

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«Московский институт электронной техники»

Кафедра СПИНТЕХ

**Отчёт по Лабораторной работе №1**  
**по предмету: «Технология программирования OpenMP»**

---

Выполнил:

Студент группы ПИН-33

Новосельцев Ст. А.

г. Москва

2021

**Сложность алгоритма:**  $O(m * n^2 * t)$ , где (m, n) - размеры матрицы, t - макс длина суммы парной суммы двух элементов строки

**10x10**

```
int main() {
    int N, M;
    cout << "Enter size (N M)" << endl;
    cin >> N >> M;
    auto testMatrix = randMatrix(N, M);
    cout << "size: " << N << "x" << M << endl;
    if (N < 10 and M < 10) {
        printMatrix(mult testMatrix);
    }

    cout << "\nin parallel mode:\n";
    time_and_result_tracing(testMatrix);

    cout << "\nin sequence mode:\n";
    time_and_result_tracing(testMatrix, parallel_mode: false);

    return 0;
}

count_seven_digit_in_pairs_sum_by_rows
Lab1
Enter size (N M)
size: 10x10

In parallel mode:
Count accessible threads: 8
Result: 876
All working time: 0.00592331 s

In sequence mode:
Count accessible threads: 1
Result: 876
All working time: 0.00278665 s
```

Многопоточная обработка медленнее в  $\approx 21.9$  раз, чем последовательная, за счет затрат на создание и закрытие потоков для легковесной задачи

**100x100**

```
int main() {
    int N, M;
    cout << "Enter size (N M)" << endl;
    cin >> N >> M;
    auto testMatrix = randMatrix(N, M);
    cout << "size: " << N << "x" << M << endl;
    if (N < 10 and M < 10) {
        printMatrix(mult testMatrix);
    }

    cout << "\nin parallel mode:\n";
    time_and_result_tracing(testMatrix);

    cout << "\nin sequence mode:\n";
    time_and_result_tracing(testMatrix, parallel_mode: false);

    return 0;
}

main
/home/s10a0novoseltcev/Dev/C++/OpenMP/Lab1/cmake-build-debug/Lab1
Enter size (N M)
size: 100x100

In parallel mode:
Count accessible threads: 8
Result: 866980
All working time: 0.0115488 s

In sequence mode:
Count accessible threads: 1
Result: 866980
All working time: 0.0655884 s
```

Многопоточная обработка быстрее в  $\approx 5.7$  раз, чем последовательная.

Данный результат можно обосновать так:

- В последовательном фрагменте производится около  $\{t, \text{ в среднем } \approx 10\}$  млн. итераций.
- В параллельном один поток совершает в среднем в 8 раз меньше итераций.
- Но прирост по времени не равен 8: нить-мастер сперва тратит время на создание потоков, затем ожидает завершения всех их и собирает выделенные ресурсы.
- Тем самым время завершения параллельного фрагмента равняется времени последнего завершения потока, что приведет к простаиванию потоков из-за необходимости их дальнейшей синхронизации.

# 1000x1000

```
template<typename T>
ostream& operator<< (ostream& lhs, const list<T*> rhs) {...}

template<typename T>
void printMatrix(list<list<T*>> matrix) {...}

int count_seven_digit_in_pairs_sum_by_rows(const list<list<int*>> matrix, bool parallel_mode=true) {...}

void time_and_result_tracing(const list<list<int*>> matrix, bool parallel_mode=true) {...}

int main() {
    int N, M;
    cout << "Enter size (N M)" << endl;
    cin >> N >> M;
    auto testMatrix = randMatrix(N, M);
    cout << "size: " << N << "x" << M << endl;
    if (N < 10 and M < 10) {
        count_seven_digit_in_pairs_sum_by_rows
    }
}
```

Lab1

/home/stadnovoseltcev/Dev/C++/OpenMP/Lab1/cmake-build-debug/Lab1

Enter size (N M)

size: 1000x1000

In parallel mode:

Count accessible threads: 8

0 [|||||] 100.0% 3990MHz 53°C Battery: 93.0% (Running on A/C)

1 [|||||] 0.8% 3999MHz 71°C Uptime: 5 days, 11:47:03

2 [|||||] 1.5% 3992MHz 50°C Tasks: 177, 991 thr; 0 running

3 [|||||] 0.8% 3955MHz 54°C Disk IO: 0.7% read: 0K write: 1.30M

4 [|||||] 100.0% 3999MHz 71°C Network: rx: 0Kib/s tx: 0Kib/s (2/2 packets)

5 [|||||] 0.8% 3986MHz 50°C Avg [|||||] 13.7% 3980MHz 71°C

6 [|||||] 0.8% 3991MHz 54°C Mem [|||||] 17.206/23.36

PID	S	CPUS	ML	Mem	Time	Virtual	Res	SW	Command	
171640	S	0.0	80	7.7	0:00.00	937M	1837M	399M	+	Lab1
171965	R	99.3	8	0.1	0:02.53	540M	3490M	3468		Lab1
171966	R	97.7	8	0.1	0:02.54	540M	3490M	3468		Lab1
171967	R	99.3	8	0.1	0:02.57	540M	3490M	3468		Lab1
171968	R	100.	8	0.1	0:02.56	540M	3490M	3468		Lab1
171969	R	95.2	8	0.1	0:02.46	540M	3490M	3468		Lab1
171970	R	99.3	8	0.1	0:02.56	540M	3490M	3468		Lab1
171971	S	98.5	8	0.1	0:02.55	540M	3490M	3468		Lab1

```
template<typename T>
ostream& operator<< (ostream& lhs, const list<T*> rhs) {...}

template<typename T>
void printMatrix(list<list<T*>> matrix) {...}

int count_seven_digit_in_pairs_sum_by_rows(const list<list<int*>> matrix, bool parallel_mode=true) {...}

void time_and_result_tracing(const list<list<int*>> matrix, bool parallel_mode=true) {...}

int main() {
    int N, M;
    cout << "Enter size (N M)" << endl;
    cin >> N >> M;
    auto testMatrix = randMatrix(N, M);
    cout << "size: " << N << "x" << M << endl;
    if (N < 10 and M < 10) {
        count_seven_digit_in_pairs_sum_by_rows
    }
}
```

Lab1

/home/stadnovoseltcev/Dev/C++/OpenMP/Lab1/cmake-build-debug/Lab1

Enter size (N M)

size: 1000x1000

In parallel mode:

Count accessible threads: 8

Result: 86698803

All working time: 10.8113 s

In sequence mode:

Count accessible threads: 1

0 [|||||] 3.1% 3974MHz 53°C Battery: 93.0% (Running on A/C)

1 [|||||] 0.8% 3999MHz 71°C Uptime: 5 days, 11:47:41

2 [|||||] 1.5% 3992MHz 50°C Tasks: 177, 988 thr; 2 running

3 [|||||] 0.8% 3955MHz 54°C Disk IO: 0.0% read: 0K write: 0K

4 [|||||] 100.0% 3999MHz 71°C Network: rx: 0Kib/s tx: 0Kib/s (0/0 packets)

5 [|||||] 0.8% 3986MHz 50°C Avg [|||||] 13.7% 3980MHz 71°C

6 [|||||] 0.8% 3991MHz 54°C Mem [|||||] 17.206/23.36

PID	S	CPUS	ML	Mem	Time	Virtual	Res	SW	Command	
171640	S	0.0	75	7.7	0:00.00	937M	1837M	399M	+	Lab1
171965	S	0.0	8	0.1	0:10.27	540M	3490M	3468		Lab1
171966	S	0.0	8	0.1	0:10.64	540M	3490M	3468		Lab1
171967	S	0.0	8	0.1	0:10.75	540M	3490M	3468		Lab1
171968	S	0.0	8	0.1	0:10.68	540M	3490M	3468		Lab1
171969	S	0.0	8	0.1	0:10.54	540M	3490M	3468		Lab1
171970	S	0.0	8	0.1	0:10.63	540M	3490M	3468		Lab1
171971	S	0.0	8	0.1	0:10.69	540M	3490M	3468		Lab1

```
template<typename T>
ostream& operator<< (ostream& lhs, const list<T*> rhs) {...}

template<typename T>
void printMatrix(list<list<T*>> matrix) {...}

int count_seven_digit_in_pairs_sum_by_rows(const list<list<int*>> matrix, bool parallel_mode=true) {...}

void time_and_result_tracing(const list<list<int*>> matrix, bool parallel_mode=true) {...}

int main() {
    int N, M;
    cout << "Enter size (N M)" << endl;
    cin >> N >> M;
    auto testMatrix = randMatrix(N, M);
    cout << "size: " << N << "x" << M << endl;
    if (N < 10 and M < 10) {
        count_seven_digit_in_pairs_sum_by_rows
    }
}
```

Lab1

/home/stadnovoseltcev/Dev/C++/OpenMP/Lab1/cmake-build-debug/Lab1

Enter size (N M)

size: 1000x1000

In parallel mode:

Count accessible threads: 8

Result: 86698803

All working time: 10.8113 s

In sequence mode:

Count accessible threads: 1

Result: 86698803

All working time: 62.381 s

0 [|||||] 0.9% 900MHz 47°C Battery: 92.0% (Running on A/C)

1 [|||||] 3.4% 900MHz 46°C Uptime: 5 days, 11:48:24

2 [|||||] 3.4% 900MHz 46°C Tasks: 177, 986 thr; 1 running

3 [|||||] 1.7% 900MHz 46°C Disk IO: 0.0% read: 0K write: 0K

4 [|||||] 1.7% 900MHz 47°C Network: rx: 0Kib/s tx: 0Kib/s (2/2 packets)

5 [|||||] 1.7% 900MHz 46°C Avg [|||||] 2.1% 900MHz 47°C

6 [|||||] 1.7% 900MHz 46°C Mem [|||||] 17.210/23.36

PID	S	CPUS	ML	Mem	Time	Virtual	Res	SW	Command	
171640	S	0.0	72	7.7	0:00.00	937M	1837M	399M	+	Lab1
171965	S	0.0	8	0.1	0:10.63	540M	3490M	3468		Lab1
171966	S	0.0	8	0.1	0:10.64	540M	3490M	3468		Lab1
171967	S	0.0	8	0.1	0:10.75	540M	3490M	3468		Lab1
171968	S	0.0	8	0.1	0:10.68	540M	3490M	3468		Lab1
171969	S	0.0	8	0.1	0:10.54	540M	3490M	3468		Lab1
171970	S	0.0	8	0.1	0:10.63	540M	3490M	3468		Lab1
171971	S	0.0	8	0.1	0:10.69	540M	3490M	3468		Lab1

Многопоточная обработка быстрее в  $\approx 5,77$  раз, чем последовательная.

Но данный случай отличается от случая-100x100 тем, что количество итераций отличается в 1000 раз.