

# Метод 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** - разработка и реализация метода параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения.

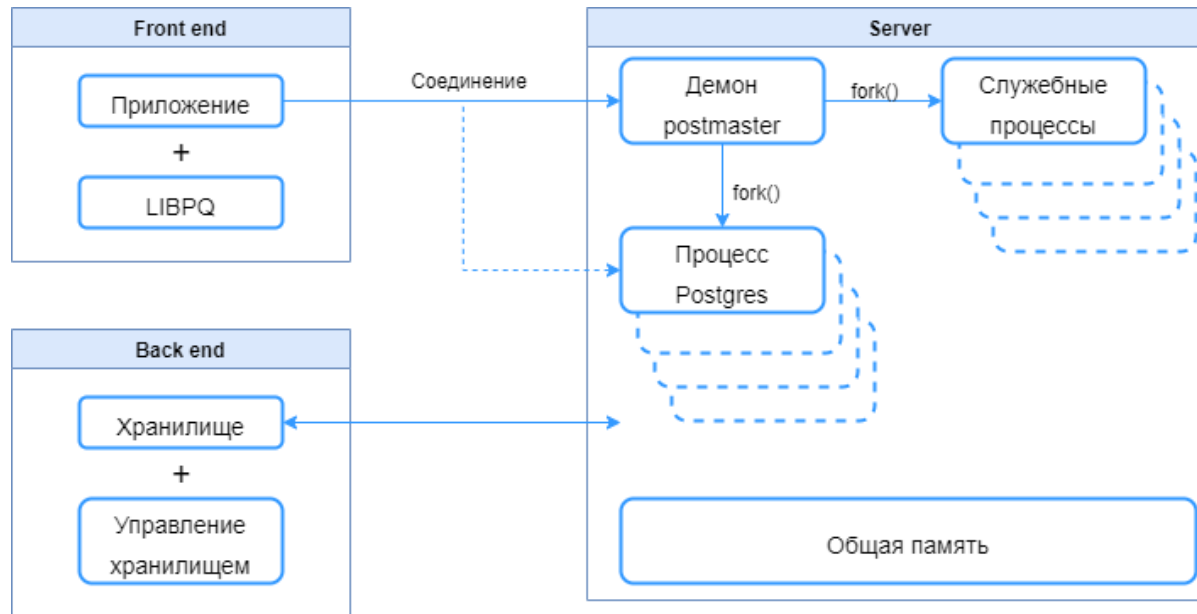
## **Задачи:**

- Анализ предметной области и существующих методов реализации многопоточного доступа в МРР системах;
- Разработка метода параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения;
- Реализация программного модуля для СУБД PostgreSQL;
- Проведение сравнительного анализа стандартного метода обработки запросов к СУБД PostgreSQL с реализуемым методом.

# Формализация задачи

- Работа пользовательского приложения, взаимодействующего с БД, основана на многопоточности. Требуется оптимизировать работу приложения путем реализации метода, позволяющего выполнять одно соединение, которое будет передано в каждый поток.

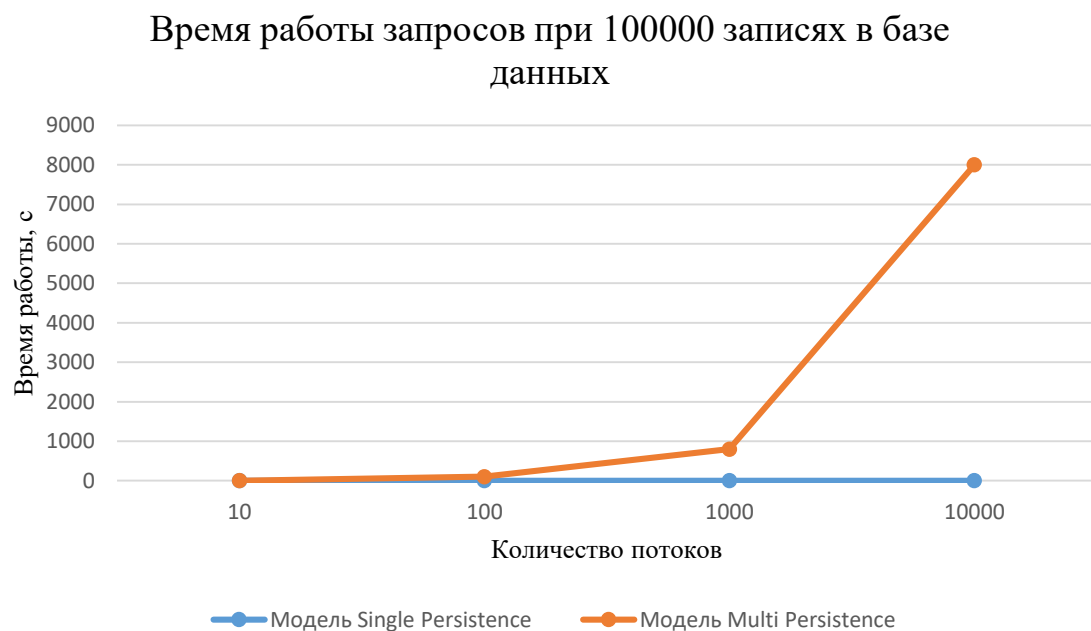
# Анализ предметной области



Выделяют 3 основные подсистемы:

- клиентская часть
- серверная часть
- хранилище данных

# Многопоточный доступ в МРР системах



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

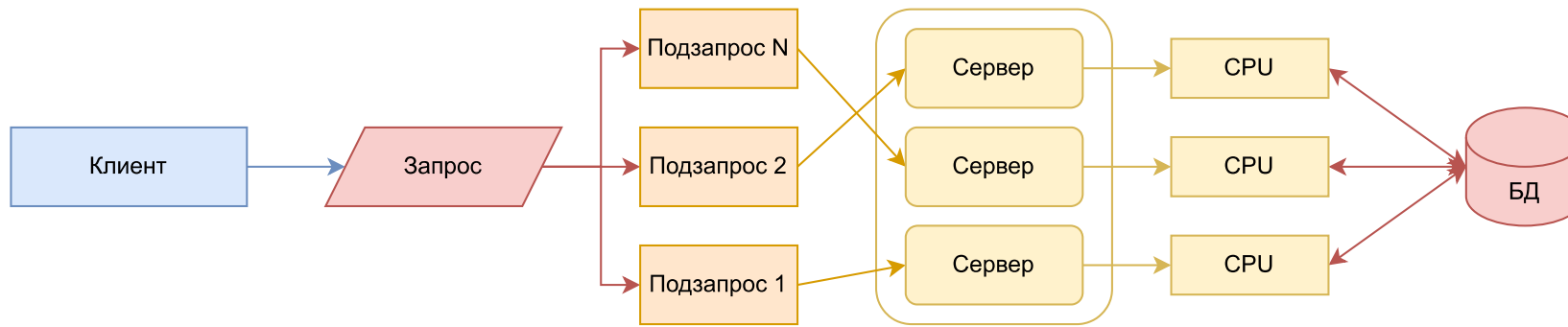
# Анализ существующих решений.

## Пул соединений

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.

## Распараллеливание запроса



### Недостатки метода:

- Применим к малому числу запросов;
- Снижает производительность;
- Эффективен только для конкретных запросов.