

# 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

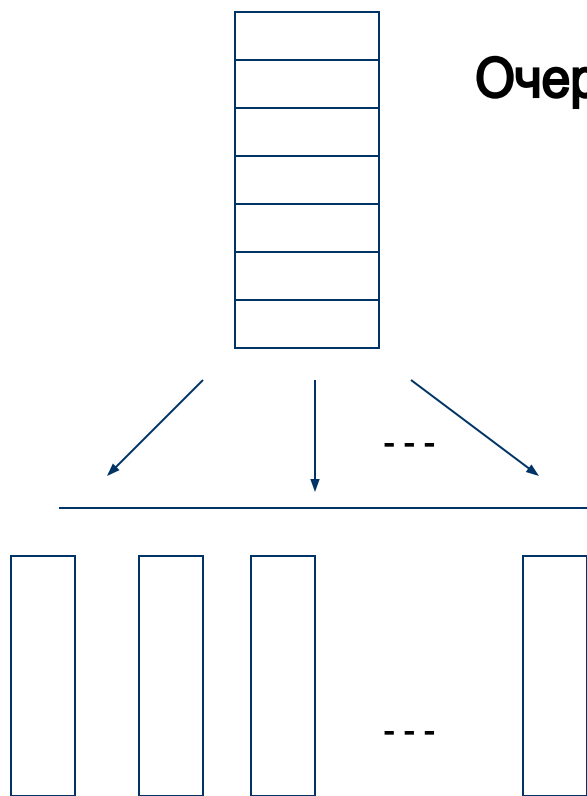
---

Распараллеливание исполнительской 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

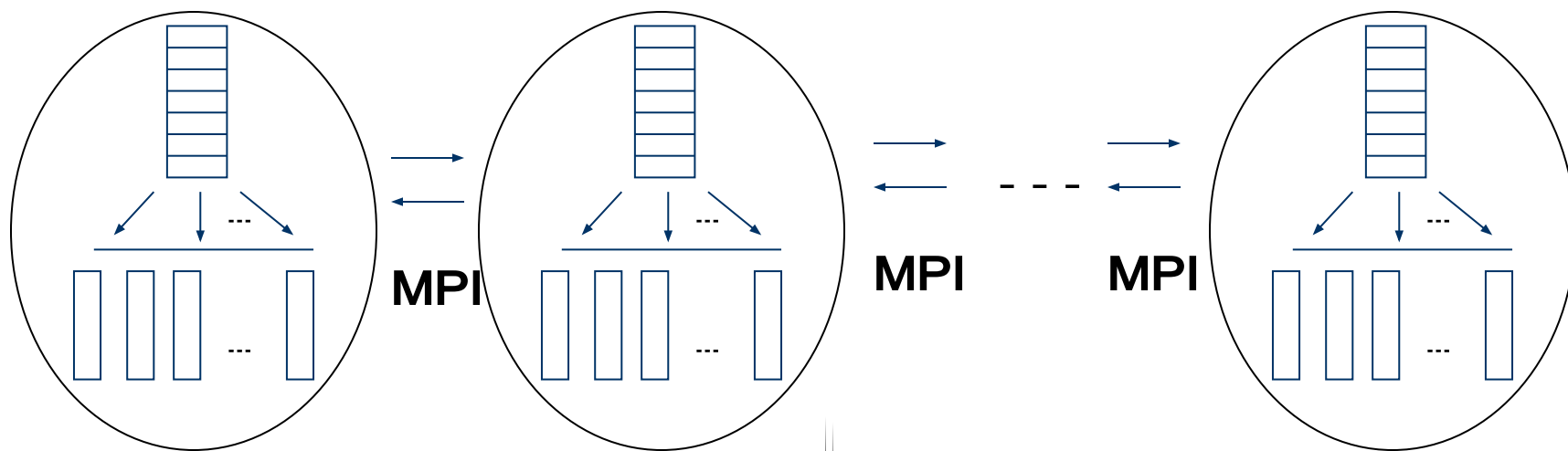


Очередь фрагментов вычислений

Потоки исполняются параллельно на одном и том же участке памяти

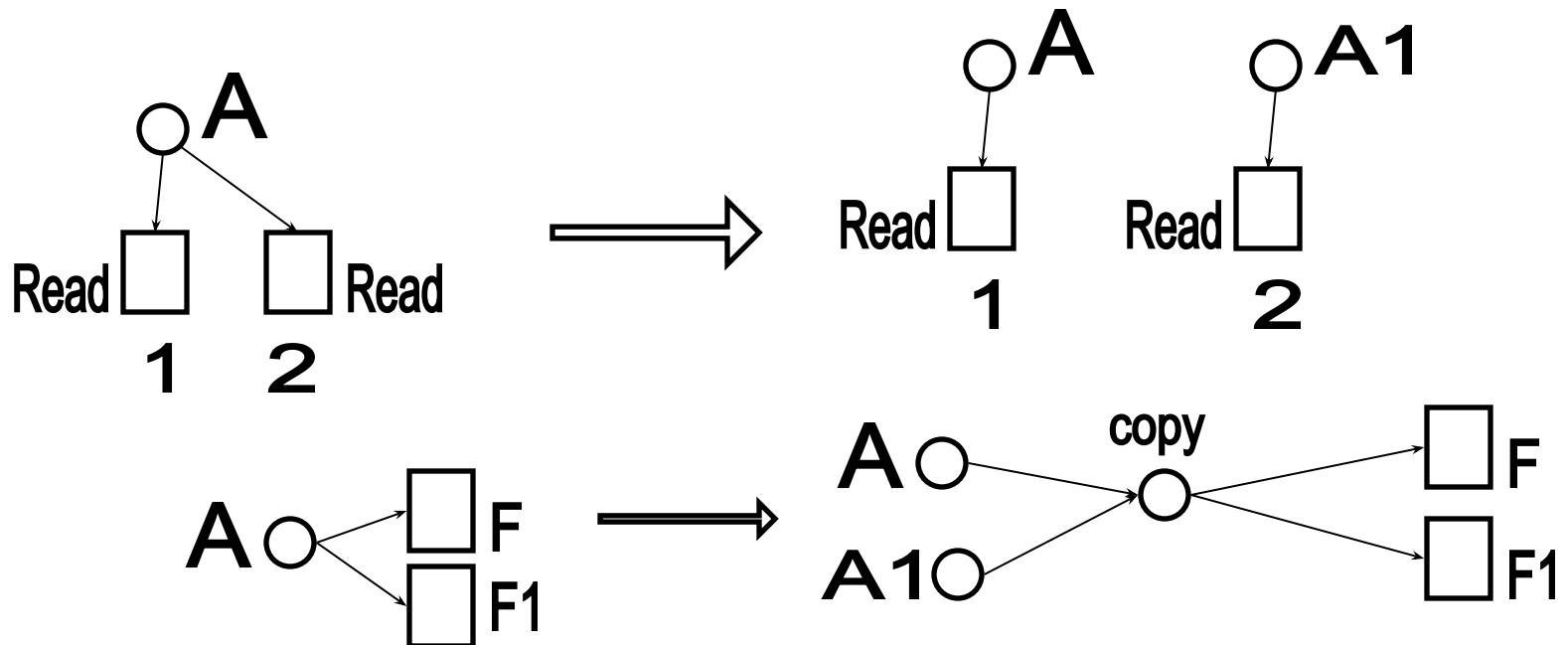
Доступ на запись данных на общих участках памяти контролируется посредством мьютексов

# Гибридная схема



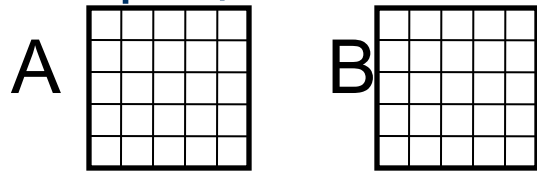
# Оптимизация плана исполнения

- 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.



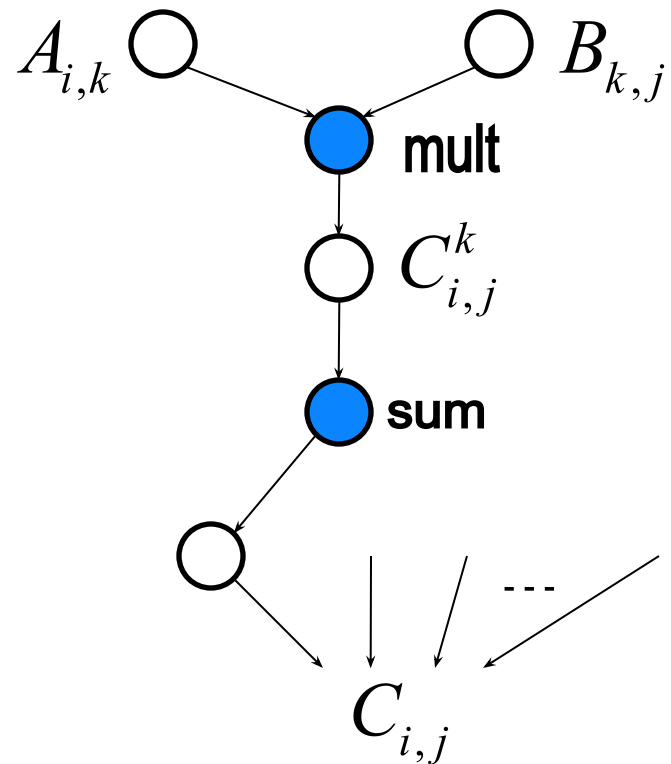
# Перемножение плотных матриц

Исходные квадратные  
матрицы

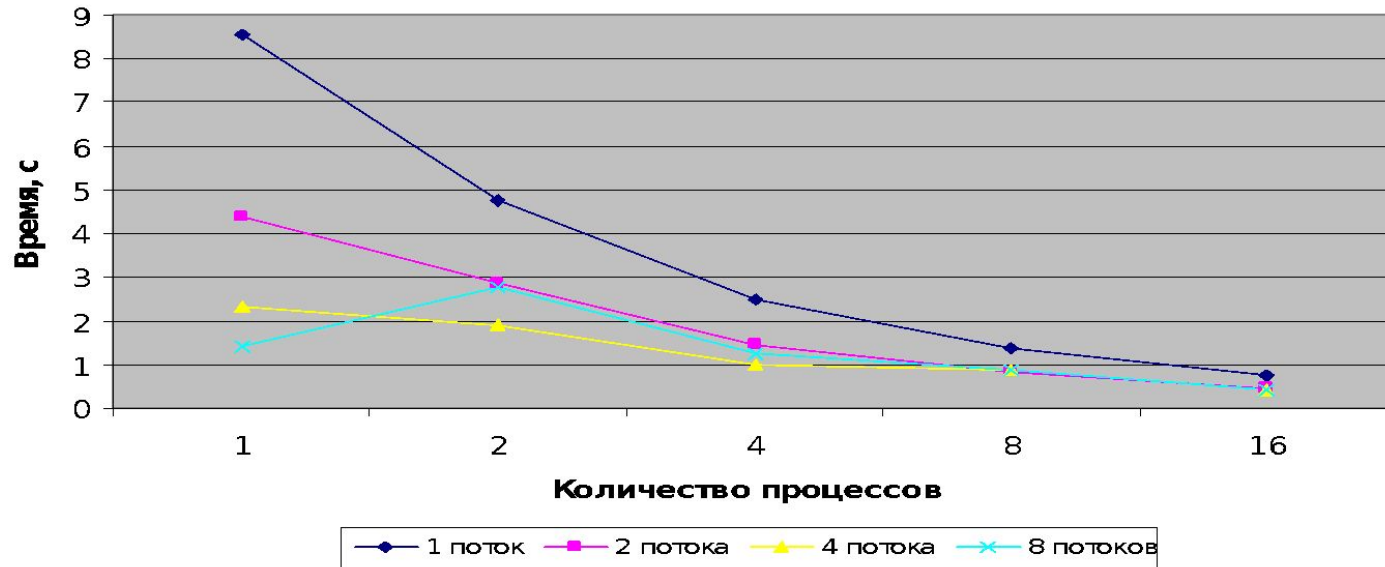


$$C_{i,j} = \sum A_{i,k} \cdot B_{k,j}$$

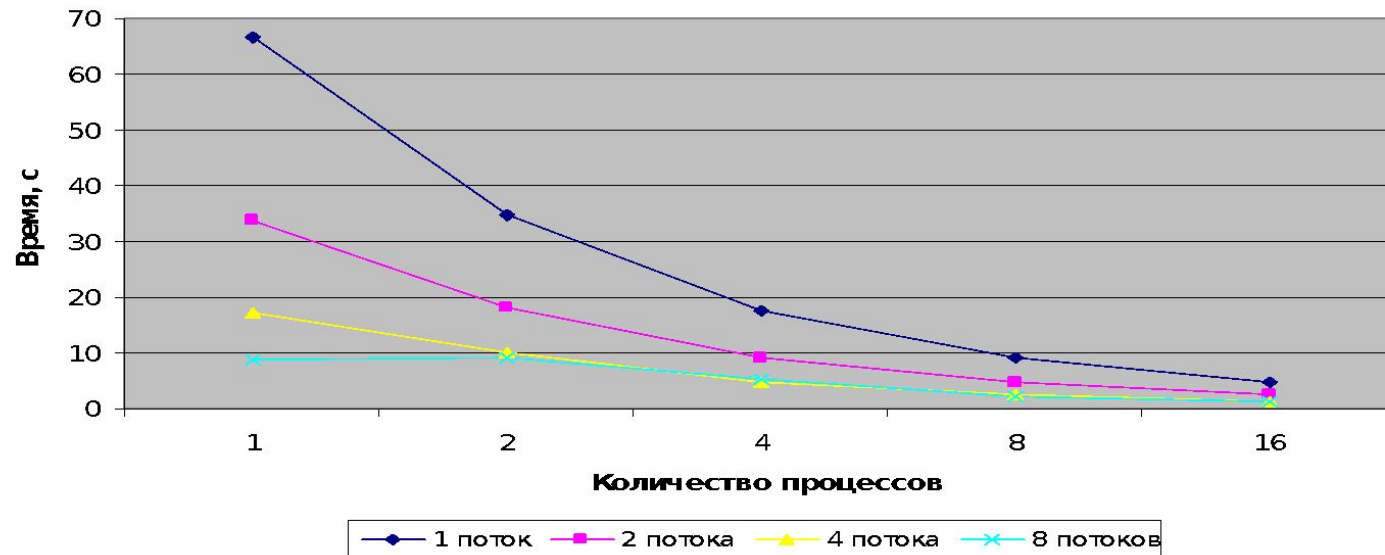
Схема вычисления



**Размер = 2000x2000, тривиальное планирование, без оптимизации**

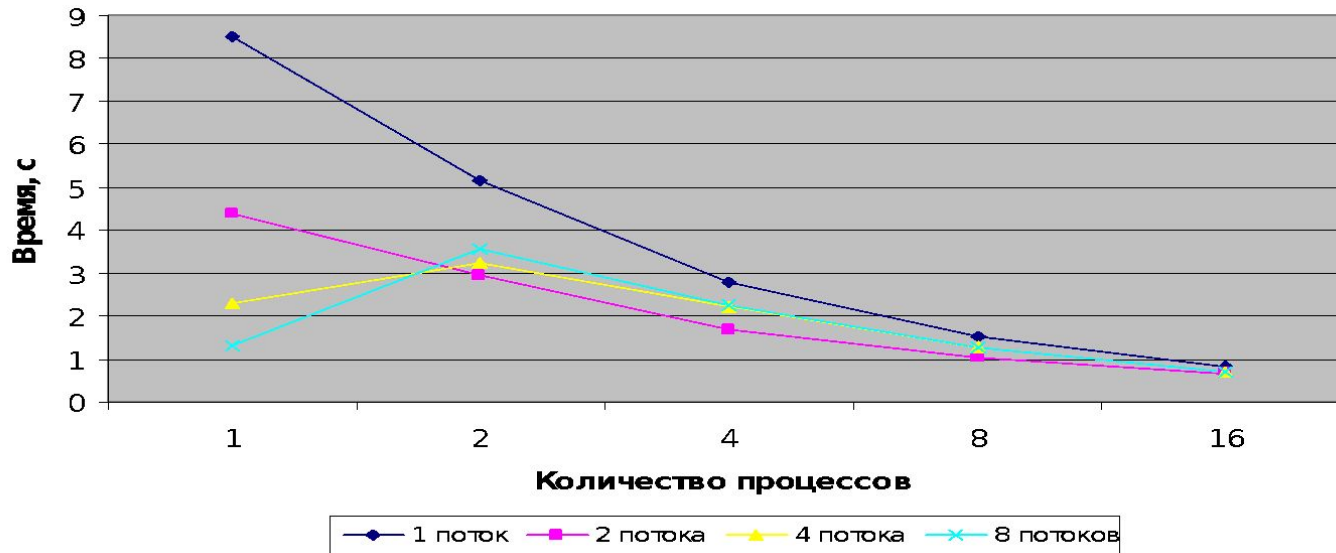


**Размер = 4000x4000, тривиальное планирование, без оптимизации**

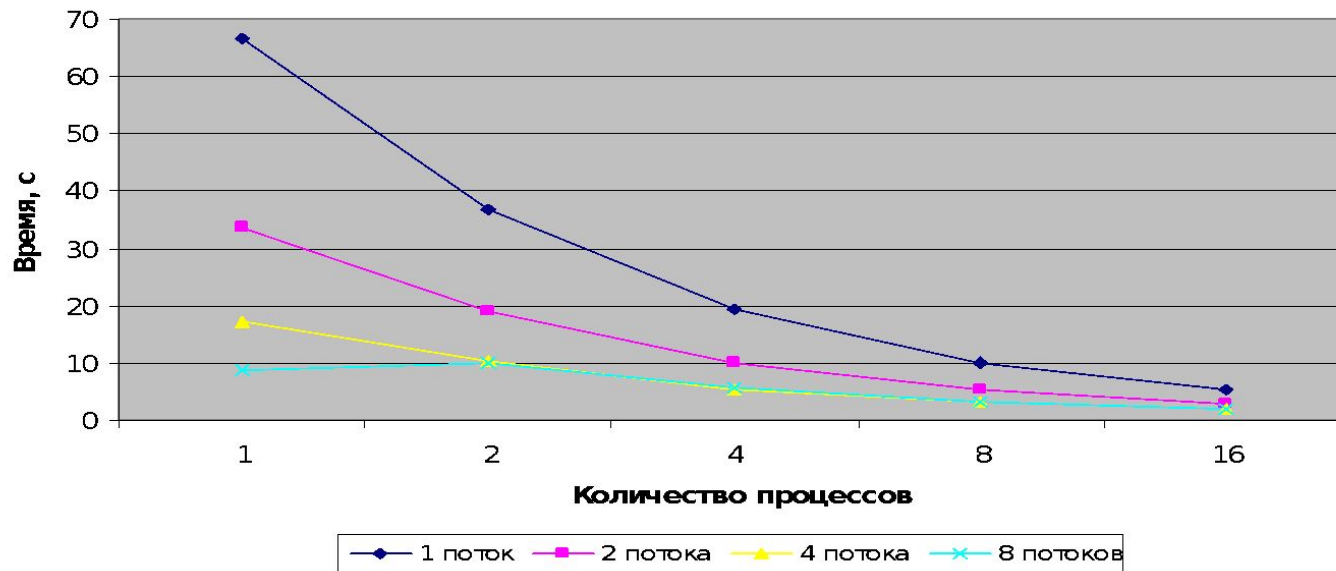




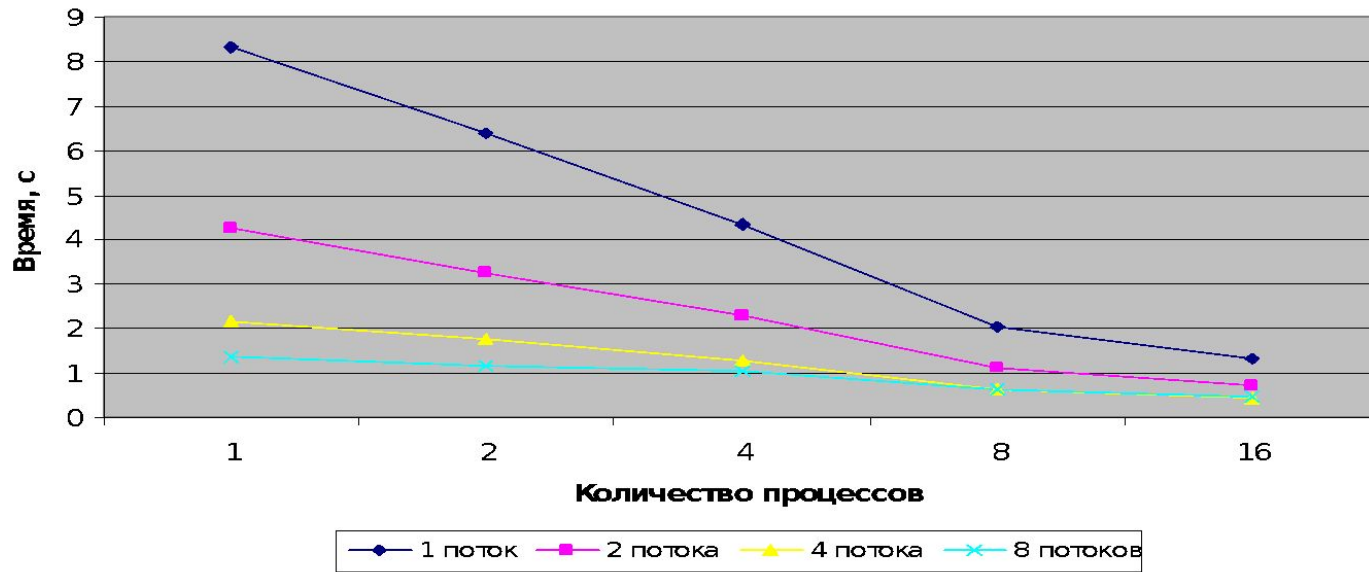
**Размер = 2000x2000, муравьиное планирование, без оптимизации**



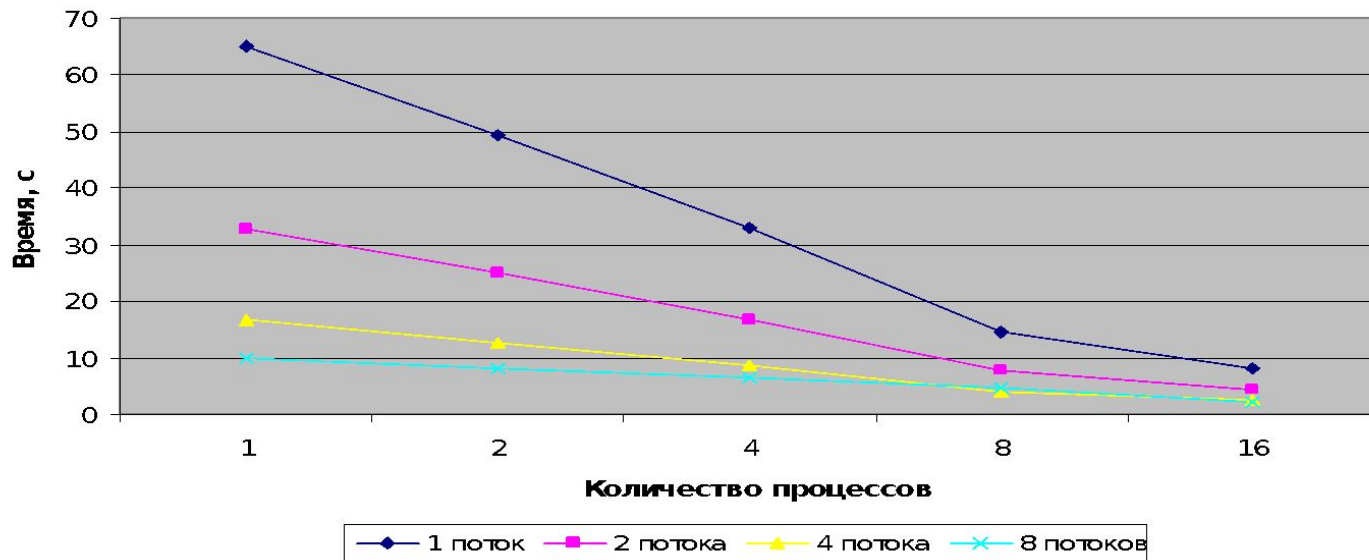
**Размер = 4000x4000, муравьиное планирование, без оптимизации**



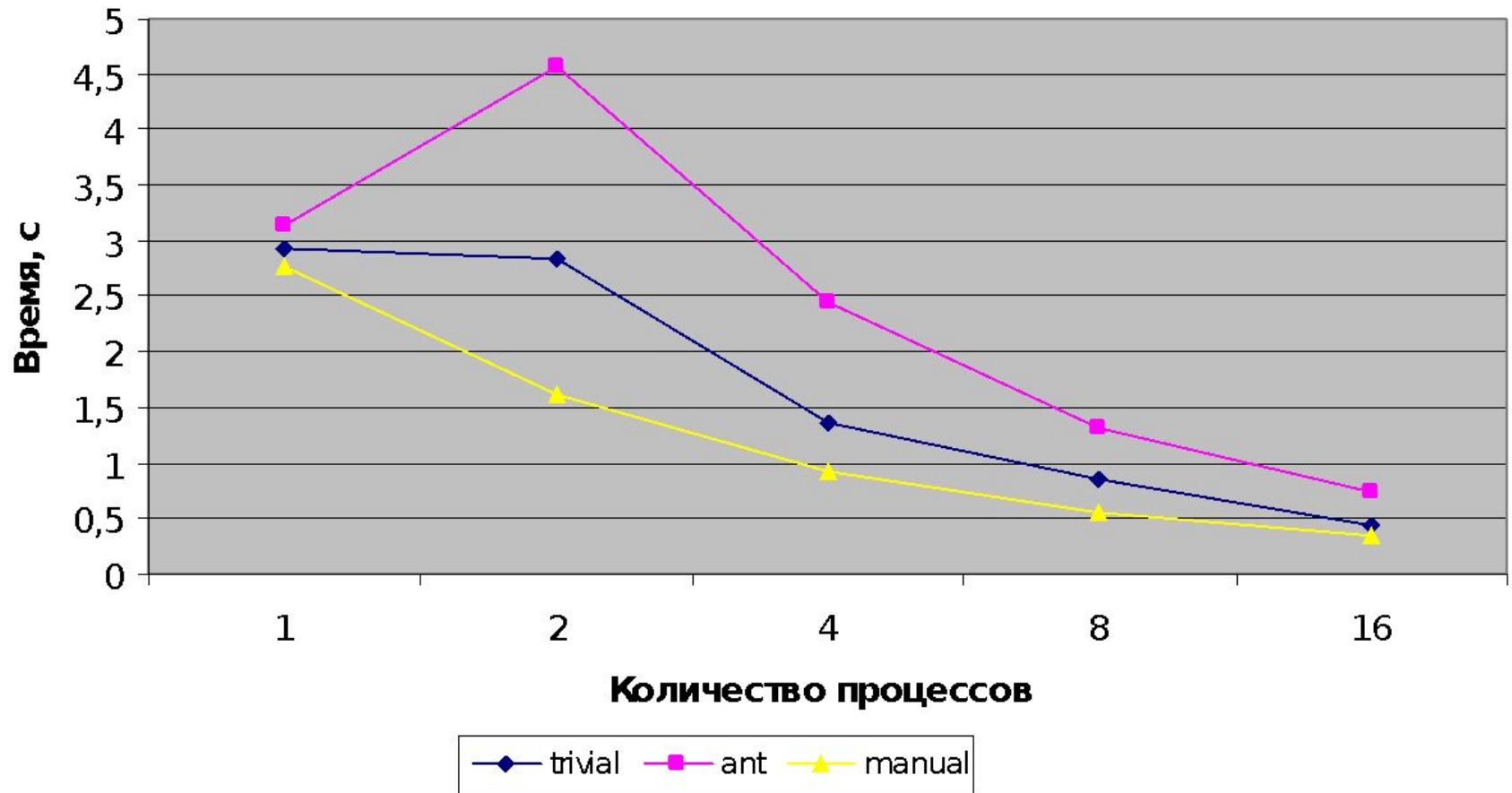
**Размер = 2000x2000, ручное планирование, без оптимизации**



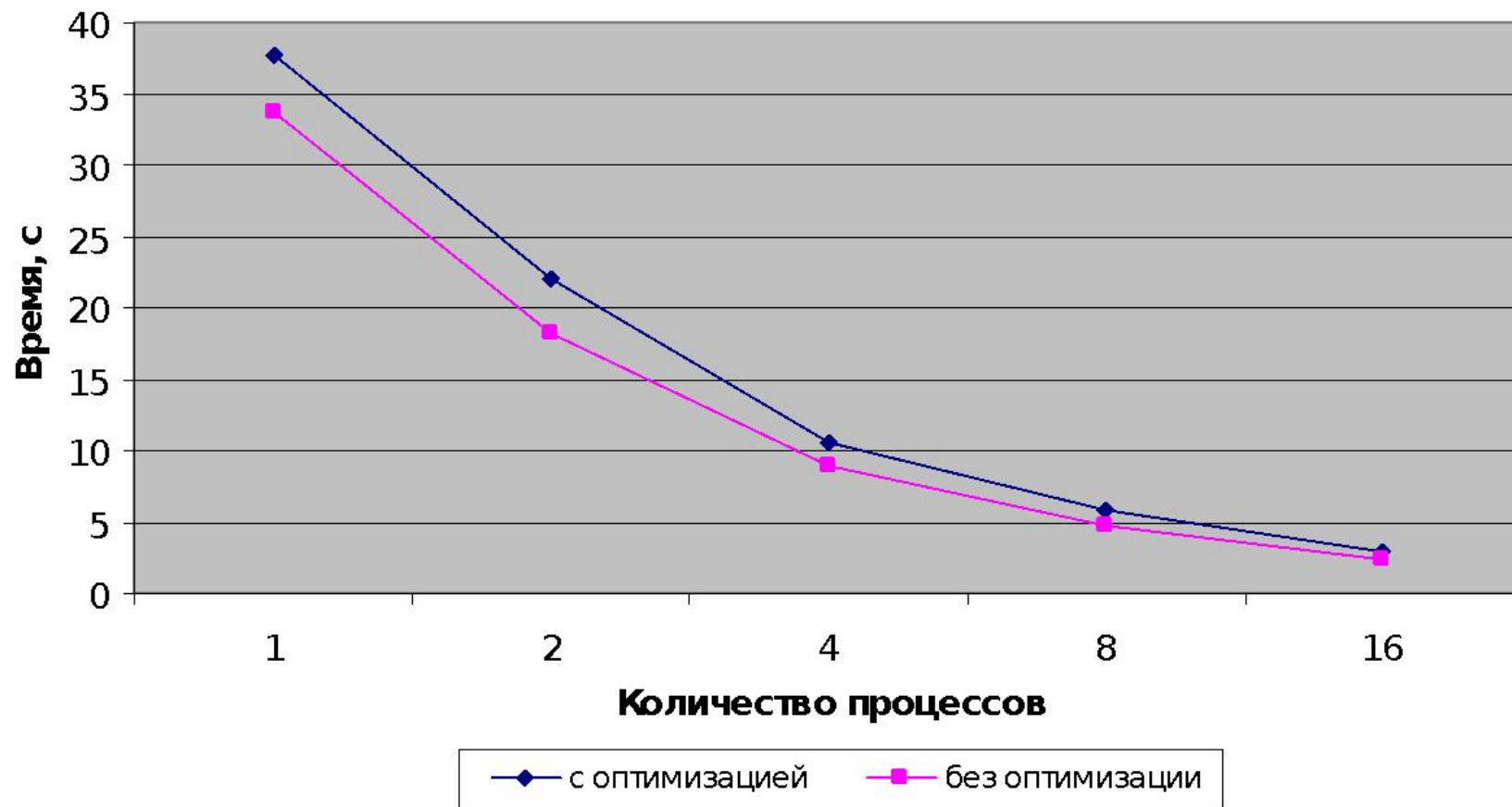
**Размер = 4000x4000, ручное планирование, без оптимизации**



### Сравнение методов (при размере 2000x2000, с оптимизацией, на 4-х потоках)



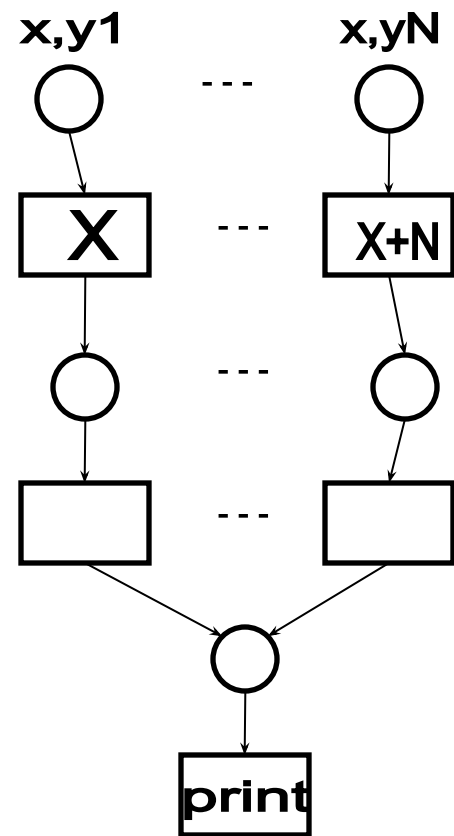
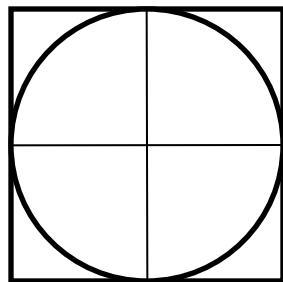
## Влияние оптимизации (при размере 4000x4000, с тривиальным планированием, на 2-х потоках)



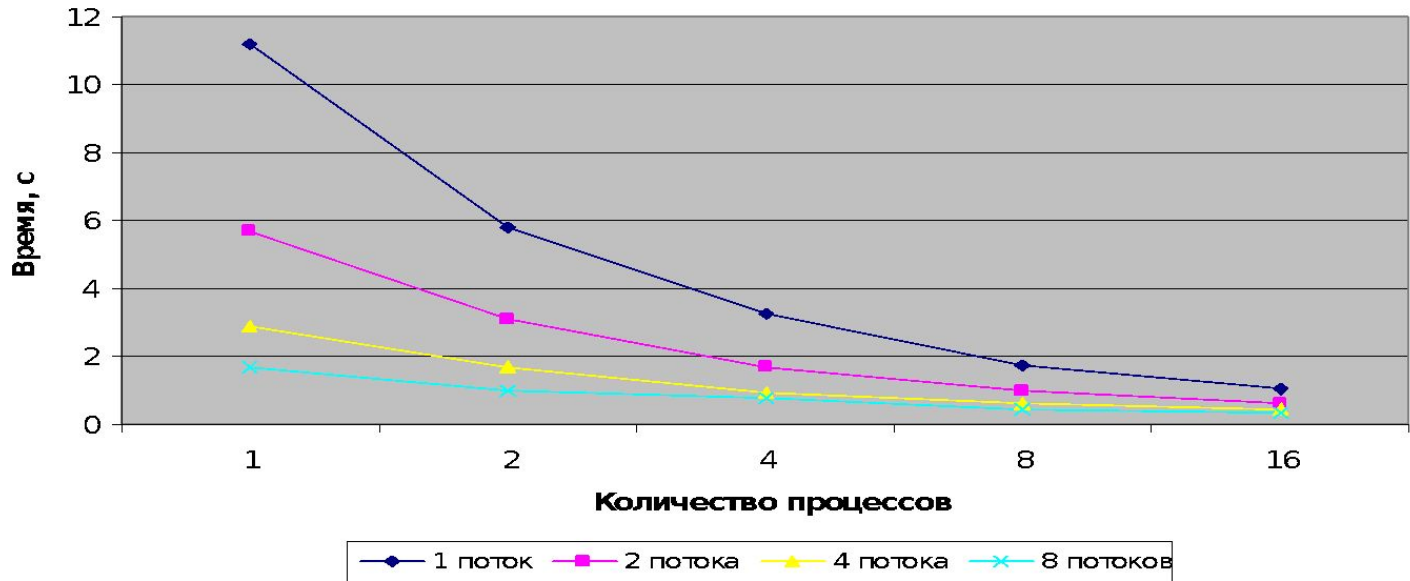
# Нахождение числа Пи методом Монте-Карло

```
srand (x+i);  
for (i=1..M)  
{ x,y = rand (0..1);  
  r = sqrt (x2 + y2);  
  if (r<1) in++;  
  else out++;  
  Pi = (4*x*y*in)/ R2 *(in+out);
```

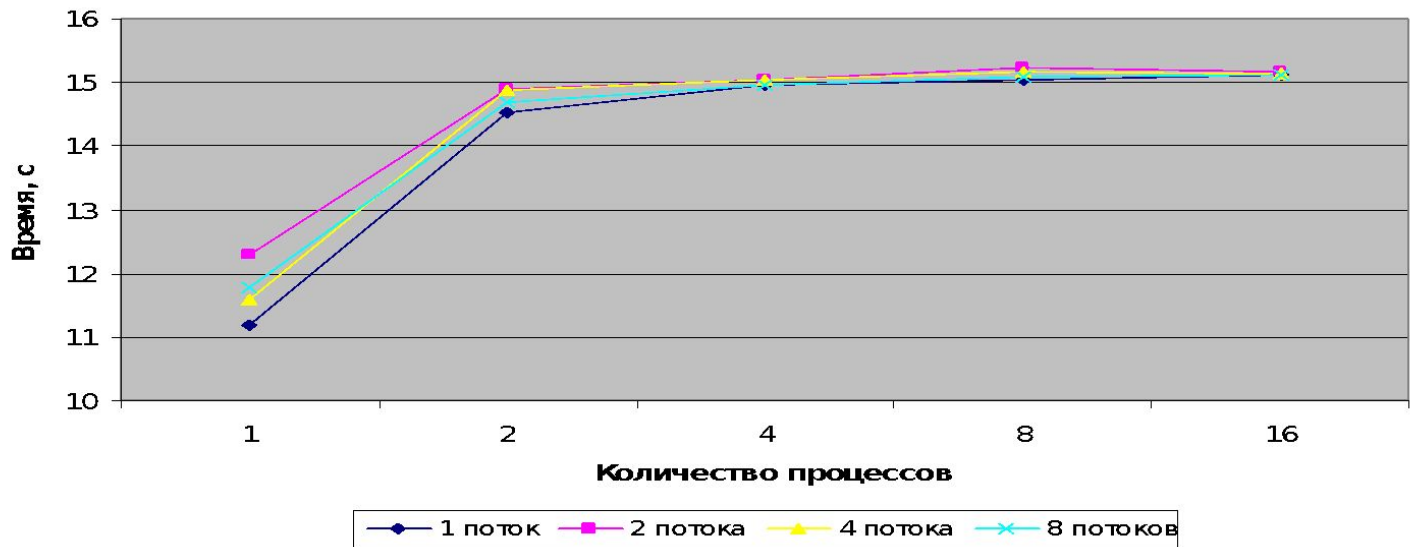
где M - количество итераций,  
R=1 - радиус круга,  
N – количество фрагментов



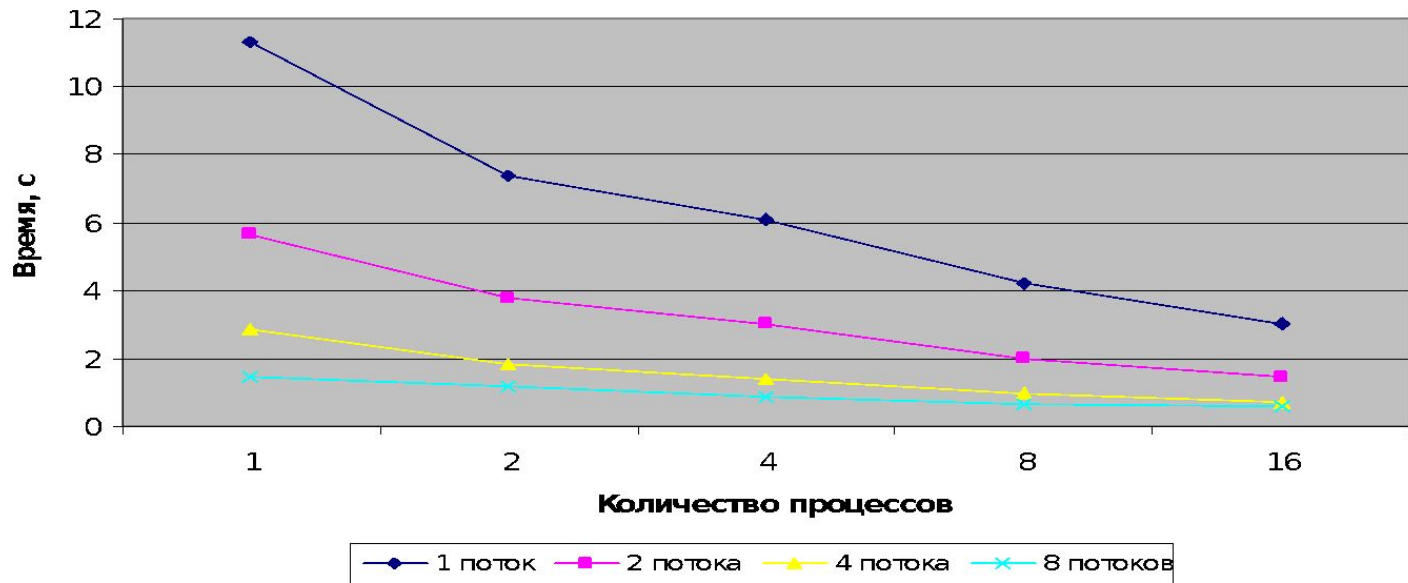
### Тривиальное планирование, 1600 фрагментов, 300 000 итераций, с оптимизацией



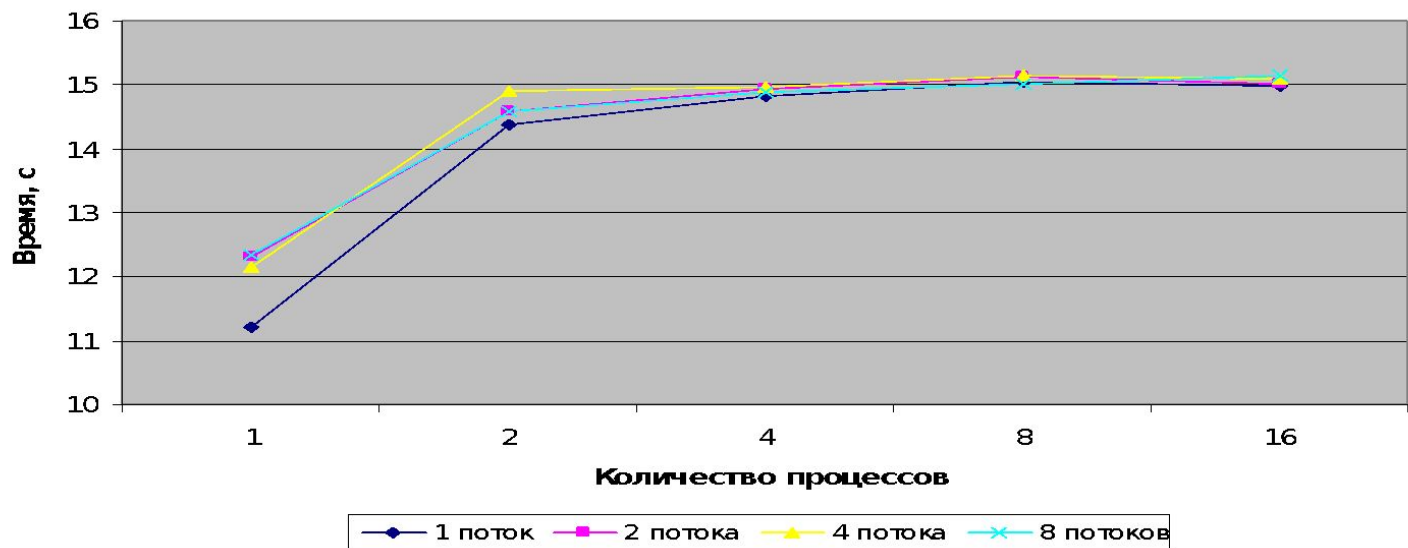
### Тривиальное планирование, 1600 фрагментов, 300 000 итераций, без оптимизации



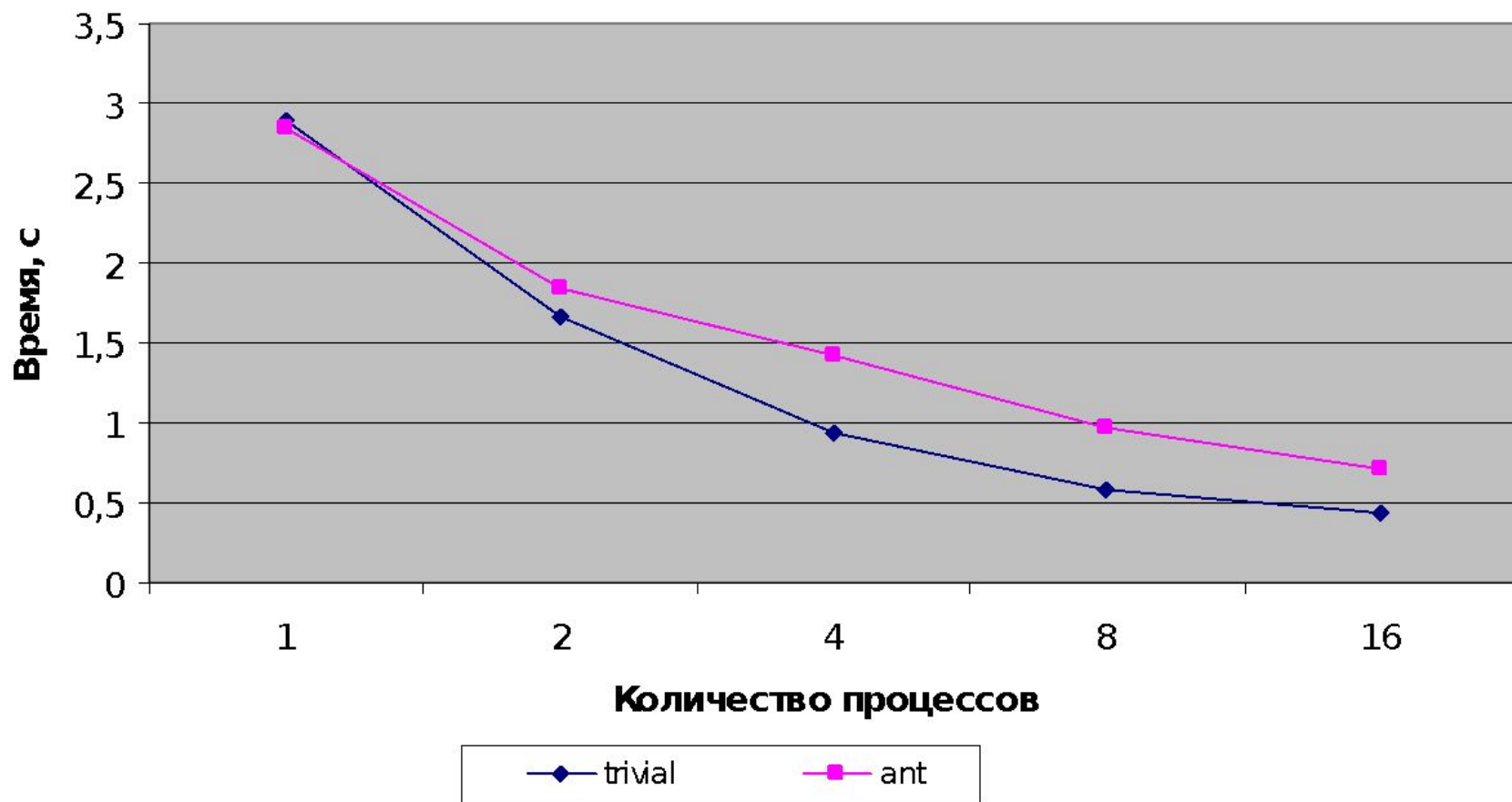
### Муравьиное планирование, 1600 фрагментов, 300 000 итераций, с оптимизацией



### Муравьиное планирование, 1600 фрагментов, 300 000 итераций, без оптимизации



### Сравнение методов (1600 фрагментов, 300 000 итераций, с оптимизацией, на 4-х потоках)





# 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.