Използване на OpenMP.

Курс „Паралелно програмиране“

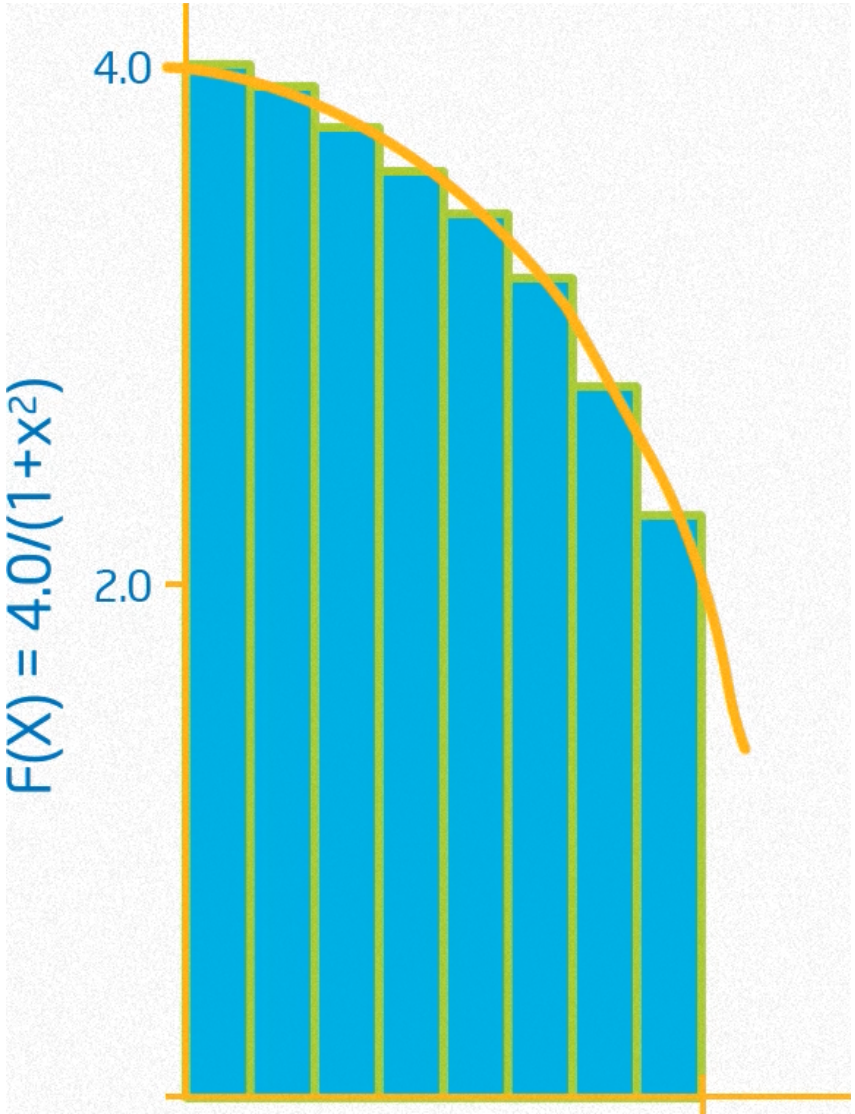
ИНСТИТУТ *за* СЪВРЕМЕННИ
ФИЗИЧЕСКИ ИЗСЛЕДВАНИЯ

Стоян Мишев

План

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$



Решение без паралелни изчисления

```
1 #include <stdio.h>
2 #include <omp.h>
3 static long num_steps = 100000000;
4 double step;
5 int main ()
6 {
7             int i;
8             double x, pi, sum = 0.0;
9             double start_time, run_time;
10            step = 1.0/(double) num_steps;
11            start_time = omp_get_wtime();
12            for (i=1;i<= num_steps; i++){
13                    x = (i-0.5)*step;
14                    sum = sum + 4.0/(1.0+x*x);
15            }
16
17            pi = step * sum;
18            run_time = omp_get_wtime() - start_time;
19            printf("\n pi with %ld steps is %lf in %lf
                  seconds\n ",num_steps,pi,run_time);
20 }
```

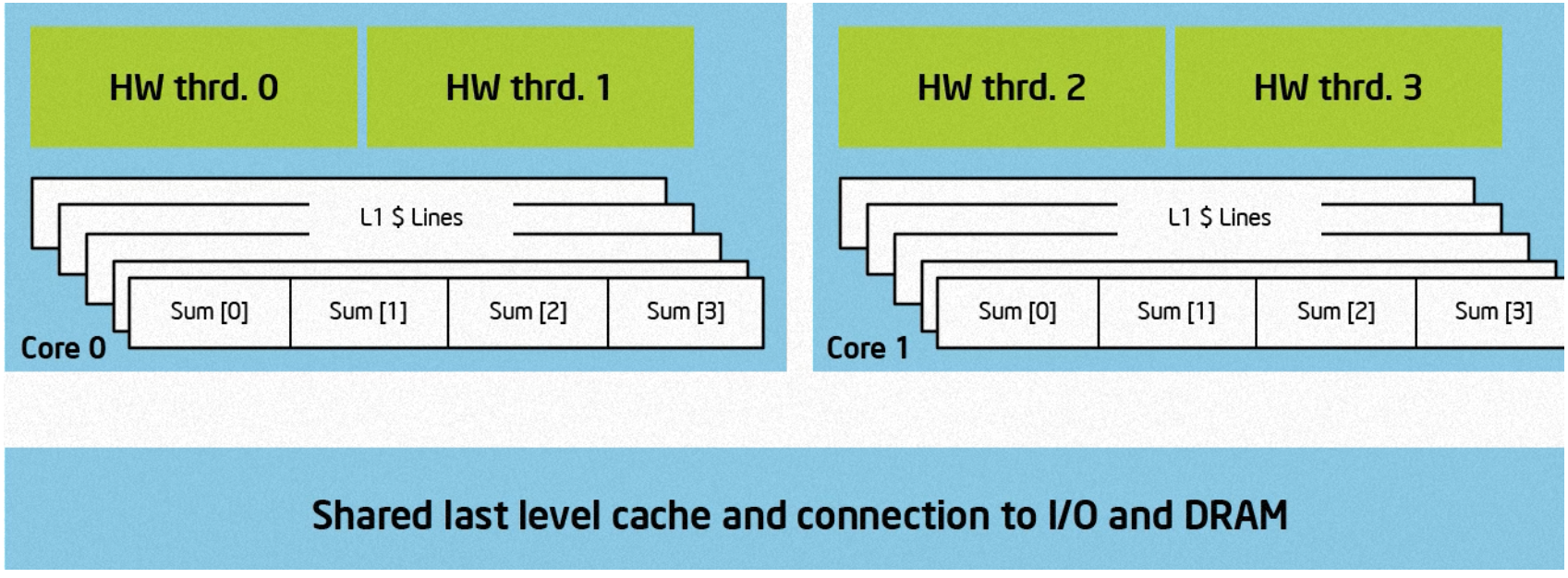
```
1 #define MAX_THREADS 4
2 static long num_steps = 100000000;
3 double step;
4 int main ()
5 {
6             int i,j;
7             double pi, full_sum = 0.0;
8             double start_time, run_time;
9             double sum[MAX_THREADS];
10            step = 1.0/(double) num_steps;
11    for (j=1;j<=MAX_THREADS ;j++) {
12       omp_set_num_threads(j);
13       full_sum=0.0;
14       start_time = omp_get_wtime();
15       #pragma omp parallel
16       {
17          int i;
18            int id = omp_get_thread_num();
19            int numthreads = omp_get_num_threads();
20            double x;
21            sum[id] = 0.0;
```

```
1           if (id == 0)
2                 printf(" num_threads = %d",numthreads);
3
4             for (i=id;i< num_steps; i+=numthreads){
5                   x = (i+0.5)*step;
6                   sum[id] = sum[id] + 4.0/(1.0+x*x);
7             }
8         }
9           for(full_sum = 0.0, i=0;i<j;i++)
10              full_sum += sum[i];
11        pi = step * full_sum;
12        run_time = omp_get_wtime() - start_time;
13        printf("\n pi is %f in %f seconds %d thrds \n",pi
              ,run_time,j);
14    }
15 }
```

Threads	1st SPMD
1	1.86
2	1.03
3	1.08
4	0.97

Решение с паралелни изчисления 1. Мащабируемост. 7

Threads	1st SPMD
1	1.86
2	1.03
3	1.08
4	0.97



```
#include <omp.h>
static long num_steps = 100000;        double step;
#define    PAD      8          // assume 64 byte L1 cache line size
#define NUM_THREADS 2
void main ()
{           int i, nthreads;  double pi, sum[NUM_THREADS][PAD];
            step = 1.0/(double) num_steps;
            omp_set_num_threads(NUM_THREADS);
   #pragma omp parallel
   {        int i, id,nthrds;
            double x;
            id = omp_get_thread_num();
            nthrds = omp_get_num_threads();
            if (id == 0)   nthreads = nthrds;
             for (i=id, sum[id]=0.0;i< num_steps; i=i+nthrds) {
                        x = (i+0.5)*step;
                        sum[id][0] += 4.0/(1.0+x*x);
             }
    }
            for(i=0, pi=0.0;i<nthreads;i++)pi += sum[i][0] * step;
}
```

В част от програмата нишките се изпълняват последователно

```
float res;

#pragma omp parallel

[       float B; int i, id, nthrds;
        id = omp_get_thread_num();
        nthrds = omp_get_num_threads();
        for(i=id; i<niters; i+=nthrds){
            B = big_job(i);

#pragma omp critical
                res += consume (B);
        }
}
```

Решение с паралелни изчисления 2. Critical.

```
1 #include <stdio.h>
2 #include <omp.h>
3
4 #define MAX_THREADS 4
5
6 static long num_steps = 100000000;
7 double step;
8 int main ()
9 {
10              int i,j;
11              double pi, full_sum = 0.0;
12              double start_time, run_time;
13              double sum[MAX_THREADS];
14
15              step = 1.0/(double) num_steps;
16
17
18 for(j=1;j<=MAX_THREADS ;j++){
19     omp_set_num_threads(j);
20     full_sum = 0.0;
21              start_time = omp_get_wtime();
```

Решение с паралелни изчисления 2. Critical.

```
1 #pragma omp parallel private(i)
2 {
3             int id = omp_get_thread_num();
4             int numthreads = omp_get_num_threads();
5             double x;
6
7             double partial_sum = 0;
8
9 #pragma omp single
10            printf(" num_threads = %d",numthreads);
11
12            for (i=id;i< num_steps; i+=numthreads){
13                    x = (i+0.5)*step;
14                    partial_sum += + 4.0/(1.0+x*x);
15            }
16 #pragma omp critical
17                    full_sum += partial_sum;
18 }
19
20            pi = step * full_sum;
21            run_time = omp_get_wtime() - start_time;
```

Домашна работа

https://www.youtube.com/watch?v=cMWGeJyrc9w&list=PLLbPZJxtMs4ZHSamRRYCtvowRS0qIwC-I
До “Introduction to OpenMP: 09 part 1 Module 5”