

# Метод параллельного выполнения запросов к системе управления базами данных PostgreSQL в пределах одного соединения

Студент: Платонова Ольга Сергеевна

Группа: ИУ7-85Б

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

Консультант: Гаврилова Юлия Михайловна

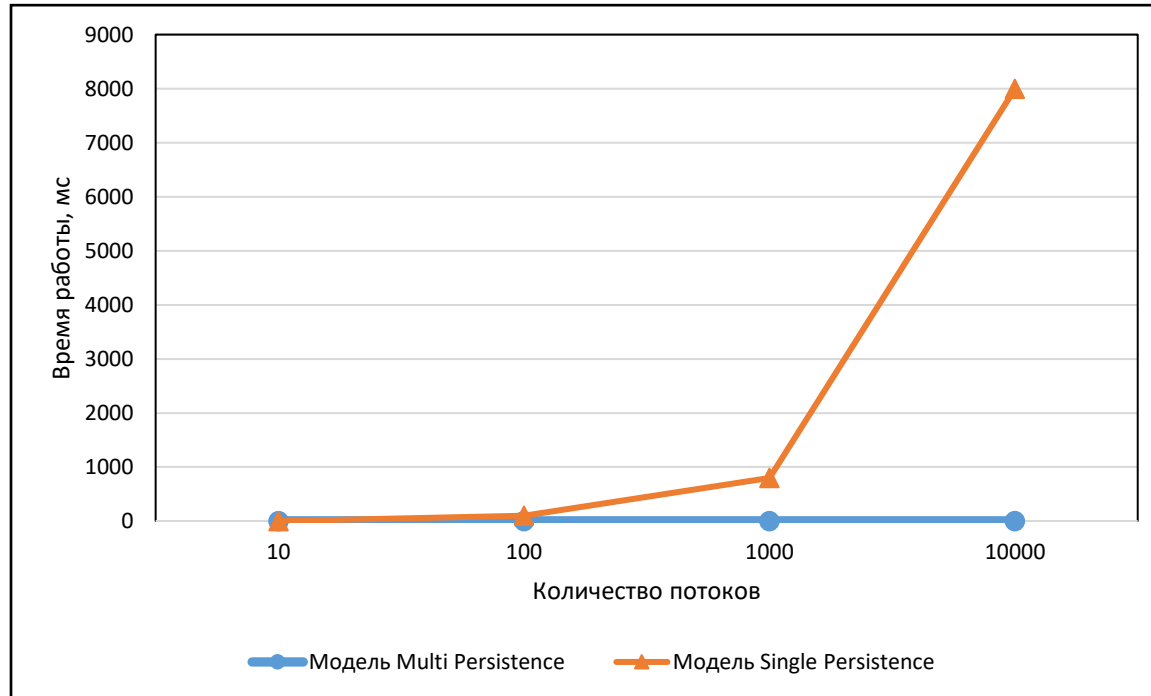
# Цель и задачи работы

**Цель** — разработать метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения.

## **Задачи:**

- Выполнить анализ предметной области и существующих методов выполнения запросов в MPP системах;
- Разработать метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения;
- Реализовать разработанный метод;
- Выполнить сравнительный анализ времени работы метода и различных реализаций выполнения запросов.

# Однопоточное и многопоточное соединение с базой данных

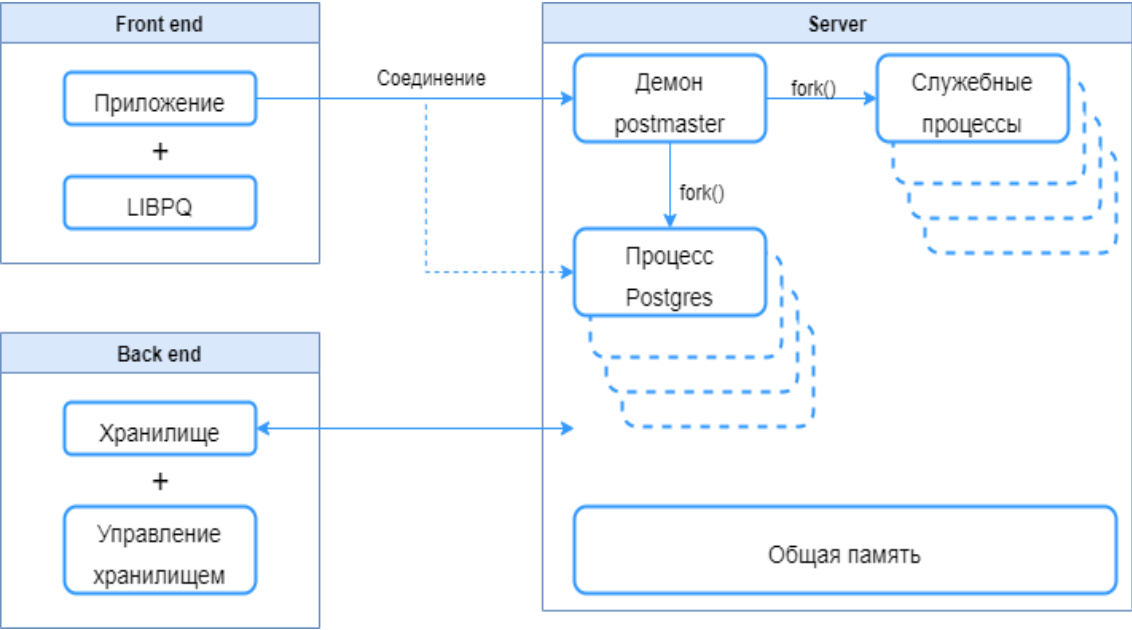


Доступ к БД объемом 100.000 записей.  
Многопоточная программа:

- в 1000 раз быстрее;
- ошибка памяти при 10.000 и более соединений.

Процесс подключения занимает  
2-3 МБ памяти.

# Создание соединения в PostgreSQL



Архитектура PostgreSQL

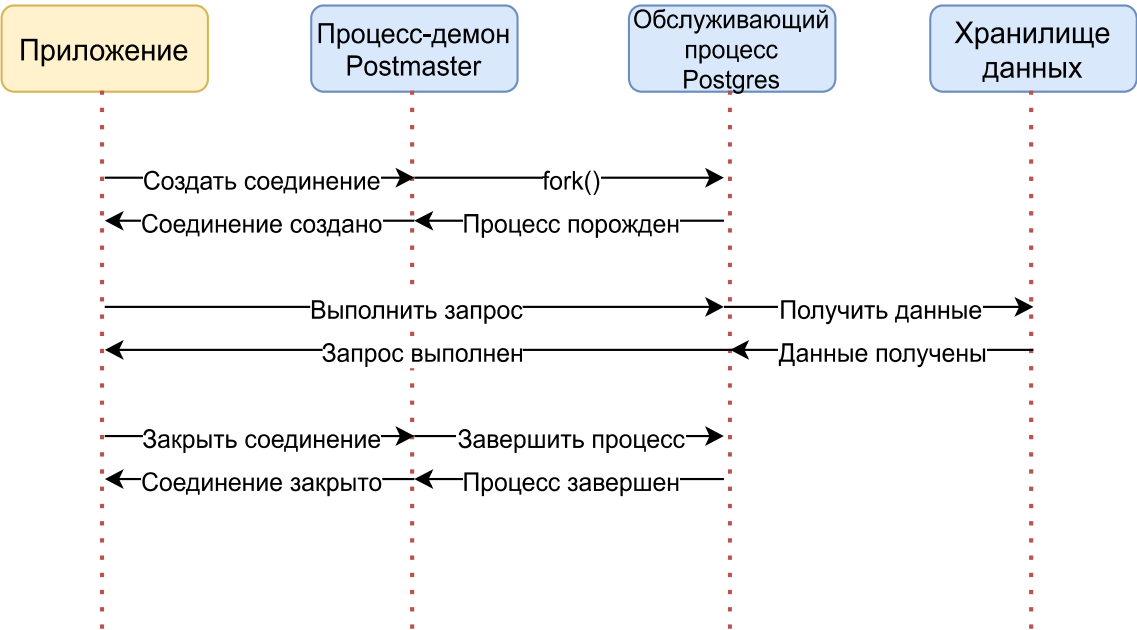


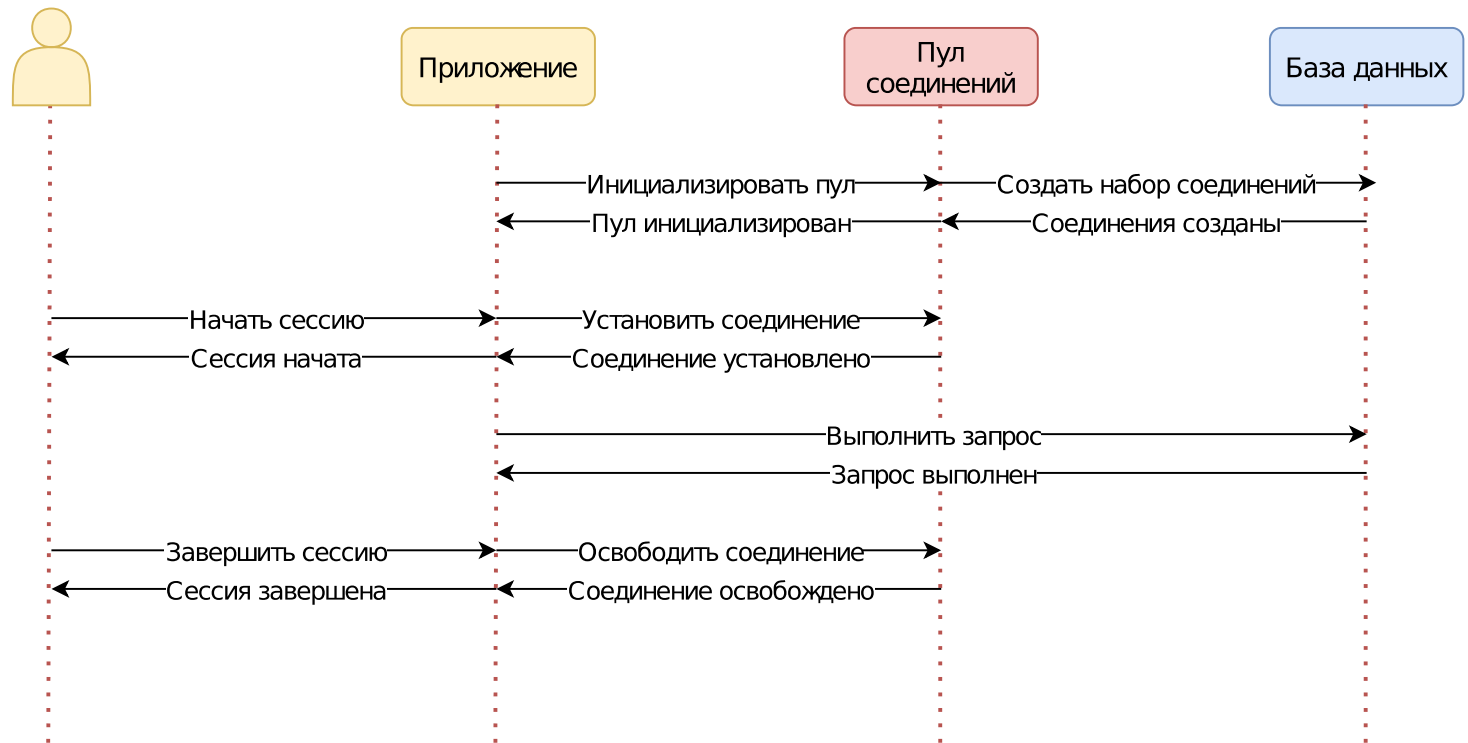
Диаграмма последовательности создания соединения

# Пул соединений

*Пул соединений* — набор открытых и готовых к использованию соединений с базой данных.

Особенно заметно повышение производительности, когда размер пула не превосходит числа потоков.

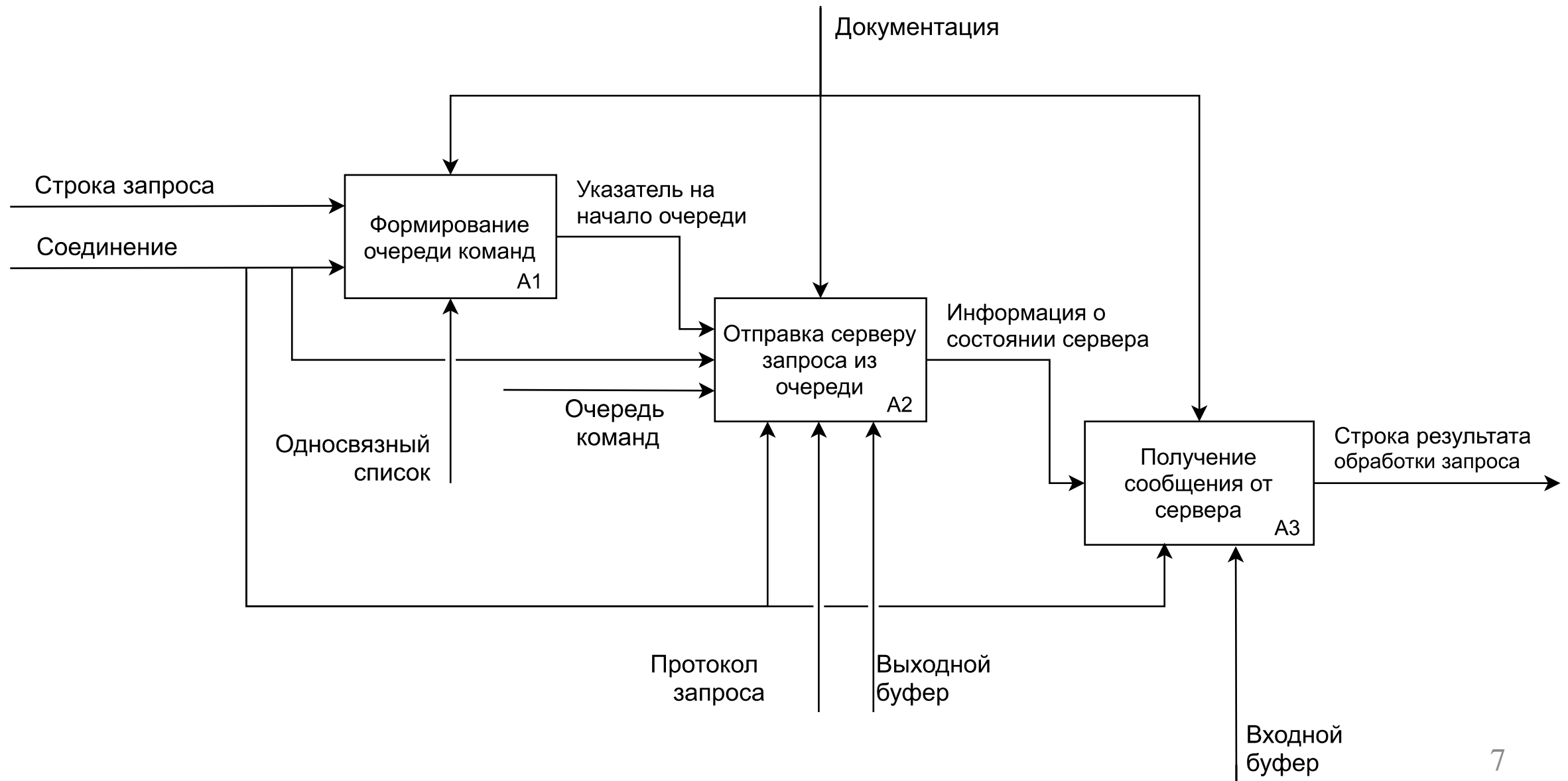
- Встроенный пул
- Пул на основе `libpq`
- Пул в качестве внешней службы



# Анализ существующих решений

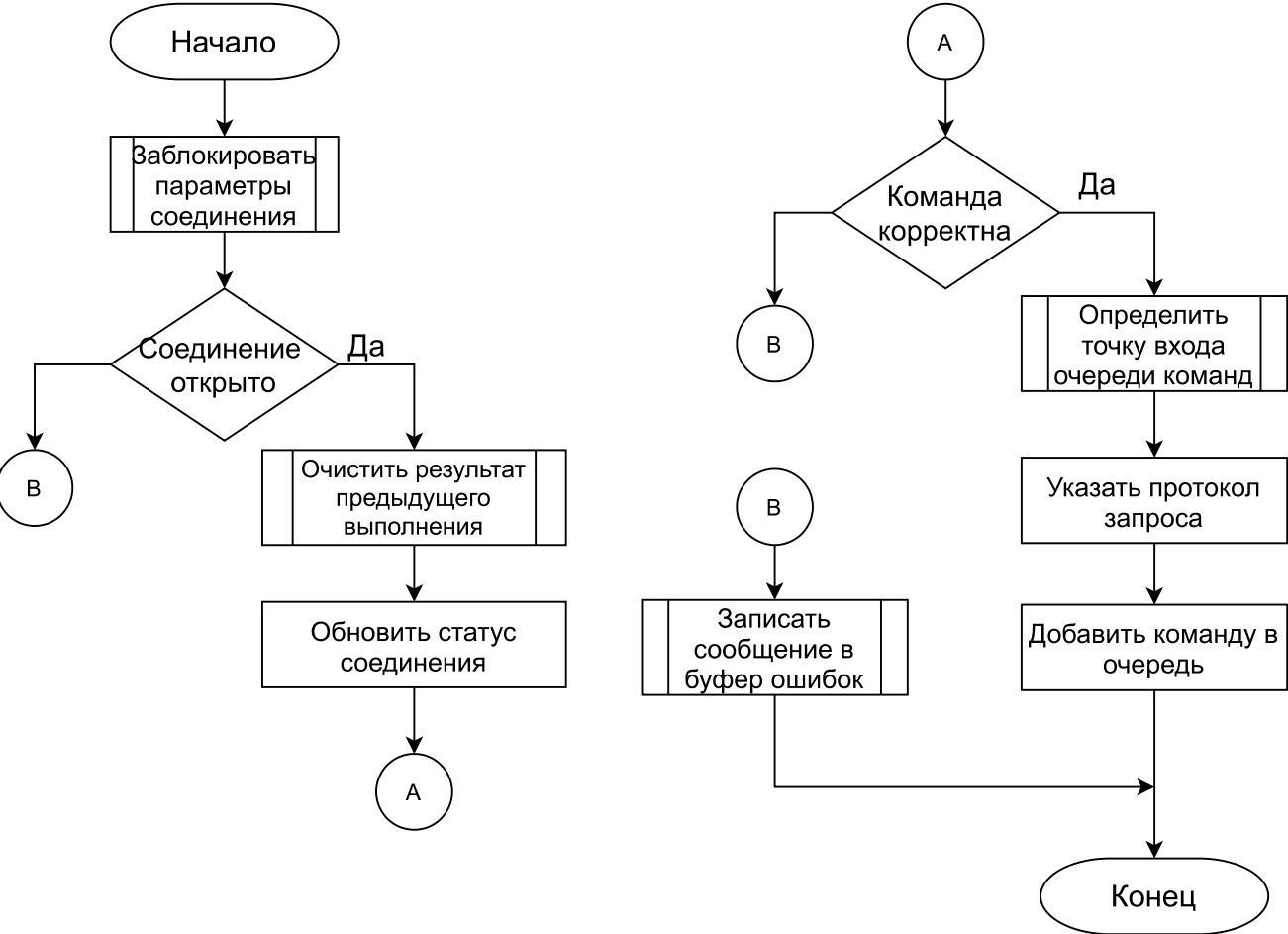
<b>Критерий \ Вид пула</b>	<b>Пул на основе libpq</b>	<b>Пул в качестве внешней службы</b>	<b>Встроенный пул</b>
Максимальный размер пула по умолчанию	100	100	32767
Наличие лицензии	Нет	Нет	Да
Необходимость поддержки отдельного пула для каждой БД	Да	Да	Да
Необходимость самостоятельной разработки и поддержки	Да	Нет	Нет

# Функциональная диаграмма разрабатываемого метода

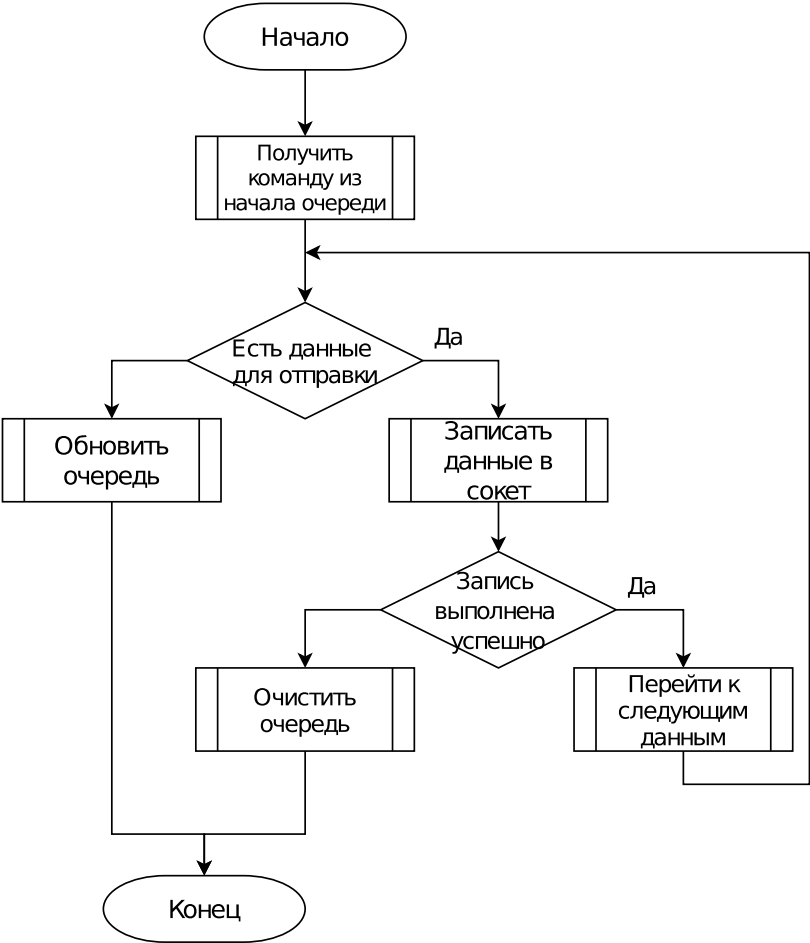


# Формирование очереди команд и отправка серверу запроса из очереди

Алгоритм формирования очереди команд

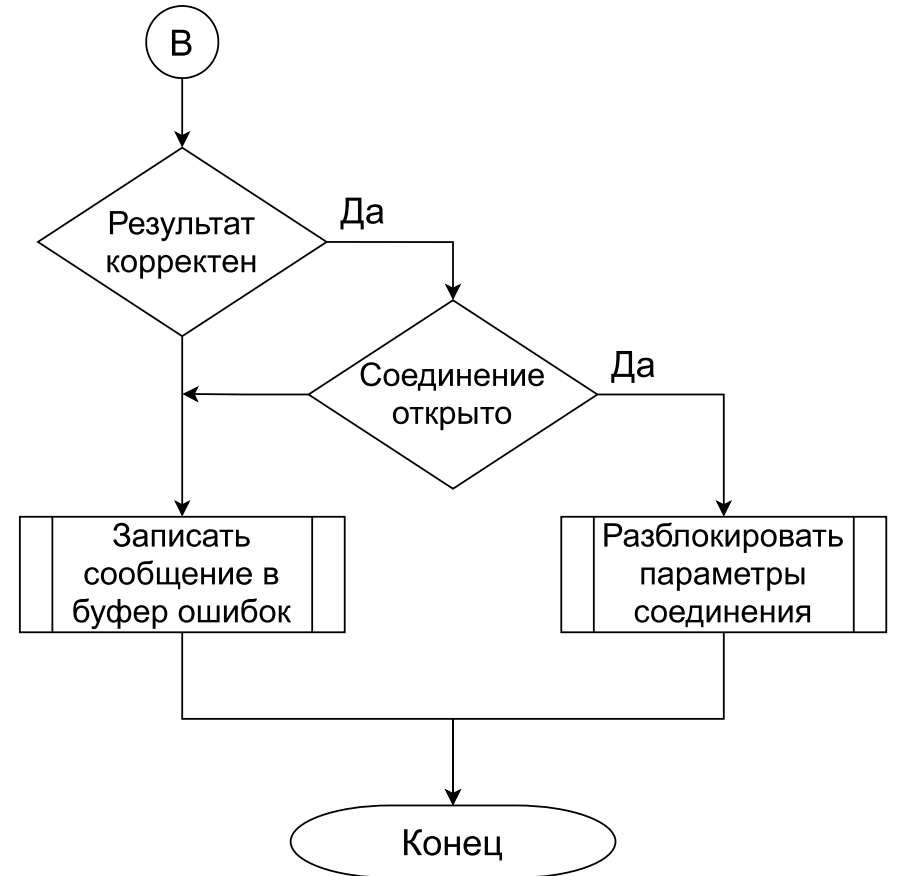
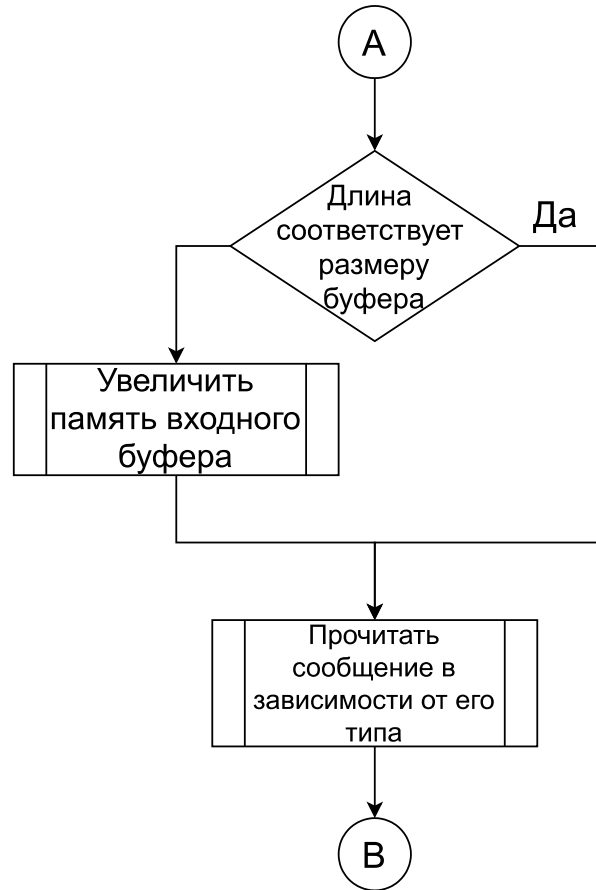
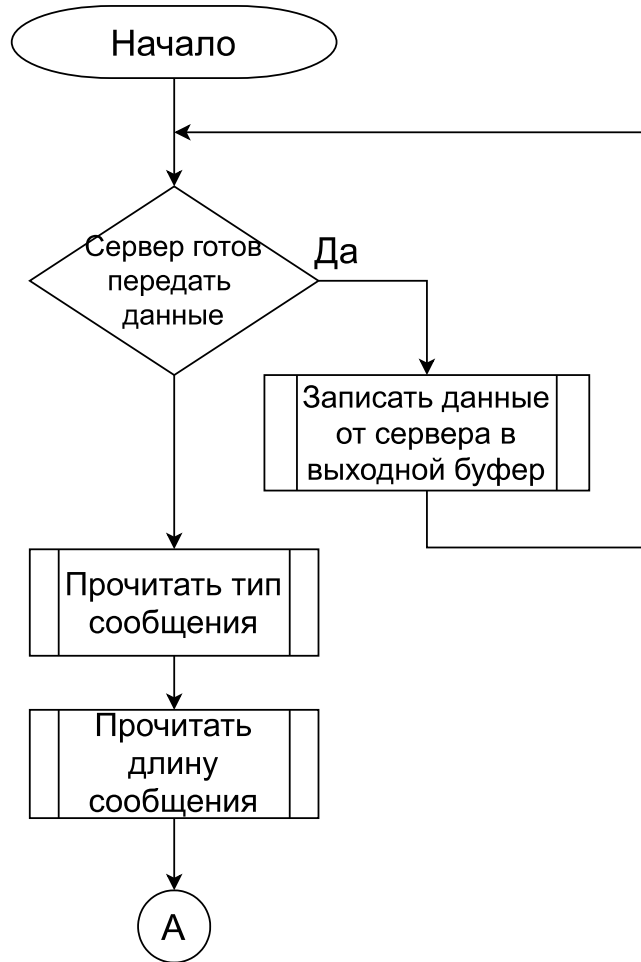


Алгоритм отправки запроса серверу

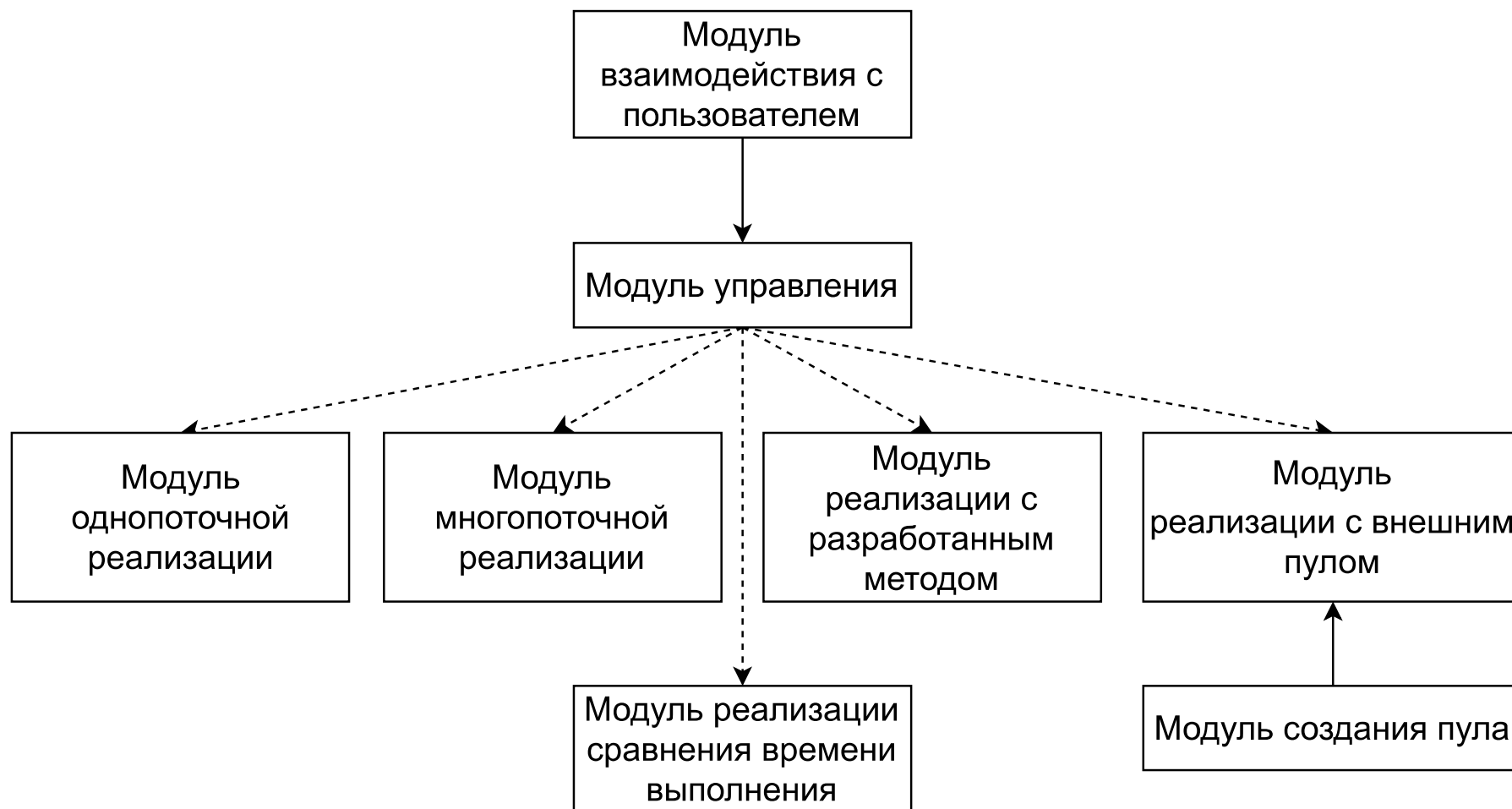




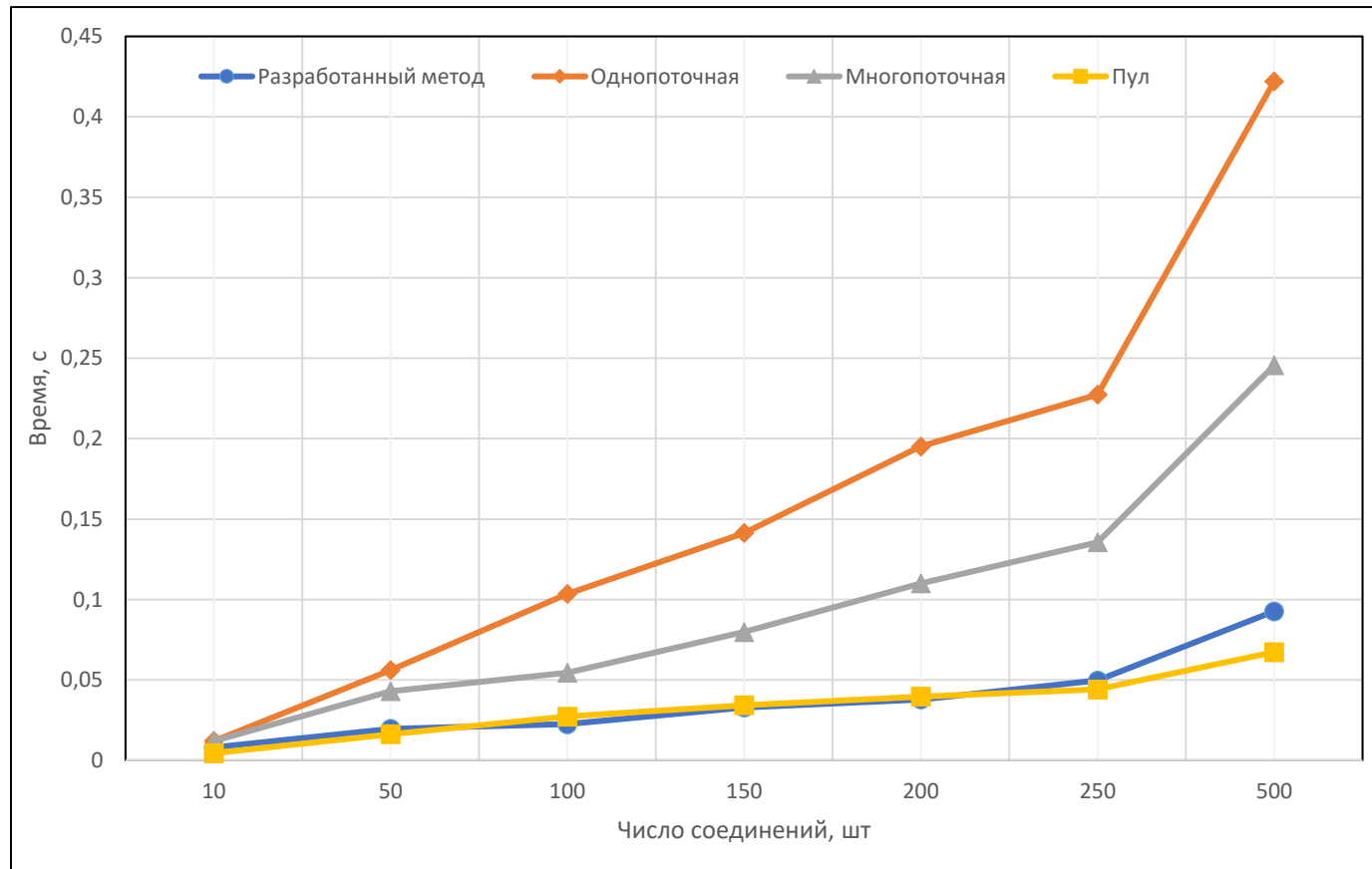
# Получение сообщения от сервера



# Внешний модуль взаимодействия с разработанным методом



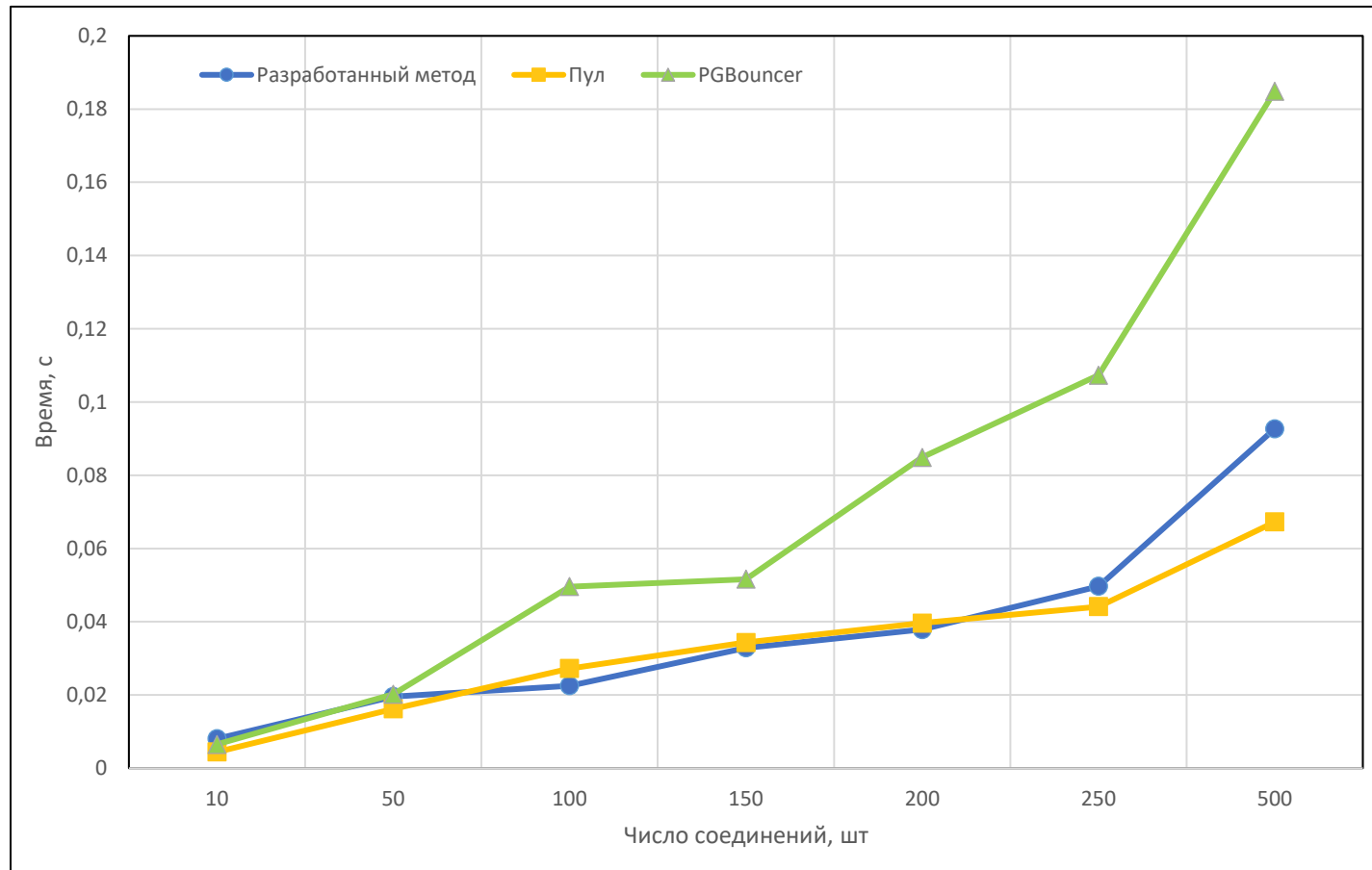
# Зависимость времени создания соединения от числа подключений к базе данных



Для каждого опыта учитывалось время:

- создания соединения;
- выполнения запроса;
- очистки результата выполнения;
- закрытия соединения.

# Зависимость времени работы реализованного метода и пулов соединений от числа подключений



Размер пула:

- $\frac{3}{4}$  фактического числа создаваемых соединений.

# Исследование требуемых ресурсов для выполнения простого запроса

Простой запрос — выборка всех данных посредством одного оператора ‘SELECT’ и одного оператора ‘FROM’.

SELECT \* FROM table100;  
Создание 10 соединений.

Реализация	Число раз выделения памяти	Суммарный объем используемой памяти
Однопоточная	729	588,870 байт
Многопоточная	812	593,508 байт
Внешний пул	831	586,212 байт
Разработанный метод	182	180,794 байт

# Заключение

*Цель достигнута:* разработан метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения.

*Поставленные задачи решены:*

- Выполнен анализ предметной области и существующих методов выполнения запросов в MPP системах;
- Разработан метод параллельного выполнения запросов к СУБД PostgreSQL в пределах одного соединения;
- Реализован разработанный метод;
- Выполнен сравнительный анализ времени работы метода и различных реализаций выполнения запросов.

# Дальнейшее развитие

- Реализация пользовательского вывода информации об ошибках в случае конкатенации нескольких запросов в одну команду.
- Разработка метода управления ресурсным пулом в случае потери потоком соединения с базой данных.