



↗ Computer Science МФТИ

# Add features in OpenMP

Субботина Анна  
МФТИ

# Hybrid OpenMP/MPI

Порядок действий:

- Сначала программа параллелится с помощью MPI – на сокетах
- Далее внутри сокета (по ядрам процессоров) программа параллелится с помощью OpenMP

Смысл:

- можно перенести часть обменов в кэш процессора без потерь времени на пересылку (аналог общей памяти) для OpenMP, что в конечном счете приводит к ускорению расчетов

# Hybrid OpenMP/MPI

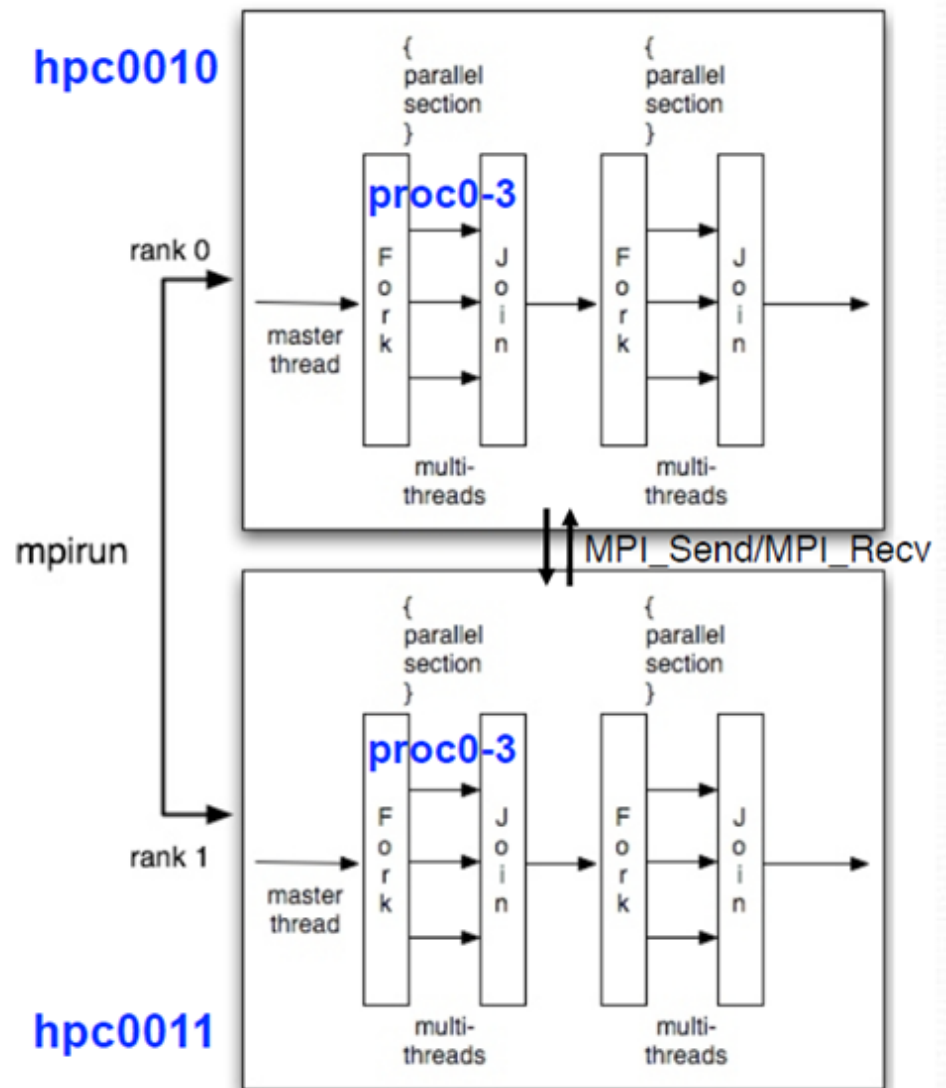
**каждый  
процесс  
порождает  
множество  
потоков**

**для запуска:**

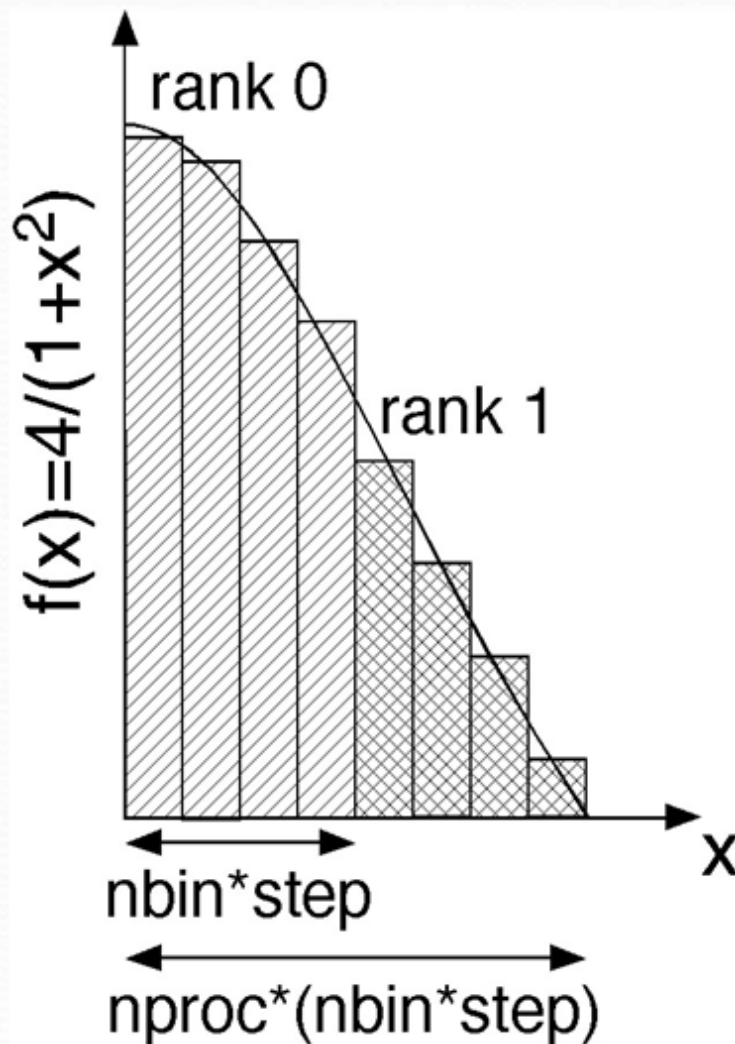
**`mpirun -np 2`**

**в коде:**

**`omp_set_num_threads(3);`**



# Вычисление числа $p_i$



- Каждый MPI процесс интегрирует по отрезку  $1/n_{\text{proc}}$ , считая площадь каждого из  $n_{\text{bin}}$  прямоугольников шириной  $\text{step}$
- В тоже время внутри каждого MPI процесса,  $n_{\text{threads}}$  потоков рассчитывают частичные суммы на OpenMP

```
#define NBIN 100000
#define MAX_THREADS 8
void main(int argc, char **argv) {
    int nbin, myid, nproc, nthreads, tid;
    double step, sum[MAX_THREADS] = {0.0}, pi = 0.0, pig;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &myid);
    MPI_Comm_size(MPI_COMM_WORLD, &nproc);
    nbin = NBIN/nproc; step = 1.0/(nbin*nproc);
    #pragma omp parallel private(tid) {
        int i; double x;
        nthreads = omp_get_num_threads(); tid = omp_get_thread_num();
        for (i = nbin*myid + tid; i < nbin*(myid+1); i += nthreads) {
            x = (i + 0.5) * step; sum[tid] += 4.0 / (1.0 + x * x);
        }
        printf("rank tid sum = %d %d %e\n", myid, tid, sum[tid]);
    }
    for (tid = 0; tid < nthreads; tid++) pi += sum[tid] * step;
    MPI_Allreduce(&pi, &pig, 1, MPI_DOUBLE, MPI_SUM, MPI_COMM_WORLD);
    if (myid == 0) printf("PI = %f\n", pig);
    MPI_Finalize();
}
```



# Вычисление числа $\pi$

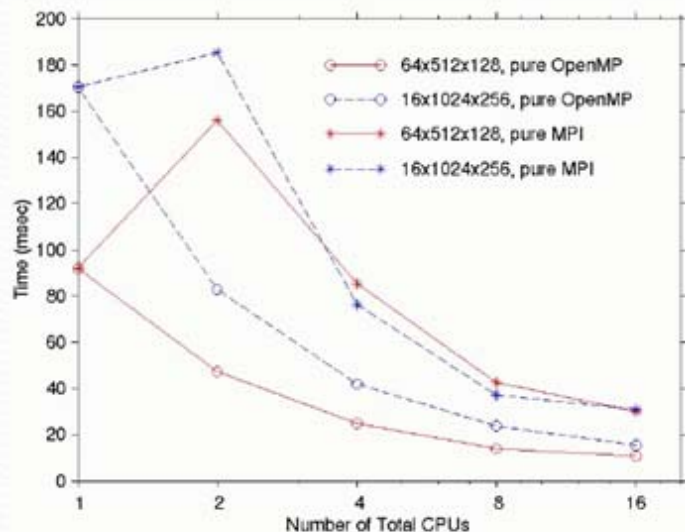
- Компиляция

```
mpicc -o hpi hpi.c -openmp
```

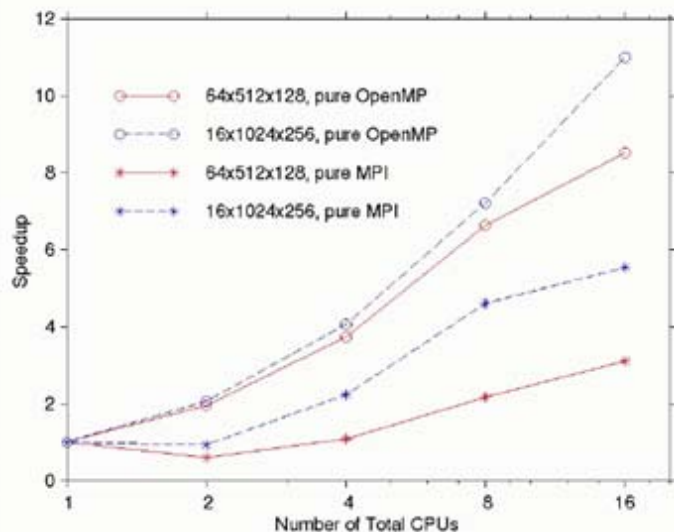
- Вывод

```
rank tid sum = 1 1 6.434981e+04  
rank tid sum = 1 0 6.435041e+04  
rank tid sum = 0 0 9.272972e+04  
rank tid sum = 0 1 9.272932e+04  
PI = 3.141593
```

# Сравнение производительности



***OpenMP vs. MPI (16 CPUs)***  
***64x512x128: 2.76 быстрее***  
***16x1024x256: 1.99 быстрее***



# MM5 Regional Weather Prediction Model

Метод	время на коммуникацию (сек)	общее время счета (сек)
64 MPI tasks	494	1755
16 MPI tasks with 4 threads/task	281	1505

*время на коммуникацию уменьшилось на 85%  
так же уменьшилось время расчета  
программы*





Вопросы?