Лабораторная работа 4
Создание многопоточных OpenMP-программ

Курносов Михаил Георгиевич

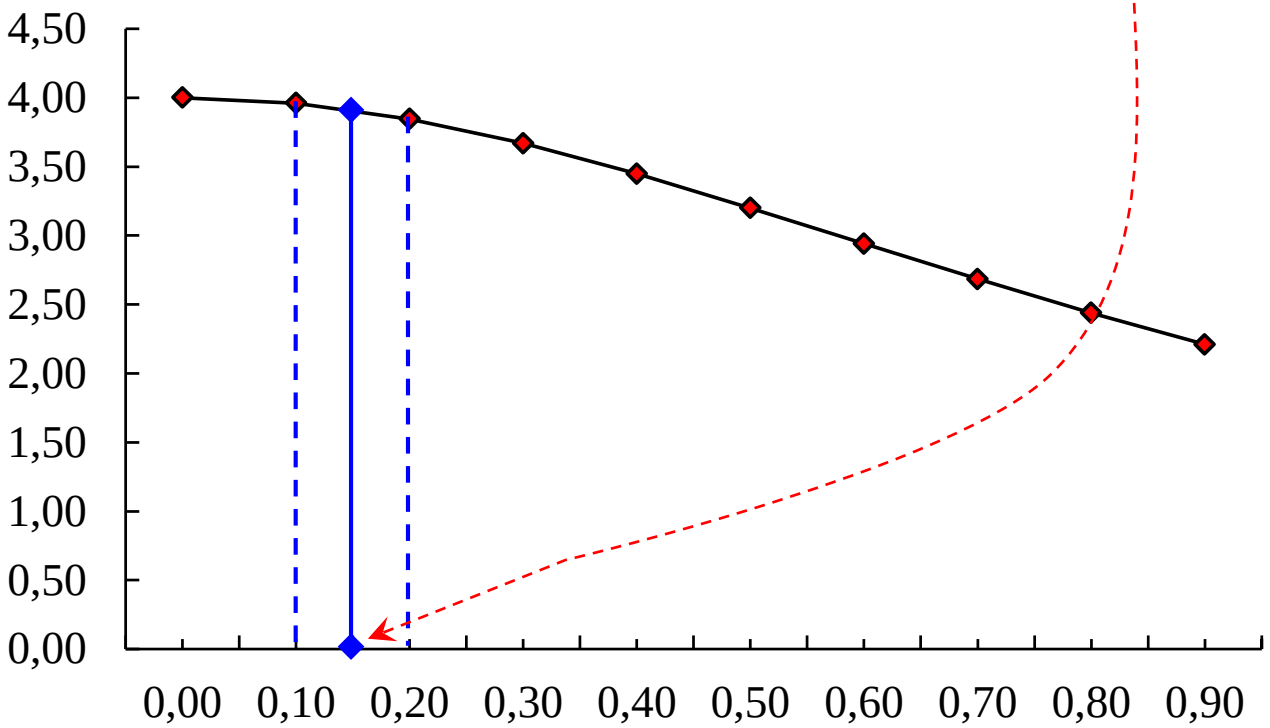
E-mail: mkurnosov@gmail.com
WWW: www.mkurnosov.net

Курс «Высокопроизводительные вычислительные системы»
Сибирский государственный университет телекоммуникаций и информатики (Новосибирск)
Осенний семестр, 2015

Задание 1: Вычисление числа π

- В качестве демонстрационной задачи предлагается приближенное вычисление числа π

$$\pi = \int_0^1 \frac{4}{1+x^2}\,dx \qquad \pi \approx h\sum_{i=1}^{n} \frac{4}{1+(h(i-0.5))^2} \qquad h = \frac{1}{n}$$



Задание 1: Вычисление числа π

- **Даны последовательная и две параллельные OpenMP-программы вычисления числа π**

  - **pi** – последовательная версия программы
  - **piomp** – для формирования результирующей суммы используется критическая секция (#pragma omp critical)
  - **piomp_red** – для формирования результирующей суммы используется директива reduction

Задание 1: вычисление числа π

- Требуется по результатам выполнения программ на кластере Jet заполнить нижеследующие таблицы
- Принятые обозначения:
  - *N* – количество потоков в программе
  - *n* – количество шагов интегрирования

1) $n = 10^7$

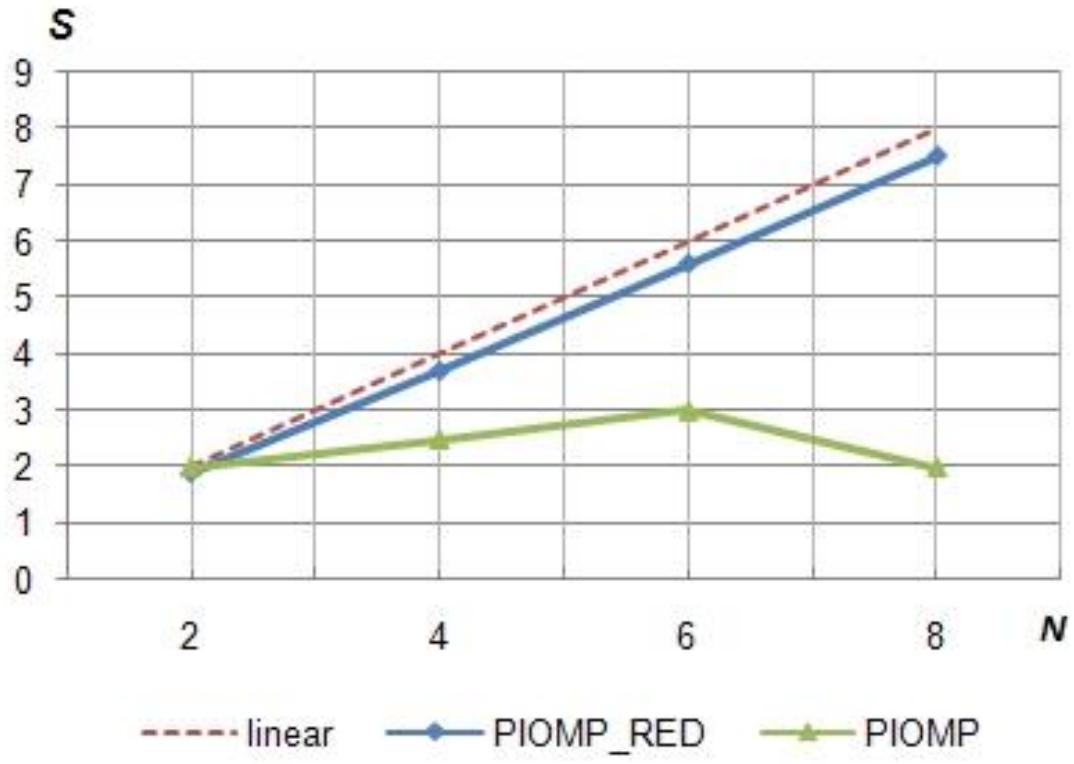
Время PI, сек.	Время PIOMP, сек.				Время PIOMP_RED, сек.			
N = 1	*N* = 2	*N* = 4	*N* = 6	*N* = 8	*N* = 2	*N* = 4	*N* = 6	*N* = 8

2) $n = 10^8$

Время PI, сек.	Время PIOMP, сек.				Время PIOMP_RED, сек.			
N = 1	*N* = 2	*N* = 4	*N* = 6	*N* = 8	*N* = 2	*N* = 4	*N* = 6	*N* = 8

Задание 1: вычисление числа π

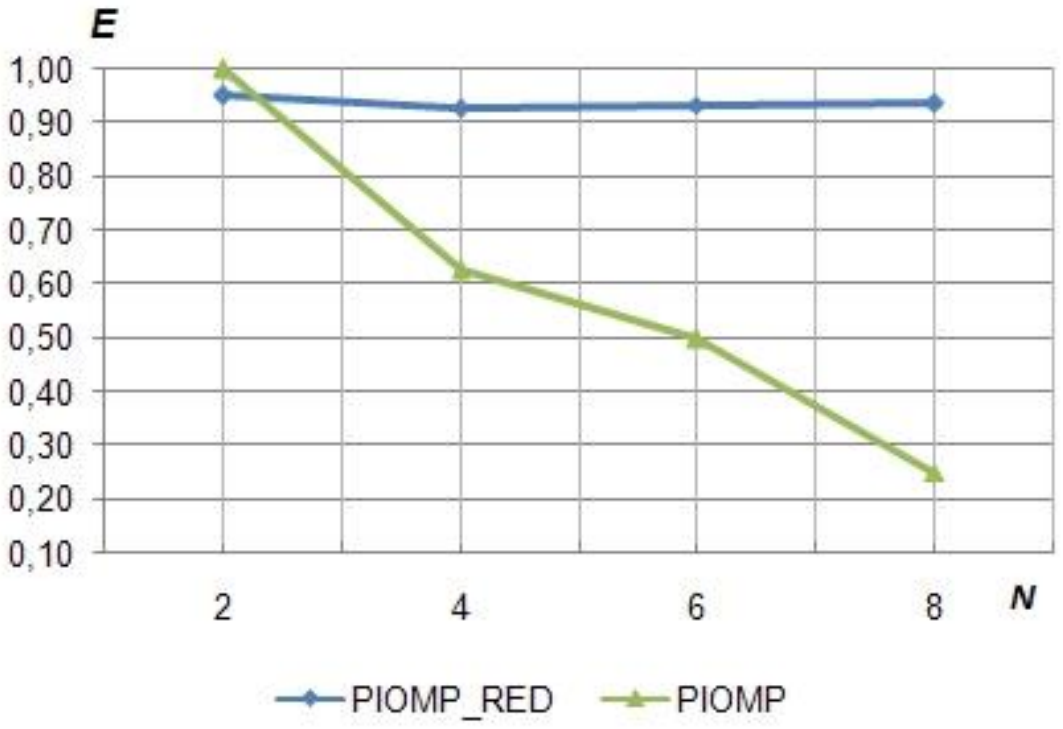
- Построить графики зависимости коэффициента *S* ускорения параллельных программ от количества *N* потоков в них (для $n = 10^8$)
- Примерный вид требуемых графиков приведен ниже



$$S = \frac{T_1}{T_N}$$

Задание 1: вычисление числа π

- Построить графики зависимости коэффициента *E* эффективности параллельных программ от количества *N* потоков в них (для $n = 10^8$)
- Примерный вид требуемых графиков приведен ниже



$$E = \frac{T_1}{N \cdot T_N}$$

Задание 2: параллельная версия DGEMM

- Разработать параллельную OpenMP-версию функции **dgemm_block** блочного умножения матриц (dgemm_block_omp)
- Построить график зависимости коэффициента ускорения программы от количества потоков в ней

Задание 3: primes

- В параллельной OpenMP-программе **primes** подсчета количества простых чисел обнаружить и исправить ошибку, связанную с не синхронизированным доступом потоков к их общей памяти (создать версию программы primes_fix)
- Оценить ускорение модифицированной программы primes_fix
- Обнаружить в программе primes_fix дисбаланс времени выполнения потоков. Добиться сбалансированного распределения вычислительной нагрузки на потоки. Оценить ускорение модифицированной программы.