

# Классификация существующих методов параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения

Группа: ИУ7-75Б

Студент: Платонова Ольга Сергеевна

Руководитель: Филиппов Михаил Владимирович,  
к.т.н., доцент кафедры ИУ-7

Консультант: Гаврилова Юлия Михайловна

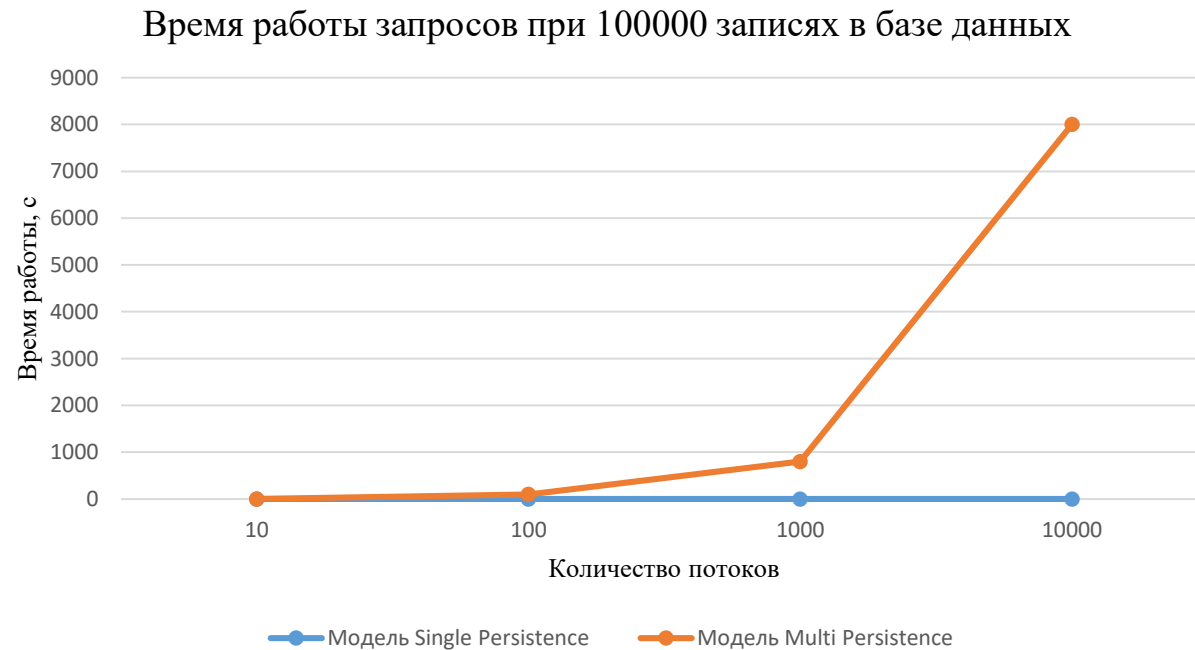
# Цели и задачи работы

**Цель данной работы** - классификация существующих методов параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения.

## **Задачи:**

- Анализ существующих СУБД и аргументация выбора PostgreSQL;
- Изучение основных положений архитектуры PostgreSQL и реализации соединения;
- Рассмотрение существующих методов выполнения запросов к СУБД в пределах одного соединения и их сравнительный анализ.

# Формализация задачи



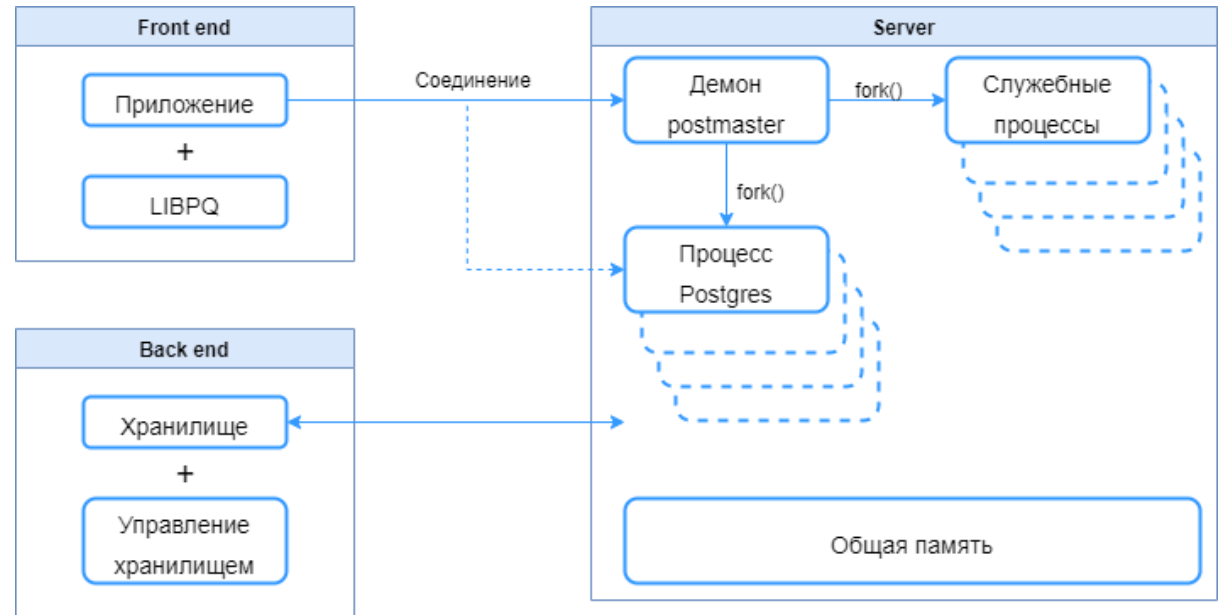
Выбор PostgreSQL аргументирован

- доступностью исходного кода;
- кроссплатформенностью.

# Анализ предметной области

Выделяют 3 основные подсистемы:

- клиентская часть
- серверная часть
- хранилище данных



# Многопоточный доступ в МРР системах

## Преимущество реализации:

- при работе с БД из 100000 записей, время выполнения запросов примерно в 1000 раз выше у однопоточной программы.
- Однопоточная модель показывает нестабильную работу на больших данных.

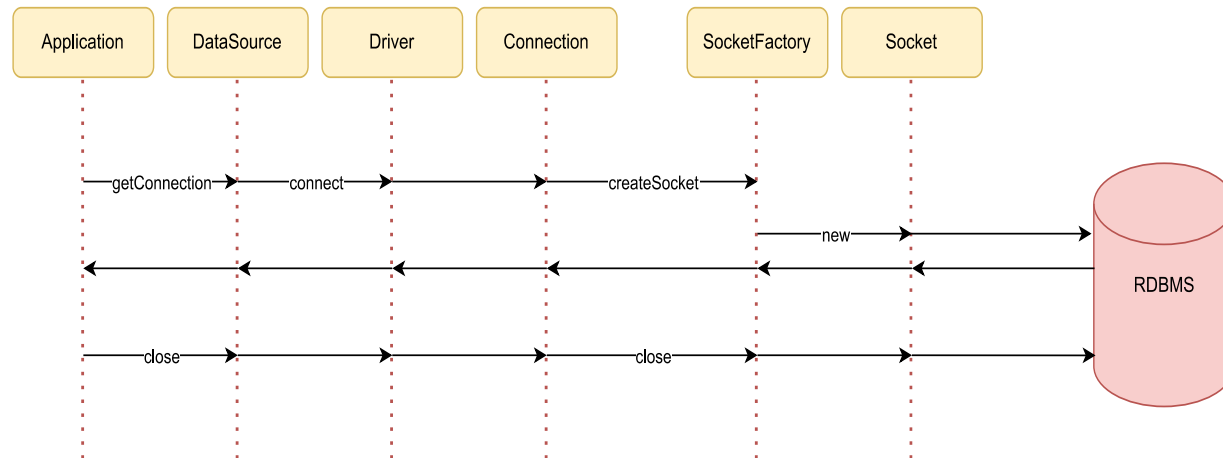
Рост количества потоков может привести к замедлению работы программы.

## Причины замедления:

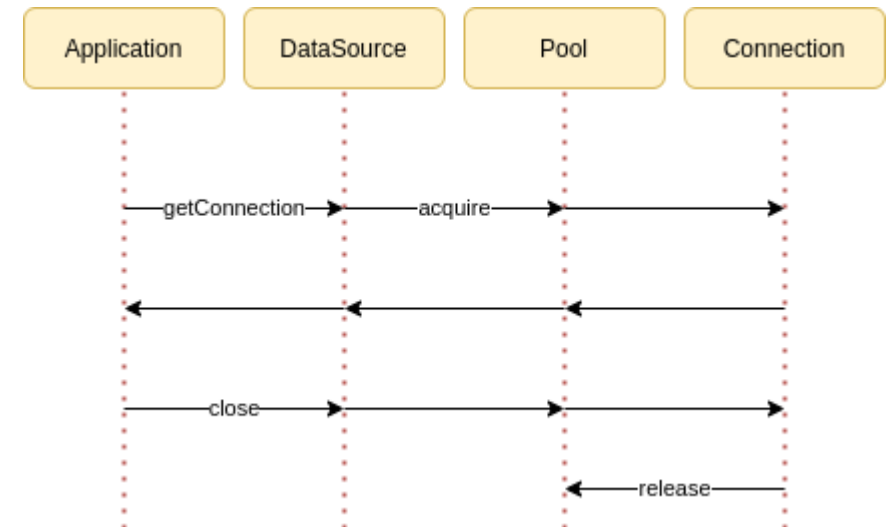
- повышенная нагрузка на системные ресурсы
- увеличение конкуренции при обращении к ресурсам
- проблема превышения количества подключений на сервере:
  - долгое ожидание дальнейших запросов
  - отклонение запросов

# Анализ существующих решений.

## Пул соединений



Цикл соединения с БД без пула



Цикл соединения с БД с использованием пула

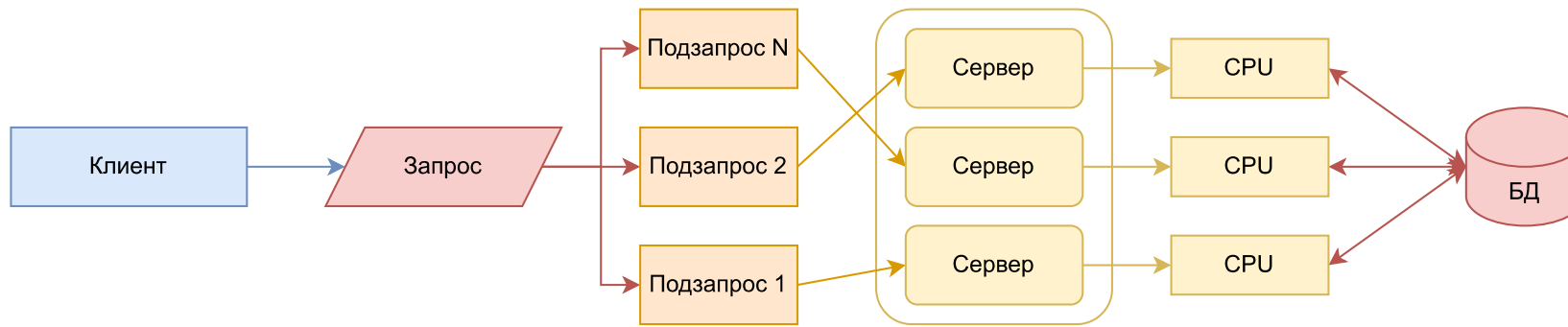
# Анализ существующих решений.

## Пул соединений

Внешний пул	Встроенный пул
Увеличение пропускной способности транзакции на 60%	
Ограничение максимального количества одновременных подключений к БД	
Необходимость поддержки отдельного пула соединения для каждой БД	
Сложность конфигурации пула	Доступен только в коммерческой версии
Сложность встраиваемости в код	

# Анализ существующих решений.

## Параллельное выполнение запроса



### Недостатки метода:

- применим к малому числу запросов;
- снижает производительность;
- эффективен только для конкретных запросов.