Разработка и оптимизация исполнительной системы фрагментированного программирования

Руководители:
Перепёлкин В.А.
Щукин Г.А.

Студенты:
Беляков С.А. гр.ПМИ-81 (2 курс)
Герман С.А. гр.ПМИ-81 (2 курс)

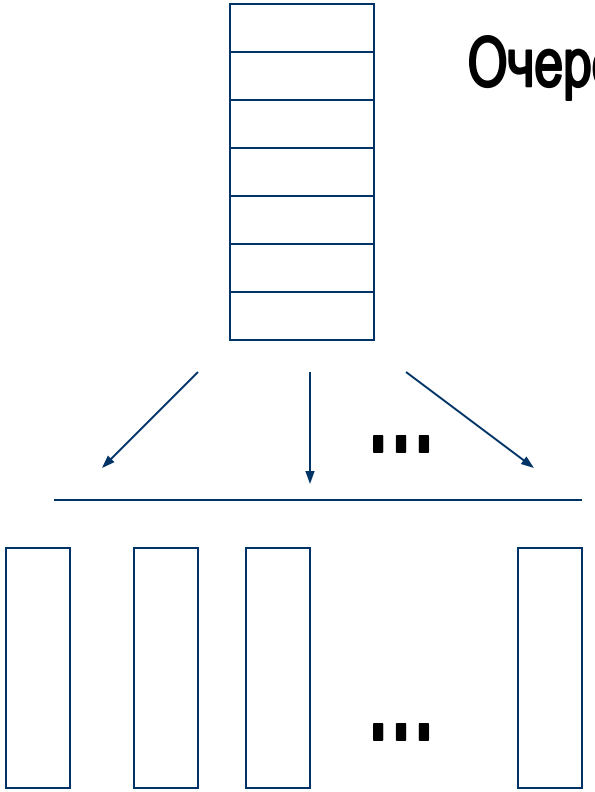
Цель работы

Распараллеливание исполнительной системы (ИС) фрагментированного программирования и её оптимизация

Постановка задачи

- Разработка многопоточной версии ИС
- Разработка гибридного варианта ИС (интеграция с модулем сетевых пересылок)
- Разработка и реализация алгоритма оптимизации плана исполнения фрагментированной программы
- Тестирование ИС на фрагментированных программах

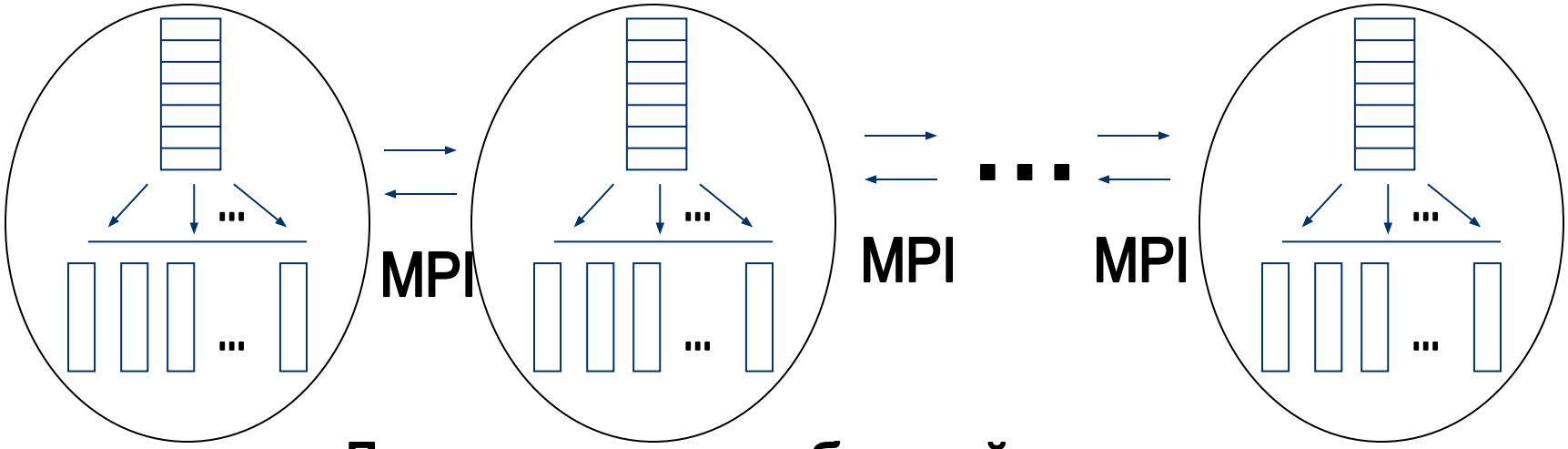
Схема реализации многопоточности



Очередь фрагментов вычислений

Потоки исполняются параллельно на одном и том же участке памяти
Доступ на запись данных на общих участках памяти контролируется посредством mutex'ов

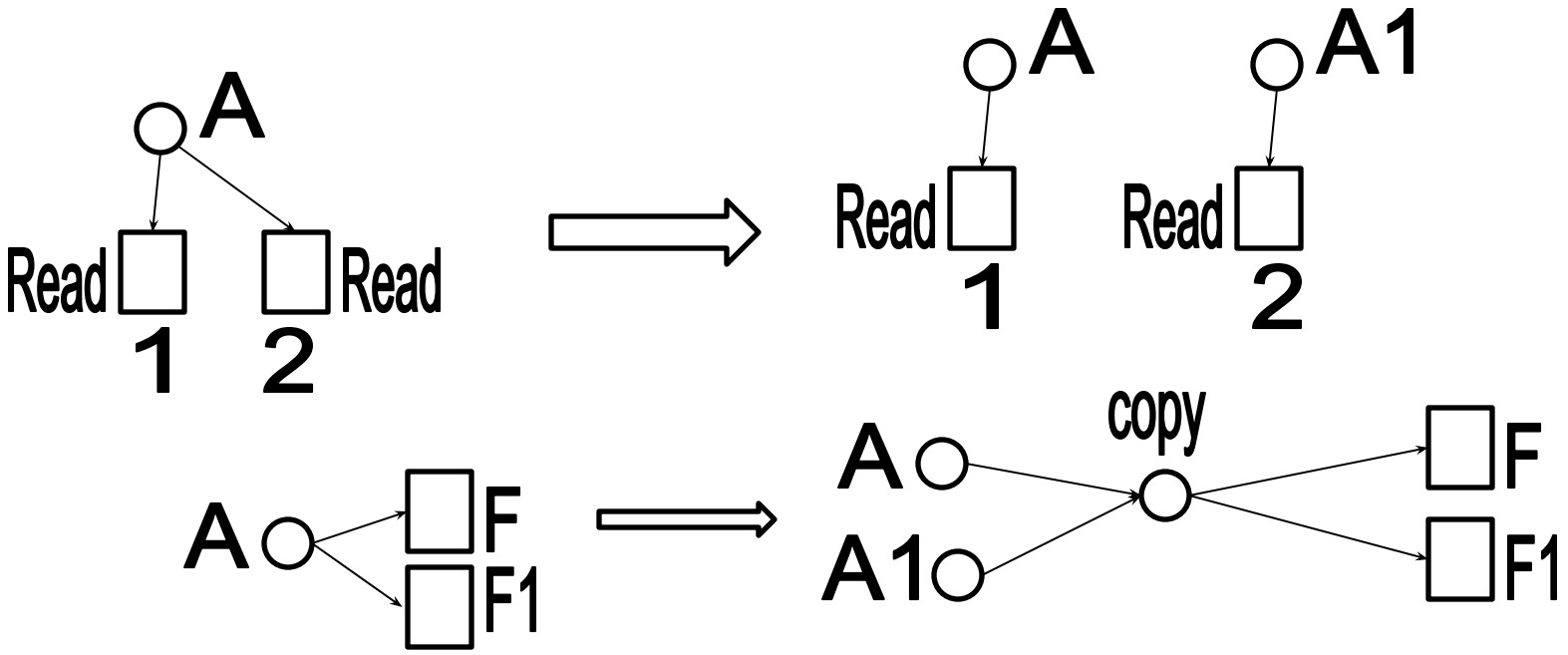
Гибридная схема



Для передачи сообщений между параллельно исполняемыми процессами используется технология MPI. В отличие от потоков, процессы работают на разных узлах и участках памяти.

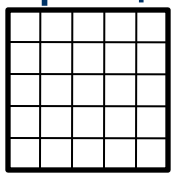
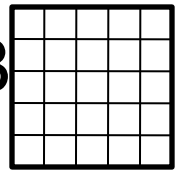
Оптимизация плана исполнения

- Для каждого фрагмента анализируется его очередь задач, после чего создаются копии фрагмента с усечёнными очередями, взятыми из очереди исходного фрагмента. Например, из A (1, 2, 3) имеем A (2, 3) и A1 (1), исполняющиеся параллельно.



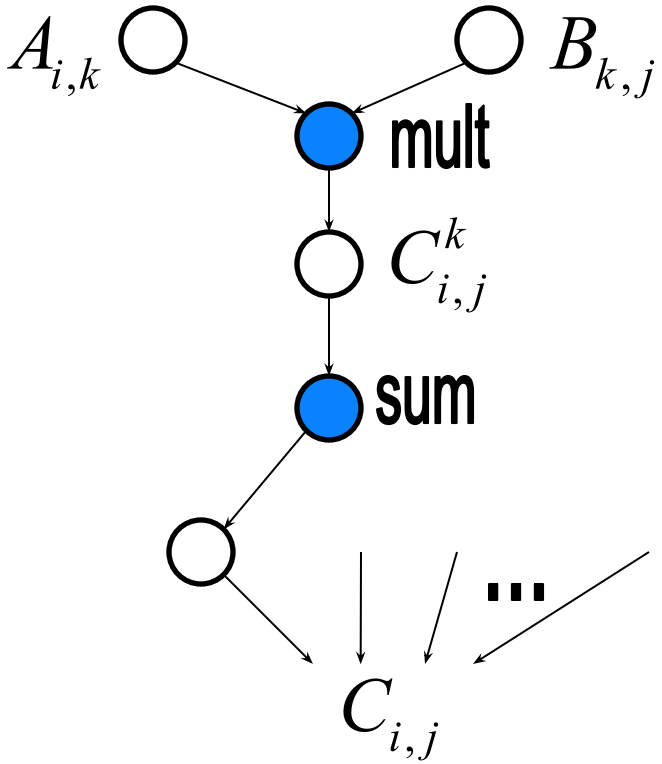
Перемножение плотных матриц

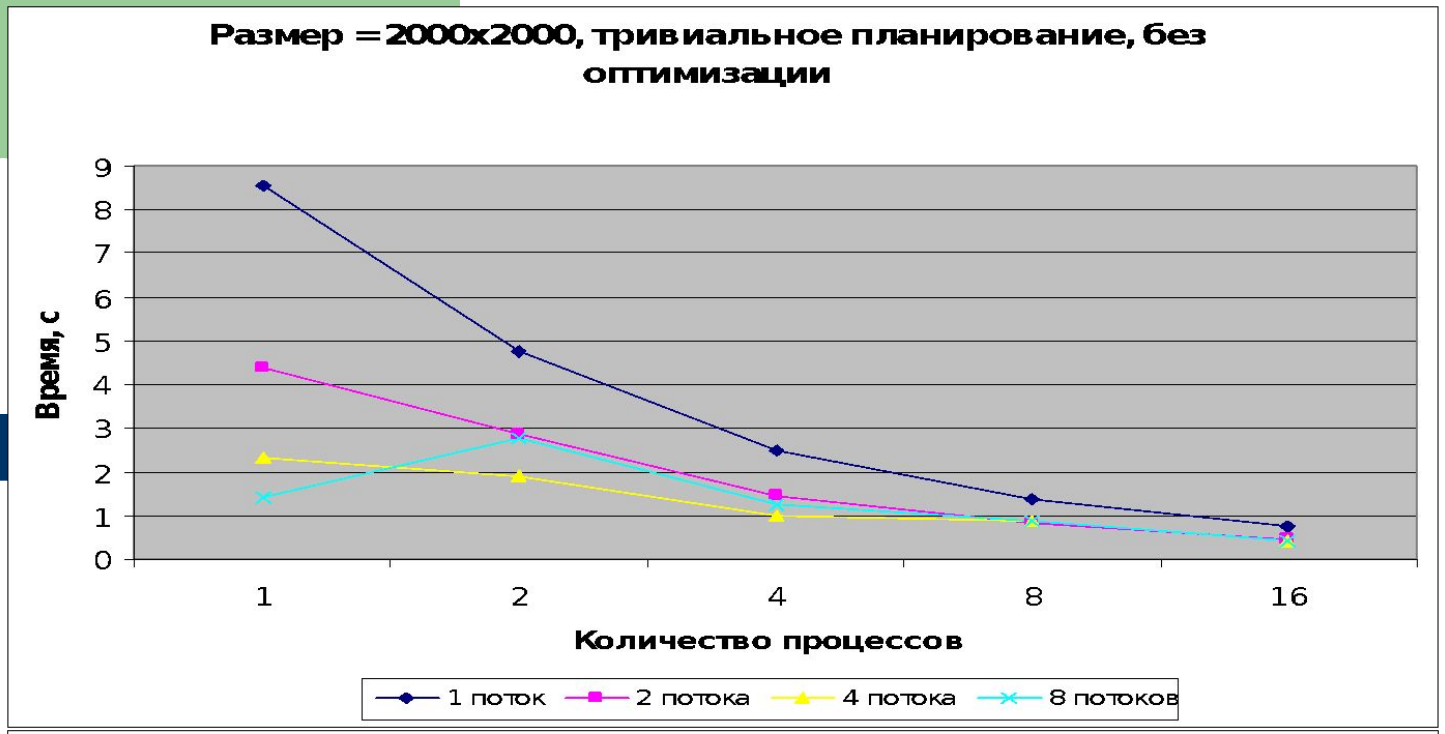
Исходные квадратные матрицы

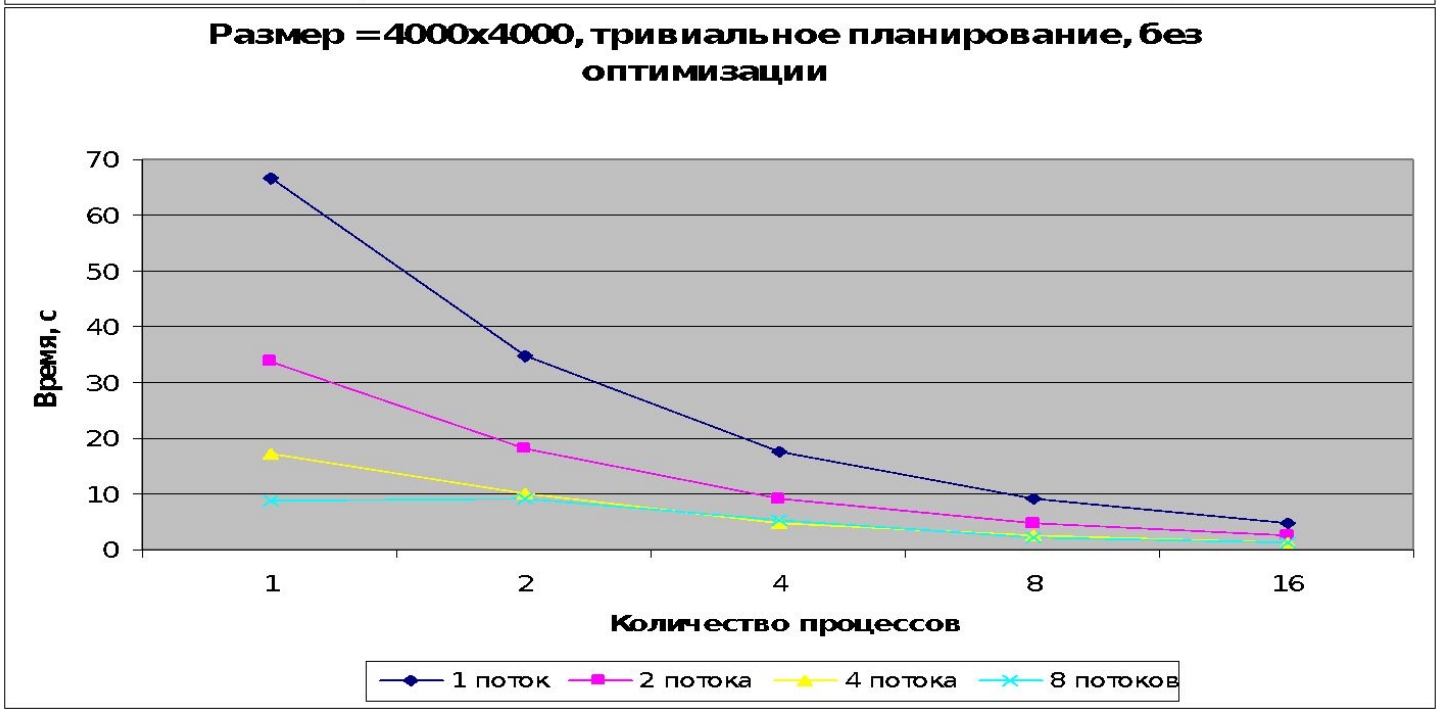
A  B 

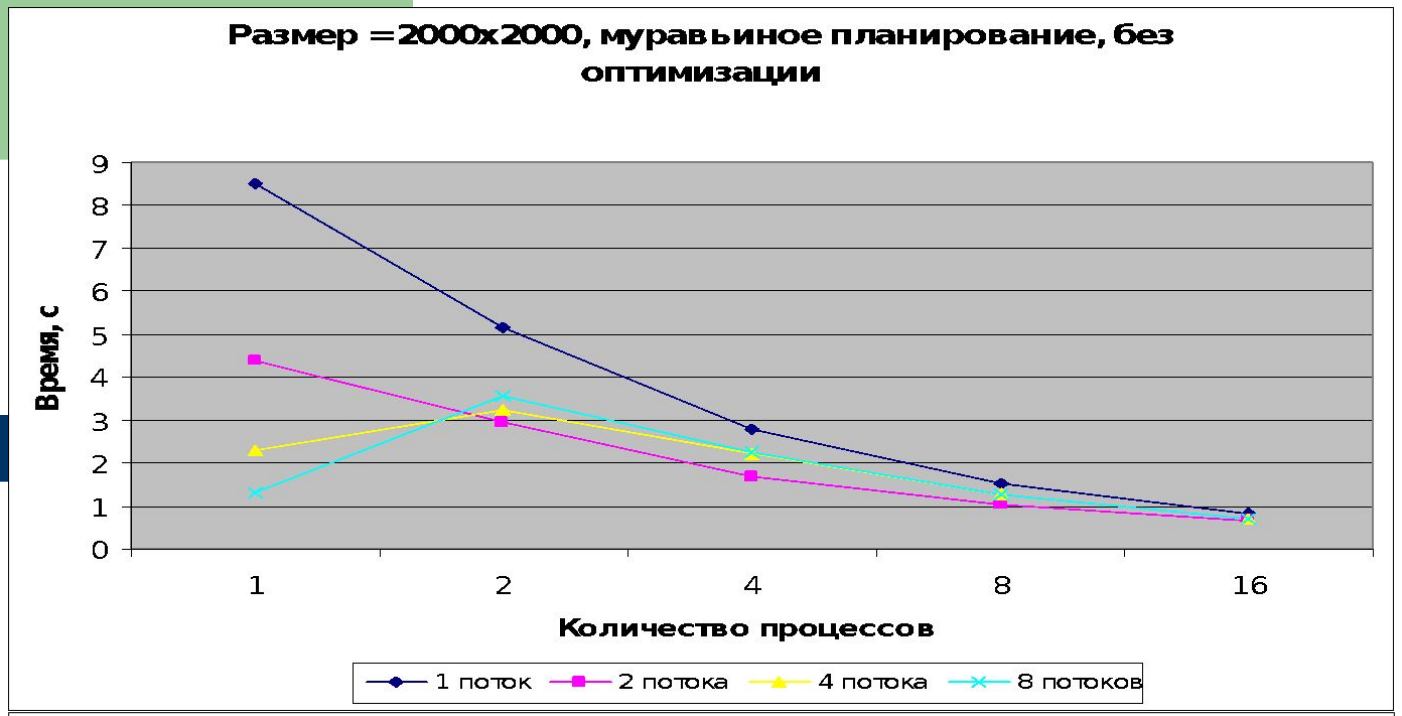
$$C_{i,j} = \sum A_{i,k} \cdot B_{k,j}$$

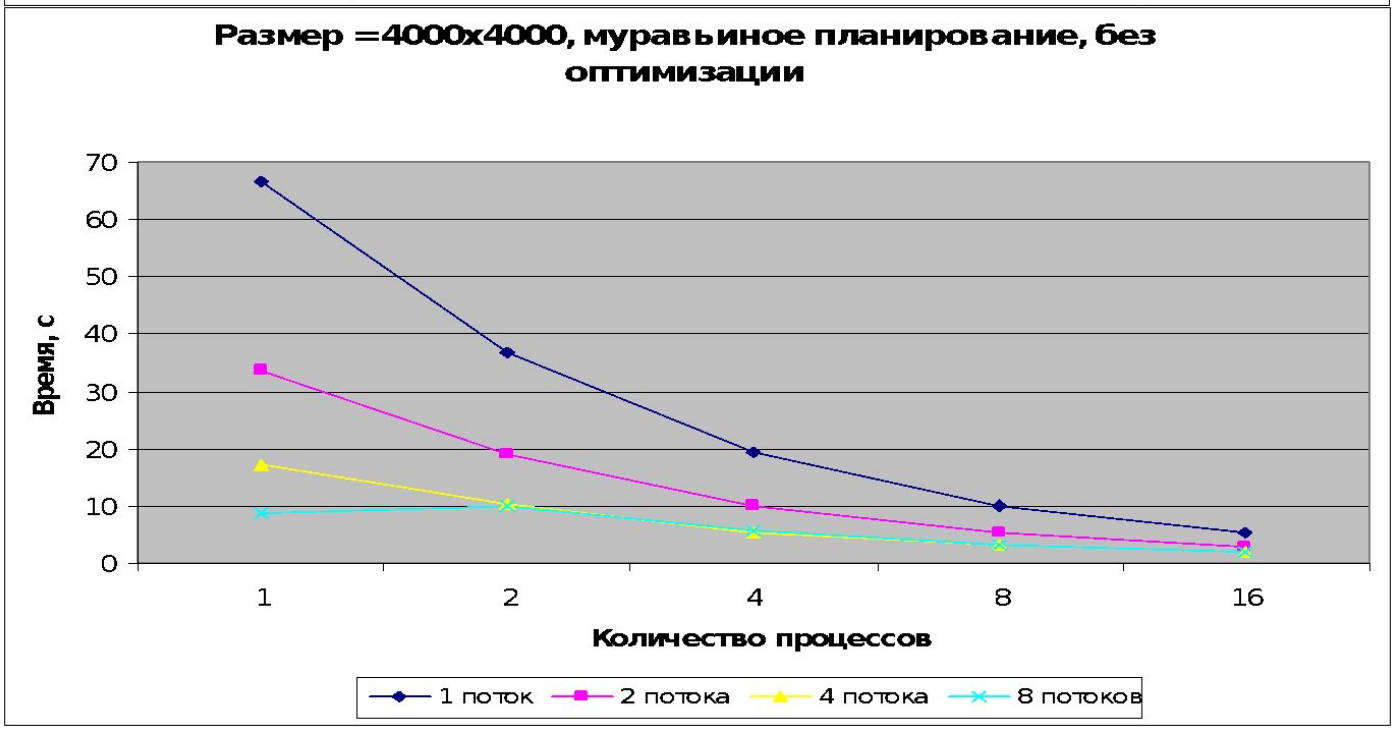
Схема вычисления

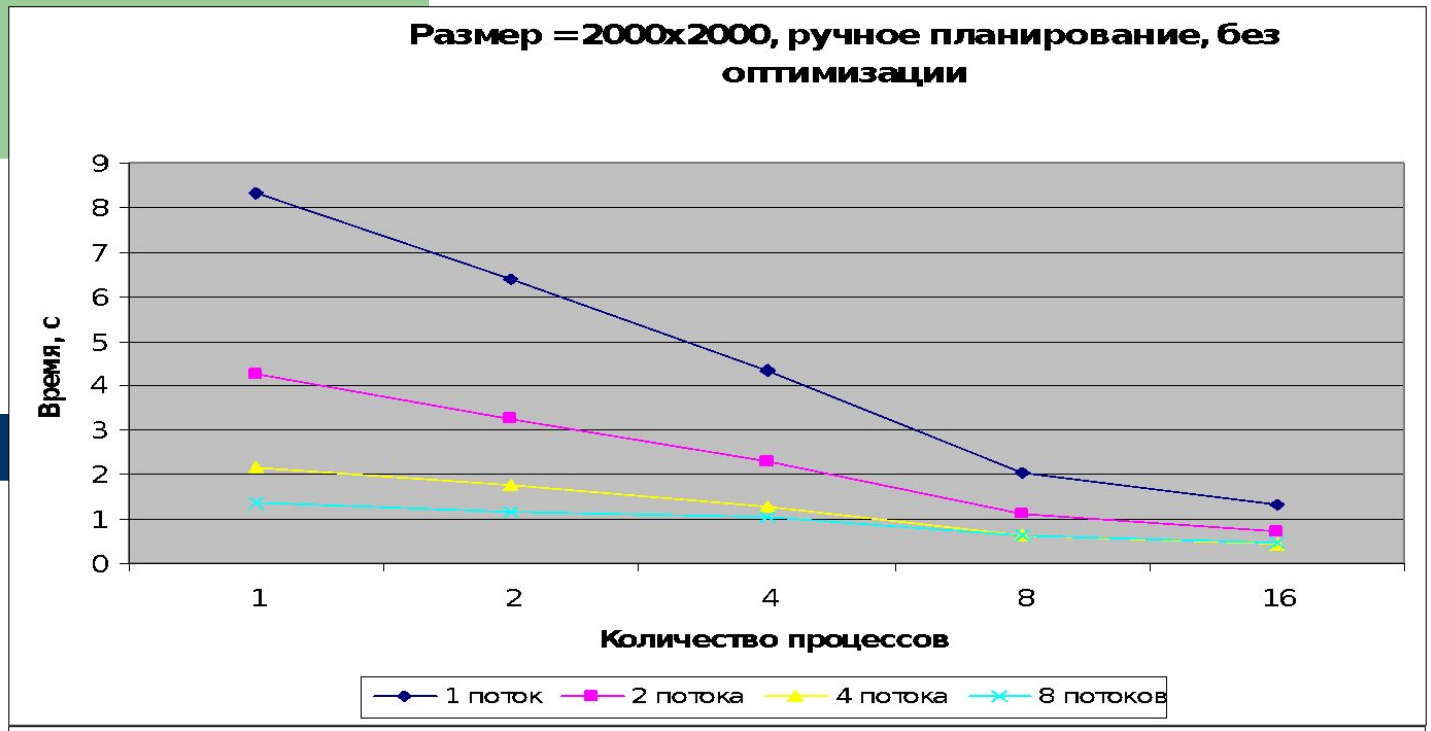


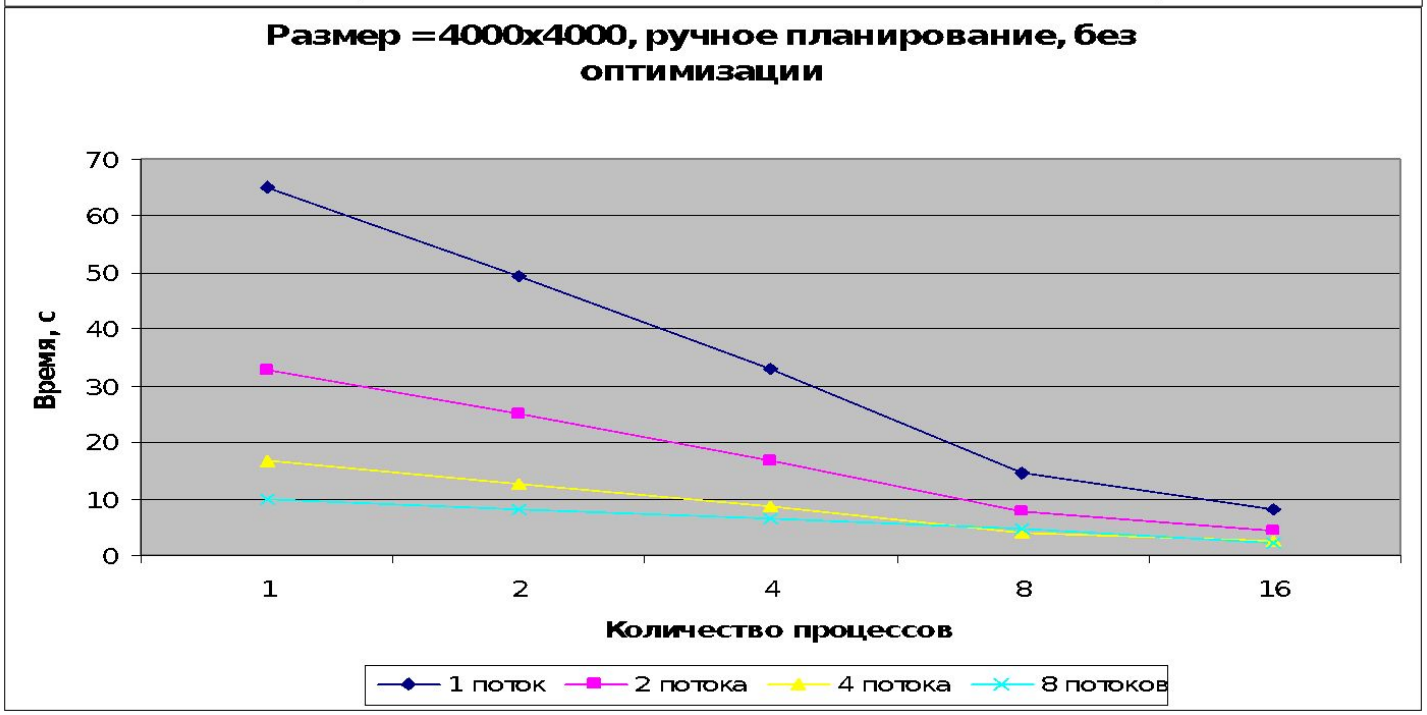

Размер =2000x2000, тривиальное планирование, без оптимизации
Время, с
0
1
2
3
4
5
6
7
8
9
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков

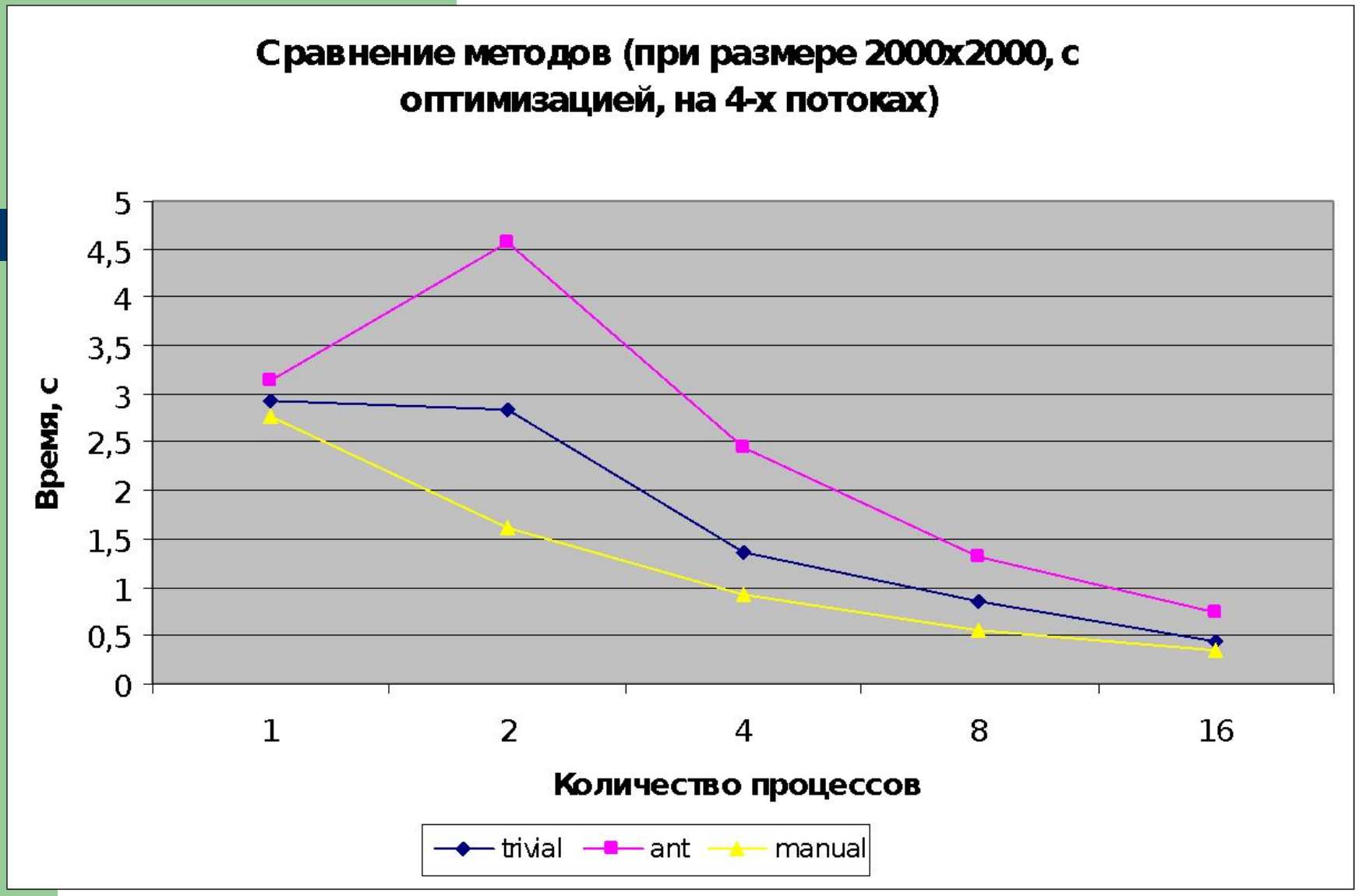

Размер =4000x4000, тривиальное планирование, без оптимизации
Время, с
0
10
20
30
40
50
60
70
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков

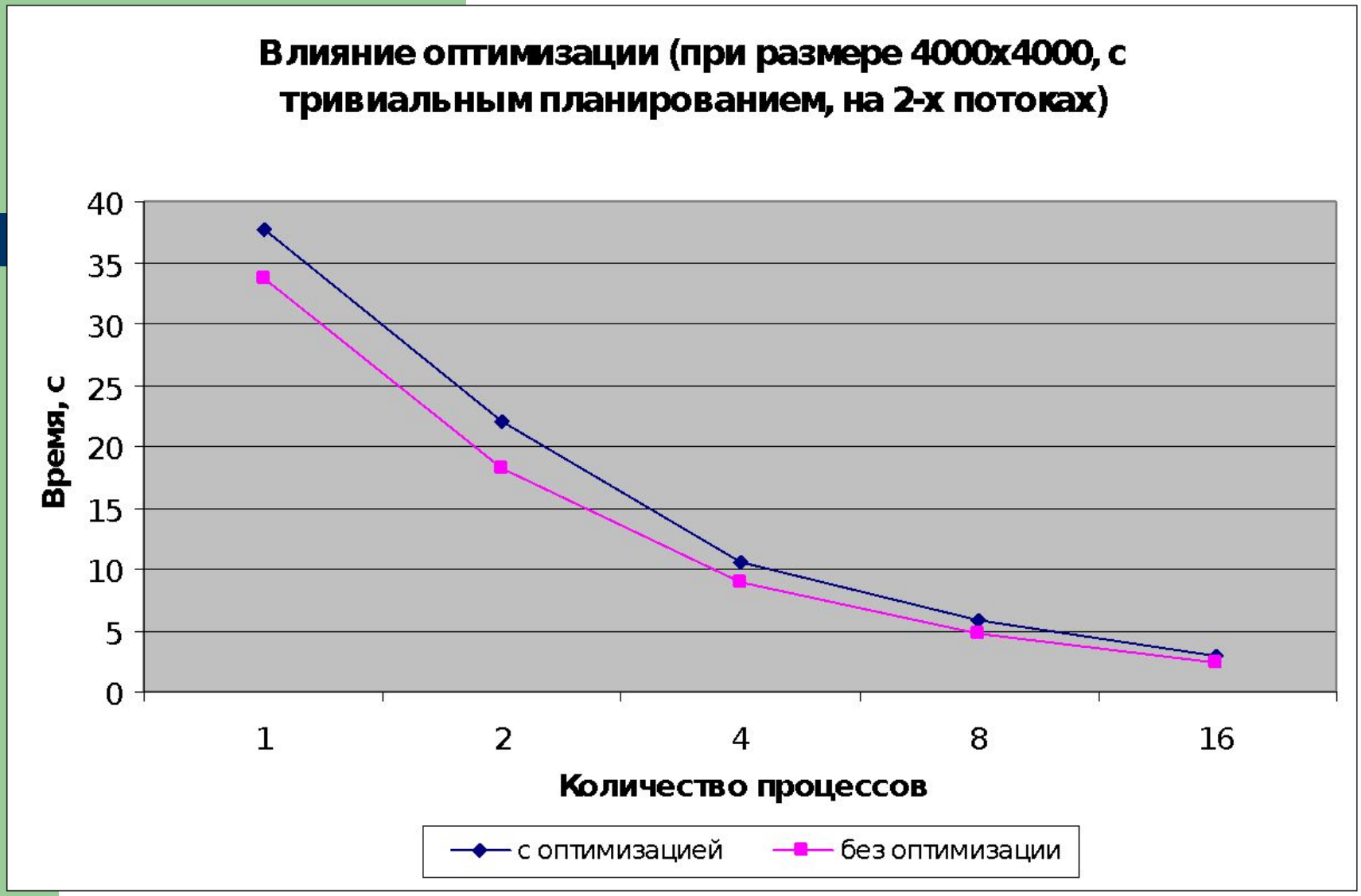

Размер =2000x2000, муравьиное планирование, без оптимизации
Время, с
0
1
2
3
4
5
6
7
8
9
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков


Размер =4000x4000, муравьиное планирование, без оптимизации
Время, с
0
10
20
30
40
50
60
70
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков


Размер =2000x2000, ручное планирование, без оптимизации
Время, с
0
1
2
3
4
5
6
7
8
9
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков


Размер =4000x4000, ручное планирование, без оптимизации
Время, с
0
10
20
30
40
50
60
70
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков

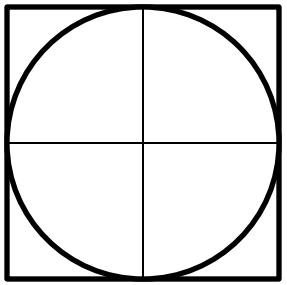

Сравнение методов (при размере 2000x2000, с оптимизацией, на 4-х потоках)
Время, с
5
4,5
4
3,5
3
2,5
2
1,5
1
0,5
0
1
2
4
8
16
Количество процессов
trivial
ant
manual

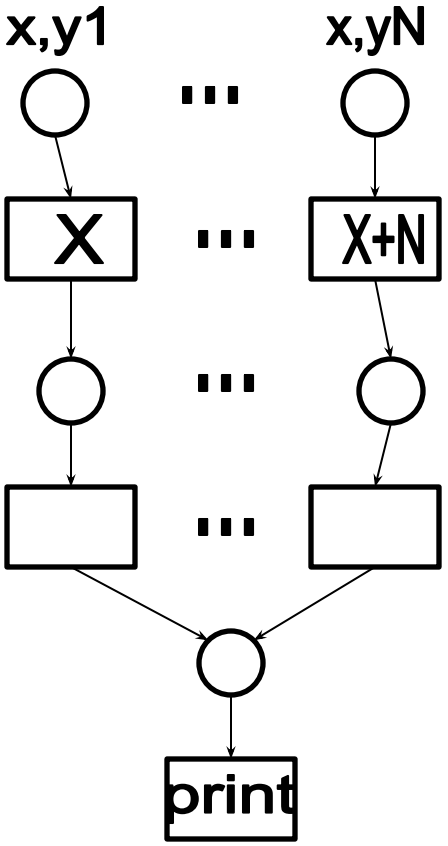

Влияние оптимизации (при размере 4000x4000, с тривиальным планированием, на 2-х потоках)
Время, с
40
35
30
25
20
15
10
5
0
1
2
4
8
16
Количество процессов
с оптимизацией
без оптимизации

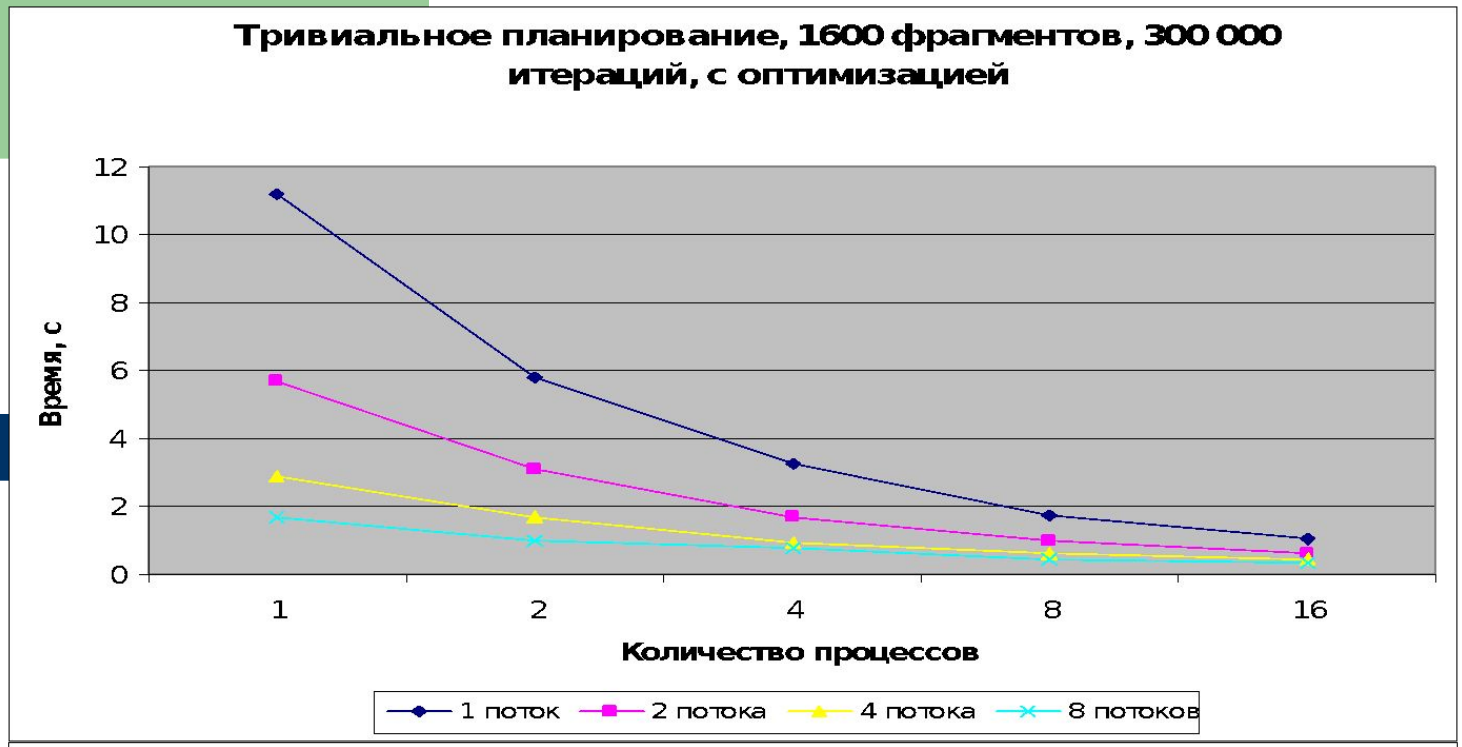
Нахождение числа Пи методом Монте-Карло

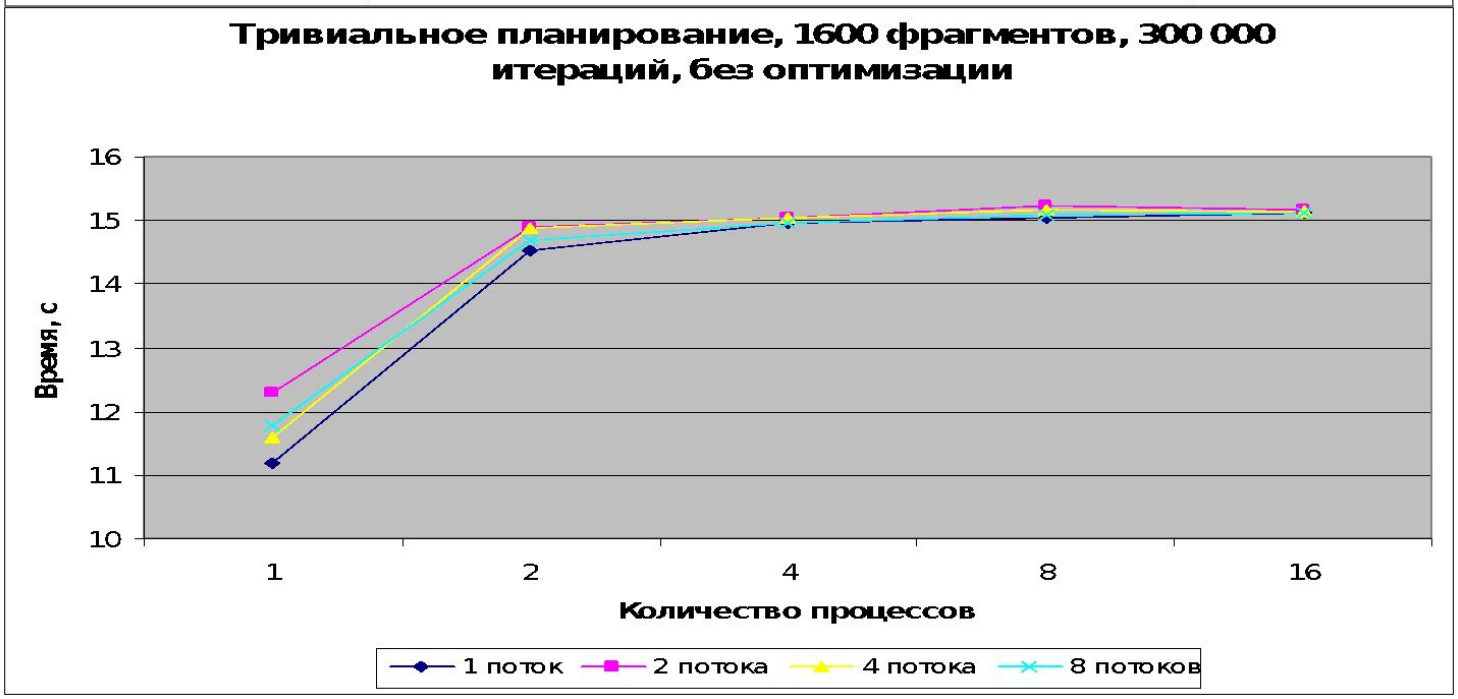
srand (x+i);
for (i=1..M)
{ x,y = rand (0..1);
r = sqrt (x^2 + y^2);
if (r<1) in++;
else out++;
Pi = (4*x*y*in)/ R^2 *(in+out);

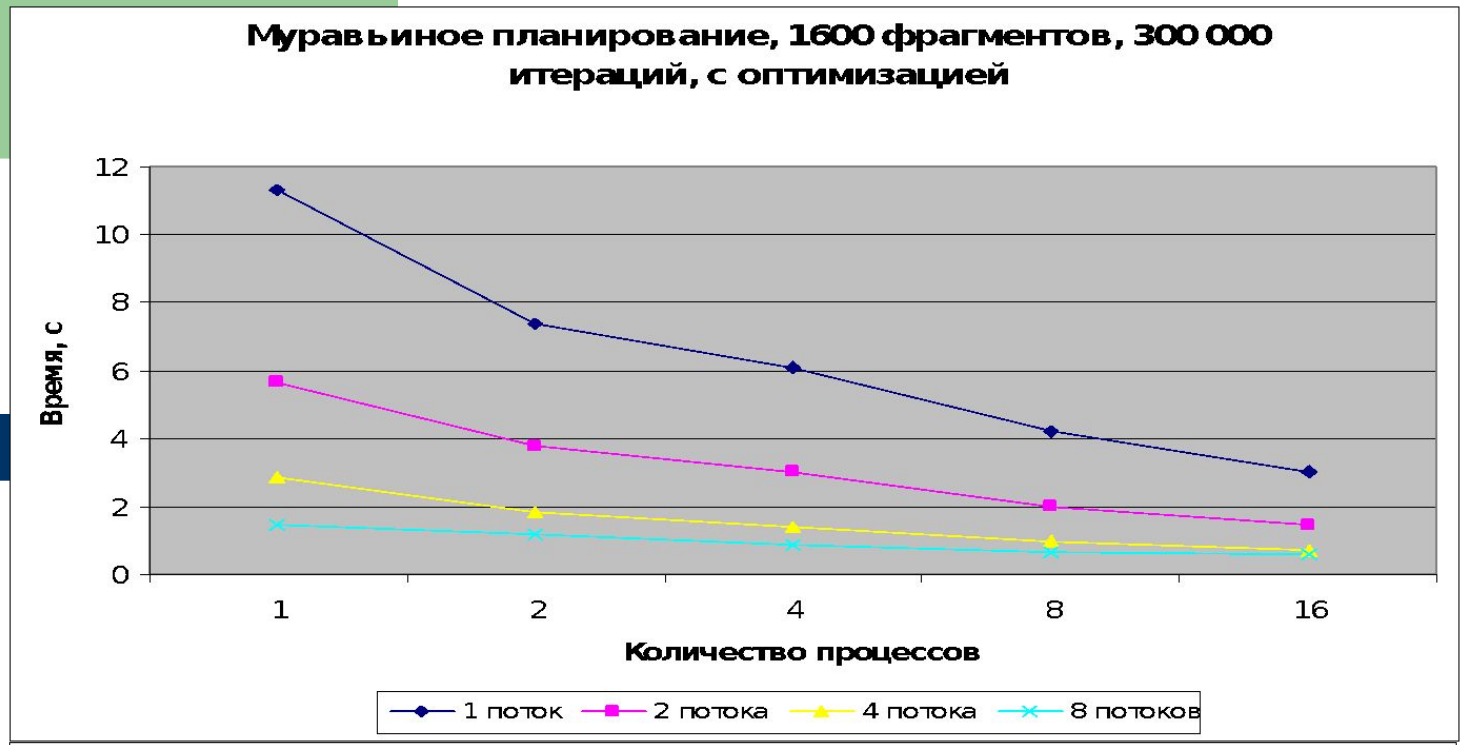
где M - количество итераций,
R=1 - радиус круга,
N – количество фрагментов

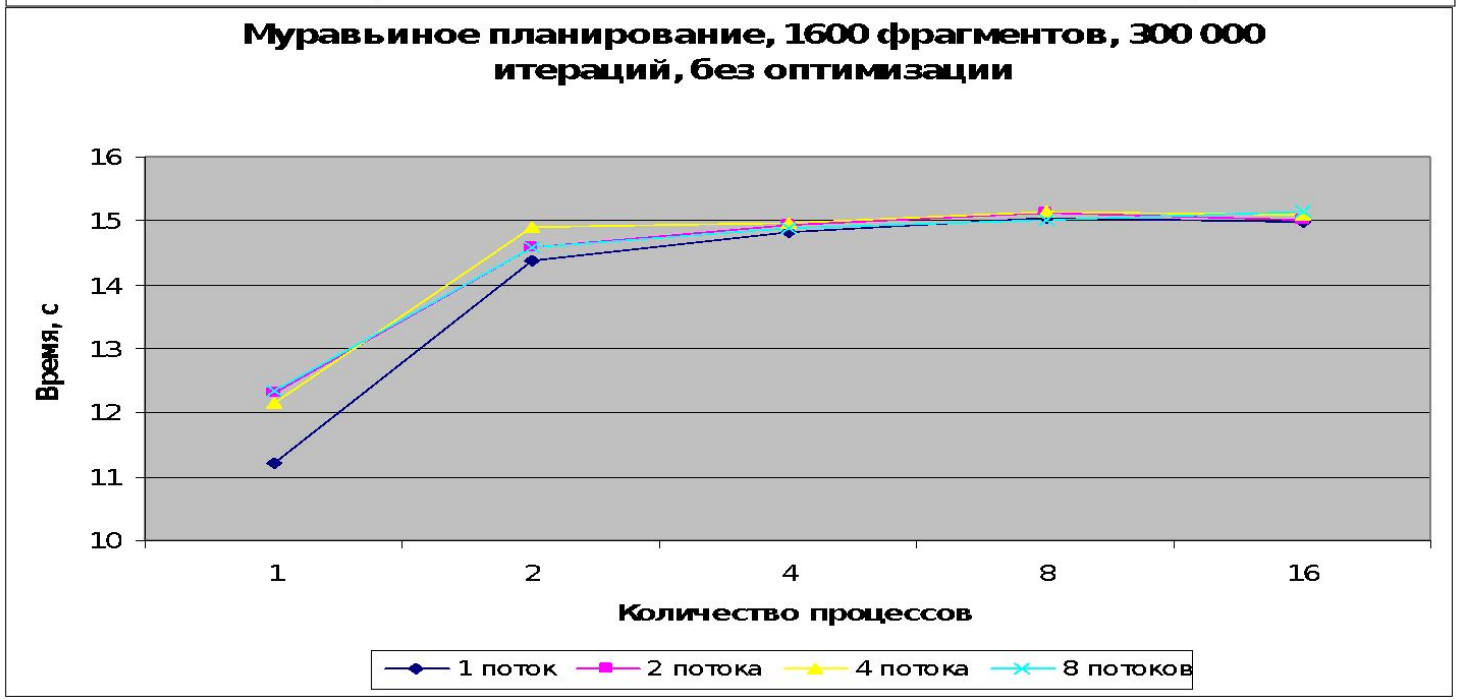


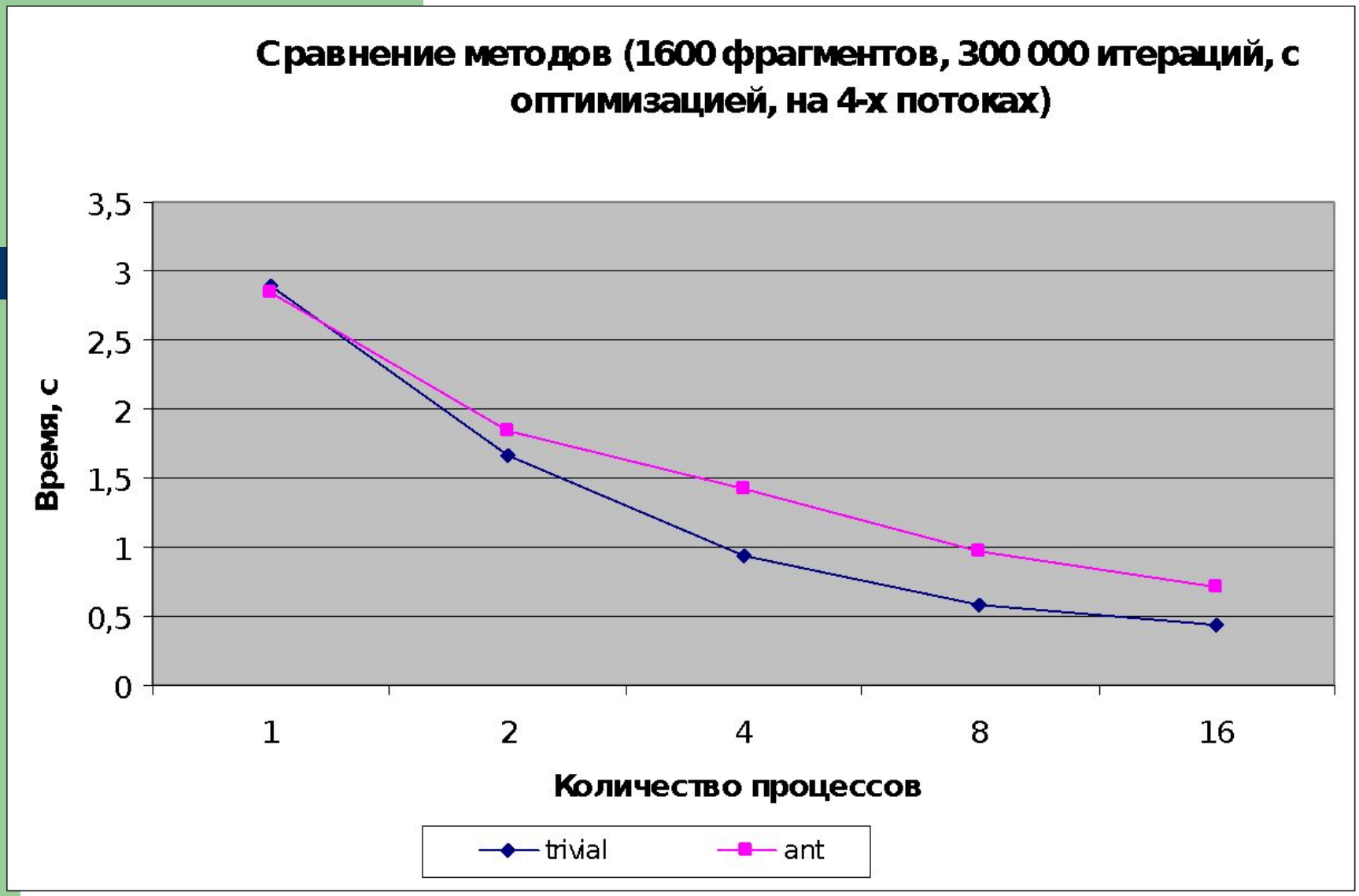



Тривиальное планирование, 1600 фрагментов, 300 000 итераций, с оптимизацией
Время, с
12
10
8
6
4
2
0
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков


Тривиальное планирование, 1600 фрагментов, 300 000 итераций, без оптимизации
Время, с
16
15
14
13
12
11
10
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков


Муравьиное планирование, 1600 фрагментов, 300 000 итераций, с оптимизацией
Время, с
0
2
4
6
8
10
12
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков


Муравьиное планирование, 1600 фрагментов, 300 000 итераций, без оптимизации
Время, с
10
11
12
13
14
15
16
1
2
4
8
16
Количество процессов
1 поток
2 потока
4 потока
8 потоков


Сравнение методов (1600 фрагментов, 300 000 итераций, с оптимизацией, на 4-х потоках)
Время, с
3,5
3
2,5
2
1,5
1
0,5
0
1
2
4
8
16
Количество процессов
trivial
ant

Результаты

- Мы познакомились с системой фрагментированного программирования, с технологией MPI, а также с системой POSIX-тредов;
- усовершенствовали ИС до гибридного варианта;
- разработан и реализован алгоритм оптимизации;
- протестировали ИС на задачах перемножения плотных матриц и нахождения числа Пи методом Монте-Карло, полученные в ходе тестирования результаты свидетельствуют о высокой эффективности системы.