

OpenMP



Определение

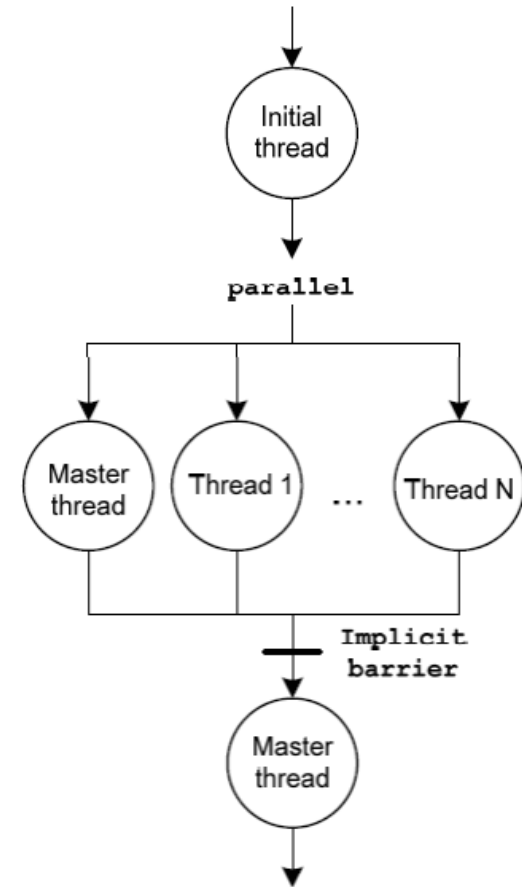
OpenMP (Open Multi-Processing)

Стандарт, определяющий набор директив компилятора, библиотечных процедур и переменных среды окружения для создания многопоточных программ.



Структура OpenMP программы

- Программа представляется в виде последовательных участков кода и параллельных секций
- Каждый поток имеет номер: 0, 1, 2, ...
- Главный поток (master) имеет номер 0



Библиотечные процедуры

- `int omp_get_thread_num()`
- `int omp_get_num_threads()`
- `void omp_set_num_threads(int n)`
- `double omp_get_wtime()`



Директивы OpenMP

`#pragma omp <директива> [опция1] [опция2]...`

- `Parallel;`
- `For;`



Директива Parallel

Опции:

- **if**
- num_threads
- shared

```
#pragma omp parallel if (var)
{
    /* Код выполняется потоками, если var = true */
}
```



Директива Parallel

Опции:

- if
- **num_threads**
- shared

```
#pragma omp parallel num_threads(n)
{
    /* Создается n потоков */
}
```



Директива Parallel

Опции:

- if
- num_threads
- **shared**



Директива For

Опции:

- **private**
- reduction
- nowait

```
#pragma omp parallel shared(A, B, C) private(i, n)
{
    n = omp_get_thread_num();

    #pragma omp for
    for (i = 0; i < 10; i++)
    {
        C[i] = A[i] + B[i];
        printf("Нить \\%d сложила элементы с номером %d\\n", n, i);
    }
}
```



Директива For

Опции:

- private
- **reduction**
- nowait

```
int i, s = 0;  
#pragma omp parallel for reduction(+ : s)  
for(i = 0; i < 100; ++i)  
    s = i;  
  
printf("Sum: %d\n", s);
```



Директива For

Опции:

- private
- reduction
- **nowait**



Компилирование программ

Компиляция выполняется командами с добавлением параметра `-openmp`:

- `mpicc -openmp example.c`
- `mpicxx -openmp example.cxx`



Запуск задач на кластере

В интерактивном режиме:

srun -n <число процессов> **-t** <время>
<имя_программы >

В пакетном режиме:

sbatch -n <число процессов> **-t** <время>
<имя_скрипта>



Скрипт

Создание скрипта run.sh

```
#!/bin/sh
```

```
#SBATCH -n 1 -t 1 --cpus-per-task 4
```

```
export OMP_NUM_THREADS=4
```

```
srun ./a.out
```

Запуск скрипта

```
sbatch run.sh
```



Пример

```
float vectorSumm(float*A, float*V,float* C,int n) {  
    double start = omp_get_wtime();  
    for(int i = 0; i < n; i++)  
        C[i] = A[i] + V[i];  
    return (float)(omp_get_wtime()-start);  
}
```

```
float vectorSummOpenMP(float*A, float*V,float* C,int n) {  
    int i;  
    double start = omp_get_wtime();  
    #pragma omp parallel default(none) shared(A,V,C,n) private(i)  
    {  
        #pragma omp for  
        for(i = 0; i < n; i++)  
            C[i] = A[i] + V[i];  
    }  
    return (float)(omp_get_wtime()-start);  
}
```



Результаты

$T_{1, c}$	$T_{2, c}$	$T_{4, c}$	S_2	$E_2, \%$	S_4	$E_4, \%$
0.26	0.14	0.07	1.86	93	3.7	92

