

Двухуровневый алгоритм генерации оптимальных циркулянтных графов

Бусько П.В.¹, Козинев Е.А.¹, Волокитин В.Д.¹, Рзаев Э.Р.², Мееров И.Б.¹

¹Нижегородский государственный университет им. Н.И. Лобачевского, Нижний Новгород

²Национальный исследовательский университет «Высшая школа экономики», Москва

«Суперкомпьютерные дни в России»
Москва, 28-29 сентября 2026 г.

Понятие циркулянтного графа (1)

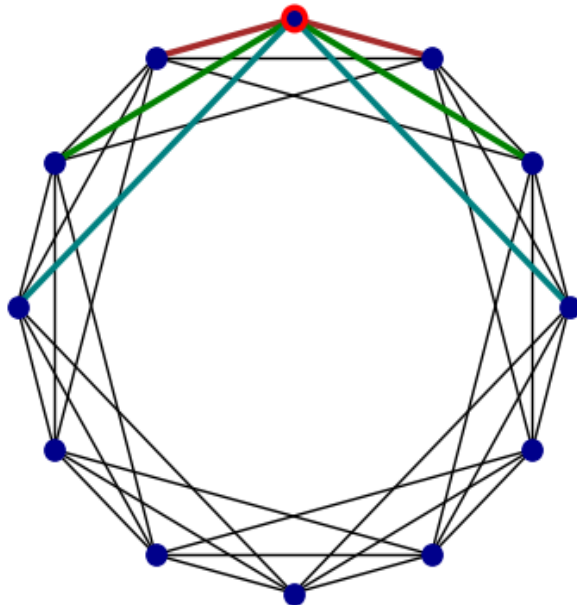
- *Циркулянтный граф (циркулянт)* – неориентированный граф $G = (V, E)$

$V = \{0, 1, \dots, N - 1\}$ – множество вершин,

$E = \{(v_i, v_j) : |i - j| \equiv s_m \pmod{N}, m = 1, \dots, K\}$, где $s_1, s_2, \dots, s_K, N \in \mathbb{Z}, s_i \leq \left\lfloor \frac{N}{2} \right\rfloor$ – множество ребер.

- *Сигнатура циркулянта*: набор $C(N; s_1, s_2, \dots, s_K)$, где N – количество вершин, K – размерность графа, $S = \{s_1, s_2, \dots, s_K\}$ – множество *образующих*, которые определяют шаги переходов между соседними вершинами.

Понятие циркулянтного графа (2)



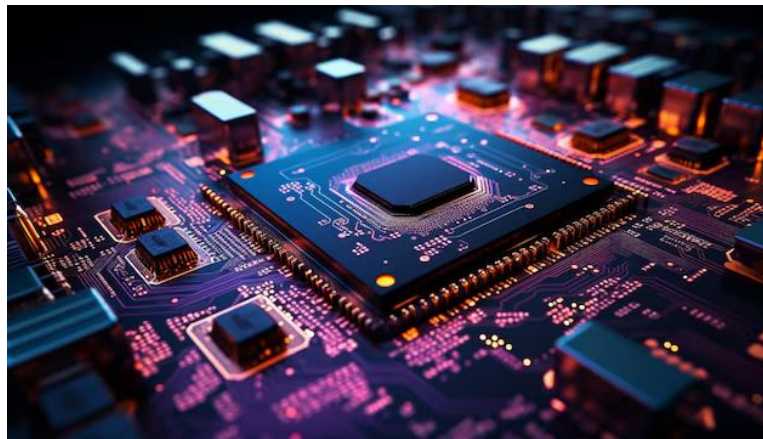
Циркулянтный граф $C(12; 1, 2, 3)$.

Образующие: $s_1 = 1$ (коричневый), $s_2 = 2$ (зеленый), $s_3 = 3$ (бирюзовый)

Области применения

- Теория кодирования
- Криптография
- Квантовые вычисления
- **Многопроцессорные вычислительные системы (*NetWork on Chip*)**

Ключевая задача при проектировании таких сетей – **выбор оптимального способа соединения ядер**, то есть топологии сети



Постановка задачи

- Пусть $d(u, v)$ – длина кратчайшего пути между вершинами v и u в графе $C(N; s_1, s_2, \dots, s_K)$
- Диаметр графа: $diam = \max_{v_i, v_j \in V} d(v_i, v_j)$
- Средняя длина кратчайшего пути: $aspl = \sum_{v_i, v_j} \frac{d(v_i, v_j)}{N \cdot (N-1)}$
- $\Omega(N, K) = \left\{ S = (s_1, \dots, s_K) : 1 \leq s_1 \leq \dots \leq s_K \leq \left\lfloor \frac{N}{2} \right\rfloor \right\}$ – множество всех допустимых наборов образующих
- Задача: $D_{min} = \arg \min_{s \in \Omega} diam(s)$
 $S_{opt} = \arg \min_{s \in D_{min}} aspl(s)$ S_{opt} – искомое множество

Поставленная задача

- NP-трудная
- Решается полным перебором
- Общее количество возможных вариантов: $C_{\lfloor \frac{N}{2} \rfloor}^K$
- **Есть необходимость в использовании параллельных вычислений**

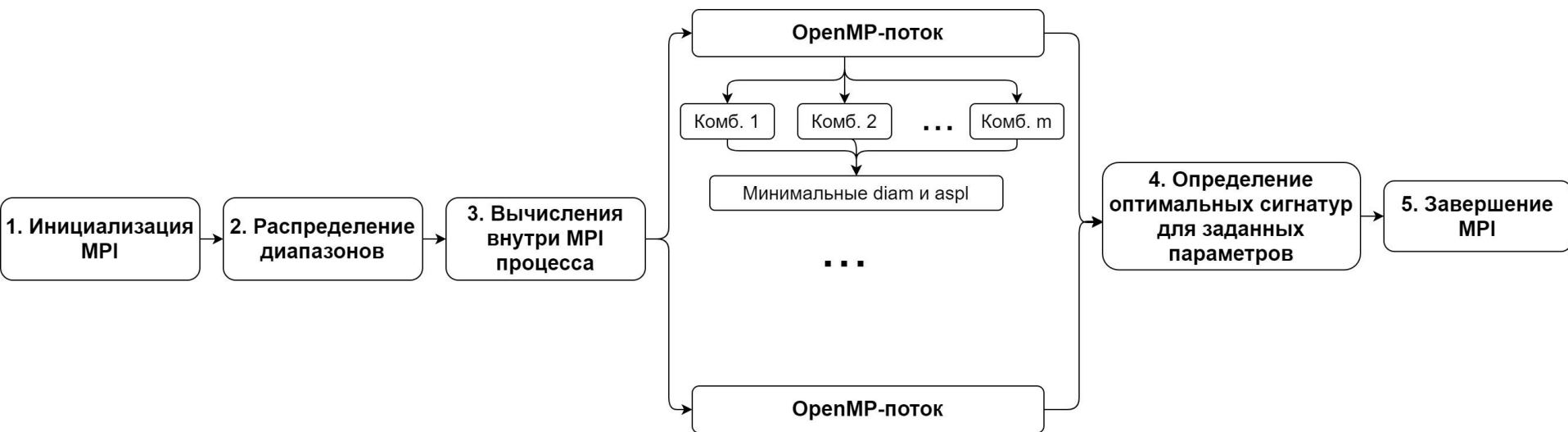
Цель работы

Оптимизация программного обеспечения для генерации оптимальных циркулянтных топологий.

Результат

- Реализован новый двухуровневый MPI + OpenMP алгоритм для генерации оптимальных циркулянтов для кластерных систем.
- Проведены вычислительные эксперименты, выполнен их анализ.

Двухуровневый алгоритм для распределенной памяти

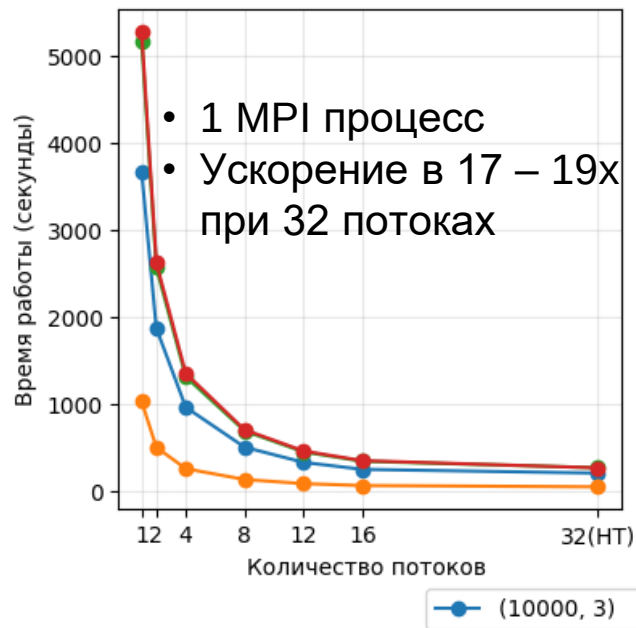


Тестовая инфраструктура

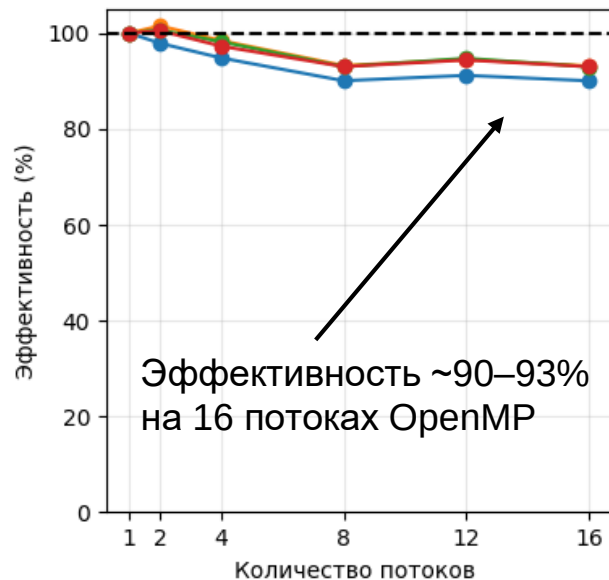
- **Суперкомпьютер «Лобачевский»**, 2 процессора Intel Sandy Bridge E5-2660, 8 ядер каждый (поддерживается гипертрединг), 64 ГБ ОЗУ. ОС Linux. компилятор Intel oneAPI Compiler 2022, Intel MPI 2021.
- **Комбинации параметров** для генерации циркулянтных графов:
 - Количество вершин, $N = 250, 500, 600, 700, 800, 900, 1000, 1500$
 - Размерность, $K = 2, 3, 4, 5$
- Запуски проводились на 1 – 32 узлах кластера при различном сочетании процессов MPI и потоков OpenMP.
- На нескольких узлах кластера запуск проводился в оптимальной конфигурации числа процессов и потоков (MPI x OpenMP)

Результаты запусков на одном вычислительном узле

Зависимость времени работы алгоритма от количества потоков

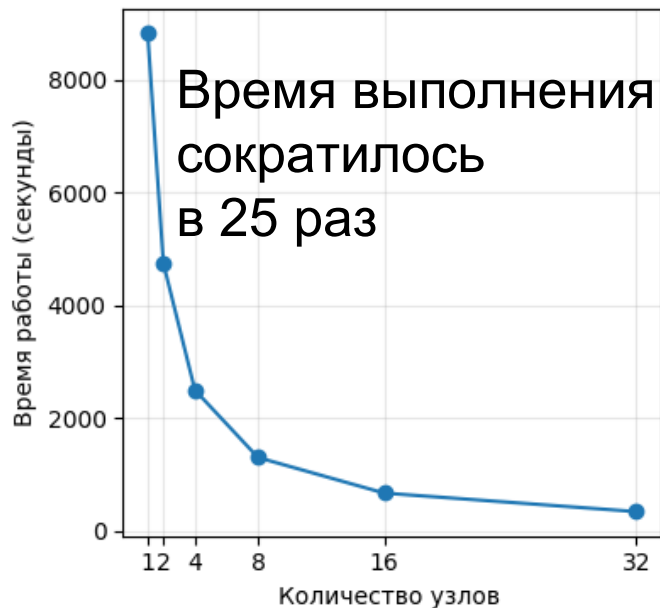


Зависимость эффективности распараллеливания от количества OpenMP-потоков

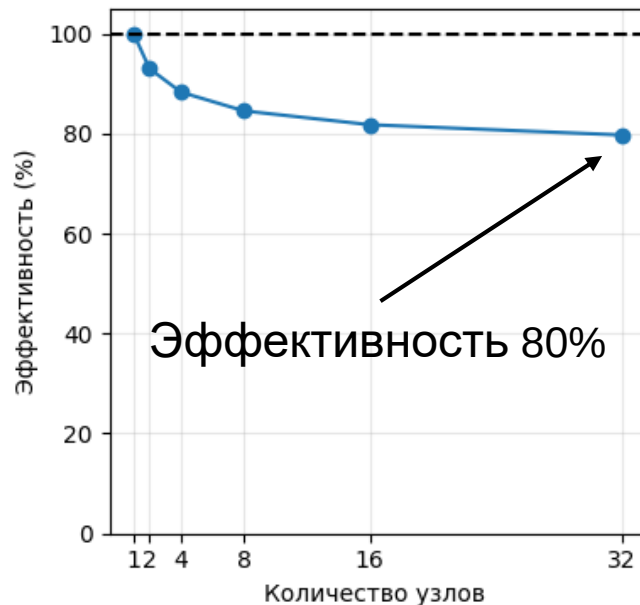


Результаты запусков на нескольких вычислительных узлах

Зависимость времени работы программы
от количества вычислительных узлов



Эффективность распараллеливания
двухуровневого алгоритма



Заключение

- Разработан двухуровневый алгоритм генерации оптимальных циркулянтных топологий для кластерных систем с использованием технологий MPI и OpenMP;
- На одном вычислительном узле достигнуто ускорение в 17-19 раз, эффективность на 16 потоках составляет 90–93%;
- При запуске на 32 вычислительных узлах время выполнения сократилось ~ в 25 раз, а эффективность составила 80%.