



PostgreSQL для администраторов баз данных и разработчиков



Проверить, идет ли запись

Меня хорошо видно & слышно?



Тема вебинара

Репликация в PostgreSQL



Михеев Ростислав

ASKIT.RU, преподаватель

Telegram: @rostislavmikheev



Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Telegram
Postgre-DBA-2025-01



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Маршрут вебинара



Преимущества репликации

Виды репликации

Физическая репликация

Логическая репликация

Гетерогенная репликация

Рефлексия

Цели вебинара

К концу занятия вы сможете

1. Оценивать ситуацию для использования репликации
2. Знать возможности и особенности каждого способа
3. Выбирать наиболее оптимальный способ в зависимости от задачи

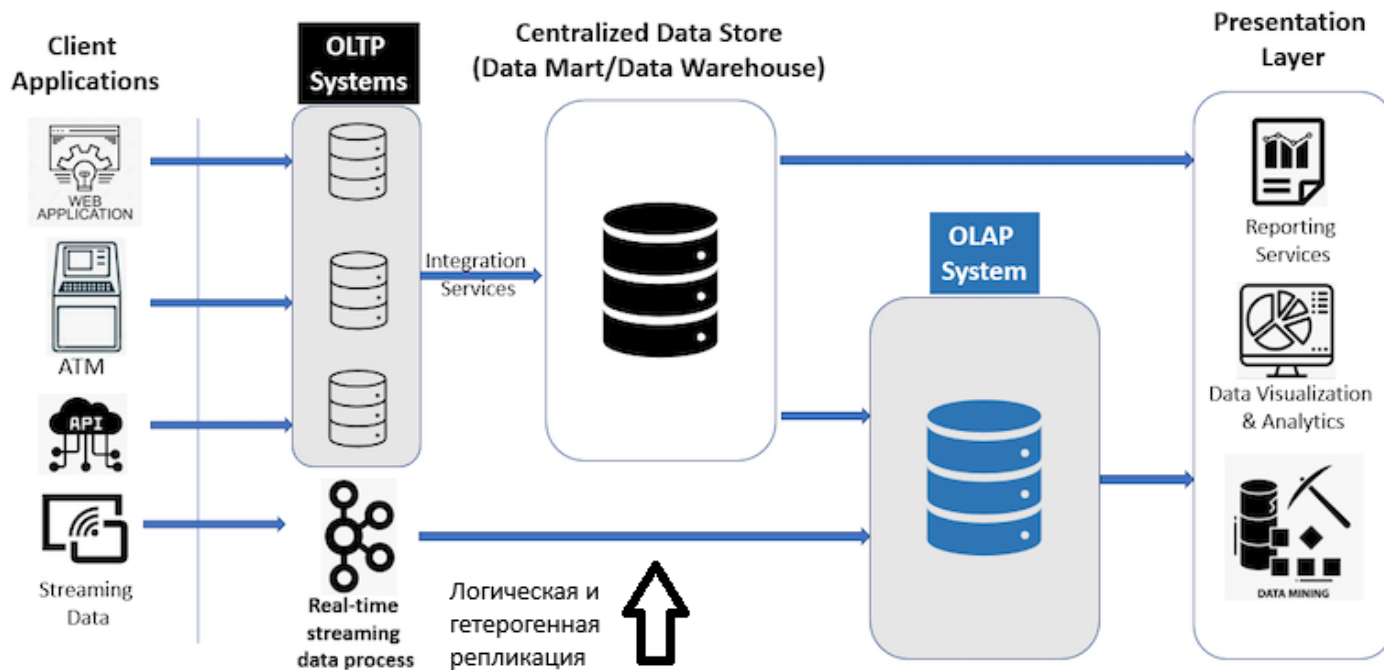
Репликация в PostgreSQL

Репликация – процесс синхронизации данных

Виды репликации

- Физическая
- Логическая
- Гетерогенная

Схема работы с данными



Note: In some cases, data architecture designs consider OLAP engine is part of Data Warehouse, in some cases one of Data Warehouse layer serves the need of OLAP engine.

Физическая репликация

- Основное назначение – отказоустойчивый кластер
- Вспомогательное назначение – перенос read-only нагрузки на другой сервер
- Используется передача WAL
- Идентичная копия primary и secondary серверов
- Из коробки нет автоматического файловера
- Подробный рассказ – в модуле про Patroni

Логическая репликация

- Основное назначение – перенос нагрузки на чтение на другие серверы
- Избирательная, фильтр в виде публикации
- Свой набор столбцов, записей, своя система индексов. Возможно использовать набор данных за большой период времени
- Большая проблема: команды DDL не реплицируются

Настройка логической репликации

- Настройка postgresql.conf: wal_level = logical
- Настройка pg_hba.conf: дать возможность подключения с удаленных IP-адресов для репликации
- Создание пользователя с правом replication и выдача прав на таблицы
- Настройка таблиц
- Создание публикации
- Создание подписки

Гетерогенная репликация

- Передача данных в другие системы (СУБД, микросервисы, очереди сообщений)
- Другое название – событийная интеграция
- Триггеры/внешние процедурные языки
- Notify и встроенная очередь
- Подключение к системе логической репликации программно
- Debezium Connector/Kafka

Внешние процедурные языки

- Их много! https://wiki.postgresql.org/wiki/PL_Matrix
- Самый распространенный - plpython
- Можно использовать с триггерами
- Самый простой способ гетерогенной репликации, но нет очередей и асинхронности
- Демонстрация

Notify/pg_notify

- Очень простой способ

```
LISTEN virtual;
```

```
NOTIFY virtual;
```

- Очень быстрый способ, команда выполняется за доли миллисекунд, транзакции не тормозят
- Встроенная очередь на каждый канал уведомлений – размер 8 Гбайт
- Максимальный размер сообщения (payload) – 8 Кбайт
- Нет отказоустойчивости, при отсутствии получателя сообщения теряются

Debezium/Kafka

- Самый функциональный, но и самый сложный способ
- Отказоустойчивость и скорость работы
- Нужен wal2json – данные об изменениях в WAL передаются в Kafka в формате json
- Основной способ для передачи изменений в Clickhouse, Apache Impala и другие системы Data Warehouse
- Демонстрация

Практика

Практика

1. Два виртуальных компьютера – настройка логической репликации

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Рефлексия

Рефлексия



С какими впечатлениями уходите с вебинара?

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Михеев Ростислав