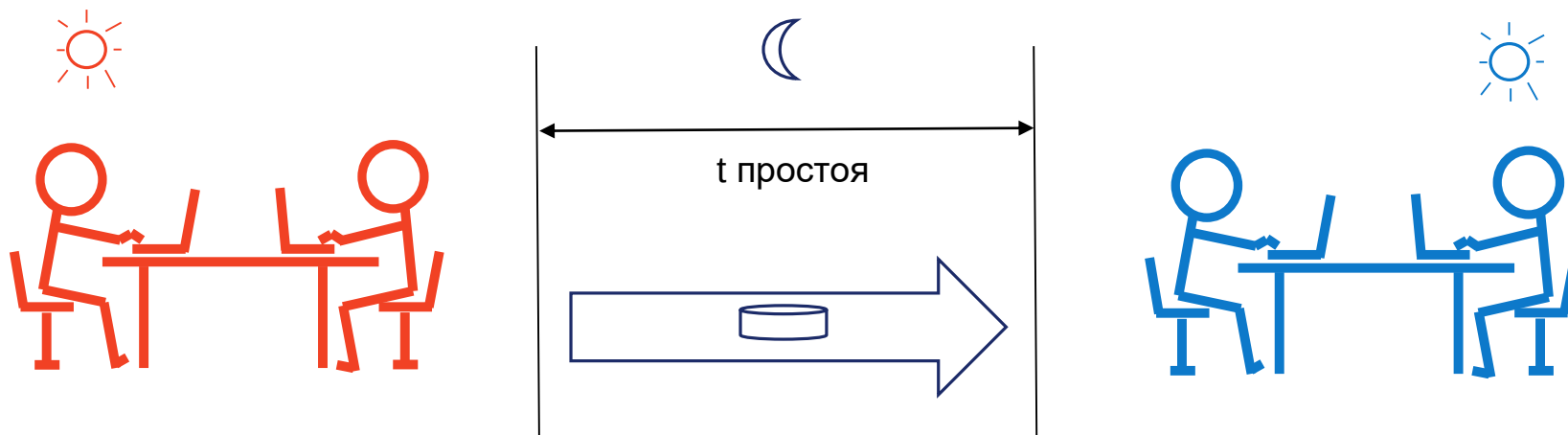
Миграция структур данных

Миграция программ на языках СУБД

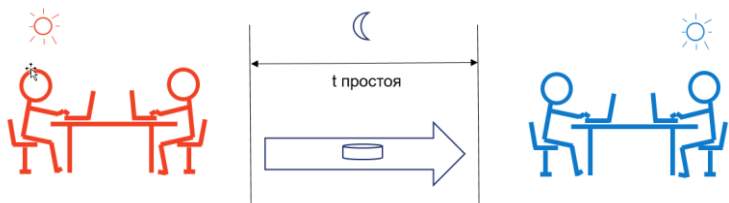
Миграция данных

Миграция клиентских программ и интерфейсов





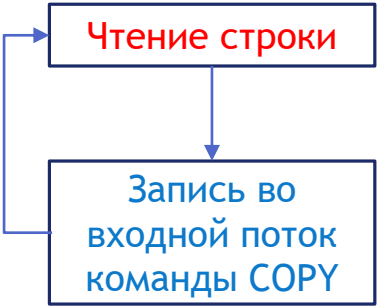
- Для многих организаций время простоя прикладной системы является критически важным
- Возможные финансовые и репутационные потери заставляют их вкладывать большие ресурсы в разработку и тестирование организационных и технических решений операции перехода
- Чем больше БД, тем выше требования к минимизации времени простоя



- Скорость в одном потоке
- Обеспечение многопоточности
- Обеспечение равномерной и полной загрузки вычислительных ресурсов
- Пропускная способность сети
- Надёжность и обработка нештатных ситуаций
- Организационно-технические решения – можно ли вообще избежать простоя?

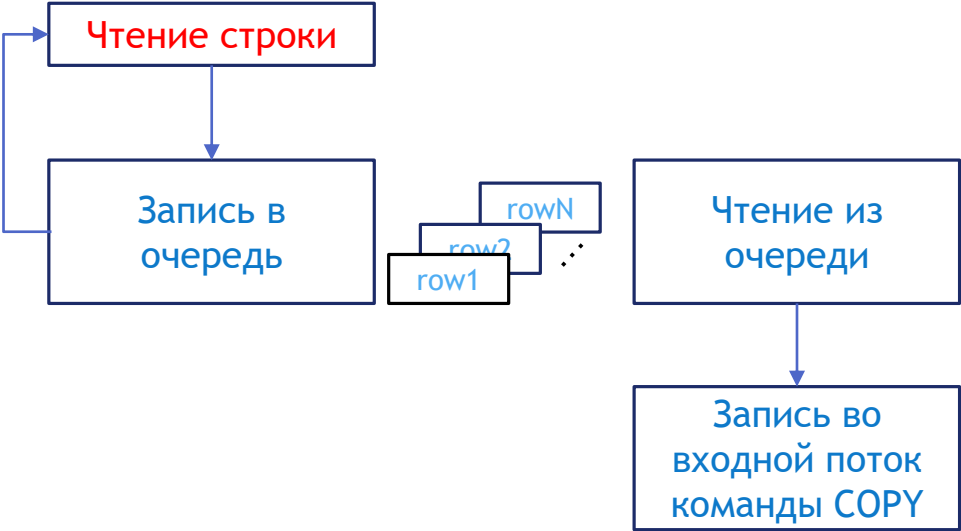
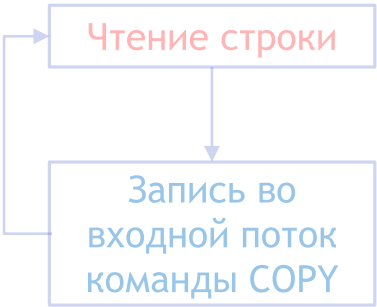
Скорость в одном потоке

```
select * from my_table  
copy my_table from stdin
```



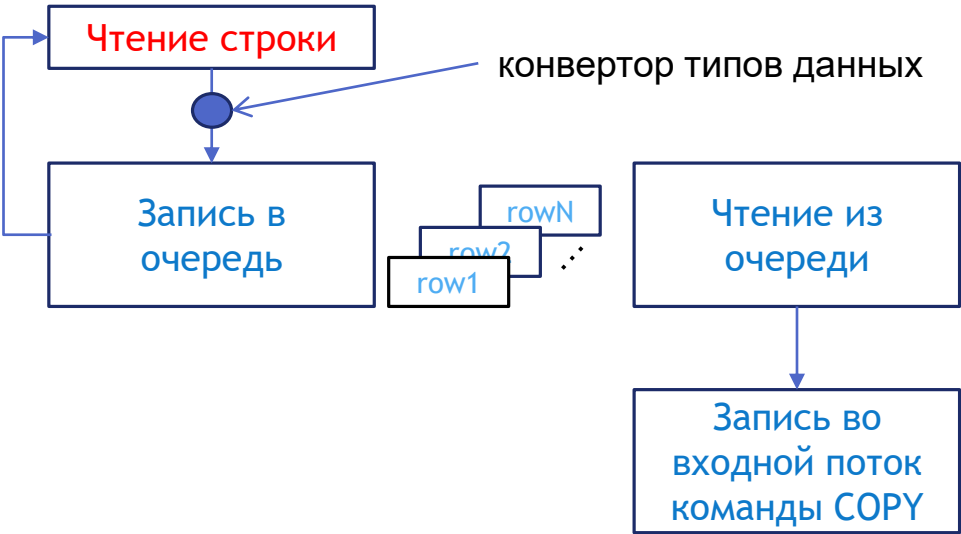
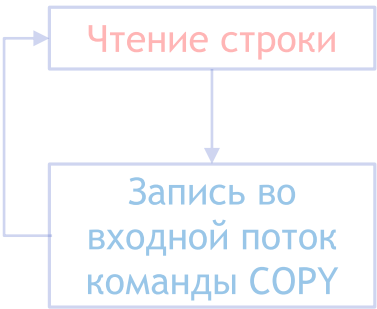
Скорость в одном потоке

```
select * from my_table  
copy my_table from stdin
```

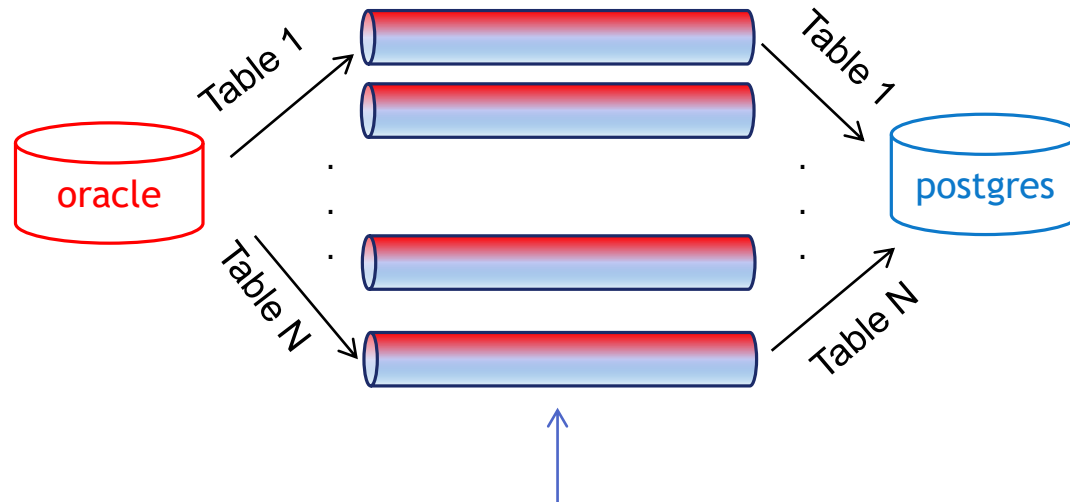


Двоичный формат даёт ускорение в 4-6 раз

```
select * from my_table  
copy my_table from stdin with format binary
```



- Каждая таблица (запрос) обрабатывается отдельным потоком
- Количество параллельных потоков - параметр



Процессы обработки таблиц

Файл с настройками – JSON структура

```
{ "sessions": 24,  
  "truncate": true,  
  "oradb": "localhost:1521:oradb", "orausername": "book", "orapassword": "book",  
  "pgghost": "localhost", "pgport": "5432", "pgdb": "postgres", "pgusername": "book", "pgpassword": "book",  
  "unload": "yes", "load": "direct", "index": "no",  
  "sessionwriters": 8,  
  "prefetchrows": 2048, "lobprefetchsize": 32768, "lobstreambuffer": 32768,
```

Общие
настройки

```
"work": [  
  { "query": "select * from book.o2p_t_publisher", "target": "book.o2p_t_publisher" },  
  { "query": "select * from book.o2p_t_genre", "target": "book.o2p_t_genre" },  
  { "query": "select * from book.o2p_t_author", "target": "book.o2p_t_author" },  
  { "query": "select * from book.o2p_t_book", "target": "book.o2p_t_book" },  
  { "query": "select * from book.o2p_t_book_view_log where id<1000000", "target": "book.o2p_t_book_view_log"},  
  { "query": "select * from book.o2p_t_book_view_log where id>=1000000", "target": "book.o2p_t_book_view_log"},  
]  
}
```

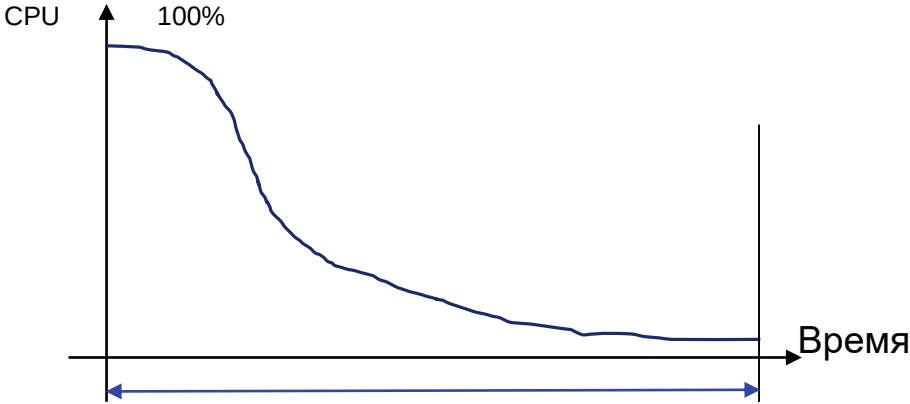
Массив объектов с
настройками миграции
таблиц

Пример

БД - 60 таблиц

Количество таблиц	Тб
1	5
9	3
50	2
Всего 60	10 Тб

CPU - 128 ядер
Выполняем перенос в 60 потоков



Как повысить равномерность?

- Делить секционированные таблицы на несколько потоков по секциям

```
select * from tab partition(p1)
```

...

```
select * from tab partition(pN)
```

- По диапазону значений ключа

```
select * from tab where id between  
1 and 100000000
```

```
select * from tab where id between  
100000001 and 200000000
```

...

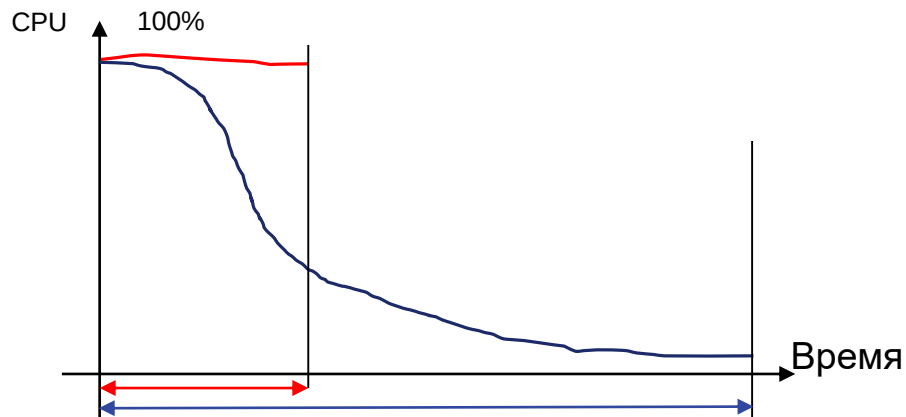
- По модулю значений ключа

```
select * from tab where mod(id,8)=0
```

```
select * from tab where mod(id,8)=1
```

...

```
select * from tab where mod(id,8)=7
```



tab

Id	field_lob
1	Содержимое файла представлено в формате JSON (JavaScript Object Notation). Указывается значением, как основной параметр, необходимый для работы скрипта, так и включен параметром загрузки, определенным для каждой таблицы и необходимости дане для конкретных столбцов, предоставляемых в экспериментальной миграции данных.
2	Содержимое файла представлено в формате JSON (JavaScript Object Notation). Указывается значением, как основной параметр, необходимый для работы скрипта, так и включен параметром загрузки, определенным для каждой таблицы и необходимости дане для конкретных столбцов, предоставляемых в экспериментальной миграции данных.
...	

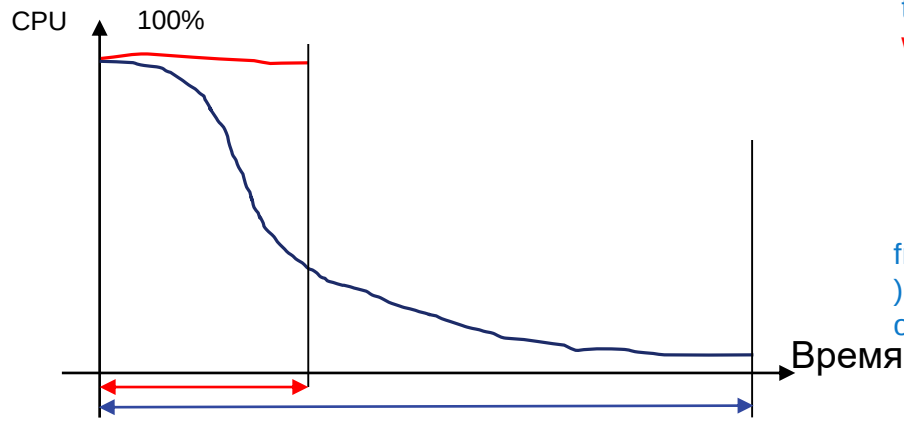
1Mб

500Mб

Как повысить равномерность?

- Делить таблицы с CLOB/BLOB на несколько равных частей

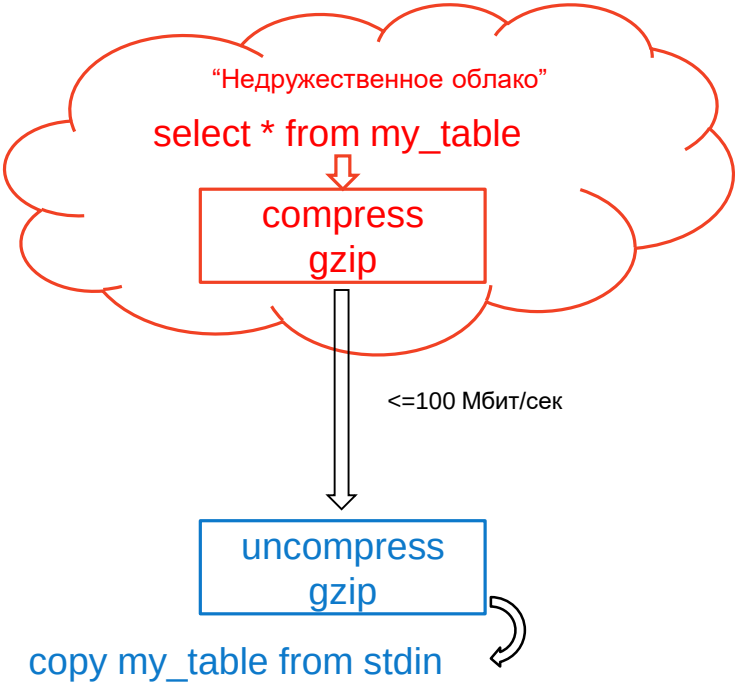
```
select min(id),max(id),count(*) cnt,sum(len), buc from(  
with length_sum  
as (SELECT SUM(length(tab.field_lob)) sm FROM tab )  
SELECT  
length(t1.field_lob) len,  
t1.id,  
width_bucket(  
sum(length(t1.field_lob))  
over (order by t1.id),  
0,  
L.sm+1,  
8) buc  
from tab t1, length_sum L  
) group by buc  
order by bak
```



MIN(ID)	MAX(ID)	CNT	SUM(LEN)	BUC
16767	17220	105	36 844 855	1
17222	17535	99	37 135 292	2
17537	17879	126	36 920 467	3
17887	18026	19	36 088 026	4
18027	18038	8	33 128 098	5
18041	19246	19	40 970 350	6
19249	19259	7	32 665 352	7
19260	20070	28	42 481 333	8

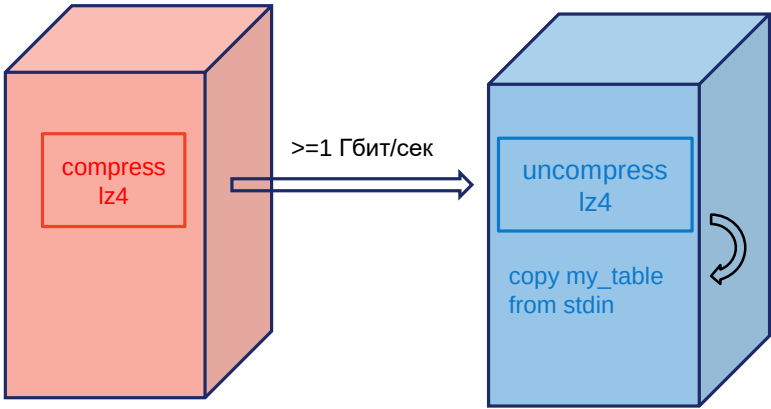
Передача сжатых данных

Пример 1



Передача сжатых данных

Пример 2



Мощные серверы, способные
перегрузить даже быстрые сети

БД крупных систем уровня организации могут содержать несколько тысяч таблиц и терабайты данных.

Нельзя надеяться, что операция миграции пройдёт “гладко”.

Что-нибудь обязательно пойдёт “не так”.

Нештатные ситуации при выполнении операции миграции

- **Исходная система оказалась не полностью изолирована от подключений**

Во время операции миграции начался ETL-процесс, связанный с интеграцией системы с другой системой. Одна из таблиц оказалась под нагрузкой массовых операций. В результате – ORA-1555 snapshot too old

- **Ошибки в исходной системе при чтении данных**

Из-за несимметричности схем данные для одной из таблиц в Postgres извлекались функцией Oracle. Всё работало во время тестов. Но появились новые данные, и в момент операции миграции функция обрабатывала их с ошибкой.

- **Внезапное появление в CLOB/BLOB поля > 1Гб**

Всё работало во время тестов. Но перед операцией миграции появились новые данные >1Гб

- ...

Что делать, если таблица или её часть не перенеслась?

```
"work": [  
{
```

```
  "load": "direct",  
  "query": "select * from t_table where id < 1000000",  "target": "t_table"
```

```
  "index": "yes"  
  "indexstatements": "alter table add constraint t_table_pk primary key (id) using index tablespace tbs;"
```

```
},  
{
```

```
  "load": "direct",  
  "query": "select * from t_table where id >= 1000000",  "target": "t_table"
```

```
  "index": "yes"  
  "indexstatements": "alter table add constraint t_table_pk primary key (id) using index tablespace tbs;"
```

```
},  
]
```

1

2

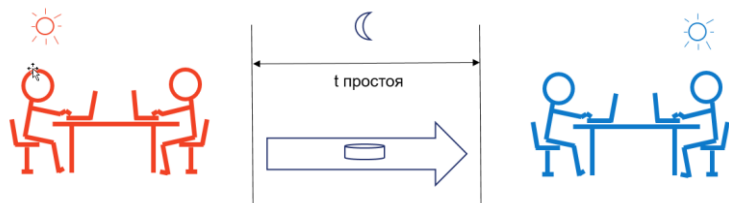
Что делать, если таблица или её часть не перенеслась?
“Умный” сценарий миграции частей таблиц!

```
"work": [  
  {  
    "workid": "t_table_part_1",  
    "sqlbefore": "pg: my_function_before(?,?)",  
    "load": "pg:select my_function_need_load(?,?)", -- вернёт 'no', если 1-я часть была загружена ранее  
    "query": "select * from t_table where id < 1000000", "target": "t_table"  
    "sqlafterdata": "pg: my_function_after_data(?,?,?)"  
    "index": "pg:select my_function_need_index(?,?)"  
    "indexstatements": "alter table add constraint t_table_pk primary key (id) using index tablespace tbs;"  
    "sqlafter": "pg: my_function_after(?,?,?)"  
  },  
  {  
    "workid": "t_table_part_2",  
    "sqlbefore": "pg: my_function_before(?,?)",  
    "load": "pg:select my_function_need_load(?,?)",  
    "query": "select * from t_table where id >= 1000000", "target": "t_table"  
    "sqlafterdata": "pg: my_function_after_data(?,?,?)"  
    "index": "pg:select my_function_need_index(?,?)"  
    "indexstatements": "alter table add constraint t_table_pk primary key (id) using index tablespace tbs;"  
    "sqlafter": "pg: my_function_after(?,?,?)"  
  }  
]
```

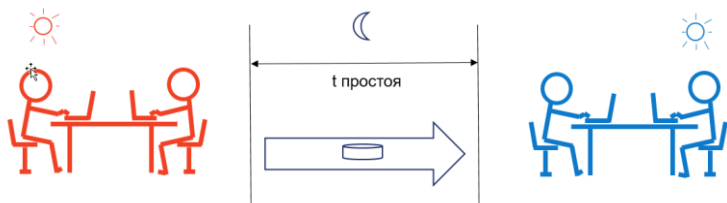
“error text” или “

1

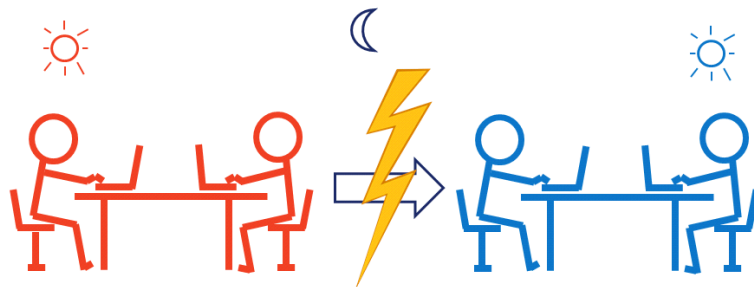
2



- Устаревшие неизменяемые данные переносить отдельно – до операции перехода
- Деление на подсистемы для перехода в несколько этапов
- Деление системы на региональные сегменты
- ...



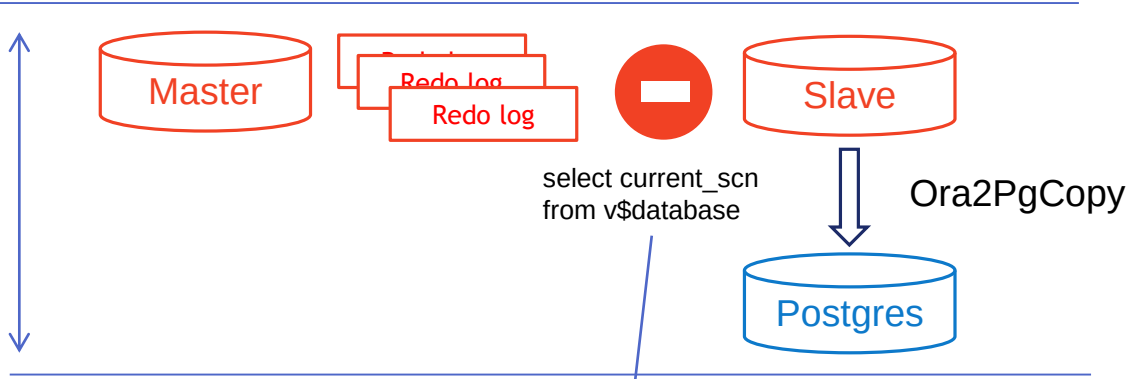
- Устаревшие неизменяемые данные переносить отдельно – до операции перехода
- Деление на подсистемы для перехода в несколько этапов
- Деление системы на региональные сегменты
- “Мгновенный” переход без остановки исходной системы для операции миграции



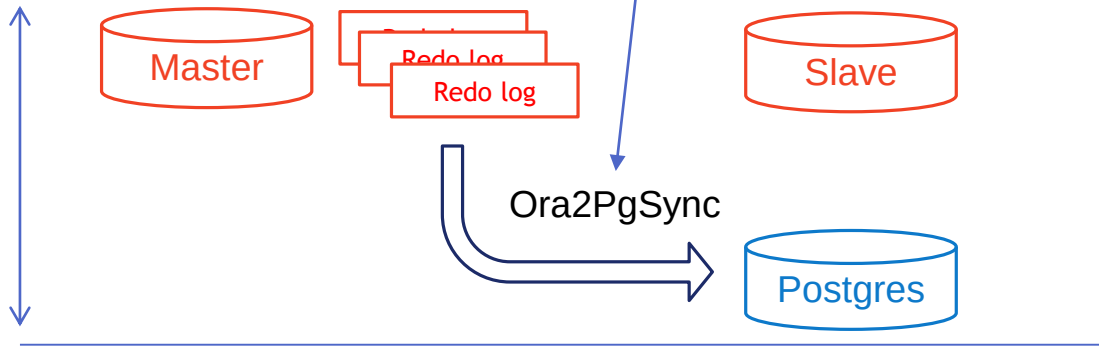
Исходное состояние до операции миграции



Время миграции основной массы данных



Время синхронизации баз данных



Файл настроек Ora2PgSync – JSON

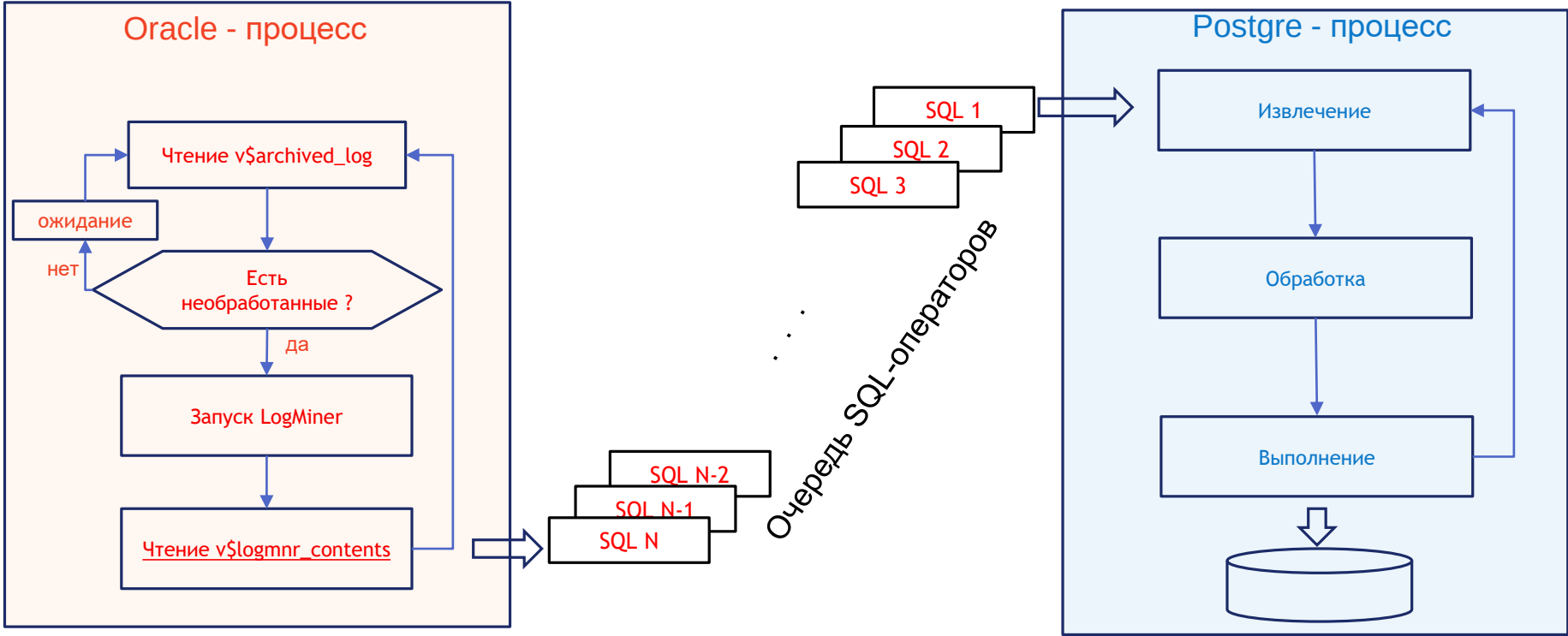
```
{ "oradb": "192.168.1.1:1521/oradb",  
  "orausername": "logmnr",  
  "orapassword": "logmnr",  
  "pgdb": "192.168.1.2:5432/postgres",  
  "pgusername": "ora2pg_user",  
  "pgpassword": "ora2pg_passwd",  
  "loglevel": "info",  
  "work": [  
    { "schema": "book", "table": "o2p_t_author"},  
    { "schema": "book", "table": "o2p_t_book"},  
    { "schema": "book", "table": "o2p_t_genre"},  
    { "schema": "book", "table": "o2p_t_publisher"},  
    { "schema": "book", "table": "o2p_t_abcd",  
      "target_schema": "book", "target_table": "o2_p_t_abcde",  
      "fields": [ { "source": "decimal_fld", "target": "decimal_fld"},  
                  { "source": "raw_fld", "target": "raw_fld"},  
                  { "source": "flag_v", "target": "flag_b",  
                    "value_template": "case when $value = 'Y' then true else false end" } ]  
    }  
  ]  
}
```

Исходные данные – v\$logmnr_contents

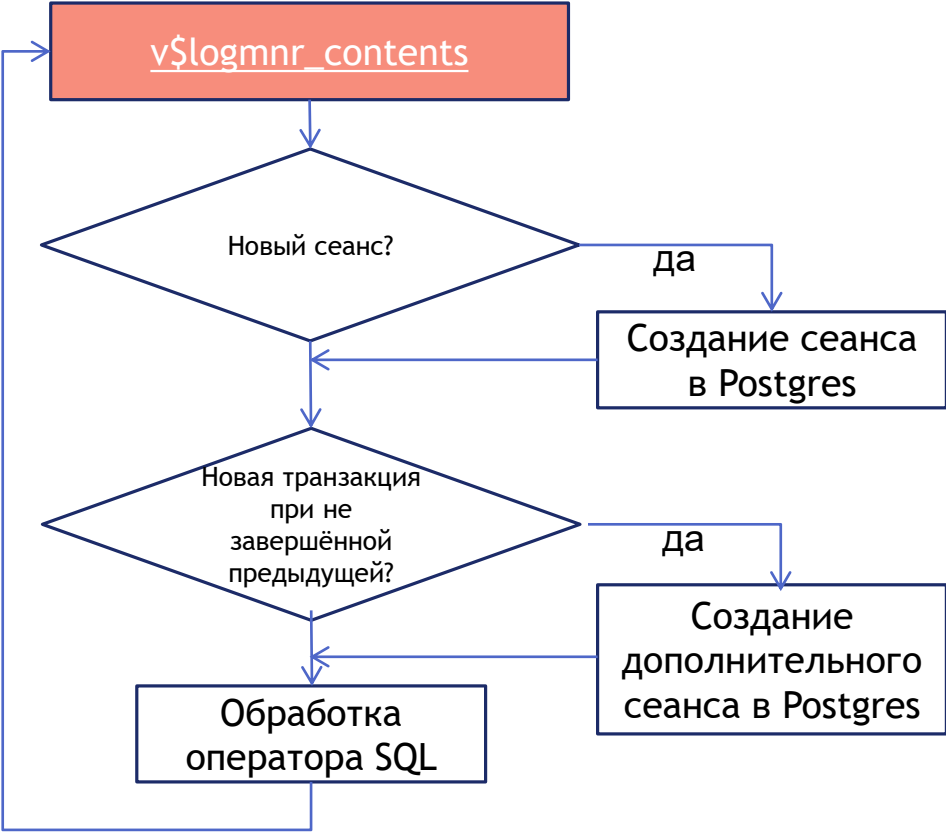
		SCN	START_SCN	COMMIT_SCN	SQL_REDO	OPERATION
транзакции	1	150662122			set transaction read write;	START
		150662122			insert into "BOOK"."O2P_T_PUBLISHER"("CODE","NAME","ADDRESS","PHONE_NUM","WEB") values ('00006','Тон','г.р.Тверь','223-322-223-322','https://book.top.ru');	INSERT
		150662124			insert into "BOOK"."O2P_T_PUBLISHER"("CODE","NAME","ADDRESS","PHONE_NUM","WEB") values ('00007','Тон-2','г.р.Астрахань','423-322-223-324','https://book.top2.ru');	INSERT
		150662129	150662122	150662129	commit;	COMMIT
	2	150662133			set transaction read write;	START
		150662133			delete from "BOOK"."O2P_T_PUBLISHER" where "CODE" = '00006' and "NAME" = 'Тон' and "ADDRESS" = 'г.р.Тверь' and "PHONE_NUM" = '223-322-223-322' and "WEB" = 'https://book.top.ru';	DELETE
		150662134	150662133	150662134	rollback;	ROLLBACK
	3	150662178			set transaction read write;	START
		150662178			update "BOOK"."O2P_T_BOOK" set "YEAR_OF" = '2002' where "CODE" = '00001' and "YEAR_OF" = '2001';	UPDATE
		150662184	150662178	150662184	commit;	COMMIT

Принцип работы Ora2PgSync

JVM

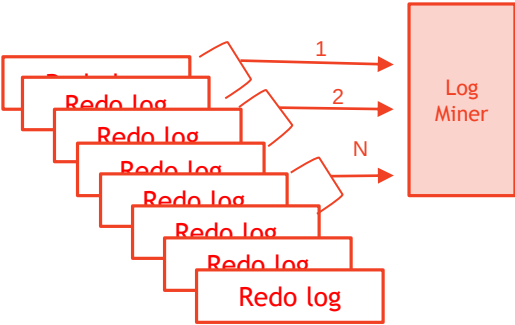


Обработка автономных транзакций



Почему нельзя использовать опцию COMMITTED_DATA_ONLY

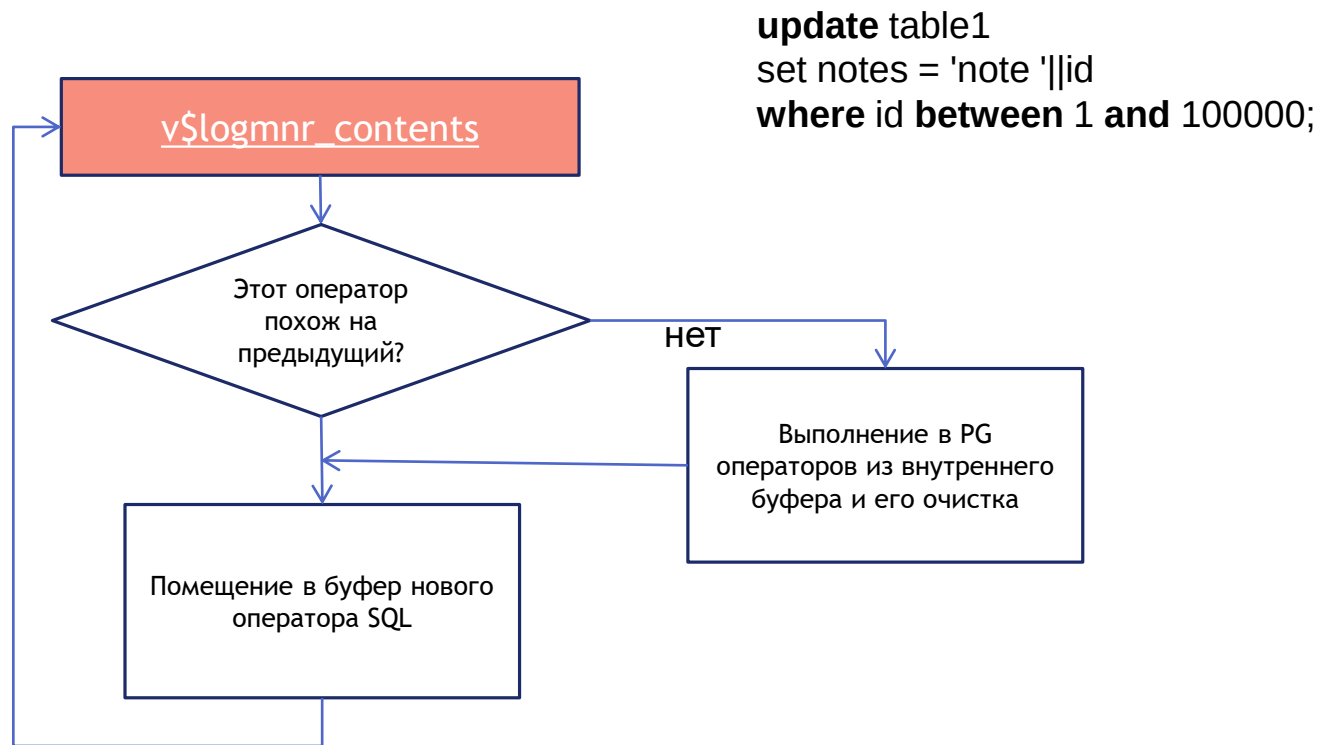
- 1 Для повышения быстродействия необходимо ограничить количество журналов, одновременно обрабатываемых в LogMiner
- 2 COMMITTED_DATA_ONLY не будет показывать операции, для которых нет commit в обрабатываемом наборе журналов
- 3 Обработывая очередной набор журналов, мы увидим null в start_scn



SCN	START_SCN	COMMIT_SCN	SQL_REDO	OPERATION
150662122			insert into "BOOK"."O2P_T_PUBLISHER"("CODE", "NAME", "ADDRESS", "PHONE_NUM", "WEB") values ('00006', 'Топ', 'гор. Тверь', '223-322-223-322', 'https://book.top.ru');	INSERT
150662124			insert into "BOOK"."O2P_T_PUBLISHER"("CODE", "NAME", "ADDRESS", "PHONE_NUM", "WEB") values ('00007', 'Топ-2', 'гор. Астрахань', '423-322-223-324', 'https://book.top2.ru');	INSERT
150662129	NULL	150662129	commit;	COMMIT

Это означает, что начало транзакции надо искать в одном из предыдущих файлов – усложнение алгоритма и снижение производительности.

Обработка массовых операций



Направления развития Ora2PgSync

- | | | |
|---|--------------------|--|
| 1 | Производительность | - “copy” для массовых insert |
| 2 | Функциональность | - Пользовательские функции обработки данных |
| 3 | Надёжность | - Продолжение синхронизации баз после аварии |
| 4 | Мониторинг | - Отображение скорости синхронизации
- Прогноз завершения |



Инкрементальная миграция данных из СУБД Oracle в СУБД PostgreSQL
(Ora2PgSync)

Руководство пользователя

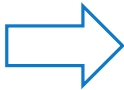
Всего страниц 16

Оглавление

1. Описание технологии и инструментов, используемых для выполнения миграции	2
1.1. Основные принципы	2
1.2. Oracle LogMiner	2
1.2.1. Описание и основы использования	2
1.2.2. Режимы работы LogMiner	2
1.2.3. Подготовка для запуска Oracle LogMiner	2
2. Подготовка к миграции	2
2.1. Настройка логирования в Oracle	6
2.1.1. Режим архивирования логов	6
2.1.2. Включение режима архивирования логов	6
2.1.3. Управление разложением архивных файлов	7
2.2. Подготовка к работе с Oracle LogMiner	8
2.2.1. Создание пользователя в Oracle и назначение ему привилегий для работы с Oracle LogMiner	8
3. Утилита ora2pgsync	8
3.1. Общее описание принципов работы	8
3.2. Структура каталогов, описание файлов	9
3.3. Конфигурационный файл. Описание и настройка	9
3.4. Файл SCN	11
3.5. Технология и алгоритм миграции. От и до	12
3.5.1. Извлечение сведений о DML-командах, выполненных в Oracle	12
3.5.2. Выполнение DML-команд в Postgres	13
3.6. Запуск	15
3.6.1. Параметры, указываемые при запуске	15
3.6.2. Рекомендации	15
4. Ограничения	16

Ora2PgSync

Анализ журналов
через Log Miner

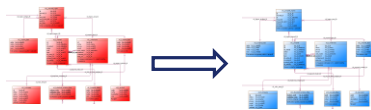


Миграция структур данных

Миграция программ на языках
СУБД

Миграция данных

Миграция клиентских программ
и интерфейсов



PL/SQL → PL/pgSQL



edu.fors.ru



Учебный центр ФОРС –
курсы по различным направлениям

Курс
“Практики миграции БД из
Oracle в PostgreSQL”:

- Лекции



- Лабораторные работы



Вопросы ???

E-mail:

forstelecom@fors.ru

lui@fors.ru

Сайт:

<http://lui.fors.ru>

<http://forstelecom.ru>

Скачать:<http://lui.forstelecom.ru/ORAPGCOPY.zip>

<http://lui.forstelecom.ru/ORAPGSYNC.zip>

Телефон:

+7 495 913 7575

доб. 111, 112, 139

Мобильный:

+7 903 505-9567

+7 903 505-9564

