Реализация теста Graph500 для параллельной СУБД PargreSQL

Научный руководитель:
кандидат физ.-мат. наук, доцент
М.Л. Цымблер

Автор работы:
Студент группы ВМИ-456
А.Ю. Сафонов

Рецензент:
кандидат пед. наук
А.Ю. Эвнин

Цель и задачи исследования

Цель:

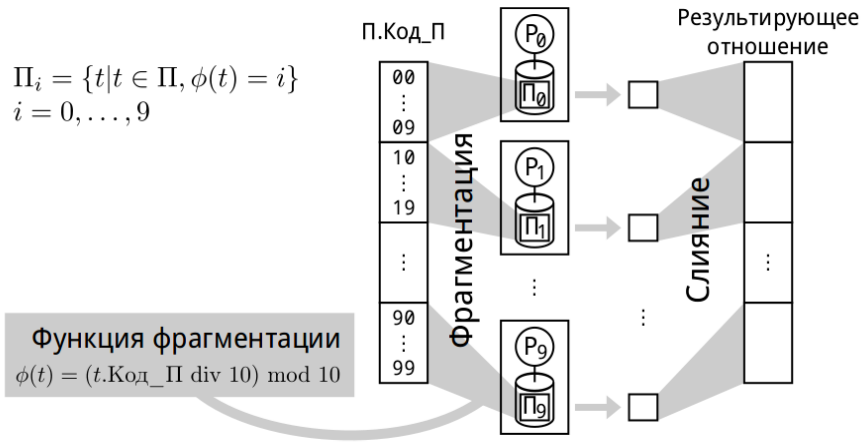
Оценка эффективности параллельной СУБД PargreSQL на задачах интенсивной обработки данных с помощью сравнительного теста Graph500

Задачи:

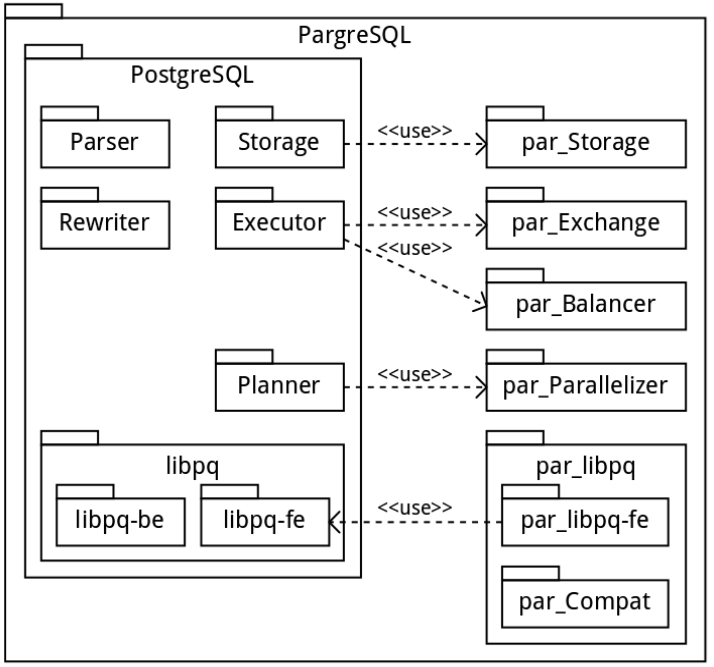
- Изучить архитектуру параллельной СУБД PargreSQL и спецификацию теста Graph500
- Разработать схему базы данных для хранения графа и промежуточных данных в соответствии со спецификацией теста Graph500
- Выполнить проектирование и разработку алгоритмов на языке SQL, реализующих тест Graph500 для параллельной СУБД PargreSQL
- Провести вычислительные эксперименты на суперкомпьютере «Торнадо ЮУрГУ», исследующие эффективность параллельной СУБД PargreSQL на тесте Graph500

Параллельная СУБД PargreSQL

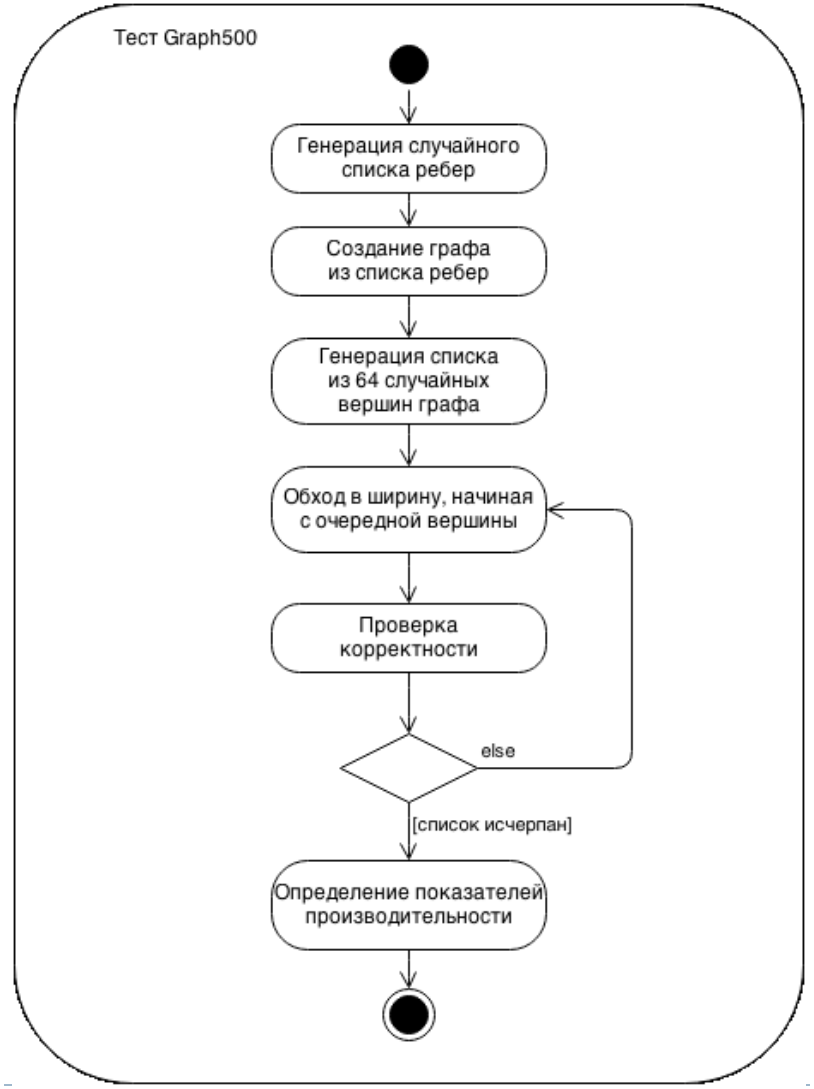
- Фрагментный параллелизм



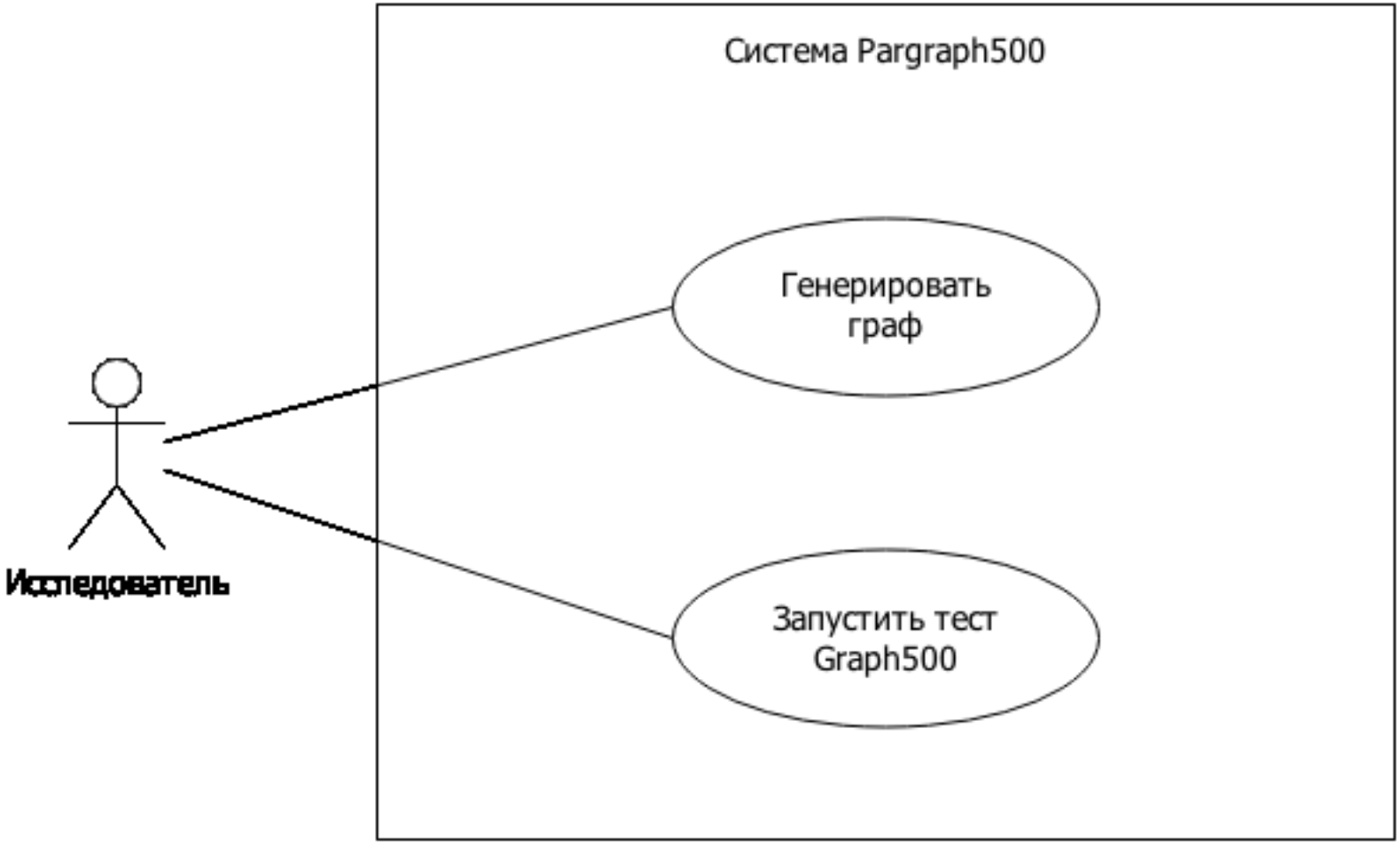
- Архитектура PargreSQL



Тест Graph500



Варианты использования системы



Модульная структура

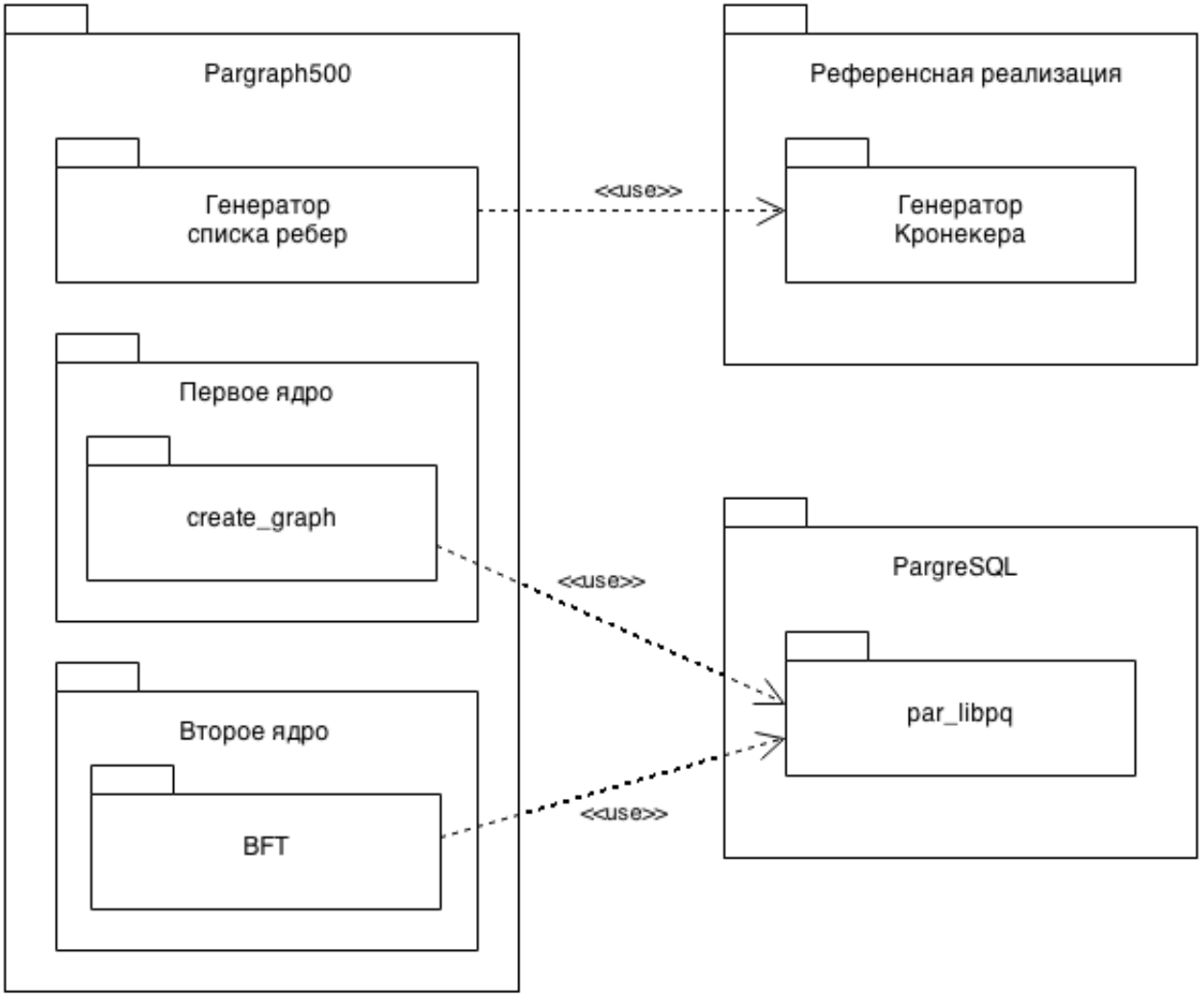


Схема классов

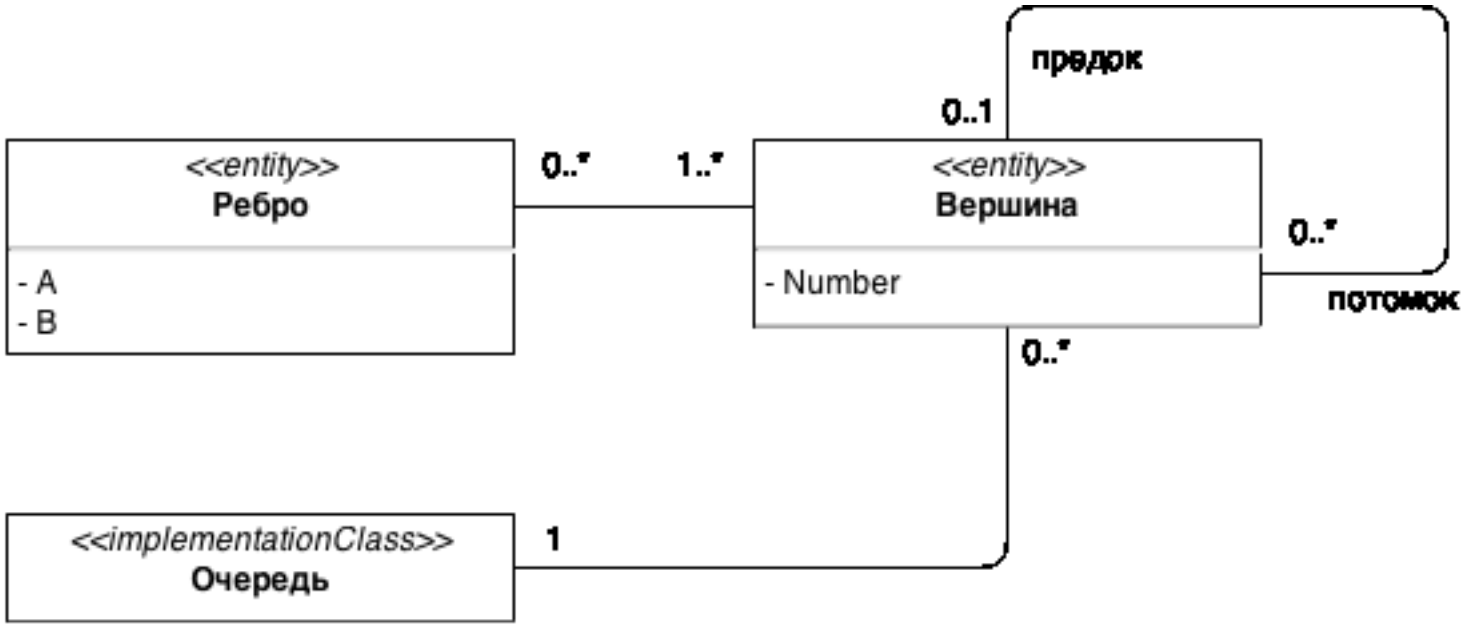
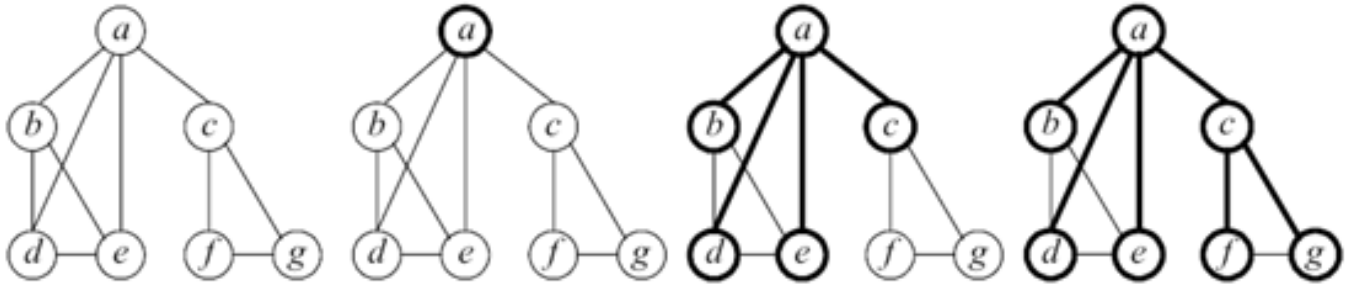


Таблица в PargreSQL

```
create table graph (
    id bigint,              -- порядковый номер дуги
    a_id bigint,            -- номер начальной вершины
    b_id bigint,            -- номер конечной вершины
    parent bigint,          -- номер вершины родителя a_id
    queue bigint            -- номер в очереди вершины a_id
) with (fragattr = id);     -- функция фрагментации
```

Обход в ширину




```
BFS (start_node) {
    for (all nodes i) {
        parent[i] = -1;
    }

    queue.push(start_node);
    parent[start_node] = start_node;

    while( !queue.empty() ) {
        node = queue.pop();
        foreach(child in expand(node)) {
            if(parent[child] == -1) {
                queue.push(child);
                parent[child] = node;
            }
        }
    }
}
```

Проверка корректности

Для каждого ключа поиска после завершения обхода графа в ширину нужно проверить, что соблюдены условия:

1. Полученное BFS-дерево является деревом и не содержит циклов.
2. Каждое ребро BFS-дерева соединяет вершины, чей уровень при обходе в ширину различается ровно на 1.
3. Каждое ребро из входного списка ребер соединяет вершины, уровень которых в BFS-дереве различается не более чем на единицу или же обе эти вершины не включены в BFS-дерево.
4. BFS-дерево связывает все вершины данной компоненты связности.
5. Каждая вершина и ее родитель соединены ребром в исходном графе.

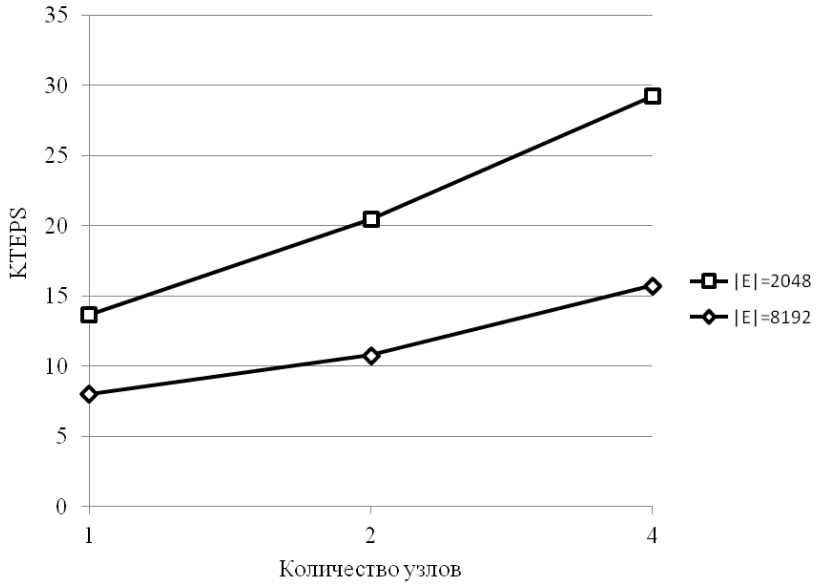
Измерение производительности

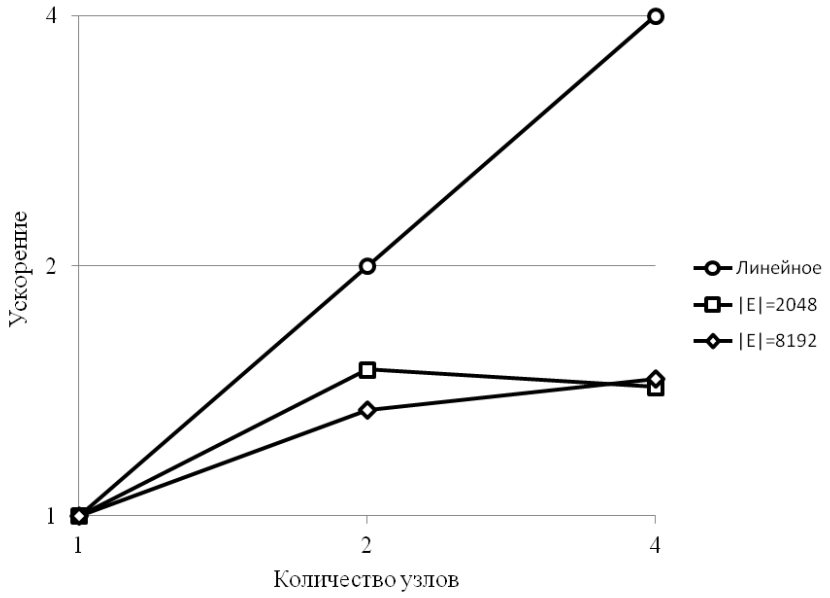
- TEPS (traversed edges per second) — количество пройденных ребер за секунду:

$$TEPS = \frac{m}{time_{K2}}$$

- m — количество ребер в итоговом дереве обхода
- $time_{K2}$ — время работы алгоритма обхода в ширину

Эксперименты





Основные результаты

- Изучена архитектура параллельной СУБД PargreSQL и спецификация теста Graph500
- Разработана схема базы данных для хранения графа и промежуточных данных в соответствии со спецификацией теста Graph500
- Выполнено проектирование и разработка алгоритмов на языке SQL, реализующих тест Graph500 для параллельной СУБД PargreSQL
- Проведены вычислительные эксперименты на суперкомпьютере «Торнадо ЮУрГУ», исследующие эффективность параллельной СУБД PargreSQL на тесте Graph500